

R 系列编程手册

目录

R 系列编程手册.....	1
1. 概要.....	3
1.1 指令的构成.....	3
1.2 数据的指定方法.....	4
1.3 执行条件.....	13
2. 编程时注意事项.....	14
2.1 指令通用的出错.....	14
2.2 执行指令时的软元件、标签的范围检查.....	14
2.3 使用同一软元件的 OUT 指令、SET/RST 指令、PLS/PLF 指令时的动作.....	15
2.4 使用计数器时的注意事项.....	17
3 指令一览表.....	22
3.1 概要.....	22
3.2 基本指令.....	22
3.3 应用指令.....	26
3.4 高速处理及定位.....	31
3.5 通讯指令.....	32
4. 软元件的作用于功能.....	33
4.1 R 系列软元件一览.....	33
4.2 R 系列软元件分配一览.....	34
4.3 用户软元件.....	35
4.2 系统软元件.....	51
4.3 变址寄存器 (Z/LZ).....	52
4.4 文件寄存器 (R).....	52
4.5 嵌套 (N).....	53
4.6 指针 (P).....	53
4.7 中断指针 I.....	53
4.8 常数.....	56
5. 指令用法简介.....	57
4.1 基本指令.....	57
4.2 结合指令.....	64
4.3 输出指令.....	67
4.4 主控指令.....	76
4.5 程序流程.....	77
4.6 触点比较指令.....	91
4.7 比较输出.....	98
4.8 二进制逻辑运算.....	104
4.9 数据块运算.....	121
4.10 浮点运算.....	129

4.11 数据转换.....	155
4.12 数据传送.....	174
4.13 循环, 移位.....	183
4.14 数据处理.....	186
4.15 时钟运算.....	209
4.16 数据表操作指令.....	217
4.17 运动控制.....	247
4.18 高速计数和脉冲测量.....	260
4.19 高速处理.....	265
4.20 凸轮顺控.....	268
4.21 模块访问指令.....	274
4.22 MODBUS	277
4.23 自由通信.....	278
4.24 高速计数.....	287
4.25 高速输出.....	296
6. 特殊软元件介绍.....	297
6.1 特殊继电器	297
6.2 特殊数据寄存器	310
7. PLC 出错代码.....	325
其他指令, 后续扩充	328

1. 概要

1.1 指令的构成

R 系列小型 PLC 中可使用的指令大部分可以分为指令和软元件两个部分，本文将对目前已经完成的指令做详细介绍，后续会扩充新加入的指令。

指令和软元件两个部分其作用如下：

- 1、指令部分：指令功能如下文所示
- 2、软元件部分：指令中使用的数据如下所示

注：其中软元件部分分为源数据，目标数据，数值数据

1.1.1 源数据 (S)

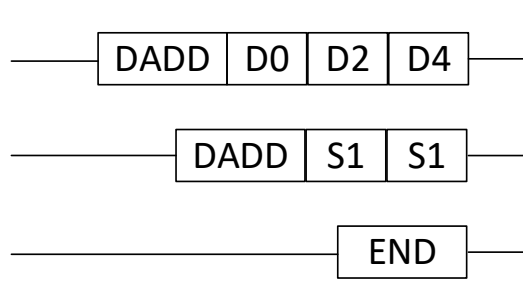
是运算中使用的数据，根据各指令中指定的标签及软元件，其情况如下所示。

种类	内容
常数	指定运算中使用的数值 由于是在创建程序时进行设置，因此在程序执行中不能更改
位软元件 字软元件	指定存储运算中使用的数据的软元件 在执行运算之前需要减数据预先存储到指定的软元件中 程序执行中，通过更改指定的软元件中存储的数据，可以更改该指令中使用的数据

1.1.2 目标数据 (D)

目标中存储运算后的数据

但是，根据指令，有时在运算前需要在目标中存储运算中使用的数据

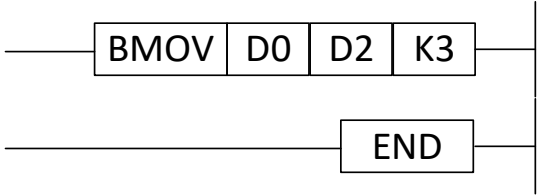


图中，DADD 加法指令指令中字软元件 D0,D2 一直是源数据 (S)，D4 是目标数据(D)，在使用时，需要在运算前在 D0,D2 中储存数据。

而 BIN16 位加法运算指令中，D0,D2 运算前是源数据 (S)，D2 在运算后又作为目标数据 (D) 储存了运算后的值。

1.1.3 软元件数/传送数/数据数/字符串数等的数值(n)

在指定使用多个软元件的指令及指定重复次数、指定处理的数据数·字符串数等的指令中、指定软元件数、传送数、数据数、字符串数等时、使用数值。



在如上指令中，D0,D2 分别为源数据（S）和目标数据（D），而 K3 为通过 BMOV 指令指定的传送数（N）

软元件数、传送数、字符串数等的容量指定允许范围为 0 ~ 65535、0 ~ 4294967295。*1
但是，软元件数、传送数、字符串数等的容量指定为 0 的情况下，该指令将变为无处理。

注：1 设置范围根据指令而有所不同。详细内容请参阅各指令。

2 传送数等的数值中使用较大值的情况下，扫描时间将延迟，应加以注意

1.2 数据的指定方法

R 系列小型 PLC 支持的各指令可使用的数据有以下这些类型

数据	分类
位数据	位数据
16 位数据	有符号 16 位数据
32 位数据	有符号 32 位数据
实数数据	单精度实数数据
BCD 数据	BCD4 位数据
	BCD8 位数据
字符串数据	字符串

作为软元件时处理的数据

数据类型	内容	可指定的软元件、常数*1
位	可以处理位数据	• 位软元件 • 字软元件的位指定
字	可以处理字数据	• 字软元件
带符号 BIN16 位	可以处理 16 有符号位数据。	• 位软元件的位数指定(K1 ~ K4) *2
带符号 BIN32 位	2 位数据或 16 位有符号数据可以处理 2 个连续的数据。	• 10 进制常数 • 16 进制常数 • 字软元件 • 双字软元件 • 位软元件的位数指定(K1 ~ K8) *2 • 10 进制常数 • 16 进制常数
双字	可以处理双字数据	• 双字软元件
BCD4 位	可以处理 BCD4 位数据。	• 字软元件

	将 16 位数据以 4 位数进行分割，将各位以 0 ~ 9 进行指定	• 位软元件的位数指定(K1 ~ K4) *2 • 10 进制常数 • 16 进制常数
BCD8 位	可以处理 BCD8 位数据。 将 32 位数据以 8 位数进行分割，将各位以 0 ~ 9 进行指定	• 字软元件 • 双字软元件 • 位软元件的位数指定(K1 ~ K8) *2 • 10 进制常数 • 16 进制常数
单精度实数	可以处理单精度实数数据(单精度浮点数据)	• 字软元件 • 双字软元件 • 实数常数
字符串	可以处理 ASCII 代码的字符串数据	• 字软元件 • 字符串常数

注：*1 通过指令进行源(s)及数值数据(n)指定的数据中，可以使用常数。

*2 关于指定方法，请参阅各数据类型的详细页面。

作为标签处理的数据

基本数据类型

数据类型(标签)	可指定的标签
位 (BOOL)	• 位型标签 • 字[带符号]型标签的位指定 • 定时器/累计定时器型标签的触点·线圈 • 计数器/超长计数器型标签的触点·线圈
字[带符号] (INT)	• 字[带符号]型标签 • 位型标签的位数指定(K1 ~ K4) • 定时器/累计定时器型标签的当前值 • 计数器型标签的当前值
双字[带符号] (DINT)	• 双字[带符号]型标签 • 位型标签的位数指定(K1 ~ K8) • 超长计数器型标签的当前值
单精度实数 (REAL)	• 单精度实数型标签
时间 (TIME)	• 时间型标签
字符串 (STRING)	• 字符串型标签
定时器 (TIMER)	• 定时器标签
累计定时器 (RETENTIVETIMER)	• 累计定时器标签
计数器 (COUNTER)	• 计数器型标签
超长计数器 (LCOUNTER)	• 超长计数器型标签
指针 (POINTER)	• 指针型标签

1.2.1 位数据

位数据的容量和数据范围

位数据是将触点及线圈等以 1 位单位处理的数据

数据名称	数据容量	值的范围
位数据	1 位	0、1

位软元件/标签中位数据的处理

每 1 点可处理 1 点的位数据

字软元件中位数据的处理

对字软元件通过进行位 No.指定，可以处理指定位 No.的位数据。

位指定的表示方法为“字软元件编号.位 No.”。

位 No.的指定是以 16 进制数在 0 ~ F 的范围内进行指定。

例如，D0 的位 5(b5)以“D0.5”进行指定，D0 的位 10(b10)以“D0.A”进行指定。

可进行字软元件的位指定的软元件如下所示

项目	软元件
可进行字软元件的位指定的软元件	• 数据寄存器(D) • 链接寄存器(W) • 链接特殊寄存器(SW) • 特殊寄存器(SD) • 文件寄存器(R)

字型标签中位数据的处理

在字[带符号]型标签中，通过进行位 No.指定，可以处理指定位 No.的位数据。

位指定的表示方法为“标签名.位 No.”。

1.2.2 16 位数据（字数据）

数据容量及数据的范围

16 位数据中，只有带符号 16 位数据

对于带符号 16 位数据，负数以 2 的补数表示。

数据名称	数据容量	值的范围	
		10 进制表示	16 进制表示
有符号 16 位数据	16 位(1 字)	-32768 ~ 32767	0000H ~ FFFFH

位软元件中 16 位数据的处理

对位软元件通过进行位数指定，可以作为 16 位数据处理

项目	表示	示例
位软元件	KN 修饰：位软元件起始编号	K4X10 K2M113

	N: 位数(在 1 ~ 4 的范围内指定)	
--	-----------------------	--

位型数组的标签中 16 位数据的处理

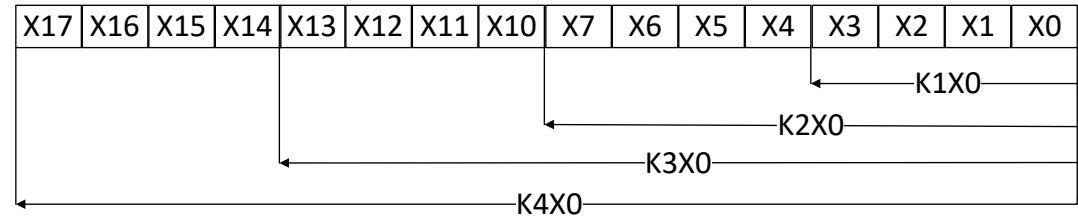
对位型数组的标签通过进行位数指定，可以作为 16 位数据处理。

对位型数组的标签进行位数指定作为 16 位数据处理时的表示方法如下所示。

位数指定	10 进制表示	16 进制表示
K1	0 ~ 15	0H ~ FH
K2	0 ~ 255	00H ~ FFH
K3	0 ~ 4095	000H ~ FFFH
K4	带符号 16 位数据: -32768 ~ 32767	0000H ~ FFFFH

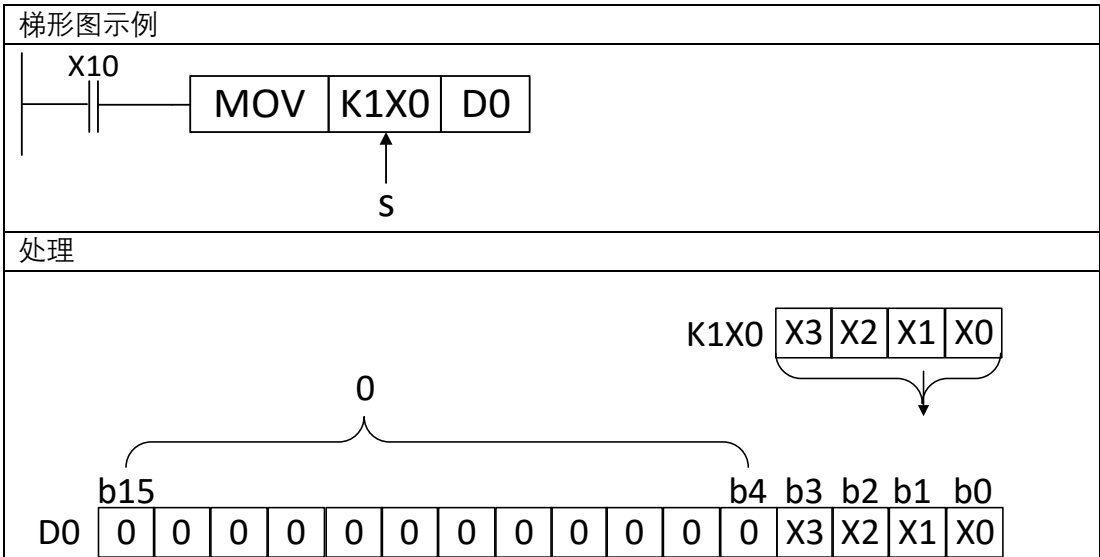
例：对 X0 进行了位数指定情况下的对象点数如下所示。

- K1X0→X0 ~ X3 的 4 点为对象
- K2X0→X0 ~ X7 的 8 点为对象
- K3X0→X0 ~ X13 的 12 点为对象
- K4X0→X0 ~ X17 的 16 点为对象



源数据(s)侧指定了位数指定的位软元件的情况下

在指令的源侧指定了位数指定的位软元件的情况下，目标侧的字软元件中，源侧进行了位数指定的位以后的高位中将存储 0。



目标数据(d)侧指定了位数指定的位软元件的情况下

指令的目标侧有位数指定的情况下，位数指定的点数将成为目标侧的对象。

位数指定的点数以后的位软元件不变化。

梯形图示例

源数据为字软元件软元件

X10

MOV

D0

K2M100

d

处理

b15

....

b8 b7

....

b0

D0

1

1

1

0

1

0

1

0

1

0

0

1

1

1

0

1

M115

....

M108 M107

....

M100

D0

1

0

0

1

1

1

0

1

(1)

(1)不变化

字软元件/标签中 16 位数据的处理

字软元件

1 点字软元件可处理 16 位数据

字型标签

1 点字型标签中可处理 16 位数据。

1.2.3 位数据(双字数据)

数据容量及数据的范围

32 位数据中，只有带符号 32 位数据

对于带符号 32 位数据，负数以 2 的补数表示。

数据名称	数据容量	值的范围	
		10 进制表示	16 进制表示
带符号 32 位数据	32 位(2 字)	-2147483648 ~ 2147483647	00000000H ~ FFFFFFFFH

位软元件中 32 位数据的处理

对位软元件通过进行位数指定，可以作为 32 位数据处理

项目	表示	示例
位软元件	KNX 位软元件起始编号 N: 位数(在 1 ~ 8 的范围内指定)	K8X10 K6B018

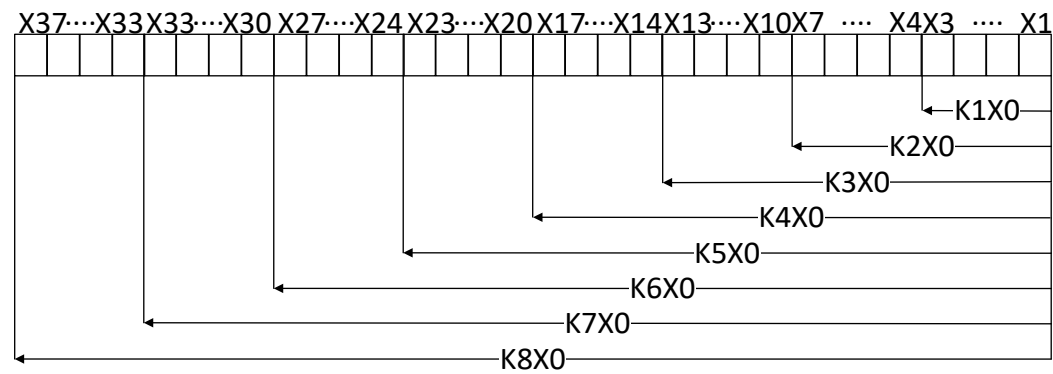
位数指定的范围

各位数指定的 32 位数据的范围如下所示。

位数指定	10 进制表示	16 进制表示
K1	0 ~ 15	0H ~ FH
K2	0 ~ 255	00H ~ FFH
K3	0 ~ 4095	000H ~ FFFH
K4	0 ~ 65535	0000H ~ FFFFH
K5	0 ~ 1048575	00000H ~ FFFFFH
K6	0 ~ 16777215	000000H ~ FFFFFFFH
K7	0 ~ 268435455	0000000H ~ FFFFFFFFH
K8	有 符号 32 位 数据：-2147483648 ~ 2147483647	00000000H ~ FFFFFFFFH

例：对 X0 进行了位数指定情况下的对象点数如下所示。

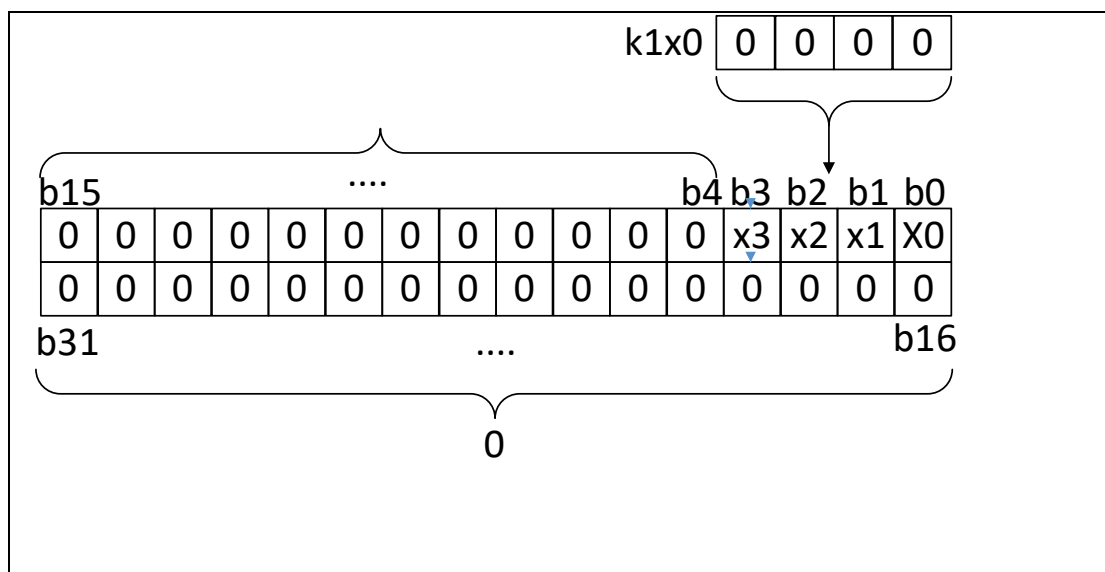
- K1X0→X0 ~ X3 的 4 点为对象
- K2X0→X0 ~ X7 的 8 点为对象
- K3X0→X0 ~ X13 的 12 点为对象
- K4X0→X0 ~ X17 的 16 点为对象
- K5X0→X0 ~ X23 的 20 点为对象
- K6X0→X0 ~ X27 的 24 点为对象
- K7X0→X0 ~ X33 的 28 点为对象
- K8X0→X0 ~ X37 的 32 点为对象



源数据(s)侧指定了位数指定的位软元件的情况下

在指令的源侧指定了位数指定的位软元件的情况下，目标侧的字软元件中，源侧进行了位数指定的位以后的高位中将存储 0。

梯形图示例
32 位指令情况下
处理



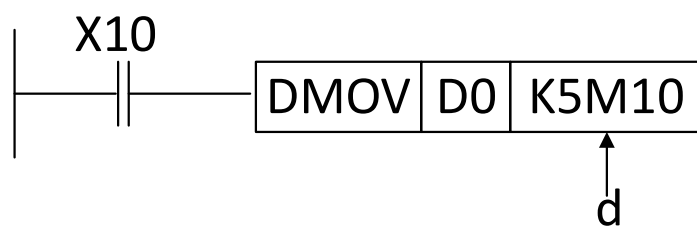
目标数据(d)侧指定了位数指定的位软元件的情况下

指令的目标侧有位数指定的情况下，位数指定的点数将成为目标侧的对象。

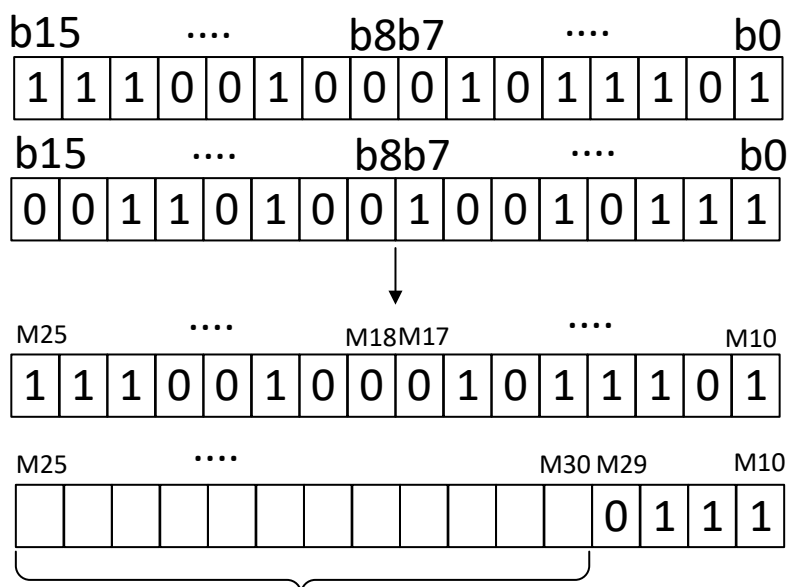
位数指定的点数以后的位软元件不变化。

梯形图示例

源数据为字软元件的情况下



处理



(1)

(1)不变化

字软元件/标签中 32 位数据的处理

字软元件

2 点的字软元件可作为 32 位数据处理。

但是，对于下述软元件，1 点软元件可处理 32 位数据。

- 超长计数器(LC)
- 超长变址寄存器(LZ)

双字型标签

双字型标签的 1 点可作为 32 位数据处理。

1.2.4 实数数据(浮点数据)

数据容量及数据的范围

实数数据中，有以 32 位表示的单精度实数数据。

实数数据只能存储到位软元件以外的软元件或单精度实数型标签中。

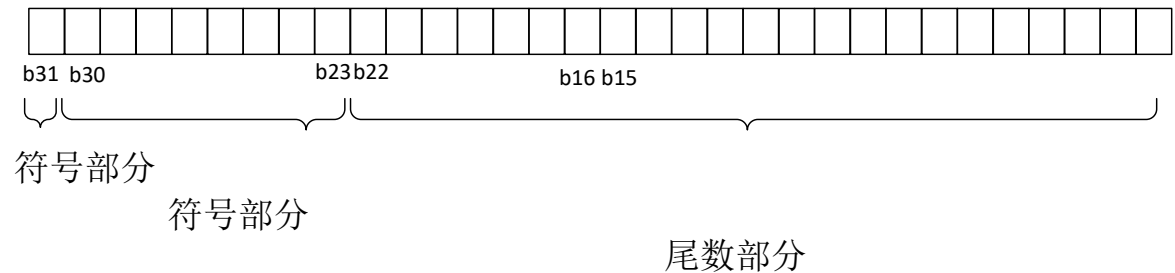
数据名称		数据容量	值的范围
单精度实数数据(单精度浮点数据)	正数	32 位(2 字)	$2^{-126} \leq \text{实数} < 2^{128}$
	零		0
	负数		$-2^{128} < \text{实数} \leq -2^{-126}$

单精度实数数据的构成

单精度实数数据由符号部、尾数部、指数部所构成，按以下方式表示。

符号部分 1. 尾数部分 * 2^{指数部分}

单精度实数数据内部表示时的位构成及含义如下所示



符号部(1 位)

数值的正负以 1 位表示。0 表示正数或 0。1 表示负数。

尾数部(23 位)

将单精度实数以 2 进制数 $1.XXXXX \cdots \times 2^N$ 表示时的 $XXXXX \cdots$ 如下所示。

指数部(8 位)

将单精度实数以 2 进制数 $1.XXXXX \cdots \times 2^N$ 表示时的 N 如下所示。单精度实数中指数部的值与 N 的关系如下所示。

指 数	FFH	FEH	FDH	...	80H	...	02H	01H	00H
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

部 (b24 ~ b30)									
N	未 使 用	127	126	...	1	...	-125	-126	未 使 用

注意事项

通过工程工具设置单精度实数的输入值的情况下

在工程工具中，对单精度实数数据以 32 位的单精度进行，因此有效位数约为 7 位数。

单精度实数型数据的输入值超过了 7 位数的情况下，第 8 位数将被进行四舍五入。

因此，四舍五入后的值超过了 -2147483648 ~ 2147483647 的范围时，将不会变为所期望的值。

?

将输入值设置为“2147483647”的情况下，第 8 位数的“6”将被四舍五入，因此处理为“2147484000”。

?

将输入值设置为“E1.1754943562”的，第 8 位数的“3”将被四舍五入，因此处理为“E1.175494”

1.2.5 字符串数据

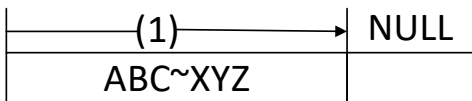
字符串数据的形式

字符串数据中有下述类型，可以将以 NULL 代码结束的字符代码串作为字符串处理

种类	字符代码	最终字符
字符串	ASCII 码	NULL(00H)

将字符串数据存储到软元件或数组中时，将以从软元件编号或数组的要素编号的低位至高位的顺序存储字符串数据。

低位————→高位



↑
‘ABC~XYZ’

(1) 字符代码列

数据的范围

字符串数据的范围如下所示

种类	最大字符串数	程序上可处理的最大字符串数
字符串	半角 255 字符(不包括终端的 NULL。)	16383 字符(不包括终端的 NULL。)

数据存储中必要的字数

字符串数据可存储到字软元件中。

字符串数据的存储必要字数如下所示。

字符串的字节数	用于存储字符串的必要字数
---------	--------------

0 字节	1[字]
奇数字节	(字符串的字节数+1)÷2[字]
偶数字节	(字符串的字节数÷2)+1[字]

字符串数据的存储目标

字符串数据的存储目标示意图如下所示。

字符串的情况下

在字符串的存储示意图中，NULL 表示 NULL 代码(00H)。

存储的字符串	从 D0 开始存储了字符串数据时的示意图	
' '(空字符串)	D0	NULL NULL
'ABC'	D0 D1	B A NULL C
'ABCD'	D0 D1 D2	B A D C NULL NULL
	从字型标签的数组 arrayA[0]开始存储了字符串数据时的示意图	
' '(空字符串)	Array[0]	NULL NULL
'ABC'	Array[0] Array[1]	B A NULL C
'ABCD'	Array[0] Array[1] Array[2]	B A D C NULL NULL

1.3 执行条件

1.3.1 执行条件的类型

R 系列小型 PLC 的指令及函数的执行条件中，有下述 5 种类型。

■ ON 时执行

是 ON 中执行型的指令，仅在指令的前条件为 ON 期间执行该指令。前条件为 OFF 的情况下，不执行该指令，不进行处理。

■ 上升沿执行

是仅在 ON 时执行 1 次型的指令，仅在指令的前条件的上升沿时(OFF→ON)执行指令，以后即使条件为 ON 也不执行该指令，不进行处理。

■ OFF 时执行

是 OFF 中执行型的指令，仅在指令的前条件为 OFF 期间执行该指令。前条件为 ON 的情况

下，不该执行指令，不进行处理。

■ 下降沿执行

是 OFF 时执行 1 次型的指令，仅在指令的前条件的下降沿时(ON→OFF)执行指令，以后即使条件为 OFF 也不执行该指令，不进行处理。

■ 常时执行

是常时执行的指令，与指令的前条件的 ON/OFF 无关，始终执行。前条件为 OFF 的情况下，该指令将进行 OFF 处理。

1.3.2 各指令的执行条件

各指令中执行条件有所不同。各指令的执行条件如下所示。

执行条件	相应指令
ON 时执行	下述以外的所有指令
上升沿执行	• 指令符号附加了“(P)”的指令 • PLS
OFF 时执行	-
下降沿执行	PLF
常时执行	LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF、LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI、ORB、INV、MEP、MEF、OUT、OUT T、OUTH T、OUTHST、OUTST、OUTHST、OUTHST、MC、MCR、END、

2.编程时注意事项

2.1 指令通用的出错

出错内容	出错代码 (SD4/诊断)
6004	软元件使用超范围
6710	命令中多个参数范围重叠
8030	程序结构性错误

2.2 执行指令时的软元件、标签的范围检查

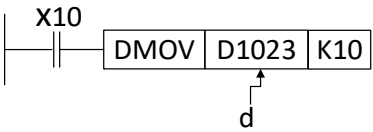
软元件、标签的范围检查

通过指令指定了软元件及标签的情况下，进行范围的检查，如果超出相应软元件/标签的范围，则发生出错。

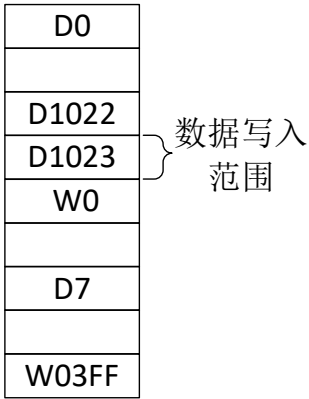
将软元件中分配的标签通过程序中的指令进行了指定的情况下也相同。

请创建其运算结果不会超出相应软元件/标签的范围的程序

例：在指定了全局软元件的情况下



软元件储存器分配示意图



d 传送目标的相应范围为 D1023 ~ D1032。由于 D1024 ~ D1032 不存在，因此仅写入 D1023。

2.3 使用同一软元件的 OUT 指令、SET/RST 指令、PLS/PLF 指令时的动作

以下介绍将使用了同一软元件的 OUT 指令、SET/RST 指令、PLS/PLF 指令在 1 个扫描中执行多次情况下的动作有关内容。

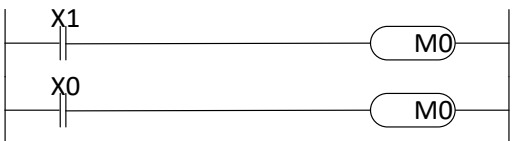
2.3.1 同一软元件的 OUT 指令的情况下

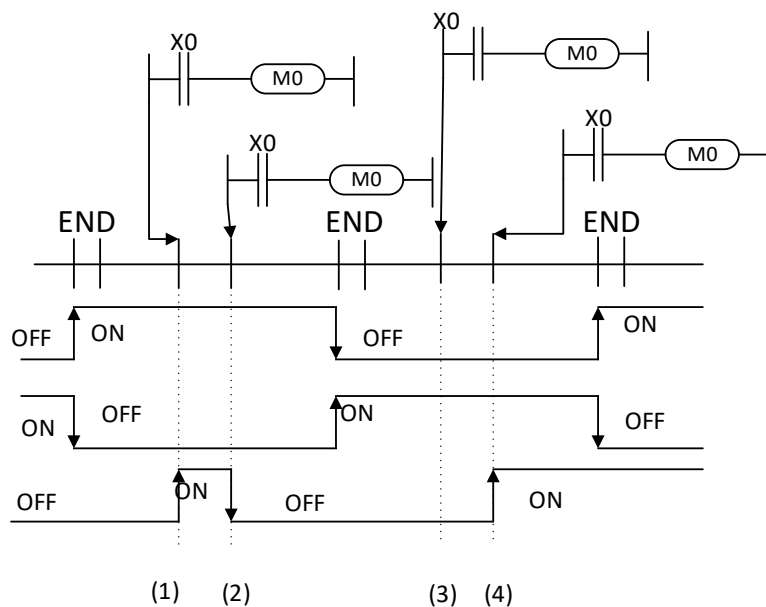
1 个扫描中请勿多次执行同一软元件的 OUT 指令。

1 个扫描中多次执行了同一软元件的 OUT 指令的情况下，执行各个 OUT 指令时，根据 OUT 指令为止的运算结果，指定软元件将变为 ON/OFF。

执行各个 OUT 指令时，指定软元件的 ON/OFF 是确定的，因此 1 个扫描中有可能重复进行 ON/OFF。

输入的 X0 及 X1 中，创建了将同一内部继电器(M0)置为 ON/OFF 的梯形图情况下的动作如下所示





(1)X0 为 ON，因此 M0 变为 ON。

(2)X1 为 OFF，因此 M0 变为 OFF。

(3)X0 为 OFF，因此 M0 保持为 OFF 不变。

(4)X1 为 ON，因此 M0 变为 ON。

通过 OUT 指令指定输出(Y)时，1 个扫描的最后执行的 OUT 指令的 ON/OFF 状态将被输出。

2.3.2 使用了同一软元件的 SET/RST 指令的情况下

■ SET 指令的情况下

SET 指令在执行指令为 ON 时将指定软元件置为 ON，在执行指令为 OFF 时无处理。

因此将同一软元件的 SET 指令在 1 个扫描中执行了多次的情况下，如果 1 个执行指令为 ON，指定软元件将变为 ON。

■ RST 指令的情况下

RST 指令在执行指令为 ON 时将指定软元件置为 OFF，在执行指令为 OFF 时无处理。

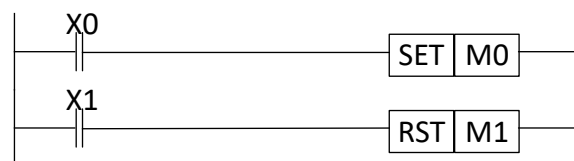
因此将同一软元件的 RST 指令在 1 个扫描中执行了多次的情况下，如果 1 个执行指令为 ON，指定软元件将变为 OFF。

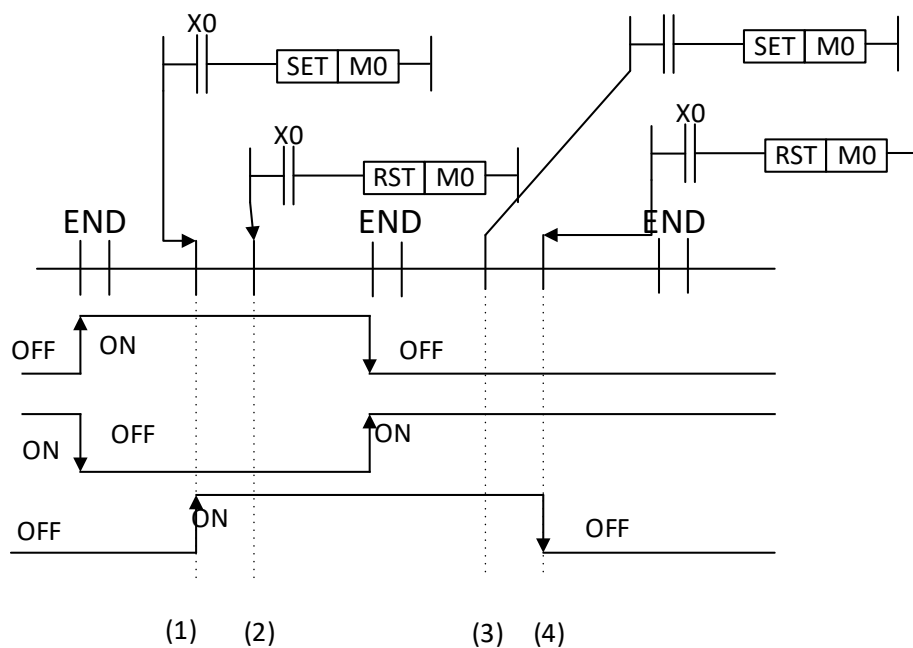
■ 1 个扫描中存在有同一软元件的 SET 指令及 RST 指令的情况下

1 个扫描中存在有同一软元件的 SET 指令及 RST 指令的情况下，SET 指令在执行指令为 ON 时将指定软元件置为 ON，RST 指令在执行

指令为 ON 时将指定软元件置为 OFF。

SET 指令及 RST 指令的执行指令为 OFF 的情况下，指定软元件的 ON/OFF 状态不变化。





- (1)X0 为 ON，因此 M0 变为 ON。
 (2)X1 为 OFF，因此 M0 保持为 ON 不变。(RST 指令将变为无处理。)
 (3)X0 为 OFF，因此 M0 保持为 ON 不变。(SET 指令将变为无处理。)
 (4)X1 为 ON，因此 M0 变为 OFF。

通过 SET/RST 指令指定输出(Y)时，1 个扫描的最后执行的 SET/RST 指令的 ON/OFF 状态将被输出。

2.4 使用计数器时的注意事项

2.4.1 计数器 (C/LC)

1) 计数器的构成

计数器和超长计数器都有被不断更新的当前值 (①)、作为最大计数的目标值 (②)、作为线圈使用的定时器线圈 (③)、和作为触点使用的定时器触点 (④)。各个属性对应符号如下：

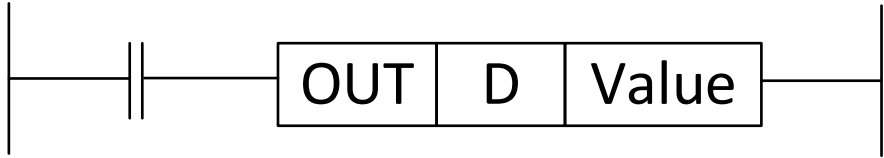
计数器类型	计数器的属性			
	当前值①	目标值②	线圈③	触点④
计数器	CV	CP	CC	CS
超长计数器	LCV	LCP	LCC	LCS

2) 计数器的输出指令

计数器指令之前的运算结果由 OFF→ON 变化时，线圈将由 OFF→ON 变化，将(d)中

指定的计数器的当前值+1，如果计数到(Value)，触点将导通。

■ 指令格式



OUT (d) Value

■ 操作数

操作数	内容	范围	数据类型	掉电锁存
(d)	计数器编号	—	计数器/超长计数器	支持
Value	计数器的目标值	C:0 ~ 32767 LC:0 ~ 2147483647	无符号 BIN16 位	—

■ 可使用软元件

操作数	位	字		双字		常数	
	X、Y、M、L、SM、F、B、SB、S	T、ST、C、D、W、SD、SW、R	Z	LC	LZ	K、H	E
(d)	—	仅支持 C ^{*1}	—	○ ^{*1}	—	—	—
Value	—	○ ^{*2}	—	—	—	○ ^{*3}	—

*1 C/LC 支持 Z、LZ 变址修饰。

*2 不能使用 T、ST、C。

*3 只能使用 10 进制常数(K)。

3) 计数器的复位指令

计数器和超长计数器可以通过 RST 和 ZRST 指令进行复位操作。RST 指令后，计数器 C/LC 进入复位状态。处于复位状态的定时器，各个属性值保持不变，计时停止，OUT 指令无效。

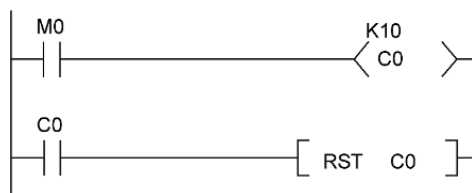


复位后计数器情况如下所示。

	计数器的类型	计数器线圈	计数器当前值	计数器目标值	计数器触点	计数器复位状态
RST	计数器	—	0	保持	OFF	锁定
	超长计数器	—	0	保持	OFF	锁定
ZRST	计数器	OFF	0	保持	OFF	释放
	超长计数器	OFF	0	保持	OFF	释放

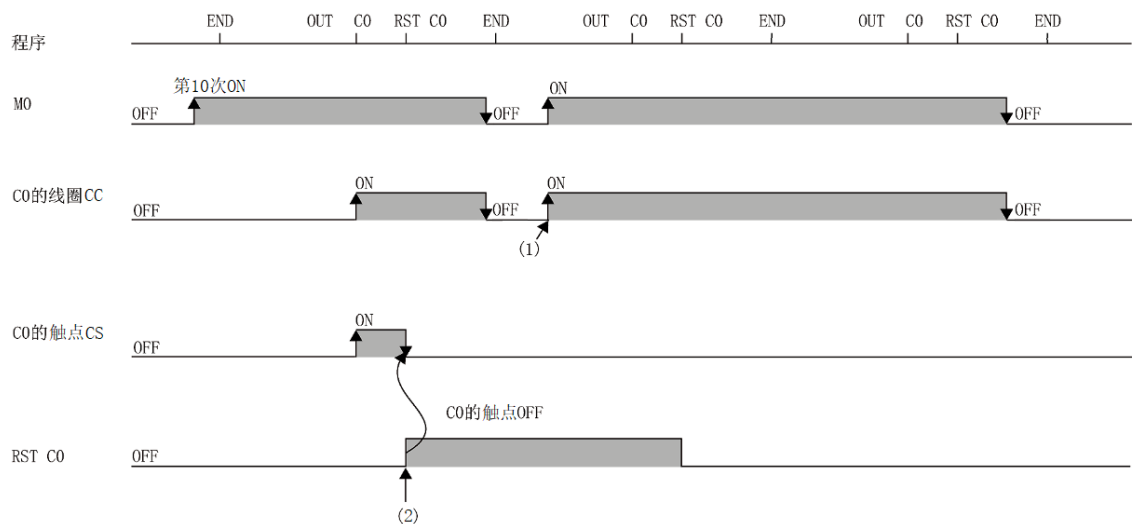
—：计数器线圈(CC/C) 状态与计数器前置运算结果一致

即使将计数器线圈的输入置为 OFF，计数器的当前值也不被清除。计数器当前值的清除(复位)及触点的 OFF 应通过 RST/ZRST 指令进行。在执行了 RST/ZRST 指令的时刻计数器值将被清除，触点也将变为 OFF。



在上述梯形图示例中，通过 M0 的 OFF→ON，C0 的线圈将变为 ON，进行当前值更新。C0 计数到时 C0 的触点将变为 ON，通过 RST C0 指令的执行 C0 的当前值将被清除。C0 的线圈保持 ON。在下一个扫描中 RST C0 指令仍处于执行状态。此时，C0 保持复位状态。

再下一个扫描中 M0 为 ON 的情况下，RST C0 指令不执行，执行 OUT C0 指令时 C0 的线圈将由 OFF→ON，因此将进行当前值的不更新。(当前值不变)



(1) 触点 CC 将变为 OFF→ON, 但 C0 被复位中, 当前值(CV)不能更新

(2) 计数值的清除、触点的 OFF

在 RST 指令执行后, 被复位的计数器此扫描周期内将不被计数。

计数器 C0 不会计数, CS 触点不会导通。

计数器的类型

有将计数器值以 16 位保持的计数器 C 及将计数器值以 32 位保持的超长计数器 LC。

计数器 C 与超长计数器 LC 是不同的软元件, 可以分别设置软元件点数。

■ 计数器 C

1 点使用 1 字。可计数范围为 0~32767。

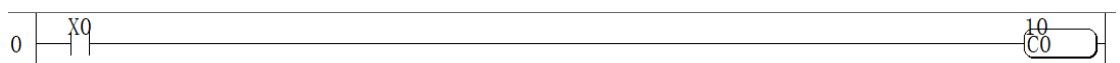
■ 超长计数器 LC

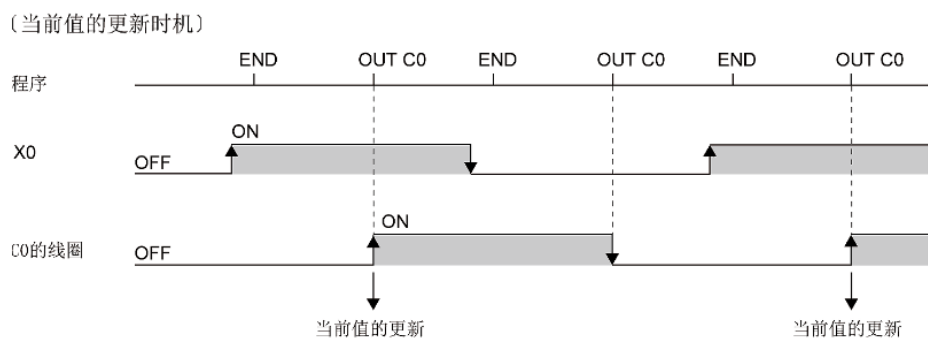
1 点使用 2 字。可计数范围为 0~2147483647。

4) 计数处理

计数器指令之前的运算结果由 OFF→ON 变化时, 计数器线圈会 OFF→ON 变化, (d) 中指定的计数器的当前值(计数值)+1, 如果计数到(当前值 $\geq$ 目标值), 常开触点将导通。

执行计数器的指令时进行计数器线圈 CC 的 ON/OFF、当前值 CV 的更新(计数值+1)以及触点 CS 的 ON/OFF 处理。当前值 CV 的更新(计数值+1)是在计数器的线圈 CC 输入的上升沿(OFF→ON)时进行。线圈 CC 输入为 OFF、ON→ON 以及 ON→OFF 时不进行当前值的更新。超长计数器处理相同。





5) 计数器目标值

扫描 OUT 计数器指令执行时, C/LC 目标值被赋值。当开始计数后, 至计数器触点导通前, 目标值可变。在执行 OUT 指令对计数器的目标值进行更新或计数器更新时, 如果当前值大于等于更新后的目标值, 计数器触点导通且变更计数器的当前值=目标值。

6) 计数器的变址

计数器中使用的软元件, 可以用变址寄存器 (Z/LZ) 进行修饰。

例: OUT C1Z0 D0

LD LC0Z0

计数器通过 (Z/LZ) 变址后, 变址前指向的计数器将保持当前计数。此计数器的线圈保持导通。切换回此计数器时根据线圈状态执行计数。

7) 计数器锁存

计数器设置为锁存软元件时, 计数器的当前值、输出触点动作及 RST 内部状态将被锁定。使用 ZRST 指令, 可以对计数器的 RST 内部状态进行复位。

8) PLC 状态与计数器

运行 stop, run 状态变化时, 计数器情况如下所示。

PLC 状态	计数器结构	计数器		超长计数器	
		非锁存	锁存	非锁存	锁存
run→stop	线圈	OFF	保持	OFF	保持
	当前值	0	保持	0	保持
	目标值	0	保持	0	保持
	触点	OFF	保持	OFF	保持
	复位状态	OFF	保持	OFF	保持

stop→run	线圈	OFF	保持	OFF	保持
	当前值	0	保持	0	保持
	目标值	0	保持	0	保持
	触点	OFF	保持	OFF	保持
	复位状态	OFF	保持	OFF	保持

9) 注意事项

- 目标值 (CP/LCP) 为 0 的情况下，其处理与 1 时相同。
 - 目标值 (CP/LCP) 为负值时，作为 0 处理。
- RST 指令执行中如果被跳过，定时器的复位状态将一直被保持直到 ZRST 清除为止。其间不能正常计数。
- 同一周期内多次对同一计数器执行 OUT 指令时，只要计数器线圈 OFF→ON，计数器当前值均会更新（计数值+1）。

注：其他软元件请参照 R 系列应用篇之软元件

2.4.2 高速计数功能

详见 4.24

3 指令一览表

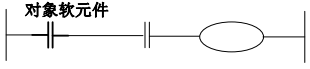
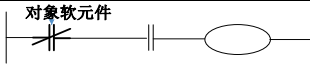
3.1 概要

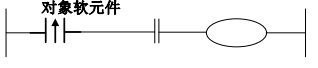
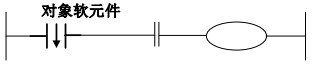
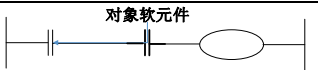
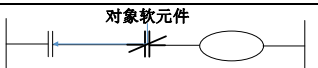
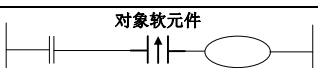
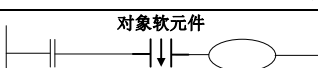
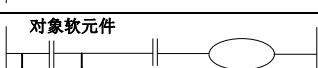
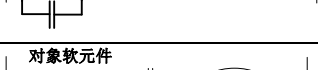
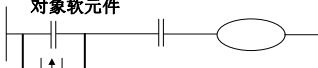
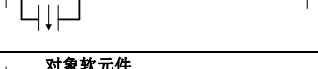
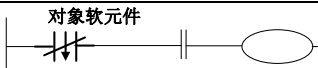
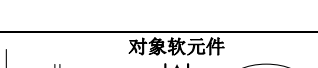
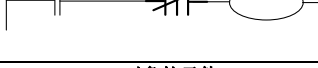
R 系列小型 PLC 的指令分为基本命令和应用命令，本章节将对此两种命令进行介绍。

3.2 基本指令

R 系列小型 PLC 的基本命令如下所示

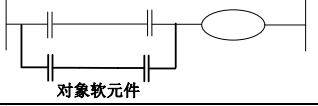
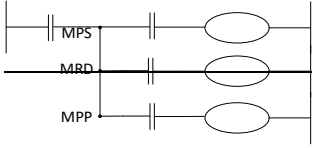
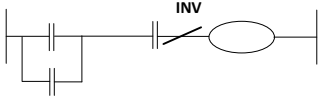
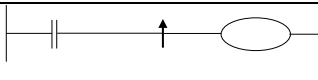
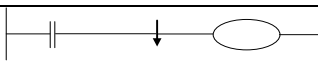
触点指令

指令	名称	示例	描述
LD	取		a 触点的逻辑运算开始
LDI	取反		b 触点的逻辑运算开始

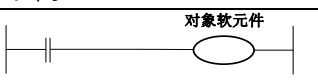
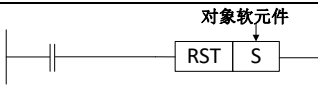
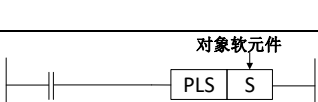
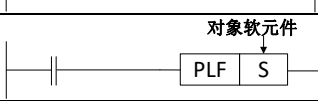
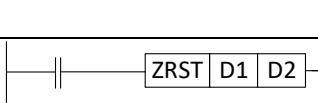
LDP	取脉冲上升沿		检测上升沿的运算开始
LDF	取脉冲下降沿		检测下降沿的运算开始
AND	与		串联 a 触点
ANI	与反转		串联 b 触点
ANDP	与脉冲上升沿		检测上升沿的串联连接
ANDF	与脉冲下降沿		检测下降沿的串联连接
OR	或		并联 a 触点
ORI	或反转		并联 b 触点
ORP	或脉冲上升沿		检测上升沿的并联连接
ORF	或脉冲下降沿		检测下降沿的并联连接
LDPI	脉冲上升沿取反		检测非脉冲上升沿
LDFI	脉冲下降沿取反		检测非脉冲下降沿
ANDPI	与脉冲上升沿取反		检测非与脉冲上升沿
ANDFI	与脉冲下降沿取反		检测非与脉冲下降沿
ORPI	或脉冲上升沿取反		检测非或脉冲上升沿
ORFI	或脉冲下降沿取反		检测非或脉冲下降沿

结合指令

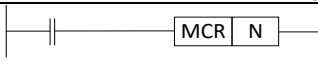
指令	名称	示例	描述
ANB	回路块与		回路块的串联连接

ORB	回路块或		回路块的并联连接
MPS	存储器进栈		压入堆栈
MRD	存储器读栈		读取堆栈
MPP	存储器出栈		弹出堆栈
INV	反转		运算结果的反转
MEP	M·E·P		上升沿时导通
MEF	M·E·F		下降沿时导通

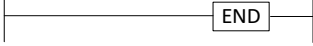
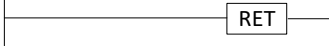
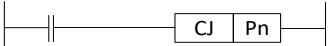
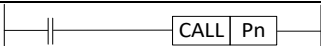
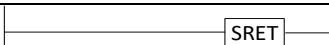
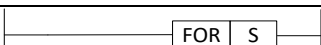
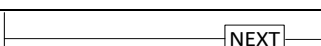
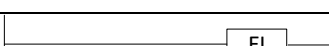
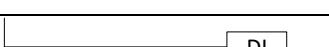
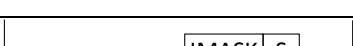
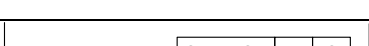
输出指令

指令	名称	示例	描述
OUT	输出		线圈驱动
SET	置位		动作保持
RST	复位		解除保持的动作，清除当前值及寄存器
PLS	脉冲		上升沿微分输出
PLF	下降沿脉冲		下降沿微分输出
ALT	交替输出		交替输出
ZRST	批量复位		批量复位
ZSET	批量置位		批量置位

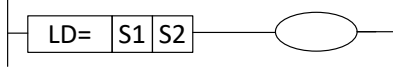
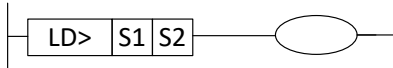
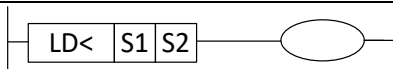
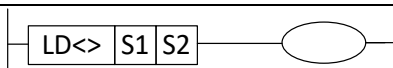
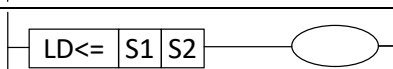
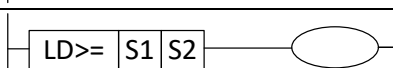
主控指令

指令	名称	示例	描述
MC	主控		连接到公共触点
MCR	主控复位		解除连接到公共触点

程序流程

指令	名称	示例	描述
NOP	空操作		无处理
END	结束		程序结束以及输入输出处理和返回 0 步
STL	步进梯形图		步进梯形图的开始
RETSTL	返回		步进梯形图的结束
CJ	跳转		条件跳转
CALL	调用		子程序调用
SRET	返回		子程序返回
FOR	循环开始		循环范围的开始
NEXT	循环结束		循环范围的结束
EI	中断使能		允许中断
DI	中断失能		禁止中断
IMASK	中断程序屏蔽		屏蔽中断程序
SIMASK	中断指针控制		使能/失能中断指针

触点比较

编号	指令	示例	描述
1	LD=		触点比较 LD =
2	LD>		触点比较 LD >
3	LD<		触点比较 LD <
4	LD<>		触点比较 LD ≠
5	LD<=		触点比较 LD ≤
6	LD>=		触点比较 LD ≥

7	AND=		触点比较 AND =
8	AND>		触点比较 AND >
9	AND<		触点比较 AND <
10	AND<>		触点比较 AND ≠
11	AND<=		触点比较 AND <=
12	AND>=		触点比较 AND >=
13	OR=		触点比较 OR =
14	OR>		触点比较 OR >
15	OR<		触点比较 OR <
16	OR<>		触点比较 OR ≠
17	OR<=		触点比较 OR <=
18	OR>=		触点比较 OR >=

3.3 应用指令

比较输出

编号	指令	示例	描述
1	CMP		比较
2	ZCP		区间比较
3	BKCMPE、 >、<、 <>、		数据块的比较

	<=、>=		
--	-------	--	--

二进制逻辑运算

编号	指令	示例	描述
1	ADD		BIN 加法运算
2	SUB		BIN 减法运算
3	MUL		BIN 乘法运算
4	DIV		BIN 除法运算
5	INC		BIN 加一
6	DEC		BIN 减一
7	WAND		逻辑与
8	WOR		逻辑或
9	WXOR		逻辑异或
10	NEG		补码

数据块运算

编号	指令	示例	描述
1	BK+		数据块的加法
2	BK-		数据块的减法

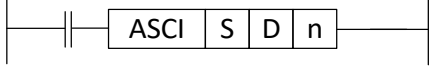
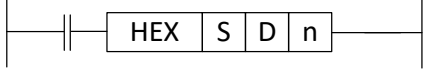
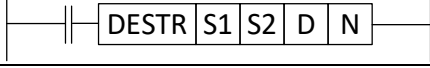
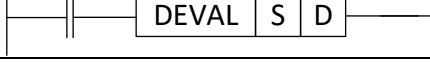
2 进制浮点数运算

编号	指令	示例	描述
1	DEADD		2 进制浮点数加法
2	DESUB		2 进制浮点数减法
3	DEMUL		2 进制浮点数乘法
4	DEDIV		2 进制浮点数除法
5	DSIN		2 进制浮点数正弦
6	DCOS		2 进制浮点数余弦

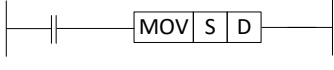
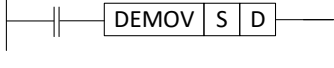
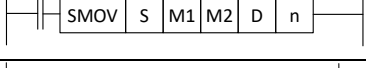
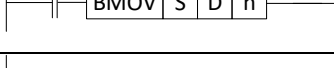
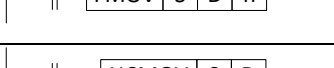
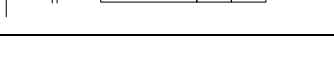
7	DTAN		2 进制浮点数正切
8	DASIN		32 进制浮点数反正弦
9	DACOS		2 进制浮点数反余弦
10	DATAN		2 进制浮点数反正切
11	DRAD		2 进制浮点数角度转弧度
12	DDEG		2 进制浮点数弧度转角度
13	DEXP		2 进制浮点数指数数
14	DLOGE		2 进制浮点数自然对数
15	DLOG10		2 进制浮点数常数对数
16	DESQR		2 进制浮点数开方

数据转换

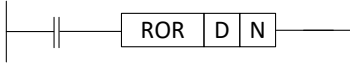
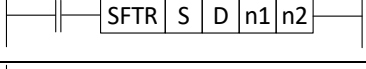
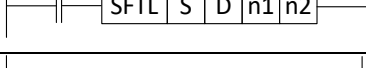
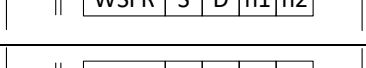
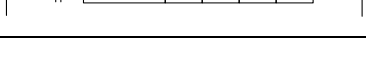
编号	指令	示例	描述
1	INT		2 进制浮点数→BIN 整数的转换
2	EBCD		2 进制浮点数→10 进制浮点数的转换
3	EBIN		10 进制浮点数→2 进制浮点数的转换
4	FLT		BIN 整数→2 进制浮点数转换
5	DEBCD		32 位 2 进制浮点数→10 进制浮点数的转换
6	DEBIN		32 位 10 进制浮点数→2 进制浮点数的转换
7	DABIN		32 位 10 进制 ASCII→BIN 的转换
8	BINDA		BIN→10 进制 ASCII 的转换
9	WTOB		字节单位数据分离
10	BTOW		字节单位的数据结合

11	ASCII		HEX→ASCII 的转换
12	HEX		ASCII→HEX 的转换
13	DESTR		32 位 2 进制浮点数→字符串的转换
14	DEVAL		32 位字符串→2 进制浮点数的转换

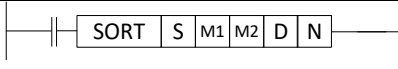
数据传送

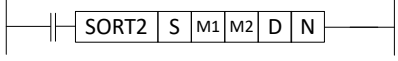
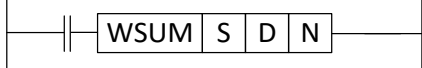
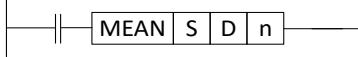
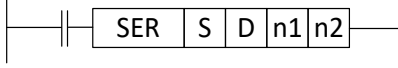
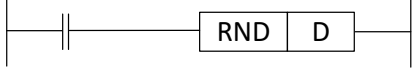
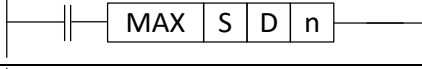
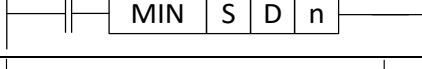
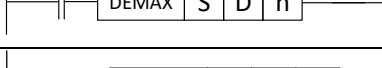
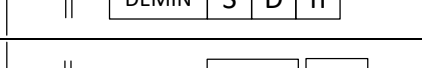
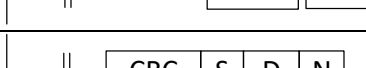
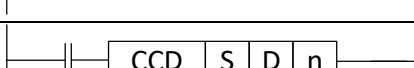
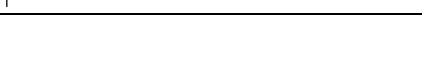
编号	指令	示例	描述
1	MOV		传送
2	DEMOV		2 进制浮点数数据传送
3	SMOV		位移动
4	BMOV		成批传送
5	FMOV		多点传送
6	HCMOV		SD 寄存器专用高速当前值传送指令

循环、位移

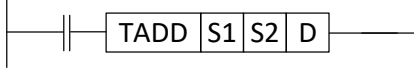
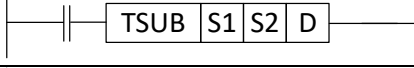
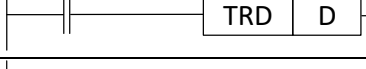
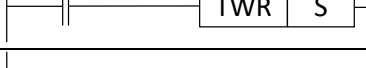
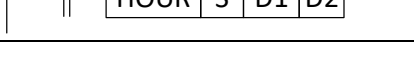
编号	指令	示例	描述
1	ROR		循环右移
2	ROL		循环左移
3	SFTR		位右移
4	SFTL		位左移
5	WSFR		字右移
6	WSFL		字左移

数据处理

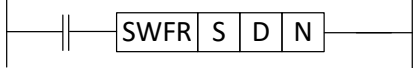
编号	指令	示例	描述
1	SORT		16 位数据排序

2	SORT2		数据块的加法
3	WSUM		算出数据合计值
4	MEAN		平均值
5	SER		数据检索
6	RND		产生随机数
7	MAX		整型数据最大值搜索
8	MIN		整型数据最小值搜索
9	DEMAX		单精度实数最大值搜索
10	DEMIN		单精度实数最小值搜索
11	SWAP		高低字节互换
12	CRC		CRC 运算
13	CCD		校验码

时钟指令

编号	指令	示例	描述
1	TADD		时钟数据加法运算
2	TSUB		时钟数据减法运算
3	TRD		读出时钟数据
4	TWR		写入时钟数据
5	HOUR		计时表

数据表操作指令

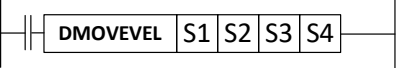
编号	指令	示例	描述
1	SFWR		至数据表的数据写入

2	SFRD		从数据表的先入数据读取
3	POP		从数据表的后入数据读取
4	FINS		数据表的数据插入
5	FDEL		数据表的数据删除
6	LIMIT		上下限位控制
7	BAND		死区控制
8	ZONE		区域控制
9	SCL		SLC 定坐标
10	SCL2		SLC2 定坐标 (X/Y 坐标组数据)

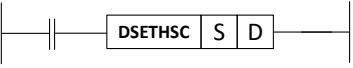
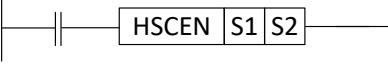
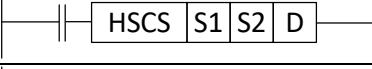
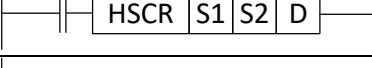
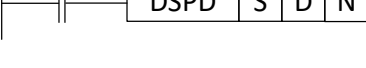
3.4 高速处理及定位

运动控制

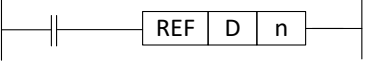
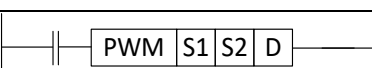
编号	指令	示例	描述
1	DDSZR		原点回归模式
2	DDVIT		中断定位模式
3	DDRVI		相对定位模式
4	DDRVA		绝对定位模式
5	DPLSV		可变速模式
6	DMOVEHOME		原点回归模式
7	DMOVEABS		中断定位模式
8	DMOVEINC		相对定位模式
9	DMOVEINT		绝对定位模式
10	DMOVEJOG		点动模式

11	DMOVEVEL		可变速模式
----	----------	--	-------

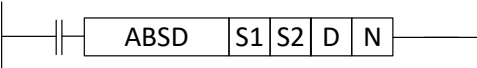
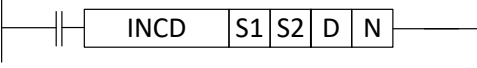
高速计数和脉冲测量

编号	指令	示例	描述
1	DSETHSC		32 位高速预置值设置
2	HSCEN		高数计数器功能开启 关闭指令
3	HSCS		高速比较置位
4	HSCR		高速比较复位
5	DSPD		32 位脉冲测量

高速处理

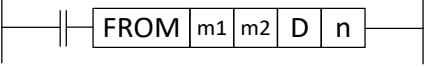
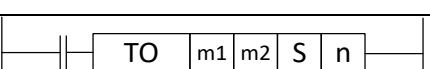
编号	指令	示例	描述
1	REF		输入输出刷新
2	PWM		脉宽调制

3.21 凸轮顺控

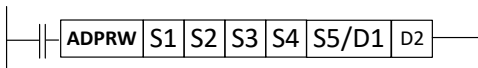
编号	指令	示例	描述
1	ABSD		凸轮控制绝对方式
2	INCD		从数据表的先入数据 读取

3.5 通讯指令

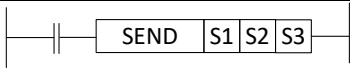
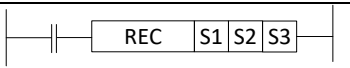
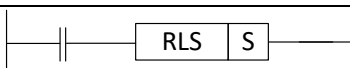
模块访问指令

编号	指令	示例	描述
1	FROM		BFM 的读出
2	TO		BFM 的写入

3.20 ModbusRtu

编号	指令	示例	描述
1	APDRW		ModbusRtu 通信指令

3.21 自由通信

编号	指令	示例	描述
1	SEND		写本地数据至目标设备
2	REC		写目标数据到本地
	RLS		释放指定串口资源

4.软元件的作用于功能

4.1 R 系列软元件一览

R 系列的软元件一览如下所示

分类	类型	软元件名称	符号	数据格式
用户软元件	位	输入	X	8 进制数
	位	输出	Y	8 进制数
	位	内部继电器	M	10 进制数
	位	锁存继电器	L	10 进制数
	位	链接继电器	B	16 进制数
	位	报警器	F	10 进制数
	位	链接特殊继电器	SB	16 进制数
	位	步进继电器	S	10 进制数
	位/字	定时器	T(TC、TS)	10 进制数
	位/字	累计定时器	ST(STC、STS)	10 进制数
	位/字	计数器	C(CC、CS)	10 进制数
	位/双字	超长计数器	LC(LCC、LCS)	10 进制数
	字	数据寄存器	D	10 进制数
	字	链接寄存器	W	16 进制数
	字	链接特殊寄存器	SW	16 进制数
系统软元件	位	特殊继电器	SM	10 进制数
	字	特殊寄存器	SD	10 进制数
变址寄存器	字	变址寄存器	Z	10 进制数
	双字	超长变址寄存器	LZ	10 进制数
文件寄存器	字	文件寄存器	R	10 进制数
嵌套	—	嵌套	N	10 进制数
指针	—	指针	P	10 进制数
	—	中断指针	I	10 进制数
常数	—	10 进制常数	K	10 进制数

	—	16 进制常数	H	16 进制数
	—	实数常数	E	—
	—	字符串常数	—	—

注意：关于定时器/累计定时器/计数器/长计数器的符号，软元件指定时按照指令确定了型号的情况下，应通过 T/ST/C/LC 指定。未确定型号时，应根据触点/线圈/当前值中指定的型号指定符号。但是，当前值也可通过 T/ST/C/LC 指定。（详见 1.2 用户软元件章节的定时器和计数器部分）

4.2 R 系列软元件分配一览

符号	默认点数	范围	范围配置	锁存范围	锁存范围配置
X	最多 64	0-377 (X, Y 共 256 点)	—	—	—
Y	最多 64		—	—	—
M	10239	0-10239	0-10239	500-10239	可配置
L	0	0	0-10239	—	—
B	0	0	0-10239	0	可配置
F	0	0	0-10239	0	可配置
SB	0	0	0-10239	—	—
S	4096	0-4095	—	500-4095	—
T	512	0-511	0-1024	0	可配置
ST	512	0-511	0-1024	0	可配置
C	256	0-255	0-1024	100-255	可配置
LC	64	0-63	0-1024	20-63	可配置
D	8000	0-7999	0-10240	256-8355	可配置
W	0	0	0-32768	—	—
SW	0	0	0-32768	—	—
SM	12288	0-12287	—	—	—
SD	12288	0-12287	—	—	高速计数器锁存
Z	12	0-11	0-11	—	—
LZ	2	0-1	0-11	—	—
R	32768	0-32767	0-32767	—	全锁存
ER	—	—	—	—	—
P	1024	0-1023	—	—	—
N	15	0-14	—	—	—

注意：

1. R 系列中 X 与 Y 的实际点数加上扩展模块中的 X 和 Y 的点数，最多共可使用 256 点。其中 PLC 的 X 和 Y 点最大数量为 64+64。
2. 软元件分配的查看与修改需要打开 Hcp Works2 中设置菜单中的软元件分配功能，且只能在非监控状态下打开。
3. 软元件分配界面的参数修改后，用户需要重新将其写入 PLC 重启后才会生效。
4. 软元件配置界面中的 M, B, SB, F, S, L, T, ST, C, LC, D 这些软元件的范围总和最大

- 值和锁存范围总和最大值是 12.0K，分配软元件点数和锁存范围时不能超过此值
5. R, W, SW 软元件最大使用范围总和是 32.0K，分配时三者范围总和不能超过此值
6. Z 和 LZ 软元件二者共可使用 16 点，分配时不能超过此范围
7. 软元件配置界面如下，仅当用户勾选修改软元件分配选项框时，才能修改软元件分配

软元件配置

高速软元件

M

7680

锁存点数

3840

起始

512

结束

4351

B

256

锁存点数

0

起始

0

结束

0

SB

512

F

128

锁存点数

0

起始

0

结束

0

S

4096

锁存点数

2048

起始

512

结束

2559

L

7680

T

512

锁存点数

0

起始

0

结束

0

ST

32

锁存点数

32

起始

0

结束

31

C

256

锁存点数

128

起始

0

结束

127

LC

64

锁存点数

32

起始

0

结束

31

D

8000

锁存点数

4000

起始

256

结束

4255

锁存已使用

5.1 K

锁存最大值

12.0 K

已使用

11.0 K

最大值

12.0 K

输入输出软元件

X

256

Y

256

标准软元件

R

2048

W

2048

SW

2048

已使用

6.0 K

最大值

32.0 K

变址软元件

Z

12

LZ

2

已使用

16

最大值

16

系统软元件

SM

12288

SD

12288

已使用

12.8 K

最大值

12.8 K

☐ 兼容A系列PLC

A1

☒ 修改软元件分配

默认值

确定

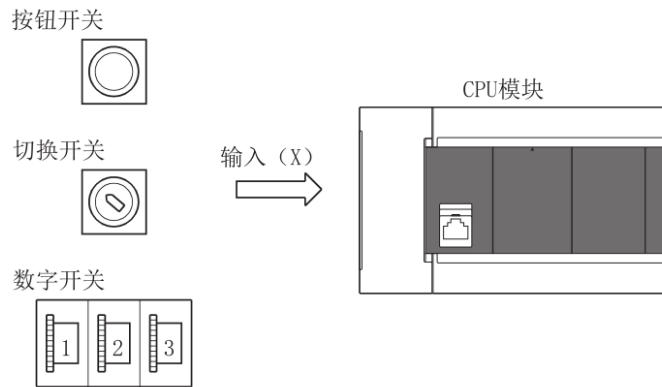
取消

4.3 用户软元件

4.3.1 输入（X）

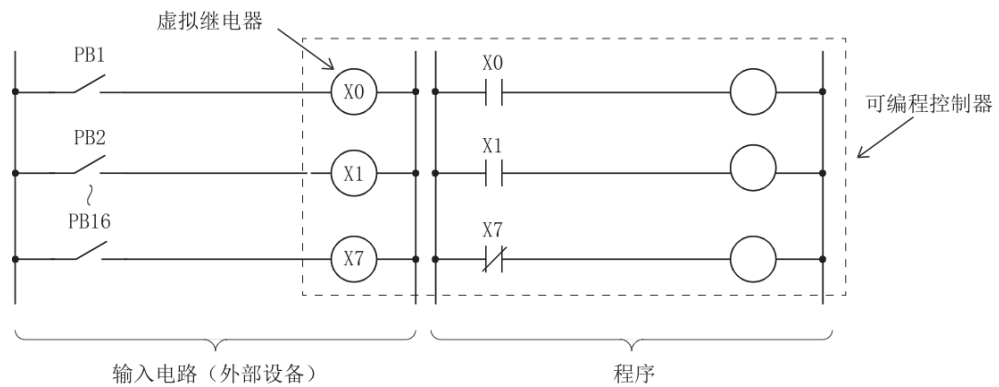
输入是用于通过按钮/切换开关/限位开关/数字开关等外部设备，向 CPU 模块发出指令及数据的软元件。

35



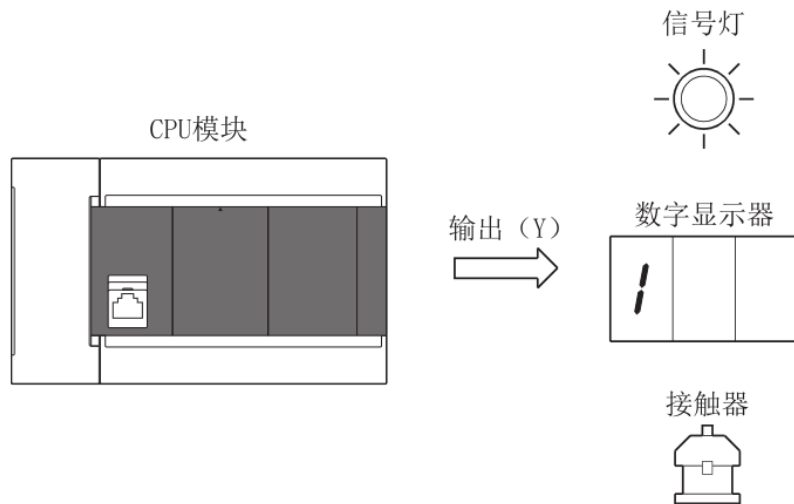
输入的相关信息

输入可认为是在 CPU 模块内对输入 1 点内置有 1 个虚拟继电器 XN。程序中，使用该继电器 XN 的常开触点/常闭触点。



4.3.2 输出 (Y)

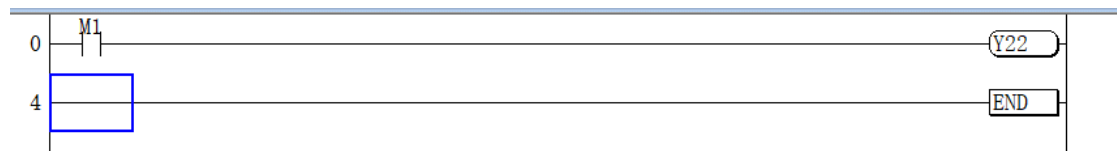
输出是将程序的控制结果输出至外部的信号灯/数字显示器/电磁开关器（接触器）/螺线管等的软元件。



4.3.3 内部继电器 (M)

是在 CPU 模块内部作为辅助继电器使用的软元件。如果进行以下操作内部继电器将全部为 OFF。

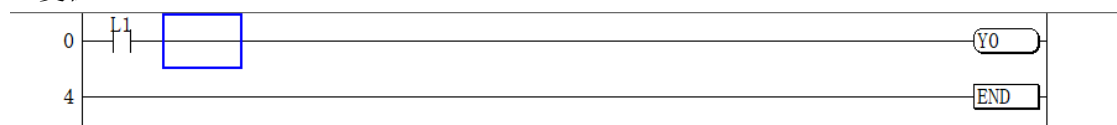
- CPU 模块的电源 OFF→ON
- 复位
- 锁存清除



4.3.4 锁存继电器 (L)

是 CPU 模块内部使用的可锁存（停电保持）的辅助继电器。即使进行以下操作，运算结果（ON/OFF 信息）也将被锁存。

- CPU 模块的电源 OFF→ON
- 复位

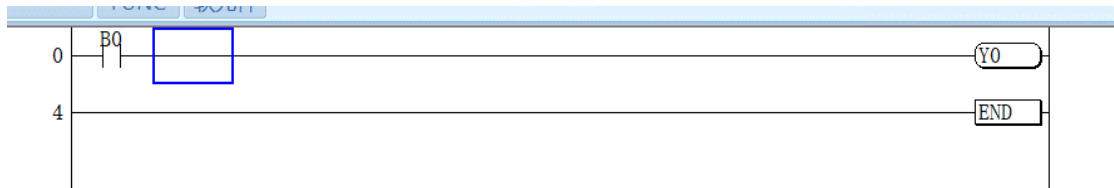


4.3.5 链接继电器（B）

是在网络模块与 CPU 模块之间作为刷新位数据时的 CPU 侧软元件使用的软元件

使用了链接继电器（B）的网络模块的刷新

在 CPU 模块内的链接继电器（B）与网络模块的链接继电器（LB）之间相互收发数据。刷新范围在网络模块的参数中设置。未用于刷新的位置可用于其他用途



4.3.6 报警器（F）

是在由用户创建的用于检测设备异常/故障的程序中使用的内部继电器。

1) 将报警器置为 ON 的方法

使用 SET F 指令。仅在输入条件的上升沿时（OFF→ON）报警器置为 ON，即输入条件变为 OFF，报警器依旧保持 ON 状态。

- 通过 OUT F 指令也可以将报警器置为 ON，但由于每个扫描都要处理，因此与使用 SET F 指令相比，扫描时间将延长。
- 通过 SET F 或 OUT F 以外的指令（例如 MOV 指令）置为 ON 时，与内部继电器动作相同。因此，不会执行

SM49 置 ON 及向 SD520, SD521、SD522（报警器检测编号表）～SD537 存储报警器编号的操作。

2) 报警器 ON 时的处理内容

只有当报警有效 SM49 为 ON 时，将报警器置为 ON 时，SM48（报警器检测）将为 ON，SD520 和 SD49 将储存最先置 ON 的报警器，SD521 将会储存报警器置 ON 的数量。SD49 会储存最小报警器编号。（SM49 为 OFF 时，即使有 F 置 ON，SM48 和 SD49 将不会动作）

SD522（报警器编号）～SD537（报警器检测编号表）中将存储变为 ON 的报警器编号。

注意：sm48、sm49、sd49 是以周期为单位更新的，在 end 的时候处理。

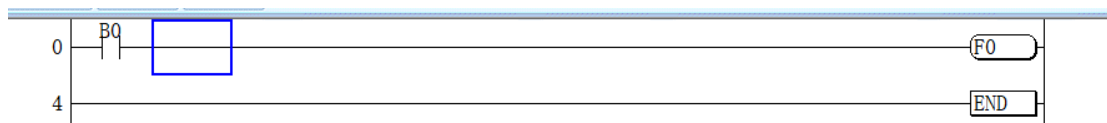
3) 将报警器置为 OFF 的方法

通过以下指令执行。

指令	使用用途
RST F 指令	用于将通过 SET F 指令置为 ON 的报警器编号置为 OFF 时。
ZRST 指令	用于将指定范围内的报警器编号批量置为 OFF 时

也可通过 OUT F 指令置为 OFF，但即使通过 OUT F 指令将报警器编号置为 OFF，

也不会执行以下所示的“报警器 OFF 时的处理内容”。通过 OUT F 指令将报警器编号置为 OFF 时，需要执行以上所示的 RST F /ZRST 指令

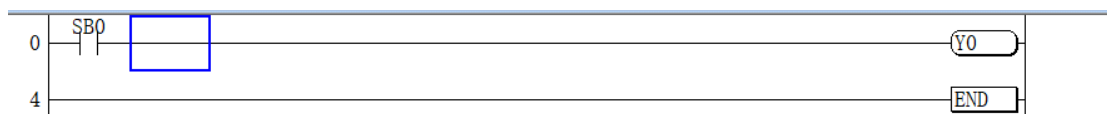


4) 报警器置为 OFF 的处理内容

当通 RST/ZRST 关闭了报警器 F 时，SD520-SD537 中的值将会立即清除。SM48 立即置 OFF，SD49 中的值立即清除，变为 0。（当 SM49 置 OFF 时，即使有 F 置 ON，SM48 和 SD49 也会变成 0）。

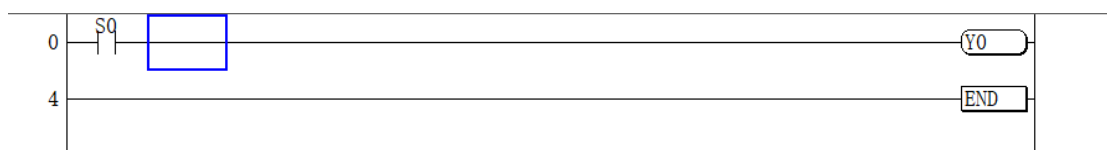
4.3.7 链接特殊继电器（SB）

网络模块的通信状态及异常检测状态将被输出到网络内的链接特殊继电器中。链接特殊继电器（SB）是以作为网络内的链接特殊继电器的刷新目标使用为目的的软元件。未用于刷新的位置可用于其他用途。



4.3.8 步进继电器（S）

在步进梯形图指令中使用的软元件。未用于步进梯形图的位置可用于与辅助继电器相同的用途。



4.3.9 定时器（T/ST）

1) 定时器的构成

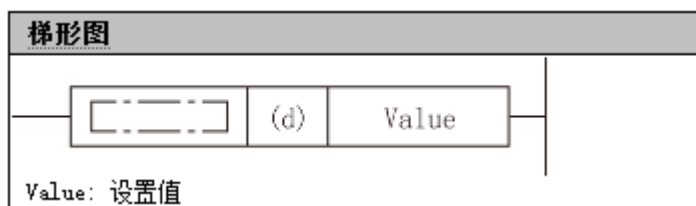
每个定时器 T/累计定时器 ST 都有被不断更新的当前值（①）、作为时限的目标值（②）、作为线圈使用的定时器线圈（③）、和作为触点使用的定时器触点（④）。各个属性对应符号如下：

定时器类型	定时器的属性			
	当前值①	目标值②	线圈③	触点④
定时器	TV	TP	TC	TS
累计定时器	STV	STP	STC	STS

2) 定时器的输出指令

输出指令之前的运算结果为 ON 时，(d)中指定的定时器/累计定时器的线圈将 ON 并进行计测，直至设置值为止。如果时限到，常开触点将导通，常闭触点将变为非导通。

■指令格式



OUT/OUTH/OUTHS (d) Value

■操作数

操作数	内容	范围	数据类型
(d)	定时器编号	—	T/ST
Value	定时器的设置值	0 ~ 32767	无符号 BIN16 位

■可使用软元件

操作数	位	字		双字		常数	
	X、Y、M、L、 SM、F、B、SB、S	T、ST、C、D、 W、SD、SW、R	Z	LC	LZ	K、H	E
(d)	—	仅支持 T、ST ^{*1}	—	—	—	—	—
Value	—	○ ^{*2}	—	—	—	○ ^{*3}	—

*1 T/ST 支持 Z、LZ 变址修饰。

*2 不能使用 T、ST、C。

*3 只能使用 10 进制常数(K)。

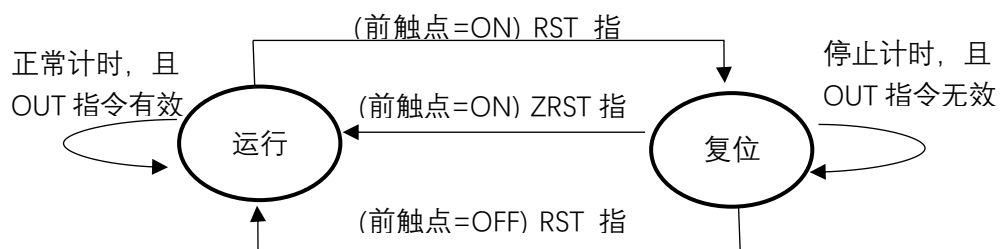
高中低速定时器输出指令

OUT 指令作为 100ms 定时器，OUTH 指令作为 10ms 定时器，OUTHS 指令作为 1ms 定时器。

例如，相同的 T0，指定 OUT T0 时为低速定时器（100ms），指定 OUTH T0 时为定时器（10ms），指定 OUTHS T0 时为高速定时器（1ms）。累计定时器也同样。

3) 定时器的复位指令

定时器 T/ST 以通过 RST 和 ZRST 指令进行复位操作。执行前触点=ON 的 RST 指令后，定时器 T/ST 进入复位状态。处于复位状态的定时器，各个属性值保持不变，计时停止，OUT 指令无效。执行前触点=OFF 的 RST 指令，以及执行前触点=ON 的 ZRST 指令，可以解除定时器的复位状态。



(4) RST/ZRST 复位指令后，各个定时器 T/ST 的属性值如下：

	定时器的类型	定时器线圈	定时器当前值	定时器目标值	定时器触点	定时器复位状态
RST	定时器	—	0	保持	OFF	锁定
	累计定时器	—	0	保持	OFF	锁定
ZRST	定时器	OFF	0	保持	OFF	释放
	累计定时器	OFF	0	保持	OFF	释放

—：定时器线圈(TC/STC)状态与定时器前触点运算结果一致。

(5) RST/ZRST 指令执行中，被复位的定时器/累计定时器保持被复位状态。



如上图所示，当 M0 导通后“RST ST0”将持续执行。X0 后的累计定时器 ST0 将保持复位状态且不能计时。当 RST T/ST 指令停止运行，复位的定时器/累计定时器根据前置运算结果状态执行对应动作。



如上图所示，当 M0 不导通后“RST ST0”停止执行。X0 后的累计定时器 ST0 将从 0 开始进行计时。

定时器类型

❖ 定时器(T)

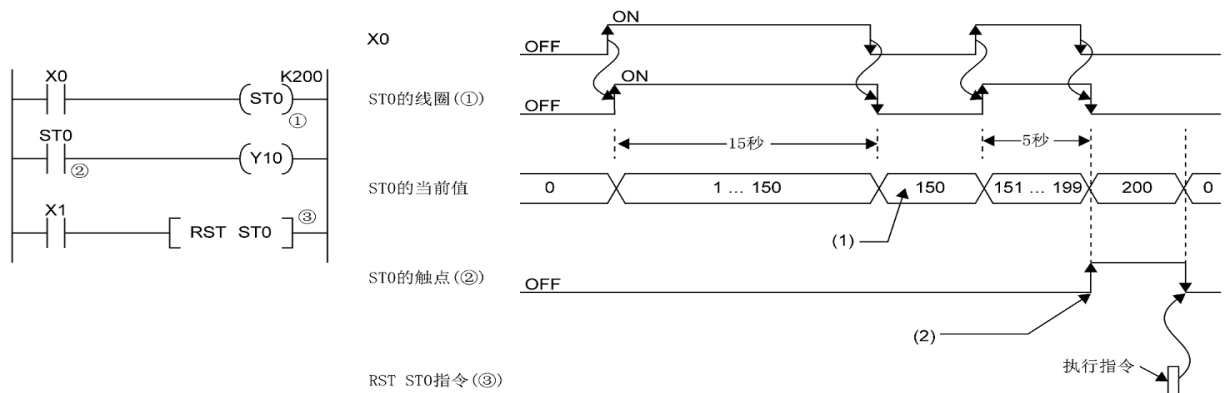
定时器的线圈 TC 变为 ON 时开始计测。定时器的当前值 TV 与目标值 TP 一致时，定时器的触点 TS 将变为 ON。将定时器的线圈 TC 置为 OFF 时，当前值 TV 将变为 0，定时器的触点也将变为 OFF。



❖ 累计定时器(ST)

累计定时器的线圈为 ON 时开始计测，当前值 STV 与目标值 STP 一致(时限到)时，累计定时器的触点 STS 将变为 ON。

即使累计定时器的线圈变为 OFF，当前值 STV 及触点 STS 的 ON 状态也将保持。再次线圈为 ON 时，从保持的当前值 STV 开始重启计测。累计定时器的当前值 STV 的清除及触点的 OFF 需要通过 RST ST 指令进行。



(1) 即使线圈(①)为 OFF 也保持当前值(TV)。

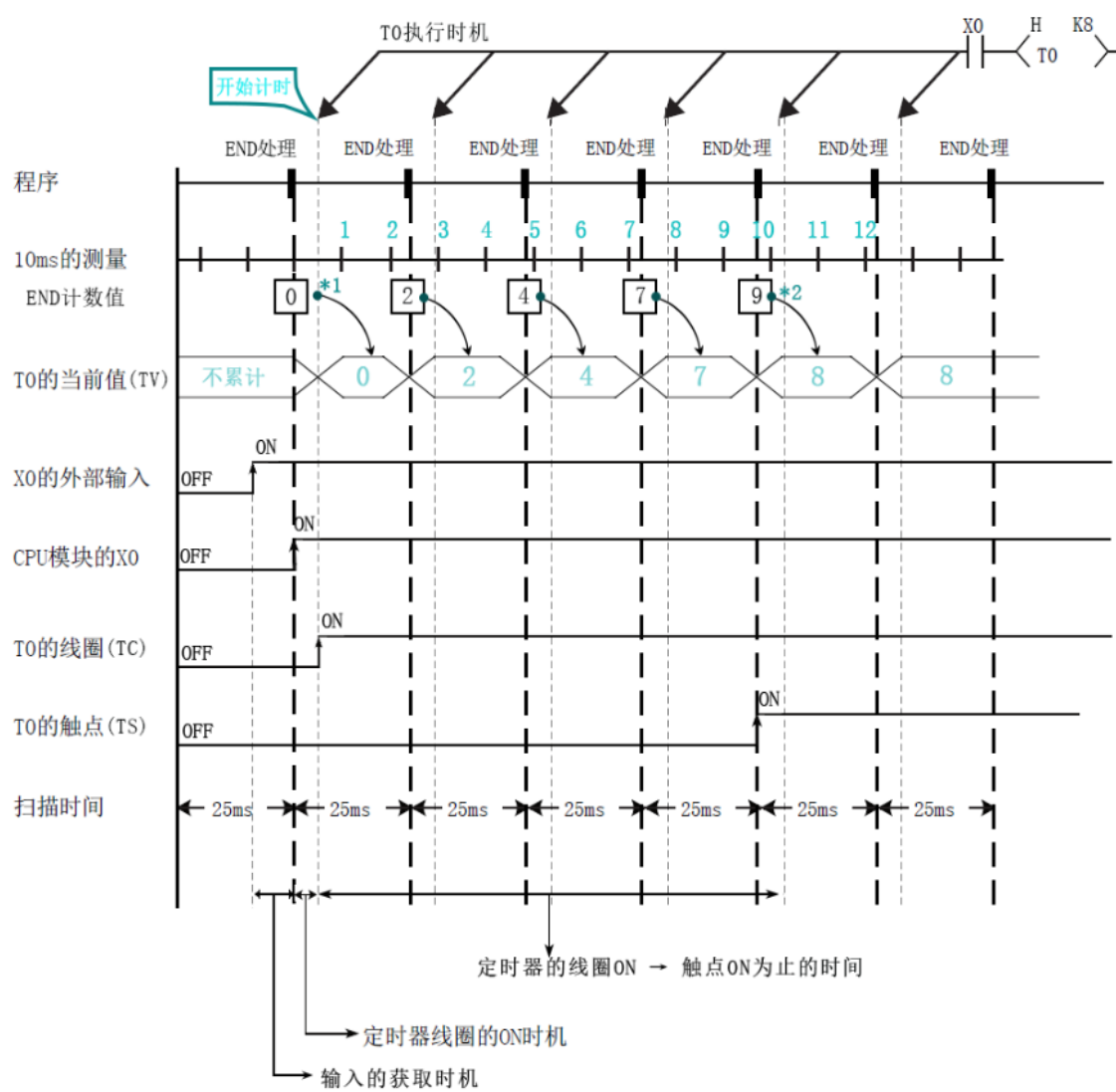
(2) 即使线圈(①)为 OFF 触点也继续保持 ON。

1) 定时器的时序和精度

本节的时序图描述了定时器 T 的正常计时过程（累计定时器 TS 同样），设置内容为：定时器时限设置 = 10ms、T0 的目标值 = 8(10ms×8 = 80ms)、扫描时间 = 25ms、输出指令：OUTH (1ms 单位)。

请注意，外部输入 X 驱动的定时器 T 的精度，受到扫描周期时间和定时器命令精

度的影响。



*1 定时器线圈 (TS) 首次触发时, 定时器当前值 (TV) 将更新 (TV=0)

*2 END处理时, TV=9, TP=8, TV>TP。此时定时器触点导通(TS=ON), 并变更当前值 (TV=TP=8)。

2) 定时器的停止

OUT 指令前置运算结果由 ON→OFF 变化时，其各个属性值的变化如下所示。

定时器的类型	线圈 TC	当前值 TV	触点 TS		定时器复位状态
			未到时限	到达时限	
定时器	OFF	0	非导通	非导通	保持
累计定时器	OFF	保持当前值	非导通	导通	保持

注意：处于复位状态中的累计定时器，不论是否执行 OUT 指令的前触点 OFF，直到 RST 指令的前触点 OFF 执行为止，复位状态都不会被解除。

3) PLC 运行状态与定时器

当 PLC 的 STOP/RUN 状态变化时，定时器的各个属性值会被清零（该定时器为非锁存属性）或保持不变（该定时器为锁存属性）。表中的复位属性，表示该定时器处于 RST 指令复位中。

PLC 状态	定时器属性	定时器 T		累计定时器 ST	
		非锁存	锁存	非锁存	锁存
run→stop	线圈	OFF	保持	OFF	保持
	当前值	0	保持	0	保持
	目标值	0	保持	0	保持
	触点	OFF	保持	OFF	保持
	复位	OFF	保持	OFF	保持
stop→run	线圈	OFF	保持	OFF	保持
	当前值	0	保持	0	保持
	目标值	0	保持	0	保持
	触点	OFF	保持	OFF	保持
	复位	OFF	保持	OFF	保持

4) 定时器目标值

- OUT 定时器指令执行时，目标值 TP 被赋值。
- 计时开始至时限为止，目标值 TP 可变。
- 执行 OUT 指令更新目标值 TP 时，更新后目标值 $TP \leq$ 如果当前值 TV，定时器触点导通（到达时限）。

注：累计定时器 STP 与 TP 同样。

5) 定时器的变址修饰

定时器可以用变址寄存器 (Z/LZ) 进行修饰。例如：

OUT T1Z0 D0

LD T0LZ0

注意：定时器通过 (Z/LZ) 变址后，变址前指向的定时器将继续计时。当此定时器计时达到目标值时，此定时器的触点导通。

注意事项

- 定时器在计时状态中（线圈 TC 为 ON 状态中的情况下），通过 CJ 指令等跳过定时器的 OUT T 指令、OUT ST 指令，跳过的计时中的定时器会继续计时。
- 目标值(TP/STP)为 0 时，执行 OUT T/ST 指令时触点将变为导通。
- 目标值(TP/STP)为负值时，作为 0 处理。
- 定时器到达时限后，即使将目标值更改，定时器也将保持为触点导通状态。
- 通过监控软元件方式变更定时器/累计定时器当前值，定时器将不进行计数。
- RST T/ST 指令执行中如果被跳过，定时器的复位状态将一直被保持直到 ZRST 清除为止。其间不能正常计时。

4.3.10 计数器 (C/LC)

1) 计数器的构成

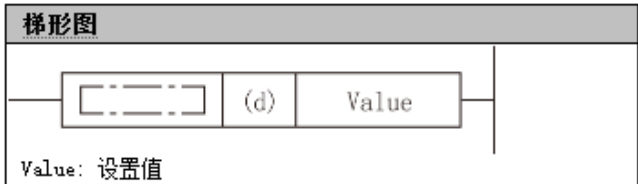
计数器和超长计数器都有被不断更新的当前值（①）、作为最大计数的目标值（②）、作为线圈使用的定时器线圈（③）、和作为触点使用的定时器触点（④）。各个属性对应符号如下：

计数器类型	计数器的属性			
	当前值①	目标值②	线圈③	触点④
计数器	CV	CP	CC	CS
超长计数器	LCV	LCP	LCC	LCS

2) 计数器的输出指令

计数器指令之前的运算结果由 OFF→ON 变化时，线圈将由 OFF→ON 变化，将 (d) 中指定的计数器的当前值+1，如果计数到(Value)，触点将导通。

■ 指令格式



OUT (d) Value

■ 操作数

操作数	内容	范围	数据类型	掉电锁存
(d)	计数器编号	—	计数器/超长计数器	支持
Value	计数器的目标值	C:0 ~ 32767 LC:0 ~ 2147483647	无符号 BIN16 位	—

■ 可使用软元件

操作数	位	字		双字		常数	
	X、Y、M、L、 SM、F、B、SB、S	T、ST、C、D、 W、SD、SW、R	Z	LC	LZ	K、H	E
(d)	—	仅支持 C ^{*1}	—	○ ^{*1}	—	—	—
Value	—	○ ^{*2}	—	—	—	○ ^{*3}	—

*1 C/LC 支持 Z、LZ 变址修饰。

*2 不能使用 T、ST、C。

*3 只能使用 10 进制常数(K)。

3) 计数器的复位指令

计数器和超长计数器可以通过 RST 和 ZRST 指令进行复位操作。RST 指令后，计数器 C/LC 进入复位状态。处于复位状态的定时器，各个属性值保持不变，计时停止，

OUT 指令无效。

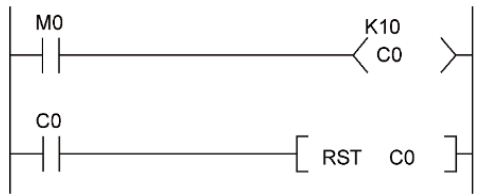


复位后计数器情况如下所示。

	计数器的类型	计数器线圈	计数器当前值	计数器目标值	计数器触点	计数器复位状态
RST	计数器	-	0	保持	OFF	锁定
	超长计数器	-	0	保持	OFF	锁定
ZRST	计数器	OFF	0	保持	OFF	释放
	超长计数器	OFF	0	保持	OFF	释放

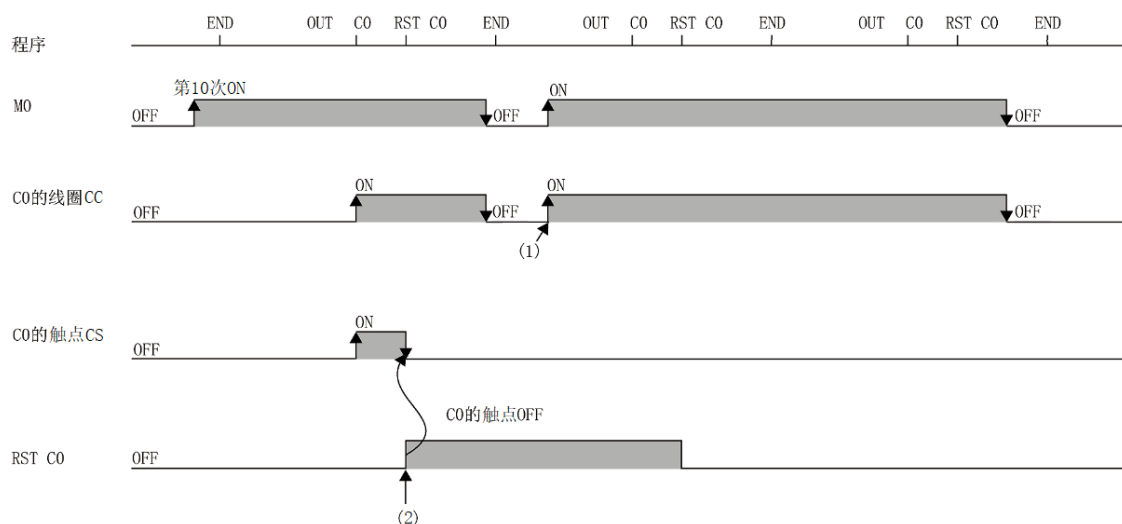
-：计数器线圈(CC/C)状态与计数器前置运算结果一致

即使将计数器线圈的输入置为 OFF，计数器的当前值也不被清除。计数器的清除(复位)及触点的 OFF 应通过 RST/ZRST 指令进行。在执行了 RST/ZRST 指令的时刻计数器值将被清除，触点也将变为 OFF。



在上述梯形图示例中，通过 M0 的 OFF→ON，C0 的线圈将变为 ON，进行当前值更新。C0 计数到时 C0 的触点将变为 ON，通过 RST C0 指令的执行 C0 的当前值将被清除。C0 的线圈保持 ON。在下一个扫描中 RST C0 指令仍处于执行状态。此时，C0 保持复位状态。

再下一个扫描中 M0 为 ON 的情况下，RST C0 指令不执行，执行 OUT C0 指令时 C0 的线圈将由 OFF→ON，因此将进行当前值的不更新。(当前值不变)



(1) 触点 CC 将变为 OFF→ON, 但 C0 被复位中, 当前值(CV)不能更新

(2) 计数值的清除、触点的 OFF

在 RST 指令执行后, 被复位的计数器此扫描周期内将不被计数。

计数器 C0 不会计数, CS 触点不会导通。

计数器的类型

有将计数器值以 16 位保持的计数器 C 及将计数器值以 32 位保持的超长计数器 LC。

计数器 C 与超长计数器 LC 是不同的软元件, 可以分别设置软元件点数。

■ 计数器 C

1 点使用 1 字。可计数范围为 0~32767。

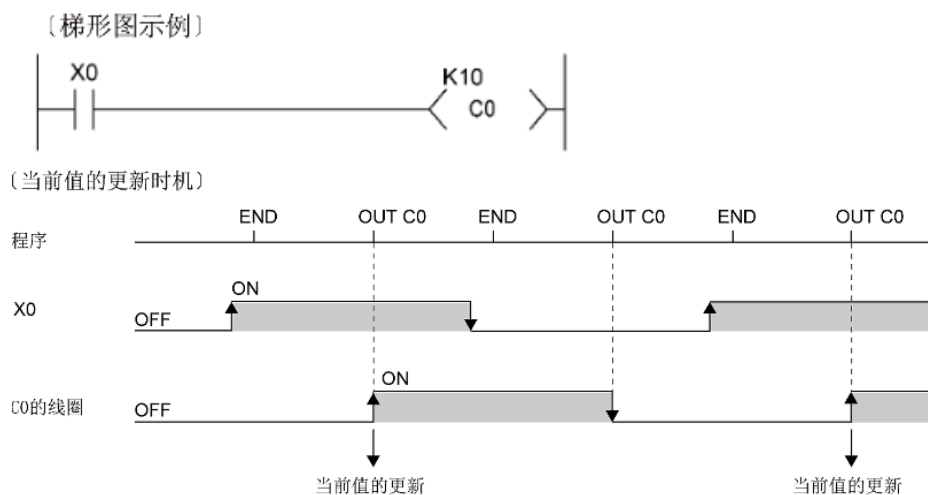
■ 超长计数器 LC

1 点使用 2 字。可计数范围为 0~2147483647。

4) 计数处理

计数器指令之前的运算结果由 OFF→ON 变化时, 计数器线圈会 OFF→ON 变化, (d) 中指定的计数器的当前值(计数值)+1, 如果计数到(当前值 $\geq$ 目标值), 常开触点将导通。

执行计数器的指令时进行计数器线圈 CC 的 ON/OFF、当前值 CV 的更新(计数值+1)以及触点 CS 的 ON/OFF 处理。当前值 CV 的更新(计数值+1)是在计数器的线圈 CC 输入的上升沿(OFF→ON)时进行。线圈 CC 输入为 OFF、ON→ON 以及 ON→OFF 时不进行当前值的更新。超长计数器处理相同。



5) 计数器目标值

扫描 OUT 计数器指令执行时，C/LC 目标值被赋值。当开始计数后，至计数器触点导通前，目标值可变。在执行 OUT 指令对计数器的目标值进行更新或计数器更新时，如果当前值大于等于更新后的目标值，计数器触点导通且变更计数器的当前值=目标值。

6) 计数器的变址

计数器中使用的软元件，可以用变址寄存器(Z/LZ)进行修饰。

例：OUT C1Z0 D0

LD LC0Z0

计数器通过(Z/LZ)变址后，变址前指向的计数器将保持当前计数。此计数器的线圈保持导通。切换回此计数器时根据线圈状态执行计数。

7) 计数器锁存

计数器设置为锁存软元件时，计数器的当前值、输出触点动作及 RST 内部状态将被锁定。使用 ZRST 指令，可以对计数器的 RST 内部状态进行复位。

8) PLC 状态与计数器

运行 stop, run 状态变化时，计数器情况如下所示。

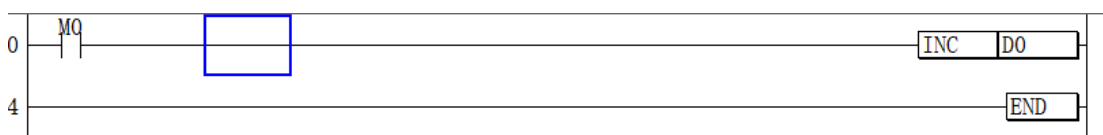
PLC 状态	计数器结构	计数器		超长计数器	
		非锁存	锁存	非锁存	锁存
run→stop	线圈	OFF	保持	OFF	保持
	当前值	0	保持	0	保持
	目标值	0	保持	0	保持
	触点	OFF	保持	OFF	保持
	复位状态	OFF	保持	OFF	保持
stop→run	线圈	OFF	保持	OFF	保持
	当前值	0	保持	0	保持
	目标值	0	保持	0	保持
	触点	OFF	保持	OFF	保持
	复位状态	OFF	保持	OFF	保持

9) 注意事项

- 目标值(CP/LCP)为 0 的情况下，其处理与 1 时相同。
 - 目标值(CP/LCP)为负值时，作为 0 处理。
- RST 指令执行中如果被跳过，定时器的复位状态将一直被保持直到 ZRST 清除为止。其间不能正常计数。
- 同一周期内多次对同一计数器执行 OUT 指令时，只要计数器线圈 OFF→ON，计数器当前值均会更新（计数值+1）。

4.3.11 数据寄存器（D）

是可存储数值数据的软元件。



4.3.12 链接寄存器（W）

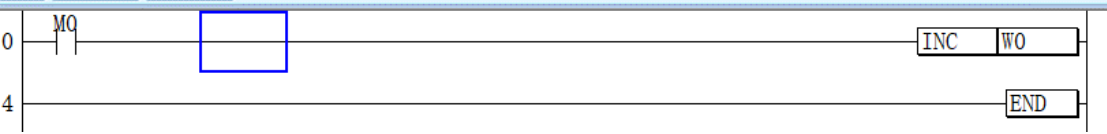
是在网络模块与 CPU 模块之间作为刷新数据时的 CPU 模块侧的软元件使用为目

的的软元件。

使用了链接寄存器（W）的网络模块的刷新

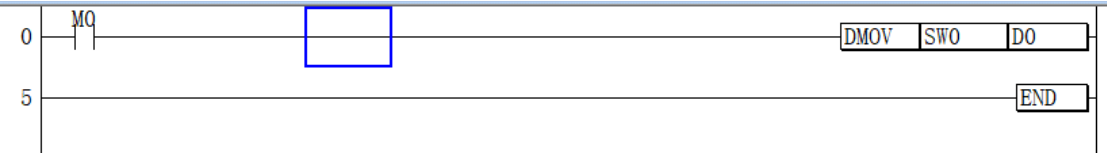
在 CPU 模块内的链接寄存器（W）与网络模块的链接寄存器（LW）之间相互收发数据。通过网络模块的参数，设置刷新范围。

未用于刷新的位置可用于其他用途。



4.3.13 链接特殊寄存器（SW）

网络的通信状态及异常检测状态的字数据信息将被输出到网络内的链接特殊寄存器。链接特殊寄存器（SW）是作为网络内的链接特殊寄存器刷新目标使用的软元件。未用于刷新的位置可用于其他用途。



4.2 系统软元件

系统软元件是系统用的软元件。其分配/容量都是固定的，用户不能更改。

4.2.1 特殊继电器（SM）

是可编程控制器内部确定规格的内部继电器，因此不能像通常的内部继电器那样用于程序中。但是，可根据需要置为 ON/OFF 以控制 CPU 模块。（XXX 页 特殊继电器一览）

4.2.2 特殊寄存器（SD）

是可编程控制器内部确定规格的内部寄存器，因此不能像通常的内部寄存器那样用于程序中。但是，可根据需要写入数据以控

制 CPU 模块。(XXX 页 特殊寄存器一览)

4.3 变址寄存器 (Z/LZ)

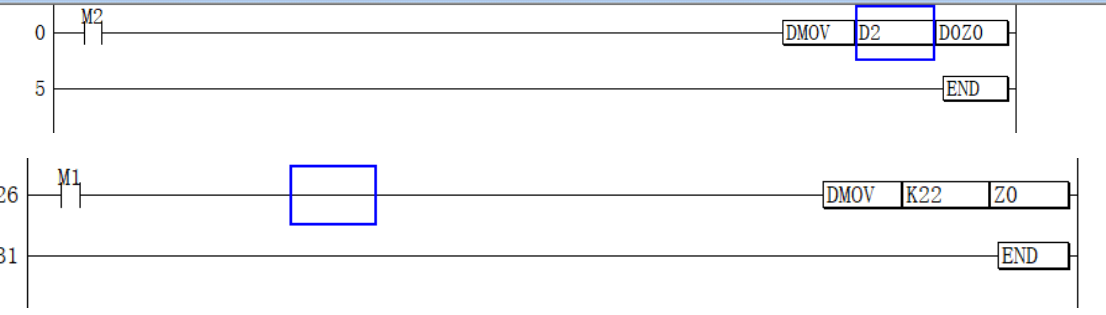
是软元件的变址修饰中使用的软元件

变址寄存器的类型

变址寄存器可分为变址寄存器 (Z) 及超长变址寄存器 (LZ)

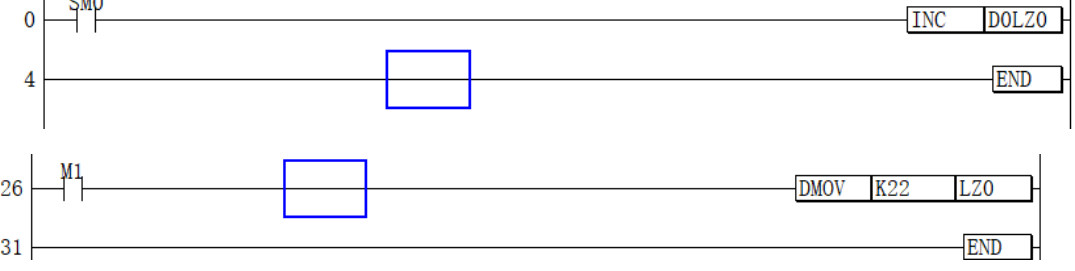
4.3.1 变址寄存器 (Z)

用于在 16 位的变址修饰中使用或者当做 16 位的普通寄存器使用



4.3.2 变址寄存器 (LZ)

用于在 32 位的变址修饰中使用或者当做 32 位的普通寄存器使用

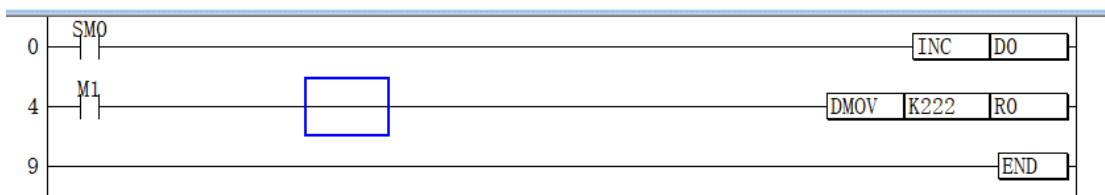


变址寄存器设置

变址寄存器 (Z) 和超长变址寄存器 (LZ) 合计可使用 16 字，可通过设置中的软元件分配设置，修改其分配的点数。

4.4 文件寄存器 (R)

是可存储数值数据的软元件。



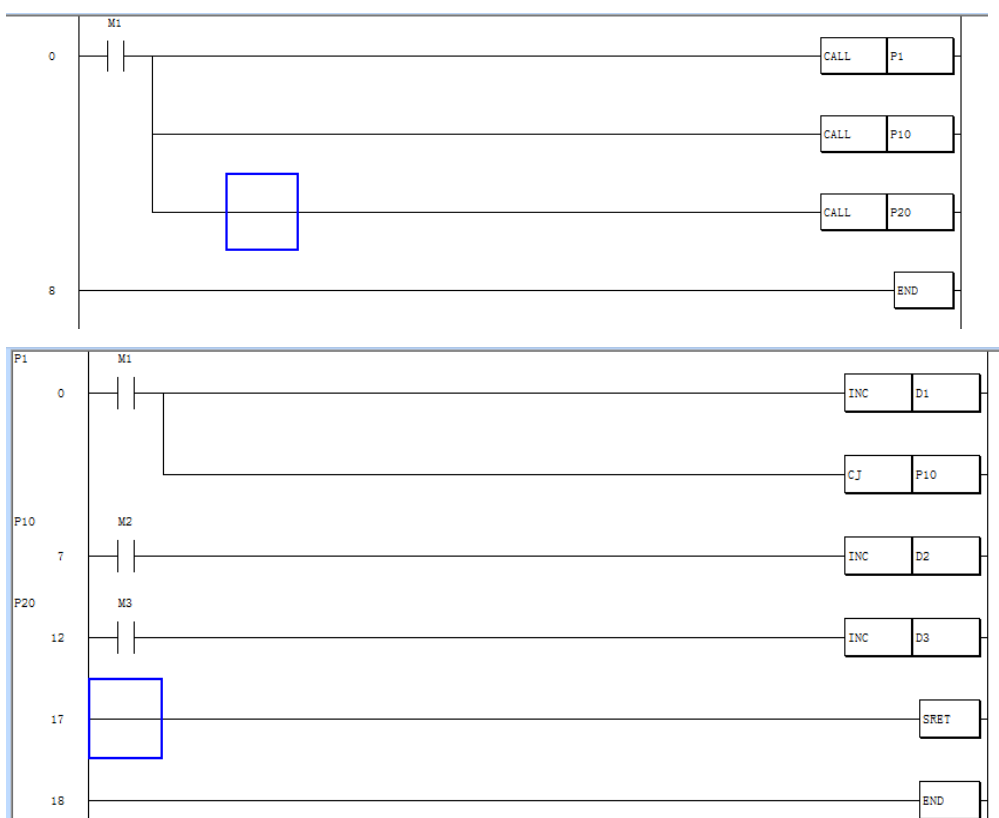
4.5 嵌套 (N)

是在主站控制指令（MC/MCR 指令）中使用，用于将动作条件通过嵌套结构进行编程的软元件。从嵌套结构的外侧以小编号（N0~N14 的顺序）进行指定。

4.6 指针 (P)

是跳转指令（CJ 指令）及子程序调用指令（CALL 指令等）中使用的软元件。指针有以下用途。

- 指定跳转指令（CJ 指令）的跳转目标。
- 指定子程序调用指令（CALL 指令等）的调用目标（待机型程序的起始）。



4.7 中断指针 I

是在中断程序起始处作为标签使用的软元件。

4.7.1 中断指针的分类

中断指针的使用需要在程序中新建中断程序，R 系列有外部输入中断，高速比较中断，恒定周期中断三种中断模式。

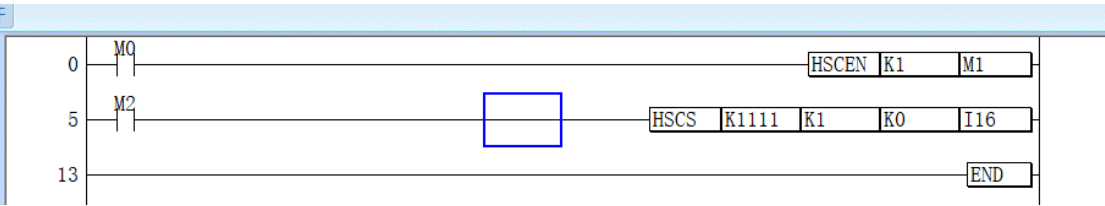
中断原因	中断指针编号	说明
外部输入中断	I0-I7	是在 CPU 模块的输入中断中使用的中断指针。最多可使用 8 点
高速比较中断	I16-I23	是在 CPU 模块的高速比较中断中使用的中断指针
恒定周期中断	I28-I31	是在通过内部定时器进行的恒定周期中断中使用的中断指针

1) 外部输入中断

其中外部输入中断的可选的中断指针为 I0-I7 与 X0-X7 一一对应,其作为外部输入中断的指针使用时需要在高速 I0 设置界面的输入设置页面设置其滤波参数（0-65535）与中断类型（不使用、上升沿、下降沿、上升下降沿）。当对应的 X 端子满足触发条件时，就会进入相应的中断程序。

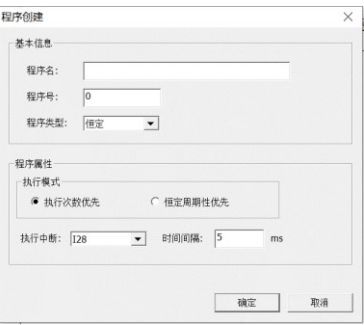
2) 高速比较中断

而高速比较中断需要使能高速计数器，结合高速比较指令共同使用。高速比较中断指针为 I16-I23。用户需要提前新建高速比较中断程序，下图中新建高速比较中断 I16。下图中高速计数器 1 被使能后，当高速计数器 1 当前值 SD1600 达到 1111 时，就会跳入 I16 对应的高速中断程序



3) 恒定周期中断

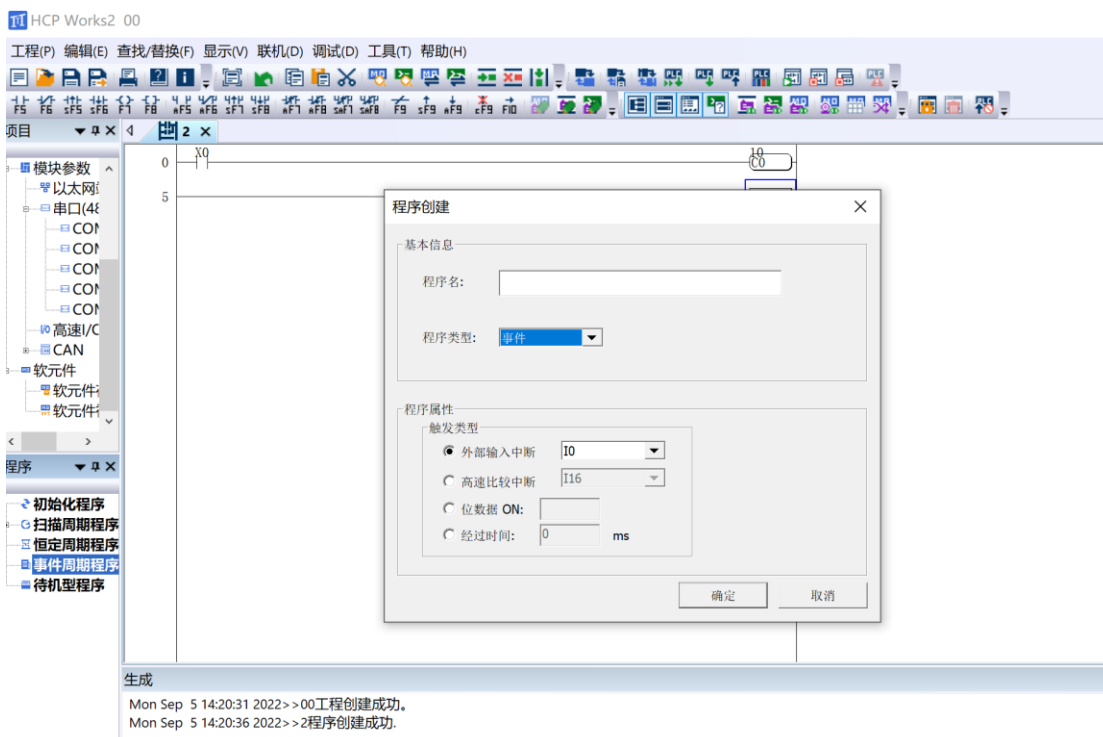
恒定周期中断通过设置新建的恒定周期程序设置的时间间隔，来控制进入恒定周期程序的时间间隔。



外部输入中断 X 端子的参数设置页面

端子	滤波(us)	中断类型
X0	2	10 不使用
X1	2	11 不使用
X2	2	12 不使用
X3	2	13 不使用
X4	2	14 不使用
X5	2	15 不使用
X6	2	16 不使用
X7	2	17 不使用

中断程序创建页面



4.7.2 中断指针的编号及中断原因的优先级

中断指针编号	中断原因	中断优先级	映射软元件	关联设置位置
I0	外部输入中断	1 ~ 5	X0	[程序创建]-[事件]-[外部输入中断]
I1	外部输入中断	1 ~ 5	X1	
I2	外部输入中断	1 ~ 5	X2	
I3	外部输入中断	1 ~ 5	X3	
I4	外部输入中断	1 ~ 5	X4	
I5	外部输入中断	1 ~ 5	X5	
I6	外部输入中断	1 ~ 5	X6	
I7	外部输入中断	1 ~ 5	X7	
I8 ~ I15	—	—	—	—
I16	高速比较中断	1 ~ 5		[程序创建]-[事件]-[高速比较中断]
I17	高速比较中断	1 ~ 5		
I18	高速比较中断	1 ~ 5		
I19	高速比较中断	1 ~ 5		

I20	高速比较中断	1 ~ 5		
I21	高速比较中断	1 ~ 5		
I22	高速比较中断	1 ~ 5		
I23	高速比较中断	1 ~ 5		
I24 ~ I27	—	—	—	—
I28	恒定周期中断	1 ~ 5		[程序创建]-[恒定]-[执行中断]
I29	恒定周期中断	1 ~ 5		
I30	恒定周期中断	1 ~ 5		
I31	恒定周期中断	1 ~ 5		
I32 ~ I63	—	—	—	—

4.7.3 多重中断优先级设置方式：

- 双击打开[参数配置]-[运行参数]-[中断]-[中断设置]
- 勾选中断设置的多重中断
- 更改各中断指针的中断优先度

项目	内容	设置范围	默认值
多重中断	设置是否允许多重中断。	•不勾选 •勾选	•不勾选
中断优先度	中断指针 I0 ~ I63 的中断优先度	1 ~ 5*1	3*2

*1 数值越小，中断优先度越高

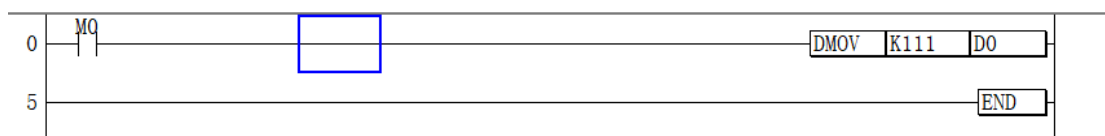
*2 多重中断为使用状态，中断优先级设置才会生效。不勾选多重中断时，所以中断均为默认优先级，设置无效。

4.8 常数

4.8.1 10 进制常数 (K)

在程序中指定 10 进制数据的软元件。以 K 进行指定。(例：K1234)

指定范围取决于使用 10 进制常数的指令的自变量数据类型。

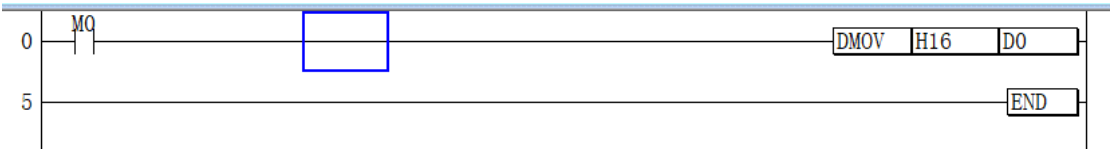


4.8.2 16 进制常数 (H)

在程序中指定 16 进制数据的软元件。以 H 进行指定。(例：H16)

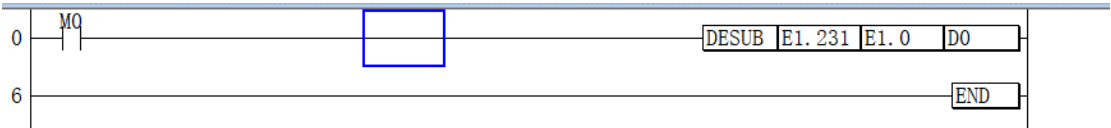
要以 BCD 指定数据时，应在 0~9 的范围内指定 16 进制数的各位。指定范围取决于使用 16 进制常数的指令的自变量数据类型。16

位时数据容量为 H0~HFFFF，32 位时数据容量为 H0~HFFFFFFF



4.8.3 实数常数 (E)

在程序中指定实数的软元件。以 E 进行指定。(例：E1.234)



4.8.4 字符串常数

是指定字符串的软元件。可使用移位 JIS 代码字符串。任一个字符串都以 NULL 字符（00H）作为字符串的结尾。以“字符串”指定。(用于在 BIN 和 ASCII 指令中使用)

5. 指令用法简介

4.1 基本指令

4.1.1 LD, LDI 指令

概要
LD、LDI 指令是连接在母线上的触点。
和后述的 AND 指令组合后，也可用在分支起点处

指令格式

执行类型	连续执行型
指令格式	LD S/LDI S

可使用软元件

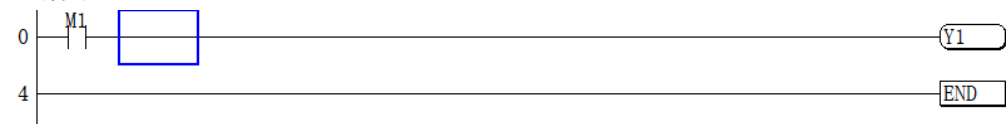
操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S	○	○	—	○*1, *2	—	—	—	—	—	

*1 支持变址修饰

*2 仅支持 T 和 C

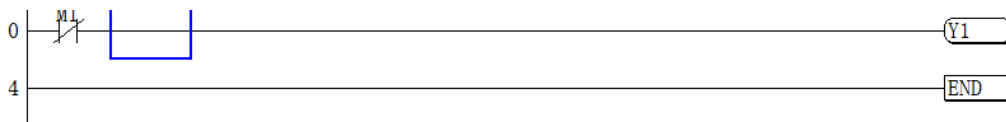
示例

1. LD 指令



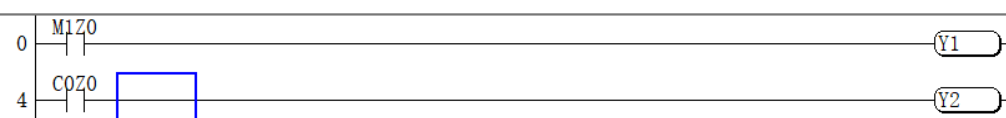
此过程中,LD M1 是 OUT Y1 的执行条件。只有当 M1 闭合后, Y1 才会输出, M1 断开, Y1 停止输出。

2. LDI 指令



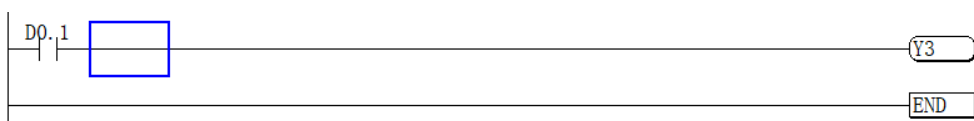
此过程中,LDI M1 是 OUT Y1 的执行条件。只有当 M1 断开后, Y1 才会输出, M1 闭合, Y1 停止输出。

3. 变址修饰



此过程中, M 和 C 使用了变址修饰, 此时实际控制 Y1,Y2 输出的软元件是变址后的 M 和 C。

4. 数据寄存器的位指定



在上图中, 实际控制 Y3 输出的软元件是 16 位寄存器 D0 中的第 1 位。(仅 16 位数据寄存器有效)

报错相关: (待添加)

注意: (待添加)

4.1.2 AND,ANI 指令

概要

AND、ANI 指令是执行串联连接 1 个触点。

串联触点的数量没有限制, 该指令可以连续多次使用。

OUT 指令后，通过触点对其他的线圈使用 OUT 指令，称为纵接输出。
只要顺序不错，这样的纵接输出可以重复使用多次

指令格式

执行类型	连续执行型
指令格式	AND S

可使用软元件

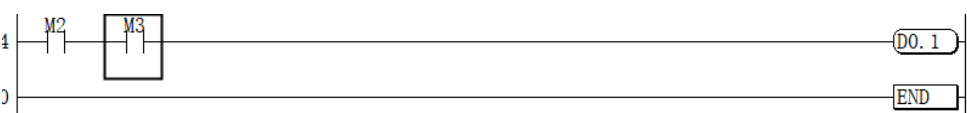
操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	○	○	—	○*1*2	—	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C

示例

1. AND 指令



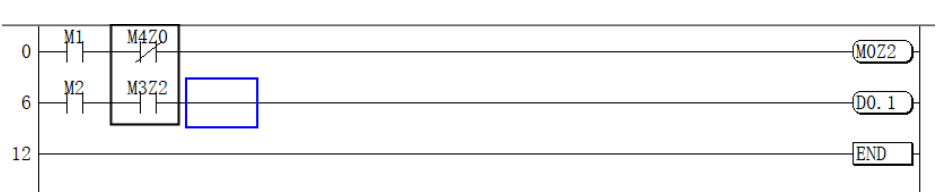
此时 M2 和 M3 同时置 ON 后，后续操作才会执行。

2. ANI 指令



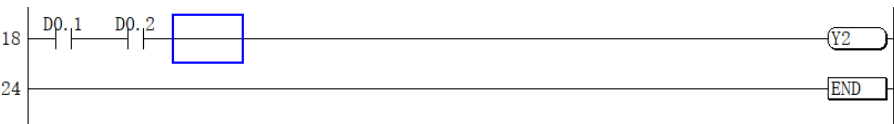
此时 M1 置 ON 同时 M4 为 OFF 时，后续操作才会执行。

3.变址修饰



此时 M4 的变址软元件控制了后续步骤的执行，例如 Z0=2 时，变址后的软元件为 M6

4.数据寄存器的位指定



此时 D0 的第一位和第二位通过 AND 指令，两个条件串联。两者同时置 ON 后，Y2 输出。

5.计数器和计时器



此时当 T0,T2 计时完成。C2 计数完成，Y1 才会输出。

报错相关：(待添加)
注意：(待添加)

4.1.3 OR.ORI 指令

概要

OR、ORI 指令可以作为并联连接 1 个触点的指令使用。
串联连接了 2 个以上的触点时，要将这样的串联回路块与其他回路并联的时候，采用后述的 ORB 指令。
OR、ORI 是从这个指令的步开始，与前面的 LD、LDI 指令的步进行并联连接。并联连接的次数不受限制。

指令格式

执行类型	连续执行型
指令格式	ORI S、OR S

可使用软元件

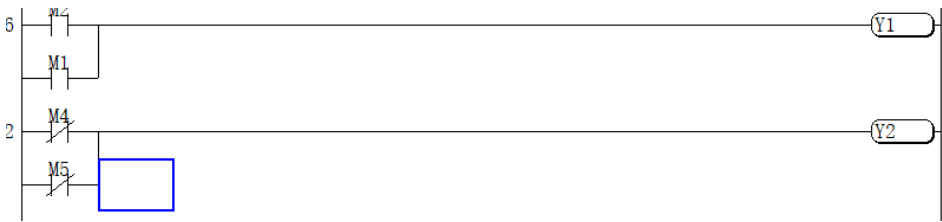
操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC LZ	K、H	E
S	○	○	—	○*1*2	—	—	—	—

*1 支持变址修饰

*2 仅支持 T、C

示例

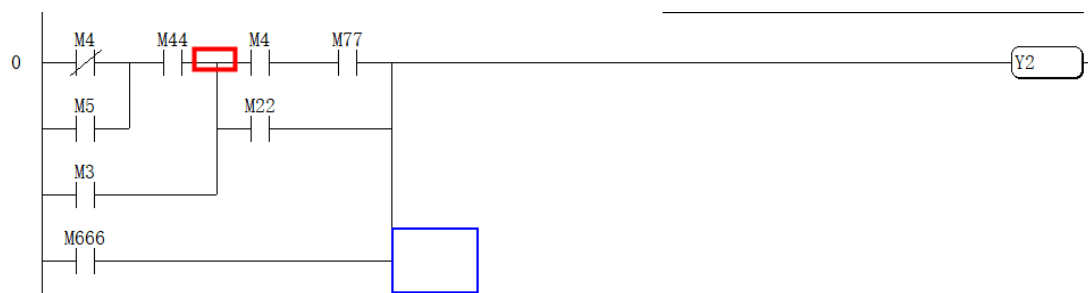
1. 指令示例



此时只要 M1 和 M2 中的任意一个软元件置 ON 后，Y1 就会输出
M4 和 M5 中，只要有一个为 OFF 时，Y2 就会输出

2. ANB 指令的关系

用 OR、ORI 指令并联连接，原则上是与前面的 LD、LDI 点连接，但是在后述的 ANB 指令后，则是与前一个的 LD、LDI 点连接。下图中红色框内即为 ANB 指令。



3. 变址修饰

OR 和 ORI 指令中使用的软元件，可以用变址寄存器(V、Z)进行修饰。



此时 M4 和 M5 的变址软元件控制了后续步骤的执行，例如 Z2=2 时，变址后的软元件为 M6

4. 数据寄存器的位指定



此时 D0 的第一位和第三位通过 OR 指令，两个条件并联。两者有任意一个置 ON 后，Y22 输出。

报错相关：(待添加)

注意：(待添加)

4.1.4 LDP、LDF、ANDP、ANDF、ORP、ORF 指令

概要

LDP,ANDP,ORP 指令是检测上升沿的触点指令，仅在指定位软元件的上升沿(从 OFF 改变到 ON 的时候)时，接通 1 个运算周期。

LDF、ANDF、ORF 指令是检测下降沿的触点指令，仅在指定位软元件的下降沿(从 ON 改变到 OFF)时，接通 1 个运算周期。

指令格式

指令	内容	执行类型	指令格式
LDP	取脉冲上升沿	脉冲执行型	与脉冲执行型一致
LDF	取脉冲下降沿		
ANDP	与脉冲上升沿		
AMDF	与脉冲下降沿		

ORP	或脉冲上升沿		
ORF	或脉冲下降沿		

可使用软元件

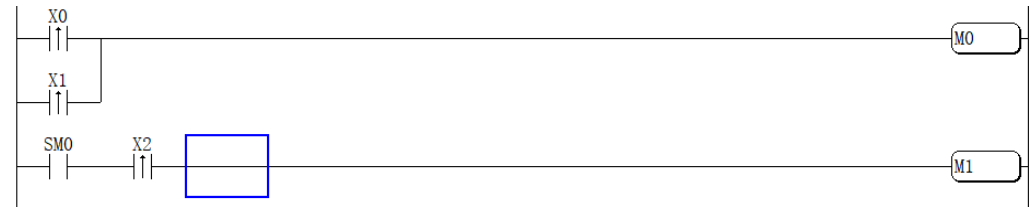
操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S	KnX、KnY、KnM、KnL、KnS	T、ST、C、D、W、SD、S	Z	LC	LZ	K、H	E
		D. X、SW. X、R. X	M、KnF、KnB、KnSB、KnS	W、R					
S	○	○	—	○*1*2	—	—	—	—	—

*1 支持变址修饰

*2 仅支持 T, C

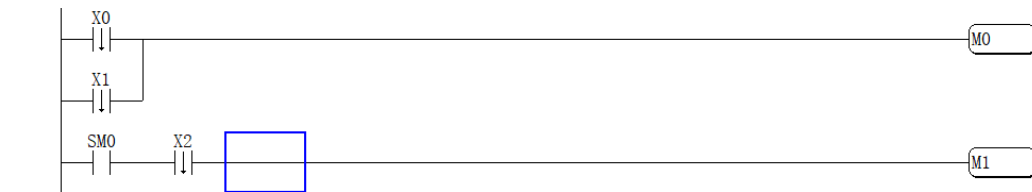
示例

1. LDP、ANDP、ORP 指令(检测到上升沿时运算开始、串联连接、并联连接)



X000 ~ X002 从 OFF 变成 ON 时，M0 或 M1 只维持 1 个运算周期为 ON

2. LDF、ANDF、ORF 指令(检测到下降沿时运算开始、串联连接、并联连接)



X000 ~ X002 从 ON 变成 OFF 时，M0 或 M1 只维持 1 个运算周期为 ON。

1. 数据寄存器(D)的位指定



LDP、LDF、ANDP、ANDF、ORP、ORF 指令用的软元件中，可以指定数据寄存器(D)的位。D0 的第 3 位决定从 OFF 变 ON 时，LDP 触点 ON(导通)/OFF(不导通)。

可以使用的数据寄存器仅 16 位的有效。

请从低位开始按照 0,1,2,...9,A,B,...F 的顺序指定位编号

报错相关：(待添加)

注意：(待添加)

4.1.4 LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI 指令

概要

LDPI,ANDPI,ORPI 指令是检测非上升沿的触点指令， 仅在指定位软元件的非上升沿(从 OFF 改变到 ON 的时候)时， 接通 1 个运算周期。

LDFI、 ANDFI、 ORFI 指令是检测非下降沿的触点指令， 仅在指定位软元件的非下降沿(从 ON 改变到 OFF)时， 接通 1 个运算周期。

指令格式

指令	内容	执行类型	指令格式
LDPI	非脉冲上升沿	脉冲执行型	与脉冲执行型一致
LDFI	非脉冲下降沿		
ANDPI	非与脉冲上升沿		
AMDFI	非与脉冲下降沿		
ORPI	非或脉冲上升沿		
ORFI	非或脉冲下降沿		

可使用软元件

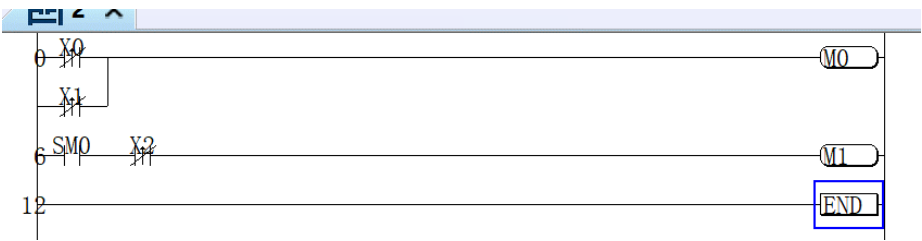
操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC LZ	K、H	E
S	○	○	—	○*1*2	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C

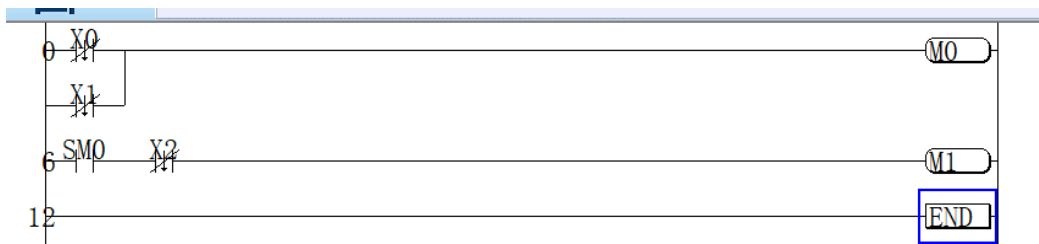
示例

1. LDPI、ANDPI、ORPI 指令(检测到非上升沿时运算开始、串联连接、并联连接)



除了 X000 ~ X002 从 OFF 变成 ON 时， M0 或 M1 维持为 ON

2.LDFI、 ANDFI、 ORFI 指令(检测到非下降沿时运算开始、串联连接、并联连接)



除了 X000 ~ X002 从 ON 变成 OFF 时，M0 或 M1 为 ON。

2. 数据寄存器(D)的位指定



LDPI、LDFI、ANDPI、ANDFI、ORPI、ORFI 指令用的软元件中，可以指定数据寄存器(D)的位。D0 的第 3 位决定从 OFF 变 ON 时，LDP 触点 ON(导通)/OFF(不导通)。

可以使用的数据寄存器仅 16 位的有效。

请从低位开始按照 0,1,2,...9,A,B,...F 的顺序指定位编号

报错相关：(待添加)

注意：(待添加)

4.2 结合指令

4.2.1 ANB 指令

概要

当分支回路(并联回路块)与前面的回路串联连接时，使用 ANB 指令。

分支的起点使用 LD、LDI 指令，并联回路块结束后，可以使用 ANB 指令和前面的回路串联连接。

有多个并联回路的时候，对每个回路块使用 ANB 指令，从而连接。

指令格式

执行类型	连续执行型
指令格式	ANB

可使用软元件

	位		字			双字		常数	
操作数	X、Y、M、L、S	D.X、W.X、S	KnX、KnY、KnM、KnL、KnS	T、ST、C、D、W、SD、S	Z	LC	LZ	K、H	E
	M、F、B、SB、S	D.X、SW.X、R.X	M、KnF、KnB、KnSB、KnS	W、R					
S	无对象软元件								

示例

1. ANB 指令(回路块的串联连接)



报错相关：(待添加)

注意：(待添加)

4.2.2 ORB 指令

概要

由 2 个以上的触点串联连接的回路称为串联回路块

指令格式

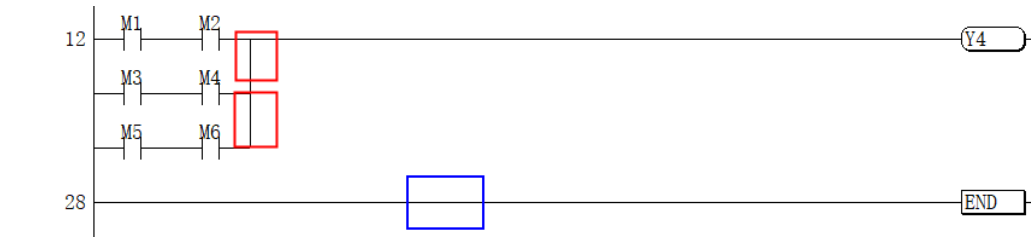
执行类型	连续执行型
指令格式	ORB

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	无对象软元件								

示例

1. ORB 指令(回路块的并联连接)



报错相关：(待添加)

注意：(待添加)

4.2.3 INV 指令

概要

INV 指令，是将 INV 指令执行前的运算结果反转的指令，无需指定软元件编号

执行类型	连续执行型
指令格式	INV

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC LZ	K、H	E
S	无对象软元件							

示例

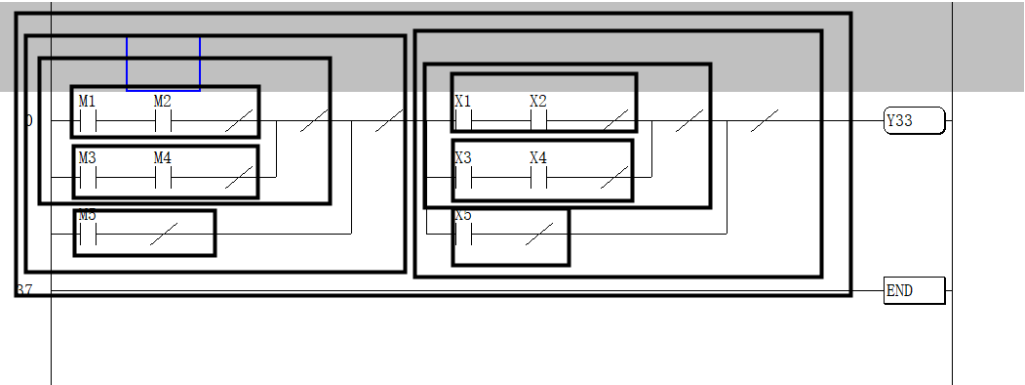
1. INV 指令(运算结果的反转)



上图中，X000 为 OFF 时，Y000 为 ON，如果 X000 为 ON 时，则 Y000 为 OFF。
INV 指令可以在与串联触点指令(AND、ANI、ANDP、ANDF 指令)相同的位置处编程。
不能像指令表上的 LD、LDI、LDP、LDF 那样与母线连接，也不能像 OR、ORI、ORP、ORF 指令那样独立地与触点指令并联使用。

2. INV 指令的动作范围

在包含 ORB 指令、ANB 指令的复杂的回路中编写 INV 指令时，INV 指令的动作范围如下图所示。



INV 指令的功能，是将 INV 指令执行前存在的 LD、LDI、LDP、LDF 指令以后的运算结果反转。

因此，如上图所示，在 ORB 指令、ANB 指令中编程时，从各自的 INV 指令的位置上见到的 LD、LDI、LDP、LDF 以后的块作为 INV 运算的对象。

报错相关：(待添加)

注意：(待添加)

4.2.4 MEP、MEF 指令

概要

MEP、MEF 指令是使运算结果脉冲化的指令，不需要指定软元件编号。

1) MEP

在到 MEP 指令为止的运算结果，从 OFF→ON 时变为导通状态。

如果使用 MEP 指令，那么在串联了多个触点的情况下，非常容易实现脉冲化处理。

2) MEF

在到 MEF 指令为止的运算结果，从 ON→OFF 时变为导通状态。

如果使用 MEF 指令，那么在串联了多个触点的情况下，非常容易实现脉冲化处理。

指令格式

执行类型	脉冲执行型
指令格式	MEP、MEF

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z		LC	LZ	K、H	E
S	无对象软元件									

示例

1. MEP 指令(运算结果的上升沿时为 ON)



2. MEF 指令(运算结果的下降沿时为 ON)



报错相关：(待添加)

注意：(待添加)

4.3 输出指令

4.3.1 OUT 指令

概要

OUT 指令是对输出继电器(Y)、辅助继电器(M)、状态(S)、定时器(T)、计数器(C)进行线圈驱

动的指令。

指令格式

执行类型	连续执行型
指令格式	OUT S

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、KnSB、KnS	T、ST、C、D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	○*1*2*4	○	—	○*2*3	—	○	—	—	—

*1 不支持 X

*2 支持变址修饰

*3 仅支持 ST, T, C

*4 支持 F

示例

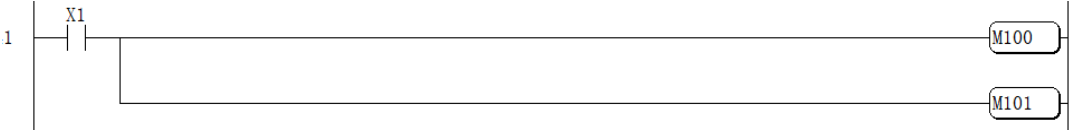
3. 使用位软元件时

用 OUT 指令编写的软元件，根据驱动触点的状态执行 ON/OFF。

并联的 OUT 命令能够多次连续使用。

下面的程序举例中，接着 OUT M100 的 OUT M101 就是这个意思。

但是，对同一软元件编号，使用多个 OUT 指令时，会变成双重输出(双线圈)，请注意。



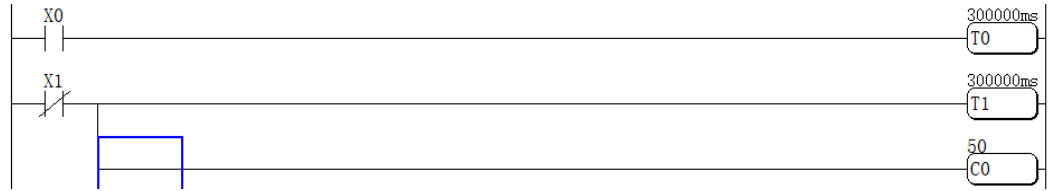
4. 使用定时器和计数器

在针对定时器的计时线圈和计数器的计数线圈的 OUT 指令后需要加上设定值。

设定值可以使用 10 进制数(K)直接指定，也可以使用数据寄存器(D)或扩展寄存器(R)间接指定。

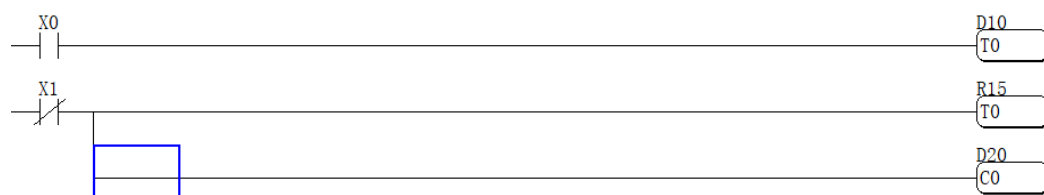
1) 直接指定

使用直接指定时，使用常数，直接确定了计时器和计数器的目标值



2) 间接指定

使用间接指定时，用户可以修改代表计数器和计数器目标值的对应软元件

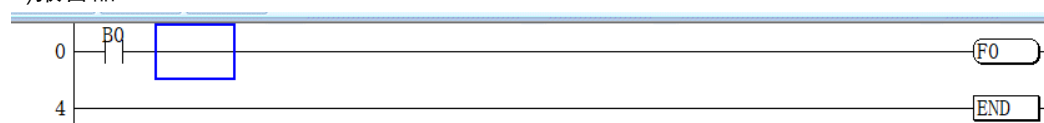


3) 定时器和计数器的范围



定时器(T,ST)、计数器(C,LC)	设定范围(K 的值或 D、R 的当前值)	实际的设定值
1ms 定时器 OUTHS	1 ~ 32,767	0.001 ~ 32.767 秒
10ms 定时器 OUTH	1 ~ 32,767	0.01 ~ 327.67 秒
100ms 定时器 OUT	1 ~ 32,767	0.1 ~ 3,276.7 秒
16 位计数器 C	1 ~ 32,767	同左
32 位计数器 LC	-2,147,483,648 ~ +2,147,483,647	同左

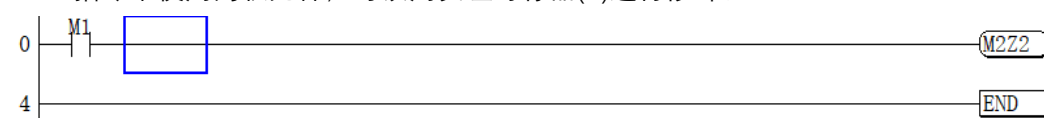
4)报警器



注意：关于定时器 T,累计定时器 ST,计数器 C,超长定时器 LC 的详细动作内容请参阅 R 系列应用篇之软元件。

3.变址修饰

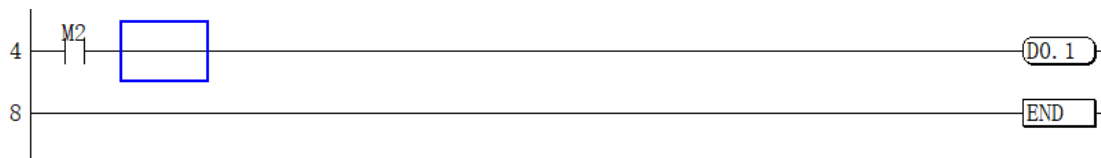
OUT 指令中使用的软元件，可以用变址寄存器(Z)进行修饰。



此时 OUT 指令置 ON 的是变址后的 M 软元件，例如 Z2 为 2 是，置 ON 的 M 是 M4。

4.数据寄存器的位指定

OUT 指令使用的软元件中，可以指定数据寄存器(D)的位



此时 OUT 指令值 ON 的是 D0 字软元件中的第 1 位，仅能在 16 位的寄存器上使用。

报错相关：(待添加)

注意：(待添加)

4.3.2 SET、RST 指令

概要

1) 位软元件的置位(SET 指令[动作保持])

SET 指令是当指令输入为 ON 时，对输出继电器(Y)、辅助继电器(M)、状态(S)以及字软元件的位指定(Dn.b)ON 的指令。

此外，即使指令输入为 OFF，通过 SET 指令置 ON 的软元件也可以保持 ON 动作。

2) 位软元件的复位(RST 指令[解除动作保持])

RST 指令是对输出继电器(Y)、辅助继电器(M)、状态(S)、定时器(T)、计数器(C)以及字软元件的位指定(Dn.b)进行复位的指令。

可以对用 SET 指令置 ON 的软元件进行复位(OFF 处理)。

3) 字软元件的当前值清除(RST 指令[当前值及寄存器的清除])

RST 指令是清除(T)、计数器(C)、数据寄存器(D)、扩展寄存器(R)和变址寄存器(V)、(Z)的当前值数据的指令。

此外，要将数据寄存器(D)和变址寄存器(V)、(Z)的内容清零时，也可使用 RST 指令。(使用常数为 K0 的 MOV 传送指令也可以得到相同效果。)

另外，使用 RST 指令也可以对累计定时器 T246~T255 的当前值和触点复位。

可以对于同一软元件，多次使用 SET、RST 指令，而且顺序也可随意

指令格式

执行类型	连续执行型
指令格式	SET S/RST S

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
SET	○*1*2	○	—	—	—	—	—	—	—
RST	○*1*2	○	—	○*2*3	—	—	—	—	—

*1 不支持 X

*2 支持变址修饰

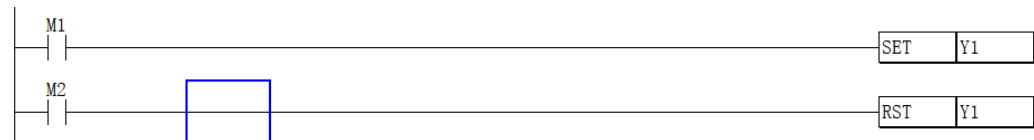
*3 仅支持 T,C

示例

1. 使用位元软件时

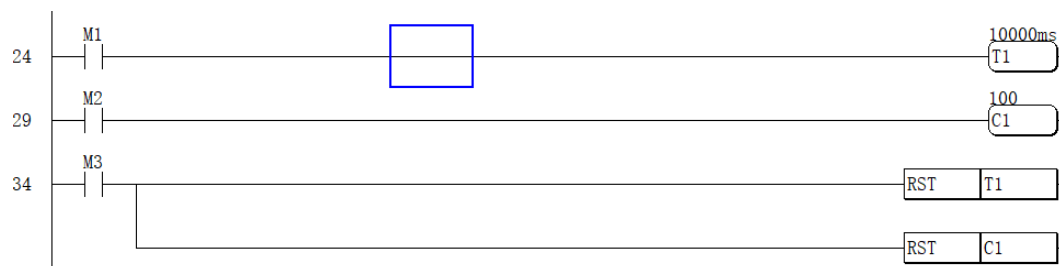
并联的 SET 指令，可以连续使用多次。

下面的程序举例中，SET Y1 后接着 RST Y1 的程序就是这个情况



2. 使用字软元件(定时器、计数器)时

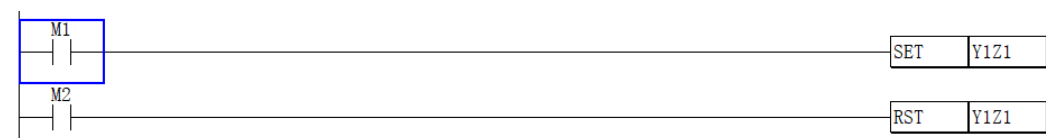
使用 RST 指令复位计数器和累计型定时器



图中，M1 和 M2 分别控制着计时和计数的开启，而 M3 导通后，将会将计时器 T1 和计数器 C1 清零。

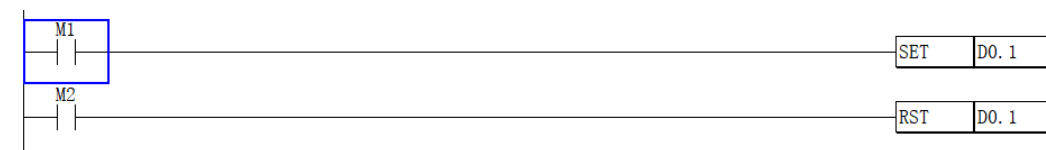
3. 变址修饰

SET 指令和 RST 指令中使用的软元件，可以采用变址寄存器(V、Z)进行修饰。



4. 数据寄存器(D)的位指定

在 SET 指令和 RST 指令使用的软元件中，可以指定数据寄存器(D)的位



图中的 M1 导通后，会将 D0 的第一位置 ON，而 M2 导通后会将其复位，可以使用的数据寄存器仅 16 位的有效。

报错相关：(待添加)

注意：(待添加)

4.3.3 PLS、PLF 指令

概要

使用 PLS 指令后，仅在驱动输入 ON 以后的 1 个运算周期内，对象软元件动作。

使用 PLF 指令后，仅在驱动输入 OFF 以后的 1 个运算周期内，对象软元件动作。

例如，驱动输入保持为 ON 时，可编程控制器从 RUN→STOP→RUN 时，PLS M0 指令动作，但是 PLS M600(电池后备)不动作。(后侧的 RUN 时)。这是因为 STOP 过程中 M600 仍然保持了动作状态。

指令格式

执行类型	连续执行型
指令格式	PLS S/PLF S

可使用软元件

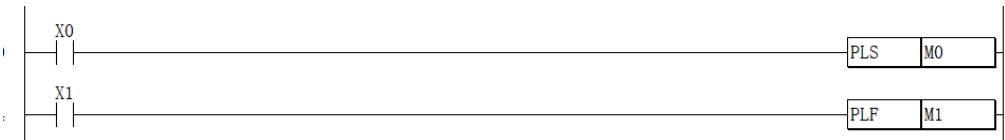
操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	○*1*2	—	—	—	—	—	—	—	—

*1 只支持 Y、M

*2 支持变址修饰

在下图中，X0 从 OFF 变为 ON 时，只有一个运算周期的 M1 为 ON。

在下图中，X1 从 ON 变为 OFF 时，只有一个运算周期的 M1 为 ON。



报错相关：(待添加)

注意：(待添加)

4.3.4 ALT/交替输出

概要

输入为 ON 时，使位软元件反转(ON ↔ OFF)用的指令。

指令格式

16 位指令	ALT(P)	32 位指令	DALT (P)
执行类型	连续执行型		
指令格式	SALT (P) D	DALT (P) D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D	○*1	○	—	—	—	—	—	—	—

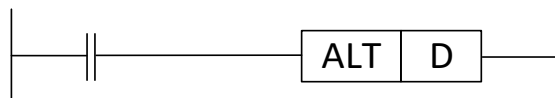
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	进行反转 的位软元件	1~32767	1~32767	实数	实数

示例

交替输出指令



当驱动条件成立时，对位元件(d)执行ON/OFF反转

该指令的功能是能流有效时，将(d)元件的状态反转。其中(d)为位变量元件

如果通过ALT指令编程，将在每个运算周期进行动作取反

报错相关：（待添加）

注意：（待添加）

4.3.5 ZRST/成批复位

概要

2 个指定的软元件之间执行成批复位的指令。

用于在中断运行后从初期开始运行时，以及对控制数据进行复位时。

指令格式

16 位指令	ZRST(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	ZRST (P) D1 D2	-	

可使用软元件

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D1	○*1*3	—	—	○*2*3	—	—	—	—	—
D2	○*1*3	—	—	○*2*3	—	—	—	—	—

*1 仅支持 Y.M.S

*2 仅支持 T,C,D,R

*3 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D1	成批复位的 最前端的位 / 字软元件 编号	-32768~32767	—	实数	—
D2	成批复位的 末尾的位 / 字软元件编 号	-32768~32767	-	实数	—

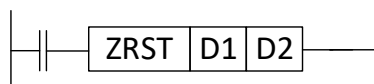
示例

1. 16 位运算(ZRST、ZRSTP)

将同一种类的 D1 ~ D2 全部复位。

D1, D2 为位软元件时

D1 ~ D2 的软元件范围全部被写入 OFF(复位)。



执行前	D ₂	...	D ₁ +9	D ₁ +8	D ₁ +7	D ₁ +6	D ₁ +5	D ₁ +4	D ₁ +3	D ₁ +2	D ₁ +1	D ₁
-----	----------------	-----	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	----------------



OFF

执行后	OFF	...	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

D1,D2 为字软元件时

D1 ~ D2 的软元件范围全部被写入 K0。

执行前	D ₂	...	D ₁ +9	D ₁ +8	D ₁ +7	D ₁ +6	D ₁ +5	D ₁ +4	D ₁ +3	D ₁ +2	D ₁ +1	D ₁
-----	----------------	-----	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	-------------------	----------------



K0

执行后	K0	...	K0	K0	K0	K0	K0	K0	K0	K0	K0	K0
-----	----	-----	----	----	----	----	----	----	----	----	----	----

相关指令

指令	功能
RST	单独复位指令
FMOV	成批写入指令

报错相关:待添加

注意:

1. 指定软元件时的注意事项

D1, D2 指定为同一种类的软元件, 且 D1 编号 ≤ D2 编号。

D1 编号 > D2 编号时, D1 中指定的软元件仅仅复位 1 点。

4.3.6 ZSET 成批置位

概要

批量置位位软元件

指令格式

ZSET 批量置位指令	
16 位指令	ZSET
执行类型	连续执行型、脉冲执行型
指令格式	ZSET(P) D1 D2

设置数据

操作数	内容	数据	
		16 位范围	16 位类型
D1	置位起始软元件编号	-	实数 (2 进制)
D2	置位结束软元件编号	-	实数 (2 进制)

*注:D1,D2 的限制范围与配置的软元件范围一致

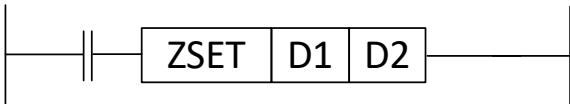
可使用软元件(ZSET)

操 作 数	位		字				双字		常数	
	X、Y、M、L、SM、F、B、SB、S	D.X、W.X、SD.X、SW.X、R.X	KnX、KnY、KnM、KnL、KnSM、KnF、KnB、KnSB、KnS	T、ST、C、D、W、Z			LC	LZ	K、H	E
S	○ ^{*1}	○	-	-	-	-	-	-	-	-
D	○ ^{*1}	○	-	-	-	-	-	-	-	-

*1 支持变址修饰

1.1 功能和动作说明

16 位运算(ZSET (P))



当指令执行时，将从(D1)中指定的位软元件到(D2)结束的位软元件全部置 ON。



错误：
操作数 D1>D2 时（错误代码：待添加）
操作数（D1），（D2）中指定的软元件范围超出相应软元件的范围时。（错误代码：6004）
注意事项：D1=D2 时不执行，不报错

4.4 主控指令

4.4.1 MC、MCR 指令

概要
执行 MC 指令后，母线(LD、LDI 点)移动到 MC 触点之后。
使用 MCR 指令，可以将其返回原来的母线位置。
通过更改软元件编号 Y、M，可以多次使用 MC 指令。
但使用同一软元件编号时，和 OUT 指令相同，会出现双线圈输出。

指令格式

执行类型	连续执行型
指令格式	MC N D /MCR

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
MC (N)	—	—	—	—	—	—	—	—	—	
MC (D)	○*1	—	—	—	—	—	—	—	—	
MCR	—	—	—	—	—	—	—	—	—	

*1 仅支持 Y、M

示例

1. MC、MCR 指令(连接到公共触点、解除连接到公共触点)

执行 MC 指令后，母线移动到 MC 触点之后。
MC 触点后的母线上连接的驱动指令，只在 MC 指令执行时才执行各个动作，不执行 MC 指令时为 OFF 执行(与触点 OFF 时的动作相同)。
在如下所示的程序举例中，当输入 X0 为 ON 时，则执行从 MC 到 MCR 的指令，
但是当 X0 为 OFF 时，各个驱动软元件的动作如下。
变为 OFF 的软元件: 定时器(累计定时器除外)，用 OUT 指令驱动的软元件
保持状态的软元件: 累计定时器、计数器、用 SET/RST 指令驱动的软元件。



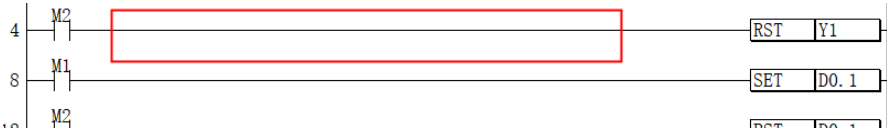
注：当使用嵌套时，嵌套级 K 的编号按顺序增大，只有 K0 才能嵌套 K1，最大嵌套等级是 15 (N0-N14)

报错相关：(待添加)
注意：(待添加)

4.5 程序流程

4.5.1 NOP 指令

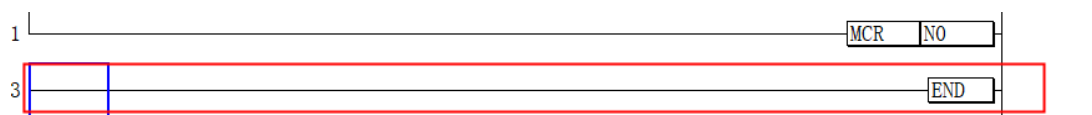
概要
NOP 指令是表示空操作的指令，下图中框内即为 NOP 指令。



报错相关：(待添加)
注意：(待添加)

4.5.2 END 指令

概要
在 R 系列小型 PLC 中,END 指令为表示程序结束的指令，由 IDE 自动填充，用户不可编辑。



报错相关：（待添加）

注意：（待添加）

4.5.3 STL、RETSTL 指令

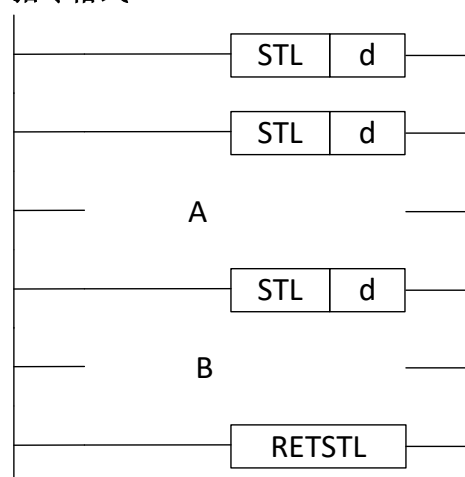
概要

使用步进梯形图指令，对使用步进继电器（S）标识的梯形图程序进行自动控制。

STL：开始步进梯形图。

RETSTL：结束步进梯形图。

指令格式



AB为步进梯形图程序

STL (d)

RETSTL

设置数据

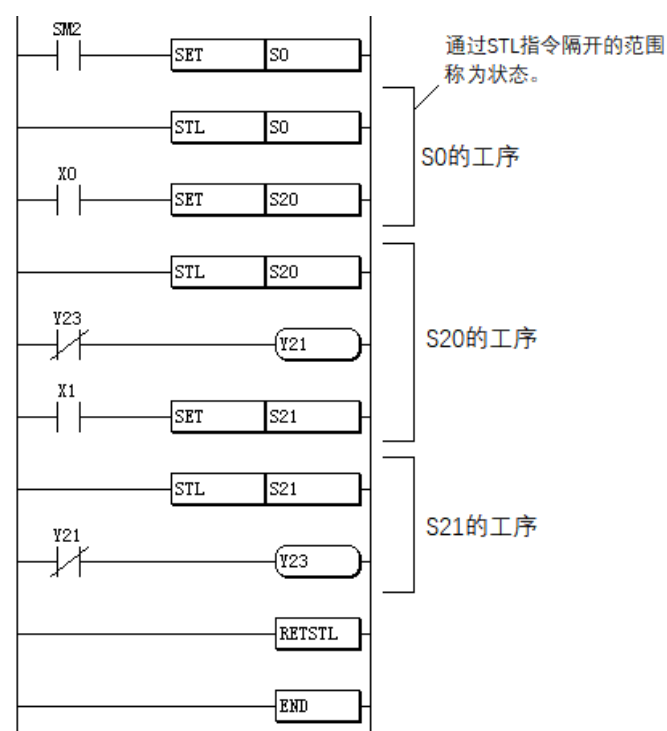
操作数	内容	范围	数据类型	掉电锁存
(d)	状态分配目标步进继电器编号	0-4095	位	否

可使用软元件

操作数	位	字		双字		常数	
	X、Y、M、L、 SM、F、B、SB、S	T、ST、C、D、 W、SD、SW、R	Z	LC	LZ	K、H	E
(d)	仅允许 S	—	—	—	—	—	—

功能

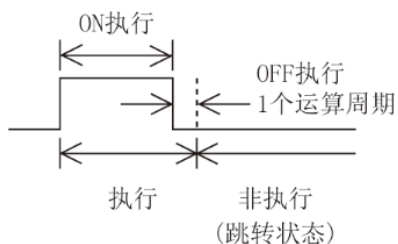
- 1. 使用步进梯形图指令的程序以机械的动作为基础，按各工序分配步进继电器 S，作为连接在状态触点(STL 触点)中的回路，进行输入条件和输出控制的顺控编程。
- 2. 在步进梯形图中，把步进继电器 S 当作 1 个控制工序，在其中进行输入条件和输出控制的顺控程序。由于工序推进时，前工序就变为不执行，所以可以通过各工序的简单顺控进行机械控制。
- 3. 通过 STL 指令指定的步进继电器编号被分配给状态。状态的开始、结束通过 SET 指令、OUT 指令、RST 指令、ZRST 指令执行。
- 4. 对于一连串的步进梯形图，要以初始状态为起始，按照状态迁移的顺序编程。在步进梯形图的最后以 RETSTL 指令作为结束。连续进行步进梯形图编程时，最后的步进梯形图以外，可省略 RETSTL 指令。



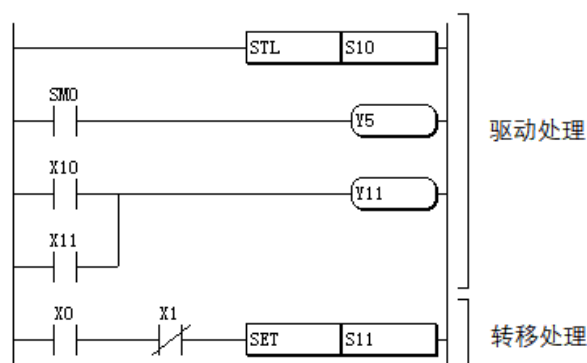
- 5. 步进梯形图的编程方式和继电器梯形图编程一样，只是使用状态 (=各个步进继电器) 按照机械控制的工艺流程来编程。可以认为状态和继电器一样，是由驱动线圈和触点 (STL 触点)构成的。线圈驱动使用 SET 指令和 OUT 指令，触点使用 STL 指令。
- 6. 与状态相连的内部回路动作如下所示。

当前状态 (STL S20)	下一个状态 (SET S21)	回路内部梯形图动作
ON 执行	未执行	与此相连的回路通过 STL 输出动作。
ON 执行	SET 执行	满足状态迁移条件时，下一个状态的步进继电器被置位，当前状态的步进继电器被清零。在这个扫描周期内这两个步进继电器都是 ON。
OFF 执行 (第 1 个扫描周期)	-	不论前触点状态如何，OUT 指令会 OFF 执行。
OFF 执行(之后)	-	指令不执行 (被跳过)。

- 7. 内部回路的执行状态的时序图如下所示。

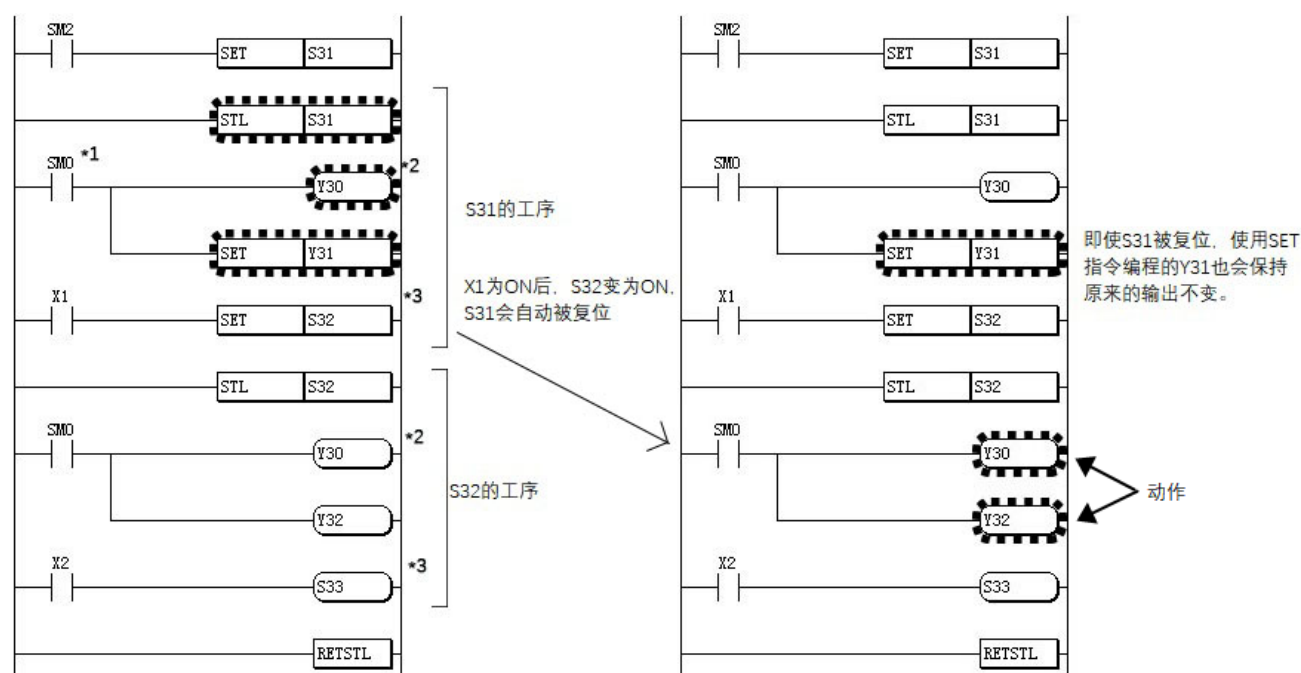


8. 各状态具备对负载的驱动处理、指定转移目标以及指定其转移条件的三个功能。如以下程序所示，按照先对负载执行驱动处理，然后执行转移处理的顺序执行。没有负载的状态中，不需要进行驱动处理。



程序动作

步进梯形图程序如下所示执行动作。



- *1 输出驱动必须有触点编程。
- *2 在不同状态中，可以重复输出线圈。
- *3 对步进继电器的 OUT 指令和 SET 指令，自动复位迁移前的步进继电器，且自我保持。
- *4 不能重复使用相同的步进继电器编号。

相关软元件

注意事项

- 未设置锁定的步进继电器(S)通过电源 ON→OFF 以及 RUN→STOP 被清除。状态有效时如果将电源 ON→OFF 以及 RUN→STOP，将无法从中途的工序重新开始。
- 多任务编程时，请避免在不同的用户程序中使用同样编号的步进继电器（S）表示状态。
- 当用户梯形图程序使用多组状态迁移（=使用多组步进继电器实现）来实现相互独立的的不同组工序控制时，请不要在 STL-RETSTL 包围的步进梯形图内步直接使用 SET 指令控制其他工序控制组的步进继电器 S。此时可以使用 M 软元件等，在 STL-RETSTL 范围外改变欲控制的其他工序控制组的步进继电器 S。
- 步进梯形图指令内部不能使用 MC、MCR、CJ 指令，也不能使用这些指令在一个扫描周期内重复执行步进梯形图程序。
- STL、RETSTL 不能使用于 FOR、NEXT 指令内部。
- STL、RETSTL 不能使用于中断程序、恒定中断程序、子程序。
- STL 命令不能使用同样编号的步进继电器（S）。

出错

监控时的运算出错代码。

出 错 代 码 (SD4)	内 容	错 误 时 动 作
6617	应该从母线开始的指令没有连接在母线上。 STL 中使用了指针 P。	停止运行
6618	主程序以外的程序(中断、子程序、恒定中断程序等) 使用了只能在主程序中使用的指令（STL、RETSTL、MC、MCR）。	停止运行
6619	FOR-NEXT 之间有不能使用的指令 STL、RETSTL、MC、MCR。	停止运行
6625	使用同样编号的步进继电器。	停止运行
6626	STL-RETSTL 之间有不能使用的指令 MC、MCR、CJ。	停止运行
6627	STL、RETSTL 关系不匹配。	停止运行

4.5.4 CALL、RESET 指令

概要

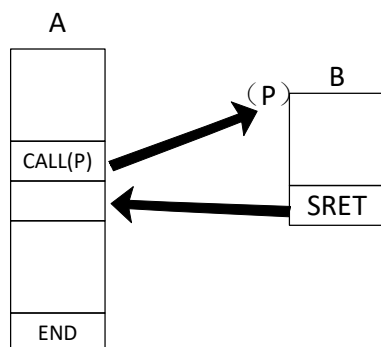
通过 CALL 指令，可以调用子程序并可以返回。

CALL Pn: 执行指针(Pn)的子程序。

SRET: 子程序结束并返回。

指令格式

执行 CALL(P) 指令时，将执行指针(P)的子程序。CALL(P)指令可以执行同一程序文件内的指针中指定的子程序及通用指针中指定的子程序。



CALL (d)

SRET

子程序:指从标签 Pn 到 RET 指令为止的程序。

设置数据

操作数	内容	范围	数据类型	掉电锁存
(d)	子程序指针	0-1023	word	否

可使用软元件

操作数	位	字		双字		其他	
	X、Y、M、L、 SM、F、B、SB、S	T、ST、C、D、 W、SD、SW、R	Z	LC	LZ	K、H	P
(d)	—	—	—	—	—	—	OK

功能

- 可以调用,[待机梯形图]中的子程序(用指针表示子程序名)。
- CALL(P)指令的嵌套最多可达 16 重。
- 子程序如果没有被调用不会被执行。
(子程序都是待机型梯形图,平常都是待机状态不会被执行.只有被 CALL 的时候才能执行)

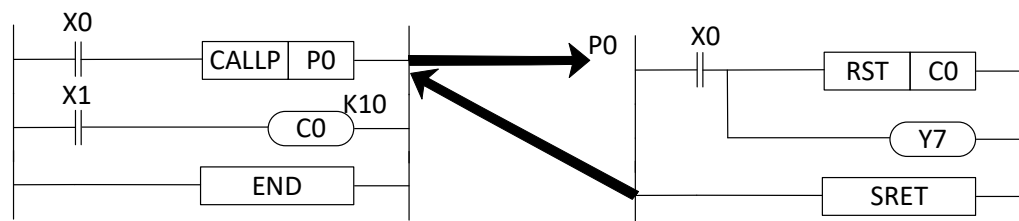
程序动作

梯形图程序如下所示执行动作。

- 子程序可以写在一下梯形图中

- 1) 待机梯形图.

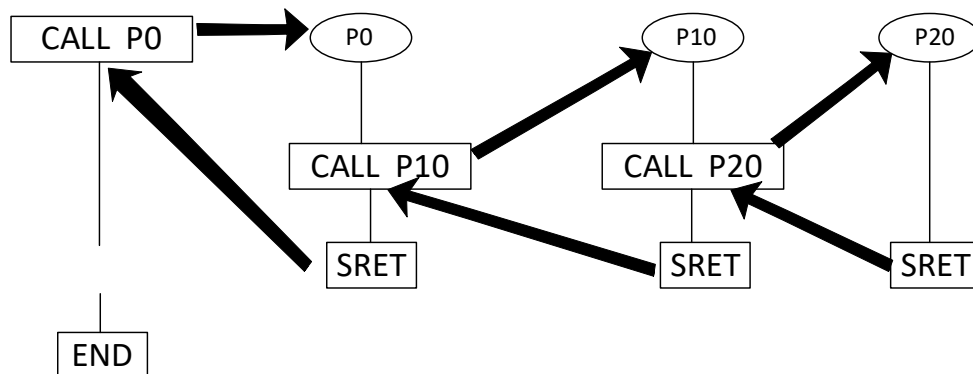
举例 1:如果指令输入为 ON, 将执行 CALL 指令, 跳转至标签(Pn)的步之处。接着执行标签 Pn 的子程序。如果执行 SRET, 将返回至 CALL 指令的下一步之处。



主扫描程序

待机型程序

2.一个程序中 CALL(P)指令的嵌套最多可达 16 重。(所有任务的嵌套加载一起不能超过 32 个.)

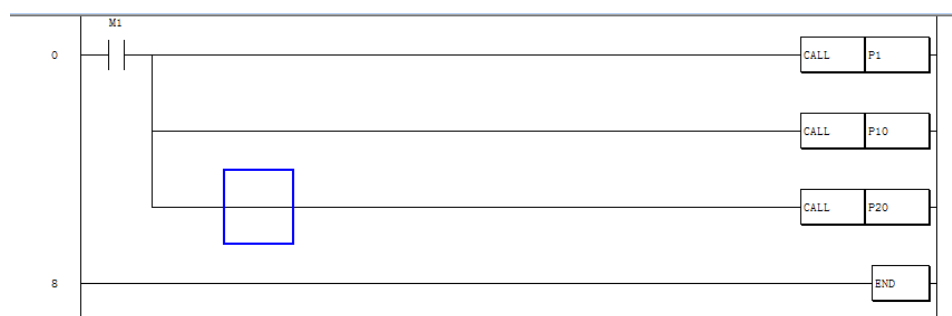


3.子程序遇到 SRET 指令,才结束.

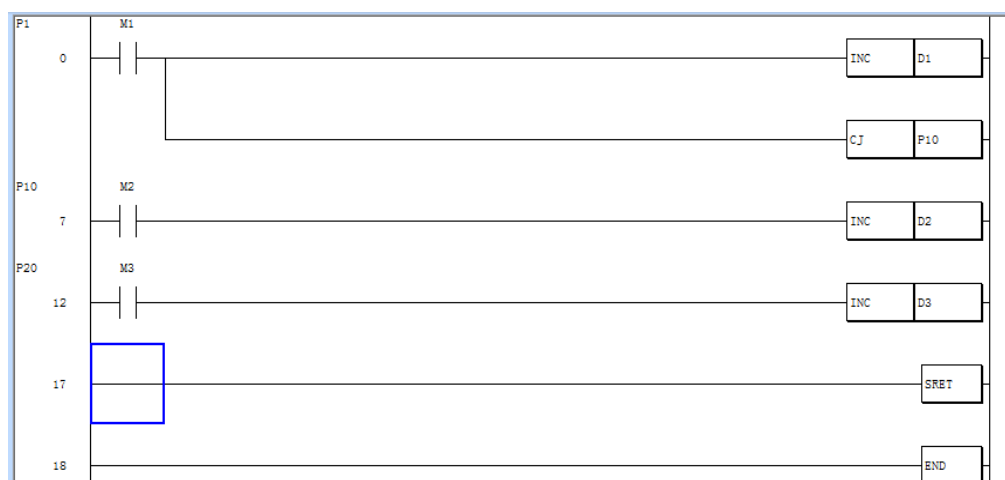
举例:

- 1) 通过指令 [CALL P1.]跳转到待机型程序执行第 0 步~17 步的 SRET 为止.(中间有其他指针也不停止)
- 2) 通过指令 [CALL P10.]跳转到待机型程序执行第 7 步~17 步的 SRET 为止.(中间有其他指针也不停止)
- 3) 通过指令 [CALL P20.]跳转到待机型程序执行第 12 步~17 步的 SRET 为止.

主程序:

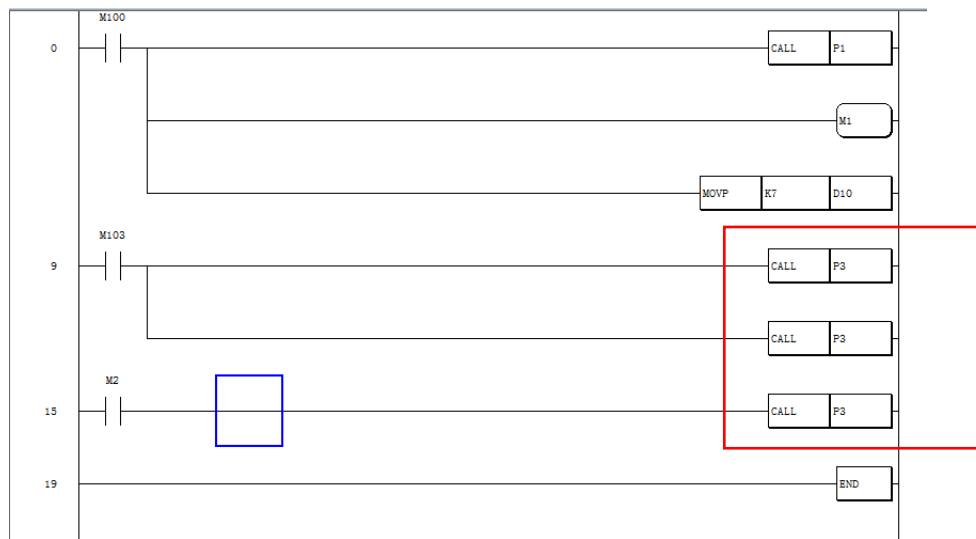


待机型程序:



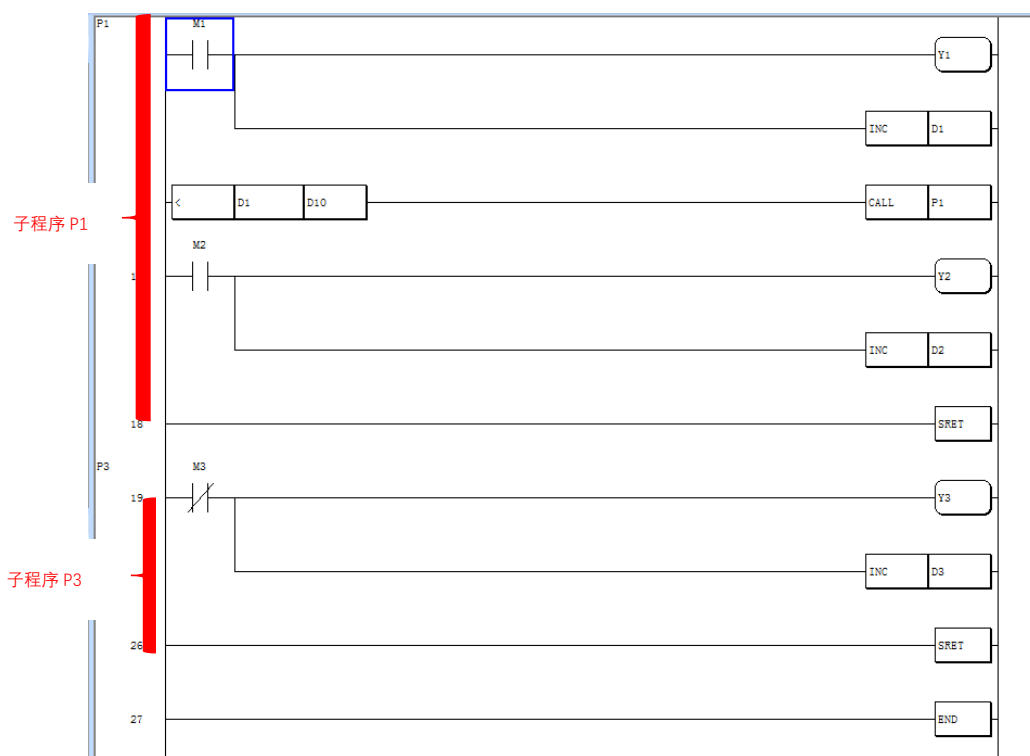
2. 多个 CALL 指令,可以同时调用同一个指针 Pn

举例:多次调用同一个指针 P3 是 OK.



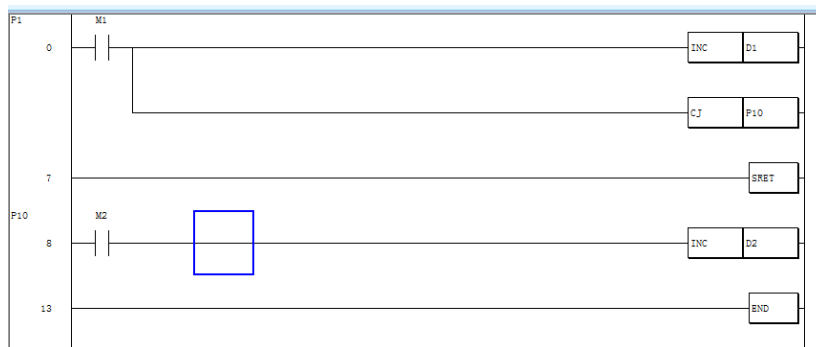
待机型程序

- (一) 待机型程序中可以有多个子程序.(如下图)
- (二) 待机型梯形图中不支持写中断程序.



注意事项

- CALL 指令在操作数(P)中的编号请勿与 CALL 指令以外(CJ 指令)使用的标签(P)的编号重复.
- CJ 指令不能越过子程序的范围.下面图中[CJ P10]指令是错误.



- SRET 指令只能用在待机梯形图中。

出错

监控时的运算出错代码。

出 错 代 码	内 容	错误时动作
8010	一个任务 CALL 指令嵌套次数超过 16.	停止运行
8013	CALL(P)指令中指定的指针的子程序不存在时。	停止运行
8040	待机程序以外的地方使用了 SRET 指令。	停止运行
8041	待机程序中,缺少 SRET 指令.(END 指令前没有 SRET 指令)	停止运行
8042	总任务的 CALL 指令嵌套次数超过 32.	停止运行

4.5.5 CJ (P) 指令

概要

使 CJ、CJP 指令开始到指针(P)为止的顺控程序不执行的指令。

可以缩短循环时间(运算周期)和执行使用双线圈的程序。

指令格式

执行类型	连续执行型
指令格式	CJ S/CJP S

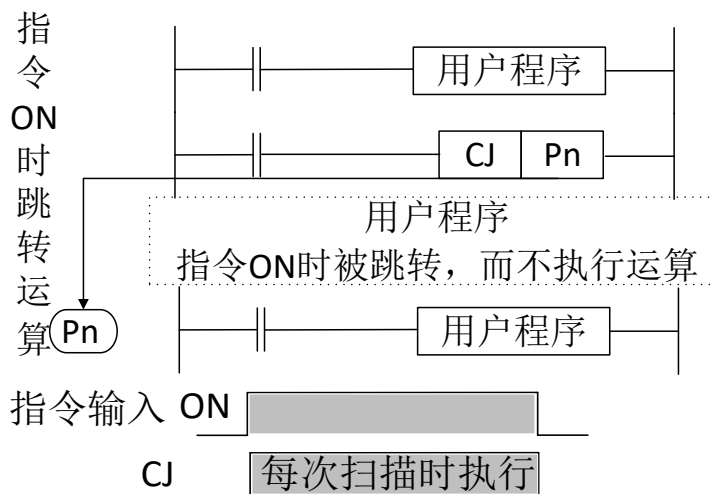
其中操作数 S 仅能使用 P,以及 P 的变址

示例

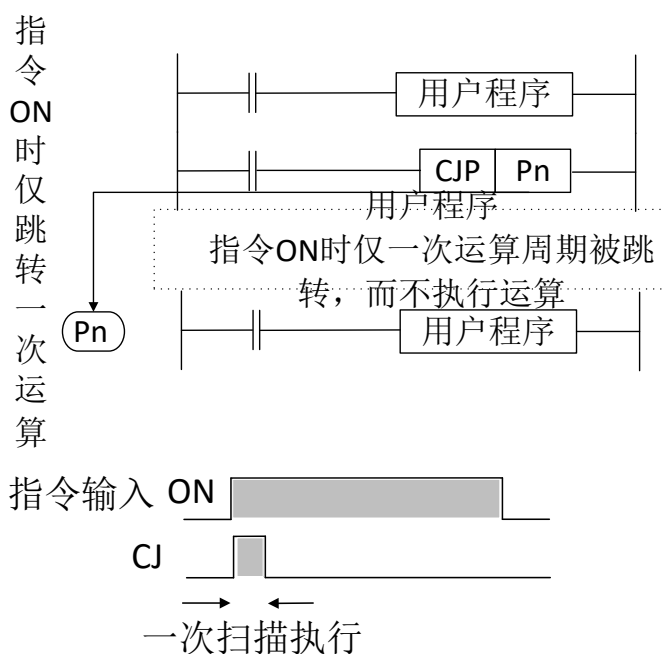
1. 16 位运算(CJ、CJP)

当指令输入为 ON 时，执行指定标记(指针编号)的程序。

1) CJ 指令时



2) CJP 指令时



报错相关：(待添加)

注意：(待添加)

4.5.6 FOR、NEXT 指令

概要

从 FOR 指令开始到 NEXT(FNC 09)指令之间的程序按指定次数重复运行。

指令格式

执行类型	连续执行型
指令格式	FOR S NEXT

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	○	—	—	—	—

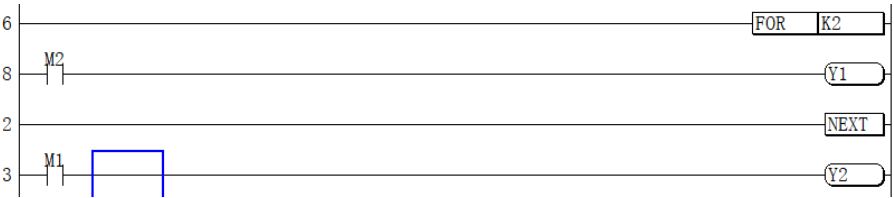
*1

支持变址修饰

示例

FOR ~ NEXT 指令的处理重复 n 次 (源数据中指定的次数 , 1-32767,当 N 指定了 -32768~0 的时候, 进行 N=1 的处理)

重复了指定次数后, 执行 NEXT 指令后的步处理。



当梯形图扫到第六步事, 图中 FOR NEXT 指令的 LD M2,OUT Y1 将会连续执行两次, 接着执行 LD M1 OUT Y2。如果用户想提前跳出 FOR NEXT 之间的指令, 可以使用 CJ 指令跳转。

注: FOR NEXT 必须同时使用, 且允许嵌套, 允许嵌套的最大数量为 5。

报错相关: (待添加)

4.5.7 EI、DI 指令

概要

EI 指令用与中断程序的使能, DI 指令用于中断程序的失能。想要使用 HCPWorks2 的中断程序, 必须在非中断程序中使用 EI 指令使能中断程序运行中断, 而在其后使用 DI 指令后, 梯形图扫到 DI 指令会禁止中断程序。EI, DI 指令是不需要指令(驱动)触点的独立指令。

指令格式

执行类型	连续执行型
指令格式	EI/DI

示例

1.EI 指令

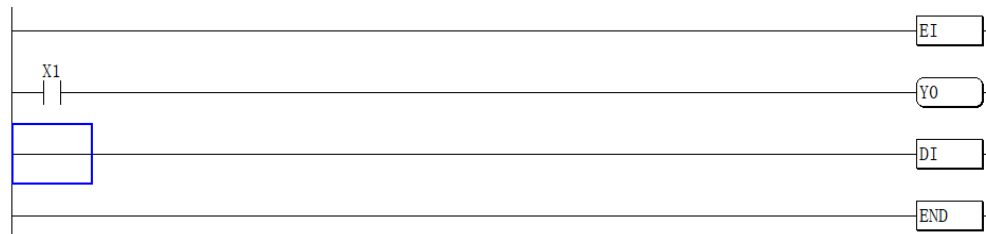
在 HCPWorks2 使用外部输入中断或者高速比较中断时, 必须先使用使能中断功能



EI 指令可写于非中断程序之中。

2. DI 指令

在 HCPWorks2 中，当需要禁止进入中断程序时，可以使用 DI 指令进行中断的失能。



当如图的扫描程序在 PLC 中运行时，扫到 EI 指令时，中断使能，这个时候中断的条件满足时可以进入中断程序。当中断执行完出来后，执行到 DI 指令后，中断程序又被禁止，此时无法进入中断程序。

报错相关：(待添加)

注意：(待添加)

4.5.8. IMASK 中断程序屏蔽

概要

用以将 (S) 中指定的软元件开始的 16 点的位状态，将指定的中断指针编号的中断程序置为允许执行状态或禁止状态

指令格式

IMASK	
16 位指令	IMASK
执行类型	连续执行型
指令格式	MASK S

设置数据

操作数	内容	数据	
		16 位范围	16 位类型
S	存储了中断屏蔽数据的软元件起始编号 (s)中指定的软元件作为起始，用到(s)+3	NA	有符号 16 位二进制数整数

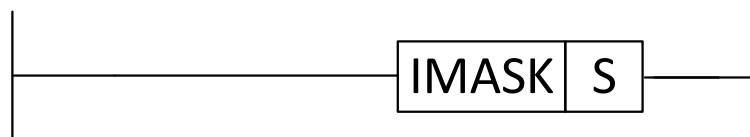
可使用软元件(IMASK)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	○*1	—	—	—	—

*1 支持变址修饰

示例

16 位运算指令 (IMASK)



根据(s)中指定的软元件开始的 16 点的位模式，指定的中断指针编号的中断程序置为执行允许状态或执行禁止状态。

- 1(ON): 中断程序的执行禁止状态
- 0(OFF): 中断程序的执行允许状态
(为了兼容已经使用的老版本程序，默认值 0 为允许)

中断分类与对应特殊软元件

中断类型	中断指针	对应软元件
外部输入中断	I0-I15	SD1400
高速比较中断	I16-I23	SD1401
恒定周期中断	I28-I31	SD1401
预留	I24-I27,I32-I63(预留)	SD1401,SD1402,SD1403

R/W	bit15-----bit0			S~S+3
SD1400	X10-X17(I15-I8)		X0-X7(I17-I0)	S
SD1401	恒定 (I31-I28)	预留	高速比较 (I23-I16)	S+1
SD1402	预留			S+2
SD1403	预留			S+3

注：

- 电源投入时或进行了 CPU 模块复位的情况下，I0 ~ I63 的中断程序将变为执行状态。
- (s) ~ (s)+3 的软元件的状态将被存储到 SD1400 ~ SD1403(IMASK 指令屏蔽模式)中。

错误代码：(s)中指定的软元件开始的 16 点的范围超出相应软元件的范围时。(错误代码：6004)

(s)中的软元件超过软元件范围 (错误代码：6004)

(s) 中的值超过 16 位数据范围时 (错误代码：6706)

4.5.9 SIMASK 指定中断指针的禁止/允许

概要

将(I)中指定的中断指针 No.按照(s)的值, 置为执行允许状态/执行禁止状态

指令格式

SIMASK	
16 位指令	SIMASK
执行类型	连续执行型
指令格式	SIMASK I S

设置数据

操作数	内容	数据	
		16 位范围	16 位类型
I	对中断进行允许/禁止设置的中断指针 No.	I0-I63	软元件名称
S	指定的中断指针 No.的允许/禁止	0: 允许 1: 禁止	有符号 16 位二进制数整数

可使用软元件(SIMASK)

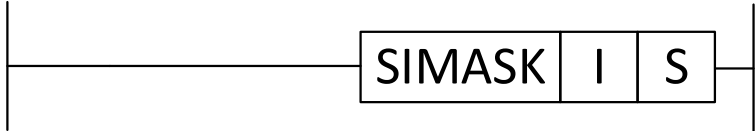
操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
I *1	—	—	—	—	—	—	—	—
S	○*2	○*2						

*1 只支持 I0-I63

*2 支持变址修饰

示例

16 位运算指令 (SIMASK)



- 将(I)中指定的中断指针 No.的中断程序, 按照(s)中指定的数据置为执行允许状态/执行禁止状态。
- (s)为 1 的情况下: 中断程序的执行禁止状态
- (s)为 0 的情况下: 中断程序的执行允许状态 (考虑到兼容老版本程序)

中断类型	中断指针	对应软元件
外部输入中断	I0-I15	SD1400
高速比较中断	I16-I23	SD1401
恒定周期中断	I28-I31	SD1401

预留	I24-I27,I32-In(预留)	SD1401,SD1402,SD1403
----	--------------------	----------------------

注：

- 电源投入时或进行了 CPU 模块复位的情况下，所有的中断指针 No.将变为执行允许状态。
- 中断指针的执行允许状态/执行禁止状态将存储到 IMASK 指令屏蔽模式中。

错误代码：

(s)中的软元件超过软元件范围（错误代码：6004）

(l)中指定的中断指针 No.超过了中断指针 No.的范围(I0 ~ I63)时（错误代码：8031）

4.6 触点比较指令

4.6.1 LD=、>、<、<>、<=、>=/触点比较

概要

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

指令格式

LD=			
16 位指令	LD=	32 位指令	LDD=
执行类型	连续执行型		
指令格式	LD= S1 S2	LDD= S1 S2	

LD>			
16 位指令	LD>	32 位指令	LDD>
执行类型	连续执行型		
指令格式	LD>S1 S2	LDD> S1 S2	

LD<			
16 位指令	LD<	32 位指令	LDD<
执行类型	连续执行型		
指令格式	LD<S1 S2	LDD<S1 S2	

LD<>			
16 位指令	LD<>	32 位指令	LDD<>
执行类型	连续执行型		
指令格式	LD<>S1 S2	LDD<> S1 S2	

LD<=			
16 位指令	LD<=	32 位指令	LDD<=
执行类型	连续执行型		
指令格式	LD<=S1 S2	LDD<= S1 S2	

LD>=			
16 位指令	LD>=	32 位指令	LDD>=
执行类型	连续执行型		
指令格式	LD>= S1 S2	LDD>= S1 S2	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
S1	—	—	○*3	○*1*2	○	—	—	○ —
S2	—	—	○*3	○*1*2	○	—	—	○ —

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnX、KnY、KnM、KnS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存比较数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
S2	保存比较数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

示例

功能和动作说明

连接在母线上的触点比较指令。

对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

16 位指令	32 位指令	导通条件	不导通条件
LD=	LDD=	S1=S2	S1≠S2
LD>	LDD>	S1>S2	S1≤S2
LD<	LDD<	S1<S2	S1≥S2
LD<>	LDD<>	S1≠S2	S1=S2
LD≤	LDD≤	S1≤S2	S1>S2
LD≥	LDD≥	S1≥S2	S1<S2

报错相关：(待添加)

注意：

1. 关于负数

S1、S2 的数据最高位为 1 时，将其值当作负数进行比较。

• 16 位运算时: b15

• 32 位运算时: b31

2. 使用 32 位计数器(包含高速计数器)时

32 位计数器(C200~C255)的比较必须用 32 位(LDD=、LDD >、LDD < 等)运算执行。
如指定了 16 位运算(LD=、LD >、LD < 等)时, 会发生程序错误或是运算错误。

3. 触点比较指令的编程

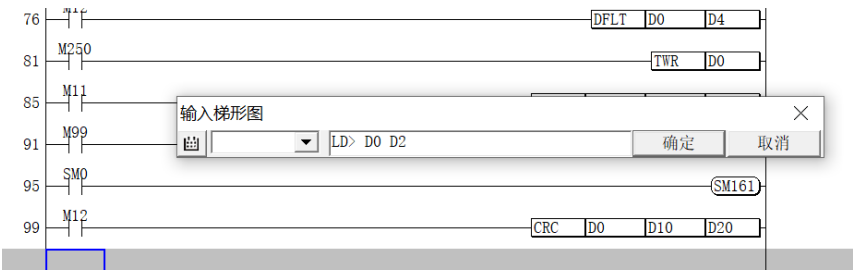
使用 HCP Works2 编程时, 不能输入 ≤ 和 ≥ 符号。

≤ 是分成 < 和 =, ≥ 是分成 > 和 = 进行输入的。

输入操作如下所示。

操作步骤

- a) 在[显示梯形图程序的编辑画面]中, 将光标停在要使用触点比较指令的位置。
- b) 输入[指令]→空格→[比较值或是软元件]→空格→[比较值或是软元件]。关于输入的例子, 请参考下面的
[HCP Works2 的指令输入画面]。
- c) 点击[确定]键。
- d) 接着, 请继续输入其他的触点以及线圈部分。



4.6.2 AND=、>、<、< >、<=、>=/触点比较

概要

执行数值的比较, 当条件满足时使触点置 ON 的触点比较运算指令。

指令格式

AND=			
16 位指令	AND=	32 位指令	ANDD=
执行类型	连续执行型		
指令格式	AND= S1 S2	ANDD= S1 S2	

AND >			
16 位指令	AND>	32 位指令	ANDD>
执行类型	连续执行型		
指令格式	AND> S1 S2	ANDD> S1 S2	

AND<			
16 位指令	AND<	32 位指令	ANDD<
执行类型	连续执行型		
指令格式	AND< S1 S2	ANDD< S1 S2	

AND<>			
16 位指令	AND<>	32 位指令	ANDD<>
执行类型	连续执行型		

指令格式	AND<>S1 S2	ANDD<> S1 S2
------	------------	--------------

AND<=			
16 位指令	AND<=	32 位指令	ANDD<=
执行类型	连续执行型		
指令格式	AND<=S1 S2	ANDD<= S1 S2	

AND>=			
16 位指令	AND>=	32 位指令	ANDD>=
执行类型	连续执行型		
指令格式	AND>= S1 S2	ANDD>= S1 S2	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
S1	—	—	○*3	○*1*2	○	—	—	○ —
S2	—	—	○*3	○*1*2	○	—	—	○ —

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnX、KnY、KnM、KnS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存比较数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
S2	保存比较数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

示例

与其他触点串联的触点比较指令。

对 S1.S2 的内容进行 BIN 比较，根据其结果控制触点的导通或是不导通。

16 位指令	32 位指令	导通条件	不导通条件
AND=	ANDD=	S1=S2	S1≠S2
AND>	ANDD>	S1>S2	S1≤S2
AND<	ANDD<	S1<S2	S1≥S2
AND<>	ANDD<>	S1≠S2	S1=S2
AND≤	ANDD≤	S1≤S2	S1>S2
AND≥	ANDD≥	S1≥S2	S1<S2

报错相关：(待添加)

注意：

1. 关于负数

S1、S2 的数据最高位为 1 时，将其值当作负数进行比较。

- 16 位运算时:b15
- 32 位运算时:b31

2. 使用 32 位计数器(包含高速计数器)时,32 位计数器(C200 ~ 255)的比较必须用 32 位 (ANDD=、ANDD>、ANDD<等)运算执行。如指定了 16 位运算 (AND=、AND>、AND<等)时，会发生程序错误或是运算错误。

3. 触点比较指令的编程

使用 HCP Works2 编程时，不能输入 $\leq$ 和 $\geq$ 符号。

$\leq$ 是分成<和=， $\geq$ 是分成>和=进行输入的。

输入操作如下所示。

操作步骤

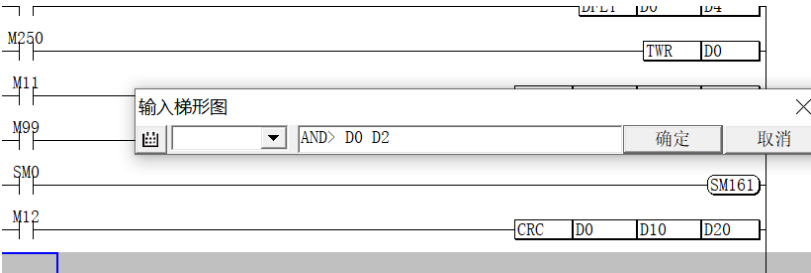
- 在[显示梯形图程序的编辑画面]中，将光标停在要使用触点比较指令的位置。
- 输入[指令]→空格→[比较值或是软元件]→空格→[比较值或是软元件]。关于输入的例子，请参考下面的

[HCP Works2 的指令输入画面]。

- 点击[确定]键。

- 接着，请继续输入其他的触点以及线圈部分。

HCP Works2 的指令输入画面



4.6.3 OR=、>、<、< >、<=、>=/触点比较

概要

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算指令。

指令格式

OR=			
16 位指令	OR=	32 位指令	ORD=
执行类型	连续执行型		
指令格式	OR= S1 S2	ORD= S1 S2	

OR >			
16 位指令	OR >	32 位指令	ORD >
执行类型	连续执行型		
指令格式	OR > S1 S2	ORD > S1 S2	

OR<			
16 位指令	OR<	32 位指令	ORD<
执行类型	连续执行型		
指令格式	OR<S1 S2	ORD<S1 S2	

OR<>			
16 位指令	OR<>	32 位指令	ORD<>
执行类型	连续执行型		
指令格式	OR<>S1 S2	ORD<> S1 S2	

OR<=			
16 位指令	OR<=	32 位指令	ORD<=
执行类型	连续执行型		
指令格式	OR<=S1 S2	ORD<= S1 S2	

OR>=			
16 位指令	OR>=	32 位指令	ORD>=
执行类型	连续执行型		
指令格式	OR>= S1 S2	ORD>= S1 S2	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
S1	—	—	○*3	○*1*2	○	—	—	○
S2	—	—	○*3	○*1*2	○	—	—	○

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnX、KnY、KnM、KnS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存比较数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
S2	保存比较数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

示例

与其他触点并联的触点比较指令。

对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

16 位指令	32 位指令	导通条件	不导通条件
OR=	ORD=	S1=S2	S1≠S2

OR>	ORD>	S1>S2	S1≤S2
OR<	ORD<	S1<S2	S1≥S2
OR<>	ORD<>	S1≠S2	S1=S2
OR≤	ORD≤	S1≤S2	S1>S2
OR≥	ORD≥	S1≥S2	S1<S2

报错相关：(待添加)

注意:

1. 关于负数

S1、S2 的数据最高位为 1 时，将其值作为负数进行比较。

- 16 位运算时:b15

- 32 位运算时:b31

2. 使用 32 位计数器(包含高速计数器)时,32 位计数器(C200 ~ C255)的比较必须用 32 位(ORD=、ORD >、ORD < 等)运算执行。如指定了 16 位运算(ORD=、OR >、OR < 等)时会发生程序错误或是运算错误。

3. 触点比较指令的编程

使用 HCP Works2 编程时，不能输入 ≤ 和 ≥ 符号。

≤ 是分成 < 和 =， ≥ 是分成 > 和 = 进行输入的。

输入操作如下所示。

操作步骤

a) 在[显示梯形图程序的编辑画面]中，将光标停在要使用触点比较指令的位置。

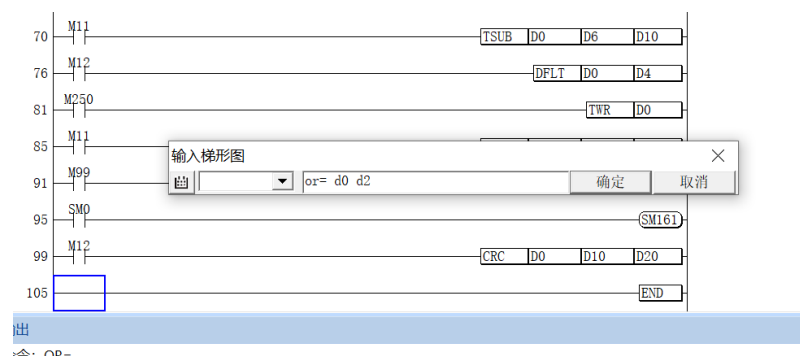
b) 输入[指令]→空格→[比较值或是软元件]→空格→[比较值或是软元件]。关于输入的例子，请参考下面的

[HCP Works2 的指令输入画面]。

c) 点击[OK]键。

d) 接着，请继续输入其他的触点以及线圈部分。

HCP Works2 的指令输入画面



4.7 比较输出

4.7.1 CMP 指令

概要

比较 2 个值，将其结果(大、一致、小)输出到位软元件中(3 点)。

指令格式

16 位指令	CMP	32 位指令	DCMP
执行类型	连续执行型、脉冲执行型		
指令格式	CMP(P) S1 S2 D	DCMP(P) S1 S2 D	

可使用软元件

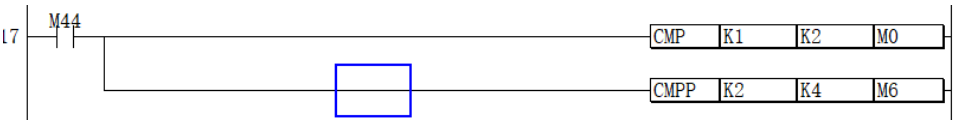
操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
S1	—	—	○	○*1	○	—	—	○
S2	—	—	○	○*1	○	—	—	○
D	○*1	○	—	—	—	—	—	—

*1

支持变址修饰

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	储存要比较的数据的软元件编号或数据	- 32768~32767	- 2147483648~2147483647	实数	实数
S2	储存要比较的数据的软元件编号或数据	- 32768~32767	- 2147483648~2147483647	实数	实数
D	输出结果的起始位软元件编号(占用 3 点)	-	1,0(占用 3 点)	-	位

示例



如图上所示，当 M44 闭合后，CMP 指令会一直执行，操作数 S1>S2 时，M0 会置 ON。

S1=S2.M1 置 ON, S1<S2 时则 M2 置 ON。故此图中为 M2 置 ON。
而下面的 CMPP 指令, 仅在 M44 闭合的瞬间执行一次, 由于 K2<K4,故 M8 会被置 ON。

报错相关: (待添加)
注意: (待添加)

4.7.2 ZCP 指令

概要

针对 2 个值(区间), 将与比较源的值比较得出的结果(小于、等于 (区域内)、大于)输出到位软元件(3 点)中。

指令格式

16 位指令	ZCP(P)	32 位指令	D ZCP(P)
执行类型	连续执行型、脉冲执行型		
指令格式	ZCP(P) S1 S2 S D	DZCP(P) S1 S2 S D	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	○	○*1	○	—	—	○	—	
S2	—	—	○	○*1	—	—	—	○	—	
S	—	—	○	○*1	—	—	—	○	—	
D	○*1	○	—	—	—	—	—	—	—	

*1 支持变址修饰

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	储存要比较的数据的软元件编号或数据	-32768~32767	- 2147483648~2147483647	实数	实数
S2	储存要比较的数据的软元件编号或数据	-32768~32767	- 2147483648~2147483647	实数	实数
S	储存要比较的数据的软元件编号或数据	-32768~32767	- 2147483648~2147483647	实数	实数

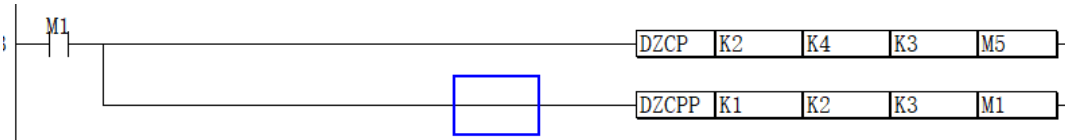
	据				
D	输出结果的起始位 软元件编号(占用3点)	-	1,0(占用3点)	-	位

示例

1.ZCP 与 ZCPP

将比较源的内容与下比较值和上比较值进行比较，根据其结果(小、区域内、大)，使、+1、+2 其中一个为 ON。

- 按代数形式进行大小的比较。例如: $-10 < 2 < 10$



如图中所示，当 M1 闭合后，DZCP 指令会一直执行。如果 $S1 > S$ 时，D 会置 on.当 $S1 \leq S \leq S2$ 时，D+1 或置 ON.当 $S > S2$ 时，D+2 会置 ON。图上 $S1 < S < S2$.故 M6 置 ON，S 在区间内

DZCPP 指令，会在 M1 闭合后执行一次，图上 $S1 < S2 < S$.故 M3 置 ON，S 大于区间内的最大值，不在区间内。

报错相关：(待添加)

注意：(待添加)

4.7.3 BKCMP=、>、<、<>、<=、>=/数据块比较

概要

这些指令是按照各个指令的比较条件来比较数据块。

指令格式

BKCMP=			
16 位指令	BKCMP(P) =	32 位指令	DBKCMP (P) =
执行类型	连续执行型、脉冲执行型		
指令格式	BKCMP (P) = S1 S2 D N	DBKCMP(P) = S1 S2 D N	

BKCMP>			
16 位指令	BKCMP(P)>	32 位指令	DBKCMP (P) >
执行类型	连续执行型、脉冲执行型		
指令格式	BKCMP (P) > S1 S2 D N	DBKCMP(P) > S1 S2 D N	

BKCMP<			
16 位指令	BKCMP(P)<	32 位指令	DBKCMP (P) <
执行类型	连续执行型、脉冲执行型		
指令格式	BKCMP (P) < S1 S2 D N	DBKCMP(P) < S1 S2 D N	

BKCMP<>			
---------	--	--	--

16 位指令	BKCOMP(P) <>	32 位指令	DBKCOMP (P) <>
执行类型	连续执行型、脉冲执行型		
指令格式	BKCOMP (P) <> S1 S2 D N	DBKCOMP(P) <> S1 S2 D N	

BKCOMP<=			
16 位指令	BKCOMP(P) <=	32 位指令	DBKCOMP (P) <=
执行类型	连续执行型、脉冲执行型		
指令格式	BKCOMP (P) <= S1 S2 D N	DBKCOMP(P) S1 S2 D N	

BKCOMP>=			
16 位指令	BKCOMP(P) >=	32 位指令	DBKCOMP (P) >=
执行类型	连续执行型、脉冲执行型		
指令格式	BKCOMP (P) >= S1 S2 D N	DBKCOMP(P) >= S1 S2 D N	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	○	—
				○*1	—	—	—	—	
				○*1	○	—	—	○	

*1 支持变址修饰

可使用操作数

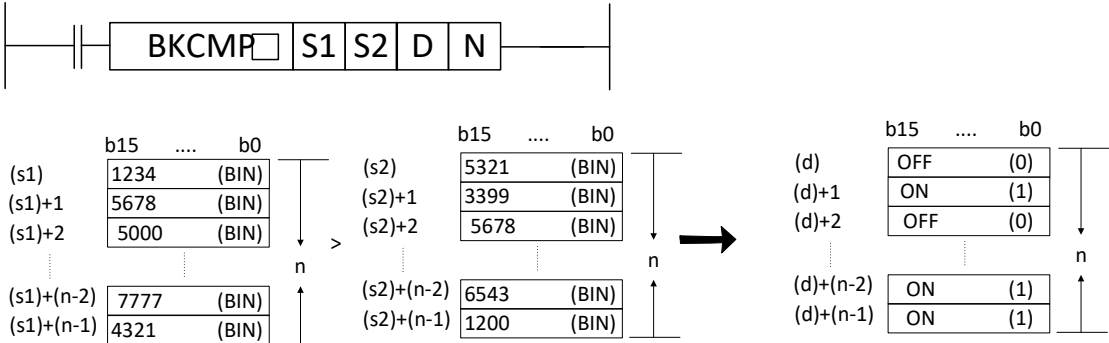
操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	存 储 了 加 法 运 算 数 据 的 起 始 软 元 件	-32768~32767	-2147483648~2147483647	实数	实数
S2	加 法 运 算 数 据 或 存 储 了 加 法 运 算 数 据 的 起 始 软 元 件	-32768~32767	-2147483648~2147483647	实数	实数
D	存 储 运 算 结 果 的 起 始 软 元 件	0~32767	0~32767	实数	实数
N	加 法 运 算 数 据 个 数	0~32767	0~32767	实数	实数

示例

1. 16 位运算(BKCMPP=、>、<、<>、<=、>=/BKCMPP=P、>P、<P、<>P、<=P、>=P)

1) • 将(s1)中指定的软元件开始的(n)点的 BIN16 位数据与(s2)中指定的软元件开始的(n)点的 BIN16 位数据进行比较, 将比较结果存储到(d)中指定的软元件开始的(n)点中。

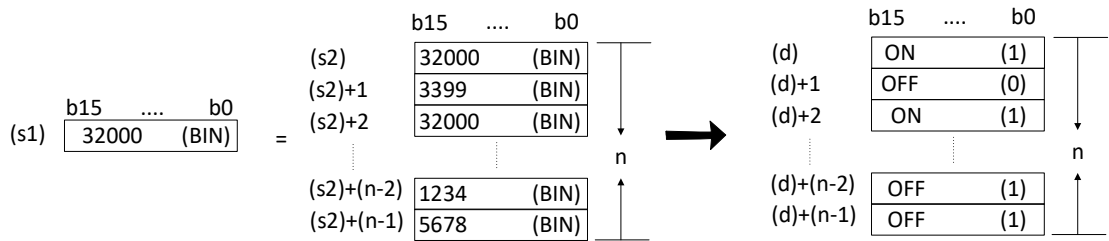
• (d)中指定的软元件开始的(n)点的相应软元件在比较条件成立时变为 ON、不成立时变为 OFF。



2)

• 比较运算是以 16 位单位进行。

• (s1)可以指定直接常数。



3) 各个指令的比较结果如下所示。

指令	比较结果 ON (1) 的条件	比较结果 OFF (0) 的条件
BKCMPP=	$S1=S2$	$S1 \neq S2$
BKCMPP>	$S1>S2$	$S1 \leq S2$
BKCMPP<	$S1<S2$	$S1 \geq S2$
BKCMPP<>	$S1 \neq S2$	$S1=S2$
BKCMPP<=	$S1 \leq S2$	$S1>S2$
BKCMPP>=	$S1 \geq S2$	$S1<S2$

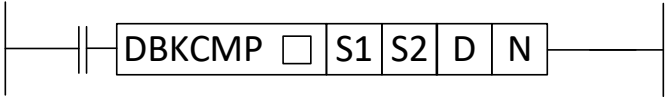
4) D 开始的 n 点的比较结果都为 ON(1)时, SM90(块比较信号)为 ON。

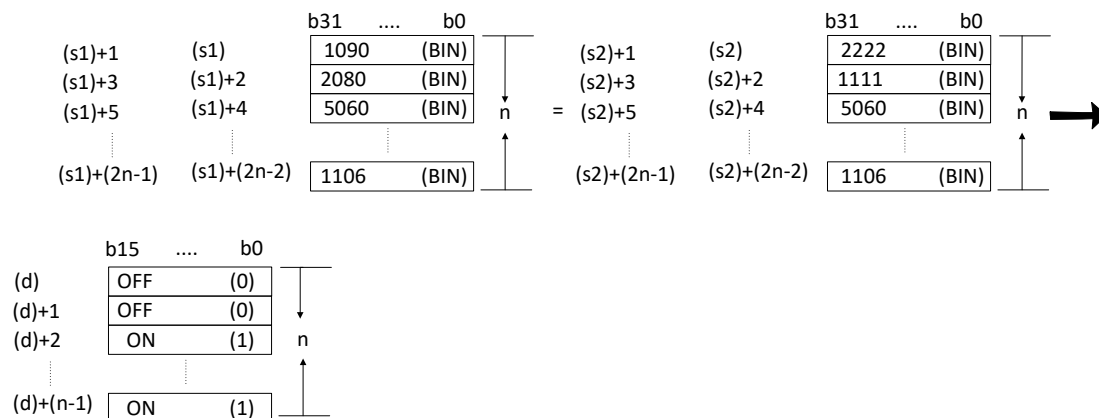
2. 32 位运算(DBKCMPP=、>、<、<>、<=、>=/DBKCMPP=P、>P、<P、<>P、<=P、>=P)

1)

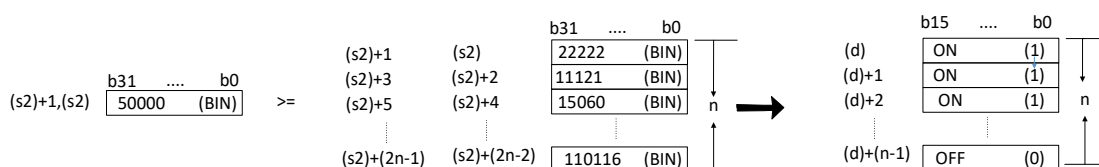
• 将(s1)中指定的软元件开始的(n)点的 BIN32 位数据与(s2)中指定的软元件开始的(n)点的 BIN32 位数据进行比较, 将比较结果存储到(d)中指定的软元件开始的(n)点中。

• (d)中指定的软元件开始的(n)点的相应软元件在比较条件成立时变为 ON、不成立时变为 OFF。





2) 可以在[S1+1 , S1]中直接指定常数



3) 各个指令的比较结果如下所示。

指令	比较结果 ON (1) 的条件	比较结果 OFF (0) 的条件
BKCMPE	$S1+1, S1 = S2+1, S2$	$S1+1, S1 \neq S2+1, S2$
BKCMPE>	$S1+1, S1 > S2+1, S2$	$S1+1, S1 \leq S2+1, S2$
BKCMPE<	$S1+1, S1 < S2+1, S2$	$S1+1, S1 \geq S2+1, S2$
BKCMPE<>	$S1+1, S1 \neq S2+1, S2$	$S1+1, S1 = S2+1, S2$
BKCMPE<=	$S1+1, S1 \leq S2+1, S2$	$S1+1, S1 > S2+1, S2$
BKCMPE>=	$S1+1, S1 \geq S2+1, S2$	$S1+1, S1 < S2+1, S2$

4) 开始的 n 点的比较结果都为 ON(1)时, SM90(块比较信号)为 ON。

相关软元件

指令	名称	内容
SM90	块比较信号	数据块指令的比较结果都为 ON(1)时变为 ON。 DBKCMPE=(FNC 194)、DBKCMPE>=(FNC 195)、DBKCMPE<=(FNC 196)、 DBKCMPE<>=(FNC 197) 、 DBKCMPE<==(FNC 198) 、 DBKCMPE>=(FNC 199)

报错相关:

以下一些情况下会发生运算错误, 错误标志位 SM67 置 ON, 错误代码保存在 SD67 中。

- S1、S2 开始的 n 点(32 位运算时为 2n 点)软元件超出相应的软元件范围时。

(错误代码: 由开发人员定义)

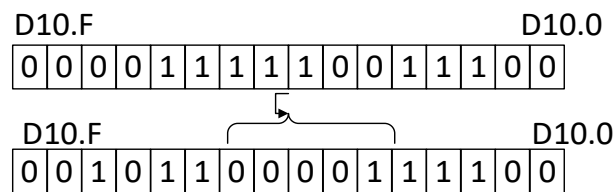
- 开始的 n 点软元件超出了相应软元件的范围时。(错误代码: 由开发人员定义)

- D 指定为“D.X”时, D 的数据寄存器和 S1 开始的 n 点 (32 位运算时为 2n 点) 软元件范围重复

时。(错误代码: 由开发人员定义)

- D 指定为“D.X”时, D 的数据寄存器和 S1 开始的 n 点 (32 位运算时为 2n 点) 软元件范围重复

进行了字软元件的位指定的情况下，存储运算结果的位指定软元件以外不变化。



4.8.1 ADD/BIN 加法运算

2 个值进行加法运算($A+B=C$)后得出结果的指令。

16 位指令	ADD(P)	32 位指令	DADD (P)
执行类型	连续执行型、脉冲执行型		
指令格式	ADD (P) S1 S2 D	DADD(P) S1 S2 D	

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	○	○*1	○	—	—	○	—	
S2	—	—	○	○*1	○	—	—	○	—	
D	—	—	○*2	○*1	○	—	—	—	—	

*2 不支持 KnX

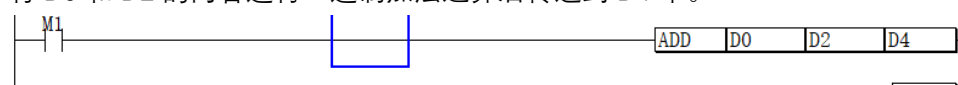
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	加法运算的数据，或是保存数据的软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数

S2	加法运算的数据，或是保存数据的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数
D	保存加法运算结果的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(ADD、ADDP)

将 D0 和 D2 的内容进行二进制加法运算后传送到 D4 中。



- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行加法运算。

$$5+(-8)=-3$$

- 和中指定常数(K)时，会自动进行 BIN 转换。

2. 32 位运算(DADD,DADDP)

将[D3, D2]和[D1, D0]的内容进行二进制加法运算后传送到[D5, D4]中。



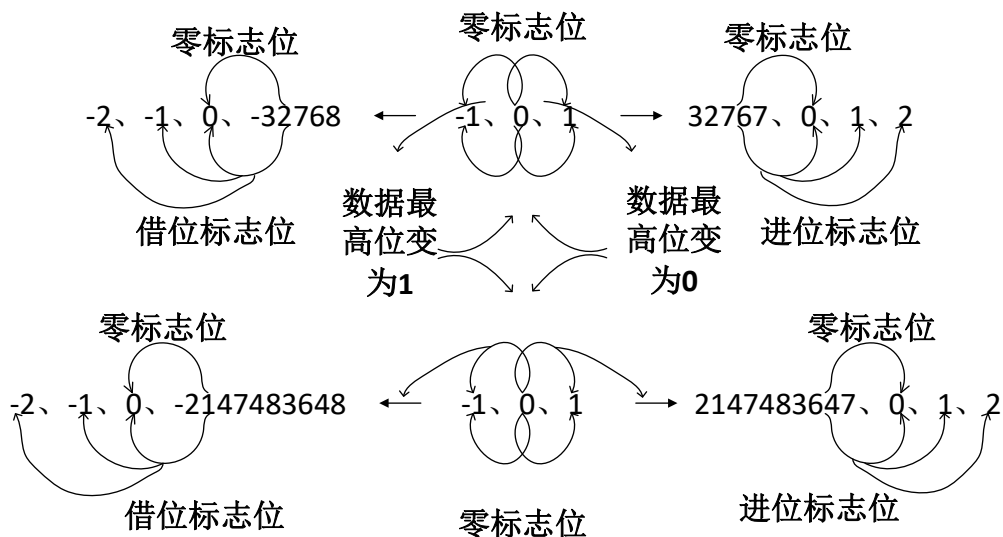
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行加法运算。

$$5,500+(-8,540)=-3,040$$

- [+1,]和[+1,]中指定常数(K)时，会自动进行 BIN 转换。

相关软元件

软元件	名称	内容
SM20	零位	ON: 运算结果为 0 时 OFF: 运算结果为 0 以外时
SM21	借位	ON: 运算结果小于 -32,768(16 位运算)或是 -2,147,483,648(32 位运算)时，借位标志位动作。 OFF: 运算结果不小于 -32,768(16 位运算)或是 -2,147,483,648(32 位运算)时
SM22	进位	ON: 运算结果大于 32,767(16 位运算) 或者 2,147,483,647(32 位运算) 时，进位标志位动作。 OFF: 运算结果不大于 32,767(16 位运算)或者 2,147,483,647(32 位运算)时



报错相关：待添加

注意：待添加

4.8.2 SUB/BIN 减法运算

概要

2 个值进行减法运算(A - B=C)后得出结果的指令。

指令格式

16 位指令	SUB(P)	32 位指令	DSUB (P)
执行类型	连续执行型、脉冲执行型		
指令格式	SUB (P) S1 S2 D		DSUB(P) S1 S2 D

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1	○	—	—	○	—
S2	—	—	○	○*1	○	—	—	○	—
D	—	—	○*2	○*1	○	—	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围	数据类型
-----	----	------	------

		16 位范围	32 位范围	16 位	32 位
S1	减法运算的数据，或是保存数据的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数
S2	减法运算的数据，或是保存数据的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数
D	保存减法运算结果的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(SUB、SUBP)

将 D0 和 D2 的内容进行二进制减法运算后传送到 D4 中。



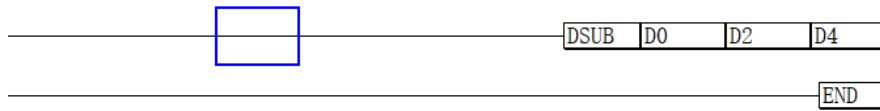
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行减法运算。

(5-(-8)=13)

- S1 和 S2 中指定常数(K)时，会自动进行 BIN 转换。

2. 32 位运算(DSUB,DSUBP)

将[D1,D0]和[D3, D2]的内容进行二进制减法运算后传送到[D5, D4]中。



- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行减法运算。

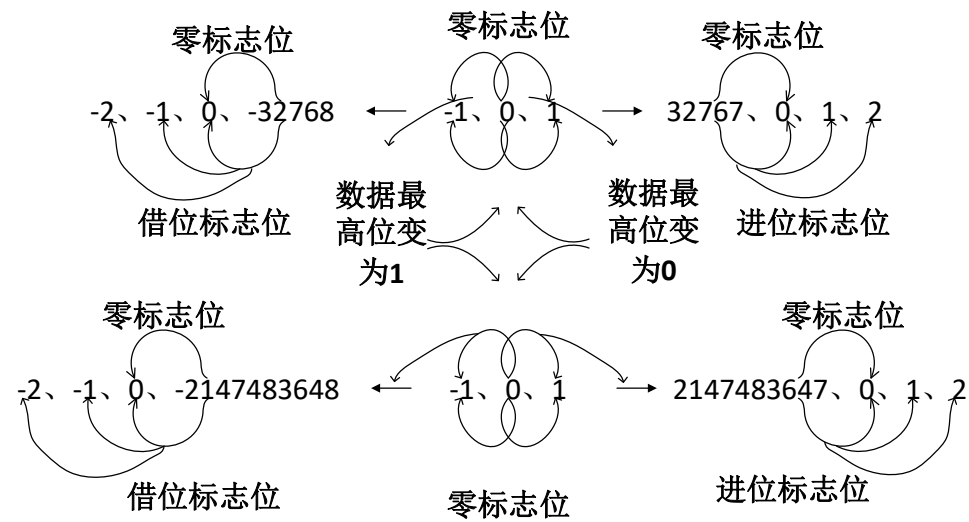
(5,500-(-8,540)=14,040)

- [S1+1, S1]和[S2+1, S2]中指定常数(K)时，会自动进行 BIN 转换。

相关软元件

软元件	名称	内容
SM20	零位	ON: 运算结果为 0 时 OFF: 运算结果为 0 以外时
SM21	借位	ON: 运算结果小于 -32,768(16 位运算)或是 -2,147,483,648(32 位运算)时，借位标志位动作。 OFF: 运算结果不小于 -32,768(16 位运算)或是 -2,147,483,648(32 位运算)时
SM22	进位	ON: 运算结果大于 32,767(16 位运算) 或者 2,147,483,647(32 位运算) 时，进位标志位动作。 OFF: 运算结果不大于 32,767(16 位运算)或者 2,147,483,647(32 位运算) 时

		运算)时
--	--	------



报错相关：待添加
注意：待添加

4.8.3 MUL/BIN 乘法运算

概要

2 个值进行乘法运算($A \times B = C$)后得出结果的指令。

指令格式

16 位指令	MUL(P)	32 位指令	DMUL (P)
执行类型	连续执行型、脉冲执行型		
指令格式	MUL (P) S1 S2 D	DMUL (P) S1 S2 D	

可使用软元件

	位	字	双字	常数
操作数	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S Z、LC、LZ、K、H、E
S1	—	—	○	○*1
S2	—	—	○	○*1
D	—	—	○	○*1

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围	数据类型
		16 位范围	32 位范围
		16 位	32 位

S1	乘法运算的数据，或是保存数据的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数
S2	乘法运算的数据，或是保存数据的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数
D	保存乘法运算结果的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(MUL,MULP)

将 D0 和 D2 的内容进行二进制乘法运算后传送到[+1,]的 32 位(双字)中。



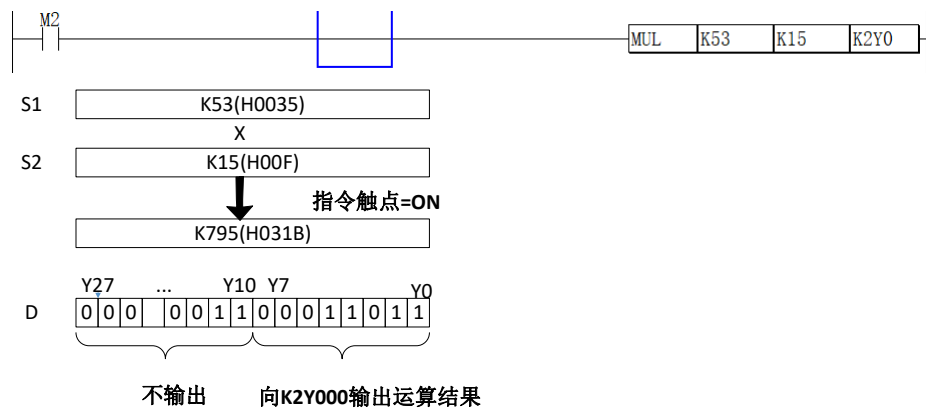
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行乘法运算。

$$5 \times (-8) = -40$$

- S1 和 S2 中指定常数(K)时，会自动进行 BIN 转换。
- [D+1, D]指定位数(K1 ~ 8)的情况

可以指定 K1 ~ K8 的位数。

例如，指定 K2 时，只能得到乘积(32 位)中的低 8 位



2. 32 位运算(DMUL,DMULP)

将 [D1, D0] 和 [D3, D2] 的内容进行二进制乘法运算后传送到 [D7, D6, D5, D4] 的 64 位(字软元件×4)中。

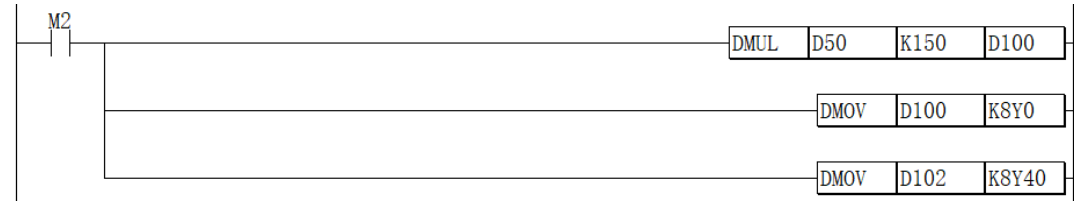


- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行乘法运算。

$$5,500 \times (-8,540) = -46,970,000$$

- [S1+1, S1]和[S2 +1, S2]中指定常数(K)时，会自动进行 BIN 转换。
- [D +3, D +2, D +1, D]中指定位数(K1 ~ 8)时，

只能获得低 32 位的结果，得不到高 32 位的结果。
 请先传送一次到字软元件中后，再执行运算。



相关软元件

软元件	名称	内容
SM304	零位	ON: 运算结果为 0 时 OFF: 运算结果为 0 以外时

报错相关：待添加
 注意：待添加

4.8.4 DIV/BIN 除法运算

概要
 2 个值进行除法运算 $A \div B = C \cdots (\text{余数})$ 后得出结果的指令。

指令格式

16 位指令	DIV(P)	32 位指令	DDIV (P)
执行类型	连续执行型、脉冲执行型		
指令格式	DIV (P) S1 S2 D	DDIV (P) S1 S2 D	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、KnSB、KnS	T、ST、C、D、W、SD、S W、R	Z	LC	LZ	K、H
S1	—	—	○	○*1	○	—	—	○
S2	—	—	○	○*1	○	—	—	○
D	—	—	○*2	○*1	○	—	—	—

*1 支持变址修饰
 *2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	除法运算的数据，或是保存	- 32768~32767	- 2147483648~2147483647	实数	实数

	数据的字 软元件编 号				
S2	除法运算 的数据， 或是保存 数据的字 软元件编 号	- 32768~32767	- 2147483648~2147483647	实数	实数
D	保存除法 运算结果 的字软元 件编号	- 32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(DIV、DIVP)

D0 的内容作为被除数，D2 的内容作为除数，商传到 D4 中，余数传到 D5 中。



- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行除法运算。(36÷(-5)=-7(商)，1(余数))
- 运算结果(商，余数)，会占用指定开始合计 2 点的软元件，所以请注意不能与其他控制重复。
- 和中指定常数(K)时，会自动进行 BIN 转换。

2. 32 位运算(DDIV、DDIVP)

[D1,D0]的内容作为被除数，[D3,D2]的内容作为除数，商传到[D5,D4]，余数传到[D7,D6]中。



- 运算结果(商，余数)，会占用指定开始合计 4 点的软元件，所以请注意不能与其他控制重复。
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行除法运算。(5,500÷(-540)= -10(商)，100(余数))。
- [S1+1, S1]和[S2+1, S2]中指定常数(K)时，会自动进行 BIN 转换。

相关软元件

软元件	名称	内容
SM304	零位	ON: 运算结果为 0 时 OFF: 运算结果为 0 以外时
SM306	进位	ON : 运算结果超过 32,767(16 位运算)或者 2,147,483,647(32 位运算)时，进位标志位动作。 OFF: 运算结果为 32,767(16 位运算)或者 2,147,483,647(32 位运算)以下时

报错相关：待添加

注意：待添加

4.8.5 INC/BIN 加一

概要

指定的软元件数据中加“1”(+1 加法)的指令。

指令格式

16 位指令	INC(P)	32 位指令	DINC (P)
执行类型	连续执行型、脉冲执行型		
指令格式	INC (P) D	DINC (P) D	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
D	—	—	○*2	○*1	○	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	保存被加 一数据的 字软元件 编号	— 32768~32767	— 2147483648~2147483647	实数	实数

示例

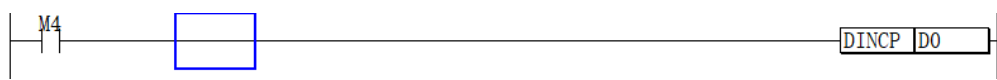
1. 16 位运算(INC、INCP)

D0 的内容加一运算后，传送到 D0 中。



2. 32 位运算(DINC、DINCP)

[D1, D0]的内容加一运算后，传送到[D1, D0]中。



报错相关：待添加

注意：

1) 16 位运算

+32,767 上加 1 后，变为-32,768，但是标志位(零、借位、进位)不动作。

2) 32 位运算

+2,147,483,647 上加 1 后，变为-2,147,483,648，但是标志位(零、借位、进位)不动作。

4.8.6 DEC/BIN 減一

概要

指定的软元件数据中减“1”(-1 加法)的指令。

指令格式

16 位指令	DEC(P)	32 位指令	DDEC (P)
执行类型	连续执行型、脉冲执行型		
指令格式	DEC (P) D	DDEC (P) D	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC LZ	K、H	E
D	—	—	○*2	○*1	○	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	保存被减 一数据的 字软元件 编号	— 32768~32767	— 2147483648~2147483647	实数	实数

示例

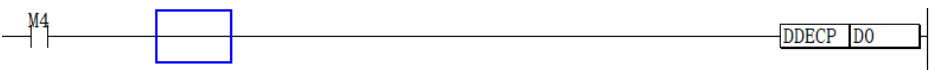
1. 16 位运算(DEC、DECP)

D0 的内容减一运算后，传送到 D0 中。



2. 32 位运算(DDEC、DDECP)

[D1, D0]的内容减一运算后，传送到[D1, D0]中。



报错相关：待添加

注意：

1) 16 位运算

+32,767 上加 1 后, 变为-32,768, 但是标志位(零、借位、进位)不动作。

2) 32 位运算

+2,147,483,647 上加 1 后, 变为-2,147,483,648, 但是标志位(零、借位、进位)不动作。

4.8.7 WAND/逻辑与

概要

2 个数值进行逻辑与运算的(AND)指令。

指令格式

16 位指令	WAND(P)	32 位指令	DWAND (P)
执行类型	连续执行型、脉冲执行型		
指令格式	WAND (P) S1 S2 D	DWAND (P) S1 S2 D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1	○	—	—	○	—
S2	—	—	○	○*1	○	—	—	○	—
D	—	—	○*2	○*1	○	—	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	逻辑与数据或保存数据的字软元件编号	-32768~32767	-2147483648~2147483647	实数	实数
S2	逻辑与数据或保存数据的字软元件编号	-32768~32767	-2147483648~2147483647	实数	实数
D	保存逻辑与结果的字软元件编号	-32768~32767	-2147483648~2147483647	实数	实数

示例

16 位运算(WAND, WANDP)

D0 和 D2 的内容以各位为单位进行逻辑与(AND)运算后，传送到 D4 中。

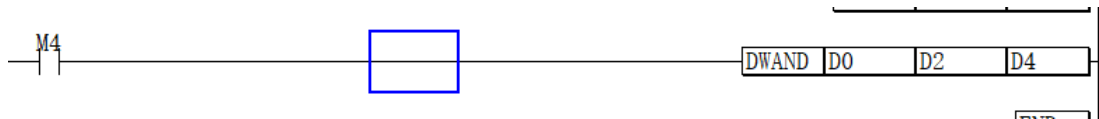


- 指令输入为 OFF 时，传送目标 D 的数据不变化。
- 指令输入为 ON 时，传送源(S1, S2)的数据不变化。
- 传送源 S1 和 S2 中指定常数(K)时，会自动进行 BIN 转换。
- 逻辑与运算以位为单位，如下表中(1∧1=1 0∧1=0 1∧0=0 0∧0=0)所示变化。

	S1	S2	D (1=ON,0=OFF)
			WAND
位单位的逻辑运算	0	0	0
	1	0	0
	0	1	0
	10	1	1

2. 32 位运算(DAND、DANDP)

[D1, D0]和[D3, D2]的内容以各位为单位进行逻辑与(AND)运算后，传送到[D 5, D4]中



- 指令输入为 OFF 时，传送目标[D+1, D]的数据不变化。
- 指令输入为 ON 时，传送源[S1+1, S1]、[S2 +1, S2]的数据不变化。
- 传送源[S1+1, S1]和[S2+1, S2]中指定常数(K)时，会自动进行 BIN 转换。
- 逻辑与运算以位为单位，如下表中(1∧1=1 0∧1=0 1∧0=0 0∧0=0)所示变化。

表中: 1=ON, 0=OFF

	S1+1, S1	S2 +1, S2	D+1, D
			DAND
位单位的逻辑运算	0	0	0
	1	0	0
	0	1	0
	1	1	1

报错相关：待添加

注意：待添加

4.8.8 WOR/逻辑或

概要

2 个数值进行逻辑或运算的(OR)指令。

指令格式

16 位指令	WOR(P)	32 位指令	DWOR (P)
执行类型	连续执行型、脉冲执行型		
指令格式	WOR (P) S1 S2 D	DWOR (P) S1 S2 D	

可使用软元件

操作数	位	字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1	○	—	—	○	—
S2	—	—	○	○*1	○	—	—	○	—
D	—	—	○*2	○*1	○	—	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	逻辑与数据或保存数据的字软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
S2	逻辑与数据或保存数据的字软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
D	保存逻辑与结果的字软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(WOR、WORP)

D0 和 D2 的内容以各位为单位进行逻辑或(OR)运算后，传送到 D4 中。



- 指令输入为 OFF 时，传送目标 D 的数据不变化。
- 指令输入为 ON 时，传送源 S1、S2 的数据不变化。
- 传送源 S1 和 S2 中指定常数(K)时，会自动进行 BIN 转换。
- 逻辑或运算以位为单位，如下表中(1 V 1=1 0 V 1=1 0 V 0=0 1 V 0=1)所示变化。

表中: 1=ON, 0=OFF

	S1	S2	D
			WOR

位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	1

2. 32 位运算(DWOR、DWORP)

[D1, D0]和[D3, D2]的内容针对各位进行逻辑或(OR)运算后, 传送到[D5, D4]中。



- 指令输入为 OFF 时, 传送目标 [D+1, D]的数据不变化。
- 指令输入为 ON 时, 传送源 [S1+1, S1], [S2+1, S2]的数据不变化。
- 传送源 [S1+1, S1]和[S2 +1, S2]中指定常数(K)时, 会自动进行 BIN 转换。
- 逻辑或运算以位为单位, 如下表中(1 V 1=1 0 V 1=1 0 V 0=0 1 V 0=1)所示变化。

表中: 1=ON, 0=OFF

位单位的逻辑运算	S1+1, S1	S2 +1, S2	D+1, D
			DWOR
	0	0	0
	1	0	1
	0	1	1
	1	1	1

报错相关: 待添加

注意: 待添加

4.8.9 WXOR/逻辑异或

概要

2 个数值进行逻辑异或运算的(XOR)指令。

指令格式

16 位指令	WXOR(P)	32 位指令	DWXOR (P)
执行类型	连续执行型、脉冲执行型		
指令格式	WXOR (P) S1 S2 D	DWXOR (P) S1 S2 D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1	○	—	—	○	—
S2	—	—	○	○*1	○	—	—	○	—
D	—	—	○*2	○*1	○	—	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	要进行逻辑异或的数据，或保存数据的字软元件编号	-32768~32767	-2147483648~2147483647	实数	实数
S2	要进行逻辑异或的数据，或保存数据的字软元件编号	-32768~32767	-2147483648~2147483647	实数	实数
D	保存逻辑异或结果的字软元件编号	-32768~32767	-2147483648~2147483647	实数	实数

示例

1. 16 位运算(WXOR、WXORP)

D0 和 D2 的内容以各位为单位，进行逻辑或(XOR)运算后，传送到 D4 中。



- 指令输入为 OFF 时，传送目标 的数据不变化。
- 指令输入为 ON 时，传送源(,)的数据不变化。
- 传送源 和 中指定常数(K)时，会自动进行 BIN 转换。
- 逻辑异或运算以位为单位，如下表中(1∨1=0 0∨0=0 1∨0=1 0∨1=1)所示变化。

表中: 1=ON, 0=OFF

	S1	S2	D
			WXOR
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	0

2. 32 位运算(DWXOR、DWXORP)

[D1, D0]和[D3, D2]的内容以各位为单位进行逻辑异或(XOR)运算后，传送到[D5, D4]中



- 指令输入为 OFF 时，传送目标[D+1, D]的数据不变化。
- 指令输入为 ON 时，传送源[S1+1, S1]、[S2+1, S2]的数据不变化。

- 传送源[S1+1, S1]和[S2+1, S2]中指定常数(K)时, 会自动进行 BIN 转换。
- 逻辑异或运算以位为单位, 如下表中(1∨1=0 0∨0=0 1∨0=1 0∨1=1)所示变化。

表中: 1=ON, 0=OFF

	S1+1, S1	S2 +1, S2	D+1, D
			DWXOR
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	0

报错相关：待添加

注意：待添加

4.8.10 NEG/补码

概要

求出数值的 2 进制补码(各位反转+1 后的值)的指令。

使用这个指令后, 可以反转数值的符号。

指令格式

16 位指令	NEG(P)	32 位指令	DNEG (P)
执行类型	连续执行型、脉冲执行型		
指令格式	NEG (P) D	DNEG (P) D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D	—	—	○*2	○*1	○	—	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	保存欲求补码的数据的字软元件编号, 以及保存目标软元件编	-32768~32767	-2147483648~2147483647	实数	实数

	号(运算结果被保存在同一字软元件编号中。)				
--	-----------------------	--	--	--	--

示例

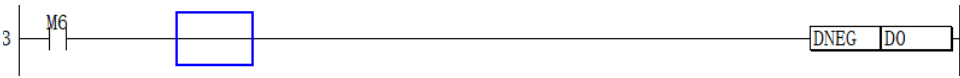
1. 16 位运算(NEG、NEGP)

将 D0 内容中的各位反转(0→1、1→0)后加一的结果保存到原先的软元件中。



2. 32 位运算(DNEG、DNEGP)

将[D1, D0]内容中的各位反转(0→1、1→0)后加一的结果保存到原先的软元件中。

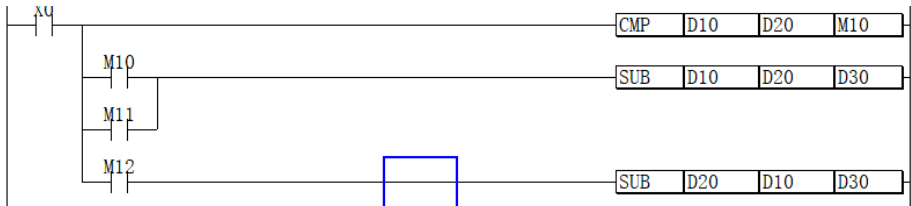


程序举例

下面介绍针对负的 BIN 值，求出其绝对值的例子。

使用 SUB(减法运算)指令的绝对值处理

即使不使用 NEG 指令(补码运算)，(D30)中也会一直保存差的绝对值。



(D10)>(D20) (D10)=(D20) (D10)<(D20)

M10=ON M11=ON M12=ON

(D10)≥D (20) 时

(D10) -(D20)→ (D30)

(D10) <(D20)时

(D20) - (D10) → (D30)

负数的显示及绝对值(参考)

在可编程控制器中，用 2 的补码显示负数。

最高位为 1 时为负数，使用补码指令可以求出其绝对值。

(D10) =2
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

(D10) =1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D10) =0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(D10) =-1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

(D10) =-2
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0



(D10) =-32767
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D10) =-32768
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

(D10) +1=1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

(D10) +1=2
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0



(D10) +1=32767
 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

(D10) +1=-32768
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

注：其中绝对值最大只能取到 32767

报错相关：待添加

注意:使用连续执行型(NEG、DNEG)指令时，每个扫描周期(各运算周期)都执行，请务必注意。

4.9 数据块运算

4.9.1 BK+/数据块的加法运算

概要

数据块的 BIN 加法运算的指令。

指令格式

16 位指令	BK+	32 位指令	DBK+
执行类型	连续执行型、脉冲执行型		
指令格式	BK+ (P) S1 S2 D N	DBK+(P) S1 S2 D N	

可使用软元件

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	○	—
				○*1	—	—	—	—	
				○*1	○	—	—	○	

*1 支持变址修饰

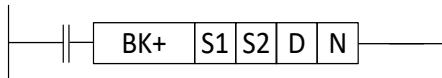
可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	存 储 了 加 法 运 算 数 据 的 起 始 软 元 件	-32768~32767	-2147483648~2147483647	实数	实数
S2	加 法 运 算 数 据 或 存 储 了 加 法 运 算 数 据 的 起 始 软 元 件	-32768~32767	-2147483648~2147483647	实数	实数
D	存 储 运 算 结 果 的 起 始 软 元 件	-32768~32767	-2147483648~2147483647	实数	实数
N	加 法 运 算 数 据 个 数	0~32767	0~32767	实数	实数

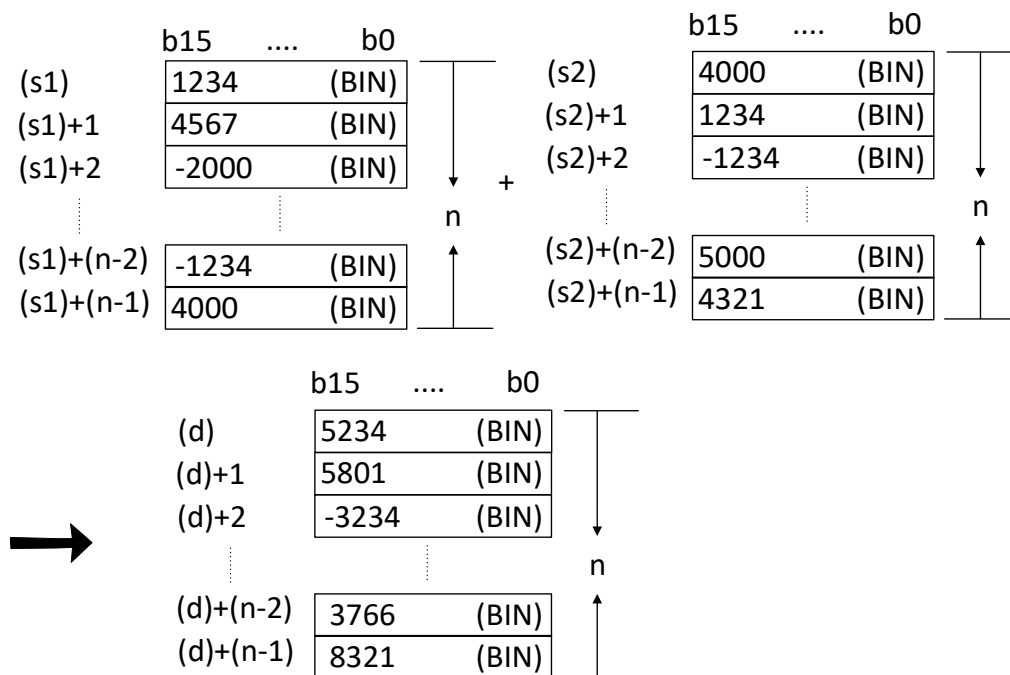
示例

1. 16 位运算(BK-/BK-P)

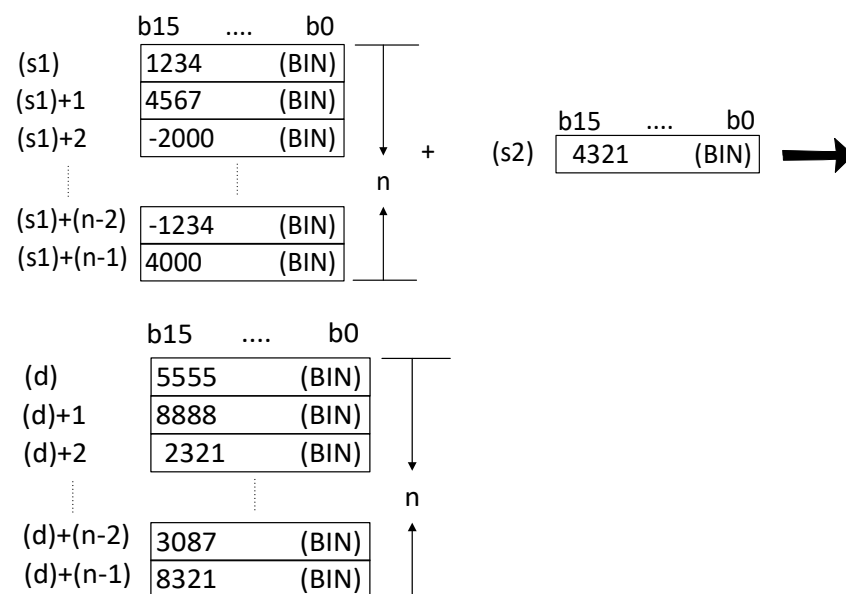
- 将(s1)中指定的软元件开始的(n)点的 BIN16 位数据与(s2)中指定的软元件开始的(n)点的 BIN16 位数据进行减法运算，将运算结果存储到(d)中指定的软元件中。
- 块减法运算以 16 位单位进行。



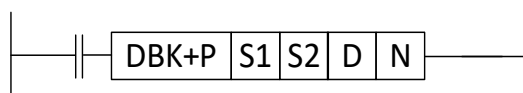
1) (s2)中指定了软元件的情况下



2) (s2)中指定了常数的情况下(有符号指定时)

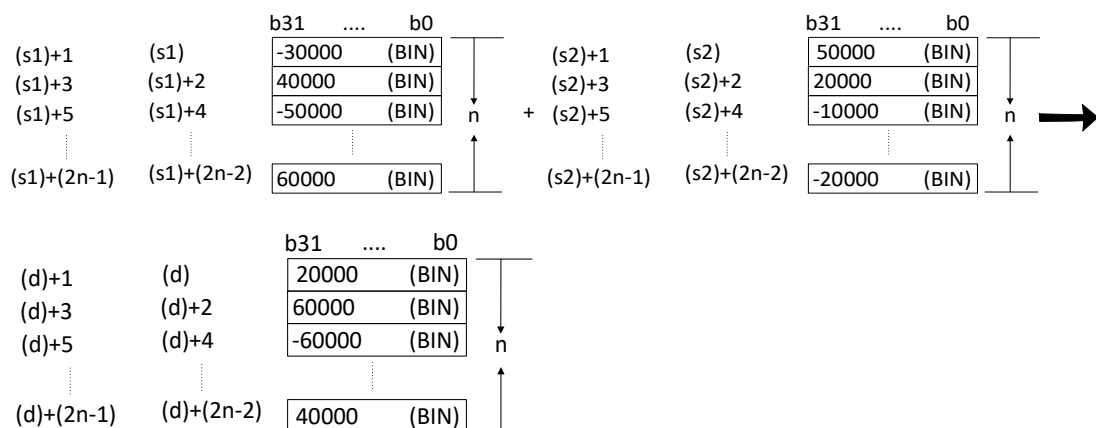


2.32 位运算 (DBK+ / DBK+P)

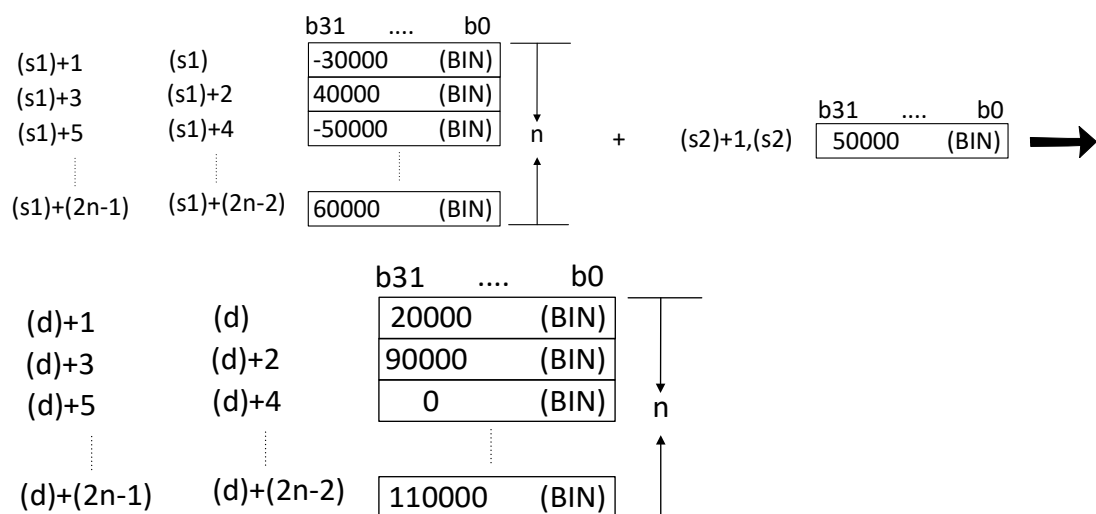


1) (s2)中指定了软元件的情况下(有符号指定时)

- 将(s1)中指定的软元件开始的(n)点的 BIN32 位数据与(s2)中指定的软元件开始的(n)点的 BIN32 位数据进行加法运算，将运算结果存储到(d)中指定的软元件中。
- 块加法运算以 32 位单位进行



2)(s2)中指定了常数的情况下(有符号指定时)



相关指令

指令	功能
BK-	数据块减法指令

报错相关：

以下一些情况下会发生运算错误，错误标志位 SM67 置 ON，错误代码保存在 SD67 中。

- S1,S2,D 开始的 n 点(32 位运算时为 2n 点)软元件超出相应的软元件范围时。

(错误代码:由开发人员定义)

- 开始的 n 点软元件和 开始的 n 点软元件重复时(32 位运算时为 2n 点)。(错误代码: :由开发人员定义)
- 开始的 n 点软元件和 开始的 n 点软元件重复时(32 位运算时为 2n 点)。(错误代码: :由开发人员定义)

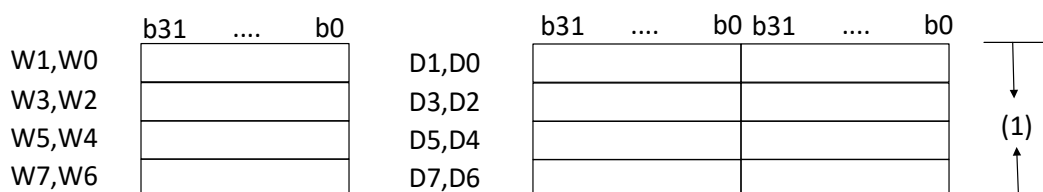
注意：•16 位指令运算结果发生了下溢及上溢时，其情况如下所示。在此情况下，进位标志不变为 ON。

$$\begin{array}{l} \text{k32767} \\ (7\text{FFFH}) \end{array} + \begin{array}{l} \text{K2} \\ (0002\text{H}) \end{array} \longrightarrow \begin{array}{l} \text{K-32767} \\ (8001\text{H}) \end{array}$$

$$\begin{array}{l} \text{K-32767} \\ (8001\text{H}) \end{array} + \begin{array}{l} \text{K-2} \\ (\text{FFFEH}) \end{array} \longrightarrow \begin{array}{l} \text{K32767} \\ (7\text{FFFH}) \end{array}$$

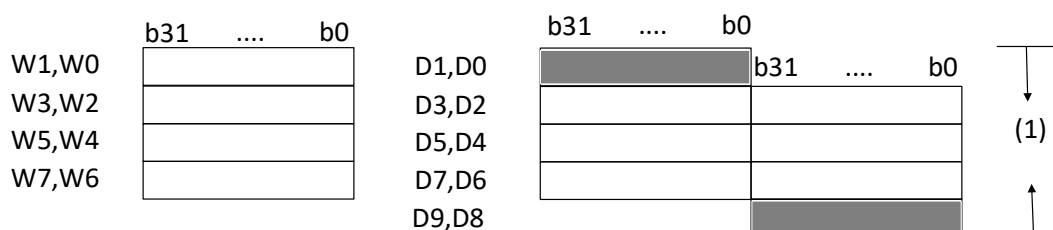
- 32 位指令(s1)或(s2)与(d)以同一软元件(完全一致)指定的情况下，可以运算。但是，(s1)或(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围有部分一致(重复)的情况下，将变为出错状态。

例：(s2)与(d)开始的 4 点的软元件完全一致的情况下



(1) 由于完全一致，因此可以运算

(s2)、(d)开始的 4 点的软元件有部分一致的情况下



(1) 由于部分一致，因此变为运算出错

- (n)中指定的值为 0 的情况下将变为无处理。
- 运算结果发生了下溢及上溢时，其情况如下所示。在此情况下，进位标志不变为 ON。

$$\begin{array}{l} \text{k2147483647} \\ (7\text{FFFFFFFH}) \end{array} + \begin{array}{l} \text{K2} \\ (\text{FFFFFFFEH}) \end{array} \longrightarrow \begin{array}{l} \text{K-2147483647} \\ (80000001\text{H}) \end{array}$$

$$\begin{array}{l} \text{K-2147483647} \\ (80000001\text{H}) \end{array} + \begin{array}{l} \text{K-2} \\ (00000002\text{H}) \end{array} \longrightarrow \begin{array}{l} \text{K2147483647} \\ (7\text{FFFFFFFH}) \end{array}$$

4.9.2 BK-/数据块的减法运算

概要

数据块的 BIN 减法运算的指令。

指令格式

16 位指令	BK-	32 位指令	DBK+
执行类型	连续执行型、脉冲执行型		
指令格式	BK- (P) S1 S2 D N		DBK-(P) S1 S2 D N

可使用软元件

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	○	—
				○*1	—	—	—	—	
				○*1	○	—	—	○	

*1 支持变址修饰

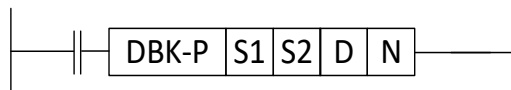
可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	存 储 了 加 法 运 算 数 据 的 起 始 软元件	-32768~32767	-2147483648~2147483647	实数	实数
S2	加 法 运 算 数 据 或 存 储 了 加 法 运 算 数 据 的 起 始 软 元 件	-32768~32767	-2147483648~2147483647	实数	实数
D	存 储 运 算 结 果 的 起 始 软元件	-32768~32767	-2147483648~2147483647	实数	实数
N	加 法 运 算 数 据 个 数	0~32767	0~32767	实数	实数

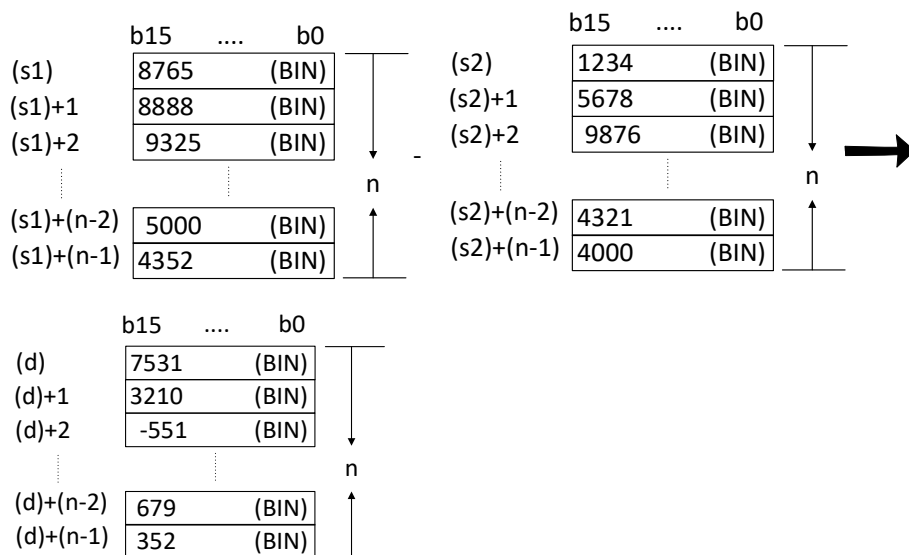
示例

1. 16 位运算(BK-/BK-P)

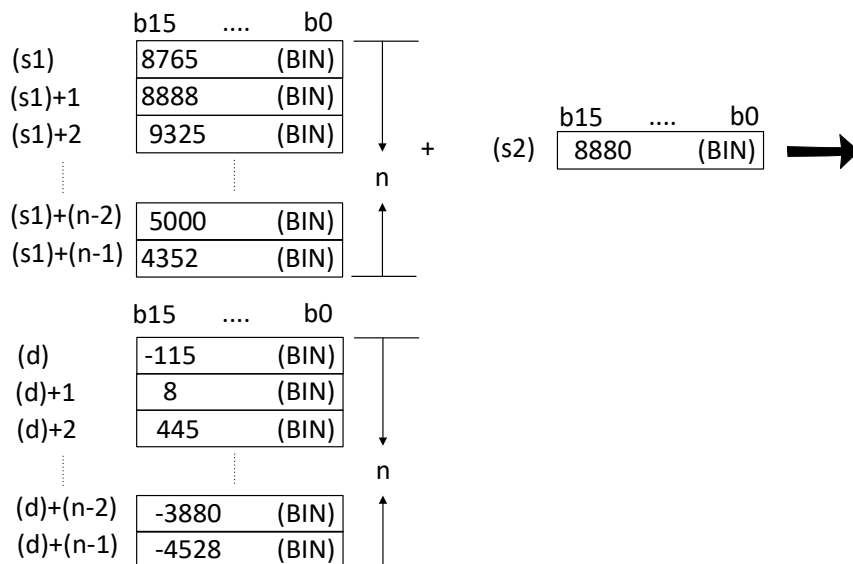
- 将(s1)中指定的软元件开始的(n)点的 BIN16 位数据与(s2)中指定的软元件开始的(n)点的 BIN16 位数据进行减法运算，将运算结果存储到(d)中指定的软元件中。
- 块减法运算以 16 位单位进行。



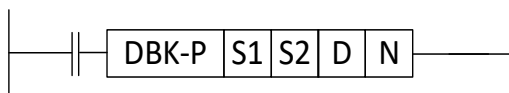
1) (s2)中指定了软元件的情况下



2) (s2)中指定了常数的情况下

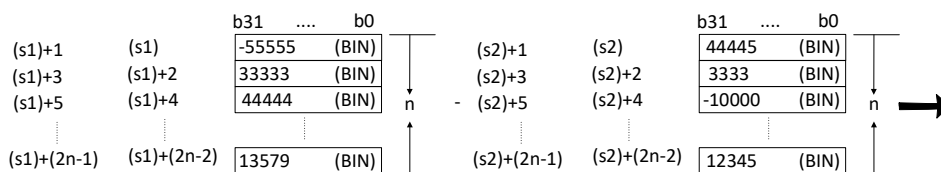


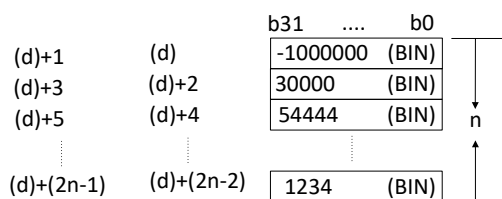
2. 32 位运算(DBK-/DBK-P))



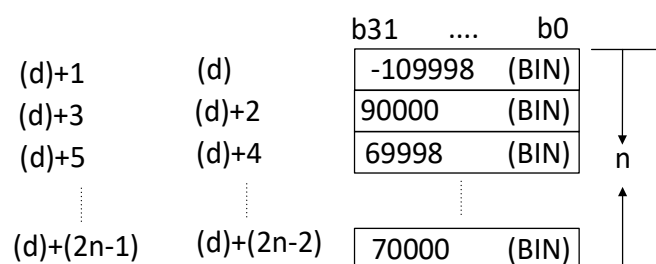
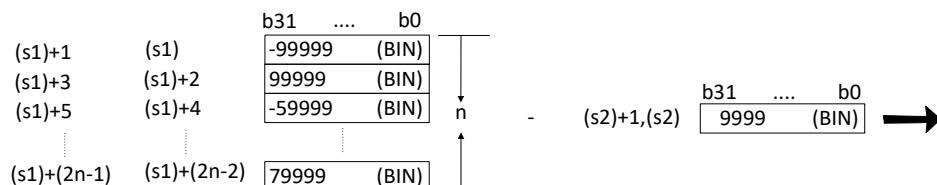
- 将(s1)中指定的软元件开始的(n)点的 BIN32 位数据与(s2)中指定的软元件开始的(n)点的 BIN32 位数据进行减法运算，将运算结果存储到(d)中指定的软元件中。
- 块减法运算以 32 位单位进行。

1) (s2)中指定了软元件的情况下(有符号指定时)





2) (s2)中指定了常数的情况下(有符号指定时)



相关指令

指令	功能
BK+	数据块加法指令

报错相关:

以下一些情况下会发生运算错误, 错误标志位 SM67 置 ON, 错误代码保存在 SD67 中。

- S1、S2、D 开始的 n 点(32 位运算时为 2n 点)软元件超出相应的软元件范围时。

(错误代码: 由开发人员定义)

- 开始的 n 点软元件和 开始的 n 点软元件重复时(32 位运算时为 2n 点)。(错误代码: 由开发人员定义)
- 开始的 n 点软元件和 开始的 n 点软元件重复时(32 位运算时为 2n 点)。(错误代码: 由开发人员定义)

注意:

- 16 位运算结果发生了下溢及上溢时, 其情况如下所示。在此情况下, 进位标志不变为 ON。

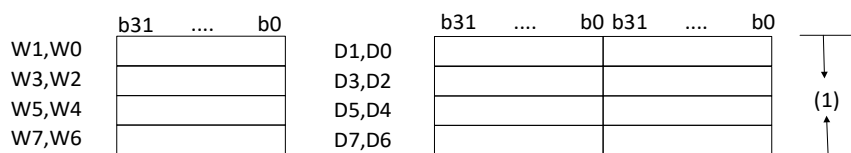
$$\begin{array}{ccc} \text{K-32767} & & \text{K2} \\ (8001\text{H}) & - & (0002\text{H}) \end{array} \longrightarrow \begin{array}{c} \text{K-32767} \\ (7\text{FFFH}) \end{array}$$

$$\begin{array}{ccc} \text{K32767} & & \text{K-2} \\ (7\text{FFFH}) & - & (\text{FFFEH}) \end{array} \longrightarrow \begin{array}{c} \text{K32767} \\ (8001\text{H}) \end{array}$$

- 32 位运算。(s1)或(s2)与(d)以同一软元件(完全一致)指定的情况下, 可以运算。但是, (s1)或(s2)开始的(n)点的软元件范围与(d)开始的(n)点的软元件范围有部分一致(重复)的情况下, 将变为出错状态。

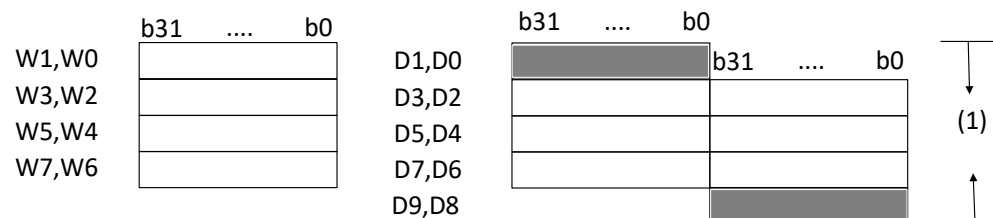
例

(s2)与(d)开始的 4 点的软元件完全一致的情况下



(1) 由于完全一致，因此可以运算

(s2)、(d)开始的 4 点的软元件有部分一致的情况下



(1) 由于部分一致，因此变为运算出错

- (n)中指定的值为 0 的情况下将变为无处理。
- 运算结果发生了下溢及上溢时，其情况如下所示。在此情况下，进位标志不变为 ON。

$$\begin{array}{l} \text{K2147483647} \\ (7\text{FFFFFFH}) \end{array} - \begin{array}{l} \text{K2} \\ (\text{FFFFFFEH}) \end{array} \longrightarrow \begin{array}{l} \text{K-2147483647} \\ (80000001\text{H}) \end{array}$$

$$\begin{array}{l} \text{K-2147483647} \\ (80000001\text{H}) \end{array} - \begin{array}{l} \text{K-2} \\ (00000002\text{H}) \end{array} \longrightarrow \begin{array}{l} \text{K2147483647} \\ (7\text{FFFFFFFH}) \end{array}$$

4.10 浮点运算

4.10.1 DEADD/2 进制浮点数加法运算

概要

1 个 2 进制浮点数加法运算的指令

指令格式

16 位指令	-	32 位指令	DEADD(P)
执行类型	连续执行型，脉冲执行型		
指令格式	-	DEADD (P) S1 S2 D	

可使用软元件

操作数	位		字			双字		常数		
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	—	○*1	—	—	—	○	○	
S2				○*1				○	○	
D	—	—	—	○*1	—	—	—	—	—	

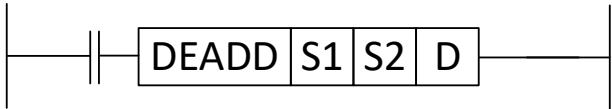
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存进行加法运算的 2 进制浮点数数据的字软元件编号	—	- 2147483648~2147483647	—	实数
S2	保存进行加法运算的 2 进制浮点数数据的字软元件编号	—	- 2147483648~2147483647	—	实数
D	保存加法运算后的 2 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	-	实数

示例

1.32 位运算



将[S1+1, S1]和[S2+1, S2]的 2 进制浮点数数据进行加法运算，并将其运算结果以 2 进制浮点数形式传送到[D+1, D]中

报错相关：（待添加）

注意：（待添加）

4.10.2 DESUB/2 进制浮点数减法运算

概要

1 个 2 进制浮点数减法运算的指令

指令格式

16 位指令	-	32 位指令	DESUB(P)
执行类型	连续执行型，脉冲执行型		
指令格式	-	DESUB (P) S1 S2 D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	○	○
S2	—	—	—	○*1	—	—	—	○	○
D	—	—	—	○*1	—	—	—	—	—

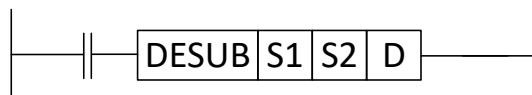
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存执行 减法运算 的 2 进制 浮点数数 据的字软 元件编号	—	- 2147483648~2147483647	—	实数
S2	保存执行 减法运算 的 2 进制 浮点数数 据的字软 元件编号	—	- 2147483648~2147483647	—	实数
D	保存减法 运算后的 2 进制浮 点数数据 的数据寄 存器编号	-	- 2147483648~2147483647	-	实数

示例

1.32 位运算



从[S1+1, S1]中减去[S2+1, S2]的 2 进制浮点数数据，并将其运算结果以 2 进制浮点数形式传

送到[D+1， D]中。

报错相关：（待添加）

注意：（待添加）

4.10.3 DEMUL/2 进制浮点数乘法运算

概要

1 个 2 进制浮点数乘法运算的指令

指令格式

16 位指令	-	32 位指令	DEMUL(P)
执行类型	连续执行型，脉冲执行型		
指令格式	-	DEMUL (P) S1 S2 D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	○	○
S2				○*1				○	○
D	—	—	—	○*1	—	—	—	—	—

*1 支持变址修饰

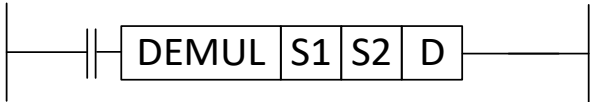
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存执行乘法运算的 2 进制浮点数数据的字软元件编号	-	- 2147483648~2147483647	-	实数

S2	保存执行乘法运算的 2 进制浮点数数据的字软元件编号	-	- 2147483648~2147483647	-	实数
D	保存乘法运算后的 2 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	-	实数

示例

1.32 位运算



将[S1+1, S1]和[S2+1, S2]的 2 进制浮点数数据相乘，并将其运算结果以 2 进制浮点数形式传送到[D+1, D]中

报错相关：(待添加)

注意：(待添加)

4.10.4 DEDIV/2 进制浮点数除法运算

概要

1 个 2 进制浮点数除法运算的指令

指令格式

16 位指令	-	32 位指令	DEDIV(P)
执行类型	连续执行型，脉冲执行型		
指令格式	-	DEDIV (P) S1 S2 D	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z		LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	—	○	○
S2				○*1					○	○

D	—	—	—	○*1	—	—	—	—	—
---	---	---	---	-----	---	---	---	---	---

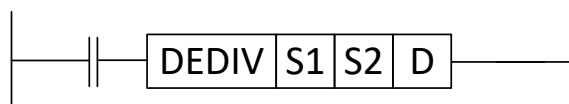
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存执行除法运算的 2 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	—	实数
S2	保存执行除法运算的 2 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	—	实数
D	保存乘法运算后的 2 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	-	实数

示例

1.32 位运算



将[S1+1, S1]和[S2+1, S2]的 2 进制浮点数数据相除，并将其运算结果以 2 进制浮点数形式传送到[D +1, D]中

报错相关：(待添加)

注意：(待添加)

4.10.5 DSIN(P):二进制浮点正弦指令

概要

对指定数据进行浮点数 SIN 运算的指令

指令格式

16 位指令	—	32 位指令	DSIN/DSINP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DSIN(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	○ 1	*	—	—
D	—	—	—	○*1	○	○ 1	*	—	—

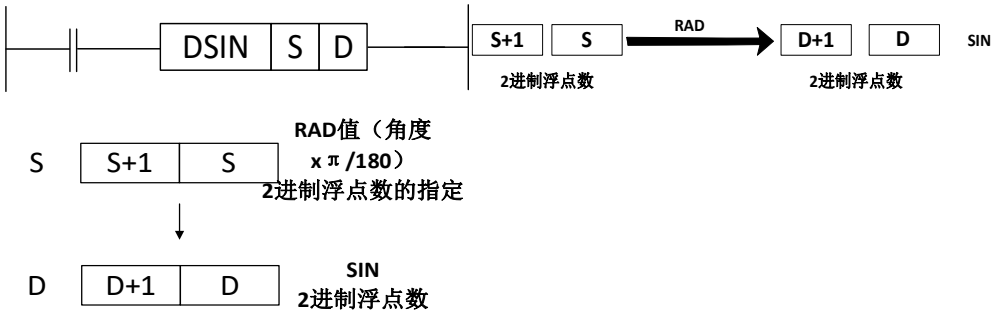
*1 支持变址修饰

可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进行 SIN 运算的数据或软元件编号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)
D	储存运算结果的软元件编号	—	$-1 \sim -2^{-126}$, $0.0, 2^{-126} \sim 1$	—	32 位 (2 进制)

示例

将[S+1, S]中指定的角度值(2 进制浮点数)转换成 SIN 值后, 传送到[D+1, D]中。



相关软元件如下所示。

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零标志(SM20)变为 ON。
SM21	借位	运算结果的绝对值 $< 2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126}), 借位标志

		(SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

报错相关：操作数 (s) (s+1)的内容仅正数有效，若为负数时，运算错误(6706)动作，不执行指令

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码：6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意: 待添加

4.10.6 DCOS(P):二进制浮点余弦指令

概要

对指定数据进行浮点数 SIN 运算的指令

指令格式

16 位指令	—	32 位指令	DCOS/DCOSP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DCOS(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	○ 1	*	—	—
D	—	—	—	○*1	○	○ 1	*	—	—

*1 支持变址修饰

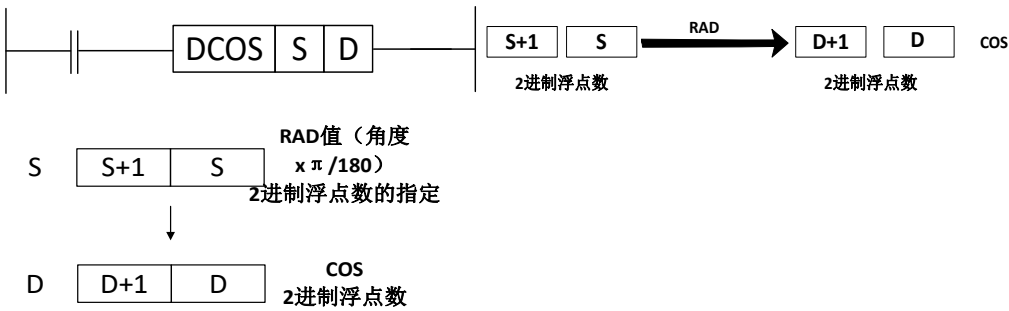
可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进 行 COS 运 算 的 数 据 或 软 元 件 编 号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)

D	储存运算结果的软元件编号	-	$-1 \sim -2^{-126}$, $0.0, 2^{-126} \sim 1$	-	32 位 (2 进制)
---	--------------	---	--	---	-------------

示例

将[S+1, S]中指定的角度值(2 进制浮点数)转换成 COS 值后，传送到[D+1, D]中。



相关软元件如下所示。

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时) 零 标志 (SM20) 变 为 ON。
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126})，借位标志 (SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

报错相关：指定软元件的内容为非正规数、非数、 $\pm \infty$ 时。(错误代码：6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意: 待添加

4.10.7 DTAN(P):二进制浮点正切指令

概要

对指定数据进行浮点数 TAN 运算的指令

指令格式

16 位指令	—	32 位指令	DTAN/DTANP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DTAN(P) S D	

可使用软元件

操作数	位		字			双字		常数		
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E	
S	—	—	—	○*1	—	○ 1	*	—	—	○
D	—	—	—	○*1	○	○ 1	*	—	—	—

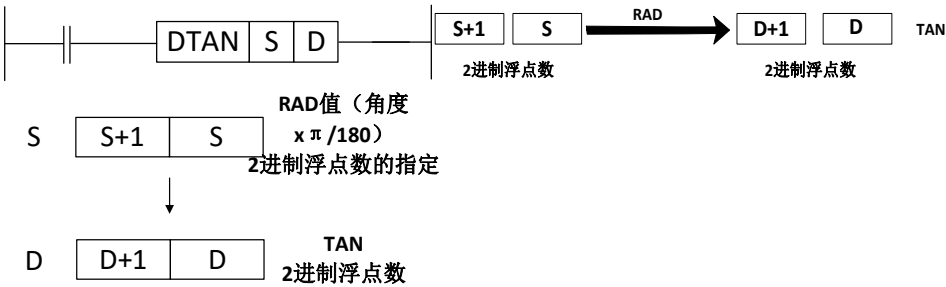
*1 支持变址修饰

可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进 行 TAN 运 算 的 数 据 或 软 元 件 编 号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)
D	储 存 运 算 结 果 的 软 元 件 编 号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)

示例

将 [S+1, S] 中指定的角度值(2 进制浮点数)转换成 TAN 值后, 传送到 [D+1, D] 中。



相关特殊软元件

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零 标 志 (SM20) 变 为 ON。
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126}), 借位标志 (SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128}), 进位标志(SM22)变为 ON。

--	--	--

报错相关：操作数（s）（s+1）的内容仅正数有效，若为负数时，运算错误(6706)动作，不执行指令

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。（错误代码：6706）

操作数（s）（d）中指定的软元件范围超出相应软元件的范围时。（错误代码：6004）

注意: 待添加

4.10.8 DASIN(P):二进制浮点反正弦指令

概要

对指定数据进行浮点数 SIN^{-1} 运算的指令

指令格式

16 位指令	—	32 位指令	DASIN/DASINP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DASIN(P) S D	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E	
S	—	—	—	○*1	—	○ 1	*	—	—	○
D	—	—	—	○*1	○	○ 1	*	—	—	—

*1 支持变址修饰

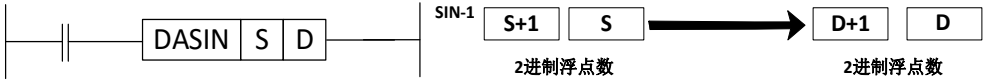
可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进 行 SIN^{-1} 运 算 的 数 据 或 包 含 SIN 值 的 软 元 件 编 号	—	$-1 \sim -2^{-126}$, $0.0, 2^{-126} \sim 1$	—	32 位（2 进制）
D	储 存 运 算 结 果 的 软 元	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位（2 进制）

	件编号				
--	-----	--	--	--	--

示例

将[S+1, S]中指定的 SIN 值(2 进制浮点数)求出角度后, 将运算结果传送到[D+1, D]中。
此外, 可以在 S 中直接指定实数



相关软元件如下所示。

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零标志(SM20)变为 ON。
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126}), 借位标志(SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128}), 进位标志(SM22)变为 ON。

报错相关: 指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码: 6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码: 6004)

当操作数 (s) 中指定的值不在 -1~1 的范围内时会产生运算错误 (错误代码: 6706)

注意: 待添加

4.10.9 DACOS(P):二进制浮点反余弦指令

概要

对指定数据进行浮点数 $\cos^{-1}$ 运算的指令

指令格式

16 位指令	—	32 位指令	DACOS/DACOSP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DACOS(P) S D	

可使用软元件

操作数	位	字	双字	常数
X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z LC LZ K、 H E

S	—	—	—	○*1	—	○*1	—	—	○
D	—	—	—	○*1	○	○*1	—	—	—

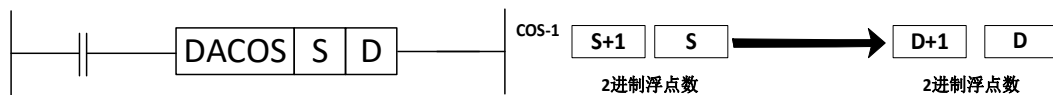
*1 支持变址修饰

可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进 行 $\cos^{-1}$ 运算的数 据或包含 COS 值软元 件编号	—	$-1 \sim -2^{-126}$, $0.0, 2^{-126} \sim 1$	—	32 位 (2 进制)
D	储 存 运 算 结 果 的 软 元 件 编 号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)

示例

将[S+1, S]中指定的 COS 值(2 进制浮点数)求出角度后, 将运算结果传送到[D+1, D]中。
此外, 可以在 S 中直接指定实数



[S+1,S]的 COS 值可在-1.0~1.0 的范围内设定。

[D+1, D]中保存的角度 (运算结果) 是保存弧度 ($-\pi/2$) ~ ($\pi/2$) 的值

相关特殊软元件

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零 标 志 (SM20) 变 为 ON。
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126}), 借位标志 (SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128}), 进位标志(SM22)变为 ON。

报错相关: 指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码: 6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码: 6004)

当操作数 (s) 中指定的值不在-1~1 的范围内时会产生运算错误 (错误代码: 6706)

注意: 待添加

4.10.10 DATAN(P):二进制浮点反正切指令

概要

对指定数据进行浮点数 $\cos^{-1}$ 运算的指令

指令格式

16 位指令	—	32 位指令	DACOS/DACOSP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DACOS(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	○ 1	*	—	—
D	—	—	—	○*1	○	○ 1	*	—	—

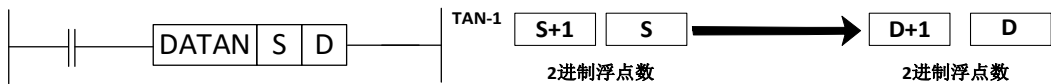
*1 支持变址修饰

可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进 行 $\tan^{-1}$ 运算的数 据或包含 TAN 值的软元 件编号	—	$-1 \sim -2^{-126}$, $0.0, 2^{-126} \sim 1$	—	32 位 (2 进制)
D	储 存 运 算 结 果 的 软 元 件 编 号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)

示例

将[S+1, S]中指定的 TAN 值(2 进制浮点数)求出角度后, 将运算结果传送到[D+1, D]中。
此外, 可以在 S 中直接指定实数



[D+1, D]中保存的角度（运算结果）是保存以弧度为单位的比 $(-\pi/2)$ 大，比 $(\pi/2)$ 小的数值

相关特殊软元件

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零标志(SM20)变为 ON。
SM21	借位	运算结果的绝对值 $< 2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126})，借位标志(SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

报错相关：指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。（错误代码：6706）

操作数（s）（d）中指定的软元件范围超出相应软元件的范围时。（错误代码：6004）

当操作数（s）中指定的值不在 -1~1 的范围内时会产生运算错误（错误代码：6706）

注意: 待添加

4.10.11 DRAD(P):二进制浮点角度转弧度指令

概要

对指定角度单位的值转换成弧度单位的指令

指令格式

16 位指令	—	32 位指令	DRAD/DRADP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DRAD(P) S D	

可使用软元件

操作数	位		字			双字		常数		
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E	
S	—	—	—	○*1	—	○ 1	*	—	—	○

D	—	—	—	○*1	○	○*	—	—	—
						1			

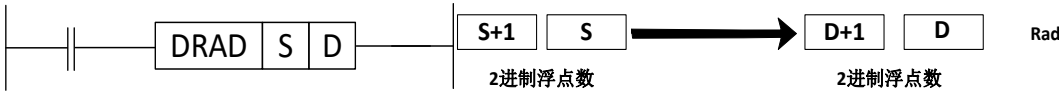
*1 支持变址修饰

可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储 存 要 转 换 成 弧 度 单 位 的 角 度 的 软 元 件 起 始 编 号 或 数 据	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)
D	储 存 运 算 结 果 的 软 元 件 起 始 编 号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)

示例

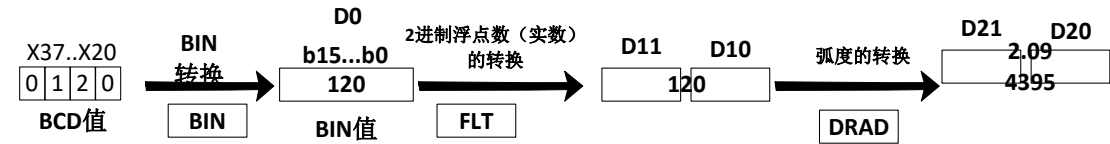
将[S+1, S] 的单位从角度单位转换成弧度单位，并保存到[D+1, D]中。此外，可以在 S 中直接指定实数



角度单位→弧度单位的转换如下：

弧度单位=角度单位* (π/180)

X0 为 ON 时，将 X20 ~ X37 中以 BCD4 位数形式设定的角度转换成弧度，以 2 进制浮点数形式保存在 D20、D21 中的程序



相关特殊软元件

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零 标 志 (SM20) 变 为 ON。

SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126}), 借位标志(SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128}), 进位标志(SM22)变为 ON。

报错相关: 指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码: 6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码: 6004)

注意: 待添加

4.10.12 DDEG(P): 二进制浮点弧度转角度指令

概要

将弧度单位的值转换成角度(DEG)单位的指令

指令格式

16 位指令	—	32 位指令	DDEG/DDEGP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DDEG(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、 H	E
S	—	—	—	○*1	—	○ 1	*	—	—
D	—	—	—	○*1	○	○ 1	*	—	—

*1 支持变址修饰

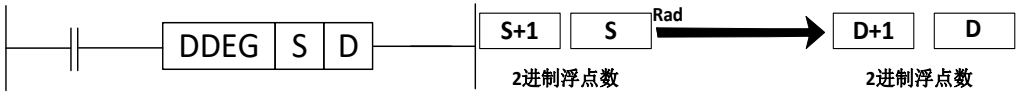
可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储 存 转 换 成 角 度 单 位 的 弧 度 的 软 元 件 起 始	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位 (2 进制)

	编 号 或 数据				
D	储 存 已 转 换 成 角 度 单 位 的 值 的 软 元 件 起 始 编 号	-	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	-	32 位 (2 进制)

示例

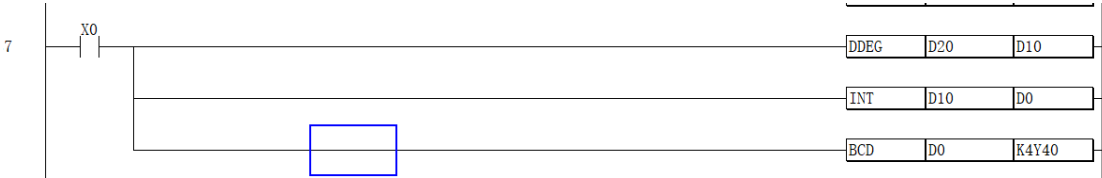
将[S+1, S] 的单位从弧度单位转换成角度单位，并保存到[D+1, D]中。



角度单位→弧度单位的转换如下：

弧度单位=角度单位* (π/180)

X0 为 ON 时，将 X20 ~ X37 中以 BCD4 位数形式设定的角度转换成弧度，以 2 进制浮点数形式保存在 D20、D21 中的程序



相关特殊软元件

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时)零标志 (SM20) 变为 ON。
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126})，借位标志 (SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

报错相关：指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码：6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意: 待添加

4.10.13 DEXP:二进制浮点数指数指令

概要

该指令是以 e(2.71828)为底的指数运算指令

指令格式

16 位指令	—	32 位指令	DEXP/DEXPP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DEXP(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	○ 1	*	—	○
D	—	—	—	○*1	—	○ 1	*	—	—

*1 支持变址修饰

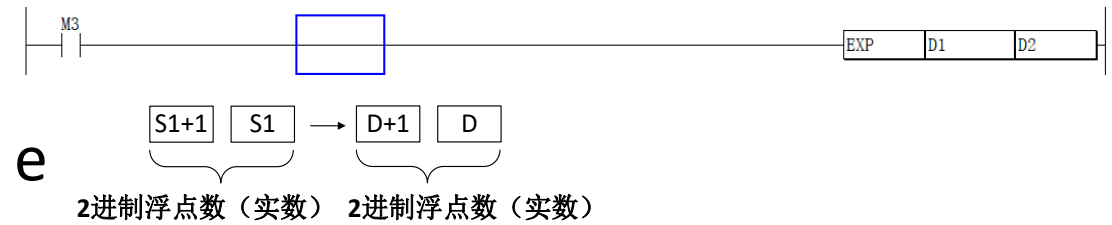
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	进行 EXP 运算的数据或软元件编号	-	-2128~-2-128, 0.0,2-128~2128	-	进行 EXP 运算的数据或软元件编号
D	储存运算结果的软元件编号	-	2-128 ~ +2128	-	储存运算结果的软元件编号

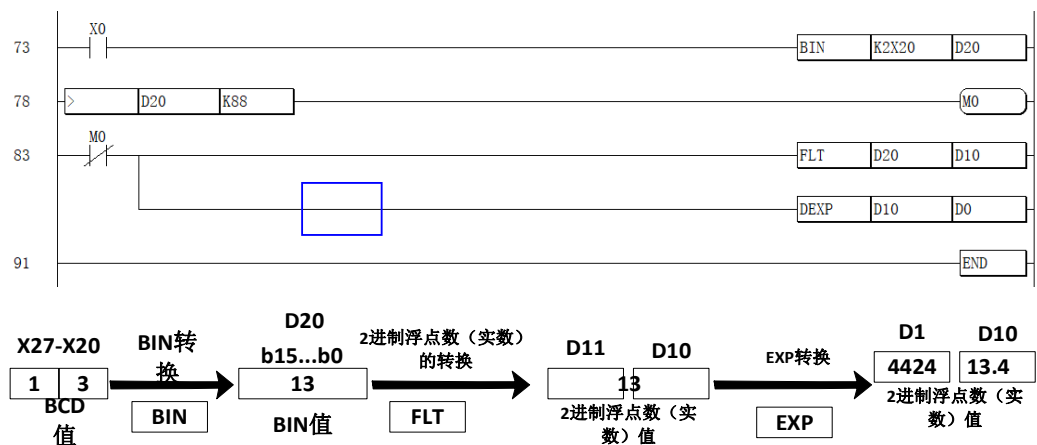
示例

将[S+1, S] 为指数做运算，将运算结果保存到[D+1, D]中。

此外，可以在 S 中直接指定实数



在指数运算中，将底（e）作为“2.71828”进行计算



相关软元件如下所示。

软元件	名称	介绍
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126})，借位标志 (SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

。

报错相关：操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

运算结果不在下面的范围时。(错误代码: 6706)

$2^{-126} \leq | \text{运算结果} | < 2^{128}$

注意:

1.由于 $\log_e 2^{128}=88.7$ ，因此当 X20 ~ X027 的 BCD 值为 88 以下时，此时运算结果不满 2^{128} 。设定了 89 以上的数值时，会发生运算错误，因此设定了 89 以上的数值时，M0 置 ON，从而不执行运算。

2. 从自然对数向常用对数的转换

在 CPU 中，执行自然对数的运算。

要求出常用对数值时，请在[S+1, S]中指定用 0.4342945 分割常用对数的值。

4.10.14 DLOGE 二进制浮点数自然对数指令

概要

比较 2 个数据(2 进制浮点数)，将结果(大于、等于或小于)输出到位软元件(3 点)中的指令

指令格式

16 位指令	—	32 位指令	DLOGE/DLOGEP
--------	---	--------	--------------

执行类型	连续执行型、脉冲执行型	
指令格式	—	DLOGE(P) S D

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	○ 1	*	—	○
D	—	—	—	○*1	—	○ 1	*	—	—

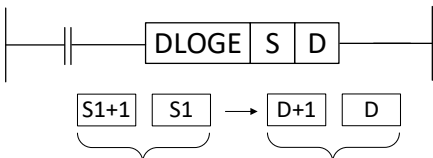
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储存保存进行自然对数运算的 2 进制浮点数数据的软元件的起始编号	-	$2^{-126} \sim 2^{128}$	-	32 位（2 进制）
D	储存运算结果的软元件编号	-	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	-	32 位（2 进制）

示例

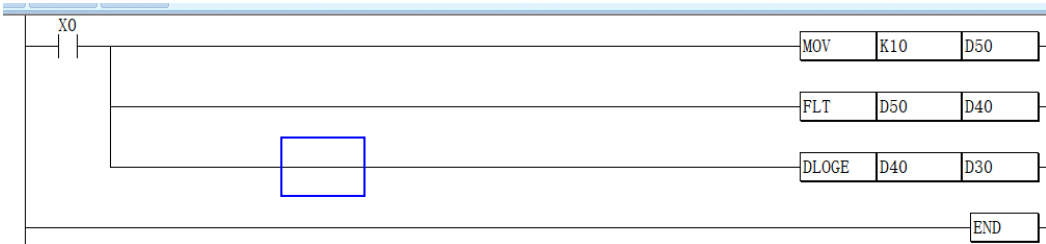
将[S+1, S] 的自然对数[以 e(2.17828)为底时的对数]运算，并将运算结果保存到[D+1, D] 中。此外，可以在 S 中直接指定实数

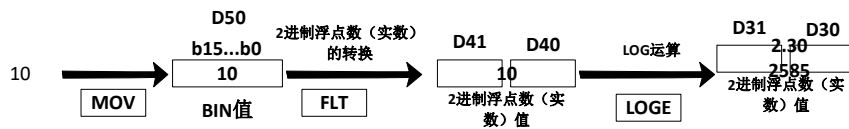


2进制浮点数（实数） 2进制浮点数（实数）

[S+1,S]中指定的值，只可以设定正数。（负数不能运算）。

X0 为 ON 后，求出 D50 中设定的“10”的自然对数，并保存到 D30、D31 中的程序。





相关软元件如下所示。

软元件	名称	介绍
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126})，借位标志(SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

报错相关：运算时操作数（s）在下面的范围时会报错。(错误代码:6706)

操作数（s）中指定的值是负数时。

操作数（s）中指定的值是 0 时

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码：6706)

操作数（s）(d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意：

待添加

4.10.15 DLOG10: 2 进制浮点数常用对数指令

概要

该指令执行自然对数运算。

指令格式

16 位指令	—	32 位指令	DLOG10/DLOG10P
执行类型	连续执行型、脉冲执行型		
指令格式	—	DLOG10(P) S D	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S	—	—	—	○*1	—	○* 1	—	—	○	
D	—	—	—	○*1	—	○* 1	—	—	—	

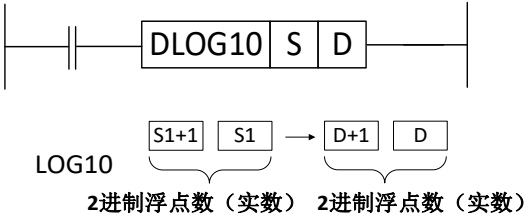
*1 支持变址修饰

可使用操作数

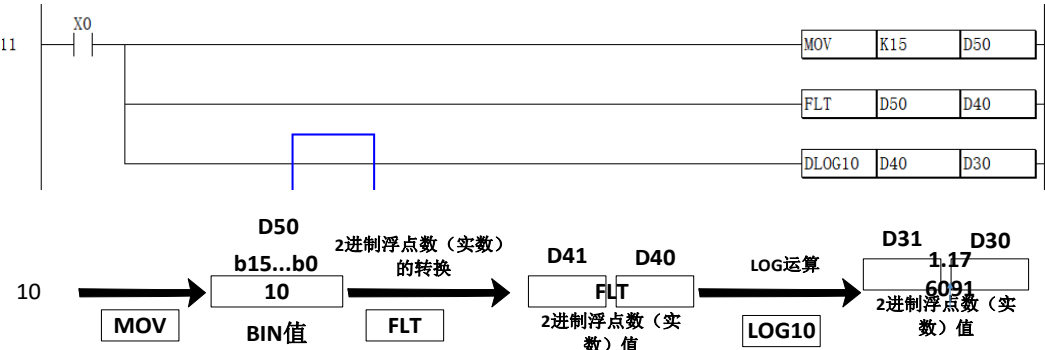
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储存进行常用对数运算的 2 进制浮点数数据的软元件的起始编号	-	$2^{-126} \sim 2^{128}$	-	32 位（2 进制）
D	储存运算结果的软元件编号	-	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	-	32 位（2 进制）

示例

将[S+1, S] 的常用对数(10 为底时的对数)运算，并将运算结果保存到[D+1, D]中。
此外，可以在 S 中直接指定实数



X0 为 ON 后，求出 D50 中设定的“15”的自然对数，并保存到 D30、D31 中的程序。



[S+1,S]中指定的值，只可以设定正数。（负数不能运算）。

相关软元件如下所示。

软元件	名称	介绍
SM20	零位	运算结果为零(尾数部分为“0”时) 零 标志 (SM20) 变 为 ON。
SM21	借位	运算结果的绝对值 $<2^{-126}$ 操作数(d)的值变为 32 位实数的最小值(2^{-126})，借位标志 (SM21)变为 ON。
SM22	进位	运算结果的绝对值 $\geq 2^{128}$ 操作数(d)的值变为 32 位实数的最大值(2^{128})，进位标志(SM22)变为 ON。

--	--	--

报错相关：运算时操作数（s）在下面的范围时会报错。（错误代码:6706）

操作数（s）中指定的值是负数时。

操作数（s）中指定的值是 0 时

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。（错误代码：6706）

操作数（s）（d）中指定的软元件范围超出相应软元件的范围时。（错误代码：6004）

注意: 待添加

4.10.16 DESQR:2 进制浮点开方指令

概要

该指令执行浮点开方运算。

指令格式

16 位指令	—	32 位指令	DESRQ/DESRP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DESRQ(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	○ 1	*	—	—
D	—	—	—	○*1	—	○ 1	*	—	—

*1 支持变址修饰

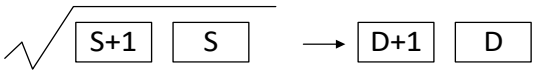
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储存进行常用对数运算的 2 进制浮点数数据的软元件的起始编号	-	$2^{-126} \sim 2^{128}$	-	32 位（2 进制）
D	储存运算结果的软元件编号	-	$-2^{128} \sim -2^{126}$, 0.0, $2^{126} \sim 2^{128}$	-	32 位（2 进制）

示例

32 位运算(DESQR, DESQRP)

将[S+1, S] 进行开方(开根号)运算(2 进制浮点数运算)后, 并将运算结果保存到[D+1, D] 中。此外, 可以在 S 中直接指定实数



相关软元件

软元件	名称	内容
SM20	零位	运算结果为 0 时为 ON

报错相关: 操作数 (s) (s+1)的内容仅正数有效, 若为负数时, 运算错误(6706)动作, 不执行指令

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码: 6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码: 6004)

注意: 待添加

4.10.17 DECMP /2 进制浮点数比较

概要

在通信等中使用的错误校验方法有水平校验以及和校验, 该指令是用于计算校验值的。

指令格式

16 位指令	ECM(P)	32 位指令	DECMP(P)
执行类型	连续执行型, 脉冲执行型		
指令格式	ECM (P) S D N	DECMP (P) S D N	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	○	○
D	—	—	○	○*1	—	—	—	○	○
N	○*1	○	—	—	—	—	—	—	—

*1 支持变址修饰

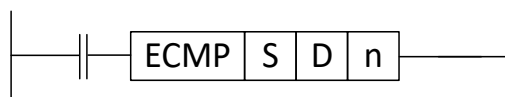
可使用操作数

操作数	内容	数据范围	数据类型
-----	----	------	------

		16 位范围	32 位范围	16 位	32 位
S	保存要比较的 2 进制浮点数数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	-
D	保存要比较的 2 进制浮点数数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	-
N	输出结果的起始位软元件编号(占用 3 点)	0~32767	0~32767	实数	实数

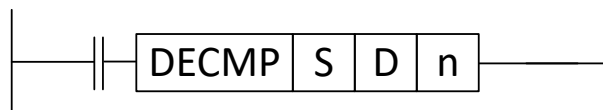
示例

1.16 位浮点数比较



比较[S1]与[S2]中指定的软元件的数据,根据结果(小于、一致、大于), [D]、[D]+1、[D]+2 中的一项将变为 ON

2. 32 位浮点数比较



比较[S1+1, S1]与[S2+1, S2]中指定的软元件的数据,根据结果(小于、一致、大于), [D]、[D]+1、[D]+2 中的一项将变为 ON

报错相关: (待添加)

注意: (待添加)

4.10.18 DENEG:2 进制浮点符号翻转指令

概要

使 2 进制浮点数(实数)数据的符号翻转的指令

指令格式

16 位指令	—	32 位指令	DENEG/DENEGP
执行类型	连续执行型、脉冲执行型		
指令格式	—	DENEG(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D	—	—	—	○*1	—	○* 1	—	—	—

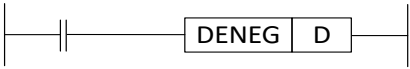
*1 支持变址修饰

可使用操作数

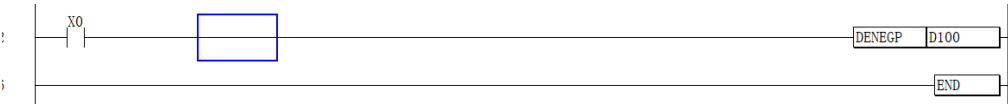
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	保存要执行 符号翻转的 2 进制浮点 数数据的软 元件的起始 编号	—	$-2^{128} \sim -2^{-126}$, $0.0, 2^{-126} \sim 2^{128}$	—	32 位（2 进制）

示例

将[D+1, D] 的 2 进制浮点数数据的符号翻转，并将运算结果保存到[D+1, D]中。



X0 为 ON 后，将 D100、D101 的 2 进制浮点数数据的符号翻转，并且保存到 D100、D101 中的程序



报错相关：指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。（错误代码：6706）

操作数（d）中指定的软元件范围超出相应软元件的范围时。（错误代码：6004）

注意: 待添加

4.11 数据转换

4.11.1 INT:二进制浮点数转换二进制整数指令

概要

将 2 进制浮点数，转换成可编程控制器中的一般数据形式 BIN 整数的指令
(2 进制浮点数→BIN 整数)

指令格式

16 位指令	INT/INTP	32 位指令	DINT/DINTP
执行类型	连续执行型、脉冲执行型		
指令格式	INT(P) S D		DINT(P) S D

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	○* 1	—	—	○
D	—	—	○*1	○*1	○	○* 1	—	—	—

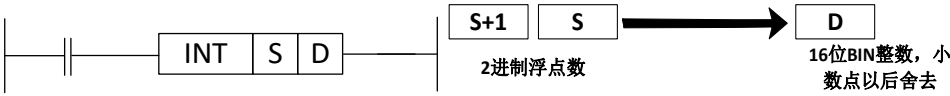
*1 支持变址修饰，LC 作为操作数时仅 32 位指令 DINT(P)可输入

可使用操作数

操 作 数	内 容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储 存 要 转 换 成 BIN 整数 的 2 进制 浮 点 数 数 据 的 数 据 寄 存 器 编 号	—	$-2^{-128} \sim -2^{-126}$ ，0.0， $2^{-126} \sim 2^{128}$	—	32 位（2 进制）
D	储 存 转 换 后 的 BIN 整数 的 数 据 寄 存 器 编 号	-32768~32767	— 2147483648~2147483647	16 位 （ 2 进 制）	32 位（2 进制）

示例

将[S+1， S] 的 2 进制浮点数转换成 BIN 整数后，并将结果传送到[D]中。



将[S+1， S] 的 2 进制浮点数转换成 BIN 整数后，并将结果传送到[D+1， D]中。



相关软元件

软元件	名称	内容
SM20	零位	运算结果为 0 时为 ON
SM21	借位	借位转换时，有因不满 1 而被舍去的情况发生时，置 ON。
SM22	进位	运算结果超出-32,768～32,767(16 位运算时)，或是-2,147,483,648～2,147,483,647(32 位运算时)的范围而产生溢出时为 ON。(运算结果不反映。)

注意：小数点以后的值被舍去

报错相关：指定软元件的内容为非正规数、非数、 $\pm\infty$ 时。(错误代码：6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意：待添加

4.11.2 BCD 指令

概要

将 BIN(2 进制数)转换成 BCD(10 进制数)后传送的指令。

可编程控制器的运算按照 BIN(2 进制数)数据进行处理，在带 BCD 译码的 7 段码显示器中显示数值时,可使用本指令。

指令格式

16 位指令	BCD(P)	32 位指令	DBCD (P)
执行类型	连续执行型、脉冲执行型		
指令格式	BCD (P) S D	DBCD (P) S D	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
S	—	—	○	○*1	○	—	—	—
D	—	—	○*2	○*1	○	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围	数据类型
-----	----	------	------

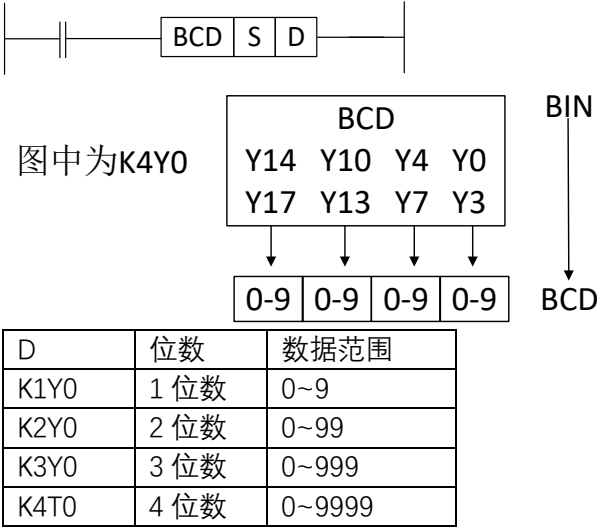
		16 位范围	32 位范围	16 位	32 位
S	保存转换源(2 进制数)数据的字软元件编号	0- 9999	0- 99999999	实数	实数
D	转换目标(10 进制数)的字软元件编号	- 32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(BCD、BCDP)

将 S 的 BIN(2 进制数)数据转换成 BCD(10 进制数)数据后传送到 中。

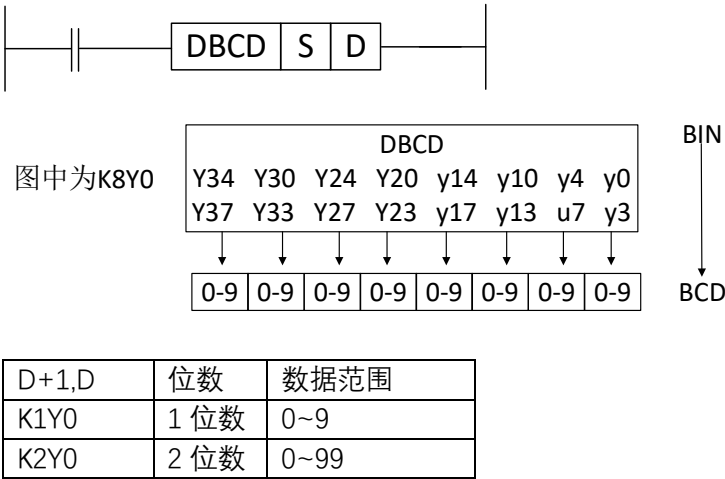
- S 的数据可以转换成 K0 ~ K9999 的 BCD(10 进制数)。
- S 和 D 指定位数的时候，参考下表。



2. 32 位运算(DBCD、DBCDP)

将[S+1, S]的 BIN(2 进制数)数据转换成 BCD(10 进制数)数据后传送到[+1,]中。

- [S+1, S]的数据，可以转换成 K0 ~ K99999999 的 BCD(10 进制数)。
- [S+1, S]和[D+1, D]指定位数的时候，参考下表。



K3Y0	3 位数	0~999
K4Y0	4 位数	0~9999
K5Y0	5 位数	0~99999
K6Y0	6 位数	0~999999
K7Y0	7 位数	0~9999999
K8Y0	8 位数	0~99999999

报错相关：待添加

注意：仅指令之前的运算结果为 OFF-ON 时执行

4.11.3 BIN 指令

概要

将 10 进制数(BCD)转换成 2 进制数(BIN)的指令。

在将数字式开关之类，以 BCD(10 进制数)设定的数值转换成可编程控制器的运算中可以处理的 BIN(2 进制数)数

据后读取的情况下，可以使用本指令。

指令格式

16 位指令	BIN(P)	32 位指令	DBIN (P)
执行类型	连续执行型、脉冲执行型		
指令格式	BIN (P) S D	DBIN (P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	○	—	—	—	—
D	—	—	○*2	○*1	○	—	—	—	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存转换源(10 进制数)数据的字软元件编号	0- 9999	0- 99999999	实数	实数
D	转换目标(2 进制数)的字软元件	— 32768~32767	— 2147483648~2147483647	实数	实数

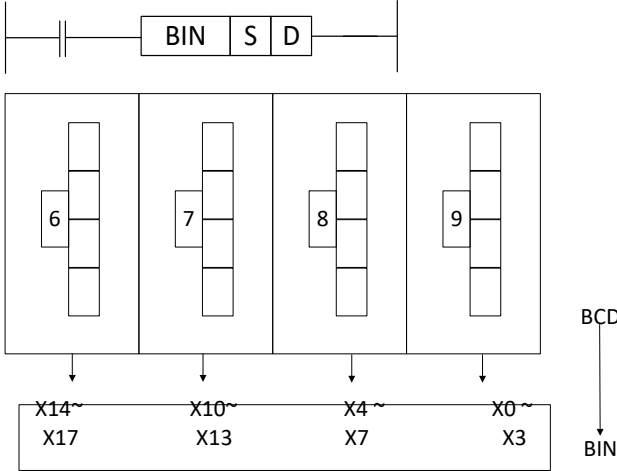
	编号				
--	----	--	--	--	--

示例

1. 16 位运算(BIN、BINP)

将 S 的 BCD(10 进制数)数据转换成 BIN(2 进制数)数据后传送到 中。

- S 的数据可以在 0~9999(BCD)的范围内转换。
- S 和 D 指定位数时，参考下表。

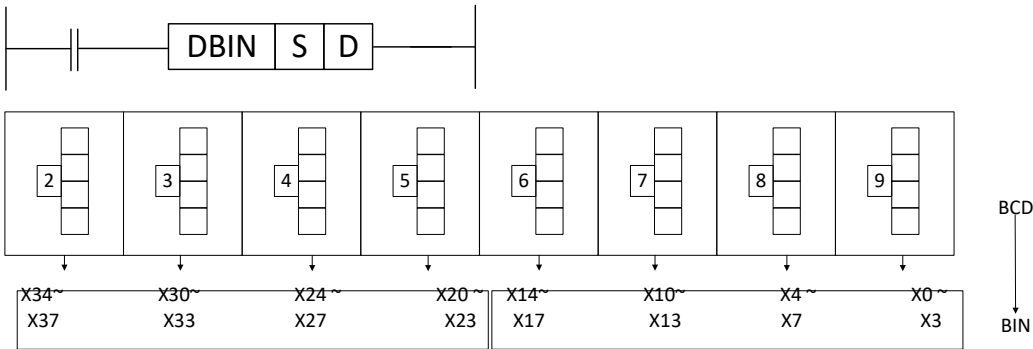


S	位数	数据范围
K1X0	1 位数	0~9
K2X0	2 位数	0~99
K3X0	3 位数	0~999
K4X0	4 位数	0~9999

2. 32 位运算(DBIN、DBINP)

将 [S+1, S] 的 BCD(10 进制数)数据转换成 BIN(2 进制数)数据后传送到 [+1,] 中。

- [S+1, S] 的数据，可以在 0~99999999(BCD)的范围内转换。
- [S+1, S] 和 [S+1, S] 指定位数时，参考下表。



S,S+1	位数	数据范围
K1X0	1 位数	0~9
K2X0	2 位数	0~99
K3X0	3 位数	0~999
K4X0	4 位数	0~9999

K5X0	5 位数	0~99999
K6X0	6 位数	0~999999
K7X0	7 位数	0~99999999
K8X0	8 位数	0~999999999

报错相关：待添加

注意：仅指令之前的运算结果为 OFF-ON 时执行

4.11.4 FLT/BIN 整数→2 进制浮点数转换

概要

将 BIN 整数值转换成 2 进制浮点数(实数)的指令。

指令格式

16 位指令	FLT(P)	32 位指令	DFLT (P)
执行类型	连续执行型、脉冲执行型		
指令格式	FLT (P) S D	DFLT(P)S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1*2	—	—	—	—	—
D	—	—	—	○*1*2	—	—	—	—	—

*1 支持变址修饰

*2 仅支持 D,R

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D1	保存 BIN 整数值的数据寄存器编号	-32768~32767	-2147483648~2147483647	实数	实数
D2	保存 2 进制浮点数(实数)的数据寄存器编号	-32768~32767	-2147483648~2147483647	实数	实数

示例

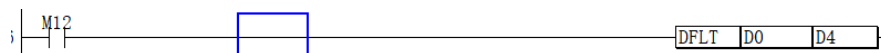
1. 16 位运算(FLT、FLTP)

将 D0 的 BIN 整数值数据转换成 2 进制浮点数(实数)值后，保存在[D3, D2]中。



2. 32 位运算(DFLT、DFLTP)

将[D1, D0]的 BIN 整数值数据转换成 2 进制浮点数(实数)值后, 保存在[D5, D4]中。



相关指令

指令	功能
INT	FLT 指令的逆转换动作指令

报错相关: (待添加)

注意: 1. 不需要常数(K、H)的浮点数转换

由于在各 2 进制浮点数(实数)运算指令中, 指定的 K、H 的值会自动转换成 2 进制浮点数(实数), 所以不需要使用 FLT 指令进行转换。

(RAD 指令、DEG 指令、EXP 指令、LOGE 指令等都不能指定 K、H。)

4.11.5 DEBCD/2 进制浮点数→10 进制浮点数的转换

概要

将软元件中的 2 进制浮点数转换成→10 进制浮点数的指令。

指令格式

16 位指令	-	32 位指令	DEBCD(P)
执行类型	连续执行型, 脉冲执行型		
指令格式	-	DEBCD (P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—

*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

S	保存 2 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	-	字符串
D	保存被转换的 10 进制浮点数数据的数据寄存器编号	-	0~32767	-	实数

示例

1. 32 位运算

将[S+1、 S]的 2 进制浮点数转换成 10 进制浮点数后，传送到[D+1， D]中。



报错相关：(待添加)

注意：(待添加)

4.11.6 DEBIN/10 进制浮点数→2 进制浮点数的转换

概要

将软元件中的 2 进制浮点数转换成→10 进制浮点数的指令。

指令格式

16 位指令	-	32 位指令	DEBIN(P)
执行类型	连续执行型，脉冲执行型		
指令格式	-	DEBIN (P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—

*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存 10 进制浮点数数据的数据寄存器编号	-	- 2147483648~2147483647	-	字符串
D	保存被转换的 2 进制浮点数数据的数据寄存器编号	-	0~32767	-	实数

示例

1.32 位运算



将[S+1, S]的 10 进制浮点数转换成 2 进制浮点数后，传送到[D +1, D]中。

报错相关：(待添加)

注意：(待添加)

4.11.7 DABIN/10 进制 ASCII→BIN 的转换

概要

将以 10 进制数字的 ASCII 码(30H ~ 39H)形式现实的数据转换成 BIN 数据的指令。

指令格式

16 位指令	DABIN	32 位指令	DDABIN
执行类型	连续执行型		
指令格式	DABIN S D	DDABIN S D	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
S1	—	—		○*1*2	—	—	—	—

S2	—	—	○*3	○*1*2	○	—	—	—	—
----	---	---	-----	-------	---	---	---	---	---

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnY、KnM、KnS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存转换结果的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
S2	保存转换结果的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

说明

将(s)中指定的软元件编号及以后中存储的10进制ASCII数据转换为BIN数据后，存储到(d)中指定的软元件中</Txt>

(s)要转换为BIN值的ASCII数据，或存储了ASCII数据的起始软元件</Txt>

(d)存储转换结果的起始软元件</Txt>

16位指令(s)~(s)+2中指定的ASCII数据，范围-32768~+32767</Txt>

[s]的低8位字节]中，要转换的数据为正时设定“20H(空格)”为负时设定“2DH(-)”

各位数的ASCII码为“30H”~“39H”的范围

各位数的 ASCII 码为“20H(空格)”、“00H(NULL)”时，作为“30H”处理

报错相关：（待添加）

注意：（待添加）

4.11.8 BINDA/BIN→10 进制 ASCII 的转换

概要

将 BIN 数据转换成 ASCII 码(30H ~ 39H)的指令

指令格式

16 位指令	BINDA	32 位指令	DBINDA
执行类型	连续执行型		
指令格式	BINDA S D	DBINDA S D	

操作数支持

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○*3	○*1*2	○	—	—	○	—
S2	—	—	—	○*1*2	—	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnY、KnM、KnS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存转换结果的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
S2	保存转换结果的软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

说明

将(s)中指定的 BIN 数据按 10 进制表示时的各位的数值转换为 ASCII 码后，存储到(d)中指定的软元件编号及以后中

(s)进行 ASCII 转换的 BIN 数据

(d)存储转换结果的软元件起始编号

16 位指令(s)范围-32768~+32767

(s)的低 8 位字节]中，BIN 数据为正时将存储 20H，为负时将存储 2DH

16 位指令对于(d)+3 中指定的软元件的数据的存储时，SM91(输出字符数切换信号)为 OFF 的情况下将存储 0，为 ON 的情况下不变化

报错相关：（待添加）

注意：（待添加）

4.11.9 WTOB/字节单位的数据分离

概要

将连续的 16 位数据按照字节(8 位)单位进行分离的指令。

指令格式

16 位指令	WTOB (P)	32 位指令	-
执行类型	连续执行型，脉冲执行型		
指令格式	WTOB (P) S D N	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—
N	—	—	—	○*1	—	—	—	○	—

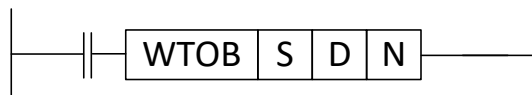
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存要按照字节单位进行分离的数据的软元件起始编号	-32768~32767	-	实数	-
D	保存已经按照字节单位分离的结果的软元件起始编号	0~32767	-	实数	-
N	要分离的字节数据个数 ($0 \leq n$)	0~32767	-	实数	-

示例

1. 16 位运算



将(s)中指定的软元件开始的(n)点的数据全部进行加法运算后，存储到(d)中指定的软元件中

报错相关：(待添加)

注意：(待添加)

4.11.10 BTOW/字节单位的数据结合概要

将 连续的 16 位数据的低 8 位(低字节)结合在一起的指令

指令格式

16 位指令	BTOW(P)	32 位指令	-
执行类型	连续执行型,脉冲执行型		
指令格式	BTOW(P) S D N	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—
N	—	—	—	○*1	—	—	—	○	—

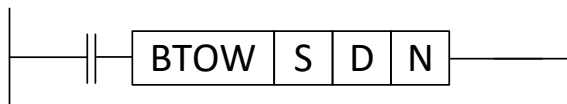
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存要按照字节单位结合的数据的软元件起始编号	-32768~32767	-	实数	—
D	保存已经按照字节单位结合的结果的软元件起始编号	0~32767	-	实数	-
N	要结合的字节数据个数 (0≤n)	0~32767	-	实数	-

示例

1. 16 位运算



将(s)中指定的软元件编号以后中存储的(n)字节的数据的低位8位，合并为字单位后，存储到(d)中指定的软元件编号以后

报错相关：（待添加）

注意：（待添加）

4.11.11 ASCI/ASCII 数据输入

概要

HEX→ASCII 转换指令

指令格式

16 位指令	ASCI	32 位指令	-
执行类型	连续执行型		
指令格式	ASCI S D N	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—
N				○*1				○	

*1 支持变址修饰

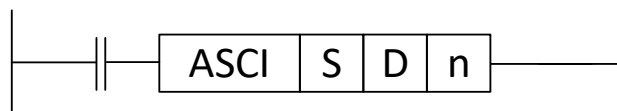
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	存储要转换的 HEX 代码的软元件起始编号	0~32767	-	实数	-
D	存储转换后的 ASCII 码的软元件起始编号	-	-	字符串	-

	号				
N	要转换的 HEX 代码的 字符数(位数)	1~32767	-	实数	实数

示例

1. HEX→ASCII 转换



将 S 中指定的半角、英文、数字字符串转换成 ASCII 码后，依次传送到 D 中

报错相关：（待添加）

注意：ASCII 指令在转换时使用的模式有 16 位模式和 8 位模式

16 位转换模式 (SM161=OFF)，8 位转换模式 (SM161=ON)

4.11.12 HEX /ASCII→HEX 的转换

概要

这个指令是将 ASCII 码转换成 HEX 的指令

指令格式

16 位指令	HEX	32 位指令	-
执行类型	连续执行型		
指令格式	HEX S D N	-	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z		LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	—	○	—
D	—	—	○	○*1	—	—	—	—	—	—
N				○*1					○	

*1 支持变址修饰

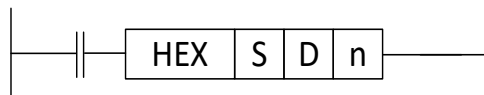
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

S	保存要转换的 ASCII 码的软元件的起始编号	0~32767	-	实数	-
D	保存转换后的 HEX 的软元件的起始编号	-	-	字符串	-
N	要转换的 ASCII 码的字符数(字节数)	1~256	-	实数	实数

示例

1. ASCII→HEX 转换



将 S 开始的软元件中保存的 ASCII 码的 n 个字符转换成 HEX 代码，然后保存到 D 开始的软元件中

报错相关：（待添加）

注意： HEXA指令在转换时使用的模式有16位转换模式与8位转换模式

16位转换模式(SM161=OFF)，8位转换模式(SM161=ON)

4.11.13 ESTR/2 进制浮点数→字符串的转换

概要

该指令用于将 2 进制浮点数数据转换成指定位数的字符串(ASCII 码)。

指令格式

16 位指令	ESTR(P)	32 位指令	DESTR(P)
执行类型	连续执行型，脉冲执行型		
指令格式	ESTR(P) S1 S2 D		DESTR(P) S1 S2 D

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	—	○
S2	—	—	—	○*1	—	—	—	—	—

D	—	—	—	○*1	—	—	—	—	—
---	---	---	---	-----	---	---	---	---	---

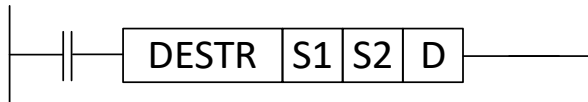
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	要转换的 2 进制浮点数据，或是保存数据的软元件的起始编号	-	- 2147483648~2147483647	-	实数
S2	保存要转换数值的显示指定的软元件起始编号	-	0~32767	-	实数
D	保存已转换的字符串的软元件起始编号	-	0~32767	-	字符串

示例

1.32 位 2 进制浮点数转字符串



将(s1+1, s1)中指定的单精度实数数据，按照(s2, s2+1, s2+2)中指定的内容转换为字符串后，存储到(d)开始的软元件中

可直接指定实数到(s1)

(s1)转换的单精度实数数据或存储了数据的软元件起始编号, 范围0、2~126 <

(s1) | < 2+128, 单精度实数

(s2)存储了转换数值的显示指定的软元件起始编号, 有符号BIN16位

(s2)中指定的软元件作为起始，用到(s2)+2

根据(s2)中指定的显示指定转换后的数据有所不同

(s2)值为0: 小数点形式

(s2)值为1: 指数形式

(s2+1)值: 全部位数，可在2~24的范围内设置

(s2+2)值: 小数部分位数

(d)存储转换后的字符串的软元件起始编号，数据类型字符串

报错相关: (待添加)

注意: (待添加)

4.11.14 EVAL /字符串→2 进制浮点数的转换

概要

用于将字符串(ASCII 码)转换成 2 进制浮点数数据的指令。

指令格式

16 位指令	EVAL(P)	32 位指令	DEVAL(P)
执行类型	连续执行型，脉冲执行型		
指令格式	EVAL (P) S D	DEVAL(P) S D	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S	—	—	—	○*1	—	—	—	—	—	
D	—	—	—	○*1	—	—	—	—	—	

*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存要转换成 2 进制浮点数数据的字符串数据的软元件的起始编号	0~32767	- 2147483648~2147483647	字符串	字符串
D	保存已转换的 2 进制浮点数数据的软元件的起始编号	0~32767	0~32767	实数	实数

示例

1.32 位运算



将(s)开始的软元件中存储的字符串转换为单精度实数后，存储到(d+1，d)中指定的软元件中

指定的字符串无论是小数点形式，还是指数形式，都可以转换成2进制浮点数数据

(d)存储转换后单精度实数的软元件起始编号，数据类型单精度实数

(s)中指定的字符串中，对于进行单精度实数转换的字符串，除去符号、小数点、指数部后的6位数有效，第7位数以后在转换时将被舍去

(s)中指定为2BH(+)或省略符号时，将作为正值进行转换。此外，将符号指定为2DH(-)时，将作为负值进行转换

(s)中指定的字符串中，最初的0以外的数值之间存在有20H(空白)或30H(0)的情况下，转换时将忽略20H、30H

在指数形式的字符串中，“E”与数值之间存储有30H(0)的情况下，转换时将忽略30H

字符串最多可设置24字符。字符串中的20H(空白)、30H(0)也作为1字符计数

报错相关：（待添加）

注意：（待添加）

4.12 数据传送

4.12.1 MOV 指令

概要

将软元件的内容传送(复制)到其他的软元件中的指令。

指令格式

16 位指令	MOV(P)	32 位指令	D MOV(P)
执行类型	连续执行型、脉冲执行型		
指令格式	MOV(P) S D	DMOV(P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	○	—
D	—	—	○	○*1	—	—	—	—	—

*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

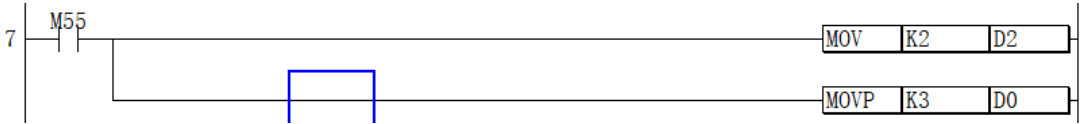
S	传送源的数据，或是保存数据的软元件编号	— 32768~32767	— 2147483648~2147483647	实数	实数
D	传送目标的软元件编号	— 32768~32767	— 2147483648~2147483647	实数	实数

示例

1.MOV, MOVP

将传送源的内容 1 点传送给传送目标。

- 指令输入为 OFF 时，传送目标不变化。
- 传送源中指定了常数(K)时，会自动执行 BIN 转换。



如图所示，M55 导通后，MOV 指令会一直将 K2 传递到 16 位的 D2 之中。MOVP 指令会将 K3 传递到 D0 中一次。

报错相关：(待添加)

注意：(待添加)

4.12.2 DEMOV/2 进制浮点数数据传送

概要

传送 2 进制浮点数数据的指令。

指令格式

16 位指令	DEMOV(P)	32 位指令	DEMOV (P)
执行类型	连续执行型，脉冲执行型		
指令格式	DEMOV (P) S D	DEMOV (P) S D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	○
D	—	—	—	○*1	—	—	—	—	—

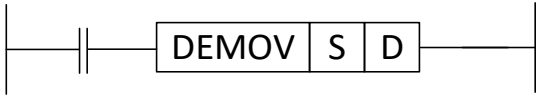
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	传送源的 2 进制浮点数数据，或是保存数据的软元件编号	-32768~32767	- 2147483648~2147483647	实数	-
D	保存 2 进制浮点数数据的软元件编号	0~32767	0~32767	实数	-

示例

1.32 位数据传送



将传送源[S+1S,]的内容(2 进制浮点数数据)传送到[D+1, D]中。此外，还可以在 中直接指定实数(E)。

报错相关：（待添加）

注意：（待添加）

4.12.3 SMOV 指令

概要

以位数为单位(4 位)进行数据分配/合成的指令。

指令格式

16 位指令	SMOV(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	SMOV(P) S M1 M2 D N	-	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
S	—	—	○	○*1	—	—	—	—

M1	—	—	—	—	—	—	—	○	—
M2	—	—	—	—	—	—	—	○	—
D	—	—	○*2	○*1	—	—	—	—	—
N	—	—	—	—	—	—	—	○	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存有要进行位移动的数据软元件的编号	— 32768~32767	—	实数	—
M1	要移动的起始位的位置	1~4	—		—
M2	要移动的位的个数	1~4	—		—
D	保存已经进行位移动的数据的软元件编号	— 32768~32767	—	实数	—
N	移动目标的起始位的位置	1~4	—		—

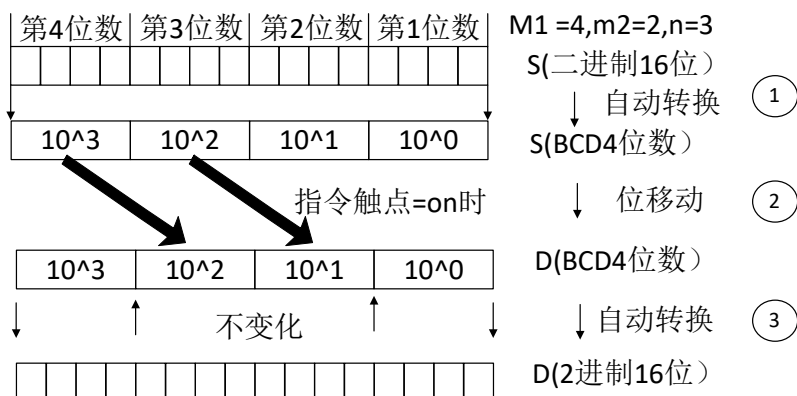
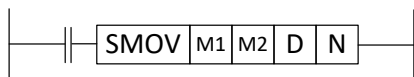
示例

1.16 位运算 (SMOV、SMOVP)

传送源 和传送目标 的内容转换(0000 ~ 9999)成 4 位数的 BCD，m1 位数起的低 m2 位数部分被传

送(合成)到传送目标 的 n 位数起始处，然后转换成 BIN，保存在传送目标 中。

- 指令输入为 OFF 时，传送目标 不变化。
- 指令输入为 ON 时，传送源 的数据以及传送目标 中的指定传送以外的位数不变化。



- 1) S 从 BIN 转换为 BCD
- 2) 从第 m1 位数起的低 m2 位数部分的数据，被传送（合成）到 D 的第 n 位数起始 m2 位数。D 的 10³ 位数，被 10⁰ 位数执行来自 S 的传送时不受任何影响。
- 3) 合成的数据（BCD）转换成 BIN 后，保存到 D 中。

2. 扩展功能

将 SM168 置 ON 后，执行 SMOV 指令时，则不能进行 BIN→BCD 转换。
位移动以 4 位为单位执行。

报错相关：（待添加）

注意：（待添加）

4.12.4 BMOV 指令

概要

对指定点数的多个数据进行成批传送(复制)。

指令格式

16 位指令	BMOV(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	BMOV(P) S D	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	—	—

D	—	—	○*2	○*1	—	—	—	—	—
N	—	—	—	—	—	—	—	○	—

*1 支持变址修饰

*2 不支持 KnX

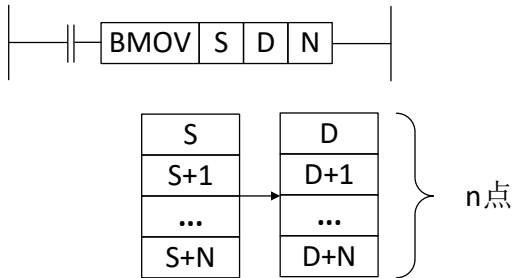
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	传送源的数据，或是保存数据的软元件编号	-32768~32767	-	实数	-
D	传送目标的软元件编号	-32768~32767	-	实数	-
N	传送点数	1~32767	-	实数	-

示例

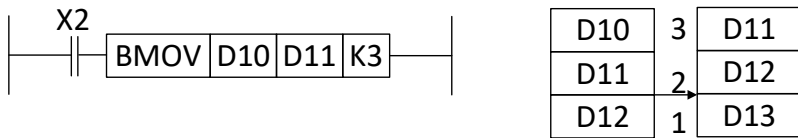
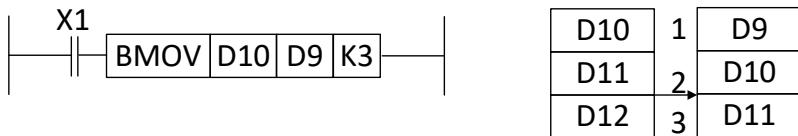
将开始的 n 点的数据成批传送到开始的 n 点中。

- 超出软元件编号范围时，在可能的范围内传送。



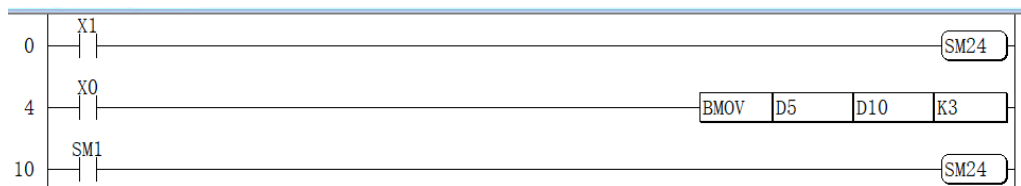
即使传送编号范围重叠也能传送

无论传送源的数据有无传送，为了防止数据源没有传送就被改写，采用编号重叠的方法，按 1~3 的顺序自动传送。



扩展功能

通过控制 BMOV(FNC 15)指令的方向反转标志位 SM24 *1，可以在 1 个程序中实现双向传送



BMOV 方向反转标志位	传送方向	S,D
SM24:OFF	S→D	D5→D10 D6→D11 D7→D12
SM24:ON	D←S	D5←D10 D6←D11 D7←D12

报错相关：待添加

注意：(s)、(d)两方均指定了位软元件的位数时，必须将(s)、(d)的位数设置为相同。

4.12.5 FMOV 指令

概要

将同一数据传送到指定点数的软元件中的进行多点传送指令。

指令格式

16 位指令	FMOV(P)	32 位指令	DFMOV(P)
执行类型	连续执行型、脉冲执行型		
指令格式	FMOV(P) S D	DFMOV(P) S D	

可使用软元件

	位		字			双字		常数	
操作数	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	—	—
D	—	—	○*2	○*1	—	—	—	—	—
N	—	—	—	—	—	—	—	○	—

*1 支持变址修饰

*2 不支持 KnX

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

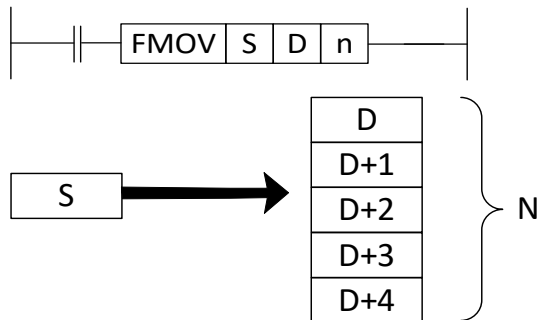
S	传送源的数据、或是保存数据的软元件编号	-32768~32767	-	实数	-
D	传送目标的起始字软元件编号(传送源的同一数据被成批传送。)	- 32768~32767	-	实数	-
N	传送点数	1~32767	-	实数	-

示例

1. 16 位运算(FMOV、FMOVP)

将 S 的内容传送到以 D 起始的 n 点的软元件中。

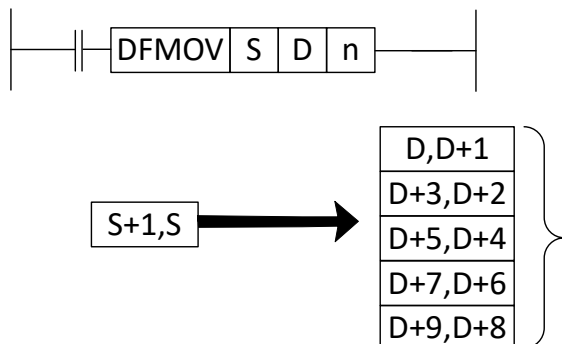
- n 点的软元件内容都相同。
- 以 n 指定的个数超出了软元件编号范围时，在可能的范围内传送。
- 指令输入为 OFF 时，传送目标 D 不变化。
- 指令输入为 ON 时，传送源 S 的数据不变化。
- 传送源 S 中指定了常数(K)时，会自动执行 BIN 转换。



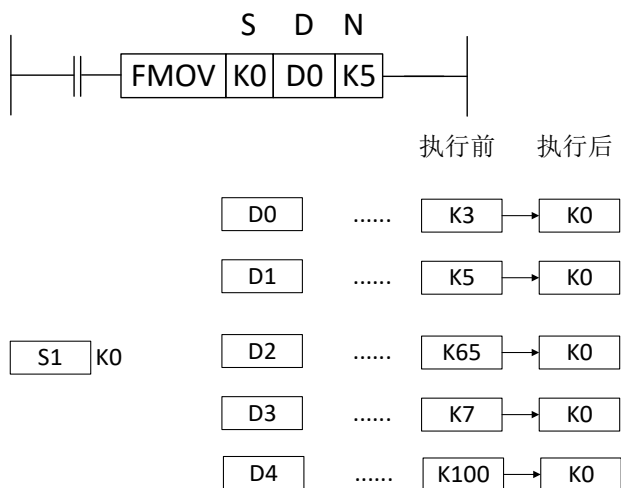
2. 32 位运算(DFMOV、DFMOVP)

将 [S+1, S] 的内容传送到以 [D+1, D] 起始的 n 点的 32 位软元件中。

- n 点的 32 位软元件内容都相同。
- 以 n 指定的个数超出了软元件编号范围时，在可能的范围内传送。
- 指令输入为 OFF 时，传送目标 [D+1, D] 不变化。
- 指令输入为 ON 时，传送源 [S+1, S] 的数据不变化。
- 传送源 [S+1, S] 中指定了常数(K)时，会自动执行 BIN 转换。



3. 指定数据多次写入



报错相关：待添加

注意：(n)中指定的值为0时，不会发生运算出错，而是变为无处理

4.12.6 HCMOV:高速当前值传送指令

概要

以高速计数器/脉冲宽度测定/PWM/定位用特殊寄存器为对象，进行读取或写入(更新)操作时使用该指令。

指令格式

16 位指令	HCMOV	32 位指令	DHCMOV
执行类型	连续执行型		
指令格式	HCMOV S D N		DHCMOV S D N

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—*1	—	—	○	○	—	—	○	—
D	—*1	—	—	○	○	—	—	○	—
N	○	—	—	○	○	—	—	○	—

*1 后续可能会追加 SM

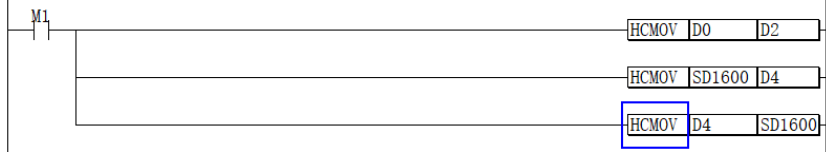
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	传送源的 软 元 件 编 号	-32768~32767	- 2147483648~2147483647	实数	实数
D	传 送 目 标	-32768~32767	-	实数	实数

	软元件编号		2147483648~2147483647		
N	相对目标地址	-32768~32767	- 2147483648~2147483647	实数	实数

示例

将(s)中指定的软元件值传送至(d)中指定的软元件。



如图所示 HCMMOV 指令可以将普通寄存器的值传送到普通寄存器中，也可将高速计数器置软元件中的值传递到普通寄存器中，也可以将普通寄存器的值传递到高速计数器值软元件中。除此之外，HCMOV 指令也可以在轴运动停止时，读取或者修改轴当前地址的值。

报错相关：(待添加)

注意: 此指令是立即执行的，当梯形图扫到此处会立即执行此指令而不是在 End 行执行此指令（高速计数相关特殊软元件请查阅 R 系列应用篇之软元件）

4.13 循环，移位

4.13.1 ROR、ROL 指令

概要

将数据进行右旋或者左旋

指令格式

执行类型	连续执行型.脉冲执行型
指令格式	ROR (P) /ROL (P) D N

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D	—	—	○	○*1	○	—	—	—	—
N	—	—	○	○*1	○	—	—	○	—

*1

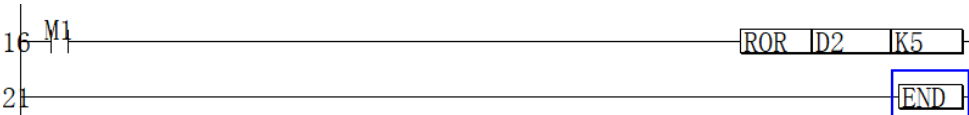
支持变址修饰

操作数	内容	数据范围	数据类型
-----	----	------	------

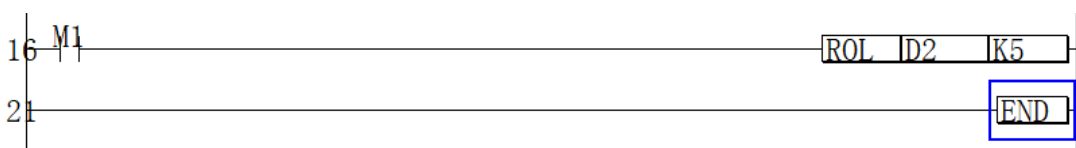
		16 位范围	32 位范围	16 位	32 位
D	旋转的软元件起始编号	-	-	位	-
N1	旋转的次数	0~15	-	实数	-

示例

将 D2 中的数据在不包含标志位的情况下继续 5 位右移



将 D2 中的数据在不包含标志位的情况下继续 5 位左移



报错相关：（待添加）

注意：（待添加）

4.13.2 SFTR、SFTL 指令

概要

将指定位长度的位数据进行右移或者左移

指令格式

执行类型	连续执行型.脉冲执行型
指令格式	SFTR (P) /SFTL (P) S D N1 N2

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	○	○	○	○*1	—	—	—	○	—
D	○	○	○	○*1	—	—	—	—	—
N1	—	—	○	○*1	○	—	—	○	—
N2	—	—	○	○*1	○	—	—	○	—

*1

支持变址修饰

可使用操作数

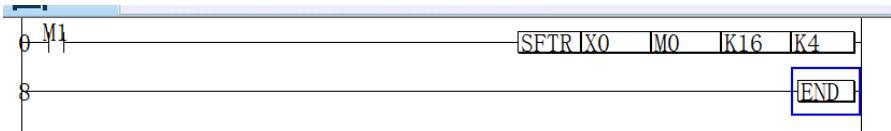
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

S	储存移位结果的软元件起始编号	-	-	位	-
D	移位的软元件起始编号	-	-	位	-
N1	移位数的数据长	0~32767	-	实数	-
N2	移位数	0~32767	-	实数	-

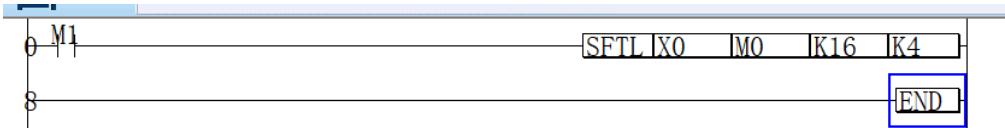
注：N1>N2

示例

将 M0 开始 16 位的数据向右移动 4 位，将 X0 开始的 4 点传送到 M12



将 M0 开始 16 位的数据向左移动 4 位，将 X0 开始的 4 点传送到 M12



报错相关：(待添加)

注意：(待添加)

4.13.3 WSFR、WSFL 指令

概要

将指定的字右移或者左移

指令格式

执行类型	连续执行型.脉冲执行型
指令格式	WSFR (P) / WSFL (P) S D N1 N2

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	○	—
D	—	—	○	○*1	—	—	—	—	—
N1	—	—	○	○*1	○	—	—	○	—
N2	—	—	○	○*1	○	—	—	○	—

*1

支持变址修饰

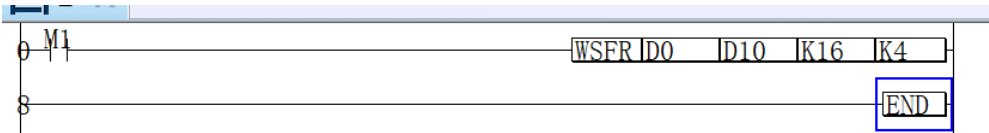
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	储存移位结果的软元件起始编号	-	-	位	-
D	移位的软元件起始编号	-	-	位	-
N1	移位数的数据长	0~32767	-	实数	-
N2	移位数	0~32767	-	实数	-

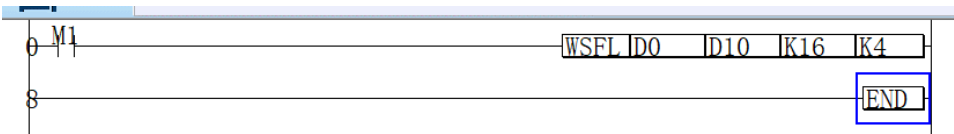
注：N1>N2

示例

将 D10 开始 16 位的字数据向右移动 4 位，将 X0 开始的 4 点传送到 D12



将 D10 开始 16 位的数据向左移动 4 位，将 X0 开始的 4 点传送到 D12



报错相关：(待添加)

注意：(待添加)

4.14 数据处理

4.14.1 SORT 排序指令

概要

该指令是用于将数据(行)和群数据(列)构成的数据表格，以指定的群数据(列)为标准，按照行单位将数据表格重新升序排列。在这个指令中，群数据(列)被保存在连续的软元件中。此外，数据(行方向)被保存在连续的软元件中。还有便于增加数据(行)，支持升序/降序排列的 SORT2 指令。

指令格式

16 位指令	SORT	32 位指令	-
执行类型	连续执行型		
指令格式	SORT S M1 M2 D N	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
M1	—	—	—	—	—	—	—	○	—
M2	—	—	—	—	—	—	—	○	—
D	—	—	—	○*1	—	—	—	—	—
N	—	—	—	○*1	—	—	—	○	—

*1 仅支持 DR

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存数据表格的软元件 起 始 编 号 [占 用 m1×m2 点]	-32768~32767	-	实数	-
M1	数据(行)数 [1~32]	-32768~32767	-	实数	-
M2	群数据(列) 数[1~6]	-32768~32767	-		-
D	保存运算结果 的软元件 起 始 编 号 [占 用 m1×m2 点]	-32768~32767	-	实数	-
N	作为排序标 准的群数据 (列)的列编 号[1~m2]	-32768~32767	-	实数	-

示例

1. 16 位运算(SORT)

在 S 开始的(m1×m2)点的数据表格(排序前)中，以 n 列的群数据为标准，按照升序重新排列数据行，然后保存在从 D 开始的(m1×m2)点的数据表格(排序后)中。

SORT S M1 M2 D N

• 下面例举排序前 m1=K3、m2=K4 的例子说明数据表格的结构。如是排序后的数据表格，请将 S 改读成 D

列号		群数 m2 个(m2=K4 时)			
		1	2	3	4
行号		管理编号	身高	体重	年龄
数据数 m1=3	1	S	S+3	S+6	S+9
	2	S+1	S+4	S+7	S+10

的情况	3	S+2	S+5	S+8	S+11
-----	---	-----	-----	-----	------

• 指令输入为 ON 时开始数据排列，m1 个扫描后数据排列结束，指令执行结束标志位 SM29 为 ON。

2. 动作举例

在“n=K2(列号 2)”和“n=K3(列号 3)”中执行下述的排序前的数据后，会如下所示动作。

此外，如果先在第 1 列中输入管理编号等的连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前数据

列号		群数 m2 个(m2=K4 时)			
行号		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=5 的情况+	1	S	S+5	S+10	S+15
		1	150	45	20
	2	S+1	S+6	S+11	S+16
		2	180	50	40
	3	S+2	S+7	S+12	S+17
		3	160	70	30
	4	S+3	S+8	S+13	S+18
		4	100	20	8
	5	S+4	S+9	S+14	S+19
		5	150	50	45

1) 以 n=K2(列号 2)执行指令时的排序结果

列号					
行号		1	2	3	4
		管理编号	身高	体重	年龄
	1	D	D+5	D+10	D+15
		4	100	20	8
	2	D+1	D+6	D+11	D+16
		1	150	45	20
	3	D+2	D+7	D+12	D+17
		5	150	50	45
	4	D+3	D+8	D+13	D+18
		3	160	70	30
	5	D+4	D+9	D+14	D+19
		2	180	50	40

相关软元件

指令	名称	内容
SM29	指令执行结束	数据排列结束时位 ON。

注意要点

• 动作过程中，请勿使操作数和数据的内容变化。

- 再次执行时，请将指令输入 OFF 一次。
- 指令的使用次数的限制
程序中仅可以使用 1 次。
- S 和 D 中指定了同一个软元件时
原来的数据按照排序后的数据顺序被改写。
到执行结束为止，请特别注意不要改变 S 的内容。

错误相关：待定

4.14.2 SORT2 数据排序

概要

以指定的群数据(列)为基准，以行为单位，将由数据(行)和群数据(列)构成的数据表进行升序/降序重新排序的指令。在这个指令中，由于是在连续的软元件中保存数据(行方向)，所以便于增加数据(行)。

此外，还有仅支持升序排列，数据结构(以列方向中连续的软元件构成数据)不同的 SORT 指令。

指令格式

16 位指令	SORT2	32 位指令	DSORT2
执行类型	连续执行型		
指令格式	SORT2 S M1 M2 D N	DSORT2 S M1 M2 D N	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
M1	—	—	—	○*1	○	—	—	○	—
M2	—	—	—	○*1	○			○	—
D	—	—	—	○*1	—	—	—	—	—
N	—	—	—	○*1	○	—	—	○	—

*1 仅支持 DR

可使用操作数

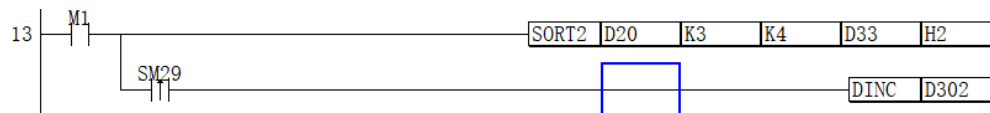
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存数据表格的软元件起始编号[占用 m1×m2 点]	-32768~32767	-2147483648~2147483647	实数	实数

M1	数据(行)数	1~32	- 2147483648~2147483647	实数	实数
M2	群数据(列)数	1~6	- 2147483648~2147483647	实数	实数
D	保存运算结果的软元件起始编号[占用m1×m2点]	-32768~32767	- 2147483648~2147483647	实数	实数
N	作为排序标准的群数据(列)的列编号[1~m2]	0~32767	0~32767	实数	实数

示例

1. 16 位运算(SORT2)

- 将(s)中指定的(n1×n2)点的 BIN16 位数据表(排序源)，以(n3)列的组数据为基准，对数据行进行升序或降序排序，并存储到(d)中指定的(n1×n2)点的 BIN16 位数据表(排序后)中。



下面例举排序前 m1=K3、m2=K4 的例子说明数据表格的结构。在排序后的数据表格中，请将 S 改读成 D。

列号		群数 m2 个(m2=K4 时)			
		1	2	3	4
行号		管理编号	身高	体重	年龄
数据数 m1=3 的情况	1	S+1,S	S+3,S+2	S+5,S+4	S+7,S+6
	2	S+9,S+8	S+11,S+10	S+13,S+12	S+15,S+14
	3	S+17,S+16	S+19,S+18	S+21,S+20	S+23,S+22

- 通过 SM165 的 ON/OFF 状态来设定排序。

	设定排序的
SM165=ON	降序
SM165=OFF	升序

- 在 m1 中使用数据寄存器 D 或扩展寄存器 R 时，为 32 位长度的数据。

例如，在 D0 中指定 m1 时，m1 为[D1, D0]的 32 位数据。

- 指令输入为 ON 时开始数据排列，m1 个扫描后数据排列结束，指令执行结束标志位 SM29 为 ON。

3. 动作举例

在“n=K2(列号 2)”和“n=K3(列号 3)”的情况下，对如下所示的排序前的数据进行排序，动作如下所示。

下面例举了 16 位运算的动作例子。执行 32 位运算时，请使用 BIN32 位构成数据表格。

此外，如果先在第 1 列中输入管理编号等连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前数据

列号	群数 m2 个(m2=K4 时)
----	------------------

行号		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=5 的情况+	1	S	S+1	S+2	S+3
		1	150	45	20
	2	S+4	S+5	S+6	S+7
		2	180	50	40
	3	S+8	S+9	S+10	S+11
		3	160	70	30
	4	S+12	S+13	S+14	S+15
		4	100	20	8
	5	S+16	S+17	S+18	S+19
		5	150	50	45

1) 以 n=K2(列号 2)为基准执行指令时的排序结果(升序的情况)

列号					
		1	2	3	4
行号		管理编号	身高	体重	年龄
	1	D	D+1	D+2	D+3
		4	100	20	8
	2	D+4	D+5	D+6	D+7
		1	150	45	20
	3	D+8	D+9	D+10	D+11
		5	150	50	45
	4	D+12	D+13	D+14	D+15
		3	160	70	30
	5	D+16	D+17	D+18	D+19
		2	180	50	40

2) 以 n=K3(列号 3)为基准执行指令时的排序结果(降序的情况)

列号					
		1	2	3	4
行号		管理编号	身高	体重	年龄
	1	D	D+1	D+2	D+3
		3	160	70	30
	2	D+4	D+5	D+6	D+7
		2	180	50	40
	3	D+8	D+9	D+10	D+11
		5	150	50	45
	4	D+12	D+13	D+14	D+15
		1	150	45	20
	5	D+16	D+17	D+18	D+19
		4	100	20	40

相关软元件

指令	名称	内容
SM29	指令执行结束	数据排列结束时位 ON。
SM165	降序排序	SM165=ON 时，降序排列。 SM165=OFF 时，升序排列。

报错相关：错误代码由开发人员定义

s)中指定的软元件范围超出相应软元件的范围时。

(d)中指定的软元件范围超出相应软元件的范围时。

(n1)中指定的数值超出下述范围时。1 ~ 32

(n2)中指定的数值超出下述范围时。1 ~ 6

(n3)中指定的数值超出下述范围时。1 ~ (n2)

注意

- 动作中可以改变操作数及数据的内容，但是排序动作可能需要多个周期才能执行完毕。
- (s)和(d)中指定同一软元件的情况下，源数据改写为排序后的数据顺序。(源数据与目标数据所用软元件重复或部分重叠时正常运行且不报错)

• 在指令执行中时，改变操作数或数据的影响如下：

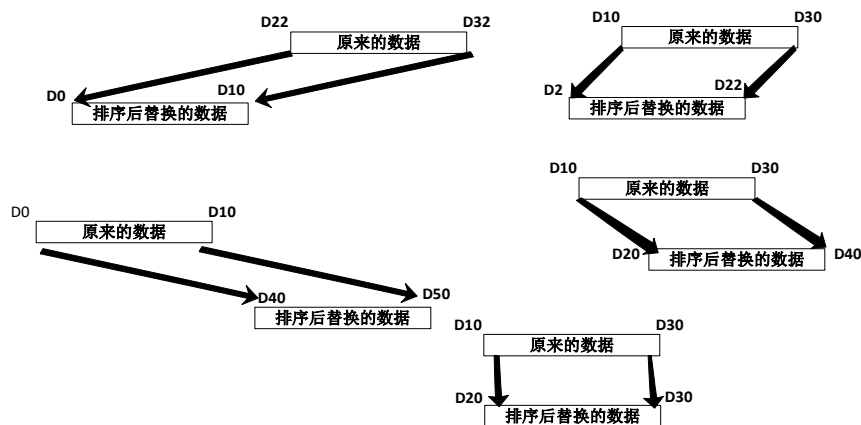
①改变参数 S 的数据源：在指令执行后第一个扫描时，会拷贝参数 S 的数据到参数 D 的位置。因此如果是拷贝后改变参数 S 中的数据，排序仍会按照改变前的老数据进行。

②改变参数 M1 数据（行）数：排序执行中，M1 参数被修改，接下来的排序会按照修改后的 M1（数据数）进行，整个排序动作正确性无法保证。

③改变参数 M2 数据（列）数：同 ②

④改变参数 D 数据：排序执行中，D 参数被修改，接下来会按照修改后的数据继续排序动作，排序后的数据将和数据源 S 的数据有差异。

⑤改变参数 N：排序执行中，n 被修改，接下来会按照修改后的 n 继续进行排序，排序动作执行的时间可能会超过 M1-1 个扫描。



4.14.3 WSUM/算出数据合计值

概要

将 2 进制浮点数，转换成可编程控制器中的一般数据形式 BIN 整数的指令

指令格式

16 位指令	WSUM(P)	32 位指令	DWSUM (P)
执行类型	连续执行型，脉冲执行型		
指令格式	WSUM (P) S D N	DWSUM (P) S D N	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—
N	—	—	—	○*1	—	—	—	○	—

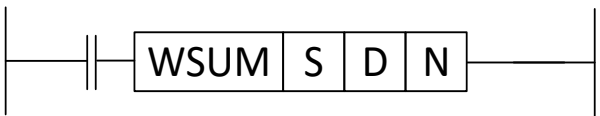
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存要算 出合计值 的数据的 软元件起 始编号	-32768~32767	- 2147483648~2147483647	实数	实数
D	保存合计 值的软元 件起始编 号	0~32767	0~32767	实数	实数
N	数据个数	0~32767	0~32767	实数	实数

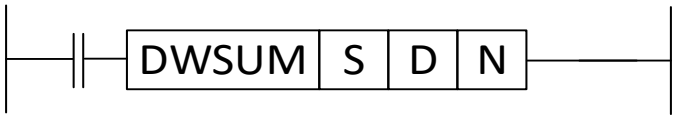
示例

1. 16 位运算



将(s)中指定的软元件开始的(n)点的数据全部进行加法运算后，存储到(d+1, d)中

2. 32 位运算



将(s)中指定的软元件开始的(n)点的数据全部进行加法运算后，存储到

(d+3, d+2, d+1, d) 中

报错相关：(待添加)

注意：(待添加)

4.14.4 MEAN/平均值

概要

求数据的平均值的指令。

指令格式

16 位指令	MEAN(P)	32 位指令	DMEAN (P)
执行类型	连续执行型、脉冲执行型		
指令格式	MEAN (P) S D N	DMEAN (P) S D N	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
S	—	—	○*1	○*3*4	—	—	—	—
D	—	—	○*2	○*3*4	—	—	—	—
N	—	—		○*5	—	—	○	—

*1 仅支持 KNX,KNY,KNM. KNS

*2 仅支持 KNY,KN.M. KNS

*3 支持变址修饰

*4 仅支持 TCDR

*5 仅支持 DR

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D1	成批复位的最前端的位/字软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数
D2	成批复位的末尾的位/字软元件编号	-32768~32767	- 2147483648~2147483647	实数	实数

示例

1. 16 位运算(MEAN、MEANP)

将 D0 开始的 2 个 16 位数据的平均值保存到 D12 中。

- 合计是求出代数和后被 n 除。
- 余数舍去。



$$(S+(S+1)+\cdots+(S+N-1))/N=D$$

2. 32 位运算(DMEAN、DMEANP)

将[D1, D0]开始的 2 个 32 位数据的平均值保存到[D13, D12]中。

- 合计是求出代数和后被 n 除。
- 余数舍去。



$$((S, S+1) + (S+3, S+2) + \cdots + (S+N*2-1, S+N*2-2))/N = D+1, D$$

报错相关:• 当 n 为 1 ~ 64 以外时, 会发生运算错误 (错误码待添加)

注意

- 软元件编号溢出时, 在可能的范围内将 n 作为较小的值处理。
- 将 D、R 指定为 32 位指令的 n 时, [n+1,n]的 32 位值便生效, 敬请注意。

DMEAN D0 D100 R0 时, 则 n=[R1, R0]。

4.14.4 SER/数据检索

概要

从数据表中检索相同数据、以及最大值、最小值的指令。

指令格式

16 位指令	SER(P)	32 位指令	DSER(P)
执行类型	连续执行型		
指令格式	SER(P) S1 S2 D N		DSER(P) S1 S2 D N

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	—	—
S2	—	—	—	○*1	—	—	—	○	—
D	—	—	—	○*1	—	—	—	—	—
N				○*1				○	

*1 支持变址修饰

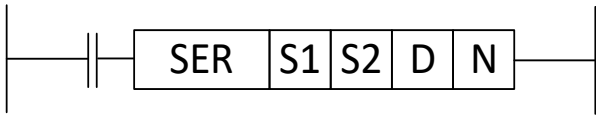
*2 仅支持 Y

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	存储了搜索相同数据、最大值、最小值的起始软元件编号	1~32767	1~32767	实数	实数
S2	存储了搜索相同数据的值或其存储目标软元件编号	1~32767	1~2147483647	实数	实数
D	存储了搜索相同数据、最大值、最小值后存储其个数的起始软元件编号	1~32767	1~32767	实数	实数
N	存储了搜索相同数据、最大值、最小值的个数	1~32767	1~32767	实数	实数

示例

1.16 位数据查找



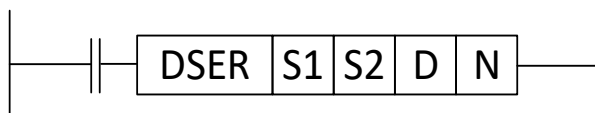
对于 (s1) 为起始的 (n) 个数据，搜索与 (s2) 的 BIN16 位数据相同的数据，将结果存储到 (d) ~ (d)+4 中

有相同数据的情况下，在 (d) 为起始的 5 点的软元件中，存储相同数据的个数、首次/最终位置及最大值、最小值的位置

无相同数据的情况下，在 (d) 为起始的 5 点的软元件中，存储相同数据的个数、首次/最终位置及最大值、最小值的位置

但是，在 (d) 为起始的 3 点的软元件 (相同数据的个数、首次/最终位置) 中，存储 0

2. 32 位数据查找



对于 (s1)+1、(s1) 为起始的 (n) 个数据，搜索与 (s2)+1、(s2) 的 BIN32 位数据相同的数据，将结果存储到 (d+1, d) ~ (d+9, d+8) 中

有相同数据的情况下，在 (d)+1、(d) 为起始的 BIN32 位数据 5 点的软元件中，存储相同数据的个数、首次/最终位置及最大值、最小值的位置

无相同数据的情况下，在 (d)+1、(d) 为起始的 BIN32 位 5 点的软元件中，存储相同数据的个数、首次/最终位置及最大值、最小值的位置

但是，在 (d)+1、(d) 为起始的 BIN32 位 3 点的软元件 (相同数据的个数、首次/最终位置) 中，存储 0

报错相关：待添加

注意：SERP 仅指令之前的运算结果为 OFF-ON 时执行

4.14.5 RND/产生随机数

概要

产生随机数的指令。

指令格式

16 位指令	RND(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	RND (P) D	-	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
D	—	—	○*3	○*1*2	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnY,KnM,KnS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	保存随机数的软元件起始编号	-32768~32767	—	实数	—

示例

1. 16 位运算(RND/RNDP)

这个指令产生 0 ~ 32767 的伪随机数，将其数值作为随机数保存到 D0 中。

在伪随机数系列中，每次计算出随机数的原始值，然后使用这随机数的原始值计算出伪随机数。



伪随机数的计算公式：（待添加）

报错相关：（待添加）

注意：（待添加）

4.14.6 MAX(P)整型数据最大值搜索

概要

从(s)中指定的软元件开始的(n)点的 BIN16/32 位数据中搜索最大值后，将最大值存储到(d)中指定的软元件中。

指令格式

(D)MAX 最大值搜索指令			
16 位指令	MAX	32 位指令	DMAX
执行类型	连续执行型、脉冲执行型		
指令格式	MAX(P) S D N	DMAX(P) S D N	

设置数据

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	查找最大值的软元件起始编号	——	——	——	——
D	存储最大值查找结果的软元件起始编号	——	——	——	——
N	查找数据数	1 ~ 65535	01 ~ 2147483647	16 位 (2 进制)	32 位 (2 进制)

*注：查找数据数超过最大值时按最大值为准

可使用软元件(MAX/DMAX)

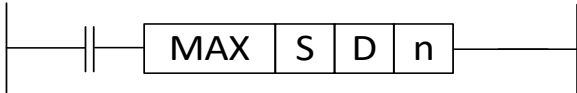
操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○ ^{*1}	—	—	—	—	—
D	—	—	—	○ ^{*1}	—	—	—	—	—
N	—	—	○	○ ^{*1}	○	—	—	○	—

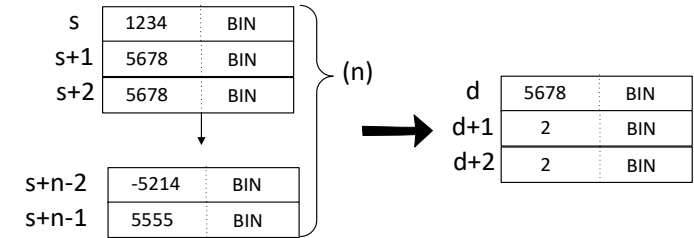
*1 支持变址修饰

示例

16 位运算(MAX (P))

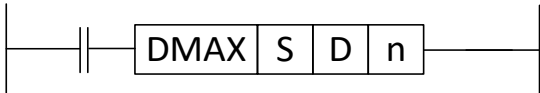


从(s)中指定的软元件开始的(n)点的 BIN16 位数据中搜索最大值后，将最大值存储到(d)中指定的软元件中。从(s)中指定的软元件中搜索，将存储了最先检测到的最大值的软元件编号是从(s)开始的第几点存储到(d)+1 中，将最大值的个数存储到(d)+2 中。

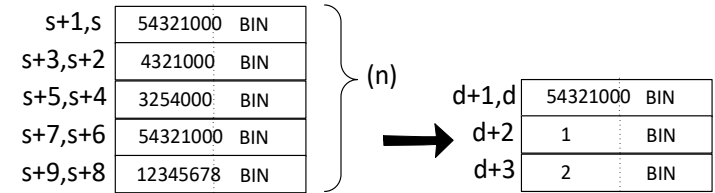


d:最大值
d+1:位置
d+2:个数

32 位运算(DMAX (P))



从(s)中指定的软元件开始的(n)点的 BIN32 位数据中搜索最大值后，将最大值存储到(d)、(d)+1 中指定的软元件中。从(s)中指定的软元件中搜索，将存储了最先检测到的最大值的软元件编号是从(s)开始的第几点存储到(d)+2 中，将最大值的个数存储到(d)+3 中。



d+1,d:最
大值
d+2:位置
d+3:个数

错误:

操作数 N < 0 时 (错误代码: 6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码: 6004)

注意事项：(n)中指定的值为 0 的情况下，将变为无处理。

4.14.7 MIN(P): 整型数据最小值搜索

概要

从(s)中指定的软元件开始至(n)点为止的BIN16/32 位数据中搜索最小值，将最小值存储到(d)中指定的软元件中

指令格式

MIN 最小值搜索指令			
16 位指令	MIN (P)	32 位指令	DMIN(P)
执行类型	连续执行型、脉冲执行型		
指令格式	MIN(P) S D N	DMIN(P) S D N	

设置数据

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
s	查找最小值的软元件起始编号	——	——	——	——
d	存储最小值查找结果的软元件起始编号	——	——	——	——
n	查找数据数	1-65535	1-2147483647	16 位 (2 进制)	32 位 (2 进制)

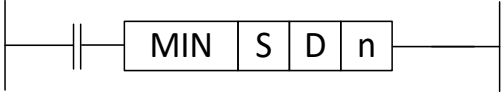
可使用软元件(MIN/DMIN*1)

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、KnSB、KnS	T、ST、C、D、W、SD、S W、R	Z		LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—	—
D	—	—	—	○*1	—	—	—	—	—	—
N	—	—	○	○*1	○	—	—	○	—	—

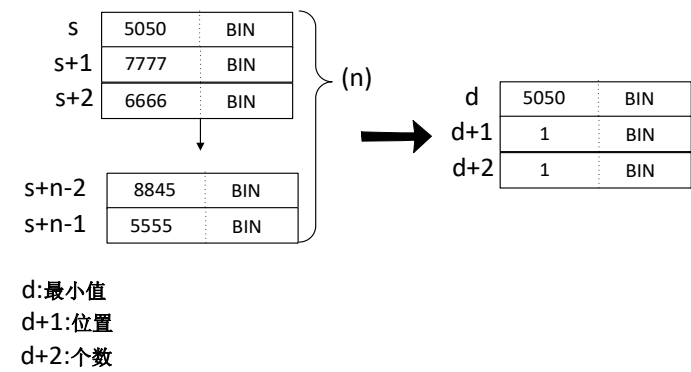
*1 支持变址修饰

示例

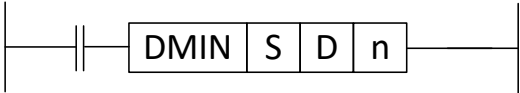
16 位运算 (MIN (P))



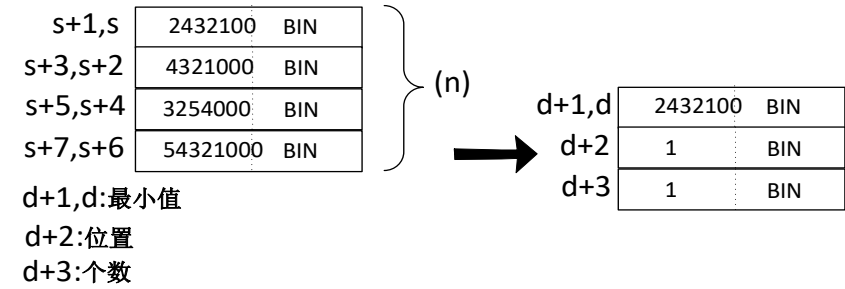
从(s)中指定的软元件开始至(n)点为止的 BIN16 位数据中搜索最小值，将最小值存储到(d)中指定的软元件中。从(s)中指定的软元件中搜索，将存储了最先检测到的最小值的软元件编号是从(s)开始的第几点存储到(d)+1 中，将最小值的个数存储到(d)+2 中



32 位运算(DMIN (P))



从(s)中指定的软元件开始至(n)点为止的 BIN32 位数据中搜索最小值，将最小值存储到(d)、(d)+1 中指定的软元件中。从(s)中指定的软元件中搜索，将存储了最先检测到最小值的软元件编号是从(s)开始的第几点存储到(d)+2 中，将最小值的个数存储到(d)+3 中



- 错误：
- 操作数 N <= 0。（错误代码： 6706）
 - 操作数（s）（d）中指定的软元件范围超出相应软元件的范围时。（错误代码： 6004）
 - 注意事项： 2.(n)中指定的值为 0 的情况下，将变为无处理。

4.14.8 DEMAX(P)单精度实数最大值搜索

概要

对从(s)开始(n)点的单精度实数的多个数据进行最大值搜索后，存储到(d)、(d)+1 中。将最先检测到最大值的数据是从(s)开始的第几点存储到(d)+2 中，将与最大值相同值的个数存储到(d)+3 中

指令格式

EMAX 最大值搜索指令			
16 位指令	EMAX	32 位指令	DEMAX
执行类型	连续执行型、脉冲执行型		
指令格式	EMAX(P) S D N	DEMAX(P) S D N	

设置数据

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	查找的数据	—	—	—	单精度实数
D	查找结果	—	—	—	单精度实数
N	查找数	1-65535	—	实数 (整数)	—

*注：查找数据数超过最大值时按最大值为准

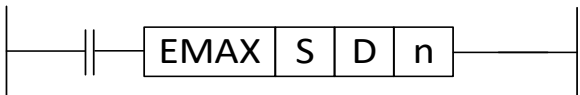
可使用软元件(EMAX/DEMAX)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○ ^{*1}	—	—	—	—	—
D	—	—	—	○ ^{*1}	—	—	—	—	—
N	—	—	○	○ ^{*1}	○	—	—	○	—

*1 支持变址修饰

示例

EMAX(P)

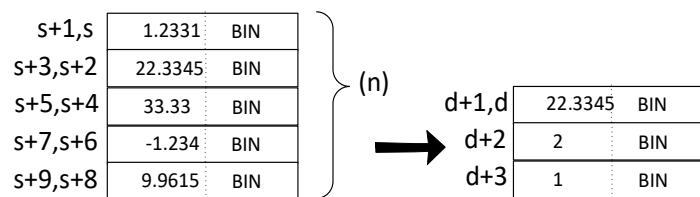


对从(s)开始(n)点的单精度实数的块数据进行最大值搜索后，将查找结果(最大值)存储到(d)中。将最先检测到最大值的数

据是从(s)开始的第几点存储到(d)中指定的查找结果(位置)中，将与最大值相同值的个数存储到(d)中指定的查找结果 (个数) 中

关于查找结果(位置)的值，将(s)中指定的块数据的起始作为第 1 点处理。

数) 中



d+1,d:最大值
 d+2:位置
 d+3:个数

d 中储存的值如下

	数据类型	内容
d	单精度实数	最大值
d+1		
d+2	16 位数据	最大值的位置
d+3	16 位数据	最大值的个数

错误：

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时

操作数 N < 0。(错误代码：6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意事项：

- 1.DEMAX 指令 和 EMAX 指令 是同一个指令
- 2.(n)中指定的值为 0 的情况下，将变为无处理。

4.14.9 DEMIN(P)单精度实数最小值搜索

概要

对从(s)开始(n)点的单精度实数的多个数据进行最小值搜索后，存储到(d)、(d)+1 中。
 将最先检测到最小值的数据是从(s)开始的第几点存储到(d)+2 中，将与最小值相同值的个数存储到(d)+3 中。

指令格式

(D)EMIN 搜索指令			
16 位指令	EMIN	32 位指令	DEMIN
执行类型	连续执行型、脉冲执行型		
指令格式	EMIN(P) S D N	DEMIN(P) S D N	

设置数据

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

S	查找的数据	—	—	—	单精度实数
D	查找结果	—	—	—	单精度实数
N	查找数	1-65535	—	实数 (整数)	—

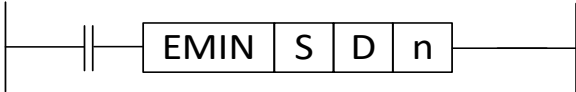
*注：查找数据数超过最大值时按最大值为准
可使用软元件(EMIN/DEMIN)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○ ^{*1}	—	—	—	—	—
D	—	—	—	○ ^{*1}	—	—	—	—	—
N	—	—	○	○ ^{*1}	○	—	—	○	—

*1 支持变址修饰

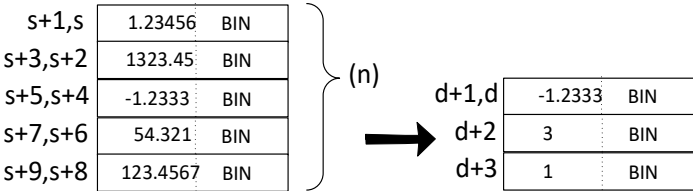
示例

EMIN(P)



对从(s)开始(n)点的单精度实数的块数据进行最小值搜索后，将查找结果(最小值)存储到(d)中。将最先检测到最小值的数据是从(s)开始的第几点存储到(d)中指定的查找结果(位置)中，将与最小值相同值的个数存储到(d)中指定的查找结果(个数)中。

关于查找结果(位置)的值，将(s)中指定的块数据的起始作为第1点处理。



d+1,d:最小值
d+2:位置
d+3:个数

数)中

d中储存的值如下

	数据类型	内容
d	单精度实数	最小值
d+1		
d+2	16 位数据	最小值的位置
d+3	16 位数据	最小值的个数

错误：

指定软元件的内容为非正规数、非数、 $\pm\infty$ 时

操作数 N < 0。(错误代码：6706)

操作数 (s) (d) 中指定的软元件范围超出相应软元件的范围时。(错误代码：6004)

注意事项：

- 1. DEMIN 指令 和 EMIN 指令 是同一个指令
- 2. (n)中指定的值为 0 的情况下，将变为无处理。

4.14.10 SWAP/高低字节互换

互换字数据的高 8 位和低 8 位的指令

指令格式

16 位指令	SWAP(P)	32 位指令	DSWAP(P)
执行类型	连续执行型,脉冲执行型		
指令格式	SWAP (P) S D N	DSWAP (P) S D N	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	—	○*1	—	—	—	—	—

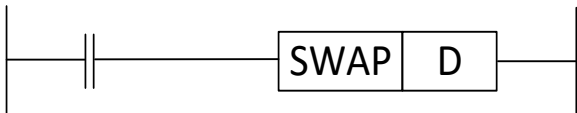
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	高低字节 互换的字 软元件	0~32767	0~32767	实数	实数

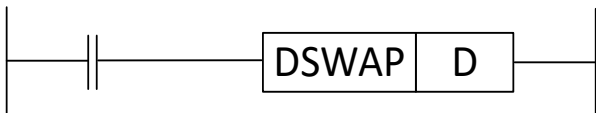
示例

1. 16 位运算



执行低8位和高8位的互换。

2. 32位运算



即使是32位指令，也是执行各自的低8位和高8位的互换

报错相关：(待添加)

注意：(待添加)

4.14.11 CRC/CRC 运算

概要

在通信等中被使用的错误校验方法之一为 CRC(Cyclic Redundancy Check:循环冗余校验)，用 CRC 指令计算出该 CRC 值。在错误校验的方法中，除了 CRC 以外还有奇偶校验以及和校验(校验和)，在求水平校验值时，可以使用 CCD 指令。

此外，在这个指令中，使用了生成 CRC 值(CRC-16)用的 $[X^{16} + X^{15} + X^2 + 1]$ 的生成多项式。

指令格式

16 位指令	CRC(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	CRC(P) S D	-	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
S	—	—	○*3	○*1*2	—	—	—	—
D	—	—	○*4	○*1*2	—	—	—	—
N				○*5			○	

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 KnX、KnY、KnM、KnS

*4 仅支持 KnY、KnM、KnS

*5 仅支持 D,R

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	保存作为 CRC 值生成对象的数据的软元件起始编号	-32768~32767	—	实数	—
D	保存被生成的 CRC 值的软元件编号	-32768~32767	-	实数	—

N	要计算CRC值的8位数据(字节)数,或是保存数据数的软元件编号	-32768~32767	-	实数	-
---	---------------------------------	--------------	---	----	---

示例

1.16 位运算

以 D0 中指定的软元件为起始的 D20 点 8 位数据(字节单位), 对其生成 CRC 值后保存到 D10 中。

在这个指令中有 8 位和 16 位的转换模式, 根据 SM161 的 ON/OFF 来切换转换模式。有关各自的动作请参考下一页以后的内容。

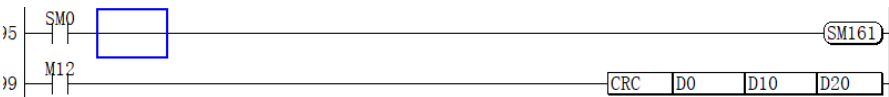
此外, 为了生成 CRC 值(CRC-16), 使用了 $[X^{16} + X^{15} + X^2 + 1]$ 的生成多项式。



6 位转换模式[SM161 = OFF]

在 16 位模式下, 对 D0 软元件的高 8 位(字节)和低 8 位(字节)进行运算。

在 D10 指定的 1 点软元件的 16 位中保存运算结果。

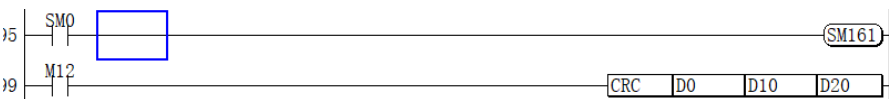


			例如 S =D100,D=D0,N=6		
			软元件	对象数据的内容	
				8 位	16 位
保 存 生 成 CRC 值的对 象 数据的地址	S	低字节	D100 低字节	01H	0301H
		高字节	D100 高字节	03H	
	S+1	低字节	D101 低字节	03H	0203H
		高字节	D101 高字节	02H	
	S+2	低字节	D102 低字节	00H	1400H
		高字节	D102 高字节	14H	
	S+…	…	-		
	S+N/2-1	低字节	-		
		高字节			
保存 CRC 值 的地址	D	低字节	D0 低字节	E4H	41E4H
		高字节	D0 高字节	41H	

8 位转换模式[SM161 = ON]

在 8 位转换模式下, 仅对 S 软元件的低 8 位(字节)执行运算。

计算结果使用 D 指定的软元件开始的 2 点, 在 D 中保存低 8 位(字节), 在 D+1 中保存高 8 位(字节)。



例如 S = D100,D=D0,N=6

			软元件	对象数据的内容
				8 位
保存生成 CRC 值的对象数据的地址	S	低字节	D100 低字节	01H
	S+1	高字节	D100 高字节	03H
	S+2	低字节	D101 低字节	03H
	S+3	高字节	D101 高字节	02H
	S+4	低字节	D102 低字节	00H
	S+5	高字节	D102 高字节	14H
	S+...	...	-	
	S+N-1	低字节	-	
保存 CRC 值的地址	D	低字节	D0 低字节	E4H
	D+1	低字节	D1 低字节	41H

2. 相关软元件

相关软元件	内容	
SM161	ON	CRC 指令在 8 位模式下动作
	OFF	CRC 指令在 16 位模式下动作

*1. 从 RUN→STOP 时清除

报错相关：(待添加)

注意: 在这个指令中, 使用 CRC 值(CRC-16)的生成多项式 $[X^{16} + X^{15} + X^2 + 1]$, 此外, 针对在 CRC 值, 还有各种标准化的生成多项式。请注意, 如果使用了不同的生成多项式, 会产生完全不同的 CRC 值。

参考: 主要的 CRC 值生成多项式

名称	生产多项式
CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC-16	$X^{16} + X^{15} + X^2 + 1$
CRC-32	$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
CRC-CCITT	$X^{16} + X^{12} + X^5 + 1$

4.14.12 CCD/校验码

概要

在通信等中使用的错误校验方法有水平校验以及和校验, 该指令是用于计算校验值的。

指令格式

16 位指令	CCD(P)	32 位指令	-
执行类型	连续执行型		
指令格式	CCD(P) S D N	-	

可使用软元件

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○*1	—	—	—	○	—
D	—	—	○	○*1	—	—	—	—	—
N				○*1				○	

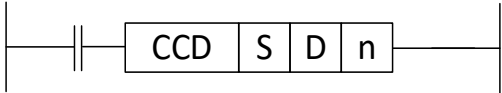
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	对象软元件的起始编号	0~32767	-	实数	-
D	保存计算出的数据的软元件的起始编号	0~32767	-	实数	-
N	数据数	1~256	-	实数	实数

示例

1. 16 位运算



计算出 S ~ S+n-1 中保存的数据的总和及水平校验，将总和数据保存在 D 中，将水平校验保存在 D+1 中

报错相关：（待添加）

注意： CCD指令在转换时使用的模式有16位转换模式与8位转换模式

16 位转换模式 (SM161=OFF)，8 位转换模式 (SM161=ON)

4.15 时钟运算

4.15.1 TADD/时钟数据加法运算

概要

将 2 个时间数据进行加法运算后，保存在字软元件中。

指令格式

--

16 位指令	TADD(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	TADD (P) S1 S2 D	-	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	—	○*1*2	—	—	—	—	—	
S2	—	—	—	○*1*2	—	—	—	—	—	
D	—	—	—	○*1*2	—	—	—	—	—	

*1 支持变址修饰

*2 仅支持 T,C,D,R

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	指定进行加法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	-32768~32767	—	实数	—
S2	指定进行加法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	-32768~32767	—	实数	—
D	保存两个时间数据(时、分、秒)加法运算的结果。(占用 3 点)	-32768~32767	—	实数	—

示例

1. 16 位运算(TADD)

将 [D0, D1, D2] 的时间数据(时、分、秒)与 [D6, D7, D8] 的时间数据(时、分、秒)进行加法运算, 其结果保存到 [D10, D11, D12](时、分、秒)中。



S1 (时)	+	S2 (时)	→	D (时)
S1+1 (分)		S2+1 (分)		D+1 (分)
S1+1 (秒)		S2+1 (秒)		D+2 (秒)

时：0~23

分：0~59

秒：0~59

- 当运算结果超出 24 小时时，进位标志位变为 ON，从单纯的加法运算值中减去 24 个小时后将该时间作为运算结果被保存。

- 运算结果为 0(0 时 0 分 0 秒)时，零位标志位变为 ON。

报错相关：(待添加)

注意: 1) 软元件的占用点数 S1、S2、D 各占用 3 点软元件。

请注意不要与机器其他控制中使用的软元件重复。

2) 使用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时,请使用 TRD(FNC 166)指令，读出特殊数据寄存器的值以后，在各个操作数中指定其字软元件。

4.15.2 TSUB/时钟数据减法运算

概要

将 2 个时间数据进行减法运算后，保存在字软元件中。

指令格式

16 位指令	TSUB(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	TSUB (P) S1 S2 D	-	

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H、E
S1	—	—	—	○*1*2	—	—	—	—
S2	—	—	—	○*1*2	—	—	—	—
D	—	—	—	○*1*2	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C,D,R

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位

S1	指定进行减法运算的时间数据(时、分、秒)的“时”。(占用3点)	-32768~32767	—	实数	—
S2	指定进行减法运算的时间数据(时、分、秒)的“时”。(占用3点)	-32768~32767	—	实数	—
D	保存两个时间数据(时、分、秒)减法运算的结果。(占用3点)	-32768~32767	—	实数	—

示例

1. 16 位运算(TSUB)

从[D0, D1, D2]的时间数据(时、分、秒)中减去[D6, D7, D8]的时间数据(时、分、秒), 其结果保存到[D10, D11, D12](时、分、秒)中。



S1 (时)	+	S2 (时)	→	D (时)
S1+1 (分)		S2+1 (分)		D+1 (分)
S1+2 (秒)		S2+2 (秒)		D+2 (秒)

时: 0~23

分: 0~59

秒: 0~59

当运算结果小于 0 时, 借位标志位变为 ON, 从单纯的减法运算值中加上 24 个小时后, 将该时间作为运算结果被保存。运算结果为 0(0 时 0 分 0 秒)时, 零位标志位变为 ON。

报错相关: (待添加)

注意:

1) 软元件的占用点数 S1、S2、S3 各占用 3 点软元件。

请注意不要与机器其他控制中使用的软元件重复。

3) 使用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时,请使用 TRD(FNC 166)指令, 读出特殊数据寄存器的值以后, 在各个操作数中指定其字软元件。

4.15.3 TRD/读出时钟数据

概要

读出可编程控制器内置实时时钟的时钟数据的指令。

指令格式

16 位指令	TRD(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	TRD (P) D	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D	—	—	—	○*1*2	—	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C,D,R

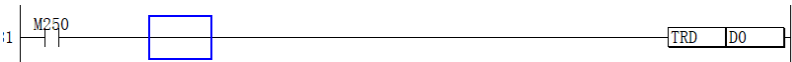
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	指定保存读出时间数据的起始软元件编号。 (占用 7 点)	-32768~32767	—	实数	—

示例

1. 16 位运算(TRD)

将可编程控制器内置实时时钟的时钟数据(SD13 ~ SD19)按照下面的格式读出到 D0 ~ D6 中。



软元件	功能	明细	可读性
SD13	秒	0~59 秒 (实时时钟用)	R
SD14	分	0~59 分 (实时时钟用)	R
SD15	时	0~23 小时 (实时时钟用)	R
SD16	日	1~31 日 (实时时钟用)	R
SD17	月	1~12 月 (实时时钟用)	R
SD18	年	西历 2 位数(0~99)	R

		(实时时钟用)	
SD19	星期	0(日)~6(六) (实时时钟用)	R

报错相关：(待添加)

注意:

1. D 软元件的占用点数占用 7 点软元件。

请注意不要与机器其他控制中使用的软元件重复。

4.15.4 TWR/写入时钟数据

概要

向可编程控制器内置实时时钟写入时钟数据的指令。

指令格式

16 位指令	TWR(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	TWR (P) S	-	

可使用软元件

	位		字		双字		常数	
操作数	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
S	—	—	—	○*1*2	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C,D,R

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	指定写入时间数据的源地址的起始软元件编号。(占用 7 点))	-32768~32767	—	实数	—

示例

将设定的时钟数据 ~ +6 写入可编程控制器内置实时时钟的时钟数据(SD13 ~ SD19)中。



软元件	功能	明细	可读性
-----	----	----	-----

SD13	秒	0~59 秒 (实时时钟用)	R
SD14	分	0~59 分 (实时时钟用)	R
SD15	时	0~23 小时 (实时时钟用)	R
SD16	日	1~31 日 (实时时钟用)	R
SD17	月	1~12 月 (实时时钟用)	R
SD18	年	西历 2 位数(0~99) (实时时钟用)	R
SD19	星期	0(日)~6(六) (实时时钟用)	R

– 执行 TWR(FNC 167)指令后，实时时钟的时钟数据即刻被更改。因此，请先将快几分钟的时钟数据传送到 D0 ~ D6 中，等到变成正确的时间时才执行指令。

– 使用这个指令设定时钟数据(时间校准)时，不需要控制特殊辅助继电器 SM15(时间停止以及时间校准)。

– 设定了不可能显示的日期时间数值时，不执行时钟数据的变更。此时，请设定正确的时钟数据后再次写入

报错相关：(待添加)

注意：

1. 软元件的占用点数

占用 S 开始的连续 7 点软元件。

请注意不要与机器其他控制中使用的软元件重复

4.15.5 HOUR/计时表

概要

以 1 个小时为单位，对输入触点持续 ON 的时间进行累加检测的指令。

指令格式

16 位指令	TWR(P)	32 位指令	DTWR(P)
执行类型	连续执行型、脉冲执行型		
指令格式	TWR (P) S	DTWR (P) S	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○*6	○*1*2	—	—	—	○	—

D1	—	—	—	○*1*3	—	—	—	—	—
D2	○*4	○*5	—	—	—	—	—	—	—

*1 支持变址修饰

*2 仅支持 T,C,D,R

*3 仅支持 D,R

*4 仅支持 Y,M,S

*5 仅支持 D,X

*6 仅支持 KNX,KNY,KNM,KNS

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	使 为 ON 的时间(以 1 个小时为单位设定)	-32768~32767	-2147483648~2147483647	实数	实数
D1	以 1 个小时为单位的当前值(指定停电保持用数据寄存器)	-32768~32767	-2147483648~2147483647	实数	实数
D2	报警输出的起始编号	-32768~32767	-2147483648~2147483647	实数	实数

示例

1. 16 位运算



当指令输入的累计 ON 时间超出了 D0 时，M2 变为 ON。D9 中不满 1 个小时的当前值，以 1 秒单位被保存。

S: 变 ON 为止的时间以 1 个小时为单位指定。

D1: 以 1 个小时为单位的当前值

D1+1: 不满 1 个小时的当前值(1 秒单位)

D2: 报警输出目标地址编号

当前值 D1 超出 S 的指定时间时，置 ON。

• 由于即使断开可编程控制器的电源后，也可以继续使用当前值数据，所以请在 D1 中指定停电保持用的数据寄存器。

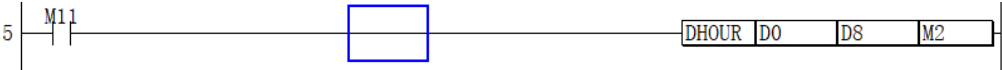
使用一般的数据寄存器时，由于可编程控制器的电源 OFF 和 STOP→RUN 的操作，当前值会被清除。

• 报警输出 D2 为 ON 以后，测量仍能继续。

• 当前值 D1 达到 16 位的最大值时停止测量。

要继续测量时，请清除 D1 ~ D1+1 的当前值。

2. 32 位运算



[S+1, S]: 变为 ON 为止的时间指定用,S1+1(高位), S1 (低位)指定。
[D1+1, D1]: 以 1 个小时为单位的当前值保存在,D1+1(高位), D1 (低位)中。
D1+2: 不满 1 个小时的当前值(1 秒单位)
D2: 报警输出的指定
当前值 D1, D1 +1 超出 S 的指定时间时, 置 ON。

- 由于即使断开可编程控制器的电源后, 也可以使用当前值数据, 所以请在 D1 中指定停电保持用的数据寄存器。
使用一般的数据寄存器时, 由于可编程控制器的电源 OFF 和 STOP→RUN 的操作, 当前值会被清除。
- 报警输出 D2 为 ON 以后, 测量仍能继续。
- 当前值[D1+1, D1]达到 32 位的最大值时停止测量。
要继续测量时, 请清除 D1 ~ D1 +2 的当前值。

报错相关: (待添加)
注意:
软元件的占用点数
占用 2 个(16 位运算)或者 3 个(32 位运算)软元件。
请注意不要与机器其他控制中使用的软元件重复。

4.16 数据表操作指令

4.16.1. SFWR 至数据表的数据写入

概要
用于先入先出及先入后出控制的数据写入指令。
指令格式

SFWR 从先入数据读取指令	
16 位指令	SFWR
执行类型	连续执行型、脉冲执行型
指令格式	SFWR(P) S D N

设置数据

操作数	内容	数据	
		16 位范围	16 位类型
S	存储了希望先入的数据的字软元件	-	有符号 16 位二进制数整数

	编号		
D	存储数据并移位的起始字软元件编号 (起始为指针, 数据从(d)+1 开始)	-	有符号 16 位二进制数整数
N	应指定存储的数据的点数+1 的值	2 ~ 32767	有符号 16 位二进制数整数

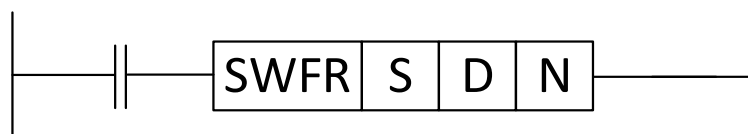
可使用软元件(SFWR)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○ ^{*1}	○	—	—	○	—
D	—	—	○	○ ^{*1}	○	—	—	—	—
n	—	—	○	○ ^{*1}	○	—	—	○	—

*1 支持变址修饰

示例

16 位运算(SFWR (P))

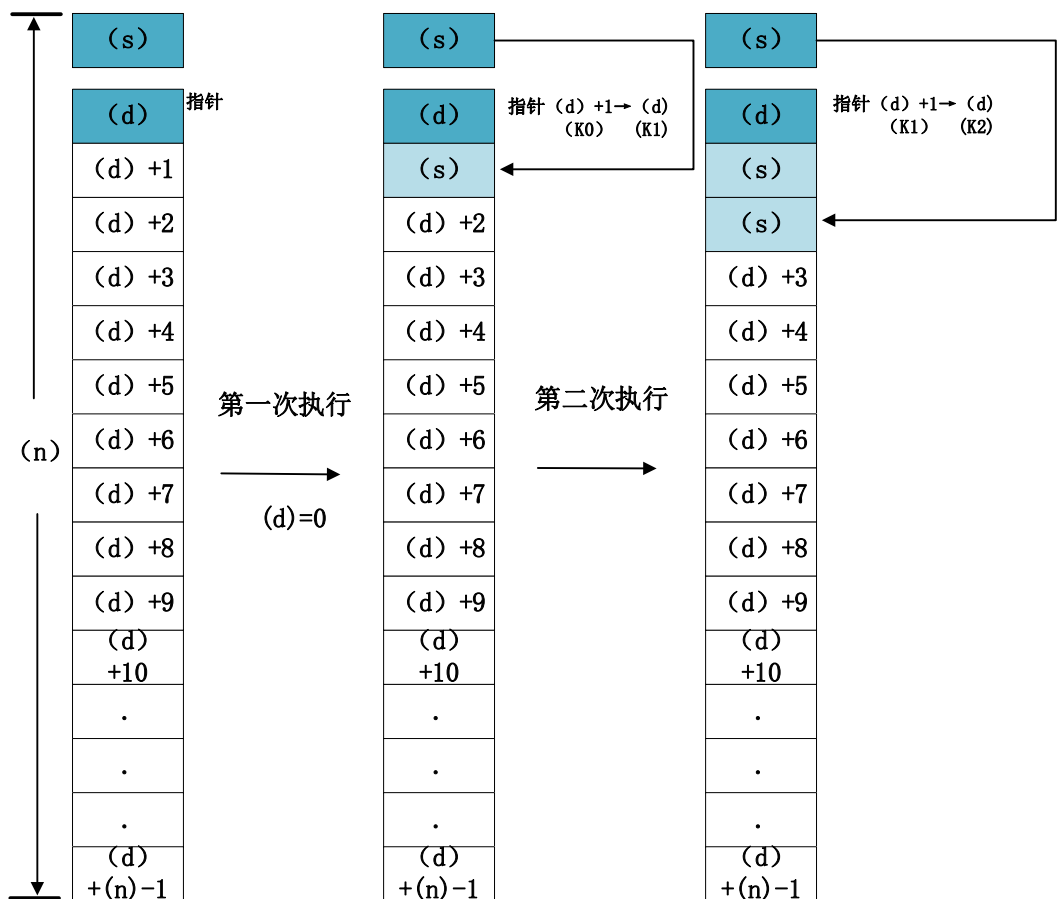


当指令执行时, 从(d)+1 开始, 将(s)的内容依次写入(n)-1 点, (d)的存储数据数+1。例如, (d)=0 的情况下写入到(d)+1, (d)=1 的情况下写入到(d)+2。

通过第 1 次执行, (s)的内容存储到(d)+1, 变为(s)的值。

如果更改(s)的内容后执行第 2 次, 该(s)的内容存储到(d)+2, (d)+2 的内容变为(s) (在连续执行型指令 SFWR 中每个运算周期

将依次存储, 因此应使用脉冲执行型指令 SFWRP 编程)。之后以此类推, 从右侧依次顺延填入, 数据的存储点数通过指针(d)的内容显示。



错误：

操作数 D 中指定了负值，或者 $N < 2$ 和 $N > 32767$ 时（错误代码：6706）

操作数 S, D, N 中指定的软元件范围超出相应软元件的范围时，或则从 D 开始 N 点的软元件点数超过相应软元件范围时。（错误代码：6004）

注意事项：连续执行型(SFWR)指令的情况下，每个扫描时间(运算周期)将依次存储(覆盖)，应加以注意。

4.16.2 SFRD 从数据表的先入数据读取

概要

用于先入先出控制的数据读取指令

指令格式

SFRD 从先入数据读取指令	
16 位指令	SFRD
执行类型	连续执行型、脉冲执行型
指令格式	SFRD(P) S D N

设置数据

操作数	内容	数据	
		16 位范围	16 位类型

S	存储了数据的起始字软元件编号 (起始为指针, 数据从(s)+1 开始)	-	有符号 16 位二进制数整数
D	存储先出的数据的字软元件编号	-	有符号 16 位二进制数整数
n	应指定为存储的数据点数+1 的值。+1 为指针。	2-32767	有符号 16 位二进制数整数

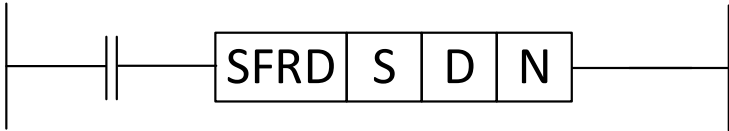
可使用软元件(SFRD)

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H
S	—	—	○	○ ^{*1}	○	—	—	—
D	—	—	○	○ ^{*1}	○	—	—	—
n	—	—	○	○ ^{*1}	○	—	—	○

*1 支持变址修饰

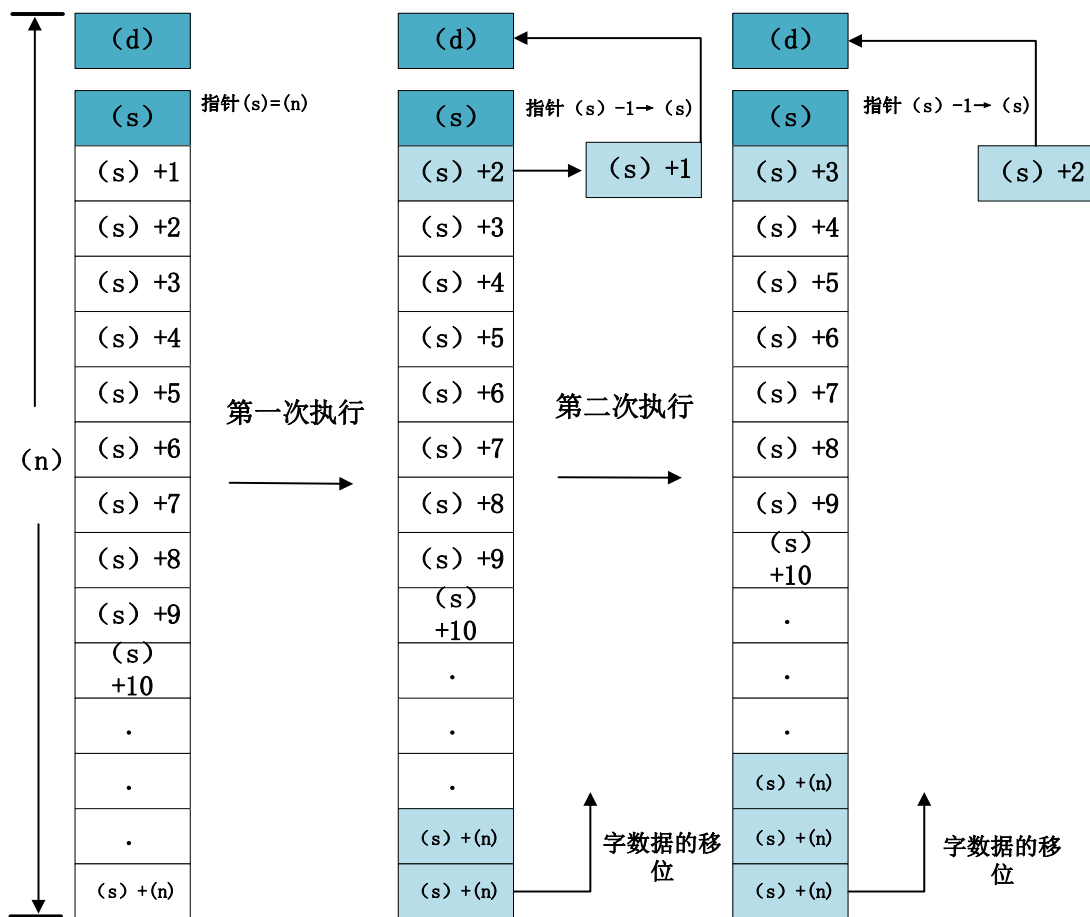
示例

16 位运算(SFRD (P))



当指令执行时, 将通过 SFWR 指令依次写入的(s)+1 传送(读取)到(d)后, 从(s)+1 将(n)-1 点各向上移位 1 字。(s)的存储的数据数-1。

(s)+1 的内容传送(读取)到(d)。与此同时, 指针(s)的内容减少, 数据各向上移位 1 字。
(在连续执行型指令 SFRD 中每个运算周期将移位, 因此应使用脉冲执行型指令 SFRDP 编程。)



错误：

操作数 S 中指定了负值，或者 $N < 2$ 和 $N > 32767$ 时（错误代码：6706）

操作数 S, D, N 中指定的软元件范围超出相应软元件的范围时，或则从 S 开始 N 点的软元件点数超过相应软元件范围时。（错误代码：6004）

注意事项：

执行读取后的数据不会因为读取而使 $(s) + (n)$ 的内容变化。

连续执行型(SFRD)指令的情况下，每个扫描时间(运算周期)将依次读取，但 $(s) + (n)$ 的内容不会变化。

指针(s)为 0 时，将变为无处理，(d)的内容不会变化

4.16.3 POP 从数据表的后入数据读取

概要

用于读取通过先入先出/先入后出控制用的移位写入指令(SFWR)写入的最后数据

指令格式

POP 从后入数据读取指令	
16 位指令	POP
执行类型	连续执行型、脉冲执行型
指令格式	POP(P) S D N

设置数据

操作数	内容	数据	
		16 位范围	16 位类型
S	存储了先入的数据(包含指针数据)的起始软元件编号(存储了数据的起始字软元件编号)	-	有符号 16 位二进制数整数
D	存储后出的数据的软元件编号	-	有符号 16 位二进制数整数
n	存储的数据的点数(包含指针数据, 因此应设置为+1 后的值。)	2-32767	有符号 16 位二进制数整数

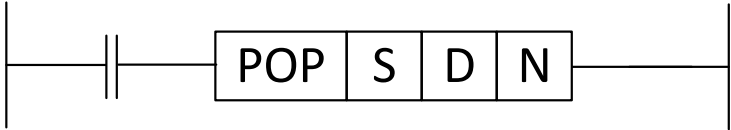
可使用软元件(POP)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○ ^{*1}	○	—	—	—	—
D	—	—	○	○ ^{*1}	○	—	—	—	—
n	—	—	○	○ ^{*1}	○	—	—	○	—

*1 支持变址修饰

示例

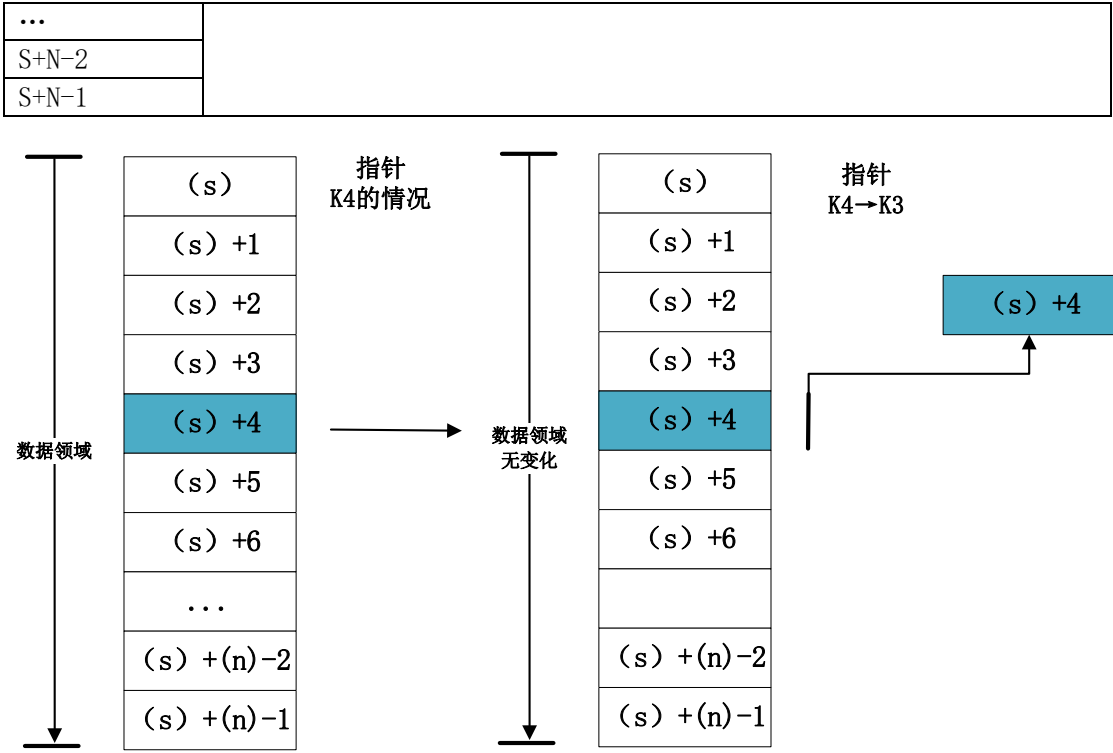
16 位运算(POP (P))



对于“(s)~(s)+(n)-1”的字软元件, 每次执行指令时会将“(s)+指针数据(s)”的软元件读取到(d) (将通过先入先出控制用的移位写入指令(SFWR)写入的最后数据读取到(d))。(n)指定 2~32767。

•指针数据(s)的值-1。

	内容
S	指针数据 (存储的数据个数)
S+1	数据领域: 通过移位写入指令 (SFWR) 先入的数据
S+2	
S+3	



错误：
操作数 s 中指定了负值， $(s) > (n)-1$ 时，或者 $N < 2$ 和 $N > 32767$ 时（错误代码：6706）
操作数 S ， D ， N 或 $(s)+(n)-1$ 中指定的软元件范围超出相应软元件的范围时。（错误代码：6004）

注意事项：
如果以连续执行型进行 POP(P)指令编程，每个运算周期将对指令进行处理。因此有可能无法实现所期望的动作，应加以注意。通常进行 POP(P)指令编程时应使其以“脉冲执行型”或“脉冲化的指定触点”执行。
指针(s)的当前值为 0 的情况下，零标志 SM20 变为 ON，POP(P)指令将变为无处理。
指针(s)的当前值为 1 的情况下，写入 0 到(s)，零标志 SM20 变为 ON。

4.16.4 FINS 数据表的数据插入

概要
将(s)中指定的 16 位数据插入到(d)中指定的数据表的第(n)号中。
执行指令后，从数据表的第(n)号开始的数据将逐个往下顺延。

FINS 从后入数据读取指令	
16 位指令	FINS
执行类型	连续执行型、脉冲执行型
指令格式	FINS(P) S D N

设置数据

操作数	内容	数据	
		16 位范围	16 位类型
S	存储插入数据的软元件起始编号	-	有符号 16 位二进制数整数
D	表的起始编号	-	有符号 16 位二进制数整数
n	插入的表位置	1-32767	有符号 16 位二进制数整数

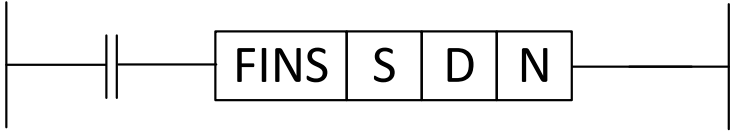
可使用软元件(FINS)

操作数	位		字			双字		常数		
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S	—	—	○	○*1	○	—	—	○	—	—
D	—	—	○	○*1	—	—	—	—	—	—
n	—	—	○	○*1	○	—	—	○	—	—

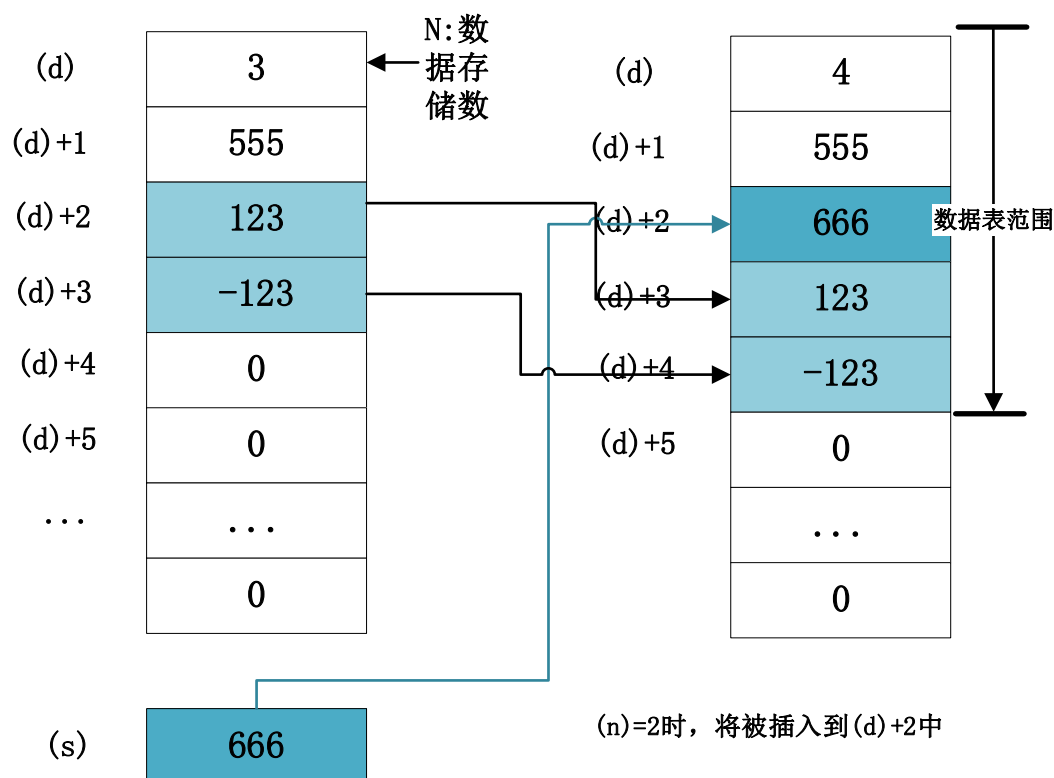
*1 支持变址修饰

示例

16 位运算(FINS (P))



将(s)中指定的 BIN16 位数据插入到(d)中指定的数据表的第(n)号中。执行指令后，从数据表的第(n)号开始的数据将逐个往下顺延。



错误:

操作数(n)的值超出(d)的表的相应软元件的范围, 插入数据的表位置(n)大于数据存储器+1 时, 或者 $N < 2$ 和 $N > 32767$ 时 (错误代码: 6706)

执行 FINS(P)指令时, 操作数 S,D,N 或数据表范围超出相应软元件的范围时。(错误代码: 6004)

注意事项:

数据表中使用的软元件的范围由用户负责管理。

数据表的范围为数据存储器(d)之后的软元件((d)+1)开始(d)个。

4.16.5 FDEL 数据表的数据删除

概要

将(d)中指定的数据表的第(n)号的数据删除后, 存储到(s)中指定的软元件中。

执行指令后, 数据表的第(n)+1 号以后的数据将逐个向前依次顺延。

指令格式

FDEL 从后入数据读取指令	
16 位指令	FDEL
执行类型	连续执行型、脉冲执行型
指令格式	FDEL(P) S D N

设置数据

操作数	内容	数据
-----	----	----

		16 位范围	16 位类型
S	存储删除数据的软元件起始编号	-	有符号 16 位二进制数整数
D	表的起始编号	-	有符号 16 位二进制数整数
n	删除的表位置	1-32767	有符号 16 位二进制数整数

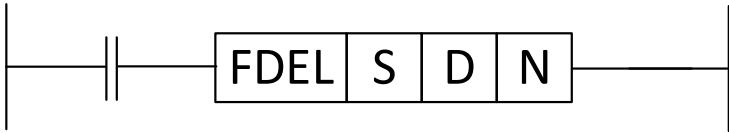
可使用软元件(FDEL)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S	—	—	○	○ ^{*1}	○	—	—	—	—
D	—	—	○	○ ^{*1}	—	—	—	—	—
n	—	—	○	○ ^{*1}	○	—	—	○	—

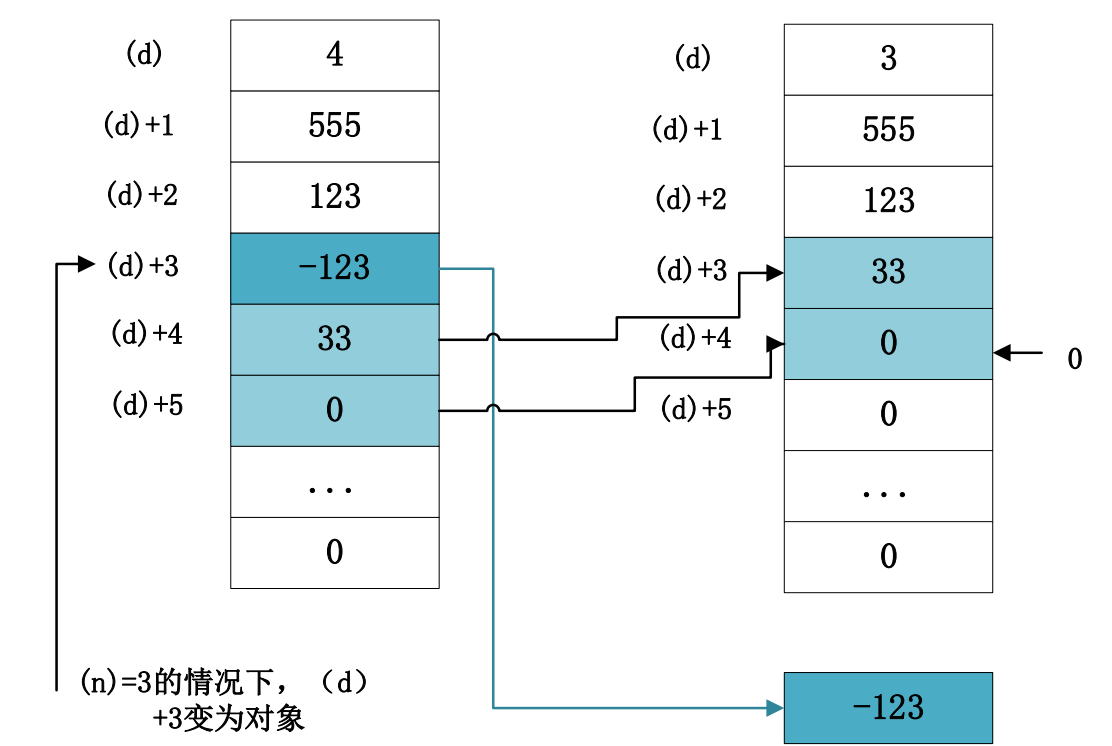
*1 支持变址修饰

示例

16 位运算(FDEL (P))



将(d)中指定的数据表的第(n)号的数据删除后，存储到(s)中指定的软元件中。执行指令后，数据表的第(n)+1 号以后的数据将逐个向前依次顺延将逐个向前依次顺延。



错误:

操作数(n)的值超出(d)的表的相应软元件的范围时, (d)的值为 0 的状况下执行了 FDEL(P)指令时, 删除数据的表位置(n)大于数据存储数时, 或者 $N < 2$ 和 $N > 32767$ 时 (错误代码: 6706)

执行 FDEL(P)指令时, 操作数 S,D,N, 或数据表范围超出相应软元件的范围时。(错误代码: 6004)

注意事项:

数据表中使用的软元件的范围由用户负责管理。

数据表的范围为数据存储数(d)之后的软元件((d)+1)开始(d)个

4.16.6 LIMIT/上下限限位控制

概要

设置输入数值的上限值/下限值然后输出的指令。

指令格式

LIMIT			
16 位指令	LIMIT	32 位指令	DLIMIT
执行类型	连续执行型、脉冲执行型		
指令格式	(D) LIMIT(P) S1 S2 S3 D		

设置数据

操作数	内容	数据			
		16 位范围	16 位类型	32 位范围	32 位类型
S1	下限限位值(最小输出界限值)	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S2	上限限位值(最大输出界限值)	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S3	需要通过上下限限位控制的输入值	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
D	保存已经过上下限限位控制的输出值的软元件起始编号	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数

可使用软元件(LIMIT)

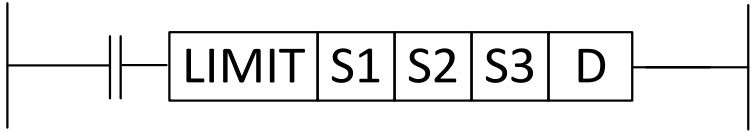
操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	○	—
S2	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	○	—
S3	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	○	—
D	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	—	—

*1 支持变址修饰

*2 仅 32 位指令支持

示例

16 位运算指令（LIMIT(P)）



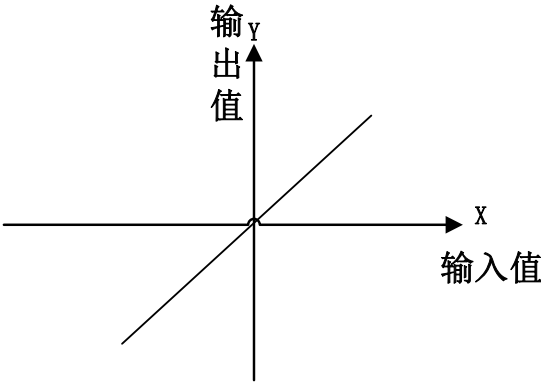
通过判断 S3 中指定的输入值(BIN16 位值)，是否在 S1、S2 指定的上下限值的范围内，以此控制保存在 D 指定的软元件中的输出值。

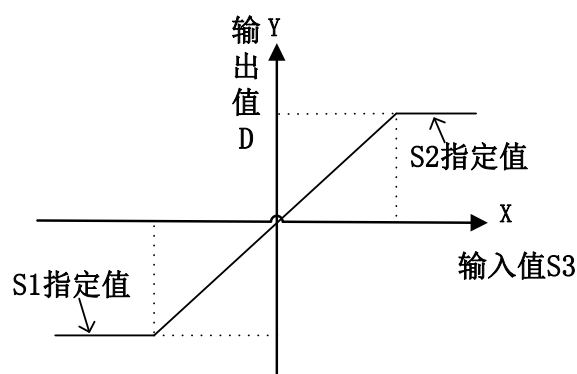
输出值如下所示被控制。

S1 下限值 > S3 输入值 时-----S1 下限值 → D 输出值

S2 上限值 < S3 输入值 时-----S2 上限值 → D 输出值

S1 下限值 ≤ S3 输入值 ≤ S2 上限值 时-----S3 输入值
→ D 输出值



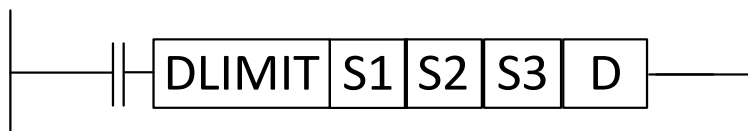


仅通过上限限位值进行控制时，在[S1]指定的下限限位值中设定“-32768”。

仅通过下限限位值进行控制时，在[S2]指定的下限限位值中设定“32768”

32 位运算指令 (DLIMIT(P))

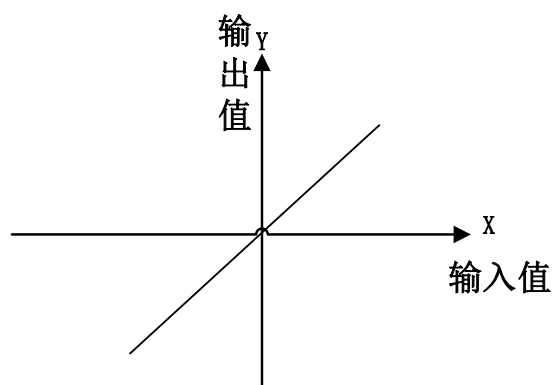
通过判断[S3 +1, S3]中指定的输入值(BIN32 位值)，是否在[S1 +1,S1]、[S2 +1,S2]指定的上下限值的范围内，以此控制保存在[D +1,D]指定的软元件中的输出值。

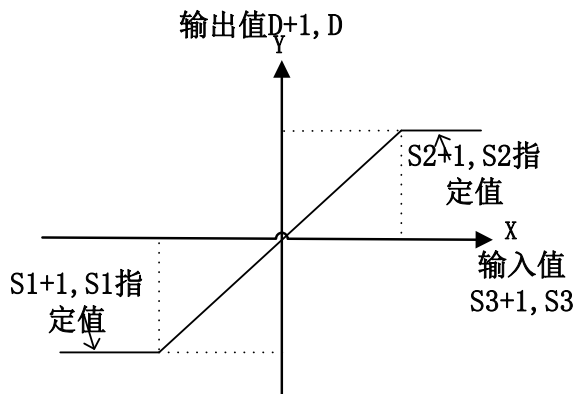


S1+1,S1 下限值>S3+1,S3 输入值 时-----S1+1,S1 下限
值→D+1,D 输出值

S2+1,S2 上限值<S3+1,S3 输入值 时-----S2+1,S2 上限
值→D+1,D 输出值

S1+1,S1 下限值≤S3+1,S3 输入值 ≤S2+1,S2 上限值 时-----
-----S3+1,S3 输入值→D+1,D 输出值





仅通过上限限位值进行控制时，在[S1+1、S1]指定的下限限位值中设定“-2,147,483,648”。
仅通过下限限位值进行控制时，在[S2+1、S2]指定的下限限位值中设定“2,147,483,647”

错误：

当 16 位运算时，S1>S2 时（错误代码：6706）

当进行 32 位运算时，S1+1,S1>S2+1,S2 时（错误代码：6706）

当 16 位运算时，进行运算的值超过 16 位数据范围时，或者 32 位运算时进行运算的值超过 32 位数据范围时。（错误代码：6706）

操作数 S1，S2，S3，D 中指定的软元件范围超出相应软元件的范围时（错误代码：6004）

4.16.7 BAND/死区控制

概要

通过判断输入值是否在指定的死区的上下限范围内，从而来控制输出值的指令

指令格式

BAND			
16 位指令	BAND	32 位指令	DBAND
执行类型	连续执行型、脉冲执行型		
指令格式	(D) BAND(P) S1 S2 S3 D		

设置数据

操作数	内容	数据			
		16 位范围	16 位类型	32 位范围	32 位类型
S1	死区(无输出区域)的下限值)	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S2	死区(无输出区域)的上限值	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S3	要通过死区控制的输出	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数

	入值				
D	保存经过死区控制的输出值的软元件编号	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数

可使用软元件(SCL)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1	○	○*2	○*2	○	—
S2	—	—	○	○*1	○	○*2	○*2	○	—
S3	—	—	○	○*1	○	○*2	○*2	○	—
D	—	—	○	○*1	○	○*2	○*2	—	—

*1 支持变址修饰

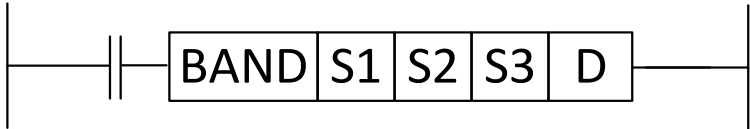
*2 仅 32 位指令支持

示例

16 位运算 (BAND(P))

通过判断 S3 指定的输入值(BIN16 位值)是否在 S1 S2 指定的死区范围内，以此来控制保存在指 D 定的软元件中的输出值。

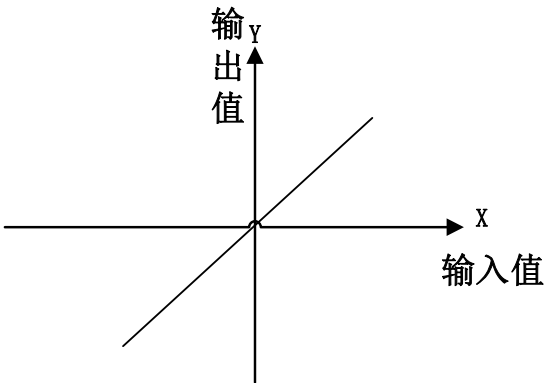
输出值如下所示被控制

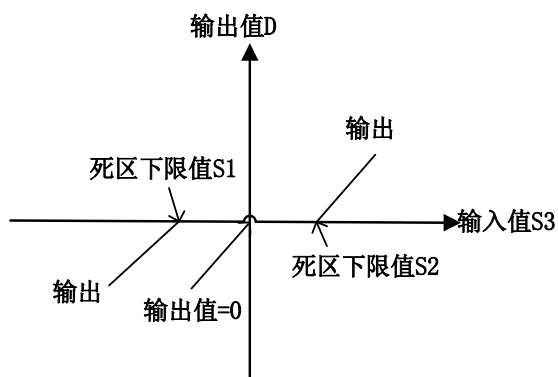


S1 下限值>S3 输入值 时-----S3 输入值 -S1 下限值→
D 输出值

S2 上限值<S3 输入值 时----- S3 输入值 -S2 上限值→
D 输出值

S1 下限值≤S3 输入值 ≤S2 上限值 时-----0→D 输出值

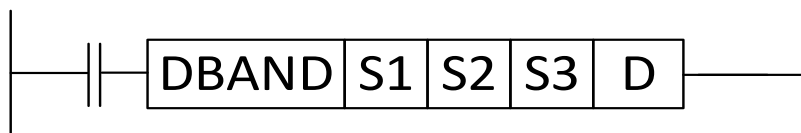




32 位运算 (DBAND(P))

通过判断[S3+1、S3]指定的输入值(BIN32 位值)是否在[S1+1、S1]、[S2+1、S2]指定的死区范围内，以此来控制保存在[D+1、D]指定的软元件中的输出值。

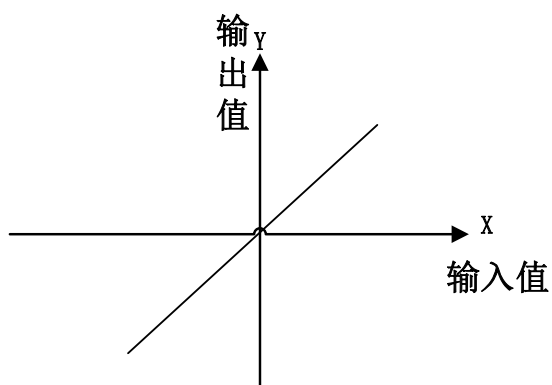
输出值如下所示被控制。

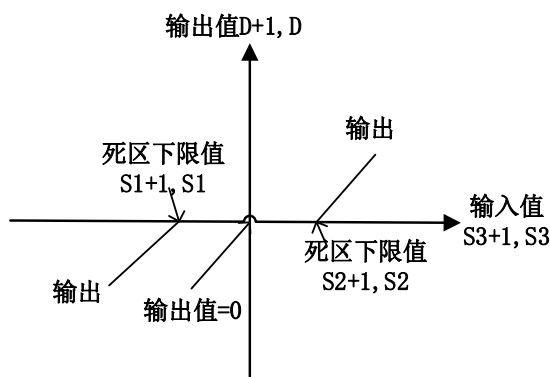


S1+1,S1 下限值>S3+1,S3 输入值 时-----S3+1,S3 输入值 -S1+1,S1 下限值→D+1,D 输出值

S2+1,S2 上限值<S3+1,S3 输入值 时----- S3+1,S3 输入值 -S2 上限值→D +1,D 输出值

S1+1, S1 下限值≤S3+1,S3 输入值 ≤S2+1,S2 上限值 时-----
--0→D+1,D 输出值





注意:

输出值溢出时，如下所示。

– 16 位运算时

输出值为带符号的 16 位 BIN 值。因此，运算结果超出 -32768 ~ 32767 时如下所示。

死区下限值 $S1=10$

输入值 $S3=-32768$

输出值 $= -32768 - 10 = 8000H - AH = 7FF6H = 32758$

– 32 位运算时

输出值为带符号的 32 位 BIN 值。因此，运算结果超出 -2147483648 ~ 2147483647 时如下所示

死区下限值 $(S1+1, S1) = 1000$

输入值 $S3 = -2147483648$

输出值 $= -2147483648 - 1000 = 8000000H - 000003E8H = 7FFFC18H = 2147482648$

错误代码:

当 16 位运算时， $S1 > S2$ 时（错误代码：6706）

当进行 32 位运算时， $S1+1, S1 > S2+1, S2$ 时（错误代码：6706）

当 16 位运算时，进行运算的值超过 16 位数据范围时，或者 32 位运算时进行运算的值超过 32 位数据范围时。（错误代码：6706）

操作数 $S1, S2, S3, D$ 中指定的软元件范围超出相应软元件的范围时（错误代码：6004）

4.16.8 ZONE/区域控制

概要

根据输入值是正数还是负数，用指定的偏差值来控制输出值的指令。

指令格式

ZONE			
16 位指令	ZONE	32 位指令	DZONE
执行类型	连续执行型、脉冲执行型		
指令格式	(D) ZONE(P) S1 S2 S3 D		

设置数据

操作数	内容	数据			
		16 位范围	16 位类型	32 位范围	32 位类型
S1	加在输入值上的负偏差值	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S2	加在输入值上的正偏差值	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S3	要通过区域控制的输入值	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
D	保存已通过区域控制的输出值的软元件起始编号	-36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数

可使用软元件(ZONE)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、KnM、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	○	—
S2	—	—	○	○ ^{*1}	—	○ ^{*2}	○ ^{*2}	○	—
S3	—	—	○	○ ^{*1}	—	○ ^{*2}	○ ^{*2}	○	—
D	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	—	—

*1 支持变址修饰

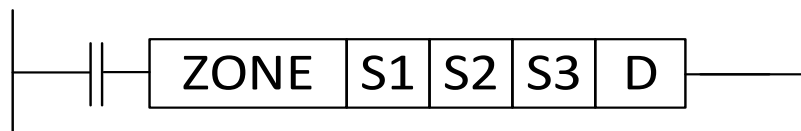
*2 仅 32 位指令支持

示例

16 位运算 (ZONE(P))

在 S3 指定的输入值上加上 S1 或 S2 指定的偏差值，然后保存到 D 指定的软元件编号中。

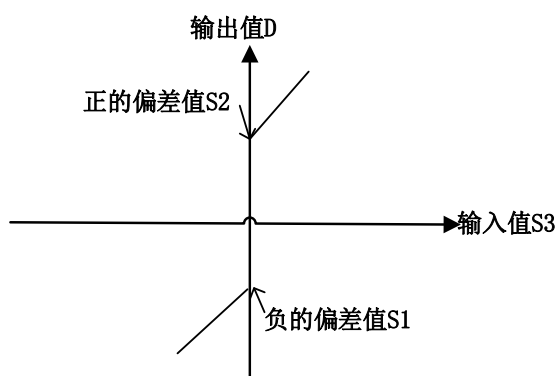
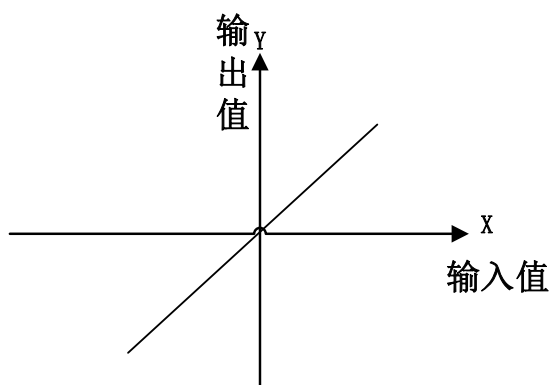
偏差值的附加，如下所示执行



S3 输入值<0 时-----S3 输入值 +S1 负的偏差值→D 输出值

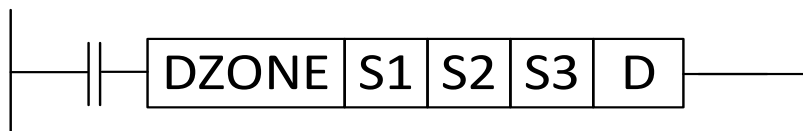
S3 输入值=0 时----- 0→D 输出值

S3 输入值>0 时-----0 S3 输入值 +S2 正的偏差值→D 输出值



32 位运算 (DZONE(P))

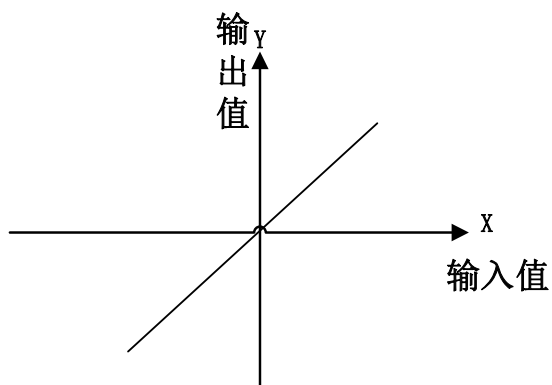
在 S3+1, S3 指定的输入值上加上 S1+1、S1 或 S2 +1、S2 指定的偏差值，然后保存到 D ,D +1 指定的软元件编号中。

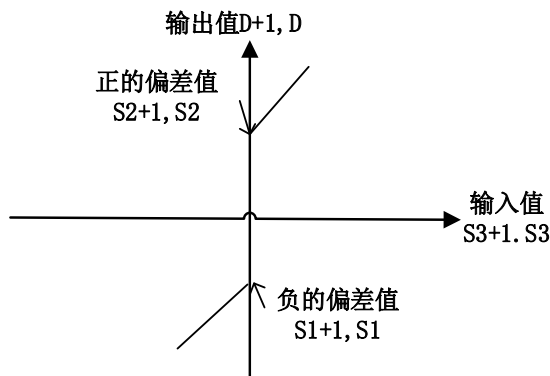


S3+1,S3 输入值<0 时-----S3+1,S3 输入值 +S1+1,S1 负的偏差值→D+1,D 输出值

S3+1,S3 输入值=0 时----- 0→D +1,D 输出值

S3+1,S3 输入值>0 时----- S3+1,S3 输入值 +S2+1,S1 正的偏差值→D +1,D 输出值





注意：

输出值溢出时，如下所示。

– 16 位运算时

输出值为带符号的 16 位 BIN 值。因此，运算结果超出 -32768 ~ 32767 时如下所示。

负的偏差值 S1=-100

输入值 S3=-32768

→ 输出值=-32768 + (-100) =8000H+FF9CH=7F9CH=32668

– 32 位运算时

输出值为带符号的 32 位 BIN 值。因此，运算结果超出 -2147483648 ~ 2147483647 时如下所示

负的偏差值(S1+1,S1)=-1000

输入值 S3=-2147483648

输出值=-2147483648+(-1000)=8000000H-000003E8H=7FFFC18H=2147482648

错误代码：

当 16 位运算时，进行运算的值超过 16 位数据范围时，或者 32 位运算时进行运算的值超过 32 位数据范围时。(错误代码：6706)

操作数 S1，S2，S3，D 中指定的软元件范围超出相应软元件的范围时（错误代码：6004）

416.9 SCL/定坐标(不同点坐标数据)

概要

根据指定的数据表格，对输入值执行定坐标后输出的指令。

此外，还有数据表格结构不同的 SCL2 指令

指令格式

SCL			
16 位指令	SCL	32 位指令	DSCL
执行类型	连续执行型、脉冲执行型		

指令格式	(D) SCL(P) S1 S2 D
------	--------------------

设置数据

操作数	内容	数据			
		16 位范围	16 位类型	32 位范围	32 位类型
S1	执行定坐标的输入值或是保存输入值的软元件编号	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S2	定坐标用的转换表格软元件的起始编号	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
D	保存被定坐标控制的输出值的软元件编号	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数

可使用软元件(SCL)

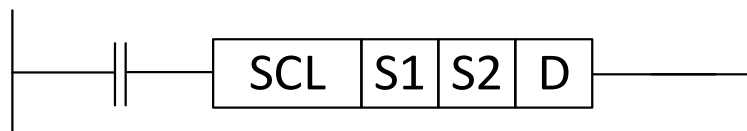
操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	○	—
S2	—	—	—	○ ^{*1}	—	—	—	—	—
D	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	—	—

*1 支持变址修饰

*2 仅 32 位指令支持

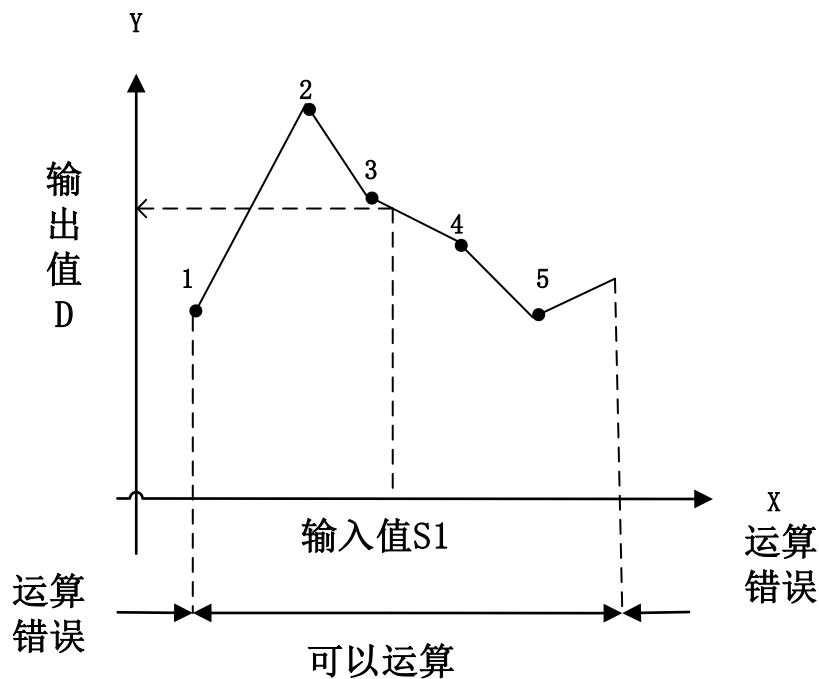
示例

16 位运算(SCL (P))



根据指定的转换特性，对 S1 指定的输入值执行定坐标，然后保存到 D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2S 指定的软元件开始的数据表格执行的。

但是，输出数据不是整数时，小数第 1 位四舍五入后输出



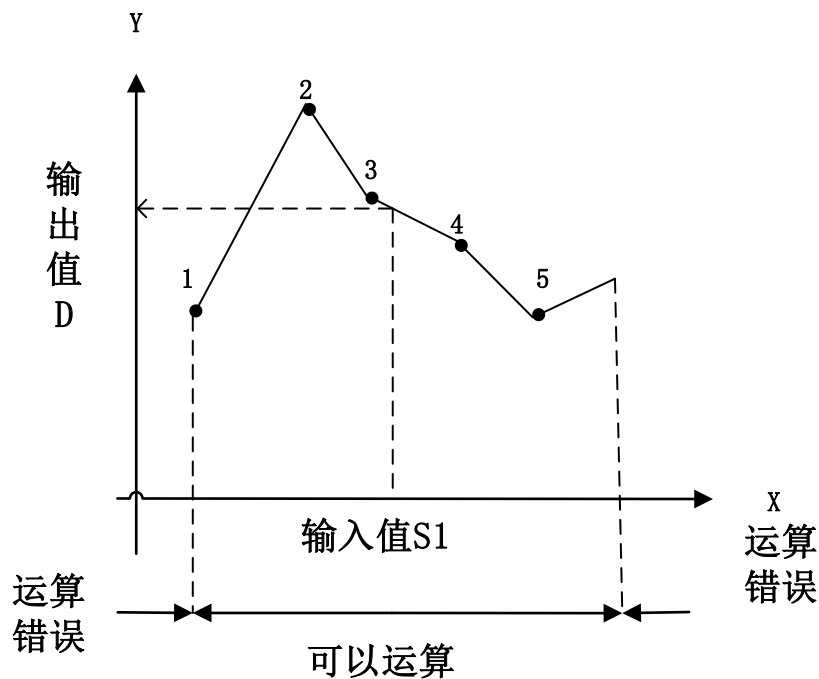
定坐标用转接设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数 (当为上图时为 5)		S2
点 1	X 坐标	S2+1
	Y 坐标	S2+2
点 2	X 坐标	S2+3
	Y 坐标	S2+4
点 3	X 坐标	S2+5
	Y 坐标	S2+6
点 4	X 坐标	S2+7
	Y 坐标	S2+8
点 5	X 坐标	S2+9
	Y 坐标	S2+10

32 位运算 (DSCL(P))

[根据指定的转换特性, 对 $S1 + 1$ 、 $S1$]指定的输入值执行定坐标, 然后保存到[$D+1$ 、 D]指定的

软元件编号中。定坐标用的转换, 是依据保存在[$S2+1$ 、 $S2$]指定的软元件开始的数据表格执行的。但是, 输出数据不是整数值时, 小数第 1 位四舍五入后输出



定坐标用转接设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数（当为上图时为 5）		S2+1, S2
点 1	X 坐标	S2+3, S2+2
	Y 坐标	S2+5, S2+4
点 2	X 坐标	S2+7, S2+6
	Y 坐标	S2+9, S2+8
点 3	X 坐标	S2+11, S2+10
	Y 坐标	S2+13, S2+12
点 4	X 坐标	S2+15, S2+14
	Y 坐标	S2+17, S2+16
点 5	X 坐标	S2+19, S2+18
	Y 坐标	S2+21, S2+20

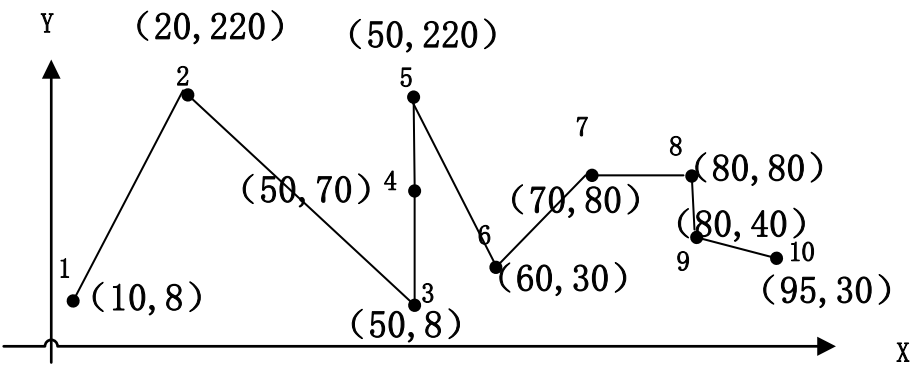
定坐标用转换表格的设定

设定项目		设定数据表格的软元件分配 16 位	设定数据表格的软元件分配 32 位
坐标点数（当为上图时为 5）		S2	S2+1, S2
点 1	X 坐标	S2+1	S2+3, S2+2
	Y 坐标	S2+2	S2+5, S2+4
点 2	X 坐标	S2+3	S2+7, S2+6
	Y 坐标	S2+4	S2+9, S2+8
...	...	...	...
	...	...	...
点 n（最终）	X 坐标	S2+ 2n-1	S2+ 4n-1, S2+ 4n-2
	Y 坐标	S2+ 2n	S2+ 4n-1, S2+ 4n

定坐标转换表格的设计例子

定坐标用转换表格的设定例子中例举了 16 位运算时的例子。执行 32 位运算时，请用 BIN32 位数据设定各设定项目中的数据。

当为下图所示的定坐标用转换特性时，请设定成如下所示的数据表格。

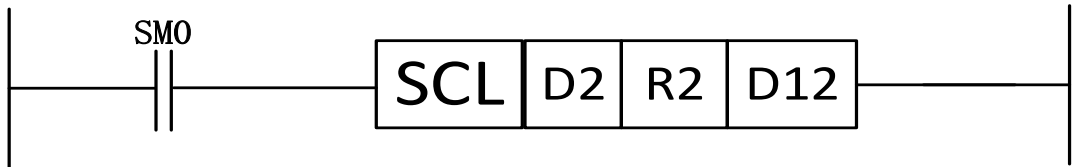


定坐标用转换设定数据表格的设定

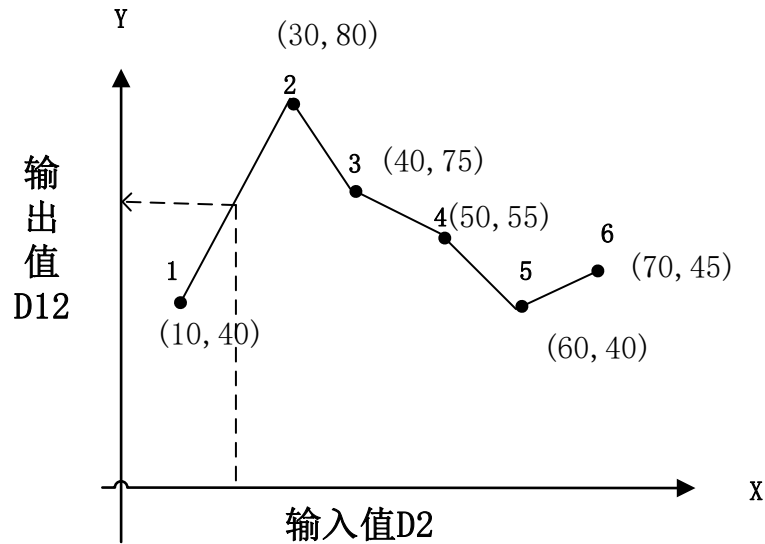
设定项目		设定软元件及设定内容			备注
		S2 指定了 R0 时		设定内容	
坐标点数		S2	R0	K10	
点 1	X 坐标	S2+1	R1	K10	
	Y 坐标	S2+2	R2	K8	
点 2	X 坐标	S2+3	R3	K20	
	Y 坐标	S2+4	R4	K222	
点 3	X 坐标	S2+5	R5	K50	指定 3 点的坐标， 则输出值为中间值。 这个例子中，将点 4 的 y 坐标指定为输出值（中间值）。 此外，3 点以上的 X 坐标相同时，也输出第 2 点的数值
	Y 坐标	S2+6	R6	K8	
点 4	X 坐标	S2+7	R7	K50	
	Y 坐标	S2+8	R8	K70	
点 5	X 坐标	S2+9	R9	K50	
	Y 坐标	S2+10	R10	K220	
点 6	X 坐标	S2+11	R11	K60	
	Y 坐标	S2+12	R12	K30	
点 7	X 坐标	S2+13	R13	K70	
	Y 坐标	S2+14	R14	K80	
点 8	X 坐标	S2+15	R15	K80	如果象这

	Y 坐标	S2+16	R16	K80	样指定 2 点的坐标，则输出值取后一个点的 y 坐标值。这个例子中，将点 9 的 y 坐标指定为输出值。
点 9	X 坐标	S2+17	R17	K80	
	Y 坐标	S2+18	R18	K40	
点 10	X 坐标	S2+19	R19	K95	
	Y 坐标	S2+20	R20	K30	

示例
 根据 R0 开始的软元件设定的定坐标转换表格，对 D2 输入的值执行定坐标，然后输出到 D12 中的程序



示意图



设定项目		软元件	数值
坐标点数（当为上图时为 6）		R2	6
点 1	X 坐标	R3	10
	Y 坐标	R4	40
点 2	X 坐标	R5	30
	Y 坐标	R6	80

点 3	X 坐标	R7	40
	Y 坐标	R8	75
点 4	X 坐标	R9	50
	Y 坐标	R10	55
点 5	X 坐标	R11	50
	Y 坐标	R12	40
点 6	X 坐标	R13	70
	Y 坐标	R14	45

错误：

数据表格的 Xn 数据没有按照升序排列时（错误代码：6706）

但是，由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。

S1 在数据表格设定的范围以外时，（错误代码：6706）

运算过程中的数值超出了 32 位数据的范围时。（错误代码：K6706）

此时，请确认各点之间的距离没有超出 65535 以上。

如果超出 65535 时，请缩短各点之间的距离。

操作数 S1，S2，D 中指定的软元件范围超出相应软元件的范围时（错误代码：6004）

4.16.10 SCL2/定坐标 2(X/Y 坐标数据)

概要

根据指定的数据表格，对输入值执行定坐标后输出的指令。

指令格式

SCL 2			
16 位指令	SCL2	32 位指令	DSCL2
执行类型	连续执行型、脉冲执行型		
指令格式	(D) SCL2(P) S1 S2 D		

设置数据

操作数	内容	数据			
		16 位范围	16 位类型	32 位范围	32 位类型
S1	执行定坐标的输入值或是保存输入值的软元件编号	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数
S2	定坐标用的转换表格软元件	- 36768 ~ 32767	有符号 16 位二进制数整数	-2147483648 ~ 2147483647	有符号 32 位二进制数整数

	的起始 编号				
D	保存被 定坐标 控制的 输出值 的软元 件编号	- 36768 ~ 32767	有符号 16 位二进 制数整数	-2147483648 ~ 2147483647	有符号 32 位二进 制数整数

可使用软元件(SCL2)

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	○	—
S2	—	—	—	○ ^{*1}	—	—	—	—	—
D	—	—	○	○ ^{*1}	○	○ ^{*2}	○ ^{*2}	—	—

*1 支持变址修饰

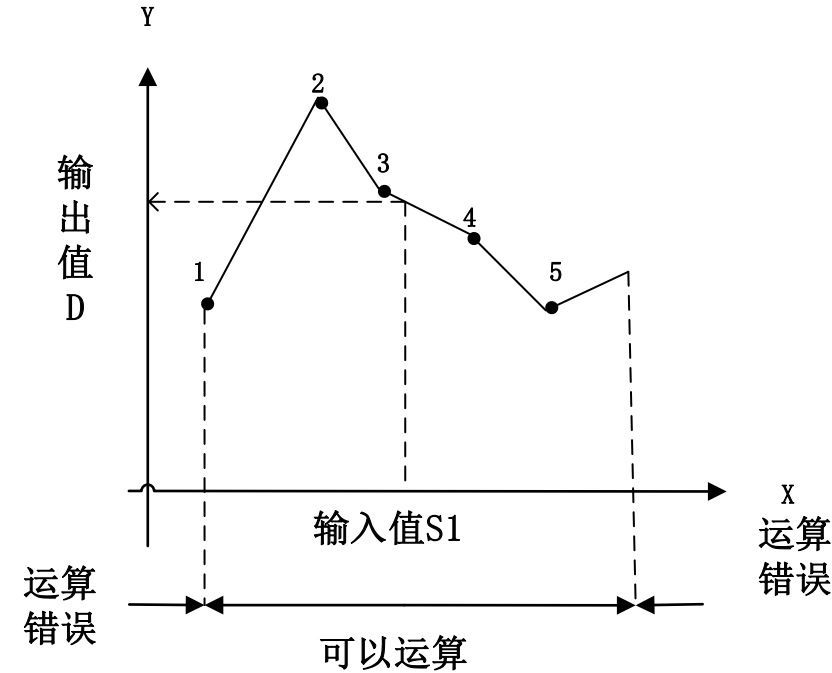
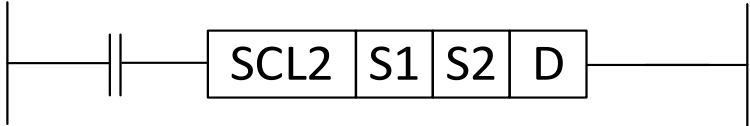
*2 仅 32 位指令支持

示例

16 位运算(SCL2 (P))

根据指定的转换特性，对 S1 指定的输入值执行定坐标，然后保存到 D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2 指定的软元件开始的数据表格执行的。

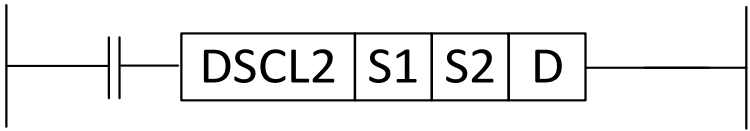
但是，输出数据不是整数值时，小数第 1 位四舍五入后输出



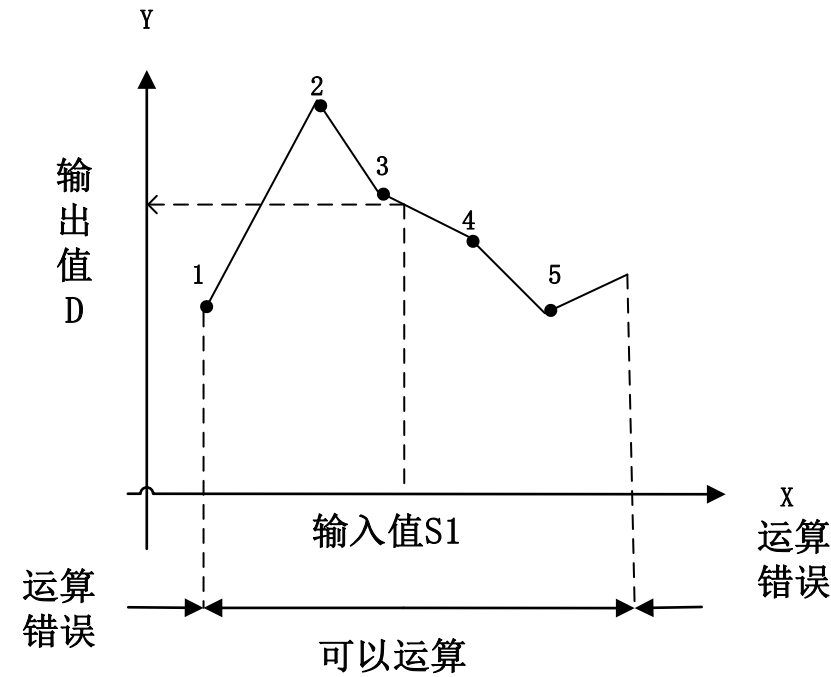
定坐标用转接设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数（当为上图时为 5）		S2
X 坐标	点 1	S2+1
	点 2	S2+2
	点 3	S2+3
	点 4	S2+4
	点 5	S2+5
Y 坐标	点 1	S2+6
	点 2	S2+7
	点 3	S2+8
	点 4	S2+9
	点 5	S2+10

32 位运算（DSCL2(P)）



根据指定的转换特性，对 S1 +1、S1 指定的输入值执行定坐标，然后保存到 D+1、D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2 +1、S2 指定的软元件开始的数据表格执行的。但是，输出数据不是整数时，小数第 1 位四舍五入后输出



定坐标用转接设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数（当为上图时为 5）		S2+1, S2
X 坐标	点 1	S2+3, S2+2
	点 2	S2+5, S2+4

Y 坐标	点 3	S2+7, S2+6
	点 4	S2+9, S2+8
	点 5	S2+11, S2+10
	点 1	S2+13, S2+12
	点 2	S2+15, S2+14
	点 3	S2+17, S2+16
	点 4	S2+19, S2+18
	点 5	S2+21, S2+20

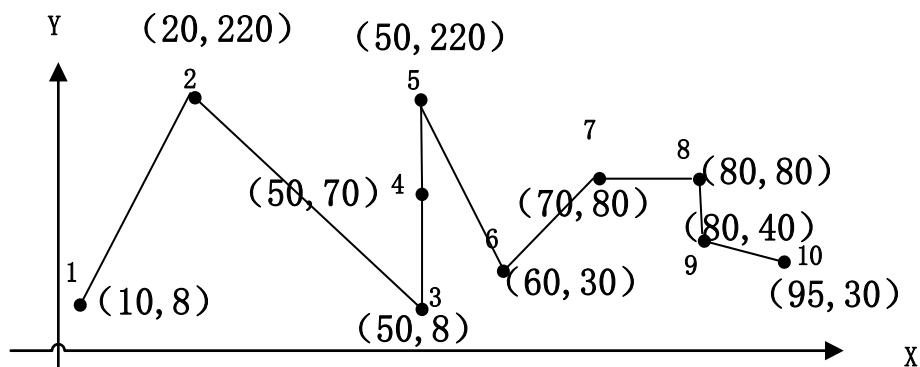
定坐标用转换表格的设定

设定项目		设定数据表格的软元件 分配 16 位	设定数据表格的软元 件分配 32 位
坐标点数（当为上图时为 5）		S2	S2+1, S2
X 坐标	点 1	S2+1	S2+3, S2+2
	点 2	S2+2	S2+5, S2+4
	...	...	...
	点 n（最终）	S2+n	S2+2n+1, S2+2n
Y 坐标	点 1	S2+n+1	S2+2n+3, S2+2n+2
	点 2	S2+n+2	S2+2n+5, S2+2n+4
	...	...	...
	点 n（最终）	S2+ 2n	S2+ 4n-1, S2+ 4n

定坐标转换表格的设计例子

定坐标用转换表格的设定例子中列举了 16 位运算时的例子。执行 32 位运算时，请用 BIN32 位数据设定各设定项目中的数据。

当为下图所示的定坐标用转换特性时，请设定成如下所示的数据表格。



定坐标用转换设定数据表格的设定

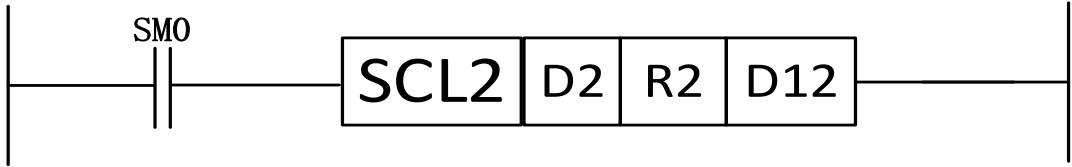
设定项目		设定软元件及设定内容			备注
		S2 指定了 R0 时		设定内容	
坐标点数		S2	R0	K10	
X 坐标	点 1	S2+1	R1	K10	*1
	点 2	S2+2	R2	K20	
	点 3	S2+3	R3	K50	
	点 4	S2+4	R4	K50	
	点 5	S2+5	R5	K50	

	点 6	S2+6	R6	K60	
	点 7	S2+7	R7	K70	
	点 8	S2+8	R8	K80	*2
	点 9	S2+9	R9	K80	
	点 10	S2+10	R10	K95	
Y 坐标	点 1	S2+11	R11	K8	
	点 2	S2+12	R12	K220	
	点 3	S2+13	R13	K8	
	点 4	S2+14	R14	K70	
	点 5	S2+15	R15	K220	
	点 6	S2+16	R16	K30	
	点 7	S2+17	R17	K80	
	点 8	S2+18	R18	K80	
	点 9	S2+19	R19	K40	
	点 10	S2+20	R20	K30	

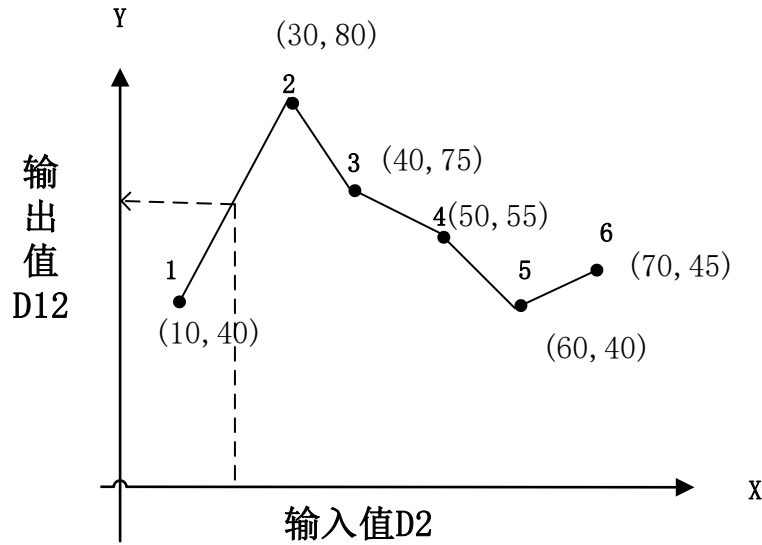
- *1: 指定 3 点的坐标，则输出值为中间值。这个例子中，将点 4 的 y 坐标指定为输出值(中间值)。此外，3 点以上的 X 坐标相同时，也输出第 2 点的数值
- *2: 如果象这样指定 2 点的坐标，则输出值取后一个点的 y 坐标值。这个例子中，将点 9 的 y 坐标指定为输出值。

示例

根据 R0 开始的软元件设定的定坐标转换表格，对 D2 输入的值执行定坐标，然后输出到 D12 中的程序



示意图



设定项目		软元件	数值
坐标点数（当为上图时为 6）		R2	6
X 坐标	1	R3	10
	2	R4	30
	3	R5	40
	4	R6	50
	5	R7	60
	6	R8	70
Y 坐标	1	R9	40
	2	R10	80
	3	R11	75
	4	R12	55
	5	R13	40
	6	R14	45

错误：

数据表格的 Xn 数据没有按照升序排列时（错误代码：6706）

但是，由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。

S1 在数据表格设定的范围以外时，（错误代码：6706）

运算过程中的数值超出了 32 位数据的范围时。（错误代码：K6706）

此时，请确认各点之间的距离没有超出 65535 以上。

如果超出 65535 时，请缩短各点之间的距离。

操作数 S1，S2，D 中指定的软元件范围超出相应软元件的范围时（错误代码：6004）

4.17 运动控制

4.17.1 DDSZR：原点回归模式

概要

原点回归模式

指令格式

16 位指令	-	32 位指令	DDSZR
执行类型	连续执行型		
指令格式	-	DDSZR S1 S2 S3 D1	

操作数明细

DDSZR S1 S2 S3 D1		原点回归模式
S1	原点回归高速	开始执行原点回归时的第一段速度
S2	原点回归低速	检测到 DOG 信号后的第二段速度
S3	零点信号数	检测到 DOG 信号后再检测到指令数量的零点信号后回到原点。动作完成。
D1	输出脉冲的位软	Y0 ~ Y7*1

	元件(Y)编号	
--	---------	--

*1. 针对不同机型，轴最大值不一样，
R1,R2,R 系列:Y0 ~ Y3;
R8:Y0 ~ Y5;
R8C,R8P:Y0 ~ Y7;

可使用操作数

操作数	内容	范围	数据类型
S1	原点回归高速	1 ~ 2147483647	s16
S2	原点回归低速	1 ~ 2147483647	s32
S3	零点信号数	0 ~ 32767	s32
D1	输出脉冲的位软元件(Y)编号	Y0 ~ Y7*1	Bit

注：s16 有符号 16 位数，s32 有符号 32 位数。

请勿将原点回归速度和零点信号数设置为<0,设置为小于 0 后报 SM329,参数异常，轴不动作。

原点回归高速<=原点回归低速时，可以运动，不报错，以原点回归低速运动。

S3 为 0 时代表不找 Z 相，忽略 Z 相输入端子的硬件信号。

可使用软元件

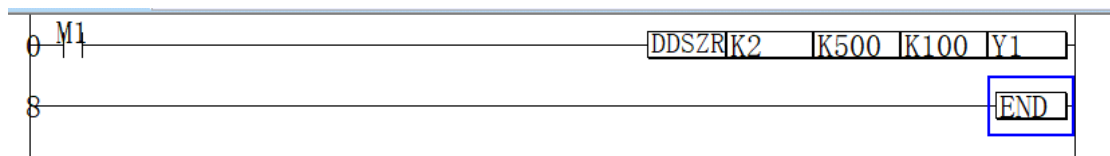
操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	—	○*1	○* 1	○* 1	○* 1	○	—	
S2	—	—	—	○*1	○* 1	○* 1	○* 1	○	—	
S3	—	—	—	—	—	—	-	○	—	
D1	仅 Y	—	—	—	—	—	—	-	—	

*1 支持变址修饰

○:支持、—:不支持

示例

下图中，当 M1 闭合后，轴一进行一种回原运动。比如 M1 闭合后，轴一以 500pps 的速度执行回原高速运动，当检测到 DOG 信号后以 100pps 的速度执行回原低速运动，最后当 PLC 检测到两次零点信号后运动停止，回原完成。具体的回原模式介绍请参考定位篇。



报错相关：(参考定位篇。)

注意: (参考定位篇。)

4.17.2 DDVIT: 中断定位模式

概要

在定位中, 用 DDVIT 指令执行中断定长进给。
该指令可通过用户程序控制中断信号指令格式

16 位指令	-	32 位指令	DDVIT
执行类型	连续执行型		
指令格式	-	DDVIT S1 S2 S3	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	○ 1	* 1	○* 1	○	—
S2	—	—	—	○*1	○ 1	* 1	○* 1	○	—
D1	仅 Y	—	—	—	—	—	—	—	—

*1 支持变址修饰

DDVIT S1 S2 D1		中断定位模式
S1	目标地址 (相对地址)	检测到中断信号后输出的脉冲数。大于 0: 正转; 小于 0: 反转。
S2	运行速度	脉冲输出频率
D1	输出脉冲的位软元件(Y)编号	Y0 ~ Y7*1

*1. 针对不同机型, 轴最大值不一样,
R1,R2,R 系列:Y0 ~ Y3;
R8:Y0 ~ Y5;
R8C,R8P:Y0 ~ Y7;

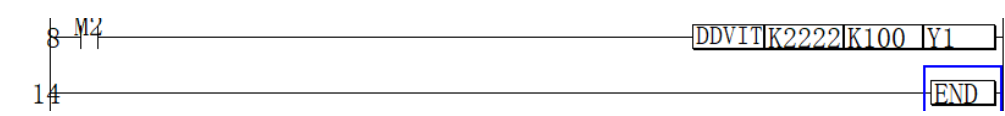
R 设置数据

操作数	内容	范围	类型
(s1)	目标地址 (相对地址)	-2147483648~+2147483647	s32
(S2)	运行速度	1 ~ 2147483647	s32
(D1)	输出脉冲的位软元件(Y)编号	Y0 ~ Y7	Bit

将脉冲数设定为 2147483648 以上时, 将变为出错状态且不动作。

示例

下图中，当 M2 闭合后，轴一进行中断定位运动，指令速度加速至 100pps，保持，直至收到中断信号，运动到轴地址为 2222 的位置，中断定位完成。具体的中断运动介绍参考定位篇。



报错相关：(参考定位篇。)

注意：(参考定位篇。)

4.17.3 DDVRA：绝对定位模式

概要

该指令通过绝对方式(采用绝对地址的位置指定)，进行 1 速定位。
以原点为基准指定位置(绝对地址)进行定位动作。起点在哪里都没有关系。

指令格式

16 位指令	-	32 位指令	DDVRA
执行类型	连续执行型		
指令格式	-	DDVRA S1 S2 S3	

操作数支持

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	—	○*1	○* 1	○* 1	○* 1	○	—	
S2	—	—	—	○*1	○* 1	○* 1	○* 1	○	—	
S3	仅 Y	—	—	—	—	—	—	—	—	

*1 支持变址修饰

DDRVA S1 S2 D1		相对定位模式
S1	绝对目标地址	输出的脉冲个数。大于 0：正转，小于 0：反转。
S2	运行速度	脉冲输出频率
D1	输出脉冲的位软元 件(Y)编号	Y0 ~ Y7*1

*1. 针对不同机型，轴最大值不一样，
R1,R2,R 系列:Y0 ~ Y3;

R8:Y0 ~ Y5;
R8C,R8P:Y0 ~ Y7;

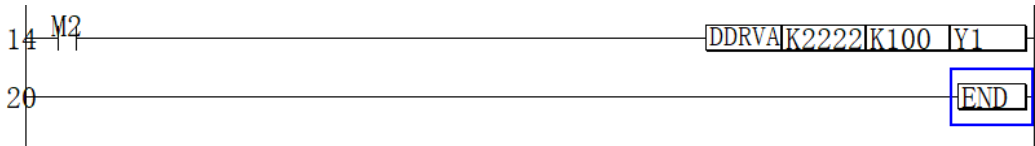
设置数据

操作数	内容	范围	类型
(S1)	绝对目标地址	-2147483648~+2147483647	s32
(S2)	运行速度	1 ~ 2147483647	s32
(D1)	输出脉冲的位软元件(Y)编号	Y0 ~ Y7	Bit

将脉冲数设定为 2147483648 以上时，将变为出错状态且不动作。

示例

下图中，当 M2 闭合后，轴一进行绝对运动，指令速度加速至 100pps,保持，直至轴 1 的当前地址变为 2222，绝对定位完成。具体的绝对运动介绍参考定位篇。



报错相关：(参考定位篇。)

注意：(参考定位篇。)

4.17.4 DDVRI：相对定位模式

概要

该指令通过增量方式(采用相对地址的位置指定)，进行 1 速定位。以当前停止的位置作为起点，指定移动方向和移动量(相对地址)进行定位动作。

指令格式

16 位指令		32 位指令	DDVRI
执行类型	连续执行型		
指令格式		DDVRI S1 S2 D1	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S	KnX、KnY、KnM、KnL、KnS	T、ST、C、D、W、SD、S	Z	LC	LZ	K、H	E
		D.X、SW.X、R.X	M、KnF、KnB、KnSB、KnS	W、R					
S1	—	—	—	○*1	○*1 1	○*1 1	○*1 1	○	—

S2	—	—	—	○*1	○*1	○*1	○	—
D1	仅 Y	—	—	—	—	—	—	—

*1 支持变址修饰

可使用操作数

操作数	内容	范围	类型
(s1)	相对目标地址	-2147483648~+2147483647	s32
(S2)	运行速度	1 ~ 2147483647	s32
(D1)	输出脉冲的位软元件(Y)编号	Y0 ~ Y7	Bit

将脉冲数设定为 2147483648 以上时，将变为出错状态且不动作。

可使用软元件

操作数	位	字		双字		常数	
	X、Y、M、L、SM、F、B、SB、S	T、ST、C、D、W、SD、SW、R	Z	LC	LZ	K、H	E
S1	—	○	○	○	○	○	—
S2	—	○	○	○	○	○	—
D1	仅 Y	—	—	—	—	—	—

示例

下图中，当 M2 闭合后，轴一进行相对运动，指令速度加速至 100pps,保持，直至轴 1 的当前地址增加 2222，相对定位完成。具体的相对运动介绍参考定位篇。



报错相关：（参考定位篇。）

注意：（参考定位篇。）

4.17.5 DPLSV:可变速运行

概要

在定位中，用可变速脉冲输出指令执行可变速度运行。该指令在改变速度时，可以带加减速动作。

16 位指令		32 位指令	DPLSV
执行类型	连续执行型		
指令格式		DPLSV S1 S2 S3 S4	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	—	○* 1	○* 1	○* 1	○	—
S2	—	—	—	○*1	○* 1	—	—	○	—
S3	—	—	—	○*1	○* 1	—	—	○	—
D1	仅 Y	—	—	—	—	—	—	—	—

*1 支持变址修饰

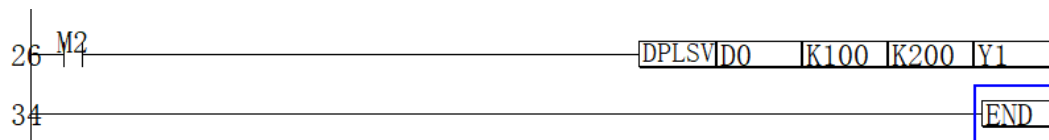
操作数	内容	范围	数据类型
S1	指令速度或存储了数据的字软元件编号	-2147483648 ~ +2147483647*2	s32
S2	加速时间	0~32767*1	s16
S3	减速时间	0~32767*1	s16
D1	输出脉冲的位软元件(Y)编号	Y0 ~ Y7	D1

*1: 加减速时间为 0 时，代表无加减速时间。此指令的加减速时间与 SD5520，SD5521 等无关，为独立参数。

*2: 受限于硬件，最高输出频率 200KHZ。

示例

下图中，当 M4 闭合后，轴一进行可变速运动。初始时 D0 为 0 不运动，当 D0 中的值设定 1000 时，速度从 0pps 开始以 100 的加速度，加速至 1000pps，若 D0 的值改为 800，则速度以 200 的减速度减至 800pps。具体的可变速运动介绍参考定位篇。



报错相关：(参考定位篇。)

注意：(参考定位篇。)

4.17.6 DMOVEHOME：原点回归模式

概要

定位指令，产生正转脉冲或者反转脉冲后，增减当前地址的内容。

CPU 模块的电源置为 OFF 后，当前地址清零，因此上电后，请务必使机械位置和 CPU 模块的当前地址的位置相吻合。用机械原点回归用的 DMOVEHOME 指令进行原点回归，使机械位置和 CPU 模块中的当前地址相吻合。

指令格式

16 位指令	-	32 位指令	DMOVEHOME
执行类型	连续执行型		
指令格式	-	DMOVEHOME S1 S2 S3 S4	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	—	—	—	—	○	—
S2	—	—	—	○*1	○* 1	○* 1	○* 1	○	—
S3	—	—	—	○*1	○* 1	○* 1	○* 1	○	
S4	—	—	—	—				○	

*1 支持变址修饰

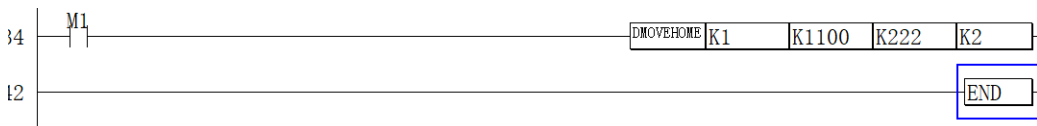
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	轴编号	—	K1-K12	—	实数
S2	原点回归 高速	—	1~2147483647	—	实数
S3	原点回归 低速	—	1~2147483647	—	实数
S4	零点信号 数	—	0~32767	—	实数

注：不同机型针对不同机型，轴最大值不一样，如 HCR 系列最大四轴，S1 最大有效值为 k4; R8P,R8C 最大为 8 轴，R9 最大为 12 轴，下同。

示例

下图中，当 M1 闭合后，轴一进行一种回原运动。比如 M1 闭合后，轴一以 1100pps 的速度执行回原高速运动，当检测到 DOG 信号后以 222pps 的速度执行回原低速运动，最后当 PLC 检测到两次零点信号后运动停止，回原完成。具体的回原模式介绍请参考定位篇。



报错相关：（参考定位篇。）

注意：（参考定位篇。）

4.17.7 DMOVEABS: 绝对定位模式

概要

该指令通过绝对方式(采用绝对地址的位置指定)，进行 1 速定位。

以原点为基准指定位置(绝对地址)进行定位动作。起点在哪里都没有关系。

指令格式

16 位指令	-	32 位指令	DMOVEABS
执行类型	连续执行型		
指令格式	-	DMOVEABS S1 S2 S3	

操作数支持

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	—	—	—	—	—	○	—	
S2	—	—	—	○*1	○* 1	○* 1	○* 1	○	—	
S3	—	—	—	○*1	○* 1	○* 1	○* 1	○		

*1 支持变址修饰

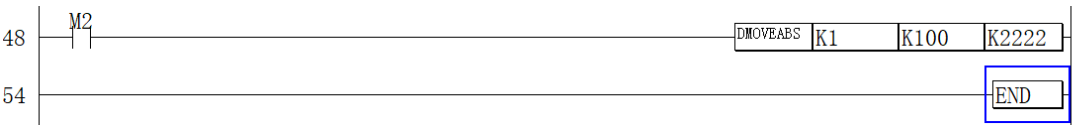
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	轴编号	—	K1-K12	—	实数
S2	运行速度	—	1~2147483647	—	实数
S3	绝对目标地址	—	-214748364 ~ 2147483647	—	实数

注：不同机型针对不同机型，轴最大值不一样，如 HCR 系列最大四轴，S1 最大有效值为 k4; R8P,R8C 最大为 8 轴，R9 最大为 12 轴，下同。

示例

下图中，当 M2 闭合后，轴一进行绝对运动，指令速度加速至 100pps,保持，直至轴 1 的当前地址变为 2222，绝对定位完成。具体的绝对运动介绍参考定位篇。



报错相关：(参考定位篇。)

注意：(参考定位篇。)

4.17.8 DMOVEINC：相对定位模式

概要

该指令通过增量方式(采用相对地址的位置指定)，进行1速定位。以当前停止的位置作为起点，指定移动方向和移动量(相对地址)进行定位动作。

指令格式

16 位指令	-	32 位指令	DMOVEINC
执行类型	连续执行型		
指令格式	-	DMOVEINC S1 S2 S3	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	—	—	—	—	○	—
S2	—	—	—	○*1	○* 1	○* 1	○* 1	○	—
S3	—	—	—	○*1	○* 1	○* 1	○* 1	○	—

*1 支持变址修饰

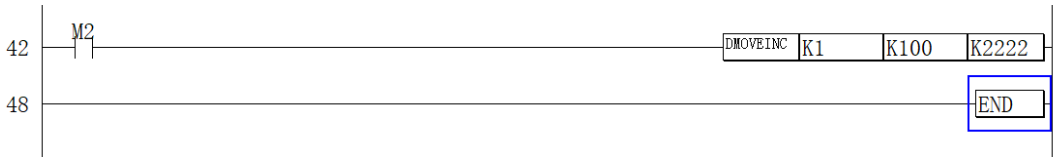
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	轴编号	—	K1-K12	—	实数
S2	运行速度	—	1~2147483647	—	实数
S3	相 对 目 标 地址	—	-214748364 ~ 2147483647	—	实数

注：不同机型针对不同机型，轴最大值不一样，如 HCR 系列最大四轴，S1 最大有效值为 k4; R8P,R8C 最大为 8 轴，R9 最大为 12 轴，下同。

示例

下图中，当 M2 闭合后，轴一进行相对运动，指令速度加速至 100pps ,保持，直至轴 1 的当前地址增加 2222，相对定位完成。具体的相对运动介绍参考定位篇。



报错相关：（参考定位篇。）

注意：（参考定位篇。）

4.17.9 DMOVEINT：中断定位模式

概要

在定位中，用 DMOVEINT 指令执行中断定长进给。

该指令可通过用户程序控制中断信号指令格式

16 位指令	-	32 位指令	DMOVEINT
执行类型	连续执行型		
指令格式	-	DMOVEABS S1 S2 S3	

操作数支持

操作数	位		字			双字		常数		
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E	
S1	—	—	—	—	—	—	—	○	—	
S2	—	—	—	○*1	○ 1	* 1	○* 1	○* 1	○	—
S3	—	—	—	○*1	○ 1	* 1	○* 1	○* 1	○	

*1 支持变址修饰

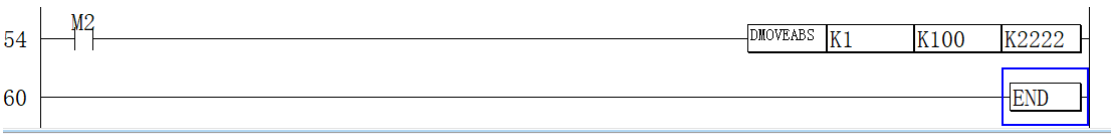
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	轴编号	—	K1-K12	—	实数
S2	运行速度	—	1~2147483647	—	实数
S3	目标地址	—	-214748364 ~ 2147483647	—	实数

注：不同机型针对不同机型，轴最大值不一样，如 HCR 系列最大四轴，S1 最大有效值为 k4；R8P,R8C 最大为 8 轴，R9 最大为 12 轴，下同。

示例

下图中，当 M2 闭合后，轴一进行中断定位运动，指令速度加速至 100pps，保持，直至收到中断信号，运动到轴地址为 2222 的位置，中断定位完成。具体的中断运动介绍参考定位篇。



报错相关：（参考定位篇。）
注意：（参考定位篇。）

4.17.10 DMOVEJOG：点动模式

概要

在定位中，使用 DMOVEJOG 指令执行点动运行。中途不可变更速度。

16 位指令	-	32 位指令	DMOVEJOG
执行类型	连续执行型		
指令格式	-	DMOVEJOG S1 S2 S3	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	—	—	—	—	○	—
S2	—	—	—	○*1	○ 1	* 1	○* 1	○* 1	—
S3	—	—	—	○*1	○ 1	* 1	○* 1	○* 1	—

*1 支持变址修饰

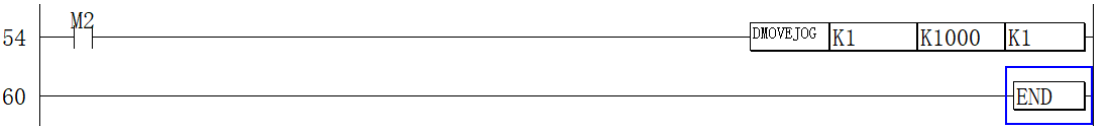
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	轴编号	—	K1-K12	—	实数
S2	运行速度	—	1~2147483647	—	实数
S3	旋转方向	—	0, 1	—	实数

注：不同机型针对不同机型，轴最大值不一样，如 HCR 系列最大四轴，S1 最大有效值为 k4; R8P,R8C 最大为 8 轴，R9 最大为 12 轴，下同。

示例

下图中，当 M4 闭合后，轴一进行点动运动，指令速度加速至 1000pps，保持反向运动。
具体的点动功能介绍参考定位篇。



报错相关：（参考定位篇。）

注意：（参考定位篇。）

4.17.11 DMOVEVEL:可变速运行

概要

在定位中，用可变速脉冲输出指令执行可变速速度运行。该指令在改变速度时，可以带加减速动作。

16 位指令	-	32 位指令	DMOVEVEL
执行类型	连续执行型		
指令格式	-	DMOVEVEL S1 S2 S3 S4	

操作数支持

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	—	—	—	—	○	—
S2	—	—	—	○*1	○* 1	—	—	○	—
S3	—	—	—	○*1	○* 1	—	—	○	—
S4	—	—	—	○*1	○* 1	○* 1	○* 1	○*1	

*1 支持变址修饰

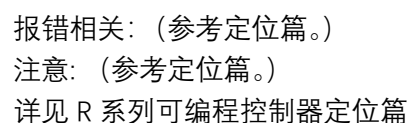
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	轴编号	—	K1-K12	—	实数
S2	加速时间	—	1~2147483647	—	实数
S3	减速时间	—	1~2147483647	—	实数
S4	指令速度 或者存储了数据的 字软元件 编号	—	-2147483648~ +2147483647	—	

注：不同机型针对不同机型，轴最大值不一样，如 HCR 系列最大四轴，S1 最大有效值为 k4; R8P,R8C 最大为 8 轴，R9 最大为 12 轴，下同。

示例

下图中，当 M4 闭合后，轴一进行可变速运动。初始时 D0 为 0 不运动，当 D0 中的值设定 1000 时，速度从 0pps 开始以 100 的加速度，加速至 1000pps，若 D0 的值改为 800，则速度以 200 的减速度减至 800pps。具体的可变速运动介绍参考定位篇。



4.18.1 DSETHSC: 高速计数器预置值设置

设置高速计数器当前值

0 M1 DSETHSC K1 KO

5 END

如上图所示,M2 闭合后, 会将计数器 1 的当前值设置为 0。

报错相关：（待添加）

注意: 此指令是立即执行的，当梯形图扫到此处会立即执行此指令而不是在 End 行执行此指令

4.18.2 HSCEN:高数计数器功能开启关闭指令

概要

高数计数器功能开启关闭指令

指令格式

16 位指令	HSCEN	32 位指令	-
执行类型	连续执行型		
指令格式	HSCEN S1 S2	-	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC LZ	K、H	E
S1	—	—	—	—	—	—	○	—
S2	○	—	—	—	—	—	○	—

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S	高速计数器 编号	K1-8	-	实数	-
D	传送软元件 值	0,1	-	实数	-

示例



如图所示, HSCEN 指令控制着计数器 1 的使能状态, 当 M2 闭合后, 如果 M1 为 ON, 计数器 1 就为使能, M2 为 OFF, 计数器 1 为失能。M2 不闭合, 计数器一直为失能。

报错相关：（待添加）

注意：（待添加）

4.18.3 HSCS 高速比较置位指令

概要

每次计数时比较高速计数器中计数的值与指定值，然后立即设置位软元件的指令。或者比较完成后进入目标高速计数中断程序。

指令格式

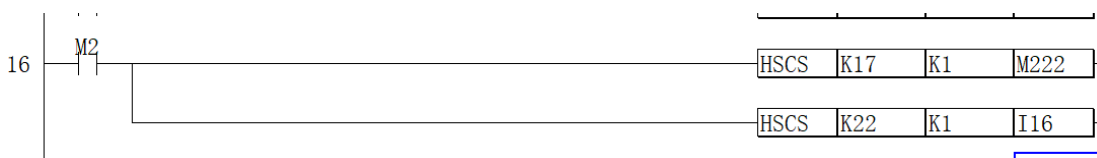
16 位指令	HSCS	32 位指令	D HSCS
执行类型	连续执行型		
指令格式	HSCS S1 S2 D	D HSCS S1 S2 D	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC LZ	K、H	E
S1	—	—	—	○	○		○	—
S2	—	—	—	—	—	—	○	—
D	○	—	—	—	—	—	—	—

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	比较值或包含比较值的软元件	-32768~32767	-2147483648~2147483647	实数	实数
S2	高速计数器编号	K1-8	K1-8	实数	实数
D	比较完成后置位的软元件	0,1	0,1	实数	实数

示例



在如上程序中，当 M2 闭合后，计数器 1 正常计数到 k17 的时候，M222 软元件会被置 ON，当计数到 K22 的时候，程序会跳转进入高速中断程序 I16，当执行完成后回到主程序。

报错相关：（待添加）

注意：（待添加）

4.22.5 HSCR 高速比较复位指令

概要

每次计数时比较高速计数器中计数的值与指定值，然后立即复位位软元件的指令。

指令格式

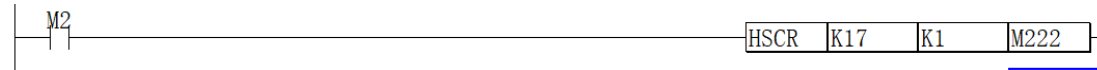
16 位指令	HSCR	32 位指令	D HSCR
执行类型	连续执行型		
指令格式	HSCR S1 S2 D	D HSCR S1 S2 D	

操作数	位		字		双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H E
S1	—	—	—	○	○			○ —
S2	—	—	—	—	—	—	—	○ —
D	○	—	—	—	—	—	—	—

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	比较值或包含比较值的软元件	-32768~32767	-2147483648~2147483647	实数	实数
S2	高速计数器编号	K1-8	K1-8	实数	实数
D	比较完成后复位的软元件	0,1	0,1	实数	实数

示例



在如上程序中，当 M2 闭合后，计数器 1 正常计数到 k17 的时候，M222 软元件会被复位，

报错相关：（待添加）

注意：（待添加）

4.18.4 DSPD 脉冲测量指令

概要

在指定时间内对输入的脉冲进行计数

指令格式

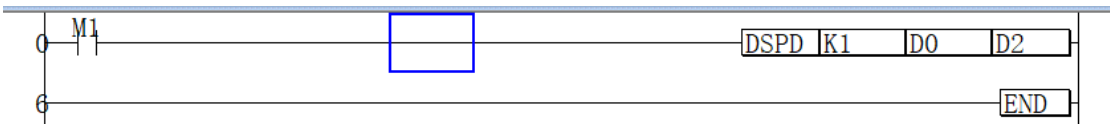
16 位指令	-	32 位指令	DSPD
执行类型	连续执行型		
指令格式	-	DSPD S1 S2 D	

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○	—	—	—	○	—
S2	—	—	—	○	○	○	○	○	—
D	—	—	—	○	○	○	○	—	—

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	脉冲输入计数的轴编号	-	K1-8	-	实数
S2	测定时间 (单位: ms)	-	-2147483648 ~ +2147483647*1	-	实数
D	存储的计算结果的起始地址编号 32 位	-	-2147483648 ~ +2147483647*1	-	实数

示例



在如上程序中，当 M1 闭合后，计数器 1 会以 D0 位测量周期进行脉冲密度的测量，结果将会保存到 D2 中

特殊继电器	功能	动作		默认	R/W
		ON	OFF		
SM1712	高速计数器通道 1 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1713	高速计数器通道 2 脉冲密度/转速测定完成	完成	未完成	OFF	R/W

SM1714	高速计数器通道 3 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1715	高速计数器通道 4 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1716	高速计数器通道 5 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1717	高速计数器通道 6 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1718	高速计数器通道 7 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1719	高速计数器通道 8 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1720	高速计数器通道 9 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1721	高速计数器通道 10 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1722	高速计数器通道 11 脉冲密度/转速测定完成	完成	未完成	OFF	R/W
SM1723	高速计数器通道 12	完成	未完成	OFF	R/W

报错相关：（待添加）

注意: DSPD 指令更改测试时间的情况下，每次测定时间结束时反映更改后的测定时间
使用 DSPD 指令时，请勿在测定时间内输入超出高速计数器环长 1/2 的脉冲。如果输出了，将无法测量脉冲密度，测定范围在 1-100KHZ。超出此范围测量值可能会不准确。

4.19 高速处理

4.19.1 REF / 输入输出刷新

概要

产生随机数的指令。

指令格式

16 位指令	REF(P)	32 位指令	-
执行类型	连续执行型、脉冲执行型		
指令格式	REF (P) D N	-	

可使用软元件

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
D	○*1*2	—	—	—	—	—	—	—	—
N	—	—	—	—	—	—	—	○	—

*1 支持变址修饰

*2 仅支持 X,Y

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
D	刷新的位软 元件(X、Y) 编号	0~32767	0~32767	实数	实数
N	刷新的位软 元件点数	0~32767	0~32767	实数	实数

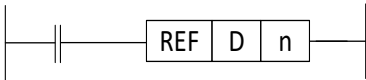
示例

1. 16 位运算(REF、REFP)

刷新输出(X,Y)时

刷新输出(D)开始的 n 点。

若应用中需要最新的输入信息以及希望立即输出运算结果时，可以使用立即刷新指令。



报错相关：待添加

注意：待添加

4.19.2 PWM/脉宽调试输出

概要

指定了脉冲的周期和 ON 时间的脉冲输出的指令。

指令格式

16 位指令	PWM	32 位指令	DPWM
执行类型	连续执行型		
指令格式	PWM S1 S2 D	DPWM S1 S2 D	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E

S1	—	—	—	○*1	—	—	—	—	—
S2	—	—	—	○*1	—	—	—	—	—
D	○*1*2	—	—	—	—	—	—	—	—

*1 支持变址修饰

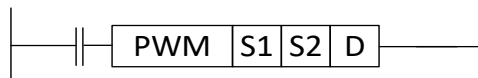
*2 仅支持 Y

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	脉宽(ms)数据或是保存数据的字软元件编号	1~32767	1~2147483647	实数	实数
S2	周期(ms)数据或是保存数据的字软元件编号	1~32767	1~2147483647	实数	实数
D	输出脉冲的软元件(Y)编号	0~3	0~3	实数	实数

示例

1. 16 位运算(PWM)



将 S1 中指定的 ON 时间及 S2 中指定的周期的脉冲输出到 D 中指定的输出目标中由梯形图中前的触点，决定 PWM 使能/使能

在 S1 中指定脉宽 t。

设定范围: 2 ~ 32,767μs

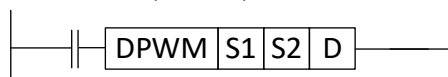
• 在 S2 中指定周期 T 0。

设定范围: 5 ~ 32,767μs

• 在 D 在中指定输出脉冲的 Y 编号。

设定范围: Y000、Y001、Y002、Y003

2. 32 位运算(DPWM)



将 S1 中指定的 ON 时间及 S2 中指定的周期的脉冲输出到 D 中指定的输出目标中由梯形图中前的触点，决定 PWM 使能/使能

在 S1 中指定脉宽 t。

设定范围: 2 ~ 2147483647μs

• 在 S2 中指定周期 T 0。

设定范围: 5 ~ 2147483647μs

报错相关: (待添加)

注意: S1 必须 ≤ S2, 使用此功能前必须先确定本机型的相应高速输出端子是否配置为

PWM 脉冲输出

4.20 凸轮顺控

4.20.1 ABSD/凸轮控制绝对方式

概要

对应计数器的当前值，产生多个输出模式的指令。

指令格式

16 位指令	ABSD	32 位指令	DABSD
执行类型	连续执行型		
指令格式	ABSD S1 S2 D N	DABSD S1 S2 D N	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1*2	○	○ 6	*○* 6	—	—
S2	—	—	—	○*1*3	—	○ 6	*—	—	—
D	○*1*4	○*5	—	—	—	—	—	—	—
N	—	—	—	○	○	—	—	○	—

*1 支持变址修饰

*2 仅支持 TCDR

*3 仅支持 C，SD（只能在 32 位指令中填写高速计数器当前值软元件,实时读取）

*4 仅支持 YM

*5 仅支持 D.X

*6 仅限 32 位指令使用

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存数据表格（上升沿、下降沿）的起始软元件编号	-32768~32767	-2147483648~- 2147483647	S16	S32
S2	与数据表格比较的当前值监控用计	-32768~32767	-2147483648~- 2147483647	S16	S32

	数器编号				
D	输出的起始位软元件编号	-	-	Bit	Bit
N	表格的行数以及输出的位软元件的点数	$[1 \leq n \leq 64]$	$[1 \leq n \leq 64]$	S16	S16

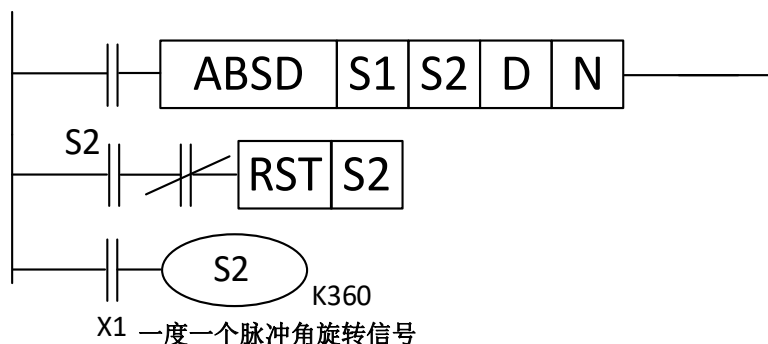
示例

1. 16 位运算(ABSD)

平台旋转 1 次(0 ~ 360 度)期间, 使输出 ON/OFF, 以此为例进行说明。

(1 度 1 个脉冲的旋转角度信号)

将 S1 开始的 n 行表格数据(占用 n 行×2 点)与计数器的当前值 S2 做比较, 在旋转 1 次期间, 对从 D 开始的连续 n 点输出进行 ON/OFF 控制。



1) 先使用传送指令, 在 S1 ~ S2 + 2n + 1 中写入如下所示的数据。

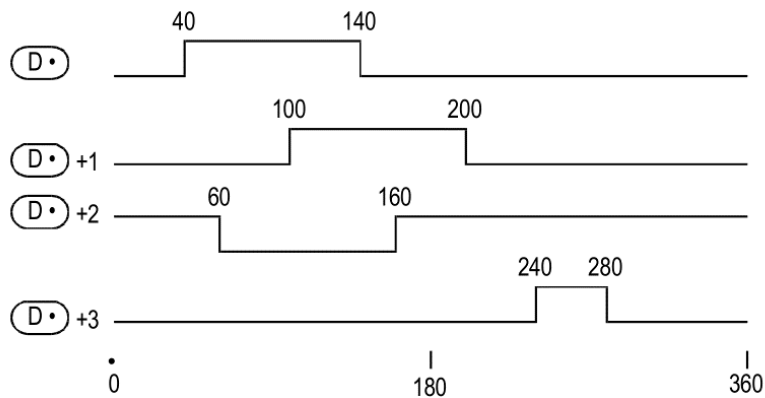
上升点		下降点		对象输出
	数据值(例)		数据值(例)	
S1	40	S1+1	140	D
S1+2	100	S1+3	200	D+1
S1+4	160	S1+5	60	D+2
S1+6	240	S1+7	280	D+3
...	-	...	-	...
S1+2N	-	S1+2N+1	-	D+N-1

例如, 上升点数据在偶数编号的软元件, 下降点数据在奇数编号的软元件中, 以 16 位数据进行保存。

2) 输出模式

指令输入为 ON 后, 以 D 开始的 n 点也如下进行变化。

各上升点·下降点, 可通过改写 S1~S2+N*2 的数据以个别进行变更。

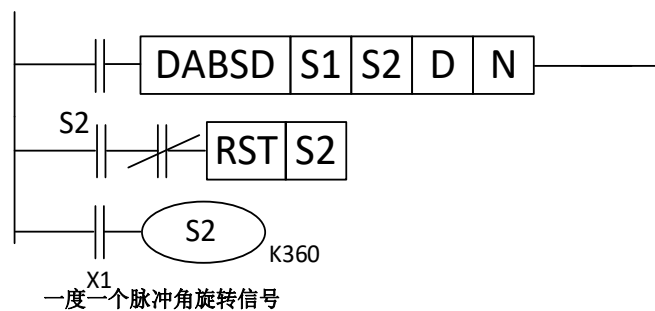


2. 32 位运算(DABSD)

平台旋转 1 次(0 ~ 360 度)期间，使输出 ON/OFF，以此为例进行说明。

(1 度 1 个脉冲的旋转角度信号)

将[S1+1, S1]开始的 n 行表格数据(占用 n 行×4 点)，与计数器的当前值 S2 做比较，在旋转 1 次期间，对从 D 开始的连续 n 点输出进行 ON/OFF 控制。



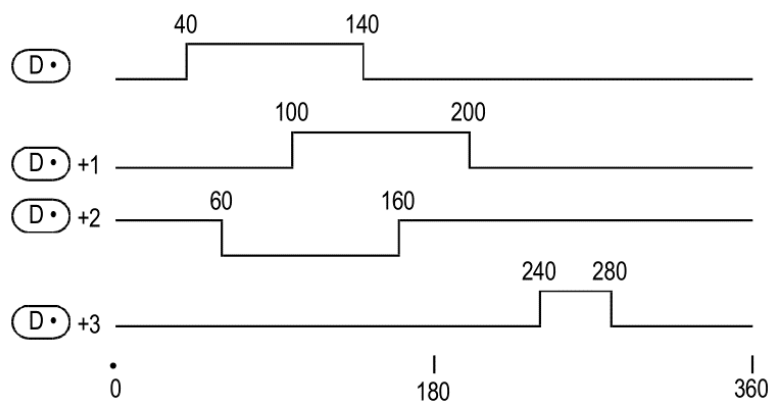
1) 先使用传送指令，在[S1, S1+1] ~ [S1+4n+2, S1 +4n+3]中写入如下所示的数据。

上升点		下降点		对象输出
	数据值(例)		数据值(例)	
S1+1,S1	40	S1+3,S1+2	140	D
S1+ ,S1+4	100	S1+7,S1+6	200	D+1
S1+ ,S1+8	160	S1+ 11,S1+10	60	D+2
S1+ ,S1+12	240	S1+15,S1+14	280	D+3
...	-	...	-	...
S1+4N+1,S1+4N		S1+4N+3,S1+4N+2		D+N-1

例如，上升点数据在偶数编号的软元件，下降点数据在奇数编号的软元件中，以 32 位数据进行保存。

指令输入为 ON 后，以 D 开始的 n 点也如下进行变化。

各上升点·下降点，可通过改写[S1+1, S1] ~ [S1+(n×2)+3, S1 +(n×2)+2]的数据以个别进行变更。



注意要点

1. 高速计数器的指定

DABSD 指令中也可以指定高速计数器 S2。

但是此时，对于计数器的当前值，在输出模式中会由于运算周期而造成响应延迟。

2. 在 S1 中指定位元件的位数时

1) 软元件编号

请指定 16 的倍数(0、16、32、64...)。

2) 位数

– ABSD(16 位运算)时仅为 K4

– DABSD(32 位运算)时仅为 K8

3. 其他的注意事项

- 由 n 的值决定对象的输出点数。(1≤n≤64)
- 即使指令输入为 OFF，输出也不改变。

错误相关：

6004 软元件使用超过范围

6706 运算错误或参数的值超过可设置范围

6710 //命令中多个参数设置的范围重叠

4.20.2 INCD/凸轮控制相对方式

概要

使用一对计数器，产生多个输出模式的指令。

指令格式

16 位指令	INCD	-	-
执行类型	连续执行型		
指令格式	INCD S1 S2 D N	-	

可使用软元件

操作数	位	字	双字	常数
-----	---	---	----	----

	X、Y、M、L、S M、F、B、SB、S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	○	○*1*2	○	—	—	—	—
S2	—	—	—	○*1*3	—	—	—	—	—
D	○*1*4	○*5	—	—	—	—	—	—	—
N	—	—	—	○*1	○	—	—	○	—

*1 支持变址修饰

*2 仅支持 TCDR

*3 仅支持 C

*4 仅支持 YM

*5 仅支持 D.X

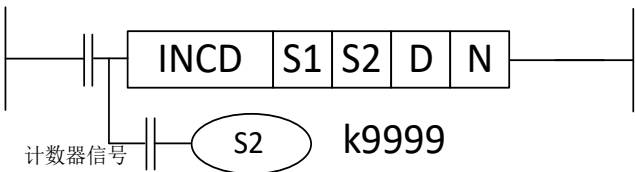
可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	保存设定值的起始字软元件编号	-32768~32767	-	S16	-
S2	监控当前值用的计数器的起始编号	-32768~32767	-	S16	-
D	输出的起始位软元件编号	-	-	BIT	-
N	输出的位软元件的点数	[1≤n≤64]	-	S16	-

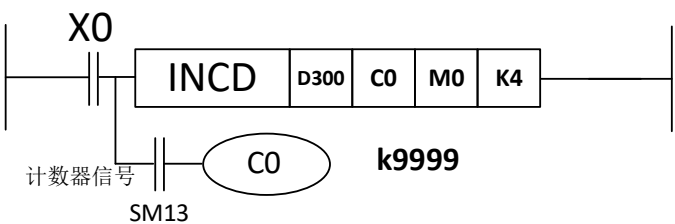
示例

1. 16 位运算(INCD)

对 S1 开始的 n 行表格数据(占用 n 行×1 点), 与计数器的当前值 S2 进行比较, 如果一致, 则复位, 并依次对输出进行 ON/OFF 控制



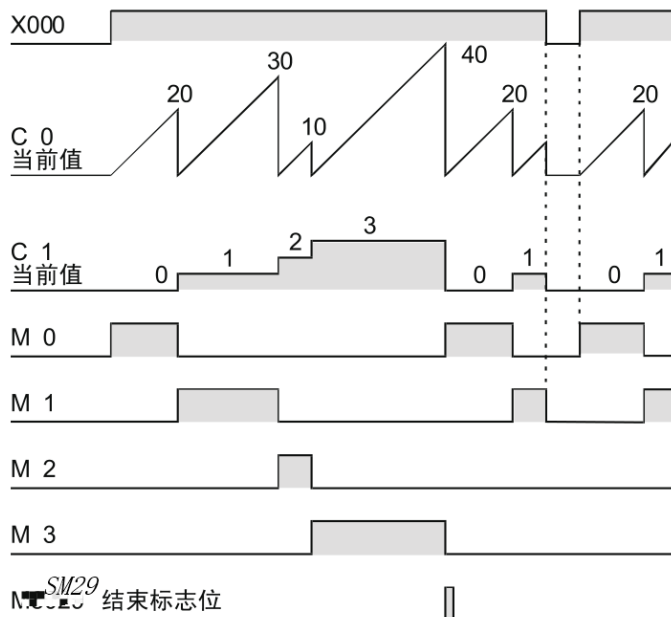
动作



1) 时序图

使用传送指令, 写入下表中的数据。

保存软元件		输出	
	数据值(例)		例如
S1	D300=20	D	M0
S1+1	D301=30	D+1	M1
S1+2	D302=10	D+2	M2
S1+3	D303=40	D+3	M3
...	-	...	-
S1+N-1	-	D+N-1	-



-) 指令触点如果为 ON，则 M0 输出也为 ON。
- 3) 当 C0 的当前值达到比较值 D300 时，输出(M0)被复位，工序计数器 C1 的计数值被+1，计数器 C0 的当前值也被复位。
- 4) 接下来的输出 M1 为 ON。
- 5) C0 当前值与比较值 D301 比较，如果达到比较值，则输出 M1 被复位，工序计数器 C1 的计数值被+1，计数器 C0 的当前值也被复位。
- 6) 同样地方式一直要比较到 n(K4)指定的点数为止。(1≤n≤64)
- 7) 在 n 指定的最后工序结束后，执行结束标志位 M8029 保持 1 个运算周期为 ON。
- 由于 M8029 是多个指令的指令使用执行结束的标志位，所以直接在指令的后面作为触点使用，请务必作为该指令专用的结束标志位。
- 8) 回到最初，重复输出。

注意要点

1. 在 s1 中指定位软元件的位数时
软元件编号中，请指定 16 的倍数(0、16、32、64…)。

错误相关：

- 6004 软元件使用超过范围
- 6706 运算错误或参数的值超过可设置范围
- 6710 命令中多个参数设置的范围重叠

4.21 模块访问指令

4.21.1FROM/BFM 的读出

概要
将特殊功能单元/模块的缓冲存储区(BFM)中的内容读入可编程控制器的指令。

指令格式

16 位指令	FROM(P)	32 位指令	DFROM (P)
执行类型	连续执行型		
指令格式	FROM (P) M1 M2 D N	DFROM (P)M1 M2 D N	

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z		LC	LZ	K、H	E
M1	—	—	—	○*1	—	—	—	—	○	—
M2	—	—	—	○*1	—	—	—	—	○	—
D	—	—	—	○*1	—	—	—	—	—	—
N				○*1					○	

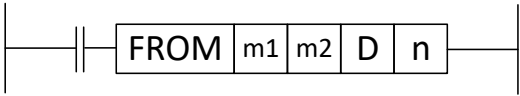
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
M1	特殊功能单元/模块的单元号	0~7	0~7	实数	实数
M2	传送源缓冲存储区(BFM)编号	0~32766	0~32766	实数	实数
D	传送目标的软元件编号	1~32767	1~32767	实数	实数
N	传送点数	1~32767	1~32767	实数	实数

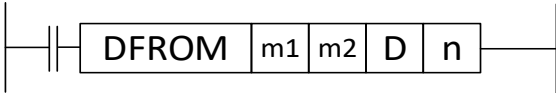
示例

1.16 位 BFM 数据读取



从扩展中指定的智能功能模块内的 (M1) 中指定的缓冲存储器 (M2) 读取 (n) 字的数据，存储到 (D) 中指定的软元件之中

2. 32位BFM数据读取



从扩展中指定的智能功能模块内的 (M1) 中指定的缓冲存储器 (M2) 读取 (n)*2字的数据，存储到 (D) 中指定的软元件之中

报错相关：（待添加）

注意：FROMP仅指令之前的运算结果为OFF-ON时执行

4.21.2 TO/BFM 的写入

概要

将数据从可编程控制器中写入到特殊功能单元/模块的缓冲存储区(BFM)中的指令。

指令格式

16 位指令	TO(P)	32 位指令	DTO (P)
执行类型	连续执行型		
指令格式	TO (P) M1 M2 D N		D TO (P)M1 M2 D N

可使用软元件

操作数	位		字				双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D. X、W. X、S D. X、SW. X、 R. X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z		LC	LZ	K、H	E
M1	—	—	—	○*1	—	—	—	—	○	—
M2	—	—	—	○*1	—	—	—	—	○	—
D	—	—	—	○*1	—	—	—	—	—	—
N				○*1					○	

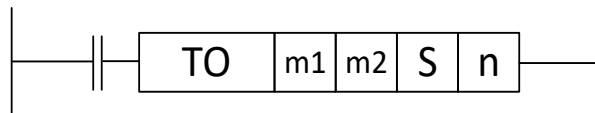
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
M1	特殊功能单元/模块的单元号	0~7	0~7	实数	实数
M2	写入数据的缓冲存储器的起始地址	0~32766	0~32766	实数	实数
D	写入数据或存储了写入数据的软元件起始编号	1~32767	1~32767	实数	实数
N	写入数据数	1~32767	1~32767	实数	实数

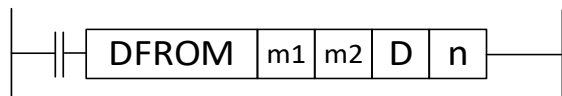
示例

1.16 位 BFM 数据写入



将(S)中指定的软元件开始(n)点的数据，写入到(M1)中指定的智能功能模块内的缓冲存储器的(M2)中指定的地址以后

2. 32位BFM数据读取



将(S)中指定的软元件开始(n)*2点的数据，写入到(M1)中指定的智能功能模块内的缓冲存储器的(M2)中指定的地址以后

报错相关：（待添加）

注意：（待添加）

4.22 MODBUS

4.22.1 ADPRW 自由通信

概要

MODBUS 主站通信指令

指令格式

16 位指令	ADPRW(P)	32 位指令	-
执行类型	连续执行型		
指令格式	ADPRW(P) S1 S2 S3 S4 S5/D1 D2	-	

可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	○	—
S2	—	—	—	○*1	—	—	—	○	—
S3	—	—	—	○*1	—	—	—	○	—
S4	—	—	—	○*1	—	—	—	○	—
D1	○*1	—	—	○*1	—	—	—	○	—
D2	○*1	○	—	—	—	—	—	○	—

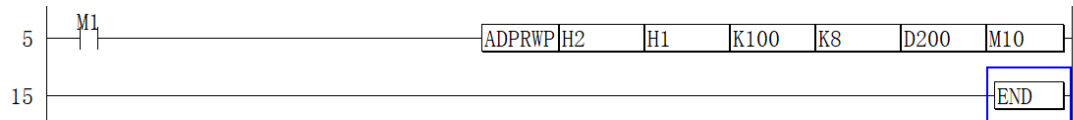
*1 支持变址修饰

可使用操作数

操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	从站站号	1~247	-	实数	—
S2	功能代码	-	-	实数	—

S3	MODBUS 地址	-	-	实数	-
S4	访问点数		-	实数	-
D	存储数据的软元件起始编号	0~32767	-	实数	-
N	输出指令执行状态的起始位软元件编号	0~32767	-	实数	-

示例



目标从站站号为S1，根据参数S3 S4 D1执行动作。上图中对2号从站执行动作，具体的内容为从地址K100读取8点的线圈寄存器，读取结果储存到D200开始的8点软元件中，执行结果显示在以M10为起始的3点软元件中。

报错相关：（待添加）

- 注意：
1. 进行广播的情况下，请将从站站号设为0
 2. 通信执行状态D2依照ADPRW命令的通信执行中/正常结束/异常结束的各状态进行输出
 3. 在起始处占用3点的D2的指定软元件，请注意不要与其他寄存器重复。

4.23 自由通信

4.23.1 发送数据 SEND

概要

将本机内指定的数据写到目的设备数据的指令，发送个数不包括起始符和终止符。

指令格式

16 位指令	SEND	32 位指令	
执行类型	连续执行型		
指令格式	SENDS1 S2 S3		

可使用软元件

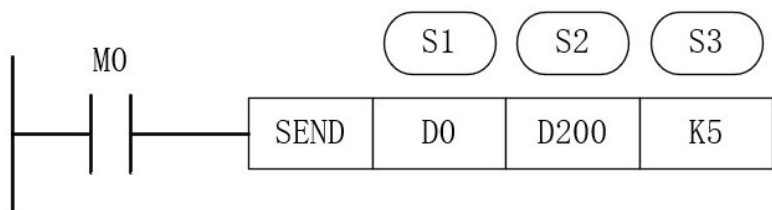
操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、S	D、X、W、X、S D、X、SW、X、 R、X	KnX、KnY、Kn M、KnL、KnS M、KnF、KnB、 KnSB、KnS	T、ST、C、 D、W、SD、S W、R	Z	LC	LZ	K、H	E

S1	—	—	—	○*1	—	—	—	—	—
S2	—	—	—	○*1	—	—	—	○	—
S3	—	—	—	—	—	—	—	○	—

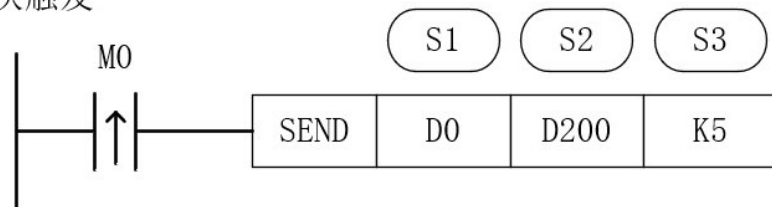
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	发送软元件首地址	—	—	实数	—
S2	需要发送的个数	K1~K512	—	实数	—
S3	通信端口 (串 口 号)	K1~K5	—	实数	—

示例：

连续触发 ①



单次触发 ②



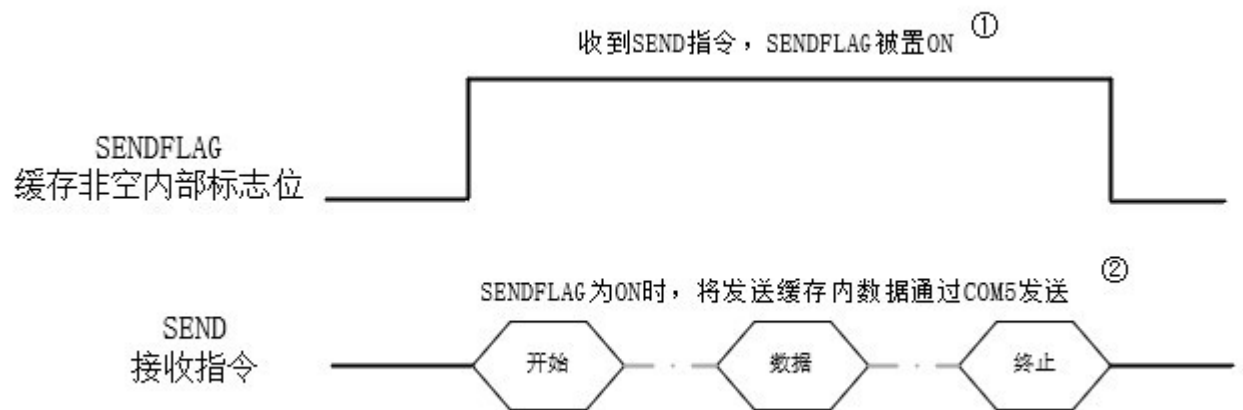
- S1:发送数据地址从 D0 开始
- S2:发送 D200(数值) 个数据 (以字节为单位), 若 D200=0, 不发送。16 位模式下, 要发送 D1~D18, 则 D200=36, 若 S2 为奇数, 则最后一个 16 位寄存器只发送低 8 位。8 位模式下, 要发送 D1~D18, 则 D200=18。
- S3:指定 COM5 发送。范围 K1~K5, 代指 COM1~COM5.
- 相应端口未使能下, 使用 SEND 指令, PLC 系统应报故障码:
7601: 自由协议未设定, SEND 指令不可执行

报错相关: (待添加)

注意:

- ① K1-K5 任一端口未设定自由协议, 使用 SNED 指令, 均报 7601 故障码
- ② 若发送中途 PLC 切换到 STOP, 当前发送立即停止, 同时后台自动清空发送缓存。

时序图：



- ① 接收到 SEND 指令时，自由格式驱动任务被唤醒，在自由格式驱动任务内，将缓存队列内数据通过 COM5 发送，根据 IDE 设置返回发送结果。当自由格式驱动端口发送与接收空闲时，自由格式驱动任务被挂起。
- ② 如果缓存队列内有多帧数据需要发送，则全部数据发送完成后任务才会被挂起。

4.23.2 接收数据 REC

概要

将目的设备的数据写到本机内指定地址的指令，接收指令不影响其他逻辑指令的执行
指令格式

16 位指令	REC	32 位指令	
执行类型	连续执行型		
指令格式	REC S1 S2 S3		

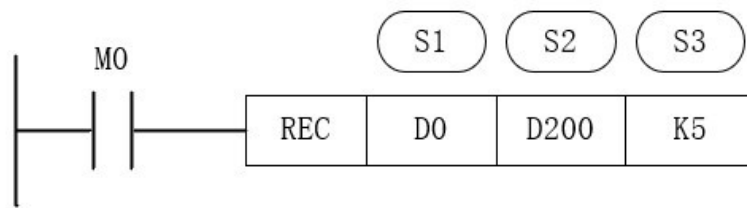
可使用软元件

操作数	位		字			双字		常数	
	X、Y、M、L、S M、F、B、SB、 S	D.X、W.X、S D.X、SW.X、 R.X	KnX、KnY、Kn M、KnL、KnS M、KnF、Kn B、KnSB、Kn S	T、ST、C、 D、W、SD、 SW、R	Z	LC	LZ	K、H	E
S1	—	—	—	○*1	—	—	—	—	—
S2	—	—	—	○*1	—	—	—	○	—
S3	—	—	—	—	—	—	—	○	—

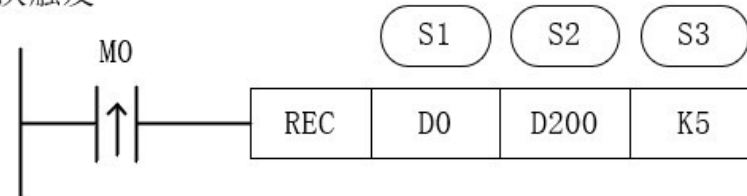
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	发送软元件首地址	—	—	实数	—
S2	需要接收的个数	K1~K512	—	实数	—
S3	通信端口 (串口号)	K1~K5	—	实数	—

示例：

连续触发 ①



单次触发 ②

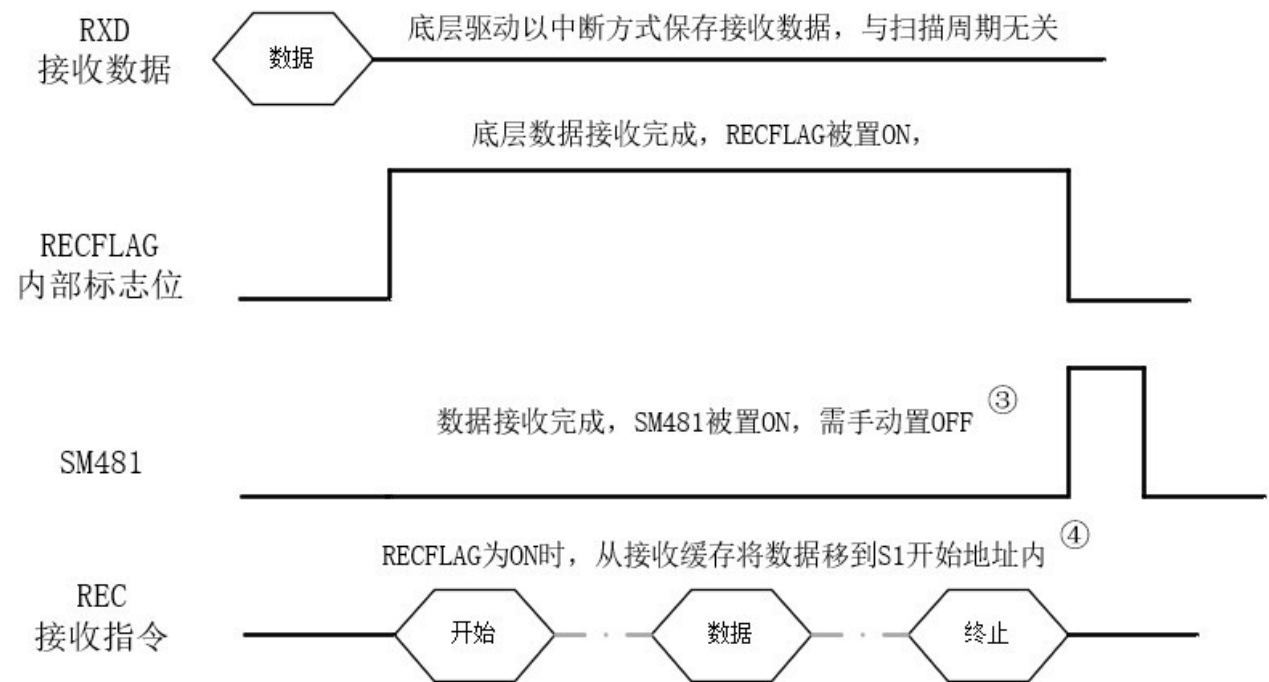


- S1:接收数据地址从 D0 开始
- S2:接收 D200(数值) 个数据 (以字节为单位), 若 D200=0, 不接收。16 位模式下要接收 D1~D18, 则 D200=36, 若 S2 值为奇数, 则接收到的最后一个字节放在最后 16 位字节的低 8 位, 高 8 位 0X00 填充。8 位模式下要接收 D1~D18, 则 D200=18。
- S3:指定 COM5 接收。范围 K1~K5, 代指 COM1~COM5
- 相应端口未使能下, 使用 REC 指令, PLC 系统应报故障码:
7602: 自由协议未设定, REC 指令不可执行

报错相关: (待添加)

注意: K1~K5 任一端口未设定自由协议时, 使用 REC 指令, 均报 7602 故障码

时序图：



- ① M0 常 ON 时，每扫描周期 REC 指令被触发一次。REC 指令内部判断 RECFLAG，RECFLAG 为 OFF 时，S1 开始的地址内数据为 0x00。当 RECFLAG 为 ON 时，接收缓存数据被取出，清除接收缓存，RECFLAG 置 OFF，SM481 置 ON。
- ② M0 闭合一次，REC 指令被触发一次。REC 指令内部判断 RECFLAG，RECFLAG 为 OFF 时，S1 开始的地址内数据为 0x00。当 RECFLAG 为 ON 时，接收缓存数据被取出，清除接收缓存，RECFLAG 置 OFF，SM481 置 ON。
- ③ 一帧数据接收完(包括接收错误情况，错误代码在 SD481 中查看)或者接收超时(指超过重发次数)，SM481 会被置 ON。
- ④ 当用户需要接收的数据长度大于实际接收到的数据长度时，差值用 0x00 填充。当用户需要接收到的数据长度小于实际接收到的数据长度时，多出的数据将会从首位开始填充接收数组。

注意：

① 接收超时会发生“重试”动作，超时时间可设置范围 0~6000mS，其中设置为 0 时表示该通道不需要接收或者接收在发送前。

② 若接收途中 PLC 切换到 STOP 状态，则该端口停止接收数据，并后台清空接收缓存。

4.23.3 释放串口 RLS

概要

将指定的串口资源进行释放的指令。在进行自由格式通信时，如果超时时间设定过长，或者重复次数过多，可通过 RLS 指令立即释放占用的串口资源，清空错误标志(串行通信发生重试/串行通信发生超时)和清空缓存以便进行其他通信操作。

指令格式

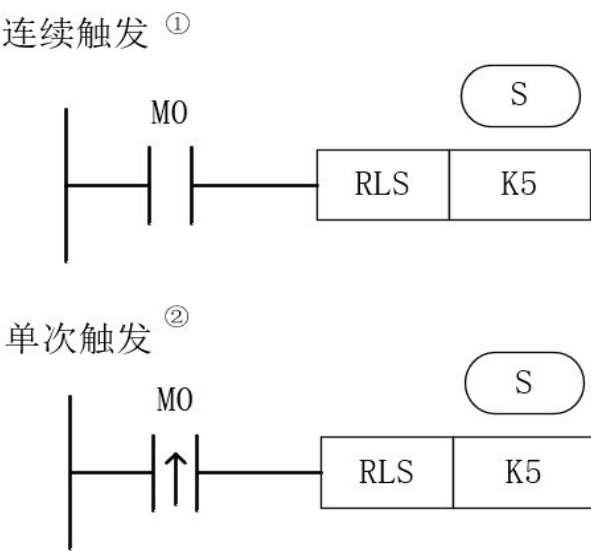
16 位指令	RLS	32 位指令	
执行类型	连续执行型		
指令格式	RLS S		

可使用软元件

操作数	位		字		双字		常数	
	X、Y、M、L、S	D.X、W.X、S	KnX、KnY、KnM、KnL、KnS	T、ST、C、D、W、SD、Z	LC	LZ	K、H	E
	M、F、B、SB、S	D.X、SW.X、R.X	M、KnF、KnB、KnSB、KnS	SW、R				
S	—	—	—	—	—	—	○	—

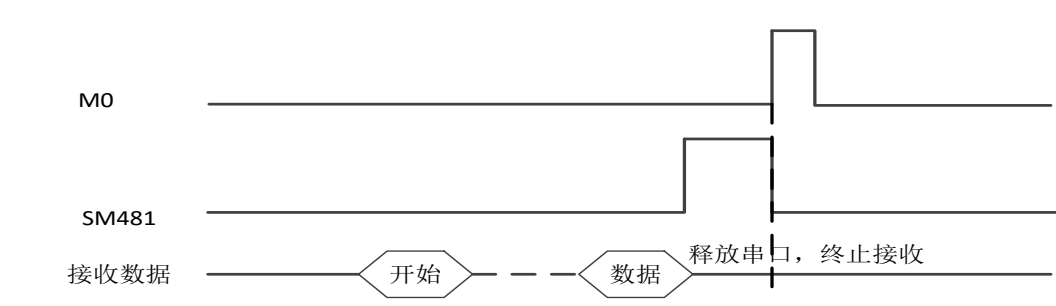
操作数	内容	数据范围		数据类型	
		16 位范围	32 位范围	16 位	32 位
S1	通 信 串 口 号	—	—	实数	—

示例：

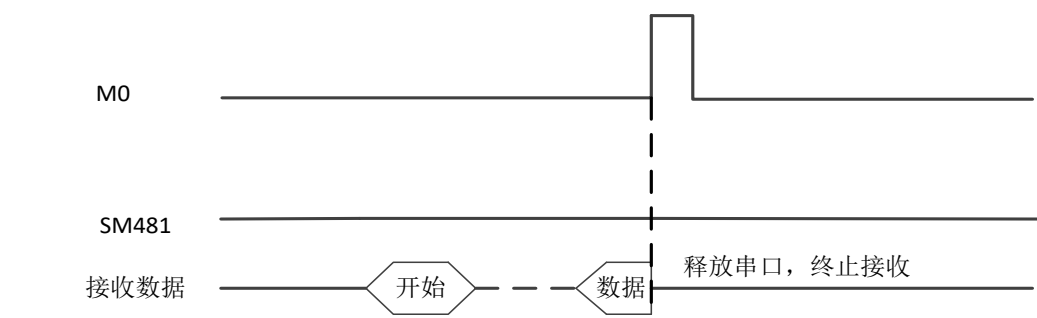


- S:释放 COM5。范围 K1~K5,代指 COM1~COM5。
释放串口时，将接收完成标志位“SM481”或正在发送标志“SM480”置 OFF。
- 相应端口未使能下，使用 RLS 指令，PLC 系统应报故障码：
7603：自由协议未设定，RLS 指令不可执行
报错相关：（待添加）
注意：K1~K5 任一端口未设定自由协议时，使用 RLS 指令，均报 7603 故障码

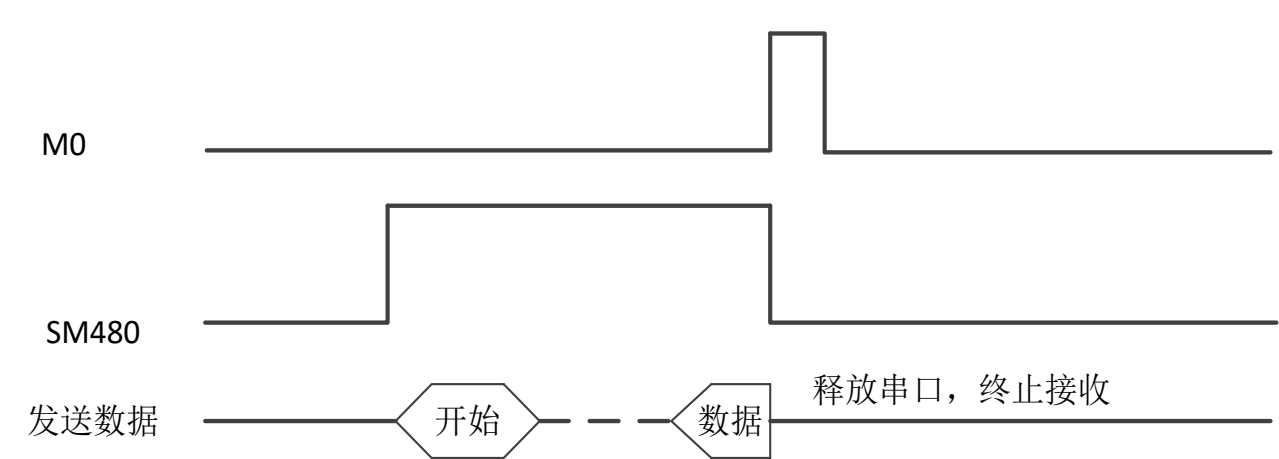
接收已完成：



接收未完成：



处于发送时使用释放指令的时序图：



4.24 高速计数

4.24.1 R 系列高速计数功能

概要

使用 CPU 模块的通用输入端子，对高速脉冲的输入数进行计数。也可以对输入脉冲数进行比较输出

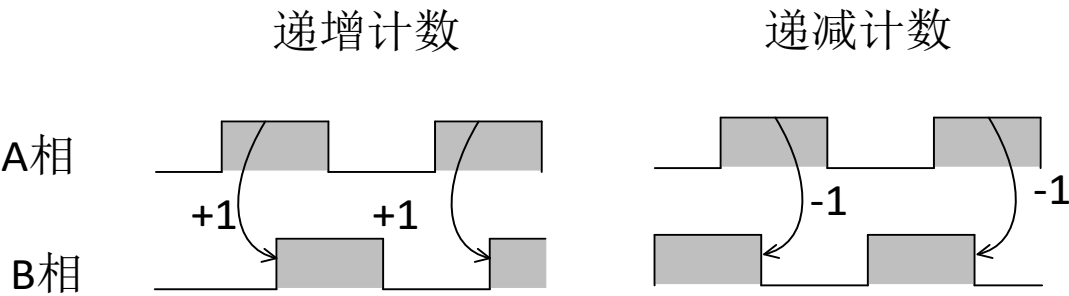
■普通模式

作为一般的高速计数器使用时选择此项。

- (1) CPU 模块目前同时支持 4 个 32 位高速计数器通道。
- (2) 每个高速计数器通道在都可读取 1~200kHz 响应频率的脉冲
- (3) 计数范围的最大值和最小值。最大值和最小值的指定范围是 (-2147483648~2147483647)，且最大值要大于最小值。
- (4) 每个通道的高速计数器脉冲输入模式有：2 相 1 倍频、2 相 2 倍频、2 相 4 倍频、1 相 2 输入、1 相 1 输入（软）、1 相 1 输入（硬）

2 相 1 倍频：

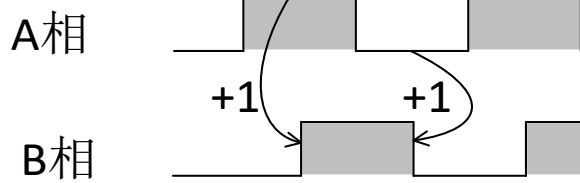
动作	计数时刻
递增计数	A 相输入 ON 而 B 相输入 OFF→ON 时递增 1
递减计数	A 相输入 ON 而 B 相输入 ON→OFF 时递减 1



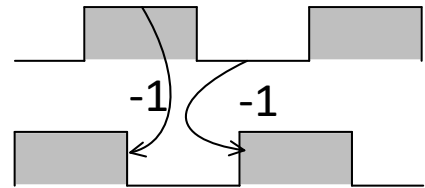
2 相 2 倍频：

动作	计数时刻
递增计数	A 相输入 ON 而 B 相输入 OFF→ON 时递增 1 A 相输入 OFF 而 B 相输入 ON→OFF 时递增 1
递减计数	A 相输入 ON 而 B 相输入 ON→OFF 时递减 1 A 相输入 OFF 而 B 相输入 OFF→ON 时递减 1

递增计数



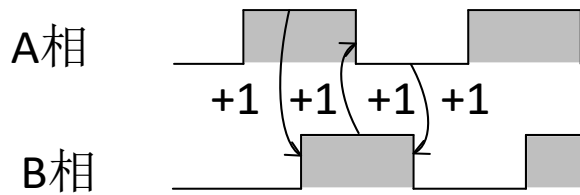
递减计数



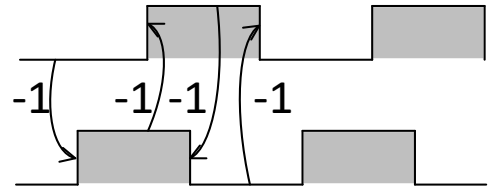
2 相 4 倍频:

动作	计数时刻
递增计数	B 相输入 OFF 而 A 相输入 OFF→ON 时递增 1 A 相输入 ON 而 B 相输入 OFF→ON 时递增 1 B 相输入 ON 而 A 相输入 ON→OFF 时递增 1 A 相输入 OFF 而 B 相输入 ON→OFF 时递增 1
递减计数	A 相输入 OFF 而 B 相输入 OFF→ON 时递减 1 B 相输入 ON 而 A 相输入 OFF→ON 时递减 1 A 相输入 ON 而 B 相输入 ON→OFF 时递减 1 B 相输入 OFF 而 A 相输入 ON→OFF 时递减 1

递增计数



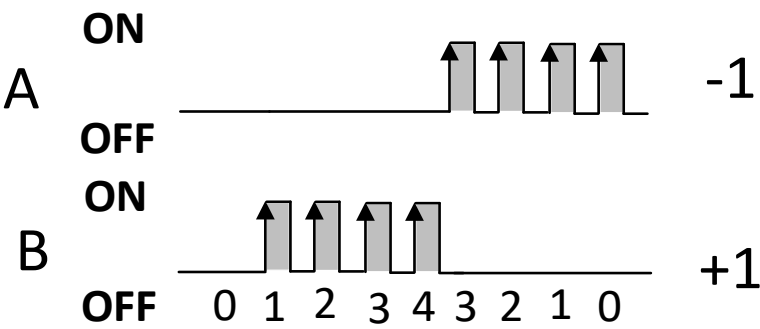
递减计数



1 相 2 输入:

动作	计数时刻
递增计数	A 相输入 OFF 而 B 相输入 OFF→ON 时递增 1
递减计数	B 相输入 OFF 而 A 相输入 OFF→ON 时递减 1

递减计数

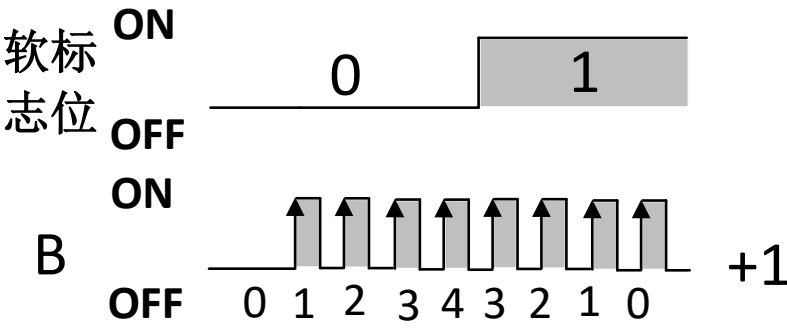


递增计数

1 相 1 输入（软）：

动作	计数时刻
递增计数	一相一输入软元件标志位为 0 时而 B 相输入 OFF→ON 时递增 1
递减计数	一相一输入软元件标志位为 1 时而 B 相输入 OFF→ON 时递增 1

递减计数

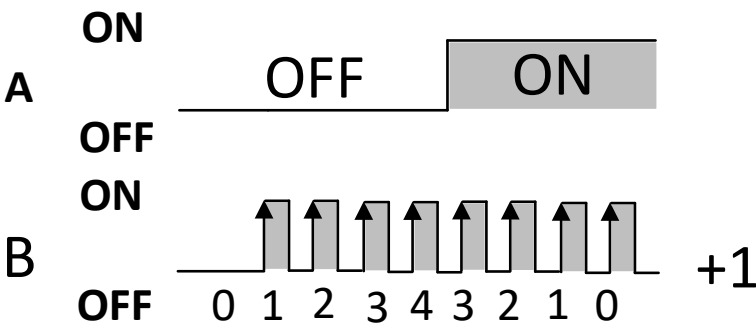


递增计数

1 相 1 输入（硬）：

动作	计数时刻
递增计数	A 相硬件信号输入为 OFF 时而 B 相输入 OFF→ON 时递增 1
递减计数	A 相硬件信号输入为 ON 时而 B 相输入 OFF→ON 时递增 1

递减计数



递增计数

(5) 每一路高速计数器占用 2 个输入点， 分别作为 A 相、B 相。

R1/R2.R8/R8C/R8P 都包以下 4 组高速计数器

通道	高速计数器类型	X0	X1	X2	X3	X4	X5	X6	X7
通道 1	1 相 1 输入 (S/W)		B						
	1 相 1 输入 (H/W)	A	B						
	1 相 2 输入	A	B						
	2 相 2 输入	A	B						
通道 2	1 相 1 输入 (S/W)				B				
	1 相 1 输入 (H/W)			A	B				
	1 相 2 输入			A	B				
	2 相 2 输入			A	B				
通道 3	1 相 1 输入 (S/W)						B		
	1 相 1 输入 (H/W)					A	B		
	1 相 2 输入					A	B		
	2 相 2 输入					A	B		
通道 4	1 相 1 输入 (S/W)								B
	1 相 1 输入 (H/W)							A	B
	1 相 2 输入							A	B
	2 相 2 输入							A	B

R8/R8C/R8P 还有 X10-X17,4 组高速计数端子

通道	高速计数器类型	X10	X11	X12	X13	X14	X15	X16	X17
通道 5	1 相 1 输入 (S/W)		B						
	1 相 1 输入 (H/W)	A	B						
	1 相 2 输入	A	B						
	2 相 2 输入	A	B						
通道 6	1 相 1 输入 (S/W)				B				
	1 相 1 输入 (H/W)			A	B				
	1 相 2 输入			A	B				
	2 相 2 输入			A	B				
通道 7	1 相 1 输入 (S/W)						B		
	1 相 1 输入 (H/W)					A	B		
	1 相 2 输入					A	B		
	2 相 2 输入					A	B		
通道 8	1 相 1 输入 (S/W)								B
	1 相 1 输入 (H/W)							A	B
	1 相 2 输入							A	B
	2 相 2 输入							A	B

注：

A: A 相输入（在 1 相 1 输入（H/W）时，变成方向切换）

B: B 相输入

(6) 可以设置输入端子的输入滤波时间，滤波时间单位是 μs 。

(7) 使用梯形图命令可以启动/停止高速计数器。

(8) 使用梯形图命令可以启用/禁止高速计数器输入比较功能，当计数器当前值与比较值一致时，输出结果 ON 或者 OFF 输出至指定输出端口，且输出结果保持。

(9) 输入比较功能可提供 16 位比较值指定和 32 位比较值指令。

(10) 输入比较功能可选择输出至 Y0~Y7，不同计数器通道不可选择同一输出端口。

通过 IDE 高速 I/O 界面可设置比较输出引脚 Y0~Y7。

(11) 每个高速计数器有 16 个比较值存放空间，梯形图命令可设置比较值，并通过前置的触点(使能/除能)启动·禁止该值的比较输出动作。

(12) 在 1 相 1 输入（软）模式中，有对应软元件可以切换计数方向。

注意：相关指令请查阅 高速计数和脉冲测量章节

相关软元件：

特殊继电器	功能	动作		默认	R/W
		ON	OFF		
SM1600	高速计数器通道 1 使能状态	计数器使能了	计数器失能了	OFF	R
SM1601	高速计数器通道 2 使能状态	计数器使能了	计数器失能了	OFF	R
SM1602	高速计数器通道 3 使能状态	计数器使能了	计数器失能了	OFF	R
SM1603	高速计数器通道 4 使能状态	计数器使能了	计数器失能了	OFF	R
SM1604	高速计数器通道 5 使能状态	计数器使能了	计数器失能了	OFF	R
SM1605	高速计数器通道 6 使能状态	计数器使能了	计数器失能了	OFF	R
SM1606	高速计数器通道 7 使能状态	计数器使能了	计数器失能了	OFF	R
SM1607	高速计数器通道 8 使能状态	计数器使能了	计数器失能了	OFF	R
SM1608	高速计数器通道 9 使能状态	计数器使能了	计数器失能了	OFF	R
SM1609	高速计数器通道 10 使能状态	计数器使能了	计数器失能了	OFF	R
SM1610	高速计数器通道 11 使能状态	计数器使能了	计数器失能了	OFF	R
SM1611	高速计数器通道 12 使能状态	计数器使能了	计数器失能了	OFF	R

注：状态寄存器应从 FPGA 的寄存器中读出当前计数器的使能状态。

特殊寄存器	功能	范围	默认	R/W
SD1600	高速计数器通道 1 当前值	-2147483648~+2147483647	0	R
SD1601				
SD1630	高速计数器通道 2 当前值	-2147483648~+2147483647	0	R
SD1631				
SD1660	高速计数器通道 3 当前值	-2147483648~+2147483647	0	R
SD1661				
SD1690	高速计数器通道 4 当前值	-2147483648~+2147483647	0	R
SD1691				
SD1720	高速计数器通道 5 当前值	-2147483648~+2147483647	0	R
SD1721				
SD1750	高速计数器通道 6 当前值	-2147483648~+2147483647	0	R
SD1751				
SD1780	高速计数器通道 7 当前值	-2147483648~+2147483647	0	R
SD1781				
SD1810	高速计数器通道 8 当前值	-2147483648~+2147483647	0	R
SD1811				
SD1840	高速计数器通道 9 当前值	-2147483648~+2147483647	0	R
SD1841				
SD1870	高速计数器通道 10 当前值	-2147483648~+2147483647	0	R
SD1871				
SD1900	高速计数器通道 11 当前值	-2147483648~+2147483647	0	R
SD1901				

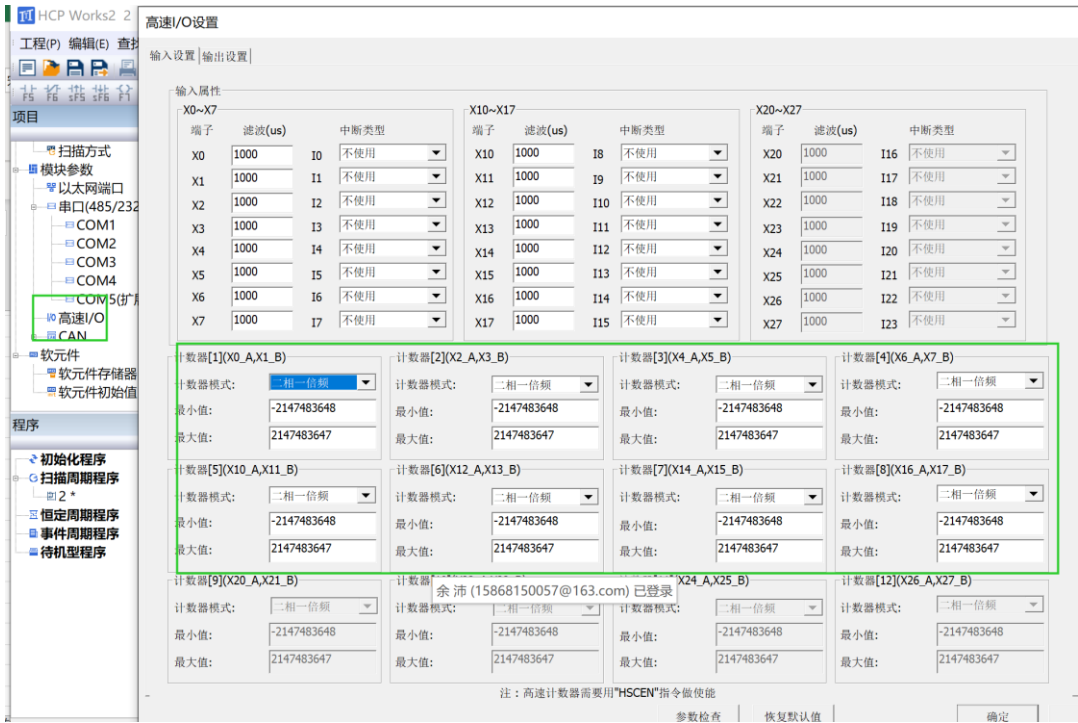
SD1930	高速计数器通道 12 当前值	-2147483648~+2147483647	0	R
SD1931				

注：上述特殊寄存器是掉电保持的。
注意：高速计数功能的具体用法请查看高速计数和脉冲测量章节

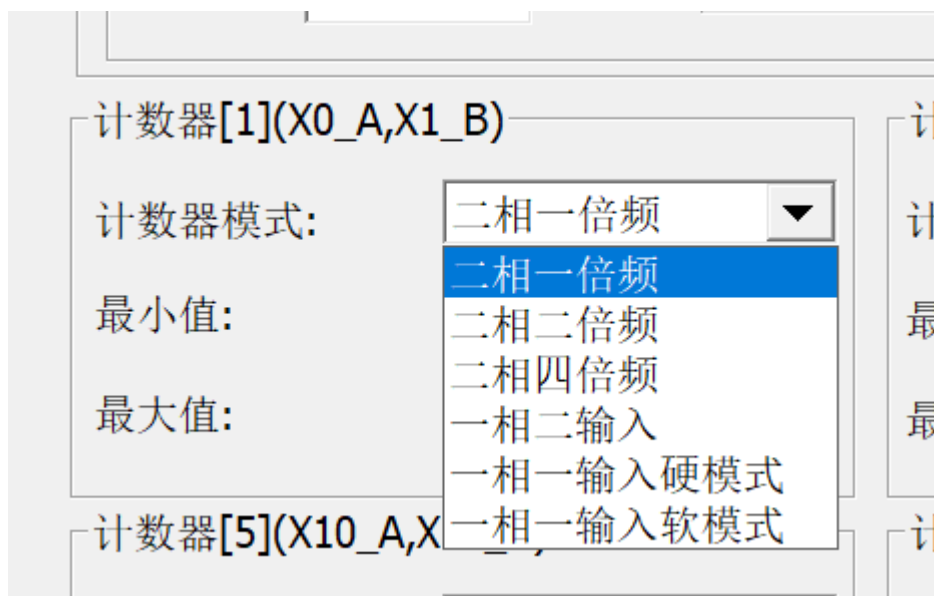
4.24.2 R 系列高速计数功能案例

本节内容以 R8P 使用二项一倍频计数为例介绍高速计数功能的用法

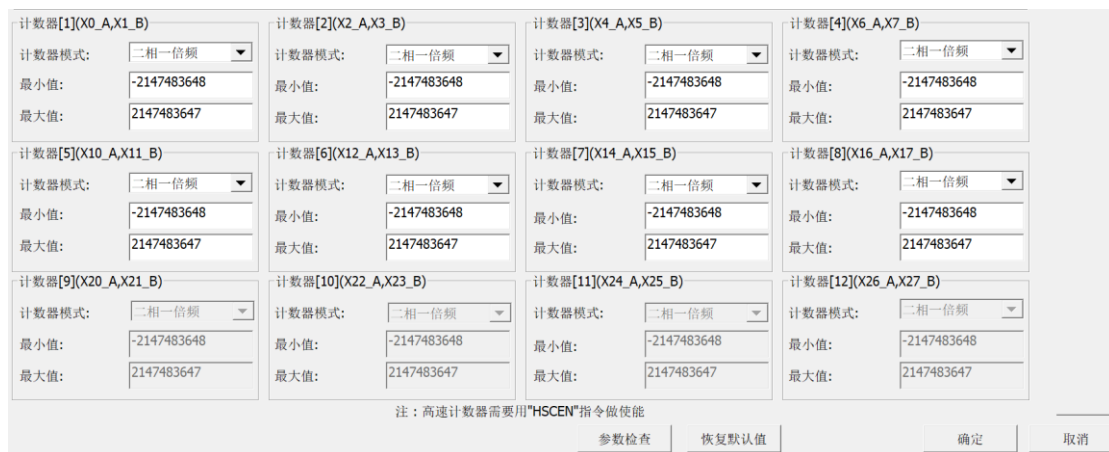
- 1) 用户连接 PLC 后确认 PLC 的计数器数量与高速计数端子位置（参考上节内容）
- 2) 确认完毕后用户点击 HCP WORKS2 项目栏中的高速 IO
- 3) 如下图所示 R8P 一共有 8 组计数端子



- 4) 用户可以选择各组计数器的计数模式，计数器的最大值与最小值（图中默认为极限范围）



5) 设置完成后，点击参数检查按钮



5) 如果有报错，根据相应提示内容修改，直至无报错。

6) 如果没有问题，点击确定后这些参数会在程序写入 PLC 后生效

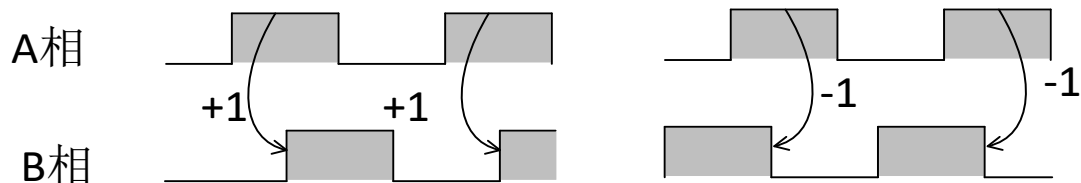
7) 示例中，使用计数器 1，计数模式为二相一倍频，计数逻辑如下

2 相 1 倍频：

动作	计数时刻
递增计数	A 相输入 ON 而 B 相输入 OFF→ON 时递增 1
递减计数	A 相输入 ON 而 B 相输入 ON→OFF 时递减 1

递增计数

递减计数



8) 在使用前用户需要将计数器 1 的端子连接 AB 相计数信号来源，计数端子高速 IO 中可以查阅。计数器 1 的 A 相为 X0，B 相为 X1，连接目标设备。

计数器[1](X0_A,X1_B)

计数器模式:

二相一倍频

最小值:

-2147483648

最大值:

2147483647

9) 如下图程序所示，用户想要使用高速计数相关功能必须先使能该计数器

10) HSCEN+计数器编号+计数器使能控制位

当 M1 闭合后，HSCEN 指令生效。

下图中 PLC 会根据 M2 的状态 ON/OFF,决定计数器 1 的生效和关闭。



10) DSETHSC+计数器编号+预置目标值

当 M3 闭合后，DSETHSC 指令生效。

PLC 会把 D0 中的实时值赋给计数器 1 的当前值也就是 SD1600 中

注：如无必要，建议此指令使用上升沿或者触发

名称	当前值	显示格式	数据类型	注释
SD1600	0	10进制数	有符号16位整数	

11) HSCS+对比目标值+计数器编号+需要置位的软元件

当 M4 闭合后，HSCS 指令生效。

PLC 会开始实时检测计数器的当前值也就是 SD1600 的实时值，当 SD1600 计数计到值与 D2 中的目标值一致的时候，会将 M5 从 OFF 置位 ON。

12) HSCR+对比目标值+计数器编号+需要复位的软元件

当 M6 闭合后，HSCR 指令生效。

PLC 会开始实时检测计数器的当前值也就是 SD1600 的实时值，当 SD1600 计数计到值与 D2 中的目标值一致的时候，会将 M7 复位

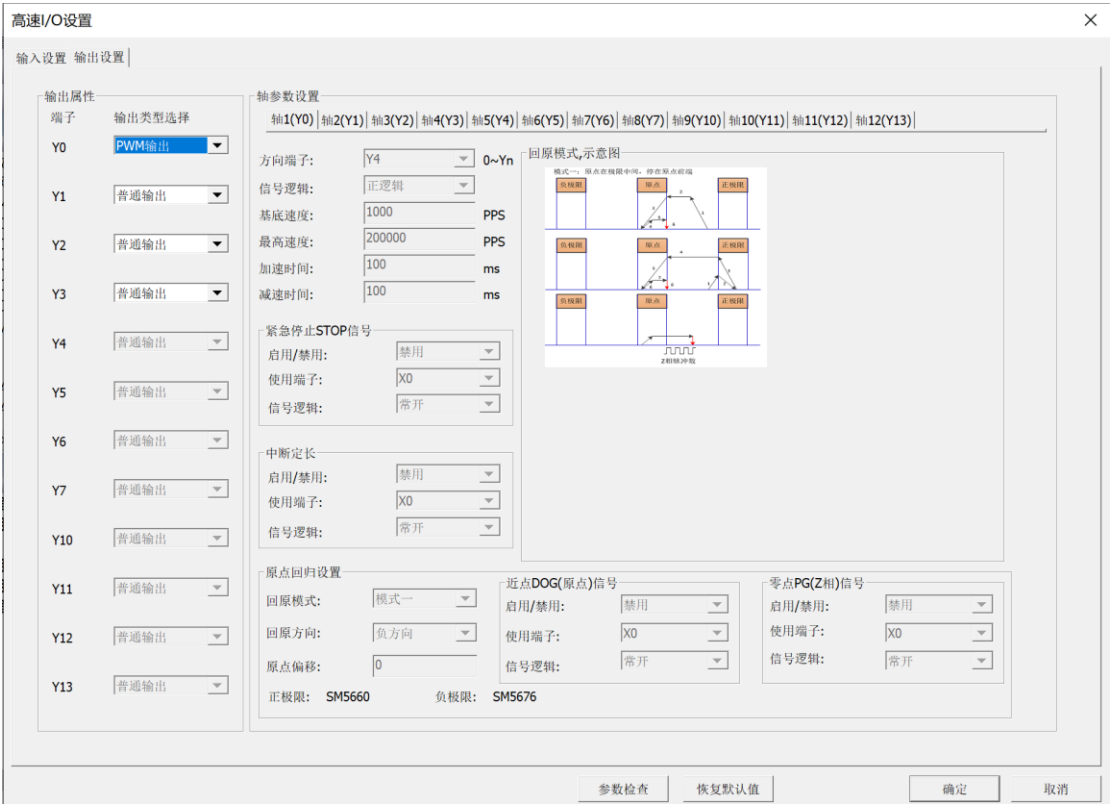
注：其他几种计数器使用时与此一致，一相一输入软模式 中的计数逻辑软标志位请查阅 6.1.2 章节，一相一输入硬模式，硬件输入信号由用户接入相应端子，详情参考上一节内容。

4.25 高速输出

4.25.1 R 系列 PWM 输出

概要
R 系列的 PWM 功能

用户在使用 PWM 输出前，需要先将目标高速端子设为 PWM 输出。



在完成设置后，用户可以使用 PWM 脉宽调试输出功能输出 PWM 波。（具体指令详见高速处理章节 PWM 指令）
注：不同机型的高速输出端子数量不同，使用前需确认机型的高速输出端子规格。

6. 特殊软元件介绍

6.1 特殊继电器

6.1.1 特殊辅助继电器（SM0-SM800）

特殊辅助继电器如下所示。

R：读取专用、R/W：读取/写入用

编号	名称	内容	R/W	关联软元件
SM0	RUN 监控 a 触点	OFF:PLC STOP ON:PLC RUN	R	
SM1	RUN 监控 b 触点	OFF:PLC RUN ON:PLC STOP	R	
SM2	初始脉冲 a 触点	OFF:PLC 运行第一个周期以外的时间 ON:PLC 运行的第一个周期	R	
SM3	初始脉冲 b 触点	OFF: PLC 运行的第一个周期 ON: PLC 运行第一个周期以外的时间	R	
SM4	错误发生	OFF: PLC 无出错 ON: PLC 错误发生	R	SD4
SM5 未实现	电池电压过低	OFF: ON:	R	SD5
SM6 未实现	电池电压过低锁存	OFF: ON:	R	SD6
SM7	检测出瞬间停止	OFF: 未检测到瞬间停止 ON:检测到瞬间停止	R	SD7
SM8	检测出停电中	OFF: 未检测到停电 ON:检测到停电中	R	SD8
SM9	DC24V 停电	-	R	SD9
SM10	不可使用	-	R	
SM11	10MS 时钟	OFF: 每隔 5MS ON: 每隔 5MS	R	
SM12	100MS 时钟	OFF: 每隔 50MS ON: 每隔 50MS	R	
SM13	1s 时钟	OFF: 每隔 500MS ON: 每隔 500MS	R	
SM14	1min 时钟	OFF: 每隔 30S ON: 每隔 30S	R	
SM15	停止计时以及预置，实时时钟用	ON 时，时钟停止 在 ON→OFF 的边沿写入时间，再次动作。	R/W	SD13-19
SM16	时间读出后的显示被停止，实时时钟用	OFF: 实时时钟正常显示 ON: 实时时钟正常计时，显示	R/W	SD13-19

		停止		
SM17	±30 秒的修正, 实时时钟用	在 OFF→ON 的边沿对秒进行修正。 (秒为 0~29 时, 秒变为 0。此外, 秒为 30~59 秒时, 进位到分钟, 秒变为 0)	R/W	SD13-19
SM18	检测出安装(一直为 ON), 实时时钟用	-		
SM19	实时时钟(RTC)错误, 实时时钟用	OFF: 其他情况下 ON: 校准时间时, 当特殊数据寄存器的数据超出设定范围时为 ON。	R	
SM20	零位	OFF: 其他情况下 ON: 加减法运算结果为 0 时接通	R	
SM21	借位	OFF: 其他情况下 ON: 减法运算结果超过最大的负值时接通	R	
SM22	进位	OFF: 其他情况下 ON: 加法运算结果发生进位时, 或者移位结果发生溢出时接通	R	
SM23	不可使用			
SM24	指定 BMOV 方向(FNC15)	OFF: 从操作数 2 到操作数 3 ON: 从操作数 3 到操作数 2	R/W	
SM25	未实现	不可使用	R/W	
SM26	未实现	不可使用		
SM27	未实现	不可使用		
SM28	未实现	不可使用		
SM29	指令执行结束		R	
SM30	未实现	不可使用		
SM31	未实现	不可使用		
SM32	未实现	不可使用		
SM33	未实现	不可使用		
SM34	未实现	不可使用		
SM35	强制 RUN 模式	OFF: 其他情况下 ON: 强制 RUN 模式开启	R/W	
SM36	强制 RUN 指令	OFF: 其他情况下 ON: 执行强制 RUN 模式	R/W	
SM37	强制 STOP 指令	OFF: 其他情况下 ON: 执行强制 STOP 模式	R/W	
SM38	参数的设定	未实现		
SM39	恒定扫描模式	OFF: 其他情况下 ON: 在恒定扫描模式下	R	
SM40	禁止转移	未实现		
SM41	转移开始	未实现		
SM42	启动脉冲	未实现		
SM43	原点回归结束	未实现		
SM44	原点条件	未实现		

SM45	禁止所有输出复位	未实现		
SM46	STL 状态动作	未实现		
SM47	STL 监控有效	未实现		
SM48	信号报警器动作	OFF: 其他情况下 ON:信号报警器发生报警	R	
SM49	信号报警器有效	OFF: 其他情况下 ON:信号报警器使能	R/W	
SM50	未实现	不可使用		
...	未实现	不可使用		
SM90	不可使用	不可使用		
SM91	COMRD、BINDA 指令 输出字符数切换信号	OFF: 默认情况下 ON:指令输出字符切换	R/W	
SM92	不可使用	-		
...	不可使用	-		
SM167	不可使用	-		
SM168	SMOV 处理 HEX 数据的 功能	OFF:其他情况下 ON:SMOV 指令处理 HEX 数据 时		
SM169	不可使用			
...	不可使用			
SM303	不可使用	-		
SM304	零位,	OFF:其他情况下 ON: 乘除运算结果为 0 时		
SM305	不可使用	-		
SM306	进位,	OFF:其他情况下 ON: 除法运算结果溢出时 0		
SM307	不可使用	-		
...	不可使用	-		
SM399	不可使用	-		
SM400	串行通信相关	详见 1.4 节		
...				
SM499				
SM500	不可使用	-		
...	不可使用	-		
SM800	不可使用	-		

6.1.2 高速计数特殊继电器

编号	名称	状态信息	读写性	备注
SM1600	高速计数器通道 1 使能 状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1601	高速计数器通道 2 使能 状态	OFF:计数器失能了 ON:计数器使能了	R	

SM1602	高速计数器通道 3 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1603	高速计数器通道 4 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1604	高速计数器通道 5 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1605	高速计数器通道 6 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1606	高速计数器通道 7 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1607	高速计数器通道 8 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1608	高速计数器通道 9 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1609	高速计数器通道 10 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1610	高速计数器通道 11 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1611	高速计数器通道 12 使能状态	OFF:计数器失能了 ON:计数器使能了	R	
SM1612	保留			
SM1613	保留			
SM1614	保留			
SM1615	保留			
SM1616	高速计数器通道 1	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1617	高速计数器通道 2	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1618	高速计数器通道 3	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1619	高速计数器通道 4	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1620	高速计数器通道 5	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1621	高速计数器通道 6	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1622	高速计数器通道 7	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1623	高速计数器通道 8	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		

SM1624	高速计数器通道 9	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1625	高速计数器通道 10	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1626	高速计数器通道 11	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1627	高速计数器通道 12	OFF:未发生	R/W	
	硬件预置触发标记 (外置 Z 信号)	ON:发生		
SM1628	保留			
SM1629	保留			
SM1630	保留			
SM1631	保留			
SM1632	软件预置触发标记 高速计数器通道 1	OFF:未发生 ON:发生	R/W	
SM1633	软件预置触发标记 高速计数器通道 2	OFF:未发生 ON:发生	R/W	
SM1634	软件预置触发标记 高速计数器通道 3	OFF:未发生 ON:发生	R/W	
SM1635	软件预置触发标记 高速计数器通道 4	OFF:未发生 ON:发生	R/W	
SM1636	软件预置触发标记 高速计数器通道 5	OFF:未发生 ON:发生	R/W	
SM1637	软件预置触发标记 高速计数器通道 6	OFF:未发生 ON:发生	R/W	
SM1638	软件预置触发标记 高速计数器通道 7	OFF:未发生 ON:发生	R/W	
SM1639	软件预置触发标记 高速计数器通道 8	OFF:未发生 ON:发生	R/W	
SM1640	软件预置触发标记 高速计数器通道 9	OFF:未发生 ON:发生	R/W	
SM1641	软件预置触发标记 高速计数器通道 10	OFF:未发生 ON:发生	R/W	
SM1642	软件预置触发标记 高速计数器通道 11	OFF:未发生 ON:发生	R/W	
SM1643	软件预置触发标记 高速计数器通道 12	OFF:未发生 ON:发生	R/W	
SM1644	保留			
SM1645	保留			
SM1646	保留			
SM1647	保留			
SM1648	计数器下溢标记 高速计数器通道 1	OFF:未发生 ON:发生	R/W	

SM1649	计数器下溢标记 高速 计数器通道 2	OFF:未发生 ON:发生	R/W	
SM1650	计数器下溢标记 高速 计数器通道 3	OFF:未发生 ON:发生	R/W	
SM1651	计数器下溢标记 高速 计数器通道 4	OFF:未发生 ON:发生	R/W	
SM1652	计数器下溢标记 高速 计数器通道 5	OFF:未发生 ON:发生	R/W	
SM1653	计数器下溢标记 高速 计数器通道 6	OFF:未发生 ON:发生	R/W	
SM1654	计数器下溢标记 高速 计数器通道 7	OFF:未发生 ON:发生	R/W	
SM1655	计数器下溢标记 高速 计数器通道 8	OFF:未发生 ON:发生	R/W	
SM1656	计数器下溢标记 高速 计数器通道 9	OFF:未发生 ON:发生	R/W	
SM1657	计数器下溢标记 高速 计数器通道 10	OFF:未发生 ON:发生	R/W	
SM1658	计数器下溢标记 高速 计数器通道 11	OFF:未发生 ON:发生	R/W	
SM1659	计数器下溢标记 高速 计数器通道 12	OFF:未发生 ON:发生	R/W	
SM1660	保留			
SM1661	保留			
SM1662	保留			
SM1663	保留			
SM1664	计数器上溢标记 高速 计数器通道 1	OFF:未发生 ON:发生	R/W	
SM1665	计数器上溢标记 高速 计数器通道 2	OFF:未发生 ON:发生	R/W	
SM1666	计数器上溢标记 高速 计数器通道 3	OFF:未发生 ON:发生	R/W	
SM1667	计数器上溢标记 高速 计数器通道 4	OFF:未发生 ON:发生	R/W	
SM1668	计数器上溢标记 高速 计数器通道 5	OFF:未发生 ON:发生	R/W	
SM1669	计数器上溢标记 高速 计数器通道 6	OFF:未发生 ON:发生	R/W	
SM1670	计数器上溢标记 高速 计数器通道 7	OFF:未发生 ON:发生	R/W	
SM1671	计数器上溢标记 高速 计数器通道 8	OFF:未发生 ON:发生	R/W	
SM1672	计数器上溢标记 高速 计数器通道 9	OFF:未发生 ON:发生	R/W	
SM1673	计数器上溢标记 高速 计数器通道 10	OFF:未发生 ON:发生	R/W	

SM1674	计数器上溢标记 计数器通道 11	高速	OFF:未发生 ON:发生	R/W	
SM1675	计数器上溢标记 计数器通道 12	高速	OFF:未发生 ON:发生	R/W	
SM1676	保留				
SM1677	保留				
SM1678	保留				
SM1679	保留				
SM1680	计数方向状态 计数器通道 1	高速	OFF:正计数 ON:负计数	R	
SM1681	计数方向状态 计数器通道 2	高速	OFF:正计数 ON:负计数	R	
SM1682	计数方向状态 计数器通道 3	高速	OFF:正计数 ON:负计数	R	
SM1683	计数方向状态 计数器通道 4	高速	OFF:正计数 ON:负计数	R	
SM1684	计数方向状态 计数器通道 5	高速	OFF:正计数 ON:负计数	R	
SM1685	计数方向状态 计数器通道 6	高速	OFF:正计数 ON:负计数	R	
SM1686	计数方向状态 计数器通道 7	高速	OFF:正计数 ON:负计数	R	
SM1687	计数方向状态 计数器通道 8	高速	OFF:正计数 ON:负计数	R	
SM1688	计数方向状态 计数器通道 9	高速	OFF:正计数 ON:负计数	R	
SM1689	计数方向状态 计数器通道 10	高速	OFF:正计数 ON:负计数	R	
SM1690	计数方向状态 计数器通道 11	高速	OFF:正计数 ON:负计数	R	
SM1691	计数方向状态 计数器通道 12	高速	OFF:正计数 ON:负计数	R	
SM1692	保留				
SM1693	保留				
SM1694	保留				
SM1695	保留				
SM1696	中断触发标记 计数器通道 1	高速	OFF:未发生 ON:发生	R/W	
SM1697	中断触发标记 计数器通道 2	高速	OFF:未发生 ON:发生	R/W	
SM1698	中断触发标记 计数器通道 3	高速	OFF:未发生 ON:发生	R/W	
SM1699	中断触发标记 计数器通道 4	高速	OFF:未发生 ON:发生	R/W	

SM1700	中断触发标记 计数器通道 5	高速	OFF:未发生 ON:发生	R/W	
SM1701	中断触发标记 计数器通道 6	高速	OFF:未发生 ON:发生	R/W	
SM1702	中断触发标记 计数器通道 7	高速	OFF:未发生 ON:发生	R/W	
SM1703	中断触发标记 计数器通道 8	高速	OFF:未发生 ON:发生	R/W	
SM1704	中断触发标记 计数器通道 9	高速	OFF:未发生 ON:发生	R/W	
SM1705	中断触发标记 计数器通道 10	高速	OFF:未发生 ON:发生	R/W	
SM1706	中断触发标记 计数器通道 11	高速	OFF:未发生 ON:发生	R/W	
SM1707	中断触发标记 计数器通道 12	高速	OFF:未发生 ON:发生	R/W	
SM1708	保留				
SM1709	保留				
SM1710	保留				
SM1711	保留				
SM1712	高速计数器通道 1		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1713	高速计数器通道 2		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1714	高速计数器通道 3		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1715	高速计数器通道 4		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1716	高速计数器通道 5		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1717	高速计数器通道 6		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1718	高速计数器通道 7		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1719	高速计数器通道 8		OFF:完成 ON:未完成	R/W	
	脉冲密度/转速测定完成				
SM1720	高速计数器通道 9		OFF:完成	R/W	

	脉冲密度/转速测定完成	ON:未完成		
SM1721	高速计数器通道 10	OFF:完成	R/W	
	脉冲密度/转速测定完成	ON:未完成		
SM1722	高速计数器通道 11	OFF:完成	R/W	
	脉冲密度/转速测定完成	ON:未完成		
SM1723	高速计数器通道 12	OFF:完成	R/W	
	脉冲密度/转速测定完成	ON:未完成		
SM1724	保留	OFF:完成 ON:未完成	R/W	
SM1725-1759	保留			
SM1760	高速计数器通道 1	OFF:完成	R/W	
	1 相 1 输入软模式计数方向	ON:未完成		
SM1761	高速计数器通道 2	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1762	高速计数器通道 3	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1763	高速计数器通道 4	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1764	高速计数器通道 5	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1765	高速计数器通道 6	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1766	高速计数器通道 7	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1767	高速计数器通道 8	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1768	高速计数器通道 9	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		
SM1769	高速计数器通道 10	OFF:负计数	R/W	
	1 相 1 输入软模式计数方向	ON:正计数		

SM1770	高速计数器通道 11	OFF:负计数 ON:正计数	R/W	
	1 相 1 输入软模式计数方向			
SM1771	高速计数器通道 12	OFF:负计数 ON:正计数	R/W	
	1 相 1 输入软模式计数方向			
SM1772-1807	保留	OFF:负计数 ON:正计数		

6.1.3 运动控制特殊继电器

编号				
SM5300	PWM 通道动作中	OFF: 停止中 ON: 动作中 00	R	通道 1
SM5301			R	通道 2
SM5302			R	通道 3
SM5303			R	通道 4
SM5304			R	通道 5
SM5305			R	通道 6
SM5306			R	通道 7
SM5307			R	通道 8
SM5308			R	通道 9
SM5309			R	通道 10
SM5310			R	通道 11
SM5311			R	通道 12
SM5316	PWM 输出结束标志位	OFF: 正常结束以外的状态 ON: 正常结束	R/W	通道 1
SM5317			R/W	通道 2
SM5318			R/W	通道 3
SM5319			R/W	通道 4
SM5320			R/W	通道 5
SM5321			R/W	通道 6
SM5322			R/W	通道 7
SM5323			R/W	通道 8
SM5324			R/W	通道 9
SM5325			R/W	通道 10
SM5326			R/W	通道 11
SM5327			R/W	通道 12
SM5500	定位轴定位指令驱动中	OFF: 停止中 ON: 驱动中	R	通道 1
SM5501			R	通道 2
SM5502			R	通道 3
SM5503			R	通道 4
SM5504			R	通道 5

SM5505			R	通道 6
SM5506			R	通道 7
SM5507			R	通道 8
SM5508			R	通道 9
SM5509			R	通道 10
SM5510			R	通道 11
SM5511			R	通道 12
SM5516	定位轴脉冲输出中监视	OFF: 停止中 ON: 输出中	R	通道 1
SM5517			R	通道 2
SM5518			R	通道 3
SM5519			R	通道 4
SM5520			R	通道 5
SM5521			R	通道 6
SM5522			R	通道 7
SM5523			R	通道 8
SM5524			R	通道 9
SM5525			R	通道 10
SM5526			R	通道 11
SM5527			R	通道 12
SM5532	定位轴发生定位出错	OFF: 无出错 ON: 发生出错	R/W	通道 1
SM5533			R/W	通道 2
SM5534			R/W	通道 3
SM5535			R/W	通道 4
SM5536			R/W	通道 5
SM5537			R/W	通道 6
SM5538			R/W	通道 7
SM5539			R/W	通道 8
SM5540			R/W	通道 9
SM5541			R/W	通道 10
SM5542			R/W	通道 11
SM5543			R/W	通道 12
SM5628	定位轴脉冲停止指令	OFF: 不停止脉冲输出 ON: 脉冲输出即时停止	R/W	通道 1
SM5629			R/W	通道 2
SM5630			R/W	通道 3
SM5631			R/W	通道 4
SM5632			R/W	通道 5
SM5633			R/W	通道 6
SM5634			R/W	通道 7
SM5635			R/W	通道 8
SM5636			R/W	通道 9
SM5637			R/W	通道 10
SM5638			R/W	通道 11
SM5639			R/W	通道 12

SM5644	定位轴脉冲减速停止指令 (带剩余距离运行)	OFF: 不停止脉冲输出 ON: 脉冲输出减速停止	R/W	通道 1
SM5645			R/W	通道 2
SM5646			R/W	通道 3
SM5647			R/W	通道 4
SM5648			R/W	通道 5
SM5649			R/W	通道 6
SM5650			R/W	通道 7
SM5651			R/W	通道 8
SM5652			R/W	通道 9
SM5653			R/W	通道 10
SM5654			R/W	通道 11
SM5655			R/W	通道 12
SM5660	定位轴正转极限	OFF: 正转极限 OFF ON: 正转极限 ON	R/W	通道 1
SM5661			R/W	通道 2
SM5662			R/W	通道 3
SM5663			R/W	通道 4
SM5664			R/W	通道 5
SM5665			R/W	通道 6
SM5666			R/W	通道 7
SM5667			R/W	通道 8
SM5668			R/W	通道 9
SM5669			R/W	通道 10
SM5670			R/W	通道 11
SM5671			R/W	通道 12
SM5676	定位轴反转极限	OFF: 反转极限 OFF ON: 反转极限 ON	R/W	通道 1
SM5677			R/W	通道 2
SM5678			R/W	通道 3
SM5679			R/W	通道 4
SM5680			R/W	通道 5
SM5681			R/W	通道 6
SM5682			R/W	通道 7
SM5683			R/W	通道 8
SM5684			R/W	通道 9
SM5685			R/W	通道 10
SM5686			R/W	通道 11
SM5687			R/W	通道 12
SM5772	定位轴旋转方向设置	OFF: 正转时当前地址增加 ON: 反转时当前地址增加	R/W	通道 1
SM5773			R/W	通道 2
SM5774			R/W	通道 3
SM5775			R/W	通道 4
SM5776			R/W	通道 5
SM5777			R/W	通道 6
SM5778			R/W	通道 7

SM5779			R/W	通道 8
SM5780			R/W	通道 9
SM5781			R/W	通道 10
SM5782			R/W	通道 11
SM5783			R/W	通道 12
SM5804	定位轴原点回归方向指定	OFF: 反转方向原点回归开始 ON: 正转方向原点回归开始	R/W	通道 1
SM5805			R/W	通道 2
SM5806			R/W	通道 3
SM5807			R/W	通道 4
SM5808			R/W	通道 5
SM5809			R/W	通道 6
SM5810			R/W	通道 7
SM5811			R/W	通道 8
SM5812			R/W	通道 9
SM5813			R/W	通道 10
SM5814			R/W	通道 11
SM5815			R/W	通道 12
SM5868	零点信号计数开始时刻	OFF: 近点狗后端 ON: 近点狗前端	R/W	通道 1
SM5869			R/W	通道 2
SM5870			R/W	通道 3
SM5871			R/W	通道 4
SM5872			R/W	通道 5
SM5873			R/W	通道 6
SM5874			R/W	通道 7
SM5875			R/W	通道 8
SM5876			R/W	通道 9
SM5877			R/W	通道 10
SM5878			R/W	通道 11
SM5879			R/W	通道 12

6.1.4 串行通信特殊继电器

编号	名称	内容	R/W	备注
SM400	自由格式通信发送中标志	指令开始执行时, 置 ON 发送完成时, 置 OFF	R	COM1
SM420			R	COM2
SM440			R	COM3
SM460			R	COM4
SM480			R	COM5
SM401	自由通信接收完成标志	接收完成 或超过重试次数时, 置 ON; 需要用户程序置 OFF	R/W	COM1
SM421			R/W	COM2
SM441			R/W	COM3

SM461			R/W	COM4
SM481			R/W	COM5
SM402	自由格式通信 缓冲位数	8 位 OFF 16 位 ON	R	COM1
SM422			R	COM2
SM442			R	COM3
SM462			R	COM4
SM482			R	COM5
SM410	MODBUS 通信出错	发生 MODBUS 通信出错时为 ON。为出错时为 OFF。	R	COM1
SM430			R	COM2
SM450			R	COM3
SM470			R	COM4
SM490			R	COM5
SM411	MODBUS_RTU 通信中	从开始执行命令直到命令执行结束标志 ON 为止，MODBUS 串行通信中为 ON	R	COM1
SM431			R	COM2
SM451			R	COM3
SM471			R	COM4
SM491			R	COM5
SM412	串行通信发生重试	从站在超时设置时间内无响应时，在主站发生发送重试的动作后为 ON。需要用户程序置 OFF，或者释放指令清除	R/W	COM1
SM432			R/W	COM2
SM452			R/W	COM3
SM472			R/W	COM4
SM492			R/W	COM5
SM413	串行通信发生超时	发生响应超时时为 ON。需要用户程序置 OFF，或者释放指令清除	R/W	COM1
SM433			R/W	COM2
SM453			R/W	COM3
SM473			R/W	COM4
SM493			R/W	COM5

注：自由格式通信专指自由格式通信时有效，modbus通信专指modbus通信有效，串行通信指所有的通信协议都有效。

6.2 特殊数据寄存器

6.2.1 特殊辅助寄存器(SD0~SD800)

编号	名称	内容	R/W	备注
SD0	看门狗定时器	初始值如右侧所示(1ms 单位) (电源 ON 时从系统 ROM 传送过来)	R	本 次 不 对 应
		通过程序改写的值，在执行了 END 后生效		
SD1 未实现	PLC 类型以及系统版本	记载了版本最后一个号	R	所 有 版 本 号

				SD21~23
SD2	内存容量	未实现	R	
SD3	内存种类	未实现	R	
SD4	错误 M 编号	PLC 发生出错时储存的错误编号	R	
SD5	电池电压	未实现	R	
SD6	检测出电池电压低的等级	未实现	R	
SD7	检测出瞬间停止	保存 M8007 的动作次数。 电源断开时清除。	R	
SD8	检测出停电的时间	储存了停电的时间，单位 10ms	R	
SD9 未实现	DC24V 掉电单元号		R	
SD10	扫描当前值	储存了扫描当前值，单位 0.1ms	R	
SD11	MIN 扫描时间	扫描时间的最小值 (0.1ms 单位)	R	
SD12	MAX 扫描时间	扫描时间的最大值 (0.1ms 单位)	R	
SD13	秒	0 ~ 59 秒 (实时时钟用)	R	
SD14	分	0 ~ 59 分 (实时时钟用)	R	
SD15	时	0 ~ 23 小时 (实时时钟用)	R	
SD16	日	1 ~ 31 日 (实时时钟用)	R	
SD17	月	1 ~ 12 月 (实时时钟用)	R	
SD18	年	西历 2 位数(0 ~ 99) (实时时钟用)	R	
SD19	星期	0(日) ~ 6(六) (实时时钟用)	R	
SD20	输入滤波器的调节	未实现		
SD21	系统版本号 1	第一个版本号 10 进制	R	
SD22	系统版本号 2	第二个版本号 10 进制	R	
SD23	系统版本号 3	第三个版本号 10 进制	R	
SD24	FPGA 版本低 16 位	第一个 FPGA 版本号 16 进制	R	
SD25	FPGA 版本高 16 位	第二个 FPGA 版本号 16 进制	R	
SD26	不可使用			
SD27	不可使用			
SD28	不可使用			
SD29	不可使用			
SD30	不可使用			
SD31	不可使用			
SD32	不可使用			
SD35	不可使用			
SD36	不可使用			

SD37	不可使用			
SD38	不可使用			
SD39	恒定扫描时间	目前只能查看恒定扫描时间	R	
SD40	步进梯形图相关			
SD41				
SD42				
SD43				
SD44				
SD45				
SD46				
SD47				
SD48	不可使用		R	
SD49	ON 状态最小编号	储存 ON 状态的 F 软元件最小编号	R/W	
SD50	不可使用			
SD51				
SD52				
SD53				
SD54				
SD55				
SD56				
SD57				
SD58				
SD59				
SD60	I/O 构成错误			
SD61	不可使用			
SD62	不可使用			
SD63	不可使用			
SD64	不可使用			
SD65	不可使用			
SD66	不可使用			
SD67	不可使用			
SD68	不可使用			
SD69	M8065~7 的错误步编号	储存了所有的错误号		
SD70	判断并联链接错误的时间 500ms	未实现		
SD71	不可使用			
SD72	不可使用			
SD73	不可使用			
SD74	不可使用			
SD75				
SD76				
SD77				
SD78				
SD79				
SD80				
SD81				

SD82				
SD83				
SD84				
SD85				
SD86				
SD87				
SD88				
SD89				
SD90				
SD91			R/W	
SD92				
SD93				
SD94				
SD95				
SD96				
SD97				
SD98				
SD99	高速计数环形计数器相关	未实现		
SD100	不可使用	-		
SD101	PLC 类型以及系统版本	未实现		
SD102	内存容量	未实现		
SD103	不可使用	-		
SD104	功能扩展内固有的机型代码	未实现		
SD105	功能扩展内存的版本	未实现		
SD106	不可使用	-		
SD107	软元件注释登录数	未实现		
SD108 未实现	特殊模块的连接台数			
SD109 未实现	发生输出刷新错误的 Y 编号			
SD110	不可使用	-		
SD111	不可使用	-		
SD112	功能扩展板 A1/A2 专用	-		
SD113		-		
SD114		-		
SD115	不可使用	-		
...	不可使用	-		
SD309	不可使用	-		
SD310	RND 指令相关	低位		
SD311		高位		
SD312	不可使用	-		
...	不可使用	-		
SD399	不可使用	-		
SD400				
...	串行通信相关	查看 2.4 节		
SD499				

SD500	ALARM 发生 CODE 1	PLC 系统错误码存储器 1		
SD501	ALARM 发生 CODE 2	PLC 系统错误码存储器 2		
SD502	ALARM 发生 CODE 3	PLC 系统错误码存储器 3		
SD503	ALARM 发生 CODE 4	PLC 系统错误码存储器 4		
SD504	ALARM 发生 CODE 5	PLC 系统错误码存储器 5		
SD505	ALARM 发生 CODE 6	PLC 系统错误码存储器 6		
SD506	ALARM 发生 CODE 7	PLC 系统错误码存储器 7		
SD507	ALARM 发生 CODE 8	PLC 系统错误码存储器 8		
SD508	ALARM 发生 CODE 9	PLC 系统错误码存储器 9		
SD509	ALARM 发生 CODE 10	PLC 系统错误码存储器 10		
SD510	ALARM 发生 CODE 11	PLC 系统错误码存储器 11		
SD511	ALARM 发生 CODE 12	PLC 系统错误码存储器 12		
SD512	ALARM 发生 CODE 13	PLC 系统错误码存储器 13		
SD513	ALARM 发生 CODE 14	PLC 系统错误码存储器 14		
SD514	ALARM 发生 CODE 15	PLC 系统错误码存储器 15		
SD515	ALARM 发生 CODE 16	PLC 系统错误码存储器 16		
SD516	不可使用	-		
SD517	不可使用	-		
SD518	不可使用	-		
SD519	不可使用	-		
SD520	最先检测出的报警器 No.将被存储。	用于储存最先监测到的报警器 编号		
SD521	检测出报警器的个数将 被存储。	用于储存监测到的报警器数量		
SD522	报警器检测编号将被存 储。1	报警器检测编号存储器 1		
SD523	报警器检测编号将被存 储。2	报警器检测编号存储器 2		
SD524	报警器检测编号将被存 储。3	报警器检测编号存储器 3		
SD525	报警器检测编号将被存 储。4	报警器检测编号存储器 4		
SD526	报警器检测编号将被存 储。5	报警器检测编号存储器 5		
SD527	报警器检测编号将被存 储。6	报警器检测编号存储器 6		
SD528	报警器检测编号将被存 储。7	报警器检测编号存储器 7		
SD529	报警器检测编号将被存 储。8	报警器检测编号存储器 8		
SD530	报警器检测编号将被存 储。9	报警器检测编号存储器 9		
SD531	报警器检测编号将被存 储。10	报警器检测编号存储器 10		
SD532	报警器检测编号将被存 储。11	报警器检测编号存储器 11		
SD533	报警器检测编号将被存 储。12	报警器检测编号存储器 12		

SD534	报警器检测编号将被存储。13	报警器检测编号寄存器 13		
SD535	报警器检测编号将被存储。14	报警器检测编号寄存器 14		
SD536	报警器检测编号将被存储。15	报警器检测编号寄存器 15		
SD537	报警器检测编号将被存储。16	报警器检测编号寄存器 16		
SD538	不可使用	-		
SD600	不可使用	-		
...	不可使用	-		
SD800	不可使用	-		

6.2.2 高速计数特殊寄存器

编号	名称	范围	默认	R/W
SD1600	高速计数器通道 1 当前值	-2147483648 ~ +2147483647	0	R
SD1601				
SD1630	高速计数器通道 2 当前值	-2147483648 ~ +2147483647	0	R
SD1631				
SD1660	高速计数器通道 3 当前值	-2147483648 ~ +2147483647	0	R
SD1661				
SD1690	高速计数器通道 4 当前值	-2147483648 ~ +2147483647	0	R
SD1691				
SD1720	高速计数器通道 5 当前值	-2147483648 ~ +2147483647	0	R
SD1721				
SD1750	高速计数器通道 6 当前值	-2147483648 ~ +2147483647	0	R
SD1751				
SD1780	高速计数器通道 7 当前值	-2147483648 ~ +2147483647	0	R
SD1781				
SD1810	高速计数器通道 8 当前值	-2147483648 ~ +2147483647	0	R
SD1811				
SD1840	高速计数器通道 9 当前值	-2147483648 ~ +2147483647	0	R
SD1841				
SD1870	高速计数器通道 10 当前值	-2147483648 ~ +2147483647	0	R
SD1871				
SD1900	高速计数器通道 11 当前值	-2147483648 ~ +2147483647	0	R
SD1901				
SD1930	高速计数器通道 12 当前值	-2147483648 ~ +2147483647	0	R
SD1931				

6.2.3 运动控制特殊寄存器

编号	名称	内容	R/W	备注
SD5500	定位轴当前地址低位	显示定位轴当前地址	R	通道 1
SD5540			R	通道 2
SD5580			R	通道 3
SD5620			R	通道 4
SD5660			R	通道 5
SD5700			R	通道 6
SD5740			R	通道 7
SD5780			R	通道 8
SD5820			R	通道 9
SD5860			R	通道 10
SD5900			R	通道 11
SD5940			R	通道 12
SD5501	定位轴当前地址高位	显示定位轴当前地址（最大 32 位）	R	通道 1
SD5541			R	通道 2
SD5581			R	通道 3
SD5621			R	通道 4
SD5661			R	通道 5
SD5701			R	通道 6
SD5741			R	通道 7
SD5781			R	通道 8
SD5821			R	通道 9
SD5861			R	通道 10
SD5901			R	通道 11
SD5941			R	通道 12
SD5504	定位轴当前速度低位	显示定位轴的当前速度	R	通道 1
SD5544			R	通道 2
SD5584			R	通道 3
SD5624			R	通道 4
SD5664			R	通道 5
SD5704			R	通道 6
SD5744			R	通道 7
SD5784			R	通道 8
SD5824			R	通道 9
SD5864			R	通道 10
SD5904			R	通道 11
SD5944			R	通道 12
SD5505	定位轴当前速度高位	显示定位轴的当前速度（最大 32 位）	R	通道 1
SD5545			R	通道 2
SD5585			R	通道 3
SD5625			R	通道 4

SD5665			R	通道 5
SD5705			R	通道 6
SD5745			R	通道 7
SD5785			R	通道 8
SD5825			R	通道 9
SD5865			R	通道 10
SD5905			R	通道 11
SD5945			R	通道 12
SD5510	定位轴出错代码	显示定位轴的出错代码	R/W	通道 1
SD5550			R/W	通道 2
SD5590			R/W	通道 3
SD5630			R/W	通道 4
SD5670			R/W	通道 5
SD5710			R/W	通道 6
SD5750			R/W	通道 7
SD5790			R/W	通道 8
SD5830			R/W	通道 9
SD5870			R/W	通道 10
SD5910			R/W	通道 11
SD5950			R/W	通道 12
SD5512	定位轴方向引脚	用于显示和修改定位轴当前的 方向管脚（锁存）	R/W	通道 1
SD5552			R/W	通道 2
SD5592			R/W	通道 3
SD5632			R/W	通道 4
SD5672			R/W	通道 5
SD5712			R/W	通道 6
SD5752			R/W	通道 7
SD5792			R/W	通道 8
SD5832			R/W	通道 9
SD5872			R/W	通道 10
SD5912			R/W	通道 11
SD5952			R/W	通道 12
SD5516	定位轴最高速度低位	显示定位轴最高速度（最大 32 位）（锁存）	R/W	通道 1
SD5556			R/W	通道 2
SD5596			R/W	通道 3
SD5636			R/W	通道 4
SD5676			R/W	通道 5
SD5716			R/W	通道 6
SD5756			R/W	通道 7
SD5796			R/W	通道 8
SD5836			R/W	通道 9
SD5876			R/W	通道 10
SD5916			R/W	通道 11

SD5956			R/W	通道 12
SD5517	定位轴最高速度高位	显示并可以修改定位轴最高速度（锁存）	R/W	通道 1
SD5557			R/W	通道 2
SD5597			R/W	通道 3
SD5637			R/W	通道 4
SD5677			R/W	通道 5
SD5717			R/W	通道 6
SD5757			R/W	通道 7
SD5797			R/W	通道 8
SD5837			R/W	通道 9
SD5877			R/W	通道 10
SD5917			R/W	通道 11
SD5957			R/W	通道 12
SD5518	定位轴基底速度低位	显示并可以修改定位轴的基底速度（锁存）	R/W	通道 1
SD5558			R/W	通道 2
SD5598			R/W	通道 3
SD5638			R/W	通道 4
SD5678			R/W	通道 5
SD5718			R/W	通道 6
SD5758			R/W	通道 7
SD5798			R/W	通道 8
SD5838			R/W	通道 9
SD5978			R/W	通道 10
SD5918			R/W	通道 11
SD5958			R/W	通道 12
SD5519	定位轴基底速度高位	显示并可以修改定位轴的基底速度（锁存）	R/W	通道 1
SD5559			R/W	通道 2
SD5599			R/W	通道 3
SD5639			R/W	通道 4
SD5679			R/W	通道 5
SD5719			R/W	通道 6
SD5759			R/W	通道 7
SD5799			R/W	通道 8
SD5839			R/W	通道 9
SD5979			R/W	通道 10
SD5919			R/W	通道 11
SD5959			R/W	通道 12
SD5520	定位轴加速时间	显示并可以修改定位轴的加速时间（锁存）	R/W	通道 1
SD5560			R/W	通道 2
SD5600			R/W	通道 3
SD5640			R/W	通道 4
SD5680			R/W	通道 5
SD5720			R/W	通道 6

SD5760			R/W	通道 7
SD5800			R/W	通道 8
SD5840			R/W	通道 9
SD5880			R/W	通道 10
SD5920			R/W	通道 11
SD5960			R/W	通道 12
SD5521	定位轴减速时间	显示并可以修改定位轴的减速时间（锁存）	R/W	通道 1
SD5561			R/W	通道 2
SD5601			R/W	通道 3
SD5641			R/W	通道 4
SD5681			R/W	通道 5
SD5721			R/W	通道 6
SD5761			R/W	通道 7
SD5801			R/W	通道 8
SD5841			R/W	通道 9
SD5881			R/W	通道 10
SD5921			R/W	通道 11
SD5961			R/W	通道 12
SD5525	定位轴原点回归模式	显示并可以修改定位轴的原点回归模式（锁存）	R/W	通道 1
SD5565			R/W	通道 2
SD5605			R/W	通道 3
SD5645			R/W	通道 4
SD5685			R/W	通道 5
SD5725			R/W	通道 6
SD5765			R/W	通道 7
SD5805			R/W	通道 8
SD5845			R/W	通道 9
SD5885			R/W	通道 10
SD5925			R/W	通道 11
SD5965			R/W	通道 12
SD5526	定位轴原点回归速度低位	显示并可以修改定位轴的原点回归速度（锁存）	R/W	通道 1
SD5566			R/W	通道 2
SD5606			R/W	通道 3
SD5646			R/W	通道 4
SD5686			R/W	通道 5
SD5726			R/W	通道 6
SD5766			R/W	通道 7
SD5806			R/W	通道 8
SD5846			R/W	通道 9
SD5886			R/W	通道 10
SD5926			R/W	通道 11
SD5966			R/W	通道 12
SD5527		显示并可以修改定位轴的原点	R/W	通道 1

SD5567	定位轴原点回归速度高位	回归速度（锁存）	R/W	通道 2
SD5607			R/W	通道 3
SD5647			R/W	通道 4
SD5687			R/W	通道 5
SD5727			R/W	通道 6
SD5767			R/W	通道 7
SD5807			R/W	通道 8
SD5847			R/W	通道 9
SD5887			R/W	通道 10
SD5927			R/W	通道 11
SD5967			R/W	通道 12
SD5528	定位轴原点回归爬行速度低位	显示并可以修改定位轴的原点回归爬行速度（锁存）	R/W	通道 1
SD5568			R/W	通道 2
SD5608			R/W	通道 3
SD5648			R/W	通道 4
SD5688			R/W	通道 5
SD5728			R/W	通道 6
SD5768			R/W	通道 7
SD5808			R/W	通道 8
SD5848			R/W	通道 9
SD5888			R/W	通道 10
SD5928			R/W	通道 11
SD5968			R/W	通道 12
SD5529	定位轴原点回归爬行速度高位	显示并可以修改定位轴的原点回归爬行速度（锁存）	R/W	通道 1
SD5569			R/W	通道 2
SD5609			R/W	通道 3
SD5649			R/W	通道 4
SD5689			R/W	通道 5
SD5729			R/W	通道 6
SD5769			R/W	通道 7
SD5809			R/W	通道 8
SD5849			R/W	通道 9
SD5889			R/W	通道 10
SD5929			R/W	通道 11
SD5969			R/W	通道 12
SD5530	定位轴原点偏移地址低位	显示并可以修改定位轴的原点偏移地址（锁存）	R/W	通道 1
SD5570			R/W	通道 2
SD5610			R/W	通道 3
SD5650			R/W	通道 4
SD5690			R/W	通道 5
SD5730			R/W	通道 6
SD5770			R/W	通道 7
SD5810			R/W	通道 8

SD5850			R/W	通道 9
SD5890			R/W	通道 10
SD5930			R/W	通道 11
SD5970			R/W	通道 12
SD5531	定位轴原点偏移地址高位	显示并可以修改定位轴的原点偏移地址（锁存）	R/W	通道 1
SD5571			R/W	通道 2
SD5611			R/W	通道 3
SD5651			R/W	通道 4
SD5691			R/W	通道 5
SD5731			R/W	通道 6
SD5771			R/W	通道 7
SD5811			R/W	通道 8
SD5851			R/W	通道 9
SD5891			R/W	通道 10
SD5931			R/W	通道 11
SD5971			R/W	通道 12
SD5532	定位轴原点回归零点信号数	显示并可以修改定位轴的原点回归零点信号数	R/W	通道 1
SD5572			R/W	通道 2
SD5612			R/W	通道 3
SD5652			R/W	通道 4
SD5692			R/W	通道 5
SD5732			R/W	通道 6
SD5772			R/W	通道 7
SD5812			R/W	通道 8
SD5852			R/W	通道 9
SD5892			R/W	通道 10
SD5932			R/W	通道 11
SD5972			R/W	通道 12
SD5533	定位轴原点回归停留时间	显示并可以修改定位轴的原点回归停留时间	R/W	通道 1
SD5573			R/W	通道 2
SD5613			R/W	通道 3
SD5653			R/W	通道 4
SD5693			R/W	通道 5
SD5733			R/W	通道 6
SD5773			R/W	通道 7
SD5813			R/W	通道 8
SD5853			R/W	通道 9
SD5893			R/W	通道 10
SD5933			R/W	通道 11
SD5973			R/W	通道 12

6.2.4 串行通信特殊寄存器

编号	名称	内容	R/W	备注
SD400	自由格式通信发送结果	0: 正确 2: 发送长度设置错误 如设置为 0 或大于 256 时 10: 自由格式发送缓冲区溢出 发生溢出后需要使用释放指令 (RLS) 清除错误。缓冲区大小 256	R/W	COM1
SD420			R/W	COM2
SD440			R/W	COM3
SD460			R/W	COM4
SD480			R/W	COM5
SD401	自由格式通信接收结果	0: 正确 10: 接收长度设置错误 11: 接收数据短 12: 接收数据长 13: 接收错误 14: 接收超时 15: 无起始符 16: 无终止符	R/W	COM1
SD421			R/W	COM2
SD441			R/W	COM3
SD461			R/W	COM4
SD481			R/W	COM5
SD402	自由格式通信接收数据个数	按字节计, 不包含起始符、终止符	R	COM1
SD422			R	COM2
SD442			R	COM3
SD462			R	COM4
SD482			R	COM5
SD403	串行通信设置	设置在 PLC 中的通信属性会被记录。	R	COM1
SD423			R	COM2
SD443			R	COM3
SD463			R	COM4
SD483			R	COM5
SD404	串行通信动作模糊显示	0: 通道未启用 1: 下载/HMI 监控协议 2: 预留 3: 预留 5: 自由协议通信 9: MODBUS RTU 主站 10: MODBUS RTU 从站	R	COM1
SD424			R	COM2
SD444			R	COM3
SD464			R	COM4
SD484			R	COM5
SD405	自由格式通信起始符、终止符	起始符高 8 位, 终止符低 8 位	R	COM1
SD425			R	COM2
SD445			R	COM3
SD465			R	COM4
SD485			R	COM5
SD406	自由格式通信帧超时	设置在 PLC 中的帧超时会被记录。 单位 mS	R	COM1
SD426			R	COM2
SD446			R	COM3
SD466			R	COM4
SD486			R	COM5

SD407	串 行 通 信 当前的重试次数	因从站响应超时而进行通信重试时，当前的重试次数会被记录	R	COM1
SD427			R	COM2
SD447			R	COM3
SD467			R	COM4
SD487			R	COM5
SD408	串 行 通 信 从站响应超时（通信超时）	从站响应超时的设置值会被记录。 单位 mS	R	COM1
SD428			R	COM2
SD448			R	COM3
SD468			R	COM4
SD488			R	COM5
SD409	串 行 通 信 超时时的重试次数	超时时的重试次数的设置值会被记录	R	COM1
SD429			R	COM2
SD449			R	COM3
SD469			R	COM4
SD489			R	COM5
SD410	MODBUS 通信出错代码	在 MODBUS 串行通信中发生的最新出错代码会被记录	R/W	COM1
SD430			R/W	COM2
SD450			R/W	COM3
SD470			R/W	COM4
SD490			R/W	COM5
SD411	MODBUS 通信出错的详细内容	最新出错的详细内容会被记录	R/W	COM1
SD431			R/W	COM2
SD451			R/W	COM3
SD471			R/W	COM4
SD491			R/W	COM5
SD412	本站站号	设置的从站站号会被记录	R/W	COM1
SD432			R/W	COM2
SD452			R/W	COM3
SD472			R/W	COM4
SD492			R/W	COM5

错误码	说明
10	指令设置接收的数据长度超过 256 个或者设置为 0 时（不管是 8 位还是 16 位模式）
11	帧间隔超过帧超时时间没有收到指定的个数，报接收数据短
12	帧间隔超过帧超时时间接收超过指定的个数，报接收数据长
13	接收数据的校验位，数据位等设置不符，报接收错误
14	通信超时时间内没有收到数据，报接收超时
15	通信设置里面设置了起始符而没有收到。报无起始符
16	通信设置里面设置了终止符而没有收到。报无终止符

串行通信设置

用于读取设置串行通信的格式（只读）

位	名称	内容	
		0 (bit = OFF)	1 (bit = ON)
b0	—	—	—

b1、b2	奇偶位	(b2 、 b1) = (0 、 0) : 无 (b2 、 b1) = (0 、 1) : 奇 数 (b2、b1) = (1、1): 偶数	
b3	停止位	1bit	2bit
b4 ~ b7	波特率 (bpS)	(b7 、 b6 、 b5 、 b4) = (1 、 0 、 0 、 0) : 9600 (b7 、 b6 、 b5 、 b4) = (1 、 0 、 0 、 1) : 19200 (b7 、 b6 、 b5 、 b4) = (1 、 0 、 1 、 0) : 38400 (b7 、 b6 、 b5 、 b4) = (1 、 0 、 1 、 1) : 57600 (b7、b6、b5、b4) = (1、1、0、1): 115200 (b7、b6、b5、b4) = (1、1、1、1): 256000	
b8 ~ b15	—	—	—

6.2.5 IP 地址特殊软元件

编号	名称	内容	可读写性	默认值	是否锁存
SD240	IP 地址	IP 地址的第 1 个字节	R/W	192	是
SD241		IP 地址的第 2 个字节	R/W	168	是
SD242		IP 地址的第 3 个字节	R/W	188	是
SD243		IP 地址的第 4 个字节	R/W	200	是
SD244	子网掩码	子网掩码的第 1 个字节	R/W	255	是
SD245		子网掩码的第 2 个字节	R/W	255	是
SD246		子网掩码的第 3 个字节	R/W	255	是
SD247		子网掩码的第 4 个字节	R/W	0	是
SD248	网关	网关的第 1 个字节	R/W	192	是
SD249		网关的第 2 个字节	R/W	168	是
SD250		网关的第 3 个字节	R/W	188	是
SD251		网关的第 4 个字节	R/W	1	是

注：1.IP 地址特殊软元件中的数据格式为十进制

2.软元件的内容修改后，只会在 PLC 重启后生效

3. R 系列 PLC IP 地址初始值为 192.168.188.200，子网掩码为 255.255.255.0，默认网关为 192.168.188.1

IP 地址特殊软元件修改方式汇总

序号	内容
1	在梯形图上使用指令对其值进行赋值，其值修改后将在下一次 PLC 重新上电时生效
2	支持在注册窗口中或监视窗口对其值进行实时修改，其值修改后将在下一次 PLC 重新上电时生效
3	支持在工程中修改以太网相端口设置中及基本设置页面修改（需要将程序写入 PLC 后重启生效
4	通过 HMI 设备使用“modbus”等通信方式进行相应特殊寄存器的值的修改，其值修改后将在下一次 PLC 重新上电时生效

7. PLC 出错代码

7.1 PLC 系统出错代码一览

错误码	内容	处理方法	详细信息	诊断实机	错误等级
1	操作错误				
2	指针错误				
3	程序错误				
4	地址错误				
5	参数错误				
6	通信错误				
9	计算错误				
10	类型不匹配				
11	循环次数错误				
12	软元件分配对齐错误				
20	内存错误				
21	FUNC 执行错误				
30	任务执行错误				
50	步数过大错误				
99	看门狗错误				
4000	当扫描程序超过一定时间的				

	错误. 举例: 通过 CJ 达到无线循环.				
4001	软元件初始值文件读取失败				
6004	软元件使用超过范围				
6007	BCD 错误				
6008	除数为 0				
6617	应该从母线开始的指令没有连接在母线上。 STL 中使用了指针 P。				
6618	主程序以外的程序(中断、子程序、恒定中断程序等)使用了只能在主程序中使用的指令 (STL、RETSTL、MC、MCR)。				
6619	FOR-NEXT 之间有不能使用的指令 STL、RETSTL、MC、MCR。				
6625	使用同样编号的步进继电器。				
6626	STL-RETSTL 之间有不能使用的指令 MC、MCR、CJ。				
6627	STL、RETSTL 关系不匹配。				
6144	用户报警器报错				
6630	STL 结构性报错				
6706	运算错误或参数的值超过可设置范围				

6710	命令中多个参数设置的范围重叠				
6900	锁存数据读取失败				
6901	锁存数据为旧数据				
7309	从站响应超时，经过通信参数超时中设置的时间后，从站仍无响应				
7310	主站未设定，Modbus 指令不可执行				
7311	从站 MODBUS 地址映射表、字符串转码错误				
8000	IO 构成错误				
8001	模块数 K0~K7 错误				
8002	获取 FPGA 信息错误. 举例: 获取版本, 获取中断信息, 获取模块数信息 等				
8010	超过嵌套最大数				
8011	通过返回指令对嵌套数进行减法运算，在该过程中减法运算结果变为负值时				
8012	只有 RET, 没有指针				
8013	没有指针				
8015	CJ 的指针不再同一个文件				
8016	CJ 的指针类型错误				
8017	CJ 指针不存在				
8018	步数过大				
8019	指针使用错误. 比如指针				

	不存在,子程序没有有用 sret 结束等等..				
8020	FOR NEXT 结构性错误.举例： 只有 NEXT ,只有 FOR 指令等等				
8025	MC-MCR 指令使用有误.				
8030	程序结构性错误				
8031	不可知性的中断指针				

其他指令，后续扩充