

Koyo

Value & Technology

DL250-1/DL260 系列轴定位模块

Z-01PM

技术资料

光洋电子(无锡)有限公司

目次

1. 一般规格	4
2. 电气规格	4
3. 功能规格	5
4. 基本功能	6
4-1. 脉冲输出	6
4-2. 座標系	7
4-3. 電子齿轮	7
4-4. 送进速度	7
4-5. 超调	8
4-6. 间隙补偿	8
4-7. 行程监控	8
4-8. 加减速功能	9
4-9. 原点搜索	10
4-10. 定位图形	10
4-11. 停止功能	10
4-12. 停电保持功能	10
4-13. 文件管理功能	11
4-14. 显示功能	11
4-15. 外部输出端子	12
5. 输出继电器	14
5-1. PLC输出继电器	14
5-2. PLC输入继电器 (Z-01PM→PLC)	14
5-3. PLC输出继电器 (PLC→Z-01PM)	16
5-4. PLC输出继电器的配合	17
5-5. Z-01PM的内部输出继电器	18
6. 运转功能	19
6-1. 手动运转	20
6-2. 点动运转	20
6-3. 原点搜索	20
6-4. 自动运转	29
7. 数据转送	30
7-1. 指针方式	30
7-2. WT方式	31
7-3. 共用RAM	32
7-4. Z-01PM(0.12 或以前版本)使用补充说明	33
8. 监视功能	36
8-1. 监视数据	36
8-2. 监视数据的传送 (PLC ← PM)	37
9. 命令	38
9-1. 命令详情	39
10. 参数	46
10-1. 系统参数	46
10-2. 原点参数	51
10-4. 参数的传送 (PLC → PM)	52
11. 数据寄存器	56
11-1. 各寄存器的说明	56
11-2. 数据的流动	57
11-3. 数据的传送	57
12. 程序	62
12-1. 程序中的文字、記号	62
12-2. 程序的命令形态	63
12-3. 定位控制命令	66
12-4. 程序编制	87
12-5. 程序的传送	88
12-6. 文件的读出	94
13. 配線	95
14. 运转	97
14-1. 状态迁移	97

1 4-2. 各状态的处理	99
1 4-3. 起动	107
1 4-4. 手动/点动运转	108
1 4-5. 原点检索	108
1 4-6. 自动运转	108
1 4-7. 简易程序	109
1 5. 出错码	112
1 6. 附录	115
1 6-1. 时序图	115
1 6-2. 状态迁移一览	121
1 6-3. 存储器映像	123
1 6-4. 用WT方式的数据传送概要	124
1 7. 安全方面的注意事项	126

1. 一般规格

动作环境温度	: 0 ~ 55℃
保存环境温度	: -20 ~ 70℃
动作环境湿度	: 30 ~ 95% (不得凝露)
保存环境湿度	: 30 ~ 95% (不得凝露)
使用环境气氛	: 不得有腐蚀性气体
使用环境污染度	: 2
耐电压	: AC 1500V 1分钟 (FG-LG、SG-LG) AC 500V 1分钟 (FG-SG) 截止电流 0.5mA
绝缘电阻	: DC 500V 10MΩ以上 (FG、SG、LG 各相互间)
耐振动	: 按照JIS C0040
耐冲击	: 按照JIS C0041
抗干扰	: 按照NEMA ICS3-304 脉冲干扰 1000V / 1μs
不要辐射	: 按照FCC A档
RFI	: 150、430MHz 10W / 100mm
(-2) 消费电流(5V)	: 350mA

2. 电气规格

《外部供给2.4V电源》

输入电压范围	: DC 2.1.6 ~ 2.6.4V (2.4V±10%)
消费电流	: 7.0mA

《EMR、ERR、HOME(HOM)、INP、OVT±(OT±)、共用输入(I1、I2)》

() 内为脉冲表示

输入电压范围	: DC 2.1.6 ~ 2.6.4V (2.4V±10%)
额定输入电流	: 4.9mA (加2.4.0V)
最大输入电流	: 6.0mA (加2.6.4V)
输入阻抗	: 4.9kΩ
最小ON电压	: 2.0.0V 以上
最小ON电流	: 4.0mA 以上
最大OFF电压	: 3.0V 以下
最大OFF电流	: 1.3mA 以下
(-1) OFF→ON延迟时间	: 5.0μs ~ 400μs
(-1) ON→OFF延迟时间	: 100μs ~ 400μs

《INT、Z输入》

输入电压范围	: DC 2.1.6 ~ 2.6.4V (2.4V±10%)
额定输入电流	: 4.9mA (加2.4.0V)
最大输入电流	: 6.0mA (加2.6.4V)
输入阻抗	: 4.9kΩ
最小ON电压	: 2.0.0V 以上
最小ON电流	: 4.0mA 以上
最大OFF电压	: 3.0V 以下
最大OFF电流	: 1.3mA 以下
OFF→ON延迟时间	: 1.0μs ~ 100μs
ON→OFF延迟时间	: 1.0μs ~ 100μs

《SON、SCLR(CL)、共用输出(O1、O2)》

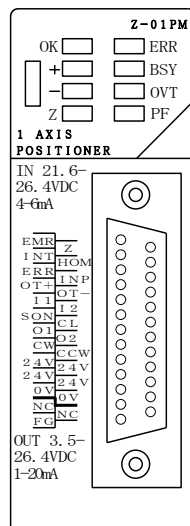
() 内为脉冲表示

输出形式	: 开路集电极
输出电压范围	: DC 3.5 ~ 2.6.4V
最大输出电流	: 2.0mA

- 漏電流 : 0.1 mA 以下
- 残留電压 : 0.6 V 以下
- 峰值吸收器 : 稳压二极管
- 保险丝 : 没有
- 《脉冲输出 CW、CCW》
- 输出形式 : 集电极开路
- 输出電压范围 : DC 21.6 ~ 26.4 V
- 输出電流范围 : 10 ~ 20 mA
- 漏電流 : 0.1 mA 以下
- 残留電压 : 0.3 V 以下
- (-1) 上升時間 : 0.4 μs 以下
- 下降時間 : 0.2 μs 以下
- 峰值吸收器 : 稳压二极管
- 保险丝 : 没有

3. 功能规格

- 控制方式 : 开环控制
- 控制軸数 : 1
- 位置指令范围 : -8388608 ~ 8388607 指令单位
- 位置指令方式 : 绝对、增量
- 座標系 : 以机械原点为原点的座标, 无限长定位座标
- 速度指令单位 : 1 ~ 20000000 指令单位
(但pps换算为 1~400000)
- 超调 : 0 ~ 250% 10% 单位
- 加減速時間 : 10 ~ 99990 ms 10 ms 单位
- 加減速特性 : 直線、S字
- 基准速度 : 0 ~ 20000000 指令单位
- 停止功能 : 紧急停止(内部、外部)、暂停
- 间隙补偿 : 0 ~ 9999 脉冲
- 電子齿轮 : $1/50 \leq G \leq 50$ $G = M/D$ M、D = 1 ~ 9999
- 原点移位量 : -8388608 ~ 8388607 指令单位
- 软限位 : -8388608 ~ 8388607 指令单位 的2点(也可无效)
- 运转方式 : 自动运转、手动运转、点动运转、原点检索
- 原点搜索 : 4种形式
- 定位形式 : 5种形式
- 中断 : 外部 1点、内部 1点
- 静态时间 : 0 ~ 99990 ms 10 ms 单位
- 停電保持功能 : 有
- 输入占有点数 : 16点
- 输出占有点数 : 16点
- (-3) CPU模块 : DL250-1, DL260, SZ-4M
- 可装槽数 : 1~7 (0号槽不可装插。)
- (-1) 外觀 : 如右图所示



4. 基本功能

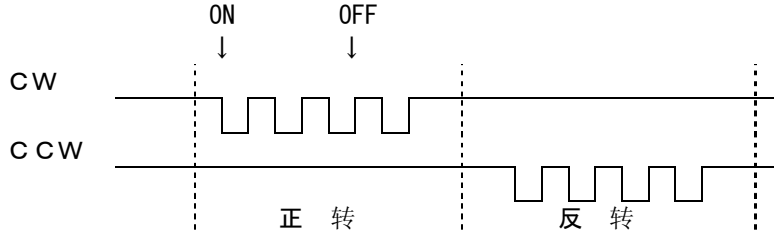
4-1. 脉冲输出

脉冲输出方法，可用参数设定选择以下2种方法：

① A方式（正·反转脉冲）

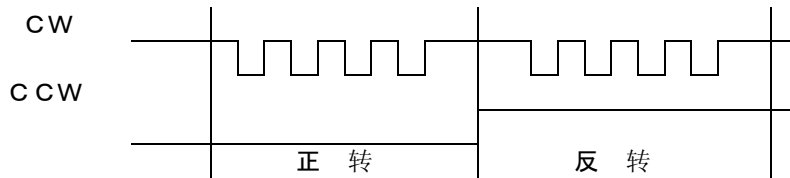
CW输出端子输出正转脉冲、CCW输出端子输出反转脉冲。

※下面的波形是输出端子波形。而且、LED显示（+、-）在CW、CCW的脉冲输出时点亮。



② B方式（方向·脉冲）

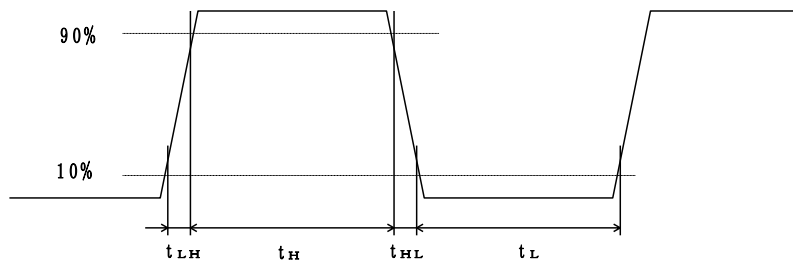
CW输出端子输出脉冲、CCW输出端子输出方向。



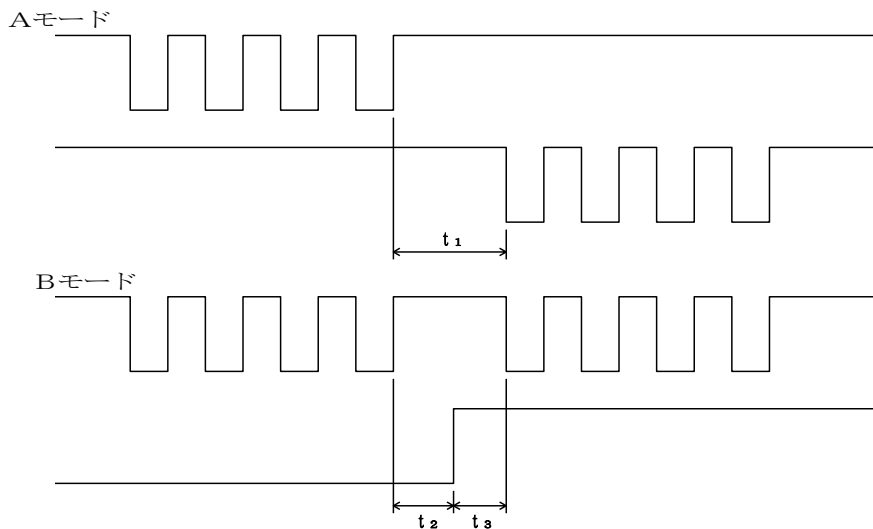
※正转方向为、在Z-01PM内部管理的即时值是增加的回转方向。

反转方向为、在Z-01PM内部管理的即时值是减少的回转方向。

脉冲波形规格



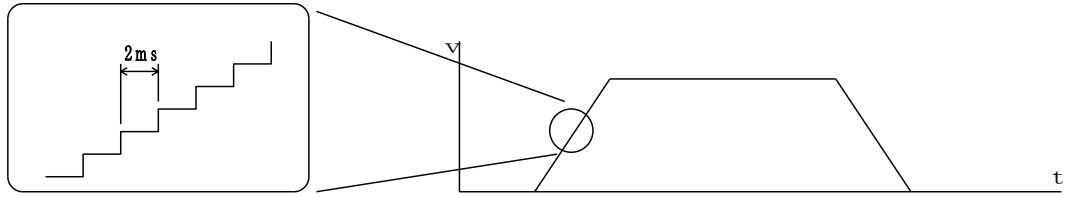
(-1) $t_{LH} \cong 0.4 \mu s$ 、 $t_{HL} \cong 0.2 \mu s$ $T_L : T_H \cong 1 : 1$



$t_1、t_2、t_3 \cong 10 \mu s$

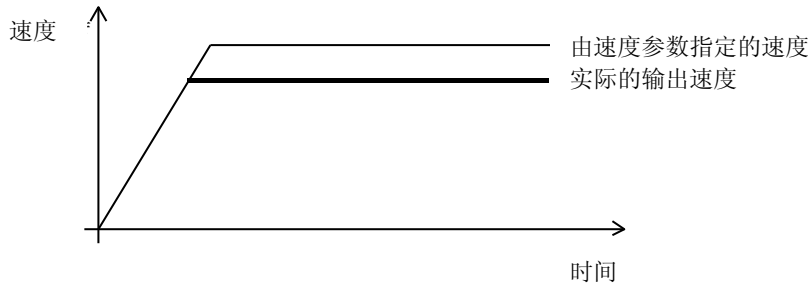
脉冲频率的控制时间

Z-01PM的输出脉冲频率、以2ms为单位进行控制。



输出脉冲频率，由设定的速度参数

但是、由速度参数设定的速度、与实际输出的脉冲频率有时不一致。这种场合，控制情况如下。



(-3)

输出脉冲的分辩率，大致为速度极限值的1/4096。

4-2. 座标系

Z-01PM、由机械原点为0的座标系进行位置管理。

机械原点用原点搜索来决定。

机械原点每进行一次原点搜索发生变化。只要不进行再次原点搜索，就不会改变。

对于定尺寸定位等无限长定位也能对应。

座标系	机械原点为0	无限长定位
电源投入时	现在位置=0	
座标 (指令单位)		
原点检索起动	(不使用输入传感器时，因不能停止动作，所以要用将原点检索起动继电器OFF，或者其他什么手段进行紧急停止处理)	
原点检索完了时	现在位置=0	
软极限	有效	无效

4-3. 电子齿轮

依靠使用电子齿轮可以进行不是脉冲单位的编程，而可用mm等指令单位进行编程。

电子齿轮位置量和速度指令量都有影响。

由电子齿轮设定相对于Z-01PM输出脉冲数(电子齿轮M)的机械系统移动量(电子齿轮D)的关系。

$$(\text{输出脉冲数}) = (\text{指令量}) \times (M / D) \quad \text{但} \quad 1 / 50 \leq M / D \leq 50$$

指令单位为编程用的单位，利用电子齿轮可变换实际的输出脉冲数。

若设定为M=D时，则电子齿轮无效。

4-4. 送进速度

送进速度、由电子齿轮决定如下。

$$F' = (\text{指令速度}) \times (M / D) \quad \underline{F' \text{ 要设定在 } 400 \text{ KHz 以下。}}$$

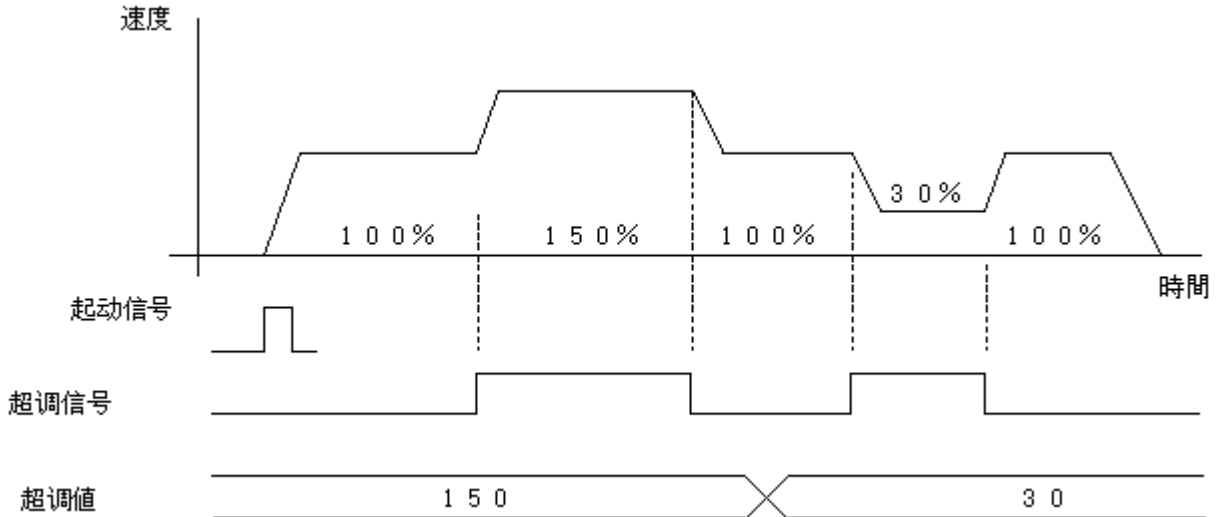
指令速度为単位時間的移动量。

4-5. 超调

超调是改变运转中的送进速度的功能，相对于设定的送进速度，能在0~250%以1%为单位进行变速。

$$(\text{送进速度}) = (\text{送进速度设定值}) \times (\text{超调设定值}) \times 0.01$$

0%则动作停止(BUSY信号保持ON)，其他值的送进速度变更如下。

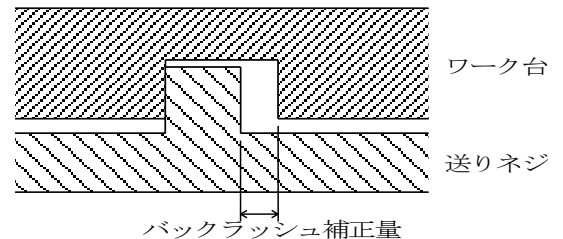


超调在超调信号ON期间实行。
超调能用自动、手动、点动运转（原点搜索时不能使用）。

4-6. 间隙补偿

这是补偿机械系统产生的间隙的功能。
轴的移动方向每次反转时，多输出补偿间隙部分的脉冲。

- 间隙补偿量、在0~9999脉冲的范围内，在参数里设定。
- 补偿在轴的移动方向每次反转时进行，多输出在参数里设定的脉冲数。
- 间隙补偿为机械补偿，因此即时位置不显示。



间隙补偿在所有的运动状态下进行。

4-7. 行程监控

是监控机械可动范围的功能，有机械式和软式。

(-3) 1) 机械式（行程末端限位OVT）

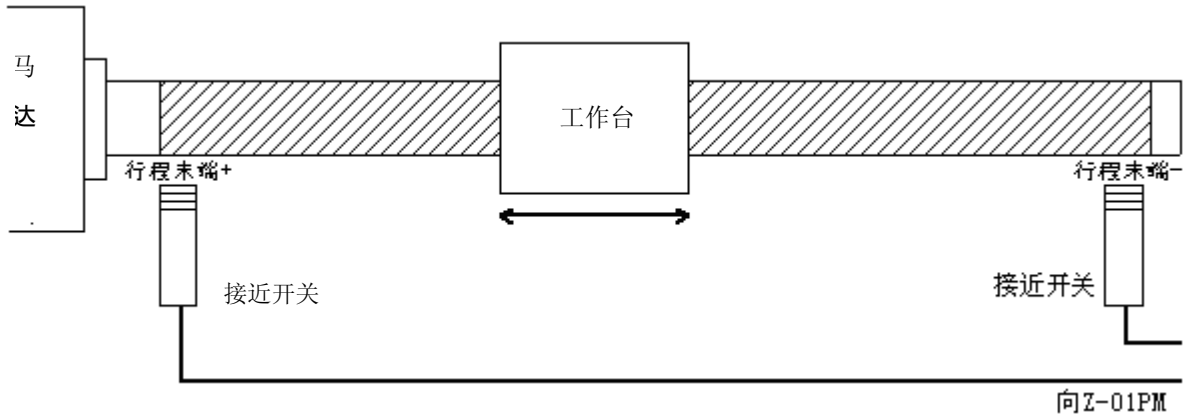
在机械的可动末端设置接近开关或限位开关、由这些开关来监视可动末端。
开关检出的信号通过Z-01PM的输入输出连接器输入到内部。
开关的输入触点，可利用参数选择a或b触点的任何一个。

Z-01PM如果检测到开关ON（b触点时为OFF）的信号时，则认为可动部分已到达可动末端，就使动作停止。

在自动、点动运转时如有此输入，则认为出错，就紧急停止。

手动运转时，用该输入减速停止运转。与手动运转相反侧的OVT有输入时，则成为紧急停止出错。

在相反侧的OVT输入状态下，可进行手动运转（在解除OVT输入之前，不检出出错）。



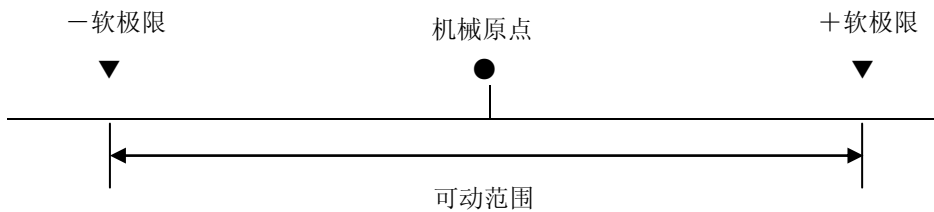
2) 软式 (软极限)

是监视机械可动部分即时位置的功能。

预先用座标值将可动范围设定在参数里, 在超过其设定范围时, 减速停止。

在预先设定移动量 (点动运转等) 的动作中, 若移动量超过其设定范围时, 发生出错 (不动作)。

软极限设定值两端均为「0」时为无效。



软极限设定值, 座标的上限值、下限值要设定得留有余地, 以发生遮盖。

所谓遮盖, 为座标值越过上限(下限)值, 成为下限(上限)值的情况。

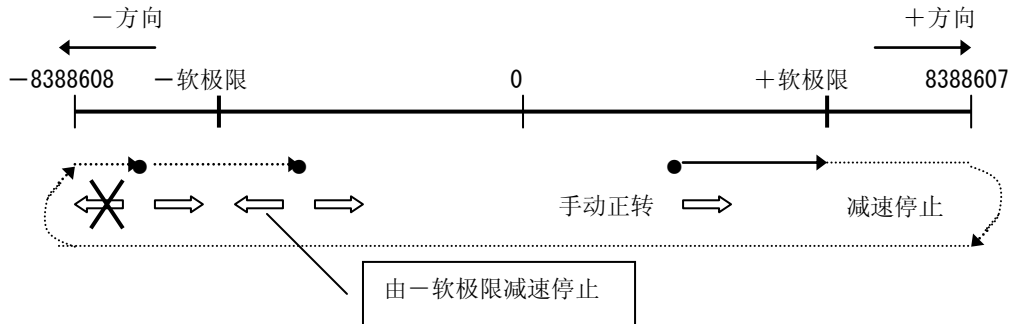
例如: 在正转方向移动中, 检出软极限(+)(正转限位继电器ON)而减速停止, 因遮盖, 现在位置成为(-)值的场合,

现在位置还在软极限(-)范围外时: 手动反转将无动作(不能回到原来的位置), 手动正转可动作(反转极限继电器ON状态)。

现在位置还在软极限(-)范围内时: 手动正转、手动反转都动作(在动作过程中, 软极限也有效)。

遮盖后的点动动作无效(会成为系统出错)。

而且, 即使在反转方向移动中, 正转/反转也相反, 但原理相同。



4-8. 加减速功能

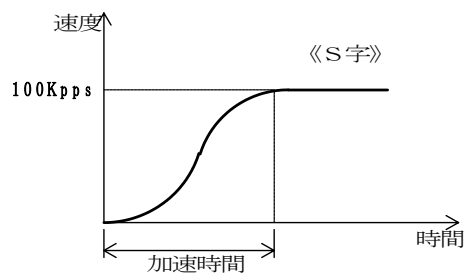
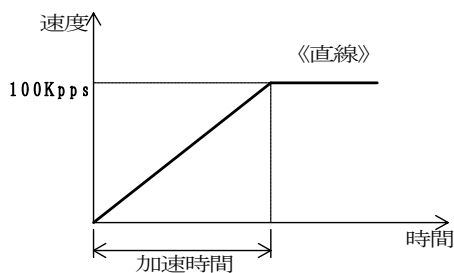
Z-01PM对所有定位都自动进行加减速。

加减速曲线可选择如下2种。

- 直线加减速
- S字加减速

加速时间定义为速度 0 → 100 Kpps 之间的升速时间。

减速时间定义为速度 100 Kpps → 0 之间的减速时间。

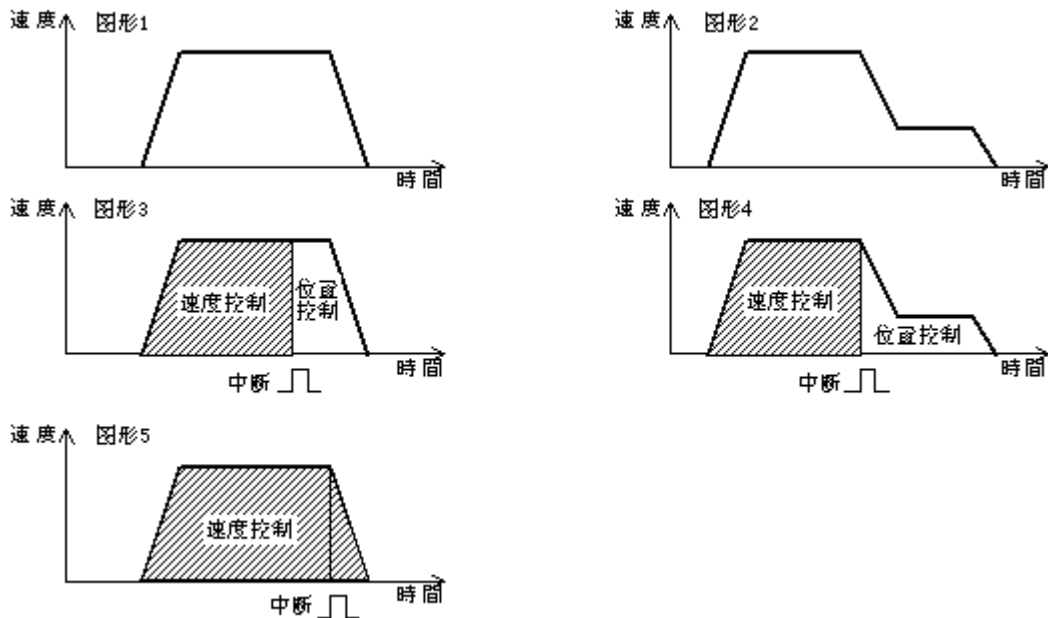


4-9. 原点搜索

Z-01PM有4种原点搜索方法。
可利用参数设定来选择方法。
原点搜索的详情，请参阅 6-3. 原点搜索。

4-10. 定位图形

Z-01PM有5种基本的定位图形。



在Z-01PM上，将这些基本图形进行组合能实现各种各样的动作。
定位动作的详情，请参阅 1-2. 程序。

4-11. 停止功能

在Z-01PM上，由于某种原因必须停止定位动作时，有停止功能。
停止功能有暂停和紧急停止。

1) 暂停

只有自动运转可以将动作暂时停止。
来自PLC的暂停ON时，PAUSE命令（PM内程序）实行后，减速停止，成为暂停。
在暂停中是「BUSY」状态。
在暂停中如果自动运转起动ON，则从暂停的地方重新开始动转。
此外，在暂停中如果工序复位ON，则可将暂停以后的工序取消。
在暂停中，其他运转（手动、点动、检索）不能起动。

2) 紧急停止

成为以下状态时，将会紧急停止。

- BUSY中、ENABLE从ON→OFF；
 - 外部紧急停止信号（EMR）ON；
 - 伺服出错信号（ERR）ON；
 - 系统出错；
 - 外部24V降低；
- (-3) • 运转过程中、正/反转限位（OVT、软极限）输入ON；
紧急停止时，Z-01PM按紧急停止减速时间减速停止，将伺服ON信号OFF。

4-12. 停电保持功能

Z-01PM内部能够将各种参数、程序存储在EEPROM里。
能够存储系统参数、原点参数、特殊参数各1组，数据寄存器32个、
程序10个（每个CNC语言120步）。
进行存储时、要用命令写入EEPROM（参阅9. 命令）。

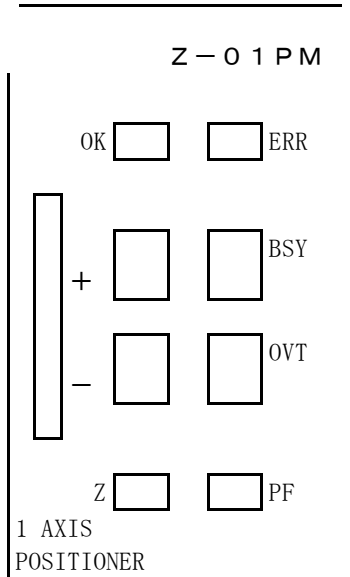
4-1-3. 文件管理功能

在Z-01PM的程序内，可以文件形式存储注解。
请参阅Z-01PM 文件更新工具。

4-1-4. 显示功能

Z-01PM用8点LED显示信息。

(-1)



OK：在Z-01PM可以动作的状态时点亮。

＋：在正转脉冲输出时点亮。

－：在反转脉冲输出时点亮。

Z：在Z相信号输入时点亮。

ERR：发生错误时点亮。（Error）

BSY：BUSY状态时点亮。（Busy）

OVT：正 / 反转限位（包括软极限）输入时点亮。（超程）

PF：外部电源的电压降低时点亮。（Power fail）

(-3)

4-15. 外部输出端子

U-01PM上的D-Sub 25脚接插件有以下功能的输出端子。
关于插脚的配置，请参阅「13. 配线」章。

4-15-1. 输入端子

信号名	内容	初期值	触点变更	共用输入	插脚号
EMR	紧急停止输入	b 触点	○	○(a触点)	1
INT	外部中断输入	a 触点	×	×	2
ERR	出错输入	b 触点	○	○(a触点)	3
CWL	正转极限输入	b 触点	○	○(a触点)	4
CCWL	反转极限输入	b 触点	○	○(a触点)	1 7
Z	Z 相输入	a 触点	×	×	1 4
HOME	原点找准输入	a 触点	○	○(a触点)	1 5
INP	到位输入	a 触点	○	○(a触点)	1 6
IN1	共用输入 1	a 触点	×	—	5
IN2	共用输入 2	a 触点	×	—	1 8

- 紧急停止输入（EMR）
让Z-01PM紧急停止时输入。
初期值为b触点，可以用参数变更为a触点及共用输入。
- 外部中断输入（INT）
在Z-01PM运输过程中要中断时输入。
该输入固定为a触点，不可变更为共用输入。
- 出错输入（ERR）
这是用来连接来自马达驱动器的错误输出的输入端子。
出错输入时的动作与紧急停止输入同。
初期值为b触点，可以用参数变更为a触点及共用输入。
- 正反转限位输入（CWL、CCWL）
是行程末端限位（OVT+、OVT-）检出用的输入端子。
其中任何一个输入后，作为出错而停止。
初期值为b触点，可以用参数变更为a触点及共用输入。
- Z相输入（Z）
是原点搜索动作时的原点检出用输入端子。
该输入固定为a触点，不可变更为共用输入。
- 原点找准输入（HOME）
是原点搜索动作时静止位检出用输入端子。
初期值为a触点，可以用参数变更为b触点及共用输入。
- 到位输入（INP）
是用来检出来自外部的到位信号的输入端子。
运转停止后，如输入该信号，则Z-01PM定位完毕。
初期值为a触点，可以用参数变更为b触点及共用输入。
- 共用输入（IN1，IN2）
是通用的输入端子。
在自动运转程序中，可以用于条件判断等。
共用输入1、2都固定为a触点。

4-15-2. 输出端子

※输出端子全部是开路集电极输出，输出时三极管ON。

信号名	内容	共用输出	脚号
S O N	伺服ON输出	○	6
S C L	偏差复位输出	○	19
C W	正转脉冲输出	×	8
C C W	反转脉冲输出	×	21
O U T 1	通用出力1	—	7
O U T 2	通用出力2	—	20

- • 伺服ON输出（S O N）
是表示Z-01PM正在控制马达的输出。
E N A B L E信号ON后、系统检查结果正常时输出。
可用参数变更为共用输出。
- • 偏差复位输出
是将马达的偏差复位的输出。
与伺服ON输出同时输出，只ON50ms。
此外，原点搜索完毕后ON50ms。
可用参数变更为共用输出。
- • 正转 / 反转脉冲输出（C W, C C W）
是马达动作脉冲输出端子。详情请参阅「4-1. 脉冲输出」。
- • 通用输出（O U T 1, O U T 2）
是通用的输出端子。
可由自动运转程序ON / O F F。

5. 输出继电器

从 PLC 来看，Z-01PM 占有输入 16 点、输出 16 点，共计 32 点输出继电器。（参照下表）
PLC 侧的程序里，通过该输出继电器的 ON 来操作 Z-01PM，能够监视输入继电器在，
掌握运转状态。

另外，在 Z-01PM 内部也有输出继电器，但该继电器只能由自动运转程序来操作。

5-1. PLC 输出继电器

(-1) 注) Q0~Q17 要保 3 次扫描以上 ON 或 OFF。

「1 次扫描：ON」的那种高速输出时，PM 有可能输出会失败。

输入继电器编号	名称	输出继电器编号	名称
I n + 0 0	OK	Q n + 0 0	ENABLE
I n + 0 1	READY	Q n + 0 1	原点搜索起动
I n + 0 2	BUSY	Q n + 0 2	自动运转起动
I n + 0 3	系统出错	Q n + 0 3	手动正转起动
I n + 0 4	数据出错	Q n + 0 4	手动反转起动
(-1) I n + 0 5	CW 继电器	Q n + 0 5	点动正转起动
(-1) I n + 0 6	CCW 继电器	Q n + 0 6	点动反转起动
I n + 0 7	正转限位 (OVT+)	Q n + 0 7	暂停
I n + 1 0	反转限位 (OVT-)	Q n + 1 0	超调
I n + 1 1	原点搜索完毕	Q n + 1 1	出错复位
I n + 1 2	定位完毕	Q n + 1 2	工序复位
I n + 1 3	暂停中	Q n + 1 3	内部中断
I n + 1 4	无进给时间	Q n + 1 4	辅助码清除
I n + 1 5	辅助码输出	Q n + 1 5	(予约)
I n + 1 6	(予约)	Q n + 1 6	(予约)
I n + 1 7	(予约)	Q n + 1 7	(予约)

5-2. PLC 输入继电器 (Z-01PM→PLC)

- OK (I n + 0 0)
Z-01PM 电源投入后进行自诊断，判断为系统正常时该继电器 ON。
判断为系统异常时，该继电器不 ON。
而且，异常时由 LED 显示来告知外部。
- READY (I n + 0 1)
Z-01PM 在 ENABLE 信号 ON 后，即进行以下内容的检查。
! 外部紧急停止信号 (EMR) 输入状态
! 伺服出错信号 (ERR) 输入状态
! 外部 24V 输入状态
! 检查的结果判断为正常时，伺服 ON 输出即 ON，1 秒后该继电器 ON。
! 检查的结果判断为异常时，由 LED 告知外部，异常内容由出错状态输出 (参阅 15. 出错码)。

- B U S Y (I n + 0 2)
Z - 0 1 P M在以下状态时，该继电器 O N。
- (-3) 自动运转中
手动运转中
点动运转中
原点搜索中
自动运转中是从自动运转起动 O N起到检出定位程序的 E N D 为止的时间。
- 系统出错 (I n + 0 3)
Z - 0 1 P M由于某种原因不能动作时 O N。
具体的原因如下。
 - • B U S Y中、E N A B L E由 O N → O F F
 - • 外部紧急停止信号 (E M R) O N
 - • 伺服出错信号 (E R R) O N
 - • 系统出错
 - • 外部 2 4 V降低
 - • 自动运转中、正 / 反转限位 (O V T +、-) 输入 O N，
 该继电器一旦 O N后，如果不消除出错原因，出错复位不 O N，则该继电器不能 O F F。
出错内容由出错状态 (参阅 1 5 . 出错码) 输出。
- 数据出错 (I n + 0 4)
接收到来自 P L C或外围装置的数据中有错误时 O N。
如果出错复位不 O N，由不 O F F。
出错内容在出错状态 (参阅 1 5 . 出错码) 里输出。
- (-1) • C W继电器 (I n + 0 5)
C W方向脉冲输出时 O N。
- (-1) • C C W继电器 (I n + 0 6)
C C W方向脉冲输出时 O N。
- 正转限位 (I n + 0 7)
正转方向的限位输入 (O V T +) O N时，或超过正转方向的软极限时，该继电器 O N。
即时值在软极限范围内，正转限位输入是 O F F时，该继电器 O F F。
- 反转限位 (I n + 1 0)
逆转方向的限位输入 (O V T -) O N时，或超过反转方向的软极限时，该继电器 O N。
即时值在软极限范围内，反转限位输入是 O F F时，该继电器 O F F。
- 原点搜索完毕 (I n + 1 1)
Z - 0 1 P M进行原搜索，完毕时 O N。
原点搜索完毕后，在下次原点检索起动时 O F F。
- 定位完毕 (I n + 1 2)
Z - 0 1 P M脉冲输出完毕，而且到位信号输入时 O N。
O N状态保持到有下次脉冲输出。
- 暂停 (I n + 1 3)
Z - 0 1 P M在自动运转中暂停指令 O N时，定位动作停止，该继电器 O N。
- 无进给时间中 (I n + 1 4)
无进给时间计时期間 O N。
无进给时间一到即 O F F。

- 辅助码输出（ I_{n+15} ）
从Z-01PM向辅助码领域输出号码时ON。
利用辅助码清除ON来OFF。

5-3. PLC输出继电器（PLC→Z-01PM）

- ENABLER（ Q_{n+00} ）
是使Z-01PM成为可动作状态的继电器。
在OK信号呈ON的状态下，该继电器ON后，Z-01PM进行检查，如果正常，READY则ON。
该继电器一旦ON后，要保持常时ON状态。
在BUSY中如果该继电器OFF，则Z-01PM判断为系统异常，作紧急停止处理。（电平检出）
- 原点搜索起动（ Q_{n+01} ）
是使原点搜索起动的继电器。
该继电器ON后，即开始原点搜索。
(-3) 在原点搜索进程中，如果该继电器OFF，则原点搜索停止。（电平检出）
- 自动运转起动（ Q_{n+02} ）
是使自动运转起动的继电器。
该继电器ON后，按定位程序开始自动运转。
在自动运转过程中即使该继电器OFF，自动运转也继续进行。（边沿检出）
- 手动正转起动（ Q_{n+03} ）
该继电器ON后，开始手动运转，向正转方向移动。
OFF后即动作停止。（电平检出）
- 手动反转起动（ Q_{n+04} ）
该继电器ON后，开始手动运转，向反转方向移动。
OFF后即动作停止。（电平检出）
- 点动正转起动（ Q_{n+05} ）
该继电器ON后，开始点动运转，向正转方向移动。
在点动运转时，即使该继电器OFF，点动运转也继续进行。（边沿检出）
- 点动反转起动（ Q_{n+06} ）
该继电器ON后，开始点动运转，向反转方向移动。
在点动运转时，即使该继电器OFF，点动运转也继续进行。（边沿检出）
- 暂停（ Q_{n+07} ）
在自动运转过程中，该继电器ON后，即减速停止。
该继电器OFF后，如将自动运转起动ON，则自动运转将从暂停的地方再开始。
（边沿检出）
- 超调（ Q_{n+10} ）
该继电器ON后，进行速度超调。
原点搜索时不能使用。该继电器OFF后即解除。（电平检出）
- 出错复位（ Q_{n+11} ）
是为了清除SYSTEM ERROR 或 DATA ERROR的信号。
该继电器ON后，即清除这些出错，也清除出错状态。（边沿检出）
- 工序复位（ Q_{n+12} ）
在暂停时该继电器ON后，将程序运转状态复位，回到空载状态。
如果再次将自动运转起动继电器「ON」，则程序开头开始定位。
（边沿检出）

- 内部中断（ Q_{n+13} ）
该继电器ON后，发生中断。
（边沿检出）
- 辅助码清除（ Q_{n+14} ）
该继电器ON后，辅助码输出OFF。（边沿检出）

5-4. PLC输出继电器的配合

有多个输出继电器同时「ON」，从PLC输出进行转送时，根据下面所示的优先顺序处理。

。

《起动信号》

高	Q_{n+01} ：原点搜索起动
↓	Q_{n+03} ：手动正转起动
	Q_{n+04} ：手动反转起动
	Q_{n+05} ：点动正转起动
↓	Q_{n+06} ：点动反转起动
低	Q_{n+02} ：自动运转起动

《控制信号》

高	Q_{n+07} ：暂停
↓	Q_{n+13} ：内部中断
低	Q_{n+10} ：超调

5-5. Z-01PM的内部输出继电器

Z-01PM的内部输出继电器，如下表所示。I 0 ~ I 17、Q 0 ~ Q 17 与分配在PLC上的继电器相同。

在PLC程序里，继电器只能使用I 0 ~ I 17、Q 0 ~ Q 17。

I 20 ~ I 27、Q 20 ~ 27，在自动运转程序中，可以进行条件判断或强制ON / OFF。此外，由于不使用请求，可以进行高速应答。

(-1) 注) Q 0 ~ Q 17 要保持3次以上扫描 ON或OFF。

像「1次扫描: ON」那样的高速输出时，PM有可能输出失败。

输入继电器号	名称	输出继电器号	名称
I 0	OK	Q 0	ENABLE
I 1	READY	Q 1	原点搜索起动
I 2	BUSY	Q 2	自动运转起动
I 3	系统出错	Q 3	手动正转起动
I 4	数据出错	Q 4	手动反转起动
(-1) I 5	CW继电器	Q 5	点动正转起动
(-1) I 6	CCW继电器	Q 6	点动反转起动
I 7	正转限位	Q 7	暂停
I 10	反转限位	Q 10	超调
I 11	原点搜索完毕	Q 11	出错复位
I 12	定位完毕	Q 12	工序复位
I 13	暂停中	Q 13	内部中断
I 14	无进给时间	Q 14	辅助码清除
I 15	辅助码输出	Q 15	(予约)
I 16	(予约)	Q 16	(予约)
I 17	(予约)	Q 17	(予约)
I 20	正转限位输入(OVT+)	Q 20	伺服ON输出
I 21	反转限位输入(OVT-)	Q 21	偏差复位输出
I 22	HOME输入	Q 22	通用输出1
I 23	外部紧急停止输入	Q 23	通用输出2
I 24	ERR输入	Q 24	(予约)
I 25	到位输入	Q 25	(予约)
I 26	通用输入1	Q 26	(予约)
I 27	通用输入2	Q 27	(予约)

注) I 20 ~ I 27 表示Z-01PM 外部输入状态。

有关输入继电器的ON / OFF参阅下页。

- 正转限位输入 (I 20)
表示正转限位输入 (OVT+) 的状态。
- 反转限位输入 (I 21)
表示反转限位输入 (OVT-) 的状态。

- HOME输入（I 2 2）
表示HOME输入的状态。
- 外部紧急停止（EMR）输入（I 2 3）
表示外部紧急停止输入端子的状态。
- ERR输入（I 2 4）
表示ERR输入端子的状态。
- 到位输入（I 2 5）
表示到位输入端子的状态。
- 通用输入1（I 2 6）
表示通用输入1端子的状态。
- 通用输入2（I 2 7）
表示通用输入2端子的状态。

关于 I 2 0 ~ I 2 7 输入状态

在系统参数（输入输出数据1 参阅P 3 8）的设定里，分别有「A触点」、「B触点」的设定。

A触点的时候： 外部 ON → 内部继电器ON
外部 OFF → 内部继电器OFF

B触点的时候： 外部 OFF → 内部继电器ON
外部 ON → 内部继电器OFF

- 伺服ON输出（Q 2 0）
表示伺服ON输出端子的状态。
输出时ON，不输出时OFF。
由系统参数设定为通用输出时，可以进行强制ON / OFF。
由系统参数设定为予約时，则常时OFF。
- 偏差复位输出（Q 2 1）
表示偏差复位输出端子的状态。
输出时ON，不输出是时OFF。
由系统参数设定为通用输出时，可以进行强制ON / OFF。
由系统参数设定为予約时，则常时OFF。
- 通用输出1（Q 2 2）
表示通用输出1端子的状态。
输出时ON，不输出是时OFF。
也可以进行强制ON / OFF。
由系统参数设定为予約时，则常时OFF。
- 通用输出2（Q 2 3）
表示通用输出2端子的状态。
输出时ON，不输出是时OFF。
也可以进行强制ON / OFF。
由系统参数设定为予約时，则常时OFF。
- Q 2 4 ~ Q 2 7
预留

6. 运转功能

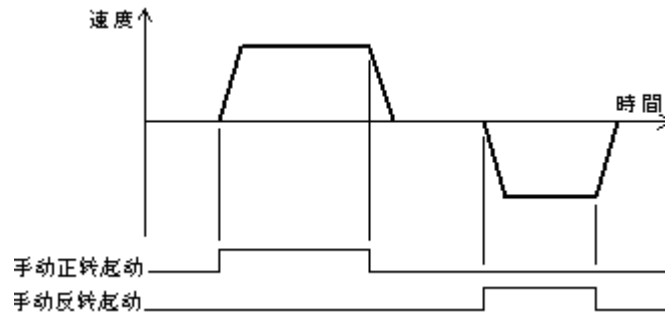
Z-01PM有4种运转方式。

6-1. 手动运转

手动运转通过手动正转 / 反转起动 ON 来起动。

手动正转起动 ON 后，向 + 方向、手动反转起动 ON 后，向 - 方向移动。

手动运转，在手动正转 / 反转起动 ON 的时间内动作。（如 OFF 后，即减速停止）



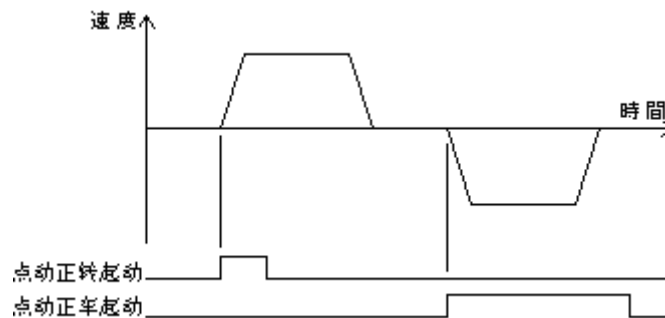
6-2. 点动运转

点动运转通过点动正转 / 反转起动 ON 来起动。

点动正转起动 ON 后，向 + 方向、手动反转起动 ON 后，向 - 方向移动。

点动运转，只要将点动正转 / 反转起动 OFF → ON 操作即移动指定的量。

（在移动过程中即使操作 OFF，中途也不停止）



6-3. 原点搜索

是用来决定各台机械上规定的机械原点的功能。

电源投入后，一定要实行1次。

原点搜索通过原点搜索起动 ON 来起动。

在 origin search execution process 中将原点检索起动 OFF，则减速停止，终止原点检索。

(-1) 原点搜索起动时的原点搜索方式、动作方向、各种速度由原点参数（参照 P 10-2）设定。

6-3-1. 原点搜索方法

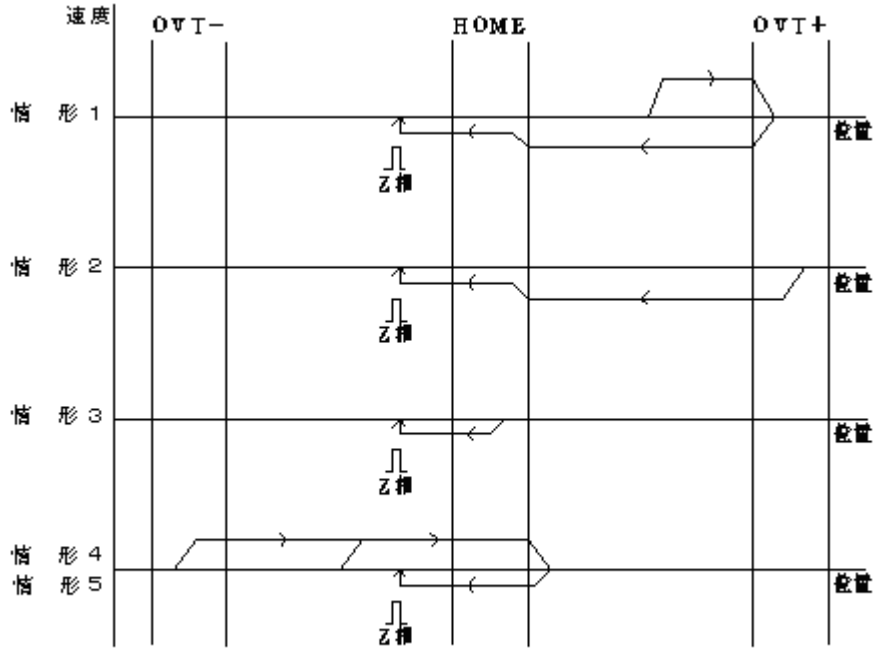
原点搜索方法有 4 种。

1) 方式 0

电气原点在机械的中央附近时使用。

《动作说明》

起动方向：+ 方向、 Z 相方向：- 方向



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动 ON 后，向 + 方向以搜索速度移动。
- ②OVT+ 信号 ON 后即减速，向 - 方向反转。
- ③HOME 信号 ON 后，减速到蠕动速度。
- ④HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

情形 2：原点搜索开始点在 [OVT+] 上的情形。

- ①原点搜索起动 ON 后，向 - 方向以搜索速度移动。
- ②HOME 信号 ON 后，减速到蠕动速度。
- ③HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

情形 3：原点搜索开始点在 [HOME] 上的情形。

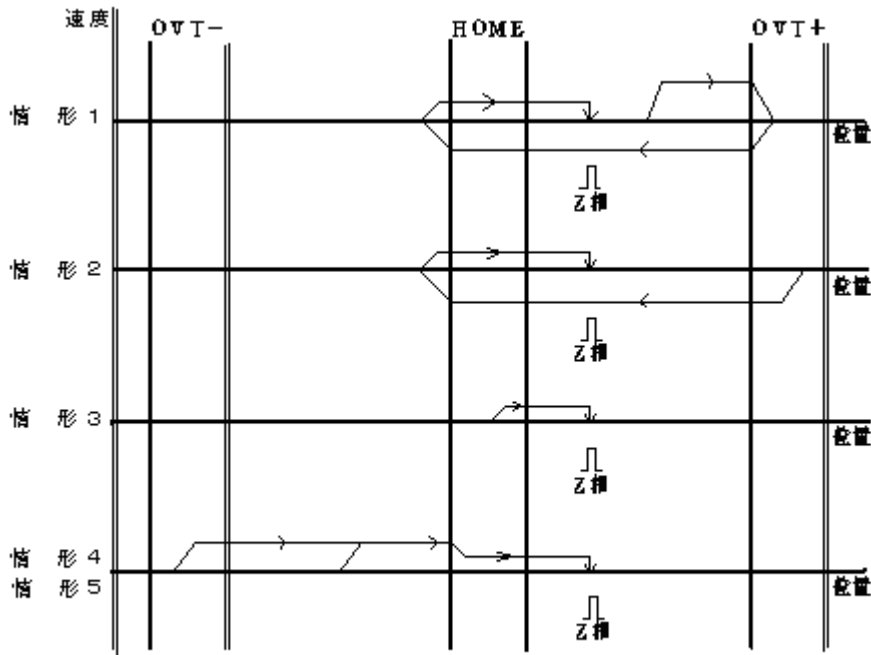
- ①原点搜索起动 ON 后，向 - 方向以蠕动速度移动。
- ②HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

情形 4：原点搜索开始点在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点在 [OVT-] 上的情形

- ①原点搜索起动 ON 后，向 + 方向以搜索速度移动。
- ②HOME 信号由 ON → OFF 后即减速，向 - 方向反转。
- ③HOME 信号 ON 后，减速到蠕动速度。
- ④HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

起动方向：+方向、 Z相方向：+方向



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动 ON 后，向+方向以搜索速度移动。
- ②OVT+ 信号 ON 后即减速，向-方向反转。
- ③HOME 信号 ON→OFF 后减速，向+方向以蠕动速度反转。
- ④HOME 信号由 ON→OFF 后，Z 相信号由 OFF→ON，即原点搜索完毕。

情形 2：原点搜索开始点在 [OVT+] 上的情形。

- ①原点搜索起动 ON 后，向-方向以搜索速度移动。
- ②HOME 信号 ON→OFF 后减速，向+方向以蠕动速度反转。
- ③HOME 信号由 ON→OFF 后，Z 相信号由 OFF→ON，即原点搜索完毕。

情形 3：原点搜索开始点在 [HOME] 上的情形。

- ①原点搜索起动 ON 后，向-方向以蠕动速度移动。
- ②HOME 信号由 ON→OFF 后，Z 相信号由 OFF→ON，即原点搜索完毕。

情形 4：原点搜索开始点在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点在 [OVT-] 上的情形

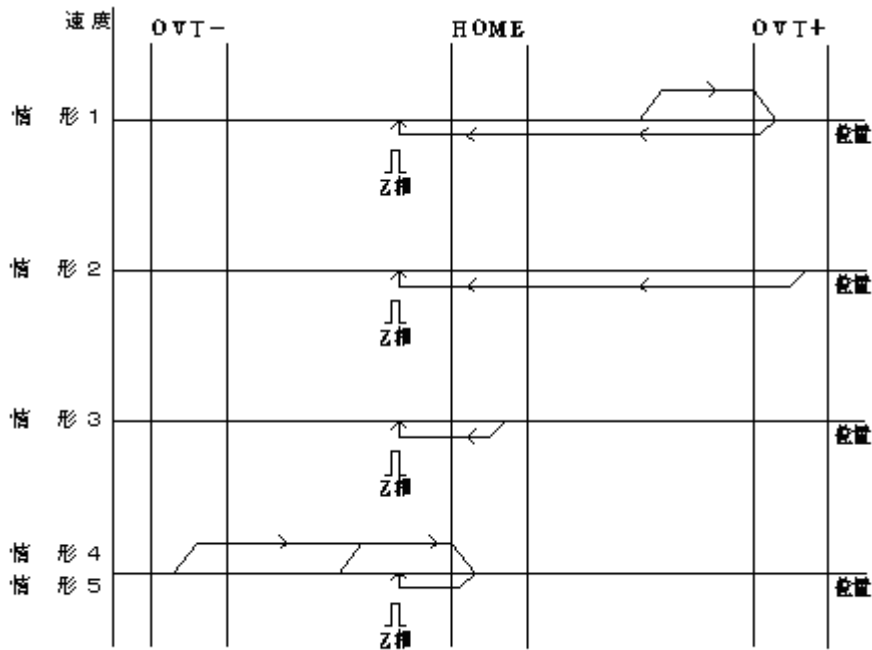
- ①原点搜索起动 ON 后，向+方向以搜索速度移动。
- ②HOME 信号 ON 后，减速到蠕动速度。
- ③HOME 信号由 ON→OFF 后，Z 相信号由 OFF→ON，即原点搜索完毕。

2) 方式 1

电气原点在机械的两端附近时使用。

《动作说明》

起动方向：+方向、 Z相方向：-方向



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动 ON 后，向+方向以搜索速度移动。
- ②OVT+信号 ON 后即减速，向-方向反转，以蠕动速度移动。
- ③HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

情形 2：原点搜索开始点，在 [OVT+] 上的情形。

- ①原点搜索起动 ON 后，向-方向以蠕动速度移动。
- ②HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

情形 3：原点搜索开始点，在 [HOME] 上的情形。

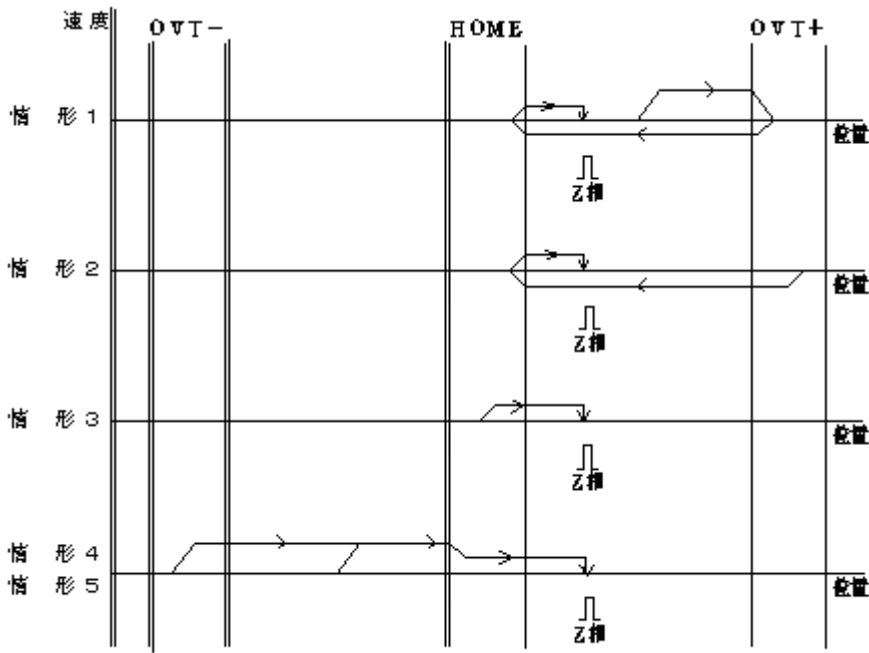
- ①原点搜索起动 ON 后，向-方向以蠕动速度移动。
- ②HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

情形 4：原点搜索开始点，在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点，在 [OVT-] 上的情形

- ①原点搜索起动 ON 后，向+方向以搜索速度移动。
- ②HOME信号 ON 后即减速，向-方向反转，以蠕动速度移动。
- ③HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

起动方向：+方向、 Z相方向：+方向



情形1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动ON后，向+方向以搜索速度移动。
- ②OVT+信号ON后即减速，向-方向反转，以蠕动速度移动。
- ③HOME信号ON后即减速，向+方向反转。
- ④HOME信号由ON→OFF后，Z相信号由OFF→ON，即原点搜索完毕。

情形2：原点搜索开始点，在 [OVT+] 上的情形。

- ①原点搜索起动ON后，向-方向以蠕动速度移动。
- ②HOME信号ON后即减速，向+方向反转。
- ③HOME信号由ON→OFF后，Z相信号由OFF→ON，即原点搜索完毕。

情形3：原点搜索开始点，在 [HOME] 上的情形。

- ①原点搜索起动ON后，向+方向以蠕动速度移动。
- ②HOME信号由ON→OFF后，Z相信号由OFF→ON，即原点搜索完毕。

情形4：原点搜索开始点，在 [OVT-] 与 [HOME] 之间的情形

情形5：原点搜索开始点，在 [OVT-] 上的情形

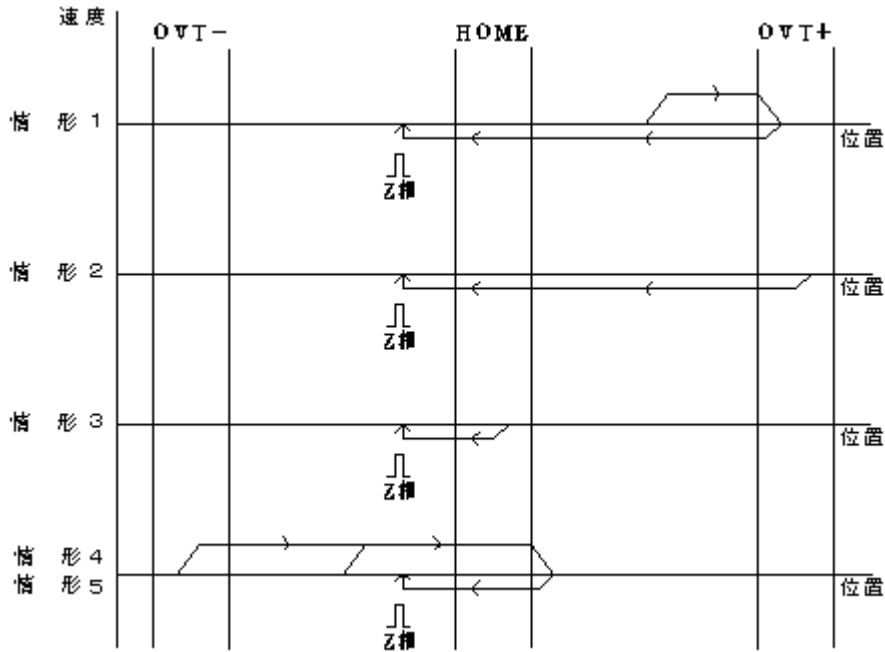
- ①原点搜索起动ON后，向+方向以搜索速度移动。
- ②HOME信号ON后即减速为蠕动速度。
- ③HOME信号由ON→OFF后，Z相信号由OFF→ON，即原点搜索完毕。

3) 方式 2

电气原点在机械的两端附近时使用。

《动作说明》

起动方向：+方向、 Z相方向：-方向



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动 ON 后，向+方向以搜索速度移动。
- ②OVT+信号 ON 后即减速，向-方向反转，以蠕动速度移动。
- ③HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

情形 2：原点搜索开始点，在 [OVT+] 上的情形。

- ①原点搜索起动 ON 后，向-方向以蠕动速度移动。
- ②HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

情形 3：原点搜索开始点，在 [HOME] 上的情形。

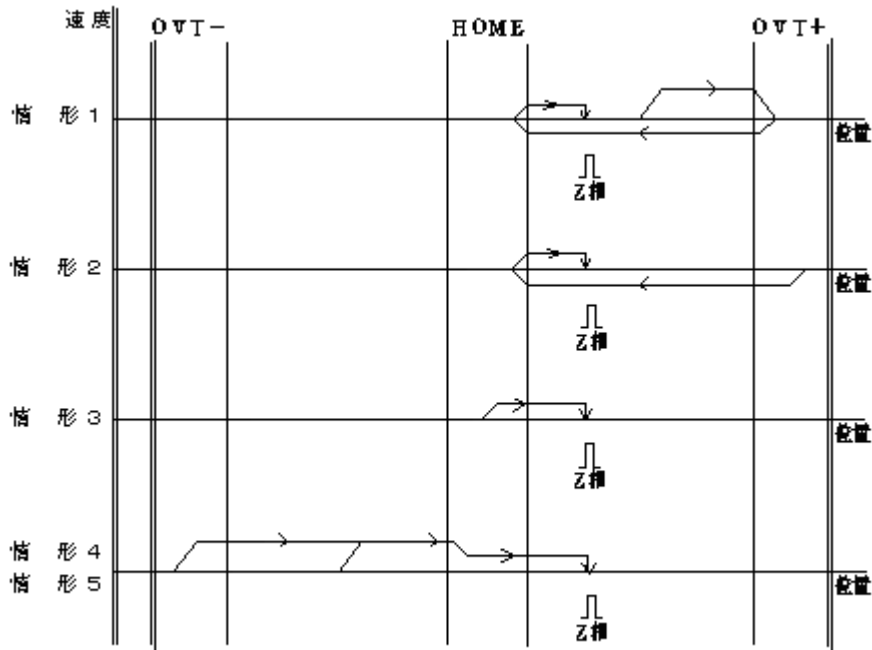
- ①原点搜索起动 ON 后，向-方向以蠕动速度移动。
- ②HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

情形 4：原点搜索开始点，在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点，在 [OVT-] 上的情形

- ①原点搜索起动 ON 后，向+方向以搜索速度移动。
- ②HOME信号 ON 后即减速，向-方向反转，以蠕动速度移动。
- ③HOME信号由 ON→OFF 后，Z相信号由 OFF→ON，即原点搜索完毕。

注) 与方式1的不同处，为情形4、5的HOME信号输入时的处理不同。



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动 ON 后，向 + 方向以搜索速度移动。
- ②OVT+ 信号 ON 后即减速，向 - 方向反转，以蠕动速度移动。
- ③HOME 信号 ON 后即减速，向 + 方向反转。
- ④HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

情形 2：原点搜索开始点，在 [OVT+] 上的情形。

- ①原点搜索起动 ON 后，向 - 方向以蠕动速度移动。
- ②HOME 信号 ON 后即减速，向 + 方向反转。
- ③HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

情形 3：原点搜索开始点，在 [HOME] 上的情形。

- ①原点搜索起动 ON 后，向 + 方向以蠕动速度移动。
- ②HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

情形 4：原点搜索开始点，在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点，在 [OVT-] 上的情形

- ①原点搜索起动 ON 后，向 + 方向以搜索速度移动。
- ②HOME 信号 ON 后即以蠕动速度移动。
- ③HOME 信号由 ON → OFF 后，Z 相信号由 OFF → ON，即原点搜索完毕。

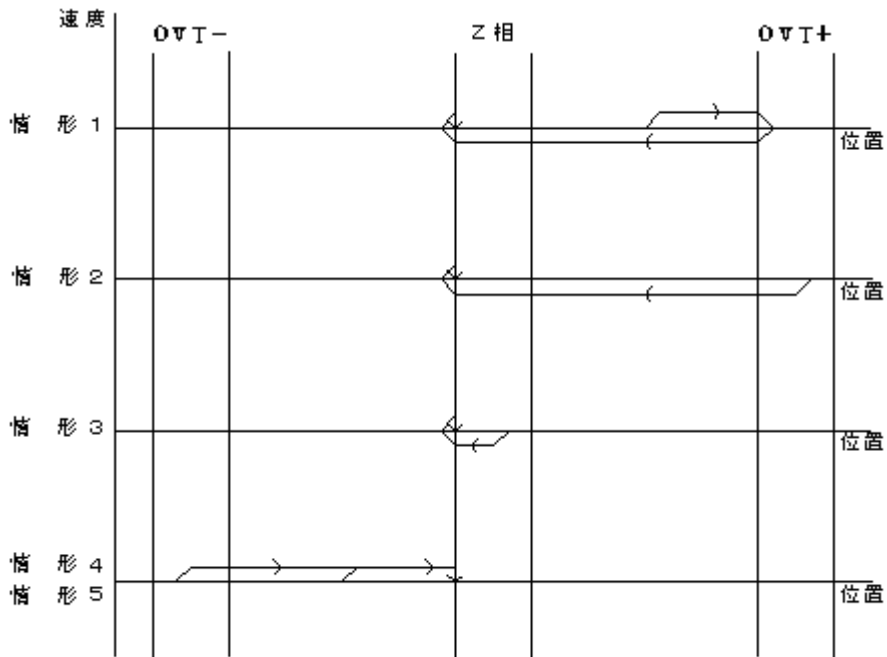
注) 与方式1的动作相同。

4) 方式 3

没有HOME信号进行原点搜索。

《动作说明》

起动方向：+方向、 Z相方向：-方向



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动ON后，向+方向以蠕动速度移动。
- ②OVT+信号ON后即减速，向-方向反转。
- ③Z相信号由ON→OFF后即减速，向+方向反转。
- ④Z相信号由OFF→ON后，即原点搜索完毕。

情形 2：原点搜索开始点，在 [OVT+] 上的情形。

- ①原点搜索起动ON后，向-方向以蠕动速度移动。
- ②Z相信号由ON→OFF后即减速，向+方向反转。
- ③Z相信号由OFF→ON后，即原点搜索完毕。

情形 3：原点搜索开始点，在 [HOME] 上的情形。

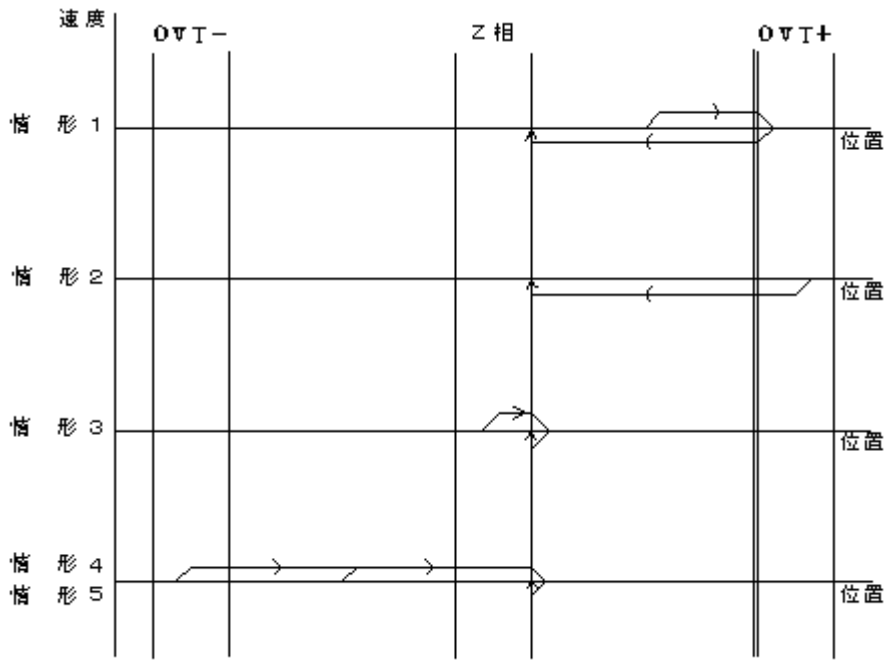
- ①原点搜索起动ON后，向-方向以蠕动速度移动。
- ②Z相信号由ON→OFF后即减速，向+方向反转。
- ③Z相信号由OFF→ON后，即原点搜索完毕。

情形 4：原点搜索开始点，在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点，在 [OVT-] 上的情形

- ①原点搜索起动ON后，向+方向以蠕动速度移动。
- ②Z相信号由OFF→ON后，即原点搜索完毕。

起动方向：+方向、 Z相方向：+方向



情形 1：原点搜索开始点，在 [HOME] 与 [OVT+] 之间的情形

- ①原点搜索起动 ON 后，向 + 方向以蠕动速度移动。
- ②OVT+ 信号 ON 后即减速，向 - 方向反转。
- ③Z 相信号由 OFF → ON 后，即原点搜索完毕。

情形 2：原点搜索开始点，在 [OVT+] 上的情形。

- ①原点搜索起动 ON 后，向 - 方向以蠕动速度移动。
- ②Z 相信号由 OFF → ON 后，即原点搜索完毕。

情形 3：原点搜索开始点，在 [HOME] 上的情形。

- ①原点搜索起动 ON 后，向 + 方向以蠕动速度移动。
- ②Z 相信号由 ON → OFF 后即减速，向 - 方向反转。
- ③Z 相信号由 OFF → ON 后，即原点搜索完毕。

情形 4：原点搜索开始点，在 [OVT-] 与 [HOME] 之间的情形

情形 5：原点搜索开始点，在 [OVT-] 上的情形

- ①原点搜索起动 ON 后，向 + 方向以蠕动速度移动。
- ②Z 相信号由 ON → OFF 后即减速，向 - 方向反转。
- ③Z 相信号由 OFF → ON 后，即原点搜索完毕。

6-3-2. 机械原点移位

如果设定了原点参数中机械原点移位量的值, 则电气原点搜索完毕后, 就自动进行向机械原点的定位。移动完毕后, 该位置定为机械原点, 即时值为 0。
机械原点移位量的设定为「0」时, 该功能不动作。

《动作说明》

原点搜索(方式 0)、Z 相方向(-)、起动方向(+)、机械原点移位量(一值)的情形



6-4. 自动运转

这是实行储存在 Z-01PM 内的 E² PROM 或 PROM 的定位程序的功能。
通过自动运转起动 ON, 使程序运行。

6-4-1. 程序号的指定

Z-01PM 的内部 E² PROM 里能储存 10 个(程序 0~9)程序。
此外在 PROM 里, 登录着 9 个预先准备的简易程序(程序 20~28, 参见 14-7 简易程序)。
要进行自动运转, 需要设定起动哪个程序。(参见 P 25)。
自动运转起动 ON 后, Z-01PM 确认设定的程序号, 使该程序运行。

电源投入时、程序号定为「0」。(电源断开时, 起动程序号不记忆。)
在自动运转起动前, 一定要设定程序号。

6-4-2. 暂停

自动运转过程中, 暂停 ON 后, 即减速停止中断定位动作。
然后, 自动运转起动继电器由 OFF → ON 后, 从中断处继续进行定位动作。
在暂停中, 不能用其他运动方式进行定位动作。

此外, 暂停后继续进行自动运转, 在暂停的「程序行」内发生中断时, 虽然内部中断继电器 Q13ON, 但中断处理不执行。

6-4-3. 工程复位

在暂停中, 如果工序复位 ON, 则可取消中断以后的定位运动。
工序复位 ON 后, 如果自动运转起动由 OFF → ON, 则程序从头开始进行定位动作。
此外, 发生紧急停止时, 以后的工序也被全部取消。

7. 数据转送

要使 Z-01PM 动作，需要与 PLC 交换参数和程序等各种数据。

与 Z-01PM 传送数据，分为 2 大类：

- 指针方式 将数据送给 PLC 的数据寄存器，Z-01PM 自动读出，或者，自动写入数据寄存器的方法。
- WT 方式 由 RD / WT 命令从 PLC 进行传送的方法。

7-1. 指针方式

将要给 Z-01PM 的数据（参数数据，特征数据）给 PLC 的数据寄存器，分别给出储存在 PLC 的特殊寄存器里数据的寄存器起始地址，Z-01PM 读出特殊寄存器的值，将该值作指针使用，要求 PLC 自动读出数据。

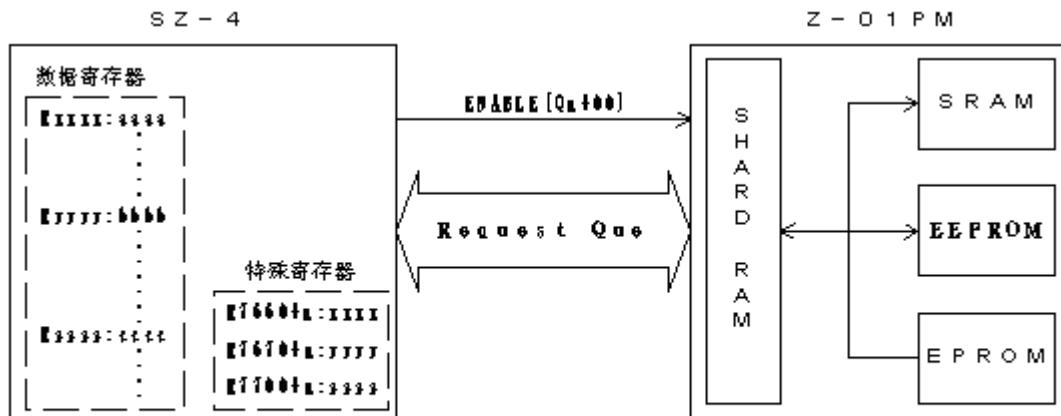
另外，通过把存放从 Z-01PM 读出数据（监视数据）的寄存器的起始地址存放在 PLC 的特殊寄存器中，PLC 可以自动地将监视数据从 Z-01PM 读出到数据寄存器里。

这种方式在用户程序中，即使不使用数据转送命令，也可进行数据转送，因此可以节约用户程序的容量。

但是，数据转送要依靠 PLC 的扫描时间，因此应答速度与 WT 方式比要慢。

而且，Z-01PM 处理的所有数据不能都进行 READ / WRITE。

用指针方式不能处理的数据：程序、特殊参数、PC 寄存器以外的数据寄存器。



《写入顺序 (PLC → Z-01PM)》

- ① 用户预先将参数数据和特征数据设置在数据寄存器里。
(由用户程序或强制写入均可)
- ② 将存储参数数据的数据寄存器的起始地址，以及存储特征数据的数据寄存器的起始地址，用 HEX 设置在特殊寄存器里。
R 7 6 6 0 + n : 参数数据的起始地址
R 7 6 7 0 + n : 特征数据的起始地址
(n 为装 Z-01PM 的槽号)
- ③ 将 ENABLE ON。
- ④ Z-01PM 收到 ENABLE 已 ON，要求特殊寄存器的值。
- ⑤ 返回来的值如果不是 FFFF 或 0000、则依次要求参数数据，特征数据。
- ⑥ 收到的数据里如没有异常，READY 则 ON。数据有异常时，将数据出错 ON，输出出错码。
(要使 READY ON，需要满足其他各种条件。)

设定方法
LDR 02000
OUTW R7662

- ※ 特征数据，在 Z-01PM 不是 BUSY 状态时，每次扫描者要求更新数据。
参数数据，在 ENABLE 已 ON 时，只 1 次读入 Z-01PM。

《读出顺序（PLC ← Z-01PM）》

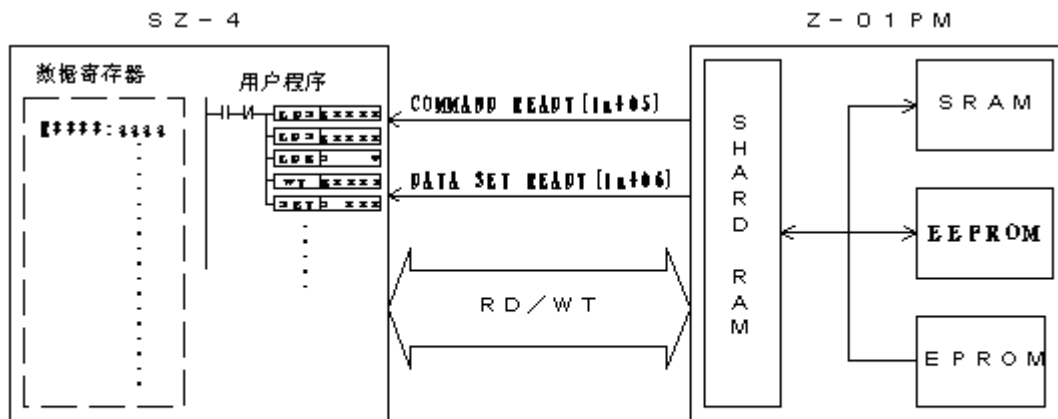
- ①将存储监控数据的数据寄存器的起始地址用HEX设置在特殊寄存器里。
R7700+n：监控数据的起始地址
(n为装插Z-01PM的槽号)
- ②将ENABLE ON。
- ③Z-01PM接收到ENABLE已ON后，要求特殊寄存器的值。
- (-1) ④返回来的值如果不是FFFF 或 0000，则将READY ON后，
每50~100ms将数据写入数据寄存器。

7-2. WT方式

给Z-01PM的所有数据可以通过共用RAM传送。

用WT命令进行转送，因此与前述指针方式相比，可以实现高速数据传送。

但有用户程序容量增加，编制用户程序时要考虑时序等缺点。



《写入顺序（PLC → Z-01PM）》

- (-1) ①PLC确认命令区数据为「5555」后，将转送的数据用指定的格式，由WT命令设置在共用RAM里。
- ②Z-01PM每次扫描（表示PM内部处理主程序）参照共用RAM的命令区。
参照的结果，如果命令已被写入，则Z-01PM立即解析命令，将在数据区内的数据取入模块内。
- ③Z-01PM取入完毕后，将「5555」写入命令区，WT处理终了，回到命令允许接收状态。

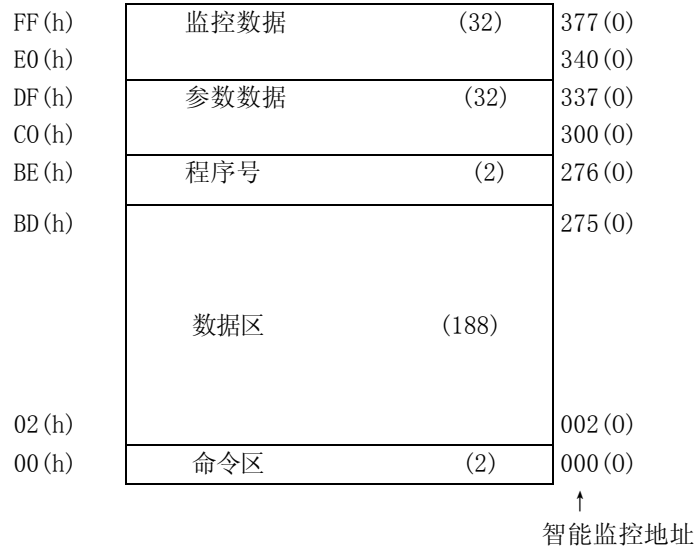
《读出顺序（PLC ← Z-01PM）》

- ①PLC确认命令区数据为「5555」后，将想要的数用指定的格式，由WT命令设置在共用RAM里。
- ②Z-01PM确认命令后，将要求的数据设置在共用RAM里。
设置完毕后，将「3333」写入命令区。
- ③PLC执行WT命令后，由RD命令确认在命令区里，「3333」已被设置，再次由RD命令读出要求的数据。
- (-1) ④读出完毕后，PLC用WT命令写入完了码「AAAA」。
- ⑤Z-01PM确认「AAAA」后，写入「5555」，回到命令允许接收状态。

7-3. 共用RAM

用WT方式与Z-01PM进行数据传送时，通过共用RAM进行。
用户对可以进行存取共用RAM的领域（256字节）定义如下：

《存储器映像》



1) 命令区域 (00-01h)

该区域设置对Z-01PM进行的命令（控制命令）。

Z-01PM根据该命令判断存储在数据区里的数据是什么种类的数据，以及判断PLC要求什么数据。

电源投入时，该区域是「5555」，为允许接收命令的状态。

《命令区域的用途定义》

地址 (HEX)	地址 (OCT)	容 量 (字节)	数 据 种 类	数 据 范 围 (HEX)			
				+ 3	+ 2	+ 1	+ 0
0 0	0 0	2	命 令 码	--	--	0 0 0 0 ~ F F F F	

2) 数据区域 (02-BDh)

数据写入时，是设置要写入Z-01PM里去的数据的区域。

根据设置在命令区的代码，数据被适当处理。

当要读出数据时，从Z-01PM向该区域输出数据。

用途的定义因命令而异。

3) 程序号 (BE-BFh)

该区域设定自动运转起动的程序号。

(-1) 《程序号码区域的用途定义》

地址 (HEX)	地址 (OCT)	容 量 (字节)	数 据 种 类	数 据 范 围 (BCD)			
				+ 3	+ 2	+ 1	+ 0
B E	2 7 6	2	程 序 号	--	--	0 0 0 0 ~ 0 0 1 8	

4) 参数数据 (C0-DFh)

在该区域里，存储着频繁变更的参数。

通过将数据写入该区域，可以改变参数数据。

※停电保持的数据不可变更。(E2PROM)

※参数数据，最初需要全部设定。

《参数数据区域的用途定义》

地址 (HEX)	地址 (OCT)	容 量 (字节)	数据种类	数 据 范 围 (B C D)			
				+ 3	+ 2	+ 1	+ 0
C 0	3 0 0	4	手动速度	0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0			
C 4	3 0 4	4	点动速度	0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0			
C 8	3 1 0	4	点动移动量	0 0 0 0 0 0 0 0 ~ 0 8 3 8 8 6 0 8			
C C	3 1 4	2	超调值	--	--	0 0 0 0 ~ 0 2 5 0	
C E	3 1 6	1 8	预备				

5) 监控数据 (E0-FFh)

在该区域里，写入Z-01PM内部的数据。

每50~100ms更新数据。

《监控数据区域的用途定义》

地址 (HEX)	地址 (OCT)	容 量 (字节)	数 据 种 类	数 据 范 围 (B C D) * 1			
				+ 3	+ 2	+ 1	+ 0
E 0	3 4 0	4	现在位置表示* 2	* 3 0 0 0 0 0 0 0 ~ 0 8 3 8 8 6 0 7			
E 4	3 4 4	4	现在移动量表示	8 0 0 0 0 0 0 0 ~ 8 8 3 8 8 6 0 8			
E 8	3 5 0	4	现在速度表示* 4	0 0 0 0 0 0 0 0 ~ 2 0 0 0 0 0 0 0			
E C	3 5 4	2	出错码	--	--	0 0 0 0 ~ F F F F	
E E	3 5 6	2	剩余暂停时间显示	--	--	0 0 0 0 ~ 9 9 9 9	
F 0	3 6 0	2	行号表示	--	--	0 0 0 0 ~ 9 9 9 9	
F 2	3 6 2	2	辅助码	--	--	0 0 0 0 ~ 0 2 5 5	
F 4	3 6 4	3	输入继电器状态显示	0 0 0 0 0 0 ~ F F F F F F			
F 7	3 6 7	3	输出继电器状态显示	0 0 0 0 0 0 ~ F F F F F F			
F A	3 7 2	6	予約				

* 1：出错码、输入继电器 (I0-I27) / 输出继电器 (Q0-Q27) 状态显示用 H E X 显示。

* 2：现在位置，在电源投入时，原点搜索执行时，清除为「0」。

出错时，在清除出错时，现在位置保持出错前的状态。

此外，可由命令在现在位置里写入数值，但在 B U S Y 中不可写入。

* 3：0：表示+方向、8：表示-方向。

* 4：用绝对值表示。单位 M / D = 1 时：P P S (P / S)

M / D ≠ 1 时：指令单位 / S

※ 3) ~ 5) 的区域，即使不作调整，也可读/写。

7-4. Z-01PM(0.12 或以前版本)使用补充说明

特别注意：在Z-01PM内部数据区中，监控Z-01PM当前位置与写入一个当前位置是不一样的，监控Z-01PM当前位置时，其低字节在前，高字节在后；而写入一个当前位置时，其低字节在后，高字节在前，这与U-01SP不一样。

例 1

要监控 Z-01PM 的当前位置 (Z-01PM 假定放在 1 号槽, 并将位置保存在 R2001, R2000 中)



如果当前位置是+1234567, 那么 R2001=123, R2000=4567

例 2

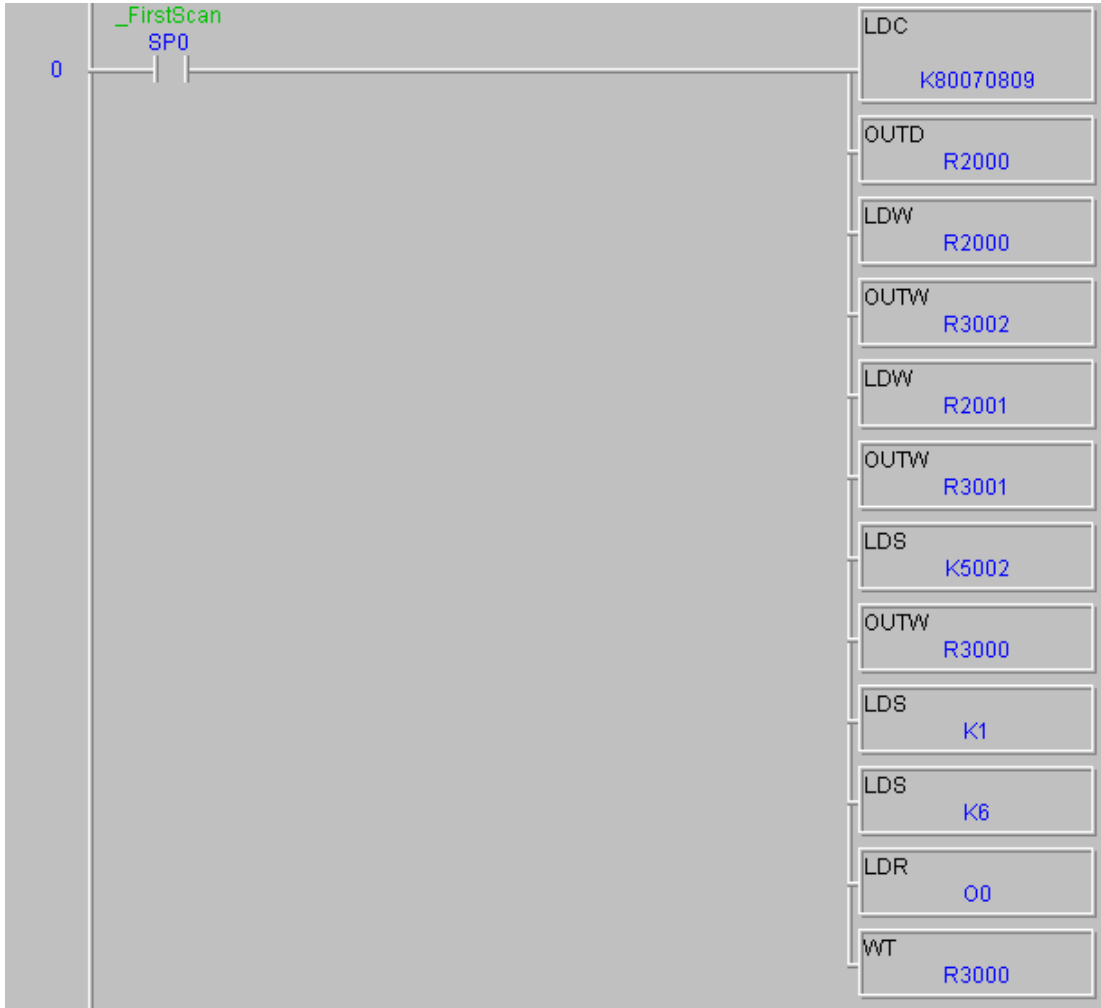
要把一个位置值 (-8388608~ +8388607) 写入 Z-01PM 的当前位置中请采用如下类似格式: 如把+70809 写入当前位置 (Z-01PM 假定放在 1 号槽):



在 Z-01PM 数据区中, 低字节在后, 高字节在前, 这与 U-01SP 不一样。

例 3

再如把-70809 写入当前位置(Z-01PM 假定放在 1 号槽):



8. 监视功能

Z-01PM有监视内部信息的功能。

8-1. 监视数据

Z-01PM能监视的内容有7种。

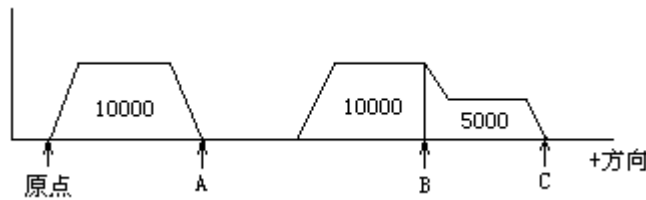
1) 现在位置（4字节：BCD）

以机械原点为原点用座标（绝对位置）显示现在位置。

当进行无限长定位时，显示原点搜索后移动量的累计值。

0 → 1 → 2 …… 0 8 3 8 8 6 0 7 → 8 8 3 8 8 6 0 8 …… 8 0 0 0 0 0 1 → 0 → 1 → 2

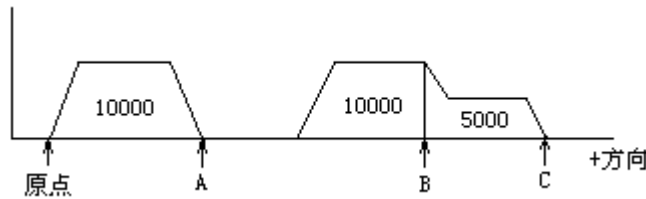
进行无限长定位时，因现在位置没有意义，因此要参照现在移动量。



现在位置表示
A点：10000
B点：20000
C点：25000

2) 现在移动量（4字节：BCD）

显示从定位开始地点起的相对位置。



现在位置表示
A点：10000
B点：10000
C点：15000

3) 现在速度（4字节：BCD）

以绝对值显示现在速度（不是设定速度，而是实际的输出脉冲频率）。单位M/D=1时：PPS（P/S）

M/D≠1时：指令单位/S

4) 出错码（2字节：HEX）

显示出错码。（参照1.5. 出错码）

出错复位ON后，被清除。

5) 剩余暂停时间（2字节：HEX）

定时暂停动作时，显示剩余的暂停时间。

6) 行号（2字节：BCD）

显示程序中的行编号。

	行号
D10=#R2000	1
G00 X1000 F1000	2
G05 X2000 F1500	3
G00 X3000 F500	4
G04 K50	5
G00 X(D10) F1000	6
END	7

7) 辅助码（2字节：BCD）

显示输出的辅助码。

辅助码清除ON后,被清除。
该数据在指针方式下读不出。

8-2. 监视数据的传送 (PLC ← PM)

对传送监视数据的方法进行具体说明。

1) 指针方式

《传送例》

Z-01PM的装插槽号 : n (1~7)
 存储监视数据的数据寄存器的起始地址 : R3000
 监视数据的内容
 现在位置 (4字节) : 80016843
 现在移动量 (4") : 00503658
 现在速度 (4") : 00005300
 出错码 (2") : 0000
 剩余暂停时间 (2") : 0000
 行编号 (2") : 0023
 辅助码 (2") : 0030
 输入继电器状态显示 (3") : 400487
 输出继电器状态显示 (3") : 000005

• 在ENABLE ON前,用HEX将存储监视数据的数据寄存器的起始地址设定在特殊寄存器里。

(-3) $R770n \quad \boxed{600} \quad 600(h) = 3000(O)$
 n: 装插槽号. (1~7)

※ PLC的特殊寄存器地址,因Z-01PM的装插槽号而异。

• 确认Z-01PM的OK已经ON后,将ENABLE ON。

按上述顺序进行后,Z-01PM在ENABLE ON后立即取入特殊寄存器的值。
 然后将全部监视数据(13个寄存器的内容),每50~100ms转送到指定的数据寄存器里。
 被转送的监视数据的次序如下:

R3000	6843	现在位置(下位)	R3010	0023	行编号
1	8001	" (上位)	2	0030	辅助码
2	3658	现在移动量(下位)	3	0487	输入继电器(下位)
3	0050	" (上位)	4	0540	输出(下位)/输入(上位)
4	5300	现在速度(下位)	5	0000	输出继电器(上位)
5	0000	" (上位)	6		
6	0000	出错码	6		
7	0000	剩余暂停时间	7		

用指针方式不监视数据时,特殊寄存器里要设定为「FFFF或0000」。
 出错码,输入继电器状态用HEX传送,其他监视数据,用BCD传送。

2) WT方式

可监视共用RAM的监视数据区的数据。

用RD命令任何时候都可监视。

(无需调停)

传送的数据与用指针方式传送的数据相同。

9. 命令

命令可用共用RAM的地址从02~BDh为止最多188字节，分开进行Z-01PM所有数据的编辑。

《命令一览（PLC→Z-01PM）》

代码	命令	格式
1001	系统参数的读出	(代码)
1002	系统参数的写入	(代码)+(数据)
1004	系统参数的读出 (E ²)	(代码)
1008	系统参数的写入 (E ²)	(代码)
1101	原点参数的读出	(代码)
1102	原点参数的写入	(代码)+(数据)
1104	原点参数的读出 (E ²)	(代码)
1108	原点参数的写入 (E ²)	(代码)
1201	特殊参数的读出	(代码)
1202	特殊参数的写入	(代码)+(数据)
1204	特殊参数的读出 (E ²)	(代码)
1208	特殊参数的写入 (E ²)	(代码)
4001	数据寄存器 (长语句)的读出	(代码)+(起始数据序号)+(数据数)
4002	数据寄存器 (长语句)的的写入	(代码)+(起始数据序号)+(数据数)+(数据)
4004	数据寄存器 (长语句)的读出 (E ²)	(代码)+(起始数据序号)+(数据数)
4008	数据寄存器 (长语句)的写入 (E ²)	(代码)
4101	数据寄存器 (语句)的读出	(代码)+(起始数据序号)+(数据数)
4102	数据寄存器 (语句)的写入	(代码)+(起始数据序号)+(数据数)+(数据)
(-1) 5002	现在位置写入	(代码)+(现在位置)
AAAA	作业完了	(代码)

《命令一览（Z-01PM→PLC）》

代码	命令	格式
(-1) 3333	读出数据设定完了	(代码)
5555	作业完了	(代码)

《代码构成》

第1位	0：从E ² PROM向SRAM复写 1：从SRAM读出 2：从SRAM写入 4：从E ² PROM读出 8：向E ² PROM写入
第2位	無規定
第3位	無規定
第4位	1：有关参数 2：有关程序

4：有关数据寄存器
 5：作业完了
 8：初始化
 A：作业完了

9-1. 命令详情

(-1) 取消线 **=====**：有 **=====** 线的命令，对用户不公开，要用工具支持。

形式：1 0 0 0

功能：将存储在 E² P R O M 里的系统参数在 S R A M 里展开。

地址 (H E X)	命令区域 / 数据区域 格式化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 0	0 0

《系统参数的读出》

代码：1 0 0 1 h

形式：1 0 0 1

功能：将展开在 S R A M 里的系统参数数据读出到共同 R A M。

地址 (H E X)	命令区域 / 数据区域 格式化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 0	0 1

《系统参数的写入》

代码：1 0 0 2 h

形式：1 0 0 2 + [数据]

功能：将写在共同 R A M 里的系统参数数据展开在 S R A M 里。

详情参阅 1 0 . 参数。

《系统参数的读出（E²）》

代 码：1 0 0 4 h

形 式：1 0 0 4

功 能：将存储在E² P R O M里的系统参数数据读出到共同R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 0	0 4

《系统参数的写入（E²）》

代 码：1 0 0 8 h

形 式：1 0 0 8

功 能：将展开在S R A M里的系统参数数据，写入E² P R O M。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 0	0 8

形 式：1 1 0 0

功 能：将存储在E² P R O M里的原点参数展开在S R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 1	0 0

《原点参数的读出》

代 码：1 1 0 1 h

形 式：1 1 0 1

功 能：将展开在S R A M里的原点参数读出到共用R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 1	0 1

《原点参数的写入》

代 码：1 1 0 2 h

形 式：1 1 0 2 + [数据]

功 能：将写在共用R A M里的原点参数展开在S R A M里。

详情参阅1 0 . 参数。

《原点参数的读出（E²）》

代 码：1 1 0 4 h

形 式：1 1 0 4

功 能：将存储在E² P R O M里的原点参数读出到共用R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 1	0 4

《原点参数的写入（E²）》

代 码：1 1 0 8 h

形 式：1 1 0 8

功 能：将展开在S R A M里的原点参数写入E² P R O M。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 1	0 8

形 式：1 2 0 0

功 能：将存储在E² P R O M里的特殊参数复写到S R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 2	0 0

《特殊参数的读出》

代 码：1 2 0 1 h

形 式：1 2 0 1

功 能：将展开在S R A M里的特殊参数读出到共用R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 2	0 1

《特殊参数的写入》

代 码：1 2 0 2 h

形 式：1 2 0 2 + [数据]

功 能：将写在共用R A M里的特殊参数展开在S R A M里。

详情请参阅1 0 . 参数。

《特殊参数的读出（E²）》

代 码：1 2 0 4 h

形 式：1 2 0 4

功 能：将存储在E² P R O M里的特殊参数读出到共用R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 2	0 4

《特殊参数的写入（E²）》

代 码：1 2 0 8 h

形 式：1 2 0 8

功 能：将展开在S R A M里的特殊参数写入E² P R O M。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	1 2	0 8

形 式：2 0 0 0 + [程序编号]

功 能：将存储在E² P R O M里的程序复写到S R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	程序编号		2 0	0 0

形 式：2 0 0 1

功 能：将展开在S R A M里的程序读出到共用R A M里。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	2 0	0 1

形 式：2 0 0 2 + [起始地址] + [数据数] + [数据]

功 能：将写在共用R A M里的程序写入S R A M。

详情请参阅1 2. 程序。

形式：2 0 0 4 + [程序编号]

功能：将存储在E²PROM里的程序读出到共用RAM里。

地址	命令区域 / 数据区域 格式化							
(HEX)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	程序编号		2 0	0 4

形式：2 0 0 8

功能：将展开在SRAM里的程序写入E²PROM。

地址	命令区域 / 数据区域 格式化							
(HEX)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	程序编号		2 0	0 8

形式：2 1 0 4

功能：将存储在E²PROM里的程序的文件名读出到共用RAM里。

地址	命令区域 / 数据区域 格式化							
(HEX)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	2 1	0 4

形式：2 8 8 8 + [程序编号]

功能：将存储在E²PROM里的程序消去。

实际是消去文件名，文件名消去后，程序无效。。

地址	命令区域 / 数据区域 格式化							
(HEX)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	程序编号		2 8	8 8

形式：4 0 0 0

功能：将存储在E²PROM里的数据寄存器的值复写到SRAM的数据寄存器里（长语句）。

地址	命令区域 / 数据区域 格式化							
(HEX)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	4 0	0 0

《数据寄存器(长语句)的读出》

代码：4 0 0 1 h

形式：4 0 0 1 + [起始数据编号] + [数据数]

功能：将展开在SRAM里的数据寄存器（长语句）读出到共用RAM里。

地址	命令区域 / 数据区域 格式化							
(HEX)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	数据数		起始数据编号		4 0	0 1

《数据寄存器(长语句)的写入》

代 码：4 0 0 2 h

形 式：4 0 0 2 + [起始数据编号] + [数据数] + [数据]

功 能：将写在共用 R A M 里的数据写入 S R A M 的数据寄存器里（长语句）。

详情请参阅 1 1 . 数据寄存器。

《数据寄存器(长语句)的读出（E²）》

代 码：4 0 0 4 h

形 式：4 0 0 4 + [起始数据编号] + [数据数]

功 能：将存储在 E² P R O M 里的数据寄存器的值读出到共用 R A M 里。

地 址	命 令 区 域 / 数 据 区 域 格 式 化							
(H E X)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	数据数		起始数据编号		4 0	0 4

《数据寄存器(长语句)的写入（E²）》

代 码：4 0 0 8 h

形 式：4 0 0 8

功 能：将展开在 S R A M 里的数据寄存器的值写入 E² P R O M 。

地 址	命 令 区 域 / 数 据 区 域 格 式 化							
(H E X)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	4 0	0 8

《数据寄存器(语句)的读出》

代 码：4 1 0 1 h

形 式：4 1 0 1 + [起始数据编号] + [数据数]

功 能：将展开在 S R A M 里的数据寄存器(语句)的值读出到共用 R A M 里。

地 址	命 令 区 域 / 数 据 区 域 格 式 化							
(H E X)	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	数据数		起始数据编号		4 1	0 1

《数据寄存器(语句)的写入》

代 码：4 1 0 2 h

形 式：4 1 0 2 + [起始数据编号] + [数据数] + [数据]

功 能：将定在共用 R A M 里的数据写入 S R A M 的数据寄存器里（语句）。

详情请参阅 1 1 . 数据寄存器。

形式：8 8 8 8 + 7 7 7 7 + C C C C + 3 3 3 3 + F F F F

功能：进行E²PROM的初始化。E²PROM的内容全部清除（程序、寄存器消去参数为出厂状态）。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	3 3	3 3	C C	C C	7 7	7 7	8 8	8 8
8	--	--	--	--	--	--	F F	F F

《作业完了（P L C）》

代 码：A A A A h

形 式：A A A A

功 能：告知Z-01PM，作业已经完毕。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	A A	A A

《作业完了（Z-01PM）》

(-1) 代 码：3 3 3 3 h

形 式：3 3 3 3

功 能：告知P L C，读出数据设定完毕。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	3 3	3 3

《作业完了（Z-01PM）》

代 码：5 5 5 5 h

形 式：5 5 5 5

功 能：告知P L C，作业已经完毕。

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	--	--	--	--	--	--	5 5	5 5

10. 参数

参数是定义有关动作的数据。

Z-01PM里有系统参数、原点参数和特殊参数。

为使Z-01PM动作必须进行参数设定。（特殊参数除外）

10-1. 系统参数

(-1) 《系统参数（50字节）》

#	容量 (字节)	数据种类	数据范围（BCD）*1			
			+3	+2	+1	+0
1	2	控制数据	--	--	0000 ~ FFFF	
2	2	输出数据（1）*2	--	--	0000 ~ FFFF	
3	2	输出数据（2）*2	--	--	0000 ~ FFFF	
4	2	电子齿轮（M）	--	--	0001 ~ 9999	
5	2	电子齿轮（D）	--	--	0001 ~ 9999	
6	4	超动时偏置速度	00000001 ~ 20000000			
7	4	速度极限值	00000001 ~ 20000000			
8	4	手动速度*3	00000001 ~ 20000000			
9	4	点动速度*3	00000001 ~ 20000000			
10	4	点动移动量*3	00000000 ~ 08388608			
11	2	超调值*3	--	--	0000 ~ 0250	
12	2	间隙补偿量	--	--	0000 ~ 9999	
13	2	加速时间	--	--	0001 ~ 9999	
14	2	减速时间	--	--	0001 ~ 9999	
15	2	紧急停止减速时间	--	--	0001 ~ 9999	
16	4	软极限（+）	<u>0</u> 0000000 ~ <u>0</u> 8388607*4			
17	4	软极限（-）	<u>8</u> 0000000 ~ <u>8</u> 8388608			
(-3)(-1) 18	2	S字加减速补偿值*5	--	--	0000 ~ 0100	

* 1：控制数据、输出数据（1）、（2）由HEX设定

(-1) * 2：输出数据（1）、（2），数据传送到「E²」后，在下回电源投入时，PM初始化里，新设定有效。

(-1) * 3：手动速度、点动速度 / 移动量、超调值，也可传给共用RAM参数数据区域（参见P26）。

用智能监视进行参数数据区的数据改写时，新的参数数据立即有效，但如果不执行向「E²」的数据传送命令，则下次电源投入时，用智能监视改写的数据无效。

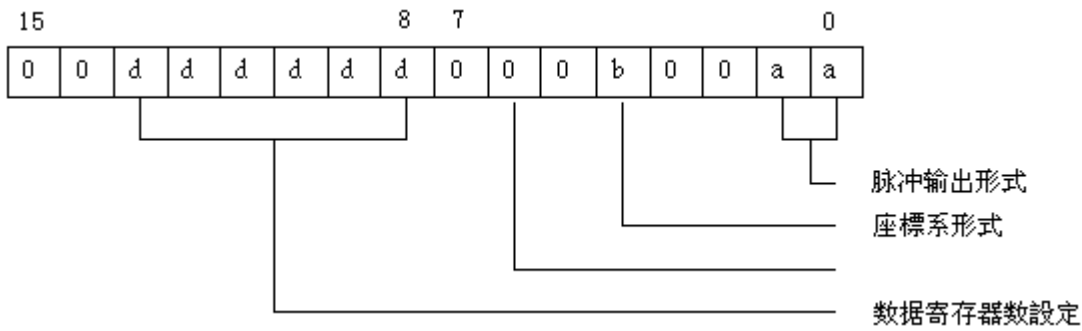
* 4：0表示+方向，8表示-方向。另外，+方向到8388607为止，而且0=80000000。

(-3)(-1) * 5：将S形加减速补偿量设定为「0」时，成为直线加减速。

※预设值用下划线表示。

1) 控制数据

设定系统有关数据。
数据有2个字节,各字节各具意义。



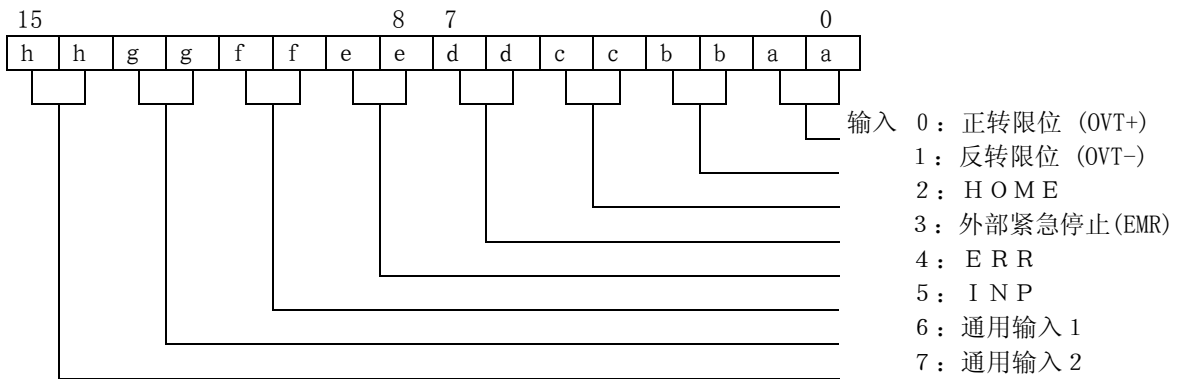
- a a = 0 0 : 予約
- 0 1 : A方式输出
- 1 0 : B方式输出
- 1 1 : 予約
- b = 0 : 形成以机械原点为原点的座标系。
- 1 : 不形成座标系 (定程定位等无限长定位等)

(-1)

d d d d d d = 0 ~ 2 0 (HEX) 2 0
: 用指针方式将P L C的数据寄存器的值进行传送时的数据寄存器的传送数

2) 输出输入数据 (1)

对Z-01PM外部I/O的功能进行定义。
数据有2个字节,各字节各具意义。

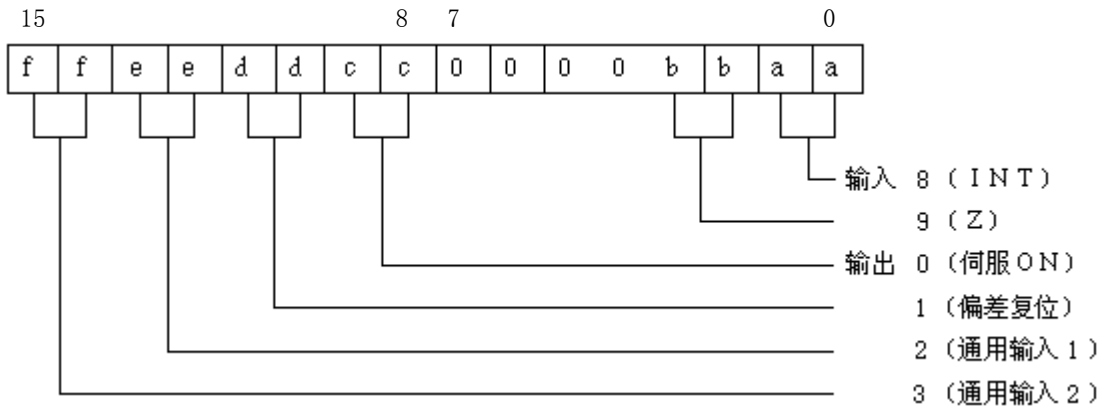


- a a = 0 0 : 正转限位输入 (a 触点)
- 0 1 : " (b 触点)
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- b b = 0 0 : 反转限位输入 (a 触点)
- 0 1 : " (b 触点)
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- c c = 0 0 : HOME输入 (a 触点)
- 0 1 : " (b 触点)
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- d d = 0 0 : 外部紧急停止输入 (a 触点)
- 0 1 : " (b 触点)
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- e e = 0 0 : ERR输入 (a 触点)
- 0 1 : " (b 触点)
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- f f = 0 0 : 到位输入 (a 触点)
- 0 1 : " (b 触点)
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- g g = 0 0 : 予約
- 0 1 : 予約
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約
- h h = 0 0 : 予約
- 0 1 : 予約
- 1 0 : 通用输入 (a 触点)
- 1 1 : 予約

3) 输出数据 (3)

对 Z-01PM 外部 I/O 的功能进行定义。

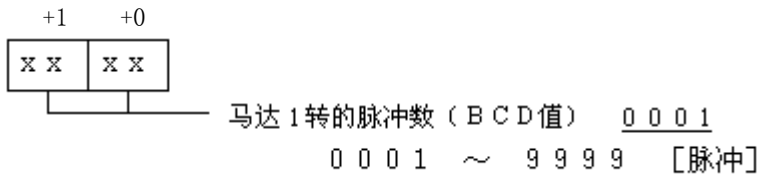
数据有 2 个字节，各字节各具意义。



- | | |
|---|--|
| <p>a a = <u>0 0</u>: 外部中断输入 (a 触点)
 0 1: 予約
 1 0: 予約
 1 1: 予約</p> <p>b b = <u>0 0</u>: Z 相输入 (a 触点)
 0 1: 予約
 1 0: 予約
 1 1: 予約</p> <p>c c = <u>0 0</u>: 伺服 ON 输出 (a 触点)
 0 1: 予約
 1 0: 通用输入 (a 触点)
 1 1: 予約</p> | <p>d d = <u>0 0</u>: 偏差复位输出 (a 触点)
 0 1: 予約
 1 0: 通用输入 (a 触点)
 1 1: 予約</p> <p>e e = <u>0 0</u>: 予約
 0 1: 予約
 1 0: 通用输入 (a 触点)
 1 1: 予約</p> <p>f f = <u>0 0</u>: 予約
 0 1: 予約
 1 0: 通用输入 (a 触点)
 1 1: 予約</p> |
|---|--|

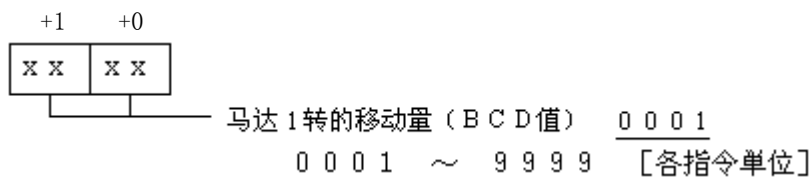
4) 电子齿轮 (M)

进行电子齿轮的设定。电子齿轮用电子齿轮 (M) 和电子齿轮 (D) 2 个作 1 个设定，因此双方都要设定。



5) 电子齿轮 (D)

进行电子齿轮的设定。



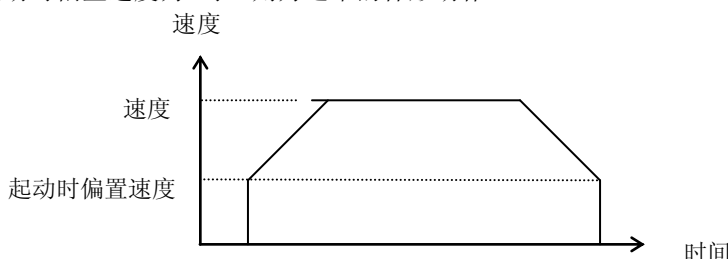
6) 起动时偏置速度

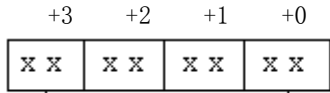
进行起动时偏置速度的设定。

所有的运转按下图动作。

起动时偏置速度 ≥ 速度时，进行快速加速、快速减速。

起动时偏置速度为 0 时，则为通常的梯形动作。

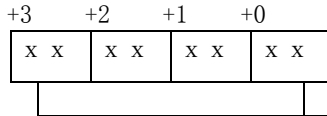




起动时偏置速度 (BCD值) 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0 [各指令单位 / s]
 设定不可超过速度极限值。

7) 速度极限值

进行速度极限值的设定。
这个速度极限值，适用于全部移动速度。

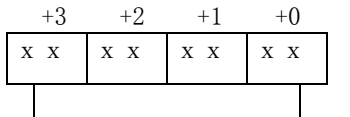


速度极限值 (BCD值) 0 0 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0 [各指令单位 / s]

速度极限值的设定应满足： $\text{速度极限值} \times M / D \leq 4 0 0 0 0 0$ 。

8) 手动速度

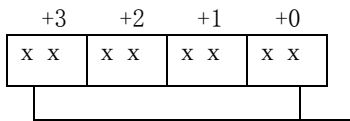
进行手动速度的设定。



手动速度 (BCD值) 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0 [各指令单位 / s]
 设定不可超过速度极限值。

9) 点动速度

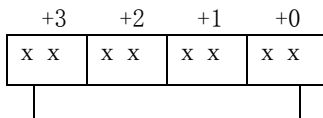
进行点动速度的设定。



点动速度 (BCD值) 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0 [各指令单位 / s]
 设定不可超过速度极限值。

10) 点动移动量

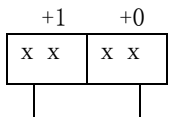
进行点动移动量的设定。



点动移动量 (BCD值) 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 0 ~ 0 8 3 8 8 6 0 8 [各指令单位]

11) 超调值

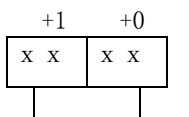
进行超调值的设定。



超调值 (BCD值) 0 1 0 0
 0 0 0 0 ~ 0 2 5 0 [%]

12) 间隙补偿量

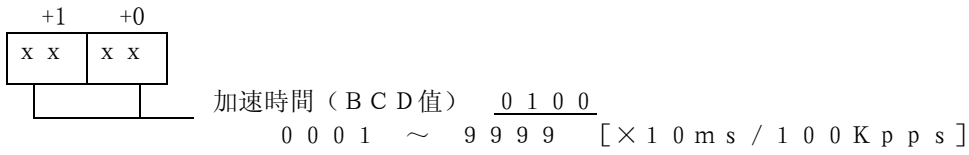
进行间隙补偿量的设定。



间隙补偿量 (BCD值) 0 0 0 0
 0 0 0 0 ~ 9 9 9 9 [脉冲]

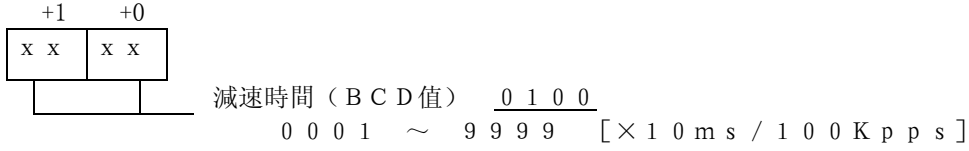
1 3) 加速時間

进行加速時間的設定。



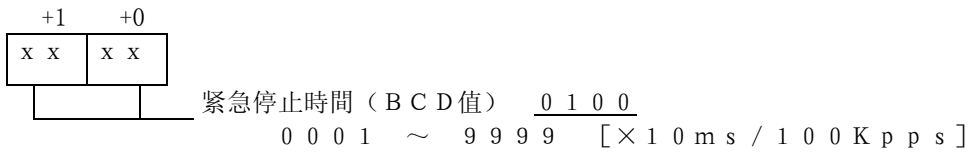
1 4) 減速時間

进行減速時間的設定。



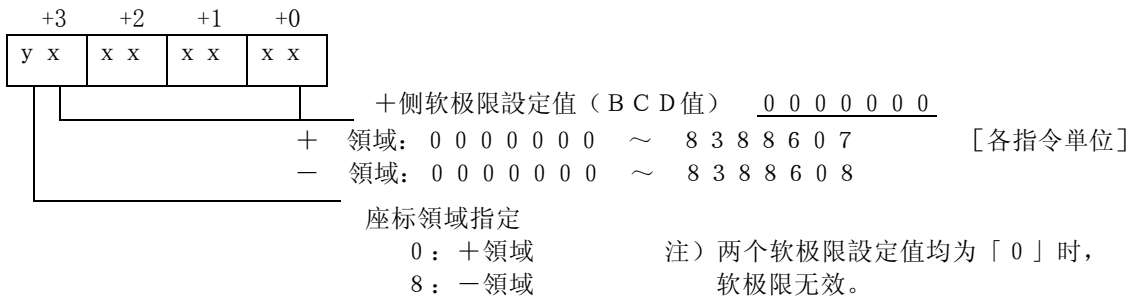
1 5) 緊急停止時間

进行緊急停止時間的設定。



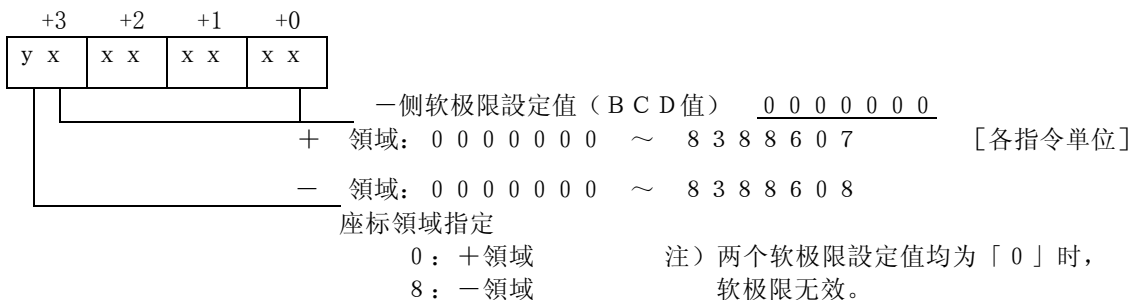
1 6) 軟極限 (+)

进行+側軟極限的設定。



1 7) 軟極限 (-)

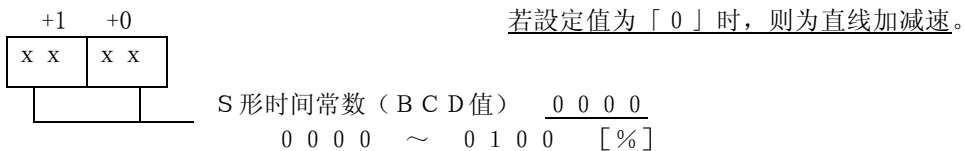
进行-側軟極限的設定。



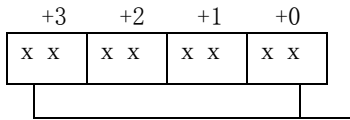
(-3) (-1)

1 8) S形加減速補償值

設定S形加減速時的補償值。

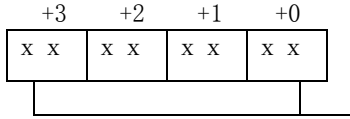


3) 蠕动速度
进行蠕动速度的设定。



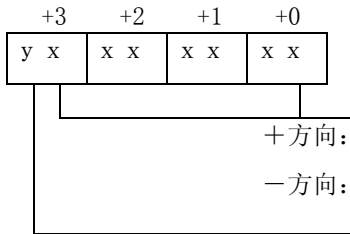
蠕动速度 (BCD值) 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0 [各指令单位 / s]
 设定不可超过速度极限值。

4) 移位速度
进行移位速度的设定。



移位速度 (BCD值) 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 1 ~ 2 0 0 0 0 0 0 0 [各指令单位 / s]
 设定不可超过速度极限值。

5) 移位量
进行移位量的设定。



移位量 (BCD值) 0 0 0 0 0 0 0 0
 十方向: 0 0 0 0 0 0 0 0 ~ 8 3 8 8 6 0 7
 一方向: 0 0 0 0 0 0 0 0 ~ 8 3 8 8 6 0 8 [各指令单位]

移位方向指定
0: 十方向
8: 一方向

1 0 - 4 . 参数的传送 (PLC → PM)

对参数传送的方法作具体说明。

1) 指针方式

在指针方式下, 系统参数和原点参数一起处理。

特殊参数, 在指针方式下不传送。必要时, 可用WT方式传送。

《传送例》

Z-01PM的装插槽口 : n (1~7)
 存储参数的数据寄存器的起始地址 : R 2 0 0 0 (OCT)

一传送的参数

	控制数据	: 0 0 0 1	減速時間	: 0 0 1 0
	输入输出数据 (1)	: 0 1 4 5	紧急停止減速時間	: 0 0 0 5
	" (2)	: 0 0 0 0	软极限 (+)	: 0 1 0 0 0 0 0 0
	電子齿轮 (M)	: 0 0 1 2	软极限 (-)	: 8 1 0 0 0 0 0 0
(-3) (-1)	" (D)	: 0 0 0 1	S形加減速补偿值	: 0 0 5 0
(-3)	起动时偏置速度	: 0 0 0 0 0 0 1 0	起动速度	: 0 0 0 0 0 0 0 0
	速度极限值	: 0 0 0 1 6 0 0 0		
	手动速度	: 0 0 0 0 0 0 8 3	检索数据	: 0 0 0 1
	点动速度	: 0 0 0 0 0 0 8 3	检索速度	: 0 0 0 0 1 0 0 0
	点动移动量	: 0 0 0 0 1 0 0 0	蠕动速度	: 0 0 0 0 0 0 8 3
	超调值	: 0 1 0 0	移位速度	: 0 0 0 0 1 0 0 0
	间隙补偿量	: 0 0 0 0	移位量	: 0 0 0 2 6 4 0 0
	加速時間	: 0 0 1 0		

• • 用BCD按以下顺序将参数设定在数据寄存器里。

R 2 0 0 0	0 0 0 1	控制数据	R 2 0 0 2	0 0 1 0	減速時間
1	0 1 4 5	输入输出数据 (1)	3	0 0 0 5	緊急停止時間
2	0 0 0 0	" (2)	4	0 0 0 0	软极限+ (下位)
3	0 0 1 2	電子齿轮 (M)	5	0 1 0 0	" (上位)
4	0 0 0 1	電子齿轮 (D)	6	0 0 0 0	软极限- (下位)
5	0 0 1 0	起动时偏置速度 (下位)	7	8 1 0 0	" (上位)
6	0 0 0 0	" (上位)	R 2 0 3 0	0 0 5 0	S形加減速补偿值 (-1) (-3)
7	6 0 0 0	速度极限值 (下位)	1	0 0 0 0	未使用
R 2 0 1 0	0 0 0 1	" (上位)	2	0 0 0 0	" (上位)
1	0 0 8 3	手动速度 (下位)	3	0 0 0 1	检索数据 (原点)
2	0 0 0 0	" (上位)	4	1 0 0 0	检索速度 (下位)
3	0 0 8 3	点动速度 (下位)	5	0 0 0 0	" (上位)
4	0 0 0 0	" (上位)	6	0 0 8 3	蠕动速度 (下位)
5	1 0 0 0	点动移动量 (下位)	7	0 0 0 0	" (上位)
6	0 0 0 0	" (上位)	R 2 0 4 0	1 0 0 0	移位速度 (下位)
7	0 1 0 0	超调值	1	0 0 0 0	" (上位)
R 2 0 2 0	0 0 0 0	间隙补偿量	2	6 4 0 0	移位量 (下位)
1	0 0 1 0	加速時間	3	0 0 0 2	" (上位)

• 将存储着参数的数据寄存器的起始地址用HEX设定在特殊寄存器里。

R 2 0 0 0 时 4 0 0 (h)

R 7 6 6 n 4 0 0 n: 装插槽号NO. (1~7)

※PLC的特殊寄存器地址,因Z-01PM装插槽号而异。

• • 确认Z-01PM的OK已ON,并交ENABLE ON。

按上述顺序进行数据设定后,Z-01PM在ENABLE ON后,立即进行1次数据转送。

注)输入输出数据,仅电源投入时由「EEPROM」传送的数据有效。电源投入后的输入输出数据的变更无效。

在不用指针方式传送参数时,要在特殊寄存器里设定「FFFF或0000」。

2) WT方式

在WT方式下，系统参数与原点参数分别处理。

向共用RAM内指定的区域里传送，并要在命令区里设定命令。

共用RAM内的格式如下。

《系统参数》

地 址 (H E X)	命 令 区 / 数 据 区							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	输出数据 (1)		输出数据 (2)		控制数据		1 0	0 2
8	起动时偏置速度				電子齿轮 (D)		電子齿轮 (M)	
1 0	手动速度				速度极限值			
1 8	点动移动量				点动速度			
2 0	減速時間		加速時間		間隙补偿量		超调值	
2 7	软极限 (-)		软极限 (+)				紧急停止減速時間	
(-3)(-1) 3 0	未使用				S形加減速补偿值		软极限(-)	

《原点参数》

地 址 (H E X)	命 令 区 / 数 据 区							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	检索速度				检索数据		1 1	0 2
8	移位速度				蠕动速度			
1 0	---	---	---	---	移位量			

将由指针方式处理的数据，实际传送给共用RAM后，情况如下：
在命令区里自动写入命令，Z-01PM据此进行处理。

《系统参数》

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	0 0	0 0	0 1	4 5	0 0	0 1	* 1 1 0	0 2
8	0 0	0 0	0 0	0 1	0 0	0 1	0 0	1 2
1 0	0 0	0 0	0 0	8 3	0 0	0 1	6 0	0 0
1 8	0 0	0 0	1 0	0 0	0 0	0 0	0 0	8 3
2 0	0 0	1 0	0 0	1 0	0 0	0 0	0 1	0 0
2 8	0 0	0 0	1 0	0 0	0 0	0 0	0 0	0 5
3 0	0 0	0 0	0 0	0 0	0 0	5 0	8 1	0 0

(-3)

《原点参数》

地 址 (H E X)	命 令 区 域 / 数 据 区 域 格 式 化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	0 0	0 0	1 0	0 0	0 0	0 1	* 1 1 1	0 2
8	0 0	0 0	1 0	0 0	0 0	0 0	0 0	8 3
1 0	--	--	--	--	0 0	0 2	6 4	0 0

* 1：在命令区里自动写入的命令
1 0 0 2：系统参数的写入
1 1 0 2：原点参数的写入

1 1. 数据寄存器

Z-01PM在SRAM领域有数据寄存器。

数据寄存器有32位与16位。

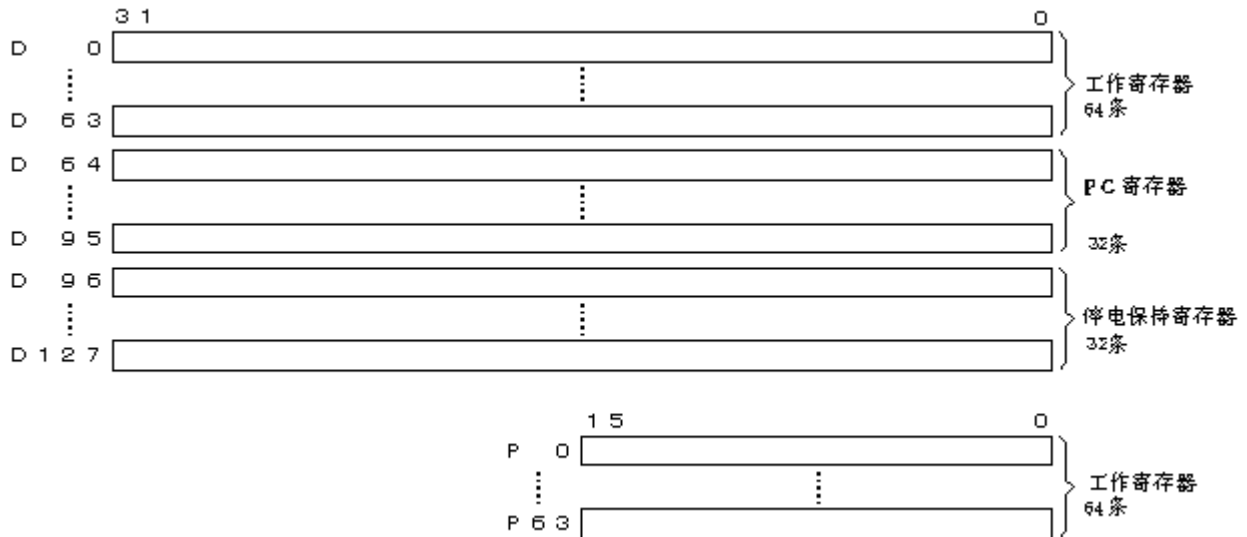
32位寄存器(D寄存器)备有128条,3种寄存器。

分别称为工作寄存器、PC寄存器和停电保持寄存器,用途各异。

16位寄存器(P寄存器)备有64条,只有工作寄存器一种。

数据传送方法,所有寄存器都是WT方式,或用控制命令(G63)进行。

但只有PC寄存器可用指针方式进行数据传送。



1 1-1. 各寄存器的说明

1) 工作寄存器

在32位寄存器(以下称D寄存器)和16位寄存器(以下称P寄存器)里各备有64条。

D寄存器和P寄存器都会因电源OFF而使内容被消除。

而且,在电源投入时全部写0。

D寄存器按Z-01PM内的程序,可用于比较运算、数据传送、数据的间接指定等。

P寄存器按Z-01PM内的程序,可用于比较运算、数据传送、但不可用于数据的间接指定。

2) PC寄存器

在D寄存器里备有32条。

用指针方式,可以自动地从PLC的数据寄存器进行数据传送。

数据传送,在PM系统的主程序(每50~100ms1次)内进行。在Z-01PM在BUSY状态时,不能进行数据传送。

数据传送一定要从D64起进行。

传送的数据,由系统参数的控制数据的8-13位设定。

(参阅10-1.系统参数。设定值:如为20h,则32条全部可传送。)

此寄存器也可作工作寄存器使用。

与工作寄存器一样,会因电源OFF而内容被消除。

3) 停电保持寄存器

在D寄存器里备有32条。

电源投入时,停电保持(保存在EEPROM里)的数据复制到此寄存器。

使用命令,通过将此寄存器的值写入停电保持区域(EEPROM)里,可实现停电保持。

注)由D寄存器处理的数值,是「BCD」。能够输入的BCD数值范围为:

正值范围:0~08388607

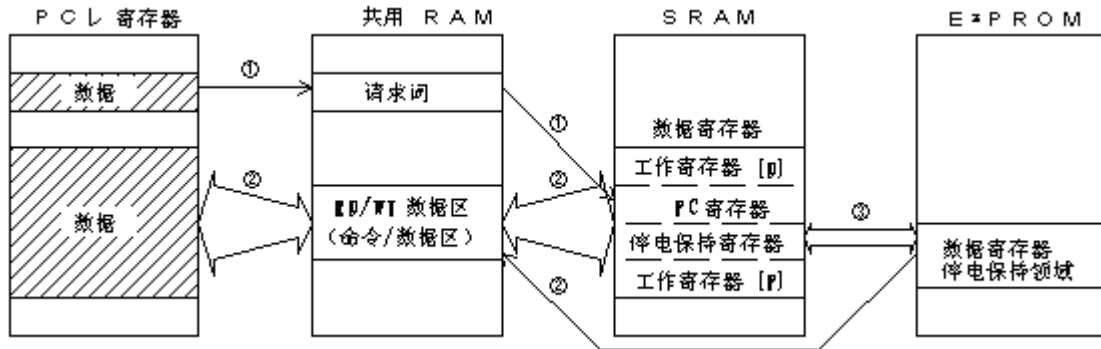
负值范围:88388608~0(0=80000000。)

8：表示负值。

由P寄存器处理的数值，是「HEX」，输入范围为：0～FFFF。

1 1 - 2 . 数据的流动

数据的流动如下图所示。



①用指针方式的数据流动（仅限PC寄存器）

从由特殊寄存器指定的PLC的数据寄存器，将数据复制到SRAM的PC寄存器里。

Z-01PM的扫描中，对PLC要求1次数据，将应要求而送来的数据从共用RAM传送给SRAM。数据的更新为每50～100ms进行1次。但在BUSY中时，因PM不要求数据，所以不能更新。

②RD/WT方式的数据流动

根据由PLC发出的读出命令，由命令指定的数据从SRAM或E²PRO M读出，复写到共用RAM的数据区里。

复写到共用RAM里的数据，根据RD命令可以从PLC读出。

根据由PLC发出的写入命令，在共用RAM的数据区域里的数据，向指定的SRAM的领域转送。

注)在BUSY中时，如发出写入命令，则发生出错，但不会紧急停止。

此外，在BUSY中时，即使让出错复位信号(Q11)ON，也不能解除出错状态。而且，读出命令即使是BUSY中也接收，将要求数据设定到共用RAM里。

③E²PRO M与SRAM间的数据流动

根据从PLC发出向E²PRO M的写入命令，SRAM的停电保持寄存器的值，向E²PRO M复制。从E²PRO M向SRAM停电保持寄存器的数据复制，在电源投入时进行。

此外，在BUSY中时，如发出向E²PRO M的写入命令，将发生与上述注)同样的症状。

1 1 - 3 . 数据的传送

对传送数据的方法作具体说明。

1) 指针方式

在指针方式下，只有PC寄存器(D64-95)的32条数据寄存器可以传送数据。

传送的数据数，由系统参数的控制数据的8-13位设定，传送开始寄存器一定从「D64」开始。

《传送例》

Z-01PM的装插槽号	: n (1~7)
存储了数据的数据寄存器的起始地址	: R4000
传送的数据数	: 16

自动地更新PM内D寄存器的值。

但是，Z-01PM呈BUSY状态时，因不对PLC要求数据，所以不进行数据更新。

2) WT方式

在WT方式下，可以处理所有的数据寄存器。

D寄存器与P寄存器区别对待。

一次能够传送的量，D寄存器最大为32条，P寄存器最大为64条。

向共用RAM的指定领域里传送，需要在命令区设定命令。

共用RAM内的格式如下：

《D寄存器写入数据时，共用RAM内的格式》

地址 (HEX)	命令区域 / 数据区域 格式化							
	+7	+6	+5	+4	+3	+2	+1	+0
0	数据1 (下位)		数据数		起始数据号		40	02
8	数据3 (下位)		数据 2				数据1 (上位)	
10	数据5 (下位)		数据 4				数据3 (上位)	
70	数据29 (下位)		数据 28				数据27 (上位)	
78	数据31 (下位)		数据 30				数据29 (上位)	
80	--	--	数据 32				数据31 (上位)	

起始数据号：设定存储起始数据的数据寄存器号码。(BCD)

数据数：设定传送的数据的数量。(BCD)

D寄存器 数据写入程序例

Z-01PM的装插槽号 : 2
 存储数据的数据寄存器的起始地址 : R4000
 写入的数据(D寄存器)数 : 16

```

LD SP1
LDS K2
LDS K2
LDR O0
RD R3000
LDEQ R3000 K5555 ← PM命令接收許可状态判断
ANDN I2 ← BUSY继电器
LDS K2 ← 2号槽(SZ只有0号基架)
LDS K70 ← 命令(2) + 先头号(2) + 数据数(2) + 数据(64)
LDR O0 ← 从共用RAM地址「0」起写入
WT R4000 ← 从R4000起传送70字
    
```

注) 从R 4 0 0 0起, 需要记述以下数据。

```

R 4 0 0 0    4 0 0 2 (数据寄存器长语句写入命令)
              0 1    0 0 3 2 (从D 3 2起写入的意思: B C D)
              0 2    0 0 1 6 (传送1 6个D寄存器的数据的意思: B C D)
              0 3    数据 1 (下位)
              0 4    数据 1 (上位)
              0 5    数据 2 (下位)
              0 6    数据 2 (上位)
              0 7    数据 3 (下位)
                    ⋮
              4 0    数据1 5 (下位)
              4 1    数据1 6 (上位)
              4 2    数据1 6 (下位)

```

D寄存器 数据读出程序例

```

Z-01PM的装插槽号                : 2
存储读出数据的数据寄存器的起始地址 : R 2 0 0 0
读出的数据 (D寄存器) 的数量        : 1 6

L D   S P 1
A N D N   M 0
L D S   K 2                ← 2号槽 (S Z只有0号基架)
L D S   K 2
L D R   O 0
R D   R 3 0 0 0
L D E Q   R 3 0 0 0   K 5 5 5 5 ← P M命令接收许可状态判断
S E T   M 0
L D   M 0
P D   M 1
L D   M 1
L D S   K 2
L D S   K 6                ← 命令(2) + 起始号(2) + 数据数(2)
L D R   O 0                ← 向共用R A M地址「0」写入
W T   R 4 0 0 0           ← 从R 4 0 0 0起传送6字节
L D   M 0
L D S   K 2
L D S   K 2
L D R   O 0
R D   R 3 0 0 0
L D E Q   R 3 0 0 0   K 3 3 3 3 ← P M数据设定完毕判断
L D S   K 2
L D S   K 6 4
L D R   O 2                ← 读出共用R A M地址「2」
R D   R 2 0 0 0           ← 将6 4字节从R 2 0 0 0起存入
R S T   M 0
L D S   K 2
L D S   K 2
L D R   O 0
W T   R 4 0 0 3           } 向P M报告读出完毕。

```

注) 从R 4 0 0 0起需要记述以下数据。

```

R 4 0 0 0    4 0 0 2 (数据寄存器长语句读出命令)
              0 1    0 0 1 6 (从D 1 6起读出的意思: B C D)
              0 2    0 0 3 2 (传送3 2个D寄存器的数据的意思: B C D)
              0 3    A A A A (读出完毕)

```

P寄存器的格式如下：

《P寄存器 数据写入时，共用RAM内的格式》

地址 (HEX)	命令区域 / 数据区域 格式化							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	数据 1		数据		起始数据编号		4 1	0 2
8	数据 5		数据 4		数据 3		数据 2	
1 0	数据 9		数据 8		数据 7		数据 6	
⋮								
7 0	数据 5 7		数据 5 6		数据 5 5		数据 5 4	
7 8	数据 6 1		数据 6 0		数据 5 9		数据 5 8	
8 0	--	--	数据 6 4		数据 6 3		数据 6 2	

程序与D寄存器相同。

1 2. 程序

Z-01PM的自动运转，按自己的程序进行运转。

1 2-1. 程序中的文字、記号

在Z-01PM中使用的文字、記号、数字如下：

《命令码，操作数》

- G：指定命令码。
- M：指定辅助码。
- N：指定标号号码。
- X：指定坐标值。
- F：指定送进速度。
- K：指定常数。
- D：指定Z-01PM内部的数据寄存器（D0~127）。
- P：指定Z-01PM内部的数据寄存器（P0~63）。
- I：指定Z-01PM内部的输入（I0~31）。
- Q：指定Z-01PM内部的输出（Q0~24）。

《数字》

- 0~9：表示10进制数8进制数。
- ：表示負的数值。

《記号》

- #：指定PLC的寄存器，继电器时使用。
 - #R：数据寄存器
 - #Q：输出继电器
- ()：由间接指定给值时使用。
- #I：输入继电器
- #M：内部继电器

《运算符》

- +
 -
 - *
 - /
 - =
- ：加。
：減。
：乘。
：除。
：代入数值。

《比較算符》

- =
 - <
 - >
 - <=
 - =<
 - >=
 - =>
 - <>
 - ><
 - AND
 - OR
- ：等于
：小于
：大于
：小于等于
："
：大于等于
："
：不等于
："
：与
：或

1 2 - 2 . 程序的命令形态

Z - 0 1 P M 的程序由 3 种命令形态组成。

1) 定位控制命令

定位控制命令以如下形式记述。

①	②	③	
N 1 0 0	G 0 0	X 5 0 0	F 1 0 0

① 标号号码 (Nxxxx)

作为程序中转移地址的标号使用。

在记号“N”之后，记述 0 ~ 9 9 9 9 为止的 1 0 进数字。

标号号码可以省略，但不能单独存在。

② 命令码 (Gxx)

是对 Z - 0 1 P M 发命令的码。

在记号“G”之后，记述 0 0 ~ 9 6 为止的 1 0 进数字。

在同一行上不可记述多个命令码。

命令码不可省略。

③ 操作数

记述执行命令码时需要的信息。

种类因命令码而异，也有不需要的。

2) 辅助码控制命令

是为使与定位控制连动，与 P L C 连接的各种辅助装置（夹头、阀门等）动作的命令。

在记号“M”之后，记述 1 ~ 2 5 5 为止的 1 0 进数字。

辅助码控制命令单独记述。

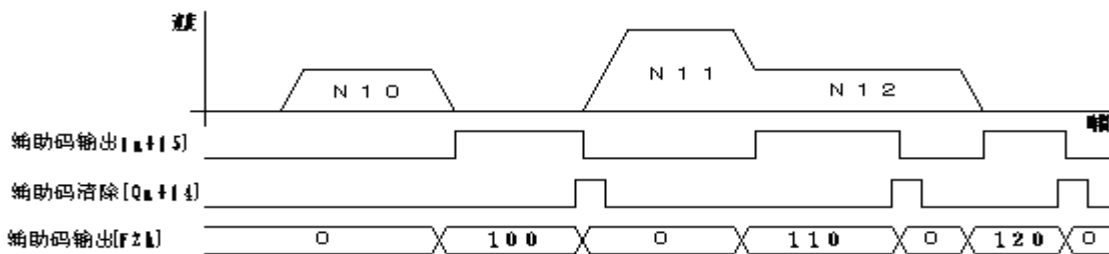
辅助码控制命令执行后，由共用 R A M 监视区的输出 (F 2 h) 里输出辅助码。同时使辅助码输出 (I n + 1 5) O N。

辅助码输出，如辅助码清除 (Q n + 1 4) 不 O N，则不能 O F F。

《程序例》

```

N 1 0   G 0 0   X 5 0 0   F 1 0 0
        M 1 0 0
        G 6 1   Q 1 2 = K 1
N 1 1   G 0 5   X 1 0 0 0   F 2 0 0
        M 1 1 0
N 1 2   G 0 0   X 1 5 0 0   F 1 0 0
        M 1 2 0
    
```



3) 运算命令

Z-01PM的数据寄存器,或PLC的数据寄存器、继电器的运算都可以在程序中执行。

①运算内容

- 代入 (○○○○=□□□□)
将□□□□的内容代入○○○○。
- 加算 (○○○○=□□□□+△△△△)
将□□□□的内容与△△△△的内容之「和」代入○○○○。
- 减算 (○○○○=□□□□-△△△△)
将□□□□的内容与△△△△的内容之「差」代入○○○○。
- 乘算 (○○○○=□□□□*△△△△)
将□□□□的内容与△△△△的内容的「积」代入○○○○。
- 除算 (○○○○=□□□□/△△△△)
将□□□□的内容与△△△△的内容的「商」代入○○○○,并将「余」代入○○○○+1。

②运算可用参数

- 数据寄存器 (D0~127)
连符号4字节长整数 [BCD]
数值范围: -8388608~8388607
- 数据寄存器 (P0~63)
连符号2字节长整数 [HEX]
数值范围: 0~FFFF (-32768~32767)
- PLC的数据寄存器 (#Rxxxx)
连符号2字节长整数 [HEX]
数值范围: 0~FFFF (-32768~32767)
- PLC的继电器 (#I、#Q、#M)
1位的数据 (0、1)
- Z-01PM的继电器 (I、Q)
1位的数据 (0、1)
- 常数
连符号4字节长整数 [BCD]
数值范围: -8388608~8388607
在常数前,一定要加「K」。

③可运算的参数组合

- 代入
#R=#R、D、P、K常数
D=#R、D、P、K常数
P=#R、P、K常数
(-1) I=K常数0、1
Q=K常数0、1 (Q0=K0 或 Q0=K1)
#I=K常数0、1
#Q=K常数0、1
#M=K常数0、1

常数0、1: 设定为0或1。

• 加减乘除运算

- #R=(#R、D、P、K常数) +-*/ (#R、D、P、K常数)
- D=(#R、D、P、K常数) +-*/ (#R、D、P、K常数)
- P=(#R、P、K常数) +-*/ (#R、P、K常数)

④运算执行时的注意事项

- 在 $(\#R, P) = (D, \text{常数})$ 的代入时，D和常数会变换成2字节长的二进制。
在变换时，高位被舍去。

《例》

在 $\#R2000 = -30000$ 里，将 $[-30000]$ 变换成二进制，为 $[FFFFFF8AD0]$
舍去高位2字节后，将 $[8AD0]$ 代入 $R2000$ 。

- 在 $(\#R, P) = (D, \text{常数})$ 的代入时，D和常数的范围，是 $-32768 \sim 32767$ 。若设定了超过此范围的数值时，则不能进行正常运算。

《例》

在 $\#R2000 = -50000$ 里， $[-50000]$ 变换成二进制后后，成为 $[FFFFFF3CB0]$ 。
但 $[3CB0]$ 变换成BCD后，成为 $[15536]$ 。

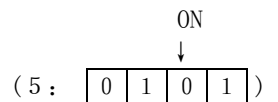
- (-1) • 能代入 $\#I$ 、 $\#Q$ 、 $\#M$ 、 I 、 Q 的数值，是 $[1]$ 或 $[0]$ 。

将其他数值记述在右边时，数值的低位代入。

《例》

若 $\#Q2 = K1$ ，则对PLC要求 $[\#Q2 = ON]$ 。

若 $\#Q2 = K5$ 时，对PLC也要求 $[\#Q2 = ON]$ 。



- $D = (\#R, P)$ 的代入时，其 $\#R$ 和 P 进行4字节长的BCD变换。

《例》

若 $D0 = \#R2000$ ， $R2000$ 为 $[9F5C]$ 时，则将BCD变换后的值
 $[-24740]$ 代入 $D0$ 。

- 进行四则运算时，右边的数值，全变换成4字节的二进制进行运算。

将运算结果进行数据变换成左边的参数形式代入。

《例》

若 $D0 = \#R2000 + K5000$ 、 $R2000 = KA8DE$ 时，则将右边的数值变换成
4字节长的带符号二进制。

$A8DE \rightarrow FFFFA8DE$

$5000 \rightarrow 00001388$

进行运算。

$FFFFA8DE + 00001388 = FFFFBC66$

因左边是D寄存器，将运算结果进行BCD变换。

$FFFFFBC66 \rightarrow -17306$

将变换后的数据代入 $D0$ 。

- 将四则运算结果代入D寄存器时，如果运算结果超过了数值范围，则将运算结果中最接近的设定可能值代入。

《例》

$D0 = K8388600 + K100 \rightarrow D0 = K8388607$

- 用0去除时，则运算出错，停止运算。

1 2 - 3. 定位控制命令

在 Z - 0 1 P M 上使用的定位控制命令，分为以下的 3 种：

1) 驱动控制命令

是输出脉冲控制驱动系统的命令。

代码	功 能	解 說
G 0 0	定位	梯形控制的定位。
G 0 5	连续定位	无停止的连续定位。
G 2 0	定速送进定位 1	按控制速度动作。中断输入后，按同一速度定位。
G 2 1	定速送进定位 2	按控制速度动作。中断输入后，按不同速度定位。
G 2 2	定速送进定位 3	按控制速度动作。中断输入后，减速停止。
G 2 5	带中断连续定位	是与 G 0 5 同等的定位。以完了为条件可使用中断输入。
G 2 6	带中断定位	是与 G 0 0 同等的定位。以完了为条件可使用中断输入。
G 2 8	原点检索	原点检索。

2) 程序控制命令

是控制程序的执行状态的命令。

代码	功 能	解 說
G 0 4	静止计时器	等待程序执行的计时器。
(-3) G 1 0	中断 1	中断信号输入后，使以在执行的定位动作中断而执行别的程序。
(-3) G 1 4	中断 2	中断信号输入后，使以在执行的定位动作中断而执行别的程序。 在别的程序执行完后，从被中断的定位的下一块起执行。
G 6 0	条件跳转	条件成立时，跳转到指定的标号号码。
G 6 1	条件等待	等待程序执行到条件成立为止。
G 7 0	子程序调用	调用子程序
G 7 2	子程序的开始	定义子程序的开始。
G 7 4	子程序结束	定义子程序的结束。
G 7 5	无条件跳转	跳转到指定的标号号码。
PAUSE	暂停	暂停程序的执行。
E N D	结束	结束程序的执行。

3) 其他

参数变更或目标值的给与方法等诸设定。

代码	功 能	解 說
G 3 0	加速时间的设定	设定加速时间
G 3 1	减速时间的设定	设定减速时间
(-3) G 3 2	S 字形加减速补偿值	S 字形加减速补偿值。
G 6 3	整块传送	PLC \rightleftarrows Z - 0 1 P M 之间数据传送
G 9 0	绝对值指令	本命令以后的位置指令值为绝对值
G 9 1	相对值指令	本命令以后的位置指令值为相对值

● G 0 0：定位

在指定的座标按指定的速度移动。

G 0 0	①	X xxxxxxxx	②	F xxxxxxxx
		x		x

- ①目标座标值 / 移动量： [直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

当为绝对值指令时，则记述目标的座标值。

当为相对值指令时，则记述目标的移动量。

在间接指定时，记述的寄存器的内容为设定值。

- ②送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 0 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

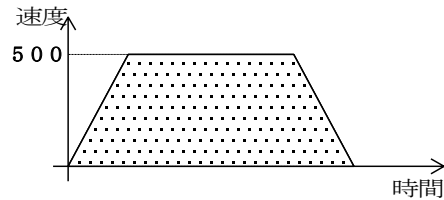
记述送进速度。

在间接指定时，记述的寄存器的内容为设定值。

《程序例》

从现在位置向座标（3 0 0），以送进速度（D 1 5：0 0 0 0 0 5 0 0）进行定位。

G 0 0 X 3 0 0 F (D 1 5)



● G 0 5：连续定位

以指定的送进速度移动到目标值，不减速停止进行下一个定位。

G 0 5	①	X xxxxxxx	②	F xxxxxxx
		x		x

- ①目标座标值 / 移动量： [直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

当为绝对值指令时，记述目标的座标值。

当为相对值指令时，到过目标的移动量。

在间接指定时，记述的寄存器的内容为设定值。

- ②送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述送进速度。

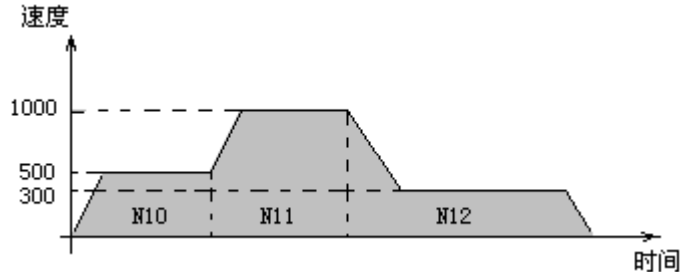
在间接指定时，记述的寄存器的内容为设定值。

《程序例》

从现在位置向座标 (3 0 0)，以送进速度 (D 1 5 : 0 0 0 0 0 5 0 0) 进行定位，并以送进速度 (1 0 0 0) 到座标 (1 0 0 0) 进行定位。

以送进速度座标 (3 0 0) 到达座标 (D 2 0 : 0 0 0 0 2 0 0 0) 定位后停止。

```
N 1 0 G 0 5 X 3 0 0 F ( D 1 5 )
N 1 1 G 0 5 X 1 0 0 0 F 1 0 0 0
N 1 2 G 0 0 X ( D 2 0 ) F 3 0 0
```



《注意事項》

1) 连续定位的移动方向要同一方向。

2) 本命令下面的行里，只能记述 G 0 0、G 0 5、G 2 5、G 2 6 和辅助码输出命令 5 种。

本命令的下一行里记述了辅助码时，在下面的行里，只能记述 G 0 0、G 0 5、G 2 5、G 2 6 4 种命令。

```
G 0 5 X 5 0 0 F 5 0 0
M 1 0 0
G 2 5 X 1 0 0 0 F 7 0 0
G 0 0 X 2 0 0 0 F 1 0 0 0
```

● G 2 0：定速送进定位 1

在中断信号输入前，一直以设定的速度移动。
 中断信号输入后，移动指定的量后停止。

G 2 0	①	②
	X xxxxxxxx x	F xxxxxxxx x

①目标移动量： [直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述到达目标的移动量。
 在间接指定时，记述的寄存器的内容为设定值。
 设定值作为相对值处理。

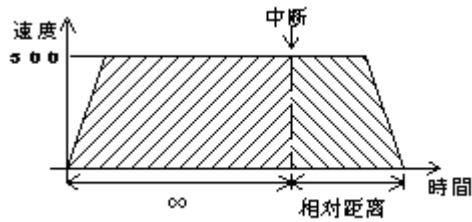
②送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述送进速度。
 在间接指定时，记述的寄存器的内容为设定值。

《程序例》

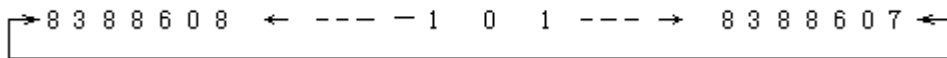
以速度（5 0 0）移动，中断输入后，移动（D 5 3： 0 0 0 0 1 0 0 0）的量后停止。

```
G 2 0 X ( D 5 3 ) F 5 0 0
```



《注意事項》

- 1) 中断信号，无论外部中断、内部中断都有效。
- 2) 在没有使坐标系无效的情况下，进行超过座标值范围的定位时，座标值作如下变化。



● G 2 1：定速送进定位 2

在中断信号输入前，一直以设定速度移动。

中断信号输入后，变更为指定的速度，移动指定的量后停止。

G 2 1	① X xxxxxxxx x	② F xxxxxxxx x	③ F xxxxxxxx x
-------	----------------------	----------------------	----------------------

- ①目标移动量： [直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述到达目标的移动量。

在間接指定时，記述的寄存器的内容为设定值。

设定值作为相对值处理。

- ②送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 0 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述送进速度。

在間接指定时，記述的寄存器的内容为设定值。

- ③送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 0 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

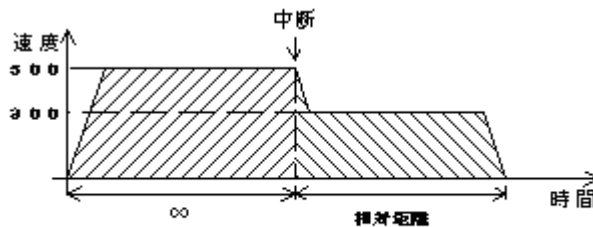
记述中断输入后的送进速度。

在間接指定时，記述的寄存器的内容为设定值。

《程序例》

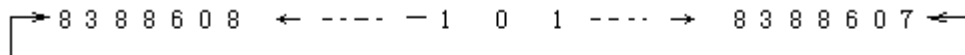
以速度（5 0 0）移动，中断输入后，以速度（3 0 0）移动（D 5 3：0 0 0 0 1 0 0 0）的量后停止。

G 2 1 X (D 5 3) F 5 0 0 F 3 0 0



《注意事項》

- 1) 中断信号，无论外部中断、内部中断都有效。
- 2) 在没有使坐标系无效的情况下，进行超过坐标值范围的定位时，坐标值作如下变化。



— 驱动控制命令 —

● G 2 2：定速送进定位 3

在中断信号输入前，一直以设定的速度在设定的方向上移动。
中断信号输入后，则减速停止。

(-1)

G 2 2	① X	② F	xxxxxxx
	±		x

①送进方向： X + 在 C W 方向移动
X - 在 C C W 方向移动

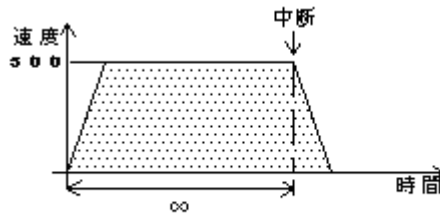
②送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 0 (B C D)
[間接指定] xxxxxxxx = D 0 ~ D 1 2 7

記述送进速度。
在間接指定时，記述的寄存器的内容为设定值。

《程序例》

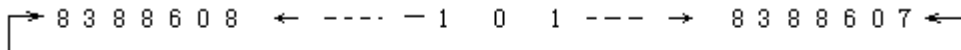
以速度（ 5 0 0 ）移动，中断输入后则减速停止。

G22 X+ F500



《注意事項》

- 1) 中断信号，无论外部中断、内部中断都有效。
- 2) 在没有使坐标系无效的情况下，进行超过座标值范围的定位时，座标值作如下变化。



— 驱动控制命令 —

● G 2 3：恒线速度定位

是绕线控制专用的定位。
在绕线控制中，能够控制得使绕线速度一定。

G 2 3	①	X xxxxxxx	②	F xxxxxxx
		x		x

①目标移动量： [直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (B C D)
[間接指定] xxxxxxxx = D 0 ~ D 1 2 7

記述到达目标的移动量（总脉冲数）。
在間接指定时，記述的寄存器的内容为设定值。
设定值作为相对值处理。

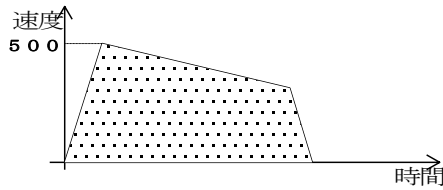
②送进速度： [直接指定] xxxxxxxx = 1 ~ 1 0 0 0 0 (B C D)
[間接指定] xxxxxxxx = D 0 ~ D 1 2 7

記述最初的送进速度。
在間接指定时，記述的寄存器的内容为设定值。

(-1) 这里设定的速度单位为 mm / s 。

《程序例》

以初速（500）移动（1000）后停止。



《注意事項》

- 1) 要设定绕线控制用的参数。
- 2) 在使用本命令的程序里，电子齿轮固定为 1：1。
- 3) 在同一程序中，不可记述除 G 2 3 以外的驱运控制命令。

```

N 1 0  G 2 3  X 1 0 0 0  F 5 0 0 )
          G 0 4  K 1 0 0
N 1 1  G 2 3  X 1 0 0 0  F 3 0 0 )  O K
          G 2 3  X 2 0 0  F 5 0
          E N D

N 1 0  G 2 3  X 1 0 0 0  F 5 0 0 )
          G 0 4  K 1 0 0
N 1 1  G 2 3  X 1 0 0 0  F 3 0 0 )  N G ( 出 错 )
          G 0 0  X 2 0 0  F 5 0
          E N D
    
```


● G 2 5：带中断连续定位

基本动作与 G 0 5 相同。

在到达目标值之前中断信号输入后，则移行到下面的定位。

G 2 5	①	②
	X xxxxxxxx x	F xxxxxxxx x

①目标座标值 / 移动量：[直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (BCD)
[間接指定] xxxxxxxx = D 0 ~ D 1 2 7

当为绝对值指令时，记述目标的座标值。

当为相对值指令时，记述目标的移动量。

在間接指定时，记述的寄存器的内容为设定值。

②送进速度：[直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 (BCD)
[間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述送进速度。

在間接指定时，记述的寄存器的内容为设定值。

《程序例》

在执行下述程序时，动作因中断信号的有无而异。

```
N 1 0  G 2 5  X 5 0 0  F 5 0 0
N 1 1  G 2 5  X 1 0 0 0  F 3 0 0
N 1 2  G 2 6  X 1 5 0 0  F 2 0 0
```

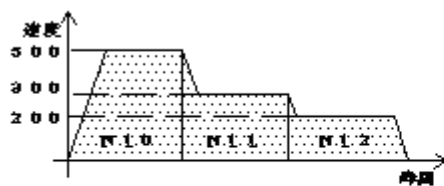


图 1：没有中断信号输入的情况

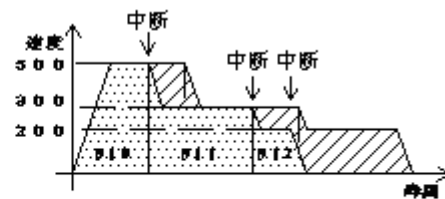


图 2：有中断信号输入的情况

《注意事项》

1) 连续定位的移动方向要同一方向。

2) 在本命令下面的行里，只能记述 G 0 0、G 0 5、G 2 5、G 2 6 和辅助码输出命令五种命令。

在本命令的下一行里，记述了辅助码时，在下面的行里，只能记述 G 0 0、G 0 5、G 2 5、G 2 6 4 种命令。

```
G 2 5  X 5 0 0  F 5 0 0
M 1 0 0
G 2 5  X 1 0 0 0  F 7 0 0
G 0 0  X 2 0 0 0  F 1 0 0 0
```

3) 中断信号，无论外部中断、内部中断都有效。

● G 2 6：带中断定位

基本动作与 G 0 0 相同。

在到达目标值之前中断信号输入后，即减速停止。

G 2 6	①	X xxxxxxx	②	F xxxxxxx
		x		x

①目标座标值 / 移动量： [直接指定] xxxxxxxx = - 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7 (B C D)
 [間接指定] xxxxxxxx = D 0 ~ D 1 2 7

当为绝对值指令时，记述目标的座标值。

当为相对值指令时，记述目标的移动量。

在間接指定时，记述的寄存器的内容为设定值。

②送进速度： [直接指定] xxxxxxxx = 1 ~ 2 0 0 0 0 0 0 0 (B C D)

[間接指定] xxxxxxxx = D 0 ~ D 1 2 7

记述送进速度。

在間接指定时，记述的寄存器的内容为设定值。

《程序例》

在执行下述程序时，动作因中断信号的有无而异。

G 2 6 X 2 0 0 0 F 5 0 0

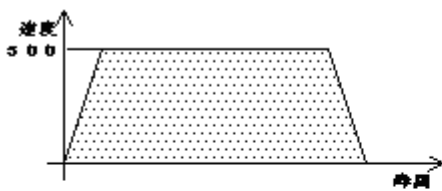


图 1：没有中断信号输入的情况

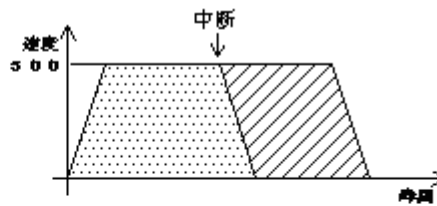


图 2：有中断信号输入的情况

《注意事项》

- 1) 中断信号，无论外部中断、内部中断都有效。

—驱动控制命令—

● G 2 8：原点检索

是进行原点检索的命令。

本命令实行后，即按照参数开始原点检索。

G 2 8

《注意事項》

- 1) 原点检索的详情，请参阅 6-3. 原点检索。

—程序控制命令—

● G 0 4：静止时间

按指定的时间停止执行程序。

G 0 4	① K xxx
	x

① 等待時間：[直接指定] xxxx = 0 ~ 9 9 9 9 (BCD)

[間接指定] xxxx = D 0 ~ D 1 2 7

当为間接指定时，記述的寄存器的内容为设定值。

寄存器的内容超过 9 9 9 9 时，作为 9 9 9 9。

单位为 [× 1 0 m s]。

当设定为「0」时，则此命令无效。

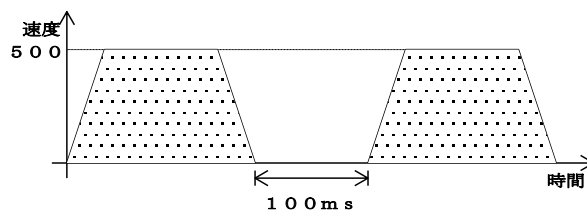
《程序例》

移动到座标 (1 0 0 0) 后，停止 (1 0 0 m s) 再移动到座标 (2 0 0 0)。

```

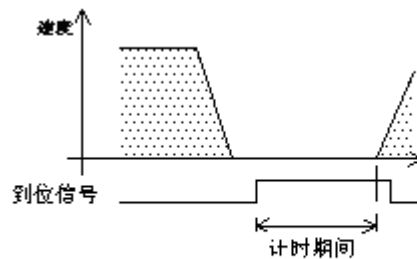
N 1 0  G 0 0  X 1 0 0 0  F 5 0 0
          G 0 4  K 1 0
N 1 1  G 0 0  X 2 0 0 0  F 5 0 0

```



《注意事項》

- 1) 静止时间的计时开始，在到位信号的检查完毕后进行。



—程序控制命令—

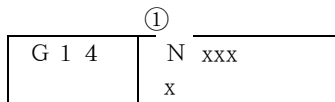
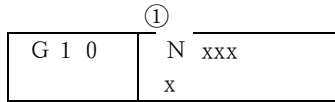
(-3)

- G 1 0：中断 1
- G 1 4：中断 2

将执行中的定位中途停止，执行别的处理（中断处理）。

在中断控制命令中，不同中断形式（中断 1：跳转型 / 中断 2：调用型）存在 2 种命令。

对内部中断输入、外部中断输入都适应。



①标号编号：[直接指定] xxxx = 0 ~ 9 9 9 9 (BCD)

[間接指定] xxxx = D 0 ~ D 1 2 7

记述跳转对象处的标号号码。

标号号码可在 0 ~ 9 9 9 8 范围内指定。而由 9 9 9 9 解除中断许可。

在間接指定时，記述的寄存器的内容为设定值。

寄存器的内容，超过 9 9 9 9 时，作为 9 9 9 9 (中断许可解除)。

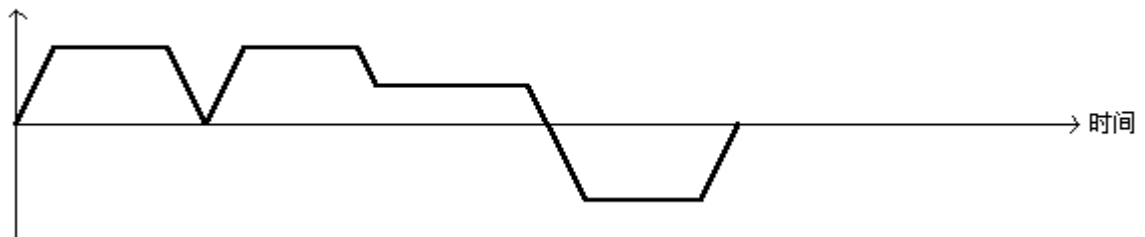
《程序例》

```

G 1 * N 1 0                中断处理的定义
N 1  G 0 0 X 1 0 0 F 1 0 0
N 2  G 0 5 X 2 0 0 F 1 0 0 ←在这里加上中断
      G 0 0 X 2 5 0 F 5 0
N 3  G 0 0 X 1 5 0 F 1 0 0
      E N D
N 1 0 G 7 2                中断处理程序
      G 0 0 X 0      F 1 0 0
      G 7 4
    
```

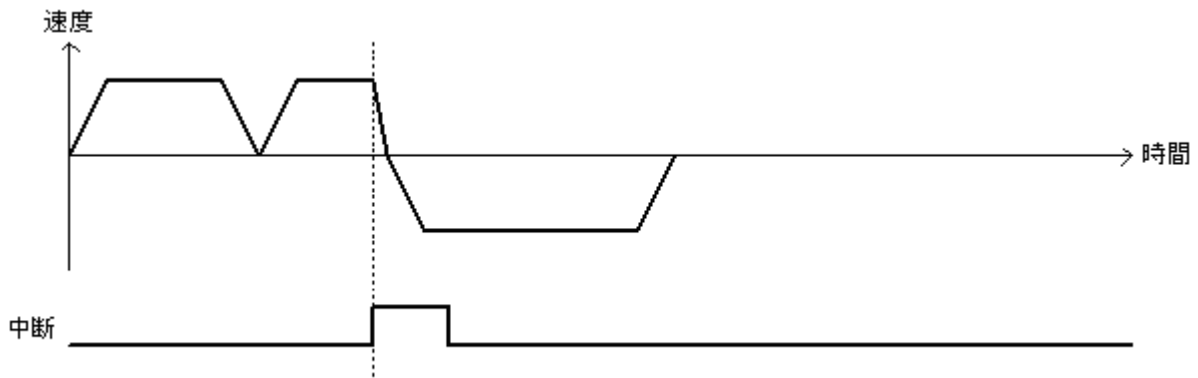
· 中断未发生的情况

速度



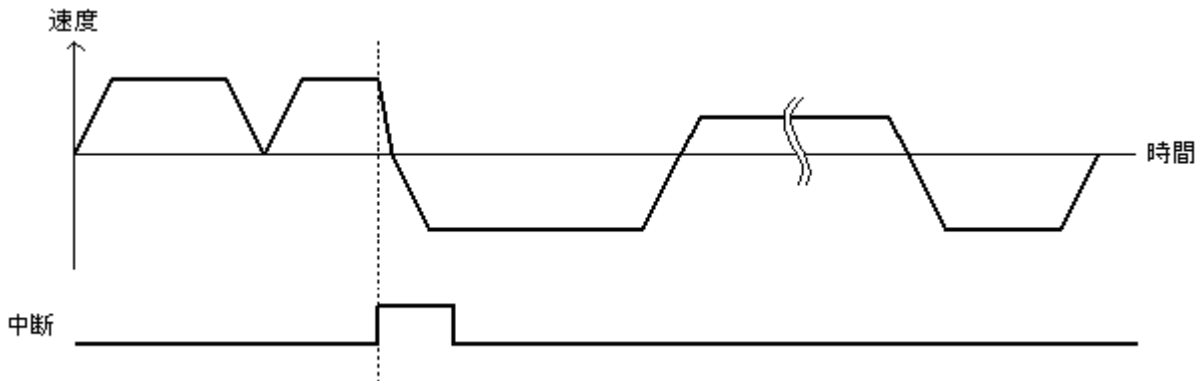
· 中断 1 发生的情况

中断发生后，即按紧急停止时间进行紧急停止，执行中断处理程序。



· 中断 2 发生的情况

中断发生后，即按紧急停止时间进行紧急停止，执行中断处理程序。而且执行后，从发生中断的定位（N 2）的下一个定位动作起再开始运转。



《注意事项》

- 1) 中断 1 和 2，只能其中之一有效。而且在有效期内，不可使用 G 2 5、G 2 6。
- 2) 如下面的程序那样，可以不构成子程序来使用。这种场合，中断 1 和 2 都在 N 1 0 执行后，由 E N D 来结束程序。

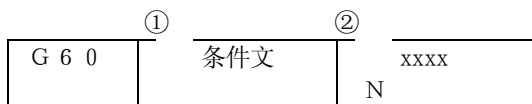
```

                G 1 *  N 1 0                中断处理的定义
N 1      G 0 0  X 1 0 0  F 1 0 0
N 2      G 0 5  X 2 0 0  F 1 0 0  ←在这里加上中断
                G 0 0  X 2 5 0  F 5 0
N 3      G 0 0  X 1 5 0  F 1 0 0
                G 7 5  N 9 9
N 1 0    G 0 0  X 0      F 1 0 0
N 9 9    E N D
    
```

—程序控制命令—

● G 6 0：条件跳转

在记述的条件成立时，跳转到指定的标号号码。



①条件文：记述条件。

②Nxxxx：[直接指定] xxxx = 0 ~ 9 9 9 9 (B C D)
[间接指定] xxxx = D 0 ~ D 1 2 7

记述跳转目标。

在间接指定时，记述的寄存器的内容为设定值。

《条件文的记述方法》

条件文按「被比较数据、算符、比较数据」的顺序进行记述。

- D 6 3 = K 1 0 0 0 : D 6 3 的内容等于 1 0 0 0
- P 1 0 <> P 1 1 : P 1 0 的内容不等于 P 1 1 的内容
- # R 2 0 0 0 > K 1 0 : P L C 的寄存器 R 2 0 0 0 的内容大于 1 0 (B C D)
- # M 1 0 0 = K 1 : P L C 的内部继电器等于 1 (此时为ON)

被比较数据

- 数据寄存器 (D)
- 数据寄存器 (P)
- 继电器 (I . Q)
- P L C 的数据寄存器 (# R)

P L C 的继电器（# I、# Q、# M）

比较数据

数据寄存器（D）

数据寄存器（P）

常数（K—8388608～K8388607）

运算符

” = ” : 被比较数据等于比较数据
 ” > ” : 被比较数据大于比较数据
 ” < ” : 被比较数据小于比较数据
 ” >= ” 、 ” = > ” : 被比较数据大于或等于比较数据
 ” <= ” 、 ” = < ” : 被比较数据小于或等于比较数据
 ” <> ” 、 ” > < ” : 被比较数据不等于比较数据

《程序例》

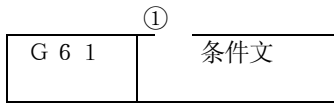
使用条件跳转命令，使程序反复执行。

```

      P 0 = K 1 0 ; 设定反复次数
      D 1 0 0 = K 1 0 0 0 ; 设定目标值
N 1 0 G 0 0 X ( D 1 0 0 ) F 1 0 0 0 ; 转到 D 1 0 0 的位置
      D 1 0 0 = D 1 0 0 + K 1 0 0 0 ; 目标值 + 1 0 0 0
      P 0 = P 0 - K 1 ; 将剩余反复次数减 1
      G 6 0 P 0 <> K 0 N 1 0 ; 如剩余反复次数不为 0，则跳转到 N 1 0
      E N D ; 程序的结束
  
```

● G 6 1：条件等待

停止执行程序直到条件成立为止。



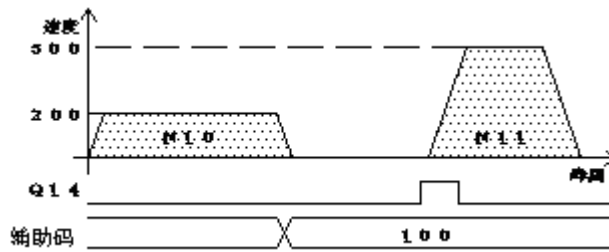
①条件文：記述条件。

《程序例》

使用条件等待命令，取得与处围设备同步动作。

```

N 1 0  G 0 0  X 5 0 0  F 2 0 0    ; 移动到目标值
        M 1 0 0                    ; 输出辅助码
        G 6 1  # Q 1 4 = K 1      ; 在 P L C 的输出继电器 Q 1 4  O N 之前，停止执行程序
                                       ; Q 1 4  O N 后，即移行到下一行
N 1 1  G 0 0  X 1 0 0 0  F 5 0 0  ; 移动到目标值
        E N D                      ; 程序的结束
    
```

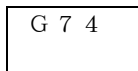
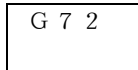
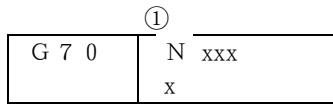


—程序控制命令—

- G 7 0：调用子程序
- G 7 2：子程序开始
- G 7 4：子程序结束

调用程序中的子程序。

子程序由 G 7 2、G 7 4 定义。



- ① Nxxxx： [直接指定] xxxx = 0 ~ 9 9 9 9 (BCD)
 [間接指定] xxxx = D 0 ~ D 1 2 7

记述跳转目标。

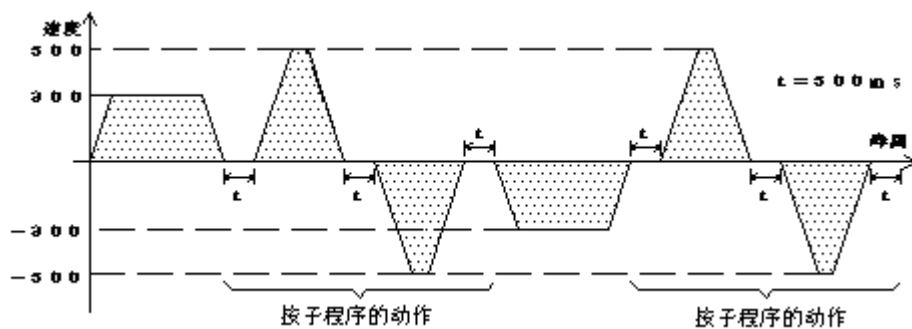
在間接指定时，記述的寄存器的内容为设定值。

《程序例》

在程序里，将进行相同动作的部分做一个子程序。

```

G 0 0 X 1 0 0 0 F 3 0 0 ; 移动到目标值
G 7 0 N 1 0 ; 调用子程序
G 0 0 X 0 F 3 0 0 ; 移动到目标值
G 7 0 N 1 0 ; 调用子程序
E N D ; 程序结束
N 1 0 G 7 2 ; 子程序开始
G 9 1 ; 由相对值指令变更
G 0 0 X 1 0 0 F 5 0 0 ; 移动规定移动量
G 0 4 K 5 0 ; 停止500ms
G 0 0 X - 1 0 0 F 5 0 0 ; 移动规定移动量
G 9 0 ; 由绝对值指令变更
G 7 4 ; 子程序结束
    
```



《注意事項》

- 1) 不可从子程序中调用子程序。
- (-1) 2) 在子程序内也可使用跳转命令。
但如果从子程序内，跳转到主程序后「END」时，作为出错。
- 3) 子程序要记述在END命令后面。

● G 7 5：無条件跳转

无条件跳转到指定的标号号码。

G 7 5	① N xxx
	x

- ① Nxxxx: [直接指定] xxxxx = 0 ~ 9 9 9 9 (BCD)
- [間接指定] xxxxx = D 0 ~ D 1 2 7

记述跳转目标。

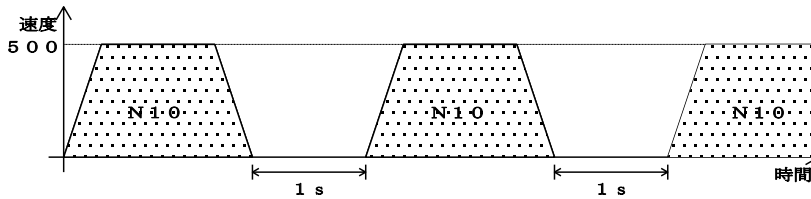
在間接指定时，記述的寄存器的内容为设定值。

《程序例》

反复进行移动 1 0 0 0， 停止 1 秒钟的动作。

```

G 9 1 ; 设定相对值指令
N 1 0 G 0 0 X 1 0 0 0 F 5 0 0 ; 移动指定的移动量
G 0 4 K 1 0 0 ; 停止 1 秒間
G 7 5 N 1 0 ; 跳转到 N 1 0
E N D ; 程序结束
    
```



《注意事項》

- (-1) 1) 在子程序内可以使用跳转命令，但如在「G 7 4：返回之前「E N D」时，作为出错。

《会出错的程序例》

```

G 0 0 X 1 0 0 0 F 3 0 0 ; 移动到目标值
G 7 0 N 1 0 ; 调用子程序
G 0 0 X 0 F 3 0 0 ; 移动到目标值
G 7 0 N 1 0 ; 调用子程序
N 8 G 0 0 X 1 0 0 0 F 3 0 0 ; 移动到目标值
E N D ; 程序结束
N 1 0 G 7 2 ; 子程序开始
G 9 1 ; 相对值指令变更
G 0 0 X 1 0 0 F 5 0 0 ; 移动指定的移动量
G 0 4 K 5 0 ; 停止 5 0 0 m s
G 0 0 X - 1 0 0 F 5 0 0 ; 移动指定的移动量
G 9 0 ; 绝对值指令变更
G 7 5 N 8 ; 向 N 8 跳转
G 7 4 ; 子程序结束
    
```

● P A U S E：暂停

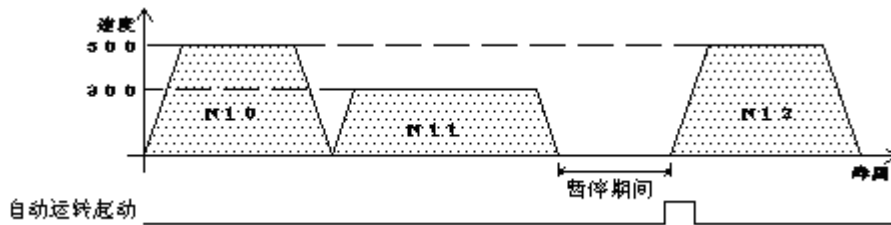
暂停程序的执行。
执行本命令后，通过自动运转起动 O N，从下一行起开始动作。

P A U S E

《程序例》

使程序在中途停止。

```
N 1 0  G 0 5  X 1 0 0 0  F 5 0 0      ; 向目标值移动
N 1 1  G 0 0  X 2 0 0 0  F 3 0 0      ; 向目标值移动
        P A U S E                      ; 暂停
N 1 2  G 0 5  X 3 0 0 0  F 5 0 0      ; 向目标值移动
        E N D                          ; 程序结束
```



—程序控制命令—

● E N D：结束

是结束执行程序的命令。

E N D

《注意事項》

- 1) 每个程序必需有此命令。
若没有本命令时，则为出错。

—其他—

(-3)

- G 3 0：加速時間的設定
- G 3 1：減速時間的設定
- G 3 2：S形加減速補償值的設定

临时改变加速时间，减速时间，S形加减速补偿值的参数设定值。

①	
G 3 0	K xxx x

①	
G 3 1	K xxx x

- ① Nxxxx: [直接指定] xxxx = 0 ~ 9 9 9 9 (B C D)
[間接指定] xxxx = D 0 ~ D 1 2 7
記述加速 / 減速時間。
在間接指定时，記述的寄存器的内容为设定值。
单位为 [× 1 0 m s]。

②	
G 3 2	K Xxx x

- ② Nxxxx: [直接指定] xxxx = 0 ~ 0 1 0 0 (B C D)
[間接指定] xxxx = D 0 ~ D 1 2 7
記述S形加減速補償值。
在間接指定时，記述的寄存器的内容为设定值。
单位为 [%]。

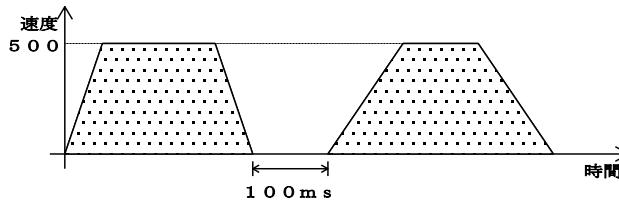
《程序例》

在程序中途，改变加速 / 減速時間。

```

N 1 0  G 0 0  X 1 0 0 0  F 5 0 0
        G 0 4  K 1 0
        G 3 0  K 5 0
        G 3 1  K 5 0
N 1 1  G 0 0  X 2 0 0 0  F 5 0 0
        E N D

```



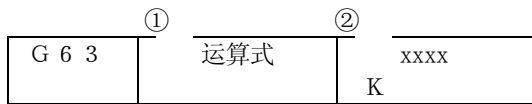
《注意事項》

- 1) 如执行本命令，则命令执行后所有动作的加速 / 減速時間全部变更。
- 2) 在下面的情况下，由本命令设定的值无效。
 - 执行了 E N D 命令时
 - 暂停后，执行了工序复位时
 - 紧急停止时

—其他—

● G 6 3：数据的整块传送

在数据寄存器（D）与 P L C 的数据寄存器之间进行数据传送（# R）。



①运算式：記述代入式。

②K xxxx：[直接指定] xxxx = 1 ~ 1 6（B C D）
記述要传送的数据数。

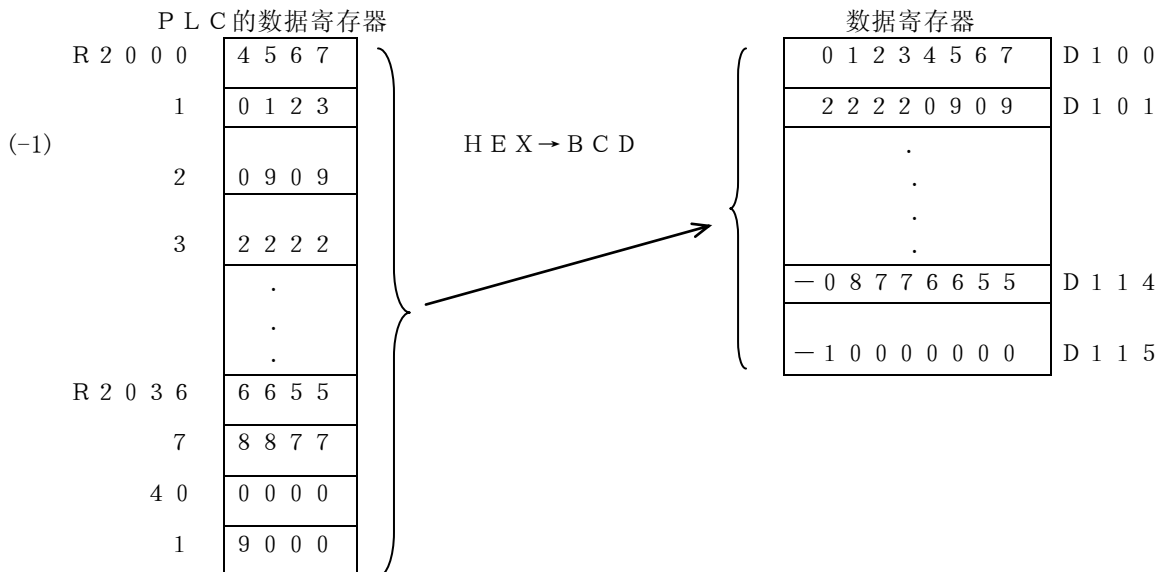
《运算式的記述方法》

运算式按「传送原寄存器号、=、传送对象寄存器号」的顺序进行记述。

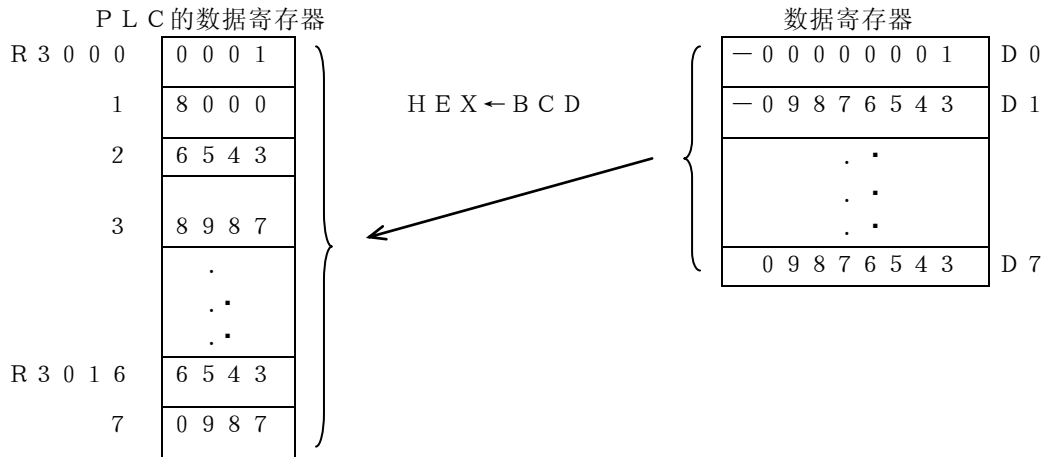
- D 1 0 0 = # R 2 0 0 0 : 将 P L C 的寄存器 R 2 0 0 0 向 D 1 0 0 传送
 # R 3 0 0 0 = D 0 : 将数据寄存器 D 0 向 P L C 的寄存器 R 3 0 0 0 传送

《程序例》

G 6 3 D 1 0 0 = # R 2 0 0 0 K 1 6



G 6 3 # R 3 0 0 0 = D 0 K 8



—其他—

- G 9 0：绝对值指令
- G 9 1：相对值指令

是设定目标值给予方法的命令。
 如执行 G 9 0，则在以后的定位中，设定值作为座标值处理。
 如执行 G 9 1，则在以后的定位中，设定值作为移动量处理。

G 9 0

G 9 1

《程序例》

将进行相同动作的程序，用绝对值指令和相对值指令记述。

—绝对值指令时—

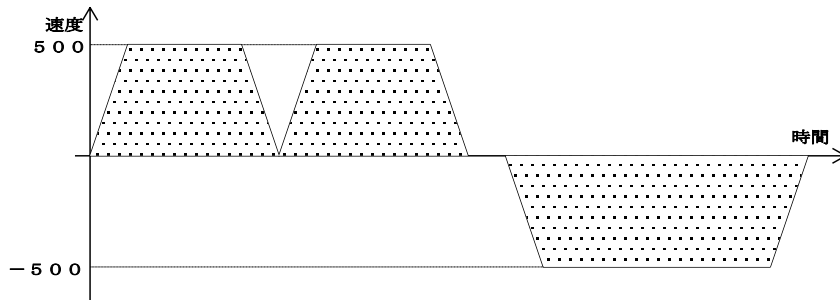
```

G 9 0
N 1 0 G 0 0 X 1 0 0 0 F 5 0 0
N 1 1 G 0 0 X 2 0 0 0 F 3 0 0
N 1 2 G 0 0 X 0 F 5 0 0
END
    
```

—相对值指令时—

```

G 9 1
N 1 0 G 0 0 X 1 0 0 0 F 5 0 0
N 1 1 G 0 0 X 1 0 0 0 F 3 0 0
N 1 2 G 0 0 X - 2 0 0 0 F 5 0 0
END
    
```



《注意事项》

- 1) 在程序起动时，当没有记述任何命令时，则：
 - 有坐标系设定时，为绝对值指令
 - 无坐标系设定时，为相对值指令

- 2) 在以下场合再起动时，停止前的状态无效。
 - 执行了 E N D 命令时
 - 暂停后，执行了工序复位时
 - 紧急停止时

1 2 - 4 . 程序编制

编制程序时，要遵守如下规则：

- • 1 行 1 个命令。在 1 行里不可记述 2 个或 2 个以上命令。
- • 命令码，各操作数，要留 1 个空格分开。

使用专用计算机工具来编制程序，传送给 Z - 0 1 P M。

《程序例》

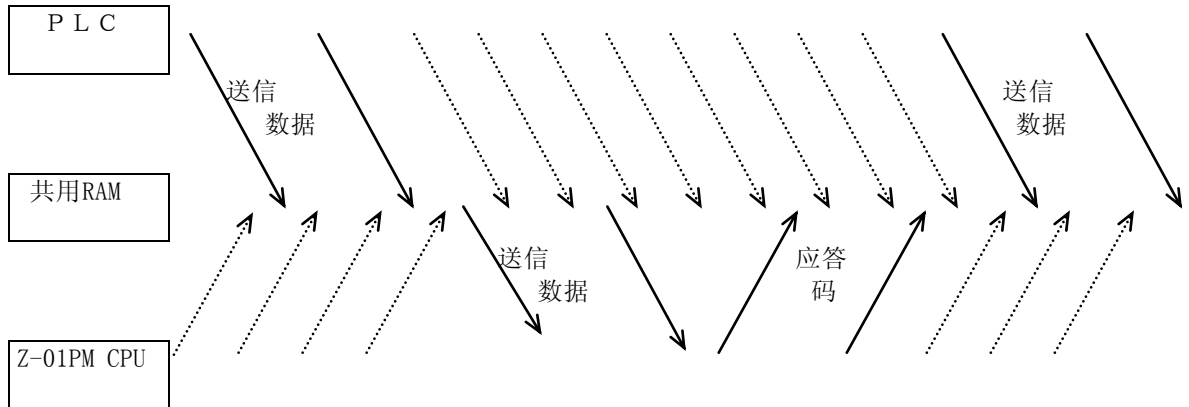
```
G28
D0=K10
N10 G00 X1000 F1000
M100
G61 Q12=K1
N11 G05 X2000 F500
G05 X3000 F200
G00 X4000 F100
G91
N12 G00 X1000 F1000
G04 K10
G60 D0<>K0 N12
G90
N13 G00 X0 F2000
END
```

1 2 - 5 . 程序的传送

在 Z - 0 1 P M 上传送程序, 原则上以使用计算机工具为前提, 但如果按照数据传送的顺序, 用 WT 命令也可传送。

1 2 - 5 - 1 . 程序传送的模式

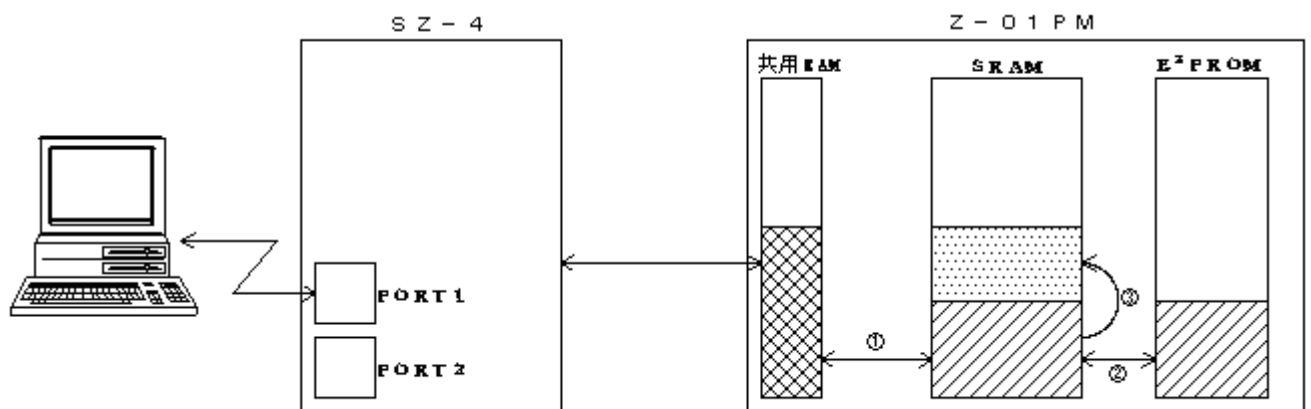
程序的传送与其他数据不同, 由一次数据传送有可能完不了。因此, 程序的传送以如下模式进行。



从 P L C 对 Z - 0 1 P M 的共用 R A M 写入送信数据。
 Z - 0 1 P M 的 C P U , 查询共用 R A M 的命令区, 对命令是否已写入进行常时监视。(虚线)
 数据写入共用 R A M 后, Z - 0 1 P M 的 C P U , 将数据传送到内部。
 数据传送完毕后, 将应答码返回共用 R A M 的命令区。
 P L C 常时监视命令区, 等待应答码返回。
 应答码返回后, 发送下面的数据。

1 2 - 5 - 2 . 程序的存储

程序的传送用 ASCII 码进行。
 从 P L C 传送的数据, 保存在 S R A M 里 (不能直接传送给 E² P R O M)。(①) 存储在 S R A M 里的程序, 有能直接执行, 还必须传送到 E E P R O M 里并进行保存。
 程序的执行, 直接从 E E P R O M 读出该程序来进行 (P 2 5 参照), 要从 S R A M 保存到 E E P R O M , 则要执行其它命令 (②)。



1 2 - 5 - 3 . 程序的传送 (→ Z - 0 1 P M)

程序由ASCII码传送。

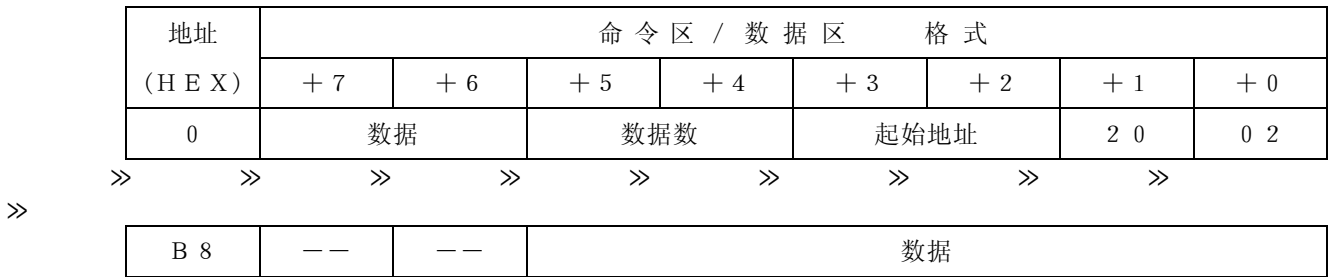
参数之间用SPACE(20h)断开、行的结束用NULL(00h)程序文件的结束用SUB(1Ah)。

《例》

```
G28                ;47 32 38 00
D0=10              ;44 30 3D 31 30 00
N10 G00 X1000 F1000 ;4E 31 30 20 47 30 30 20 58 31 30 30 30 20 46 31 30 30 30 00
M100               ;4D 31 30 30 00
G61 Q12=1          ;47 36 31 20 51 31 32 3D 31 00
N11 G05 X2000 F500 ;4E 31 31 20 47 30 35 20 58 32 30 30 30 20 46 35 30 30 00
G05 X3000 F200     ;47 30 35 20 58 33 30 30 30 20 46 32 30 30 00
G00 X4000 F100     ;47 30 30 20 58 34 30 30 30 20 46 31 30 30 00
G91                ;47 39 31 00
N12 G00 X1000 F1000 ;4E 31 32 20 47 30 30 20 58 31 30 30 30 20 46 31 30 30 30 00
G04 K10            ;47 30 34 20 4B 31 30 00
G60 D0<>0 N12      ;47 36 30 20 44 30 3C 3E 30 20 4E 31 32 00
G90                ;47 39 30 00
N13 G00 X0 F2000   ;4E 31 33 20 47 30 30 20 58 30 20 46 32 30 30 30 00
END                ;45 4E 44 00 1A
```

程序传送的命令如下：

2 0 0 2 + [起始地址] + [数据数] + [数据]



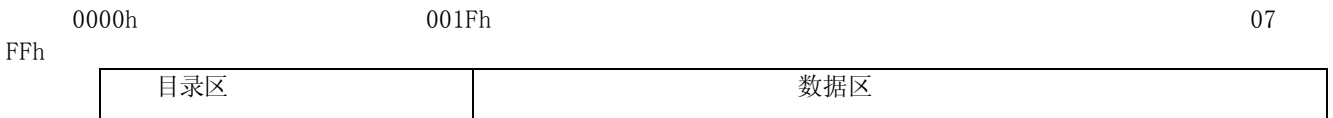
起始地址：指定数据传送的起始地址（SRAM）设定范围为0~77FF（HEX）。

数据数：设定传送数据的字节数。命令、起始地址、数据数的设定值除外。

设定范围为1~B8（HEX）。

数据：设定数据。数据长可变，最大184字节。

保存程序的SRAM的区域如下：



《目录区》



- • 编号（2字节长：BIN形式）
存储程序编号。
设定范围：0～9
- • 予約
使用Z-01PM，写入时设定为0。
（存储在E²PRO M里时，写入开始地址）
- • 大小（2字节长：BIN形式）
存储程序大小（字节数）。
设定范围：1～7E0
- • 预备
是预备领域，写入时设定为0。
- • 注解区
在注解区里，写入对程序的注解。
Z-01PM可使用2种格式。

①存储MS-DOS文件信息的格式（从计算机传送过来的）

+00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17h
00	文件名								2E	扩展符	00	TIME	DATE	预备									

- +0字节 : 设定NULL（00h）。
- +1～8 : 将文件名靠左侧设定。
在空白的部分里，设定为SPACE（20h）。
- +9 : 设定周期”。”（2Eh）。
- +A～C : 靠左侧设定文件的扩展符。
- +D : 设定NULL（00h）。
- +E～F : 设定传送原文件的时间数据（MS-DOS形式）。
- +10～11 : 设定传送原文件的日期数据（MS-DOS形式）。
- +12～17 : 预备领域。设定NULL（00h）。

②注解文存储的格式（用WT命令从PLC传送时）

+00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17h
注解文													00	预备									

- +0～13 : 靠左侧设定注解文。
- +14 : 设定NULL（00h）。
- +15～17 : 预备领域。设定NULL（00h）。

※由先头字节是不是NULL（00h）来判断注解区的格式是哪一种。

《数据区》

程序用ASCII码存储。

《传送例》

将12-5-3的程序向Z-01PM传送。

因1次传送不了，分2次传送。

文件名：TEST1.PRG

時間：15時31分48秒

日期：95年10月31日

程序号：1

第1次传送，传送到第9行（G91）。

数据如下：

	下位 →→→→→ 上位	
02 20		;命令
00 00		;起始地址
82 00		;传送字节数
01 00		;程序号码
00 00		;预备
A8 00		;程序大小
00 00		;预备
00 54 45 53 54 31 20 20 20 2E 50 52 47 00		;文件名(TEST1.PRG)
F8 7B		;時間
5F 1F		;日期
00 00 00 00 00 00		;预备
47 32 38 00		;G28
44 30 3D 31 30 00		;D0=10
4E 31 30 20 47 30 30 20 58 31 30 30 30 20 46 31 30 30 30 00		;N10 G00 X1000 F1000
4D 31 30 30 00		;M100
47 36 31 20 51 31 32 3D 31 00		;G61 Q12=1
4E 31 31 20 47 30 35 20 58 32 30 30 30 20 46 35 30 30 00		;N11 G05 X2000 F500
47 30 35 20 58 33 30 30 30 20 46 32 30 30 00		;G05 X3000 F200
47 30 30 20 58 34 30 30 30 20 46 31 30 30 00		;G00 X4000 F100
47 39 31 00		;G91

} 程序

传送到共用 R A M里，情况如下：

地址 (H E X)	命令区 / 数据区 格式							
	+ 7	+ 6	+ 5	+ 4	+ 3	+ 2	+ 1	+ 0
0	0 0	0 1	0 0	8 2	0 0	0 0	2 0	0 2
8	5 4	0 0	0 0	0 0	0 0	A 8	0 0	0 0
1 0	2 E	2 0	2 0	2 0	3 1	5 4	5 3	4 5
1 8	1 F	5 F	7 B	F 8	0 0	4 7	5 2	5 0
2 0	3 2	4 7	0 0	0 0	0 0	0 0	0 0	0 0
2 8	0 0	3 0	3 1	3 D	3 0	4 4	0 0	3 8
3 0	2 0	3 0	3 0	4 7	2 0	3 0	3 1	4 E
3 8	3 1	4 6	2 0	3 0	3 0	3 0	3 1	5 8
4 0	3 0	3 0	3 1	4 D	0 0	3 0	3 0	3 0
4 8	3 2	3 1	5 1	2 0	3 1	3 6	4 7	0 0
5 0	4 7	2 0	3 1	3 1	4 E	0 0	3 1	3 D
5 8	3 0	3 0	3 0	3 2	5 8	2 0	3 5	3 0
6 0	3 0	4 7	0 0	3 0	3 0	3 5	4 6	2 0
6 8	2 0	3 0	3 0	3 0	3 3	5 8	2 0	3 5
7 0	3 0	3 0	4 7	0 0	3 0	3 0	3 2	4 6
7 8	4 6	2 0	3 0	3 0	3 0	3 4	5 8	2 0
8 0	0 0	3 1	3 9	4 7	0 0	3 0	3 0	3 1

Z-01PM确认在命令区里已设定好命令后，将数据传送到SRAM里。
 传送完毕后，在命令区里设定完成码[5555]。
 PLC（或计算机）确认在命令区里完成码已设定好后，开始传送第2次的数据。

在第2次传送中，传送剩余部分。
 数据如下：

下位 →→→→→ 上位

```

02 20 ;命令
82 00 ;起始地址
44 00 ;传送字节数
4E 31 32 20 47 30 30 20 58 31 30 30 30 20 46 31 30 30 30 00 ;N12 G00 X1000 F1000
47 30 34 20 4B 31 30 00 ;G04 K10
47 36 30 20 44 30 3C 3E 30 20 4E 31 32 00 ;G60 D0<>0 N12
47 39 30 00 ;G90
4E 31 33 20 47 30 30 20 58 30 20 46 32 30 30 30 00 ;N13 G00 X0 F2000
45 4E 44 00 1A ;END
    
```

传送到共用RAM里，情况如下：

地址 (HEX)	命令区 / 数据区 格式							
	+7	+6	+5	+4	+3	+2	+1	+0
0	3 1	4 E	0 0	4 4	0 0	8 2	2 0	0 2
8	3 1	5 8	2 0	3 0	3 0	4 7	2 0	3 2
1 0	3 0	3 0	3 1	4 6	2 0	3 0	3 0	3 0
1 8	3 1	4 B	2 0	3 4	3 0	4 7	0 0	3 0
2 0	3 0	4 4	2 0	3 0	3 6	4 7	0 0	3 0
2 8	0 0	3 2	3 1	4 E	2 0	3 0	3 E	3 C
3 0	2 0	3 3	3 1	4 E	0 0	3 0	3 9	4 7
3 8	4 6	2 0	3 0	5 8	2 0	3 0	3 0	4 7
4 0	4 4	4 E	4 5	0 0	3 0	3 0	3 0	3 2
4 8	--	--	--	--	--	--	1 A	0 0

全部程序传送后，一定要传送到EEPROM保存。
 传送到SRAM里的数据，不能直接执行。
 此外，SRAM内的数据，停电不能保持。

Z-01PM将数据设定完毕后，在命令区里设定完成码（5555h）

PLC（或计算机）确认完成码后读出数据，读出完毕后，在命令区里设定完成码（AAAAh）。

Z-01PM读出的结果，如果不是SUB码（1Ah），因还要传送数据，故再次监视命令区。

Z-01PM确认完成码后，传送剩余的数据。

第2次传送，共用RAM情况如下：

地址 (HEX)	命令区 / 数据区 格式							
	+7	+6	+5	+4	+3	+2	+1	+0
0	45	00	30	30	30	32	55	55
8	--	--	--	--	1A	00	44	4E

其后的顺序，与第1次相同。

(-3)

12-6. 文件的读出

Z-01PM将由E²PROM保存的程序用文件的形式进行管理。

通过执行命令，可以读出文件信息，因文件信息共有320字节，所以数据传送分2次进行。

计算机（或PLC）将读出的命令设定在命令区后，Z-01PM将文件信息设定在共用RAM里。

第1次传送中，共用RAM的情况如下：

```

0 ~ 1 h:    命令区（完成码）
2 ~ 21 h:   程序号为0的文件信息
22 ~ 41 h:   "    1    "
42 ~ 61 h:   "    2    "
62 ~ 81 h:   "    3    "
82 ~ A1 h:   "    4    "
```

计算机（或PLC）确认完成码（或数据设定就绪）后，读出数据。

读出完毕后，在命令区里设定完成码（AAAAh）。

Z-01PM确认完成码后，将下面的数据设置到共用RAM里。

第2次传送中，共用RAM情况如下：

```

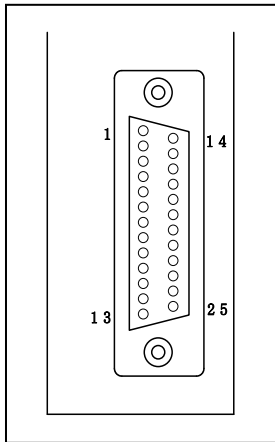
0 ~ 1 h:    命令区（完成码）
2 ~ 21 h:   程序号为5的文件信息
22 ~ 41 h:   "    6    "
42 ~ 61 h:   "    7    "
62 ~ 81 h:   "    8    "
82 ~ A1 h:   "    9    "
```

其后的顺序，与第1次相同。

※文件信息的格式，与目录区的格式相同。

1 3. 配线

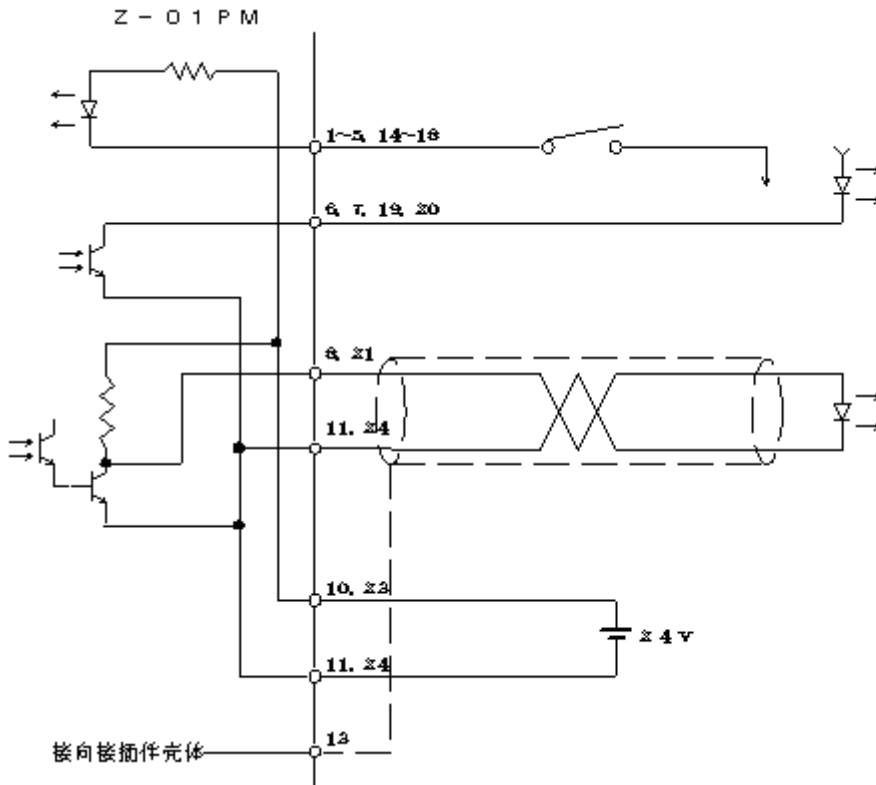
Z-01PM与外部设备用25芯的D-SUB接插件连接。



- | | |
|-----------------|-------------------|
| 1: 紧急停止输入 (EMR) | 14: Z相输入 (Z) |
| 2: 中断输入 (INT) | 15: 原点挡块输入 (HOME) |
| 3: 出错 (ERR) | 16: 到位输入 (INP) |
| 4: 正转限位输入 (CWL) | 17: 反转限位输入 (CCWL) |
| 5: 通用输入1 (IN1) | 18: 通用输入2 (IN2) |
| 6: 伺服ON输出 (SON) | 19: 偏差复位输出 (SCL) |
| 7: 通用输出1 (OUT1) | 20: 通用输出2 (OUT2) |
| 8: 正转脉冲输出 (CW) | 21: 反转脉冲输出 (CCW) |
| 9: 24V | 22: 24V |
| 10: 24V | 23: 24V |
| 11: 0V | 24: 0V |
| 12: NC | 25: NC |
| 13: FG | |

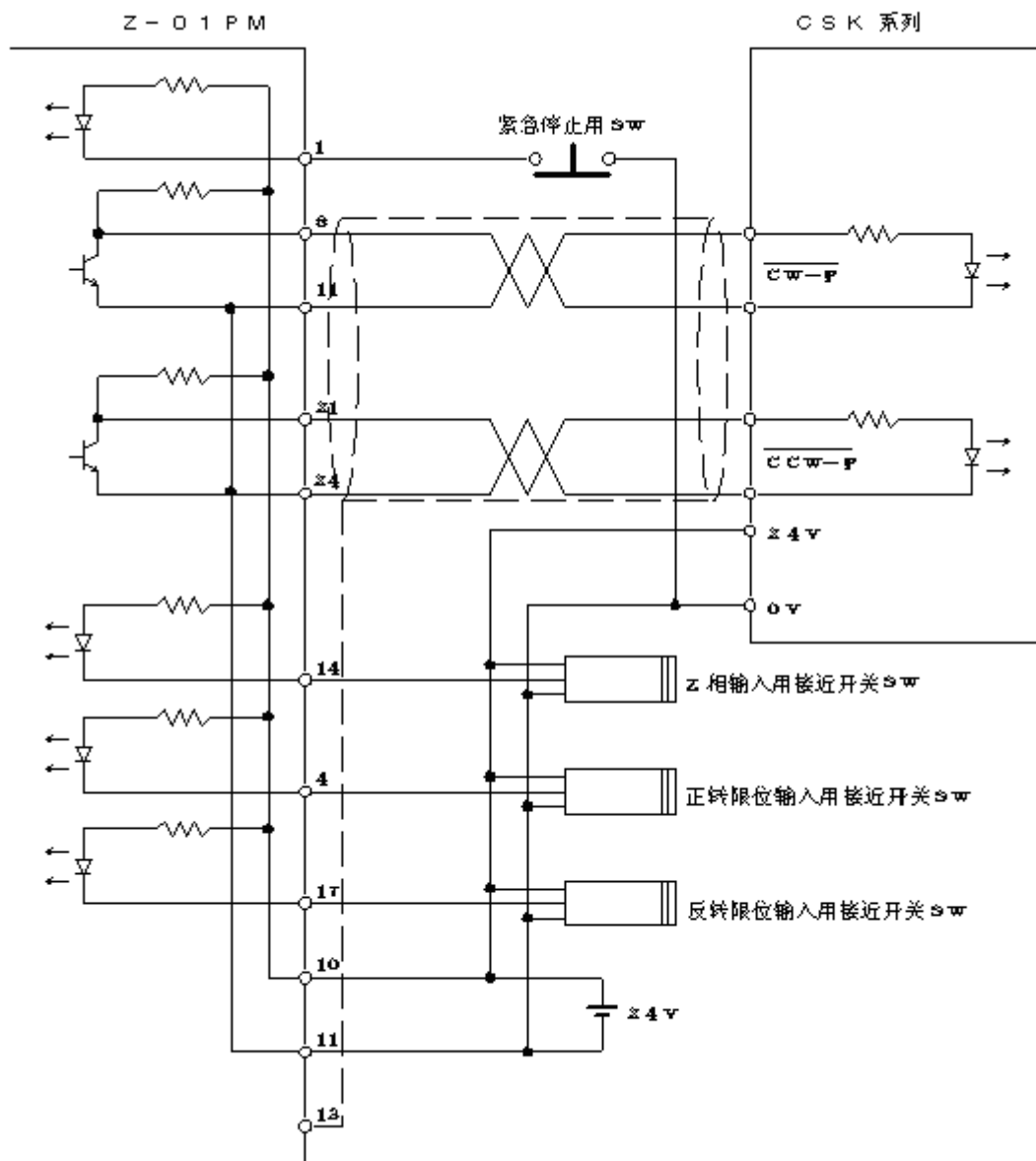
※在NC上, 不要配线。

《内部回路构成》



《配线例》

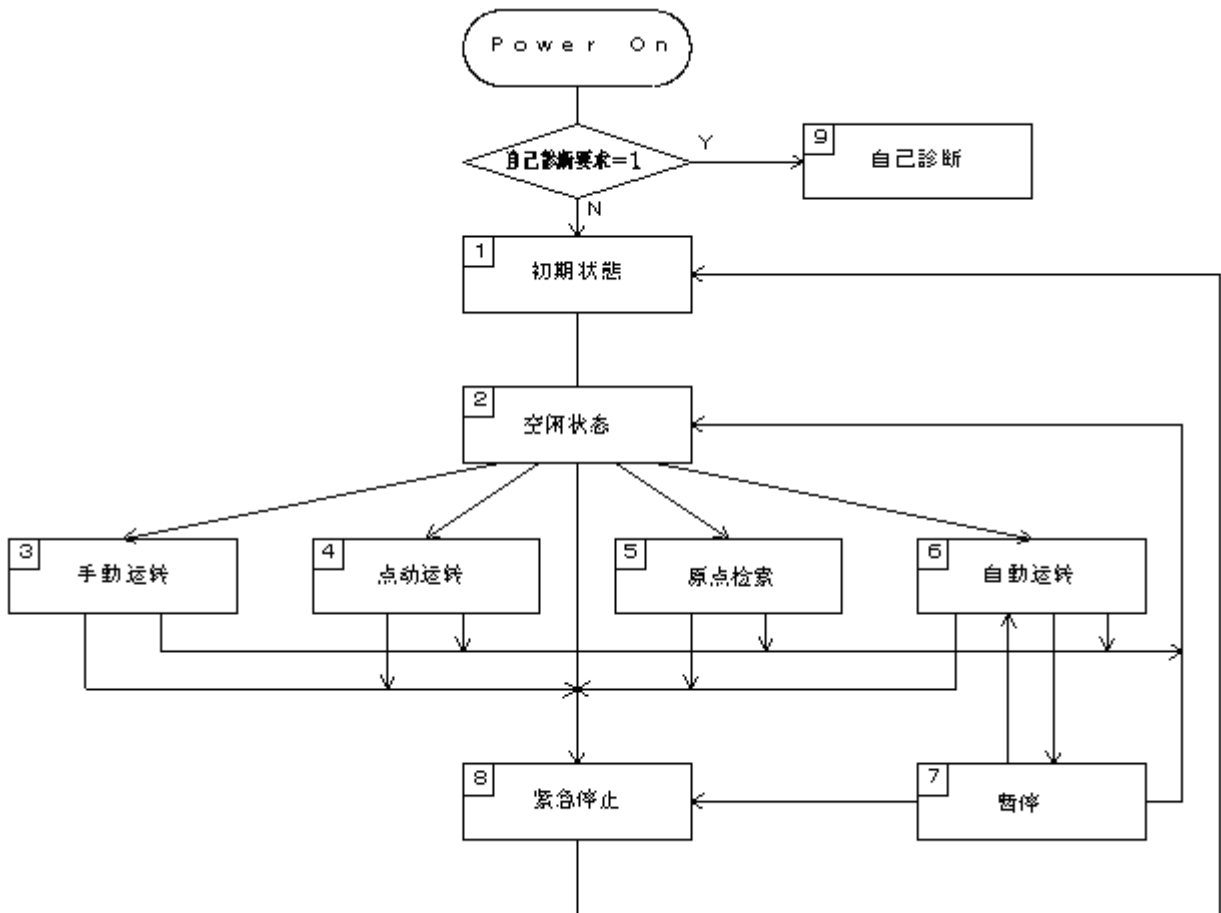
- 与CSK系列（オリエンタルモータ公司产）的连接



1 4. 运转

1 4 - 1. 状态迁移

Z - 0 1 P M的状态迁移图如下：



各状态的说明

《电源投入时》

Z - 0 1 P M解除复位后，检查是否有自诊断要求。
 有自诊断要求时，进入自诊断处理。
 无自诊断要求时，进行存储器检查。
 进行存储器检查的结果，如无异常则输出 O K，移到初始状态。
 有异常时，L E D 闪烁告知异常，而且不移到初始状态。

《初始状态（状态 1）》

在该状态下，可使用命令进行数据的读写。
 由 P L C 将 E N A B L E O N 后，进行如下处理：
 1) P L C 特殊寄存器的读出，及各种数据的传送。
 2) 外部紧急停止输入状态的确认（检查有无输入）。
 3) E R R 输入状态的确认（检查有无输入）。
 4) 正 / 反转限位状态的确认（检查有没有同时输入）。
 5) 外部 2 4 V 输入的确认。

这些处理完毕后，将伺服 O N、R E A D Y 输出，移到空闲状态。有异常时，输出出错码，L E D 灯亮告知异常。
 而且不移到空闲状态。

《空闲状态（状态2）》

在此状态下，可使用命令读写数据。

而且各种起动信号输入后即可动作。（待机状态）

《手动运转（状态3）》

在此状态下，进行手动运转。

在空闲状态时，如手动正转起动或手动反转起动O N后，即移到此状态。

如手动正转起动或手动反转起动O F F，则再次移至空闲状态。

指令不可使用。

《点动运转（状态4）》

在此状态下，进行点动运转。

在空闲状态时，点动正转起动或点动反转起动O N，即移到此状态。

点动动作结束，再次移到空闲状态。

指令不可使用。

《原点检索（状态5）》

在此状态下，进行原点检索。

在空闲状态时，原点检索起动O N后，即移到此状态。

原点检索起动O F F，或原点检索结束，则再次移至空闲状态。

指令不可使用。

《自动运转（状态6）》

在此状态下，进行自动运转。

在空闲状态或暂停状态时，如自动运转起动O N，即移到此状态。

自动运转结束，则再次移至空闲状态。

指令不可使用。

《暂停（状态7）》

此状态为暂停状态。

在自动运转中，如暂停O N，或执行P A U S E命令，即移到此状态。

自动运转起动O N，则再次移到自动运转。

工序复位O N后，即移至空闲状态。

《紧急停止（状态8）》

此状态为紧急停止状态。

在状态2～7，当紧急停止的条件成立时，移至此状态。

在此状态下，进行紧急停止处理，输出出错码。

出错原因消除，并将出错复位O N后，则移至初始状态。

使用命令可以读出数据。（不可写入）

《自诊断（状态9）》

在此状态下，进行自诊断。

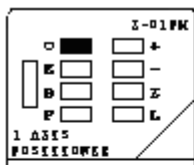
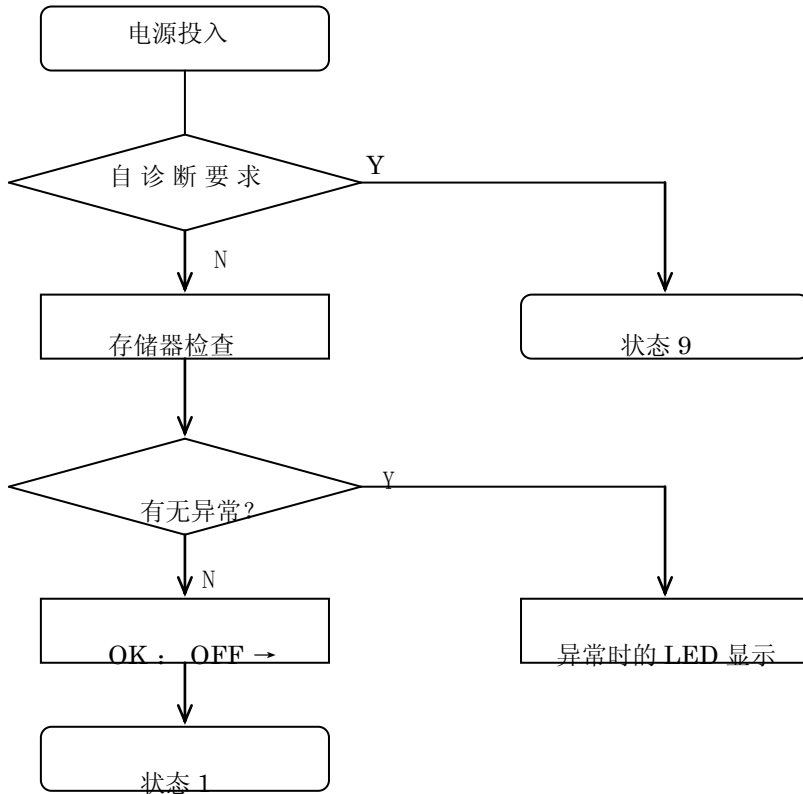
在电源投入时，如有自诊断的要求，即移到此状态。

一旦进入此状态后，就不能移至其他状态。（只能电源O F F）。

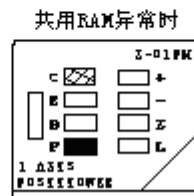
1 4 - 2 . 各状态的处理

表示各状态的处理。
 这些流程图，主要表述了有关状态迁移的处理。
 (严格地说还有其他处理，此外不作表述)

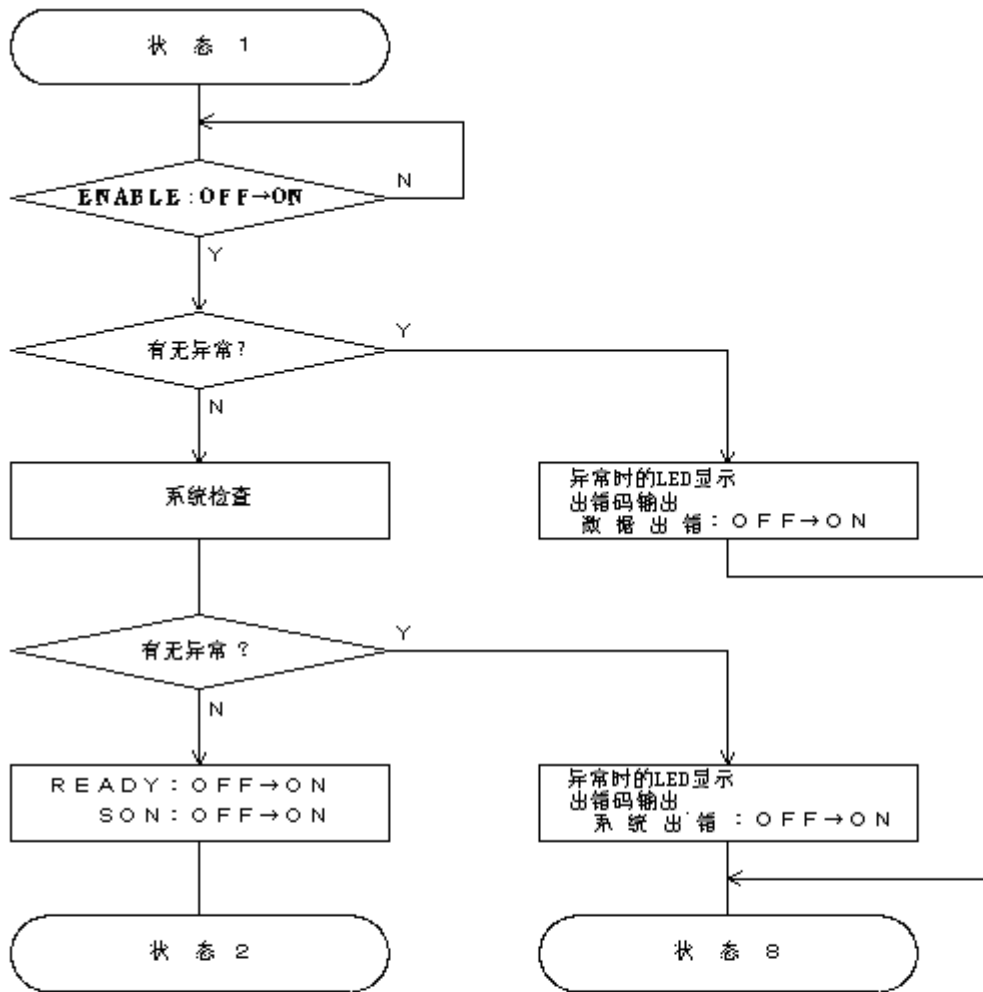
1) 电源投入→状态 1

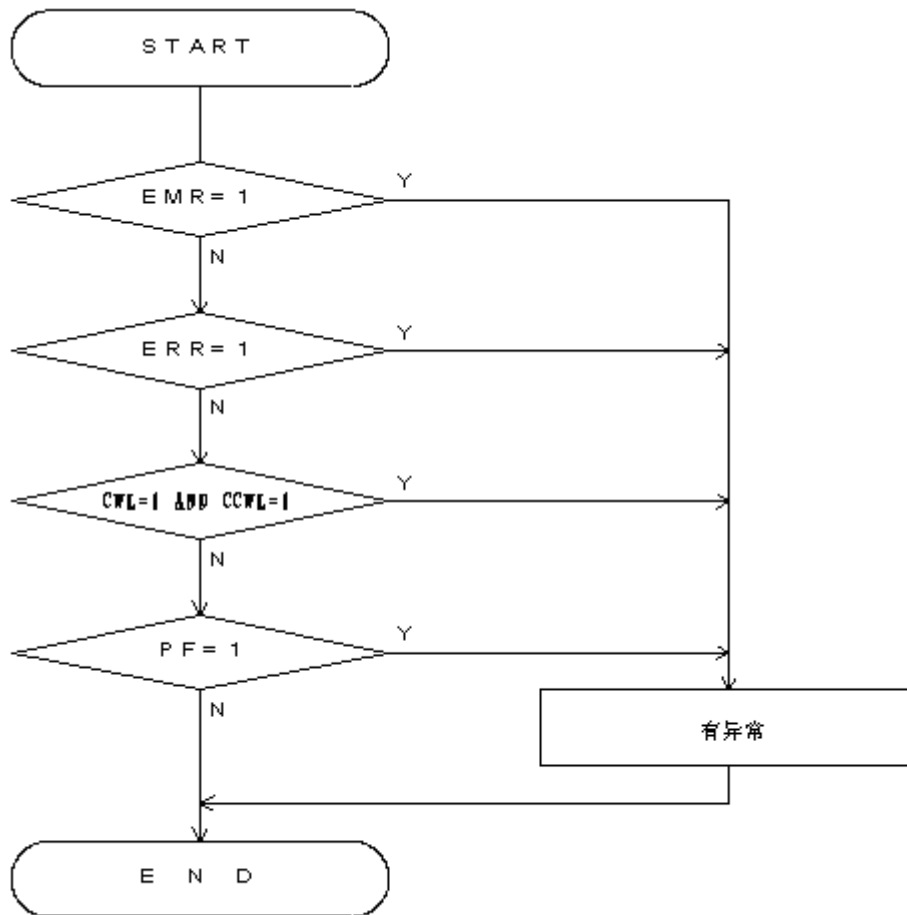


■ : 灯亮
 □ : 灯灭
 ▨ : 闪烁(0.5s间隔)

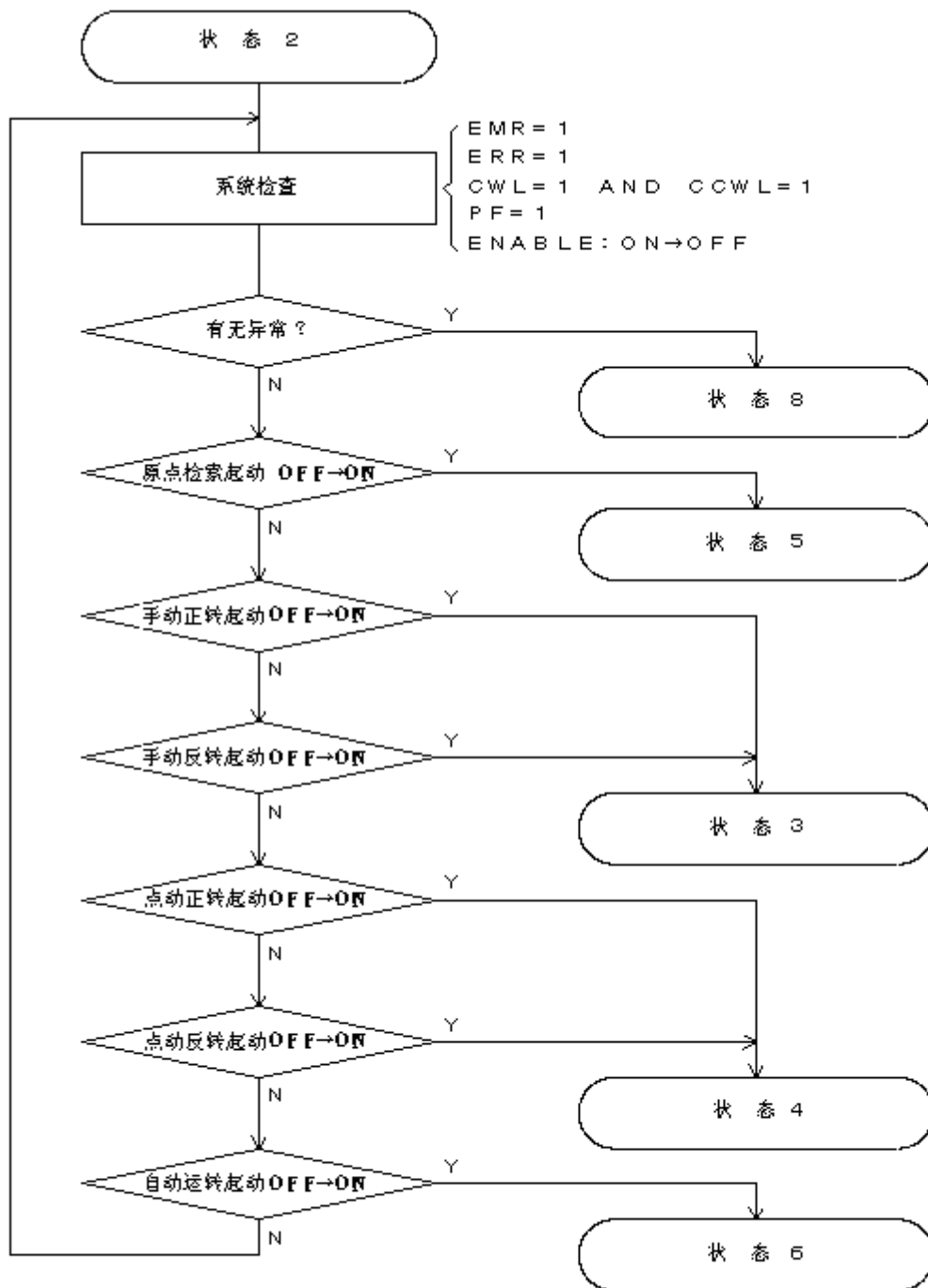


2) 状态 1 → 状态 2

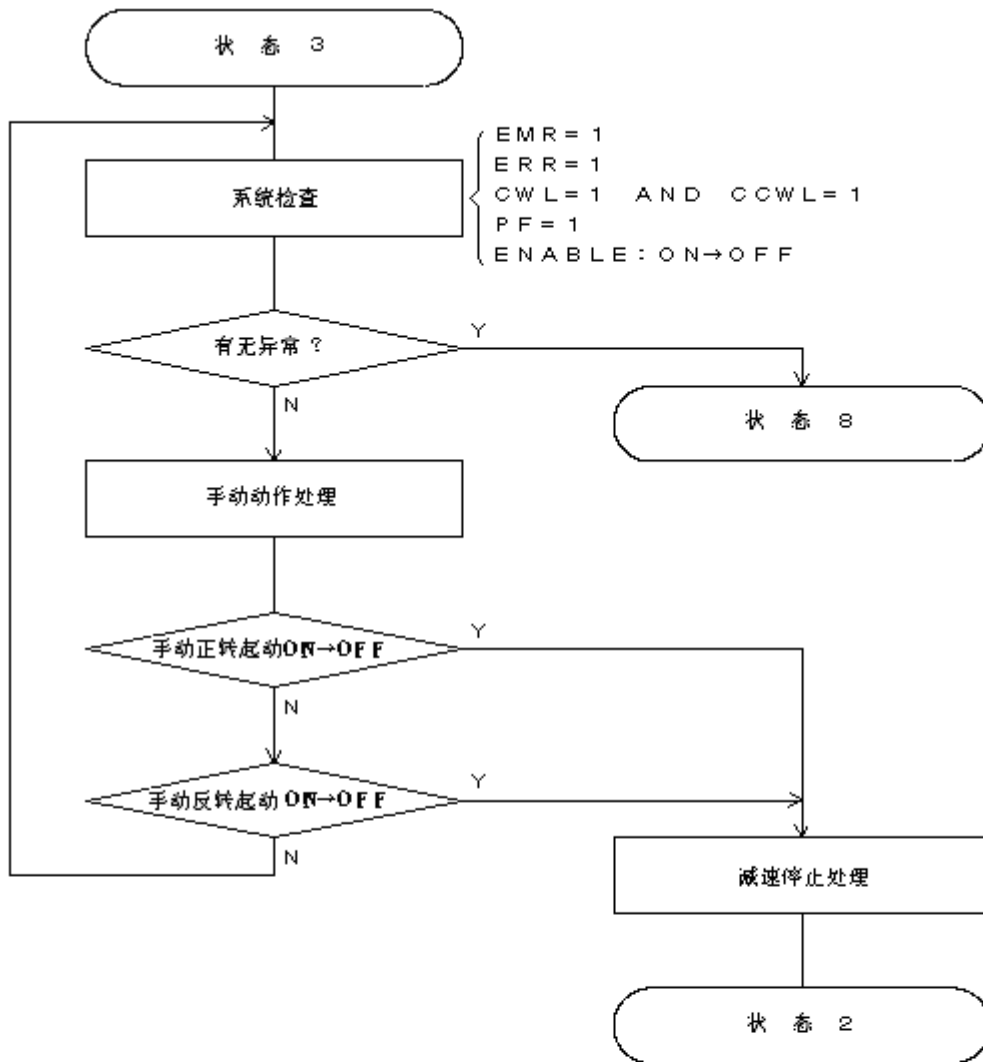




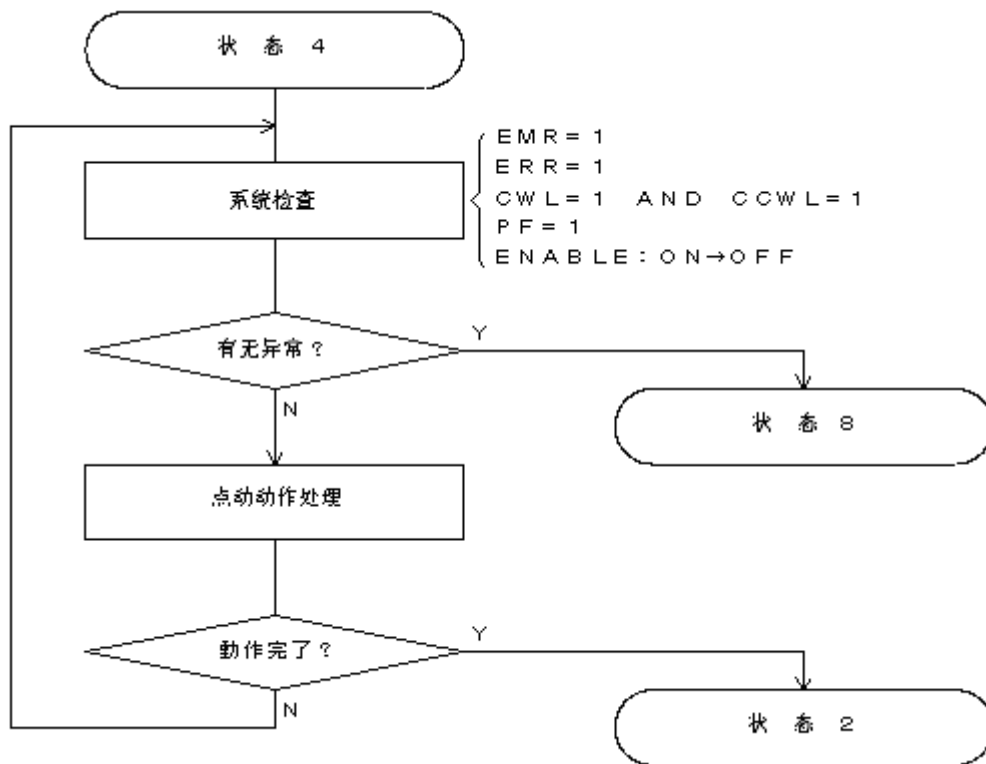
3) 状态 2 → 状态 3、4、5、6



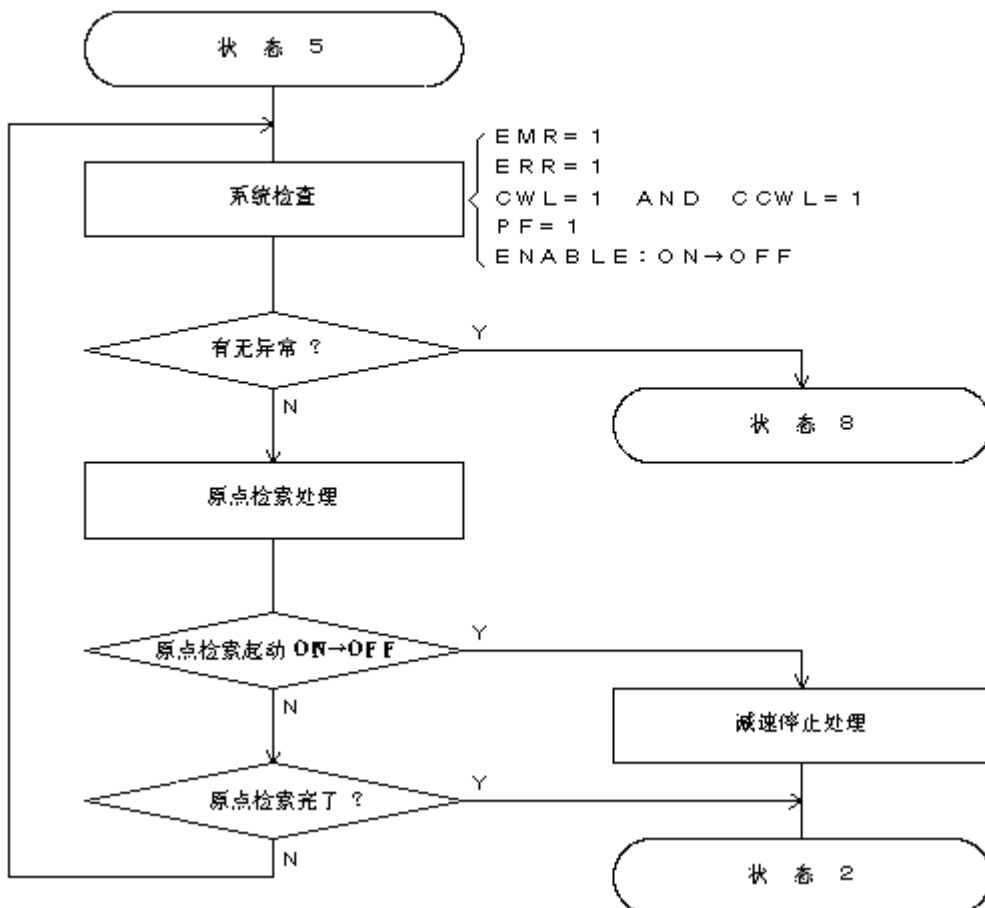
4) 状态 3



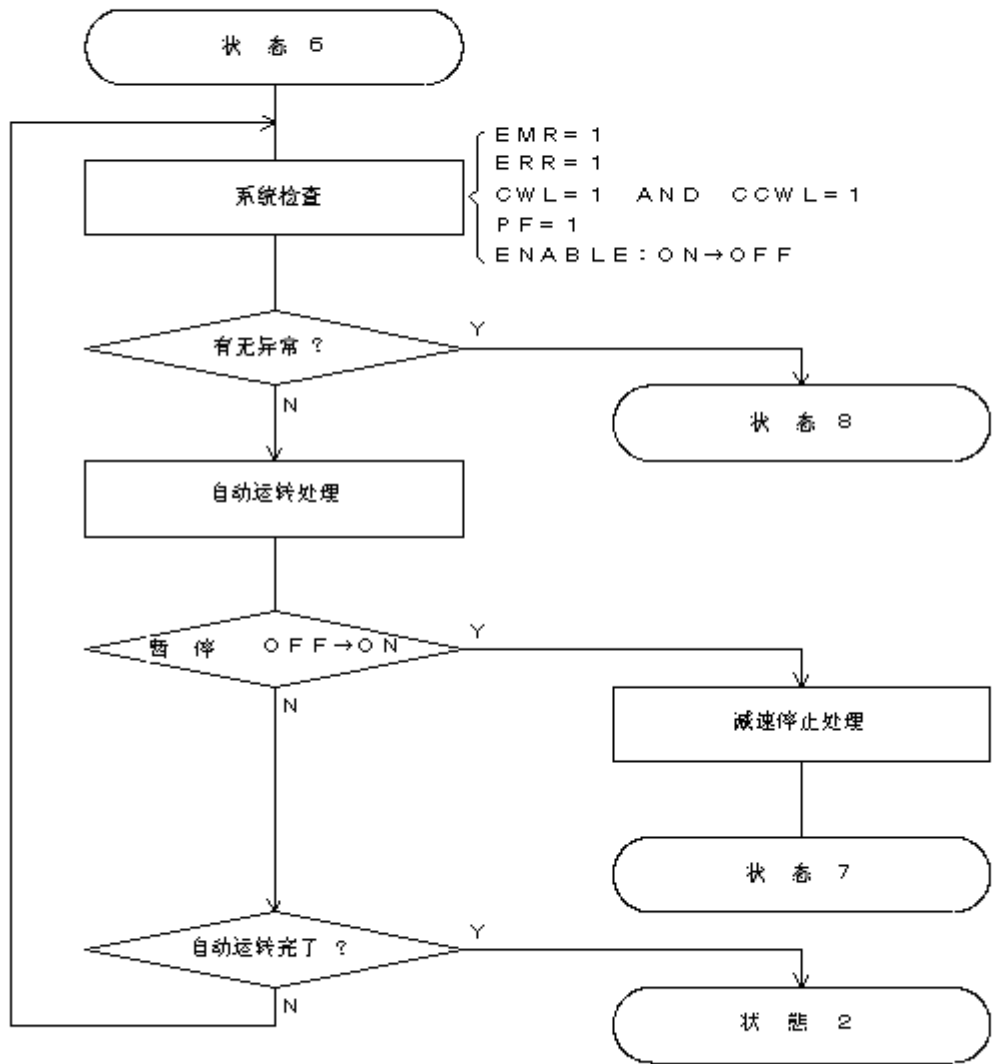
5) 状态 4



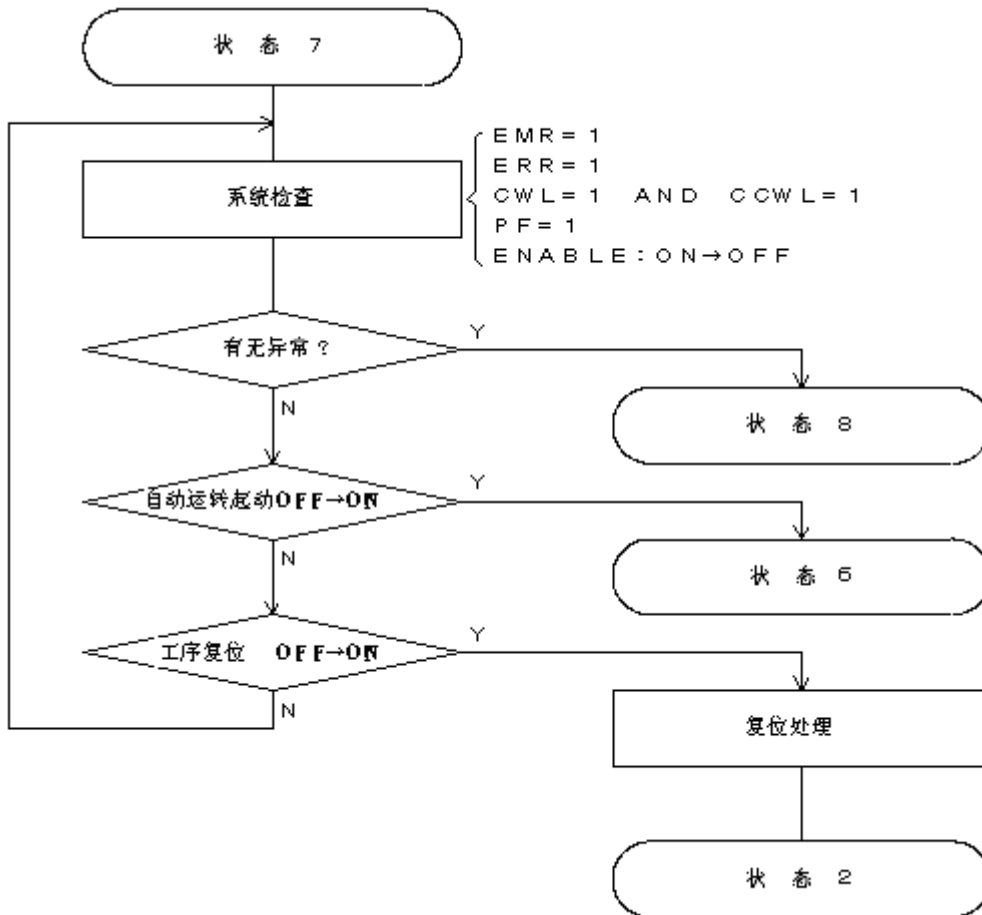
6) 状态 5



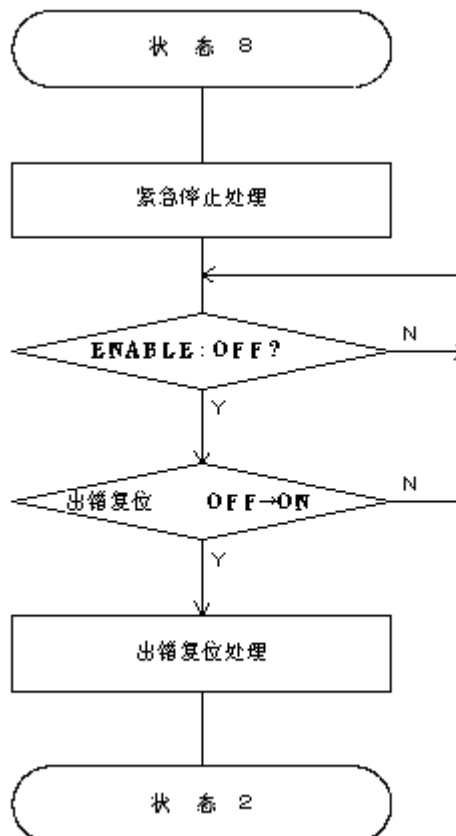
7) 状态 6



8) 状态 7



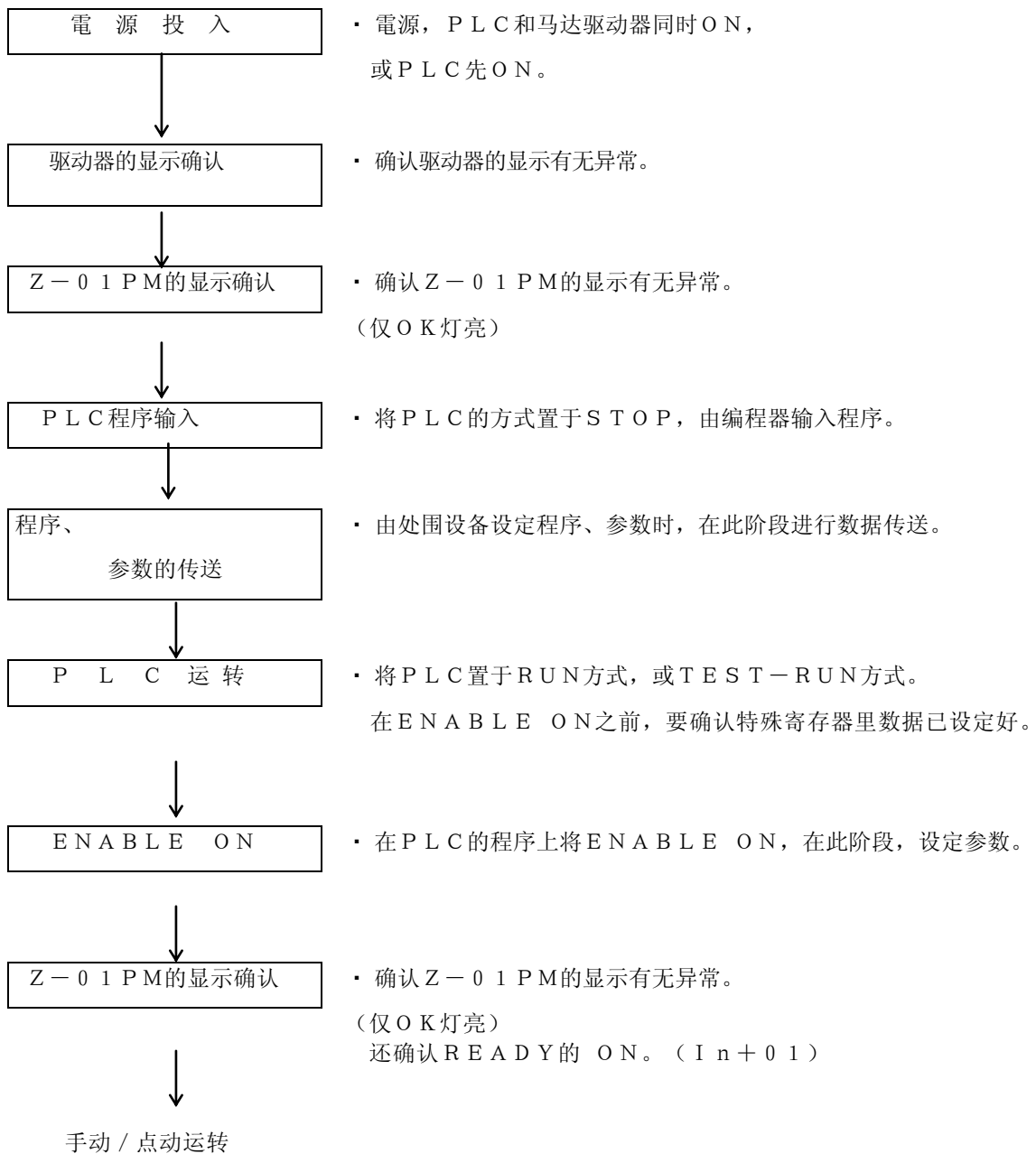
9) 状态 8



※关于状态 9，另有规定。

1 4 - 3 . 起 动

Z-01PM配线完毕后,按下面的顺序进行起动。



1 4 - 4 . 手 动 / 点 动 运 转

手动动作确认

- 由 P L C 使手动正转起动 O N。
确认马达向+方向以指令速度转动。
- 然后，使手动反转起动 O N，确认马达向-方向以指令速度转动。

点动动作确认

- 由 P L C 使点动正转起动 O N。
确认马达向+方向以指令速度转动，并在指定量动作后停止。
- 对反转方向也进行同样的确认。

限位开关的动作确认

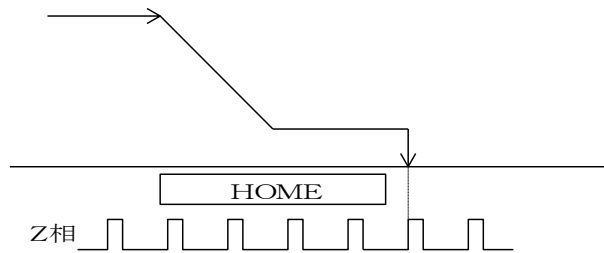
- 进行手动动作，动作到达正转限位时，确认即使手动正转起动继续 O N，动作也停止。
- 对反转限位也进行同样的确认。

转原点检索

1 4 - 5 . 原 点 检 索

原点检索动作确认

- 由 P L C 使原点检索起动 O N。
确认原点检索是否按指定的方式进行。
- 原点检索动作完毕后的停止位置，确认是否是合适的位置。
- Z 相信号在通过 H O M E 后，信号之前输入或之后输入，开关的应答速度不一致，停止位置会有偏差。
要将 H O M E 信号的通过位置，调整在 Z 相信号的中间附近。



自动运转

1 4 - 6 . 自 动 运 转

程序号码的写入

- 将起动的程序号码写入共用 R A M。

自动运转确认

- 由 P L C 使自动运转起动 O N。
确认是否在按程序进行定位动作。

调整終了

14-7. 简易程序

Z-01PM的自动运转程序, 需要由用户编制、传送。进行简易的定位动作时, 可以使用预先准备的简易程序。

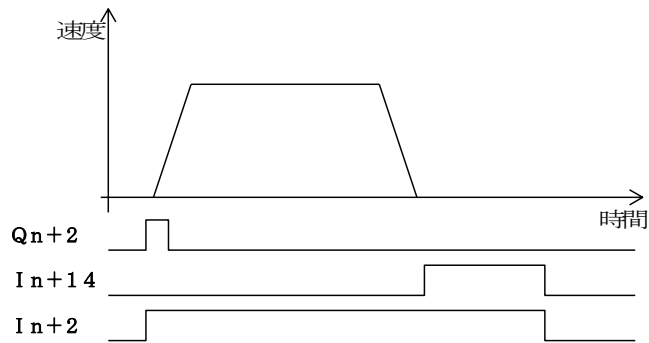
该程序存储在Z-01PM的PROM里, 可通过指定程序号码, 使自动运转起动ON来使用。

另外, 位置, 速度等参数, 可以通过将数据写入指定的数据寄存器来进行设定。

Z-01PM备有9种简易程序。

《程序号: 10》

```
G 0 0 X ( D 6 4 ) F ( D 6 5 )
G 0 4 K ( D 6 6 )
E N D
```



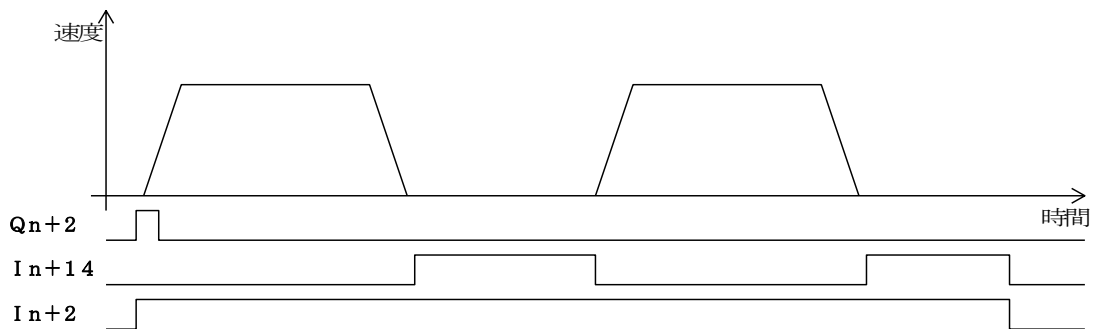
《程序号: 11》

```
G 9 1
G 0 0 X ( D 6 4 ) F ( D 6 5 )
G 0 4 K ( D 6 6 )
E N D
```

※是程序号码10变为相对值指令的程序。

《程序号: 12》

```
G 0 0 X ( D 6 4 ) F ( D 6 5 )
G 0 4 K ( D 6 6 )
G 0 0 X ( D 6 7 ) F ( D 6 8 )
G 0 4 K ( D 6 9 )
E N D
```



《程序号: 1 3》

```

G 9 1
G 0 0 X ( D 6 4 ) F ( D 6 5 )
G 0 4 K ( D 6 6 )
G 0 0 X ( D 6 7 ) F ( D 6 8 )
G 0 4 K ( D 6 9 )
E N D

```

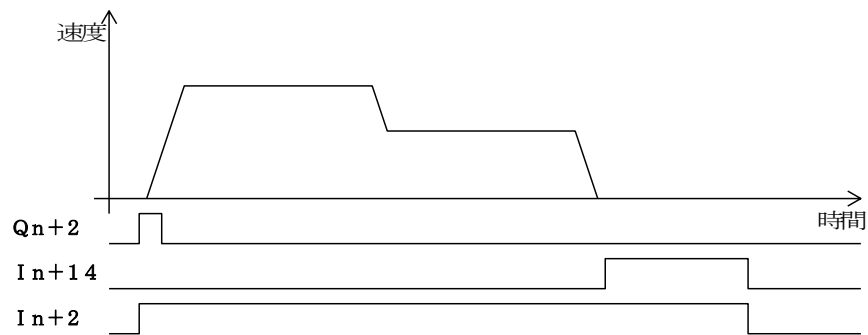
※是程序号 1 2 变为相对值指令的程序。

《程序号: 1 4》

```

G 0 5 X ( D 6 4 ) F ( D 6 5 )
G 0 0 X ( D 6 6 ) F ( D 6 7 )
G 0 4 K ( D 6 8 )
E N D

```



《程序号: 1 5》

```

G 9 1
G 0 5 X ( D 6 4 ) F ( D 6 5 )
G 0 0 X ( D 6 6 ) F ( D 6 7 )
G 0 4 K ( D 6 8 )
E N D

```

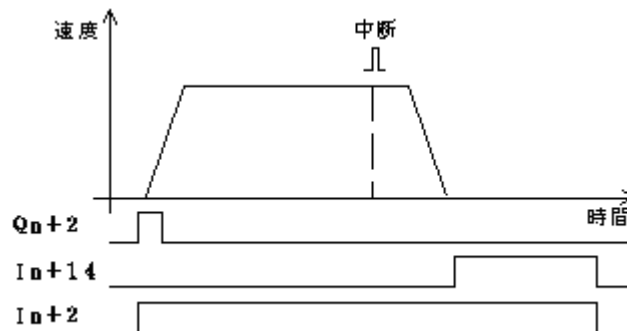
※是程序号码 1 4 变为相对值指令的程序。

《程序号: 1 6》

```

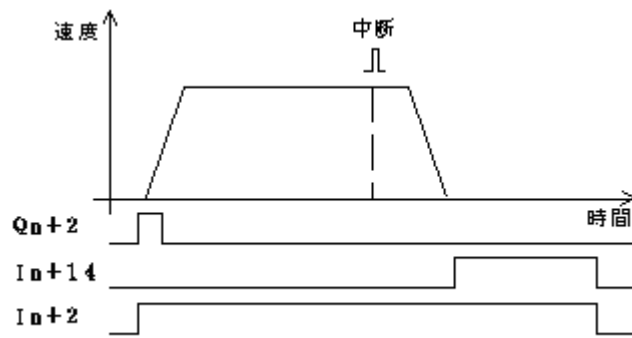
G 2 0 X ( D 6 4 ) F ( D 6 5 )
G 0 4 K ( D 6 6 )
E N D

```



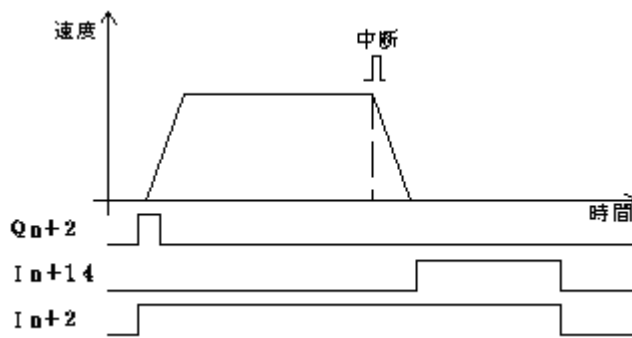
《程序号： 1 7 》

```
G 2 1 X ( D 6 4 ) F ( D 6 5 ) F ( D 6 6 )
G 0 4 K ( D 6 7 )
E N D
```



《程序号： 1 8 》

```
G 2 2 X+ F ( D 6 4 )
G 0 4 K ( D 6 5 )
E N D
```



(-1)

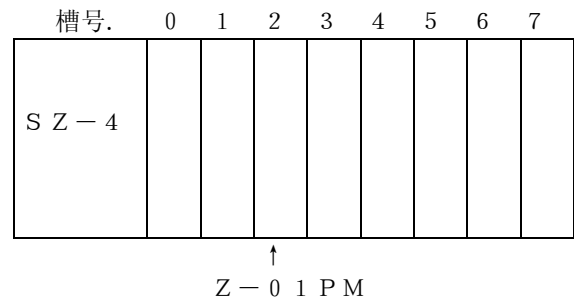
1 5. 出错码

出错可从S Z - 4 外围设备读出。

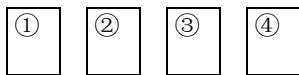
[例] 右图的构成由S - 0 1 P 读出的情况

4 7 菜单
 基架? 0 ↓
 槽号? 2 ↓
 地址? 3 5 4 ↓
 字节? 2 ↓ ↓: 回车

显示 4 位的出错码。



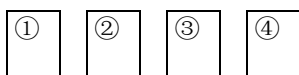
1) 系统出错



①②: 表示出错发生时的系统的运转状态。
 ③④: 表示出错内容。

- 运转状态 (①②)
 - 0 0: 空闲状态
 - 0 1: 原点检索
 - 0 2: 自动运转 (也要参阅C N C 语言解析出错项目)
 - 0 3: 手动正转
 - 0 4: 手动反转
 - 0 5: 点动正转
 - 0 6: 点动反转
- 出错内容 (③④)
 - 0 1: P L C 的 I R 信号为「0」(I R 发生)。
 - 0 2: O V T + 与 O V T - 同时 O N。
 - 0 3: O V T + 为 O N。(仅自动、点动运转时)
 - 0 4: O V T - 为 O N。(仅自动、点动运转时)
 - 0 5: S Z 的 E N A B L E 信号为 O F F。(仅 B U S Y 中)
 - 0 6: P M 的 E M R 信号 O N。
 - 0 7: 内部软极限 + O N。(仅自动、点动运转时)
 - 0 8: 内部软极限 - O N。(仅自动、点动运转时)
 - 0 9: P M 的 E R R 信号 O N。
 - 1 0: 外部电源电压降低出现异常。
 - 2 1: 连续模式 (G 0 5, G 2 5 等) 的移动量不够。
(仅自动运转时)
 - 2 2: 终止模式的移动量不够。(仅自动运转时)
 - 2 3: 不存在要执行的程序。(仅自动运转时)
 - 2 4: 在自动运转中, 不存在 J M P 目标行。(仅自动运转时)

2) 数据 (参数) 设定出错



①: 4 固常
 ②: 4 固常
 ③④: 表示以下出错内容。

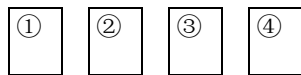
发生时序: 自己诊断 (E N A B L E 继电器 O N) 时

2 - 1) 数据出错

- 4 4 0 1: 对于 P M 的要求, 从 S Z 回来了出错。(内部出错)
- 4 4 0 2: P L C 设定的命令不存在。(写入了不存在的命令码)
- 4 4 0 3: 读出、写入 C N C 程序时, 指定了 0 ~ 9 以外的程序号。
- 4 4 0 4: C N C 程序读出时, 指定的程序不存在。

- 4 4 4 0：向PM的S R A M写入时，P M在B U S Y中。
- 4 4 4 1：从PM的E E P R O M读入时，E E P R O M正在写入处理中。
- 2-1) 数据出错(续)
- 4 4 5 0：设定PM的现在位置时，数据不是B C D，或在范围以外。
- 4 4 6 0：清除程序时，P M在B U S Y中。
- 4 4 6 1：清除程序时，P M在E E P R O M的写入处理中。
- 4 4 6 2：清除的程序号，在0~9以外时。
- 4 4 7 0：P L C设定的程序号(共用R A M A d d：2 7 6)不是B C D。
- 4 4 7 1：P L C设定的程序号，在0~9、20~28以外时。
- 4 4 8 0：读写D寄存器时，数据数32以上，或在范围外。
- 4 4 8 1：读写P寄存器时，数据数64以上，或在范围外。
- 2-2) 系统参数传送出错
- 4 4 9 0：系统参数的电子齿轮M不是B C D。
- 4 4 9 1：" 的电子齿轮M是0。
- 4 4 9 2：" 的电子齿轮D不是B C D。
- 4 4 9 3：" 的电子齿轮D是0。
- 4 4 9 4：" 的电子齿轮M / D在范围外。
- 4 4 9 5：" 的偏置速度不是B C D。
- 4 4 9 7：" 的速度极限值不是B C D。
- 4 4 9 8：" 的速度极限值在范围外。
- 4 4 9 9：" 的偏置速度大于速度极限值。
- 4 4 9 A：" 的手动速度不是B C D。
- 4 4 9 B：" 的手动速度是0。
- 4 4 9 C：" 的手动速度大于极限值。
- 4 4 9 D：" 的点动速度不是B C D。
- 4 4 9 E：" 的点动速度是0。
- 4 4 9 F：" 的点动速度大于速度极限值。
- 4 4 A 0：" 的速度极限值 $\times M / D$ 大于4 0 0 0 0 0。
- 4 4 A 1：" 的点动移动量不是B C D。
- 4 4 A 2：" 的点动移动量在范围外。
- 4 4 A 3：" 的超调值不是B C D。
- 4 4 A 4：" 的超调值在范围外。
- 4 4 A 5：" 的间隙补偿量不是B C D。
- 4 4 A 6：" 的加速时间不是B C D。
- 4 4 A 7：" 的减速时间不是B C D。
- 4 4 A 8：" 的紧急停止减速时间不是B C D。
- 4 4 A 9：" 的软极限(+)不是B C D。
- 4 4 A A：" 的软极限(+)在范围外。
- 4 4 A B：" 的软极限(-)不是B C D。
- 4 4 A C：" 的软极限(-)在范围外。
- 4 4 A D：" 的软极限(+)小于软极限(-)。
- (-3) 4 4 A E：" 的S形加减速补偿值不是B C D，或在范围外。
- (-3) 2-3) 原点参数传送时出错
- 4 4 C 0：原点参数的检索速度不是B C D。
- 4 4 C 1：" 的检索速度大于速度极限。
- 4 4 C 2：" 的蠕动速度不是B C D。
- 4 4 C 3：" 的蠕动速度大于速度极限。
- 4 4 C 4：" 的移位速度不是B C D。
- 4 4 C 5：" 的移位速度大于速度极限。
- 4 4 C 6：" 的移位移动量不是B C D。
- 4 4 C 7：" 的移位移动量在范围外。
- (-3) 2-4) 特殊参数传送时出错
- 4 4 D 0：特殊参数的材料厚度不是B C D，或是0。
- 4 4 D 1：" 的轴直径不是B C D，或是0。
- 4 4 D 2：" 的马达1转的脉冲数不是B C D，或是0。
- 4 4 D 4：" 的传动比不是B C D。
- 4 4 D 5：" 的传动比在范围外。

3) CNC语言解析出错



- ①: 0 固定
 ②: 2 固定
 ③④: 表示以下出错内容:

发生时序: 自动运转中, 有问题的程序解析时, 但不特定问题程序。

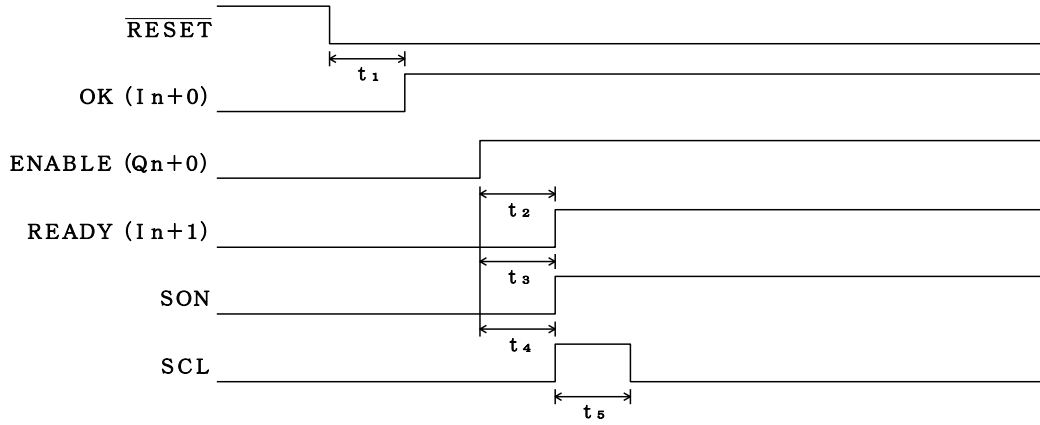
- 0 2 8 0: 程序中存在未定的码 (G 0 0 R 1 0 0 F 2 0 0)
 0 2 8 1: 标号号码 (N) 里使用了范围以外的数值
 0 2 8 2: 程序中存在未定义的 G 码
 0 2 8 3: 在 M 码号里使用了 0 ~ 2 5 5 以外的数值
 0 2 8 4: P 寄存器号码使用了 0 ~ 6 3 以外的数值
 0 2 8 5: 常数里使用了范围以外的数值 (- 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7)
 0 2 8 6: 在 P M 侧的输入继电器号码 (I) 里使用了范围外的数值
 0 2 8 7: 在 P M 侧的输出继电器号码 (Q) 里使用了范围外的数值
 0 2 8 8: 在送进速度里使用了范围以外的数值 (1 ~ 2 0 0 0 0 0 0 0)
 0 2 8 A: 在 P L C 侧的输入继电器号码 (# I) 里使用了范围外的数值
 0 2 8 B: 在 P L C 侧的内部继电器号码 (# M) 里使用了范围外的数值
 0 2 8 C: 在 P L C 侧的输出继电器号码 (# Q) 里使用了范围外的数值
 0 2 8 D: 在 P L C 侧的寄存器号码 (# R) 里使用了范围外的数值
 0 2 8 F: 在座标里使用了范围外的数值 (- 8 3 8 8 6 0 8 ~ 8 3 8 8 6 0 7)
 0 2 9 A: 在等待时间里使用了范围外的数值 (0 ~ 9 9 9 9)
 0 2 9 B: 使用连续定位时, 到达指定的速度所需的加减速距离不够
 (G 0 0、G 0 5、G 2 5、G 2 6)
 0 2 9 C: 在加减速时间里, 使用了范围外的数值 (G 3 0、G 3 1)
 0 2 9 F: 由子程序起动了其他子程序
 0 2 A 0: 在连续定位中, 使用了不可使用的命令 (G 0 5、G 2 5)
 0 2 A 1: 在子程序的标号号码里, 使用了范围外的数值
 0 2 A 2: 在子程序的内部记述了程序终了
 0 2 A 7: 在 P M 的内部寄存器 (D) 里使用了范围外的号码 (D 0 ~ D 1 2 7)
 0 2 A 8: 在绝对模式下使用了图形 3、4 (G 2 0、G 2 1)
 0 2 A A: 没有“(”对应于“)” {G 0 X (D 0 F (D 1 0 0) : 0 2 A A 发生)
 {G 0 X D 0) F (D 1 0 0): 0 2 8 0 发生}
 0 2 A C: 连续定位的动作方向反转时 (G 0 5、G 2 5)
 0 2 B 0: 程序的书写格式有错误 (在同一行里, 正确的程序后有记号)
 0 2 B 1: 在连续定位中存在图形 3、4 (G 0 5、G 2 5)
 0 2 B 2: 在子程序的起始行里, 不存在子程序的定义命令 (无 G 7 2)
 0 2 B B: 整块传送命令 (G 6 3) 的参数设定错误 (D 寄存器号, R 寄存器号设定错)
 0 2 B C: 整块传送命令 (G 6 3) 传送大小的数值范围超过 (成为 K ≧ 1 7)
 0 2 B F: 条件判定 (G 6 0, 6 1) 的「比较对象」的设定错误
 0 2 C 0: 条件判定 (G 6 0, 6 1) 的「比较本体」的设定错误
 0 2 C 1: 条件判定 (G 6 0, 6 1) 的「相关算符」的设定错误
 0 2 C 5: 用 0 去除
 0 2 D 0: S U B (G 7 2) 由 C A L L (G 7 0) 以外执行
 0 2 D 1: 没有执行 C A L L (G 7 0), 却执行了 R E T (G 7 4)
 0 2 D 2: 用运算命令进行了不能使用的组合运算 (例 P 0 = P 1 + D 0)
 0 2 D 6: 在 G 2 6 的后面没有定位命令 (G 0 0), 而 E N D 时。

※ 出错按出错码小号起依次检查。发生出错时, 检查在那里终了, 因此不会同时发生多个。
 数据修正后, 使 E N A B L E 再次 O N 后, 检查其他设定。

1 6 . 附 录

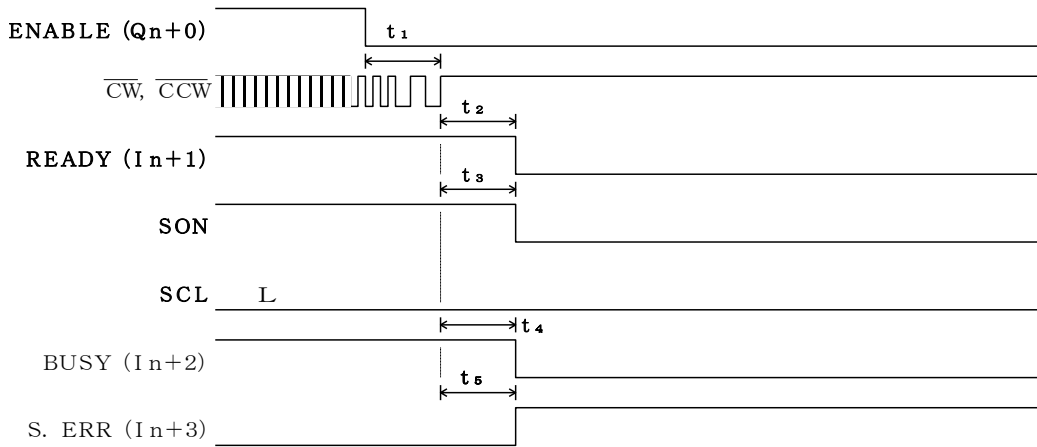
1 6 - 1 . 时 序 图

《电源投入时》

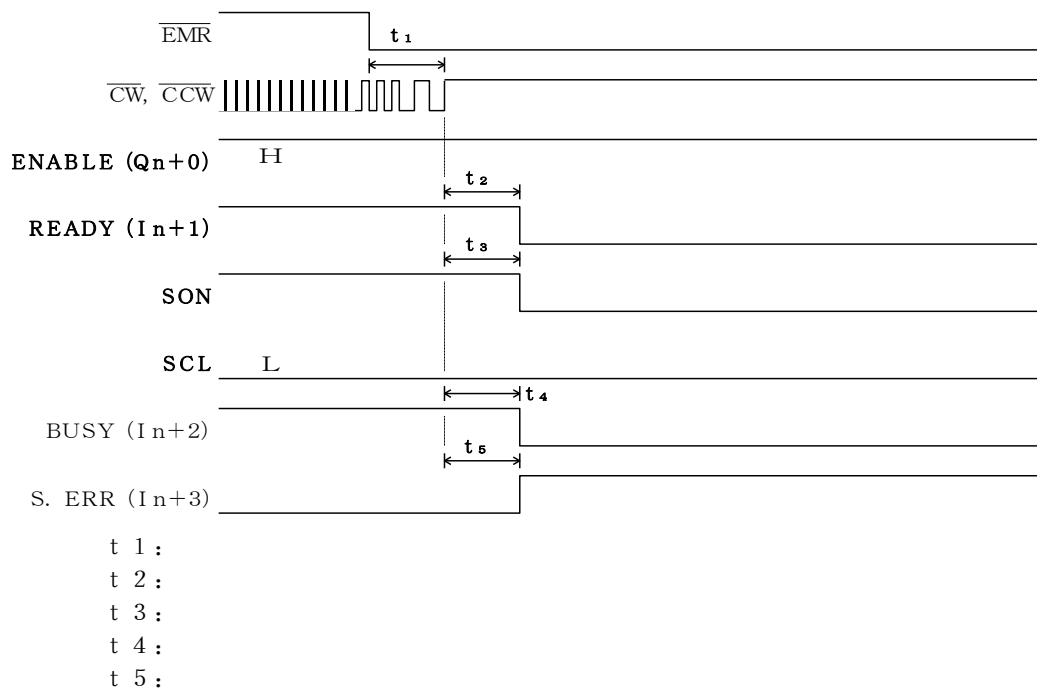


- t 1 :
- t 2 :
- t 3 :
- t 4 :
- t 5 : 5 0 m S

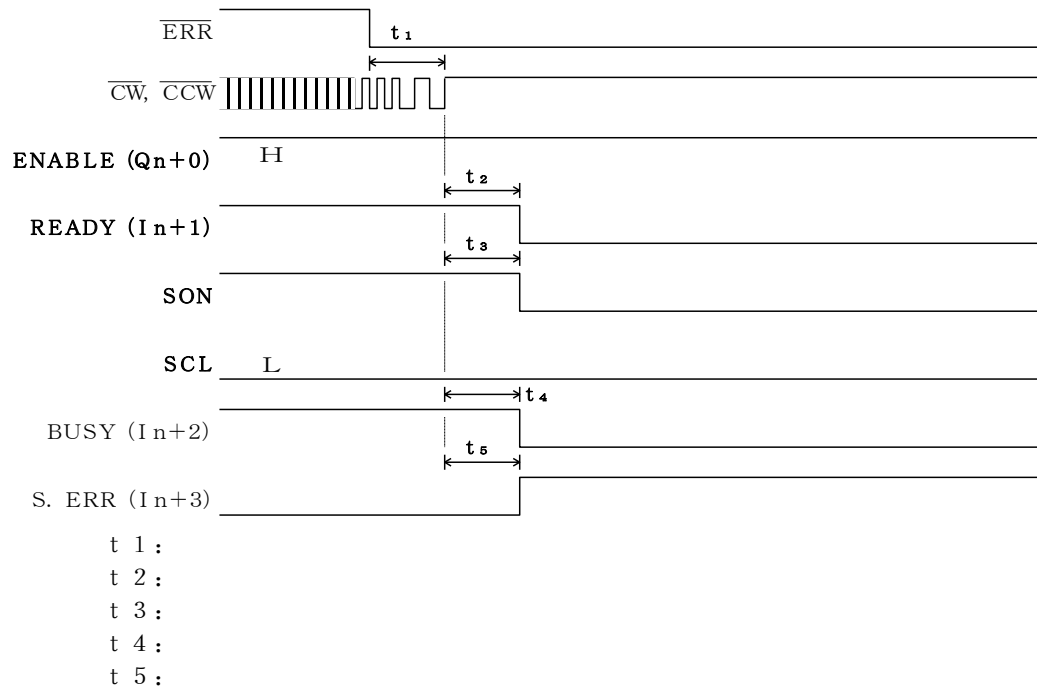
《ENABLE OFF时》

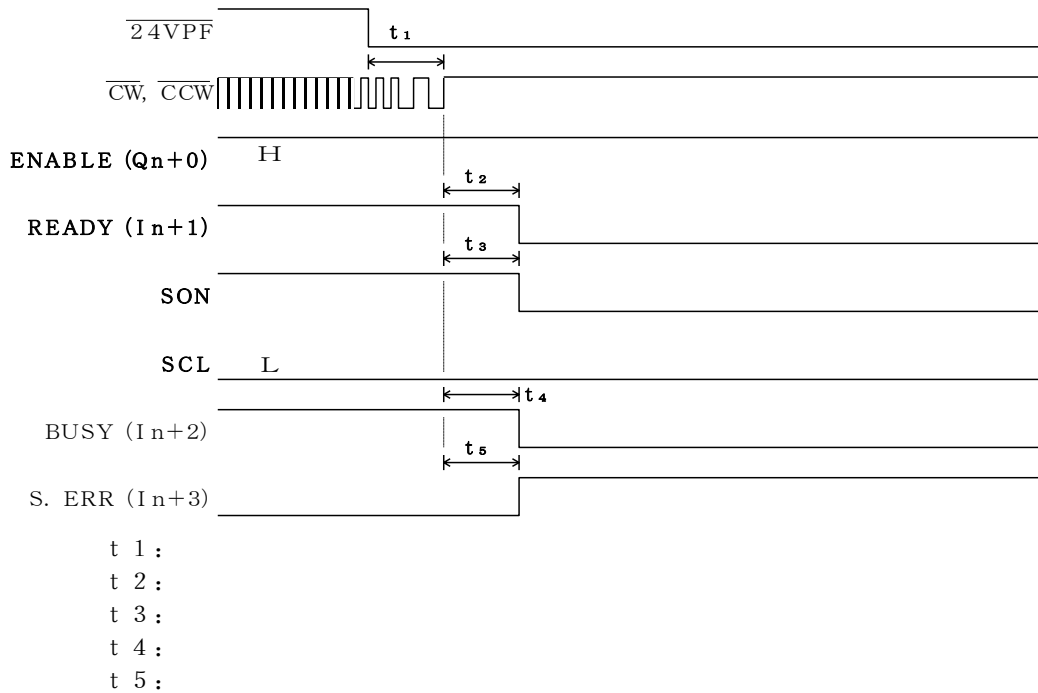


- t 1 :
- t 2 :
- t 3 :
- t 4 :
- t 5 :

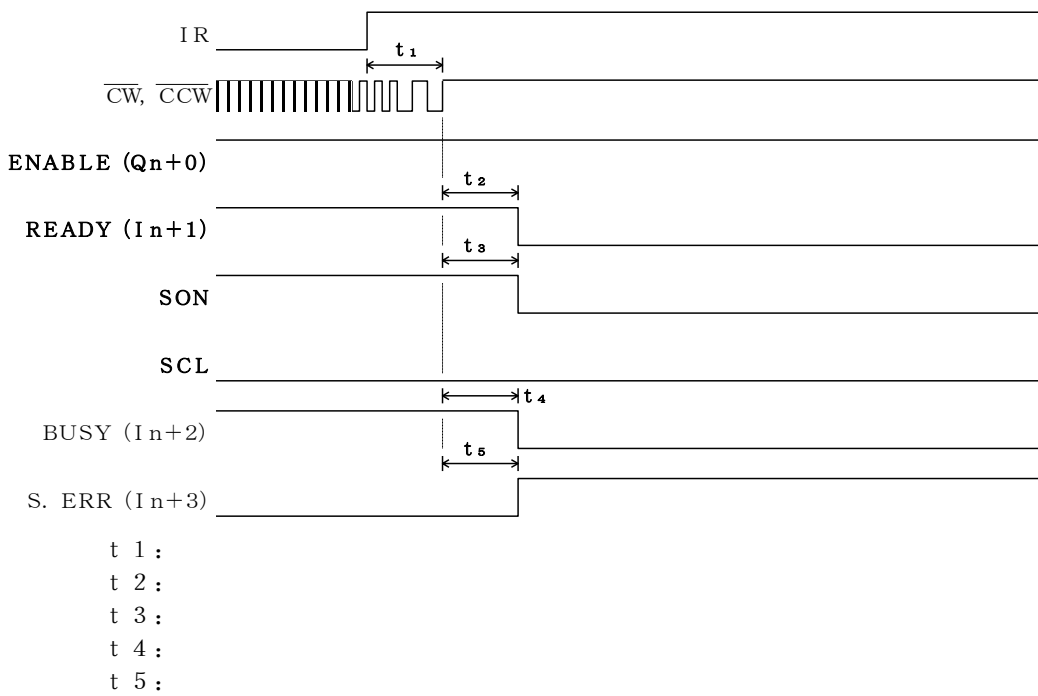


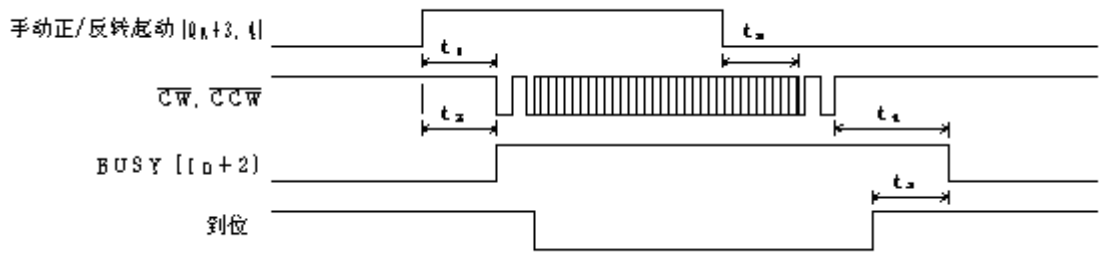
《伺服出错输入时》





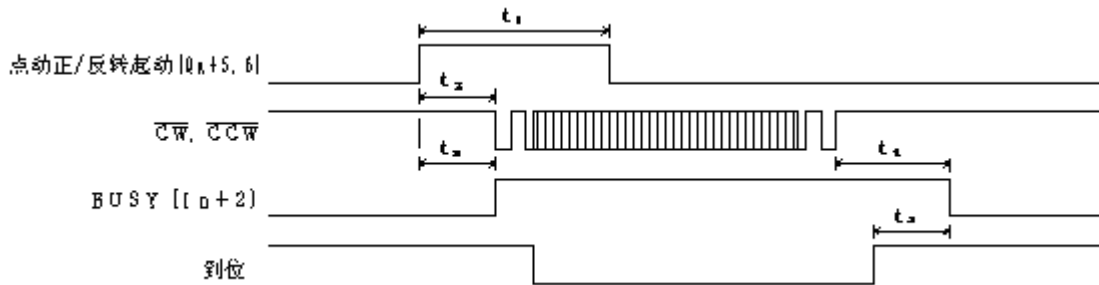
《 I R 信号输入时》





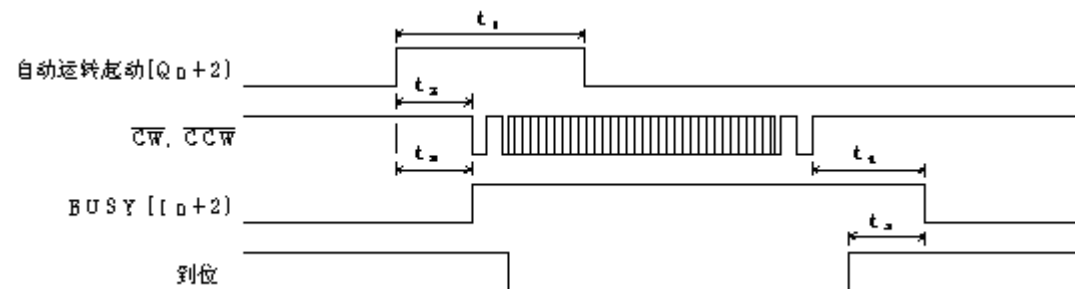
- t 1 :
- t 2 :
- t 3 : (減速開始時間)
- t 4 : (到位信号H时的延迟时间)
- t 5 :

《点动运转》

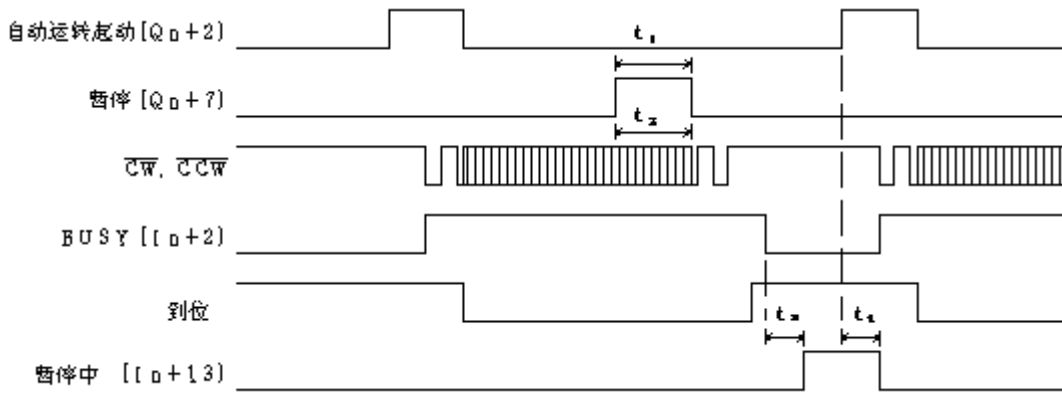


- t 1 : 1次扫描以上
- t 2 :
- t 3 : (減速開始時間)
- t 4 : (到位信号H时的延迟时间)
- t 5 :

《自动运转（起动 / 停止）》

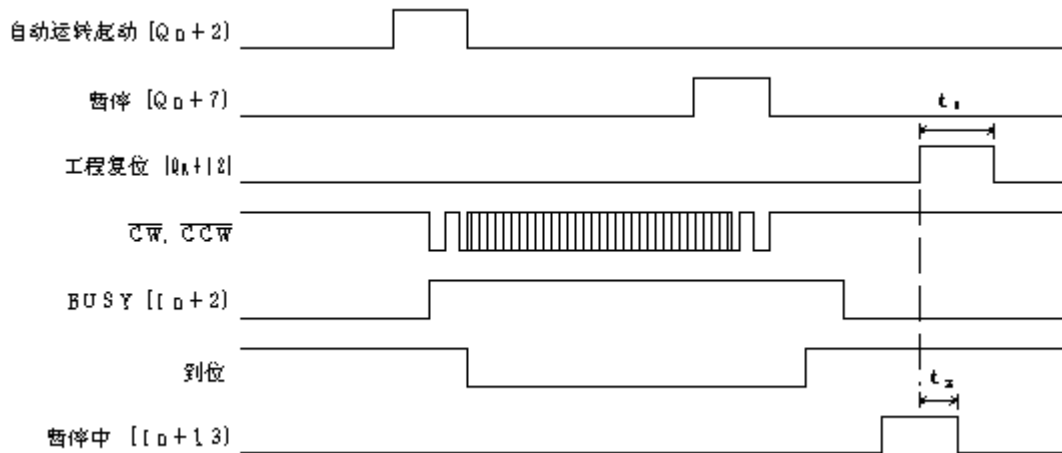


- t 1 : 1次扫描以上
- t 2 :
- t 3 :
- t 4 : (到位信号H时的延迟时间)
- t 5 :



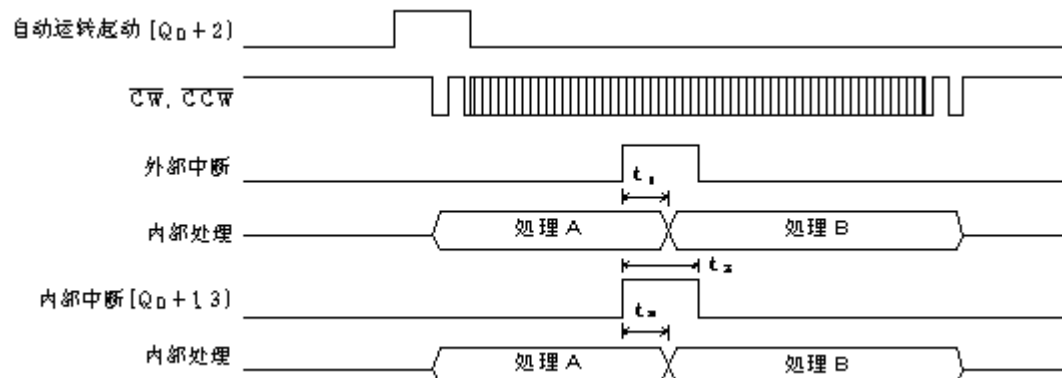
t 1: 1次扫描以上
t 2: (減速開始時間)
t 3:
t 4:

《自动运转（工程复位）》



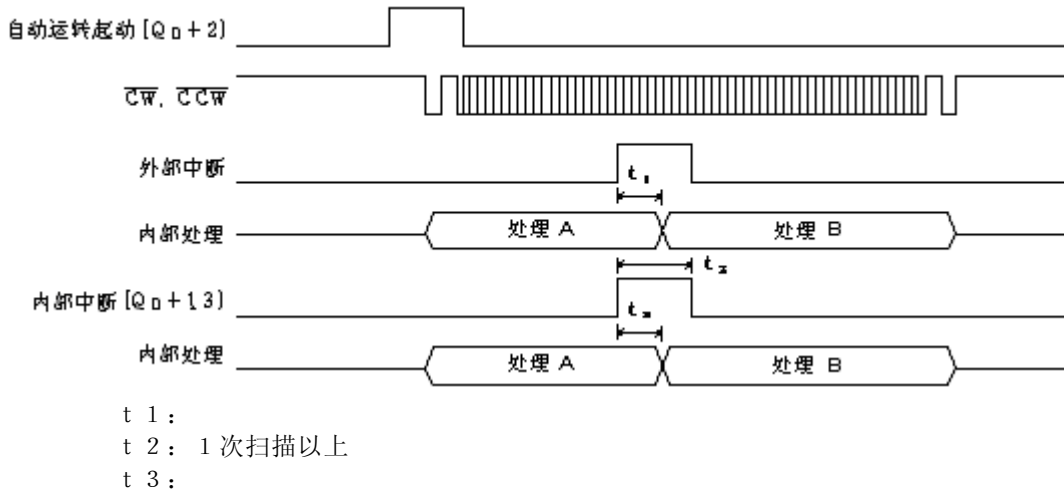
t 1: 1次扫描以上
t 2:

《自动运转（中断处理）》

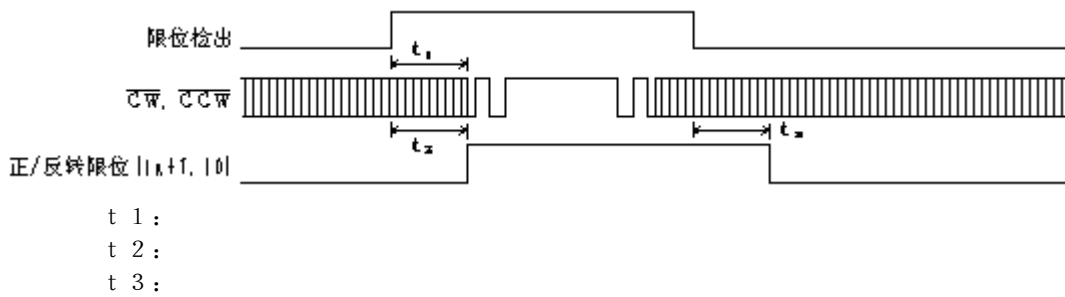


t 1:
t 2: 1次扫描以上
t 3:

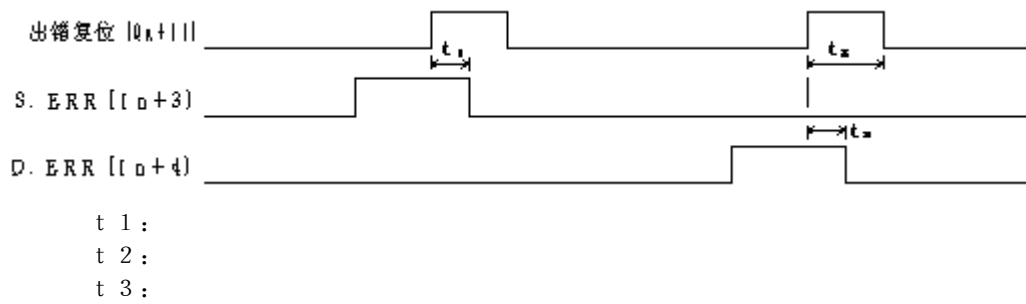
《自动运转（辅助码）》



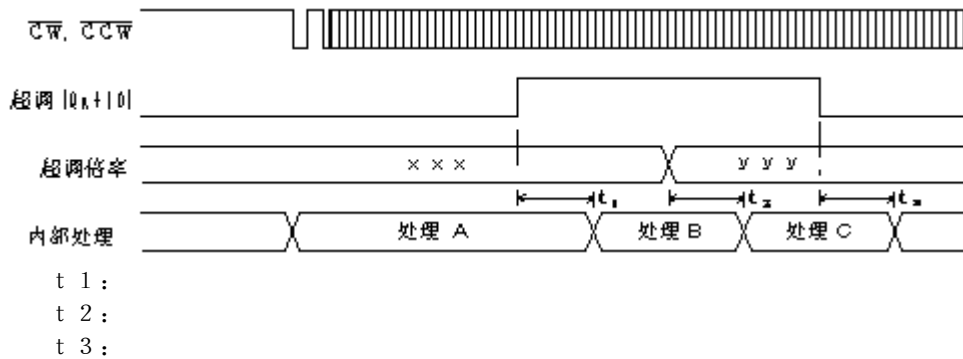
《限位检出》



《出错复位》



《超调》



1 6 - 2 . 状态迁移一览

功 能 \ 状 态		状 态										
		0	1	2	3	4	5	6	7	8	9	
Q n + 0 E N A B L E	O N	×	②	—	—	—	—	—	—	—	—	×
	O F F	×	—	⑧	⑧	⑧	⑧	⑧	⑧	⑧	—	×
Q n + 1 原点检索起动	O N	×	—	⑤	—	—	—	—	—	—	—	×
	O F F	×	—	—	—	—	②	—	—	—	—	×
Q n + 2 自动运转起动	O N	×	—	⑥	—	—	—	—	⑥	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 3 手动正转起动	O N	×	—	③	—	—	—	—	—	—	—	×
	O F F	×	—	—	②	—	—	—	—	—	—	×
Q n + 4 手动反转起动	O N	×	—	③	—	—	—	—	—	—	—	×
	O F F	×	—	—	②	—	—	—	—	—	—	×
Q n + 5 点动正转起动	O N	×	—	④	—	—	—	—	—	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 6 点动反转起动	O N	×	—	④	—	—	—	—	—	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 7 暂停	O N	×	—	—	—	—	—	⑦	—	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 1 0 超调	O N	×	—	—	Y	—	—	Y	—	—	—	×
	O F F	×	—	—	Y	—	—	Y	—	—	—	×
Q n + 1 1 出错复位	O N	×	Y	Y	—	—	—	—	—	①	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 1 2 工序复位	O N	×	—	Y	—	—	—	—	②	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 1 3 内部中断	O N	×	—	—	—	—	—	Y	—	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×
Q n + 1 4 辅助码清除	O N	×	—	Y	—	—	—	Y	Y	—	—	×
	O F F	×	—	—	—	—	—	—	—	—	—	×

—：無視
×：不参照

○：状态迁移
Y：进行处理

功 能 \ 状 态		状 态									
		0	1	2	3	4	5	6	7	8	9
正转限位输入 CWL	ON	×	Y*1	Y	②	②	Y	②	—	—	×
	OFF	×	—	—	—	—	—	—	—	—	×
反转限位输入 CCWL	ON	×	Y*1	Y	②	②	Y	②	—	—	×
	OFF	×	—	—	—	—	—	—	—	—	×
原点挡块输入 HOME	ON	×	—	—	—	—	Y	Y*2	—	—	×
	OFF	×	—	—	—	—	Y	Y*2	—	—	×
紧急停止输入 EMR	ON	×	Y*1	⑧	⑧	⑧	⑧	⑧	⑧	—	×
	OFF	×	—	—	—	—	—	—	—	—	×
出错输入 ERR	ON	×	Y*1	⑧	⑧	⑧	⑧	⑧	⑧	—	×
	OFF	×	—	—	—	—	—	—	—	—	×
到位输入 INP	ON	×	—	Y	Y	Y	Y	Y	Y	—	×
	OFF	×	—	—	—	—	—	—	—	—	×
中断输入 INT	ON	×	—	—	—	—	—	Y	—	—	×
	OFF	×	—	—	—	—	—	—	—	—	×
Z相输入 Z	ON	×	—	—	—	—	Y	Y*2	—	—	×
	OFF	×	—	—	—	—	Y	Y*2	—	—	×
通用输入 IN*	ON	×	—	—	—	—	—	Y	—	—	×
	OFF	×	—	—	—	—	—	Y	—	—	×
外部24V输入 24VPF	ON	×	Y*1	⑧	⑧	⑧	⑧	⑧	⑧	—	×
	OFF	×	—	—	—	—	—	—	—	—	×

—：無視

○：状态迁移

×：不参照

Y：进行处理

*1：ENABLE OFF→ON時参照

*2：仅G28执行时参照

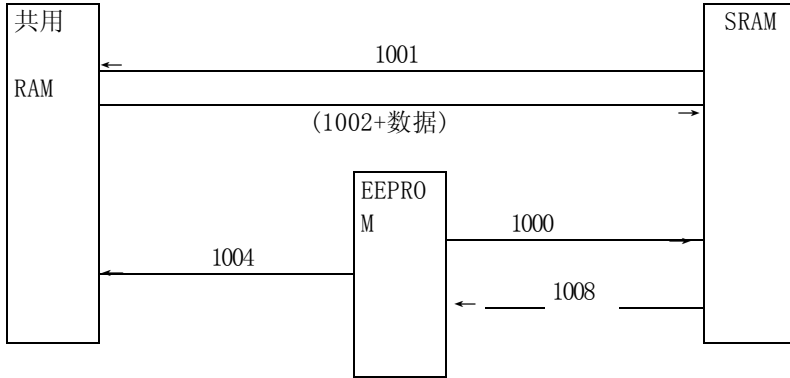
1 6 - 3 . 存储器映像

FFFFFFFh	内部 R A M		
FFFFF000h			00000000-0001FFFFh:ON-CHIP ROM(128K Byte)
FFFFFFFh	预约		00020000-001FFFFFh:RESERVE
FFFF8800h			00200000-00207FFFh:SRAM(32K Byte)
FFFF87FFh	内装外围模块		00208000-003FFFFFFFh:SRAM IMAGE
FFFF8000h			00400000-00407FFFh:EEPROM(32K Byte)
FFFF7FFFh	预约		00408000-007FFFFFFFh:EEPROM IMAGE
00FF8000h			00800000-008007FFFh:SHARED RAM(2K Byte)
00FF7FFFh	L C A 图像	↑	00800800-00BFFFFFFFh:SHARED RAM IMAGE
00C00008h		区域	00C00000-00C00007h:LCA(8 Byte)
00C00007h	L C A	3	00C00008-00FF7FFFh:LCA IMAGE
00C00000h		↓	00FF8000-FFFF7FFFh:RESERVE
00BFFFFFFh	共用 R A M 图像	↑	FFFF8000-FFFF87FFh:REGISTER
00800800h		区域	FFFF8800-FFFFEFFFh:RESERVE
008007FFh	共用 R A M	2	FFFFF000-FFFFFFFFh:ON-CHIP RAM(4K Byte)
00800000h		↓	
007FFFFFFh	E E P R O M 图像	↑	
00408000h		区域	
00407FFFh	E E P R O M	1	
00400000h		↓	
003FFFFFFh	S R A M 图像	↑	
00208000h		区域	
00207FFFh	S R A M	0	
00200000h		↓	
001FFFFFh	预约	↑	
00020000h		区域	
0001FFFFh	内藏 R O M	↓	
00000000h			

详情参阅 E S - A 4 5 3 - * 硬件设计规格书。

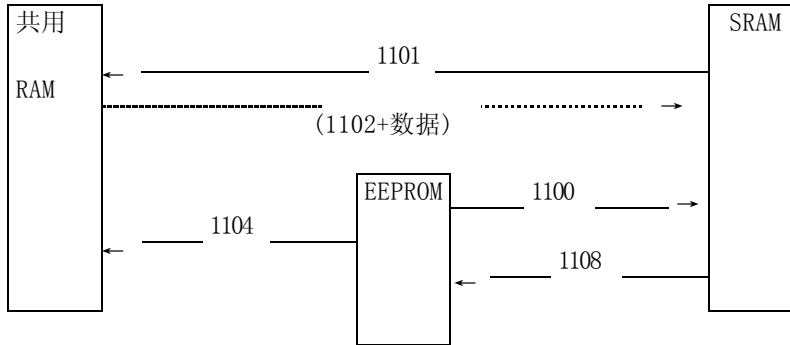
1 6 - 4 . 用 W T 方式的数据传送概要

1) 系统参数



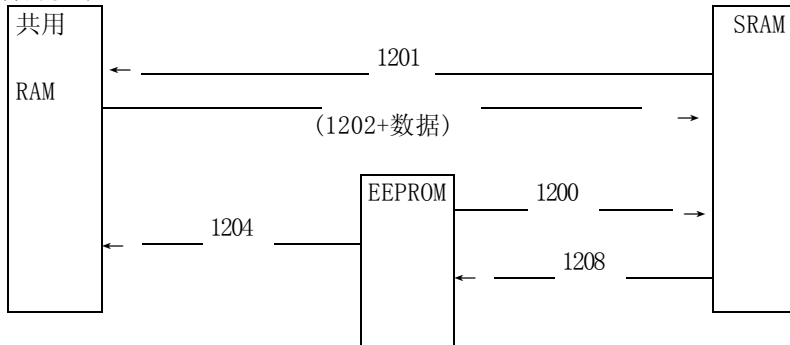
注) 数据数同时处理「50字节」。
由WT传送「1002」时，
传送52字节。

2) 原点参数



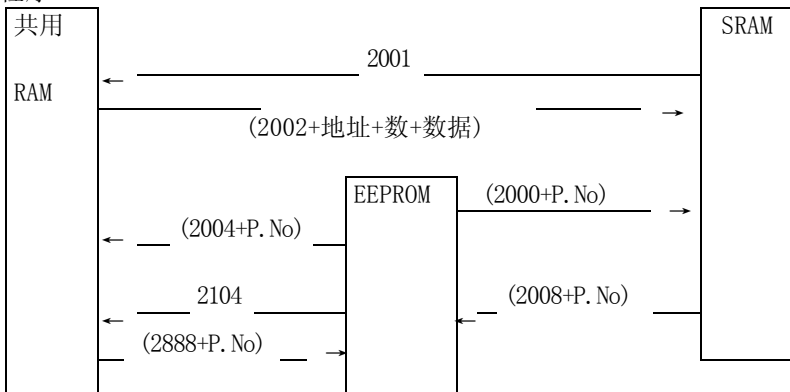
注) 数据数同时处理「18字节」。
由WT传送「1102」时，
传送20字节。

3) 特殊参数

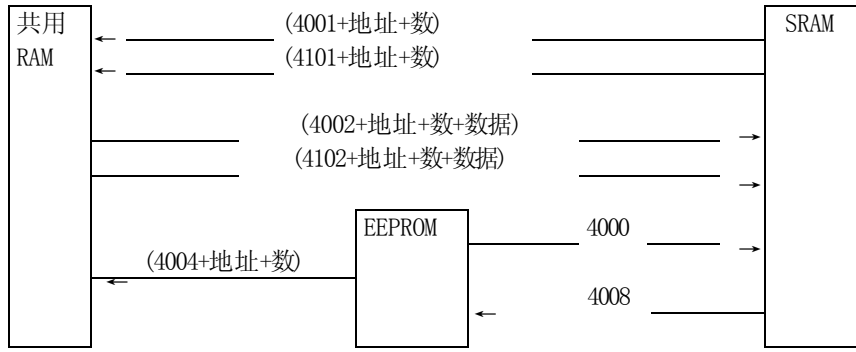


注) 数据数同时处理「10字节」。
由WT传送「1202」时，
传送12字节。

4) 程序



5) 数据寄存器



17. 安全方面的注意事项

〔使用环境和条件〕

- 不要在易燃、易爆的环境中使用，以免引起人身事故和火灾。
- 本产品不要用于有关人身安全的用途。使用于即使万一有故障或误动作，也不会危及人体的用途。
- 在规格规定的环境（振动、冲击、温度、湿度等）的范围内使用，保管，以免引起火灾或损坏产品。
- 要在了解产品后再使用。

〔安装及接线〕

- 要正确接线。以免引起火灾或故障。
- 不可使用规定以外的电源电压，以免引起火灾，触电或故障。
- 要按规定进行接线和配置，以免引起火灾或故障。
- 接线方法要不产生应力，以免引起触电或火灾。
- 接线要在电源切断的状态下进行，以免引起触电或故障。

保证期和保证范围

〔保证期间〕

供货保证期为交货到订货者指定场所后 1 年。

〔保证范围〕

在上述保证期内，由于供货方的责任发生故障时，该器件故障部分的更换，或修理由供货方负责。但下述情况不属于保证对象范围：

- （1）因使用方保管和使用不当引起的情况。
- （2）故障的原因不是由于所供产品引起。
- （3）由供货者以外进行改造，修理引起的情况。
- （4）其他由于天灾，灾害等不是供货者责任的情况。

此外，这里所说的保证，仅指所供物品本身的保证，不包括因所供物品的故障诱发的损失。

光洋电子(无锡)有限公司

Koyo ELECTRONICS (WUXI) CO., LTD.

地址：江苏省无锡市滨湖区建筑西路 599 号 1 栋 21 层

邮编：214072

电话：0510-85167888

传真：0510-85161393

<http://www.koyoele.com.cn>

KEW-M3232A

2015 年 8 月