



运动控制器

P

运动控制器 Q 系列 (Q173CPU(N)/Q172CPU(N)) 编程手册 SV13/SV22(运动 SFC)

Q

Q 系列运动控制器
SV13/SV22(运动 SFC)编程手册
(Q173CPU(N)/Q172CPU(N))

运动控制器

Q 系列

SV13/SV22(运动 SFC)

Q173CPU(N)

Q172CPU(N)

编程手册

三菱电机自动化(上海)有限公司

地址: 上海漕宝路 103 号自动化仪表城 5 号楼 1~3 层
邮编: 200233
电话: 021-61200808 传真: 021-61212444
网址: www.mitsubishielectric-automation.cn

书号	IB(NA)-0300042CHN(0411)
印号	MEAS-Qmotion-P-SFC(0411)

内容如有变更, 恕不另行通知

● 安全注意事项 ●

(使用前请阅读这些安全注意事项)

在使用该设备时,请仔细阅读本使用手册以及手册中提及的其它相关手册。同时,也要注意安全防范以及正确操作模块。

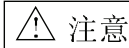
这些安全注意事项仅适用该设备。请阅读CPU模块用户使用手册中的PLC系统安全注意事项。

安全注意事项分为两类:“危险”和“注意”。



危险

说明不正确的操作可能导致危险情况发生,造成死亡或重伤害。



注意

说明不正确的操作可能导致危险情况发生,造成中度或轻度的人身伤害。

根据环境不同,标有△注意的程序也可能导致严重的结果。任何情况下,按照使用要求进行操作是非常重要的。

请将本手册妥善保存,以备不时之需。务必要将本手册送至最终用户手中。

安全操作

1. 防止触电

⚠ 危险

- 电源接通或模块仍在工作时，请勿打开前盖盒或端子盖，否则将可能导致触电。
- 在前盖盒或端子盖移开时，请勿运行模块。否则您可能会因为接触到裸露的高压端子或带电部分导致触电。
- 除了在配线或定期检查时，其它任何时候，即使在电源关闭时，也不要将前盖盒或端子盖打开。
- 在进行配线工作或检查时，将电源关闭至少十分钟后，用仪器检测电压。否则将可能导致触电。确保将运动控制器、伺服放大器以及伺服电机接地。(接地电阻：100 Ω 以下)请勿与其它设备共同接地。
- 必须由合格的技术师进行配线作业以及检查工作。
- 在安装好运动控制器、伺服放大器以及伺服电机之后，再为其它模块配线。否则将可能导致触电或危险情况发生。
- 湿手不可触碰开关，否则可能遭受触电。
- 应尽量避免电缆遭到破坏，夹压，高强度，重压，否则可能导致触电。
- 在电源接通时，不要触碰运动控制器、伺服放大器或伺服电机。否则可能导致触电。
- 不要触碰运动控制器和伺服放大器的内置电源、内置接地或信号线，否则可能导致触电。

2. 预防火灾

⚠ 注意

- 运动控制器、伺服放大器、伺服电机以及再生电阻要安装在不易燃烧的材料上，直接安装在易燃材料上或易燃材料附近都可能引发火灾。
- 如果运动控制器或伺服放大器发生故障，切断伺服放大器电源，否则，持续大电流可能引发火灾。
- 当使用再生电阻时，如果出现异常信号，切断电源，否则，由于再生制动晶体管故障可能出现异常过热现象，导致火灾的发生。
- 经常测量安装了伺服放大器或再生电阻的控制面板和使用电线的内部热度，疏于上述操作可能导致火灾的发生。

3. 预防损伤

⚠注意

- 各端子的输入电压要符合指定手册的标准，否则可能导致端子破坏或损伤。
- 端子接线要正确，否则这样的操作可能导致破坏或损伤。
- 正负极性(+/-)要正确，以免导致破坏或损伤。
- 当电源接通和断开后的一段时间内，不要触摸伺服放大器的散热片、再生电阻和伺服电机等。因为这些部分是非常热的，可能会导致烫伤。
- 在触摸伺服电机轴或连接机械前，始终要切断电源，因为这些部分可能导致造成损伤。
- 在整个测试操作或示范操作期间不要走近机器，因为这样做可能造成损伤。

4. 各种预防措施

严格遵守下列预防措施。

错误的模块操作可能导致故障发生、伤害或触电。

(1) 系统结构

⚠注意

- 务必在运动控制器和伺服放大器电源上安装漏电断路器。
- 若伺服放大器等产品的使用说明手册中规定须安装电磁接触器以便在出现故障等情况时断开电源，则必须安装它。
- 在外部安装紧急停止电路，这样遇到紧急事故能立即停止操作，切断电源。
- 按说明手册上所列的组合方法使用运动控制器、伺服放大器、伺服电机和再生电阻。其他类型的组合可能导致火灾或故障。
- 若要将安全标准（如机器人安全规则等）应用于使用运动控制器、伺服放大器、伺服电机的系统，必须确定该安全标准在系统中成立。
- 如果运动控制器或伺服放大器的异常操作不同于系统要求的安全指示操作，在外部构建一个安全电路。
- 在系统强制停止，紧急停止，伺服关闭或电源关闭期间，伺服电机的惯性可能造成问题时，使用动态制动。
- 即使使用动态制动，也要确定系统考虑了惯量。

⚠注意

- 在紧急停止、强制停止、伺服系统关闭或电源切断的情况期间，垂直轴的下滑可能造成问题，应使用动态制动和电磁制动。
- 动态制动措施必须只用于紧急停止、强制停止、伺服系统关闭产生错误的状况下，这些制动措施不允许被用于正常的制动。
- 装配在伺服电机上的刹车（电磁制动）同样不允许被用于正常的制动。
- 系统必须有一个机械范围，这样即使当最大转速通过行程限位开关时，机器能自行停止运转。
- 使用有合适的直径、热阻和抗弯性的电线和电缆，而且同系统能兼容。
- 使用的电线和电缆，其长度范围要遵照使用说明书的标准。
- 系统中使用的零部件（除了运动控制器、伺服放大器和伺服电机）的等级和特性必须能与运动控制器、伺服放大器和伺服电机兼容。
- 在轴上安装一个封盖，这样在操作过程中伺服电机中的旋转零部件不会被碰到。
- 这有一些情况，由于电磁制动使用寿命或机械结构（如在滚珠丝杆和伺服电机是同步连接情况下等等），导致电磁制动失灵，因此在机械部分上安装一个停止装置以确保安全。

② 参数设置和编程

⚠注意

- 设置与运动控制器、伺服放大器、伺服电机和再生电阻型号、系统用途相适应的参数。错误的设定可能导致保护功能失效。
- 再生电阻型号和容量参数值应符合操作模式、伺服放大器和伺服系统电源模块，错误的设定可能导致保护功能失效。
- 设置的电磁制动输出和动态制动输出有效参数值应与系统应用兼容，错误的设定可能导致保护功能失效。
- 设置的行程限位输入有效参数值应与系统应用兼容，错误的设定可能导致保护功能失效。

⚠注意

- 设置的伺服电机编码器类型（增量、绝对位置类型等）参数值应与系统应用兼容，错误的设定可能导致保护功能失效。
- 设置的伺服电机容量和类型（标准、低惯量、扁平等）参数值应与系统应用兼容，错误的设定可能导致保护功能失效。
- 设置的伺服放大器容量和类型参数值应与系统应用兼容，错误的设定可能导致保护功能失效。
- 在使用说明手册指定的条件下使用程序指令。
- PLC 程序容量设置参数值、软元件容量、锁存使用范围、I/O 分配设置和错误检测期间持续运行是否有效的设定应与系统应用兼容，错误的设定可能导致保护功能失效。
- 程序中使用的软元件有些有固定用途，请按使用说明书规定的条件使用。
- 当通讯被一个错误所终止时，输入设置和赋值到链接数据注册表将先于保持数据，这样使用说明手册上详细说明了的错误传输互锁程序必须被使用。
- 使用在专用功能模块使用说明手册上指定的互锁程序，这些程序对应专门的功能模块程序。

③ 搬运和安装

⚠注意

- 根据产品重量，用正确的方法来搬运产品。
- 在搬运伺服电机时，使用专用伺服电机吊栓。在搬运前请勿在电机上安装机器。
- 堆叠产品不能超过限制。
- 当搬运运动控制器或伺服放大器时，请勿拖拉连接的电线或电缆。
- 当搬运伺服电机，请勿拖拉电缆、轴或编码器。
- 当搬运运动控制器或伺服放大器时，请勿拿住前部，因为它可能坠落。
- 当搬运、安装或取下运动控制器或伺服放大器时，不能握住产品边缘部分。
- 根据产品指导说明书来安装这些模块，安装位置应能承受产品的重量。

⚠注意

- 请勿站在机器上，也请勿在机器上放重物。
- 要始终遵守产品安装指南。
- 保持运动控制器或伺服放大器和控制面板内部表面的清洁，保持运动控制器和伺服放大器、运动控制器或伺服放大器和其它设备清洁。
- 请勿安装或操作已经损坏的或缺少零部件的运动控制器、伺服放大器或伺服电机。
- 冷却风扇请勿阻塞伺服电机的进风口 / 出风口。
- 请勿让如螺丝钉或金属屑等导电性异物或油等易燃物质进入运动控制器、伺服放大器或伺服电机内。
- 运动控制器、伺服放大器和伺服电机是精密机器，因此请勿使其跌落或受强力冲击。
- 根据使用说明书要求，安全地固定运动控制器和伺服放大器到机器上，如果固定不牢靠，运行时会有脱落的危险。
- 带减速机的伺服电机要按指定方向安装。否则，可能会导致漏油。
- 在下列环境条件下储存和使用模块。

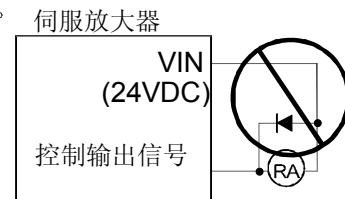
环境	条件	
	运动控制器/伺服放大器	伺服电机
环境温度	根据各个使用说明书	0°C 到+40°C (不结冰)
环境湿度	根据各个使用说明书	80% RH 以下 (不结露)
存储温度	根据各个使用说明书	-20°C 到+65°C
周围环境	室内 (不受阳光直晒) 无腐蚀性气体、可燃气体、油滴、或灰尘	
海拔高度	海拔 1000m 以下	
振动	根据各个使用说明书	

- 当连接同步编码器或伺服电机轴末端时，请勿使用铁锤等重物进行敲打，以免导致编码器损坏。
- 请勿在伺服电机轴上应用超过其承受能力的负载，否则可能导致轴损坏。
- 当长时间不使用模块时，则切断与运动控制器或伺服放大器连接的电源线。
- 将运动控制器和伺服放大器放置在防静电的乙烯塑料包中储存。
- 当长时间储存时，请联系我们的销售代表。

④ 接线

⚠ 注意

- 正确安全地接线, 再次确认连接是否正确, 配线后固定螺丝是否紧固。否则, 可能导致伺服电机脱落。
- 接线后, 把封盖安装回原来位置。
- 请勿在伺服放大器输出端口安装功率电容器、浪涌吸收器或无线噪声滤波器(选件FR-BIF)。
- 正确的连接输出端(U, V, W 端), 否则将导致伺服电机不能正常运行。
- 请勿将伺服电机连接到商用电源上, 这样可能导致故障。
- 请勿弄错安装在 DC 继电器上浪涌吸收二极管的方向, 该二极管是用于控制制动信号输出。不正确的安装可能导致当错误发生时信号不能被输出或保护功能无效。
- 当电源连接时, 不要连接或断开各单元之间的连接电缆, 编码器电缆或 PLC 扩展电缆。
- 安全地固定电缆接头固定用螺丝钉和机械装置, 否则可能导致电缆线在运行过程中脱落。
- 请勿捆扎电源线或电缆。



⑤ 试运行和调整

⚠ 注意

- 在运行前确认调整程序和每一个参数, 运行时可能出现无法预知的情况。
- 极端地调整和改变可能导致不稳定地运行。
- 在开始时使用绝对位置系统功能、或更换运动控制器或绝对值电机时, 要执行原点回归。

6) 使用方法

⚠注意

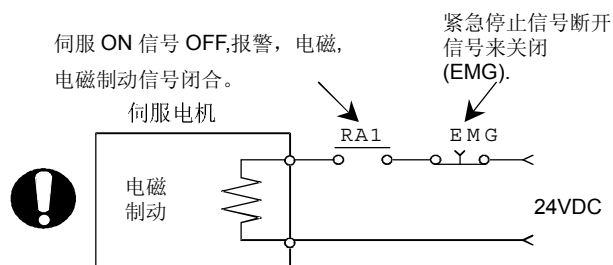
- 如果运动控制器、伺服放大器或伺服电机出现冒烟、声音异常、气味异常等现象，立即切断电源。
- 改变当程序或参数，保养和维修机器后，机器在开始实际运行前务必要执行试运行。
- 模块的分解和修理必须由专业人员来运行。
- 请勿对模块进行任何的改装。
- 电磁干扰可能会影响运动控制器或伺服放大器附近的电子设备，可用噪声滤波器或护罩来减小干扰。
- 当运动控制器使用 CE Mark-compliant 设备，请参阅“EMC 安装指导手册”（书号 IB(NA)-67339），同样参阅 EMC 安装指导手册有关伺服放大器，变频器和其它设备的信息。
- 根据下列条件使用模块

项目	条件			
	Q61P-A1	Q61P-A2	Q63P	Q64P
输入功率	100 到 120VAC ^{+10%} / _{-15%} (85 到 132VAC)	200 到 240VAC ^{+10%} / _{-15%} (170 到 264VAC)	24VDC ^{+30%} / _{-35%} (15.6 到 31.2VDC)	100 到 120VAC ^{+10%} / _{-15%} / 200 到 240VAC ^{+10%} / _{-15%} (85 到 132VAC/ 170 到 264VAC)
输入频率	50/60Hz ±5%			
可承受的瞬间掉电时间	20ms 以下			

7) 异常处理

⚠注意

- 如果运动控制器或伺服放大器在自我诊断时显示故障，根据指导运行说明书详细核实细节，然后重新启动运行。
- 为防止停电或产品故障发生事故，使用带有电磁制动的伺服电机，或在外部安装一个制动装置。
- 构建一个双重的电路结构，这样电磁制动运行电路能通过外部设置的紧急故障停机信号来运行。



⚠注意

- 如果出现错误，排除错误原因，报警解除后，重新启动运行。
- 当中断电源被恢复后，模块可能突然重新启动，这种情况下不要靠近机器。（设计机器时要确保即使机器突然重新启动员工的安全仍然能被保证。）

(8) 维修、保养和更换部件

⚠注意

- 根据使用说明手册，执行每天和定期的检查工作。
- 备份运动控制器和伺服放大器的程序和参数后，进行维护和检查工作。
- 当打开或关闭任何一个开口时，在清洁中不能将手指或收伸进其中。
- 根据使用说明手册，定期更换易耗部件，例如电池。
- 不要触摸 IC 或接头触点。
- 不要将运动控制器或伺服放大器放置在金属上，这样可能导致漏电；也不能放置在木头、塑料或乙烯材料上，否则可能导致静电积累。
- 在检查的过程中，不要进行一个高阻表测试（绝缘电阻测定）。
- 当更换运动控制器或伺服放大器时，要一直正确地设定新的模块。
- 当运动控制器或绝对值电机已经被替换，采用下列方之一法来执行一个原点回归操作，否则可能发生位置偏移。
 - 1) 用编程软件在运动控制器上写入伺服系统数据后，再次接通电源，然后执行一个原点回归操作。
 - 2) 使用编程软件中的备份功能，在替换前上传备份数据。
- 完成维修和检测后，确认绝对位置检测功能的位置检测正确。
- 不能短路、充电、过热、烧弃或分解电池。
- 电解电容在发生故障期间会产生气体，不要将脸靠近运动控制器或伺服放大器。
- 电解电容和风扇将会损坏。定期更换它们以防止故障后的二次破坏，更换工作可由我们的销售代表执行。

9) 废弃品的处理

当你丢弃运动控制器、伺服放大器、电池(原电池)和其它部件时,请遵从各个国家(地区)的法律。

⚠注意

- 本产品不是被设计或制造用在会影响或危害人类生命的设备或系统上。
- 当考虑本产品运行用于特殊的应用,例如机械、客运、医疗、航空、原子能、电能或海底重复应用时,请联系您最近的三菱销售代表。
- 虽然本产品是在严格的质量控制下制造,但当用语因产品坏掉而可能引起严重事故时,仍然强烈地建议您安装安全设备来预防严重的事故。

(10) 一般注意事项

⚠注意

- 使用说明手册中提供的所有图解显示封盖和安全防护装置移除的状态来解释详细的运行。当运行产品时,要将封盖和防护装置返回到指定的位置,然后根据使用说明手册进行操作。

前言

感谢您选择 Q173CPU(N)/Q172CPU(N) 运动控制器。
请仔细阅读本手册以便取得该设备的最佳应用效果。

目录

安全注意事项	A- 1
修订记录	A-11
目录	A-12
关于手册	A-18
1 概述	1- 1 到 1-94
1.1 概述	1- 1
1.2 特点	1- 3
1.2.1 运动 CPU 的特点	1- 3
1.2.2 性能规格	1- 6
1.2.3 操作控制/转移控制规格	1- 9
1.2.4 Q173CPU(N)/Q172CPU(N)与 A173UHCPU/A172SHCPUN 的区别	1-14
1.2.5 定位专用软元件/特殊继电器/特殊寄存器	1-16
1.3 硬件构成	1-54
1.3.1 运动系统构成	1-54
1.3.2 Q173CPU(N) 系统全构成	1-60
1.3.3 Q172CPU(N) 系统全构成	1-62
1.3.4 软件包	1-64
1.3.5 运动系统限制	1-68
1.4 多 CPU 系统	1-70
1.4.1 概述	1-70
1.4.2 PLC CPU 和运动 CPU 的安装	1-71
1.4.3 使用 Q 系列 I/O 模块和智能功能模块时的注意事项	1-72
1.4.4 模块安装的数量限制	1-73
1.4.5 多 CPU 系统处理时间	1-74
1.4.6 如何复位多 CPU 系统	1-75
1.4.7 在 CPU DOWN 故障发生时通过 QCPU 或 Q173CPU(N)/Q172CPU(N)处理	1-76
1.5 系统设置	1-79
1.5.1 系统数据设置	1-79
1.5.2 通用系统参数	1-80
1.5.3 个别参数	1-86
1.6 I/O 号分配	1-91
1.6.1 模块和智能功能模块 I/O 号	1-91
1.6.2 QCPU 和 Q173CPU(N)/Q172CPU(N)的 I/O 号	1-93
1.6.3 通过 SW6RN-GSV□P 设置 I/O 号	1-94
2 启动多 CPU 系统	2- 1 到 2- 2
2.1 多 CPU 系统启动流程图	2- 1

3 多CPU 系统中PLC CPU 和运动CPU之间的通讯	3- 1 到3-26
3.1 共享 CPU 内存的自动刷新功能	3- 1
3.2 从 PLC CPU 到运动 CPU 的控制指令 (运动专用指令).....	3-20
3.3 读/写软元件数据.....	3-21
3.4 共享 CPU 内存.....	3-22
4 运动CPU 程序结构	4- 1 到4- 4
4.1 SV13/SV22 实模式下的运动控制	4- 2
4.2 SV22 虚模式下的运动控制.....	4- 3
5 运动专用 PLC 指令	5- 1 到5-46
5.1 运动专用 PLC 指令.....	5- 1
5.1.1 运动专用 PLC 指令的限制项目	5- 1
5.2 从 PLC CPU 到运动 CPU 的运动 SFC 启动请求:	
S(P).SFCS (PLC 指令: S(P).SFCS)	5- 8
5.3 从 PLC CPU 到运动 CPU 的伺服程序启动请求:	
S(P).SVST (PLC 指令: S(P).SVST)	5-11
5.4 从 PLC CPU 到运动 CPU 的当前值更改指令:	
S(P).CHGA (PLC 指令: S(P).CHGA)	5-16
5.5 从 PLC CPU 到运动 CPU 的转速更改指令:	
S(P).CHGV (PLC 指令: S(P).CHGV)	5-29
5.6 从 PLC CPU 到运动 CPU 的转矩限制值更改请求指令:	
S(P).CHGT (PLC 指令: S(P).CHGT)	5-33
5.7 从 PLC CPU 向运动 CPU 写入: S(P).DDWR (PLC 指令: S(P).DDWR)	5-37
5.8 从运动 CPU 的软元件读取: S(P).DDR D (PLC 指令: S(P).DDR D)	5-41
5.9 到其它 CPU 中断指令: S(P).GINT (PLC 指令: S(P).GINT)	5-45
6 运动 SFC 程序	6- 1 到6-28
6.1 运动 SFC 程序结构	6- 1
6.2 运动 SFC 图表符号列表.....	6- 2
6.3 分流和耦合图表	6- 5
6.4 运动 SFC 程序名称	6- 9
6.5 步.....	6-10
6.5.1 运动控制步	6-10
6.5.2 运算控制步	6-11
6.5.3 子程序调用/启动步.....	6-12
6.5.4 清除步	6-14
6.6 转移	6-15
6.7 跳转, 指针	6-17
6.8 END.....	6-17
6.9 分支, 耦合	6-18
6.9.1 串联转移.....	6-18

6.9.2 选择性分支, 选择性耦合	6-19
6.9.3 并列分支, 并列耦合	6-20
6.10 Y/N 转移	6-22
6.11 运动 SFC 注释	6-26

7 运算控制程序

7- 1 到 7-90

7.1 运算控制程序	7- 1
7.2 软元件说明	7- 7
7.3 常数说明	7- 9
7.4 二项式运算	7-10
7.4.1 赋值: =	7-10
7.4.2 加: +	7-12
7.4.3 减: -	7-13
7.4.4 乘: *	7-15
7.4.5 除: /	7-16
7.4.6 取余: %	7-17
7.5 位运算	7-18
7.5.1 位反转(补数): ~	7-18
7.5.2 位逻辑与: &	7-19
7.5.3 位逻辑或: 	7-20
7.5.4 位异或: ^	7-21
7.5.5 位右移: >>	7-22
7.5.6 位左移: <<	7-23
7.5.7 符号反转(2 的补数): ~	7-24
7.6 标准功能	7-25
7.6.1 正弦: SIN	7-25
7.6.2 余弦: COS	7-26
7.6.3 正切: TAN	7-27
7.6.4 反正弦: ASIN	7-28
7.6.5 反余弦: ACOS	7-29
7.6.6 反正切: ATAN	7-30
7.6.7 平方根: SQRT	7-31
7.6.8 自然对数: LN	7-32
7.6.9 指数运算: EXP	7-33
7.6.10 绝对值: ABS	7-34
7.6.11 四舍五入: RND	7-35
7.6.12 舍: FIX	7-36
7.6.13 入: FUP	7-37
7.6.14 BCD \rightarrow BIN 转换: BIN	7-38
7.6.15 BIN \rightarrow BCD 转换: BCD	7-39
7.7 类型转换	7-40
7.7.1 带符号的 16 位整数值转换: SHORT	7-40
7.7.2 不带符号的 16 位整数值转换: USHORT	7-41
7.7.3 带符号的 32 位整数值转换: LONG	7-42
7.7.4 不带符号的 32 位整数值转换: ULONG	7-43
7.7.5 带符号的 64 位浮点值转换: FLOAT	7-44
7.7.6 不带符号的 64 位浮点值转换: UFLOAT	7-45

7.8 位软元件状态.....	7-46
7.8.1 ON (常开触点) : (无)	7-46
7.8.2 OFF (常闭触点) : !	7-47
7.9 位软元件控制.....	7-48
7.9.1 软元件置位: SET=	7-48
7.9.2 软元件复位: RST=	7-50
7.9.3 软元件输出: DOUT	7-52
7.9.4 软元件输入: DIN	7-53
7.9.5 位软元件输出: OUT	7-54
7.10 逻辑运算.....	7-55
7.10.1 逻辑肯定: (无).....	7-55
7.10.2 逻辑否定: !	7-56
7.10.3 逻辑与: *	7-57
7.10.4 逻辑或: +	7-58
7.11 比较运算.....	7-59
7.11.1 等于: ==	7-59
7.11.2 不等于: !=	7-60
7.11.3 小于: <	7-61
7.11.4 小于等于: <=	7-62
7.11.5 大于: >	7-63
7.11.6 大于等于: >=	7-64
7.12 运动专用功能(CHGV, CHGT).....	7-65
7.12.1 转速改变请求: CHGV	7-65
7.12.2 转矩限制值改变请求: CHGT	7-71
7.13 其它指令.....	7-73
7.13.1 事件任务允许: EI	7-73
7.13.2 事件任务禁止: DI	7-74
7.13.3 空: NOP	7-75
7.13.4 块转移: BMOV	7-76
7.13.5 向共享 CPU 内存写入软元件数据: MULTW	7-79
7.13.6 从其它 CPU 的共享 CPU 内存读取软元件数据: MULTR	7-81
7.13.7 向智能功能模块/特殊功能模块写入软元件数据: TO	7-84
7.13.8 从智能功能模块/特殊功能模块读取软元件数据: FROM	7-86
7.13.9 等待时间: TIME	7-88
7.14 注释: //	7-90

8 转移程序	8- 1 到 8- 2
---------------	--------------------

8.1 转移程序.....	8- 1
---------------	------

9 运动控制程序	9- 1 到 9-16
-----------------	--------------------

9.1 伺服指令表	9- 1
9.2 伺服电机/虚拟伺服电机轴当前值变更	9- 7
9.3 同步编码器轴当前值变更控制 (仅用于 SV22)	9-10
9.4 一转内凸轮轴当前值变更控制(仅用于 SV22)	9-13

9.5 编程指令.....	9-15
9.5.1 撤消·开始.....	9-15
9.5.2 使用运动软元件间接指定.....	9-15

10 运动软元件	10- 1 到 10- 6
-----------------	----------------------

10.1 运动寄存器 (#0 到 #8191).....	10- 1
10.2 惯性定时器 (FT).....	10- 6

11 运动 SFC 参数	11- 1 到 11-20
---------------------	----------------------

11.1 任务定义.....	11- 1
11.2 连续转移次数和任务运行.....	11- 2
11.2.1 连续转移次数.....	11- 2
11.2.2 任务运行.....	11- 3
11.3 多任务的执行状态.....	11- 7
11.4 任务参数.....	11- 8
11.5 程序参数.....	11-10
11.6 如何启动运动 SFC 程序.....	11-16
11.6.1 自动启动.....	11-16
11.6.2 从运动 SFC 程序启动.....	11-16
11.6.3 从 PLC 启动 (PLC 指令 S(P).SFCS).....	11-16
11.7 如何结束运动 SFC 程序.....	11-17
11.8 如何从一个运动 SFC 程序转至另一个运动 SFC 程序.....	11-17
11.9 如何管理执行中程序.....	11-17
11.10 当 CPU 电源关闭或复位时执行的操作.....	11-18
11.11 当 CPU RUN/STOP 切换时执行的操作.....	11-18
11.12 当 PLC Ready 标志 (M2000) OFF/ON 时执行的操作.....	11-19
11.13 故障发生时的操作.....	11-20

12 用户文件	12- 1 到 12- 8
----------------	----------------------

12.1 项目.....	12- 1
12.2 用户文件表.....	12- 2
12.3 运动 SFC 程序中的在线更改.....	12- 3
12.3.1 在线更改操作方法.....	12- 4
12.3.2 程序传送.....	12- 7

13 限位开关输出动能	13- 1 到 13- 8
--------------------	----------------------

13.1 操作.....	13- 1
13.2 限位开关设置数据.....	13- 4

14 ROM 运算功能	14- 1 到 14-12
--------------------	----------------------

14.1 有关 ROM 运算功能.....	14- 1
14.2 LED·开关规格.....	14- 3
14.3 ROM 运算功能详情.....	14- 5
14.4 编程软件包“ROM 写入”的运算步骤.....	14-11

15 网络通讯	15- 1到15-10
15.1 网络通讯规格.....	15- 2
15.2 网络通讯访问范围.....	15- 3
15.2.1 采用 MELSECNET/10(H) 或以太网的网络结构.....	15- 3
15.2.2 采用 CC-Link 的网络结构.....	15- 5
15.2.3 采用 RS422/485 的网络结构.....	15- 6
15.2.4 混合采用 MELSECNET/10 (H), 以太网, CC-Link, RS422/485 的网络结构.....	15- 7
16 主周期监视功能	16- 1到16- 2
17 从伺服放大器上读取伺服参数的功能	17- 1到17- 2
17.1 关于伺服参数读取请求软元件.....	17- 1
17.2 从伺服放大器上读取伺服参数功能的操作步骤.....	17- 2
18 故障代码表	18- 1到18-16
18.1 故障代码读取步骤.....	18- 1
18.2 运动 SFC 故障代码表 (通过 GSV□P 确认).....	18- 2
18.3 运动 SFC 参数故障 (通过 GSV□P 确认).....	18-10
18.4 多 CPU 故障代码 (通过 GX Developer 确认).....	18-11
18.4.1 自诊断故障代码.....	18-11
18.4.2 自诊断故障消除 (通过 GSV□P 执行).....	18-16
附录	附录- 1到附录-31
附录 1 处理时间.....	附录- 1
附录 1.1 运算控制/转移指令处理.....	附录- 1
附录 2 程序举例.....	附录- 9
附录 2.1 连续执行多 CPU 专用指令的程序例.....	附录- 9
附录 2.2 通过一次指令执行多个多 CPU 指令的程序例.....	附录- 11
附录 2.3 进行运动控制 SFC 程序组合举例.....	附录- 13
附录 2.4 通过运动 SFC 程序在子程序重启时的连续执行例.....	附录- 24
附录 2.5 通过运动 SFC 程序在停止后的连续执行例.....	附录- 28

关于手册

本手册仅描述说明运动控制器的硬件。

以下手册与本产品相关。

请根据列表索取必要的手册。

本用户手册并没有详细描述 PLC CPU 模块，电源模块，基板和 I/O 模块的硬件规格和操作方法。

有关这些方面的详细描述，请参考 QCPU (Q 模式) 用户手册和组件 I/O 模块用户手册。

相关手册

	手册名	手册编号
运动 CPU 的硬件和软件	Q173CPU(N)/Q172CPU(N) 运动控制器用户手册 本手册说明运动 CPU 模块，Q172LX 伺服外部信号接口模块，Q172EX 串行绝对同步编码器接口模块，Q173PX 手动脉冲生成器接口模块，电源模块，伺服放大器 SSCNET 电缆和同步编码器电缆的规格。 (另售)	IB(NA)-0300040CHN
	Q173CPU(N)/Q172CPU(N) 运动控制器(SV13/SV22) 编程手册(实模式篇) 本手册说明伺服参数，位置指令，软元件列表和出错列表。 (另售)	IB(NA)-0300043CHN
	Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22) 编程手册 (虚模式篇) 本手册描述通过虚拟主轴来执行同步控制的专用指令，以及可创建机械系统程序的机械模块的专用指令 本手册说明伺服参数，位置指令，软元件列表和出错列表。 (另售)	IB(NA)-0300044CHN

	手册名	手册编号
PLC CPU 的硬件和软件	基本型 QCPU (Q 模式) 用户手册 (硬件设计, 维护和检修篇) 本手册说明 CPU 模块, 电源模块, 基本模块, 扩展电缆, 内存卡 和其它产品的规格。 (另售)	SH(NA)-080333C
	基本型 QCPU (Q 模式) 用户手册 (功能解释, 编程基础篇) 本手册说明使用 QCPU (Q 模式)创建程序所必需的功能, 编程方法, 软元件等等。 (另售)	SH(NA)-080331C
	高性能型 QCPU (Q 模式) 用户手册 (硬件设计, 维护和检修篇) 本手册说明 CPU 模块, 电源模块, 基本模块, 扩展电缆, 内存卡和其它产品的规格。 (另售)	SH(NA)-080233C
	高性能型 QCPU (Q 模式) 用户手册 (功能解释, 编程基础篇) 本手册说明使用 QCPU (Q 模式)创建程序所必需的功能, 编程方法, 软元件等等。 (另售)	SH(NA)-080232C
	QCPU (Q 模式)/QnACPU 编程手册 (通用指令篇) 本手册说明如何使用顺控指令, 基本指令, 应用指令和微电脑程序。 (另售)	SH(NA)-080450CHN
	QCPU (Q 模式)/QnACPU 编程手册 (PID 控制指令篇) 本手册说明用来进行 PID 控制的专用指令。 (另售)	SH-080040
	QCPU (Q 模式)/QnACPU 编程手册 (SFC) 本手册说明 MELSAP3 的系统结构, 性能规格, 功能, 编程, 调试, 出错代码和其它有关内容。 (另售)	SH(NA)-080283C
	I/O 模块用户手册 本手册说明 I/O 模块, 接头, 接头/端子排转换模块以及其它产品的规格。 (另售)	SH(NA)-080329C

1 概述

1.1 概述

本编程手册描述了运动CPU模块(Q173CPU(N)/Q172CPU(N))中操作系统软件包“SW6RN-SV13Q□□”，“SW6RN-SV22Q□□”的运动SFC程序及多CPU系统。本手册中使用了下列缩写。

统称/缩写	说明
Q173CPU(N)/Q172CPU(N), 运动 CPU 模块	Q173CPUN/Q172CPUN/Q173CPN-T/Q172CPN-T/ Q173CPU/Q172CPU 运动 CPU 模块
Q172LX/Q172EX/Q173PX 或 运动模块	Q172LX 伺服外部信号输入模块 Q172EX(-S1)串行 ABS 同步编码器输入模块 Q173PX(-S1)手动脉冲发生器输入模块
MR-H-BN	伺服放大器 MR-H□BN
MR-J2□-B	伺服放大器 MR-J2S-□B/MR-J2M-B/MR-J2-□B/MR-J2-03B5
AMP 或伺服放大器	伺服放大器 MR-H□BN/MR-J2S-□B/MR-J2M-B/ MR-J2-□B/MR-J2-03B5, 矢量变频器 FREQROL-V500 系列
QCPU, PLC CPU 或 PLC CPU 模块	Qn(H)CPU
多 CPU 系统或运动系统	Q 系列多 CPU 系统简称
CPUn	多 CPU 系统中 CPU 模块的 CPU No.n (n= 1 至 4) 简称
编程 S/W 包	SW6RNC-GSV□E 与 GX Developer 软件包
编程软件	"SW□RN-□P"的总称
操作系统	"SW□RN-SV□Q□"的总称
GSV 或 GSV□P	SW6RNC-GSVPROE 的简称 (集成启动支持软件包)
SV13	传送装配使用的操作系统软件(运动 SFC): SW6RN-SV13Q□
SV22	自动机器使用的操作系统软件(运动 SFC): SW6RN-SV22Q□
GX Developer	SW6D5C-GPPW-E(GX Developer 功能软件包)的简称
手动脉冲发生器或 MR-HDP01	手动脉冲发生器 MR-HDP01)的简称
串行 ABS 同步编码器或 MR-HENC	串行 ABS 同步编码器 (MR-HENC) 的简称
SSCNET ^(注)	运动控制器与伺服放大器间的高速同步通讯网络
降温风扇单元	降温风扇单元 (Q170FAN)
分线单元	分线单元(Q173DV)
电池单元	电池单元 (Q170BAT)
A□0BD-PCF	A10BD-PCF/A30BD-PCF SSC I/F 板

(注) SSCNET: 伺服系统控制网络

1 概述

统称/缩写	说明
示教模块或 A31TU-D3K□/A31TU-DNK□	A31TU-D3K□/A31TU-DNK□
智能功能模块	MELSECNET/H 模块/Ethernet 模块/CC-Link 模块/串行通讯模块的简称
矢量变频器(FR-V500)	FREQROL-V500 系列矢量变频器的简称

备注

欲知 QCPU，创建 PLC 程序的外围设备，I/O 模块与智能功能模块，请参考相关手册。欲知 SV13/SV22 实模式的定位程序与参数设计方法，请参考“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册 (实模式)”。欲知 SV22 虚模式的机械结构程序设计方法，请参考“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22) 编程手册 (虚模式)”。

1.2 特点

运动 CPU 与多 CPU 系统有如下特点。

1.2.1 运动 CPU 的特点

(1) Q 系列 PLC 多 CPU 系统

(a) 各 CPU 控制过程的下载分配如下，运动 CPU 进行复杂的伺服控制，PLC CPU 控制机器及信息，可实现系统的灵活配置。

(b) 运动 CPU 与 PLC CPU 可灵活选用，这样就可实现最多带有 4 个 CPU 模块的多 CPU 系统。

根据所用的轴数选择运动 CPU 模块。

Q173CPU(N) : 32 轴

Q172CPU(N) : 8 轴

根据所用的程序容量选择 PLC CPU 模块。(多 CPU 系统需一个或多个 PLC CPU。)

Q00CPU : 8k 步

Q01CPU : 14k 步

Q02CPU, Q02HCPU : 28k 步

Q06HCPU : 60k 步

Q12HCPU : 124k 步

Q25HCPU : 252k 步

(c) 其它 CPU 的软元件数据可作为自 CPU 数据使用，因为多 CPU 自动刷新功能可实现多 CPU 系统中各 CPU 之间数据自动收发。

(d) 运动 CPU 的软元件数据存取与运动 SFC 程序启动可通过运动专用 PLC 指令由 PLC CPU 执行。

(2) 运动 SFC 程序的编程

(a) 由于机器运行顺序可根据运行步通过流程图的形式写出，所以编出的程序浅显易懂可维护性强。

(b) 因为运动 CPU 判定转换条件并开始定位启动，所以不会因为受 PLC 扫描时间的影响而出现响应时间或偏差。

- (c) 运动 SFC 步处理方式 (仅执行激活步) 保证了高速度及高响应处理。
 - (d) 运动 CPU 不仅能执行定位控制, 也能进行数值运算及软元件 SET/RST 等, 不需要通过 PLC CPU 操作, 减少了运行时间。
 - (e) 通过运动 SFC-专用转移条件, 启动条件一成立指令即可传送到伺服放大器。
 - (f) 运动 SFC-专用转移条件, 允许启动后即转至下一步, 不需等待定位完成。
 - (g) 可设置运动 SFC 程序以高速响应外部中断输入。
 - (h) 可设置运动 SFC 程序与运动运算周期同步, 并按固定周期执行 (1.77ms, 3.55ms, 7.11ms, 14.2ms)。
- (3) 高速运行处理
- (a) 运动 CPU 的最低运行周期为 0.88[ms] (是原来的 4 倍)。适合高频率运行。
 - (b) Q 系列 PLC CPU 中可实现高速顺序控制(基本指令为 0.034[μ s]) (使用 Q02CPU 时, 基本指令为 0.079[μ s])。
- (4) 通过 SSCNET 高速串行通讯连接运动控制器与伺服放大器
- 用 SSCNET 高速串行通讯来连接运动控制器与伺服放大器, 并集中控制伺服参数的更改, 伺服监视及试运行等, 并可减少配线。

(5) 与用途适应的操作系统软件包

为运动CPU的内部闪存存储器安装操作系统软件,可成为适合机器的运动控制器。并可与软件包的功能提高相符合。

(a) 传送装配用 (SV13)

通过专用伺服命令进行直线插补,圆弧插补,螺旋插补,恒速控制,速度控制,位置跟踪控制等。它是传送机械及包装机械的最佳选择。

(b) 自动机器用 (SV22)

通过机器支持语言实现同步控制及软件凸轮控制。它是自动机器的最佳选择。

1 概述

1.2.2 性能规格

(1) Q173CPU(N)/Q172CPU(N) 的基本规格

a) 模块规格

项目	Q173CPUN	Q173CPUN-T	Q173CPU	Q172CPUN	Q172CPUN-T	Q172CPU
示教模块	—	可用	—	—	可用	—
电流消耗(5VDC) [A]	1.25	1.56 ^(注)	1.75	1.14	1.45 ^(注)	1.62
外形尺寸 [mm]	98 (H) × 27.4 (W) × 114.3 (D)		118 (H) × 27.4 (W) × 89.3 (D)	98 (H) × 27.4 (W) × 114.3 (D)		118 (H) × 27.4 (W) × 89.3 (D)
质量 [kg]	0.23	0.24	0.22	0.22	0.23	0.21

(注): 包括示教模块的电流消耗 0.26 [A]

b) 运动控制规格 /性能规格

• 运动控制规格

项目	Q173CPUN(-T)	Q173CPU	Q172CPUN(-T)	Q172CPU
控制轴数目	32 轴		8 轴	
运行周期 (缺省)	SV13	0.88ms/ 1 至 8 轴 1.77ms/ 9 至 16 轴 3.55ms/17 至 32 轴	0.88ms/1 至 8 轴	
	SV22	0.88ms/ 1 至 4 轴 1.77ms/ 5 至 12 轴 3.55ms/13 至 24 轴 7.11ms/25 至 32 轴	0.88ms/1 至 4 轴 1.77ms/5 至 8 轴	
插补功能	直线插补 (最多 4 轴), 圆形插补 (2 轴), 螺旋插补 (3 轴)			
控制方式	PTP(点对点) 控制, 速度控制, 速度/位置控制, 定长进给, 恒速控制, 位置跟踪控制, 速度切换控制, 高速振动控制, 同步控制 (SV22)			
加速/ 减速控制	自动梯形加速/减速, S-曲线加速/减速			
补偿功能	回差补偿, 电子齿轮			
编程语言	运动 SFC, 专用命令, 机器支持语言 (SV22)			
程序容量	14k 步			
定位 点数	3200 点 (可间接指定)			
编程工具	IBM PC/AT			
外围 I/F	USB/RS-232/SSCNET			
原点回归功能	近点 dog 型 (2 种), 计数器型 (3 种), 数据设置类型 (2 种), dog 支架型, 制动器停止类型 (2 种), 限位开关组合型 (有原点回归重试功能及原点偏移功能)			
JOG 运行功能	有			
手动脉冲发生器运行功能	可连接 3 个模块			

1 概述

• 运动控制规格 (续)

项目	Q173CPUN(-T)	Q173CPU	Q172CPUN(-T)	Q172CPU
同步编码器运行功能	可连接 12 个 模块		可连接 8 个 模块	
M 代码功能	提供 M 代码输出功能和 M 代码完成等待功能			
限位开关输出功能	输出点数 32 点 监视数据: 运动控制数据/字软元件			
绝对位置系统	将电池安装于伺服放大器中即可对应(每轴可以选用绝对方式或增量方式) (注): 使用矢量变频器时只可使用增量方式。			
SSCNET I/F 数目	5CH ^{注-1)}		2CH	
运动相关模块安装数目	Q172LX: 可用 4 个 模块 Q172EX: 可用 6 个 模块 Q173PX: 可用 4 个 模块 ^{注-2)}		Q172LX: 可用 1 个 模块 Q172EX: 可用 4 个 模块 Q173PX: 可用 3 个 模块 ^{注-2)}	

注-1): 使用分线单元(Q173DV) 或分线电缆(Q173J2B △C BL□M/Q173HB△C BL□M)。

注-2): 使用(sv22) 时增量同步编码器可用台数。连接手动脉冲发生器时只可使用 1 个模块。

1 概述

• 运动 SFC 性能规格

项目		Q173CPU(N)/Q172CPU(N)		
程序容量	总代码 (运动 SFC 图+ 运算控制+ 转移)	287k 字节		
	总文本 (运算控制+转移)	224k 字节		
运动 SFC 程序	运动 SFC 程序数	256(No.0 至 255)		
	运动 SFC 图大小/1 程序	至多 64k 字节 (包括运动 SFC 图注释)		
	运动 SFC 步数/1 程序	至多 4094 步		
	选择分支数/1 分支	255		
	并列分支数/1 分支	255		
	并列分支嵌套	至多 4		
操作控制程序 (F/FS) / 转移程序 (G)	操作控制程序数	4096 F(一次执行型)与 FS(扫描执行型) 相结合 (F/FS0 至 F/FS4095)		
	转移程序数	4096(G0 至 G4095)		
	代码大小/1 程序	至多 64k 字节(32766 步)		
	块(行) 数/1 程序	至多 8192 块 (4 步(最少)/1 块时)		
	字数/1 块	至多 128 (包括注释)		
	操作数/1 块	至多 64 个(运算因子: 常数, 字软元件, 位软元件)		
	() 嵌套/1 块	至多 32		
	表达式	操作控制程序 转移程序	算术式/位条件式 算术式/位条件式/比较条件式	
执行规格	同时执行程序数	至多 256		
	同时激活程序数	至多 256 步/所有程序		
	执行任务	一般任务	运动主周期中执行	
		事件任务 (可屏蔽执行)	固定周期	固定周期中执行 (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms)
			外部中断	在中断模块 QI60 中设置输入 (16 点) 为 ON 时执行
			PLC 中断	有来自 PLC CPU 中断命令时执行
NMI 任务	在中断模块 QI60 中设置输入 (16 点) 为 ON 时执行			
I/O 点数(X/Y)	8192 点			
实际 I/O 点数 (PX/PY)	256 点			
软元件数目 (仅限运动 CPU 内) (包括定位专用软元件)	内部继电器 (M)	共 (M + L) : 8192 点		
	锁存继电器 (L)			
	链接继电器 (B)	8192 点		
	信号器 (F)	2048 点		
	特殊继电器 (M)	256 点		
	数据寄存器 (D)	8192 点		
	链接寄存器 (W)	8192 点		
	特殊寄存器 (D)	256 点		
	运动寄存器 (#)	8192 点		
	惯性定时器 (FT)	1 点(888μs)		

1 概述

1.2.3 操作控制 /转移控制规格

(1) 操作控制 /转移控制规格表

项目	规格		备注																																																																														
表达式	算术表达式		结果为数字 使用常数及字软元件间接指定数据的计算																																																																														
	条件表达式	位条件表达式	判定位软元件 ON 或者 OFF																																																																														
		比较条件表达式	结果为真或假 使用常数及文本软元件间接指定数据, 进行算术表达式的比较																																																																														
位软元件	<table border="1"> <thead> <tr> <th rowspan="2">软元件</th> <th rowspan="2">符号</th> <th colspan="2">可达性</th> <th colspan="3">可用任务</th> <th rowspan="2">举例说明</th> </tr> <tr> <th>读</th> <th>写</th> <th>正常</th> <th>事件</th> <th>NMI</th> </tr> </thead> <tbody> <tr> <td rowspan="2">输入</td> <td>输入模块未安装范围</td> <td>X</td> <td>○</td> <td>○</td> <td rowspan="8">○</td> <td rowspan="8">○</td> <td rowspan="8">○</td> <td>X100</td> </tr> <tr> <td>输入模块安装范围</td> <td>PX</td> <td>○</td> <td>×</td> <td>PX180</td> </tr> <tr> <td rowspan="2">输出</td> <td>输出模块未安装范围</td> <td>Y</td> <td>○</td> <td>○</td> <td>Y100</td> </tr> <tr> <td>输出模块安装范围</td> <td>PY</td> <td>○</td> <td>○</td> <td>PY1E0</td> </tr> <tr> <td colspan="2">内部继电器</td> <td>M</td> <td>○</td> <td>○</td> <td>M20</td> </tr> <tr> <td colspan="2">锁存继电器</td> <td>L</td> <td>○</td> <td>○</td> <td>L1000</td> </tr> <tr> <td colspan="2">链接继电器</td> <td>B</td> <td>○</td> <td>○</td> <td>B3FF</td> </tr> <tr> <td colspan="2">信号器</td> <td>F</td> <td>○</td> <td>○</td> <td>F0</td> </tr> <tr> <td colspan="2">特殊继电器</td> <td>M</td> <td>○</td> <td>○</td> <td>M9000</td> </tr> </tbody> </table> <p style="text-align: right;">○: 可用 ×: 不可使用</p> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;"> <p>备注</p> <p><可写位软元件限制></p> <p>1) 只有在输入模块未安装范围内才可能写至软元件 X。</p> <p>2) 特殊继电器已预先确定了系统中的用途。不能写至用户设置软元件以外。</p> <p>(注): 下列软元件范围禁止 SET/RST。</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>不能 SET/RST 范围</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td>M2001 至 M2032</td> <td>启动接受软元件</td> </tr> </tbody> </table> <p>(注): 下列软元件范围禁止 DOUT 输出。</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>DOUT 输出禁止范围</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td>指定包括 M2000 至 M2127</td> <td>专用软元件</td> </tr> <tr> <td>M9000 至 M9255</td> <td>特殊继电器</td> </tr> </tbody> </table> </div>		软元件	符号	可达性		可用任务			举例说明	读	写	正常	事件	NMI	输入	输入模块未安装范围	X	○	○	○	○	○	X100	输入模块安装范围	PX	○	×	PX180	输出	输出模块未安装范围	Y	○	○	Y100	输出模块安装范围	PY	○	○	PY1E0	内部继电器		M	○	○	M20	锁存继电器		L	○	○	L1000	链接继电器		B	○	○	B3FF	信号器		F	○	○	F0	特殊继电器		M	○	○	M9000	不能 SET/RST 范围	备注	M2001 至 M2032	启动接受软元件	DOUT 输出禁止范围	备注	指定包括 M2000 至 M2127	专用软元件	M9000 至 M9255	特殊继电器	输入 X/输出 Y, 实际输入用 PX/实际输出用 PY 表示。 PX, PY 的 I/O 数由系统设定分配。 (操作控制程序/转移程序中, 按照系统设置信息自动替换 PX/PY 的表示。)
软元件	符号	可达性			可用任务			举例说明																																																																									
		读	写	正常	事件	NMI																																																																											
输入	输入模块未安装范围	X	○	○	○	○	○	X100																																																																									
	输入模块安装范围	PX	○	×				PX180																																																																									
输出	输出模块未安装范围	Y	○	○				Y100																																																																									
	输出模块安装范围	PY	○	○				PY1E0																																																																									
内部继电器		M	○	○				M20																																																																									
锁存继电器		L	○	○				L1000																																																																									
链接继电器		B	○	○				B3FF																																																																									
信号器		F	○	○				F0																																																																									
特殊继电器		M	○	○	M9000																																																																												
不能 SET/RST 范围	备注																																																																																
M2001 至 M2032	启动接受软元件																																																																																
DOUT 输出禁止范围	备注																																																																																
指定包括 M2000 至 M2127	专用软元件																																																																																
M9000 至 M9255	特殊继电器																																																																																

1 概述

操作控制 /转移控制规格表 (续)

项目	规格							备注
字软件	软件件		可达性		可用任务			举例说明
		符号	读	写	正常	事件	NMI	
	数据寄存器	D	○	○	○	○	○	DOL
	链接寄存器	W	○	○				W1F:F
	特殊寄存器	D	○	○				D9000
	运动寄存器	#	○	○				#0F
惯性定时器	FT	○	×	FT				
<p style="text-align: right;">○:可用 ×:不可使用</p>								
注意		<p><可写文本软件的限制> 1)只有在输入模块未安装范围内才可能写至软件件 X。 2)特殊继电器已预先确定了系统中的设施。不能写至 用户设置软件件以外。</p>						
数据类型	(无)	16-位 整型 (有符号)		-32768 至 32768				K10,D100 等.
		16-位 整型 (无符号)		0 至 65535				
	L	32-位 整型 (有符号)		-2147483648 至 2147483647				2000000000, W100L 等.
		32-位 整型 (无符号)		0 至 4294967295				
F	64-位 浮点小型(双精度实数型)		IEEE 格式				1.23, #10F 等.	
常数	K	十进制常数	上述数据类型符号 'L' 或 '.' (十进制点) 在常数尾部说明了数据的类型。省略数据类型的常数可视为可用的最小类型。					K-100, H0FFL 等. 'K' 可省略。
	H	十六进制常数						
指令数	二进制操作		6				共 62	
	位运算		6					
	有符号		1					
	标准函数		15					
	类型变换		6					
	位软件件状态		2					
	位软件件控制		5					
	逻辑运算		4					
	比较运算		6					
	运动专用函数		2					
其他		9						
输入 PX 及 输出 PY 的读/写响应	输入响应		指令执行时直接读出控制					
	输出响应		指令执行时直接写入控制					

1 概述

2 操作控制 / 转换说明表

分类	符号	功能	格式	基本步数	可用步		可用表达式			
					F/FS	G	计算表达式	位条件表达式	比较条件表达式	Y/N 转移条件表达式
二进制操作	=	赋值	(D)=(S)	4	○	○	○	—	—	—
	+	加	(S1)+(S2)	4	○	○	○	—	—	—
	-	减	(S1)-(S2)	4	○	○	○	—	—	—
	*	乘	(S1)*(S2)	4	○	○	○	—	—	—
	/	除	(S1)/(S2)	4	○	○	○	—	—	—
	%	取余	(S1)%(S2)	4	○	○	○	—	—	—
位运算	~	位取反 (求补数)	~(S)	2	○	○	○	—	—	—
	&	位逻辑与	(S1)&(S2)	4	○	○	○	—	—	—
		位逻辑或	(S1) (S2)	4	○	○	○	—	—	—
	^	位异或	(S1)^(S2)	4	○	○	○	—	—	—
	>>	位右移	(S1)>>(S2)	4	○	○	○	—	—	—
	<<	位左移	(S1)<<(S2)	4	○	○	○	—	—	—
符号	-	符号取反 (2 的补数)	-(S)	4	○	○	○	—	—	—
标准功能	SIN	正弦	SIN(S)	2	○	○	○	—	—	—
	COS	余弦	COS(S)	2	○	○	○	—	—	—
	TAN	正切	TAN(S)	2	○	○	○	—	—	—
	ASIN	反正弦	ASIN(S)	2	○	○	○	—	—	—
	ACOS	反余弦	ACOS(S)	2	○	○	○	—	—	—
	ATAN	反正切	ATAN(S)	2	○	○	○	—	—	—
	SQRT	平方根	SQRT(S)	2	○	○	○	—	—	—
	LN	自然对数	LN(S)	2	○	○	○	—	—	—
	EXP	指数	EXP(S)	2	○	○	○	—	—	—
	ABS	绝对值	ABS(S)	2	○	○	○	—	—	—
	RND	四舍五入	RND(S)	2	○	○	○	—	—	—
	FIX	向下舍入	FIX(S)	2	○	○	○	—	—	—
	FUP	向上舍入	FUP(S)	2	○	○	○	—	—	—
BIN	BCD→BIN	BIN(S)	2	○	○	○	—	—	—	
BCD	BIN→BCD	BCD(S)	2	○	○	○	—	—	—	
类型换算	SHORT	转换成 16 位整数 (有符号)	SHORT(S)	2	○	○	○	—	—	—
	USHORT	转换成 16 位整数 (无符号)	USHORT(S)	2	○	○	○	—	—	—
	LONG	转换成 32 位整数 (有符号)	LONG(S)	2	○	○	○	—	—	—
	ULONG	转换成 32 位整数 (无符号)	ULONG(S)	2	○	○	○	—	—	—
	FLOAT	有符号数据并转换成 64 位浮点数	FLOAT(S)	2	○	○	○	—	—	—
	UFLOAT	无符号数据并转换成 64 位浮点数	UFLOAT(S)	2	○	○	○	—	—	—

1 概述

操作控制 /转换指令表 (续)

分类	符号	功能	格式	基本步数	可用步		可用表达式			
					F/FS	G	算术表达式	位条件表达式	比较条件表达式	Y/N 转换条件表达式
位软元件状态	(无)	ON (常开接点)	(位条件表达式)	2	○	○	—	○	—	○
	!	OFF (常闭接点)	!(位条件表达式)	2	○	○	—	○	—	○
位软元件控制	SET	软元件置位	SET(D)	3	○	○	—	○	—	—
			SET(D)=(条件表达式)	4	○	○	—	○	○	—
	RST	软元件复位	RST(D)	3	○	○	—	○	—	—
			RST(D)=(条件表达式)	4	○	○	—	○	○	—
	DOUT	软元件输出	DOUT(D),(S)	4	○	○	—	○	—	—
	DIN	软元件输入	DIN(D),(S)	4	○	○	—	○	—	—
OUT	位软元件输出	OUT(D)=(条件表达式)	4	○	○	—	○	○	—	
逻辑运行	(无)	逻辑真	(条件表达式)	0	○	○	—	○	○	○
	!	逻辑非	!(条件表达式)	2	○	○	—	○	○	○
	*	逻辑与	(条件表达式) * (条件表达式)	4	○	○	—	○	○	○
	+	逻辑或	(条件表达式) + (条件表达式)	4	○	○	—	○	○	○
比较运行	==	等于	(计算表达式) == (计算表达式)	4	○	○	—	—	○	○
	!=	不等于	(计算表达式) != (计算表达式)	4	○	○	—	—	○	○
	<	小于	(计算表达式) < (计算表达式)	4	○	○	—	—	○	○
	<=	小于等于	(计算表达式) <= (计算表达式)	4	○	○	—	—	○	○
	>	大于	(计算表达式) > (计算表达式)	4	○	○	—	—	○	○
	>=	大于等于	(计算表达式) >= (计算表达式)	4	○	○	—	—	○	○
运动专用函数	CHGV	速度改变请求	CHGV((S1),(S2))	4	○	○	—	—	—	—
	CHGT	转矩限制值改变请求	CHGT((S1),(S2))	4	○	○	—	—	—	—

1 概述

运算控制 /转移指令表 (续)

分类	符号	功能	格式	基本步数	可用步		可用表达式			
					F/FS	G	算术表达式	位条件表达式	比较条件表达式	Y/N 转换条件表达式
其他	EI	事件任务使能	EI	1	○	○	—	—	—	—
	DI	事件任务禁止	DI	1	○	○	—	—	—	—
	NOP	空操作	NOP	1	○	○	—	—	—	—
	BMOV	块传送	BMOV(D),(S),(n)	6	○	○	—	—	—	—
	MULTW	将软元件数据写至本机共享内存	MULTW(D),(S),(n),(D1)	8	○	○	—	—	—	—
	MULTR	软元件数据从共享内存读出	MULTR(D),(S1),(S2),(n)	7	○	○	—	—	—	—
	TO	将软元件数据写至智能功能模块/专用功能模块	TO(D1),(D2),(S),(n)	7	○	○	—	—	—	—
	FROM	将软元件数据从智能功能模块/专用功能模块读出	FROM(D),(S1),(S2),(n)	7	○	○	—	—	—	—
	TIME	等待时间	TIME(S)	7	—	○	—	—	—	—

○: 可用
—: 不可使用

(3) 运算控制 /转移程序的 1 个程序代码数量的粗略计算式

$$\begin{aligned}
 & 2 + (1 + 1 \text{ 个块中基本步数} \\
 & + (32 \text{ 位常数个数} / 1 \text{ 个块}) \times 1 \\
 & + (64 \text{ 位常数个数} / 1 \text{ 个块}) \times 3 \times \text{块数(步)} \\
 & \hspace{15em} (1 \text{ 步} = 2 \text{ 个字节})
 \end{aligned}$$

1 概述

1.2.4 Q173CPU(N) /Q172CPU(N) 与 A173UHCPU/A172SHCP 的区别

(d) Q173CPU(N) /Q172CPU(N) 与 A173UHCPU/A172SHCP 的区别

项目			Q173CPU(N)	Q172CPU(N)	A173UHCPU	A172SHCP 不	
运动控制	控制轴数		32 轴	8 轴	32 轴	8 轴	
	运行周期	SV13	0.88ms/1 至 8 轴 1.77ms/9 至 16 轴 3.55ms/17 至 32 轴 (默认) (可按参数设定。)	0.88ms/1 至 8 轴 (默认) (可按参数设定。)	3.55ms/1 至 20 轴 7.11ms/21 至 32 轴	3.55ms/1 至 8 轴	
		SV22	0.88ms/1 至 4 轴 1.77ms/5 至 12 轴 3.55ms/13 至 24 轴 7.11ms/25 至 32 轴 (默认) (可按参数设定。)	0.88ms/1 至 4 轴 1.77ms/5 至 8 轴 (默认) (可按参数设定。)	3.55ms/1 至 12 轴 7.11ms/13 至 24 轴 14.2ms/25 至 32 轴	3.55 ms/1 至 8 轴	
	程序容量		14k 步			13k 步	
	定位点数		3200 点/轴(可间接指令定位数据。)				
	编程工具		IBM PC/AT, A31TU-D		PC9800 系列, IBM PC/AT, A30TU, A31TU		
	外围 I/F		USB/RS-232/SSCNET		RS-422/SSCNET		
	原点回归功能		近点 dog 型(2 种), 计数器型(3 种), 数据设置型(2 种),dog 支架型, 制动器停止类型(2 种), 限位开关组合型 (提供复位功能, 及定位转移功能)		近点 dog 型, 计数器型, 数据设置类型 1		
	手动脉冲发生器运行功能		可连接 3 个模块			可连接 1 个模块	
	同步编码器运行功能		可连接 12 个模块	可连接 8 个模块	可连接 4 个模块		
	限位开关输出功能		输出点: 32 点, 监视数据: 运动控制数据/文本软元件				
	SSCNET 接口数 (包括个人电脑中的 SSCNET 接口 1CH)		5CH (注 1)	2CH	4CH	2CH	
	运动槽数		至多 64 槽 (Q 系列至多有 7 个扩展基地址。)		8 槽	2 槽	
	运动相关 模块数		Q172LX: 4 个 模块 Q172EX: 6 个 模块 Q173PX: 4 个 模块 ^(注-2)	Q172LX: 1 个 模块 Q172EX: 4 个 模块 Q173PX: 3 个 模块 ^(注-2)	A172SENC: 4 个 模块	A172SENC: 1 个 模块	
	运动 SFC	正常任务		运动 主周期中执行			
执行规格		执行任务 (可屏蔽执行。)	固定周期	固定周期执行 (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms)		固定周期执行 (1.77ms, 3.55ms, 7.11ms, 14.2ms)	
			外部中断	中断模块 (QI60) 中设置输入(16 点)为 ON 时执行		中断模块 (A1SI61) 中设置输入(16 点)为 ON 时执行	
			PLC 中断	来自 PLC CPU 中断时执行。		PLC CPU 提供 1 中断点时执行	
NMI 任务		中断模块 (QI60) 中设置输入(16 点)为 ON 时执行		中断模块 (A1SI61) 中设置输入(16 点)为 ON 时执行			
I/O (X/Y) 点数		8192 点			2048 点		
实 I/O (PX/PY) 点数		共 256 点					

1 概述

Q173CPU(N) /Q172CPU(N) 与 A173UHCPU/A172SHCP 的区别 (续)

项目		Q173CPU(N)	Q172CPU(N)	A173UHCPU	A172SHCP 不	
运动 SFC	内部继电器 (M)	M+L 合计: 8192 点		M+L (S) 合计: 8192 点	共 1M+L(S): 2048 点	
	锁存继电器 (L)					
	链接继电器 (B)	8192 点			1024 点	
	信号器 (F)	2048 点			256 点	
	定时器接点 (TT)	—		2048 点	256 点	
	定时器线圈 (TC)	—		2048 点	256 点	
	计数器接点 (CT)	—		1024 点	256 点	
	计数器线圈 (CC)	—		1024 点	256 点	
	专用继电器 (M)	256 点				
	数据寄存器 (D)	8192 点			1024 点	
	链接寄存器(W)	8192 点			1024 点	
	当前值定时器 (T)	—		2048 点	256 点	
	当前值计数器(C)	—		1024 点	256 点	
	特殊寄存器 (D)	256 点				
	运动寄存器 (#)	8192 点				
	惯性定时器 (FT)	1 点 (888 μ s)				
其他	软元件内存	独立		通用		
	PCPU 与 SCPU 的数据交换	多 CPU 自动再生数据交换方式		直接数据交换方式, 软元件内存为 2 口内存。		
	固定参数	每转脉冲数	1 至 2147483647[PLS]		1 至 65535[PLS]	
		每转脉冲总数	在单位为 [PLS] 的情况下 1 至 2147483647[PLS]		在单位为 [PLS] 的情况下 1 至 65535[PLS]	
		单位倍率	—		× 1 倍, × 10 倍, × 100 倍, × 1000 倍	
	PLC 就绪标志 (M2000)	用开关(STOP→RUN)将其打开或开关 RUN 与设置寄存器都处于“1”状态时 M2000 变为 ON。		由 PLC 程序使 M2000 ON		
	紧急停止输入	任意位软元件(PX, M)由参数指定。 (可使用伺服放大器的紧急停止端子)		CPU 基板模块紧急停止 (不可使用伺服放大器的紧急停止端子)		
停电保持功能的内存备份用电池	内置充电电池 (持续关机时间超过 1 月请加装外部电池 (A6BAT/MR-BAT)) (注-3)		A6BAT/MR-BAT			

注-1: 使用分线单元(Q173DV) 或分离电缆(Q173J2B △ CBL □ M/Q173HB △ CBL □ M)。

注-2: 使用 SV22 增量型同步编码器时, 可用 4 个模块; 但连接手动脉冲发生器时, 只能使用 1 个模块。

注-3: 增添外部电池(A6BAT/MR-BAT)时, 可用 Q173DV (使用 Q173CPU(N) 时), 或者 Q170BAT (使用 Q172CPU(N) 时)。

1 概述

1.2.5 定位专用软元件 /特殊继电器 /特殊寄存器

(a) 定位专用软元件

下部分描述了各定位专用软元件的情况。

Q173CPU(N)中 1 至 32 轴有效, Q172CPU(N)中 1 至 8 轴有效。

(b) 内部继电器表

●一般配置

SV13		SV22	
软元件号	作用	软元件号	作用
M0 至	用户软元件 (2000 点)	M0 至	用户软元件 (2000 点)
M2000 至	公共软元件 (320 点)	M2000 至	公共软元件 (320 点)
M2320 至	特殊继电器分配软元件 (状态) (80 点)	M2320 至	特殊继电器分配软元件 (状态) (80 点)
M2400 至	轴状态 (20 点 × 32 轴)	M2400 至	轴状态 (20 点 × 32 轴) 实模式.....各轴 虚模式....输出模块
M3040 至	不可使用	M3040 至	不可使用
M3072 至	公共软元件 (指令信号) (64 点)	M3072 至	公共软元件 (指令信号) (64 点)
M3136 至	特殊继电器分配软元件 (指令信号) (64 点)	M3136 至	特殊继电器分配软元件 (指令信号) (64 点)
M3200 至	轴指令信号 (20 点 × 32 轴)	M3200 至	轴指令信号 (20 点 × 32 轴) 实模式.....各轴 虚模式....输出模块
M3840 至	用户软元件 (960 点)	M3840 至	不可使用 (160 点)
M4799		M4000 至	虚拟伺服电机轴状态 (注-1,2) (20 点 × 32 轴) (仅用于机械结构设定状态轴)
		M4640 至	同步编码器轴状态 (注-2) (4 点 × 12 轴)
		M4688 至	不可使用 (注)
M4799		M4799	

●总配置(续)

SV13		SV22	
软元件号	作用	软元件号	作用
M4800	用户软元件 (3392点)	M4800	虚拟伺服电机轴指令信号(注-1,2) (20点×32轴) (仅用机械结构设定轴)
至		M5440	同步编码器轴 指令信号(注-2) (4点×12轴)
		M5488	不可使用(注-1) (112点)
M8191		M5600 至 M8191	用户软元件 (2592点)

注-1) : 只用于sv22实模式中时, 可作为用户软元件。

注-2) : 应用虚模式时, 不要将M4000至M5999设置为锁存范围。

1 概述

1) 公共软元件 (SV13/SV22) 公共软元件表

SV13			SV22			刷新周期	存取周期	信号指示	备注 (注-3)
软元件号	信号名称		软元件号	信号名称					
M2000	PLC 就绪标志		M2000	PLC 就绪标志		/	主周期	指令信号 (注-1)	M3072
M2001	轴 1	启动接受标志 (32 点)	M2001	轴 1	启动接受标志 (32 点)	运算周期	/	状态信号	
至	至		至	至					
M2032	轴 32		M2032	轴 32					
M2033	不可使用		M2033	不可使用		—	—	—	
M2034	个人电脑链接通讯出错标志		M2034	个人电脑链接通讯出错标志		运算周期	/	状态信号	
M2035	运动 SFC 故障历史清除请求标志 ^(注-4)		M2035	运动 SFC 故障历史清除请求标志 ^(注-4)		/	主周期	指令信号 (注-1)	M3080
M2036	不可使用		M2036	不可使用		—	—	—	
至			至						
M2038			M2038						
M2039	运动 SFC 故障检测标志		M2039	运动 SFC 故障检测标志		/	立即	状态信号	
M2040	速度切换点指定标志		M2040	速度切换点指定标志		/	启动时	指令信号 (注-1)	M3073
M2041	系统设置出错标志		M2041	系统设置出错标志		运算周期	/	状态信号	
M2042	所有轴伺服 ON 指令		M2042	所有轴伺服 ON 指令		/	运行周期	指令信号 (注-1)	M3074
M2043	不可使用		M2043	实/虚模式切换请求		/	虚模式 转移时	状态信号	M3075
M2044			实/虚模式切换状态						
M2045			实/虚模式切换故障检测						
M2046			同步偏差警告						
M2047			运动槽故障检测标志		M2047				运动槽故障检测标志
M2048	JOG 运行同时启动指令		M2048	JOG 运行同时启动 指令		/	主周期	指令信号 (注-1)	M3076
M2049	所有轴伺服 ON 指令		M2049	所有轴伺服 ON 指令		运行周期	/	状态信号	
M2050	启动缓冲器满		M2050	启动缓冲器满					
M2051	手动脉冲发生器 1 使能标志		M2051	手动脉冲发生器 1 使能标志		/	主周期	指令信号 (注-1)	M3077
M2052	手动脉冲发生器 2 使能标志		M2052	手动脉冲发生器 2 使能标志					
M2053	手动脉冲发生器 3 使能标志		M2053	手动脉冲发生器 3 使能标志					M3079

1 概述

公共软元件表 (续)

SV13			SV22			刷新周期	存取周期	信号指示	备注 (注-3)			
软元件号	信号名称		软元件号	信号名称								
M2054	运行周期超过标志		M2054	运行周期超过标志		运行周期	/	状态信号				
M2055	不可使用		M2055	不可使用		—	—	—				
至												
M2060												
M2061	轴 1	速度改变中标志(注-2) (32 轴)	M2061	轴 1	速度改变中标志(注-2) (32 轴)	运行周期	/	状态信号				
至	至		至	至								
M2092	轴 32		M2092	轴 32								
M2093	不可使用		M2093	不可使用		—	—	—				
至			至							M2100		
M2101			轴 1	同步编码器 当前值改变中标志 (注-2) (12 轴)	运行周期	/	状态信号					
至			至					M2112	轴 12			
M2113			不可使用		—	—	—					
至									至			
M2127			M2127									
M2128	轴 1	自动减速标志(注-2) (32 轴)	M2128	轴 1	自动减速标志(注-2) (32 轴)	运行周期	/	状态信号				
至	至		至	至								
M2159	轴 32		M2159	轴 32								
M2160	不可使用		M2160	输出	主轴侧	运行周期	/	状态信号				
至			至	M2161	轴 1					辅助输入轴侧		
M2222			输出	主轴侧	离合器 状态 (注-2)							
M2223			轴 32	辅助输入轴侧								
M2224			不可使用		—					—	—	
至												
M2239			M2239									
M2240	轴 1	速度改变 0 接受中标志 (注-2) (32 轴)	M2240	轴 1	速度改变 0 接受中标志 (注-2) (32 轴)	运行周期	/	状态信号				
至	至		至	至								
M2271	轴 32		M2271	轴 32								
M2272	不可使用		M2272	不可使用		—	—	—				
至			至									
M2319										M2319		

请求寄存器说明

编号	功能	位软元件	请求寄存器
1	PLC 就绪标志	M2000	D704
2	速度切换点指定标志	M2040	D705
3	所有轴伺服 ON 指令	M2042	D706
4	实/虚拟模式切换请求 (仅 SV22)	M2043	D707
5	JOG 运行同时启动指令	M2048	D708
6	手动脉冲发生器 1 使能标志	M2051	D755
7	手动脉冲发生器 2 使能标志	M2052	D756
8	手动脉冲发生器 3 使能标志	M2053	D757

注 1) : D704 至 D708 与 D755 至 D757 寄存器的处理。

因不能通过 QCPU 将每个位接通 / 断开, 上述位软元件被分配到至 D 寄存器。

各寄存器最低位从 0 → 1 时各位软元件开启, 从 1 → 0 时各位软元件关闭。

请通过 S(P).DDR0 与 S(P).DDWR 命令执行上述功能。

关于命令请参考“第 5 章 运动专用 PLC 命令”。

注 2) : 9 轴及 9 轴以上的软元件区域不可使用于 Q172CPU(N)。

注 3) : 同样适用于备注栏中软元件。

注 4) : 可用于 SW6RN-SV13Q□/SV22Q□ (版本 00N 或更新)。



注意

- 在运动 SFC 与 PLC 程序中执行同一软元件时, 后面执行的数据有效。

1 概述

2 特殊继电器分配软元件(状态) (SV13/SV22) 特殊继电器分配软元件(状态)表

软元件号	信号名称	刷新周期	写入周期	信号指示	备注 ^(注)		
M2320	熔丝断检测标志	出现故障		状态信号	M9000		
M2321	AC/DC DOWN 检测标志				M9005		
M2322	电力不足标志				M9006		
M2323	电力不足停电锁定标志				M9007		
M2324	自诊断错误标志				M9008		
M2325	诊断出错标志				M9010		
M2326	常 ON	主周期			M9036		
M2327	常 OFF				M9037		
M2328	时钟数据出错标志	出现故障			M9026		
M2329	PCPU WDT 出错标志				M9073		
M2330	PCPU 就绪标志	请求时			M9074		
M2331	测试模式 ON 标志				M9075		
M2332	外部紧急停止输入标志	运算周期			M9076		
M2333	手动脉冲发生器轴设定出错标志	出现故障时			M9077		
M2334	测试模式请求出错标志				M9078		
M2335	伺服程序设置出错标志				M9079		
M2336	CPU No.1 复位标志	状态改变时			M9240		
M2337	CPU No.2 复位标志				M9241		
M2338	CPU No.3 复位标志				M9242		
M2339	CPU No.4 复位标志				M9243		
M2340	CPU No.1 出错标志				M9244		
M2341	CPU No.2 出错标志				M9245		
M2342	CPU No.3 出错标志				M9246		
M2343	CPU No.4 出错标志				M9247		
M2344	伺服参数读取中标志	请求时			M9105		
M2345	CPU No.1 MULTR 完成标志	命令完成时			M9216		
M2346	CPU No.2 MULTR 完成标志				M9217		
M2347	CPU No.3 MULTR 完成标志				M9218		
M2348	CPU No.4 MULTR 完成标志				M9219		
M2349	不可使用	—			—	—	—
至							
M2399							

(注): 备注栏内状态输出。

3) 各轴状态 (SV13/SV22)

各轴状态表

软元件号	信号名称	软元件号	信号名称
M2400 至 M2419	轴 1 状态	M2720 至 M2739	轴 17 状态
M2420 至 M2439	轴 2 状态	M2740 至 M2759	轴 18 状态
M2440 至 M2459	轴 3 状态	M2760 至 M2779	轴 19 状态
M2460 至 M2479	轴 4 状态	M2780 至 M2799	轴 20 状态
M2480 至 M2499	轴 5 状态	M2800 至 M2819	轴 21 状态
M2500 至 M2519	轴 6 状态	M2820 至 M2839	轴 22 状态
M2520 至 M2539	轴 7 状态	M2840 至 M2859	轴 23 状态
M2540 至 M2559	轴 8 状态	M2860 至 M2879	轴 24 状态
M2560 至 M2579	轴 9 状态	M2880 至 M2899	轴 25 状态
M2580 至 M2599	轴 10 状态	M2900 至 M2919	轴 26 状态
M2600 至 M2619	轴 11 状态	M2920 至 M2939	轴 27 状态
M2620 至 M2639	轴 12 状态	M2940 至 M2959	轴 28 状态
M2640 至 M2659	轴 13 状态	M2960 至 M2979	轴 29 状态
M2660 至 M2679	轴 14 状态	M2980 至 M2999	轴 30 状态
M2680 至 M2699	轴 15 状态	M3000 至 M3019	轴 31 状态
M2700 至 M2719	轴 16 状态	M3020 至 M3039	轴 32 状态

● 各轴详细内容

软元件号	信号名称	
M2400 + 20n	定位启动完成	
M2401 + 20n	定位完成	
M2402 + 20n	到位	
M2403 + 20n	指令到位	
M2404 + 20n	速度控制	
M2405 + 20n	速度/位置切换锁定信号	
M2406 + 20n	零点通过信号	
M2407 + 20n	故障检测信号	
M2408 + 20n	伺服故障检测信号	
M2409 + 20n	原点回归请求信号	
M2410 + 20n	原点回归完成信号	
M2411 + 20n	外部信号	FLS 信号
M2412 + 20n		RLS 信号
M2413 + 20n		STOP 信号
M2414 + 20n		DOG/CHANGE 信号
M2415 + 20n	伺服就绪信号	
M2416 + 20n	转矩限制中信号	
M2417 + 20n	不可使用	
M2418 + 20n	虚模式不可继续运行警告信号 (SV22)	
M2419 + 20n	M-码输出信号	

注①：上面软元件号中的 n 与轴号相对应。

Q173CPU(N)：轴 No.1 至 No.32 ($n=0$ 至 31)

Q172CPU(N)：轴 No.1 至 No.8 ($n=0$ 至 7)

注②：9 轴或多于 9 轴的软元件区域不可用于 Q172CPU(N) 中。

4) 公共软元件 (指令信号) (SV13/SV22)

公共软元件 (指令信号)表

软元件号	信号名称	刷新周期	写入周期	信号指示	备注 (注-1), (注-2)
M3072	PLC 就绪标志		主周期	指令 信号	M2000
M3073	速度切换点指定标志		启动		M2040
M3074	所有轴伺服 ON 指令		运行周期		M2042
M3075	实/虚模式切换请求		虚模式转换		M2043
M3076	JOG 运行同时启动指令		主周期		M2048
M3077	手动脉冲发生器 1 使能标志				M2051
M3078	手动脉冲发生器 2 使能标志				M2052
M3079	手动脉冲发生器 3 使能标志				M2053
M3080	运动 SFC 故障历史清除请求标志 ^(注-3)				M2035
M3081	不可使用		—		—
至					
M3135					

注-1: 使上述软元件从 OFF 到 ON 时, 即接通了与之对应的备注栏中软元件。将 ON 变到 OFF 则断开了那些软元件。

直接 ON/OFF 备注栏中软元件时, 软元件状态不一致。此外, 如数据寄存器的请求与上述软元件的请求同时进行, 则后者有效。

注-2: 同样适用于备注栏中各软元件。

注-3: 可用于 SW6RN-SV13Q□/SV22Q□ (00N 或更高)。

M3080 不能自动断开, 用户需自行断开。

5) 特殊继电器分配软元件(指令信号) (SV13/SV22)

特殊继电器分配软元件表(指令信号)

软元件号	信号名称	再生周期	固定周期	信号指示	备注 (注-1),(注-2)
M3136	时钟数据设置请求	/	主周期	指令 信号	M9025
M3137	时钟数据读取请求				M9028
M3138	故障复位				M9060
M3139	伺服参数读取请求标志				M9104
M3140	不可使用	—	—	—	—
至					
M3199					

注 ①：使上述软元件从 OFF 到 ON 时，即开启了与之对应的备注栏中软元件。从 ON 变到 OFF 则断开了那些软元件。
直接 ON/OFF 备注栏中软元件时，软元件状态不一致。

注 ②：同样适用于备注栏中各软元件。

6 轴指令信号 (SV13/SV22)

轴指令信号表

软元件号	信号名称	软元件号	信号名称
M3200 至 M3219	轴 1 指令信号	M3520 至 M3539	轴 17 指令信号
M3220 至 M3239	轴 2 指令信号	M3540 至 M3559	轴 18 指令信号
M3240 至 M3259	轴 3 指令信号	M3560 至 M3579	轴 19 指令信号
M3260 至 M3279	轴 4 指令信号	M3580 至 M3599	轴 20 指令信号
M3280 至 M3299	轴 5 指令信号	M3600 至 M3619	轴 21 指令信号
M3300 至 M3319	轴 6 指令信号	M3620 至 M3639	轴 22 指令信号
M3320 至 M3339	轴 7 指令信号	M3640 至 M3659	轴 23 指令信号
M3340 至 M3359	轴 8 指令信号	M3660 至 M3679	轴 24 指令信号
M3360 至 M3379	轴 9 指令信号	M3680 至 M3699	轴 25 指令信号
M3380 至 M3399	轴 10 指令信号	M3700 至 M3719	轴 26 指令信号
M3400 至 M3419	轴 11 指令信号	M3720 至 M3739	轴 27 指令信号
M3420 至 M3439	轴 12 指令信号	M3740 至 M3759	轴 28 指令信号
M3440 至 M3459	轴 13 指令信号	M3760 至 M3779	轴 29 指令信号
M3460 至 M3479	轴 14 指令信号	M3780 至 M3799	轴 30 指令信号
M3480 至 M3499	轴 15 指令信号	M3800 至 M3819	轴 31 指令信号
M3500 至 M3519	轴 16 指令信号	M3820 至 M3839	轴 32 指令信号

● 各轴详细内容

软元件号	SV13	SV22
M3200 + 20n	停止指令	停止指令
M3201 + 20n	快速停止指令	快速停止指令
M3202 + 20n	正转 JOG 启动指令	正转 JOG 启动指令
M3203 + 20n	反转 JOG 启动指令	倒转 JOG 启动指令
M3204 + 20n	完成信号 OFF 指令	完成信号 OFF 指令
M3205 + 20n	速度/位置切换使能指令	速度/定位切换使能指令
M3206 + 20n	不可使用	不可使用
M3207 + 20n	故障复位指令	故障复位指令
M3208 + 20n	伺服故障复位指令	伺服故障复位指令
M3209 + 20n	启动时外部停止输入无效指令	启动时外部停止输入无效指令
M3210 + 20n	不可使用	不可使用
M3211 + 20n		
M3212 + 20n	进给当前值更新请求指令	进给当前值更新请求指令
M3213 + 20n	不可使用	地址离合器参考设置指令
M3214 + 20n		凸轮参考定位设置指令
M3215 + 20n	伺服 OFF 指令	伺服 OFF 指令
M3216 + 20n	不可使用	不可使用
M3217 + 20n		
M3218 + 20n		
M3219 + 20n	FIN 信号	FIN 信号

注 ①：上述软元件号中的 n 与轴号相对应。

Q173CPU(N)：轴 No.1 至 No.32 ($n=0$ 至 31)

Q172CPU(N)：轴 No.1 至 No.8 ($n=0$ 至 7)

注 ②：9 轴或多于 9 轴的软元件区域不可用于 Q172CPU(N) 中。

7) 虚拟伺服电机轴状态 (仅 SV22)

虚拟伺服电机轴状态表

软元件号	信号名称	软元件号	信号名称
M4000 至 M4019	轴 1 状态	M4320 至 M4339	轴 17 状态
M4020 至 M4039	轴 2 状态	M4340 至 M4359	轴 18 状态
M4040 至 M4059	轴 3 状态	M4360 至 M4379	轴 19 状态
M4060 至 M4079	轴 4 状态	M4380 至 M4399	轴 20 状态
M4080 至 M4099	轴 5 状态	M4400 至 M4419	轴 21 状态
M4100 至 M4119	轴 6 状态	M4420 至 M4439	轴 22 状态
M4120 至 M4139	轴 7 状态	M4440 至 M4459	轴 23 状态
M4140 至 M4159	轴 8 状态	M4460 至 M4479	轴 24 状态
M4160 至 M4179	轴 9 状态	M4480 至 M4499	轴 25 状态
M4180 至 M4199	轴 10 状态	M4500 至 M4519	轴 26 状态
M4200 至 M4219	轴 11 状态	M4520 至 M4539	轴 27 状态
M4220 至 M4239	轴 12 状态	M4540 至 M4559	轴 28 状态
M4240 至 M4259	轴 13 状态	M4560 至 M4579	轴 29 状态
M4260 至 M4279	轴 14 状态	M4580 至 M4599	轴 30 状态
M4280 至 M4299	轴 15 状态	M4600 至 M4619	轴 31 状态
M4300 至 M4319	轴 16 状态	M4620 至 M4639	轴 32 状态

●各轴详细内容

软元件号	信号名称
M4000 + 20n	定位启动完成
M4001 + 20n	定位完成
M4002 + 20n	不可使用
M4003 + 20n	指令到位
M4004 + 20n	速度控制
M4005 + 20n	不可使用
M4006 + 20n	
M4007 + 20n	故障检测
M4008 + 20n	不可使用
M4009 + 20n	
M4010 + 20n	
M4011 + 20n	
M4012 + 20n	
M4013 + 20n	
M4014 + 20n	
M4015 + 20n	
M4016 + 20n	
M4017 + 20n	
M4018 + 20n	
M4019 + 20n	M-码输出信号

注 ①：上面软元件号中的 n 与轴号相对应。

Q173CPU(N)：轴 No.1 至 No.32 (n=0 至 31)

Q172CPU(N)：轴 No.1 至 No.8 (n=0 至 7)

注 ②：9 轴 或多于 9 轴的软元件区域 不可使用于 Q172CPU(N) 中。

注 ③：机械结构程序中不可使用的轴区域可被作用户软元件。

8 虚拟伺服电机轴指令信号 (仅 SV22)

虚拟伺服电机轴指令信号表

软元件号	信号名称	软元件号	信号名称
M4800 至 M4819	轴 1 指令信号	M5120 至 M5139	轴 17 指令信号
M4820 至 M4839	轴 2 指令信号	M5140 至 M5159	轴 18 指令信号
M4840 至 M4859	轴 3 指令信号	M5160 至 M5179	轴 19 指令信号
M4860 至 M4879	轴 4 指令信号	M5180 至 M5199	轴 20 指令信号
M4880 至 M4899	轴 5 指令信号	M5200 至 M5219	轴 21 指令信号
M4900 至 M4919	轴 6 指令信号	M5220 至 M5239	轴 22 指令信号
M4920 至 M4939	轴 7 指令信号	M5240 至 M5259	轴 23 指令信号
M4940 至 M4959	轴 8 指令信号	M5260 至 M5279	轴 24 指令信号
M4960 至 M4979	轴 9 指令信号	M5280 至 M5299	轴 25 指令信号
M4980 至 M4999	轴 10 指令信号	M5300 至 M5319	轴 26 指令信号
M5000 至 M5019	轴 11 指令信号	M5320 至 M5339	轴 27 指令信号
M5020 至 M5039	轴 12 指令信号	M5340 至 M5359	轴 28 指令信号
M5040 至 M5059	轴 13 指令信号	M5360 至 M5379	轴 29 指令信号
M5060 至 M5079	轴 14 指令信号	M5380 至 M5399	轴 30 指令信号
M5080 至 M5099	轴 15 指令信号	M5400 至 M5419	轴 31 指令信号
M5100 至 M5119	轴 16 指令信号	M5420 至 M5439	轴 32 指令信号

● 各轴详细内容

软元件号	信号名称
M4800 + 20n	停止指令
M4801 + 20n	快速停止指令
M4802 + 20n	正转 JOG 开始指令
M4803 + 20n	反转 JOG 开始指令
M4804 + 20n	完成信号 OFF 指令
M4805 + 20n	不可用
M4806 + 20n	
M4807 + 20n	故障复位指令
M4808 + 20n	不可用
M4809 + 20n	启动时外部停止输入无效指令
M4810 + 20n	不可用
M4811 + 20n	
M4812 + 20n	
M4813 + 20n	
M4814 + 20n	
M4815 + 20n	
M4816 + 20n	
M4817 + 20n	
M4818 + 20n	
M4819 + 20n	

注 1) : 上述软元件号中 n 的取值和轴编号值相对应。

Q173CPU(N) : 轴 No.1 至 No.32 ($n=0$ 至 31)

Q172CPU(N) : 轴 No.1 至 No.8 ($n=0$ 至 7)

注 2) : 9 个或 9 个以上的软元件不能用于 Q172CPU(N)。

注 3) : 机械结构程序中未用的轴存储区域可作为用户软元件使用。

9) 同步编码器轴状态 (仅 SV22)

同步编码器轴状态表

软件件号	信号名称	
M4640	轴 1	故障监测
M4641		外部信号 TREN
M4642		虚模式不能继续运行警告
M4643		不可用
M4644	轴 2	故障监测
M4645		外部信号 TREN
M4646		虚模式不能继续运行警告
M4647		不可用
M4648	轴 3	故障监测
M4649		外部信号 TREN
M4650		虚模式不能继续运行警告
M4651		不可用
M4652	轴 4	故障监测
M4653		外部信号 TREN
M4654		虚模式不能继续运行警告
M4655		不可用
M4656	轴 5	故障监测
M4657		外部信号 TREN
M4658		虚模式不能继续运行警告
M4659		不可用
M4660	轴 6	故障监测
M4661		外部信号 TREN
M4662		虚模式不能继续运行警告
M4663		不可用
M4664	轴 7	故障监测
M4665		外部信号 TREN
M4666		虚模式不能继续运行警告
M4667		不可用
M4668	轴 8	故障监测
M4669		外部信号 TREN
M4670		虚模式不能继续运行警告
M4671		不可用
M4672	轴 9	故障监测
M4673		外部信号 TREN
M4674		虚模式不能继续运行警告
M4675		不可用
M4676	轴 10	故障监测
M4677		外部信号 TREN
M4678		虚模式不能继续运行警告
M4679		不可用
M4680	轴 11	故障监测
M4681		外部信号 TREN
M4682		虚模式不能继续运行警告
M4683		不可用
M4684	轴 12	故障监测
M4685		外部信号 TREN
M4686		虚模式不能继续运行警告
M4687		不可用

10) 同步编码器轴指令信号 (仅 SV22)

同步编码器轴指令信号表

软件元件号	信号名	
M5440	轴 1	故障复位
M5441		不可用
M5442		不可用
M5443		不可用
M5444	轴 2	故障复位
M5445		不可用
M5446		不可用
M5447		不可用
M5448	轴 3	故障复位
M5449		不可用
M5450		不可用
M5451		不可用
M5452	轴 4	故障复位
M5453		不可用
M5454		不可用
M5455		不可用
M5456	轴 5	故障复位
M5457		不可用
M5458		不可用
M5459		不可用
M5460	轴 6	故障复位
M5461		不可用
M5462		不可用
M5463		不可用
M5464	轴 7	故障复位
M5465		不可用
M5466		不可用
M5467		不可用
M5468	轴 8	故障复位
M5469		不可用
M5470		不可用
M5471		不可用
M5472	轴 9	故障复位
M5473		不可用
M5474		不可用
M5475		不可用
M5476	轴 10	故障复位
M5477		不可用
M5478		不可用
M5479		不可用
M5480	轴 11	故障复位
M5481		不可用
M5482		不可用
M5483		不可用
M5484	轴 12	故障复位
M5485		不可用
M5486		不可用
M5487		不可用

b) 数据寄存器表

● 一般配置

SV13		SV22	
软元件号	应用	软元件号	应用
D0 至	各轴监视软元件 (20 点 × 32 轴)	D0 至	各轴监视软元件 (20 点 × 32 轴) 实模式.....各轴 虚模式.....输出模块
D640 至	控制改变寄存器 (2 点 × 32 轴)	D640 至	控制改变寄存器 (2 点 × 32 轴)
D704 至	共用软元件 (指令信号) (54 点)	D704 至	共用软元件 (指令信号) (54 点)
D758 至	共用软元件 (监视器) (42 点)	D758 至	共用软元件 (监视器) (42 点)
D800 至	用户软元件 (7392 点)	D800 至	虚拟伺服电机轴监视软元件 (注) (10 点 × 32 轴) (仅限用于机械系统设轴)
		D1120 至	同步编码器轴监视软元件 (注) (10 点 × 12 轴)
		D1240 至	凸轮轴监视软元件 (注) (10 点 × 32 轴)
		D1560 至	用户软元件 (6632 点)
D8191		D8191	

(注): 该软元件仅在 SV22 实模式时可作为用户软元件使用。

1) 各轴监视软元件 (SV13/SV22)

各轴监视软元件表

软元件号	信号名称	软元件号	信号名称
D0 至 D19	轴 1 监视软元件	D320 至 D339	轴 17 监视软元件
D20 至 D39	轴 2 监视软元件	D340 至 D359	轴 18 监视软元件
D40 至 D59	轴 3 监视软元件	D360 至 D379	轴 19 监视软元件
D60 至 D79	轴 4 监视软元件	D380 至 D399	轴 20 监视软元件
D80 至 D99	轴 5 监视软元件	D400 至 D419	轴 21 监视软元件
D100 至 D119	轴 6 监视软元件	D420 至 D439	轴 22 监视软元件
D120 至 D139	轴 7 监视软元件	D440 至 D459	轴 23 监视软元件
D140 至 D159	轴 8 监视软元件 A	D460 至 D479	轴 24 监视软元件
D160 至 D179	轴 9 监视软元件	D480 至 D499	轴 25 监视软元件
D180 至 D199	轴 10 监视软元件	D500 至 D519	轴 26 监视软元件
D200 至 D219	轴 11 监视软元件	D520 至 D539	轴 27 监视软元件
D220 至 D239	轴 12 监视软元件	D540 至 D559	轴 28 监视软元件
D240 至 D259	轴 13 监视软元件	D560 至 D579	轴 29 监视软元件
D260 至 D279	轴 14 监视软元件	D580 至 D599	轴 30 监视软元件
D280 至 D299	轴 15 监视软元件	D600 至 D619	轴 31 监视软元件
D300 至 D319	轴 16 监视软元件	D620 至 D639	轴 32 监视软元件

● 各轴详细内容

软元件号	SV13/SV22(实模式)	SV22(虚模式)	信号检测	
D0 + 20n D1 + 20n	进给当前值	进给当前值/滚筒周速	监视软元件	
D2 + 20n D3 + 20n	实当前值	实当前值		
D4 + 20n D5 + 20n	偏差计数器值	偏差计数器值		
D6 + 20n	轻微错误代码	轻微错误代码		
D7 + 20n	严重错误代码	严重错误代码		
D8 + 20n	伺服错误代码	伺服错误代码		
D9 + 20n	原点回归再移动量	保持		
D10 + 20n D11 + 20n	近点 DOG ON 后移动量	保持		
D12 + 20n	执行程序号	—		
D13 + 20n	M-代码	—		
D14 + 20n	转矩限制值	转矩限制值		
D15 + 20n	恒速控制的数据设置点	—		
D16 + 20n D17 + 20n	移动量变更寄存器	—		指令软元件
D18 + 20n D19 + 20n	停止输入时当前值	保持		监视软元件

①：以上软元件号中 n 的取值与轴编号值相一致。

Q173CPU(N)：轴 No.1 至 No.32 (n=0 至 31)

Q172CPU(N)：轴 No.1 至 No.8 (n=0 至 7)

②：9 个或 9 个以上的软元件空间不可用于 Q172CPU(N)

2) 控制改变寄存器 (SV13/SV22)

控制改变寄存器表

软元件号	信号名称	软元件号	信号名称
D640 D641	轴 1 JOG 速度设定寄存器	D672 D673	轴 17 JOG 速度设定寄存器
D642 D643	轴 2 JOG 速度设定寄存器	D674 D675	轴 18 JOG 速度设定寄存器
D644 D645	轴 3 JOG 速度设定寄存器	D676 D677	轴 19 JOG 速度设定寄存器
D646 D647	轴 4 JOG 速度设定寄存器	D678 D679	轴 20 JOG 速度设定寄存器
D648 D649	轴 5 JOG 速度设定寄存器	D680 D681	轴 21 JOG 速度设定寄存器
D650 D651	轴 6 JOG 速度设定寄存器	D682 D683	轴 22 JOG 速度设定寄存器
D652 D653	轴 7 JOG 速度设定寄存器	D684 D685	轴 23 JOG 速度设定寄存器
D654 D655	轴 8 JOG 速度设定寄存器	D686 D687	轴 24 JOG 速度设定寄存器
D656 D657	轴 9 JOG 速度设定寄存器	D688 D689	轴 25 JOG 速度设定寄存器
D658 D659	轴 10 JOG 速度设定寄存器	D690 D691	轴 26 JOG 速度设定寄存器
D660 D661	轴 11 JOG 速度设定寄存器	D692 D693	轴 27 JOG 速度设定寄存器
D662 D663	轴 12 JOG 速度设定寄存器	D694 D695	轴 28 JOG 速度设定寄存器
D664 D665	轴 13 JOG 速度设定寄存器	D696 D697	轴 29 JOG 速度设定寄存器
D666 D667	轴 14 JOG 速度设定寄存器	D698 D699	轴 30 JOG 速度设定寄存器
D668 D669	轴 15 JOG 速度设定寄存器	D700 D701	轴 31 JOG 速度设定寄存器
D670 D671	轴 16 JOG 速度设定寄存器	D702 D703	轴 32 JOG 速度设定寄存器

1 概述

3) 通用软元件

通用软元件表

软元件号	信号名称	信号检测	软元件号	信号名称	信号检测	
D704	PLC 就绪标志请求	指令软元件	D740	轴 21	指令软元件	
D705	速度切换点指定标志请求		D741	轴 22		
D706	所有轴伺服 ON 指令请求		D742	轴 23		
D707	实/虚模式切换请求(SV22)		D743	轴 24		
D708	JOG 运行同时启动指令请求		D744	轴 25		
D709	不可用		D745	轴 26		
D710	JOG 操作同时启动轴设定寄存器		D746	轴 27		
至			D747	轴 28		
D713			D748	轴 29		
D714	手动脉冲发生器轴 1 设置寄存器		D749	轴 30		
D715			D750	轴 32		
D716	手动脉冲发生器轴 2 设置寄存器		D751	轴 32		
D717			D752	手动脉冲发生器 1 平滑倍率设置寄存器		
D718	手动脉冲发生器轴 3 设置寄存器		D753	手动脉冲发生器 2 平滑倍率设置寄存器		
D719			D754	手动脉冲发生器 3 平滑倍率设置寄存器		
D720	轴 1		D755	手动脉冲发生器 1 使能标志请求		
D721	轴 2		D756	手动脉冲发生器 2 使能标志请求		
D722	轴 3		D757	手动脉冲发生器 3 使能标志请求		
D723	轴 4		D758	不可用	—	
D724	轴 5		D759	PCPU 就绪完成标志状态 (0 : OFF/1 : ON)	监视软元件	
D725	轴 6		D760	不可用	—	
D726	轴 7		至			
D727	轴 8		D791	伺服放大器类型	监视软元件	
D728	轴 9		至			
D729	轴 10		D799			
D730	轴 11		手动脉冲发生器 1 脉冲输入倍率设置寄存器	/		
D731	轴 12					
D732	轴 13					
D733	轴 14					
D734	轴 15					
D735	轴 16					
D736	轴 17					
D737	轴 18					
D738	轴 19					
D739	轴 20					

4) 虚拟伺服电机轴监视软元件 (仅 SV22)

虚拟伺服电机轴监视软元件表

软元件号	信号名称	软元件号	信号名称
D800 至 D809	轴 1 监视软元件	D960 至 D969	轴 17 监视软元件
D810 至 D819	轴 2 监视软元件	D970 至 D979	轴 18 监视软元件
D820 至 D829	轴 3 监视软元件	D980 至 D989	轴 19 监视软元件
D830 至 D839	轴 4 监视软元件	D990 至 D999	轴 20 监视软元件
D840 至 D849	轴 5 监视软元件	D1000 至 D1009	轴 21 监视软元件
D850 至 D859	轴 6 监视软元件	D1010 至 D1019	轴 22 监视软元件
D860 至 D869	轴 7 监视软元件	D1020 至 D1029	轴 23 监视软元件
D870 至 D879	轴 8 监视软元件	D1030 至 D1039	轴 24 监视软元件
D880 至 D889	轴 9 监视软元件	D1040 至 D1049	轴 25 监视软元件
D890 至 D899	轴 10 监视软元件	D1050 至 D1059	轴 26 监视软元件
D900 至 D909	轴 11 监视软元件	D1060 至 D1069	轴 27 监视软元件
D910 至 D919	轴 12 监视软元件	D1070 至 D1079	轴 28 监视软元件
D920 至 D929	轴 13 监视软元件	D1080 至 D1089	轴 29 监视软元件
D930 至 D939	轴 14 监视软元件	D1090 至 D1099	轴 30 监视软元件
D940 至 D949	轴 15 监视软元件	D1100 至 D1109	轴 31 监视软元件
D950 至 D959	轴 16 监视软元件	D1110 至 D1119	轴 32 监视软元件

● 各轴详细内容

软元件号	信号名称
D800 + 10n D801 + 10n	当前值
D802 + 10n	轻微错误代码
D803 + 10n	严重错误代码
D804 + 10n	执行程序号
D805 + 10n	M 代码
D806 + 10n D807 + 10n	主轴差速齿轮后的当前值
D808 + 10n	故障检测输出轴号
D809 + 10n	不可用

注-1)：上述软元件号中 n 的取值和轴的编号值相一致。

Q173CPU(N)：轴 No.1 至 No.32 (n=0 至 31)

注-2)：Q172CPU(N)：轴 No.1 至 No.8 (n=0 至 7)

注-3)：轴以上的软元件区域不可用于 Q172CPU(N)。

机械结构程序中未用的轴存储区域可作为用户软元件使用。

5) 同步编码器轴监视软元件（仅限SV22）

同步编码器轴监视软元件一览表

软元件号	信号名称
D1120 至 D1129	轴 1 监视软元件
D1130 至 D1139	轴 2 监视软元件
D1140 至 D1149	轴 3 监视软元件
D1150 至 D1159	轴 4 监视软元件
D1160 至 D1169	轴 5 监视软元件
D1170 至 D1179	轴 6 监视软元件
D1180 至 D1189	轴 7 监视软元件
D1190 至 D1199	轴 8 监视软元件
D1200 至 D1209	轴 9 监视软元件
D1210 至 D1219	轴 10 监视软元件
D1220 至 D1229	轴 11 监视软元件
D1230 至 D1239	轴 12 监视软元件

● 各轴详细内容

软元件号	信号名称
D1120 + 10n D1121 + 10n	当前值
D1122 + 10n	轻度错误代码
D1123 + 10n	严重错误代码
D1124 + 10n D1125 + 10n	不可用
D1126 + 10n D1127 + 10n	主轴差速齿轮后的当前值
D1128 + 10n	故障检测输出轴编号
D1129 + 10n	不可用

注 ①：上述软元件号中的 n 表示与轴号相对应的数值

Q173CPU(N)：轴编号 1 至 编号 32 ($n=0$ 至 31)

Q172CPU(N)：轴编号 1 至 编号 8 ($n=0$ 至 7)

注 ②：9 轴以上的软元件区域不可用于 Q172CPU(N)。

6 凸轮轴监视软元件 (仅 SV22)

凸轮轴监视软元件一览表

软元件号	信号名称	软元件号	信号名称
D1240 至 D1249	轴 1 监视软元件	D1400 至 D1409	轴 17 监视软元件
D1250 至 D1259	轴 2 监视软元件	D1410 至 D1419	轴 18 监视软元件
D1260 至 D1269	轴 3 监视软元件	D1420 至 D1429	轴 19 监视软元件
D1270 至 D1279	轴 4 监视软元件	D1430 至 D1439	轴 20 监视软元件
D1280 至 D1289	轴 5 监视软元件	D1440 至 D1449	轴 21 监视软元件
D1290 至 D1299	轴 6 监视软元件	D1450 至 D1459	轴 22 监视软元件
D1300 至 D1309	轴 7 监视软元件	D1460 至 D1469	轴 23 监视软元件
D1310 至 D1319	轴 8 监视软元件	D1470 至 D1479	轴 24 监视软元件
D1320 至 D1329	轴 9 监视软元件	D1480 至 D1489	轴 25 监视软元件
D1330 至 D1339	轴 10 监视软元件	D1490 至 D1499	轴 26 监视软元件
D1340 至 D1349	轴 11 监视软元件	D1500 至 D1509	轴 27 监视软元件
D1350 至 D1359	轴 12 监视软元件	D1510 至 D1519	轴 28 监视软元件
D1360 至 D1369	轴 13 监视软元件	D1520 至 D1529	轴 29 监视软元件
D1370 至 D1379	轴 14 监视软元件	D1530 至 D1539	轴 30 监视软元件
D1380 至 D1389	轴 15 监视软元件	D1540 至 D1549	轴 31 监视软元件
D1390 至 D1399	轴 16 监视软元件	D1550 至 D1559	轴 32 监视软元件

● 各轴详细内容

软元件号	信号名
D1240 + 10n	不可用
D1241 + 10n	执行凸轮编号
D1242 + 10n	执行行程值
D1243 + 10n	
D1244 + 10n	凸轮轴 1 转内当前值
D1245 + 10n	
D1246 + 10n	不可用
D1247 + 10n	
D1248 + 10n	
D1249 + 10n	

注-1：上述软元件号中 n 的取值和轴的编号值相一致。

Q173CPU(N)：轴 No.1 至 No.32 ($n=0$ 至 31)

Q172CPU(N)：轴 No.1 至 No.8 ($n=0$ 至 7)

注-2：9 轴以上的软元件区域不可用于 Q172CPU(N)。

注-3：在机械结构程序中不可使用的轴区域可被用作用户软元件。

② 特殊继电器

特殊继电器是运动 CPU 内部规格固定的内部继电器。因此，它们不能以与普通内部继电器一样的方式应用于运动 SFC 程序。

不过，如有需要，它们可以 ON/OFF 以便控制运动 CPU。

下文各表中的标题意义如下所示。

项目	说明
编号	• 表示特殊继电器的软元件号。
名称	• 表示特殊继电器的名称。
内容	• 表示特殊继电器的内容。
详细内容	• 表示特殊继电器功能的详细信息。
设置方(设置时)	<ul style="list-style-type: none"> • 表示继电器是否由系统或用户设置，以及由系统设置时执行设置。 <设置方法> S：由系统设置(运动 CPU) U：由用户设置 (利用运动 SFC 程序或外部设备的测试操作) S/U：由系统(运动 CPU)和用户设置 <设置后> 仅限于系统(运动 CPU)设置 主进程：每个主处理过程中反复设置(CPU 的空闲时间处理) 初始进程：仅限初始时设置(当电源开启或执行重启时) 状态变化：只在状态发生变化时设置 故障：当发生故障时设置 请求：只在用户请求(特殊继电器等)时设置 运算周期：运动 CPU 的运算周期中设置

1 概述

特殊继电器列表

编号	名称	意义	详细内容	设置方 (设置时间)	备注	
M9000	熔丝烧熔检测	OFF : 正常 ON : 有熔丝烧熔模块	• 当一个以上自 CPU 管理的输出模块熔丝烧熔时开启。 恢复正常状态仍保持接通状态。	S(发生故障)		
M9005	AC/DC DOWN 检测	OFF : 无 AC/DC DOWN ON : 有 AC/DC DOWN	• 应用 AC 电源模块过程中如发生不多于 20 毫秒的电力暂时中断时开启, 并通过电源从关到开进行复位。 • 应用 DC 电源模块过程中如发生不多于 10 毫秒的电力暂时中断时开启, 并通过电源从关到开进行复位。			
M9006	电量不足	OFF : 正常 ON : 电量不足	• 当外接电池组电压低于规定值时开启, 待外接电池组电压恢复正常后断开。 • 同时发生 "BAT. LED"。 • 只有当其由系统设置为"应用外接电池组"时, 检测外部电池组电压。			
M9007	电量不足锁存器	OFF : 正常 ON : 电量不足	• 当外接电池组电压低于规定值时开启, 待外接电池组电压恢复正常后断开。 • 同步发生 "BAT. LED"。 • 只有当其由系统设置为"应用外接电池组"时, 检测外部电池组电压。			
M9008	自诊断故障	OFF : 正常 ON : 故障	• 自诊断发现故障时开启。 恢复正常状态后仍保持开启状态。			
M9010	故障诊断	OFF : 正常 ON : 故障	• 诊断发现故障时开启。 恢复正常状态后仍保持开启状态。			新 ^(注-1)
M9025	时钟数据设置请求	OFF : 忽略 ON : 有设置请求	• 当 M9025 从关变为开时, 将存储于 D9205 至 D9028 的时钟数据写到时钟软元件中。			U
M9026	时钟数据故障	OFF : 正常 ON : 故障	• 由时钟数据 (D9025 至 D9028) 值有错误开启。			S(请求时)
M9028	时钟数据读取请求	OFF : 忽略 ON : 读取请求	• 当 M9028 开启时, 以 BCD 码形式读取 D9025 至 D9028 的时钟数据。	U		
M9036	常开	ON _____ OFF _____	• 常开, 与 RUN/STOP 开关状态无关的情况。	S(主进程)		
M9037	常闭	ON _____ OFF _____	• 常闭, 不考虑 RUN/STOP 开关状态无关的情况。			
M9060	故障清除	OFF → ON : 故障清除	• 清除故障。	U	新 ^(注-1)	
M9073	PCPU WDT 故障标志	ON : 异常 OFF : 正常	• 当运动 CPU 自我检测功能检测到监视时钟故障时开启, 当运动 CPU 检测到一个 WDT 故障时启动中的轴不减速立即停止 • 故障原因存储在"运动 CPU WDT 故障原因 (D9184)"中。	S(发生故障)		
M9074	PCPU READY 完成标志	ON : PCPU READY 完成 OFF : PCPU READY 未完成	• 当 PLC 就绪标志从关到开时, 检测固定参数伺服参数以及限位开关输出数据等, 并且如无故障被检测到, 此标志为 ON。 • PLC 就绪(M2000)信号关闭时关闭。	S(请求)		
M9075	测试模式 ON 标志	ON : 正在运行 TEST 模式 OFF : 停止 TEST 模式运行	• 此标志状态用于判别从外围设备建立的 TEST 模式当前运行与否。 • 如有来自外围设备的 TEST 模式请求时, 没有进入 TEST 模式。"TEST 模式请求故障标志"将开启。	S(请求)		
M9076	外部紧急停止输入标志	ON : 紧急停止输入 OFF 中 OFF : 紧急停止输入 ON 中	• 此标志状态表示紧急停止的 ON/OFF。	S(运算周期)		

(注-1) : Q 系列运动控制器新增加。

特殊继电器列表 (续)

编号	名称	意义	详细内容	设置方 (设置时)	备注	
M9077	手动脉冲发生器轴设置故障标志	ON : 在 D714 至 D719 中至少有 1 个异常。 OFF : 从 D714 至 D719 所有设置都正常。	<ul style="list-style-type: none"> 判别在手动脉冲发生器轴设置寄存器中的设置是否正常。 当此继电器变 ON,故障内容存储在手动脉冲发生器轴设置故障寄存器 (D9185 至 D9187)。 	S(发生故障)		
M9078	TEST 模式请求故障标志	ON : 异常 OFF : 正常	<ul style="list-style-type: none"> 如有来自外围设备的 TEST 模式请求时, 没有进入 TEST 模式。 此继电器变 ON 时,故障内容存储在 TEST 模式请求故障寄存器(D9182 至 D9183)。 	S(发生故障)		
M9079	伺服程序设置故障标志	ON : 异常 OFF : 正常	<ul style="list-style-type: none"> 此标志状态表示运动 SFC 程序指定的伺服系统定位数据是否正常,如检测到故障,此标志便会变 ON。 伺服程序的故障内容存储在 D9189 和 D9190。 	S(发生故障)		
M9104	伺服参数读取请求标志	OFF 至 ON : 读取伺服参数	<ul style="list-style-type: none"> 从关到开的过程中 D9104 所设置的伺服参数读取请求轴的伺服参数表现在伺服系统放大器运动 CPU 上。 	U		
M9105	伺服参数读取中标志	ON : 读取伺服参数中 OFF : 未读取伺服参数	<ul style="list-style-type: none"> 此标志在读取伺服参数到运动 CPU 过程中变 ON,在读取完成后自动断开。 	S(读取)		
M9216	CPU No.1 MULTR 完成标志	OFF 至 ON : CPU No.1 完成读取	<ul style="list-style-type: none"> 通过 MULTR 指令从 CPU No.1 中正常读取数据时变 ON。 	S(完成读取)		
M9217	CPU No.2 MULTR 完成标志	OFF 至 ON : CPU No.2 完成读取	<ul style="list-style-type: none"> 通过 MULTR 指令从 CPU No.2 中正常读取数据时变 ON。 			
M9218	CPU No.3 MULTR 完成标志	OFF 至 ON : CPU NO.3 完成读取	<ul style="list-style-type: none"> 通过 MULTR 指令从 CPU No.3 中正常读取数据时变 ON。 			
M9219	CPU No.4 MULTR 完成标志	OFF 至 ON : CPU NO.4 完成读取	<ul style="list-style-type: none"> 通过 MULTR 指令从 CPU No.4 中正常读取数据时变 ON。 			
M9240	CPU No.1 复位标志	OFF : CPU No.1 复位释放 ON : CPU No.1 复位	<ul style="list-style-type: none"> CPU No.1 复位解除时为 OFF。 CPU No.1 在复位中时, 为 ON(也包括当一个 CPU 从其基板上拆下的情况)。 其他 CPU 也进行复位。 	S(状态变化)	新 ^(注-1)	
M9241	CPU No.2 复位标志	OFF : CPU No.2 复位释放 ON : CPU No.2 复位	<ul style="list-style-type: none"> 2 号 CPU 复位解除时为 OFF。 CPU No.1 在复位中时, 为 ON(也包括当一个 CPU 从其基板上拆下的情况)。 其他 CPU 发生“多 CPU DOWN(错误代码:7000)”错误。 			
M9242	CPU No.3 复位标志	OFF : CPU No.3 复位释放 ON : CPU No.3 复位	<ul style="list-style-type: none"> 3 号 CPU 复位解除时为 OFF。 CPU No.1 在复位中时, 为 ON(也包括当一个 CPU 从其基板上拆下的情况)。 其他 CPU 发生“多 CPU DOWN(错误代码:7000)”错误。 			
M9243	CPU No.4 复位标志	OFF : CPU No.4 复位释放 ON : CPU No.4 复位	<ul style="list-style-type: none"> 4 号 CPU 复位解除时为 OFF。 CPU No.1 在复位中时, 为 ON(也包括当一个 CPU 从其基板上拆下的情况)。 其他 CPU 发生“多 CPU DOWN(错误代码:7000)”错误。 			

注-1 : Q 系列运动控制器新增加。

注-2 : 为解除停止故障, 停止消除错误原因后 CPU No.1 复位 → 复位中被解除。

1 概述

特殊继电器列表 (续)

编号	名称	意义	详细内容	设置方 (设置时)	备注
M9244	CPU No.1 故障标志	OFF : CPU No.1 正常 ON : CPU No.1 停止故障	•在 CPU No.1 正常时变 OFF(包括继续运行故障) •CPU No.1 有停止故障时变 ON (注-2)	S(状态变化)	新 ^[注-1]
M9245	CPU No.2 故障标志	OFF : CPU No.2 正常 ON : CPU No.1 停止故障	•在 CPU No.2 正常时变 OFF (包括继续运行故障) •CPU No.1 有停止故障时变 ON (注-2)		
M9246	CPU No.3 故障标志	OFF : CPU No.3 正常 ON : CPU No.1 停止故障	•在 CPU No.3 正常时变 OFF (包括继续运行故障) •CPU No.1 有停止故障时变 ON (注-2)		
M9247	CPU No.4 故障标志	OFF : CPU No.4 正常 ON : CPU No.1 停止故障	•在 CPU No.4 正常时变 OFF (包括继续运行故障) •CPU No.1 有停止故障时变 ON (注-2)		

注-1) : Q 系列运动控制器新增加。

注-2) : 为解除停止故障, 停止消除错误原因后 CPU No.1 复位 → 复位中被解除。

③ 特殊寄存器

特殊寄存器是运动 CPU 内部规格固定的内部寄存器。因此，他不能以与普通寄存器一样的方式应用于运动 SFC 程序。

不过，如有需要，它可以写入数据以控制运动 CPU。

如无特殊指定，特殊寄存器内的数据存储为 BIN 值。

下文各表中的标题意义如下所示。

项目	说明
编号	• 表示特殊继电器的软元件号。
名称	• 表示特殊继电器的名称。
内容	• 表示特殊继电器的内容。
详细内容	• 表示特殊继电器功能的详细信息。
设置方 (设置时)	<ul style="list-style-type: none"> • 表示继电器是否由系统或用户设置，以及由系统设置时执行设置。 <设置方法> S : 由系统设置(运动 CPU) U : 由用户设置 (利用运动 SFC 程序或外部设备的测试操作) S/U : 由系统(运动 CPU)和用户设置 <设置后> 仅限于系统(运动 CPU)设置 . 主进程 : 每个主处理过程中反复设置(CPU 的空闲时间处理) 初始进程 : 仅限初始时设置(当电源开启,或执行重启时) 状态变化 : 只在状态发生变化时设置 故障 : 当发生故障时设置 请求 : 只在用户请求(特殊继电器等)时设置 运算周期 : 运动 CPU 的运算周期中设置

特殊寄存器列表

编号	名称	意义	详细内容	设置方(设置时)	备注									
D9000	熔丝烧熔编号	熔丝烧熔模块编号	•检测到最小熔丝烧熔模块的 I/O 模块编号存储于 D9000.	S(发生故障)	新 ^(註)									
D9005	AC/DC DOWN 计数器编号	AC/DC DOWN 次数	• CPU 模块运行时, 输入电压每变为少于额定电压的 85[%] (AC 电源/65[%] DC 电源) 时, 存储值便会加 1, 并且以 BIN 编码的形式存储。											
D9008	故障诊断	故障诊断编号	•自诊断发现故障时, 故障编号以 BIN 编码形式存储起来 关于“故障编码详细情况”参见“18.4 多 CPU 故障编码”。											
D9010	诊断故障发生时间	诊断故障发生的时间	• D9008 中数据更新的年份(公历, 右面两位)和月份用 BCD 编码存储为两位数。 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 至 B8</div> <div style="border: 1px solid black; padding: 2px;">B7 至 B0</div> </div> 例如:1995 年 10 月 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">年 (0 至 99)</div> <div style="border: 1px solid black; padding: 2px;">月 (1 至 12)</div> </div> H9510											
D9011			• D9008 中数据更新的日期, 时间用 BCD 编码存储为两位数。 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 至 B8</div> <div style="border: 1px solid black; padding: 2px;">B7 至 B0</div> </div> 例如: 25 日, 早上 10 点 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">日 (1 至 31)</div> <div style="border: 1px solid black; padding: 2px;">时 (0 至 23)</div> </div> H2510											
D9012			• D9008 中数据更新的分钟和秒存储为 2 位 BCD 码。 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 至 B8</div> <div style="border: 1px solid black; padding: 2px;">B7 至 B0</div> </div> 例如: 35 分 48 秒。 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">分 (0 至 59)</div> <div style="border: 1px solid black; padding: 2px;">秒 (0 至 59)</div> </div> H3548											
D9013	故障信息分类	故障信息分类编码	•判断存储在故障信息 (D9014)中的错误信息是什么。 •存储以下编码。 0: 无 1: 模块号/CPU 号/基板号 2: 参数号											
D9014	故障信息	故障信息	•存储故障诊断(D9008)对应的故障信息。 存储以下两种信息 1) 模块号/CPU 号/基板号 •根据在多 CPU 系统中发生的故障, 存储模块编号、CPU 编号 (参考存储的每一个故障编号) CPU No.1 : 1, CPU No.2 : 2, CPU No.3 : 3, CPU No.4 : 4 2) 参数号											
D9015	CPU 运行状态	CPU 运行状态	•如下图所示的 CPU 运行状态存储于 D9015。 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15</div> <div style="border: 1px solid black; padding: 2px;">B12 B11</div> <div style="border: 1px solid black; padding: 2px;">B8 B7</div> <div style="border: 1px solid black; padding: 2px;">B4 B3</div> <div style="border: 1px solid black; padding: 2px;">B0</div> </div> <div style="display: flex; justify-content: center; margin-top: 5px;"> <div style="margin-right: 20px;">← 2)</div> <div style="margin-right: 20px;">← 1)</div> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">1) CPU 运行状态</td> <td>0: RUN</td> </tr> <tr> <td></td> <td>2: STOP</td> </tr> <tr> <td>2) STOP 原因</td> <td>0: RUN/STOP 开关</td> </tr> <tr> <td>注: 优先级最高</td> <td>4: 故障</td> </tr> </table> </div>			1) CPU 运行状态	0: RUN		2: STOP	2) STOP 原因	0: RUN/STOP 开关	注: 优先级最高	4: 故障	S(主处理)
1) CPU 运行状态	0: RUN													
	2: STOP													
2) STOP 原因	0: RUN/STOP 开关													
注: 优先级最高	4: 故障													
D9017	扫描时间	扫描时间 (1 毫秒为单位)	•当前主周期以 1 毫秒为单位存储。 •设置范围 (0 至 65535[毫秒])	新 ^(註)										
D9019	最大扫描时间	最大扫描时间 (1 毫秒为单位)	•主周期的最大值以 1 毫秒为单位存储 •设置范围 (0 至 65535[毫秒])											
D9025	时钟数据	时钟数据 (年, 月)	•以 BCD 形式存储年份(公历, 右面 2 位) 和 月份 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">B15 to B12</div> <div style="border: 1px solid black; padding: 2px;">B11 to B8</div> <div style="border: 1px solid black; padding: 2px;">B7 to B4</div> <div style="border: 1px solid black; padding: 2px;">B3 to B0</div> </div> 例如: 1993 年 7 月 <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">年</div> <div style="border: 1px solid black; padding: 2px;">月</div> </div> H9307	S/U(请求)										

註: -1) : Q 系列运动控制器新增加。

1 概述

特殊寄存器列表 (续)

编号	名称	意义	详细内容	设置方 (设置时)	备注														
D9026	时钟数据	时钟数据 (日, 时)	<ul style="list-style-type: none"> 以 BCD 形式存储日期  <p>例如: 31日 上午 10点 H3110</p>	S/U(请求)															
D9027	时钟数据	时钟数据 (分, 秒)	<ul style="list-style-type: none"> 以 BCD 形式存储秒.  <p>例如: 35分, 48秒 H3548</p>																
D9028	时钟数据	时钟数据 (星期)	<ul style="list-style-type: none"> 以 BCD 形式存储星期  <p>例如: 星期五 H0005</p> <p>此处必须设为“0”</p> <table border="1" data-bbox="997 772 1125 963"> <thead> <tr> <th>星期几</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>星期日</td> </tr> <tr> <td>1</td> <td>星期一</td> </tr> <tr> <td>2</td> <td>星期二</td> </tr> <tr> <td>3</td> <td>星期三</td> </tr> <tr> <td>4</td> <td>星期四</td> </tr> <tr> <td>5</td> <td>星期五</td> </tr> <tr> <td>6</td> <td>星期六</td> </tr> </tbody> </table>			星期几		0	星期日	1	星期一	2	星期二	3	星期三	4	星期四	5	星期五
星期几																			
0	星期日																		
1	星期一																		
2	星期二																		
3	星期三																		
4	星期四																		
5	星期五																		
6	星期六																		
D9060	故障解除	所清除故障的编号	<ul style="list-style-type: none"> 存储所清除故障的编号 	U	新 ^(注)														
D9061	多 CPU 编号	CPU 编号	<ul style="list-style-type: none"> 存储自 CPU 的编号 	S(初始处理)															
D9104	伺服参数 读取请求轴编号	伺服参数 读取轴编号	<ul style="list-style-type: none"> 设定读取伺服参数的伺服放大器轴编号。 Q173CPU(N) : 1 至 32 (轴 1 至 32) Q172CPU(N) : 1 至 8 (轴 1 至 8) 	U															
D9182 D9183	测试模式请求故障	测试模式请求错误 发生时的控制中的 轴信息	<ul style="list-style-type: none"> 各轴停止中: 0/控制中: 1, 信息作为位数据存储 D9182 : b0 至 b15(轴 1 至 轴 16) D9183 : b0 至 b15(轴 17 至 轴 32) 	S(发生故障)															
D9184	运动 CPU WDT 故障原因	显示 WDT 故障的 原因	如下所示存储故障代码 1: 软件异常 1 2: 超出运算周期 3: Q 总线 WDT 故障 4: WDT 故障 30: 信息处理器硬件故障 201 至 215: Q 总线硬件故障 250 至 253: 伺服放大器接口硬件故障 300: S/W 故障 3 301: 8 点或以上的 15 CPSTART 指令同时启动 302: ROM 运行时, 内置 FLASH ROM 写入系统设定数据, 程序和参数故障																
D9185 D9186 D9187	手动脉冲发生器 轴设置故障	手动脉冲发生器轴 设置故障信息	<ul style="list-style-type: none"> 手动脉冲发生器轴设置故障标记(M9077)开启时, 存储手动脉冲发生器轴设置内容 (正常: 0/设置故障: 1) D9185: 将手动脉冲发生器轴设置故障存储到 b0 至 b2 (P1 至 P3)。将平滑倍率设置故障存储于 b3 至 b5 (P1 至 P3)。 D9186: 将一个脉冲输入放大倍率设置故障存储于 b0 至 b15 (轴 1 至 轴 16)。 D9187: 将一个脉冲输入放大倍率设置故障存储于 b0 至 b15 (轴 17 至 轴 32)。 																

(注): Q 系列运动控制器新增加。

特殊寄存器列表 (续)

编号	名称	意义	详细内容	设置方 (设置时)	备注								
D9188	运动运算周期	运动运算周期	• 运动运算周期以 [μs]为单位存储的时间.	S(操作周期)	新 ^(註)								
D9189	错误程序编号	伺服程序的错误程序编号	伺服程序错误设置标志(M9079)开启时, 错误的伺服程序编号将被存储	S(发生错误)									
D9190	错误项目信息	伺服程序错误代码	伺服程序错误设置标志(M9079)开启时, 错误项目相对应的错误代码将被存储										
D9191 D9192	伺服系统放大器 装载信息	伺服系统放大器装载 信息	• 在初始进程中检查伺服放大器的装载状态(装载: 1/非装载; 0), 作为位数据 存储. D9191: b0 至 b15(轴 1 至 轴 16) D9192: b0 至 b15(轴 17 至 轴 32) • 当电源开启后,轴从未装载状态转到装载状态,作为装载后的轴处理(不过,从 装载到未装载状态的轴保持装载状态)	S(初始处理)									
D9193 D9194 D9195	实/虚模式切换 错误	实/虚模式切换错误 代码	• 从实模式到虚模式,或从虚模式到实模式切换过程中发生模式不能切换错误, 或者在虚模式下发生一个虚拟不能维持模式错误时,存储错误信息.	S(发生错误)									
D9196	PC 链接通讯错误 代码	PC 链接通讯错误代 码	• 存储下列错误代码 00: 无错误 01: 接受时序错误 02: CRC 错误 03: 通讯响应代码错误 04: 接收帧错误 05: 通讯任务启动错误 (正常通讯重启时, 各错误代码复位为"00")										
D9197	设置的运动周期	运动 CPU 设置的操 作周期	• 设置操作周期以[μs]为单位存储的时间.	S(初始处理)									
D9200	开关状态	CPU 开关状态	• CPU 以下列格式存储开关状态.  <table border="1" data-bbox="603 1254 1157 1512"> <tr> <td>1) CPU 开关状态</td> <td>0: RUN 1: STOP 2: L.CLR</td> </tr> <tr> <td>2) 存储卡开关</td> <td>一直 OFF</td> </tr> <tr> <td>3) DIP 开关</td> <td>B8 到 B12 与系统设置开关 1 的 SW1 到 SW5 相对应 0: OFF/1: ON B13 到 B15 未被用到.</td> </tr> </table>	1) CPU 开关状态	0: RUN 1: STOP 2: L.CLR	2) 存储卡开关	一直 OFF	3) DIP 开关	B8 到 B12 与系统设置开关 1 的 SW1 到 SW5 相对应 0: OFF/1: ON B13 到 B15 未被用到.	S(主处理)	新 ^(註)		
1) CPU 开关状态	0: RUN 1: STOP 2: L.CLR												
2) 存储卡开关	一直 OFF												
3) DIP 开关	B8 到 B12 与系统设置开关 1 的 SW1 到 SW5 相对应 0: OFF/1: ON B13 到 B15 未被用到.												
D9201	LED 状态	CPU-LED 状态	• CPU 的 LED 处于以下的哪个状态, 以以下的位模式存储。 • 0: 灭 1: 亮 2: 闪烁  <table border="1" data-bbox="622 1724 981 1915"> <tr> <td>1): RUN</td> <td>5): BOOT</td> </tr> <tr> <td>2): ERROR</td> <td>6) 未用</td> </tr> <tr> <td>3): M.RUN</td> <td>7): 未用</td> </tr> <tr> <td>4): BAT.ALARM</td> <td>8): MODE</td> </tr> </table> MODE 的位格式 0: 灭 1: 绿色 2: 橙色	1): RUN	5): BOOT	2): ERROR	6) 未用	3): M.RUN	7): 未用	4): BAT.ALARM	8): MODE	S(状态转换)	新 ^(註)
1): RUN	5): BOOT												
2): ERROR	6) 未用												
3): M.RUN	7): 未用												
4): BAT.ALARM	8): MODE												

1 概述

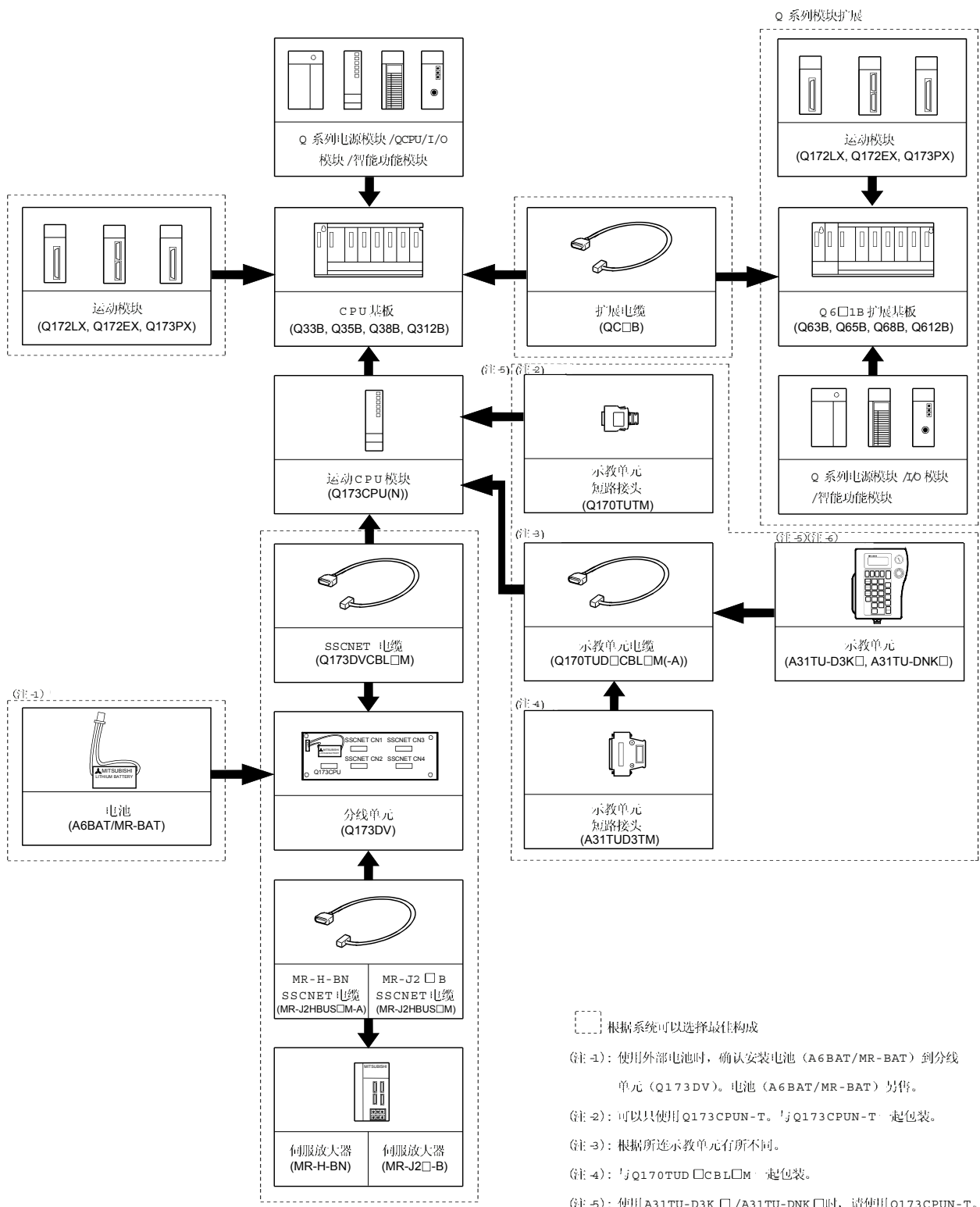
1.3 硬件构成

本节说明了Q173CPU(N) /Q172CPU(N) 的系统构成, 使用该系统的注意事项及构成设备。

1.3.1 运动系统构成

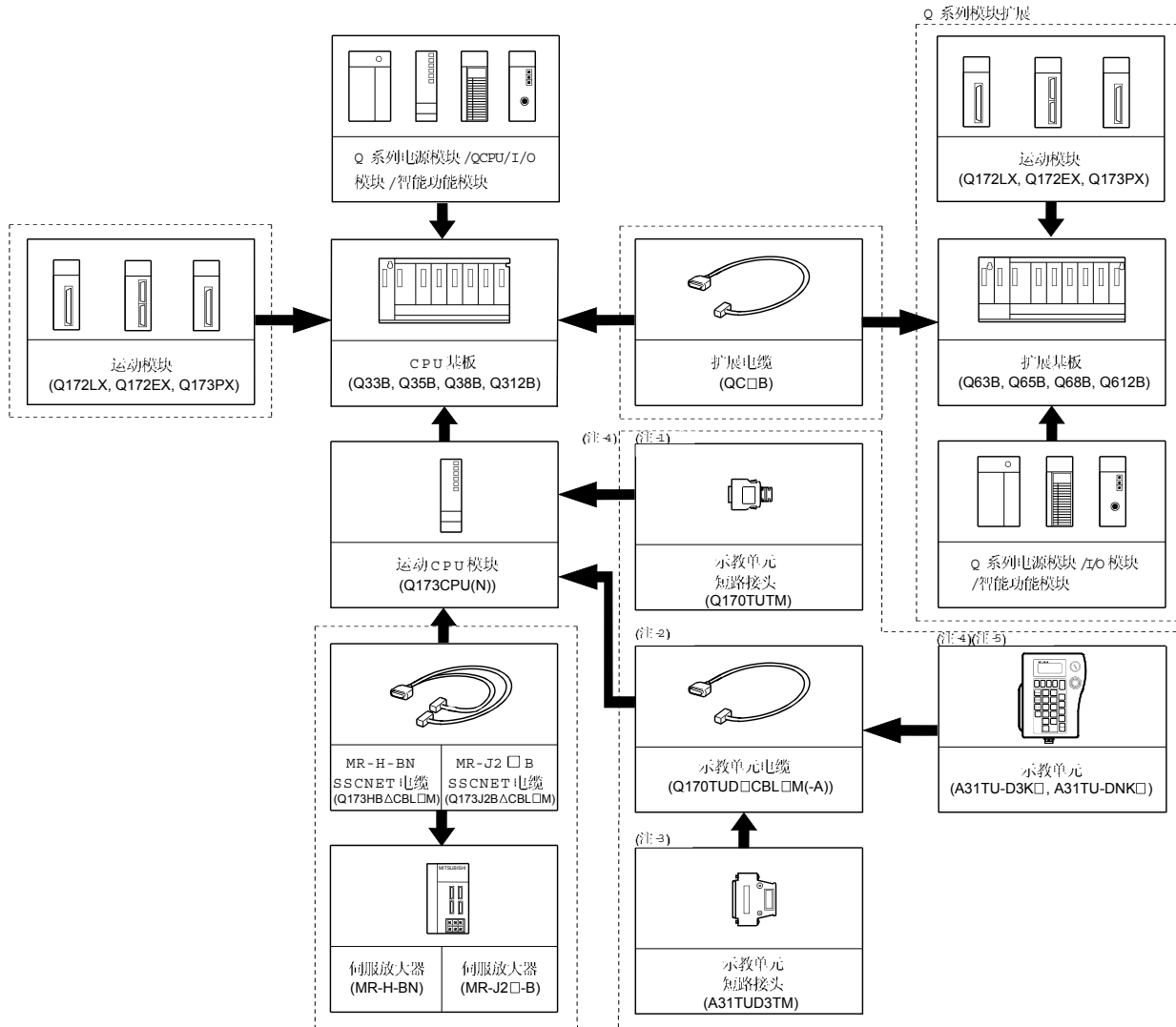
Q173CPU(N) /Q172CPU(N) 系统中的设备构成, 与外围设备构成及系统构成概要参见下页。

(1) Q173CPU(N) 系统的软元件构成
 (a) 使用分线单元/外部电池时



- 根据系统可以选择最佳构成
- (注-1): 使用外部电池时, 确认安装电池 (A6BAT/MR-BAT) 到分线单元 (Q173DV)。电池 (A6BAT/MR-BAT) 另售。
- (注-2): 可以只使用 Q173CPUN-T。与 Q173CPUN-T 一起包装。
- (注-3): 根据所述示教单元有所不同。
- (注-4): 与 Q170TUD-CBL-M 一起包装。
- (注-5): 使用 A31TU-D3K / A31TU-DNK 时, 请使用 Q173CPUN-T。示教单元只用于 SV13。
- (注-6): A31TU-D3K / A31TU-DNK 仅有旧版本。

b) 使用分接电缆时



□ 根据系统可以选择最佳构成

(注-1): 可以只使用 Q173CPUN-T。与 Q173CPUN-T 一起包装。

(注-2): 根据所连示教单元有所不同。

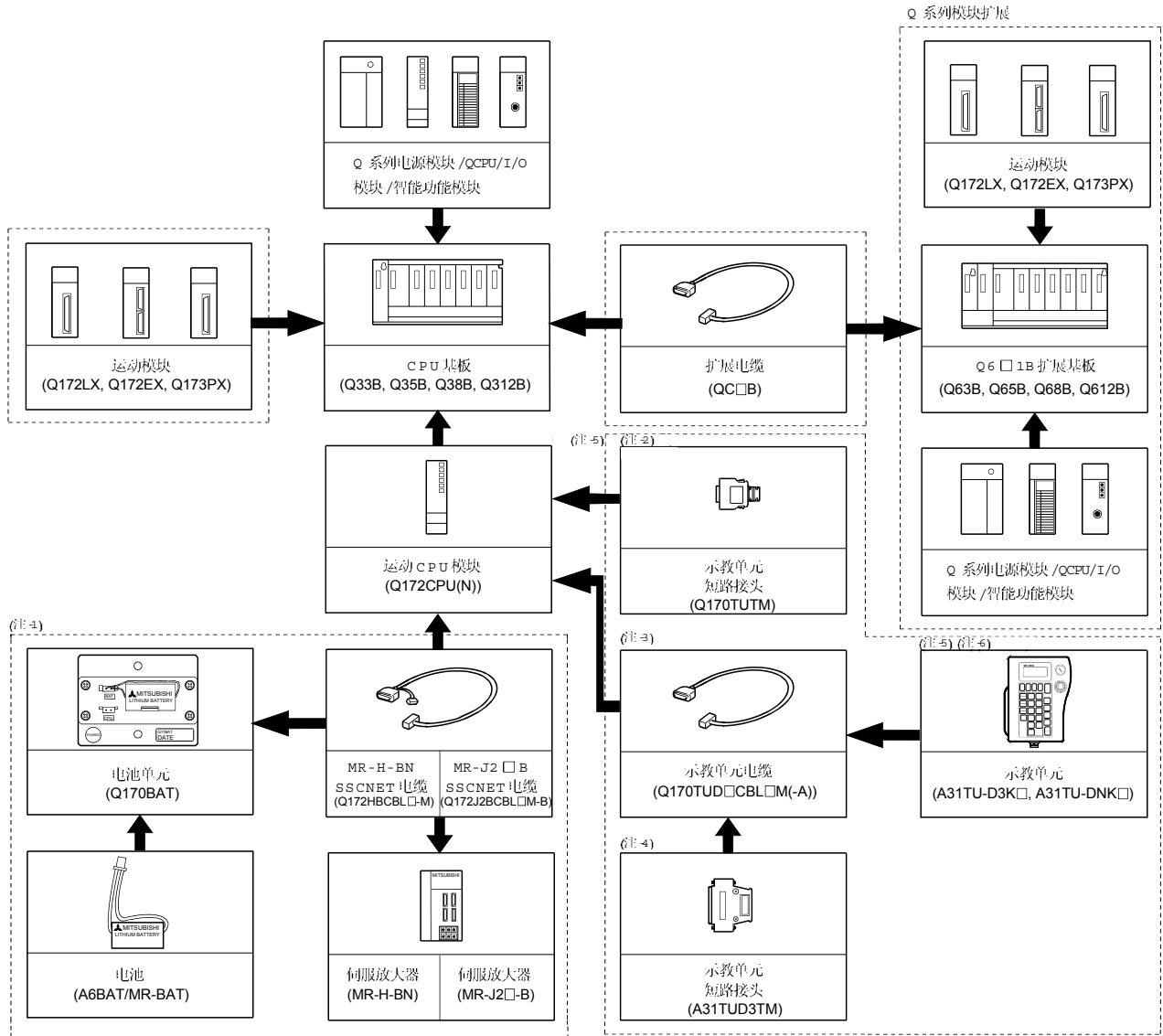
(注-3): 与 Q170TUD□CBL□M 一起包装。

(注-4): 使用 A31TU-D3K□ / A31TU-DNK□ 时, 请使用 Q173CPUN-T。

示教单元只用于 SV13。

(注-5): A31 TU-D3K□ / A31 TU-DNK□ 仅有日文版本。

Q Q172CPU(N) 系统的软元件构成
 a) 使用外部电池时



根据系统可以选择最佳构成

(注-1): 使用外部电池时, 确认使用 SSCNET 电缆(Q172J2BCBL□M-B/2)并且设置
 电池 (A6BAT/MR-BAT)。同时安装电池(A6BAT/MR-BAT)到电池模块
 (Q170BAT)。电池 (A6BAT/MR-BAT) 另售。

(注-2): 可以只使用 Q172CPUN-T。与 Q172CPUN-T 一起包装。

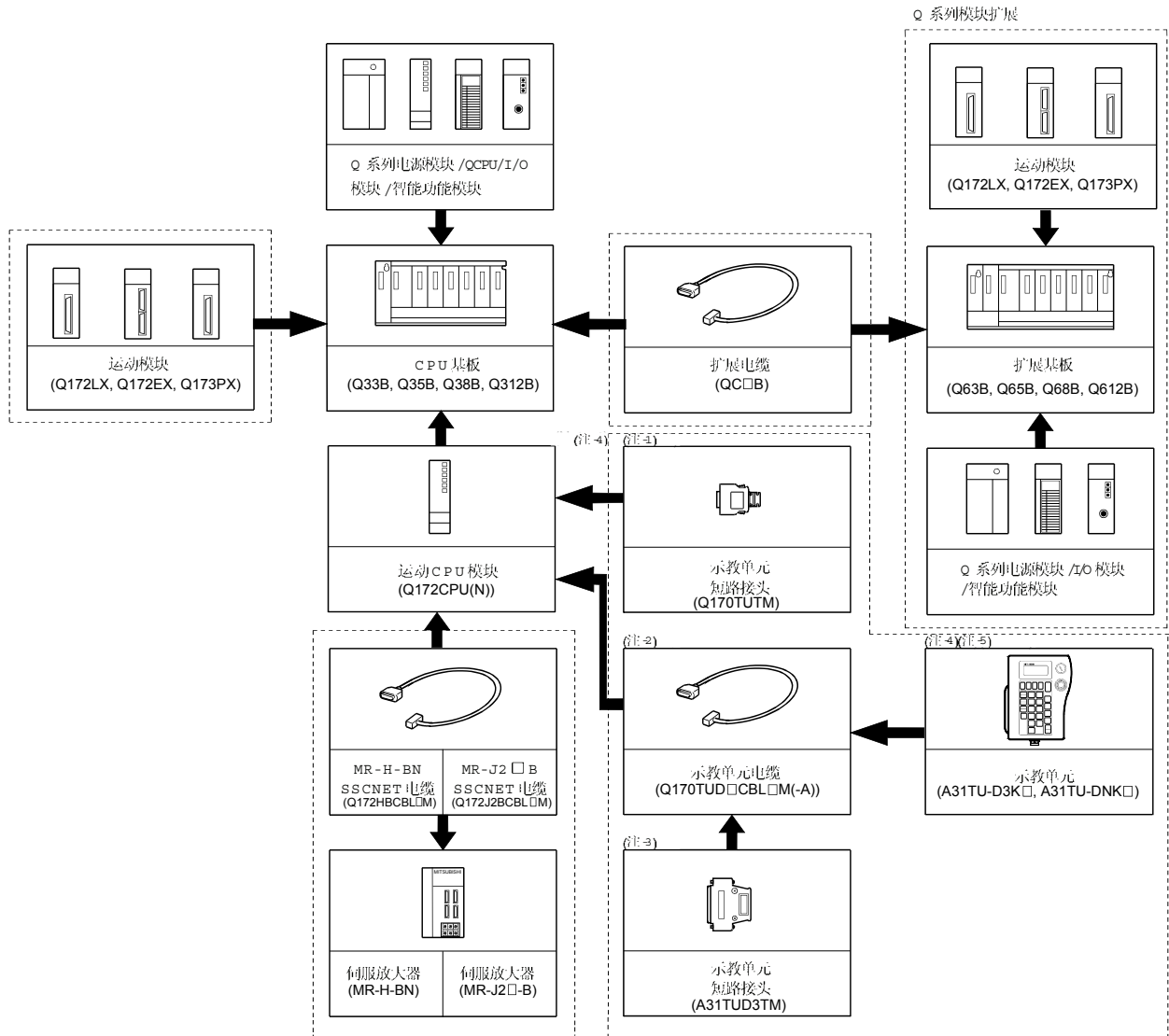
(注-3): 根据所连示教单元有所不同。

(注-4): 与 Q170TUD□CBL□M 一起包装。

(注-5): 使用 A31TU-D3K□ / A31TU-DNK□时, 请使用 Q172CPUN-T。
 示教单元只用于 SV13。

(注-6): A31 TU-D3K□ / A31 TU-DNK□仅有日文版本。

b) 不使用外部电池时



根据系统可以选择最佳构成

注-1: 可以只使用Q172CPUN-T。与Q172CPUN-T一起包装。

注-2: 根据所连示教单元有所不同。

注-3: 与Q170TUD□CBL□M一起包装。

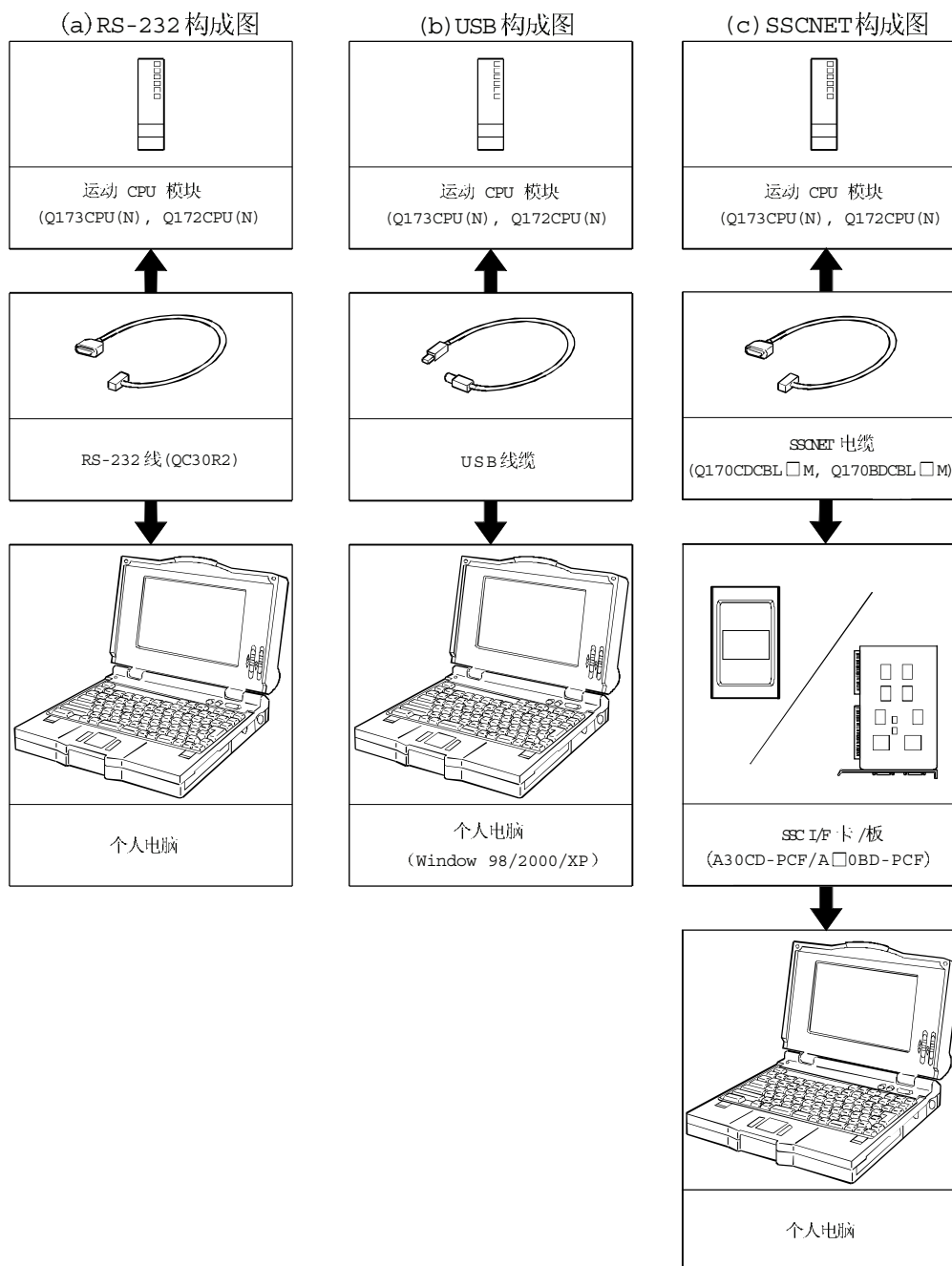
注-4: 使用A31TU-D3K□/A31TU-DNK□时, 请使用Q172CPUN-T。

示教单元用于SV13。

注-5: A31TU-D3K□/A31TU-DNK□仅有日文版本。

(3) Q173CPU(N) / Q172CPU(N) 的外围设备构成

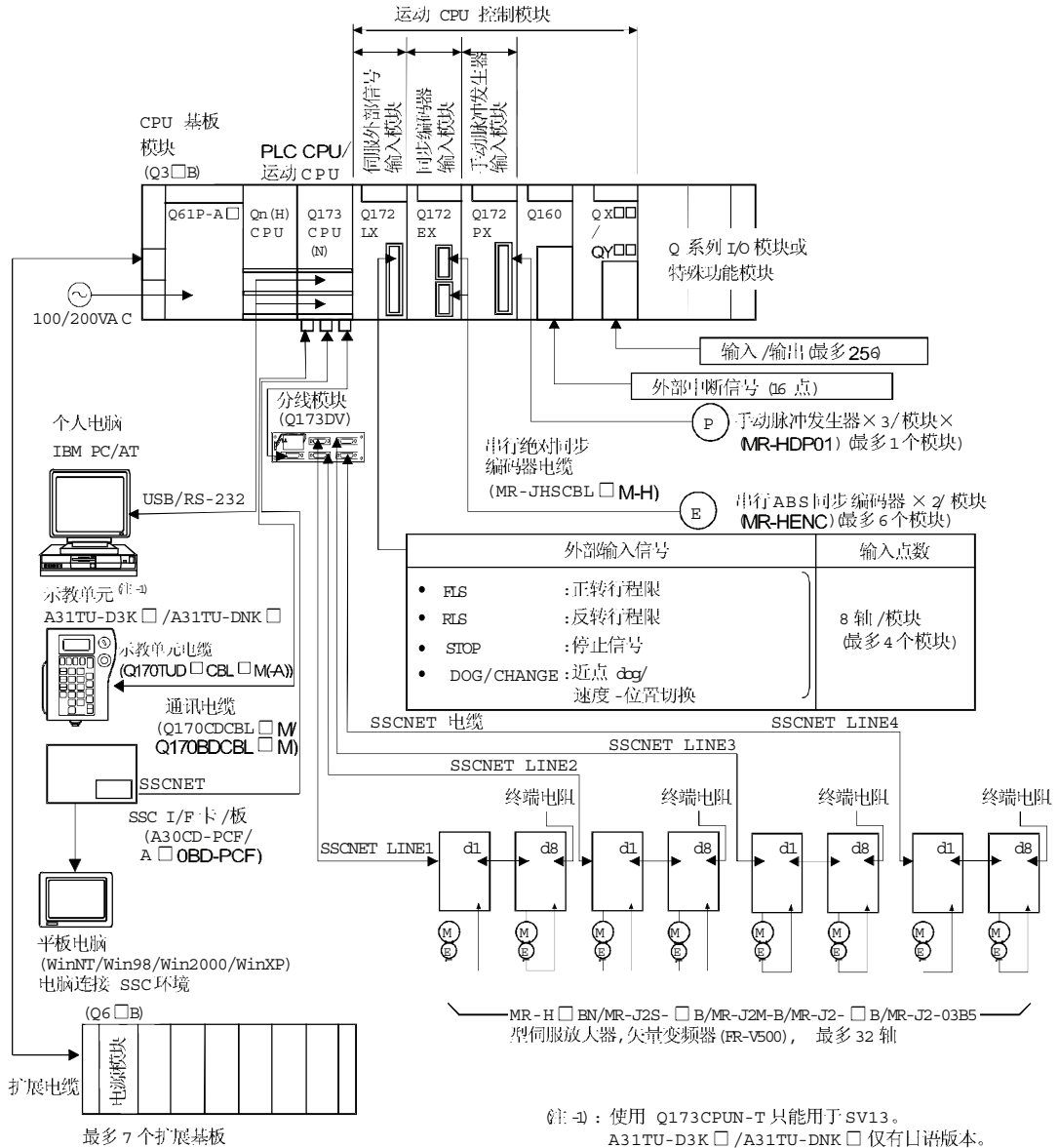
如以下 (a) (b) (c) 所示。



(注): QCPU 的 GPP 功能, 请参照 PLC 操作手册。各程序软件包的操作方法, 请参照相应软件的帮助。

1 概述

1.3.2 Q173CPU(N) 系统总构成

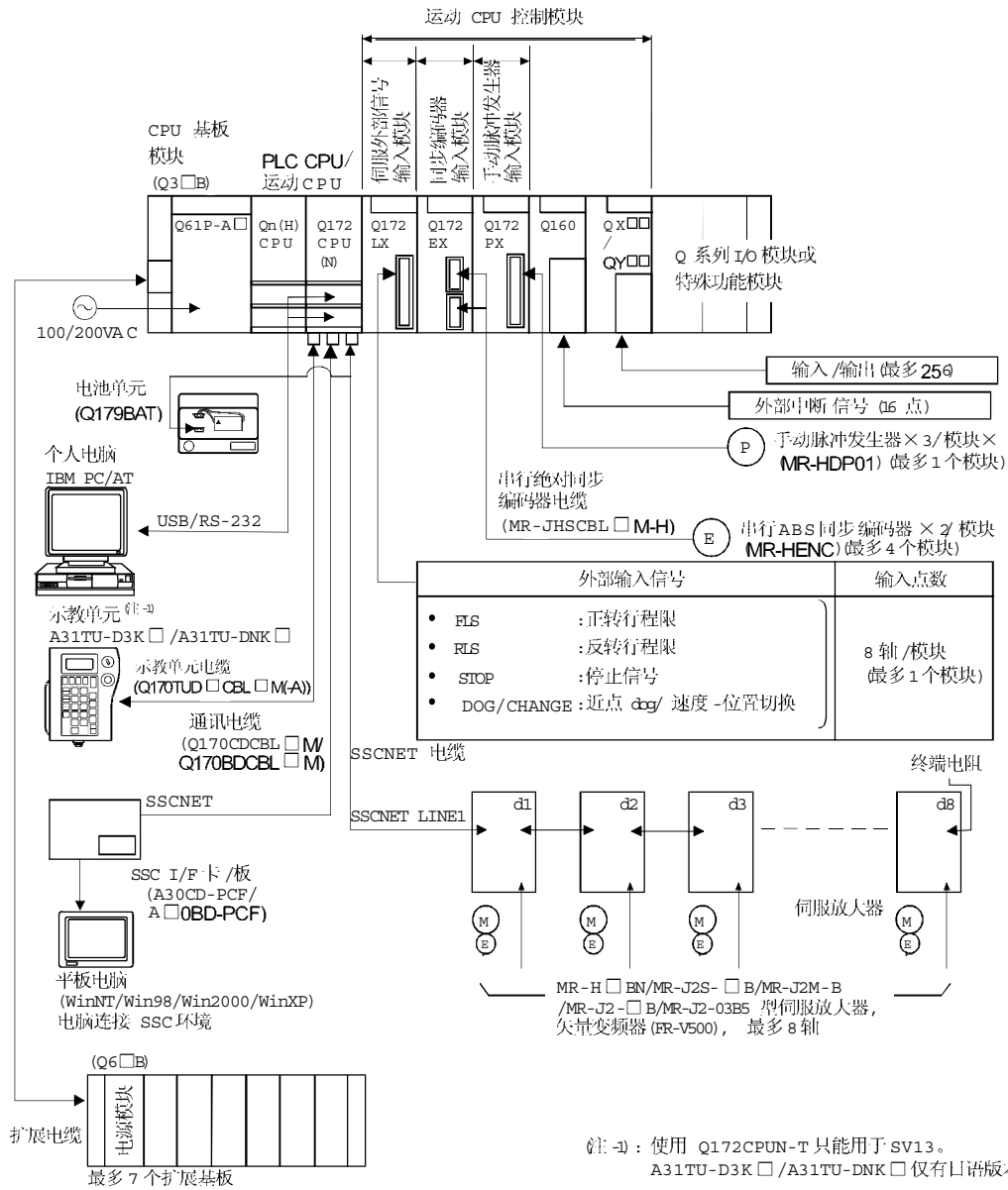


注意

- 如运动控制器或伺服放大器的异常动作与安全方向动作不同，在运动控制器或伺服放大器外部构造安全回路。
- 系统所用部件（运动控制器，伺服放大器及伺服电机之外）的额定值及特性必须与运动控制器，伺服放大器及伺服电机相兼容。
- 设置参数值使之与运动控制器，伺服放大器，伺服电机及再生电阻型号和系统应用相兼容。如设置不正确，保护功能可能失效。
- 当使用示教单元时，必须在运动 CPU (Q173CPUN-T/Q172CPUN-T)与示教单元之间安装示教单元电缆。并且，卸下示教单元后或不使用示教单元时，连接示教单元的短路接头。

1 概述

1.3.3 Q172CPU(N) 系统总构成



注意

- 如运动控制器或伺服放大器的异常动作与安全方向动作不同，在运动控制器或伺服放大器外部构造安全回路。
- 系统所用部件（运动控制器，伺服放大器及伺服电机之外）的额定值及特性必须与运动控制器，伺服放大器及伺服电机相兼容。
- 设置参数值使之与运动控制器，伺服放大器，伺服电机及再生电阻型号和系统应用相兼容。如设置不正确，保护功能可能失效。
- 当使用示教单元时，必须在运动 CPU (Q173CPUN-T/Q172CPUN-T)与示教单元之间安装示教单元电缆。并且，卸下示教单元后或不使用示教单元时，连接示教单元的短路接头。

1 概述

1.3.4 软件包

(1) 操作系统软件包

应用	软件包		备注
	Q173CPU(N)	Q172CPU(N)	
传送装配用 SV13 (运动 SFC)	SW6RN-SV13QB	SW6RN-SV13QD	
自动机用 SV22 (运动 SFC)	SW6RN-SV22QA	SW6RN-SV22QC	

(2) 集成启动支持软件包

型号	内容		备注
SW6RNC-GSVPROE	SW6RNC-GSVE (集成启动支持软件 (1 CD-ROM))	传送装配用软件 : SW6RN-GSV13P	
		自动机器用软件 : SW6RN-GSV22P	
		凸轮数据创建软件 : SW3RN-CAMP	
		数字示波器软件 : SW6RN-DOSCP	
		通讯系统软件 : SW6RN-SNETP	
		文档打印软件 : SW3RN-DOCPRNP	
		SW20RN-DOCPRNP	
		SW6RNC-GSVHELPE(操作手册(1 CD-ROM))	
		安装手册	
SW6RNC-GSVSETE		SW6RNC-GSVPROE	
		A30CD-PCF(SSC I/F 卡(PCMCIA TYPE II 1CH/卡))	
		Q170CDCBL3M(A30CD-PCF 电缆 3m)	

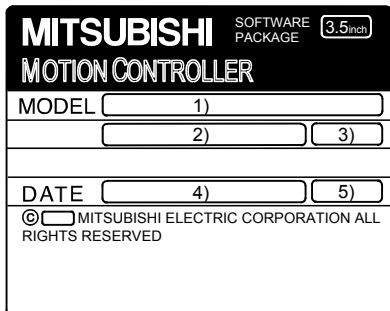
(注) : 编程软件的操作环境仅限 WindowsNT® 4.0/
Windows® 98/Windows® 2000/Windows® XP。

(3) PLC 软件包

型号	软件包	备注
GX Developer	SW□□5C-GPPW-E	

(注) : □ = 使用 ☞ 或更新。

(4) 操作系统 (OS) 类型 / 版本
 (a) 操作系统 (OS) 中的确认方法



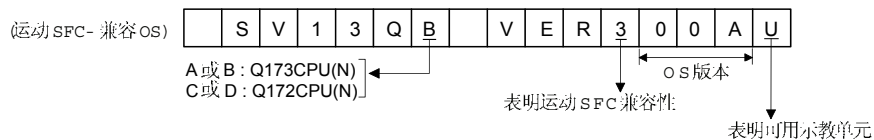
- 1) OS 软件型号
- 2) 软件版本
- 3) OS 软件版本
- 4) 序列号
- 5) FD 号

示例) 使用 Q173CPU(N), SV13 及版本 A 时。

- 1) SW6RN-SV13QB
- 2) BCD-B14W276
- 3) A

(b) 程序软件中的确认方法

连接的 CPU 的操作系统 (OS) 类型 / 版本在安装编程软件的计算机屏幕上显示。



1 概述

⑤ 由运动 CPU 及软件版本导致的功能和 PLC CPU 的限制

根据运动 CPU 模块, 操作系统及编程软件版本, 可用功能及 PLC CPU 有所限制。各版本及功能组合如下所示。

功能	操作系统 软件版本 ^(注-1)	编程 软件版本	CPU 模块版本 ^(注-2)				章节号
			Q173 CPU	Q173 CPUN	Q172 CPU	Q172 CPUN	
ROM 操作	H	C	M	—	N	—	第 14 章
ROM 操作 (用于附加参数 (原点回归参数等))	N	C	T	M	U	M	—
在线修改	J	F	—	—	—	—	12.3 节
改善 CPU 共享内存的自动刷新功能	H	C	M	—	N	—	3.1 (3) 节
网络通讯	H	C	M	—	N	—	第 15 章
主操作周期监视	D	—	—	—	—	—	第 16 章
从伺服放大器读出伺服参数	D	—	—	—	—	—	第 17 章
运动 SFC 指令	MULTR	D	—	—	—	—	7.13.6 节
	MULTW	D	—	J	—	K	7.13.5 节
	OUT	D	—	—	—	—	7.9.5 节
	FROM	H	C	—	—	—	7.13.8 节
	TO	H	C	—	—	—	7.13.7 节
运动专用指令 (SVST 指令等)	H	—	M	—	N	—	5.3 到 5.6 节
矢量变频器的连接可否	K	F	—	—	—	—	—
基本型 QCPU (功能版本“B”) (Q00CPU, Q01CPU)	M	—	—	—	—	—	—
添加原点回归功能	L	F	—	—	—	—	—

—: 无版本限制。

注 -1) : SV13/SV22 为完全相同版本。

注 -2) : Q172CPUN-T/Q173CPUN-T 与版本 A 对应。

1 概述

⑥ 个人电脑的操作环境

操作环境如下：

在安装 WindowsNT/98/2000/XP 的 IBM PC/AT 上可正常运行。

项目	WindowsNT [®] 4.0(服务包 2 或更高) (注)或 Windows [®] 98	Windows [®] 2000	Windows [®] XP
CPU	奔腾 133MHz 以上	奔腾 II 233MHz 以上	奔腾 II 450MHz 以上
内存	推荐 32MB 以上	推荐 64MB 以上	推荐 192MB 以上
硬盘可用空间	硬盘可用空间如下表所示		
驱动器	3.5 英寸(1.44MB) 软驱, CD-ROM 驱动器		
显示	分辨率 800×600, 256 色以上		

(注)：不可使用 USB 连接

下列容量须取决于所安装的软件

● SW6RNC-GSVE

型号	容量	
SW6RN-GSV13P	40MB	
SW6RN-GSV22P	40MB	
SW3RN-CAMP	2MB	
SW6RN-DOSCP	12MB	
SW6RN-SNETP	标准	11.5MB
	定制 (全选时)	12MB
SW3RN-DOCPRNP	34MB	
SW20RN-DOCPRNP	33MB	

● SW6RNC-GSVHELPE

型号	容量
SW6RN-GSV13P	23MB
SW6RN-GSV22P	25MB
SW3RN-CAMP	3MB
SW6RN-DOSCP	5MB
SW6RN-SNETP	3MB
SW3RN-DOCPRNP	4MB
SW20RN-DOCPRNP	4MB

(注-1)：WindowsNT[®]，Windows[®] 是微软公司在美国和 / 或其他国家所持有的商标或注册商标。

(注-2)：Pentium[®] 为英特尔公司或其子公司在美国和 / 或其他国家所持有的商标或注册商标。

要点
(1) 当本软件操作中明确 Windows 操作时，参见 Windows 的使用手册或其他供应商提供的指南书。
(2) 因 WindowsNT® 4.0/Windows® 98/Windows® 2000/Windows® XP 的系统字号不同，画面可能无法正确显示。确定使用小号字。

1.3.5 运动系统的限制

- ㉑ 运动 CPU 不可用作安装在 QA1S6□B 扩展基板上的模块的控制 CPU。必须使用 PLC CPU 作为控制 CPU。
- ㉒ 运动 CPU 模块上安装内存卡的接头是用做将来功能扩展。
- ㉓ 运动 CPU 模块不可单独使用。始终和 PLC CPU 模块 (支持多 CPU 系统的版本) 组合使用。另外, 必须在 PLC CPU 模块右侧安装。PLC CPU 模块不可安装在运动 CPU 模块的右边位置。
- ㉔ PC CPU 模块必须安装在运动 CPU 模块的右侧。运动 CPU 模块不可安装在 PC CPU 模块的右边位置。
- ㉕ 在“Q 模式”下使用 PLC CPU。
- ㉖ 运动 CPU 模块不可设置为智能功能模块或图形操作终端 (GOT) 的控制 CPU。
- ㉗ 连接运动 CPU 与伺服放大器的 SSCNET 电缆及连接运动 CPU 与 A31TU-D3K□/A31TU-DNK□ 的示教单元连接电缆从模块的基板引出。设计控制面板时, 确定留出足够的空间以便引出电缆。
- ㉘ 运动 CPU 是 Q 系列多 CPU 系统的一个模块元件。有必要为各 PLC CPU 设置 Q 系列多 CPU 系统的参数。同时必须通过系统设置将运动 CPU 模块设置为支持多系统。

- ⑨ 确认使用运动CPU作为运动CPU 专用运动模块 (Q172LX, Q172EX, Q173PX) 的控制CPU。如误将PLC CPU 设置并安装为控制CPU，则不能正常运行。其他PLC CPU将运动CPU 视为32点智能模块。
从其他机器不可访问运动CPU。
- ⑩ 当构造多CPU系统时，确认模块构成以使CPU基板上各个模块的DV5V电流消耗总和不超过电源模块的输出容量。
- ⑪ 运动模块 (Q172LX, Q172EX, Q173PX) 选择是否必要, 参见系统设计的“Q173CPU(N)/Q172CPU(N) 运动控制器用户手册”中“第3章 设计”。

1.4 多 CPU 系统

1.4.1 概述

(1) 多 CPU 系统

多个(最多4模块) PLC CPU 及运动CPU安装在 CPU 基板上,且各CPU控制多CPU系统中插槽控制CPU基板/扩展基板的 I/O 模块及智能功能模块。各运动CPU控制SSCNET连接的伺服放大器。

(2) 分散系统构成

- a) 在多处理器中分散伺服控制, 机器控制及信息控制等任务可实现弹性系统构成。
- b) 可使用多个运动CPU增加控制轴数。使用3个Q173CPU(N)S可控制最多96个轴。
- c) 可使用多PLC CPU减少总系统的PLC扫描时间并在其中分散PLC控制负载。

(3) 多CPU系统中CPU间通讯

- a) 多CPU系统中CPU间数据传送通过多CPU自动刷新功能自动进行。这可以把其他CPU的软元件数据作为自CPU的软元件数据使用。
- b) 可通过运动专用PLC指令从PLC CPU访问运动CPU软元件数据并启动运动SFC程序。

1.4.2 PLC CPU 及运动 CPU 的安装

CPU 基板上可安装最多共 4 个 PLC CPU 及运动 CPU，按顺序安装到基板 CPU 插槽（紧挨电源模块右侧的插槽）的相邻插槽中。

在 PLC CPU 与运动 CPU 之间，或运动 CPU 之间不可空余未安装插槽。

当安装 2 或 3 个运动 CPU 时，在提供的 1 或多个 PLC CPU 右侧的插槽里安装。

(PLC CPU 不可安装在运动 CPU 的右边。)

① 当使用高性能型 QCPU 时

CPU 数	PLC CPU/ 运动 CPU 安装位置																									
2	<table border="1"> <tr> <td></td> <td>CPU 0</td> <td>1</td> <td>2</td> </tr> <tr> <td>电源</td> <td>PLC CPU</td> <td>运动 CPU</td> <td>I/O, 第</td> </tr> <tr> <td></td> <td></td> <td></td> <td>I/O, 第</td> </tr> </table>		CPU 0	1	2	电源	PLC CPU	运动 CPU	I/O, 第				I/O, 第													
	CPU 0	1	2																							
电源	PLC CPU	运动 CPU	I/O, 第																							
			I/O, 第																							
3	<table border="1"> <tr> <td></td> <td>CPU 0</td> <td>1</td> <td>2</td> </tr> <tr> <td>电源</td> <td>PLC CPU</td> <td>PLC CPU</td> <td>运动 CPU</td> </tr> <tr> <td></td> <td></td> <td></td> <td>I/O, 第</td> </tr> </table>		CPU 0	1	2	电源	PLC CPU	PLC CPU	运动 CPU				I/O, 第	<table border="1"> <tr> <td></td> <td>CPU 0</td> <td>1</td> <td>2</td> </tr> <tr> <td>电源</td> <td>PLC CPU</td> <td>运动 CPU</td> <td>运动 CPU</td> </tr> <tr> <td></td> <td></td> <td></td> <td>I/O, 第</td> </tr> </table>		CPU 0	1	2	电源	PLC CPU	运动 CPU	运动 CPU				I/O, 第
	CPU 0	1	2																							
电源	PLC CPU	PLC CPU	运动 CPU																							
			I/O, 第																							
	CPU 0	1	2																							
电源	PLC CPU	运动 CPU	运动 CPU																							
			I/O, 第																							
4	<table border="1"> <tr> <td></td> <td>CPU 0</td> <td>1</td> <td>2</td> </tr> <tr> <td>电源</td> <td>PLC CPU</td> <td>PLC CPU</td> <td>PLC CPU</td> </tr> <tr> <td></td> <td></td> <td></td> <td>运动 CPU</td> </tr> </table>		CPU 0	1	2	电源	PLC CPU	PLC CPU	PLC CPU				运动 CPU	<table border="1"> <tr> <td></td> <td>CPU 0</td> <td>1</td> <td>2</td> </tr> <tr> <td>电源</td> <td>PLC CPU</td> <td>PLC CPU</td> <td>运动 CPU</td> </tr> <tr> <td></td> <td></td> <td></td> <td>运动 CPU</td> </tr> </table>		CPU 0	1	2	电源	PLC CPU	PLC CPU	运动 CPU				运动 CPU
		CPU 0	1	2																						
电源	PLC CPU	PLC CPU	PLC CPU																							
			运动 CPU																							
	CPU 0	1	2																							
电源	PLC CPU	PLC CPU	运动 CPU																							
			运动 CPU																							
	<table border="1"> <tr> <td></td> <td>CPU 0</td> <td>1</td> <td>2</td> </tr> <tr> <td>电源</td> <td>PLC CPU</td> <td>运动 CPU</td> <td>运动 CPU</td> </tr> <tr> <td></td> <td></td> <td></td> <td>运动 CPU</td> </tr> </table>		CPU 0	1	2	电源	PLC CPU	运动 CPU	运动 CPU				运动 CPU													
	CPU 0	1	2																							
电源	PLC CPU	运动 CPU	运动 CPU																							
			运动 CPU																							

② 当使用基本型 QCPU 时

多 CPU 系统最多包括 3 个模块 (PLC CPU × 1, 运动 CPU × 1, PC CPU × 1)。

1.4.3 使用 Q 系列 I/O 模块及智能功能模块的注意事项

- (1) 运动 CPU 可控制的模块
运动 CPU 可控制 I/O 模块(QX□, QY□, QH□, QX□Y□, Q6□AD□, Q6□DA□) 中断模块 (QI60) 及运动模块 (Q172LX, Q172EX, Q173PX)。
- (2) 与多 CPU 系统的兼容
- a) 所有 I/O 模块(QX□, QY□, QH□, QX□Y□, Q6□AD□, Q6□DA□) 均支持多 CPU 系统。
 - b) 中断模块 (QI60), 不受制于功能升级, 支持多 CPU 系统。
 - c) 智能功能模块只在版本号为 B 或更新版本时才支持多 CPU 系统。这些模块不被运动 CPU 控制, 所以确认使用 PLC CPU 作为控制 CPU。
 - d) 所有运动模块(Q172LX, Q172EX, Q173PX) 均支持多 CPU 系统。这些模块不被 PLC CPU 控制, 所以确认使用运动 CPU 作为控制 CPU。
- (3) 非控制 CPU 的访问范围
- a) 运动 CPU 只可访问自 CPU 控制的模块。不可访问其他 CPU 控制的模块。
 - b) 非控制 CPU 对运动 CPU 所控模块的访问范围如下所示:

访问目标		组外设置 I/O (从 PLC CPU 设置)	
		未收到	收到
输入 (X)		×	○
输出 (Y)		×	×
缓冲存储器	读	×	×
	写	×	×

备注

- 智能功能模块的功能版本可在智能功能模块的额定铭牌上或在 GX Developer 系统监视产品信息列表中查询。
- 运动 CPU 可控 I/O 模块型号参见《Q173CPU (N) / Q172CPU (N) 运动控制器用户手册》。

1 概述

1.4.4 模块安装的数量限制

① 运动 CPU 中受制于安装限制的模块如下所示。在下表所列的限制范围内使用。

说明	型号	各 CPU 最大可安装模块	
		Q173CPU(N)	Q172CPU(N)
伺服外部信号接口模块	Q172LX	4 模块	1 模块
串行 ABS 同步接口模块	Q172EX	6 模块 ^②	4 模块 ^②
手动脉冲发生器接口模块	Q173PX (Note-2)	4 模块 ^② (使用增量同步编码器时)	3 模块 ^② (使用增量同步编码器时)
		1 模块 (只使用手动脉冲发生器时)	1 模块 (只使用手动脉冲发生器时)
输入模块	QX□	共 256 点	
输出模块	QY□		
输入/输出混合模块	QH□ QX□Y□		
模拟输入模块	Q6□AD□		
模拟输出模块	Q6□DA□		
中断模块	QI60		

②：只适用 SV22。

③：当手动脉冲发生器及同步编码器与 SV22 同时使用时，安装在最小号插槽里的 Q173PX 被用作手动脉冲发生器输入。

- ② 运动 CPU 控制的模块不可安装在扩展基板 QA1S6□B 上。在 CPU 基板 Q3□B 或扩展基板 Q6□B 上安装。
- ③ 可使用包括 1 个 CPU 基板和 7 个扩展基板的共 8 个基板。然而，各系统可用插槽数(模块数)包括空槽数应在 64 之内。如模块被安装在 65 号插槽或以后的插槽里，故障 (SP. UNIT LAY ERROR) 将发生。确认所有模块安装在 1 到 64 号插槽里。(即使 CPU 基板及扩展基板提供的插槽总数超过 65 (如当使用 6 个 12- 插槽基板时)，只要模块安装在 1 到 64 号插槽内，故障就不会发生。)

1.4.5 多 CPU 系统的处理时间

(1) 多 CPU 系统的处理

多 CPU 系统的各 CPU 模块访问安装在主基板和扩展基板上的自 CPU 控制的模块，虽然其他 CPU 通过总线（基板模式及扩展电缆）访问。然而，多 CPU 模块不可同时使用总线。当多 CPU 同时访问总线时，后面执行总线访问的 CPU 停留在“等待状态”直到正在使用总线的 CPU 完成。在多 CPU 系统里，上述等待时间（CPU 等待状态时间）导致 I/O 延迟或扫描时间延长。

(2) 等待状态时间变为最大时

在多 CPU 系统里，在以下条件下，自 CPU 的等待时间变为最长：

- 多 CPU 系统中共使用 4 台 PLC CPU/ 运动 CPU 时。
- 使用扩展基板时。
- 处理大量数据的智能功能模块安装在扩展基板时。
- 使用共 4 个 CPU 且 4 个 CPU 同时访问安装在扩展基板上的模块时。
- 在 PLC CPU 与运动 CPU 间有大量自动刷新点时。

(3) 缩短多 CPU 系统的处理时间

用以下方法可缩短多 CPU 系统的处理时间：

- 将带大量访问点或较多刷新的 MELSECNET/10H 及 CC-Link 的模块安装在 CPU 基板。
- 只用一个 PLC CPU 控制所有带大量访问点或较多刷新的模块如 MELSECNET/10H 及 CC-Link 以便他们不被 2 个或 2 个以上 CPU 同时访问。
- 减少 MELSECNET/10H, CC-Link 的刷新点数等。
- 减少 PLC CPU/ 运动 CPU 自动刷新点数。

1.4.6 如何复位多 CPU 系统

对于多 CPU 系统，通过复位 1 号 CPU 的 PLC CPU 复位全系统。
 CPU 1 号 CPU 的 QCPU 复位时，所有的 CPU, I/O 模块及智能功能模块将复位。

要复位多 CPU 系统中发生 CPU 停止故障的任一 CPU 时，复位 1 号机的 PLC CPU 或重启电源(即，打开电源 ON, OFF, 然后 ON)。

(产生停止故障的 2 号~4 号的 PLC CPU/运动 CPU 不能通过复位相应的 CPU 进行复位。)



要点
<p>(1) 在多 CPU 系统中，CPU No 2, 3 或 4 的 PLC CPU/运动 CPU 不可被个别复位。当 CPU 2, 3 或 4 号机的 PLC CPU 或运动 CPU 被复位而多 CPU 系统在运行时，其他 CPU 发生 MULTI CPU DOWN 故障 (出错代码: 7000) 且总系统停止。注意根据 CPU No 2, 3 或 4 的 PLC CPU 或运动 CPU 的复位时序，其他 CPU 的 QCPU 可能因 MULTI CPU DOWN 之外的故障停止。</p> <p>(2) 复位 CPU No 2, 3 或 4 发生 MULTI CPU DOWN 故障。(与多 CPU 设定的动作模式无关。)(参见 1.4.7 节《多 CPU 设置中的操作模式设置》。)</p>

1.4.7 QCPU 或 Q173CPU(N) / Q172CPU(N) 发生 CPU DOWN 故障时的处理

在多 CPU 系统中,当 CPU No.1 发生 CPU DOWN 故障时,系统操作与 CPU No.2, 3 或 4 发生此类故障时不同。

(1) CPU No.1 发生 CPU DOWN 故障时

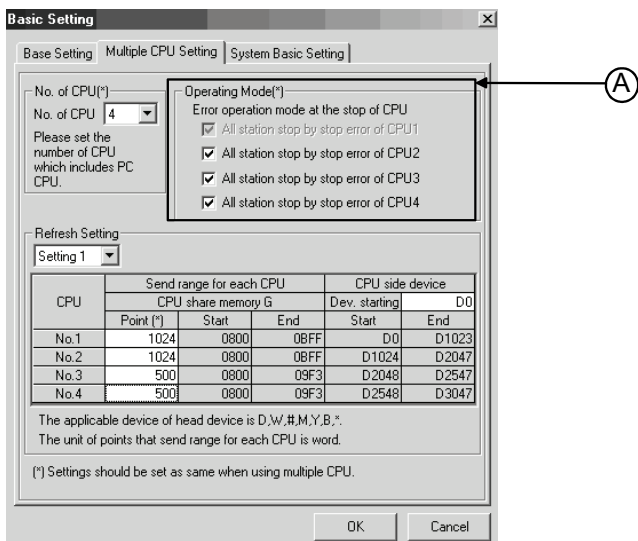
a) CPU No.1 的 QCPU 发生 CPU DOWN 故障时, CPU No.2, 3 和 4 的所有 QCPU/Q173CPU(N)/Q172CPU(N) 发生 MULTI CPU DOWN 故障 (出错代码: 7000) 且多 CPU 系统停止。(注 1)

b) 用以下步骤复位系统:

- 1) 使用 GX Developer 的 PC 诊断功能检查 CPU No.1 中故障发生的原因。
- 2) 消除故障发生原因。
- 3) 复位 CPU No.1 的 QCPU 或重启电源。
复位 CPU No.1 的 QCPU 或重启电源复位多 CPU 系统中的所有 CPU, 系统可复原。

(2) CPU No.2, 3 或 4 发生 CPU DOWN 故障时

如 CPU No.2, 3 或 4 的 QCPU, Q173CPU(N) 或 Q172CPU(N) 发生 CPU DOWN 故障, 根据多 CPU 设置中的“运行模式”设置, 总系统可停止或不停止。在默认值存在情况下, 当任一 CPU 发生 CPU 停止故障时所有 CPU 将停止。如不希望 QCPU, Q173CPU(N) 或 Q172CPU(N) 发生故障后停止所有 CPU, 选择不使所有 CPU 停止的 CPU。(参见箭头 A。)



- a) 当已设定为“CPU No.n 出错时停止所有 CPU”项目中的 CPU 发生停止故障时,其他 CPU 的所有 QCPU/Q173CPU(N)/Q172CPU(N) 将发生 MULTI CPU DOWN 故障(出错代码: 7000)且多 CPU 系统将停止。(注-1)
- b) 当未设定为“CPU No.n 出错时停止所有 CPU”项目中的 CPU 发生 CPU 停止故障时,其他 CPU 的所有 CPU 将发生 MULTI CPU ERROR (出错代码: 7010)且继续运行。

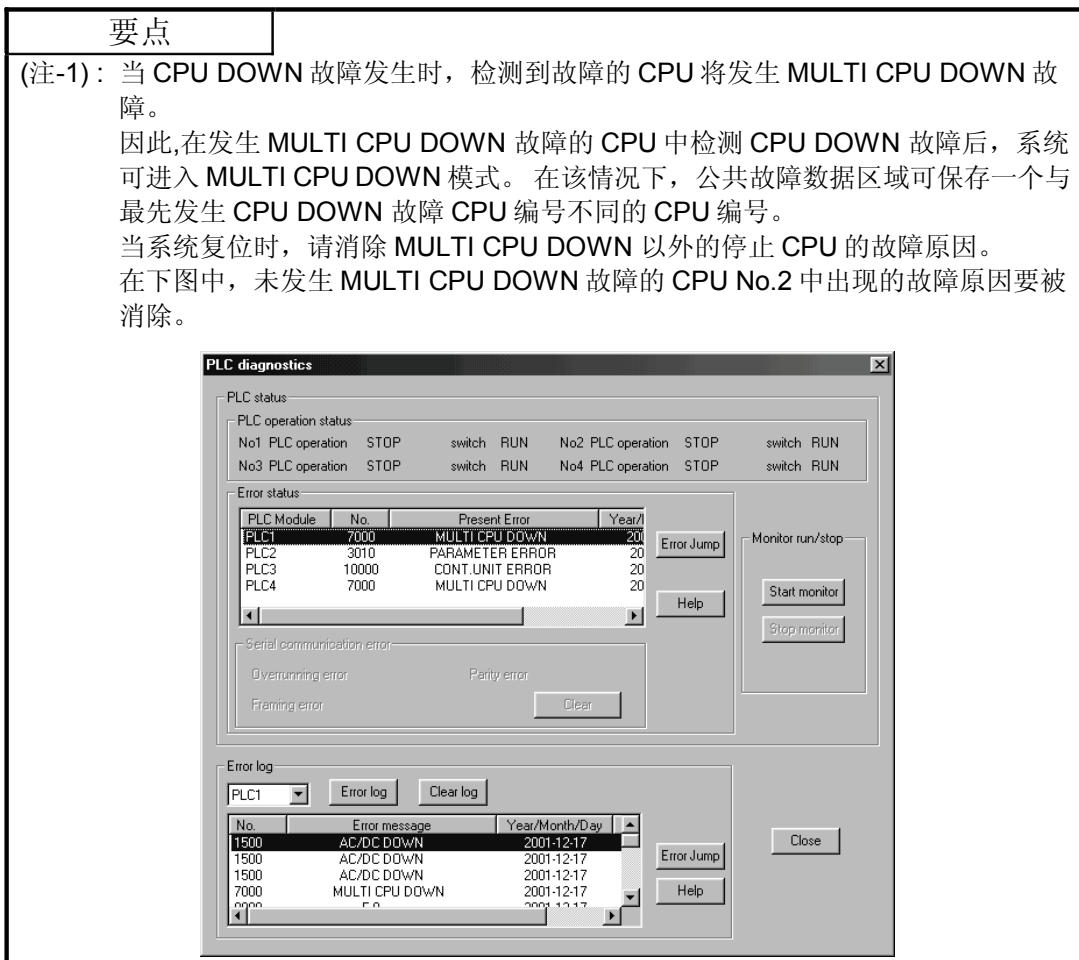
要点

(注-1): 当 CPU DOWN 故障发生时,检测到故障的 CPU 将发生 MULTI CPU DOWN 故障。

因此,在发生 MULTI CPU DOWN 故障的 CPU 中检测 CPU DOWN 故障后,系统可进入 MULTI CPU DOWN 模式。在该情况下,公共故障数据区域可保存一个与最先发生 CPU DOWN 故障 CPU 编号不同的 CPU 编号。

当系统复位时,请消除 MULTI CPU DOWN 以外的停止 CPU 的故障原因。

在下图中,未发生 MULTI CPU DOWN 故障的 CPU No.2 中出现的故障原因要被消除。



- c) 用以下步骤复位系统:
- 1) 使用 GX Developer 的 PC 诊断功能检查发生故障的 CPU 及故障原因。
 - 2) 如 Q173CPU(N)/Q172CPU(N) 发生故障且出错代码为 10000, 使用 GSV 故障列表检查原因。
 - 3) 消除故障原因。
 - 4) 复位 CPU No.1 的 QCPU 或重启电源。
 - 5) 复位 CPU No.1 的 QCPU 或重启电源复位多 CPU 系统中所有的 CPU, 系统将复原。

1 概述

③ 运动 CPU 发生故障时操作 运动 CPU 发生故障时操作如下所示

种类	故障类型	操作	备注
不能运行故障	系统设置故障	从最开始就不运行 (不运转)。	<ul style="list-style-type: none"> 所有实际输出 PY 点断开。 不影响其他 CPU。
	WDT 故障	因故障类型而不同。	<ul style="list-style-type: none"> 所有实际输出 PY 点断开。 根据参数设置其他 CPU 也可能停止。
	自诊断故障	CPU 停止故障发生时停止。	
	其他 CPU DOWN 故障	相当于 STOP (M2000 OFF) 的操作。根据“CPU 停止故障发生时操作模式”设置。	<ul style="list-style-type: none"> 所有实际输出 PY 点断开。
可以继续运行故障	自诊断故障	故障发生时操作继续	<ul style="list-style-type: none"> 只有相应程序停止(根据故障类型继续运行)。 实际输出 PY 保持输出。 不影响其他 CPU。
	运动 SFC 故障	单个系统或轴停止不需要运动 CPU 所有处理都停止。	
	轻微错误		
	严重错误		
	伺服故障		
	伺服程序设置故障		

1 概述

1.5 系统设置

1.5.1 系统数据设置

下表列出了需设置的系统数据项目。

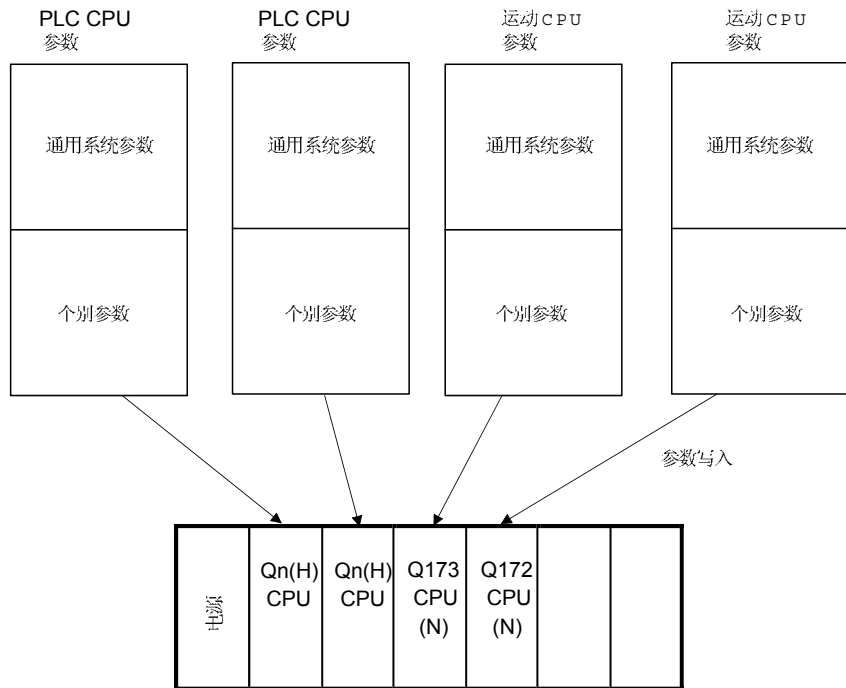
	项目	设置范围	初始值	备注	
公共系统参数	基板设置	CPU 基板	2/3/5/8/10/12 插槽	CPU 基板：2 插槽	设定 CPU 基板或扩展基板上插槽数。
		扩展基板	无/2/3/5/8/10/12 插槽	无	
	多 CPU 设置	多 CPU 数	2/3/4 模块	2 模块	设置包括 PLC CPU 的多 CPU 总数。
		自动刷新设置	对于设置 1 到 4，1 个 CPU 最多可设置软元件 (D/W/#/M/Y/B)2k 字。	无设置	使用多 CPU 共享内存存在 CPU 之间设置自动刷新
		CPU 停止错误时操作模式	CPU No. 1/2/3/4 发生故障时停止/不停止所有 CPU。(根据安装的多 CPU 数设置范围有所不同。)	CPU No. 1/2/3/4 发生故障时停止所有 CPU	设置是否在各 CPU 发生 CPU 停止故障时停止整个系统。
	运动插槽设置	模块配置	在 CPU 基板及扩展基板插槽以内。	无设置	在 CPU 基板和/或扩展基板中配置自 CPU 控制的模块。
模块个别设置		根据模块有所不同	根据模块有所不同	设置自 CPU 控制的各模块的细节	
个别参数	基本系统设置	运算周期设置	0.8 ms/1.7 ms/3.5 ms/7.1 ms/14.2ms/自动设置	自动设置	设置运动控制的运算周期
		STOP 到 RUN 的操作	切换 (STOP 到 RUN) M2000 设定 /切换 (STOP 到 RUN) +寄存器为 1M 2000 设定为 ON。	切换 (STOP 到 RUN) M2000 为 ON	设置 PLC 就绪标志(M2000)为 ON 的条件
		紧急停止输入设置 (注)	无设置/X (PX) (0 到 1FFF)/M (0 到 8191)	无设置	设置紧急停止的位软元件
		锁存范围设定	M (0 到 8191)/B (0 到 1FFF)/F (0 到 2047)/D (0 到 8191)/W (0 到 1FFF)	无设置	设置软元件内存的锁存范围。
	自 CPU 安装位置设置	设定 0/1/2 插槽设置对应自 CPU/其他 CPU/CPU (空的)。(根据安装的多 CPU 数设置范围有所不同。)	无设置 (当安装 2 个 CPU 时，插槽 0 被固定为自 CPU。)	设置在 CPU 基板上的自 CPU 安装位置	
	伺服放大器/电机设置	Q173CPU(N)可含最多 4 系统/32 轴。Q172CPU(N)可含最多 1 系统/8 轴。	无设置	设置伺服放大器及伺服电机的型号，轴号等。	
	快速读设置	一个 Q172EX/Q173PX 模块及一个输入模块	无设置	设置高速读出数据。高速读出功能参见《Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册 (实模式)》。	
	电池设置	外部电池不用/使用	不用外部电池	设置是否使用外部电池。如电源关闭一个月或更长时间，必须用外部电池备份数据。外部电池参见《Q173CPU(N)/Q172CPU(N) 用户手册》。	

(注)：除了紧急停止输入设置，伺服放大器的紧急停止端子也可用来执行紧急停止。

1.5.2 通用系统参数

(1) 操作多 CPU 系统的参数

在多 CPU 系统中，各 CPU 的通用系统参数及个别参数被设置并写入各 CPU。运动 CPU 中，与整个多 CPU 系统相关的系统设置的项目必须与 PLC CPU 中的参数设置相同。



② 多 CPU 系统中的通用参数

在运动 CPU 中, 初始化期间 CPU No. 1 的 PLC CPU 中的参数与下表中的参数进行验证。

参数不一致引发 PARAMETER ERROR (出错代码: 3012), 因此下表所列参数必须在运动 CPU 与 CPU No. 1 的 PLC CPU 之间同样设置。(如系统设置在运动 CPU 中被改变时, 有必要进行复位。因此, 参数只在初始化期间检查。)

PLC CPU 可通过 GX Developer 中的《多 CPU 参数应用》使用其他 CPU 的参数。运动 CPU 不具备此项功能, 然而, 各运动 CPU 必须分别设置通用参数。

参数类型		核实项目	备注
在运动 CPU 中的名称	PLC CPU 中名称		
多 CPU 设置	CPU 数目		CPU 模块数
	操作模式		CPU 停止错误发生时操作模式
	自动刷新设置		自动刷新点数
运动插槽设置	I/O 分配	控制 CPU	控制 CPU 号
基板设置		基板	基板总数
		基板	基板号
			基板插槽数

- 只有运动 CPU 侧系统设置中的模块数被验证。

- 如 PLC CPU 侧不进行基板设置则不进行验证。

② 多CPU设置

在SW6RN-GSV □ P 里的多CPU设置（运动CPU设置）里及GX Developer 里的多CPU设置（PLC CPU设置）里以下项目设置相同。

- CPU 模块数
- CPU 停止故障发生时操作模式
- 自动刷新点数 (所有CPU的1到4设置相同)

?SW6RN-GSV □ P 中的多CPU设置(运动CPU设置)

Basic Setting

Base Setting Multiple CPU Setting System Basic Setting

No. of CPU(*)
No. of CPU 2
Please set the number of CPU which includes PC CPU.

Operating Mode(*)
Error operation mode at the stop of CPU
 All station stop by stop error of CPU1
 All station stop by stop error of CPU2
 All station stop by stop error of CPU3
 All station stop by stop error of CPU4

Refresh Setting
Setting 1

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Dev. starting	End
No.1	256	0800	08FF	w0	w0FF
No.2	256	0800	08FF	w100	w1FF
No.3					
No.4					

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

OK Cancel

Annotations: CPU 模块数 (points to No. of CPU), CPU停止错误时的操作模式 (points to Operating Mode), 自动刷新点数 (points to Refresh Setting).

?GX Developer 中的多PLC设置(PLC CPU设置)

Multiple PLC settings

No. of PLC (*)
No. of PLC 2

Out of group input/output settings (*)
 The input condition outside the group is taken
 The output condition outside the group is taken

Operating mode (*)
Error operation mode at the stop of PLC
 All station stop by stop error of PLC1
 All station stop by stop error of PLC2
 All station stop by stop error of PLC3
 All station stop by stop error of PLC4

Refresh settings
Change screens Setting 1

PLC	Send range for each PLC			PLC side device	
	Point (*)	Start	End	Dev. starting	End
No.1	256	0800	08FF	w100	w1FF
No.2	256	0800	08FF	w200	w2FF
No.3					
No.4					

The applicable device of head device is B,M,Y,D,W,R,ZR.
The unit of points that send range for each PLC is word.

(*) settings should be set as same when using multiple PLC.

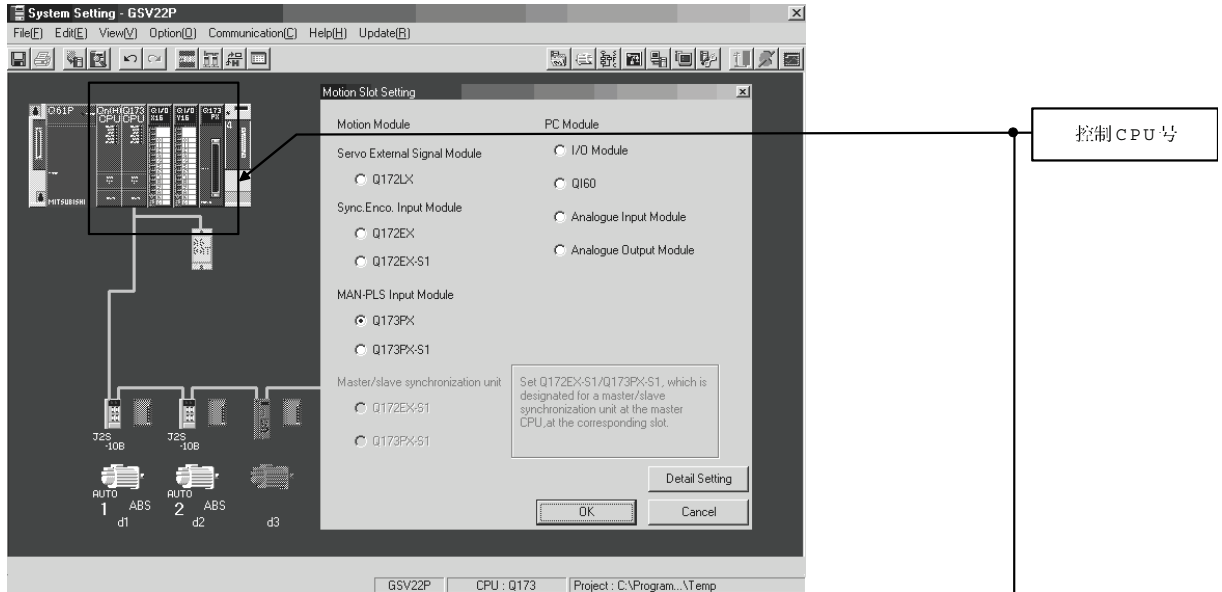
Diversion of multiple PLC parameter Check End Cancel

Annotations: CPU 模块数 (points to No. of PLC), CPU停止错误时的操作模式 (points to Operating mode), 自动刷新点数 (points to Refresh settings).

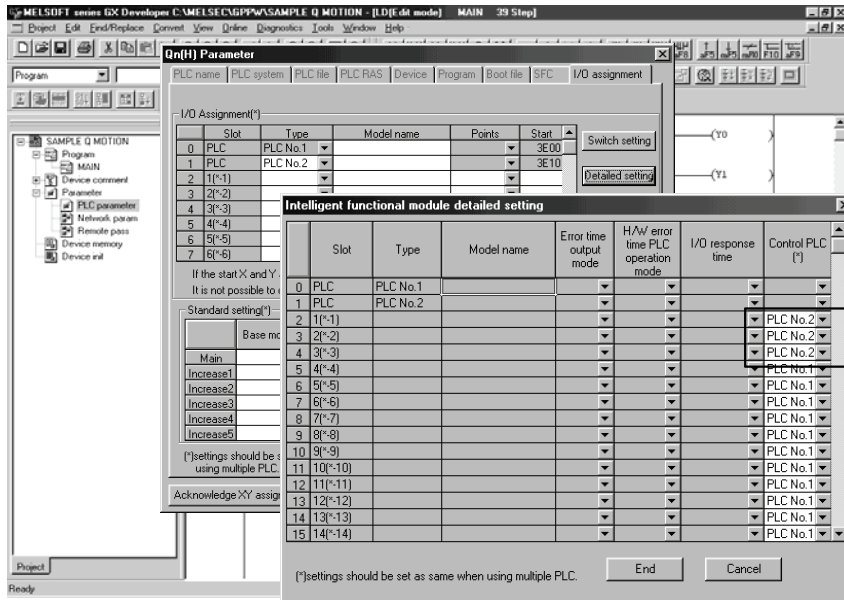
b) 运动插槽设置

在 SW6RN-GSV □ P 的运动插槽设置 (运动 CPU 设置) 中设置自 CPU 控制的模块。
在 GX Developer 中, 设置控制运动 CPU 的插槽作为 I/O 分配设置 (PLC CPU 设置) 中的运动 CPU 的 CPU 号。

SW6RN-GSV □ P 中运动插槽设置 (运动 CPU 设置)



GX Developer 中 I/O 分配设置 (PLC CPU 设置)



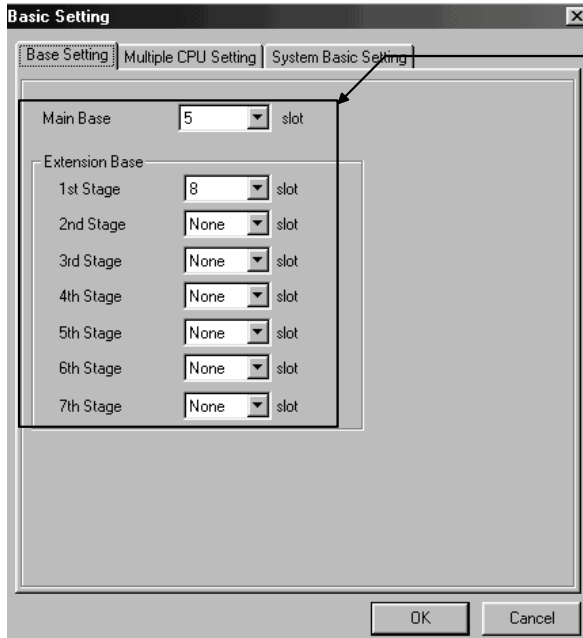
1 概述

④ 基板设置

在 SW6RN-GSV □ P 内的基板设置(运动 CPU 设置)与 GX Developer 里的 I/O 分配设置(PLC CPU 设置)之间同样设置各基板里的基板总数及插槽数。

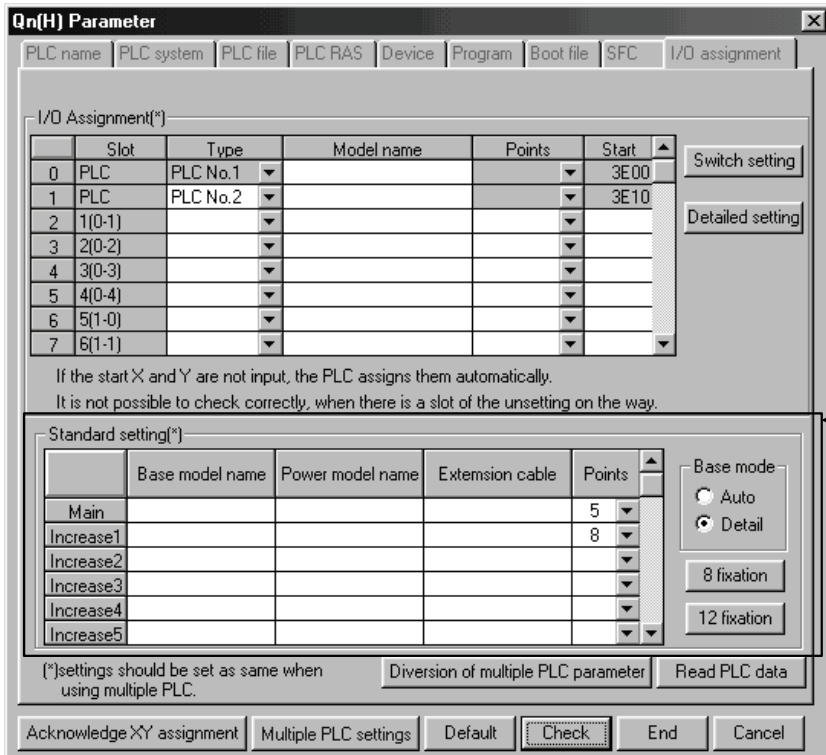
在 GXDeveloper 中, 可通过设置基板模式“自动”省去详细设置。

• SW6RN-GSV □ P 中基板设置(运动 CPU 设置)



各基板总数及插槽数

?GX Developer 中 I/O 分配设置(PLC CPU 设置)



(注): 可以只设置运动 CPU, 不设置 PLC CPU

要点

运动 CPU 将 GOT 视作基板上占用 32 点数的智能功能模块（扩展基板数及插槽号在 GOT 的参数中设置。）运动 CPU 的系统设置（基板设置）中，该运动 CPU 的扩展基板数及基板插槽数在 GOT 的参数中设置。并且设置包括了 GOT 的参数中设置的插槽号的插槽数。

<示例>

示例"扩展基板数：2 插槽号：0"如下所示。（在基板设置中第二块基板插槽数应设置为“2”。）



参见 GOT 需用的图形软件手册。

当基板连接电缆连接到扩展基板，而该基板的基板插槽数在系统设置中设置为“无”时(基板设置“SP.UNIT LAY ERR”(出错代码:2124)将发生)。

1.5.3 个别参数

(1) 系统基本设置

以下阐述了系统基本设置中需设置的项目。

a) 运算周期设置

- 1) 设置运动运算周期 (计算定位指令并发送到伺服放大器的周期)。设置范围为 0.8ms/1.7ms/3.5ms/7.1ms/14.2ms/ 自动设置。与 0.8ms 对应的实际运行周期为 0.888...ms。类似的, 1.7ms 对应 1.777...ms, 3.5ms 对应 3.555...ms, 7.1ms 对应 7.111...ms, 14.2ms 对应 14.222...ms。
- 2) 缺省值为“自动设置”。当选择“自动设置”时, 根据下表设置运行周期, 该表基于系统设置中设定的伺服放大器轴数。

操作系统	轴数	运行周期设置
SV13	1 到 8 轴	0.8 ms
	9 到 16 轴	1.7 ms
	17 到 32 轴	3.5 ms
SV22	1 到 4 轴	0.8 ms
	5 到 12 轴	1.7 ms
	13 到 24 a 轴	3.5 ms
	25 到 32 轴	7.1 ms

- 3) 如运动运算时间已超出运算周期, 运算周期超出标志 (M2054) 为 ON。即使在选择了“自动设置”时, 根据控制情况的不同运动运算也可能超出运算周期。运动运算 (单位 μs) 的实际时间保存在 D9188, 而运算周期 (单位 μs) 的当前设置保存在 D9197。监视这些特殊的寄存器并调整运行周期的设定值以便运动运算的实际时间不超出设定的运行周期。(在运动 CPU 中可能发生 WDT 或其他故障。)
- 4) MR-H□BN 不支持 0.8 [ms] 的运算周期。即使系统设置中设定为 0.8[MS], 1.7 [ms] 被用做实际的运算周期。
- 5) MR-J2S-□B 对应 0.8 [ms], 1.7 [ms] 的运算周期对应 B0 或更新版本。当使用 A4 或更早版本的 MR-J2S-□B 时, 请设置运算周期为 3.5 [ms] 或 7.1 [ms]。
- 6) 矢量变频器不支持 0.8 [ms] 及 1.7 [ms] 的运算周期。如系统设置中设定 FR-V500, 3.5[ms] 被用做实际运行周期即使设定周期为 0.8 [ms] 或 1.7 [ms]。

(b) STOP → RUN 的运行设置

设置其中“PLC就绪”标志 (M2000) 打开的条件。选择以下其中一个：

1) 切换时M2000 变为ON (STOP → RUN) (缺省)

M2000 从 OFF 转到 ON 的条件

- 将 RUN/STOP 开关从 STOP 侧扳到 RUN 侧。
- RUN/STOP 开关设置在 RUN 侧时，打开电源或取消复位。

M2000 从 ON 转到 OFF 的条件

- 将 RUN/STOP 开关从 RUN 侧扳到 STOP 侧。

2) 切换时 M2000 ON (STOP → RUN) + 寄存器设置为 1

(当开关设置到 RUN 侧且设置寄存器设置为 1 时，M2000 变为 ON)

M2000 从 OFF 转到 ON 的条件

- RUN/STOP 开关设置到 RUN 侧时，“PLC就绪”标志设定寄存器置为 1 (D704)。(运动 CPU 检测出 D704 中的最低位从 0 到 1 的变化)。

M2000 从 ON 转到 OFF 的条件

- RUN/STOP 开关设置到 RUN 侧时，“PLC就绪”标志设定寄存器置为 0 (D704)。(运动 CPU 检测出 D704 中的最低位从 1 到 0 的变化)。
- 将 RUN/STOP 开关从 RUN 侧改变到 STOP 侧。

(c) 紧急停止输入设置

指定用来执行紧急停止的位软元件，该紧急停止可立即停止所有的伺服放大器轴。可指定 X (EX) 或 M。没有设定缺省值。设定的位软元件被指定为触点 B 且对应位软元件的 ON/OFF 执行以下管理。

- 位软元件变 OFF — 紧急停止输入为 ON (紧急停止)
- 位软元件变 ON — 紧急停止为 OFF (紧急停止解除)

(d) 锁存范围设置

分别为 M, B, F, D 及 W 设置以下锁存范围。

- 用锁存清除键可清除锁存的范围。(锁存 (1))
- 用锁存清除键不可清除锁存的范围。(锁存 (2))

② 个别模块设置

各模块的设置项目如下所示。

各模块的设置项目

模块名称		项目	设置范围	初始值	可用模块数	
					Q173CPU (N)	Q172CPU (N)
Q172LX	伺服外部信号输入模块	外部信号设置	设置 8 轴输入使用的轴数	使用 1 到 8 轴	4	1
		DOG/CHANGE 从 OFF 到 ON/ON 到 OFF	DOG/CHANGE 输入从 OFF 到 ON 或从 ON 到 OFF	从 OFF 到 ON		
		输入响应时间	0.4/0.6/1 ms (DOG/CHANGE 响应时间)	0.4 ms		
Q172EX	串行编码器输入模块	串行编码器使用设置	使用/未使用	未使用	6 (SV22)	4 (SV22)
		输入响应时间	0.4/0.6/1 ms (TREN 反应时间)	0.4 ms		
		高速读出设置	使用/未使用	未使用		
Q173PX	手动脉冲发生器输入模块	串行编码器/手动脉冲发生器使用设置	使用/未使用	未使用	1 (SV13)	1 (SV13)
		输入响应时间	0.4/0.6/1 ms (TREN 反应时间)	0.4 ms	4 (SV22)	3 (SV22)
		高速读出设置	使用/未使用	未使用		
QI60	中断模块	输入响应时间	0.1/0.2/0.4/0.6/1 ms	0.2 ms	1	1

各模块的设置项目(续)

模块名称		项目	设置范围	初始值	可用模块数	
					Q173CPU (N)	Q172CPU (N)
QX□	输入模块	起始 I/O 号	00 到 FF0 (16 点为单位)	0	共 256 点 以下	共 256 点 以下
		I/O 点数	0/16/32/64/128/256	16		
		高速读出设置	设置是否使用	不使用		
		输入响应时间设置(括弧内 为高速输入模块时)	1/5/10/20/70 ms (0.1/0.2/0.4/0.6/1 ms)	10 ms (0.2 ms)		
QY□	输出模块	起始 I/O 号	00 到 FF0 (16 点为单位)	0		
		I/O 点数	0/16/32/64/128/256	16		
QH□/QX□Y□	输入/输出组 合模块	起始 I/O 号	00 到 FF0 (16 点为单位)	0		
		I/O 点数	0/16/32/64/128/256	16		
		输入反应时间设置	1/5/10/20/70 ms	10 ms		
		快速读设置	使用/未使用	未使用		
Q6□AD□	模拟输入 模块	起始 I/O 号	00 到 FF0 (16 点为单位)	0		
		输入范围设置	4 到 20mA/0 到 20mA/1 到 5V/0 到 5V/-10 到 10V/0 到 10V/使用者范 围	4 到 20mA		
		温漂补偿	有/无	使用		
		分辨率模式	通常/高	通常		
		操作模式	通常模式(A/D 转换)/偏置增益模式 的设置	通常 (A/D 转换)		
Q6□DA□	模拟输出 模块	起始 I/O 号	00 到 FF0 (16 点为单位)	0		
		输出范围设置	4 到 20mA/0 到 20mA/1 到 5V/0 到 5V/-10 到 10V/用户范围	4 到 20mA		
		HOLD/CLEAR 功能设置	HOLD/CLEAR	不使用		
		输出模式	通常(不同步)/同步输出模式的设定	通常 (转换)		
		分辨率模式	通常/高	通常		
		操作模式	通常(D/A 转换)/ 偏置增益设置	通常 (D/A 转换)		

③ 系统设置故障

在以下情形下, 运动 CPU 发生系统出现故障:

故障名称	出错代码 (注-1)	故障原因	检查计时	故障发生时操作	
LAY ERROR (SL * *)	10000 (注-2)	• 系统设置中设定了的插槽空或安装了不同的模块。	当电源开启时/ 该键被复位	不可启动 (运动 CPU 系统 设置错误)	
AXIS No. MULTIDEF		• 系统设置中设置了重复轴号。			
AMP No. SETTING		• 系统设置中未设置任何轴。			
AXIS No. ERROR		• 系统数据未被写入。			
I/O POINTS OVER		• 系统设置中实际 I/O 点设置超过 256。			
SP. UNIT LAY ERROR	2121	• CPU 模块安装在非 CPU 插槽或 0~2 以外插槽中。			不可启动(多 CPU 系统 CPU DOWN 故障)
SP. UNIT LAY ERROR	2124 (注-3)	• 模块安装在 65 插槽或以后的插槽中。 • 模块安装在基板设置中设置为“无”的基板中。			
SP. UNIT LAY ERROR	2126	• CPU 模块之间有空闲插槽。 • PLC CPU 模块之间安装了 PLC CPU 之外的模块。			
PARAMETER ERROR	3010	• 参数中设置的 CPU 模块数目与多 CPU 系统中的实际安装的不同。			
PARAMETER ERROR	3012	• 多 CPU 系统参数中基准轴与设定内容不同。			
PARAMETER ERROR	3013	多 CPU 系统中多 CPU 自动刷新设置为以下情形之一。 • 当位软元件被设置为刷新元件时, 刷新起始软元件设置了 16 的倍数以外的编号。 • 指定一个不可指定软元件。 • 传送点数为奇数。			
MULTI EXE. ERROR	7010	• 多 CPU 系统中安装的 CPU 故障。 • 多 CPU 系统中安装了版本不一致的 CPU。 (在功能版本为 B 的 PLC CPU 中检测到 1 个错误。) • 多 CPU 系统电源 ON 后, CPU No.2 到 4 中的任一 CPU 号的复位解除。 (该故障只在复位解除的 CPU No.上发生。)			

注-1) : 该出错代码存储于多 CPU 间共享内存的自 CPU 操作信息区域的诊断错误区。

注-2) : 当显示出错代码 10000 时, M2041 (“系统设置故障”标志) 为 ON, 且上述故障名称在编程软件包的故障列表监视器上显示。

注-3) : 有必要在运动 CPU 的系统设置中进行基板设置, 即使是那些未安装自 CPU 控制模块的基板。

1.6 I/O 号分配

多CPU系统中使用的I/O号包括运动CPU用来与I/O模块/智能功能模块进行通讯的I/O号及PLC CPU与运动CPU之间通讯所用的I/O号。以下对I/O号及I/O号的分配进行说明。

1.6.1 I/O 模块及智能功能模块的I/O 号

在多CPU系统中，从PLC CPU看，I/O号的“OH”位置(插槽)与单独CPU状态下不同。然而，运动CPU控制模块的I/O号可被独立分配。

(1) I/O 号的“OH”位置

- a 在多CPU系统中，在多CPU参数中设定的数目的插槽被PLC CPU/运动CPU占用。
- b I/O模块及智能功能模块被安装在PLC CPU/运动CPU右侧的插槽内。
- c 运动CPU控制的模块的I/O号可被独立分配。PLC CPU控制的模块的I/O号朝右侧按顺序分配，把最左边的I/O模块或智能功能模块作为“OH”，依次进行分配。
- d I/O号表示
 - 用运动CPU接收ON/OFF数据被视为输入(PX)，而从运动CPU输出ON/OFF数据被视为输出(PY)。
 - I/O号用十六进制数表示。

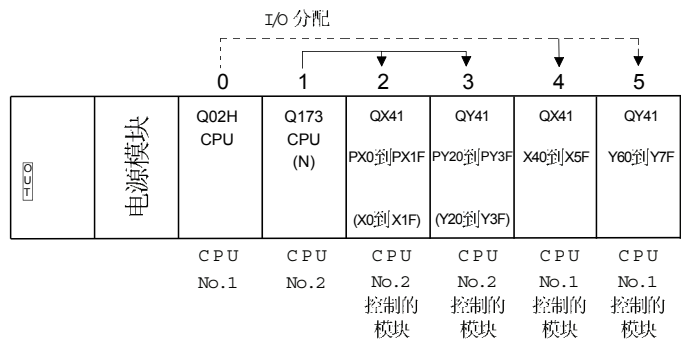
(2) 分配I/O号到运动CPU控制模块

三菱推荐I/O号分配应设置为所有CPU的通用连续编号。

然而，运动CPU控制的输入模块,输出模块及输入/输出混合模块的I/O编号也可与PLC CPU控制模块的I/O号独立设置。

(运动CPU控制模块的I/O号用PX/PY表示。)

运动CPU控制模块的I/O号在PLC CPU的I/O分配中无效。



③ 用 PLC CPU 设置运动 CPU 控制模块

当运动 CPU 控制模块在 PLC CPU 的 I/O 分配设置设定中时，参照下表。

(PLC CPU 将 Q172LX, Q172EX 及 Q173PX 视做占有 32 点的智能功能模块。) 类型和点数可不设置。

模块名称	类型	点数	备注
输入模块	输入	根据模块选择	<ul style="list-style-type: none"> 对于控制 CPU，设置与运动 CPU 相对应的 CPU(必须)。 可不设置点数和类型。
输出模块	输出		
输入/输出混合模块	混合 I/O		
模拟输入模块	模拟输入	16 点	
模拟输出模块	模拟输出		
中断模块 (QI60)	中断		
Q172LX	智能	32 点	
Q172EX		32 点	
Q173PX		32 点	

要点

- (1) 设置运动 CPU 的 I/O 软元件在 PX/PY000 到 PX/PYFFF 范围内。设置实际 I/O 点数在 256 点数之内。(I/O 号可以不连续。)
- (2) 对于运动 CPU，Q172LX, Q172EX, Q173PX 及 QI60 不包含在实际 I/O 点数内。

1.6.2 QCPU 与Q173CPU(N)/Q172CPU(N)的I/O 编号

在多 CPU 系统中使用以下指令，把 I/O 编号被分配到 PLC CPU/ 运动 CPU，进行 PLC CPU 与运动 CPU 之间的通讯：

- 多 CPU 专用指令
- 运动 CPU 专用指令
- 多 CPU 间通讯专用指令

PLC CPU/ 运动 CPU 的 I/O 编号在已安装插槽是固定的，不能改变。

下表列出了安装于多 CPU 系统基板的 PLC CPU/ 运动 CPU 的 I/O 编号。

CPU 安装位置	QCPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 编号	3E00H	3E10H	3E20H	3E30H

在以下情形下使用 PLC CPU/ 运动 CPU 的 I/O 编号：

- 当使用 S. TO 指令向自 CPU 的共享 CPU 内存写入数据时。
- 当使用 FROM 指令从其他 CPU 的共享 CPU 内存读出数据时。
- 当使用智能功能模块软元件 (U □ \G □) 从其他 CPU 的共享 CPU 内存读出数据时。
- 当使用 "S(P). DDRD" 指令 PLC CPU 直接从运动 CPU 读出软元件数据时。
- 当使用 "S(P). DDWR" 指令从 PLC CPU 直接向运动 CPU 写入软元件数据时。

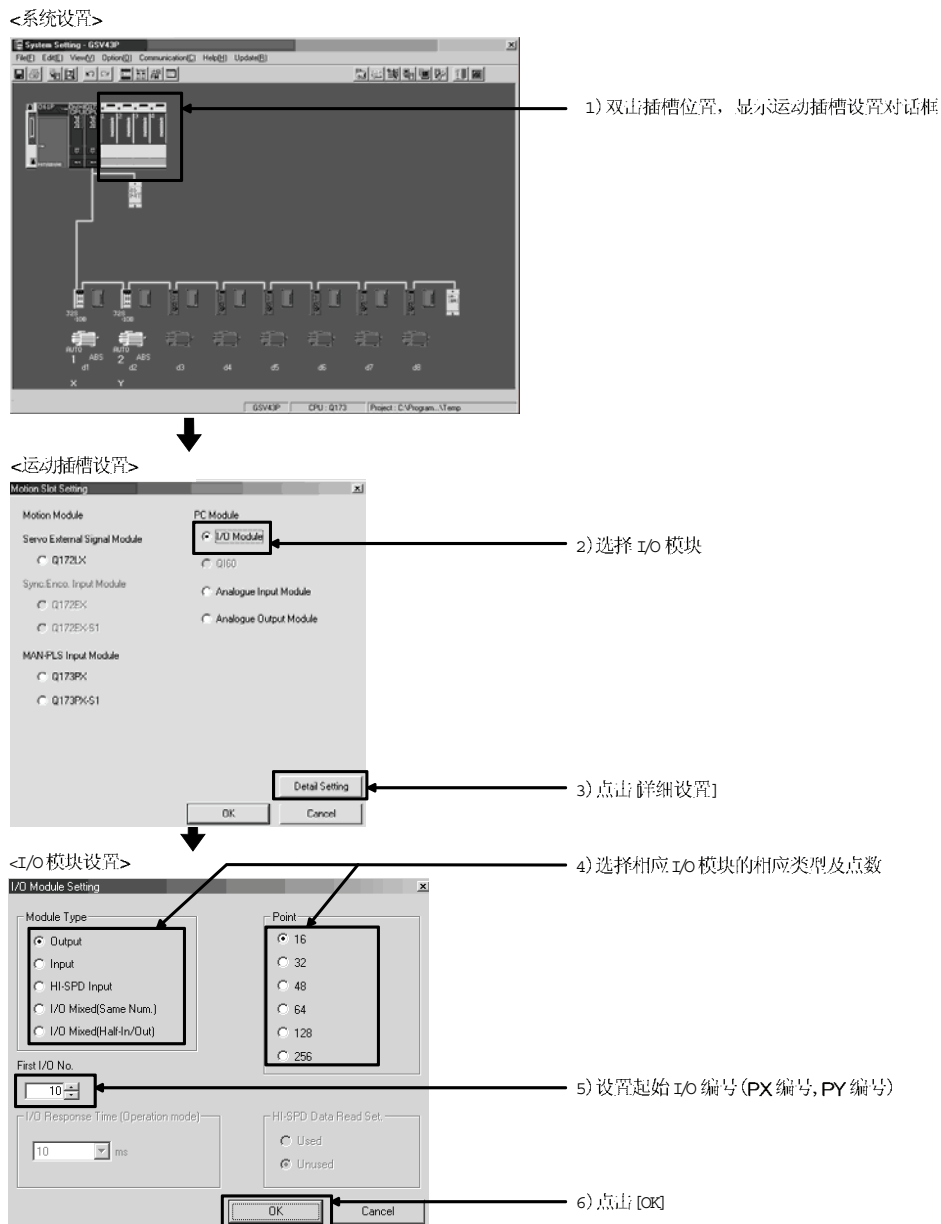
备注

- PLC CPU 与运动 CPU 间的通讯参见“3 多 CPU 系统中 PLC CPU 与运动 CPU 间的通讯”。

1.6.3 使用 SW6RN-GSV □ P 设置 I/O 编号

SW6RN-GSV □ P 系统设置中的运动 CPU 的 I/O 编号设置步骤如下所示。
 在运动 CPU 中, 在系统设置中通过设置各 CPU 基板或扩展基板插槽里使用的模块, 相应插槽的控制 CPU 被指定为自 CPU。输入模块, 输出模块及混合 I/O 模块也需要设置 I/O 号。

系统设置的详细操作步骤参见 SW6RN-GSV □ P 的帮助。



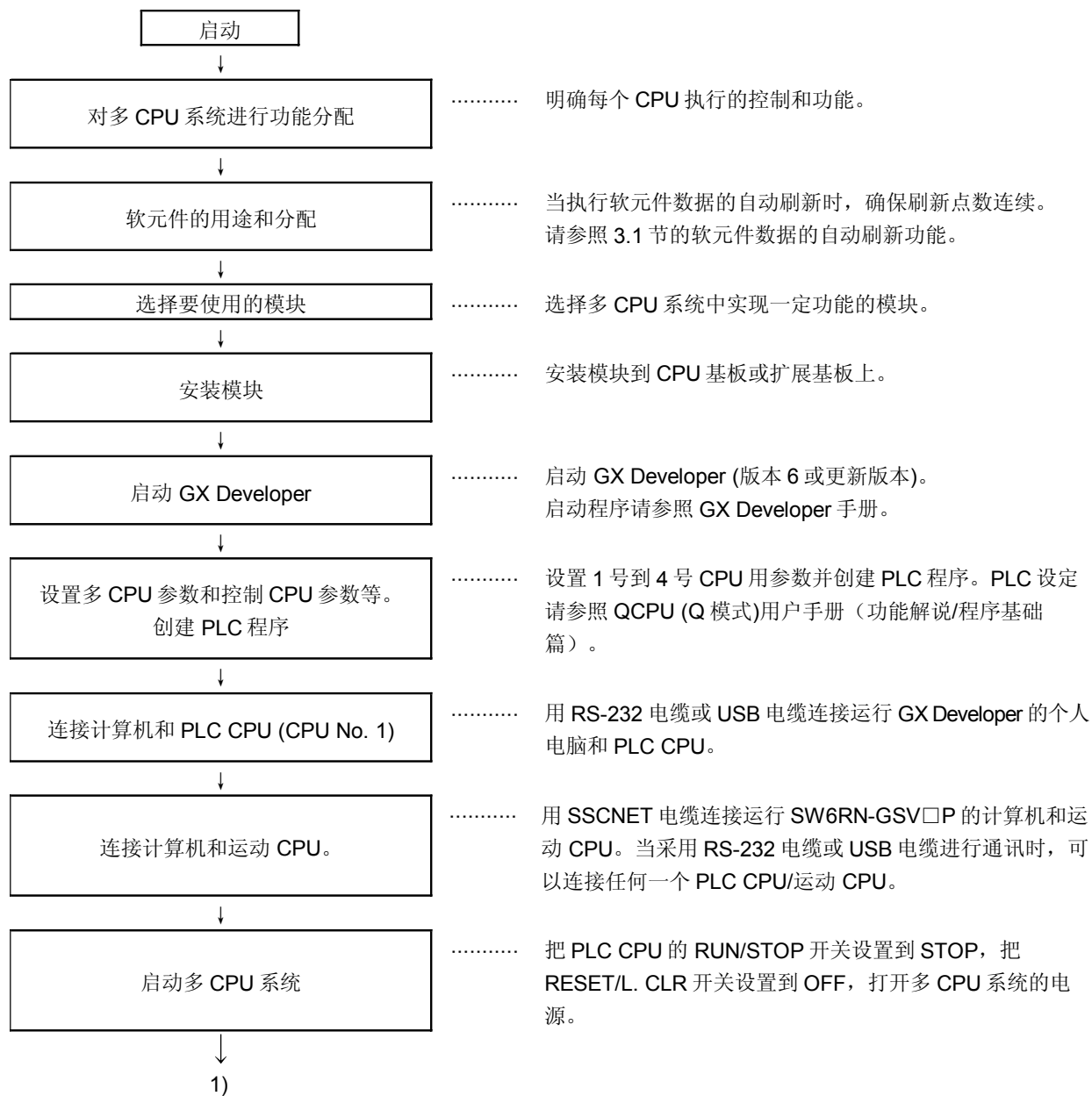
要点

不象 PLC CPU 那样如果省略了 I/O 分配, 将进行 I/O 自动分配。在运动 CPU 中, 确认在系统设置中为每个使用的模块设置起始 I/O 号。

2. 多 CPU 系统的启动

这部分描述启动 CPU 系统的标准顺序。

2.1 多 CPU 系统的启动流程



2 多 CPU 系统的启动



(注):运动 CPU 操作系统软件的安装参照"Q173CPU(N)/Q172CPU(N) 用户手册"第 5 章

3. 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

在多 CPU 系统中 PLC CPU 和运动 CPU 之间可执行如下任务

- 通过 CPU 共享内存的自动刷新实现 CPU 之间的数据传输。
- 通过运动专用指令，将控制指令从 PLC CPU 传输到运动 CPU。
- 通过专用指令，从 PLC CPU 到运动 CPU 读取 / 写入软元件数据。

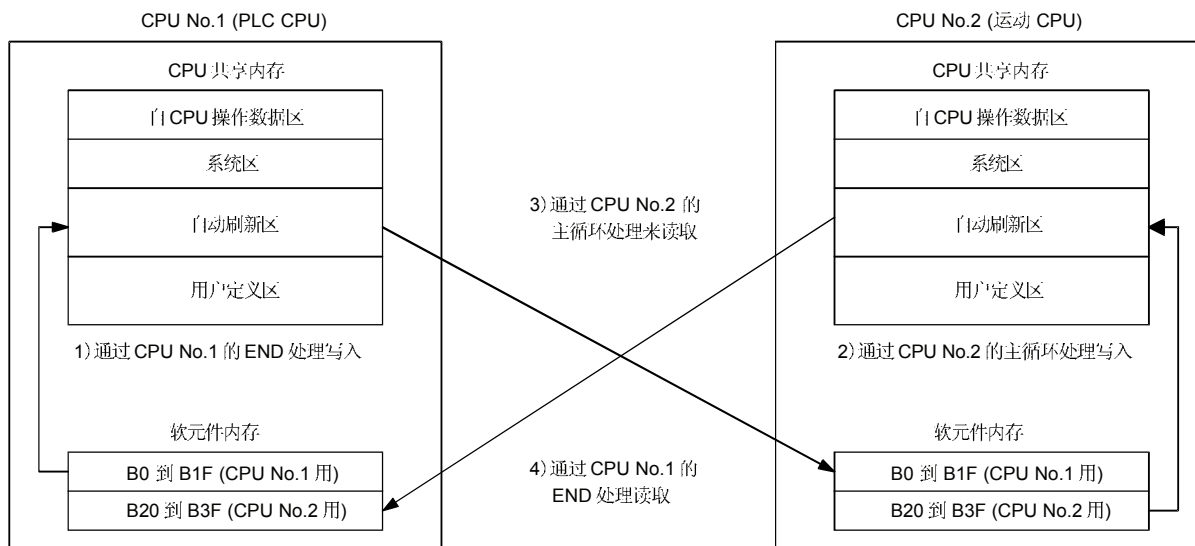
3.1 CPU 共享内存的自动刷新

(1) CPU 共享内存的自动刷新

(a) 在 PLC CPU 的 END 处理时或在运动 CPU 的主循环处理时(运动控制外的空闲时间), 自动进行 CPU 共享内存的自动刷新。

当自动刷新使用时, 其它 CPU 软元件内存的数据被自动读取, 所以其它 CPU 软元件数据可以作为自 CPU 的软元件数据使用。

CPU No.1 的 PLC CPU 的 32 点(B0 到 B1F)和 CPU No.2 的运动 CPU 的 32 点(B20 到 B3F)进行的自动刷新如下图所示。



CPU No.1 (PLC CPU) 在 END 处理时的处理内容

1) : CPU No.1 的传送软元件 B0 到 B1F 的数据传送到共享内存的自动刷新区。

4) : CPU No.2 的共享内存的自动刷新区的数据传送到 B20 到 B3F。

CPU No.2 (运动 CPU)在主循环处理时的处理内容

2) : CPU No.2的发送软元件 B20 到 B3F 的数据传送到共享内存的自动刷新区。

3) : CPU No.1 的共享内存的自动刷新区的数据传送到 B0 到 B1F。

通过如上操作，写入到 CPU No.1 的 B0 到 B1F 的数据可以读取为 CPU No.2 的 B0 到 B1F。写入到 CPU No.2 的 B20 到 B3F 的数据，可以自由读取为 CPU No.1 的 B20 到 B3F。CPU No.1 的 B0 到 B1F 可以用 CPU No.1 读取或写入，但 B20 到 B3F 与 CPU No.2 的刷新区仅可以用 CPU No. 1 读取，不能用 CPU No. 1 写入。同样的，CPU No.2 的 B20 到 B3F 可以被 CPU No.2 自由读取和写入，但 B0 到 B1F 与 CPU No.1 的刷新区仅可以被 CPU No. 2 读取，不能被 CPU No.2 写入。

(b) 执行自动刷新功能

不管 PLC CPU 和 运动 CPU 是在 RUN 或 STOP 状态，自动刷新功能都可以被执行。在 PLC CPU 或运动 CPU 中，当出现一个 CPU DOWN 出错时，自动刷新功能不能执行。

当一个 CPU 发生一个 CPU DOWN 出错，其它没有 CPU DOWN 出错的 CPU 在 CPU DOWN 出错发生前立刻保留数据。例如，当在操作块图表中 B20 是 ON 时，如果 CPU No.2 产生 CPU DOWN 出错，CPU No.1 的 B0 保持 ON。如果需要，用其它 CPU DOWN 检测信号 M9244 到 M9247，可以执行互锁。

(c) 为执行自动刷新功能，运动 CPU 的传送点数，CPU 和储存数据的软元件（自动刷新功能被执行的软元件），必须在系统设置的多 CPU 设置里设置。对于 PLC CPU，在 PC 参数的多 CPU 设置里必须相应设置适用参数。

项目		说明
刷新软元件的型号	位	Y, M, B (设置起始软元件量作为 32 位模块里的 16 倍数。)
	字	D, W, # (在 2 个字模块中设置。)
刷新软元件范围设置的数量		4 个 范围 (位与字可以混合)
每 CPU 的刷新字数		最大 8k
每 CPU 的传输字数		最大 2k (在双字模块中设置)



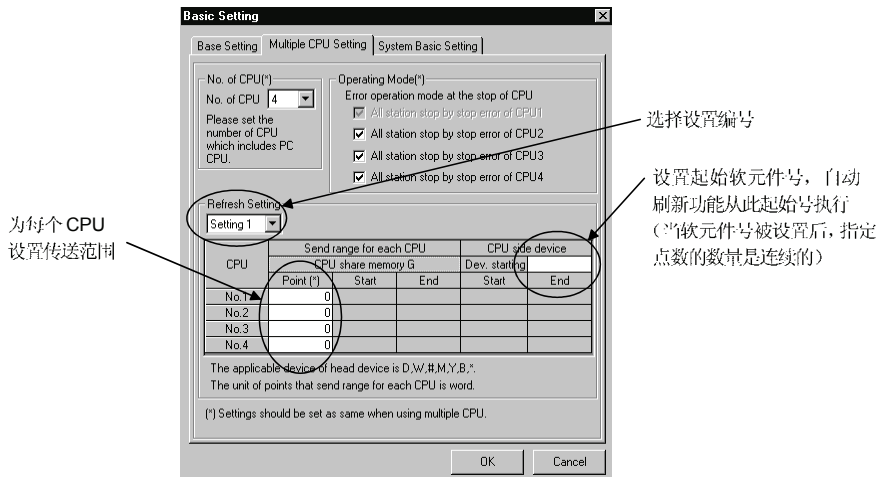
注意

- 如果需要，用其它 CPU DOWN 检测信号 M9244 到 M9247，在自动刷新功能运行时可以执行互锁。

(2) 自动刷新设置 1(自动设置)

(a) 当执行 CPU 共享内存的自动刷新功能时，用系统设置的多 CPU 设置，设置 CPU 传送点数和数据储存的软元件的数量。

关于 PLC CPU 的设置，参照“QCPU (Q 模式) 用户手册 (功能介绍 / 程序编程基础)”



(b) 为各 CPU 设置数字选择 / 发送范围(刷新范围)。

1) 4 范围的刷新设置可以用设置选择来设置。

例如：ON/OFF 数据可以用位软元件设置来刷新，而其它数据可以用字软元件设置来刷新。

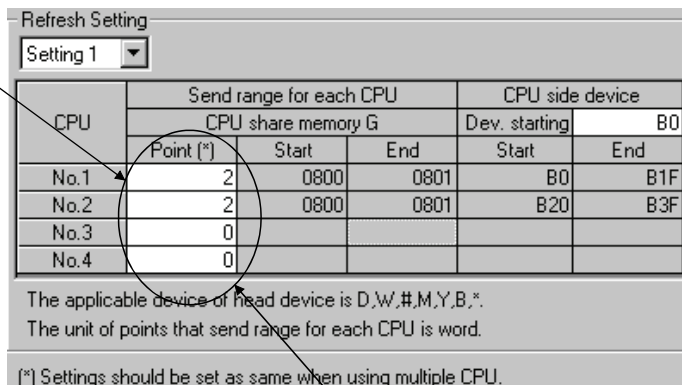
2) 设置在 2 点数 (2 字) 单元里的 CPU 共享内存里的点数数字，为各 CPU 设置字范围内。2 点数如果字软元件指定为 CPU 一侧软元件，或 32 点数如果为软元件是指定的。）

“0” 设置为点数数字表现 CPU 的发送范围的 CPU 数据不会被刷新。

假设 CPU No.1 的 32 点数(B0 到 B1F)，和 CPU No.2 的 32 点数(B20 到 B3F)将被刷新的。因在 CPU 共享内存里的一点数对应 16 位软元件点数，CPU No.1 和 No.2 的传送点数数字变成 2。

3) 组成所有 4 个范围的传送点数的最大数字是每 CPU (PLC CPU 或 运动 CPU)2k 字或所有 CPU 8k 点数 (8k 字)

每 CPU 2k 点数 (2k 字)
所有 CPU 8k 点数 (8k 字)
设置以 2 点数 (2 字) 为一个单元



在 CPU 共享内存里设置 2 点，为 CPU 一侧软元件指定位软元件，建立 32 位软元件点数。

当点数设置到 0，CPU No.3 和 4 的数据不会被刷新的。

- 4) 在自动刷新功能执行时占用的 CPU 共享内存覆盖设置 1 到 4 所有区域。
 当传送点数被设置，应用的 CPU 共享内存的起始和最终地址是十六进制显示的。
 传送点数设置在设置 1 和 2 的 CPU 用在设置 2 里的 CPU 共享内存的最终地址。
 (在如下例子中, CPU No.1 和 No.2 应用区域最多到 811H, 当 CPU No.4 应用区域最多到 821H。)传送点数设置只在设置 1 的 CPU 用在设置 1 里的 CPU 共享内存的最终地址。(在如下例子中, CPU No.3 在设置 1 里应用最终地址。)

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Start	End
No.1	16	0802	0811	W0	W0F
No.2	16	0802	0811	W10	W1F
No.3	0				
No.4	32	0802	0821	W20	W3F

The applicable device of head device is D,W,#,M,Y,B,*.
 The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

- 5) 在多 CPU 系统里, 为所有 CPU 设置相同的传送点数。
 如果任何一个 CPU 有一个不同的传送点数, 一个 PARAMETER ERROR 将会出现。

(c) CPU 侧软元件

下列软元件可以应用于自动刷新。(其它软元件不能在 SW6RN-GSV□P里设置)

可设置软元件	限制
数据寄存器 (D) 链接寄存器 (W) 运动寄存器 (#)	无
链接继电器 (B) 内部继电器 (M) 输出 (Y)	<ul style="list-style-type: none"> 指定 0 或 16 的倍数作为起始号。 一传送点占用 16 点数。

- 1) 对于 CPU 侧软元件, 从要设置的软元件号开始连续使用在一个设置范围里的为 CPU No.1 到 4 设置的发送总点数对应的软元件从软元件设置的成功启动时应用。
 设置软元件号, 以保证设置传送点数足够的软元件。
 当位软元件被指定位 CPU 侧软元件, 传送点数是 16 的倍数。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

2) 设置 CPU 侧软元件如下。

- 设置 1 到 4 可以使用不同的软元件。
如果软元件范围没有重叠, 对于设置 1 到 4 可能使用相同的软元件可以在设置 1 到 4 中被使用。

设置 1: 链接继电器

Refresh Setting
Setting 1

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Start	End
No.1	2	0800	0801	B0	B1F
No.2	2	0800	0801	B20	B3F
No.3	4	0800	0803	B40	B7F
No.4	2	0800	0801	B80	B9F

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

设置 2: 链接寄存器

Refresh Setting
Setting 2

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Start	End
No.1	16	0802	0811	w0	w0F
No.2	16	0802	0811	w10	w1F
No.3	0				
No.4	32	0802	0821	w20	w3F

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

设置 3: 链接继电器

Refresh Setting
Setting 3

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Start	End
No.1	2	0812	0813	B100	B11F
No.2	2	0812	0813	B120	B13F
No.3	4	0804	0807	B140	B17F
No.4	4	0822	0825	B180	B1BF

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

设置 1 到 4 可以使用不同软元件

对于设置 1 到 4 可以使用相同软元件
显示到左的设置 1, 从 B0 到 B9F 的 160 点数被使用的。
因此设置 3 可以使用 BA0 以后的软元件。软元件数偶数部分不可以被重叠, 如在设置 1 指定 B0 到 B9F, 在设置 3 指定 B90 到 B10F。

SW6RN-GSV □ P 里起始和最终地址被自动计算。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

- 各 CPU 在设置 1 到 4 的软元件可以单独设置的。例如，为 CPU No.1 可以设置链接继电器和为 CPU No.2 可以设置内部继电器。

CPU No.1 的刷新设置

Refresh Setting

Setting 1

Refresh Setting

Setting 2

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Dev. starting	End
No.1	16	0802	0811	W0	W0F
No.2	16	0802	0811	W10	W1F
No.3	0				
No.4	32	0802	0821	W20	W3F

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

当 CPU No.1 的 CPU 一侧
软元件与 CPU No.2 的不同时。

CPU No.2 的刷新设置

Refresh Setting

Setting 1

Refresh Setting

Setting 2

CPU	Send range for each CPU			CPU side device	
	Point (*)	Start	End	Dev. starting	End
No.1	16	0802	0811	W0	W0F
No.2	16	0802	0811	W10	W1F
No.3	0				
No.4	32	0802	0821	W20	W3F

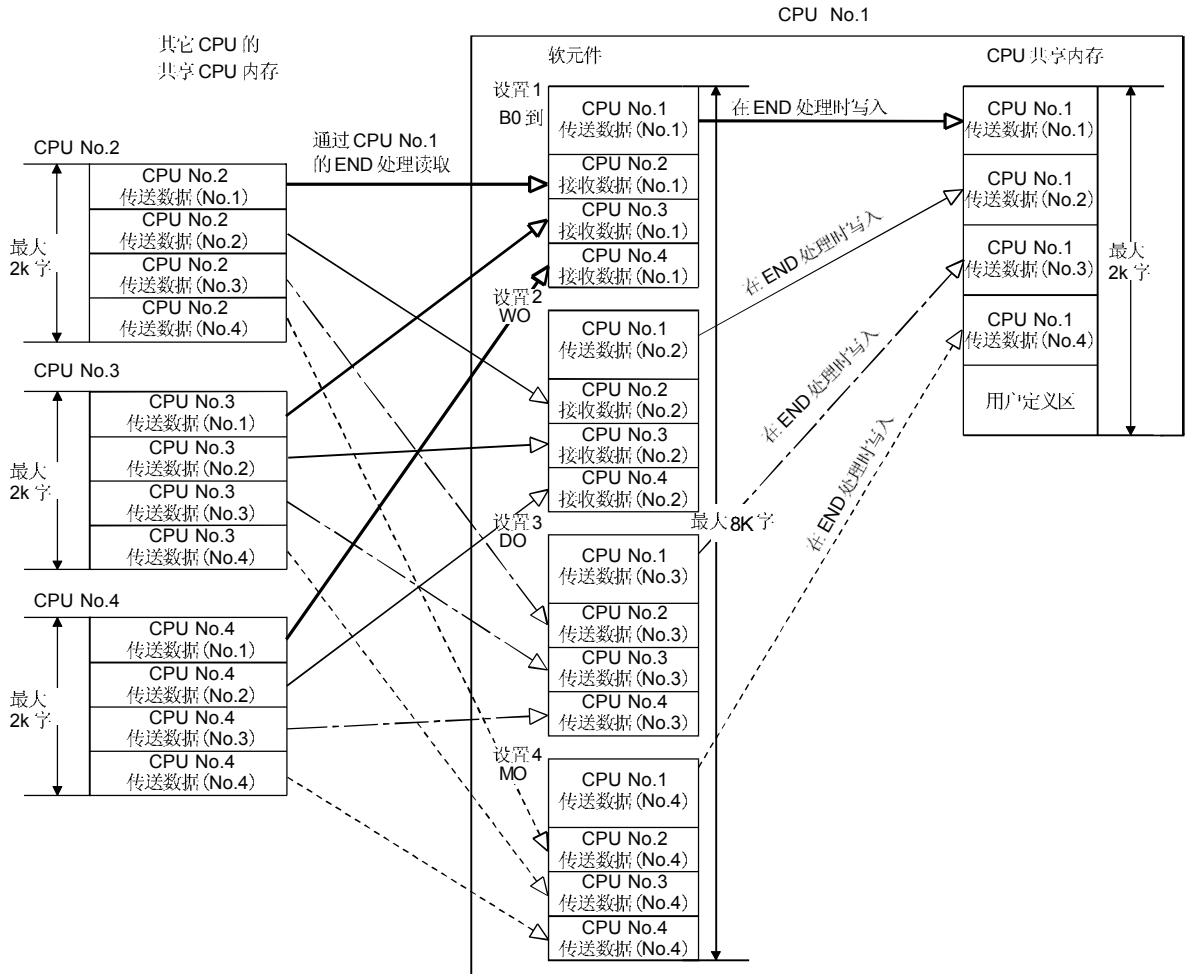
The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

当 CPU No.1 的 CPU 一侧
软元件与 CPU No.2 的相同时。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

- 3) 如下方块图显示自动刷新操作, 在这 4 个范围里, 设置 1: 链接继电器(B), 设置 2: 链接寄存器(W), 设置 3: 数据寄存器(D), 设置 4: 内部继电器(M)。



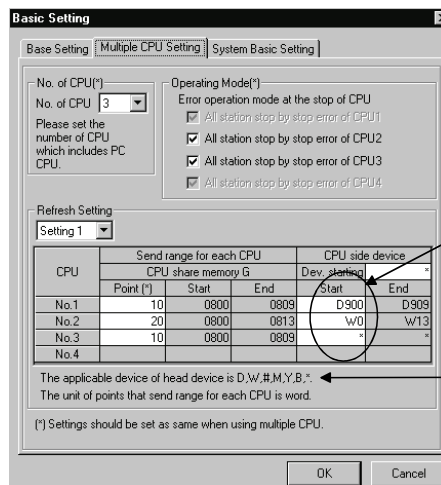
(3) 自动刷新设置 2(手动设置)

运动 CPU 的可应用版本和软件请参照 "1.3.4(5)" 部分。

(a) 当使用运动 CPU 的自动刷新设置(Manual setting)时。有如下优点。

- 1) 在 PLC CPU 和运动 CPU 之间的执行自动刷新设置的软元件设置可以随意执行的。
- 2) 因为系统不是用虚拟设置在运动 CPU 之间执行自动刷新设置，所以不需无谓地自动刷新用户软元件，主周期也可以缩短。
- 3) 可以直接执行运动软元件 (#) 的自动刷新到 PLC CPU。同样，可以直接执行 PLC CPU 数据的自动刷新到运动软元件 (#)。

关于 PLC CPU 的设置，请参照 "QCPU (Q 模式) 用户手册 (功能介绍 / 程序编程基础)"



起始软元件可以被每一个 CPU 任意设置。

除自 CPU 外 "DUMMY(*)" 可以设置为起始软元件

运动软元件 (#) 可以安装为起始软元件

(b) 各 CPU 的设置选择 / 发送范围(刷新范围)

- 1) 4 范围的刷新设置可以用设置选择来设置。例如，ON/OFF 数据可以用位软元件设置来刷新，而其它数据可以用字软元件设置来刷新。
- 2) 在共享 CPU 内存中，点数是以 2 点为单位。在各 CPU 的发送范围内设置的点数是 2 个字。(如果 CPU 侧软元件是字软元件就用 2 个点。如果是位软元件就用 32 个点)。
"0" 设置为 CPU 传送范围点数那么 CPU 数据是不被刷新的。
- 3) 每 CPU (PLC CPU 或运动 CPU)，组成所有 4 个范围的传送点数的最大值是 2k 字，或所有 CPU 的最大值是 8k 点数 (8k 字)
- 4) 如果 "*" 设置为起始软元件，用来设置各自动刷新设置的 A 栏，在 B 栏每个 CPU 的起始软元件可以由用户任意设置。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

- 5) “DUMMY” 设置可以设置到自动刷新设置的起始软元件 B 栏。(“DUMMY” 设置不可以设置到自 CPU。) “DUMMY” 设置必须设置 “*” 作为起始软元件 B 栏。自 CPU 不执行自动刷新到其它执行 “DUMMY(*)” 设置的 CPU 上。

Refresh Setting

Setting 1

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	*
	Point (*)	Start	End	Start	End
No.1	0				
No.2	0				
No.3	0				
No.4					

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

空白部分可以被设置

- 6) 在多 CPU 系统中为所有 CPU 设置相同的传送点数。
如果任何一个 CPU 有一个不同的传送点数，会产生一个 PARAMETER ERROR。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

(c) CPU 侧软元件

如下软元件可以用作自动刷新。(其它软元件可以在 SW6RN-GSV□P 里设置。)

可设置软元件	限制
数据寄存器 (D) 链接寄存器 (W) 运动寄存器 (#)	无
链接继电器 (B) 内部继电器 (M) 输出 (Y)	<ul style="list-style-type: none"> 指定 0 或 16 的倍数作为起始号。 一个传送点占用 16 点数。

自 CPU(CPU No.2)

刷新设置 1

Refresh Setting

Setting 1

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	20	0800	0813	D200	D219
No.2	30	0800	081D	D100	D129
No.3	10	0800	0809	W0	W9
No.4	50	0800	0831	*	*

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

如果软元件数字没有重叠,就是正确的。

在设置 1 的 CPU No.4 的软元件没有被 CPU No.2 刷新。

自 CPU(CPU No.2)

刷新设置 2

Refresh Setting

Setting 2

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	20	0814	0827	M480	M799
No.2	30	081E	083B	M0	M479
No.3	10	080A	0813	B0	B9F
No.4	50	0832	0863	*	*

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

如果软元件数字没有重叠,就是正确的。

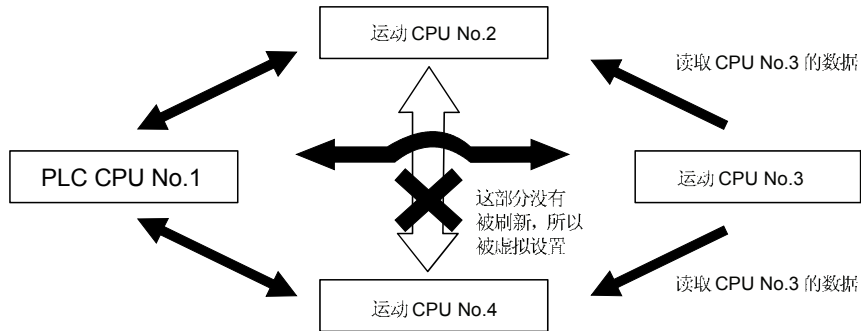
在设置 1 的 CPU No.4 的软元件没有被 CPU No.2 刷新。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

[虚拟设置(DUMMY SETTING)]

通常，在操作过程中，各运动 CPU 和 PLC CPU 的状态监视器的指令，在 PLC CPU 和运动 CPU 之间自动刷新设置被执行。然而，自动刷新在各运动 CPU 之间不能执行。在这种情况下因为不是制造成在运动 CPU 间用虚拟设置执行自动刷新设置的，为自动刷新，不需使用用户软元件，主周期也可以被缩短的。

用“虚拟设置”来自动刷新设置举例如下：



PLC CPU (CPU No.1)
自动刷新设置1

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	10	0800	0809	M0	M159
No.2	20	0800	0813	M160	M479
No.3	30	0800	081D	M480	M959
No.4	40	0800	0827	M960	M1539

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

运动 CPU (CPU No.3)
自动刷新设置1

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	10	0800	0809	M1024	M1183
No.2	20	0800	0813	*	*
No.3	30	0800	081D	B0	B1DF
No.4	40	0800	0827	*	*

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

CPU No.2 和 CPU No.4 的软件没有用 CPU No.3 刷新。

PLC CPU (CPU No.2)
自动刷新设置1

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	10	0800	0809	M1024	M1183
No.2	20	0800	0813	M0	M319
No.3	30	0800	081D	B0	B1DF
No.4	40	0800	0827	*	*

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

CPU CPU No.4 的软件没有用 CPU No.2 刷新。

运动 CPU (CPU No.4)
自动刷新设置1

CPU	Send range for each CPU			CPU side device	
	CPU share memory G			Dev. starting	
	Point (*)	Start	End	Start	End
No.1	10	0800	0809	M1024	M1183
No.2	20	0800	0813	*	*
No.3	30	0800	081D	B0	B1DF
No.4	40	0800	0827	M0	M639

The applicable device of head device is D,W,#,M,Y,B,*.
The unit of points that send range for each CPU is word.

(*) Settings should be set as same when using multiple CPU.

CPU No.2 的软件没有用 CPU No.4 刷新。

虽然设置的例子是自动刷新设置 1 的情况，自动刷新设置 2 — 4 也是类似设置的。

(4) 自动刷新设置的排版举例

当在带 PLC CPU 的运动 CPU 里读取/写入运动专用软元件时的自动刷新设置的排版举例如下。

(a) SV13

- 总结构

内部继电器的图表

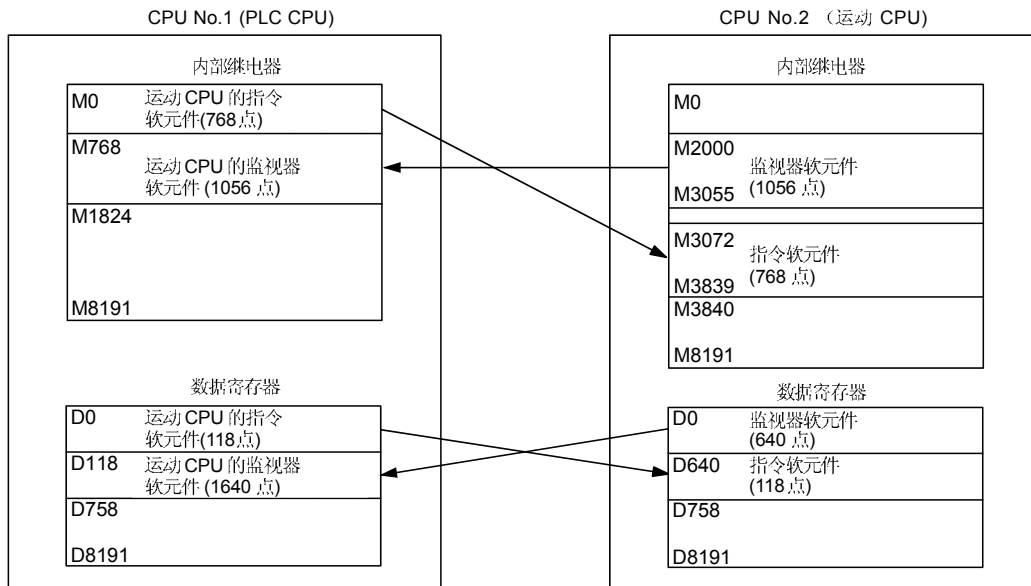
软元件编号	应用
M0 到	用户软元件 (2000 点)
M2000 到	公共软元件 (320 点)
M2320 到	特殊继电器分配软元件 (状态) (80 点)
M2400 到	轴状态 (20 点 × 32 轴)
M3040 到	不可用
M3072 到	公共软元件 (指令信号) (64 点)
M3136 到	特殊继电器分配软元件 (指令信号) (64 点)
M3200 到	轴指令信号 (20 点 × 32 轴)
M3840 到	用户软元件 (4352 点)
M8191	

数据寄存器的图表

软元件编号	应用
D0 到	各轴监视软元件 (20 点 × 32 轴)
D640 到	控制变更寄存器 (2 点 × 32 轴)
D704 到	公共软元件 (公共信号) (54 点)
D759 到	公共软元件 (监视器) (42 点)
D800 到	用户软元件 (7392 点)
D8191	

1) PLC CPU (1 个模块) + 运动 CPU (1 个模块)

操作概要和自动刷新设置如下。



• 自动刷新设置 1

PLC CPU (CPU No.1)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0
	点	启动	结束	启动	结束
No.1	48			M0	M767
No.2	66			M768	M1823
No.3					
No.4					

运动 CPU (CPU No.2)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	*
	点	启动	结束	启动	结束
No.1	48			M3072	M3839
No.2	66			M2000	M3055
No.3					
No.4					

• 自动刷新设置 2

PLC CPU (CPU No.1)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	D0
	点	启动	结束	Start	结束
No.1	118			D0	D117
No.2	640			D118	D757
No.3					
No.4					

运动 CPU (CPU No.2)

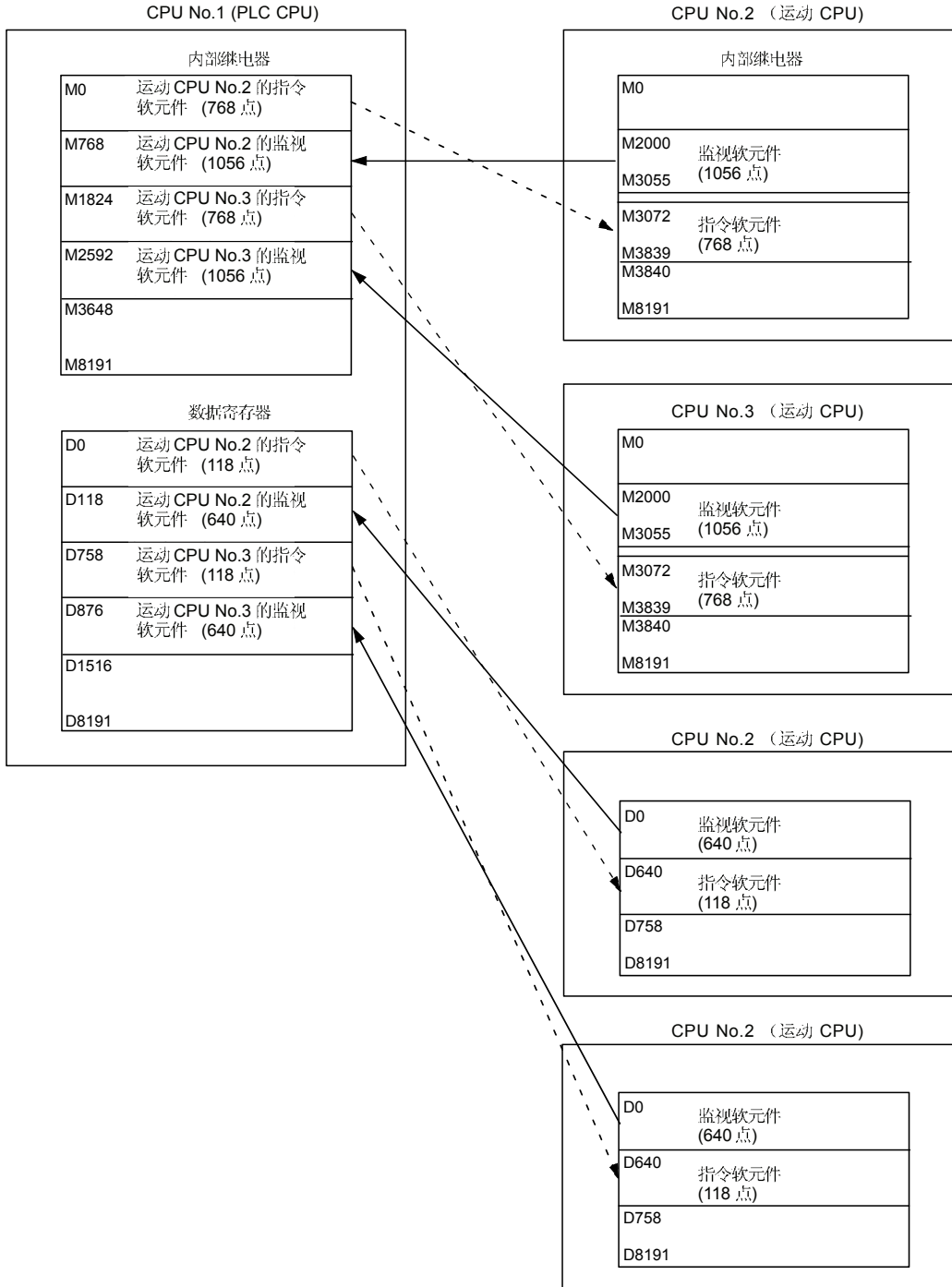
CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	*
	点	启动	结束	启动	结束
No.1	118			D640	D757
No.2	640			D0	D639
No.3					
No.4					

要点

虽然在以上分配例中，可设置 32 轴，但是为减少要使用的轴的数目，可减少自动刷新点数。

2) PLC CPU (1 模块) + 运动 CPU (2 模块)

操作概要和自动刷新设置如下。



3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

- 自动刷新设置 1

PLC CPU (CPU No.1)						运动 CPU (CPU No.2)					
CPU	各 CPU 的发送范围			CPU 侧软元件		CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0		CPU 共享内存 G			起始软元件	*
	点	启动	结束	启动	结束		点	启动	结束	启动	结束
No.1	48			M0	M767	No.1	48			M3072	M3839
No.2	66			M768	M1823	No.2	66			M2000	M3055
No.3	0					No.3	0				
No.4						No.4					

- 自动刷新设置 2

PLC CPU (CPU No.1)						运动 CPU (CPU No.2)					
CPU	各 CPU 的发送范围			CPU 侧软元件		CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0		CPU 共享内存 G			起始软元件	*
	点	启动	结束	启动	结束		点	启动	结束	启动	结束
No.1	118			D0	D117	No.1	118			D640	D757
No.2	640			D118	D757	No.2	640			D0	D639
No.3	0					No.3	0				
No.4						No.4					

- 自动刷新设置 3

PLC CPU (CPU No.1)						运动 CPU (CPU No.2)					
CPU	各 CPU 的发送范围			CPU 侧软元件		CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0		CPU 共享内存 G			起始软元件	*
	点	启动	结束	启动	结束		点	启动	结束	启动	结束
No.1	48			M1824	M2591	No.1	48			*	*
No.2	0			M2592	M3647	No.2	0			*	*
No.3	66					No.3	66				
No.4						No.4					

(注): 虚拟设置(dummy), 可以使 CPU No.2 的额外软元件不被刷新。

- 自动刷新设置 4

PLC CPU (CPU No.1)						运动 CPU (CPU No.2)					
CPU	各 CPU 的发送范围			CPU 侧软元件		CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0		CPU 共享内存 G			起始软元件	*
	点	启动	结束	启动	结束		点	启动	结束	启动	结束
No.1	118			D758	D875	No.1	118			*	*
No.2	0					No.2	0				
No.3	640			D876	D1515	No.3	640			*	*
No.4						No.4					

(注): 虚拟设置(dummy), 可以使 CPU No.2 的额外软元件不被刷新。

• 自动刷新设置 1
运动CPU (CPU No.3)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0
	点数	起始	结束	起始	结束
No.1	48			*	*
No.2	66			*	*
No.3	0				
No.4					

(注): 虚拟设置(dummy), 可以使 CPU No.3 的额外软元件不被刷新。

• 自动刷新设置 2
运动CPU (CPU No.3)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0
	点数	起始	结束	起始	结束
No.1	118			*	*
No.2	640			*	*
No.3	0				
No.4					

(注): 虚拟设置(dummy), 可以使 CPU No.2 的额外软元件不被刷新。

• 自动刷新设置 3
运动CPU (CPU No.3)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0
	点数	起始	结束	起始	结束
No.1	48			M3072	M3839
No.2	0				
No.3	66			M2000	M3055
No.4					

• 自动刷新设置 4
运动CPU (CPU No.3)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0
	点数	起始	结束	起始	结束
No.1	118			D640	D757
No.2	0				
No.3	640			D0	D639
No.4					

要点	在"PLC CPU (1 模块) + 运动 CPU (3 模块)"和 SV13 组合的情况下, 因为 PLC CPU 构成的读取/写入设置不能被执行使所有 CPU 的软元件按以上所述刷新。
----	--

(b) SV22

• 总结构

内部继电器的图表

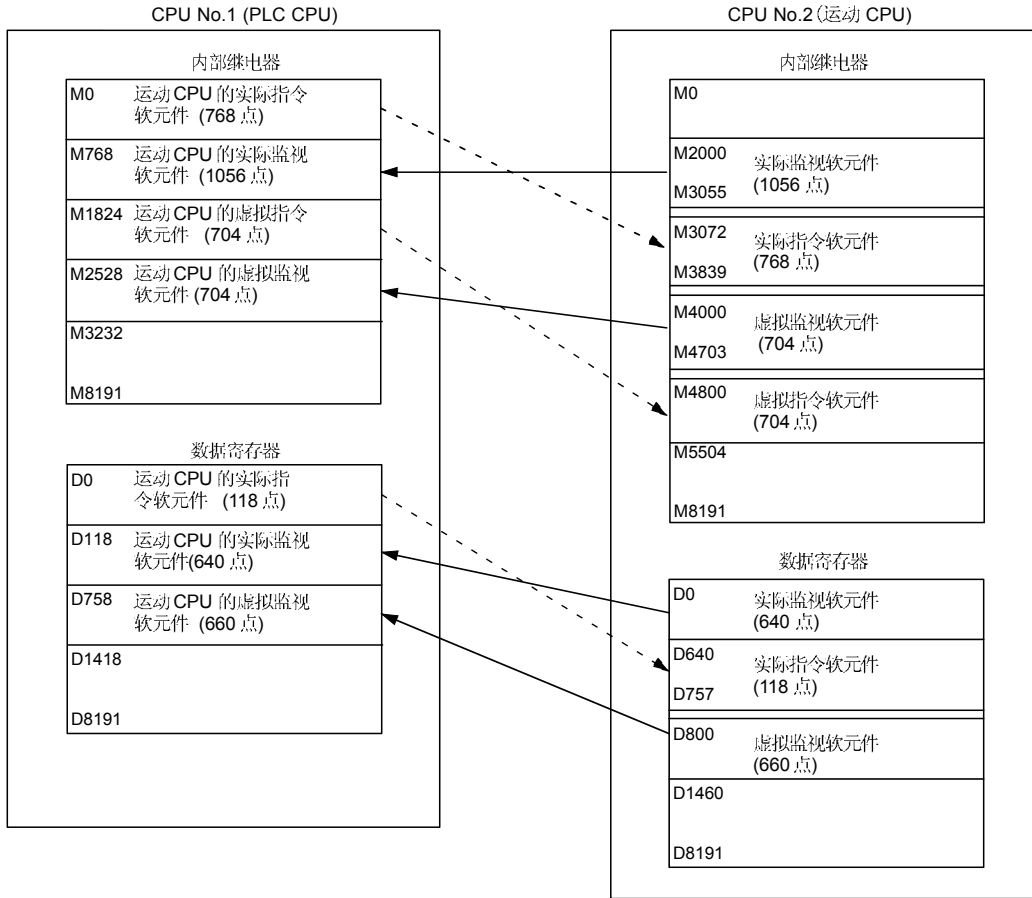
软元件编号	应用
M0 到	用户软元件 (2000 点)
M2000 到	公共软元件 (320 点)
M2320 到	特殊继电器分配软元件 (状态) (64 点)
M2400 到	轴状态 (20 点 × 32 轴)
M3040 到	不可用
M3072 到	公共软元件 (指令信号) (64 点)
M3136 到	特殊继电器分配软元件 (指令信号) (64 点)
M3200 到	轴指令信号 (20 点 × 32 轴)
M3840 到	不可用
M4000 到	虚拟伺服电机轴状态 (20 点 × 32 轴) (仅机械系统设置轴)
M4640 到	同步编码器轴状态 (4 点 × 12 轴)
M4688 到	不可用
M4800 到	虚拟伺服电机轴状态 (20 点 × 32 轴) (仅机械系统设置轴)
M5440 到	同步编码器轴状态 指令信号 (4 点 × 12 轴)
M5488 到	不可用
M5600 到 M8191	用户软元件) (2592 点)

数据寄存器的图表

软元件编号	应用
D0 到	各轴监视软元件 (20 点 × 32 轴)
D640 到	控制变更寄存器 (2 点 × 32 轴)
D704 到	公共软元件 (公共信号 I) (54 点)
D759 到	公共软元件 (监视器) (42 点)
D800 到	虚拟伺服电机轴监视软元件状态 (10 点 × 32 轴) (仅机械系统设置轴)
D1200 到	同步编码器监视软元件状态 (10 点 × 12 轴)
D1240 到	凸轮轴监视软元件 (10 点 × 32 轴)
D1560 到	用户软元件 (6632 点)
D8191	

2) PLC CPU (1 模块) + 运动 CPU (2 模块)

操作概要和自动刷新设置如下。



3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

• 自动刷新设置 1

PLC CPU (CPU No.1)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M0
	点数	起始	结束	起始	结束
No.1	48			M0	M767
No.2	66			M768	M1823
No.3	0				
No.4					

运动 CPU (CPU No.2)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	*
	点数	起始	结束	起始	结束
No.1	48			M3072	M3839
No.2	66			M2000	M3055
No.3	0				
No.4					

• 自动刷新设置 2

PLC CPU (CPU No.1)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	D0
	点数	起始	结束	起始	结束
No.1	118			D0	D117
No.2	640			D118	D757
No.3					
No.4					

运动 CPU (CPU No.2)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	*
	点数	起始	结束	起始	结束
No.1	118			D640	D757
No.2	640			D0	D639
No.3					
No.4					

• 自动刷新设置 3

PLC CPU (CPU No.1)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	M1824
	点数	起始	结束	起始	结束
No.1	44			M1824	M2527
No.2	44			M2528	M3231
No.3					
No.4					

运动 CPU (CPU No.2)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	*
	点数	起始	结束	起始	结束
No.1	44			M4800	M5503
No.2	44			M4000	M4703
No.3					
No.4					

• 自动刷新设置 4

PLC CPU (CPU No.1)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	D758
	点数	起始	结束	起始	结束
No.1	0				
No.2	660			D758	D1417
No.3					
No.4					

运动 CPU (CPU No.2)

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	*
	点数	起始	结束	起始	结束
No.1	0				
No.2	660			D800	D1459
No.3					
No.4					

要点

在"PLC CPU (1 模块) + 运动 CPU (2 模块)"和 SV22 组合的情况下, 因为 PLC CPU 构成的读取/写入设置不能被执行使所有 CPU 的软元件按以上描述刷新。

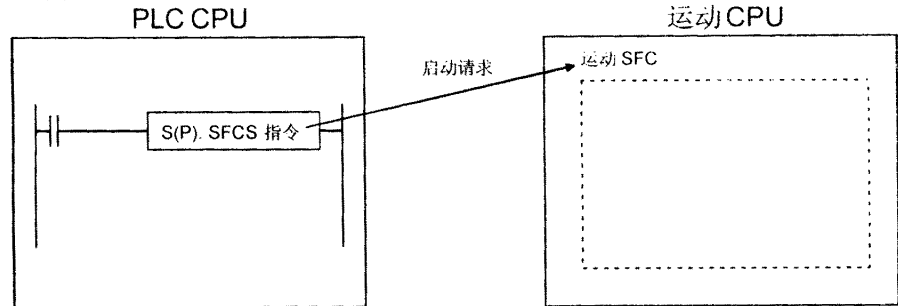
3.2 从 PLC CPU 到运动 CPU 的控制指令 (运动专用指令)

用如下表格中的运动专用 PLC 指令，可以从 PLC CPU 控制运动 CPU。
 更多关于各指令的详情，请参照第 5 章“专用 PLC 指令”。
 (从一个运动 CPU 到另一个运动 CPU，控制指令不能执行。)

指令名称	说明
S(P).SFCS	启动运动 SFC 程序的请求 (程序编号可以被指定的)。
S(P).SVST	启动指定伺服程序的请求。
S(P).CHGA	指定轴的当前值变更请求。
S(P).CHGV	指定轴的速度变更请求。
S(P).CHGT	指定轴的转矩控制值变更请求
S(P).GINT	其它 CPU (运动 CPU) 事件任务的执行请求。

用运动专用指令的 S(P). SFCS 指令，从 PLC CPU 到运动 CPU 的运动 SFC 可以被启动的。

<例子>



要点

一个 PLC CPU 可以同时执行最多 32 个“运动专用指令”和“不包括 S(P). GINT 的专用指令”。

当运动专用指令和“不包括 S(P). GINT”的专用指令被同时执行时，指令将按顺序被处理。如果没有被处理完的指令数超过 33 时，将会出现 OPERATION ERROR(错误代码:4107)。

3.3 读取/写入软元件数据

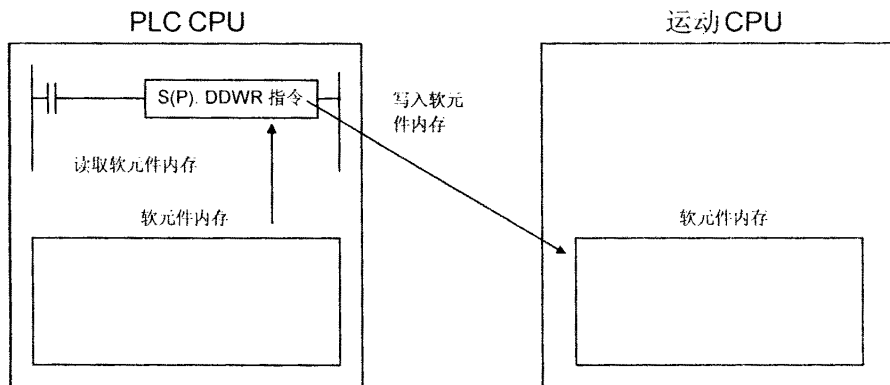
用下列表格中的专用指令，软元件数据可以由 PLC CPU 写入或读取到运动 CPU。更多各指令的详情，请参照第 5 章“运动专用 PLC 指令”

(用另一 PLC CPU，数据不能写入或读取到 PLC CPU；用运动 CPU，数据不能写入或读取到 PLC CPU；用另一运动 CPU，数据不能写入或读取到运动 CPU)

指令名称	说明
S(P).DDWR	自 CPU (PLC CPU)的软元件数据写入到其它 CPU (运动 CPU)的软元件。
S(P).DDR D	其它 CPU (运动 CPU)的软元件数据读取到自 CPU (PLC CPU)的软元件。

例如，用 S(P). DDWR 专用指令，PLC CPU 的软元件数据，可以写入到运动 CPU 的软元件数据。

<例子>



要点

- (1) 一个 PLC CPU 可以同时执行最多 32 个“运动专用指令”和不包括“S(P). GINT 的专用指令”。
当运动专用指令和不包括“S(P). GINT”的专用指令被同时执行时，指令将按顺序被处理。
如果没有被处理完的指令数超过 33 时，将会出现 OPERATION ERROR (错误代码: 4107)。
- (2) 通过 S(P). DDRD/S(P). DDWR 指令的数据刷新与通过 CPU 共享内存的自动刷新功能的数据刷新不同步。不要执行 S(P). DDRD/S(P). DDWR 指令执行数据在 CPU 共享内存刷新的软元件。

3.4 CPU 共享内存

在多 CPU 系统里的 CPU 之间，CPU 共享内存被用作传送数据，从 0H 到 FFFH 共 4096 字的容量。

CPU 共享内存有 4 个区域“自 CPU 操作数据区域”，“系统区域”，“自动刷新区域”，和“用户自定义区域”。

当 CPU 共享内存的自动刷新功能被设置时，从 800H 开始的自动刷新点数对应的区域将被用当作自动刷新区域。

用户自定义区域从自动刷新区域的最终地址开始。

如果自动刷新点数是 18(12H 点)，从 800H 到 811H 的区域变成自动刷新区域和 812H 后的区域变成用户定义区域。

如下图表显示 CPU 共享内存的构成和 PLC 程序的构成。

CPU 共享内存		自 CPU		其它 CPU		
		写入 ^(注-1)	读取	写入	读取 ^(注-2)	
0H	到 1FFH	自 CPU 操作数据区域	不允许	不允许	不允许	允许
200H			系统区域	不允许	不允许	不允许
7FFH	800H	自动刷新区域		不允许	不允许	不允许
到	到 FFFH	用户定义区域	允许	不允许	不允许	允许

备注

(注-1): 在 PLC CPU 里，用 S. TO 指令写入到自 CPU 的用户自定义区。
运动 CPU 没有 S.TO 指令，因此它不能写入到自 CPU 的用户自定义区域。

(注-2): 用 FROM 指令 / 智能功能模块软元件(U □ \G □)在 PLC CPU 里读取运动 CPU 的共享内存。
运动 CPU 中，用 MULTR 指令读取其它 CPU 的共享内存。

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

(1) 自 CPU 操作数据区域 (0H 到 1FFH)

(a) 在多 CPU 系统中, 自 CPU 的下列数据是保存起来的。

表格 3.1 内容表格储存在自 CPU 操作数据区域

共享内存地址	名称	说明	详细介绍 ^(注)	相应的特殊寄存器
0H	可用数据/不可用数据	“可用数据/不可用数据”标志	这部分一般用来检查数据是否储存在自 CPU 的自 CPU 操作数据区域(1H 到 1FH) • 0: 数据没有储存在自 CPU 操作数据区域。 • 1: 数据储存在自 CPU 操作数据区域	—
1H	诊断出错	诊断出错	当诊断时产生的异常号被储存为 BIN 代码。	D9008
2H	诊断出错 发生时间	诊断出错 发生时间	当异常号储存在 CPU 共享内存的 1H 地址时, 年份和月份被储存在 2 位 BCD 代码。	D9010
3H			当异常号储存在 CPU 共享内存的 1H 地址时, 日期和小时被储存在 2 位 BCD 代码。	D9011
4H			当异常号储存在 CPU 共享内存的 1H 地址时, 分钟和秒被储存在 2 位 BCD 代码。	D9012
5H	出错数据种类代码	出错数据种类代码	储存显示储存的公共出错数据的种类和独立出错数据的种类代码。	D9013
6H	出错数据	出错数据	储存与当诊断时产生的异常号对应的公共数据。	D9014
7H 到 1CH	没有使用	—	没有使用	—
1DH	开关状态	开关状态	储存 CPU 的开关状态。	D9200
1EH	LED 状态	CPU-LED 状态	储存 CPU LED 的位模式。	D9201
1FH	CPU 操作状态	CPU 操作状态	储存 CPU LED 的操作状态。	D9015

(注): 详情请参照适用特殊寄存器。

- (b) 自 CPU 操作数据区域每次被刷新时, 相应寄存器已被改变。然而, 刷新时间可能被延迟一个以上的主周期。
(在运动控制时, 利用空闲时间来进行更新。最大主周期时间: 几毫秒到几百毫秒)
- (c) 用 FROM 指令, 自 CPU 操作数据区域的数据可以被其它 CPU 的 PLC CPU 里读取。
然而, 在数据更新, 且存在延误时, 用已读数据作为监视对象。
- (d) 运动专用 PLC 指令(30H 到 33H)使用的自 CPU 操作数据区域。
CPU 的自 CPU 高速中断接受标志的全部状态被储存在如下地址里。

共享内存地址	名称	说明
30H(48)	CPU1 的自 CPU 高速中断接受标志	用来检查 CPU _n 的自 CPU 高速中断接受标志的区域是否可以被接受。 0: CPU 的自 CPU 高速中断接受标志可以被接受。 1: CPU 的自 CPU 高速中断接受标志不可以被接受。
31H(49)	CPU2 的自 CPU 高速中断接受标志	
32H(50)	CPU3 的自 CPU 高速中断接受标志	
33H(51)	CPU4 的自 CPU 高速中断接受标志	

3 在多 CPU 系统中 PLC CPU 和运动 CPU 之间的通讯

(2) 系统区域 (204H 到 20DH)

该区域用于被 PLC CPU/ 运动 CPU 的操作系统(OS)使用的区域。当执行专用多 CPU 通讯指令时, OS 使用这个区域。

- 被运动专用 PLC 指令(204H 到 20DH)使用的系统区域。
全部状态储存在如下所示

共享内存地址	名称	说明										
204H(516)	启动接受标志(轴 1 到 16)	启动应用标志从 1 到 32 轴逐位储存。 (作为位的实际设置 173CPU(N) : J1 到 J32/ Q172CPU(N) : J1 到 J8.) OFF : 启动接受标志可用 ON : 启动接受标志禁止										
		<table border="1"> <tr> <td colspan="2" style="text-align: right;">b15</td> <td colspan="2" style="text-align: right;">b1</td> <td style="text-align: right;">b0</td> </tr> <tr> <td>204H(516)地址</td> <td>J16</td> <td>.....</td> <td>J2</td> <td>J1</td> </tr> </table>	b15		b1		b0	204H(516)地址	J16	J2	J1
b15		b1		b0								
204H(516)地址	J16	J2	J1								
205H(517)	启动接受标志(轴 17 到 32)											
		<table border="1"> <tr> <td>205H(517)地址</td> <td>J32</td> <td>.....</td> <td colspan="2">J17</td> </tr> </table>	205H(517)地址	J32	J17						
205H(517)地址	J32	J17									
206H(518)	速度变更标志(轴 1 到 16)	速度变更标志由 1 到 32 轴逐位储存。 (作为位的实际设置 173CPU(N) : J1 到 J32/ Q172CPU(N) : J1 到 J8.) OFF : 启动接受可用 ON : 启动接受禁止										
		<table border="1"> <tr> <td colspan="2" style="text-align: right;">b15</td> <td colspan="2" style="text-align: right;">b1</td> <td style="text-align: right;">b0</td> </tr> <tr> <td>206H(518)地址</td> <td>J16</td> <td>.....</td> <td>J2</td> <td>J1</td> </tr> </table>	b15		b1		b0	206H(518)地址	J16	J2	J1
b15		b1		b0								
206H(518)地址	J16	J2	J1								
207H(519)	速度变更标志(轴 17 到 32)											
		<table border="1"> <tr> <td>207H(519)地址</td> <td>J32</td> <td>.....</td> <td colspan="2">J17</td> </tr> </table>	207H(519)地址	J32	J17						
207H(519)地址	J32	J17									
208H(520)	同步编码器当前值变更标志(轴 1 到 12)	同步编码器当前值变更标志由 1 到 32 轴逐位储存。 (作为位的实际设置 173CPU(N) : E1 到 E32/ Q172CPU(N) : E1 到 E8.) OFF : 启动接受可用 ON : 启动接受禁止										
		<table border="1"> <tr> <td colspan="2" style="text-align: right;">b15</td> <td colspan="2" style="text-align: right;">b1</td> <td style="text-align: right;">b0</td> </tr> <tr> <td>208H(520)地址</td> <td>E16</td> <td>.....</td> <td>E2</td> <td>E1</td> </tr> </table>	b15		b1		b0	208H(520)地址	E16	E2	E1
b15		b1		b0								
208H(520)地址	E16	E2	E1								
20CH(524)	凸轮轴一转当前值变更标志(轴 1 到 16)	凸轮轴一转当前值变更标志由 1 到 32 轴逐位储存。 (作为位的实际设置 Q 173CPU(N) : C 1 到 C 32/ Q172CPU(N) : C 1 到 C 8.) OFF : 启动接受可用 ON : 启动接受禁止										
		<table border="1"> <tr> <td colspan="2" style="text-align: right;">b15</td> <td colspan="2" style="text-align: right;">b1</td> <td style="text-align: right;">b0</td> </tr> <tr> <td>20CH(524)地址</td> <td>C16</td> <td>.....</td> <td>C2</td> <td>C1</td> </tr> </table>	b15		b1		b0	20CH(524)地址	C16	C2	C1
b15		b1		b0								
20CH(524)地址	C16	C2	C1								
20DH(525)	凸轮轴一转当前值变更标志(轴 17 到 32)											
		<table border="1"> <tr> <td>20DH(525)地址</td> <td>C32</td> <td>.....</td> <td colspan="2">C17</td> </tr> </table>	20DH(525)地址	C32	C17						
20DH(525)地址	C32	C17									

(3) 自动刷新区域

本区用于多 CPU 系统的自动刷新。本区域不能用 S. TO 指令写入或用 FROM 指令读取。

(4) 用户定义区

通过 FROM/S. TO 指令和 PLC CPU 的智能功能模块软元件，本区域用于多 CPU 系统中各 CPU 之间的通讯。(在各 CPU 之间的通讯用运动 CPU 里的操作控制程序的 MULTR 指令和 MULTW 指令)

有关 MULTR 指令或 MULTW 指令参见 7.13.5 到 7.13.6 节

在自动刷新区域里的点数被设置后该区域才能被使用。

(如果自动刷新功能没有被执行，从 800H 到 FFFH 的区域可以用作用户定义区域。)

4. 运动 CPU 程序的构成

在运动 SFC 中运动 CPU 程序是以流程图的格式创建的。在实模式 SV13/SV22 的运动 SFC 程序里的运动控制步指定的实模式伺服程序中，执行伺服电机的运动控制。用由运动控制步指定的虚模式伺服程序，控制机械系统里的虚拟伺服电机，使在 SV22 虚模式里的同步控制允许。

更多关于运动 SFC 程序的详细资料，实模式的运动控制，虚模式的运动控制如下：

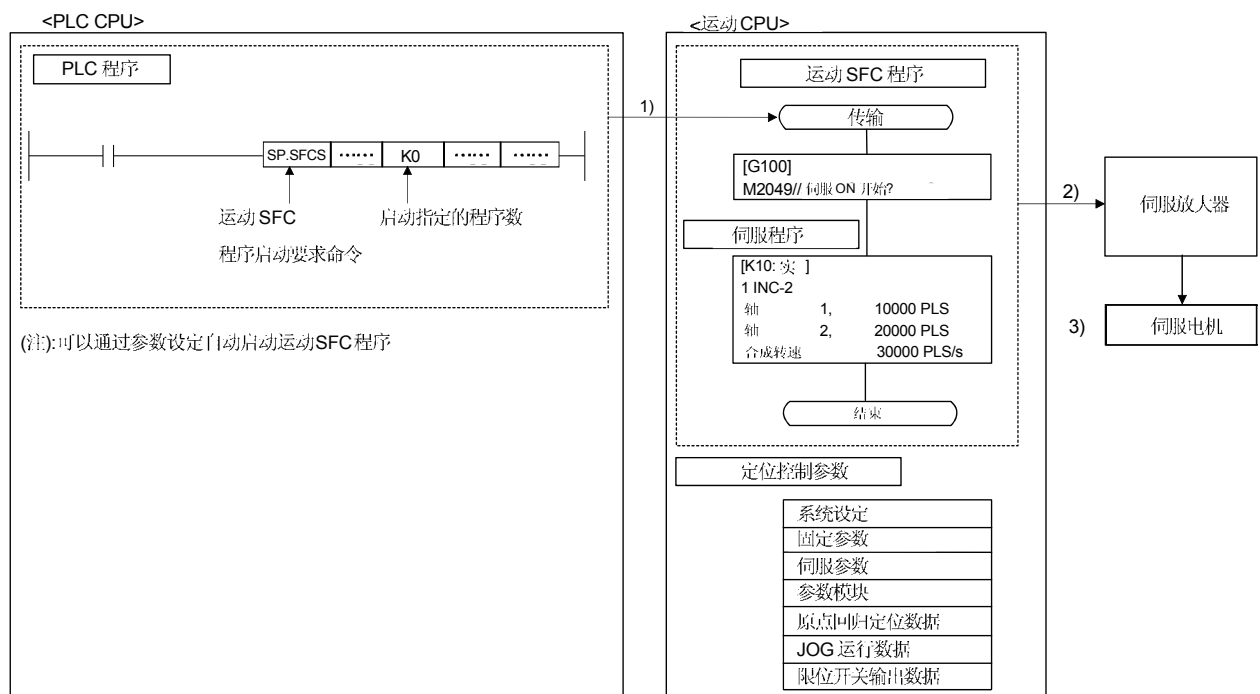
项目	参考手册
运动 SFC 程序	本文档的第 6 部分
在 SV13/SV22 实模式里的运动控制 (伺服程序)	Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册 (实模式)
在 SV22 虚模式里的运动控制 (机械系统程序)	Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22) 编程手册(虚模式)

4 运动 CPU 程序的构成

4.1 SV13/SV22 实模式下的运动控制

- (a) 在 SV13/SV22 实模式里，用伺服程序直接控制伺服电机系统。
- (b) 必须设置定位参数和创建伺服程序 / 运动 SFC 程序。
- (c) 定位控制 的程序如下所示：
 - 1) 运动 SFC 程序要求用 PLC 程序的 S(P).SFCS 指令启动。
(运动 SFC 程序也可以通过参数设置来自动启动。)
 - ↓
 - 2) 执行指定的运动 SFC 程序执行定位控制。(输出到伺服放大器)
 - ↓
 - 3) 控制伺服电机。

SV13/SV22实模式程序结构

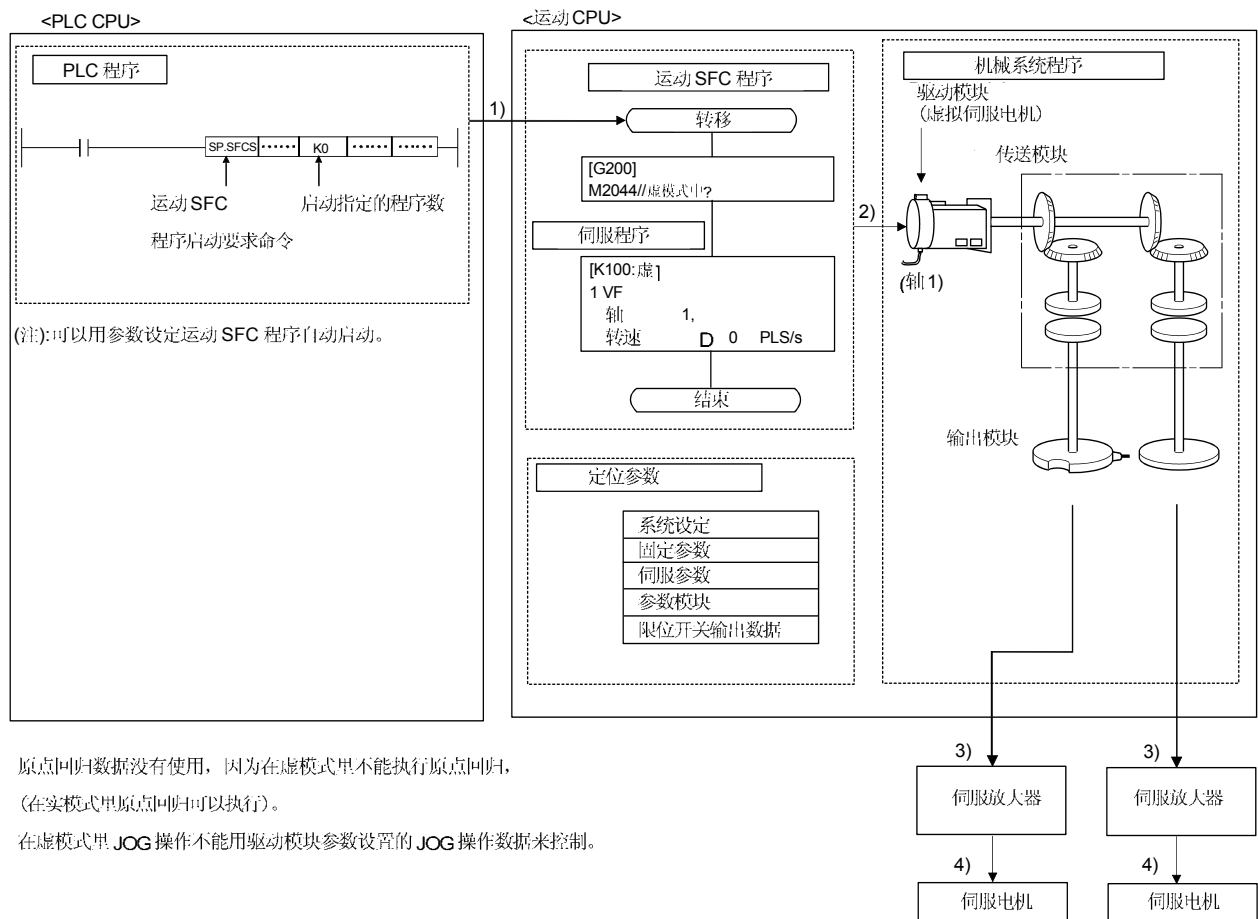


4 运动 CPU 程序的构成

4.2 SV22 虚模式下的运动控制

- (a) 用在(SV22)虚模式下的由虚拟主轴和机械模块组成的机械系统程序中，执行基于软件的同步控制。
- (b) 除了实模式里应用的定位参数，伺服程序/运动 SFC 程序，必须创建机械系统程序。
- (c) 在虚模式里的定位控制的程序：
 - 1) 用 PLC 程序的 S(P).SFCS 指令时，必须创建虚模式的运动 SFC。
(用参数设置，运动 SFC 程序也可以自动启动。)
 - ↓
 - 2) 启动机械系统程序里的虚拟伺服电机。
 - ↓
 - 3) 输出从传送模块获得的操作结果到伺服放大器，将其设置成输出模块组。
 - ↓
 - 4) 控制伺服电机。

SV22 虚拟模式程序构成



5. 运动专用 PLC 指令

5.1 运动专用 PLC 指令

- (1) 可针对装有运动 SFC 的 SV13/SV22 操作系统的运动 CPU 使用运动专用 PLC 指令，详情见下图。

指令	说明
S(P).SFCS	指定运动 SFC 程序的启动请求。
S(P).SVST	指定伺服程序的启动请求。
S(P).CHGA	指定轴的当前值变更请求。
S(P).CHGV	指定轴的速度变更请求。
S(P).CHGT	指定轴的转矩限制值变更请求。
S(P).DDWR	向运动 CPU 软元件写入 PLC CPU 软元件数据。
S(P).DDRDR	向 PLC CPU 软元件读入运动 CPU 软元件数据。
S(P).GINT	运动 SFC 程序的事件任务执行请求。

(注)：每个指令的详细说明，参见下节之后的内容。

5.1.1 运动专用 PLC 指令的限制事项

- (1) 来自 CPU_n 的自 CPU 高速中断接受标志。运动专用 PLC 指令的常见注意事项见下文。

- (a) 来自 CPU_n 的自 CPU 高速中断接受标志显示在如下表格中。

来自 CPU_n 的自 CPU 高速中断接受标志是“无操作”，即当不可接受时指令被执行。

当在运动 CPU 里，运动专用 PLC 指令是可接受的，CPU 中共享内存的自 CPU (运动 CPU) 的 CPU 高速中断接受标志不可接受的，对于请求指令的处理。

当处理完成，变成一个指令可接受的状态，来自 CPU_n 的自 CPU 高速中断接受标志可以被接受。

5 运动专用 PLC 指令

共享 CPU 内存地址 ()是十进制地址	说明	读取举例 (当目标是 CPU No.2)
30H(48)	CPU No.1 执行指令的最低等级位 (30H(48))。	U3E1/G48.0
31H(49)	CPU No.2 执行指令的最低等级位 (31H(49))。	U3E1/G49.0
32H(50)	CPU No.3 执行指令的最低等级位 (32H(50)) 。	U3E1/G50.0
33H(51)	CPU No.4 执行指令的最低等级位 (33H(51)) 。	U3E1/G51.0

- (b) 当多 CPU 专用指令从一个 PLC CPU 执行到相同 CPU 时，“来自 CPU_n 的自 CPU 高速中断接受标志”在执行指令时设置到 ON/OFF。
因此，当各指令在执行状态时仅执行一次，除“来自 CPU_n 的自 CPU 高速中断接受标志”，必须用内部继电器(M10)等来用于互锁。

(2) 运动专用 PLC 指令的执行

- (a) 运动专用 PLC 指令在固定周期执行类型 PLC 以及中断 PLC 中可执行。但是，作为完全软元件，程序是根据固定周期执行类型 PLC 开始执行的，而执行程序类型的中断 PLC 是与之不同的。
- (b) 一个运动 CPU 最多可同时接受来自其他多 CPU 的 32 条指令。
(除了 S(P).GINT 指令。)如果执行 33 条以上的指令，则运动 CPU 恢复完成状态[4C08]故障。
运动 CPU 可接受 32 条以下的指令，但具体可接受的指令的数目是根据运动 CPU 所包含的其他 CPU 的数量而改变的。计算公式见下表。

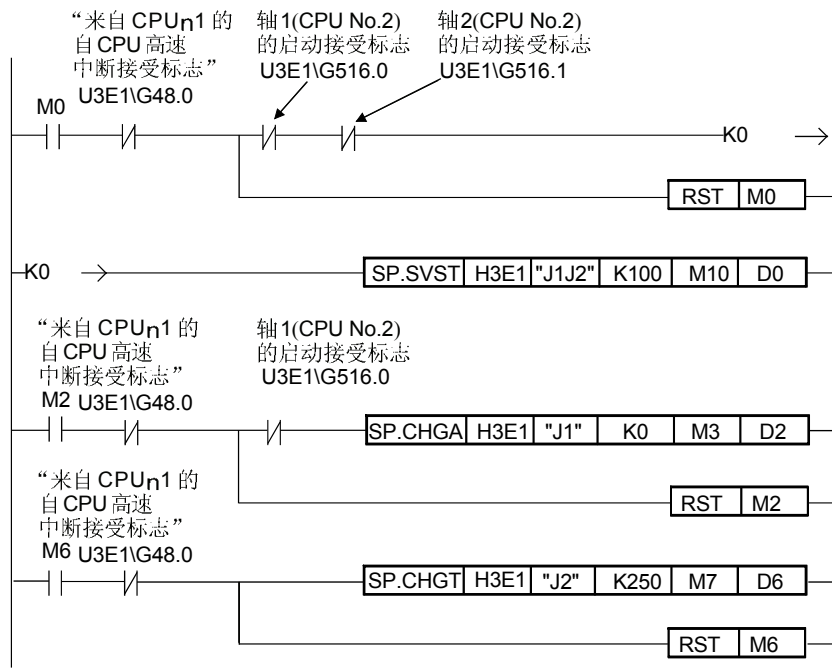
$$(\text{一个运动 CPU 最多可接受的指令的数目}) = 32 - ((\text{所有 CPU 的数目}) - 2)[\text{指令数}]$$

- (c) 局部软元件以及文件寄存器作为程序，通过 END 处理写入软元件。不要使用以下的软元件。
- 每个指令完成软元件
 - S(P).DDRD 指令的 D1 (存储读取数据的自 CPU 的起始软元件。)

- (d) 在共享 CPU 内存中使用标志。该内存与各指令(对于互锁条件下相同 CPU 号的运动 CPU 的相同轴不执行多指令)对应 (程序举例 1)。
- (e) S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGVS(P).CHGT/S(P).DDWR/S(P).DDRD 等指令不可同时执行。因此，必须通过“CPU 中的自 CPU 高速中断接受标志”执行互锁。
一个 PLC CPU 用“来自 CPU_n 的自 CPU 高速中断接受标志”，最多可被同时执行 32 条运动专用 PLC 指令。(除了 S(P).GINT 指令。)如果执行 33 条或超过 33 条的指令，则 PLC CPU 回到 OPERATION ERROR [4107]。
- (f) 当因接触点开启，多运动专用 PLC 指令被直接执行时，指令可能不被执行的。在这种情况下，参考编程举例创建程序。
- (g) 当操作控制步(Fn/FSn)的运动专用功能和在运动 CPU 里的运动控制程序(Kn)。在 PLC CPU 里的指令执行上，没有可以区分的标志，必须通过用户程序进行互锁。

<编程举例 1>

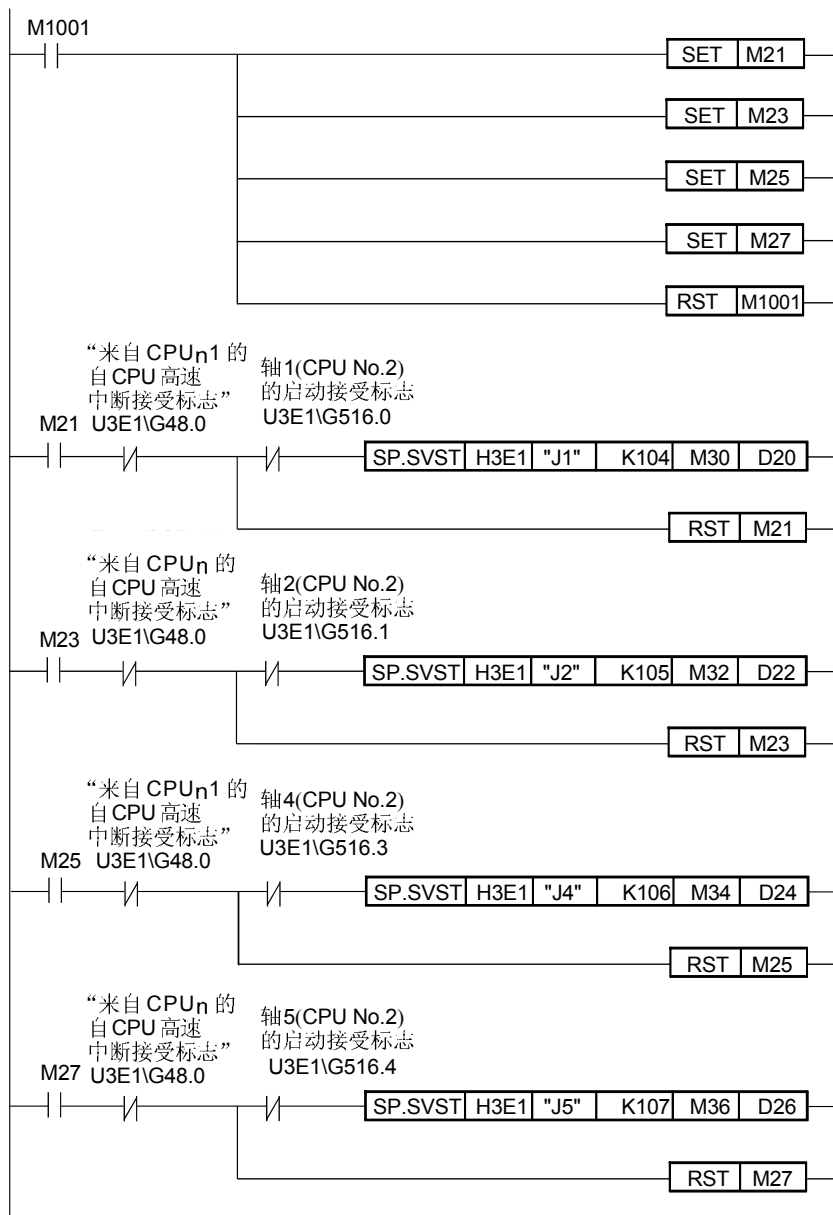
执行多指令到相同 CPU No. 的运动 CPU 的相同轴上的程序



5 运动专用 PLC 指令

<编程举例 2>

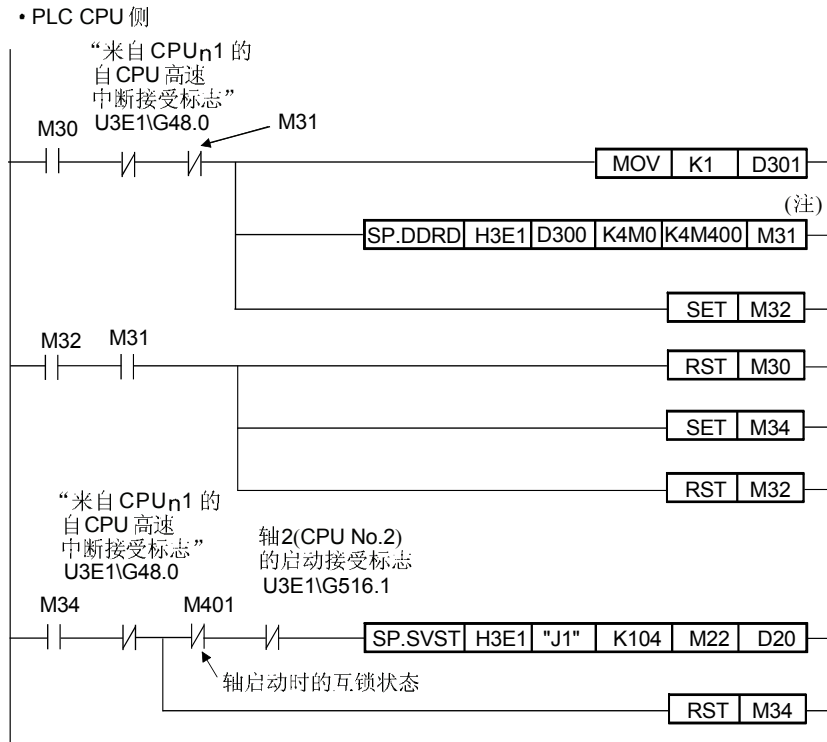
当常开触点开启,直接执行多运动专用 PLC 指令的程序



5 运动专用 PLC 指令

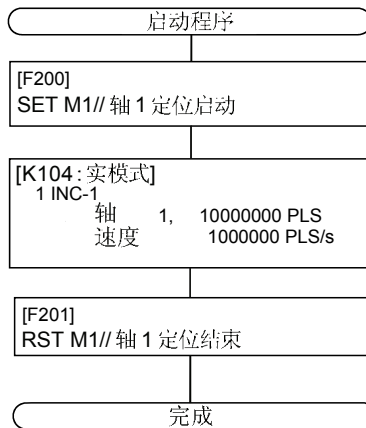
<编程举例 3>

执行操作控制步(Fn/FSn)和运动控制步(Kn)的运动专用功能的程序



(注):执行 S(P).DDR D 指令后, CPU No.2 的 M400 之后 4 点数据存储在 M0 之后 4 点数据。

• 运动 CPU 一侧



要点

在运动 CPU 通讯处理之前,可以访问 PLC CPU。然而,如果在 PLC CPU 内经常执行运动专用 PLC 指令,PLC CPU 的扫描时间不但会被延长而且在运动 CPU 的通讯处理时,延迟时间也会被延长。

必要时,通过 S(P).DDWR/S(P).DDR D/S(P).CHGV 等指令来执行 PLC CPU 的运动专用 PLC 指令。

5 运动专用 PLC 指令

(3) 结束状态

当 CPU 专用指令的异常结束时出错代码会储存在结束状态中。储存的出错代码如下显示。(标有“*”的出错代码是运动 CPU 专用的。)

结束状态 (出错代码)(H)	出错原因	处置方法
0	正常结束	确认一个程序，并且修正到正确的 PLC 程序里。
4C00 *	指定软元件不能应用于运动 CPU，它在软元件范围之外。	
4C01 *	执行运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C02 *	运动 SFC 程序号在 0 到 255 范围之外。	
4C03 *	执行的伺服程序号在 0 到 4095 范围之外。	
4C04 *	用 SVST 指令设置的轴号不正确。	
4C05 *	用 CHGA 指令设置的轴号不正确。	
4C06 *	用 CHGV 指令设置的轴号不正确。	
4C07 *	用 CHGT 指令设置的轴号不正确。	
4C08 *	<ul style="list-style-type: none"> 当执行 S(P).SFCS/S(P).SVST/S(P).CHGA 指令。 当执行 S(P).DDRD/S(P).DDWR 指令。 在 S(P).DDRD/S(P).DDWR 总数表格里，PLC CPU 的运动 CPU 同时有 32 以上指令请求。运动 CPU 不能处理它们。	
4C09 *	指令原因的 CPU 号不正确。	
4C0A *	数据出错（指定不能在运动 CPU 里解码的指令。）	
4C80	目标 CPU 的硬件出错	
4C81		
4C83		
4C84		
4C90	目标 CPU 的执行指令的数量。 在 S(P).SFCS, S(P).SVST, S(P).CHGA, S(P).SHGV, S(P).CHGT, S(P).DDRD 和 S(P).DDWD 总数表格里，PLC CPU 的运动 CPU 同时有 32 以上指令请求。运动 CPU 不能处理它们。	

(4) 被运动专用指令(30H 到 33H)使用的自 CPU 操作数据区域。

来自 CPU_n 的自 CPU 高速中断接受标志的结束状态储存在如下地址。

CPU 共享 内存地址	名称	说明
30H(48)	来自 CPU 1 的自 CPU 高速中断接受标志	该区域用来检查来自 CPU _n 的自 CPU 高速中断接受标志是否被接受。 0: 来自 CPU _n 的自 CPU 高速中断接受标志可用。 1: 来自 CPU _n 的自 CPU 高速中断接受标志禁止。
31H(49)	来自 CPU 2 的自 CPU 高速中断接受标志	
32H(50)	来自 CPU 3 的自 CPU 高速中断接受标志	
33H(51)	来自 CPU 4 的自 CPU 高速中断接受标志	

5 运动专用 PLC 指令

- (5) 被运动专用指令 (204H 到 20DH)使用的系统区域。
各标志的结束状态储存在如下地址。

CPU 共享 内存地址	名称	说明										
204H(516)	启动接受标志 (轴 1 到 16)	1 到 32 轴的启动接受标志被逐位储存。 (一个位的实际设置 Q173CPU(N) : J1 到 J32/ Q172CPU(N) : J1 到 J8.) OFF:启动接受可用 ON :启动接受不可用										
205H(517)	启动接受标志 (轴 17 到 32)	<table border="1" style="margin-left: 20px;"> <tr> <td style="width: 100px;">204H(516)地址</td> <td style="width: 100px;">b15 J16</td> <td style="width: 100px;">.....</td> <td style="width: 20px;">b1 J2</td> <td style="width: 20px;">b0 J1</td> </tr> <tr> <td>204H(517)地址</td> <td>C32</td> <td>.....</td> <td></td> <td>J17</td> </tr> </table>	204H(516)地址	b15 J16	b1 J2	b0 J1	204H(517)地址	C32		J17
204H(516)地址	b15 J16	b1 J2	b0 J1								
204H(517)地址	C32		J17								
206H(518)	速度变更标志 (轴 1 到 16)	1 到 32 轴的速度变更标志被逐位储存。 (一个位的实际设置 Q173CPU(N) : J1 到 J32/ Q172CPU(N) : J1 到 J8.) OFF:启动接受可用 ON :启动接受不可用										
207H(519)	速度变更标志 (轴 17 到 32)	<table border="1" style="margin-left: 20px;"> <tr> <td style="width: 100px;">204H(518)地址</td> <td style="width: 100px;">.....</td> <td style="width: 20px;">b1 J2</td> <td style="width: 20px;">b0 J1</td> </tr> <tr> <td>205H(519)地址</td> <td>.....</td> <td></td> <td>J17</td> </tr> </table>	204H(518)地址	b1 J2	b0 J1	205H(519)地址		J17		
204H(518)地址	b1 J2	b0 J1									
205H(519)地址		J17									
208H(520)	同步编码器当前值变更标志 (轴 1 到 12)	1 到 16 轴的同步编码器当前值变更标志被逐位储存。 (一个位的实际设置 Q173CPU(N) : E1 到 E12/ Q172CPU(N) : E1 到 E8.) OFF:启动接受可用 ON :启动接受不可用										
		<table border="1" style="margin-left: 20px;"> <tr> <td style="width: 100px;">208H(520)地址^{3S}</td> <td style="width: 100px;">J32</td> <td style="width: 100px;">.....</td> <td style="width: 20px;">b1 E2</td> <td style="width: 20px;">b0 E1</td> </tr> <tr> <td></td> <td></td> <td>.....</td> <td></td> <td>J17</td> </tr> </table>	208H(520)地址 ^{3S}	J32	b1 E2	b0 E1				J17
208H(520)地址 ^{3S}	J32	b1 E2	b0 E1								
			J17								
20CH(524)	凸轮轴一转中当前值变更标志 (轴 1 到 16)	1 到 32 轴的凸轮轴一转中当前值变更标志被逐位储存。 (一个位的实际设置 Q173CPU(N) : C1 到 C32/ Q172CPU(N) : C1 到 C8.) OFF:启动接受可用 ON :启动接受不可用										
20DH(525)	凸轮轴一转中当前值变更标志 (轴 17 到 32)	<table border="1" style="margin-left: 20px;"> <tr> <td style="width: 100px;">20CH(524)地址</td> <td style="width: 100px;">b15 C16</td> <td style="width: 100px;">.....</td> <td style="width: 20px;">b1 C2</td> <td style="width: 20px;">b0 C1</td> </tr> <tr> <td>20DH(525)地址</td> <td>C32</td> <td>.....</td> <td></td> <td>C17</td> </tr> </table>	20CH(524)地址	b15 C16	b1 C2	b0 C1	20DH(525)地址	C32		C17
20CH(524)地址	b15 C16	b1 C2	b0 C1								
20DH(525)地址	C32		C17								

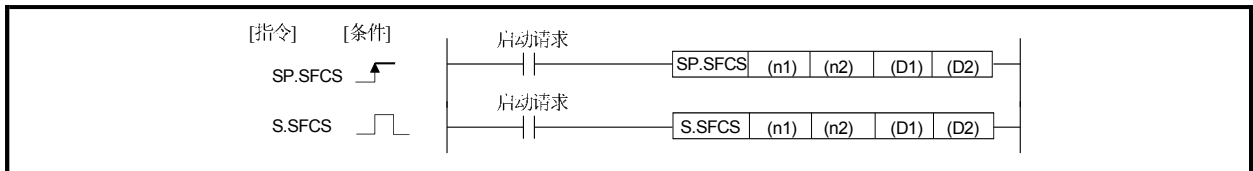
5 运动专用 PLC 指令

5.2 从 PLC CPU 到运动 CPU 的运动 SFC 启动请求: S(P).SFCS (PLC 指令: S(P).SFCS)

- 从 PLC CPU 到运动 CPU(S(P).SFCS)的运动 SFC 启动请求指令

设置数据 ④①②③	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接 指定 软元件	MELSECNET/10 直接 J□\□		特殊功能 模块 U□\G□	变址 寄存器 Z□	常数 K, H	其它
	位	字				位	字				
(n1)		○			○					○	
(n2)		○			○					○	
(D1)	○				○						
(D2)		○									

○: 可用 △: 部分可用
(注意): 设置数据 (n1)到 (D2): 可索引



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 实际指定值如下。 ^(注-1) CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H	16 位 二进制
(n2)	运动 SFC 程序号的启动	16 位 二进制
(D1)	完成软元件 (D1+0): 在指令启动接受结束时, 启动进行扫描的软元件 (D1+1): 在指令启动接受异常结束时, 启动进行扫描的软元件 (“D1+0” 在异常结束时也会开启。)	位
(D2)	储存结束状态的软元件	16 位 二进制

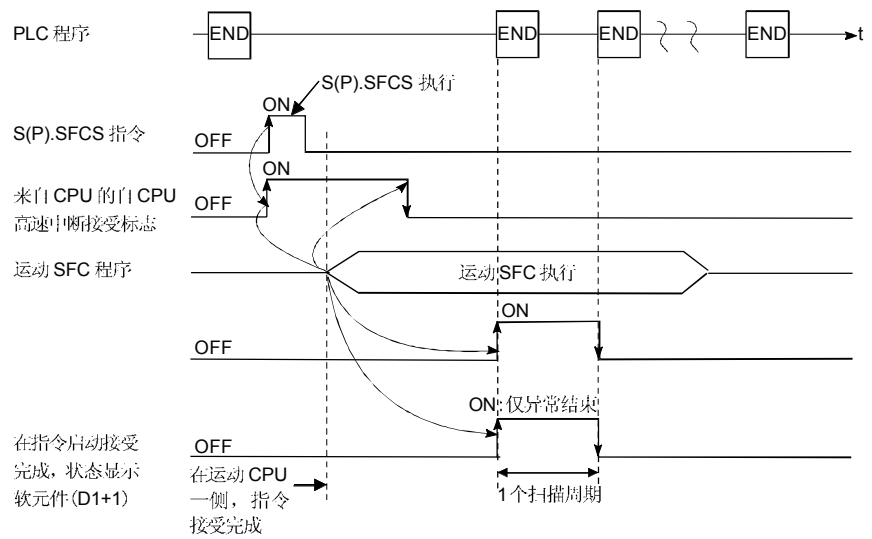
(注-1): 在多 CPU 构成中, 运动 CPU 不能使用 CPU No.1。

- (1) 对于在多 CPU 系统里的运动 CPU, 本指令是专用指令。
除运动 CPU 外, 在 CPU 中执行时, 会出现错误。

5 运动专用 PLC 指令

- (2) 用(n2)指定的程序号的运动 SFC 程序的启动请求。运动 SFC 程序可以启动任何正常任务，事件任务和 NMI 任务。
- (3) 当运动 CPU 的操作系统软件是 SV22 时，不管是在实模式/虚模式/模式切换时，本指令总是有效的。
- (4) 执行 S(P).SFCS 指令时，在 CPU 中 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDR/ S(P).DDWR 不能同时被执行。
当运动专用 PLC 指令持续开启时，必须在执行指令的完成软件元件开启后执行下一个指令。

[在 S(P).SFCS 指令执行时的自 CPU 操作]



5 运动专用 PLC 指令

[出错]

在如下情况下的异常结束，出错代码储存在结束状态指定的软元件(D2)中。

结束状态 (出错代码)(H) ^(注)	原因	处置方法
4C00	指定软元件不能应用于运动 CPU，它在软元件范围之外。	确认一个程序， 并且修正到正 确的 PLC 程序 里。
4C01	执行了与运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C02	运动 SFC 程序号在 0 到 255 范围之外。	
4C08	由 PLC CPU 到运动 CPU 的指令 S(P).SFCF/S(P).SVST， S(P).CHGA，S(P).GINT 总数超过了 32 个。此时运动 CPU 不 能处理这些指令。	
4C09	指令原因的 CPU 号不正确。	

(注)：0000H (正常)

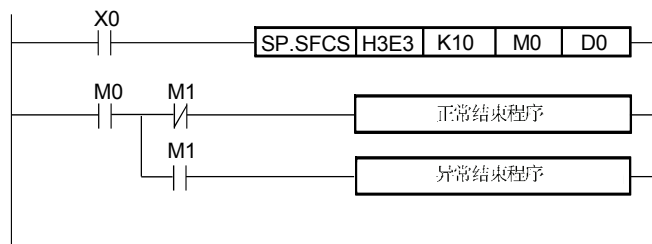
在如下情况下，出错标志(SM0)启动操作出错，出错代码储存在 SD0 里。

出错代码 ^(注)	原因	处置方法
2110	指定用“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序， 并且修正到正 确的 PLC 程序 里。
2114	指定用“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU，指定用“(目标 CPU 的起始 I/O 号)/16”设置的 自 CPU。	
4002	指定的指令出错。	
4004	指令由除可用软元件的其它软元件组成	
4100	指定用“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH， 3E4H。	

(注)：0000H (正常)

[编程举例]

本程序启动运动 CPU No.4 的 SFC 程序 No.10



5 运动专用 PLC 指令

5.3 从 PLC CPU 到运动 CPU 的伺服程序启动请求：

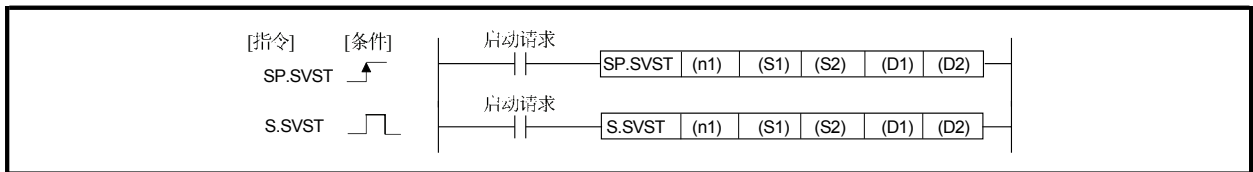
S(P).SVST (PLC 指令: S(P).SVST)

运动 CPU 和软件的可应用版本, 请参照“1.3.4”部分。

• 从 PLC CPU 到运动 CPU (S(P).SVST) 的伺服程序启动请求指令

设置数据 (注)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接 指定 软元件	MELSECNET/10 直接 J□\□		特殊功能 模块 U□\G□	变址 寄存器 Z□	常数 K, H	其它
	位	字				位	字				
(n1)		○			○					○	
(S1)											○
((S2))		○			○					○	
(D1)	○										
(D2)		○									

○: 可用 △: 部分可用
(注意): 设置数据 (n1) 到 (D2): 可索引



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 实际指定值如下。(注-1) CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H	16-位 二进制
(S1)	启动的轴号("Jn") (注-2)。	字顺序
(S2)	启动的伺服程序号。	16-位 二进制
(D1)	全部软元件 (D1+0): 在指令启动接受结束时, 启动扫描的软元件 (D1+1): 在指令启动接受异常结束时, 启动扫描的软元件 ("D1+0" 在异常结束时也会启动。)	位
(D2)	储存结束状态的软元件	16-位 二进制

(注-1): 在多 CPU 构成中, 运动 CPU 不能使用 CPU No.1 。

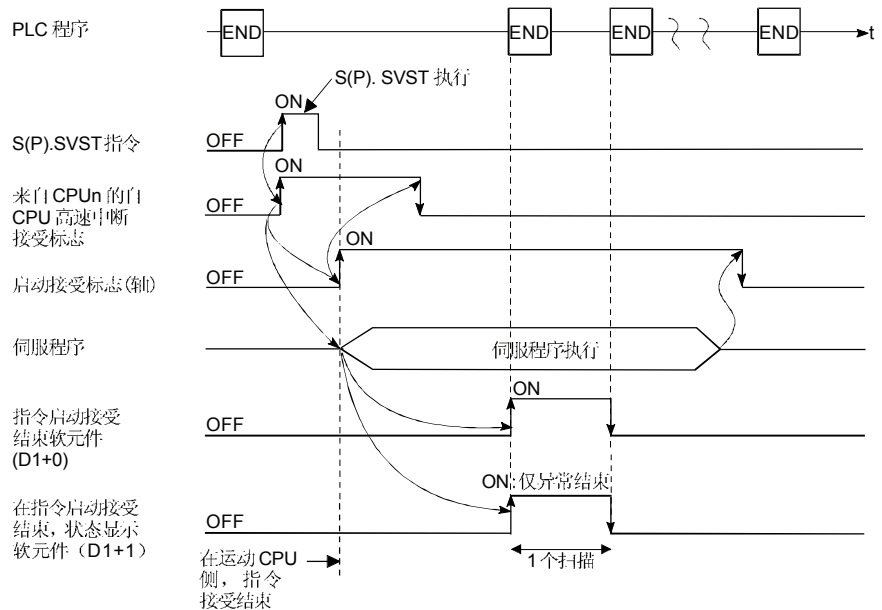
(注-2): "n" 显示于轴号对应的数值。

Q173CPU(N) : 轴 No.1 到 No.32 (n=1 到 32) / Q172CPU(N) : 轴 No.1 到 No.8 (n=1 到 8)。

[描述]

- (1) 对于在多 CPU 系统里的运动 CPU，本指令是专用指令。除运动 CPU 外，在 CPU 中执行时，会出现错误。
- (2) 用(S2)指定的伺服程序启动请求。
- (3) 当运动 CPU 的操作系统是 SV22 时，不管实模式 / 虚模式 / 模式切换时，本指令总是有效的。
- (4) 执行 S(P).SFCS 指令时，在 CPU 中不能同时执行 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDRD/ S(P).DDWR。当运动专用 PLC 指令被连续启动时，在执行指令的结束软元件启动后，必须通过来自 CPU 的自 CPU 高速中断接受标志进行互锁。
- (5) 在运动 CPU 的运动控制步(Kn)里，执行伺服程序时，必须通过用户程序进行互锁，因为在 PLC CPU 里没有可以区分轴的标志。运动软元件的启动接受标志 (M2001 到 M2032)或定位启动结束标志(M2400+20n)被用于互锁条件。
- (6) 必须用 CPU 共享内存启动接受标志进行互锁，这样可以避免在相同一运动 CPU 好的相同轴上执行多重指令。

[操作]



- (1) 在目标 CPU 的 CPU 共享内存里，各轴的启动接受状态可以通过启动接受标志确认。

- (2) S(P).SVST 指令接受和正常/异常结束可以用结束软元件(D1)或结束时状态显示软元件(D2)来确认。
- (a) 结束软元件
用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。
- (b) 结束时状态显示软元件
根据指令结束状态启动和关闭。
- 正常结束：OFF
 - 异常结束：用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。

[设置范围]

- (1) 启动轴的设置
在字顺序中，启动轴设置为(S1) 设置“J + 轴号”。

	(S1) 可用范围
Q173CPU(N)	1 到 32
Q172CPU(N)	1 到 8

最多可以设置 8 轴。如果设置多轴时，设置应该划分空间在系统设置（参见 1.5 节）中设定的轴号被当作用来启动的轴号。
轴号启动不需要顺序。

例) 当设置多轴 (轴 1, 轴 2, 轴 10, 轴 11) 时。
“J1J2J10J11”

- (2) 伺服程序号的设置

(S2) 可用范围
0 到 4095

5 运动专用 PLC 指令

[启动接受标志 (系统区域)]

启动接受标志的结束状态储存在 CPU 共享内存的启动接受标志的地址里。

CPU 共享内存地址 () 是十进制地址	说明									
204H(516) 205H(517)	<p>逐位储存 1 到 32 轴的启动接收标志。 (一个位的实际设置为 Q173CPU(N) : J1 到 J32/ Q172CPU(N) : J1 到 J8。)</p> <p>OFF : 启动接受标志可用 ON : 启动接受标志不可用</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>204H(516) 地址</td> <td style="text-align: center;">J16</td> <td style="text-align: center;">J2 J1</td> </tr> <tr> <td>205H(517) 地址</td> <td style="text-align: center;">J32</td> <td style="text-align: center;">J17</td> </tr> </table>	b15	b1	b0	204H(516) 地址	J16	J2 J1	205H(517) 地址	J32	J17
b15	b1	b0								
204H(516) 地址	J16	J2 J1								
205H(517) 地址	J32	J17								

[错误]

在如下情况下的异常结束中，出错代码储存在的结束状态指定的软元件(D2)中。

结束状态 ^(注) (出错代码)(H)	出错原因	处置方法
4C00	指定的软元件不能应用于运动 CPU, 它在软元件范围之外。	确认一个程序, 并且修正到正确的 PLC 程序里。
4C01	执行了与运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C03	执行的伺服程序号在 0 到 4095 范围之外。	
4C04	用 SVST 指令设置的轴号不正确。	
4C02	运动 SFC 程序号在 0 到 255 范围之外。	
4C08	由 PLC CPU 到运动 CPU 的指令 S(P).SFCS/ S(P).SVST/ S(P).CHGA/ S(P).GINT.运动总数超过了 32 个。运动 CPU 不能处理这些指令。	
4C09	指令原因的 CPU 号不正确。	

(注) : 0000H (正常)

5 运动专用 PLC 指令

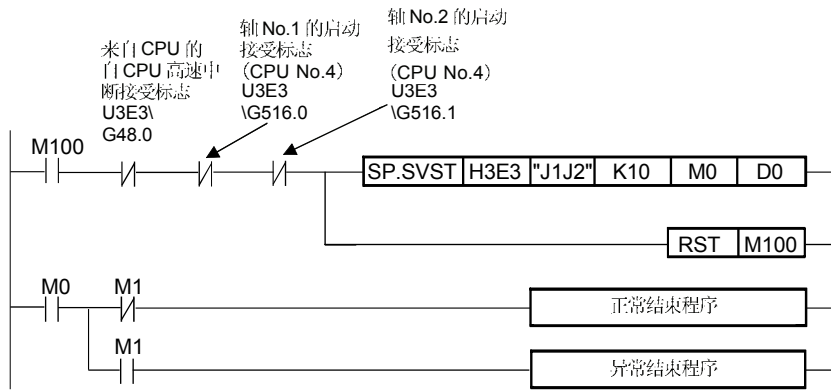
在如下情况下，出错标志(SM0)启动操作出错，出错代码储存在 SD0 里。

出错代码 ^{注)}	出错原因	处置方法
2110	指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序，并且修正到正确的 PLC 程序里。
2114	指定了“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU，指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU。	
4004	指令由除可用软元件的其它软元件组成	
4100	指定了“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH, 和 3E4H。	

(注) : 0000H (正常)

[编程举例]

对于 PLC CPU No.1 的运动 CPU No.4 的轴 No.1 和 No.2，用于要求启动伺服程序 No.10 的程序。



5 运动专用 PLC 指令

5.4 从 PLC CPU 到 运动 CPU 的当前值变更指令: S(P).CHGA (PLC 指令: S(P).CHGA)

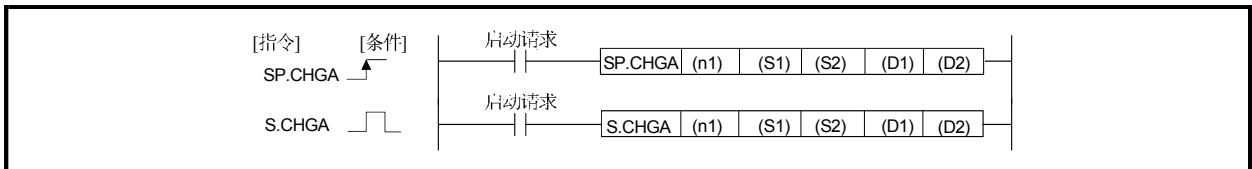
运动 CPU 和软件的可应用版本, 请参照“1.3.4”部分。

• 从 PLC CPU 到运动 CPU (S(P). CHGA) 的当前值变更指令

设置数据 ^(注)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接指定 软元件	MELSECNET/10 直接 J□\□		特殊功能 模块 U□\G□	变址寄存 器 Z□	常数 K, H	其它
	位	字				位	字				
(n1)		○			○					○	
(S1)											○
(S2)		○			○					○	
(D1)	○										
(D2)		○									

○: 可用 △: 部分可用

(注): 除 (S1) 之外的设置数据: 可索引



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 实际指定值如下。 ^(注-1) CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4 : 3E3H	16 位 二进制
(S1)	轴号("Jn") ^(注-1) 执行当前值变更。 Q173CPU(N) : J1 到 J32/Q172CPU(N) : J1 到 J8	字顺序
	同步编码器轴号("En")执行当前值变更。 Q173CPU(N) : E1 到 E12/Q172CPU(N) : E1 到 E8	
	凸轮轴号 ("Cn")执行一转中的当前值变更。 Q173CPU(N) : C1 到 C32/Q172CPU(N) : C1 到 C8	
(S2)	当前值变更的设置	32 位 二进制
(D1)	全部软元件	位
	(D1+0): 在指令启动接受结束时, 启动扫描的软元件 (D1+1): 在指令启动接受异常结束时, 启动扫描的软元件 (“D1+0”在异常结束时也会启动。)	
(D2)	储存结束状态的软元件	16 位 二进制

(注-1): 在多 CPU 构成中, 运动 CPU 不能使用 CPU No.1 。

(注-2): “n”表示与轴号对应的数字值。

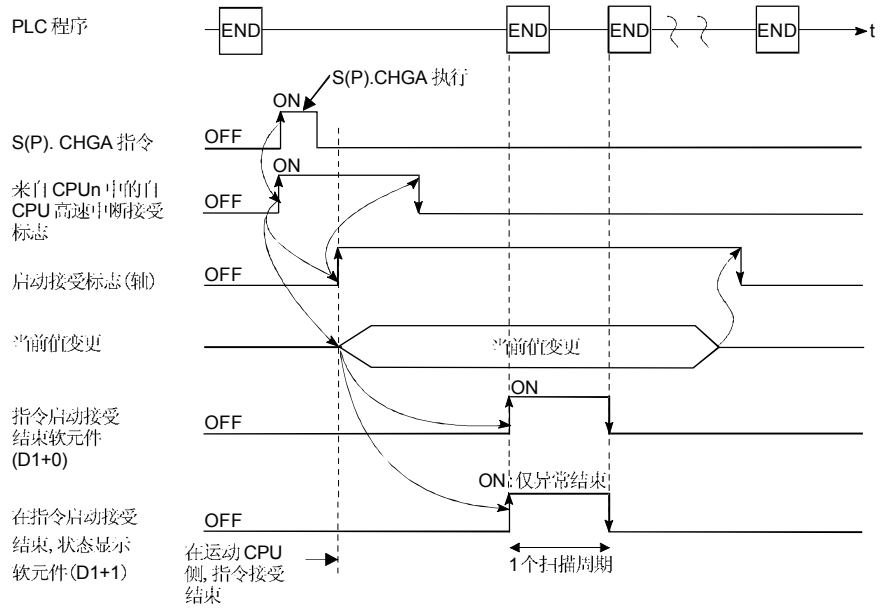
Q173CPU(N) : 轴 No.1 到 No.32 (n=1 到 32) / Q172CPU(N) : 轴 No.1 到 No.8 (n=1 到 8)

● 当轴号"Jn"用(S1)指定时

[控制]

- (1) 对于在多 CPU 系统里的运动 CPU，本指令是专用指令。
除运动 CPU 外，在 CPU 中执行时，会出现错误。
- (2) (S1)指定的轴（停止轴）号的当前值变更成(S2)指定的当前值。
- (3) 当运动 CPU 的操作系统软件是 SV22 时，不管实模式 / 虚模式 / 模式切换时，本指令总是有效的。
- (4) 对于执行 S(P).SFCS 指令的，在 CPU 不能同时执行 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDRD/S(P).DDWR 指令。
当运动专用 PLC 指令被连续启动时，在执行指令的结束软元件启动后，必须通过 CPU 的自出现 CPU 高速中断接受标志进行互锁。
- (5) 在运动 CPU 的运动控制步(Kn)里，执行伺服程序时，必须通过用户程序进行互锁，因为在 PLC CPU 里没有可以用于区分轴的标志。
运动软元件的启动接受标志 (M2001 到 M2032)被用作运动 CPU 中的互锁条件。
- (6) 必须通过 CPU 共享内存的启动接受标志进行互锁，这样可以避免在同一运动 CPU 的同一轴上执行多重指令。
- (7) 对在 S(P).SVST 里执行的轴执行 CHGA 指令的伺服程序可以进行当前值变更。

[操作]



- (1) 在目标 CPU 的 CPU 共享内存里, 各轴的启动接受状态可以用启动接受标志来确认。
- (2) S(P).SVST 指令接受和正常/异常结束可以用结束软元件(D1)或结束时状态显示软元件(D2)来确认。
 - (a) 结束软元件
 - 正常结束 : OFF
 - 异常结束 : 用指令结束的扫描 END 处理启动, 用下一次 END 处理断开。
 - (b) 结束时状态显示软元件

根据指令结束状态启动和断开。

[设置范围]

- (1) 启动轴的设置
在字顺序里, 启动轴设置为(S1)设置“J + 轴号”。

	(S1) 可用范围
Q173CPU(N)	1 到 32
Q172CPU(N)	1 到 8

可以设置的的轴只有 1 轴。
在系统设置(参照 1.5 节)中, 设定的轴号被当作启动轴号。

(2) 当前值变更的设置

(S2) 可用范围
-2147483648 到 2147483647

[启动接受标志(系统区域)]

启动接受标志的结束状态储存在 CPU 共享内存的启动接受标志的地址里。

CPU 共享内存地址 () 是十进制地址	说明															
204H(516) 205H(517)	<p>1 到 32 轴的启动接受标志被逐位储存。 (一个位的实际设置为 Q173CPU(N) : J1 到 J32/ Q172CPU(N) : J1 到 J8。)</p> <p>OFF :启动接受标志可用 ON :启动接受标志禁止</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td style="text-align: center;">b15</td> <td></td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>204H(516)地址</td> <td>J16</td> <td style="text-align: center;">.....</td> <td>J2</td> <td>J1</td> </tr> <tr> <td>205H(517)地址</td> <td>J32</td> <td style="text-align: center;">.....</td> <td></td> <td>J17</td> </tr> </table>		b15		b1	b0	204H(516)地址	J16	J2	J1	205H(517)地址	J32		J17
	b15		b1	b0												
204H(516)地址	J16	J2	J1												
205H(517)地址	J32		J17												

[出错]

在如下情况下的异常结束，出错代码储存在的结束状态指定的软元件(D2)中。

结束状态 (注) (出错代码)(H)	出错原因	处置方法
4C00	指定软元件不能应用于运动 CPU, 它在软元件范围之外。	确认一个程序, 并且修正到正确的 PLC 程序里。
4C01	执行了与运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C05	由 CHGA 指令设置的轴号不正确。	
4C08	由 PLC CPU 到运动 CPU 的指令 S(P).SFCS/ S(P).SVST/ S(P).CHGA/ S(P).GINT.运动总数超过了 32 个。运动 CPU 不能处理这些指令。	
4C09	指令原因的 CPU 号不正确。	

(注) : 0000H (正常)

5 运动专用 PLC 指令

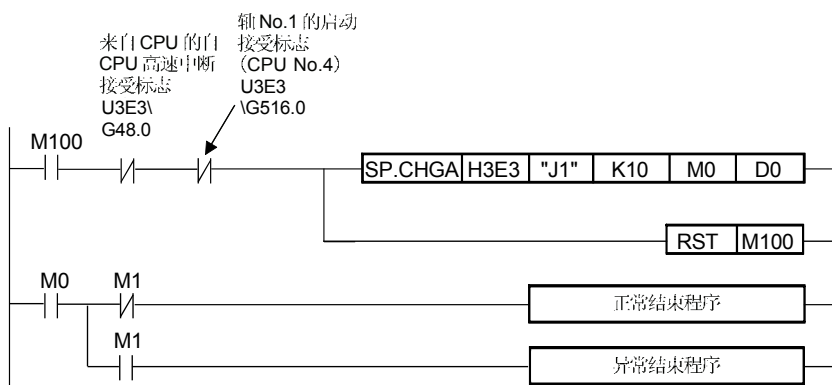
在如下情况下，出错标志(SM0)启动操作出错，出错代码储存在 SD0 里。

出错代码 ^{注)}	出错原因	处置方法
2110	指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序，并且修正到正确的 PLC 程序里。
2114	指定了“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU，指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU。	
4004	指令由除可用软元件的其它软元件组成	
4100	指定了“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH, 和 3E4H。	

(注) : 0000H (正常)

[编程举例]

对于 PLC CPU (CPU No.1)的运动 CPU (CPU No.4)的轴 No.1 的当前变更的程序

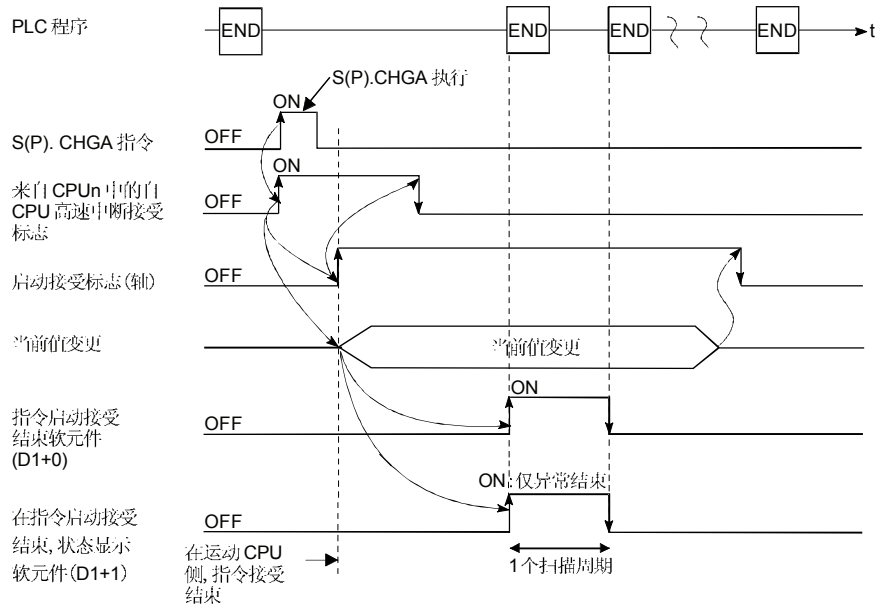


● 当轴号"En"被(S1)指定

[控制]

- (1) 对于在多 CPU 系统里的运动 CPU，本指令是专用指令。
除运动 CPU 外，在 CPU 中执行时，会出现错误。
- (2) 在虚模式中，(S1)指定的同步编码器轴当前值变更成(S2)指定的当前值。
- (3) 当运动 CPU 的操作系统软件是 SV22 时，不管是实模式 / 虚模式 / 模式切换时，本指令总是有效的。
- (4) 对执行 S(P).CHGA 指令的 CPU，不能同时执行 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDRD/ S(P).DDWR。
当运动专用 PLC 指令被连续启动时，在执行指令的结束软元件启动后，必须用 CPU 的自 CPU 高速中断接受标志进行互锁。
- (5) 在运动 CPU 控制步(Kn)中，执行伺服程序时，必须通过用户程序进行互锁，因为在 PLC CPU 里没有可以用于区分轴的标志。
运动软元件的同步编码器当前值的变更标志 (M2101 到 M2112)被用作运动 CPU 中的互锁条件。
- (6) 必须通过 CPU 共享内存的当前值变更标志进行互锁，这样可以避免在同一运动 CPU 的同一同步编码器轴上执行多重指令。
- (7) 在 S(P).SVST 中指令中执行伺服程序(对同步编码器轴执行 CHGA 指令)时，可以执行当前值变更。

[操作]



- (1) 同步编码器轴的当前值状态可以用在目标CPU的CPU共享内存里的当前值变更来确认。
- (2) S(P).SVST 指令接受和正常/异常结束可以用结束软元件(D1)或结束时状态显示软元件(D2)来确认。
 - 正常结束：OFF
 - 异常结束：用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。
 - (a) 结束软元件
 - 正常结束：用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。
 - 异常结束：用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。
 - (b) 结束时状态显示软元件
 - 正常结束：根据指令结束状态启动和关闭。
 - 异常结束：根据指令结束状态启动和关闭。

[设置范围]

- (1) 执行当前值变更的同步编码器轴的设置
在字顺序里，执行当前值变更的同步编码器轴设置为(S1)设置“E+ 同步编码器轴号”。

	(S1) 可用范围
Q173CPU(N)	1 到 12
Q172CPU(N)	1 到 8

可以设置的轴只有 1 轴。
在系统设置（参照 1.5 节）中，设置的轴号被当作启动轴号。

5 运动专用 PLC 指令

(2) 当前值变更的设置

(S2) 可用范围
-2147483648 到 2147483647

[同步编码器当前值变更标志(系统区域)]

同步编码器当前值变更标志的结束状态储存在CPU共享内存的步编码器当前值变更标志的地址里。

CPU 共享内存地址 () 是十进制地址	说明						
204H(516) 205H(517)	1 到 16 轴的同步编码器当前值变更标志被逐位储存。 (一个位的实际被设置为 Q173CPU(N) :E1 到 E12/ Q172CPU(N) : E1 到 E8.) OFF :启动接受标志可用 ON :启动接受标志禁止						
	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">E16</td> <td style="text-align: center;">••••••••</td> <td style="text-align: center;">E2 E1</td> </tr> </table> <p style="text-align: center;">208H(520) 地址</p>	b15	b1	b0	E16	••••••••	E2 E1
b15	b1	b0					
E16	••••••••	E2 E1					

[出错]

在如下情况下的异常结束，出错代码储存在结束状态指定的软元件(D2)中。

结束状态 (注) (出错代码)(H)	出错原因	处置方法
4C00	指定的软元件不能应用于运动 CPU, 它在软元件范围之外。	确认一个程序, 并且修正到正确的 PLC 程序里。
4C01	执行了与运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C05	由 CHGA 指令设置的轴号不正确。	
4C08	由 PLC CPU 到运动 CPU 的指令 S(P).SFCS/ S(P).SVST/ S(P).CHGA/ S(P).GINT.运动总数超过了 32 个。运动 CPU 不能处理这些指令。	
4C09	指令原因的 CPU 号不正确。	

(注) : 0000H (正常)

5 运动专用 PLC 指令

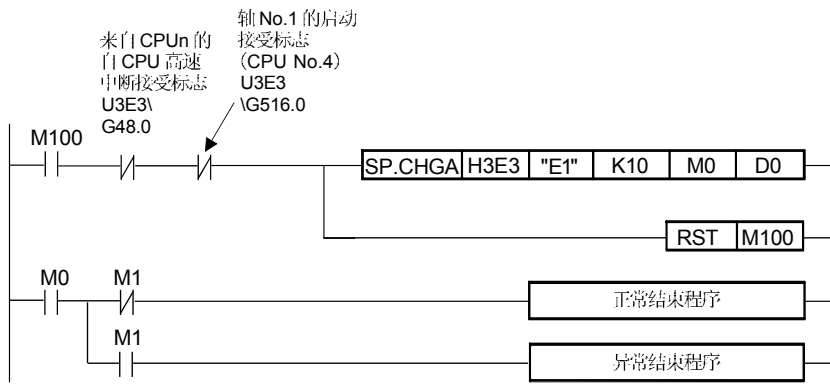
在如下情况下，出错标志(SM0)启动操作出错，出错代码储存在 SD0 里。

出错代码 ^{注)}	出错原因	处置方法
2110	指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序，并且修正到正确的 PLC 程序里。
2114	指定了“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU，指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU。	
4004	指令由除可用软元件的其它软元件组成	
4100	指定了“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH, 3E4H。	

(注) : 0000H (正常)

[编程举例]

对于 PLC CPU (CPU No.1)的运动 CPU (CPU No.4)的轴 No.1 的当前值的程序

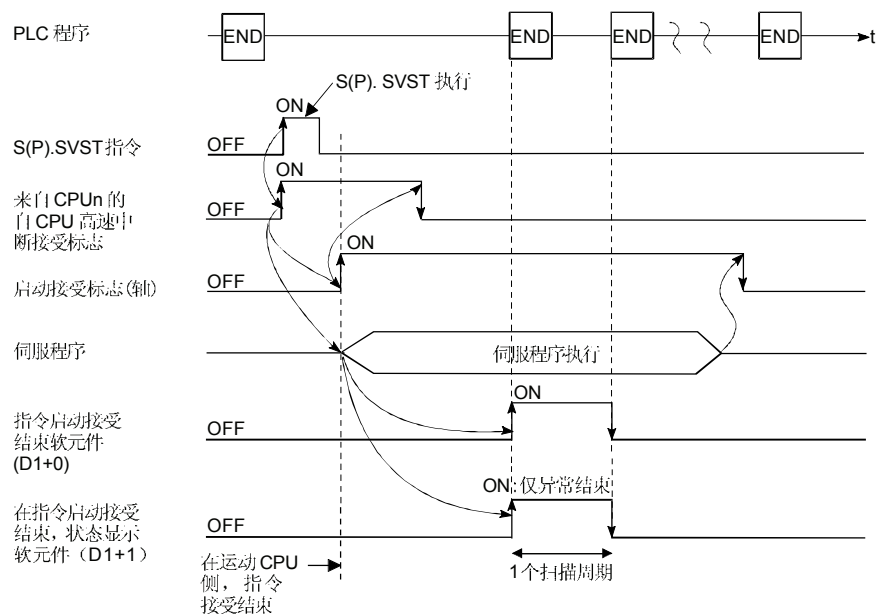


● 当轴号"Cn"被(S1)指定

[控制]

- (1) 对于在多 CPU 系统里的运动 CPU，本指令是专用指令。除运动 CPU 外，对 CPU 中执行时，会出现错误。
- (2) 在虚模式中，(S1)指定的凸轮轴一转中的轴当前值变更成(S2)指定的当前值。
- (3) 当运动 CPU 的操作系统软件是 SV22 时，不管处于实模式 / 虚拟模式 / 模式切换时，本指令总是有效的。
- (4) 对于执行 S(P). CHGA 指令的 CPU，不能同时执行 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDR/ S(P).DDWR 。
当运动专用 PLC 指令被连续启动时，在执行指令的结束软元件启动后，必须用 CPU 的自 CPU 高速中断接受标志采取互锁。
- (5) 在运动 CPU 的运动控制步(Kn)中，执行伺服程序时，必须通过用户程序进行互锁，因为在 PLC CPU 里没有可以用于区分轴的标志。
- (6) 必须用 CPU 共享内存的凸轮轴一转中的当前值变更标志执行互锁，这样可以避免在同一运动 CPU 的同一同步编码器轴上执行多重指令。
- (7) 对在 S(P).SVST 里执行的同步编码器轴执行 CHGA 指令的伺服程序，可以执行当前值变更。

[操作]



- (1) 凸轮轴一转中的当前值状态可以用在目标CPU的CPU共享内存里的凸轮轴的一转中的当前值变更标志来确认。
- (2) S(P).SVST 指令接受和正常/异常结束可以用结束软元件(D1)或结束时状态显示软元件(D2)来确认。
 - (a) 结束软元件
用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。
 - (b) 结束时状态显示软元件
根据指令结束状态启动和关闭。
 - 正常结束 : OFF
 - 异常结束 : 用指令结束的扫描 END 处理启动，用下一次 END 处理关闭。

[设置范围]

- (1) 执行一转中的当前值变更的凸轮轴的设置。
在字顺序里，执行一转中的当前值变更的凸轮轴设置为(S1)设置“C+ 凸轮轴号”。

	(S1) 可用范围
Q173CPU(N)	1 到 32
Q172CPU(N)	1 到 8

可以设置的轴只有 1 轴。
在参数设置（参照 1.5 节）中，设置的轴号被当作启动轴号。

5 运动专用 PLC 指令

(2) 当前值变更的设置

(S2) 可用范围
-2147483648 到 2147483647

[凸轮轴一转中的当前值变更标志(系统区域)]

凸轮轴一转中的当前值变更标志的结束状态储存在CPU共享内存的凸轮轴一转中的当前值变更标志的地址里。

CPU 共享内存地址 () 是十进制地址	说明															
20CH(524) 20DH(525)	<p>凸轮轴一转中的当前值变更标志从 1 到 16 轴逐位位储存。 (一个位的实际设置为 Q173CPU(N):C1 到 C32/ Q172CPU(N):C1 到 C8.)</p> <p>OFF :启动接受标志可用 ON :启动接受标志禁止</p> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td style="text-align: center;">b15</td> <td></td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>20CH(524) 地址</td> <td style="text-align: center;">C16</td> <td style="text-align: center;">••••••••</td> <td style="text-align: center;">C2</td> <td style="text-align: center;">C1</td> </tr> <tr> <td>20DH(525) 地址</td> <td style="text-align: center;">C32</td> <td style="text-align: center;">••••••••</td> <td></td> <td style="text-align: center;">C17</td> </tr> </table>		b15		b1	b0	20CH(524) 地址	C16	••••••••	C2	C1	20DH(525) 地址	C32	••••••••		C17
	b15		b1	b0												
20CH(524) 地址	C16	••••••••	C2	C1												
20DH(525) 地址	C32	••••••••		C17												

[出错]

在如下情况下的异常结束中，出错代码储存在结束状态指定的软元件(D2)中。

结束状态 (注) (出错代码)(H)	出错原因	处置方法
4C00	指定的软元件不能应用于运动 CPU, 它在软元件范围之外。	确认一个程序, 并且修正到正确的 PLC 程序里。
4C01	执行与运动 CPU 的操作系统不相对应的多 CPU 系统指令的。	
4C05	由 CHGA 指令设置的轴号不正确。	
4C08	由 PLC CPU 到运动 CPU 的指令 S(P).SFCS/ S(P).SVST/ S(P).CHGA/ S(P).GINT.运动总数超过了 32 个。运动 CPU 不能处理这些指令。	
4C09	指令原因的 CPU 号不正确。	

(注) : 0000H (正常)

5 运动专用 PLC 指令

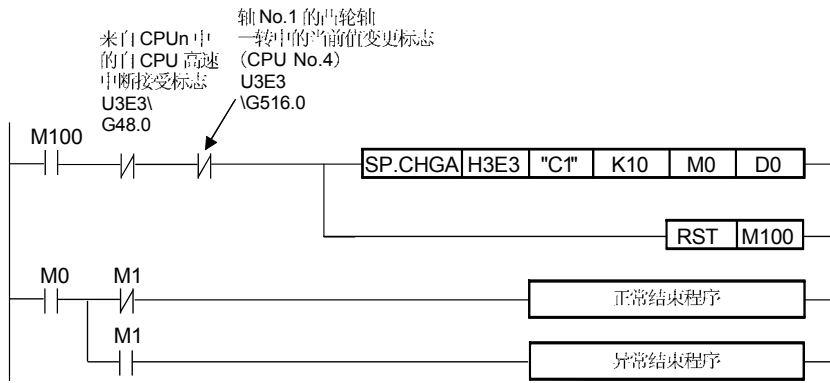
在如下情况下，出错标志(SM0)启动操作出错，出错代码储存在 SD0 里。

出错代码 ^{注)}	出错原因	处置方法
2110	指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序，并且修正到正确的 PLC 程序里。
2114	指定了“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU，指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU。	
4004	指令由除可用软元件的其它软元件组成	
4100	指定了“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH, 3E4H。	

(注) : 0000H (正常)

[程序举例]

对于 PLC CPU (CPU No.1)的运动 CPU (CPU No.4)的轴 No.1 的当前值变更的程序



5 运动专用 PLC 指令

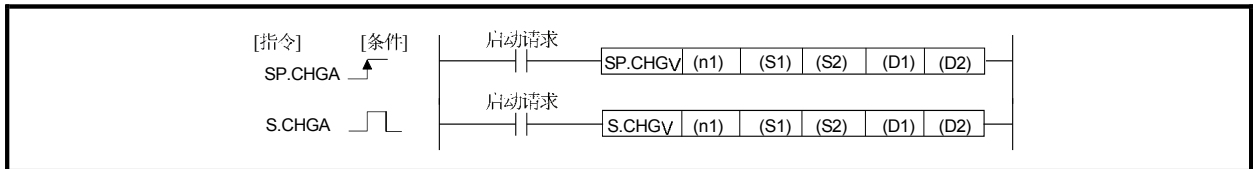
5.5 从 PLC CPU 到运动 CPU 的速度变更指令: S(P).CHGA (PLC 指令: S(P).CHGV)

有关运动 CPU 和软件的适用版本参见第 1.3.4 节

• 速度变更指令(S(P).CHGV)

设置数据 (注意)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接 指定 软元件	MELSECNET/10 直接 J□\□		特殊功能 模块 U□\G□	变址 寄存器 Z□	常数 K, H	其它
	位	字				位	字				
(n1)		○			○					○	
(S1)											○
((S2))		○			○					○	
(D1)	○										
(D2)		○									

○ : 可用 △ : 部分可用
(注) 数据设置除 (S1) 外: 可索引



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 下列为实际指定值 ^(注-1) 2 号 CPU: 3E1H, 3 号 CPU : 3E2H, 4 号 CPU: 3E3H	16 位 二进制
(S1)	轴号("Jn") ^(注-2) 控制速度变更	字顺序
(S2)	当前值设置变更	16 位 二进制
(D1)	完全软元件 (D1+0) : 在指令启动接收正常结束时启动进行扫描的软元件 (D1+1) : 在指令接收异常结束时启动进行扫描的软元件 (“D1+0”在异常结束时也处于工作状态。)	位
(D2)	存储结束状态的软元件	16 位 二进制

(注-1) : 运动 CPU 不能在多 CPU 配置系统中使用 CPU No.1。

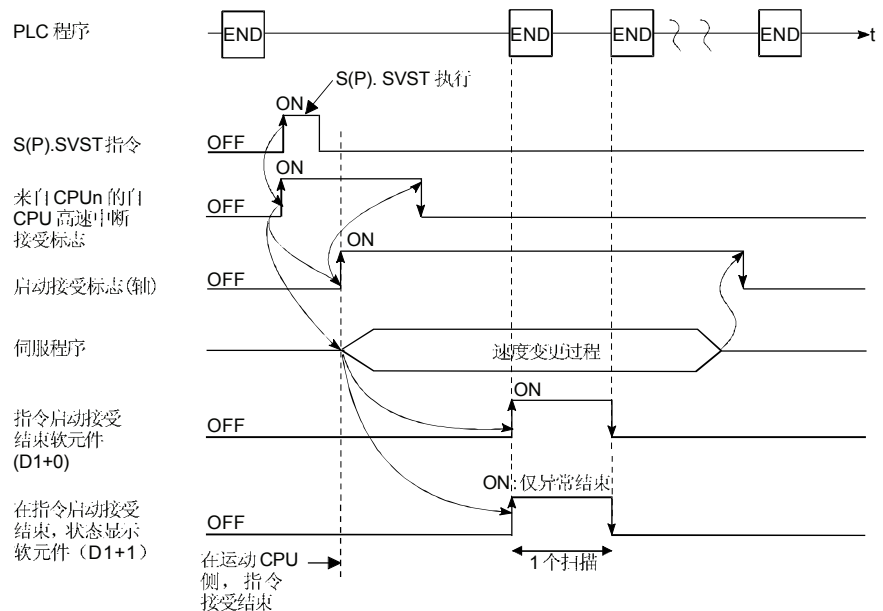
(注-2) : “n”表示轴号所对应的数值。

Q173CPU(N) : 轴 No.1 到 No.32 (n=1 到 32) / Q172CPU(N) : 轴 No.1 到 No.8 (n=1 到 8)

[控制]

- (1) 该指令只应用于多 CPU 系统中的运动 CPU，当应用于其他 CPU 时会出现错误。
- (2) 速度变更是通过在定位中或 JOG 操作中(S1)指定的轴来实现的。
- (3) 对于执行 S(P).CHGV 指令的 CPU, 不能同时执行 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDRD/ S(P).DDWR。
运动专用 PLC 指令继续启动执行时，需要利用来自 CPU_n 的自 CPU 高速中断接收标志进行互锁。
- (4) 在运动 CPU 的操作控制步中发生速度变更时，则需通过用户程序进行互锁，因为此时无任何标志来区分 PLC CPU 中的速度变更。运动元件中的速度变更标志(M2061 到 2092)在运动 CPU 中作为互锁条件使用。
- (5) 通过 CPU 共享内存速度变更标志来进行互锁，这样可以避免在同一运动 CPU 的同一轴上执行多重指令。

[操作]



[设置范围]

- (1) 设置轴来执行速度变更
 执行速度变更的轴在字顺序中设置为(S1)设置“J+ 轴号”。

	(S1)可用范围
Q173CPU(N)	1 到 32
Q172CPU(N)	1 到 8

可设置的轴的数量仅为 1 个。
 在系统设置中设置的轴号(参见第 1.5 节)作为启动轴号。

- (2) 速度变更设置

(S2)可用范围
-2147483648 ~ 2147483647

[速度变更标志(系统部分)]

启动接收标志的结束状态存放在共享 CPU 存储器的启动接收标志的地址上。

共享 CPU 存储器地址 ()是十进制地址	说明																			
206H(518) 207H(519)	启动接收标志被储存在每位 1 到 32 条轴中。 (位实际设置 Q173CPU(N) : J1 到 J32/) Q172CPU(N) : J1 到 J8.) OFF: 启动接收可用 ON : 启动接收禁止																			
	<table border="1"> <thead> <tr> <th colspan="2"></th> <th>b15</th> <th colspan="2"></th> <th>b1</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>206H(518) 地址</td> <td>J16</td> <td>.....</td> <td></td> <td>J2</td> <td>J1</td> </tr> <tr> <td>207H(519) 地址</td> <td>J32</td> <td>.....</td> <td></td> <td></td> <td>J17</td> </tr> </tbody> </table>			b15			b1	b0	206H(518) 地址	J16		J2	J1	207H(519) 地址	J32			J17
		b15			b1	b0														
206H(518) 地址	J16		J2	J1															
207H(519) 地址	J32			J17															

[错误]

下表为异常结束状态，出错代码存储在指定的结束状态软元件中 (D2)。

结束状态(注) (出错代码)(H)	原因	解决方法
4C00	指定的软元件不能应用于运动 CPU, 它在软元件范围之外。	确认一个程序, 并且修正到正确的 PLC 程序里。
4C01	执行了与运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C06	由 CHGA 指令设置的轴号不正确。	
4C09	指令原因的 CPU 号不正确。	

(注): 0000H (正常)

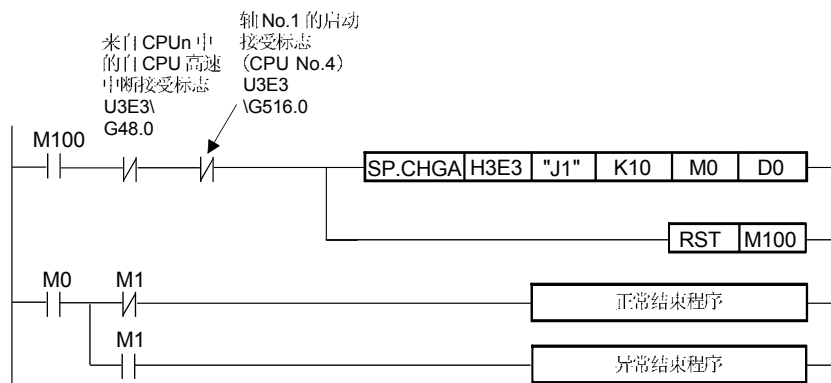
出错标志(SM0)表现在下表中的操作出错状况中，出错代码存放在 SDO 中。

出错代码 ^{注)}	出错原因	处置方法
2110	指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序, 并且修正到正确的 PLC 程序里。
2114	指定了“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU, 指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU。	
4004	指令由除可用软元件的其它软元件组成。	
4100	指定了“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH, 和 3E4H。	

(注): 0000H (正常)

[程序实例]

PLC CPU (CPU No.1)的运动 CPU (CPU No.4)的轴 No.1 的当前值变更的程序



5 运动专用 PLC 指令

5.6 从 PLC CPU 到运动 CPU 的转矩值变更请求指令 S(P).CHGT (PLC 指令:S(P).CHGT)

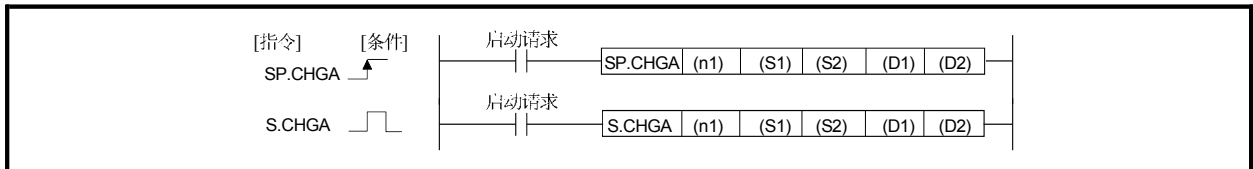
运动 CPU 和软件的可应用版本参见 1.3.4 节。

• 从 PLC CPU 到运动 CPU 的转矩值变更请求指令(S(P).CHGT)

设置数据 (注)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接指定 软元件	MELSECNET/10 直接 J□\□		特殊功能 模块 U□\G□	变址 寄存器 Z□	常数 K, H	其它
	位	字				位	字				
(n1)		○			○					○	
(S1)											○
((S2))		○			○					○	
(D1)	○										
(D2)		○									

○: 可用 △: 部分可用

(注): 设置 (S1) 以外的数据: 可索引



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 下列为实际指定值 (注 1) CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4: 3E3H	16 位 二进制
(S1)	执行转矩值变更的轴号("Jn")(注 2)	字顺序
(S2)	转矩值变更的设置	16 位 二进制
(D1)	结束软元件 (D1+0): 指令的启动正常接收结束时进行扫描的软元件 (D1+1): 指令的启动接收异常结束时进行扫描的软元件 ("D1+0" 在异常结束状态下亦处于工作状态)	位
(D2)	存储结束状态的软元件	16 位 二进制

(注-1): 运动 CPU 不能用在多 CPU 配置状态下使用 CPU No.1。

(注-2): "n" 表示与轴号对应的数值。

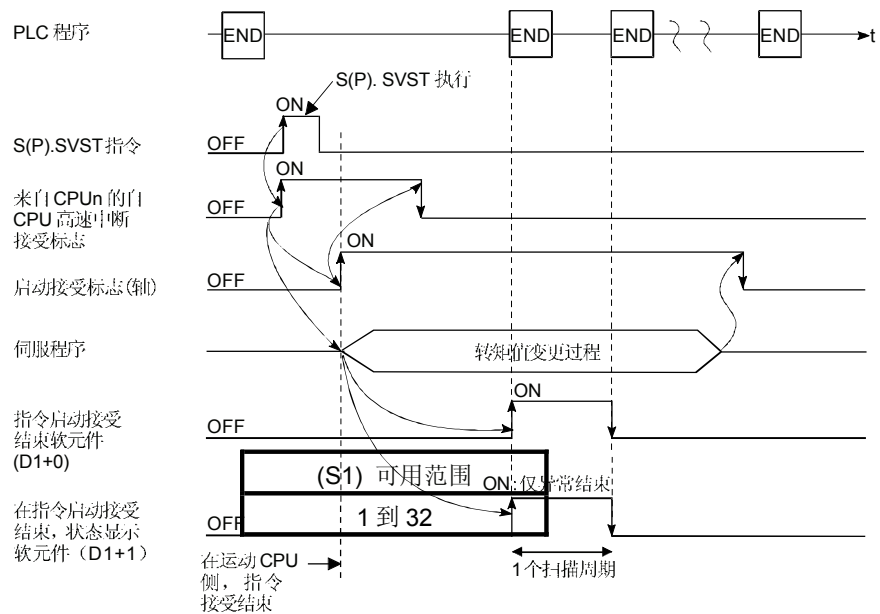
Q173CPU(N): 轴 No.1 到 No.32 (n=1 到 32) / Q172CPU(N): 轴 No.1 到 No.8 (n=1 到 8)

[控制]

- (1) 该指令只应用于多 CPU 系统中的运动 CPU，当应用于其他 CPU 时会出错。
- (2) 在实际操作或停止过程中，不管状态如何，指定的(S1)的转矩值可变更为(S2)。
- (3) 对于执行 S(P).CHGT 指令的 CPU，不能同时执行 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDR/ S(P).DDWR。

运动专用 PLC 指令继续启动执行时，需要通过来自 CPU_n 中的自 CPU 高速中断接收标志进行互锁。

[操作]



[设置范围]

- (1) 设置轴来执行转矩值的变更
执行转矩值变更的轴在字顺序“ ”中设置为(S1)设置“J+ 轴号”
系统设置中设置的轴号(参见 1.5 节)作为启动轴号。

可设置的轴的数量仅为 1 个。

- (2) 转矩值变更设置

(S2) 可用范围
1 到 500

[错误]

下表为异常结束状态时，出错代码存放在指定的结束状态软元件中 (D2)。

结束状态(注) (出错代码)(H)	原因	解决方法
4C00	指定的软元件不能应用于运动 CPU 中, 它在软元件范围之外。	确认一个程序, 并且修正到正 确的 PLC 程序 里。
4C01	执行了与运动 CPU 的操作系统不相对应的多 CPU 系统指令。	
4C07	CHGT 指令设置轴号不正确	
4C09	指令原因的 CPU 号不正确。	

(注):0000H (正常)

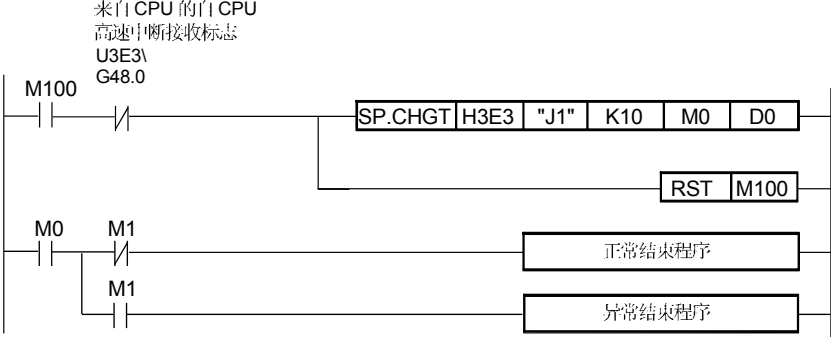
出错标志(SM0)在下表中的操作出错状况中为“1”，并且有一个出错代码存放在 SD0 中。

出错代码 ^(注)	原因	解决方法
2110	指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU 号。	确认一个程序, 并且修正到正 确的 PLC 程序 里。
2114	指定了“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU。	
2117	除运动 CPU, 指定了“(目标 CPU 的起始 I/O 号)/16”设置的 CPU。	
4004	指令由除可用软元件以外的其它软元件组成	
4100	指定了“(目标 CPU 的起始 I/O 号)/16”指定的 0 到 3DFH, 和 3E4H。	

(注):0000H (正常)

[编程实例]

变更运动 CPU 轴 No.1 转矩值的程序
(CPU No.4)从 PLC (CPU No.1)到 10(%)



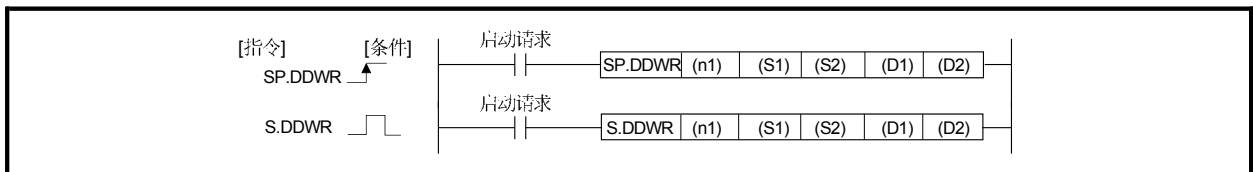
5 运动专用 PLC 指令

5.7 从 PLC CPU 到运动 CPU: S(P).DDWR (PLC 指令: $\boxed{S(P).DDWR}$)

• 从 PLC CPU 到运动 CPU ($\boxed{S(P).DDWR}$) 的写入指令

设置数据 ^(注)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接指定 软元件	MELSECNET/10 直接 J□\□		特殊功能 模块 U□\G□	变址 寄存器 Z□	常数 K, H	其他
	位	字				位	字				
(n1)		○	○		○					○	
(S1)		○	○		○						
(S2)		○	○	△	○						
(D1)		○	○	△							
(D2)	○	○	○								

○:可用 △:部分可用
(注) 设置数据(n1)到(n2):可索引



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 下列为实际指定值 (注 1) CPU No.1: 3E0H, CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H	16 位二进制
(S1)	储存自 CPU 的起始软元件的控制数据	16 位二进制
(S2)	储存自 CPU 的起始软元件的写入数据	
(D1)	储存目标运动 CPU 的起始软元件的写入数据	
(D2)	指令结束时进行扫描的位软元件	位

(注-1): 运动 CPU 不能用在多 CPU 配置状态下使用 CPU No.1。

[控制数据]

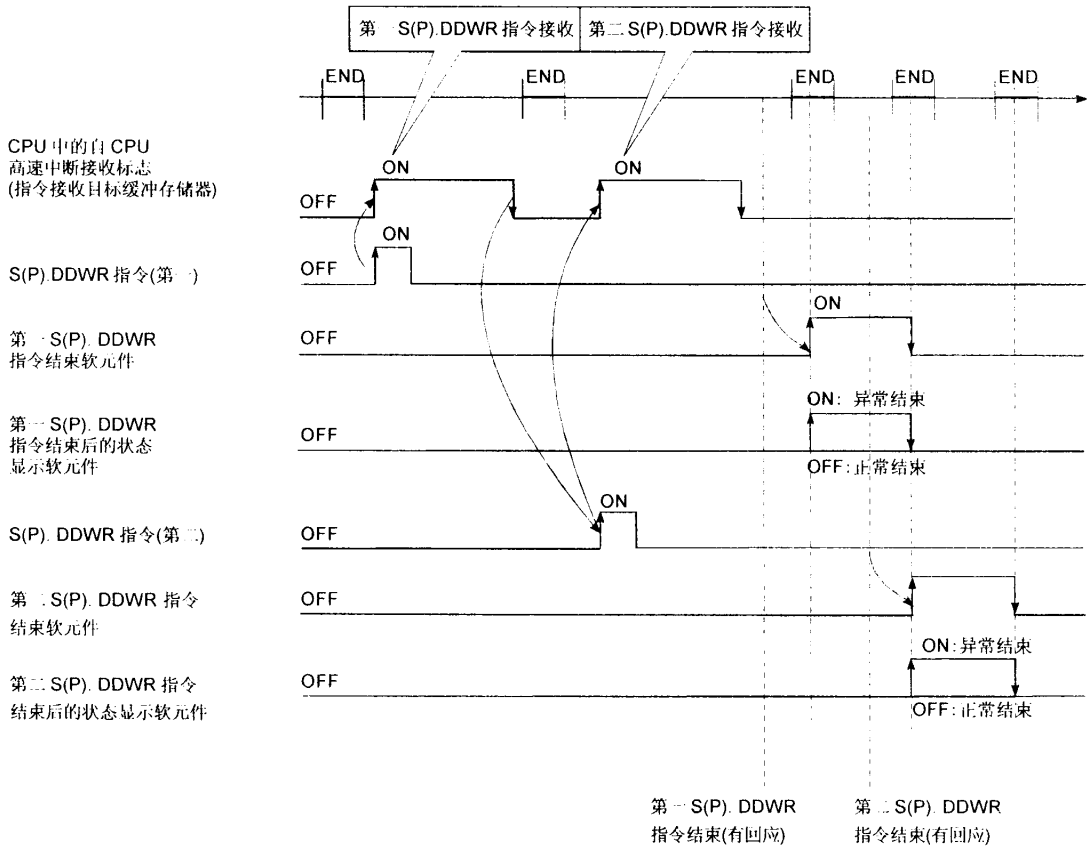
软元件	事项	设置数据	设置范围	设置者
S1+0	结束状态	储存指令结束时的状态结果: 0 : 无出错(正常结束) 除 0 之外 : 出错代码	—	系统
S1+1	写入数据数	设置写入数据数	1 到 16	用户

[控制]

- (1) 该指令只应用于多 CPU 系统中的运动 CPU，当应用于其他 CPU 时会出错。
从自 CPU 中(S2)指定的软元件，将由 S1 数据指定的控制数据的写入数据数存储在多 CPU 系统中目标 CPU 的(n1)中(D1)指定的字软元件中。
- (2) 可以对(S2)和(D1)指定数字的位元件。然而，数字指定有四位数字，而位元件数只是 16 的倍数。这就可能在其他值都设置的状态下产生出错代码(4004)。
- (3) 如果目标 CPU 在不接受指令的情况下，即使执行了 S(P).DDWR 指令，程序仍然不能操作。该情况下，需要重复执行 S(P).DDWR 指令。
对于执行 S(P).DDWR 指令的 CPU，不能同时执行(S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDR/ S(P).DDWR。可通过目标 CPU (运动 CPU)的共享存储器内的数据来确认指令是否可接受。当运动 PLC 指令继续启动时，那么在执行结束软元件后执行一个新的指令。
- (4) 目标 CPU 软元件的范围调整不是在 S(P).DDWR 指令执行的基础上通过自 CPU，而是通过目标 CPU 执行的，并且在软元件范围内出现异常结束状态。
- (5) S(P).DDWR 指令与正常/异常结束可以在结束过程中通过结束软元件(D1)或状态显示软元件(D2)来确认。
 - (a) 结束软元件
在指令结束时扫描的结束过程中接通，在下一个结束过程中断开。
 - (b) 结束中的状态显示软元件
根据指令结束的状态接通或断开
 - 正常结束: 断开
 - 异常结束: 在指令结束时扫描的结束过程中启动，在下一个结束过程中断开。
- (6) 目标 CPU 指定的(n1)结束被接受时，SM390 启动。
目标 CPU 指定的(n1)不能用复位状态或出错原因(5000 到 5999)正确写入时，SM390 关闭。

5 运动专用 PLC 指令

[执行 S(P).DDWR 指令时的自 CPU 操作]



[出错]

下表为异常结束状态，出错代码存放在控制数据(S1+0: 结束状态)

结束状态(注) (错误代码)(H)	原因	解决方法
4C00	指定软元件不能应用于运动 CPU, 它在软元件范围之外。	确认并更正一个程序, 使之来更正 PLC 程序
4C08	S(P).DDR/S(P).DDWR 总数表中的 PLC CPU 有至少 32 个运动 CPU 的指令请求, 而运动 CPU 不能处理它们。	
4C09	指令原因的 CPU 号不正确。	

(注): 0000H(正常)

5 运动专用 PLC 指令

出错标志(SM0)在下表中的操作出错状况中，且有一个出错代码存放在 SD0 中。

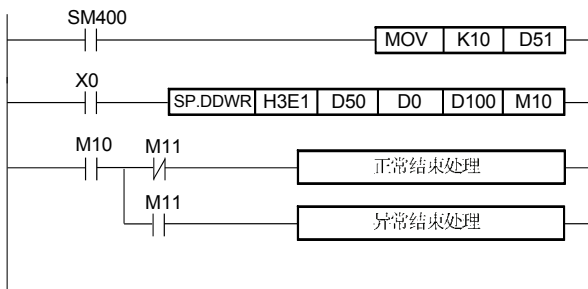
出错代码(注)	出错原因	处置方法
2110	指定需要通过“(目标 CPU 的起始 I/O 号)/16”设置的 CPU	确认并更正一个程序，使之来更正 PLC 程序
2114	指定通过“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU	
2117	指定通过“(目标 CPU 的起始 I/O 号)” /16 设置的除运动 CPU 之外的 CPU	
4002	指定的指令发生出错。	
4004	指令由除可用软元件组成。	
4100	指定需要通过“(目标 CPU 起始 I/O 号)” /16 设置的 0 到 3DFH 和 3E4H	
4101	待写入的数据数不在 1 到 16 范围内。	
	待写入的数据数超过存储软元件的范围	

(注):0000H (正常)

[编程示例]

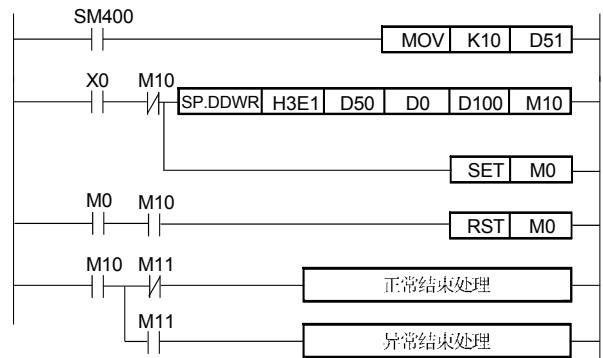
<例 1>

当 X0 闭合时，从 CPUNo.2 的 D100 开始，自 CPU(CPUNo.1)的 D0 中存储 10 点数据的程序



<例 2>

当 X0 闭合时，从 CPUNo.2 的 D100 开始，自 CPU(CPUNo.1)的 D0 中存储 10 点数据的程序



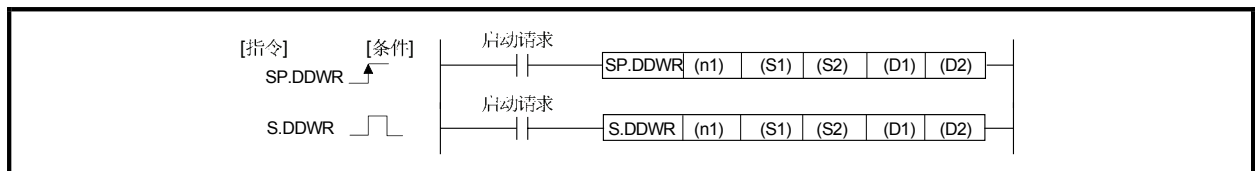
5 运动专用 PLC 指令

5.8 读取运动 CPU 的软元件：S(P).DDRD (PLC 指令：S(P).DDRD)

• 读取运动 CPU 的软元件指令：S(P).DDRD

设置数据 (注)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接指定 软元件	MELSECNET/10 直接 J□□□		特殊功能 模块 U□G□	变址 寄存器 Z□	常数 K, H	其他
	位	字				位	字				
(n1)		○	○		○					○	
(S1)		○	○		○						
(S2)		○	○	△	○						
(D1)		○	○	△							
(D2)	○	○	○								

○ :可用 △ :部分可用
(注)：设置数据(n1)到(n2):可索引。



[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的起始 I/O 号)/16 下列为实际指定值 (注 1) CPU No.1: 3E0H, CPU No.2 : 3E1H, CPU No.3 : 3E2H, CPU No.4: 3E3H	16 位二进制
(S1)	存储自 CPU 的起始软元件的控制数据	16 位二进制
(S2)	存储自 CPU 的起始软元件的写入数据	
(D1)	存储目标运动 CPU 的起始软元件的写入数据	
(D2)	指令结束时进行扫描的位元件	位

(注-1): 运动 CPU 不能用在多 CPU 配置状态下使用 CPU No.1。

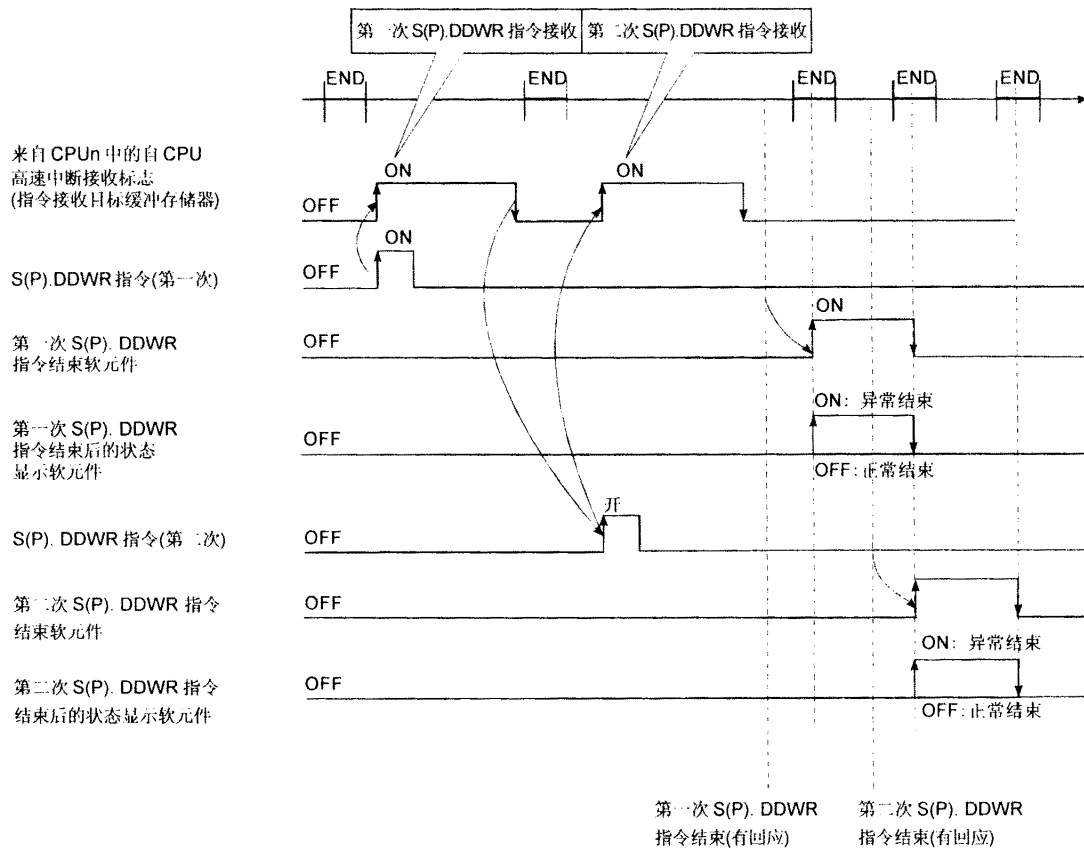
[控制数据]

软元件	事项	设置数据	设置范围	设置者
S1+0	结束状态	储存指令结束时的状态： 0 : 无出错(正常结束) 除 0 之外 : 出错代码	—	系统
S1+1	写入数据数	设置写入数据数	1 到 16	用户

[控制]

- (1) 该指令只应用于多 CPU 系统中的运动 CPU，当应用于其他 CPU 时会出错。
从自 CPU 中(S2)指定的软元件，将由 S1 数据指定的控制数据的写入数据数软元件存储在多 CPU 系统中目标 CPU 的(n1)中(D1)指定的字软元件中。
- (2) 可以对(S2)和(D1)指定数字的位元件。然而，数字指定有四位数，而位元件数只是 16 的倍数。这就可能在其他值都设置的状态下产生出错代码(4004)。
- (3) 如果目标 CPU 在不接受指令的情况下，即使执行了 S(P).DDWR 指令，程序仍然不能操作。该情况下，需要重复执行 S(P).DDWR 指令。
对于执行 S(P).DDWR 指令的 CPU，不能同时执行(S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).CHGV/S(P).CHGT/S(P).DDR/ S(P).DDWR。可通过目标 CPU (运动 CPU)的共享存储器内的数据来确认指令是否可接受。当运动 PLC 指令继续启动时，那么在执行结束软元件后执行一个新的指令。
- (4) 目标 CPU 软元件的范围调整不是在 S(P).DDWR 指令执行的基础上通过自 CPU，而是通过目标 CPU 执行的，并且在软元件范围内出现异常结束状态。
- (5) S(P).DDWR 指令与正常/异常结束可以在结束过程中通过结束软元件(D1)或状态显示软元件(D2)来确认。
 - (a) 结束软元件
在指令结束时扫描的结束过程中接通，在下一个结束过程中断开。
 - (b) 结束中的状态显示软元件
根据指令结束的状态接通或断开
 - 正常结束: 断开
 - 异常结束: 在指令结束时扫描的结束过程中启动，在下一个结束过程中关闭。
- (6) 目标 CPU 指定的(n1)结束被接受时，SM390 启动。
目标 CPU 指定的(n1)不能用复位状态或出错原因(5000 到 5999)正确写入时，SM390 关闭。

[执行 S(P).DDR D 指令时的自 CPU 操作]



[出错]

下表为异常结束状态，出错代码存放在控制数据(S1+0: 结束状态)

结束状态(注) 代码(H)	原因	解决方法
4C00	指定软元件不能应用于运动 CPU, 它在软元件范围之外。	确认并更正一个程序, 使之来更正 PLC 程序
4C08	从 PLC CPU 到运动 CPU 的指令 S(P).DDR D/S(P).DDR W 总数超过了 32 个。运动 CPU 不能处理这些指令。	
4C09	指令原因的 CPU 号不正确。	

(注): 0000H(正常)

5 运动专用 PLC 指令

出错标志(SM0)在下表中的操作出错状况中, 且有一个出错代码存放在 SD0 中。

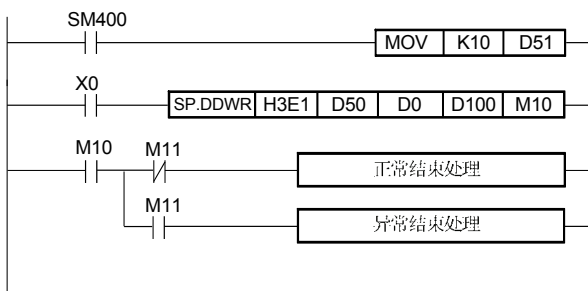
出错代码(注)	原因	解决方法
2110	指定需要通过“(目标 CPU 的起始 I/O 号)/16”设置的 CPU	确认并更正一个程序, 使之来更正 PLC 程序
2114	指定通过“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU	
2117	指定通过“(目标 CPU 的起始 I/O 号)” /16 设置的除运动 CPU 之外的 CPU	
4002	指定的指令发生出错。	
4004	指令由除可用软元件之外的软元件组成。	
4100	指定需要通过“(目标 CPU 起始 I/O 号)” /16 设置的 0 到 3DFH 和 3E4H	
4101	待写入的数据数不在 1 到 16 中。	
	待写入的数据数超过存储软元件的范围	

(注): 0000H (正常)

[程序实例]

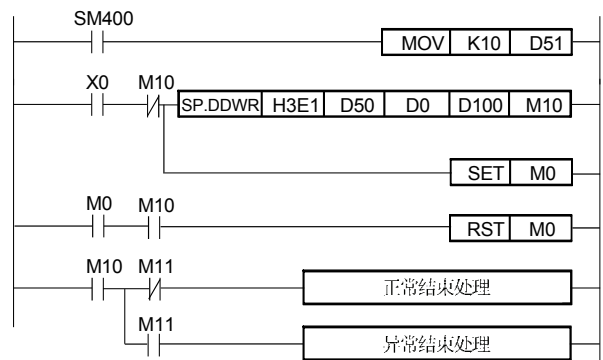
<例1>

当 X0 闭合时, 从自 CPU(CPUNo.1)的 D100 开始, CPU 的 D0 中存储 10 点数据的程序



<例2>

当 X0 闭合时, 从自 CPU(CPUNo.1)的 D100 开始, CPUNo.2 的 D0 中存储 10 点数据的程序



5 运动专用 PLC 指令

5.9 到其他 CPU 的中断指令：S(P).GINT (PLC 指令： S(P).GINT)

• 其他 CPU (S(P).GINT) 的中断指令

设置数据 ^(注)	可用软元件										
	内部软元件 (系统, 用户)		文件 寄存器	指定位	间接指定 软元件	MELSECNET/10 直接 J□\□		特殊功能模 块 U□\G□	变址 寄存器 Z□	常数 K, H	其他
	位	字				位	字				
(n1)		○	○							○	
(n2)										○	

○ :可用 △ :部分可用
(注): 设置数据(n1)到(n2):可索引



(注-1): 运动 CPU 不能在多 CPU 配置状态下使用 CPU No.1。

[设置数据]

设置数据	说明	数据类型
(n1)	(目标 CPU 的第 1 个 I/O 号)/16 下列为实际指定值 (注 1) CPU No.1: 3E0H, CPU No.2 : 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H	16 位二进制
(n2)	中断指令号。(0 到 15)	16 位二进制

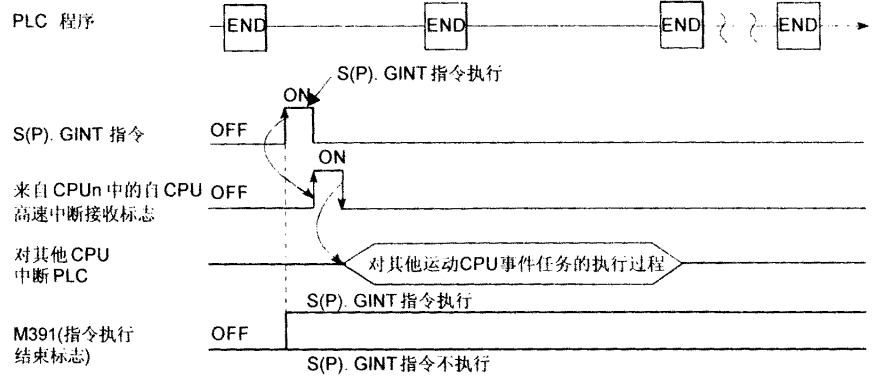
[控制]

当 S(P).GINT 执行指令启动(OFF → ON)时, 该指令通过 PLC 程序对运动 CPU 执行中断。运动 CPU 执行 PLC CPU 中断过程中, 执行由“事件任务的 PLC 中断”设置的运动 SFC 程序(操作程序状态)。

- (1) 只要运动 CPU 的操作系统是 SV22, 无论是在实模式/虚模式/模式切换中哪个状态, 该指令都有效。
- (2) 运动 CPU 处于 DI(中断禁止)时, 可使事件过程处于等待 EI(中断允许)指令执行的状态中。
- (3) 对目标 CPU 的指令发送结束后 SM390 启动。同时打开 SM391(S(P).GINT 指令结束标志)。

- (4) 对目标 CPU 发送指令没有完成时, SM390 关闭。该指令无法发送时, 则关闭 SM391(S(P). GINT 指令执行结束标志)。
- (5) 在 S(P).GINT 指令的目标共享 CPU 存储器中, 对从 CPU 到自 CPU 高速中断接收标志的指令执行数量没有限制。

[操作]



[出错]

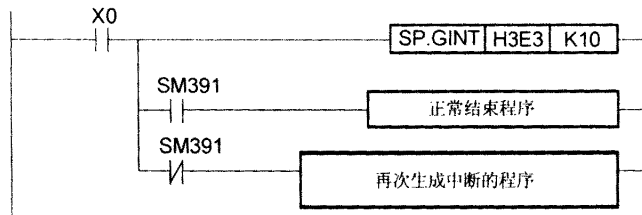
出错标志(SM0)表现在下表中的操作出错状况中, 并且有一个出错代码存放在 SD0 中。

结束状态(注) (出错代码)(H)	原因	解决方法
2110	指定需要通过“(目标 CPU 的起始 I/O 号)/16”设置的 CPU	确认并更正一个程序, 使之来更正 PLC 程序
2114	指定通过“(目标 CPU 的起始 I/O 号)/16”设置的自 CPU	
2117	指定通过“(目标 CPU 的起始 I/O 号)/16”设置的除运动 CPU 之外的 CPU	
4100	指定需要通过“(目标 CPU 起始 I/O 号)/16”设置的 0 到 3DFH 和 3E4H。	
4C08	从 PLC CPU 到运动 CPU 的指令 S(P).SFCS/S(P).SVST/S(P).CHGA/S(P).GINT 总数超过 32 个。运动 CPU 不能处理这些指令。	

(注):0000H (正常)

[程序实例]

运动 CPU No.1 中断的程序



6 运动 SFC 程序

关于运动 SFC 程序错误的细节请参照“18 出错代码列表”。

6.1 运动 SFC 程序结构

运动 SFC 程序由开始，步，转移，结束等组成。如下所示：



上述的即将运行的运动 SFC 程序将运行下列各项操作。

- (1) 步 (F0) 被激活，和步 (F0)一起被指定的操作被运行 (定位就绪)。如此处于有效状态的步叫做有效步。
- (2) 检查和转移 (G0)一起被指定的条件能否满足 (定位程序能否开始)。有效步 (F0) 在条件结束的条件下停止，而且下个步 (K0) 被激活(伺服程序(K0)开始)。
- (3) 检查步 (K0) (伺服程序 K0 的定位结束)的操作结束条件，而且在操作结束(条件结束)条件下控制进入下个步。
- (4) 如上所述随着激活步由(1)到 (3)，控制被运行，并且在步 END 结束。

关于以上运动 SFC 程序执行计时的细节请参照“11.2.2 任务操作”部分。

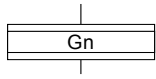
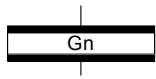
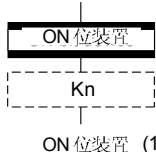
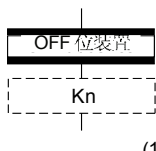
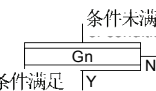
要点
当包括所有的运动 SFC 程序时，同一时间能够保持有效的步数最高可达 256。步数超过 256 将会造成运动 SFC 程序错误 16120。 以下是运动 SFC 程序的各个符号。 F/FS：操作控制步，K：定位控制步，G：判断

6.2 运动 SFC 图表符号列表

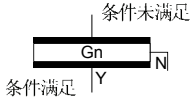
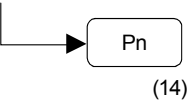
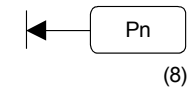
以下是运动 SFC 程序组成部分。在运动 SFC 程序中的操作顺序或转移控制过程由指示线连接起来加以表示。

分类	名称	符号 (代码大小 (字节))	表示列表	功能
程序开始/ 结束	START	 (0)	程序名称	<ul style="list-style-type: none"> 由一个程序名称指示进入程序 在子程序调用时指定这个程序名称 一个程序只有唯一的一个名称
	END	 (8)	END	<ul style="list-style-type: none"> 显示程序结束 当子程序调用执行时返回到调用源程序 多重程序名称或程序没有符号。
步	运动控制步	 (8)	CALL Kn	<ul style="list-style-type: none"> 开始伺服程序(K0 到 K4095)
	一次执行类型操作控制步	 (8)	CALL Fn	<ul style="list-style-type: none"> 执行一次操作控制程序(F0 到 F4095)。
	扫描执行类型操作控制步	 (8)	CALL FSn	<ul style="list-style-type: none"> 重复操作控制程序 FSn(FS0 到 FS4095)直到下个转移条件允许
	子程序调用/ 开始步	 (8)	GSUB 程序名称	<ul style="list-style-type: none"> 当 GSUB 的下一项是 WAIT 的时候，运行“子程序调用”并且转移控制到指定程序。在 END 执行时控制转回到调用源程序。 当 GSUB 的下一项非 WAIT 时，运行“子程序开始”，开始指定程序并且移动到下一项（低级部份）。开始源和最终源程序被同时运行，而且调用最终程序在 END 执行时结束。
清除步	 (8)	CLR 程序名称	<ul style="list-style-type: none"> 停止并且结束指定程序运行。在结束之后，它通过重新启动程序从起始（开始步）开始。 当指定程序是在“子程序调用”期间的时候，子程序也被停止运行 当指定程序是在“子程序开始”之后的时候，子程序不会被停止运行。 当“子程序调用”运行的时候对子程序进行清除，指定子程序被停止运行，回到调用源程序，而且移动到下一项。 	

6 运动 SFC 程序

分类	名称	符号 (代码大小 (字节))	表示列表	功能
转移	转移 (读取前转移)	 (8)	SFT Gn	<ul style="list-style-type: none"> 当前为运动控制步时，通过形成移动条件 Gn(G0 到 G4095)转入下个步而不等待运动操作结束。 当前为操作控制步时，在操作执行之后通过结束转移条件转入下个步。 当前为"子程序调用"或"开始步"时，通过形成移动条件转入下个步而不等待子程序的运行结束
	WAIT	 (8)	WAIT Gn	<ul style="list-style-type: none"> 当前为运动控制步时，等待运动操作结束然后通过结束转移条件 Gn(G0 到 G)转入下个步。 当前为操作控制步时，在操作执行之后通过形成移动条件转入下个步(和移动相同的操作。) 当前为"子程序调用"或"开始步"时，等待子程序操作的结束然后通过结束转移条件转入下个步
	WAIT ON	 ON 位装置 (14)	WAITON 位装置	<ul style="list-style-type: none"> 为下一个运动控制步的开始做准备并且当指定位装置打开时立刻发出一个指令。 总是使该移动和运动控制步成对
	WAIT OFF	 (14)	WAITOFF 位装置	<ul style="list-style-type: none"> 为下一个运动控制步的开始做准备并且当指定位装置关闭时立刻发出一个指令。 总是使该移动和运动控制步成对
	转移 Y/N		IFBm IFT1 SFT Gn : JMP IFEm IFT2 SFT Gn+? : JMP IFEm IFEm	<ul style="list-style-type: none"> 当前为运动控制步时，通过形成移动条件 Gn(G0 到 G4095)转入下个步而不等待运动操作结束。如果没有形成转移条件，转入右方向连接步。 当前为操作控制步时，在操作执行之后通过结束转移条件转入下个步。如果没有形成转移条件，转入右方向连接步。 当前为"子程序调用"或"开始步"时，通过结束移动条件转入下个步而不等待子程序的运行结束。如果没有形成转移条件，转入右方向连接步。

6 运动 SFC 程序

分类	名称	符号 (代码大小 (字节))	表示列表	功能
转移	WAIT Y/N		IFBm IFT1 WAIT Gn : JMP IFE _m IFT2 WAIT Gn+? : JMP IFE _m IFE _m	<ul style="list-style-type: none"> • 当前为运动控制步时, 等待运动运行结束并且如果条件满足 Gn(G0 到 G4095)转入下个步。如果条件不满足, 转入右方向连接步。 • 当前为操作控制步时, 在操作执行之后如果条件满足转入下个步。如果条件不满足, 转入右方向连接步。(和移动相同的操作)。 • 当前为 "子程序调用 "或" 开始步"时, 等待子程序的运行结束, 然后如果条件满足转入下个步。如果条件不满足, 转入右方向连接步。
跳转	跳转		JMP Pn	<ul style="list-style-type: none"> • 跳转到自程序的指定指针 Pn (P0 到 P16383)
指针	指针		Pn	<ul style="list-style-type: none"> • 指示一个跳转目标指针 (标签) • 该指针能设置在步, 转移, 支路点或耦合点 • 能在一个程序设置 P0 到 P16383。相同号码也可能在其他的程序中使用。

6.3 支路和耦合图表目录

以下是支路和耦合模式，在运动 SFC 图表中指定步和转移顺序

名称 (代码大小 (字节))	运动 SFC 图表符号	表示列表	功能
系列转移 (对应的符号大小)		对应运动 SFC 图表符号的表示列表在 6.2 节显示	<ul style="list-style-type: none"> 在系列中连接的步和转移按照从顶端到低端的顺序处理。 步和转移不需要被交替地排列。 当一个转移被省略的时候，执行无条件的移动数据处理。
选择支路 (支路数 + 2) × 10)		CALL Kn IFBm IFT1 SFT Gn CALL Fn : JMP IFEm	<ul style="list-style-type: none"> 在支路前的步或移动被执行后，移动条件首先允许的路径被执行。 选择支路目标应该总是从转移开始，这些必须全部是移动或者 WAIT。（一起使用移动和 WAIT 将会引起一个并联支路。）
选择耦合 (8)		IFT2 SFT Gn' CALL Fn' : (JMP IFEm) IFEm CALL Fn''	<ul style="list-style-type: none"> 在一个选择支路的分枝路径数据处理之后，执行转到一个耦合点。 步或转移可能在一个耦合之前或之后。
并联支路 (支路数 × 22 + 耦合点数 × 2 + 12)		CALL Kn PABm PAT1 CALL Fn SFT Gn' : JMP PAEm	<ul style="list-style-type: none"> 并联连接的多重路径 (步) 被同时的执行。 每个并联支路目标可能由步或转移开始。
并联耦合 (8)		PAT2 CALL Fn' SFT Gn'' : (JMP PAEm) PAEm CALL Fn'' :	<ul style="list-style-type: none"> 在耦合点处等待执行，等待并联支路的支路路径结束执行，并且当所有路径执行结束时转到下一项。 步或转移可能在一个耦合之前或之后。 当该耦合在一个 FS 步之前时，在等候的时候执行扫描。在等待结束时，不执行扫描。
跳转转移 (对应的符号大小)	<正常跳转> <耦合跳转> 	CALL Fn JMP Pn	1)正常跳转 <ul style="list-style-type: none"> 在该跳转转移之前的步或转移被执行后，执行移动到自己程序里面指定的指针 Pn。 跳转目标可能是步或转移。 当从 FS 步到一个转移发生跳转的时候，在等待跳转目标的转移条件结束的时候执行扫描。 2)耦合跳转 <ul style="list-style-type: none"> 当对一个并联支路里面的另一个路径的一次跳转发生在并联支路之后的时候，产生“耦合跳转”并且在跳转目标等待执行。
		CALL Fn' Pn CALL Kn	

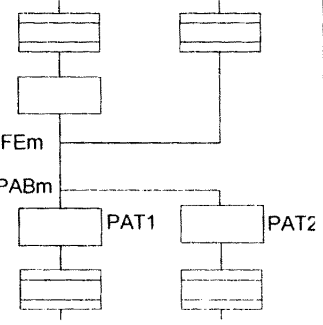
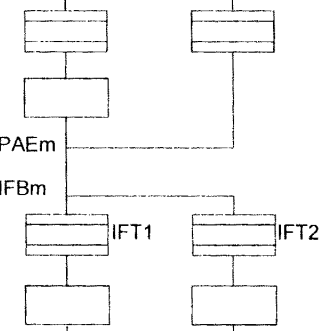
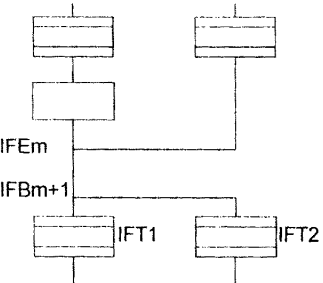
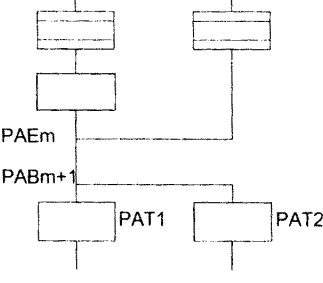
6 运动 SFC 程序

联合基本的类型支路/耦合提供下列各项应用类型，这些类型被定义在基本的类型中。

	名称	运动 SFC 图表符号	表示列表	功能
应用类型	选择支路 并联支路		CALL Kn IFBm IFT1 SFT Gn PABm PAT1 CALL Fn :	<ul style="list-style-type: none"> 在一个选择支路之后，能运行一个并联支路。
	并联耦合 选择耦合		JMP PAEm PAT2 CALL Fn' : (JMP PAEm) PAEm JMP IFE m IFT2 SFT Gn' CALL Fn' : (JMP IFE m) IFE m SFT Gn''	<ul style="list-style-type: none"> 选择的耦合点能与选择支路→并联支路的并联耦合的耦合点相同。在运动 SFC 图表中注意，这个类型以并联耦合→选择耦合的顺序显示，如左图所示。 在这种情况下，指针 (Pn) 不能够设置在并联耦合点 (PAEm) 和选择的耦合点 (IFE m) 之间。
	并联支路 选择支路		SFT Gn PABm PAT1 CALL Fn IFBm IFT1 SFT Gn' CALL Fn' : JMP IFE m IFT2 SFT Gn'' CALL Fn'' : (JMP IFE m) IFE m JMP PAEm PAT2 CALL Fn'' : CALL Kn (JMP PAEm) PAEm SFT Gn'''	<ul style="list-style-type: none"> 在一个并联支路之后，能运行一个选择支路。
	选择耦合 并联耦合		JMP IFE m IFT2 SFT Gn'' CALL Fn'' : (JMP IFE m) IFE m JMP PAEm PAT2 CALL Fn'' : CALL Kn (JMP PAEm) PAEm SFT Gn'''	<ul style="list-style-type: none"> 并联耦合点能与并联支路→选择支路的选择耦合的耦合点相同。在运动 SFC 图表中注意，这个类型以选择耦合→并联耦合的顺序显示，如左图所示。 在这种情况下，指针 (Pn) 不能够设置在选择耦合点 (IFE m) 和并联耦合点 (PAEm) 之间。

6 运动 SFC 程序

	名称	SFC 图表符号	表示列表	功能
应用类型	选择支路 选择支路		CALL Kn IFBm IFT1 SFT Gn IFBm+1 IFT1 SFT Gn' : JMP IFE _{m+1}	<ul style="list-style-type: none"> • 在一个选择支路之后, 能运行一个选择支路。
	选择耦合 选择耦合		IFE _{m+1} SFT Gn" : (JMP IFE _{m+1}) IFE _{m+1} JMP IFE _m IFE _m SFT Gn" CALL Fn' : (JMP IFE _m) IFE _m SFT Gn" :	<ul style="list-style-type: none"> • 选择支路→选择支路的二个选择性耦合点可能是相同的。在运动 SFC 图表中注意, 这个类型以选择耦合→选择耦合的顺序显示, 如左图所示。 • 在这种情况下, 指针 (Pn) 不能够设置在选择耦合点 (IFE_{m+1})和选择耦合点(IFE_m)之间
	并联支路 并联支路		CALL Kn PABm PAT1 SFT Gn PABm+1 PAT1 CALL Fn' : JMP PAE _{m+1}	<ul style="list-style-type: none"> • 在一个并联支路之后, 一个并联支路能被运行。 • 一个并联支路能被嵌套到四个级别。
	并联耦合 并联耦合		PAT2 CALL Fn" : (JMP PAE _{m+1}) PAE _{m+1} JMP PAE _m PAT2 CALL Fn" : CALL Kn JMP PAE _m PAE _m SFT Gn" :	<ul style="list-style-type: none"> • 并联支路-并联支路的那两个并联耦合点可能是相同的。在运动 SFC 图表中注意, 这个类型以并联耦合→并联耦合的顺序显示, 如左图所示。 • 在这种情况下, 指针 (Pn) 不能够设置在并联耦合点 (PAE_{m+1})和并联耦合点(PAE_m)之间。

	名称	SFC 图表符号	表示列表	功能
应用 类型	选择耦合 并联支路		: (JMP IFE m) IFE m PAB m PAT 1 CALL Fn : JMP PAE m PAT 2 CALL Fn' : (JMP PAE m) PAE m :	<ul style="list-style-type: none"> 选择的耦合点和并联支路点可能是相同的。在运动 SFC 图表中注意, 这个类型以选择耦合 并联耦合的顺序显示, 如左图所示。 在这种情况下, 指针 (Pn) 不能够设置在选择耦合点 (IFE m 和并联支路点(PAB m)之间)
	并联耦合 选择支路		JMP PAE m PAE m IFB m IFT 1 SFT Gn : JMP IFE m IFT 2 SFT Gn' : (JMP IFE m) IFE m :	<ul style="list-style-type: none"> 并联耦合点和选择支路点可能是相同的。在运动 SFC 图表中注意, 这个类型以并联耦合 选择耦合的顺序显示, 如左图所示。 执行在并联耦合点处等待, 并且转到选择支路。 在这种情况下, 指针 (Pn) 不能够设置在并联耦合点 (PAE m)和选择支路点(IFB m)之间
	选择耦合 选择支路		: (JMP IFE m) IFE m IFB m+1 IFT 1 SFT Gn : JMP IFE m+1 IFT 2 SFT Gn' : (JMP IFE m+1) IFE m+1 :	<ul style="list-style-type: none"> 选择的耦合点和选择支路点可能是相同的。在运动 SFC 图表中注意, 这个类型以选择耦合 选择耦合的顺序显示, 如左图所示。 在这种情况下, 指针 (Pn) 不能够设置在选择耦合点 (IFE m)和选择支路点(IFB m+1)之间。
	并联耦合 并联支路		(JMP PAE m) PAE m PAB m+1 PAT 1 CALL Fn : JMP PAE m+1 PAT 2 CALL Fn' : (JMP PAE m+1) PAE m+1 :	<ul style="list-style-type: none"> 并联耦合点和并联支路点可能是相同的。在运动 SFC 图表中注意, 这个类型以并联耦合 并联耦合的顺序显示, 如左图所示。 执行在并联耦合点处等待并且转到并联支路。 在这种情况下, 指针 (Pn) 不能够设置在并联耦合点 (PAE m)和并联支路点(PAB m+1)之间

6.4 运动 SFC 程序名称

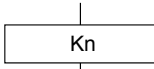
从 No.0 到 No.255 中设置“运动 SFC 程序名称”到运动 SFC 程序 (在运动 SFC 程序编辑屏上的“运动 SFC 程序管理窗”中进行这一设置。)

在 16 个字里面设置运动 SFC 程序名称。在“子程序调用 / 开始步(GSUB)”和“清除步 (CLR)”指定运动 SFC 程序名称。No.0 到 No.255 中运动 SFC 程序是以一个程序为一个文件格式保存的。预先设置的“运动 SFC 程序名称”被用做用户文件管理的运动 SFC 程序文件名称。(有关细节参照章“12 用户文件”)

要点
(1) 运动 SFC 程序能够被设置到 No.0 到 No.255 之中的任何一个。没有特殊程序有特别的作用。
(2) "\$"不能用作运动 SFC 程序名称的第一个字。
(3) 运动 SFC 程序名称中不能使用"/:; , . * ? " < > "

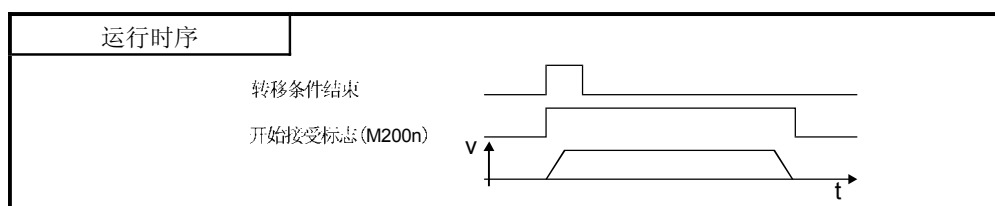
6.5 步

6.5.1 运动控制步

名称	符号	功能
运动控制步		开始伺服程序 Kn。 指定范围: 从 K0 到 K4095

[操作]

- (1) 打开指定轴的开始接收标志, 运行指定的伺服程序 Kn(n=0 到 4095)。
- (2) 开始运行伺服程序 Kn(n=0 到 4095)。



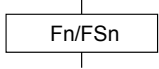
[错误]

- (1) 当指定伺服程序 Kn 不存在的时候, 将会发生运动 SFC 程序错误 [16200]并且在错误检测时停止执行运动 SFC 程序。

[指令]

- (1) 当在运行的运动 SFC 程序中更改当前值时, 在伺服程序中指定 CHGA 指令并且在运动控制步中调用。
- (2) 如果由于和运动控制步一起被指定的伺服程序的开始或开始阶段发生主要的或次要的错误, 伺服程序停止, 运动 SFC 程序继续运行。当运动 SFC 程序在错误检测时停止的时候, 在转移(转移条件)中提供错误检测条件。

6.5.2 操作控制步

名称	符号	功能
操作控制步		运行操作控制程序 Fn/FSn。 指定范围: F0 到 F4095/FS0 到 FS4095

[操作]

- (1) 一次执行类型操作控制步 Fn
在 Fn 的情况下，运行指定操作控制程序 Fn(n=0 到 4095)一次。
- (2) 扫描执行类型操作控制步 FSn
在 Fn 的情况下，重复指定操作控制程序 FSn(n=0 到 4095)直到下个转移条件允许。

[错误]

- (1) 当指定操作控制程序 Fn/ FSn 不存在的时候，将会发生运动 SFC 程序错误 [16201] 而且在错误检测时停止运行运动 SFC 程序。

[指令]

- (1) 关于可能在运行控制程序中描述的操作表达式请参照章“7 操作控制程序”
- (2) 当操作控制程序运行时如果操作或相似的错误发生, 运动 SFC 程序继续运行。

6 运动 SFC 程序

6.5.3 子程序调用/启动步

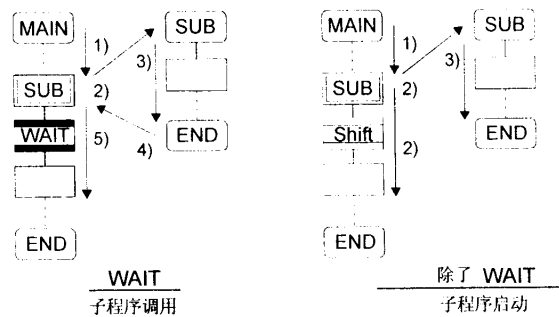
名称	符号	功能
子程序调用/ 启动步	程序名称	调用/启动指定程序名称的运动 SFC 程序。

[操作]

- (1) 调用/开始指定程序名称的运动 SFC 程序。
- (2) 控制因子程序调用/开始步连接的转移类型而改变。
 - (a) WAIT (子程序调用)

当子程序调用步运行的时候，控制移动到指定程序，如下图所示，而且当被调用程序的 END 被运行的时候，控制回到调用源程序。
 - (b) 不包括WAIT (子程序启动)

当子程序开始步被运行的时候，控制开始指定程序然后移动到下一项。如下图因为开始源和目标运动 SFC 程序被并联运行。被开始的程序在 END 执行结束。



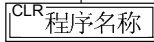
[错误]

- (1) 当指定运动 SFC 程序在子程序调用/启动中不存在的时候，将会发生运动 SFC 程序错误 [16005] 而且在错误检测时停止运行运动 SFC 程序。
- (2) 当调用/启动运动 SFC 程序已经在子程序调用/启动中开始，将会发生运动 SFC 程序错误 [16006] 而且在错误检测时停止运行运动 SFC 程序。
- (3) 当在一个子程序调用/启动中开始自身程序，将会发生运动 SFC 程序错误 [16110]，而且在错误检测时停止运行运动 SFC 程序。
- (4) 运行中的运动 SFC 程序 2 被运动 SFC 程序 1 调用/启动，同时运行中的运动 SFC 程序 2 一个子程序调用/启动的时候，即将被调用/启动的子程序是运动 SFC 程序 1(调用源/启动程序)，将会发生运动 SFC 程序错误 [16111]，而且运行中的调用/启动源运动 SFC 程序 2 在错误检测点停止。

[指令]

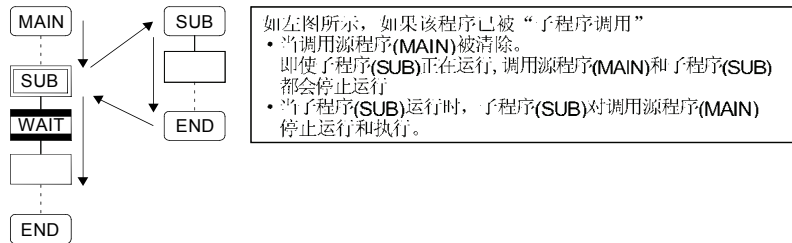
- (1) 在子程序调用 / 启动嵌套的深度上没有限制。
- (2) 对于一个子程序开始，如果开始目标运动 SFC 程序由于一个错误停止，开始源运动 SFC 程序继续数据处理。
- (3) 对于一个子程序调用，一旦调用目标运动 SFC 程序由于一个错误停止，调用源运动 SFC 程序停止运行。

6.5.4 清除步

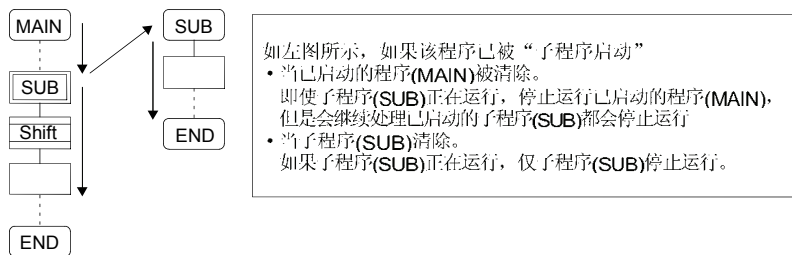
名称	符号	功能
清除步		停止指定程序名称的运动 SFC 程序。

[操作]

- (1) 停止运行的指定运动 SFC 程序。
- (2) 即便被设置为自动开始，指定清除的运动 SFC 程序停止后不会自动开始。
- (3) 指定程序可能是它的本身程序。
- (4) 如果指定程序是被调用的子程序，被调用的子程序也会被停止。



- (5) (在下面显示)



- (6) 当从指定程序开始的伺服程序正在开始的时候，伺服程序继续数据处理。

[错误]

- (1) 当和清除步一起被指定的运动 SFC 程序不存在的时候，将会发生运动 SFC 程序错误 [16203]。

[指令]

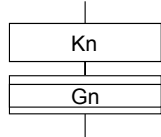
- (1) 当和清除步一起被指定的运动 SFC 程序没有开始的时候，将不会发生错误，而且这个步会被忽略。
- (2) 如果运行的运动 SFC 程序被清除步停止，输出将被保留。

6.6 转移

你可以在转移中描述条件表达式和操作表达式。这里描述的操作表达式将被重复，运行直到转移条件允许的时候，正如在扫描执行类型操作步时那样。
 关于能够在转移条件中描述的条件表达式 / 操作表达式，参照章节“8 转移程序”。

(1) 运动控制步的组合

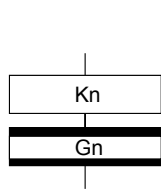
(a) 运动控制步 +Shift(转移)



[操作]

- 通过形成转移条件 Gn 移动到下个步，而不需等待在运动控制步开始的伺服程序 Kn 的操作结束

(b) 运动控制步+WAIT(等待)

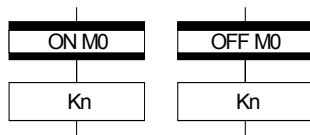


[操作]

- 等待在运动控制步开始的伺服程序 Kn 的操作结束，然后通过形成转移条件 Gn 转入到下个步。
- 在转移条件 Gn 中不需要伺服程序 Kn 的操作结束条件。开始时或在开始期间开始的伺服程序 Kn 的一个故障停止也被视为一个操作结束。

(c) WAITON/ WAITOFF+运动控制步

[操作]



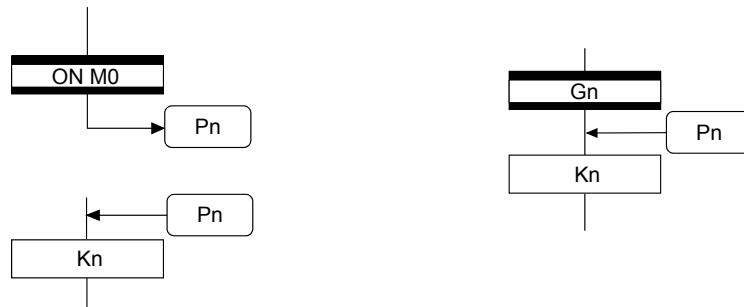
- 为紧邻 WAITON/ WAITOFF 的运动控制步的开始作准备，而且当指定位软元件转入 ON/OFF 时立刻开始。当运动控制步执行时没有被 WAITON/WAITOFF 所使用并且在运动控制步之前的转移条件允许后，将为开始做准备。这将会引起在转移条件结束时和开始时之间的延迟/开始时间的变化，但是和 WAITON/ WAITOFF 的组合能够消除上述的延迟/开始时间的变化。

• 指定的位软元件

软元件	范围
X	X0 到 X1FFF
Y	Y0 到 Y1FFF
M	M0 到 M819
特殊 M	M9000 到 M9255
L	L0 到 L8191
B	B0 到 B1FFF
F	F0 到 F2047

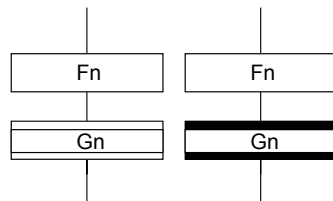
[指令]

- 一定要使转移和运动控制步成对使用。如果紧接 WAITON/ WAITOFF 的步不是一个运动控制步时，将会发生运动 SFC 程序错误 [16102]，而且运行的运动 SFC 程将会在错误检测时停止。
- 如果在 WAITON/ WAITOFF 紧接之后的跳转目标是一个运动控制步，将不发生错误(左下)
- WAITON/WAITOFF 紧接之后可能是一个指针 (右下)



- 如果和运动控制步一起被指定的伺服程序因为主要的 / 次要的错误不能开始，运动 SFC 程序继续进行，而且执行移动到下一项，独立于 WAITON/ WAITOFF 位装置状态之外。想在错误检测时停止运动 SFC 程序，在下个转移(转移条件)中提供错误检测条件。
- 下列各项指令能够用在运动控制步和WAITON/WAITOFF组合使用中 (直线插补法控制，弧线插补法控制，螺旋插补，速度切换控制，定位跟踪控制，恒速控制和高速振荡。)

(2) 和操作控制步的组合



[操作]

- 在操作控制步中，Shift 和 WAIT 都运行相同的操作，在执行操作控制程序 Fn 之后，通过形成转移条件 Gn 进入下个步。

(3) 和子程序调用 / 启动步的组合

参照章节“6. 5. 3 子程序调用 / 启动步”。

6.7 跳转, 指针

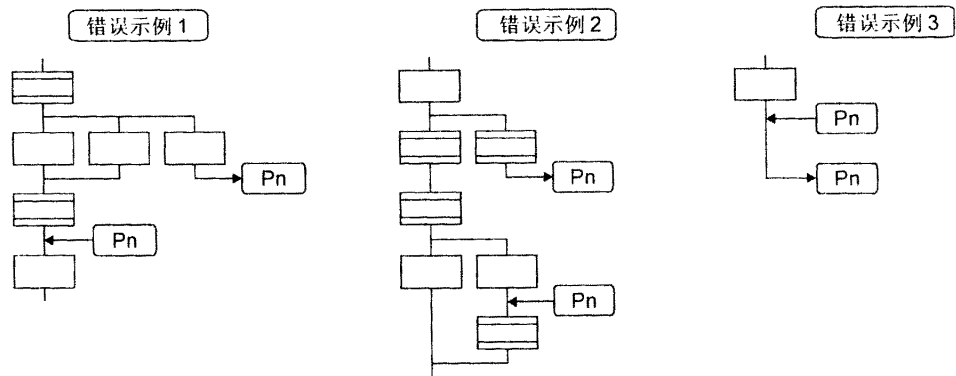


[操作]

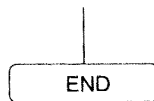
- 设置跳转将会使程序跳转到指定指针 Pn。
- 你可以在步, 转移, 支路点和耦合点设置指针。
- 你可以在一个程序里面从 P0 到 P16383 范围内设置指针 Pn。

[指令]

- 不能够做出这样一个跳转设置, 使得跳转将在并联支路 - 并联耦合里退出。直接连接。(下面是错误示例 1)
- 不能够从并联支路- 并联耦合之外到并联支路- 并联耦合之内作跳转设置 (下面是错误示例 2)
- 不能够在标记和跳转即将继续的地方进行跳转设置。(下面是错误示例 3)



6.8 END



[操作]

- 结束程序。(在事件任务或 NMI 任务的情况下, 操作随着程序参数的结束操作设置而改变。关于细节参照章节“11.5 程序参数”。)
- 作一个子程序调用将会回到调用源运动 SFC 程序

[指令]

- 在一个程序里面 END 可以被多次设置。
- END 不能被设置在一个并联支路和一个并联耦合之间。
- 在运动 SFC 程序以 END 结束之后, 输出会被保持。

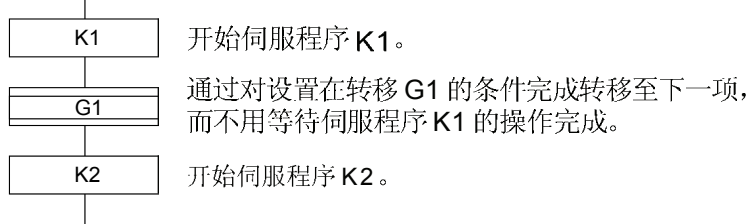
6.9 支路，耦合

6.9.1 串联转移

转移到随后的串联步或转移。

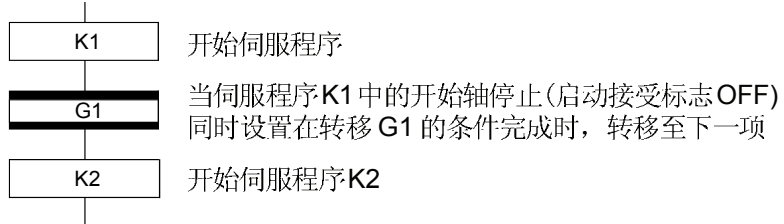
- (1) 不等待操作结束而开始执行伺服程序或子程序并且移动到下一项在转移中设置移动。

在这种情况下, 转移(移动)可能被省略。当你省略了转移的时候, 将执行一个无条件的移动转移



要点	对于一个子程序开始, 自程序和子程序在并联中被数据处理。
----	------------------------------

- (2) 开始伺服程序或子程序而且在操作结束时进入下个步在一个转移中设置 WAIT。



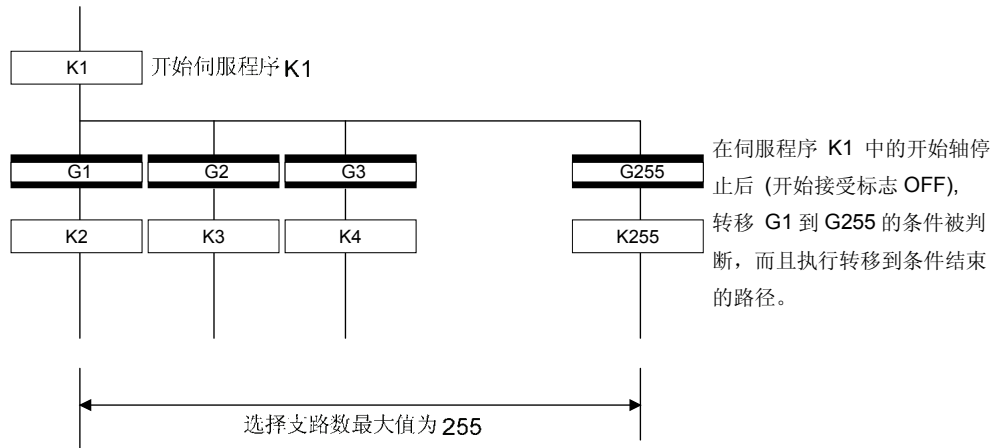
要点	<p>(1) 上述在下一伺服程序 K2 中开始的轴的开始接受标志将不被包含在联动装置中。为了要把它作为联锁软元件使用, 用户应该在转移条件 G1 中设置它。</p> <p>(2) WAIT 必须被设定进行到当操作结束时的下个步。然而, 当没有可特殊地被设置为联动装置的条件时, 在转移程序(Gn)中设置 "NOP(空操作)"。</p>
----	---

6.9.2 选择支路，选择耦合

(1) 选择支路

在并联中被连接的多重转移的条件之中，只执行首先被判断为条件允许的路径。转移必须是所有的 Shift 或 WAIT。

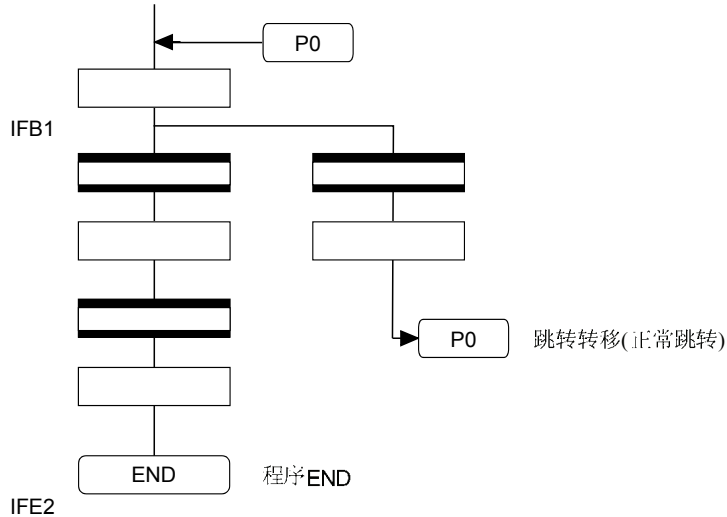
(例)WAIT



要点
(1) 转移条件判断不总是被从左到右运行。
(2) 同时使用 Shift 和 WAIT 将导致并联支路。

(2) 选择耦合

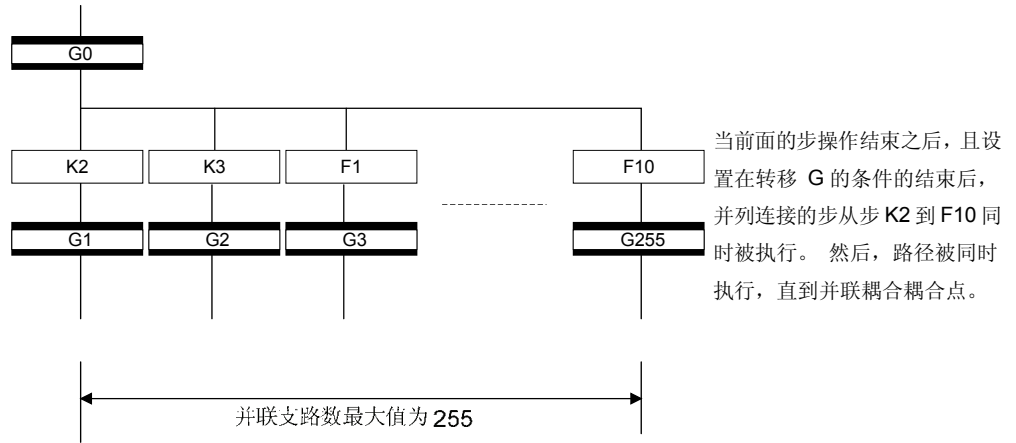
在一条选择支路之后，将多个路径数据处理结束后将这些路径重新耦合到一条路径，将会得到一个选择的耦合。然而，你也可以设置不作任何耦合，如下图所示。



6.9.3 并列支路，并列耦合

(1) 并列支路

并联连接的多重路径被同时运行。每个并联支路目标可以以步或转移开始。

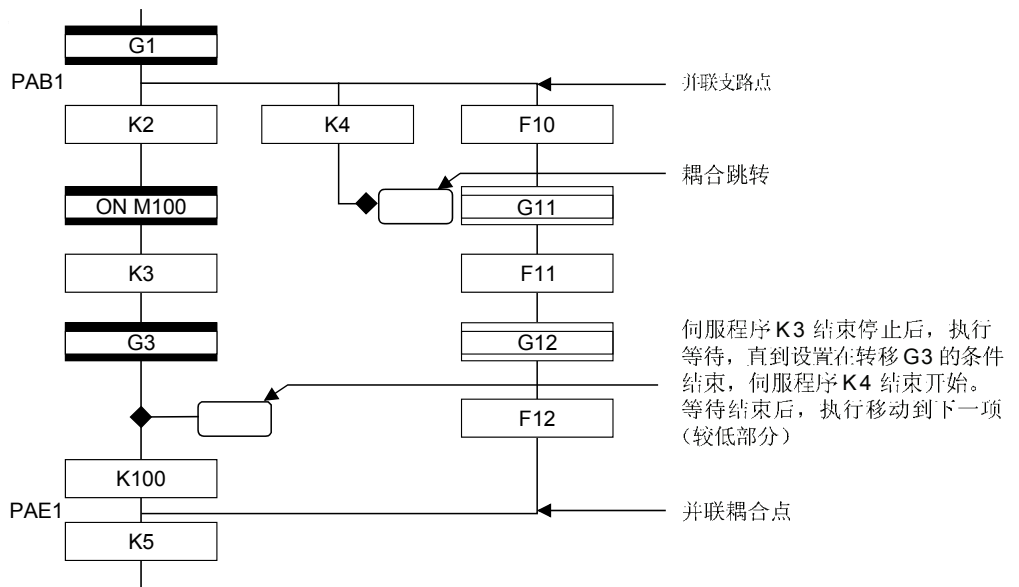


要点
 "Shift" 或 "WAIT" 能被设置到并联支路前的转移。
 "WAITON" 和 "WAITOFF" 不能被设置。

(2) 并联耦合

一个并联支路必须被一个并联耦合耦合。在并联支路-并联耦合里面能够将跳转设置到另外的一条支路路径。在这种情况下，跳转目标是中间的并联耦合点(耦合跳转)。

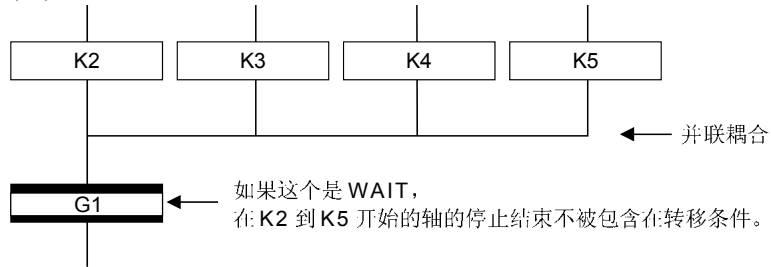
不能够设置跳转使其从并联支路-并联耦合里面退出。



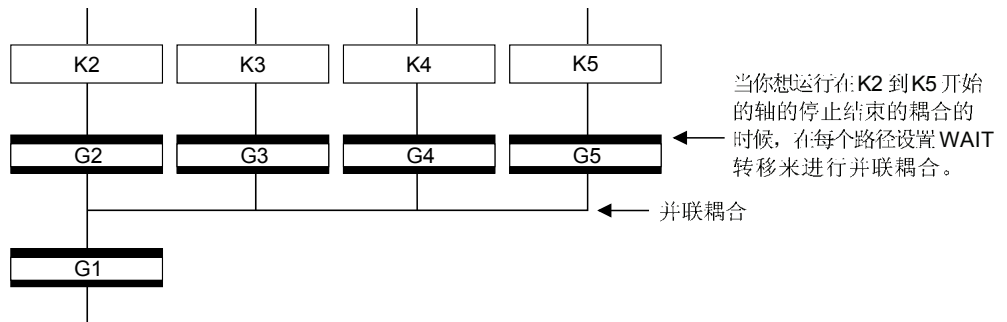
要点
并联支路数不需要与在并联耦合点的耦合数相匹配。 (在 6.9.3 (2) 节的图表例子中，并联支路数是 3 而耦合数是 2。)

在一个并联耦合后，将 WAIT 转移设置在右边，且如果并联耦合在运动控制步之前，轴的停止结束不包含在等待条件中。为了要运行停止结束上的一个并联耦合，在并联耦合之前设置 WAIT 转移。

(×)



(○)

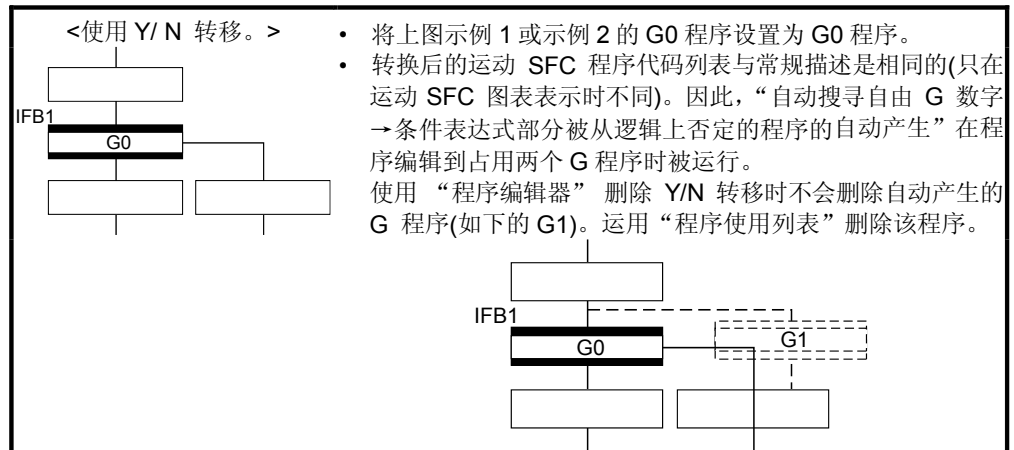
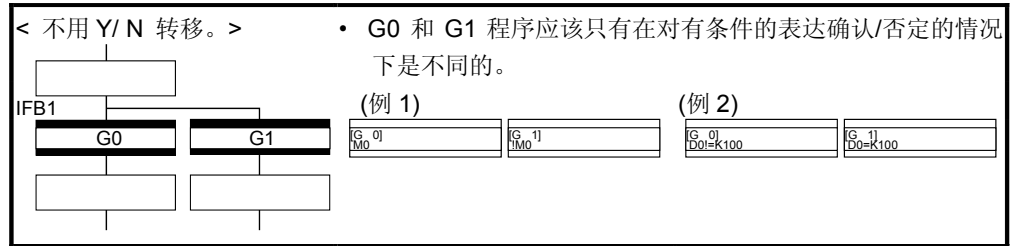


6.10 Y/N 转移

在转移条件允许或禁止的时候路径分支中，可使用“Shift Y/N 转移”或“WAIT Y/N 转移”。

名称	符号	功能
Shift Y/N 转移		<ul style="list-style-type: none"> 当设置在 Gn 的转移条件允许时，执行移动到较低的步。当条件禁止时，执行移动到右向连接步。 在 "Shift Y/N" 和 "WAIT Y/N" 之间的差异与 "Shift" 和 "WAIT" 之间的差异是一样的。
WAIT Y/N 转移		

一个 Y/ N 转移的使用，以便容易地描述下面的二个路径的选择支路程序。



(1) 自由 G 数字自动搜寻特征

(a) 当没有设置自动编号时

向前搜寻自由数字, 在“Shift Y/N”或“WAIT Y/N”符号处以“设置 G 数字 +1”开始。

当搜寻直到 4095 后, 没有发现自由数字, 将从 0 到“设置 G 数字 - 1”进行搜寻。

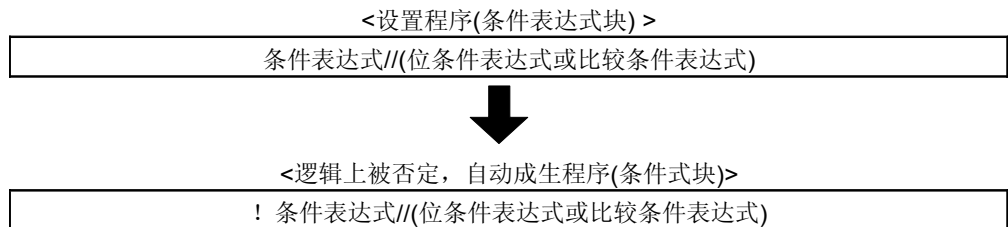
(b) 当设置自动编号时

在自动编号范围内向前(或向后)搜寻自由数字, 在“Shift Y/N”或“WAIT Y/N”符号处以“自动编号 G 数字 +1 (或 -1)”开始。(搜索的方法与在自动编号设置中的方法相同。)

(2) 自动的逻辑非程序生成特征

自动地生成一个程序, 在逻辑上否定设置在“Shift Y/N”或“WAIT Y/N”转移程序的条件式块 (最后的块)。

基本内容如下所示。



示例如下。

<设置程序(条件表达式块)>

(例 1)

M0 //位软元件 ON

(例 2)

D0!=K100 //数据记录 D0 不是 K100

<逻辑上被否定, 自动成生程序(条件表达式块)>

(例 1)

!(M0) //位软元件 OFF

(例 2)

!(D0!=K100) //数据记录 D0 是 K100

要点

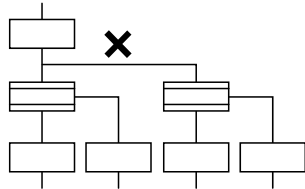
参阅章节 "1.2.1(2) 操作控制/转移指令列表", 查询在 "Shift Y/N" 或 "WAIT Y/N" 转移程序中的条件式中的可用指令。

(3) 运动 SFC 图表的指令

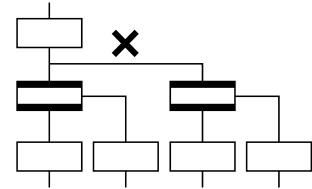
无意义或与Y/N 转移的定义相冲突的任何运动 SFC 图表都将会在编辑(或运动SFC 图表转移)的时候发生错误。如下示例将给出它们的模式和指令。

(a) 当“移动 Y/N”或“WAIT Y/N”被连接为选择支路或并联支路时：错误

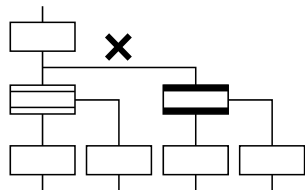
- "Shift Y/N"被用作选择支路



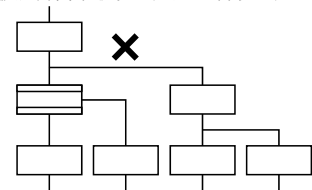
- "WAIT Y/N"被用作选择支路



- "Shift Y/N"和" WAIT Y/N"被用作并联路

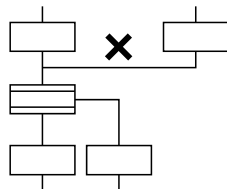


- "Shift (或 WAIT) Y/N"和其他步/转移一起被用作并联支路或选择支路

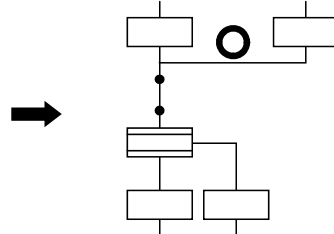


(b) 当耦合在“Shift Y/N”或“WAIT Y/N”之前的时候: 在中间提供“耦合-支路连续”

- 不允许和 "Shift Y/N" 或 " WAIT Y/N" 直接耦合

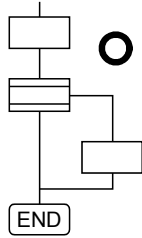


- 在中间提供 "耦合-支路连续"

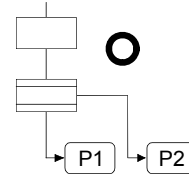


(c) 以下模式可被设置。

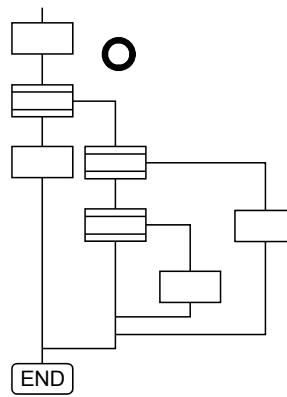
- 从 "Shift Y/N" 或 "WAIT Y/N"结束 (END)



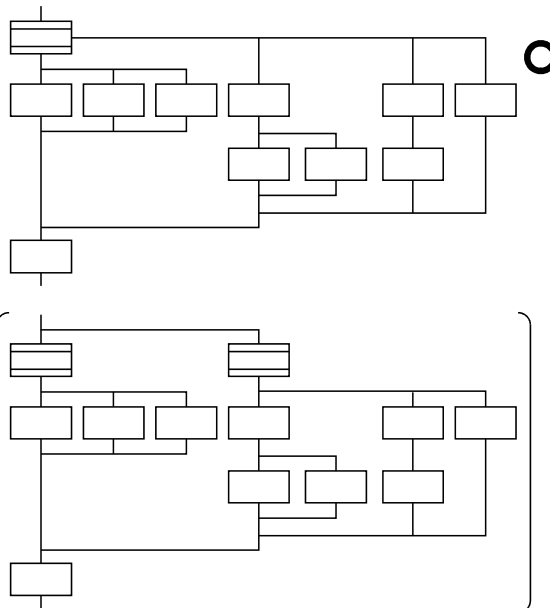
- 从 "Shift Y/N" 或 "WAIT Y/N"跳转



- 从 "Shift Y/N" 或 "WAIT Y/N"到 "Shift Y/N" 或 "WAIT Y/N"的连续 (选择支路-选择支路)



- 当从 "Shift Y/N" 或 "WAIT Y/N"的 Y/N 一方存在两条或两条以上的连接线时, 选择支路连续到选择支路或并联支路



6 运动 SFC 程序

6.11 运动 SFC 注释

运动 SFC 图表中的步 / 转移的每个符号都能被设置注释。通过在运动 SFC 程序编辑屏上改变到“注释显示”的显示方式，在运动 SFC 图表中显示注释。
 因为运动 SFC 注释被存储在 CPU 代码区，从 PC 进行读取会显示运动 SFC 图表及注释

分类	名称	符号	注释设置
程序开始/结束	START		不能进行注释设置。
	END		
步	运动控制步		最多 80 字 以 20 字×4 行显示
	一次执行类型操作控制步		
	扫描执行类型操作控制步		
	子程序调用/启动步		
	清除步		
转移	Shift(预读转移)		最多 80 字 以 20 字×4 行显示
	WAIT		
	WAITON		
	WAITOFF		
	Shift Y/N		
	WAIT Y/N		
跳转	Jump		最多 64 字 以 16 字×4 行显示
指针	Pointer		

要点
(1) 运动 SFC 注释被存储在 CPU 代码区。CPU 代码区存储运动 SFC 图表代码，操作控制(F/FS)程序代码，转移(G)程序代码和运动 SFC 注释。注意不要设置过多注释以避免代码区溢出。代码区大小请参阅章节"1.2.2 (2) 运动 SFC 性能说明"
(2) 在注释陈述中不能使用", "

7. 操作控制程序

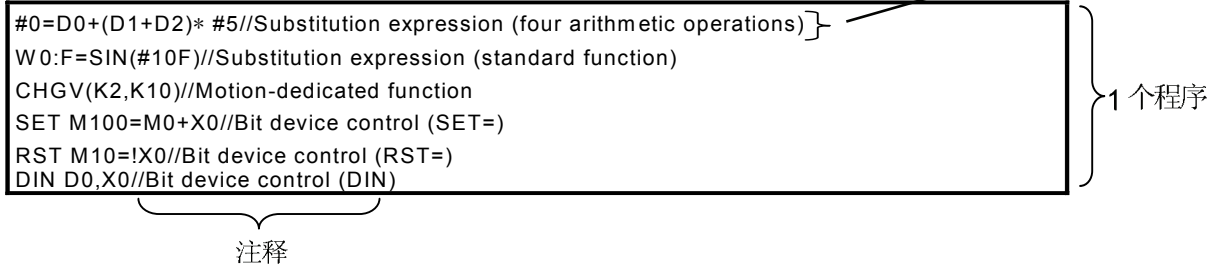
参考“18.2 运动 SFC 错误代码列表”一栏中运算出错的错误代码。
(参考“Q173CPU(N)/Q172CPU(N)运动控制器(SV13/SV22)编程手册(实模式)”和“Q173CPU(N)/Q172CPU(N)运动控制器(SV22)编程手册(虚模式)”运算出错中的轻度出错。)

7.1 操作控制程序

(1) 操作控制程序

- (a) 替换运算公式，运动专用功能和位软元件控制指令可在运算控制程序中设定。
- (b) 可以设定一个运算控制程序中的多重块。
- (c) 一个运算控制程序中的块数没有限制。但是，一个程序必须在 64K 字节以内。
- (d) 一个块中的字最大数目为 128 个。
- (e) 不能设定转移条件，转移条件只能在转移程序中设定。
- (f) 在运算控制程序中，返回位条件表达式的逻辑值(I/O)，比较条件公式仅可作为源软元件进行软元件设定(SET=1)或软元件复位(RST=)。

运算控制程序示例如下所示：



(2) 操作和功能的优先级

操作和功能有如下的优先级：

使用圆括号允许自由指定运算顺控程序。

优先级	项目 (运算, 功能)	
高	圆括号内的计算(...)	
↑	标准功能 (SIN, COS 等等)	
	类型转换 (USHORT, LONG 等等)	
	位变换(~), 逻辑求反(!), 符号变换(-)	
	乘 (*), 除(/), 剩余(%)	
	加(+), 减(-)	
	位左移(<<), 位右移(>>)	
	比较运算: 小于(<), 小于或等于(<=), 大于 (>), 大于或等于(>=)	
	比较运算: 等于(==), 不等于(!=)	
	位逻辑 AND (&)	
	位异或 OR (^)	
	位逻辑 OR ()	
	逻辑 AND (*)	
	逻辑 OR (+)	
	替换 (=)	
	低	
	↓	

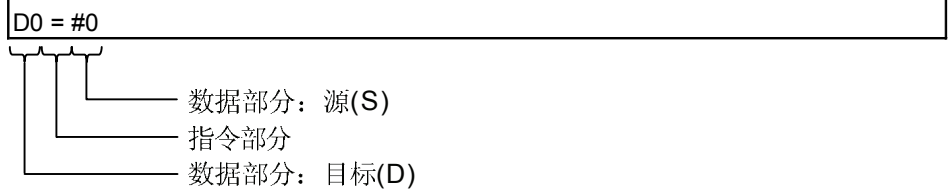
(3) 指令结构

操作控制程序中可使用的许多指令可分为指令部分和数据部分。

这些指令和数据部分用于下列目的。

- 指令部分.....说明指令的功能。
- 数据部分.....说明指令中使用的数据。

“替换：=”结构示例



(a) 源(S)

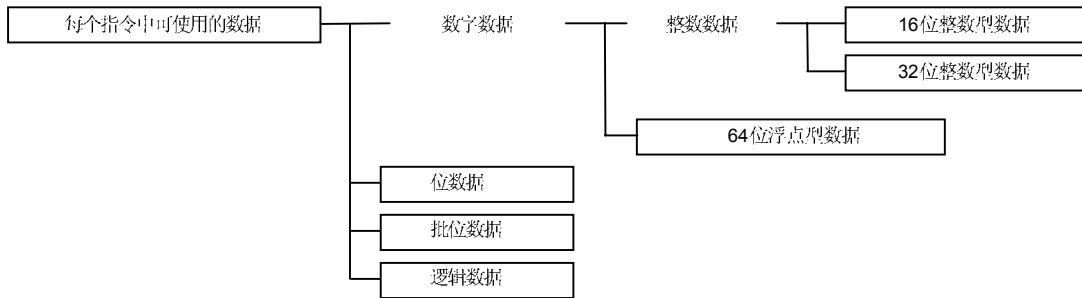
- 1) 源就是运算中使用的数据。
- 2) 它随下图所示每个指令中指定的软元件而不同：
 - 位或字软元件
指定用于存储运算中使用的数据的软元件。
进行运算之前，数据必须存储在指定的软元件中。
在执行程序时，改变允许在那个指令中允许改变数据的软元件中的数据。
 - 常数
指定运算中使用的数值。
因为程序编写时已设定为常数，所以程序运行时它不能改变。

(b) 目标(D)

- 1) 运算后的数据存储为目标数据。
- 2) 目标数据总是设定软元件用于存储数据。

(4) 如何指定数据

下列是每个指令中可使用的六种不同的数据



(a) 16位整数型数据

16 位整数型数据即 16 位整数值数据。
字软元件使用 1 点的增量。
数据范围如下图所示：

	十进制表示	十六进制表示
数据范围	K-32768 到 K32767	H0000 到 HFFFF

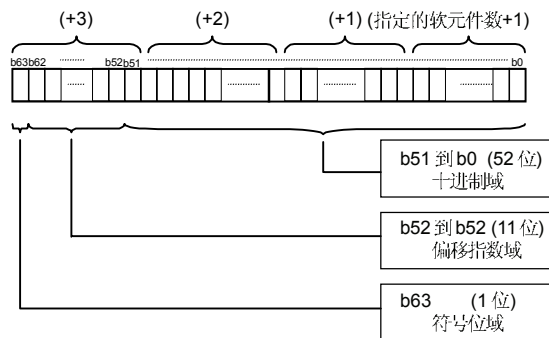
(b) 32位整数型数据

32 位整数型数据即 32 位整数值数据。
字软元件使用 2 点的增量；(指定的软元件号)
(指定的软元件号 +1) 数据范围如下图所示：

	十进制表示	H 十六进制表示
数据范围	K-2147483648L 到 K2147483647L	H00000000L 到 HFFFFFFFL

(c) 64位浮点型数据

64 位浮点型数据即 IEEE 格式化的，64 位浮点值数据。
字软元件使用 4 点的增量；(指定的软元件号)
(指定的软元件号 +1)(指定的软元件号 +2)(指定的软元件号 +3)。
1) 内部位置如下图所示



2) 代表的值如下图所示：(偏移值为 H3FF)

$$(-1)^{[\text{符号位域}]} * (1.0 + [\text{十进制域}]) * 2^{([\text{偏移指数域}] - [\text{偏移值}])}$$

3) 数据范围如下图所示

	十进制表示	十六进制表示
数据范围	K-1.79E+308 到 K-2.23E-308, K0.0, K2.23E-308 到 K1.79E+308	H0000000000000000 H0010000000000000 到 H7FE1CCF385EBC89F, H8000000000000000, H8010000000000000 到 HFFE1CCF385EBC89F

4) 64 位浮点型数据运算中可能会产生舍入误差。

特别是在比较操作中使用 64 位浮点型数据时，请注意舍入误差可能会产生期望的操作。

示例) 在下面的转换程序中，由于有舍入误差，所以根据 #200F 的值得到的比较运算的结果可能会不真实。

```
#100F=SQRT(#200F)
#300F=#100F*#100F
#200F==#300F
```

(d) 位数据

位数据即 1 位的增量中处理触点 / 线圈或相似软元件的数据。它用于软元件设定 (SET=) 和软元件复位 (RST=)

示例 1

```
SET M0
```

位数据

(e) 批位数据

批位数据即在 16/32 的增量中处理位数据的数据。

它用于软元件输入 (DIN) 和软元件输出 (DOUT)。

如下图所示是在 16 点还是在 32 点的增量中处理位数据是受到用作输入目标 / 输出源的软元件的数据类型支配的。

	16 点增量	32 点增量
程序示例	DIN #0, M0 DOUT M0, D0	DIN #0L, M0 DOUT M0, D0L
使用的软元件	(指定的软元件号)到 (指定的软元件号+15) 在上述程序示例中为 M0 到 M15	(指定的软元件号)到 (指定的软元件号+31) 在上述程序示例中为 M0 到 M31

(f) 逻辑数据

逻辑数据是由位或比较条件表达式返回的值并表明结果是真还是假。通常，它在转换程序的条件表达式中使用。在运算控制程序中，逻辑数据在位条件公式中可使用软元件设定(SET=)或软元件复位(RST=)。

例 1

```
SET M0 = X10
```

逻辑数据
位数据

例 2

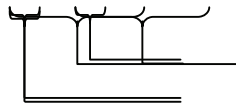
```
SET M5 = ! X10 * M100
```

逻辑数据
位数据

例 3(转移程序)

```
D0==K100
```

逻辑数据



7 操作控制程序

7.2 软元件说明

字和位软元件说明如下所示：

(1) 字软元件说明

	软元件说明			软元件号(n)的指定范围
	16 位整数型	32 位整数型 (“n”为偶数)	64 位浮点型 (“n”为偶数)	
数据寄存器	Dn	DnL	DnF	0 到 8191
链接寄存器	Wn	WnL	Wn:F	0 到 1FFF
特殊寄存器	Dn	DnL	DnF	9000 到 9255
运动软元件	#n	#nL	#nF	0 到 8191 (运动 SFC 专用软元件： 8000 到 8191)
惯性定时器	—	FT	—	—

- (a) 为了区别，32 位浮点型由 L 终止且 64 位浮点型由 F 终止(F 为链接寄存器用)。
- (b) 对于 32 位整数和 64 位浮点型，请以偶数指定软元件数。(它不能设定为奇数。)
- (c) 惯性定时器以每 888 μ s 递增。(惯性定时器为 32 位整数。)

(2) 位软元件说明

	软元件说明	软元件号(n) 的指定范围
输入继电器	Xn/PXn	0 到 1FFF
输出继电器	Yn/PYn	0 到 1FFF
内部继电器	Mn	0 到 8191
锁存继电器	Ln	0 到 8191
链接继电器	Bn	0 到 1FFF
报警器	Fn	0 到 2047
特殊继电器	Mn	9000 到 9255

- (a) 当在 DIN 或 DOUT 中将软元件作为批位数据时，指定 n 为 16 的倍数。

(3) 软元件号的间接指定

在上述的字 / 位软元件说明中，软元件号(n)可以间接指定。

(a) 使用字软元件的软元件号(n)的间接指定。

- 软元件号已被间接指定的字软元件不能使用。
- 对于间接指定，您可以使用 16 位或 32 位整数型字软元件。
64 位浮点型字软元件不能使用。

(说明示例)

正确示例	错误示例
#(D10)	#(D(D5))
D(#10L)F	D(#4F)

(b) 使用运算表达式的字软元件的软元件号(n)的间接指定的操作表达式。

- 软元件号(n)可通过使用下列数据和运算的计算公式来间接指定。

可使用的数据	16 位整数型字软元件
	32 位整数型字软元件
	16 位整数型常数
	32 位整数型常数
可使用的运算	加: +
	减: -
	乘: *
	除: /
	余数: %
	符号变换: -

- 不能使用被间接指定的字软元件的软元件号。
- 仅可使用一个运算。

(说明示例)

正确示例	错误示例
#(D10-K5)	#(D(D5)F+K20)
D(#10L%H6L)F	D(#4L<<K2)

(注): 当您要使用非以上的计算结果来间接指定软元件号时，以下列两块来描述。

D0=SHORT(ASIN(#0F)) W0=#(D0)

7 操作控制程序

7.3 常数说明

16 位整数型，32 位整数型以及 64 位浮点型的常数说明如下图所示：

	16 位整数型	32 位整数型	64 位浮点型
十进制表示	K-32768 K32767	到 K-2147483648L 到 K2147483647L	K-1.79E+308 到 K-2.23E-308, K0.0, K2.23E-308 到 K1.79E+308
十六进制表示	H0000 到 HFFFF	H00000000L 到 HFFFFFFFL	—

- (a) 32 位整数型由 L 结尾，64 位浮点型提供了十进制小数点和指数部分以明确表示它们的数据。
- (b) 无数据型的常数可认为是能适用的最小类型。
- (c) 十进制表示中的常数由 K 开头且十六进制表示中的由 H 开头。K 可以被省略。
- (d) 64 位浮点型不能用十六进制中表示。

7 操作控制程序

F/FS	G
○	○

7.4 二项式运算

7.4.1 赋值: =

格式	(D)=(S)	基本步骤数	4
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—
(D)	—	○	○	○	—	—	—	—	—	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	要替换的字/常数/计算公式	(D)的数据类型
(D)	将存储运算结果的字软元件	

[功能]

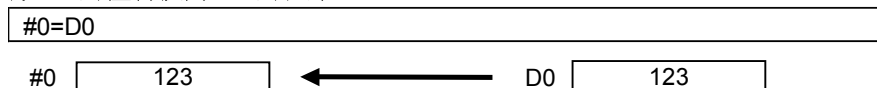
- (1) 以(S)指定的数据值替换(D)中指定的字软元件。
- (2) 当(S)和(D)在数据类型上不同时, (S)中的数据转换成(D)中的数据类型且替换结果数据。
(当(D)为16位或32位整数型且(S)为64位浮点型时, (S)的余数部分可忽略不计。

[出错]

- (1) 当出现下列情况时会发生运算出错:
 - (S)中的数据在(D)中的数据类型范围之外; 或:
 - (D)或(S)是间接指定的软元件且它们的软元件 No. 在指定范围之外。

[程序示例]

- (1) 将 D0 的值替换为 #0 的程序。



- (2) 将K123456.789替换为D0L

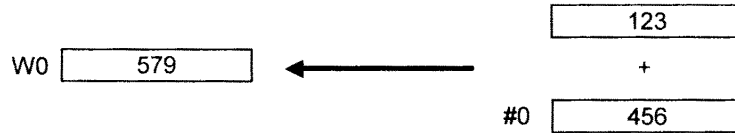
D0L=K123456.789



64 位浮点型转换成 32 位整数型及替换结果。

- (3) 将K123 和 #0 的相加结果替换为 W0:

W0 = K123+#0



7 操作控制程序

F/FS	G
○	○

7.4.2 加: +

格式	(S1)+(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	被加数据	较大的(S1)或(S2)的数据类型
(S2)	加数据	

[功能]

- (1) 以(S2)指定的数据被加到以(S1)指定的数据中。
- (2) 当(S1)和(S2)的数据类型不同时, 在进行运算以前, 较小数据类型的数据会转换成较大数据类型的数据。

[出错]

- (1) 当出现下列情况时会发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它们的软元件 No.在指定的范围之外。

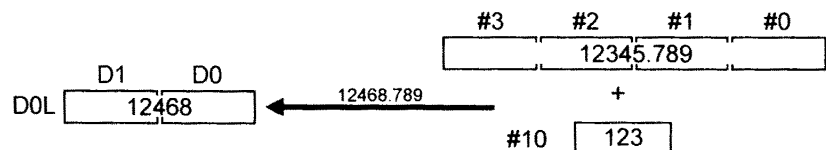
[程序示例]

- (1) 将 K123 和 #0 的相加结果替换为 W0 的程序:

```
W0 = K123+#0
```

- (2) 将 #0F 和 #10 的相加结果替换为 D0L:

```
D0L = #0F+#10
```



64 位浮点型数据用于加法, 且结果转换成 32 位整数型并被替换。

7 操作控制程序

F/FS	G
○	○

7.4.3 减: -

格式	(S1)-(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	被减数据	较大的(S1)或(S2)的数据类型
(S2)	减数据	

[功能]

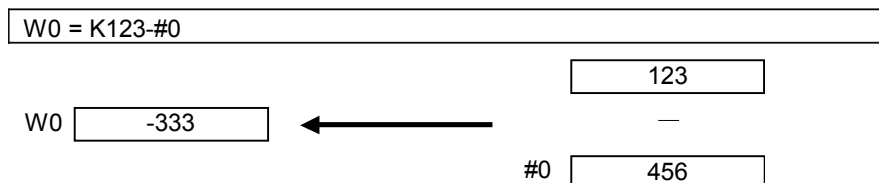
- (1) 以(S2)指定的数据从以(S1)指定的数据中减去。
- (2) 当(S1)和(S2)的数据类型不同时, 在进行运算之前, 较小类型的数据会转换成较大类型的数据。

[出错]

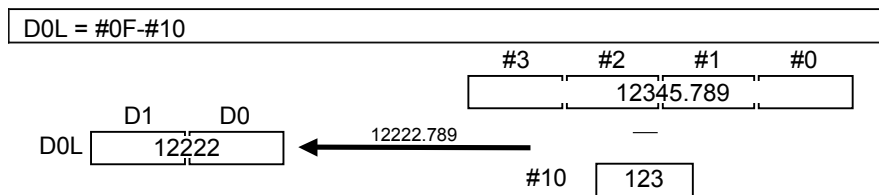
- (1) 当出现下列情况时会发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它们的软元件No.在指定的范围之外。

[程序示例]

- (1) 将从K123减去#0的数据转换成W0的程序:



(2) 将从 #0F 减去 #10 的数据转换成 D0L 的程序:



64 位浮点型数据

7 操作控制程序

F/FS	G
○	○

7.4.4 乘：*

格式	(S1)*(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	被乘数据	较大的(S1)或(S2)的数据类型
(S2)	乘数据	

[功能]

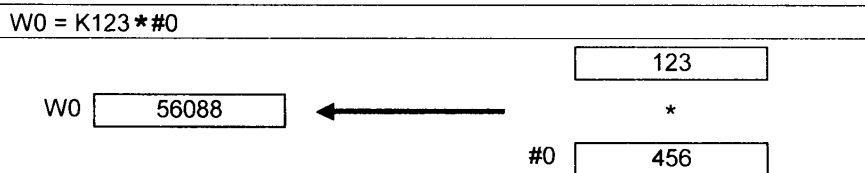
- (1) 以(S1)指定的数据乘以(S2)指定的数据。
- (2) 当(S1)和(S2)的数据类型不同时，在进行运算之前，较小类型的数据会转换成较大类型的数据。

[出错]

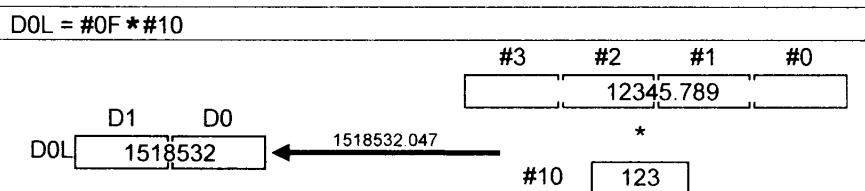
- (1) 当出现下列情况时会发生运算出错：
 - (S1)和(S2)是间接指定的软元件且它们的软元件 No. 在指定的范围之外。

[程序示例]

- (1) 将 K123 乘以 #0 的结果替换为 W0 的程序。



- (2) 将 #0F 乘以 #10 的结果替换成 D0L 的程序。



64 位浮点型可用于乘法，且它的结果转换成 32 位整数型并替换。

7 操作控制程序

F/FS	G
○	○

7.4.5 除：/

格式	(S1)/(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○：可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	被除数据	较大的(S1)或(S2)的数据类型
(S2)	除数据	

[功能]

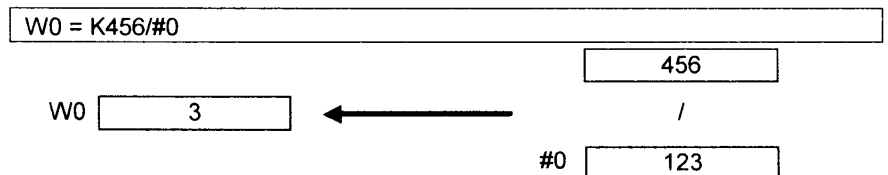
- 以(S1)指定的数据除以(S2)指定的数据以得到商。
- 当(S1)和(S2)的数据类型不同时，在进行运算之前，较小类型的数据会转换成较大类型的数据。

[出错]

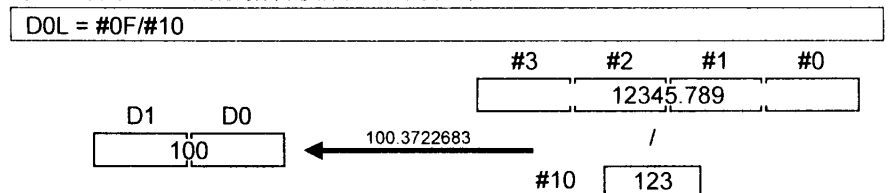
- 当出现下列情况时会发生运算出错：
 - (S1)为0；
 - (S1)或(S2)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- 将 K456 除以 #0 的商替换成 W0 的程序



- 将 #0F 除以 #10 的数据替换成 D0L 的程序



64 位浮点型数据可用于除法，且它的商转换成 32 位整数型并替换。

7 操作控制程序

F/FS	G
○	○

7.4.6 取余：%

格式	(S1)%(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○：可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	被除数据	较大的(S1)或(S2)的数据类型
(S2)	除数据	

[功能]

- (1) 以(S1)指定的数据除以(S2)指定的数据以得到余数。
- (2) 当(S1)和(S2)的数据类型不同时，在进行运算之前，较小类型的数据会转换成较大类型的数据。

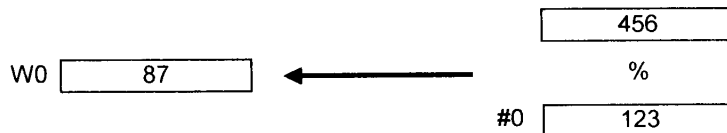
[出错]

- (1) 当出现下列情况时会发生运算出错：
 - (S1)为0;
 - (S1)或(S2)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) 将 K456 除以 #0 并将余数替换为 W0 的程序。

```
W0 = K456%#0
```



7 操作控制程序

F/FS	G
○	○

7.5 位运算

7.5.1 位取反(补充):~

格式	~(S)	基本步骤数	4
----	------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	—	○	○	○	—	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	位将被取反的数据	(S)的数据类型 (整数型)

[功能]

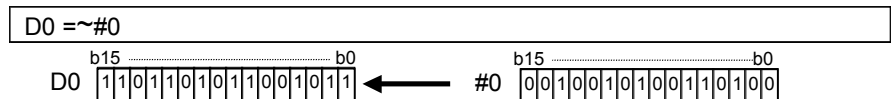
- (1) 查找以(S)指定的数据的位变换值。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) 查找#0的位变换值并将该值替换成D0的程序。



7 操作控制程序

F/FS	G
○	○

7.5.2 位逻辑与：&

格式	(S1)&(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将逐位 AND 的数据。	较大的(S1)或(S2)的数据类型 (整数型)
(S2)		

[功能]

- (1) 查找以(S1)指定的数据和以(S2)指定的数据的逐位逻辑与。
- (2) 当(S1)和(S2)的数据类型不同时，在进行运算之前，较小类型的数据会转换成较大类型的数据。此时，请注意带符号的数据已转换。

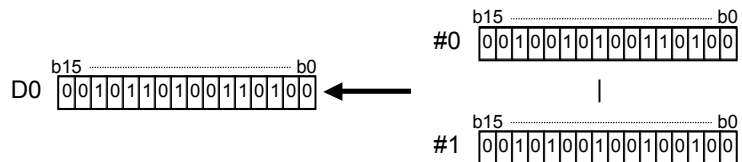
[出错]

- (1) 当出现下列情况时发生运算出错：
 - (S1)或(S2)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) AND#0 和 #1 并将结果替换为 D0 的程序。

D0 = #0



7 操作控制程序

F/FS	G
○	○

7.5.3 位逻辑或：|

格式	(S1) (S2)	基本步骤数	4
----	-------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将逐位 OR 的数据。	较大的(S1)或(S2)的数据类型 (整数型)
(S2)		

[功能]

- (1) 查找以(S1)指定的数据和以(S2)指定的数据的逐位逻辑或。
- (2) 当(S1)和(S2)的数据类型不同时，在进行运算之前，较小类型的数据会转换成较大类型的数据。此时，请注意带符号的数据已转换。

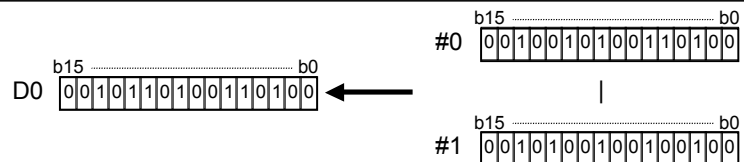
[出错]

- (1) 当出现下列情况时发生运算出错：
 - (S1)或(S2)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) OR#0 和 #1 并将结果替换为 D0 的程序。

D0 = #0 | #1



7 操作控制程序

F/FS	G
○	○

7.5.4 位异或逻辑 OR: ^

格式	(S1)^(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将逐位异或的数据。	较大的(S1)或(S2)的数据类型 (整数型)
(S2)		

[功能]

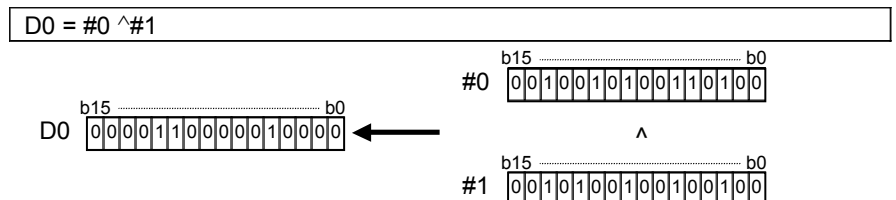
- (1) 查找以(S1)指定的数据和以(S2)指定的数据的逐位异或。
- (2) 当(S1)和(S2)的数据类型不同时, 在进行运算之前, 较小类型的数据会转换成较大类型的数据。此时, 请注意带符号的数据已转换。

[出错]

- (1) 当出现下列情况时发生运算出错:
 - (S1)或(S2)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) 异或 #0 和 #1 并将结果替换为 D0 的程序。



7 操作控制程序

F/FS	G
○	○

7.5.5 位右移: >>

格式	(S1)>>(S2)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将右移的数据	(S1)的数据类型 (整数型)
(S2)	右移数	

[功能]

- (1) 以(S1)指定的数据以(S2)指定的次数向右移动。
- (2) 如果(S1)中最低是1, 则1输入右移结果中最高位。
如果(S1)中最低是0, 则0输入右移结果中最高位。
- (3) 当(S1)为16位整数型且(S2)是负数或不小于16时, 其结果为0。
- (4) 当(S1)为32位整数型且(S2)是负数或不小于32时, 其结果为0。

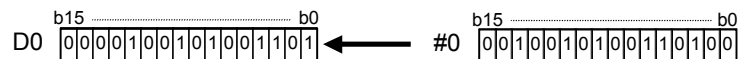
[出错]

- (1) 当出现下列情况时发生运算出错:
 - (S1)或(S2)为间接指定的软元件且它们的软元件 No. 在指定的范围之外。

[程序示例]

- (1) 将#0向右移动两位的位置并将结果替换为D0的程序。

D0 = #0>>K2



7 操作控制程序

F/FS	G
○	○

7.5.6 位左移: <<

格式	(S1)<<(S2)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	—	○	○	○	—	○	—	—
(S2)	—	○	○	—	○	○	○	—	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将左移的数据	(S1)的数据类型 (整数型)
(S2)	左移数	

[功能]

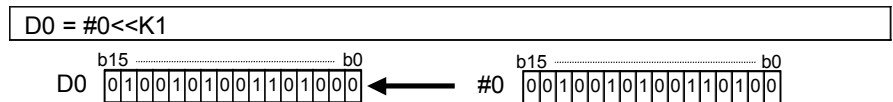
- (1) 以(S1)指定的数据以(S2)指定的次数向左移动。
- (2) 0 输入右移结果中最低位。
- (3) 当(S1)为 16 位整数且(S2)是负数或不小于 16 时, 其结果为 0。
- (4) 当(S1)为 32 位整数且(S2)是负数或不小于 32 时, 其结果为 0。

[出错]

- (1) 当出现下列情况时发生运算出错:
 - (S1)或(S2)为间接指定的软元件且它们的软元件 No. 在指定的范围之外。

[程序示例]

- (1) 将 #0 向左移动一位的位置并将结果替换为 D0 的程序。



7 操作控制程序

F/FS	G
○	○

7.5.7 符号取反(2的补码):—

格式	-(S)	基本步骤数	2
----	------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将变换符号的数据	(S)的数据类型

[功能]

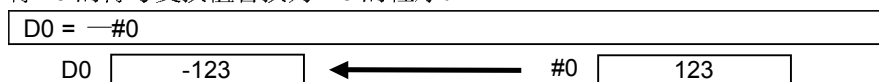
- (1) 查找以(S)指定的数据的符号变换值。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)为间接指定的软元件且它们的软元件 No.在指定的范围之外。

[程序示例]

- (1) 将#0的符号变换值替换为D0的程序。



7 操作控制程序

F/FS	G
○	○

7.6 标准功能

7.6.1 正弦: SIN

格式	SIN(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行 SIN(正弦)运算的角度数据	浮点型

[功能]

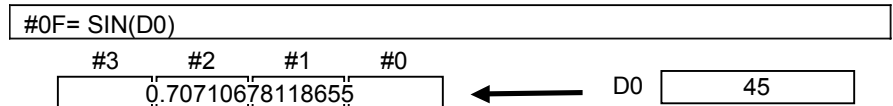
- (1) SIN(正弦)运算在以(S)指定的数据上进行。
- (2) 以(S)指定的数据在角(度)单位中。
- (3) 如果(S)是整数, 在进行运算之前将转换成浮点数。

[出错]

- (1) 出现下列情况时发生运算出错:
 - (S)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) 进行 D0 的 SIN 运算并将结果替换为 #0F 的程序:



7 操作控制程序

F/FS	G
○	○

7.6.2 余弦：COS

格式	COS(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行 COS(余弦)运算的角度数据	浮点型

[功能]

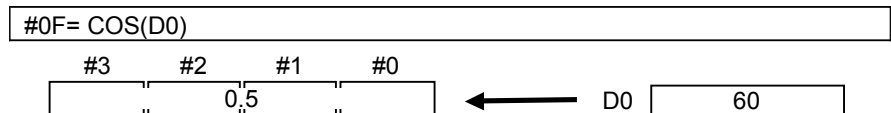
- (1) COS(余弦)运算在以(S)指定的数据上进行。
- (2) 以(S)指定的数据在角(度)单元中。
- (3) 如果(S)是整数型，在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)为间接指定的软元件且它们的软元件号在指定的范围之外。

[程序示例]

- (1) 进行 D0 的 COS 运算并将结果替换为 #0F 的程序：



7 操作控制程序

F/FS	G
○	○

7.6.3 正切: TAN

格式	TAN(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行 TAN(正切)运算的角度数据	浮点型

[功能]

- (1) TAN(正切)运算在以(S)指定的数据上进行。
- (2) 以(S)指定的数据在角(度)单位中。
- (3) 如果(S)是整数型, 在进行运算之前将转换成浮点型。

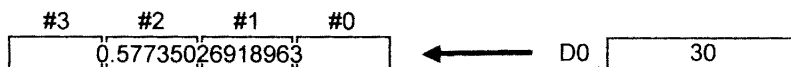
[出错]

- (1) 出现下列情况时发生运算出错:
 - (S)为间接指定的软元件且它们的软元件号在指定的范围之外; 或
 - (S)为 $90+(180*n)$ 。(“n”是一个整数)

[程序示例]

- (1) 进行 D0 的 TAN 运算并将结果替换为 #0F 的程序:

```
#0F= TAN(D0)
```



7 操作控制程序

F/FS	G
○	○

7.6.4 反正弦：ASIN

格式	ASIN(S)	基本步骤数	2
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行 SIN^{-1} (反正弦)运算的 SIN 值数据	浮点型

[功能]

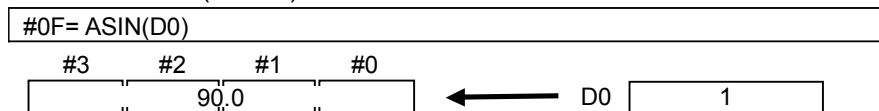
- (1) SIN^{-1} (反正弦)运算在以(S)指定的 SIN 值数据上进行计算得出角度值。
- (2) 以(S)指定的 SIN 值在 -1.0 到 1.0 中。
- (3) 运算结果以角(度)为单位。
- (4) 如果(S)是整数型，在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)在 -1.0 到 1.0 的范围之外；或：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 进行 D0 的 SIN^{-1} (反正弦)运算并将结果替换为 #0F 的程序：



7 操作控制程序

F/FS	G
○	○

7.6.5 反余弦：ACOS

格式	ACOS(S)	基本步骤数	2
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行 COS^{-1} (反余弦)运算的 COS 值数据	浮点型

[功能]

- (1) COS^{-1} (反余弦)运算在以(S)指定的 COS 值数据上进行计算得出角度值。
- (2) 以(S)指定的 COS 值在 -1.0 到 1.0 中。
- (3) 运算结果以角(度)为单位。
- (4) 如果(S)是整数型，在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)在 -1.0 到 1.0 的范围之外；或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 进行 D0 的 COS^{-1} (反正弦)运算并将结果替换为 #0F 的程序：



7 操作控制程序

F/FS	G
○	○

7.6.6 反正切: ATAN

格式	ATAN(S)	基本步骤数	2
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行 TAN^{-1} (反正弦)运算的 TAN 值数据	浮点型

[功能]

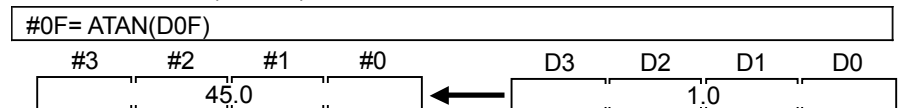
- (1) TAN^{-1} (反正弦)运算在以(S)指定的 TAN 值数据上进行计算得出角度值。
- (2) 运算结果以角(度)为单位。
- (3) 如果(S)是整数型, 在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 进行 D0 的 TAN^{-1} (反正弦)运算并将结果替换为 #0F 的程序:



7 操作控制程序

F/FS	G
○	○

7.6.7 平方根: SQRT

格式	SQRT(S)	基本步骤数	2
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行平方根运算的数据	浮点型

[功能]

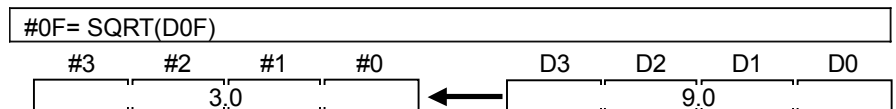
- (1) 计算以(S)指定的数据的平方根。
- (2) 只有正数可以由(S)指定。(不能以负数进行运算。)
- (3) 如果(S)是整数型，在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)为负数；或：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 查找 D0F 的平方根并将结果替换成 #0F 的程序。



7 操作控制程序

F/FS	G
○	○

7.6.8 自然对数: LN

格式	LN(S)	基本步骤数	2
----	-------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行自然对数运算的数据	浮点型

[功能]

- (1) 计算以(S)指定的数据的以 e 为底的自然对数。
- (2) 只有正数可以由(S)指定。(不能以负数进行运算。)
- (3) 如果(S)是整数型，在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)为负数；或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 查找 D0F 的自然对数并将结果替换成 #0F 的程序。

#0F= LN(D0F)							
#3	#2	#1	#0	D3	D2	D1	D0
			2.3025850929940			10.0	

7 操作控制程序

F/FS	G
○	○

7.6.9 指数运算：EXP

格式	EXP(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行指数运算的数据	浮点型

[功能]

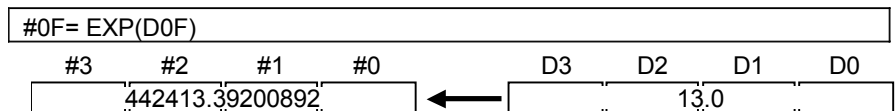
- (1) 进行由(S)指定的以 e 为底的指数运算。
- (2) 如果(S)是整数型，在进行运算之前将转换成浮点型。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 进行 D0F 的指数运算并将结果替换成 #0F 的程序。



7 操作控制程序

F/FS	G
○	○

7.6.10 绝对值: ABS

格式	ABS(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将进行绝对值转换的数据	浮点型

[功能]

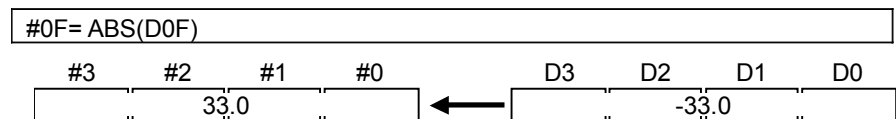
- (1) 进行以(S)指定的数据的绝对值计算。

[出错]

- (1) 出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 查找 D0F 的绝对值并将结果替换成 #0F 的程序。



7 操作控制程序

F/FS	G
○	○

7.6.11 四舍五入：RND

格式	RND(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○：可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	小数部分将被四舍五入的数据	(S)的数据类型

[功能]

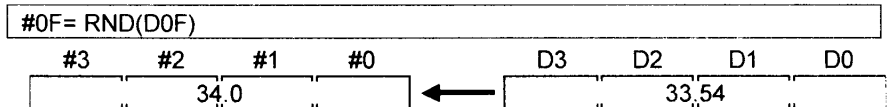
- (1) 计算以(S)指定的数据的四舍五入小数部分值。
- (2) 如果(S)为负数，则查找到(S)的绝对值，且它的小数部分将四舍五入并带符号。
- (3) 如果(S)为整数型，则它的值保持不变且不进行任何转换处理。

[出错]

- (1) 出现下列情况时发生运算出错：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 查找 D0F 的四舍五入的小数部分值并将结果替换成 #0F 的程序。



- (2) 查找 D4F 的四舍五入的小数部分值并将结果替换成 #0F 的程序。
(当 D4F 为负数时。)



7 操作控制程序

F/FS	G
○	○

7.6.12 舍: FIX

格式	FIX(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	小数部分将被舍弃的数据	(S)的数据类型

[功能]

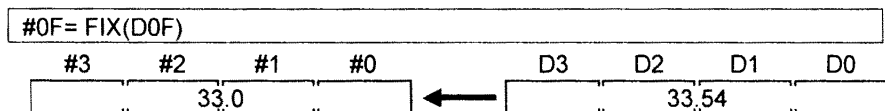
- (1) 计算得出不大于以(S)指定的数据的最大整数。
- (2) 如果(S)为正数, 则其绝对值会变小, 如果它是负数, 则它的绝对值会变大。
- (3) 如果(S)为整数型, 则它的值保持不变且不进行任何转换处理。

[出错]

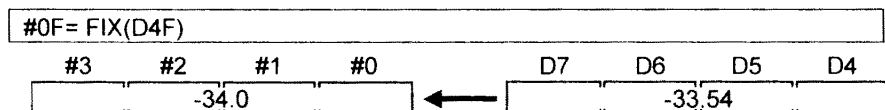
- (1) 出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 查找 D0F 的舍弃的小数部分值并将结果替换成 #0F 的程序。



- (2) 查找 D4F 的舍弃的小数部分值并将结果替换成 #0F 的程序。(当 D4F 为负数时。)



7 操作控制程序

F/FS	G
○	○

7.6.13 入: FUP

格式	FUP(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	小数部分将被入(进位)的数据	(S)的数据类型

[功能]

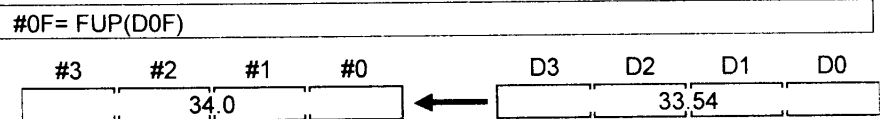
- (1) 计算得出不小于以(S)指定的数据的最小整数。
- (2) 如果(S)为正数, 则其绝对值会变大, 如果它是负数, 则它的绝对值会变小。
- (3) 如果(S)为整数型, 则它的值保持不变且不进行任何转换处理。

[出错]

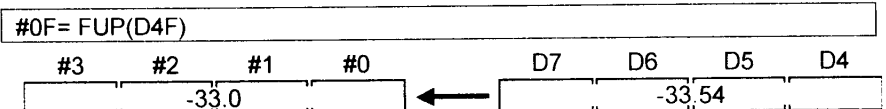
- (1) 出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 查找 D0F 的进位的小数部分值并将结果替换成 #0F 的程序。



- (2) 查找 D4F 的进位的小数部分值并将结果替换成 #0F 的程序。(当 D4F 为负数时。)



7 操作控制程序

F/FS	G
○	○

7.6.14 BCD → BIN 转换:BIN

格式	BIN(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	—	○	○	—	○	○	○	—	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成 BIN 数据的 BCD 数据	(S)的数据类型 (整数型)

[功能]

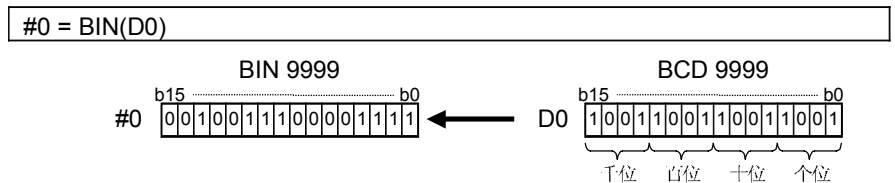
- (1) 以(S)指定的 BCD 数据转换成 BIN 数据。
- (2) 如果(S)为 16 位整数型，则数据范围为 0 到 9999。
- (3) 如果(S)为 32 位整数型，则数据范围为 0 到 99999999。

[错误]

- (1) 出现下列情况时发生运算出错：
 - 非 0 到 9 的值在(S)的任一数字中；或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0 的 BCD 数据转换成 BIN 数据并将结果替换为 #0 的程序。



7 操作控制程序

F/FS	G
○	○

7.6.15 BIN → BCD 转换:BCD

格式	BCD(S)	基本步骤数	2
----	--------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	—	○	○	○	—	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成 BCD 数据 BIN 数据	(S)的数据类型 (整数型)

[功能]

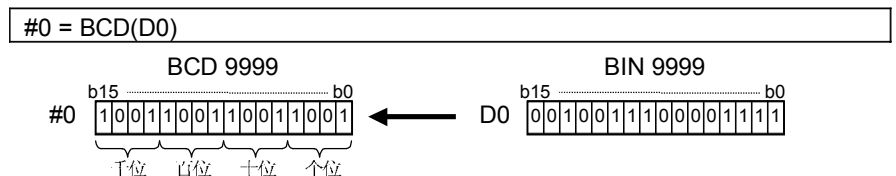
- (1) 以(S)指定的 BIN 数据转换成 BCD 数据。
- (2) 如果(S)为 16 位整数型，则数据范围为 0 到 9999。
- (3) 如果(S)为 32 位整数型，则数据范围为 0 到 99999999。

[错误]

- (1) 出现下列情况时发生运算出错：
 - 当(S)为 16 位整数型时，数据为非 0 到 9999；或：
 - 当(S)为 32 位整数型时，数据位非 0 到 99999999；或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0 的 BIN 数据转换成 BCD 数据并将结果替换为 #0 的程序。



7 操作控制程序

F/FS	G
○	○

7.7 类型转换

7.7.1 带符号的 16 位整数值转换：SHORT

格式	SHORT(S)	基本步骤数	2
----	----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成带符号的 16 位整数值的数据	16 位整数型

[功能]

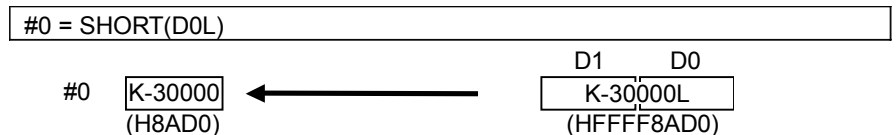
- (1) 以(S)指定的数据转换成带符号的 16 位整数型。
- (2) (S)的数据范围是 -32768 到 32767。
- (3) 当(S)是 64 位浮点型时，进行转换前它的小数部分会被舍弃。
- (4) 如果(S)是 16 位整数型，它的值返回不变，也不进行转换处理。

[错误]

- (1) 出现下列情况时发生运算出错：
 - 当(S)的数据在 -32768 到 32767 的范围之外；或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0L 的数据转换成带符号的 16 位整数值并将结果替换为 #0 的程序。



7 操作控制程序

F/FS	G
○	○

7.7.2 不带符号的 16 位整数值转换: USHORT

格式	USHORT(S)	基本步骤数	2
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成不带符号的 16 位整数值的数据	16 位整数型

[功能]

- (1) 以(S)指定的数据转换成不带符号的 16 位整数型。
- (2) (S)的数据范围是 0 到 65535。
- (3) 当(S)是 64 位浮点型时, 进行转换前它的小数部分会被舍弃。
- (4) 如果(S)是 16 位整数型, 它的值返回不变, 也不进行转换处理。

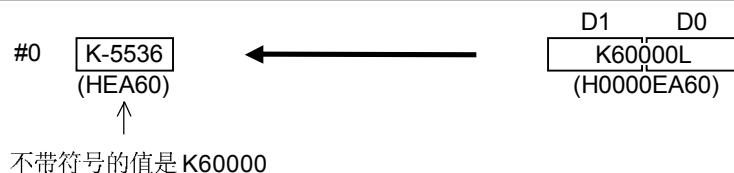
[错误]

- (1) 出现下列情况时发生运算出错:
 - 当(S)的数据在 0 到 65535 的范围之外; 或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0L 的数据转换成不带符号的 16 位整数值并将结果替换为 #0 的程序。

```
#0 = USHORT(D0L)
```



7 操作控制程序

F/FS	G
○	○

7.7.3 带符号的 32 位整数值转换: LONG

格式	LONG(S)	基本步骤数	2
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成带符号的 32 位整数值的数据	32 位整数型

[功能]

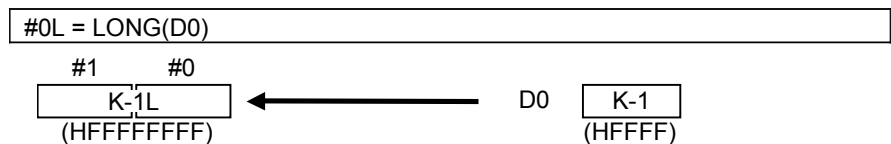
- (1) 以(S)指定的数据转换成带符号的 32 位整数型。
- (2) (S)的数据范围是 -2147483648 到 2147483647。
- (3) 当(S)是 64 位浮点型时, 进行转换前它的小数部分会被舍弃。
- (4) 如果(S)是 32 位整数型, 它的值返回不变, 也不进行转换处理。

[错误]

- (1) 出现下列情况时发生运算出错:
 - 当(S)的数据在 -2147483648 到 2147483647 的范围之外; 或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0L 的数据转换成带符号的 32 位整数值并将结果替换为 #0L 的程序。



7 操作控制程序

F/FS	G
○	○

7.7.4 不带符号的 32 位整数值转换：ULONG

格式	ULONG(S)	基本步骤数	2
----	----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成不带符号的 32 位整数值的数据	32 位整数型

[功能]

- (1) 以(S)指定的数据转换成不带符号的 32 位整数型。
- (2) (S)的数据范围是 0 到 4294967295。
- (3) 当(S)是 64 位浮点型时，进行转换前它的小数部分会被舍弃。
- (4) 如果(S)是 32 位整数型，它的值返回不变，也不进行转换处理。

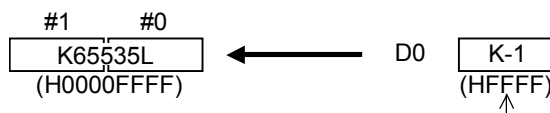
[错误]

- (1) 出现下列情况时发生运算出错：
 - 当(S)的数据在 0 到 4294967295 的范围之外；或
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0L 的数据转换成不带符号的 32 位整数值并将结果替换为 #0L 的程序。

```
#0L = ULONG(D0)
```



不带符号的值是 K65535

7 操作控制程序

F/FS	G
○	○

7.7.5 带符号的 64 位浮点值转换: FLOAT

格式	FLOAT(S)	基本步骤数	2
----	----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成带符号的 64 位浮点值的数据	64 位浮点型

[功能]

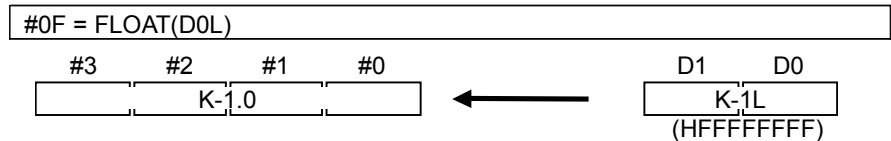
- (1) 以(S)指定的数据转换成带符号的 64 位浮点值。
- (2) 如果(S)是 64 位浮点型时, 它的值返回不变, 也不进行转换处理。

[错误]

- (1) 出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0L 的数据转换成带符号的 64 位浮点值并将结果替换为 #0F 的程序。



7 操作控制程序

F/FS	G
○	○

7.7.6 不带符号的 64 位浮点值转换：UFLOAT

格式	UFLOAT(S)	基本步骤数	2
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(S)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被转换成不带符号的 64 位浮点值的数据	64 位浮点型

[功能]

- (1) 以(S)指定的数据转换成不带符号的 64 位浮点值。
- (2) 如果(S)是 64 位浮点型时，它的值返回不变，也不进行转换处理。

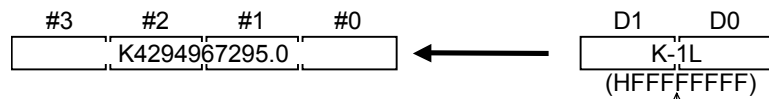
[错误]

- (1) 出现下列情况时发生运算出错：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 将 D0L 的数据转换成不带符号的 64 位浮点值并将结果替换为 #0F 的程序。

```
#0F = UFLOAT(D0L)
```



不带符号的值是 K4294967295

7 操作控制程序

F/FS	G
○	○

7.8 位软元件状态

7.8.1 ON(常开触点):(无)

格式	(S)	基本步骤数	2
----	-----	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S)	○	—	—	—	—	—	—	—	—	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	在位条件公式中使用的位软元件	逻辑型(I/O)

[功能]

- (1) 当位条件公式中以(S)指定的位软元件是ON(1)时返回真, 或当该位软元件是OFF(0)时返回假。

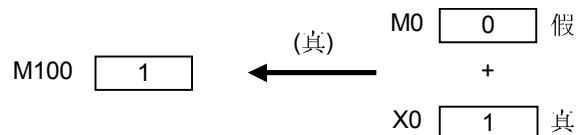
[错误]

- (1) 出现下列情况时发生运算出错:
 •(S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 当M0或X0是ON(1)时设置M100的程序。

```
SET M100 = M0 + X0
```



7 操作控制程序

F/FS	G
○	○

7.8.2 OFF(常闭触点):(I)

格式	!(S)	基本步骤数	2
----	------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	○	—	—	—	—	—	—	—	—	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	在位条件公式中使用的位软元件	逻辑型(I/O)

[功能]

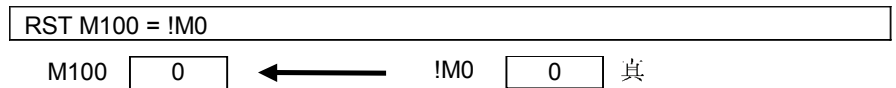
- (1) 当位条件公式中以(S)指定的位软元件是 OFF(0)时返回真, 或当该位软元件是 ON(1)时返回假。

[错误]

- (1) 出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 当 M0 是 OFF(0)时复位 M100 的程序。



7 操作控制程序

F/FS	G
○	○

7.9 位软元件控制

7.9.1 软元件置位：SET=

格式	SET(D)=(S)	基本步骤数	2
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	○	○

○：可使用

(注 -1):PX 是禁止写入且不能在(D)中使用。

(注 -2):M2001 到 M2032 不能在(D)中使用。

[设定数据]

设定数据	说明	结果的数据类型
(D)	软元件设置用位数据	位逻辑型
(S)	决定是否进行软元件设置的条件数据	(I/O)

[功能]

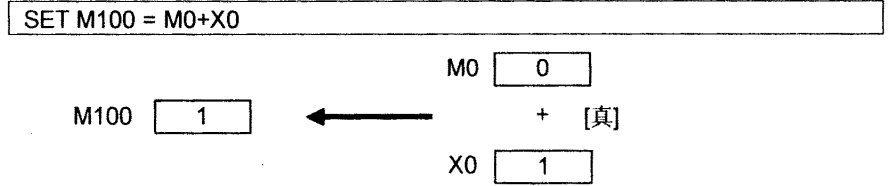
- (1) 如果以(S)指定的数据为真的，则设置以(D)指定的位数据。
- (2) (S)可以省略。
此时，格式是“SET(D)”且软元件是无条件设置的。
- (3) 当在转换程序的最后一块中将该指令设置为转移条件时，则以(S)指定的数据是真还是假会返回为逻辑型数据。此时，(S)不能被省略。

[错误]

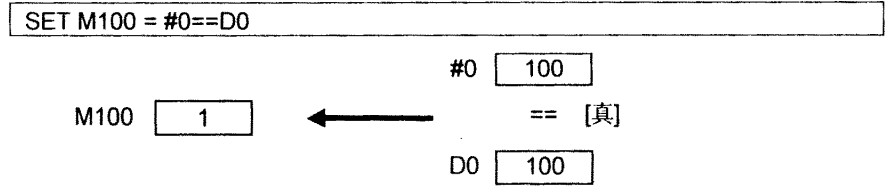
- (1) 当出现下列情况时发生运算出错：
 - (D)或(S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

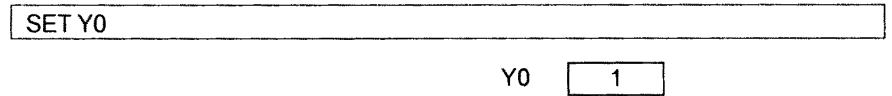
- (1) 当 M0 或 X0 是 1 时设置 M100 的程序。



- (2) 当 #0=D0 时，设置 M100 的程序。



- (3) 无条件设置 Y0 的程序。



7 操作控制程序

F/FS	G
○	○

7.9.2 软元件复位: RST=

格式	RST(D)=(S)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	○	○

○:可使用

(注 -1):PX 是禁止写入且不能在(D)中使用。

(注 -2):M2001 到 M2032 不能在(D)中使用。

[设定数据]

设定数据	说明	结果的数据类型
(D)	软元件设置用位数据	位逻辑型(I/O)
(S)	决定是否进行软元件复位的条件数据	

[功能]

- (1) 如果以(S)指定的数据为真的, 则复位以(D)指定的位数据。
- (2) (S)可以省略。
此时, 格式是“SET(D)”且软元件是无条件复位的。
- (3) 当在转换程序的最后一块中将该指令设置为转移条件时, 则以(S)指定的数据是真还是假会返回为逻辑型数据。此时,(S)不能被省略。

[错误]

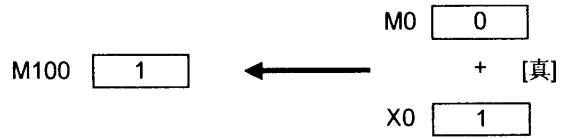
- (1) 当出现下列情况时发生运算出错:
 - (D)或(S)是间接指定的软元件且它的软元件号在指定的范围之外。

7 操作控制程序

[程序示例]

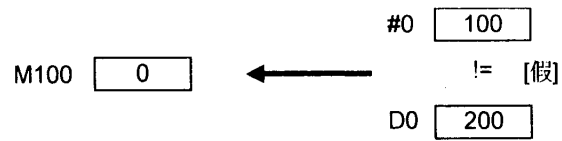
- (1) 当 M0 或 X0 是 1 时复位 M100 的程序。

RST M100 = M0 + X0



- (2) 当 #0=D0 时，复位 M100 的程序。

RST M100 = #0 != D0



- (3) 无条件复位 Y0 的程序。

RST Y0

Y0

7 操作控制程序

F/FS	G
○	○

7.9.3 软元件输出：DOUT

格式	DOUT(D), (S)	基本步骤数	4
----	--------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	—	○	○	—	○	○	○	—	○	—	—

○: 可使用

(注 -1): PX 和特殊继电器 M 且不能在(D)中使用。

(注 -2): M2000 到 M2127 不能在(D)中使用。

[设定数据]

设定数据	说明	结果的数据类型
(D)	输出目标位数据	批量位
(S)	输出源数据	

[功能]

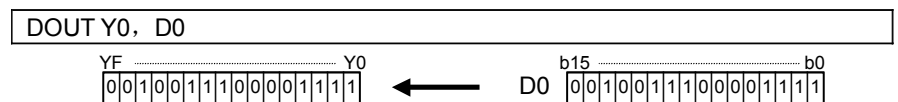
- (1) 以(S)指定的数据输出到以(D)指定的位数据。
- (2) 指定 16 的倍数为以(D)指定的位数据的软元件号。
- (3) 如果(S)的类型是 16 位整数型，则(S)数据的 16 点将以(D)指定的那个开始按照顺序从最低有效位输出到位软元件中。
- (4) 如果(S)的类型是 32 位整数型，则(S)数据的 32 点将以(D)指定的那个开始按照顺序从最低有效位输出到位软元件中。

[错误]

- (1) 当出现下列情况时发生运算出错：
 - (D)或(S)是间接指定的软元件且它的软元件号在指定的范围之外。
 - (D)是间接指定的软元件且它的软元件号不是 16 的倍数。

[程序示例]

- (1) 将 D0 的数据输出到 Y0-YF 的程序。



7 操作控制程序

F/FS	G
○	○

7.9.4 软元件输入：DIN

格式	DIN(D), (S)	基本步骤数	4
----	-------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D)	—	○	○	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(D)	输入目标位数据	(D)的数据类型 (整数型)
(S)	输入源位数据	

[功能]

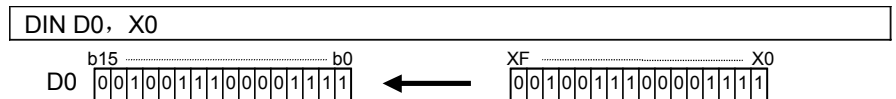
- (1) 以(S)指定的位数据输出到以(D)指定的位数据。
- (2) 指定 16 的倍数为以(D)指定的位数据的软元件号。
- (3) 如果(S)的类型是 16 位整数型，则(S)数据的 16 点将以(D)指定的那个开始按照顺序从最低有效位输出到位软元件中。
- (4) 如果(S)的类型是 32 位整数型，则(S)数据的 32 点将以(D)指定的那个开始按照顺序从最低有效位输出到位软元件中。

[错误]

- (1) 当出现下列情况时发生运算出错：
 - (D)或(S)是间接指定的软元件且它的软元件号在指定的范围之外。
 - (D)是间接指定的软元件且它的软元件号不是 16 的倍数。

[程序示例]

- (1) 将 X0 到 XF 的数据输出到 D0 的程序。



7 操作控制程序

F/FS	G
○	○

7.9.5 位软元件输出：OUT

参考“1.3.4”部分中的运动 CPU 和软件的相应版本。

格式	OUT(D)=(S)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(D)	○	—	—	—	—	—	—	—	—	—	—
(S)	○	—	—	—	—	—	—	—	—	○	○

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(D)	软元件输出用位软元件	位逻辑型 (I/O)
(S)	决定软元件输出的条件数据	

[功能]

- 如果以(S)指定的数据为真，则设置以(D)指定的位数据。如果以(S)指定的数据为假，则复位以(D)指定的位数据。
- 当在转换程序的最后一块中将该指令设置为转移条件时，则以(S)指定的数据是真还是假会返回为逻辑型数据。此时，(S)不能被省略。

[错误]

- 当出现下列情况时发生运算出错：
 - (D)或(S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- 当 M0 是 ON(1)时设置 M100 的程序以及当 M0 是 OFF(0)时复位 M100 的程序。

OUT M100 = M0

- 当 M0 和 M1 同时打开时设置 M100 的程序以及除此之外复位 M100 的程序。

OUT M100 = M0 * M1

- 当 D0 等于 D2000 时设置 M100 的程序以及当 D 不等于 D2000 时复位 M100 的程序。

OUT M100 = (D0 == D2000)

7 操作控制程序

F/FS	G
○	○

7.10 逻辑运算

7.10.1 逻辑应答: (无)

格式	(S)	基本步骤数	—
----	-----	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	○	—	—	—	—	—	—	—	—	○	○

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被逻辑应答的数据	逻辑型(I/O)

[功能]

- (1) 以(S)指定的逻辑型数据是真还是假返回不变。(逻辑应答。)

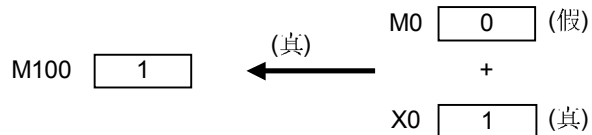
[错误]

- (1) 当出现下列情况时发生运算出错:
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 当 M0 或 X0 是 ON(1)时设置 M100 的程序。

```
SET M100 = M0 + X0
```



7 操作控制程序

F/FS	G
○	○

7.10.2 逻辑求反： !

格式	!(S)	基本步骤数	2
----	------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	○	—	—	—	—	—	—	—	—	○	○

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	将被逻辑求反的数据	逻辑型(I/O)

[功能]

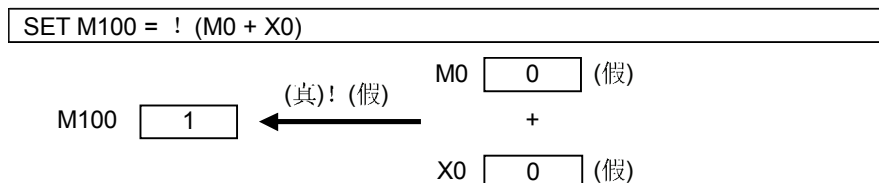
- (1) 以(S)指定的数据被逻辑求反。

[错误]

- (1) 当出现下列情况时发生运算出错：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 当“M0 或 X0 不是 ON(1)”时设置 M100 的程序。(当 M0 和 X0 均为 OFF 时(0))



7 操作控制程序

F/FS	G
○	○

7.10.3 逻辑与:*

格式	(S1)*(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	○	—	—	—	—	—	—	—	—	○	○
(S2)	○	—	—	—	—	—	—	—	—	○	○

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被 AND 的数据	逻辑型 (I/O)
(S2)		

[功能]

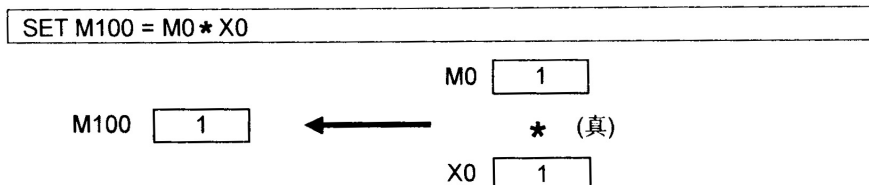
- (1) 以(S1)指定的数据和以(S2)指定的数据逻辑与。

[错误]

- (1) 当出现下列情况时发生运算出错：
 •(S)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 当 M0 和 X0 均是 1 时设置 M100 的程序。



7 操作控制程序

F/FS	G
○	○

7.10.4 逻辑或：+

格式	(S1)+ (S2)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	○	—	—	—	—	—	—	—	—	○	○
(S2)	○	—	—	—	—	—	—	—	—	○	○

○：可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被 OR 的数据	逻辑型(I/O)
(S2)		

[功能]

- (1) 以(S1)指定的数据和以(S2)指定的数据逻辑OR。

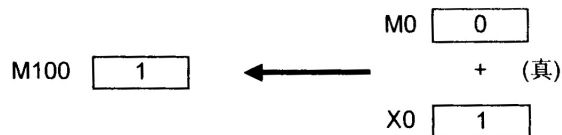
[错误]

- (1) 当出现下列情况时发生运算出错：
 •(S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 当M0或X0是1的时候设置M100的程序。

SET M100 = M0 + X0



7 操作控制程序

F/FS	G
○	○

7.11 比较运算

7.11.1 等于：==

格式	(S1)==(S2)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被比较的数据	逻辑型 (I/O)
(S2)		

[功能]

- 以(S1)指定的数据和以(S2)指定的数据将进行比较，如果它们相等，则结果就是真的。
- 当(S1)和(S2)在数据类型上不同时，进行比较之前，较小数据类型的数据将会转换成较大数据类型的数据。

[错误]

- 当出现下列情况时发生运算出错：
 - (S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- 比较 #0 和 D0 是否相等的程序。

#0 == D0	
#0	100
(真) ←	==
D0	100

7 操作控制程序

F/FS	G
○	○

7.11.2 不等于: !=

格式	(S1) != (S2)	基本步骤数	4
----	--------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被比较的数据	逻辑型(I/O)
(S2)		

[功能]

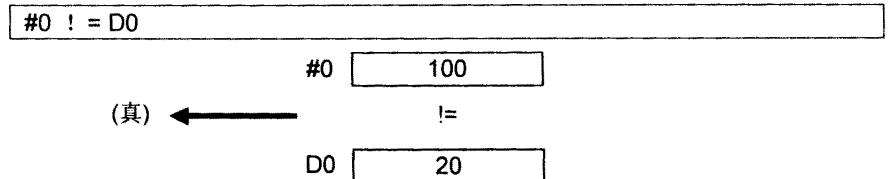
- 以(S1)指定的数据和以(S2)指定的数据将进行比较, 如果它们不相等, 则结果为真。
- 当(S1)和(S2)在数据类型上不同时, 进行比较之前, 较小数据类型的数据将会转换成较大数据类型的数据。

[错误]

- 当出现下列情况时发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- 比较#0和D0是否不相等的程序。



7 操作控制程序

F/FS	G
○	○

7.11.3 小于: <

格式	(S1)<(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被比较的数据	逻辑型(I/O)
(S2)		

[功能]

- (1) 如果以(S1)指定的数据小于以(S2)指定的数据, 则结果就是真的。
- (2) 当(S1)和(S2)在数据类型上不同时, 进行比较之前, 较小数据类型的数据将会转换成较大数据类型的数据。

[错误]

- (1) 当出现下列情况时发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 比较 #0 是否小于 D0 的程序。

#0 < D0	
#0	10
(真) ←	<
D0	20

F/FS	G
○	○

7.11.4 小于或等于: <=

格式	(S1)<=(S2)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被比较的数据	逻辑型 (I/O)
(S2)		

[功能]

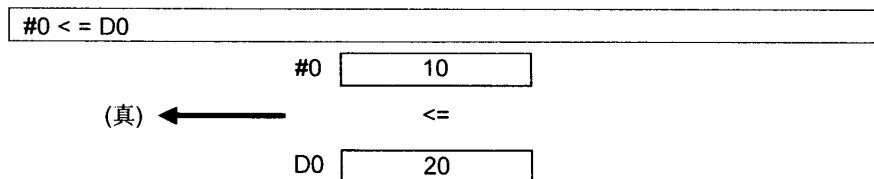
- (1) 如果以(S1)指定的数据小于或等于以(S2)指定的数据, 则结果就是真的。
- (2) 当(S1)和(S2)在数据类型上不同时, 进行比较之前, 较小数据类型的数据将会转换成较大数据类型的数据。

[错误]

- (1) 当出现下列情况时发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 比较 #0 是否小于或等于 D0 的程序。



7 操作控制程序

F/FS	G
○	○

7.11.5 大于: >

格式	(S1)>(S2)	基本步骤数	4
----	-----------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被比较的数据	逻辑型
(S2)		(I/O)

[功能]

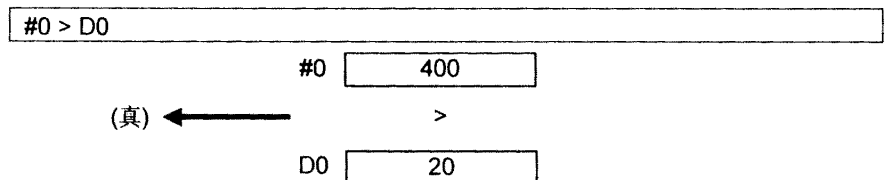
- 如果以(S1)指定的数据大于以(S2)指定的数据, 则结果就是真的。
- 当(S1)和(S2)在数据类型上不同时, 进行比较之前, 较小数据类型的数据将会转换成较大数据类型的数据。

[错误]

- 当出现下列情况时发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- 比较#0是否大于D0的程序。



7 操作控制程序

F/FS	G
○	○

7.11.6 大于或等于: >=

格式	(S1)>=(S2)	基本步骤数	4
----	------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S1)	—	○	○	○	○	○	○	○	○	—	—
(S2)	—	○	○	○	○	○	○	○	○	—	—

○:可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	将被比较的数据	逻辑型 (I/O)
(S2)		

[功能]

- (1) 如果以(S1)指定的数据大于或等于以(S2)指定的数据, 则结果就是真的。
- (2) 当(S1)和(S2)在数据类型上不同时, 进行比较之前, 较小数据类型的数据将会转换成较大数据类型的数据。

[错误]

- (1) 当出现下列情况时发生运算出错:
 - (S1)和(S2)是间接指定的软元件且它的软元件号在指定的范围之外。

[程序示例]

- (1) 比较#0是否大于或等于D0的程序。

#0 >=D0	
#0	400
(真) ←	>=
D0	20

7 操作控制程序

F/FS	G
○	○

7.12 运动专用功能(CHGV, CHGT)

7.12.1 转速改变请求: CHGV

格式	CHGV(S1), (S2)	基本步骤数	4
----	----------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	给出转速改变请求的轴号	—
(S2)	指定的转速	

[功能]

- 按照以下步骤进行转速改变:
 - 以(S1)指定的轴对应的转速改变标志位(M2061 到 M2092)为 ON。
 - 以(S1)指定的轴转速改变为以(S2)指定的轴转速。
 - 关闭转速改变标志位。
- 可以在(S1)中设置的轴 No. 须在以下范围内:

Q172CPU(N)	Q173CPU(N)
1 到 8	1 到 32

对于插补控制, 任意设置一个插补轴。当进行线性插补控制时, 转速改变随下面的伺服图表中设置的定位转速指定方法而变化。

定位转速指定方法	操作
合成转速指定	进行转速改变以使组合转速变成以(S2)指定的转速。
最长轴指定	进行转速改变以使最长轴转速变成以(S2)指定的转速。
基准轴转速指定	进行转速改变以使基准轴转速变成以(S2)指定的转速。

7 操作控制程序

(3) 操作随(S2)中设置的指定转速的符号而变化。

指定转速的符号	操作
正	转速改变
0	暂时停止
负	返回

(4) 可在(S2)中设置的指定转速须在下面的范围内。

• 实模式

	mm		inch		度		PLS	
	设置范围	单位	设置范围	单位	设置范围	单位	设置范围	单位
转速改变请求	0 到 600000000	$\times 10^2$ mm/min	0 到 600000000	$\times 10^3$ inch/min	0 到 2147483647	$\times 10^3$ degree/min	0 到 10000000	PLS/s
返回请求	-1 到-600000000	$\times 10^2$ mm/min	-1 到-600000000	$\times 10^3$ inch/min	-1 到-2147483647	$\times 10^3$ degree/min	-1 到-10000000	PLS/s

• 虚模式

	PLS	
	设置范围	单位
转速改变请求	0 到 10000000	PLS/s
返回请求	-1 到-10000000	PLS/s

(5) 伺服图表中通过 CHGV 指令的转速改变仅在启动时有效。

(6) 启动时通过指定负转速和进行转速改变请求，允许轴在该点时开始减速并在减速完成时朝相反方向返回。

控制模式	伺服指令	操作								
线性控制	<table border="0"> <tr> <td>ABS-1</td> <td>INC-1</td> </tr> <tr> <td>ABS-2</td> <td>INC-2</td> </tr> <tr> <td>ABS-3</td> <td>INC-3</td> </tr> <tr> <td>ABS-4</td> <td>INC-4</td> </tr> </table>	ABS-1	INC-1	ABS-2	INC-2	ABS-3	INC-3	ABS-4	INC-4	减速完成时，轴反向移动，以指定转速的绝对值返回到定位的启动点，并在该点停止(等待)。 对于圆弧插补，轴以圆弧轨迹返回。
ABS-1	INC-1									
ABS-2	INC-2									
ABS-3	INC-3									
ABS-4	INC-4									
圆弧插补控制	<table border="0"> <tr> <td>ABS circular</td> <td>INC circular</td> </tr> </table>	ABS circular	INC circular							
ABS circular	INC circular									
固定位移进给	<table border="0"> <tr> <td>FEED-1</td> <td>FEED-2</td> <td>FEED-3</td> </tr> </table>	FEED-1	FEED-2	FEED-3						
FEED-1	FEED-2	FEED-3								
恒定速度控制	<table border="0"> <tr> <td>CPSTART1</td> <td>CPSTART2</td> </tr> <tr> <td>CPSTART3</td> <td>CPSTART4</td> </tr> </table>	CPSTART1	CPSTART2	CPSTART3	CPSTART4	减速完成时，轴反向移动，以指定转速的绝对值返回到前一点，并在该点停止(等待)。				
CPSTART1	CPSTART2									
CPSTART3	CPSTART4									
速度控制(I)	VF VR	减速完成时，轴反向移动，以指定转速的绝对值返回。直到输入停止指令后轴才停止。								
速度控制(II)	VVF VVR									
速度/位置控制	VPF VPR VPSTART	轴不能返回。 转速改变请求被视为正常转速改变请求。 会发生轻度出错[305] ^(注) 且轴转速会被控制在转速极限值。								
位置随动控制	PFSTART									
速度切换控制	VSTART									
JOG 操作										
高速振动	OSC	不能进行速度改变。 会发生轻度出错[310] ^(注) 。								
原点回归	ZERO	不能进行速度改变。 会发生轻度出错[301] ^(注) 。								

(注): 轻度出错[301]: 原点回归时发生速度改变。
 轻度出错[305]: 设置速度在: 0 到转速极限值之外。
 轻度出错[310]: 高速振动时进行转速改变。

[控制]

- (a) 如果转速改变为负转速, 则控制会以上述图表7.12.1(3)中标明的启动时的控制模式进行。
- (b) 返回指令速度是新速度的绝对值。
- (c) 当轴在返回位置等待时:
 - 1) 信号状况(n: 轴号, m: 轴号-1)
 - 启动接受(M2000+n) ON
(在执行 CHGV 指令前不变)
 - 定位启动完成(M2400+20m) ON
(在执行 CHGV 指令前不变)
 - 定位完成(M2401+20m) OFF
 - 位置到达(M2402+20m) ON
 - 位置到达命令(M2403+20m) OFF
 - 转速改变“0”接受标志位(M2240+20m) ON
 - 2) 将转速改变到重启动用的正转速。
 - 3) 打开停止指令以结束定位。
 - 4) 再次进行负转速改变会被忽略。

- (d) 在速度控制模式中当轴反转时：
- 1) 将转速改变到正转速以再次变更移动方向。
 - 2) 打开停止指令以进行停止操作。
 - 3) 如果再次进行负转速改变，则以相反方向进行转速改变。

[出错]

- (1) 当出现下列情况时会发生运算错误且不能进行转速改变：
 - (S1)指定的轴号在范围之外；或
 - (S2)是间接指定的软元件且它的软元件号在指定范围之外。
- (2) 当出现下列情况时会发生轻度出错且不能进行转速改变：
 - 以(S1)指定的轴是原点返回型；或
 - 以(S1)指定的轴在减速。(轻度出错:303)
- (3) 当出现下列情况时会发生轻度出错且将以转速极限值控制轴：
 - 以(S2)指定的转速的绝对值大于转速极限值。(轻度出错: 305)

要点
如果恒定速度控制时新的转速的绝对值大于以伺服图表指定的转速，则会以图表中指定的转速进行返回控制。(恒定转速控制时转速改变用的转速控制。)此时，不会发生出错。

7 操作控制程序

[程序示例]

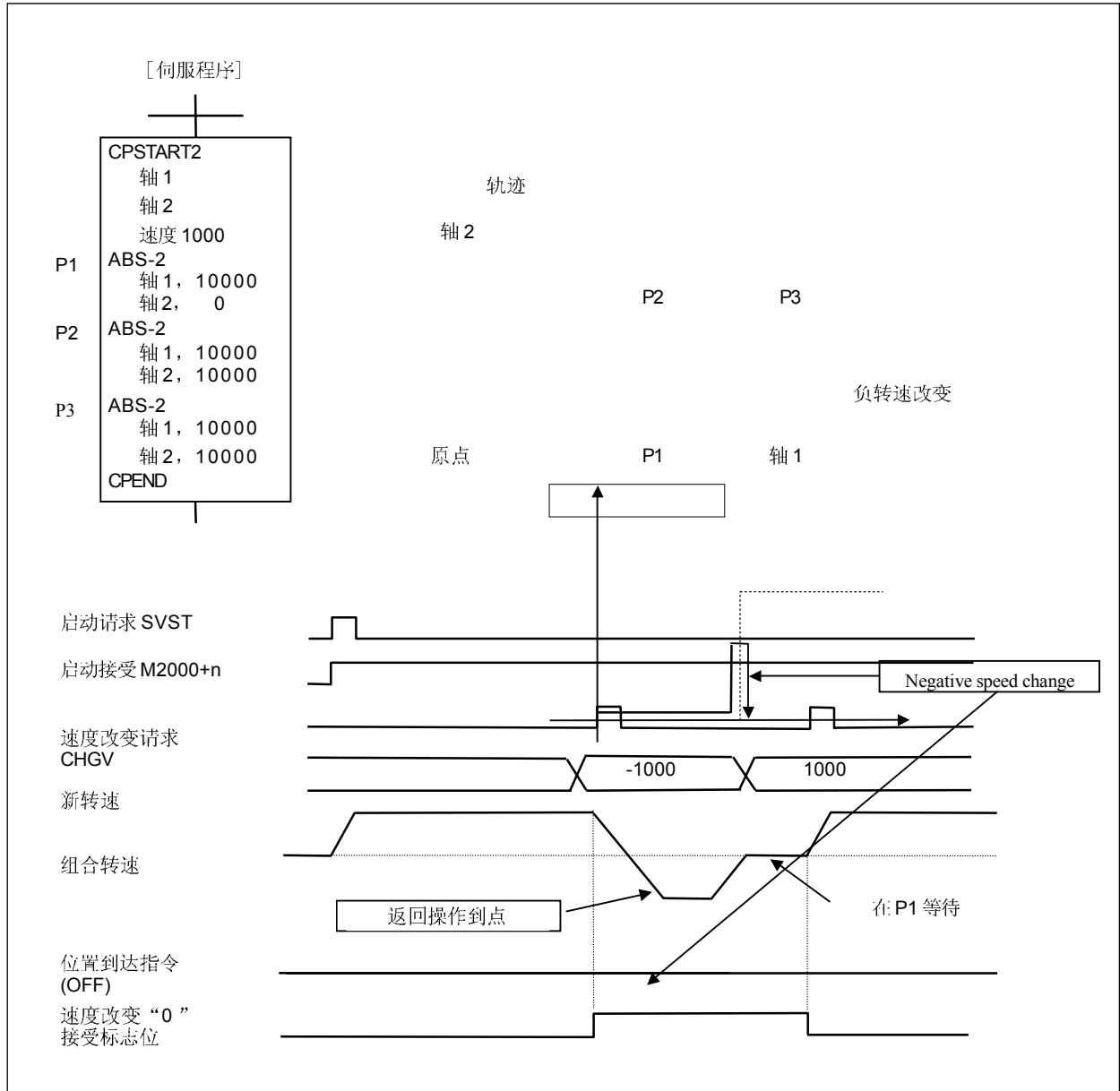
- (1) 改变轴 2 的定位速度的程序:

```
CHGV(K2, K10)
```

- (2) 将轴 1 的定位速度改变为负值的返回程序。

```
CHGV(K1, K-1000)
```

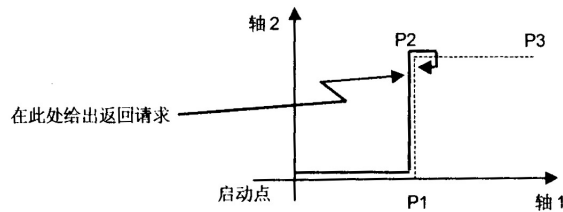
当在恒定转速控制中作出返回请求时将进行下列操作:



如上图所示在执行定位到P2时将转速改变到负转速,轴会沿着程序指定的轨迹返回到P1并在P1等待。

要点

- (1) 如果转速改变在从伺服程序启动请求作出时作出，则直到“定位启动完成信号”状态改变到 ON 时，转速改变可能无效。当以启动时几乎相同时间进行转速改变时，务必编写一个在“定位启动完成信号”变为 ON 以后进行转速改变的程序。
- (2) 当轴在恒定转速控制下使用 M 代码 FIN 等待功能在停止时等待 FIN 而作出的返回请求将被忽略。
- (3) 在以上示例中，如果返回请求在 P2 之前或在轴减速通过 P2 时给出，则轴会返回到 P2。
- (4) 将会有相等于从执行 CHGV 指令到转速发生实际改变的操作周期的最长响应时间的时间延迟。



7 操作控制程序

F/FS	G
○	○

7.12.2 转矩极限值改变请求: CHGT

格式	CHGT((S1), (S2))	基本步骤数	4
----	------------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(S1)	—	—	—	—	—	○	—	—	—	—	—
(S2)	—	○	○	—	—	○	○	—	○	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(S1)	给出转矩极限值改变请求的轴 No.	—
(S2)	指定的转矩极限值	

[功能]

- 以(S1)指定的轴的转矩极限值改变为以(S2)指定的轴的转矩极限值。
- 在实模式中，任何完成了伺服启动的轴均可随时改变其转矩极限值，且与它的状态，启动，停止，伺服 ON 或伺服 OFF 无关。
- 可在(S1)中设置的轴的号在以下范围内：

Q172CPU(N)	Q173CPU(N)
1 到 8	1 到 32

- 可在(S2)中设置的转矩极限值在 1 到 500[%]的范围内。
- 这里指定的转矩极限值和伺服图表中指定的转矩极限值有如下关系：

启动时

启动时，根据伺服图表中设置的“P.torque”或指定的参数块中的“转矩极限值”，转矩极限值会给定到启动轴的伺服上。

对于插补启动，转矩极限值将会给定到将被插补的轴数上。



执行 CHGT 指令会将预设的转矩极限值只给定到指定的轴上。



随后，在伺服程序启动或 JOG 启动时给定到伺服的转矩极限值仅在它低于 CHGT 中指定的转矩极限值时生效。

转矩极限值固定处理在每个轴进行。

启动期间

- (a) 如果下列转矩限值已经设置，它将不会改变到高于 CHGT 指令中指定的转矩限值。
- 恒定速度控制或速度切换控制中在中间点的转矩限值。
 - 转速 / 位置改变控制中在切换到位置控制的点的转矩限值。
 - 转速控制中的转矩限值。
- (b) CHGT 指令接受高于伺服程序或参数块中设置的转矩限值的转矩限值。
- (6) 通过 CHGT 指令改变的转矩限值仅在电源打开时有效。

[出错]

- (1) 当出现下列情况会发生运算错误且不能进行转矩限值改变：
- (S1) 中指定的轴 No. 在指定范围之外；或
 - (S2) 是间接指定的软元件且它的软元件号在指定范围之外。
- (2) 当出现下列情况会发生轻度出错且不能进行转矩限值改变：
- 以 (S2) 指定的转矩限值在 1 到 500[%] 的范围之外
(轻度出错：311)；或
 - CHGT 指令为任何未启动的轴执行
(轻度出错：312)。

[程序示例]

- (1) 改变轴 2 的转矩限值的程序。

```
CHGT(K2, K10)
```

要点

- (1) 在虚模式中 CHGT 指令无效(忽略)。当在虚模式的运算中改变转矩限值时，请在机械系统程序的输出参数模块中设置“转矩限值软元件”。
- (2) 将会有相等于从执行 CHGT 指令到转矩限值发生实际改变的操作周期的最长响应时间的时间延迟。

7 操作控制程序

F/FS	G
○	○

7.13 其它指令

7.13.1 事件任务使能: EI

格式	EI	基本步骤数	1
----	----	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
—	—	—	—	—	—	—	—	—	—	—	—

○: 可使用

[设定数据]

无设置数据

[功能]

- (1) 允许执行事件任务。
- (2) 指令仅可以和正常任务一起使用。

[错误]

- (1) 当出现下列情况时发生运算出错:
 - 指令和非正常任务一起使用。

[程序示例]

- (1) 允许执行事件任务的程序。

EI

7 操作控制程序

F/FS	G
○	○

7.13.2 事件任务禁止：DI

格式	DI	基本步骤数	1
----	----	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
—	—	—	—	—	—	—	—	—	—	—	—

○：可使用

[设定数据]

无设置数据

[功能]

- (1) 禁止执行事件任务。
- (2) 如果在执行DI指令后发生外部中断或PLC中断，则在执行EI指令后会执行一次对应的事件任务。(如果在DI时发生两次或两次以上的外部中断或PLC中断，在执行EI指令后也只执行一次对应的事件任务。)
- (3) DI时，不执行固定周期事件任务。
- (4) 不能禁止执行NMI任务。
- (5) DI状态在电源打开或以RESET/L.CLR切换进行复位时建立。

[错误]

- (1) 当出现下列情况时发生运算出错：
 - 该指令和非正常任务一起使用。

[程序示例]

- (1) 禁止执行事件任务的程序：

DI

7 操作控制程序

F/FS	G
○	○

7.13.3 空：NOP

格式	NOP	基本步骤数	1
----	-----	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
—	—	—	—	—	—	—	—	—	—	—	

○：可使用

[设定数据]

无设置数据

[功能]

(1) 空操作指令，不影响前面的运算。

[错误]

(1) 对于空操作：NOP，没有运算出错。

7 操作控制程序

F/FS	G
○	○

7.13.4 块传送: BMOV

格式	BMOV(D), (S), (n)	基本步骤数	4
----	-------------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D)	—	○	—	—	—	—	○	—	—	—	—
(S)	—	○	—	—	—	—	○	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(D)	传送目标软元件启动号	—
(S)	传送源软元件启动号	
(n)	待传送的字数	

[功能]

- (1) 以(S)指定的字软元件中的n个字的内容被批量传送到以(D)指定的字软元件中的n个字。
- (2) 当(S1)和(S2)在数据类型上不同时, 进行传送之前, 较小数据类型的数据将会转换成较大数据类型的数据。
- (3) 在(D)或(S)指定 Nn(凸轮号)允许凸轮数据的批量传送。在运动控制器中, 相同凸轮号的凸轮数据必须是已经寄存过的。
以 n 指定的传送的字数应当与指定的凸轮号的分辨率相匹配。

凸轮数据写入时

凸轮数据存储区域被重写。

- 在凸轮运转期间也可将数据传送到凸轮数据区域。
请注意在以相同凸轮号进行运转时不要进行写入。

凸轮数据读出时

凸轮数据存储区域被重写。

- 当前设置状态中的凸轮数据被读出。

(4) 可在(D),(S)和(n)中设置的字软元件如下图所示:

设置数据	字软元件 ^(注-2)			凸轮号指定
	Dn	Wn	#n	Nn ^(注-1)
(D)	○	○	○	○
(S)	○	○	○	○
(n)	○	○	○	—

(注-1):“Nn”表示凸轮号。

(注-2): 软元件号不能间接指定。

(5) 可以设置为“Nn”的凸轮号在如下范围内:

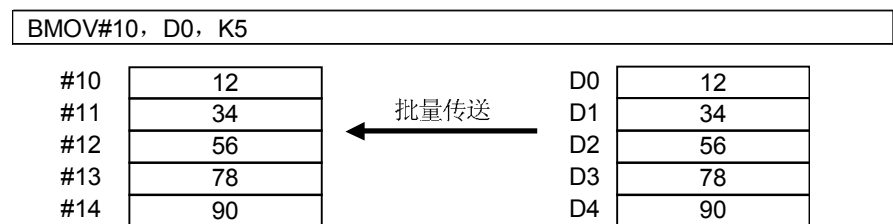
Q173CPU(N)/Q172CPU(N)
1 到 64
101 到 164
201 到 264
301 到 364

[出错]

- (1) 当出现下列情况时会发生运算出错:
- 以(D)或(S)指定的凸轮号的凸轮数据未被寄存到运动控制器中。
 - 以(D)或(S)指定的凸轮号的分辨率与以“n”指定的传送的字数不同。
 - (S)到(S)+(n-1)在软元件范围之外。
 - (D)到(D)+(n-1)在软元件范围之外; 或
 - (n)是零或负数; 或
- } 当指定的(n)是字软元件。
- (2) 当在外围软件的程序编辑中进行转换时, 出现下列情况则会发生运算出错:
- (S)到(S)+(n-1)在软元件范围之外。
 - (D)到(D)+(n-1)在软元件范围之外; 或
 - (n)是零或负数; 或
- } 当指定的(n)是字软元件。

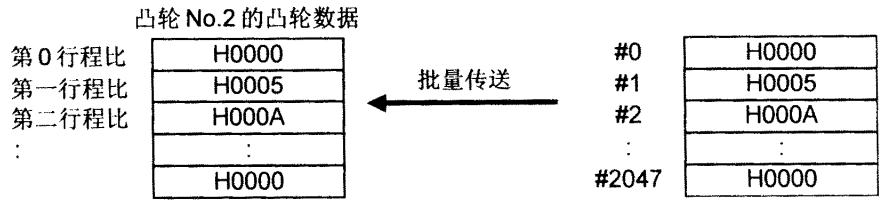
[程序示例]

- (1) 将软元件中以D0开始的数据的5个字批量传送到以#10开始的软元件的5个字中的程序中的程序。



- (2) 将软元件中以#0开始的数据的2048个字批量传送到凸轮 No.2 的数据区域的程序。
(分辨率2048)

BMOV N2, #0, K2048



要点

凸轮行程比设置在 0 到 7FFFH 范围内。

7 操作控制程序

F/FS	G
○	○

7.13.5 将软元件数据写入 CPU 共享内存: MULTW

请参考第“1.3.4”部分运动 CPU 和软件的对应版本。

格式化	MULTW(D), (S), (n), (D1)	基本步骤数	8
-----	--------------------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16 位整数型	32 位整数型 (L)	64 位浮点型 (F)	惯性定时器	16 位整数型 (K/H)	32 位整数型 (K/H, L)	64 位浮点型 (K)			
(D)	—	○	—	—	—	○	—	—	—	—	—
(S)	—	○	—	—	—	—	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—
(D1)	○	—	—	—	—	—	—	—	—	—	—

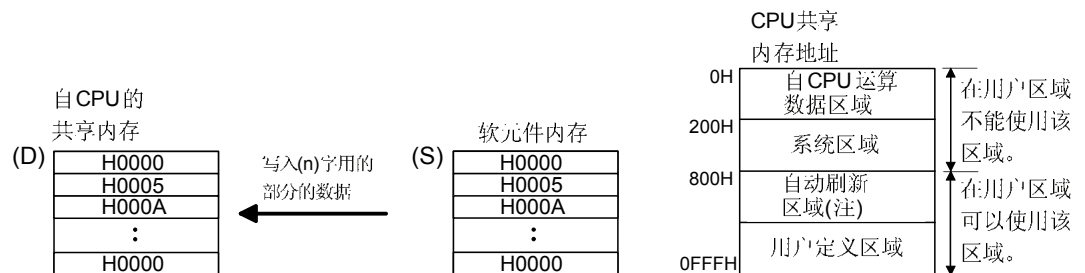
○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(D)	写入目标软元件自 CPU 的 CPU 共享内存地址。 (800H 到 FFFH)	—
(S)	存储写入数据的起始软元件号。	
(n)	将被写入的字数。(1 到 256)	
(D1)	自 CPU 软元件通过写入完成而打开。	

[功能]

- 在自 CPU 模块的软元件以(S)指定后, 数据的(n)字用的部分被写入到自 CPU 模块中以(D)指定的 CPU 共享模块的内存地址中。
在软元件数据写入完成后, 以(D1)指定的完成位软元件打开。



(注): 当未设置自动刷新时, 它不能作为用户定义区域使用。而且, 当自动刷新设置时, 自动刷新发射范围变成了用户定义区域。

- 通过用户程序进行完成位软元件的复位。

(3) 直到执行了 MULTW 指令且完成位软元件打开, 才可以执行另一个 MULTW 指令。当在 MULTW 指令执行完成且完成位软元件打开之前就执行另一个 MULTW 指令, 则另一个 MULTW 指令就会变成出错。

(4) 可在(D), (S), (n)和(D1)中设置的字软元件如下图所示:

设置数据	字软元件 ^(注)			位软元件				
	Dn	Wn	#n	M	B	F	X	Y/PY
(D)	○	○	○	—	—	—	—	—
(S)	○	○	○	—	—	—	—	—
(n)	○	○	○	—	—	—	—	—
(D1)	—	—	—	○	○	○	○	○

(注) 软元件 No.不能间接指定。

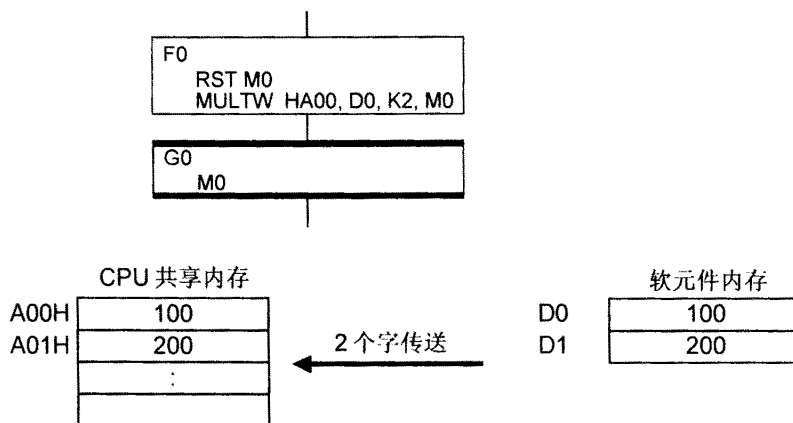
(5) 因为将被写入的字(n)数的变量中 MULTW 指令的处理时间变长, 所以它推荐在正常任务或 7.1ms 以上的事件任务中的执行不要阻碍运动运算的执行。

[出错]

- (1) 当出现下列情况时会发生运算出错:
- 将被写入的字(n)数在 1 到 256 的范围之外;
 - 写入目标软元件的自 CPU 的 CPU 共享内存地址(D)在 800H 到 FFFH 的范围之外。
 - 写入目标软元件的自 CPU 的 CPU 共享内存地址(D)+ 将被写入的字(n)数在 CPU 共享内存地址的 800H 到 FFFH 的范围之外。
 - 存储写入数据的起始软元件号(S)+ 将被写入的字(n)数在软元件范围之外。
 - 当在 MULTW 指令执行完成且完成位软元件打开之前就执行另一个 MULTW 指令。

[程序示例]

(1) D0 的 2 个字被写入到从 A00H 开始 CPU 共享内存中, 并在确认写入完成后转到下一步骤。



7 操作控制程序

F/FS	G
○	○

7.13.6 从其它 CPU 的 CPU 共享内存中读取软元件数据：MULTR

请参考第“1.3.4”部分运动 CPU 和软件的对应版本。

格式	MULTW(D), (S1), (S2), (n)	基本步骤数	7
----	---------------------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D)	—	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

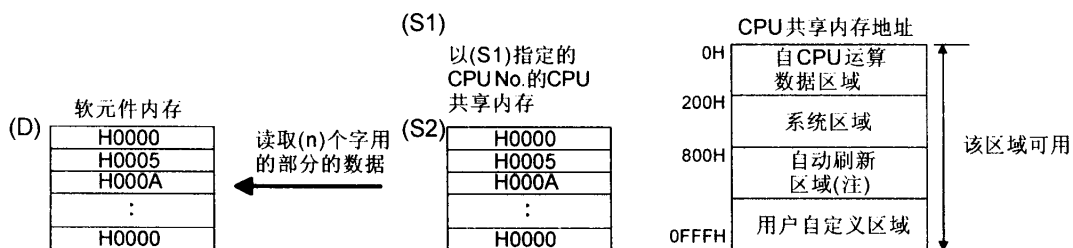
○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(D)	存储读取数据的起始软元件号	—
(S1)	将被读取的 PLC CPU/运动 CPU 的起始 I/O 号。 (CPU No.1: 3E0H, CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H)	
(S2)	将被读取的数据的共享 CPU 内存起始地址。 (000H 到 FFFH)	
(n)	将被读取的字数。(1 到 256)	

[功能]

- (1) 以(S1)指定的其它CPU数据的(n)个字用的部分从CPU共享内存中以(S2)指定的地址中读取，且在以(S2)指定软元件后存储。



(注): 当未设置自动刷新时, 它不能作为用户定义区域使用。而且, 当自动刷新设置时, 自动刷新传送范围变成了用户自定义区域。

(2) 可在(D), (S), (n)和(D1)中设置的字软元件如下图所示:

设置数据	字软元件 ^(注)			位软元件				
	Dn	Wn	#n	M	B	F	X	Y/PY
(D)	○	○	○	—	—	—	—	—
(S)	○	○	○	—	—	—	—	—
(n)	○	○	○	—	—	—	—	—
(D1)	○	○	○	—	—	—	—	—

(注) 软元件号不能间接指定。

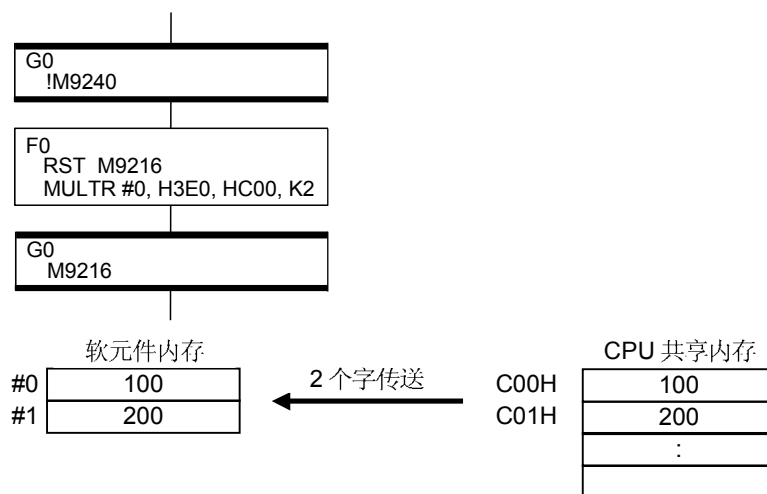
- (3) 当数据从以(S1)指定的目标 CPU 中正常读取时, 与目标 CPU 对应的读取完成标志 M9216 到 M9219(CPU No.1: M9216, CPU No.2: M9217, CPU No.3: M9218, CPU No.4: M9219)会打开。如果数据不能正常读取, 与目标 CPU 对应的读取完成标志不会打开。
- (4) 因为将被写入的字(n)数的变量中 MULTW 指令的处理时间变长, 所以它推荐在正常任务或 7.1ms 以上的事件任务中的执行不要阻碍运动运算的执行。
- (5) 当多重 MULTR 指令同时执行到相同 CPU 时, 读取完成标志 M9216 到 M9219 打开 / 作为最后执行的 MULTR 的结果。
- (6) 使用用户程序复位读取完成标志(M9126 到 M9219)。

[错误]

- (1) 当出现下列情况时会发生运算出错:
 - 将被读取的字(n)数在 1 到 256 的范围之外。
 - 将被读取的数据的CPU共享内存起始地址在CPU共享内存地址的000H到FFFH的范围之外。
 - 将被读取的数据的CPU共享内存起始地址(S2)+ 将被读取的字数在 CPU 共享内存地址的 000H 到 FFFH 的范围之外。
 - 存储读取数据的起始软元件号(D)+ 将被写入的字(n)数在软元件范围之外。
 - 在(S1)中设置了异常 3E0H/3E1H/3E2H/3E3H。
 - 自 CPU 以(S1)指定。
 - 复位将被读取的 CPU。
 - 将被读取的 CPU 中检测到错误。

[程序示例]

- (1) 从 CPU NO.1 共享 CPU 内存 C00H 开始读取 2 个字节，读到以 #0 开始的内存中。
当读取完成后转到下一步。



7 操作控制程序

F/FS	G
○	○

7.13.7 将软元件数据写入到智能模块 / 特殊功能模块: TO

请参考第“1.3.4”部分运动 CPU 和软件的对应版本。

格式	TO(D1), (D2), (S), (n)	基本步骤数	7
----	------------------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D1)	—	○	—	—	—	○	—	—	—	—	—
(D2)	—	○	—	—	—	○	—	—	—	—	—
(S)	—	○	—	—	—	—	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

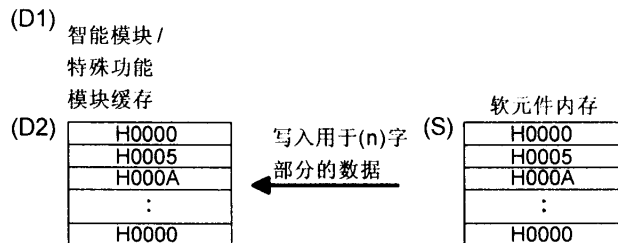
○: 可使用

[设定数据]

设定数据	说明	结果的数据类型
(D1)	智能模块/特殊功能模块的起始 I/O 号(000H 到 FF0H)	—
(D2)	写入数据的缓存的起始地址。	
(S)	存储写入数据的起始软元件号。	
(n)	要写入的字数。(1 到 256)	

[功能]

- 在由(D1)指定的自 CPU 所控制的智能模块 / 特殊功能模块中, 从(S)指定的软元件将数据的(n)字部分写入到用缓存的(D2)指定的开始地址。



- 通过系统设置而设置的模块的起始 I/O 号由(D1)指定。

电源模块	Q02H CPU	Q173 CPU(N)	QX40	Q64AD	Q64DA
			起始 I/O 号 No. 00H	起始 I/O 号 No. 10H	起始 I/O 号 No. 20H

当在 D/A 转换模块(Q64DA)中执行 TO 指令时,(D1)通过系统设置而设置为 20H。

(3) 可在(D1), (D2), (S), (n)中设置的字软元件如下如所示:

设置数据	字软元件 ^(注)			位软元件				
	Dn	Wn	#n	M	B	F	X	Y/PY
(D1)	○	○	○	—	—	—	—	—
(D2)	○	○	○	—	—	—	—	—
(S)	○	○	○	—	—	—	—	—
(n)	○	○	○	—	—	—	—	—

(注): 软元件号不能间接指定。

(4) 调整可执行的任务, 参照运算处理时间的传送字数以使指令不受运动运算执行的阻碍, 因为处理时间在将被写入的字(n)数的变量中变长。

(5) 下列模拟模块可作为运动 CPU 的控制模块使用。

- Q62DA
- Q64DA
- Q68DAV
- Q68DAI
- Q64AD
- Q68ADV
- Q68ADI

[错误]

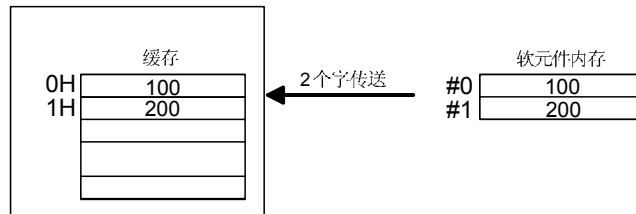
- (1) 当出现下列情况时会发生运算出错:
- 将被写入的字(n)数在 1 到 256 的范围之外。
 - 在执行指令时运动 CPU 不能和智能模块 / 特殊功能模块通讯。
 - 在执行指令时检测到智能模块 / 特殊功能模块异常。
 - 以(D1)指定的 I/O 的号与受自 CPU 控制的智能模块 / 特殊功能模块数不同。
 - 以(D2)指定的地址在缓存的范围之外。
 - 存储写入数据的起始软元件号(S)+ 将被写入的字(n)数在软元件范围之外。

[程序示例]

- (1) 来自 #0 的 2 个字被写入到智能模块 / 特殊功能模块的开始缓存地址中。
(起始 I/O 号: 010H)。

```
TO H010, H0, #0, K2
```

智能模块 / 特殊功能模块
(起始 I/O 号: 010H)



7 操作控制程序

F/FS	G
○	○

7.13.8 从智能模块 / 特殊功能模块中读取软元件数据：FROM

请参考第“1.3.4”部分运动CPU和软件的对应版本。

格式	FROM(D), (S1), (S2), (n)	基本步骤数	7
----	--------------------------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位整数型	32位整数型(L)	64位浮点型(F)	惯性定时器	16位整数型(K/H)	32位整数型(K/H, L)	64位浮点型(K)			
(D1)	—	○	—	—	—	—	—	—	—	—	—
(S1)	—	○	—	—	—	○	—	—	—	—	—
(S2)	—	○	—	—	—	○	—	—	—	—	—
(n)	—	○	—	—	—	○	—	—	—	—	—

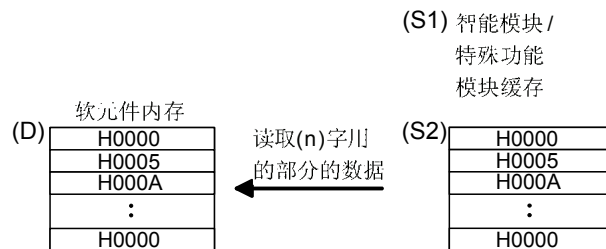
○：可使用

[设定数据]

设定数据	说明	结果的数据类型
(D)	存储读取数据的起始软元件号	—
(S1)	智能模块/特殊功能模块的起始 I/O 号 (000H 到 FF0H)	
(S2)	将被读取的缓存的起始地址。	
(n)	将被读取的字数。(1 到 256)	

[功能]

- (1) 在以(S1)指定的由CPU控制的智能模块/特殊功能模块中，数据的(n)个字用的一部分从缓存中以(S2)指定的地址读取，并在以(S2)指定软元件后存储。



- (2) 通过系统设置而设置的模块的起始 I/O 号由(D1)指定。

电源模块	Q02H CPU	Q173 CPU(N)	QX40 起始 I/O 号 No. : 00H	Q64AD 起始 I/O 号 No. : 10H	Q64DA 起始 I/O 号 No. : 20H	
------	----------	-------------	-------------------------------	--------------------------------	--------------------------------	--

当在 D/A 转换模块(Q64DA)中执行 TO 指令时,(D1)通过系统设置而设置为 20H。

(3) 可在(D), (S1), (S2), (n)中设置的字软元件如下如所示:

设置数据	字软元件 ^(注)			位软元件				
	Dn	Wn	#n	M	B	F	X	Y/PY
(D)	○	○	○	—	—	—	—	—
(S1)	○	○	○	—	—	—	—	—
(S2)	○	○	○	—	—	—	—	—
(n)	○	○	○	—	—	—	—	—

(注): 软元件号不能间接指定。

(4) 调整可执行的任务, 参照运算处理时间的传送字数以使指令不受运动运算执行的阻碍, 因为处理时间在将被写入的字(n)数的变量中变长。

(5) 下列模拟模块可作为运动 CPU 的控制模块使用。

- Q62DA
- Q64DA
- Q68DAV
- Q68DAI
- Q64AD
- Q68ADV
- Q68ADI

[错误]

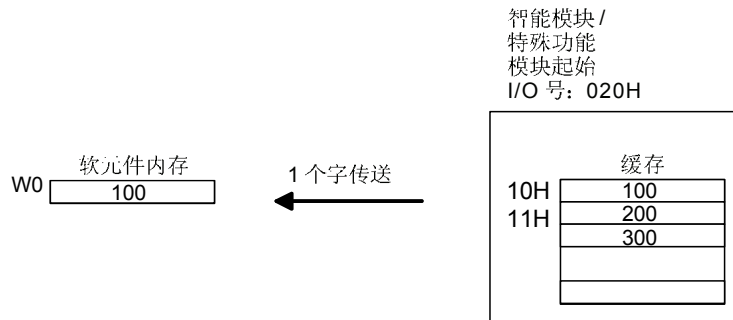
(1) 当出现下列情况时会发生运算出错:

- 将被读取的字(n)数在 1 到 256 的范围之外。
- 在执行指令时运动 CPU 不能和智能模块 / 特殊功能模块通讯。
- 在执行指令时检测到智能模块 / 特殊功能模块异常。
- 以(S1)指定的 I/O 的号与受自 CPU 控制的智能模块 / 特殊功能模块数不同。
- 以(S2)指定的地址在缓存的范围之外。
- 存储读取数据的起始软元件号(D)+ 将被读取的字(n)数在软元件范围之外。

[程序示例]

(1) 从智能模块 / 特殊功能模块(起始 I/O 号: 020H)的缓存地址 10H 中读取 1 个字, 并将它储存到 W0 中。

```
FROM W0, H0, H020, H10, K1
```



7 操作控制程序

F/FS	G
○	○

7.13.9 等待时间：TIME

格式	TIME(S)	基本步骤数	2
----	---------	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	○	○	—	—	○	○	—	—	—	—

○：可使用

[设定数据]

设定数据	说明	结果的数据类型
(S)	等待时间(0 到 2147483647)ms	逻辑型(I/O)

[功能]

- (1) 等待状态持续以(S)指定的时间。当经过时间少于预设时间时结果为假，或当经过预设时间时结果为真且执行转移。
- (2) 当 16 位整数型字软元件用于在(S)中指定 32768 到 65535ms 的任一项时，使用 USHORT 将它转换成不带符号的 16 位整数值。(参考程序示例。)

[错误]

- (1) 当出现下列情况时发生运算出错：
 - (S)是间接指定的软元件且它的软元件号在指定的范围之外；或
 - 以(S)指定的数据(间接指定中软元件数据)在 0 到 2147483647 的范围之外。

[程序示例]

- (1) 设置等待时间为 60s 的程序(当指定常量时。)

```
TIME K60000
```

- (2) 16 位整数型间接指定(#0)用的等待时间为 32768 到 65535ms 时所用的程序。

```
TIME USHORT(#0)
```

- (3) 当指定时间经过时设置(复位)位软元件的程序。

```
SET M100 = TIME K60000
```

要点

- | |
|---|
| <ol style="list-style-type: none">(1) 当等待时间以字软元件间接指定时，接受的第一个值会被用作进行控制所用的软元件值。在等待状态时如果软元件值被改变，设置时间就不能再改变。(2) TIME 指令等同于条件公式，因此可在转移(G)的程序的最后一排设置。(3) 当有 TIME 指令设置的相同数目的转移(Gn)程序在多重运动 SFC 程序中使用，避免将它们同时运转。(如果它们同时运转，程序中运转的第一个等待时间将会变成非法)。(4) 另一个转移程序(Gn)可以通过多重运动SFC同时执行指令的时间。(多重活动步骤小于 256。)(5) 在通过 TIME 指令等待的时间内，等待时间不能停止。(6) 当使用 TIME 指令时如果执行运动 SFC 程序验证，即使当编程软件的运动 SFC 程序等于运动 CPU，也可能发生验证出错。 |
|---|

7 操作控制程序

F/FS	G
○	○

7.14 注释：//

格式	//	基本步骤数	—
----	----	-------	---

[可使用的数据]

设定数据	可使用的数据										
	位软元件	字软元件				常数			计算公式	位条件公式	比较条件公式
		16位 整数型	32位 整数型 (L)	64位 浮点型 (F)	惯性 定时器	16位 整数型 (K/H)	32位 整数型 (K/H, L)	64位 浮点型 (K)			
(S)	—	—	—	—	—	—	—	—	—	—	—

○:可使用

[设定数据]

无设定数据

[功能]

(1) 从//之后到程序块末端的字串是注释。

[错误]

(1) 对于注释无运算出错。

[程序示例]

(1) 已注释替换程序的示例。

D0=D1 //将 D0 值(16 位整数型数据)替换为 D1。

8. 转移程序

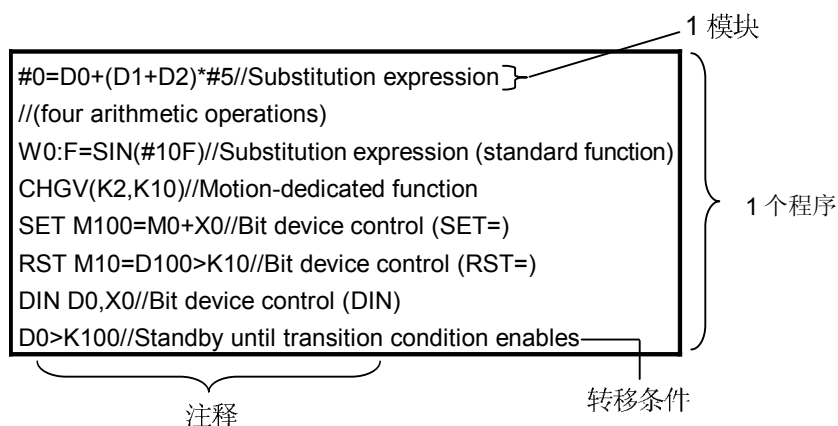
8.1 转移程序

(1) 转移程序

- (a) 替换运算表达式，专有运动函数，位软元件控制指令和转移条件能在转移程序中设置。
- (b) 一个转移程序中可设置多个模块。
- (c) 单个转移程序中可设置的模块数没有限制。
注意:一个程序不得大于 64K 字节。
- (d) 一个模块中可以包含的最大字数是 128。
- (e) 转移条件必须在一个转移程序的最后一个模块中设置。转移程序会一直循环运行下去直到程序的转移条件得到满足，转移条件得到满足后程序就会转移到下一步。
转移条件只能被设置在最后一个模块。
- (f) 在一个模块中没有设置任何运动的程序 (NOP) 可以作为一段特殊的转移程序被生成。

这种程序用在当一段伺服程序完成后你想继续下一步程序，而此时没有什么必要去设置特殊的条件作为互锁。详细的说明请参阅 6.9 节“分支，耦合”。

下面是一个转移程序的例子：



在最后一个模块中可以被设置为转移条件的有：位条件表达式，比较条件表达式和返回逻辑值的软元件设置 (SET=) 或软元件复位 (RST=)，在软元件设置 (SET=) 或软元件复位 (RST=) 的情况下，这些位条件表达和比较表达式被指定的值是真还是假就是转移条件。当转移条件满足时，软元件设置或复位就被执行并且操作就会转移到下一步。

下面给出转移条件内容描述的举例

分类	内容描述的例子
位条件表达式	M0
	!M0+X10*M100
比较条件表达式	(D0>K100)+(D100L!=K20L)
软元件设置 (SET=)	SET Y0=M100
软元件复位 (RST=)	RST M10=D0==K100

要点
(1) 转移程序有别于运行控制程序，在于转移条件是在最后的模块中设置的。其他的设置与运行控制程序的相同。
(2) 当把最后模块中的软元件设置 (SET=) 或软元件复位 (RST=) 设为转移条件时，(S) 中指定的位或比较条件表达式就是不可忽略的。
(3) 只有位条件或比较条件表达式不能在最后模块以外的其他模块中设置。而软元件设置 (SET=) 或软元件复位 (RST=) 在最后模块以外也可以设置。

表 9.2 伺服指令列表 (续)

定位控制	指令符号	处理	定位数据																		其他	WAIT-ON/OFF	步数																	
			通用					圆弧			OSC			参数块						其他																				
			参数块编号	轴地址或移动量	指令速度	延迟时间	M·代码	辅助限制值	辅助点	半径	中心点	导程数	初始角	振幅	频率	基准轴编号*1	控制单位	速度限制值	加速时间	减速时间				快速减速时间	转知界限值	当输入停止时减速处理	圆弧插补的最大允许误差范围	S·曲线比率	循环条件	程序编号	指令速度(恒定速度)	取消	启动	跳跃	FIN加速/减速					
			虚模式有效	轴	地址或移动量	指令速度	延迟时间	M·代码	辅助限制值	辅助点	半径	中心点	导程数	初始角	振幅	频率	基准轴编号*1	控制单位	速度限制值	加速时间				减速时间	快速减速时间	转知界限值	当输入停止时减速处理	圆弧插补的最大允许误差范围	S·曲线比率	循环条件	程序编号	指令速度(恒定速度)	取消	启动	跳跃	FIN加速/减速				
			1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	2	1	2		
			1	—	2	2	1	1	1	2	2	2	2	2	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	2			
螺旋插补控制	指定辅助点	ABH ↗	指定辅助点绝对螺旋插补	△	○	○	△	△	○						△	△	△	△	△	△	△	△	△	△	△	△											10 到 27			
		INH ↗	指定辅助点绝对增量插补	△	○	○	△	△	○							△	△	△	△	△	△	△	△	△	△	△	△													
	指定半径	ABH ↖	小于 CW180°的指定半径绝对螺旋插补 I	△	○	○	△	△		○	○					△	△	△	△	△	△	△	△	△	△	△	△													
		ABH ↗	大于或等于 CW180°的指定半径绝对螺旋插补	△	○	○	△	△		○	○					△	△	△	△	△	△	△	△	△	△	△	△													
		ABH ↖	小于 CCW180°的指定半径绝对螺旋插补	△	○	○	△	△		○	○					△	△	△	△	△	△	△	△	△	△	△	△													
		ABH ↗	大于或等于 CCW180°的指定半径绝对螺旋插补	△	○	○	△	△		○	○					△	△	△	△	△	△	△	△	△	△	△	△													
		INH ↖	小于 CW180°的指定半径增量螺旋插补	△	○	○	△	△		○	○					△	△	△	△	△	△	△	△	△	△	△	△													
		INH ↗	大于或等于 CW180°的指定半径增量螺旋插补	△	○	○	△	△		○	○					△	△	△	△	△	△	△	△	△	△	△	△													
	指定中心点	ABH ↖	指定中心点绝对螺旋插补 CW	△	○	○	△	△			○	○				△	△	△	△	△	△	△	△	△	△	△	△													
		ABH ↗	指定中心点绝对螺旋插补 CCW	△	○	○	△	△			○	○				△	△	△	△	△	△	△	△	△	△	△	△													
		INH ↖	指定中心点增量螺旋插补 CW	△	○	○	△	△			○	○				△	△	△	△	△	△	△	△	△	△	△	△													
		INH ↗	指定中心点增量螺旋插补 CCW	△	○	○	△	△			○	○				△	△	△	△	△	△	△	△	△	△	△	△													
	Fixed-螺距 feed	轴 1	FEED-1	轴 1 定长补给启动	△	○	○	△	△							△	△	△	△	△	△	△	△	△	△	△												4 到 17		
		轴 2	FEED-2	轴 2 线性插补定长补给启动	△	○	○	△	△							△	△	△	△	△	△	△	△	△	△	△	△											5 到 19		
轴 3		FEED-3	轴 3 线性插补定长补给启动	△	○	○	△	△							△	△	△	△	△	△	△	△	△	△	△	△											7 到 21			
速度控制(I)	正转	VF	正转速度控制(I)启动	△	○	○		△							△	△	△	△	△	△	△	△	△	△	△												3 到 15			
	反转	VR	反转速度控制(I)启动	△	○	○		△							△	△	△	△	△	△	△	△	△	△	△	△														

○：必须设置的项目， △：需要时才设置的项目。

*1：只适用于指定基准轴轴。

*2：(B)指的是位软元件。

表 9.2 伺服指令列表 (续)

定位控制	指令符号	处理	定位数据																	步数													
			通用				圆弧		OSC		参数块						其他																
			参数块编号	轴地址或移动量	指令速度	延迟时间	M·代码	转矩限制值	辅助点	半径	中心点	导程数	初始角	振幅	频率	控制单位	速度限制值	加速时间	减速时间		快速减速时间	转矩界限值	当输入停止时减速处理	圆弧插补的最大允许误差范围	S·曲线比率	循环条件	程序编号	指令速度-恒定速度	取消	启动	跳跃	FIN加速/减速	WAIT-ON/OFF
			虚模式有效	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1		1	1	1	1	1	1	1	1	1	1	1	1	1
速度控制(I)	正转	VVF	正转速度控制(I)启动	△	○	○	△	△							△	△	△	△	△	△	△	△				△					3 到 16		
	反转	VVR	反转速度控制(I)启动	△	○	○	△	△							△	△	△	△	△	△	△	△				△							
速度定位控制	正转	VPF	正转速度定位控制启动	△	○	○	△	△	△						△	△	△	△	△	△	△	△	△				△				4 到 18		
	反转	VPR	反转速度定位控制启动	△	○	○	△	△	△						△	△	△	△	△	△	△	△	△				△						
	再启动	VPSTART	转速度定位控制再启动		○																					△					2 到 4		
速度开关控制		VSTART	速度开关控制启动	△											△	△	△	△	△	△	△	△	△				△				1 到 13		
		VEND	速度开关控制结束																								△				1		
		ABS-1	速度开关控制结束点地址		○	○	○	△	△	△																		△				4 到 9	
		ABS-2			○	○	○	△	△	△																		△				5 到 10	
		ABS-3			○	○	○	△	△	△																		△				7 到 12	
		INC-1	移动到速度开关控制结束点		○	○	○	△	△	△																		△				4 到 9	
		INC-2			○	○	○	△	△	△																		△				5 到 10	
		INC-3			○	○	○	△	△	△																		△				7 到 12	
		VABS	速度开关点绝对指定		○	○		△	△																							4 到 6	
		VINC	速度开关点增量指定		○	○		△	△																								
位置跟踪控制		PFSTART	位置跟踪控制启动	△	○	○	○	△							△	△	△	△	△	△	△	△	△				△				4 到 16		
恒定速度控制		CPSTART1	轴 1 恒定速度控制启动	△	○	○									△	△	△	△	△	△	△	△	△				△		△		3 到 15		
		CPSTART2	轴 2 恒定速度控制启动	△	○	○									△	△	△	△	△	△	△	△	△				△		△		3 到 17		
		CPSTART3	轴 3 恒定速度控制启动	△	○	○									△	△	△	△	△	△	△	△	△				△		△		4 到 17		
		CPSTART4	轴 4 恒定速度控制启动	△	○	○									△	△	△	△	△	△	△	△	△				△		△				
		ABS-1	恒定速度控制中过点的绝对指定		○	○		△	△																		△		△	△	2 到 10		
		ABS-2			○	○		△	△																		△		△	△	3 到 11		
		ABS-3			○	○		△	△																		△		△	△	4 到 12		
		ABS-4			○	○		△	△																		△		△	△	5 到 13		
		ABS↗			○	○		△	△	○																	△		△	△	5 到 14		
		ABS↘			○	○		△	△	○																	△		△	△	4 到 13		
	ABS↻			○	○		△	△	○																	△		△	△				

○：必须设置的项目，△：需要时才设置的项目。
 *1：只适用于指定基准轴速。
 *2：(B) 指出是位软元件件。

表 9.2 伺服指令列表 (续)

定位控制	指令符号	处理	定位数据																	步数													
			通用			圆弧			OSC		参数块							其他															
			参数块编号	轴	地址或移动量	延迟时间	M·代码	转矩限制值	辅助点	半径	中心点	导程数	初始角	振幅	频率	基准轴编号*1	控制单位	速度限制值	加速时间		减速时间	快速减速时间	转矩界限值	当输入停止时减速处理	圆弧插补的最大允许误差范围	S·曲线比率	循环条件	指令速度(恒定速度)	取消	启动	跳跃	FIN加速/减速	WAIT-ON/OFF
			虚模式有效	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○	○	○	○	○	○	○	○	○	○	○	○
步数	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	2	1	2	1	2		
间接字数	1	—	2	2	1	1	1	2	2	2	1	2	2	2	1	1	2	1	1	1	1	1	2	1	*2	—	*2	—	*2	1	*2		
恒定速度控制	ABS ↙	恒速控制中通过点的绝对指定	○	○			△	△	○																		△		△	△	4 到 13		
	ABS ↘		○	○			△	△	○																		△		△	△	5 到 14		
	ABS ↙		○	○			△	△		○																	△		△	△	9 到 13		
	ABS ↘		○	○			△	△		○																	△		△	△	8 到 14		
	ABH ↙	恒速控制中通过点的绝对螺旋指定	○	○			△	△	○		○																△		△		9 到 13		
	ABH ↘		○	○			△	△	○	○																	△		△		9 到 14		
	ABH ↙		○	○			△	△	○	○																	△		△		9 到 14		
	ABH ↘		○	○			△	△		○	○																△		△		9 到 14		
	ABH ↙		○	○			△	△		○	○																△		△		9 到 14		
	ABH ↘		○	○			△	△		○	○																△		△		9 到 14		
	INC-1	恒速控制中通过点的增量指定	○	○			△	△																			△		△	△	2 到 13		
	INC-2		○	○			△	△																			△		△	△	3 到 13		
	INC-3		○	○			△	△																			△		△	△	4 到 13		
	INC-4		○	○			△	△																			△		△	△	5 到 13		
	INC ↙		○	○			△	△	○																		△		△	△	4 到 13		
	INC ↘		○	○			△	△	○																		△		△	△	4 到 14		
	INC ↙		○	○			△	△	○																		△		△	△	5 到 14		
	INC ↘		○	○			△	△	○																		△		△	△	5 到 14		
	INH ↙		恒速控制中通过点的螺旋增量指定	○	○			△	△	○		○																△		△		9 到 13	
	INH ↘			○	○			△	△	○	○																△		△		9 到 14		
	INH ↙	○		○			△	△	○	○																	△		△		9 到 14		
	INH ↘	○		○			△	△	○	○																	△		△		9 到 14		
	INH ↙	○		○			△	△		○	○																△		△		9 到 14		
	INH ↘	○		○			△	△		○	○																△		△		9 到 14		
	CPEND	恒速控制结束					△																								1 到 2		

○：必须设置的项目，△：需要时才设置的项目。
 *1：只适用于指定基准轴速。
 *2：(B)指出是位软件元件。

表 9.2 伺服指令列表 (续)

定位控制	指令符号	处理	定位数据																				默认								
			通用					圆弧			OSC		参数块							其他											
			参数块编号	轴地址或移动量	指令速度	延迟时间	M·代码	转矩限制值	辅助点	半径	中心点	导程数	初始角	振幅	频率	基准轴编号*1	控制单位	速度限制值	加速时间	减速时间	快速减速时间	转矩界限值		当输入停止时减速处理	圆弧插补的最大允许误差范围	S·曲线比率	循环条件	程序编号	指令速度(恒定速度)	取消启动	跳跃
虚模式有效	○	○	○	○	○	○	—	○	○	○	—	—	—	○	—	○	○	○	○	—	—	○	○	○	○	○	—	○	○	○	
步数	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	2	2	1	2	1	2	
间接字数	1	—	2	2	1	1	1	2	2	2	1	2	2	2	1	2	1	1	1	1	1	2	1	*2	—	2	*2	—	*2	1	*2
相同控制的重叠 (适用于速度开关控制和非恒定速度控制)	FOR-TIMES																								○						
	FOR-ON	重复区域起始设置																							○						
	FOR-OFF																								○						
	NEXT	重复区域结束设置																													
同时启动	START	同时启动																							○						
起始位置返回	ZERO	原点回归启动	○																												
高速振动	OSC	高速振动	△	○			△						○	○	○							△							△		

○：必须设置的项目，△：需要时才设置的项目。
 *1：只适用于指定基准轴速。
 *2：(B)指出是位软元件件。

9.2 伺服电机虚拟伺服电机轴当前值变更

在实模式下指定轴当前值发生改变。

虚模式下指定虚拟伺服电机轴当前值发生改变。

伺服指令	定位方式	控制轴数量	外围设备上设置的项目																速度变化			
			通用				圆弧			参数块								其他				
			参数块编号	轴地址或移动量	指令速度	延迟时间	M·代码	转矩限制值	辅助点	半径	中心点	控制单位	速度限制值	加速时间	减速时间	快速减速时间	转矩界限值	当输入停止时减速处理		圆弧插补的最大允许误差范围	S·曲线比率	取消
CHGA	绝对	1	○	○																		禁止

○：必须设置的项目

△：需要时才设置的项目

[控制]

使用CHGA进行控制

- (1) 执行的CHGA指令按下面的步骤改变当前值。
 - (a) 相应的指定轴的开始接受标志(M2001 到 M2008或M2001 到 M2032)被打开。
 - (b) 指定轴的当前值改变到指定地址。
 - (c) 启动接受标志在当前值改变完成时断开。

(2) 指定轴当前值在实模式下进行改变。

(3) 虚模式下指定伺服电机轴当前值进行改变。

(4) 使用轴编号可以在以下范围内进行设置。

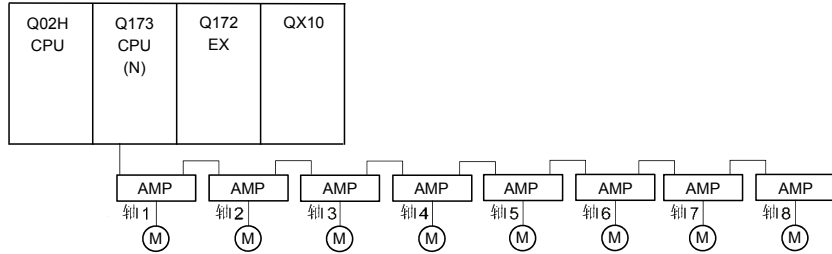
Q172CPU(N)	Q173CPU(N)
轴 1 到 8	轴 1 到 32

(5) CHGA 指令使当前值改变的地址在电源打开后有效。

[程序示例]

给出一段下面条件下在实模式下当前值变更控制程序的描述。

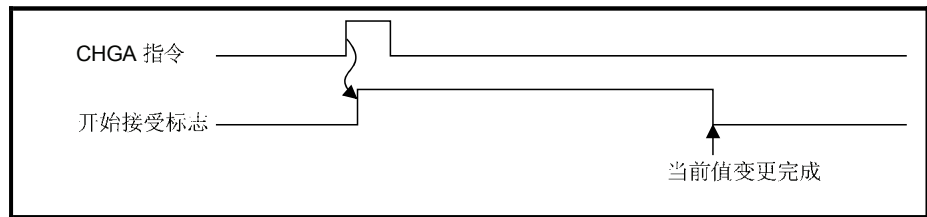
- (1) 系统构成
执行轴 2 的当前值变更控制。



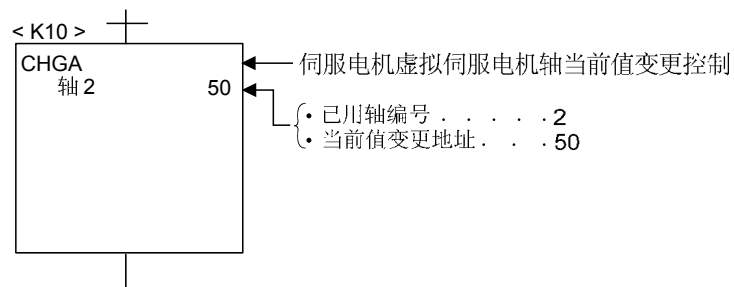
- (2) 当前值变更控制条件
(a) 当前值变更控制条件表示如下

项目	设置
伺服程序编号	10
控制轴编号	2
当前值变更地址	50

- (3) 运动时序



- (4) 伺服程序



要点
<p>(1) 当前值变更指令</p> <ul style="list-style-type: none"> • 当 PLC 就绪标志(M2000)或 PCPU 就绪标志(M9074)断开时，一个次要错误^(注) [100] 将会产生从而使当前值不会发生变化。 • 这种变更只有在停机时才会发生。如果当前值要变更时指定的轴启动了，一个次要错误^(注) [101]（相应轴的开始接受信号打开）就会产生因而当前值变更就会停止。 • 如果相应轴的伺服电机未就绪，一个主要错误^(注) [1004] 将会发生因而当前值变更就不会发生。 • 如果相应轴遇到伺服错误，一个主要错误^(注) [1005] 将会发生因而当前值变更就不会发生。 <p>对于 SV22</p> <ul style="list-style-type: none"> • 设置虚拟伺服电机轴当前值变更程序时请在“程序模式分配”设置的虚拟模式程序编号范围内进行。 • 设置伺服电机(输出)轴当前值变更程序时请在实模式程序编号范围内进行。 • 如果一个虚拟伺服电机轴当前值变更在实模式下运行，一个伺服程序错误^(注) [903] 发生因而当前值变更就会停止。 • 如果伺服电机(输出)轴当前值变更程序在虚模式下运行，一个伺服程序错误^(注) [904] 发生因而当前值变更就会停止。 • 如果在模式发生改变时当前值变更，一个伺服程序错误^(注) [907] (实 → 虚)或[908] (虚 → 实)发生因而当前值变更就会停止。

(注)：对于次要、主要和伺服程序错误的相关内容请参阅“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册(实模式)”或“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22)编程手册(虚模式)”。

9 运动控制程序

9.3 同步编码轴当前值变更控制（只用于 SV22）

指定同步编码轴当前值的变更在虚模式下。

伺服指令	定位方式	控制轴数量	外围设备上设置的项目														速度变化					
			通用				圆弧		参数块						其他							
			参数块编号	轴地址或移动量	指令速度	延迟时间	M·代码	转矩限制值	辅助点	半径	中心点	控制单位	速度限制值	加速时间	减速时间	快速减速时间		转矩界限值	当输入停止时减速处理	圆弧插补的最大允许误差范围	S·曲线比率	取消
CHGA-E	绝对	1	○	○																		不可能

○：必须设置的项目

△：需要时才设置的项目

[控制]

使用CHGA-E指令进行控制

- (1) 执行的CHGA-E指令按下面的步骤改变同步编码轴的当前值。
 - (a) 相应指定的同步编码轴的变更标志(M2101 到 M2112)被打开。
指定轴的当前值改变到指定地址。
 - (b) 指定同步编码轴的当前值改变到指定地址。
 - (c) 当前值改变完成时同步编码轴变更标志被断开。
- (2) 已用轴编号可以在以下范围内进行设置。

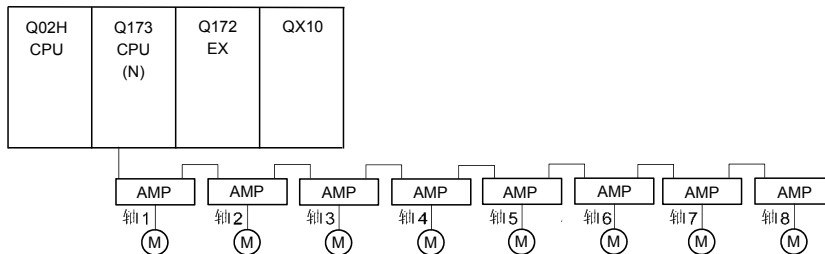
Q172CPU(N)	Q173CPU(N)
轴 1 到 8	轴 1 到 32

- (3) CHGA-E 指令使当前值改变的地址在电源断开后也有效。

[程序示例]

给出一段下面条件下在实模式下当前值变更控制程序的描述。

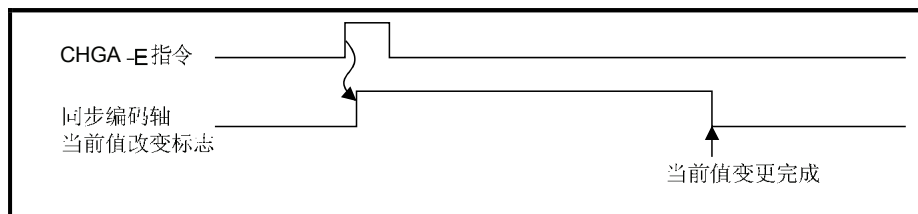
- (1) 系统构成
 执行同步编码轴 P1 的当前值变更控制。



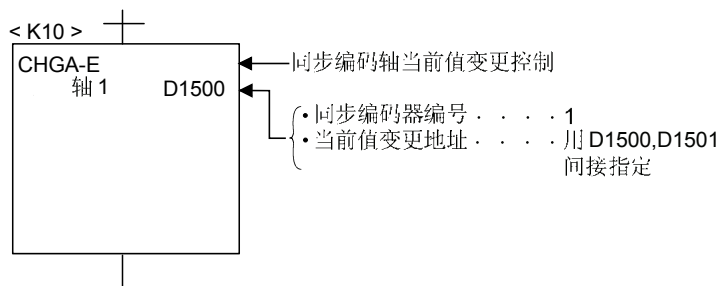
- (2) 当前值变更控制条件
 (a) 当前值变更控制条件表示如下。

项目	设置
伺服程序编号	10
控制轴编号	2
当前值变更地址	50

- (3) 运动时序



- (4) 伺服程序



要点
<p>(1) 同步编码器当前值变更指令</p> <ul style="list-style-type: none"> • 如果在虚模式下操作同步编码器当前值就会发生变更(在同步编码器进行脉冲输入期间)。 如果当前值发生改变, 反馈当前值就以新当前值为基础继续。 • 同步编码器当前值变更不会影响到输出模块的当前值。 • 设置同步编码轴当前值变更程序时请在“程序模式分配”规定的虚模式编号范围内进行。 • 当 PLC 就绪标志(M2000)或 PCPU 就绪标志(M9074)关闭时, 一个次要错误^(注) [100] 将会产生从而使当前值不会发生变化。 • 如果一个同步编码器当前值变更发生在实模式下, 一个伺服程序错误^(注) [903] 或 [905] 就会发生因而当前值变更就会停止。([903]适用于当前值变更程序在虚模式程序编号范围内设置时, 而[905] 适用于当前值变更程序在实模式程序编号范围内设置时。) • 如果在模式发生改变时当前值变更, 一个伺服程序错误^(注) [907](实→虚) 或[908](虚→实) 发生因而当前值变更就会停止。

(注): 对于次要、主要和伺服程序错误的相关内容请参阅“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册(实模式)”或“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22)编程手册(虚模式)”。

9 运动控制程序

9.4 一转内凸轮轴当前值变更控制(只用于 SV22)

指定凸轮轴单圈旋转当前值的变更在虚模式下。

伺服指令	定位方式	控制轴数量	外围设备上设置的项目																速度变化			
			通用						圆弧			参数块						其他				
			参数块编号	轴地址或移动量	指令速度	延迟时间	M·代码	转矩限制值	辅助点	半径	中心点	控制单位	速度限制值	加速时间	减速时间	快速减速时间	转矩界限值	当输入停止时减速处理		圆弧插补的最大允许误差范围	S·曲线比率	取消
CHGA-C	绝对	1	○	○																		禁止

○：必须设置的项目

△：需要时才设置的项目

[控制]

使用 CHGA-C 指令进行控制

- 执行中的 CHGA-C 指令改变指定的凸轮轴单周当前值到相应地址。
- 凸轮轴可以启动。
- 已用轴编号可以在以下范围内进行设置。

Q172CPU(N)	Q173CPU(N)
轴 1 到 8	轴 1 到 32

- CHGA-C 指令使当前值改变的相应地址在电源断开后也有效。

[程序示例]

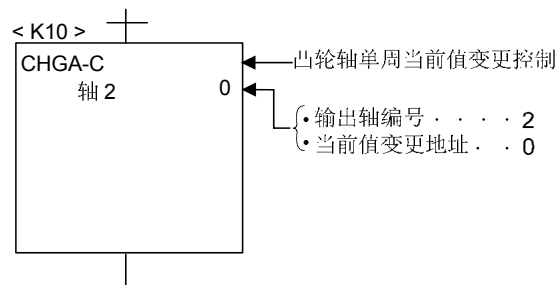
给出一段在下面条件下凸轮轴单周当前值变更控制程序的描述。

(1) 当前值变更控制条件

(a) 当前值变更控制条件表示如下

项目	设置
伺服程序编号	10
输出轴编号	2
当前值变更地址	0

(2) 伺服程序



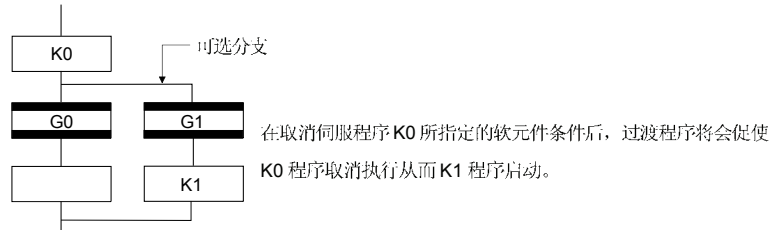
要点
<p>(1) 凸轮轴单圈旋转当前值变更指令</p> <ul style="list-style-type: none"> • 如果一个新的单圈旋转当前值超出了 0 到（单圈旋转当前值脉冲数-1）的范围，次要错误^(注) [6120]就会产生因而当前值不会发生变化。 • 设置凸轮轴单圈旋转当前值变更程序时请在“程序模式分配“规定的虚模式编号范围内进行。 • 当 PLC 就绪标志(M2000)或 PCPU 就绪标志(M9074)断开时，一个次要错误^(注) [100]将会产生从而使当前值不会发生变化。 • 如果一个凸轮轴单圈旋转当前值变更发生在实模式下，一个伺服程序错误^(注) [903] 或 [905]就会发生因而当前值变更就会停止。([903]适用于当当前值变更程序在虚模式程序编号范围内设置时，而[905] 适用于当当前值变更程序在实模式程序编号范围内设置时。) • 如果在模式发生改变时当前值变更，一个伺服程序错误^(注) [907]（实 → 虚）或[908]（虚→实）发生因而当前值变更就会停止。

(注)：对于次要、主要和伺服程序错误的相关内容请参阅“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册(实模式)”或“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22)编程手册(虚模式)”。

9.5 编程指令

9.5.1 取消·启动

当一个取消启动被设置在已经在运动SFC程序的运动控制步中启动的伺服程序的设置项中时，正在运行的伺服程序的取消操作将是合法的但在取消操作后的启动操作将被忽略，指定的伺服程序也不会被启动。



9.5.2 使用运动软元件间接指定

- (1) #0 到 #8191 运动寄存器不能在机械系统程序中被间接指定。
运动寄存器的值使用在伺服或机械系统程序中，用以替代数据寄存器(D)或链接寄存器(W)。
- (2) 在伺服程序和机械系统程序中惯性定时器(FT)不能用来进行间接指定。

10 运动软元件

10.运动寄存器

运动寄存器(#0 到 #8191)和惯性定时器(FT)作为CPU 专有软元件是可用的。
它们适用于操作控制 (F/FS) 程序或转移 (G) 程序中。

10.1 运动寄存器 (#0 到 #8191)

运动软元件	项目	参考手册
运动寄存器 (#)	点的数量	8192 点 (#0 到 #8191)
	数据大小	16 位/点
	锁存	只有一个用户的软元件被锁存。 (所有的点都执行清空操作。)
	可用任务	正常事件和 NMI
	存取	全局范围的读写操作

(1) 运动寄存器列表

(a) 所有操作系统的共有点

软元件号	应用	信号方向
#0 到	用户软元件 (8000 点)	• 被锁存器清除
#8000 到 #8063	运动 SFC 专有软 元件 (64 点)	• 只有在上电或复位时被清除 ^(注-1) • 在运动 SFC 历史错误请求标志为 ON 时清除(上电或复 位时保持) ^(注-2)
#8064 到 #8191	伺服监视软元件 (128 点)	• 只有在上电或复位时被清除。

(注 -1): SW6RN-SV13Q □ /SV22Q □(Ver.com 或更早)。

(注 -2): SW6RN-SV13Q □ /SV22Q □(Ver.com 或更新)。

要点

运动寄存器#不能作为机械系统程序间接指定的软元件被设置。

10 运动软元件

1) 运动 SFC 专有软元件(#8000 到 #8063)

下面表示的就是运动 SFC 专有软元件。

当信号方向为“状态”时软元件表示刷新周期, 当信号方向为“指令”时则为取周期。

软元件号	信号名称	信号方向		刷新周期	导出周期
		状态	指令		
#8000 到	第七个历史出错信息 (最久的出错信息)	运动 SFC 历史出 错(8 个错误) (64 点)	○	—	当出错发生时
#8008 到	第六个历史出错信息				
#8016 到	第五个历史出错信息				
#8024 到	第四个历史出错信息				
#8032 到	第三个历史出错信息				
#8040 到	第二个历史出错信息				
#8048 到	第一个历史出错信息				
#8056 到 #8063	最新出错信息				

a) 运动 SFC 历史出错软元件

CPU 上电后发生的出错信息最多可以有八个作为历史信息存储下来。最近发生的错误存储在软元件 #8056 到 #8063 中。所有出错，包括运动 SFC 控制出错、常规的次要、主要、伺服、伺服程序以及模式转换错误，都可以存储在历史信息中。错误发生时，“运动 SFC 异常(M2039)”也同时被设置。

编号	信号名称		内容	
			运动 SFC 控制错误	常规错误
+0	出错的运动 SFC 程序编号		0 到 255 :发生错误的运动 SFC 程序编号 -1 :独立于运动 SFC 程序之外	-1
+1	出错类型		1 :F/FS 2 :G -1 :K 或其他 (非 F/FS, G 和 SFC 图)。 -2 :运动 SFC 图	3 :次要/主要错误 4 :次要/主要错误(虚拟伺服电机轴) (只有 SV22) 5 :次要/主要错误(不对称编码轴) (只有 SV22) 6 :伺服错误 7 :伺服程序错误 8 :模式转换错误 (只有 SV22) 9 :手动脉冲发生器轴设置错误 10 :模式测试请求错误 11 :WDT 错误 12 :个人电脑连接通讯错误
+2	出错的程序编号		0 到 4095 : F/FS, G, K 程序编号 0 到 255 : GSUB 程序编号 -1 :独立于 F/FS, G, K, GSUB	0 到 4095 : 当出错类型为 “3”、“4”、“7” 时为伺服程序编号 -1 :其他
+3	出错的模块编号/运动 SFC 列表的行编号/轴编号		0 到 8191 : F/FS 或 G 程序的模块编号 (行号), 当出错类型为 “1” 或 “2” 时 0 到 8188 :运动 SFC 列表行编号, 当出错类型为 “-2” 时 -1 :独立于模块当出错类型为 “-1 “ 或为 “1” 或 “2”	1 到 32 :对应的轴编号当出错类型时 “3” 到 “6” 中的任一个时 -1 :其他
+4	出错代码		16000 以后, 包含 16000 (请参阅 “18 出错码列表”)	<ul style="list-style-type: none"> 当出错类型是 “3” 到 “6” 的任意值时, 常规出错代码(小于)。 当出错类型为 “7” 时, 出错代码存储在 D9190 中。 当出错类型为 “8” 时, 出错代码存储在 D9193 中。 当出错类型为 “9” 或 “10” 时为-1 当出错类型为 “11” 时, 出错代码存储在 D9184 中。 当出错类型为 “12” 时, 出错代码存储在 D9196 中。
+5	出错发生时间	年/月	设置 PLC 时钟数据(D9025, D9026, D9027) (BCD 码, 年数以其最后两位表示)	
+6		日/时		
+7		分/秒		

b) 运动SFC异常检测标志(M2039)

(刷新周期: 扫描时间)

当运动 CPU 检测到任何故障时就会设置运动 SFC 异常(M2039)。故障发生时，数据会按照下一步骤被设置到出错软元件中。

- 1) 把出错代码设置到每个轴或出错软元件中。
- 2) 打开每个轴或出错的异常检测信号。
- 3) 把出错信息设置到上面所说的“运动 SFC 历史出错软元件(#8000 到 #8063)”中。
- 4) 打开运动 SFC 异常检测标志(M2039)。

在用户程序中读了“运动 SFC 异常检测标志(M2039)”中的历史出错信息后就把它复位之后，一旦新错误产生“运动 SFC 异常 (M2039)”就重新被打开。

要点
(1) 把“运动 SFC 异常检测标志(M2039)”复位并不会使“运动 SFC 历史出错软元件(#8000 到 #8063)”也被复位(清零)。上电后，他们始终控制着历史出错信息。
(2) 设置时钟数据和读时钟请求(M9028)在用户程序中进行。

2) 伺服监视软元件 (#8064 到 #8191)

伺服监视软元件存储了每个轴的关于“伺服放大器类型”、“电机电流值”和“电机速度值”的信息。

下面给出存储数据的具体内容。

轴编号	软元件号	信号名称																		
1	#8064 到 #8067	<table border="1"> <thead> <tr> <th>信号名称^(注-1)</th> <th>信号内容</th> <th>刷新周期</th> <th>信号方向</th> </tr> </thead> <tbody> <tr> <td>+0 伺服放大器类型</td> <td>1 : MR-H-BN 4 : MR-J2S-B 2 : MR-J-B 5 : MR-J2-M 3 : MR-J2-B 6 : MR-J2-03B5 65 : FR-V500</td> <td>当伺服放大器通电时</td> <td rowspan="3">监视软元件</td> </tr> <tr> <td>+1 电机电流值</td> <td>-5000 到 5000 (×0.1[%])</td> <td rowspan="3">3.55ms</td> </tr> <tr> <td>+2 电机速度值</td> <td>-50000 到 50000 (×0.1[r/min])</td> </tr> <tr> <td>+3</td> <td></td> </tr> </tbody> </table>				信号名称 ^(注-1)	信号内容	刷新周期	信号方向	+0 伺服放大器类型	1 : MR-H-BN 4 : MR-J2S-B 2 : MR-J-B 5 : MR-J2-M 3 : MR-J2-B 6 : MR-J2-03B5 65 : FR-V500	当伺服放大器通电时	监视软元件	+1 电机电流值	-5000 到 5000 (×0.1[%])	3.55ms	+2 电机速度值	-50000 到 50000 (×0.1[r/min])	+3	
信号名称 ^(注-1)	信号内容					刷新周期	信号方向													
+0 伺服放大器类型	1 : MR-H-BN 4 : MR-J2S-B 2 : MR-J-B 5 : MR-J2-M 3 : MR-J2-B 6 : MR-J2-03B5 65 : FR-V500					当伺服放大器通电时	监视软元件													
+1 电机电流值	-5000 到 5000 (×0.1[%])					3.55ms														
+2 电机速度值	-50000 到 50000 (×0.1[r/min])																			
+3																				
2	#8068 到 #8071																			
3	#8072 到 #8075																			
4	#8076 到 #8079																			
5	#8080 到 #8083																			
6	#8084 到 #8087																			
7	#8088 到 #8091																			
8	#8092 到 #8095																			
9	#8096 到 #8099																			
10	#8100 到 #8103																			
11	#8104 到 #8107																			
12	#8108 到 #8111																			
13	#8112 到 #8115																			
14	#8116 到 #8119																			
15	#8120 到 #8123																			
16	#8124 到 #8127																			
17	#8128 到 #8131																			
18	#8132 到 #8135																			
19	#8136 到 #8139																			
20	#8140 到 #8143																			
21	#8144 到 #8147																			
22	#8148 到 #8151																			
23	#8152 到 #8155																			
24	#8156 到 #8159																			
25	#8160 到 #8163																			
26	#8164 到 #8167																			
27	#8168 到 #8171																			
28	#8172 到 #8175																			
29	#8176 到 #8179																			
30	#8180 到 #8183																			
31	#8184 到 #8187																			
32	#8188 到 #8191																			

(注-1): 在每个表示出的轴的伺服监视软元件号的值都被加上了“+0, +1 …”。

备注

伺服监视软元件(#8064 到 #8191)在 SW6RN-SV13Q□/SV22Q□ (00D 以后版本)中有效。

10 运动软元件

10.2 惯性定时器 (FT)

运动软元件	项目	规格
惯性定时器(FT)	点数	1 点 (FT)
	数据大小	32 位/点 (-2147483648 到 2147483647)
	锁存	无锁存。通电或复位时被清零，并开始持续计数。
	可用任务	正常，事件和 (NMI)
	存取	只读
	定时器规格	888 μ s 定时器 (当前值 (FT) 每 888 μ s 被累加 1。)

11. 运动 SFC 参数

有两种不同的运动 SFC 参数：“任务参数”以控制任务（正常任务，事件任务，NMI 任务）和“程序参数”以设置每个运动 SFC 程序：
详情如下：

11.1 任务定义

执行运动 SFC 程序时，每个程序的程序参数只可设置一次。
大致可分为下列三种不同的任务。

任务类型	内容
正常任务	运动主周期中执行 (空闲时间)。
事件任务	1. 在固定周期中执行 (0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms). 2. 外部中断(QI60 的 16 点)事件任务输入接通时执行。 3. PLC CPU 中断输入时执行。
NMI 任务	外部中断(QI60 的 16 点) NMI 任务输入接通时执行。

11.2 连续转移与任务操作数

11.2.1 连续转移数

激活步的执行 → 下次转移条件的判定 → 条件满足时进行转移 (激活步的转移) 被定义为对应任务的执行周期中运动 SFC 程序执行控制的一个单一基本操作。此操作作用于激活步数的一次处理。下一周期连续进行此操作。在这种情况下, 下一周期转移条件满足时, 执行转移目的步。

当转移条件满足时 (可连续执行单一基本操作), 表达转移目的步的连续转移控制, 在同一执行周期中接连执行。

在这种情况下, 设置连续转移数。

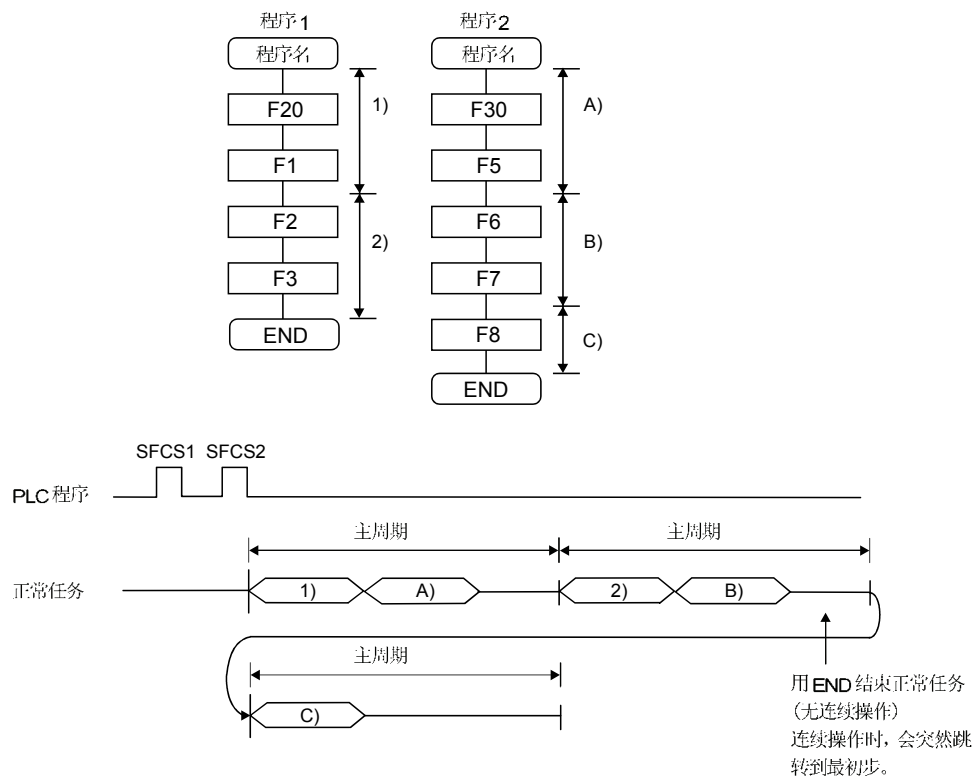
通常用正常任务执行运动 SFC 程序的控制运行。

要点
用事件和 NMI 任务给运动 SFC 的每个程序设置连续转移数。

11.2.2 任务操作

(1) 正常任务的操作
操作]

在运动 CPU 处理的主周期(空闲时间)中执行运动 SFC 程序。



要点]

- a) 包括运动控制步的运动 SFC 程序, 应设置为正常任务。
- b) 执行事件或 NMI 任务时, 取消正常任务的执行。
注意, 正常任务允许在操作控制步中使用事件任务屏蔽指令 (DI), 事件任务不能在事件任务屏蔽指令 (DI) 和事件任务使能指令 (EI) 中间的区域执行。

② 事件任务的操作

操作]

有事件发生时，由事件任务执行运动 SFC 程序。有如下事件：

① 固定周期

周期性执行运动 SFC 程序，周期为 0.88ms, 1.77ms, 3.55ms, 7.11ms 或 14.2ms。

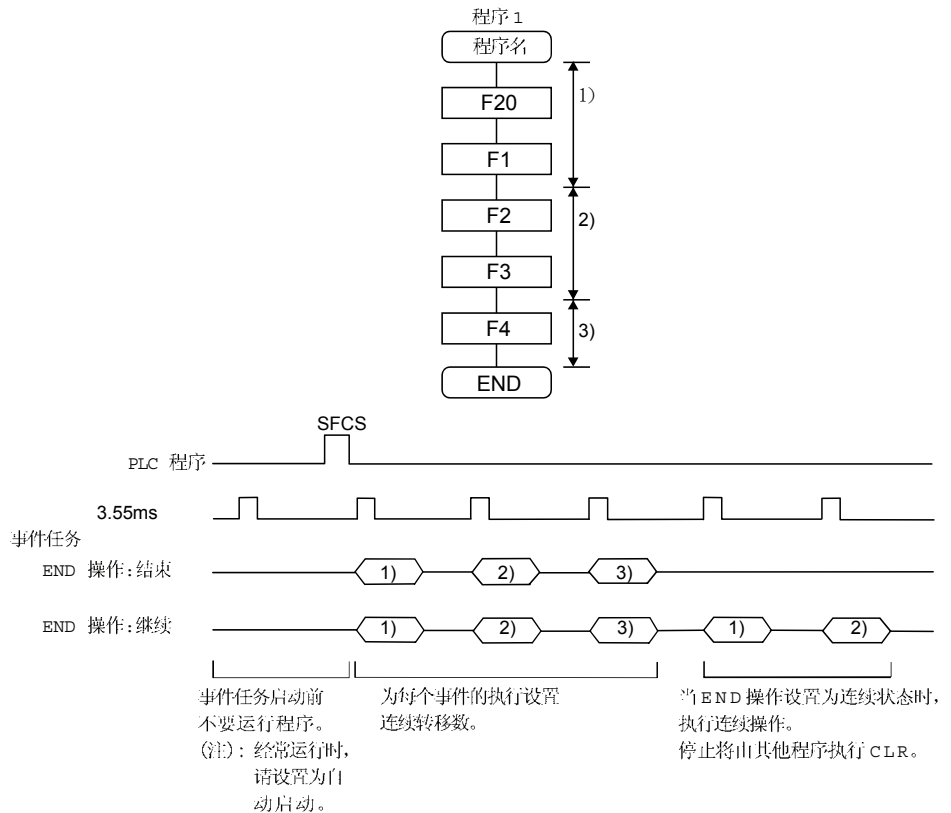
② 外部中断 (I0 到 I15 共 16 个点)

将 QI60 (16 点中断模块)的 16 点装载到运动槽中，当事件任务输入设置启动时，运行运动 SFC 程序。

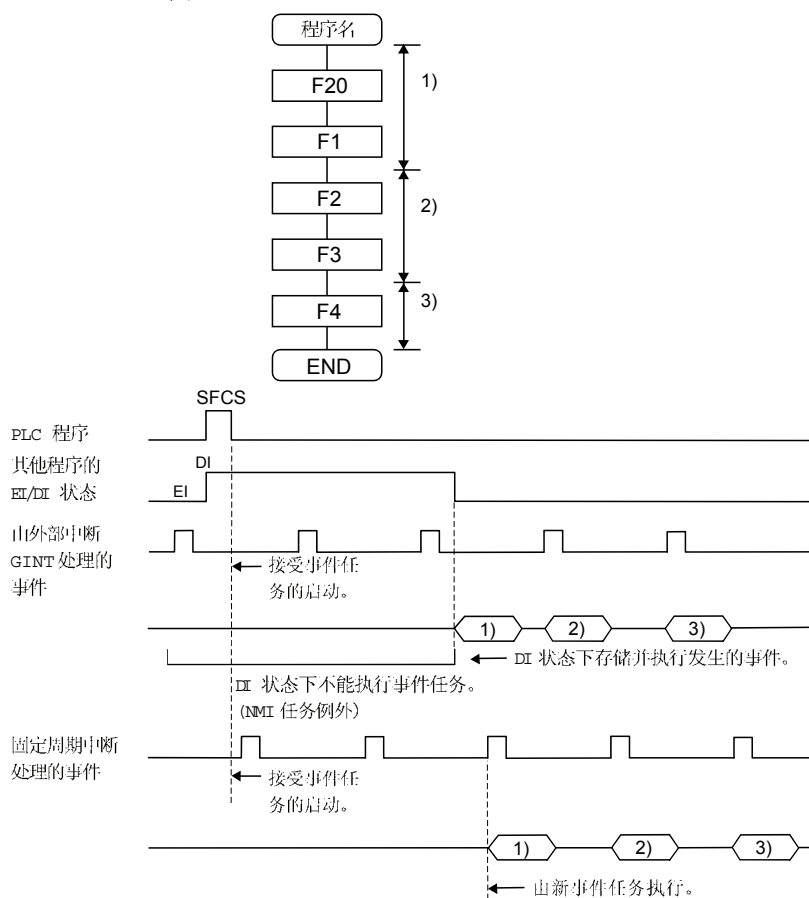
③ PLC 中断

当 PLC 程序执行 S(P).GINT 指令时，执行运动 SFC 程序。

<例 1>



<例 2>



要点]

- a) 在一个运动 SFC 程序中可设置多事件, 但不能设置多个固定周期。
- b) 多运动 SFC 程序可由一个事件执行。
- c) 事件任务中不能执行运动控制步。
- d) 当正常任务使事件任务不可能时, 不能执行该事件任务。事件任务不可能时发生的事件, 在其可能时可被执行。

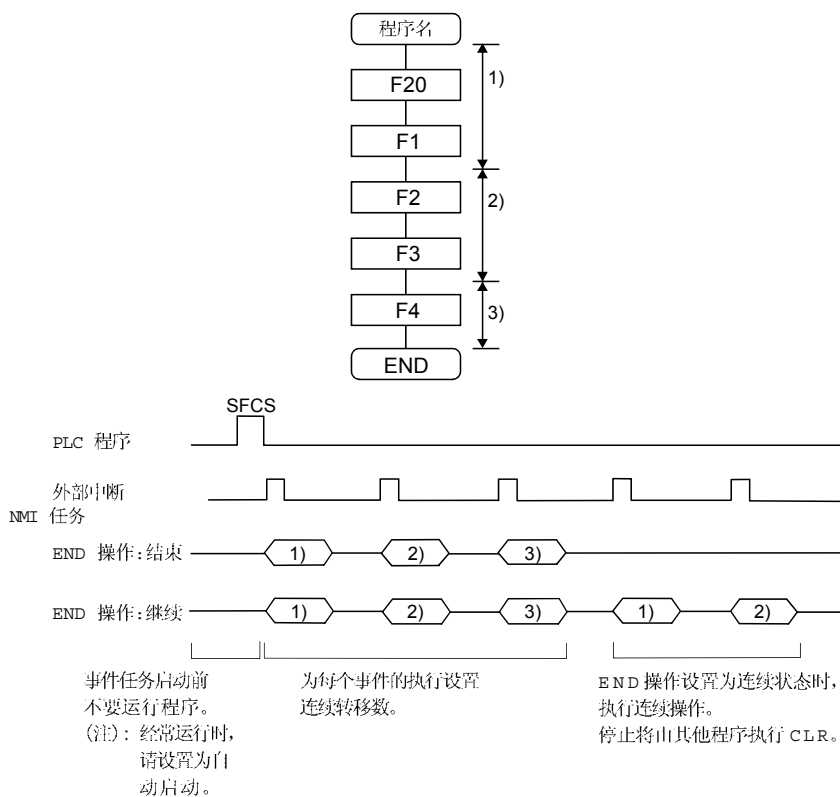
出错]

由事件任务的运动 SFC 程序执行运动控制步时, 会出现运动 SFC 程序错误 [16.113] 并停止其运行。

③ NMI 任务的操作

操作]

外部中断 (QI60 的 16 点) 设置成 NMI 任务的输入接通时, 执行运动 SFC 程序。



要点]

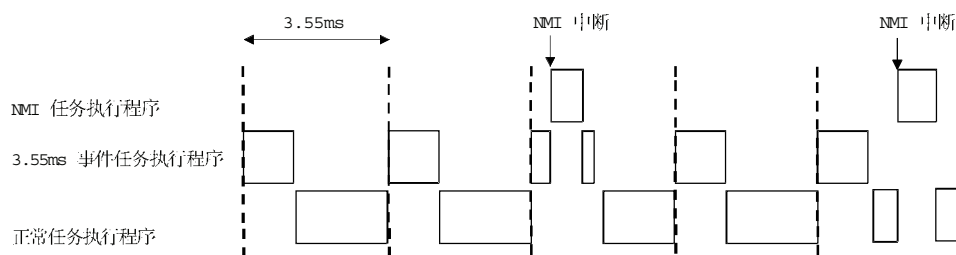
- a) 在正常任务, 事件任务及 NMI 任务中, NMI 任务的优先权最高。
- b) 如果事件任务被正常任务屏蔽, 执行 NMI 任务中断, 而不是屏蔽。

出错]

在 NMI 任务中执行运动控制步。
此时会出现 运动 SFC 程序错误 [16113] 并停止其运行。

11.3 多任务的执行状态

由多任务执行运动 SFC 程序时，各运动 SFC 程序的执行状态如下。



由 NMI 任务执行程序时，3.55ms 固定周期事件任务与由 NMI 任务运行的程序，以及正常任务就如同一张图表。

- ① 3.55ms 固定周期事件任务每隔 3.55ms 执行一次；
- ② 输入 NMI 中断时，NMI 任务有优先执行权；
- ③ 如上所示，正常任务可自由执行。

要点]

将其他任务的执行区域设置为子程序，将子程序设置为其他任务，运动 SFC 程序就可由其他任务部分地执行。

<例>

- | | |
|---------------|------------------|
| 0号.主运动 SFC 程序 | 正常任务 |
| 1号.子程序 | 事件任务 (3.55ms 周期) |



注意

- NMI 任务和事件任务执行多次后，正常任务可能只刚刚执行完一次。

11.4 任务参数

编号	项目		设置项目	初始值	备注
1	连续转移数	正常任务 (通用)	1 到 30	3	启动 PLC 就绪标志 (M2000) 时使用此参数, 并用于之后的控制。 设置/修改参数值时, 断开 PLC 就绪标志 (M2000)。
2	中断设置		不管事件任务或 NMI 任务是否用于外部中断输入(I0 到 I15), 均设置。	事件任务	

(1) 连续转移数

说明]

“激活步的执行 → 下次转移条件的判定 → 条件满足时进行转移 (激活步的转移)”被定义为对应任务的执行周期中运动 SFC 程序执行控制的一个单一基本操作。此操作用于激活步数的一次处理。下一周期连续进行此操作。在这种情况下, 下一周期转移条件满足时, 可执行转移目的步。

当转移条件满足时 (可连续执行单一基本操作), 表示转移目的步的连续转移控制在同一执行周期中接连执行。

在这种情况下, 设置连续转移数。

通常用正常任务执行运动 SFC 程序的控制运行。

要点
用事件和 NMI 任务给运动 SFC 的每个程序设置连续转移数。

出错]

PLC 就绪标志 (M2000) 由断开到接通时, 导入并检查参数。

设置值超出范围时, 出现下列运动 SFC 错误, 用初始值控制。

出错代码 (注)	故障原因		故障处理	对策
	名称	内容		
17000	正常任务连续转移计算错误	正常任务执行的运动 SFC 程序中, 其连续转移计算超出 1 到 30 这一范围。	用初始值 3 控制。	断开 PLC 就绪标志 (M2000), 更正错误, 将值设置在范围之内并写入 CPU。

(注) : 0000H (正常)

② 中断设置

说明]

不管装载到运动槽中QI60 中断模块的 16 中断输入点 (I0 到 I5) 是否作为NMI或者事件任务使用, 均设置。

每点均可自由设置。

所有点均默认为事件任务。

出错]

无错。

11 运动 SFC 参数

11.5 程序参数

为每个运动 SFC 程序设置下列参数。

编号	项目	设置范围	初始值	备注
1	启动设置	是否自动启动	未设置	启动 PLC 就绪标志 (M2000) 时使用此参数, 并用于之后的控制。 设置/修改参数值时, 断开 PLC 就绪标志 (M2000)。
2	执行任务	只能是正常任务, 事件任务或 NMI 任务中的一种。 设置事件任务后, 继续设置激活的事件。 通常设置下列 1 到 3 其中一种。 1. 固定周期 0.88ms, 1.77ms, 3.55ms, 7.11ms, 14.2ms 或无 2. 外部中断 (从设置到事件任务中的选择) 可从 I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 及 I15. 设置中断。 3. PLC 中断 由 OR 设置 1 到 3。 (可重复设置。) 多运动 SFC 程序可共享同一事件。	正常任务	
		设置 NMI 任务后, 继续设置激活的中断输入。 1. 外部中断 (从设置到 NMI 任务中的选择) 可从 I0, I1, I2, I3, I4, I5, I6, I7, I8, I9, I10, I11, I12, I13, I14 及 I15. 设置中断。	无	
3	连续转移数	1 到 10 为事件任务或 NMI 任务的程序设置连续转移数。	1	
4	END 操作	结束/继续 为事件任务或 NMI 任务的程序设置 END 步的操作模式。	结束	

要点

“END 操作”对子程序无效。“END 操作”由“结束”控制。

(1) 启动设置

说明]

下列控制由“是否为自动启动”而改变。

• 正常任务运行的程序

编号	项目	自动启动	非自动启动
1	启动控制	主周期中，打开 PLC 就绪标志 (M2000)后，按正常任务的连续转移数从初始（最初）步启动该程序。	由 PLC 的运动 SFC 启动指令 ($\boxed{S(P).SFCS}$)启动程序，或由运动 SFC 程序的子程序调用/启动(GSUB)。 <ul style="list-style-type: none"> • 由 $\boxed{S(P).SFCS}$ 指令启动 主周期中，执行 $\boxed{S(P).SFCS}$ 指令后，一发生正常任务，即按对应的连续转移数从初始（最初）步启动该程序。 <ul style="list-style-type: none"> • 启动子程序 下一主周期中，执行 GSUB 后，按正常任务的连续转移数从第一步启动该程序。 <ul style="list-style-type: none"> • 调用子程序 同一周期中，从第一步立即执行程序。
		之后，一发生正常任务，即可由对应的连续转移数执行该程序。（子程序调用程序的“执行任务”和“连续转移数”的设置无效，作为正常任务来控制。）	
2	END 控制 \boxed{END}	结束程序本身。 再次由 PLC 的运动 SFC 启动指令 ($\boxed{S(P).SFCS}$)启动程序，或由运动 SFC 程序的子程序调用/启动。.	

• 事件任务运行的程序

编号	项目	自动启动	非自动启动
1	启动控制	打开 PLC 就绪标志 (M2000)后，一发生有效事件，即按对应的连续转移数从初始（最初）步启动该程序。	由 PLC 的运动 SFC 启动指令 ($\boxed{S(P).SFCS}$)启动程序，或由运动 SFC 程序的子程序调用/启动。 <ul style="list-style-type: none"> • 由 $\boxed{S(P).SFCS}$ 指令启动 执行 $\boxed{S(P).SFCS}$ 指令后，一发生正常任务，即按对应的连续转移数从初始（最初）步启动该程序。 <ul style="list-style-type: none"> • 启动子程序 执行 GSUB 后，一发生有效事件，即按对应的连续转移数从第一步启动该程序。 <ul style="list-style-type: none"> • 调用子程序 从第一步立即执行程序。
		之后，一发生有效事件，即可由对应的连续转移数执行该程序。（子程序调用程序，按调用源程序的“执行任务”和“连续转移数”控制。）	
2	END 控制 \boxed{END}	同 END 操作。	

• NMI 任务运行的程序

编号	项目	自动启动时	非自动启动时
1	启动控制	打开 PLC 就绪标志 (M2000)后, 一发生有效事件, 即按对应的连续转移数从初始 (最初) 步启动该程序。	由 PLC 的运动 SFC 启动指令 (S(P).SFCS)启动程序, 或由运动 SFC 程序的子程序调用/启动。 <ul style="list-style-type: none"> • 由 S(P).SFCS 指令启动 执行 S(P).SFCS 指令后, 一发生有效事件, 即按对应的连续转移数从初始 (最初) 步启动该程序。 <ul style="list-style-type: none"> • 启动子程序 执行 GSUB 后, 一发生有效事件, 即按对应的连续转移数从第一步启动该程序。 <ul style="list-style-type: none"> • 调用子程序 从第一步立即执行程序。
		之后, 一发生有效事件, 即可由对应的连续转移数执行该程序。	
2	END 控制 END	同 END 操作。	

出错
无错。

要点
编写正常任务执行的程序, 不以 END 结束程序, 在自动启动时, 会跳回启动步, 恢复初始值。

② 执行任务

说明]

设置时序(任务)来执行程序。

确定程序是否只由“正常任务(主周期), 事件任务(固定周期, 外部中断, PLC 中断)与 NMI 任务(外部中断)”中的一种运行。

设置事件任务时, 即在“固定周期, 外部中断(事件任务)与 PLC 中断”设置多事件。

但是多固定周期不能设置于一个运动 SFC 程序。

例) 中断设置: 输入事件任务 I6, I7, I8, I9, I10, I11, I12, I13, I14 与 I15

运动 SFC 程序 No. 10 - 事件: 固定周期 (3.55ms)

运动 SFC 程序 No. 20 - 事件:

固定周期 (1.77ms) + 外部中断 (I6)

运动 SFC 程序 No. 30 - 事件:

外部中断s (I7, I15) + PLC CPU 中断

设置 NMI 任务时, 即可设置外部中断 (NMI 任务) 的多中断输入。

例) 中断设置: 输入 NMI 任务 I0, I1, I2, I3, I4, I5

运动 SFC 程序 No. 10 - NMI : I0

运动 SFC 程序 No. 20 - NMI : I1 + I2

运动 SFC 程序 No. 30 - NMI : I5

出错]

PLC 就绪标志 (M2000) 由 OFF 到 ON 时, 应用该程序参数, 并在启动运动 SFC 程序(自动启动, 从 PLC 或子程序启动)时检查。

出现不合法值时, 设置下列运动 SFC 错误, 或控制初始值。

出错代码 (注)	故障起因		故障处理	对策
	名称	内容		
17010	执行任务 设置为不合法。	设置正常任务, 事件任务和 NMI 任务的多事件, 或者不设置其中一个。	控制初始值 (正常任务)	关闭 PLC 就绪标志 (M2000), 更正错误, 将值设置在范围之内并写入 CPU。
17011	执行任务 设置为不合法。(事件)	设置事件任务的两个以上固定周期。		

(注) : 0000H (正常)

要点
<p>因可为每个运动 SFC 程序编号设置执行任务，就不必写入单个控制（机器操作）的多程序，以分离执行的定时处理。</p> <p>例如，在正常任务运行运动 SFC 程序条件下，可轻易实现由子程序来开启下列区域：固定周期运行的区域以及外部中断运行的区域。</p>

③ 连续转移数

说明]

给由事件任务或 NMI 任务执行的程序设置连续转移数。
其设置可参照“11.4 任务参数”。

出错]

PLC 就绪标志 (M2000) 由关闭到打开时，导入该程序参数，并在启动运动 SFC 程序 (自动启动，从 PLC 或子程序启动) 时检查。

出现不合法值时，或设置下列运动 SFC 错误，或由初始值控制。

出错代码 (注)	故障起因		故障处理	对策
	名称	内容		
17001	事件任务连续转移计数错误	由事件任务启动的运动 SFC 程序的连续转移数超出 1 到 10 这一范围。	控制 1 的初始值。	关闭 PLC 就绪标志 (M2000)，更正错误，将值设置在范围之内并写入 CPU。
17002	NMI 任务连续转移计数错误	由 NMI 任务启动的运动 SFC 程序的连续转移数超出 1 到 10 这一范围。		

(注) : 0000H (正常)

(4) END 操作

说明]

给由事件任务或 NMI 任务执行的程序设置 END 步的操作。
 详细说明如下。

- NMI 任务运行的程序

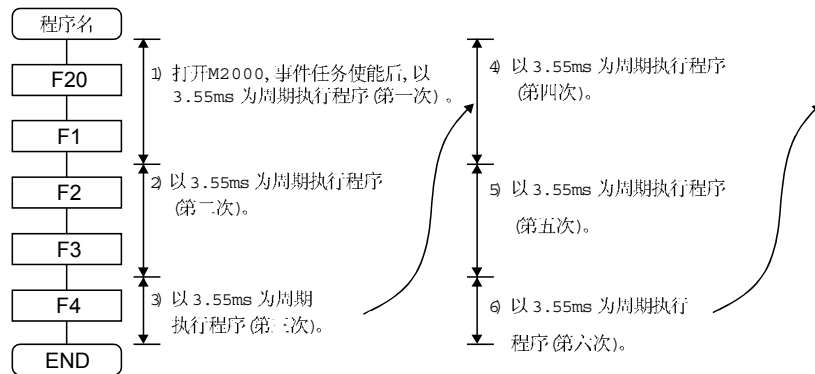
编号	项目	“结束”时	“继续”时
1	END 执行时控制	结束该程序。	由事件/中断来结束，并执行该程序。
2	END 执行后重启	再次由 PLC 的运动 SFC 启动指令 (S(P).SFCS) 启动程序，或由运动 SFC 程序的子程序调用/启动。	下次事件/中断发生时重启，并按对应程序的连续转移数从初始（最初）步运行。 之后，发生事件/中断时，由对应程序的连续转移数执行该程序。
3	结束后由清除步 CLR 重启	再次由 PLC 的运动 SFC 启动指令 (S(P).SFCS) 启动程序，或由运动 SFC 程序的子程序调用/启动。	

要点
子程序调用程序的 END 操作为“结束”控制。

- 以下操作实例假定 END 操作可继续。

程序参数

- 自动启动
- 执行任务事件 3.55ms
- 连续转移数 = 2
- 结束操作“继续”



11.6 如何启动运动 SFC 程序

打开 PLC 就绪标志(M2000)，执行运动 SFC 程序。
运动 SFC 可以由下面三种方法启动。

- ① 自动启动
- ② 从运动 SFC 程序启动
- ③ 从 PLC 启动

在程序参数中，为每个运动 SFC 程序设置启动方式。参数设置请参照“11.5 参数设置之程序参数”。

11.6.1 自动启动

操作]

打开 PLC 就绪标志(M2000)即自动启动。

11.6.2 从运动 SFC 程序启动

操作]

通过执行 SFC 程序中子程序调用 /启动步来启动。
子程序调用 /启动详情请参照“6 运动 SFC 程序”。

11.6.3 从 PLC 启动 (PLC 指令 S(P).SFCS)

执行 PLC 程序中的 S(P).SFCS 可启动 SFC 程序。详情请参照“5 运动专用 PLC 指令”。

11.7 如何结束运动 SFC 程序

操作]

- ① 运动 SFC 程序由其内部的 END 设置结束。
- ② 运动 SFC 程序可通过断开 PLC 就绪标志 (M2000) 来结束。
- ③ 程序可由清除步 结束。
清除步详情请参照 “6.5.4 清除步”。

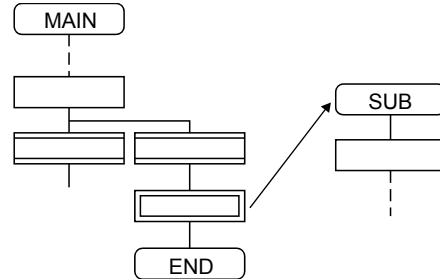
要点]

- ① 一个运动 SFC 程序可设置多个 END。

11.8 如何从一个运动 SFC 程序转换到另一个运动 SFC 程序

启动子程序运行来停止运动 SFC 程序的运行并切换到另一运动 SFC 程序。

启动子程序来改变运动 SFC 程序。



11.9 如何管理执行中程序

无确切信息指定正在执行哪一个种运动 SFC 程序。

使用用户程序 (运动 SFC 程序 / PLC 程序) 来控制执行中的程序。

11.10 CPU 电源断开或重启时的操作

CPU 电源断开或重启时，运动 SFC 程序运行如下所示。

- (1) CPU 电源断开或重启时，运动 SFC 程序停止执行。
- (2) CPU 电源断开或一键重启时，可保存运动寄存器 #0 到 #7999 的内容，需要的时候，在运动 SFC 程序中可初始化这些值。
- (3) 打开 CPU 电源或重启时，运动 SFC 程序的运行如下。
 - 在 PLC 程序下，打开 PLC 就绪标志 (M2000) 可重新运行已设置为自动启动的 SFC 程序。
 - 启动后，其他运动 SFC 程序亦重新执行。

11.11 由 RUN/STOP 切换 CPU 的操作

操作 RUN/STOP 开关时，PLC 就绪标志 (M2000) 按基本系统设置的“STOP 到 RUN 操作”接通 / 断开。

“STOP 到 RUN 操作”详情请参照“1.5.3 个别参数”。

PLC 就绪标志 (M2000) 参照下一节。

11.12 PLC 就绪标志 (M2000) 断开 /打开时的操作

本部分解释 PLC 就绪标志 (M2000) 的接通 /断开相关情况。

在设置基本系统时,在“STOP 到RUN操作”中,PLC 就绪标志 (M2000) 打开 /断开条件的不同。

详情请参照“1.5.3 个别参数”。

[M2000 OFF → ON]

PLC 就绪标志 (M2000) 由断开到接通时若无错误,则 PCPU 就绪标志 (M9074) 亦打开。PCPU 就绪标志 (M9074) 接通时,可执行运动 SFC 程序。

一开始运动 SFC 程序即执行自动启动。

[M2000 ON → OFF]

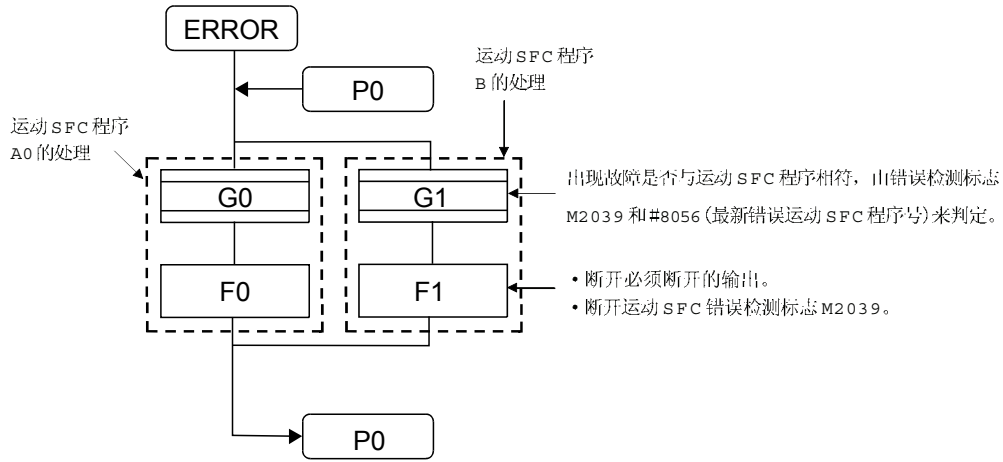
PLC 就绪标志 (M2000) 断开时,运动 SFC 程序停止执行,PCPU 就绪标志 (M9074) 亦断开,因实际输出 PY 的全部点都断开。

要点

PLC 就绪标志(M2000) 断开时,运动 SFC 程序停止但其实际输出 PY 并未断开。
--

11.13 出现故障时的操作

因出现故障而引起运动 SFC 程序的停止时，输出可保存。
出现故障时停止输出，执行下列运动 SFC 程序。



12. 用户文件

用户文件列表及目录结构如下所示。

12.1 工程

以“工程”为基础管理用户文件。

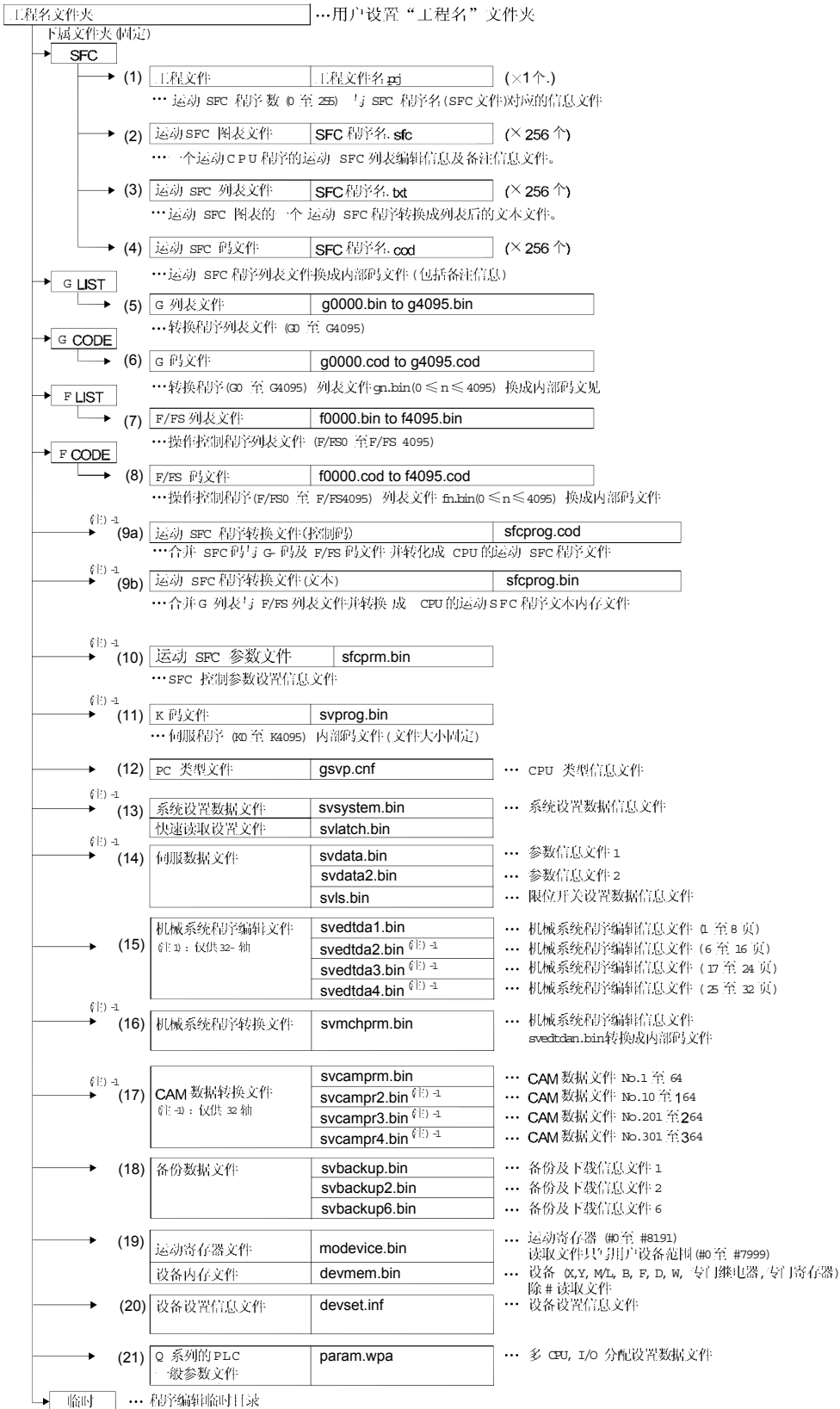
设置“工程名”后，就生成如下页所示的“工程名”文件夹，并按文件类型生成下属文件夹(SFC, G LIST, G CODE, F LIST, F CODE)。在 SFC 下属文件夹下，还生成该项目文件(项目名 .Bj) 及编辑文件夹(临时)。

要点
(1) 显示工程管理时，设置“工程名”。
(2) 该“工程名”长度不得超过 256 个字。
(3) 该“工程路径名”和“工程名”的长度不得超过 256 个字。 (例) "C:\Usr\.....\工程名\"

12.2 用户文件列表

用户文件列表如下所示

注 1) : 表示储存在 CPU 内存中的文件 (数据)



注 1) 仅供 SV22

12.3 运动 SFC 程序的在线修改

在线修改用于在作定位控制时 (M.RUN LED: ON)，将运动 SFC 程序写入内部 SRAM。多 CPU 系统启动时，可重复执行程序更正和操作检测。

可在线修改的数据如下所示。

应用数据		在线修改	备注
系统设置数据	系统设置	×	
	伺服设置数据	×	
运动 SFC 程序	运动 SFC 参数	×	
	运动 SFC 图表	○	仅在程序停止时才能作在线修改。
	操作控制步 (F/FS)	○	
	转换 (G)	○	
	伺服程序 (K)	○	不可作模式任务设置的在线修改。
机械系统程序 (仅供 SV22)		×	
凸轮数据 (仅供 SV22)		×	

○：可能 ×：不可能

要点
<p>① 在定位控制中，可执行程序的在线写入。操作时请小心。</p> <p>② 在 ROM 在线修改操作模式下可将程序写入运动 CPU 的内部 SRAM。此时，其内容在下次电源开启或复位时，与写入内部 FLASH ROM 的程序相同。</p> <p>③ 多位用户同时在线修改一个运动 CPU 程序时，可能无法执行程序的修改。请不要进行此操作。</p> <p>④ 如果在下列操作中，用户个人进行在线修改时，可能出现监视故障或操作失败。请不要进行此操作。</p> <ul style="list-style-type: none"> • 运动 SFC 程序的监视模式 • 测试模式 • 运动 SFC 程序的调试模式 <p>⑤ 执行新加运动 SFC 图表的在线修改时，因为运动 SFC 参数不能在线修改所以它作为一般任务操作 (默认值)。</p> <p>⑥ 在 SV22 模式中，以改变“程序 / 伺服程序编辑页面 - [模式分配设置]”来进行在线修改，其改变的内容无法反映出来。</p> <p>⑦ 外围设备与运动 CPU 电缆连接失败，或者运动 CPU 电源断开或复位时，该程序中断。在程序软件通讯画面下重新写入程序。</p>

12.3.1 在线修改的操作方式

在程序软件下的“程序编辑页面(转换)菜单 [在线修改设置]”，选择运动 SFC 程序的“在线修改 OFF/ON”。运动 SFC 程序的在线修改有以下三种方式。

- 使用程序编辑页面 [SFC 图解写入] 时— 在线修改运动 SFC 程序
- 使用操作控制 /转换程序编辑页面 [转换] 时— 在线修改操作控制 /转换程序编页面 [转换]
- 使用伺服程序编辑页面 [存储] 时— 在线修改伺服程序

① 使用程序编辑页面 [SFC 图解写入] 时

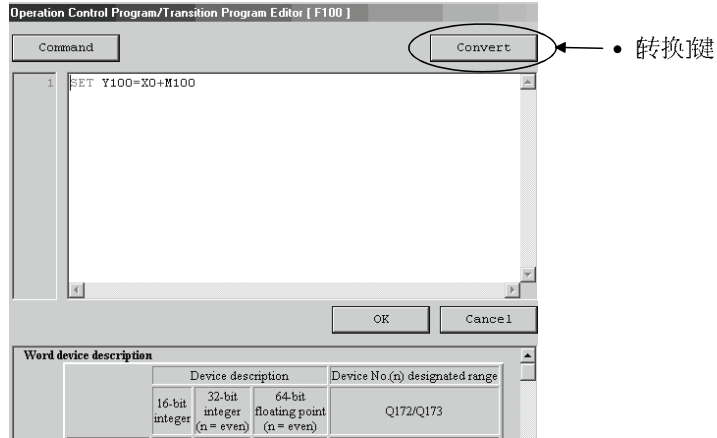
选择 [SFC 图解写入] 键进行运动 SFC 程序的在线修改。

运动 SFC 程序停止时可以进行在线修改。程序执行时进行在线修改会出现警告。

(可用程序批处理监视器检查运动 SFC 程序的执行 /停止状态。)如果在在线修改中出现程序的启动请求，该运动 SFC 会出现启动故障 (出错代码 16007: 在线修改)，程序亦不会启动。



- 使用操作控制 / 转换程序编辑页面 [转换] 时。
 选择 [转换] 键进行操作控制 / 转换程序的在线修改。
 执行操作控制 / 转换程序时，可进行在线修改。
 下次扫描周期可执行已做在线修改的程序。



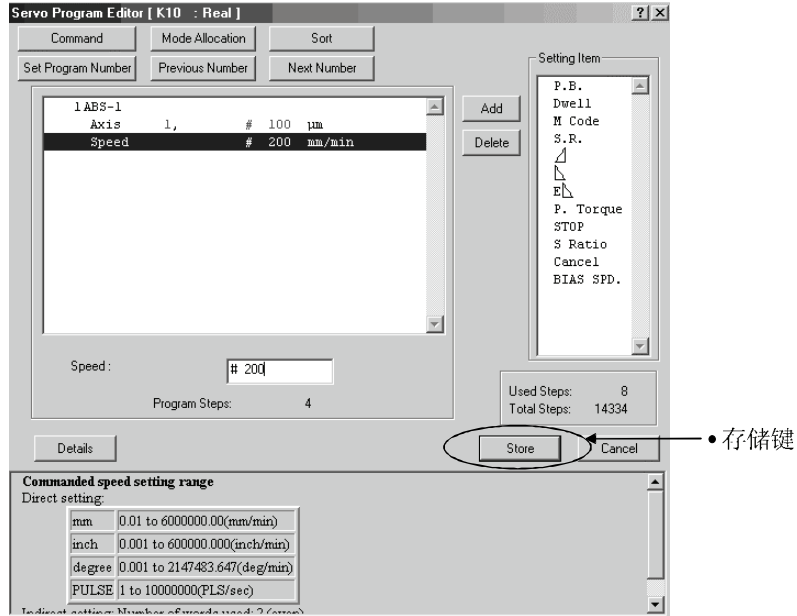
操作控制 / 转换程序的在线修改在下列条件下的操作如下图所示。请小心执行。

程序	条件	操作
<p>FSn</p> <p>↓</p> <p>Gn</p> <p>或</p> <p>FSn</p> <p>↓</p> <p>Gn</p>	<ul style="list-style-type: none"> FSn 等待 Gn 条件满足时，可执行 FSn 操作控制程序的在线修改。 	<ul style="list-style-type: none"> 在线修改完成后，FSn 重复进行在线修改后的操作控制程序直到 Gn 条件满足。
<p>Gn</p> <p>或</p> <p>Gn</p>	<ul style="list-style-type: none"> 等待 Gn 条件满足时，可执行 Gn 程序的在线修改。 (要写入的程序条件句不包括 TIME 指令。) 	<ul style="list-style-type: none"> 在线修改完成后，直到在线修改的操作控制程序条件满足，Gn 才将转换到下一步。
<p>Gn</p> <p>或</p> <p>Gn</p>	<ul style="list-style-type: none"> 等待 Gn 条件完全时，执行 Gn 程序的在线修改，包括 TIME 指令。 	<ul style="list-style-type: none"> 运行写入完成后，不必管 TIME 指令的等待时间和下一步的执行，Gn 就结束了。
<p>Kn</p> <p>或</p> <p>Gn</p>	<ul style="list-style-type: none"> 执行 Kn 伺服程序时，Gn 程序的在线修改。 	<ul style="list-style-type: none"> 执行伺服程序后，执行已改变的 Gn 程序。

- (3) 当使用伺服程序编辑页面 **Store** 时
 选择 **Store** 键在编辑伺服程序时执行在线修改。

执行伺服程序时，可进行在线修改。

下次启动伺服程序时，可执行已做在线修改的程序。



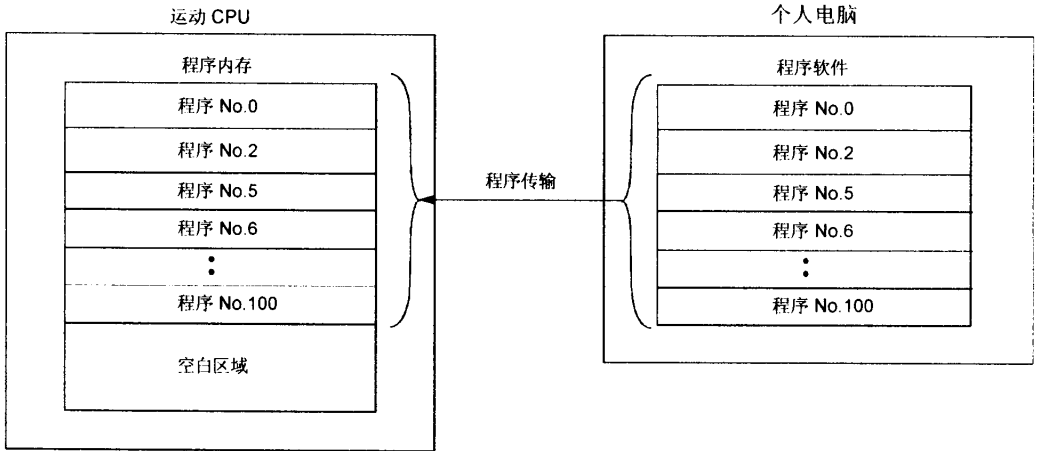
伺服程序的在线修改在下列条件下的操作如下图所示。请小心执行。

程序	条件	操作
<p>ON 位软元件</p> <p>Kn</p> <p>或</p> <p>OFF 位软元件</p> <p>Kn</p>	<ul style="list-style-type: none"> 等待 WAITON/WAITOFF 条件满足，可在 WAITON 时或 WAITOFF 后，执行伺服程序 Kn 的在线修改。 	<ul style="list-style-type: none"> WAITON/WAITOFF 条件满足后，启动未进行在线修改的伺服程序。 *下次启动伺服程序时，可执行在线修改后的程序。
<p>Gn</p> <p>Kn</p> <p>或</p> <p>Gn</p> <p>Kn</p>	<ul style="list-style-type: none"> 等待 Gn 条件满足并执行 Gn 后，进行伺服程序 Kn 的在线修改。 	<ul style="list-style-type: none"> Gn 条件完全后，可执行在线修改后的伺服程序。

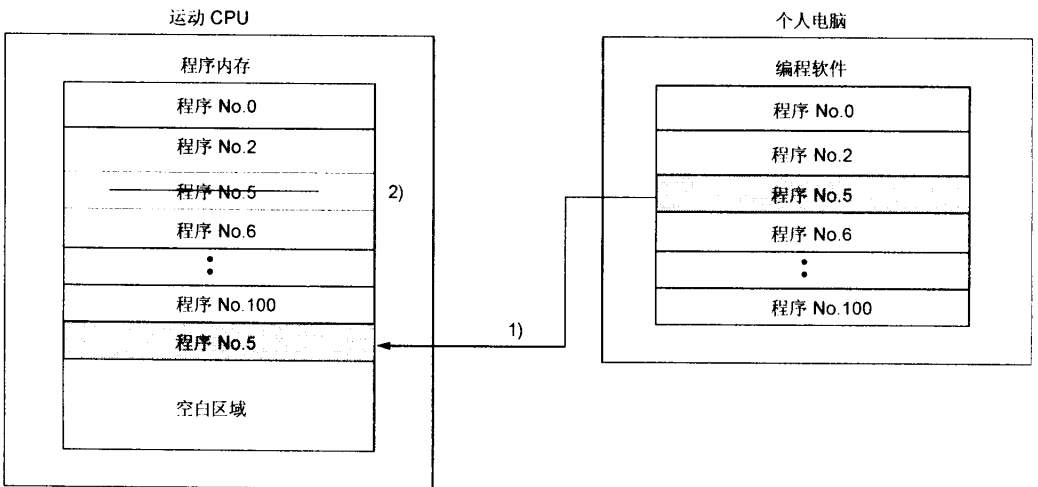
12.3.2 程序转换

在将程序软件中将程序转换成运动 CPU 的程序内存的操作大致如下。

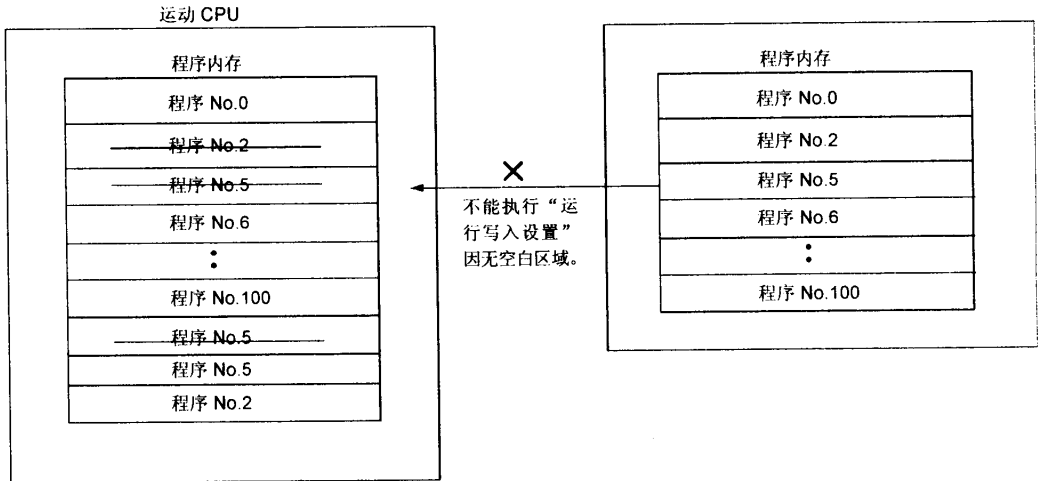
- (1) 用 [Communication] 菜单- [Transfer] 来写入程序
 - (a) 转换后，程序被存入运动 CPU 的程序内存。



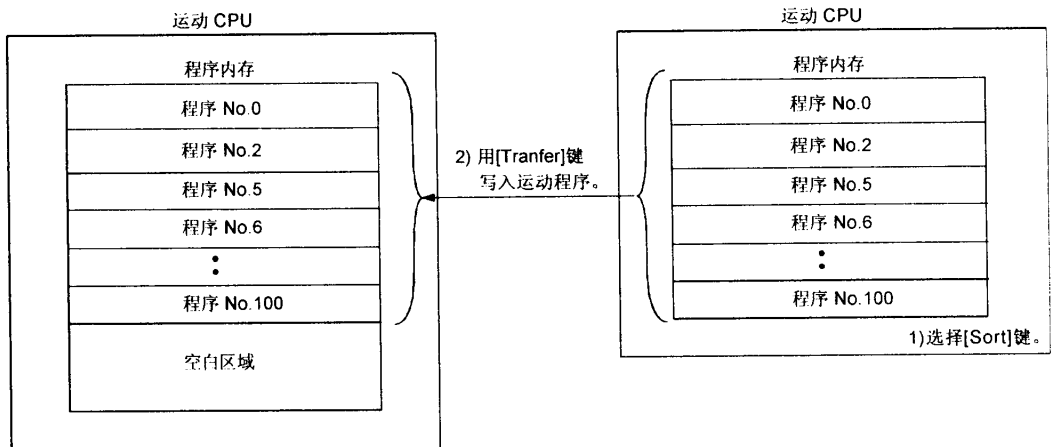
- (2) 用 [在线修改] 来写入程序
 - (a) 在线修改后，已存储的在线修改的程序在储存之前的程序之后的空白区域。(参照 1))
 - 之前写入的程序无效，新程序有效。(参照 2))



- (b) 若重复执行在线修改，会丢失程序内存中的空白空间，在线修改也可能不被执行。此时，程序软件会发出出错信息，并设置“在线修改 OFF”。



- (c) 在情况 b)中写入程序无效时。操作程序如下。
- 1)选择程序软件中的“程序 编辑画面 [Option] 菜单 - [Sort]”。
 - 2)运动 CPU 处于停止状态时，用[Communication] 菜单 - [Transfer]执行程序的写入。



13. 限位开关输出功能

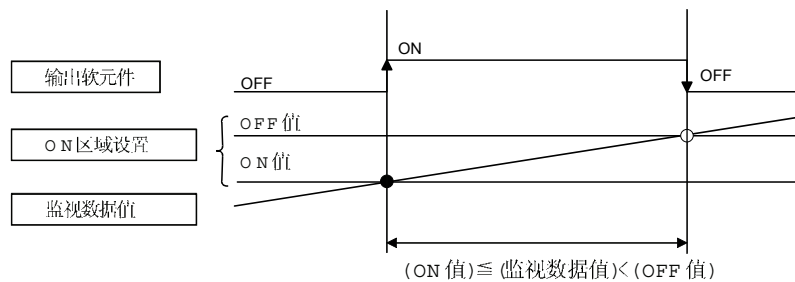
此功能应用于输出与设置在每个输出软件元件中的监视数据的数据范围相对应的ON/OFF信号。运动控制数据或可选的字可用做监视数据。(详情参见“13.2 限位输出设定数据。”)可以设置输出软件元件最大值为32点可不计其轴数量。

13.1 操作

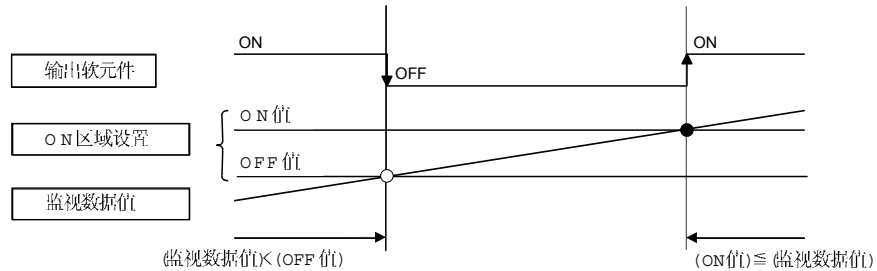
- ① 此功能中，当监视数据值由(ON值)和(OFF值)设置在ON输出区域中时，向输出软件元件进行ON输出。
 - ② (ON值)，(OFF值)与监视数据值是作为符号数据处理的。在ON输出区域内，向输出软件元件操作ON输出。ON输出区域由(ON值)和(OFF值)之间的大小关系控制，如下所示。

(ON值)和(OFF值)之间关系	ON输出区域
(ON值) < (OFF值)	(ON值) ≤ (监视数据值) < (OFF值)
(ON值) > (OFF值)	(ON值) ≤ (监视数据值) (监视数据值) < (OFF值)
(ON值) = (OFF值)	在整个区域内输出OFF

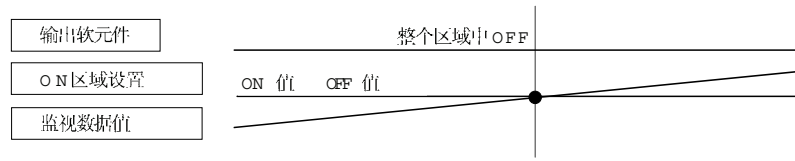
1) (ON值) < (OFF值)



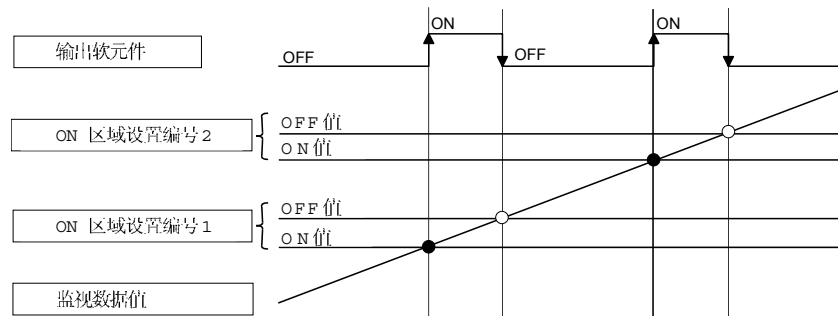
2) (ON值) > (OFF值)



3 (ON 值) = (OFF 值)



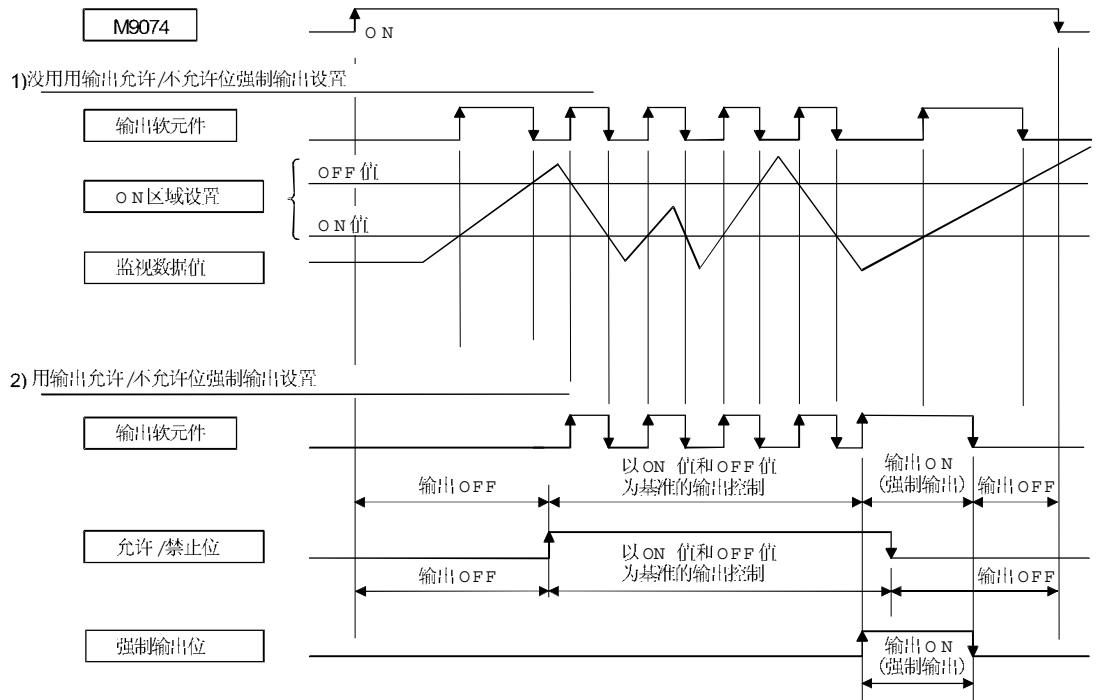
- b) 在 PCPU 就绪状态 (M9074: ON) 过程中, 由 PLC 就绪标记 (M2000) 从 OFF 到 ON 的转换, 以每个监视数据为基准控制限位开关输出。
当 PLC 就绪标记从 ON 转到 OFF, PCPU 就绪标记 (M9074) 转为 OFF。如 (ON 值) 和 (OFF 值) 指定字软元件, 则当 PLC 就绪标记 (M2000) 从 OFF 转到 ON 时, 字软元件内容输入到内部区域。
之后, 在每个操作周期输入字软元件内容, 并控制限位开关输出。
- (c) 多重输出 (多达 32 点) 可设置为一个监视数据。在每个设置中, 输出软元件可以相同。如在同一输出软元件中进行了多 ON 区域设置, 则输出此区域输出结果的逻辑加。



- 2) 输出允许 / 不允许位可逐点设置并执行限位开关输出允许 / 不允许。
当输出允许 / 不允许接通, 执行限位开关输出控制, 当断开时则输出断开。
如无设置, 则输出恒允许。
- 3) 强制输出位可和转换限位开关输出强制输出逐点 ON。
当强制输出位开启, 则输出开启。赋予优先权以控制“输出允许 / 不允许位”的设置。
如无设置, 则恒无强制输出。

13 限位开关输出功能

④ 当多监视数据，ON 区域，输出允许 / 不允许位和强制输出位设置到同一输出软元件，则输出结果的逻辑加。



13 限位开关输出功能

13.2 限位输出设置数据

以下所示为限制输出数据列表
 输出软元件可设置多达 32 点。
 (以编号 1 到编号 5 的项目设置为一个点。)

编号	项目		设置范围	取周期	刷新周期	备注
1	输出软元件		位软元件(X,Y,M,L,B)	—	操作周期	
2	监视数据		运动控制数据/字软元件 (D,W,#,绝对地址) (16-位 整型/32-位 整型/ 64-位浮点型)	操作周期	—	ON : 允许 OFF: 不允许 无 : 恒允许
3	ON 区域 设置	ON 值	字软元件 (D,W,#) /常数(K,H)			
		OFF 值	字软元件 (D,W,#) /常数(K,H)			
4	输出允许/不允许位		位软元件 (X,Y,M,L,B,F,特殊继电器 M)/ 无 (默认)			
5	强制输出位		位软元件 (X,Y,M,L,B,F,特殊继电器 M)/无 (默认)			

(1) 输出软元件

- (a) 设置位软元件向预置的监视数据输出 ON/OFF 信号。
 (b) 输出软元件，可应用以下软元件。

项目	软元件 编号 设置范围
输入继电器 ^(注-1)	X0 到 X1FFF
输出继电器 ^(注-2)	Y0 到 Y1FFF
内部继电器 ^(注-3)	M0 到 M8191
锁存继电器	L0 到 L8191
链接继电器	B0 到 B1FFF

(注-1): PX 为禁止写入，不可用作输出软元件。

对于 X，只有未装载输入卡的范围可以被使用。

(注-2): 实际输出软元件范围(PY)也包括在内。

(注-3): M2001 到 M2032 不可用于输出软元件。

在运动 CPU 中，因其影响定位操作，设置定位专用软元件时，小心使用。

2) 监视数据

- a) 此数据用于执行限位开关输出功能。此数据为输出 ON/OFF 信号的比较数据。此输出软元件的 ON/OFF 根据 ON 区域设置而控制。
- b) 运动控制数据或可选字软元件数据可作为监视数据。
 - 1) 运动控制数据

项目	单位	数据类型	轴号设置范围	
			Q173CPU(N)	Q172CPU(N)
当前反馈值	定位指令	32-位 整数型	1 到 32	1 到 8
当前实际值				
偏差计数值				
电机电流(指令输出电压: ACF)	0. 1%(0.01V)	16-位 整数型	1 到 32	1 到 8
电机速度	0. 1r/min	32-位 整数型		
每次旋转中凸轮轴当前值	PLS			
当前反馈值(暂时)				
差分当前值(暂时)				
差分编码器当前值				
编码器当前值				
			1 到 12	1 到 8

2) 字软元件数据

项目	软元件号设置范围
数据寄存器	D0 到 D8191
链结寄存器	W0 到 W1FFF
运动寄存器	#0 到#8191

3) 当可选软元件数据被设置时，设置以下数据类型用以比较

数据类型	软元件号设置范围
16-位整数型	设置软元件编号为偶数编号。
32-位整数型	
64-位浮点型	

(3) ON 区域设置

a) 使输出软元件转换 ON/OFF 到监视数据的数据范围。

b) 以下软元件可用作数据范围内的 ON 值和 OFF 值。
软元件 / 常数数据类型可以被设置为监视数据类型。

项目	软元件编号设置范围
数据寄存器	D0 到 D8191
链接寄存器	W0 到 W1FFF
运动寄存器	#0 到 #8191
常数	Hn/Kn

(4) 输出允许 / 不允许位

a) 操作过程中，当禁止限位开关输出时，设置输出允许 / 不允许位状态。

1) 运用以下控制

输出允许/不允许位		控制说明
无设置(恒允许)		以 ON 区域设置(ON 值, OFF 值)为基准, 转换限位开关 输出 ON/OFF。
设置	ON(允许)	
	OFF(不允许)	

b) 可用软元件

项目	软元件号设置范围
输入继电器 ^(注-1)	X0 到 X1FFF
输出继电器 ^(注-2)	Y0 到 Y1FFF
内部继电器	M0 到 M8191
锁存继电器	L0 到 L8191
链接继电器	B0 到 B1FFF
警报器	F0 到 F2047
特殊继电器	M9000 到 M9255

(注 -1): 实际输入范围 (PX) 也包括在内。

(注 -2): 实际输入范围 (PY) 也包括在内。

⑤ 强制输出位

① 操作过程中，如要强制控制限位开关输出，可设置“强制输出位”。

1) 运用以下控制。

输出允许/不允许位状态		控制描述
无设置		以“输出允许/不允许位”及 ON 区域设置(ON 值, OFF 值)为基准, 转换限位开关 输出 ON/OFF。
设置	OFF(不允许)	
	ON(允许)	

② 可用软元件

项目	软元件号 设置范围
输入继电器	X0 到 X1FFF
输出继电器	Y0 到 Y1FFF
内部继电器	M0 到 M8191
锁存继电器	L0 到 L8191
链接继电器	B0 到 B1FFF
警报器	F0 到 F2047
特殊继电器	M9000 到 M9255

14. ROM 操作功能

用户程序和参数预先存储在运动 CPU 模块的内部 FLASH ROM 内存中，机器运行基于此功能的内部 FLASH ROM 内存数据。

有关运动 CPU 和软件的应用版本参见 1.3.4 节。

14.1 有关 ROM 操作功能

以下为 ROM 操作功能的大致步骤。

- 1) 电源 ON 或在“RAM 操作模式”中执行多 CPU 系统复位。
- 2) 系统设置, 程序和参数由 SW6RN-GSV □ P 创建, 并且执行试运行和调整。
- 3) 电源 ON 或在“安装 • 写入 ROM 的模式”中执行多 CPU 系统复位。
- 4) 通过执行 SW6RN-GSV □ P 的 ROM 写入请求, 系统设置, 将程序和用于运动 CPU 模块的内部 SRAM 参数写入内部 FLASH ROM 。
- 5) 当在“ROM 操作模式”中启动运动 CPU 时, 在向内部 SRAM 写入到内部 FLASH ROM 的系统设置, 程序和参数后开始正常运行。

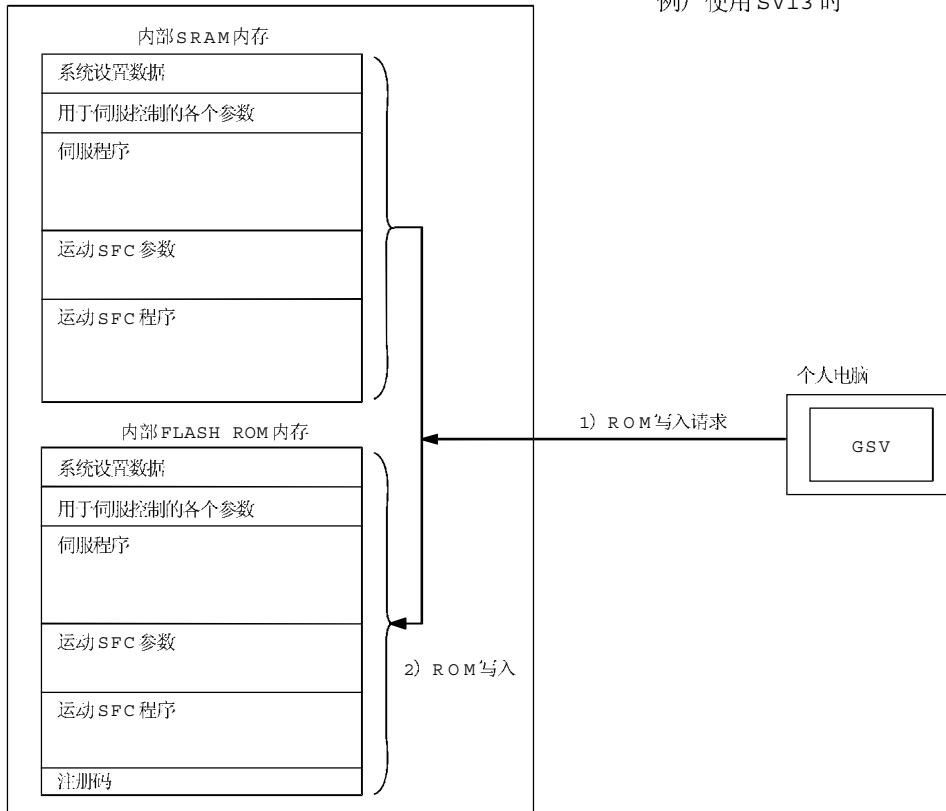
要点
(1) 运行模式可以用运动 CPU 模块的 DIP 开关切换。
(2) 运行模式可以用运动 CPU 模块的“模式 LED”和“BOOT LED”确认。

下页显示大致处理情况。

• 安装•写入ROM的模式

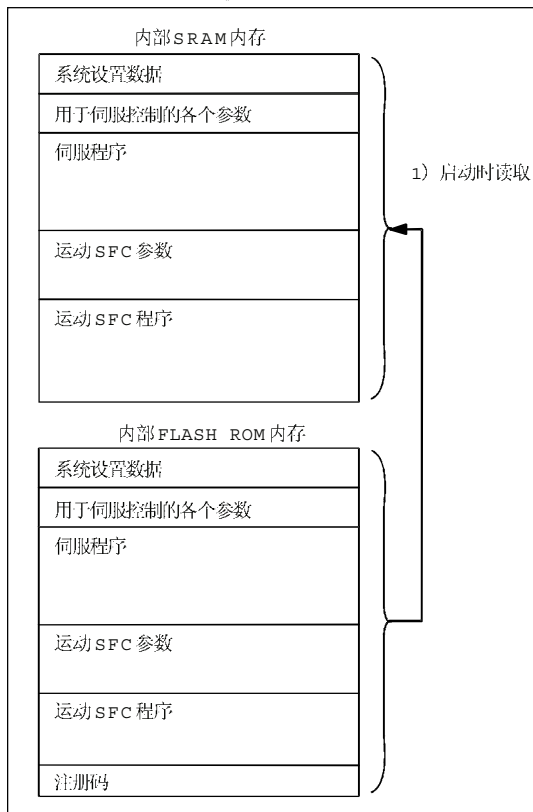
运动CPU模块

例) 使用SV13时



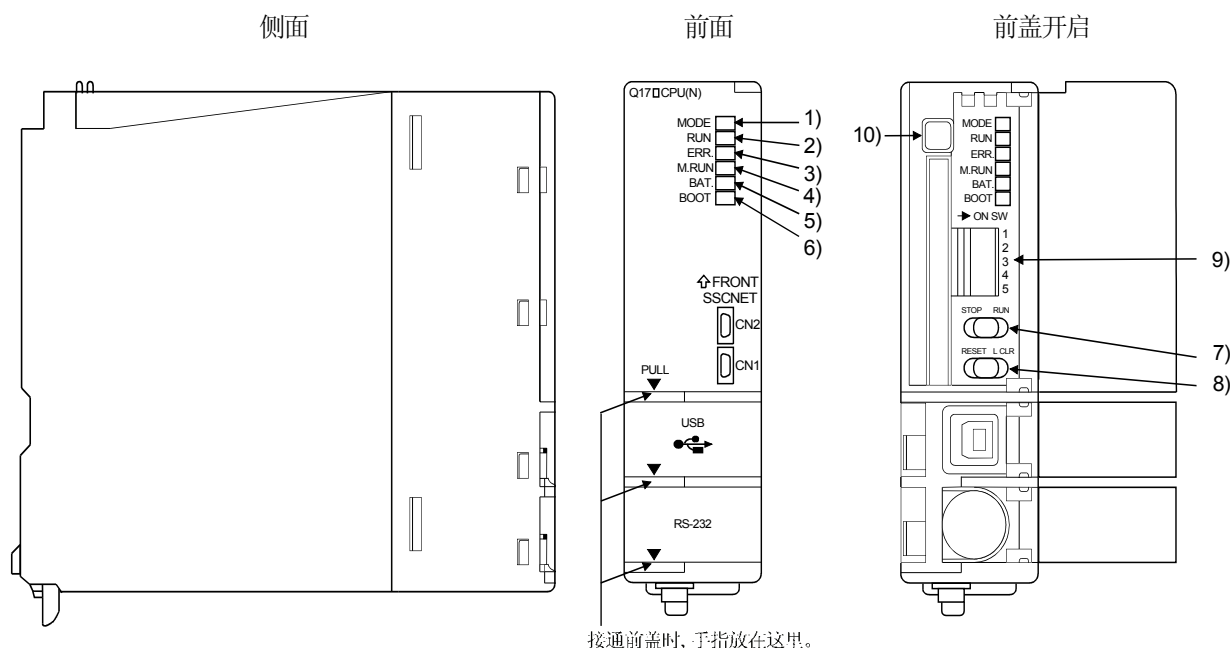
• 由ROM操作的模式

运动CPU模块



14.2 LED • 开关规格

(1) 部件名称



编号	名称	应用
1)	MODE LED(模式判断)	点亮(绿色) : 正常模式 点亮(橙色) : 安装 • ROM 写入模式
2)	RUN LED	点亮 : 运动 CPU 正常接通 不点亮 : 运动 CPU 故障。当启动时发生故障或 WDT 错误时, 断开 RUN LED 。
3)	ERR. LED	点亮 : 故障发生时, LED 点亮如下。 1) WDT 错误 2) 系统设置故障 3) 伺服故障 4) 运动 SFC 故障 5) 不停止运行的自诊断故障检测 (电池故障除外)。 闪烁 : 不停止运行的自诊断故障检测。 不点亮 : 正常
4)	M.RUN LED	点亮 : 执行运动控制时。 闪烁 : 开始清除锁存数据时。 不点亮 : 执行运动控制时, 或检测到不停止运行的自诊断故障时。
5)	BAT. LED	点亮 : 电池故障发生时。(使用外部电池时。)
6)	BOOT LED	点亮 : ROM 操作模式。 不点亮 : RAM 操作模式/安装模式 • ROM 写入模式。

(2) 开关应用

编号	名称	应用										
7)	RUN/STOP 开关 (瞬时开关)	<ul style="list-style-type: none"> • 移到 RUN/STOP. RUN : 执行运动 SFC 程序。 STOP : 停止运动 SFC 程序。 										
8)	RESET/L.CLR 开关 (注-1) (瞬时开关)	<p>RESET : 将开关设到"RESET" 位置一次, 复位硬件。在操作故障之后应用复位, 初始化操作。</p> <p>L.CLR : 清除锁存区域所有用参数设置的数据。 (LATCH CLEAR 同时也清除锁存区域之外的数据。)</p> <ul style="list-style-type: none"> • 锁存数据清除操作方法 1) 将"RUN/STOP" 开关设置为 "STOP"。 2) 数次移动"RESET/L.CLR" 开关到 "L.CLR" 直到"M.RUN LED" 闪烁。 ("M.RUN LED" 闪烁: 锁存数据清除完成。) 3) 再次将"RESET/L.CLR"开关移到 "L.CLR"。("M.RUN LED" 关闭。) 										
9)	DIP 开关 	DIP 开关 1	不可使用。正常 OFF。(出厂设置为 OFF 位置)									
		DIP 开关 2	<ul style="list-style-type: none"> •ROM 操作设置(出厂设置为 OFF 位置) <table border="0"> <tr> <td>SW2</td> <td>SW3</td> <td></td> </tr> <tr> <td>OFF</td> <td>OFF</td> <td>→ RAM 操作模式</td> </tr> <tr> <td>ON</td> <td>OFF</td> <td>→ 不要设置</td> </tr> </table>	SW2	SW3		OFF	OFF	→ RAM 操作模式	ON	OFF	→ 不要设置
		SW2	SW3									
		OFF	OFF	→ RAM 操作模式								
		ON	OFF	→ 不要设置								
DIP 开关 3	<table border="0"> <tr> <td>OFF</td> <td>ON</td> <td>→ 不要设置</td> </tr> <tr> <td>ON</td> <td>ON</td> <td>→ ROM 操作模式</td> </tr> </table>	OFF	ON	→ 不要设置	ON	ON	→ ROM 操作模式					
OFF	ON	→ 不要设置										
ON	ON	→ ROM 操作模式										
DIP 开关 4	不要使用。正常 OFF。(出厂设置为 OFF 位置)											
DIP 开关 5 (安装 • ROM 写入开关)	<p>ON : 安装模式 • ROM 写入模式</p> <p>OFF : 正常模式 (RAM 操作模式/ ROM 操作模式)</p> <p>从外围设备往运动 CPU 模块中安装操作系统时, 打开 DIP 开关 5。安装完成后移动开关并重启。</p>											
10)	内存卡 EJECT 按钮	用来从运动 CPU 中弹出内存卡。										

注-1) : 不可通过各 QCPU / 运动 CPU (2 号到 4 号) 来复位多 CPU 系统。

如果被复位, 其它 CPU 会在出现 "MULTI CPU DOWN" 的地方停止整个多 CPU 系统 (出错代码 : 7000)。

整个多 CPU 系统复位是指复位 QCPU 的 CPU No.1。

14.3 ROM 操作功能详情

a) 运行模式

当电源 ON 或多 CPU 系统复位时，多 CPU 的“运行模式”由运动 CPU 模块的 DIP 开关 2, 3, 5 的状态设置。

以下显示 DIP 开关设置，各运行模式以及各运行模式的运行概况。

a) DIP开关设置和各运行模式

DIP 开关设置			运行模式
SW2	SW3	SW5	
OFF	OFF	ON	安装模式 • ROM 写入模式
OFF	ON	ON	不可设置(注-1)
ON	OFF	ON	不可设置(注-1)
ON	ON	ON	安装模式 • ROM 写入模式
OFF	OFF	OFF	RAM 操作模式
OFF	ON	OFF	不可设置(注-2)
ON	OFF	OFF	不可设置(注-2)
ON	ON	OFF	ROM 操作模式

注-1：如果设置出错，在“安装模式 • ROM 写入模式”中运行。

注-2：如果设置出错，在“RAM 操作模式”中运行。

b) 各运行模式的运行概况

DIP 开关设置	运行模式
安装模式 • ROM 写入模式	<ul style="list-style-type: none"> • MODE LED 点亮为橙色。 • BOOT LED 断开。 • 可以安装操作系统。 • 用户程序以及 ROM 运行参数可以写入到运动 CPU 模块的内部 FLASH ROM。 • 在 RAM 运行模式中，检查运行之后在 ROM 运行时执行 ROM 写入。将存储在运动 CPU 模块的内部 FLASH ROM 中的用户程序和参数一起写入运动 CPU 模块的内部 FLASH。 • 不论运动 CPU 模块前面的 RUN/STOP 开关状态如何，机器变为 STOP 状态。 • 不可使用数字示波器功能。
RAM 操作模式	<ul style="list-style-type: none"> • MODE LED 点亮为绿色。 • BOOT LED 熄灭。 • 运行执行基于存储在运动 CPU 模块的内部 SRAM 中的用户程序和参数。
ROM 操作模式	<ul style="list-style-type: none"> • MODE LED 点亮为绿色。 • BOOT LED 熄灭。 • 当电源 ON 或多 CPU 系统复位时，将存储在内部 FLASH ROM 中的用户程序和参数读入内部 SRAM 后开始运行。在 ROM 运行模式期间，即使用户程序和参数被 GSV/□P 改变，在下一个电源 ON 或系统复位前，机器仍然恢复到内部 FLASH ROM 的内容。即使伺服放大器由自动调整设置运行，而且自动调整数据反映到运动 CPU 的伺服参数上，机器仍然在下一个电源 ON 或系统复位前恢复到内部 FLASH ROM 的内容。

要点

电源 ON 之后，即使 DIP 开关设置中途改变，“操作模式”不变。确定接通或复位多 CPU 系统电源来改变 DIP 开关设置。

② ROM 中的应用数据

以下为通过 ROM 写入模式，写入内部 FLASH ROM 的数据说明。下面这些数据之外的备份数据（绝对系统中伺服电机的当前位置，起始位以及锁存软元件等等。）不可写入内部 FLASH ROM 中。

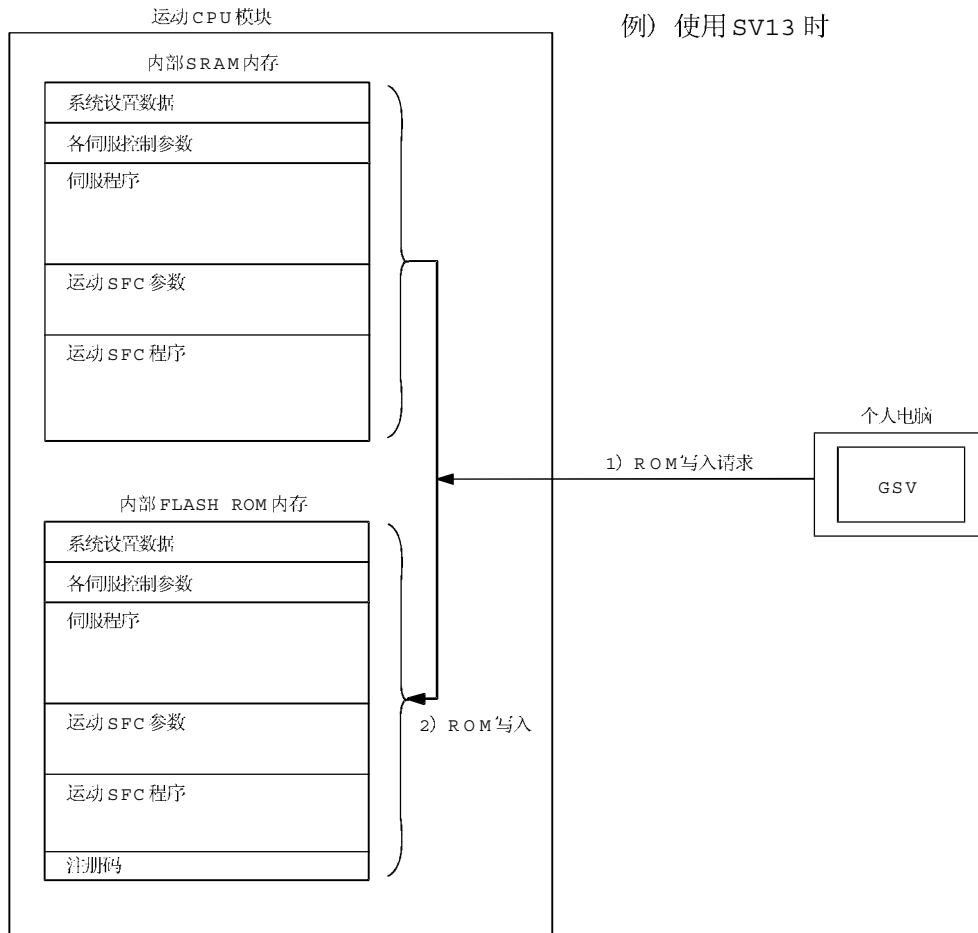
④ ROM 中应用数据说明

SV13	SV22
系统设置数据	系统设置数据
伺服控制的各个参数	伺服控制的各个参数
伺服程序	伺服程序
运动 SFC 参数	运动 SFC 参数
运动 SFC 程序	运动 SFC 程序
—	机械系统程序 ^(注-1)
—	凸轮数据 ^(注-1)

注 -1): 机械系统程序和凸轮数据为使用 SV22 时“ROM 中应用数据”。

b) ROM中应用数据运行

当通过GSV□P的“ROM写入”菜单请求向运动CPU模块写入时，在擦除运动CPU模块内部FLASH ROM的用户内存区域后将ROM中存储在内部SRAM的应用数据一起写入FLASH ROM。

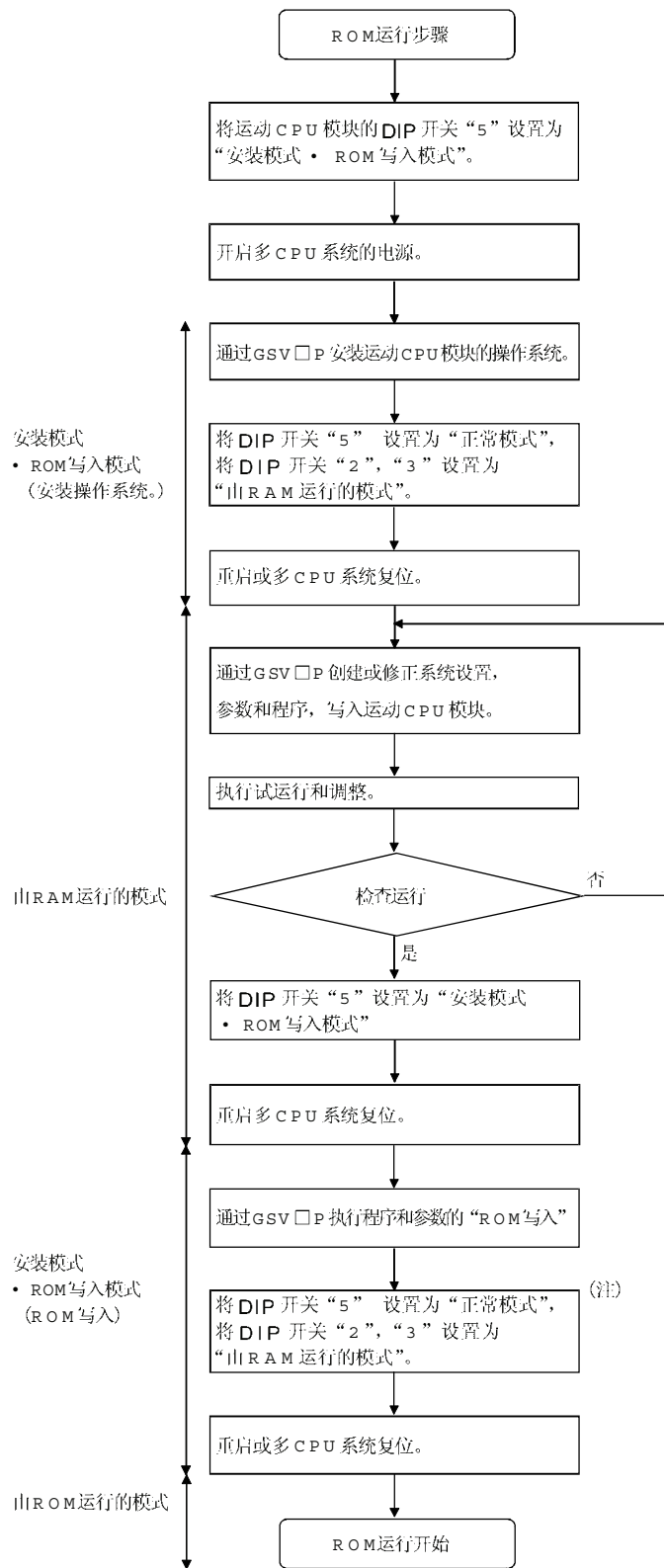


注-2)：“注册码”为操作系统写入数据，用来判断写入内部FLASH ROM的程序和参数是否正常。

要点	
	<p>(1) 通过 GSV□P 的“通讯”→“传送”菜单选择 RAM 时，运动 CPU 模块的内部 SRAM 内存在“安装模式·ROM 写入模式”和“ROM 操作模式”成为目标对象。</p>
	<p>(2) 运动 CPU 模块的内部 SRAM 内存在 GSV□P 的“备份·装载”运行中成为目标对象。在更换 CPU 模块时，在执行“备份·装载”之后执行 ROM 运行时执行“ROM 写入”，然后再设到“ROM 操作模式”。</p>
	<p>(3) 内部 FLASH ROM 的使用寿命为 100000 次写入。如果超过其使用寿命使用，将会出现“写入错误”。然后要更换模块进行 ROM 运行。</p>
	<p>(4) 在线写入在 ROM 操作模式时执行，经过在线修改后的程序在下一次停止时执行。然后在下一次电源 ON 或复位时恢复到写入内部 FLASH ROM 的程序内容。</p>

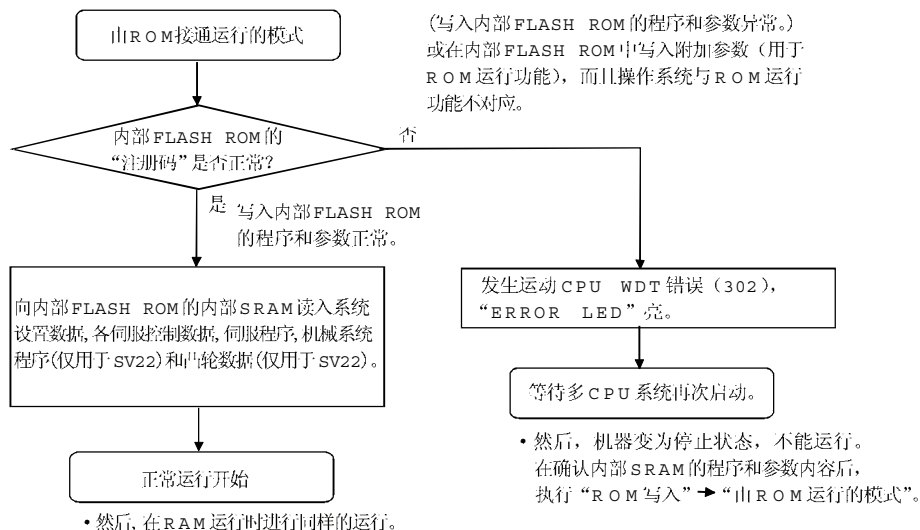
③ ROM 操作步骤

以下显示执行ROM 操作的流程图。



(注): 请勿处理安装操作系统的程序和参数写入。

- ④ 在“由ROM运行的模式”时运行
 以下为在“由ROM运行的模式”时的运行。

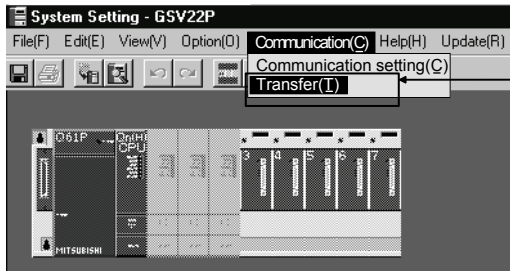


要点
如果写入内部 FLASH ROM 的数据的 ROM 运行与运动 CPU 模块同时执行，通过不适用于附加参数（ROM 操作功能）的操作系统，作用于附加参数（ROM 操作功能）和编程软件，运动 CPU WDT 错误（出错代码：302）将会发生，ROM 操作将无法执行。 在这种情况下，使用附加参数用操作系统(ROM 操作功能)。(参见第 1.3.4.节)

14.4 编程软件包“ROM 写入”的操作步骤

以下为 ROM 写入的操作步骤。

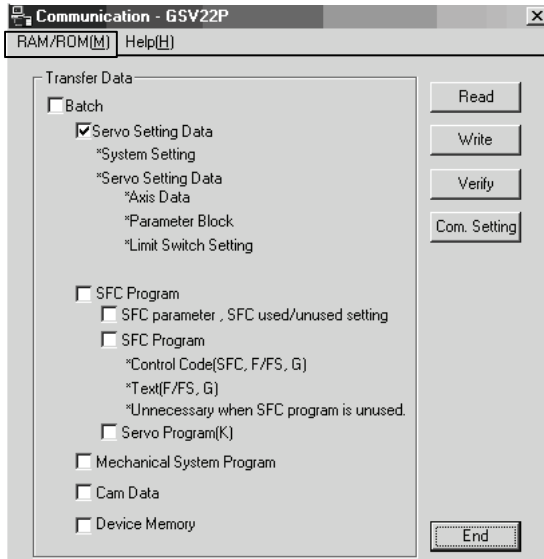
• 系统设置画面



运行步骤

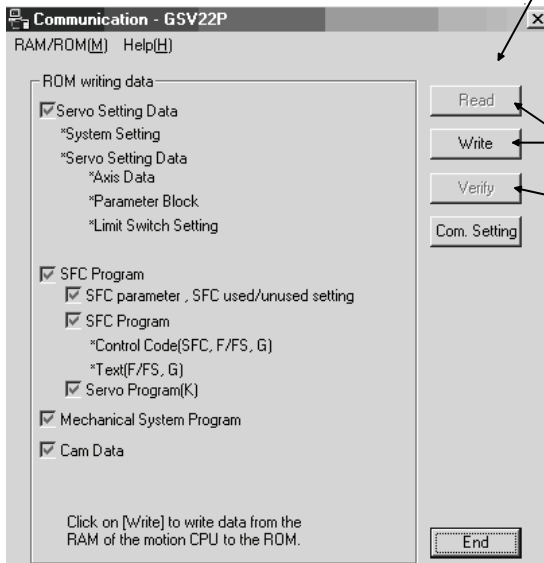
1) 点击系统设置菜单画面的“Communication”-“Transfer”之后显示“ROM/ RAM”通讯对话框。
(注):在 ROM 写入选择“传送”。

• “选择 RAM 时。”(选择“传送”时的缺省画面)



2) 点击“RAM/ROM”菜单画面的“ROM”。在运动 CPU 中,向 ROM 写入 RAM 的程序和数据。
(注):在“RAM”点击
左边画面为通讯对话框(选择 RAM 时)。对运动 CPU 的 RAM 执行读、写和验证。在 ROM 写入时点击 RAM 后,向运动 CPU 写入所有数据。

• “选择 ROM 时。”(ROM 写入窗口)



4) 左边画面为“RAM/ROM”通讯对话框(选择 ROM 时)。

4) 选择“选择 ROM 时”画面的写入。
(注):在 ROM 写入时所有的数据一起写入。

不选此项。

• 使用 SW6RN-GSV13P 时,不可将机械系统程序和齿轮数据写入 ROM。
上述指示不在画面中显示。

要点

在 ROM 写入时,确定事先将所有的数据写入运动 CPU 的 RAM 中。

15. 网络通讯

运动 CPU 及软件对应版本请参照“1.3.4”。

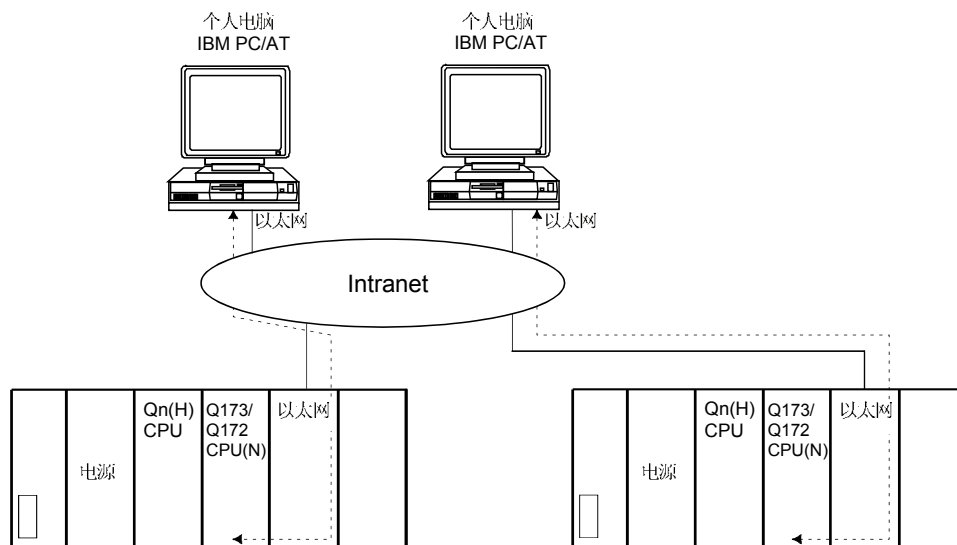
个人电脑与运动 CPU 间的通讯可通过运动 CPU (Q173CPU(N)/Q172CPU(N)) 中的 Q 系列网络模块 (MELSECNET/10(H), 以太网, CC-LINK, RS-232 等) 来实现。

以下各网络模块 MELSECNET/10(H), 以太网, CC-LINK 及串行通讯, 操作模式请参照下列使用手册。

- ① MELSECNET/10(H) 模块: QJ71LP21-25, QJ71LP21G, QJ71BR11, QJ72LP25-25, QJ72LP25G, QJ72BR15
 - Q 系列 CPU (Q 模式) 用户使用手册 (硬件设计, 维修及检查)
 - Q 系列对应 MELSECNET/H 网络系统参考使用手册 (PLC 到 PLC 网络)
 - Q 系列对应 MELSECNET/H 网络系统参考使用手册 (远程 I/O 网络)
- ② 以太网输入模块: QJ71E71, QJ71E71-B2, QJ71E71-100
 - Q 系列对应以太网接口模块使用手册 (硬件设计)
 - Q 系列对应以太网接口模块使用手册 (基础)
 - Q 系列对应以太网接口模块使用手册 (应用)
 - Q 系列对应以太网接口模块使用手册 (网络功能)
 - Q 系列对应 MELSEC 通讯协议参考使用手册
- ③ CC-LINK 模块: QJ61BT11
 - 控制 & 通讯链接主 / 本地模块类型 QJ61BT11 (硬件设计)
 - GX Configurator-CC 版本 1 操作手册
 - 控制 & 通讯链接主 / 从模块类型 QJ61BT11 (详情)
- ④ 串行通讯模块: QJ71C24, QJ71C24-R2
 - 串行通讯模块使用手册 (硬件设计)
 - Q 系列对应的串行通讯模块使用手册 (基本)
 - Q 系列对应的串行通讯模块使用手册 (应用)
 - Q 系列对应的 MELSEC 通讯协议参考手册

15.1 网络通讯规格

- ① 运动 CPU 的网络通讯可通过 SW6RN-GSV□P 实现。
- ② 运动 CPU 的网络通讯存取范围与 Qn(H) CPU 的相同。其存取范围请参照“15.2”。
- ③ 给连接 MELSECNET/10 (H) 及以太网外围设备的 CPU 和网络模块的控制 CPU 设置路径参数，可连接 8 个网络点进行通讯。
- ④ 因运动 CPU 不能成为网络模块的控制 CPU，故没有网络模块及网络参数的设定项目。然而，在其它网络与连接运动 CPU 的外围设备相连时，需设置路径参数。
- ⑤ 由远程控制监视器或程序通过以太网模块的 **intranet** 来编辑运动 CPU 程序。



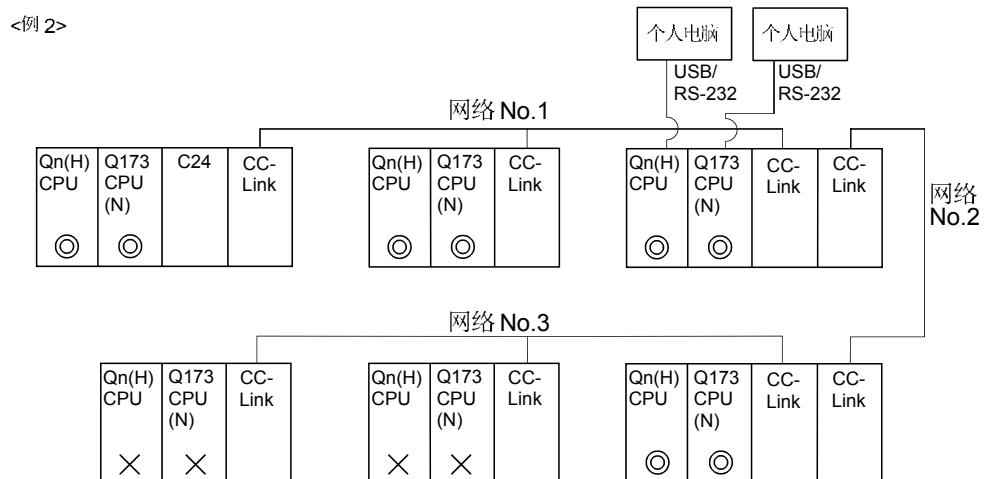
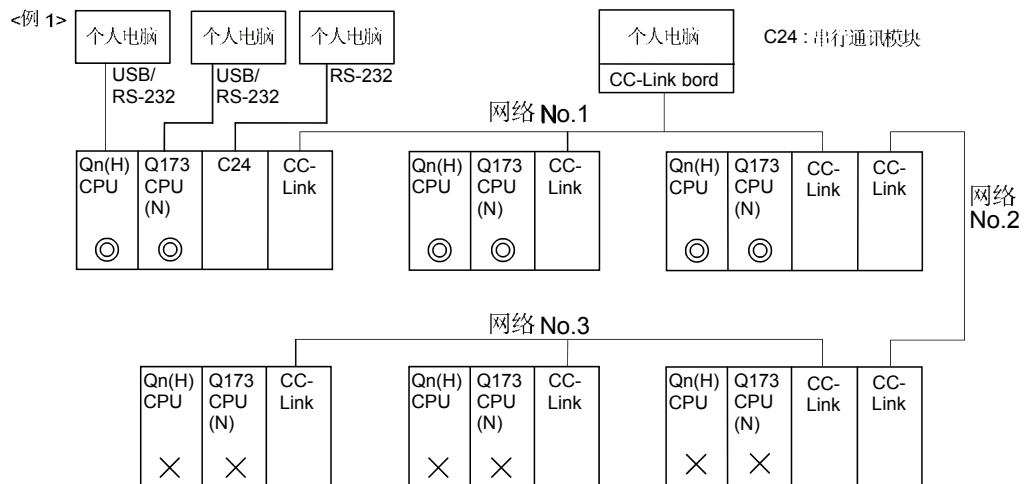
15.2 网络通讯存取范围

15.2.1 MELSECNET/10 (H) 或以太网的网络配置

- ① 通过网络由个人电脑中与USB/RS-232的 CPU 或串行通讯模块相连的编程软件 (GX Developer, SW6RN-GSV□P 等) 来存取 CPU 。
- ② 将配以以太网的个人电脑连接到MELSECNET/10 (H) 或以太网板到以太网到 MELSECNET/10 (H) 或以太网, 可通过网络从个人电脑的编程软件中存取其它 CPU。
- ③ 通过给网络模块和与个人电脑相连的 CPU 设置路径参数, 上述 (1)and(2) 的存取范围可访问 8 个网络点。

15.2.2 通过 CC-LINK 的网络配置

- ① 通过CC-LINK由个人电脑中与USB/RS-232的 CPU 或串行通讯模块相连的编程软件 (GX Developer, SW6RN-GSV □P 等) 来存取 CPU 。
- ② 将配以 CC-LINK 板的个人电脑连接到 CC-LINK, 可通过 CC-LINK 从个人电脑的编程软件中存取其它CPU。
- ③ 上述 ① 的存取范围仅在 CPU 连接制 CC-LINK时适用。通过指定 CC-LINK 模块的 I/O No., 可选择 CC-LINK 网络 。
- ④ 上述 ② 的存取范围仅在 CPU 连接制 CC-LINK 时适用。



- ◎ : 可通讯
- : 可通讯 (需设置路径参数。)
- × : 不可通讯

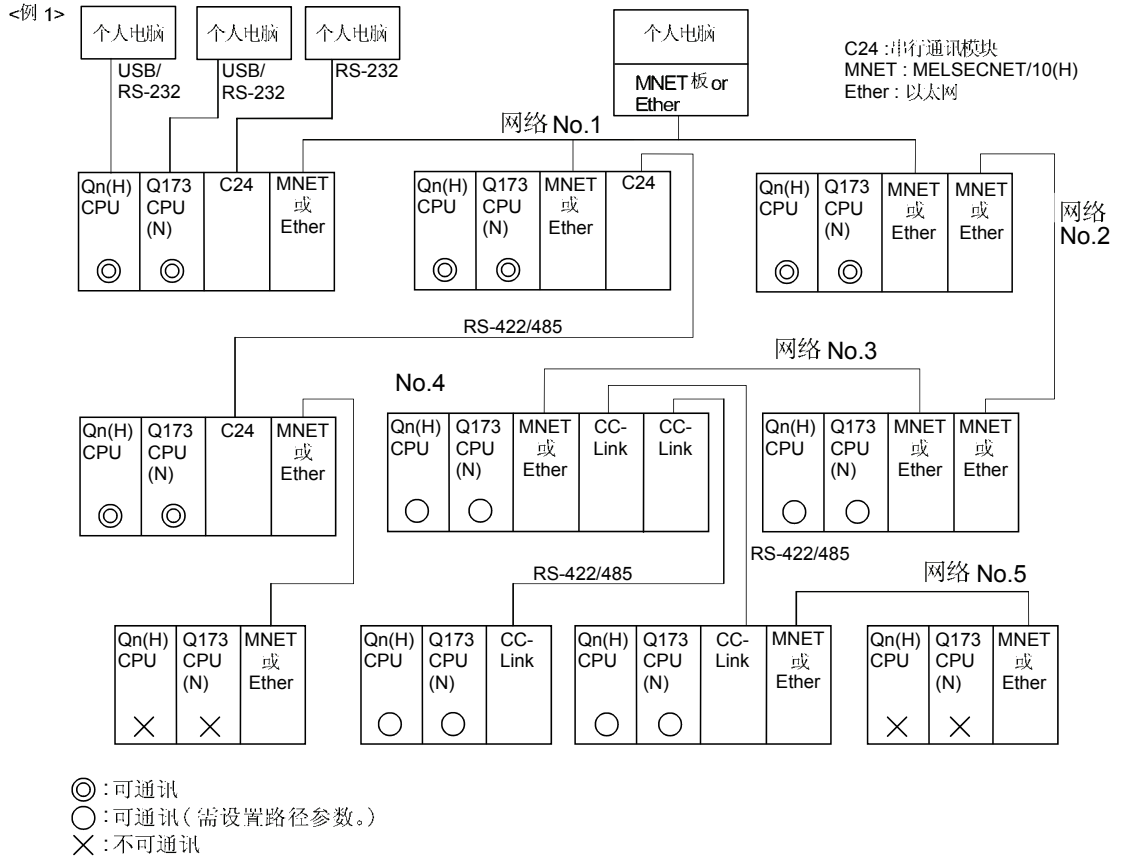
15.2.4 MELSECNET/10 (H), 以太网, CC-LINK, RS-422/485 混合时的网络配置

- (1) MELSECNET/10 (H) 或以太网被定义为“网络”而 CC-LINK 或 RS-422/485 被定义为“链接”，其不同组合是否能从编程软件 (GX Developer, SW6RN-GSV □ P 等) 中存取，如下图所示。

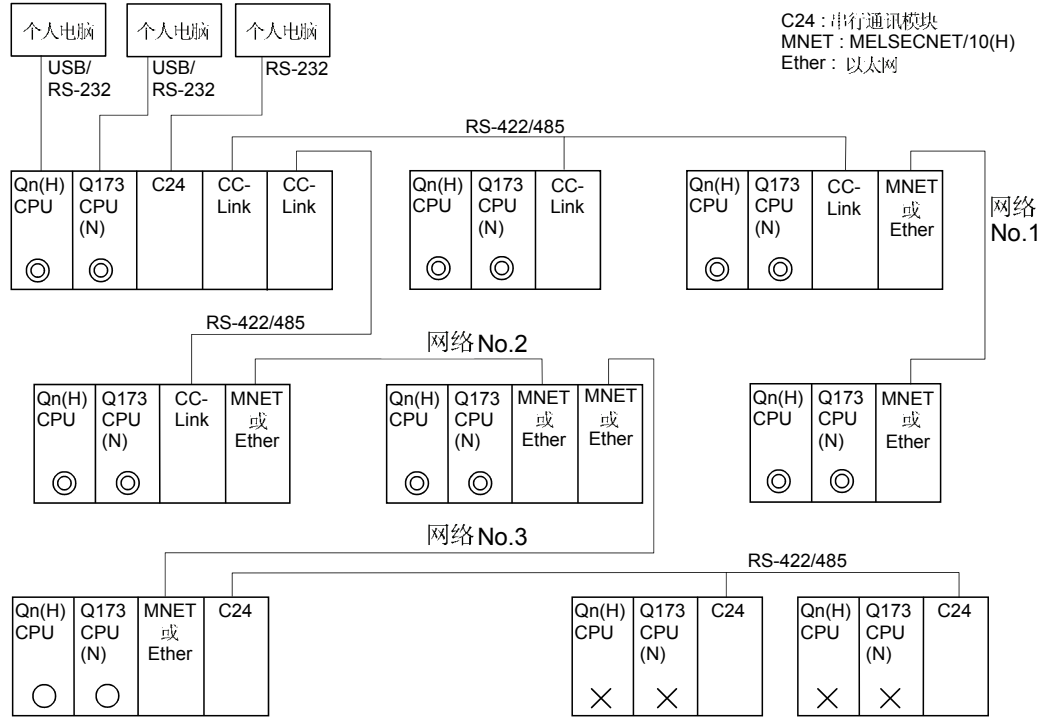
网络通讯	可用/不可用
编程软件 → CPU (C24) → 网络 → 链接 → CPU	○
编程软件 → CPU (C24) → 链接 → 网络 → CPU	○
编程软件 → 网络 → 链接 → CPU	○
编程软件 → 链接 → 网络 → CPU	○
编程软件 → CPU (C24) → 网络 → 链接 → 网络 → CPU	×
编程软件 → CPU (C24) → 链接 → 网络 → 链接 → CPU	×
编程软件 → 网络 → 链接 → 网络 → CPU	×
编程软件 → 链接 → 网络 → 链接 → CPU	×

○：可用 ×：不可用

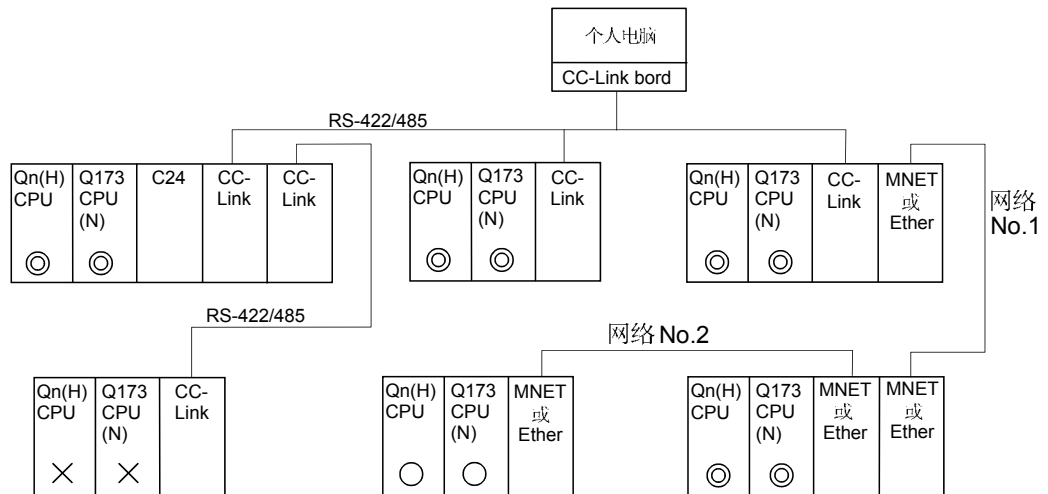
- ② 在“网络”中设置路径参数可存取 8 个网络点。
- ③ 因不能进入路径，仅能在连接网络时存取。可通过指定模块的 I/O 号来选择连接的网络。



<例 2>



<例 3>



- ◎ : 可通讯
- : 可通讯 (需设置路径参数。)
- × : 不可通讯

16. 主周期监视功能

有关运动 CPU 和软件的应用版本参见 “1.3.4” 节。

- ① 用于 CPU 处理的主环路信息 (除了运动控制在空闲时间里执行的处理周期) 存储在特殊寄存器里。
- ② 在主周期内执行共享 CPU 内存的自动刷新以及运动 SFC 程序的通常任务。将其作为程序处理时间等的基准。
- ③ 使用以下几种方法可以缩短主周期。
 - Ⓐ 延长运行周期的设置。
 - Ⓑ 在运动 SFC 程序中减少要执行的事件任务程序数量。
 - Ⓒ 同时在运动 SFC 程序中减少要执行的通常任务程序数量。
 - Ⓓ 减少共享 CPU 内存的自动刷新点数。
- ④ 当主周期变长 (超过大约 1.6[s] 时), 在运动 CPU 中可能会出现 WDT 故障。
- ⑤ 下面是有关主周期监视寄存器的详细情况。

编号	名称	释义	详情
D9017	扫描时间	当前扫描时间 (1ms 单位)	<ul style="list-style-type: none"> • 当前主周期以 1[ms] 为单位存储。 • 设置范围 (0 到 65535[ms])
D9019	最大扫描时间	最大扫描时间 (1ms 单位)	<ul style="list-style-type: none"> • 主周期的最大值以 1[ms] 为单位存储。 • 设置范围 (0 到 65535[ms])

17. 从伺服放大器上读取伺服参数的功能

有关运动 CPU 和软件的应用版本参见 “1.3.4” 节。

- ① 本功能将伺服放大器(通过伺服放大器设置软件直接修改)的伺服参数反映到运动 CPU 的伺服参数上。
- ② 要反映的参数是伺服放大器 1 号到 39 号伺服参数。
- ③ 在每个轴中从伺服放大器读取伺服参数。
- ④ 执行从伺服放大器上读取伺服参数的功能时用户不必对运动 CPU 的伺服参数再次进行设置。
- ⑤ 执行从伺服放大器上读取伺服参数的功能后, 如果需要将改变的伺服参数反映到编程软件包 (SW6RN-GSV □ P) 上, 请从运动 CPU 读取伺服参数并且保存数据。

17.1 关于伺服参数读取请求软元件

- ① 设置伺服放大器轴号, 在伺服参数读取请求轴号 (D9104) 中读取参数, 并且接通伺服参数读取请求标志 (M9104), 从伺服放大器读取伺服参数。
- ② 当伺服参数读取标志 (M9105) 开启时, 伺服参数读取请求标志无效。使用本条件作为互锁。
- ③ 从伺服参数读取请求标志断开到接通期间, 从伺服放大器读取伺服参数。
- ④ 伺服参数读取请求标志没有自动断开。在用户侧执行软元件 OFF 处理。
- ⑤ 从伺服放大器执行伺服参数读取功能后, 当向其它轴执行伺服参数读取请求时, 断开伺服参数读取请求标志 (M9104) 然后接通它, 在伺服参数读取请求轴号 (D9104) 中设置相应轴, 并且断开伺服参数读取请求标志 (M9104) 然后接通它。
- ⑥ 从伺服放大器执行伺服参数读取功能后, 当再次向相同轴执行伺服参数读取请求时, 接通伺服参数读取请求标志 (M9104) 然后断开它, 并且断开伺服参数请求标志 (M9104) 然后接通它。

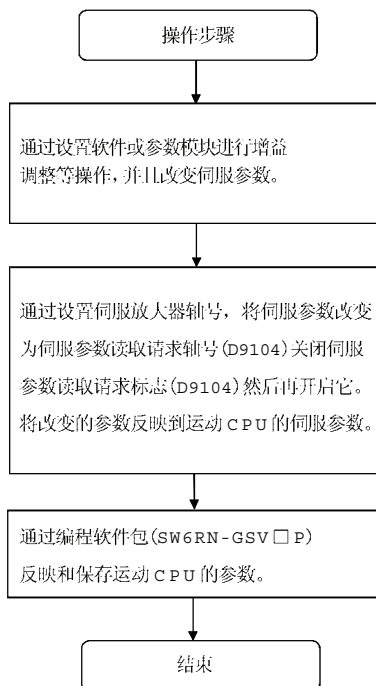
17 从伺服放大器上读取伺服参数的功能

- ⑦ 当断开伺服参数读取请求标志 (M9104) 然后接通它时, 如果没有使用伺服参数读取请求轴或电源断开, 不执行来自伺服放大器的伺服参数读取操作。
- ⑧ 当伺服参数读取请求轴号 (D9104) 超出设置范围时, 即使断开伺服参数读取请求标志 (M9104) 然后再接通它, 机器仍会变得“无操作”。
- ⑨ 以下为伺服参数读取请求软元件列表。

编号	名称	释义	详细情况
M9104	伺服参数读取请求标志	OFF 到 ON : 伺服参数读取	• 从断开到接通的时段内, 设置为 D9104 的伺服参数读取请求轴的伺服参数从伺服放大器反映到运动 CPU 上。
M9105	伺服参数读取标志	ON : 伺服参数读取 OFF : 不包括伺服参数读取	• 当从伺服放大器读取伺服参数到运动 CPU 时接通。读取后自动断开。
D9104	伺服参数读取请求轴号	伺服参数读取轴号	• 设置伺服放大器轴号, 读取伺服参数。 • 设置范围 Q173CPU(N) : 1 到 32 (轴 1 到 32) Q172CPU(N) : 1 到 8 (轴 1 到 8)

17.2 从伺服放大器上读取伺服参数功能的操作步骤

以下图示反映在编程软件包 (SW6RN-GSV□P) 上的伺服参数读取步骤 (通过从伺服放大器读取伺服参数的功能)。



18. 出错代码列表

运行运动CPU 时出现错误,其出错信息会储存到出错历史寄存器 (#8000 到 #8063) 和特殊继电器 M 和特殊寄存器 D 中。

18.1 出错代码读取程序

运动 SFC 程序时出现错误,可 SW6RN-GSV □ P 读取出错代码和出错信息。其读取程序如下。

- 1) 启动 SW6RN-GSV □ P。
- 2) 将 Q173CPU(N) / Q172CPU(N) 连接到外围设备。
- 3) 选择 [New project] 创建工程 - SW6RN-GSV □ P 的 [Read from Motion CPU] 菜单, 并从运动 CPU 中读取此工程。
- 4) 选择 [Monitor] - [Error list] - [Motion SFC error history] 和 [Error list] 菜单。
- 5) 显示出错代码和出错信息。

SW6RN-GSV □ P 操作方式请参照 SW6RN-GSV □ P 应用指南。
运动 CPU 出错历史出现时间采用内部运动 CPU 的时钟功能。
使用用户程序设置时钟数据和时钟数据的读取请求 (M9028)。
至于错误自诊断代码,由 GX Developer 的 PC 检测确认。
GX Developer 操作程序请参照其使用手册。

18.2 运动 SFC 出错代码列表 (由 GSV □P 确认)

运动寄存器的“出错历史软元件(#8000 到 #8063)”可存储运动 SFC 控制期间发生的八个错误。

运动 SFC 程序的“出错代码”如下。

请参照“Q173CPU(N)/Q172CPU(N) 运动控制器 (SV13/SV22) 编程手册 (实模式)” / “Q173CPU(N)/Q172CPU(N) 运动控制器 (SV22) 编程手册 (虚模式)”中轻度错误, 严重错误, 伺服错误及伺服程序错误。

表 18.1 运动 SFC 专门软元件 (#8000 到 #8063)

软元件号	信号名称		信号方向		刷新周期	导入周期		
			状态	命令				
#8000 到	过去的第七个出错信息 (最初出错信息)	运动 SFC 出错历史 (8 个错误) (64 点)		○	—	发生错误时	—	
#8008 到	过去的第六个出错信息	编 号	信号名					
#8016 到	过去的第五个出错信息	+0	出错运动 SFC 程序号					
#8024 到	过去的第四个出错信息	1	错误类型					
#8032 到	过去的第三个出错信息	2	错误程序号					
#8040 到	过去的第二个出错信息	3	错误块号 / 运动 SFC 列表 / 线号 / 轴号					
#8048 到	过去的第一个出错信息	4	出错代码					
#8056 到	最新出错信息	5	错误发生					年/月
#8063		6	时间					日/时
		7						分/秒

表 18.2 运动 SFC 程序启动错误 (16000 到 16099)

出错代码	故障原因		故障处理	对策
	名称	说明		
16000	PLC 就绪 OFF (SFCS)	• 由 S(P).SFCS 指令启动时, PLC 就绪标志 (M2000) 或 PCPU 就绪标志 (M9074) 为 OFF	未启动运动 SFC 程序	PLC 就绪标志 (M2000) 和 PCPU 就绪标志 (M9074) 为 ON, 启动互锁
16001	运动 SFC 程序号错误 (SFCS)	• 由 S(P).SFCS 指令启动时, 运动 SFC 程序号指定范围为 0 到 255		检查运动 SFC 程序号并更正 PLC 程序
16002	无运动 SFC 程序 (SFCS)	• 由 S(P).SFCS 指令启动时, 不存在指定的运动 SFC 程序		检查运动 SFC 程序号并更正 PLC 程序, 或创建新的运动 SFC 程序
16003	双启动错误	• 由 S(P).SFCS 指令启动时, 同一运动 SFC 程序亦启动		由用户控制双启, 在 PLC 程序中由用户启动信号作为互锁
16004	PLC 就绪 OFF (GINT)	• PLC 就绪标志 (M2000) 或 PCPU 就绪标志 (M9074) 为 OFF 时, 执行 S(P).GINT 指令	无法处理由“PLC 中断”执行的运动 SFC 程序的激活步	由 PLC 就绪标志 (M2000) 和 PCPU 就绪标志 (M9074) 为 ON, 执行 S (P) .GINT 互锁
16005	无运动 SFC 程序	• 自动启动或由 GSUB 启动运动 SFC 程序时, 不存在指定的运动 SFC 程序	未启动指定的运动 SFC 程序。由 GSUB 启动时, 源运动 SFC 停止运行	检查运动 SFC 程序号并更正 PLC 程序, 或创建新的运动 SFC 程序
16006	双启动错误	• 自动启动或由 GSUB 启动运动 SFC 程序时, 已启动同一运动 SFC 程序		由用户控制双启, 在转移条件中由用户的启动信号作为互锁
16007	在线修改	• 启动了由在线修改重写运动 SFC 图表的运动 SFC 程序	未启动指定的运动 SFC 程序	在线修改完成后启动

表 18.3 运动 SFC 翻译检测错误 (16100 到 16199)

出错代码	故障原因		故障处理	对策	
	名称	说明			
16100	运动 SFC 程序错误 (语法错误)	• 存在代码但有语法错误 • 虽然不在分支耦合内, 但存在选择分支耦合或平行耦合的标识转移代码	停止执行应用运动 SFC 程序号, 同时执行子程序中的调用源程序	运动 SFC 程序代码中断 断开 PLC 就绪标志 (M2000) 并重新写运动 SFC 程序。若电池使用过期, 请更换外部电池	
16101		• 由 SFT 或 WAIT 之外的转移引导选择性分支目标			
16102		• 运动控制步无法跟随 WAITON/WAITOFF(但可跟指针 (Pn) 或转移 (Pn))			
16103		• 不需平行耦合, END 步可跟随平行分支			
16104		运动 SFC 代码错误			• 使用了不可用的代码, 内部代码被中断
16105		转移代码错误 1			• 转移目标信息内部代码(列表代码) 错误
16106		转移代码错误 2			• 转移目标信息内部代码(标识信息) 错误
16107		转移代码错误 3			• 转移目标信息内部代码(标识号) 错误
16108		转移代码错误 4			• 转移目标信息内部代码(标识地址) 错误
16109		转移目标错误			• 在转移目标中不存在指定点
16110	GSUB 设置错误 1	• 由 GSUB 调用/启动程序本身			

表 18.3 运动 SFC 翻译检测错误 (16100 到 16199) (续)

出错代码	故障原因		故障处理	对策
	名称	说明		
16111	GSUB 设置错误 2	•由 GSUB 调用/启动主程序	停止执行应用运动 SFC 程序号同时执行子程序中的调用源程序	GSUB 无法调用其程序或主程序。 更正 运动 SFC 程序
16112	平行分支嵌套超出	•平行分支路径的平行分支嵌套超过四级		平行分支嵌套 最多有四级 创建分支目标处理子程序并更正程序
16113	执行任务错误	•尝试用事件任务或 NMI 任务执行运动控制步 Kn		运动 SFC 程序中的运动控制步不能在事件任务或 NMI 任务执行
16120	同步激活步计算超出	•执行的同步激活步数超过 256		同步活性步数最多可达 256, 重新检测 运动 SFC 程序

表 18.4 运动 SFC 程序运行错误 (16200 到 16299)

出错代码	故障原因		故障处理	对策
	名称	说明		
16200	无指定程序 (Kn)	•不存在运动控制步指定的伺服程序(Kn)	停止执行应用运动 SFC 程序号, 同时执行子程序中的调用源程序	创建指定的伺服程序
16201	无指定程序(Fn/FSn)	•不存在操作控制步指定的伺服程序(Fn/FSn)		创建指定的操作控制程序
16202	无指定程序(Gn)	•不存在指定的转换程序 (Gn)		创建指定的转换程序
16203	无指定程序(运动 SFC)	•不存在清除步指定的 运动 SFC 程序		更正指定运动 SFC 程序名称或创建指定运动 SFC 程序
16204	未设置操作表达式/条件表达式	•转换指定程序 (Gn) 未设置条件表达式		确保在转换程序的最后块中设置条件表达式
16205	Fn/FSn 程序代码错误	•操作程序(Fn/FSn)中有内部代码错误		运动 SFC 程序代码中断 断开 PLC 就绪标志 (M2000) 并重新写运动 SFC 程序。若电池使用过期, 请更换外部电池
16206	Gn 程序代码错误	•转换程序(Gn)中有内部代码错误		
16207	指定了无效的软元件	•在程序中设置了无效软元件 (T, C)	更正未设置有效软元件的程序	

表 18.5 操作控制 /转移执行错误 (16300 到 16599)

出错代码	故障原因		故障处理	对策
	名称	说明		
16301	事件任务使能(EI) 执行错误	• 事件任务使能在除正常任务以外的其他任务执行	停止正在执行的块处理, 并执行下一块	事件任务使能只能由正常任务执行, 修改程序
16302	事件任务屏蔽(DI) 执行错误	• 事件任务屏蔽在除正常任务外的其他任务执行		事件任务屏蔽只能由正常任务执行, 修改程序
16303	块转移(BMOV) 执行错误	<ul style="list-style-type: none"> • 用(D) 或(S)指定的凸轮数据没有存储在运动控制器中 • 用(D) 或(S)指定的凸轮分辨率与(n)指定的转移字数不同 • 用(D) 或(S)指定的 PCPU 内存地址超出 SRAM 范围 • (S) 到(S)+(n-1) 超过软元件范围 • (D) 到(D)+(n-1) 超过软元件范围 • (n) 为 0 或负数 		<ul style="list-style-type: none"> • 修改程序使凸轮数据为已存的凸轮数据 • 修改程序使(n) 与凸轮分辨率相符 • 修改程序并用一偶数指定 PCPU 内存地址 • 改正(n)使块转移范围在软元件范围内 • 将 (n) 改成正数
16304	等待时间(TIME) 执行错误	<ul style="list-style-type: none"> • 间接指定 (S) 的软元件号无效 • (S) 数据超出 0 到 2147483647 这一范围 		<ul style="list-style-type: none"> • 修改程序使间接指定(S) 的软元件号正确 • 更改程序使 (S) 数据在 0 到 2147483647 范围内
16308	速度更改请求(CHGV) 执行错误	• 指定的轴号超出范围		修改程序使指定轴号在范围内
16309	转矩限制更改请求(CHGT) 执行错误			
16316	赋值(=) 执行错误	<ul style="list-style-type: none"> • (S) 数据超出 (D)数据类型范围 • 由 (D)间接指定的软元件号无效 		<ul style="list-style-type: none"> • 修改程序使 (S) 数据在(D)数据类型范围内 • 修改程序使间接指定 (D) 的软元件号正确
16320	操作 (/)执行错误	• 除数为 0		修改程序使除数不为 0
16321	余数(%)执行错误			
16322	软元件设置(SET) 执行错误	<ul style="list-style-type: none"> • 由 (D) 间接指定的软元件号无效 • (D) 为不可写入的软元件 		<ul style="list-style-type: none"> • 修改程序使间接指定 (D) 的软元件号正确 • 修改程序并在(D)中设置使能软元件
16333	软元件复位(RST) 执行错误			
16334	软元件设置 (SET=) 执行错误			
16335	软元件复位(RST=) 执行错误			
16336	软元件输出 (DOUT) 执行错误			
16337	软元件输入 (DIN) 执行错误	• 由(D)间接指定的软元件号无效		修改程序使间接指定 (D) 的软元件号正确
16338	位软元件输出 (OUT=) 执行错误			
16380	有符号 16 位整数换算(SHORT) 执行错误	• (S) 数据不在有符号 16 位整数范围内	修改程序使 (S) 数据在有符号 16 位整数范围内	

表 18.5 操作控制 /转移执行错误 (16300 到 16599) (续)

出错代码	故障原因		故障处理	对策
	名称	说明		
16381	无符号 16 位整数换算(USHORT) 执行错误	• (S) 数据超出无符号 16 位整数值范围	停止正在执行的块处理, 并执行下一块	修改程序使 (S) 数据在无符号 16 位整数范围内
16382	有符号 32 位整数换算 (LONG) 执行错误	• (S) 数据超出有符号 32 位整数值范围		修改程序使 (S) 数据在有符号 32 位整数范围内
16383	无符号 32 位整数换算 (ULONG) 执行错误	• (S) 数据超出无符号 32 位整数值范围		修改程序使 (S) 数据在无符号 32 位整数范围内
16398	正切 (TAN) 执行错误	• (S) 等于 $90+(180*n)$ 其中 n 为整数		修改程序使(S)不等于 $90+(180*n)$ 其中 n 为整数
16399	反正弦 (ASIN) 执行错误	• (S) 超出-1.0 到 1.0 这一范围		修改程序使(S)在-1.0 到 1.0 这一范围内
16400	反余弦 (ACOS) 执行错误			
16402	平方根 (SQRT) 执行错误	• (S) 为负数		修改程序使(S)为正数。
16403	BCD → BIN 转换 (BIN) 执行错误	• (S) 的任何一位数值为 0 到 9 以外的数		修改程序使(S)的每位数都在 0 到 9 范围内
16404	BIN → BCD 转换 (BCD) 执行错误	• (S) 值超出 可转化成 BCD 数据的 BIN 数据范围		修改程序使(S) 值在有效范围内
16405	自然对数(LN) 执行错误	• (S) 为 0 或负数		修改程序使(S) 为正数
16420	将软元件数据写入共享 CPU 内存 (MULTW) 执行错误	<ul style="list-style-type: none"> • 要写入的字数 (n) 超出 1 到 256 这一范围 • 写入目标软元件的 CPU, 其共享 CPU 内存地址(D) 超出 800H 到 FFFH 这一范围 • 写入目标软元件的 CPU, 其共享 CPU 内存地址(D) 和写入的字数(n)超出共享 CPU 内存地址的范围(800H 到 FFFH) • 存储写入数据的起始软元件号 (S) +要写入的字数(n)超出软元件范围。 • MULTW 指令执行前及完成位软元件打开前, 执行了 MULTW 指令 		<ul style="list-style-type: none"> • 修改程序使要写入的字数 (n) 在 1 到 256 这一范围内 • 修改程序使写入目标软元件的 CPU, 其共享 CPU 内存地址(D)在 800H 到 FFFH 这一范围内 • 修改程序使存储写入数据的起始软元件号 (S) 和写入的字数(n)在软元件范围内 • MULTW 指令的完成位软元件打开后再执行 MULTW 指令

表 18.5 操作控制 /转移执行错误 (16300 到 16599) (续)

出错代码	故障原因		故障处理	对策
	名称	说明		
16421	从其他 CPU 的共享 CPU 内存中读取软元件数据(MULTR) 执行错误	<ul style="list-style-type: none"> 要读取的字数 (n) 超出 1 到 256 这一范围 要读取的数据中的共享 CPU 内存起始地址超出共享 CPU 地址范围 (000H 到 FFFH) 要读取的数据中的共享 CPU 内存起始地址(S2) 和要读取的字数超出共享 CPU 地址范围 (000H 到 FFFH) 存储读取数据的起始软元件号 (D) 和要读取的字数超出软元件范围 3E0H/3E1H/3E2H/3E3H 未设置在 (S1)中 用(S1)指定该 CPU 复位读取的 CPU 读取的 CPU 中检测到错误 	停止正在执行的块处理, 并执行下一块	<ul style="list-style-type: none"> 修改程序使要读取的字数 (n) 在 1 到 256 这一范围内 修改程序使要读取的数据中的共享 CPU 内存起始地址(S2)在共享 CPU 地址范围 (000H 到 FFFH) 内 修改程序使存储读取数据的起始软元件号 (D) 和要读取的字数(n)在软元件范围内 修改程序使 3E0H/3E1H/3E2H/3E3H 设置在 (S1)中 修改程序使该 CPU 不被(S1)指定 检查复位标志 (M9240 到 M9243) 处于 OFF, 修改程序并执行 MULTR 指令 若读取的 CPU 中检测到错误, 更换 CPU
16422	将软元件数据写入智能功能模块/特殊功能模块 (TO) 执行错误	<ul style="list-style-type: none"> 要写入的字数 (n) 超出 1 到 256 这一范围 执行指令时, 运动 CPU 无法与智能功能模块/特殊功能模块通讯 执行指令时, 智能功能模块/特殊功能模块检测到异常 用 (D1) 指定的 I/O 号与该 CPU 控制的智能功能模块/特殊功能模块不同 用(D2) 指定的地址超出内存缓冲范围 存储写入数据的起始软元件号 (D) 和要写入的字数超出软元件范围 		<ul style="list-style-type: none"> 修改程序使要写入的字数 (n) 在 1 到 256 这一范围内 若存在故障, 更换智能功能模块/特殊功能模块 修改程序使用(D1) 指定的起始 I/O 号为该 CPU 控制的智能功能模块/特殊功能模块 修改程序使 (D2)在内存缓冲范围内 修改程序使存储写入数据的起始软元件号 (S) 和要写入的字数(n)在软元件范围内

表 18.5 操作控制 /转移执行错误 (16300 到 16599) (续)

出错代码	故障原因		故障处理	对策
	名称	说明		
16423	从智能功能模块/特殊功能模块 (FROM) 读取软元件数据 执行错误	<ul style="list-style-type: none"> 要读取的字数 (n) 超出 1 到 256 这一范围 执行指令时, 运动 CPU 无法与智能功能模块/特殊功能模块通讯 执行指令时, 智能功能模块/特殊功能模块检测到异常 用 (S1) 指定的 I/O 号与该 CPU 控制的智能功能模块/特殊功能模块不同 用 (S2) 指定的地址超出内存缓冲范围 存储写入数据的起始软元件号 (D) 和要写入的字数(n) 超出软元件范围 		<ul style="list-style-type: none"> 修改程序使要写入的字数 (n) 在 1 到 256 这一范围内 若存在故障, 更换智能功能模块/特殊功能模块 修改程序使用(S1) 指定的起始 I/O 号为该 CPU 控制的智能功能模块/特殊功能模块 修改程序使 (S2)在内存缓冲范围内 修改程序使存储写入数据的起始软元件号 (D) 和要写入的字数(n)在软元件范围内
16462	间接指定 16 位运动软元件(#(n)) 读取错误	<ul style="list-style-type: none"> 间接指定软元件号超出范围 		
16463	间接指定 32 位运动软元件 #(n)L 读取错误			
16464	间接指定 64 位运动软元件运动 #(n)F 读取错误	<ul style="list-style-type: none"> 间接指定软元件号超出范围, 或为一奇数 		
16465	间接指定 16 位数据寄存器 (D(n)) 读取错误	<ul style="list-style-type: none"> 间接指定软元件号超出范围 	停止正在执行的块处理, 并执行下一块	
16466	间接指定 32 位数据寄存器 (D(n)L) 读取错误			修改程序使间接指定软元件号正确
16467	间接指定 64 位数据寄存器 (D(n)F) 读取错误	<ul style="list-style-type: none"> 间接指定软元件号超出范围, 或为奇数 		
16468	间接指定 16 位链接寄存器 (W(n)) 读取错误	<ul style="list-style-type: none"> 间接指定软元件号超出范围 		
16469	间接指定 32 位链接寄存器 (W(n)L) 读取错误			
16470	间接指定 64 位链接寄存器 (W(n)F) 读取错误	<ul style="list-style-type: none"> 间接指定软元件号超出范围, 或为一奇数 		

表 18.5 操作控制 /转移执行错误 (16300 到 16599) (续)

出错代码	故障原因		故障处理	对策
	名称	说明		
16486	间接指定输入继电器 (X(n)) 读取错误	•间接指定软元件号超出范围	停止正在执行的块处理，并执行下一块	修改程序使间接指定软元件号正确
16487	间接指定输出继电器 (Y(n)) 读取错误			
16488	间接指定内部/锁存继电器(M(n)/L(n)) 读取错误			
16489	间接指定链接继电器 (B(n)) 读取错误			
16490	报警器 (F(n)) 读取错误			
16516	间接指定 16 位批输入继电器 (X(n)) 读取错误	•间接指定软元件号超出范围，或不是 16 的倍数		
16517	间接指定 32 位批输入继电器 (X(n)) 读取错误			
16518	间接指定 16 位批输出继电器 (Y(n)) 读取错误			
16519	间接指定 32 位批输出继电器 (Y(n)) 读取错误			
16520	间接指定 16 位批内部/锁存继电器 M(n)/L(n) 读取错误	•间接指定软元件号超出范围，或不是 16 的倍数		
16521	间接指定 32 位批内部/锁存继电器 M(n)/L(n) 读取错误			

表 18.5 操作控制 /转移执行错误 (16300 到 16599) (续)

出错代码	故障原因		故障处理	对策
	名称	说明		
16522	间接指定 16 位批内部/锁存继电器(B(n))读取错误	•间接指定软元件号超出范围, 或不是 16 的倍数	停止正在执行的块处理, 并执行下一块	修改程序使间接指定软元件号正确
16523	间接指定 32 位批内部/锁存继电器(B(n))读取错误			
16524	间接指定 16 位批报警器(F(n))读取错误			
16525	间接指定 32 位批报警器(F(n))读取错误			

18.3 运动 SFC 参数错误 (由 GSV□P 确认)

表 18.6 PLC 就绪标志 (M2000) OFF → ON 错误 (17000 到 17009)

出错代码	故障原因		故障处理	对策
	名称	说明		
17000	正常任务连续转移计算错误	• 正常任务启动的运动 SFC 程序的正常任务连续转移超出 1 到 30 这一范围	使用控制的初始值 3	断开 PLC 就绪标志 (M2000), 在设置范围内将设置值写入 CPU
17001	事件任务连续转移计算错误	• 事件任务启动的运动 SFC 程序的连续转移数超出 1 到 10 这一范围	使用控制的初始值 1	
17002	NMI 正常任务连续转移计算错误	• NMI 任务启动的运动 SFC 程序的连续转移数超出 1 到 10 这一范围		

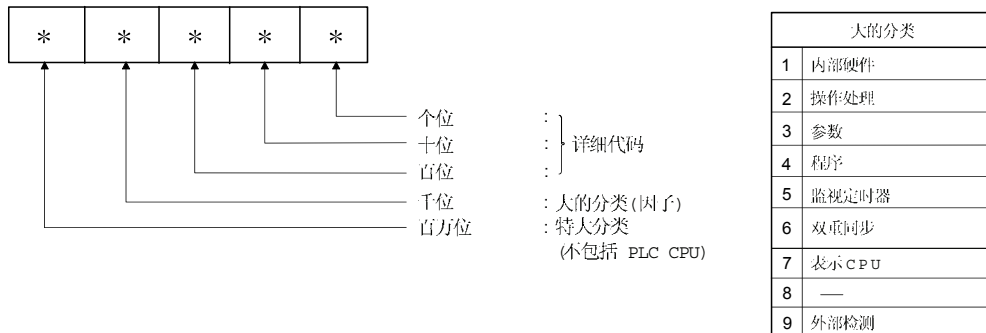
表 18.7 SFC 程序启动错误 (17010 到 17019)

出错代码	故障原因		故障处理	对策
	名称	说明		
17010	执行任务设置无效	• 在正常任务, 事件任务和 NMI 任务中, 设置一个以上任务或一个都未设置	使用控制的初始值(正常任务)	断开 PLC 就绪标志 (M2000), 在范围内设置值并写入 CPU
17011	执行任务设置无效 (事件)	• 已设置该事件任务的两个或多个固定周期		

18.4 多 CPU 出错代码 (由 GX Developer 确认)

18.4. 自诊断出错代码

本部分解释自诊断出错代码，该代码存储于 D9008。
 通过 PC 检测 / GX Developer 的 GSV □P 的软元件监视器来确认。
 出错代码的每个数位含义如下。



运动 CPU 的特征错误为 10000。(不在 PLC CPU 上发生的出错代码。)

表 18.8 发生在运动 CPU 多 CPU 故障(1000 到 10000)

中等分类	出错信息	出错代码	出错信息分类代码	发生 CPU		LED 状态		CPU 操作状态	检测时机
				单组成	多组成	RUN	ERROR		
CPU (硬件) 故障	MAIN CPU DOWN	1000	—	—	—	OFF	闪烁	停止	总是
		1001							
		1002							
		1003							
		1004							
		1005							
		1006							
		1007							
		1008							
		1009							
CPU (硬件) 故障	CPU, RAM 错误 (RAM ERROR)	1105	—	○	○	OFF	闪烁	停止	电源 ON/ 复位
	FUSE BREAK OFF	1300	—	—	—	OFF/ON	闪烁/ON	停止/继续 (注-7)	总是
	模块故障 (硬件)	SP. UNIT DOWN	1401	模块号	○	○	OFF	闪烁	停止
基板	Q 总线错误 (CONTROL-BUS ERROR)	1413	模块号	○	○	OFF	闪烁	停止	总是
		1414	模块号	○	○	OFF	闪烁 s	停止	总是
		1415	基板号 (注-2)	○	◎	OFF	闪烁	停止	总是
		1416	模块号 (注-1)	—	◎	OFF	闪烁	停止	电源 ON/ 复位
电源	AC/DC DOWN (AC/DC DOWN)检测	1500	—	○	○	ON	OFF	继续	总是
电池	(BATTERY ERROR)	1600	驱动器名称	○	○	ON	OFF	继续	总是
		1601							
智能功能模块/多 CPU 模块的操作处理	智能功能模块安装错误 (SP. UNIT LAY ERROR)	2121	模块号	—	◎	OFF	闪烁	停止	电源 ON/ 复位
		2124							
		2125							
		2126	模块号 (注-1)						

(注-1): CPU 号储存于正常信息分类的槽号中。

(注-2): “出错信息分类代码”的“普通信息分类代码”中的基板号: 0: CPU 基板, 1 到 7: 延伸基板数。

(注-3): 停止错误或 CPU 号 (除复位的 CPU 号外) 与 MULTI CPU DOWN 同步运行, 按时间先后存储到出错信息分类中。

(注-4): 运动 CPU (除 PLC CPU 外) 出现错误时, 通过 PLC CPU 从 GX Developer 中进行 CPU (除 PLC CPU 外) 的 PC 检测, 此时出错代码为“1000”。

(注-5): 打开运动 SFC 错误检测信号 (M2039) 即开始检测错误。此时错误自诊断标志 (M9008) 和 错误检测标志 (M9010) 未打开。运动 SFC 错误检测信号 (M2039) 动 ON → OFF 时复位, 设置于 D9008 的出错代码“1000”。

(注-6): 发生停止错误时, 关闭运动 RUN LED。(RUN LED 条件不变。)

(注-7): 发生错误时, CPU 操作状态可由参数设置。(LED 的显示亦不断变化。)

18 出错代码列表

出错代码	故障内容及原因	对策	备注
1000	主 CPU 失控或出故障。 (1) 因噪音或其他原因操作失误。 (2) 硬件出故障。	(1) 测试噪音等级。 (2) 复位并重建 RUN 状态。若出现同样错误, 则有 CPU 硬件错误。解释错误类型并向销售代表咨询。	
1001			
1002			
1003			
1004			
1005			
1006			
1007			
1008			
1009			
1105	CPU 的共享 CPU 内存出故障。	(1) 测试噪音等级。 (2) 复位并重建 RUN 状态。若出现同样错误, 则有 CPU 硬件错误。解释错误类型并向销售代表咨询。	
1300	输出模块保险丝熔断。	检查输出模块的 ERR. LED 并更换 LED 变亮的模块。	
1401	初始通讯时, 运动模块或智能功能模块无响应。	运动专门模块, 智能功能模块, CPU 模块或基板有硬件错误。解释错误类型并向销售代表咨询。	
1413	Q 总线检测到错误。 检测到 CPU 或扩展基板故障。 打开电源或复位时检测到总线错误。	特殊功能模块, CPU 模块或基板有硬件错误。解释错误类型并向销售代表咨询。	
1414			
1415			
1416			
1500	电源出现一时中断。电源被切断。	检查电源。	
1600	(1) CPU 电压不足。 (2) 未安装 CPU 电池的主接头。	(1) 更换电池。 (2) 若电池用于 RAM 或备份电源, 安装主接头。	
1601	电池的电压不足。	更换电池。	
2121	A CPU 模块未安装于 CPU 槽, 0 到 2 槽。	将 CPU 模块安装于 CPU 槽或 0 到 2 槽。	
2124	(1) 模块安装于 65 或以上的槽。 (2) 模块安装于基板设置为“无”的基板中。	(1) 从 65 或以上的槽中移出模块。 (2) 从基板设置为“无”的基板中移出模块。	
2125	(1) 安装了 PLC CPU 不承认的模块。 (2) T 智能功能模块无响应。	(1) 在 PLC CPU 中安装可用模块。 (2) 智能功能模块有硬件错误。解释错误类型并向销售代表咨询。	
2126	多 CPU 系统的 CPU 模块位置为下列任一种。 (1) CPU 模块间未安装槽。 (2) PLC CPU 模块间安装了除 PLC CPU 以外的模块。	(1) 多 CPU 系统的 CPU 模块间未安装槽。(若有, 请撤消。) (2) 移出 PLC CPU 模块间的模块 (除 PLC CPU 模块), 并转移到多 CPU 系统 PLC CPU 模块中的槽。	

○ : 出现在检测到错误的 CPU (CPU 号) 中。
 ◎ : 多 CPU 合成时, 出现在所有 CPU 号中。
 - : 未发生。

表 18.8 发生在运动 CPU 中的多 CPU 错误(续)

中等分类	出错信息	出错代码	出错信息分类代码	发生 CPU		LED 状态		CPU 操作状态	检测时机
				单组成	多组成	RUN	ERROR		
参数	PARAMETER ERROR	3001	文档名称	○	○	OFF	闪烁	停止	电源 ON/ 复位/从停止到 运行时
		3010		—	⊙				
		3012		—	○				
		3013		—	⊙				
多 CPU	Other issue opportunity CPU weight occasion error (MULTI CPU DOWN)	7000	模块号 (注-1) (注-3)	—	○	OFF	闪烁	停止	总是
		7002	模块号 (注-1)	—	○				电源 ON/ 复位
		7003		—	○				
	多 CPU 启动错误 (MULTI EXE. ERROR)	7010	模块号 (注-1)	—	○	OFF	闪烁	停止	电源 ON/ 复位
	多 CPU 启动错误 (MULTI CPU ERROR)	7020	模块号 (注-1)	—	○	ON	ON	继续	总是
除 PLC CPU 外的 CPU 错误	除 PLC CPU 外的 CPU 错误(CONT. UNIT ERROR) (注-4) (注-5)	10000	—	除 PLC CPU 外		ON	ON : 系统 设置错误/伺 服错误 OFF : 其他错 误	停止 : 系统设置 错误 继续 : 其他错误	电源 ON/ 复位/从停止到 运行时

(注-1) : CPU 号储存于正常信息分类的槽号中。

(注-2) : “出错信息分类代码”的“普通信息分类代码”中的基板号: 0 : CPU 基板, 1 到 7 : 扩展基板数。

(注-3) : 停止错误或 CPU 号(除复位的 CPU 号外)与 MULTI CPU DOWN 同步运行, 按时间先后存储到出错信息分类中。

(注-4) : 运动 CPU (除 PLC CPU 外)出现错误时, 通过 PLC CPU 从 GX Developer 中进行 CPU (除 PLC CPU 外)的 PC 检测, 此时出错代码为“1000”。

(注-5) : 打开运动 SFC 错误检测信号 (M2039) 即开始检测错误。此时错误自诊断标志 (M9008) 和 错误检测标志 (M9010) 未打开。运动 SFC 错误检测信号 (M2039) 动 ON → OFF 时复位, 设置于 D9008 的出错代码“1000”。

(注-6) : 发生停止错误时, 关闭运动 RUN LED。(RUN LED 条件不变。)

18 出错代码列表

出错代码	故障内容及原因	对策	备注
3001	参数内容被破坏。	(1) 读取外围设备的详细出错信息，检查并更正参数项目与数值（参数号）相一致。 (2) 更正参数后若仍发生错误，可能是 CPU 的内部 RAM 或内存错误。解释错误症状并向销售代表咨询。	
3010	参数设置的 CPU 模块与实际安装在多 CPU 系统中的不同。	使(多 CPU 设置的当前计算) - (I/O 任务的 CPU (空) 设置)与 CPU 的实际安装相符。	
3012	参数设置的参考 CPU 号与多 CPU 系统设置的不同。	使参数设置与参考 CPU 号 (CPU No.1) 相符。	
3013	多 CPU 系统中，多 CPU 自动刷新设置为下列任一种。 (1) 位软元件作为刷新软元件时，刷新起始软元件不是 16 的倍数。 (2) 指定了不可指定的软元件。 (3) 转移点数为奇数。	检查下列多 CPU 自动刷新参数并更正。 (1) 指定位软元件时，刷新起始软元件是 16 的倍数。 (2) 指定可指定的软元件作为刷新软元件。 (3) 转移点数为偶数。	
7000	多 CPU 系统中，在操作模式下，“选择所有站由 CPU 停止错误”的 CPU 出现 CPU 故障。 (发生在 CPU 中，除已选择系统停止的 CPU。)	读取外围设备的个别出错信息，检查 CPU 故障及错误并修改。	
7002	多 CPU 系统中，CPU No.1 在打开电源时出现停止错误，其他 CPU 无法启动。（这种错误出现在 CPU No.2 到 4）	复位并重新运行 PLC CPU。若发生同一错误，则有 CPU 硬件故障。解释错误症状并向销售代表咨询。	
7003			
7010	(1) 多 CPU 系统中安装了有故障的 CPU。 (2) 多 CPU 系统中安装了不兼容的 CPU 版本。(此错误由功能版本 B 的 PLC CPU T 检测。) (3) 多 CPU 系统开启电源后，释放复位的 CPU 号 No.2 到 4。（只在释放复位的 CPU 号时发生此错误。）	读取外围设备的个别出错信息后，功能版本 A 的 CPU 号或终止模块改为功能版本 B 的。	
7020	多 CPU 系统中，在操作模式下，“选择所有站由 CPU 停止错误”的 CPU 出现 CPU 故障。 (在 PLC CPU（除发生 CPU 错误的 CPU 号以外）发生此错误。)	读取外围设备的个别出错信息，检查 CPU 故障及错误并修改。	
10000	发生运动 CPU 特征错误。 设置系统错误时(伺服程序设置错误，轻度异常，重度异常，伺服错误，实模式/虚模式转换错误，运动 SFC 程序错误)检测运动 CPU。	使用应用 CPU 模块的软件包检查发生的错误。	

- ：出现在检测到错误的 CPU (CPU 号)中。
 ◎：多 CPU 合成时，出现在所有 CPU 号中。
 -：未发生。

18.4.2 自诊断错误消除 (由 GSV □P 执行)

只有当错误允许 CPU 继续操作时, CPU 可释放错误。按下列步骤释放错误。

- 1) 查找错误原因。
- 2) 存储出错代码并释放到特殊寄存器 D9060 中。
- 3) 打开特殊继电器 M9060 。
- 4) 释放目标错误。

清除错误复位 CPU 后, 特殊继电器、特殊寄存器及故障 LED 返回发生错误时的状态。消除错误后若再次发生同样错误, 会再次寄存。

附录

附录 1 处理时间

附录 1.1 运算控制 / 移位指令处理时间

(1) 运算指令

分类	符号	指令	运算表达式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
二进制运算	=	赋值	#0=#1	6.30
			D800=D801	10.20
			#0L=#2L	8.70
			D800L=D802L	13.56
			#0F=#4F	8.88
			D800F=D804F	15.30
	+	加	#0=#1+#2	9.72
			D800=D801+D802	13.50
			#0L=#2L+#4L	11.52
			D800L=D802L+D804L	16.68
			#0F=#4F+#8F	13.26
			D800F=D804F+D808F	19.20
	-	减	#0=#1-#2	10.02
			D800=D801-D802	13.14
			#0L=#2L-#4L	10.68
			D800L=D802L-D804L	22.50
			#0F=#4F-#8F	12.06
			D800F=D804F-D808F	19.26
	*	乘	#0=#1*#2	8.76
			D800=D801*D802	12.66
			#0L=#2L*#4L	8.46
			D800L=D802L*D804L	18.12
			#0F=#4F*#8F	12.30
			D800F=D804F*D808F	19.14
/	除	#0=#1/#2	10.08	
		D800=D801/D802	13.02	
		#0L=#2L/#4L	13.62	
		D800L=D802L/D804L	20.52	
		#0F=#4F/#8F	14.16	
		D800F=D804F/D808F	20.04	
%	取余	#0=#1%#2	10.74	
		D800=D801%D802	15.06	
		#0L=#2L%#4L	13.20	
		D800L=D802L%D804L	20.76	
位运算	~	位取反 (补码)	#0=~#1	7.68
			D800=~D801	11.22
			#0L=~#2L	9.60
			D800L=~D802L	14.64
	&	位与	#0=#1	3.78
			D800=D801&D802	12.78
			#0L=#2LL	10.80
			D800L=D802L&D804L	18.24

分类	符号	指令	运算式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
位运算		位或	#0=#1#2	8.40
			D800=D801 D802	12.36
			#0L=#2L#4L	10.68
			D800L=D802L D804L	12.54
	^	位异或	#0=#1^#2	8.76
			D800=D801^D802	10.80
			#0L=#2L^#4L	10.62
			D800L=D802L^D804L	15.60
	>>	位右移	#0=#1>>#2	11.76
			D800=D801>>D802	15.00
			#0L=#2L>>#4L	11.82
			D800L=D802L>>D804L	18.06
	<<	位左移	#0=#1<<#2	10.50
			D800=D801<<D802	12.24
			#0L=#2L<<#4L	12.18
			D800L=D802L<<D804L	15.90
符号	-	符号取反 (2的补码)	#0=#1	7.02
			D800=-D812	11.70
			#0L=-#2L	8.76
			D800L=-D802L	14.34
			#0F=-#4F	11.28
			D800F=-D804F	15.84
标准函数	SIN	正弦	#0F=SIN(#4F)	19.80
			D800=SIN(D804F)	25.68
	COS	余弦	#0F=COS(#4F)	13.20
			D800=COS(D804F)	24.54
	TAN	正切	#0F=TAN(#4F)	19.86
			D800=TAN(D804F)	30.78
	ASIN	反正弦	#0F=ASIN(#4F)	21.18
			D800=ASIN(D804F)	33.48
	ACOS	反余弦	#0F=ACOS(#4F)	23.52
			D800=ACOS(D804F)	34.80
	ATAN	反正切	#0F=ATAN(#4F)	15.30
			D800=ATAN(D804F)	19.62
	SQRT	平方根	#0F=SQRT(#4F)	10.68
			D800=SQRT(D804F)	15.42
	LN	自然对数	#0F=LN(#4F)	16.92
			D800=LN(D804F)	22.26
	EXP	指数运算	#0F=EXP(#4F)	18.54
			D800=EXP(D804F)	25.14
	ABS	绝对值	#0F=ABS(#4F)	12.90
			D800=ABS(D804F)	16.02
RND	四舍五入	#0F=RND(#4F)	12.24	
		D800=RND(D804F)	12.42	
FIX	舍	#0F=FIX(#4F)	11.40	
		D800=FIX(D804F)	20.28	
FUP	入	#0F=FUP(#4F)	12.00	
		D800=FUP(D804F)	16.92	

分类	符号	指令	运算表达式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
标准函数	BIN	BCD→BIN 变换	#0=BIN(#1)	8.82
			D800=BIN(D801)	12.30
			#0L=BIN(#2L)	11.16
			D800L=BIN(D802L)	14.82
	BCD	BIN→BCD 变换	#0=BCD(#1)	13.92
			D800=BCD(D801)	17.70
			#0L=BCD(#2L)	14.94
			D800L=BCD(D802L)	26.10
类型变换	SHORT	变换为 16 位整数型 (带符号)	#0=SHORT(#2L)	10.14
			#0=SHORT(#4F)	14.70
			D800=SHORT(D802L)	14.40
			D800=SHORT(D804F)	17.40
	USHORT	变换为 16 位整数型 (无符号)	#0=USHORT(#2L)	9.90
			#0=USHORT(#4F)	14.52
			D800=USHORT(D802L)	14.10
			D800=USHORT(D804F)	16.50
	LONG	变换为 32 位整数型 (带符号)	#0=LONG(#2)	9.00
			#0=LONG(#4F)	12.48
			D800=LONG(D802)	12.90
			D800=LONG(D804F)	18.60
	ULONG	变换为 32 位整数型 (无符号)	#0=ULONG(#2)	9.30
			#0=ULONG(#4F)	47.22
			D800=ULONG(D802)	7.62
			D800=ULONG(D804F)	50.10
	FLOAT	变换为 64 位浮点小数点型 (带符号)	#0=FLOAT(#4)	9.12
			#0=FLOAT(#4L)	9.48
			D800F=FLOAT(D804)	13.56
			D800F=FLOAT(D804L)	15.00
	UFLOAT	变换为 64 位浮点小数点型 (无符号)	#0=UFLOAT(#4)	7.92
			#0=UFLOAT(#4L)	10.26
			D800F=UFLOAT(D804)	13.26
			D800F=UFLOAT(D804L)	15.06
位软元件状态	(无)	ON (正常打开触点) (条件满足时)	SET M1000 = M0	13.74
			SET M1000 = X100	14.26
			SET M1000 = PX0	14.82
	!	OFF(正常关闭触点) (条件满足时)	SET M1000 = !M0	13.38
			SET M1000 = !X100	14.40
			SET M1000 = !PX0	14.82
位软元件控制	SET	软元件设置	SET M1000	3.42
			SET Y100	10.74
			SET PY0	14.58
	RST	软元件复位	RST M1000	3.30
			RST Y100	10.02
			RST PY0	11.16
	DOUT	软元件输出	DOUT M0,#0	9.42
			DOUT M0,#0L	10.14
			DOUT Y100,#0	9.48
			DOUT Y100,#0L	12.30
			DOUT PY0,#0	8.76
			DOUT PY0,#0L	15.48

分类	符号	指令	运算表达式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
位软元件控制	DIN	软元件输入	DIN #0,M0	8.88
			DIN #0L,M0	10.20
			DIN #0,X0	9.12
			DIN #0L,X0	9.66
			DIN #0,PX0	10.56
			DIN #0L,PX0	11.10
	OUT	位软元件输出	OUT M1000 = M0	19.26
			OUT Y0 = M0	21.90
			OUT PY0 = M0	20.88
逻辑预算	*	与	SET M1000 = M0*M1	15.96
			SET M1000 = X100*X101	14.70
			SET M1000 = PX0*PX1	17.40
	+	或	SET M1000 = M0+M1	15.66
			SET M1000 = X100+X101	14.94
			SET M1000 = PX0+PX1	16.20
比较运算	==	等于(条件满足时)	SET M1000 = #0==#1	11.40
			SET M1000 = D800==D801	14.10
			SET M1000 = #0L==#2L	13.98
			SET M1000 = D800L==D802L	18.42
			SET M1000 = #0F==#4F	14.64
			SET M1000 = D800F==D804F	18.48
	!=	不等于(条件满足时)	SET M1000 = #0!=#1	12.72
			SET M1000 = D800!=D801	15.24
			SET M1000 = #0L!=#2L	13.98
			SET M1000 = D800L!=D802L	18.54
			SET M1000 = #0F!=#4F	16.02
			SET M1000 = D800F!=D804F	18.66
	<	小于(条件满足时)	SET M1000 = #0<#1	10.56
			SET M1000 = D800<D801	16.14
			SET M1000 = #0L<#2L	16.26
			SET M1000 = D800L<D802L	18.78
			SET M1000 = #0F<#4F	16.32
			SET M1000 = D800F<D804F	16.32
	<=	小于或等于(条件满足时)	SET M1000 = #0<=#1	12.60
			SET M1000 = D800<=D801	16.14
			SET M1000 = #0L<=#2L	14.04
			SET M1000 = D800L<=D802L	18.42
			SET M1000 = #0F<=#4F	16.50
			SET M1000 = D800F<=D804F	19.32
>	大于(条件满足时)	SET M1000 = #0>#1	12.18	
		SET M1000 = D800>D801	15.72	
		SET M1000 = #0L>#2L	14.64	
		SET M1000 = D800L>D802L	19.74	
		SET M1000 = #0F>#4F	15.30	
		SET M1000 = D800F>D804F	19.86	

分类	符号	指令	运算表达式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
比较运算	=>	大于或等于(条件满足时)	SET M1000 = #0=>#1	12.12
			SET M1000 = D800=>D801	15.84
			SET M1000 = #0L=>#2L	14.16
			SET M1000 = D800L=>D802L	19.38
			SET M1000 = #0F=>#4F	16.44
			SET M1000 = D800F=>D804F	21.84
运动专用功能	CHGV	速度变更请求	CHGV(K1,#0)	13.80
			CHGV(K1,D800)	15.72
			CHGV(K1,#0L)	14.70
			CHGV(K1,D800L)	18.36
	CHGT	转矩限制值变更请求	CHGT(K1,#0)	6.84
			CHGT(K1,D800)	8.70
			CHGT(K1,#0L)	3.60
			CHGT(K1,D800L)	11.40
其他	EI	事件任务使能	EI	3.78
	DI	事件任务禁止	DI	3.66
	NOP	空操作	NOP	1.44
	BMOV	块传送	BMOV #0,#100,K10	4.80
			BMOV D800,D100,K10	11.94
			BMOV #0,#100,K100	34.80
			BMOV D800,D100,K100	37.98
			BMOV N1,#0,K512	67.86
			BMOV N1,D800,K512	73.14
	MULTW	向自 CPU 的共享 CPU 内存写入软元件数据	MULTW H800,#0,K1,M0	21.72
			MULTW H800,D800,K1,M0	22.14
			MULTW H800,#0,K10,M0	22.86
			MULTW H800,D800,K10,M0	28.92
			MULTW H800,D800,#0,K100,M0	42.36
			MULTW H800,D800,K100,M0	44.70
			MULTW H800,#0,K256,M0	81.06
			MULTW H800,D800,K256,M0	85.38
	MULTR	从其它 CPU 的共享内存读取数据	MULTR #0,H3E0,H800,K1	44.16
			MULTR D800,H3E0,H800,K1	44.76
			MULTR H800,#0,K10,M0	51.48
			MULTR #0,H3E0,H800,K10	51.00
			MULTR D800,H3E0,H800,K10	134.88
			MULTR #0,H3E0,H800,K100	135.60
			MULTR #0,H3E0,H800,K256	269.94
			MULTR D800,H3E0,H800,K256	270.96
	TO	向智能功能模块/特殊功能模块写入数据	TO H0,H0,#0,K1	27.78
			TO H0,H0,D800,K1	27.30
			TO H0,H0,#0,K10	34.50
TO H0,H0,D800,K10			34.80	
TO H0,H0,#0,K100			105.78	
TO H0,H0,D800,K100			120.90	
TO H0,H0,#0,K256			227.52	
TO H0,H0,D800,K256			249.24	

分类	符号	指令	运算表达式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
其他	FROM	从智能功能模块/特殊功能模块读取数据	FROM #0,H0,H0,K1	31.20
			FROM D800,H0,H0,K1	28.14
			FROM #0,H0,H0,K10	36.30
			FROM D800,H0,H0,K10	37.44
			FROM #0,H0,H0,K100	119.70
			FROM D800,H0,H0,K100	116.82
			FROM #0,H0,H0,K256	247.98
			FROM D800,H0,H0,K256	246.90
	TIME	等待	TIME K1	13.26
			TIME #0	19.50
			TIME D800	16.62

② 转移条件表达式

分类	符号	指令	运算表达式	Q173CPU(N)/Q172CPU(N) 单位 [μs]
位软元件控制	(无)	ON (常开触点) (条件成立时)	M0	2.82
			X100	6.88
			PX0	7.62
	!	OFF (关闭触点) (条件成立时)	!M0	3.24
			!X100	8.46
			!PX0	9.24
逻辑运算	*	逻辑 AND	#0==#1	10.32
			D800==D801	11.28
			#0L==#2L	12.36
	+	逻辑 OR	#0!=#1	5.28
			D800!=D801	9.78
			#0L!=#2L	11.10
比较运算	==	等于	#0==#1	1.26
			D800==D801	9.48
			#0L==#2L	7.74
			D800L==D802L	13.32
			#0F==#4F	9.36
			D800F==D804F	12.66
	!=	不等于	#0!=#1	1.38
			D800!=D801	9.42
			#0L!=#2L	8.16
			D800L!=D802L	12.06
			#0F!=#4F	9.60
			D800F!=D804F	12.24
	<	小于	#0<#1	1.68
			D800<D801	9.90
			0L<#2L	7.50
			D800L<D802L	12.48
			#0F<#4F	9.30
			D800F<D804F	13.80
	<=	小于或等于	#0<=#1	2.76
			D800<=D801	9.48
			#0L<=#2L	2.82
			D800L<=D802L	13.02
			#0F<=#4F	4.26
			D800<=D804F	11.40
>	大于	#0>#1	6.48	
		D800>D801	4.80	
		#0L>#2L	7.98	
		D800L>D802L	7.38	
		#0F>#4F	9.12	
		D800F>D804F	11.40	
>=	大于或等于	#0>=#1	1.26	
		D800>=D801	9.36	
		#0L>=#2L	2.70	
		D800L>=D802L	12.06	
		#0F>=#4F	4.32	
		D800F>=D804F	13.08	

(3) F 与 G 组合的处理时间 (F/G 中说明的程序为 NOP)

	F 单独	G 单独	F+G	G SUB	CLR	JMP/耦合
Q173CPU(N)/ Q172CPU(N) [μs]	31.92	28.38	34.5	87.24	47.3	22.86

	并列分支(2个)		并列分支(5个)	
	分支时	连接时	分支时	连接时
Q173CPU(N)/ Q172CPU(N) [μs]	50.82	50.34	83.94	116.34

	选择分支(2个)		选择分支(5个)	
Q173CPU(N)/ Q172CPU(N) [μs]	139.68		196.02	

④:启动或消除的程序有很大不同

要点

处理时间过长可导致运动 CPU WDT 错误或伺服异常，尤其对于事件/NMI 任务管理的运动 SFC 程序，注意处理时间不可过长。

附录 2 示例程序

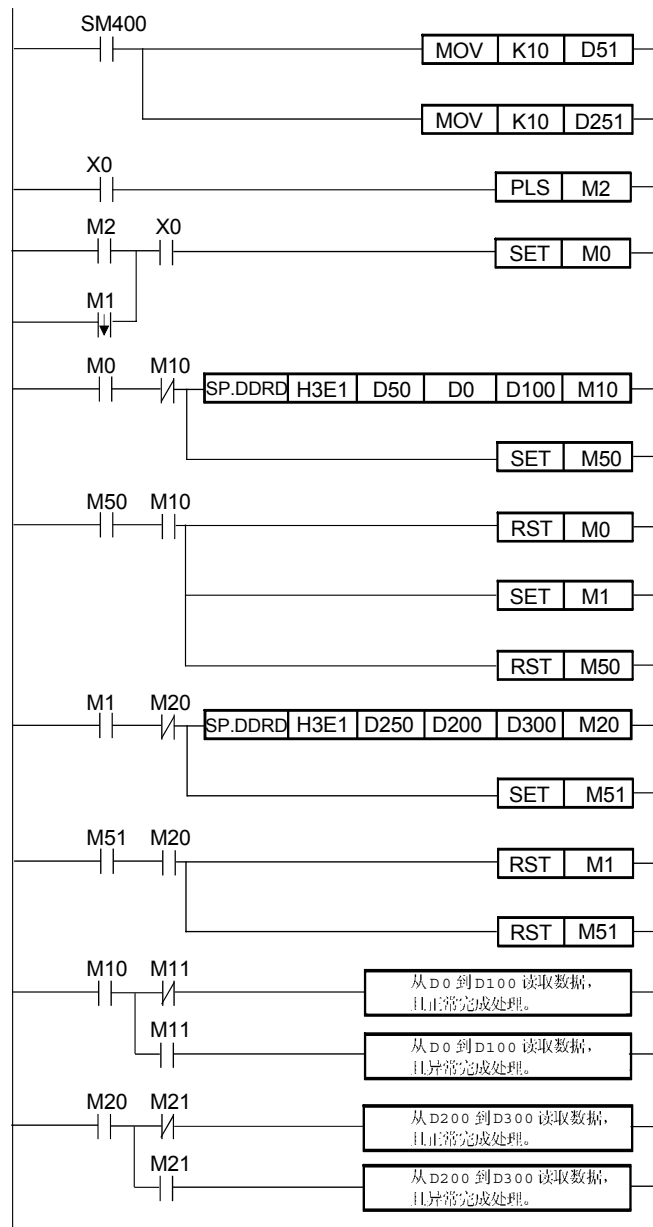
附录 2.1 连续执行多 CPU 专用指令的程序示例

这是在向运动 CPU 的多专用指令中朝着同一运动 CPU 连续出版指令的程序示例。

当一指令不被认可时，即使该指令在执行，变为“空操作”。

以下是在 X0 为 ON 期间从安装在 CPU2 号机的运动 CPU 的 D0 到 PLC CPU 的 D100 重复读 10 点数据，和从运动 CPU 的 D200 到 PLC CPU 的 D300 连续依次读取 10 点数据的程序举例。当 S(P).DDR D 指令执行一个扫描周期使软元件为 ON 后执行下一个 S(P).DDR D 指令。

<示例>



以上示例有以下限制:

- 1) 运动CPU的多CPU指令不可用作中断程序/固定周期执行型程序及低速执行型程序。否则,指令可能在时序中不能正常执行。

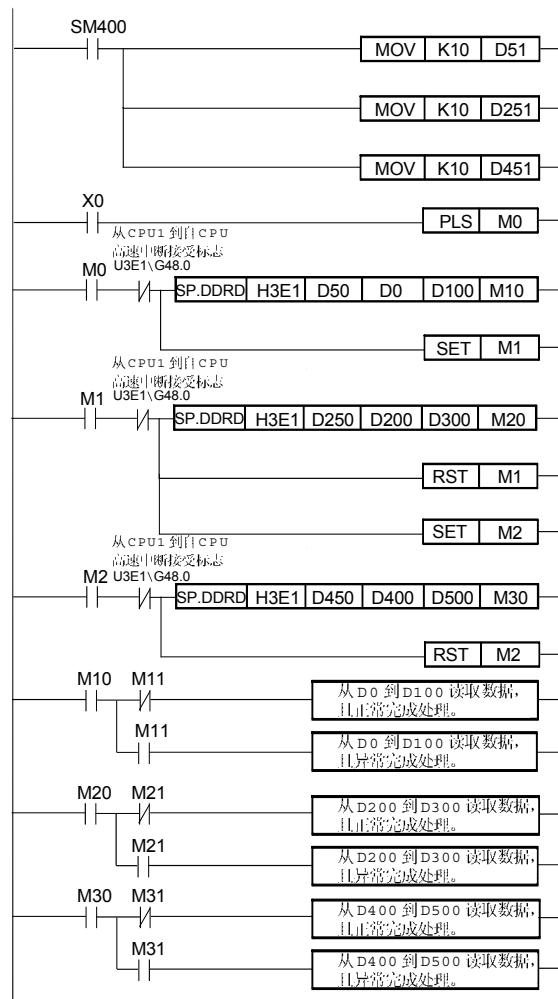
附录 2.2 使用一次指令执行多个多 CPU 指令的程序示例

这是使用一个指令高速执行多个相同运动 CPU 的程序示例。

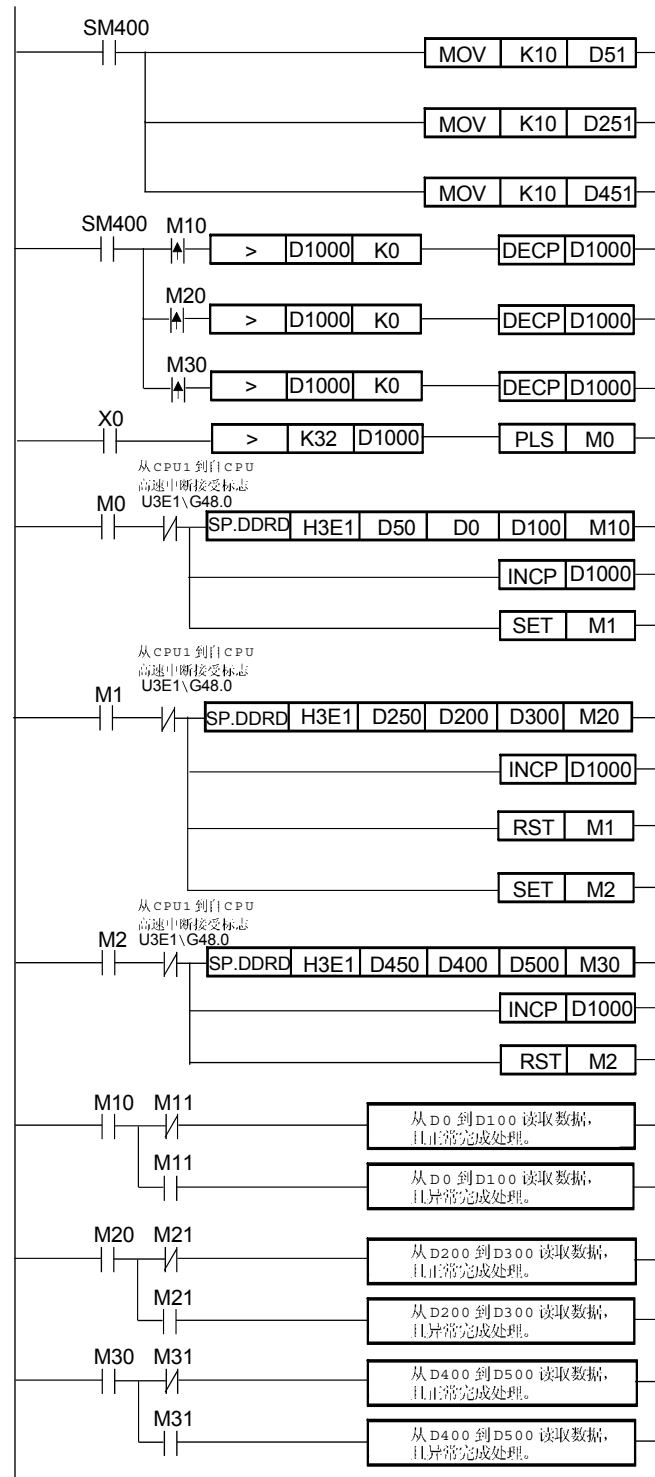
在本情形下，必须与“从 CPU 到自 CPU 高速中断接受标志”进行互锁。当一指令不被认可时，即使该指令在执行，变为“空操作”。

以下示例 1 是在 X0 起动时，从安装在 CPU 2 号机的运动 CPU 的 D0 到 PLC CPU 的 D100 读取 10 点数据，从运动 CPU 的 D200 到 PLC CPU 的 D300 的读取 10 点数据，及从运动 CPU 的 D400 到 PLC CPU 的 D500 的读取 10 点数据。此时，在执行一次指令的多 CPU 专用执行指令数不可超过一个运动 CPU 的最大可执行指令数 (参见第 5 章)。如最大可接受指令数为 32，当多 CPU 专用执行指令数超过 32 时，用来不执行多专用指令的程序如下所示为示例 2。

<示例 1>



<示例 2>



示例 2 的有以下限制。

- 1) 运动 CPU 的多 CPU 指令不可用作中断程序 / 固定周期执行型程序及低速执行型程序。否则, 指令可能在时序中不能正常执行。

附录 2.3 进行运动控制运动 SFC 程序组合举例

(1) 执行运动控制的运动 SFC 程序示例

该程序示例说明参见以下各功能。

示范程序功能列表

序号	项目	说明
1	定位专用软元件监视	运动 CPU (CPU 2 号机)的定位专用软元件状态在 PLC CPU (CPU 1 号机)的"M2400 到 M3039"及 "D0 到 D639"上反映。
2	时钟数据读取	打开时钟数据读取请求(M9028), 可将时钟数据设置为出错历史。
3	紧急停止	分配到 PX0 的紧急停止输入打开时, 接通所有轴, 且执行运动控制。紧急停止输入关闭时, 伺服放大器紧急停止, 取消运动控制, 且断开输出 (PY)。
4	运动控制	在下列模式中, 根据 PX1 及 PX2 的条件执行运动控制: <ul style="list-style-type: none"> • PX2 : OFF PX1 : OFF JOG 模式 • PX2 : OFF PX1 : ON 手动脉冲发生器模式 • PX2 : On PX1 : OFF 原点回归模式 • PX2 : On PX1 : On 编程模式
5	JOG 模式	打开 PX3 到 PX6 的各信号, 执行以下 JOG 操作: <ul style="list-style-type: none"> • PX3 : 1 轴 JOG 正转 • PX4 : 1 轴 JOG 逆转 • PX5 : 2 轴 JOG 正转 • PX6 : 2 轴 JOG 逆转
6	手动脉冲发生器模式	执行以下手动脉冲发生器操作: <ul style="list-style-type: none"> • 与手动脉冲发生器 P1 一起执行 1 轴手动脉冲发生器操作。 • 与手动脉冲发生器 P1 一起执行 2 轴手动脉冲发生器操作。
7	原点回归模式	执行以下原点回归模式。 <ul style="list-style-type: none"> • 打开 PX3, 执行 1 轴原点回归。 • 打开 PX4, 执行 2 轴原点回归。
8	编程模式	执行以下程序操作: <ul style="list-style-type: none"> • 当 PX3 检测 OFF 到 ON 时, 执行 2 号轴的定位, 然后执行 1 号轴的定位然后等待 1000[ms]。 • PX4 开启时, 执行 2 号轴的定位, 然后执行 1, 2 号轴直线插补及定位检查, 程序等待直到 1, 2 号轴从相反方向以双倍速度执行直线插补完成后且断开 PX4。

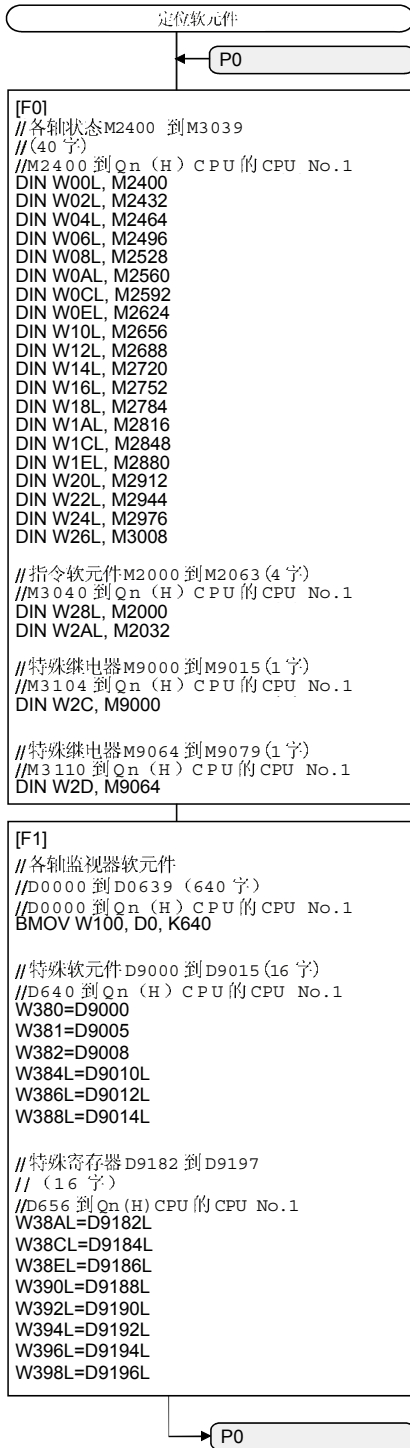
② 运动 SFC 程序的处理内容
运动 SFC 程序列表

序号	程序名称	任务	自动操作	可连接 移位数	处理内容
0	定位软元件	正常	启动	3	<ul style="list-style-type: none"> 该程序在 Q173CPU(N)的运行时间自动启动, 并总被执行。 监视器的定位专用软元件(位软元件)被传送到"W0 到"。 监视器的定位专用软元件(字软元件)被传送到"W100 到"。 (注): "W0 到" 被分配到 PLC CPU (CPU 1 号机)的"M2400 到", 且 "W100 到"被自动刷新设置分配到"D0 到"
20	主程序	正常	启动	3	<ul style="list-style-type: none"> 该程序在 Q173CPU(N)的运行时间自动启动, 并总被执行。 监视数据被取出, 且时钟数据读取请求(M9028)打开。 当紧急停止被取消时, 子程序启动“110号: 运动控制”。 紧急停止时“110号: 运动控制”也停止, 并断开输出(PY)。
110	运动控制	正常	不启动	3	<ul style="list-style-type: none"> 所有轴伺服处于启动状态。 根据 PX1, PX2 的以下条件执行子程序调用 1) PX2: OFF PX1: OFF 120号: JOG 2) PX2: OFF PX1: ON 130号: 手动脉冲发生器 3) PX2: ON PX1: OFF 140号: 原点回归 4) PX2: ON PX1: ON 150号: 编程操作
120	JOG	正常	不启动	3	(1) 设置 1 轴及 2 轴的 JOG 运行速度。 (2) 当 PX3 打开时, 启动 1 轴 JOG 正转; 当 PX4 打开时, 启动 1 轴 JOG 逆转。 (3) 当 PX5 打开时, 启动 2 轴 JOG 正转; 当 PX6 打开时, 启动 2 轴 JOG 逆转。 (4) 当 PX2/PX1 为 OFF 时重复以上(2), (3)操作。为 ON 时, 停止 1 轴及 2 轴的 JOG 正转及逆转指令并结束程序。
130	手动脉冲发生器	正常	不启动	3	<ul style="list-style-type: none"> 设置 1 轴及 2 轴的 1 脉冲放大倍率。 用 P1 控制 1 轴, 并设置用 P2 控制 2 轴, 打开 P1, P2 的手动脉冲发生器使能标志。 除了 PX2: OFF, PX1: ON (手动脉冲发生器模式)之外, 关闭 P1, P2 的手动脉冲发生器使能标志, 并结束程序。

运动 SFC 程序列表 (续)

序号	程序名称	任务	自动操作	可连接 移位数	处理内容
140	原点回归	正常	不启动	3	<ul style="list-style-type: none"> 当 PX3 为 ON 时, 启动 “K140: 1 轴原点回归”; 当 PX4 打开时, 启动 “K141: 2 轴原点回归”。 PX2 : ON; PX1 : 为除 OFF 之外的状态时结束程序(原点回归模式)。
150	编程操作	正常	不启动	3	<ul style="list-style-type: none"> 当 PX3 检测 OFF 到 ON, 执行 1 轴定位之后, 等待 1000[ms] 再执行 2 轴定位。 当 PX4 启动时, 执行完直线插补定位检查之后, 在相反方向以双倍速度执行 1, 2 号轴直线插补定位, 并等待直到 PX4 断开。 PX2: ON; PX1 : 除 ON 之外时结束程序 (编程模式)。

(a) 0 号:定位软元件



(1) 传送 M2400 到 M3039 (32 轴用) 各轴状态到 “W0 到”。

(2) 传送指令软元件 M2000 到 M2064 到 “W28 到”。

(3) 传送特殊继电器 M9000 到 M9015 到 “W2C 到”

多 CPU 间的自动刷新, 及 Q173CPU (N) (CPU No.2) 的 “W0 到” 设置为用 Qn (H) CPU (CPU No.1) 的 “M2400 到” 进行刷新, 因此 Q173CPU (N) (CPU No.2) 的条件可通过监视以下软元件和 CPU No.2 的 Qn (H) CPU 一起获得。

QnH CPU (CPU No.1) 的软元件	Q173CPU (N) (CPU No.2) 的对应软元件
M2400 到 M3039	M2400 到 M3039
M3040 到 M3103	M2000 到 M2064
M3104 到 M3119	M9000 到 M9015

注: 本示例中的 32 轴刷新数据, 刷新点数为系统规定的最小值, 以减少处理时间。

(1) 传送各监视软元件 D0 到 D639 (32 轴用) 到 “W100 到”

(2) 传送特殊寄存器 D9000 到 D9015 到 “W380 到”

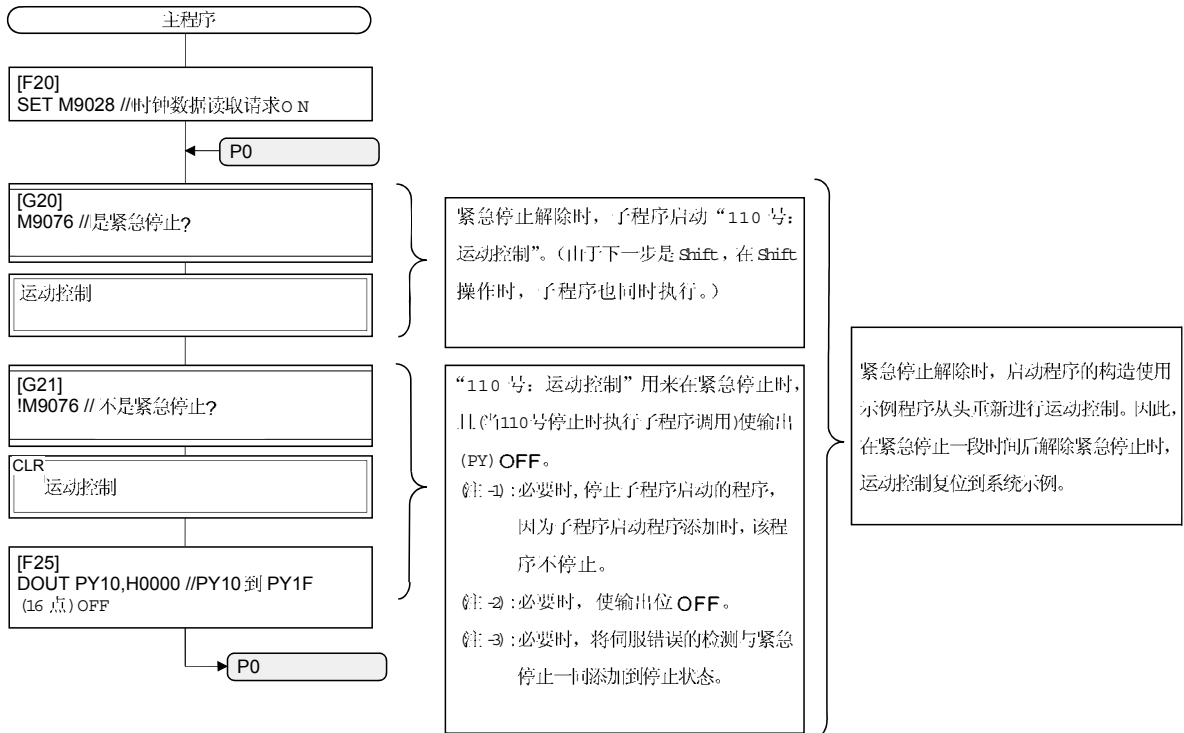
(3) 传送特殊寄存器 D9182 到 D9197 到 “W38A 到”

多 CPU 间的自动刷新, 及 Q173CPU (N) (CPU No.2) 的 “W100 到” 设置为用 Qn (H) CPU (CPU No.1) 的 “D0 到” 进行刷新, 因此 Q173CPU (N) (CPU No.2) 的条件可通过监视以下软元件和 CPU No.1 的 Qn (H) CPU 一起获得。

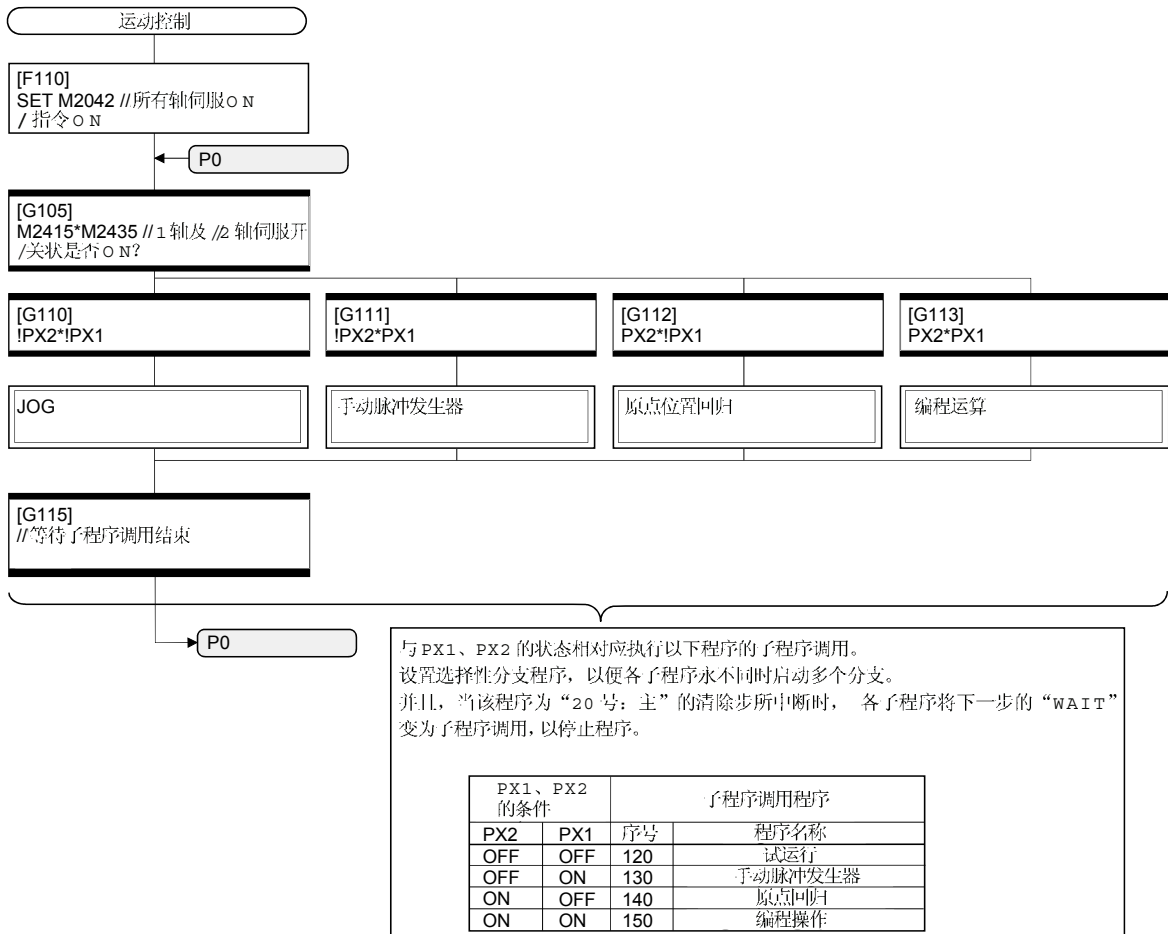
QnH CPU (CPU No.1) 的软元件	Q173CPU (N) (CPU No.2) 的对应软元件
D0000 到 D0639	D0000 到 D0630
D0640 到 D0655	D9000 到 D9015
D0656 到 D0671	D9182 到 D9197

注: 本示例中的 32 轴刷新数据, 刷新点数为系统规定的最小值, 以减少处理时间。

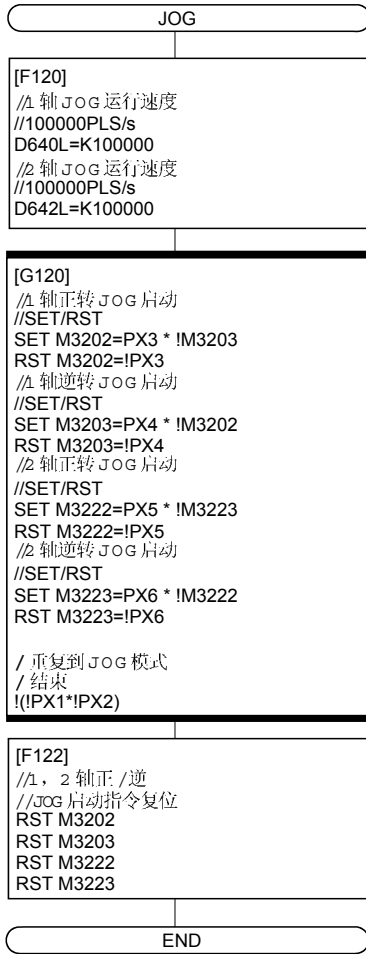
(b) 20号:主程序



(c) 110号:运动控制



(d) 120 号: JOG



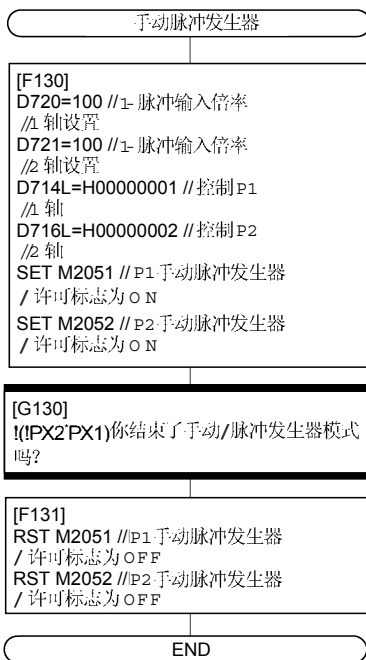
当 PX3 到 PX6 的各信号打开/断开时, 对应的 JOG 指令软元件为 SET/RST。
 同时打开, 使同一轴的正转 JOG 启动及逆转 JOG 启动。

信号名称	对应的 JOG 指令软元件
PX3	M3202 (1 轴正转 JOG)
PX4	M3203 (1 轴逆转 JOG)
PX5	M3222 (1 轴正转 JOG)
PX6	M3223 (1 轴逆转 JOG)

注): 各信号的 ON/OFF 区别可用 Y/N 来表示。
 但是, 在以下所述只可用 SET=/RST= 描述的处理情形下, 即使步很多, 处理时间也可大大缩短, 因为它被设置为低。

1、2 轴的正/逆转 JOG 状态在 JOG 模式结束时关闭, JOG 模式转为其他安全模式时, JOG 运动停止。

(e) 130 号: 手动脉冲发生器

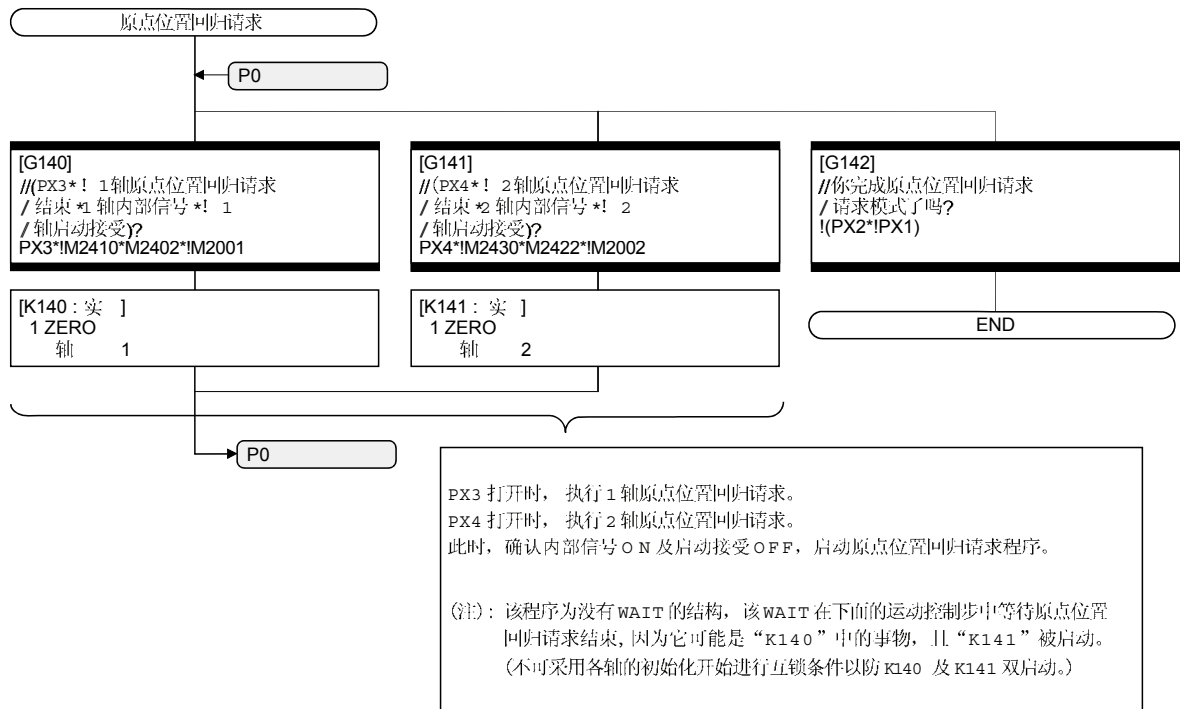


以下设定执行 P1 控制 1 轴/P2 控制 2 轴的手动脉冲发生器的操作。

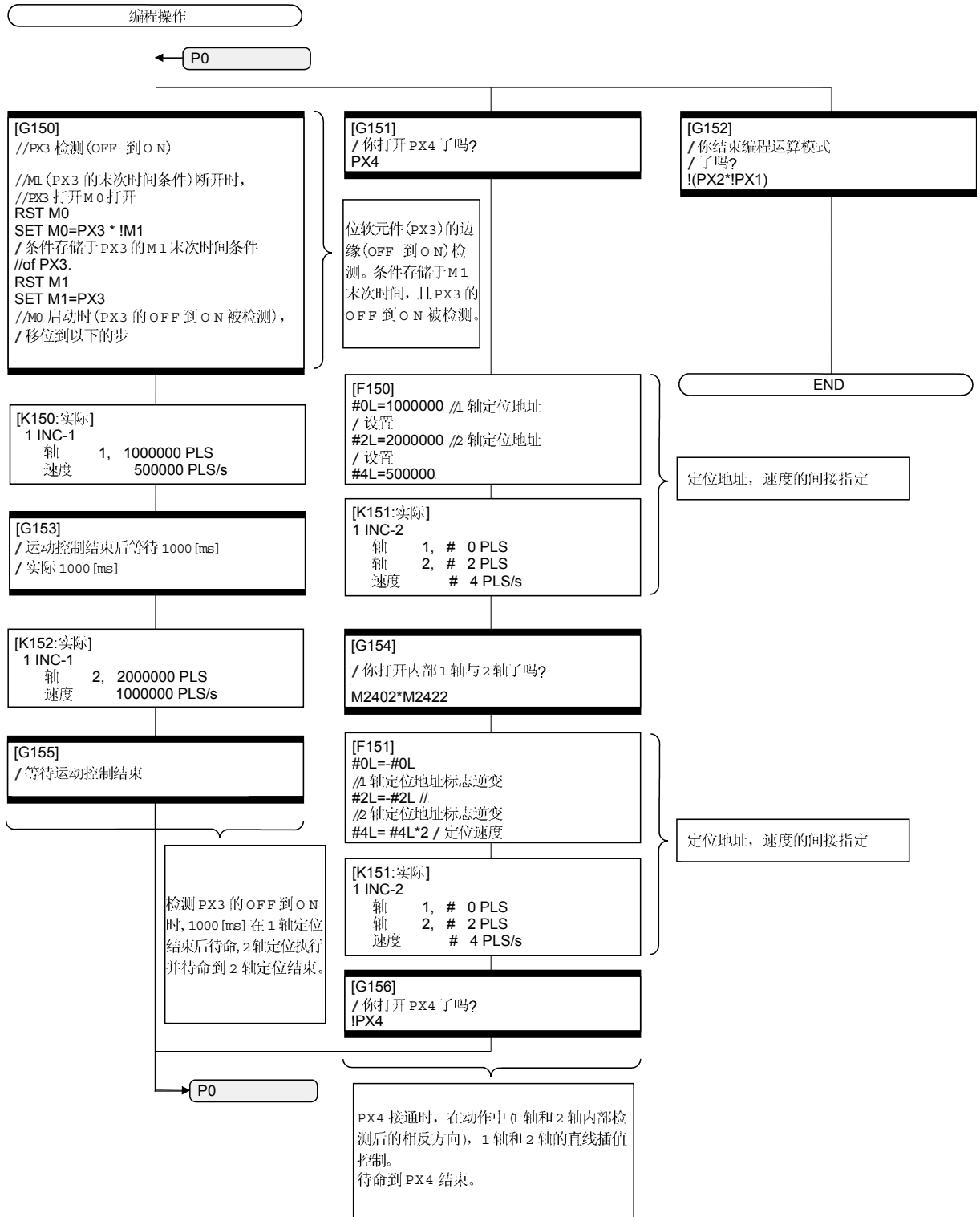
- 1 轴和 2 轴的 1 个脉冲输入放大倍率设置。
- 设置手动脉冲发生器的轴号以 P1 控制 1 轴/P2 控制 2 轴。
- 打开 P1、P2 的手动脉冲发生器许可标志。

1、2 轴手动脉冲发生器许可标志在 JOG 模式结束时断开, 当其转为其他安全模式时, 手动脉冲发生器操作停止。

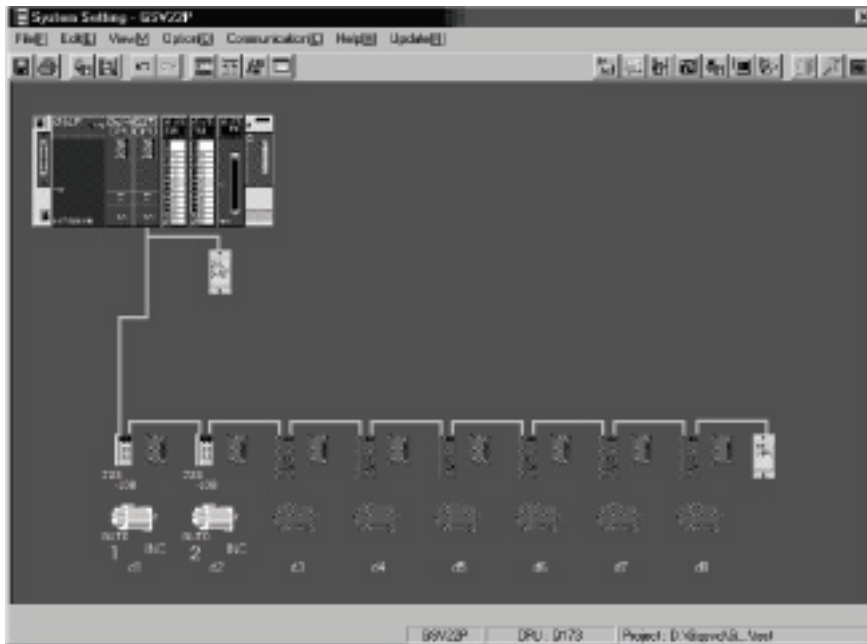
140 号:原点回归



⑨ 150 号:编程操作



(2) 运动 CPU 的系统设置数据
系统设置如下所示。



(a) 模块设置
手动脉冲发生器输入模块 (Q173PX : 插槽 3)

轴号	说明
P1	手动脉冲发生器, 同步编码器 (INC)
P2	手动脉冲发生器, 同步编码器(INC)
P3	手动脉冲发生器, 同步编码器(INC)
I/O 反应时间	0.4[ms]

(b) 基本设置
1) 多 CPU 设置

设置项	说明
多 CPU 数	2 个模块
操作模式	CPU No.1/2 机发生停止错误时, 所有 CPU 停止。

2) 自动刷新设置 1

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	
	点数	开始	结束	开始	结束
No.1	0	---	---	---	---
No.2	50	0800	0831	W0	W31
No.3					
No.4					

该软元件区在"M2400"中与 Qn(H) CPU No.1 一起设置。(监视器的位软元件被 Q173CPU(N) 侧的运动 SFC 程序传送至 "W0 至" 。

3) 自动刷新设置 2

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	
	点数	开始	结束	开始	结束
No.1	0	---	---	---	---
No.2	640	0832	0AB1	W100	W37F
No.3					
No.4					

该软元件区在"D0"中与 Qn (H) CPU 1 号机一起设置。(监视器的字软元件被 Q173CPU(N) 侧的运动 SFC 程序传送至 "W100 至" 。

4) 自动刷新设置 3

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	
	点数	开始	结束	开始	结束
No.1					
No.2					
No.3					
No.4					

该设置区用作监视的定位软元件之外的用途。

5) 自动刷新设置 4

CPU	各 CPU 的发送范围			CPU 侧软元件	
	CPU 共享内存 G			起始软元件	
	点数	开始	结束	开始	结束
No.1					
No.2					
No.3					
No.4					

6) 系统设置

设置项	说明
运行周期设置	自动
操作模式	用开关(停止到运行)打开 M2000
紧急停止输入	PX0

7) 锁存范围设置

项目	符号	锁存(1)		锁存(2)	
		开始	结束	开始	结束
内部继电器	M				
链接继电器	B				
报警器	F				
数据寄存器	D				
链接寄存器	W				

锁存 (1) : 可用锁存清除进行清除。

锁存 (2) : 不可用锁存清除进行清除。

8) PLC 模块设置

模块类型	点数	占用软元件	基板	插槽号	I/O 响应时间
输入	16	000-00F	CPU 基板	1	10[ms]
输出	16	010-01F	CPU 基板	2	

3) Qn(H) CPU 1 号机的参数设置

PC 参数项		Qn(H) 参数				
		说明				
1	CPU 数	2 个模块				
2	操作模式	CPU 停止期间错误操作模式				
		CPU No.1	停止错误发生时所有站停止			
		CPU No.2	停止错误发生时所有站停止			
3	组外输入设置	采用组外输入条件				
	组外输出设置	不采用组外输出条件				
4	刷新设置	设置 CPU No.1				
		各 CPU 发送范围			CPU 侧软元件	
		共享 CPU 内存 G				
		点	开始	结束	起始软元件	M2400
			开始	结束	开始	END
		CPU No.1	0	—	—	—
	CPU No.2	50	0800	0831	M2400	M3199
	设置 CPU 2 号机					
	各 CPU 发送范围			CPU 侧软元件		
	共享 CPU 内存 G					
点	开始	结束	起始软元件	D0		
	开始	结束	开始	END		
CPU No.1	0	—	—	—		
CPU No.2	640	0832	0AB1	D0	D639	

附录 2.4 用运动 SFC 程序进行子程序重启的连续执行示例

(1) 操作说明

这是从运动控制步连续执行的程序示例，该运动控制步在运动控制进行期间用清除步停止子程序后重启时中途停止。

伺服被紧急停止解除打开，且该程序中 PX4 启动时 2 轴线性插补的定位控制被执行。在确认 PX4 断开后，一个周期运行完成。定位操作期间执行紧急停止时，定位操作中断且伺服电机停止。当紧接着紧急停止解除时，通过中断定位操作来进行复位。

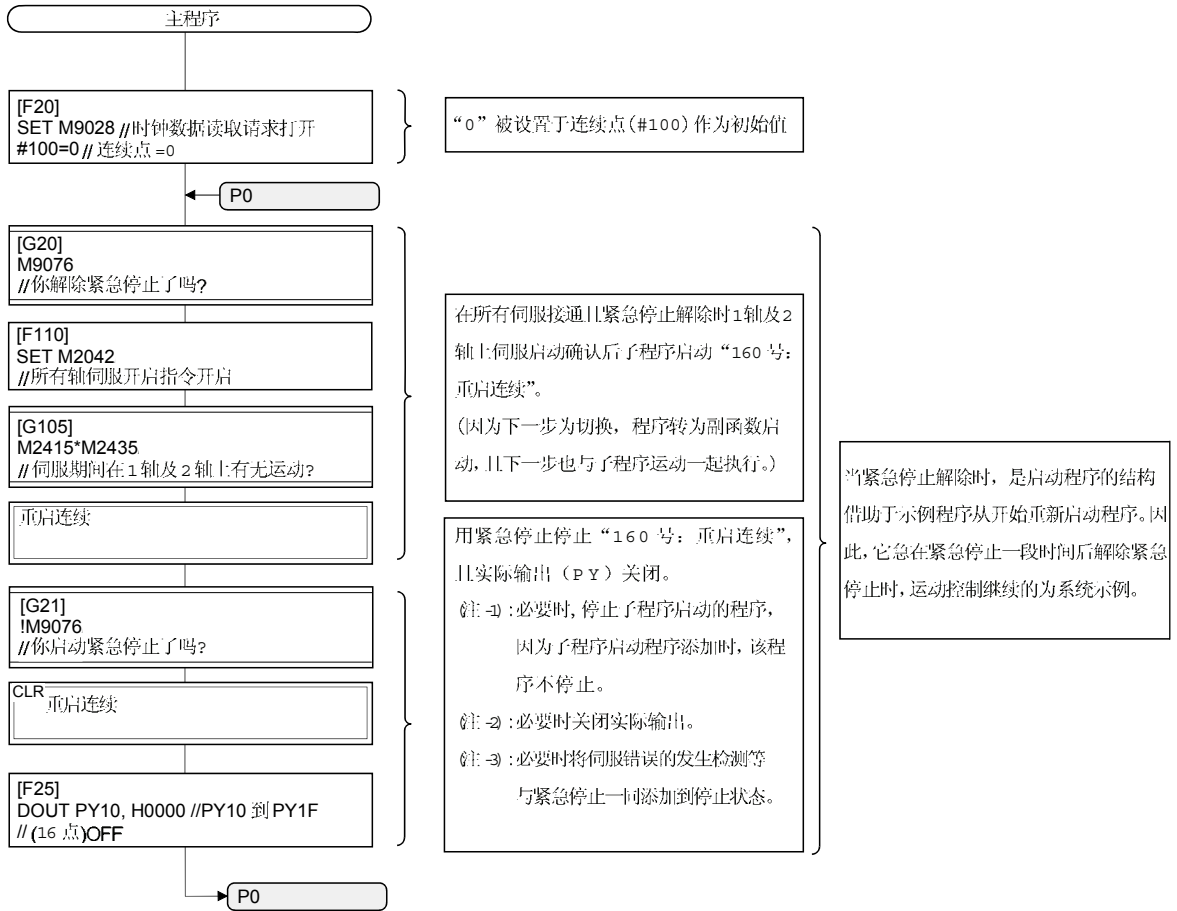
借助以下步骤用本程序示例执行子程序重启的连续执行。

- (a) 执行带子程序的运动控制时，不论用户软元件中运动控制步定位是否完成，运动控制被记忆。
- (b) 从停止以上 (a) 所记忆信息的运动控制步进行子程序重启复位。
- (c) 定位期间停止而后复位时，运动控制步应该定位。
- (d) 不论定位期间伺服电机是否停止，使用定位结束信号 (M2401+20n) 作为判断信号。

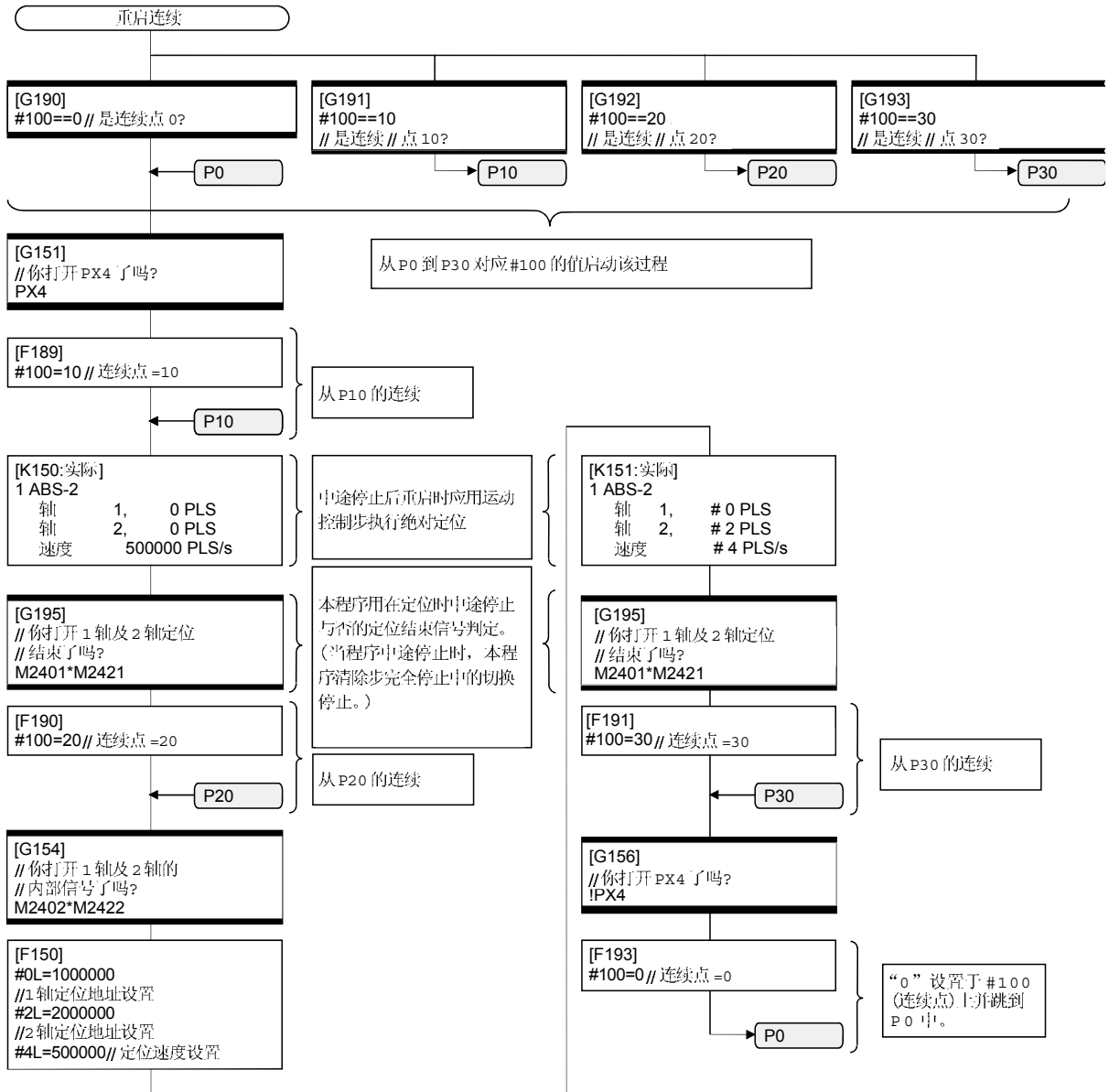
② 运动 SFC 程序的处理内容
运动 SFC 程序列表

序号	程序名称	任务	自动操作	可连接移位 位数	处理内容										
20	主程序	正常	启动	3	<ul style="list-style-type: none"> • Q173CPU(N)处于 RUN 时, 该程序自动启动并总被执行。 • 读取监视数据, 打开时钟数据读取请求 (M9028)。 • 在连续点(#100:用户软元件)设置 "0"作为初始植。 • 紧急停止解除时, 打开所有轴伺服且 1 轴和 2 轴的伺服启动确认后, 子程序启动 " 160 号:重启连续"。 • "160 号:重启连续"在紧急停止时停止, 并断开输出 (PY)。 										
160	重启连续	正常	不启动	3	<p>(1) 根据以下(2) 1) 到 9)连续点(#100)的值执行程序跳转。</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>#100</th> <th>跳跃目标</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>以下(2) 1)</td> </tr> <tr> <td>10</td> <td>以下(2) 3)</td> </tr> <tr> <td>20</td> <td>以下(2) 5)</td> </tr> <tr> <td>30</td> <td>以下(2) 8)</td> </tr> </tbody> </table> <p>(2) 执行以下运动控制。</p> <ol style="list-style-type: none"> 1) 该程序处于等待状态直到 PX4 为 ON。 2) 在连续点上(#100)设置"10"。 3) 1 轴、2 轴从(0,0)作线性控制(2 轴绝对定位)。 4) 确认 1 轴、2 轴的定位结束信号开启后, 设置 "20"在连续点 (#100)上。 5) 确认 1 轴及 2 轴定位信号。 6) 1 轴、2 轴在线性控制中位于(1000000, 2000000) (绝对 2 轴定位)。 7) 确认 1 轴、2 轴的定位结束信号接通后, 设置 "30"在连续点 (#100)上。 8) 该程序处于等待状态直到 PX4 断开。 9) "0" 设置在连续点上(#100)。 	#100	跳跃目标	0	以下(2) 1)	10	以下(2) 3)	20	以下(2) 5)	30	以下(2) 8)
#100	跳跃目标														
0	以下(2) 1)														
10	以下(2) 3)														
20	以下(2) 5)														
30	以下(2) 8)														

② 20 号：主程序



b) 160 号:重启连续



附录 2.5 用运动 SFC 程序停止后连续执行示例

(1) 操作说明

该程序示例说明因来自输入模块的紧急停止, 运动 SFC 程序被外部输入信号 ON 所停止, 且由于以下所示停止, 程序被外部信号 OFF 连续执行。

本程序中当 PX4 接通时, 伺服被紧急停止解除打开, 且 2 轴线性插补的定位控制被执行。确认 PX4 OFF 时, 一周期的运行完成。

当定位操作中 PX5 打开时, 定位操作被停止指令停止并在打开 PX5 时从中断的定位操作复位。在 WAIT 移位中 PX5 为 ON 期间不执行到下一步的移位。

在定位操作中执行紧急停止时, 定位操作被中断且伺服电机停止。

在紧急停止被解除时程序从中断的定位操作复位。

借助以下步骤用本程序示例执行连续停止。

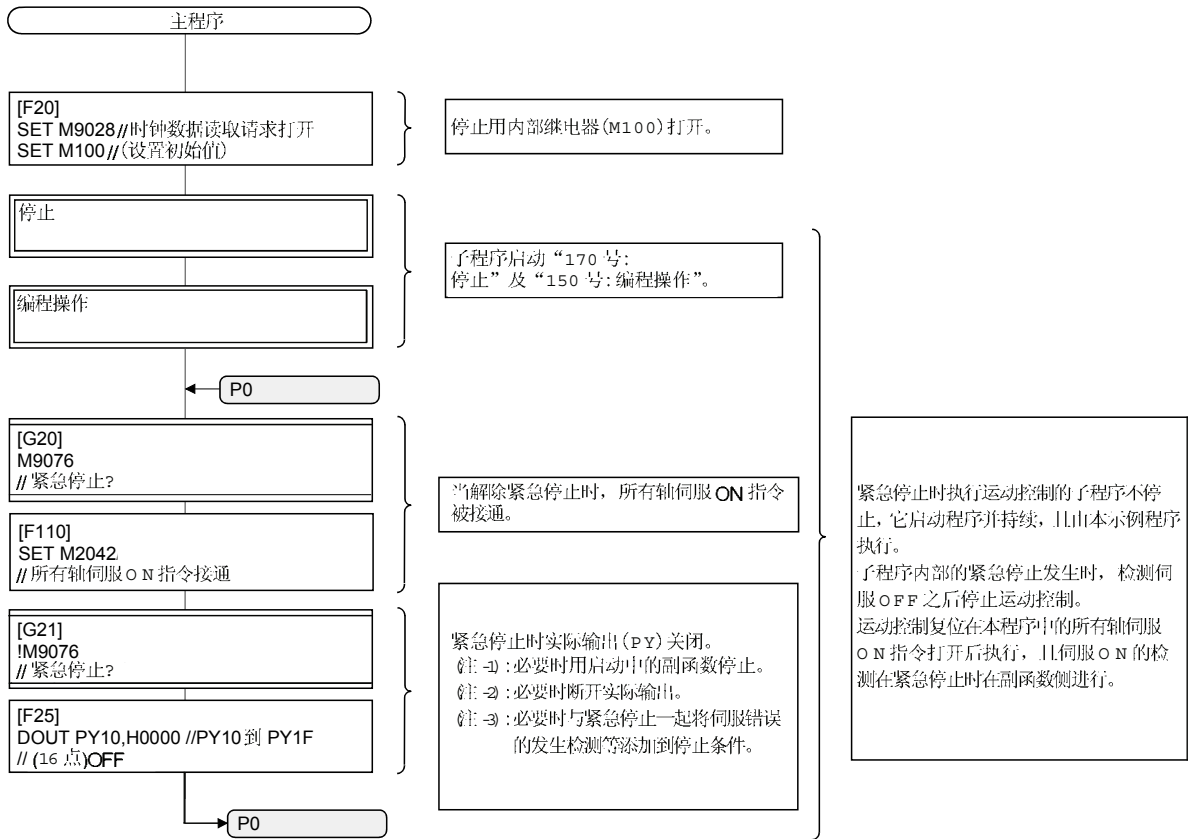
- ㉑ PX5 启动时, 程序可启动停止用停止指令 (M3200+20n) 及内部继电器 (M100)。
- ㉒ PX5 断开时, 程序可断开停止用停止指令 (M3200+20n) 及内部继电器 (M100)。
- ㉓ 当在定位过程中停止后复位时, 运动控制步应有与之对应的绝对位置。
- ㉔ 定位结束信号 (M2401+20n) 被用来决定, 是否在定位时中途停止。
- ㉕ 当在定位中被用作步时, 运动控制步在等待断开后复位。
- ㉖ “停止用内部继电器 (M100) 断开” 被替换为必须停止的 WAIT 移位条件。

② SFC 程序处理内容

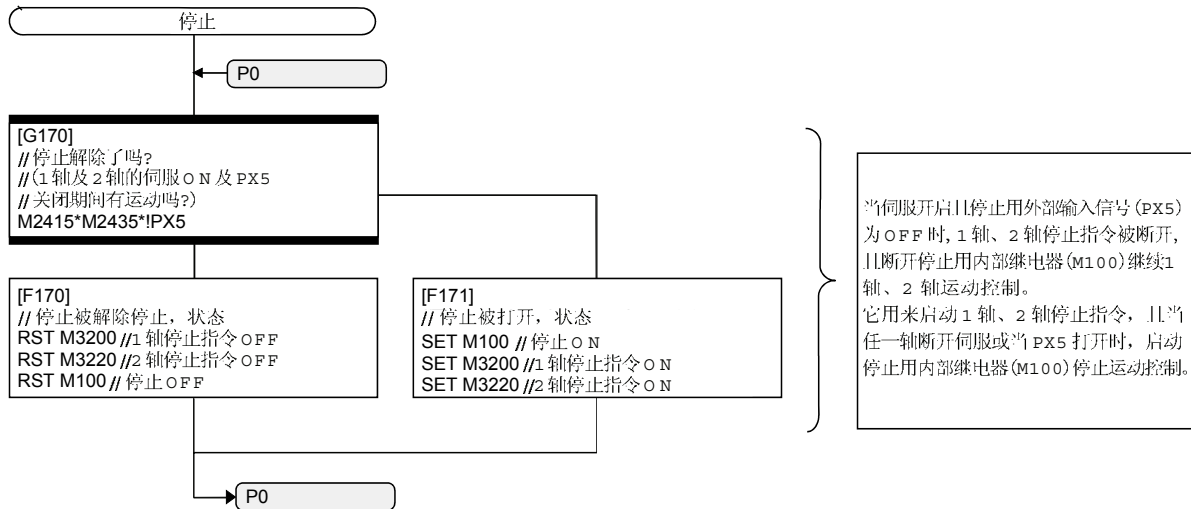
SFC 程序列表

序号	程序名称	任务	自动操作	连接性移位 位数	处理内容
20	主程序	正常	启动	3	<ul style="list-style-type: none"> • 该程序在 Q173CPU 的 RUN 时自动启动, 并总被执行。 • 监视数据被取出, 且时钟数据读取请求(M9028)打开。 • 停止用内部继电器(M100)的初始化条件打开。 • 子程序启动 " 170 号 :停止"。 • 子程序启动" 150 号: 编程操作"。 • 解除紧急停止时, 所有轴伺服打开。 • 紧急停止时断开实际输出(PY)。
170	停止	正常	不启动	3	<p>(1) 当来自输入模块的停止输入信号(PX5)关闭时, 执行以下(2)的处理方法, 且在伺服开启期间, 在规定的唯一的情形下, 1 轴及 2 轴执行以下(3)。</p> <p>(2) 1 轴及 2 轴停止指令被断开, 且停止用内部继电器被断开。</p> <p>(3) 1 轴及 2 轴停止指令被打开, 且停止用内部继电器被打开。</p>
150	编程操作	正常	不启动	3	<p>(1) 执行以下运动控制。</p> <ol style="list-style-type: none"> 1) 该程序处于等待状态直到 PX4 打开。 2) 1 轴及 2 轴设于线性插补(绝对 2 轴定位)的(0,0)中。 3) 确认 1 轴及 2 轴的定位结束信号接通。 4) 确认 1 轴及 2 轴的定位开启。 5) 1 轴及 2 轴设于线性控制(绝对 2 轴定位)的(1000000, 2000000)中。 6) 确认 1 轴及 2 轴的定位结束信号接通。 7) 该程序处于等待状态直到 PX4 断开。 <p>(2) 当以上(1) 3) 及 6)的定位结束信号断开时, 等待断开, 且(定位中断时)重新执行运动控制步骤(1) 2) 或 5)。</p> <p>(3) 直到停止用内部继电器(M100)将之打开, 它不会转到以上(1) 1) 及 7)的下一步。</p>

(a) 20 号:主程序



(b) 170 号:停止



(c) 150 号:编程操作

