

机器控制器MP900/MP2000系列

用户手册 运动程序篇

客户服务热线(帮您解决技术问题)

电话 **400-821-3680** 传真 **021-5385-2008**

周一至周五(节假日除外)9:00~11:30, 12:30~16:30 ※24小时接收传真

销售

- 安川電機(中国)有限公司
上海市黄浦区黄河路21号鸿祥大厦11-12楼
邮编: 200003
电话: 021-53852200
传真: 021-53853299
- 安川電機(中国)有限公司 北京分公司
北京市东城区东长安街1号东方广场东方经贸城西三办公楼1011室
邮编: 100738
电话: 010-85184086
传真: 010-85184082
- 安川電機(中国)有限公司 广州分公司
广州市天河区体育东路138号金利来数码网络大厦1108-10室
邮编: 510620
电话: 020-38780005
传真: 020-38780565
- 安川電機(中国)有限公司 成都分公司
成都市总府路2号时代广场B座711室
邮编: 610016
电话: 028-86719370
传真: 028-86719371

总公司

- 株式会社 安川電機
日本福岡県北九州市八幡西区城石2-1
邮编: 806-0064
电话: 0081-93-645-8800
传真: 0081-93-631-8837

YASKAWA

最终使用者若为军事单位, 或将本产品用于兵器制造等用途时, 本产品将成为《外汇及外国贸易法》规定的出口产品管制对象, 在出口时, 需进行严格检查, 并办理所需的出口手续。为改进产品, 本产品的规格, 额定值及尺寸若有变更, 恕不另行通告。
关于本资料内容的咨询, 请与本公司代理店或上述营业部门联系。

资料编号 SICP C880700 19A

© Published in China 2012年 4月编制 04-11

12-4-1

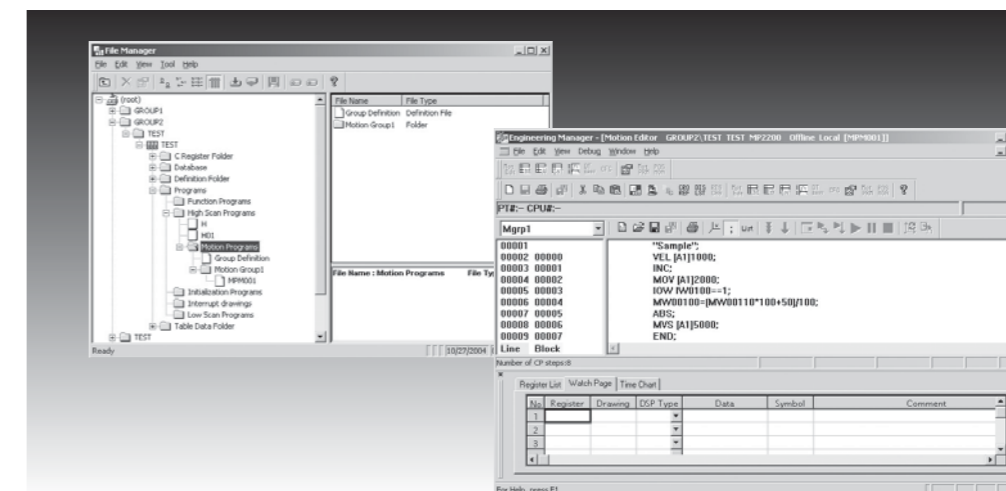
严禁转载·复制

YASKAWA

机器控制器MP900/MP2000系列

用户手册 运动程序篇

为了安全使用本产品, 请务必阅读该使用说明书。
另外, 请妥善保管该使用说明书, 并将其交至最终用户手中。



资料编号 SICP C880700 19A

与安全有关的符号说明

本手册中有关安全的内容，使用如下标识。


有关作业安全标识的叙述，其内容十分重要，请务必遵守。



表示错误使用时，将会引发危险情况，导致人身伤亡。




表示如果进行错误操作，将会导致危险情况的发生，可能会造成中等程度的受伤或轻伤，或物品损失。

另外，即使是  **注意** 标识中所述事项，有时也可能会造成严重的后果。




表示禁止（绝对不能做的事）。

例如严禁烟火时，则表示为 。



表示强制（必须要做的事）。

例如接地时，则表示为 。

图标的说明

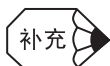
为使读者了解说明内容的区分，本书中设计了如下图标。并在必要的地方使用这些图标，以助读者理解。



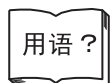
- 表示需要熟记的重要事项。
同时也表示发生警报，但还不至于造成装置损坏的注意事项。



- 表示具体程序举例、操作实例。



- 表示补充事项或记住后会便于使用的功能。



- 表示对难于理解的用语进行解释，以及对事先没有说明而后出现的用语进行说明。

本手册的简介

■ 本手册详细地说明了机器控制器 MP900/MP2000 系列（以下称为 MP 系列）运动命令的相关信息。对 MP 系列的下列项目进行了说明。

- 概要、规格及编程方法
- 基本编程
- 高级编程
- 顺序命令
- 变量

■ 为了能正确使用 MP 系列系统，请仔细阅读本手册。请妥善保管本手册，以便需要时参阅。

本手册的资料构成

■ MP900 系列有 MP910、MP920、MP930、MP940 四种。

MP2000 系列有 MP2100 和 MP2300 两种。

资料构成根据上述产品构成修订。相关手册在下一页介绍，敬请参阅。

相关手册

- 相关手册包括下表所示的内容。请根据需要进行阅读。
- 请在充分理解产品的规格、使用限制等条件的基础上使用产品。

资料名称	资料编号	内容
机器控制器 MP930 用户手册 设计与维护篇	SIZ-C887-1.1	对 MP930 的功能、使用、安装步骤等使用方法进行了说明。
机器控制器 MP920 用户手册 设计与维护篇	SIZ-C887-2.1	对 MP920 的功能、使用、安装步骤等使用方法进行了说明。
机器控制器 MP910 用户手册 设计与维护篇	SIZ-C887-3.1	对 MP910 的功能、使用、安装步骤等使用方法进行了说明。
机器控制器 MP940 用户手册 设计与维护篇	SIZ-C887-4.1	对 MP940 的功能、使用、安装步骤等使用方法进行了说明。
机器控制器 MP2100 用户手册 设计与维护篇	SIJPC88070001	对 MP2100 的功能、使用、安装步骤等使用方法进行了说明。
机器控制器 MP920 用户手册 运动模块篇	SI-C887-2.5	详细说明了 MP920 运动模块 (SVA-01、PO-01) 的功能、规格、使用方法。
机器控制器 MP920 用户手册 通讯模块篇	SIZ-C887-2.6	详细说明了 MP920 通信模块 (215IF、217IF、218IF) 的功能、规格、使用方法。
机器控制器 MP2300 用户手册 基本模块篇	SIJPC88070003	对 MP2300 的功能、使用、安装步骤等使用方法进行了说明。
机器控制器 MP2300 用户手册 通讯模块篇	SIJPC88070004	详细说明了 MP2300 通信模块的功能、规格、使用方法。
机器控制器 MP900/MP2000 系列 用户手册 梯形程序篇	SI-C887-1.2	对 MP 系列的梯形程序中使用的编程语言进行了说明。
机器控制器 MP900/MP2000 系列 编程装置用软件 MPE720 用户 手册	SIJPC88070005	详细说明了 MP 系列的编程系统 :MPE720 的安装和操作方法。

本手册的使用方法

■ 本手册的使用对象

本手册以下列人员为对象。

- 进行 MP 系列系统设计的人员。
- 进行 MP 系列运动程序编程的人员。

■ 缩略语及缩写符号

本手册使用如下所示的缩略语及缩写符号。

- PP: 编程面板
- PC: 可编程控制器
- MP900:MP910、MP920、MP930, MP940 的总称
- MP2000:MP2100、MP2300 的总称

安全注意事项

本节对如何正确使用机器的安全注意事项进行了说明。在安装、运行、维护和检查前，请务必熟读本手册和其它附属资料，以便正确使用。请在充分了解机器的相关知识、安全信息和注意事项后使用。

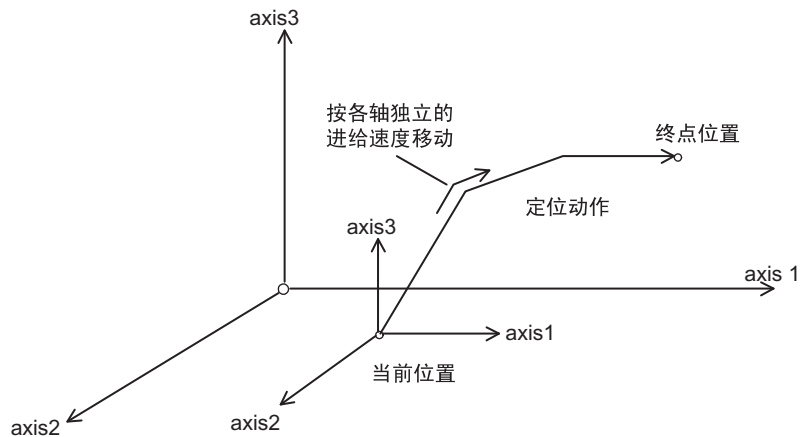
■ 使用注意事项

⚠ 注意

- 在进行下列轴移动命令编程时，请务必进行轨迹检查，以免设备和工件之间的碰撞。

需要检查的轴移动命令

- “定位 (MOV)” 命令
- “直线插补 (MVS)” 命令
- “圆弧插补 (MCC、MCW)” 命令
- “螺旋插补 (MCC、MCW)” 命令
- “时间指定定位 (MVT)” 命令
- “跳过 (SKP)” 命令
- “原点复归 (ZRN)” 命令
- “外部定位 (EXM)” 命令



“定位 (MOV)” 命令的基本轨迹举例

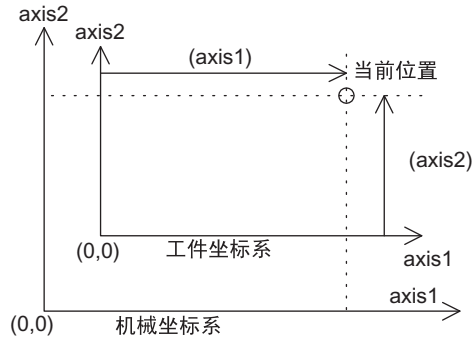
如果疏于上述检查，则有可能造成设备损坏，甚至有可能导致人身伤害事故。

⚠ 注意

- 如果误发出下列坐标命令，则之后的移动动作会完全不同。在运行前，请务必检查是否正确指定了上述命令。

需要检查的坐标命令

- “绝对值模式 (ABS)” 命令
- “增量值模式 (INC)” 命令
- “当前值变更 (POS)” 命令
- “机械坐标指令 (MVM)” 命令



“当前值变更 (POS)” 命令产生的新工件坐标系举例

如果疏于上述检查，则有可能造成工具损坏，甚至有可能导致人身伤害事故。

■ 一般注意事项

使用时的注意事项

- MP 系列并非为在有生命危险的情况下使用的器械或系统而设计、制造的。
要把本资料所记载的产品用于移动设备、医疗设备、航空航天设备、原子能控制设备、海底中转设备或系统等特殊用途时，请向本公司营业部门咨询。
- MP 系列是在严格的质量管理下制造的，但将 MP 系列用于可能引起生命危险、重大损失的重要设备时，请设置安全装置以避免事故的发生。
- 本手册中使用的插图和照片，仅为典型示例，可能会与交付的产品有所不同。
- 因产品的改良、规格变更以及为了提高手册的使用便利性，本手册可能会随时变更。
- 变更后，手册的资料编号也随之更新，并作为修订版发行。关于修订版的版数在封底有记载。
- 因损坏、丢失需订购本手册时，请与本公司代理店或手册封底记载的最近的营业网点联系，并告知资料编号。
- 安装在产品上的铭牌出现磨损或破损时，请向本公司代理店或手册封底记载的最近的营业网点订购。
- 经客户改造的产品，不在本公司质量保证范围内。因改造产品引起的一切伤害和损失，本公司概不负责。

目录

与安全有关的符号说明	iii
图标的说明	iv
本手册的简介	v
本手册的资料构成	v
相关手册	vi
本手册的使用方法	vii
安全注意事项	viii
第 1 章 运动程序概要	1-1
1.1 何谓运动程序	1-2
1.1.1 运动程序的作用	1-2
1.1.2 功能一览	1-4
1.1.3 运动程序概要	1-13
1.1.4 程序的并列执行	1-20
1.1.5 程序编辑	1-21
1.2 编程方法	1-22
1.2.1 输入格式	1-22
1.2.2 控制轴	1-26
1.2.3 进给速度	1-31
1.2.4 运动命令一览	1-34
第 2 章 运动命令	2-1
2.1 轴移动命令	2-2
2.1.1 定位 (MOV)	2-2
2.1.2 直线插补 (MVS)	2-5
2.1.3 圆弧插补 (MCW、MCC)	2-8
2.1.4 螺旋插补 (MCW、MCC)	2-13
2.1.5 原点复归 (ZRN)	2-15
2.1.6 跳过功能 (SKP)	2-19
2.1.7 时间指定定位 (MVT)	2-20
2.1.8 外部定位 (EXM)	2-22
2.2 控制命令	2-23
2.2.1 绝对值 (ABS) 模式	2-23
2.2.2 增量值 (INC) 模式	2-25
2.2.3 当前值变更 (POS)	2-26
2.2.4 坐标平面指定 (PLN)	2-28
2.2.5 机械坐标指令 (MVM)	2-29
2.2.6 程序当前位置更新 (PLD)	2-30
2.2.7 待时 (TIM)	2-31
2.2.8 程序的结束 (END)	2-32
第 3 章 高级编程	3-1
3.1 高级控制命令	3-2
3.1.1 入位检查 (PFN)	3-2
3.1.2 第 2 入位检查 (INP)	3-4
3.1.3 单块忽视 (SNG)	3-6
3.1.4 用户函数调用 (UFC)	3-7

3.1.5	输入输出变量待机 (IOW)	3-15
3.1.6	子程序调用 (MSEE)	3-16
3.1.7	子程序结束 (RET)	3-17
3.1.8	1 个扫描周期的 WAIT (待机) 命令 (EOX)	3-18
3.1.9	分支命令 (IF ELSE IEND)	3-19
3.1.10	循环命令 (WHILE WEND)	3-20
3.1.11	并列执行命令 (PFORK, JOINTO, PJOINT)	3-22
3.1.12	选择执行命令 (SFORK, JOINTO, SJOINT)	3-26
3.2	速度与加减速命令	3-28
3.2.1	加速时间变更 (ACC)	3-28
3.2.2	减速时间变更 (DCC)	3-30
3.2.3	S 字时间参数变更 (SCC)	3-32
3.2.4	进给速度变更 (VEL)	3-34
3.2.5	插补进给速度比率设定 (IFP)	3-36
3.2.6	插补进给最高速度设定 (FMX)	3-37
3.2.7	插补加速时间变更 (IAC)	3-39
3.2.8	插补减速时间变更 (IDC)	3-41

第 4 章 顺序命令 - - - - - 4-1

4.1	顺序命令概要	4-3
4.1.1	运算命令	4-3
4.1.2	数值运算组合	4-4
4.1.3	逻辑运算组合	4-4
4.2	数值运算	4-5
4.2.1	赋值 (=)	4-5
4.2.2	加法运算 (+)	4-6
4.2.3	减法运算 (-)	4-7
4.2.4	乘法运算 (*)	4-8
4.2.5	除法运算 (/)	4-9
4.2.6	余数 (MOD)	4-10
4.3	逻辑运算	4-11
4.3.1	逻辑或 ()	4-11
4.3.2	逻辑与 (&)	4-12
4.3.3	逻辑异或 (^)	4-13
4.3.4	取反 (!)	4-14
4.4	数值比较	4-15
4.4.1	数值比较命令 (==、<>、>、<、>=、<=)	4-15
4.5	数据操作	4-16
4.5.1	位右移 (SFR)	4-16
4.5.2	位左移 (SFL)	4-17
4.5.3	块传送 (BLK)	4-18
4.5.4	清除 (CLR)	4-19
4.6	基本函数	4-20
4.6.1	正弦 (SIN)	4-20
4.6.2	余弦 (COS)	4-21
4.6.3	正切 (TAN)	4-22
4.6.4	反正弦 (ASN)	4-23
4.6.5	反余弦 (ACS)	4-24
4.6.6	反正切 (ATN)	4-25
4.6.7	平方根 (SQT)	4-26

4.6.8	BCD → BIN (BIN)	-4-28
4.6.9	BIN → BCD (BCD)	-4-29
4.6.10	指定位 ON (S { })	-4-30
4.6.11	指定位 OFF (R { })	-4-31

第 5 章 变量 (寄存器) - - - - - 5-1

5.1	变量	5-2
5.1.1	变量概要	5-2
5.1.2	全局变量和局部变量	5-4
5.2	变量使用方法	5-7
5.2.1	系统变量 (S 寄存器)	5-7
5.2.2	数据变量 (M 寄存器)	5-8
5.2.3	输入变量 (I 寄存器)	5-9
5.2.4	输出变量 (O 寄存器)	5-12
5.2.5	C 变量 (C 寄存器)	5-15
5.2.6	D 变量 (D 寄存器)	5-16

附录 - - - - - 附录 -1

附录 A.	运动命令一览	附录 -2
A.1.	运动命令一览	附录 -2
附录 B.	运动程序警报	附录 -7
B.1.	运动程序警报的存储地点	附录 -7
B.2.	运动程序警报一览	附录 -9

第 1 章

运动程序概要

本章对运动程序的制作方法进行说明。运动程序通过 MPE720(编程装置)制作,向 MP 系列传送后执行运行。

1.1 何谓运动程序	1-2
1.1.1 运动程序的作用	1-2
1.1.2 功能一览	1-4
1.1.3 运动程序概要	1-13
1.1.4 程序的并列执行	1-20
1.1.5 程序编辑	1-21
1.2 编程方法	1-22
1.2.1 输入格式	1-22
1.2.2 控制轴	1-26
1.2.3 进给速度	1-31
1.2.4 运动命令一览	1-34

1.1 何谓运动程序

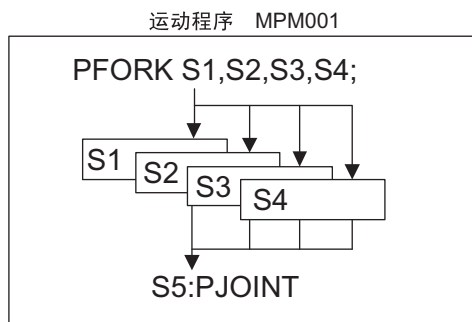
本节对运动程序的概要进行说明。在进行编程之前请务必阅读。

1.1.1 运动程序的作用

可使用 MP 系列对工业机械所需的特定运动进行编程。
主要运动程序的特点如下所示。

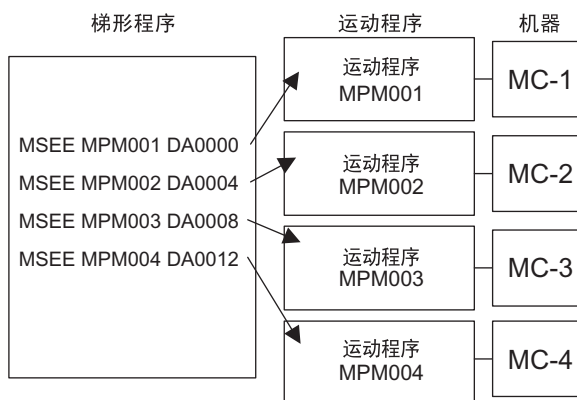
(1) 并列执行

- 在一个运动程序内，可并列执行四个程序。
- 使用 PFORK 命令最多可并列执行四个程序。
- 通过四个并列程序，可自由地组合控制坐标轴。



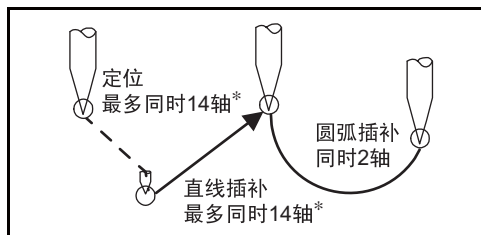
(2) 同时控制多个机器

用 MP 系列的一台机器可同时控制多台机器。



(3) 多种移动命令

- 定位 最多 14 轴 *
 - 直线插补 最多 14 轴 *
 - 圆弧插补 2 轴
 - 螺旋插补 3 轴
- 等基本命令可简单地指令。



* 最多 14 轴为 MP930 的示例。

- MP910 可控制：控制轴=最多 28 轴，直线插补=最多 14 轴。
- MP920 可控制：控制轴=最多 60 轴，直线插补=最多 16 轴。
- MP2100 可控制：控制轴=最多 16 轴，直线插补=最多 16 轴。
- MP2300 可控制：控制轴=最多 16 轴，直线插补=最多 16 轴。

(4) 多种运算命令可进行自由计算

- 整型加减乘除
- 实型加减乘除
- 逻辑运算
- 三角函数
- 指数
- 对数

运算命令举例

```
MW81D=AB01H*SIN(100)+50;
MW5=BCD(MW0001);
IF MF467 <> MF897;
IF MW0001 > 7;
IF OB0 == 1;
OB0 = (IB0|IB2|IB3) & IB1;
MW6 = SQT(MW809);
```

(5) 多种控制命令

- 条件分支命令
 - 循环命令
 - 定时命令 (TIM)
 - 子程序 (MSEE)
 - 并列执行命令 (PFORK)
 - 选择执行命令 (SFORK)
- 等控制命令可使用。

条件分支命令

```
IF <条件表达式>
·
· 条件成立时的处理
·
·
ELSE
·
· 条件不成立时的处理
·
·
IEND
```

循环命令

```
WHILE <条件表达式>
·
· 处理
·
·
WEND
```

1.1.2 功能一览

(1) MP910 的运动控制功能规格

MP910 的运动控制功能规格如下表所示。

表 1.1 MP910 的运动控制功能规格

项目		规格
控制轴数		1 ~ 最多 28 轴 (14 轴 /1 接口, 最多 2 接口)
控制规格	PTP 控制	直线、旋转、无限长
	插补	直线最多 14 轴、圆弧 2 轴、螺旋 3 轴
	速度控制	无
	转矩限制	有 (仅根据参数设定的转矩限制)
指令单位		mm、inch、deg、pulse
指令最小设定单位		1, 0.1, 0.01, 0.001, 0.0001, 0.00001
最大指令值		-2147483648 ~ +2147483647 (带符号 32 位数)
速度指令单位		mm/min、inch/min、deg/min、pulse/min
加减速型		直线、非对称、S 字 (在定位时不能采用非对称)
额定速度比功能		定位: 以轴为单位 0.01 ~ 327.67% 插补: 以组为单位 0.01 ~ 327.67%
坐标系		直角坐标
原点复归		4 种 DEC+C 相、原点 LS、DEC+ 原点 LS、C 相 有原点设定功能
程序	语言	专用运动语言
	任务数	最多可同时并列运行 4 个程序。
	程序数	最多 256 个
	程序容量	相当于 80 kB (字符) (根据梯形程序的使用量有增减, 最大 100 kB)
命令符号		轴移动命令 : 8 种 MOV、MVS、MCW、MCC、ZRN、SKP、MVT、EXM 基本控制命令 : 6 种 ABS、INC、POS、PLN、MVM、PLD 速度与加减速命令 : 7 种 ACC、SCC、VEL、IAC、IDC、IFP、FMX 高级控制命令 : 4 种 PFN、INP、SNG、UFC 控制命令 : 10 种 MSEE、TIM、IOW、END、RET、EOX、IF ELSE IEND、 WHILE WEND、PFORK JOINTO PJOINT、 SFORK JOINTO SJOINT 运算与顺序控制命令 : 36 种 =、+、-、*、/、MOD、 、^、&、!、()、S{}、R{}、 SIN、COS、TAN、ASN、ACS、ATN、SQRT、BIN、BCD、 ==、<>、>、<、>=、<=、TON、TOF、SFR、SFL、PON、 NON、BLK、CLR

(3) MP930 的运动控制功能规格

MP930 的运动控制功能规格如下表所示。

表 1.3 MP930 的运动控制功能规格

项目		规格
控制轴数		1 ~ 最多 14 轴
控制规格	PTP 控制	直线、旋转、无限长
	插补	直线最多 14 轴，圆弧 2 轴，螺旋 3 轴
	速度控制	无
	转矩限制	有（仅根据参数设定的转矩限制）
指令单位		mm、inch、deg、pulse
指令最小设定单位		1, 0.1, 0.01, 0.001, 0.0001, 0.00001
最大指令值		-2147483648 ~ +2147483647(带 32 bit 符号)
速度指令单位		mm/min、inch/min、deg/min、pulse/min
加减速型		直线、非对称、S 字 定位 (MOV) 命令中不能使用非对称。
额定速度比功能		定位：以轴为单位 0.00 ~ 327.67% 插补：以组为单位 0.00 ~ 327.67%
坐标系		直角坐标
原点复归		4 种 DOG+C 相、原点 LS、DOG+ 原点 LS、C 相 有原点设定功能
程序	语言	专用运动语言
	任务数	最多可同时并列运行 4 个程序。
	程序数	最多 256 个
	程序容量	相当于 80 kB(字符) (根据梯形程序的使用量有增减, 最大 100 kB)
命令符号		轴移动命令 :8 种 MOV、MVS、MCW、MCC、ZRN、SKP、MVT、EXM 基本控制命令 :6 种 ABS、INC、POS、PLN、MVM 速度与加减速命令 :8 种 ACC、DCC、SCC、VEL、IAC、IDC、IFP、FMX、PLD 高级控制命令 :4 种 PFN、INP、SNG、UFC 控制命令 :10 种 MSEE、TIM、IOW、END、RET、EOX、IF ELSE IEND、WHILE WEND、PFORK JOINTO PJOINT、SFORK JOINTO SJOINT 运算与顺序控制命令 :32 种 =、+、-、*、/、MOD、 、^、&、!、()、S{}、R{}、SIN、 COS、TAN、ASN、ACS、ATN、SQRT、BIN、BCD、==、<>、 >、<、>=、<=、SFR、SFL、BLK、CLR

(4) MP940 的运动控制功能规格

MP940 的运动控制功能规格如下表所示。

表 1.4 MP940 的运动控制功能规格

项目	规格	
控制轴数	1 轴	
控制规格	PTP 控制	直线、旋转、无限长
	插补	直线
	速度指令输出	有
	转矩指令输出	有
	位置控制	定位 外部定位 原点复归 插补 带位置检测功能的插补 恒速进给 恒量进给
	相位控制	有
位置控制	指令单位	mm、inch、deg、pulse
	指令最小设定单位	1, 0.1, 0.01, 0.001, 0.0001, 0.00001
	最大指令值	-2147483648 ~ +2147483647 (带 32 bit 符号)
	速度指令单位	mm/min、inch/min、deg/min、pulse/min
	加减速型	直线、非对称、S 字
	额定速度比功能	0 ~ 327.67% (0.01% 刻度)
坐标系	直角坐标	
原点复归	8 种 ①DEC1+C 相 ⑤DEC1+ZERO ②DEC2+C 相 ⑥DEC2+ZERO ③DEC1+LMT ⑦DEC1+LMT+ZERO ④C 相 ⑧ZERO	
程序	语言	专用运动语言、梯形
	任务数	最多可同时并列运行 8 个程序。
	程序数	最多 32 个
	程序容量	80 kB
速度控制	速度指令	-327.68 ~ +327.67%/ 额定速度 有转矩限制功能
	加减速类型	直线、非对称、S 字 (移动平均)
转矩控制	转矩指令	-327.68 ~ +327.67%/ 额定转矩 有速度限制功能

表 1.5 MP940 的运动控制功能规格

项目		规格
相位控制	速度指令单位	-327.68 ~ +327.67%/ 额定速度
	速度补偿	-327.68 ~ +327.67%/ 额定速度
	位置补偿	-2147483648 ~ +2147483647 pulse
命令符号	轴移动命令 :5 种 MOV、MVS、ZRN、SKP、EXM 基本控制命令 :5 种 ABS、INC、POS、MVM、PLD 速度与加减速命令 :8 种 ACC、DCC、SCC、VEL、IAC、IDC、IFP、FMX 高级控制命令 :4 种 PFN、INP、SNG、UFC 控制命令 :10 种 MSEE、TIM、IOW、END、RET、EOX、IF ELSE IEND、 WHILE WEND、SFORK JOINTO SJOINT 运算与顺序控制命令 :32 种 =、+、-、*、/、MOD、 、^、&、!、()、S{}、R{}、 SIN、COS、TAN、ASN、ACS、ATN、SQRT、BIN、BCD、 ==、<>、>、<、>=、<=、SFR、SFL、BLK、CLR	

表 1.6 MP2100 的运动控制功能规格

项目	规格
命令符号	轴移动命令 12 种 MOV、MVS、MCW、MCC、ZNR、SKP、MVT、EXM、VCS、 VCR、TCS、TCR 基本控制命令 6 种 ABS、INC、POS、PLN、MVM、PLD 速度与加减速命令 8 种 ACC、DCC、SCC、VEL、LAC、IDC、IFP、FMX 高级控制命令 4 种 PFN、INP、SNG、UFC 控制命令 10 种 MSEE、TIM、JOW、END、RET、BOX、IF ELSE IEND、 WHILE WEND、PFORK JOINTO、PJOINT、SFORK、JOINTO、 SJOINT、GOTO 运算与顺序控制命令 36 种 =、+、-、*、/、MOD、 、^、&、!、()、S{}、R{}、 SIN、COS、TAN、ASN、ACS、ATN、SQRT、BIN、BCD、 ==、<>、>、<、>=、<=、SFR、SFL、BLK、CLR 表数据操作命令 TBLBR、TBLBW 字符串操作命令 ASCII、BINASC、ASCBIN

表 1.7 MP2300 的运动控制功能规格

项目	规格
命令符号	轴移动命令 12 种 MOV、MVS、MCW、MCC、ZNR、SKP、MVT、EXM、VCS、 VCR、TCS、TCR
	基本控制命令 6 种 ABS、INC、POS、PLN、MVM、PLD
	速度与加减速命令 8 种 ACC、DCC、SCC、VEL、LAC、IDC、IFP、FMX
	高级控制命令 4 种 PFN、INP、SNG、UFC
	控制命令 10 种 MSEE、TIM、JOW、END、RET、BOX、IF ELSE IEND、 WHILE WEND、PFORK JOINTO、PJOINT、SFORK、JOINTO、 SJOINT、GOTO
	运算与顺序控制命令 36 种 =、+、-、*、/、MOD、 、^、&、!、()、S{}、R{}、 SIN、COS、TAN、ASN、ACS、ATN、SQRT、BIN、BCD、 ==、<>、>、<、>=、<=、SFR、SFL、BLK、CLR
	表数据操作命令 TBLBR、TBLBW
	字符串操作命令 ASCII、BINASC、ASCBIN

1.1.3 运动程序概要

运动程序是用文本形式的运动语言来表述的程序。该运动程序除梯形图以外还能制作 256 个程序。运动程序如下表所示有 2 种。

表 1.7 运动程序种类

分类	指定方法	特点	程序个数
主程序	MPM □□□ 1 ~ 256	可从 DWG.H 调用	主、子程序合计最多可以制作 256 个。
子程序	MPS □□□ 1 ~ 256	可从主程序中调用	

重要

请使用不同的 MPM □□□、MPS □□□ 程序编号。

运动程序指定方法有指定程序编号的直接指定和指定存储程序编号的寄存器编号的间接指定 2 种。

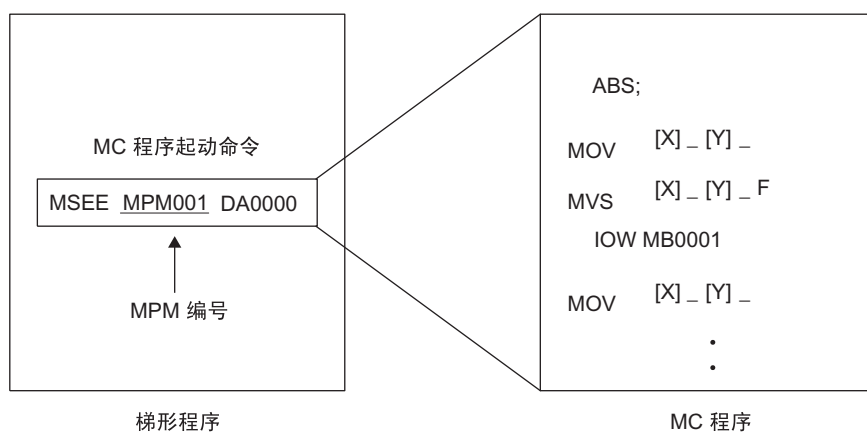


图 1.1 通过直接指定的运动程序起动

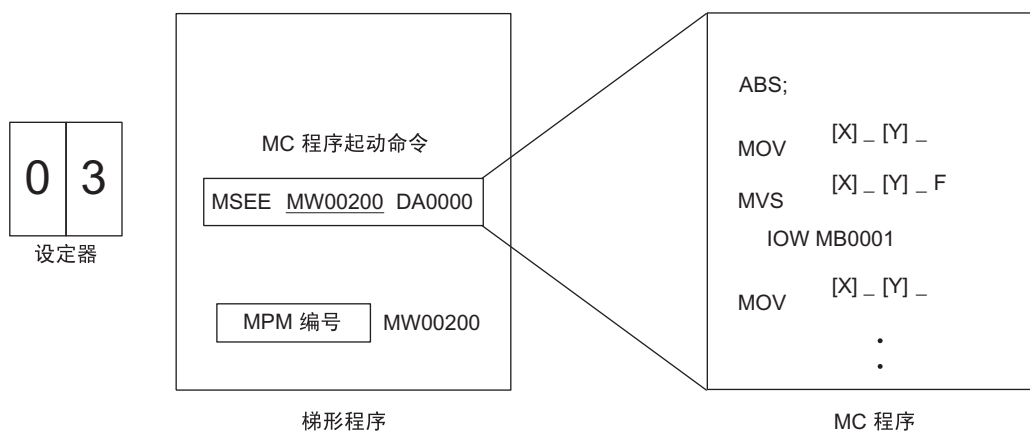
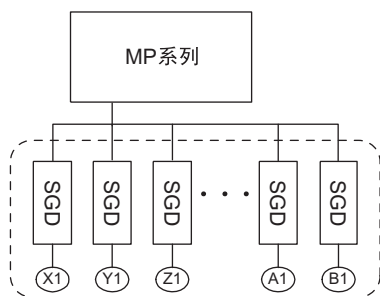


图 1.2 通过间接指定的运动程序起动

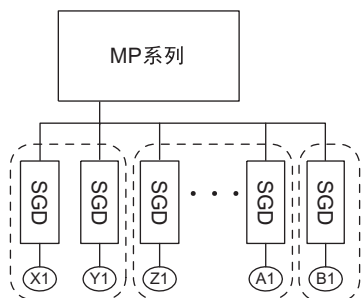
(1) 关于组

MP 系列中，把动作关联的轴群汇总为 1 个组，因为用 1 台机器控制器单独控制多台机械，实现了按组编程。用组定义可进行将轴汇总到组中去的定义。在组运行中，有单组运行和多组运行。组定义的详情请参阅“机器控制器 MP900 /MP2000 系列 编程装置用软件 MPE720 用户手册 (SIJPC88070005)”。

(a) 单组运行

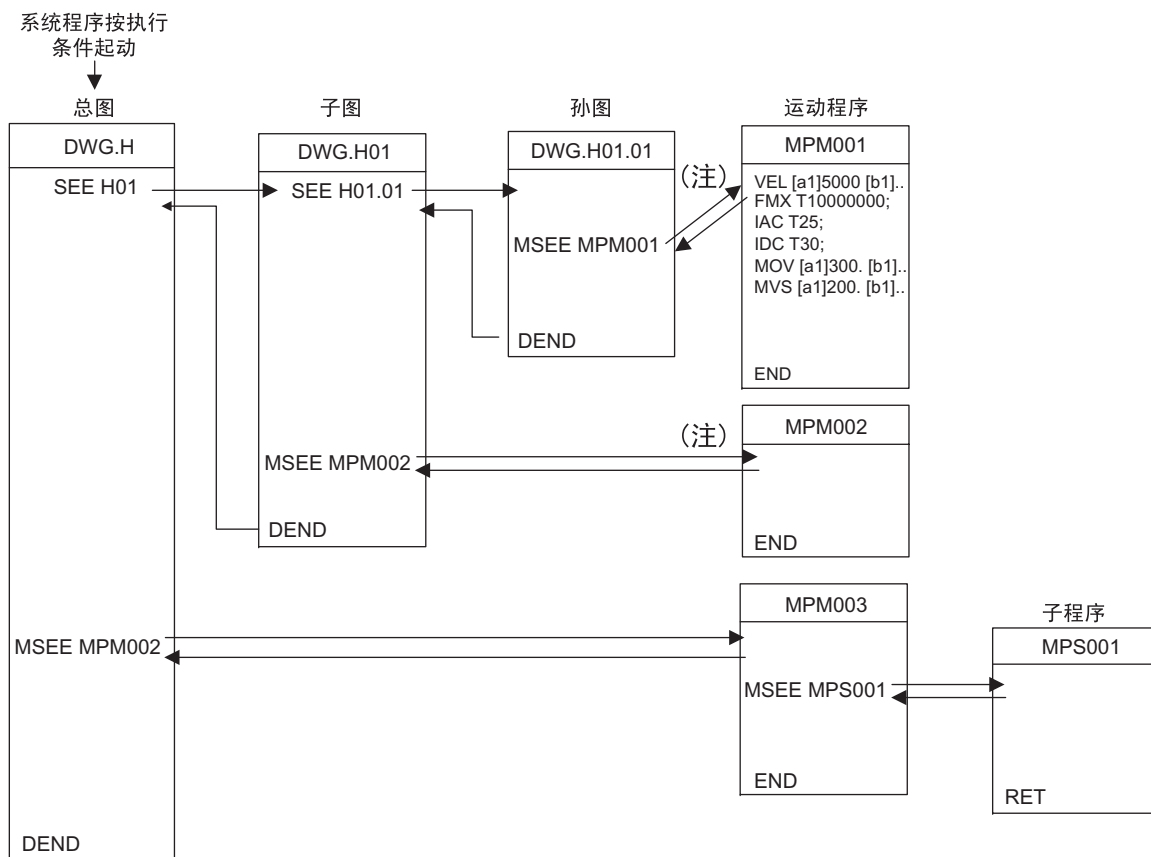


(b) 多组运行



(2) 运动程序执行处理方法

运动命令必须从 H 图通过 MSEE 命令调用。H 图时，则可从总图、子图、孙图任何一个进行调用。



H 图的梯形命令是在每一个高速扫描周期按照总图 - 子图 - 孙图的分层构造的顺序执行。

运动程序虽然是按扫描周期调用，但它不能象梯形程序那样在 1 个扫描周期内执行所有的程序。运动程序是通过系统的运动管理功能进行运动程序专用的执行控制。

重要

- 1 个扫描周期内不结束的命令

以下为 1 个扫描周期内不结束的命令

MOV、MVS、MCW/MCC、ZRN、SKP、MVT、EXM、ACC、DCC、SCC、PFN、VEL、INP、TIM、IOW

- 1 个扫描周期内结束的命令

上述命令以外的命令全部为 1 个扫描周期内结束的命令。

ABS、INC、IFP、PLN、IAC、IDC、FMX、POS、PLD、MSEE、END、RET、IF、WHILE、PFORK、SFORK，所有顺序命令

补充

调用运动程序的时候有以下限制，敬请注意。

- 用 MSEE 命令不能多次调用相同编号的运动程序。
- 不能从梯形的 MSEE 命令调用子程序 (MPSxxx)。仅可从运动程序内部 (MPMxxx、MPSxxx) 调用。
- 不能同时调用同一子程序。

(3) 任务寄存器

为了设定、监视运动程序，需要任务寄存器。运动程序任务寄存器构成如下。

第 1 个字	运动程序的状态标记
第 2 个字	运动程序的控制信号
第 3 个字	插补用额定速度比
第 4 个字	系统任务编号

(注)第 3 个字、第 4 个字仅 MP2000 系列。

(4) 运动程序的控制信号

为了执行用 MSEE 命令从 DWG.H 调用的程序，有必要输入程序控制信号（程序运行开始请求和程序停止请求等）。MSEE 任务寄存器的第 2 个字为运动程序的控制信号。控制运动程序的信号有以下几种。

Bit No.	信号名称	信号种类
0	程序运行开始请求	微分或 NO 输入
1	程序临时停止请求	NO
2	程序停止请求	NO
3	程序单块模式选择	NO
4	程序单块开始请求	微分或 NO 输入
5	警报复位请求	NO
6	程序连续运行开始请求	微分或 NO 输入（仅 MP2000 系列）
8	跳过 1 信息	NO
9	跳过 2 信息	NO
D	系统任务编号设定 *1	NO（仅 MP2000 系列）
E	插补用额定速度比设定 *2	NO（仅 MP2000 系列）

*1. 系统任务编号设定

OFF: 使用系统自动获取的系统任务。系统任务编号每次都不同。

ON: 使用设定的系统任务编号的任务。

*2. 插补用额定速度比设定

OFF: 插补用额定速度比固定为 100%

ON: 按照设定的插补用额定速度比。

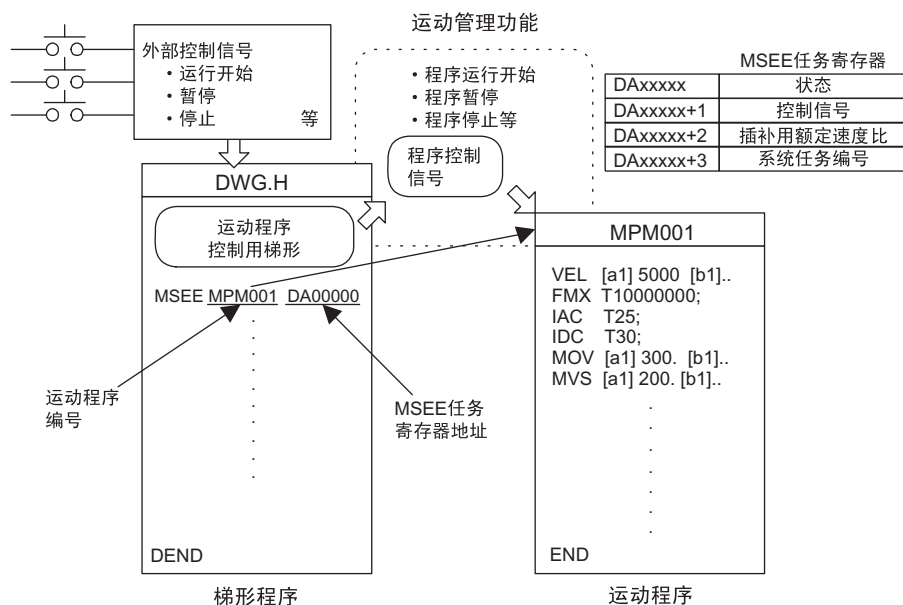
通过用梯形程序向由 MSEE 命令指定的任务寄存器 +1 输入这些信号，可控制运动程序的运行、停止、临时停止等。

梯形程序的输入请设置符合信号种类的信号。

重要

电源接通时，在开始信号为 ON 状态时，程序被起动。

运动程序的执行处理方法如下图所示。



(5) 运动程序的状态标记

MSEE 任务寄存器的第 1 个字为运动程序的状态标记，通过其可以了解运动程序的执行状态。状态标记的详细内容如下表所示。

Bit No.	状态
0	程序运行中
1	程序处于暂停状态
2	响应程序停止请求而停止（在系统中使用）
3	（在系统中使用）
4	程序单块运行停止状态
8	发生程序警报
9	因断点处于停止状态
B	调试模式状态（EWS 调试运行）
D	开始请求信号记录（仅 MP2000 系列）
E	MP900 系列：主程序重复错误 MP2000 系列：无系统任务错误
F	主程序编号超出错误

(6) 插补用额定速度比

MSEE 任务寄存器的第 3 个字用来设定在运动程序内插补类移动指令执行时的额定速度比。

单位：1=0.01%

该插补用额定速度比仅在运动程序控制信号“BitE 插补用额定速度比设定”为 ON 时有效。

(7) 系统任务编号

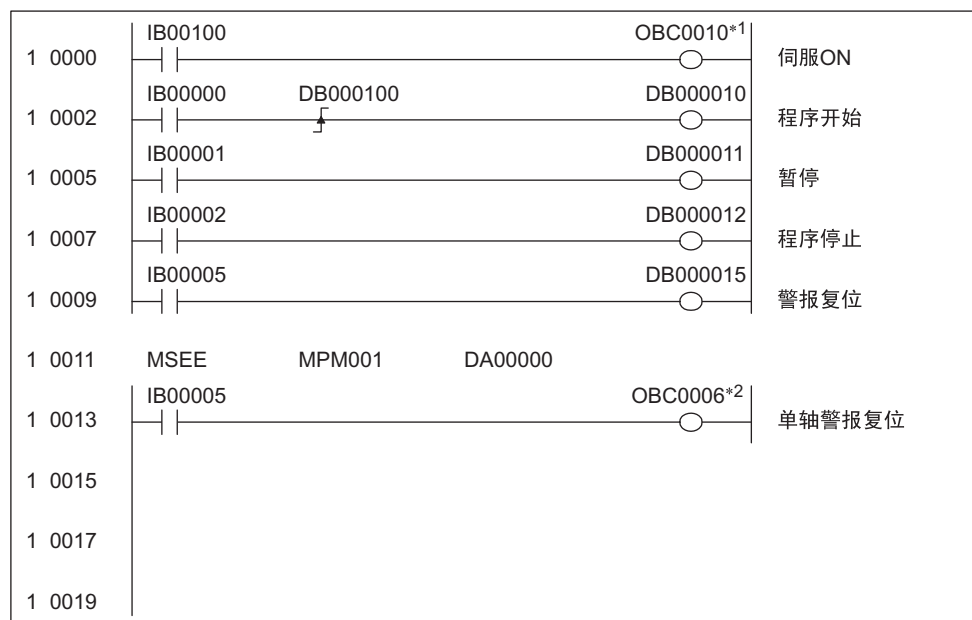
MSEE 任务寄存器的第 4 个字用来设定在运动程序执行时使用的系统任务编号。

- 范围：1 ~ 16

该系统任务编号仅在运动程序控制信号“BitD 系统任务编号设定”为 ON 时有效。在上述范围以外或指定编号的系统任务正在使用中时，状态标记“BitE 无系统任务错误”为 ON。

(8) 运动程序控制用梯形图示例

控制运动程序必需的最低限度的梯形程序如下所示。



* 1. MP2000 系列时为 OB80000。
* 2. MP2000 系列时为 OB8000F。

上述梯形程序的内容如下。

步骤编号	程序内容
1 ~ 7	将 MP 系列的外部输入信号连接的信号存储在运动程序控制信号中 IW0000 (外部输入信号) → DW00001 (MSEE 任务寄存器的第 2 个字) 程序运行开始 程序暂停 程序停止 警报复位
8	运动程序 MPM001 的调用 MSEE <u>MPM001</u> <u>DA00000</u> ① ② ① 运动程序编号 ② MSEE 任务寄存器地址
11 ~ 15	通过警报复位信号 (IB00005) 设定各轴设定参数的动作模式、设置警报复位、清除轴的警报。

当通过上述的梯形程序，MP 系列连接的外部输入输出信号 (IB00000 ~ IB00007) 作为运动程序控制信号向 DW00001 (MSEE 任务寄存器的第 2 个字) 输入时，通过系统的运动管理功能，可以进行运动程序的运行 / 暂停 / 停止等动作。

例 在 MP 系列中，起动运动程序时必要的最低限度的梯形程序外部输入信号举例如下所示。

表 1.8 外部输入信号和运动程序控制信号

外部信号地址	外部信号名称	BIT	运动程序控制信号
IB00000:	程序运行开始	B0:	程序运行开始请求
IB00001:	程序暂停	B1:	程序暂停请求
IB00002:	程序停止	B2:	程序停止请求
IB00003:	程序调试模式	B3:	程序调试模式选择
IB00004:	程序调试开始	B4:	程序调试开始请求
IB00005:	警报复位	B5:	警报复位请求

1.1.4 程序的并列执行

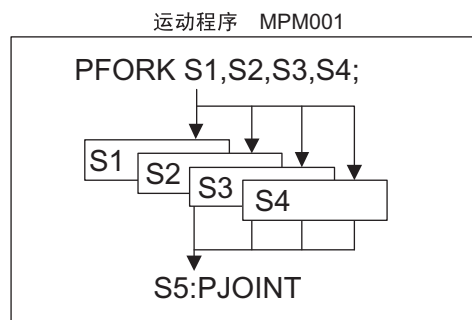
在 MP 系列中，具有并列执行功能，可进行复杂的运动控制，根据各种机器动作可自由地进行编程。运动程序的并列执行有以下 2 种方法。

(1) 根据 PFORK 命令的方法

通过运动程序的 PFORK 命令，在一个程序内最多可并列执行 4 个程序。

并列执行 1

- 在一个运动程序内，可并列执行四个程序。
- 使用 PFORK 命令最多可并列执行四个程序。



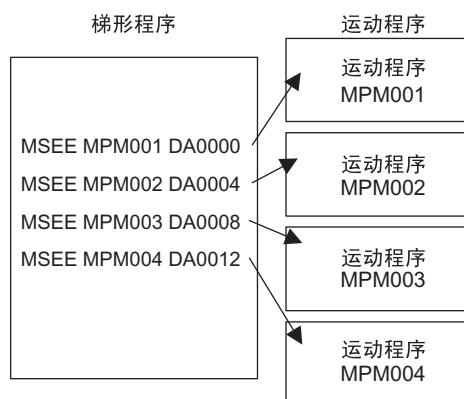
(2) 根据 MSEE 命令的方法

通过梯形程序的 MSEE 命令，可并列执行多个运动程序。

但是，用 MPE720 的“组定义画面”自动生成时，最多可并列执行 4 个运动程序。

并列执行 2

- 可并列执行多个运动程序。
- 通过梯形程序的 MSEE 命令，可并列执行多个运动程序。



1.1.5 程序编辑

运动程序的编辑在 MPE720 (编程装置) 的 Motion Program Editor 画面进行。编辑画面具有以下功能。

- 具有和市售编辑器系统的编辑功能，如剪切、粘贴、检索、替换、跳过等。
- 同时也具备调试运行、程序监视和位置示教等特殊功能。
- 把由市售编辑器制作的文件作为 MP 系列的运动程序的导入功能。
- 把 MP 系列的运动程序文件做成文本文件的导出功能。

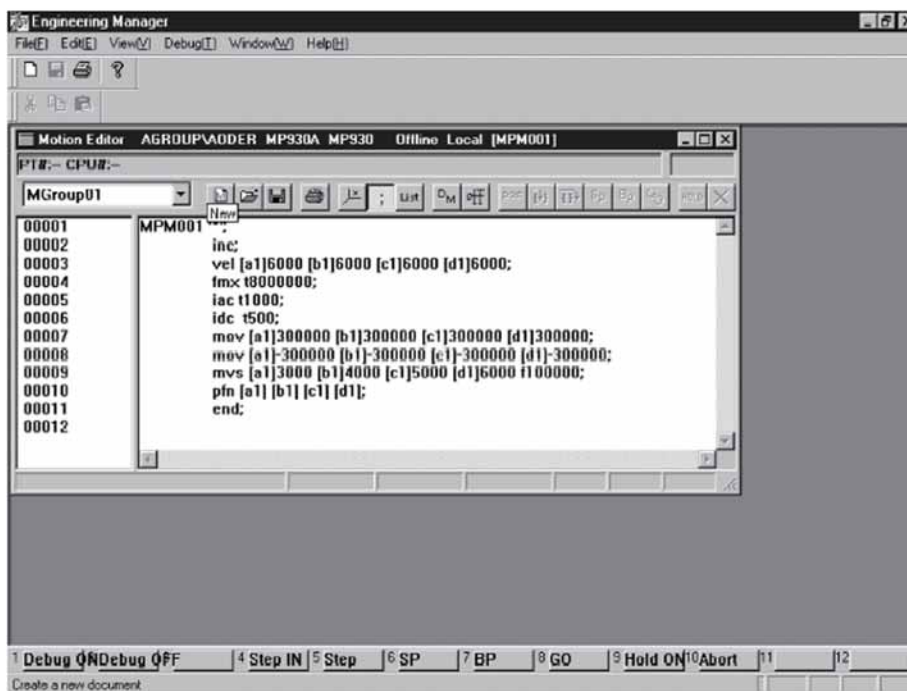


图 1.3 MPE720 的 Motion Program Editor 画面



运动程序编辑功能的详情请参阅“机器控制器 MP900/MP2000 系列编程装置用软件 MPE720 用户手册” (SIJPC88070005)。

1.2 编程方法

本节对制作运动程序的基本规则进行说明。在进行编程之前请务必阅读。

1.2.1 输入格式

(1) 运动程序示例

表 1.9 程序范例

行编号	程序	
00001	MPM001 “sample”	…程序编号和注释
00002	FMX T1000000;	…插补进给最高速度设定
00003	IAC T100;	…插补进给加速时间设定
00004	IDC T100;	…插补进给减速时间设定
00005	VEL [X1]10000 [Y1]2000 [Z1]3000;	…进给速度设定
00006	INC;	…增量值模式指定
00007	MOV [X1]100. [Y1]150. [Z1]200.;	…定位
00008	MVS [X1]100. [Y1]50. F500000;	…直线插补
00009	IOW IW0011=1;	
00010	ML0100=(MW0110*100+50)/100;	
00011	ML0200=(MW0210*100+50)/100;	
00012	ABS;	
00013	MOV [X1]ML0100 [Y1]ML0200;	
00014	POS [X1]0 [Y1]0	
00015	PFORK LA01 LA02 LA03 LA04;	…并列执行命令
00016	LA01: INC;	…标签
00017	MOV [X1]1000.;	
00018	JOINTO LA05;	
00019	LA02: INC;	
00020	MOV [Y1]2000.;	
00021	JOINTO LA05;	
00022	LA03: ABS;	
00023	MVS [Z1]1500 F50000;	
00024	MW1000=12345;	
00025	JOINTO LA05;	
00026	LA04: MW1100=1000;	
00027	IOW IB101==1;	
00028	JOINTO LA05;	
00029	LA05: PJOINT	…并列执行命令结束
00030	END;	…程序结束

(2) 输入格式

可变长度输入格式如下表所示。

表 1.10 可变长度输入格式一览

项 目	输入格式
程序编号	MPM□□□ □□□ = 1 ~ 256
标签	最多 8 个字符
运动命令	3 个字母字符 (也有超过 3 个字符的命令)
坐标符号	[abcd] ± 123.456 A B C A: 轴名称 B: 可指定符号 (+, -) C: 坐标值 详情请参阅 “1.2.2 控制轴”
插补进给速度	F3000000 根据小数点以下位数 (固定参数) 变化 小数点以下位数 = 3 时, 3000.000 mm/分 小数点以下位数 = 5 时, 30.00000 mm/分
待时	TIM T1000 10 msec 单位 (无小数点)
子程序编号	MPS□□□ □□□ = 1 ~ 256
P 指定	P100; 插补进给速度比率设定 1 ~ 100
块结束	;

(3) 读零

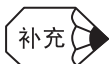
包含程序编号和寄存器 (变量) 编号, 字符后连接的数值可省略读零。

例) [X1]00123 ⇒ [X1]123
[X1]MW00010 ⇒ [X1]MW10
MPS002 ⇒ MPS2

(4) 符号

数值符号中的 +(加) 符号可以省略, -(减) 符号不能省略。

例) [X1]+123 ⇒ [X1]123
[X1]-123 ⇒ [X1]-123



插补进给速度 (Fxxxx) 指令不能使用小数点。
不能指定 F30000.000。请指定 F30000000。
(小数点以下位数 = 3 时)

(5) 可使用字符

可使用 “字符” 和其意义如下表所示。

表 1.11 可使用字符一览

字符	意 义	字符	意 义
C	C 寄存器	P	插补进给速度额定速度比
D	D 寄存器 外部定位移动量	R	圆弧半径
		SS	跳过信号编号
I	I 寄存器	T	定时值、圆弧圈数 FMX、IAC、IDC
M	M 寄存器		
O	O 寄存器	U	圆弧圆心坐标 1 (横坐标)
S	S 寄存器	V	圆弧圆心坐标 2 (纵坐标) S
F	插补进给速度	MPS	子程序编号

(6) 功能字符

可使用的“功能字符”和其意义如下表所示。

表 1.12 功能字符一览

字符	意义	字符	意义
SP	空格（分隔符）	&	运算符
TAB	制表符（分隔符）	!	运算符
;	块的结束	=	运算符
<input type="text" value="Enter"/>	改行	()	运算符
0~9	数字	= =	运算符
A~Z	字母	>	运算符
.	小数点	<	运算符
+	运算符	< >	运算符
-	运算符	> =	运算符
*	运算符	< =	运算符
/	运算符	S{}	运算符
	运算符	R{}	运算符
^	运算符		

(7) 程序编号处理

程序编号用来识别程序。有主程序和子程序 2 种，可分别编 1~256 的号。

主程序

MPM □□□

程序编号(1~256)

子程序

MPS □□□

程序编号(1~256)

图 1.4 运动程序文件名

补充

- 在主程序和子程序中，不可指定同一程序编号。
- 主程序和子程序合计最多可制作 256 个。

(8) 注释表述

为了对运动程序进行维护，可在程序内部表述程序注释。表述方法有以下 2 种。

- 用双引号括起注释语句。

“字符串”



ZRN [axis1]0 [axis2]0 [axis3]0; “所有轴的原点复归”

MVS [axis1]100.0 [axis2]200. [axis3]300.0; “3 轴直线插补”

- 不用双引号括起时，双引号开始的 1 行（到 为止）字符全部为注释。

“字符串 ”



所有轴的原点复归后，用直线插补移动到待机位置

ZRN [axis1]0 [axis2]0 [axis3]0;

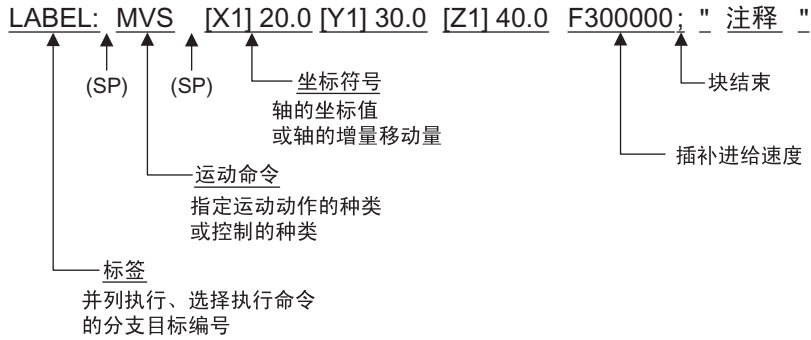
MVS [axis1]100.0 [axis2]200. [axis3]300.0;

补充

- 在注释字符串中也可使用汉字。

(9) 单块指令的制作方法

1. 按照输入格式一览，制作单块指令。单块典型例如下所示。



2. 在“运动命令”和“坐标符号”之间，请务必放入一个空格“SP”符号。
3. 块的行字符数没有限制，为了方便地阅读程序，建议用画面上可显示范围内的字符数分隔开。
4. 块结束时，请务必在“;”字符的后面加入 CR、LF 以后再改行。
5. 标签为并列执行命令 (PFORK) 及选择执行命令 (SFORK) 的分支目标编号。

(10) 标签

作为并列执行命令 (PFORK) 及选择执行命令 (SFORK) 的分支目标编号，必须添加标签。请在由英文字母、数字、符号构成的 1 ~ 8 个的字符串的最后添加“:”。

标签可使用的字符如下。

字符	数字	0 ~ 9
	英文字母	A ~ Z, a ~ z
	符号	\$_, %, ¥, @, -, __, .



```
PFORK LAB1 LAB2;
LAB1: ZRN [axis1]0 [axis2]0 [axis3]0;
      JOINTO LAB3;
LAB2: MVS [axis1]100.0 [axis2]200. [axis3]300.0;
      JOINTO LAB3;
LAB3: PJOINT;
```



- 在程序内有多个相同的标签时，将会出错（“标签被重复定义”）。
- PFORK 的分支数和标签数不同时，将会出错。
- PFORK 命令的标签间的分隔可使用空格键或逗号 (,)。空格键和逗号 (,) 可混合使用。

(例)PFORK LABEL1 LABEL2, LABEL3 LABEL4

1.2.2 控制轴

(1) 轴名称

在 MP 系列中最多可用 8 个字符任意设定轴名称。用 MPE720 的“组定义画面”进行轴名称的设定。在轴名称中可使用的字符及默认轴名称如下表所示。

可使用字符	0 ~ 9, A ~ Z, a ~ z
轴名称举例	[axis1] [X1] [CONV1]
默认轴名称	4 轴指定时 [A1] [B1] [C1] [D1] 8 轴指定时 [A1] [B1] [C1] [D1] [E1] [F1] [G1] [H1]



- 轴名称请务必使用 1 ~ 8 字符以内的英文和数字并用 [] 括起来。
- 不能对多个轴设定相同的轴名称。
- 在程序中指定和“组定义画面”中设定的轴名称不同的轴名称时，将会出错（“轴名称错误”）。

(2) 坐标符号一览

本手册中将轴名称和轴名称后面带符号的移动量或坐标值称为“坐标用语”。

本系统中使用的坐标用语意义如下表所示。

坐标用语的指定方法		意义
轴名称	[X1], [Y1], [axis]	想移动的轴名称的指定
移动量或坐标值	直接指定： 123.456 变量指定： ML0100	指定轴的坐标值或增量移动距离
圆弧插补、 螺旋插补的辅助数据	R	圆弧插补的半径值（固定为增量值）
	U	圆弧插补的圆心坐标（横轴）
	V	圆弧插补的圆心坐标（纵轴）
外部定位移动量	D	外部信号输入以后移动的距离（固定为增量值）

重要

- 用变量指定移动量或坐标值时，使用的数据类型为 LONG。
（例）ML0100, DL0300
- 有小数点时，小数点以下的位数指定为 0。
（例）300.000...小数点以下位数=3 时，将 300000 存储在变量中。
但在 I、O 寄存器中，伺服参数域不能作为移动量、坐标值的变量使用。

(3) 同步控制轴数

运动程序指令的可同时控制的轴数如下表所示。

表 1.13 同步控制轴数一览

命令符号	同步控制轴数
定位 (MOV)	最多 14 轴 *
直线插补 (MVS)	最多 14 轴 *
圆弧插补 (MCW/MCC)	2 轴
螺旋插补 (MCW/MCC)	3 轴
跳过命令 (SKP)	最多 14 轴 *
外部定位 (EXM)	1 轴
时间指定定位 (MVT)	最多 14 轴 *

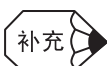
* 为 MP930 的示例。机型不同最多轴数也不同，详情请参阅“1.1.2 功能一览”。

(4) 指令单位

MP 系列中可编程的指令单位可用参数向轴单位设定。

表 1.14 指令单位一览

参数设定值	指令单位 pulse	指令单位 mm	指令单位 deg	指令单位 inch
小数点以下位数=0	1 pulse	1 mm	1°	1"
小数点以下位数=1	1 pulse	0.1 mm	0.1°	0.1"
小数点以下位数=2	1 pulse	0.01 mm	0.01°	0.01"
小数点以下位数=3	1 pulse	0.001 mm	0.001°	0.001"
小数点以下位数=4	1 pulse	0.0001 mm	0.0001°	0.0001"
小数点以下位数=5	1 pulse	0.00001 mm	0.00001°	0.00001"



指令单位 = pulse 时，小数点以下位数无效。运动程序的输入及位置监视显示的小数点也无意义。

(5) 最大指令值

1 次移动指令的最大值如下表所示。

	小数点 以下位数	指令单位 pulse	指令单位 mm	指令单位 deg	指令单位 inch
有 限 长	0	-2147483648 ~ 2147483647	-2147483648 ~ 2147483647	0 ~ 35999999	-2147483648 ~ 2147483647
	1	-2147483648 ~ 2147483647	-214748364.8 ~ 214748364.7	0 ~ 3599999.9	-214748364.8 ~ 214748364.7
	2	-2147483648 ~ 2147483647	-21474836.48 ~ 21474836.47	0 ~ 359999.99	-21474836.48 ~ 21474836.47
	3	-2147483648 ~ 2147483647	-2147483.648 ~ 2147483.647	0 ~ 35999.999	-2147483.648 ~ 2147483.647
	4	-2147483648 ~ 2147483647	-214748.3648 ~ 214748.3647	0 ~ 3599.9999	-214748.3648 ~ 214748.3647
	5	-2147483648 ~ 2147483647	-21474.83648 ~ 21474.83647	0 ~ 359.99999	-21474.83648 ~ 21474.83647

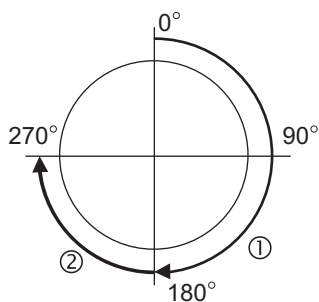
(注) 指令单位 = pulse 时

(6) “无限长轴”的“绝对值模式”指定方法

在“无限长轴”的绝对值指令（在 $0 \sim 359.999^\circ$ 范围内指定模式）中，如下图所示，指令的符号表示旋转方向，指令的角度值表示绝对位置。

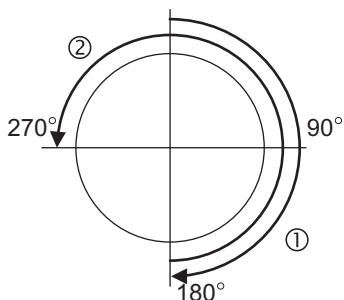
例 当前值为 180° 位置指定时的举例如下所示。

(a) 无限长轴绝对值模式例 1



```
ZRN [X1] 0;
① INC MOV [X1] 180.0;
② ABS MOV [X1] 270.0; 顺时针方向（正转方向）向  $270^\circ$  移动。
```

(b) 无限长轴绝对值模式例 2



```
ZRN [X1] 0;
① INC MOV [X1] 180.0;
② ABS MOV [X1] -270.0; 逆时针方向（反转方向）向  $270^\circ$  移动。
```

补充

在“无限长轴”的“绝对值模式”的指定下，向 0° 位置移动时，即使指定为 $+0.0$ ，也按逆时针方向移动。想顺时针方向移动，请指定 $+360.0$ 。

(7) 数值和变量的表达方法

运动程序中可使用的数值分为常量和变量。以下对数值的设定方法进行表述。

(a) 常量

表 1.15 可指定的数值表达

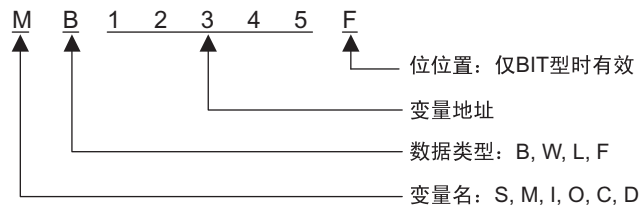
分类	范围	表述举例
10 进制整数	-2147483648 ~ 2147483647	0, 734, +823, -2493
10 进制小数	-214748.364 ~ 2147483.647 根据小数点以下位数变化	763., +824.2, -234.56 -321.12345
16 进制整数	0 ~ FFFFFFFFH	FFFABCDEH, 2345H, FH
实数	± (1.175E-38 ~ 3.402E +38), 0	123.45, -765.4321

(b) 变量

运动程序中可使用的变量有以下几种，请根据用途的不同区别使用。

表 1.16 变量的种类和表达方法

分类	变量类型	数据类型			
		BIT	WORD	LONG	FLOAT
全局变量	S 寄存器	SB	SW	SL	SF
	M 寄存器	MB	MW	ML	MF
	I 寄存器	IB	IW	IL	IF
	O 寄存器	OB	OW	OL	OF
	C 寄存器	CB	CW	CL	CF
局部变量	D 寄存器	DB	DW	DL	DF



```

MB001001=1;
MW00100=1234;
ML00100=12345678;
MF00100=1234.5678;

```

(8) 运算表达式及函数

用运算符和函数结合全局变量、局部变量、常量，进行一般的算术运算，可把结果赋给变量。运算表达式及函数准备有以下命令。

分类	命令	名称	指令格式
数值运算	=	赋值	MW - = MW - ;
	+	加法运算	MW - = MW - +MW - ;
	-	减法运算	MW - = MW - - MW - ;
	*	乘法运算	MW - = MW - * MW - ;
	/	除法运算	MW - = MW - / MW - ;
	MOD	余数运算	MW - = MOD;
逻辑运算		OR(逻辑或)	MB - = MB - MB - ;
	^	XOR(逻辑异或)	MB - = MB - ^ MB - ;
	&	AND(逻辑与)	MB - = MB - &MB - ;
	!	NOT(取反)	MB - = MB - ! MB - ;
数值比较	= =	等于	IF MW - = = MW - ;
	< >	不等于	IF MW - < > MW - ;
	>	大于	IF MW - > MW - ;
	<	小于	IF MW - < MW - ;
	> =	大于等于	IF MW - > = MW - ;
	< =	小于等于	IF MW - < = MW - ;
数据操作	SFR	右移	SFR MB - N - W - ;
	SFL	左移	SFL MB - N - W - ;
	BLK	块传送	BLK MW - MW - W - ;
	CLR	清除	CLR MB - W - ;
基本函数	SIN	正弦	SIN(MW -);
	COS	余弦	COS(MW -);
	TAN	正切	TAN(MF -);
	ASN	反正弦	ASN(MF -);
	ACS	反余弦	ACS(MF -);
	ATN	反正切	ATN(MW -);
	SQT	平方根	SQT(MW -);
	BIN	BCD → BIN	BIN(MW -);
	BCD	BIN → BCD	BCD(MW -);
	S{}	指定位 ON	S{MB - } = MB - &MB - ;
	R{}	指定位 OFF	R{MB - } = MB - &MB - ;

1.2.3 进给速度

(1) 快进给速度

快进给速度在下列轴移动时使用。

- “定位 (MOV)” 命令
- “点动运行 (JOG)” 操作
- “步骤运行 (STEP)” 操作

请用各轴的参数“快进给速度 (OLxx22 / OLxx10)”或运动程序“进给速度变更指令 (VEL)”设定快进给速度的设置值。

- 快进给速度参数

系列	参数编号	名称	设定范围	单位
MP900	OLxx22	快进给速度	0 ~ 2 ³¹ - 1	根据指令单位
MP2000	OLxx10	速度指令设定	0 ~ 2 ³¹ - 1	根据指令单位

- 运动程序中设定进给速度的方法

<ul style="list-style-type: none"> • 向快进给速度参数直接设定的方法 OLC022 = 6000 ; 第 1 个轴快进给速度设定 (OL8010) OLC062 = 5000 ; 第 2 个轴快进给速度设定 (OL8010) OLC0A2 = 7000 ; 第 3 个轴快进给速度设定 (OL8010) • 用进给速度变更命令 (VEL) 设定的方法 VEL [X1] 6000 [Y2] 5000 [Z1] 7000 ;

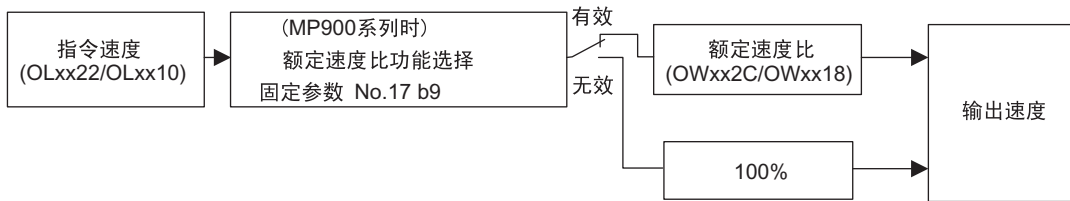
可在 0 ~ 327.67% 的范围内，切换快进给的额定速度比。用各轴的设定参数“额定速度比”设定

- 额定速度比参数

系列	参数编号	设定范围	单位
MP900	OWxx2C	0 ~ 32767	1=0.01%
MP2000	OWxx18	0 ~ 32767	1=0.01%

额定速度比¹的设定方法有运动程序、梯形程序和参数设定画面 3 种。

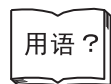
$$\text{指令速度 (OLxx22/OLxx10)} \times \text{额定速度比 (OWxx2C/OWxx18)} = \text{输出速度}$$



(注)MP2000 系列时，额定速度比一直有效。



- 额定速度比在运行中一直有效。在轴移动中可用梯形程序、运动程序、参数设定来变更。
- 根据额定速度比的设定数据的输出速度在范围以外时，为“参数设定错误”。



¹ 额定速度比
 英语的“OVERRIDE”表示“设为无效”，在此解释为变更设定值。

(2) 插补进给速度

插补进给命令的进给速度，用字符“F”后的数字指令。有时可称为“F指令”。
直线插补、圆弧插补的F指令变为指令了切线方向速度的指令。



INC MVS [X] 1200 [Y] 900 F500 ; 时
 $F = 500 = \sqrt{400^2 + 300^2}$ [指令单位/min]

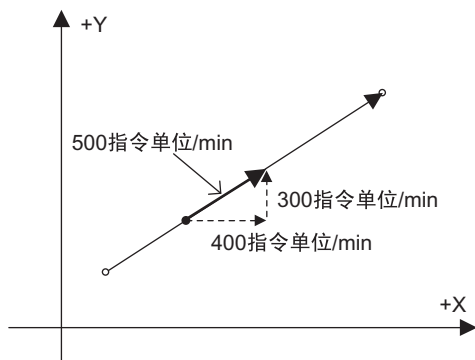


图 1.5 2轴直线插补的切线速度

MCC [X]--- [Y]--- U--- V--- F200 ; 时
 $F = 200 = \sqrt{V_x^2 + V_y^2}$ [指令单位/min]

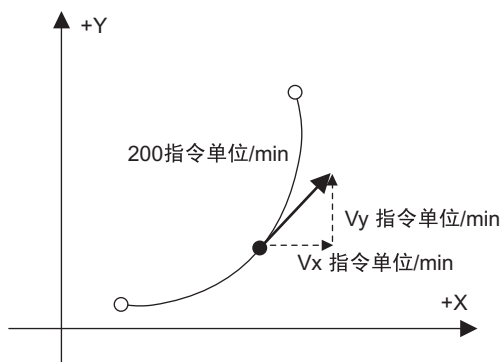


图 1.6 圆弧插补的切线速度



INC MVS [X] 100 [Y] 100 [Z] 100 F500 ; 时
 $F = 400 = \sqrt{V_x^2 + V_y^2 + V_z^2}$ [指令单位/min]

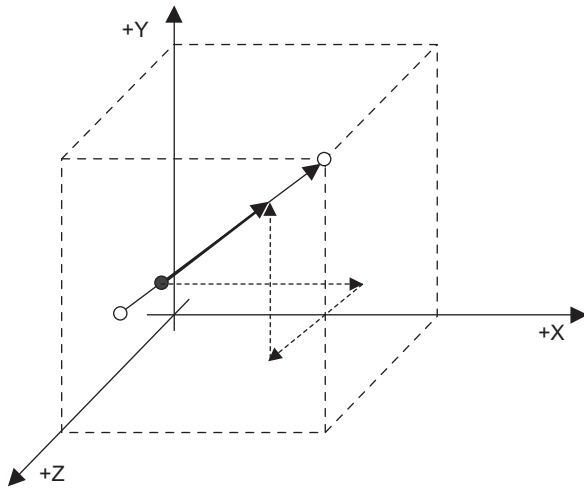
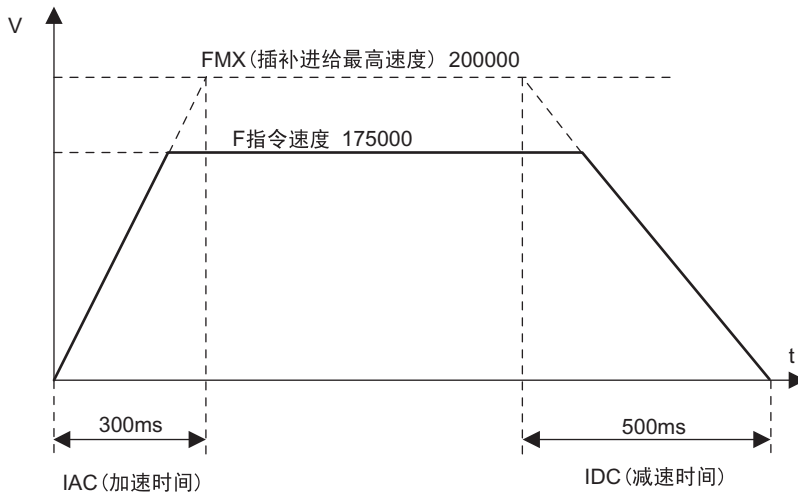


图 1.7 2 轴直线插补的切线速度

INC MVS [X] -- [Y] -- [Z] -- [S] -- F600; 时
 $F = 600 = \sqrt{V_x^2 + V_y^2 + V_s^2}$ [指令单位/min]

图 1.8 (4 轴直线插补的切线速度)

根据机械类和伺服类性能的不同，进给速度的上限有所限制。用下述运动命令来设定该进给速度的上限值。



程序举例

```
FMX T200000;
IAC T300;
IDC T500;
MVS [X]200,[Y]250,F175000;
```

图 1.9 插补进给速度限制的设定命令

当指令超过该插补进给最高速度的 F 指令时，则成为警报。

重要

运动程序中使用插补指令时，请务必在程序的开头设定 FMX 命令。

F 指令值的数值不能使用小数点。



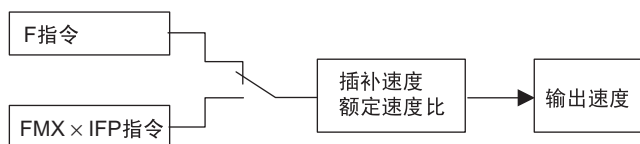
F2000000

在 0 ~ 327.67% 的范围内，可切换插补进给速度的额定速度比。可向以下寄存器进行额定速度比的设定。

- MP900 系列时
在“组定义画面”设定的寄存器（默认 = MW00001）
- MP2000 系列时
MSEE 命令的任务寄存器的第 3 个字的寄存器
但是，根据 MSEE 命令的任务寄存器的第 2 个字的寄存器的 Bit 14 “额定速度比有效选择”不同，插补额定速度比也不同。

额定速度比有效选择 OFF: 固定为 100%
额定速度比有效选择 ON: 按照设定的额定速度比。

$$\left(\begin{array}{l} \text{F指令} \\ \text{FMX} \times \text{IFP指令} \end{array} \right) \times \left(\begin{array}{l} \text{插补速度额定速度比} \\ \text{(100\% = 10000)} \end{array} \right) = \text{输出速度}$$



插补进给速度的相关运动命令如下所示。

- F 指令： “插补命令内的 F 指令”
- IFP 命令： “插补进给速度比率设定”
- FMX 命令： “插补进给最高速度”
- IAC 命令： “插补加速时间变更”
- IDC 命令： “插补减速时间变更”
- SCC 命令： “S 字时间参数变更”



- 额定速度比在运行中一直有效。
- 根据额定速度比设定数据的变化，输出速度在范围外时，用 FMX 值固定速度。



- 在插补指令的轴中含有旋转轴时，机械的移动速度不为 F 指令规定的切线速度。
- 当用 F 指令指令“F○”时，则为程序错误。
- 请不要指令使用负号的 F 指令“F-□□□”。指令时，成为警报。

1.2.4 运动命令一览

运动命令一览在书后的“附录 A 运动命令一览”中有介绍，敬请参阅。

第 2 章

运动命令

本章对运动命令中的轴移动命令及控制命令的编程方法进行了说明。

2.1 轴移动命令	2-2
2.1.1 定位 (MOV)	2-2
2.1.2 直线插补 (MVS)	2-5
2.1.3 圆弧插补 (MCW、MCC)	2-8
2.1.4 螺旋插补 (MCW、MCC)	2-13
2.1.5 原点复归 (ZRN)	2-15
2.1.6 跳过功能 (SKP)	2-19
2.1.7 时间指定定位 (MVT)	2-20
2.1.8 外部定位 (EXM)	2-22
2.2 控制命令	2-23
2.2.1 绝对值 (ABS) 模式	2-23
2.2.2 增量值 (INC) 模式	2-25
2.2.3 当前值变更 (POS)	2-26
2.2.4 坐标平面指定 (PLN)	2-28
2.2.5 机械坐标指令 (MVM)	2-29
2.2.6 程序当前位置更新 (PLD)	2-30
2.2.7 待时 (TIM)	2-31
2.2.8 程序的结束 (END)	2-32

2.1 轴移动命令

本节对轴移动命令的指令方法、程序举例进行归纳阐述。

2.1.1 定位 (MOV)

⚠ 注意

- 根据“定位 (MOV)”命令的移动轨迹不会成为直线插补的“直线”。编程时，为了不使工具和工件等发生干涉，必须进行轨迹检查。
如果疏于检查，可能发生因干涉引起的工具损坏以及相应的人身事故。

(1) 概要

定位 (MOV) 是指使各轴相互独立，以快进给速度（设定于各轴参数中的速度）从当前位置到终点位置移动轴的命令。最多可同时移动 14 轴*。不移动省略指令的轴。

根据 MOV 命令的移动轨迹，不会成为根据“直线插补”指令的直线移动。

* 为 MP930 的示例。机型不同最多轴数也不同，详情请参阅“1.1.2 功能一览”。

(2) 详细说明

MOV 命令的指令方法如下所示。

```
MOV [axis1] - [axis2] - ..... ;
      指令位置
```

下图表示根据 MOV 命令的移动轨迹。

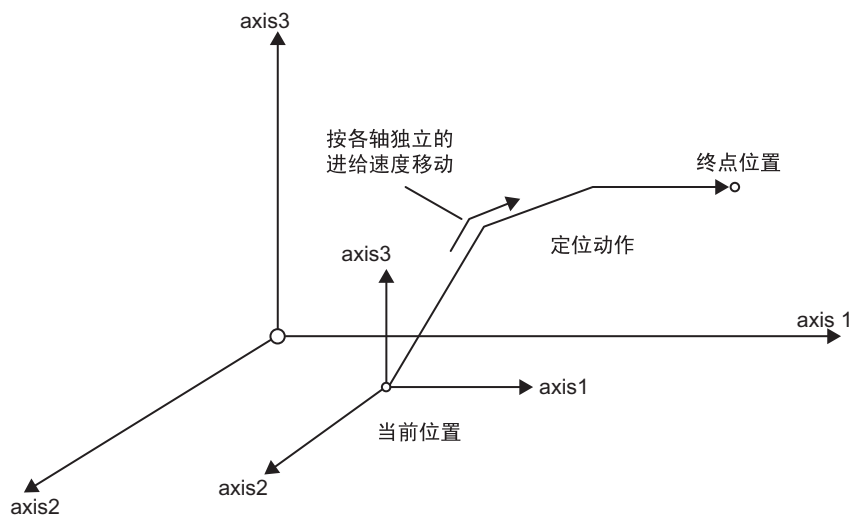


图 2.1 根据 MOV 命令的移动轨迹

指令位置根据事先指定的 ABS / INC¹ 模式，进行如下设定。

绝对值 (ABS) 模式：是在工件坐标轴上把进行轴移动的命令的坐标语句作为绝对值处理的模式

增量值 (INC) 模式：是在工件坐标轴上把进行轴移动的命令的坐标语句作为从当前位置开始的增量值处理的模式

对根据 MOV 命令的轴移动，进行入位检查，即检查其是否进入定位结束范围内。入位检查后，执行下一移动指令块。入位检查动作如下图所示。

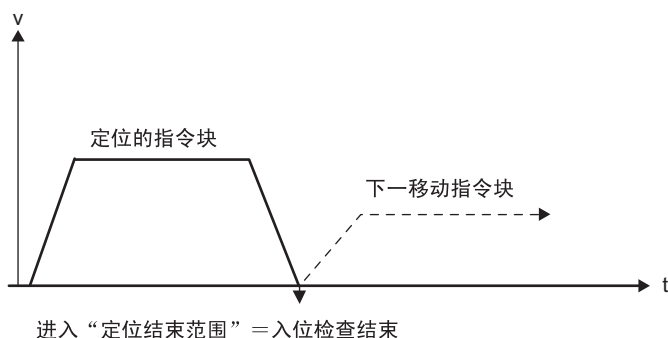
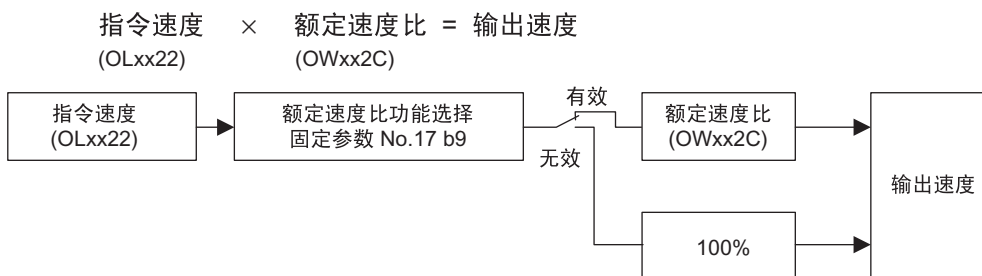


图 2.2 入位检查动作

快进给速度是设定在各轴设定参数 30 “快进给速度 (OLxx22)” * 中的速度。对于快进给，在 0 ~ 327.67% 的范围内，可切换额定速度比。额定速度比用各轴设定参数 35 “额定速度比 (OWxx2C)” 设定。



* 为 MP900 系列的示例。在 MP2000 系列中，快进给速度的参数是不同的，详情请参阅“1.2.3 进给速度”的快进给速度。

在根据定位 (MOV) 命令的移动的自动加减速控制中，通过参数设定可选择 1 段直线加减速或 S 字加减速。

用语？

¹ “ABS / INC” 命令

是指令用绝对值 (ABS 值) 还是增量值 (INC 值) 来处理坐标符号的命令。是一旦指定，直到下次执行其他替换指令之前一直有效的“模态组命令”。

定位的自动加减速的相关参数及运动命令如下所示。



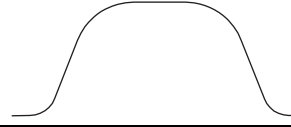
参数

	MP900	MP2000
直线加减速时间设定	0Wxx0C / 0Wxx0D	0Wxx36 / 0Wxx38
滤波时间参数	0Wxx14	0Wxx3A
运动命令控制标志滤波类型选择	0Wxx21 b4 ~ b7	0Wxx03 b8 ~ b11

- 运动命令 “加速时间变更 (ACC)”
- 运动命令 “减速时间变更 (DCC)”
- 运动命令 “S 字时间参数变更 (SCC)”

通过对上述参数及运动命令进行组合，可自由设定加减速模式。

下表为 MP900 系列的示例。

NO.	自动加减速类型	相关参数的设定	备注
1	无加减速	0Wxx0C = 0 0Wxx21b4 ~ b7 = 0	
2	1 段直线	0Wxx0C ≠ 0 0Wxx21b4 ~ b7 = 0	
3	S 字	0Wxx0C ≠ 0 0Wxx21b4 ~ b7 = 2 0Wxx14 ≠ 0	



MOV 命令中不能指定非对称加减速。

(3) 程序举例

ABS 模式下的 MOV 命令的程序举例如下所示。

```
ABS;
MOV [axis1]4000 [axis2]3000 [axis3]2000;
但
当前位置 axis1 = axis2 = axis3 = 0
```

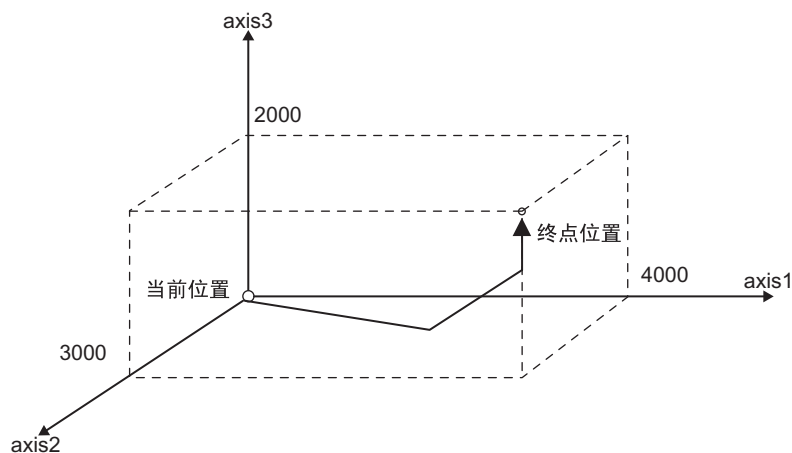


图 2.3 MOV 命令的程序举例

2.1.2 直线插补 (MVS)

⚠ 注意

- 进行“直线插补 (MVS)”的轴可以是直线轴也可以是旋转轴。但是，有旋转轴时，直线插补的轨迹将不是“直线”。编程时，为了不使工具和工件等发生干涉，必须进行轨迹检查。
如果疏于检查，可能发生因干涉引起的工具损坏以及相应的人身事故。

(1) 概要

直线插补 (MVS) 是指以插补进给速度使各轴从当前位置到终点位置进行直线移动的命令。最多可同时移动 14 轴*。不移动省略指令的轴。

* 为 MP930 的示例。机型不同最多轴数也不同，详情请参阅“1.1.2 功能一览”。

(2) 详细说明

MVS 命令的指令方法如下所示。

MVS	[axis1]	—	[axis2]	—	...	F	—;
			指令位置			插补进给速度	

移动轨迹如下图所示。

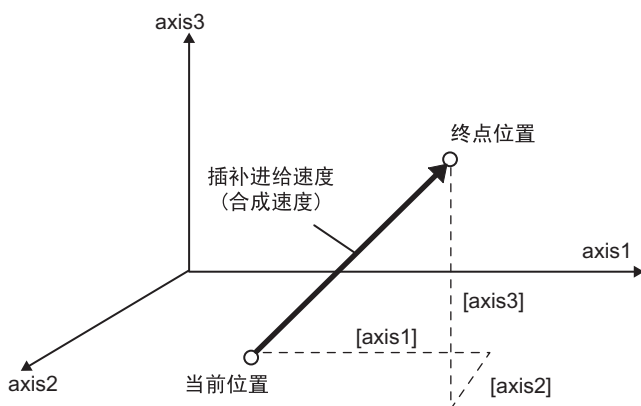


图 2.4 根据 MVS 命令的移动轨迹

指令位置根据事先指定的 ABS / INC 模式进行设定。

插补进给速度也称为 F 指令。用速度指定 (F) 或插补进给速度比率设定 (IFP) 进行指令。F 指令或 IFP 命令中在以前的块中指定过的最终指令有效。电源接通后，没有进行任何 F 指令或 IFP 命令指定的插补命令成为警报。

被指令的所有轴速度的合成速度和 F 指令值相等。但是，当向插补进给最高速度 (FMX) 指定超过设定界限值的 F 指令时，则成为警报。

- 指令 axis1、axis2 的 2 轴时

$$F = \sqrt{V_{\text{axis1}}^2 + V_{\text{axis2}}^2}$$

- 指令 axis1、axis2、axis3 的 3 轴时

$$F = \sqrt{V_{\text{axis1}}^2 + V_{\text{axis2}}^2 + V_{\text{axis3}}^2}$$

- 指令 axis1、axis2、axis3、axis4 的 4 轴时

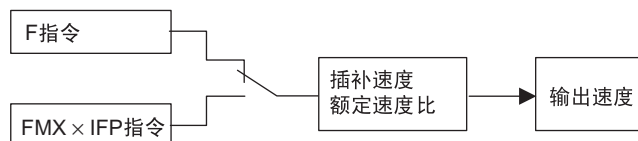
$$F = \sqrt{V_{\text{axis1}}^2 + V_{\text{axis2}}^2 + V_{\text{axis3}}^2 + V_{\text{axis4}}^2}$$

重要

在制作使用插补指令的运动程序时，请在程序的开头指令插补进给最高速度 (FMX)。不指定时，成为警报。

在实际程序运行中，可对 F 指令值乘上 0 ~ 327.67% 的额定速度比。额定速度比立即有效。关于额定速度比的设定，请参阅“1.2.3 进给速度”的插补进给速度。

$$\left(\begin{array}{l} \text{F 指令} \\ \text{FMX} \times \text{IFP 指令} \end{array} \right) \times \left(\begin{array}{l} \text{插补速度额定速度比} \\ (100\% = 10000) \end{array} \right) = \text{输出速度}$$



不对根据直线插补 (MVS) 命令的轴移动执行入位检查。指令块的脉冲分配一结束，立刻执行下面的块。如果在入位检查后执行下面的块，请在该块或下面的块中，指令入位检查 (PFN) 命令。

在根据插补命令的移动自动加减速控制中，可通过参数设定及 IAC、IDC 命令的设定选择以下的加减速。

- 1 段直线加减速
- 非对称加减速
- S 字加减速

插补进给自动加减速的相关参数及运动命令如下所示。

参数

	MP900	MP2000
滤波时间参数	0Wxx14	0Wxx3A
运动命令控制标志滤波类型选择	0Wxx21 b4 ~ b7	0Wxx03 b8 ~ b11

- 运动命令“插补进给最高速度设定 (FMX)”
- 插补命令内的 F 指令“插补进给速度”
- 运动命令“插补进给速度比率设定 (IFP)”
- 运动命令“插补加速时间变更 (IAC)”
- 运动命令“插补减速时间变更 (IDC)”
- 运动命令“S 字时间参数变更 (SCC)”

通过对上述参数及运动命令进行组合，可自由设定加减速模式。下表为 MP900 系列的示例。

NO.	自动加减速类型	相关参数及命令	备注
1	无加减速	插补加速时间变更 (IAC) = 0 插补减速时间变更 (IDC) = 0 0Wxx21 b4 ~ b7 = 0	
2	直线插补	插补加速时间变更 (IAC) ≠ 0 插补减速时间变更 (IDC) ≠ 0 0Wxx21 b4 ~ b7 = 0	
3	S 字	插补加速时间变更 (IAC) ≠ 0 插补减速时间变更 (IDC) ≠ 0 0Wxx21 b4 ~ b7 = 2 0Wxx14 ≠ 0	

根据插补命令的运动的自动加减速控制的加速及减速时间可用 IAC 或 IDC 命令进行设定。

(3) 程序举例

ABS 模式下 MVS 命令的程序举例如下所示。



```

FMX T30000000;
MW00001=10000; (*)
ABS;
MVS [axis1]4000 [axis2]3000 [axis3]2000 F1000;
但,
当前位置 :axis1 = axis2 = axis3 = 0

```

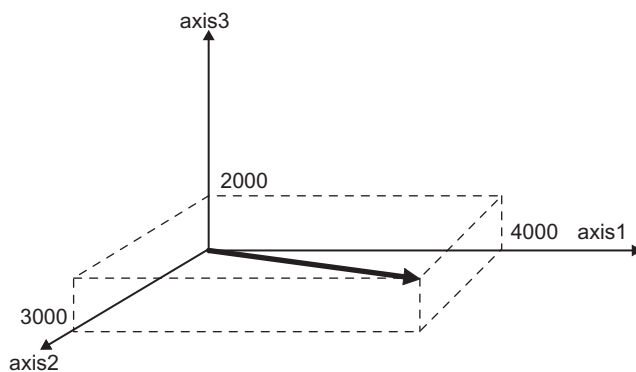


图 2.5 MVS 命令的程序举例

* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。



补充

- 仅当与插补命令相同块时可指令速度指定 (F) 指令。
- 不能以单独指令向与插补命令相同的块指令插补进给速度比率设定 (IFP)。IFP 命令的详情请参阅“3.2.5 插补进给速度比率设定 (IFP)”。
- 当 F 指令值乘以额定速度比后的速度超过插补进给最高速度 (FMX) 时，被固定为 FMX 速度。

2.1.3 圆弧插补 (MCW、MCC)

(1) 概要

圆弧插补 (MCW、MCC) 是指在指定平面上, 同时使 2 个轴在由圆心位置 (U-V-) 或半径值 (R) 确定的圆弧上, 以插补进给速度从当前位置到终点位置 (F) 移动的命令。

(2) 详细说明

圆弧插补命令的指令方法示例如下所示。

MCW	[axis1]	—	[axis2]	—	U	—	V	—	T	—	F	—	;
			A				B		C		D		
	A:		终点位置				B:		圆心位置		C:		转数
			D:						插补进给速度				
或者													
MCC	[axis1]	—	[axis2]	—	R	—	F	—	;				
			A				B		C				
	A:		终点				B:		半径		C:		插补进给速度

(注) 圆心位置指定时, 可指定多圈。(也可省略)

圆弧插补命令的旋转方向如下所示。

- MCW: 顺时针方向旋转 (CW)
- MCC: 逆时针方向旋转 (CCW)

重要

- 在圆弧插补命令之前, 请务必用坐标平面指定 (PLN) 命令指定圆弧插补平面。圆弧插补命令中必须指令圆弧的旋转方向 MCW 或者 MCC。用 [axis1]、[axis2] 指令指定平面的横轴、纵轴的终点位置和圆弧的圆心位置。
- 请与用 PLN 命令指定的横轴名、纵轴名顺序相对应来指定终点位置、圆弧圆心。

圆弧插补命令的指令方法如下图所示。

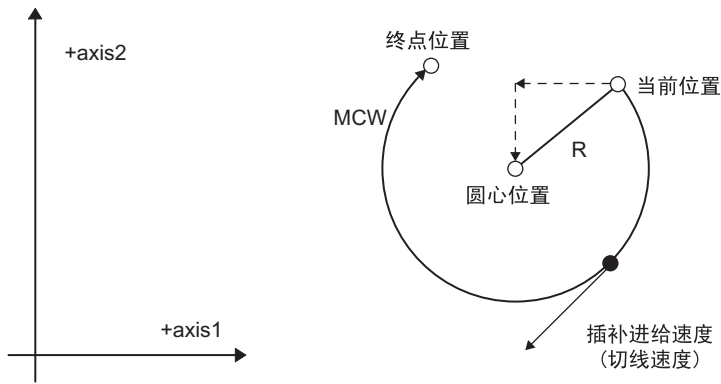


图 2.6 圆弧插补 (MCW, MCC) 命令的指令方法

用最后指定的 ABS / INC 模式执行终点位置、圆心位置的指定。

(3) 程序举例

(a) ABS 模式下的程序举例

例 ABS 模式下的 MCC 命令的程序举例如下。

```
ABS;
FMX T30000000;
MW00001=10000; ( * )
PLN [X] [Y];
MCC [X]1500 [Y]4000 U2500 V1000 F150;
```

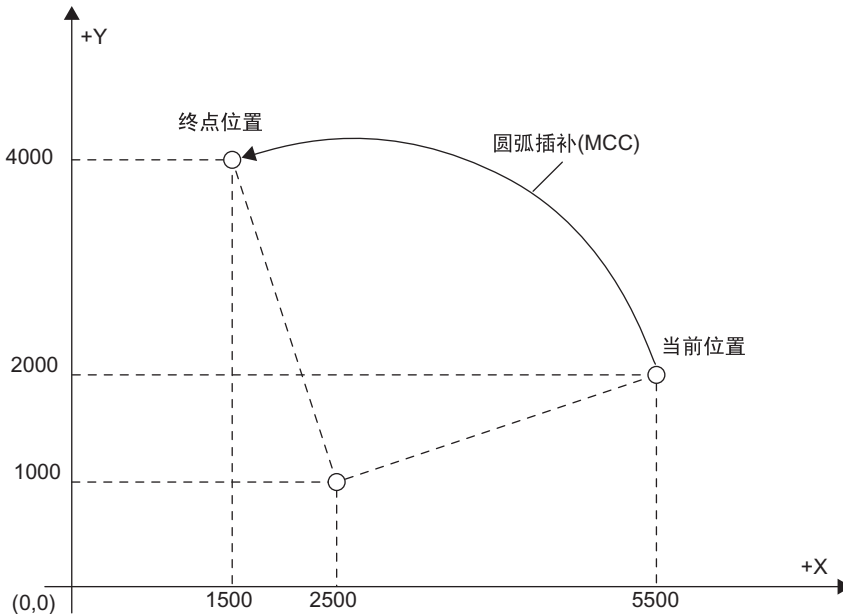


图 2.7 ABS 模式下的程序举例

* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。

(b) INC 模式下的程序举例



INC 模式下的圆弧插补 (MCC) 命令的程序举例如下所示。

```
INC;
FMX T30000000;
MW00001=10000; (* )
PLN [X] [Y];
MCC [X]-4000 [Y]2000 U-3000 V-1000 F150;
```

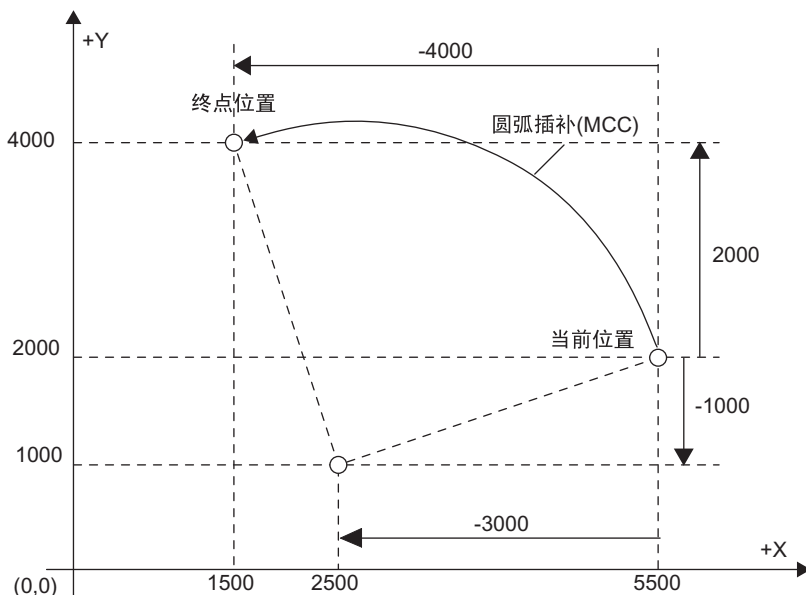


图 2.8 INC 模式下的程序举例

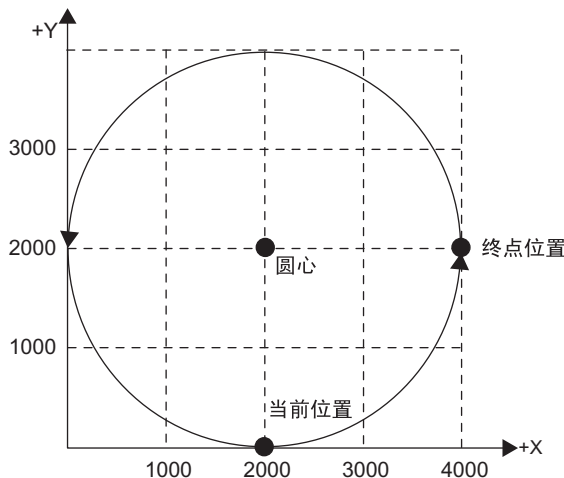
* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。

(c) 多圈圆周的程序举例



用圆心位置指定的多圈圆周的程序举例如下所示。

```
ABS;
FMX T30000000;
MW00001=10000; (* )
PLN [X] [Y];
MCC [X]4000 [Y]2000 U2000 V2000 T2 F150;
```



上述举例为 2 圈 +1/4 的圆弧。

* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。

补充

像 MCC [X]2000 [Y]0 U2000 V2000 T2 F150;
 当前位置=终点位置时, 为 3 圈。

除圆心位置外, 也可用圆弧的半径值 R 来指定圆弧。此时的圆弧插补如下图所示。

MCW [axis1] - [axis2] - R -; 的指令
 R > 0 时 : 圆弧角 180° 以下的圆弧插补
 R < 0 时 : 圆弧角 180° 以上的圆弧插补
 (注)R = 0 时 : 警报

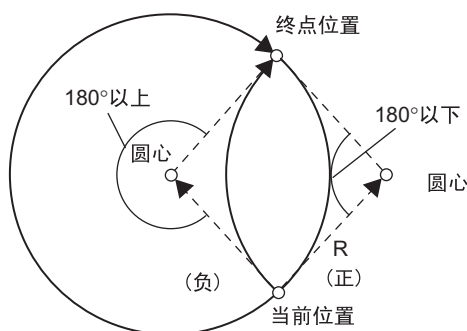


图 2.9 半径指定的圆弧插补

在圆弧插补的指令之前, 请用坐标平面指定 PLN 命令指令平面指定。没有指定时, 最后指定的平面为有效。没有任何指定时, 成为警报。

插补进给速度也可称为 F 指令。用速度指定 (F) 或速度指令 (%) (IFP) 进行指令。F 指令在以前的块中指令过的最终指令有效。没有对 F 指令进行任何指定的插补指令成为警报。

指令过的圆弧切向速度和 F 指令值相等。当向插补进给最高速度 (FMX) 指定超过设定界限值的 F 指令时, 则成为警报。

有关插补进给最高速度如下图所示。

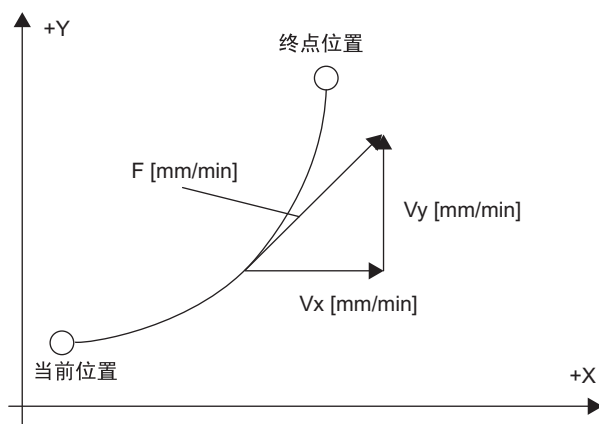


图 2.10 插补进给最高速度

在实际程序运行中, 可对 F 指令值乘上 0 ~ 327.67% 的额定速度比。额定速度比立即有效。

重要

在制作使用插补命令的运动程序时, 请在程序的开头指令插补进给最高速度 (FMX)。不指定时, 成为警报。

不管有无移动指令，如果想在入位检查后执行下面的块，请在该块或下面的块中进行指令。

在根据插补命令的移动自动加减速控制中，可通过参数设定及 IAC、IDC 命令的设定选择以下的加减速。

- 1 段直线加减速
- 非对称加减速
- S 字加减速

圆心位置指定时，通过使起点和终点位置一致，可以用 1 个块指定 1 个完整的圆。此外，通过转数指定 (T-)，也可用 1 个块指定多个圈。但是，根据半径 R 指定时，不能指令完整的圆及多圈圆周。

补充

速度指令、加减速、额定速度比等的指定方法全都与直线插补相同。详情请参阅“2.1.2 直线插补 (MVS)”。

(4) 程序举例

例 ABS 模式下的圆心位置指定的程序举例如下。

```
ABS;
FMX T30000000;
MW00001=10000; ( * )
MOV [X] 0 [Y] 0;
PLN [X] [Y];
MCW [X]0 [Y]0 U1000 V0;
```

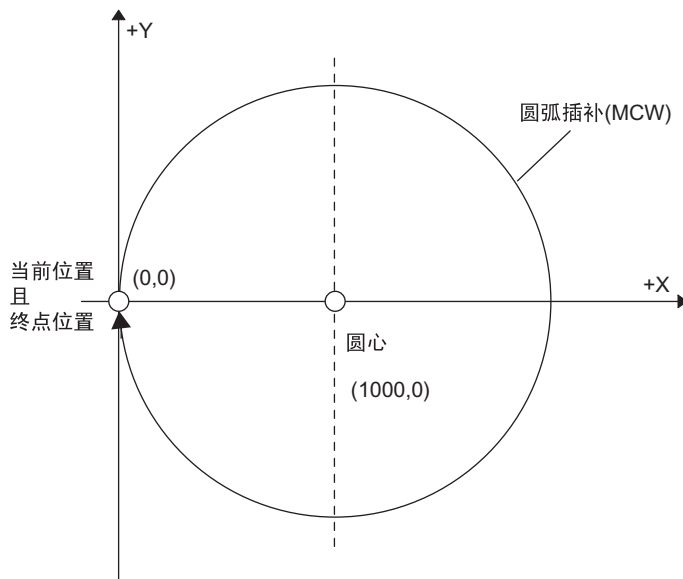


图 2.11 圆心位置指定的程序举例

* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。

补充

- 仅当与插补命令相同块时可指令速度指定 (F) 指令。
- 不能以单独指令向与插补指令相同的块指令速度指令 (%) (IFP)。
- 当 F 指令值乘以额定速度比的速度超过插补进给最高速度 (FMX) 时，被固定为 FMX 速度。

2.1.4 螺旋插补 (MCW、MCC)

⚠ 注意

- 在“螺旋插补 (MCW、MCC)”中，进行直线插补的轴可以是直线轴也可以是旋转轴。但是，由于直线插补部的轴的取向，螺旋插补的轨迹不会变成“螺旋形状”。编程时，为了不使工具和工件等发生干涉，必须进行轨迹检查。

如果疏于检查，可能发生因干涉引起的工具损坏以及相应的人身事故。

(1) 概要

螺旋插补 (MCW、MCC) 是圆弧插补命令扩展后的命令。在由指定的圆心位置或半径值 (R) 确定的圆弧上移动 (圆弧插补) 时，也同步进行直线插补的移动。

移动的圆弧插补切线速度变为插补进给速度 (F 指令)。这个移动动作即螺旋插补。圆弧指令部分的指令方面的规则，遵循圆弧插补的规则。在此不再表述。

(2) 详细说明

螺旋插补命令的指令方法如下所示。

MCW	[axis1]	—	[axis2]	—	U	—	V	—	[axis3]	—	T	—	F	—	;
			A		B		C		D		E				

A: 终点位置
B: 圆心位置
C: 直线插补的终点位置
D: 转数
E: 插补进给速度

或者

MCC	[axis1]	—	[axis2]	—	R	—	[axis3]	—	F	—	;
			A		B		C		D		

A: 终点
B: 半径
C: 直线插补的终点位置
D: 插补进给速度

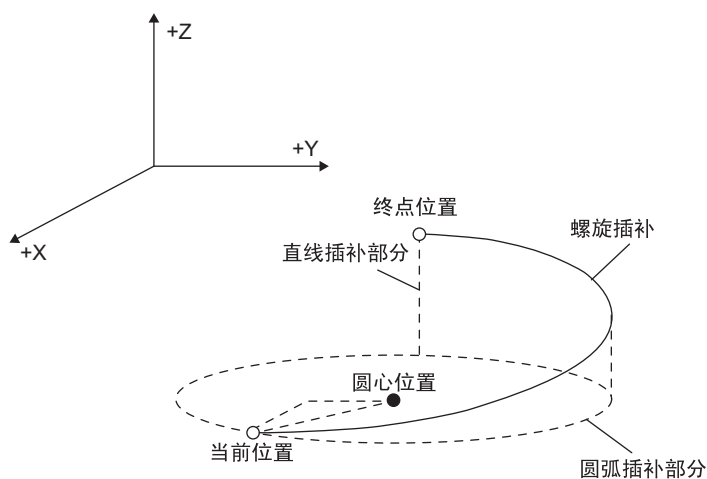


图 2.12 螺旋插补命令

可向直线插补轴符号指令未用平面指定指定的单轴。插补平面没有必要一定是直角。在螺旋插补的指令之前，请用坐标平面指定 (PLN) 命令指令平面指定。

请与用 PLN 指令指定的横轴名、纵轴名顺序相对应来指定终点位置、圆弧圆心。

插补进给速度也可称为 F 指令，可用速度指定 (F) 或速度指令 (%) (IFP) 进行指令。

F 指令中在以前的块中指令过的最终指令有效。没有对 F 指令进行任何指定的插补指令成为警报。

插补进给速度 F 表示圆弧平面中的圆弧切线速度。由此，直线轴的速度 (F') 可以是： $F' = F \times (\text{直线轴的长度}) / (\text{圆弧长度})$ 。

补充

速度指令、加减速、额定速度比等的指定方法全都与直线插补相同。详情请参阅“2.1.2 直线插补 (MVS)”。

(3) 程序举例

例

ABS 模式下的螺旋插补命令的程序举例如下所示。

```
ABS;
FMX T30000000;
MW00001=10000; (* )
MOV [X]1000 [Y]0 [Z]0;
PLN [X] [Y];
MCC [X]0 [Y]1000 U0 V0 [Z]500;
```

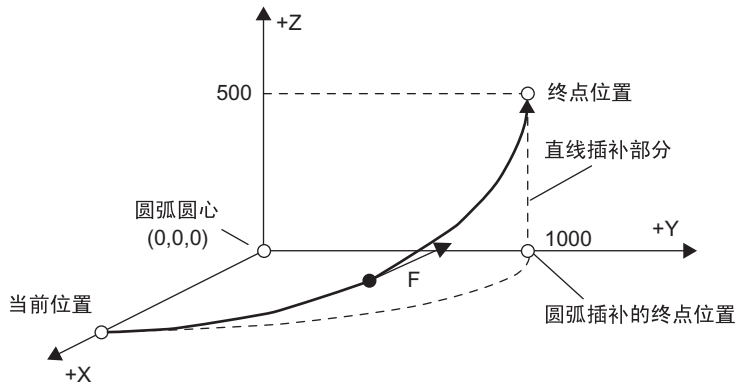


图 2.13 螺旋插补命令的程序举例

* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。

补充

- 仅当与插补命令相同块时可指令速度指定 (F) 指令。
- 不能以单独指令向插补指令相同的块指令速度指令 (%) (IFP)。
- 当 F 指令值乘以额定速度比的速度超过插补进给最高速度 (FMX) 时，被固定为 FMX 的速度。

2.1.5 原点复归 (ZRN)

(1) 概要

原点复归 (ZRN) 是进行原点复归动作的命令。最多可同时指定 14 轴*。原点复归后的停止位置被设定为机械坐标系的原点。

省略指令的轴不进行原点复归动作。原点复归动作结束后指令过的所有轴进入下面的块。

* 为 MP930 的示例。机型不同最多轴数也不同，详情请参阅“1.1.2 功能一览”。

(2) 详细说明

ZRN 命令的指令方法和移动轨迹如下所示。

```
ZRN [axis1] - [axis2] - ...;
      轴指定 + 0 (位置始终在 0 位)
```

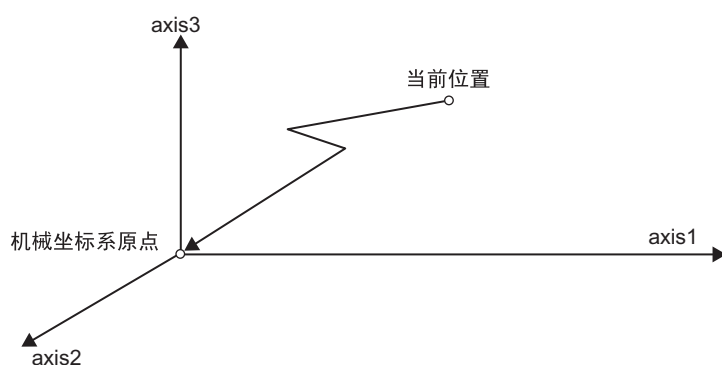


图 2.14 根据原点复归动作的移动轨迹

执行 ZRN 指令后，将复归后位置设置为机械坐标系的原点。同时，用当前值变更 (POS) 命令取消以前设定的工件坐标系。

执行 ZRN 命令，则机械坐标系和工件坐标系将一致。此后，直到当前值变更 (POS) 命令被执行为止，机械坐标系指令 (MVM) 都被视为无效指令。

重要

在 ZRN 命令中，程序暂时停止请求无效。在运行途中停止时，请进行程序停止请求。

原点复归动作的种类如下表所示。

MP900 系列：用固定参数 26 “原点复归方式”选择。

MP2000 系列：用设定参数 0Wxx3C “原点复归方式”选择

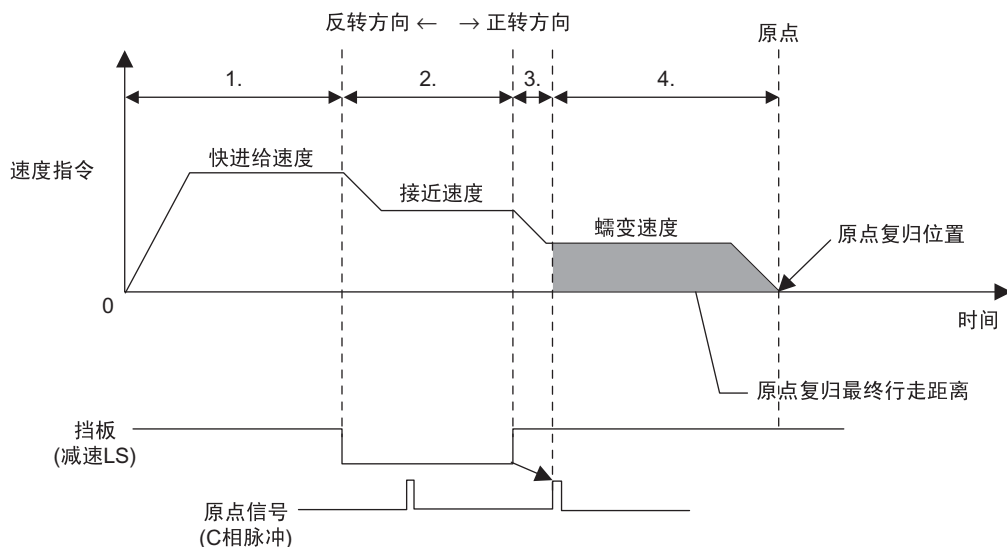
名称	设定值	MP910	MP920	MP930	MP940	MP2100	MP2300
		固定参数 26 原点复归方式				设定参数 0Wxx3C 原点复归方式	
DEC1+C 相脉冲方式	0	○	○	○	○	○	○
ZERO 信号方式	1	○	○	○	○	○	○
DEC1+ZERO 信号方式	2	○	○	○	○	○	○
C 相脉冲方式	3	○	○	○	○	○	○
DEC2+ZERO 信号方式	4	-	○	-	○	-	-
DEC1+LMT+ZERO 信号方式	5	-	○	-	○	-	-
DEC2+C 相脉冲方式	6	-	○	-	○	-	-
DEC1+LMT+C 相脉冲方式	7	-	○	-	○	-	-

(3) 原点复归方式

原点复归方式如下。

(a) 0:DEC1+C 相脉冲方式

用 3 段速度切换方式进行原点复位。



(b) 1: ZERO 信号方式

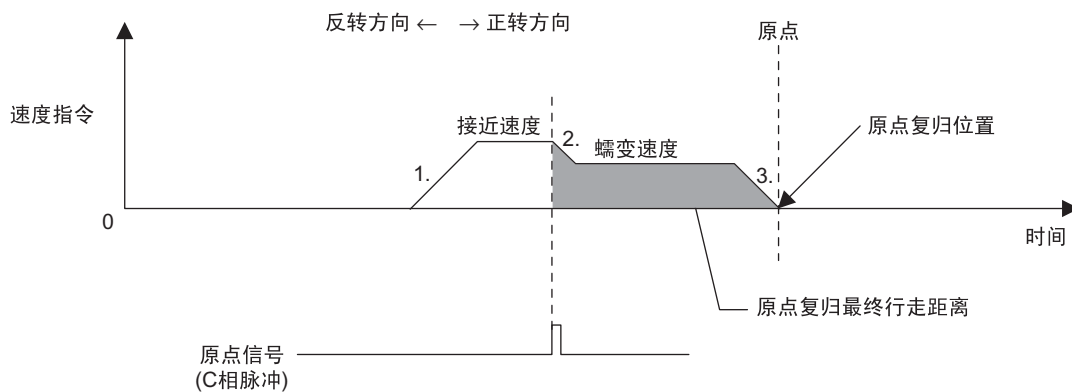
代替“C 相脉冲方式”的 C 相脉冲, 用 ZERO 信号进行原点复归。

(c) 2:DEC1+ZERO 信号方式

代替“DEC1+C 相脉冲方式”的 C 相脉冲, 用 ZERO 信号进行原点复归。

(d) 3:C 相脉冲方式

在不能安装减速 LS 等的机器中, 仅用电机的 C 相脉冲进行原点复归。



(e) 4:DEC2+ZERO 信号方式

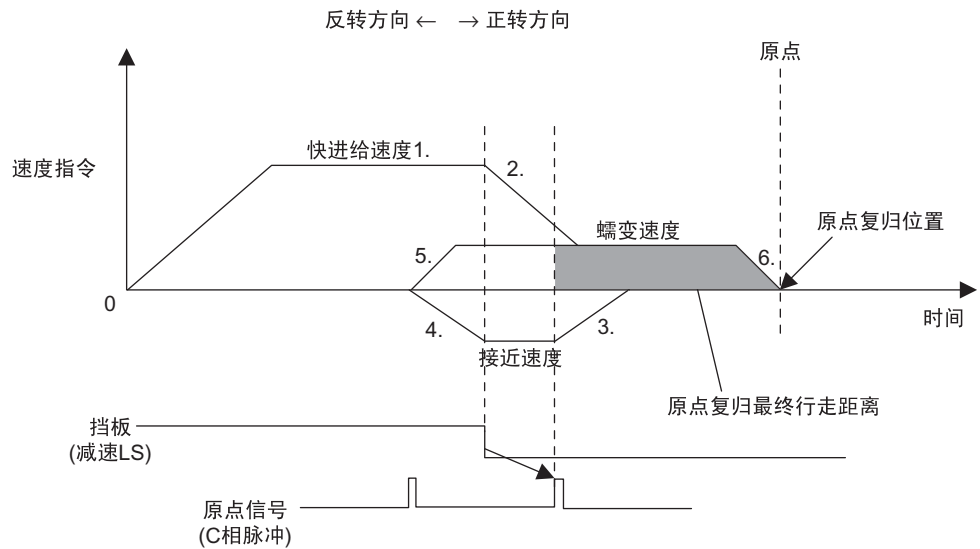
代替“DEC2+C 相脉冲方式”的 C 相脉冲, 用 ZERO 信号进行原点复归。

(f) 5:DEC1+LMT+ZERO 信号方式

“代替 DEC1+LMT+C 相脉冲方式”的 C 相脉冲, 用 ZERO 信号进行原点复归。

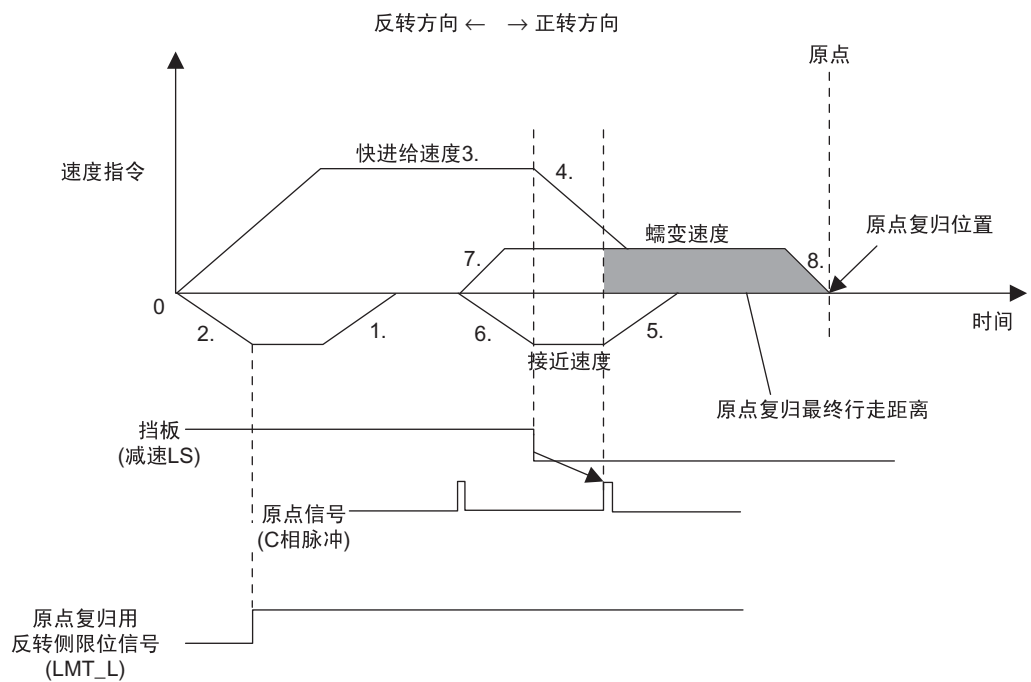
(g) 6:DEC2+C 相脉冲方式

以接近速度反转后，再以蠕变速度检索原点，进行原点复归。
该方式用于重复精度要求较高的机器。



(h) 7:DEC1+LMT+C 相脉冲方式

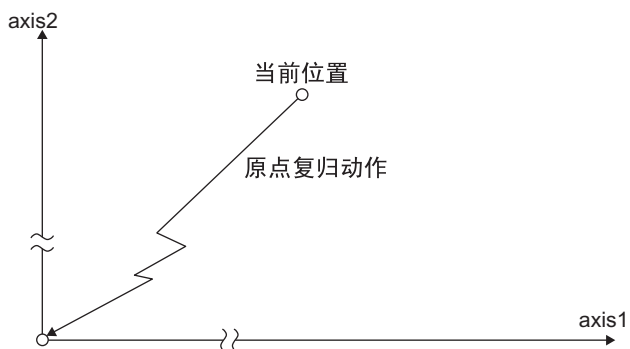
通过正 / 反LMT信号，获知当前位置，自动进行避让动作。从任何位置都可进行原点复归。



(4) 程序举例 (以 MP930+ΣI 伺服为例)

例 ABS 模式下的原点复归动作程序举例如下所示。

```
ABS;
ZRN[axis1]0 [axis2]0;
```



将停止位置设定为机械坐标系原点(0, 0)

图 2.15 原点复归动作程序举例

原点复位相关参数及伺服单元用户参数如下表所示。

- | | |
|------------|----------------------------------|
| • 原点复归方向 | 设定参数 “动作模式设定, (0Wxx0069 原点复归方向)” |
| • 原点复归进给速度 | 设定参数 30 “快进给速度 (0Lxx22)” |
| • 接近速度 | Cn-0022 “原点复归接近速度 1” |
| • 蠕变速度 | Cn-0023 “原点复归接近速度 2” |
| • 最终行走距离 | Cn-0028 “原点复归最终行走距离” |

补充

请务必指定轴后数据为 0。省略则会出错。

2.1.6 跳过功能 (SKP)

(1) 概要

跳过功能 (SKP) 是指最多同时进行 14 轴 * 直线插补动作, 以插补进给速度 (F) 从当前位置向终点位置移动的指令。移动时跳过信号为 ON 时, 移动中的轴减速停止, 剩余移动量的指令被取消。不移动省略指令的轴。

在 SKP 命令块中进行轴移动时, 当跳过输入信号为 ON 时, 轴移动将减速停止, 取消该块剩余的移动量并进入下面的块。像这样根据 SKP 命令, 可对适应外部状况的运动控制进行编程。

(2) 详细说明

SKP 命令的指令方法如下所示。

SKP	[axis1]	-	[axis2]	-	...	F	-	SS	;
	A		B			C			
	A: 指令位置								
	B: 插补进给速度								
	C: 跳过选择								

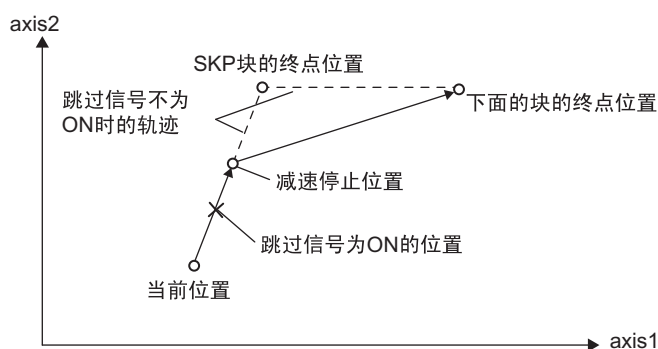


图 2.16 SKP 命令轨迹

自动生成梯形形时, 跳过输入信号可把最多 14 个轴 * 分成 4 组, 对它们进行单独控制。1 组可进行 2 点分配。

跳过选择 SS 可通过写入 1 或者 2 选择跳过输入信号。请通过组定义画面的控制信号分配, 预先决定哪个跳过信号对应 1 或 2。

跳过输入信号输入到 MSEE 命令的控制信号内。

- 跳过输入信号 1: 控制信号的 Bit 8
- 跳过输入信号 2: 控制信号的 Bit 9

* 为 MP930 的示例。机型不同最多轴数也不同, 详情请参阅 “1.1.2 功能一览”。



- 速度指定 (F) 指令仅当与 SKP 指令相同块时可指令。
- 不能以单独指令向与插补指令相同的块指令速度指令 (%) (IFP)。
- 当 F 指令值乘以额定速度比的速度超过插补进给最高速度 (FMX) 时, 被固定为 FMX 速度。
- 当跳过输入信号为 ON 时, 轴减速停止, 但直到定位结束信号为 ON 为止 SKIP 指令一直有效。

2.1.7 时间指定定位 (MVT)

(1) 概要

时间指定定位 (MVT) 是指用定位 (MOV) 动作从当前位置向终点位置移动各轴, 为了在指定时间内完成定位, 固定各轴的进给速度的命令。因该命令不是插补动作, 所以未必所有指令轴能同时完成定位。根据加减速设定的不同, 会有时间差。

最多可同时指令 14 轴*。不移动省略指令的轴。

* 为 MP930 的示例。机型不同最多轴数也不同, 详情请参阅“1.1.2 功能一览”。

(2) 详细说明

MVT 命令的指令方法如下所示。

MVT	[axis1]	—	[axis2]	—	...	T	— ;
			指令位置			定位时间 (msec)	

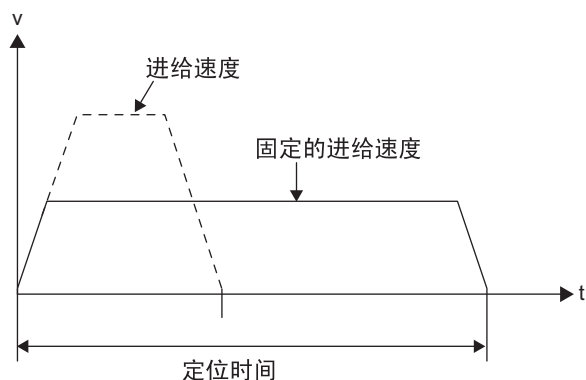


图 2.17 时间指定定位 (MVT)

使用额定速度比时, 在指定时间内不能完成定位。

使用滤波器时, 定位时间仅延迟滤波时间参数部分的时间。

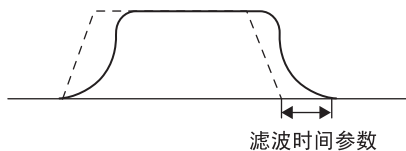


图 2.18 滤波器使用时的定位时间延迟

重要

- 定位时间设定为 0 时, 成为运动程序警报。
- 当有移动量为 0 的轴时, 成为警报。

(3) 程序举例

例 ABS 模式下的时间指定定位 (MVT) 程序举例如下所示。

```
ABS;
MVT [axis1]100[axis2]200 T1000;
```

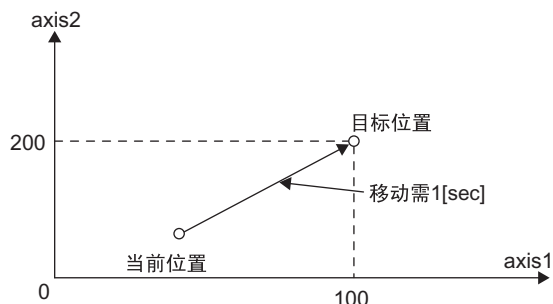


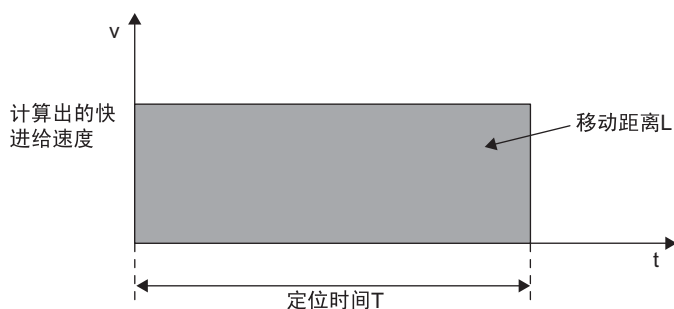
图 2.19 时间指定定位 (MVT) 程序举例



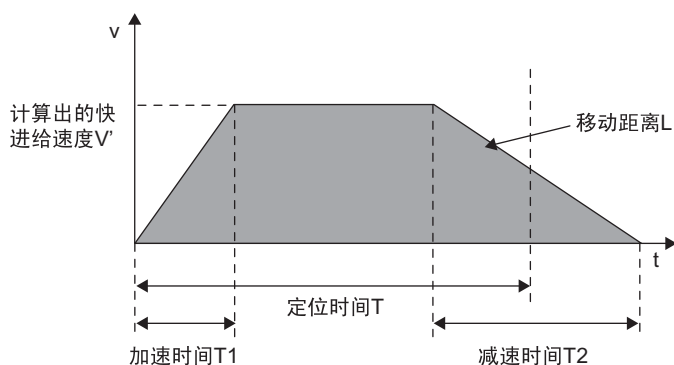
■ 关于时间指定定位 (MVT 命令)

MP 系列中 MVT 命令的动作原理都相同。

在 MP 内，MVT 命令时的进给速度作为加减速 0 计算。



实际动作，如下所示。(加速时间 $T_1 <$ 减速时间 T_2 时)



加速时间和减速时间相同 ($T_1=T_2$) 时，用 MVT 命令指定的定位时间即到减速开始的时间。

根据 MVT 命令，可以改写使用的各轴设定参数“快进给速度 (OLxx22/OLxx10)”的值。由此，因用 VEL 命令“进给速度变更”设定的进给速度也可被改写，所以执行 MVT 命令后，请再次用 VEL 命令设定进给速度。

在 SVA 和 Mechatrolink 中，加减速时间的原理不同。

- SVA: 设定参数的加减速时间根据滤波器的设定而定。
- Mechatrolink: 伺服单元的加减速时间参数根据滤波器的设定而定。

2.1.8 外部定位 (EXM)

(1) 概要

外部定位 (EXM) 是指在定位指令执行中, 输入外部定位信号时, 仅用指定轴后的移动量增量值定位后执行下面的块的命令。

(2) 详细说明

EXM 命令的指令方法如下所示。

<pre>EXM [axis] — D — ; A B</pre> <p>A: 指令位置 B: 外部定位信号输入时开始的移动量 (仅限增量值)</p>
--

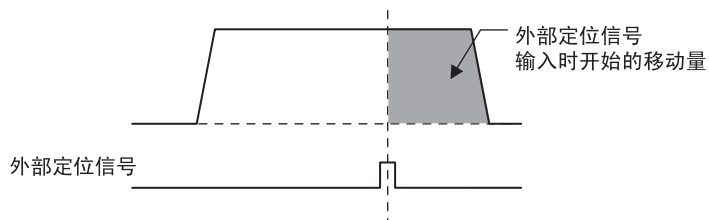


图 2.20 外部定位 (EXM) 信号

当对移动量指定负值时, 则减速停止后向负方向移动。EXM 命令仅 1 轴有效。

机器锁定时, 无视信号仅执行指定的移动量。当不输入信号时, 移动到指令位置。

重要

- 请把外部定位信号连接到伺服单元的 1CN 10 针: 外部门锁 (/EXT) 上。
- 由于使用原点 LS 的原点复归中也使用外部门锁 (/EXT) 输入信号, 因此使用时敬请注意。
- D 指定 (外部定位信号输入开始的移动量) 单位为脉冲。

2.2 控制命令

控制命令是指控制各轴运动的命令。本节对基本控制命令的编程进行说明。

2.2.1 绝对值 (ABS) 模式

⚠ 注意

- 用绝对值模式和增量值模式指令相同的坐标符号时，其移动动作完全不同。在运行之前，请务必检查是否正确指令了“ABS / INC”命令。

如果疏于检查，可能发生因干涉引起的工具损坏以及相应的人身事故。

(1) 概要

绝对值 (ABS) 模式是指把进行轴移动的命令的坐标符号作为工件坐标系上的绝对值处理的命令。

ABS 模式命令一旦被指令后，到下一个增量值 (INC) 模式命令被指令为止，一直保持 ABS 模式命令。电源接通时，为 ABS 模式命令。

(2) 详细说明

ABS 模式命令的指令方法如下所示。

```
ABS;
或者
ABS MOV [axis1] -;
```

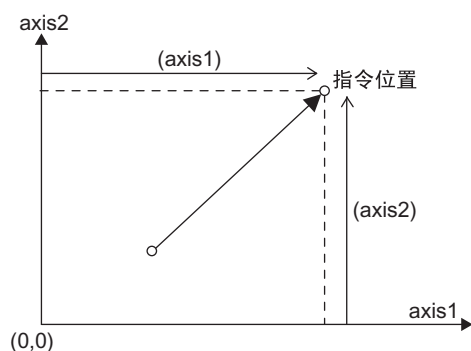


图 2.21 工件坐标上的绝对值

本书中有时把“绝对值模式”缩写为“ABS 模式”。

(3) 程序举例



ABS 模式命令的程序举例如下所示。

```
ABS MOV [axis1]100 [axis2]200;  
MOV [axis1]50 [axis2]100;  
MOV [axis1]100;  
MOV [axis2]50;
```

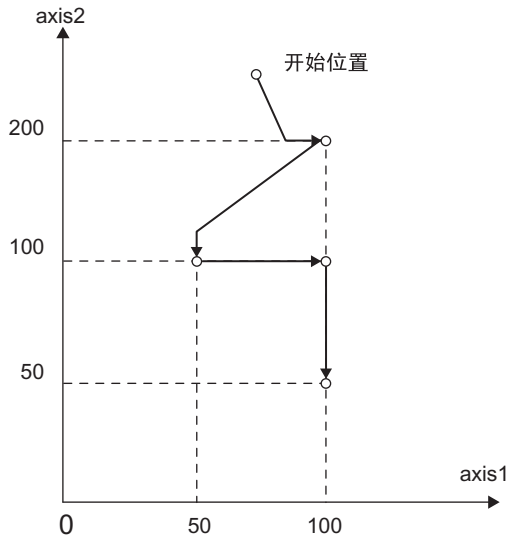


图 2.22 ABS 模式命令的程序举例

2.2.2 增量值 (INC) 模式

(1) 概要

增量值 (INC) 模式是指把进行轴移动的命令的坐标符号作为工件坐标系当前位置开始的增量值处理的命令。

INC 模式命令一旦被指令, 到下一 ABS 模式命令被指令为止, 一直保持 INC 模式命令。电源接通时, 为 ABS 模式命令。

(2) 详细说明

INC 模式命令的指令方法如下所示。

```
INC;
或者
INC MOV [axis1] -;
```

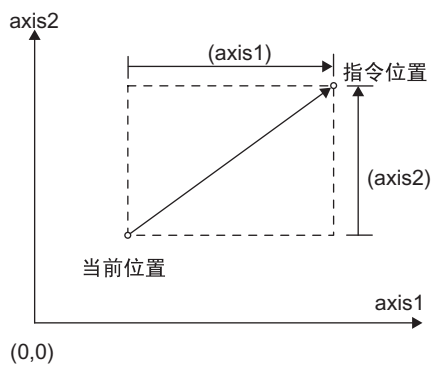


图 2.23 工件坐标上的增量值

(3) 程序举例

例 INC 模式命令的程序举例如下所示。

```
INC MOV [axis1]100 [axis2]200;
MOV [axis1]50 [axis2]100;
MOV [axis1]100;
MOV [axis2]50;
但
开始位置 :axis1 = axis2 = 0
```

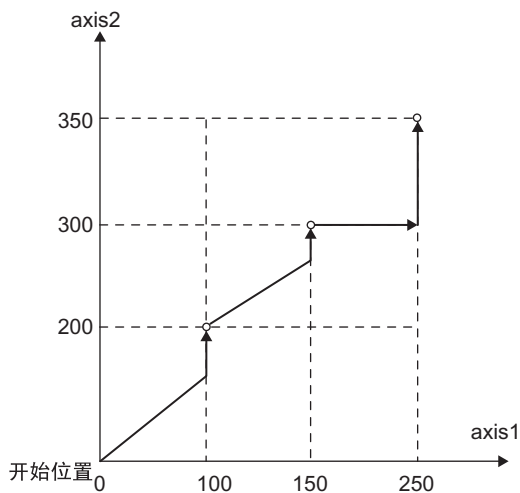


图 2.24 INC 模式命令的程序举例

2.2.3 当前值变更 (POS)

⚠ 注意

• “当前值变更 (POS)” 命令注意事项。

“当前值变更 (POS)” 命令产生新的“工件坐标值”。因此，错误指令“POS”命令时，之后的移动动作会完全不同。在运行之前，请务必指示正确的“工件坐标系”是否被指令。

如果疏于检查，可能发生因干涉引起的工具损坏以及相应的人身事故。

(1) 概要

当前值变更 (POS) 是指把当前位置改写为“希望变更的坐标值”后，产生新的坐标系的命令。本书把新设定的坐标系称为工件坐标系。在 POS 命令之后指令的移动指令在工件坐标系上移动。

(2) 详细说明

POS 命令的指令方法如下所示。

```
POS [axis1] - [axis2] - ...;
    希望变更的坐标值
```

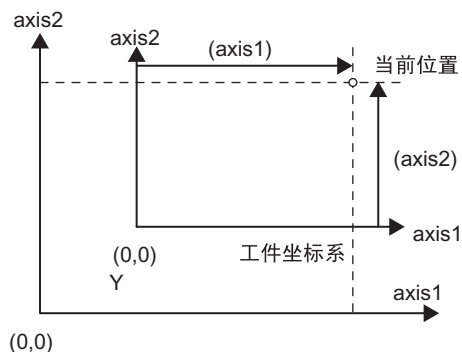


图 2.25 (工件坐标变更) POS 命令

根据 POS 命令的工件坐标系可进行多次切换。请事先设定机械坐标系。POS 命令对机械坐标系没有影响。

POS 命令最多可同时指令 14 轴*。省略指令的轴不可改写当前值。

在工件坐标系上的移动指令在机械坐标系上换算后若超过最大指令值，则不能指定。

* 为 MP930 的示例。机型不同最多轴数也不同，详情请参阅“1.1.2 功能一览”。

机械坐标系和工件坐标系的设定状态如下表所示。

表 2.1 当前值变更 (POS)

本产品的状态	增量值检测系统	绝对值检测系统
电源接通操作后	机械坐标系 : 临时设定 * ¹ 工件坐标系 : 取消 * ³	机械坐标系 : 有 * ² 工件坐标系 : 取消
原点复归 (ZRN) 执行后	机械坐标系 : 设定 工件坐标系 : 取消	工件坐标系 : 取消
POS 命令执行后	工件坐标系 : 设定	工件坐标系 : 设定
原点设定操作后	工件坐标系 : 取消	机械坐标系 : 设定 工件坐标系 : 取消

- * 1. 临时设定 : 设定将电源接通时的当前位置作为原点的机械坐标系。此后, 如果不进行原点复归, 软件 LS 功能将无效。
- * 2. 有 : 用绝对值检测编码器的位置信息来定机械坐标原点。
- * 3. 取消 : 以前设定的工件坐标系被取消, 工件坐标系和机械坐标系相同。

重要

对设定为无限长的轴, 只能在 0 ~ POSMAX 范围内进行设定。如果设定此范围以外的值, 则成为警报。

2

2.2.4 坐标平面指定 (PLN)

(1) 概要

坐标平面指定 (PLN) 命令是指定 2 个用参数设定的逻辑轴来定义坐标平面的命令。在指定坐标平面之前，请务必执行该命令。

一旦指令坐标平面，直到再定义或程序执行 END 命令为止该坐标平面有效。

(2) 详细说明

PLN 命令的指令方法如下所示。

横轴名	纵轴名
PLN [axis1]	[axis2];
指定坐标平面的 2 个轴	

(3) 程序举例



PLN 命令的程序举例如下所示。

PLN[A] [B];……………指定由 A, B 轴构成的平面。

MCW [A]50 [B]50 R50 F1000;

但

开始位置: A = B = 0

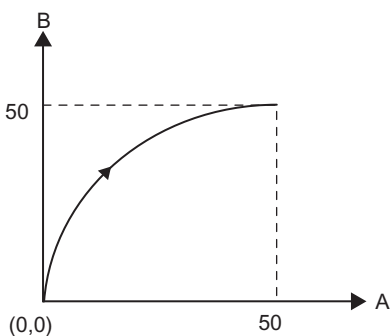


图 2.26 坐标平面指定 (PLN) 的程序举例



在指定圆弧插补及螺旋插补的终点位置和圆心位置时，请与用 PLN 命令指定的横轴名、纵轴名顺序相对应。

```

PLN [axis1] [axis2];
MCC [axis1]1500 [axis2]4000 U2500 V1000 F150;
  
```

2.2.5 机械坐标指令 (MVM)

⚠ 注意
<ul style="list-style-type: none"> • “机械坐标指令 (MVM)” <p>“机械坐标指令 (MVM)” 命令可以临时向 “机械坐标系” 上的坐标位置进行定位。由此，如果没有确认 “机械坐标系” 的原点位置就执行指令时，将会发生意想不到的移动动作。在运行之前，请务必检查是否指令了 “机械坐标系” 上的正确位置。</p> <p>如果疏于检查，可能发生因干涉引起的工具损坏以及相应的人身事故。</p>

(1) 概要

机械坐标指令 (MVM) 是指根据当前值变更命令 (POS)，在设定与机械坐标系不同的工件坐标系后，临时想在机械坐标系上移动时使用的命令。

(2) 详细说明

MVM 命令的指令方法如下所示。

<pre>MVM MOV; 或者 MVM MVS;</pre>

通过该指令，可以临时向机械坐标系的绝对位置用定位 (MOV) 命令或直线插补 (MVS) 命令进行移动。不管是 ABS 模式指定还是 INC 模式指定，该命令都以 ABS 模式进行动作。

MVM 命令仅在指定的块中有效。例如，在下面的块以后的直线插补 (MVS) 命令中，则为在工件坐标系上的移动。请参照下图。

(3) 程序举例



MVM 命令的程序举例如下所示。

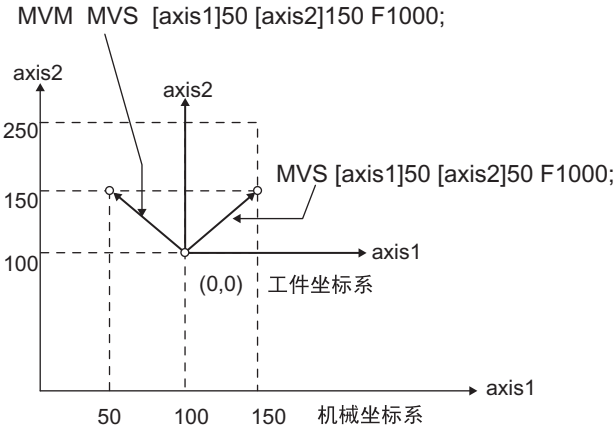


图 2.27 机械坐标指令 (MVM) 命令的程序举例

2.2.6 程序当前位置更新 (PLD)

(1) 概要

运动程序运行中，在运动程序以外（JOG 动作和 STEP 动作及使用用户函数移动轴等）使轴动作时，“程序当前位置”将不会被更新。在此状态下，如果继续执行运动程序，会移动到仅手动移动量的变更位置。为了解决该问题，可以使用更新“程序当前位置”的 PLD 命令。

另外，该命令不适用于块运行。（为块运行模式中不能停止的命令。）

(2) 指令方法

```
PLD [axis1] [axis2]...[axis];
      [axis1]...[axisn]: 不能更新未指定执行“程序当前位置”更新的轴。
```

(3) 程序举例

(a) 运动程序运行中进行手动运行时

```
MPM001 “GRP1”
MOV [axis1]1000;
      ←在此之间用 JOG 移动了 [axis1] 轴
PLD [axis1]; ←更新“程序当前位置”
MVS [axis1] - 1000;
```

(b) 在运动程序的用户函数中移动轴时

```
MPM001 “GRP1”
MOV [axis1]1000;
UFC FNC10 MB00000 IW00100 MB00020 ←用户函数中移动了 [axis1] 轴
PLD [axis1]; ←更新“程序当前位置”
MVS [axis1] - 1000;
```

 补充

PLD 命令根据不同用途由用户操作执行。在运动程序运行中，手动应用程序有时也不使用 PLD 命令。

2.2.7 待时 (TIM)

(1) 概要

待时 (TIM) 命令是指根据下一指令, 等待用字符 “T” 指定的时间后, 进入下面的块的命令。
TIM 命令不能和其他命令重复指令。

(2) 详细说明

TIM 命令的指令方法如下所示。

```
TIM T_ ;
    等待时间
```

“T” 可指定的时间范围 :0.00 ~ 600.00 sec。

不用小数点指令时间 “T” 时。此时 1=0.01 sec。不受固定参数的 “小数点以下位数” 设定的影响。

(3) 程序举例

例 TIM 命令的程序举例如下所示。

```
MOV [axis1]100;
TIM T250 ;
    2.5sec
```

定位结束后, 执行 “TIM” 命令。

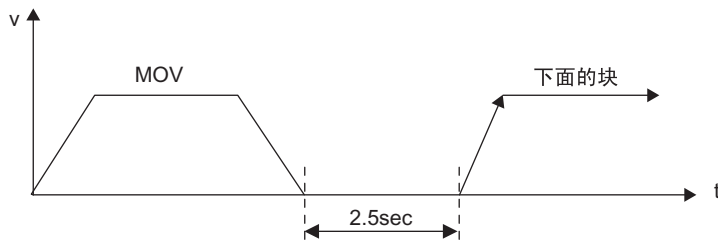


图 2.28 待时 (TIM) 命令的程序举例

2.2.8 程序的结束 (END)

(1) 概要

程序的结束 (END) 是指结束程序运行的命令。
在该块中，不能重复指令其他命令。

(2) 详细说明

END 命令的指令方法如下所示。

<code>END;</code> 程序的结束

在结束该块的执行后，通过该指令结束程序运行。
在前一块为移动指令时，程序将在入位检查后结束。

第 3 章

高级编程

本章对运动控制命令中有关高级控制命令、速度与加速度命令的编程方法进行了详细说明。

3.1 高级控制命令	3-2
3.1.1 入位检查 (PFN)	3-2
3.1.2 第 2 入位检查 (INP)	3-4
3.1.3 单块忽视 (SNG)	3-6
3.1.4 用户函数调用 (UFC)	3-7
3.1.5 输入输出变量待机 (IOW)	3-15
3.1.6 子程序调用 (MSEE)	3-16
3.1.7 子程序结束 (RET)	3-17
3.1.8 1 个扫描周期的 WAIT (待机) 命令 (EOX)	3-18
3.1.9 分支命令 (IF ELSE IEND)	3-19
3.1.10 循环命令 (WHILE WEND)	3-20
3.1.11 并列执行命令 (PFORK, JOINTO, PJOINT)	3-22
3.1.12 选择执行命令 (SFORK, JOINTO, SJOINT)	3-26
3.2 速度与加减速命令	3-28
3.2.1 加速时间变更 (ACC)	3-28
3.2.2 减速时间变更 (DCC)	3-30
3.2.3 S 字时间参数变更 (SCC)	3-32
3.2.4 进给速度变更 (VEL)	3-34
3.2.5 插补进给速度比率设定 (IFP)	3-36
3.2.6 插补进给最高速度设定 (FMX)	3-37
3.2.7 插补加速时间变更 (IAC)	3-39
3.2.8 插补减速时间变更 (IDC)	3-41

3.1 高级控制命令

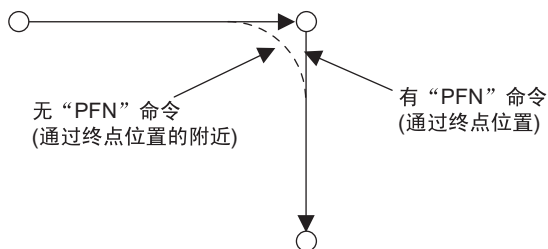
本节对运动控制命令中的高级命令的编程方法进行详细说明。

3.1.1 入位检查 (PFN)

(1) 概要

“入位检查 (PFN)¹”命令在插补命令的移动中的轴结束位置指令的输出，进入入位范围之前待机。进入入位范围后，则进入下面的块。

在根据“插补”命令的角移动时，想使其通过指定的“终点位置”时使用。



(2) 详细说明

PFN 命令的指令方法有以下 2 种。

(a) 指定和插补命令同一块时

```
MVS [axis1]100. [axis2]200. F1000 PFN;
```

用 MVS 命令进行指定轴的入位检查，进入“下面的块”。

(b) 单独指定时

```
MVS [axis1] - [axis2] ;
PFN [axis1] [axis2];
MVS [axis1] - [axis2] - ;
```

如果 [axis1], [axis2] 轴进入入位范围，则执行下面的块。

通过上述指令，对“PFN”命令前面的块的轴移动进行检测进入“定位结束范围”的“入位检查”后，进入“下面的块”。

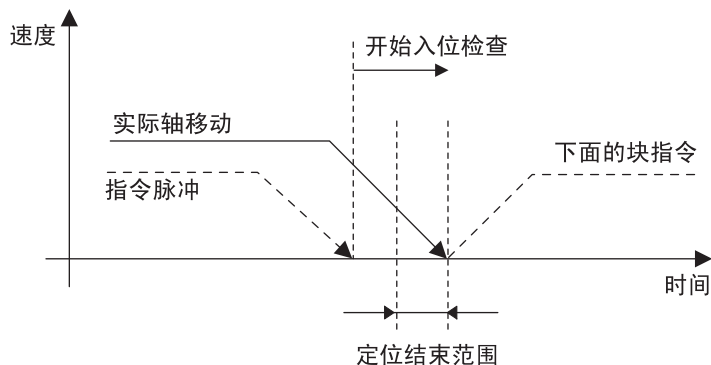


图 3.1 “入位检查过程”

¹ 入位检查

是指定块的移动开始减速后，检测进入“定位结束范围”的功能。

根据使用的控制器不同，“定位结束范围”的设定方法也不同。

- MP930、MP920 (SVB-01)、MP910、MP940、MP2300、MP2100 时
请设定为伺服单元用户参数“定位结束宽”。
- MP920 (SVA-01、SVA-02) 时
请设定为运动设定参数 No. 15 “定位结束范围设定”。

如下所述，“PFN”命令有指定为和插补命令同一块和单独指定为要进行入位检查的程序块的 2 种方法。

表 3.1 PFN 命令的指令方法 (2 种)

	指令种类	指示方法	备注
1	指定为和插补命令同一块	MVS [axis1]100. [axis2]200. F1000 PFN;	用 MVS 命令进行指定轴的入位检查，进入“下面的块”。
2	单独指定有轴指定	PFN [axis1] [axis2];	进行指定轴的入位检查（轴使用可能性的检查），进入“下面的块”。

(3) 程序举例



MVS [axis1]200. F100 PFN; 根据直线插补的移动结束等待
MOV [axis1]400. [axis2]400.; 根据定位的移动

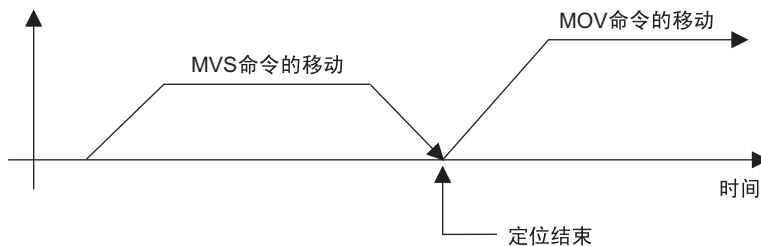
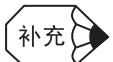


图 3.2 入位检查 (PFN)



没有“PFN”命令时，“MVS”命令的脉冲分配结束后，开始执行下面的块。

3.1.2 第2入位检查 (INP)

(1) 概要

“第2入位检查 (INP)”命令在根据“插补”移动时，想通过“终点位置”的附近（带弧度）时指令。

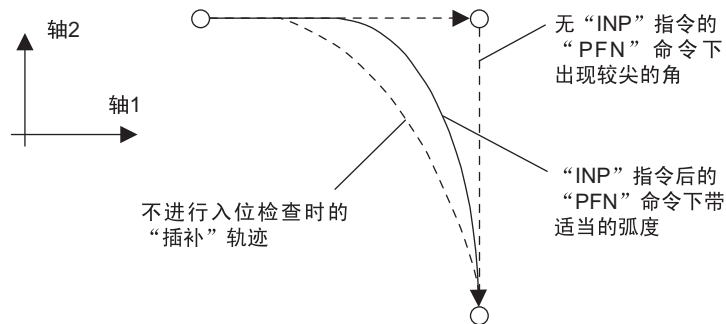


图 3.3 “第2入位检查”命令

(2) 详细说明

INP 命令的指令方法举例如下。

```
INP [axis1] - [axis2] - ... ;  
第2定位结束范围
```

根据上述指令，对以后有指令的“PFN”指定的“插补”的移动及单独的“PFN”指令，进行“第2入位检查”即进入“第2定位结束范围”检测后，进入下面的块。移动轴为复数时，在各轴都进入指定的“第2定位结束范围”后，进入下面的块。

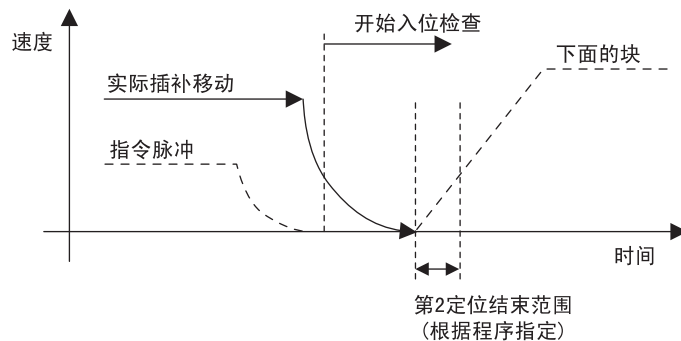


图 3.4 “第2入位检查”的指令方法

“第2定位结束范围”的设定范围如下所示。

1 ~ 65535 [指令单位]

但是，如上图所示，指令的块脉冲分配结束时即为第2入位检查的开始。因此，如果“第2位置结束范围”的设定值过大，在“脉冲分配结束时刻”时，第2入位检查也立即结束，进入下面的块。



- “INP”命令一旦设定，直到被取消为止，对所有的“插补”指令，第2入位检查有效。
- 要取消第2入位检查，只要对想取消的轴的“第2定位结束范围”指令“0”。只有指令为“0”的轴被取消。
- 可执行“入位检查(PFN)”命令。指令后，不管设定参数“第2定位结束范围”如何，都进行“入位检查”。“第2定位结束范围”为0时，进行根据“定位结束范围”的入位检查。

(3) 程序举例



```

ABS MOV [axis1]0 [axis2]0;    向原点的定位
INP [axis1](a) [axis2](b);    设定“第2定位结束范围”
MVS [axis1]100 PFN;           X轴方向、直线插补
MVS [axis2]100 PFN;           Y轴方向、直线插补
MVS [axis1]-100 PFN;          X轴方向、直线插补
  
```

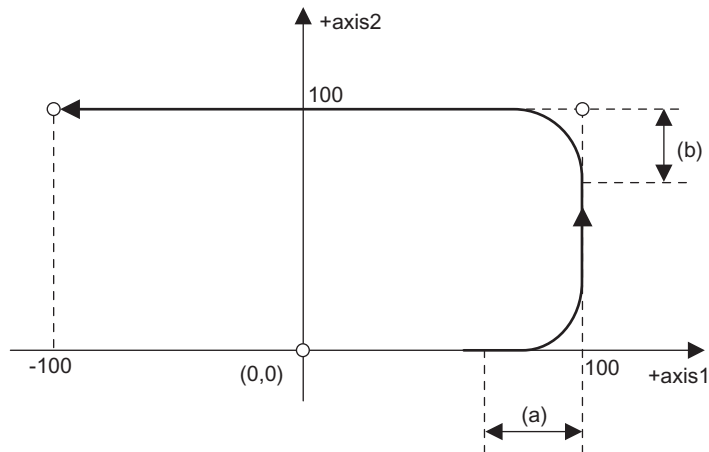


图 3.5 “第2入位检查”的程序举例

3.1.3 单块忽视 (SNG)

(1) 概要

“单块忽视 (SNG)”是在“单块运行模式¹”下运行时，仅想使某特定程序块连续运行时使用。

(2) 详细说明

SGN 命令的指令方法如下所示。

SNG: 进行轴移动的命令
或
SNG: 不进行轴移动的命令

根据上述指令，在表述 (SNG) 命令的块中，即使是“单块运行模式”，也不进行“单块停止”而进行连续运行。

SNG 命令可指令的命令如下所示。

MOV、MVS、MCW、MCC、ZRN、SKP、MVT、
EXM、ACC、SCC、PFN、VEL、INP、EOX

上述命令以外的命令，由于在 1 个扫描周期内结束，所以不能指定 SNG 命令。



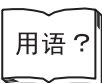
轴移动命令及速度与加速度命令以外的命令不能进行单块运行。因此，对于这些命令使用 SNG 命令是无意义的。

(3) 程序举例



```
MVS [axis1]0 [axis2]0;
SNG MVS [axis1]100 [axis2]200;
MB000101 = 1;
MB000102 = 1;
MB000103 = 1;
```

即使是“单块运行模式”，也可进行连续运行。



¹ 单块运行模式

在单块运行模式下，可以单独进行单块停止。但是，轴移动命令及速度、加速度命令以外的命令不能进行单独停止。另外，根据运动程序的内容不同，在并列执行中的单块运行会发生单块停止的变化，敬请注意。

3.1.4 用户函数调用 (UFC)

(1) 概要

“用户函数调用 (UFC)” 命令通过指定函数名，调用用户制作的函数并执行。

(2) 详细说明

UFC 命令的指令方法如下所示。

UFC 函数名 输入数据, 输入地址, 输出数据*

函数名 : ASCII 18 字节

输入数据 : 最多 16 个数据 (必须有 1 个数据)

输入地址 : 最多 1 个地址

输出数据 : 最多 16 个数据 (必须有 1 个数据)

* 可省略输入地址。[输入数据, 输出数据] 表示无输入地址。输入数据和输出数据最少均须 1 个。

根据上述指令调用用户函数。当用户函数的执行结束后，进入 “UFC” 下面的块。

(3) 程序举例



```
UFC_KANSUU MB0000 IW0010 MB00020, MA00100,
  函数名称      输入数据      输入地址
MB00001 MW00200 ML00201;
      输出数据
```

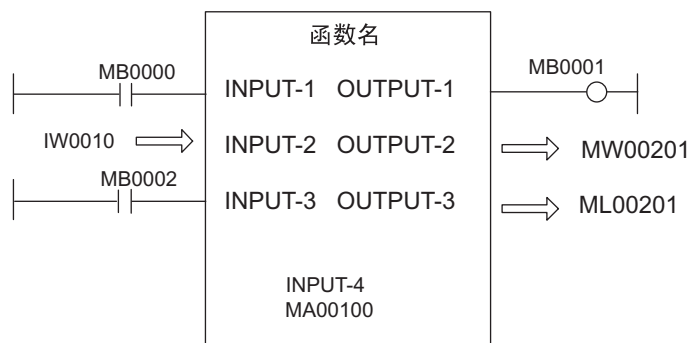


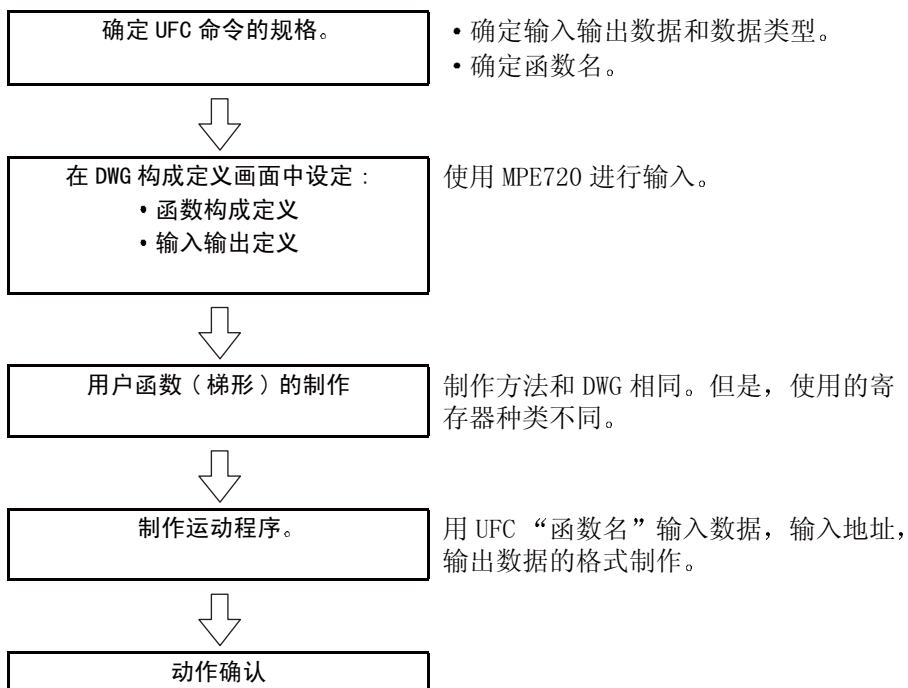
图 3.6 用户函数调用 (UFC) 的程序举例

重要

在制作用户函数时，用户函数执行结束后请将 YB00 的标志设为 ON。在运动程序中 YB00 为 ON 时，可认为用户函数已结束，进入下面的块。YB00 不为 ON 时，运动程序有可能不能进入下面的块。

(4) UFC 命令的制作步骤

UFC 命令的制作步骤如下所示。



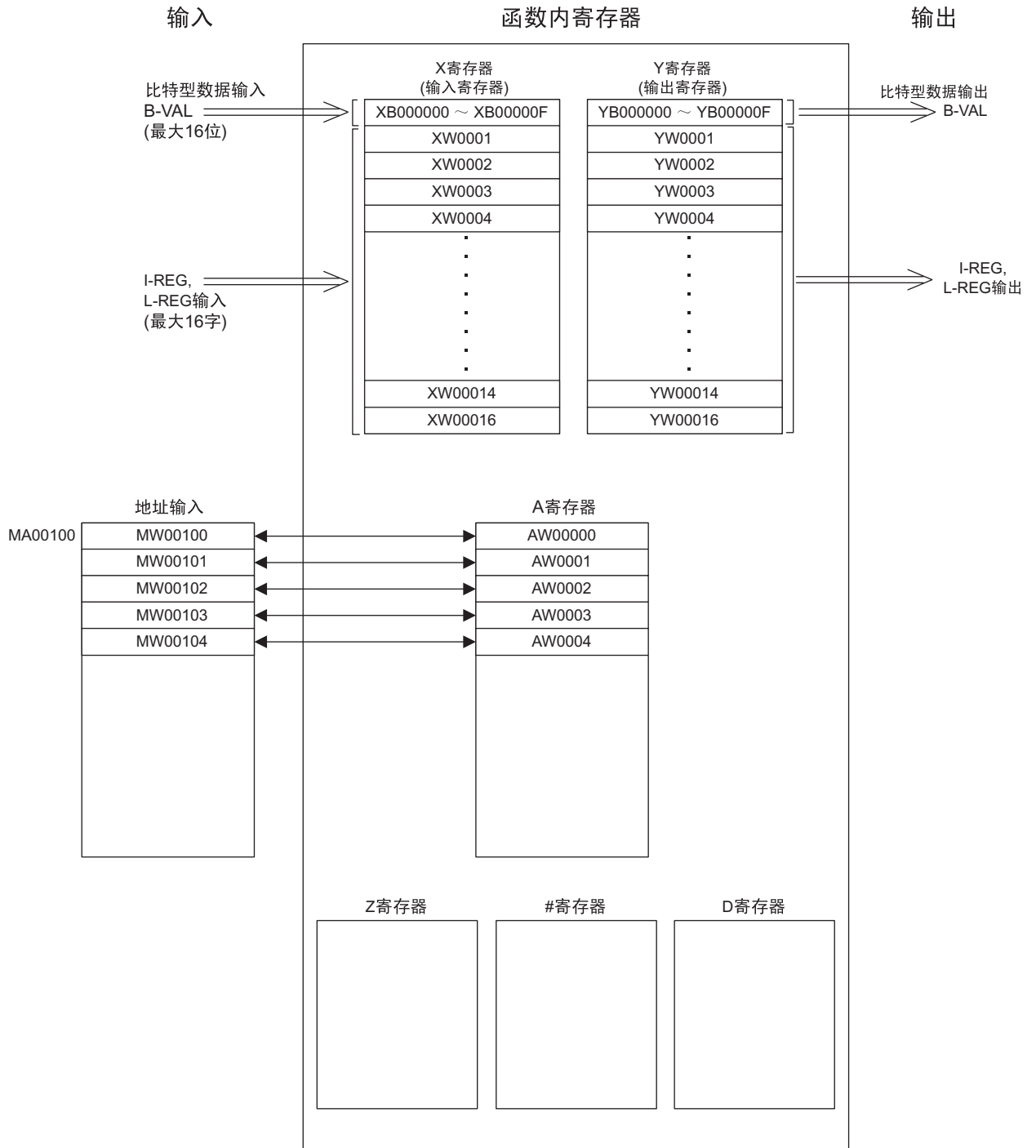
(5) 在用户函数中使用的寄存器类型

数据类型的分类如下所示。

- B-VAL ⇒ 比特型
- I-VAL ⇒ 将来用
- L-VAL ⇒
- I-REG ⇒ 整型
- L-REG ⇒ 长整型或者实型

(6) 输入输出寄存器和函数内寄存器的关系

用 UFC 命令指定的输入输出寄存器和函数寄存器的对应关系如下所示。



S、M、I、O、C 寄存器可以和 DWG 寄存器一样使用。

各函数可以使用下表所示的 11 种寄存器。

表 3.2 函数寄存器

类别	名称	指定方法	内容	特性
X	函数输入寄存器	XB, XW, XL, XFnnnnn	向函数输入 位输入 :XB000000 ~ XB0000F 整型输入 :XW00001 ~ XW00016 长整型输入 :XL00001 ~ XL00015 寄存器编号 nnnnn 用 10 进制表述。	函数单独
Y	函数输出寄存器	YB, YW, YL, YFnnnnn	从函数输出 位输出 :YB000000 ~ YB0000F 整型输出 :YW00001 ~ YW00016 长整型输出 :YL00001 ~ YL00015 寄存器编号 nnnnn 用 10 进制表述。	
Z	函数内部寄存器	ZB, ZW, ZL, ZFnnnnn	各函数原有的内部寄存器。 可作为内部处理用于函数。 寄存器编号 nnnnn 用 10 进制表述。	
A	函数外部寄存器	AB, AW, AL, AFnnnnn	把地址输入值作为基本地址的外部寄存器。 用于与 (S、M、I、O、#、Dannnnn) 的链接。 寄存器编号 nnnnn 用 10 进制表述。	
#	# 寄存器	#B, #W, #L, #Fnnnnn (#Annnnn)	仅可进行程序调用的寄存器。 仅相应 DWG 可调用。 用户用 MPE720 指定实际利用范围。 寄存器编号 nnnnn 用 10 进制表述。	
D	D 寄存器	DB, DW, DL, DFnnnnn (DAnnnnn)	各 DWG 原有的寄存器。 仅相应 DWG 可调用。 用户用 MPE720 指定实际利用范围。 寄存器编号 nnnnn 用 10 进制表述。	
S	系统寄存器	SB, SW, SL, SFnnnnn (SAnnnnn)	与 DWG 寄存器相同 由于该寄存器为 DWG / 函数通用, 不同优先级别的 DWG 调用相同的函数时, 请注意使用方法。	DWG 通用
M	数据寄存器	MB, MW, ML, MFnnnnn (MAnnnnn)		
I	输入寄存器	IB, IW, IL, IFhhhh (IAhhhh)		
O	输出寄存器	OB, OW, OL, Ofhhhh (OAhhhh)		
C	参数寄存器	CB, CW, CL, CFhhhhh (CAnnnnn)		

(注)在函数内部也可使用 SA, MA, IA, OA, DA, #A, CA。

输入输出寄存器的数据交接举例如下。

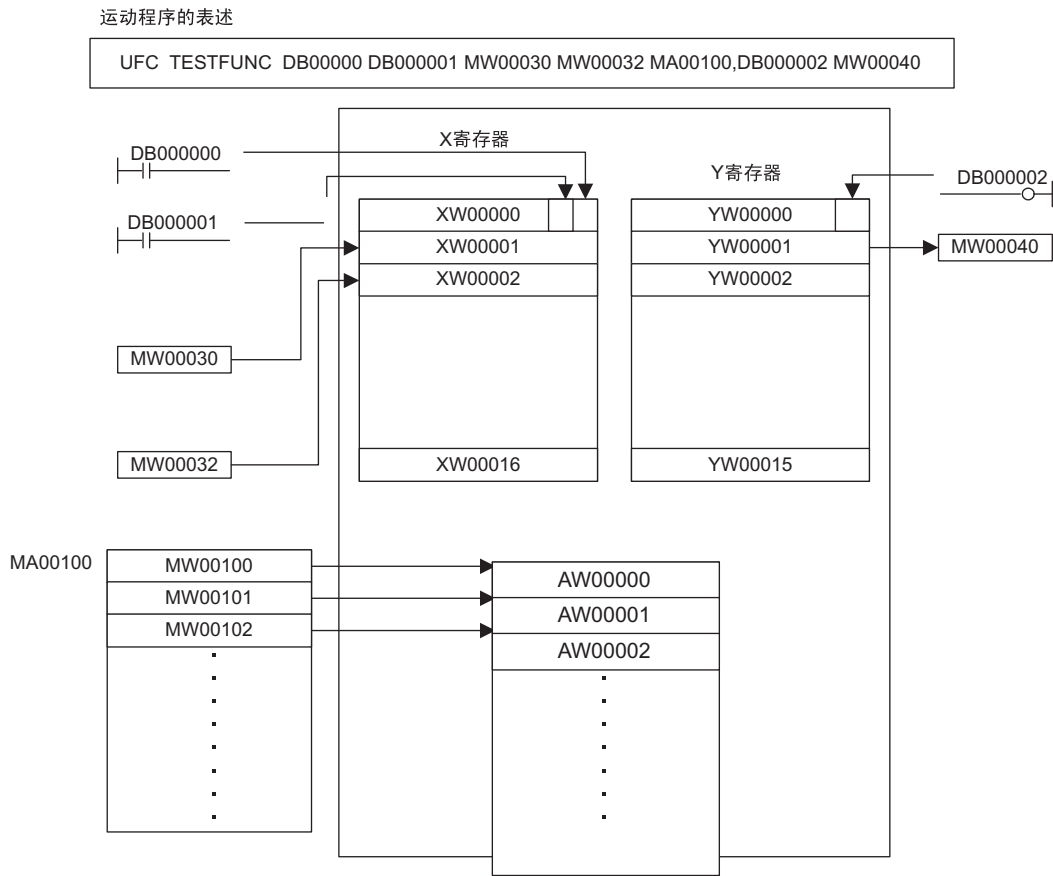


图 3.7 运动程序的表述

(7) 用户函数的制作

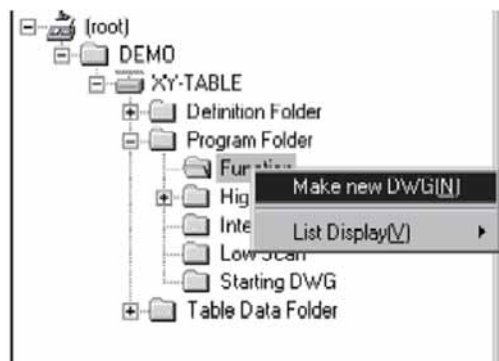
以下规格的函数为例，列出用户函数的制作步骤。

但，此处为 MP930 的示例，MP2000 系列中的参数地址是不同的。

规格	指定伺服轴 No. 和速度数据，向设定参数“快进给速度 0Lxx22”设定。
运动程序	MW00030 = 伺服轴 No. (1or2) ML00032 = 快进给速度 UFC FUNC-T1 MW00030 ML00032, DB000001;

用户函数制作步骤如下所示。

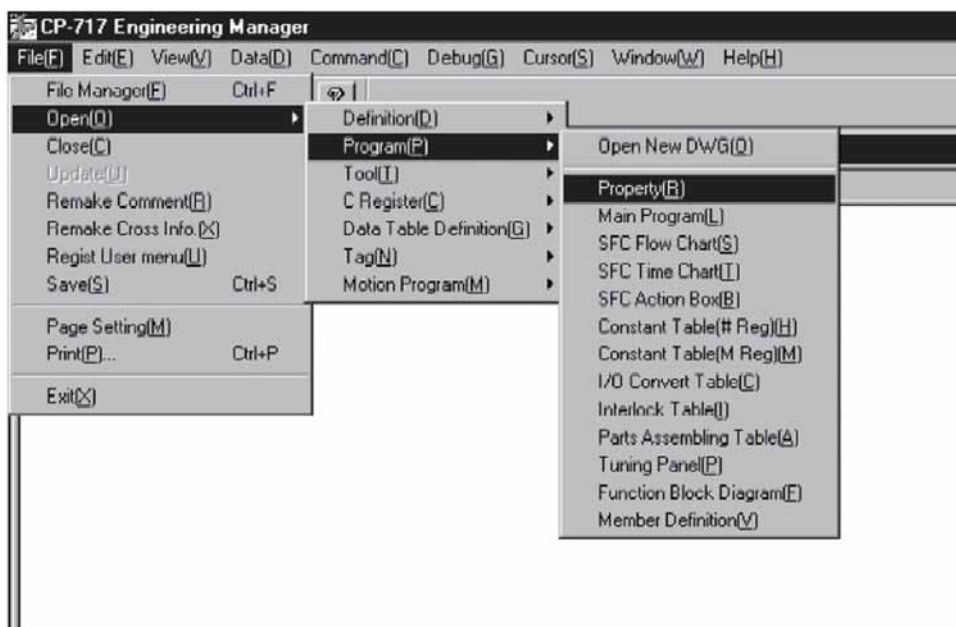
1. 在文件管理器画面中，按“Program Folder → Function”的顺序展开，把光标放在 Function 处右击，然后双击“Make new DWG [N]”。



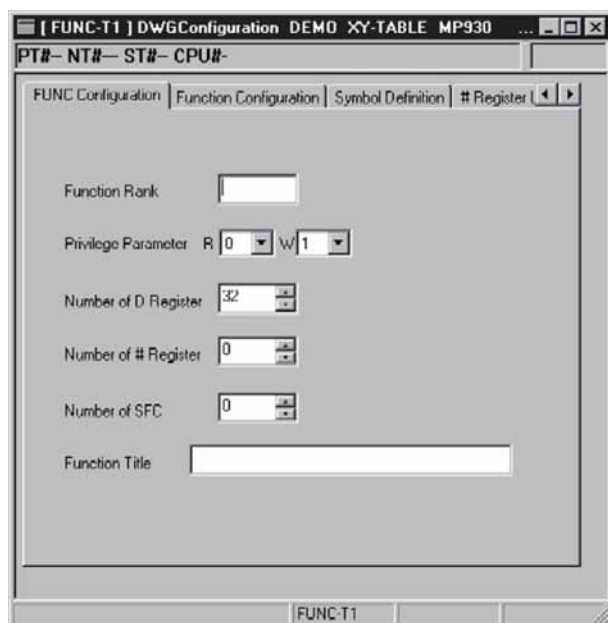
2. 请在信息框 “Input DWG Name” 中，设定 DWG Name、DWG Type，点击 “OK” 按钮。



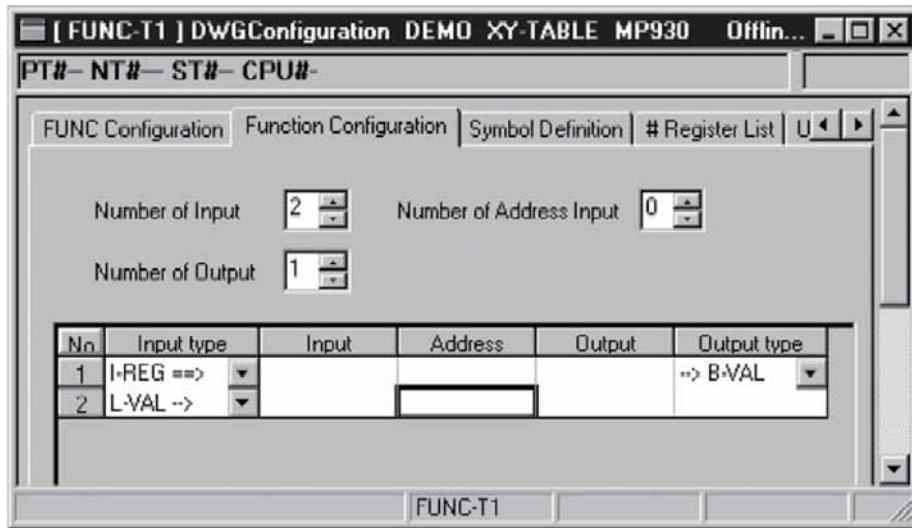
3. 显示梯形画面（无程序时为白色画面）。请从菜单的 File (F) 中按序选择 “Open (O) → Program (P) → Property (R)”，打开 DWG 构成定义画面。



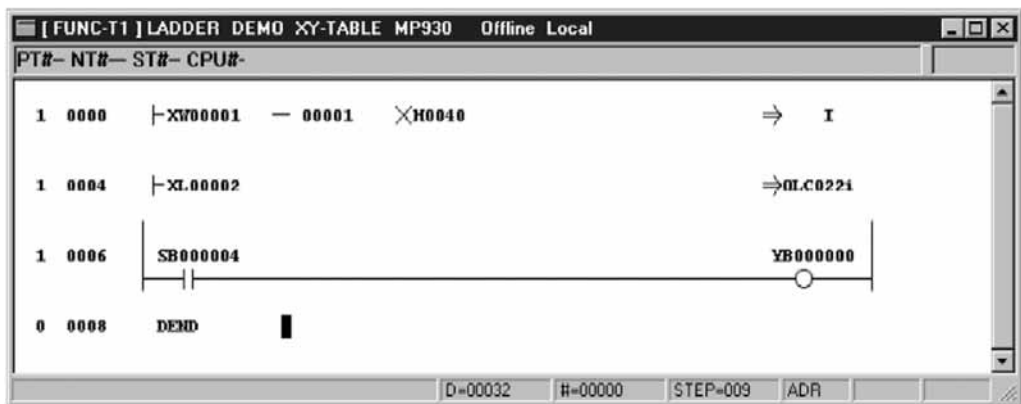
4. 请在 DWG Configuration 的 FUNC Configuration 中，设定 Number of D Register。（默认值为 32 个。）



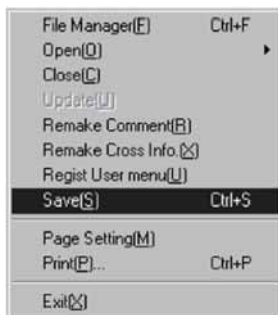
5. 请点击 Function Configuration 标记，设定函数的输入输出个数和数据类型。
 (例) UFC FUNC-T1 MW00030 ML00032,,DB000001;
 时，设定如下所示。



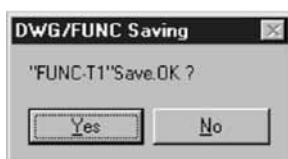
6. 请关闭 DWG Configuration 画面，在 LADDER 画面中编辑用户函数程序。



7. 请在 LADDER 菜单中，选择 “File(F)” → “Save(S)”。



8. 显示 DWG/FUNC Saving 对话框。请点击 “Yes”。



9. 显示 Saving has been completed 对话框。请点击“OK”按钮。



10. 至此，从运动程序中调用的用户函数的制作已结束。

11. 请在运动编辑器画面中，制作调用用户函数的程序。

```

MPM001 "";
  fm× t8000000;           "setup interpolation max. feed speed
  mw0001=10000;          "interpolation speed override=100.0%
  iac t300;              "time constant acceleration for interpolation
  idc t500;              "time constant acceleration for interpolation
  vel [x]6000 [y]5000;   "setup axis x,y feed speed
  zrn [x]0 [y]0;        "home position return
  mw30=1;                "servo No.=1
  ml32=500;              "feed speed=500
  ufc func=t1_mw30_ml32_db01; "user function
  mov [x]100.0 [y]100.0; "positioning (100,100)
  inc mvs [x]200.0 f5000000; "linear interpolation axis x +200.0(inc. mode)
  mvs [y]200.0;          "axis-y +200.0 (inc. mode)
  mvs [x]-200.0;         "axis-x -200.0 (inc. mode)
  mvs [y]-200.0;         "axis-y -200.0 (inc. mode)
  abs mvs [x]200.0 [y]200.0; "axis-x,y[200,200] (abs. mode)
  pln [x] [y];          "circular interpolation plane
  mcc [x]200.0 [y]200.0 u200.0 v300.0 f5000000; "circular interpolation
  mvs [x]100.0 [y]100.0;
  end;

```

12. 请执行运动程序，进行动作的确认。

3.1.5 输入输出变量待机 (IOW)

(1) 概要

“输入输出变量待机 (IOW)” 命令到变为条件表达式指定状态为止一直 WAIT (待机)，如果条件成立则进入下面的块。

(2) 详细说明

IOW 命令的指令方法如下所示。

<pre>IOW IB000001&IB000002==1; A B A: 条件表达式 B: 状态</pre>

通过上述指令，到条件表达式成立为止 WAIT (待机)，如果条件成立，结束本块的运行，进入下面的块。



- 条件表达式只能放在左侧。
- 比特型时，只能使用比较命令 “==”。
以下情况时不能使用。
IOW IB000001&IB000002 < > 0;
- 在条件表达式中也可使用 ()。
- 比特型以外的条件表达式可以使用所有的比较指令。
IOW MW00100+MW00101 > 1000;
IOW ML00100*3/100 >= 20;
IOW MW00100&MW00101 ∨ MW00102 == 3355H;

(3) 程序举例



```
IOW MB001001&MB001002 == 1;
MOV [axis1] 1000;
```

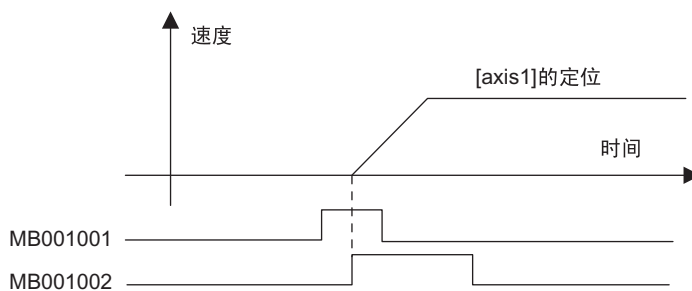


图 3.8 输入输出变量待机 (IOW) 的程序举例

3.1.6 子程序调用 (MSEE)

(1) 概要

“子程序调用 (MSEE)” 可从运动程序中调用事先在运动程序内存内储存的 “子程序” 并执行。

(2) 详细说明

MSEE 命令的指令方法如下所示。

```
MSEE MPS 一;
      调用 “子程序” 编号
```

通过上述指令，执行用 “MPS” 指定编号的 “子程序”。
可以从 “运动程序” 中调用 “子程序”，没有次数限制。

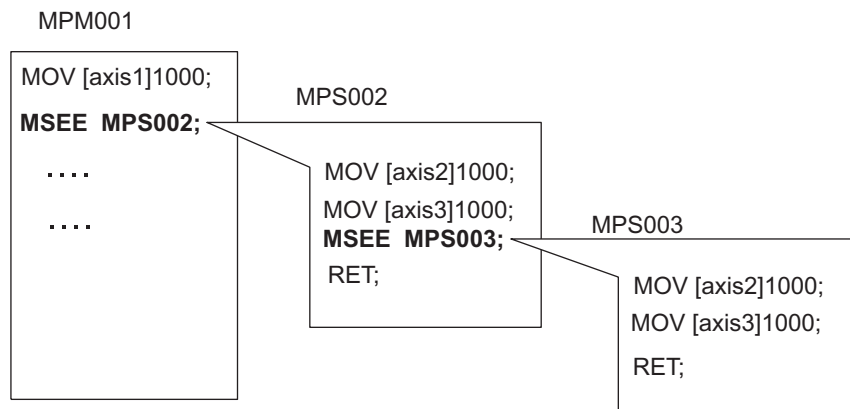


图 3.9 “子程序”

在 “子程序” 的结尾，必须指令 “子程序结束 (RET)” 命令。

重要

■ 子程序的制约条件

在子程序内部对运动程序的表述有以下限制，敬请注意。

- PFORK 命令的最多只能并列 2 个。
- 在子程序内使用 PFORK 命令时，轴移动命令只能在单侧表述。
- 主程序 MPM□□□ 和子程序 MPS□□□ 可以使用 1 ~ 256 编号，但 MPM 和 MPS 不能使用相同的编号。
- 用 MSEE 命令调用主程序的程序编号时，则不能被执行。

3.1.7 子程序结束 (RET)

(1) 概要

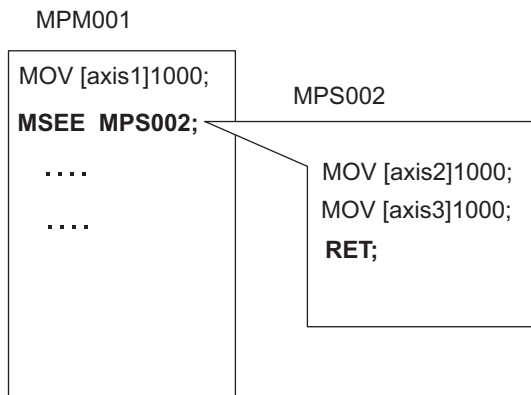
“子程序结束 (RET)” 命令是在 “子程序” 最后进行指令。

(2) 详细说明

RET 命令的指令方法如下所示。

RET; 子程序结束

通过上述指令，进入调用该子程序的程序（主程序或子程序）的“子程序调出 (MSEE)” 命令的下面的块。



3.1.8 1个扫描周期的WAIT(待机)命令(EOX)

(1) 概要

按1个扫描周期执行连续的运动程序顺序命令。在按几个扫描周期执行连续的顺序命令时，使用EOX命令。

另外，该命令也适应于块运行。（是在块运行模式下能停止的命令。）

(2) 指令方法

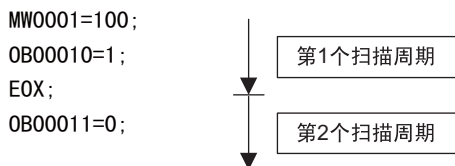
```
MW0001=100;
OB00010=1;
EOX;
OB00011=0;
```

在下一扫描周期中，执行EOX之后的块。

(3) 程序举例



(a) 上述示例时

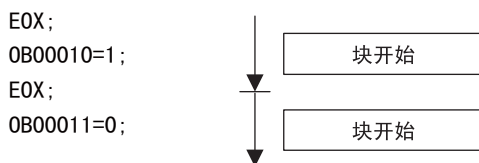


(b) WHILE 命令的使用举例

```
WHILE OB00010==1;
EOX;
WEND;
```

仅在顺序命令中，可用WHILE语句编程。

(c) 顺序命令的调试举例



■ 块运行

以块为单位使运动程序动作，确认程序是否正常的功能即“块运行”功能。通过块运行模式和块运行开始信号，能进行块单位的动作。

但是，在块运行中，有能停止命令和不能停止命令两种。

- 能停止命令：按1个扫描周期处理不完的命令（MOV、MVS等的轴移动命令 含EOX）
- 不能停止命令：按1个扫描可处理完的命令（运算命令等的顺序命令 含PLD）

3.1.9 分支命令 (IF ELSE IEND)

(1) 概要

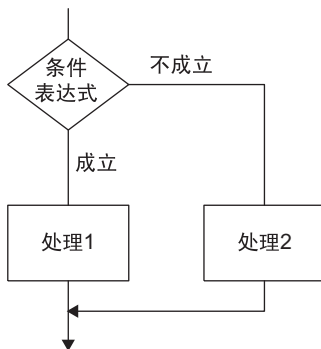
当满足条件表达式时，分支命令 (IF ELSE IEND) 执行 “IF ~ ELSE” 间的块，当不满足条件式时，执行 “ELSE ~ IEND” 间的块。

(2) 详细说明

分支命令 (IF ELSE IEND) 的指令方法如下所示。

```
IF (条件表达式);
  ... (处理 1)
ELSE;
  ... (处理 2)
IEND;
```

通过上述指令，满足条件表达式时，执行 (处理 1)；不满足条件表达式时，执行 (处理 2)。可省略 [ELSE]。此时，当不满足条件表达式时，从 “IEND” 下面的块开始继续执行。



重要

分支命令 (IF ELSE IEND) 的嵌套数的限制为最多八重。

(3) 程序举例



```
IF MB == 1;
MOV [axis1] 10000; ← MB 一为 ON 时，进行 axis1 的定位。
ELSE;
MOV [axis2] 10000; ← MB 一为 OFF 时，进行 axis2 的定位。
IEND;
```

3.1.10 循环命令 (WHILE WEND)

(1) 概要

循环命令 (WHILE WEND) 是在满足条件表达式期间，循环执行指令范围的块的功能。

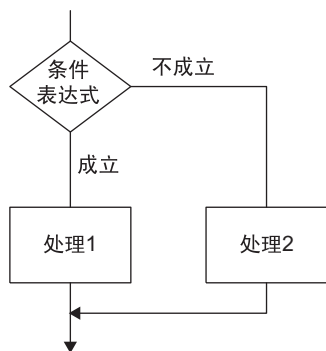
(2) 详细说明

循环命令 (WHILE WEND) 的指令方法举例如下所示。

```

WHILE (条件表达式) ;
  ...;
  ...; (处理)
  ...;
WEND ; ← 循环命令的结束
  
```

通过上述指令，在<条件表达式>成立期间，循环执行“WHILE ~ WEND”的块。当<条件表达式>不成立时，将跳到“WEND”下面的块。



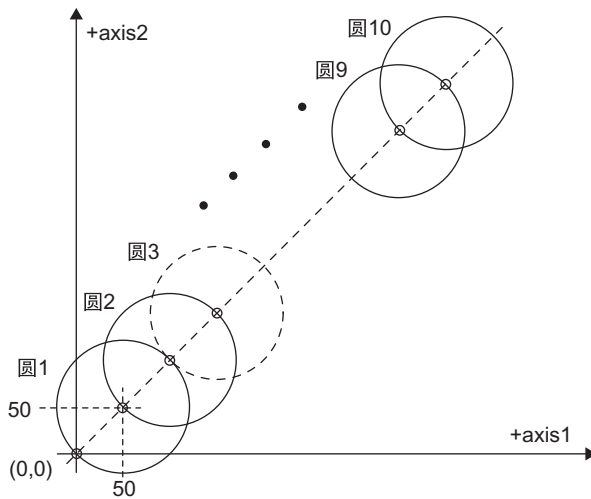
重要

- 循环命令 (WHILE WEND) 的嵌套数限制为最多八重。
- 当仅按 1 个扫描周期执行命令制作循环处理的程序时，为无限循环程序，成为监视装置错误（系统错误）。仅按 1 个扫描周期执行命令时，请务必加入 EOX(1 个扫描周期的待机命令)。
- 不在 1 个扫描周期内结束的命令
不在 1 个扫描周期内结束的命令如下所示。
MOV、MVS、MCW/MCC、ZRN、SKP、MVT、EXM、ACC、SCC、PFN、VEL、INP、TIM、IOW
- 在 1 个扫描周期内结束的命令
除上述命令以外的命令全部为在 1 个扫描周期内结束的命令。
ABS、INC、IFP、PLN、IAC、IDC、FMX、POS、PLD、MSEE、END、RET、IF、WHILE、PFORK、SFORK，所有顺序命令

(3) 程序举例

例 用下面的程序，画 10 个半径为 50 的圆。

MOV [axis1]0 [axis2]0;	← 定位
MW00001=10000;	← 插补用额定速度比 (100%)*
MW00100 = 1;	← 计数器预设
INC;	← 增量值模式指定
PLN [axis1] [axis2];	← 坐标平面指定
WHILE MW00100 <= 10 ;	← 循环命令
MCW [axis1]0 [axis2]0 U50. V50. F8000 ;	← 圆弧插补
MOV [axis1]50. [axis2]50. ;	← 定位
MW00100 = MW00100 + 1;	← 计数器增量值
WEND ;	← 循环命令的结束



* 为 MP930 的示例。机型不同，插补额定速度比的设定也不同，详情请参阅“1.2.3 进给速度”的插补进给速度。

3.1.11 并行执行命令 (PFORK, JOINTO, PJOINT)

(1) 概要

并行执行命令 (PFORK) 可以并行执行指定标签的块。在各处理并行执行后, 用 JOINTO 命令汇合到指定的标签。并行处理最多可指定到 4。

有关标签请参阅“1.2.1 输入格式标签”。

(2) 详细说明

并行执行命令 (PFORK) 的指定方法如下所示。

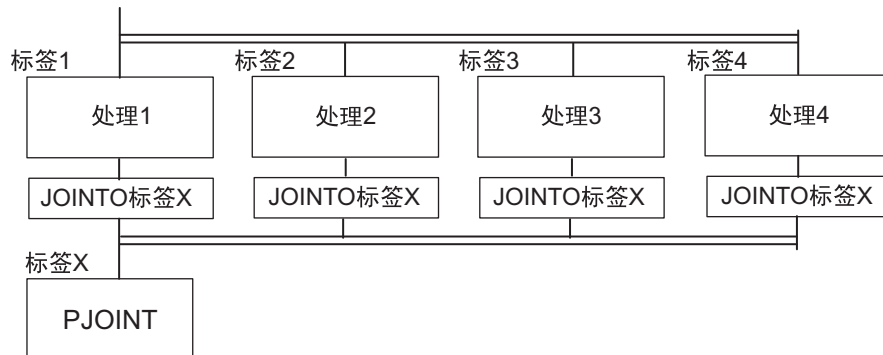
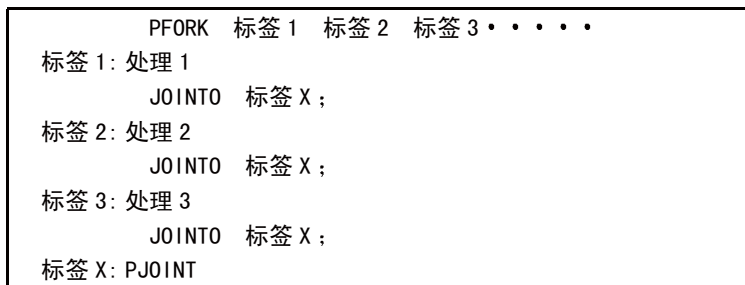


图 3.10 并行执行命令 (PFORK, JOINTO, PJOINT) 的指令方法

通过上述指令, 用 PFORK 命令并行执行指定标签的块 (处理 1, 处理 2, 处理 3, …)。在各处理并行执行后, 用 JOINTO 命令汇合到指定的标签。并行处理最多可指定到 4。

通过该命令, 可以任意指定轴移动命令和顺序的并行执行或者轴移动命令和轴移动命令的并行执行。

(a) PFORK 命令前指定的命令

PFORK 命令前指定的命令 :FMX、ABS/INC、F 指令、IFP、PLN、IAC/IDC…的值在并行执行命令下被保留在并行执行处理中。另外, 可以分别对每个并行指令这些命令。汇合后保留最左侧处理的值。

(b) 在子程序内的并列执行命令

在子程序内的并列执行命令有以下限制。

- 在子程序内的并列处理最多为 2。
- 在 MP900 系列中，轴移动命令只能在用首标签指定的程序块内表述。

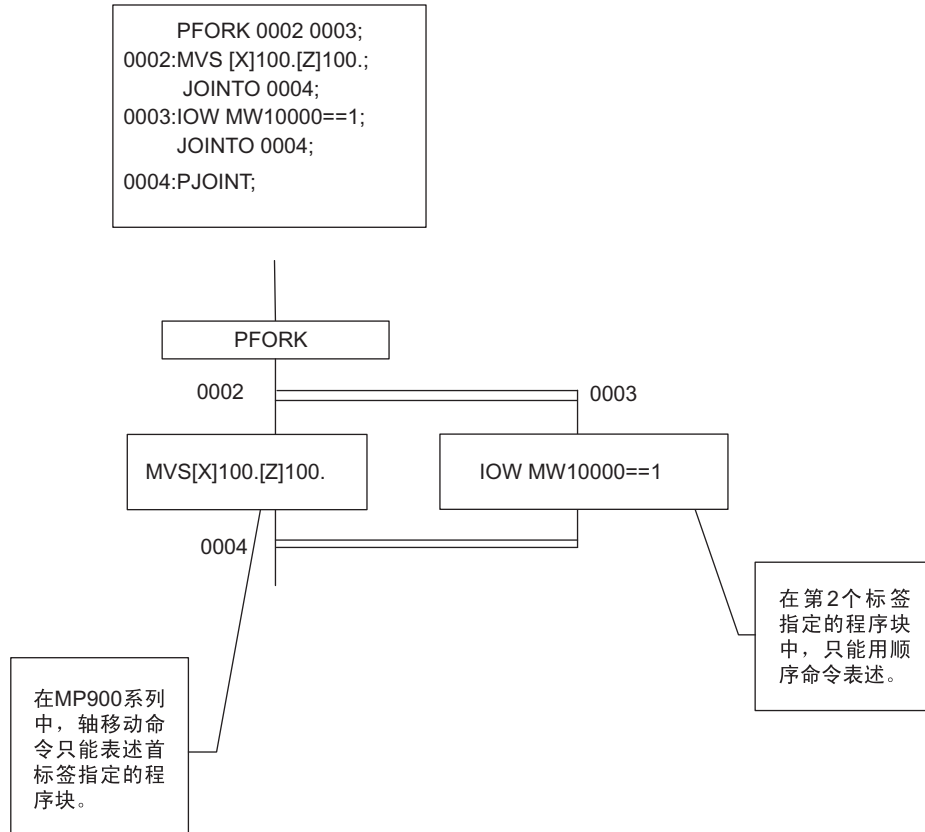


图 3.11 子程序内的并列执行命令

重要

- 在程序内有多个相同的标签时，将会出错（标签被重复定义）。
- PFORK 的分支数和标签数不同时，将会出错。

(3) 程序举例

例 (a) 基本模式

基本模式的程序举例如下所示。

```

0001:MOV [X]100. [Y]150.;
      MVS [X]200. [Y]250. F1000;
      PFORK 0002 0003 0004;
0002:MVS [X]300. [Z]100.
      JOINTO 0005;
0003:MW12345=MW10000+MW10002;
      IOW MB120001==1;
      JOINTO 0005;
0004:MVS [Z]100. [S]100. F3000;
      JOINTO 0005;
0005:PJOINT;
      MOV [X]500. [Y]500. [Z]500.;
      .
      .

```

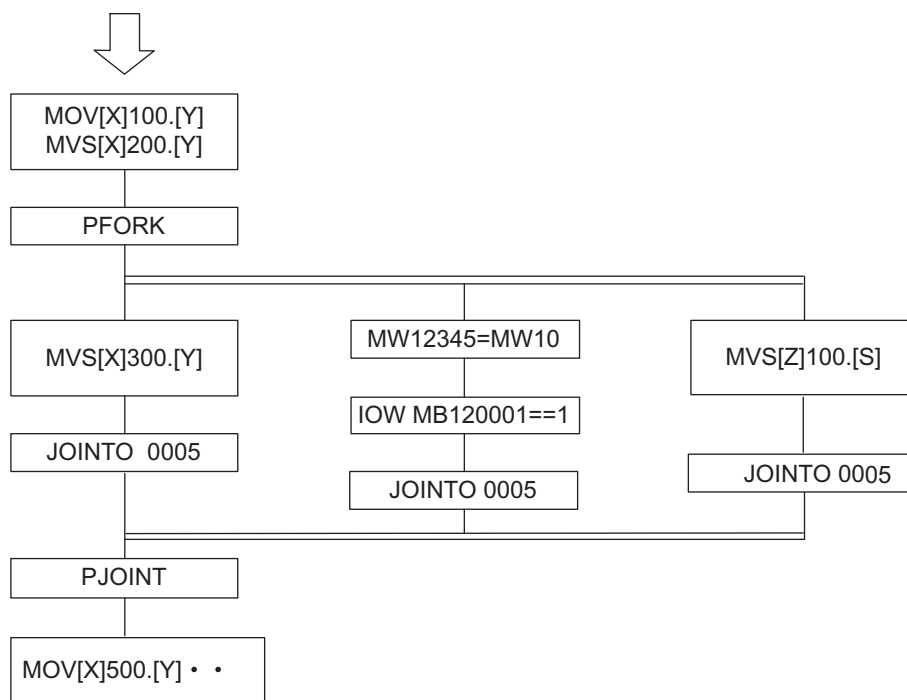
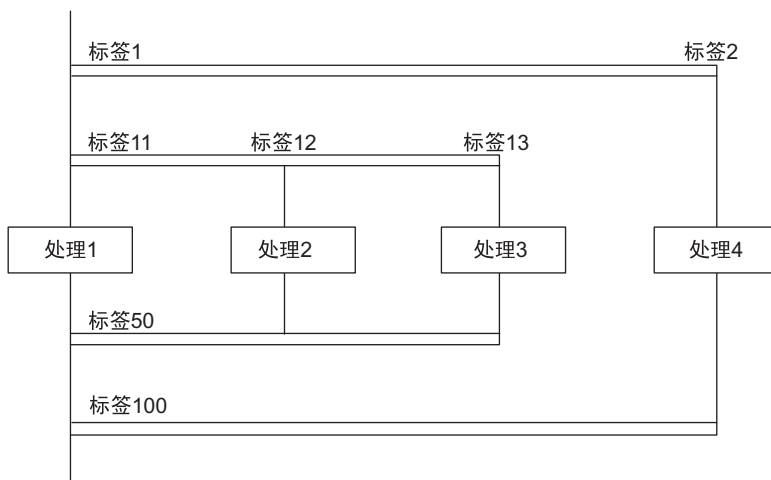


图 3.12 并列执行命令 (PFORK, JOINTO, PJOINT) 的程序举例

(b) 应用模式 1



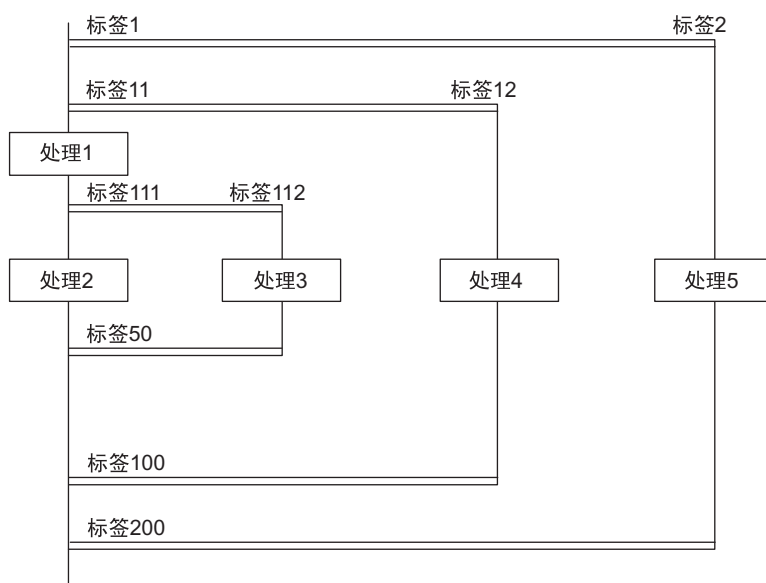
可进行以下的并列执行。



(c) 应用模式 2

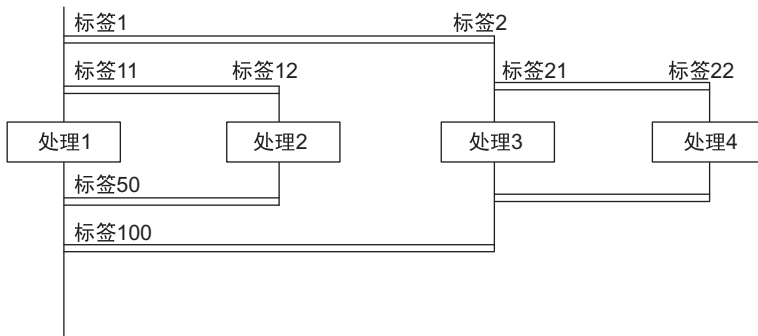


可进行以下的并列执行。



重要

不能进行以下的并列执行。



3.1.12 选择执行命令 (SFORK, JOINTO, SJOINT)

(1) 概要

满足指定的条件表达式时，执行接在“?”后的标签的块。执行完各处理后，用 JOINTO 命令汇合到指定标签的块中。条件表达式的指定个数不限。

(2) 详细说明

选择执行命令 (SFORK) 的指令方法举例如下所示。

```

SFORK 条件表达式 1? 标签 1, 条件表达式 2? 标签 2, 条件表达式 3? 标签 3,
      条件式 4? 标签 4 ;
标签 1: 处理 1
        JOINTO 标签 X
标签 2: 处理 2
        JOINTO 标签 X
标签 3: 处理 3
        JOINTO 标签 X
标签 4: 处理 4
        JOINTO 标签 X
      .
      .
      .
      标签 X: SJOINT
  
```

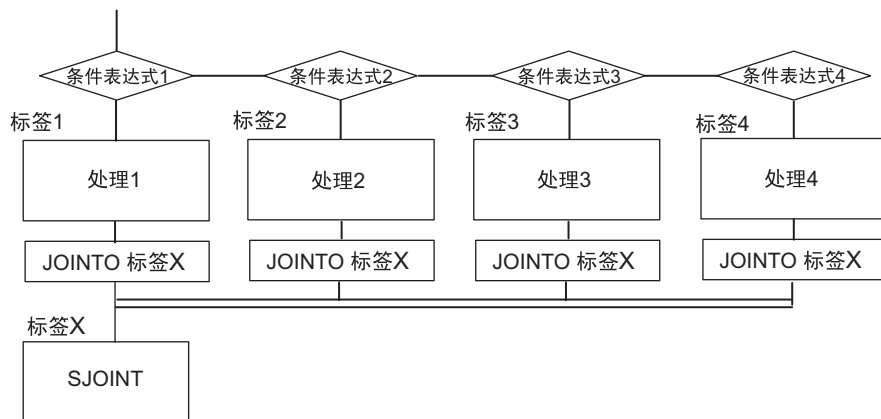


图 3.13 选择执行命令 (SFORK, JOINTO, SJOINT) 的指令方法

通过上述指令，满足用 SFORK 命令指定的条件表达式时，执行带标签的块。执行完各处理后，请用 JOINTO 命令汇合到指定的标签处。

补充

- 从首个“条件表达式 1”开始按顺序判断条件表达式。因此，即使有多个条件表达式时，也从最初条件表达式成立的标签开始执行处理。
- 请表述条件表达式必须成立的条件。条件不成立时，SFORK 命令的块中为待机状态，直到条件成立为止。

(3) 程序举例



```

0001:MOV [X]100.[Y]150. :
      MVS [X]200.[Y]250.F1000:
      SFORK MW00100==1 ? 0002,MW00100==2 ? 0003,MW00100==3 ? 0004;
0002:MVS [X]300.[Z]100.F3000:
      JOINTO 0005
0003:MVS [X]300.[Z]100.F3000:
      JOINTO 0005
0004:MVS [Z]300.[S]100.F3000:
      JOINTO 0005
0005:SJOINT;
      MOV
[X]500.[Y]500.[Z]500.

```

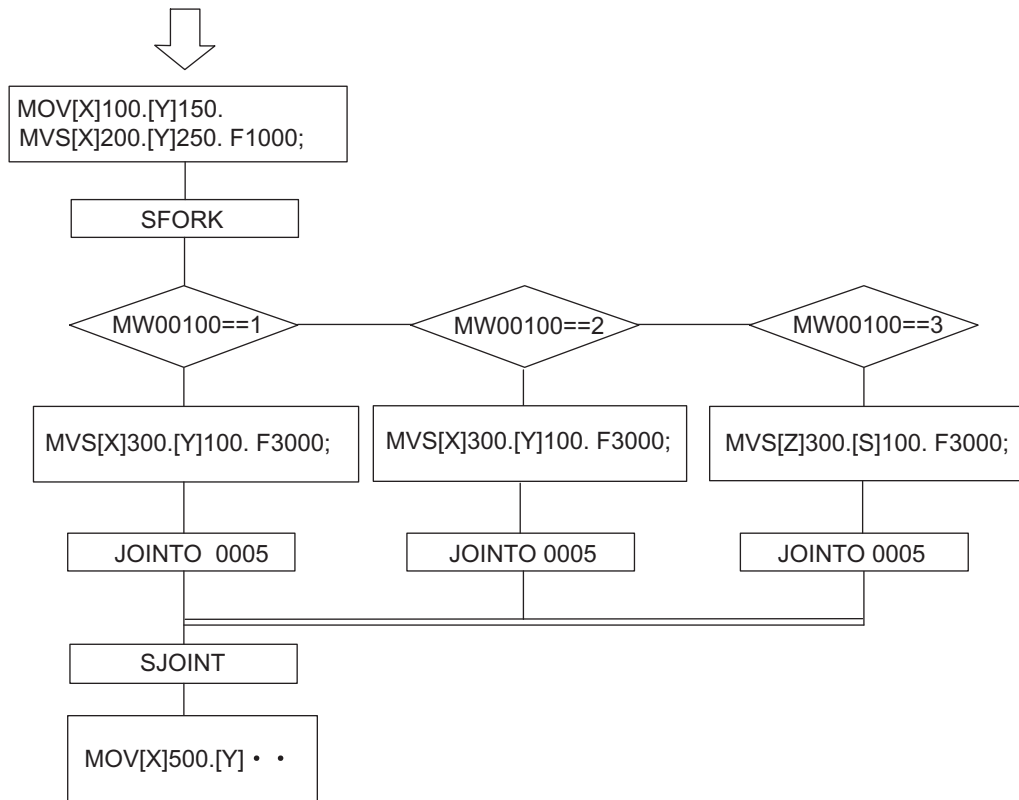


图 3.14 选择执行命令 (SFORK, JOINTO, SJOINT) 的程序举例

3.2 速度与加减速命令

本节对设定轴移动的速度与加减速速度命令的编程方法进行详细说明。

3.2.1 加速时间变更 (ACC)

(1) 概要

“加速时间变更 (ACC)”命令可以变更定位类命令 (MOV, MVT, EXM) 各轴的加减速时间。

(2) 详细说明

ACC 命令的指令方法如下所示。

ACC [axis1] - [axis2] - ...; 加速时间

通过上述指令，变更各轴的加速时间。不会变更未指定的轴的加速时间。变更 MOV (定位) 命令的加速时间。

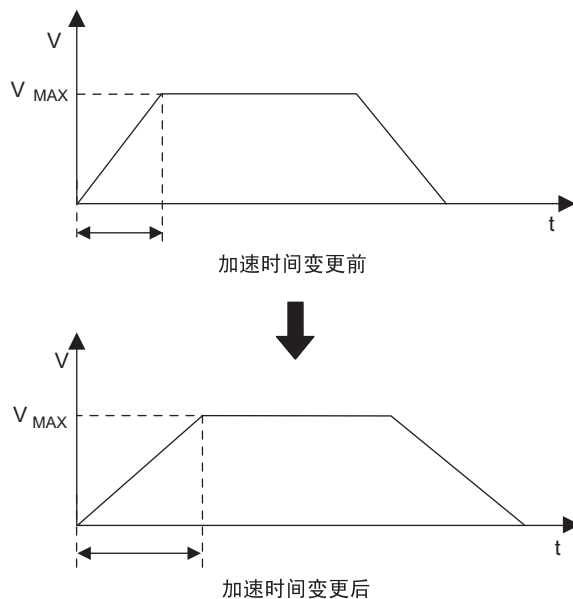


图 3.15 加速时间变更

通过执行“ACC”命令变更的加速时间直到用下一个“ACC”命令再设定为止一直有效。

“ACC”命令的指令范围如下。

1 ~ 32767 [msec]

(注)MP900 系列为“msec”；MP2000 系列则按照由“加减速速度单位选择 (0Wxx03 Bit4 ~ 7)”设定的单位。

补充

- 通过“ACC”命令，各轴设定参数的“直线加速时间设定 (0Wxx0C / 0Wxx36)”被变更。
- MP920 (SVA-01, 02) 模块时，可以取代“ACC”命令与“0Wxx0C=xxxx”进行编程。
- MP930 时，用“VEL”命令可以设定到达设定进给速度为止的时间。MP930 以外的机器控制器可以设定到达额定速度为止的时间。
- 使用 SGD-□□□N、SGDB-□□□AN 时，为加减速时间。

(3) 程序举例



```

INC;
VEL [axis1]6000;    进给速度设定
ACC [axis1]1000;    加速时间 1 秒
MOV [axis1]100000;  定位
ACC [axis1]500;     加速时间 0.5 秒
MOV [axis1]50000;

```

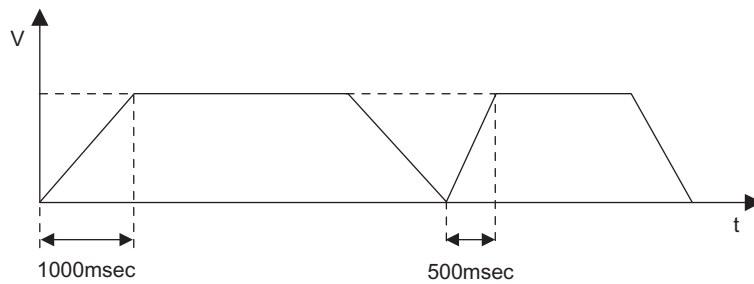


图 3.16 “加速时间变更”程序举例

3.2.2 减速时间变更 (DCC)

(1) 概要

“减速时间变更 (DCC)” 命令可以变更定位类命令 (MOV, MVT, EXM) 各轴的减速时间。

(2) 详细说明

DCC 命令的指令方法如下所示。

```
DCC [axis1] - [axis2] - ...;
      减速时间
```

通过上述指令, 变更各轴的减速时间。不会变更未指定的轴的减速时间。变更 MOV (定位) 命令的减速时间。

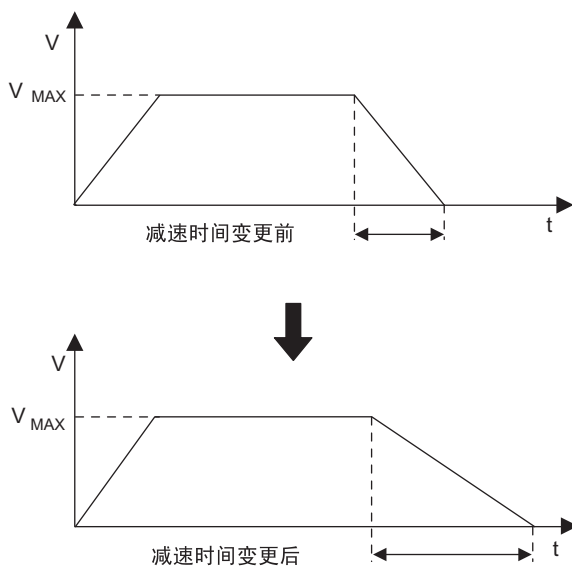


图 3.17 减速时间变更

通过执行“DCC”命令变更的减速时间直到用下一个“DCC”命令再设定为止一直有效。

“DCC”命令的指令范围如下。

1 ~ 32767 [msec]

(注)MP900 系列为“msec”; MP2000 系列则按照由“加减速速度单位选择 (0Wxx03 Bit4 ~ 7)”设定的单位。

补充

- 通过“DCC”命令, 各轴设定参数的“直线减速时间设定 (0Wxx0D / 0Wxx38)”被变更。
- MP920 (SVA-01, 02) 模块时, 可以取代“DCC”命令与“0Wxx0C=xxxx”进行编程。
- MP930 时, 用“VEL”命令设定从设定速度到指令输出结束为止的时间。
MP930 以外的机器控制器可以设定从额定速度到指令输出结束为止的时间。
- 使用 SGD-□□□N、SGDB-□□AN 时, 不能使用“DCC”命令。

(3) 程序举例



```

INC;
VEL [axis1]6000; 进给速度设定
DCC [axis1]1000; 减速时间 1 秒
MOV [axis1]100000; 定位
DCC [axis1]500; 减速时间 0.5 秒
MOV [axis1]50000;

```

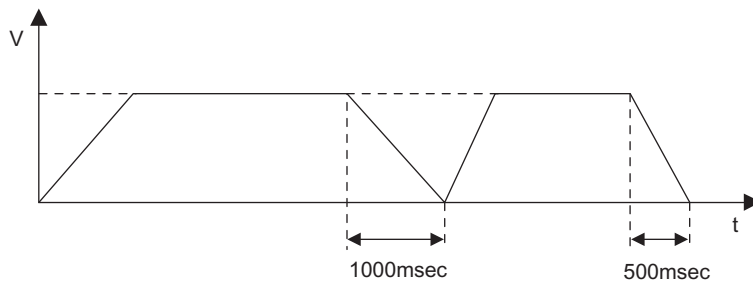


图 3.18 “减速时间变更”程序举例

3.2.3 S字时间参数变更 (SCC)

(1) 概要

“S字时间参数变更 (SCC)” 命令设定可以控制加减速时的机械性振动的 S 字加减速功能参数。变更各轴 S 字时间参数。

(2) 详细说明

SCC 命令的指令方法如下所示。

```
SCC [axis1] - [axis2] - ...;
      S字时间参数
```

通过上述指令，变更各轴的 S 字时间参数。不变更未指定的轴的 S 字时间参数。

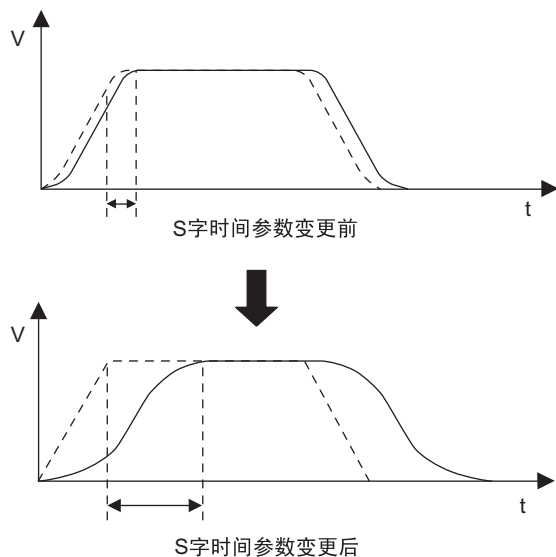


图 3.19 “S 字时间参数 (SCC)” 的变更

“SCC” 命令的指令范围如下。

0 ~ 510 [msec]



补充

- “SCC” 命令不被执行时，用设定参数 “S 字时间参数” 确定 S 字时间参数。
- “通过执行 SCC” 命令变更的 S 字时间参数直到用下一个 “SCC” 命令再设定为止一直有效。

重要

在 SVA 模块中，请不要使用该命令。

(3) 程序举例



```

INC;
SCC [axis1]250;
MOV [axis1]100000;
SCC [axis1]500;
MOV [axis1]100000;

```

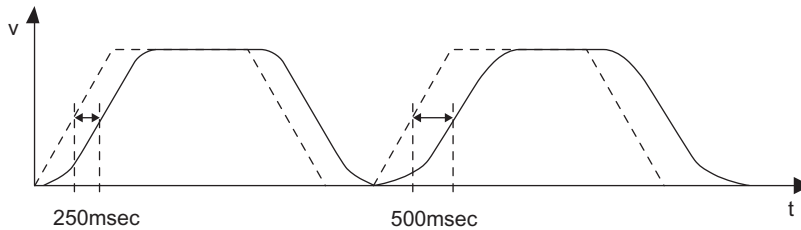


图 3.20 “S 字时间参数 (SCC)” 的程序举例

重要

■ 使 S 字加减速有效的方法。

为了使 S 字加减速有效，请设定滤波器类型。

关于滤波器类型的变更，请参考以下程序。

• MP900 系列时

```

OWC021 = OWC021 & FFOFH ;
OWC021 = OWC021 + 0020H . . . . . 滤波器类型 = 移动平均滤波器选择
IOW OWC020 == 0 . . . . . 没有运动命令?
OWC020 = 13 . . . . . 运动命令 13 = 滤波器类型变更
IOW IWC014 == 13 ; . . . . . 命令响应待机
OWC020 = 0 ; . . . . . NOP 命令指令
IOW IWC014 == 0 ;

```

• MP2000 系列时

```

OW8003 = OW8003 & F0FH ;
OW8003 = OW8003 + 0200H ; . . . . . 滤波器类型 = 移动平均滤波器选择
IOW OW8008 == 0 ; . . . . . 没有运动命令?
OW8008 = 13 ; . . . . . 运动命令 13 = 滤波器类型变更
IOW IW8008 == 13 ; . . . . . 命令响应待机
IOW IB80098 == 1 ; . . . . . 命令执行结束
OW8008 = 0 ; . . . . . NOP 命令指令
IOW IW8008 == 0 ;

```

对所有执行 S 字加减速的轴，请使用以上的程序。

3.2.4 进给速度变更 (VEL)

(1) 概要

“进给速度变更 (VEL)” 命令可以变更定位类命令 (MOV, MVT, EMX) 各轴的进给速度。

(2) 详细说明

VEL 命令的指令方法如下所示。

VEL [axis1] - [axis2] -...; 进给速度

通过上述指令，变更各轴的进给速度。不变更未指定的轴的进给速度。

“VEL” 命令的指令范围如下。

$0 \sim 2^{31} - 1 [10^n \text{ 指令单位 /min}]$

n = 小数点以下位数

(注) MP900 系列为 “ 10^n 指令单位 /min”，MP2000 系列时，按照由 “速度单位选择 (0Wxx03 Bit 0 ~ 3)” 设定的单位。



- “VEL” 命令不被执行时，用设定参数 “各轴进给速度” 或以前的 “VEL” 命令的设定数据确定进给速度。
- 通过执行 “VEL” 命令变更的速度直到下次用 “VEL” 命令再设定或执行 “MVT” 命令为止一直有效。
- 在有 “MOV” 命令的块中，不能指令 “VEL” 命令。
- 通过 “VEL” 命令变更各轴设定参数 “快进给速度 (0Lxx22 / 0Lxx10)”。因此，可取代 “VEL” 命令与 “0Lxx22=xxxxx” (MP900 系列时) 进行编程。

(a) 设定举例 (MP930 时)



- 电机额定转速 : 3000 r/min
- 固定参数设定 No. 17 : mm 单位
No. 18 : 小数点以下 3 位
No. 19 : 机械转 1 圈 10000=10.000 mm

时，因用 VEL 命令指定电机额定转速，

$3000 \text{ r/min} \times 10.000 \text{ mm} = 30000 \text{ mm/min}$

所以按以下命令 axis1 为额定转速。

VEL [axis1]30000 ;

(3) 程序举例



```
INC;  
VEL [axis1] 500 [axis2] 500;  
MOV [axis1] 1000 [axis2] 1000;  
VEL [axis1] 1000;  
MOV [axis1] 1000 [axis2] 1000;
```

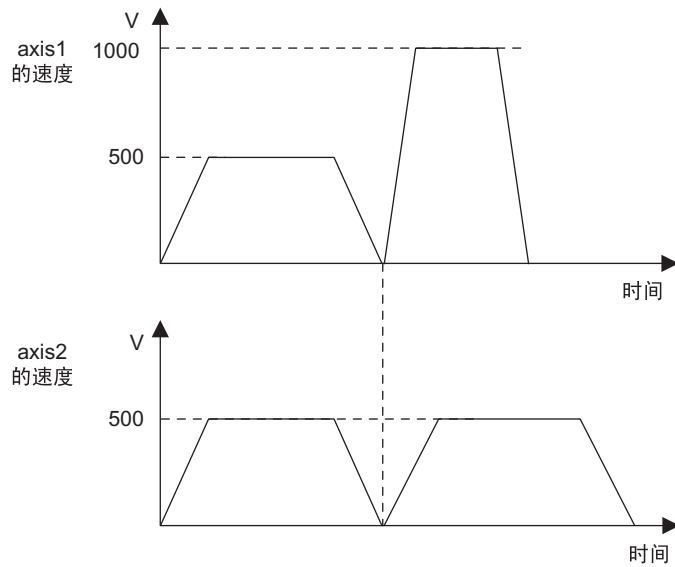


图 3.21 “进给速度变更 (VEL)” 的程序举例

3.2.5 插补进给速度比率设定 (IFP)

(1) 概要

“插补进给速度比率设定 (IFP)”命令可以用插补进给最大速度 (FMX) 的百分比来指定插补进给的速度。

(2) 详细说明

IFP 的指令方法如下所示。

```
IFP P_ ;
      速度指令值 (%)
```

通过上述指令，把插补进给最大速度 (FMX) 乘以指定百分比的值设定为插补进给速度。

“IFP”命令的指令范围如下。

1 ~ 100 [%]

重要

- 在有“插补命令”的块中，不能指定“IFP”命令。
- 在用“IFP”命令指定插补进给速度时（不使用F指令），在执行“插补命令”之前，请用“FMX”命令指定插补进给最高速度。
- 在“F”指令后执行“IFP”命令，则用“F”指定的插补进给速度被取消。
- 通过“IFP”命令设定的速度直到下一“IFP”命令执行或用“F”指令再设定为止一直有效。
- 使用插补命令的运动程序请务必在程序的开头指定“FMX”命令后再设定插补进给最高速度。没有“FMX”的指定时，指令“F”、“IFP”命令，则会出错。

(3) 程序举例



```
INC;
FMX T1000;
IFP P50
MVS [axis1]1000 [axis2]1000;
IFP P100
MVS [axis1]1000 [axis2]1000;
```

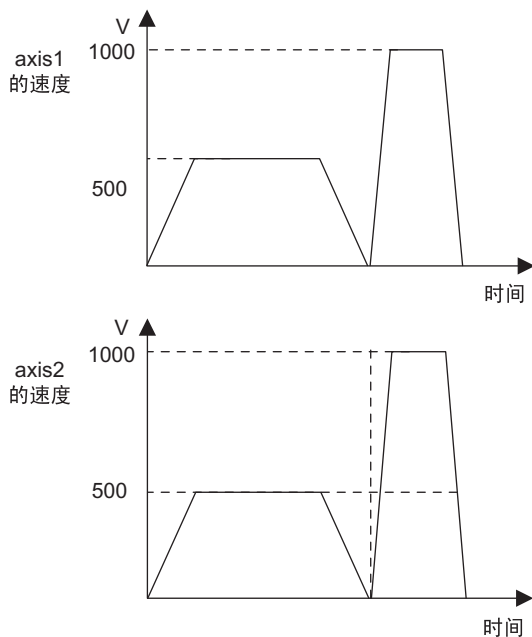


图 3.22 “插补进给速度比率设定 (IFP)”的程序举例

3.2.6 插补进给最高速度设定 (FMX)

(1) 概要

“插补进给最高速度设定 (FMX)” 命令可以设定插补进给 (MVS, MCC/MCW, SKP) 命令执行时的最高速度。通过该速度可以确定由 IAC、IDC 命令设定的插补加减速时间。

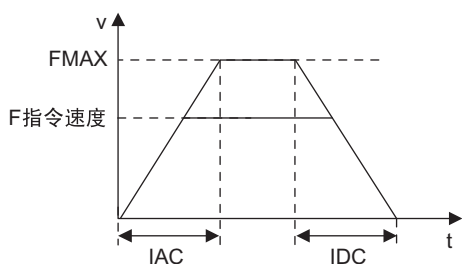
(2) 详细说明

FMX 命令的指令方法如下所示。

```
FMX T ;
      插补进给最高速度
```

通过上述指令，设定“插补进给最高速度”。在制作使用插补命令的运动程序时，请务必在程序的开头进行指定。一旦设定，直到再度设定为止一直有效。

通过“IAC”、“IDC”命令设定的插补加速时间为到 FMX 指定为止的加速时间及从 FMX 指定开始的减速时间。



“插补进给最高速度”

“插补进给速度比率设定 (IFP)” 命令把由 FMX 指定的最高速度乘以百分比的值设定为插补进给速度。

“FMX” 命令的指令范围如下。

$$1 \sim 2^{31} - 1 \text{ [指令单位 / min]}$$

(a) 设定举例 (MP930 时)



- 电机额定转速：3000 r/min
- 固定参数设定 No. 17：mm 单位
No. 18：小数点以下 3 位
No. 19：机械 1 转 10000=10.000 mm

时，因在 FMX 命令下以额定转速旋转 1 轴的电机，

$$3000 \times 10.000 \times 10^{\frac{3}{\downarrow \text{小数点以下位数}}} = 30000000$$

所以在下述命令下 axis1 为额定转速。

```
FMX T30000000 ;
```

(3) 程序举例



```

INC;
FMX T1000000;
IAC T1000;
IDC T1500;
MVS [axis1]100000 F700000;

```

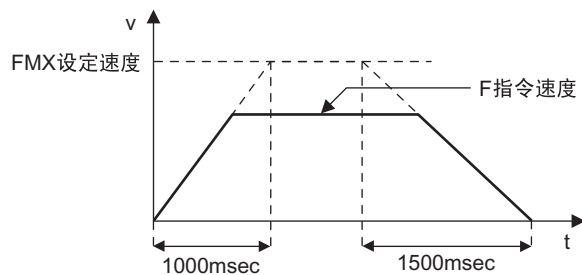


图 3.23 “插补进给最高速度设定”的程序举例

重要

- 在使用插补命令的运动程序中，如果不进行 FMX 指定，则会出错，敬请注意。
- FMX 指定不能使用小数点。设定时，请将小数点以下的位数都设为 0。

（例）小数点以下位数 = 3

指令单位 = mm

想将插补最高速度设为 50000.000 mm 时，

进行如下编程。

```
FMX T50000000;
```

3.2.7 插补加速时间变更 (IAC)

(1) 概要

“插补加速时间变更 (IAC)” 命令可以设定插补进给 (MVS、MCC/MCW、SKP) 命令执行时的加速时间。由运动程序中可以进行任意设定。

(2) 详细说明

IAC 命令的指令方法如下所示。

IAC <u> </u> I—; 加速时间

用 “IAC” 命令指定的插补加速时间是指达到由 FMX 命令指定的插补进给最高速度为止的时间。

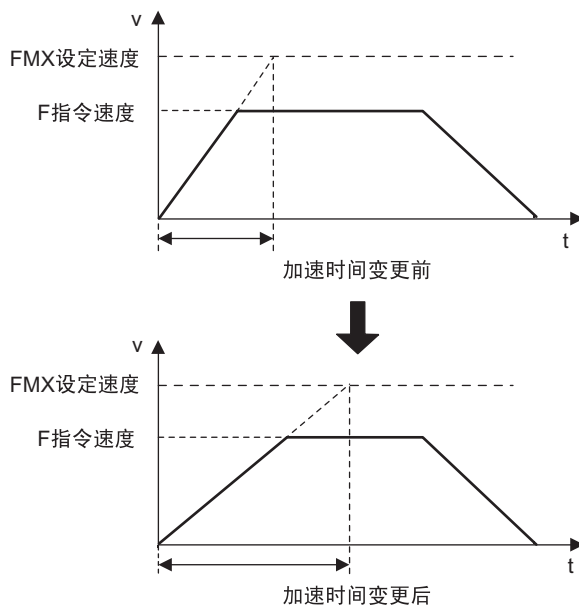


图 3.24 “插补加速时间变更”

“IAC” 命令的指令范围如下。

0 ~ 32767 [msec]



- 不执行 “IAC” 命令时，加速时间 = 0 (矩形)。
- 通过 “IAC” 命令执行变更的加速时间直到下次用 “IAC” 命令再设定为止一直有效。

(3) 程序举例



```

FMX T10000;
INC;
IAC T1000;
MVS [axis1]100000 F70000;
IAC T500;
MVS [axis1]100000;
    
```

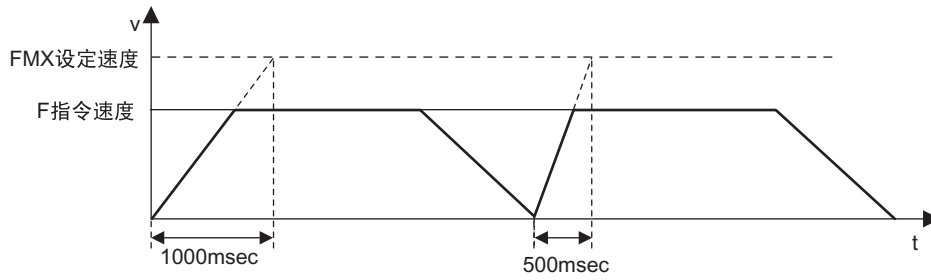


图 3.25 “插补加速时间变更”的程序举例

重要

指令“IAC”命令时，请事先用“FMX”命令设定插补进给最高速度。不指定“FMX”命令而直接指定“IAC”命令，则会出错。

3.2.8 插补减速时间变更 (IDC)

(1) 概要

“插补减速时间变更 (IDC)” 命令可以设定插补进给 (MVS、MCC/MCW、SKP) 命令执行时的减速时间。由运动程序中可以进行任意设定。

(2) 详细说明

IDC 命令的指令方法如下所示。

```
IDC T-;
    减速时间
```

用 IDC 命令指令的插补减速时间是指从 FMX 命令设定的速度开始到停止为止的时间。

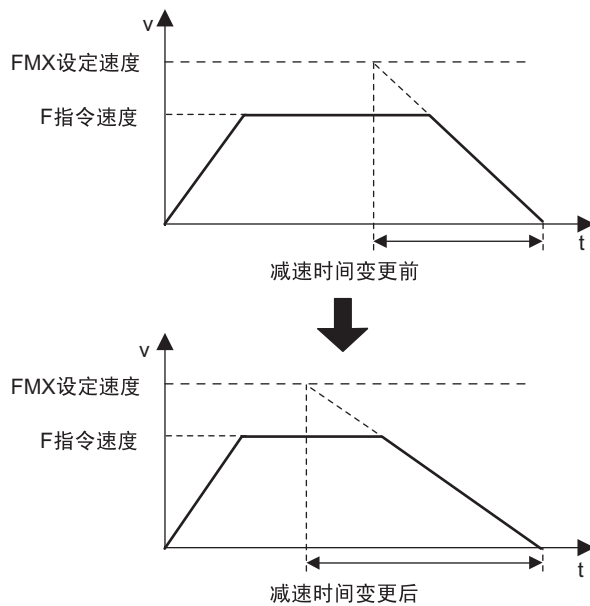


图 3.26 插补减速时间变更

“IDC” 命令的指令范围如下。

0 ~ 32767 [msec]



- 不执行“IDC”命令时，“减速时间参数”= 0(矩形)。
- 通过执行“IDC”命令变更的减速时间直到下次用“IDC”命令再设定为止一直有效。

(3) 程序举例



```
FMX T100000;  
INC;  
IDC T1000;  
MVS [axis1]100000 F70000;  
IDC T500;  
MVS [axis1]100000;
```

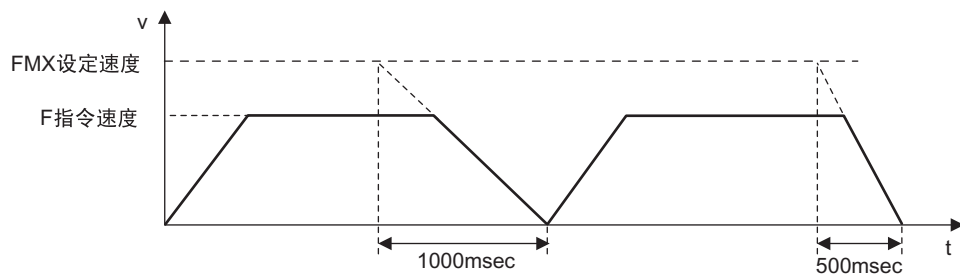


图 3.27 插补减速时间变更 (IDC) 的程序举例

重要

指令“IDC”命令时，请事先用“FMX”命令设定插补进给最高速度。不指定“FMX”命令而直接指定“IDC”命令，则会出错。

第 4 章

顺序命令

本章，对有关运算命令、分支命令及循环命令等顺序命令的编程方法进行了详细说明。

4.1 顺序命令概要	4-3
4.1.1 运算命令	4-3
4.1.2 数值运算组合	4-4
4.1.3 逻辑运算组合	4-4
4.2 数值运算	4-5
4.2.1 赋值 (=)	4-5
4.2.2 加法运算 (+)	4-6
4.2.3 减法运算 (-)	4-7
4.2.4 乘法运算 (*)	4-8
4.2.5 除法运算 (/)	4-9
4.2.6 余数 (MOD)	4-10
4.3 逻辑运算	4-11
4.3.1 逻辑或 ()	4-11
4.3.2 逻辑与 (&)	4-12
4.3.3 逻辑异或 (^)	4-13
4.3.4 取反 (!)	4-14
4.4 数值比较	4-15
4.4.1 数值比较命令 (==、<>、>、<、>=、<=)	4-15
4.5 数据操作	4-16
4.5.1 位右移 (SFR)	4-16
4.5.2 位左移 (SFL)	4-17
4.5.3 块传送 (BLK)	4-18
4.5.4 清除 (CLR)	4-19

4.6 基本函数	4-20
4.6.1 正弦 (SIN)	4-20
4.6.2 余弦 (COS)	4-21
4.6.3 正切 (TAN)	4-22
4.6.4 反正弦 (ASN)	4-23
4.6.5 反余弦 (ACS)	4-24
4.6.6 反正切 (ATN)	4-25
4.6.7 平方根 (SQT)	4-26
4.6.8 BCD → BIN (BIN)	4-28
4.6.9 BIN → BCD (BCD)	4-29
4.6.10 指定位 ON (S { })	4-30
4.6.11 指定位 OFF (R { })	4-31

4.1 顺序命令概要

本节对顺序命令的整体情况和运算组合进行说明。

4.1.1 运算命令

可以进行用运算符和函数结合全局变量、局部变量、常量的一般算术运算，并把计算结果赋给变量。运算的基本形式为：变量 = < 运算表达式 >，可以使用以下的运算和函数。

分类	命令	名称	指令格式
数值运算	=	赋值	MW- =MW- ;
	+	加法运算	MW- =MW- +MW- ;
	-	减法运算	MW- =MW- -MW- ;
	*	乘法运算	MW- =MW- *MW- ;
	/	除法运算	MW- =MW- /MW- ;
	MOD	余数运算	MW- =MOD;
逻辑运算		OR(逻辑或)	MB- =MB- MB- ;
	Λ	XOR(逻辑异或)	MW- =MW- Λ MW- ;
	&	AND(逻辑与)	MB- =MB- & MB- ;
	!	NOT(取反)	MB- =MB- &! MB- ;
数值比较	= =	等于	IF MW- =MW- ;
	< >	不等于	IF MW- < >MW- ;
	>	大于	IF MW- >MW- ;
	<	小于	IF MW- <MW- ;
	> =	大于等于	IF MW- >=MW- ;
	< =	小于等于	IF MW- <=MW- ;
数据操作	SFR	右移	SFR MB - N - W- ;
	SFL	左移	SFL MB - N - W- ;
	BLK	块传送	BLK MW - MW - W- ;
	CLR	清除	CLR MB - W- ;
基本函数	SIN	正弦	SIN(MW-);
	COS	余弦	COS(MW-);
	TAN	正切	TAN(MW-);
	ASN	反正弦	ASN(MW-);
	ACS	反余弦	ACS(MW-);
	ATN	反正切	ATN(MW-);
	SQRT	平方根	SQRT(MW-);
	BIN	BCD → BIN	BIN(MW-);
	BCD	BIN → BCD	BCD(MW-);
	S{}	指定位 ON	S{MB-} = MB- & MB- ;
	R{}	指定位 OFF	R{MB-} = MB- & MB- ;

4.1.2 数值运算组合

数值运算命令与函数相互组合使用，可以进行算术运算。

没有运算优先顺序。按从左到右的顺序进行运算。需要改变运算优先顺序时，可以使用（ ）。

运算表达式的举例如下所示。

(例) $MW00100 = MW00102 \pm MW00104 * SIN(MW00106);$

① ②

按编号顺序进行运算。

补充

■通过（ ）改变运算顺序的举例如下所示。

• $MW00100 = 100. \pm 200. * 0.5 * SIN(30.);$

(75.) ① ② ③

• $MW00100 = 100. \pm (200. * 0.5 * SIN(30.));$

(150.) ③ ① ②

• $MW00100 = 100. \pm (200. * 0.5) * SIN(30.)$

(100.) ② ① ③

4.1.3 逻辑运算组合

逻辑运算命令相互组合使用，可进行逻辑运算。

没有运算优先顺序。可以和算术运算混合运算，但不能进行实数运算。

运算表达式的举例如下所示。

(例) $MW00100 = MW00102 \& MW00104 | MW00106 \Delta MW00108$

① ② ③

按编号顺序进行运算。

补充

■通过（ ）改变运算顺序的举例如下所示。

• $MW00100 = MW00101 \& MW00102 | MW00103 \Delta MW00104;$

① ② ③

• $MW00100 = MW00101 \& MW00102 | (MW00103 \Delta MW00104);$

② ③ ①

4.2 数值运算

本节对算术运算命令进行说明。

4.2.1 赋值 (=)

(1) 概要

把表达式右边的运算结果赋给运算式左边的寄存器。

另外，数值运算和逻辑运算的运算优先顺序是不同的。详情请参阅“4.1.2 数值运算组合”及“4.1.3 逻辑运算组合”。

(2) 详细说明

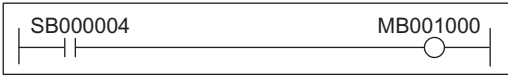



(a) 指令方法

(结果)=(运算表达式);

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
○	○	○	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	MB001000=1;	
	W	MW00100=12345;	
	L	ML00100=1234567;	
	F	MF00100=1.2345;	

4.2.2 加法运算 (+)

(1) 概要

进行右边的整数 / 实数的加法运算，把结果存储在左边的寄存器中。在右边的加法运算中，不使用寄存器也可以使用参数。整数 / 实数混合运算时，按左边的数据类型存储。

(2) 详细说明

(a) 指令方法

MW - = MW - + MW - ;

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
×	○	○	○

(3) 程序举例



type	运动程序	梯形程序
B	-	-
W	MW00101=MW00100+12345;	
L	ML00106=ML00102+ML00104;	
F	MF00202=MF00200+1.23456;	

重要

在进行数据类型不同的变量运算时，按左边数据类型结果会不同。详情请参阅“5.1 变量”。

4.2.3 减法运算 (-)

(1) 概要

进行右边的整数 / 实数的减法运算，把结果储存在左边的寄存器中。在右边的减法计算中，不使用寄存器也可以使用参数。整数 / 实数混合运算时，按左边的数据类型存储。

(2) 详细说明

(a) 指令方法

MW - = MW -- MW - ;

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
×	○	○	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	-	-
	W	MW00101=MW00100-12345;	
	L	ML00106=ML00102-ML00104;	
	F	MF00202=MF00200-1.23456;	

4.2.4 乘法运算 (*)

(1) 概要

进行右边的整数 / 实数的乘法运算，把结果存储在左边的寄存器中。在右边的乘法运算中，不使用寄存器也可以使用参数。整数 / 实数混合运算时，按左边的数据类型存储。

(2) 详细说明

(a) 指令方法

MW-- = MW-- * MW--;

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
×	○	○	○

(3) 程序举例



type	运动程序	梯形程序
B	-	-
W	MW00102=MW00100*MW00101	
L	ML00106=ML00102*ML00104;	
F	MF00202=MF00200*1.23456;	

4.2.5 除法运算 (/)

(1) 概要

进行右边的整数 / 实数的除法运算，把结果存储在左边的寄存器中。在右边的除法计算中，不使用寄存器也可以使用参数。整数 / 实数混合运算时，按左边的数据类型存储。

(2) 详细说明

(a) 指令方法

$MW- = MW- / MW-;$

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
×	○	○	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	-	-
	W	MW00102=MW00100/MW00101;	
	L	ML00106=ML00102/ML00104;	
	F	MF00202=MF00200/1.23456;	

4.2.6 余数 (MOD)

(1) 概要

当在除法运算命令的下面的块中指定时，把 MOD 作为余数存储在指定的变量中。在除法运算命令的块中，即使整数 / 实数混合运算时，余数也按左边的数据类型存储。

(2) 详细说明

(a) 指令方法

```
MW00001 = 1000 / 999;
MW00002 = MOD;   → MW00002 = 1
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
×	○	○	×

(3) 程序举例



type	运动程序	梯形程序
B	-	-
W	MW00101=MW00100/3; MW00102=MOD;	
L	ML00106=ML00102/ML00104; ML00108=MOD;	
F	-	-

(例) 长整型

```
ML00106=ML00100*ML00102/ML00104;
(173575) (100000) (60000) (34567)
ML00108=MOD;
(32975)
```

MOD 命令请务必在除法运算命令的下面的块中指定。在除法运算命令的下面的块以外执行时，不能保证运算结果的正确性。

4.3 逻辑运算

本节对执行位 / 整数的逻辑运算的命令进行说明。

没有运算优先顺序。从右边的第1项开始按顺序进行运算。但是，用()括起的部分必须首先进行运算。可以和算术运算混合运算，但不能进行实数运算。

4.3.1 逻辑或 (I)

(1) 概要

进行前一运算结果和指定寄存器的逻辑或运算，保留运算结果。不能使用实型寄存器。

表 4.1 逻辑或真值表 (C=A|B)

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

(2) 详细说明

(a) 指令方法

```
MB00100=IB000100|SB000200;
MW00100=DW00102|AAAAH;
ML00104=DL00106|IL00100;
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
○	○	○	×	○

(3) 程序举例



type	运动程序	梯形程序
B	MB001000=MB001010 MB001011;	
W	MW00100=MW00101 MW00102	
L	ML00106=ML00102 ML00104;	
F	-	-

4.3.2 逻辑与 (&)

(1) 概要

进行前一运算结果和指定寄存器的逻辑与运算，保留运算结果。不能使用实型寄存器。

表 4.2 逻辑与真值表 (C=A&B)

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

(2) 详细说明

(a) 指令方法

```

MB001000=IB001000&MB001010;
MW00100=MW00101&MW00102;
ML00100=ML00102&5555ACACH;
    
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
○	○	○	×	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	MB001000=MB001010&MB001011;	
	W	MW00101=MW00100&00FFH;	
	L	ML00106=ML00102&ML00104;	
	F	-	-

4.3.3 逻辑异或 (Λ)

(1) 概要

进行前一运算结果和指定寄存器的逻辑异或运算，保留运算结果。不能使用实型寄存器。

表 4.3 逻辑异或真值表
($C=A \Lambda B$)

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

(2) 详细说明

(a) 指令方法

```
MW00100=MW00101 Λ MW00102;
ML00100=ML00102 Λ 12345678H;
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	○	×	○

(3) 程序举例

例

type	运动程序	梯形程序
B	-	-
W	MW00101=MW00100 Λ 00FFH;	
L	ML00106=ML00102 Λ ML00104;	
F	-	-

4.3.4 取反 (!)

(1) 概要

对指定寄存器的数据进行取反运算，保留运算结果。不能使用实型寄存器。

(2) 详细说明

(a) 指令方法

```

MB001000=!MW00101;
MW00100=!MW00101;
ML00100=!ML00102;
    
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
○	○	○	×	△*

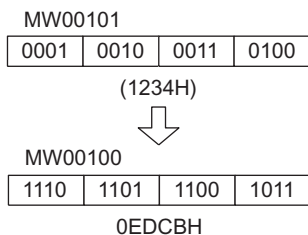
* 不可指定比特型参数

(3) 程序举例



type	运动程序	梯形程序
B	MB001000=!MB001010;	
W	MW00100=!MW00101;	
L	ML00100=!ML00102	
F	-	-

(例) MW00100=!MW00101;



4.4 数值比较

本节对于用于条件表达式判断的数值比较命令进行说明。

4.4.1 数值比较命令 (==、<>、>、<、>=、<=)

(1) 概要

用于分支命令 (IF)、循环命令 (WHILE)、输入输出等待命令 (IOW) 等的“条件表达式”的判断。比较命令有以下 6 种。

比较命令	意义
= =	等于
< >	不等于
>	大于
<	小于
> =	大于等于
< =	小于等于

(2) 详细说明

(a) 指令方法

```
IF MW00100==MW00102 + MW00104;
WHILE MB001001<>1;
```

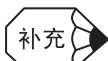
(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)
○ *	○	○	○

* 在比特型条件表达式中，仅“= =”命令有效。

(3) 程序举例

例	type	运动程序	梯形程序
	B	IF MB001000==1;	
	W	IF MW00100<>10;	
	L	IF ML00100>10000;	
	F	IF MF00100>=3.0;	



补充

梯形命令时，请把比较结果存储在比特型寄存器中。用 IFON 命令判断该比较结果。

4.5 数据操作

本节对进行数据转移、移动、初始化的数据操作命令进行说明。

4.5.1 位右移 (SFR)

(1) 概要

SFR 命令可以使由位首编号和位宽指定的位列，按指定的移动位数，向右移动。

(2) 详细说明

(a) 指令方法

SFR	MB001000	N5	W10 ;
	A	B	C
	A: 位首编号		
	B: 移动位数		
	C: 位宽		

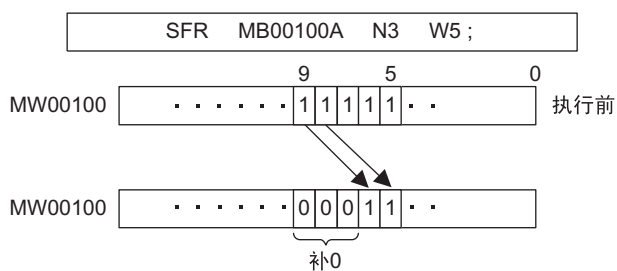
(b) 数据类型

	比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
位首编号	○	×	×	×	×
移动位数	×	○	×	×	○
位宽	×	○	×	×	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	-	-
	W	SFR MB001000 N=5 W=10;	SHFTR MB001000 N=5 W=10
	L	-	-
	F	-	-

(例) 使以 MB001005 (MW00100 的位 5) 为首位的位宽为 5 的数据向右移动 3 位。



补充

使用 SFR 命令时，若移动位数 \geq 位宽，则指定位宽的数据全变为 0。

4.5.2 位左移 (SFL)

(1) 概要

SFL 命令可以使由位首编号和位宽指定的位列，按指定的移动位数，向左移动。

(2) 详细说明

(a) 指令方法

SFL	<u>MB001000</u>	<u>N5</u>	<u>W10</u> ;
	A	B	C
	A: 位首编号		
	B: 移动位数		
	C: 位宽		

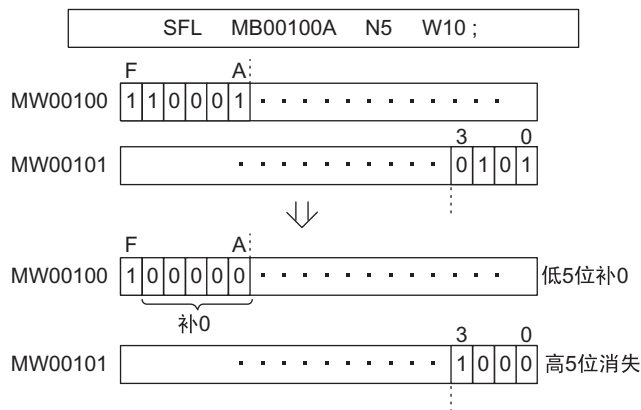
(b) 数据类型

	比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
位首编号	○	×	×	×	×
移动位数	×	○	×	×	○
位宽	×	○	×	×	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	-	-
	W	SFL MB001000 N=5 W=10	SHFTL MB001000 N=5 W=10
	L	-	-
	F	-	-

(例) 使以 MB00100A (MW00100 的位 A) 为首位的位宽为 10 的数据向左移动 5 位。



使用 SFL 命令时，若移动位数 \geq 位宽，则指定位宽的数据全变为 0。

4.5.3 块传送 (BLK)

(1) 概要

BLK 命令可以从传送源寄存器的首地址向传送目标寄存器的首地址，传送指定块数的字数据。

(2) 详细说明

(a) 指令方法

BLK	<u>MW00100</u>	<u>DW00100</u>	<u>W10</u> ;
	A	B	C
	A: 传送源首寄存器		
	B: 传送目标首寄存器		
	C: 传送块数		

(b) 数据类型

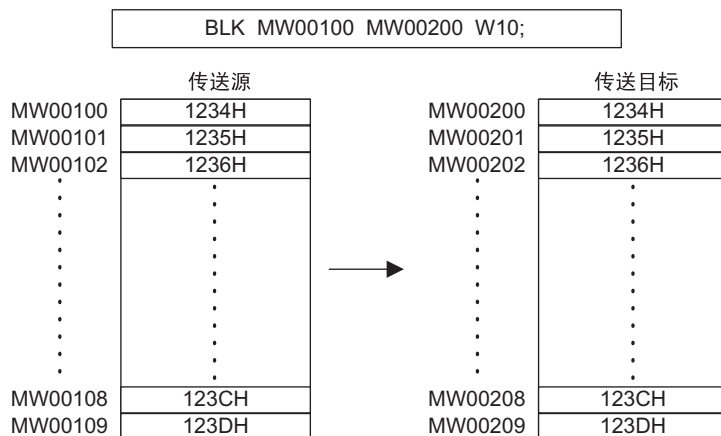
	比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
传送源首寄存器	×	○	×	×	×
传送目标首寄存器	×	○	×	×	×
传送块数	×	○	×	×	○

(3) 程序举例



type	运动程序	梯形程序
B	-	-
W	BLK MW00100 DW00100 W10;	COPYW MW00100⇒DW00100 W=10
L	-	-
F	-	-

(例) 向 MW00100 ~ MW00109 传送 MW00200 ~ MW00209。



即使传送源和传送目标重叠，传送的数据块也会被原样传送到传送目标中。

4.5.4 清除 (CLR)

(1) 概要

CLR 命令可以从要清除的数据首寄存器开始，对指定块数的字数据进行清 0。

(2) 详细说明

(a) 指令方法

CLR	MW00100	W10 ;
	A	C
	A: 数据清除首寄存器	
	B: 块数	

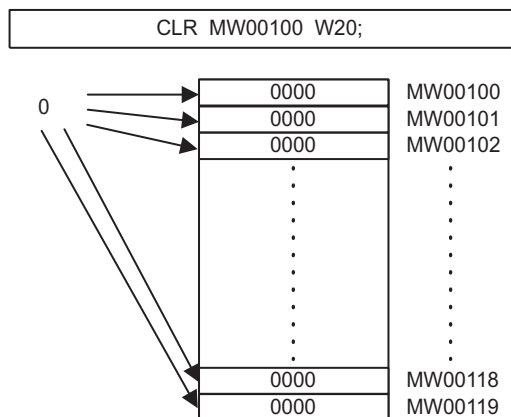
(b) 数据类型

	比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
数据清除首寄存器	×	○	×	×	×
块数	×	○	×	×	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	—	—
	W	CLR MW00100 W10;	SETW MW00100 D=00000 W=10
	L	—	—
	F	—	—

(例) 将 MW00100 ~ MW00119 的内容清 0。



4.6 基本函数

本节对三角函数、平方根、BIN、BCD 等基本函数命令进行说明。

4.6.1 正弦 (SIN)

(1) 概要

SIN 命令将整型及实型数据的正弦作为运算结果保留。

不能使用长整型数据。

(2) 详细说明

(a) 指令方法

```
MW00100=SIN(MW00101);
MW00100=SIN(90);
MF00200=SIN(MF00202);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	×	○	○

(3) 程序举例

type	运动程序	梯形程序
B	-	-
W	MW00102=SIN(MW00100);	
L	-	-
F	DF00202=SIN(DF00200);	

整型及实型的输入单位、输出结果不同。

• 整型时

-327.68 ~ 327.67 度范围内可使用。把前一运算结果（整型数据）作为输入使用，并把该运算结果保留在整型寄存器中。（输入单位 1=0.01 度）运算结果按 *10000 倍输出。

• 实型时

把前一运算结果（实型数据）作为输入使用，并把该正弦运算结果保留在实型寄存器中。（单位 = 度）

（例）

• 整型数据

```
MW00102 = SIN ( MW00100 );
(05000)          (03000)
```

等价 ⇒ 0.5=SIN30°

• 实型数据

```
MF00102 = SIN ( MF00100 );
(0.5)          (30.0)
```

重要

整型时，如果输入 -327.68 ~ 327.67 度范围外的数据，无法得出正确的运算结果。

4.6.2 余弦 (COS)

(1) 概要

COS 命令将整型及实型数据的余弦值作为运算结果保留。
不能使用长整型数据。

(2) 详细说明

(a) 指令方法

```
MW00100=COS (MW00101) ;
MW00100=COS (90) ;
MF00200=COS (MF00202)
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	×	○	○

(3) 程序举例

type	运动程序	梯形程序
B	-	-
W	MW00102=COS (MW00100) ;	
L	-	-
F	DF00202=COS (DF00200) ;	

整型及实型的输入单位、输出结果不同。

- 整型时

-327.68 ~ 327.67 度范围内可使用。把前一运算结果（整型数据）作为输入使用，并把该运算结果保留在整型寄存器中。（输入单位 1=0.01 度）运算结果按 *10000 倍输出。

- 实型时

把前一运算结果（实型数据）作为输入使用，并把该正弦运算结果保留在实型寄存器中。（单位 = 度）

(例)

- 整型数据

```
MW00102 = COS ( MW00100 ) ;      等价 => 0.5=COS60°
(05000)          (06000)
```

- 实型数据

```
MF00102 = COS ( MF00100 ) ;
(0.5)          (60.0)
```



整型时，如果输入 -327.68 ~ 327.67 度范围外的数据，无法得出正确的运算结果。

4.6.3 正切 (TAN)

(1) 概要

TAN 命令把指定的变量及参数 (单位 = 度) 作为输入使用, 并把该正切保留在实型寄存器中。

(2) 详细说明

(a) 指令方法

```
MF00100=TAN(MF00102);
MF00200=TAN(1.0);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	×	×	○	○

(3) 程序举例



type	运动程序	梯形程序
B	-	-
W	-	-
L	-	-
F	DF00202=TAN(DF00200);	

(例) 输入值 ($\theta=45.0$ 度) 的正切: 计算 $TAN(\theta)=1.0$ 。

```
DF00102=TAN(DF00100);
(1.0)      (45.0)
```

重要

TAN 命令只可使用实型数据。若指定位 / 整数 / 长整数, 编译时将会出错。

4.6.4 反正弦 (ASN)

(1) 概要

ASN 命令把指定的变量及参数作为输入使用，并把该反正弦（单位 = 度）保留在实型寄存器中。

(2) 详细说明

(a) 指令方法

```
MF00100=ASN(MF00102);
MF00200=ASN(1.0);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	×	×	○	○

(3) 程序举例

type	运动程序	梯形程序
B	—	—
W	—	—
L	—	—
F	DF00202=ASN(DF00200);	

(例) 输入值 (0.5) 的反正弦：计算 $ASN(0.5)=30.0$ 度。

```
MF00202=ASN(MF00200);
(30.0)      (0.5)
```

重要

ASN 命令只可使用实型数据。若指定位 / 整数 / 长整数，编译时将会出错。

4.6.5 反余弦 (ACS)

(1) 概要

ACS 命令把指定的变量及参数作为输入使用，并把该反余弦（单位 = 度）保留在实型寄存器中。

(2) 详细说明

(a) 指令方法

```
MF00100=ACS (MF00102);
MF00200=ACS (60.0);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	×	×	○	○

(3) 程序举例



type	运动程序	梯形程序
B	—	—
W	—	—
L	—	—
F	DF00202=ACS (DF00200);	

(例) 输入值 (0.5) 的反余弦：计算 $ACS(0.5)=60.0$ 度。

```
MF00100 = ACS ( MF00102 );
(0.5)          (60.0)
```

重要

ACS 命令只可使用实型数据。若指定位 / 整数 / 长整数，编译时将会出错。

4.6.6 反正切 (ATN)

(1) 概要

ATN 命令把整型及实型数据的反正切作为运算结果保留。
不能使用长整型数据。

(2) 详细说明



(a) 指令方法

```
MW00100=ATN(MW00101);
MW00100=ATN(100);
MF00200=ATN(MF00202);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	×	○	○

(3) 程序举例

type	运动程序	梯形程序
B	-	-
W	MW00102=ATN(MW00100);	
L	-	-
F	DF00202=ATN(DF00200);	

整型及实型的输入单位、输出结果不同。

- 整型时

-327.68 ~ 327.67 度范围内可使用。把前一运算结果（整型数据）作为输入使用，并把该运算结果保留在整型寄存器中。（输入单位 1=0.01 度）（运算结果按 *100 度值输出。）

- 实型时

把前一运算结果（实型数据）作为输入使用，并把该反正切保留在实型寄存器中。

（例）

- 整型数据

MW00100 = ATN (MW00102);	等价 ⇒	45=ATN(1.0)
(04500) (00100)		

- 实型数据

MF00100 = ATN (MF00102);
(45.0) (1.0)

4.6.7 平方根 (SQT)

(1) 概要

SQT 命令把整数 / 实数的平方根作为运算结果保留。
不能使用长整型数据。

(2) 详细说明

(a) 指令方法

```
MW00100=SQT (MW00101);
MW00100=SQT (100);
MF00200=SQT (MF00202);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	×	○	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	—	—
	W	MW00102=SQT (MW00100);	
	L	—	—
	F	DF00202=SQT (DF00200);	

整型及实型的输入单位、输出结果不同。

• 整型数据

和数学上的平方根不同，用以下计算式进行运算。

$$32768 * \text{sign}(A) * \sqrt{|A|/32768}$$

但， $\text{sign}(A)$: A 寄存器符号

$|A|$: A 寄存器绝对值

既为数学平方根值乘以 $\sqrt{32768}$ 的值。另外，输入为负数时，则计算绝对值的平方根，输出负数的运算结果。运算误差最大为 ± 2 。

• 实型数据

把前一运算结果（实型数据）作为输入使用，并把该平方根以实型数据保留。

(例)

• 整型数据

输入为正数时

$$\boxed{\text{MW00100} = \text{SQT}(\text{MW00102});} \quad \sqrt{64} \times \sqrt{32768} = 1448$$

(01448) (00064) (8) (181)

输入为负数时

$$\boxed{\text{MW00100} = \text{SQT}(\text{MW00102});} - (\sqrt{64} \times \sqrt{32768}) = 1448$$

(-01448) (-00064) (8) (181)

• 实型数据

输入为正数时

$$\boxed{\text{MF00100} = \text{SQT}(\text{MF00102});}$$

(8.0) (64.0)

输入为负数时

$$\boxed{\text{MF00100} = \text{SQT}(\text{MF00102});}$$

(-8.0) (-64.0)

4.6.8 BCD → BIN(BIN)

(1) 概要

BIN 命令将 BCD 代码数据转换为 2 进制数 (BIN 代码)。

只能使用整数数据。当指定非 BCD 代码数值时, 无法得出正确结果。

(2) 详细说明

(a) 指令方法

```
MW00100=BIN(MW00101);
MW00100=BIN(1234H);
ML00200=BIN(ML00202);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	○	×	○

(3) 程序举例

例	type	运动程序	梯形程序
	B	-	-
	W	MW00101=BIN(MW00100);	
	L	ML00102=BIN(ML00100);	
	F	-	-

(例 1)



(例 2)



当指定 BCD 代码以外的数据时, 无法得出正确结果。

4.6.9 BIN → BCD (BCD)

(1) 概要

BCD 命令将 2 进制数 (BIN 代码) 数据转换为 BCD 代码数据。

只能使用整数数据。BIN 数据为 9999 以上或为负值时, 无法得出正确结果。

(2) 详细说明

(a) 指令方法

```
MW00100=BCD (MW00101);
MW00100=BCD (1234);
ML00200=BCD (ML00202);
```

(b) 数据类型

比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
×	○	○	×	○

(3) 程序举例

type	运动程序	梯形程序
B	-	-
W	MW00101=BCD (MW00100);	
L	ML00102=BCD (ML00100);	
F	-	-

(例 1)



(例 2)



BIN 数据大于 9999 时, 无法得出正确结果。

4.6.10 指定位 ON(S{ })

(1) 概要

当逻辑运算结果为“真”时，则指定位 ON 命令将指定位置为 ON。即使逻辑运算结果为“假”，也不将指定位置为 OFF。

(2) 详细说明

(a) 指令方法

S{MB001000}=MB001010&MB001011;	
A	B
A: 指定位	
B: 逻辑运算表达式数据类型	

(b) 数据类型

	比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
指定位	○	×	×	×	×
逻辑运算表达式	○	×	×	×	○

(3) 程序举例



type	运动程序	梯形程序
B	S {MB001000} =MB001010&MB001011 ;	-
W	-	-
L	-	-
F	-	-

4.6.11 指定位 OFF (R{ })

(1) 概要

当逻辑运算结果为“真”时，则指定位 OFF 命令将指定位置为 OFF。即使逻辑运算结果为“假”，也不将指定位置为 ON。

(2) 详细说明

(a) 指令方法

R{MB001000}=MB001010&MB001011;

A B

A: 指定位

B: 逻辑运算表达式

(b) 数据类型

	比特型 (B)	整型 (W)	长整型 (L)	实型 (F)	参数
指定位	○	×	×	×	×
逻辑运算表达式	○	×	×	×	○

(3) 程序举例

例 ▶▶▶

type	运动程序	梯形程序
B	R {MB001000} =MB001010&MB001011 ;	-
W	-	-
L	-	-
F	-	-

第 5 章

变量（寄存器）

本章对在运动程序中可使用的“变量”的详细情况进行了说明。

5.1 变量	5-2
5.1.1 变量概要	5-2
5.1.2 全局变量和局部变量	5-4
5.2 变量使用方法	5-7
5.2.1 系统变量 (S 寄存器)	5-7
5.2.2 数据变量 (M 寄存器)	5-8
5.2.3 输入变量 (I 寄存器)	5-9
5.2.4 输出变量 (O 寄存器)	5-12
5.2.5 C 变量 (C 寄存器)	5-15
5.2.6 D 变量 (D 寄存器)	5-16

5.1 变量

本节对变量的整体情况进行说明。

5.1.1 变量概要

在运动程序中可用“变量”进行表述，而不直接赋予“数值”。在实际使用“变量”时，是将存储在“变量域”的数值取出后使用。

(1) 变量（寄存器）种类

在运动程序中，可以把下表中的7种寄存器作为变量使用。S、M、I、O、C寄存器是运动程序和梯形程序可通用的全局变量。#、D寄存器仅是可进行程序单独调用的局部变量。但由于局部变量保存在程序内存中，因此程序的可用空间也随之减少。

表 5.1 变量种类

类别	名称	指定方法	范围	内容	特性
S	系统寄存器	SB, SW, SL, SFnnnnn	SW00000 ~ SW01023	系统中准备的寄存器。寄存器编号 nnnnn 用 10 进制表示。系统启动时，SW00000 ~ SW00049 被清 0。	程序通用
M	数据寄存器	MB, MW, ML, MFnnnnn	MW00000 ~ MW32767	各 DWG 间通用的寄存器。用于 DWG 间的 I/F 等。寄存器编号 nnnnn 用 10 进制表示。	
I	输入寄存器	IB, IW, IL, IFhhhh	IW0000 ~ IW07FF	I/O 模块输入数据用寄存器。寄存器编号 hhhh 用 16 进制表示。编号 C000 以后的寄存器被用于和伺服监视器参数连接的接口。	
O	输出寄存器	OB, OW, OL, OFhhhh	OW0000 ~ OW07FF	I/O 模块输出数据用寄存器。寄存器编号 hhhh 用 16 进制表示。编号 C000 以后的寄存器被用于和伺服设定参数连接的接口。	
C	参数寄存器	CB, CW, CL, CFnnnnn	CW00000 ~ CW4095	程序仅可调用的寄存器。寄存器编号 nnnnn 用 10 进制表示。	
#	# 寄存器	#B, #W, #L, #Fnnnnn	#W00000 ~ #W16383	程序仅可调用的寄存器。仅相应 DWG 可调用。实际利用范围在 MPE720 中由用户指定。寄存器编号 nnnnn 用 10 进制表示。但在运动程序中不能使用。	程序单独
D	D 寄存器	DB, DW, DL, DFnnnnn	DW00000 ~ DW16383	各 DWG 固有的内部寄存器。仅相应 DWG 可调用。实际利用范围在 MPE720 上由用户指定。寄存器编号 nnnnn 用 10 进制表示。	

重要

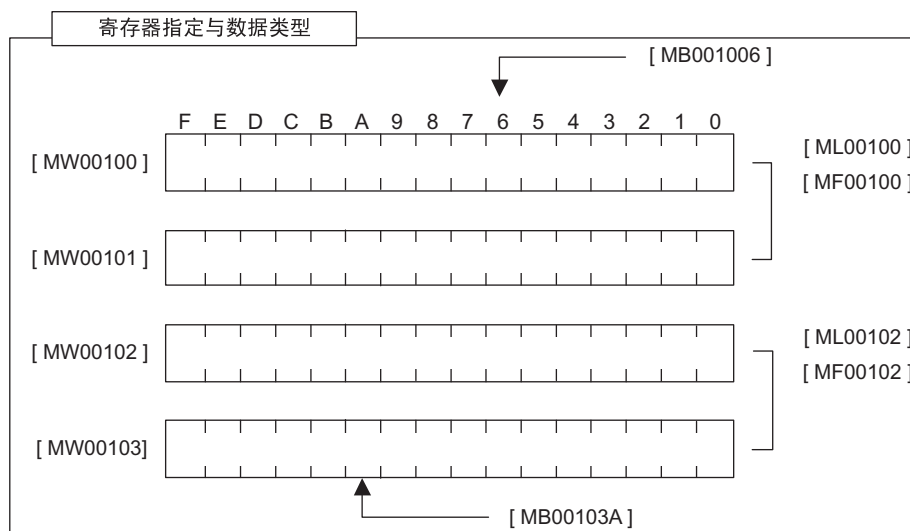
在运动程序中不能使用 # 寄存器。如果使用，运动程序编译时会成为“语法错误”。

(2) 数据类型

如下表所示，数据类型有比特型、整型、长整型、实型，可根据目的的不同区别使用。

表 5.2 数据类型

符号	数据类型	数值范围	备注
B	比特型	ON(1)、OFF(0)	用于继电器顺序和 ON/OFF 的条件判断
W	整型	-32768 ~ +32767 (8000H ~ 7FFFH)	用于数值运算。() 表示用于逻辑运算时的数值范围。
L	长整型	-2147483648 ~ +2147483647 (80000000H ~ 7FFFFFFFH)	用于数值运算。() 表示用于逻辑运算时的数值范围。
F	实型	$\pm(1.175E-38 \sim 3.402E+38)$	用于高级数值运算。



5.1.2 全局变量和局部变量

(1) 全局变量

全局变量是一种可梯形程序、用户函数、运动程序各图通用的变量。即，某个梯形程序运算的结果可以被其他用户函数和运动程序使用。全局变量大小按变量不同被分别保存在系统中。（参照下图）

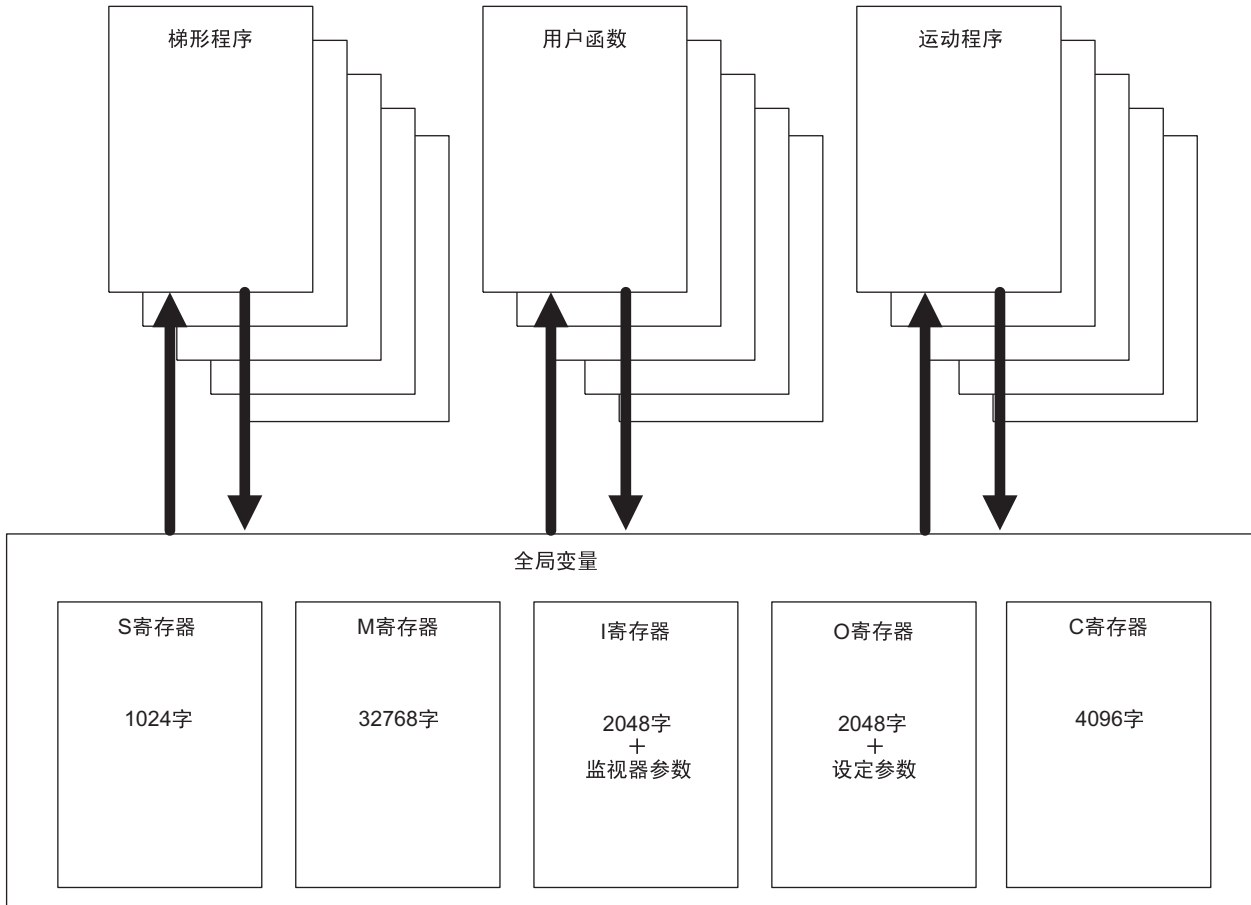
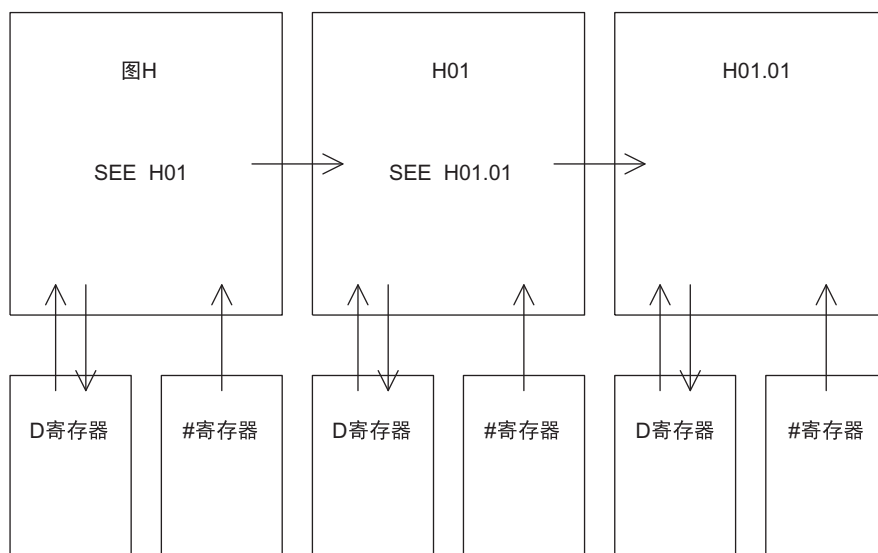


图 5.1 全局变量（为 MP930 时）

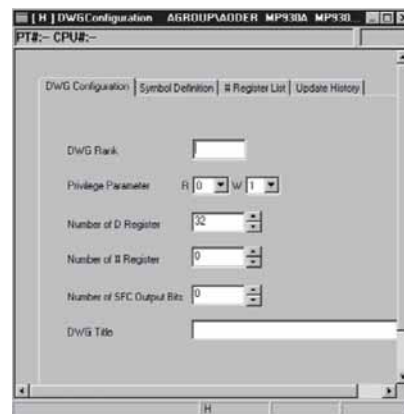
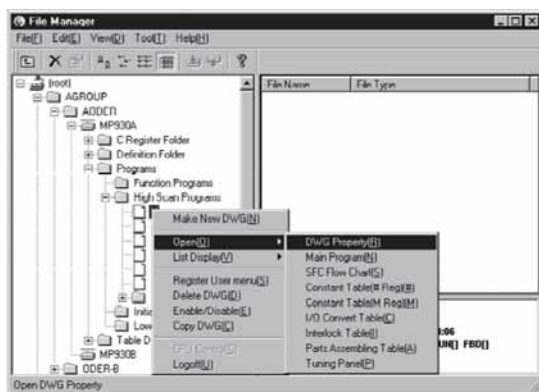
(2) 局部变量

局部变量是指在每个程序中局部使用的变量。在每个程序单位内被独立保存的变量，可存储程序内的运算结果等。

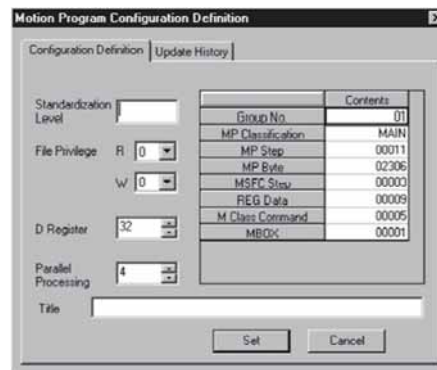
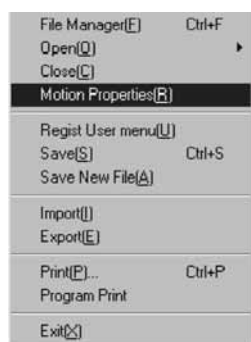


在各程序中的变量范围是在 DWG 构成定义及运动程序构成定义画面中指定的。在一个图中最多可保存 16 k 字。

DWG 构成定义画面



运动程序构成定义画面



重要

在运动程序中不能使用 # 寄存器。

5

重要

■ 变量运算时的注意事项

(a) 出错的程序

以下程序为出错程序。

- 整数数据存储于比特型变量中

```
MB000100 = 123;
MB000100 = MW00100;
```

(b) 存储在数据类型不同的变量中时

存储在数据类型不同的变量中时，会有以下结果。

- 实型数据存储于整型变量中

```
MW00100 = MF00200;    将实数转换成整数后存储。
(00001)  (1.234)
```

- 实型数据存储于长整型变量中

```
ML00100 = MF00200;    将实数转换成整数后存储。
(123457) (123456.7)
```

- 长整型数据存储于整型变量中

```
MW00100 = ML00200;    将长整型数据的低 16 位
(-00001) (65535)      原样存储。
```

- 整型数据存储于长整型变量中

```
ML00100 = MW00200;    将整型数据转换成长整型数据后
(0001234) (1234)      存储。
```

(c) 实型数据存储于整型变量中时

实型数据存储于整型变量中时，请注意舍入误差。

```
MW00100 = MF00200 + MF00202;
(0124)  (123.48)  (0.02)  运算变量值不同运算结果
(0123)  (123.49)  (0.01)  也不同。
```


5.2 变量使用方法

本节对各变量的使用方法进行说明。

5.2.1 系统变量 (S 寄存器)

(1) 概要

系统变量 (S 寄存器) 是运动控制器系统准备的寄存器，通过它可以读出系统的错误信息和运行状况等。S 寄存器是运动程序中通用的全局变量。详情请参阅机器控制器的用户手册。

(2) 详细说明

S 寄存器的指令方法如下所示。

SB000000	~	SB01023F
SW000000	~	SW01023
SL000000	~	SL01023
SF000000	~	SF01023

变量编号指定为 10 进制数表述。但指定位时的 Bit 编号指令为 16 进制数表述。

(3) 程序举例



- 位指定
OB000010=SB000402|SB000403;
- 整数指令
MW00100=SW00041;
- 长整数指令
ML00100=SL00062;

重要

系统寄存器 (S) 为数据读出专用，进行数据写入时，不能保证系统动作。

5.2.2 数据变量（M寄存器）

(1) 概要

M寄存器是指在梯形程序、用户函数和各运动程序中通用的变量，它也是可用于运动程序间及梯形程序间接口等的全局变量。

(2) 详细说明

M寄存器的指令方法如下所示。

MB000000	~	MB32767F
MW000000	~	MW32767
ML000000	~	ML32767
MF000000	~	MF32767

M寄存器可以在各种运算中作为“变量”使用并赋给它运算结果，或可用“变量”来指定定位的坐标值和速度。变量编号指定为10进制数表述。

(3) 程序举例

(a) 在轴移动命令中，用变量指令位置、速度时



<ul style="list-style-type: none"> • 参数 指令单位 = mm 小数点位置 = 3 时 	
ML00100=100000;	→ 1000.000 mm
ML00102=200000;	→ 2,000.000 mm
ML00104=300000;	→ 3,000.000 mm
ML00106=500000;	→ 5000.000 mm/min
MVS [X]ML00100 [Y]ML00102 [Z]ML00104 FML00106;	

(b) 运算中使用变量时



- 位指定
MB001001=IB000100 & IB000201;
- 整数指定
MW00101=(MW00101 | MW00102) & FFOCH;
- 长整数指定
ML00200=(M100202*ML00204/ML00206)*3;
- 实数指定
MF00200=MF00202*MF00204/MF00206*3.14;

重要

在以下运动命令中，用变量指定移动量坐标值或速度时，必须使用长整型。

MOV, MVS, MCW/MCC, ZRN, SKP, MVT, EXM, POS,
ACC, SCC, IAC, IDC, IFP, FMX, INP

5.2.3 输入变量 (I 寄存器)

(1) 概要

是使用外部输入信号及伺服监视器参数的变量。但伺服监视器参数为数据读出专用，进行数据写入时，不能保证系统动作。

(2) 详细说明

I 寄存器的指令方法如下所示。

• MP910 时
IW0000 ~ IW13FF ... 外部输入信号
IWC000 ~ IWC77F ... 伺服监视器参数
• MP920 时
IW0000 ~ IW13FF ... 外部输入信号
IWC000 ~ IWFF7F ... 伺服监视器参数
• MP930 时
IW0000 ~ IW07FF ... 外部输入信号
IWC000 ~ IWC37F ... 伺服监视器参数
• MP940 时
IW0000 ~ IW07FF ... 外部输入信号
IWC000 ~ IWF03F ... 伺服监视器参数
• MP2100、MP2300 时
IW0000 ~ IW7FFF ... 外部输入信号
IW8000 ~ IWFFFF ... 伺服监视器参数

(a) 外部输入信号的寄存器编号

根据在模块构成定义中设定的地址而定。但，MP930 的 MC350 模块内局部输入固定为 IW0000 (IB00000 ~ IB0000F 16 点)

(b) 伺服监视器参数寄存器编号

模块的种类不同控制轴数也不同。不同模块的控制轴数及最大模块数如下表所示。

表 5.3 模块的控制轴数

模块名称		控制轴数 / 模块	最大模块数
MP910		14	2
MP920	SVA-01	4	15
	SVA-02	2	16
	SVB-01	14	16
	PO-01	4	16
MP930		14	1
MP940		1	1
MP2100		16	1
MP2300		16	1

表 5.4 运动参数寄存器编号（MP900 系列时）

电路编号	1 轴	2 轴	3 轴	4 轴	5 轴	• •	14 轴
1	C000 ~ C03F	C040 ~ C07F	C080 ~ C0BF	C0C0 ~ C0FF	C100 ~ C13F	• •	C340 ~ C37F
2	C400 ~ C43F	C440 ~ C47F	C480 ~ C4BF	C4C0 ~ C4FF	C500 ~ C53F	• •	C740 ~ C77F
3	C800 ~ C83F	C840 ~ C87F	C880 ~ C8BF	C8C0 ~ C8FF	C900 ~ C93F	• •	CB40 ~ CB7F
4	CC00 ~ CC3F	CC40 ~ CC7F	CC80 ~ CCBF	CCC0 ~ CCFE	CD00 ~ CD3F	• •	CF40 ~ CF7F
5	D000 ~ D03F	D040 ~ D07F	D080 ~ D0BF	D0C0 ~ D0FF	D100 ~ D13F	• •	D340 ~ D37F
6	D400 ~ D43F	D440 ~ D47F	D480 ~ D4BF	D4C0 ~ D4FF	D500 ~ D53F	• •	D740 ~ D77F
7	D800 ~ D83F	D840 ~ D87F	D880 ~ D8BF	D8C0 ~ D8FF	D900 ~ D93F	• •	DB40 ~ DB7F
8	DC00 ~ DC3F	DC40 ~ DC7F	DC80 ~ DCBF	DCC0 ~ DCFE	DD00 ~ DD3F	• •	DF40 ~ DF7F
9	E000 ~ E03F	E040 ~ E07F	E080 ~ E0BF	E0C0 ~ E0FF	E100 ~ E13F	• •	E340 ~ E37F
10	E400 ~ E43F	E440 ~ E47F	E480 ~ E4BF	E4C0 ~ E4FF	E500 ~ E53F	• •	E740 ~ E77F
11	E800 ~ E83F	E840 ~ E87F	E880 ~ E8BF	E8C0 ~ E8FF	E900 ~ E93F	• •	EB40 ~ EB7F
12	EC00 ~ EC3F	EC40 ~ EC7F	EC80 ~ ECBF	ECC0 ~ ECFE	ED00 ~ ED3F	• •	EF40 ~ EF7F
13	F000 ~ F03F	F040 ~ F07F	F080 ~ F0BF	F0C0 ~ F0FF	F100 ~ F13F	• •	F340 ~ F37F
14	F400 ~ F43F	F440 ~ F47F	F480 ~ F4BF	F4C0 ~ F4FF	F500 ~ F53F	• •	F740 ~ F77F
15	F800 ~ F83F	F840 ~ F87F	F880 ~ F8BF	F8C0 ~ F8FF	F900 ~ F93F	• •	FB40 ~ FB7F
16	FC00 ~ FC3F	FC40 ~ FC7F	FC80 ~ FCBF	FCC0 ~ FCFE	FD00 ~ FD3F	• •	FF40 ~ FF7F

↑
模块编号偏置

表 5.5 运动参数寄存器编号（MP2000 系列时）

轴编号 电路编号	1 轴	2 轴	3 轴	4 轴	5 轴	• •	16 轴
1	8000 ~ 807F	8080 ~ 80FF	8100 ~ 817F	8180 ~ 81FF	8200 ~ 827F	• •	8780 ~ 87FF
2	8800 ~ 887F	8880 ~ 88FF	8900 ~ 897F	8980 ~ 89FF	8A00 ~ 8A7F	• •	8F80 ~ 8FFF
3	9000 ~ 907F	9080 ~ 90FF	9100 ~ 917F	9180 ~ 91FF	9200 ~ 9A7F	• •	9780 ~ 97FF
4	9800 ~ 987F	9880 ~ 98FF	9900 ~ 997F	9980 ~ 99FF	9A00 ~ 997F	• •	9F80 ~ 9FFF
5	A000 ~ A07F	A080 ~ A0FF	A100 ~ A17F	A180 ~ A1FF	A200 ~ A27F	• •	A780 ~ A7FF
6	A800 ~ A87F	A880 ~ A8FF	A800 ~ A87F	A980 ~ A9FF	AA00 ~ AA7F	• •	AF80 ~ AFFF
7	B000 ~ B07F	B080 ~ B0FF	B100 ~ B17F	B180 ~ B1FF	B200 ~ B27F	• •	B780 ~ B7FF
8	B800 ~ B87F	B880 ~ B8FF	B900 ~ B97F	B980 ~ B9FF	BA00 ~ BA7F	• •	BF80 ~ BFFF
9	C000 ~ C07F	C080 ~ C0FF	C100 ~ C17F	C180 ~ C1FF	C200 ~ C27F	• •	C780 ~ C7FF
10	C800 ~ C87F	C880 ~ C8FF	C900 ~ C97F	C980 ~ C9FF	CA00 ~ CA7F	• •	CF80 ~ CFFF
11	D000 ~ D07F	D080 ~ D0FF	D100 ~ D17F	D180 ~ D1FF	D200 ~ D27F	• •	D780 ~ D7FF
12	D800 ~ D87F	D880 ~ D8FF	D900 ~ D97F	D980 ~ D9FF	DA00 ~ DA7F	• •	DF80 ~ DFFF
13	E000 ~ E07F	E080 ~ E0FF	E100 ~ E17F	E180 ~ E1FF	E200 ~ E27F	• •	E780 ~ E7FF
14	E800 ~ E87F	E880 ~ E8FF	E900 ~ E97F	E980 ~ E9FF	EA00 ~ A97F	• •	EF80 ~ EFFF
15	F000 ~ F07F	F080 ~ F0FF	F100 ~ F17F	F180 ~ F1FF	F200 ~ F27F	• •	F780 ~ F7FF
16	F800 ~ F87F	F880 ~ F8FF	F900 ~ F97F	F980 ~ F9FF	F900 ~ F97F	• •	FF80 ~ FFFF

↑
模块编号偏置



用下式计算不同伺服监视器参数轴的寄存器编号。

MP900 系列时

- 轴监视器参数寄存器地址 = IW0000 + 电路编号
偏置 + (轴 No. -1) × 40 (HEX)

MP2000 系列时

- 轴监视器参数寄存器地址 = IW0000 + 电路编号
偏置 + (轴 No. -1) × 80 (HEX)

(3) 程序举例

进行外部输入输出信号及伺服监视器参数的读出调用。



- 位指令
MB01000=IB0010 & IB00105;
- 整数指定
MW0100=IWC016;
- 长整数指定
ML0100=ILC022;

5.2.4 输出变量（O寄存器）

(1) 概要

是使用外部输出信号及伺服设定参数的变量。

(2) 详细说明

O寄存器的指令方法如下所示。

• MP910 时
OW0000 ~ OW13FF ... 外部输出信号
OWC000 ~ OWC77F ... 伺服设定参数
• MP920 时
OW0000 ~ OW13FF ... 外部输出信号
OWC000 ~ OWFF7F ... 伺服设定参数
• MP930 时
OW0000 ~ OW07FF ... 外部输出信号
OWC000 ~ OWC37F ... 伺服设定参数
• MP940 时
OW0000 ~ OW07FF ... 外部输入信号
OWC000 ~ OWF03F ... 伺服监视器参数
• MP2100、MP2300 时
OW0000 ~ OW7FFF ... 外部输入信号
OW8000 ~ OWFFFF ... 伺服监视器参数

(a) 外部输入信号的寄存器编号

根据在模块构成定义中设定的地址而定。但，MP930的MC350模块内局部输出固定为OW0000(0B00010 ~ 0B0001F 16点)

(b) 伺服设定参数寄存器编号

模块的种类不同控制轴数也不同。不同模块控制轴数及最大模块数如下表所示。

表 5.6 模块的控制轴数

模块名称		控制轴数 / 模块	最大模块数
MP910		14	2
MP920	SVA-01	4	15
	SVA-02	2	16
	SVB-01	14	16
	PO-01	4	16
MP930		14	1
MP940		1	1
MP2100		16	1
MP2300		16	1

表 5.7 运动参数寄存器编号 (MP900 系列时)

电路编号	1 轴	2 轴	3 轴	4 轴	5 轴	• •	14 轴
1	C000 ~ C03F	C040 ~ C07F	C080 ~ C0BF	C0C0 ~ C0FF	C100 ~ C13F	• •	C340 ~ C37F
2	C400 ~ C43F	C440 ~ C47F	C480 ~ C4BF	C4C0 ~ C4FF	C500 ~ C53F	• •	C740 ~ C77F
3	C800 ~ C83F	C840 ~ C87F	C880 ~ C8BF	C8C0 ~ C8FF	C900 ~ C93F	• •	CB40 ~ CB7F
4	CC00 ~ CC3F	CC40 ~ CC7F	CC80 ~ CCBF	CCC0 ~ CCFE	CD00 ~ CD3F	• •	CF40 ~ CF7F
5	D000 ~ D03F	D040 ~ D07F	D080 ~ D0BF	D0C0 ~ D0FF	D100 ~ D13F	• •	D340 ~ D37F
6	D400 ~ D43F	D440 ~ D47F	D480 ~ D4BF	D4C0 ~ D4FF	D500 ~ D53F	• •	D740 ~ D77F
7	D800 ~ D83F	D840 ~ D87F	D880 ~ D8BF	D8C0 ~ D8FF	D900 ~ D93F	• •	DB40 ~ DB7F
8	DC00 ~ DC3F	DC40 ~ DC7F	DC80 ~ DCBF	DCC0 ~ DCFE	DD00 ~ DD3F	• •	DF40 ~ DF7F
9	E000 ~ E03F	E040 ~ E07F	E080 ~ E0BF	E0C0 ~ E0FF	E100 ~ E13F	• •	E340 ~ E37F
10	E400 ~ E43F	E440 ~ E47F	E480 ~ E4BF	E4C0 ~ E4FF	E500 ~ E53F	• •	E740 ~ E77F
11	E800 ~ E83F	E840 ~ E87F	E880 ~ E8BF	E8C0 ~ E8FF	E900 ~ E93F	• •	EB40 ~ EB7F
12	EC00 ~ EC3F	EC40 ~ EC7F	EC80 ~ ECBF	ECC0 ~ ECFE	ED00 ~ ED3F	• •	EF40 ~ EF7F
13	F000 ~ F03F	F040 ~ F07F	F080 ~ F0BF	F0C0 ~ F0FF	F100 ~ F13F	• •	F340 ~ F37F
14	F400 ~ F43F	F440 ~ F47F	F480 ~ F4BF	F4C0 ~ F4FF	F500 ~ F53F	• •	F740 ~ F77F
15	F800 ~ F83F	F840 ~ F87F	F880 ~ F8BF	F8C0 ~ F8FF	F900 ~ F93F	• •	FB40 ~ FB7F
16	FC00 ~ FC3F	FC40 ~ FC7F	FC80 ~ FCBF	FCC0 ~ FCFE	FD00 ~ FD3F	• •	FF40 ~ FF7F

↑
模块编号偏置

表 5.8 运动参数寄存器编号 (MP2000 系列时)

电路编号	1 轴	2 轴	3 轴	4 轴	5 轴	• •	16 轴
1	8000 ~ 807F	8080 ~ 80FF	8100 ~ 817F	8180 ~ 81FF	8200 ~ 827F	• •	8780 ~ 87FF
2	8800 ~ 887F	8880 ~ 88FF	8900 ~ 897F	8980 ~ 89FF	8A00 ~ 8A7F	• •	8F80 ~ 8FFF
3	9000 ~ 907F	9080 ~ 90FF	9100 ~ 917F	9180 ~ 91FF	9200 ~ 9A7F	• •	9780 ~ 97FF
4	9800 ~ 987F	9880 ~ 98FF	9900 ~ 997F	9980 ~ 99FF	9A00 ~ 997F	• •	9F80 ~ 9FFF
5	A000 ~ A07F	A080 ~ A0FF	A100 ~ A17F	A180 ~ A1FF	A200 ~ A27F	• •	A780 ~ A7FF
6	A800 ~ A87F	A880 ~ A8FF	A800 ~ A87F	A980 ~ A9FF	AA00 ~ AA7F	• •	AF80 ~ AFFF
7	B000 ~ B07F	B080 ~ B0FF	B100 ~ B17F	B180 ~ B1FF	B200 ~ B27F	• •	B780 ~ B7FF
8	B800 ~ B87F	B880 ~ B8FF	B900 ~ B97F	B980 ~ B9FF	BA00 ~ BA7F	• •	BF80 ~ BFFF
9	C000 ~ C07F	C080 ~ C0FF	C100 ~ C17F	C180 ~ C1FF	C200 ~ C27F	• •	C780 ~ C7FF
10	C800 ~ C87F	C880 ~ C8FF	C900 ~ C97F	C980 ~ C9FF	CA00 ~ CA7F	• •	CF80 ~ CFFF
11	D000 ~ D07F	D080 ~ D0FF	D100 ~ D17F	D180 ~ D1FF	D200 ~ D27F	• •	D780 ~ D7FF
12	D800 ~ D87F	D880 ~ D8FF	D900 ~ D97F	D980 ~ D9FF	DA00 ~ DA7F	• •	DF80 ~ DFFF
13	E000 ~ E07F	E080 ~ E0FF	E100 ~ E17F	E180 ~ E1FF	E200 ~ E27F	• •	E780 ~ E7FF
14	E800 ~ E87F	E880 ~ E8FF	E900 ~ E97F	E980 ~ E9FF	EA00 ~ A97F	• •	EF80 ~ EFFF
15	F000 ~ F07F	F080 ~ F0FF	F100 ~ F17F	F180 ~ F1FF	F200 ~ F27F	• •	F780 ~ EFFF
16	F800 ~ F87F	F880 ~ F8FF	F900 ~ F97F	F980 ~ F9FF	F900 ~ F97F	• •	FF80 ~ FFFF

↑
模块编号偏置

补充

用下式计算不同伺服设定参数轴的寄存器编号。

MP900 系列时

- 轴设定参数寄存器地址 = $O(I)W0000$ + 模块编号
偏置 + (轴 No. -1) × 40 (HEX)

MP2000 系列时

- 轴监视器参数寄存器地址 = $O(I)W0000$ + 模块编号
偏置 + (轴 No. -1) × 80 (HEX)

(3) 程序举例

在轴移动命令中用变量指令位置、速度。

- 位指令
 $OB01000=MB001000 \& IB00100;$
- 整数指定
 $OWC020=MW00100;$
- 长整数指定
 $OLC022=ML00100 + ML00200;$
- 实数指定
 $OF00200=MF00100 + MF00102;$

5.2.5 C 变量 (C 寄存器)

(1) 概要

C 寄存器是只能调用程序中根据参数表制作的数据的变量。不能进行数据写入。

(2) 详细说明

C 寄存器的指定方法如下所示。

CW00000 ~ CW4095

C 寄存器不能通过程序写入。

(3) 程序举例

在运算中使用变量时



- 位指令
MB001000=CB001001;
- 整数指定
MW00100=CW00100;
- 长整数指定
ML00100=CL00100;
- 实数指定
MF00100=CF00100;

5.2.6 D变量(D寄存器)

(1) 概要

是在各运动程序内固有的内部寄存器，仅相应程序可使用的变量。

(2) 详细说明

D寄存器的指定方法如下所示。

DW00000 ~ DW16383(最大)

上述变量可以在各种运算中作为“变量”使用并赋给它运算结果，或可用“变量”来指定定位的坐标值和速度。变量编号指定为10进制数表述。

大小由程序构成定义(属性)设定。(默认32WORD)

(3) 程序举例

(a) 在轴移动命令中，用变量指令位置、速度时



- 参数 指令单位 = mm
小数点位置 = 3 时

DL00100=100000;

DL00102=200000;

DL00104=300000;

DL00106=500000;

MVS [X]DL00100 [Y]DL00102 [Z]DL00104 FDL00106;

→ 1000.000 mm
→ 2000.000 mm
→ 3000.000 mm
→ 5000.000 mm/min

(b) 在运算中使用变量时



- 位指定
DB001000=IB001001 & MB000101;
- 整数指定
DW00102=CW00103 | DW00104 & DW00105;
- 长整数指定
DL00106=DL00108*ML0011/ML00200;
- 实数指定
DF00200=MF00202*DF00202*3.14;

重要

在以下运动命令中，用变量指定移动量坐标值或速度时，必须使用长整型。

MOV, MVS, MCW/MCC, ZRN, SKP, MVT, EXM, POS,
ACC, SCC, IAC, IDC, IFP, FMX, INP

附录

附录 A 运动命令一览	附录 -2
A.1 运动命令一览	附录 -2
附录 B 运动程序警报	附录 -7
B.1 运动程序警报的存储地点	附录 -7
B.2 运动程序警报一览	附录 -9

附录 A. 运动命令一览

本节对运动命令进行说明。

A. 1. 运动命令一览

运动命令一览如下表所示。

表 A. 1 运动命令一览

分类	命令	名称	指令格式	功能 / 意义
轴移动命令	MOV	定位	MOV [axis1] — [axis2] — … ; (最多可指定 14 轴)	根据快进给最多同时对 14 轴* 进行定位。
	MVS	直线插补	MVS [axis1] — [axis2] — … F — ; (最多可指定 14 轴)	根据插补进给速度 F 最多同时对 14 轴* 进行直线移动。
	MCW MCC	圆弧插补 (顺时针旋转) (逆时针旋转)	MCW [axis1] — [axis2] — R — F — ;MCC [axis1] — [axis2] — U — V — T — F — ;	同时对 2 轴, 沿着半径为 R(或指定圆心坐标) 的圆弧用切线速度 F 进行插补。 指定中心坐标时, 可用 T — 指定多圈。 (T — 也可省略)
	MCW MCC	螺旋插补 (顺时针旋转) (逆时针旋转)	MCW [axis1] — [axis2] — U — V — [axis3] — T — F — ; MCC [axis1] — [axis2] — R — [axis3] — F — ;	是圆弧插补和圆弧插补平面外的直线插补合成的, 同时以 3 轴移动。速度 F 为圆弧插补的切线速度。 指定圆心坐标时, 可用 T — 指定圈数。 (T — 也可省略)
	ZRN	原点复归	ZRN [axis1] — [axis2] — … ; (最多可指定 14 轴)	复归到各轴的原点。
	SKP	跳过命令	SKP [axis1] — [axis2] — …SS-T-; (最多可指定 14 轴)	在直线插补动作中“跳过信号”为 ON 时, 跳过剩余的移动量后进入下面的块。
	MVT	时间指定定位	MVT [axis1] — [axis2] — … T — ; (最多可指定 14 轴)	为了在指定时间内结束移动, 固定进给速度后进行定位。
	EXM	外部定位	EXM [axis1] — D — ;	在定位执行时, 输入外部定位信号后, 仅按用“D —”指定的移动增量值进行定位后执行下一命令。

* 为 MP930 的示例。机型不同最多轴数也不同, 详情请参阅“1. 1. 2 功能一览”。

表 A.1 运动命令一览 (续)

分类	命令	名称	指令格式	功能 / 意义
基本控制命令	ABS	绝对值模式	ABS;	把以后的坐标符号作为绝对值 (ABS 值) 处理。
	INC	增量值模式	INC;	把以后的坐标符号作为增量值 (INC 值) 处理。
	POS	当前值变更	POS [axis1] - [axis2] - ...;	最多同时把 14 轴 * 的当前值变更为“想变更的坐标值”。以后的移动指令在新坐标系上移动。
	PLN	坐标平面指定	PLN [axis1] [axis2]	有必要指定平面时指定坐标平面。
	MVM	机械坐标指令	MVM MOV [axis1] - [axis2] - ; 或 MVM MVS [axis1] - [axis2] - ;	想使轴在“机械坐标系”上移动时进行指令。将原点复归结束时自动设定的坐标系称为机械坐标系。不受 POS 命令影响的坐标系。
	PLD	程序当前位置更新	PLD [axis1] - [axis2] - ...;	手动更新等移动了的程序当前位置。最多可指定 14 轴。
速度与加减速命令	ACC	加速时间变更	ACC [axis1] - [axis2] - ... ;	最多可同时设定 14 轴 * 的直线加减速的加速时间。
	SCC	S 字时间参数变更	SCC [axis1] - [axis2] - ... ;	最多可同时设定 14 轴 * 的移动平均加减速的时间参数。
	VEL	进给速度变更	VEL [axis1] - [axis2] - ... ;	最多可同时设定 14 轴 * 的进给速度。
	IAC	插补加速时间变更	IAC T - ;	设定插补移动时的直线加减速的加速时间。
	IDC	插补减速时间变更	IDC T - ;	设定插补移动时的直线加减速的减速时间。
	IFP	插补进给速度比率设定	IFP P - ;	用最高速度的百分比进行插补进给时的速度指定。
	FMX	插补进给最高速度设定	FMX T - ;	设定插补进给时的最高速度。插补加速时间为从 0 开始达到该速度的时间。
高级控制命令	PFN	入位检查	MVS [axis1] - [axis2] - ... PFN; 或 PFN [axis1] [axis2];	同一块内或前一块的插补移动指令进入定位结束范围 (参数设定) 后, 再进入下面的块。
	INP	第 2 入位设定	INP [axis1] - [axis2] - ...;	之后指令的带 PFN 的插补移动指令进入“第 2 定位结束范围”后, 再进入下面的块。
	SNG	单块忽视	SNG MVS [axis1]100. [axis2]200. F1000;	有该命令的块即使为“单块运行模式”也可忽视并进行连续运行。不能单独用 SNG 命令进行指定。
	UFC	用户函数调用	UFC 用户函数名 输入数据, 输入地址, 输出数据;	调用用户制作的函数。

* 为 MP930 的示例。机型不同最多轴数也不同, 详情请参阅“1.1.2 功能一览”。

表 A.1 运动命令一览 (续)

分类	命令	名称	指令格式	功能 / 意义
顺序命令	=	赋值	(结果) = (运算表达式)	将运算结果赋给左侧。另外, 运算按从左到右的顺序进行(无优先顺序)。
	+	加法运算	MW - = MW - + MW - ; MW - = MW - + 123456; MW - = 123456 + MW - ;	进行整数 / 实数的加法运算。整数 / 实数混合时, 按实数进行运算。
	-	减法运算	MW - = MW - - MW - ; MW - = MW - -123456; MW - = 123456 - MW - ;	进行整数 / 实数的减法运算。整数 / 实数混合时, 按实数进行运算。
	*	乘法运算	MW - = MW - * MW - ; MW - = MW - * 123456; MW - = 123456 * MW - ;	进行整数 / 实数的乘法运算。整数 / 实数混合时, 按实数进行运算。
	/	除法运算	MW - = MW - / MW - ; MW - = MW - / 123456; MW - = 123456 / MW - ;	进行整数 / 实数的除法运算。整数 / 实数混合时, 按实数进行运算。
	MOD	余数运算	MW - = MW - / MW - ; MW - = MOD;	在除法运算的下面的块中指令, 则把 MOD 作为余数存储在指定寄存器中。
		OR(逻辑或)	MB - = MB - MB - ; MB - = MB - 1; MW - = MW - MW - ; MW - = MW - HOOFF;	计算位 / 整数的逻辑或。
	^	XOR (逻辑异或)	MW - = MW - ^ MW - ; MW - = MW - ^ HOOFF;	计算整数的逻辑异或。
	&	AND(逻辑与)	MB - = MB - & MB - ; MB - = MB - & 1; MW - = MW - & MW - ; MW - = MW - & HOOFF;	计算位 / 整数的逻辑与。
	!	NOT(取反)	MB - = !MB - ; MB - = !1; MW - = !MW - ; MW - = !HOOFF;	计算对位取反的值。
	()	括号	MW - = MW - & (MW - MW -);	先对括号内的逻辑运算表达式进行运算。
	S{}	指定位 ON	S{MB - } = MB - & MB - ;	若逻辑运算结果为“真”, 则将指定位置为 ON。即使逻辑运算结果为“假”, 也不将指定位置为 OFF。
	R{}	指定位 OFF	R{MB - } = MB - & MB - ;	若逻辑运算结果为“真”, 则将指定位置为 OFF。即使逻辑运算结果为“假”, 也不将指定位置为 ON。
	SIN	正弦	SIN(MW -); SIN(90);	求整数 / 实数(deg)的正弦, 返回实数值。
	COS	余弦	COS(MW -); COS(90);	求整数 / 实数(deg)的余弦, 返回实数值。
TAN	正切	TAN(MF -); TAN(45.0);	求实数(deg)的正切, 返回实数值。	
ASN	反正弦	ASN(MF -); ASN(90.0);	求实数的反正弦, 返回实数值(deg)。	

表 A.1 运动命令一览 (续)

分类	命令	名称	指令格式	功能 / 意义
顺序命令 (续)	ACS	反余弦	ACS(MF -) ; ACS(90.0) ;	求实数的反余弦, 返回实数值 (deg)。
	ATN	反正切	ATN(MW -) ; ATN(45) ;	求整数 / 实数的反正切, 返回实数值 (deg)。
	SQRT	平方根	SQT(MW -) ; SQT(100) ;	求整数 / 实数的平方根, 返回实数值。
	BIN	BCD → BIN	BIN (MW -) ;	把 BCD 数据转换成 BIN 数据。
	BCD	BIN → BCD	BCD (MW -) ;	把 BIN 数据转换成 BCD 数据。
	==	等于	IF MW - == MW - ; WHILE MW - == MW - ;	用于 IF 或 WHILE 条件表达式。左边和右边相等时为“真”。
	◇	不等于	IF MW - ◇ MW - ; WHILE MW - ◇ MW - ;	用于 IF 或 WHILE 条件表达式。左边和右边不相等时为“真”。
	>	大于	IF MW - > MW - ; WHILE MW - > MW - ;	用于 IF 或 WHILE 条件表达式。左边比右边大时, 为“真”。
	<	小于	IF MW - < MW - ; WHILE MW - < MW - ;	用于 IF 或 WHILE 条件表达式。左边比右边小时, 为“真”。
	>=	大于等于	IF MW - >= MW - ; WHILE MW - >= MW - ;	用于 IF 或 WHILE 条件表达式。左边大于等于右边时, 为“真”。
	<=	小于等于	IF MW - <= MW - ; WHILE MW - <= MW - ;	用于 IF 或 WHILE 条件表达式。左边小于等于右边时, 为“真”。
	SFR	右移	SFR MB - N - W - ;	仅按指定数向右移动字变量。
	SFL	左移	SFL MB - N - W - ;	仅按指定数向左移动字变量。
	BLK	块传送	BLK MW - MW - W - ;	从指定位 (字) 变量开始, 传送相应的块 (参数指定)。
CLR	清除	CLR MB - W - ;	把以指定位 (字) 变量为开始的变量群置为相应参数指定 OFF(0)。	

表 A.1 运动命令一览 (续)

分类	命令	名称	指令格式	功能 / 意义
控制命令	MSEE	子程序调用	MSEE MPS - ;	执行 MPS - 子程序。
	TIM	待时	TIM T - ;	等待相应的由 T 指定的时间后, 进入下面的块。
	IOW	输入输出变量待机	IOW MB - == ***;	到满足左边条件表达式为止, 停止运动程序的执行。
	END	程序结束	END;	结束运动程序
	RET	子程序结束	RET;	结束子程序。
	EOX	1 个扫描周期的 WAIT(待机) 命令	EOX;	是分割连续的顺序命令, 使其在 1 个扫描周期 WAIT(待机) 的命令。
	IF ELSE IEND	分支命令	IF (条件表达式); (处理 1) ELSE; (处理 2) IEND;	满足条件表达式时, 执行 (处理 1); 不满足条件表达式时, 执行 (处理 2)。
	WHILE WEND	循环命令	WHILE (条件表达式); ... WEND;	在持续满足条件表达式期间, 循环执行 WHILE ~ WEND 的处理。
	PFORK JOINTO PJOINT	并列执行命令	PFORK 标签 1, 标签 2, ...; 标签 1: 处理 1 JOINTO 标签 X 标签 2: 处理 2 JOINTO 标签 X 标签 . . 标签 X: PJOINT;	并列执行由标签指定的块。子程序最多只能指定 2 个标签。另外, 在用第 2 个标签指定的块中不能使用运动命令。 在并列执行处理时, 不能使用 END、RET。
SFORK JOINTO SJOINT	选择执行命令	SFORK 条件式 1? 标签 1, 条件式 2, 标签 2, ...; 标签 1: 处理 1 JOINTO 标签 X 标签 2: 处理 2 JOINTO 标签 X 标签 . . 标签 X: SJOINT;	满足条件表达式 1 时, 执行处理 1; 满足条件表达式 2 时, 执行处理 2。	

附录 B. 运动程序警报

本节对运动程序错误一览进行说明。

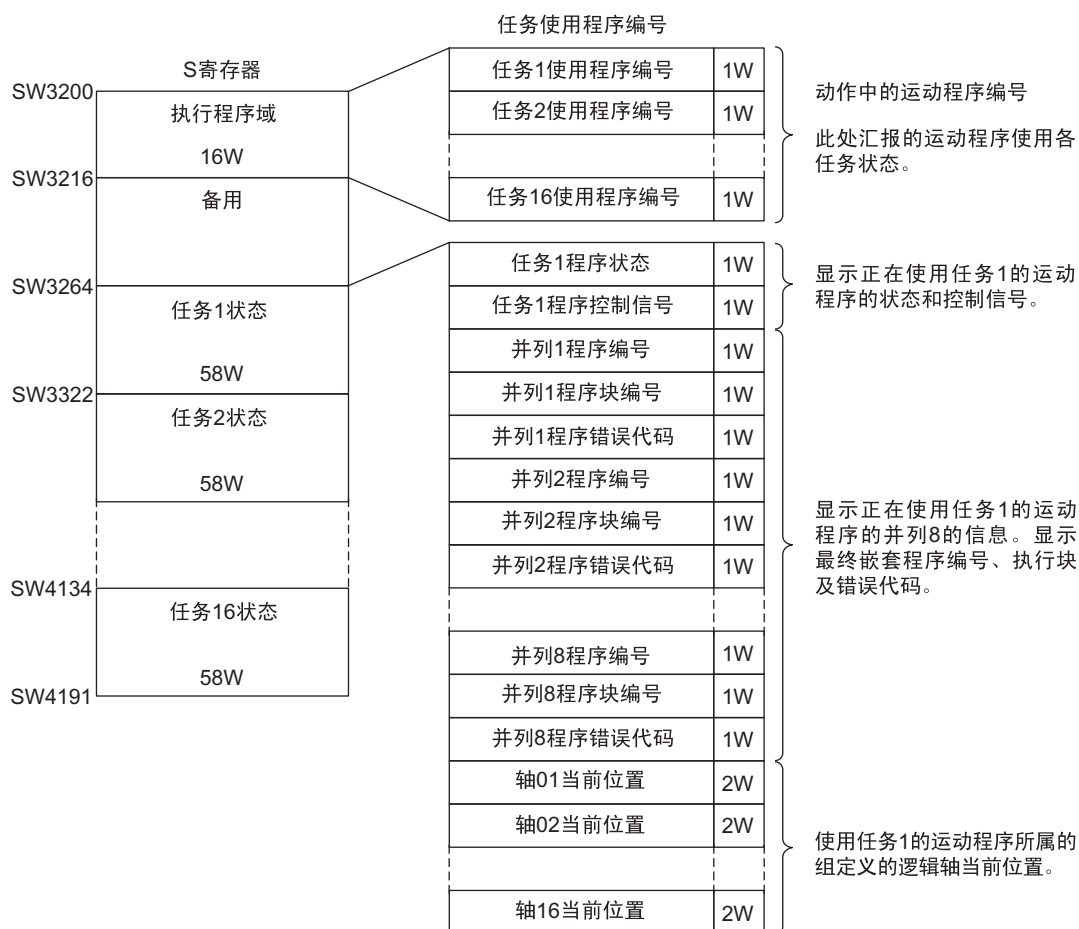
B. 1. 运动程序警报的存储地点

(1) MP900 系列时

MP900 系列时，在用组定义设定的警报输出寄存器中显示运动程序警报。按属于组的运动程序使用的并列处理数确保连续的警报输出寄存器。例如，定义警报输出寄存器为 MW00010，当并列处理数为 4 时，使用按并列编号升序的从 MW00010 到 MW00013 的输出寄存器。

(2) MP2000 系列时

MP2000 系列中，向 S 寄存器汇报运动程序的状态。在运动程序中使用的 S 寄存器域和任务状态的详情如下所示。

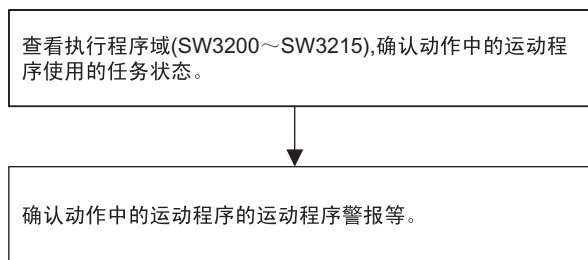


(a) 运动程序使用的任务状态

运动主程序在动作中使用一个任务状态。任务状态用来显示运动程序的运动程序警报和逻辑轴的当前位置。

任务状态有以下 2 种使用方法。

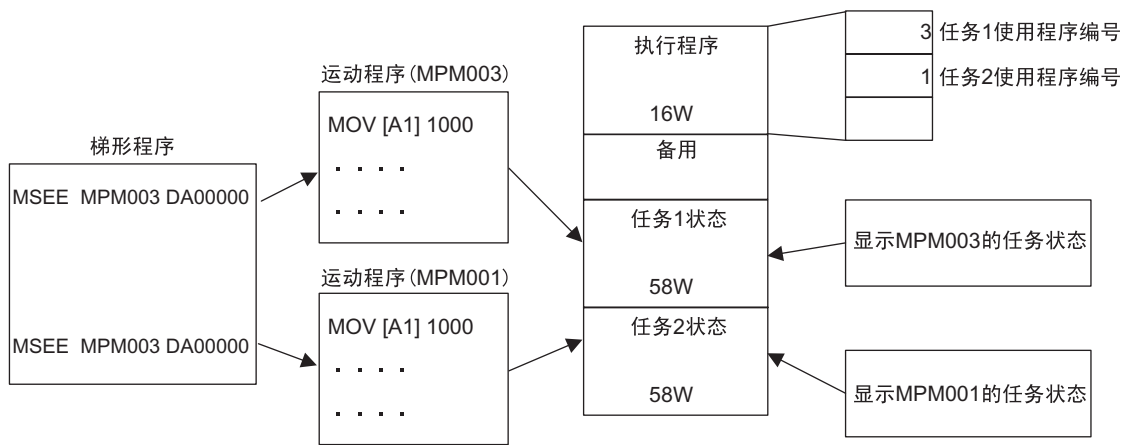
- 指定任务状态域时
要在运动程序中指定使用的任务状态时，请将 MSEE 命令的任务寄存器的第 2 个字的 Bit13 “系统编号设定” 置为 ON。由此，运动程序可以使用由 MSEE 命令的任务寄存器的第 4 个字的 “系统任务编号” 设定的任务状态。但是，指定的系统任务编号在规定范围外或正在使用时，任务寄存器的第 1 个字的 Bit14 “无系统任务错误” 将为 ON。
- 不指定任务状态域时
按照运动程序启动的顺序，从任务 1 状态开始使用任务状态。在同一扫描周期中启动多个运动程序时，按 MSEE 命令调用顺序使用任务状态。运动程序结束的同时，任务状态域被释放。想查看运动程序的运动程序警报和逻辑轴的当前位置时，请进行以下作业。



(b) 执行多个运动程序时



为在运动程序 MPM003 启动中，启动运动程序 MPM001 的示例。



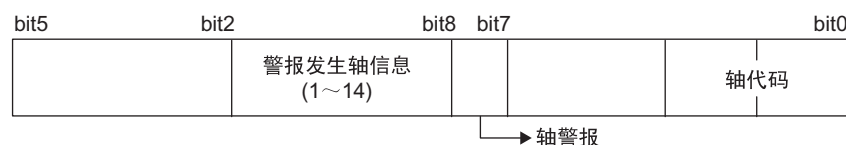
启动 MPM003 时，向执行程序域的任务 1 使用程序编号 (SW3200) 汇报 “3”，MPM003 使用任务 1 状态域。通过查看任务 1 状态的信息，可以确认运动程序警报和逻辑轴的当前位置信息。

在 MPM003 启动中，启动 MPM001 时，向执行程序域的任务 2 使用程序编号 (SW3201) 汇报 “1”，MPM001 使用任务 2 状态域。通过查看任务 2 状态的信息，可以确认运动程序警报和逻辑轴的当前位置信息。

B. 2. 运动程序警报一览

运动程序警报为 16 进制数，如下所示。

但是，请设为 HEX (H) 显示模式。



运动程序的警报代码和处理方法如下表所示。

表 B. 2 运动程序错误一览

警报代码	警报名称	警报内容	处理方法
02h	除法运算错误	进行了用数据除以 0 的运算。	修改运动程序。
10h	半径指定的 1 圈指定警报	在根据半径的圆弧插补命令、螺旋插补命令中，指定圈数 (T)。	<ul style="list-style-type: none"> 改半径指定为圆心坐标指定，进行圆弧插补命令、螺旋插补命令。 不指定圈数。
11h	插补进给速度超速	设定的插补进给速度指定超过 FMX 命令的指定范围。	重新设定插补命令的插补进给速度指定。
12h	无插补进给速度指定	从未进行插补进给速度指定。(一旦设定后，在同一运动程序内可省略。)	在插补命令中进行插补进给速度指定。
13h	加速度参数转换后超范围	通过间接指定设定的加速度参数超过设定范围。	改变进行间接指定的寄存器值。
14h	圆弧长超过 LONG_MAX。	在圆弧插补命令、螺旋插补命令中圆弧长指定超过设定范围。	在圆弧插补命令、螺旋插补命令中修改圆弧长的指定。
15h	无圆弧平面指定的纵轴指定	在圆弧插补命令、螺旋插补命令中没有进行纵轴指定。	根据 PLN 命令指定轴。
16h	无圆弧平面指定的横轴指定	在圆弧插补命令、螺旋插补命令中没有进行横轴指定。	根据 PLN 命令指定轴。
17h	超过指定轴数	在圆弧插补命令 (2 轴)、螺旋插补命令 (3 轴) 中，设定了过多的轴数。	重新设定圆弧插补命令、螺旋插补命令的轴指定。
18h	超过指定圈数	在圆弧插补命令、螺旋插补命令中设定的圈数指定超过设定范围。	重新设定圆弧插补命令、螺旋插补命令的圈数指定。
19h	半径超过 LONG_MAX	在圆弧插补命令、螺旋插补命令中半径指定超过设定范围。	在圆弧插补命令、螺旋插补命令中修改半径指定。
1Ah	圆心指定错误	在圆弧插补命令、螺旋插补命令中，没有进行正确的圆心指定。	在圆弧插补命令、螺旋插补命令中指定正确的圆心。

表 B. 2 运动程序错误一览 (续)

警报代码	警报名称	警报内容	处理方法
1Bh	紧急停止指令执行中	根据程序停止要求, 轴移动命令停止。	将运动程序控制信号的程序停止要求置为 OFF, 进行 MP930 的警报复位。
1Ch	直线插补移动量超过 LONG_MAX。	直线插补命令中, 移动量指定超过设定范围。	在直线插补命令中重新设定移动量指定。
1Dh	FMX 未定义	有插补类命令的运动程序中没有执行 FMX 命令。	执行 FMX 命令。在每个有插补类命令的程序中, FMX 命令是必要的。
1Eh	地址 T 在范围外	在 IAC/IDC/FMX 命令中, 执行超过设定范围的指定。	在 IAC/IDC/FMX 命令中, 进行重新设定。
1Fh	地址 P 在范围外	在 IFP 命令中, 执行超过设定范围的指定。	在 IFP 命令中, 进行重新设定。
80h	逻辑轴使用禁止中	固定参数的轴使用选择为“未使用选择”。	将固定参数的轴使用选择置为“使用选择”。
81h	由无限长轴指定进行了超过 POSMAX 的指定。	由无限长轴指定进行了超过设定 POSMAX 的移动距离指定。	<ul style="list-style-type: none"> 重新设定固定参数的“无限长计数器最大值”。 修改运动程序。
82h	轴移动距离超过 LONG_MAX。	轴移动距离指定超过设定范围。	修改运动程序。
84h	运动命令重复	对 1 个轴, 执行多个命令。	修改梯形程序。
85h	运动命令响应重复	错误响应返回到运动命令。	修改梯形程序。
87h	VEL 设定数据	在 VEL 命令中执行了超过设定范围的命令。	修改 VEL 命令。
88h	INP 设定数据	在 INP 命令中执行了超过设定范围的命令。	修改 INP 命令。
89h	ACC/SCC/DCC 设定数据在范围外	在 ACC/SCC/DCC 命令中执行了超过设定范围的命令。	修改 ACC/SCC/DCC 命令。
8Ah	MVT 命令中没有时间指定。	MVT 命令中 T 指定为 0。	修改 MVT 命令。

索引

- Symbols**
- # 寄存器 3-10
- Numerics**
- 10 进制小数 1-29
- 10 进制整数 1-29
- 16 进制整数 1-29
- 1 个扫描周期的 WAIT(待机) 命令 (EOX) 3-18
- A**
- ABS / INC 2-3
- B**
- BCD → BIN(BIN) 4-28
- BIN → BCD(BCD) 4-29
- 变量 1-29
- 使用方法 - - - - - 5-7
- 变量(寄存器)种类 5-2
- 标签 1-25
- 并行执行命令 (PFORK, JOINT0, PJOINT) 3-22
- C**
- C 变量 (C 寄存器) 5-15
- 参数寄存器 3-10
- 插补加速时间变更 (IAC) 3-39
- 插补减速时间变更 (IDC) 3-41
- 插补进给速度 1-32
- 插补进给速度比率设定 (IFP) 3-36
- 插补进给最高速度设定 (FMX) 3-37
- 常量 1-29
- 乘法运算 (*) 4-8
- 程序 1-7
- 程序编号处理 1-24
- 程序当前位置更新 (PLD) 2-30
- 程序的结束 (END) 2-32
- 除法运算 (/) 4-9
- D**
- D 变量 (D 寄存器) 5-16
- D 寄存器 3-10
- 待时 (TIM) 2-31
- 单块忽视 (SNG) 3-6
- 单块运行模式 3-6
- 单块指令的制作方法 1-25
- 当前值变更 (POS) 2-26
- 第 2 入位检查 (INP) 3-4
- 定位 (MOV) 2-2
- 读零 1-23
- E**
- 额定速度比 1-31
- F**
- 反余弦 (ACS) 4-24
- 反正切 (ATN) 4-25
- 反正弦 (ASN) 4-23
- 分支命令 (IF ELSE IEND) 3-19
- 符号 1-23
- 赋值 (=) 4-5
- G**
- 功能字符 1-24
- H**
- 函数内部寄存器 3-10
- 函数输出寄存器 3-10
- 函数输入寄存器 3-10
- 函数外部寄存器 3-10
- J**
- 基本函数 1-30, 4-3, 4-20
- 机械坐标指令 (MVM) 2-29
- 加法运算 (+) 4-6
- 加速时间变更 (ACC) 3-28, 3-30
- 减法运算 (-) 4-7
- 进给速度 1-31
- 进给速度变更 (VEL) 3-34
- 局部变量 1-29, 5-5
- 绝对值 (ABS) 模式 2-23
- K**
- 可使用字符 1-23
- 控制规格 1-7
- 控制命令 2-23
- 控制轴 1-26
- 控制轴数 1-7
- 快进给速度 1-31
- 块传送 (BLK) 4-18
- L**
- 逻辑或 (|) 4-11
- 逻辑异或 (^) 4-13
- 逻辑与 (&) 4-12
- 逻辑运算 1-30, 4-3, 4-11
- 逻辑运算组合 4-4
- 螺旋插补 (MCW、MCC) 2-13
- M**
- MP910 的运动控制功能规格 1-4
- MP920 的运动控制功能规格 1-5
- MP930 的运动控制功能规格 1-6
- 命令符号 1-8
- P**
- 平方根 (SQT) 4-26
- Q**
- 清除 (CLR) 4-19
- 取反 (!) 4-14
- 全局变量 1-29, 5-4
- R**
- 入位检查 (PFN) 3-2
- S**
- S 字时间参数变更 (SCC) 3-32
- 时间指定定位 (MVT) 2-20
- 实数 1-29
- 输出变量 (O 寄存器) 5-12
- 输出寄存器 3-10
- 数据变量 (M 寄存器) 5-8
- 数据操作 1-30, 4-3, 4-16
- 数据寄存器 3-10
- 数据类型 5-3
- 输入变量 (I 寄存器) 5-9
- 输入格式 1-23
- 输入寄存器 3-10
- 输入输出变量待机 (IOW) 3-15

数值比较.....	1-30, 4-3, 4-15
数值比较命令.....	4-15
数值和变量的表达方法.....	1-29
数值运算.....	1-30, 4-3, 4-5
数值运算组合.....	4-4
顺序命令概要.....	4-3
速度控制.....	1-7

T

跳过功能 (SKP)	2-19
同步控制轴数.....	1-27

W

外部定位 (EXM)	2-22
位右移 (SFR)	4-16
位置控制.....	1-7
位左移 (SFL)	4-17

X

系统变量 (S 寄存器)	5-7
系统寄存器.....	3-10
相位控制.....	1-8
选择执行命令 (SFORK, JOINTO, SJOINT)	3-26
循环命令 (WHILE WEND)	3-20

Y

用户函数调用 (UFC)	3-7
余数 (MOD)	4-10
余弦 (COS)	4-21
原点复归.....	1-7
原点复归 (ZRN)	2-15
圆弧插补 (MCW、MCC)	2-8
运动程序	
编辑 - - - - -	1-21
并列执行 - - - - -	1-20
的作用 - - - - -	1-2
示例 - - - - -	1-22
执行处理方法 - - - - -	1-15
运动命令一览.....	1-34
运算表达式及函数.....	1-30
运算命令.....	4-3

Z

增量值 (INC) 模式.....	2-25
正切 (TAN)	4-22
正弦 (SIN)	4-20
指定位 OFF(R{ })	4-31
指定位 ON(S{ })	4-30
指令单位.....	1-27
直线插补 (MVS)	2-5
轴名称.....	1-26
轴移动命令.....	2-2
注释表述.....	1-24
转矩控制.....	1-7
子程序调用 (MSEE)	3-16
子程序结束 (RET)	3-17
最大指令值.....	1-27
坐标符号一览.....	1-26
坐标平面指定 (PLN)	2-28
坐标系.....	1-7

改版履历

关于资料的改版信息与资料编号一同记载于本资料封底的右下部。

资料编号 SICP C880700 19A

© Published in XXXX 2012年 4月编制 04-11 ◇

└─ 国家或地区 ─┘ └─ 第一版发行日期 ─┘ └─ 改版编号 ─┘

└─ 发行日期 ─┘

印刷年 / 月	改版编号	项目编号	变更部分
2012年4月	◇	封面	变更：格式、厂标
		封底	变更：格式、地址、厂标
		背脊	变更：厂标
2004年11月	-	-	第1版发行