

1. PLC 构成

第一章 PLC 构成	2
1 产品概要和对应的编程语言	2
2 构成可编程控制器的各种软元件	4
3 程序内存和参数的构造	7
4 注意事项(输入输出处理, 响应滞后, 双线圈)	10

第一章 PLC 构成

1 产品概要和对应的编程语言

产品概要

BSP01 AR/T Type : 30 点

- ◆ 内藏 Flash memory (8,000 步)、万年历、RS485 通讯。
- ◆ 锂电池供数据与万年历断电保存。
- ◆ 端子台可插拔。
- ◆ 使用扩充电源时, 数位输入输出可扩充到 256 点。
- ◆ 可扩充 8 路模拟量输入, 2 路模拟量输出。

BSP01 AR/T Type : 40/60 点

- ◆ 内藏 Flash memroy (16,000 步)、万年历、RS485 通讯。
- ◆ 锂电池供数据与万年历断电保存。
- ◆ 端子台可插拔。
- ◆ 使用扩充电源时, 数位输入输出可扩充到 256 点。
- ◆ 可扩充 60 路模拟量输入, 10 路模拟量输出。

BSP01 SR Type : 14/20/26/36 点

- ◆ 内藏 Flash memory (4000 步)、RS485 通讯。
- ◆ 数位输入输出可扩充到 80 点。
- ◆ 可扩充 8 路模拟量输入, 2 路模拟量输出。

编程方式

<<指令表 IL 编程>>

是以 LD, AND, OUT 等顺控指令输入的方式, 这种方式是编写顺控程序的基本输入形式, 但控制内容难于看懂。

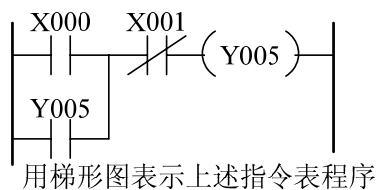
例:

步号	指令	软元件
0	LD	X000
1	OR	Y005
2	ANI	X001
3	OUT	Y005

<<梯形图 LD 编程>>

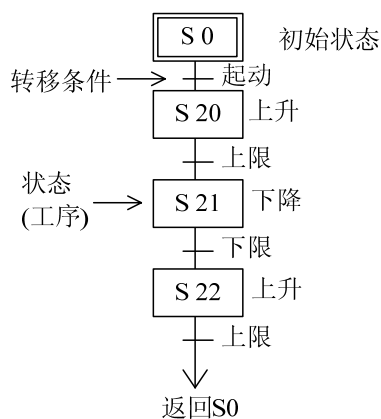
梯形图程序是采用顺控信号及软组件号, 在图形画面上作出顺控电路图的方式, 这种方法是用触点符号与线圈符号表示顺控回路, 因而容易理解程序的内容。同时还可利用回路显示的状态来监控可编程控制器的动作。

1. PLC 构成

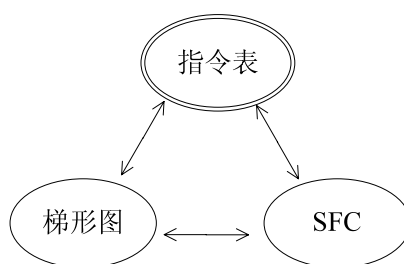


<<顺序功能图 SFC 编程>>

SFC 编程是根据机械动作的流程进行顺控设计的输入方法, 在具有个人计算机与其它的图形画面的外围设备中, 通过制作下图所示的画面可决定顺控的流程。

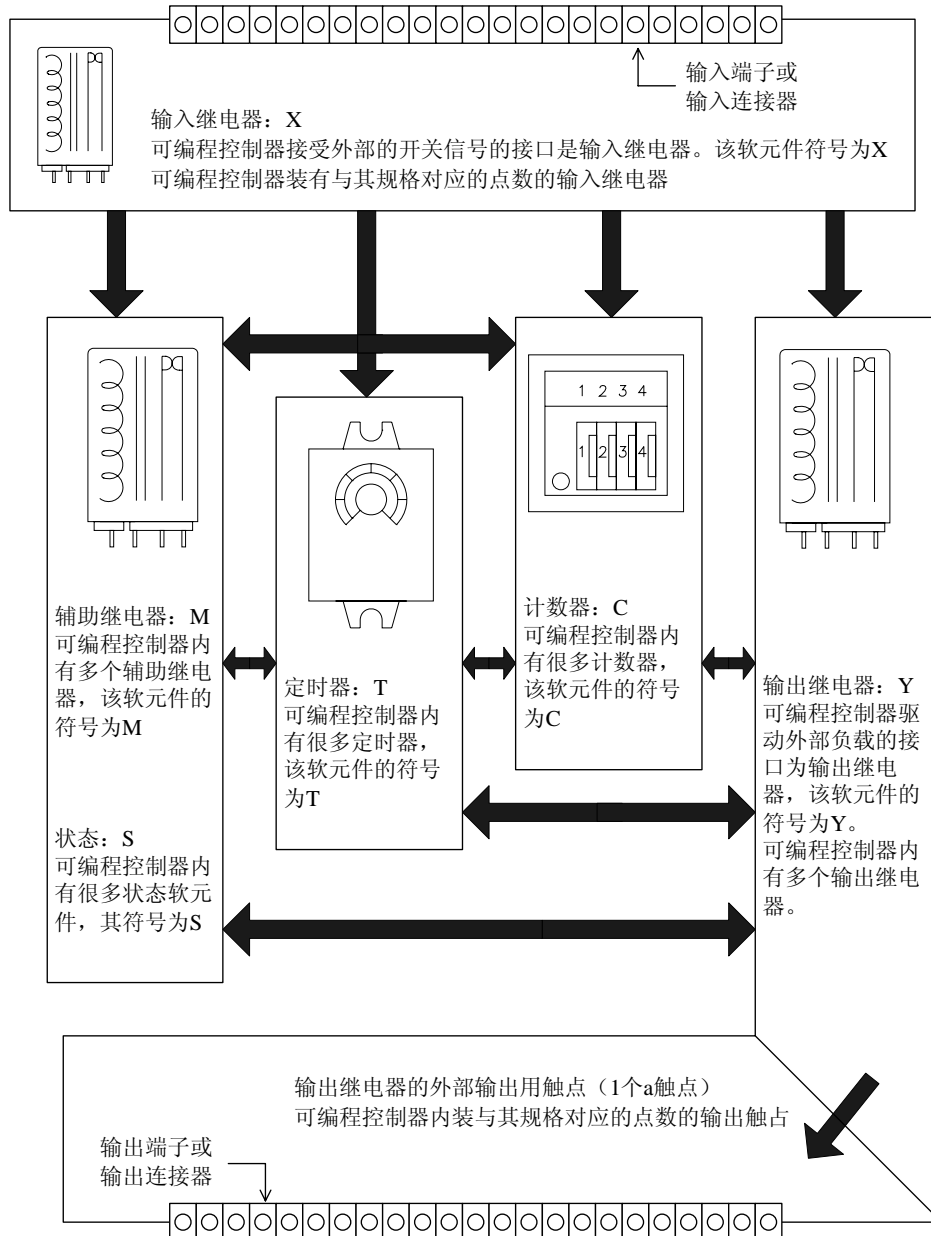


以上三种编制的顺控程序。全部以指令表方式（指令表编程时的内容）存储在可编程控制器的程序中内存中，因此，控照下图所示以各种输入方法编制的程序表示及编辑都可相互交换（即使是指令表程序，也可根据 SFC 转换的规则，通过 SFC 图对应的软件来表示以指令为基础的程序）



2 构成可编程控制器的各种软元件

在可编程控制器内有很多继电器，定时器与计数器，他们都具有无数的 a 触点（常开触点）与 b 触点（常闭触点）。这些触点与线圈相连构成了顺控回路。箭头表示信号的传递。此外，在可编程控制器中还有用于保存数据的记忆软元件——数据寄存器（D），可用于运行时暂存数据的非记忆软元件——（W）。



各种软元件的说明：

输入输出继电器 (X、Y)

在各基本单元中，按 X000—X007，X010—X017。。。Y000—Y007，Y010—Y017。。。等 8 进制数的方式分配输入继电器，输出继电器的地址号，扩展模块的地址号，接在基本单元后面，以 8 进制方式依次分别对 X、Y 编号。

1. PLC 构成

除继电器 X、Y 的编号为 8 进制数的方式外，以下其余软元件编号都是 10 进制。

辅助继电器(M)

辅助继电器是可编程控制器内部具有的继电器，这种继电器有别于输入输出继电器，它不能获取外部的输入，只在程序中使用。有的保持用继电器在可编程控制器停电的情况下也能保存其 ON/OFF 的状态。

步进继电器(S)

作为步进梯形图或 SFC 表示的工序号使用的继电器。不作为工序号使用时，与辅助继电器一样，可作为普通的触点、线圈进行编程，也可作为信号报警器，用于外部故障诊断。

定时器(T)

定时器对可编程控制器内 1ms,10ms,100ms 等时钟脉冲进行加法计算，当达到规定的设定值时，输出触点动作。利用基于时钟脉冲的定时器，可检测到 0.001-3276.7 秒。

BSP01 AR/T 机种 T192-T199, T246-T249 为子程序和中断程序专用的定时器。

BSP01 SR 机种 T196-T199, T246-T249 为子程序和中断程序专用的定时器（详见第 2 章软元件）。

T246-T255 之定时器线圈的驱动输入断开，当前值仍继续累计动作。其余定时器被清 0。

计数器(C)

计数器以不同的用途和目的分以下种类：

<<内部计数用>> 一般使用/停电保持用

16 位计数器：供增计数使用，计数范围：1-32,767

32 位计数器：供增/减计数使用，计数范围：-2,147,483,648~+2,147,483,647

这些计数器供可编程控制器的内部信号使用，其响应速度通常为 10Hz 以下(0.1s)。

<<高速计数用>> 停电保持用

32 位计数器：供增/减计数使用，计数范围：-2,147,483,648~+2,147,483,647（单相单计数，单相双计数，双相双计数）分配给特定的输入继电器。高速计数器可进行 100kHz 的计数，而与可编程控制器的扫描周期无关。

数据寄存器(D), (W), (V), (Z)

数据寄存器 D 是存储数据用的软元件，可编程控制器的数据寄存器都是 16 位（最高位为符号位），数值范围：-32768~32767，将两个寄存器组合可进行 32 位（最高位为符号位）的数据处理。数值范围：-2,147,483,648~+2,147,483,647。跟其它软元件一样数据寄存器也有供一般使用和停电保持使用两种。

补充数据寄存器 W 是运行时暂存数据的软元件，可编程控制器的补充数据寄存器都是 16 位（最高位为符号位），数值范围：-32768~32767，将两个寄存器组合可进行 32 位（最高位为符号位）的数据处理。数值范围：-2,147,483,648~+2,147,483,647。所有 W 掉电均不保存。

在数据寄存器中，还有供变址（地址索引）用的 Z、V 寄存器。Z、V 寄存器与其它软元件一起使用如下所示：

若 V0=3, Z0=5 时, D100V0=D103 C20Z0=C25 ← 元件编号+V[]或 Z[]的值
数据寄存器与变址寄存器可用于定时器与计数器的设定值的间接指定和应用指令中

常数(K), (H)

在可编程控制器所使用的各种各样的数值中，K 表示 10 进制整数，H 表示 16 进制数值，它们被用作定时器与计数器的设定值，或应用指令的操作数。

指针(P), (I)

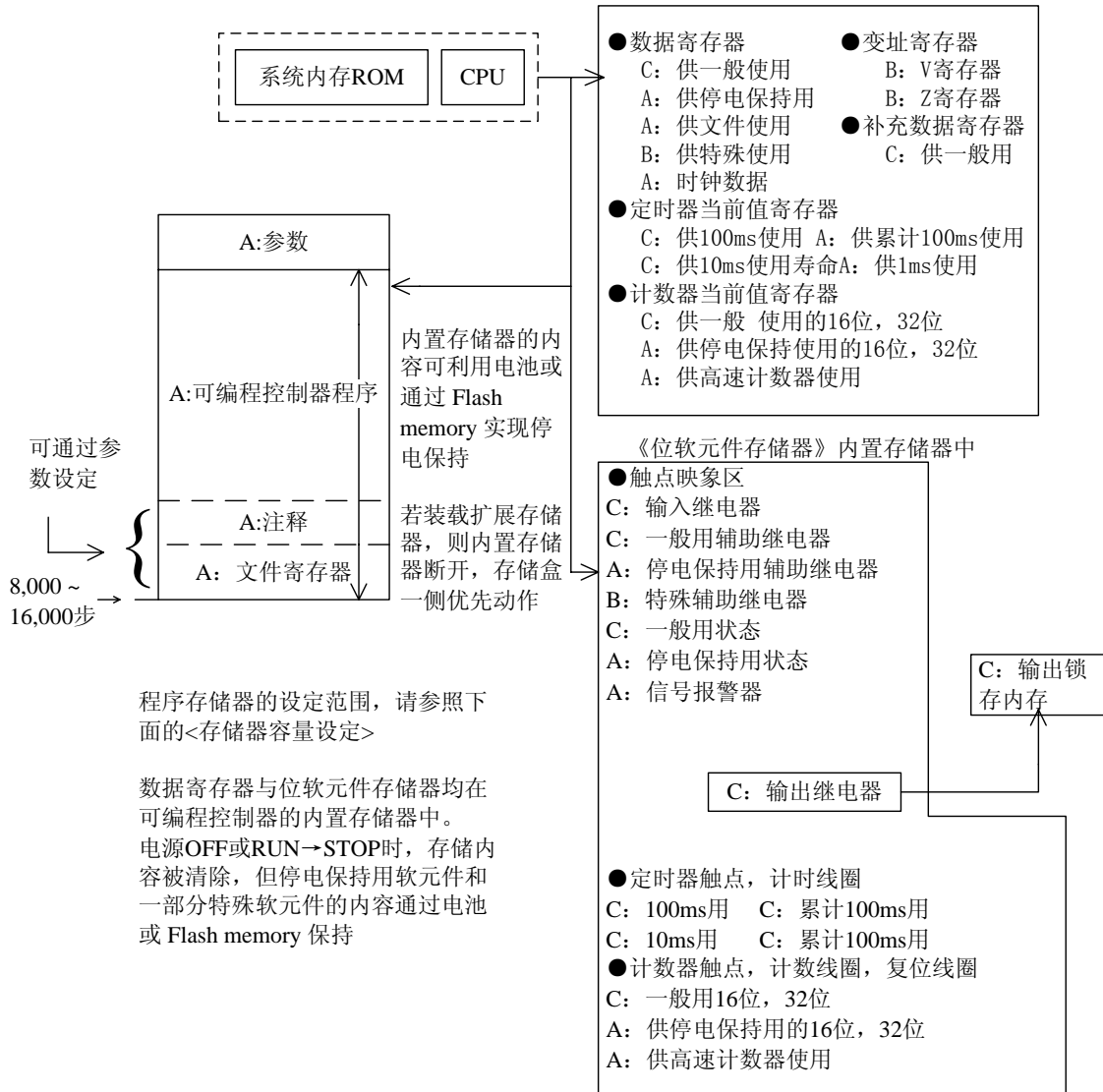
1. PLC 构成

指针用于分支与中断，分支用的指针 **P** 用于指定 **F00(CJ)**条件跳转或 **F01(CALL)**子程序的跳转目标。中断用的指针 **I** 用于指定输入中断，定时中断和计数器中断的中断程序

3 程序内存和参数的构造

存储器的构造

可编程控制器的存储器结构如所示，此外，存储器内的各软元件依据其初始化内容，可分为 A, B, C 3 种



存储器的种类	电源 OFF	电源 OFF→ON	STOP→ RUN	RUN→STOP
A: 电池、Flash memory 后备支持系列存储器	无变化			
B: 特 M, 特 D, 变址寄存器	清除	初始值设定☆	无变化☆	
C: 其它的非后备支持系列的存储器	清除		无变化	清除
			M8033 驱动无变化	

☆ 所示部分在 STOP→ RUN 时会被清除, 因此请注意

1. PLC 构成

参数的构造

参数用于规定停电保持软组件的范围，注释与文件寄存器的容量，而且参数的设定与变化都利用 BAPS-SP 进行，关于设定方法与操作细节，请参阅 BAPS-SP 帮助档。各参数的功能请参阅 BSP01 操作手册。

<<参数的种类与设定的内容>>

- ① 存储器容量的设定：D8006。
- ② 锁存范围的设定：可改变可编程控制器停电保持软组件的范围。
- ③ 口令等级
可设定口令，用于防止已编好的顺控程序的错误写入或被盗用，而且，对于编程软件的在线操作可以通过口令来设定 3 级保护层次。
- ⑤ 其它参数：可设定 RUN/STOP 功能的有效/无效、可指定无电池运行模式、可设定 PC 通用通信。

<<参数设定的初始值>>

BSP01 AR/T 機種

项目		初始值	PC 软件
存储器容量	程序容量	8K (30 点) / 16K (40&60 点)	⊙
	注释容量	0	⊙
锁存范围 (停电保持 领域)	辅助继电器 (M)	500-1023 (0-1023)	⊙
	状态 (S)	500-999 (0-1023)	⊙
	计数器 (C) (16)	100-199 (0-199)	⊙
	计数器 (C) (32)	200-255 (200-255)	⊙
	数据寄存器 (D)	200-511 (0-511)	⊙
密码		无	⊙
RUN 端输入设定		无	⊙
RUN 端输入编号		无	⊙
PC 通用通信设定		无	⊙

⊙：可变更

BSP01 SR 機種

项目		初始值	PC 软件
存储器容量	程序容量	4K (14&20&26&36 点)	⊙
	注释容量	0	⊙
锁存范围 (停电保持 领域)	辅助继电器 (M)	500-1023 (0-1023)	⊙
	状态 (S)	500-999 (0-1023)	⊙
	计数器 (C) (16)	90-99 (0-99)	⊙
	计数器 (C) (32)	220-255 (220-255)	⊙
	数据寄存器 (D)	400-511 (400-511)	⊙
密码		无	⊙
RUN 端输入设定		无	⊙
RUN 端输入编号		无	⊙

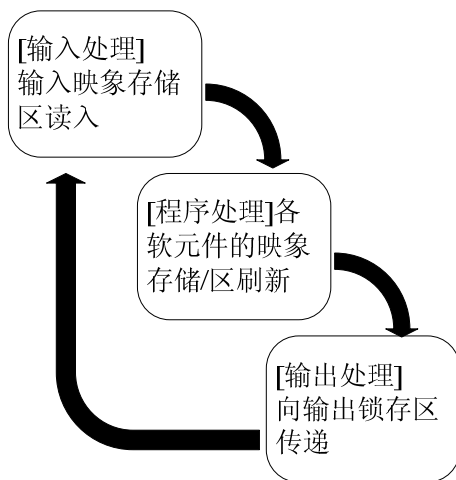
1. PLC 构成

PC 通用通信设定	无	⊙
-----------	---	---

⊙：可变更

4 注意事项(输入输出处理, 响应滞后, 双线圈)

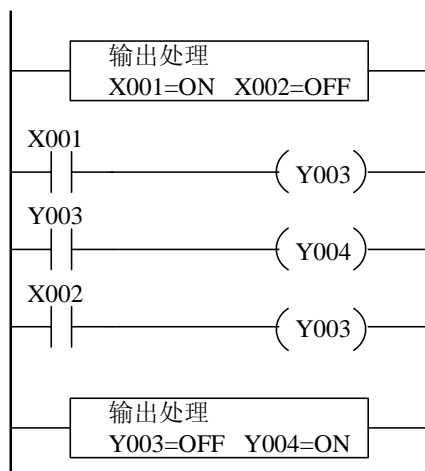
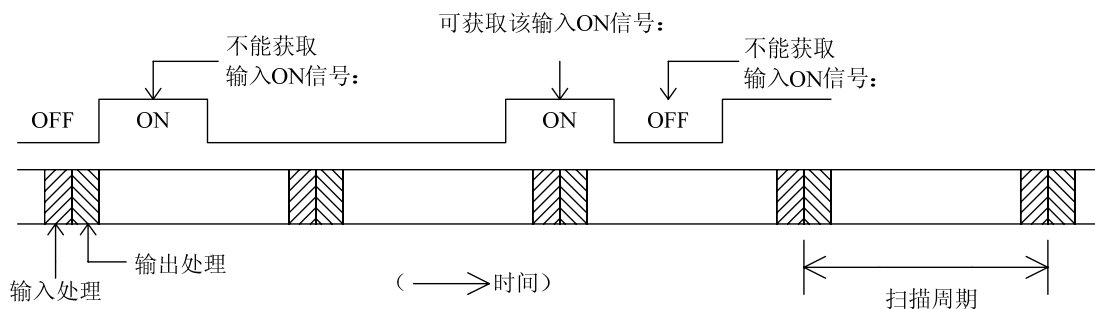
<< 输入输出继电器的动作时序和响应滞后 >>



<< 输入脉冲信号宽度的限制 >>

可编程控制器输入 ON/OFF 的时间宽度应比可编程控制器的循环时间长, 若考虑输入滤波器的响应滞后为 2ms, 循环时间为 10ms, 则 ON、OFF 的时间分别需要 12ms.

因此, 不能处理 $1,000/(12+12)=40\text{Hz}$ 以上的输入脉冲, 但是, 若采用可编程控制器的特殊功能与应用指令, 可改进这方面的情况。



如左图所示, 请考虑在多处使用同一个线圈 Y003 的情况:

例如: 取 X001=ON, X002=OFF

最初的 Y003 由于 X001 为 ON, 其映像存储区为 ON, 输出 Y004 也为 ON。

但是, 第二次的 Y003, 由输入 X002 为 OFF, 因此, 其映像存储区被改写为 OFF。

因此, 实际的外部输出为 Y003=OFF, Y004=ON

2. 各种软组件的作用与功能

第二章 软组件.....	12
1 数值的处理, 常数 K、H.....	12
2 软组件编号一览表.....	13
3 输入输出继电器的编号和功能(X/Y).....	15
4 辅助继电器的编号与功能 M.....	17
5 状态的编号和功能 S.....	19
6 定时器的编号与功能 T.....	21
7 计数器的编号与功能 C.....	25
8 内置高速计数器的编号和功能 C.....	29
9 数据寄存器的编号与功能 D.....	30
9.1 资料寄存器 D.....	30
9.2 补充资料寄存器 W.....	33
9.3 变址寄存器 V, Z.....	34
10 指标的编号与功能 P/I.....	35

第二章 软组件

1 数值的处理，常数 K、H

数值的处理

10 进制数 DEC	<ul style="list-style-type: none"> ● 定时器与计数器的设定值 K。 ● 辅助继电器 M、定时器 T、计数器 C、状态 S 等的编号（软组件编号）。 ● 指定应用指令操作数中的数值与指令动作 K。
16 进制数 HEX	<ul style="list-style-type: none"> ● 如 10 进制数，指定应用指令操作数中的数值与指令动作 H。
2 进制数 BIN	<ul style="list-style-type: none"> ● 以十进制、十六进制数对定的定时器，计数器或数据寄存器进行数值指定，但在可编程控制器内部，这些数字都用二进制数处理。而且在 BAPS-SP 上进行监控时，这些软组件自动变换为十进制或十六进制数。
8 进制数 OCT	<ul style="list-style-type: none"> ● 输入/输出继电器的软组件编号按 8 进制数进行，因此可进行 0~7、10~17、……70~77，100~107 等的编号，在 8 进制数中不存在 8 与 9。
BCD 码	<ul style="list-style-type: none"> ● BCD 码是以 4 位二进制表示十进制 0~9 数值的方法，各位处理很容易，因此可用于 BCD 输出形式的数字式开关或七段码的显示器控制等。
常数 K, H	<ul style="list-style-type: none"> ● K 是表示 10 进制整数的符号。 ● H 是表示 16 进制整数的符号。 <p>在编程用的 BAPS-SP 上进行指令数值的相关操作时，十进制数加 K 后输入，十六进制数加 H 后输入，如 K10，H102。</p>

数值的变换

	8 进制数 OCT	10 进制数 DEC	16 进制数 HEX	2 进制数 BIN		BCD	
	0	0	00	0000	0000	0000	0000
	1	1	01	0000	0001	0000	0001
	2	2	02	0000	0010	0000	0010
	3	3	03	0000	0011	0000	0011
	4	4	04	0000	0100	0000	0100
	5	5	05	0000	0101	0000	0101
	6	6	06	0000	0110	0000	0110
	7	7	07	0000	0111	0000	0111
	10	8	08	0000	1000	0000	1000
	11	9	09	0000	1001	0000	1001
	12	10	0A	0000	1010	0001	0000
	13	11	0B	0000	1011	0001	0001
	14	12	0C	0000	1100	0001	0010
	15	13	0D	0000	1101	0001	0011
	:	:	:	:	:	:	:
	143	99	63	0110	0011	1001	1001
主要用途	输入、输出继电器的软元件编号	除常数以及输入、输出继电器外的内部软元件编号	常数 H	可编程控制器内部的处理		BCD 数字式开关，7 段的显示器	

2. 各种软组件的作用与功能

2 软组件编号一览表

BSP01 系列可编程控制器有 4 种基本的编程元件, 为了分辨各种编程软元件, 给它们指定了不同的符号, 如下:

X: 输入继电器, 用于存放外部输入电路的通断状态。

Y: 输出继电器, 用于从 PLC 直接输出物理信号。

M: 辅助继电器和 S: 状态继电器: PLC 内部的运算标志。

软组件一览表:

BSP01 AR/T 机种

类别 \ 输入/输出	BSP01-30	BSP01-40	BSP01-60	增加扩展
继电器 X X000~X377 256 点	X000~X017 16 点	X000~X027 24 点	X000~X043 36 点	X000~X377 256 点
继电器 Y Y000~Y377 256 点	Y000~Y005 14 点	Y000~Y017 16 点	Y000~Y027 24 点	Y000~Y377 256 点

BSP01 SR 机种

类别 \ 输入/输出	BSP01-14SR	BSP01-20SR	BSP01-26SR	BSP01-36SR	增加扩展
继电器 X X000~X377 256 点	X000~X007 8 点	X000~X013 12 点	X000~X017 16 点	X000~X023 20 点	X000~X117 80 点
继电器 Y Y000~Y377 256 点	Y000~Y005 6 点	Y000~Y007 8 点	Y000~Y011 10 点	Y000~Y017 16 点	Y000~Y117 80 点

继电器 X, Y 共有 512 点, 但因实际 I/O 模块仅能使用上述之点数当作输出入用。其余点数可当辅助继电器用。继电器 X, Y 的编号是 8 进制, 如: X000~X007, X007 接着为 X010, 而非 X008。

BSP01 AR/T 机种

辅助继电器 M	M0~M499 500 点 一般用※1	M500~M1023 524 点 保持用※2	M1024~M7679 6656 点 保持用※3	M8000~M8511 512 点 特殊用※4
状态继电器 S	S0~S499 500 点 一般用 ※1 初始化用 S0~S9 / 原点 回归用 S10~S19	S500 ~ S1023 524 点 保持用※2 S900 ~ S999 100 点 警示用	S1024 ~ S4095 3072 点 保持用※3	
定时器 T	T0~T199 200 点 100ms (子程序用: T192~T199) ※5	T200~T245 46 点 10ms ※5	T246~T249 4 点, 1ms 累计 (子程序用: T246~T249)※3	T250~T255 6 点, 100ms 累 计※3 T256~T511 256 点, 1ms ※5
计数器 C	16 位加计数 C0~C99 100 点 一般用※1	32 位加计数 C100~C199 100 点 保持用※2	32 位高速加计数 C220~C234 35 点 保持用※2	C235~C245 1 相 1 输入 ※2 C246~C249 1 相 2 输入 ※2 C251~C254 2 相输入 ※2

2. 各种软组件的作用与功能

资料寄存器 D	D0~D199 200点 一般用※1	D200~D511 312点 保持用※2	D512~D7999 7488点 档案用 ※3 (D2000~D3299 可作为文 件寄存器使用)	D8000~D8511 512点 特殊用 ※4	
资料寄存器 V、Z	V0~V15、Z0~Z15 32点 (间接指定用)				
嵌套指针 N、P、I	N0~N7 8点 主控回路用	P0~P255 256点 跳跃、子程序分 歧指针	I00X~I50X 6点 中断输入用指标	I6XX~I8XX 3点 时间中断用指标	I010~I060 6点 计数中断用指标
常数	K	16位 -32,768~32,767		32位 -2,147,483,648~2,147,483,647	
	H	16位 0~FFFFH		32位 0~FFFFFFFFH	

BSP01 SR 机种

辅助继电器 M	M0~M499 500点 一般用※1	M500~M1023 524点 保持用※2	M1024~M1535 512点 保持用※3	M8000~M8511 512点 特殊用※4	
状态继电器 S	S0~S499 500点 一般用 ※1 初始化用 S0~S9 / 原点 回归用 S10~S19	S500~S1023 524点 保持用※2 S900~S999 100点 警示用			
定时器 T	T0~T39, T196~T199 44点 100ms (子程序用: T196~T199) ※5	T200~T245 46 点, 10ms ※5	T246~T249 4点, 1ms 累计 (子程序用: T246~T249)※3	T250~T255 6点, 100ms 累计※3	
计数器 C	16位加计数		32位高速加减算		
	C0~C89 90点 一般用※1	C90~C99 10点 保持用※2	C220~C234 15点 保持用※2	C235~C238 C241~C242 1相 1输入※2	C246~C247 1相 2输入 ※2
资料寄存器 D	D0~D399 400点 一般用※1	D400~D511 112点 保持用※2	D8000~D8511 512点 特殊用 ※4		
资料寄存器 V、Z	V0~V15、Z0~Z15 32点 (间接指定用)				
嵌套指针 N、P、I	N0~N7 8点 主控回路用	P0~P127 128点 跳跃、子程序分 歧指针	I00X~I30X 4点 中断输入用指标	I6XX~I8XX 3点 时间中断用指标	I010~I060 6点 计数中断用指标
常数	K	16位 -32,768~32,767		32位 -2,147,483,648~2,147,483,647	
	H	16位 0~FFFFH		32位 0~FFFFFFFFH	

※1：非停电保持范围。但由参数设定，可变更停电保持范围。

※2：停电保持范围。但由参数设定，可变更非停电保持范围。

※3：停电保持固定范围。不可变更保持范围。

※4：参照特殊元件一览表。

※5：非停电保持范围。且无法由参数设定变更停电保持范围。

3 输入输出继电器的编号和功能(X/Y)

输入输出继电器

输入输出继电器的编号

输入、输出继电器的编号是由基本单元固有地址号和按照与这些地址号相连的顺序给扩展设备分配的地址号组成的，这些地址号使用 8 进制数表示，遵循“逢八进一”的运算规则。例如：在八进制数 17 与 20 是相邻的两个整数。

BSP01 AR/T 机种

类别 \ 输入/输出	BSP01-30	BSP01-40	BSP01-60	增设时
继电器 X X000~X377 256 点	X000~X017 16 点	X000~X027 24 点	X000~X043 36 点	X000~X377 256 点
继电器 Y Y000~Y377 256 点	Y000~Y005 14 点	Y000~Y017 16 点	Y000~Y027 24 点	Y000~Y377 256 点

BSP01 SR 机种

类别 \ 输入/输出	BSP01-14SR	BSP01-20SR	BSP01-26SR	BSP01-36SR	增设时
继电器 X X000~X377 256 点	X000~X007 8 点	X000~X013 12 点	X000~X017 16 点	X000~X023 20 点	X000~X117 80 点
继电器 Y Y000~Y377 256 点	Y000~Y005 6 点	Y000~Y007 8 点	Y000~Y011 10 点	Y000~Y017 16 点	Y000~Y117 80 点

输入继电器是 PLC 接收外部输入的开关量信号的窗口。PLC 通过光耦合器，将外部信号的状态读入并储存在输入映像寄存器中。输入端可以外部接常开触点或常闭触点，也可接多个触点组成的串并联电路或电子传感器（例如，接近开关）。在梯形图中，可以多次使用输入继电器的常开触点和常闭触点。

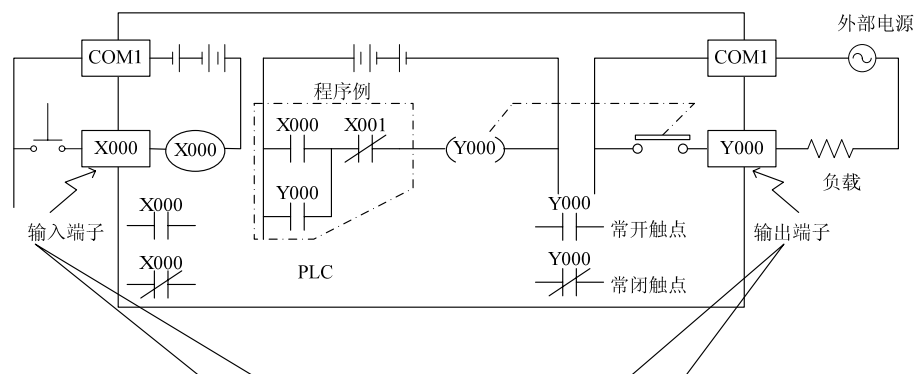
输出继电器的 PLC 向外部负载发送信号的窗口，输出继电器用来将 PLC 的输出信号传送给输出模块，再由后者驱动外部负载。

作用与功能

下图为 PLC 控制系统的示意图，X000 端子外部的输入电路接通时，它对应的输入映像寄存器为‘1’，断开时为‘0’。输入继电器的状态唯一取决于外部输入信号的状态，不受用户程序的控制，因此在梯形图中绝对不能出现输入继电器的线圈。

当 Y000 线圈导通时，继电器型输出模块中对应的硬件继电器的常开触点闭合，使外部负载工作。输出模块中的每一个继电器仅有一个常开触点，但是在梯形图中，每一个输出继电器的常开触点和常闭触点都可以多次使用。

2. 各种软组件的作用与功能

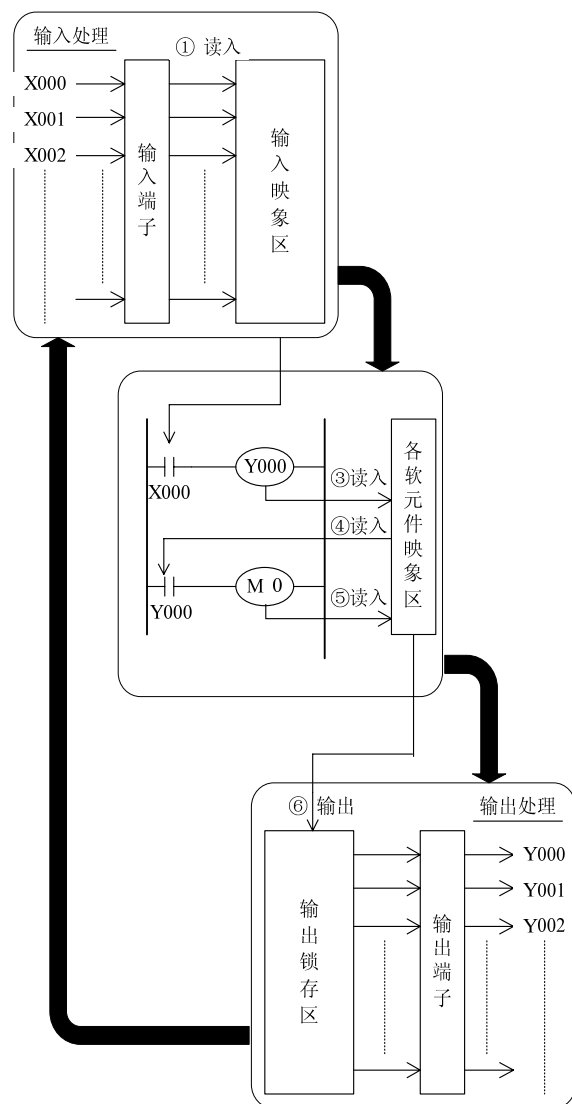


输入端子是接受外部输入开关信号的端子，PLC输入端子连接的输入元件X是具有光耦合特性的电子继电器，拥有无数常开触点（a接点）及常闭触点（b接点），此接点可在PLC内部自由使用，输入继电器是无法用程序来驱动的。

输出端子是与外部负载连接的端子。输出继电器是外部输出用接点。（晶体管，双向闸流体，继电器在PLC内是连接于输出端子）在PLC的输出继电器拥有无数电子式常开/常闭接点，可任意使用。外部输出接点和内部接点的动作如下页所示。

输入继电器的动作时序

可编程控制器通过反复执行下面所示的处理顺序，进行顺序控制，采用这种成批输入输出方式时，输出滤波器和输出组件的驱动时间及运算周期也会出现响应滞后的情况。



●输入处理

可编程控制器在执行程序之前，将可编程控制器的所有输入端子的 ON/OFF 状态，读入映象区。

在执行程序的过程中，即使输入变化，输入映象区的内容也不变，而在下一周期的输入处理时，读入该变化。

此外，即使输入触点出现 ON→OFF，OFF→ON 的变化时，在判定 ON/OFF 之前，输入滤波器会造成响应滞后约 10ms。

●程序处理

可编程控制器根据程序内存的指令内容，从输入映象区或其它软组件的映象区读出各软组件的 ON/OFF 的状态。从 0 步开始依次进行运算，然后将结果写入映象区，因此各软组件的映象存储区随着程序的内部触点根据输出映象存储区的内容执行动作。

●输出处理

一旦所有指令执行结束，将输出 Y 的映象存储区的 ON/OFF 状态传输至锁存区，这成为了可编程控制器的实际输出。可编程控制器内的外部输出用触点，按照输出用软组件的响应滞后时间动作。

4 辅助继电器的编号与功能 M

辅助继电器的

辅助继电器 (M) 是用软件实现的, 它们不能接受外部的输入信号, 也不能直接驱动外部负载, 是一种内部的状态标志, 相当于继电器控制系统中的中间继电器。

辅助继电器 M 的编号如下表所示: 编号按十进制数分配

BSP01 AR/T 机种

一般用	停电保持用	停电保持专用	特殊用
M0~M499 500 点 ※1	M500~M1023 524 点 ※2	M1024~M7679 6656 点 ※3	M8000~M8511 512 点

BSP01 SR 机种

一般用	停电保持用	停电保持专用	特殊用
M0~M499 500 点 ※1	M500~M1023 524 点 ※2	M1024~M1535 512 点 ※3	M8000~M8511 512 点

※1 非停电保持领域。使用参数设定, 可变更停电保持领域。

※2 停电保持领域。使用参数设定, 可变更非停电保持领域。

※3 停电保持固定范围。不可变更保持范围。

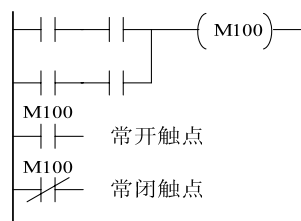
功能与动作的示例:

可编程控制器内有许多辅助继电器, 这类辅助继电器的线圈与输出继电器一样, 由可编程控制器内的各种软组件的触点驱动。

辅助继电器常有无数的电子常开触点与常闭触点, 在可编程控制器内可随意使用, 但是, 该触点不能直接驱动外部负载, 外部负载的驱动要通过输出继电器进行。

一般用

一般用辅助继电器如果在 PLC 突然断电时, 输出继电器和通用辅助继电器将全部变为 OFF。若再次上电, 除了因外部输入信号而变为 ON 的以外, 其余的仍然保持为 OFF。



TP03 可编程控制器内的一般用辅助继电器和停电保持用辅助继电器的分配可通过 BAPS-SP 的参数设定进行调整。

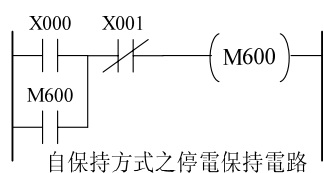
停电保持用

某些控制系统要求记忆停电前的状态, 再运行时再现该状态。

停电保持用辅助继电器又名保持用继电器就是用于上述目的。它利用可编程控制器内装的备用电池或 Flash memory 进行停电保持。它们只是在 PLC 重新上电后的第一个扫描周期保持继电器瞬间的状态。

在将停电保持专用继电器作为一般辅助继电器使用的场合, 应在程序最前面的地方用 RST 或 ZRST 指令清除内容。

此外, 在使用简易 PC 间链接或并接链接时, 一部分辅助继电器作为链接被占用



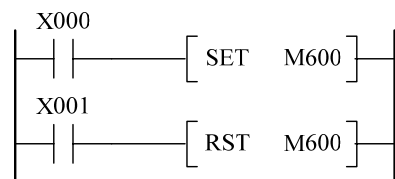
左圖表示了 M600 动作的停电保持示例。在该电路中, 如果 X000 接通, M600 动作的话, 即使 X000 断开, M600 也自己保持动作, 因此, 即便停电造成 X000 开路, 再运行时 M600 也继续动作。

但是, 再运行时如果 X001 的常闭触点开路, 则 M600 不会动作

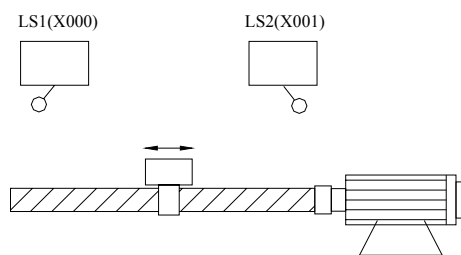
2. 各种软组件的作用与功能

如左图使用 SET、RST 命令时。

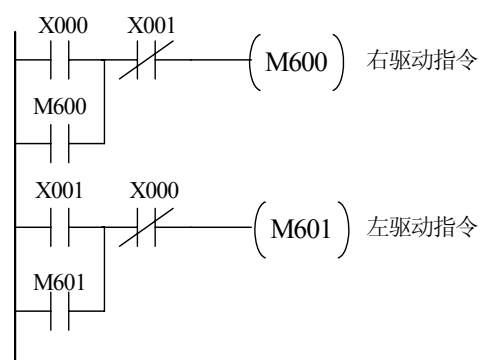
停电保持用的用途示例



設定、清除方式之停電保持電路



再运行时，其前进方向与停电前的前进方向相同。



X000=ON (左限位开关) → M600=ON → 右驱动 → 停电 → 平台中途停止 → 再运行 (M600=ON) → X001=ON (右限位开关) → M600=OFF, M601=ON → 左驱动。

特殊用途

可编程控制器内有 512 个的特殊辅助继电器，这些继电器各有其特定的功能，可分为以下两类：

a. 〈触点利用型的特殊辅助继电器〉：利用可编程控制器的驱动线圈，用户可使用该触点。

M8000：运行监视器

M8002：初始脉冲

M8012：100ms 周期振荡

用户不可使用尚未定义的特殊辅助继电器。

b. 〈线圈驱动型特殊辅助继电器〉：用户驱动这些线圈，则可作特定的运行。

M8033：按要求保持存储器

M8034：输出全部禁止

M8039：恒定扫描

其中存在驱动时有效与 END 指令执行后有效两种情况，请注意

5 状态的编号和功能 S

状态

状态继电器是用于编制顺序控制程序的一种编程元件，它与第四章介绍的 STL、RET 指令一起使用。一般用状态继电器没有停电保持功能，而停电保持状态继电器利用可编程控制器内装的备用电池或 Flash memory 进行保存其 ON/OFF 的状态。

状态的编号 S 如下表所示（编号按 10 进制分配）

BSP01 AR/T 机种

状态继电器 S	S0~S499 500 点 一般用 ※1	S500 ~ S1023 524 点 保持用 ※2	S1024 ~ S4095 3072 点 保持用※3
	初始化用 S0~S9 /原点回归用 S10~S19	S900 ~ S999 100 点 警示用	

BSP01 SR 机种

状态继电器 S	S0~S499 500 点 一般用 ※1	S500 ~ S1023 524 点 保持用 ※2
	初始化用 S0~S9 原点回归用 S10~S19	S900 ~ S999 100 点 警示用

※1 非停电保持领域。使用参数设定，可变更停电保持领域。

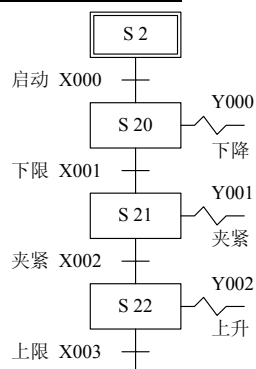
※2 停电保持领域。使用参数设定，可变更非停电保持领域。

※3 有关停电保持的特性，无法用参数来改变。

功能与动作示例

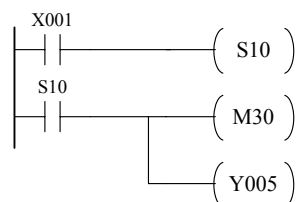
状态 S 是对工序步进控制简易编程的重要软组件，经常与步进梯形图指令 STL 结合使用。

一般用/保持用



如图所示的工序步进控制中，如果启动信号 X000 为 ON，则状态 S20 置位 ON，下降用的电磁阀 Y000 开始动作，其结果是，若下限限位开关 X001 为 ON，则状态 S21 置位 ON，夹紧用的电磁阀 Y001 动作。如果夹紧动作确认的限位开关 X002 为 ON，则状态 S22 置位 ON。随着状态动作的转移，状态自动返回原状态。

一般用状态在电源断开后，都变为 OFF 状态，但停电保持用状态能记忆电源停电前一刻的 ON/OFF 状态，因此，也能从中途工序开始运行。



状态与辅助继电器一样，有无数常开常闭触点，顺控程序内可随意使用。

此外，在不用于步进梯形图指令时，状态 S 也与辅助继电器 M 一样可在一般的顺控中使用。

可编程控制器可通过 BAPS-SP 参数的设定，变更一般用状态和停电用状态的分配。

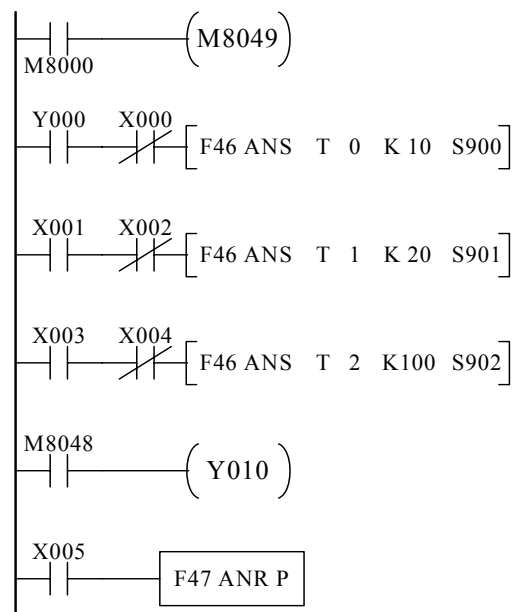
2. 各种软组件的作用与功能

信号报警器用

供信号报警器用的状态，也可作为诊断外部故障用的输出。

例如，编写下图所示的外部故障诊断电路，监视特殊数据寄存器 D8049 的内容，显示出 S900~S999 的动作状态中的最小编号。

当发生多个故障时，消除最小编号的故障即可知道下一个故障的编号，



若驱动特殊辅助继电器 M8049，则监视进入有效状态。

在驱动前进输出 Y000 后，如果前进端检测 X000 在一秒钟内不工作，则 S900 动作。

如果上限 X001 与下限 X002 同时不工作时间超过两秒钟，则 S901 动作。

在间隔时间不满 10 秒的机械中连续运行模式输入 X003 为 ON 时，如果机械的一周运行过程中动作开关 X004 不工作，则 S902 动作。

如果 S900~S999 中的任意项为 ON，则特殊辅助继电器 M8048 动作，故障显示输出 Y010 也会动作。

通过复位按钮 X005 将因为外部故障诊断程序而动作的状态变为 OFF，X005 每 ON 一次，最小编号的动作状态依次复位。

当不驱动特殊辅助继电器 M8049 时，供停电保持用的状态与一般状态一样，可在顺控的程序内使用。

6 定时器的编号与功能 T

定时器的编号

BSP01 AR/T 机种

定时器 T	T0~T199 200 点 100ms (子程序用: T192~T199)	T200~T245 46 点 10ms	T246~T249 4 点, 1ms 累计 (子程序用: T246~T249)	T250~T255 6 点, 100ms 累计	T256~T511 256 点, 1ms
-------	--	------------------------	--	----------------------------	-------------------------

BSP01 SR 机种

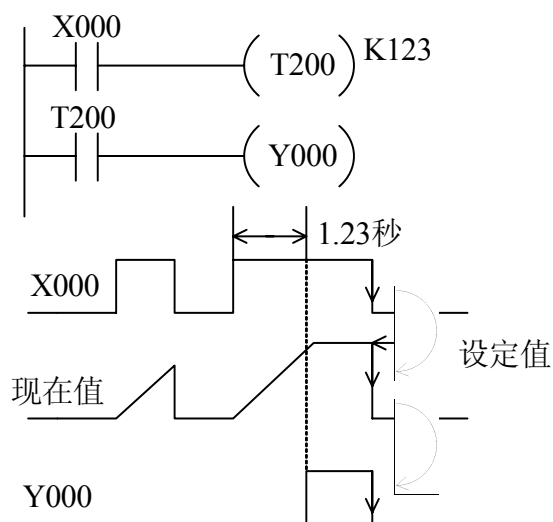
定时器 T	T0~T39, T196~T199 44 点 100ms (子程序用: T196~T199)	T200~T245 46 点 10ms	T246~T249 4 点, 1ms 累计 (子程序用: T246~T249)	T250~T255 6 点, 100ms 累计
-------	--	------------------------	---	----------------------------

不用作定时器的定时器编号, 也可用作数值存储用的资料寄存器

功能

定时器累计可编程控制器内的 1ms, 10ms, 100ms 等的时钟脉冲, 当达到所定的设定值时输出触点动作。设定值可用常数 K 作为设定值, 也用数据寄存器 D 的内容进行间接指定

一般用

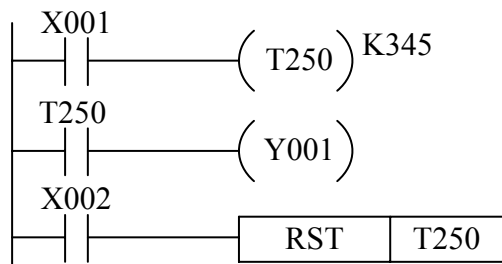


如果定时器线圈 T200 的驱动输入 X000 为 ON, T200 用当前值计数器累计 10ms 的时钟脉冲, 如果该值等于设定值 K123 时, 定时器的输出触点动作。

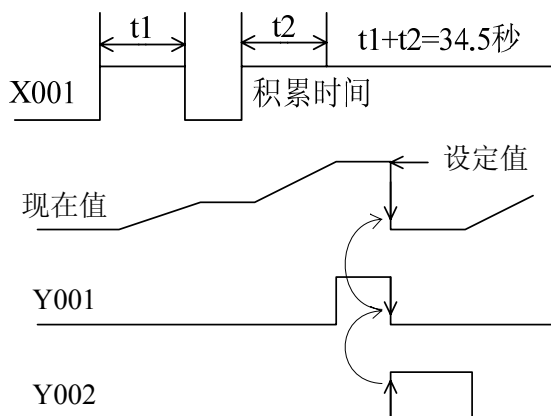
也就是说输出触点在线圈驱动 1.23 秒后动作。

驱动输入 X000 断开或停电, 定时器复位, 输出触点复位。

累计用



2. 各种软组件的作用与功能



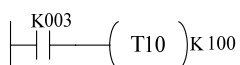
如果定时器线圈 T250 的驱动输入 X001 为 ON 时，则 T250 用当前值计数器将累积积 100ms 的时钟。如果该值达到设定值 K345 时，定时器的输出触点动作。

在计算过程中，即使输入 X001 断开，或停电时，再启动时，继续计算，其累计计算动作时间为 34.5 秒。

如果复位输入 Y002 为 ON 时，定时器复位，输出触点也复位。

设定值的指定方法

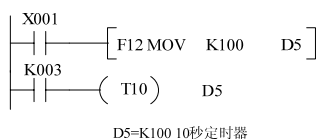
常数指定



T10 是以 100ms (0.1S) 为单位的定时器，将 100 指定为常数，则 $0.1S \times 100 = 10S$ 的定时器工作

K 常数 (10 进制整数) 10 秒定时器

间接指定D



将间接指定数据寄存器的内容预先写入程序或通过数值开关输入。在指定为停电保持用寄存器时，请注意电池电压不足会造成设定值不稳定的情况

数值软组件的处理

定时器的当前值可通过应用指令等作为数值使用。

作为数据软组件使用时，请参照内部计数器的编号与功能。

程序内的注意事项

在子程序与中断程序内请采用 T192-T199 定时器，这种定时器在执行线圈指令或执行 END 指令时计时。

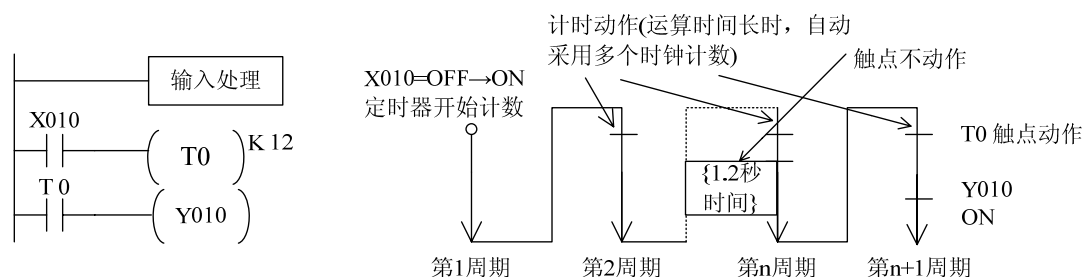
如果计时达到设定值，则在执行线圈指令或 END 指令时。输出触点动作，普通的定时器只是在执行线圈指令时计时，请参照下述定时器动作与定时器精度。因此，仅在某种条件下线圈指令用于执行中的子程序或中断程度时不计时，不能正常动作。

如果在子程序或中断程序内采用 1ms 累计定时器，在其达到设定值后，必须注意的是，在执行最初的线圈指令时，输出触点动作。

2. 各种软组件的作用与功能

定时器动作的细节与定时器的精度

除了中断执行型的定时器外，在线圈驱动后，定时器开始计时，在计时完了后的最初的线圈抛售执行时，输出触点动作。



从上述动作图可知，从驱动线圈开始到触点动作结束的定时器触点动作精度，大致可用下式表示：

$$(T+T_0) \sim (T-\alpha)$$

α ：与 1ms, 10ms, 100ms 定时器对应，分别为 0.001, 0.01, 0.1 秒

T：定时器设定时间(秒)

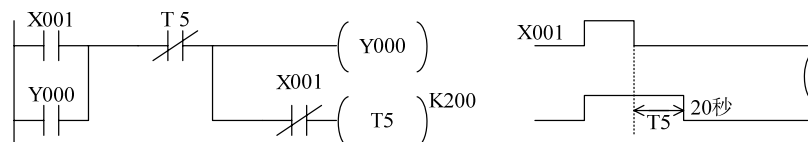
T_0 ：扫描周期(秒)

编程时，定时器触点写在线圈指令前时，最大误差为+2T。

当定时器的设定值为 0 时，在执行下一个扫描的线圈指令时，输出触点开始动作，此外，中断执行型的 1ms 定时器在执行线圈指令后，以中断方式对 1ms 时钟脉冲计数。

动作示例

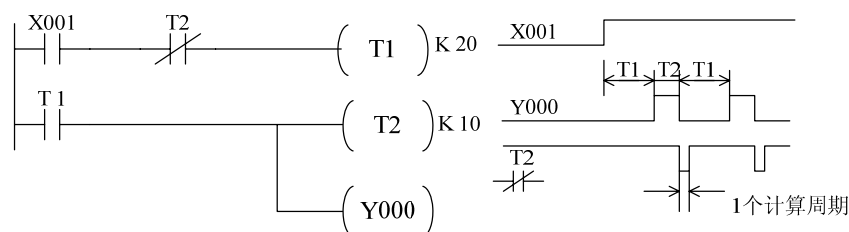
《输出延时关断定时器》



闪烁点

编程时，定时器触点写在线圈指令前时，最大的误差为+2T

当定时器的设定值为 0 时，在执行下一个扫描的线圈指令时，输出触点开始动作，此外，中断执行型的 1ms 定时器在执行线圈指令后，以中断方式对 1ms 时钟脉冲计数。

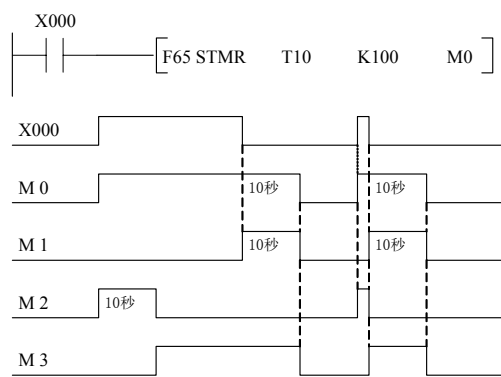


此外，通过利用 F66 ALT 指令也可进行闪烁动作。

2. 各种软组件的作用与功能

<< 采用应用指令 F65 可进行多个定时器 >>

使用这项指令能很容易地制作输出延时关断定时器，单脉冲输出定时器和闪烁定时器。

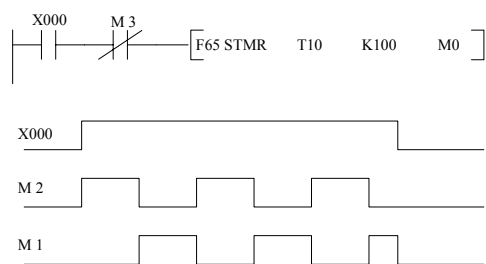


- m 所指定的值为指定定时器的设定值，在本例中为 10 秒。

- M0 为 OFF 延迟定时器。

- M1 为输入点 ON—OFF 后的 ONE SHOT 定时器。

- M2,M3 为闪烁的动作如左图所示。



- M3 之接法如左图，则 M1,M2 不作闪烁之输出动作。

- 当 X000 变成 OFF 时，M0,M1,M3 则变成 OFF，T10 则被复归。

- 在这里使用的定时器请勿重复使用于其它电路上。

此外，如果采用 F64 TTMR 的示教定时器指令，则可以根据开关的输入时间设定定时器的时间。

7 计数器的编号与功能 C

计数器的编号

计数器的编号如下表所示。编号以 10 进制分配

BSP01 AR/T 机种

计数器 C	16 位累计		32 位加减算	32 位高速加减算		
	C0~C99 100 点 一般用※1	C100~C199 100 点 保持用※2	C200~C234 35 点 保持用※2	C235~C245 1 相 1 输入 ※2	C246~C249 1 相 2 输入 ※2	C251~C254 2 相输入 ※2

BSP01 SR 机种

计数器 C	16 位累计		32 位加减算	32 位高速加减算		
	C0~C89 90 点 一般用※1	C90~C99 10 点 保持用※2	C220~C234 15 点 保持用※2	C235~C238 C241~C242 1 相 1 输入 ※2	C246~C247 1 相 2 输入 ※2	C251~C253 2 相输入 ※2

※1：非停电保持领域。使用参数设定，可变更停电保持领域。

※2：停电保持领域。使用参数设定，可变更非停电保持领域。

<< 32 位计数器增计数/减计数切换用的辅助继电器编号 >>

计数器号	方向切换	计数器号	方向切换	计数器号	方向切换	计数器号	方向切换
C200	M8200	C209	M8209	C218	M8218	C227	M8227
C201	M8201	C210	M8210	C219	M8219	C228	M8228
C202	M8202	C211	M8211	C220	M8220	C229	M8229
C203	M8203	C212	M8212	C221	M8221	C230	M8230
C204	M8204	C213	M8213	C222	M8222	C231	M8231
C205	M8205	C214	M8214	C223	M8223	C232	M8232
C206	M8206	C215	M8215	C224	M8224	C233	M8233
C207	M8207	C216	M8216	C225	M8225	C234	M8234
C208	M8208	C217	M8217	C226	M8226	——	——

计数器的特点

16 位计数器与 32 位计数器的特点如下表所示。可按计数方向的切换与计数的范围的使用条件来分开使用。

项目	16 位计数器	32 位计数器
计数方向	顺数	顺/倒可切换使用（看上表）
设定值	0~32,767	-2,147,483,648~+2,147,483,647
指定的设定值	常数 K 或资料寄存器	同左，但数据完成寄存器要一对（2 个）
当前值的变化	顺数后不变化	顺数后变化（循环计数器）
输出接点	顺数后保持动作	顺数保持动作。倒数复位
复位动作	执行 RST 指令时，计数器的当前值为零，输出接点恢复	
当前值寄存器	16 位	32 位

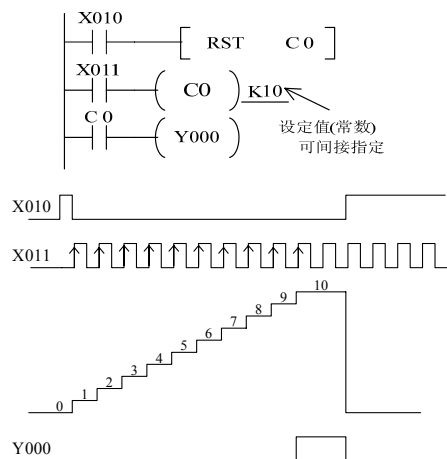
2. 各种软件组件的作用与功能

功能与动作示例

一般用计数器和停电保持用状态的分配，可通过在 BAPS-SP 改变参数设定进行变更。

16 位计数器一般用/停电保持用

16 位 2 进制增计数器，其有效设定值为 K1~K32,767（10 进制常数）。设定值 K0 和 K1 具有相同的含义，即在第一次计数开始时输出触点就动作。



如果切断可编程控制器的电源，则一般用计数器的计数值被清除，而停电保持用的计数器则可存储停电前的计数值，因此计数器可按上次数值累计计数。

计数输入 X011 每驱动 C0 线圈一次，计数器的当前值就增加。在执行第十次的线圈指令时，输出触点动作。以后即使计数输入 X011 再动作，计数器的当前值不变。如果复位输入 X010 为 ON，则执行 RST 指令，计数器的当前值为 0，输出触点复位。

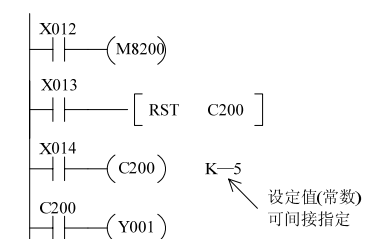
计数器的设定值，除上述常数 K 设定外，还可由数据寄存器编号指定。例如，指定 D10，如果 D10 的内容为

123，与设定 K123 是一样的。

在以 MOV 等指令将设定值以上的数据写入当前值寄存器时，则在下次输入时，输出线圈接通，当前值寄存器变为设定值。

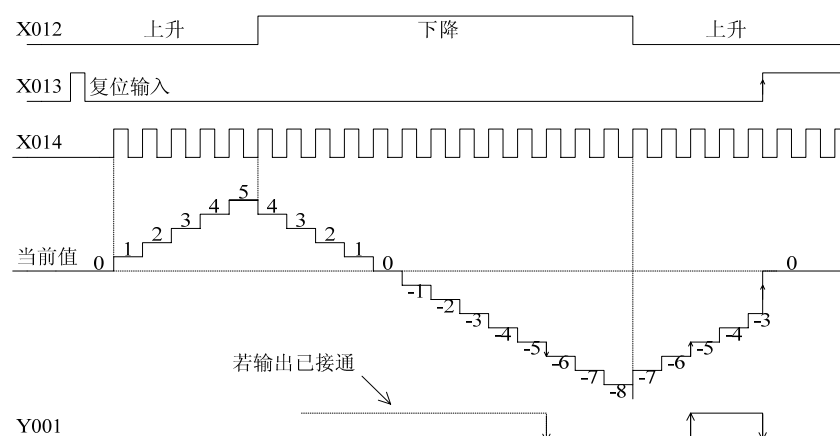
32 位计数器一般用/停电保持用

32 位的 2 进制增计数/减计数的设定值有效范围为 -2,147,483,648~+2,147,483,647（10 进制常数）。利用特殊的辅助继电器 M8200~M8234 指定增计数/减计数的方向。



如果对 C△△△驱动 M8△△△，则为减计数，不驱动时，则为增计数。

根据常数 K 或资料寄存器 D 的内容。设定值可正可负，将连号的资料寄存器的内容视为一对，作为 32 位的数据处理。因此，在指定 D0 时，D1 和 D2 两项作为 32 位设定值处理。



利用计数输入 X014 驱动 C200 线圈时，可增计数或减计数。在计数器的当前值由 -6~-5 增加时，输出触点置位，在由 -5~-6 减少时，输出触点复位。

2. 各种软组件的作用与功能

一般用/停电保持用

当前值的增/减与输出触点的动作无关，但是如果从 2,147,483,647 开始增计数，则成为 -2,147,483,648。同样，如果从 -2,147,483,648 开始减计数，则成为 2,147,483,647。这类动作被称为环形计数。

如果复位输入 X013 为 ON，则执行 RST 指令，计数器的当前值变为 0，输出触点也复位。

使用供停电保持用的计数器时，计数器的当前值，输出触点的动作与复位状态停电保持。

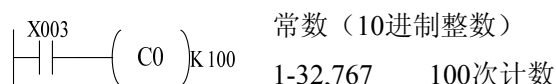
32 位计数器也可作为 32 位数据寄存器使用。但是，32 位计数器不能作为 16 位应用指令中的软组件。

在以 D-MOV 指令等把设定值以上的数据写入当前值数据寄存器时，则在以后计数输入时可继续计数，触点也不变化。

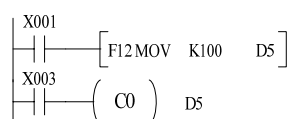
设定值的指定方法

16 位计数器

常数指定K



间接指定D



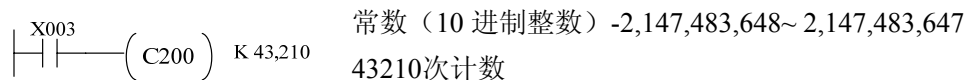
将间接指定数据寄存器的内容预先写入程序或通过数值开关输入

在指定为停电保持用寄存器时，请注意电池电压不足会造成设定值不稳定的情况

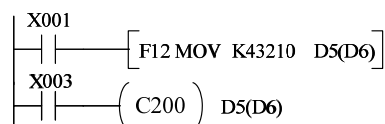
D5=K100 (计数100)

32 位计数器

常数指定K



间接指定D



将间接指定数据寄存器2个为一组，在使用32位命令写入设定值的同时，请不要在其它程序上重复该数据寄存器。

计数器的响应速度

计数器对可编程控制器的内部信号 X, Y, M, S, C 等触点的动作进行循环扫描并计数时，例如：在将 X011 作为计数输入时，其接通和断开的持续时间，必须比可编程控制器的扫描时间长（通常在数值 10Hz 以下）。对于这个问题将涉及到后面即将提及的高速计数器通过对特定的输入作中断处理来进行计数，执行数 KHz 的计数，而与扫描时间无关。

2. 各种软组件的作用与功能

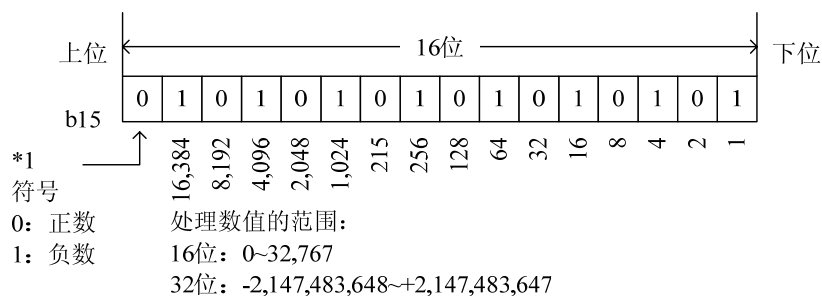
数值软组件的处理

计数器与定时器根据设定值动作，利用此种输出触点时，将计数值（当前值）作为数值用于控制。

计数器的当前值寄存器一样，作为 16 位或 32 位数据软组件处理。

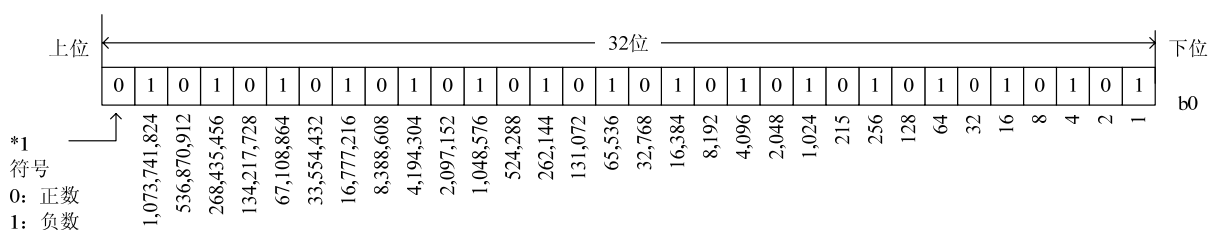
16 位 (C)

《计数器，定时器（仅限于 16 位）的当前值寄存器，设定值寄存器的结构》



*1: 仅作为数据寄存器的代用品使用时有效

32 位 (C)



<< 应用指令的使用示例 >>

X000	[F12 MOV C20 D10]	C20(当前值)→D10传送
X000	[F10 CMP K100 C30 M0]	将10进制整数100和C30(当前值)比较, 将其结果输出到 M0~M2
X000	[F18 BCD C10 K2Y000]	将C10(当前值)的内容作BCD变换, 输出到 Y000~Y007。控制七段码显示器
X000	[F22 MUL C5 K2 D4(D5)]	将C5(当前值)翻倍, 向D5, D4传送
X000	[F12 DMOV C200 D0(D1)]	C200(当前值)→D1, D0传送
X000	[F11 DZCP K100 K20000 C200 M10]	

将C200(当前值)与10进制整数100~20,000的区间作比较，将其结果向M10~M11输出

应用指令的使用示例，如何将计数器与定时器作为数值软组件活用的方法，请参照后述的指令说明。

8 内置高速计数器的编号和功能 C

内置高速计数器的编号

各可编程控制器的内置高速计数器的如下表所示：

按计数器的编号 C 分配在输入 X000~X005，X000~X005 不可重复使用

而不作为高速计数器使用的输入编号可在顺控程序作为普通的输入继电器使用。

此外，不作为高速计数器使用的高速计数器编号也可作为数值存储用的 32 位数据寄存器使用。

U：增计数输入；D 减计数输入，A：A 相输入

B：B 相输入；R：复位输入，S：启动输入

BSP01 AR/T 机种

	1 相 1 计数输入										1 相 2 计数输入				2 相 2 计数输入					
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C251	C252	C253	C254	
X000	U/D						U/D			U/D		U	U		U		A	A		A
X001		U/D					R			R		D	D		D		B	B		B
X002			U/D					U/D			U/D		R		R		R	A		R
X003				U/D				R			R			U				B		
X004					U/D				U/D	S				D				R		
X005						U/D			R		S			R	S					S

C250 / C255 保留不能使用

BSP01 SR 机种

	1 相 1 计数输入										1 相 2 计数输入				2 相 2 计数输入					
	C235	C236	C237	C238			C241	C242				C246	C247			C251	C252	C253		
X000	U/D						U/D					U	U			A	A			
X001		U/D					R					D	D			B	B			
X002			U/D					U/D				R				R	A			
X003				U/D				R									B			

C239~C240,C243~C245,C248~C250,C254~C255 保留不能使用

{表的阅读法：}

输入 X000，C235 单相单输入计数，不具有中断复位与中断启动输入功能。

如果使用 C235，则不可使用 C241，C244，C246，C247，C249，C251，C252，C254 和中断指标 I00□。

其余功能请参阅操作手册 4. 高速计数器

9 数据寄存器的编号与功能 D

9.1 资料寄存器 D

资料寄存器的编号

资料寄存器 D 的编号如下表所示 (编号以 10 进制分配) :

BSP01 AR/T 机种

资料寄存器 D	D0~D199 200 点 一般用 ※1	D200~D511 312 点 保持用 ※2	D512~D7999 7488 点 档案用 ※3 (D2000 ~D3299 可作为文件寄存器使用)	D8000~D8511 512 点 特殊用 ※4
---------	----------------------------	------------------------------	--	--------------------------------

BSP01 SR 机种

资料寄存器 D	D0~D399 400 点 一般用 ※1	D400~D511 112 点 保持用 ※2	D8000~D8511 512 点 特殊用 ※4
---------	----------------------------	------------------------------	--------------------------------

※1：非停电保持领域。使用参数设定，可变更停电保持领域。

※2：停电保持领域。使用参数设定，可变更非停电保持领域。

※3：停电保持固定范围。不可变更保持范围。

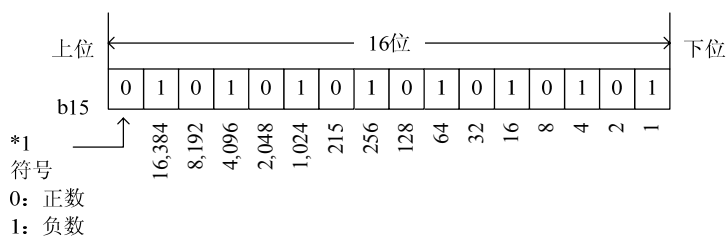
※4：参照特殊组件一览表。

寄存器的结构与功能

数据寄存器是存储数值数据的软组件，其种类如下所示。这些寄存器都是 16 位（最高位正负符号），将两个数据寄存器组合，可存储 32 位（最高位为正负符号）的数值数据。

16 位 (D)

一个数据寄存器（16位的数值范围为：）-32,768~+32,767。



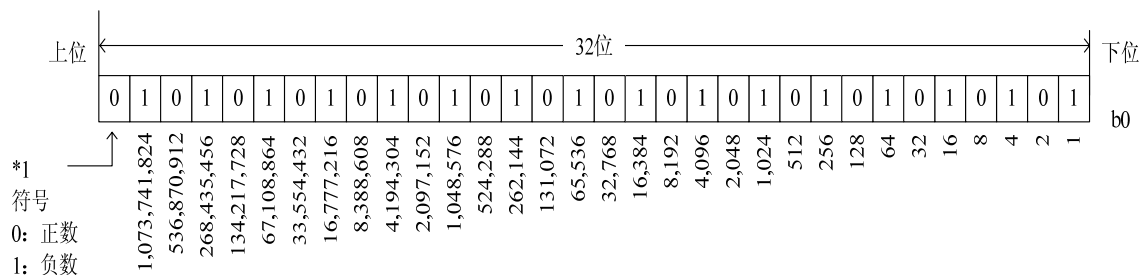
数据寄存器的数值读出与写入一般采用应用指令。此外，也可从数据还取单元（显示器）与编程设备直接读出/写入。

2. 各种软组件的作用与功能

32 位 (D)

以两个相邻的数据寄存器表现 32 位的数据。(高位为大的号码, 低位为小的号码, 在变址寄存器中, V 为高位, Z 为低位。)

因此可处理-2,147,483,648~+2,147,483,647 的数值。



在指定 32 位时, 如果指定了低位(例:D0), 则高位为继其之后的编号(例如: D1)会自动占有。低位可用偶数或奇数的任意一种软元件编号指定, 考虑 BAPS-SP 监视功能, 建议指定低位采用偶数软元件编号。

一般用/停电保持用

一旦在数据寄存器中写入数据, 只要不再写入其它数据, 就不会变化, 但是, 在 RUN→STOP 时或停电时, 所有数据被清除为 0。如果驱动特殊的辅助继电器 M8033, 则可以保持。对此相对停电保持用的数据寄存器在 RUN/STOP 和停电时也可保持其内容。

利用 BAPS-SP 的参数设定, 可改变可编程控制器的一般用与停电保持用的分配, 除停电保持专用的软组件范围外。

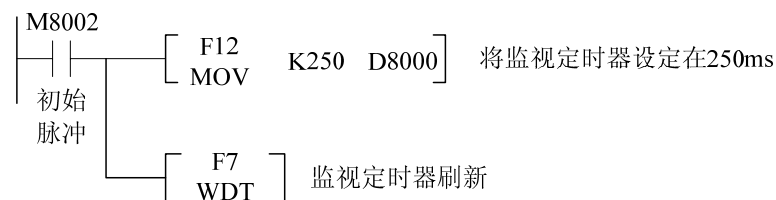
而且将停电保持专用的数据寄存器作为一般用途时, 请在程序的起始步采用 RST 或 ZRST 指令, 以清除其内容。

在使用 PC 间简易链接或并联链接的情况下, 一部分的数据寄存器被链接所占用的寄存器使用方法请参照可编程控制器的寄存器 D。

特殊用途

特殊用途的数据寄存器是指写入特定目的的数据, 或事先写入特定内容的数据寄存器。其内容在电源接通时被置于初始值, 一般清除为 0, 具有初始值的内容, 则利用系统 ROM 将其写入。

例如: 在 D8000 中, 监视定时器的时间通过系统 ROM 进行初始设定, 要将其改变时, 利用传送指令 F12 MOV, 在 D8000 中写入目标时间。



关于特殊数据寄存器的停电保持特殊请参照程序内存和参数的结构以及基本功能的补充事项。

其它有关特殊资料寄存器的种类和功能的说明, 请参照基本功能的补充事项。

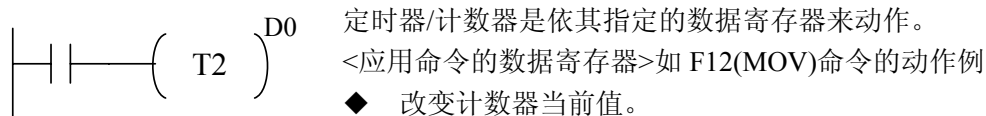
2. 各种软件组件的作用与功能

动作例

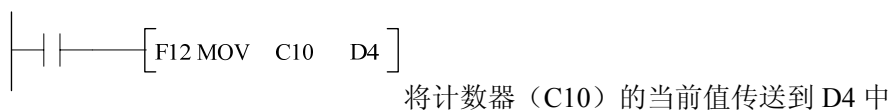
数据寄存器的数值数据可拿来作为控制。在本项说明其基本命令及应用命令的代表例，另可有效活用数据寄存器，请参考后述的应用命令。

<基本命令数据寄存器>

- ◆ 可指定定时器及计数器的设定值。



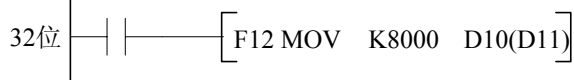
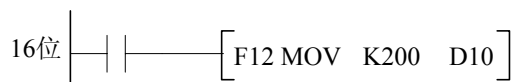
- ◆ 将定时器/计数器的当前值读到数据寄存器中。



- ◆ 将数值储存在数据寄存器。

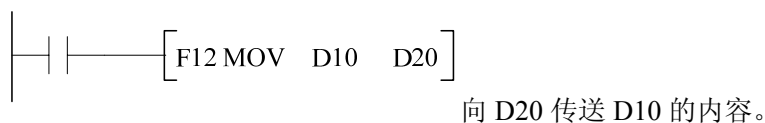


- ◆ 在数据寄存器中存储数据

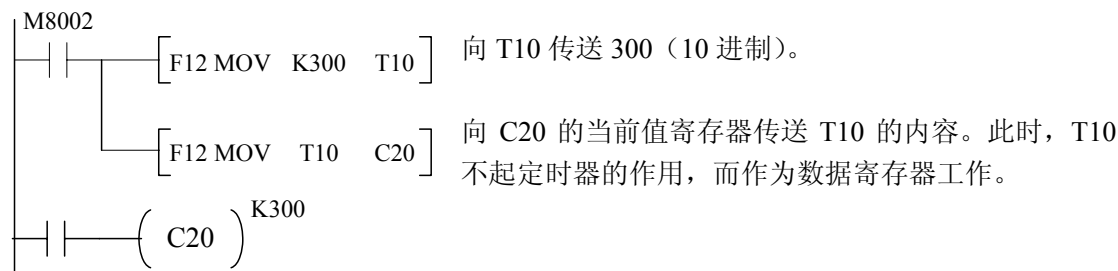


超过 32,767 的数值是 32 位的数值，因此采用双重 (D) 指令。如果数据寄存器指定为低位 (D10)，则高位 D (11) 自动被占用

- ◆ 数据寄存器的内容转送到其它数据寄存器。



<未使用的定时器和计数器当成数据寄存器>以 F12(MOV)命令为例



使用 32 位时，需要同样 16 位的数据寄存器(例 C0,C1)2 个为 32 位来表示，但亦可使用 32 位的计数器(例 C200 等)1 个为 32 位。

9.2 补充资料寄存器 W

补充资料寄存器的编号

补充资料寄存器 W 的编号如下表所示 (编号以 10 进制分配) :

BSP01 AR/T Type : 40/60 点

补充资料寄存器 W	W0~W9999	10000 点	※1
-----------	----------	---------	----

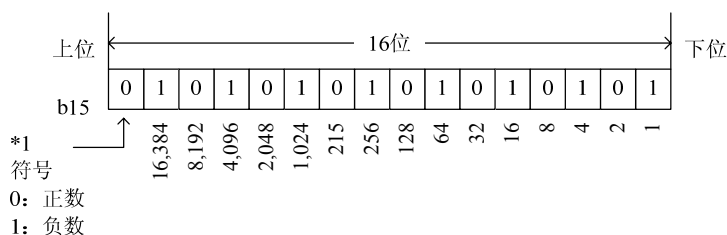
※1: 该寄存器上电后其值不确定, 故用户使用 W 之前需要对其赋值。另外所有的 W 寄存器均非掉电保持用。

寄存器的结构与功能

补充数据寄存器是存储数值数据的软组件, 该寄存器用于弥补 D 寄存器数量上的不足。这些寄存器都是 16 位 (最高位正负符号), 将两个数据寄存器组合, 可存储 32 位 (最高位为正负符号) 的数值数据。其种类如下所示。

16 位 (W)

一个补充数据寄存器 (16 位的数值范围为:) -32,768~+32,767。

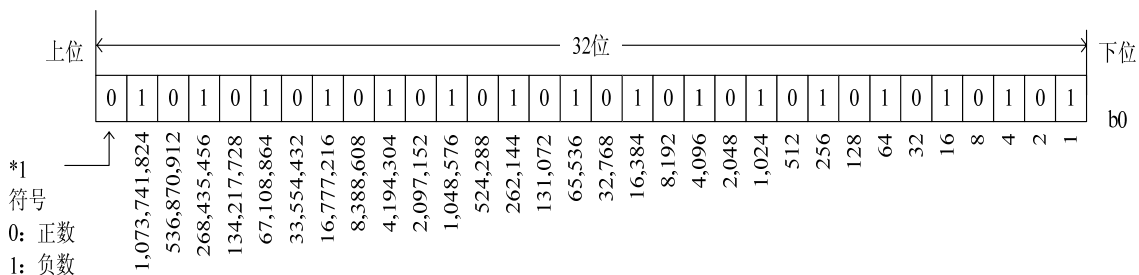


补充数据寄存器的数值读出与写入一般采用应用指令。此外, 也可从数据还取单元 (显示器) 与编程设备直接读出/写入。

32 位 (W)

以两个相邻的补充数据寄存器表现 32 位的数据。(高位为大的号码, 低位为小的号码, 在变址寄存器中, V 为高位, Z 为低位。)

因此可处理 -2,147,483,648~+2,147,483,647 的数值。



在指定 32 位时, 如果指定了低位(例:W0), 则高位为继其之后的编号(例如: W1)会自动占有。低位可用偶数或奇数的任意一种软元件编号指定, 考虑 BAPS-SP 监视功能, 建议指定低位采用偶数软元件编号。

使用方法

使用时只要注意以下两点: 第一、W 其值上电后不确定, 故上电后一定要对其初始化; 第二、所有 W 都不掉电保持。其它使用方法均与 D 寄存器一致。

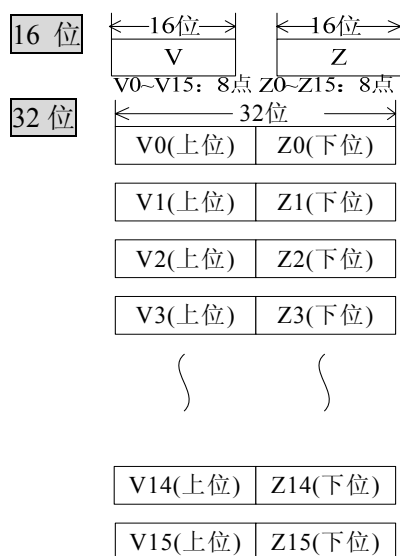
9.3 变址寄存器 V, Z

功能与结构

变址寄存器 V 与 Z 同普通的数据寄存器一样，是进行数值数据的读入，写出的 16 位数据寄存器。V0~V15, Z0~Z15 共 32 个。

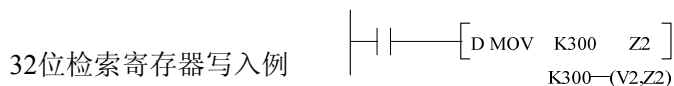
这种寄存器除了和普通的数据寄存器有相同的使用方法外，在应用指令的操作数中，还可以同其它的软组件编号或数值组合使用，可在程序中改变软组件编号或数值内容是一个的特殊寄存器。

此外，请注意 LD、AND、OUT 等可编程控制器的基本顺控指令，或步进梯形图指令的软组件编号能和变址寄存器组合使用。



V, Z 两种变址寄存器，同上述的数据寄存器同样的结构

在处理 32 位应用指令的软元件或处理超过 16 位范围的数值时，必须使用 Z0~Z15。这就如左图所示的 V, Z 的组合，TP03 可编程控制器将 Z 作为 32 位寄存器的低位侧进行动作。因此，即便指定了 V0~V15 的高位侧，也无法进行变址，此外，如果作为 32 位指定，由于 V（高位），Z（低位）被同时参照，如果在 V 高位侧，留下其他用途的数值，则会导致行数非常大的运算错误，即便在 32 位应用指令中使用的变址值没有超过 16 位数值范围，对于 Z 的数值写入也会出现如左图所示的情况，使用 DMOV 等 32 位指令，请同时改写 V（高位），Z（低位）32 位变址寄存器写入举例。



软组件的变址 可能变址的软组件，其变址内容如下所示：

10 进制数的软组件、数值：M, S, T, C, D, KnM, KnS, P, K.

例如：V0=K5，执行 D20V0 时，被执行的是软组件编号为 D25 (D20+5)。此外，也可变更常数值。例如：指定 K30V0 时，被执行的是作为 10 进制的数值 K35 (K30+5)。

8 进制数的软组件：X, Y, KnX, KnY。

例如：Z1=K8，执行 X0Z1 时，被执行的软组件编号为 X10 (X0+8 8 进制数加法)。对于软组件编号为 8 进制数的软组件变址来说，V, Z 的内容要被换算成 8 进制数，然后做加法运算，因此，假定 Z1=K10，X0Z1 被指定为 X12，请务必注意此数不是 X10。

16 进制数的数值：H

例如：以 V5=K30 指定常数 H30，则被认为是 H4E (30H+K30)。此外，以 V5=H30 指定常数 H30V5，则被认为是 H60 (30H+30H)。

变址示例和注意事项

关于应用指令中的操作数的变址方法和使用上的注意事项，请参照根据变址寄存器的操作数变址。

10 指标的编号与功能 P/I

指标的编号

指标[P],[I]编号如下表所示(编号以 10 进制分配)。使用输入中断用指标时，分配给中断的输入号码，不能同时使用于「高速计数」及「脉波密度(F56)」。

BSP01 AR/T 机种

分支用	中断输入用			时间中断用	计数中断用
P0~P255 256 点	输入	上升沿	下降沿	I6□□	I010 I040
	X000	I001	I000	I7□□	I020 I050
	X001	I101	I100	I8□□	I030 I060
	X002	I201	I200	3 点	6 点
	X003	I301	I300		
	X004	I401	I400		
	X005	I501	I500		
	6 点				

BSP01 SR 机种

分支用	中断输入用			时间中断用	计数中断用
P0~P127 128 点	输入	上升沿	下降沿	I6□□	I010 I040
	X000	I001	I000	I7□□	I020 I050
	X001	I101	I100	I8□□	I030 I060
	X002	I201	I200	3 点	6 点
	X003	I301	I300		
	4 点				

功能与动作例：

分支用指标和中断用指针的功能和动作如下：

几乎所有的指针都和应用命令组合使用，因此，详细的使用方法和说明请参考说明书。

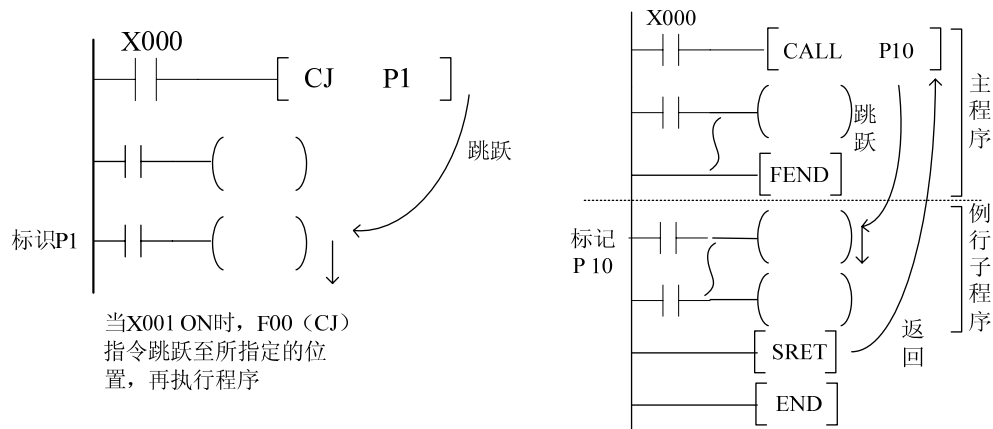
分支用指标和中断用指标的功能与动作如下：

分支用

1 F00 (CJ) 条件跳跃

2 F01 (CALL) 调用子程式

2. 各种软组件的作用与功能



中断用

中断用指令共有 3 个种类,应用命令 FNC03(IRET)中断跳回, FNC04(EI)允许中断, FNC05(DI)禁止中断的组合使用。

1. 输入中断用: 特定的输入编号, 不受到 PLC 扫描周期的影响, 当此信号读取时, 则实行中断子程序。输入中断时, 即使比扫描时间更短的信号也可以读取, 在 PLC 控制途中, 有必要优先处理的短时间脉波信号。
2. 时间中断用: 被指定的中断时间周期(10ms~99ms)时, 则中断子程序执行, PLC 的扫描时间外的固定时间的中断处理程序。
3. 计数中断用: 根据 PLC 内部之高速计数器比较结果, 来实行中断子程序, 根据高速计数器的计数结果来优先控制。

3. 基本顺控指令说明

第三章 基本顺控指令说明.....	38
1 基本指令一览表.....	38
2 [LD]/ [LDI]/ [OUT]/[OUT I]说明	40
3 AND, ANI 指令	41
4 OR, ORI 指令	42
5 LDP,LDF,ANDP,ANDF,ORP,ORF 指令	44
6 ORB 指令	46
7 ANB 指令	47
8 MPS,MRD,MPP 指令.....	48
9 MC,MCR 指令.....	52
10 INV 指令	53
11 PLS,PLF 指令	54
12 SET、RST 指令	55
13 计数器的输出复位(OUT, RST)指令	56
14 NOP,END 指令	57
15 SMCS, SMCR 指令	58
16 JCS, JCR 指令	60
17 编程的注意事项.....	62
17.1 程序的步骤与执行顺序.....	62
17.2 双重输出双线圈动作与其对策.....	63
17.3 不能编程的回路及其对策.....	65

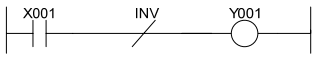




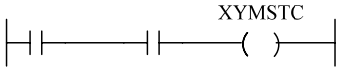
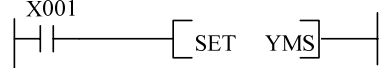
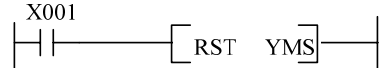


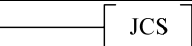

第三章 基本顺控指令说明

1 基本指令一览表

基本指令一览表

符号	功能	电路表示	步数
[LD]	运算开始常开接点		1
[LDI]	运算开始常闭接点		1
[AND]	串联常开接点		1
[ANI]	串联常闭接点		1
[OR]	并联常开接点		1
[ORI]	并联常闭接点		1
[LDP]	上升沿运算开始		2
[LDF]	下降沿运算开始		2
[ANDP]	上升沿检出串联连接		2
[ANDF]	下降沿检出串联连接		2
[ORP]	脉冲上升沿检出并联连接		2
[ORF]	脉冲下降沿检出并联连接		2
[ANB]	并联回路串联		1
[ORB]	串联回路并联		1
[MPS]	运算存储		1
[MRD]	存储读出		1

3. 基本顺控指令说明

[MPP]	存储读出并复归		1
[INV]	反向		1
[MC]	主控		3
[MCR]	主控复归		2
[NOP]	无动作		
[END]	程序扫描结束		
[STL]	步进梯形图编程		1
[RET]	步进梯形图编程结束		1
[PLS]	上升沿使能线圈		1
[PLF]	下降沿使能线圈		1
[P]	标识		——
[I]	中断标识		——
[OUT]	线圈		Y&M: 1 S&特殊 M: 2 T: 3 C: 3 (32位), 5(16位)
[SET]	置位线圈		Y, M: 1 S, 特殊 M: 2 T, C: 2
[RST]	复归线圈		D&V&Z&W 特殊 D: 3
[SMCS]	回路分支开始		1
[SMCR]	回路分支结束		1
[JCS]	跳跃分支开始		1
[JCR]	跳跃分支结束		1

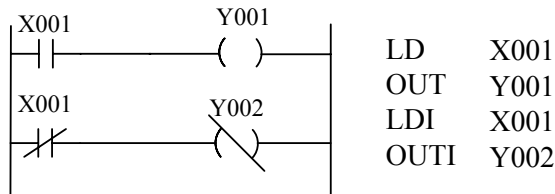
2 [LD]/ [LDI]/ [OUT]/[OUT I]说明

[LD]/ [LDI]/ [OUT]/[OUT I]

指令说明

- (1) [LD] (取) 常开接点与母线连接指令；可用于 X, Y, M, T, C 和 S。
- (2) [LDI] (取反) 常闭接点与母线连接指令；可用于 X, Y, M, T, C 和 S。
- (3) [OUT] (输出) 线圈驱动指令，用于将逻辑运算的结果驱动一个指定的线圈。如：输出接点、辅助继电器，步进点，计时/计数器输出线圈，但不能用于输入线圈 X，可用于 Y, M, T, C 和 S。
- (4) [OUT I] (取反输出)：线圈驱动指令[OUT]的反指令；只能用于 Y 和 M。

编程举例



计时器/计数器的输出

在对定时器、计数器使用 OUT 指令中可以用常数 K, H 指定预设值，也可以用数据寄存器 D 来间接指定预设值。

时间常数 K 的设定范围及与对应的时间实际设定值范围如下：

定时器计数器	K 值设定范围	实际的设定值	步序数
1ms	1~32,767	0.001~32.676 秒	3
10ms	1~32,767	0.01~327.67 秒	3
100ms		0.1~3,276.7 秒	
16 位计数器	1~32,767	同左	3
32 位计数器	-2,147,483,648~+2,147,483,647	同左	5

3 AND, ANI 指令

[AND]/[ANI] 指令

指令说明

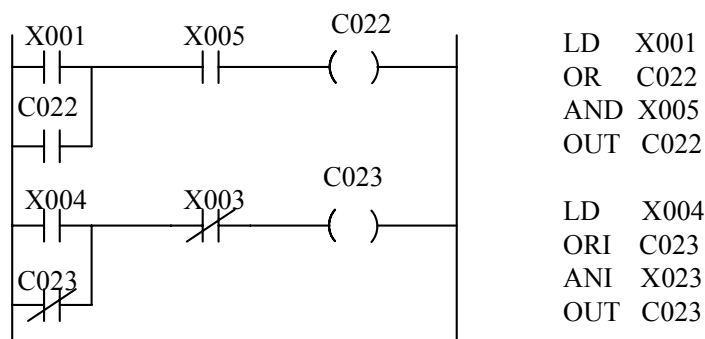
AND（与）常开接点串联指令；可用于 X, Y, M, T, C 和 S。

ANI（与非）常闭接点串联指令；可用于 X, Y, M, T, C 和 S。

1) AND 和 ANI 指令用于单个接点串联，串联接点的数量不限，重复使用次数不限；

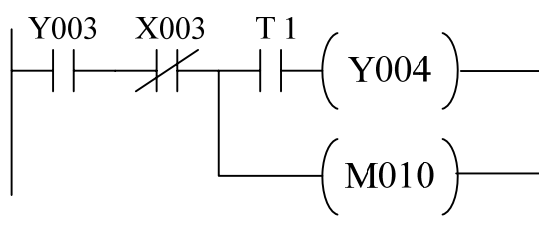
2) [AND]/[ANI]指令只能用于单个接点串联连接，若要将两个以上接点并联而成的电路块串联，要使用 ANB 指令，ANB 指令是并联电路块的串联指令，后面是不带软元件的。

编程举例



MPS, MPP 指令的关系

但如果梯形图程序为下图所示，就要使用到后面所述的 MPS 和 MPP 指令了。



4 OR, ORI 指令

[OR]/[ORI]

指令说明

[OR]为常开接点并联；可用于 X, Y, M, T, C 和 S。

[ORI]为常闭接点并联；可用于 X, Y, M, T, C 和 S。

当梯形图的控制线路由几个接点并联组成时，就要用到 OR 和 ORI 指令；

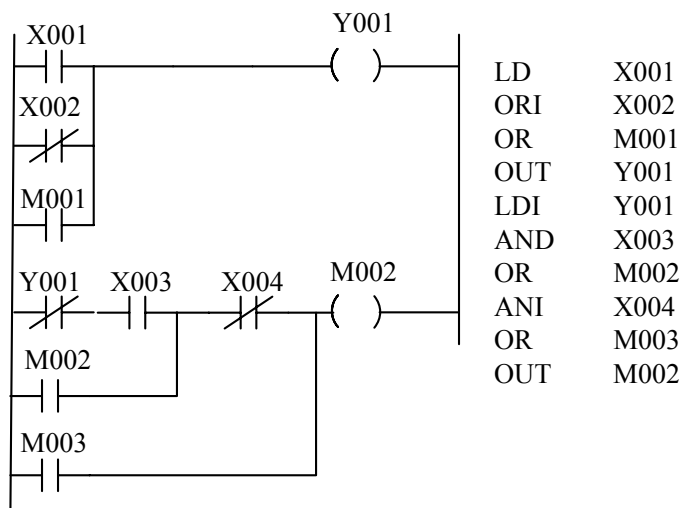
- 1) [OR]/[ORI]指令与前面的[LD]/[LDI]指令并联使用，次数不受限制。
- 2) [OR]/[ORI]指令只能用于单个接点并联连接，若要将两个以上接点串联而成的电路块并联，要使用 ORB 指令，ORB 指令是串联电路块的并联指令，后面是不带软元件的。

串联电路块的并联指令 ORB

ORB（串联电路块与）：将两个或两个以上串联电路块并联连接的指令；

两个以上接点串联的电路块，串联电路块并联连接时，在支路始端用 LD 和 LDI 指令，在支路终端用 ORB 指令。

编程

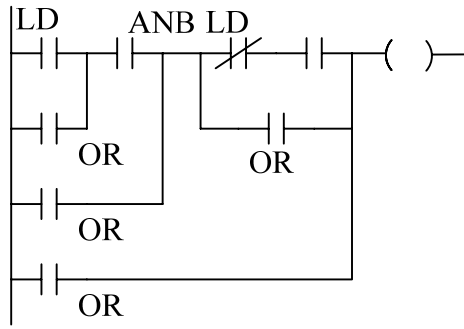


并联电路块的串联指令 ANB

ANB（并联电路块与）：将并联电路块的始端与前一个电路串联连接的指令；

两个以上接点并联的电路称作并联电路块，并联电路块串联连接时要用 ANB 指令。

3. 基本顺控指令说明



5 LDP,LDF,ANDP,ANDF,ORP,ORF 指令

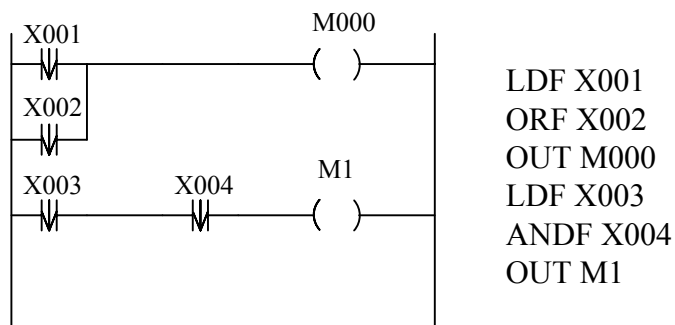
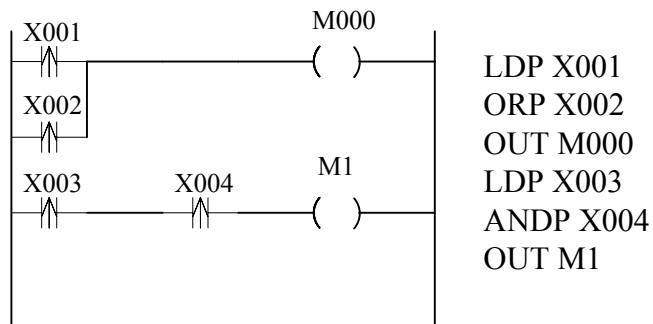
[LDP]/ [LDF]/ [ANDP]/ [ANDF]/ [ORP]/ [ORF]指令

指令说明

[LDP]/ [ANDP]/ [ORP]指令是指此命令操作的元件会在上升沿触发（OFF→ON）时导通一个扫描周期。

[LDF]/ [ANDF]/ [ORF]指令是指此命令操作的元件会在下降沿触发（ON→OFF）时导通一个扫描周期。

编程



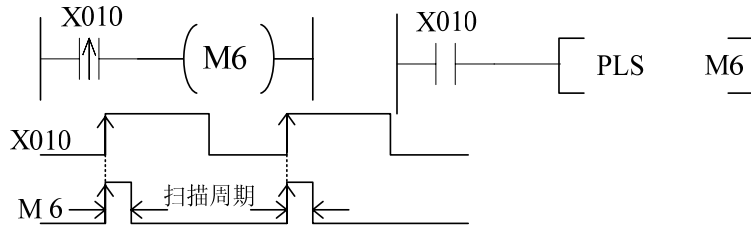
在上图中，X001~X004 由 ON~OFF 时或由 OFF—ON 变化时，M0 或 M1 仅导通一个扫描周期。

3. 基本顺控指令说明

输出线圈的实际驱动状况说明:

- 下面的电路具有相同的驱动效果。

OUT 指令及脉冲指令



两种情况下，当 X010 由 OFF—ON 变化时，M6 只导通 1 个扫描周期。

上升沿检出及应用指令的脉冲执行形式



当 X020 从 OFF—ON 变化时，D0 里的数据被传送了一次，两条程序驱动效果相同

MOV 指令前面条件逻辑结果为 ON 数据连续传送,前面条件逻辑结果为 OFF 状态时停止传送.

MOV P 指令前面条件逻辑结果为 OFF 到 ON 状态变化时传送数据，有几次这样的变化就传送几次数据。

6 ORB 指令

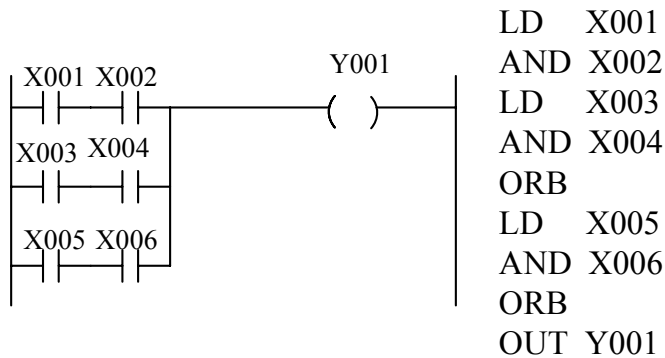
[ORB]指令

指令说明

ORB（串联电路块或）：是将两个或连个以上的串联电路进行并联连接的指令。
两个以上接点串联的电路称作串联电路块，串联电路块并联连接时，在支路始端用[LD]/ [LDI]指令，在支路终端时使用[ORB]指令

- 1) [ORB]和[ANB]一样，是一个单独的指令，不带软元件，后面不跟任何元件编号。
- 2) 多重并联电路中，如每个串联电路块都用[ORB]指令，则并联电路次数不受限制。[ORB]指令也可连续使用，此时在一条母线上[LD]/ [LDI]指令重复使用的次数要少于8次。

编程



7 ANB 指令

[ANB]指令

指令说明

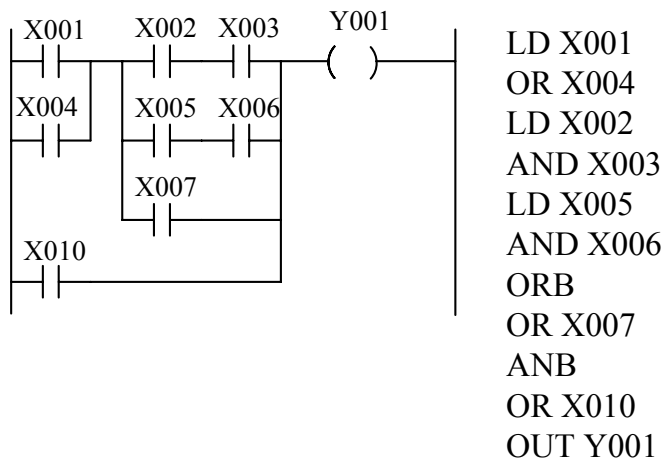
ANB（并联电路块与）：将并联电路块的始端与前一个电路串联连接的指令；

1) 两个以上接点并联的电路称作并联电路块，并联电路块串联连接时要用 ANB 指令；在与前一个电路串联的时候，用 LD 与 LDI 指令作为分支电路的始端，分支电路的并联电路块完成以后，用 ANB 指令来完成两电路的串联。

2) ANB 指令不带软元件，后面不跟任何元件编号；

多个电路并联时，如每个并联块都用 ANB 指令顺次串联，则并联电路数不受限制。ANB 指令可以集中起来使用，但在同一条母线上 LD 和 LDI 指令重复使用必须少于 8 次。

编程



8 MPS,MRD,MPP 指令

[MPS]/ [MRD]/ [MPP]指令

指令说明

- (1) MPS (PUSH): 进栈指令。
- (2) MRD (READ): 读栈指令。
- (3) MPP (POP): 出栈指令。

这组指令可将接点的状态先进栈保护, 当后面需要接点的状态时, 再出栈恢复, 以保证与后面的电路正确连接。

1) 本 PLC 中, 有 8 个可存储中间运算结果的存储器, 它们相当于微机中的堆栈, 是按照先进后出的原则进行存取一段存储器区域。

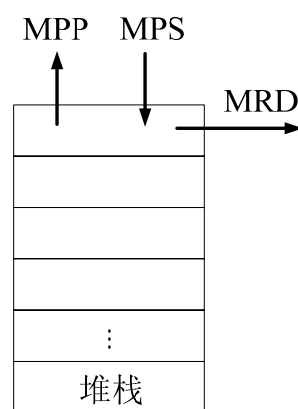
2) 使用一次 MPS 指令, 该时刻的运算结果就压入栈底第一个栈单元中 (称之为栈顶)。再次使用 MPS 指令时, 当时的运算结果压入栈顶, 而原先压入的数据一次向栈的下一个栈单元推移。

3) 使用 MPP 指令, 各数据依次向上一个栈单元传送。栈顶数据在弹出后就从栈内消失。

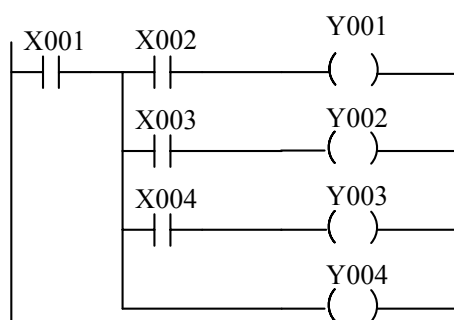
4) MRD 是栈顶数据的读出专用指令, 但栈内的数据不发生下压或上托的传送。

5) MPS、MRD、MPP 指令均不带软元件, 后面不跟任何元件编号。

6) MPS 和 MPP 应该配对使用, 连续使用的次数应少于 8 次。



编程

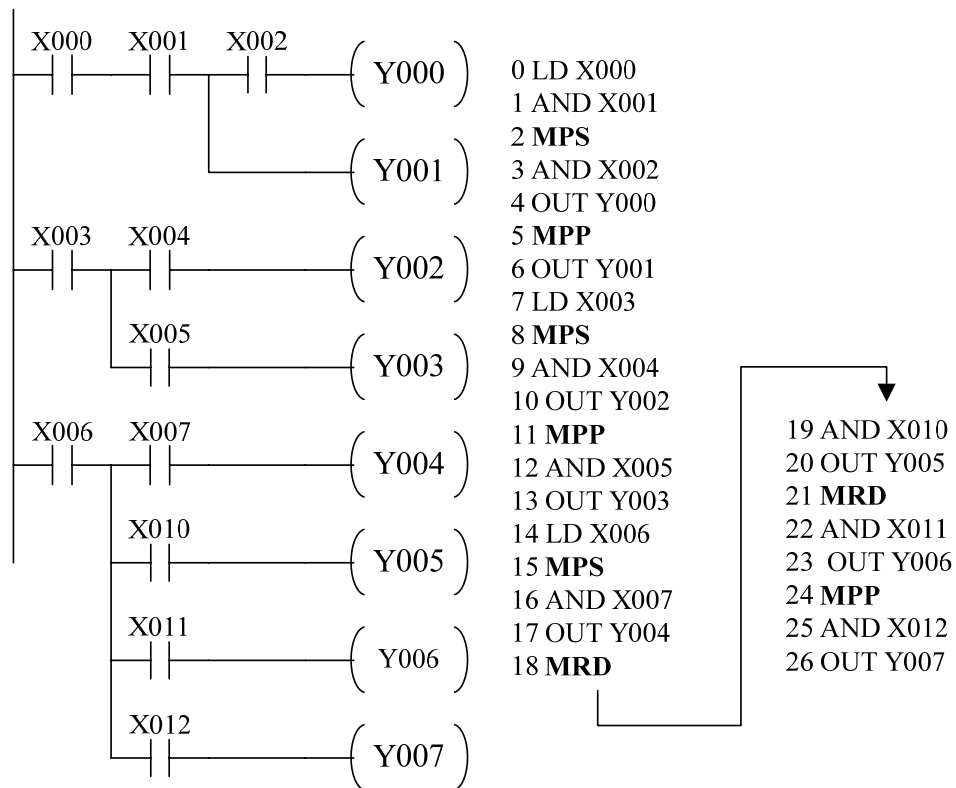


```
LD X001
MPS
AND X002
OUT Y001
MRD
AND X003
OUT Y002
MRD
AND X004
OUT Y003
MPP
OUT Y004
END
```

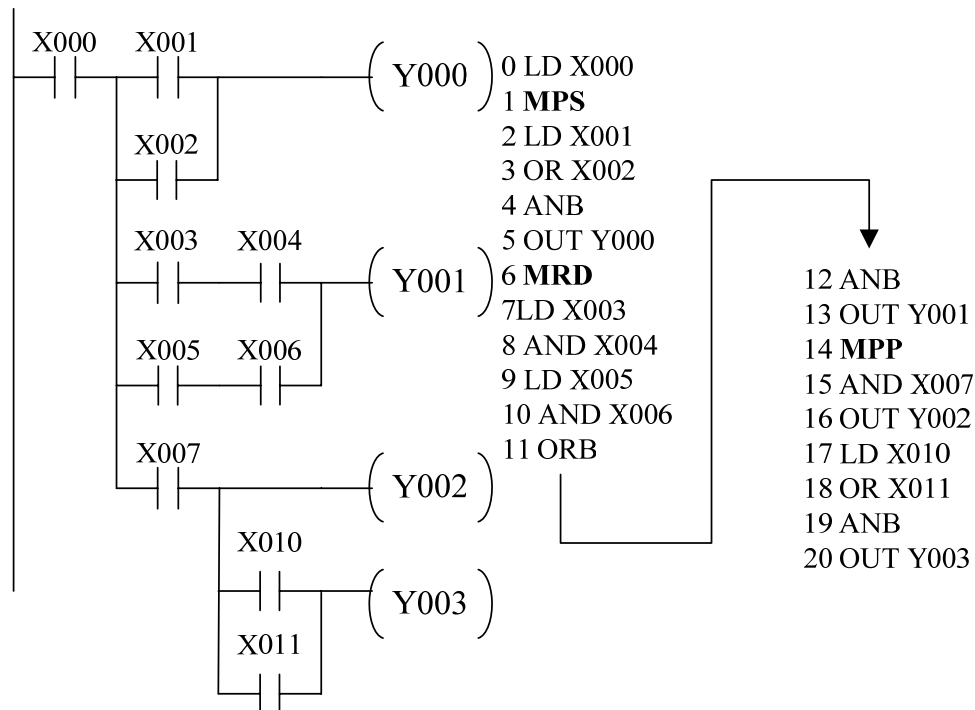

3. 基本顺控指令说明

编程示例:

一段堆栈

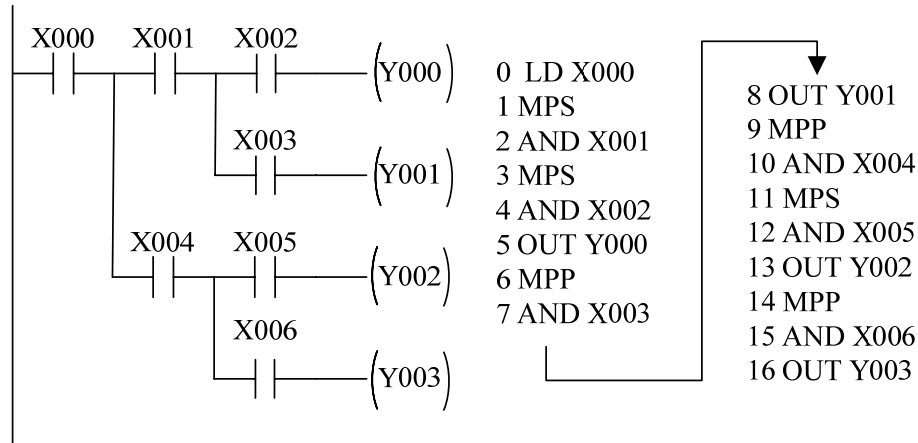


一段堆栈, ANB, ORB 指令并用

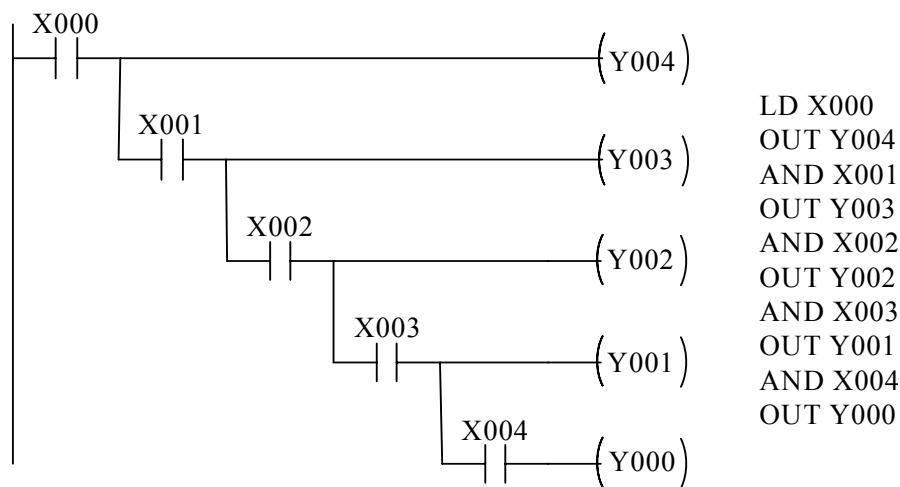
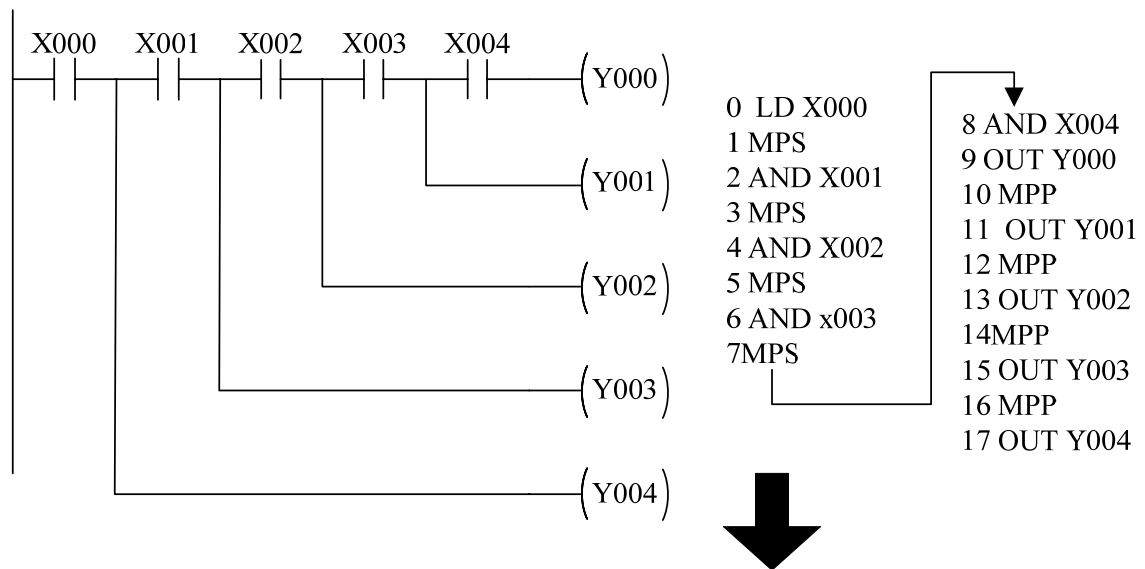


3. 基本顺控指令说明

二段堆栈



四段堆栈



上图所示的回路需采用三重 MPS 指令编程。

3. 基本顺控指令说明

但是，如果改用下面的回路，则不必办用 MPS 指令，编程也很方便。

9 MC,MCR 指令

[MC]/[MCR] 指令

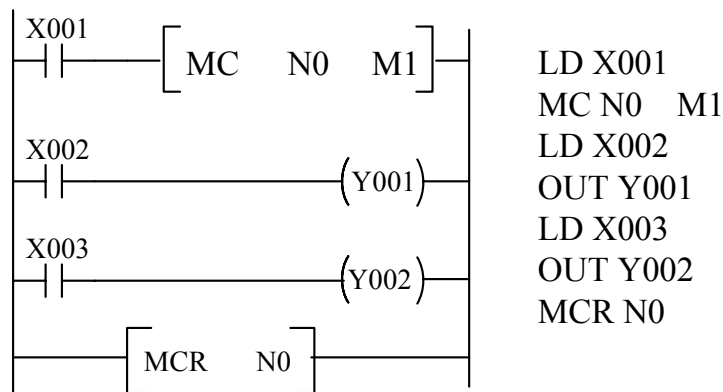
- (1) MC (主控): 公共串联接点的连接指令 (公共串联接点另起新母线)
- (2) MCR (主控复位): MC 指令的复位指令。

这两个指令分别设置主控电路块的起点和终点

指令说明

- 1) 在下图中, 当输入 X001 接通时, 执行[MC]与[MCR]之间的指令, 当 X01 断开时, MC 与 MCR 指令间各组件将为如下状态: 累计定时器、计数器, 用 SET/RST 指令驱动的元件将保持当前的状态; 非累积定时器及用 OUT 指令驱动的元件, 将处于断开的状态。
- 2) 执行[MC]指令后, 母线将 (LD、LDI) 移至 MC 接点, 要返回原母线, 用返回指令 MCR。MC/MCR 指令必须成对使用。
- 3) 使用不同的 Y、M 元件号, 可多次使用 MC 指令。但是若使用同一元件号, 将同 OUT 指令一样, 会出现双线圈输出。
- 4) MC 指令可嵌套使用, 即在 MC 指令内再使用 MC 指令, 此时嵌套级的编号就顺次由小增大。用 MCR 指令逐级返回时, 嵌套级的编号则顺次由大减小。

编程



当在[MC]指令中使用[MC]时, 主控点的编号顺次要由小到大。
 (N0→N1→N2→N3→N4→N5→N6→N7)。在将该指令返回时, 采用 MCR 指令, 则从大的嵌套级开始消除。(N7→N6→N5→N4→N3→N2→N1→N0)
 嵌套级最大可编写 8 级 (N7)

10 INV 指令

[INV]指令

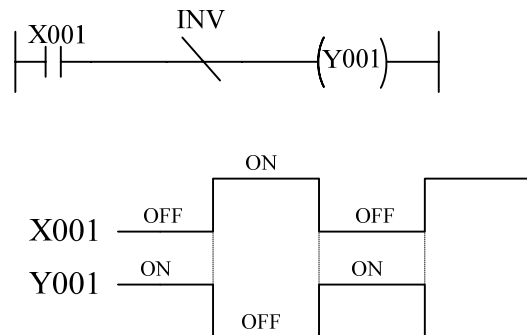
取反指令 INV 在梯形图中用一条 45° 的短斜线表示，将该指令之前的运算结果进行取反；

指令说明

取反指令 INV 无软元件，无须指定元件编号，其在使用程序内的动作状况如下所示。

执行 INV 指令前的运算结果	执行 INV 指令后的运算结果
OFF→	ON
ON→	OFF

编程



如图所示，当输入继电器 X001 断开时，输出线圈 Y001 导通，当 X001 导通时 Y001 断开。在能输入 AND、ANI、ANDP、ANDF 指令步的相同位置处，可编写取反指令 INV。INV 指令不能象指令表中的 LD、LDI、LDP、LDF 那样与母线连接，也不能象 OR、ORI、ORP、ORF 指令那样单独使用。

11 PLS,PLF 指令

[PLS]/[PLF]指令

- (1) PLS: 微分输出指令, 上升沿有效;
- (2) PLF: 微分输出指令, 下降沿有效;

这两个指令用于目标元件的脉冲输出, 当输入信号跳变时产生一个宽度为扫描周期的脉冲。

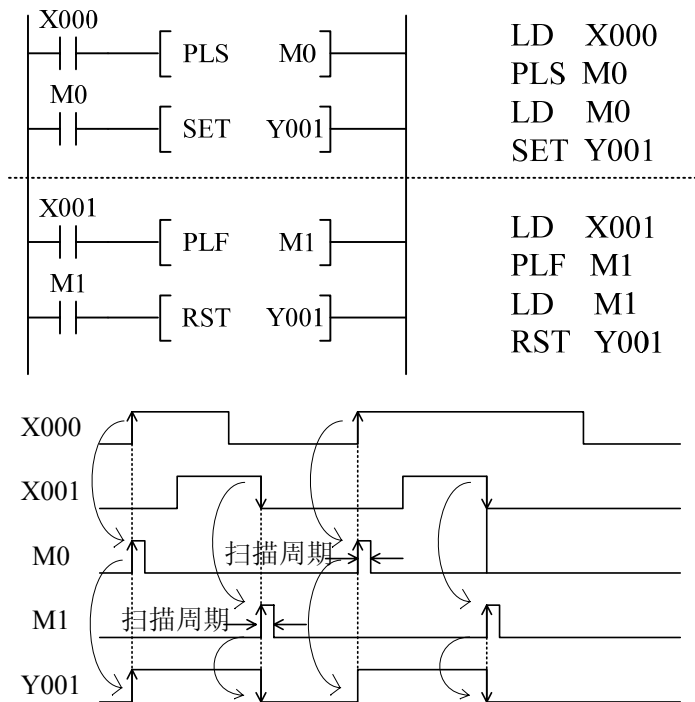
指令说明

[PLS]指令使用时, 驱动输入点 ON 后, 被驱动的组件只有一个扫描时间。

[PLF]指令使用时, 驱动输入点 OFF 后, 被驱动的组件 Y、M 只动作一个扫描时间。

例如输入点 X000 和 X001 依下图的时序接通, PLC 由运行→停止→运行时, 分析程序的时序逻辑关系可知, 当 X000 导通的上升沿时, M0 线圈得电并保持一个扫描周期, M0 常开闭合使 Y001 得电置 1, 而到 X001 接通的下降沿时, M1 线圈得电并保持一个扫描周期, M1 常开闭合使 Y001 复位置 0。

编程



12 SET、RST 指令

[SET]/[RST]指令

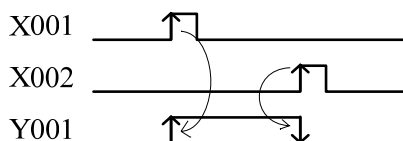
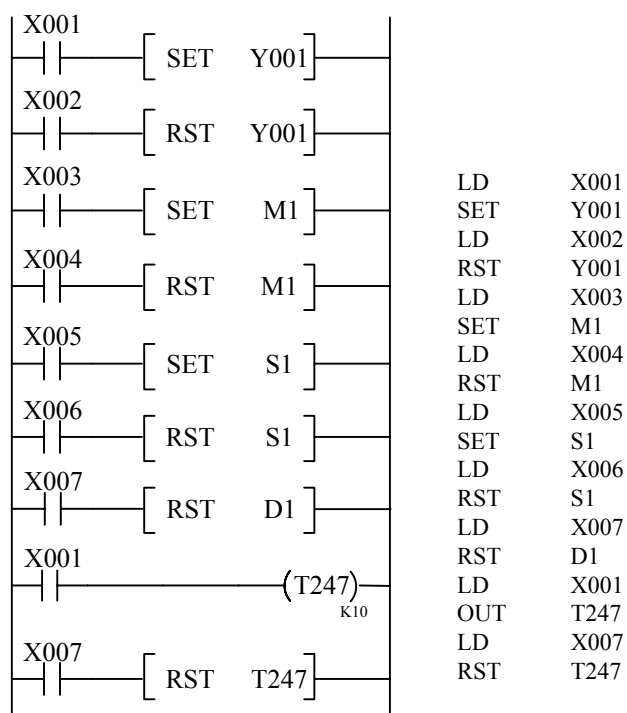
- (1) SET (置位): 置位指令, 使线圈得电保持
- (2) RST (复位): 复位指令, 使线圈失电保持

在应用程序中使用 SET 和 RST 指令, 可以方便地在用户程序的任何地方对某个状态或时间设置标志和清除标志。

指令说明

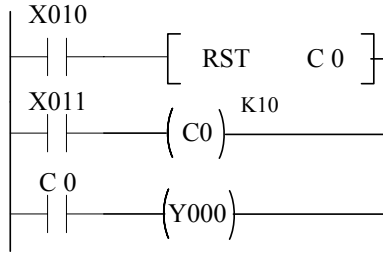
- 1) SET 和 RST 指令具有自保持功能, 下图所示的程序中, 当 X001 接通, 即使再断开, Y001 则保持接通, X002 一旦接通, 即使再断开, 则 Y001 也保持断开。
- 2) SET 和 RST 指令的使用没有顺序限制, 并且 SET 和 RST 之间可以插入别的程序, 但只在最后执行的一条才有效。
- 3) SET 和 RST 指令的目标元件除了 Y、M、S 之外, 还有 T、C、D, W 即可对数据寄存器 D、补充数据寄存器和变址寄存器内的值可进行清零操作, 以及对定时器 T 和计数器 C 进行复位, 使当前计时器和计数值清零。

编程



13 计数器的输出复位(OUT, RST)指令

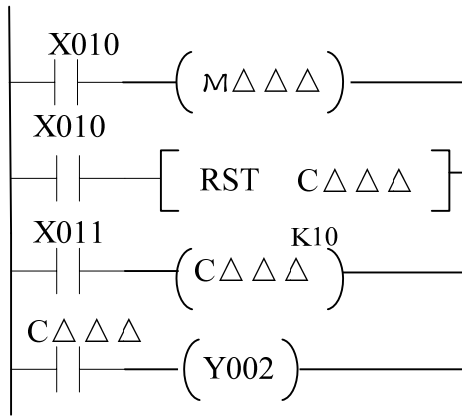
编程



上述程序的逻辑关系说明:

- 1) 当输入点由 X011 从 OFF—ON 时, 计数器 C0 开始增计数, 当计数值达到设定值 K10 时, 输出触点 C0 动作, 输出线圈 Y000 接通; 当 X011 再由 OFF—ON 时, 计数器当前值不变, 输出线圈 Y000 仍处于接通状态;
- 2) OUT C 命令后要指定计数的常数 K, 或者使用数据寄存器间接指定。只有当另一输入点 X010 ON 时, 计数器 C0 会被重新置为 0, 输出接点 Y000 复归。

高速计数器编程



- 使用 C235~C245 的单相单输入计数器, 需使用特殊辅助继电器 M8235~M8245 来指定其计数方向。X010: ON 时为减计数, X010: OFF 时为增计数。
- X010 ON 时, 计数器 C 口口口输出触点复归, 计数器当前值归 0;
- 如果程序中使用带有复位输入功能的计数器(C241,C242,.....), 当相应的复位输入为 ON 时, 通过中断动作, 可以产生与上述指令相同的效果, 可不必为此编程。
- X011 ON 时, 对由计数器编号决定的计数输入 X000~X005 的 ON/OFF 进行计数。附有开始输入功能的计数器(C244,C245...), 如果其相应的开始输入点不为 ON 时, 则不能进行计数。
- 计数器的当前值增加, 在达到设定值时 (K 常数、W 内容或 D 内容) 时, 输出触点则被 SET, 低于现在值时, 则被 RST。

14 NOP,END 指令

[NOP]/ [END]指令

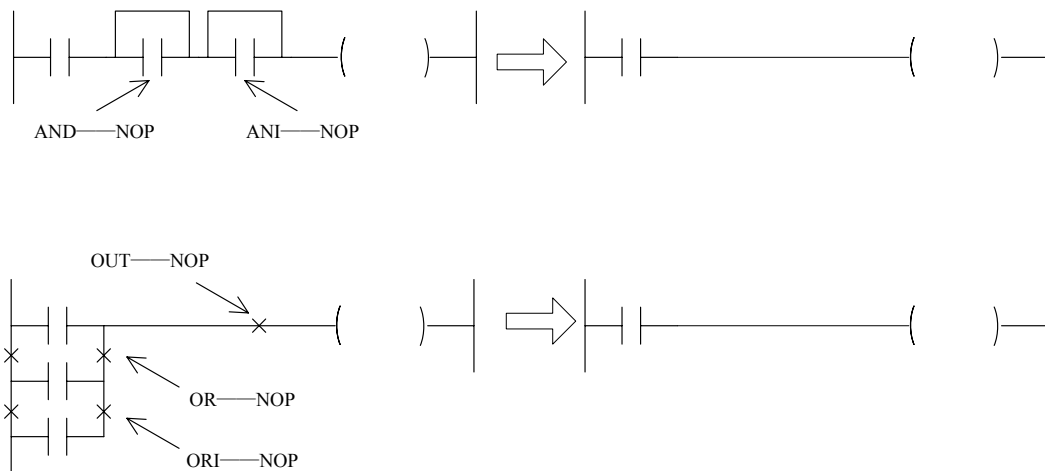
- (1) NOP (空操作): 空一条指令 (或用于删除一条指令);
- (2) END (结束): 程序结束指令。

在程序的调试过程中, 恰当地使用 NOP 和 END 指令, 会带给用户诸多便利。

指令说明

- 1) NOP 是空操作指令, CPU 不执行目标指令。NOP 指令在程序中占一个步序, 该指令在梯形图中没有对应的元件来表示它, 但是可以从梯形图中的步序得到反映
- 2) 执行指令程序全清操作后, 全部指令都变成 NOP
- 3) 在程序中可通过实现插入 NOP 指令, 以备在修改或增加指令时, 可使步进编号的更改次数减少到最少。

在已完成程序, 插入 NOP 指令时, 则程序会变化, 需注意:



- 4) END 指令用于程序的结束, 无目标软元件。END 指令还可以在程序调试中设置断点, 可以先分段插入 END 指令, 再逐段调试, 调试好后, 删去 END 指令

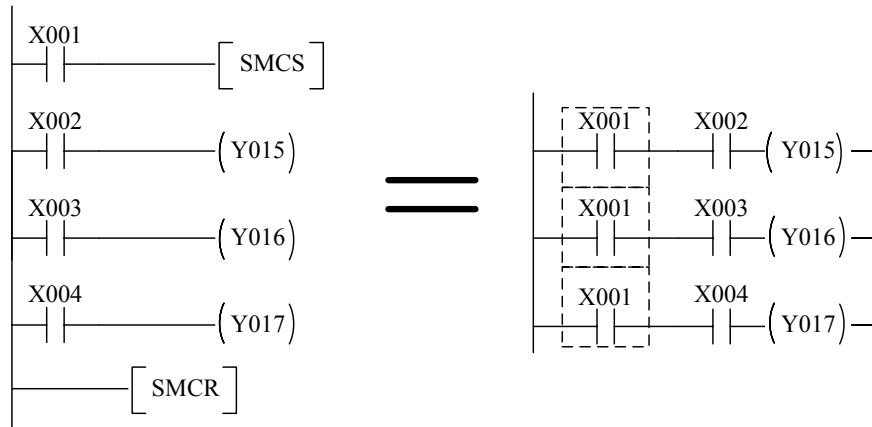
15 SMCS, SMCR 指令

[SMCS], [SMCR]指令

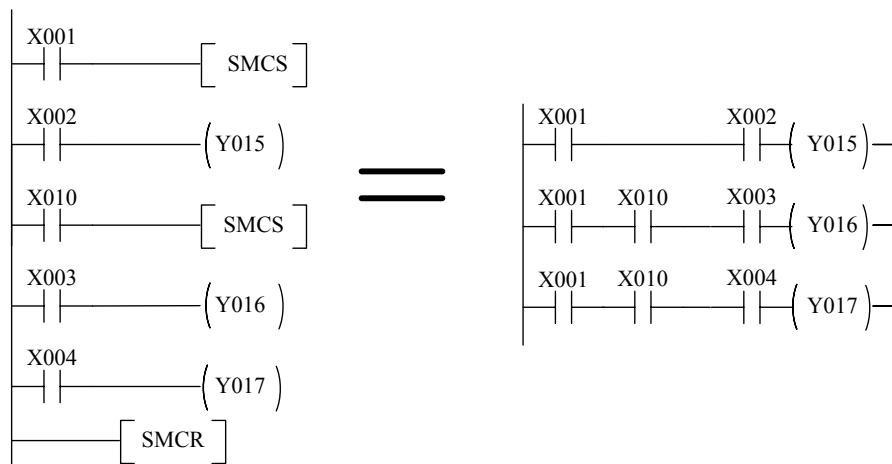
- (1) [SMCS] 回路分支开始：相当于一条件母线，当该指令前的条件 ON 后，条件母线导通。
- (2) [SMCR] 回路分支开结束：条件母线结束。

在程序中[SMCS], [SMCR]必须成对使用。[SMCS]在程序中可连续或不连续的使用多次，主要用于某段线路需要在其他多处线路中重复出现时，这样可简化程序。

例：



[SMCS]指令可以在程序中使用多次，如下图所示。每使用一次，则增加一副母线的条件。多次使用[SMCS]指令后，仅一个[SMCR]指令便会全部清除所有条件。



[SMCS]之后[SMCR]之前的每个指令都会与[SMCS] 之前的那个条件于 CPU 内部作运算。当共同线路相当复杂，或出现多次时，利用该指令可适当简化程序。

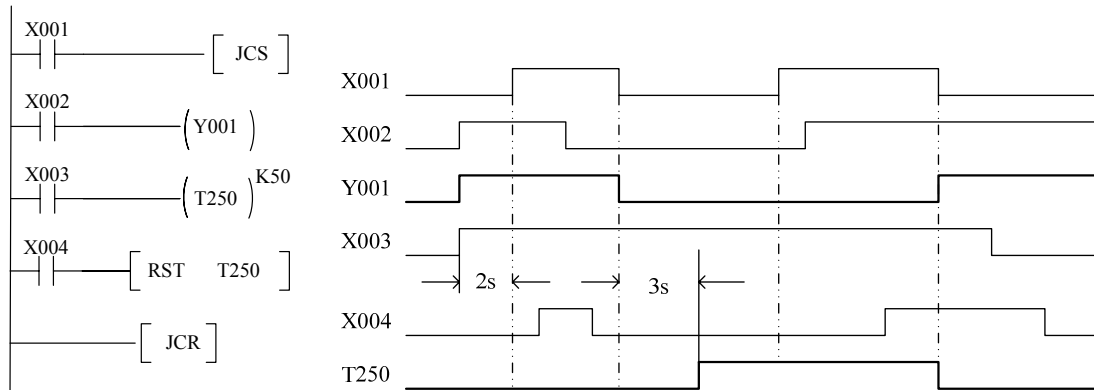
注：[SMCS]之后不可直接接 OUT, TMR, CNT 与应用指令。

16 JCS, JCR 指令

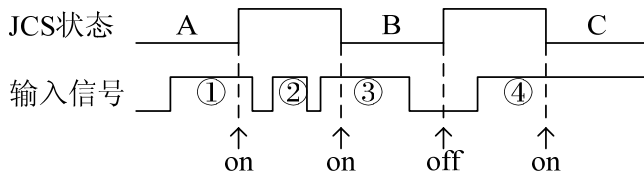
[JCS], [JCR]指令

- (1) [JCS] 跳跃分支开始
- (2) [JCR] 跳跃分支结束

[JCS]之后, [JCR] 之前的所有指令都不会执行, 亦即在[JCS]条件输入为 ON 期间, 寄存器的内容保持不变。在[JCS]和 [JCR] 之间不应有[END]指令, 否则程序出错, EER 指令灯亮。



注 1: 要特别注意是计时器的计时信号, 以及计数器, 应用指令的输入信号 (OFF→ON 的变化) 与[JCS] ON/OFF 状态的相对时间关系。



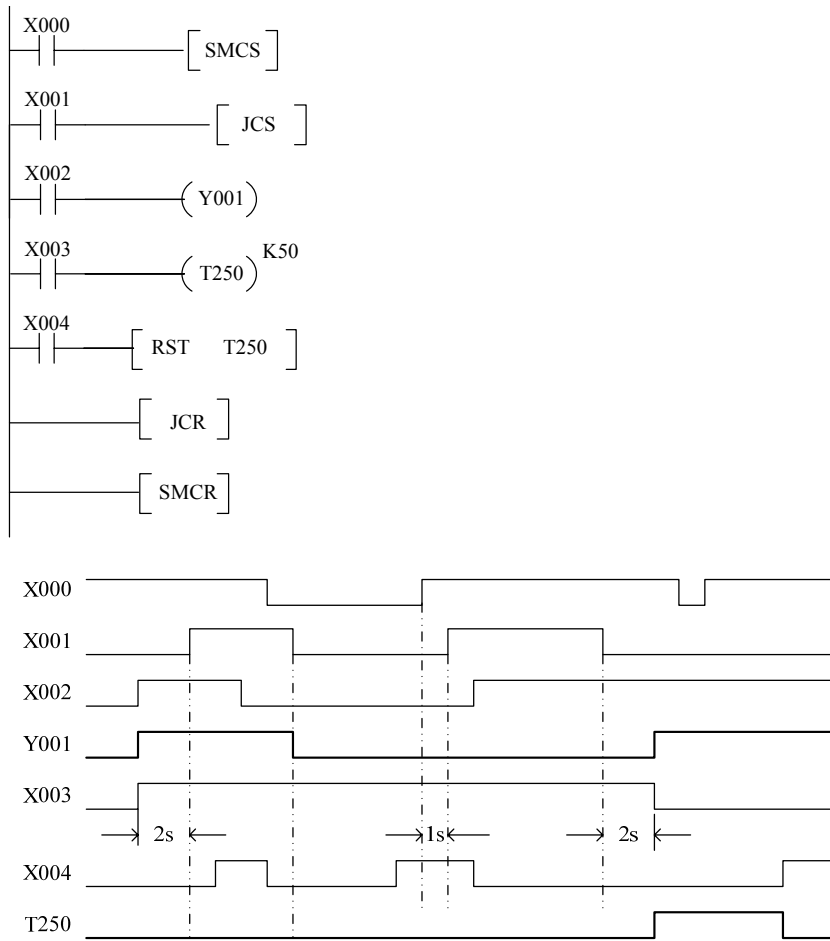
- ◆ X002 处于①中的 OFF→ON Y001 会动作, 因为[JCS]状态为 OFF。
- ◆ X002 处于②中的 OFF→ON Y001 不动作, 因为[JCS]状态为 ON。
- ◆ ③中的 OFF→ON 不动作, 因为[JCS]状态为 ON。
- ◆ [JCS]的状态在 X002 处于③ON 期间变成 OFF 时, Y001 仍然不动作; 因为[JCS]状态(A)中 Y001 会随输入信号由 OFF 变 ON 时, 在[JCS] ON 期间不会受输入信号 ON→OFF 或 OFF→ON 状态改变的影响, 当[JCS] OFF (状态 B 时) 后输入信号发生 OFF→ON 的情况, Y001 会由 ON→OFF
- ◆ ④中的输入信号 OFF→ON 时, Y001 仍为 OFF 不动作, 因为[JCS]状态此时为 ON 状态。
- ◆ [JCS]状态在④ON 期间变成 OFF, Y001 动作。

注 2: [JCS]状态 ON 时, [JCS], [JCR]之间会影响标志位的指令并不会不执行。

注 3: 在[JCS], [JCR]之间的[END]指令一定会执行, 不受[JCS]处于 ON 或 OFF 状态的影响。此时程序即中止执行而进行下一个扫描周期。

注 4: [JCS]和[JCR]之间可插入[SMCS]与[SMCR]之间。

3. 基本顺控指令说明



注 6: 要[JCS], [JCR]之间可插入另一个[JCS], 而仅使用一个[JCR]作为状态结束。

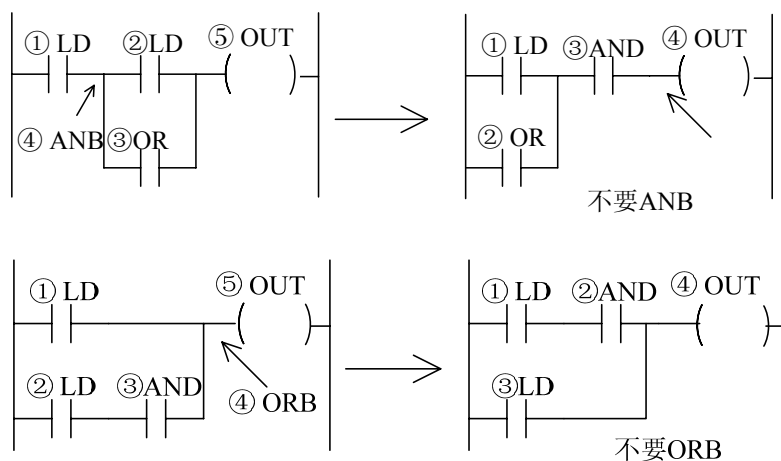
17 编程的注意事项

17.1 程序的步骤与执行顺序

- 1) 梯形图中的阶梯都是始于左母线，终于右母线。每行的左边是接点的组合，表示驱动逻辑线圈的条件，而表示结果的逻辑线圈只能接在右边的母线上，接点是不能出现在线圈的右边的。
- 2) 接点应画在水平线上，不要画在垂线上。
- 3) 并联串联时，应将接点多的支路放在梯形图的左方。串联块并联时，应将接点多的并联支路放在梯形图的上方。
- 4) 不宜使用双线圈输出。

接点的结构和步序

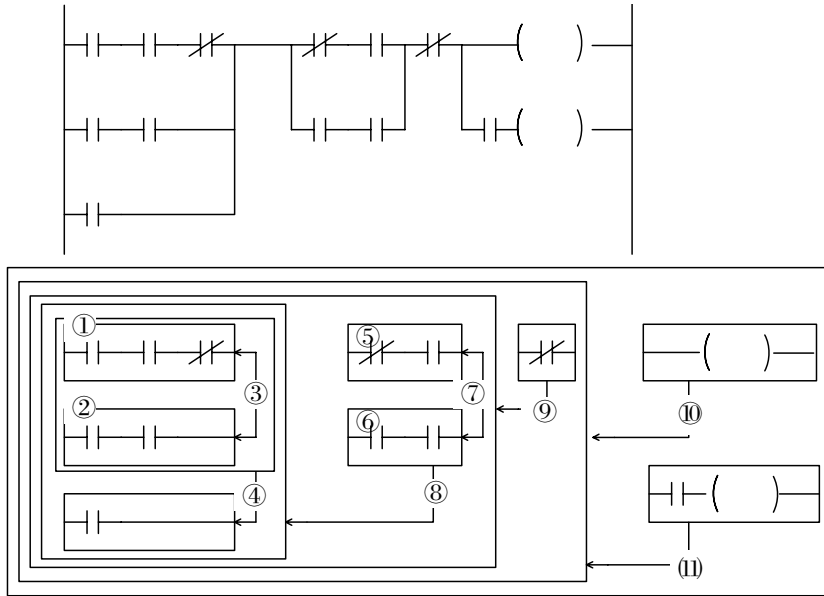
同一程序的电路，根据接点的构成方式，可使程序单纯化和节省程序容量。



程序的执行顺序

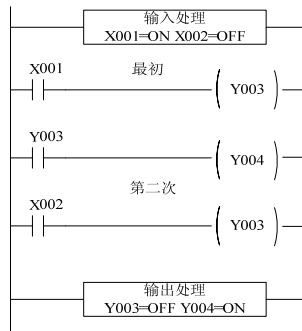
程序是由上而下，由左至右来处理。程序指令的流程是由下面方块图来执行。

3. 基本顺控指令说明



17.2 双重输出双线圈动作与其对策

若在顺控程序中进行线圈的双重输出（双线圈），则在后面的动作优先执行。

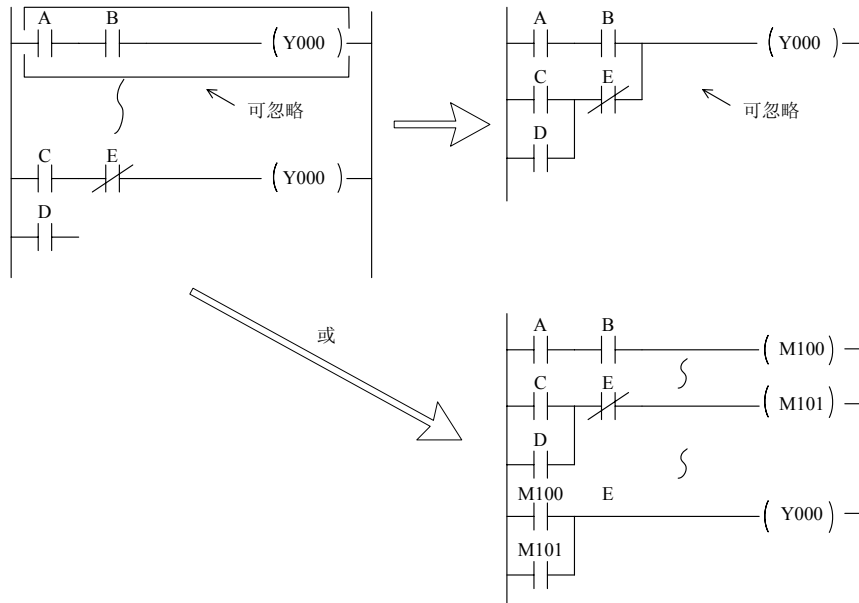


如左图所示，请注意在多处使用同一线圈 Y003 的情况。
 例如：X001=ON，X002=OFF 时，
 起初 Y003，因 X001 接通，因此其映像存储区变为 ON，输出 Y004 接通。
 但第二次的 Y003 因 X002 断开，因此其映像存储区内为 OFF。
 因此，实际外部输出为 Y003=OFF，Y004=ON。

双重输出的对策

双重输出在程序上并不违反规则，但上述的动作十分复杂，因此请按以下的示例程序。

3. 基本顺控指令说明

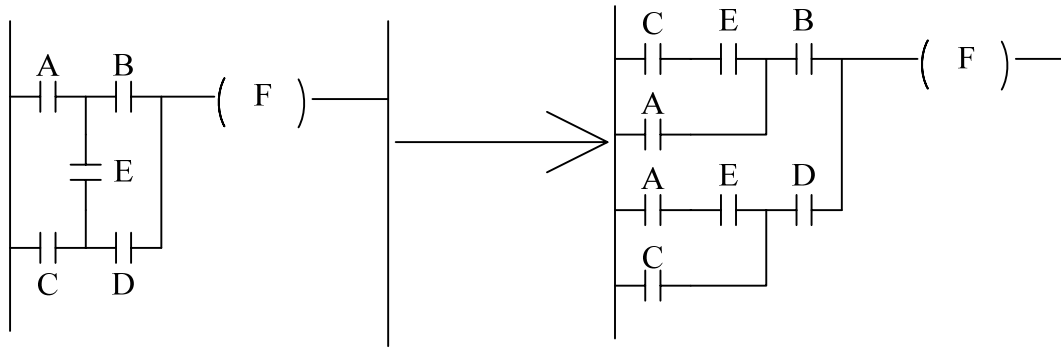


也可以使用跳跃指令和步进梯形图指令，将上述程序改成为同一输出效果。当使用步进梯形图指令时，在主程序中与状态程序中，请注意双重输出和同一输出的使用。

17.3 不能编程的回路及其对策

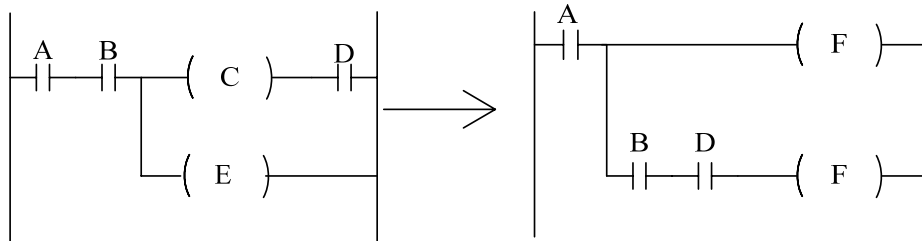
桥式电路

如图示：改变双向回路共流动的方向。(将没有 D 和没有 B 的回路并联连接。)



线圈的接续位置

- 请不要在线圈的右侧写触点。
- 建议触点间的线圈先编程。



第四章 步进梯形图指令.....	67
1 BAPS-SP SFC 编程简介.....	68
2 步进指令 STL, RET.....	70
3 步进梯形图指令的动作与 SFC 表示.....	73
4 SFC 的特点.....	75
5 编程 SFC 流程的预备知识.....	77
6 SFC 流程的形态.....	81
6.1 跳转与重复流程.....	81
6.2 分支与汇合的组合流程.....	82
7 初始状态的作用.....	83
8 中间状态的程序.....	84
8.1 没有分支与汇流的一般流程.....	84
8.2 带有跳转与重复的一般状态.....	85
9 分支与汇合状态的程序.....	86
9.1 选择性分支与汇合状态.....	86
9.2 并行分支与汇合状态.....	87
9.3 分支与汇合的组合.....	88
10 单流程示例.....	89
11 选择性分支与汇合流程示例.....	93
12 并行分支与汇合流程示例.....	95
13 初始化状态(F60 IST)命令的活用.....	96

第四章 步进梯形图指令

PLC 的程序设计语言——顺序功能图 (SFC)

顺序功能图 (Sequential Function Chart) 是一种描述顺序控制系统功能的图解表示法。对于复杂的顺控系统,内部的互锁关系非常复杂,若用梯形图来编写,其程序步就会很长,可读性也会大大降低。SFC 语言以顺序功能图形式表示机械动作,状态转移方式编程,特别适用于编制复杂的顺控程序。

用 SFC 语言来编制复杂的顺控的编程思路:

(1) 按结构划程序设计的要求,将一个复杂的控制过程分解为若干个工作步,这些工作步成为状态。状态与状态之间由转移分隔。相邻的状态具有不同的动作。当相邻两状态之间的转移条件得到满足时,就实现转移,即上一状态的动作结束而下一状态的动作开始。

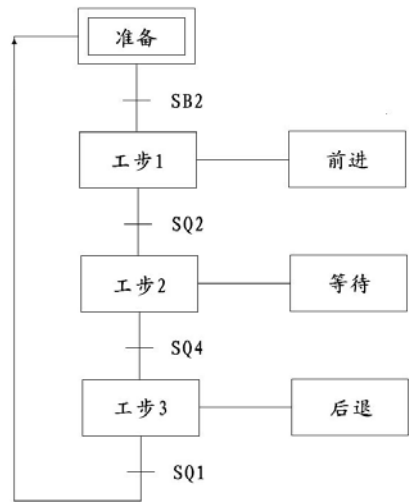
(2) SFC 语言的元素,从图上来看主要由状态、转移和有向线段等组成。

① **状态**表示过程中的一个工步(动作)。状态符号用单线框表示,框内是状态的组件号。一个控制系统还必须要有有一个初始状态,对应与运行的原点,初始状态的符号是双线框。

② **转移**表示从一个状态到另一个状态的变化。转移之间要用有向线段连接,以表示转移的方向。有向线段上的垂直短线和它旁边标注的数字符号表示状态转移的条件。

③ 与状态对应的动作用该状态右边的一个或几个矩形框来表示。

机床的工作原理

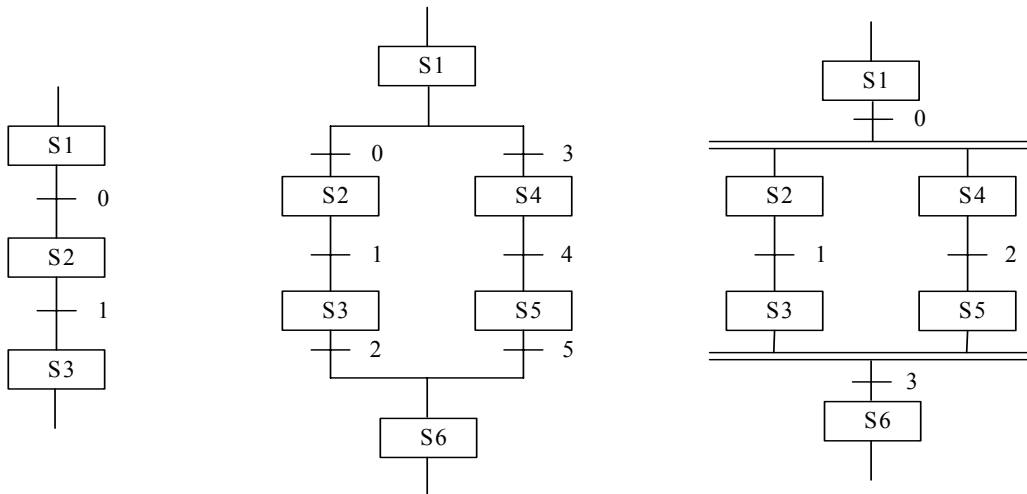


(3) SFC 顺序功能图的基本形式,按结构分为 3 种形式:

① **单流程结构**:是指其状态是一个接着一个地顺序进行,每个状态仅连接一个转移,每个转移也仅连接着一个状态;

② **选择结构**:指在某一状态后有几个单流程分支,当相应的转移条件满足时,一次只能选择进入一个单流程分支。选择结构的转移条件是在某一状态后连接一条水平线,水平线后再连接各个单流程分支的第一个转移。各个单流程分支结束后,也要用一条水平线表示,而且其下不允许再有转移。

③ **并行结构**:是指在某一转移下,若转移条件满足,将同时触发并行的几个单流程分支,这些并行的顺序分支应画在两条双水平线之间。




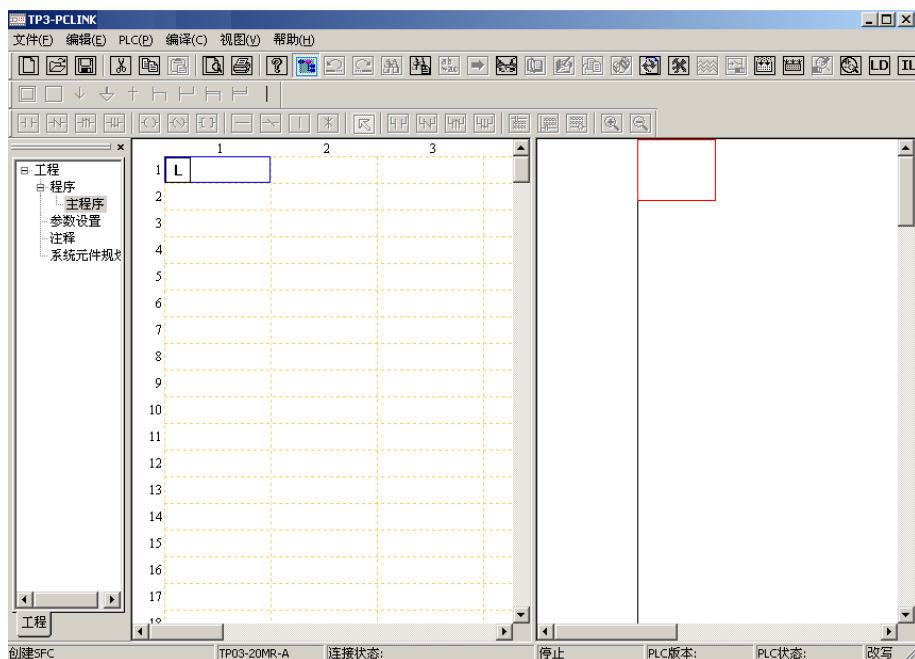
(a) 单流程结构

(b) 选择结构

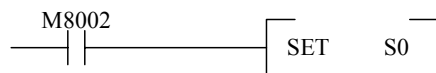
(c) 并行结构


1 BAPS-SP SFC 编程简介

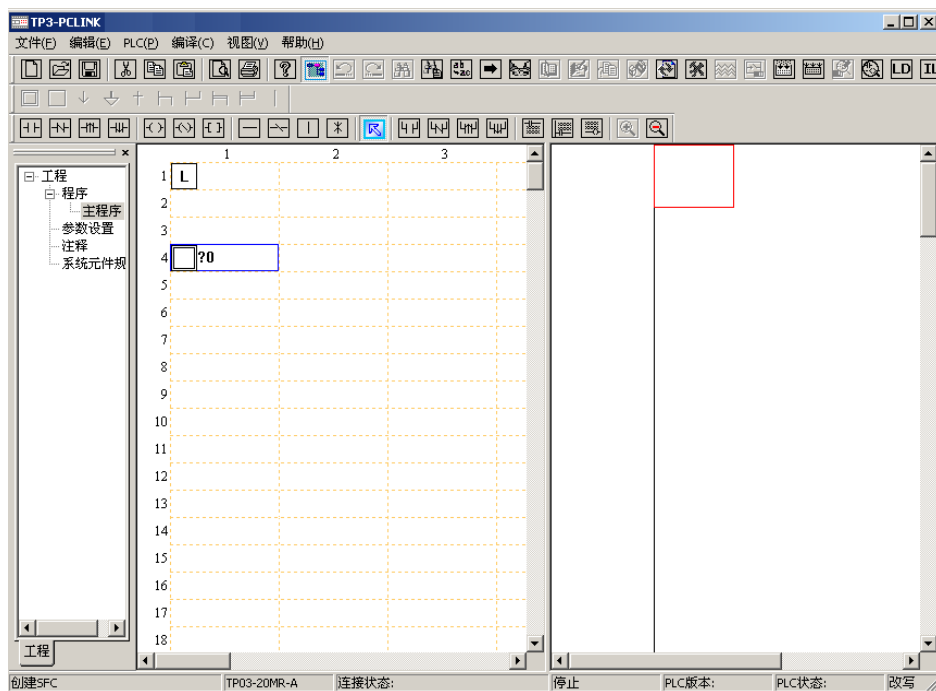
首先新建 BAPS-SP 工程，然后点击工具栏上的“”，出现如下的界面



其中左边一栏用来编辑 SFC 的流程结构图，右边一栏用来编辑梯形图指令。点击左边的“L”可以在右边编写 SFC 的初始化程序，例如：



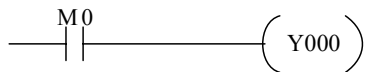
编写完初始化程序后就可以在左侧的第四行开始编写 SFC 的初始步，点击工具栏上的“”如下图：



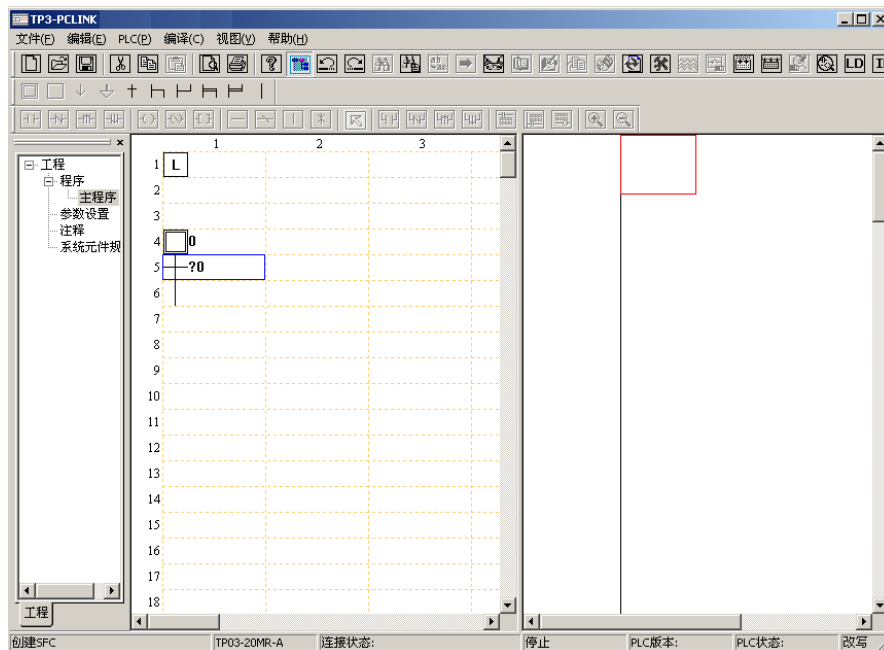
4.步进梯形图指令说明

添加初始化步 S0 后，然后就可以在右侧编写初始化步 S0 相应的梯形图指令。如果 S0 初始步没有编写梯形图指令，则显示如上图的“? 0”，编写梯形图指令“?”就会自动消除。

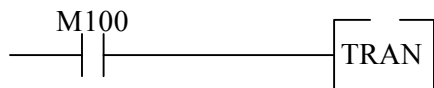
例如：



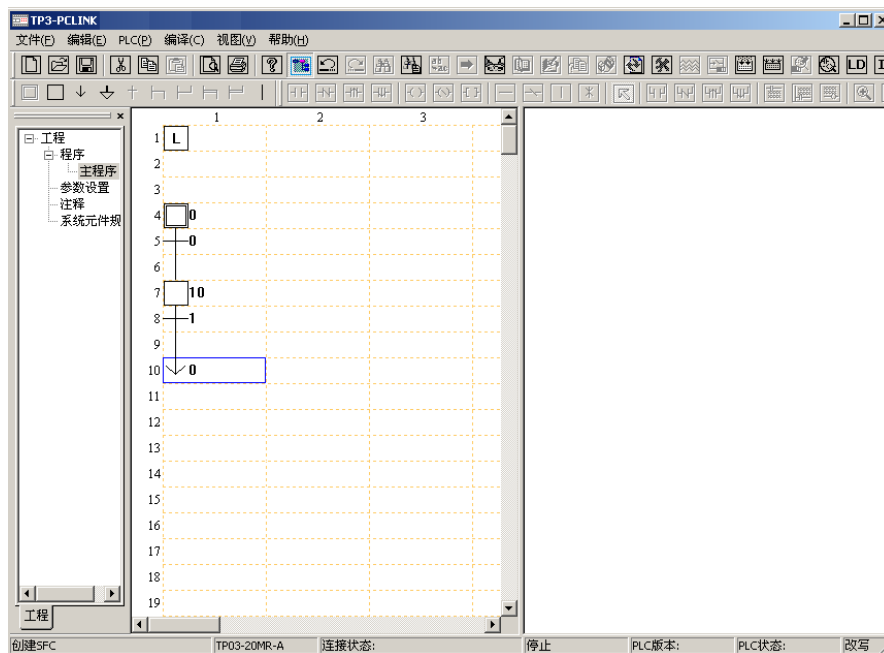
完成初始步 S0 的梯形图指令后，就可以在第五行添加转换条件。如下图：



添加转换条件流程后，就可以在右侧编写转换条件，未编写转换条件显示“? 0”，编写转换条件后，“?”自动消除。例如：



完成初始步和转换条件后，就可以在下面增加其他的步及转换条件。如下图：



2 步进指令 STL, RET

指令符号	功能	电路表示	步数
[STL]	步进阶梯开始		
[RET]	步进阶梯结束		

STL、和 RET 是一对步进指令，表示步进开始和步进结束。

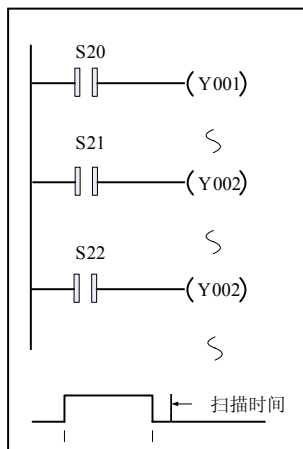
指令说明

(1) STL (步进阶梯): 与主母线连接常开接点指令, STL 接点是用两个小矩形组成的常开接点来表示的。

(2) RET (返回): 返回主母线指令。

步进指令 STL 是执行内部组件步进点 (S) 所用的工步进控制的开始, 初始状态必须由 S0~S9 开始。而 [RET] 则是步进点 (S) 的结束, 同样也必须由 S0~S9 结束, 程序回到主母线。步进梯形图的最后一一定要有 RET 指令, 一个 PLC 程序最多可写入 S0~S9 共计 10 个步进流程, 而每一个步进流程都要 RET 指令作结束。请按照下面的规则来写状态梯形图, SFC 和梯形图是可相互转换的。

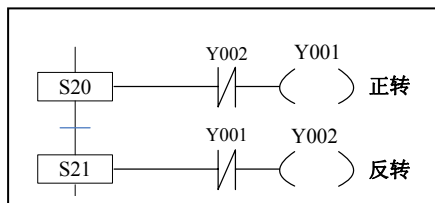
编程与动作



状态的动作与输出的重复使用

- 注: 状态号 S 不可重复使用。
- 如果步进流程结束没有写入 RET 指令, PC06 软件的编辑器会自检并自动添加; 当顺序功能图编辑逻辑关系正确, 在助记符和梯形图中都能看到添加的 RET 指令。
- 如果 STL 触点接通则与之相连的回路动作; 如果 STL 触点断开, 则与其相连的回路不动作。在一个扫描周期后, 不再执行指令 (跳转状态)。
- 如左图所示, 在不同的状态之间, 编程时可输出同一线圈 (Y002)。此时, S21 或 S22 接通时, Y002 线圈会在不同的步接通。而在梯形图编程过程中, 由于双重线圈处理动作复杂, 因此建议不对双重线圈编程。

此外, 当使用 SFC 进行顺序功能图编辑时, 在主程序中对状态内的同一输出线圈 (Y002) 编程, 或在一个状态内对相同的输出线圈编程, 则与梯形图中双线圈一样处理, 请注意。



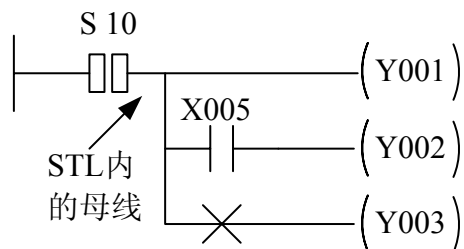
输出的互锁

- 在状态转移过程中, 仅在瞬间 (一个扫描周期) 两种时接通。因此为了避免不能同时接通的一对输出同时接通, 需要根据各可编程控制器的使用手册, 在可编程控制器外部设置上互锁。

定时器的重复使用

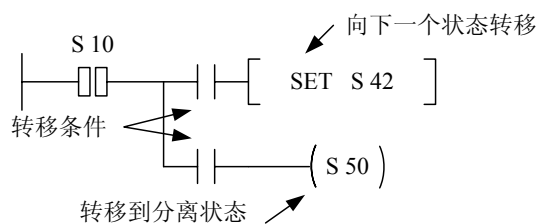
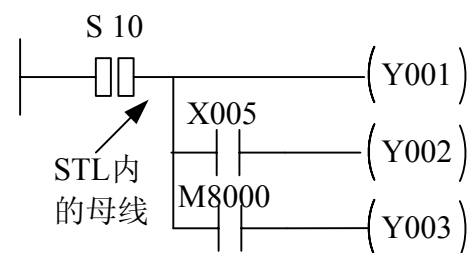
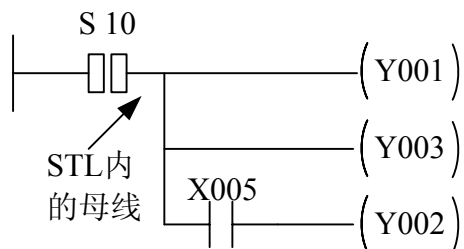
- 在 SFC 顺序功能图中, 定时器线圈与输出线圈一样也可在不同状态下对同一定时器设点, 但是在相邻状态下则不能设定; 如果在相邻状态设定同一定时器, 当状态转移时定时器线圈会不能断开, 当前值不能复位。

针对状态的指令使用



输出的驱动方法

如左图所示，从状态内的母线，一旦写入 LD 或 LDI 指令后，对不需要的触点的指令就不能再编程。需要按下图所示的方法改变这样的回路。



OUT 指令与 SET 指令对于 STL 指令后的状态 (S) 具有同样的功能，都将自动复位转移源。此外，还有自保持功能。但是，使用 OUT 指令时，在 SFC 图中用于向分离的状态转移。

可在状态内处理的顺控指令一览表










指令		LD/LDI/LDP/LDF, AND/ANI/ANDP/ANDF/ OR/ORI/ORF,INV,OUT SET/RST,PLS/PLF	ANB/ORB MPS/MRD/MPP	MC/MCR
状态				
初始状态/一般状态		可使用	可使用	不可使用
分支, 汇合状态	输出处理	可使用	可使用	不可使用
	转移处理	可使用	不可使用	不可使用

在中断程序与子程序中，不能使用 STL 指令

在 STL 指令，不禁止使用跳转指令，但其动作复杂，建议不要使用。

4.步进梯形图指令说明

SFC 中图形符号说明：

符号	说明
	梯形图模式，表示内部编辑程序这一般梯形图而非步进梯形的程序
	初始步进点，代表是 SFC 的初始步进用图形，可使用的装置范围为 S0~S9。
	一般步进点，可使用的装置范围为 S10~S1023
	步进点跳跃，使用在步进点状态转移到非相邻的步进点
	步进点转移条件，各个步进点之间的状态转移条件。
	选择分支图形，由同一步进点将状态以不同转移条件转移到相对应的步进点
	选择汇合图形，由两个以上不同步进点将状态转移经转移条件转移到相同的步进点。
	并联分支，由同一步进点将状态以同一转移条件转移到两个以上的步进点
	并联汇合点，由两个以上不同步进点状态同时成立时，经同一转移条件转移到相同的步进点。

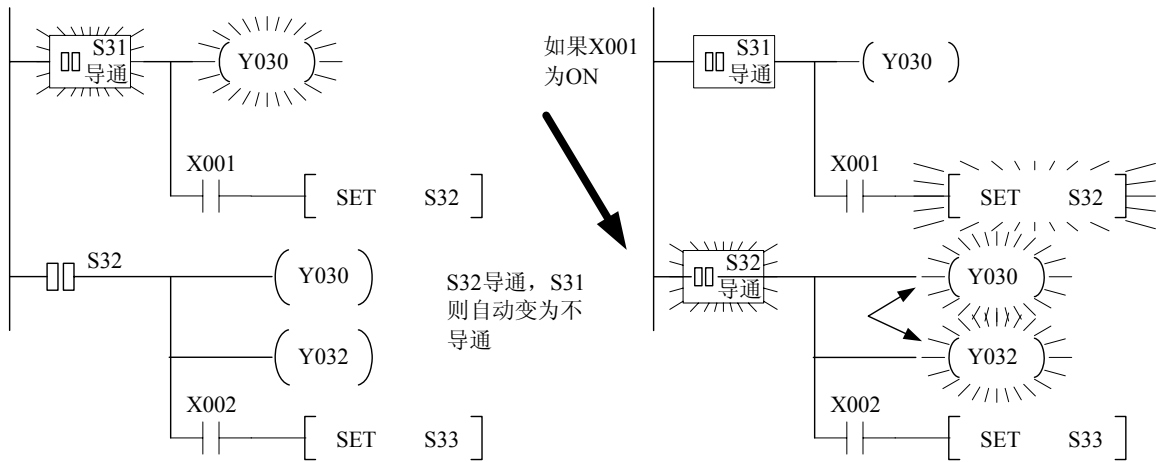
3 步进梯形图指令的动作与 SFC 表示

指令的作用

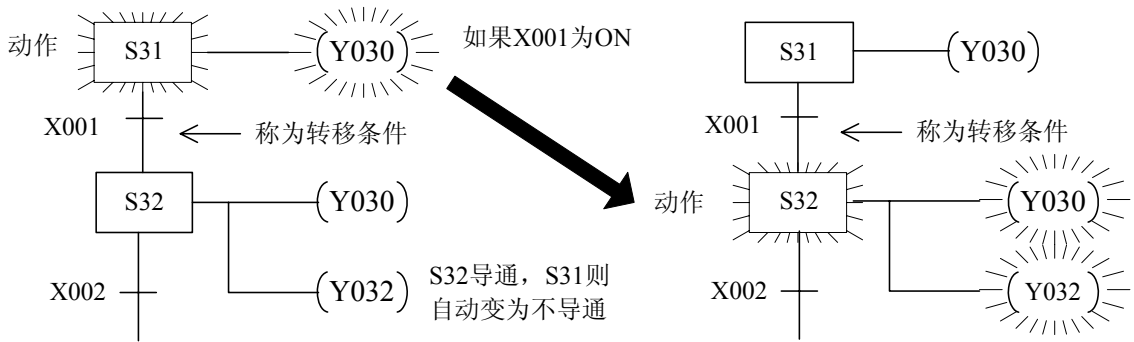
BSP01 系列可编程控制器内置 SFC 图（顺序功能图）的顺控编辑功能。应用程序 SPC 图可转换为指令表或梯形图程序，也可从指令表或梯形图转换为 SFC 图。

- 在 SFC 中，将每个状态看作一个微控工序，将输入条件与输出控制按顺序编程，这种控制最大的特点是在当前工序运行时，前一工序不接通，各道工序顺次运行，达到步进控制的目的。

步进梯形图指令可用梯形图表示及其动作如下图所示：



如果以 SFC 图表示左图所示的步进梯形图回路，则其表示如下图所示

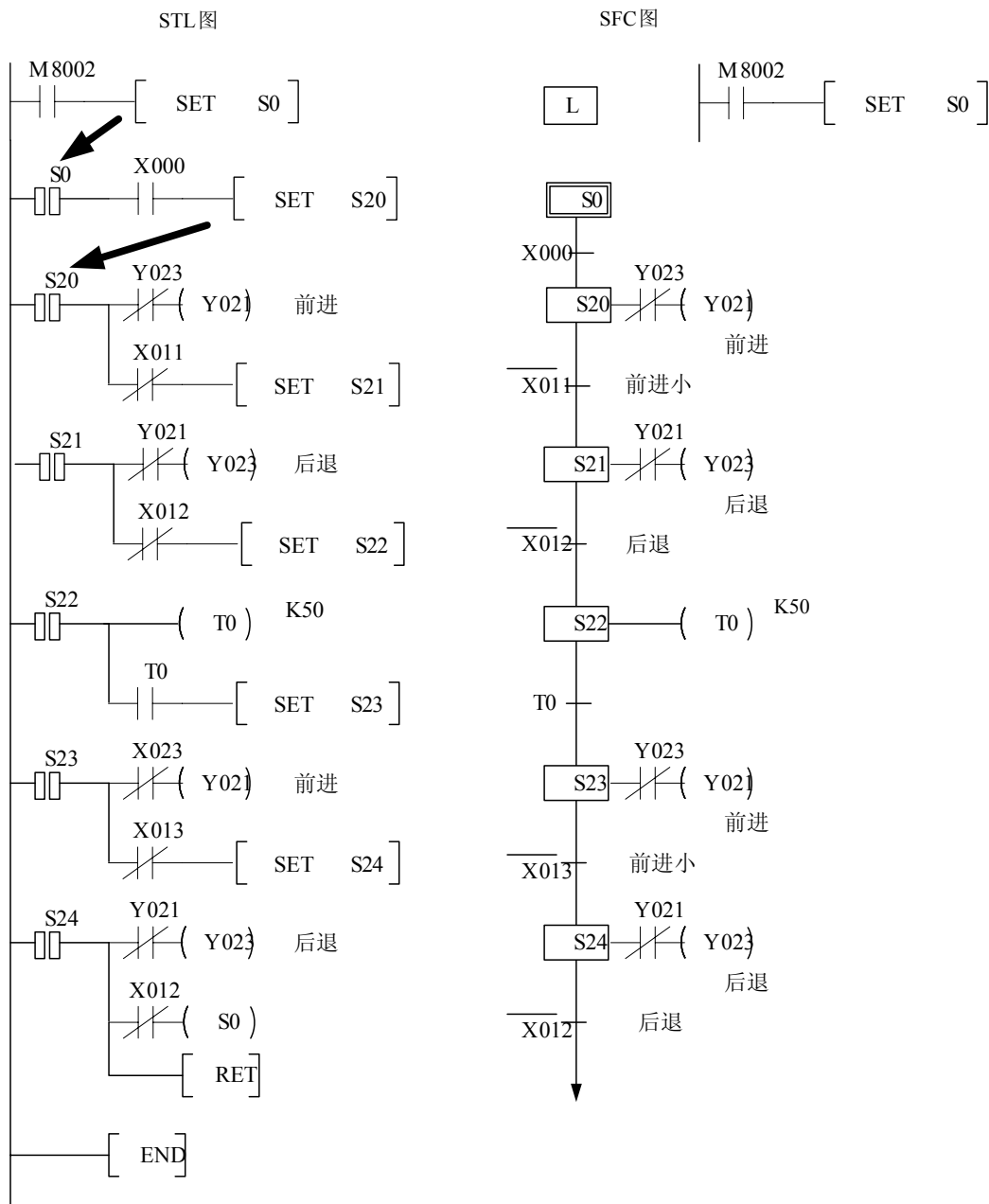


在 SFC 图中，每道工序设备所起的作用以及整个控制流程都能表示得通俗易懂，顺控设计由此变得容易，即便对第三都也可传输严密的动作，因此有利于维护，规格修改和故障排除。SFC 图与步进梯形图指令按一定的规则编程，可相互转换，其实质内容全部一样，也可使用大家熟悉的梯形图，编辑 SPC 图时需要相应的外围设备和编程软件。

4.步进梯形图指令说明

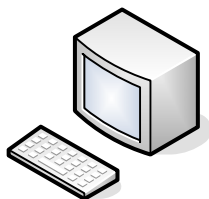
指令的实际表现

如前所述，步进梯形图指令与 SFC 图，其实质内容相同，其实际的程序表示如下。STL 图表现在其梯形图的风格，而 SFC 图由于基于状态（工序）的流程以机械控制表现得更流程化。



编程用设备

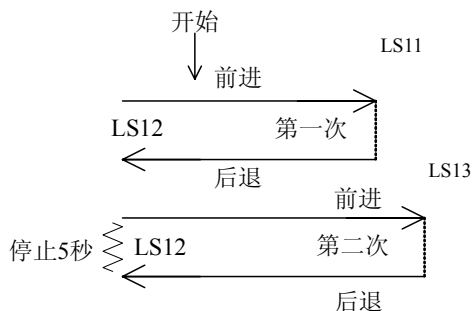
使用个人电脑（PC06 用户软件）可编辑 SFC 图；使用 SFC 图编写的顺控程序可以以指令形式保存在 BSP01 可编程控制器中。



4 SFC 的特点

简单的动作示例

将机械动作写成文件给他人阅读时，一般需要根据时序图或机构图以分条的形式写也其动作的过程，请看下例：



动作

1. 按下启动按钮台车前进，当微动开关 LS11 动作时后退。(LS11 为常 ON，当前进极限时为 OFF，其余微动开关也一样)
2. 后退时，微动开关 LS12 动作，停止 5 秒后再前进，微动开关 LS13 动作后再后退。
3. 当微动开关 LS12 再动作时，台车驱动马达停止。

当此动作用文章正确的叙述后，复杂的机械动作变为简单，机械技术者与电气技术者之间可商讨更深入的问题。

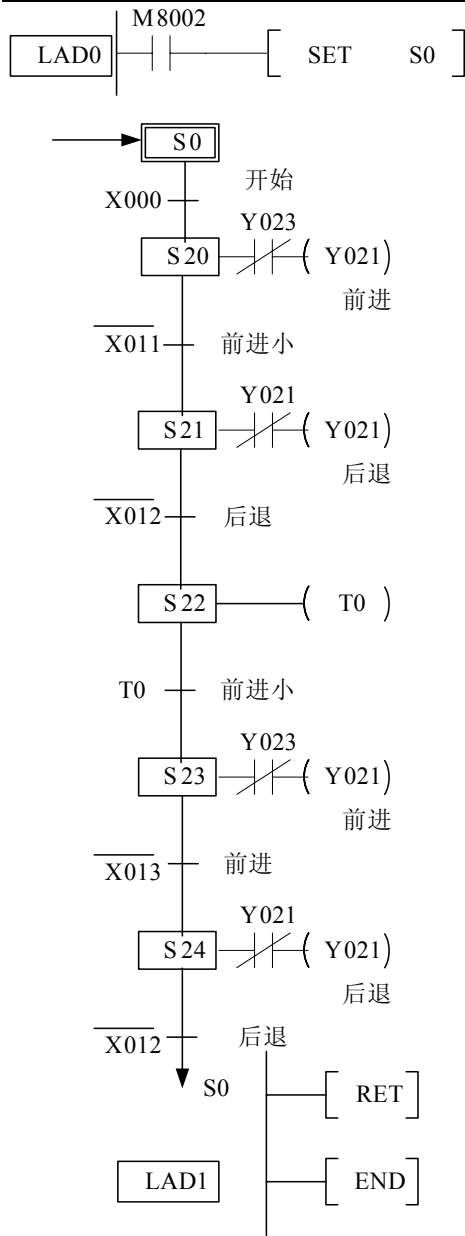
基本上电气技术者要设计 PLC 程序，没有 PLC 的动作图，是无法设计的。

但是，工程步进的动作对于机械的电气设计者，非常的复杂，需要相当的经验且须设计时间。且 PLC 图在第三者来看是相当复杂的，此 PLC 图的设计者将无法离开机械维护的责任。

单流程的处理

工序转移的基本形式是单流程形式的控制，在要求单纯动作的顺序控制中，有单流程已足够了，但在具有各种输入条件与操作人员操作情况下，可能过与后述的选择公支及并行分支流相结合，从而简便地处理复杂的条件。

4.步进梯形图指令说明



- 在梯形图电路块 LAD0 中，采用可编程控制器由停止——运行转换时，瞬间动作的辅助继电器 M8002，使初始状态 S0 置位（ON）

- 对于机械有初始工序，分配了这种可编程控制器中被称为 S0~S9 的初始状态软组件。

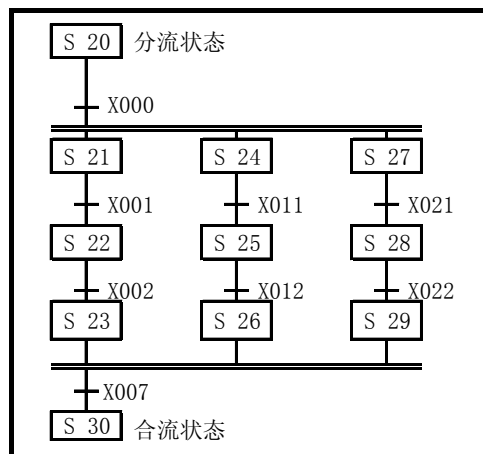
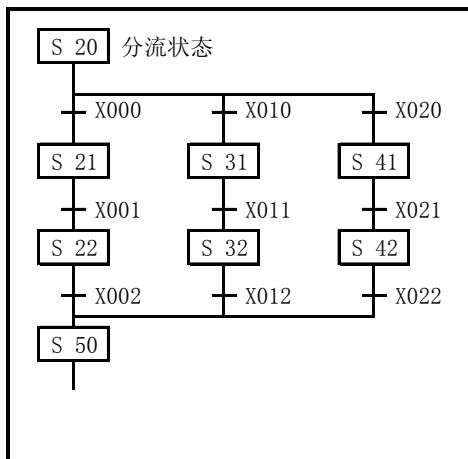
- 对各动作工序分配了 S20~S899 等状态，其中也有停电保持用的状态，即使在停电时也可保存其动作状态，此外，S10~S19 在采用 IST 指令时，可用于特殊目的。

- 在可编程控制器内有定时器，计数器和辅助继电器等许多软元件，可随意使用。这种采用了定时器 T0，该定时器以 0.1 秒时钟为单位动作。因此，设定值为 K50 时，在线圈驱动 5 秒后，输出触点动作。

多项工序的选择性处理及同时处理

选择执行多项流程中的某一项流程被称为选择性分支；

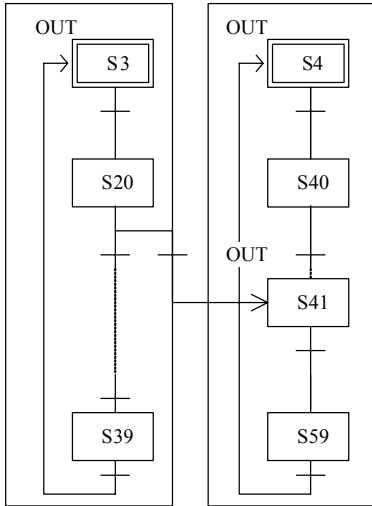
多项流程同时进行的分支，称为并行分支。



5 编程 SFC 流程的预备知识

流程的分离

SFC 图程序拥有多个初始状态，而各个初始状态都为分离程序。



以上图为例，附属于起始状态 S3 之 S20~S39 的 STL 命令程序执行完了后，再执行初始状态 S4 之关联程序。

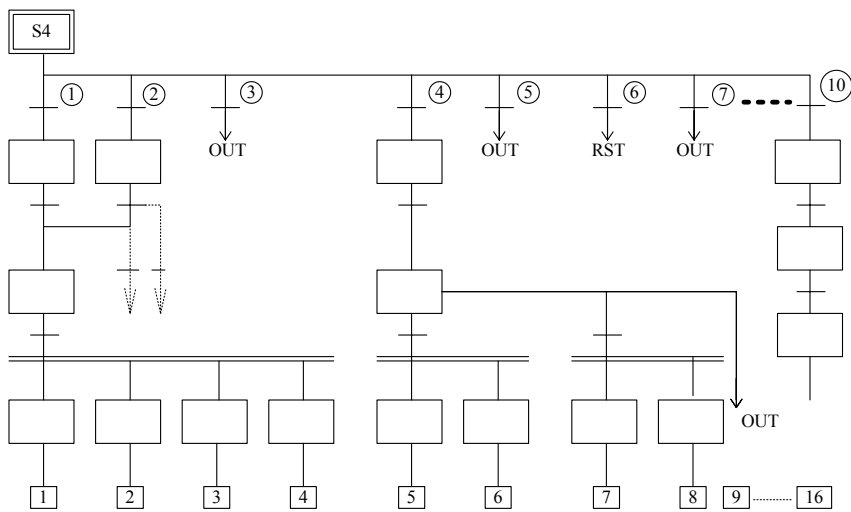
但在程序中可以运用 STL 命令以外的命令去执行其它状态编号。

在左图中初始状态 S3 的程序中含有 OUT S41 命令。初始状态 S4 程序中有 LD S39 命令，关键不要混杂 STL 指令。

分支回路数的限制

一个并进分支或选择性分支的回路数限定为 10 条以下：

但是，有多条并行分支或选择性分支时，每个初始状态的回路总数不超过 16 条。

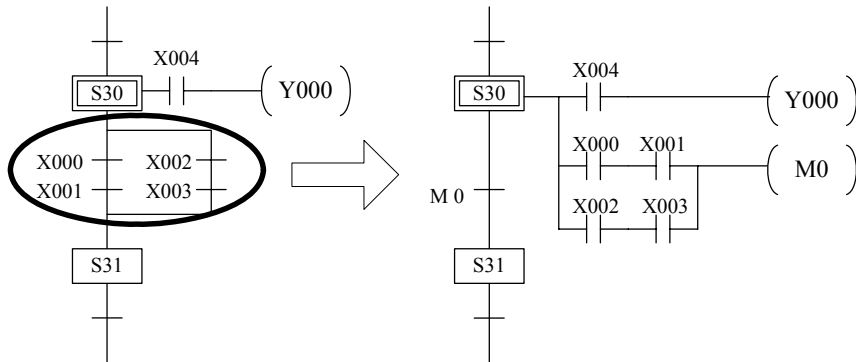


不能进行从汇合线或汇合前的状态开始向分离状态的转移处理或复位处理。一定要设置空状态，从分支线上向分离状态进行转移与复位处理。

4.步进梯形图指令说明

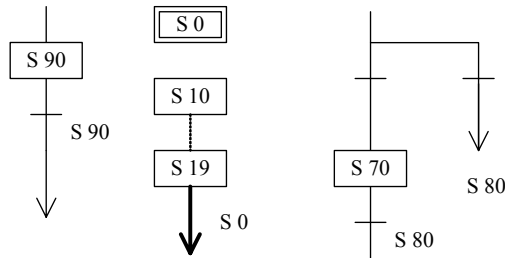
复杂移行条件的程序

BSP01 SFC 程序中的每个节点的条件只能为一个，如果实际应用中需编写下方左图中的程序，可以用右图的方法进行编程。



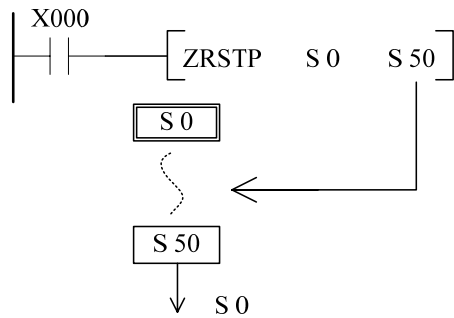
移行条件电路中不可使用 MPS,MRD,MPP 命令，请以右边的方法编程。

▽、J的动作

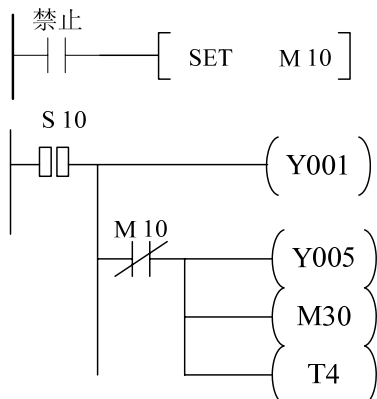


在流程中表示状态的复位处理时，以符号▽表示。

符号J则表示向上的状态转移重复或向下面的状态转移（跳转），或者向分离的其他流程上的状态转移。



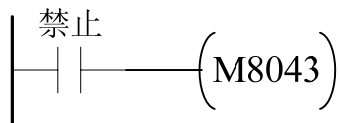
状态的清除和禁止输出



4.步进梯形图指令说明

禁止动作中的步进任意输出

PLC 的全部输出继电器(Y)都为 OFF。



特殊继电器

为有效的编写 SFC 图，需要采用数种特殊继电器，其主要内容如下：

组件号	名称	功能和用途
M8000	运行监视 (a 接点)	可编程控制器在运行过程中，需要一直接通的继电器，可作为驱动的程序输入条件可作为可编程控制器运行状态的显示来使用。
M8002	起始脉冲 (b 接点)	在可编程控制器由停止—运行时，仅在瞬间（一个扫描周期）接通的继电器。用于程序的起始设定或初始状态的置位。
M8040	转移禁止	驱动该继电器，则禁止在所有状态之间转移，然而即使在禁止转移状态下，由于状态内的程序仍然动作，因此，输出线圈等不会自动断开。
M8046	STL 状态动作	任一状态接通时，M8046 自动接通，适用于避免与其它流程同时启动或用作工序的动作标志
M8041	步进开始	IST 指令用旗标
M8047	STL 监视有效	驱动该继电器，则编程功能则可自动读出正在动作的状态并加以显示。

停电保持状态

停电保持状态是用电池保持其动作状态。在机械动作中途发生停电之后，再通电时从这里继续动作的情况下使用这些状态。

RET 指令的作用

RET 指令一定要在一系列的 STL 指令的最后编写。

执行此指令，意味着步进梯形图回路的结束，在希望中断一系列的工序而在主程序编程时，同样需要 RET 指令。

RET 指令可多次编程。

在 STL 指令的最后，没有编写 RET 指令时，软件会自动在程序的结尾加上 RET 指令。

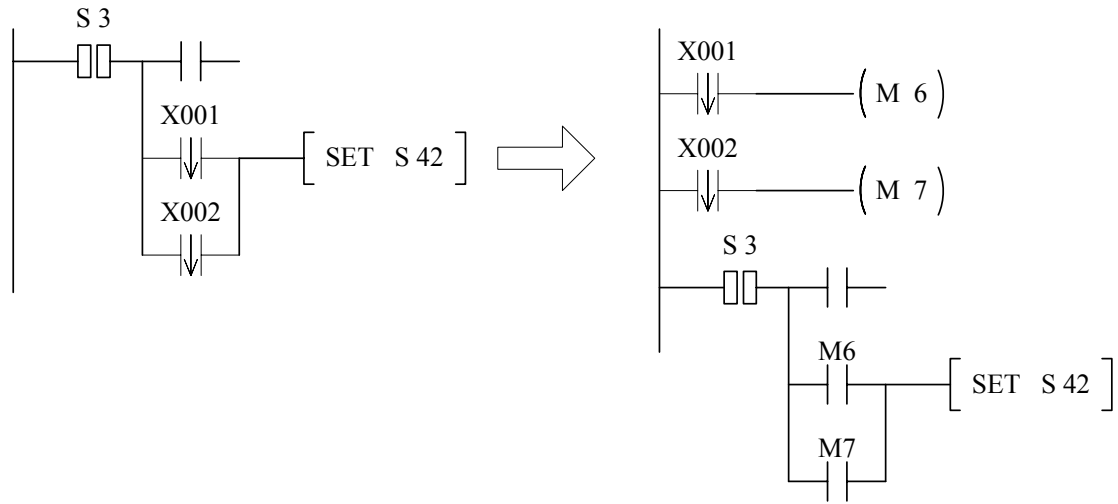
4.步进梯形图指令说明

注意事项

上升沿，下降沿检测触点使用时的注意事项

在状态内使用 LDP, LDF, ANDP, ANF, ORP, ORF 的上升沿，下降沿检测触点时，状态断开时变化的触点，在状态再次接通时被检出。

对于状态断开时变化的条件，必需上升沿，下降沿检测时，请按下图所示，修改程序。



通过 X001 下降沿向 S42 转移后，若 X002 下降，此时因 S3 断开，X002 的下降沿无法检出，S3 再次接通时，被检测。因此，S3 的第二次动作时，立即向 S42 转移。

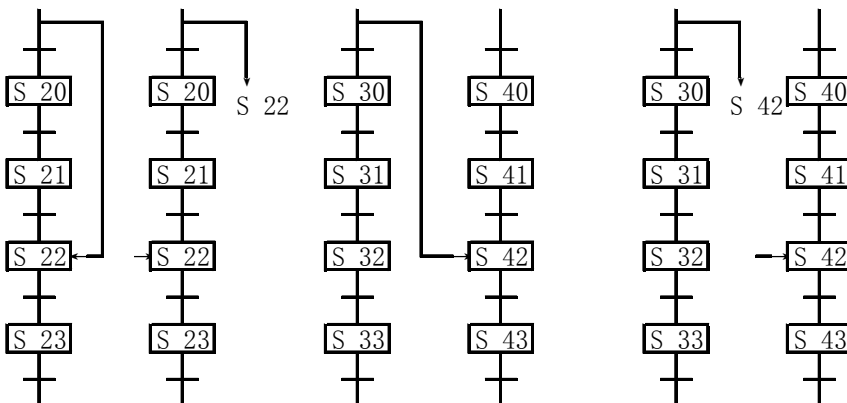
6 SFC 流程的形态

6.1 跳转与重复流程

表示 SFC 图单流程的动作模式与选择性分支及并进分支组合的动作模式

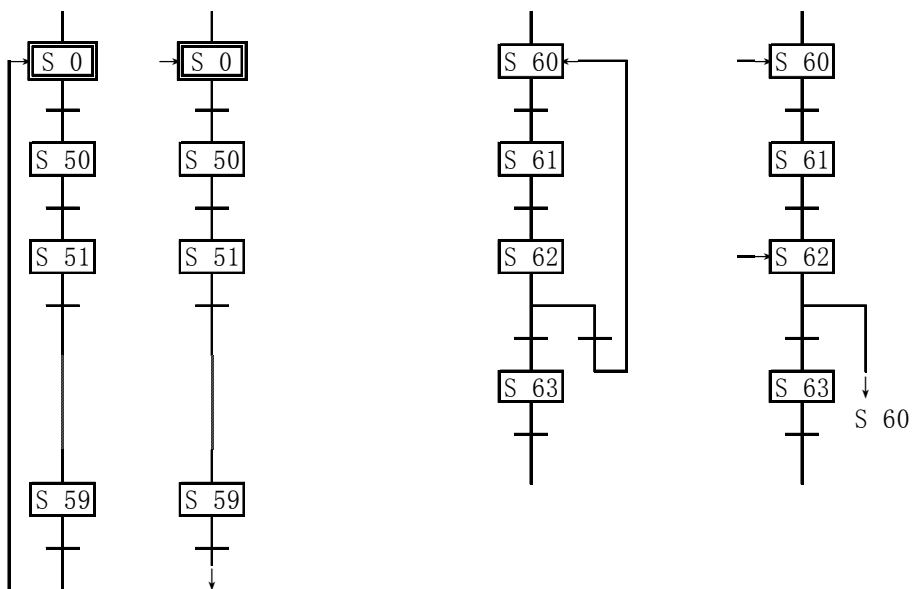
跳转

向下面状态的直接转移或向系列外的状态转移称为跳转，以符号↓表示转移的目标状态。

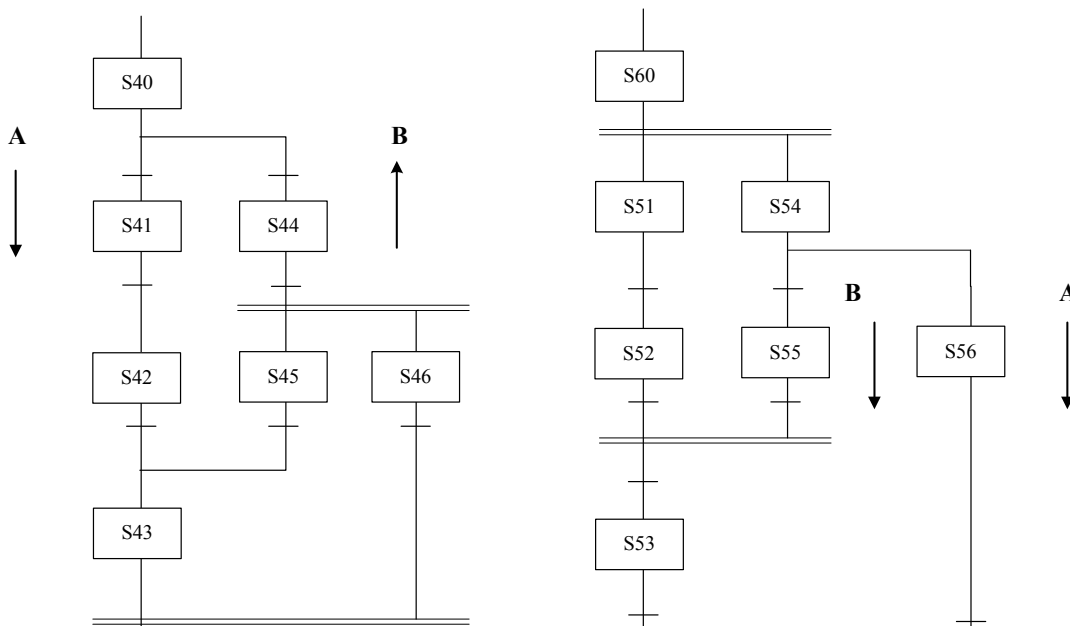


重复

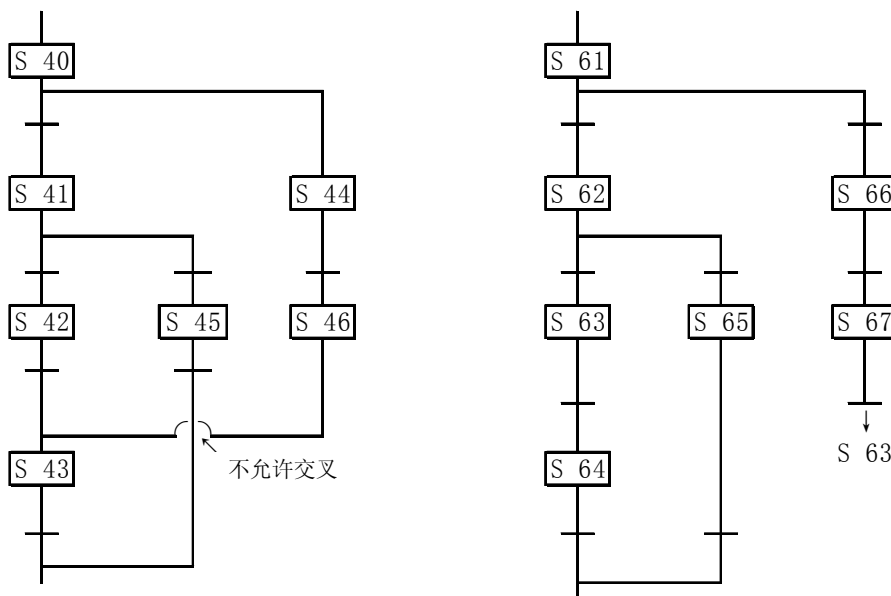
向上面状态的转移称为重复，与上面一样，用符号↓表示转移的目标状态



6.2 分支与汇合的组合流程



如上图所示的流程都是可能的程序。B 的流程没有问题，但在 A 流程的情况下，在并进汇合处有等待动作的状态，请务必注意。

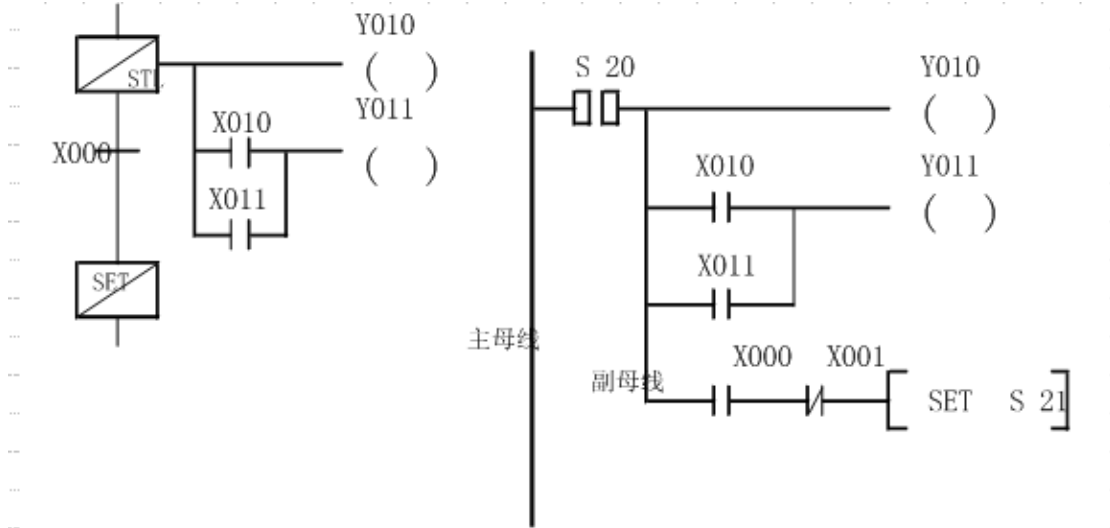


不能作流程交叉的 SFC 图，上图所示的流程要按右边所示的流程重新编程，利用它可实现以指令为基础的程序向 SFC 图的逆转换。

8 中间状态的程序

8.1 没有分支与汇流的一般流程

下图是从 SFC 图中抽出来的具有代表性的一种状态。每种状态具有驱动负载、指定转移目标以及指定转移条件三种功能，使用继电器顺序控制表示 SFC 图时，是下面的步进梯形图。程序用 SFC 图或用步进梯形图均可编写。编程顺序为先进进行负载的驱动处理，接着进行转移处理。当然如果是不需要驱动负载的驱动处理。



0	STL	S 20
1	OUT	Y 010
2	LD	X 010
3	OR	X 011
4	OUT	Y 011
5	LD	X 000
6	ANI	X 001
7	SET	S 21

用于状态的 SET, RST 指令为 2 步指令

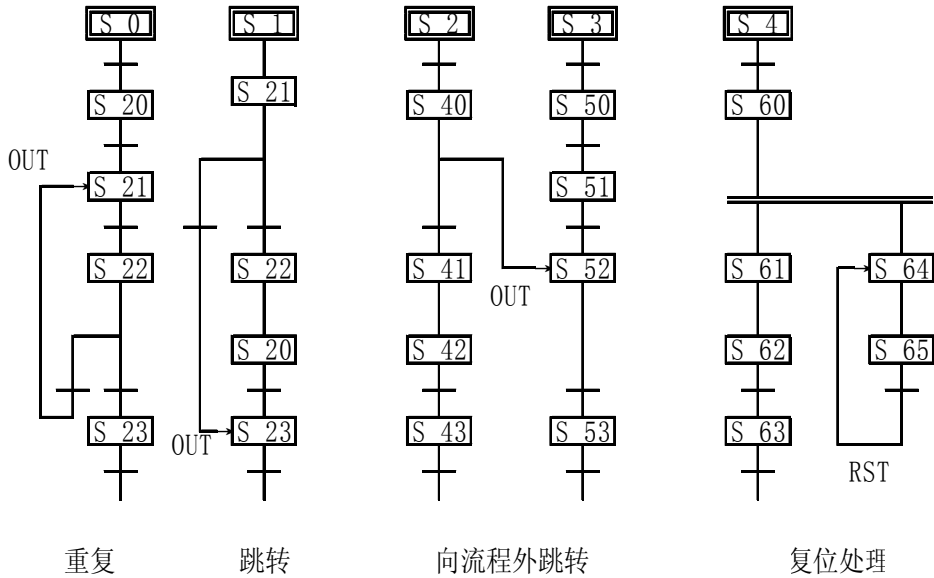
以指令表来表示上图的程序，如左图所示。

STL 指令为与主母线连接的常开触点指令，接着就可在副母线上直接连接线圈，或者可以通过触点驱动线圈。

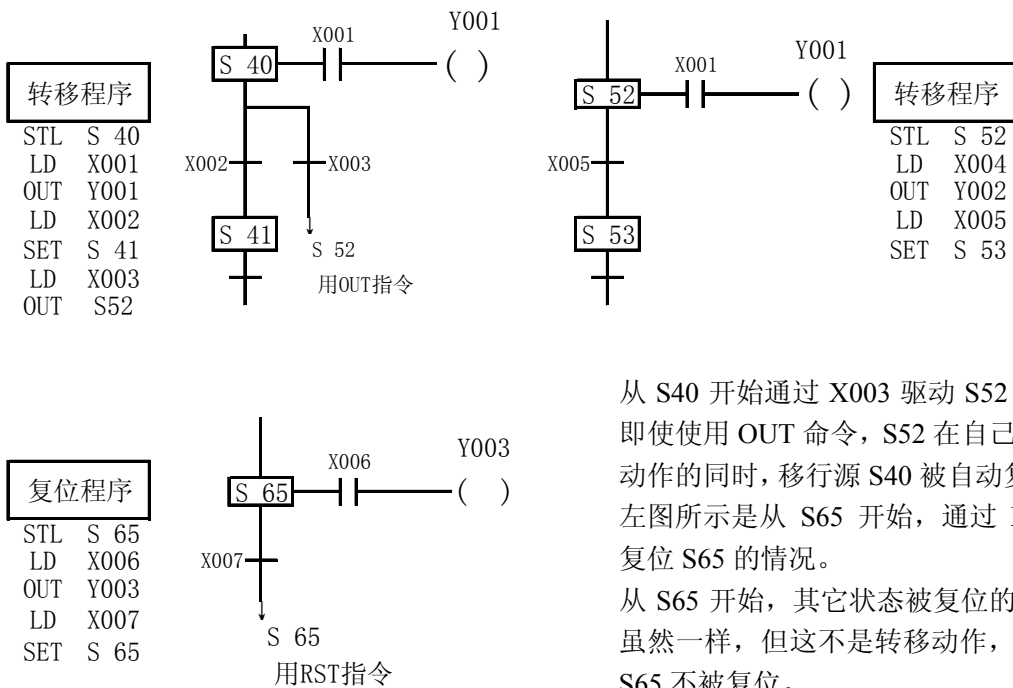
连接在副母线上的触点使用 LD (LDI) 指令。若要返回原来的主母线时使用 RET 返回指令，通过 STL 触点驱动状态 S，在该 S 移动前的那个状态自动复位。

对于连贯的 SFC 图，执行种种状态的程序，如果将所有状态都编入程序中，则编也就结束。其状态顺序编号可自由选择。但是，在一系列的 STL 指令前要有初始状态，最后一定要写入 RET 指令。

8.2 带有跳转与重复的一般状态



如上图所示，向上方状态的转移（重复）、向下方状态转移（跳转）、向流程外的转移等的分离状态转移，如 4-5 项所述，用 \hat{e} 符号表示转移目标状态号。如下所示，用 OUT 指令编程，4-5-1 节的交叉流程的情况也是一样。

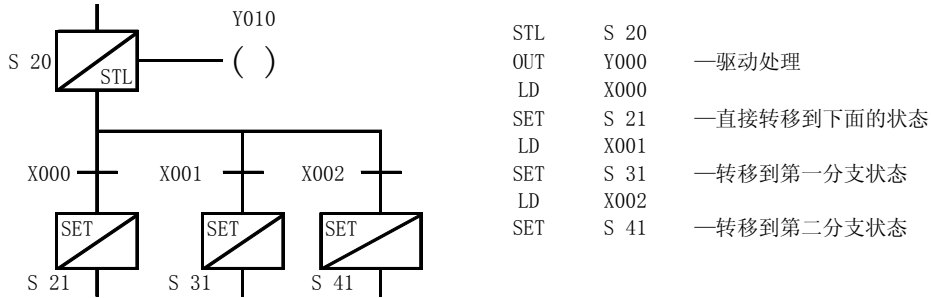


从 S40 开始通过 X003 驱动 S52 时，即使使用 OUT 命令，S52 在自己保持动作的同时，移行源 S40 被自动复位。左图所示是从 S65 开始，通过 X007 复位 S65 的情况。从 S65 开始，其它状态被复位的情况虽然一样，但这不是转移动作，因此 S65 不被复位。

9 分支与汇合状态的程序

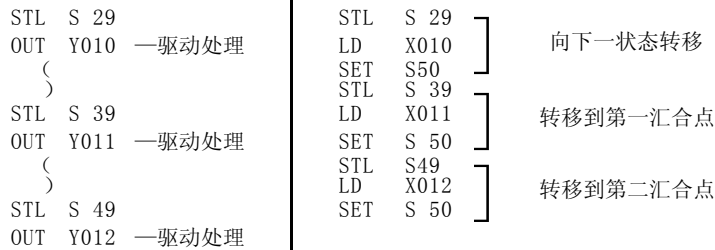
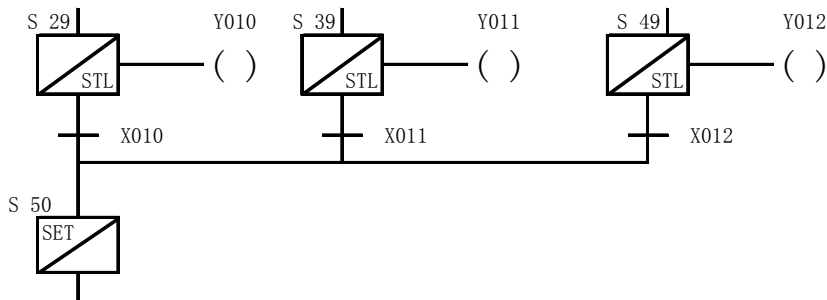
9.1 选择性分支与汇合状态

选择性分支示例：



与一般状态的编程一样，先进行驱动处理，然后进行转移处理。所有的转移处理按顺序继续进行。

选择汇合示例：

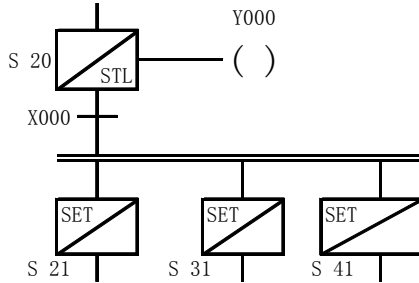


首先只进行汇合前状态的驱动处理，然后按顺序继续进行汇合状态转移处理。这就成为向SFC画面进行逆变换的必要规则。

请注意程序的顺序号，分支列与汇合列不能交叉。

9.2 并行分支与汇合状态

并行分支示例

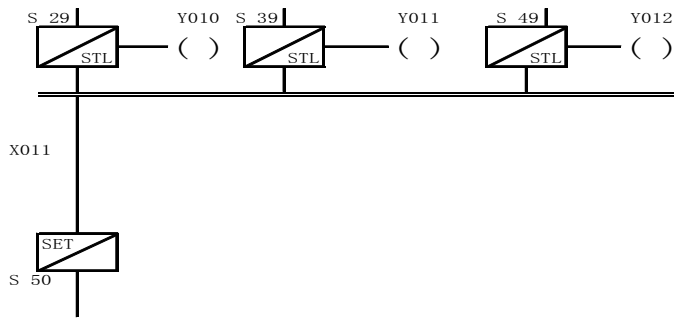


```

STL   S 20
OUT   Y000    —驱动处理
LD    X000
SET   S 21    —直接转移到下面的状态
SET   S 31    —转移到第一分支状态
SET   S 41    —转移到第二分支状态
    
```

与一般状态的程序一样，首先进行驱动处理，然后进行转移处理，所有的转移处理按顺序继续进行。

并行汇合示例



```

(
)
STL   S 29
OUT   Y010    —驱动处理
(
)
STL   S 39
OUT   Y011    —驱动处理
(
)
STL   S 49
OUT   Y012    —驱动处理
(
)

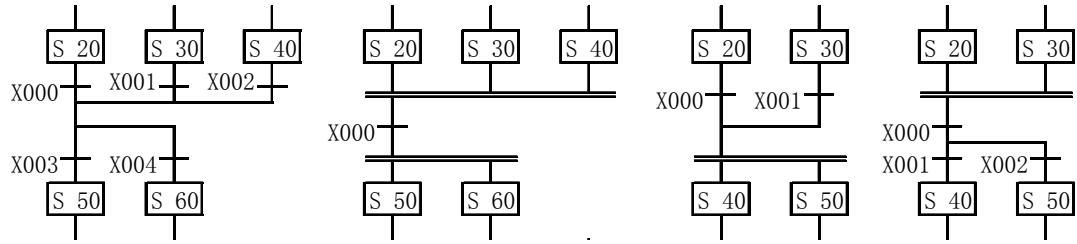
```

```

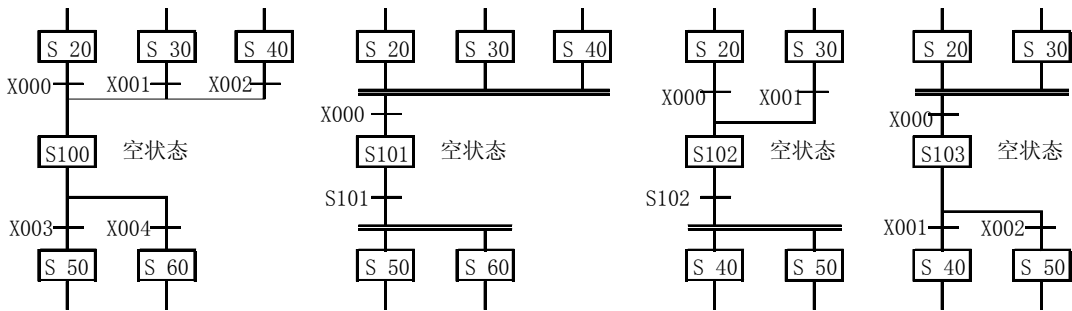
→ STL S 29
→ STL S39
→ STL S49
LD   X010
AND  X011
AND  X012
SET  S 50
(
)
    
```

连续的STL指令表示并行汇合。
并行的分支限制为8路以下。

9.3 分支与汇合的组合



变成以下形式：

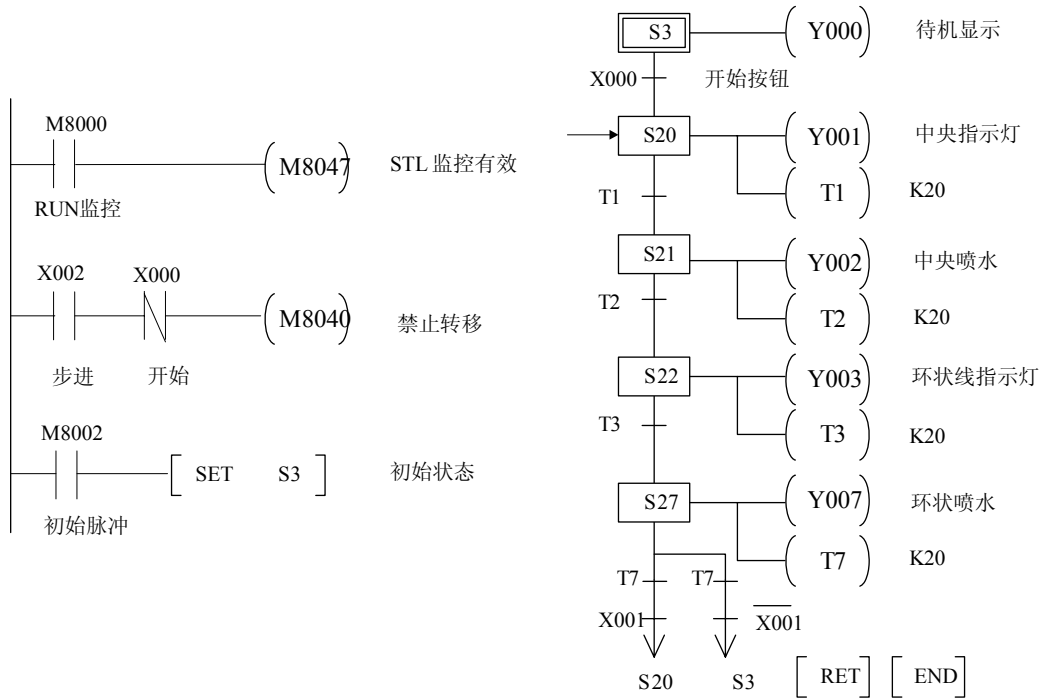


STL S20	STL S 20	STL S 20	STL S 20
LD X000	STL S 30	LD X000	STL S 30
SET S100	STL S 40	SET S102	LD X000
STL S 30	LD X000	STL S 30	SET S103
LD X001	SET S101	LD X001	STL S103
SET S100	STL S101	SET S102	
STL S 40	LD S101	STL S102	LD X001
LD X002	SET S 50	LD S102	SET S 40
SET S100	SET S 60	SET S 40	
STL S100		SET S 50	LD X002
			SET S50
LD X003			
SET S 50			
LD X004			
SET S60			

如上图之合流线之后直接分歧线是不好的作法，在这之间有一虚拟状态点比较好写。

10 单流程示例

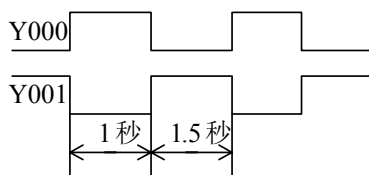
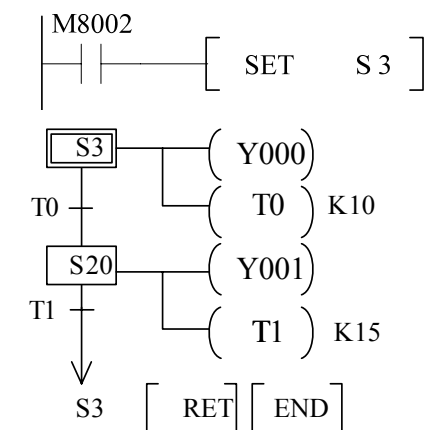
喷水控制示例



上图

1. 单次运转(X001=OFF, X002=OFF)当启动按钮 X000 动作, 会按照 Y000→Y001→Y002→Y003→Y007→Y000 的顺序动作, 并返回到待机状态。用 2 秒的定时器依次各输出。
2. 连续运转(X001=ON), 重复 Y001~Y007 的动作。
3. 步进运转(X002=ON) 每按一次启动按钮, 顺序各输出动作。

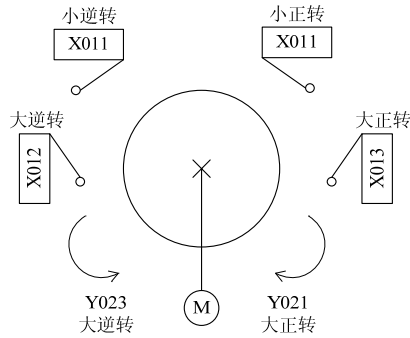
闪烁回路示例



4.步进梯形图指令说明

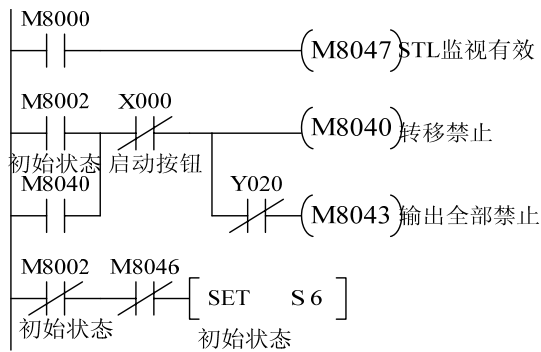
- 1、当可编程控制器运行，初始脉冲（M8002）驱动状态 S3。
- 2、状态 S3 变为 ON 以后，会输出 Y000 同时定时器计时，1 秒钟以后，计时结束会向状态 S20 转移。
- 3、状态 S20 变为 ON 时，输出 Y001 定时器计时，1.5 秒以后，返回状态 S3。

凸轮轴的旋转控制

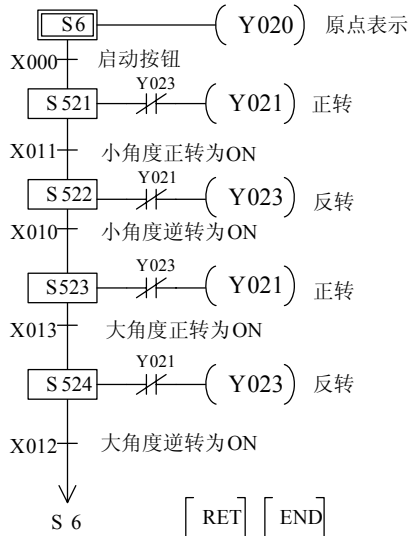


正转角度大小 2 个位置中设有限位开关 X013, X011, 并且在反转角度的大小 2 个位置中设有限位开关 X012, X010。

按下启动按钮，执行正转小→逆转小→正转大→逆转大的动作，然后停止。



限位开关为 X010~X013 为常 OFF，当凸轮所设定角度到达为 ON。

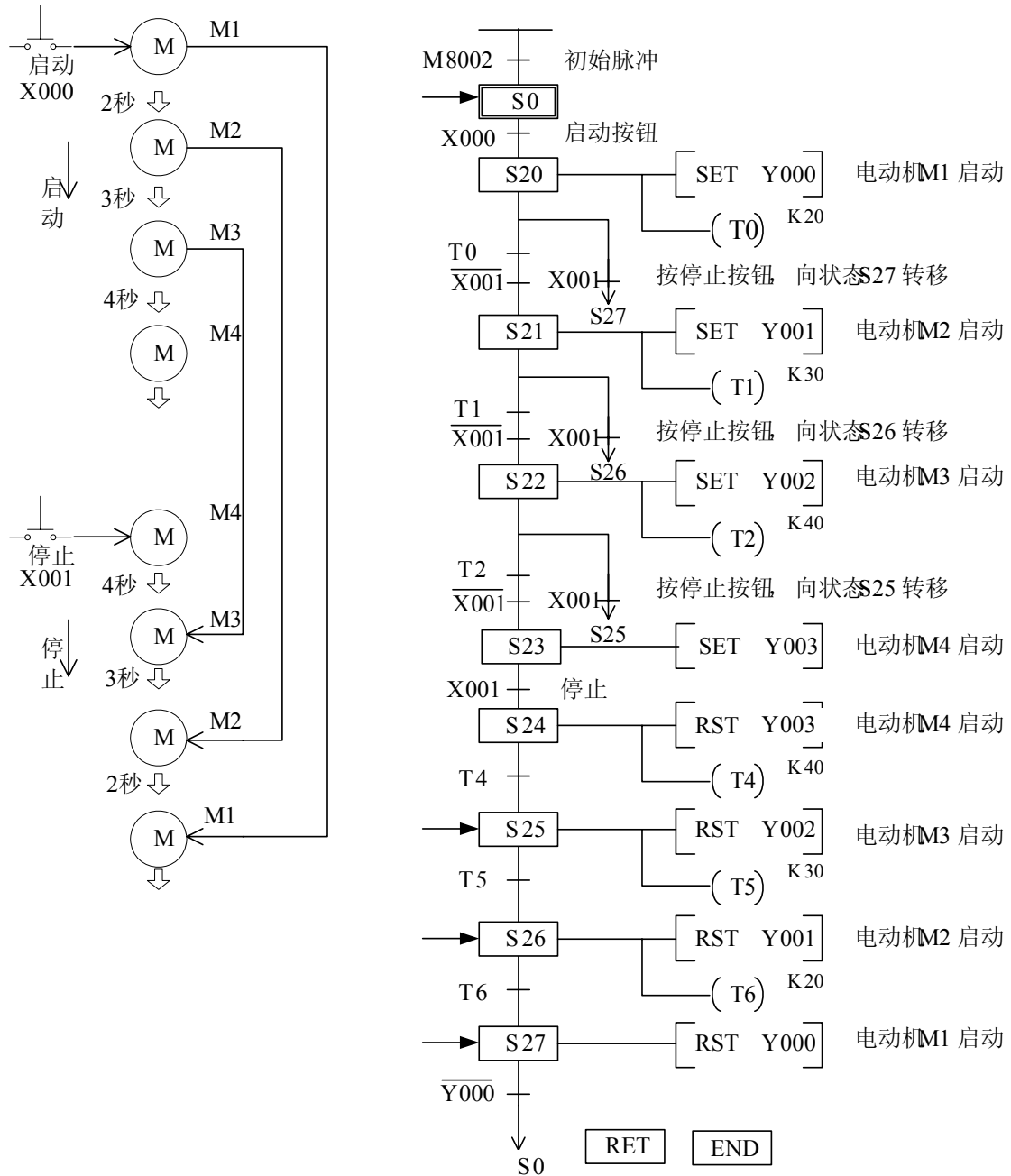


- 当 M8047 动作时，动作状态监视有效，S0~S899 动作后，END 命令实行后，M8046 动作。
- SFC 图的状态点是由电池来保持，在动作的途中即便停电，开始按钮再按时将会在继续动作，但在尚未按下开始键时 Y020 以下的输出动作完全禁止。

4.步进梯形图指令说明

顺序启动与停止示例

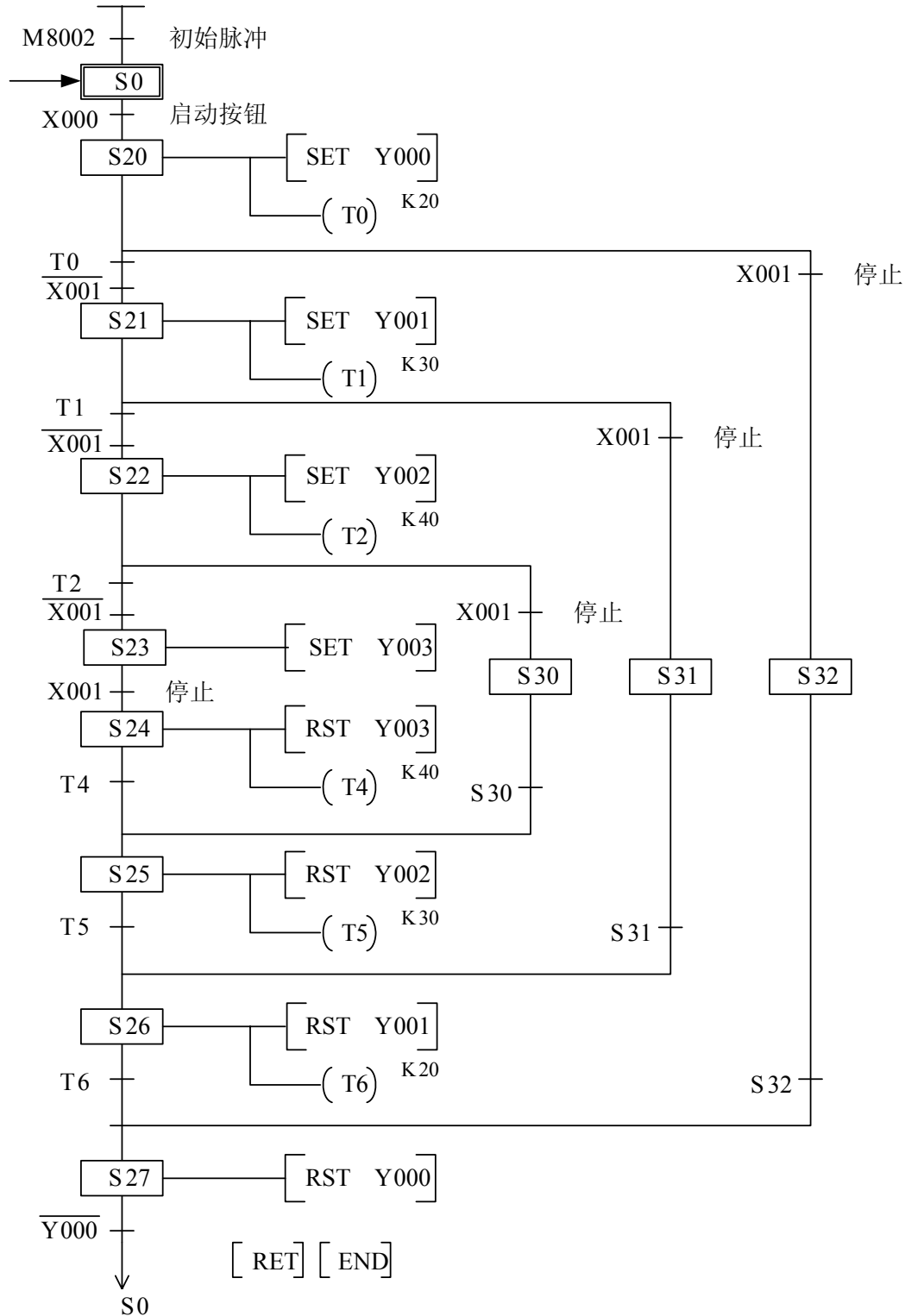
由定时器控制电机 M1~M4 按顺序启动，以相反顺序停止。
这种 SFC 流程以单流程为基础进行状态的跳转。



该例所示为该 SFC 图根据条件，跳过一部分流程向后面的状态转移。也可向前面的状态跳转。

4.步进梯形图指令说明

上页的跳跃流程也可用如下所示的选择分支与汇合流程来表示，流程的流向必须是从上到下，除分支、汇合线外，不能交叉。

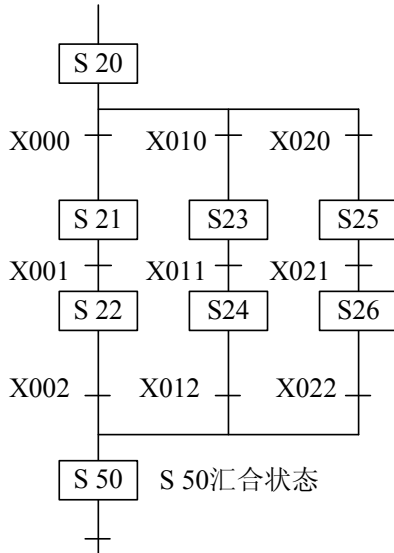


例如状态 S20 的动作时，若 X001 接通，则状态 S32 动作，然后其触点动作，从而直接跳转到状态 S27。

在分支线路上一定要有一个以上的状态，所以需设置空状态。

11 选择性分支与汇合流程示例

选择性分支的动作

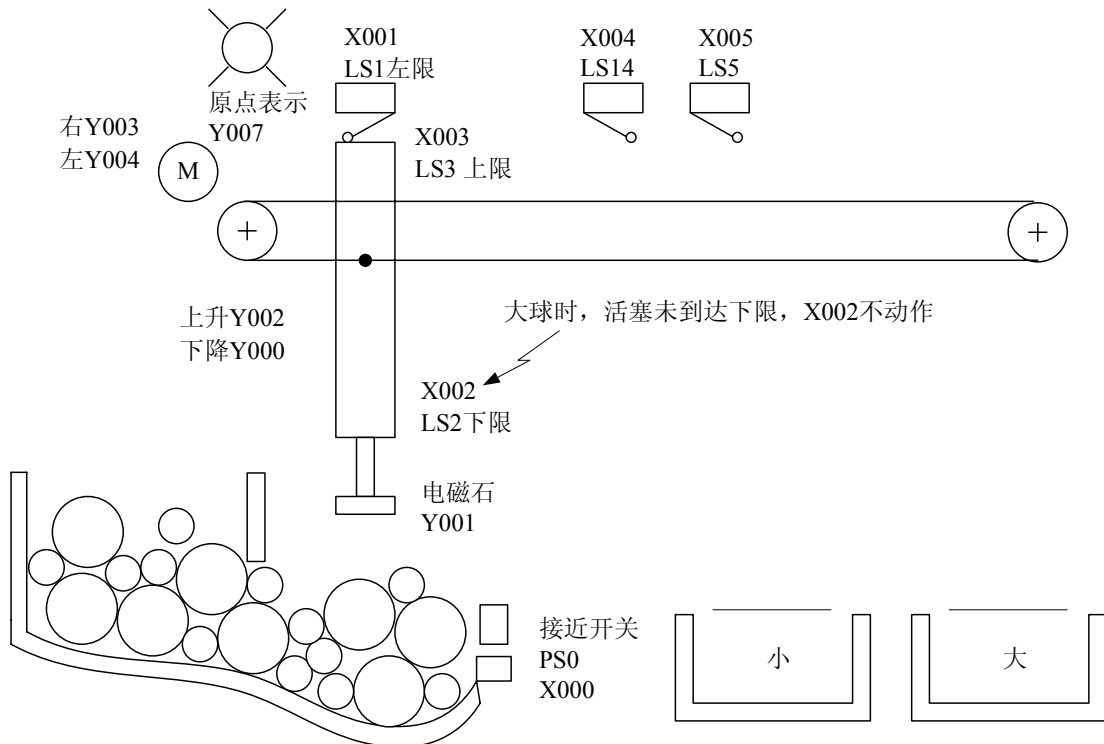


- 从多个流程中选择一个流程执行被称为选择性分支。
- 以左图为例，必须X000、X010、X020不同时接通。
- 例如：在S20动作时，若X000接通，则动作状态就向S21转移，S20变为不动作。因此，即使以后X010、X020动作，S23、S25也不会动作。
- 汇合状态S50，可被S22、S24、S26中任意一个驱动。

选择性分支的动作

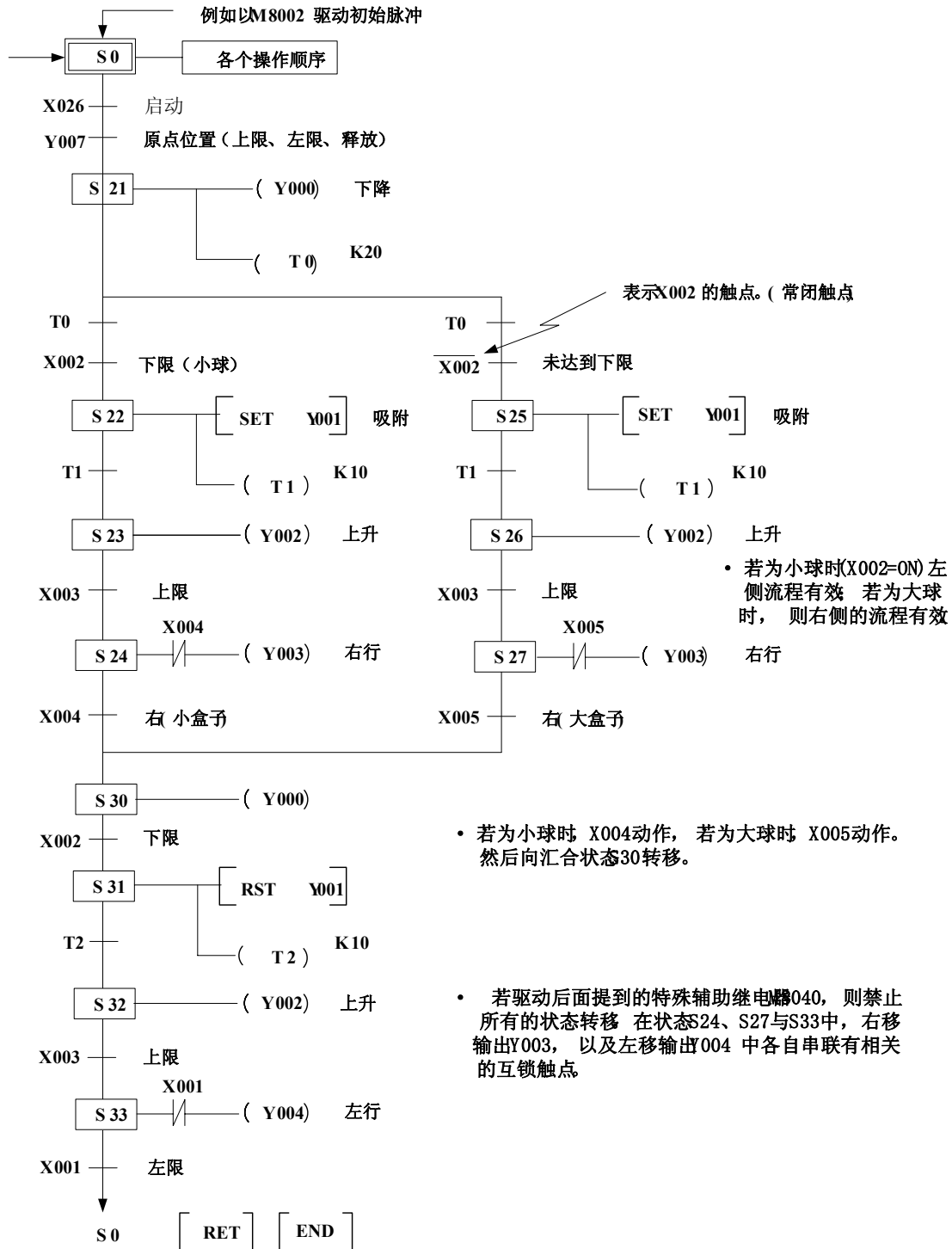
下图为使用传送点，将大、小球分类选择传送的机械。

左上方为原点，其动作顺序为下降、吸住、上升、右行、下降、释放、上升、左行。此外，机械臂下降，当电磁铁压着大球时，下限限位开关LS2断开；压着小球时，LS2导通。

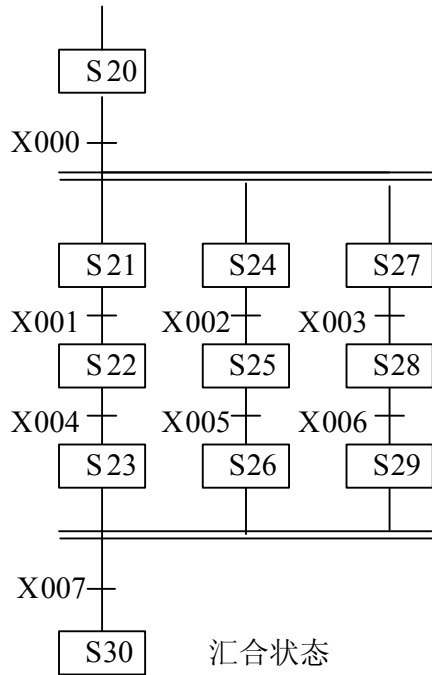


4.步进梯形图指令说明

像这种大小分类选择或判别合格与否的 SFC 图，可用下图所示的选择分支与汇合的 SFC 图表示。



12 并行分支与汇合流程示例

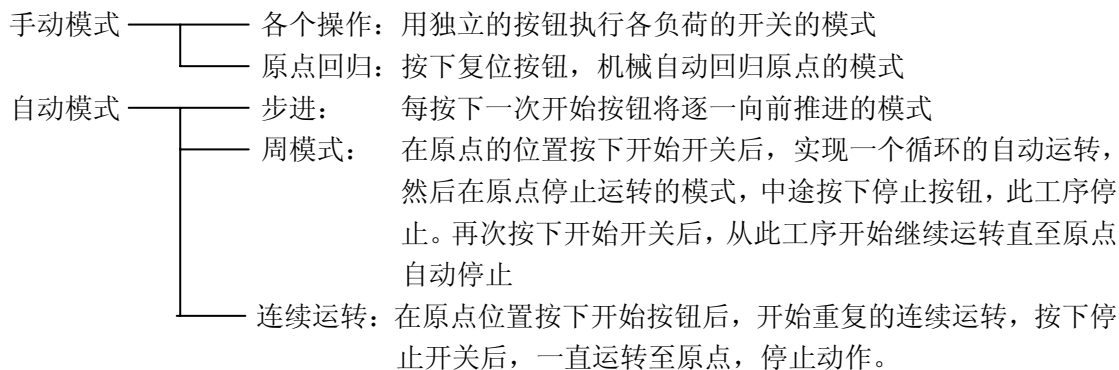


- 多个流程都在同时执行的分支被称为并行分支。
- 以上图为例，S20 动作后，X0 ON，状态 S21,S24,S27 会同时有效，各流程动作开始。
- 各流程动作全部完成后，当 X07 接通时，汇合状态 S30 动作，而状态 S23,S26,S29 全部不动作。
- 这种汇合，又称为等待汇合。(先完成的流程要等所有流程动作结束后，再汇合后再继续动作。)

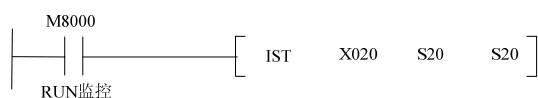
例如：将零件 A、B、C 分别并行加工，零件加工后进行装配，这也是并行型分支与汇合流程。

13 初始化状态(F60 IST)命令的活用

机械的逆转模式一般如下所示，使用其中一部分或全部模式。



通常，此类控制可通过编制步进梯形图指令（SFC 流程）的程序来实现。在 BSP01 系列 PLC 的应用指令中，有简单地控制此类机械的固定的方便指令。



应用指令 F60 (IST)，是对于上述运作模式中的状态或特殊辅助继电器实施自动控制的一整套命令。

通过使用 IST 命令，不再需要各运作模式间的切换或重复控制程序，只要将重点放在编制状态内的机械动作的程序上，就可完成顺序设计。

有关此命令的详细情况请参照[应用指令说明]的 F60 (IST) 命令。

5. 应用指令的表示与执行形式

第五章 应用指令的表示与使用.....	98
1 应用命令内的表示与执行形式.....	98
2 应用命令内的数值使用.....	103
3 利用变址寄存器的操作数修改.....	105
4 常数 K、H、E（10 进制/16 进制/实数）的指定.....	107

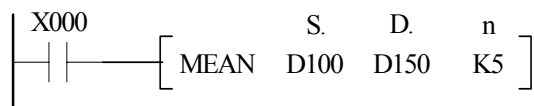
第五章 应用指令的表示与使用

本章主要介绍了 BSP01 系列 PLC 的应用指令及其编程方法。一般，一条基本指令只完成一个特定的操作，而一条应用指令却能完成一系列的操作，相当于执行一个子程序，所以应用指令的功能更为强大。基本指令和其梯形图符号之间是相互对应的。而应用指令采用梯形图助记符相结合的形式，意在表达本指令要作什么。应用指令在整个程序中的使用次数也是有限制的。

1 应用命令内的表示与执行形式

指令与操作数

- PLC 的应用指令可指定其功能编号 F00-F 口口口，而各指令也可以选用助记符来表示，例如 F45 是 MEAN，表示“求平均值”。
- 应用指令由其功能 F 编号和后面的助记符来构成一完整指令；



- MEAN 指令的助记符，表示求平均值指令
- S: 源操作数，简称源，指令执行后不改变其内容的操作数。在可利用变址修改软元件标号的情况下，用加上“·”[S·]表示，当操作数不止一个时，用[S1·]、[S2·]等来表示。
- D: 目标操作数，简称目，指令执行后将改变其内容的操作数。同源一样，也可以做变址修饰；当目标操作数数量多时，用[D1·]、[D2·]等来表示。
- m, n: 其他操作数，常用来表示常数或对源和目作出补充说明。表示常数时，K 后面跟十进制数，H 后跟的为十六进制数。
- 程序步 指令执行所需的步数，一般来说，功能指令的功能号和助记符占一步，每个操作数占 2~4 步（16 位操作数是 2 步，32 位操作数是 4 步）。

操作数的可用软元件

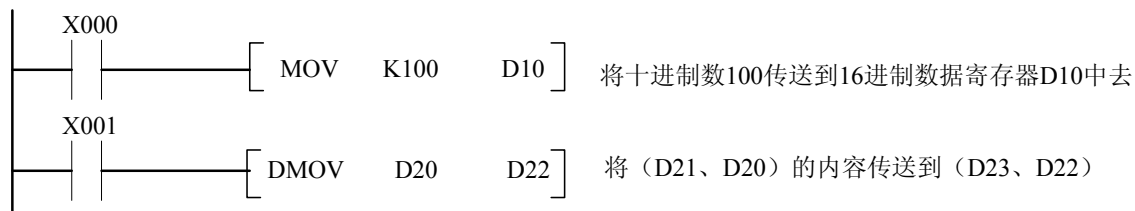
- 可使用 X,Y,M,S 等位组件。
- 将这些位元件组合，用 KnX, KnY, KnM, KnS 等形式表示，作为数值数据进行处理。请参考后述的「位元件的使用」
- 可处理数据缓存器 D、补充数据缓冲器 W 或定时器 T 及计数器 C 的当前值寄存器。数据寄存器 D 为 16 位，在处理 32 位数据时使用一对的数据寄存器来组合。例如：将数据寄存器 D0 指定为 32 位指令的操作数时，处理 (D0、D1) 32 位数据 ((D1 为高 16 位，D0 为低 16 位)。T,C 的当前寄存器也可以作为一般寄存器处理；但 C200~C255 的 32 位计数器的 1 点可处理 32 位的数据，不能指定为 16 位指令的操作数使用。

5. 应用指令的表示与执行形式

指令的形态与执行形式

在 BSP01 系列 PLC 中，会根据处理数值的大小，将应用指令分为「16 位指令」和「32 位指令」，此外，根据指令各自的执行形式，有「连续执行型」和「脉波执行型」等特点。应用指令可以将这些形式组合使用或单独使用。

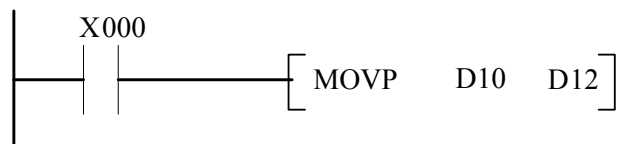
- 在数值处理的应用指令中，根据数值数据的位长分为 16 位和 32 位。



- 凡是能前缀显示符号 (D) 的功能指令，就能处理 32 位数据；
- 32 位数据是由 2 个相邻寄存器构成；
- 32 位的计数器(C200-C255)的一个软元件为 32 位，不可作为 16 位的操作数使用。

脉冲执行/连续执行指令

脉冲执行型



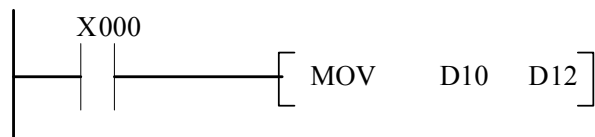
在脉冲执行方式下，指令 MOV 只在条件 X000 从 OFF→ON 才执行一次数据传送操作，为缩短扫描时间，因此尽可能使用脉冲执行指令。

符号 P 是表示这条指令还有脉冲执行方式。

当 F24(INC)、F25(DEC)等根据其指令的内容，如果采用连续执行型指令，则每个扫描周期，其源的内容都变化。

注意：脉冲执行型指令在第一个扫描周期不执行。

连续执行型



上图为连续执行型指令，PLC 是以循环扫描方式工作的，如果执行条件 X001 接通，上述指令在每一个扫描周期中都被重复执行一次。

- 驱动输入 X0 与 X1 断开时，指令都不执行，特殊符号标记以外的指令，目标也不变化。

标志的处理

《一般标志》

- 根据应用指令的种类，有下述的标志会动作。

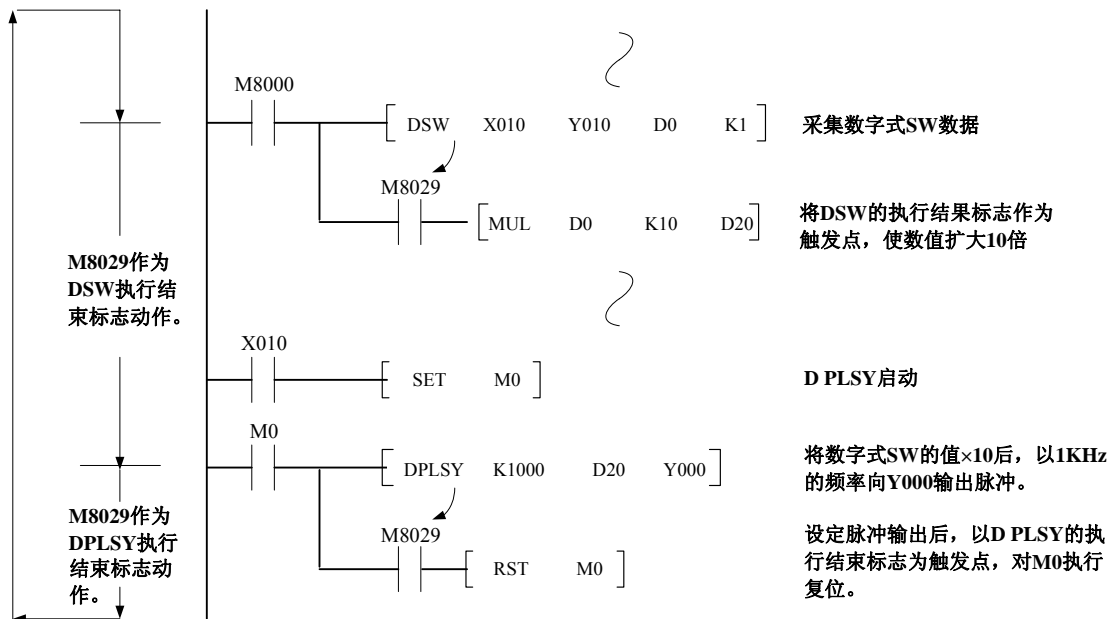
(例) M8020: 零标识 M8022: 进位标识
 M8021: 借位标识 M8029: 执行结束

这些标志在各种指令 ON 时，会随之接通或断开，但在 OFF 时，或出现错误时不变化。

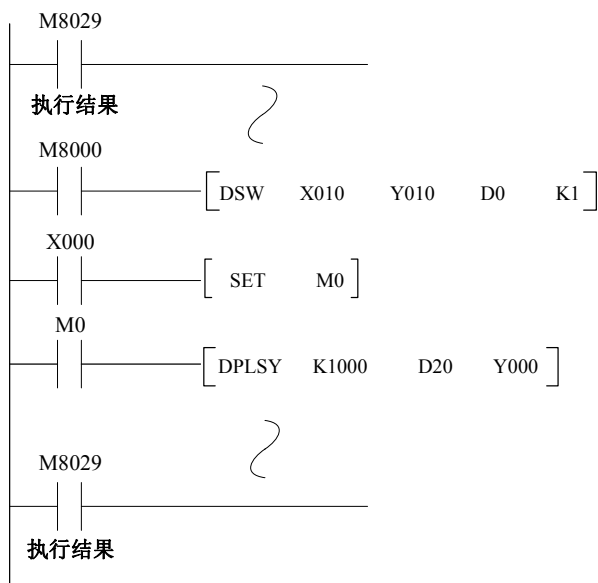
在影响这种标志的指令数量很多时，每次执行这些指令时，接通 / 断开状态都有变化，请参照下述所示的标志程序示例。

- 多个标志的程序示例（执行结果标示例）。

使用同一个标志动作的应用指令有多个时，请在各指令的后面编写标志接点。



5. 应用指令的表示与执行形式



如左图所示，对于使同一个标志位动作的多个应用指令，其执行完成标志。如果被总结成一个，进行程序编写，那样将不容易判断。究竟是以哪个指令的执行内容来进行标志位控制，同时也无法获得对应每条指令的标志位。

《运算出错标志》

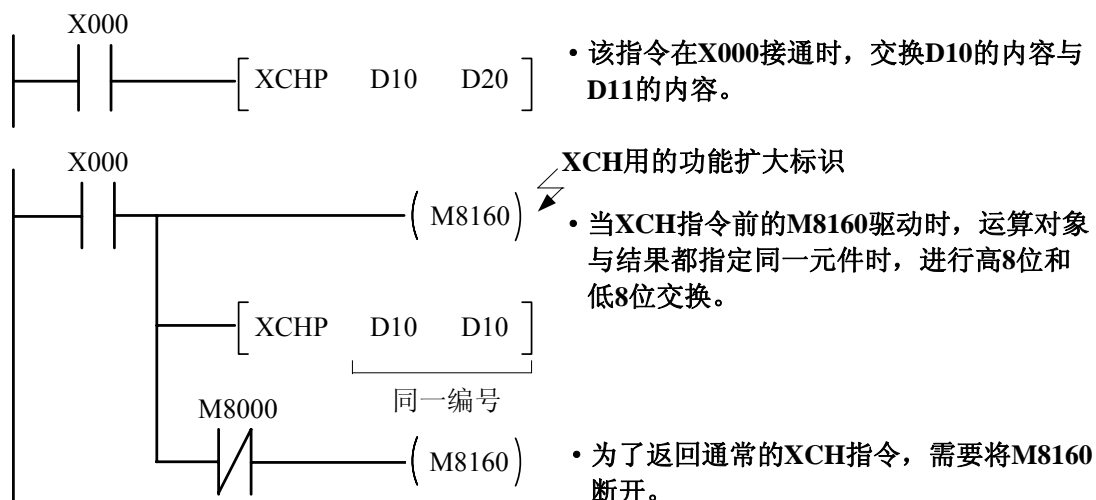
如果在应用指令的结构、可用软元件及其编号范围等方面有错误时，在运算执行过程中会出现错误，下列标志位会动作并会记录出错信息。

M8067
D8067

运算异常发生时，M8067 动作并保持，在 D8067 存入异常码，
 当有其它新异常发生时，则 D8067 会更新为新的异常码
 当异常解除时，M8067 变为 OFF，D8067 在 PLC 从 STOP→RUN 时清除为 0

《扩展功能用标志》

在部分应用指令中，同时使用由该应用指令确定的固有特殊辅助继电器，可进行功能扩展，以下例说明：



此外，在中断程序中使用的指令需要功能扩展标志时，请在功能扩展标志驱动前，编写 DI 指令(中断禁止)，在功能扩展标志断开后编写 EI 指令。(中断许可)

指令的同时驱动限制

在应用指令中，即使一些指令能多次编程，也有同时动作点数的限制。

《6 项指令以下》

F53 (DHSCS), F54 (DHSCR), F55 (DHSZ)

《2 项指令以下》

F72(DSW), F74(SEGL)

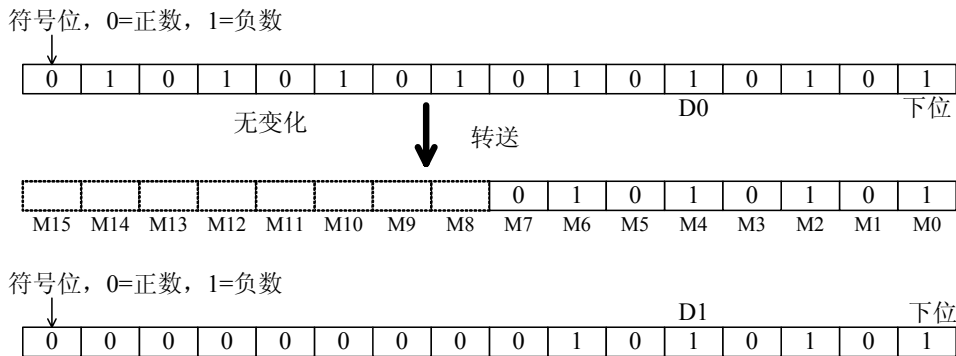
《1 项指令以下》

F52(MTR), F57 (PLSY), F58(PWM), F59 (PLSR), F60(IST), F62(ABSD), F68(ROTC),
F69(SORT), F70(TKY), F71(HKY), F75(ARWS), F80 (RS), F87 (MBUS), F156 (ZRN),
F157 (PLSV), F158 (DRVI), F159 (DRVA), F190(DTLK), F191(RMIO) , F193(DTLK2)。

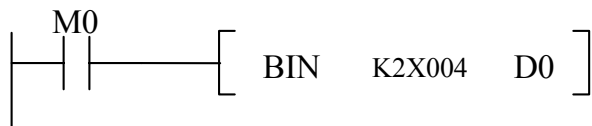
2 应用命令内的数值使用

位组件的使用

- 像 X、Y、M、S 只有 ON 或 OFF 两种状态，用一个二进制位就能表达的组件，称为位组件，而 T、C、D、W 等处理数值的软元件被称为字元件，即使是位元件，通过组合使用也可以处理数值，在这种情况下，以位数 Kn 和起始的软元件号的组合来表示。
- 采用 4 位为单位，位数成为 K1-K4(16 位数据), K1-K8(32 位数据)，
例如 K2M0，为 M0~M7，2 位数的资料。



- 16 位数据转送到 K1M0~K3M0 的指定时，高位数据(即最高 4 个位)因数据长度指定不足而不转送；32 位数据时亦相同。
- 16 位(或 32 位)运算中，位组件对于指定位数 K1-K3(K1-K7)时，而高位不足部分均补 0 处理，此时最高位符号为 0，所以数据作为正数处理。



- 被指定的位元件编号可以自由指定，但建议指定在 X、Y 的场合，最低位编号尽可能设定为 0(X000、X010、X020、...Y000、Y010、Y020)，使用 M、S 的场合，设定数为 8 的倍数最为理想，为避免造成混乱，尽量设定为 M0,M10,M20...等。

附记

<连续字符的指定>

以 D1 为起始的一系列数据寄存器就是 D1、D2、D3、D4.....等。

通过位指定，在字的场合，也可将其作为一系列的位处理。如下所示。

K1X000 K1X004 K1X010 K1X014....., K2Y010 K2Y020 K2Y030

K3M0 K3M12 K3M24 K3M36....., K4S16 K4S32 K4S48

也就是说，不要跳过软元件，按照各位的单元，如上述使用软元件。

但是，若在 32 位运算时，使用 K4Y000，则上位 16 位都为 0。在需要 32 位数据时，请指定 K8Y000。

5. 应用指令的表示与执行形式

浮点小数运算的使用

在 PLC 内部使用的是采用二进制的整数。

在整数的除法中，例如 $40/3=13$ 余 1 的答案。

在整数的开方运算中，舍去小数点。

BSP01 系列 PLC 中，为更精确地进行这些运算，采用浮点数运算。

- 浮点数运算功能对以下指令有效。

F49(FLT), F110(DECMP), F111(DEZCP), F118(DEBCD),

F119D(EBIN), F120(DEADD), F121(DESUB), F122(DEMUL),

F123D(EDIV), F127(DESQR), F129(INT)

<10 进制浮点数>

- 二进制浮点数是用户难于判断的数值，因此可将其变为十进制浮点值。
- 利用编号连续的一对数据寄存器处理十进制浮点值，编号小的一侧为尾数段，编号大的一侧为指数段。

例如，使用数据寄存器(D1,D0)时，由 MOV 指令将数据写入 D0,D1。

10 进制浮点值=「尾数 D0」 $\times 10^{\text{[指数D1]}}$

尾数 D0=(1,000~9,999)或 0

指数 D1= -41 ~ +35

D0、D1 的最高位为正负符号的位，都作为 2 的补码处理。

此外，在尾数 D0 中，例如 100 并不存在。在为 100 的场合，变为 1000×10^{-1} （尾数为：1000，指数为：-1）。10 进制浮点数的处理范围如下所示。

最小绝对值 1175×10^{-41} 最大绝对值 3402×10^{35}

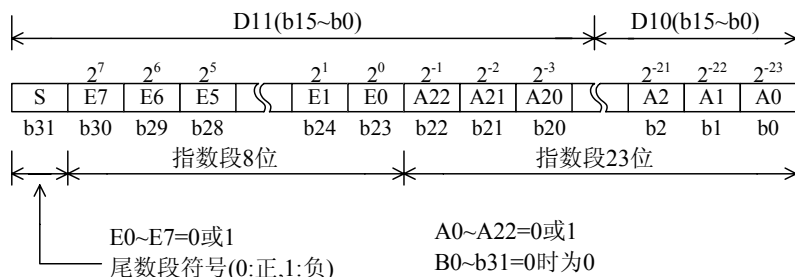
- 10 进制浮点值在下述指令中有效。

2 进制浮点值→10 进制浮点值变换：F118(DEBCD)

10 进制浮点值→2 进制浮点值变换：F119(EBIN)

《2 进制浮点值》

2 进制浮点值是使用一对编号连续的数据寄存器，例如(D11, D10)的场合，结果如下：



正负符号是根据 b31 来决定，不能使用补码。

- 零标志(M8020)，借位标志(M8021)，进位标志(M8022)的使用，浮点运算的各种标志动作如下所示。

零标志：其结果为 0 时为 1。

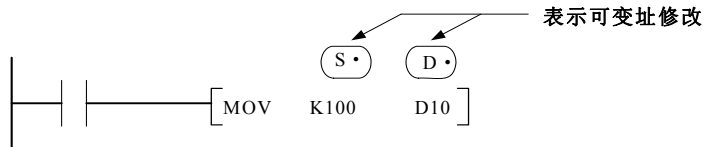
负标志：其结果不是最小值且非为 0 时为 1。

进位标志：其结果超出绝对值可以使用范围时为 1。

3 利用变址寄存器的操作数修改

可使用的应用命令

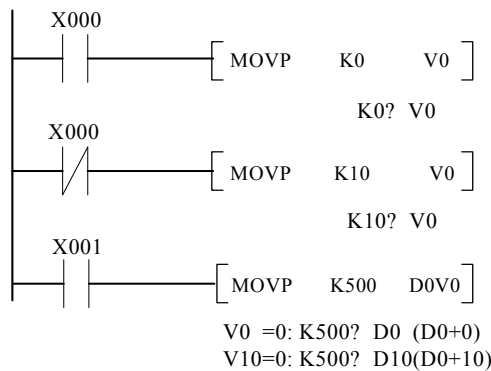
在应用指令的说明中，表示变址修饰的操作数的方法，如下图所示，在表示源 S 及目标 D 的符号中加上「·」记号，以便能与不带修改功能的操作数进行区分。



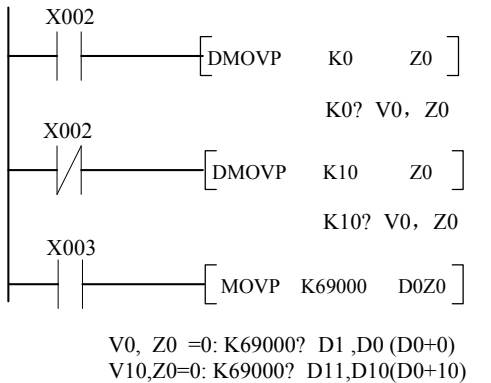
变址修改示例

关于变址寄存器的结构和功能在「2-9-2 变址寄存器」中有详细介绍，敬请查阅。

《数据寄存器编号修饰》



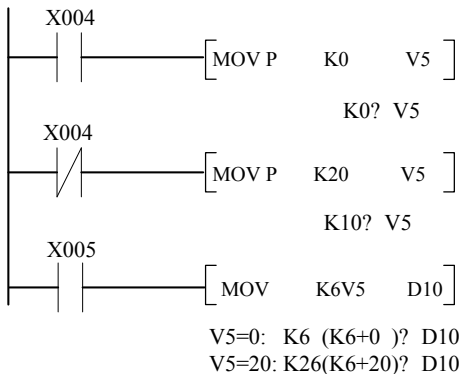
• 16位指令操作数的修改
 将K0或K10的内容向变址寄存器V0传送。
 X001接通，当V0=0时(D0+0=D0)，则
 K500的内容向D0传送。若V0=10时
 [(D0+10)=D10]，则K500的内容向D10传送



• 32位指令操作数的修改
 因为DMOV指令是32位的指令，因此在该指令中使用的变址寄存器有必要指定为32位。在32位指令中指定了变址寄存器Z侧(Z0~Z15)时，包含了与此组合的V侧(V0~V15)，将它们作为32位寄存器动作。

*1:即使Z0中写入的数值不超过16位的数值范围(0~32,767)，也必须用32位的指令将V,Z两方改写。如果只写入Z侧，则在V侧留有其他数值，会使数值产生很大运算错误。

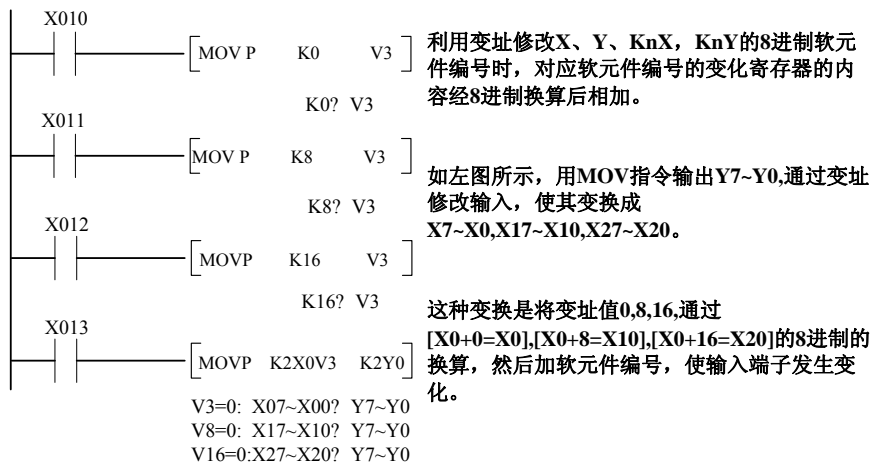
《常数 K 的修饰》



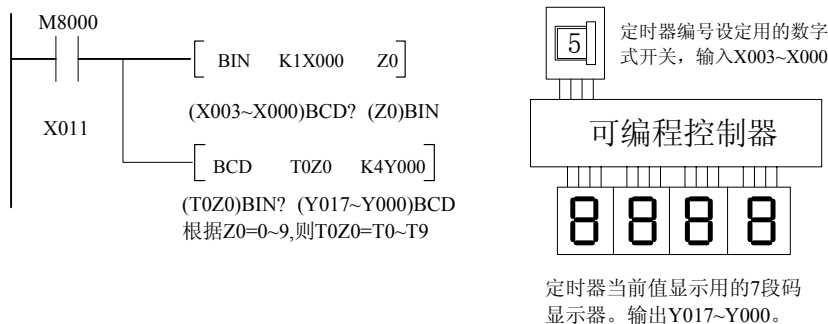
常数的情况也同软元件编号的修改一样。
 X005为ON，如果V5=0时，[K6+0=K6]，
 将K6的内容向D10传送。
 如果V5=20时，[K6+20=K26]，将K26的
 内容向D10传送。

5. 应用指令的表示与执行形式

《输入/输出继电器(8 进位软元件编号)的修改》

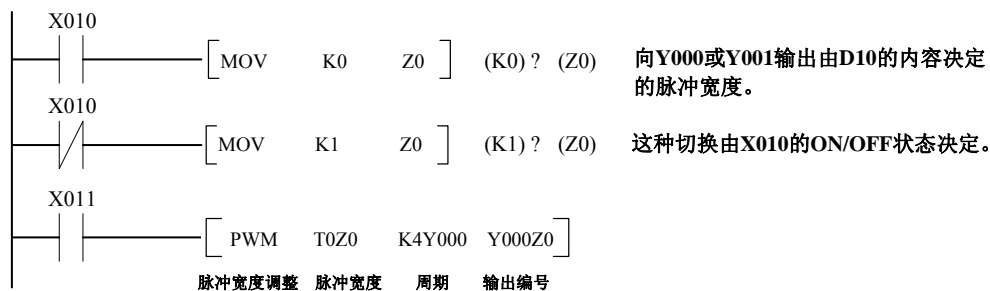


《定时器当前值显示范例》



《使用次数受限制的指令修饰》

将对象组件编号以索引寄存器修饰，则可藉由程序将对象组件编号作变更。针对使用次数受限制的指令来说，使用此方法就和多次编辑同一指令有同样的效果。



F58 指令只能执行一次编程的指令，但在没有必要同时驱动多个输出的情况下，可用修改输出编号的方法来变更被控制的对象。

此外，在指令执行中即便 Z 是有变化，上述的切换动作无效。为使切换变更有效，请将驱动指令的条件置为 OFF 一次。

注意事项

- 利用变址修改的 16 位计数器不能作为 32 位计数器使用，作为变址修改的结果，需要使用 32 位计数器的场合，请在计数器 C200 以后附加上 Z0~Z15。
- V,Z 自身或位指定用 Kn 的“n”不可修改。(K4M0Z0 有效，K0Z0M0 则无效)
- 无法对 LD,AND,OUT 等 PLC 基本控制指令及步进梯形图指令进行变址修改。

4 常数 K、H、E（10 进制/16 进制/实数）的指定

顺控程序中处理常数时，使用常数 K（10 进制数）、常数 H（16 进制数）或 E（浮点数）。在编程用的外围设备中，有关指令上的数值操作中，10 进制数的数值中附加 K，16 进制数的数值中附加 H，浮点数（实数）的数值中附加 E 后输入。（例如：10 进制数…K100，16 进制数…H64，实数…E1.23 或是 E1.23+10）。

其作用和功能如下所示。

常数 K（10 进制数）

“K”是表示 10 进制整数的符号。主要用于指定定时器和计数器的设定值，或是应用指令的操作数中的数值。（例如：K1234）

10 进制常数的指定范围如下所示。

- 使用字数据（16 位）时……………K-32768~K32767
- 使用 2 个字数据（32 位）时………K-2147483648~K2147483647

常数 H（16 进制数）

“H”是表示 16 进制数的符号。主要用于指定应用指令的操作数的数值。（例如 H1234）而且，各位数在 0~9 的范围内使用的时候，各位的状态（1 或 0）和 BCD 代码相同，因此可以指定 BCD 数据。

（例如：H1234 以 BCD 指定数据时，请在 0~9 的范围内指定 16 进制数的个位数）

16 进制常数的设定范围如下所示

- 使用字数据（16 位）时……………H0~HFFFF（BCD 数据的时候为 H0~H9999）
- 使用 2 个字数据（32 位）时………H0~HFFFFFFFF(BCD 数据的时候为 H0~H99999999)

常数 E（实数）

“E”是表示实数（浮点数）的符号。主要用于指定应用指令的操作数的数值。（例如 E1.234 或是 E1.234+3）

实数的制定范围为， $-1.0 \times 2^{128} \sim -1.0 \times 2^{-126}$ ，0， $1.0 \times 2^{-126} \sim 1.0 \times 2^{128}$

在顺控程序中，实数可以指定“普通表示”和“指数表示”两种。

- 普通表示……就将设定的数值指定。
例如，10.2345 就以 E10.2345 指定。
- 指数表示……设定的数值以（数值） $\times 10^n$ 指定。
例如，1234 以 E1.234+3 指定。
“E1.234+3”的“+3”表示 10 的 n 此方（+3 为 10^3 ）。

第六章 应用指令说明

1 应用指令一览表

应用指令的种类如下表所示：《按功能事情顺序排列》

分类	应用指令			16/32 Bit	P	步数	
	指令号	符号	指令功能			16bit	32bit
程序流程	00	CJ	条件跳转	16	√	3	—
	01	CALL	子程序调用	16	√	3	—
	02	SRET	子程序返回	16		1	—
	03	IRET	中断返回	*1		1	—
	04	EI	中断许可	*1		1	—
	05	DI	中断禁止	*1	√	1	—
	06	FEND	主程序结束	*1		1	—
	07	WDT	看门狗定时器	*1	√	1	—
	08	FOR	循环回路开始	*1		3	—
	09	NEXT	循环回路结束	*1		1	—
传送与 比较	10	CMP	比较	16/ 32	√	7	13
	11	ZCP	区间比较	16/ 32	√	9	17
	12	MOV	传送	16/ 32	√	5	9
			将常数传送到存储器, 一字				
			将常数传送到存储器, 二字/ 存储器传送到存储器, 一字				
			存储器传送到存储器, 二字				
			非字节或字/特殊数据范围				
	13	SMOV	移位传送	16	√	11	—
	14	CML	倒转传送	16/ 32	√	5	9
	15	BMOV	块传送	16	√	7	—
16	FMOV	多点传送	16/ 32	√	7	13	
17	XCH	交换	16/ 32	√	5	9	
18	BCD	BCD 转换	16/ 32	√	5	9	
19	BIN	BIN 转换	16/ 32	√	5	9	
四则运算	20	ADD	BIN 加法	16/ 32	√	7	13
	21	SUB	BIN 减法	16/ 32	√	7	13
	22	MUL	BIN 乘法	16/ 32	√	7	13
	23	DIV	BIN 除法	16/ 32	√	7	13
	24	INC	BIN 加 1	16/ 32	√	3	5
	25	DEC	BIN 减 1	16/ 32	√	3	5
	26	WAND	逻辑与	16/ 32	√	7	13
	27	WOR	逻辑或	16/ 32	√	7	13
	28	WXOR	逻辑异或	16/ 32	√	7	13
	29	NEG	补码	16/ 32	√	3	5

6. 应用指令解说

分类	应用指令			16/32 Bit	P	步数	
	指令 号	符号	指令功能			16bit	32bit
循环移位	30	ROR	循环右移	16/ 32	√	5	9
	31	ROL	循环左移	16/ 32	√	5	9
	32	RCR	进位循环右移	16/ 32	√	5	9
	33	RCL	进位循环左移	16/ 32	√	5	9
	34	SFTR	位右移	16	√	9	——
	35	SFTL	位左移	16	√	9	——
	36	WSFR	字右移	16	√	9	——
	37	WSFL	字左移	16	√	9	——
	38	SFWR	移位写入	16	√	7	——
	39	SFRD	移位读出	16	√	7	——
数据处理	40	ZRST	批次复归	16	√	5	——
	41	DECO	译码	16	√	7	——
	42	ENCO	编码	16	√	7	——
	43	SUM	ON 位数	16/ 32	√	5	9
	44	BON	检查特定位的状态	16/ 32	√	7	13
	45	MEAN	平均值	16/ 32	√	7	13
	46	ANS	信号警报置位	16		7	——
	47	ANR	信号警报复位	16	√	1	——
	48	SQR	开方	16/ 32	√	5	9
	49	FLT	BIN 整数—浮点数	16/ 32	√	5	9
高速处理	50	REF	输入输出刷新	16	√	5	——
	52	MTR	矩阵输入	16		9	——
	53	HSCS	高速计数置位	32		——	13
	54	HSCR	高速计数复位	32		——	13
	55	HSZ	高速计数区间比较	32		——	17
	56	SPD	脉冲密度	16		7	——
	57	PLSY	脉冲输出	16/32		7	13
	58	PWM	脉冲调整	16		7	——
	59	PLSR	带加减速的脉冲输出	16/32		9	17
	方便指令	60	IST	初始化状态	16		7
61		SER	数据查找	16/32		9	17
62		ABSD	凸轮控制绝对方式	16/32		9	17
63		INCD	凸轮控制增量方式	16		9	——
64		TTMR	示教定时器	16		5	——
65		STMR	特殊定时器	16		7	——
66		ALT	交替输出	16	√	3	——
67		RAMP	斜坡信号	16		9	——
68		ROTC	旋转工作台控制	16		9	——
69		SORT	数据排列	16		11	——

6. 应用指令解说

分类	应用指令			16/32 Bit	P	步数	
	指令 号	符号	指令功能			16bit	32bit
外围设备 输入输出	70	TKY	数字键输入	16/32		7	13
	71	HKY	16 键输入	16/32		9	17
	72	DSW	数字式开关	16		9	——
	73	SEGD	7 段码解码	16	√	5	——
	74	SEGL	7 段码按时间分割显示	16		7	——
	75	ARWS	箭头开关	16		9	——
	76	ASC	ASCII 码	16		11	——
	77	PR	ASCII 码打印输出	16		5	——
外围设备 SER	80	RS	串行数据传送	16		9	11
	82	ASIC	HEX-ASCII 转换	16	√	7	——
	83	HEX	ASCII-HEX 转换	16	√	7	——
	84	CCD	校验码	16	√	7	——
	85	VRRD	电位器值读出	16	√	5	——
	86	VRSC	电位器刻度	16	√	5	——
	87	MBUS	MODBUS	16		11	——
	88	PID	PID 控制回路	16		9	——
9 浮点运算	110	ECMP	浮点比较	32	√	——	13
	111	EZCP	浮点区间比较	32	√	——	17
	118	EBCD	2 进制浮点-10 进制浮点转换	32	√	——	9
	119	EBIN	10 进制浮点-2 进制浮点转换	32	√	——	9
	120	EADD	浮点加法	32	√	——	13
	121	ESUB	浮点减法	32	√	——	13
	122	EMUL	浮点乘法	32	√	——	13
	123	EDIV	浮点除法	32	√	——	13
	127	ESQR	浮点开方	32	√	——	9
	129	INT	2 进制浮点-BIN 整数转换	16/32	√	5	9
	130	SIN	正弦	32	√	——	9
	131	COS	余弦	32	√	——	9
	132	TAN	正切	32	√	——	9
	133	ASIN	反正弦	32	√	——	9
	134	ACOS	反余弦	32	√	——	9
	135	ATAN	反正切	32	√	——	9
	136	RAD	弧度运算	32	√	——	9
	137	DEG	浮点弧度→角度	32	√	——	9
147	SWAP	上下字节变换	16/32	√	3	5	
定位	156	ZRN	原点回归	16/32		9	17
	157	PLSY	可变度的脉冲输出	16/32		7	13
	158	DRVI	相对定位	16/32		9	17
	159	DRVA	绝对定位	16/32		9	17

6. 应用指令解说

分类	应用指令			16/32 Bit	P	步数	
	指令号	符号	指令功能			16bit	32bit
时钟连算	160	TCMP	时钟数据比较	16	√	11	—
	161	TZCP	时钟区间比较	16	√	9	—
	162	TADD	时钟数据加法	16	√	7	—
	163	TSUB	时钟数据减法	16	√	7	—
	166	TRD	读 RTC 数据	16	√	3	—
	167	TWR	设置 RTC 数据	16	√	3	—
外围设备	170	GRY	10 进制-格雷码变换	16/32	√	5	9
	171	GBIN	格雷码- 10 进制变换	16/32	√	5	9
运算	172	MATH	四则运算	16/32	√	5	9
	173	EMATH	浮点数四则运算	32	√	—	9
外围通信	188	CRC	循环冗余码校验	16	√	7	—
	190	DTLK	Data Link	16		3	—
	191	RMIO	Remote I/O	16		3	—
	193	DTLK2	Data Link2	16		7	—
接点比较	224	LD	(S1)=(S2)	16/32		5	9
	225		LD (S1)>(S2)	16/32		5	9
	226		LD (S1)<(S2)	16/32		5	9
	228		LD (S1)≠(S2)	16/32		5	9
	229		LD (S1)≡(S2)	16/32		5	9
	230		LD (S1)≡(S2)	16/32		5	9
	232		AND (S1)=(S2)	16/32		5	9
	233		AND (S1)>(S2)	16/32		5	9
	234		AND (S1)<(S2)	16/32		5	9
	236		AND (S1)≠(S2)	16/32		5	9
	237		AND (S1)≡(S2)	16/32		5	9
	238		AND (S1)≡(S2)	16/32		5	9
	240		OR (S1)=(S2)	16/32		5	9
	241		OR (S1)>(S2)	16/32		5	9
	242		OR (S1)<(S2)	16/32		5	9
	244		OR (S1)≠(S2)	16/32		5	9
	245		OR (S1)≡(S2)	16/32		5	9
	246		OR (S1)≡(S2)	16/32		5	9

F00~F09 程序流程

程序流程

功能编号	助记符	名称	页码
F00	CJ	条件跳跃	1
F01	CALL	调用子程序	3
F02	SRET	子程式返回	3
F03	IRET	中断返回	5
F04	EI	允许中断	5
F05	DI	禁止中断	5
F06	FEND	主程序结束	9
F07	WDT	监视定时器	10
F08	FOR	循环开始	11
F09	NEXT	循环结束	11

6. 应用指令解说

F00 CJ 条件跳跃

F		CJ				条件跳跃														
0				P																
/	位元件				字元件															
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z				

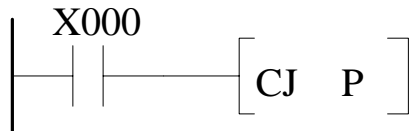
指令格式：



P: 条件跳转的目的标号

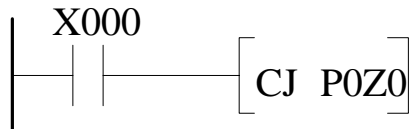
当用户在使用 PLC 时，如有部分程式在运行中不需要每次都执行时，可使用该指令以缩短执行时间。

例：



指针（P）的范围为：P0~P255（BSP01 AR/T 机种），P0~P127（BSP01 SR 机种）。其中 P63 特指 END，请不要对其编程，否则出错。

指针编号可作变址修改。如下图：



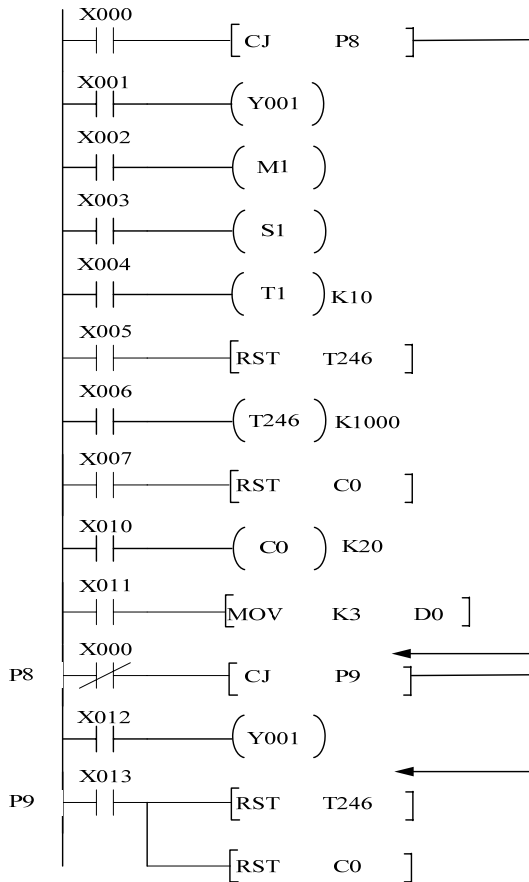
跳至 P (0+ (Z0)) 执行。

条件跳转执行中，各元件动作说明：

1. Y、M、S 保持跳转前的状态；
2. 执行计时中的 10ms,100ms 计时器会暂停计时；
3. 执行计时中的 1ms 计时器 T246~T249 会继续计时，且输出接点正常动作。
4. 执行副程序用的计时器 T192~T199 会继续计时，且输出接点正常动作；
5. 执行计时中的高速计数器会继续计数，且输出接点正常动作；
6. 一般计数器会停止计数；
7. 积算型计数器及计时器的清除指令若在跳转前被驱动，则在跳转执行中该装置仍处于清除状态；
8. 一般应用指令不会执行。
9. 执行中的应用指令 FNC53 (DHSCS)、FNC54 (DHSCR)、FNC55 (DHSZ)、FNC56 (SPD)、FNC57 (PLSY)、FNC58 (PWM) 继续执行。

6. 应用指令解说

示例：



下表为程式跳转过程中，各个元件状态发生变化时的结果：

元件	跳转前的接点状态	跳转中的接点动作	跳转中的线圈动作
Y、M、S	X001、X002、X003 OFF	X001、X002、X003 ON	Y001、M1、S1 OFF
	X001、X002、X003 ON	X001、X002、X003 OFF	Y001、M1、S1 ON
10ms,100ms 计时器	X4 OFF	X4 ON	计时器不动作
	X4 ON	X4 OFF	计时停止，X0 OFF 后继续计时
1ms 计时器	X5、X6 OFF	X6 ON	计时器不动作
	X5 OFF、X6 ON	X6 OFF	计时停止，X0 OFF 后继续计时
计数器	X007、X010 OFF	X010 ON	计数器不计数
	X007 OFF、X010 ON	X010 OFF	计数停止，X0 OFF 后继续计数
应用指令	X011 OFF	X011 ON	应用指令不执行
	X011 ON	X011 OFF	被跳转过的应用指令 不执行

- Y001 变成双线圈，无论在跳转内还是跳转外，都当作一般的双线圈处理。
- 累计定时器计数器的复位指令在跳转外时，计时线圈及计数线圈复位（接点恢复及当

6. 应用指令解说

前值的清除)有效。

6. 应用指令解说

F01 CALL 调用子程序

F 1		CALL	P	调用子程序												
/	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

指令格式：

—— [CALL P]

指针 (P) 的范围为：P0~P255 (BSP01 AR/T 机种)，P0~P127 (BSP01 SR 机种)。其中 P63 特指 END，不能作为 FNC01 (CALL) 的指针。

指针编号可作变址修改。

F02 SRET 子程式返回

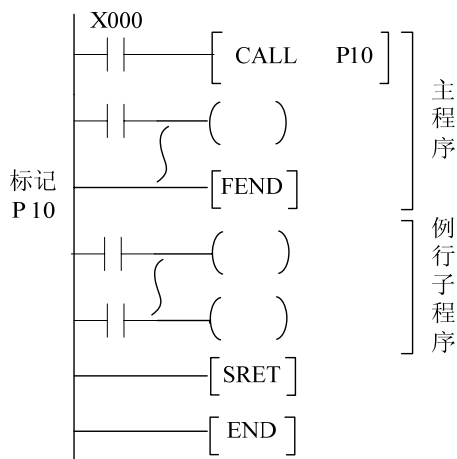
F 2		SRET		子程序返回												
/	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

指令格式：

—— [SRET]

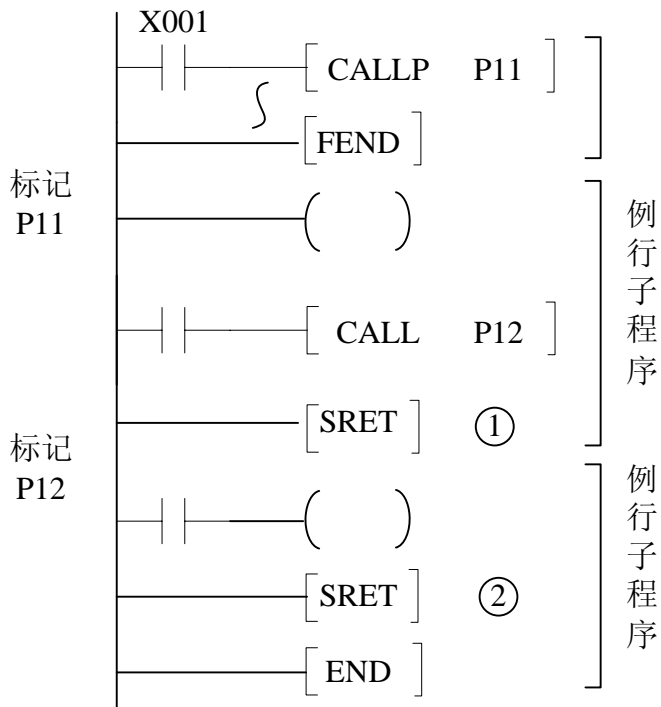
对子程序返回，无适用软元件。

例 1：



- 当 X000 为 ON 时则执行 CALL 命令，跳跃到 P10 执行。在此执行子程序，当执行 SRET 命令时，则回到原步序。
- 而指针程序请在于 FEND 命令后编写。
- 与 CJ 命令共同使用时，请勿使用同一个指标编号 P。

例 2:



- X001=OFF→ON 后，只执行一次 CALL P11 指令，跳跃至 P11。
- 在 P11 的子程序中若执行 CALL P12 的指令，则执行 P12 的子程序，执行完 SRET 指令，则回到 P11 的子程序，再次执行 SRET 指令后回到主程序。
- 嵌套最多 16 层。
- 子程序内的定时器，请采用 T192~T199 或 T246~T249

6. 应用指令解说

F03 IRET 中断返回

F																					
3		IRET																			
		位元件				字元件															
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z				

F04 EI 允许中断

F																					
4		EI																			
		位元件				字元件															
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z				

F05 DI 禁止中断

F																					
5		DI																			
		位元件				字元件															
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z				

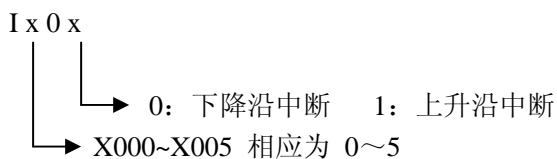
以上三个指令均为不需要驱动触点的单独指令。

- 可编程控制器平时呈中断禁止状态，如果需要中断功能，请使用 FNC4 (EI) 指令使能中断功能。
- 中断用指针 (I***)，必须在 FEND 指令后作为标记编程。
- 中断的类型：
 1. 外部信号输入中断
 2. 定时器中断
 3. 高速计数器中断
 下面依次进行说明。

《外部信号输入中断》

采用输入 X000~X005 的输入信号，执行中断例行程序。因为能够不受控制器的运算周期的影响处理外部输入信号，适用于调整控制及短时间脉冲的拾取。

对应的 6 点中断指针的编号和动作见下表：

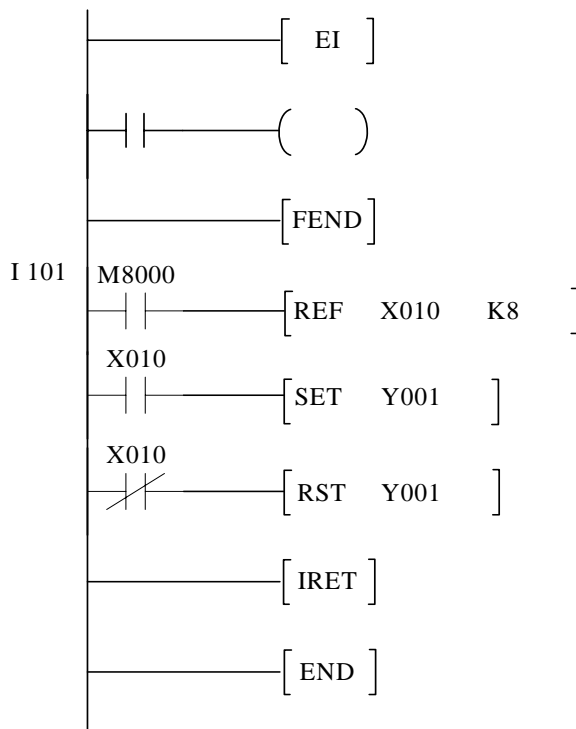


6. 应用指令解说

输入	指针编号		禁止中断指令
	上升沿中断	下降沿中断	
X000	I001	I000	M8050
X001	I101	I100	M8051
X002	I201	I200	M8052
X003	I301	I300	M8053
X004	I401	I400	M8054
X005	I501	I500	M8055

- 指针编号不能重复使用,对同一输入不能同时使用其对应的上升沿中断和下降沿中断编号。
- 如果 M8050~M8055 置 ON, 则禁止对应的输入产生中断。

示例: 采用最新输入信息进行中断处理

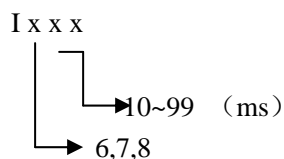


- X001 的上升沿检测,为 ON 则执行中断例行程序,执行输入刷新,根据 X010 的 ON/OFF 状态,设定 Y001 或者复位。

《定时器中断》

不受控制器扫描周期的影响,每隔 10ms~99ms 执行中断子程序。在主程序的运算周期很长的情况下,需要高速处理特定的程序,或者在顺控扫描中,需要每隔一定时间执行程序时,非常适用。

对应的 3 点中断指针的编号和动作见下表:

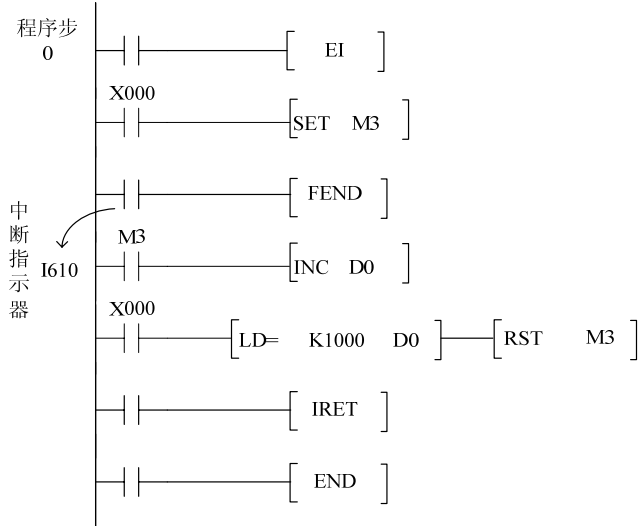


6. 应用指令解说

指针编号	中断周期	禁止中断指令
I6xx	xx: 10~99 的整数, 表示中断间隔时间。	M8056
I7xx		M8057
I8xx		M8058

- 指针编号 (I6,I7,I8) 不能重复使用。
- M8056~M8058 置 ON, 则禁止对应的定时器产生中断。

示例: 每个 10ms 将数据加 1, 并同设定值进行比较。



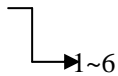
- 每隔 10ms, 使 D0 的当前值加 1。
- 当 D0 的当前值达到 1000 时, M3 复位。

《高速计数器中断》

利用高速计数器的当前值实施的中断, 与 FNC53 (DHSCS) 的比较置位指令并用, 当高速计数器的当前值达到规定值时, 执行中断子程序。

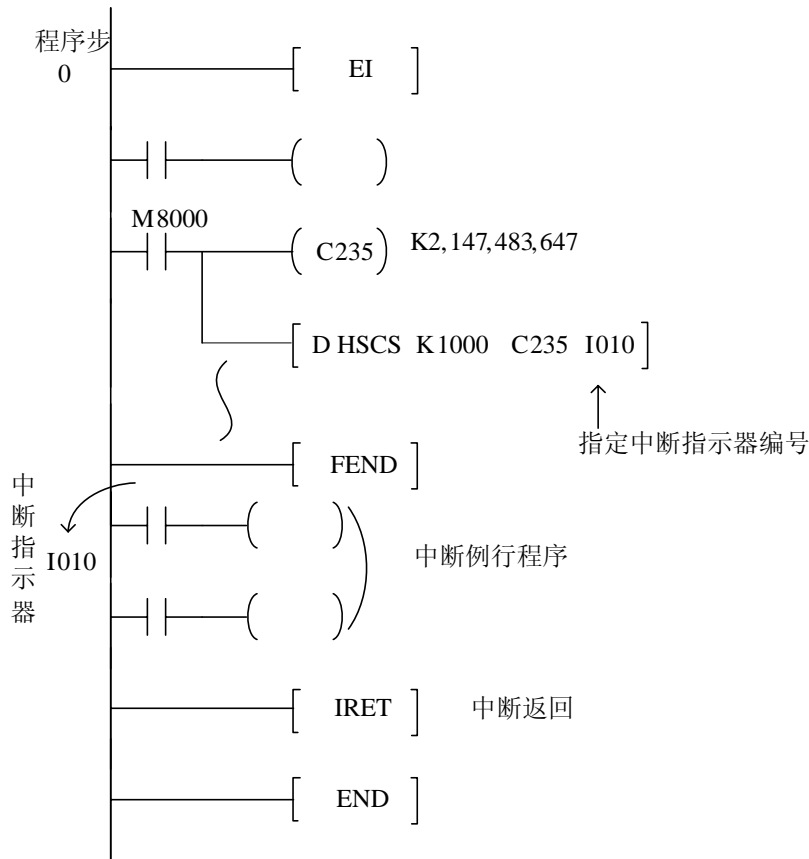
对应的 6 点中断指针的编号和动作见下表:

I0x0



指针编号	禁止中断指令
I010	M8059
I020	
I030	
I040	
I050	
I060	

示例：



- 驱动高速计数器的线圈，在 FNC (DHSCS) 指令中指定中断指针。
- 若 C235 的当前值由 999~1000 变化时，执行中断子程序。
- 对应高速计数器的当前值，若只进行输出继电器或辅助继电器的接通/断开控制，可利用 FNC53 (DHSCS)，FNC54 (DHSCR)，FNC55 (DHSZ) 指令以简化程序。

《多个中断输入》

- 多个中断依次发生时，以先发生的为优先。完全同时发生时，以小的指针编号为优先。
- 在中断例行程序的执行过程中，禁止其它的中断。但是能够保存该期间发生的中断信息，等该例行程序处理完后再执行，至多为 8 个中断。

《输入中断的脉宽》

- 为了用外部信号执行输入中断，需要输入 50us 以上脉宽的 ON 或 OFF 信号。

《输入输出的恢复》

- 在中断处理过程中控制输入继电器及输出继电器时，使用输入输出继电器恢复命令 FNC (REF)，可以通过取得最新的输入信息或立即输出运算结果的方式进行对控制器的运算周期不受影响的高速控制。

《注意事项》

- 作为中断指针采用的输入继电器的编号，请不要与采用相同输入范围的高速计数器及脉冲密度等的应用指令的编号相重复。
- 子程序及中断例行程序内的定时器，请采用例行程序用的定时器 T192~T199。如果采用一般的定时器，除了不能进行计时外，在使用 1ms 累计定时器时亦需加以注

6. 应用指令解说

意。

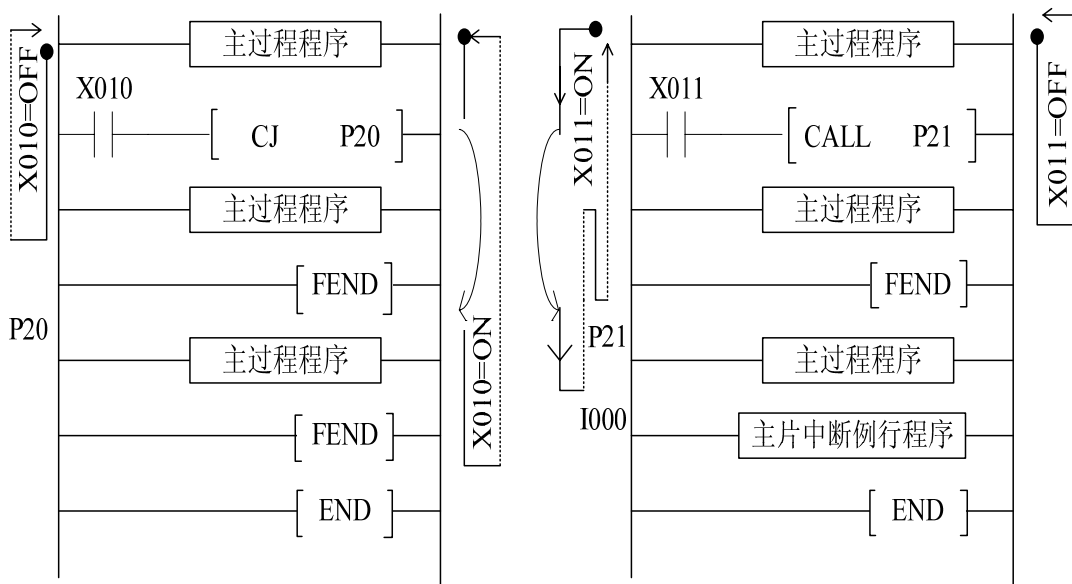
6. 应用指令解说

F06 FEND 主程序结束

F		FEND				主程序结束										
6																
	位元件					字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z

本指令为不需要驱动触点的单独指令，表示主程序的结束。

此指令与 END 指令执行效果相同，执行输出处理、输入处理、监视定时器的刷新、向 0 步程序返回。



- CALL、CALL P 指令的程序必须设计在 FEND 指令后，且要有 SRET 指令结束该副程序。中断程序也必须设计在 FEND 指令后，且要有 IRET 指令结束中断程序。
- 执行 CALL、CALL P 指令后，SRET、IRET 指令执行前；或 FOR 指令执行后，NEXT 的指令执行前，若执行 FEND 指令，则会发生程序异常。
- 若使用多个的 FEND 指令时，请将子程序和中断程序设计于最后的 FEND 和 END 指令间。

6. 应用指令解说

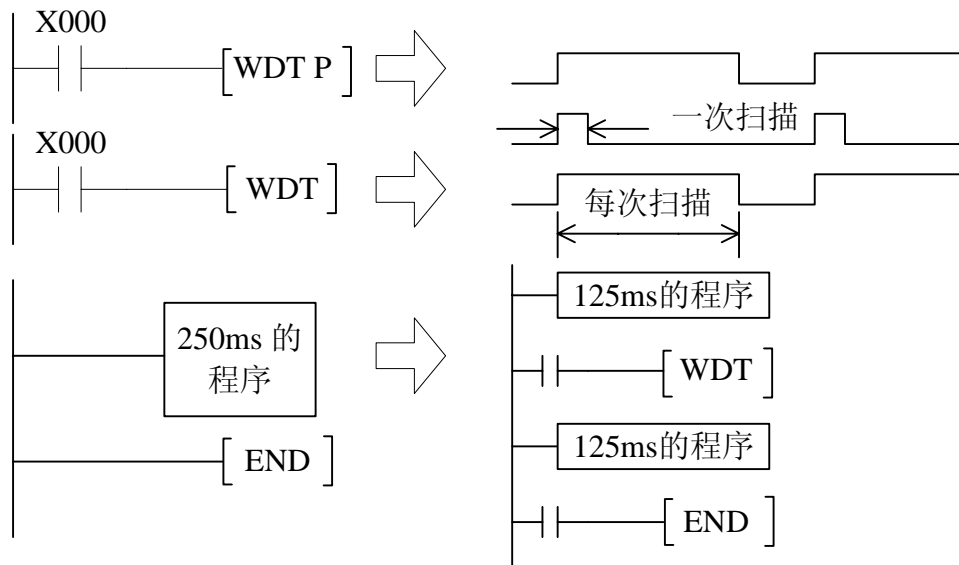
F07 WDT 监视定时器

F		WDT				监视定时器												
1				P														
位元件					字元件													
X	Y	M	S		K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		

本指令用于监视定时器刷新，防止由于扫描周期的延迟导致控制器出错。

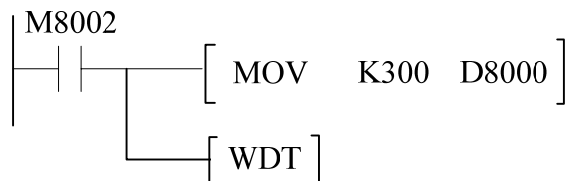
如果扫描时间超过一定的值 PLC 将停止运行。这种情况下，应将 WDT 指令插到合适的程序步中，刷新定时器值。监视定时器的值由 D8000 设定，范围为 200ms~1600ms。

例：



若监视定时器的值设定为 200ms，当程序的扫描时间为 250ms 时，分割 2 份，在此中间放入 WDT，则前半与后半程序都在 200ms 以下。

- 通过改写 D8000 的内容，可改变监视定时器的检测时间，如下图：



- 在系统构成中连接有很多台定位、凸轮开关、ID 界面、链接、模拟量等特殊扩展设备时，控制器 RUN 时所进行的缓冲存储器的初始化时间会延长，从而使扫描时间延迟。此外，当执行多条 FROM/TO 指令，或向多个缓冲存储器传送数据时，其时间也会延长。在这种情形下，便会发生逾时监视定时器异常，此时请在起始步附近输入上述程序以延长监视定时器的时间。

6. 应用指令解说

F08 FOR 循环开始

F 8		FOR		循环开始		S ·												
S ·	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		
					*	*	*	*	*	*	*	*	*	*	*	*	*	*

指令格式：

—— [FOR S ·]

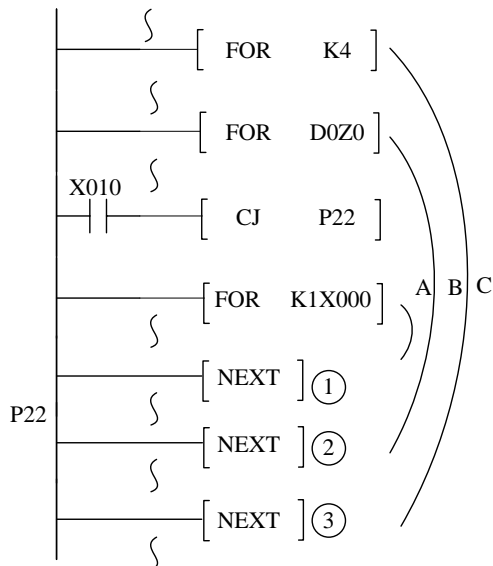
F09 NEXT 循环结束

F 9		NEXT		循环结束														
S ·	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		

本指令为不需要驱动触点的单一指令，与 FNC08 (FOR) 一起使用。

本指令重复执行 FOR~NEXT 指令之间的指令 n 次后(由 FOR 指令指定),才处理 NEXT 指令以后的步。n=1~32,767 时有效, n=-32,767~0 时, 当作 1 处理。

示例：



- [C]的程序执行 4 次后向 NEXT 指令 (3) 以后的程序转移；若在[C]的程序执行一次的过程,寄存器 D0Z0 的内容为 6,则[B]的程序执行 6 次;可以使用 CJ 指令跳过 FOR~NEXT 之间的程序,如 X010=ON 时。
- 嵌套最多可达 16 层, FOR~NEXT 必须成对出现,否则将出错。
- 循环次数多时会造成扫描周期的延长,可能导致监视定时器出错,请引起注意。

F10~F19 数据传输、比较

数据传输、比较

功能编号	助记符	名称	页码
F10	CMP	数据比较	1
F11	ZCP	区域比较	2
F12	MOV	数据传送	3
F13	SMOV	位移动	4
F14	CML	反相传送	5
F15	BMOV	成批传送	6
F16	FMOV	多点传送	7
F17	XCH	交换	8
F18	BCD	BIN→BCD 变换	9
F19	BIN	BCD→BIN 变换	10

6. 应用指令解说

F10 CMP 数据比较

F		CMP		数据比较	S1·	S2·	D·									
10	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·					*	*	*	*	*	*	*	*	*	*	*	*
S2·					*	*	*	*	*	*	*	*	*	*	*	*
D·		*	*	*												

指令格式：

$$\text{---} \left[\text{CMP} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \right]$$

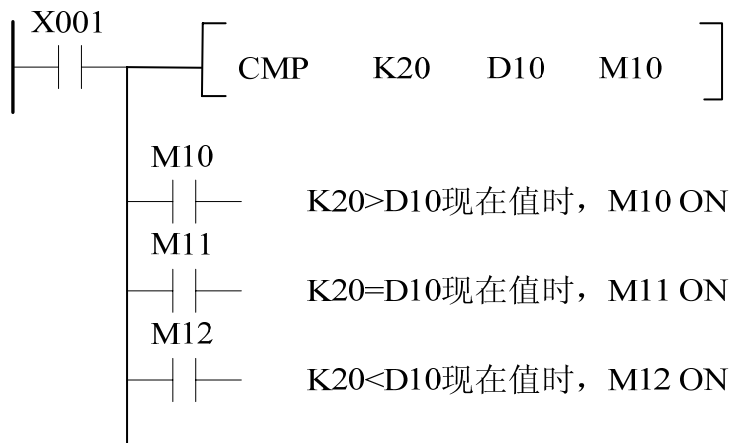
S1·: 比较值 1

S2·: 比较值 2

D·: 比较结果, 占用连续 3 点

将运算元 S1· 和 S2· 的内容作大小比较, 其比较结果放在 D· 中。

例:



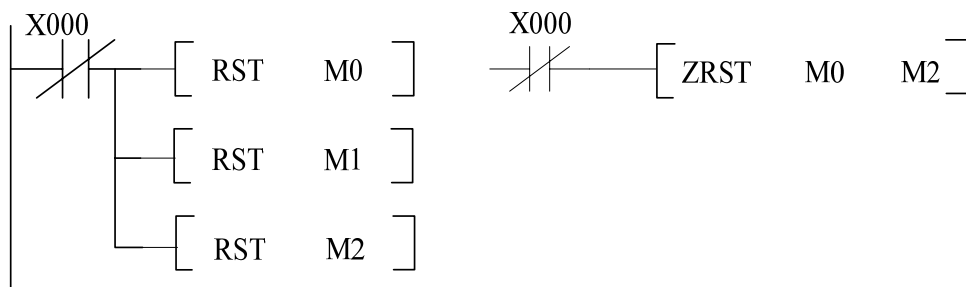
S1/S2· 的数据按有符号整数数比较大小。

目标地址自动占用下两个。如指定 Y001, 则自动占用 Y002, Y003。

32 位指令目标操作数不能指定 V, 只能指定 Z。指定为 Zn 时, (Vn, Zn) 组成 32 为数据。(以下各应用指令如无特殊说明, 皆如此。)

指令不执行时, 目标数据不受影响。

想要清除比较结果, 可用复位指令或全部复位指令:



6. 应用指令解说

F11 ZCP 区域比较

F					数据比较										S1 • S2 • S • D •			
11	D	ZCP	P															
	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		
S1 •					*	*	*	*	*	*	*	*	*	*	*	*		
S2 •					*	*	*	*	*	*	*	*	*	*	*	*		
S •					*	*	*	*	*	*	*	*	*	*	*	*		
D •		*	*	*														

指令格式:

$$\text{——} \left[\text{ZCP} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S1•: 区域比较下限值

S2•: 区域比较上限值

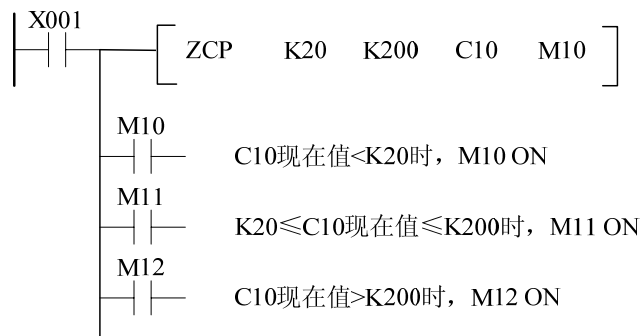
S•: 比较值

D•: 比较结果, 占用连续 3 点

比较值 S•与下限值 S1•及上限值 S2•作比较, 其比较结果放在 D•中。

当下限值 S1•>上限值 S2•时, 则指令以下限值 S1•作为上下限值进行比较。

例:



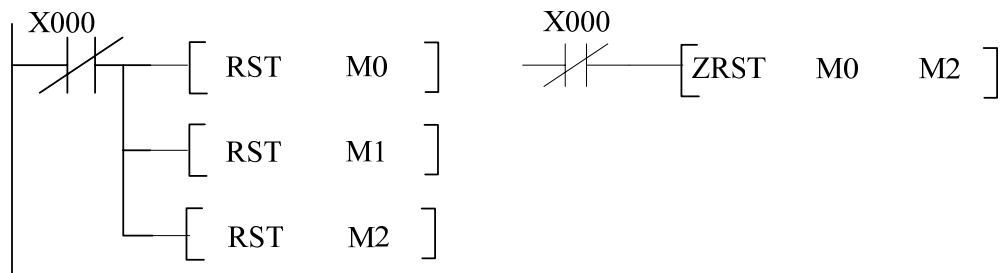
S1•,S2•,S0•的数据按有符号整数数比较大小。

要求 S1•≤S2•。当 S2•<S1•时, 把 S2•看成 S1•进行运算。

目标地址自动占用下两个。如指定 M0,则自动占用 M1,M2。

指令不执行时, 目标数据不受影响。

若要清除结果, 请使用 RST 或 ZRST 指令:



6. 应用指令解说

F12 MOV 数据传送

F		MOV		数据传送	S•	D•										
12	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S•					*	*	*	*	*	*	*	*	*	*	*	*
D•					*	*	*	*	*	*	*	*	*	*	*	*

指令格式：

—— [MOV S• D•]

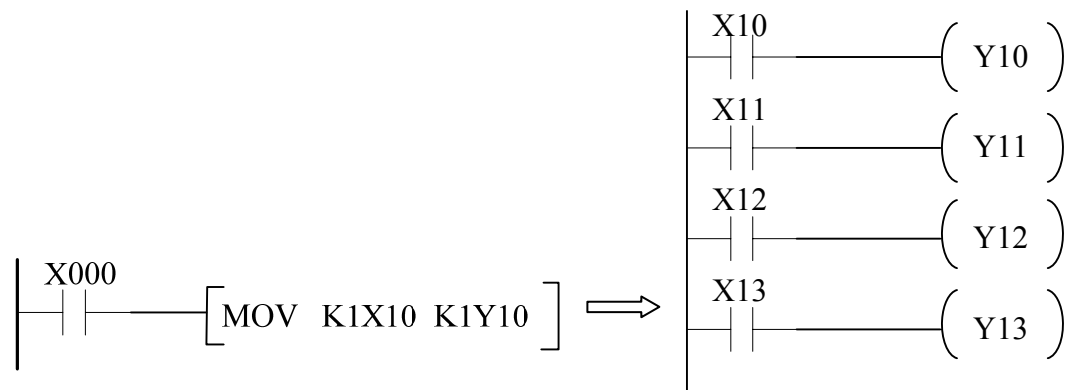
S•：数据来源

D•：数据传送的目标地址

将 S•的内容直接传送至 D•中，当指令不执行时，D•中的内容不发生变化。

32 位指令用 DMOV 指令。操作数自动占用两个字。

16 位元资料传送，当指令执行时，将 X10~X13 四个位元的内容传送至 Y10~Y13 内，功能与下面的程式一样。



6. 应用指令解说

F13 SMOV 位移动

F		SMOV		位移动	S • m1 m2 D • n											
13			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*	*	*
m1					*	*										
m2					*	*										
D •								*	*	*	*	*	*	*	*	*
n					*	*										

指令格式：

—— [SMOV S • m1 m2 D • n]

S•: 数据来源

m1: 数据来源传送起始位数

m2: 数据来源传送的个数

D•: 数据传送的目标地址

n: 传送目的地的起始位数

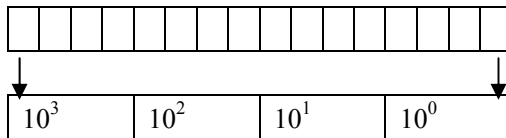
m1/ m2/ n: 1~4

例：

—— [SMOV D10 K4 K2 D20 K3]

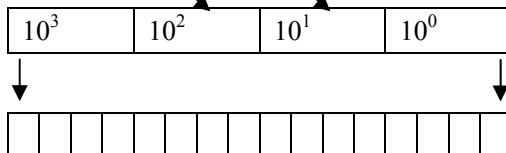
M8168=OFF 时：

D10 的 BIN 码



D10 转换成的 BCD 码

D20 的 BCD 码



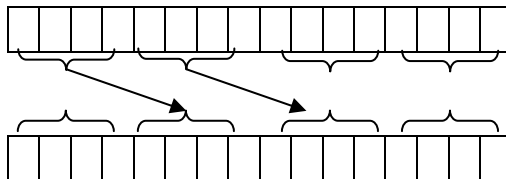
自动转换成 BIN 码



把源数据 D10 的 BCD 转换值从其第 4 位起的低 2 位向目标数据 D20 的第 3 位传送数据。当 D10 的 BCD 值超越 0~9,999 范围则会出错。

M8168=ON 时：

D10 的 BIN 码



D20 的 BIN 码

不进行 BCD 码转换，照原样以 4 位为单位进行位移动。

6. 应用指令解说

F15 BMOV 成批传送

F		BMOV		成批传送	S •	D •	n									
15	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S •							*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*		
n					*	*										

指令格式：

— [BMOV S • D • n]

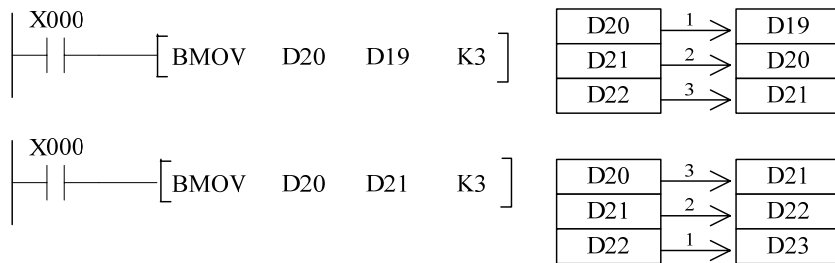
S•：数据来源起始地址

D•：数据传送的目标地址

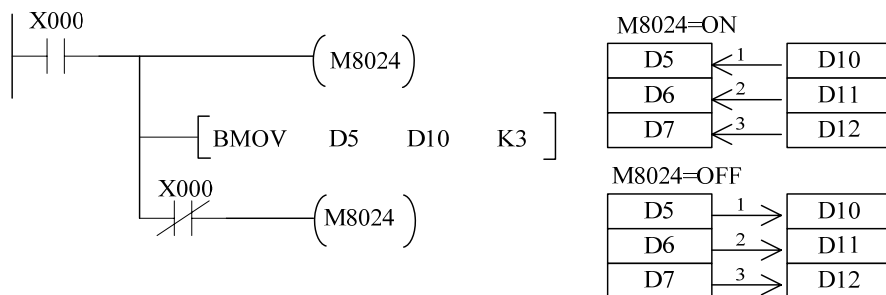
n：传送区块长度 (n≤512)

将以源起始地址指定的软元件为开头的n点数据向以目标地址指定的软元件为开头的n点软元件成批传送。(超过软元件编号范围时，在可能的范围内传送)

如下图传送编号范围有重叠时为了防止输送源数据在传送之前就改写，根据编号重叠的方式，按 1~3 的顺序自动传送。



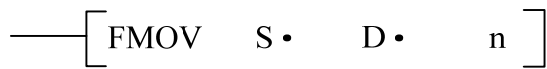
在 M8024 置于 ON 时，执行指令时传送方向反转。



F16 FMOV 多点传送

F		FMOV		多点传送												S •	D •	n	
16	D		P																
	位元件				字元件														
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S •					*	*	*	*	*	*	*	*	*	*	*	*	*		
D •								*	*	*	*	*	*	*					
n					*	*													

指令格式:



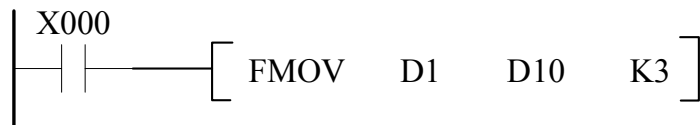
为同一数据的多点传送指令。

S•的内容被传送至 D•所指定的软元件为开头的 n 的软元件中,n 点软元件的内容都一样。超过目标软元件的范围时, 向可能的范围传送。

D•: 数据传送的目标地址,且目标地址不能为 V,Z 寄存器。

n: n<=512

例:



执行的结果: D1----->D10
 D1----->D11
 D1----->D12

6. 应用指令解说

F17 XCH 交换

F		XCH		交换								D1·	D2·			
17	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D1·								*	*	*	*	*	*	*	*	*
D2·								*	*	*	*	*	*	*	*	*

指令格式：

—— [XCH D1· D2·]

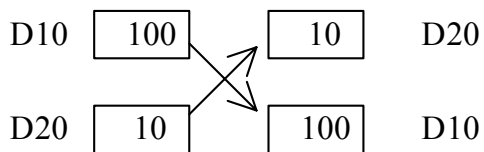
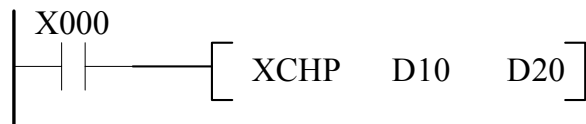
D1·：相互交换的数据 1

D2·：相互交换的数据 2

将 D1·与 D2·所指定的装载的内容值互相交换。

本指令一般都是使用脉冲型指令 XCHP

例：



务必注意，使用连续执行型指令，每个扫描周期都会进行数据交换。

当 M8160=ON，且 D1·，D2·是同一个软元件时，低 8 位和高 8 位进行交换，32 位指令也一样。

当 M8160=ON，且 D1·，D2·不同，则出错标志 M8067 为 ON，D8067 写入出错代码。该指令不执行。

当 M8160=ON 时所执行的功能与 F147 (SWAP) 指令相同。

6. 应用指令解说

F18 BCD BIN→BCD 变换

F		BCD				BCD 变换						S•	D•				
18	D				P												
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S•							*	*	*	*	*	*	*	*	*	*	
D•								*	*	*	*	*	*	*	*	*	

指令格式：

— [BCD S• D•]

S•：数据来源

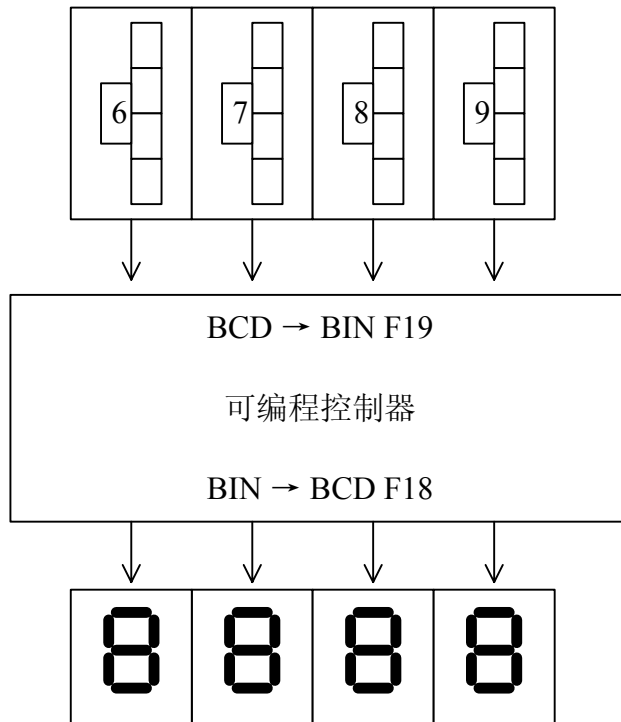
D•：存放位置

功能：数据来源 S• 的内容 BIN 值作 BCD 的转换，存放于 D• 中。

16 位指令时，如转换结果超出 0~9999 范围出错。

32 位指令时，如转换结果超出 0~99999999 范围出错。

出错时，M8067=ON, D8067 写入出错代码。该指令不执行。



四则演算和累增、递减等命令在 PLC 内的运算都是以 BIN 方式来执行。

当 PLC 要读入外部 BCD 的数字开关数据时要用 FNC19 (BCD→BIN) 转换转送指令，而当要向 BCD 的七段显示器输出时，要用 FNC18 (BIN→BCD) 转换转送指令。

但在 FNC72 (DSW)，FNC74 (SEGL) FNC75 (ARWS) 等的特殊命令时，会自动地进行

6. 应用指令解说

BCD/BIN 转换。

6. 应用指令解说

F19 BIN BCD→BIN 变换

F		BIN			BCD→BIN 变换							S•	D•							
19	D			P																
	位元件				字元件															
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	Z	V				
S•							*	*	*	*	*	*	*	*	*	*				
D•								*	*	*	*	*	*	*	*	*				

指令格式：

—— [BIN S• D•]

S•：数据来源

D•：存放位置

功能：将数据来源 S•的内容（BCD：0~9999 或 0~99999999）值作 BIN 的转换，存放于 D•中。

S•如果不是 BCD 码，出错，M8067=ON, D8067 写入出错代码。该指令不执行。

- 当 PLC 要读取 BCD 数字开关的设定值时使用。当资料来源并非为 BCD 值时，将会出错。
- 因为常数 K 自动地转换成二进制，所以不成为这个指令的使用软元件。

F20~F29 四则逻辑运算

四则逻辑运算

功能编号	助记符	名称	页码
20	ADD	加法运算	1
21	SUB	减法运算	2
22	MUL	乘法运算	3
23	DIV	除法运算	4
24	INC	递增运算	5
25	DEC	递减运算	5
26	WAND	逻辑与运算	6
27	WOR	逻辑或运算	6
28	WXOR	逻辑异或运算	6
29	NEG	求补运算	7

6. 应用指令解说

F20 ADD 加法运算

F		ADD		加法运算								S1·	S2·	D·		
20	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·					*	*	*	*	*	*	*	*	*	*	*	*
S2·					*	*	*	*	*	*	*	*	*	*	*	*
D·								*	*	*	*	*	*	*	*	*

指令格式：

—— [ADD S1· S2· D·]

S1· : 被加数

S2· : 加数

D· : 和

标	零	M8020
志	借位	M8021
位	进位	M8022

功能：S1· 和 S2· 按有符号数相加，结果放入 D 中。

如果运算结果为零，M8020 置位。

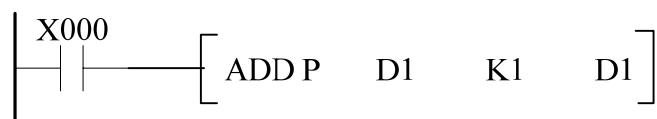
如果运算结果小于最小值，M8021 置位。

如果运算结果超过最大值，M8022 置位。

16 位运算结果在 -32,768 ~ +32,767 之间。

32 位运算结果在 -2,147,483,648 ~ +2,147,483,647 之间。

- 两个数据来源以 BIN 方式相加转送到目的数据区，而各数据的最上位位为正(0)负(1)的符号位，因此可作代数相加。(5+(-8)=-3)
- 演算结果为 0 时，零旗标则动作。演算结果超过 32,767(16 位演算)或 2,147,483,647(32 位演算)时，其进位旗标会动作。(参考次页)演算结果未满 -32,768(16 位演算)或 -2,147,483,648(32 位演算)时，其负旗标会动作。(参考次页)
- 在 32 位演算时要指定 Word 组件下位 16 位的组件，但会用到指定编号的上位 Word，为了不重复使用编号，在指定组件时都指定偶数编号。
- 可以将数据来源和目的数据区指定同一编号，此时若使用连续实行命令(ADD, D ADD)时，则每次扫描时间加算结果都在变化，请注意。

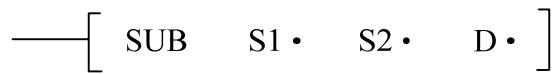


- 如上图所示的顺控程序，每一次 X000 从 OFF—ON 变化时，D1 的内容被加 1，与后述的 INC P 命令相似。

F21 SUB 减法运算

F		SUB		减法运算								S1·	S2·	D·			
21	D		P														
		位元件				字元件											
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·						*	*	*	*	*	*	*	*	*	*	*	*
S2·						*	*	*	*	*	*	*	*	*	*	*	*
D·								*	*	*	*	*	*	*	*	*	*

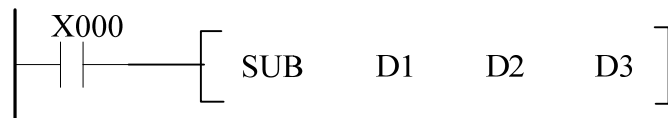
指令格式:



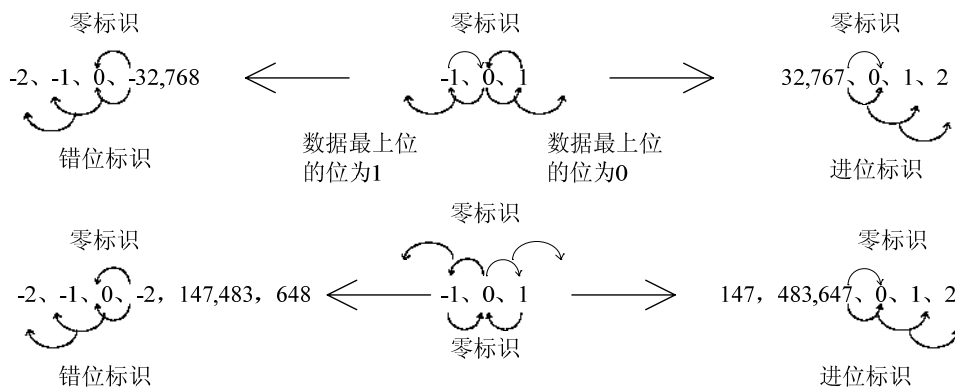
- S1· : 被减数
- S2· : 减数
- D· : 差

将 S1· 及 S2· 中的数据以 BIN 方式进行相减, 结果存在 D· 中
各资料的最高位元符号位元 0 表正数, 1 表负数, 以代数减法运算。

例:



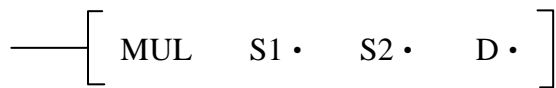
- 当 X000 ON 时, 被减数 D1 内容以代数形式减去减数 D2 内容, 差存放在 D3 中。
- 各标志的动作、32 位运算组件的指定方法、连续执行型和脉冲执行型的差异等均与上页的 ADD 指令相同。
- 标志的动作与数值的正负关系如下所示。



F22 MUL 乘法运算

F		MUL				乘法运算						S1·	S2·	D·			
22	D	P															
		位元件				字元件											
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·						*	*	*	*	*	*	*	*	*	*	*	*
S2·						*	*	*	*	*	*	*	*	*	*	*	*
D·								*	*	*	*	*	*	*	*	*	*

指令格式:



S1· : 被乘数

S2· : 乘数

D· : 积

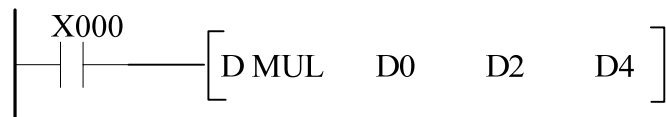
将 S1· 及 S2· 中的数据以 BIN 方式进行相乘, 结果存在 D· 中
各资料的最高位元符号位元 0 表正数, 1 表负数, 以代数乘法运算。

<16 位演算>



- 各数据元所指定的内容演算后的积, 储存在所指定的组件区(下位侧), 同时也占用上一组件共同构成 32 位。如上图示例 (D0)=8, (D2)=9 时, (D5, D4)=72。
- 结果的最高位位为正(0)负(1)的符号。
- D· 为位组件时, 可进行 K1~K8 的位指定。指定为 K4 时, 只能求得乘积的低 16 位。

<32 位演算>

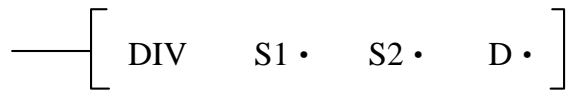


- 如上图示例 : (D1, D0)=8, (D3, D2)=9 时, (D7, D6, D5, D4)=72。
- 32 位演算时其结果储存组件为位组件时, 此时其结果只有下位 32 位存在, 并无存于上位 32 位, 故请使用 Word 组件作运算。
- 使用 Word 组件时, 其演算结果是以 64 位数据储存, 因此无法监看其结果。
- D· 无法指定 Z 组件。

F23 DIV 除法运算

F		DIV		除法运算							S1·	S2·	D·			
23	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·					*	*	*	*	*	*	*	*	*	*	*	*
S2·					*	*	*	*	*	*	*	*	*	*	*	*
D·								*	*	*	*	*	*	*	*	

指令格式:



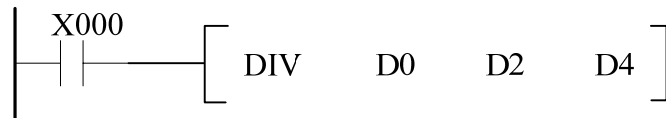
S1· : 被除数

S2· : 除数

D· : 商

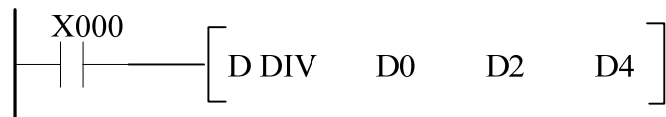
将 S1· 及 S2· 中的数据以 BIN 方式进行相除, 结果存在 D· 中
各资料的最高位元符号位元 0 表正数, 1 表负数, 以代数除法运算。

<16 位演算>



- 如上图示例: D0 为被除数, D2 为除数, D4 存入运算所得的商, D5 存入运算所得的余数。

<32 位演算>



- 如上图示例: (D1, D0) 为被除数, (D3, D2) 为除数, (D5, D4) 存入运算所得的商, (D7, D6) 存入运算所得的余数。
- D· 无法指定 Z 组件。

注

- 除数为 0 时发生运算错误, 命令并不执行, D8067=6706。
- D· 指定为位组件时, 无法得到余数。

6. 应用指令解说

- 商与余数的最上位为正(0)负(1)的符号。商为负值时被除数和除数任一为负值，余数为负值时被除数为负值。

6. 应用指令解说

F24 INC 递增运算

F					递增运算											D·	
24	D	INC	P														
	位元件					字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
D·								*	*	*	*	*	*	*	*	*	

指令格式：

$$\text{---} \left[\text{INC} \quad \text{D} \cdot \right]$$

- 本指令一般都是使用脉冲型指令，否则当指令执行时，每个扫描周期 D· 都会被加 1，所以务必引起注意。
- 16 位运算时，如果从+32,767 加 1，则变为-32,768，标志不动作；
32 位运算时，如果从+2,147,483,647 加 1，则变为-2,147,483,648，标志不动作。

F25 DEC 递减运算

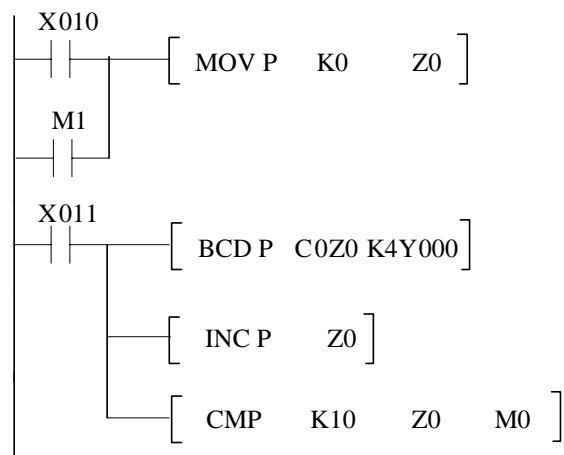
F					递减运算											D·	
25	D	DEC	P														
	位元件					字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
D·								*	*	*	*	*	*	*	*	*	

指令格式：

$$\text{---} \left[\text{DEC} \quad \text{D} \cdot \right]$$

- 本指令一般都是使用脉冲型指令，否则当指令执行时，每个扫描周期 D· 都会被减 1，所以务必引起注意。
- 16 位运算时，如果从-32,768 减 1，则变为+32,767，标志不动作；
32 位运算时，如果从-2,147,483,648 减 1，则变为+2,147,483,647，标志不动作。

<应用例>



- 将计数器 C0~C9 的当前值作 BCD 变换后输出到 K4Y000。
- 预先通过复位输入 X010 清除 Z0。
- X011 每 ON 一次，依次输出 C0,C1..C9 的当前值。

6. 应用指令解说

F26 AND 逻辑与运算

F					逻辑与 (WAND) 运算						S1 ·	S2 ·	D ·			
26	D	WAND	P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·					*	*	*	*	*	*	*	*	*	*	*	*
D1 ·					*	*	*	*	*	*	*	*	*	*	*	*
D2 ·								*	*	*	*	*	*	*	*	*

指令格式:

—— [WAND S1 · S2 · D ·]

F27 OR 逻辑或运算

F					逻辑或 (WOR) 运算						S1 ·	S2 ·	D ·			
27	D	WOR	P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*
D ·								*	*	*	*	*	*	*	*	*

指令格式:

—— [WOR S1 · S2 · D ·]

F28 XOR 逻辑异或运算

F	W				逻辑异或 (WXOR) 运算						S1 ·	S2 ·	D ·			
28	D	WXOR	P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*
D ·								*	*	*	*	*	*	*	*	*

指令格式:

—— [WXOR S1 · S2 · D ·]

S1 · : 资料来源 1

S2 · : 资料来源 2

D · : 运算结果

F29 NEG 求补运算

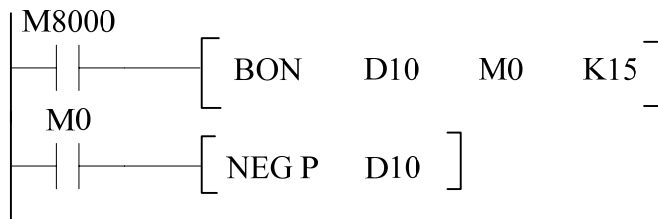
F		NEG		求补运算	D·											
29	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D·								*	*	*	*	*	*	*	*	*

指令格式:



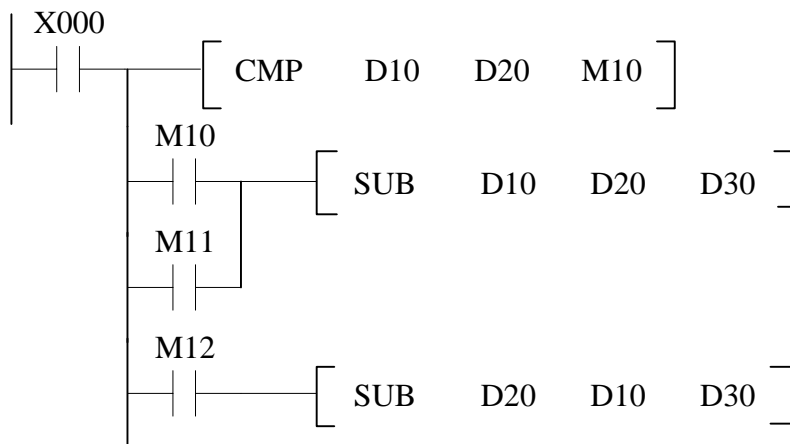
- 将 D· 指定组件的内容各位去反后再加 1，然后将其结果放回 D· 中。
- 本指令一般都是使用脉冲型指令，使用连续执行型指令则在每一个扫描周期都执行该指令，请引起注意。
- 使用这个指令，可得到与负的 BIN 值相对应的绝对值。
注：-32,768 经此运算后，仍为 -32,768。

《应用回路 1》负数的绝对值化



D10 的第 15 位为 1 时，M0 置 ON，
当 M0 为 ON 时，对 D10 使用补码指令。

《应用回路 2》减法的绝对值处理



以上回路中即便不使用补码运算，D30 也通常表示减法运算中差的绝对值。

F30~F39 旋转，移位

旋转，移位

功能编号	助记符	名称	页码
30	ROR	循环右移	1
31	ROL	循环左移	1
32	RCR	带进位循环右移	2
33	RCL	带进位循环左移	2
34	BSFR	位右移	3
35	BSFL	位左移	3
36	WSFR	字右移	5
37	WSFL	字左移	5
38	SFWR	移位写入	7
39	SFRD	移位读出	8

6. 应用指令解说

F32 RCR 带进位循环右移

F		RCR	P	带进位循环右移										D·	n		
32	D																
	位元件					字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
D·								*	*	*	*	*	*	*	*	*	
n					*	*											

指令格式:

$$\text{——} \left[\text{RCR} \quad \text{D} \cdot \quad \text{n} \right]$$

F33 RCL 带进位循环左移

F		RCL	P	带进位循环左移										D·	n		
33	D																
	位元件					字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
D·								*	*	*	*	*	*	*	*	*	
n					*	*											

指令格式:

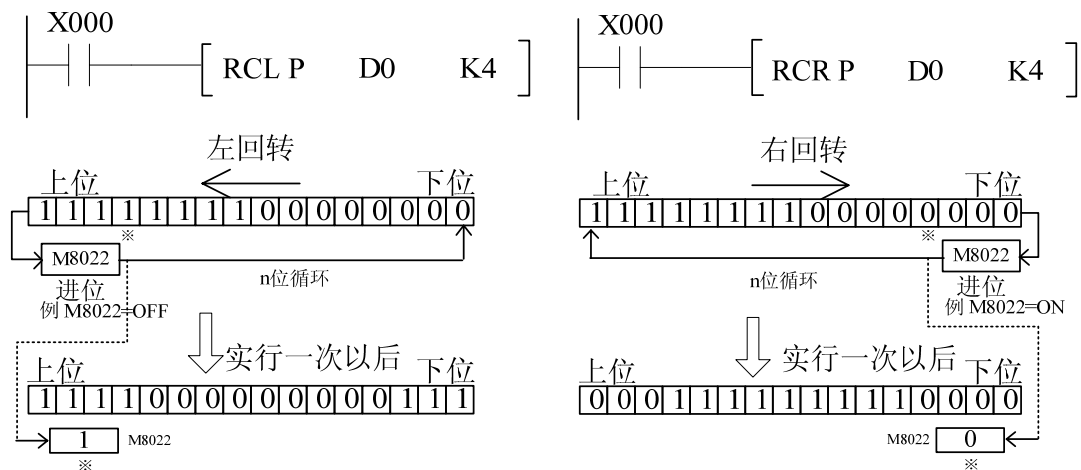
$$\text{——} \left[\text{RCL} \quad \text{D} \cdot \quad \text{n} \right]$$

16 位指令 $n \leq 16$; 32 位指令 $n \leq 32$ 。

使 16 位或 32 位数据的各位信息带进位标志 M8022 的左右回转指令。

《带进位左回转》

《带进位右回转》



指令解释: X000 由 OFF 向 ON 变化时, 进行带进位标志 M8022 的 K4 位左(右)回转。

- 连续执行型指令每一个扫描周期都进行回转动作, 请引起注意。
- 32 位命令也一样。
- 在位数指定组件时, 只有 K4 (16 位命令)、K8 (32 位命令) 有效。(例 K4Y010, K8M0)

6. 应用指令解说

F34 SFTR 位右移

F		SFTR				位右移						S · D · n1 n2					
34		P															
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S ·	*	*	*	*													
D ·		*	*	*													
n1					*	*											
n2					*	*											

指令格式:

—— [SFTR S · D · n1 n2]

F35 SFTL 位左移

F		SFTL				位左移						S · D · n1 n2					
35		P															
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S ·	*	*	*	*													
D ·		*	*	*													
n1					*	*											
n1					*	*											

指令格式:

—— [SFTL S · D · n1 n2]

S · : 位移装置的起始编号

D · : 欲位移装置的起始编号

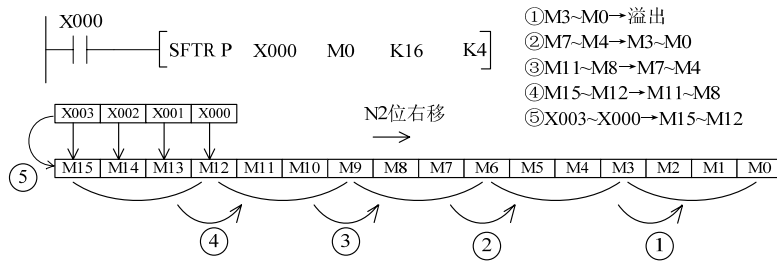
n1: 欲位移的资料长度, n1=1~1024

n2: 一次移位的位元数, n2=1~n1

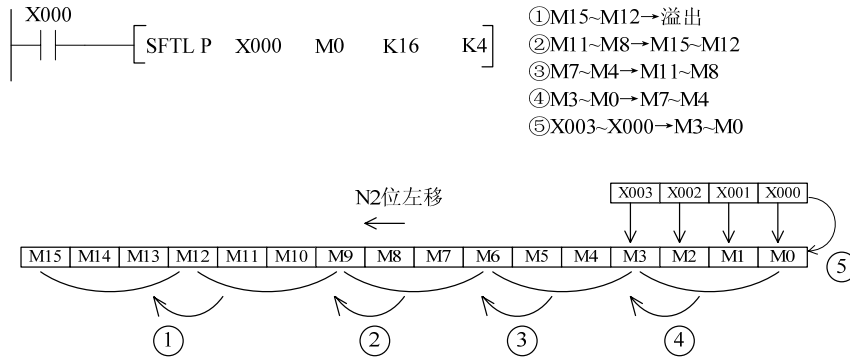
- 将 D · 开始的起始编号, 具有 n1 个位元 (位移暂存器的长度) 的位元装置, 以 n2 位元个数来右移, 而 S · 开始起始的编号以 n2 位元移入 D · 中来填补位元空位。
- 本指令一般都是使用脉冲型指令, 使用连续执行型时, 每个扫描周期都执行, 请引起注意。

6. 应用指令解说

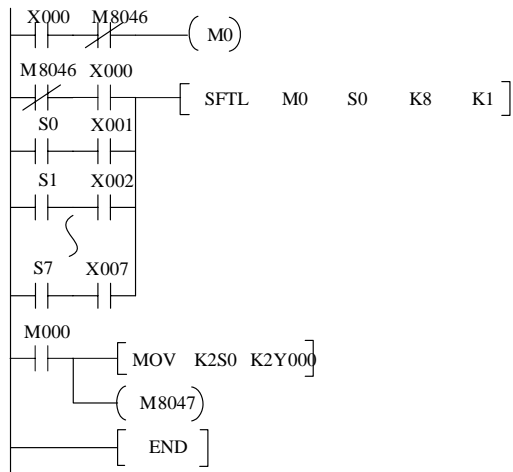
位右移



位左移



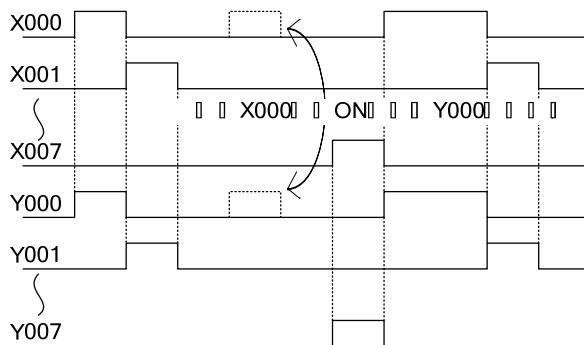
1 位数据的有条件步进



数据输入起始数据

以 M0 的 1 位作为超始输入

构成 S0~S7 的 8 位字长的移位寄存器



6. 应用指令解说

F36 WSFR 字右移

F		WSFR		字右移	S · D · n1 n2											
36			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·							*	*	*	*	*	*	*	*		
D ·								*	*	*	*	*	*	*		
n1					*	*										
n2					*	*										

指令格式:

—— [WSFR S · D · n1 n2]

F37 WSFL 字左移

F		WSFL		字左移	S · D · n1 n2											
37			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·							*	*	*	*	*	*	*	*		
D ·								*	*	*	*	*	*	*		
n1					*	*										
n2					*	*										

指令格式:

—— [WSFL S · D · n1 n2]

S · : 移位装置的起始编号

D · : 欲移位装置的起始编号

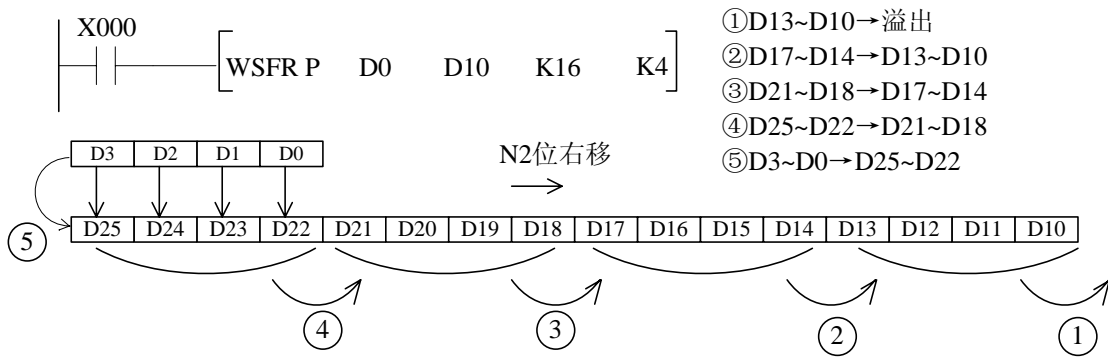
n1: 以字为单位的欲位移的资料长度, n1=1~512

n2: 一次移位的字元数, n2=1~n1

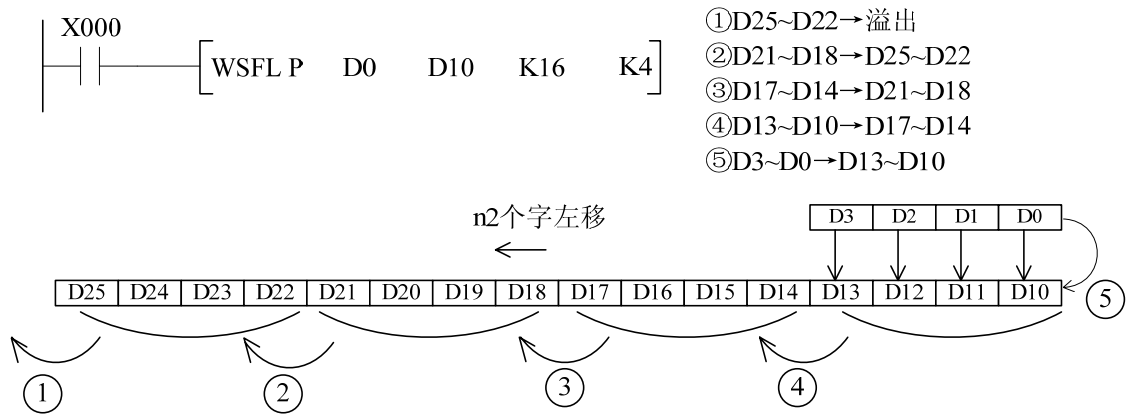
- 将 D · 开始的起始编号的 n1 个字元 (位移暂存器的长度) 长度的资料串列, 以 n2 个移位数来右 (左) 移, 而 S · 开始起始的编号以 n2 个字元数移入 D · 中来填补字元空位。
- 本指令一般都是使用脉冲型指令
- 当 S · /D · 都指定为位组合装置时, 请做相同的位指定

6. 应用指令解说

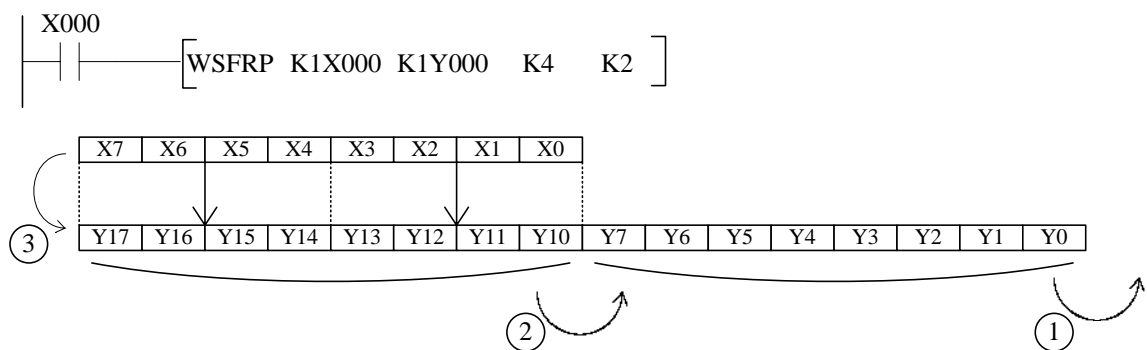
《字右移》



《字左移》



位指定软元件的移位



请将位元件做相同位指定

6. 应用指令解说

F38 SFWR 移位写入

F		SFWR				移位写入						S·	D·	n			
38				P													
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·					*	*	*	*	*	*	*	*	*	*	*	*	
D·								*	*	*	*	*	*	*			
n					*	*											

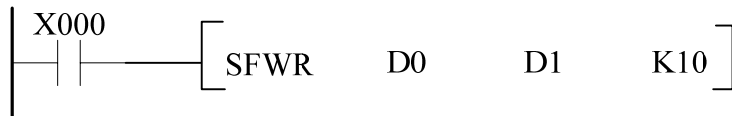
指令格式：

$$\text{——} \left[\text{SFWR} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

S·：移位写入的源址装置

D·：写入目标资料串列的起始装置

n：欲写入的资料长度，n=2~512（实际写入的资料长度为 n-1，起始装置 D· 做为存储写入点数指针用）。



- 为控制先入先出的数据写入指令。
- 应预先将 D1 复位成 0。
- 当驱动 X000 由 OFF 置 ON 时，D0 的内容被存入 D2，D1 的内容变为 1。当 X000 再次由 OFF 置 ON 时，D0 的内容被存入 D3，D1 的内容变为 2。（使用连续执行型指令时，每一个扫描周期都执行一次保存。）
- 指针 D1 的内容为写入数据的当前点数，D1 的内容超过 n-1 时，则指令停止执行，进位标志 M8022 工作。

6. 应用指令解说

F39 SFRD 移位读出

F		SFRD			移位读出							S·	D·	n			
39			P														
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·								*	*	*	*	*	*	*			
D·								*	*	*	*	*	*	*	*	*	
n					*	*											

指令格式：

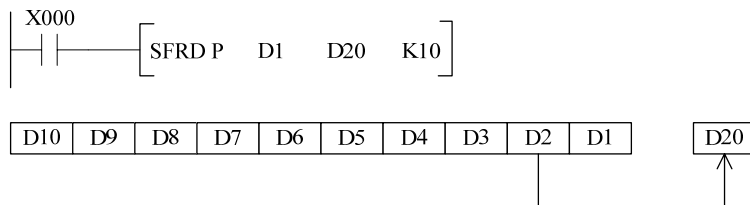
— [SFRD S· D· n]

S·：移位读出资料串列的起始装置

D·：资料串列读出存放的目标装置

n：欲移位读出的资料长度，n=2~512（实际读出的资料长度为 n-1，读出起始装置 S· 做为判断指令停止执行的指针用）。

- 为控制先入先出的数据读出指令。

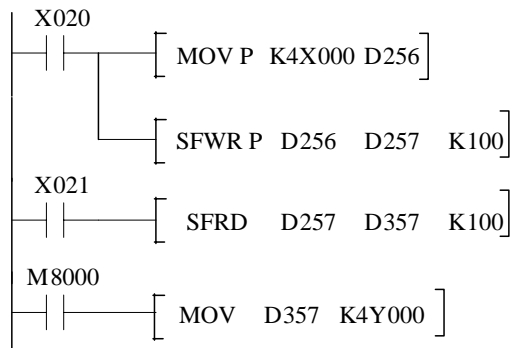


- 当驱动 X000 由 OFF 置 ON 时，D2 的内容被读出到 D20。与此同时，指针 D1 的内容减 1，左侧的数据向右侧移动一个字元（D10 的内容不变）。当 X000 再次由 OFF 置 ON 时，再次将 D2 的内容读出到 D20，D1 的内容再减 1。（使用连续执行型指令时，每一个扫描周期都执行一次读出和移位。）
- 指针 D1 的内容为 0 时，指令停止执行，同时零点标志 M8020 动作。

<先进先出的控制例>位移写入，位移读出例

产品的编号一边登录一边顺序入仓，为了物品的出仓能先进先出，其范例如下所示。

产品编号是以 16 进制 4 位数，最大仓量为 99 点以下。



X000~017 为产品编号的输入，转送到 D256。

D257 为索引、D258~D356 为 99 点储存产品编号用的数据缓存器。

对应出仓的先进编号，送出到 D357。

6. 应用指令解说

取出的产品编号以 16 进 4 位数到 Y000~Y017 来表示。

F40~F49 数据处理

数据处理

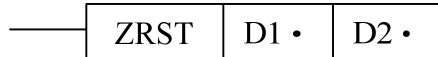
功能编号	助记符	名称	页码
40	ZRST	全部复位	1
41	DECO	解码器	2
42	ENCO	编码器	3
43	SUM	ON 位元数量	4
44	BON	ON 位元判定	5
45	MEAN	平均值	6
46	ANS	警示线圈设定	7
47	ANR	警示线圈复位	8
48	SOR	BIN 数据 BIN 开方运算	9
49	FLT	BIN 整数→2 进制	10

6. 应用指令解说

F40 ZRST 全部复位

F		ZRST		全部复位	D1 ·	D2 ·										
40			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D1 ·		*	*	*							*	*	*	*		
D2 ·		*	*	*							*	*	*	*		

指令格式：



D1 · : 全部清除的起始装置

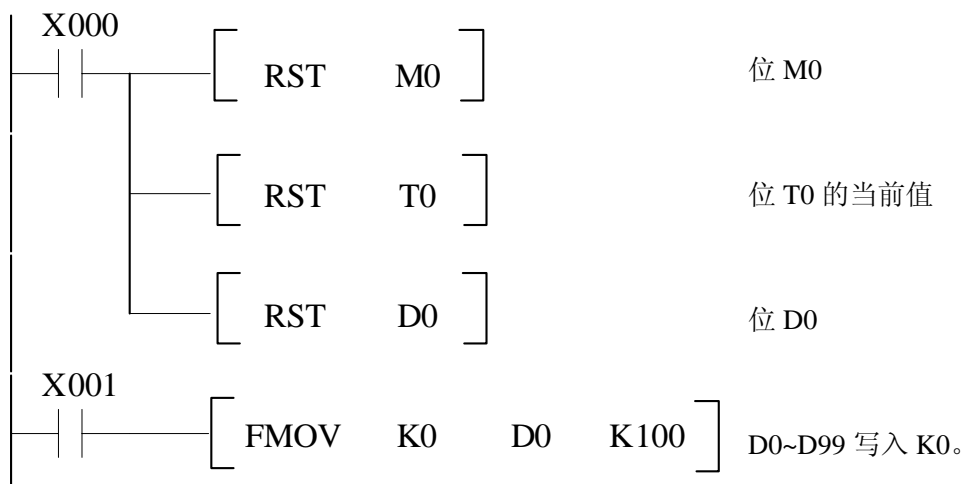
D2 · : 全部清除的结束装置, D1 · 编号 ≤ D2 · 编号, 且必须指定相同类型的装置。否则本指令出错, D8067=6705。



当 X000 从 OFF 到 ON 时, 从 M100 到 M200 全部清除

- 当 D1 · 的运算元编号 > D2 · 运算元编号时, 只有 D · 1 指定的运算元被清除。
- 这个指令以 16 位执行, 但是 D1 · /D2 · 可指定 32 位计数器。但是不能混合指定, 如 D1 · 为 16 位计数器, 而 D2 · 为 32 位计数器这种情况。
- 本指令一般都是使用脉冲型指令。

作为软元件的单独复位指令, 对于位元件的 Y, M, S 和字元件的 T, C, D 可使用 RST 指令。作为常数 K0 的成批写入指令的 F16 FMOV 指令, 可以把写入 KnY, KnM, KnS, T, C, D 的软元件中。



6. 应用指令解说

F41 DECO 解码器

F		DECO			解码器								S ·	D ·	n	
41			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·	*	*	*	*	*	*					*	*	*	*	*	*
D ·		*	*	*							*	*	*	*		
n					*	*										

指令格式：

$$\text{——} \left[\text{DECO} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

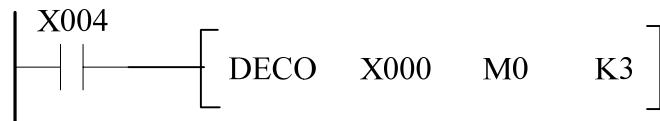
S · : 解码来源装置

D · : 存放解码结果装置, D · 为位元装置时 n=1~8, D 为字元装置时 n=1~4.

n: 解码位元长度, n=1~8, n=0 时不处理, 0~8 以外会出错。

来源装置 S · 的低位 n 位元作解码, 并将其 2^n 位元长度的结果存于 D · 中。

本指令一般都是使用脉冲型指令。



X002	X001	X000
0	1	1

M7	M6	M5	M4	M3	M2	M1	M0
0	0	0	0	1	0	0	0

资料来源为 $1+2=3$, 则 M3 置 1, 其他位清零。

D · 是位元元件时, n=8 时, $2^8=256$ 点。

D · 是字元元件时, n=4 时, $2^4=16$ 点, n<4 时, D · 的高位做零扩展。

6. 应用指令解说

F42 ENCO 编码器

F		ENCO			编码器							S·	D·	n		
42			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·	*	*	*	*							*	*	*	*	*	*
D·											*	*	*	*	*	*
n					*	*										

指令格式:

$$\text{——} \left[\text{ENCO} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

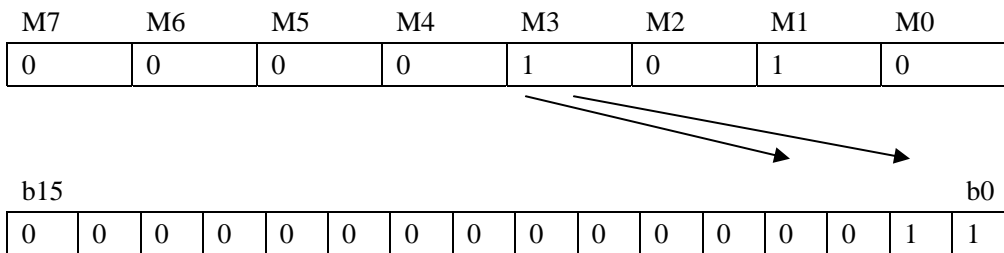
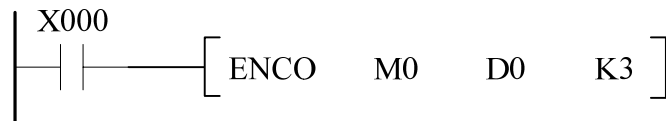
S· : 编码来源装置

D· : 存放编码结果装置

n: 编码位元长度, S· 为位元装置时, n=1~8, S· 为字元装置时, n=1~4。

来源装置 S· 的 $2^{\text{exp } n}$ 位元长度的资料作编码, 并将其结果存于 D· 中。

本指令一般都是使用脉冲型指令。



资料来源内有多位元为 1 时, 低位的 1 不予处理。资料来源全部为 0 时候出错。

S· 是位元元件时, n=8 时, $2^8=256$ 点。

S· 是字元元件时, n=4 时, $2^4=16$ 点。

6. 应用指令解说

F43 SUM ON 位元数量

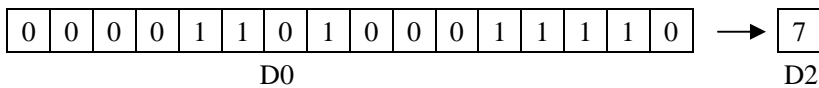
F		SUM				ON 位元数量						S·	D·				
43	D				P												
	位元件					字元件											
	X	Y	M	S		K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·						*	*	*	*	*	*	*	*	*	*	*	*
D·									*	*	*	*	*	*	*	*	*

指令格式:

$$\text{---} \left[\text{SUM} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S· : 来源装置

D· : 存放计数值的目的装置



- 如果 D0 中 16 个位元全部为 0 时，零标识 M8020 会动作。
- 使用 32 位元指令时，D· 仍占用 2 个暂存器，如上图所示，(D1, D0) 的 32 位中的 1 的个数写入 D2，同时 D3 全部为 0。

6. 应用指令解说

F44 BON ON 位元判定

F		BON		ON 位元判定								S·	D·	n		
44	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·					*	*	*	*	*	*	*	*	*	*	*	*
D·		*	*	*												
n					*	*										

指令格式：

$$\text{---} \left[\text{BON} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

S·：来源装置

D·：存放判断结果装置

n：指定判定位元，n=0~15（16位元指令），n=0~31（32位元指令）。

S·的第n个位元为1时D·置1，为0则D·置0。

例：

$$\begin{array}{|c} \text{X000} \\ \hline \text{---} \end{array} \left[\text{BON} \quad \text{D10} \quad \text{M0} \quad \text{K15} \right]$$

b15	0	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	b0	
	D10																	→	M0=OFF

1	0	0	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	b0
D10																	→	M0=ON

进行16位运算时，n=0~15 进行32位运算时 n=0~31.

6. 应用指令解说

F45 MEAN 平均值

F		MEAN		平均值								S·	D·	n			
45	D		P														
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·							*	*	*	*	*	*	*	*			
D·								*	*	*	*	*	*	*	*	*	
n					*	*											

指令格式:

$$\text{---} \left[\text{MEAN } S \cdot \quad D \cdot \quad n \right]$$

S· : 来源装置

D· : 存放平均值的目的装置

n: 指定求平均值的个数, n=1~64

$$\begin{array}{|l} \text{X000} \\ \hline \text{---} \left[\text{MEAN } D100 \quad D150 \quad K5 \right] \end{array}$$

$(D100+D101+D102+D103+D104) / 5 \longrightarrow D150$

- 将 n 点的源数据的平均值 (代数和被 n 除) 存入目标地址中。余数舍去。超过软元件编号时, 在可能的范围内取 n 的最小值。

- n 在 1~64 以外时, 会发生错误。

当 S· 指定的范围越出 n 时, 本指令在有效范围内作平均值计算。

如:

$$\begin{array}{|l} \text{X000} \\ \hline \text{---} \left[\text{MEAN } D8510 \quad D0 \quad K3 \right] \end{array}$$

$(D8510+D8511) / 2 \longrightarrow D0$

6. 应用指令解说

F46 ANS 警示线圈设定

F		ANS			警示线圈设定							S·	m	D·							
46				P																	
	位元件				字元件																
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z					
S·											*										
m					*	*															
D·				*																	

指令格式:

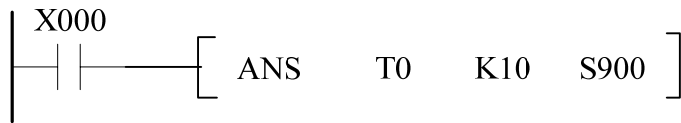
$$\text{——} \left[\text{ANS} \quad \text{S} \cdot \quad \text{m} \quad \text{D} \cdot \right]$$

S· : 侦测警报计时器,仅可使用 T0~T199

m : 计时时间, 可指定 m=1~32,767 (单位 100ms)。

D· : 警报点装置, D 仅可使用 S900~S999。

※ 1: 对于 BSP01 SR 机种 S· 仅可使用 T0~39, T196~T199
用于驱动信号报警期的方便指令。



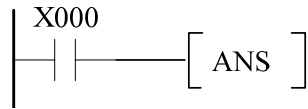
- 如果 X000 接通 1 秒以上, 则 S900 被置位, 此后即使 X000 变 OFF (定时器被复位), S900 仍保持动作状态。
若不满 1 秒, X000 或 X001 变为 OFF 时, 定时器复位。
- 如果预先将 M8049 (信号报警有效) 置 ON, 则信号报警器 S900~S999 中处于 ON 状态的最小编号被存入 D8049 中。另外, 当 S900~S999 中任意一个为 ON 时, M8048 (报警器动作) 置 ON。

6. 应用指令解说

F47 ANR 警示线圈复位

F		ANR		警示线圈复位	无对应装置
47			P		

指令格式:



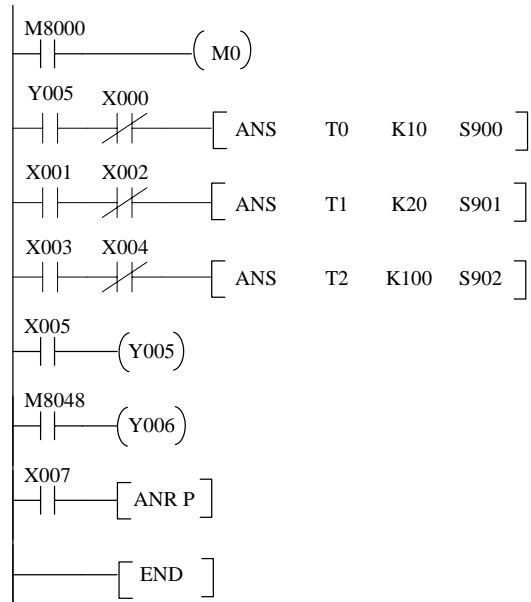
如果 X000 接通, 则信号报警器 S900~S999 中正在动作的报警点被复位。

如果同时有多个报警点动作时, 则复位编号最小的一个报警点。此时, 如果信号报警器有效标志 M8049 处于 ON 状态, 则寄存器 D8049 的内容会及时更新, 始终保持剩下报警点中最小的编号。

若将 X000 再次接通, 则下一个编号的状态被复位。

若采用 ANRP 指令, 则在每个扫描周期中按顺序复位, 请引起注意。

如下图编写外部故障诊断电路, 由特殊数据缓存器 D8049 来监视 S900~S999 之间动作状态编号最小的号码。多个故障同时发生时, 最小编号的故障解除后, 接着显示下一个故障编号。



- 当特殊辅助继电器 M8049 动作时, 才能监视。
- 前进输出 Y005 驱动时前进检出, X000 在一秒钟内若无动作时, S900 动作。
- DOG 异常时, 上限 X001 和 X002 在 2 秒以上同时不动作时, S901 动作。
- 在 T2 时间未满足 10 秒时, 连续运转输入点 X003 ON 时, 机械在 1 循环运转中动作开关 X004 没有动作时, S902 动作。
- S900~S999 间有 ON 时, 特殊辅助继电器 M8048 动作, 故障表示输出 Y006 动作。
- 执行外部故障诊断程序的动作前, 先将状态清除按键 X007 为 OFF; X007 为 ON 时, 较小号码会顺序被清除。

<警示线圈有效 M8049>

当 M8049 驱动时, S900~S999 的动作状态最小号码, 则储存在 D8049 内。

<警示线圈的动作 M8048>

M8049 被驱动时, 状态 S900~S999 间有动作时, M8048 会动作。

6. 应用指令解说

F48 SQR BIN 开方运算

F		SQR			BIN 开方运算							S·	D·									
48	D			P																		
	位元件				字元件																	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z						
S·					*	*							*	*								
D·													*	*								

指令格式：

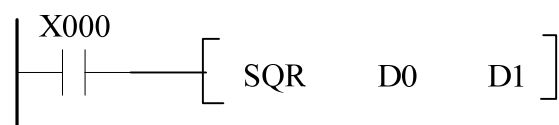
$$\text{——} \left[\text{SQR} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：开平方根的资料来源

D·：存放结果的目的装置

将 S· 中的数据开平方，结果存放在 D· 指定的装置。

例：



当 X000 ON 时，D0 中的数据开方后，结果存放在 D1 中。

- S· 只能是非负数据才有效，若为负数时，则运算出错，标志 M8067 置 ON，指令不被执行。
- 运算结果 D· 只求整数，小数部分被舍去，借位标志信号 M8021 置 ON。
- 运算结果为 0 时，零标志 M8020 置 ON。

6. 应用指令解说

F49 FLT BIN 整数→2 进制浮点数转换

F		FLT				BIN 整数→2 进制浮点数转换						S·	D·			
49	D			P												
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·													*	*		
D·													*	*		

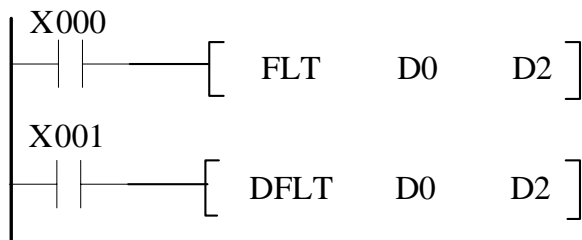
指令格式：

$$\text{——} \left[\text{FLT} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：BIN 整数→2 进制浮点数转换的资料来源

D·：存放结果的目的装置

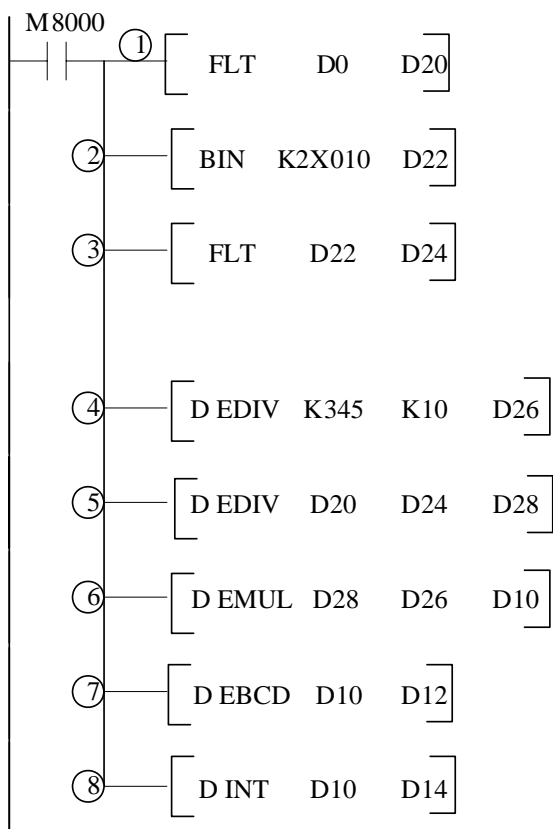
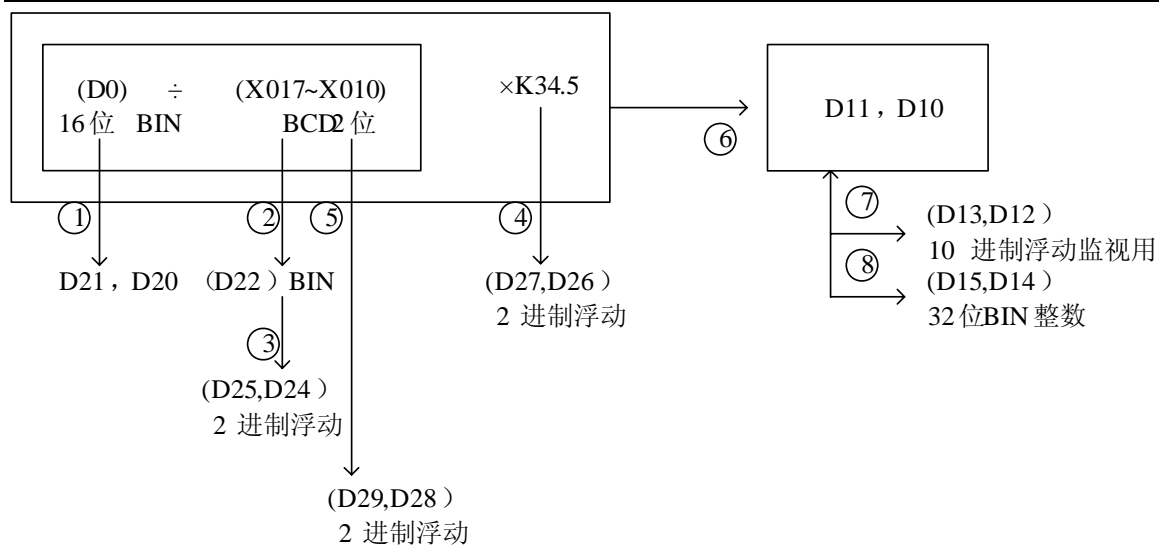
例：



- BIN 整数值与 2 进制浮点值间的转换指令。
常数 K、H 在各浮点运算指令中被自动转换，因此在本 FLT 指令中不能使用。
- 这个指令的逆变换指令为 FNC129 (INT)。

浮点运算示例表示如下例题的顺控示例

6. 应用指令解说



F50~F59 高速计数处理

高速计数处理

功能编号	助记符	名称	页码
50	REF	输入输出刷新	1
52	MTR	矩阵输入	2
53	HSCS	比较置位	4
54	HSCR	比较复位	6
55	HSZ	区间比较	7
56	SPD	脉冲密度	8
57	PLSY	脉冲输出	9
58	PWM	脉宽调制	11
59	PLSR	带加减速脉冲输出	12

6. 应用指令解说

F50 REF 输入输出刷新

F		REF				输入输出刷新						D · n				
50		P														
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D ·	*	*														
n					*	*										

指令格式：

—— [REF D · n]

D · : 输入输出刷新起始软元件号

n: 刷新点数

对于 X 只能刷新 X0~X17, 一共 16 点的输入

对于 Y 只能刷新 Y0~Y7, 一共 8 点的输出

※ 1: 对于 BSP01-14SR 机种

X 只能刷新 X0~X7, 一共 8 点的输入

Y 只能刷新 Y0~Y5, 一共 6 点的输出, 当刷新点数 n 等于 8 的时候实际只刷新 Y0~Y5

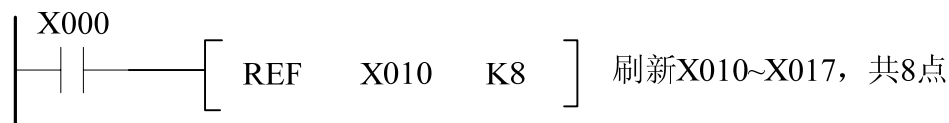
※ 2: 对于 BSP01-20SR 点机种

X 只能刷新 X0~X13, 一共 12 点的输入, 当刷新点数 n 等于 16, 且从 X0 开始刷新, 实际只刷新 X0~X13, 同样当刷新点数 n 等于 8, 且从 X10 开始刷新, 实际只刷新 X10~X13

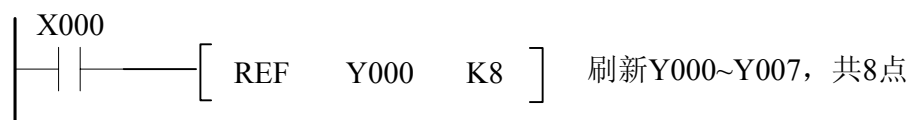
Y 只能刷新 Y0~Y7, 一共 8 点的输出

可编程控制器采用集中输入输出刷新方式, 输入端信息在 0 步运算前存入输入映像存储区。输出端在执行 END 指令后, 由输出映像存储区通过锁存存储器输出。但是, 在运算过程中, 需要最新的输入信息以及希望立即输出运算结果时, 可以使用该输入输出刷新指令。

例 1: 输入刷新:



例 2: 输出刷新:

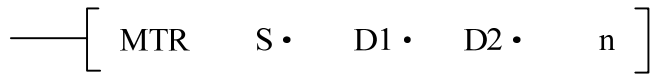


- 指定起始软元件编号 D · 时, 请将最低位编号置为 0, 如 X000, X010, Y000 那样。
- 刷新点数 n 应为 8 或 16, 否则出错。
- 一般 REF 指令可运用在 FOR~NEXT 指令之间, CJ 指令之间。
- 在有输入输出动作的中断处理中, 执行该指令以获得最新的输入信息和及时输出运算结果。

F52 MTR 矩阵输入

F		MTR		滤波调整										S · D1 · D2 · n		
52																
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·	*															
D1 ·		*														
D2 ·		*	*	*												
n					*	*										

指令格式:



S · : 矩阵扫描输入起始软元件, 其最右编号应为 0, 如 X000, X010 等, 且占用连续 8 点

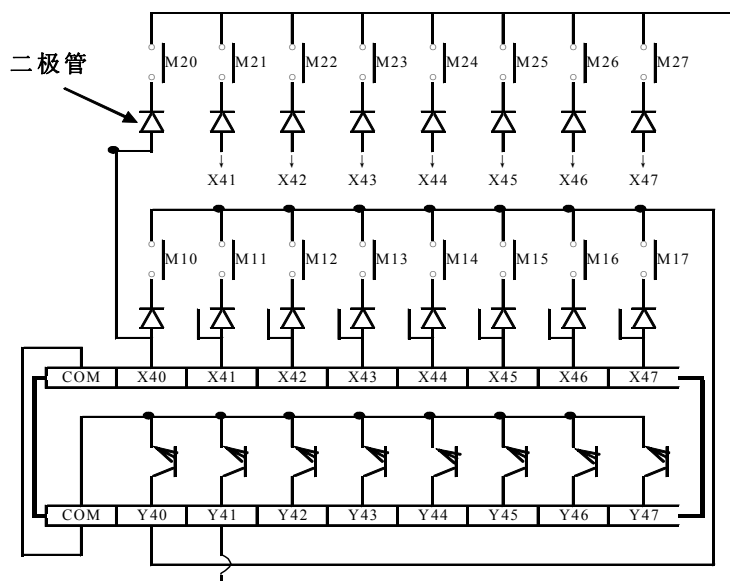
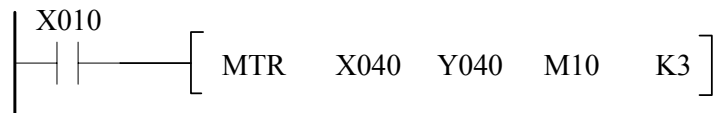
D1 · : 矩阵扫描输出起始软元件, 同样其最右编号应为 0

D2 · : 矩阵扫描值存储起始软元件, 同样其最右编号应为 0

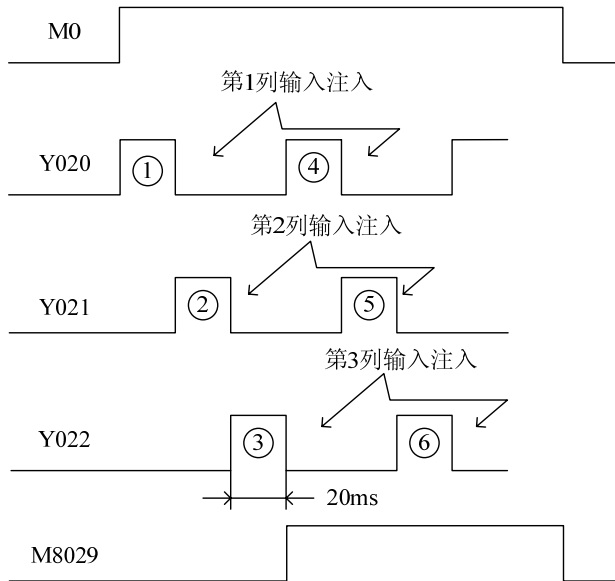
n: 矩阵扫描的列数, n=2~8 有效

本指令为使用 8 点输入与 n 点输出, 按顺序读入 8 点 n 列的输入信号的指令。

例:



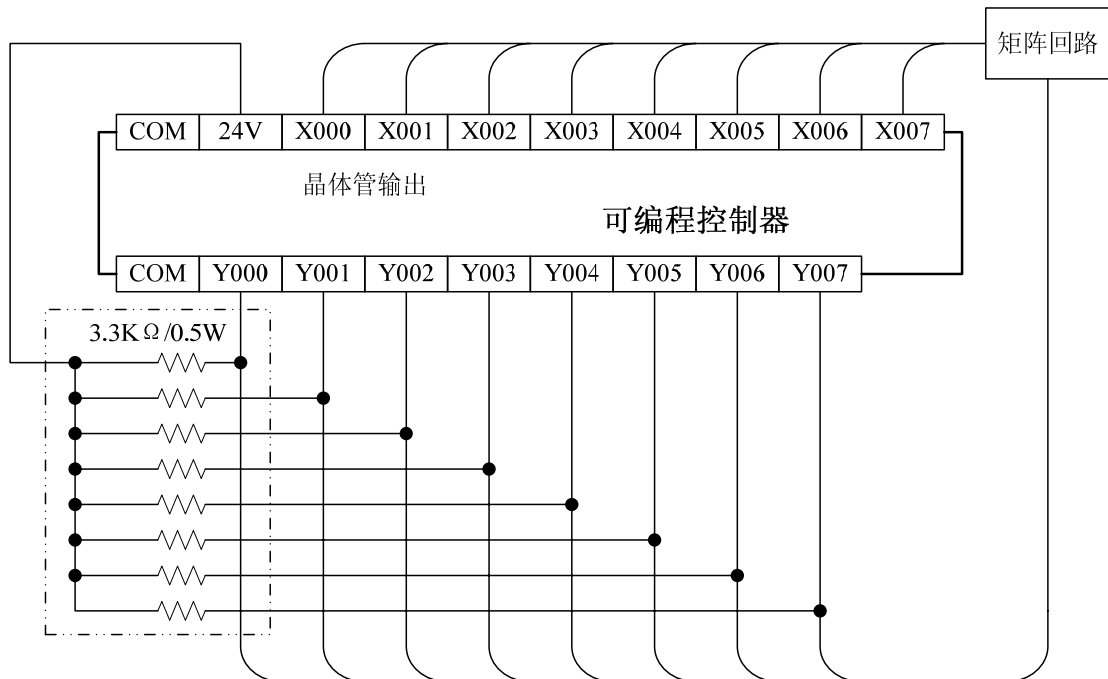
6. 应用指令解说



- 以指定的输入 S · 为起始，占有 8 点输入。
- 以指定的输出 D1 · 为起始，占有 3 点输出。
- 图例为：n=3 点的输出 Y040、Y041、Y042 依次反复 ON。每次依次反复获得第一列、第二列、第三列的输入，存入 M10~M17、M20~M27、M30~M37 中。
- 各输出按每 20ms 顺序中断，进行即时输入输出处理。

《MTR 命令使用的输入编号》

1. MTR 命令的输入编号通常请在使用在 X020 以后(16 点基本形为 X010 以后)
2. 使用 MTR 命令时，晶体管输出需接(3.3K/0.5W)的电阻。



F53 HSCS 比较置位

F					比较置位											S1·	S2·	D·		
53	D	HSCS																		
		位元件				字元件														
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S1·						*	*	*	*	*	*	*	*	*	*		*			
S2·													*							
D·		*	*	*																

指令格式：

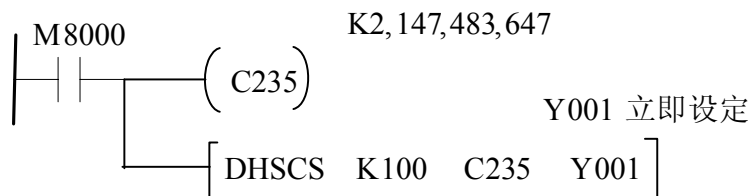
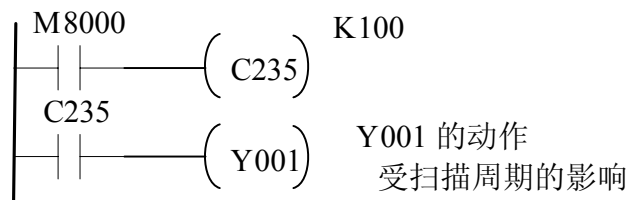
$$\text{---} \left[\text{DHSCS} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \right]$$

S1· : 比较值

S2· : 高速计数器编号，必须指定高速计数器 C235~C249，C251~C254

D· : 比较结果，D·也可指定中断指示器 I010~I060

例：



- 高速计数器是根据计数输入的 OFF→ON 以中断方式计数，计数器的当前值等于设定值时，计数器的输出接点立即工作。但是若不使用 FNC53 指令时，向外部输出与顺控有关，故而受扫描周期的影响，在 END 处理后驱动寻输出。
- 使用 FNC53 指令，能中断处理比较和外部输出，所以 C235 的当前值变为 99→100 或 101→100 时，Y001 立即置位。

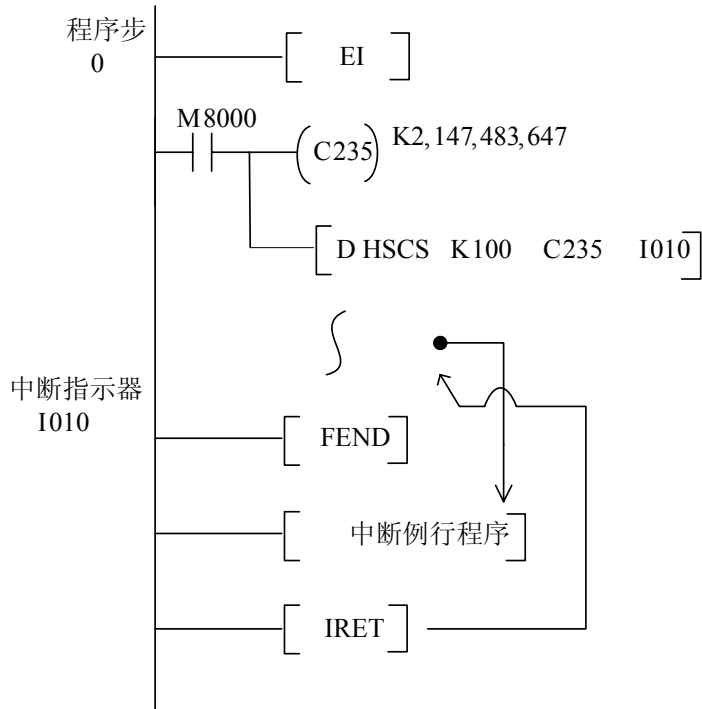
《注意事项》

- 该指令在是 32 位专用指令，必须作为 DHSCS 指令输入。
- 这些指令在脉冲输入时才比较结果和动作，因此即使通过传送指令改变当前值，只要没有计数输入，比较输出就不发生变化。
- F53, F54, F55 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在总计 6 个指令以下。
- 若使用 DHSCS、DHSCR、DHSZ 等命令，可编程控制器高速计数器的最大允许频率将

6. 应用指令解说

会受到很大影响。

《计数中断》



- DHSCS 指令的 D · 可以指定为中断指示器 I010~I060。(不可重复使用编号)
- 在 S2 · 指定的高速计数器的当前值变化为 S1 · 指定的值时，执行 D · 指定标号的中断程序。
- 特殊辅助继电器 M8059=ON 时，I010~I060 的中断被全部禁止。

关于中断处理的详细说明，请参照 FNC03 (IRET) ~FNC05 (DI)。

F54 HSCR 比较复位

F					比较置位							S1·	S2·	D·			
54	D	HSCR															
		位元件				字元件											
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·						*	*	*	*	*	*	*	*	*	*		*
S2·													*				
D·			*	*	*								*				

指令格式：

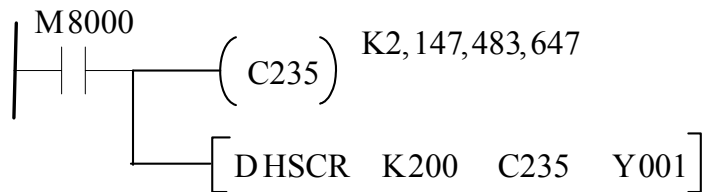
—— [DHSCR S1· S2· D·]

S1· : 比较值

S2· : 高速计数器编号, 必须指定高速计数器 C235~C249, C251~C254

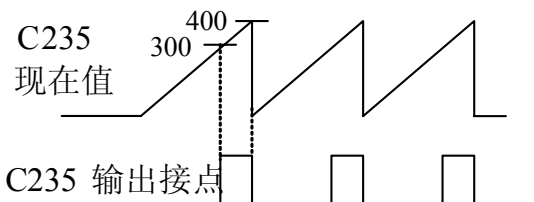
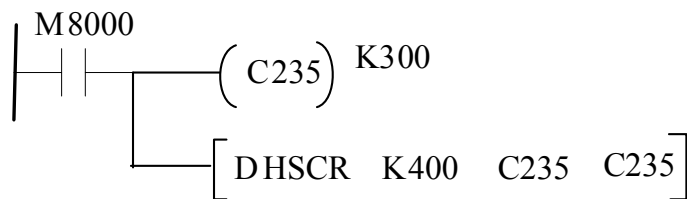
D· : 比较结果, D· 也可指定为 S2· 一样的高速计数器编号

例：



- 若用 F54 指令, 由于比较和外部输出采用中断处理, C235 的当前值 199→200 或 201→200 时, 不受扫描周期的影响, Y001 立即复位, 关于扫描周期的影响请参照上述 FNC53。

《自行复位回路示例》



- C235 的当前值变为 400 时, C235 立即复位, 当前值为 0, 输出触点不工作。
- 该指令为 32 位专用指令, 必须作为 DHSCR 指令的输入。其它的注意事项请参照 FNC53 的《注意事项》。

6. 应用指令解说

F55 HSZ 区间比较

F		HSZ			区间比较							S1 · S2 · S · D ·				
55	D															
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·					*	*	*	*	*	*	*	*	*	*		*
S2 ·					*	*	*	*	*	*	*	*	*	*		*
S ·												*				
D ·		*	*	*												

指令格式:

—— [DHSZ S1 · S2 · S · D ·]

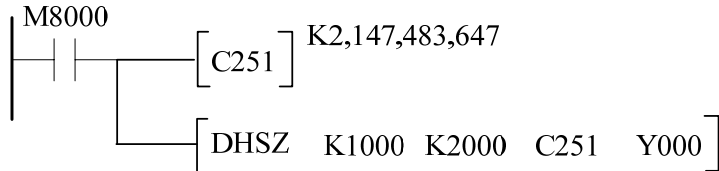
S1 · : 区域比较下限值

S2 · : 区域比较上限值, $S1 \cdot \leq S2 \cdot$

S · : 高速计数器编号, 必须指定高速计数器 C235~C249, C251~C254

D · : 比较结果, 占用连续 3 个装置。

例:



比较输出的动作:

K1000 > C251 当前值 Y000 ON

K1000 \leq C251 当前值 \leq K2000 Y001 ON

K1000 < C251 当前值 Y002 ON

- S1 · 和 S2 · 的内容为 $S1 \cdot \leq S2 \cdot$
- 若利用 FNC55, 比较与外部输出一起中断处理, 不受扫描周期的影响。
- 指令使能后, 在第一个扫描周期内的输出是根据 S · 的当前值与 S1 · 和 S2 · 比较的结果作为输出。
- 该指令为 32 位专用指令, 必须作为 DHSZ 指令的输入。其它的注意事项请参照 FNC53 的《注意事项》。

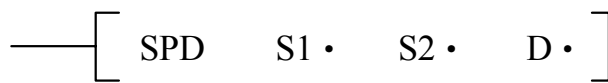
6. 应用指令解说

F56 SPD 脉冲密度

F					脉冲密度											S1 ·	S2 ·	D ·	
56		SPD																	
	位元件				字元件														
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S1 ·	*																		
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*	*	*	
D ·											*	*	*	*	*	*	*		

指令数：每个输入 1 个指令（但可用索引寄存器修饰）

指令格式：

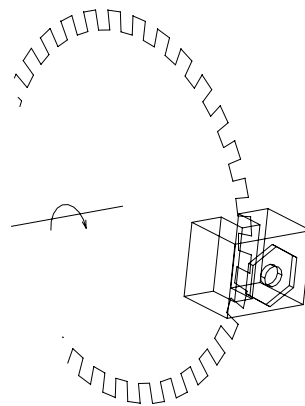
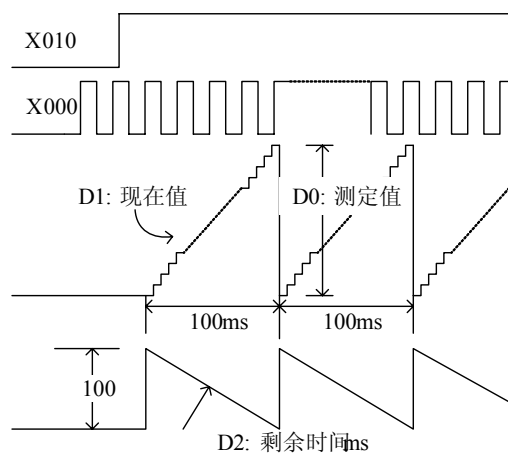
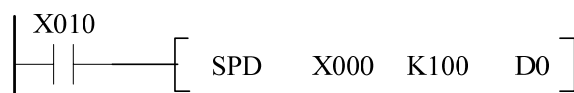


S1 · ：外部脉冲输入端，BSP01-SR 机种只可指定 X0~X3。其他机种 S1 · 可指定 X0~X5，

S2 · ：接收脉冲时间（单位为 ms）

D · ：结果存放位置，占用连续 3 个装置

例：



- 将 S1 · 指定的输入脉冲在 S2 · 指定的时间（单位为 ms）内计数，将其结果存入 D · 指定的软元件中。
- 通过反复操作，能在 D · 中得到脉冲密度（即与旋转速度成比例的值）。D · 占有 3 点的软元件。
- 在图例中，X010 置 ON 时，D1 对 X000 的 OFF→ON 动作计数，100ms 后将其结果存入 D0。随之 D1 复位，再次对 X000 的动作计数。
- D2 用于测定剩余时间。
- 在此被指定的输入 X000~X005 不能与高速计数器及中断输入重复使用。
- 输入 X000~X005 的 ON/OFF 的最大频率与 1 相高速计数同样处理。且与高速计数、FNC57 (PLSY) 以及 FNC59 (PLSR) 指令同时使用时，必须将这些处理频率合计值限

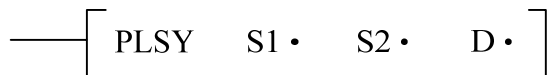
6. 应用指令解说

制在规定频率以下。

F 57 PLSY 脉冲输出

F		PLSY	脉冲输出								S1 ·	S2 ·	D ·				
	57		D														
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*	
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*	
D ·		*															

指令格式：



S1 · : 脉冲输出频率

16 位指令：1~32,767Hz

32 位指令：1~100,000Hz

S2 · : 脉冲输出数目

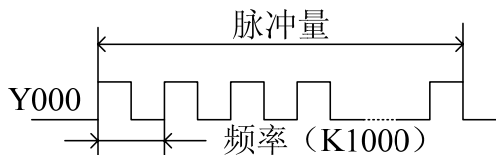
16 位指令：1~32,767 个

32 位指令：1~2,147, 483,647 个

D · : 脉冲输出软元件， 仅可指定 Y000 和 Y001（请使用晶体管输出的控制器）

※ 1: BSP01 SR 机种不支持该指令

例：



- 是以指定的频率生产定最脉冲的指令，在指令执行中更改 S1 · 指定的字软元件的内容，输出频率也随之发生变化。
- S2 · 的值指定为零时，则对产生的脉冲不做限制。在指令执行过程中，变更 S2 · 指定的字软元件后，将从下一个指令驱动开始执行变更加内容。
- X010 为 OFF 后，输出停止，再次置 ON 时，从初始状态开始动作，发出连续脉冲时，X010 为 OFF，Y000 也为 OFF。
- 脉冲的占空比为 0.5，输出控制不受扫描周期的影响。采用中断处理。
- 设定脉冲发完后，执行结束标志 M8029 动作。
- 控制器请使用晶体管输出。
- 将 2 个 FNC57(PLSY)命令或是 2 个 FNC59(PLSR)命令同时进行程序编辑，则可个别对 Y000 及 Y001 作独立的脉波输出。
- 对 1 个 FNC57(PLSY)命令及 1 个 FNC59(PLSR)命令同时进行程序编辑，则可个别对

6. 应用指令解说

Y000 与 Y001 作独立的脉波输出。

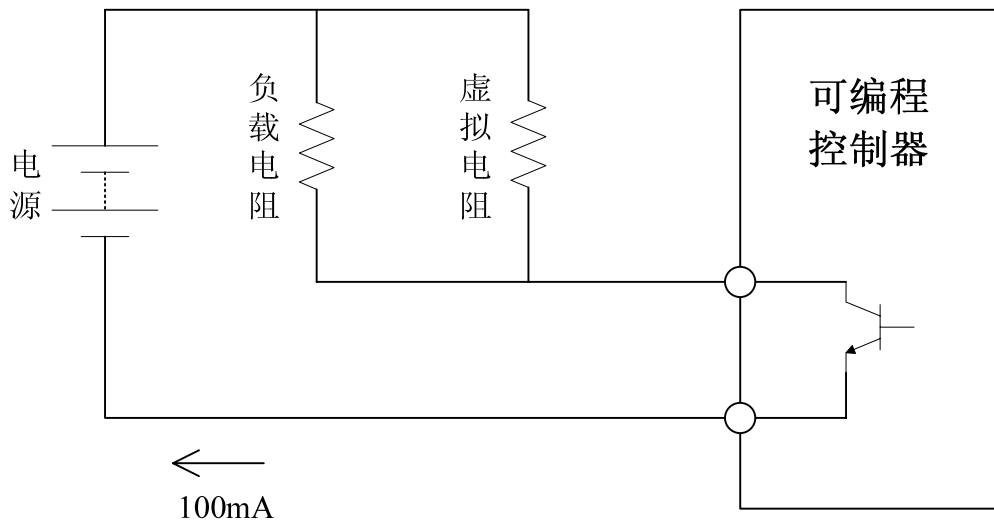
<注意事项>

1. 请勿同时驱动使用同一输出继电器(Y000 或 Y001)的脉波输出命令。若同时驱动,会造成双重线圈的使用而无法以正常机能运作。

2. 脉波输出命令的驱动接点 OFF 之后的再驱动请于下列条件成立后执行。

条件: 前一次所驱动的脉波输出命令之「脉波输出中监视器(Y000:[M8147], Y001:[M8148])OFF 之后, 需经过 1 个演算周期以上才可再驱动。

由于脉波输出命令的再驱动必须要有 1 次以上的 OFF 演算, 若是比上记条件较早作再驱动时, 则最初的命令实行(演算)会产生「演算异常」后, 第 2 次的命令再度驱动时再开始脉波输出。



- 不可与 FNC58(PWM)命令所指定的输出编号重复使用。
- FNC58(PWM)命令已指定的输出编号, 不可再与其重复。

6. 应用指令解说

F58 PWM 脉宽调制

F					脉宽调制											S1·	S2·	D·		
58		PWM																		
	位元件				字元件															
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z				
S1·					*	*	*	*	*	*	*	*	*	*	*	*	*			
S2·					*	*	*	*	*	*	*	*	*	*	*	*	*			
D·		*																		

指令格式：

$$\text{——} \left[\text{PWM} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \right]$$

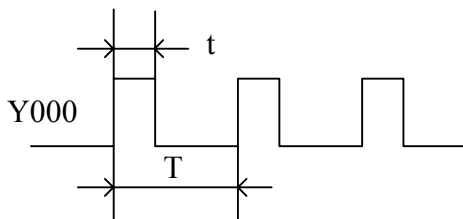
S1· : 脉冲输出宽度, $t=0\sim 3000\text{ms}$

S2· : 脉冲输出周期, $T=1\sim 3000\text{ms}$, 但 $\text{S1} \cdot \leq \text{S2} \cdot$

D· : 脉冲输出软元件, 仅可指定 Y000 和 Y001 (请使用晶体管输出的控制器)

※ 1: BSP01 SR 机种不支持该指令

例：



- 该输出的 ON/OFF 可进行中断处理执行
- 在上例中, 使用 D10 的内容超过 50 时, 就出现错误。
- X010 置于 OFF 时, Y000 为 OFF。

《注意事项》

- PLC 请使用晶体管输出, 而为了高频率脉波输出, 如前页所记载的请充分提供负载电流。(FNC57 (PLSY))
- 与 FNC57 (PLSY) 或 FNC59 (PLSR) 指令所指定输出编号不可重复使用。

6. 应用指令解说

F59 PLSR 带加减速脉冲输出

F		PLSR	带加减速脉冲输出													S1 · S2 · S3 · D ·
59	D															
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*
S3 ·					*	*	*	*	*	*	*	*	*	*	*	*
D ·		*														

指令格式:

—— [PLSR S1 · S2 · S3 · D ·]

S1· : 脉冲输出的最大频率

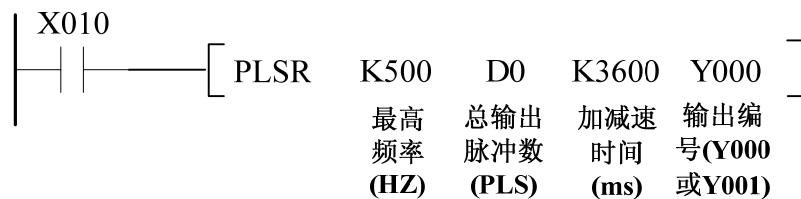
S2· : 输出脉冲的总数量

S3· : 加减速时间 ms

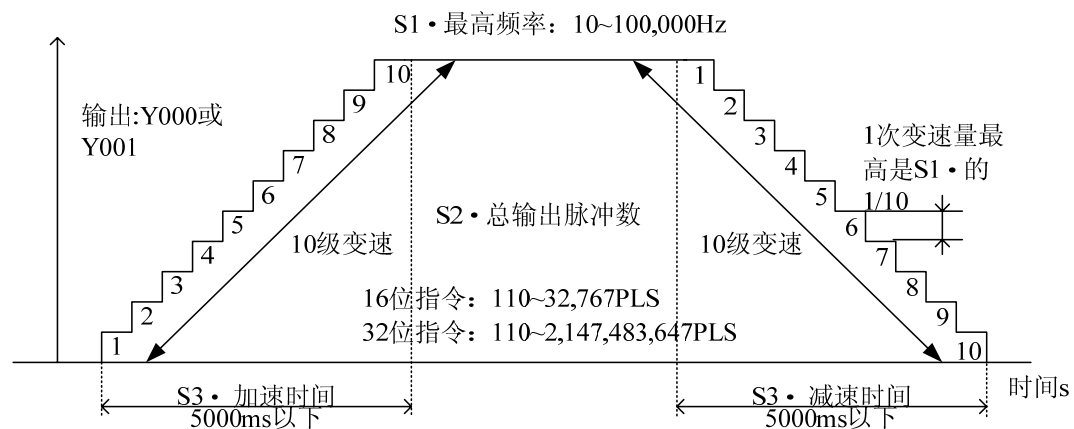
D· : 脉冲输出软元件, 仅可指定 Y000 和 Y001 (请使用晶体管输出的控制器)

※ 1: BSP01 SR 机种不支持该指令

例:



- 带加速减速功能有定尺寸传送用的脉冲输出指令。
针对指定的最高频率, 进行定加速, 在达到所指定的输出脉冲数后, 进行定减速



- 各操作数的设定内容如下：

S1· 最高频率：

可设定范围：10~100,000 (Hz)

频率以 10 的倍数指定。

最高频率指定值的 1/10 可作为减速时的一次变量 (频率)，因此，请设定在步进电机等不失速的范围之内。

S2· 总输出脉冲数：

可设定范围：16 位运算，110~32,767 (PLS)

32 位运算时，110~2147483647 (PLS)

设定不满 110 时，脉冲不能正常输出。

使用 **DPLSR** 指令时，在此例中 (D1、D0) 作为 32 位设定值处理。

S3· 加减速时间：

可设定范围：5000 (ms) 以下，但请遵照以下①~③条件。

加速时间和减速时间以相同值动作。

①加减速时间请设在可能性编程控制器的扫描时间最大值的 10 倍以上，指定不到 10 倍时，加减速时序不一定。

②作为加减速时间可设定的最小值公式如下：

$$S_3 \geq \frac{90000}{S_1} \times 5$$

③作为加减速时间可设定的最大值的公式如下：

$$S_3 \leq \frac{S_2}{S_1} \times 818$$

D· 脉冲输出号码：

· 只能指定 Y000 或 Y001。

· 输出一定为晶体管输出。

- 该命令的输出频率为 10~100,000Hz，最高速度、加减速时的变速速度超过此范围时，自动在范围值内调低或进位
- 输出控制不受扫描周期影响，进行中断处理。
- X010 为 OFF 时，中断输出，再度置于 ON 时，从初始动作开始。
- 在指令执行中即使改写操作数，运转也不反映。变更内容从下一次指令驱动开始有效。
- 设定的脉冲输出完毕时，执行完毕标志 M8029 置 ON。

F60~F69 方便指令

方便指令

功能编号	助记符	名称	页码
F60	IST	状态初始化	1
F61	SER	数据查找	6
F62	ABSD	凸轮控制	8
F63	INCD	凸轮控制	10
F64	TTMR	示教定时器	12
F65	STMR	特殊定时器	13
F66	ALT	ON/OFF 交替输出	14
F67	RAMP	斜坡信号	16
F68	ROTC	旋转工作台控制	17
F69	SORT	数据排序	18

6. 应用指令解说

可以再次驱动。

IST 指令的特殊辅助继电器

IST 指令使用的辅助继电器可分为根据自身状态自动控制的和有必要根据运行准备和控制目的用程序控制的两部分。

IST 指令自动控制

转移禁止 M8040

该辅助继电器工作时，所有的状态转移被禁止

搬运：M8040 经常工作。

原复，循环一次，从按下停止按钮直到按起动按钮一直保持运作

单步：M8040 经常工作。但只有按了起动按钮时变为不动作，开始传送。

其他：PLC STOP→RUN 切换时，按起动按钮解除。

转移开始：M8041

从初始状态 S2 到下一个状态的传送条件的辅助继电器。

手动、原复：不动作

单步、循环一次：只在按下起动按钮时动作。

连续按起动按钮时保持动作，按停止按钮时解除。

起动脉冲：M8042

仅在按起动按钮时瞬时动作。

原复完毕 M8043

在原复模式中，机器返回原点时，通过用户的程序使用该特殊继电器动作。

原点条件 M8044

检测机器的原点条件，驱动该特殊继电器。在全部模式是成为有效信号

全部输出复位禁止 M8045

若在手动、原复、自动模式间进行转换时，机器不在原点位置时，则执行全部输出与动作状态的复位。但若已驱动 M8045，只在动作状态复位。

STL 监控有效 M8047

驱动 M8047 后，现正动作中的状态序号 S0~S899 按从小到大的顺序存入特殊辅助继电器 D8040~D8047 中，因此，可在监控 8 点的动作状态序号。此外，若这些状态的任何一个动作，特殊辅助继电器 M8046 将动作。

模式选择输入的分配

使用 IST 命令，各模式的输入如下分配的连续输入编号。当不使用连续编号和一部份模式省略时，可变成下列的辅助继电器来指定输入的初始组件编号。

X030：个别操作

X031：原点复归

X032：步进

X033：一次运转

X034：连续运转

X035：原点复归开始

X036：自动开始

6. 应用指令解说

X037: 停止

X030~X034 请勿同时动作。(可利用选择开关)

输入不为连续编号

例: X030: 各个操作

X035: 原点复归

X033: 步进

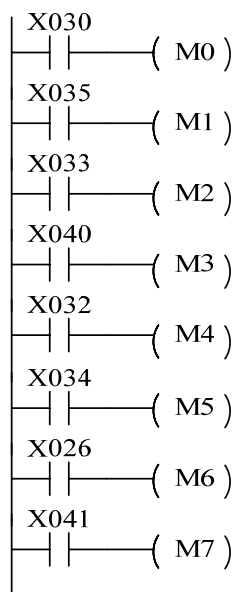
X040: 一次运转

X032: 连续运转

X034: 原点复归开始

X026: 自动开始

X041: 停止



只有连续/原点复归模式时

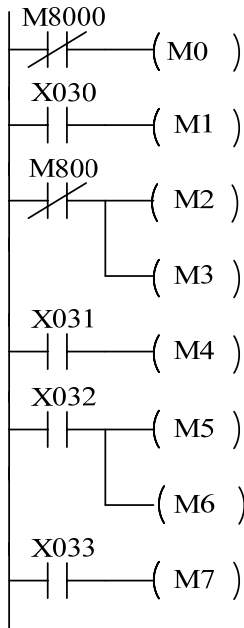
例: X030: 原点复归

X031: 连续运转

X032: 自动开关和原点复归开始

X033: 停止

6. 应用指令解说



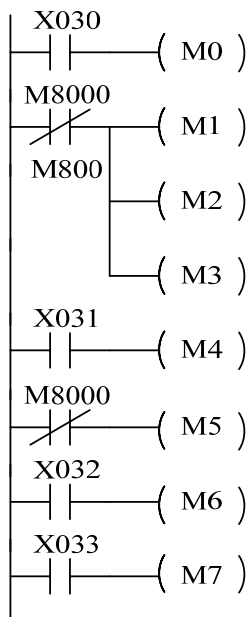
只有连续/个别模式

例：X030：各个操作

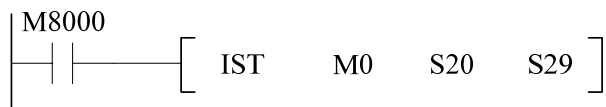
X031：连续运转

X032：自动开关

X033：停止



此例中使用 M0 作为模式指定起始输入



IST 命令使用辅助继电器，有分为命令本身的情况而自我产生的动作，和根据运转准备和控制目的来写的程序。

IST 命令的自动控制的

<移行禁止 M8040>

此辅助继电器动作时，全部的状态移行被禁止。

各别:常时 M8040 动作。

原复一次:从停止按钮按下到开始按钮按下为止，动作保持。

个进:常时 M8040 动作，但只有开始按钮按下时，不动作，且移行。

其它:PLC 在 STOP—RUN 切换时保持其动作，开始按钮按下时被解除。在移行禁止状态时，状态内的输出继续保持。

<移行开始 M8041>

初始状态 S2 移行到下一状态的辅助继电器。

个别，原复:不动作。

步进，一次:只有在开始按钮时动作。

连续:开始按钮按下时，动作保持，停止按钮按下时，动作解除。

<开始脉波 M8042>

开始按钮按下时只动作一瞬间。

有关于此控制，请参考次页。

用顺控程序驱动的

<原复完了 M8043>

原复模式在机构原点复归完毕后，要由使用者在程序中，让此特殊辅助继电器动作。

<原点条件 M8044>

机械原点条件检出，来驱动此特殊辅助继电器，此为全模式有效信号。

<全部输出清除禁止 M8045>

个别、原复、自动模式间的切换，机械不在原点位置时，全部输出和动作状态全部清除，M8045 驱动时，只有动作状态被清除。

<STL 监视有效 M8047>

当 M8047 驱动时，现在在动作的状态号码(S0~S899)，会以较小的编号依序，储存在 D8040~D8047，可监视 8 点动作状态的编号。而此状态点若有任一点动作时，特殊辅助继电器 M8046 会动作。

下记的 PLC 电路内以外为固定的电路部分电路的控制内容与其它程序要配合。

6. 应用指令解说

F61 SER 数据查找

F		SER		数据查找								S1·	S2·	D·	n				
61	D		P																
	位元件								字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S1·							*	*	*	*	*	*	*	*					
S2·					*	*	*	*	*	*	*	*	*	*	*	*			
D·								*	*	*	*	*	*	*					
n					*	*													

指令格式：

—— [SER S1· S2· D· n]

S1· : 数据查找资料区的起始元件序号

S2· : 指令查找的数据内容

D· : 存放检索结果表的起始装置，占用连续 5 点。

n: 被比较的资料区的长度，n=1~256（16 位元指令），n=1~128（32 位指令）。

本指令为对数据表进行相同数据检索、最大值、最小值检索的指令。

例：

X000


检索表的构成和数据例：

被检索元件	被检索数据例	比较数据	数据位置	最大值	相同	最小值
D100	D100=K100	D0=K100	0			
D101	D101=K111		1			
D102	D102=K100		2		相同	
D103	D103=K98		3			
D104	D104=K23		4			
D105	D105=K66		5			最小
D106	D106=K100		6			相同
D107	D107=K95		7			
D108	D108=K210		8		最大	
D109	D109=K88		9			

6. 应用指令解说

检索结果表

元件号	内容	备注
D50	3	相同数据个数
D51	0	相同数据位置（最初）
D52	6	相同数据位置（最始）
D53	5	最小值最终位置
D54	8	最大值最终位置

- 进行的是代数上的大小比较，即有符号数的比较。
- 最大值、最小值有多个时，显示后面的位置。
- 使用 32 位指令时 D· 同样以 32 位为单位存储检索结果。
- 在以 D· 为起始的 5 点元件中，如上表所示，存入相同数据及最小值，最大值的位置。不存在相同数据时，上例中 D50~D52=0。

注：当 S1·、S2· 及 D· 范围越界时，会出现错误。

6. 应用指令解说

F62 ABSD 凸轮控制绝对方式

F		ABSD	凸轮控制绝对方式								S1·	S2·	D·	n		
62	D															
	位元件					字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·							*	*	*	*	*	*	*	*		
S2·												*				
D·		*	*	*												
n					*	*										

注：本指令为 16 位指令时，n=4，为 32 位指令时，n=8。

本指令使用次数：一次（但索引可修饰）

指令格式：

—— [ABSD S1· S2· D· n]

S1·：比较表起始装置，运算元指定为 KnX、KnY、KnM、KnS 时，16 位元指令须指定 K4，32 位元指令须指定 K8，且 X、Y、M、S 编号须为 16 的倍数

S2·：计数器编号。16 位元指令须指定 C0~C199，32 位元指令须指定 C200~C255

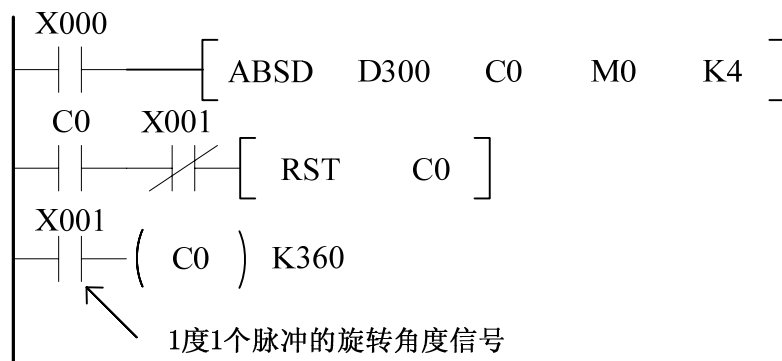
D·：比较结果输出起始编号

n：多段比较的组数，n=1~64

※ 1：对于 BSP01 SR 机种 S2·计数器编号，16 位元指令须指定 C0~C99，32 位元指令须指定 C220~C255

本指令为对应于计数器当前值产生多个输出波形的指令。以平台旋转一周期间控制辅助继电器 M0~M3 ON/OFF 为例，加以说明。

例：



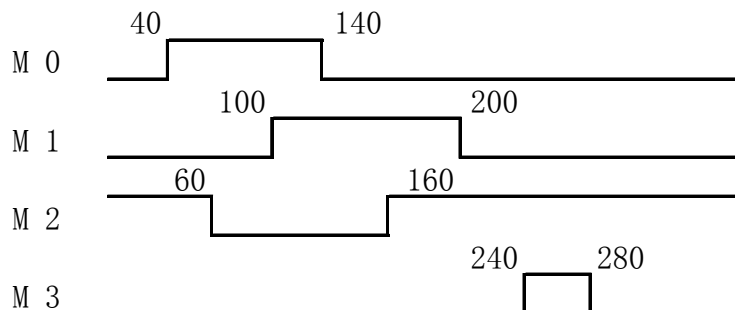
预先使用传送指令将以下数据写入 D300~D307 中。

上升点	下降点	对象输出
D300= 40	D301=104	M0
D302=100	D303=200	M1
D304=160	D305= 60	M2

6. 应用指令解说

D306=240	D307=280	M3
----------	----------	----

X000 置 ON 时，M0~M3 如以下变化。各上升点/下降点可根据 D300~D307 的数据的更改而变化。



- 由 n 值决定输出对象的点数。
- X000 置于 OFF 时，波形输出保持不变。
- 使用 DABSD 命令时，S2·可指定高速计数器。但是输出波形会由于扫描周期的影响而造成响应的滞后。需高速应答性时，请使用 HSZ 命令的高速比较机能。

6. 应用指令解说

F63 INCD 凸轮控制增量方式

F		INCD		凸轮控制增量方式												S1 • S2 • D • n			
63																			
	位元件				字元件														
	X	Y	M	S	K	H	K _Δ X	K _Δ Y	K _Δ M	K _Δ S	T	C	D	W	V	Z			
S1 •							*	*	*	*	*	*	*	*					
S2 •												*							
D •		*	*	*															
n					*	*													

注：本指令为 16 位指令时， $\Delta=4$

指令格式：

$$\text{——} \left[\text{INCD} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

S1 • : 比较起始元件，指定为 KnX,KnY,KnM,KnS 时，须指定 K4，且 X, Y, M, S 编号须为 16 的倍数。

S2 • : 计数器编号，16 位元指令须指定 C0~C198，会占用 2 个连续编号计数器。

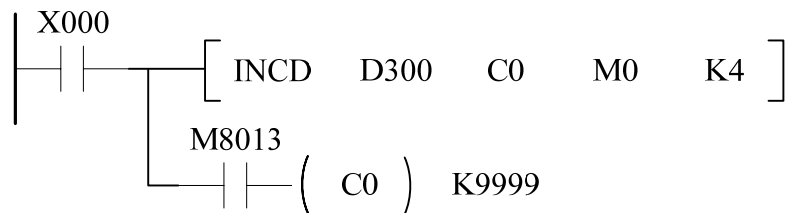
D • : 比较结果的起始编号

n: 多段比较的组数 n=1~64

※ 1: 对于 BSP01 SR 机种，S2 • 计数器编号，16 位元指令须指定 C0~C98，会占用 2 个连续编号计数器。

本指令为用一对计数器产生多个输出波形的指令。

例：



根据时序表，对控制 n=4 点 M0~M3 的例子予以说明。

● 预先使用传送命令将下列数据写入 S1 • 中。

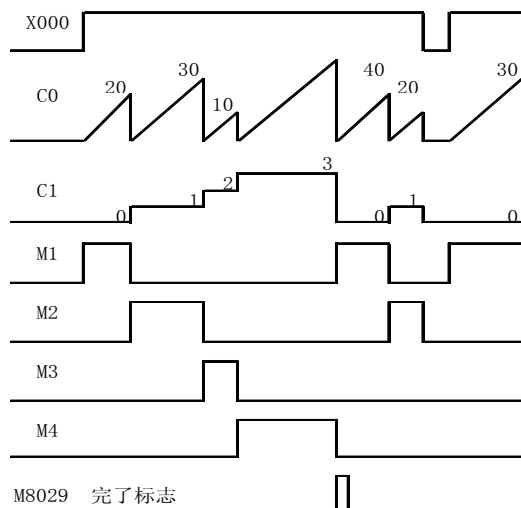
D300=20

D301=30

D302=10

D303=40

6. 应用指令解说



- 计数器 C0 达到按 D300~D303 设定的值时，按顺序自动复位。
- 工作计数器 C1 计数其复位次数。
- 对应工作计数器 C1 的当前值，M0~M3 按顺序工作。
- n 指定的一系列输出动作结束时，标志 M8029 动作，再次返回进行同样的工作。
- X000 置于 OFF 时，C0、C1 被清除，M0~M3 也 OFF。再次将 X000 置于 ON 时，从初始开始工作。

6. 应用指令解说

F64 TTMR 示教定时器

F		TTMR		示教定时器										D·	n	
64																
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D·													*	*		
n					*	*										

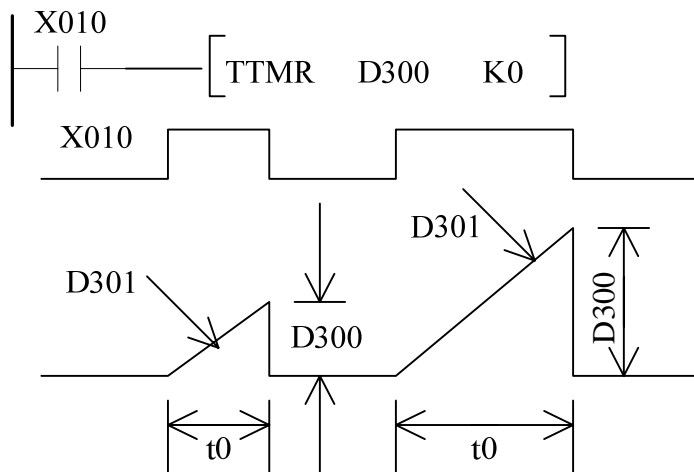
指令格式：

— [TTMR D· n]

D·：储存按钮开关 ON 时间的元件编号，D· 会占用 2 个连续编号的装置

n：倍数设定，n=0~2

例：



- 用 D301 测定的按钮 X010 的按动时间并乘以由 n 指定的倍率存入 D300 中。由此通过按钮可以调整定时器的设定时间。
- 按钮 X010 的按动时间为 $\tau 0$ 秒时，根据 n 的值，实际的 D300 的值如下：

n	D300
K0	$\tau 0$
K1	$10 \tau 0$
K2	$100 \tau 0$

- X010 为 OFF 时，D301 复位，D300 不变。

6. 应用指令解说

F65 STMR 特殊定时器

F		STMR		特殊定时器												S·	m	D·	
65	D		P																
	位元件				字元件														
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S·											*								
m					*	*													
D·		*	*	*															

指令格式：

—— [STMR S· m D·]

S·：定时器编号，指定范围：T0~T199

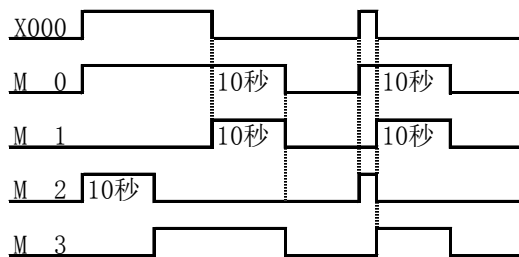
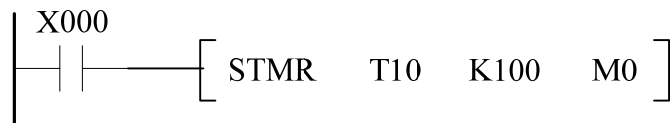
m：计时器设定值，单位 100ms，范围：1~32767

D·：输出元件的起始编号，占用 4 个连续编号装置

※ 1：对于 BSP01 SR 机种 S·计数器编号，指定范围：T0~T39，T196~T199

本指令为可简洁制作延时定时器、单触发定时器、闪烁定时器的指令

例：



- m 指定的值为 S·指定的定时器的设定值，此例中为 10 秒。
- M0 为延时定时器。
- M1 为输入 ON→OFF 后的单触发定时器。
- M2、M3 为闪烁用。
- X000 置于 OFF，设定时间后 M0、M1、M3 变为 OFF，T10 也复位。
- 在此使用的定时器在其他一般的电路中请不要重复使用。

6. 应用指令解说

F66 ALT ON/OFF 交替输出

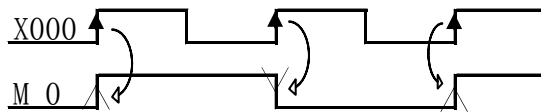
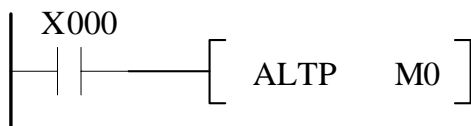
F		ALT			ON/OFF 交替输出							D·				
66				P												
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D·		*	*	*												

指令格式：

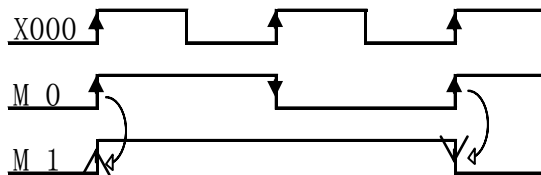
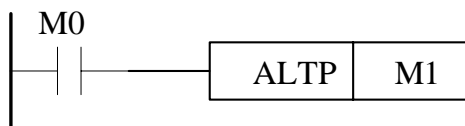
—— [ALT D·]

D·：目的地元件

例 1：

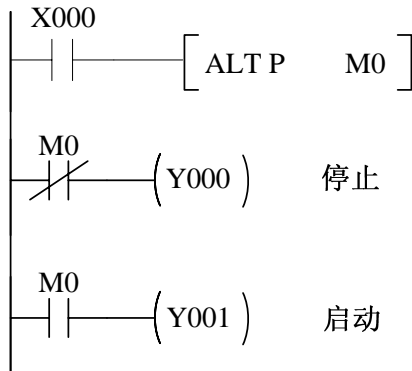


- 驱动输入 X000 每次 OFF→ON 的变化时，M0 反向。
使用连续执行型指令时，每个运算周期都反向动作，请注意。
- 上图的 M0 作为输入，用 ALTP 指令驱动 M1 时，就能得到多级的分频输出。



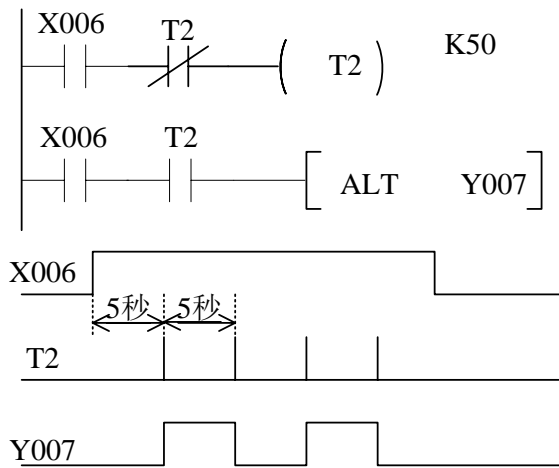
反复动作的应用:

《由 1 个输入起动/停止》



- 按下按钮 X000 时，起动输出 Y001 动作。
- 再次按下按钮 X000 时，停止输出 Y000 动作。

《闪烁动作》



- 输入 X006 置于 ON 时，定时器 T2 的接点每隔 5 秒瞬时动作。
- T2 的接点每次 ON 时，输出 Y007 交替 ON/OFF。

6. 应用指令解说

F67 RAMP 斜坡信号

F					斜坡信号											S1 · S2 · D · n
67		RAMP		P												
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·													*	*		
S2 ·													*	*		
D ·													*	*		
n					*	*										

指令格式：

—— [RAMP S1 · S2 · D · n]

S1 · : 斜坡信号初始值元件

S2 · : 目标值元件

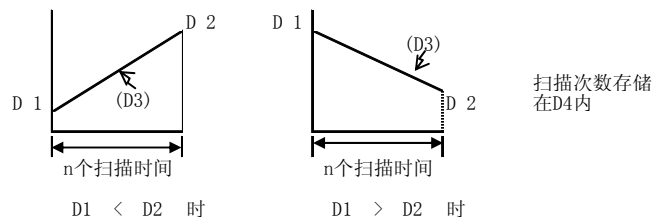
D · : 过程值

n: 扫描次数, n=1~32767

例：



- 预先把所定的初始值与目标值写入 D1, D2 中, 若 X000 置于 ON 时, D3 的内容从 D1 的值到 D2 的值慢慢变化。其移动时间为 n 个扫描时间。

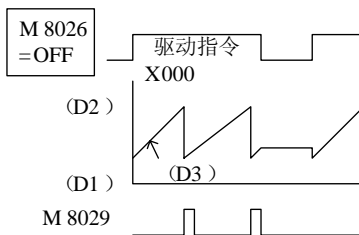
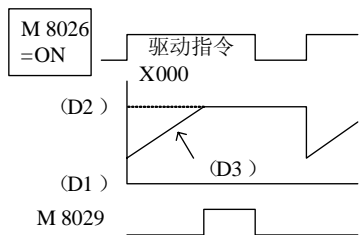


- 把所定的扫描时间（稍长于实际程序的扫描时间）写到 D8039 中, 并驱动 M8039, 可编程控制器为恒定扫描运行模式, 该值例如在 20ms 时, 上例中经 20 秒 (D3) 值由 D1 向 D2 变化。
- 运作途中 X000 置于 OFF 时, 成为运行中断状态, 再次将 X000 置于 ON 时, D4 被清除, 再从 D1 开始动作。
- 执行完毕后, 标志 M8029 置 ON, D3 的值回到为 D1 的值。
- 将该指令与模拟输出相结合, 可以输出缓冲起动/停止指令。
- X000 在 ON 的状态下 RUN 开始时, D4 请预先清除, (D4 为停电保持时)。

模式标志的动作

在可编程控制器中, 根据模式标志 M8026 的动作, D3 的内容变化如下：

6. 应用指令解说



6. 应用指令解说

F68 ROTC 旋转工作台控制

F 68	ROTC			旋转工作台控制								S · m1 m2 D · n				
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·													*	*		
m1					*	*										
m2					*	*										
D ·		*	*	*												

指令格式：

$$\text{——} \left[\text{ROTC} \quad \text{S} \cdot \quad \text{m1} \quad \text{m2} \quad \text{D} \cdot \right]$$

S: 计数用寄存器，占用 3 个连续编号装置。

m1: 工作台分割线，m1=2~32,767

m2: 低速区间数，m2=0~32,767 (m1 ≥ m2)

D: 信号输出起始点，占用 8 个连续编号装置。

例：

$$\begin{array}{|c} \text{X010} \\ \hline \text{—|—|—} \end{array} \left[\text{ROTC} \quad \text{D200} \quad \text{K10} \quad \text{K2} \quad \text{M0} \right]$$

为取放 m1=10 分割的旋转工作台上的工件，按照要求取放的窗口就近旋转工作台的指令。

- 请设置为检测工作台正转/反转的 2 相开关及工件 0 号转到 0 号窗口时动作的开关 X002，用 X000~X002 驱动 M0~M2(X 和 M 的起始序号可随意)。
- D200 占用 3 个连续编号装置，D200 作为计数寄存器使用，D201 作为调用窗口号码设定，D202 作为调用工件号码设定。D201 和 D202 在传送指令前预先设定。
- M0: A 相信号
M1: B 相信号
M2: 0 点检测信号
M3: 高速正转
M4: 低速正转
M5: 停止
M6: 低速反转
M7: 高速反转

X010 置于 ON 驱动此指令时，可以自动得到 M3~M7 的结果。X010 置于 OFF 时，M3~M7 为 OFF

- X010 为 ON、0 点检测信号 (M2) 为 ON 时，计数用寄存器 D200 的内容清为零。需要预先进行该清除操作后开始运行。

6. 应用指令解说

		5	150	50	45
--	--	---	-----	----	----

- 起始数据寄存器由 S · 指定
- 在第一行可输入人员号码等连续号码，用其内来判断原来的行号。

《D 0=K2 执行指令时》

列号 行号	1	2	3	4
	人员号码	身高	体重	年龄
1	D200 4	D205 100	D210 20	D215 8
2	D201 1	D206 150	D211 45	D216 20
3	D202 5	D207 150	D212 50	D217 45
4	D203 3	D208 160	D213 70	D218 30
5	D204 2	D209 180	D214 50	D219 40

《D 0=K3 执行指令时》

列号 行号	1	2	3	4
	人员号码	身高	体重	年龄
1	D200 4	D205 100	D210 20	D215 8
2	D201 1	D206 150	D211 45	D216 20
3	D202 2	D207 180	D212 50	D217 40
4	D203 5	D208 150	D213 50	D218 45
5	D204 3	D209 160	D214 70	D219 30

- 运算结果的数据以 D · 指定的元件为起始，占有 $m1 \times m2$ 个数据寄存器，S · 与 D · 为同一元件时，请特别注意运行完毕前，不要改变 S · 的内容。
- 该指令的执行需 $m1$ 次扫描周期，数据排序完毕，标志 M8029 动作。

F70~F79 外围设备 I-O 指令

外围设备 I-O 指令

功能编号	助记符	名称	页码
F70	TKY	十字键输入	1
F71	HKY	十六键输入	2
F72	DSW	数字开关输入	4
F73	SEGD	七段码	6
F74	SEGL	七段	8
F75	ARWS	方向开关	10
F76	ASC	ASCII 码转换	11
F77	PR	ASC II 码打印	12

6. 应用指令解说

F70 TKY 十字键输入

F					十字键输入											S·	D1·	D2·
70	D	TKY																
	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		
S·	*	*	*	*														
D1·								*	*	*	*	*	*	*	*	*		
D2·		*	*	*														

指令使用次数：1次（但可以使用变址寄存器修饰）

指令格式：

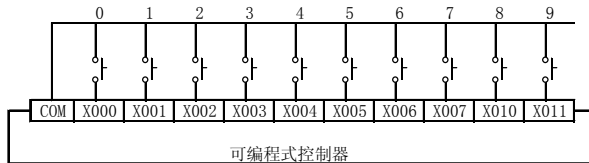
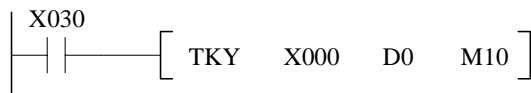
$$\text{---} \left[\text{TKY} \quad \text{S} \cdot \quad \text{D1} \cdot \quad \text{D2} \cdot \right]$$

S·：按键输入起始元件，占用连续10点

D1·：按键输入值存放处

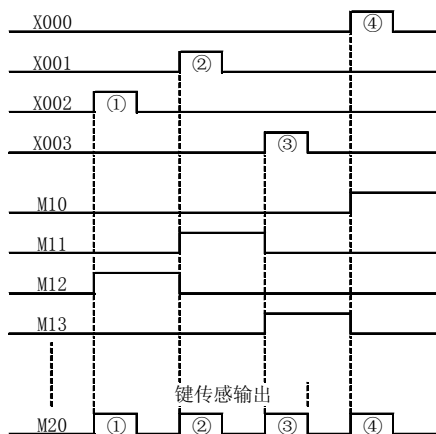
D2·：按键输出信号

例：



- 以下图所示①②③④的顺序输入十字键，则D0的内容成为2,130。输入的最大值为9,999，超过时则发生溢出。(实际上D0的内容为2进制方式储存)
- 当使用 **D** TKY 命令时，则组合D1,D0 2个寄存器使用，当数值超过99,999,999时，也发生溢出。
- 从按下X002开始，至按下其它的按键为止，M12保持ON，其它的按键亦相同。
- 相同的，对应X000~X011的动作，M10~M19会动作。
- 按任何按键时，在按下按键的期间，按键检出M20会ON，同时按多个键时，以最先按下的键为有效。
- 当输入X030 OFF时，D0的内容不会改变，但是M10~M20全部成OFF。

6. 应用指令解说



6. 应用指令解说

F71 HKY 十六键输入

F					十六字键输入										S · D1 · D2 · D3 ·	
71	D	HKY														
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·	*															
D1 ·		*														
D2 ·											*	*	*	*	*	*
D3 ·		*	*	*												

指令使用次数：1 次（但可以使用变址寄存器修饰）

指令格式：

$$\text{——} \left[\text{HKY} \quad \text{S} \cdot \quad \text{D1} \cdot \quad \text{D2} \cdot \quad \text{D3} \cdot \right]$$

S · ： 按键输入起始元件，占用连续 4 点

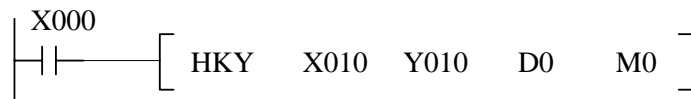
D1 · ： 按键扫描输出起始元件，占用连续 4 点

D2 · ： 按键输入值存放处

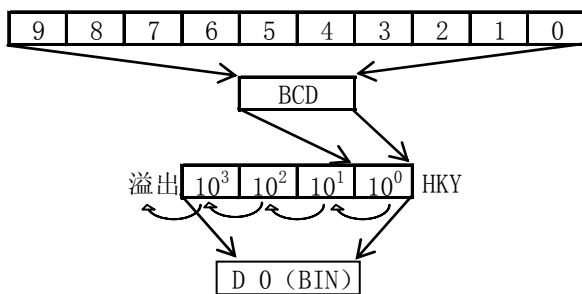
D3 · ： 按键输出信号

本指令是用 16 键写入数值和输入功能的指令。

例：



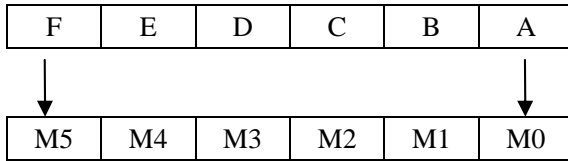
《数字键》



- 每次按数字键，以 BIN 形式向 D0 存入上限值为 9,999 的数值，超出此值则溢出。
- 使用 DHKY 指令时，D1、D0 在 0~99999999 间有效。
- 按多个键时，只有先按下的键有效。
- Y010~Y013 一次循环扫描动作后，执行完毕标志 M8029 工作。

6. 应用指令解说

《功能键》

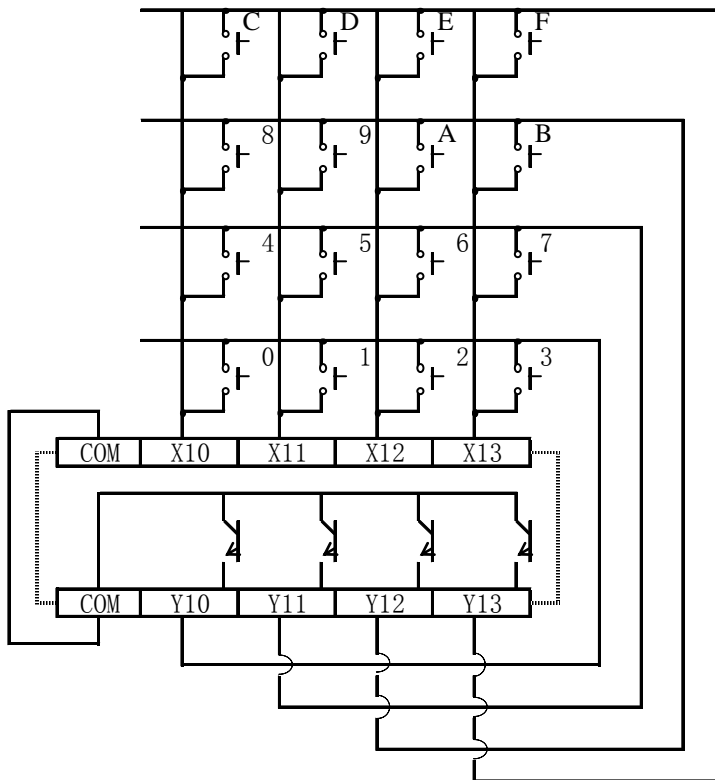


- 按 A 键时，M0 保持 ON，再按 D 键时，M0 变 OFF，而 M3 保持 ON。
- 多个键操作时，先按的键有效。

《键检测输出》

- 按下 A~F 的任何一键时，只在按下期间 M6 工作。
- 按下 0~9 的任何一键时，只在按下期间 M7 工作。
- 驱动输入 X000 置于 OFF 时，D0 不变化，而 M0~M7 变 OFF。

外部线路



标识 M8167 的功能：

M8167=ON 时，则 HKY 指令可以输入 0~F 的 16 进制值；

M8167=OFF 时，则 HKY 指令 A~F 当成功能键使用。

补充说明：

本指令执行时，必须经过 8 次扫描时间才可有效的抓取一个按键的输入值，当扫描周期太长或太短都可能造成按键输入不良，因此，可运用下列技巧来克服。

1. 当扫描周期太短时，可能造成 I/O 来不及反应而无法读取按键的输入值，此时，可将扫描时间加以固定。

6. 应用指令解说

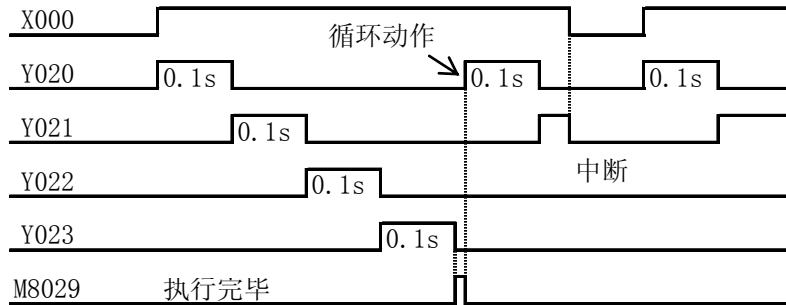
2. 当扫描周期太长时，可能会使按键反应变得迟钝，可将此指令写在时间中断副程式内，固定时间执行此指令。

6. 应用指令解说

注：当输入为非 BCD 码，本指令会出现错误。

《第一组输入》连接 X020~X023 的 4 位 BCD 码数字开关根据 Y020~Y023 的顺序读入，并存入 D0 中。

《第二组输入》连接 X024~X027 的 4 位 BCD 码数字开关根据 Y020~Y023 的顺序读入，并存入 D1 中。（只有 n=2 时有效）

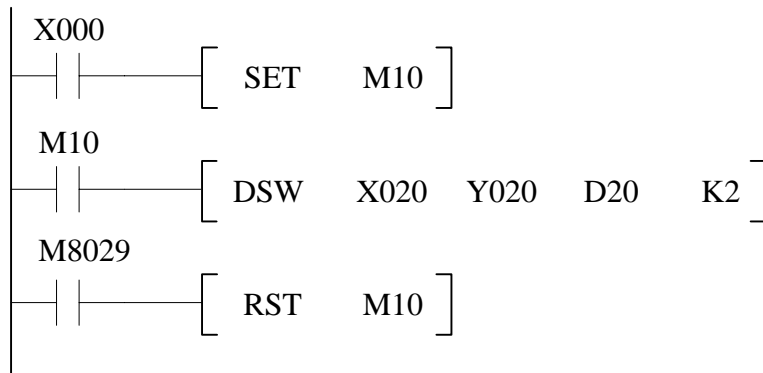


- X000 置于 ON 时，Y010~Y013 顺序工作（100ms），一次循环工作后，执行完毕标志 M8029 动作。
- 当需要连续读入 DSW 的值时，请使用晶体管输出的可编程控制器。

补充说明：

当 PLC 为继电器输出时，可使用下图所示方式来完成：

1. 当 X000=ON 时 DSW 指令被执行时，当 X000 变成 OFF 时，M10 会继续保持 ON，直到 DSW 指令的扫描端完成一次循环输出时才 OFF。
2. 条件接点 X000 使用按钮开关，X000 每按一次，DSW 指令所指定的扫描端会在一次循环输出完毕时，M10 才会复归成 OFF，指令才会停止执行，指拨开关资料会完整的读取。因此，此种情况下，即使扫描端使用继电器输出，继电器的寿命也不会因此缩短。



6. 应用指令解说

F73 SEGD 七段码译码

F		SEGD			七段码译码							S·		D·			
73				P													
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·					*	*	*	*	*	*	*	*	*	*	*	*	
D·							*	*	*	*	*	*	*	*	*	*	

指令格式：

—— [SEGD S· D·]

S·：译码的来源元件

D·：译码后的输出元件

将 S· 低 4 位指定的 0~F（十六进制数）数据译成七段码显示的数据存入 D· 中，D· 的高 8 位不变。

M8273 指定输出显示的逻辑关系。

当 M8273=OFF 时，

《七段码译码表》如下

源		七段组合 数字	预设定								表示的数字
16 进制数	位组合格式		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000	B0	0	0	1	1	1	1	1	1	0
1	0001	B5 B6 B1	0	0	0	0	0	1	1	0	1
2	0010	B4 B2	0	1	0	1	1	0	1	1	2
3	0011	B3	0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	B
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	D
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

位元件的起始(例如 Y000)
或字元件的最后位为 B0

6. 应用指令解说

当 M8273=ON 时，

《七段码译码表》如下

源		七段组合 数字	预设定							表示的数字	
16 进制数	位组合格式		B7	B6	B5	B4	B3	B2	B1		B0
0	0000	B0	1	1	0	0	0	0	0	0	0
1	0001	B5 B6 B1	1	1	1	1	1	0	0	1	1
2	0010	B4 B2	1	0	1	0	0	1	0	0	2
3	0011	B3	1	0	1	1	0	0	0	0	3
4	0100		1	0	0	1	1	0	0	1	4
5	0101		1	0	0	1	0	0	1	0	5
6	0110		1	0	0	0	0	0	1	0	6
7	0111		1	1	0	1	1	0	0	0	7
8	1000		1	0	0	0	0	0	0	0	8
9	1001		1	0	0	1	0	0	0	0	9
A	1010		1	0	0	0	1	0	0	0	A
B	1011		1	0	0	0	0	0	1	1	B
C	1100		1	1	0	0	0	1	1	0	C
D	1101		1	0	1	0	0	0	0	1	D
E	1110		1	0	0	0	0	1	1	0	E
F	1111		1	0	0	0	1	1	1	0	F

6. 应用指令解说

F74 SEGL 七段码时分显示

F 74	SEGL				七段码时分显示							S ·	D ·	n			
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S ·					*	*	*	*	*	*	*	*	*	*	*	*	
D ·		*															
n					*	*											

指令使用次数：2 次（但可以使用变址寄存器修饰）

指令格式：

$$\text{——} \left[\text{SEGL} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

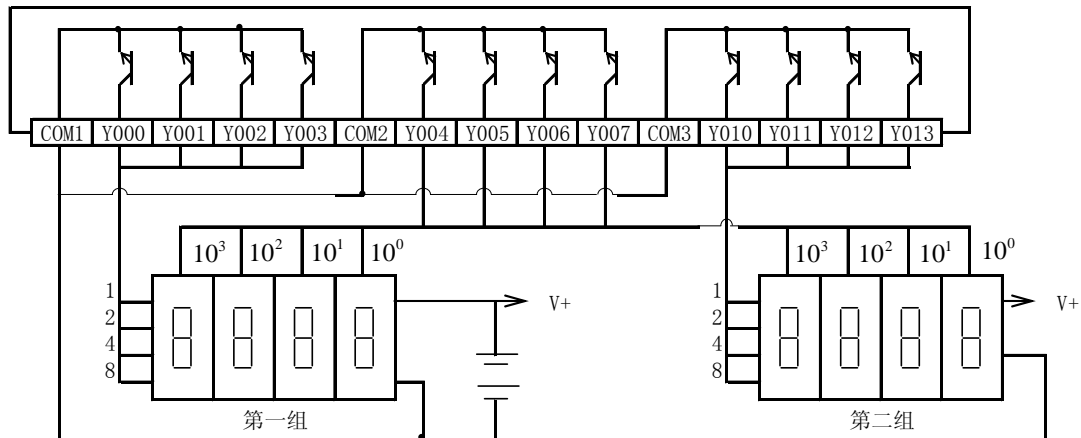
S · : 欲显示七段码的来源元件

D · : 七段码显示器扫描输出起始装置

n: 输出信号及扫描信号的极性设定, n=0~7

本指令为控制 4 位 1 组或 2 组的带锁存七段码的指令。

例：



《4 位 1 组时》n=0~3

- D0（虽然是二进制，但经过 BCD 换算，0~9,999 的范围有效）的 BCD 换算的各位依次（Y000~Y003）输出。
- 由选通脉冲信号（Y004~Y007）依次锁存 4 位 1 组的带锁存七段码。

《4 位 2 组时》n=4~7

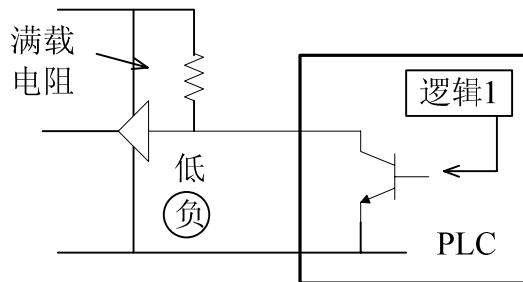
- 同样，D0 向（Y000~Y003）输出，D1 向（Y010~Y013）输出，D1、D0 均 BCD 变换为 0~9,999 时有效，否则出错。
- 选通脉冲信号（Y004~Y007）两组共同使用。
- 用该指令为进行 4 位（1 组或 2 组）的显示，需要运算周期 4 倍的时间，4 位数输出结

6. 应用指令解说

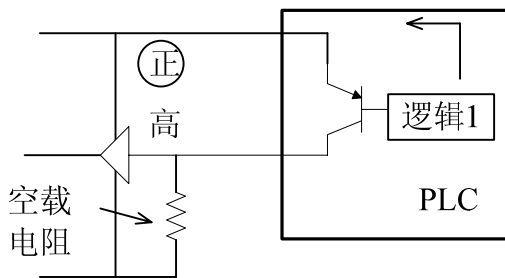
束后，完毕标志 M8029 动作。

- 该指令的驱动输入在 ON 时，执行反复动作，但在一系列的動作途中，驱动输入置于 OFF 时，停止动作，再驱动时从初始动作开始。
- 该指令与可编程控制器的扫描周期同时执行。为执行一系列的显示，可编程控制器的扫描周期需要 10ms 以上，不足 10ms 时，使用恒定扫描模式，请用 10ms 以上的扫描周期定时运行。
- 本可编程控制器的晶体管输出的 ON 电压约为 1.5V，7 段码请使用与此相应的输出电压。
- 参数 n 值的设定：用来设定晶体管为正极或负极性的回路，连接的七段显示器是一组 4 位数或二组 4 位数。

《可编程控制器的逻辑》



NPN 晶体管输出回路：当内部信号为 1 时，输出低电平，称此为负逻辑。



PNP 晶体管输出回路：当内部信号为 1 时，输出高电平，称此为正逻辑。

《7 段显示器的逻辑》

区分	正逻辑	负逻辑
资料输入	高电平变为 BCD 数据	低电平变为 BCD 数据
选通脉冲信号	高电平保持锁存的数据	低电平保持锁存的数据

《参数 n 的选择》

七段显示组数	一组				二组			
	相同		不同		相同		不同	
PLC 输出端与显示器资料输入端的逻辑	相同	不同	相同	不同	相同	不同	相同	不同
PLC 输出端与显示器资料扫描信号的逻辑	相同	不同	相同	不同	相同	不同	相同	不同
n	0	1	2	3	4	5	6	7

PLC 的晶体管输出逻辑与 7 段显示器的逻辑是否相同或者不同时，可通过参数 n 的设定值来相互匹配。

假设 PLC 的输出为负逻辑，7 段显示器的资料输入端也为负逻辑，而 7 段显示器的选通脉

6. 应用指令解说

冲信号为正逻辑，则如果是 4 位 1 组时， $n=1$ ，4 位二组时， $n=5$ 。

6. 应用指令解说

F75 ARWS 方向开关

F		ARWS	方向开关										S · D1 · D2 · n			
75	D															
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·	*	*	*	*												
D1 ·											*	*	*	*	*	*
D2 ·		*														
n					*	*										

指令使用次数：1次（但可以使用变址寄存器修饰）

指令格式：

$$\text{---} \left[\text{ARWS} \quad \text{S} \cdot \quad \text{D1} \cdot \quad \text{D2} \cdot \quad \text{n} \right]$$

S · ： 按键输入起始元件，占用连续4点

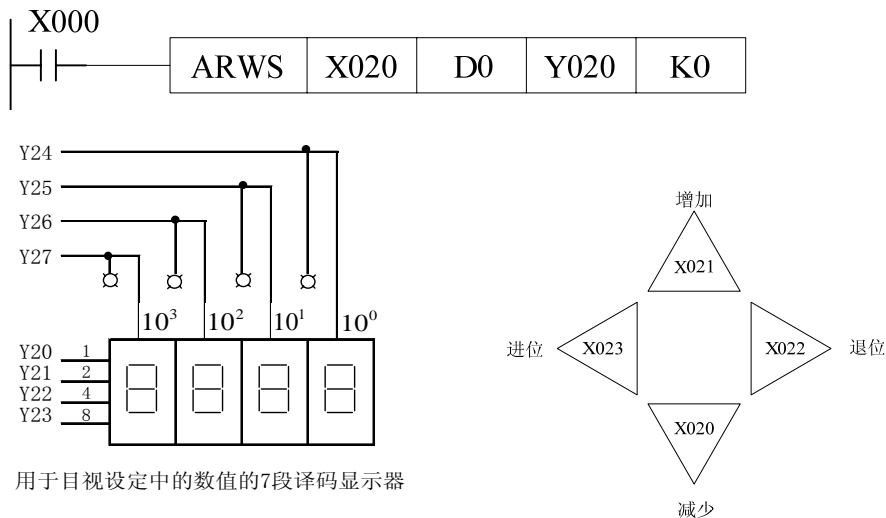
D1 · ： 欲通过箭头开关输入数据的元件

D2 · ： 七段显示器扫描输出起始元件，占8点，用于目视设定中的数值

n： 同指令 SEGL 中的 n，但 n 取值：0~3

本指令时通过位移动与各位数据增减用的箭头键盘开关输入数据的指令。

例：



用于目视设定中的数值的7段译码显示器

- D0 中存储的是 16 位二进制 BCD（换算到 0~9,999 有效）数据，但为方便起见，以下说明用 BCD 码
- 驱动输入 X000 置于 ON 时，指定为 10³ 位，每次按退位键输入时，指定位按 10³-10²-10¹-10⁰-10³ 变化。此外，每次按进位键输入时，指定位按 10³-10⁰-10¹-10²-10³ 变化。指定位可以根据选通脉冲信号（Y024~Y027）用 LED 显示。
- 对于被指定的位，每次按增加键 D0 的内容按 0-1-2-3-4-5-6-7-8-9-0-1 变化。按减少键时，按 0-9-8-7-----1-0-9 变化。其内容可以用七段显示器显示。

如上所述，该指令可以一边看显示器，一边将目的数值写入 D0 中。

补充说明：

指令所指定的输出点 Y20~Y27 必须使用晶体管输出。

6. 应用指令解说

使用本指令时，请固定扫描时间，或者是将本指令放在时间中断插入副程式数中执行。

6. 应用指令解说

F76 ASC ASCII 码转换

F		ASC		ASCII 码转换								D·				
76																
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D·											*	*	*	*		

指令格式:

—— [ASC S D·]

S : 由计算机输入的 8 个字母数字

D· : 存放转换后的 ASCII 码元件起始地址

例:



指令执行结果: 字母 A~H 经 ASCII 码转换后传送到 D300~D303 中。

	高 8 位	低 8 位
D300	42(B)	41(A)
D301	44(D)	43(C)
D302	46(F)	45(E)
D303	48(H)	47(G)

- 该指令适于在外部显示器上选择显示出错等信息
- M8161 置于 ON 后执行该指令时, 向 D· 只传送低 8 位, 占有与传送字符相同数量的元件, 此时高 8 位为 0。

	高 8 位	低 8 位	
D300	00	41	A
D301	00	42	B
D302	00	43	C
D303	00	44	D
D304	00	45	E
D305	00	46	F
D306	00	47	G
D307	00	48	H

6. 应用指令解说

F77 PR ASC II 码打印

F					ASCII 码打印										S·	D·		
77			PR															
	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		
S·											*	*	*	*				
D·		*																

指令使用次数：1 次（但可以使用变址寄存器修饰）

指令格式：

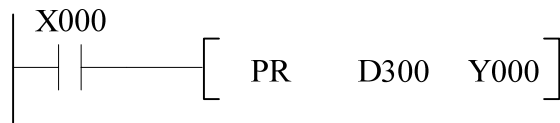
$$\text{——} \left[\text{PR} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：存放欲发送的 ASCII 码起始元件。

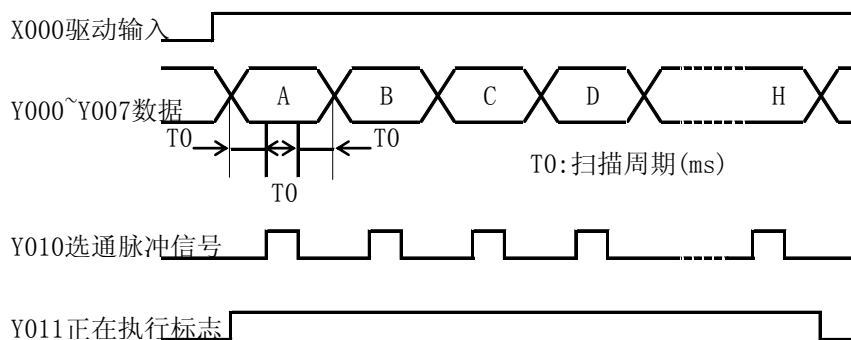
D·：输出 ASCII 码的外部输出点，占用连续 10 点。

本指令为将 ASCII 码数据输出到 Y 上的指令。

例：



- 如同前页 ASCII 数据若在 D300 ~D303 中，发送的顺序以 A 为起始，最后发送 H。
- 发送输出为 Y000（低位）~Y007（高位），其它还有选通脉冲信号 Y010，正在执行标志 Y011。



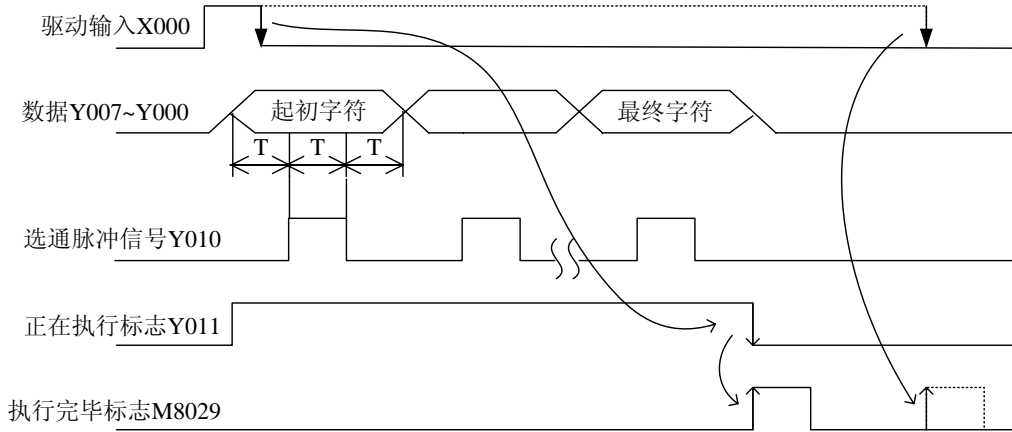
- 驱动输入 X000 在指令执行时被置 OFF 时传送立即停止，再次置 ON 时，从初始开始动作。
- 该指令与扫描周期（上图 T0）同时执行时，扫描周期短时，请用恒定扫描模式，过长时可以使用定时中断驱动。
- 可编程控制器须使用晶体管输出型。

6. 应用指令解说

《16 字节顺序输出》

FNC77 (PR) 为依次串联输出 8 位并行数据的指令。当特殊辅助继电器 M8027=OFF 时, 为 8 字节串联输出; 当 M8027=ON 时, 为 1~16 字节串联输出, 在传送数据时, 若出现 00H(NUL) 时, 即停止指令执行, 剩余数据则不输出。

《PR 指令的动作》 M8027=ON 时



T: 运算周期或中断时间

- 指令驱动 X000 上升沿时, 指令开始执行, 数据输出中驱动 X000 置 OFF 时, 不停止输出。
- 驱动 X000 即使连续处于 ON 状态, 一次循环输出结束后, 停止输出。但是, 标志 M8029 直到 X000 OFF 时才动作。

F81~F88 外部设备 SER

外部设备.SER

功能编号	助记符	名称	页码
80	RS	串行数据传送	1
81	PRUN	八进制位传送	8
82	ASCI	HEX 转为 ASCII	9
83	HEX	ASCII 转为 HEX	11
84	CCD	校验码	13
85	VRRD	电位器值读出	15
86	VRSC	电位器刻度	17
87	MBUS	MBUS 格式串行数据传送	18
88	PID	PID 运算	26

6. 应用指令解说

F80 RS 串行数据传送

F					串行数据传送										S	m	D	n	K
80			RS																
	位元件				字元件														
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S													*	*					
m					*	*							*	*					
D													*	*					
n					*	*							*	*					
K					0,1	0,1													

指令符号：

$$\text{——} \left[\text{RS} \quad \text{S} \quad \text{m} \quad \text{D} \quad \text{n} \quad \text{K} \right]$$

S: 为发送数据地址

m: 为发送数据长度 (0~255)

D: 为接收数据地址

n: 为接收数据长度 (0~255)

K: 串口选择, 常数 0~1

0: RS485 通讯接口, 1: RS485 或 RS232 扩充卡之通讯接口。

此命令乃是使用 RS-232、RS-485 扩充卡之通讯接口或是内建 RS485 通讯接口, 以达到串行式无协议数据的传送与接收。

- RS485 通讯接口数据的传送格式, 可以利用特殊缓存器 D8120 来设定。RS 命令在执行中, 即使改变了 D8120 的设定, 实际上并不接受。
- RS485 或 RS232 扩充卡之通讯接口数据的传送格式, 可以利用特殊缓存器 D8320 来设定。RS 命令在执行中, 即使改变了 D8320 的设定, 实际上并不接受。
- 在不作讯息传送的环境中将讯息传送点数设在「K0」。另外, 若不作讯息接收的环境中, 请将讯息接收点数设定「K0」。
- 虽然可以在一个程序中按所需使用大量的 RS、MBUS、DTLK、RMIO 等通信指令, 但要确保在同一时刻一个串口只有一个通信指令被驱动, 切换时应确保 OFF 时间等于或大于一个扫描周期。

通信规格

<通信格式「D8120」, 「D8320」>

通信格式 D8120、D8320 乃是除了来自 FNC80(RS)命令的无顺序通信之外, 于其他通讯命令或计算器连结时也可使用的特殊数据缓存器。

但在使用 FNC80(RS)命令时, 其他通讯命令或计算器连结的相关设定会成为无效, 请依照下列注意事项格式设定。

6. 应用指令解说

位号	名称	内容	
		0(位 OFF)	1(位 ON)
B0	数据长度	7 bit	8 bit
B1 B2	奇偶性	B2,B1 (0,0):无 (0,1):奇数(ODD) (1,0):偶数(EVEN)	
B3	停止位	1 bit	2 bit
B4 B5 B6 B7	通信速率 (bps)	B7,B6,B5,B4 (0,1,1,1):9,600 (1,0,0,0):19,200 (1,0,0,1):38,400 (1,0,1,0):57,600	B7,B6,B5,B4 (1,0,1,1):76,800 (1,1,0,0):128,000 (1,1,0,1):153,600 (1,1,1,0):307,200
B8*1	起始符	无	有, 初期值:STX(02H)
B9*1	终止符	无	有, 初期值:ETX(03H)
B10~B15*2	不可使用		

*1:起始符及结束符的内容可由使用者作变更。使用计算器连结时, 请务必归零使用。

*2:B10~B15 为其他通讯命令或计算器连结时的设定项目。使用 FNC80(RS)命令时, 请务必以「0」使用。

● 通讯格式的设定范例

数据长度	8 bit
Parity	无
停止位	2 bit
传送速度	19,200 bps
起始符	无
终止符	无
通信口选择	RS485 通信口

在上表的通信设定请依下列程序进行设定, 或是依周边机器的串行通信做设定。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D8120	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1
↓																
D8120	89H															



使用的特殊继电器和数据寄存器:

a) 使用 RS485 通讯接口时

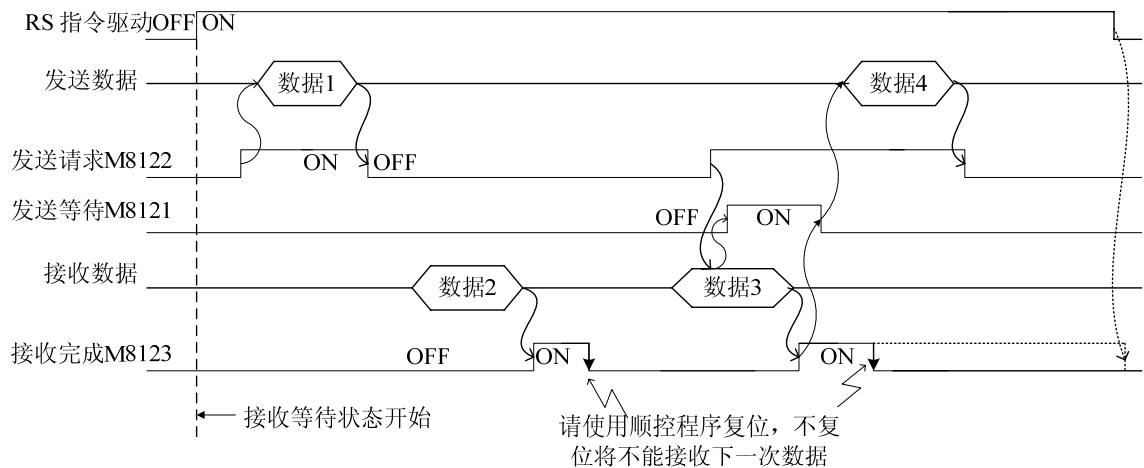
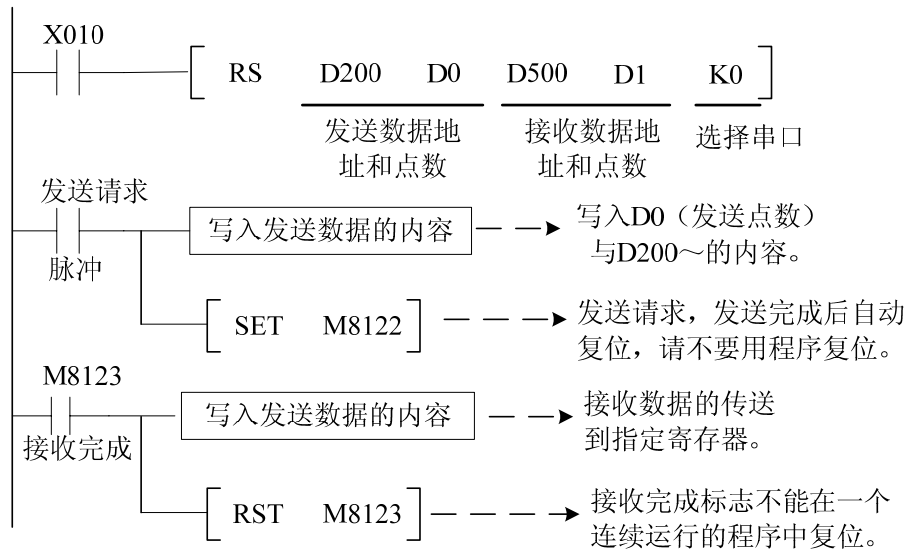
- 1) 发送等待 (M8121): 当在接收数据过程中有发送请求时置 1, 待接收完成后发送数据时自动清除。
- 2) 发送请求 (M8122): 当 M8122 被一个接收等待或接收完成状态下的脉冲指令设置时, 开始发送从 (S) 开始的 (m) 长度的数据, 当发送完成时, M8122 自动复位。
- 3) 接收完成 (M8123): 接收完成后 M8123 ON, 请先把接收数据传送到指定寄存器后再对 M8123 复位进入接收等待状态。
- 4) 超时判定 (M8129): 在接收数据中途规定时间内不再重新开始接收, 作为超时输出标志 M8129 变为 ON, 接收结束; 在 M8123 清除时 M8129 自动清除。
- 5) 通讯格式设定 (D8120): 参照前文设定 RS 指令通讯格式。
- 6) 发送数据剩余个数 (D8122)
- 7) 接收数据个数 (D8123)
- 8) 起始符 (D8124): 用户可设定起始符。
- 9) 结束符 (D8125): 用户可设定结束符。
- 10) 超时判定时间 (D8129): 设定超时判定用时间(5~255)*10ms。

b) 使用 RS485 或 RS232 扩充卡之通讯接口时 (具体定义参照上文)

- 1) 发送等待 (M8321)
- 2) 发送请求 (M8322)
- 3) 接收完成 (M8323)
- 4) 超时判定 (M8329)
- 5) 通讯格式设定 (D8320)
- 6) 发送资料剩余个数 (D8322)
- 7) 接收数据个数 (D8323)
- 8) 起始符 (D8324)
- 9) 结束符 (D8325)
- 10) 超时判定时间 (D8329)

发送接收的时序

RS 指令不仅规定了从 PLC 的发送数据首地址和点数，还规定了接收数据的存储首地址和最大可接收点数。采用 RS 指令发送、接收数据顺序如下所示（选择 RS485 通信接口）。



发送请求 M8122

- 输入条件 X010 ON 执行 RS 命令，控制器即进入等待收数据的状态。
- 在等待接收数据的状态或在接收数据结束的状态时，将 M8122 用脉波信号 SET ON，则执行将 D200 开始连续 D0 点的数据发送出去，在发送结束时，M8122 会自动 RESET 为 OFF。

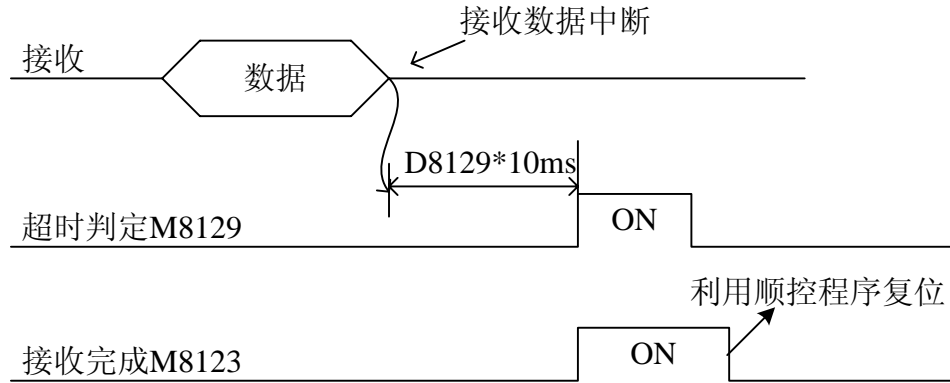
接收完成 M8123

- 在接收结束标志 M8123 ON 时，将所接收的数据，搬移至其它组件储存后，将 M8123 RESET 为 OFF。
- 若将 M8123 RESET 为 OFF 时，即再度进入等待接收的状态。输入条件 X010 ON 执行 RS 命令，控制器即进入等待接收数据的状态。
- 当设定(D1)=0 时，执行 RS 命令，则 M8123 不会动作，也不会进入等待接收的状态。若 D1 ≥ 1 后，将 M8123 ON 再 OFF 后，即可进入等待接收状态。

6. 应用指令解说

超时判定 M8129

- 接收数据途中中断时，从此时点开始至 D8129 设定的时间内若是未再重新开始接收数据，则会被视为逾时而 M8129 会 ON，变成接收结束状态。
M8129 会随 M8123 程序复位时自动复位。
使用这个功能，接收数据无结束符情况下也可接收完成。

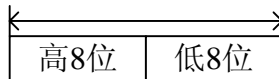
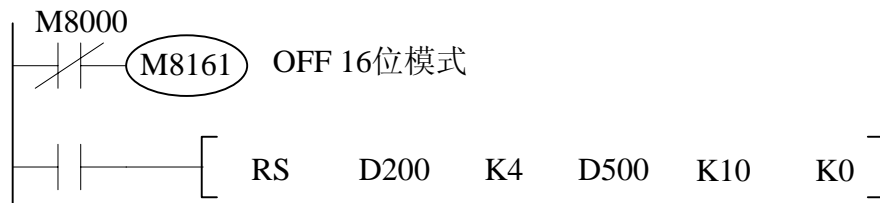


超时判定时间

- 设定上列超时判定的使用时间。
设定时间时是将设定值 X10ms，有效值为 5~255，而当 D8129 设置超出有效范围时则变成 50ms。
将逾时判定时间设为 50ms 为例。



<16 位数据的处理>当 M8161=OFF 时(M8161 为 ASCII,HEX,CCD 等指令共享)



将16位数据，分割成高低8位进行数据传送接收。

6. 应用指令解说

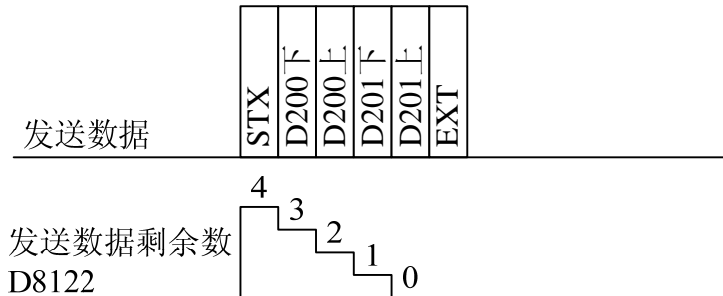
发送数据

STX	D200 下	D200 上	D201 下	D201 上	ETX
起始符	↑ S. 指定的起始地址号 M 指定的发送字节数				终止符

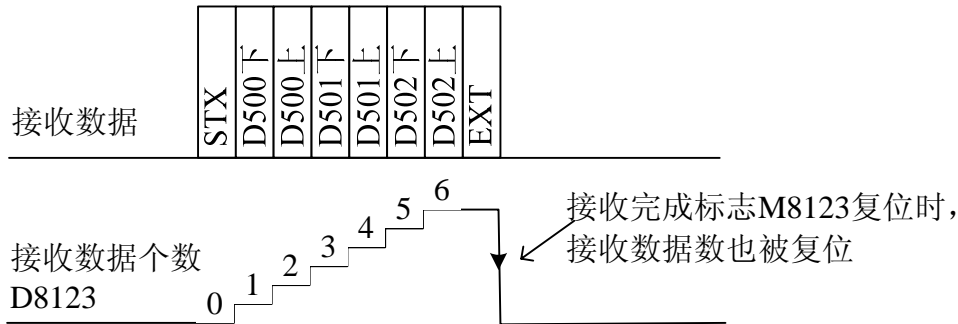
接收数据

STX	D500 下	D500 上	D501 下	D501 上	D502 下	D502 上	ETX
起始符	↑ D. 指定的起始地址号 不能超过 n 指定的接收上限点数, 由终止符 EXT 或 n 点接收表示接收完成。						终止符

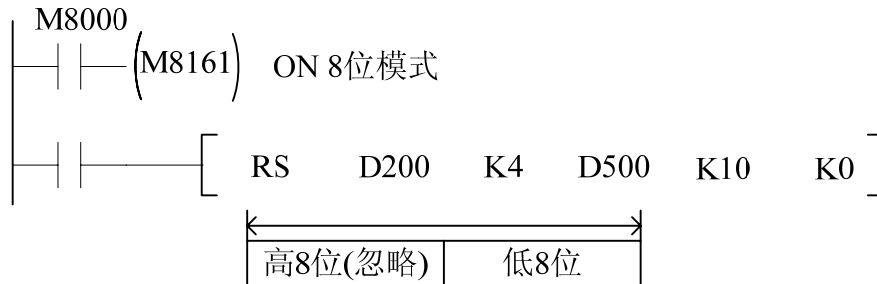
(1) 传送数据及传送数据剩余数



(2) 接收数据及接收数据数



<8 位数据的处理(扩张机能)>当 M8161=ON 时(M8161 为 ASCII,HEX,CCD 等命令共享)



16位数据忽略高8位, 仅低8位有效

6. 应用指令解说

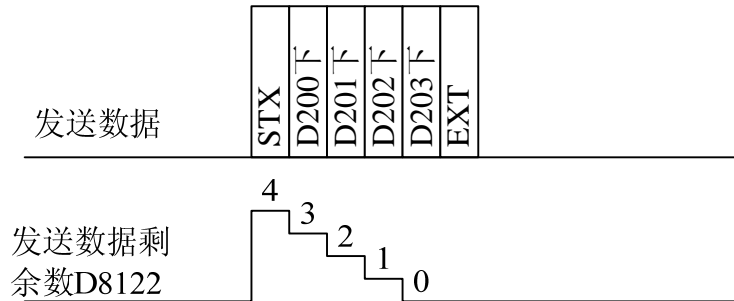
发送数据

STX	D200 下	D201 下	D202 下	D203 下	ETX
起始符	↑ S. 指定的起始地址号 M 指定发送字节数				终止符

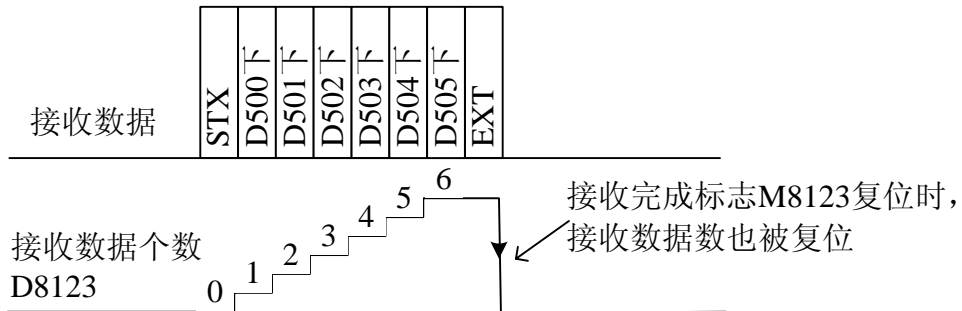
接收数据

STX	D500 下	D501 下	D502 下	D503 下	D504 下	D505 下	ETX
起始符	↑ S. 指定的起始地址号 不能超过 n 指定的接收上限点数, 由终止符 EXT 或 n 点接收表示接收完成。						终止符

(1) 传送数据及传送数据剩余数



(2) 接收数据及接收数据数



6. 应用指令解说

F81 PRUN 八进制位传送

F					八进制位传送								S·	D·			
81	D	PRUN	P														
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·							*		*								
D·								*	*								

指令格式：

$$\text{——} \left[\text{PRUN} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：传送来源装置，KnX,KnM 中的 n 为 1~8，指定元件号的最低位设为 0

D·：传送目的装置

本指令为以 8 进制数处理指定的元件号，并传送数据的指令。

例 1：



指令执行时 X10~X17→M0~M7，X20~X27→M10~M17，其中 M8，M9 值不变。

例 2：



指令执行时 M0~M7→X10~X17，M10~M17→Y20~Y27，其中 M8，M9 值不传送。

6. 应用指令解说

F82 ASCII HEX 转为 ASCII

F		ASCII				HEX 转为 ASCII						S ·	D ·	n			
82				P													
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S ·					*	*	*	*	*	*	*	*	*	*	*	*	
D ·								*	*	*	*	*	*	*			
n					*	*											

指令格式：

$$\text{——} \left[\text{ASCII} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

S ·：转换来源装置

D ·：转换目的装置

n：转换的位数，n=1~256

本指令为将 HEX 码转换为 ASCII 码的指令，存在 16 位和 8 位两种转换模式。

例：

$$\left| \begin{array}{l} \text{X000} \\ \text{——} \end{array} \right| \text{——} \left[\text{ASCII} \quad \text{D100} \quad \text{D200} \quad \text{K4} \right]$$

其中： D100=0ABCH

D101=1234H

D102=5678H.

ASCII: [0]=30H [1]=31H [2]=32H [3]=33H [4]=34H [5]=35H
 [6]=36H [7]=37H [8]=38H [9]=39H [A]=41H [B]=42H
 [C]=43H [D]=44H [E]=45H [F]=46H

《16 位转换模式》M8161=OFF 时（M8161 为 RS，ASCII，HEX，CCD 等指令共用）

- 将 S · 的 HEX 资料的各位数转换为 ASCII 码后，向 D · 的高 8 位和低 8 位分别传送，转换的位数以 n 来设定。
- D · 分为低 8 位和高 8 位，分别存储 ASCII 数据。

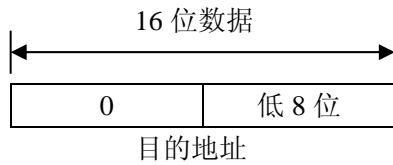
6. 应用指令解说

转换结果:

n D·	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200 低	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200 高		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201 低			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201 高				[C]	[B]	[A]	[0]	[4]	[3]
D202 低					[C]	[B]	[A]	[0]	[4]
D202 高						[C]	[B]	[A]	[0]
D203 低	不	变	化				[C]	[B]	[A]
D203 高								[C]	[B]
D204 低									[C]

《8位转换模式》M8161=ON时（M8161为RS，ASCI，HEX，CCD等指令共用）

- 将S·的HEX资料的各位数转换为ASCII码后，向D·的低8位传送，转换的位数以n来设定。
- D·的高8位为0。



转换结果:

n D·	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D201		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D202			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D203				[C]	[B]	[A]	[0]	[4]	[3]
D204					[C]	[B]	[A]	[0]	[4]
D205						[C]	[B]	[A]	[0]
D206	不	变	化				[C]	[B]	[A]
D207								[C]	[B]
D208									[C]

使用打印等输出BCD数据时，需在本指令前调用BIN→BCD转换指令。

6. 应用指令解说

F83 HEX ASCII 转为 HEX

F					ASCII 转为 HEX							S ·	D ·	n		
83		HEX		P												
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·					*	*	*	*	*	*	*	*	*	*	*	*
D ·								*	*	*	*	*	*	*	*	*
n					*	*										

指令格式:

$$\text{——} \left[\text{HEX} \quad \text{S} \cdot \quad \text{D} \cdot \quad \text{n} \right]$$

S · : 转换来源装置

D · : 转换目的装置

n: 转换的位数, n=1~256

本指令为将 ASCII 码转换为 HEX 码的指令, 存在 16 位和 8 位两种转换模式。

例:

$$\left| \begin{array}{l} \text{X000} \\ \text{——} \\ \text{——} \end{array} \right. \left[\text{HEX} \quad \text{D200} \quad \text{D100} \quad \text{K4} \right]$$

《16 位转换模式》M8161=OFF 时 (M8161 为 RS, ASCI, HEX, CCD 等指令共用)

- 将 S · 的高低各 8 位的 ASCII 字符数据转换为 HEX 数据后, 每 4 位向 D · 传送。转换的位数以 n 来设定。
- 在 HEX 指令中, 存入 S · 的数据不是 ASCII 码时, 则运算错误, 不能 HEX 转换。尤其是 M8161 为 OFF 时, S · 的高 8 位也需要存储 ASCII 码, 请注意。

转换结果:

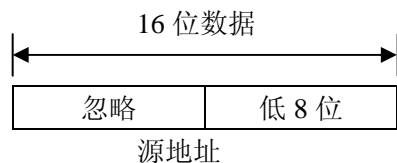
S ·	ASCII 码	HEX 转换
D200 低	30H	0
D200 高	41H	A
D201 低	42H	B
D201 高	43H	C
D202 低	31H	1
D202 高	32H	2
D203 低	33H	3
D203 高	34H	4
D204 低	35H	5

n	D ·		
	D102	D101	D100
1	不变化		··· 0H
2	· 为 0		·· 0AH
3			· 0ABH
4			0ABCH
5		··· 0H	ABC1H
6		·· 0AH	BC12H
7		· 0ABH	C123H
8		0ABCH	1234H
9	··· 0H	ABC1H	2345H

6. 应用指令解说

《8 位转换模式》M8161=ON 时 (M8161 为 RS, ASCII, HEX, CCD 等指令共用)

- 将 S · 的低 8 位的 ASCII 字符数据转换为 HEX 数据后, 每 4 位向 D · 传送。转换的位数以 n 来设定。



转换结果:

S ·	ASCII 码	HEX 转换
D200	30H	0
D201	41H	A
D202	42H	B
D203	43H	C
D204	31H	1
D205	32H	2
D206	33H	3
D207	34H	4
D208	35H	5

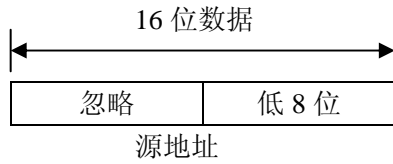
n \ D ·	D102	D101	D100
1	不变化 · 为 0		··· 0H
2			·· 0AH
3			· 0ABH
4			0ABCH
5		··· 0H	ABC1H
6		·· 0AH	BC12H
7		· 0ABH	C123H
8		0ABCH	1234H
9	··· 0H	ABC1H	2345H

- 输入数据为 BCD 码时, 在本指令执行后, 需要进行 BCD→BIN 转换。

6. 应用指令解说

《8 位转换模式》M8161=ON 时 (M8161 为 RS, ASCII, HEX, CCD 等指令共用)

- 将 S · 指定的元件为起始的 n 点数据(仅低 8 位)的总和与水平校验数据存储于 D · 于 D · +1 元件中。
- 可以用于通信数据的校验。



上例程序转换情况如下:

S ·	数据内容
D100	K100=01100100
D101	K111=01101111
D102	K100=01100100
D103	K98=01100010
D104	K123=01111011
D105	K66=01000010
D106	K100=01100100
D107	K95=01011111
D108	K210=11010010
D109	K88=01011000
合计	K1091
水平校验	10000101

6. 应用指令解说

F85 VRRD 电位器值读出

F		VRRD				电位器值读出										S·	D·		
85				P															
		位元件				字元件													
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		
S·						*	*												
D·									*	*	*	*	*	*	*				

指令格式:

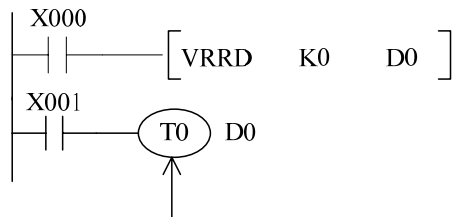
—— [VRRD S· D·]

S·: 电位器编号

D·: 存放电位器刻度的装置

※ 1: BSP01 SR 机种不支持该指令

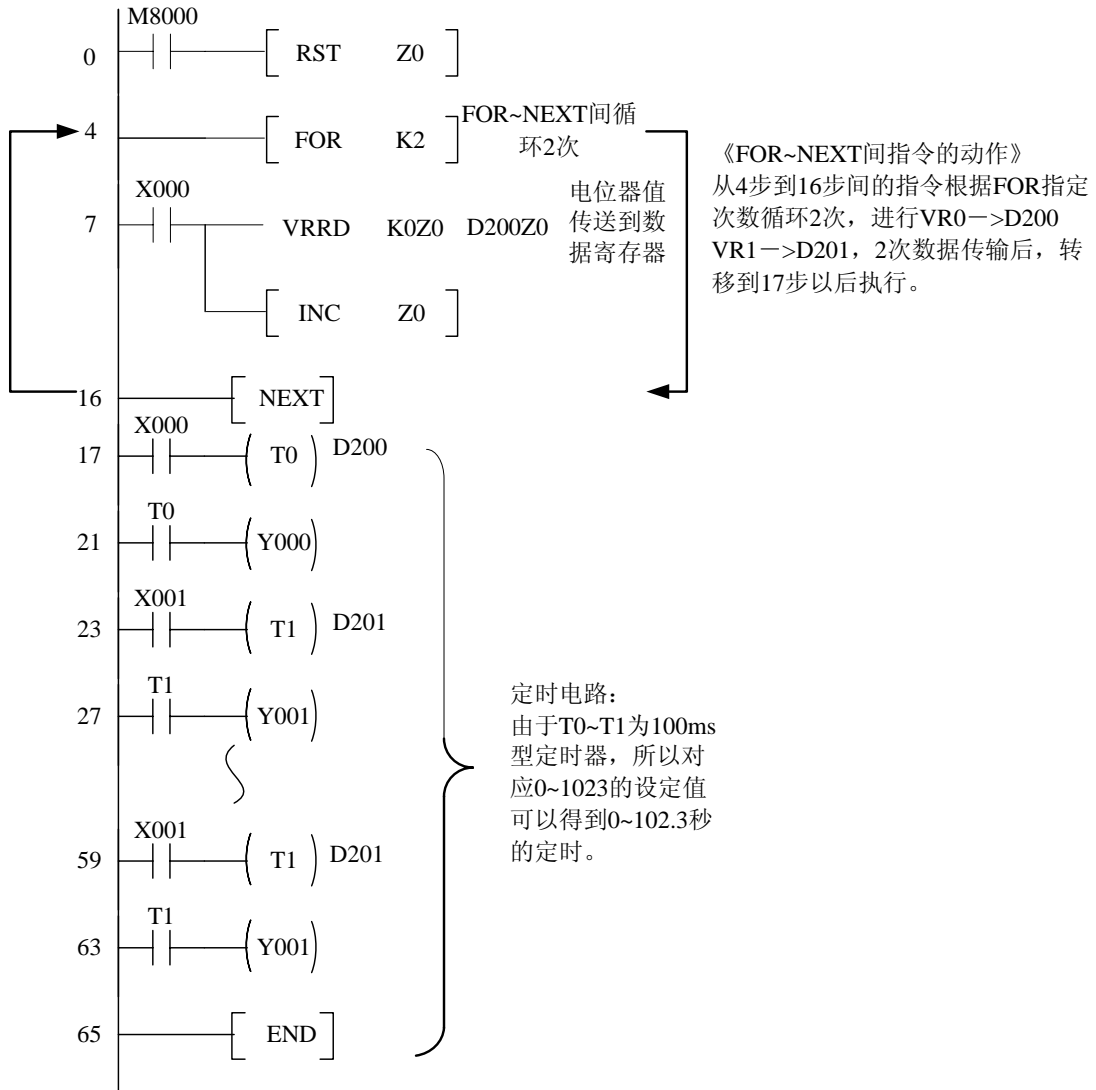
例:



作为模拟定时器的用途例

- 将电位器 No.0 的模拟值转换为 10 位的 BIN 数值(0~1023), 存放在 D0 中, 作为定时器的设定值使用。
- 当定时器的设定值需要 1024 以上的数值时, 请使用 FNC22(MUL)命令, 将读取值做乘积后, 以间接指定方式来做定时器的设定。
- 对应旋钮 VR0~VR1, VRRD 命令的指定值为 K0~K1。在下图中的程序, 以索引缓存器 (Z0=0~1)来修饰, 则 K0Z0=K0~K1。如下图:

6. 应用指令解说



6. 应用指令解说

F86 VRSC 电位器刻度

F		VRSC		电位器刻度	S·	D·										
86			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·					*	*										
D·								*	*	*	*	*	*	*	*	*

指令格式：

$$\text{——} \left[\text{VRSC} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·: 电位器编号

D·: 存放电位器刻度的位置

※ 1: BSP01 SR 机种不支持该指令

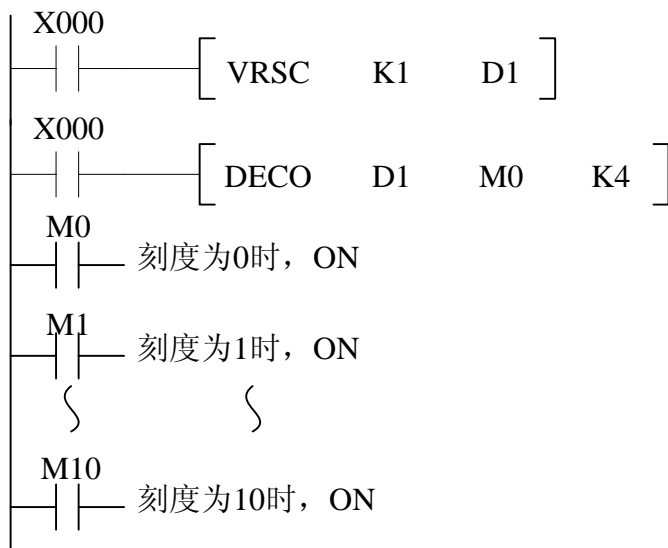
VRSC 指令用来读取PLC主机2点，编号为No.0、No.1。电位器刻度值（刻度值为0~10），并将刻度值，存放于D·中

例：



- 将电位器 No.1 的刻度 0~10 以 BIN 值存入 D1 中。
- 旋钮在旋转刻度时通过四舍五入化成 0~10 的整数值。

《旋转开关的使用》



- 根据电位器刻度 0~10，辅助继电器 M0~M10 中的 1 点置 ON。
- 通过 F41 DECO 指令，需占用辅助继电器 M0~M15。

F87 MBUS 运算

F		MBUS				MBUS 通信						S	m	D	n	K	
87																	
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S													*	*			
m					*	*							*	*			
D													*	*			
n					*	*							*	*			
K					0,1	0,1											

指令符号:

$$\text{——} \left[\text{MBUS} \quad \text{S} \quad \text{m} \quad \text{D} \quad \text{n} \quad \text{K} \right]$$

S: 为发送数据地址

m: 为发送数据长度 (0~255)

D: 为接收数据地址

n: 为接收数据长度 (0~255)

K: 串口选择, 常数 0~1

0: RS485 通讯接口, 1: RS485 或 RS232 扩充卡之通讯接口。

本指令为 RS485 或 RS232 扩充卡之通讯接口 (所有型号之机种皆可选购), RS485 通讯接口 (仅 H 型号之机种内建), 以达到串行式 Modbus 协议数据的传送与接收。以上两个接口可同时实现 MBUS 指令通信功能, 也可单独实现该功能。

MBUS 指令可作为主站通讯;

发送数据寄存器保存的命令格式为 HEX 命令代码, 包括地址、功能码和通讯数据。MBUS 指令根据设定的模式如为 RTU 模式添加 CRC 校验码 (2bytes) 后发送; 如为 ASCII 模式则添加起始符 (3AH)、LRC 校验码 (2bytes) 和结束符 (0DH+0AH), 并将命令数据转变为 ASCII 码格式发送至发送 BUFF。

接收数据寄存器中的保存的命令格式为地址、功能码和通讯数据, 不保存起始符、结束符和校验码。

- RS485 通讯接口数据的传送格式, 可以利用特殊缓存器 D8120 来设定。MBUS 命令在执行中, 即使改变了 D8120 的设定, 实际上并不接受。
- RS485 或 RS232 扩充卡之通讯接口数据的传送格式, 可以利用特殊缓存器 D8320 来设定。MBUS 命令在执行中, 即使改变了 D8320 的设定, 实际上并不接受。
- 在不作讯息传送的环境中将讯息传送点数设在「K0」。
- 虽然可以在一个程序中按所需使用大量的 RS、MBUS、DTLK、RMIO 等通信指令, 但要确保在同一时刻一个串口只有一个通信指令被驱动, 切换时应确保 OFF 时间等于或大于一个扫描周期。



6. 应用指令解说

通信规格

<通信格式「D8120」, 「D8320」>

通信格式 D8120、D8320 乃是除了来自 FNC87(MBUS)命令的 Modbus 协议通信之外, 于其他通讯命令或计算器连结时也可使用的特殊数据缓存器。

但在使用 FNC87(MBUS)命令时, 其他通讯命令或计算器连结的相关设定会成为无效, 请依照下列注意事项格式设定。

位号	名称	内容	
		0(位 OFF)	1(位 ON)
B0	数据长度	7 bit	8 bit
B1 B2	奇偶性	B2,B1 (0,0):无 (0,1):奇数(ODD) (1,0):偶数(EVEN)	
B3	停止位	1 bit	2 bit
B4 B5 B6 B7	通信速率 (bps)	B7,B6,B5,B4 (0,1,1,1):9,600 (1,0,0,0):19,200 (1,0,0,1):38,400 (1,0,1,0):57,600	B7,B6,B5,B4 (1,0,1,1):76,800 (1,1,0,0):128,000 (1,1,0,1):153,600 (1,1,1,0):307,200
B8~B12 *1	不可使用		
B13	Modbus 模式	(0): RTU 模式	(1): ASCII 模式
B14~B15*1	不可使用		

*1:B8~B12, B14, B15 为其他通讯命令或计算器连结时的设定项目。使用 FNC87(MBUS)命令时, 请务必以「0」使用。

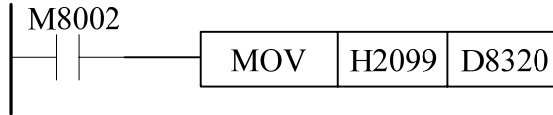
● 通讯格式的设定范例

数据长度	8 bit
Parity	无
停止位	2 bit
传送速度	38,400 bps
模式	ASCII 模式
串口选择	RS485 扩充通信卡

在上表的通信设定请依下列程序进行设定, 或是依周边机器的串行通信做设定。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D8320	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	1
↓																
D8320	2099H															

6. 应用指令解说



使用的特殊继电器和数据寄存器:

a) 使用 RS485 通讯接口时

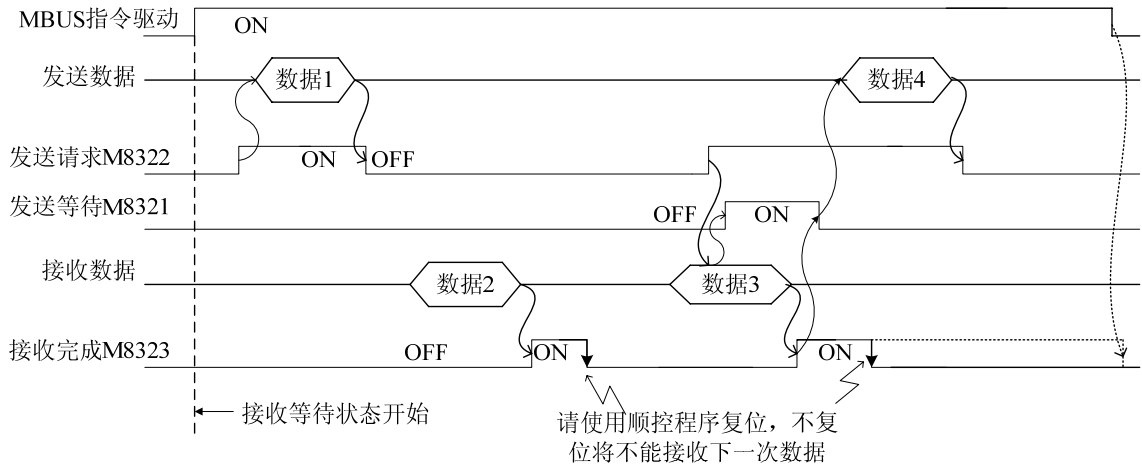
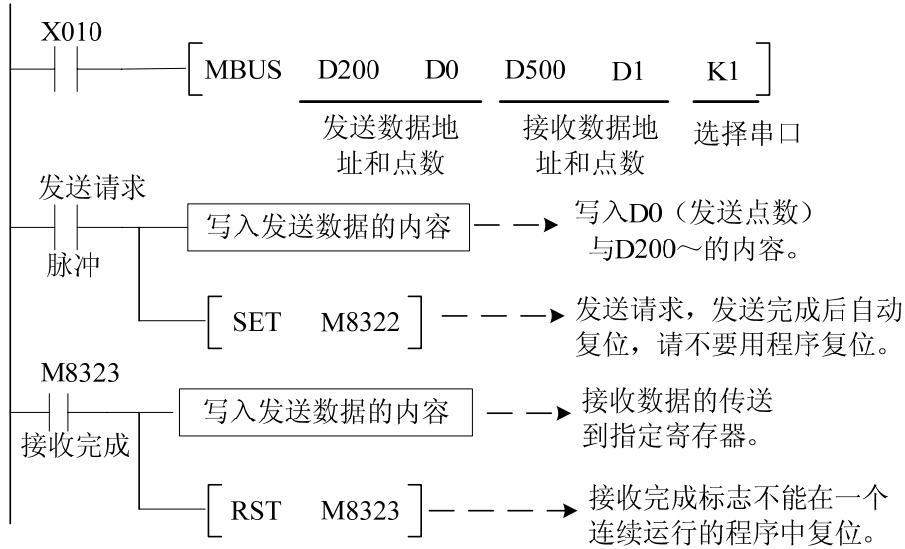
- 1) 发送等待 (M8121): 当在接收数据过程中有发送请求时置 1, 待接收完成后发送数据时自动清除。
- 2) 发送请求 (M8122): 当 M8122 被一个接收等待或接收完成状态下的脉冲指令设置时, 开始发送从 (S) 开始的 (m) 长度的数据, 当发送完成时, M8122 自动复位。
- 3) 接收完成 (M8123): 接收完成后 M8123 ON, 请先把接收数据传送到指定寄存器后再对 M8123 复位进入接收等待状态。
- 4) 出错指示 (M8124): 接收出错 (RTU 模式时 CRC 出错, ASCII 时 LRC 或结束符出错)。
- 5) 超时判定 (M8129): 在接收数据中途规定时间内不再重新开始接收, 作为超时输出标志 M8129 变为 ON, 接收结束; 在 M8123 清除时 M8129 自动清除。
- 6) 通讯格式设定 (D8120): 参照前文设定 MBUS 指令通讯格式。
- 7) 发送数据剩余个数 (D8122)
- 8) 接收数据个数 (D8123)
- 9) 超时判定时间 (D8129): 设定超时判定用时间(5~255)*10ms。

b) 使用 RS485 或 RS232 扩充卡之通讯接口时 (具体定义参照上文)

- 1) 发送等待 (M8321)
- 2) 发送请求 (M8322)
- 3) 接收完成 (M8323)
- 4) 超时判定 (M8329)
- 5) 出错指示 (M8324)
- 6) 通讯格式设定 (D8320)
- 7) 发送资料剩余个数 (D8322)
- 8) 接收数据个数 (D8323)
- 9) 超时判定时间 (D8329)

发送接收的时序

MBUS 指令不仅规定了从 PLC 的发送数据首地址和点数, 还规定了接收数据的存储首地址和最大可接收点数。采用 MBUS 指令发送、接收数据顺序如下所示 (例选择 RS485 扩充通信卡)。



发送请求 M8322

- 输入条件 X010 ON 执行 MBUS 命令, 控制器即进入等待收数据的状态。
- 在等待接收数据的状态或在接收数据结束的状态时, 将 M8322 用脉冲信号 SET ON, 则执行将 D200 开始连续 D0 点的数据发送出去, 在发送结束时, M8322 会自动 RESET 为 OFF。

接收完成 M8323

- 在接收结束标志 M8323 ON 时, 将所接收的数据, 搬移至其它组件储存后, 将 M8323 RESET 为 OFF。
- 若将 M8323 RESET 为 OFF 时, 即再度进入等待接收的状态。输入条件 X010 ON 执行 MBUS 命令, 控制器即进入等待接收数据的状态。
- 当设定(D1)=0 时, 执行 MBUS 命令, 则 M8323 不会动作, 也不会进入等待接收的状态。若 D1 ≥ 1 后, 将 M8323 ON 再 OFF 后, 即可进入等待接收状态。

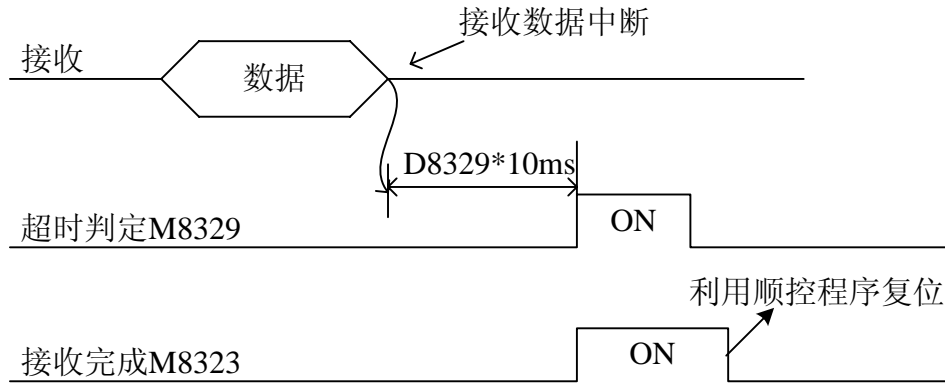
超时判定 M8329

- 接收数据途中中断时, 从此时点开始至 D8329 设定的时间内若是未再重新开始接收数据, 则会被视为逾时而 M8329 会 ON, 变成接收结束状态。

6. 应用指令解说

M8329 会随 M8323 程序复位时自动复位。

使用这个功能，接收数据(ASCII 模式)无结束符情况下也可接收完成。



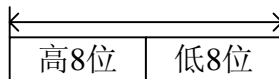
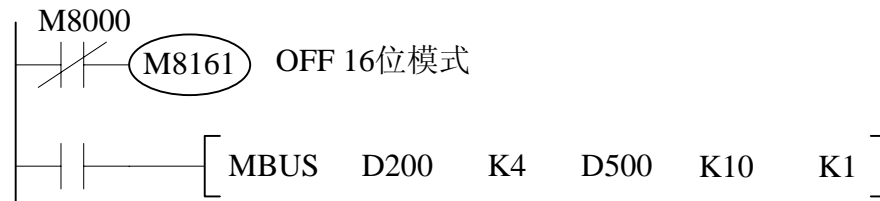
超时判定时间

- 设定上列超时判定的使用时间。

设定时间是将设定值 X10ms, 有效值为 5~255, 而当 D8329 设置超出有效范围时则变成 50ms。将逾时判定时间设为 50ms 为例。



<16 位数据的处理>当 M8161=OFF 时(M8161 为 RS,ASCI,HEX,CCD 等指令共享)



将16位数据，分割成高低8位进行数据传送接收。

发送数据 (ASCII 模式和 RTU 模式不同)

STX	D200 下	D200 上	D201 下	D201 上	校验码	ETX	
起始符 (3A)					LRC (ASCII)	结束符 (0D0A)	
RTU 模式 (无)	↑					CRC (RTU)	RTU 模式 (无)
	S. 指定的起始地址号						
	M 指定的发送字节数						

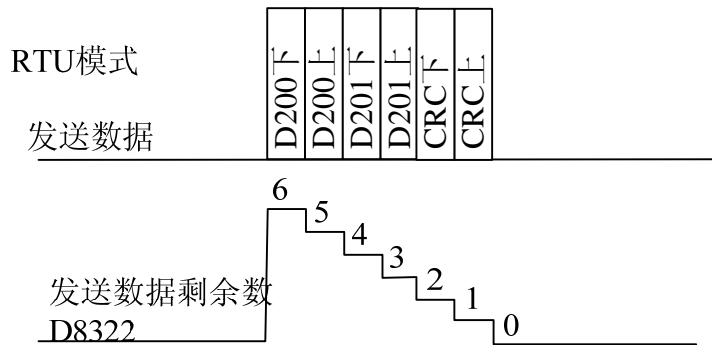
6. 应用指令解说

接收数据

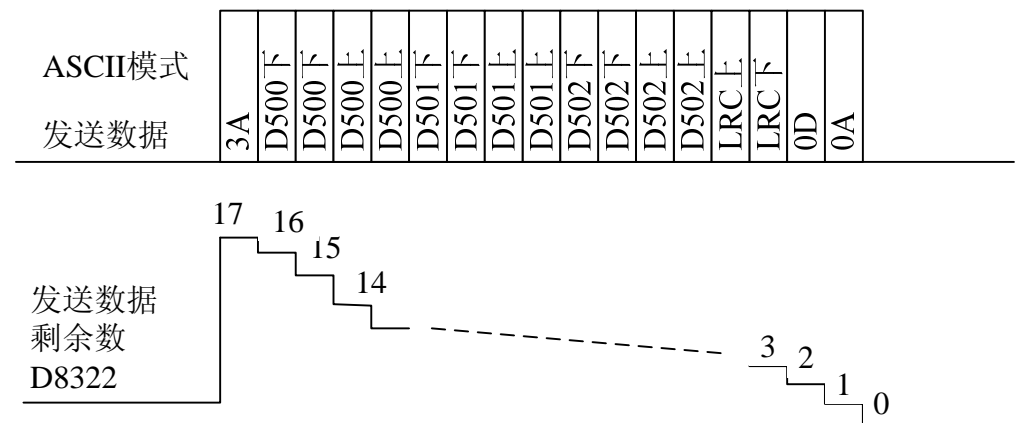
STX	D500 下	D500 上	D501 下	D501 上	D502 下	D502 上	校验码	ETX
起始符 (3A)							LRC (ASCII)	结束符 (0D0A)
RTU (无)	↑ D. 指定的起始地址号 不能超过 n 指定的接收上限点数，由终止符 EXT 或 n 点接收表示接收完成。						CRC (RTU)	

(3) 传送数据及传送数据剩余数

RTU 模式下



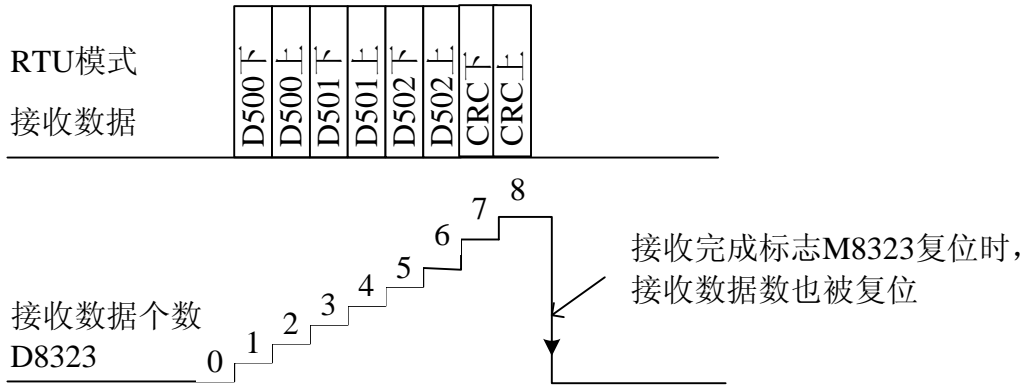
ASCII 模式下



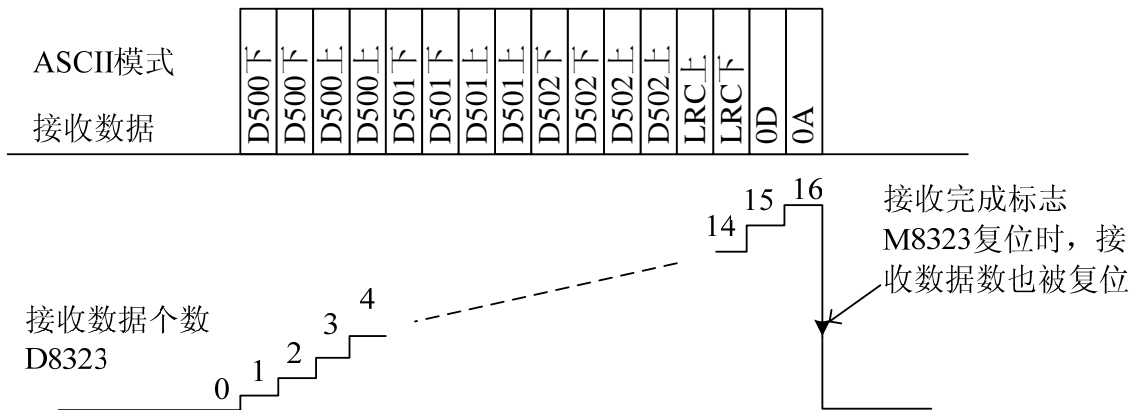
(4) 接收数据及接收数据数

RTU 模式下

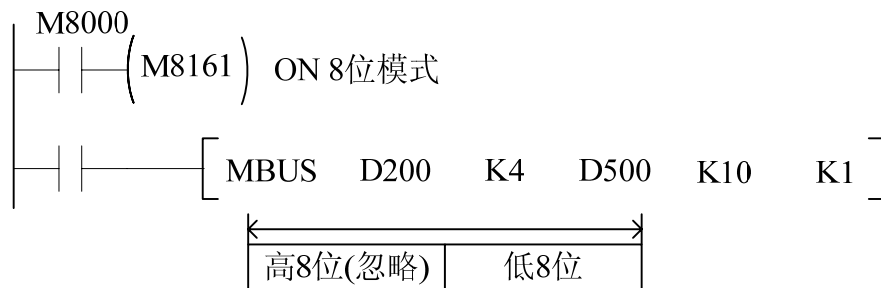
6. 应用指令解说



ASCII 模式下



<8 位数据的处理(扩张机能)>当 M8161=ON 时(M8161 为 RS,ASCI,HEX,CCD 等命令共享)



16位数据忽略高8位, 仅低8位有效

发送数据 (ASCII 模式和 RTU 模式不同)

STX	D200 下	D201 下	D202 下	D203 下	校验码	ETX
起始符 (3A)					LRC (ASCII)	结束符 (0D0A)

6. 应用指令解说

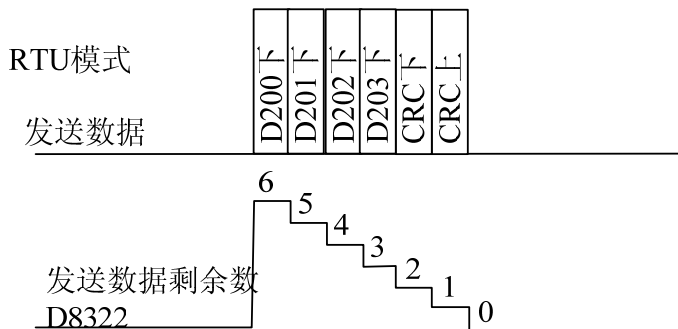
RTU 模式 (无)	↑ S. 指定的起始地址号 M 指定的发送字节数	CRC (RTU)	RTU 模式 (无)
------------	--------------------------------	-----------	------------

接收数据

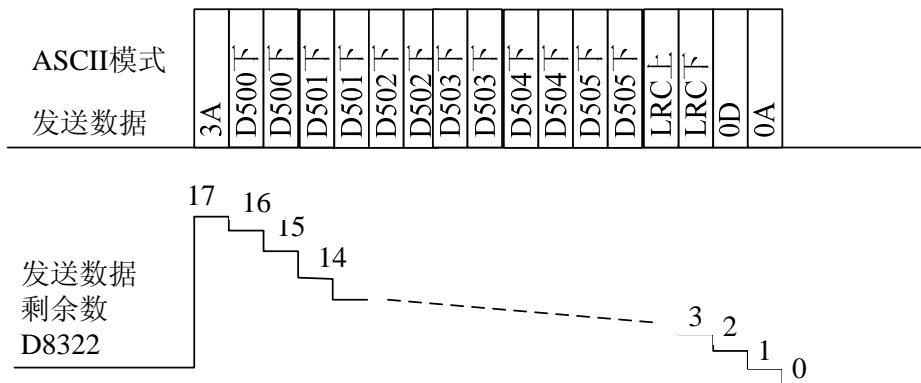
STX	D500 下	D501 下	D502 下	D503 下	D504 下	D505 下	校验码	ETX
起始符 (3A)							LRC (ASCII)	结束符 (0D0A)
RTU (无)	↑ D. 指定的起始地址号 不能超过 n 指定的接收上限点数, 由终止符 EXT 或 n 点接收表示接收完成。						CRC (RTU)	

(1) 传送数据及传送数据剩余数

RTU 模式下



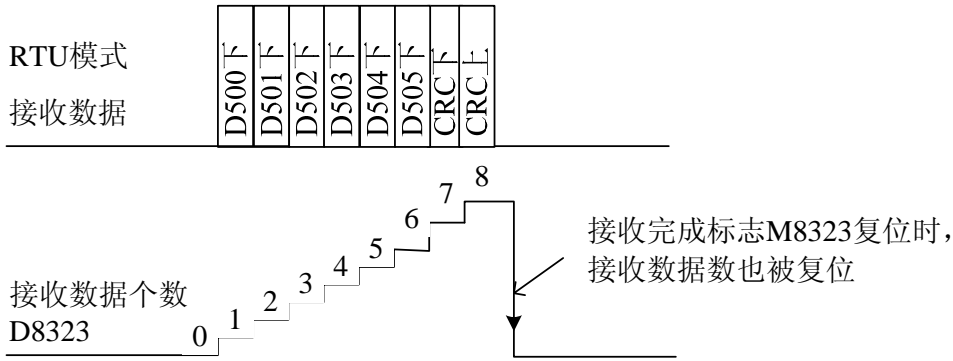
ASCII 模式下



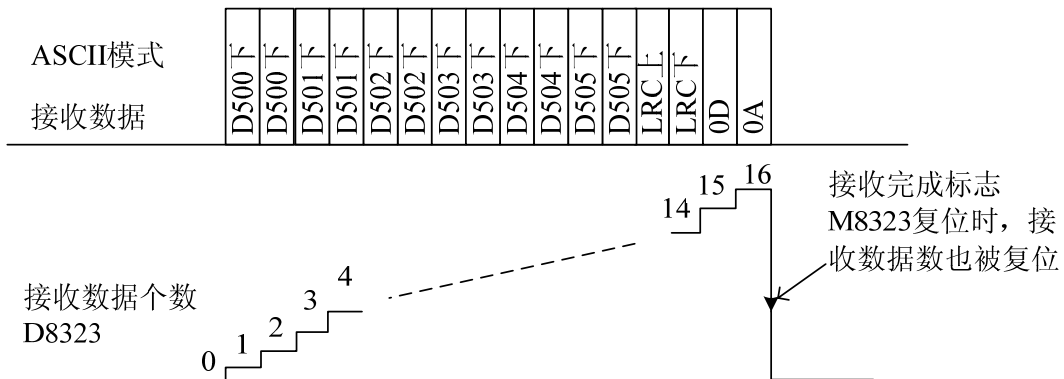
(2) 接收数据及接收数据数

RTU 模式下

6. 应用指令解说



ASCII 模式下



F88 PID 运算

F	PID		PID 运算													S1	S2	S3	D	
88	位元件				字元件															
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z				
S1													*							
S2													*							
S3													*							
D													*							

指令符号

— [PID S1 S2 S3 D]

S1: 目标值 (SV)

S2: 现在值 (PV)

S3: 参数, S3~S3+6 设定控制参数。

D: 输出值 (MV)

※ 1: 对于 BSP01 AR/T 机种 S3 仅使用 D0~D7975

※ 2: 对于 BSP01 SR 机种 S3 仅使用 D0~D487

S1: 设定目标值 SV

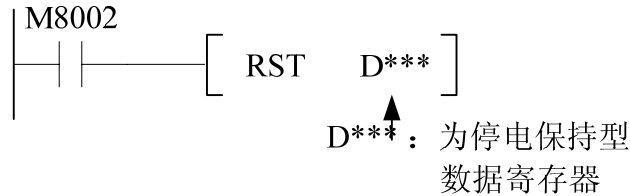
S2: 设定测定值 PV

} 执行程序时运算结果 MV 被存入 D 中。

6. 应用指令解说

S3~S3+6:设定控制参数

该指令是 PID 运算控制的专用指令，于取样时间到达后的该次扫描时间才执行 PID 运算动作。参数全部设定完成后，开始执行 PID 指令（即在开始执行 PID 运算前，必须先以 MOV 等命令写入 PID 控制用的参数设定值），其结果暂存于 D 中，D 的内容请指定无停电保持功能的暂存器（如果要指定具停电保持功能的寄存器，请于程序开始加入 RUN 的时候该寄存器的内容归 0）。



程序会自动占用自 S3 起始的 25 个数据寄存器，（但下页显示的控制参数的 ACT 设定，BIT1,BIT2 和 BIT5 均为“0”时，只占自 S3 开始的 20 个数据寄存器）。

参数设定

控制用参数的设定值在 PID 运算前必须通过 MOV 等指令写入。另外，指定停电保持区域的数据寄存器时，编程控制器的电源 OFF 后设定值仍保持。因此不需进行再次写入。

S3	取样时间(Ts)	1~32767[ms](但是不可设定比扫描时间短)
S3+1	动作方向(ACT)	BIT0 0:正动作 1:逆动作 BIT1 0:无输入变动量警报 1:有输入变动量警报 BIT2 0:无输出变动量警报 1:有输出变动量警报 BIT3 不可使用 BIT4 0:自动调谐不动作 1:执行自动调谐 BIT5 0:输出值上下限未设定 1:输出值上下限设定有效 BIT6~BIT15 不可使用 BIT5 和 BIT2 请勿同时 ON。
S3+2	输入滤波常数(a)	0~99[%] 0 时没有输入滤波
S3+3	比例增益(Kp)	1~32767[%]
S3+4	积分时间(TI)	1~32767[x100ms] 0 时为无穷大(无积分动作)
S3+5	微分增益(KD)	0~100[%] 0 时没有积分增益
S3+6	微分时间(TD)	1~32767[x10ms] 0 为无微分动作
S3+7~S3+19	执行 PID 运算时，内部处理用。	
S3+20	输入变化量(增)警报设定值	0~32767(S3+1<ACT>的 bit1=1 时有效)
S3+21	输入变化量(减)警报设定值	0~32767(S3+1<ACT>的 bit1=1 时有效)
S3+22	输出变化量(增)警报设定值	0~32767(S3+1<ACT>的 bit2=1,bit5=0 时有效)

6. 应用指令解说

	或者输出上限设定值	-32768~32767(S3+1<ACT>的 bit2=1,bit5=0 时有效)
S3+23	输入变化量 (减)警报设定值	0~32767(S3+1<ACT>的 bit2=1,bit5=0 时有效)
	或者输出下限设定值	-32768~32767(S3+1<ACT>的 bit2=1,bit5=0 时有效)
S3+24	警报输出	BIT0 输入变化量(增) BIT1 输入变化量(减) BIT2 输出变化量(增) BIT3 输出变化量(减) (S3+1<ACT>的 bit1=1 或者 bit=1 时有效)

但是, S3+20~S3+24 在 S3+1<ACT>的 bit1=1, bit2=2 或者 bit5=1 时被占用。

PID 命令虽可同时执行数次(回路次数无限制),但在运算时所使用的 S3 或是 D 等编号请不要重复。

PID 命令在定时器中断、子程序、步进梯形图、跳转指令中也可使用。

取样时间 T_s 的误差最大在-(1 个扫描周期+1ms) ~ +(1 个扫描周期)之间。当 T_s 的值很小时,此变动会造成问题,此时请以常数扫描模式执行,或是在定时器中断内作程式编辑。

若取样时间 $T_s \leq$ PLC 控制器的 1 个运算周期时会发生下列 PID 的运算异常(K6740),并以 $T_s =$ 运算周期作 PID 运算。此时建议在定时器中断(I6 口口~I8 口口)内使用 PID 命令。

输入滤波器定数有助于缓和测定值的变化。

增加微分增益有助于缓和输出值的急剧变化。

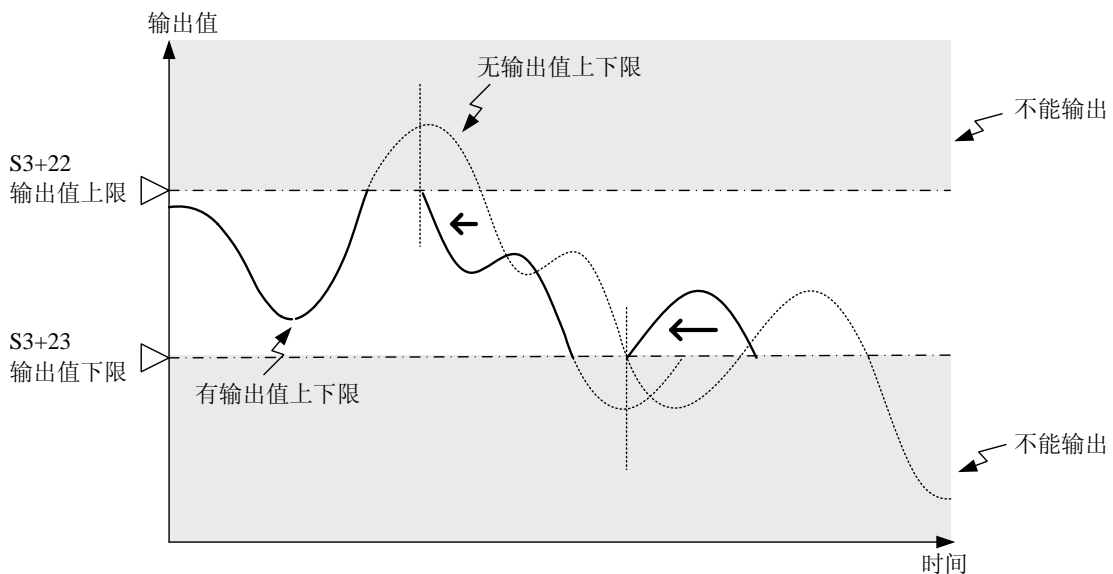
动作方向(S3+1(ACT))

动作方向[bit 0]

系统的动作方向以正动作,逆动作指定。

输出值上下限设定 [bit5]

输出值上下限设定有效(S3+1(ACT)的 bit5=1)时,输出值则如下图所示。使用此设定有助于抑制 PID 控制积分项的增大。使用此功能时,请务必将 S3+1(ACT)的 bit2 OFF。



6. 应用指令解说

警报设定(输入变化量, 输出变化量)[bit1,bit2]

使 S3+1(ACT)的 bit1、bit2ON, 则操作者可任意检测输入变化量及输出变化量。检测需依照 S3+20~S3+23 的值进行。若是超过了所设定的输入变化量, 警报标记的 S3+24 的各个位元会于 PID 命令执行后转为 ON。(参照下图)

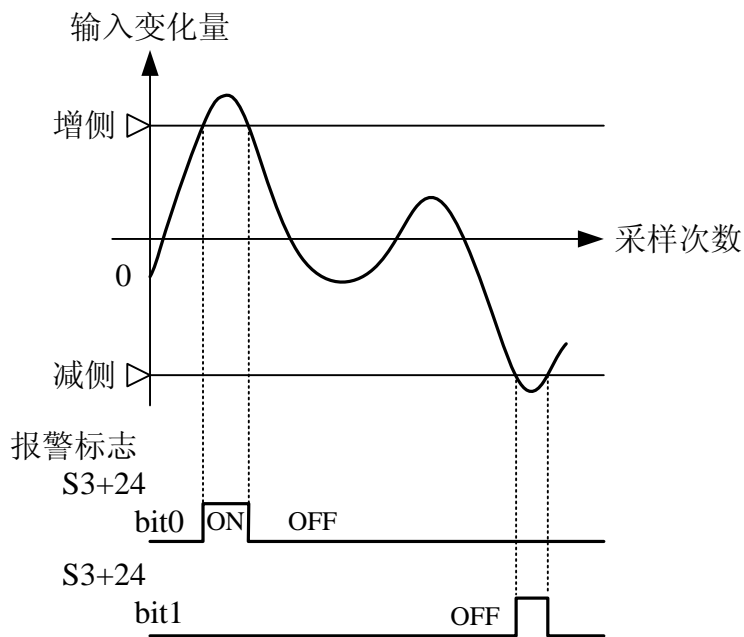
但是, 将 S3+21、S3+23 作为警报值使用时, 设定的值会被作为负数值使用。还有, 使用输出变化量的警报功能时, 请务必将 S3+1(ACT)的 bit5 作 OFF。

a)变化量为

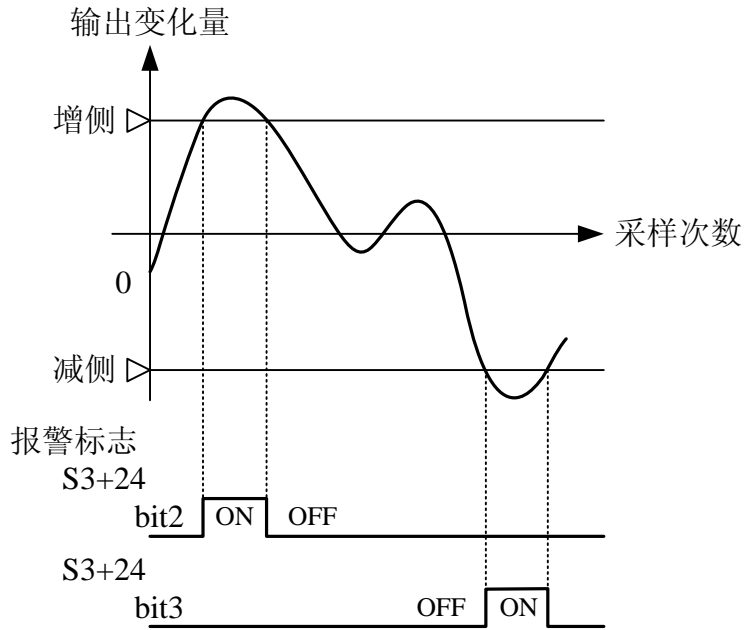
(前回的值)-(本回的值)=变化量

b)警报标记的动作(S3+24)

i)输入变化量(bit1=1)



ii)输出变化量(bit2=1)



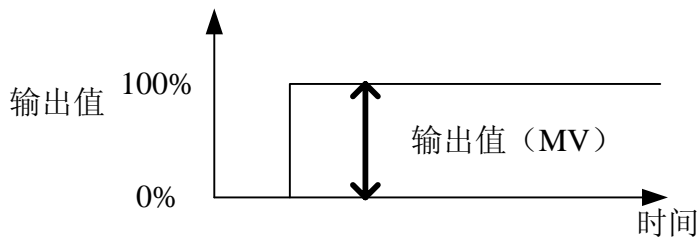
PID3 个求法

为了执行 PID 控制得到良好的控制结果，必须符合适合于控制对象的各常数（参数）的最佳值。这里必须求得 PID 的 3 个常数（比例增益 K_p ，积分时间 T_I ，微分时间 T_D ）的最佳值。作为其求解方法有阶跃反应法，这里就该阶跃反应法加以说明。

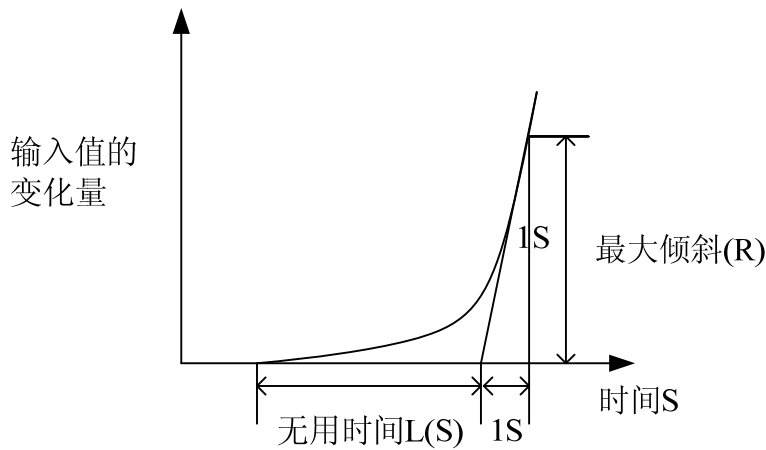
阶跃反应法是对控制系统施加 0~100% 的阶跃状输出，由输入变化判断动作特征（最大倾斜 R ，无用时间 L ），来求得 PID 的 3 个常数的方法。

1 阶跃状输出也可通过 0~75% 或 0~50% 求得。

动作特性



6. 应用指令解说



<动作特性和 3 个常数>

	比例增益 K_p (%)	积分时间 $T1(\times 100ms)$	微分时间 $TD(\times 100ms)$
仅有比例控制 P	$\frac{1}{RL} \times \text{输入值}$ (MV)	—	—
PI 控制	$\frac{0.9}{RL} \times \text{输入值}$ (MV)	33L	—
PID 控制	$\frac{1.2}{RL} \times \text{输入值}$ (MV)	20L	50L

控制参数的设定值或 PID 运算中的数据发生错误时，则运算错误 M8067 变为 ON，根据其错误内容 D8067 中存入下述数据：

代码	错误内容	处理状态	处理方法
K6705	应用指令的操作数在对象软元件之外	PID 命令 运算停止	请确认控制数据的内容
K6706	应用指令的操作数的元件地址号范围和 数据超出		
K6730	采样时间 TS 在对象范围以外($TS < 0$)		
K6732	输入滤波常数在对象范围以外		
K6733	比例增益在对象范围以外		
K6734	积分时间在对象范围以外		
K6735	微分增益在对象范围以外		
K6736	微分时间在对象范围以外	PID 命令	
K6740	采样时间 \leq 运算周期		

6. 应用指令解说

K6742	测定值变化量溢出	运算继续	
K6743	偏差值溢出		
K6744	积分计算值溢出		
K6745	微分增益溢出导致微分值溢出		
K6746	微分计算值溢出		
K6747	PID 运算结果溢出		
K6750	自动调谐结果不良	自动调谐结束	自动调谐开始时的测定值和目标值的差为 150 以下或自动调谐开始时的测定值和目标值的差的 1/3 以上则结束确认测定值、目标值，请再次进行自动调谐。
K6751	自动调谐动作方向不一致	自动调谐继续	从自动调谐开始时的测定值预测的动作方向和自动调谐用输出时实际动作方向不一致。请使目标值、自动调谐用输出值、测定值的关系后，再次进行自动调谐。
K6752	自动调谐动作不良	自动调谐结束	自动调谐中测定值因上下变化不能正确动作，请使采样时间远远大于输出的变化周期，增大输入滤波常数。设定变更后，请再次进行自动调谐。

要点：

必须在 PID 运算执行前，将正确的测定值读入 PID 测定值 PV 中，特别对模拟量输入模块的输入值进行 PID 运算时，需注意其转换时间。

PID 指令的基本运算式

本指令根据速度形，测定值微分形运算式，进行 PID 运算。PID 控制根据 S3 中指定的动作方向的内容，执行正动作或逆动作的运算式。

另外，在运算中必要的各值 S3 以后指定的被使用的参数的内容。

PID 基本运算式

动作方向	PID 运算方式
正动作	$\Delta MV = Kp\{(EV_n - EV_{n-1}) + \frac{T_s}{T_I} EV_n + D_n\}$ $EV_n = PV_{nf} - SV$ $D_n = \frac{T_D}{T_s + \alpha_D T_D} (-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{\alpha_D T_D}{T_s + \alpha_D T_D} \cdot D_{n-1}$ $MV_n = \sum \Delta MV$

6. 应用指令解说

逆动作	$\Delta MV = Kp\left\{(EV_n - EV_{n-1}) + \frac{T_s}{T_I} EV_n + D_n\right\}$ $EV_n = SV - PV_{nf}$ $D_n = \frac{T_D}{T_s + \alpha_D \cdot T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{\alpha_D \cdot T_D}{T_s + \alpha_D \cdot T_D} \cdot D_{n-1}$ $MV_n = \sum \Delta MV \Delta$
-----	--

记号说明

EVn:本次采样时的偏差

EVn-1:一个周期的偏差

SV:目标值

PVnf:本次采样时的测定值（滤波后）

PVnf-1:一个周期前的测定值（滤波后）

PVnf-2:二个周期前的测定值（滤波后）

ΔMV :输出的变化值

MVn:本次的操作量

Dn:本次的微分量

Dn-1:一个周期前的微分项

Kp:比例增益

Ts:采样周期

T1:积分常数

TD:微分常数

α_D :微分增益

PVnf 是根据读入的测定值由下列运算式求得的值。

[后的测定值 PVnf]=PVn+L (PV_{nf-1}-PVn)

PVn: 本次采样时的测定值

L:滤波系数

PV_{nf-1}: 一个周期前的测定值（滤波后）

F110~F119, F120-F129, F130~F137 浮点指令

浮点指令

功能编号	助记符	名称	页码
F110	ECMP	二进制浮点数比较	265
F111	EZCP	二进制浮点数区域比较	266
F112	EMOV	二进制浮点数数据传送	267
F118	EBCD	二进制浮点数→十进制浮点数	268
F119	EBIN	十进制浮点数→二进制	269
F120	EADD	二进制浮点数加法运算	270
F121	ESUB	二进制浮点数减法运算	271
F122	EMUL	二进制浮点数乘法运算	272
F123	EDIV	二进制浮点数除法运算	273
F124	EXP	二进制浮点数指数运算	274
F125	LOGE	二进制浮点数自然对数运算	276
F126	LOG10	二进制浮点数常用对数运算	277
F127	ESQR	二进制浮点数开方运算	278
F128	ENEG	二进制浮点数符号翻转	279
F129	INT	二进制浮点→BIN 整数变换	280
F130	SIN	浮点 SIN 运算	281
F131	COS	浮点 COS 运算	282
F132	TAN	浮点 TAN 运算	283
F133	ASIN	ASIN 浮点 ASIN 运算	284
F134	ACOS	浮点 ACOS 运算	285
F135	ATAN	浮点 ATAN 运算	286
F136	RAD	角度转换为弧度	287
F137	DEG	弧度转换为角度	288

6. 应用指令解说

F111 EZCP 二进制浮点数区域比较

F				二进制浮点数区域比较							S1 · S2 · S · D ·						
111	D	EZCP		P													
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1 ·					*	*							*	*			*
S2 ·					*	*							*	*			*
S ·					*	*							*	*			*
D ·		*	*	*													

指令格式:

$$\text{---} \left[\text{EZCP} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S1·: 区域比较的二进制浮点数的下限值

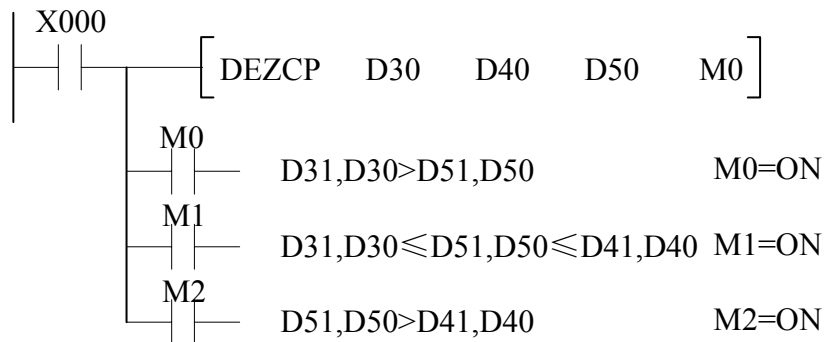
S2·: 区域比较的二进制浮点数的上限值

S·: 二进制浮点数的比较值

D·: 比较结果, 占连续三点

注: 设置时, 注意 $S1 \leq S2$, 当 $S1 > S2$ 时, 把 $S2$ 的值当作与 $S1$ 相同值进行比较

例:



X000OFF 时, 即使 EZCP 指令不执行, M0~M2 保持 X000 为 OFF 前的状态。

将 S, S+1 的内容与两个指定的二进制浮点数 S1、S2 之间的范围作比较。将结果 (ON 或 OFF) 存放在以 D 为起始的三个位元件中。

常数 K, H 被指定为源数据时, 自动转换成二进制浮点数后再处理。

6. 应用指令解说

F112 EMOV 二进制浮点数数据传送

F		EMOV		二进制浮点数数据传送								S·	D·				
112	D		P														
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S·													*	*			*
D·													*	*			

指令格式：

$$\text{——} \left[\text{EMOV } S \cdot \quad D \cdot \right]$$

S·：传送源的二进制浮点数数据，或是保存数据的软元件编号

D·：保存二进制浮点数数据的软元件编号

例：

$$\begin{array}{|c} \text{X000} \\ \hline \text{——} \left[\text{DEMOV } D10 \quad D0 \right] \end{array}$$

(D11, D10) → (D1, D0)

将传送源(S+1, S·)的内容（二进制浮点数数据）传送到(D+1, D·)中。

此外，还可以在 S·中直接指定实数（E）。

$$\begin{array}{|c} \text{X000} \\ \hline \text{——} \left[\text{DEMOV } E-1.23 \quad D0 \right] \end{array}$$

-1.23 → (D1, D0)

6. 应用指令解说

F118 EBCD 二进制浮点数→十进制浮点数

F 118	D	EBCD	P	二进制浮点数→十进制浮点数								S·	D·				
/	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·														*	*		
D·														*	*		

指令格式：

$$\text{——} \left[\text{EBCD} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：资料来源（二进制浮点数）

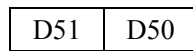
D·：变换结果（十进制浮点数）

例：



把源数据指定的元件内的二进制浮点数转换为十进制浮点数，存入目的地址中。

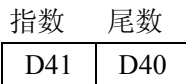
二进制浮点数



尾数部分 23 位，指数部分 8 位，符号 1 位



十进制浮点数



尾数部分 指数部分

$$D40 \times 10^{D41}$$

$$10 \text{ 进制浮点值} = \text{「尾数 D40」} \times 10^{\text{[指数D41]}}$$

尾数 D40=(1,000~9,999)或 0

指数 D41= -41 ~ +35

小数运算在可编程控制器内部全部以二进制浮点数值为基础执行，但是由于二进制浮点数值为不易判别的值，所以可该指令将其变换为十进制浮点数。外部设备很容易监控。

6. 应用指令解说

F119 EBIN 十进制浮点数→二进制浮点数

F		EBIN		十进制浮点数→二进制浮点数	S·	D·										
119	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·													*	*		
D·													*	*		

指令格式：

$$\text{——} \left[\text{EBIN} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：资料来源（十进制浮点数）

D·：变换结果（二进制浮点数）

例：



把源数据指定的元件 D50 内的十进制浮点数转换为二进制浮点数，存于 D41，D40 当中
十进制浮点数

指数	尾数
D51	D50

尾数 指数

$D50 \times 10^{D51}$

10 进制浮点值=「尾数 D50」 $\times 10^{[\text{指数} D51]}$

尾数 D50=(1,000~9,999)或 0

指数 D51= -41 ~ +35

二进制浮点数

D41	D40
-----	-----

尾数部分 23 位，指数部分 8 位，符号 1 位

6. 应用指令解说

F120 EADD 二进制浮点数加法运算

F 120	D	EADD	P	二进制浮点数加法运算								S1·	S2·	D·			
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1·					*	*							*	*			*
S2·					*	*							*	*			*
D·													*	*			

指令格式：

$$\text{---} \left[\text{EADD} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \right]$$

S1·: 被加数

S2·: 加数

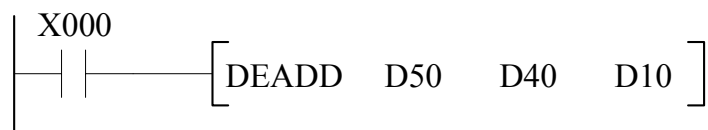
D·: 和

将 S1·所指定的暂存器上的数与 S2·所指定的暂存器上的数相加，和存放在 D·所指定的暂存器中，加法运算以二进制浮点数形式进行。

S1·或 S2·可指定的内容如为常数，指令会将常数转换成二进制浮点数后进行加法运算。

S1·和 S2·可指定相同的暂存器号，一般情况下，都是使用脉冲执行型指令 DEADD P。

例：



$$(D51, D50) + (D41, D40) \rightarrow (D11, D10)$$

当 X000ON 时，将二进制浮点数 (D51, D50) 与二进制浮点数 (D41, D40) 相加，结果存放在 (D11, D10) 中。

6. 应用指令解说

F121 ESUB 二进制浮点数减法运算

F		ESUB				二进制浮点数减法运算							S1·	S2·	D·		
121	D			P													
/	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1·					*	*							*	*			*
S2·					*	*							*	*			*
D·													*	*			

指令格式：

$$\text{——} \left[\text{ESUB } S1 \cdot \quad S2 \cdot \quad D \cdot \right]$$

S1·: 被减数

S2·: 减数

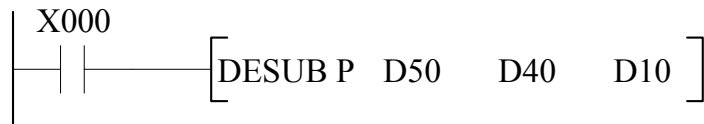
D·: 差

将 S1·所指定的暂存器上的数减去 S2·所指定的暂存器上的数，差存放在 D·所指定的暂存器中，减法运算以二进制浮点数形式进行。

S1·或 S2·可指定的内容如为常数，指令会将常数转换成二进制浮点数后进行减法运算。

S1·和 S2·可指定相同的暂存器号，一般情况下，都是使用脉冲执行型指令 DESUB P。

例：



$$(D51, D50) - (D41, D40) \rightarrow (D11, D10)$$

当 X000ON 时，将二进制浮点数 (D51, D50) 减去二进制浮点数 (D41, D40)，差存放在 (D11, D10) 中。

6. 应用指令解说

F122 EMUL 二进制浮点数乘法运算

F 122	D	EMUL	P	二进制浮点数乘法运算								S1·	S2·	D·			
/	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1·					*	*							*	*			*
S2·					*	*							*	*			*
D·													*	*			

指令格式：

—— [EMUL S1· S2· D·]

S1·: 被乘数

S2·: 乘数

D·: 积

将 S1·所指定的暂存器上的数乘以 S2·所指定的暂存器上的数，积存放在 D·所指定的暂存器中，乘法运算以二进制浮点数形式进行。

S1·或 S2·可指定的内容如为常数，指令会将常数转换成二进制浮点数后进行乘法运算。

S1·和 S2·可指定相同的暂存器号，一般情况下，都是使用脉冲执行型指令 DEMUL P。

例：

(D51, D50) × (D41, D40) → (D11, D10)

当 X000ON 时，将二进制浮点数 (D51, D50) 乘以二进制浮点数 (D41, D40)，积存放在 (D11, D10) 中。

6. 应用指令解说

F123 EDIV 二进制浮点数除法运算

F 123	D	EDIV	P	二进制浮点数除法运算								S1·	S2·	D·			
/	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S1·					*	*							*	*		*	
S2·					*	*							*	*		*	
D·													*	*			

指令格式：

$$\text{——} \left[\text{EDIV} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \right]$$

S1·: 被除数

S2·: 除数（内容不能为0，否则，被认为运算错误，指令不执行）

D·: 商

将 S1·所指定的暂存器上的数除以 S2·所指定的暂存器上的数，商存放在 D·所指定的暂存器中，除法运算以二进制浮点数形式进行。

S1·或 S2·可指定的内容如为常数，指令会将常数转换成二进制浮点数后进行除法运算。

S1·和 S2·可指定相同的暂存器号，一般情况下，都是使用脉冲执行型指令 DEDIV P。

例：

$$\left| \begin{array}{l} \text{X000} \\ \text{——} \end{array} \right| \text{——} \left[\text{DEDIV P} \quad \text{D50} \quad \text{D40} \quad \text{D10} \right]$$

$(\text{D51}, \text{D50}) \div (\text{D41}, \text{D40}) \rightarrow (\text{D11}, \text{D10})$

当 X000ON 时，将二进制浮点数 (D51, D50) 除以二进制浮点数 (D41, D40)，商存放在 (D11, D10) 中。

6. 应用指令解说

F124 EXP 二进制浮点数指数运算

F		EXP		二进制浮点数指数运算								S·	D·				
124	D		P														
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S·													*	*			*
D·													*	*			

指令格式：

$$\text{——} \left[\text{EXP} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·: 指数

D·: 幂

以 e (2.71828) 为底数, 以(S·+1, S·)为指数做运算, 将结果保存到(D·+1, D·)中。
此外, 也可在 S·中直接指定实数。

$$e^{(S \cdot + 1, S \cdot)} \rightarrow (D \cdot + 1, D \cdot)$$

出错：

当运算结果不在以下范围时, 会发生运算出错, 出错标志位 (M8067) 为 ON, 在 D8067 中保存错误代码 (K6706)。

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

例：



$$e^{(D11, D10)} \rightarrow (D21, D20)$$

指数 幂

要点：

- 1) 由于 $\text{Log}_e e = 88.7$ 。所以当(D11, D10)设定了 89 以上的数值时会发生运算出错。
- 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。

$$10^x = e^{\frac{x}{0.4342945}}$$

6. 应用指令解说

F125 LOGE 二进制浮点数自然对数运算

F		LOGE				二进制浮点数自然对数运算							S·	D·							
125	D				P																
	位元件					字元件															
	X	Y	M	S		K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E			
S·														*	*			*			
D·														*	*						

指令格式：

$$\text{——} \left[\text{LOGE} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S： 幂

D： 指数

执行(S+1, S·)的自然对数（以 e (2.71828) 为底数的对数）运算，并将结果保存到(D+1, D·)中。此外，也可在 S·中直接指定实数。

在(S+1, S·)中指定的值，只可以设定正数。（负数不能运算。）

$$\text{Log}_e(\text{S} \cdot + 1, \text{S} \cdot) \rightarrow (\text{D} \cdot + 1, \text{D} \cdot)$$

出错：

当 S·中指定的值为负数或“0”时，会发生运算出错，出错标志位 (M8067) 为 ON，在 D8067 中保存错误代码 (K6706)。

例：



$$\text{Log}_e(\text{D}11, \text{D}10) \rightarrow (\text{D}21, \text{D}20)$$

6. 应用指令解说

F126 LOG10 二进制浮点数常用对数运算

F 126	D	LOG10	P	二进制浮点数常用对数运算								S·						D·
		位元件				字元件												
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S·														*	*			*
D·														*	*			

指令格式：

$$\text{——} \left[\text{LOG10} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S： 幂

D： 指数

执行(S+1, S·)的自然对数（以 10 为底数的对数）运算，并将结果保存到(D+1, D·)中。此外，也可在 S·中直接指定实数。

在(S+1, S·)中指定的值，只可以设定正数。（负数不能运算。）

$$\text{Log}_{10}(\text{S} \cdot +1, \text{S} \cdot) \rightarrow (\text{D} \cdot +1, \text{D} \cdot)$$

出错：

当 S·中指定的值为负数或“0”时，会发生运算出错，出错标志位（M8067）为 ON，在 D8067 中保存错误代码（K6706）。

例：

$$\begin{array}{|c} \text{X000} \\ \hline \text{——} \left[\text{DLOG10} \quad \text{D10} \quad \text{D20} \right] \end{array}$$

$$\text{Log}_{10}(\text{D11}, \text{D10}) \rightarrow (\text{D21}, \text{D20})$$

6. 应用指令解说

F127 ESQR 二进制浮点数开方运算

F 127	D	ESQR				二进制浮点数开方运算						S·	D·				
/	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S·					*	*							*	*			*
D·													*	*			

指令格式：

$$\text{——} \left[\text{ESQR} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：欲开方的源元件（只有正数有效，如为负数，则视为运算错误，M8067 ON）

D·：平方根

将 S·所指定的暂存器上的数开方，平方根存放在 D·所指定的暂存器中，开方运算以二进制浮点数形式进行。

S·指定的内容如为常数，指令会将常数转换成二进制浮点数后进行开方运算。

例：



$$\sqrt{(D51, D50)} \rightarrow (D41, D40)$$

当 X000ON 时，将二进制浮点数 (D51, D50) 开方，平方根存放在 (D41, D40) 中。

6. 应用指令解说

F128 ENEG 二进制浮点数符号翻转

F					二进制浮点数符号翻转								D·					
128	D	ENEG			P													
		位元件				字元件												
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
D·														*	*			

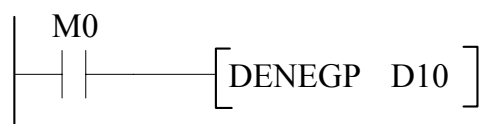
指令格式:

$$\text{——} \left[\text{ENEG} \quad \text{D} \cdot \right]$$

D·: 保存要执行符号翻转的二进制浮点数数据的软元件的起始编号

将(D·+1, D·)的二进制浮点数数据的符号翻转, 保存在(D·+1, D·)中。

例:



(D11, D10) → (D11, D10)

1.2345 → -1.2345

6. 应用指令解说

F129 INT 二进制浮点→BIN 整数变换

F 129	D	INT	P	二进制浮点→BIN 整数变换								S·	D·					
S·	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E	
D·					*	*							*	*				

指令格式：

$$\text{---} \left[\text{INT} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：欲变换之来源

D·：变换结果

S·所指定的暂存器内容以 2 进制浮点数形态变换成 BIN 整数暂存于 D·所指定的暂存器当中，BIN 整数小数被舍去。

本指令的动作与 F 49FLT 指令相反。

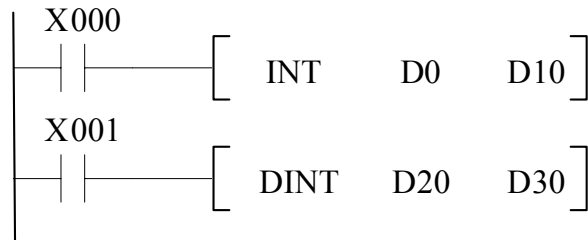
变换结果若为 0 时，零标志 M8012=ON。变换结果有小数点被舍去时，借位标志 M8021=ON

变换结果若超过下列范围时，进位标志 M8022=ON。

16 位元指令：-32,768~32,767

32 位元指令：-2,147,483,648~2,147,483,647

例：



当 X000=ON 时，将 2 进制小数点（D1，D0）变换成 BIN 整数将结果存放在 D10 中，BIN 整数小数点小数舍去。

当 X001=ON 时，将 2 进制小数点（D21，D20）变换成 BIN 整数将结果存放在（D31，D30）中，BIN 整数小数点小数舍去。

6. 应用指令解说

F130 SIN 浮点 SIN 运算

F		SIN			浮点 SIN 运算							S·	D·				
130	D			P													
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S·													*	*			*
D·													*	*			

指令符号:

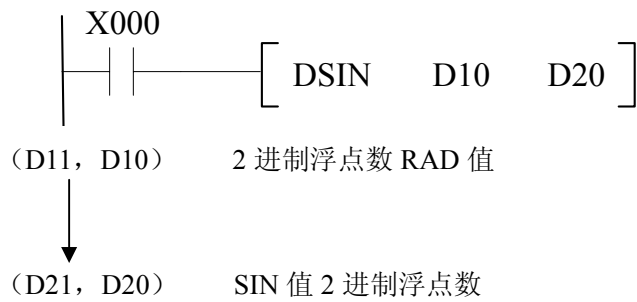
— [SIN S· D·]

S·: 指定的 RAD 值

D·: SIN 运算结果

S·所指定的 RAD 值等于 (角度 × π / 180) 求取 SIN 值后存放在 D·所指定的暂存器中。

例



6. 应用指令解说

F131 COS 浮点 COS 运算

F		COS			浮点 COS 运算							S·	D·				
131	D			P													
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S·													*	*			*
D·													*	*			

指令格式：

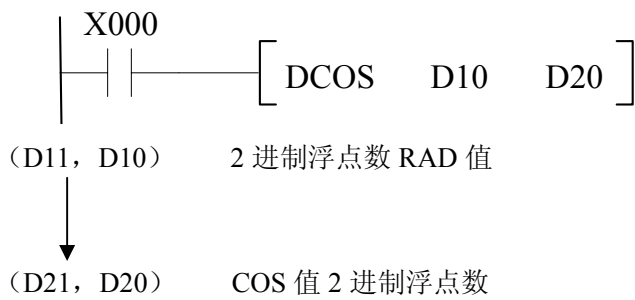
—— [COS S· D·]

S·：指定的 RAD 值

D·：COS 运算结果

S·所指定的 RAD 值等于 (角度 × π / 180) 求取 COS 值后存放在 D·所指定的暂存器中。

例



6. 应用指令解说

F132 TAN 浮点 TAN 运算

F		TAN			浮点 TAN 运算						S·	D·								
132	D			P																
	位元件				字元件															
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E			
S·													*	*			*			
D·													*	*						

指令格式：

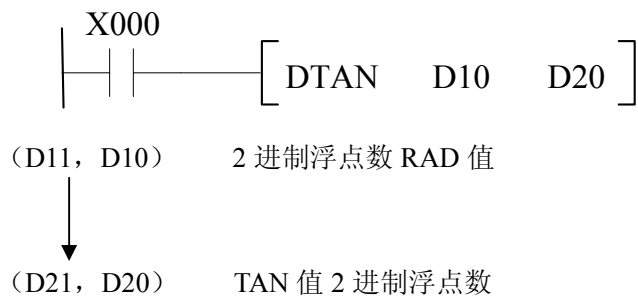
$$\text{——} \left[\text{TAN} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·：指定的 RAD 值

D·：TAN 运算结果

S·所指定的 RAD 值等于（角度 $\times \pi / 180$ ）求取 TAN 值后存放在 D·所指定的暂存器中。

例



F133 ASIN 浮点 ASIN 运算

F		ASIN				浮点 ASIN 运算								S ·	D ·			
133	D																	
	位元件				字元件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E	
S													*	*			*	
D													*	*				

指令符号:

$$\text{——} \left[\text{ASIN} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S: 指定 ASIN 的数据来源

D: ASIN 运算结果

S 所指定的内容 ARC SIN (SIN 的反函数) 值作为 2 进制浮点数存放于 D 中。

$-1 \leq S < 1$

例



(D11, D10) 2 进制浮点数 RAD 值



(D21, D20) ASIN 值 2 进制浮点数

F134 ACOS 浮点 ACOS 运算

F		ACOS				浮点 ACOS 运算						S·	D·					
134	D				P													
	位元件					字元件												
	X	Y	M	S		K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S														*	*			*
D														*	*			

指令符号:



S: 指定 ACOS 数据来源

D: ACOS 运算结果

S 所指定的内容 ARC COS (COS 的反函数) 值作为 2 进制浮点数存放于 D 中。

$-1 \leq S < 1$

例



(D11, D10) 2 进制浮点数 RAD 值



(D21, D20) ACOS 值 2 进制浮点数

6. 应用指令解说

F135 ATAN 浮点 ATAN 运算

F		ATAN			浮点 ATAN 运算							S·	D·										
135	D			P																			
	位元件				字元件																		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E						
S													*	*			*						
D													*	*									

指令符号:

$$\text{——} \left[\text{ATAN} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S: 指定 ATAN 数据来源

D: ATAN 运算结果

S 所指定的内容 ARCTAN (TAN 的反函数) 值作为 2 进制浮点数存放于 D 中。

$-\pi/2 \sim \pi/2$

例



(D11, D10) 2 进制浮点数 RAD 值



(D21, D20) ATAN 值 2 进制浮点数

F136 RAD 角度转换为弧度

F		RAD			角度转换为弧度							S·	D·								
136	D			P																	
	位元件				字元件																
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E				
S													*	*			*				
D													*	*							

指令符号:



S 和 D 都是 2 进制浮点形式

该指令实现从角度单位到弧度单位的转换

例:



F137 DEG 弧度转换为角度

F		DEG				弧度转换为角度						S •	D •				
137	D				P												
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	E
S													*	*			*
D													*	*			

指令符号：

— [DEG S D]

S 和 D 都是 2 进制浮点形式

该指令实现从角度单位到弧度单位的转换



F147 上下字节变换指令

F147 上下字节变换指令

功能编号	助记符	名称	页码	
F147	SWAP	上下字节变换指令	1	

6. 应用指令解说

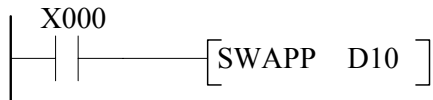
F147 SWAP 上下字节变换

F		SWAP			上下字节变换							S·				
147	D			P												
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S								*	*	*	*	*	*	*	*	*

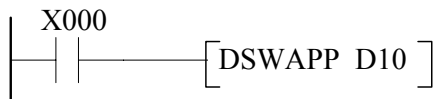
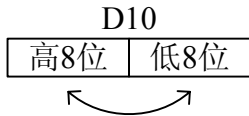
指令格式：

— [SWAP S·]

S·：欲执行上下位8位互相交换之单元



16位指令时，低8位与高8位交换。



32位指令时，各个低8位与高8位交换。



需要注意的是，该指令作为连续执行型指令时，各运算周期都变换。
此指令的作用与 F17 XCH 的扩展功能相同。

F156~F159 定位指令

F156~F159 定位指令

功能编号	助记符	名称	页码
F156	ZRN	原点回归	1
F157	PLSV	可变速脉冲输出	3
F158	DRVI	相对位置控制	4
F159	DRVA	绝对位置控制	6

6. 应用指令解说

F156 ZRN 原点回归

F		ZRN		原点回归	S1 · S2 · S3 · D ·												
156	D																
		位元件				字元件											
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·						*	*	*	*	*	*	*	*	*	*	*	*
S2 ·						*	*	*	*	*	*	*	*	*	*	*	*
S3 ·	*	*	*	*													
D ·		*															

指令格式:

$$\text{——} \left[\text{ZRN} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{S3} \cdot \quad \text{D} \cdot \right]$$

- 在执行 F158 (DRVI) 的相对位置控制和 F159 (DRVA) 的绝对位置控制时, 可编程控制器利用自身产生的正/反转脉冲进行当前值的增减, 并将其保存至当前值寄存器 (Y000: [D8141, D8140], Y001: [D8143, D8142])。由此, 机械的位置始终保持着, 但当可编程控制器断电时, 位置将消失。因此上电和初始运行时, 必须执行原点回归, 将机械动作的原点位置的数据事先写入。

S1: 原点回归速度

指定原点回归开始时的速度。

[16 位命令]: 10~32,767(Hz)

[32 位命令]: 10~100,000(Hz)

S2: 爬行速度

指定近点信号(DOG)变为 ON 后的低速部份的速度。

10~32,767(Hz)

S3: 近点信号

指定近点信号输入(a 接点输入)

当指定输入继电器 (X) 以外的元件时, 由于会受可编程控制器演算周期的影响, 会引起原点位置的偏移增大。

D: 脉冲输出对象编号

仅能指定 Y000 或 Y001。控制器的输出请务必以晶体管输出。

※ 1: BSP01 SR 机种不支持该指令

《清零信号的输出功能》

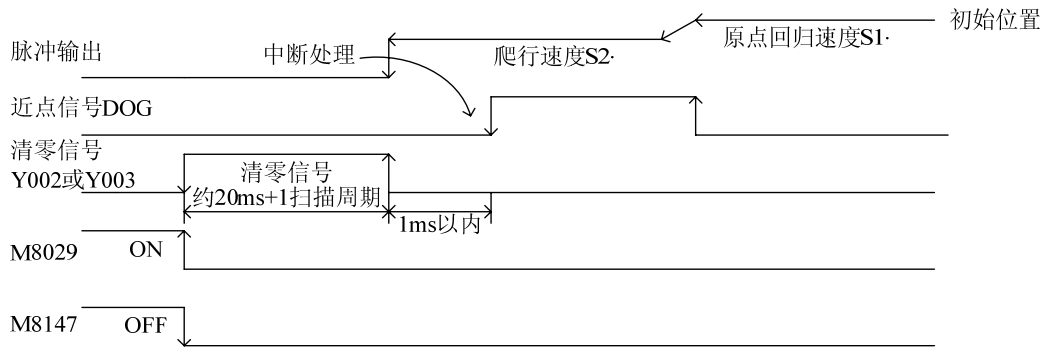
- 当 M8140 为 ON 时, 原点回归完成时, 可输出对伺服马达的清零信号。

- 清零信号的输出编号, 是分别依据脉冲输出编号所决定的:

脉冲输出[Y000]—>清除输出[Y002]

脉冲输出[Y001]—>清除输出[Y003]

6. 应用指令解说



《原点回归动作》

- 原点回归动作按照下述顺序进行。
 - ① 驱动命令后，以原点回归速度 S1·开始移动。
 - 当原点回归过程中，如果命令驱动接点 OFF，将不减速而停止。
 - 命令驱动接点 OFF 后，当脉冲输出中监视器（Y000：M8147，Y001：M8148）处于 ON 时，将不接受指令的再次驱动。
 - ② 当近点信号（DOG）由 OFF 变为 ON 时，减速至爬行速度 S2·。
 - ③ 近点信号（DOG）由 ON 变为 OFF 时，在脉冲输出停止的同时，向当前值寄存器（Y000：[D8141, D8140]，Y001：[D8143, D8142]）写入 0。另外，当 M8140（清零信号输出机能）ON 时，则同时输出清零信号。之后，当执行完成旗标（M8029）动作的同时，脉冲输出中监视器（Y000：[M8147]，Y001：[M8148]）变为 OFF。

《相关元件地址号》

- [D8141（上位），D8140（下位）]：输出到 Y000 的脉冲数。（使用 32 位）
[D8143（上位），D8142（下位）]：输出到 Y001 的脉冲数。（使用 32 位）
[M8145]：Y000 脉冲输出停止（立即停止）
[M8146]：Y001 脉冲输出停止（立即停止）
[M8147]：Y000 脉冲输出中监视器（BUSY/READY）
[M8148]：Y001 脉冲输出中监视器（BUSY/READY）

《注意事项》

- 由于不具备对应 DOG 搜寻功能，因此原点回归动作请从近点信号的前端开始进行。
- 原点回归中，当前值寄存器（Y000：[D8141, D8140]，Y001：[D8143, D8142]）的数值会朝向减少的方向变化。

6. 应用指令解说

F157 可变速脉冲

F		PLSV		可变速脉冲输出							S ·	D1 ·	D2 ·				
157	D																
		位元件				字元件											
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S ·						*	*	*	*	*	*	*	*	*	*	*	*
D1 ·			*														
D2 ·		*	*	*													

指令格式：

—— [PLSV S · D1 · D2 ·]

- 本指令为附带旋转方向的可变速脉冲输出命令。
- S·: 输出脉冲频率
 [16 位指令]: 1~32,767 (Hz), -1~-32,768 (Hz)
 [32 位指令]: 1~100,000 (Hz), -1~-100,000 (Hz)
- D1·: 脉冲输出对象编号
 仅能以 Y000 或 Y001 指定。控制器的输出请务必使用晶体管输出。
- D2·: 旋转方向信号的输出对象编号
 对应 S·的正负做如下动作: S·为正则置 ON; S·为负则置 OFF。
- ※ 1: BSP01 SR 机种不支持该指令
- 即使在脉冲输出中, 仍可任意变更输出脉冲频率 S·。
- 由于在起动/停止时不执行加减速, 如果有必要进行缓冲时, 请利用 F67 (RAMP) 等指令改变输出脉冲频率 S·的数值。
- 脉冲输出过程中, 命令驱动接点 OFF 时, 将不进行减速而直接停止。

《相关元件地址号》

- [D8141 (上位), D8140 (下位)]: 输出到 Y000 的脉冲数。逆转时会减少。(使用 32 位)
- [D8143 (上位), D8142 (下位)]: 输出到 Y001 的脉冲数。逆转时会减少。(使用 32 位)
- [M8145]: Y000 脉冲输出停止 (立即停止)
- [M8146]: Y001 脉冲输出停止 (立即停止)
- [M8147]: Y000 脉冲输出中监视器 (BUSY/READY)
- [M8148]: Y001 脉冲输出中监视器 (BUSY/READY)

注意事项

- 有关命令的驱动时间需加以注意。

6. 应用指令解说

F158 DRVI 相对位置控制

F					相对位置控制							S1 · S2 · D1 · D2 ·				
158	D	DRVI														
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*
D1 ·		*														
D2 ·		*	*	*												

指令格式:

$$\text{——} \left[\text{DRVI} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D1} \cdot \quad \text{D2} \cdot \right]$$

- 本指令为以相对驱动方式执行单速位置控制的指令。

S1: 输出脉冲数(相对指定)

[16 位命令]: -32,768~+32,767

[32 位命令]: -2,147,483,648~+2,147,483,647

S2: 输出脉冲频率

[16 位命令]: 10~32,767 (Hz)

[32 位命令]: 10~100,000 (Hz)

但是不能小于下页计算公式所示频率。

D1: 脉冲输出对象编号

仅能以 Y000 或 Y001 指定。控制器的输出请务必使用晶体管输出。

D2: 旋转方向信号的输出对象编号

对应 S 的正负做如下动作: S 为正则置 ON; S 为负则置 OFF。

※ 1: BSP01 SR 机种不支持该指令

- 输出脉冲数 S1 对应的相对位置如下:

向 Y000 的输出时: [D8141 (高位), D8140 (低位)] (使用 32 位)

向 Y001 的输出时: [D8143 (高位), D8142 (低位)] (使用 32 位)

反转时, 当前值寄存器的内容会减少。

- 命令执行中, 即使改变操作数的内容, 仍无法反映于目前的运转中。需待下次命令驱动时才有效。
- 命令执行中, 当命令驱动接点 OFF 时, 将减速停止。此时执行完成旗标 M8029 不产生动作。
- 所谓相对控制方式, 是指以当前位置为基点, 移动指定的带旋转方向的一定脉冲个数的控制方式。
- 实际能输出的输出脉冲频率的最低频率数, 根据以下公式决定。
输出脉冲频率的最低频率数=

$$\sqrt{\text{最高速度}[\text{D8147, D8146}]\text{Hz} \div (2 \times (\text{加减速时间}[\text{D8148}]\text{ms} \div 1000))}$$

《相关元件说明》

- [D8145]: 执行 FNC158 (DRVI), FNC159 (DRVA) 指令时的基地速度。
控制步进电机时, 设定速度时需考虑步进电机的共振区域和自动起动频率。
设定范围: 最高速度 (D8147, D8146) 的 1/10 以下。
超过该范围时, 自动降为最高速度的 1/10 数值运行。
- [D8147(高位), D8146(低位)]:
执行 FNC158 (DRVI), FNC159 (DRVA) 指令时的最高速度。
S2·指定的输出脉冲频率必须小于该最高速度。
设定范围: 10~100000(Hz)
- [D8148]: 执行 FNC158 (DRVI), FNC159 (DRVA) 指令时的加减速时间。
加减速时间表示到达最高速度 (D8147, D8146) 所需的时间。
因此, 当输出脉冲频率 S2·低于最高速度 (D8147, D8146) 时, 实际加减速时间会缩短。
设定范围: 50~5000(ms)
- [M8145]: Y000 脉冲输出停止 (立即停止)
- [M8146]: Y001 脉冲输出停止 (立即停止)
- [M8147]: Y000 脉冲输出中监控 (BUSY/READY)
- [M8148]: Y001 脉冲输出中监控 (BUSY/READY)

6. 应用指令解说

F159 DRVA 绝对位置控制

F		DRVA	绝对位置控制										S1 · S2 · D1 · D2 ·			
159	D															
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*
D1 ·		*														
D2 ·		*	*	*												

指令格式:

$$\text{——} \left[\text{DRVA } S1 \cdot S2 \cdot D1 \cdot D2 \cdot \right]$$

- 本指令为以绝对驱动方式执行单速位置控制的指令。

S1: 目标位置（绝对指定）

[16 位命令]: -32,768~+32,767

[32 位命令]: -2,147,483,648~+2,147,483,647

S2: 输出脉冲频率

[16 位命令]: 10~32,767 (Hz)

[32 位命令]: 10~100,000 (Hz)

D1: 脉冲输出对象编号

仅能以 Y000 或 Y001 指定。控制器的输出请务必使用晶体管输出。

D2: 旋转方向信号的输出对象编号

对应 S 和当前位置的差值做如下动作：差值为正则置 ON；差值为负则置 OFF。

※ 1: BSP01 SR 机种不支持该指令

- 输出脉冲数 S1 对应的相对位置如下：

向 Y000 的输出时：[D8141（高位）,D8140（低位）]（使用 32 位）

向 Y001 的输出时：[D8143（高位）,D8142（低位）]（使用 32 位）

反转时，当前值寄存器的内容会减少。

- 命令执行中，即使改变操作数的内容，仍无法反映于目前的运转中。需待下次命令驱动时才有效。
- 命令执行中，当命令驱动接点 OFF 时，将减速停止。此时执行完成旗标 M8029 不产生动作。
- 所谓绝对控制方式，是指以原点位置为基点，移动指定的带旋转方向的一定脉冲个数的控制方式。
- 实际能输出的输出脉冲频率的最低频率数，根据以下公式决定。

输出脉冲频率的最低频率数=

$$\sqrt{\text{最高速度}[\text{D8147},\text{D8146}]\text{Hz} \div (2 \times (\text{加减速时间}[\text{D8148}]\text{ms} \div 1000))}$$

《相关元件说明》

- [D8145]: 执行 FNC158 (DRVI), FNC159 (DRVA) 指令时的基地速度。
控制步进电机时, 设定速度时需考虑步进电机的共振区域和自动起动频率。
设定范围: 最高速度 (D8147, D8146) 的 1/10 以下。
超过该范围时, 自动降为最高速度的 1/10 数值运行。
- [D8147(高位), D8146(低位)]:
执行 FNC158 (DRVI), FNC159 (DRVA) 指令时的最高速度。
S2·指定的输出脉冲频率必须小于该最高速度。
设定范围: 10~100000(Hz)
- [D8148]: 执行 FNC158 (DRVI), FNC159 (DRVA) 指令时的加减速时间。
加减速时间表示到达最高速度 (D8147, D8146) 所需的时间。
因此, 当输出脉冲频率 S2·低于最高速度 (D8147, D8146) 时, 实际加减速时间会缩短。
设定范围: 50~5000(ms)
- [M8145]: Y000 脉冲输出停止 (立即停止)
- [M8146]: Y001 脉冲输出停止 (立即停止)
- [M8147]: Y000 脉冲输出中监控 (BUSY/READY)
- [M8148]: Y001 脉冲输出中监控 (BUSY/READY)

F160~F167 时钟运算

F160~F167 时钟运算

功能编号	助记符	名称	页码
F160	TCMP	时钟数据比较	1
F161	TZCP	时钟区域比较	2
F162	TADD	时钟数据加法运算	3
F163	TSUB	时钟数据减法运算	4
F166	TRD	时钟数据读取	5
F167	TWR	时钟数据写入	6

6. 应用指令解说

F160 TCMP 时钟数据比较

F		TCMP		时钟数据比较												S1 · S2 · S3 · S · D ·		
160			P															
	位组件				字符件													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z		
S1 ·					*	*	*	*	*	*	*	*	*	*	*	*		
S2 ·					*	*	*	*	*	*	*	*	*	*	*	*		
S3 ·					*	*	*	*	*	*	*	*	*	*	*	*		
S ·											*	*	*	*				
D ·		*	*	*														

指令格式：

—— [TCMP S1 · S2 · S3 · S · D ·]

S1·: 指定比较时间之“时”，指定范围为「0~23」

S2·: 指定比较时间之“分”，指定范围为「0~59」

S3·: 指定比较时间之“秒”，指定范围为「0~59」

S·: 指定时间数据之“时”，指定范围为「0~23」

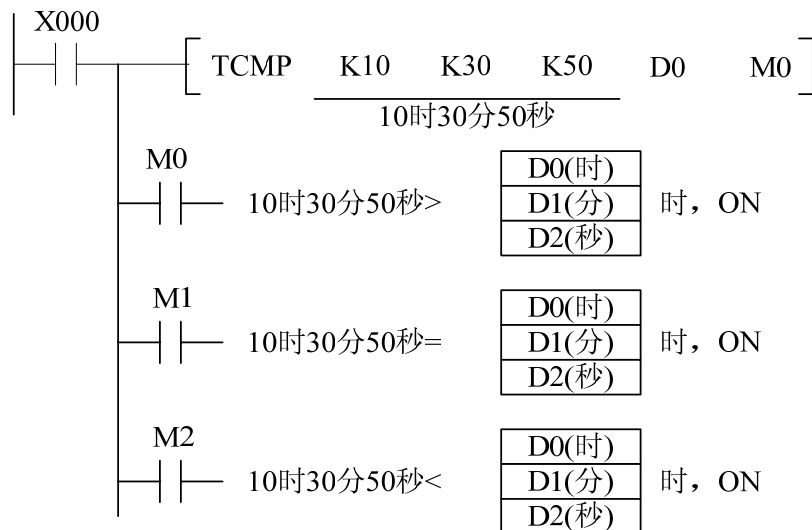
S+1: 指定时间数据之“分”，指定范围为「0~59」

S+2: 指定时间数据之“秒”，指定范围为「0~59」

D·: 比较结果，占三个连续的位元

※ 1: BSP01 SR 机种不支持该指令

例：



即使用X000停止执行TCMP指令时，M0~M2
仍然保持X000变为OFF前的状态

- 将源数据「S1·, S2·, S3·」的时间与以 S·起始的 3 点时间数据相比较，根据比较结果输出以 D·为起始的 3 点 ON/OFF 状态。

F161 TZCP 时钟区域比较

F		TZCP			时钟区域比较							S1 · S2 · S · D ·				
161				P												
	位组件				字符件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·											*	*	*	*		
S2 ·											*	*	*	*		
S ·											*	*	*	*		
D ·		*	*	*												

指令格式:

$$\text{—— [TZCP S1 · S2 · S3 · D ·]}$$

S1·: 指定比较时间之下限值

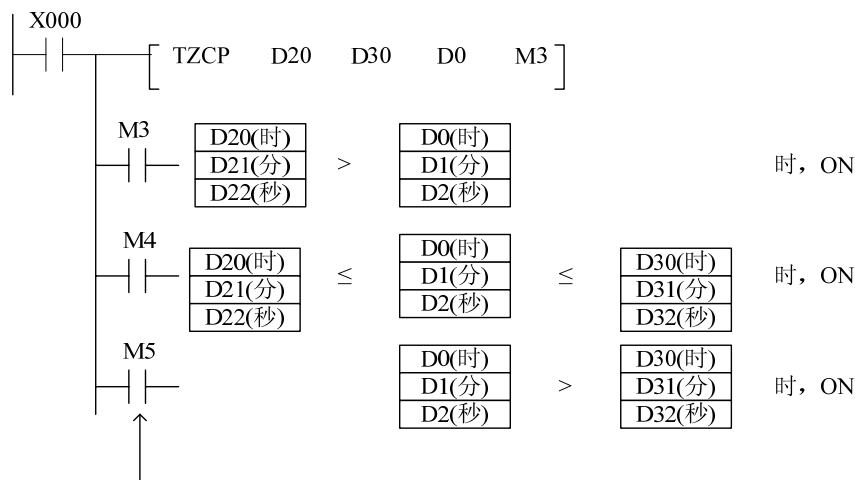
S2·: 指定比较时间之上限值

S·: 指定时间数据

D·: 比较结果, 占三个连续的位元

※ 1: BSP01 SR 机种不支持该指令

例:



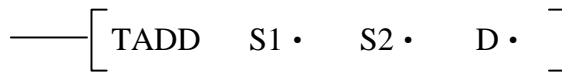
即使使用X000停止执行TZCP指令时, M3~M5 仍然保持X000变为OFF前的状态

- 将以S·为起始的3点时钟数据同上下两点S1·、S2·的所指定的时钟区域相比较, 根据比较结果输出以D·为起始的3点位元的ON/OFF状态。
 S1·、S1·+1、S1·+2: 设定比较时间下限值的”时”、”分”、”秒”。
 S2·、S2·+1、S2·+2: 设定比较时间上限值的”时”、”分”、”秒”。
 S·、S·+1、S·+2: 为指定时间的”时”、”分”、”秒”。
- 下限值S1·应小于等于上限值S2·, 当下限值S1·>上限值S2·时, 则指令以下限值S1·作为上下限值进行比较。

F162 TADD 时钟数据加法运算

F		PLSY			时钟数据加法运算						S1·	S2·	D·			
162				P												
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1·											*	*	*	*		
S2·											*	*	*	*		
D·											*	*	*	*		

指令格式



S1: 时间被加数。

S2: 时间加数。

D: 时间和

※ 1: BSP01 SR 机种不支持该指令

例:



D10 10(时)	+	D20 3(时)	→	D30 13(时)
D11 30(分)		D21 10(分)		D31 40(分)
D12 10(秒)		D22 5(秒)		D32 15(秒)

10时30分10秒

3时10分5秒

13时40分15秒

- 将S1·所指定的万年历数据的时、分、秒与S2·所指定的万年历数据的时、分、秒相加，所得到的结果存于指定D·所指定的缓存器时、分、秒当中。
- 当加算结果若超过24小时时，则进位旗标M8022=ON，将运算结果减去24小时后作为最终运算结果保存。

例:

18(时)	+	10(时)	→	4(时)
10(分)		20(分)		30(分)
30(秒)		5(秒)		35(秒)

- 加算结果若是等于0 (0 时0 分0 秒)，零旗标M1020=ON。

F163 TSUB 时钟数据减法运算

F		TSUB		时钟数据减法运算												S1·	S2·	D·	
163			P																
	位元件				字元件														
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z			
S1·											*	*	*	*					
S2·											*	*	*	*					
D·											*	*	*	*					

指令格式:

$$\text{——} \left[\text{TSUB} \quad \text{S1} \cdot \quad \text{S2} \cdot \quad \text{D} \cdot \right]$$

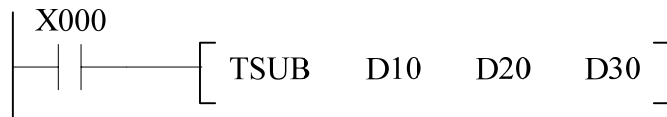
S1: 时间被减数

S2: 时间减数

D: 时间差

※ 1: BSP01 SR 机种不支持该指令

例:



D10 10(时)	-	D20 3(时)	→	D30 7(时)
D11 30(分)		D21 10(分)		D31 20(分)
D12 10(秒)		D22 5(秒)		D32 5(秒)

10时30分10秒

3时10分5秒

7时20分5秒

- 将S1·所指定的万年历数据的时、分、秒与S2·所指定的万年历数据的时、分、秒相减，所得到的结果存于指定D·所指定的缓存器时、分、秒当中。
- 当加算结果小于0小时时，则进位旗标M8022=ON，将运算结果加上24小时后作为最终运算结果保存。

例:

5(时)	-	18(时)	→	11(时)
20(分)		10(分)		10(分)
40(秒)		5(秒)		35(秒)

5时20分40秒

18时10分5秒

11时10分35秒

- 运算结果若是等于0 (0 时 0 分 0 秒)，零旗标 M1020=ON。

F166 TRD 时钟数据读取

F		TRD		时钟数据读取	D·											
166			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
D·											*	*	*	*		

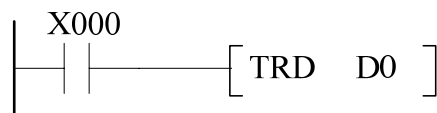
指令格式:



D·: 万年历现在时间读出后存放之起始装置, 占 7 点

※ 1: BSP01 SR 机种不支持该指令

例:



- 按照下列格式读取控制器的实时时钟数据。读取源为保存时钟数据的特殊数据寄存器 (D8013~D8019)。

	元件	项目	时钟数据		元件	项目
特殊数据 寄存器实 时时钟用	D8018	年 (公历)	2000~2099	→	D0	年 (公历)
	D8017	月	1~12	→	D1	月
	D8016	日	1~31	→	D2	日
	D8015	时	0~23	→	D3	时
	D8014	分	0~59	→	D4	分
	D8013	秒	0~59	→	D5	秒
	D8019	星期	0(日)~6(六)	→	D6	星期

F167 TWR 时钟数据写入

F		TWR		P	时钟数据写入							S·				
167																
位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·											*	*	*	*		

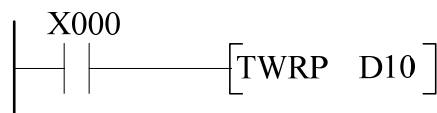
指令格式：



S·: 存放欲写入万年历新设定值之起始装置

※ 1: BSP01 SR 机种不支持该指令

例：



- 将设定时钟的数据写入控制器的实时时钟中，为了写入时钟数据，必须预先设定由 S·指定的元件地址号起始的 7 点元件。

	元件	项目	时钟数据		元件	项目	
时钟设定 用数据	D10	年（公历）	2000~2099	→	D8018	年（公历）	特殊数据 寄存器实 时时钟用
	D11	月	1~12	→	D8017	月	
	D12	日	1~31	→	D8016	日	
	D13	时	0~23	→	D8015	时	
	D14	分	0~59	→	D8014	分	
	D15	秒	0~59	→	D8013	秒	
	D16	星期	0(日)~6(六)	→	D8019	星期	

F170~F171 外围设备

F170~F171 外围设备

功能编号	助记符	名称	页码
F170	GRY	GRY 码转换	1
F171	GBIN	GRY 逆转换	2
F172	MATH	四则运算	3
F173	EMATH	浮点数四则运算	4

6. 应用指令解说

F170 GRY BIN—GRY 码变换

F		GRY				BIN→GRY 码变换						S·	D·				
170	D	P															
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
S·					*	*	*	*	*	*	*	*	*	*	*	*	
D·								*	*	*	*	*	*	*	*	*	

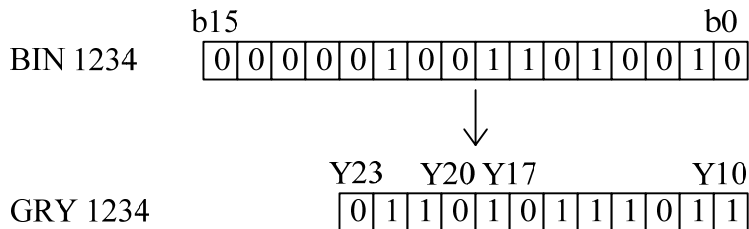
指令格式：

$$\text{---} \left[\text{GRY} \quad \text{S} \cdot \quad \text{D} \cdot \right]$$

S·: 来源装置

D·: 存放 GRY 码的装置

示例：



- 将 BIN 数据转换为葛莱码并传送的指令。
- 最大可以进行 32 位的葛莱码转换。
- 对于 S·的数值，仅在一下范围内有效。
 - 16 位运算时：0~32,767
 - 32 位运算时：0~2,147,483,647

F171 GBIN GRY—BIN 码变换

F		GBIN				GRY→BIN 码变换						S·	D·			
171	D	P														
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·					*	*	*	*	*	*	*	*	*	*	*	*
D·								*	*	*	*	*	*	*	*	*

指令格式：

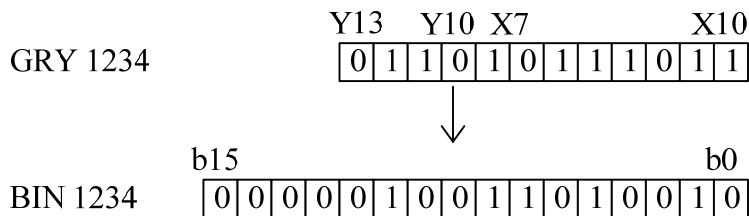


S: 来源装置

D: 存放 GRY 码逆变换的装置

将S所指定装置的格雷码值逆变换为BIN值后存放到D所指定之装置中。

示例：



- 将葛莱码转换为 BIN 数据并传送的指令。
- 最大可以进行 32 位的葛莱码逆转换。
- 对于 S 的数值，仅在一下范围内有效。
 - 16 位运算时：0~32,767
 - 32 位运算时：0~2,147,483,647

6. 应用指令解说

F172 MATH 四则运算

F		MATH		四则运算								D·	S·			
172	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·													*	*		
D·													*	*		

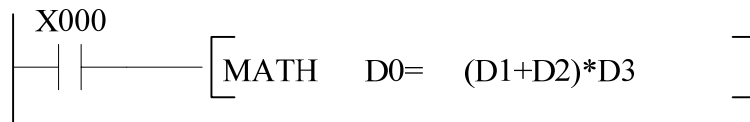
指令格式:

$$\text{——} \left[\text{MATH } D\cdot = S\cdot \right]$$

S·: 来源装置,各个元件之间用操作符连接,以表示进行运算,最多可以有 16 个元件进行运算

D·: 存放运算结果的装置

示例:



- 将 S·进行四则运算后将结果存储到 D·
- S·内操作符可以用+,-,*/连接各操作数,且可以用()。
- S·内最多可以进行 16 个操作数的运算
- 进行四则运算时候,D7950-D7999 可能会被占用作为中间结果的临时存储区域,(SR 机种 D462-D511 可能会被占用)。
- 操作数为有符号数
 - 16 位运算时: -32768~32,767
 - 32 位运算时: -2,147,483,648~2,147,483,647

6. 应用指令解说

F173 EMATH 浮点数四则运算

F		EMATH		浮点数四则运算								D·	S·			
172	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S·													*	*		
D·													*	*		

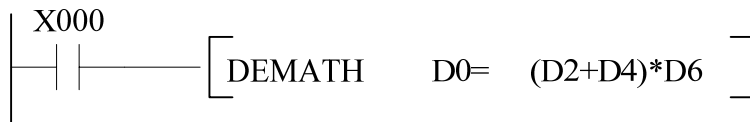
指令格式:

$$\text{——} \left[\text{DEMATH} \quad \text{D} \cdot = \quad \text{S} \cdot \right]$$

S·: 来源装置,各个元件之间用操作符连接,以表示进行运算,最多可以有 16 个元件进行运算

D·: 存放运算结果的装置

示例:



- 将 S·进行四则运算后将结果存储到 D·
- S·内操作符可以用+,-,*/连接各操作数,且可以用()。
- S·内可以用 SIN,COS,TAN,ASIN,ACOS,ATAN 来对操作数进行计算,但这些三角运算后的操作数必须加括号,例如:SIN(D0),COS(D1+D2)
- S·内最多可以进行 16 个操作数的运算
- 进行四则运算时候,D7950-D7999 可能会被占用作为中间结果的临时存储区域,(SR 机种 D462-D511 可能会被占用).
- 操作数为浮点数

F188~192 外围通信

外围通信

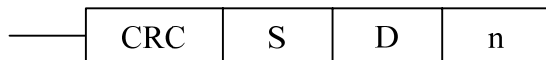
功能编号	助记符	名称	页码
188	CRC	循环冗余校验	1
190	DTLK	资料连线	3
191	RMIO	远距离 I/O 连线	9
193	DTLK2	资料连线 2	15

6. 应用指令解说

F188 CRC 循环冗余校验

F		CRC		循环冗余校验								S	D	n		
188			P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S													*	*		
D													*	*		
n					*								*	*		

指令符号：



S: 数据起始元件

D: 生成 CRC 值后的存放位置

n: 数据点数

以 S 中指定的软元件为起始的 n 点 8 位数据 (字节单位), 对其生成的 CRC 值后保存到 D 中。在这个指令中有 8 位和 16 位的转换模式, 根据 M8161 的 ON/OFF 来切换转换模式。

为了生成 CRC 值, 使用了 $[X^{16}+X^{15}+X^2+X^1]$ 的生成多项式, 此外, 针对 CRC 值, 还有各种标准化的生成多项式, 请注意, 如果使用不同的生成多项式, 会产生完全不同的 CRC 值

参考: 主要的 CRC 生成多项式:

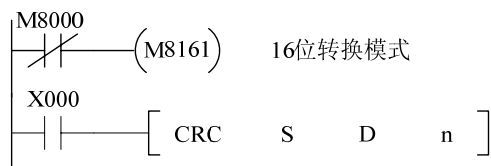
名称	生成多项式
CRC-12	$X^{12}+X^{11}+X^3+X^2+X+1$
CRC-16	$X^{16}+X^{15}+X^2+1$
CRC-32	$X^{32}+X^{26}+X^3+X^{22}+X^{16}+X^{12}+X^{11}+X^{10} X^8+X^7+X^5+X^4X^2+X+1$
CRC-CCITT	$X^{16}+X^{12}+X^5+1$

相关软元件

软元件	内容	备注
[M8161]	ON	CRC 在 8 位模式下动作
	OFF	CRC 在 16 位模式下动作
		RUN→OFF 时清除

16 位转换模式[M8161=OFF]

在 16 位模式下, 对 S 软元件的高 8 位 (字节) 和低 8 位 (字节) 进行运算。在 D 指定的 1 点软元件的 16 位中保存运算结果。



6. 应用指令解说

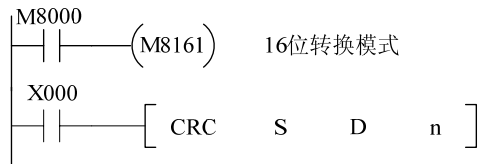
当 S=D100, D=D0, n=6 时:

			软元件	对象数据	
				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	0101H
		高字节	D100 高字节	01H	
	S+1	低字节	D101 低字节	03H	CD03H
		高字节	D101 高字节	CDH	
	S+2	低字节	D102 低字节	6BH	056BH
		高字节	D102 高字节	05H	
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	42H	8242H
		高字节	D0 高字节	82H	

8 位转换模式[M8161=ON]

在 8 位转换模式下, 仅对 S 软元件的低 8 位执行运算。

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, 在 D+1 中保存高 8 位, 计算



当 S=D100, D=D0, n=6 时:

			软元件	对象数据内容	
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	低字节	D101 低字节	01H	
	S+2	低字节	D102 低字节	03H	
	S+3	低字节	D103 低字节	CDH	
	S+4	低字节	D104 低字节	6BH	
	S+5	低字节	D105 低字节	05H	
	S+n-1	低字节	——		
低字节		——			
保存 CRC 值的地址	D	低字节	D0 低字节	42H	
	D+1	低字节	D1 低字节	82H	

6. 应用指令解说

F190 DTLK 资料连线

F		DTLK		资料连线										K		
190																
K	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
					0, 1	0, 1										

BSP01 采用此模式，他们能链接一个小规模系统中的数据。BSP01 主站最多可以 15 台 BSP01 从站链接通讯。

DTLK 在 2 个通讯接口同时使用时，只能优先使用先符合使能条件的一条指令。

通讯帧格式及波特率设置由 D8120、D8320 决定（根据串口选择）；

指令符号：

—— [DTLK K]

K： 串口选择，范围为常数 0~1；

0： 内建 RS485 通讯接口；

1： RS485 或 RS232 扩充卡之通讯接口。

※ 1： BSP01 SR 机种不支持该指令

RS485 或 RS232 扩充卡之通讯接口（所有型号之机种皆可选购），RS485 通讯接口（仅 BSP01 AR/T 型号之机种内建），以上两个接口可实现资料连线通信功能，但不能同时设定为资料连线功能；

	规格
通信速度	EIA RS-485 标准
传送速度	9600bps~307200bps
连接从站数	最多 15 局
连接区域	D0~D157, M2000~M3023
每 1 站连线资料长度	最大 64 位信息+8 字信息
传输线	隔离双绞线，共线连接，2 线式 总长 500m(76800bit/s), 1km(38400bit/s)

配线方法

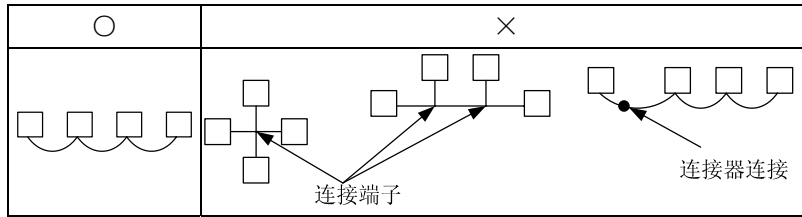


注 1： SHLD 端子请做第三种接地，不做第三种接地时，噪音将造成误动作。

注 2： 通信线的分支请不要超过 3 条线。

注 3： R 为通信终端电阻，规格为 120 Ω，1/4W。

6. 应用指令解说



使用的相关标记及数据寄存器

1) 辅助继电器

辅助继电器	特性	名称	描述	回应类型
M8400	只读	主站通讯错误	当主站发生通讯错误时为 ON	从
M8401	只读	从站 1 通讯错误	当从站 1 发生通讯错误时为 ON	主/从
M8402	只读	从站 2 通讯错误	当从站 2 发生通讯错误时为 ON	主/从
...
M8414	只读	从站 14 通讯错误	当从站 14 发生通讯错误时为 ON	主/从
M8415	只读	从站 15 通讯错误	当从站 15 发生通讯错误时为 ON	主/从
M8416	只读	数据通讯状态	DTLK 当与其它站点通讯时为 ON	主/从
M8417	只读	Data Link 模式	扩充通讯接口为 Data Link 功能运行中为 ON	主/从
M8418	只读	Data Link 模式	RS485 通讯接口为 Data Link 功能运行中为 ON	主/从

2) 资料寄存器

资料寄存器	特性	名称	描述	回应类型
D8173	只读	站点号	存贮自己的站点号	主/从
D8174	只读	从站点总数	存储从站点的总数	主/从
D8175	只读	刷新范围	存储刷新范围 (Data Link 通讯模式)	主/从
D8176	写	站点号设置	设置自己的站点号	主/从
D8177	写	总从站点数设置	设置从站点的总数	主
D8178	写	Data Link 模式设置	设置刷新范围 (Data Link 通讯模式)	主
D8179	读/写	重试次数设置	设置重试次数	主
D8180	读/写	通讯超时设置	设置通讯超时 (Time-Out)	主
D8401	只读	当前网络扫描时间	存储当前网络扫描时间	主/从
D8402	只读	最大网络扫描时间	存储最大网络扫描时间	主/从
D8403	只读	主站点的通讯错误计数	主站点的通讯错误次数	从
D8404	只读	从站点 1 的通讯错误次数	从站点 1 的通讯错误次数	主/从
D8405	只读	从站点 2 的通讯错误次数	从站点 2 的通讯错误次数	主/从
...
D8411	只读	从站点 8 的通讯错误次数	从站点 8 的通讯错误次数	主/从
...
D8417	只读	从站点 14 的通讯错误次数	从站点 14 的通讯错误次数	主/从

6. 应用指令解说

D8418	只读	从站点 15 的通讯错误次数	从站点 15 的通讯错误次数	主/从
D8419	只读	主站点的通讯错误代码	主站点的通讯错误代码	从
D8420	只读	从站点 1 的通讯错误代码	从站点 1 的通讯错误代码	主/从
D8421	只读	从站点 2 的通讯错误代码	从站点 2 的通讯错误代码	主/从
...
D8427	只读	从站点 8 的通讯错误代码	从站点 8 的通讯错误代码	主/从
...
D8433	只读	从站点 14 的通讯错误代码	从站点 14 的通讯错误代码	主/从
D8434	只读	从站点 15 的通讯错误代码	从站点 15 的通讯错误代码	主/从

设置

当程序运行或 BSP01 电源打开时，Data Link 的每一个设置都变为有效；

1) 设定站号 (D8176)

设定 0~15 的值到特殊数据寄存器 D8176 中，0 为主站号，1~15 表示为从站 1~15。

2) 设定从站总数 (D8177)

设定 1~15 的值到特殊数据寄存器 D8177 中表示从站总数（默认为 7），从站不需要此设定。

用户应根据需要设置从站总数，可提高数据刷新速度。

3) 设置刷新范围 (D8178)

设定 0~2 的值到特殊数据寄存器 D8178 中（默认为 0）表示为主站刷新范围、即 Data Link 通讯的工作模式，从站不需要此设定；

D8178 的值		0	1	2
Data Link 模式		模式 0	模式 1	模式 2
刷新范围	位软组件 (M)	0 点	32 点	64 点
	字软组件 (D)	4 点	4 点	8 点

不同模式通讯时刷新的软组件

站点号	模式 0		模式 1		模式 2	
	位软组件 (M)	字软组件 (D)	位软组件 (M)	字软组件 (D)	位软组件 (M)	字软组件 (D)
第 0 号	—	D0~D3	M2000~M2031	D0~D3	M2000~M2063	D0~D7
第 1 号	—	D10~D13	M2064~M2095	D10~D13	M2064~M2127	D10~D17
第 2 号	—	D20~D23	M2128~M2159	D20~D23	M2128~M2191	D20~D27
第 3 号	—	D30~D33	M2192~M2223	D30~D33	M2192~M2255	D30~D37
第 4 号	—	D40~D43	M2256~M2287	D40~D43	M2256~M2319	D40~D47
第 5 号	—	D50~D53	M2320~M2351	D50~D53	M2320~M2383	D50~D57
第 6 号	—	D60~D63	M2384~M2415	D60~D63	M2384~M2447	D60~D67
第 7 号	—	D70~D73	M2448~M2479	D70~D73	M2448~M2511	D70~D77
第 8 号	—	D80~D83	M2512~M2543	D80~D83	M2512~M2575	D80~D87
第 9 号	—	D90~D93	M2576~M2607	D90~D93	M2576~M2639	D90~D97
第 A 号	—	D100~D103	M2640~M2671	D100~D103	M2640~M2703	D100~D107
第 B 号	—	D110~D113	M2704~M2735	D110~D113	M2704~M2767	D110~D117
第 C 号	—	D120~D123	M2768~M2799	D120~D123	M2768~M2831	D120~D127

6. 应用指令解说

第 D 号	—	D130~D133	M2832~M2863	D130~D133	M2832~M2895	D130~D137
第 E 号	—	D140~D143	M2896~M2927	D140~D143	M2896~M2959	D140~D147
第 F 号	—	D150~D153	M2960~M2991	D150~D153	M2960~M3023	D150~D157

4) 设定重试次数 (D8179)

设定 0~10 的值到特殊数据寄存器 D8179 中 (默认为 3), 从站点不需要此设定; 如果一个主站试图以此重试次数与从站通讯, 此从站点将发生通讯错误。

5) 设定通讯超时 (D8180)

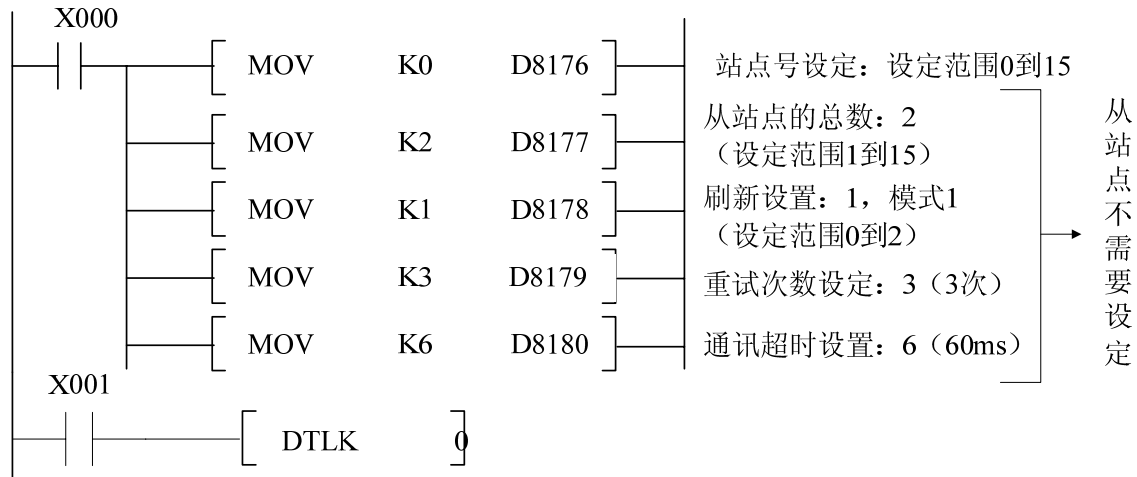
设定 5~255 的值到特殊数据存储器 D8180 中 (默认为 5), 此值乘以 10 为通讯超时的等待时间 (ms)。

6) 当前网络扫描时间 (D8401)

当前 Data Link 网络扫描一次所花的时间, 此值乘以 10 为扫描时间 (ms)。

7) 网络最大扫描时间 (D8402)

用于进行设置的程式:



出错代码

当通讯发生错误时, 特殊辅助线圈 M8400~M8415 为通讯出错状态指示, 错误代码被保存至特殊数据寄存器 (D8419~D8434) 中。

错误	代码含义	错误站点	检查站点	说明	检查点
01H	通讯超时错误	L	M	主站发送请求到从站后, 无应答, 时间超过通讯超时时间	检查配线、电源 run/stop 状态等。
02H	站号编号错误	L	M	站号编号不符合主站和从站间的关系	检查配线
03H	通讯计数器错误	L	M	通讯计数器不符合主站和从站间的关系	检查配线
04H	通讯格式错误	L	M, L	从站的通讯格式不正确	检查配线, DTLK 的资源设置;
11H	通讯超时错误	M	L	从站发送应答给主站后, 主站不再将请求发送到下一从站	检查配线、电源 run/stop 状态等。
14H	通讯格式错误	M	L	主站的通讯格式不正确	检查配线, DTLK 的资

6. 应用指令解说

					源设置
21H	无从站错误	L	L *1	网络里的站号编号不正确	检查站点编号设置
22H	站点编号错误	L	L *1	站点编号不符合主站和从站间的关系	检查配线
23H	通讯计数器错误	L	L *1	通讯计数器不符合主站和从站间的关系	检查配线
31H	无接收通讯参数错误	L	L *2	从站在通讯参数设置前就从主站接收请求	检查配线、低电源、run/stop 状态
32H	其他错误	L	L *1	网络通讯命令错误	检查网络设置

M: 主站

L: 从站

*1: 另外一个从站

2*: 独立站点

通信时序与传输所需的时间

主站与各从站的通信和主站的扫描周期非同步。

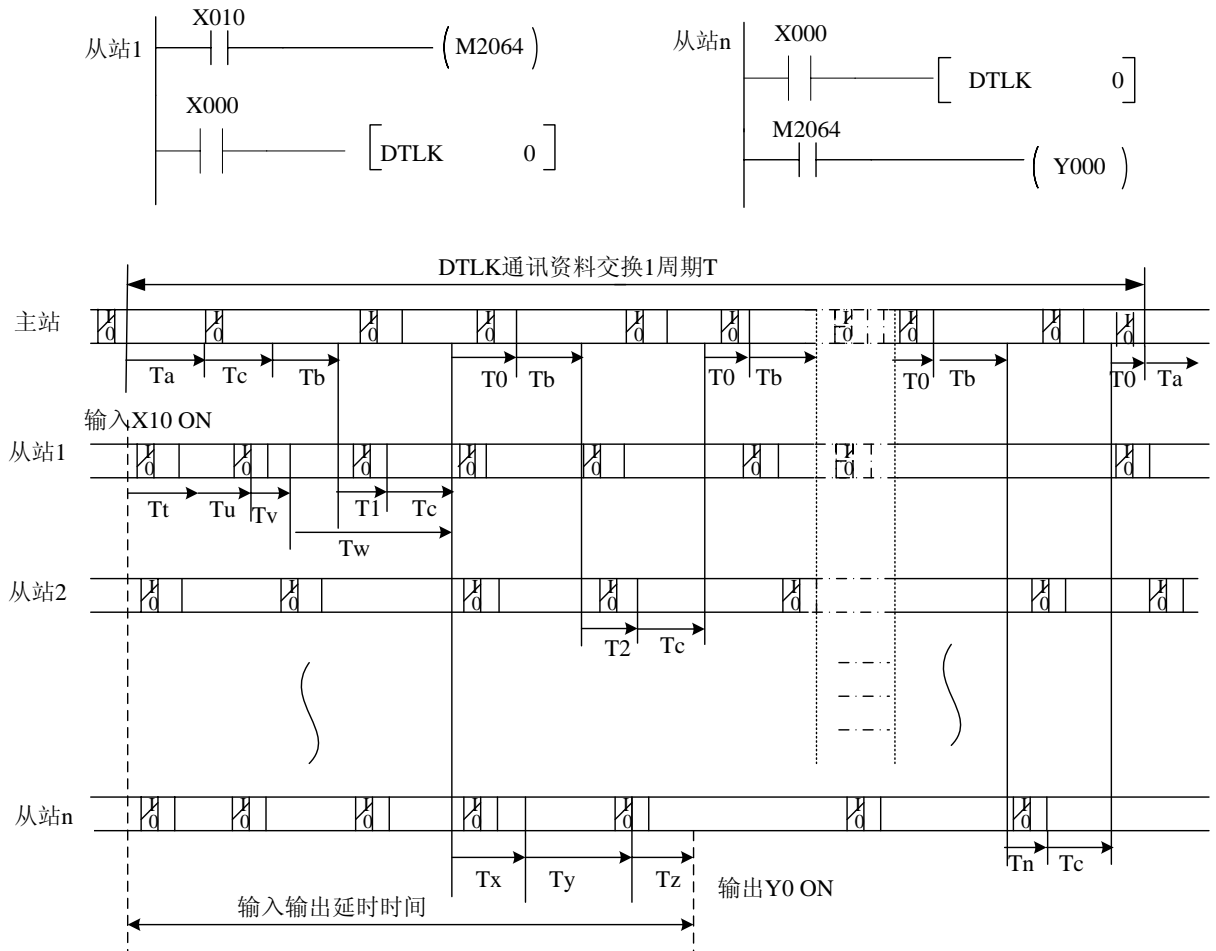
从站与主站的通信和从站的扫描周期非同步。

通讯时序图及延迟时间

资料连线在接收资料时，将发生延时，结合通讯扫描周期，时序如下：

例：从站 1 的 M2064 由 X010 控制，通过 TDLK 指令将 M2064 状态传送到其他网络的各站点；

6. 应用指令解说



传输所需时间

资料连续主站与所有从站通讯所需的时间（通信周期时间，此时间不占主站的扫描周期）

$$T = T_a + T_c + [T_b + T_n + T_c + T_0] * n_1 \{ + [T_b + T_n + D8180 * 10] * n_2 \};$$

其中 T_a : 主站向从站发送网络配置命令的传输时间;

T_b : 主站向从站发送资料交换请求命令的传输时间;

T_c : 网络资料交换的传输时间 (DTLK 模式不同所需时间不同);

T_0 : 主站检测到通讯状态的时间 ($0 \sim$ 一个扫描周期);

T_n : 从站 n 检测到通讯状态的时间 ($0 \sim$ 一个扫描周期);

($n_1 + n_2$): 为主站设定 DTLK 从站的个数 (D8177 = 1~15), n_1 为实际存在个数; n_2 为连接丢失的从站个数 (设定后实际不存在或通讯连接不正常个数 $0 \sim 15$);

D8180 为设定 timeout 的值;

延时时间

T_t : 输入端的延时;

T_u : PLC 检测到输入状态的时间 (最大 1 扫描周期);

T_v : 发送 PLC 接收输入状态后到程式扫描的时间;

T_w : 演算结果传送完毕的时间 (最大 1 网络扫描周期 T);

6. 应用指令解说

Tx: 接收 PLC 将资料写入 PLC 寄存器的时间 (最大 1 扫描周 n);

Ty: 接收 PLC 程式演算到输出的时间 (1 扫描周期);

Tz: 输出端的延时;

不同波特率下的命令传输时间:

通讯波特率 (bps)	Ta (ms)	Tb (ms)	Tc (ms)		
			DTLK 模式 0	DTLK 模式 1	DTLK 模式 2
9600	21.8	12.6	31.0	40.1	67.6
19200	10.9	6.3	15.5	20.1	33.8
38400	5.5	3.2	7.8	10.0	16.9
57600	3.7	2.1	5.2	6.7	11.3
76800	2.8	1.6	3.9	5.0	8.5
128000	1.7	1.0	2.4	3.0	5.1
153600	1.4	0.8	2.0	2.5	4.3
307200	0.7	0.4	1.0	1.3	2.2

F191RMIO 远距离 I/O 连线

F		RMIO		远距离 I/O 连线												K	
191																	
	位元件				字元件												
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z	
K					0, 1	0, 1											

功能: BSP01 采用此模式,可增加主站设备输入输出的点数, BSP01 主站最多可以 4 台 BSP01 从站链接通讯。

RMIO 在 2 个通讯接口同时使用时,只能优先使用先符合使能条件的一条指令。

通讯帧格式及波特率设置由 D8120、D8320 决定 (根据串口选择);

注 1: BSP01 在远程 I/O 模式下作为从站时,只能用作主站设备的扩充 I/O 功能,只能运行 RMIO 指令而不能执行其他用户程序。

注 2: BSP01 被使能为远程 I/O 模式从站时,只能停止运行程式才能转为非远程 I/O 模式。

指令符号:



6. 应用指令解说

—— [RMIO K]

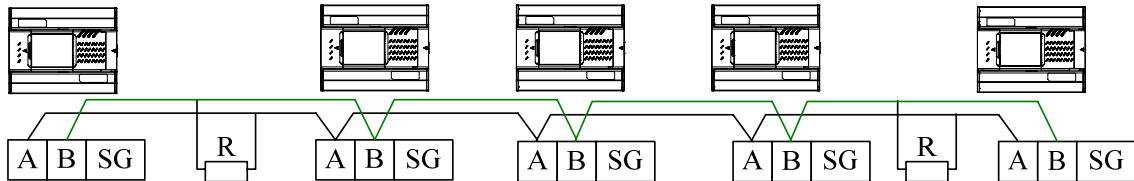
K: 串口选择, 常数 0~1;

0: 内建 RS485 通讯接口;

1: RS485 或 RS232 扩充卡之通讯接口。

※ 1: BSP01 SR 机种不支持该指令

远距离 I/O, BSP01 主站最多可 4 如 BSP01 从站通信。



项目	说明	
通信规格	EIA RS485 标准	
传送速度	9600bps~307200bps	
连接从站数	最多 4 从站	
远距离 I/O 区域 (主站配制)	从站 1	输入: 36 点 (M4200~M4235); 输出 24 点 (M4600~M4623)
	从站 2	输入: 36 点 (M4240~M4275); 输出 24 点 (M4624~M4647)
	从站 3	输入: 36 点 (M4280~M4315); 输出 24 点 (M4648~M4671)
	从站 4	输入: 36 点 (M4320~M4355); 输出 24 点 (M4672~M4695)
传输线	隔离双绞线, 共线连接, 2 线式, 总长 500m (76800bit/s), 1km (38400bit/s)	

RS485 或 RS232 扩充卡之通讯接口 (所有型号之机种皆可选购), RS485 通讯接口 (仅 BSP01 AR/T 型号之机种内建), 以上两个接口可实现远程 IO 通信功能, 不能同时使用。

注: 远距离从站必须为 BSP01 的基本组, 不能使用扩充模组或其他模组。

使用的相关标记及数据寄存器

1) 辅助继电器 M

辅助继电器	特性	名称	描述	回应类型
M8335	只读	数据通讯状态	RMIO 当与其它站点通讯时为 ON	主/从
M8336	只读	主站通讯错误	当主站发生通讯错误时为 ON	从
M8337	只读	从站 1 通讯错误	当从站 1 发生通讯错误时为 ON	主/从
M8338	只读	从站 2 通讯错误	当从站 2 发生通讯错误时为 ON	主/从
M8339	只读	从站 3 通讯错误	当从站 3 发生通讯错误时为 ON	主/从
M8340	只读	从站 4 通讯错误	当从站 4 发生通讯错误时为 ON	主/从
M8341	只读	为远程 IO 模式	扩展通讯为远程 IO 模式	主/从
M8342	只读	为远程 IO 模式	RS485 为远程 IO 模式	主/从

2) 资料寄存器 D

资料寄存器	特性	名称	描述	回应类型
D8373	只读	站点号	存贮自己的站点号	主/从

6. 应用指令解说

D8374	只读	从站点总数	存储从站点的总数	主/从
D8376	写	站点号设置	设置自己的站点号	主/从
D8377	写	总从站点数设置	设置从站点的总数	主
D8379	读/写	重试次数设置	设置重试次数	主
D8380	读/写	通讯超时设置	设置通讯超时 (Time-Out)	主/从
D8331	只读	当前网络扫描时间	存储当前网络扫描时间	主
D8332	只读	最大网络扫描时间	存储最大网络扫描时间	主
D8333	只读	主站点的通讯错误计数	主站点的通讯错误次数	从
D8334	只读	从站点 1 的通讯错误次数	从站点 1 的通讯错误次数	主/从
D8335	只读	从站点 2 的通讯错误次数	从站点 2 的通讯错误次数	主/从
D8336	只读	从站点 3 的通讯错误次数	从站点 3 的通讯错误次数	主/从
D8337	只读	从站点 4 的通讯错误次数	从站点 4 的通讯错误次数	主/从
D8338	只读	主站点的通讯错误代码	主站点的通讯错误代码	从
D8339	只读	从站点 1 的通讯错误代码	从站点 1 的通讯错误代码	主/从
D8340	只读	从站点 2 的通讯错误代码	从站点 2 的通讯错误代码	主/从
D8341	只读	从站点 3 的通讯错误代码	从站点 3 的通讯错误代码	主/从
D8342	只读	从站点 4 的通讯错误代码	从站点 4 的通讯错误代码	主/从

设置

当程序运行或 BSP01 电源打开时，远程 I/O 的每一个设置都变为有效。

1) 设定站号 (D8376)

设定 0~4 的值到特殊数据寄存器 D8376 中，0 为主站号，1~4 表示为从站 1~4。

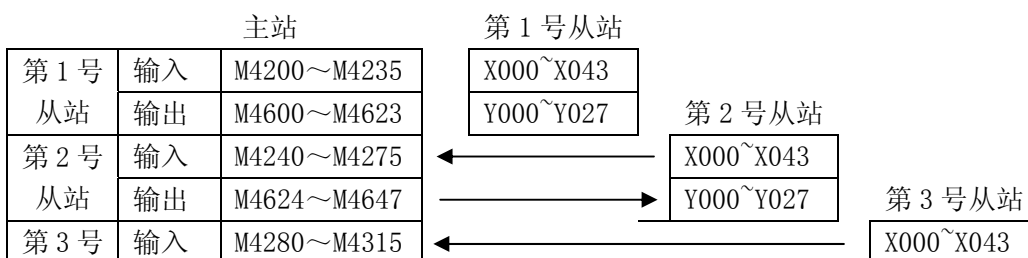
2) 设定从站总数 (D8377)

设定 1~4 的值到特殊数据寄存器 D8377 中表示从站总数（默认为 4），从站不需要此设定。

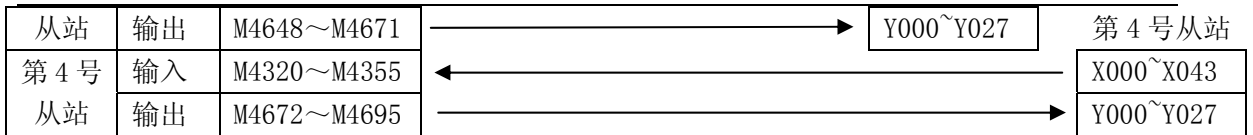
用户应根据实际需要设置从站数，可提高通信效率及数据刷新速度。

通讯时对应使用组件

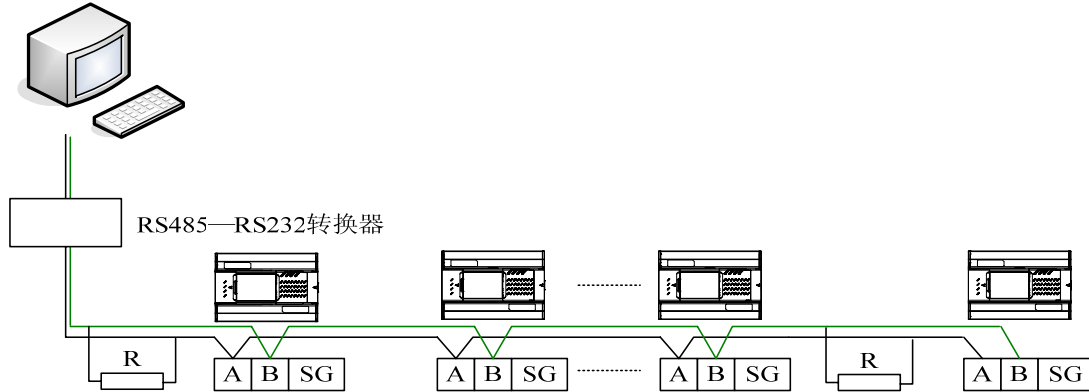
主站的远程 I/O 区域，组成情形为：



6. 应用指令解说



配线方法

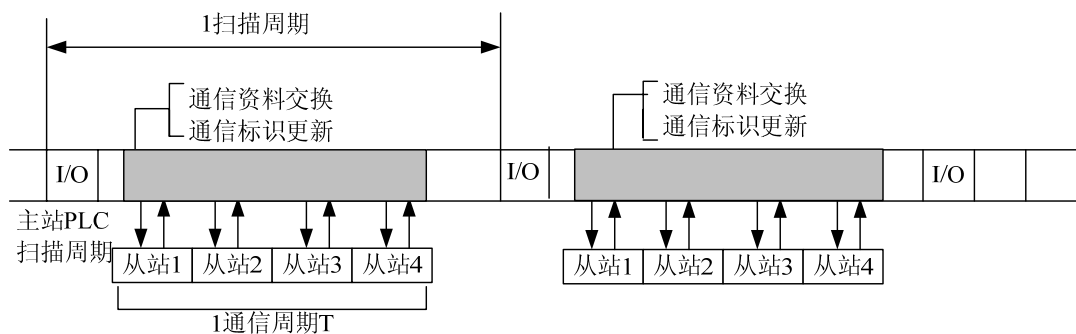


注 1. SHLD 端子请做第三种接地，不做第三种接地时，噪音将造成误动作。

注 2: 通信线的分支请不要超过 3 条线。

注 3: R 为通信终端电阻，规格为 120Ω，1/4W。

通信时序与传输所需的时间



通信的时序

主站与各从站的通信，远距离 I/O 资料交换及通信标识的更新和从站的扫描周期同步，此处理时间（1 通信周期时间）将增加主站的扫描时间

当主站与从站的通信发行异常时，远距离 I/O 通信及 PLC 的运转停止且进入异常状态。

异常发生的可能原因如下：

6. 应用指令解说

CRC 检查异常的发生

从站于停止模式或异常状态

从站未接线，或连接线继线等

当主站于停止模式或异常状态时，不与任何从站作通信。

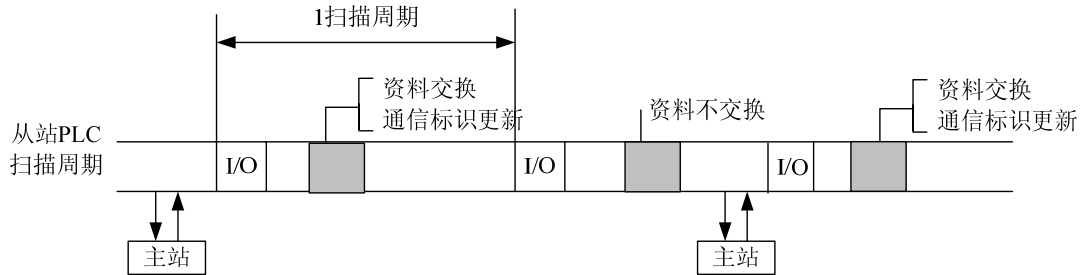
主站与从站间的通讯格式配置不一致。

从站的通信时序

从站与主站的通信与从站的扫描周期异步。

从站在与主站通信完毕后扫描周期做连结资料的交换及通信标识的更新。

此处理将增加从站的扫描时间约 0.2ms。



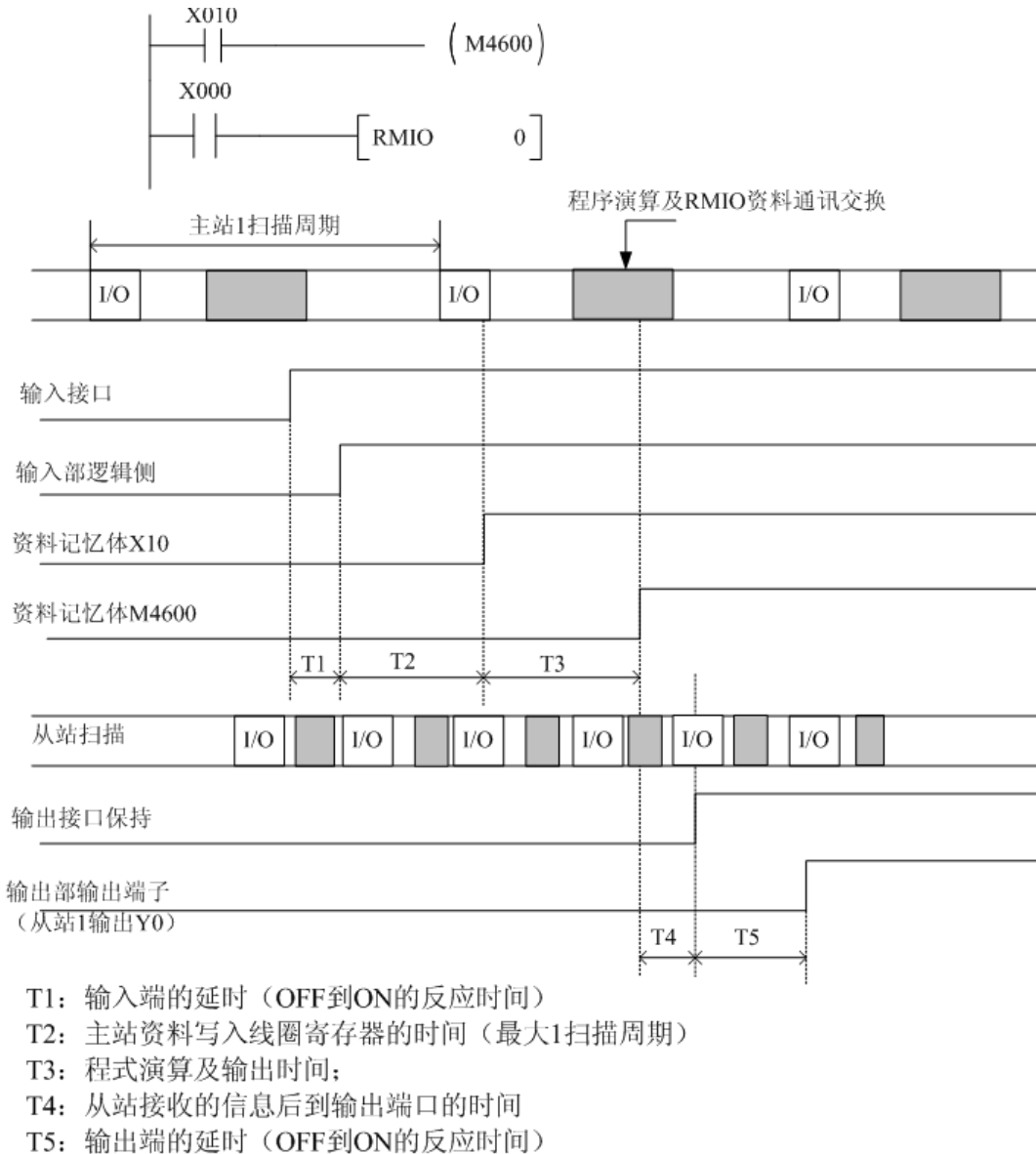
传送所需的时间：远距离 I/O，主站与所有的从站通信所需的时间（通信周期时间，此时间占掉主站 SCAN TIME）如下：

波特率 (bps)	每站间通信 时间 T_n (ms)	超时等待 时间 t (ms)	主站通信所需时间 T (ms)	主站与 4 从站正 常通信时间 (ms)
9600	42	D8380*10	$T_n * n_1 + t * n_2$ (n_1 : 正常从站数; n_2 : 通信超时从站 数)	168
19200	21			84
38400	11			44
57600	7			28
76800	6			24
128000	4			16
153600	3			12
307200	2			8

如有设置的从站通信异常，则需增加重复通信的时间（每次重复通信时间如上表 T_n ）。

通信延迟时间：远距离 I/O 在接收资料时，将发生延迟如下：

6. 应用指令解说



出错代码

当通讯发生错误时，特殊辅助线圈 M8336~M8340 为通讯出错状态指示，错误代码被保存至特殊数据寄存器 (D8338~D8342) 中。

错误	代码含义	错误站点	检查站点	说明	检查点
01H	通讯超时错误	L	M	主站发送请求到从站后，无应答，时间超过通讯超时时间	检查配线、电源 run/stop 状态等。
02H	站号编号错误	L	M	站号编号不符合主站和从站间的关系	检查配线
03H	通信命令错误	L	M	网络通讯命令错误	检查配线，网络设置
04H	通讯格式错误	L	M	从站的通讯格式不正确	检查配线，RMIO 的资源设置；
11H	通讯超时错误	M	L	从站发送应答给主站后，主站不再将请求发送到下一从站	检查配线、电源 run/stop 状态等。

6. 应用指令解说

13H	通信命令错误	M	L	网络通讯命令错误	检查配线，网络设置
14H	通讯格式错误	M	L	主站的通讯格式不正确	检查配线，RMIO 的资源设置
21H	无从站错误	L	L *1	网络里的站号编号不正确	检查站点编号设置
22H	站点编号错误	L	L *1	站点编号不符合主站和从站间的关系	检查配线
23H	通信命令错误	L	L *1	网络通讯命令错误	检查配线，网络设置
24H	通讯格式错误	L	L *1	从站的通讯格式不正确	检查配线，RMIO 的资源设置

M: 主站

L: 从站

*1: 另外一个从站

6. 应用指令解说

F193 DTLK2 资料连线

F		DTLK2		资料连线								S1 ·	S2 ·	K		
193																
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1 ·													*	*		
S2 ·					*	*							*	*		
K					0,1	0,1										

BSP01 采用此模式，他们能链接一个小规模系统中的数据。BSP01 主站最多可以 15 台 BSP01 从站链接通讯。

DTLK2 在 2 个通讯接口同时使用时，只能优先使用先符合使能条件的一条指令。

通讯帧格式及波特率设置由 D8120、D8320 决定（根据串口选择）；

指令符号：

— [DTLK2 S1 · S2 · K]

S1: 数据起始地址，范围为 D0~D7999

S2: 数据长度，范围为 1~40

K: 串口选择，范围为常数 0~1；

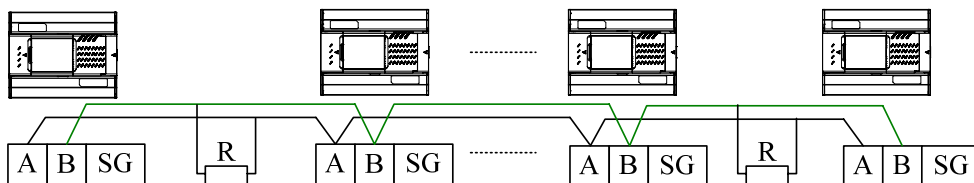
0: 内建 RS485 通讯接口；

1: RS485 或 RS232 扩充卡之通讯接口。

RS485 或 RS232 扩充卡之通讯接口（除 SR 机种皆可选购），RS485 通讯接口（仅 BSP01AR/T 型号之机种内建），以上两个接口可实现资料连线通信功能，但不能同时设定为资料连线功能；

	规格
通信速度	EIA RS-485 标准
传送速度	9600bps~307200bps
连接从站数	最多 15 局
连接区域	D0~D7999，起始地址由 DTLK2 指令设置
每 1 站连线资料长度	最大 40 字信息，由 DTLK2 指令设置
传输线	隔离双绞线，共线连接，2 线式 总长 500m (76800bit/s)，1km (38400bit/s)

配线方法

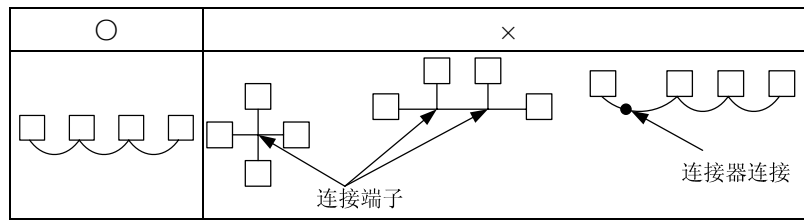


6. 应用指令解说

注 1: SHLD 端子请做第三种接地, 不做第三种接地时, 噪音将造成误动作。

注 2: 通信线的分支请不要超过 3 条线。

注 3: R 为通信终端电阻, 规格为 $120\ \Omega$, $1/4W$ 。



使用的相关标记及数据寄存器

1) 辅助继电器

辅助继电器	特性	名称	描述	回应类型
M8400	只读	主站通讯错误	当主站发生通讯错误时为 ON	从
M8401	只读	从站 1 通讯错误	当从站 1 发生通讯错误时为 ON	主/从
M8402	只读	从站 2 通讯错误	当从站 2 发生通讯错误时为 ON	主/从
...
M8414	只读	从站 14 通讯错误	当从站 14 发生通讯错误时为 ON	主/从
M8415	只读	从站 15 通讯错误	当从站 15 发生通讯错误时为 ON	主/从
M8416	只读	数据通讯状态	DTLK 当与其它站点通讯时为 ON	主/从
M8417	只读	Data Link 模式	扩充通讯接口为 Data Link 功能运行中为 ON	主/从
M8418	只读	Data Link 模式	内建通讯接口为 Data Link 功能运行中为 ON	主/从

2) 资料寄存器

资料寄存器	特性	名称	描述	回应类型
D8173	只读	站点号	存贮自己的站点号	主/从
D8174	只读	从站点总数	存储从站点的总数	主/从
D8175			保留	
D8176	写	站点号设置	设置自己的站点号	主/从
D8177	写	总从站点数设置	设置从站点的总数	主
D8178			保留	
D8179	读/写	重试次数设置	设置重试次数	主
D8180	读/写	通讯超时设置	设置通讯超时 (Time-Out)	主
D8401	只读	当前网络扫描时间	存储当前网络扫描时间	主/从
D8402	只读	最大网络扫描时间	存储最大网络扫描时间	主/从
D8403	只读	主站点的通讯错误计数	主站点的通讯错误次数	从
D8404	只读	从站点 1 的通讯错误次数	从站点 1 的通讯错误次数	主/从
D8405	只读	从站点 2 的通讯错误次数	从站点 2 的通讯错误次数	主/从
...
D8411	只读	从站点 8 的通讯错误次数	从站点 8 的通讯错误次数	主/从

6. 应用指令解说

...
D8417	只读	从站点 14 的通讯错误次数	从站点 14 的通讯错误次数	主/从
D8418	只读	从站点 15 的通讯错误次数	从站点 15 的通讯错误次数	主/从
D8419	只读	主站点的通讯错误代码	主站点的通讯错误代码	从
D8420	只读	从站点 1 的通讯错误代码	从站点 1 的通讯错误代码	主/从
D8421	只读	从站点 2 的通讯错误代码	从站点 2 的通讯错误代码	主/从
...
D8427	只读	从站点 8 的通讯错误代码	从站点 8 的通讯错误代码	主/从
...
D8433	只读	从站点 14 的通讯错误代码	从站点 14 的通讯错误代码	主/从
D8434	只读	从站点 15 的通讯错误代码	从站点 15 的通讯错误代码	主/从

设置

当程序运行或 BSP01 电源打开时，Data Link 的每一个设置都变为有效；

1) 设定站号 (D8176)

设定 0~15 的值到特殊数据寄存器 D8176 中，0 为主站号，1~15 表示为从站 1~15。

2) 设定从站总数 (D8177)

设定 1~15 的值到特殊数据寄存器 D8177 中表示从站总数（默认为 7），从站不需要此设定。

用户应根据需要设置从站总数，可提高数据刷新速度。

3) 设定重试次数 (D8179)

设定 0~10 的值到特殊数据寄存器 D8179 中（默认为 3），从站点不需要此设定；如果一个主站试图以此重试次数与从站通讯，此从站点将发生通讯错误。

4) 设定通讯超时 (D8180)

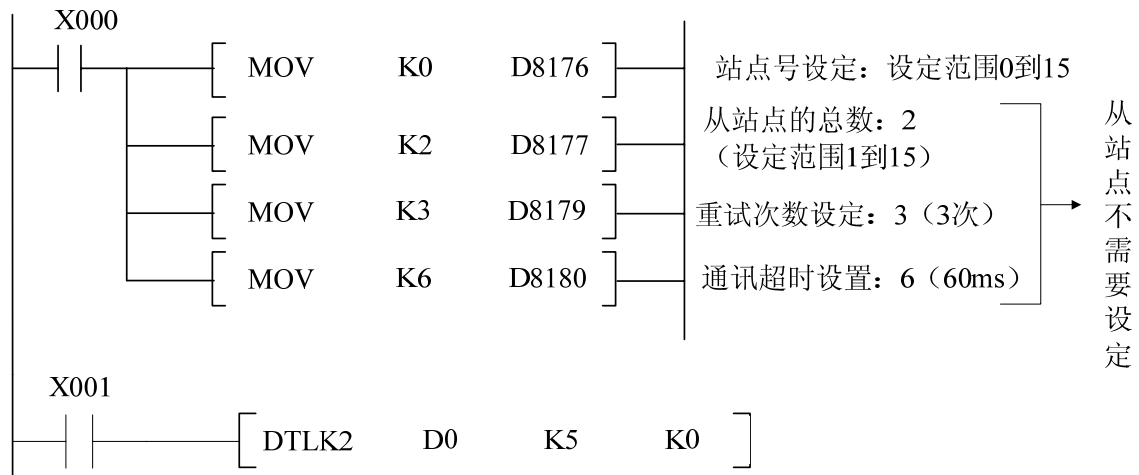
设定 5~255 的值到特殊数据寄存器 D8180 中（默认为 5），此值乘以 10 为通讯超时的等待时间 (ms)。

5) 当前网络扫描时间 (D8401)

当前 Data Link 网络扫描一次所花的时间，此值乘以 10 为扫描时间 (ms)。

6) 网络最大扫描时间 (D8402)

用于进行设置的程式：



6. 应用指令解说

出错代码

当通讯发生错误时，特殊辅助线圈 M8400~M8415 为通讯出错状态指示，错误代码被保存至特殊数据寄存器（D8419~D8434）中。

错误	代码含义	错误站点	检查站点	说明	检查点
01H	通讯超时错误	L	M	主站发送请求到从站后，无应答，时间超过通讯超时时间	检查配线、电源 run/stop 状态等。
02H	站号编号错误	L	M	站号编号不符合主站和从站间的关系	检查配线
03H	通讯计数器错误	L	M	通讯计数器不符合主站和从站间的关系	检查配线
04H	通讯格式错误	L	M, L	从站的通讯格式不正确	检查配线，DTLK 的资源设置；
11H	通讯超时错误	M	L	从站发送应答给主站后，主站不再将请求发送到下一从站	检查配线、电源 run/stop 状态等。
14H	通讯格式错误	M	L	主站的通讯格式不正确	检查配线，DTLK 的资源设置
21H	无从站错误	L	L *1	网络里的站号编号不正确	检查站点编号设置
22H	站点编号错误	L	L *1	站点编号不符合主站和从站间的关系	检查配线
23H	通讯计数器错误	L	L *1	通讯计数器不符合主站和从站间的关系	检查配线
31H	无接收通讯参数错误	L	L *2	从站在通讯参数设置前就从主站接收请求	检查配线、低电源、run/stop 状态
32H	其他错误	L	L *1	网络通讯命令错误	检查网络设置

M: 主站

L: 从站

*1: 另外一个从站

*2: 独立站点

F224~249 触点比较指令

触点比较指令

功能编号	助记符 名称	页码
224	LD (S1)=(S2)	1
225	LD (S1)>(S2)	1
226	LD (S1)<(S2)	1
228	LD (S1)≠(S2)	1
229	LD (S1)≧(S2)	1
230	LD (S1)≦(S2)	1
232	AND (S1)=(S2)	2
233	AND (S1)>(S2)	2
234	AND (S1)<(S2)	2
236	AND (S1)≠(S2)	2
237	AND (S1)≧(S2)	2
238	AND (S1)≦(S2)	2
240	OR (S1)=(S2)	3
241	OR (S1)>(S2)	3
242	OR (S1)<(S2)	3
244	OR (S1)≠(S2)	3
245	OR (S1)≧(S2)	3
246	OR (S1)≦(S2)	3

F224~230 接点状态比较 LD

F		LD		接点状态比较 LD	S1	S2										
224~230	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*	*

S1: 比较值 1

S2: 比较值 2

S1 与 S2 的内容 作比较的指令，比较结果为等于时，该指令导通。不等于时，该指令不导通。

LD 的指令可直接与母线连接使用。

F No	16 位元指令	32 位元指令	导通条件	非导通条件
224	LD=	D LD=	S1=S2	S1≠S2
225	LD>	D LD>	S1>S2	S1≤S2
226	LD<	D LD<	S1<S2	S1≥S2
228	LD<>	D LD<>	S1≠S2	S1=S2
229	LD≤	D LD≤	S1≤S2	S1>S2
230	LD≥	D LD≥	S1≥S2	S1<S2

S1 与 S2 的最左位（16 位指令：b15、32 位指令：b31）为 1 时，该比较值被视为负值来比较。

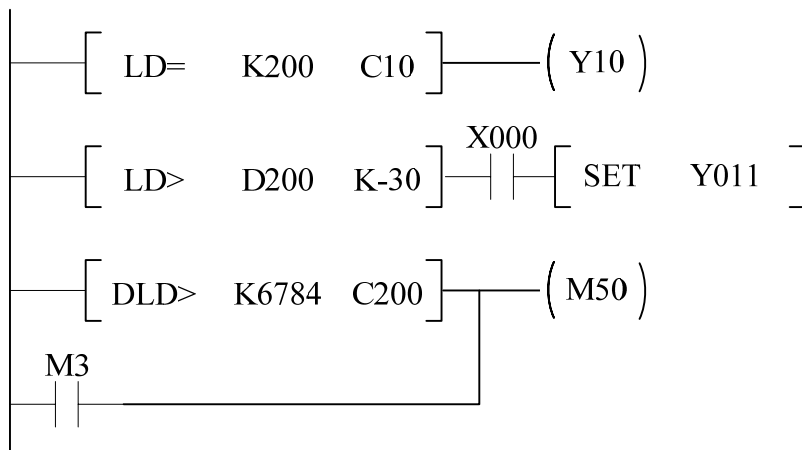
32 位长度计数器(C200~)代入本指令作比较时，一定要使用32 位指令(DLD※)，若是使用16 位指令(LD※)时，CPU 判定为”程序错误”、主机面板上ERROR红色指示灯闪烁，CPU 不能 RUN。

程序范例

C10 的内容等于K200 时，Y10= 0n。

当D200 的内容大于-30，而且X0= 0n 的时候Y11= 0n 并保持住。

C200 的内容小于6,784 或者是M3= 0n 的时候M50= 0n。



F232~238 接点状态比较 AND

F		AND		接点状态比较 AND	S1	S2										
232~238	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*	*

S1: 比较值 1

S2: 比较值 2

S1 与 S2 的内容 作比较的指令，比较结果为等于时，该指令导通。不等于时，该指令不导通。LD 的指令是与接点串联的比较指令。

F No	16 位元指令	32 位元指令	导通条件	非导通条件
232	AND=	D AND=	S1=S2	S1≠S2
233	AND>	D AND>	S1>S2	S1≤S2
234	AND<	D AND<	S1<S2	S1≥S2
236	AND≠	D AND≠	S1≠S2	S1=S2
237	AND≤	D AND≤	S1≤S2	S1>S2
238	AND≥	D AND≥	S1≥S2	S1<S2

S1 与 S2 的最左位 (16 位指令: b15、32 位指令: b31) 为 1 时，该比较值被视为负值来比较。

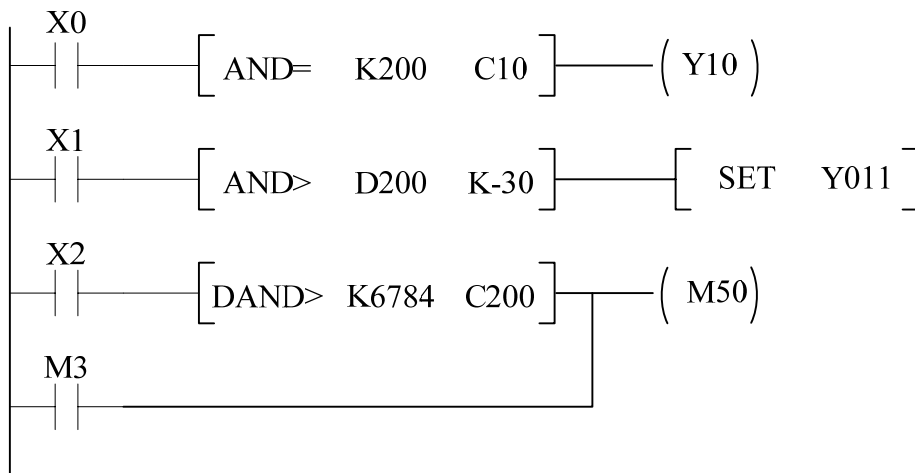
32 位长度计数器 (C200~) 代入本指令作比较时，一定要使用 32 位指令 (DLD ※)，若是使用 16 位指令 (AND※) 时，CPU 判定为“程序错误”、主机面板上 ERROR 红色指示灯闪烁，CPU 不能 RUN。

程序范例

当 X0= 0n 时，C10 的现在值又等于 K200 时，Y10= 0n。

当 X1= 0n 而寄存器 D200 的内容又大于 -30 的时候，Y11= 0n 并保持住。

当 X2= 0n 且 C200 的内容小于 6,784 的时候或者 M3= 0n 时，M50= 0n



F240~246 接点状态比较 OR

F		OR		接点状态比较 OR	S1 S2											
240~246	D		P													
	位元件				字元件											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	W	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*	*

S1: 比较值 1

S2: 比较值 2

S1 与 S2 的内容 作比较的指令，比较结果为等于时，该指令导通。不等于时，该指令不导通。OR 的指令是与接点并联的比较指令。

F No	16 位元指令	32 位元指令	导通条件	非导通条件
240	OR=	D OR=	S1=S2	S1≠S2
241	OR>	D OR>	S1>S2	S1≤S2
242	OR<	D OR<	S1<S2	S1≥S2
244	OR<>	D OR<>	S1≠S2	S1=S2
245	OR≤	D OR≤	S1≤S2	S1>S2
246	OR≥	D OR≥	S1≥S2	S1<S2

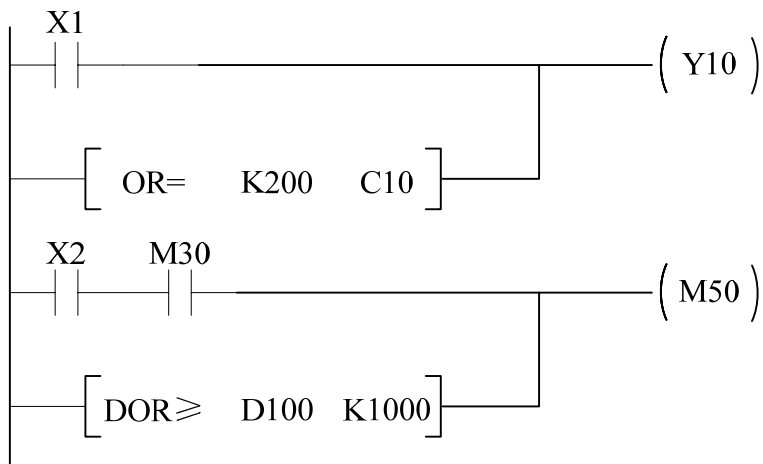
S1 与 S2 的最左位（16 位指令：b15、32 位指令：b31）为 1 时，该比较值被视为负值来比较。

32 位长度计数器(C200~)代入本指令作比较时，一定要使用32 位指令(DOR※)，若是使用16 位指令(OR※)时，CPU 判定为”程序错误”、主机面板上ERROR 红色指示灯闪烁，CPU 不能RUN。

程序范例

当X1= 0n 时，或者是C10的现在值等于K200 时，Y10= 0n。

当X2及M30都等于0n的时候或者是32位缓存器D101、D100的内容大于或等于K1,000 时，M50= 0n。

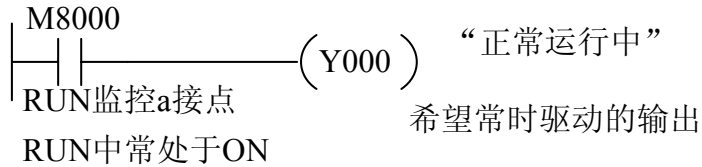


第七章 基本功能的补充说明

1 特殊元件的补充说明

可编程控制器的运行标志

利用显示可编程控制器运行状态的 RUN 监视 M8000, M8001, 可作为指令的驱动条件以及正常运行中的显示。



M8001在RUN中常时处于OFF状态

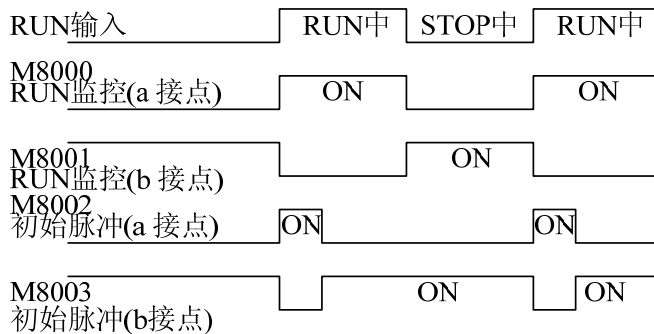
初始脉冲 M8002 在可编程控制器的运行开始后, 仅在一个运算周期为 ON, 其余时间均保持 OFF 状态。

该脉冲在程序初始化和写入规定值等情况时作为程序初始设定信号使用。



M8003 仅在 RUN 后一个运算周期为 OFF, 其余时间均保持 ON 状态。

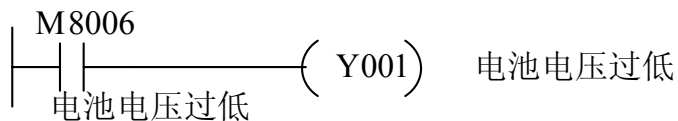
标志动作时序



电池电压过低检测

电池电压过低检测和外部输出。

此软元件用于检测存储器后备的锂电池电压过低的特殊软元件。当可编程控制器检测出电池电压过低时, BATT 错误指示灯点亮, 希望向外部通报时可以使用顺序控制程序。

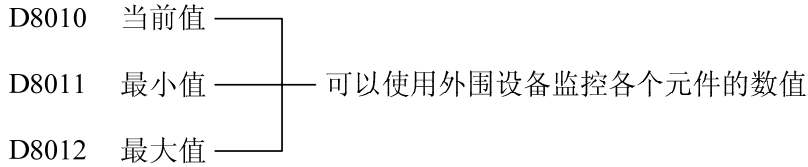


M8007 用于电池电压过低的状态锁存

7. 基本功能的补充说明

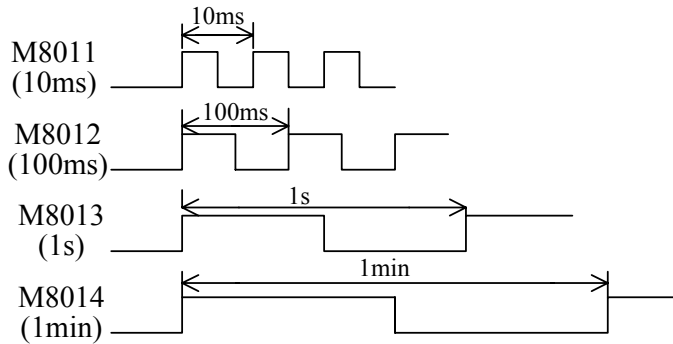
运算时间（扫描时间的监控）

可编程控制器的运算时间保存于 D8010~D8012 中。



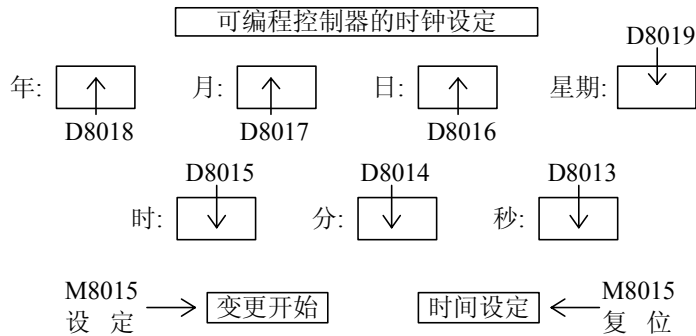
在这些数值中包括后文所述的恒定扫描时间的等待时间。

本机提供了以下四种内部时钟，当可编程控制器通电后，该四种时间会保持振荡。



注即便可编程控制器处于 STOP 状态，时钟仍然保持运作。因此，RUN 监控（M8000）的上升沿和时钟的开始时间不同步。

实时时钟



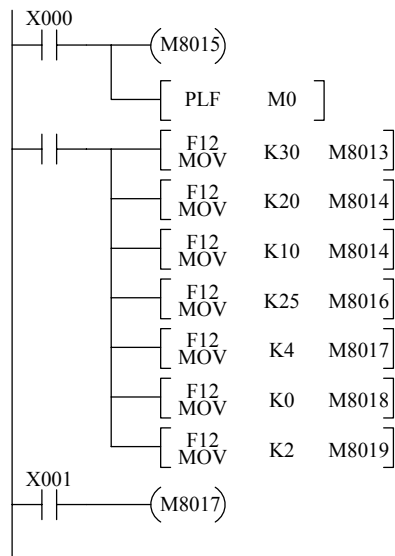
使用编程设备的情况。

强制置位

使用数据寄存器的当前值更改功能，向各个数据寄存器中输入希望校准的时间数值，当到达设定时间时，强制复位。

7. 基本功能的补充说明

通过普通程序执行的方法



时钟校准时的注意事项。

当 M8015 处于 OFF 时，无法进行时间寄存器的变更。

请在 M8015 处理 ON 时输入新时间。

设定时间输入时，请预先设定比当前时间早数分钟的时间

当到达设定时间时，M8015 由 ON——OFF，状态转变，新时间生效开始计时。

当输入不存在的日期时，无法进行时间变更，请再次输入正确的时间数值。

D8018 的值为 2000~2099 表示 2000~2099 年。

清除指令

M8031 非保持存储器全部清除，M8032 保持存储器全部清除

元件地址号	清除元件
M8031 非保持区域	输入继电器 X，输出继电器 Y，普通辅助继电器 M。普通状态 S 的接点影像 定时器 T 的接点，计时线圈 普通计数器接点，计数线圈，复位线圈 普通数据寄存器 D 的当前值寄存器 定时器 T 的当前值寄存器 普通计数器 C 的当前值寄存器
M8032 保持区域	保持用辅助继电器 M，保持用状态 S 的接点影像 累计定时器 T 的接点與线圈 保持用计数器和高速计数器的接点、计数线圈、复位线圈 保持用数据寄存器 D 的当交电值寄存器 累计和 1ms 用定时器 T 的当前值寄存器 保持用计数器和高速计数器的当前值寄存器

7. 基本功能的补充说明

存储器保持停止

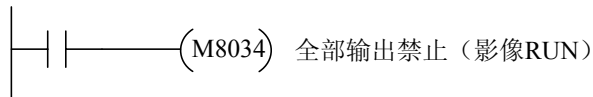
STOP 中的输出保持

预先驱动特殊辅助继电器 M8033，即使可编程控制器 RUN——STOP 后，仍然可以保持运行中的输出状态。

例如希望用可编程控制器驱动加热的情况时，在驱动加热器和其它设备停止可编程控制器，执行程序更改后能够再次运行。

全部输出禁止的指令

通过驱动 M8034 清除输出锁存存储器，可以使所有输出继电器变为 OFF，可编程控制器仍在影像存储器上运行

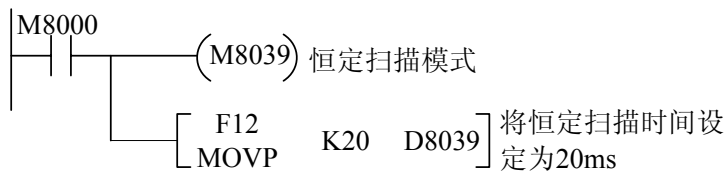


恒定扫描模式

运算处理时间的固定化

驱动辅助继电器 M8039，将目标扫描时间以 1ms 为单位预先写入数据完成寄存器 D8039 中，则可编程控制器的运算周期不会低于该值。

即使运算提早结束时，在剩余时间内进行等待，随后返回 0 步。



当使用 FNC67(RAMP)，FNC71(HKY)，FNC74(SEGL)，FNC75(ARWS)，FNC77(PR)等扫描同步执行指令时，建议使用该恒定扫描模式或利用定时器中断的定时进行驱动。

特别是使用 FNC71(HKY)指令时，按键输入的滤波会导致应答延迟，请务必将扫描时间设置在 20ms 以上使用。

注：在 D8010~8012 的扫描时间显示中包含了利用恒定扫描模式的指定时间。

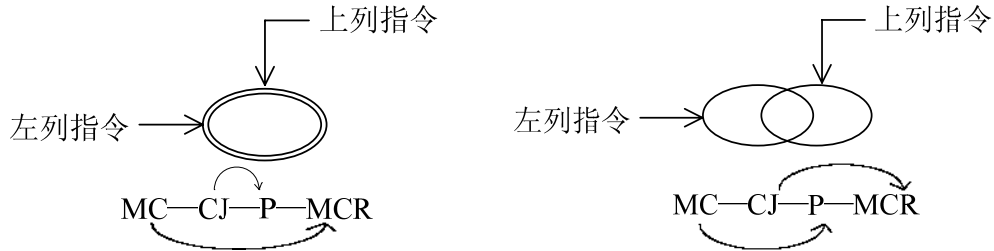
状态间的转移禁止

驱动 M8040 后，即使具备了转移条件也无法进行状态转移，而且停止状态内的输出将继续其动作，关于状态内的输出复位请参考步进指令说明

2 程序流程控制指令的相互关系

关于 MC-MCR 指令和 CJ 指令的相互关系，已在 F00CJ 指令说明中详细说明过，其它各种程序流程控制指令之间的想到关系如下所示：

下表中 \circ 表示包含关系， \odot 表示前后区间重复



	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET
MC-MCR	\circ	\odot 例 1	\odot	\odot	\odot
	$\odot\Delta$	$\odot\Delta$ 例 2	$\odot\Delta$	$\odot \times(6607)$	$\odot \times(6605)$
CJ-P	\circ	\odot	\odot	\odot	\odot
	$\odot\Delta$	$\odot\Delta$	$\odot\Delta$	$\odot\Delta$	$\odot\Delta$
EI-DI	\circ	\odot	\odot	\odot	\odot
	$\odot\Delta$	$\odot\Delta$	$\odot\Delta$	$\odot\Delta$	$\odot\Delta$
FOR-NEXT	$\odot \times(6607)$	\odot	\odot	\odot 16 层	$\odot \times(6607)$
	$\odot \times(6607)$	$\odot\Delta$	$\odot\Delta$	$\odot\Delta *2$	$\odot \times(6607)$
STL-RET	$\odot \times(6605)$	\odot	\odot	\odot 1 个 STL 内	\odot
	$\odot \times(6605)$	$\odot\Delta$	$\odot\Delta$	$\odot \times(6607)$	$\odot\Delta$
P-SRET	$\odot \times(6606)$	\odot	\odot	\odot	$\odot \times(6606)$
	$\odot \times(6608)$	$\odot\Delta$	$\odot\Delta$	$\odot \times(6607)$	$\odot \times(6605)$
I-IRET	$\odot \times(6606)$	\odot	\odot	\odot	$\odot \times(6606)$
	$\odot \times(6606)$	$\odot\Delta$	$\odot\Delta$	$\odot \times(6607)$	$\odot \times(6606)$
FEND-END	\odot	\odot	\odot	\odot	\odot
	$\odot \times(6608)$	$\odot\Delta \times(6701)$	$\odot\Delta$	$\odot \times(6607)$	$\odot \times(6605)$
O-FEND	\odot	\odot	\odot	\odot	\odot
	$\odot\Delta \times(6608)$	$\odot\Delta$	$\odot\Delta$	$\odot \times(6607)$	$\odot \times(6605)$
O-END (无 FEND)	\odot	\odot	\odot	\odot	\odot
	$\odot \times(6608)$	$\odot \times(6701)$	$\odot\Delta$	$\odot \times(6607)$	$\odot \times(6605)$

\circ ：能够没有问题的相互组合使用。

\times ：禁止使用的组合，() 中为错误序号

Δ ：虽然不属于严禁使用的范围，但是可能会导致动作的复杂化，建议尽量避免使用。

7. 基本功能的补充说明

P-SRET	I-IRET	FEND-END	备注
○ ×(6608)	○ ×(6608)	○ ×(6608)	1.无异常显示。忘记 DI 的状态。 2.R FOR NEXT NEXT。如实线般动作
⊗ ×(6606)	⊗ ×(6606)	⊗ ×(6608)	
○ △	○ △	○ △	
⊗ △	⊗ △	⊗ △	3.头的 FEND 和 END 有效。并非所欲制作及执行的程序。无异常显示。
○ ○	○ ○	○ ○	
⊗ ○	⊗ ○	⊗ ○	除一部份指令例外。具有包含关系的指令可以进行组合使用，但是必须注意下列特例情况： 1.MC-MCR 不可在 FOR~NEXT，STL-RET，P-SRET，I-IRET 等指令中使用。 2.STL-RET 不可在 FOR~NEXT，P-SRET，I-IRET 等指令中使用。 3.MC-MCR，FOR-NEXT，P-SRET，I-IRET 的指令中不可使用 I，IRET，SRET，FEND，END 等指令。
○ ×(6607)	○ ×(6607)	○ ×(6607)	
⊗ ×(6701)	⊗ ×(6607)	⊗ ×(6607)	
○ ×(6605)	○ ×(6605)	○ ×(6605)	
⊗ ×(6606)	⊗ ×(6606)	⊗ ×(6605)	
○ ×(6606)	○ ×(6606)	○ ×(6709)	
⊗ ×(6606)	⊗ ×(6606)	⊗ ×(6709)	
○ ×(6606)	○ ×(6606)	○ ×(6606)	
⊗ ×(6606)	⊗ ×(6606)	⊗ ×(6606)	
○ ○	○ ○	○	
⊗ ×(6709)	⊗ ×(6709)	⊗	
○ ×(6606)	○ ×(6606)	○	
⊗ ×(6709)	⊗ ×(6606)	⊗	
○ ×(6606)	○ ×(6606)	○	
⊗ ×(6709)	⊗ ×(6706)	⊗	

3 ASC II 文字排列

<ASC II 码表(7 位码、16 进制表现)>

16 进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		DLE	SP	0	@	P	,	p.								
1	SOH	DC1	!	1	A	Q	a.	q.								
2	STX	DC2	"	2	B	R	b.	r.								
3	ETX	DC3	#	3	C	S	c.	s.								
4	EOT	DC4	\$	4	D	T	d.	t.								
5	ENO	NAK	%	5	E	U	e.	u.								
6	ACK	SYN	&	6	F	V	f.	v.								
7	BEL	ETB	'	7	G	W	g.	w.								
8	BS	CAN	(8	H	X	h.	x.								
9	HT	EM)	9	I	Y	i.	y.								
A	LF	SUB	*	:	J	Z	j.	z.								
B	VT	ESC	+	;	K	[k.	{								
C	FF	FS	,	<	L	\	l.									
D	CR	GS	-	=	M]	m.	}								
E	SO	RS	□	>	N	^	n.	~								
F	SI	US	/	?	O	_	o.	DEL								

<ASC II 码的例>

10 进数	ASC II (16 进数)
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39

英文字母	ASC II (16 进数)	英文字母	ASC II (16 进数)
A	41	N	4E
B	42	O	4F
C	43	P	50
D	44	Q	51
E	45	R	52
F	46	S	53
G	47	T	54
H	48	U	55
I	49	V	56
J	4A	W	57
K	4B	K	58
L	4C	Y	59
M	4D	Z	5A

编码	ASC II (16 进数)
STX	02
ETX	03

4 错误代码一览表

M 寄存器		D 寄存器			是否继续运行
号码	功能	号码	错误代码说明		
8060	HMI 口构成错误	8060	200x 扩充卡不匹配 6006: 无扩充通讯卡		Y
8061	PC 硬件检查	8061	错误代码	0000: 无异常	Y
				6101: RAM 错 6102: ROM 错 6103: 本体 I/O 总线错 6104: 用户程序错误 6105: 监视定时器溢出 6106: RAM 地址线错	N
8063	通信检测	8063	错误代码	0000: 无异常	Y
				6301: DTLK 错误 6302: RMIO 错误	N
8064	参数错误	8064	错误代码	0000: 无异常	Y
				6401: 程序和参数不一致 6402: 存储器容量设定不良 6403: 保持区域设定不良 6409: 其它不良	N
8065	语法错误	8065	错误代码	0000: 无异常	Y
				6501: 指令地址组合不良 6504: 标号重复 6505: 元件地址号范围溢出 6506: 使用未定义指令 6507: 标号 (P) 定义不良 6508: 中断标号 (I) 定义不良 6509: 其它	N
8066	回路错误	8066	错误代码	0000: 无异常	Y

7. 基本功能的补充说明

				<p>6603: MPS 连续使用 8 次以上 6604: MPS MRD MPP 关系错误 6605: STL 连续使用 10 次以上 6606: 调用不存在的标号 6607: 在主程序中有 I, SRET 6609: CALL 嵌套超过 16 次 6610: for next 不匹配 6611: 有 JCS, 无 JCR 6612: 有 STL, 无 RET 6613: 有 MC, 无 MCR 6614: 有 SMCS, 无 SMCR 6615: I, IRET 关系不正确 6616: 在 STL_RET 中有 MC, MCR, SMCS, SMCR, SRET; 6617: FOR NEXT 嵌套超过 16 次 6618: 无 END 6621: 其它错误。</p>	N
8067	运算错误	8067	错误代码	<p>6705: 地址矛盾 6706: 参数错 6730: PID 采样时间 TS 在对象范围以外 (Ts<0) 6732: PID 输入滤波常数在对象范围以外 6733: PID 比例增益在对象范围以外 6734: PID 积分时间在对象范围以外 6735: PID 微分增益在对象范围以外 6736: PID 微分时间在对象范围以外 6740: PID 采样时间 ≤ 运算周期 6742: PID 测定值变化量溢出 6743: PID 偏差值溢出 6744: PID 积分计算值溢出 6745: PID 微分增益溢出导致微分值溢出 6746: PID 微分计算值溢出 6747: PID 运算结果溢出 6750: PID 自动调谐结果不良 6751: PID 自动调谐动作方向不一致 6752: PID 自动调谐动作不良</p>	Y
8069	扩展 I/O 错			<p>6903: 扩展 I/O 错。 6904: AD/DA 设置错误 6905: AD 无电源</p>	Y