H2U 系列可编程控制器
指令及编程手册

V0.88

深圳市汇川控制技术有限公司

前 言

H2U 系列通用 PLC 是深圳市汇川控制技术有限公司研发的高性比控制产品，指令丰富，高速信号处理能力强，运算速度快，允许的用户程序容量可达 24K 步，且不需外扩存储设备。

控制器配备了两个通讯硬件端口，方便现场接线；通讯端口支持多种通讯协议，包括 MODBUS 主站、从站协议，尤其方便了与变频器等设备的联机控制；

提供了严密的用户程序保密功能，子程序单独加密功能，方便用户特有控制工艺的知识产权保护。

控制器提供了多种编程语言，用户可选用梯形图、指令表、步进梯形图、SFC 顺序功能图等鞭策方法。指令系统为广大工程技术人员所熟悉，而本公司提供的 AutoShop 编程环境，更是融合了众多 PLC 编程环境的优点，丰富的在线帮助信息，使得编程时无需查找说明资料，方便易用。

对高速信号的处理部分，MTQ 版本提供了六路高速脉冲输入、五路高速脉冲输出功能，处理能力增强。

我们热忱欢迎读者以多种形式咨询和交流使用方法,反馈的手册中的错误和遗漏。

公司网页：WWW.INOVANCE.CN

信息交流：HCCONTROL@163.COM

目 录

手册版本信息

版本	发布时间	修订说明
V0.8	2009-03-18	首次发布
V0.88	2009-04-18	修订系统变量说明 增加 PLC 原理和基本知识介绍 增加 MTQ 版本的使用说明

指令详解索引

简单逻辑指令

应用指令（以 FNC NO 为序）

分类	FNC NO.	指令助记符	功能	页码
循环移位	30	ROR	循环右移	84
	31	ROL	循环左移	84
	32	RCR	带进位循环右移	85
	33	RCL	带进位循环左移	85
	34	SFTR	位右移	85
	35	SFTL	位左移	86
	36	WSFR	字右移	87
	37	WSFL	字左移	87
	38	SFWR	移位写入	88
	39	SFRD	移位读出	88
数据处理	40	ZRST	批次复位	89
	41	DECO	译码	89
	42	ENCO	编码	90
	43	SUM	ON 位数	91
	44	BON	ON 位数判定	91
	45	MEAN	平均值	92
	46	ANS	信号报警位置	92
	47	ANR	信号报警器复位	93
	48	SQR	BIN 开方	93
	49	FLT	整数→浮点数转换	94
高速处理	50	REF	输入输出刷新	94
	51	REFF	滤波器调整	95
	52	MTR	矩阵输入	96
	53	HSCS	比较置位（高速计数器）	97
	54	HSCR	比较复位（高速计数器）	98
	55	HSZ	比较区间（高速计数器）	99
	56	SPD	脉冲密度	104
	57	PLSY	脉冲输出	105
	58	PWM	脉冲调制	107
	59	PLSR	带加减速的脉冲输出	107

分类	FNC NO.	指令助记符	功能	页码
方便指令	60	IST	---	
	61	SER	数据查找	109
	62	ABSD	凸轮控制(绝对方式)	110
	63	INCD	凸轮控制(增量方式)	112
	64	TTMR	示教定时器	113
	65	STMR	特殊定时器	114
	66	ALT	交替输出	115
	67	RAMP	斜波信号	115
	68	RTOC	旋转工作台控制	116
	69	SORT	数据排列	118
外围设备 I/O	70	TKY	数字键输入	119
	71	HKY	16 键输入	120
	72	DSW	数字式开关	122
	73	SEGD	7 段码译码	123
	74	SEGL	7 段码扫描显示	124
	75	ARWS	方向开关	125
	76	ASC	ASCII 码转换	127
	77	PR	ASCII 码打印输出	127
	78	FROM	BFM 读出	128
	79	TO	BFM 写入	129

分类	FNC NO.	指令助记符	功能	页码
外设设备	80	RS	串行数据传送	130
	81	PRUN	8 进制位传送	136
	82	ASCI	HEX-ASCII 转换	137
	83	HEX	ASCII-HEX 转换	138
	84	CCD	校验码	139
	88	PID	PIC 运算	141
浮点数	110	ECMP	2 进制浮点数比较	145
	111	EZCP	2 进制浮点数区间比较	145
	118	EBCD	2 进制-10 进制浮点数转换	145
	119	EBIN	10 进制-2 进制浮点数转换	147
	120	EADD	2 进制浮点数加法	147
	121	ESUB	2 进制浮点数减法	148
	122	EMUL	2 进制浮点数乘法	149
	123	EDIV	2 进制浮点数除法	149
	127	ESQR	2 进制浮点数开方	150
	129	INT	2 进制浮点数-BIN 整数转换	151
	130	SIN	浮点数 SIN 运算	151
	131	COS	浮点数 COS 运算	152
	132	TAN	浮点数 TAN 运算	153
	147	SWAP	上下字节变换	153
定位	155	ABS	ABS 位置数读取	154
	156	ZRN	原点回归	155
	157	PLSV	可变度脉冲输出	157
	158	DRVI	相对定位	158
	159	DRVA	绝对定位	161

分类	FNC NO.	指令助记符	功能	页码
时钟连算	160	TCMP	时钟数据的比较	163
	161	TZCP	时钟数据区域比较	164
	162	TADD	时钟数据加法	165
	163	TSUB	时钟数据减法	166
	166	TRD	时钟数据读出	167
	167	TWR	时钟数据写入	167
	169	HOUR	计时器	168
外围设备	170	GRY	格雷码变换	169
	171	GBIN	格雷码逆变换	170
	176			
	177			
接点比较	224	LD=	(S1)=(S2)	171
	225	LD>	(S1)>(S2)	171
	226	LD<	(S1)<(S2)	171
	228	LD<>	(S1)<>(S2)	171
	229	LD<=	(S1)<=(S2)	171
	230	LD>=	(S1)>=(S2)	171
	232	AND=	(S1)=(S2)	172
	233	AND >	(S1)>(S2)	172
	234	AND <	(S1)<(S2)	172
	236	AND <>	(S1)<>(S2)	172
	237	AND <=	(S1)<=(S2)	172
	238	AND >=	(S1)>=(S2)	172
	240	OR=	(S1)=(S2)	173
	241	OR >	(S1)>(S2)	173
	242	OR <	(S1)<(S2)	173
	244	OR <>	(S1)<>(S2)	173
	245	OR <=	(S1)<=(S2)	173
	246	OR >=	(S1)>=(S2)	173

应用指令（以指令助记符为序，未含简单逻辑指令）

分类	指令助记符	FNC NO.	功能	页码
A	ABS	155	ABS 现在值读出	154
	ABSD	62	凸轮控制(绝对方式)	110
	ADD	20	BIN 加法	79
	ALT	66	交替输出	115
	AND=	232	(S1)=(S2)	172
	AND>	233	(S1)>(S2)	172
	AND<	234	(S1)<(S2)	172
	AND<>	236	(S1)<>(S2)	172
	AND<=	237	(S1)<=(S2)	172
	AND>=	238	(S1)>=(S2)	172
	ANR	47	信号报警复位	93
	ANS	46	信号报警置位	92
	ARWS	75	箭形开关	125
	ASC	76	ASCII 码转换	127
	ASCI	82	HEX → ASCII 转换	137
B	BCD	18	BCD 转换	79
	BIN	19	BIN 转换	79
	BMOV	15	成批转换	77
	BON	44	ON 位数判定	91
C	CALL	01	子程序调用	70
	CCD	84	检验码	139
	CJ	00	条件跳转	69
	CML	14	取反传送	77
	CMP	10	比较	74
	COS	131	浮点数 COS 运算	152
D	DEC	25	BIN 减 1	82
	DECO	41	译码	89
	DI	05	中断禁止	71
	DIV	23	BIN 除法	81
	DRVA	159	绝对定位	161
	DRVI	158	相对定位	158
	DSW	72	数字式开关	122

分类	指令助记符	FNC NO.	功能	页码
E	EADD	120	2 进制浮点数加法	147
	EBCD	118	2 进制-10 进制浮点数转换	145
	EBIN	119	10 进制-2 进制浮点数转换	147
	ECMP	110	2 进制浮点数比较	145
	EDIV	123	2 进制浮点数除法	149
	EI	04	中断许可	71
	EMUL	122	2 进制浮点数乘法	149
	ENCO	42	编码	90
	ESOR	127	2 进制浮点数开方	150
	ESUB	121	2 进制浮点数减法	148
	EZCP	111	2 进制浮点数区间比较	145
F	FEND	06	主程序结束	72
	FLT	49	BIN 整数→2 进制浮点数转换	94
	FMOV	16	多点传送	78
	FOR	08	循环范围开始	73
	FROM	78	BFM 读出	128
G	GBIN	171	格雷码逆变换	170
	GRY	170	格雷码变换	169
H	HEX	83	ASCII-HEX 转换	138
	HKY	71	16 键输入	120
	HOUR	169	计时仪	168
	HSCR	54	比较复位（高速计数器）	98
	INCD	63	比较置位（高速计数器）	112
	HSZ	55	区间比较（高速计数器）	99

分类	指令助记符	FNC NO.	功能	页码
I	INC	24	BIN 加 1	82
	INCD	63	凸轮控制(增量方式)	112
	INT	129	浮点数-整数转换	151
	IRET	03	中断返回	71
L	LD=	224	(S1)=(S2)	171
	LD>	225	(S1)>(S2)	171
	LD<	226	(S1)<(S2)	171
	LD<>	228	(S1)<>(S2)	171
	LD<=	229	(S1)<=(S2)	171
	LD>=	230	(S1)>=(S2)	171
M	MEAN	45	平均值	92
	MOV	12	传送	75
	MTR	52	矩阵输入	96
	MUL	22	BIN 乘法	81
N	NEG	29	求补码	83
	NEXT	09	循环范围终了	72
O	OR=	240	(S1)=(S2)	172
	OR >	241	(S1)>(S2)	172
	OR <	242	(S1)<(S2)	172
	OR <>	244	(S1)<>(S2)	172
	OR <=	245	(S1)<=(S2)	172
	OR >=	248	(S1)>=(S2)	172
P	PID	88	PID 运算	141
	PLSV	157	可变速脉冲输出	157
	PLSY	57	脉冲输出	105
	PLSR	59	有加减速脉冲输出	107
	PR	77	ASCII 键打印输出	127
	PRUN	81	8 进制输送	136
	PWM	58	脉冲幅度调整	107
R	RAMP	67	斜坡信号	115
	RCL	33	带进位的循环左移	85
	RCR	32	带进位的循环右移	85
	REF	50	输入输出刷新	94
	REFF	51	滤波器调整	95
	ROL	31	循环左移	84
	ROR	30	循环右移	84
	ROTC	68	旋转工作台控制	116
	RS	80	串行数据输送	130

分类	指令助记符	FNC NO.	功能	页码
S	SEGD	73	7 段码译码	123
	SEGL	74	7 点码按时间分割显示	124
	SER	61	数据查找	109
	SFRD	39	位移读出	88
	SFTL	35	位左移	86
	SFTR	34	位右移	85
	SFWR	38	移位写入	88
	SIN	130	浮点数SIN运算	151
	SMOV	13	移位传送	75
	SORT	69	数据排列	118
	SPD	56	脉冲密度	104
	SQR	48	BIN 开方	93
	SRET	02	子程序返回	70
	STMR	65	特殊定时器	114
	SUB	21	BIN 减法	80
	SUM	43	ON 位数	91
	SWAP	147	上下字节交换	153
T	TADD	162	时钟数据加法	165
	TAN	132	浮点数TAN运算	153
	TCMP	160	时钟数据比较	163
	TKY	70	数字键输入	119
	TO	79	BFM 写入	129
	TRD	166	时钟数据读出	167
	TSUB	163	时钟数据减法	166
	TTMR	64	示教定时器	113
	TWR	167	时钟数据写入	167
	TZCP	161	时钟数据区间比较	164

分类	指令助记符	FNC NO.	功能	页码
W	WDT	07	监控定时器	72
	WOR	27	逻辑字或	83
	WSFL	37	字左移	87
	WSFR	36	字右移	87
	WXCR	28	逻辑字异或	83
X	XCH	17	交换	78
Z	ZCP	11	区间比较	74
	ZRN	156	原点回归	155
	ZRST	40	原点回归批次复位	89

手册快查引导

您若有如下疑问，可参照指引：

序号	希望查阅的内容	请参阅页面
1	简单逻辑指令的解释	查阅 P55～P60 指令详解
2	应用指令的解释	先根据指令名查阅 P10～P12，再详查指令详解
3	计时器的选用	查阅 3.5 节
4	高速计数器的选用	查阅 3.6 节
5	高速输出指令的使用	查阅指令 HSCS/HSCR/HSZ/PLSY/PLSV/PWM
6	输入中断的使用与设置	查阅 3.8 节
7	定时中断的使用与设置	查阅 3.8 节
8	高速计数中断的使用与设置	查阅 3.8 节
9	STL/SFC 的编程方法	查阅 5.2 节
10	通讯格式的设置方法	查阅 RS 指令详解（P130～P136）
11	各种通讯协议的设置方法	查阅 RS 指令详解（P130～P136）
12	如何使用 MODBUS 主站指令	查阅 RS 指令详解（P130～P136）
13	如何用 MODBUS 访问 H2U	MODBUS 内置从机协议定义
14	如何使用 H2U-3A-BD 扩展小板	查阅 8.5 附录
15	MTQ 的高速功能与使用	参见 8.6 附录，查阅高速指令
16	脉冲捕捉功能	参见中断 P 说明、M8170~M8175 变量说明
17		

（本页特意留空）

1. 梯形图及梯形图程序

1.1 梯形图的编程特点：

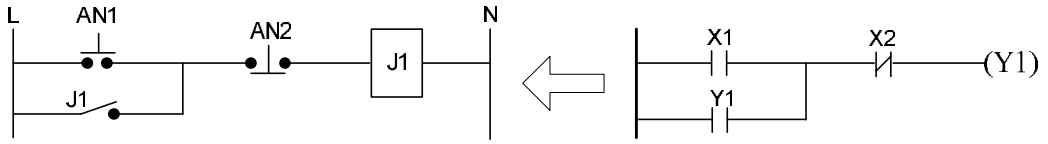
PLC 中梯形图编程方法是仿照继电器控制系统的电气原理设计的一种设计方法，设计中使用的元件如按钮 X、中间继电器 M、时间继电器 T、计数器 C、触点等，都和实际的电气元件的特性相似。

梯形图中常用“触点”和“线圈”元件，触点元件有“常开型”和“常闭型”，分别对应电工术语中的“A 接点”和“B 接点”，同一个继电器的“触点”可被无限次使用，或者认为一个继电器（无论中间继电器 M、时间继电器 T、计数器 C）元件具有无限个“A 接点”和“B 接点”。

对于时间继电器、计数器，具有线圈（信号触发端）和触点，部分元件还具有掉电保持特性，选择合适序号的元件，以得到所需特性的元件。

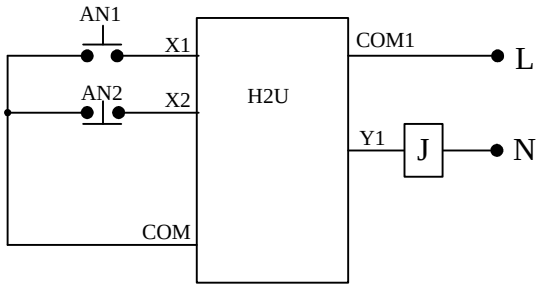
随着现代 PLC 的发展，PLC 不仅可以完成顺序逻辑控制功能，还能完成数值计算功能，如数值比较、四则运算、函数运算等，数值宽度有 16bit、32bit、浮点等，在 H2U 系列 PLC 中提供了大量的寄存器 D 元件，可在梯形图程序中用于数值运算。

梯形图的设计思想与传统继电器控制系统的设计方法基本相同，以常见的电磁开关的电气原理为例：



电磁开关电气接线原理图　　等效梯形图

从图中可见，J1 为继电器或接触器，AN1 为启动 J1 的按钮，使用其常开接点；而 AN2 为断开 J1 的按钮，使用了其常闭接点；另外使用了 J1 的常开型辅助触点作为状态保持用。若按上图右侧的梯形图编程，按下图设计 PLC 的信号输入连接，便可实现相同的起停控制功能了。



1.2 梯形图编程时使用的元件符号：

梯形图设计中使用的元件符号及特性说明如下表，通过将这些“触点”元件的“与”“或”逻辑组合，输出到元件“线圈”：

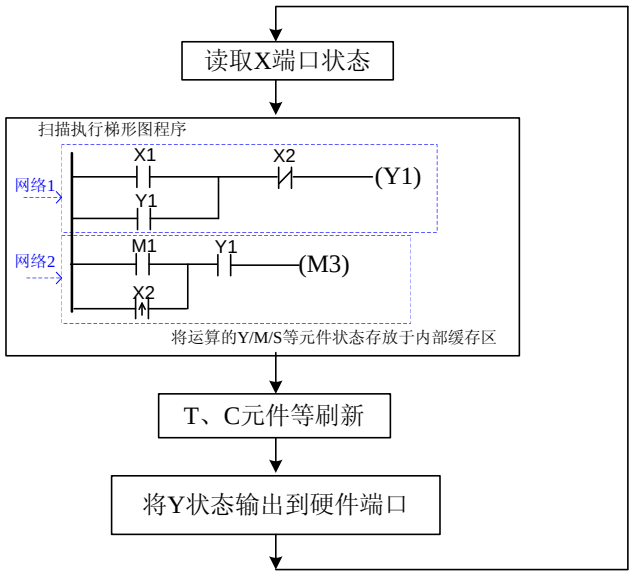
符号	说明	动作特性
⊣ ⊢	触点元件，代表元件的常开型触点，有输入 X 信号触点、输出 Y 的触点、中间继电器 M、时间继电器 T、计数器 C 的输出触点等。对于 Y、M、T、C 等元件，在未动作状态也为 OFF。	X:当 X 端口信号接点闭合时，状态为 ON；端口信号为断开状态时，触点状态为 OFF
		Y：当 Y 继电器的“线圈”得电时为 ON，否则为 OFF。Y 最后状态将对应于 PLC 的输出 Y 端口的状态。
		M：当 M 继电器的“线圈”得电时为 ON，否则为 OFF
		S：当 S 作为普通标志元件使用时，S 继电器的“线圈”得电时为 ON，否则为 OFF
		T：当对应的时间继电器线圈得电，且计时时间达到设定的时间，状态为 ON；否则为 OFF
		C：当对应的计数器的读数达到设定的时间，状态为 ON；否则为 OFF
⊣/⊢	触点元件，代表元件的常开型触点，有输入 X 信号触点、输出 Y 的触点、中间继电器 M、时间继电器 T、计数器 C 的输出触点等。	逻辑与状态刚好与 ⊣ ⊢ 的信号相反
⊣↑⊢	触点元件，仅在触点的上升沿有效	当触点元件（XYM）的状态由 OFF→ON 的上升沿变化时，该信号为有效，这个触点信号在一个扫描周期内有效，若下一状态不再变化，该信号恢复为“OFF”
⊣↓⊢	触点元件，仅在触点的下降沿有效	当触点元件（XYM）的状态由 ON→OFF 的下降沿变化时，该信号为有效，这个触点信号在一个扫描周期内有效，若下一状态不再变化，该信号恢复为“OFF”
—/—	状态取反	将当前信号点的状态进行取反
⊣STL⊢	步进梯形图中表示 S 状态信号	状态元件号对应的状态有效时，为 ON，其他状态号则均为 OFF
—( )	线圈元件，在梯形图中是被激励的对象	Y、M 元件的线圈“得电”时，其常开型触点动作闭合，其常闭常闭型触点动作端口，“失电”时恢复原来状态
		T 元件的线圈“得电”时，开始计时，“失电”时恢复为默认状态。当计时时间达到设定值时，其常开型触点动作闭合，其常闭常闭型触点动作端口。
		C 元件的线圈“得电”的瞬间，计数值增加 1，当计数值达到设定值时，其常开型触点动作闭合，其常闭常闭型触点动作端口。清取其“线圈”的操作，可使其计数值和触点恢复为默认状态。
		注意： X 输入元件没有线圈，用户程序不能修改其状态，只由外部的用户线路决定其状态。

—〔　〕	操作指令，对元件或线圈、参数等进行操作	完成逻辑操作、数据处理等众多功能。如〔RST　Y0〕、〔SET M2〕、〔MOV　K5　D100〕、〔JC　P1〕等指令。

1.3 PLC 的执行原理

当编程人员将设计编译好的梯形图程序下载到 PLC 的内存后，PLC 便可以对用户程序进行扫描执行了。

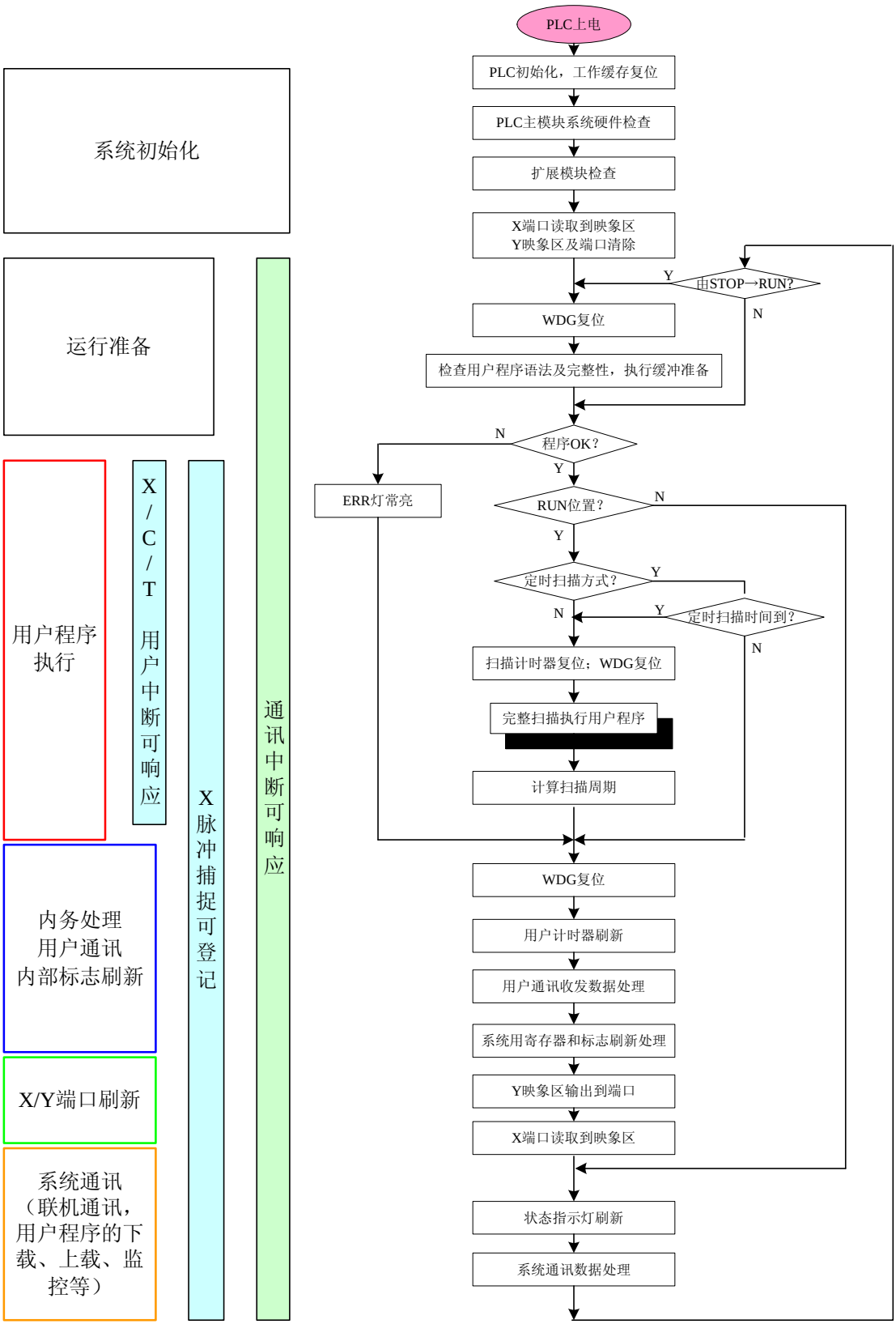
PLC 运行时，主要进行执行 X 输入检测、用户程序扫描运算、其他元件的状态刷新、将 Y 状态缓存状态输出到 PLC 的 Y 硬件端口等，这些工作内容周而复始的进行，其中的扫描执行用户程序是 PLC 的核心工作，过程如下例图：



每次执行用户程序前，首先将 X 硬件端口的状态读取后存放到 X 变量缓存区。

用户程序的扫描执行，是以用户程序的网络为单元进行逐步演算的，所谓“网络”是有联线关联的一组元件块，参见上图中的两个网络。执行演算从第一个网络开始，依次向下演算第二个、第三个…，直到最后一个网络。而对每个网络进行演算方式是，则由左至右，逐个将元件的“触点”状态进行逻辑计算综合，直到最右边，输出到元件的“线圈”，或根据逻辑决定是否执行某个操作。

梯形图中，左侧目前相当于电源的“火线”，其默认的（电位）状态为 ON，每经过一个元件后，逻辑运算结果暂存都被刷新，有时也称中间计算暂存状态为“能流”，中间逻辑计算结果为 ON，即“能流”为有效，本网络的输出状态即为输出电的能流状态；若最右端为操作类型，若能流为有效，就进行操作，否则不进行操作。



由上至下，直到主程序的所有网络都扫描执行完毕，还有各定时器的刷新、例行的通讯等数据的处理后，PLC 系统程序将 Y 寄存器缓存区的变量状态输出到 Y 硬件端口中。然后

又开始下一轮的用户程序扫描，如此周而复始，直到控制用户执行的“RUN/STOP”开关被拨动到 STOP 位置为止。

对于整个 PLC 而言，其系统软件还需完成一些运行准备、系统通讯、中断处理等工作，系统软件运行流程如上图所示。对于复杂的用户程序，在系统扫描用户程序过程中，还可以采用“中断”处理的方法响应“用户中断”信号，对重要信号（也有称重要“事件”）作及时处理。

所谓“中断”处理，就是 CPU 检测到特定信号时，立即停下（或中断）当前的例行工作，去执行特定的子程序，子程序执行完毕，才返回到先前被停下的工作点，继续执行例行工作。中断信号的请求能得到及时的响应处理，是“中断”功能的主要特点。

在 PLC 中，有高速信号输入（X0~X5）、高速计数、定时等中断（有时称为“用户中断”），还有通讯中断，包括系统通讯、用户程序发起的通讯等。在 PLC 中，各中断享有同一优先级，但不同中断类型，其允许区间稍有不同（参见前页插图）。

1.4 PLC 数值的基本知识

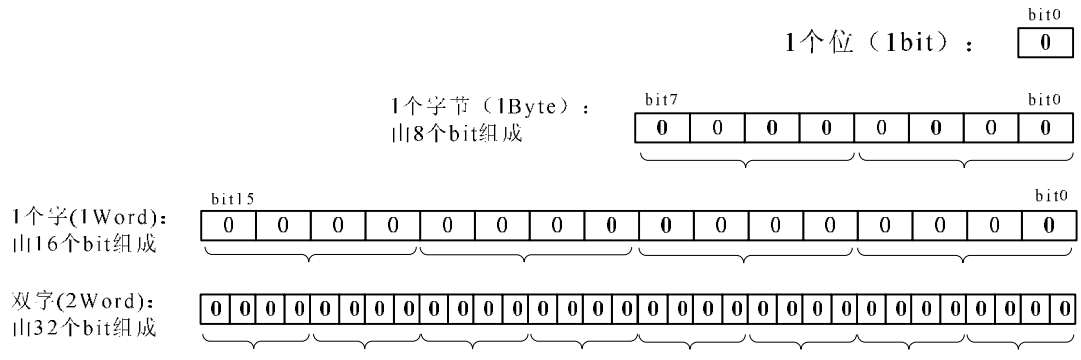
H2U 系列 PLC 内部采用高性能 32bit 作为核心处理器，其工作原理与其他的计算机设备是相似的。所有的 CPU 处理器采用的都是二进制码作为内部处理数据的格式，“数据”在计算机内部是以“信号电平”的方式进行处理的，其中信号电平只有“低”或“高”两个状态，分别对应于二进制数的“0”或“1”，信号电路中不容易出现误判，可确保处理结果的正确性。

二进制数

“二进制”用于计算机计算则是最简捷方便的进制，对于 1 位数的计算有：

0＋0＝0；0＋1＝1；1＋0＝1；1＋1＝10（有进位，此时需用 2 个位来表示）

这些计算只需用典型的“与”、“或”、“非”逻辑电路就可组合完成运算了。



当需要处理的数值比较大时，就需用多个二进制位来表示，位数越多，可表示的数值越大，现在常用的 CPU 位(bit)数有：

位数	可一次处理的最大数值	应用说明
4bit	15	消费类简单产品中还有使用，已很少
8bit	255	如 8051，常用于简单的控制系统中
16bit	65,535	如 808x，工业控制中有使用，使用较少
32bit	4,294,967,295	如 ARM，目前广泛应用于工控、消费类产品
64bit	18446744073709551615	通用计算机中使用

位数少的 CPU，并非不能处理大的数值，只不过需要多次运算，有时还需要编程人员熟悉算法。就像大车一次可以搬运的货物，用小车就需要往返多次才能搬完，车越小，需要的次数越多，耗时也越多。

H2U 系列 PLC 元件中，常用的数据宽度是 1Word（即 16bit）；部分计数器为 2Word（32bit）。对于 16bit 的无符号数据，用 2 进制表示的最大值为 1111,1111,1111,1111，换算为十进制就是 65,535。

十六进制

当二进制数值小的时候，尚能阅读，当位数比较多的时候，就比较难读难写了，将每 4 位二进制数分成一组，用 1 个数来代表，就成了 16 进制数（HEX）；一个 16bit 二进制数用 4 位十六进制数来表示，易读性大为增加。在十六进制数中数值 10～15（十进制）的数，分别以 A~F 的字符来代替。

八进制

由于传统习惯，在计算机中，以 8bit 宽度的数值、硬件端口数使用方式的为最多，8bit 被定义为 1Byte（即 1 个字节）；在 PLC 中也 8 个硬件端口作为分组，利于访问操作（读或写），如输入 X 端口、输出 Y 端口的编号就仍沿用八进制方式。

八进制数是由 3 位二进制数组成的，数字范围为 000～111，即 0～7，不可能存在 8、9。由于 CPU 一般为 8、16、32bit 等，但用于数据计算时，一般还是用十六进制，而不用八进制。

十进制

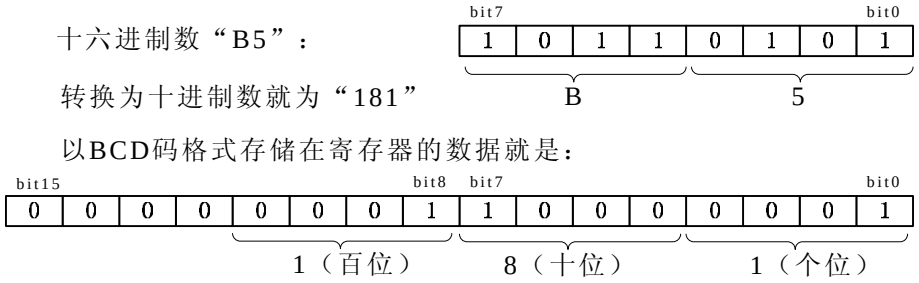
我们生活中习惯使用的数据是采用“十进制”，基本数字为 0～9 共 10 个数，若“9”＋“1”计数，便进位处理得到“10”。

日常生活中也有其他进制的，如星期日、星期一、….星期六分别以数字 0、1、…6 共 7 个数代表，就可理解为“七进制”，只不过“七进制”不便于计算，使用不多而已。

BCD 码

最符合人们阅读习惯的数字格式是十进制，在人们监控或设置工作参数时，往往需要采用十进制格式进行数据显示，而计算机内部使用的是 HEX 格式，故需采用一种底层为每 4 个二进制位组成一个数字位，而每个数字位只能为十进制数的 0～9，由此组成的数值，这种格式数字在存储器中的编码称为 BCD 码(Binary-Coded Decimal)。

在 PLC 内部，原理上用 4 位二进制数代表 1 位十进制数，在每一位 BCD 码中，不存在 HEX 格式中的 A~F。对于一个 8bit 宽度的寄存器单元，能存储的最大 BCD 数只能是 99，因此将 HEX 格式转换为 BCD 码后，会占用更大的存储空间。



PLC 内部总是按 HEX 格式进行数据计算的，在驱动非智能的显示设备（如数码管）显示数据之前，往往需要将 PLC 内部的十六进制（HEX）格式数据先转换为 BCD 码，然后进行显示输出；将用户以十进制方式设置的参数存入 PLC 内存之前，则往往需要将该 BCD 码转换为十六进制（HEX）格式。

H2U 系列 PLC 内部提供了 HEX 与 BCD 两种格式相互转换的命令，在需要进行显示输出，或设置开关读取的时候，执行该格式转换指令。

人们在电脑显示器上看到的十进制读数，都是经过了计算机自动作 BCD 转换后才显示的；监控时修改的参数，则是电脑软件作了 HEX 转换后写入的，无需人为干预而已。

各种进制数的对照举例：

二进制 BIN	八进制 OCT	十进制 DEC	十六进制 HEX	BCD 码	二进制 BIN	八进制 OCT	十进制 DEC	十六进制 HEX	BCD 码
0000	0	0	0	0	1000	不存在	8	8	8
0001	1	1	1	1	1001		9	9	9
0010	2	2	2	2	1010		10	A	不存在
0011	3	3	3	3	1011		11	B	
0100	4	4	4	4	1100		12	C	
0101	5	5	5	5	1101		13	D	
0110	6	6	6	6	1110		14	E	
0111	7	7	7	7	1111		15	F	

进制的转换

二进制、八进制、十六进制等进制的转换非常简单，例如 8 位的二进制数“10110101”，写成十六进制时，从右向左按 4 位一组分为“1011,0101”，用十六进制表示为“B5”；

写成八进制时，从右向左按 3 位一组分为“10,110,101”，用八进制表示为“265”；

要将二进制数换算为十进制数，则计算要复杂很多，最通用的方法可采用权重累加法，从最右边一位开始计算：

第 1 位（bit0）为 1 时，权重为 1 ，（即 2^0），否则为 0；

第 2 位（bit1）为 1 时，权重为 2 ，（即 2^1），否则为 0；

第 3 位（bit2）为 1 时，权重为 4 ，（即 2^2），否则为 0；

第 4 位（bit3）为 1 时，权重为 8 ，（即 2^3），否则为 0；

第 5 位（bit4）为 1 时，权重为 16，（即 2^4），否则为 0；

第 6 位（bit5）为 1 时，权重为 32 ，（即 2^5），否则为 0；

第 7 位（bit6）为 1 时，权重为 64 ，（即 2^6），否则为 0；

第 8 位（bit7）为 1 时，权重为 128 ，（即 2^7），否则为 0;　……

对于本例子中，将“10110101”转换为十进制数即为（128+0+32+16+0+4+0+1）=181。

对于 16bit 转换为十进制，如本例中的“B5”，也采用十六进制的权重累加法，从最右边一位开始计算：

第 1 位 HEX 数的权重为 1 ，（即 16^0），即该位的实际值×1；

第 2 位 HEX 数的权重为 16 ，（即 16^1），即该位的实际值×16；

第 3 位 HEX 数的权重为 256 ，（即 16^2），即该位的实际值×256；

第 4 位 HEX 数的权重为 4096 ，（即 16^3），即该位的实际值×4096；…..

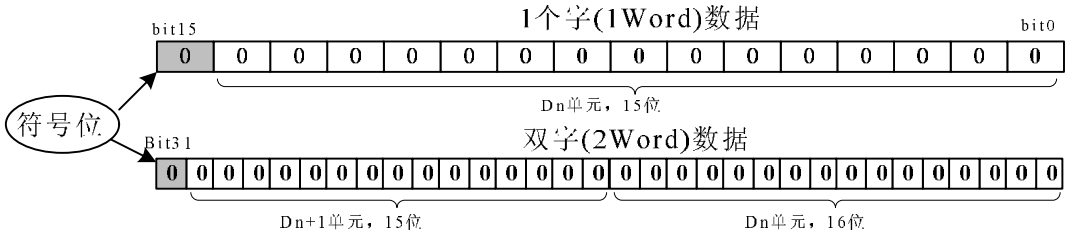
对于本例子中，将“B5”转换为十进制数即为（B×16＋5×1）=（11×16＋5）＝181。

读者熟悉了 HEX 转换为十进制的方法，可先将二进制或八进制划分为十六进制（每 4bit 一组），然后再作十进制转换，计算比较简捷。

有符号数与无符号数

PLC 内部的数据可以进行四则运算，运算结果可能产生负数，这样的计算结果就产生了“有符号数”，事实上 H2U 内部的寄存器 D、32bit 计数器 C 的数据、所有四则和函数运算指令都可按“有符号数”进行运算操作。

16bit 的 D 寄存器中最高位（bit15）便用于代表值的符号，因此 D 寄存器值的取值范围是-32,768～32,767。当用双字（32bit，2 个连续的 D 寄存器）表示一个数据时，用最高位（bit31）代表值的符号，因此 D 寄存器值的取值范围是-2,147,483,648～2,147,483,647。符号位如下图：



当符号位为 0 时，表示为正数，故 1Word 的正数是最大值为 HEX 格式的 H7FFF，即 32767；2Word 的正数是最大值为 HEX 格式的 H7FFFFFFF，即 2,147,483,647。

当符号位为 1 时表示负数，是其数值的补码，其绝对值的计算方法是“先将有符号数逐位取反，然后再加 1”，例如 HEX 格式的 HFFFF，其绝对值＝H0000+1＝1，即“HFFFF”代表-1；又例如 HEX 格式的 H8000，其绝对值＝H7FFF+1＝32768，即有符号数 H8000 代表-32768，是 1Word 寄存器最小的负值。同理，2Word 的最小负值为有符号数 H80000000，即-2,147,483,648。

进行数值比较大的加减运算时，要注意符号的处理，尤其是出现进位或借位操作时，要进行“借位标志”、“进位标志”的判断及相应处理，否则可能导致计算结果出错。

无符号数，即没有符号位，默认都为正数，对于 1Word 的寄存器，其取值范围是 0～65535，有些计时、计数的应用场合，就只有正数，需按无符号数处理，在作加减运算时，需要防止计算结果溢出，导致计算错误。

当进行逻辑运算时（如“逻辑与”、“逻辑或”等运算指令），操作数是当无符号数进行处理的，符号位（bit15）与其它位同等参与逻辑运算。

浮点数

浮点数在 PLC（或计算机）中用以近似表示任意实数，具体格式是由一个整数或定点数（即尾数）乘以某个基数（计算机中通常是 2）的整数次幂，这种表示方法类似于基数为 10 的科学记数法。

一个浮点数可用 m × b^e 来表示。其中 m 为尾数，形如±d.ddd...dd；b 为基数；e 为指数。例如 988436216 用十进制浮点数表示就可为 9.8844×10^{8},因尾数有四舍五入，精度有下降。但可以看到，使用浮点数可表示更大范围的数值。

由此可以看出，在计算机中表示一个浮点数，其结构如下：

尾数部分（定点小数）		阶码部分（定点整数）	
数符±	尾数 m	阶符±	阶码 e

在 PLC 或计算机内使用的浮点数，都采用了国际标准的格式，为了计算的方便，仪表都采用二进制浮点数格式。一个浮点数占用 32bit 的存储器单元。实际使用浮点数时，并不需要用户对浮点格式有特别的了解，计算机会对输入的实数自动作标准格式化处理。

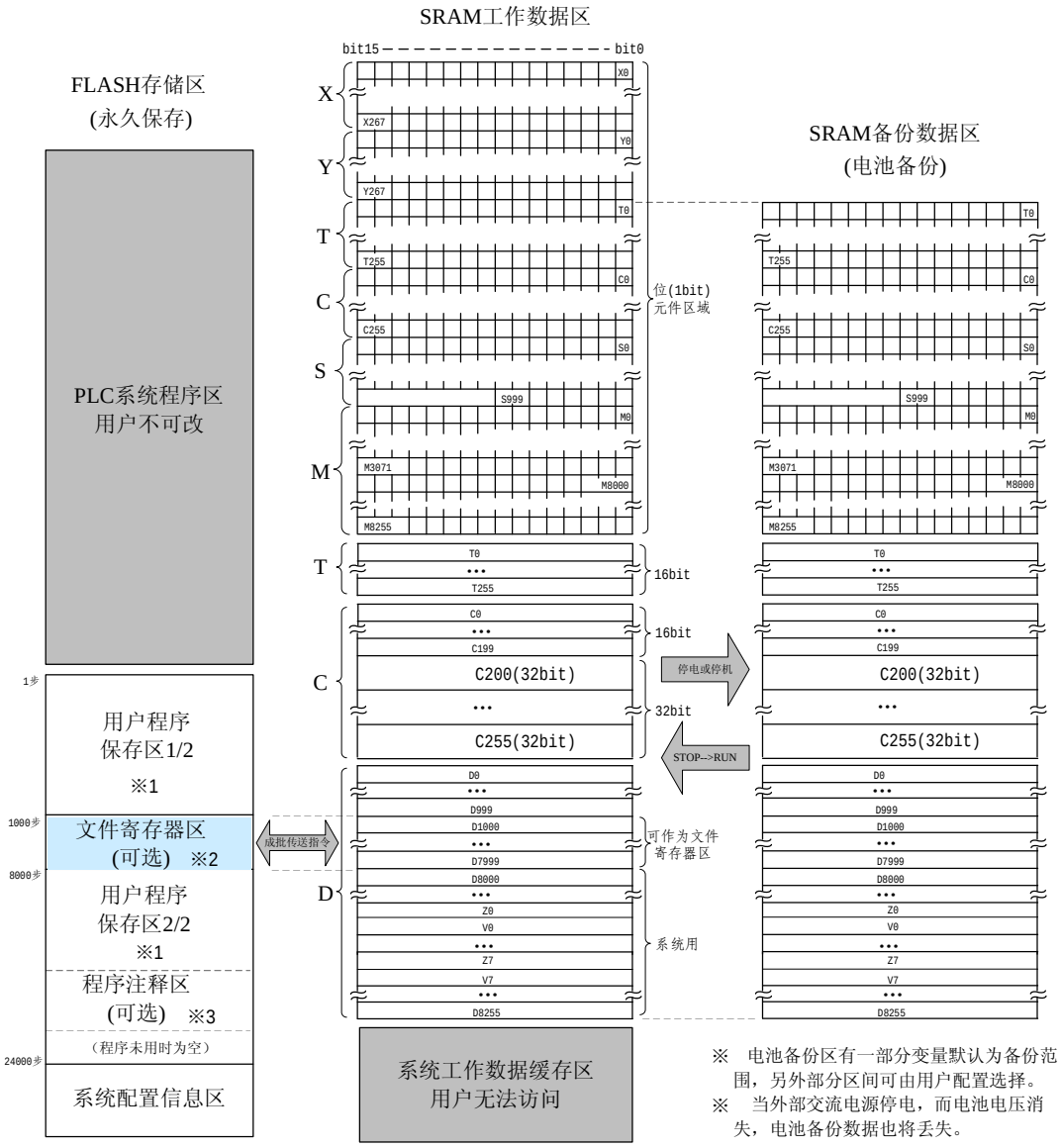
浮点计算是指浮点数参与的运算，这种运算通常伴随着因为无法精确表示而进行的近似或舍入。在 PLC 中有模拟量信号处理和运算时，可能用到浮点数。

H2U 系列 PLC 的内存结构

对于微机或单片机的系统，除了 CPU 内核以外，各种特性的内存是其主要配置。H2U 系列 PLC 中有如下几种内存：

类型	用途	特性
FLASH	保存系统程序	永久保存
	保存用户程序	永久保存，除非人为删除，或下载刷新
	文件寄存器数据	永久保存，除非人为删除或改写
SRAM	用于存放 PLC 的软元件、工作数据	有电池供电时，即使外部停电时数据也不会丢失

如前所述，PLC 内的软元件有“位元件”（触点元件），有 16bit 的“字元件”（寄存器 D 、计数器 C、计数器 T 等），还有 32bit 的“双字元件”（部分高速计数器 C），在 PLC 内部如下组织：



※1 用户程序保存区最大为24K步(Word)，存放用户程序时可自动回避文件寄存器区。

※2 文件寄存器保存区可定义，最大为7K步(Word)，占用用户程序空间。

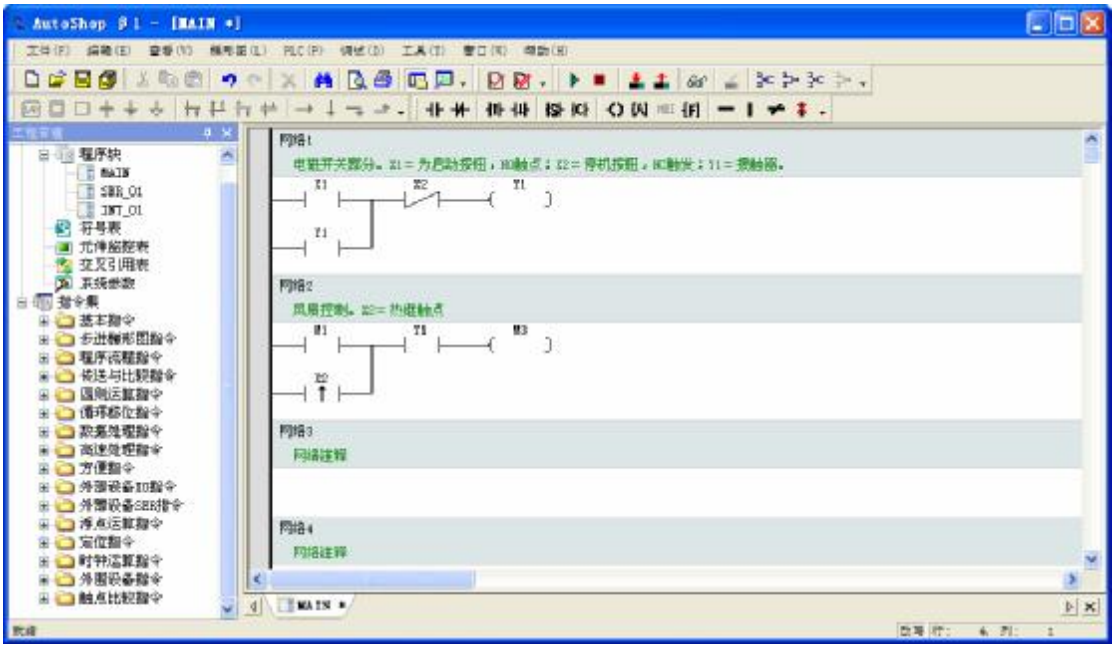
※3 用户程序注释保存区紧接梯形图程序，空间大小由注释内容决定，共享用户程序空间。

2. H2U 系列 PLC 的使用方法

2.1 使用 PLC 的软件硬件需求

项目	配置要求
电脑一台	PENTIUM 100MHz 以上主频；内存 256MB；鼠标等 具有 DB9 型 RS232 串行通讯口，（否则需准备 USB-RS232 转换器；或 USB-Mini DIN8 型专用下载电缆） 运行 Windows 2000/XP 操作系统； 硬盘剩余空间不小于 200MB；
AutoShop 编程软件	汇川控制技术公司开发的 AutoShop 软件，安装于 PC 电脑中，用于用户程序的编写、下载、监控调试等。也可采用其他兼容机型的编程环境。
H2U PLC 主模块	H2U 系列 PLC 主模块一只，可根据应用需要准备扩展模块
下载电缆一条	市售 RS232-Mini DIN8 插头的 PLC 程序下载专用电缆，用于用户程序的下载、调试、监控等，还可用于 HMI 连接。 对于没有配备 DB9 型 RS232 串口的电脑，也可准备 USB-Mini DIN8 型专用下载电缆。
电源联线和其他	用于 PLC 供电的电源线，根据需要可准备导线、拨码开关、螺丝批等常用工具

其中 AutoShop 编程软件为汇川控制技术公司研发的编程后台软件，在该软件环境下，可进行 H2U 系列 PLC 用户程序的编写、下载和监控等功能。



2.2 编程与用户程序下载

AutoShop 环境提供了梯形图、步进梯形图、SFC、指令表等编程语言，用户可选用自己熟悉的编程语言进行编程，根据 PLC 应用系统的控制工艺要求，设计程序。编程过程中，可随时进行按  编译，及时检查和修正编程错误。

程序设计完毕，在 PLC 和电脑正常连接，并已通电的情况下，按  即可下载用户程序，程序下载完毕，将 PLC 上 RUN/STOP 拨动开关拨至“RUN”位置，PLC 即可开始运行用户程序。

在 PLC 运行用户程序时，按  键即可进行运行的停止和运行命令操作；按  可监控 PLC 内各种继电器和寄存器 D 的状态和读数，在当前编程画面上显示出来，方便了程序调试。

2.4 与 HMI 的配合使用

H2U 系列 PLC 提供了 MODBUS 协议，也支持 FX2N/3U 系列 PLC 的监控协议，因此目前市售的 HMI 产品，基本上都可以与 H2U 系列 PLC 配合使用，包括连接电缆均可由市面购得。

关于 H2U 所支持的协议的种类和使用的详细说明，可参见 RS 通讯指令的解释。

3. 软元件说明

系统支持的软元件种类：

序号	元件类型	功能与分类
1	输入继电器 X	对应 PLC 的硬件开关量输入的位元件
2	输出继电器 Y	对应 PLC 的控制输出的位元件
3	中间继电器 M	普通中间继电器 M 位元件
		系统特殊继电器 M 位元件
4	状态继电器 S	步进控制用状态标志位元件
5	计时器 T	具有 1ms、10ms、100ms 步长的 16bit 计时器
6	计数器 C	具有 16bit/32bit 增/减型计数器
		高速计数器、单/双相各种计数器
7	寄存器 D	数据寄存器 D
		数据间接寻址寄存器 V、Z
		文件寄存器 D
8	指针 P、I	跳转指针 P
		子程序指针 P
		中断子程序 I，有高速输入、定时、计数等中断
9	常数 K、H	二进制、十进制、十六进制、浮点数等

3.1 输入继电器 X

输入继电器X代表PLC外部输入信号状态的元件，通过X端口来检测外部信号状态，0代表外部信号开路，1代表外部信号闭合。

用程序指令方法不能修改输入继电器的状态，其接点信号（常开型、常闭型）在用户程序中都可无限次使用。

继电器信号以 X0，X1,…X7，X10，X11，等符号标识，其序号是以 8 进制方式编号。控制器的计数器信号、外部中断信号、脉冲捕捉等功能是通过 X0～X7 端口输入。

型 号	输 入	输 出
H2U-1616MR/T	X000-X017	Y000-Y017
H2U-2416MR/T	X000-X027	Y000-Y017
H2U-3624MR/T	X000-X044	Y000-Y027
H2U-3624MTQ	X000-X044	Y000-Y027
H2U-3232MR/T	X000-X037	Y000-Y037
H2U-4040MR/T	X000-X047	Y000-Y047
H2U-6464MR	X000-X077	Y000-Y077

当接入扩展模块后，扩展模块上 X 端口的编号按紧接主模块上 X 端口的编号，依次向后编号，例如当主模块为 H2U-1616MR，现在要接入 H2U-1600EX 型扩展模块，因主模块最后的 X 端口编号为 X17，则扩展模块的 X 在编程时的访问编号为 X20~X37。

注意，扩展模块的编号总是从 8 进制个位为 0 开始的，例如，当主模块为 H2U-3624MR，其最后的 X 端口编号为 X44，扩展模块的 X 在编程时的访问编号为 X50~X67，即主模块上空缺的 X45~X47 的端口号被丢弃。扩展模块上 Y 端口也采取了同样的处理方法。

3.2 输出继电器 Y

输出继电器是直接关联到外部用户控制装置的硬件端口的软元件，在逻辑上与PLC的物理输出端口一一对应。PLC每次扫描完用户程序后，会将Y继电器的元件状态传送到PLC的硬件端口上，0表示输出端口开路；1表示输出端口闭合。

Y 继电器编号以 Y0，Y1,…Y7，Y10，Y11,…，等符号标识，其序号是以 8 进制方式编号。Y 继电器元件可在用户程序中无限次使用；

硬件上，根据输出元件的不同，可分为继电器型、晶体管型等；若有输出扩展模块端口，按照由主模块开始，依次序进行编号。

当接入扩展模块后，扩展模块上 Y 端口的编号按紧接主模块上 Y 端口的编号，依次向后编号，例如当主模块为 H2U-1616MR，现在要接入 H2U-0016EYR 型扩展模块，因主模块最后的 Y 端口编号为 Y17，则扩展模块的 X 在编程时的访问编号为 Y20~Y37。 注意，扩展模块的端口编号总是从 8 进制个位为 0 开始的。

3.3 辅助继电器 M

辅助继电器M元件用作用户程序执行过程中中间变量，如同实际电控系统中的辅助继电器，用于状态信息的传递，也可将多个M变量组成为字变量使用，M变量与外部端口没有直接

的联系，但可通过程序语句将X复制到M，或将M复制到Y的方式与外界发生联系，一个M变量可无限次使用。

辅助继电器 M 以 M0，M1,…,M8255 等符号标识，其序号是以 10 进制方式编号。M8000 以上的变量为系统专用变量，用于 PLC 用户程序与系统状态的交互；部分 M 变量具有掉电保存功能。

M 数量总计	一般用	停电保持用	停电保持专用	特殊用
3082 点	M0-M499 384 点 ※1	M500-M1023 524 点 ※3	M1024-M3071 2048 点 ※3	M8000-M8255 256 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．有关停电保持的特性，无法用参数来改变。

可编程控制器内的一般用辅助继电器、停电保持用辅助继电器的区域分配，可通过参数设定来进行调整。

可编程控制器内有大量的特殊辅助继电器。(参见系统特殊元件表)。这些特殊辅助继电器各有其特定的功能，可分为以下两类。

●触点利用型的特殊辅助继电器，为 PLC 系统自动驱动线圈，用户程序只能读取使用，如：

M8000：运行监视器（在运行中接通），常用于需要驱动信号的指令之前。

M8002：初始脉冲（仅在运行开始时瞬间接通），常用于只需执行一次初始化指令。

M8012：100ms 时钟脉冲，用于产生固定间隔翻转的信号。

●线圈驱动型特殊辅助继电器，为用户程序驱动线圈，用于控制 PLC 的工作状态和执行模式等，如：

M8030 ： 电池发光两极管熄灯指令

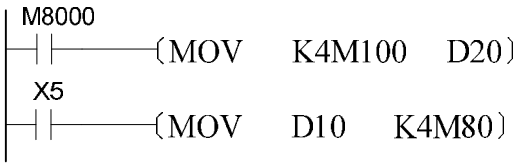
M8033 ： 停止时保持输出

M8034 ： 输出全部禁止

M8039 ： 恒定扫描

请注意，存在驱动时有效与 END 指令执行后有效两种情况；用户不可使用尚未定义的特殊辅助继电器。

●可以将连续的 M 变量按字节或字来进行访问（读或写），例如：



其中 K4M100 表示将 M100、M101、M102、….、M115 共 16 个单元，组成一个字的单元进行读操作，（M100 作为字的 bit0、…、M115 作为字的 bit15），这样可提高编程效率。

3.4 状态继电器 S

状态继电器 S 用于步进程序的设计和执行处理，利用 STL 步进指令控制步进状态 S 的转移，简化编程设计。

若没有采用 STL 编程方式，S 可当作普通的位元件，就如 M 变量一样来使用。状态 S 变量以 S0，S1,…S999 等符号标识，其序号是以 10 进制方式编号。部分 S 变量具有掉电保存功能。如下表：

一般用			停电保持用			报警器用
S0-S499 500 点 ※3	S0-S9 （10）点	S0-S9 （10）点	S200-S899 400 点 ※2	—	—	S900-S999 100 点※3

※1：非停电保持领域。通过参数的设定可变更停电保持的领域。

※2：停电保持领域。通过参数的设定可变更非停电保持的领域。

※3：停电保持特性。不可通过参数的设定变更。

3.5 计时器 T

计时器用于完成定时功能。每个计时器含有线圈、接点、计数时值寄存器，当计时器线圈“得电”（能流有效）时，计时器器开始计时，若计时值达到预设的时间值时，其接点动作，a接点（NO接点）闭合，b接点（NC接点）断开。若线圈“失电”（能流无效）时，计时器的接点恢复初始状态，计时值自动清除。也有部分计时器的具有累计、掉电保持等特性，重新上电后仍维持掉电前的数值。

计时器 T 以 T0，T1,…T255 等符号标识，其序号是以 10 进制方式编号。计时器有不同的计时步长，如有 1ms、10ms、100ms 等，部分具有掉电保持特性，如下表说明：

100ms 型 0.1~3276.7s	100ms 型 0.01~327.67s	10ms 型 0.01~327.67s	1ms 型 0.001~32.767s	100ms 累计型 0.1~3276.7s
T0~T199 共 200 点; 其中 T192~T199 可 用于中断/子程序	—	T200~T245 共 46 点	T246~T249 共 4 点 执行中断的保持用	T250~T255 共 6 点 保持用

●没有用作定时器使用的定时器编号，也可用作数值存储用的数据寄存器。

●定时器累计可编程控制器内的 1ms , 10ms , 100ms 等的时钟脉冲，当计时的时间达到设定数值时，其触点只有在执行线圈指令或 END 指令时，输出触点才能动作。

●采用程序存储器内的常数（K）作为设定值．也可用数据寄存器（D）的内容进行间接指定。注意，D 的内容必需在开始计时前设定好，当开始计数后，D 的数据变化只有在下一次启动计时的时候才能生效。

●从驱动定时器的线圈开始到定时器的触点动作，可能的定时长度说明如下：

最长的情况为（$T+T_0+a$），其中：T 为设定的定时时间；T_0 为程序扫描执行时间；a 为定时器的计时步长。

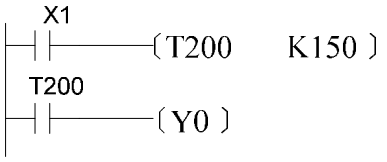
最短的的情况为（T－a）。

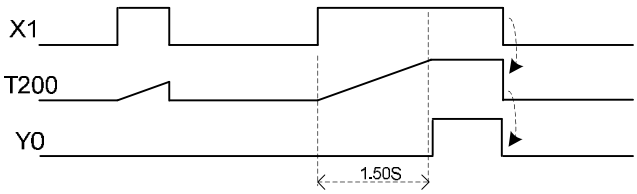
若计时器的触点指令位于线圈指令之前，最不理想的定时长度为（$T+2T_0$）。

●利用定时器的 b 触点，可以实现延时断开、自激振荡的输出信号等。

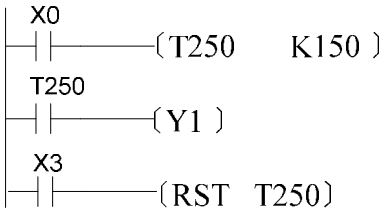
●PLC 还提供了特殊定时器指令，如 TTMR、STMR 等，请参见相应指令的说明。

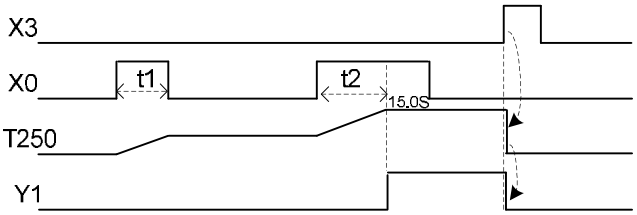
【使用举例 1】：普通计时器 T200 为 10ms 步长的计数器，实际动作延迟为 150×10ms＝1500ms，即 1.50s，动作原理为：



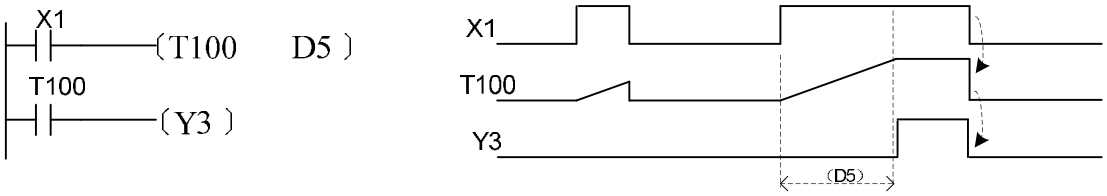


【使用举例 2】：对于有掉电保持的计时器 T250，驱动信号为 OFF，或 PLC 掉电时，其内部计数值维持不变，下次驱动信号为 ON 时，继续计时，直到满足计时到设定值时，输出触点闭合。当复位计时器线圈时，计时值清除，输出触点断开，如下图。因计数器 T250 为 100ms 步长的，实际动作延迟累计为 150×100ms＝15000ms，即 15.0s，即图中的（t1＋t2）时间：





【使用举例 3】：定时器的设定动作值可通过寄存器 D 来进行设定，如下图。（计数器计时过程中，若寄存器 D 内数值变化时，在下一次启动计时器启动时生效。）



3.6 计数器 C

计数器用于完成计数功能，每个计数器含有线圈、接点、计时时值寄存器，每当计数器线圈的驱动信号由OFF→ON时，计数器器读数增加1，若计时值达到预设的时间值时，其接点动作，a接点（NO接点）闭合，b接点（NC接点）断开；若清除计时值，输出a接点即断开，b接点（NC接点）闭合。部分计时器的具有掉电保持、累计等特性，重新上电后仍维持掉电前的数值。

计数器以 C0, C1,…,C255 进行标识，顺序按 10 进制编号。

计数器中有 16bit、32bit 宽度；有单向计数型、增减计数型、双相计数型等，部分计数器的计数值还具有掉电保持特性等，使用时根据需要选择合适的计数器。

16 位顺计数器 0~32,767 计数		32 位顺/计数器 -2,147,483,648~+2,147483647		
一般用	停电保持用	停电保持专用	特殊用	高速计数器
C0~C99 100 点 ※1	C100~C199 100 点 ※2	C200~C219 20 点 ※1	C220~C234 15 点 ※2	C235~C255 ※1, ※2

※1 非停电保持领域。通过设定参数可变更停电保持领域。

※2 停电保持领域。通过设定参数可变更非停电保持领域。

※3 试通过设定参数不可改变有关停电保持的特性。

不作为计数器使用的计数器编号，可以作为数据记忆用的数据寄存器使用。

对于 32bit 计数器 D200～D234，由特殊辅助继电器 M8200～M8234 作为增计数/减计数器切换控制，见下表：

计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换
C200	M8200	C209	M8209	C218	M8218	C226	M8226
C201	M8201	C210	M8210	C219	M8219	C227	M8227
C202	M8202	C211	M8211	—	—	C228	M8228
C203	M8203	C212	M8212	C220	M8220	C229	M8229
C204	M8204	C213	M8213	C221	M8221	C230	M8230
C205	M8205	C214	M8214	C222	M8222	C231	M8231
C206	M8206	C215	M8215	C223	M8223	C232	M8232
C207	M8207	C216	M8216	C224	M8224	C233	M8233
C208	M8208	C217	M8217	C225	M8225	C234	M8234

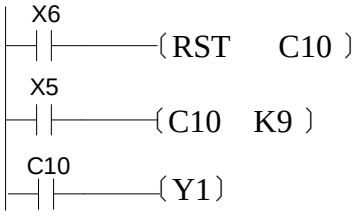
16bit 计数器与 32bit 计数器的特点如下表所示。可按计数方向的切换与计数范围的使用条件来分开使用。

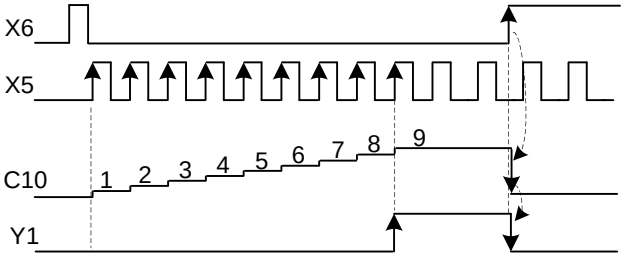
项目	16 位计数器	32 位计数器
计数方向	顺数	顺/倒切换使用（见上表）
设定值	1~32,767	-2,147,483,648~+2,147483647
指定的设定值	常数 K 或数据寄存器	常数 K，也可用 2 个 D 数据寄存器
当前值的变化	顺数后不变化	顺数后变化（循环计数器）
输出接点	顺数后保持动作	顺数保持动作，倒数复位
复位动作	执行 RST 命令时，计数器的当前值为零，输出接点复位	
当前值寄存器	16 位	32 位

16bit 计数器

●一般用计数器和停电保持用状态的分配，可通过系统参数配置进行变更设定。

●对于 16bit 计数器，其有效设定值为 K1～K32，767 (10 进制常数)；设定值 K0 和 K1 具有相同效果，即在第一次计数开始时输出触点就动作。如下例





计数输入 X5 每驱动 C10 线圈一次，计数器的当前值就增加，在执行第 9 次的线圈指令时．输出触点动作。以后即使计数输入 X5 再动作，计数器的当前值不变。如果复位输入 X6 为 ON，则执行 RST 指令，计数器的当前值清为 0，输出触点复位。

●计数器的设定值，除用上述常数 K 设定外，还可由数据寄存器编号指定。如上例中，指定 D20 ，如果 D20 的内容为 9，则与设定 K9 是一样的。

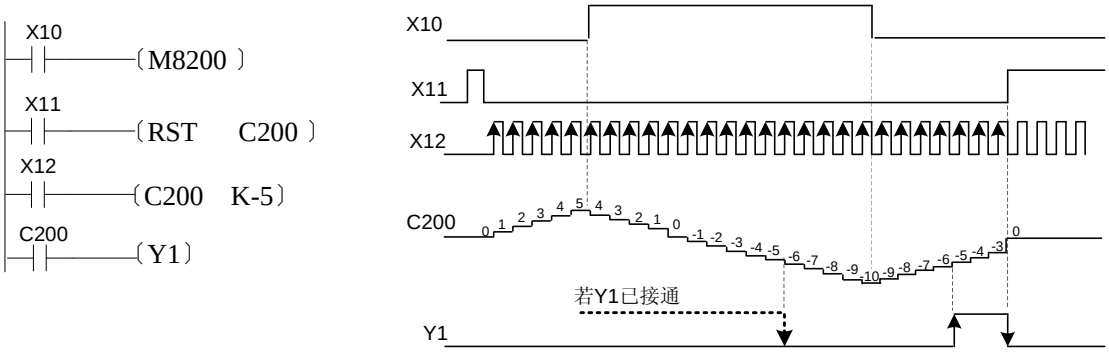
●在以 MOV 等指令将设定值以上的数据写入当前值寄存器时，则在下次输入时，输出线圈接通，当前值寄存器变为设定值。

●对于一般用计数器，如果切断可编程控制器的电源，则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，因此计数器可按上一次数值累计计数。

32bit 计数器

●对于 32bit 计数器，增计数/减计数的设定值有效范围为 -2,147,483,648～

+2,147483647(10 进制常数)，可用常数 K 或数据寄存器 D 的内容进行设定。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。



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，再输入一个脉冲后，则成为-2,147,483,648 。同样，如果从-2,147,483,648 开始减计数，再输入一个脉冲，则成为 2,147,483,647。(这类动作被称为环形计数)；如果复位输入 X11 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、输出触点动作与复位状态停电保持。

● 32bit 计数器也可作为 32bit 数据寄存器使用。但是，32bit 计数器不能作为 16 位应用指令中的软元件。

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

●对于 16bit 计数器，最高位（bit15）为符号位，处理的数据为 0～32767 范围，即只能为正数；

●对于 32bit 计数器，最高位（bit31，即高字节的最高位）为符号位，处理的数据范围为 -2,147,483,648～2,147,483,647；

高速计数器

H2U 系列 PLC 的内置高速计数器如下表所示，按计数器的编号（C）分配在输入 X000~X007 。而不作为高速计数器使用的 X 输入端口可在顺控程序内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。

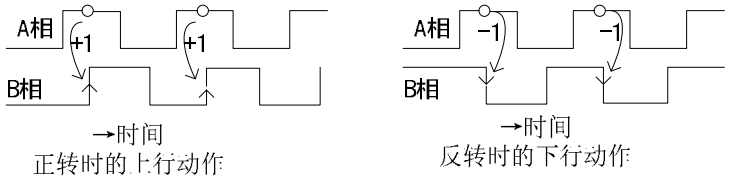
●高速计数器有如下几种类型：

1）1 相 1 计数型，只需要 1 个计数脉冲信号输入端，由对应的特殊 M 寄存器决定为增计数或减计数；部分计数器还具有硬件复位、起停的信号输入端口；

2）1 相 2 计数型，有 2 个计数脉冲信号输入端，分别为增计数脉冲输入端和减计数脉冲输

入端；部分计数器还具有硬件复位、起停的信号输入端口；

3）2 相 2 计数型，即 AB 两相计数脉冲计数器，是根据 AB 两相的相位决定计数的方向，计数方法是：当 A 脉冲为高电平时，B 相的脉冲上升沿作加计数，B 相的脉冲下降沿作减计数。通过读取 M8251-M8255 的状态，可监控 C251-C255 的增计数 / 减计数状态。



双相式编码器输出的是有 90 度相位差的 A 相和 B 相，据此高速计数器自动地进行增计数 / 减计数动作。

●通过特殊变量的设定，可以进行 4 倍频的 AB 相计数，可提供计数精度。

●部分计数器还具有硬件复位、起停的信号输入端口；

项目	单相单计数输入	单相双计数输入	双相单双计数输入
计数方向的指定	根据 M8235-M8245 的启动与否，C235-C245 作增/减计数。	对应于增计数输入或减计数输入的动作，计数器自动地增/减计数。	A 相输入处于 ON 同时，B 相输入处于 OFF→ON 时增计数动作，ON→OFF 时减计数动作。
计数方向监控	—	通过监控 M8246-M8255，可以知道增（OFF）减（ON）情况	

[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

增计数/减计数切换用特殊辅助继电器

种类	计数器号	UP/DN 指定
单相单计数输入	C235	M8235
	C236	M8236
	C237	M8237
	C238	M8238
	C239	M8239
	C240	M8240
	C241	M8241
	C242	M8242
	C243	M8243
	C244	M8244
	C245	M8245

计数方向监控用特殊辅助继电器

种类	计数器号	UP/DN 指定
单相双计数输入	C246	M8246
	C247	M8247
	C248	M8248
	C249	M8249
	C250	M8250
双相双计数输入	C251	M8251
	C252	M8252
	C253	M8253
	C254	M8254
	C255	M8255

● 高速计数器编号与对应的 X 端口配套使用，即指定了高速计数器 Cxxx 后，对应的 X 输

入端即被指定，故编程时不要让 X 端口有重复使用的情况，否则会出错。定义如下表：

分配输入	单相单计数输入										
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
X000	U/D						U/D			U/D	
X001		U/D					R			R	
X002			U/D					U/D			U/D
X003				U/D				R			R
X004					U/D				U/D		
X005						U/D			R		
X006										S	
X007											S

双计数及 A/B 相计数器如下表：

分配输入	单相双计数输入					A/B 相计数				
	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255
X000	U	U		U		A	A		A	
X001	D	D		D		B	B		B	
X002		R		R			R		R	
X003			U		U			A		A
X004			D		D			B		B
X005			R		R			R		R
X006				S					S	
X007					S					S

U：上升输入 ；D：下降输入 ；A：A 相输入 ；B：B 相输入 ；R：复位输入 ；S：开始输入

不作高速计数器使用的输入端子、可以作一般输入使用。

【表的阅读举例 1】

表中 C235 为单相单输入计数，使用 X0 输入口,不需要中断复位与中断启动端口；

如果使用 C235 计数器，即默认使用了 X0 输入端口，便不可再使用 C241 , C244 , C246 , C247 , C249 , C251 , C252 , C254 和中断 I00 口或者 M8l70（脉冲捕捉），也不能再使用 SPD 脉冲密度指令，因为这些计数器、中断、脉冲捕捉也需用到 X0 端口，形成了端口冲突。

【表的阅读举例 2】

表中 C254 为 2 相 2 输入计数器，即 AB 相计数器，

X0 口作为 A 相输入，X1 口作为 B 相输入，X2 口作为中断复位输入，X6 口作为中断启动输入；

如果使用 C235 计数器，即默认使用了 X0、X1、X2、X6 输入端口，与这些端口相关的计数器、中断口或者脉冲捕捉等，便都不能再使用了。

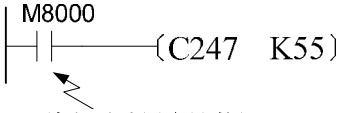
计数器使用说明：

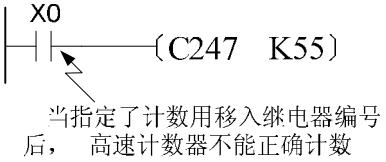
●高速计数器根据特定的输入执行动作，在相关信号的跳变沿，采用中断方式处理进行高速动作，故与 PLC 的扫描时间无关。

●高速计数器的当前值达到设定值时，如要立即进行输出处理，请使用高速脉冲比较指令 HSCS、HSCR、HSZ 等应用指令，具体参见指令解释。

●高速计数器的当前值达到设定值时，如要立即进行一些逻辑处理，可使用高速计数中断，使用高速脉冲比较指令 HSCS，将指令的操作指定为 I0x0 中断（其中 x＝1~6 中断号），当然必需编写好对应中断号的子程序。

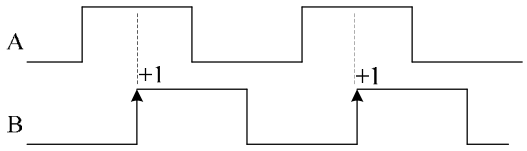
●高速计数器的线圈驱动用触点，在高速计数时，请采用一直接通的触点。



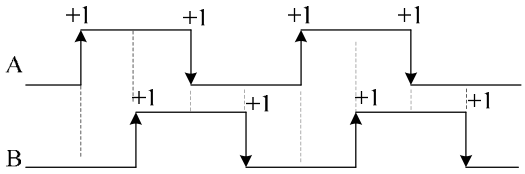


●如果对高速计数器的线圈编程，则与其对应的输入继电器的输入滤波器会自动变为 20 m s (X000 , X00l) ，或 50 m s (X002-X005)（初始值为 10ms）。此外，不作为高速计数器输入使用的输入继电器的输入滤波器维持初始值 10ms 。

●A/B 相高速计数器 T251～T255 有 1 倍频和 4 倍频两种频率模式，分别由特殊寄存器 M8195～M8199 设定，见下例：



M8195=0，T251为1倍频的A/B计数



M8195=1，T251为4倍频的A/B计数

●高速计数器均采用了硬件方式计数，对输入脉冲的总频率没有软件方面的限制；双相高速计数器的信号，占用两个脉冲输入口，对 PLC 的等效脉冲数影响按 2 倍计算，若 T251～T255 的 A/B 输入 4 倍频模式时，为软件计数模式，高速输入频率降为 25kHz。

●由于高速 X 计数、高速 Y 脉冲输出均采用中断方式进行处理，故信号路数较多时，可能会影响程序的执行速度，向高速计数器输入信号时，其所用频率要低于上述频率。如果输入超过这一频率的信号，可能会发生监视定时器（WDT）错误。

3.7 寄存器 D

数据寄存器 D

寄存器用于数据的运算和存储，如对定时器、计数器、模拟量参数的运算和运算、等，每个寄存器的宽度为 16bit。若采用 32bit 指令，则自动将相邻的寄存器组成为 32bit 寄存器使用，地址较低的为低字节，而地址较高的为高字节。

H2U 系列 PLC 多数指令中参与运算的数据是按有符号数进行处理的，对于 16bit 的寄存器，bit15 为符号位（0 表示正数，1 表示负数）；对于 32bit 的寄存器，高字节的 bit15 为符号位，数值范围为-32 , 768 ~+32，767。

当需要处理 32bit 的数据时，可将相邻的 2 个 D 寄存器组成为 32bit 双字，例如以 32bit 格式访问 D100 时，此时将高地址 D101 寄存器作为高字，同时将高字节的 bit15 作为双字的符号位，可处理-2 , 147 , 483 , 648-2 ,147 , 483 , 647 的数值。

寄存器以 D0, D1,…,D9,999 为标识，按 10 进制进行编号。

一般用	停电保持用	停电保持专用		特殊用	指定用
		普通用	（文件用）		
D0~D199 200 点 ※1	D200~D511 312 点 ※2	D512~D7999 7488 点 ※3	根据参数设定，可以将 D1000 以后作为文件寄存器	D8000~D8255 256 点	V0~V7 Z0~Z7 16 点 ※3

※1：非停电保持 领域。通过设定参数可变更停电保持领域。

※2：停电保持领域。通过设定参数可变更非停电保持领域。

※3：通过设定参数无法变更停电保持的特性。

●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）。在指定 32 位时，如果指定了低位（例如：D0），则高位为继其之后的编号（例如， D1）被自动占有。低位可用奇数或偶数的任意一种软元件编号指定，考虑到外围设备的监视功能，建议低位采用偶数软元件编号。

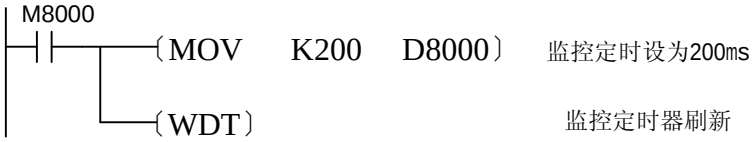
●一旦在数据寄存器中写入数据，只要不再写入其他数据，就不会变化。但是，在 RUN→STOP 时或停电时，所有数据被清除为 0。（如果驱动特殊的辅助继电器 M8033，则可以保持。）对此相对停电保持用的数据寄存器在 RUN / S TOP 和停电时也可保持其内容。

●利用系统参数配置功能．可改变 D 寄存器的一般用与停电保持用的分配；

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作为默认区域被占用。

●特殊用途的数据寄存器是指写入特定目的的数据，用于实现控制器的一些特殊功能，可理解为用户程序与 PLC 系统程序进行数据交互的特殊单元。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 MOV 传送指令，在 D8000 中写入目标时间。



另外还有一些特殊 D 寄存器，用于系统工作状态参数缓存，查询这些寄存器，可用于判断运行参数。

●关于特殊数据寄存器的停电保持特点请参照“特殊寄存器说明”。

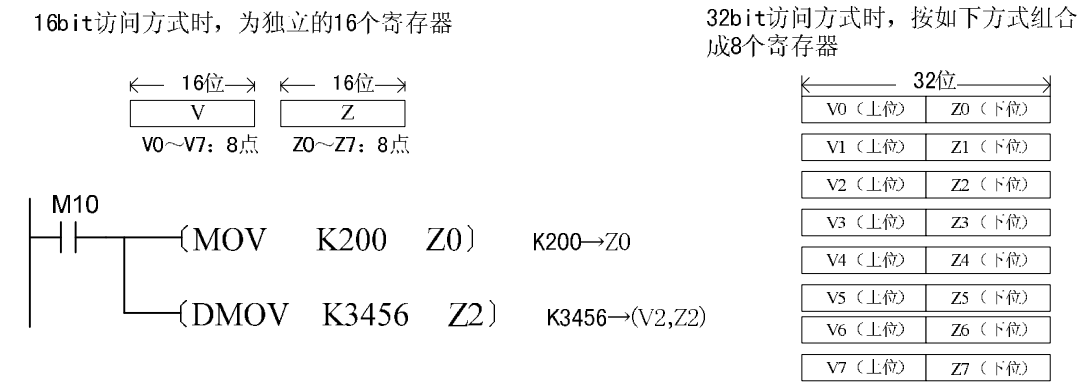
●数据寄存器可以处理各种数值数据，通过利用它，可以进行各种控制。如作为定时器与计数器的设定值被指定，用于数据的各种运算等，在后续的指令解释中，对支持使用 D 寄存器的指令有详细的说明。

数据寄存器 V、Z

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～V7，Z0～Z7 共有 16 个。

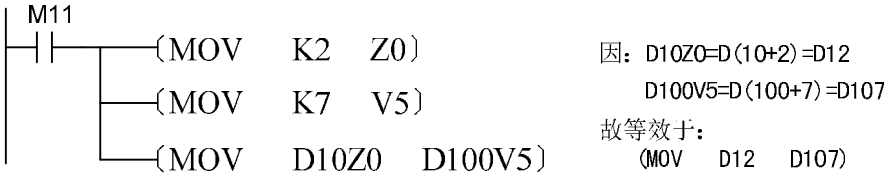
变址寄存器除了和普通的数据寄存器有相同的使用方法外，在应用指令的操作数中，还可以同其他的软元件编号或数值组合使用。但需注意 LD , AND , OUT 等基本顺控指令或步进梯形图指令的软元件编号不能同变址寄存器组合使用。

V、Z 寄存器可采用 16bit 和 32bit 方式进行访问，如下图说明：

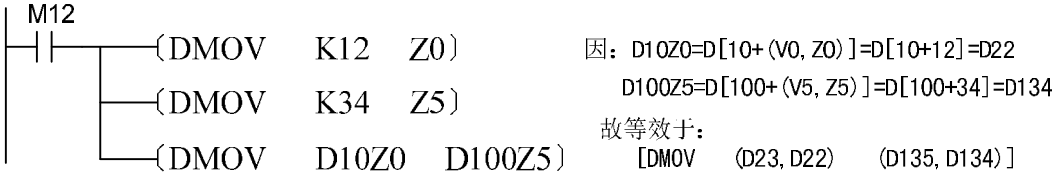


按照惯例，在处理 32 位应用指令中的软元件或处理超过 16 位范围的数值时，（为 32bit 寄存器方式），V（高位）、Z 低位）被同时访问，指定的寄存器名必须为 Z0～Z7。即使指定了 V0~V7 的高位侧，也无法进行变址。

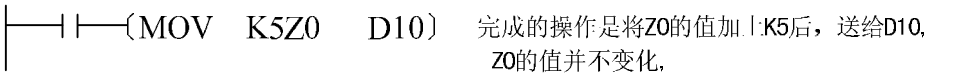
16bit 变址应用举例：



32bit 变址应用举例：



常数变址的特例：

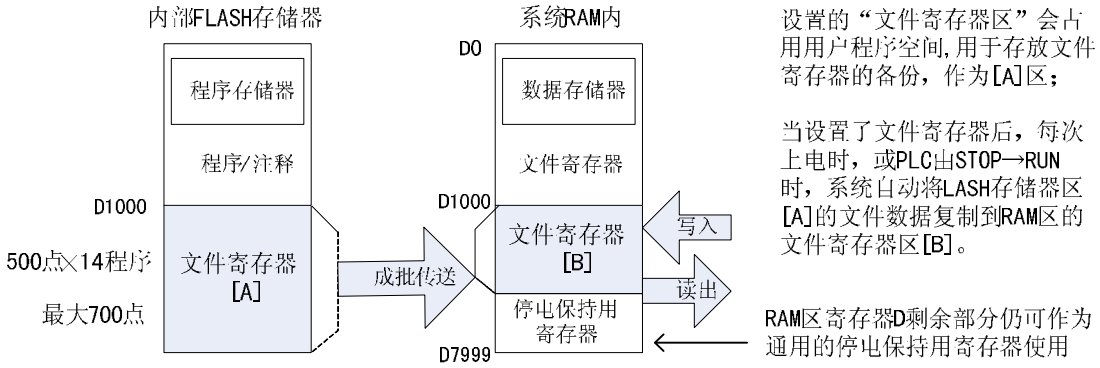


当 V、Z 间接寻址方式用于循环指令中（V、Z 随循环变量变化），进行成片数据区的操作，或用于查表操作等，简化编程，提高指令效率。

文件寄存器 D

数据寄存器 D1000 以后是普通用的保持用寄存器，可设定作为最大 7000 点的文件寄存器使用。通过参数设定，可指定 1～14 个块（1 个块相当于 500 个文件寄存器），但是每增加 1 个记录块就要减少 500 步的程序储存区域，用于备份文件寄存器。将 D1000 以后的一部分设定为文件寄存器时，剩余部分仍可作为通用的保持寄存器使用。

文件寄存器区域定义及处理如下示意图：

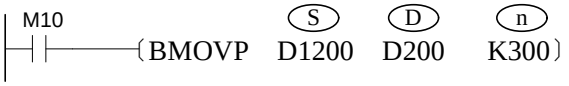


当控制器从 STOP→RUN 时内存中设定的文件寄存器区域［A］，系统程序自动将之批次传送主系统 RAM 中的数据存储区［B］部，数据存储区中已变化的内容将被初始化。此后，除块传送指令 BMOV 外，程序中对元件的操作都将是针对寄存器区域[B]中的元件。

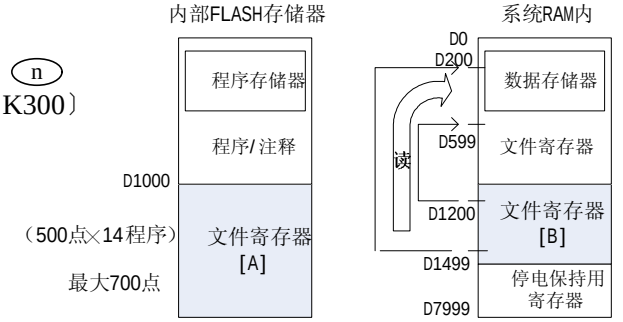
●应用指令（BMOV 除外）中的操作数或定时器、计数器的间接指定值，或作为 RST 指令内的软元件，若指定为 D1000 以后的软元件，与[B]部中一般的数据寄存器采用相同的处理

方式进行读写。

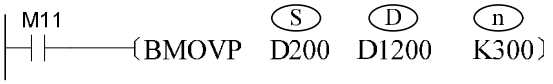
读取操作：



当M10=ON时，会进行如右图的操作；
通过Ⓢ和Ⓓ的指定，也可将普通区域数据搬移到文件区,进行写入操作；
若将Ⓢ和Ⓓ指定为同一地址，则成为同编号寄存器的更新操作.

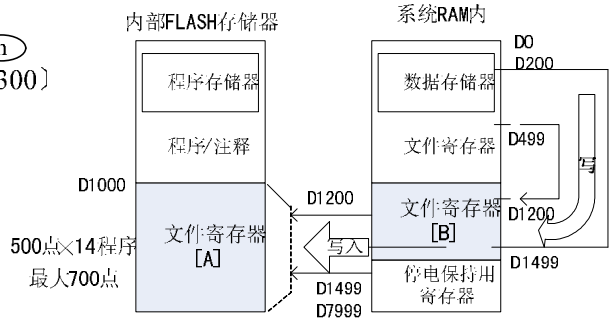


写入操作：



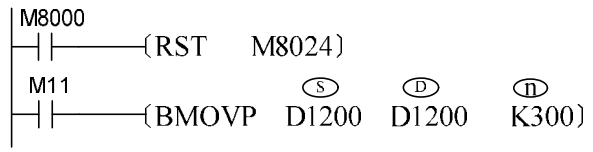
当M10=ON时，会进行如右图的操作；
首先将普通区的数据写入到文件寄存器[B]区，再复制一份到FLASH内存[A]区，确保数据掉电不会丢失。

若M8024=1，BMOV的操作源S和目标寄存器D正好互换

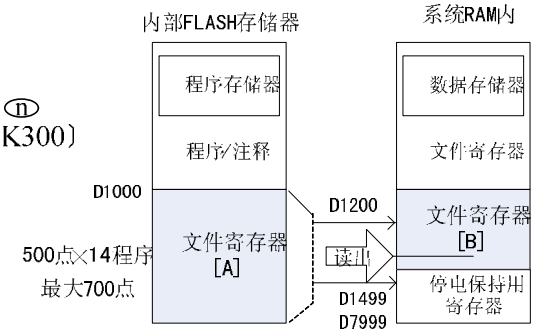


●当需要利用顺控程序保存数据储存区中变化的数据时，请利用块传送指令 BMOV 的同编号更新模式，将文件寄存器[A]区域，更新为变化的值。

读出操作：



若M8024=1，BMOV的操作源S和目标寄存器D正好互换,即为写入操作



●在外围设备上对文件寄存器进行监视时，将数据存储器内的数据寄存器[B]区域读出。而且从外围设备进行文件寄存器软元件的“当前值变更”、“强制复位”或“ PC 内存的全部清除”的情况下，是对程序存储区内的文件寄存器区域 [A]部进行修改，随后向数据寄存器区域 [B]部自动传送。

●因 FLASH 内存的写入寿命为 10 万次，向文件寄存器中作写入操作时，注意不要采用连续型写入指令反复进行写入操作，以免损坏存储器。

3.8 子程序与中断指针 P、I

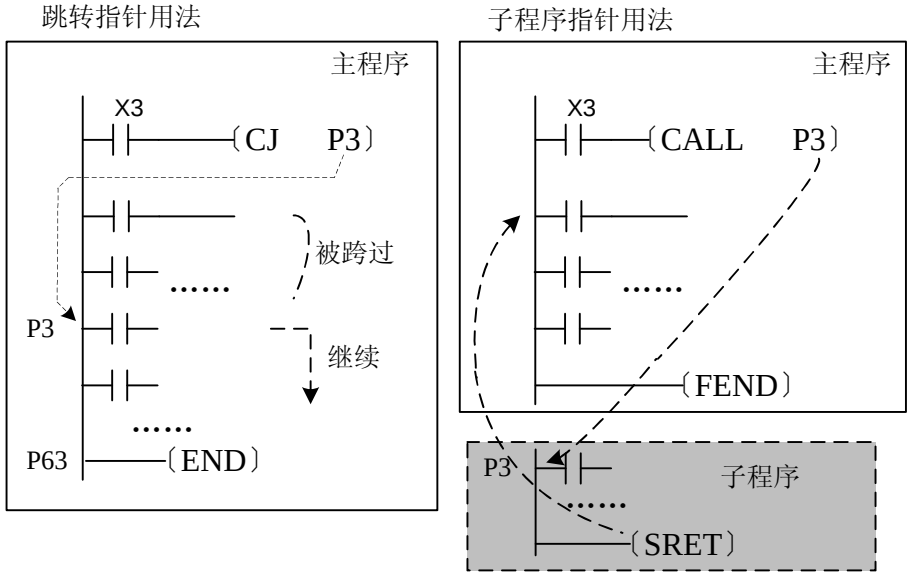
指针（P）用于跳转程序的入口地址和子程序启始地址的标识；指针（I）则用于中断程

序的启始地址标识，其编号采用十进制数分配，如下表所示：

分支用	结束跳转用	输入中断用	定时中断用	高速计数器中断用
P0~P62; P64~P127 共 127 点	P63 共 1 点	I00x(X0) I10x(X1) I20x(X2) I30x(X3) I40x(X4) I50x(X5) x=0 上升沿中断 x=1 下降沿中断 共 12 点	1600 1700 1800 共 3 点	1010 1040 1020 1050 1030 1060 共 6 点

因外部输入中断、高速计数、脉冲频率测量等功能都是通过 X0~X7 端口输入的，故这几项功能所使用的 X 端口不能有重复使用的现象，故使用输入中断指针时，注意端口的功能安排，检查高速计数器、脉冲密度指令所用的输入端口号情况。

跳转指针（P）和子程序指针（P）的使用及差别如下图所示。跳转指针（P）引导的指令语句仍在主程序内，只是用于在满足条件是跨过一部分指令语句；但子程序指针（P）则用于一段子程序，若主程序中条件满足，调用子程序，在子程序执行完毕（SRET）后，要返回原调用（CALL）指令的下一步继续执行。



●两种 P 指针使用同一种编号体现，在定义 P 指针时不要有重复；

●P63 指针为专用指针编号，指向程序的结束语句 END，注意不要再对 P63 编程。

指针（I）用于指定中断程序的启始地址，而中断子程序是在“中断允许”的情况下，当信号条件满足的瞬间，PLC 系统暂停主程序的正常执行（记住当前暂停点），从指定的 I 指针所指定的地址入口，开始执行中断子程序，直到执行了 IRET 指令后，返回主程序的暂停点，

继续执行。因 PLC 系统对中断信号采取了高优先的响应处理，故不受扫描时间的影响。

PLC 系统提供了三种类型的中断，分别是：

l）X 输入中断：控制器的 X0~X5 可分别设定为中断输入端口，每个中断输入口又有上升沿中断、下降沿中断，通过中断号来进行划分：如“I100”中断号代表 X1 口的上升沿中断，而“I101”则代表 X1 口的下降沿中断。

2）定时器中断用：在各指定的中断循环时间（1ms-99ms）执行中断子程序。在需要有别于可编程控制器的运算周期的循环中断处理控制中使用。

系统提供了 3 个定时中断，定时中断的周期可编程决定。定时中断使用系统内部的定时器，不占用 T0~T255。

3）计数器中断用：根据可编程控制器内置的高速计数器的比较结果（HSCS），执行中断子程序，优先处理计数结果的控制。

当 HSCS 指令的输出目标设为 I010～I060 时，便使用了高速计数器中断，编程时需编制好相应的中断子程序，开启响应的中断允许标志，才能进行中断响应。

对应中断的“中断允许”标志如下表，各标志可以独立设置：

中断允许/禁止设置			
M8050	驱动 I00□中断禁止	X输入中断，共有12个中断，分别对应X0~X5端口的上升沿中断、下降沿中断。 □　中： 0=上升沿中断； 1=下降沿中断	每个标志对应1个外部中断的控制； 当该M标志为OFF时，允许对应的X中断； 当该M标志为ON时，禁止对应的X中断；
M8051	驱动 I10□中断禁止		
M8052	驱动 I20□中断禁止		
M8053	驱动 I30□中断禁止		
M8054	驱动 I40□中断禁止		
M8055	驱动 I50□中断禁止		
M8056	驱动 I600中断禁止	定时中断0	
M8057	驱动 I700中断禁止	定时中断1	
M8058	驱动 I800中断禁止	定时中断2	
M8059	驱动 计数器中断禁止	高速计数中断，共6个	为ON时，禁止I010～I060的中断

每个中断对应的“中断允许”标志开启后，还需要开启“全局中断允许”，即执行 EI 指令（FNC04）后才最后才能使能中断功能；若执行全局中断禁止 DI 指令（FNC05），则禁止所有的中断的响应。当启用了输入编号的中断允许设定标志，输入信号满足中断设定时，将执行对应的中断子程序。

每个中断子程序的末尾均要有 IRET 指令，以表示中断子程序完毕，PLC 执行了该语句后，便会跳回本中断程序开始执行之前的位置。

若需要对出现在 X0～X6 端口的瞬间脉冲信号作出反应，但对反应动作时间没有特别要求，就可以使用“脉冲捕捉”功能，PLC 会将出现在 X0～X5 端口的上升沿信号保存在 M8170~M8175 单元，主程序中可作为判断处理的依据，响应处理完毕，可人为将之清除。

3.9 常数 K、H

H2U 系列可编程控制器根据不同的用途和目的，使用 5 种类型的数值。其作用和功能如下：

类型	编程中应用说明
十进制数，DEC	●定时器和计数器的设定值（K 常数） ●辅助继电器（M ），定时器（T ），计数器（C) ，状态 S 等的编号（软元件编号） ●指定应用指令操作数中的数值与指令动作（K 常数）
十六进制数，HEX	●同 10 进制数一样，用于指定应用指令中的操作数与指定动作（H 常数）
二进制，BIN	●以十进制数或十六进制数对定时器、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，在外围设备上进行监控时，这些软元件将如图所示自动变换为十进制数（也可切换为 16 进制）
八进制，OCT	●输入继电器、输出继电器的软元件编号以 8 进制数值进行分配。因此，可进行［0-7 , 10-17 ， … 70-77 , 100-107］的进位，在 8 进制数中，不存在［8 , 9］
BCD	●BCD 是以 4 位二进制表示十进制数各位 0-9 数值的方法。各位的处理很容易，因此，可用于 BCD 输出形的数字式开关或七段码的显示器控制等方面
BIN 浮点数	●可编程控制器具有可进行高精度的浮点运算功能，内部用二进制（BIN）浮点数进行浮点运算
十进制浮点数	●十进制浮点值只用于监视，便于阅读。

常数 K

[K]是表示 10 进制整数的符号。主要用于指定定时器或计数器的设定值或应用指令操作数中的数值。16bit 指令中，常数 K 的取值范围为-32768 ～32767；32bit 指令中，常数 K 的取值范围为-2,47,483,648～2,147,483,647。

常数 H

[H]是 16 进制数的表示符号。主要用于指定应用指令的操作数的数值。常数 H 的取值范围为 0000～FFFF；32bit 指令中，常数 K 的取值范围为 0000,0000～FFFF,FFFF。

3.10 控制器软元件规格

输入端口 X	X0～X377，最大可达 256 点； XY 总和最大 256 点		X0～X5:具有中断功能； X0～X17:滤波时间可设；		8 进制命名规则
输出端口 Y	Y0～Y377，最大可达 255 点； XY 总和最大 256 点		[Y0～Y1]具高速脉冲输出功能； MTQ 版[Y0～Y4] 具高速脉冲输出功能		8 进制命名规则
辅助继电器 M	M0～M499 500 点，通用※1		[M500～M1023] 524 点，保存用 ※2 继电器	[M1024～M3071] 2048 点，保存用 ※3	M8000～M8255 256 点，特殊用
状态 S	S0～S499 共 500 点 ※1 初始用 S0～S9		[S500～S899]共 400 点，掉电保存用 ※2		[S900 ～ S999] 共 100 点，报警用※2
定时器 T	T0～T199 共 200 点，100ms。 子程序用：T192～T199		T200～T245 共 6 点，10ms	[T246～T249]共 4 点，1ms 累计※3	[T250～T255]共 6 点，100ms 累计※3
16 位向上计数器 C	C0～C99100 点，通用※1			[C100～C199]100 点，保存用※2	
32 位计数器 C	32 位 可逆		32 位 高速计数可逆 最多 6 点		
	C200～C219 20 点，通用 ※1	[C220～C234] 15 点，掉电保存用※2	[C235～C245] 单相单向计数输入※2	[C246～C250]单相双向计数输入※2	[C251～C255] 2 相计数输入※2
数据寄存器 D，V，Z	D0～D199 共 200 点，通用 ※1	[D200～D511] 共 312 点，保存用※2	[D512 ～ D7999] 共 7488 点，保存用※3	[D8000～D8255]共 256 点，特殊用	V7～V0，Z7～Z0 共 16 点，变址用
嵌套指针	N0～N7 8 点，主控用	P0 ～ P127 共 128 点，跳转子程序	I00*～I50*共 6 点，输入中断指针	I6** ～ I8** 共 3 点，定时中断指针	I010～I060 共 6 点，计数中断指针
常数	K（十进制）	16 位 -32,768～32,767		32 位 -2,147,483,648 ～ 2,147,483,647	
	H（十六进制）	16 位 0～FFFFH		32 位 0～FFFFFFFFH	
	F（浮点数）	-		32 位 1175×10^{-41}～3402×10^{35}	

[ ]内的元件为电池保存区

※1：非电池保存区。根据参数设定，可以变更为电池保存区。

※2：电池保存区。根据参数设定，可以变更非电池保存区。

※3：电池保存固定区，区域特性不能变更。

4. 逻辑指令表

在基本指令当中，有部分指令采用“功能号”编码方式，若以手持编程器输入程序，输入方式可使用键盘中相对应的指令按键输入或使用功能编号方式输入。每一个指令的功能和使用方法在第7章内有详细说明。

指令符号	FUN NO	功能	操作数类型	指令步长
LD		加载常开接点	S、X、Y、M、T、C	1
LDI		加载常闭接点	S、X、Y、M、T、C	1
LDP	90	取脉冲上升沿	S、X、Y、M、T、C	1
LDF	91	取脉冲下降沿	S、X、Y、M、T、C	1
AND		串联常开接点	S、X、Y、M、T、C	1
ANI		串联常闭接点	S、X、Y、M、T、C	1
ANB		串联回路方块	无	1
ANDP	92	与脉冲上升沿检测串行连接		3
ANDF	93	与脉冲(F)下降沿检测串行连接		3
OR		并联常开接点	S、X、Y、M、T、C	1
ORI		并联常闭接点	S、X、Y、M、T、C	1
ORB		并联回路方块	无	1
ORP	94	或脉冲上升沿检测并行连接		3
ORF	95	或脉冲(F)下降沿检测并行连接		
OUT		驱动线圈	S、Y、M	1
SET		置位动作保存线圈指令	S、Y、M	1
RST		接点或缓存器清除	S、Y、M、T、C、D	3
PLS		脉冲上升沿检测线圈指令		
PLF		脉冲(F)下降沿检测线圈指令		
MC		主控公用串行接点用线圈指令	N0~N7	3
MCR		主控复位公用串行接点解除指令	N0~N7	3
MPS		存入堆栈	无	1
MRD		读出堆栈(能流指针不变)	无	1
MPP		读出堆栈	无	1
NOP		无动作	无	1
INV	98	运算结果取反	无	1
END		程序结束	无	1
P		指针	0~127	1
I		中断插入指针	I101/I201/301/I401	1

5. STL/SFC 指令

5.1 STL 编程指令

STL	程序跳至副母线	S	1
RET	程序返回主母线	无	1

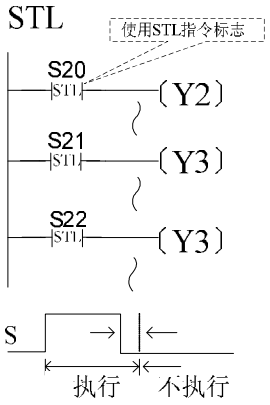
步进梯形图指令（STL,RET）

步进梯形图是一种根据被控设备的运行过程，分解为若干个状态或工序，针对每一个状态进行逻辑编程的方式，再根据信号条件进行状态间的切换。编程时采用 STL 梯形图，这种编程方法思路清晰，简化了逻辑设计，方便调试和维护。

步进梯形图指令可用梯形图表示，在步进梯形图中，将状态（S）看作为一个控制工序，从中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工序进行时，与前一工序不接通，以各道工序的简单顺序，即可控制设备。

步进梯形图有相应的编程规则，既包含了普通梯形图的编程方法，又与普通的梯形图编程有一定的差异，说明如下：

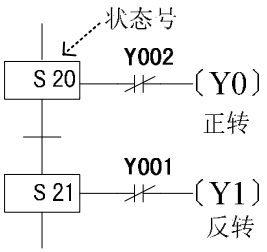
●步进梯形图程序以 STL 指令开始（注意与普通梯形图中 S 不同），以 RET 指令结束，中间的程序以 S 状态引导，后续该 S 状态的所有操作逻辑，包括条件满足时切换为下一状态的操作。



●如果STL触点S接通，则与其相连的回路动作；若S触点断开，则与其相连的回路不动作。但是在一个扫描周期以后，不再执行指令（跳转状态）。

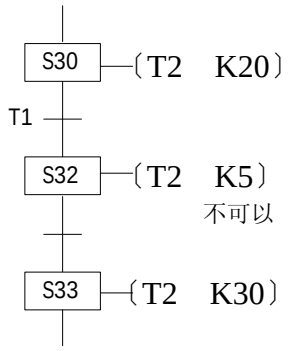
●在不同的状态S，可对同样的输出软元件（如Y3）。此时，S21或S22接通时，Y3被输出。但同样存在“双线圈”处理问题，请注意。

●状态S号编号不可重复使用。



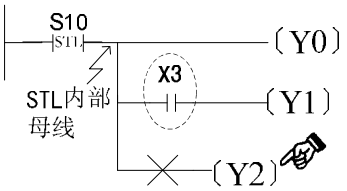
输出的互锁问题：

●在状态的转移过程中，会存在两种状态同时接通瞬间（一个扫描周期）。因此，为了避免不能同时接通的一对箱出同时接通，需要在可编程控制器外部设置互锁，同时要在相应的程序上设置互锁。



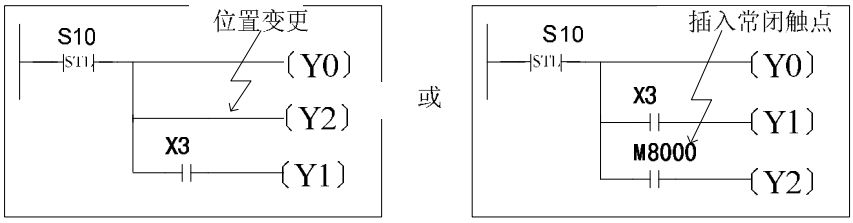
定时器重复使用的问题：

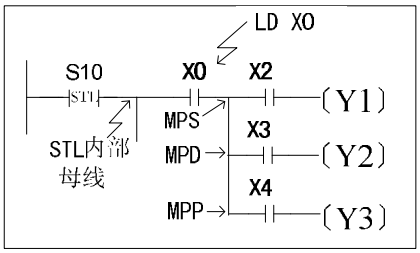
●定时器线圈与输出线圈一样，也可在不同状态间对同一软元件编程。但是，在相邻状态中则不能编程。如果在相邻状态下编程，则工序转移时定时器线圈不断开，当前值不能复位。



输出驱动方法

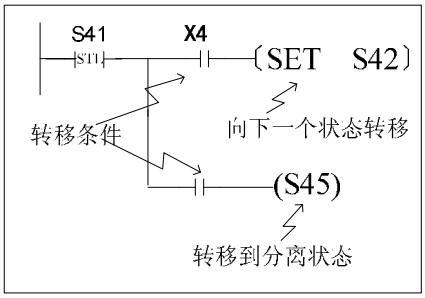
●从状态内的母线，一旦写入LD或LDI指令后，不能再使用不需要触点的指令（如左图所示），需要按下图所示的方法改变这样的回路。





栈操作指令MPS/MRD//MPP的位置

●在状态内，不能从 STL 内母线中直接使用 MPS / MRD / MPP 指令。而是须在 LD 或 LDI 指令以后编制程序。如左图所示。



状态的转移方法

●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

状态 \ 命令		LD/LDI/LDP/LDF,AND/ANI/ ANDP/ANDF,OR/ORI/ORF, INV，OUT,SET/RST，PLS/PLF	ANB/ORB MPS/MRD/ MPP	MC/MCR
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内，不能使用 STL 指令。

●在 STL 指令内不禁止使用跳转指令，但其动作复杂，建议不要使用。

5.2 SFC 顺序功能图编程

H2U 可编程控制器内置有利用 SFC 图（顺序功能图）的顺控功能，SFC 采用类似流程图的方式，将控制程序按流程图方式直观表述，使得编程调试、维护的大为简化。SFC 图设计时使用的符号定义如下：

	启始步进点图形，用于 S0～S9 状态的启始编程点，一个用户程序中只有一个该启始符。
	梯形图块图形，表示内部为一般步行梯形图的程序。常带梯形图块编号，如 LAD0、LAD1…等
	一般步进梯形图程序块图形，可使用 S10～S889 状态变量
	状态转移条件图形，用于标明上下相邻两状态转移的条件。
	状态分离图形，用于标明不相邻的两个状态的跳转。
	向上状态转移图形，用于标明向上转移的状态
	状态复位图形，将程序的状态复位到启始状态 S0
	选择分支图形，由同一步进点按不同条件转移到相应步进点。
	选择汇合图形，由两个以上步进点状态，经相应的转移条件后，转移到相同的步进点。
	并行分支图形，由同一步进点将综合体以同一转移条件转移到两个以上步进点。

⊥⊥	并行分支汇合图形，由两个以上不同步进点状态同时成立时，以同一转移条件转移到相同的步进点。

5.2.1 SFC 的编程特点：

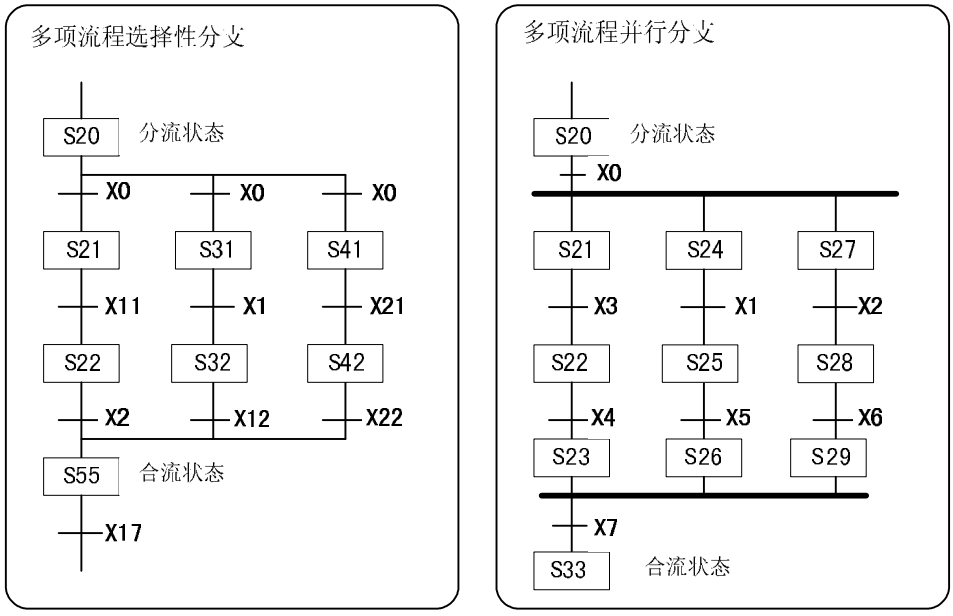
●在该梯形图块LADO中，采用可编程控制器由 STOP→RUN 转换时，瞬间动作的辅助继电器 M8002 ，使初始状态 S0 置位（ON）；

●可编程控制器中 S0-S9 为初始状态软元件；

●对各动作工序分配了 S20-S889 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 可用于特殊目的；

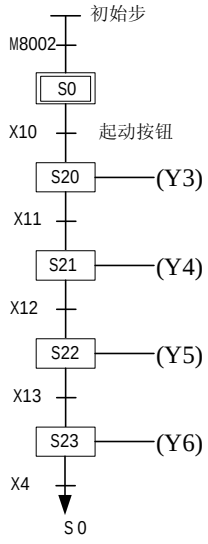
●可编程控制器内的定时器、计数器和辅助继电器等软元件，可随意使用；

●当有多项工序的选择、或有多个需要同时进行的工序时，采用如下方法：



可见在 SFC 图中，每道工序中设备的动作清晰易懂，其顺控设计容易，方便调测维护。

●SFC 图与步进梯形图指令都按一定的规则编程，可相互转换，其内容是一样的．也可使用大家熟悉的继电器梯形图。



5.2.2 SFC 的编程方法：

以下以举例的方式来逐项说明 SFC 编程的方法。

初始状态的作用

●初始状态位于 SFC 图的最前面，可使用状态号 S0-S9 。

●初始状态也要通过其他状态（如上图示例 S23 所示）来驱动时，

需要在运行开始时，利用其他方法事先驱动。

●下图所示例子是在可编程控制器由 STOP→RUN 切换时，利用只有瞬间动作的特殊辅助继电器 M8002 来驱动。

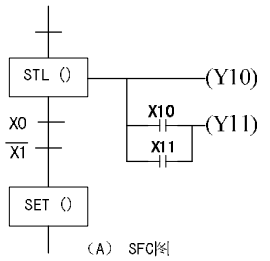
●初始状态以外的一般状态一定要通过来自其他状态的 STL 指令驱动，不能从状态以外驱动。

●将这种通过 STL 指令以外的触点驱动的状态称为初始状态。一定在流程的最前面表述。此外，对应初始状态的 STL 指令，必须在其之后的一系列 STL 指令之前编程。

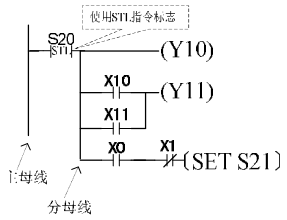
没有分支与汇合的一般流程

下图 A 为典型的 SFC 图，每个状态具有驱动负载、指定转移目标以及指定转移条件三种功能。使用继电器顺控方式表示 SFC 图时，是下图（B）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状态，则不需要进行负载的驱动处理。



（A） SFC图

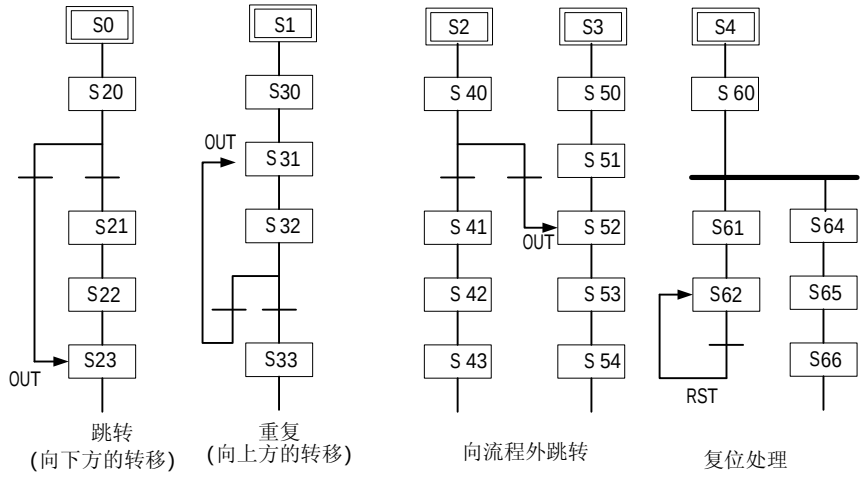


（B） 步进梯形图

```
STL  S20
OUT Y10
LD X10
OR X11
OUT Y11
LD X0
ANI X1
SET 21
```

当然上述步进梯形图也可用指令表程序来等效描述，如右图所示。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。在一系列的 STL 指令前面要有初始状态，最后一定要写入 RET 指令。

带有跳转与重复的一般状态



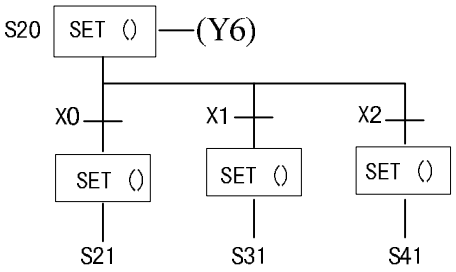
如上图所示，向下方状态的转移（跳转）、向上方状态的转移（重复）、向流程外的转移

等的分离状态转移，用 OUT 指令编程。

选择性分支与汇合状态

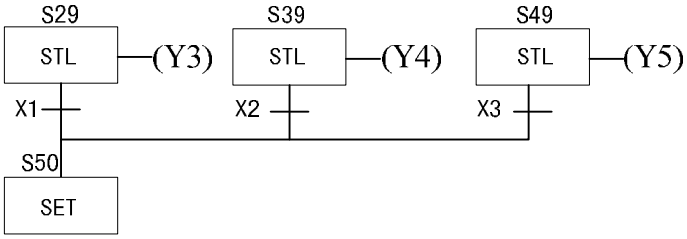
和一般状态的处理相同顺序，首先进行驱动输出处理，然后再进行状态转移处理。

等效是指令列表如右图：



等效指令表

```
STL S20
OUT Y6
LD X0
SET S21
LD X1
SET S31
LD X2
SET S41
```




```
STL S29
OUT Y3
STL S39
OUT Y4
STL S49
OUT Y5
STL S29
LD X1
SET S50
STL S39
LD X2
SET S50
STL S49
LD X3
SET S50
```

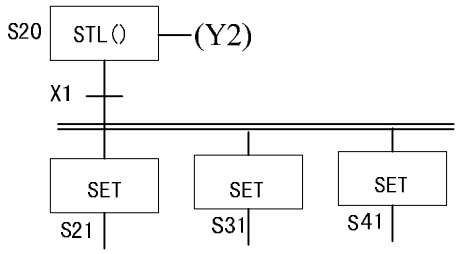
上图例为分支汇合的典型例子，右侧为等效指令表。

在分支与汇合的转移处理程序中，不能用：

MPS, MRD , MPP , ANB , ORB 指令；

另外，即使负载驱动回路也不能直接在 STL 指令后面使用 MPS 指令。

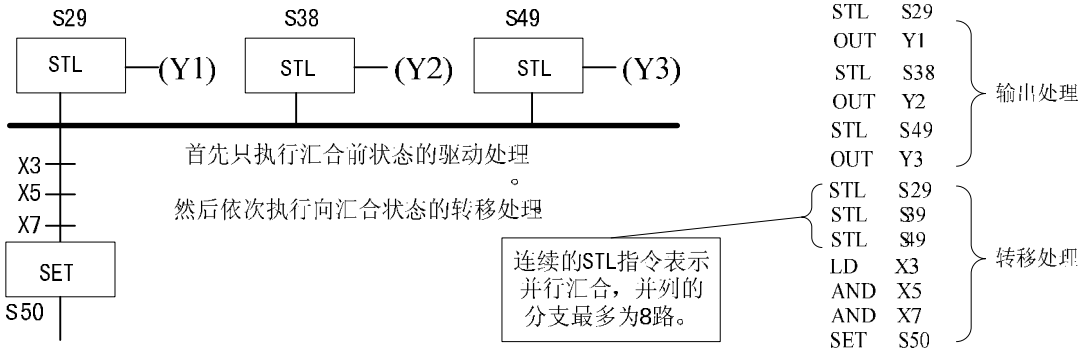
并行分支与汇合状态



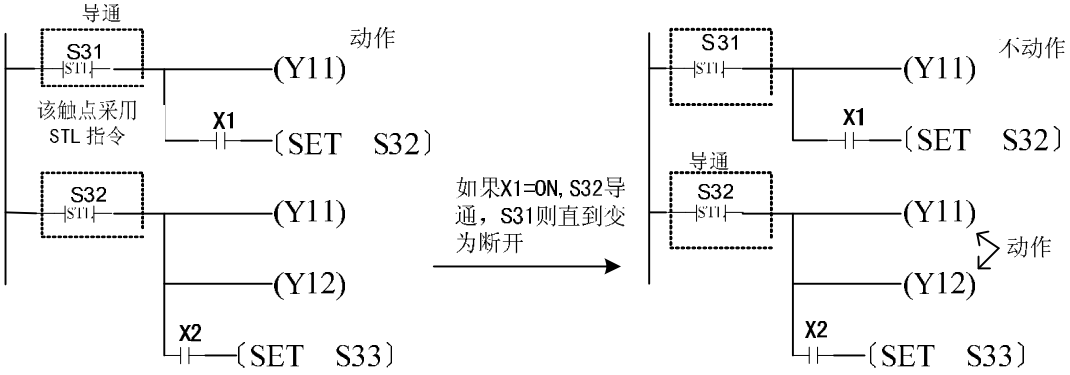
和一般状态的处理相同顺序，首先进行驱动输出处理，然后再进行状态转移处理。

等效是指令列表如右：

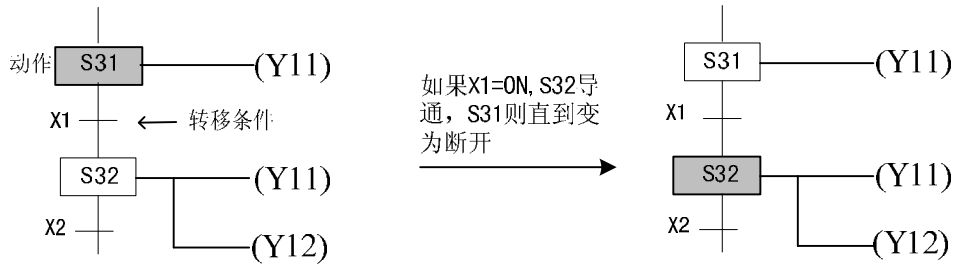
```
STL S20
OUT Y2
LD X1
SET S21
SET S31
SET S41
```



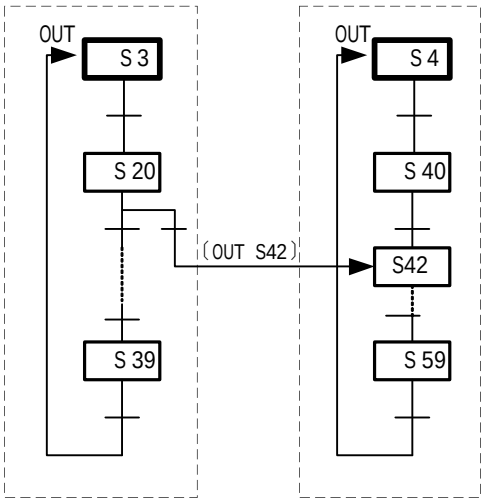
步进梯形图指令及其动作如下图所示：



若以 SFC 图表示上图所示的步进梯形图回路，则其表示如下图所示：



具有多个初始状态的 SFC 图的程序将各初始状态分离编程。



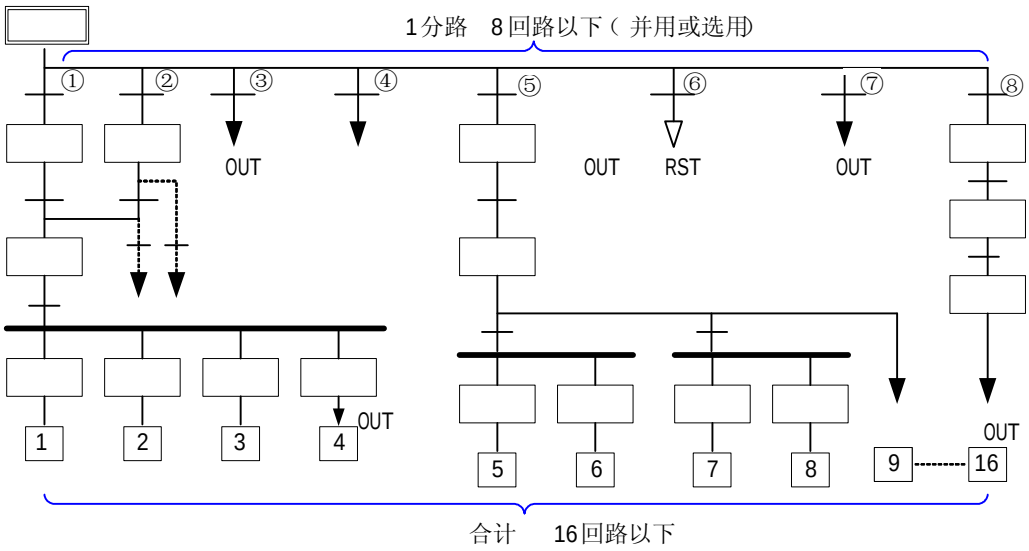
如左图所示，初始状态S3对应 其 STL指令的程序，而初始状态S4 则对应另一程序；

在自身的程序中，能够以STL以外的指令使用对方的状态号。如左图所示，在初始状态S3 的程序中包含OUT　S42的指令。

此外，初始状态S4 的程序中包含LD S39的指令。

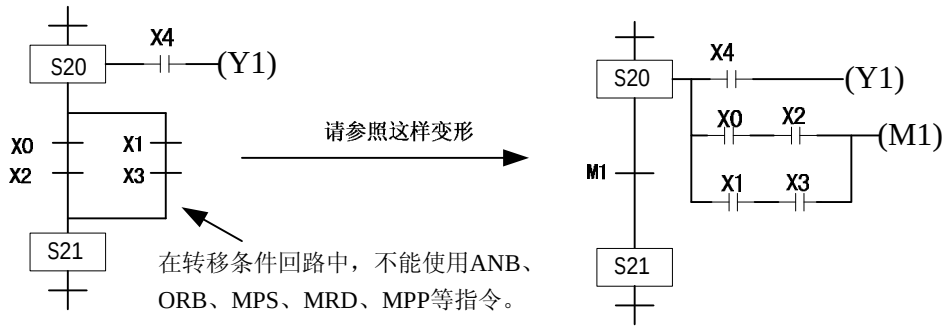
重要的是不可混杂STL指令。

一条并行分支或选择性分支的回路数限定为 8 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。如下图：

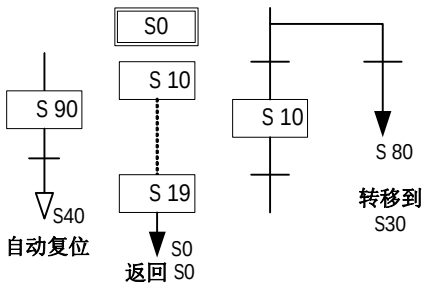


不能进行从汇合线或汇合前的状态开始向分离状态的转移处理或复位处理，应设置空状态，由分支线上向分离状态进行转移与复位处理。

对于状态转移条件比较复杂的情况，建议作简化处理，例如：



状态的转移与复位：



●在流程中，符号↓则表示向上面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。符号▽则表示状态的复位处理。

●状态标志 S 也可以采用 ZRST 指令对一个区间的标志进行批量复位

●编写（SFC）图程序时，可使用如下特殊辅助继电器，提高编程效率，如下表所示说明。

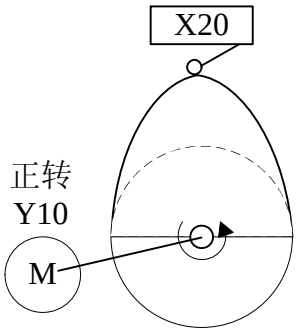
软元件号	名你	功能和用途
M8000	RUN 监视	可编程控制器在运行过程中，需要一直接通的继电器。可

		作为驱动的程序的输入条件或作为可编程控制器运行状态的显示来使用。
M8002	初始脉冲	在可编程控制器由 STOP→RUN 时，仅在瞬间（1 个扫描周期）接通的继电器。用于程序的初始设定或初始状态的置位。
M8040	禁止转移	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 动作	任一状态接通时，M8046 自动接通。用于避免与其他流程同时启动或用作工序的动作标志。
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中的状态并加以显示。详细事项请参照各外围设备的手册。

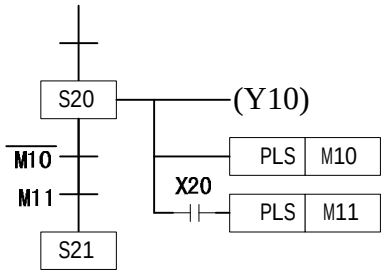
●停电保持用状态，是用电池保持其动作状态。在机械动作中途发生停电之后，再通电时从这里继续运行的情况下使用这些状态。

●RET 指令一定在一系列的 STL 指令的最后编写，没有编写 RET 指令时，会出现［程序出错］，可编程控制器不能运行。执行此指令，表明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。

●转移条件成立后状态的处理



构成转移条件的限位开关X20已经动作，而且在转动一次之后将进行下一次转移。

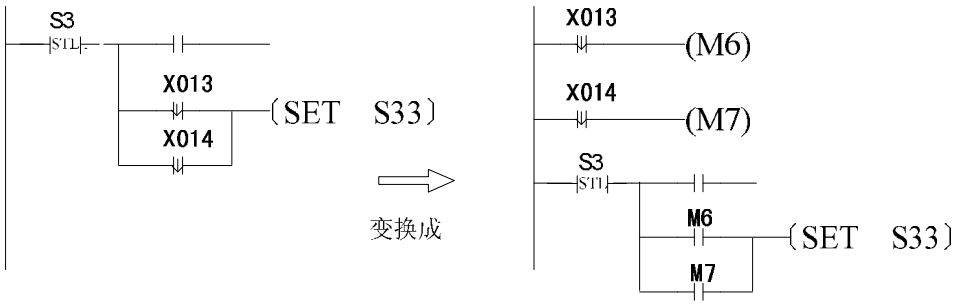


如上图所示的应用中，将转移条件脉冲化，S20 首次动作，通过 M10 使不产生转移。

●上升沿／下降沿检测触点使用时的注意事项：

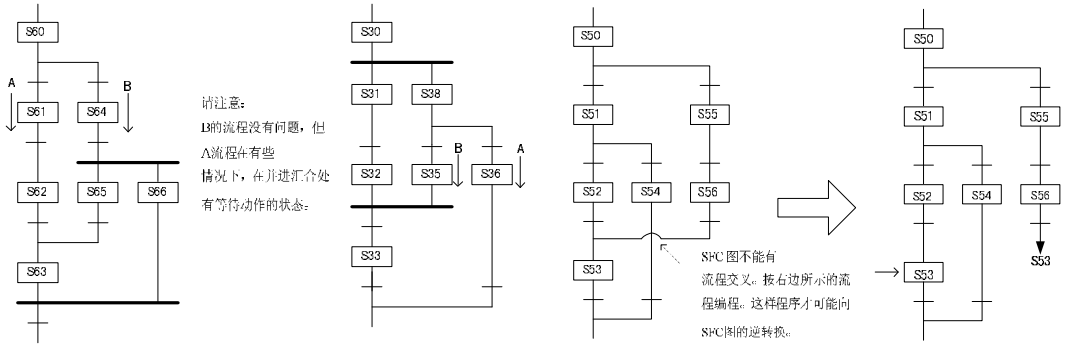
在状态内使用 LDP、LDF、ANDP、ANF、ORP、ORF 的上升沿／下降沿检侧触点时．状态断开时变化的触点，在状态再次接通时被检出。

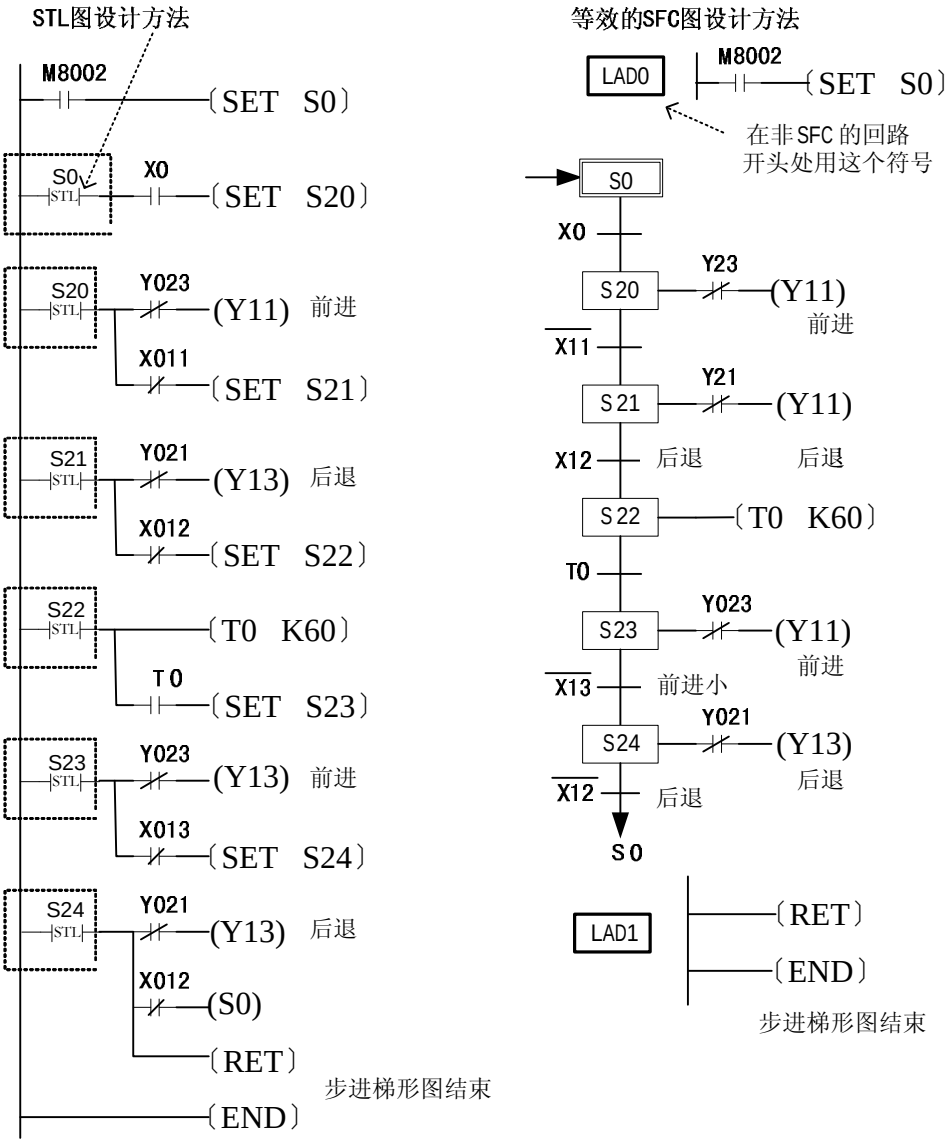
对于状态断开时变化的条件，必需上升沿／下降沿检侧时，请按下图所示，修改程序：



通过 X013 下降沿向 S33 转移后，若 X014 下降．此时因 S3 断开，X014 的下降沿无法检出，S3 再次接通时，被检测。因此，S3 第 2 次动作时，立即向 S33 转移。

分支与汇合的组合流程





6. 应用指令表

应用指令是指除逻辑处理指令之外的编程指令，指令功能涉及程序流控制、数值计算、高速信号处理、通讯与扩展、特殊控制功能等多个方面。这些指令除了具有指令名之外，往往还有“功能号”的编号，便于使用手持编程器设备进行编程。按功能分类列表如下，每一个指令的功能和使用方法在第7章内有详细说明。

分类	FNC NO.	指令符号			指令功能
		16bit	32bit	P指令	
程序流	00	CJ	-	✔	有条件跳转
	01	CALL	-	✔	子程序调用
	02	SRET	-	-	子程序返回
	03	IRET	-	-	中断返回

分类	FNC NO.	指令符号			指令功能
		16bit	32bit	P 指令	
	04	EI	-	-	开中断
	05	DI	-	-	关中断
	06	FEND	-	-	主程序结束
	07	WDT	-	✔	监控定时器
	08	FOR	-	-	循环范围开始
	09	NEXT	-	-	循环范围终了
传送与比较	10	CMP	✔	✔	比较
	11	ZCP	✔	✔	区域比较
	12	MOV	✔	✔	传送
	13	SMOV	-	✔	移位传送
	14	CML	✔	✔	倒转传送
	15	BMOV	-	✔	一并传送
	16	FMOV	✔	✔	多点传送
	17	XCH	✔	✔	交换
	18	BCD	✔	✔	BCD 转换
	19	BIN	✔	✔	BIN 转换
四则逻辑运算	20	ADD	✔	✔	BIN 加法
	21	SUB	✔	✔	BIN 减法
	22	MUL	✔	✔	BIN 乘法
	23	DIV	✔	✔	BIN 除法
	24	INC	✔	✔	BIN 加 1
	25	DEC	✔	✔	BIN 减 1
	26	WAND	✔	✔	逻辑字与
	27	WOR	✔	✔	逻辑字或
	28	WXOR	✔	✔	逻辑字异或
	29	NEG	✔	✔	求补码
循环移位	30	ROR	✔	✔	循环右移
	31	ROL	✔	✔	循环左移
	32	RCR	✔	✔	带进位循环右移
	33	RCL	✔	✔	带进位循环左移
	34	SFTR	✔	✔	位右移
	35	SFTL	-	✔	位左移
	36	WSFR	-	✔	字右移
	37	WSFL	-	✔	字左移
	38	SFWR	-	✔	“先进先出”写入
	39	SFRD	-	✔	“先进先出”读出
数据处理	40	ZRST	-	✔	区间复位
	41	DECO	-	✔	解码
	42	ENCO	-	✔	编码
	43	SUM	✔	✔	ON 位数
	44	BON	✔	✔	ON 位数判定
	45	MEAN	✔	✔	平均值
	46	ANS	-	-	报警器置位
	47	ANR	-	✔	报警器复位
	48	SOR	✔	✔	BIN 平方根
	49	FLT	✔	✔	浮点数与十进制数间转换

分类	FNC NO.	指令符号			指令功能
		16bit	32bit	P 指令	
高速处理	50	REF	-	✔	输入输出刷新
	51	REFE	-	✔	滤波器调整
	52	MTR	-	-	距阵输入
	53	HSCS	✔	-	比较置位（高速计数器）
	54	HSCR	✔	-	比较复位（高速计数器）
	55	HSZ	✔	-	比较区间（高速计数器）
	56	SPD	-	-	脉冲密度
	57	PLSY	✔	-	脉冲输出
	58	PWM	-	-	脉冲幅宽调制
	59	PLSR	✔	-	带加减速的脉冲输出
方便指令	60				
	61	SER	✔	✔	数据搜索
	62	ABSD	✔	-	绝对值式凸轮顺控
	63	INCD		-	增量式凸轮顺控
	64	TIMR	-	-	示教定时器
	65	STMR	-	-	特殊定时器
	66	ALT	-	-	交替输出
	67	RAMP	-	-	斜波信号
	68	ROTC	-	-	旋转台控制
	69	SORT	-	-	列表数据排列
外部设备 I/O	70	TKY	✔	-	0-9 数字键输入
	71	HKY	✔	-	16 键输入
	72	DSW	-	-	数字开关
	73	SEGD	-	✔	7 段编码
	74	SEGL	-	-	带锁存的 7 段显示
	75	ARWS	-	-	矢量开关
	76	ASC	-	-	ASCII 码转换
	77	PR	-	-	ASCII 码打印输出
	78	FROM	✔	✔	特殊功能块读出
	79	TO	✔	✔	特殊功能块写入
外设设备 SER	80	RS	-	-	串行数据传送
	81	PRUN	✔	✔	并联运行
	82	ASCI	-	✔	HEX→ASCII 转换
	83	HEX	-	✔	ASCII→HEX 转换
	84	CCD	-	✔	校正代码
	85				
	86				
	87				
	88	PID	-	-	PID 运算
	89				
浮点数	110	ECMP	✔	✔	2 进制浮点数比较
	111	EZCP	✔	✔	2 进制浮点数区间比较
	118	EBCD	✔	✔	2 进制→10 进制浮点数转换
	119	EBIN	✔	✔	10 进制→2 进制浮点数转换

分类	FNC NO.	指令符号			指令功能
		16bit	32bit	P 指令	
	120	EADD	✔	✔	2 进制浮点数加
	121	ESUB	✔	✔	2 进制浮点数减
	122	EMUL	✔	✔	2 进制浮点数乘
	123	EDIV	✔	✔	2 进制浮点数除
	127	ESOR	✔	✔	2 进制浮点数开平方
	129	INT	✔	✔	2 进制浮点数→BIN 整数转换
	130	SIN	✔	✔	浮点数 SIN 运算
	131	COS	✔	✔	浮点数 COS 运算
	132	TAN	✔	✔	浮点数 TAN 运算
	147	SWAP	✔	✔	上下字节变换
定位指令	155	ABS	✔	-	ABS 位置数据读取
	156	ZRN	✔	-	原点回归
	157	PLSV	✔	-	可变脉冲输出
	158	DRVI	✔	-	相对位置控制
	159	DRVA	✔	-	绝对位置控制
时钟运算	160	TCMP	-	✔	时钟数据的比较
	161	TZCP	-	✔	时钟数据区域比较
	162	TADD	-	✔	时钟数据加法
	163	TSUB	-	✔	时钟数据减法
	166	TRD	-	✔	时钟数据读出
	167	TWR	-	✔	时钟数据写入
	168				
	169	HOUR	✔	-	计时表
	170	GRY	✔	✔	格雷码转换
	171	GBIN	✔	✔	格雷码逆转换
接点比较	224	LD=	✔	–	(S1)=(S2)
	225	LD>	✔	–	(S1)>(S2)
	226	LD<	✔	–	(S1)<(S2)
	228	LD<>	✔	–	(S1)<>(S2)
	229	LD<=	✔	–	(S1)<=(S2)
	230	LD>=	✔	–	(S1)>=(S2)
	232	AND=	✔	–	(S1)=(S2)
	233	AND>	✔	–	(S1)>(S2)
	234	AND<	✔	–	(S1)<(S2)
	236	AND<>	✔	–	(S1)<>(S2)
	237	AND<=	✔	–	(S1)<=(S2)
	238	AND>=	✔	–	(S1)>=(S2)
	240	OR=	✔	–	(S1)=(S2)
	241	OR>	✔	–	(S1)>(S2)
	242	OR<	✔	–	(S1)<(S2)
	244	OR<>	✔	–	(S1)<>(S2)
	245	OR<=	✔	–	(S1)<=(S2)
	246	OR>=	✔	–	(S1)>=(S2)

7. 指令解释

7.1 基本指令解释

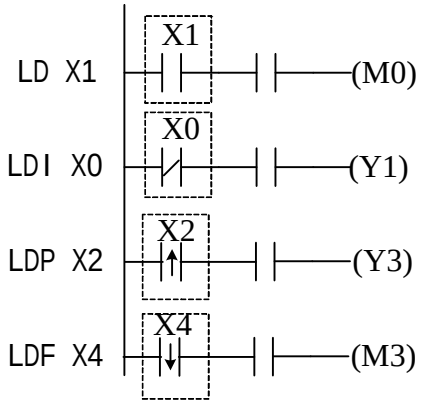
指令	操 作 数						
LD LDI	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
LDP LDF	✔	✔	✔	✔	✔	✔	

LD/LDI/ LDP/LDF 指令用于左母线开始的接点，其中：

LD/LDI指令分别是把A 接点和B 接点的当前能流状态保存，同时把取来的接点状态存入累计缓存器内。

LDP 指令用于取用接点信号的上升沿，若本次扫描中检测到对应信号的上升跳变，则触点有效，下一次扫描时，触点即变成无效。

LDF指令用于取用接点信号的下降沿，若本次扫描中检测到对应信号的下降跳变，则触点有效，下一次扫描时，触点即变成无效。



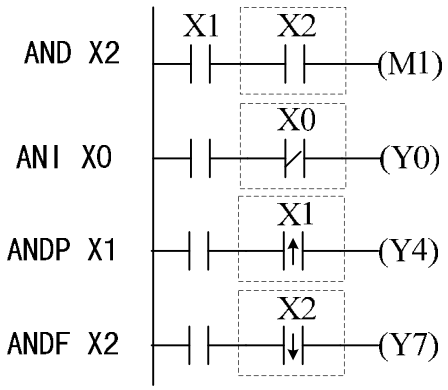
指令	操 作 数						
AND ANI	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
ANDP ANDF	✔	✔	✔	✔	✔	✔	

AND/ANI /ANDP/ANDF 指令用于串联接点的状态运算，其操作是先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“与”（AND）的运算，并将结果存入累计缓存器内。

AND/ANI指令分别是将A 接点和/B 接点的状态参与AND运算；

ANDP指令是将接点的上升沿跳变状态参与AND运算；

ANDF指令是将接点的下降沿跳变状态参与AND运算；



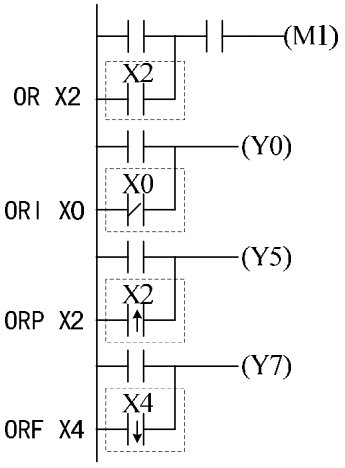
指令	操　作　数						
OR ORI	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
ORP ORF	✔	✔	✔	✔	✔	✔	

OR/ORI 指令用于并联接点的状态运算，其操作是先读取目前所指定接点的状态，再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累计缓存器内。

OR/ORI指令分别是将A 接点和/B 接点的状态参与OR运算；

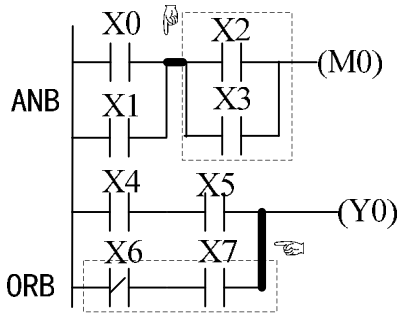
ORP指令是将接点的上升沿跳变状态参与OR运算；

ORF指令是将接点的下降沿跳变状态参与OR运算。



指令	操　作　数
ANB ORB	无。 参与块运算的是最近两次LD（或LDI/LDP/LDF）区间的计算能流

ANB 和ORB 是将前一保存的逻辑结果与目前累计缓存器旳内容作 “与”和“或”的运算。




```
☛LD X0
  OR X1
☛LD X2
  OR X3
☞ANB
  OUT M0
```

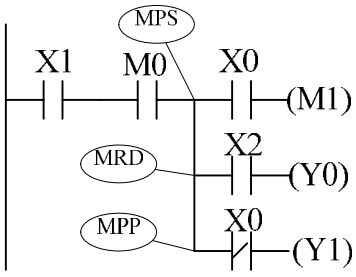
```
☛LD X4
  AND X5
☛LDI  X6
  AND  X7
☞ORB
  OUT Y0
```

指令	操　作　数
MPS　MRD　MPP	无

MPS：将目前累计缓存器的内容存入堆栈。（堆栈指针加一）

MRD：读取堆栈内容存入累计缓存器。（堆栈指针不动）

MPP：自堆栈取回前一保存的逻辑运算结果，存入累计缓存器。（堆栈指针减一）

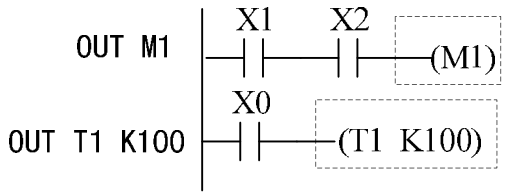



```
LD X1
AND M0
MPS
AND X0
OUT M1
MRD
```

```
AND X2
OUT Y0
MPP
AND X0
OUT Y1
```

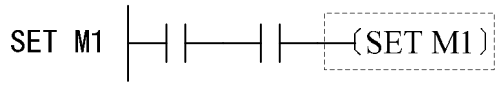
指令	操　作　数						
OUT	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
	✘	✔	✔	✔	✔	✔	

将OUT 指令之前的逻辑运算结果输出至指定的元件。



指令	操　作　数						
SET	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
		✔	✔	✔			

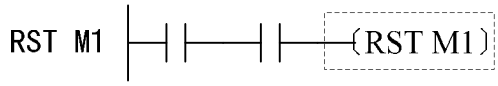
当SET 指令被驱动，其指定的组件被设定为ON，且被设定的组件会维持ON，不管SET 指令是否仍被驱动。可利用RST 指令将该组件设为OFF。



指令	操　作　数							
RST	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255	V,Z
		✔	✔	✔	✔	✔	✔	✔

当RST 指令被驱动，其指定的组件被设定为OFF，且被设定的组件会维持OFF，不管RST 指令是否仍被驱动。可利用SET 指令将该组件设为ON。

RST指令也可用于D、V、Z变量复位，将指定D、V、Z元件的值清为0。



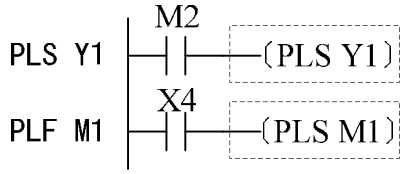
元件	操作结果
S，M，Y	线圈及接点被设定为OFF
T，C	目前计时或计数值会被设为0，且线圈及接点被设定为OFF。
D，V，Z	元件的值清为0

指令	操　作　数						
PLS PLF	X0~ X377	Y0~ Y377	M0~M3071	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
		✔	✔				

当PLS 指令被上升沿驱动时，其指定的元件被设定为ON状态，该ON状态仅持续1个扫描周期；

当PLF 指令被下降沿驱动时，其指定的元件被设定为ON状态，该ON状态仅持续1个扫描周期。

程序举例：



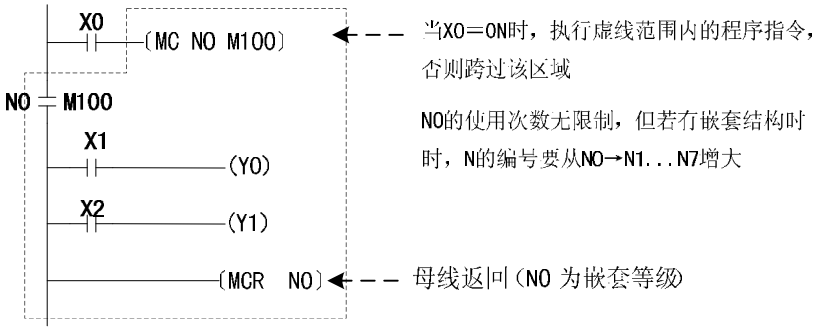
指令	操　作　数
MC　MCR	N0~N7

MC 为主控起始指令，当MC 指令执行时，位于MC 与MCR 指令之间的指令照常执行。当MC 指令OFF 时，位于MC 与MCR 指令之间的指令动作如下所示：

定时器	计时值归零，线圈失电，接点不动作
计数器	线圈失电，计数值及接点保持目前状态
OUT 指令驱动的线圈	全部不受电
SET，RST 指令驱动的组件	保持目前状态
应用命令	全部不动作

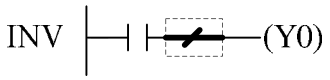
MCR 为主控结束指令，置于主控程序最后，在MCR 指令之前不可有接点指令。

MC-MCR 主控程序指令支持巢状程序结构，最多可8 层，使用时依N0~N7 的顺序，



指令	操　作　数
INV	无

将INV 指令之前的逻辑运算结果反相后存入累计缓存器内。当INV指令之前能流为ON，经过INV后能流变为OFF；反之，变为ON。



指令	操　作　数
NOP	无

指令NOP 在程序不做任何运算，因此执行后仍会保持原逻辑运算结果，没有实际操作，在AutoShop编译时，会自动将之删除，减少程序空间的浪费，加快运行速度。

指令	操　作　数
FEND END	无

在主程序的末尾才加入 FEND 指令，以指明用户主程序的结束，PLC 执行时由用户程序的地址0 扫描到END 指令，执行之后，返回到地址0 重新作扫描执行。

只在梯形图程序或指令程序最后才加入 END 指令。PLC 执行时对超过END 指令之后的程序空间不再处理。

在AutoShop编程环境中，无需用户输入FEND或END指令，系统在下载时会自动加入。

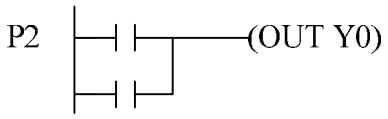
指令	操　作　数
P	P0~P127 1）用于标记主程序中跳转的地址起始，其中P63为指向END的专用地址。 2）用于标记子程序的起始地址，每个子程序都以SRET为结束。
I	I00*～I50*,6 点，输人中断指针； I6**～I8**,3点，定时中断指针； I010～I060,6点，计数中断指针

指针 (P)

指针P 用于跳跃指令CJ 及子程序呼叫指令CALL，使用不须从编号0 开始，但是编号不能重复使用，否则会发生不可预期的错误。使用时机如下所示：

1. 使用于指令CJ，指示程序执行跳跃的目的地址，并在目标程序的开头输入同编号的指针P。如下所示：

2. 使用于指令CALL，指示子程序的目的地址，并在子程序的开头输入同编号的指针P。如下所示：



7.2 应用指令解释

16bit	32bit	P	FNC	**CJ** P***	条件跳转
✔		✔	00	**CJP** P***	
3 步				P000~P127	

1）当能流有效时，程序自动从CJ （或CJP）指令的地址跳转至由P***指定的地址后继续执行，中间地址的程序指令被跳过，不予执行；

2）当能流无效时，程序继续往下执行，此时CJ （或CJP）指令不被执行。

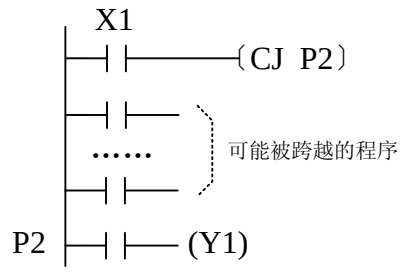
若被跨越的中间地址区的程序中有 TMR 定时器或计数器，且已被驱动，则动作反应为：

执行情况	CJ 有跳转	CJ 无跳转
T192~T199	正常执行	正常执行
其他定时器	停止计时	
C235~C255	正常执行	
其他计数器	停止计数	

对 P***地址指针的要求如下：

- 由 CJ （或 CJP）引用的地址指针，必须在主程序结束（FEND 指令）之前的范围；
- P63 特指 END 的地址，不要定义到其它程序步；
- 与子程序不同，P***开始的程序后不需要 SRET 语句作为结束；
- P***的定义地址不要有重复；
- 注意跳转的条件，防止死循环或程序运行超时。

程序举例：



16bit	32bit	P	FNC	**CALL** P***	子程序调用
✔		✔	01	**CALLP** P***	
3 步				P000~P127	

当能流有效时，程序调用由 P***指定的子程序。子程序执行完毕，会返回到该 CALL（或 CALLP）语句的下一指令， 继续执行后续语句。

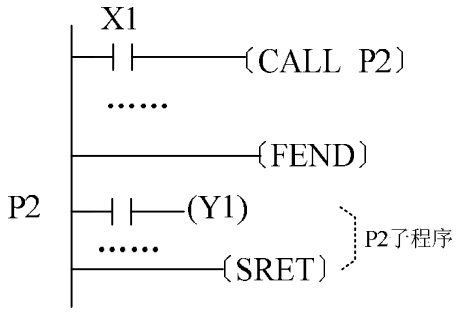
对 P***地址指针的要求如下：

- 由 P***开始的子程序，必须在主程序结束（FEND 指令）之后的范围。在 AutoShop 环境中，子程序为单独编辑，无此限制，但不得位于主程序中；
- 子程序必须以 SRET 语句结束；
- P***的子程序可被多处调用，也可被其他子程序调用，但嵌套层数不得超过 5 层；
- 在子程序内不得调用自身，防止死循环或程序运行超时。
- 在子程序中，可采用 T192~T199 或 T246 ~T 249 作为定时器。

16bit	32bit	P	FNC	**SRET**	子程序完毕。
✔			02		
3 步				无需触点驱动，无操作数的单独指令。	

SRET语句位于子程序的结束处，执行了该指令后，会退回到调用该子程序的语句处，继续随后的程序执行。

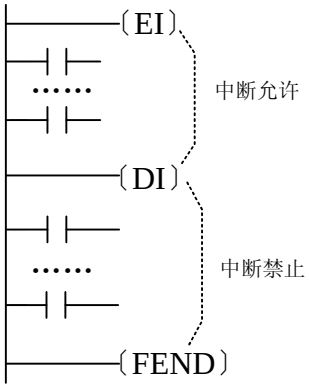
程序举例：



16bit	32bit	P	FNC 03	**IRET**	中断程序完毕	无需触点驱动，无操作数的单独指令。
✔			FNC 04	**EI**	中断允许	
1 步			FNC 05	**DI**	中断禁止	

IRET语句位于中断子程序的结束处，执行了该指令后，会返回到调用该中断子程序前的语句处，继续程序执行。

PLC程序开始运行时，默认为中断禁止状态；执行了EI语句后，中断功能允许；当中断为允许状态，执行了DI语句后，即进入中断禁止状态。



中断的种类与设置：

1）外部信号输入中断：可定义X0~X5输入信号的上升沿或下降沿进行中断，对于不需要即时响应的X信号，还可以采用脉冲捕捉的功能；

2）　定时器中断：以10ms～99ms的固定周期发生的中断，通过设置对M8059标志的设置，可禁止所有计数器的中断。

外部信号输入中断指针与设置：

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

中断子程序选用不同的编号，即选择了不同的端口及中断触发沿；对于同一X输入，不能同时对上升中断和下降中断编号。对于一个X输入端口，只能使用一种触发沿，触发沿通过　来设定；如果在对M8050-M8055编号过程中"ON"，则禁止了对应X端口的中断功能。

中断的编程规定与执行特性：

- 在DI-EI 指令间（中断禁止区间）发生中断，亦能对其记忆并在EI指令后执行。
- 中断子程序必须写在FEND指令之后，子程序尾部必须以IRET结束，在AutoShop环境内，不要写在主程序中，子程序尾可省略IRET；
- 指针编号不能重复使用；
- 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。但若在中断子程序内对 EI、DI 编程，可以接受最多为二重的中断。
- 在中断处理过程中控制输入继电器及输出继电器时，使用输入输出刷新指令（REFF），可以通过读取最新的输入状态、或者立即输出运算结果，实现高速控制；
- 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（FNC56 ）］等的应用命令的编号相重复。
- 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 1 ms 累计定时器时亦需加注意；
- 如果指定输入中断指针 I 口 0 口，则输入继电器的输入滤波特性自动关闭。因此，无需采用 REFE (FNC51)指令及特殊数据寄存器 D8020(输入滤波器调整)。另外，不作为输入中断指针用的输入继电器的输入滤波器能维持 10ms(初始值)。

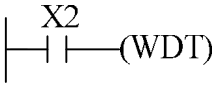
16bit	32bit	P	FNC	**FEND**	主程序完毕。
✔			06		
1 步				无需触点驱动，无操作数的单独指令。	

FEND语句位于主程序的结束处，执行了该指令后，即结束了本次用户程序的扫描。

CALL 命令调用的子程序必须写在FEND 命令后，并且在该子程序结束加上SRET 命令；中断子程序也必须写在FEND 之后，并在该中断程序结束加上IRET 指令。在AutoShop中，子程序或中断程序必须单独编写，在程序的末尾可不必加SRET或IRET。

16bit	32bit	P	FNC	**WDT**	监视定时器复位。
✔		✔	07		
1 步				无操作数的单独指令。	

PLC系统内有用于监视用户程序执行一次的时间是否超时的定时器，若超时即会停止用户程序的执行并报警，执行WDT指令即可将该监视定时器复位，让监视定时器重新开始计时，避免超时错误。



若用户程序所执行的操作过于复杂（例如过多的循环计算），执行时有可能出现运行超时错误，编程时若必要，可用WDT指令；

另外，监视定时器的超时判断默认值为200ms，也可根据需要修改D8000的设定值。

16bit	32bit	P	FNC 08	**FOR**	循环范围开始	无需触点驱动，无操作数的单独指令。
✔						
3 步			指令格式：**FOR** (S1)			

FOR指令用于一个循环的起始，同时指明循环执行的次数，必须与NEXT指令配套使用。其中：

(S1) 为循环次数控制变量。

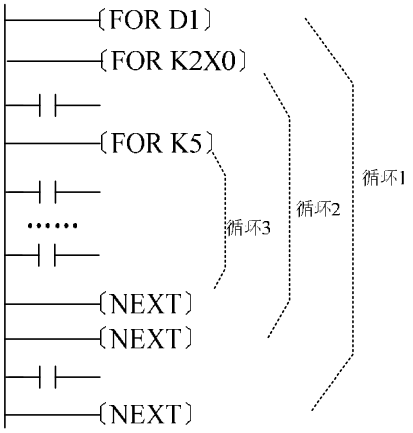
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

参见NEXT指令的解释与举例。

16bit	32bit	P	FNC 09	**NEXT**	循环范围结束	无需触点驱动，无操作数的单独指令。
✔						
3 步			指令格式：**NEXT** (无操作数)			

NEXT指令用于指示循环区域的尾部。由FOR 指令指定FOR~NEXT 循环来回执行N 次后跳出FOR~NEXT 循环往下继续执行。

在FOR~NEXT 指令的循环区间，可以嵌入另一个FOR~NEXT循环，但规定：从最外层的FOR~NEXT计算，最多可内嵌4层FOR~NEXT 循环。运行时PLC会以各FOR~NEXT层对应解析执行。



有下列情况者，都会出错：

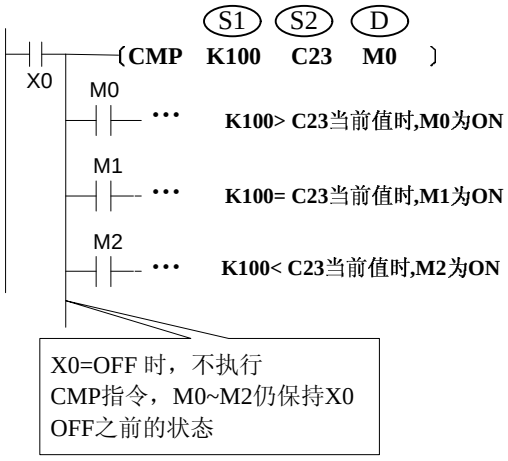
- NEXT 指令在 FOR 指令之前；
- 有 FOR 指令而无 NEXT 指令；
- 在 FEND,END 指令以后有 NEXT 指令；
- FOR 指令与 NEXT 指令个数不一致等。

16bit	32bit	P	FNC 10	**CMP**	两个数比较
✔	✔	✔			
7 步	13 步	13 步	指令格式：**CMP** (S1) (S2) (D)		

本指令完成两个操作变量的比较，将比较结果输出给指定的位变量，操作数均按有符号数进行代数比较操作。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔	✔	✔											

其中 (D) 会占用 3 个连续地址的位变量，参见下例：



16bit	32bit	P	FNC 11	**ZCP**	区间比较
✔	✔	✔			
7 步	13 步	13 步	指令格式：　**ZCP** (S1) (S2) (S) (D)		

需要触点驱动，有 4 个操作变量。当控制能流有效时，按有符号数进行代数比较操作，以 (S1) (S2) 为区间，将 (S) 的值位于该区间的位置作为结果，存入 (D) 为起始地址的 3 个连续位变量中。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔	✔	✔											

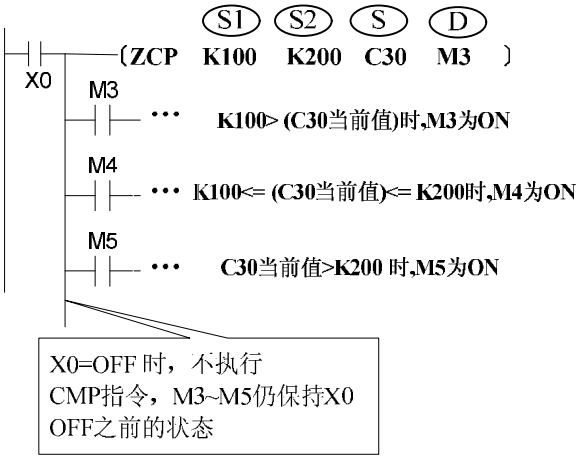
其中：

(S1) 为比较区间的下限；

(S2) 为比较区间的上限；

(S) 为比较变量；

(D) 为比较结果存储单元，会占用 3 个连续地址的位变量，参见下例：



16bit	32bit	P	FNC 12	**MOV**	数据移动
✔	✔	✔			
5 步	9 步	9 步	指令格式： **MOV** (S) (D)		

需要触点驱动，有 2 个操作变量，将 (S) 的值复制到 (D) 中。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

当为32bit指令（**DMOV**）时，(S) 和 (D) 都会使用相邻高地址的变量单元参与运算。例如语句：〔**DMOV D1 D5**〕的操作结果是：**D1→D5；D2→D6**

16bit	32bit	P	FNC 13	**SMOV**	移位传送
✔		✔			
11 步		11 步	指令格式： **SMOV** (S) (m1) (m2) (D) (n)		

需要触点驱动，最多有 5 个操作变量，其中：

Ⓢ为待复制的数据源变量；

ⓜ1为数据源传送的起始位号，这里的“位”是指 BCD 格式的位，（1～4）范围；

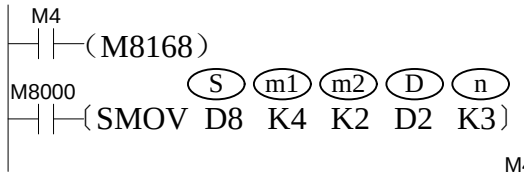
ⓜ2为数据源传送的位数，这里的“位”也是指 BCD 格式的位，（1～m1）范围；

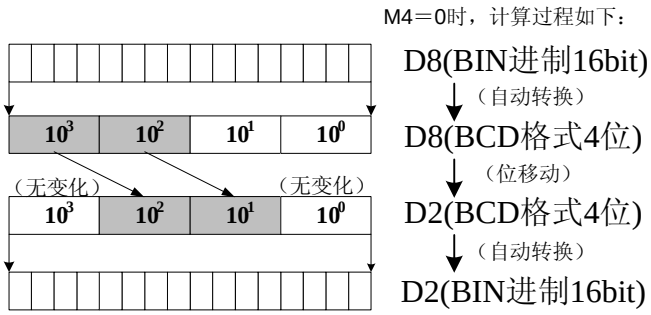
Ⓓ为数据源传送的目的变量；

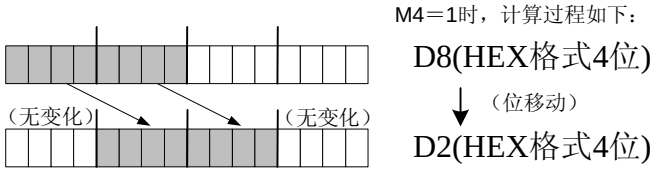
ⓝ为数据源传送的目的变量的起始位，这里的“位”也是指 BCD 格式的位，（m2～4）范围。

操作数	位元件				字元件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ							✔	✔	✔	✔	✔	✔	✔	✔	✔
ⓜ1					✔	✔									
ⓜ2					✔	✔									
Ⓓ								✔	✔	✔	✔	✔	✔	✔	✔
ⓝ					✔	✔									

数据位的传送过程与特殊标志M8168的状态有关，参见下例：







16bit	32bit	P	FNC 14	**CML**	数据取反传送
✔	✔	✔			
5 步	9 步	9 步	指令格式： **CML** (S) (D)		

需要触点驱动，有 2 个操作变量，将(S)的 BIN 值逐位取反后复制到(D)中。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

当(D)的位数不足16bit时，将(S)取反后按低位对齐传送到(D)变量中；

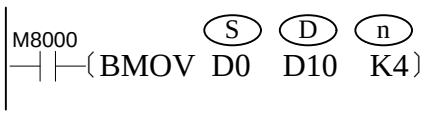
当为32bit指令（**DMOV**）时，(S)和(D)都会使用相邻高地址的变量单元参与运算。例如语句：〔**DCML D1 D5**〕的操作结果是：**/D1→D5；/D2→D6**

16bit	32bit	P	FNC 15	**BMOV**	数据成批传送
✔		✔			
7 步		7 步	指令格式： **BMOV** (S) (D) (n)		

需要触点驱动，有 3 个操作变量，将由(S)指定起始地址的(n)个变量值复制到由(D)指定起始地址的(n)个单元中。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，n＝1～512														

其中(n)的取值范围是1～512。

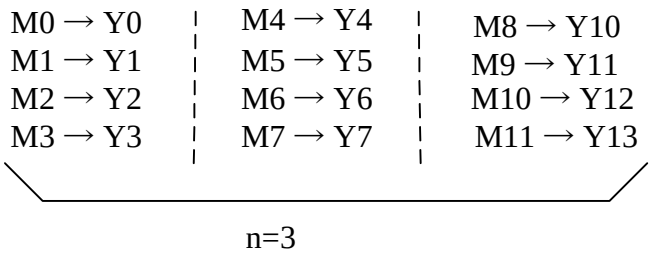


完成的操作是：
D0 → D10
D1 → D11
D2 → D12
D3 → D13

当操作数为位元件时，(S)和(D)位数必须相等。

```
M8000                  (S)    (D)    (n)
─┤├─{BMOV  K1M0    K1Y0    K3}
```

完成的操作是：



当特殊变量 M8024＝1 时，成批传送的方向相反，即将由(D)指定起始地址的(n)个变量值复制到由(S)指定起始地址的(n)个单元中。

16bit	32bit	P	FNC 16	**FMOV**	数据多点传送
✔	✔	✔			
7 步	13 步	13 步	指令格式： **FMOV** (S) (D) (n)		

需要触点驱动，有 3 个操作变量，将由(S)的数据复制到由(D)指定起始地址的(n)个单元中。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，n＝1～512														

其中(n)的取值范围是1～512。

16bit	32bit	P	FNC 17	**XCH**	数据交换
✔	✔	✔			
5 步	9 步	9 步	指令格式： **XCH** (S) (D)		

需要触点驱动，有 2 个操作变量，将(S)和(D)的值彼此交换。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)								✔	✔	✔	✔	✔	✔		

(D)								✔	✔	✔	✔	✔	✔		

当特殊变量M8160＝1时，且(D)与(S)为同一地址，完成的操作将是高8位与低8位的交换，相当于SWAP指令的操作。

16bit	32bit	P	FNC 17	**BCD**	BCD 交换
✔	✔	✔			
5 步	9 步	9 步	指令格式： **BCD** (S) (D)		

需要触点驱动，有 2 个操作变量，将(S)的值进行 BCD 变换后存入(D)中。该指令常用于将数据显示前的数据格式处理。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)								✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

使用16bit指令，当转换结果超过9999时会出错；使用32bit指令，当转换结果超过99999999时会出错。

16bit	32bit	P	FNC 17	**BIN**	BIN 交换
✔	✔	✔			
5 步	9 步	9 步	指令格式： **BIN** (S) (D)		

需要触点驱动，有 2 个操作变量，将(S)的值进行 BIN 变换后存入(D)中。该指令常用于将外部端口读入数据（如编码盘设置）处理成能直接用于运算的 BIN 格式。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)								✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

转换前(S)中的每一个BCD位的值都须为0～9范围，否则会出错。

16bit	32bit	P	FNC 20	**ADD**	BIN 加法运算
✔	✔	✔			
7 步	13 步	13 步	指令格式： **ADD** (S1) (S2) (D)		

需要触点驱动，有 3 个操作变量，将(S1)和(S2)的值进行 BIN 代数相加后存入(D)中，

参与运算的变量都按有符号数处理，最高位为符号位，0 为正数，1 为负数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

若计算结果为0，则0标志(M8020)会置位；

若计算结果超过32,767（16bit运算）或-2,147,483,647（32bit运算）时，进位标志(M8021)会置位；

若计算结果不满-32,768（16bit运算）或-2,147,483,648（32bit运算）时，借位标志(M8022)会置位；

进行32bit运算时，指令中变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

16bit	32bit	P	FNC 21	**SUB**	BIN 减法运算
✔	✔	✔			
7 步	13 步	13 步	指令格式：　**SUB** (S1) (S2) (D)		

需要触点驱动，有 3 个操作变量，将(S1)和(S2)的值进行 BIN 代数相减后存入(D)中，参与运算的变量都按有符号数处理，最高位为符号位，0 为正数，1 为负数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

若计算结果为0，则0标志(M8020)会置位；

若计算结果超过32,767（16bit运算）或-2,147,483,647（32bit运算）时，进位标志(M8021)会置位；

若计算结果不满　-32,768（16bit运算）或-2,147,483,648（32bit运算）时，借位标志(M8022)会置位；

进行32bit。

进行32bit运算时，指令中变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

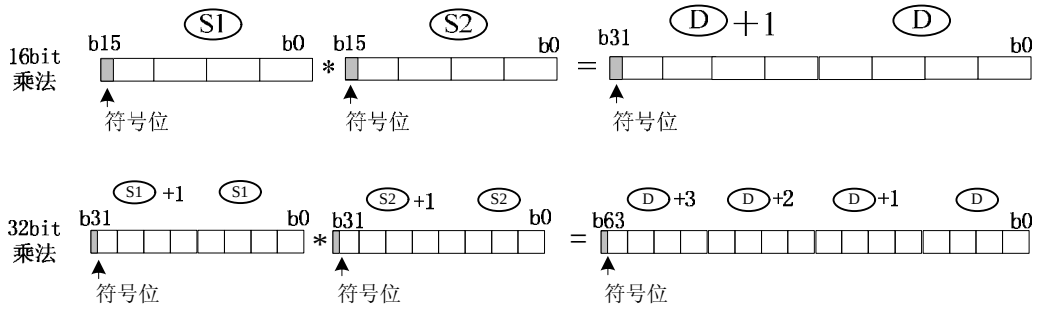
16bit	32bit	P	FNC 22	**MUL**	BIN 乘法运算
✔	✔	✔			
7 步	13 步	13 步	指令格式： **MUL** (S1) (S2) (D)		

需要触点驱动，有 3 个操作变量，将(S1)和(S2)的值进行 BIN 代数相乘后存入(D)中，参与运算的变量都按有符号数处理，最高位为符号位，0 为正数，1 为负数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔		

表中V、Z元件仅在16bit运算时可用。

进行32bit运算时，指令中变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖；计算的结果只能为32bit，对于超出32bit范围的计算，最好采用浮点运算指令EMUL进行计算。



16bit	32bit	P	FNC 23	**DIV**	BIN 除法运算
✔	✔	✔			
7 步	13 步	13 步	指令格式： **DIV** (S1) (S2) (D)		

需要触点驱动，有 3 个操作变量，将被除数(S1)和除数(S2)的值进行 BIN 代数相乘后存入(D)中，参与运算的变量都按有符号数处理，最高位为符号位，0 为正数，1 为负数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔		

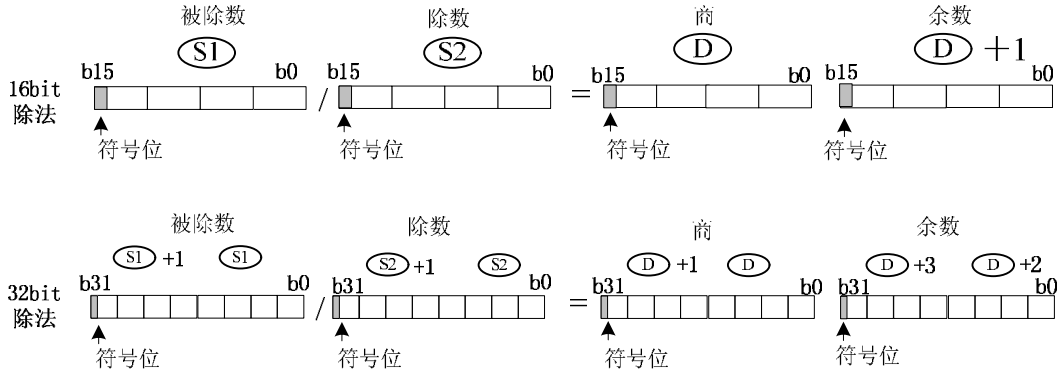
表中V、Z元件仅在16bit运算时可用。

进行32bit运算时，指令中 (S1) 和 (S2) 变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖；计算所得的商存入 (D) 所指单元，余数存入 (D) 相邻高地址单元中。

若除数 (S2) 为0，会发生计算错误；

若将位元件（KnX/KnY/KnM/KnS）指定为 (D)，不能得到余数；

若被除数为负数，余数即为负数。



16bit	32bit	P	FNC 24	**INC**	BIN 加 1 运算
✔	✔	✔			
3 步	5 步	5 步	指令格式： **INC** (D)		

指令每执行一次， (D) 中的数值增加 1。

16bit	32bit	P	FNC 25	**DEC**	BIN 减 1 运算
✔	✔	✔			
3 步	5 步	5 步	指令格式： **DEC** (D)		

指令每执行一次， (D) 的数值减 1。

增减计算都按无符号方式进行，对 0 标志、进位、借位标志都不刷新。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)								✔	✔	✔	✔	✔	✔	✔	✔

进行32bit运算时，指令中 (D) 变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

16bit	32bit	P	FNC 26	**WAND**	逻辑与
✔	✔	✔			
7 步	13 步	13 步	指令格式：　**WAND** (S1) (S2) (D)　　(32bit 指令符为 **DAND**)		

本指令执行时，将(S1)和(S2)中 BIN 值的各位对应作“逻辑与”运算，将结果存入(D)变量。

16bit	32bit	P	FNC 27	**WOR**	逻辑或
✔	✔	✔			
7 步	13 步	13 步	指令格式：　**WOR** (S1) (S2) (D)　　(32bit 指令符为 **DOR**)		

本指令执行时，将(S1)和(S2)中 BIN 值的各位对应作“逻辑或”运算，将结果存入(D)变量。

16bit	32bit	P	FNC 28	**WXOR**	逻辑异或
✔	✔	✔			
7 步	13 步	13 步	指令格式：　**WXOR** (S1) (S2) (D)　　(32bit 指令符为 **DXOR**)		

本指令执行时，将(S1)和(S2)中 BIN 值的各位对应作“逻辑异或”运算，将结果存入(D)变量。

上述三个指令的操作数适用变量类型如下表，当为 32bit 指令时，寄存器变量则占用后续相邻地址的共 2 个单元：

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

16bit	32bit	P	FNC 29	**NEG**	求补运算
✔	✔	✔			
7 步	13 步	13 步	指令格式：　**NEG** (D)		

需要触点驱动，有 1 个操作变量。将(D)的数值逐位取反、再加 1，存回(D)中。

使用 NEG 指令，可得到与负的 BIN 值相对应的绝对值。

16bit	32bit	P	FNC 30	**ROR**	循环右移
✔	✔	✔			
5 步	9 步	9 步	指令格式： **ROR** (D) (n)		

将(D)的内容循环右移(D)位。

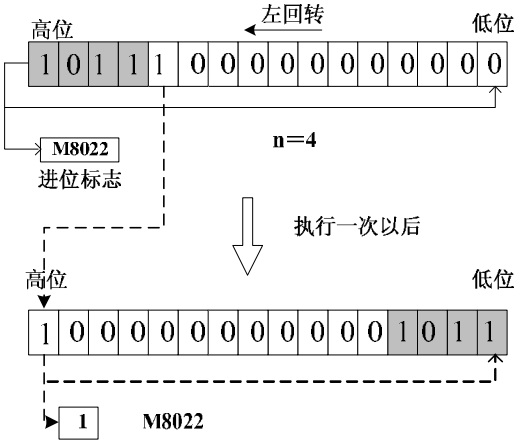
16bit	32bit	P	FNC 30	**ROL**	循环左移
✔	✔	✔			
5 步	9 步	9 步	指令格式： **ROL** (D) (n)		

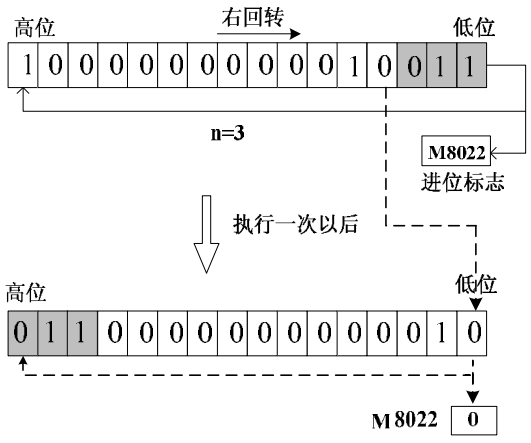
将(D)的内容循环左移(D)位。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(n)	常数，n=1~16（16bit）；n=1~32（32 bit）														

若(D)中指定KnY、KnM、KnS时，只有K4（16bit）及K8（32 bit）有效；

循环移动的最终位被存入进位标志中。





16bit	32bit	P	FNC 32	**RCR**	带进位循环右移
✔	✔	✔			
5 步	9 步	9 步	指令格式： **RCR** (D) (n)		

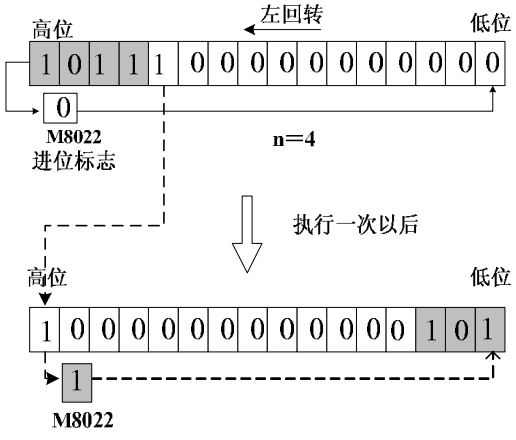
将(D)的内容带进位循环右移(D)位。

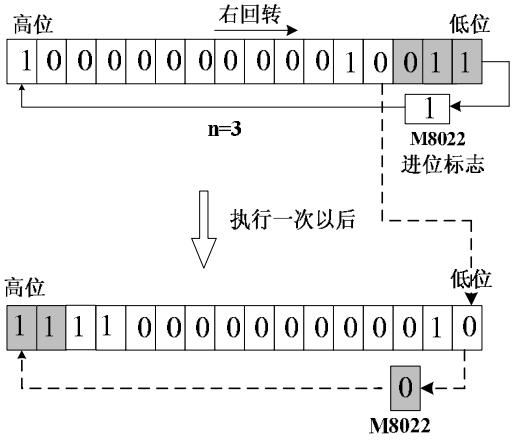
16bit	32bit	P	FNC 33	**RCL**	带进位循环左移
✔	✔	✔			
5 步	9 步	9 步	指令格式： **RCL** (D) (n)		

将(D)的内容带进位循环左移(D)位。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(n)	常数，n=1~16（16bit）；n=1~32（32 bit）														

若(D)中指定KnY、KnM、KnS时，只有K4（16bit）及K8（32 bit）有效；





16bit	32bit	P	FNC 34	**SFTR**	位右移
✔		✔			
7 步		7 步	指令格式： **SFTR** (S) (D) (n1) (n2)		

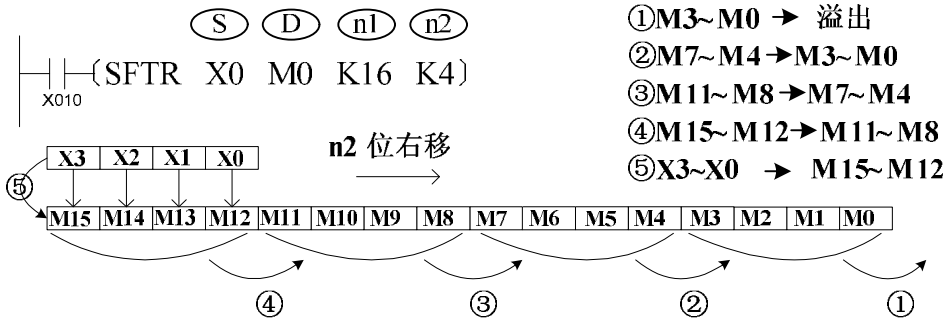
对于位变量，将(S)地址起始的(n2)位变量与(D)地址起始的(n1)变量，按向右方向移动(n2)位后，将结果保存在(D)中。

16bit	32bit	P	FNC 35	**SFTL**	位左移
✔		✔			
7 步		7 步	指令格式： **SFTL** (S) (D) (n1) (n2)		

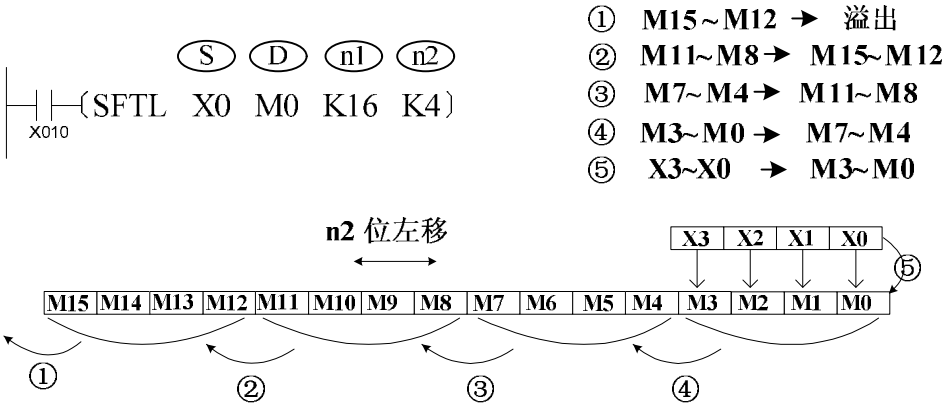
对于位变量，将(S)地址起始的(n2)位变量与(D)地址起始的(n1)变量，按向左方向移动(n2)位后，将结果保存在(D)中。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)		✔	✔	✔											
(D)	✔	✔	✔	✔											
(n1)	常数，n1≤1024														
(n2)	常数，n2≤n1														

SFTR命令举例：



SFTL命令举例：



16bit	32bit	P	FNC 36	**WSFR**	字右移
✔		✔			
9 步		9 步	指令格式： **WSFR** (S) (D) (n1) (n2)		

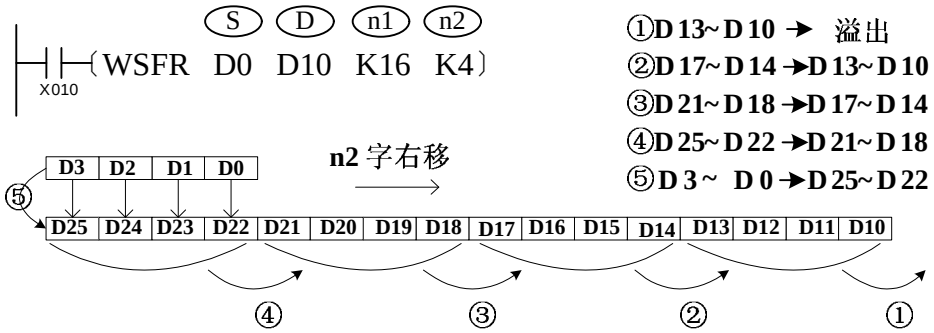
以字为单位，将(S)地址起始的(n2)字变量与(D)地址起始的(n1)字变量，按向右方向移动(n2)个字

16bit	32bit	P	FNC 37	**WSFL**	字左移
✔		✔			
9 步		9 步	指令格式： **WSFL** (S) (D) (n1) (n2)		

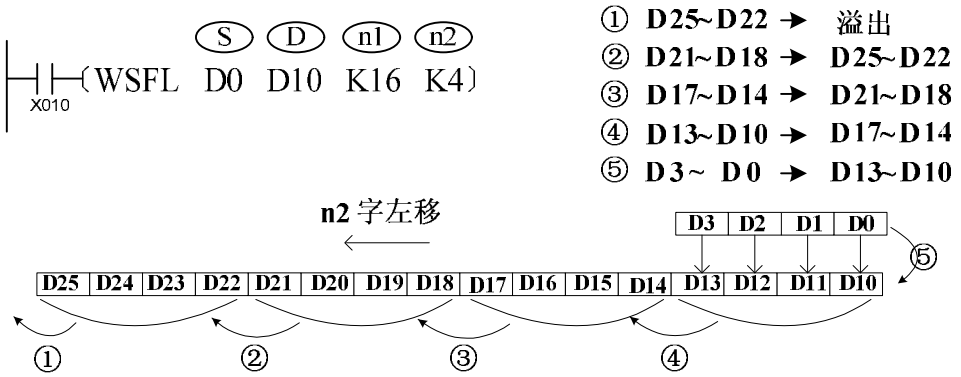
以字为单位，将(S)地址起始的(n2)字变量与(D)地址起始的(n1)字变量，按向左方向移动(n2)个字。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n1)	常数，n1≤512														
(n2)	常数，n2≤n1														

WSFR命令举例：



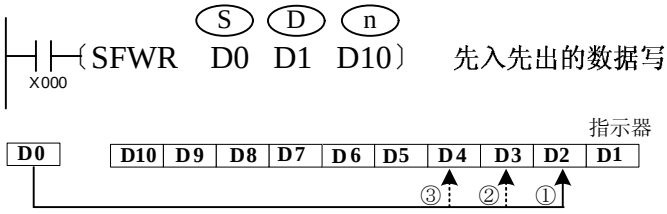
WSFL命令举例：



16bit	32bit	P	FNC 38	**SFWR**	“先进先出”写入
✔		✔			
7 步		7 步	指令格式： **SFWR** S D n		

将 S 的值写入由 D 地址起始，个数为 n 的“先进先出”队列中

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✔	✔	✔	✔	✔	✔	✔	✔	✔		
D								✔	✔	✔	✔	✔	✔		
n	常数，2≤n≤512														



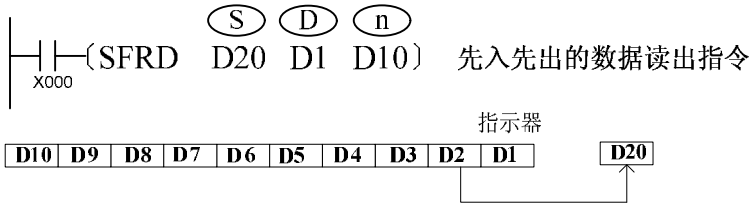
当 X0＝1 时，D0 的内容被存入 D2，D1 的内容变为 l。当 X0 再次从 OFF→ON 时，这个 D0 的内容被存入 D3, D1 的内容变为 2。连续执行型指定中，每一个扫描周期都执行依次保存。

若 Dl 的内容超过 n - 1，则指令不处理，而进位标志 M8022 会置 1。

16bit	32bit	P	FNC 39	**SFRD**	“先进先出”读出
✔		✔			
7 步		7 步	指令格式： **SFRD** S D n		

从“先进先出”队列 S 的首项读出到 D 中，然后将队列 S 逐字右移1个字，将队列指针递减。若指针已经为0，则指令不处理前述操作，而0标志M8020会置1

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，2≤n≤512														



16bit	32bit	P	FNC 40	**ZRST**	区间复位
✔		✔			
5 步		5 步	指令格式：　**ZRST**　(D1)　(D2)		

将(D1)至(D2)区间的变量全部清0 。(D1)和(D2)可指定字变量，也可为Y、M、S位变量。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D1)		✔	✔	✔							✔	✔	✔		
(D2)		✔	✔	✔							✔	✔	✔		

其中要求：

(D1)和(D2)必须为同一类型的软元件；

编号(D1)应不大于(D2)，若两者相同时，仅复位指定的软元件；

本指令为16bit，但(D1)和(D2)可指定32bit的计数器，但应同为32bit型或同为16bit型；

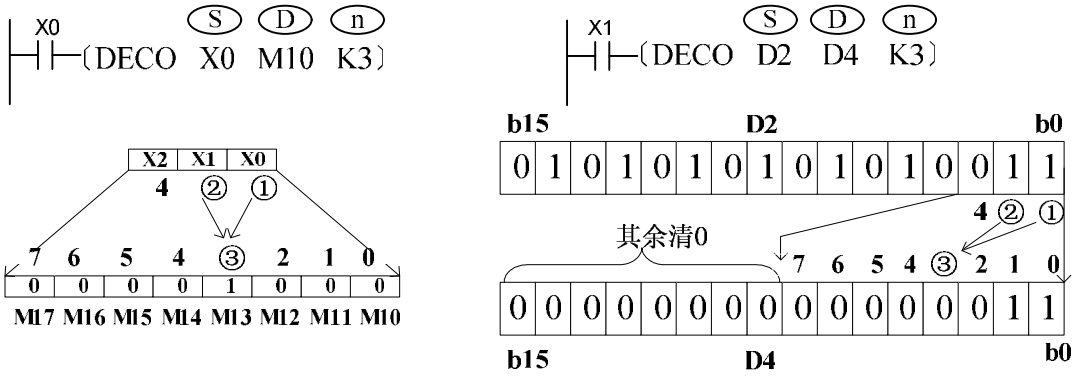
16bit	32bit	P	FNC 41	**DECO**	解码
✔		✔			
7 步		7 步	指令格式：　**DECO**　(S)　(D)　(n)		

计算(S)的最后(2^(n))位的值，作为bit位指针，将(D)的对应位置1，其他位清0。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✔	✔	✔	✔	✔	✔					✔	✔	✔	✔	✔
(D)		✔	✔	✔							✔	✔	✔		
(n)	常数，n=1~8。若 n＝0，指令不执行；其他值则执行出错。														

- 源地址的低 n 位（n≤4）被解码至目标地址。n≤3 时，目标的高位都转为 0；
- n = 0 时不命令不执行， n = 0~8 以外时为运算错误；
- n=8 时，如果译码命令Ⓓ为位软元件时，其点数是 256 点。
- 驱动输入为 OFF 时，指令不执行，正在动作的译码输出保持动作。

编程举例：



16bit	32bit	P	FNC 42	**ENCO**	解码
✔		✔			
7 步		7 步	指令格式： **ENCO** Ⓢ Ⓓ ⓝ		

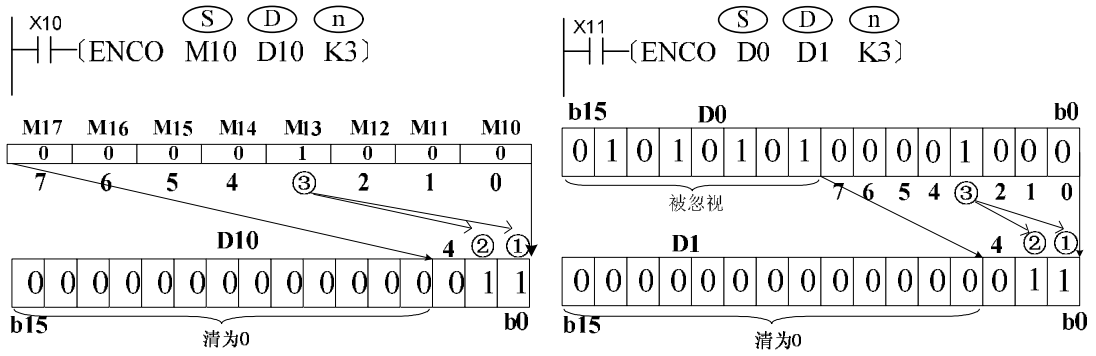
计算Ⓢ的最后ⓝ位的值，作为bit位指针，将Ⓓ的对应位置1，其他位清0。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ	✔	✔	✔	✔	✔	✔					✔	✔	✔	✔	✔
Ⓓ		✔	✔	✔							✔	✔	✔		
ⓝ	常数，n=1~8。若 n＝0，指令不执行；其他值则执行出错。														

源地址内有多个位是 1 时，只计算高位侧；Ⓢ的所有位都为 0 时会出现运算错误；

驱动输入为 OFF 时，指令不被执行，编码输出不变化。

n=8 时，编码指令的Ⓢ如果是位元件，其点数是 256 点。

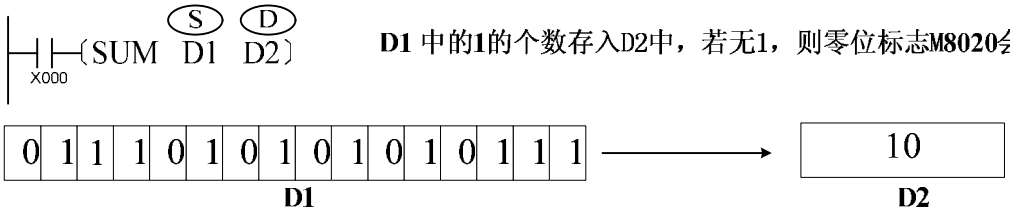


16bit	32bit	P	FNC 43	**SUM**	ON 位数
✔	✔	✔			
5 步	9 步	9 步	指令格式： **SUM** (S) (D)		

计算(S)的BIN进制值中为1的位个数，存入(D)。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

使用D SUM和D SUM P 指令的情况下，（D1，D0）的32位中的1的个数写入D2，同时D3全部为0。

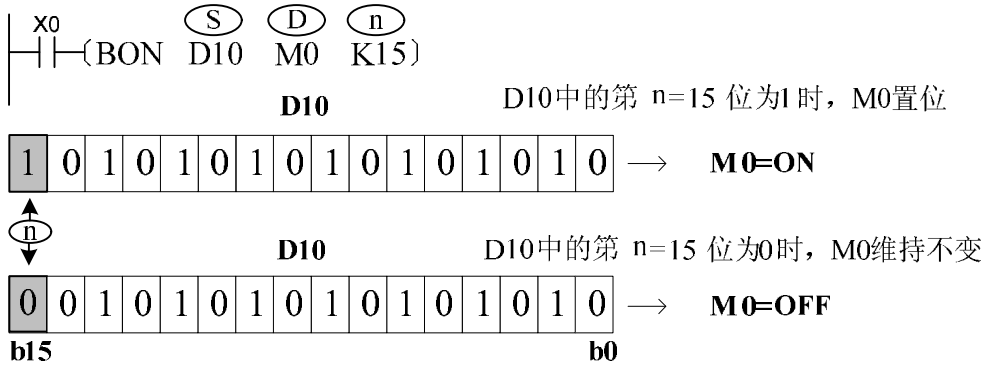


16bit	32bit	P	FNC 44	**BON**	ON 位数
✔	✔	✔			
5 步	9 步	9 步	指令格式： **BON** (S) (D) (n)		

测试(S)的(n)位的状态，结果存入(D)。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔	✔	✔											

(n)	n=0～15（16bit）；n=0～31（32bit）



16bit	32bit	P	FNC 45	MEAN	平均值
✔	✔	✔			
7 步	13 步	13 步	指令格式： **MEAN** (S) (D) (n)		

将由(S)开始的(n)个变量的平均值（先求和，再除以n），存入(D)。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔	✔	✔
(n)	常数，n=1～64，其他值时，计算会出错。														

若计算中有余数，余数将被舍弃；

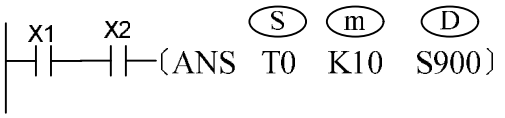
若指定的变量地址超出有效范围，则只有有效的地址部分被处理。

16bit	32bit	P	FNC 46	ANS	报警器置位
✔					
7 步			指令格式： **ANS** (S) (m) (D)		

驱动信号报警器的方便指令。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)											✔				
(D)				✔											
(m)	常数，m=1～32767，（单位为 100ms）。														

其中(D)的范围为S900~S999。例如如下指令：

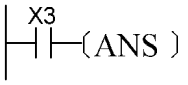


如果 X1 和 X2 同时接通 l 秒以上，则 S900 被置位，以后即使 X1 或 X2 为 OFF 状态，S900 仍保持动作状态。若不满 1 秒， X1 或 X2 变为 OFF 时，定时器复位。

如果预先使 M8049（信号报警器有效）置 ON，则信号报警器 S900~S999 中最小 ON 状态编号被存入 D8049（ON 状态最小编号）另外，当 S900~S999 中任意一个为 ON 时，M8048（报警器动作置 ON）。

16bit	32bit	P	FNC 47	**ANR**	报警器置位
✔		✔			
1 步		1 步	指令格式： **ANS** (无操作数)		

清除报警器信号的方便指令。例如：



如果 X3 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，则复位最新的一个报警点。

若将 X3 再次接通，则下一编号的状态被复位。实际使用中多用 ANSP 指令。

16bit	32bit	P	FNC 48	**SQR**	求平方根
✔	✔	✔			
5 步	9 步	9 步	指令格式： **SQR** (S) (D)		

将(S)按BIN值开平方运算，结果存入(D)。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔							✔		
(D)													✔		

仅在(S)为正数时计算才有效，如负数时运算错误标志 M8067 会工作，指令不被执行；

运算结果舍去小数为整数。舍去时，借位标志 M8021 会动作；

运算结果是0时，零位标志M8020会动作。

16bit	32bit	P	FNC 49	FLT	BIN 整数至浮点数的转换指令
✔	✔	✔			
5 步	9 步	9 步	指令格式： **FLT** Ⓢ Ⓓ		

将整数Ⓢ转换为浮点数，结果存入Ⓓ和Ⓓ＋1单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ													✔		
Ⓓ													✔		

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是 FNC129（INT）。

指令举例：

```
               Ⓢ      Ⓓ
  M8
─┤├─〔FLT   D10   D120〕    当M8=ON时，将16bit数D10中二进制数转换为二进制浮点数后，
                              存放到（D121,D120）
  M10
─┤├─〔FLT   D20   D130〕    当M10=ON时，将32bit数（D11,D10）中二进制数转换为二进制浮点
                              数后，存放到（D131,D130）
```

16bit	32bit	P	FNC 50	REF	输入输出端口状态刷新
✔		✔			
5 步		5 步	指令格式： **REF** Ⓓ ⓝ		

将Ⓓ地址开始的ⓝ个元件状态进行立即更新。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓓ	✔	✔													
ⓝ	常数，n=8~256，且为 8 的倍数														

由于PLC访问端口是按字节访问的特性，故要求：

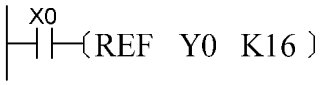
Ⓓ的地址应为X0、X10、…Y0、Y10、...编号元件；

ⓝ的值必须是8的倍数。

正常情况下，输入端口X的状态读取在每次程序扫描执行之前进行，输出端口Y的状态刷新则在每次程序扫描执行之后批次进行，这样IO处理会有一定的延迟。若应用中需要最新的输入信息以及希望立即输出运算结果时，可以使用立即刷新指令REF。

一般 REF 指令用于中断子程序中的高速响应处理，可用在 FOR~NEXT 指令之间、CJ

指令之间等。例如：



执行上述程序时，若 X0 为 ON 状态，会立即将 Y0~Y17 的状态进行刷新。实际的端口状态变化延迟决定于输出元件（如继电器）的响应时间。

16bit	32bit	P	FNC 51	**REFF**	输入滤波调整
✔		✔			
5 步		5 步	指令格式： **REFF** (n)		

将X0~X17输入端口的滤波时间常数设定为(n)。

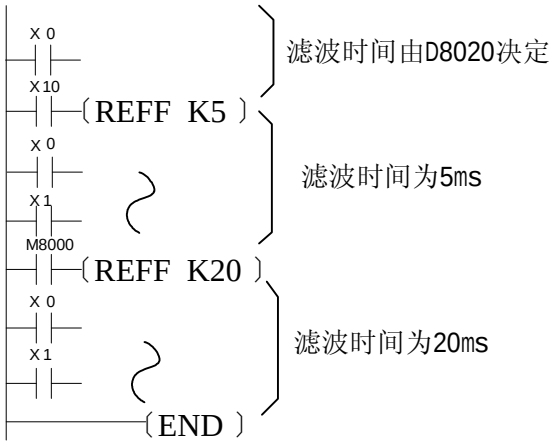
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(n)	常数，n=0~60，单位为 ms														

可编程控制器中，X000~X017 使用了数字滤波器，默认的滤波时间常数由 D8020 设定，通过 REFF 指令可将其值改变为 0~60ms。但实际上该输入设有最小的 C-R 滤波，X0~X5 约为 10μs；X6~X17 约 50μs；

其余的 X 端口则只有硬件 RC 滤波，滤波时间常数约为 10ms，不能修改；

当使用了高速计数器，或 X 输入端中断功能，则相关端口的滤波时间自动为最短时间，无关的端口的滤波时间仍为原设定值。

举例如下：



X10 为 ON 时，将 X000~X017 的输入滤波时间设为 5ms，REFF 指令在每扫描周期执行；X10 为 OFF 时，该指令不执行；

当运行了 REFF 指令后，随后的滤波时间即按刚设定的值处理，直到另一条 REFF 指令的

执行；当运行到 END(或 FEND)后，滤波时间恢复到 D8020 的设定值。

16bit	32bit	P	FNC 52	**MTR**	矩阵输入
✔		✔			
5 步		5 步	指令格式： **MTR** (S) (D1) (D2) (n)		

通过将多个X端口与若干个Y端口组成矩阵输入网络，以扩大输入信号的通道数。其中：

(S)为矩阵输入的硬件X端口的起始地址，要求为X0、X10…等地址号；

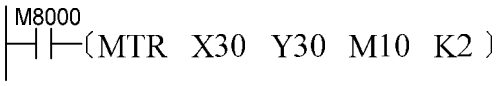
(D1)为矩阵扫描输出的硬件Y端口的起始地址，要求为Y0、Y10…等地址号；

(D2)为矩阵扫描读取状态的存放单元的起始地址，要求为Y0、M0、S0等地址号；

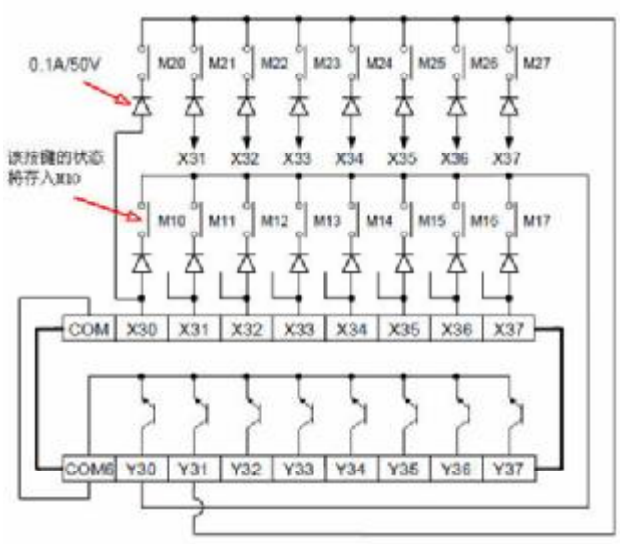
(n)为矩阵扫描的列数，即扫描用Y输出的个数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✔														
(D1)		✔													
(D2)		✔	✔	✔											
(n)	常数，n=2~8														

例如指令：



适用如下接线：



考虑到 X 输入滤波应答延迟 10ms，Y30、Y31 输出按每 20ms 顺序中断，进行即时输入输出处理；

每次自动读取操作完成后，标志 M8029 置位一个扫描周期；

若通过 8 点 X 输入和 8 点晶体管 Y 输出，可获得 64 点的扫描输入，但是此时所有输入的读取需要 20ms×8 列＝160ms 时间，不适应高速输入操作，故一般使用 X20 以后的端口作扫描输入；

该指令在程序中只能使用一次。

16bit	32bit	P	FNC 53	**HSCS**	比较置位（高速计数器）
		✔			
		13 步	指令格式： **HSCS** (S1) (S2) (D)		

当(S2)计数器的当前值等于设定值(S1)时，立即置位(D)。其中：

(S1)为设定的比较值，其值的宽度（bit位数）决定于(S2)计数器的位数；

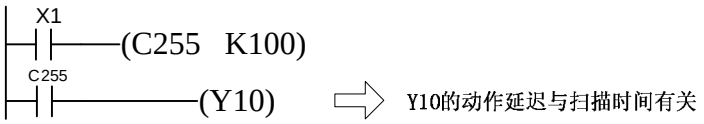
(S2)变量必须为高速计数器C235~C240 ，C241~C244，C246~C249，C251~C254，因涉及的计数器均为32bit计数器，故必须采用32bit指令DHSCS；

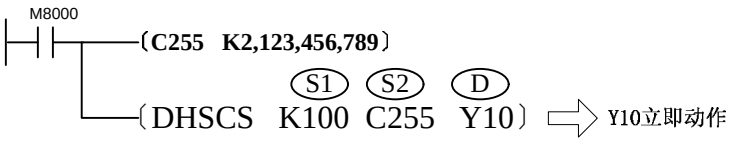
(D)为比较结果的存放单元，也可以是调用计数中断子程序：当为Y0~Y17范围端口时，为立即输出；当为Y20以后的端口时，会等到本次用户程序扫描完毕才会输出；当为M、S变量时，也为立即刷新；

当(D)项为I010～I060时，即为调用高速计数器中断0~5的子程序。当然必需编写好相应的中断子程序、开启相应中断允许标志和全局中断允许标志等，才能正常响应定时器中断。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)												✔			
(D2)		✔	✔	✔											

例如指令





使用说明：

使用 HSCS 指令时，应保证所使用的计数器已被启用（见上例），否则该计数器的值将不会

有变化；

计数器是以中断方式响应计数器的输入信号，及时比较，若本次比较时满足匹配关系，比较输出立即置位，例如上图中，若 C255 的当前值变为 99→100 或 101 →100 时，Y010 立即置位，且一直保持该状态，除非有另外的复位指令操作；

指令的比较输出只决定于脉冲输入时的比较结果动作，即使采用 DMOV 指令等改写作为比较对象的字软元件的内容，若没有脉冲输入，比较输出也不会变化；单纯的指令驱动能流也不能改变比较结果；

指令输出若为 Y 端口，必须为 Y0~Y17 范围，这样才能保证输出得到立即响应；多次驱动 HSCS 指令或与 HSCR、HSZ 指令同时驱动，对象输出 Y 的高 2 位作为同一序号的软元件。例：使用 Y000 时为 Y000~ Y007 , Y010 时为 Y010~Y017 等;

当 HSCS 指令的输出目标为中断 I010～I060 时，每个中断号只能使用 1 次，不可重复。

HSCS、HSCR、HSZ 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。

16bit	32bit	P	FNC 54	**HSCR**	比较复位（高速计数器）
		✔			
		13 步	指令格式： **HSCR** (S1) (S2) (D)		

当(S2)计数器的当前值等于设定值(S1)时，立即复位(D)。其中：

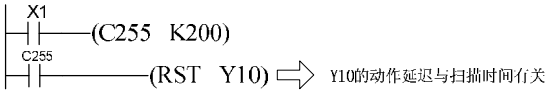
(S1)为设定的比较值，其值的宽度（bit位数）决定于(S2)计数器的位数；

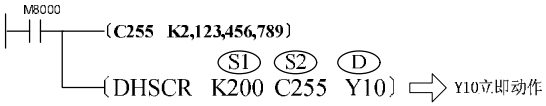
(S2)变量必须为高速计数器C235~C240 ，C241~C244，C246~C249，C251~C254，因涉及的计数器均为32bit计数器，故必须采用32bit指令DHSCR；

(D)为比较结果的存放单元：当为Y0~Y17范围端口时，为立即输出；当为Y20以后的端口时，会等到本次用户程序扫描完毕才会输出；当为M、S变量时，也为立即刷新。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)												✔			
(D2)		✔	✔	✔											

例如指令：





使用说明：

HSCR 指令的动作原理和 HSCS 指令相似，只是 HSCR 的比较输出动作与 HSCS 指令刚好相反，即计数器的值达到相等时，指定的输出复位，因此使用中的一些规定可参考 HSCR 的说明。

16bit	32bit	P	FNC 55	**HSZ**	（高速计数器）区间比较 （脉冲表格比较；频率控制模式）
		✔			
		17 步	指令格式： **HSZ** S1 S2 S D		

根据计数器 S 的当前值，与设定的比较区间 S1 和 S2 进行比较，将比较结果立即输出到以 D 地址起始的3个单元中。其中：

S1 为设定的比较区间区间下限，其值的宽度（bit位数）决定于 S 计数器的位数，其值必须不大于 S2，即 S1 ≤ S2；

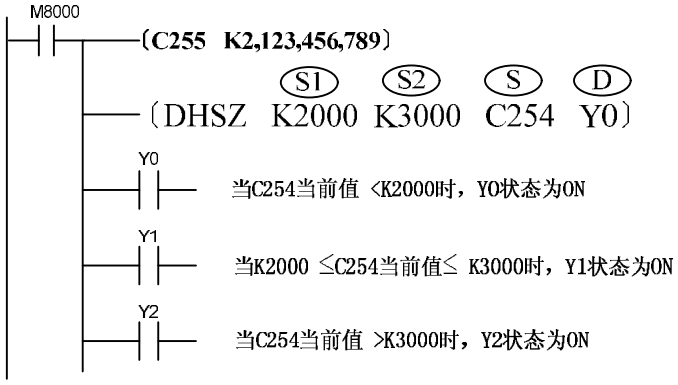
S2 为设定的比较区间区间上限，其值的宽度（bit位数）决定于 S 计数器的位数，其值必须不小于 S1，即 S1 ≤ S2；

S 变量必须为高速计数器C235~C240 ，C241~C244，C246~C249，C251~C254，因涉及的计数器均为32bit计数器，故必须采用32bit指令DHSZ；

D 为比较结果的存放单元，占用以 D 起始的3个连续地址的单元：当为Y0~Y17范围端口时，为立即输出；当为Y20以后的端口时，会等到本次用户程序扫描完毕才会输出；当为M、S变量时，也为立即刷新。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
S2					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
S												✔			
D		✔	✔	✔											

例如指令



使用说明

本指令的动作原理和 HSCS、HSCR 等指令相似，差别是采用了两个比较值，比较输出使用了 3 个连续的地址单元，因此使用中的一些规定可参考 HSCR 的使用说明；

HSZ 指令也是以中断方式进行工作的，只有当计数器对应的输入端有计数脉冲时，比较才会进行，对应的输出才会被刷新；

表格高速比较模式

将指令〔DHSZ ⓈⓘⓈ② Ⓢ Ⓓ〕中的Ⓓ指定为特殊辅助继电器 M8130，即表明为高速表格比较模式，指令中的各变量将按表格方式进行解析，说明如下：

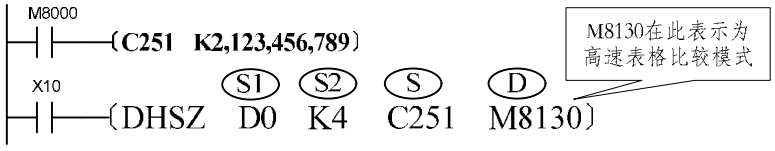
ⓈⓘⓈ只对应数据寄存器 D 变量，用于表示比较表格的起始地址。可用带 V 或 Z，指令启动后不再受 V 或 Z 的影响；

Ⓢ②只可用常数变量 K、H，用于表示表格的行数，被限制为 1≤(K ，H)≤128 ，可用带 V 或 Z，指令启动后不再受 V 或 Z 的影响；

Ⓢ可以指定为高速计数器 C235~C255；

Ⓓ为 M8130，指明为高速表格比较模式。

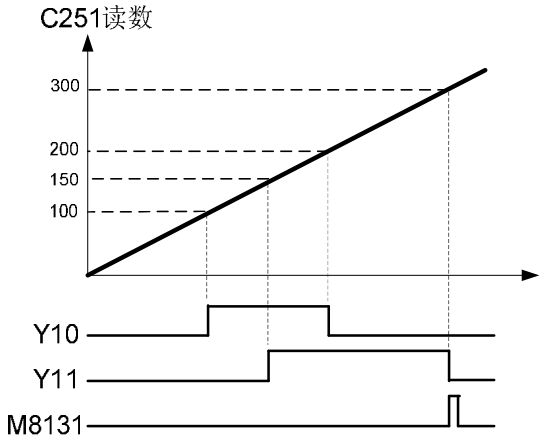
例如：



等效的比较表格为：

㊍表格起始 变量为 D0	比较值（32bit）		Y 输出编号	ON/OFF 状态	表格计数器 D8130
	低字	高字			
㊎表格行数 为 K4	D0	D1	D2	D3	0
	D4	D5	D6	D7	1
	D8	D9	D10	D11	2
	D12	D13	D14	D15	3
参数举例	K100	K0	H10	K1	执行时计数器 0→1→2→3→0 依次循环
	K150	K0	H11	K1	
	K200	K0	H10	K0	
	K300	K0	H11	K0	
说明	在接收到第 1000 个脉冲时有动作 (表格中各行比较值应依次增大)		Y10 端口动作 (若为 H11 则表示 Y10)	动作是置为 ON (若为 K0 则表示 OFF 动作)	

执行过程说明：



当㊌所指定的高速计数器C251 的当前值等於（D1、D0）設定值的時候D2所指定的輸出Y被復歸成OFF（D3=K0）或是ON（D3=K1）並保持住。而輸出Y的動作完全以中断方式來处理。

当C251 的当前值与表格的第一组设定值相等时，D8130=K1、与第一组设定值相等时，D8130=K2，如此的往下順序執行比较操作，直到最后一组比较动作完成时，M8131=ON一个扫描周期，之后D8130 清除为0，再返回到第一组进行比较。

当指令的条件接点X10变成OFF时，指令执行被中断，表格计数器D8130被清0，但指令相关的输出状态全部被保持。

本指令在被第一次扫描执行，直到END指令后，比较表格的各项设置即确定下来，因此表格

中的各参数设置需在本指令之前设置完成。

表格比较指令在用户程序中只能使用一次。此外，与其他用途使用的 HSCS/HSCR/HSZ 指令结合，可以同时驱动的指令被限制在 6 点以下。

由指令 HSZ 和 PLSY 指令实现频率控制模式

将指令〔DHSZ ㊊ ㊋ Ⓢ Ⓓ〕中的Ⓓ指定为频率控制模式说明用特殊辅助继电器M8132，通过与DPLSY指令的组合，可实现一个高速计数器的当前值控制DPLSY输出频率的功能。

说明如下：

㊊只对应数据寄存器 D 变量，用于表示比较表格的起始地址。可用带 V 或 Z，指令启动后不再受 V 或 Z 的影响；

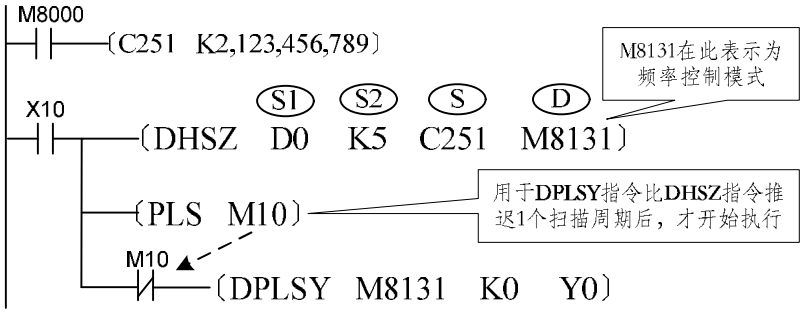
㊋只可用常数变量 K、H，用于表示表格的行数，被限制为 1≤(K ，H)≤128 ，可用带 V 或 Z，指令启动后不再受 V 或 Z 的影响；

Ⓢ可以指定为高速计数器 C235~C255；

Ⓓ为 M8132，指明为根据高速计数值 HSZ 指令来控制 PLSY 输出频率模式。

本指令在用户程序中只能使用一次；表格中的各个寄存器值需事先设定好；

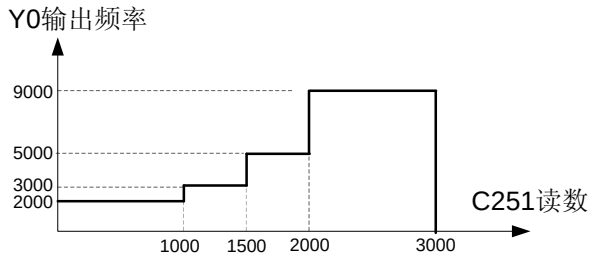
例如：



表示根据 C251 的当前频率，控制 Y0 输出频率的工作模式，等效的比较与输出频率表格为：

ⓈⒾ表格起始变量为D0	比较值（32bit）		输出频率（32bit）		表格计数器D8131
	低字	高字	低字	高字	
ⓈⒹ表格行数为K5	D0	D1	D2	D3	0
	D4	D5	D6	D7	1
	D8	D9	D10	D11	2
	D12	D13	D14	D15	3
	D16	D17	D18	D19	4
参数举例	K1000	K0	K2000	K0	执行时计数器0→1→2→3→4→0依次循环
	K1500	K0	K3000	K0	
	K2000	K0	K5000	K0	
	K3000	K0	K9000	K0	
	K0	K0	K0	K0	
说明	接收脉冲后进行比较，（如第1000个脉冲）匹配时改变输出频率。(表格中各行比较值应依次增大，最后一行可设为0。)		Y0 端口的输出频率改变为对应表格栏的设定值		

执行过程说明：



预先将所定的数据写入构成表格的数据寄存器，并有指令启动ⓈⒾ指定的高速计数器C251，运行中请勿改变表格内容的设置；

当C251的当前值小于(D1,D0)时,PLSY指令的输出频率为（D3 ， D2）的值；

当C251的当前值等于(D1,D0)时,PLSY指令的输出频率变为（D7 ， D6）的值；当C251的当前值等于(D5,D4)时,PLSY指令的输出频率变为（D11 ， D10）的值；依此类推；

最后一行的操作完毕，完成标志M8133动作。并回到第一行重复运作；

若希望在最后一行停止动作时，将最后的表格的频率置为K0；

驱动输入X010为OFF时，脉冲输出变成OFF，表格计数D8131也复位；

该项指令在初次指令执行后的END指令完成表格制作，其后开始有效。因此，为了使PLSY

指令，从驱动输入 X10 为 ON 后的第 2 个扫描周期开始动作，采用〔PLS　M10〕的触点。

注意事项：

采用频率控制模式时，编程中使用其他的 PLSY 指令以及 PLSR 指令，无法同时得到 2 路脉冲输出。

16bit	32bit	P	FNC 56	**SPD**	脉冲密度
✔					
7 步			指令格式：　**SPD** (S1) (S2) (D)		

将(S1)指定端口在(S2)时间内检测到的脉冲数，保存到(D)地址单元。其中(D)占用3个连续的单元，(D)+1为实时脉冲计数值；(D)+2为完成本次采样周期的剩余时间。

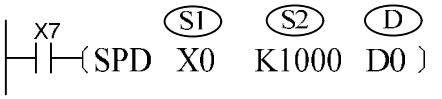
(S1)脉冲信号输入端口，只能为X0~X5；

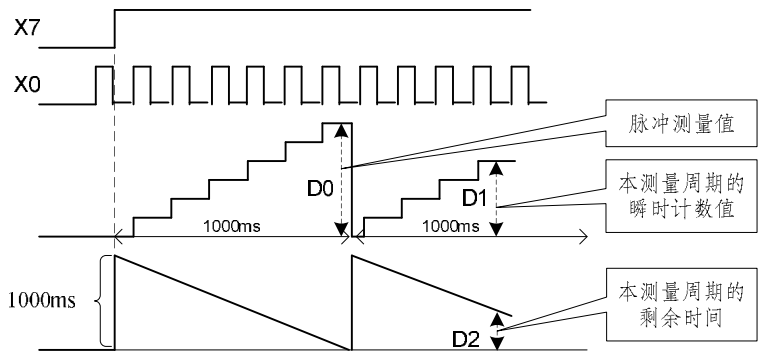
(S2)为设定的脉冲检测时间长度（ms）；

(D)为设定时间长度(S2)内接收的脉冲个数；

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)	✔														
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)											✔	✔	✔	✔	✔

例如指令





在图例中，X7 置 ON 时，DI 对 X0 的 OFF→ON 动作计数，1000ms 后将其结果存入 D0，随之 DI 复位，再次对 X000 的动作计数。

D2 用于测定剩余时间。

因此根据 D0、(S2)的设定值就可以求得脉冲的频率；若脉冲信号取自旋转编码器，可求得转速等；

在此被指定的输入 X000~X005 端口，不能再用于高速计数器或中断输入。

SPD 指令指定脉冲输入端口的 ON/ OFF 的最大频率与其 1 相高速计数器的频率限制相同，且与高速计数器、 PLSY 以及 PLSR 指令同时使用时，必须将这些处理频率合计值限制在规定频率以下。

16bit	32bit	P	FNC 57	**PLSY**	脉冲输出
✔	✔				
7 步	13 步		指令格式： **PLSY** (S1) (S2) (D)		

由于继电器不适合高频率动作，只有晶体管输出型PLC才能使用该指令。指令功能是由(D)指定的端口，以(S1)的频率，输出(S2)个脉冲，脉冲发送完毕，M8029标志被置位。其中：

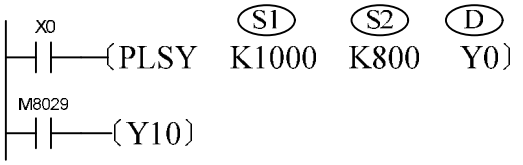
(D)为脉冲输出端口，对于MT型晶体管输出型主模块，只能选择Y0、Y1；对于MTQ型晶体管输出型主模块，可选择Y0/Y1/Y2/Y3/Y4等端口；

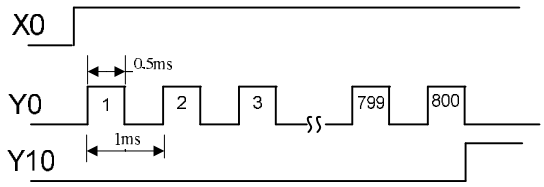
(S1)为设定的输出脉冲频率，对于16bit指令（PLSY），设定范围为1~32,767；对于32bit指令（DPLSY），设定范围为1~100,000（即1Hz～100kHz）；

(S2)为设定的脉冲输出个数，对于16bit指令（PLSY），设定范围为1~32,767；对于32bit指令（DPLSY），设定范围为1~2,147,483,647。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔													

例如指令





使用 PLSY(16bit 指令)时，(S1)和(S2)都只能是 16bit 宽度，且为有符号数；

使用 DPLSY(32bit 指令)时，(S1)和(S2)若为 D、C、T 变量，则按 32bit 宽度有符号数处理；

当执行到 PLSY 指令后，Y 即开始输出脉冲；运行中若改变(S2)元件（为 D、C、T 变量）的参数值，对当前输出的脉冲数没有影响，将从下一次启动该指令时生效；

在 PLSY 输出脉冲过程中，若指令能流 X0 变为 OFF，则输出脉冲被停止；若 X0 变为 ON，将从当前 PLSY 指令的设置重新开始脉冲输出。

使用说明：

- PLSY 所用的输出 Y 端口不要与 PWM 或 PLSR 指令所用的 Y 端口重复；
- 可以使用 2 个 PLSYPLSY 指令，或 2 个 PLSR 指令，分别对应 Y0、Y1 端口；
- 若使用了 HSZ 与 PLSR 指令组合的高速频率控制模式，无法同时得到 2 点频率输出；
- 若当前正在执行 PLSY 和 PLSR 的两路脉冲输出，无法同时运行 SPD 指令；
- HSCS、HSCR、HSZ 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；
- 使用 PLSY 指令时，使用了如下特殊寄存器：

寄存器		定　　义	备 注
D8140	低字	PLSY 或 PLSR 指令中设定的输出至 Y0 口的脉冲总数	可用指令： DMOV K0 D81xx 进行清除操作
D8141	高字		
D8142	低字	PLSY 或 PLSR 指令中设定的输出至 Y1 口的脉冲总数	
D8143	高字		
D8136	低字	已向 Y0 及 Y1 输出的脉冲个数的累计值	
D8137	高字		

16bit	32bit	P	FNC 58	**PWM**	脉宽调制
✔					
7 步			指令格式： **PWM** (S1) (S2) (D)		

由于继电器不适合高频率动作，只有晶体管输出型PLC才能使用该指令。指令功能是以(S1)指定的脉冲宽度，(S2)指定的脉冲周期，由(D)指定的端口持续输出脉冲。其中：

(S1)为设定的输出脉冲宽度，必须有(S1)≤(S2)，设定范围为0~32,767ms；

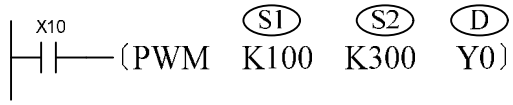
(S2)为设定的脉冲输出周期，必须有(S1)≤(S2)，设定范围为1~32,767ms；

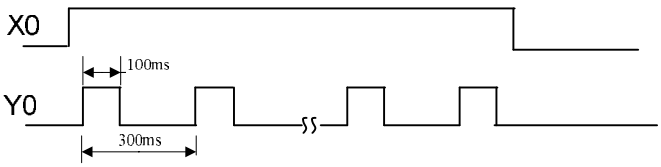
(D)为脉冲输出端口，MT型主模块只能选择Y0、Y1端口，MTQ型主模块则可选Y0、Y1、Y2、Y3、Y4等端口，不要与PLSY,PLSR指令的输出端口重复。

本指令是以中断方式执行，当指令能流为OFF时，输出停止。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔													

例如指令：





16bit	32bit	P	FNC 59	**PLSR**	带加减速脉冲输出
✔	✔				
7 步	17 步		指令格式： **PLSR** (S1) (S2) (S3) (D)		

由于继电器不适合高频率动作，只有晶体管输出型PLC才能使用该指令。指令功能是在指定的高速输出端口，以加减方式逐级改变输出频率，输出指定的脉冲数。其中：

(S1)为设定的输出脉冲的最高频率，取10的整数倍数值，设定范围为10~100,000Hz；

(S2)为设定的输出脉冲数，16bit指令，设定范围为110~32,767；32bit指令，设定范围为110~2,147,483,647；设定的脉冲数小于110时，不能正常输出脉冲；

(S3)为设定的加减速时间，减速时间与加速时间相同，ms单位，设定时请注意：

1. 加减速时间应为控制程序最大扫描时间（D8012）的10倍以上，否则加减速时间不确定；
2. 加减速的最小值计算公式为：(S3) ≥ 90000/(S1)；
3. 加减速的最大值计算公式为：(S3) ≤ （(S2) * 818）/(S1)；
4. 加减速过程的速度按10级逐级变化，若难以满足上述1/2/3的条件，请降低(S1)的设置。

(D)为脉冲输出端口，MT型主模块只能选择Y0、Y1端口，MTQ型主模块则可选Y0、Y1、Y2、Y3、Y4等端口，不要与PLSY指令的输出端口重复。

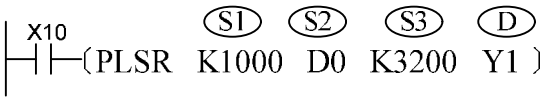
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S3)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔													

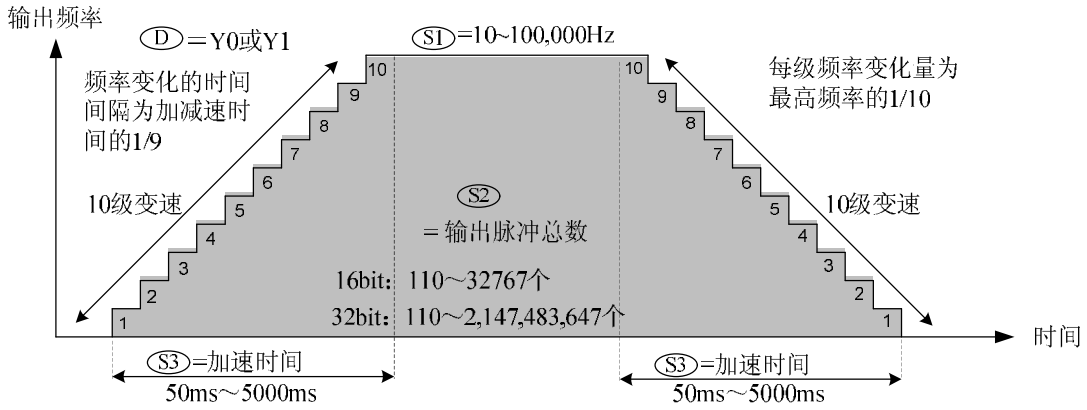
使用说明：

- 本指令是以中断方式执行，不受扫描周期的影响；
- 当指令能流为OFF时，输出停止；当能流由OFF→ON时，脉冲输出处理重新开始；
- 在脉冲输出过程中，改变操作数，对本次输出没有影响，修改的内容在指令下次执行的时候生效。指令执行完毕，M8029标志置为ON;
- 同时使用2个PLSY或2个PLSR指令，可以在Y0、Y1端口得到各自独立的2路高速脉冲输出；
- 同时使用1个PLSY和1个PLSR指令，可以在Y0、Y1端口得到各自独立的2路高速脉冲输出；
- 组合使用HSZ和PLSR指令（频率控制模式）时，仅能在Y0或Y1中选择1点，此时也不能通过PLSY获得另外1路脉冲输出；
- 与PWM指令的输出端口号不能重复；
- 再次启动PLSR指令时，需在上次脉冲输出操作结束（Y0结束时M8147＝0；Y1结束时

M8148＝0）后，再延迟1个扫描周期，方可再启动该指令；

例如指令：





自 Y0 或 Y1 输出脉冲过程中，占用如下特殊寄存器：

寄存器		定　义	备 注
D8140	低字	PLSY 或 PLSR 指令中设定的输出至 Y0 口的脉冲总数	可用指令： DMOV K0 D81xx 进行清除操作
D8141	高字		
D8142	低字	PLSY 或 PLSR 指令中设定的输出至 Y1 口的脉冲总数	
D8143	高字		
D8136	低字	已向 Y0 及 Y1 输出的脉冲个数的累计值	
D8137	高字		

16bit	32bit	P	FNC 61	**SER**	数据查找
✔	✔	✔			
7 步	17 步	17 步	指令格式： **SER** S1 S2 D n		

该指令用于从一组数据中，查找相同数据的单元、同时对最大值、最小值的检索。其中：

S1 为数据组的的起始地址；

S2 为待检索的数据；

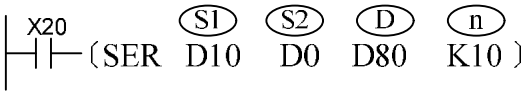
D 为检索结果存放区的起始地址；

n 为被检索数据区的长度。

当使用32bit指令时，(S1) (S2) (D) 均指向32bit变量，(n) 也按32bit变量宽度进行计算。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)							✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔
(n)					✔	✔							✔		
	取值范围：16bit 指令：n＝1～256；32bit 指令：n＝1～128														

使用举例：



(S1)	被检索数据例	(S2)	元件序号	参数状况
D 10	(D 10)=K100	比较数据 (D10) =K100	0	相等
D 11	(D 11)=K123		1	
D 12	(D 12)=K100		2	相等
D 13	(D 13)=K98		3	
D 14	(D 14)=K111		4	
D 15	(D 15)=K66		5	最小
D 16	(D 16)=K100		6	相等
D 17	(D 17)=K100		7	相等
D 18	(D 18)=K210		8	最大
D 19	(D 19)=K88		9	

(n)

检索结果

(D)	参数	定义
D80	4	相等参数的个数
D81	0	第一个相等参数的序号
D82	7	最后一个相等参数的序号
D83	5	最小参数的序号
D84	8	最大相等参数的序号

使用说明：

- 当指令能流X20为ON时，方才进行比较；
- 比较的方法为有符号数的代数比较方法进行，例如 －8＜2；
- 当最小值、最大值有多个时，分别显示序号最大的元件；
- 存储检索结果的单元占用(D)开始的5个连续单元。若不存在相等数据时，上例中的D80～D82均为0。

16bit	32bit	P	FNC 62	**ABSD**	凸轮控制绝对方式
✔	✔				
9 步	17 步		指令格式：　**ABSD** (S1) (S2) (D) (n)		

该指令完成的操作是多区段比较，用于实现凸轮控制，比较用的表格、计数器等均按绝对方式设置。该指令是主程序中扫描执行，比较结果受扫描时间的滞后影响。其中：

(S1)为比较表格的起始元件地址；

(S2)为计数器元件序号，使用32bit指令时，可为32bit计数器；

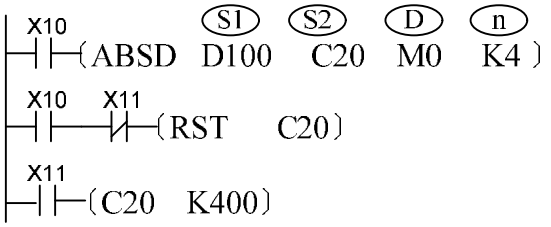
(D)为比较结果存放区的起始地址，占用(n)个连续地址的bit变量单元；

(n)为多段比较数据的组数。

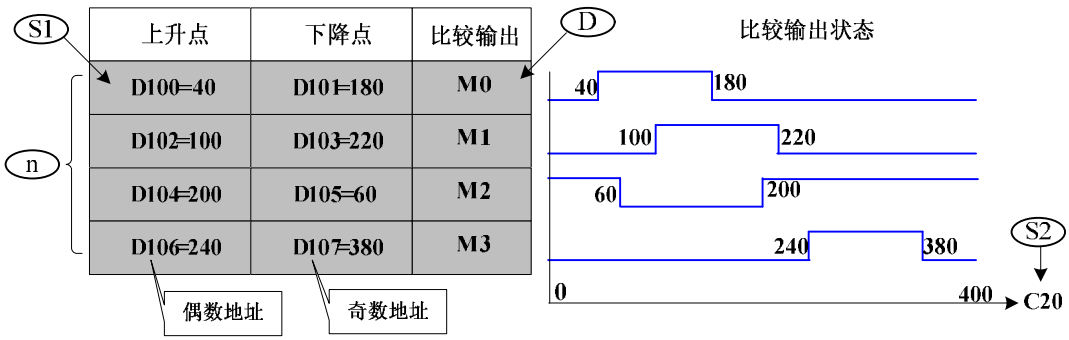
当使用32bit指令时，(S1) (S2) (D)均指向32bit变量，(n)也按32bit变量宽度进行计算。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)							✔	✔	✔	✔	✔	✔	✔		
(S2)												✔			
(D)		✔	✔	✔											
(n)	常数，n＝1～64；														
(S1)操作数为KnX、KnY、KnM、KnS 时，若为16bit指令，必须指定K4；若为32bit指令，必须指定K8且X、Y、M、S 的元件编号必须是8的倍数； (S1)操作数在16bit指令时只能指定C0~C199；32bit指令时则只能指定C200~C254；															

使用举例：



若已给相关变量按如下赋值，当X10＝ON时，执行结果如下图：



使用说明：

- ABSD指令执行前，应给相关表格的各变量用MOV指令赋值；

- 即使DABSD指令中采用了高速指令，比较输出也受用户程序扫描时间的滞后影响，对于需要及时响应的应用，可采用HSZ高速比较指令；
- 程序中只能使用ABSD 指令一次。

16bit	32bit	P	FNC 63	**INCD**	凸轮控制增量方式
✔					
9 步			指令格式： **INCD** (S1) (S2) (D) (n)		

该指令完成的操作是多区段比较，用于实现凸轮控制，比较用的表格、计数器等均按增量方式设置。该指令是主程序中扫描执行，比较结果受扫描时间的滞后影响。其中：

(S1)为比较表格的起始元件地址；

(S2)为计数器元件序号，其相邻的(S2)＋1单元则被用于计算比较匹配后计数器复位的次数。使用32bit指令时，可为32bit计数器；

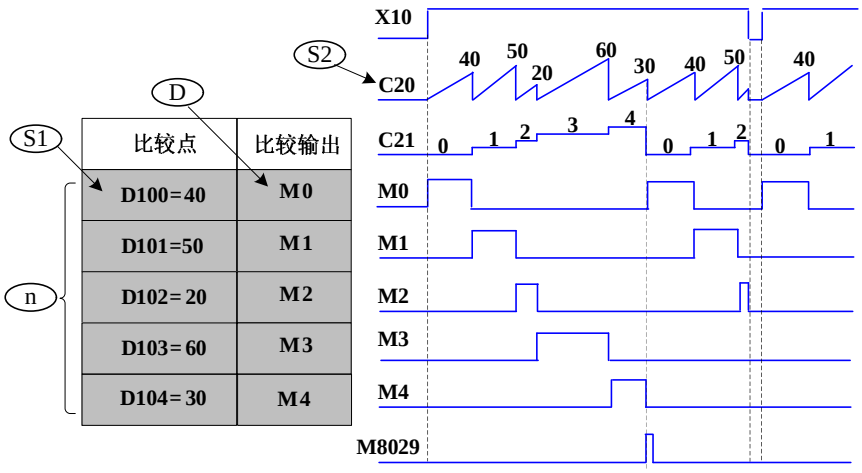
(D)为比较结果存放区的起始地址，占用(n)个连续地址的bit变量单元；

(n)为多段比较数据的组数。

当使用32bit指令时，(S1) (S2) (D)均指向32bit变量，(n)也按32bit变量宽度进行计算。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)							✔	✔	✔	✔	✔	✔	✔		
(S2)												✔			
(D)		✔	✔	✔											
(n)	常数，n＝1～64；														
(S1)操作数为KnX、KnY、KnM、KnS 时，若为16bit指令，必须指定K4；若为32bit指令，必须指定K8且X、Y、M、S 的元件编号必须是8的倍数； (S1)操作数在16bit指令时只能指定C0~C199；32bit指令时则只能指定C200~C254；															

使用举例：



使用说明：

- INCD指令执行前，应给相关表格的各变量用MOV指令赋值；
- 比较输出也受用户程序扫描时间的滞后影响，对于需要及时响应的应用，可采用HSZ高速比较指令；
- 程序中只能使用INCD 指令一次。

16bit	32bit	P	FNC 64	**TTMR**	演示定时器
✔					
5 步			指令格式： **TTMR** (D) (n)		

该指令的功能是将指定输入端口的按键保持时间乘以(n)倍数后存入变量(D)，一般用于参数设定。其中：

(D)为按键保持时间乘以倍数后的乘积，而(D)＋1单元则用于保存按键的时间保存时间，时间单位为秒；

(n)为倍数设定，注意实际倍数为10^n计算方式，例如：

n＝K0时，实际倍数为×1；

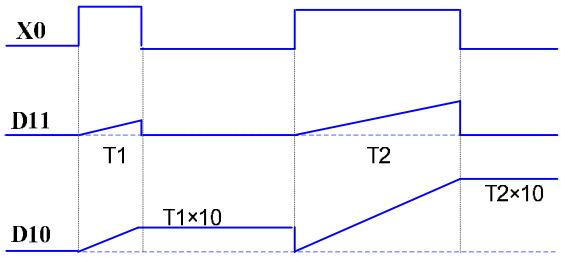
n＝K1时，实际倍数为×10；

n＝K2时，实际倍数为×100，依此类推。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)													✔		
(n)					✔	✔									

使用举例：

在X0闭合期间，D10=D11×10；
当X0断开后，D10值维持不变。



16bit	32bit	P	FNC 65	**STMR**	特殊定时器
✔					
5 步			指令格式： **STMR** S m D		

该指令的功能是根据指令能流，产生4种延时动作的专用指令。其中：

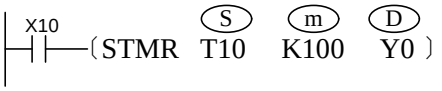
S 用于产生延迟动作的计时器序号，可用T0~T199；

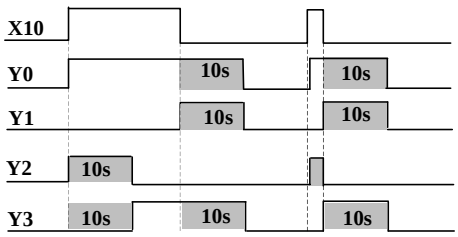
m 为延时设定值，单位为100ms，设定值范围为K1～K32767；

D 为延时动作输出元件的起始编号，共占用4个连续单元。

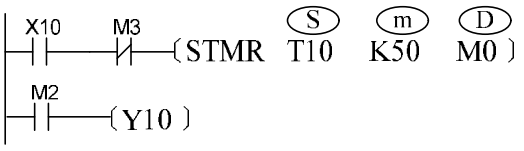
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											✔				
m	常数，m＝1～32767														
D		✔	✔	✔											

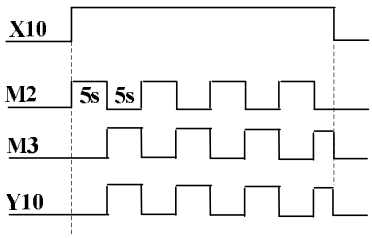
使用举例：





若在指令能流中引入 D 的元件，可方便地实现振荡器输出：





使用说明：

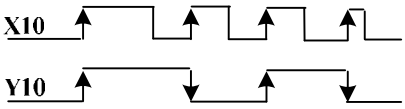
在本指令中使用的计时器不得再用于其他指令中，重复使用。

16bit	32bit	P	FNC 66	**ALT**	ON/OFF 交替
✔		✔			
3 步		3 步	指令格式： **ALT** (D)		

该指令的功能是能流有效时，将(D)元件的状态反转。其中(D)为位变量元件。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)		✔	✔	✔											

一般使用ALTP指令，使用举例：



如下指令产生的动作与之相同：

16bit	32bit	P	FNC 67	**RAMP**	斜坡指令
✔					
9 步			指令格式： **RAMP** (S1) (S2) (D) (n)		

该指令的功能是在给定的两个数据中间，在指定的时间区间，进行线性插值，按扫描执行的时间依次输出过程值，直到区间末端的终点值为止。其中：

(S1)斜坡信号的起始值单元；

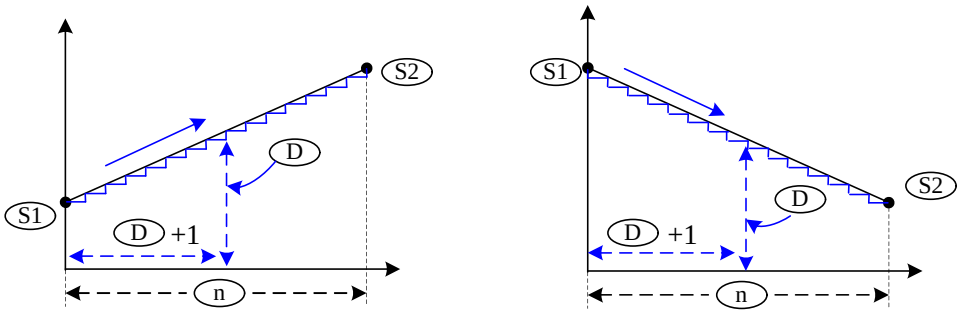
(S2)斜坡信号的终点值单元；

(D)为线性插值信号的过程值存放单元，而插值次数的计时器存放在(D)＋1单元；

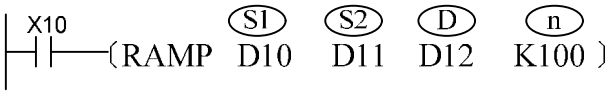
(n)完成插补过程的程序扫描执行次数。由于插值输出是在正常主循环中进行的，为了保证插值输出的线性，需要将程序执行设置为固定扫描方式（见M8039、D8039说明）。

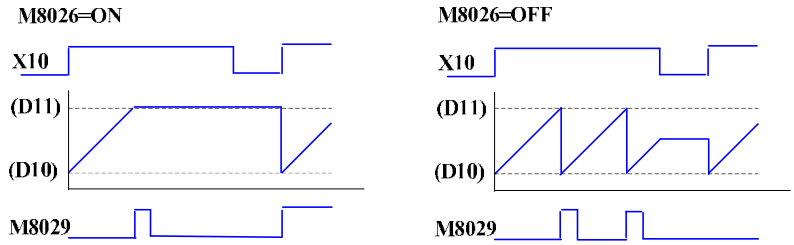
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													✔		
S2													✔		
D													✔		
n	常数，1～32767														

插值计算按整型数计算，丢弃了计算小数。指令功能如下图：



RAMP指令执行有2种模式，由M8026标志进行选择；每次插值运算完毕，M8029置位一个扫描周期。执行特点如下例：





16bit	32bit	P	FNC 68	ROTC	旋转台控制指令
✔					
9 步			指令格式：　ROTC　S　m1　m2　D		

该指令是用于旋转工作台上工件取放控制的简捷指令，旋转工作台的位置检测信号需按指定方式配置才能正常工作。其中：

S 计数变量的起始单元；

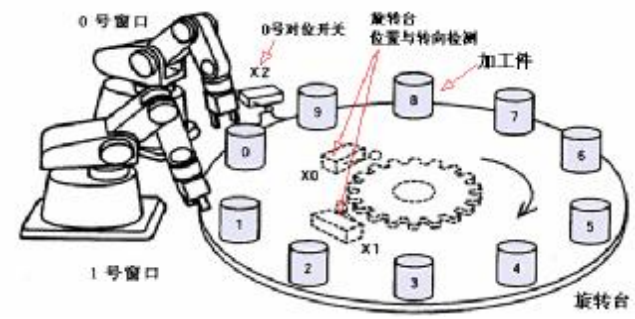
m1 旋转工作台上的工位数，必须 m1 ≧ m2；

m2 旋转工作台上的低速工位数，必须 m1 ≧ m2；

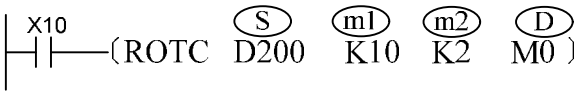
Ⓓ为旋转台位置检测信号存放的起始单元，占用随后的8个位变量单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ													✔		
Ⓓ		✔	✔	✔											
ⓜ1	2～32767，m1≧m2														
ⓜ2	0～32767，　m1≧m2														

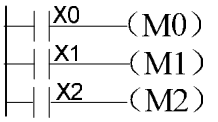
信号配置方式如下图，图中X0、X1分别接AB正交编码器的A相和B相输出信号，也可采用机械开关得到正交相位的信号；X2接用于0号工位的检测输入（当旋转到0号工位时为ON状态），由此3个信号即可检测旋转工作台的当前转动速度和转向和工位。



应用举例：



该代码实际使用的变量空间说明如下：

变量	功能定义	操作说明
D200	作为计数寄存器使用	由用户程序事先设定好该3个单元
D201	调用窗口号码设定	
D202	调用工件号码设定	
M0	A相信号	用户程序每次扫描本语句前执行： 
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	当X10为OFF时，M3~M7均为OFF；
M6	低速反转	
M7	高速反转	

在后续的用户程序中，将M3~M7从Y输出口中输出，控制外部执行元件即可。

程序中只能使用1次ROTC指令。

16bit	32bit	P	FNC 69	**SORT**	数据排序
✔					
17 步			指令格式： **SORT** (S) (m1) (m2) (D) (n)		

该指令是将m1行×m2列的数组（由(S) (m1) (m2)描述），以第(n)列参数排序后，存放于由(D)单元起始的变量区域。其中：

(S)为第1行（或称第1条记录）的首个变量的起始单元；

(m1)数组的行数，或称记录数；

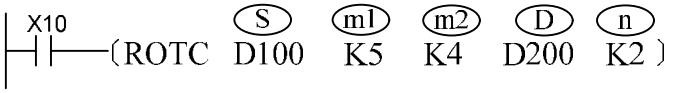
(m2)数组的列数，或称每条记录的栏目数；

(D)为排序后存放的起始单元，占用随后的变量单元数目与排序前的数组变量数目相同；

(n)为以排序为依据的数组列号。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(m1)	常数，1～32														
(m2)	常数，1～6														
(D)													✔		
(n)					✔	✔							✔		

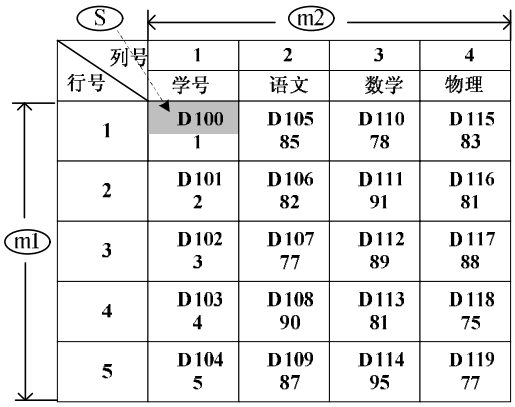
其中(n)的值在1～(m2)范围。例如指令：



当X10＝ON时，开始排序运算，指令执行完毕，M8029被置位1个程序扫描周期；

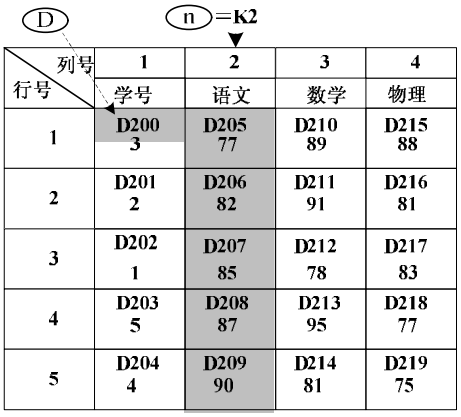
若需要再次排序，需将X10＝OFF一次。

上述指令的等效表格及其数据举例：



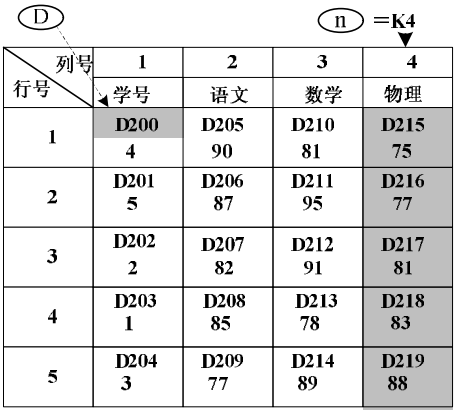
行号＼列号	1	2	3	4
	学号	语文	数学	物理
1	D100 1	D105 85	D110 78	D115 83
2	D101 2	D106 82	D111 91	D116 81
3	D102 3	D107 77	D112 89	D117 88
4	D103 4	D108 90	D113 81	D118 75
5	D104 5	D109 87	D114 95	D119 77

按指令要求 n ＝K2排序后的表格数据结果：



行号＼列号	1	2	3	4
	学号	语文	数学	物理
1	D200 3	D205 77	D210 89	D215 88
2	D201 2	D206 82	D211 91	D216 81
3	D202 1	D207 85	D212 78	D217 83
4	D203 5	D208 87	D213 95	D218 77
5	D204 4	D209 90	D214 81	D219 75

若指令中 n ＝K4，则排序后的表格数据结果如下：



行号＼列号	1	2	3	4
	学号	语文	数学	物理
1	D200 4	D205 90	D210 81	D215 75
2	D201 5	D206 87	D211 95	D216 77
3	D202 2	D207 82	D212 91	D217 81
4	D203 1	D208 85	D213 78	D218 83
5	D204 3	D209 77	D214 89	D219 88

16bit	32bit	P	FNC 70	**TKY**	0～9 数字键输入
✔	✔				
7 步	13 步		指令格式： **TKY** S D1 D2		

该指令是指定10个连续的位变量单元（如X输入端口），依次代表10进制的0～9按键，当有按键动作（状态为ON）时，以按键的动作顺序，可输入十进制0～9999的4位数值；若采用32bit指令，则可输入十进制0～99,999,999的8位数值。其中：

S 为按键的起始输入端口，使用随后的共10个位单元（如X端口）；

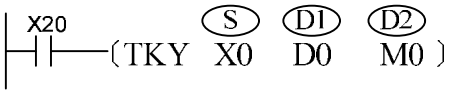
D1 为已输入数值的存放单元；

D2 为按键组当前状态的暂存起始单元，占用随后的共11个位单元。

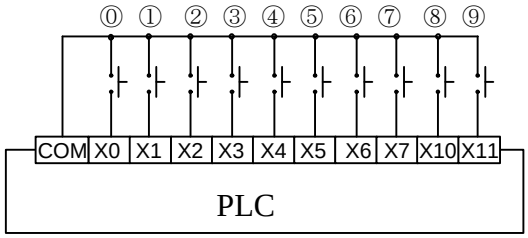
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

(S)	✔	✔	✔	✔											
(D1)								✔	✔	✔	✔	✔	✔	✔	✔
(D2)		✔	✔	✔											

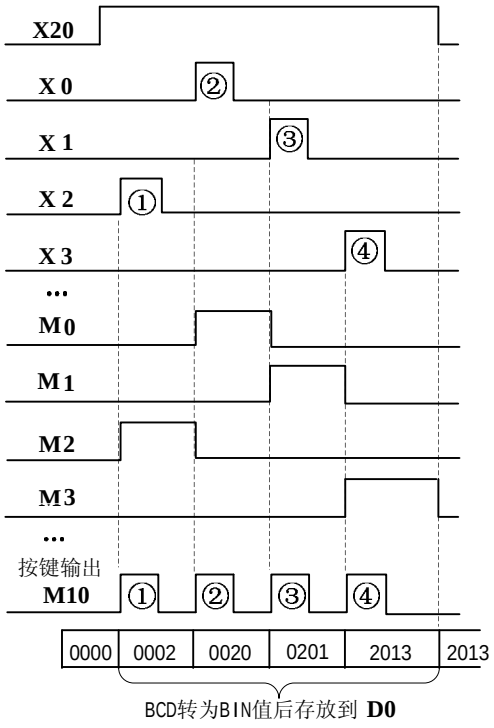
例如指令



对应的硬件接线图为：



若要输入数字“2013”，按顺序按键②⓪①③即可，PLC 内部变量的动作如下图。



- 由指令中参数的设置，X0～X11 分别对应 0～9 数字键；M0~M9 对应键的状态；当有任何按键按下时，按键输出单元 M10 都会置位；
- 按键值（如 2013）被转换为 BIN 格式后存入指定的 (D1) 单元 D0；（D0＝0x7DD），即使驱动的能流变为 OFF，D0 也不会改变；
- 当有多个按键按下时，先检测到的按键有效；当输入的数字超过 5 位后，最先输入的数字变化溢出，只留下输入的最后 4 个数字。

若采用32bit指令（DTKY），(D1) 占用32bit的变量，对上例即为D1、D0，分别为高字和低字。

16bit	32bit	P	FNC 71	**HKY**	16 键输入
✔	✔				
9 步	17 步		指令格式： **HKY** (S) (D1) (D2) (D3)		

该指令是读取4×4矩阵型共16个按键，依次代表10进制的0～9按键，以及A~F的功能按键，当有按键动作（状态为ON）时，以按键的动作顺序，可输入十进制0～9999的4位数值，或A~F的功能键；若采用32bit指令，则可输入十进制0～99,999,999的8位数值，或A~F的功能键。其中：

Ⓢ为按键的扫描输入X端口的起始端口号，使用随后的共4个X端口；

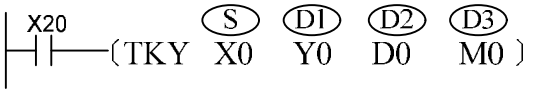
Ⓓ1为按键的扫描输出Y端口的起始端口号，使用随后的共4个Y端口；

Ⓓ2为按键输入值存放单元,0~9999；当为32bit指令时可输入0～99,999,999范围内的数值；

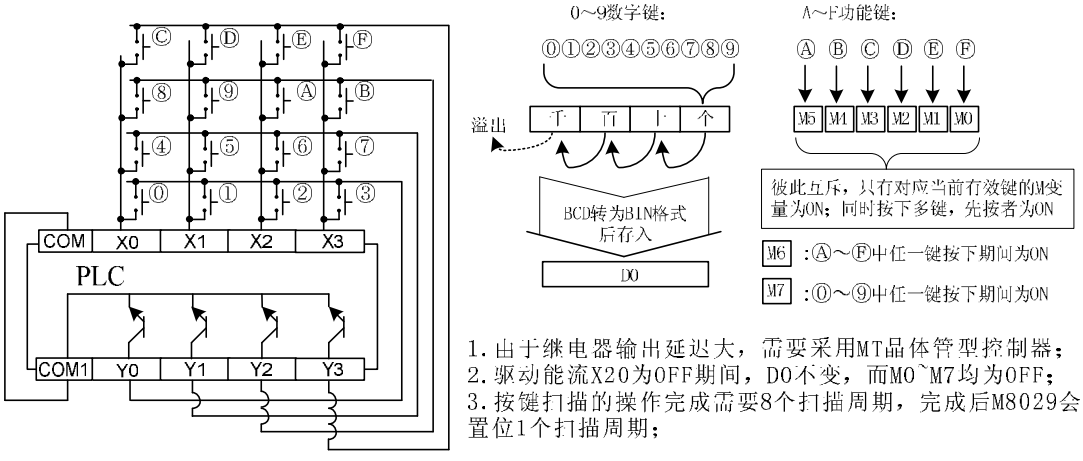
Ⓓ3为按键输入状态起始单元地址，共占用连续的8个位变量单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ	✔														
Ⓓ1		✔													
Ⓓ2										✔	✔	✔	✔	✔	✔
Ⓓ3		✔	✔	✔											

例如指令



对应的接线图及参数响应说明如下：



- 由于键扫描操作需要多个执行周期完成，为避免X端口滤波的影响，请采用“恒定扫描”模式，或定时中断处理。
- 扩展功能说明：

当将特殊变量M8167置为ON，本指令将⓪～Ⓕ按键按16进制数据进行存储，保存到(D2)单元。

16bit	32bit	P	FNC 72	**DSW**	读取数字开关设定
✔					
9 步			指令格式： **DSW** (S) (D1) (D2) (n)		

该指令是读取矩阵型设置开关的状态，以4个BCD设定开关为1组，将设定值读取后存放到指定单元，最多可读取2组设定开关。其中：

(S)为按键的扫描输入X端口的起始端口号，若(n)＝1，使用随后的共4个X端口；若(n)＝2，则使用随后的共8个X端口；

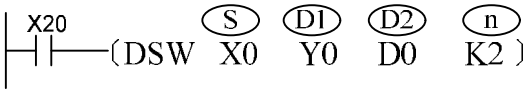
(D1)为按键的扫描输出Y端口的起始端口号，使用随后的共4个Y端口；

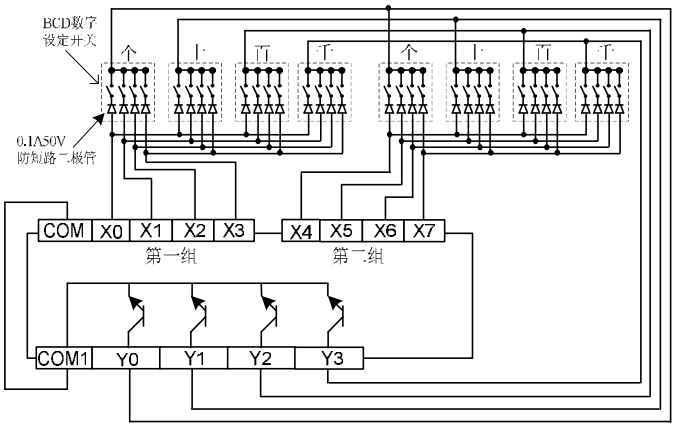
(D2)为按键输入值存放单元,0~9999；

(n)为数字设定开关的组数，只能取1～2。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✔														
(D1)		✔													
(D2)											✔	✔	✔	✔	✔
(n)	常数，n＝1～2														

例如指令：

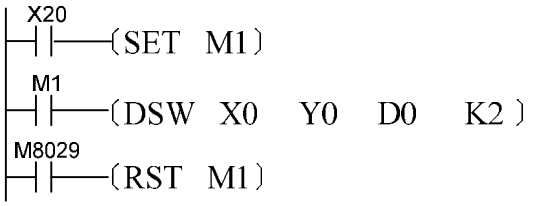




当X20＝ON时，执行扫描读取数字开关设定的操作：
1. 第一组数字开关的设定值，经BIN转换后存入D0;
2. 第二组数字开关的设定值，经BIN转换后存入D1;
3. 一次读取循环完毕，M8029会置位一个扫描周期。

使用说明：

- 需要使用晶体管输出型PLC才能正常检测数字开关；
- 一个数字开关设定的读取操作需要多个扫描周期才能完成，若采用按键启动读取操作，建议采用如下程序语句，保证可读取周期的完整：



16bit	32bit	P	FNC 73	**SEGD**	七段码译码
✔		✔			
5 步		9 步	指令格式： **SEGD** Ⓢ Ⓓ		

该指令是数据源的低4位，翻译成7段显示码，存放到目的变量的低8位中。其中：

Ⓢ为待译码的数据源；

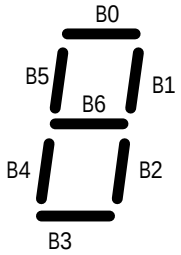
Ⓓ为译码后存放7段码的变量。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Ⓓ								✔	✔	✔	✔	✔	✔	✔	✔

例如指令：

操作是，当X20为ON时，将D0内数据低4位译码后，输出到Y10～Y17端口。

翻译用的对应表如下表。该表格不需用户准备，PLC系统内部已具备该对照表。

数　据		数码管组合	内部译码表值								译码后字符
HEX数	BIN数		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000	 每位对应一个笔段 1＝笔段点亮 0＝笔段熄灭	0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		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	0	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	1	0	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	1	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

16bit	32bit	P	FNC 74	**SEGL**	七段码扫描显示
✔					
7 步			指令格式：　**SEGL** Ⓢ Ⓓ ⓝ		

该指令是利用8个或12个Y端口，用于4位或8位七段锁存数码管的显示驱动，显示方式为扫描驱动方式。其中：

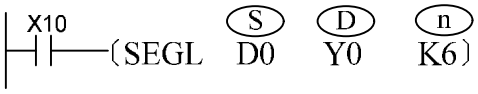
Ⓢ为待显示的数据，其值在BCD转换后才送到数码管进行显示；

Ⓓ为显示驱动用的Y端口起始地址号；

ⓝ为根据显示数据的组数、信号逻辑等相关的设定值，见下面的详细描述。

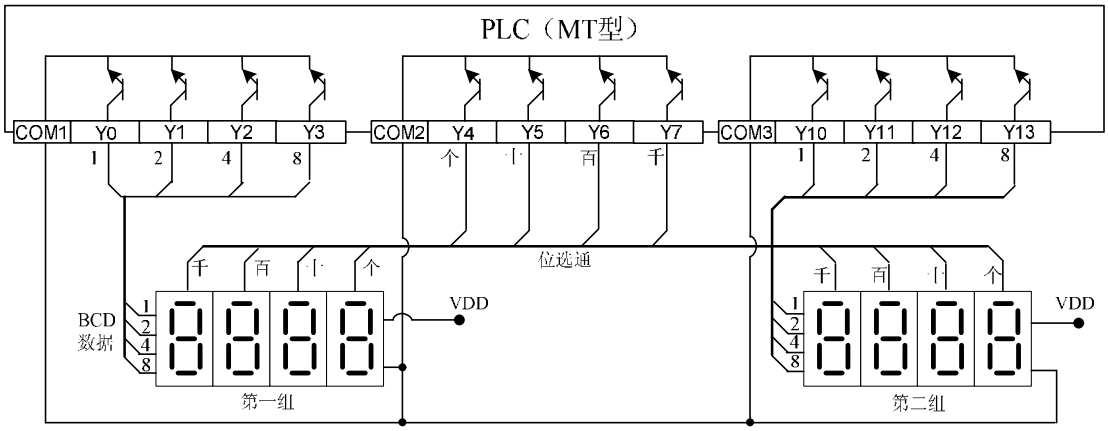
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Ⓓ		✔													
ⓝ	常数，n＝0~7														

例如指令：



对应的硬件接线如下图，在第一组数码管上显示D0的内容，在第二组数码管上显示D1的内

容，若D0或D1的读数超过9999，程序运行就会出错：



接线图中所使用的数码管，自带显示数据锁存、7段译码和驱动、负逻辑型（输入端口为低电平时，表示输入的数据为1，或被选通）的7段数码显示管。显示处理中，PLC的Y4~Y7端口每次只有1个口为ON，作为位选通信号，此时Y0~Y3口上的数据即为送给对应位的BCD码数据，当位选通信号由ON→OFF时，即被锁存到数码管内的锁存器中，经过内部的译码和驱动后，数码管将数字显示出来。PLC系统依次将Y4～Y7循环作同样处理，直到将4位都处理完毕。同样的道理，Y10~Y13是第二组4位数码管的数据输出端口，共用Y4~Y7的位选通线，处理方法相同，两组的显示处理是同时进行的。

完成一次显示刷新需要12个扫描周期，处理完成后，M8029标志置为ON一个扫描周期。

(n)的选择：根据可编程控制器的正负逻辑、七段码的正负逻辑等因素，按以下原则选择：

显示组数	1组				2组			
Y数据输出极性	PNP		NPN		PNP		NPN	
选通与数据极性	相同	相反	相同	相反	相同	相反	相同	相反
(n)的取值	0	1	2	3	4	5	6	7

H2U系列晶体管输出型的Y输出极性为NPN型。该指令在程序中最多只能使用2次。

16bit	32bit	P	FNC 75	**ARWS**	方向开关
✔					
7 步			指令格式： **ARWS** (S) (D1) (D2) (n)		

该指令可指定X作为编辑按键、Y端口驱动4位7段数码管，用作寄存器的参数编辑的简易界面。其中：

Ⓢ为指定按键输入的起始地址，占用后续的共4个位单元；

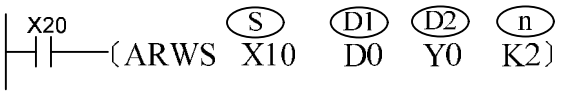
Ⓓ1用于被显示及修改的变量，只能显示一个16bit宽度的变量；

Ⓓ2用作数码管显示驱动的Y端口起始地址，占用后续的共8个Y端口；

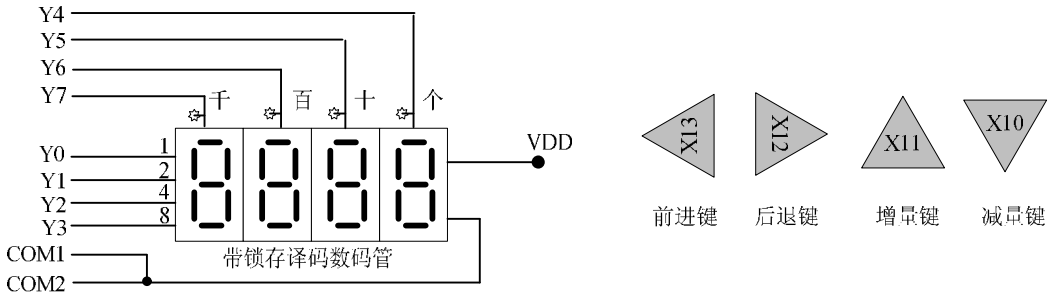
ⓝ为信号逻辑的设定值，参见前面SEGL指令中关于ⓝ的详细描述。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ	✔	✔	✔	✔											
Ⓓ1											✔	✔	✔	✔	✔
Ⓓ2		✔													
ⓝ	常数，n＝0~2														

例如指令：



对应的硬件接线如下图所示，PLC应为晶体管输出型：



操作方法：

1. 图中数码管显示 D0 的数值，按下 X10~X13 可修改其中的数值，D0 的数值只能在 0～9999 之间；
2. 当 X20 为 ON 时，光标位为千位。每次按后退键（X12）时，指定位按“千 → 百→十→个→千”的顺序切换；若按前进键（X13），则切换顺序相反；光标位由选通脉冲信号（Y004~Y007）连接的 LED 指示；
3. 对于光标位，每次按增量键（X11）该位的内容按 0 → 1→ 2→……8→9→0→1 变化。按减量键（X10）时，则按 0→9→8→7→…… 1→0→9 变化，修改的值立即生效。

指令使用说明：

当用户程序扫描时间短时，请使用恒定扫描模式，或在定时中断内按固定时间间隔运行。

16bit	32bit	P	FNC 76	**ASC**	ASCII 转换
✔					
11 步			指令格式： **ASC** (S) (D)		

该指令将指定ASCII字符（串）转换为对应的Hex格式数值，输出到指定变量单元中，该指令一般用于向外输出特定的信息字符串，用于状态提示或告警。其中：

(S)为待转换的ASCII字符串，为指令中以常数变量输入，长度为8个字符；

(D)为进行ASCII转换后的数值存放的起始单元号，占用随后的共4或8个变量单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	在指令输入时，由常数变量输入，允许长度为 8 个字符。														
(D)											✔	✔	✔		

例如指令：

当X20＝ON时，D200～D2003的赋值如右图

	高字节	低字节
D200	74（t）	53（S）
D201	70（p）	6F（o）
D202	63（e）	70（p）
D203	00	46（d）

若将特殊寄存器M8161置为ON，则每个ASCII字符在转换后占用1个16bit变量，如下图，每个变量的高字节填0处理：

	高字节	低字节
D200	00	53（S）
D201	00	74（t）
D202	00	6F（o）
D203	00	70（p）
D204	00	70（p）
D205	00	63（e）
D206	00	64（d）
D207	00	00

16bit	32bit	P	FNC 77	**PR**	ASCII 打印输出
✔					
5 步			指令格式： **PR** (S) (D)		

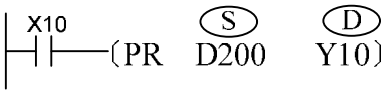
该指令将指定变量单元的数值，通过Y输出端口以同步方式逐个字节对外输出。其中：

Ⓢ为待输出的变量单元起始地址，长度共为4个（M8027＝1）或8个单元（M8027＝1）；

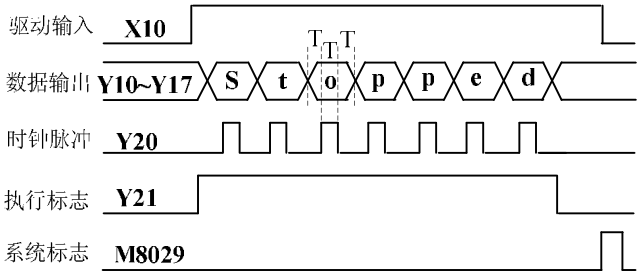
Ⓓ为进行输出打印的的Y端口起始号。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ											✔	✔	✔		
Ⓓ		✔													

例如指令：



若D200~D2003中的ASCII码为“Stopped”，则对应的输出端口信号及其时序如下图：



T：程序扫描执行时间

使用说明：

- 必须使用晶体管输出型 PLC，才能完成该指令功能；
- 打印过程中，驱动信号 X10 变为 OFF 时，打印输出即被中断。X10 再次为 ON 时，打印动作重新开始；
- 若置特殊寄存器 M8027 为 ON，则采用只发送每个 16bit 变量中的低字节，最多可以处理 8 个 16bit 变量；
- 打印输出过程中，遇到“00”的字符时，会自动结束打印操作；而 M8029 完成标志只有在驱动能流信号无效后才会置 ON 一个扫描周期；
- 指令按扫描周期（图中 T）执行，若扫描周期短时，请用恒定扫描模式；若过长则可以在定时中断程序中执行。

16bit	32bit	P	FNC 78	**FROM**	BFM 区读取
✔	✔	✔			
9 步	17 步	17 步	指令格式：　**FROM**　Ⓜ1　Ⓜ2　Ⓓ　Ⓝ		

该指令用于读取特殊扩展模块的BFM区寄存器的数据读取操作。其中：

Ⓜ1为特殊扩展模块的地址编号，取值范围0～7，最靠近主模块的为0，依次编号，最多允许有7个特殊模块；

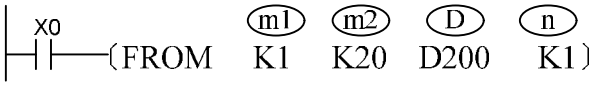
(m2)为特殊模块内BFM的寄存器地址号，取值范围0～32767；

(D)为读取参数在主模块中的存放地址，当读取的寄存器数多于1个时，占用随后的单元；

(n)本次读取参数的个数（按Word计算），取值范围1～32767，为按寄存器地址依次读取。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)								✔	✔	✔	✔	✔	✔	✔	✔
m1=0～7； m2=0～32767； n=1～32767; 位元件指定时，16bit 指令，可用 K1~K4；32bit 指令，可用 K5~K8															

例如指令：



表示当X0为ON时，将#1号特殊模块中的第20号寄存器（16bit宽度）的内容读取后，存放到主模块的D200寄存器中，只读取1个Word的数据。当X0为OFF时，不执行操作。

当用32bit指令时，(D)指定的地址为低16bit地址，(D)＋1为高16bit地址。

16bit	32bit	P	FNC 79	**TO**	BFM 区写入
✔	✔	✔			
9 步	17 步	17 步	指令格式：　**TO**　(m1) (m2) (D) (n)		

该指令用于向特殊扩展模块的BFM区寄存器的作数据写入操作。其中：

(m1)为特殊扩展模块的地址编号，取值范围0～7，最靠近主模块的为0，依次编号，最多允许有7个特殊模块；

(m2)为特殊模块内BFM的寄存器地址号，取值范围0～32767；

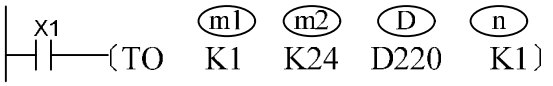
(D)为主模块中参数寄存器地址，其参数作为写操作数据的来源。当写操作的寄存器数多于1个时，会读取随后单元的参数；

(n)本次写入参数的个数（按Word计算），取值范围1～32767，为按寄存器地址依次写入。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

Ⓓ								✔	✔	✔	✔	✔	✔	✔	✔
m1=0～7； m2=0～32767； n=1～32767; 位元件指定时，16bit 指令，可用 K1~K4；32bit 指令，可用 K1~K8																

例如指令：



表示当X1为ON时，将主模块的D220寄存器中的数据，写入到#1号特殊模块中的第24号寄存器中，只读取1个Word的数据。当X1为OFF时，不执行操作。

当用32bit指令时，Ⓓ指定的地址为低16bit地址，Ⓓ＋1为高16bit地址。

Ⓝ表示操作的参数个数，16bit指令中，n＝2表示2 Word；而32bit指令中，n＝1即表示2 Word的数据，请注意。

关于FROM/TO指令的使用说明：

与FROM/TO指令有关的特殊寄存器的说明：

1. M8164 （FROM /TO 指令的传送点数可变模式）

   若M8164＝ON，执行FROM/TO指令时，特殊数据寄存器D8164（FROM/TO指令的传送点数指定寄存器）的内容作为传送点数n进行处理；

2. M8028（允许FROM/TO指令执行时中断插入设置标志）

   若M8028 = OFF时， FROM/ TO 指令执行时自动进入中断禁止状态，输入中断或定时器中断将不能执行。这期间发生的中断在 FROM/TO 指令完成后，立即执行。另外， FROM 、TO 指令也可以在中断程序中使用。

   若M8028 = ON时， FROM/TO 指令执行时，如发生中断则执行中断程序，但是中断程序中不能使用 FROM/TO指令.

3. 用FROM/TO指令访问扩展模块是比较耗时的操作，执行多个 FROM / T0 指令或传送多个缓冲存储器数据时，运算时间会延长。为了防止运行超时，可在FROM / T0前加入延长监视定时器时间的指令，或者错开 FROM / T0 指令的执行时间。

16bit	32bit	P	FNC 80	**RS**	串行数据传送
✔					
9 步			指令格式： **RS** Ⓢ Ⓜ Ⓓ Ⓝ		

该指令是一个通讯收发指令，将指定寄存器区域的数据，自动向串口依次发送，将串口接收到的数据存放到指定区域，相当于用户程序直接访问通讯缓冲区，借助用户程序对通讯收发缓冲区的处理，实现自定义协议的通讯。H2U系列对RS指令还有扩展功能，当选择了MODBUS主站协议，此时RS指令即为MOD指令功能。其中：

Ⓢ为待发送数据存放的寄存器区的起始地址；

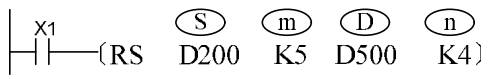
ⓜ为待发送数数据的长度（字节数），取值范围0～256；

Ⓓ为通讯接收数据的存放寄存器区的起始地址；

ⓝ本通讯接收的数据长度（字节数），取值范围0～256。

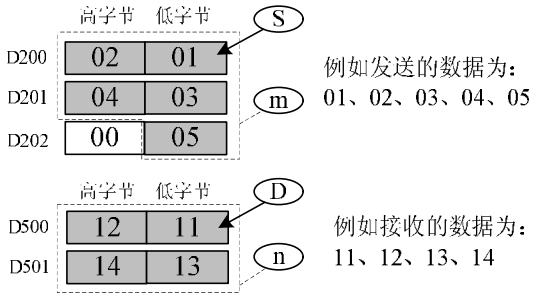
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ													✔		
ⓜ					✔	✔							✔		
Ⓓ													✔		
ⓝ					✔	✔							✔		

例如指令：

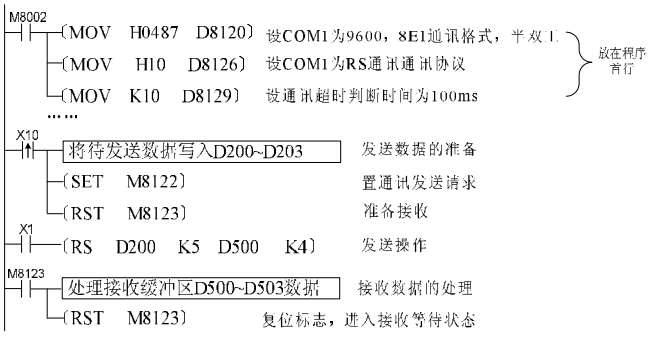


对于H2U系列PLC，RS指令只能用于COM1通讯口；COM0通讯口不支持RS指令。

当X1为ON时，指令执行后通讯的收发数据存放如右图。



实际编程时，需要作一些串行通讯的配置和准备，如设定串口的收发模式、波特率、位数、校验位、软件协议的设定、超时判断条件、收发缓冲区的数据准备、收发标志处理等，才能按预期的要求进行通讯。仍以上述语句为例，一个比较完整的RS通讯设置程序如下：



通讯端口的硬件格式和协议选择，涉及许多专用寄存器和标志，说明如下：

1、 串口的通讯格式设定：

项目	设定位	定义	COM0端口说明	COM1端口说明
			D8110	D8120
数据位	Bit0	0b-7Bits	MODBUS-RTU从站协议及指令只支持8位数据位，否则将造成通信出错	
		1b-8Bits		
校验位	Bit2~Bit1	00b-无校验(N)		
		01b-奇校验(O)		
		11b-偶校验(E)		
停止位	Bit3	0-1Bits		
		1-2Bits		
波特率	Bit7~Bit4	0011b-300 bps		
		0100b-600 bps		
		0101b-1200 bps		
		0110b-2400 bps		
		0111b-4800 bps		
		1000b-9600 bps		
		1001b-19200 bps		
		1010b-38400 bps		
		1011b-57600 bps		
		1100b-115200 bps		
起始字符	Bit8	0b-无		
		1b-(D8124)		
终止字符	Bit9	0b-无		
		1b-(D8125)		
半双工/全双工	Bit10	0b-全双工		RS指令适用
		1b-半双工		

2、 协议的设定：

COM0通讯口的可用协议设置：

COM0协议	D8116设定	半/全双工模式	通信格式
下载协议/HMI监控协议	01h	由跳线JP0决定	固定为“7E1”
MODBUS-RTU从站	02h	半双工	由D8110决定
MODBUS-ASC从站	03h	半双工	由D8110决定
其他协议（含RS指令）	不支持		

COM1通讯口的可用协议设置：

COM1协议	D8126设定	半/全双工模式	通信格式
RS指令	00h	由D8120的Bit10设定*	由D8120决定
HMI监控协议	01h	半双工	固定
并联协议主站	50h	半双工	固定
并联协议从站	05h	半双工	固定
N:N协议主站	40h	半双工	固定
N:N协议从站	04h	半双工	固定
计算机链接协议	06h	半双工	由D8120决定
MODBUS-RTU从站	02h	半双工	
MODBUS-ASC从站	03h	半双工	
RS指令	10h	由D8120的Bit10设定*	
MODBUS RTU指令	20h	半双工	
MODBUS-ASC指令	30h	半双工	

*RS指令半双工/全双工模式由D8120的Bit10设定：

1:半双工RS485（使用标配端口）

0:全双工RS232C/RS422（需要使用扩展小板H2U-232BD或H2U-422BD）

3、 通讯相关的特殊寄存器

寄存器	功能定义	寄存器	功能定义
M8110	保留	D8110	通讯格式，界面配置设定，默认为0
M8111	发送等待中（RS指令）	D8111	站号设置，界面配置设定，默认为1
M8112	发送标志（RS指令） 指令执行状态（MODBUS）	D8112	传送剩余数据数量（仅对RS指令）
M8113	接收完成标志（RS） 通讯错误标志（MODBUS）	D8113	接收到的数据数量（仅对RS指令）
M8114	接收中（仅对RS指令）	D8114	起始字符STX（仅对RS指令）
M8115	保留	D8115	终止字符ETX（仅对RS指令）
M8116	保留	D8116	通讯协议设定，界面配置设定，默认为0

M8117	保留	D8117	计算机链接协议接通要求数据起始地址号
M8118	保留	D8118	计算机链接协议接通要求发送数据数量
M8119	超时判断	D8119	通讯超时时间判断，界面配置设定，默认为10（100ms）

4、 协议的选择方法：

上面描述了各串口支持的协议种类，但每个通讯口在同一时间只能支持一种协议，而协议类型必须事先选择，才能保证通讯正常。这些协议的确定原则如下：

4.1 COM0的适用协议的确定原则：

① 停机状态，协议固定为下载协议/HMI监控协议；

② 停机转运行时，若跳线JP0接通，协议为下载协议(或称HMI监控协议)；

③ 停机转运行时，若跳线JP0断开，协议由D8116决定，D8116在PLC第一个扫描周期内确定的值对协议有效，运行后D8116的更改不能改变协议，D8116与协议对应关系见协议设置表；

④ PLC运行后，协议将维持不变。

4.2 COM1的适用协议的确定原则：

在停机状态或第一次运行时，系统按优先级检查协议设置:

① N:N协议；

② 并联主站协议；

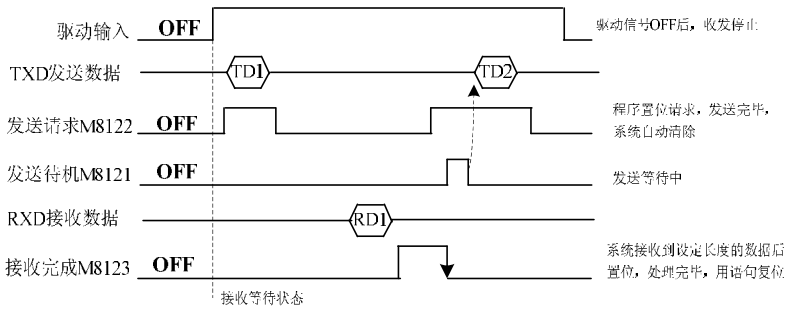
③ 并联从站协议；

④ 计算机链接协议

若存在优先级别较高的协议，就按检测到的协议进行通讯，将不再检查优先级低的协议，若上述协议均没有配置，则按：

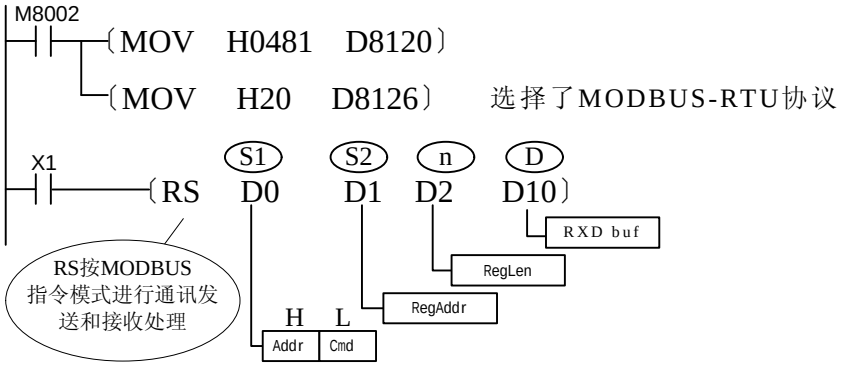
⑤ COM1口的通讯协议由D8126决定。一旦PLC运行后，协议将维持不变；

5、 通讯相关的专用标志的时序与操作说明；



6、 RS指令的扩展功能：

当COM1口的通讯协议配置中选择了MODBUS主站协议（D8126＝H20或H30）时，RS指令将以MODBUS通讯协议进行通讯。通讯过程中占用的寄存器定义与标准RS指令不同，请予注意：



(S1) 为从机地址（高字节）、通讯命令（低字节，按MODBUS协议定义）；

(S2) 为访问的寄存器地址号；

(n) 为读或写的数据长度，按Word计算；

(D) 为读回来数据的存放单元起始地址，占用后续地址单元，长度由 (n) 决定。

操作数	字　元　件										
	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)									✔		
(S2)	✔	✔							✔		
(n)									✔		
(D)									✔		

该指令用于具有MODBUS－RTU协议从站设备（如MD320/300/280等系列变频器）进行通讯，将

非常方便，在Ⓓ单元即可直接获取从机的数据。

编程举例：应用中，PLC需要不断的读#1从机内，地址为100的寄存器，数据存于D10单元，编程时初始化是：

D8126 = H0020　　设定通信协议为MODBUS RTU指令

D8120 = H0081　设定COM1通信格式为：9600，8N1

D0 = H0103　　Addr&Cmd从机地址为01和MODBUS命令码为03，读寄存器

D1 = H0064　　RegAddr要操作的从机的寄存器地址

D2 = H0001　　RegLen要操作的寄存器的个数

D10　　Buf本PLC数据缓冲区，本例中读命令通信成功后数据存于D10

通讯发送、接收过程中不占用用户寄存器缓存，由系统自行处理。

❙ 使用说明：

程序中可多次使用RS指令，但一次只能有一处RS指令被使能，否则通讯信息会错乱。

16bit	32bit	P	FNC 81	**PRUN**	并联运行
✔	✔	✔			
5 步	9 步	9 步	指令格式： **PRUN** Ⓢ Ⓓ		

该指令是将Ⓢ起始连续地址的位变量（以8进制为宽度单位），成批复制到Ⓓ起始位变量组中。其中：

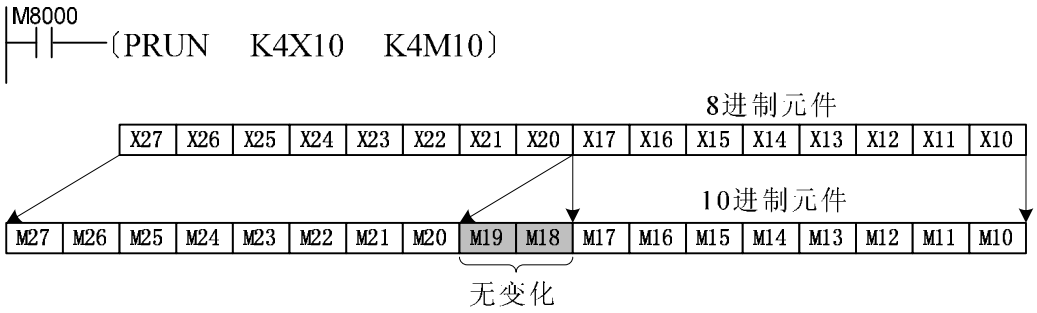
Ⓢ为待复制的位变量的起始地址，要求地址的个位必需为0，如X10，M20等；

Ⓓ为复制的目的位变量起始地址，同样要求地址的个位必需为0，如M30，Y10等。

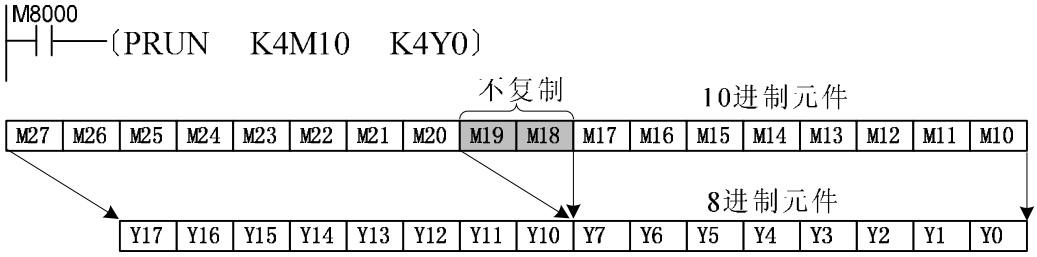
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ							✔		✔						
Ⓓ								✔	✔						

其中Kn中，允许n＝1～8。

例如指令1：



例如指令2：



16bit	32bit	P	FNC 82	**ASCI**	HEX－ASCII 转换
✔		✔			
7 步		7 步	指令格式： **ASCI** (S) (D) (n)		

该指令是将(S)的值转换成ASCII码后，存储到(D)为起始地址的变量中。其中：

(S)为待转换的变量地址或常数数值；

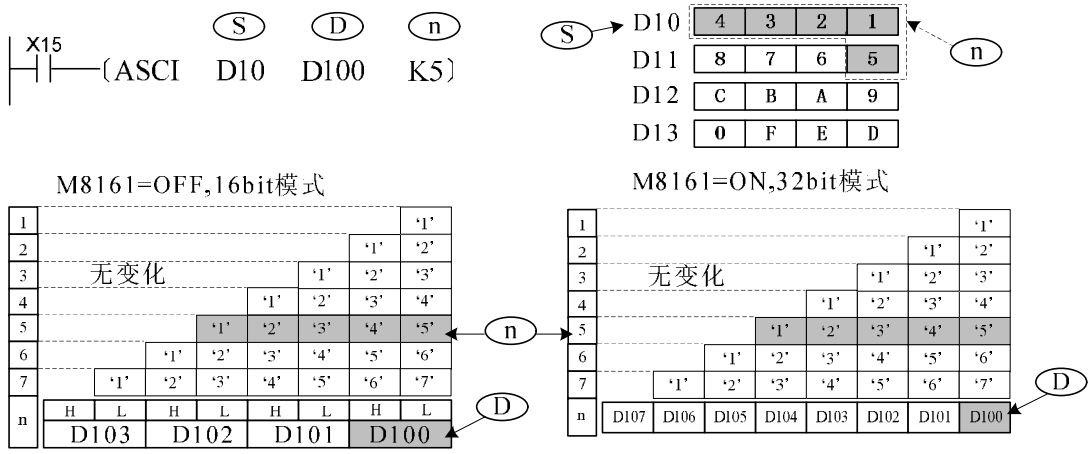
(D)为转换后ASCII码的存放起始地址；

(n)为转换的字符位数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，n=1~256														

ASCI数值转换遵照ASCII与HEX进制数值对照表，如：ASCII‘0’对应HEX‘H30’;ASCII‘F’对应HEX‘H46’等。

例如指令：



其中，M8161标志决定了计算结果存放目的变量的宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节分别存储；当M8161=ON时，为8bit模式，只有变量的低字节存储结果，因此实际使用变量区域的长度增加。

注：**RS/ HEX/ ASCI/ CCD**等指令共用M8161模式标志，编程时注意。

16bit	32bit	P	FNC 83	**HEX**	ASCII－HEX 转换
✔		✔			
7 步		7 步	指令格式： **HEX** S D n		

该指令是将S起始变量的值转换成ASCII码后，存储到D为起始地址的变量中，转换的字符数、存储模式可以设定。其中：

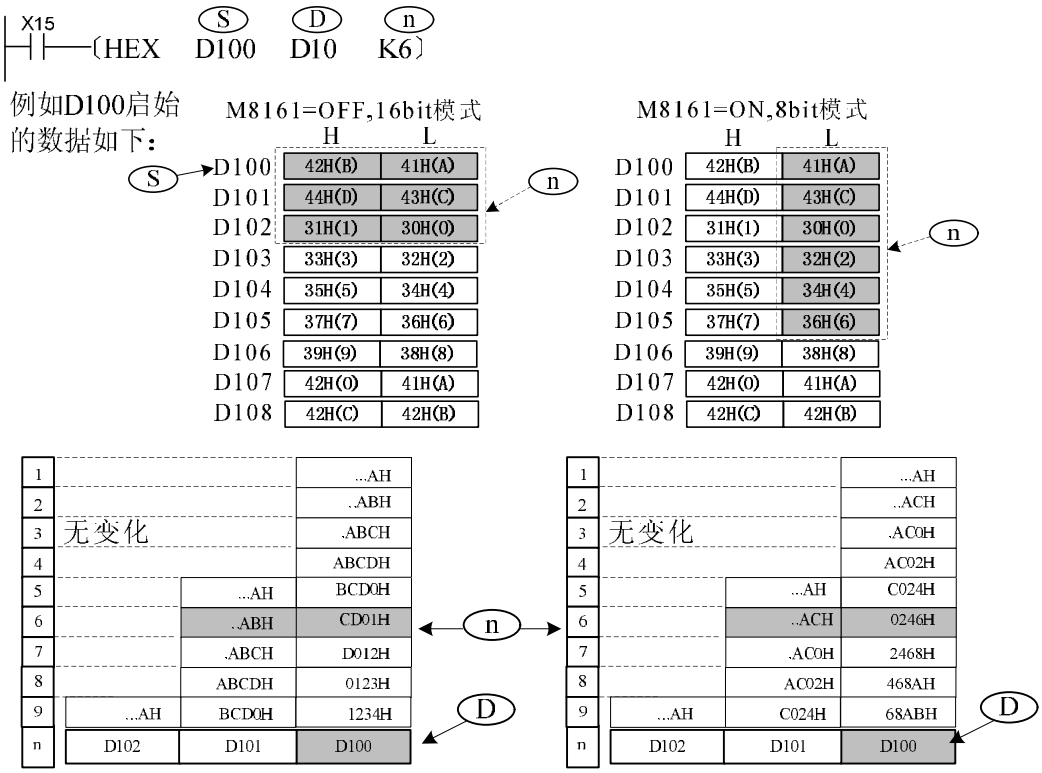
S为待转换的变量地址或常数数值，若为寄存器变量，以32bit变量宽度（即4个ASCII字符）为单位进行转换分隔；

D为转换后ASCII码的存放起始地址，占用的变量空间与n有关；

n为转换的ASCII字符位数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
D								✔	✔	✔	✔	✔	✔	✔	✔
n	常数，n=1~256														

例如指令：



其中，M8161标志决定了变量宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节都参与运算；当M8161=ON时，为8bit模式，只有变量的低字节参与运算，高字节的内容丢弃，因此实际使用变量区域的长度增加。

注：

RS/ HEX/ ASCI/ CCD等指令共用M8161模式标志，编程时注意；

Ⓢ数据区的源数据必需为ASCII码字符，否则转换出错；

若输出的数据为BCD格式，HEX转换后，需要进行BCD－BIN转换，才是正确的数值。

16bit	32bit	P	FNC 84	**CCD**	求校验和
✔		✔			
7 步		7 步	指令格式： **CCD** Ⓢ Ⓓ ⓝ		

该指令是对Ⓢ启始的ⓝ个变量进行两种校验和运算，将直接加法的求和运算结果存于Ⓓ；将逐个异或逻辑运算的结果存于Ⓓ＋1单元中。其中：

Ⓢ为待求校验和运算的变量启始地址，使用后续地址的变量单元；

Ⓓ用于存放校验和；Ⓓ＋1单元用于存放逐项异或逻辑运算结果；

(n)为用于校验的变量字节数。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，n=1~256														

例如指令：

M8161标志决定了变量宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节都参与运算；当M8161=ON时，为8bit模式，只有变量的低字节参与运算，高字节的内容被丢弃，因此实际使用变量区域的长度增加，见下例图；

“累加和”就是将指定的n个变量直接相加的计算结果；

“异或”逻辑运算的方法则是：

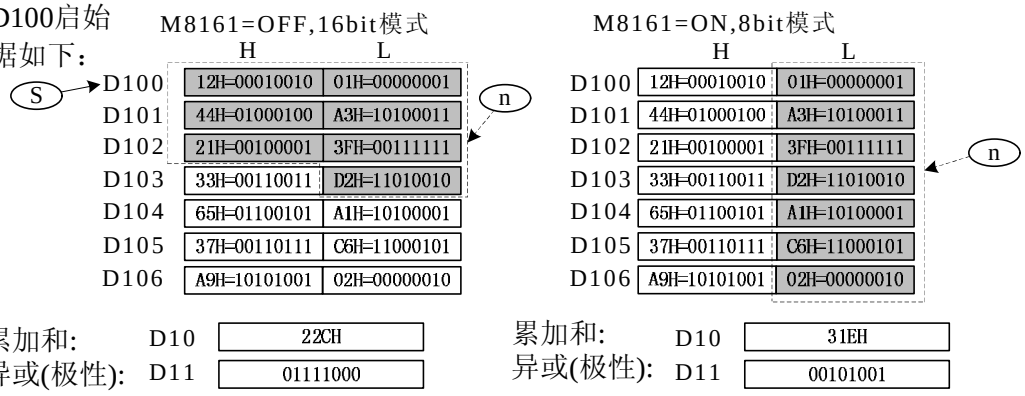
1） 将参与运算的变量都换算为二进制数；

2） 先统计每个变量的bit0为1的个数，若为偶数个，则异或结果的bit0为0；若为奇数个，则异或结果的bit0为1；

3） 再统计每个变量的bit1为1的个数，若为偶数个，则异或结果的bit1为0；若为奇数个，则异或结果的bit1为1；

4） 依次类推，逐个计算bit2～bit7，所得的二进制数换算为HEX数值即为异或结果（或称极性值）。

例如D100启始的数据如下：



- **RS/ HEX/ ASCI/ CCD**等指令共用M8161模式标志，编程时注意该标志的处理。

16bit	32bit	P	FNC 88	**PID**	PID 运算
✔					
9 步			指令格式： **PID** (S1) (S2) (S3) (D)		

该指令是完成PID运算，用于闭环控制系统参数的控制。其中：

(S1)为PID控制的目标值；

(S2)为实测的反馈值；

(S3)为PID运算所需设定参数、中间结果保持的缓存区启始地址，会占用随后地址的共25个变量单元，故取值范围是D0~D7975，最好是非电池保持区，否则首次启动运算前需对缓存区作初始化清0处理。其中每个单元的功能和参数描述详见本节说明；

(D)为PID计算结果的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)													✔		
(S2)													✔		
(S3)													✔		
(D)													✔		

例如指令：

其中参数说明如下：

D9中存放的是PID调节的目标值，D10为闭环反馈值，注意D9、D10必需为同一量纲，如都为0.01MPa单位，或都为1℃单位等等；

D100～D124共25个单元用于存放PID运算的设定值与过程值，这些值在首次PID计算前就必需逐项进行设定；

D130单元则用于存放计算后的控制输出值，用于控制执行的动作。

就(S3)启始的各单元参数值的功能和设定方法说明如下：

单元	功能	设定说明
(S3)	采样时间（TS）	设定范围 1～32767（ms），但需大于 PLC 程序扫描周期
(S3)+1	动作方向（ACT）	bit0：0＝正动作；1＝逆动作 bit1：0＝输入变化量报警无效；1＝输入变化量报警有效 bit2：0＝输出变化量报警无效；1＝输出变化量报警有效 bit3：不可使用 bit4：0＝自动调谐不动作；1＝执行自动调谐 bit5：0＝输出值上下限设定无效；I＝输出值上下限设定有效 bit6 ~ bit15 不可使用 另外，请不要使 bit5 和 bit2 同时处于 ON 。
(S3)+2	输入滤波常熟（α）	0~99[%]， 0＝没有输入滤波
(S3)+3	比例增益（Kp）	1~32767[%]
(S3)+4	积分时间（T1）	0~32767（×100ms），0＝作为∞处理（无积分）
(S3)+5	微分增益（KD）	0~100[%]，0＝无积分增益
(S3)+6	微分时间（TD）	0~32767（×10ms），0＝无微分处理
(S3)+(7～19)	PID 运算的内部处理占用，首次运行前应清为 0	
在<ACT>的 bit1=1，bit2=1 或 bit5=1 时，(S3)+（20～24）被占用，定义如下：		
(S3)+20	输入变化量（增侧）报警设定值	0~32767，（<ACT>的 bit1=1 时有效）
(S3)+21	输入变化量（减侧）报警设定值	0~32767，（<ACT>的 bit1=1 时有效）
(S3)+22	输出变化量（增侧）报警设定值	0~32767，（<ACT>的 bit2=1，bit5=0 时有效）
		输出上限设定值 -32768~32767 （<ACT>的 bit1=1，bit5=1 时有效）
(S3)+23	输出变化量（减侧）报警设定值	0~32767（S3+1<ACT>的 bit2=1，bit5=0 时有效）
		输出上限设定值-32768~32767 （<ACT>的 bit1=0，bit5=1 时有效）
(S3)+24	报警输出	bit0 输入变化量（增侧）溢出 bit1 输入变化量（减侧）溢出 Bit2 输出变化量（增侧）溢出 Bit3 输出变化量（减侧）溢出 （<ACT>的 bit1=1 或 bit2=1 时有效）

PID计算的理论公式：

正逻辑	$MV = K_P * E(t) + K_D * PV(t)S + K_I * E(t)\frac{1}{S}$ $E(t) = SV - PV$
负逻辑	$MV = K_P * E(t) + K_D * PV(t)S + K_I * E(t)\frac{1}{S}$ $E(t) = PV - SV$

其中：

PV ： 当前反馈值；　　*SV* ： 设定的目标值

E(t) ： 反馈与设定值的偏差。正逻辑时*E(t) = PV- SV* ，负逻辑时*E(t) = PV- SV*

Kp ： 比例增益；　　*Ti*： 积分时间；　　*Kd*： 微分增益

E(t) S：的积分值；　　*PV(t)/S* ： *PV(t)*的微分值

MV ： PID输出值

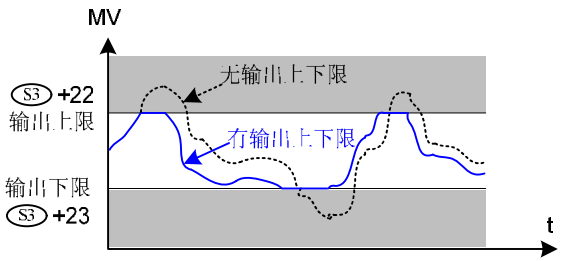
PID的动作(ACT)逻辑：

正逻辑（也称正方向）：当实测反馈值（PV）大于设定目标值（SV）是，调节输出（MV）必需下降，才能达到平衡，这种PID运算方式称为正逻辑。如变频驱动的恒压供水系统的频率调节，恒温控制系统的加热功率调节等，属于正逻辑PID控制。

负逻辑（也称反方向）：当实测反馈值（PV）大于设定目标值（SV）是，调节输出（MV）必需上升，才能达到平衡，这种PID运算方式称为负逻辑。如恒温控制系统的散热风扇运转速度控制，属于负逻辑PID控制。

输出上下限设定：

对于多数应用系统，PID的调节输出范围是有限的，如电平信号幅值范围、变频器调节频率范围等。根据实际情况设定PID运算中的输出上限、输出下限，让闭环系统的执行装置在正常运行参数下工作。



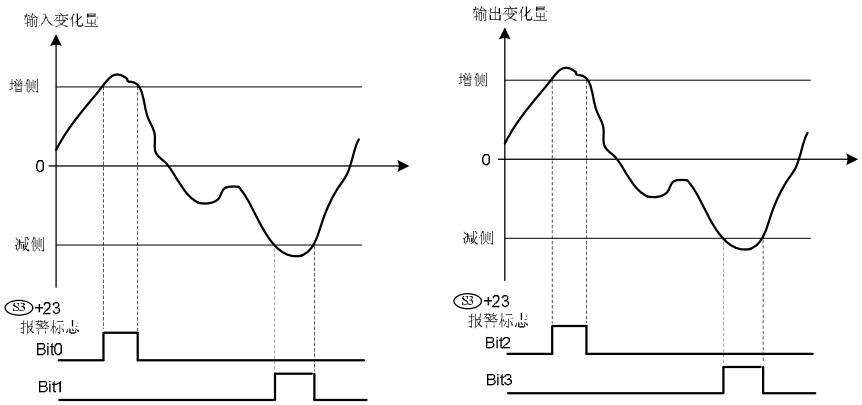
超限报警设定：

若需要检查控制器的输入量（反馈量）的变化量是否超限（上限或下限）、PID 的运算输出的变化量是否过大（超过上限或下限），可在 ACT（S3 +1）单元中，将 ACT 的 bit1＝

ON，bit2＝ON，启用报警功能；在Ⓢ3＋（20～23）单元中分别设定变化量报警限值，运行中就可在Ⓢ3＋24 单元读取参数的超限状态了。这在一些需要判断调节状态的场合，简化了运算。

使用输出变化量的报警功能时，Ⓢ3+1（ACT）的 bit5 请必须被设置为 OFF。

这里所述的“变化量”＝（上次的数值）－（本次的数值），对应动作示意图如下：



可见“减侧”的告警下限设定值为负值。为避免输入量波动过大，一般对输入的信号进行硬件滤波处理，还可在用户程序中进行软件滤波处理。

PID常数的设定：

PID的Kp、Ki、Kd、Ts等常数选择，对PID控制特性至关重要，这三个参数对稳定性的定性影响为：

1） 比例增益Kp过大，将出现过冲，甚至无法稳定；Kp过小，调节过程缓慢；设置恰当时，系统PID调节的响应速度效果好；

2） 积分时间Ti过大，调节过程缓慢；Ti过小，输出突变大，难以稳定；设置恰当时，稳定快，偏差小；

3） 微分增益Kd过大，系统极易自激振荡，无法稳定；Kd小对稳定影响小；设置恰当时利于快速逼近设定值。由于该参数对稳定的影响大，多数应用中常不使用微分调节，即将其增益设为0；

4） 采用周期Ts对应于执行PID运算的时间间隔，其设定值应大于PLC程序的扫描执行时间，为保证PID调节效果，PLC程序最好采用固定扫描周期的模式运行，或将PID运行放在定时中断中运行。

编程说明：

PID指令在程序中可多次使用，可同时执行，但各PID指令中使用的Ⓢ3变量区域不要有重叠；在定时中断、子程序中执行PID指令时，需事先清除Ⓢ3＋7缓存单元。

16bit	32bit	P	FNC 110	**ECMP**	二进制浮点数比较
	✔	✔			
	13 步	13 步	指令格式： **ECMP** (S1) (S2) (D)		

该指令是进行2个浮点数变量的比较，将比较的结果输出到(D)启始的3个变量中。其中：

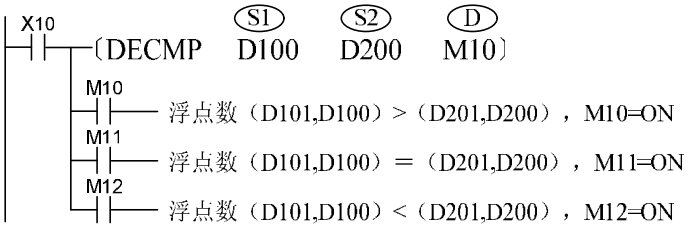
(S1)为待比较的二进制浮点数1；

(S2)为待比较的二进制浮点数2；

(D)为比较结果的存放单元，共占用3个（位）变量单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔							✔		
(S2)					✔	✔							✔		
(D)		✔	✔	✔											

例如指令：



X10=OFF 时，不执行DECP指令，M10~M12仍保持X10=OFF之前的状态

若(S1)或(S2)为 K、H 常数，系统会自动转换为浮点数参与运算。

16bit	32bit	P	FNC 111	**EZCP**	二进制浮点数区域比较
	✔	✔			
	13 步	13 步	指令格式： **EZCP** (S1) (S2) (S) (D)		

该指令是进行二进制浮点数变量的区间比较，将比较的结果输出到(D)启始的3个变量中。其中：

(S1)为二进制浮点变量区间的下限；

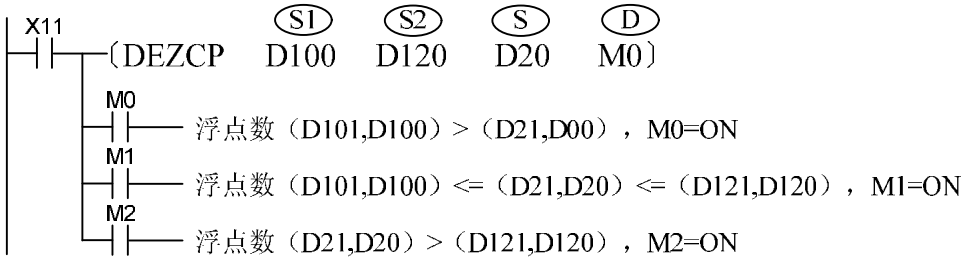
(S2)为二进制浮点变量区间的上限；

(S)待比较的二进制浮点变量；

(D)为比较结果的存放单元，共占用3个（位）变量单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔							✔		
(S2)					✔	✔							✔		
(S)					✔	✔							✔		
(D)		✔	✔	✔											

例如指令：



若X11=OFF时，不执行DEZCP指令，M10/M11/M12保持X11=OFF以前的数值不变。

16bit	32bit	P	FNC 118	**EBCD**	二进制浮点数→十进制浮点转换
	✔	✔			
	9 步	9 步	指令格式：**EBCD** (S) (D)		

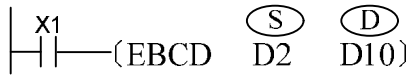
该指令是进行二进制浮点数转换为十进制浮点的运算。其中：

(S)为二进制浮点变量；

(D)为转换为十进制浮点数结果的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

例如指令：



将二进制浮点数（D3,D2）转换成十进制浮点数后，存放于（D11,D10）单元。

PLC 内部浮点数据计算均为二进制形式，转换为十进制，可方便监控。

16bit	32bit	P	FNC 119	**EBIN**	十进制浮点数→二进制浮点转换
	✔	✔			
	9 步	9 步	指令格式： **EBIN** (S) (D)		

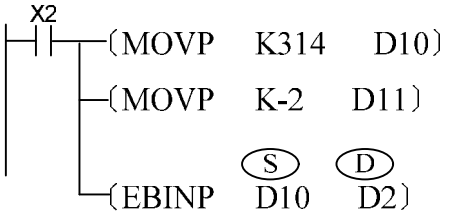
该指令是进行十进制浮点数转换为二进制浮点的运算。其中：

(S)为十进制浮点变量；

(D)为转换为二进制浮点数结果的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

例如指令：



将十进制浮点数3.14（先放于D11,D10）转换成二进制浮点数后，存放于（D3,D2）单元。

16bit	32bit	P	FNC 120	**EADD**	二进制浮点加法运算
	✔	✔			
	13 步	13 步	指令格式： **EADD** (S1) (S2) (D)		

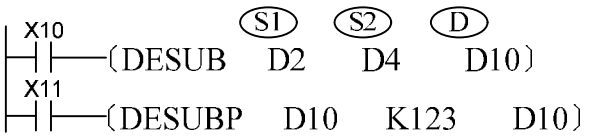
该指令是进行二进制浮点的加法运算。其中：

(S1)和(S2)分别为二进制浮点的加数和被加数；

(D)为二进制浮点加法和的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔							✔		
(S2)					✔	✔							✔		
(D)													✔		

例如指令：



当 X1＝ON 时，二进制浮点数（D3,D2）与二进制浮点数（D5,D4）相加后，二进制浮点数和存放于（D11,D10）；

当 X2 由 OFF 变为 ON 时，二进制浮点数（D11,D10）的值增大 4321。这里常数 K4321 运算前已自动被调整为二进制浮点数；

和的存放单元可以与加数或被加数为同一单元，若采用连续执行指令，则每程序每扫描一次，计算就会被执行一次。

16bit	32bit	P	FNC 121	**ESUB**	二进制浮点减法运算
	✔	✔			
	13 步	13 步	指令格式： **ESUB** (S1) (S2) (D)		

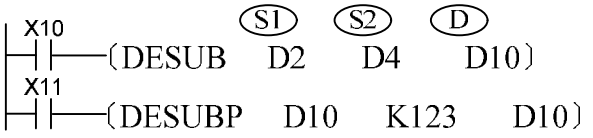
该指令是进行二进制浮点的减法运算。其中：

(S1) 和 (S2) 分别为二进制浮点的被减数和减数；

(D) 为二进制浮点减法差的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔							✔		
(S2)					✔	✔							✔		
(D)													✔		

例如指令：



当 X10＝ON 时，二进制浮点数（D3,D2）减去二进制浮点数（D5,D4）后，二进制浮点数差存放于（D11,D10）；

当 X11 由 OFF 变为 ON 时，二进制浮点数（D11,D10）的值减小 123。这里常数 K123 运算前已自动被调整为二进制浮点数；

差的存放单元可以与减数或被减数为同一单元，若采用连续执行指令，则每程序每扫描一次，

计算就会被执行一次。

16bit	32bit	P	FNC 122	**EMUL**	二进制浮点乘法运算
	✔	✔			
	13 步	13 步	指令格式： **EMUL** (S1) (S2) (D)		

该指令是进行二进制浮点的乘法运算。其中：

(S1)和(S2)分别为二进制浮点的被乘数和乘数；

(D)为二进制浮点乘法积的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔							✔		
(S2)					✔	✔							✔		
(D)													✔		

例如指令：

```
X12
─┤├──〔DEMUL   D2    D4    D10〕
X13                (S1)  (S2)  (D)
─┤├──〔DEMULP  D10   K3    D10〕
```

当 X12＝ON 时，二进制浮点数（D3,D2）乘以二进制浮点数（D5,D4）后，二进制浮点数积存放于（D11,D10）；

当 X13 由 OFF 变为 ON 时，二进制浮点数（D11,D10）的值乘以 3 倍后存回（D11,D10）。这里常数 K3 运算前已自动被调整为二进制浮点数；

积的存放单元可以与乘数或被乘数为同一单元，若采用连续执行指令，则每程序每扫描一次，计算就会被执行一次。

16bit	32bit	P	FNC 123	**EDIV**	二进制浮点除法运算
	✔	✔			
	13 步	13 步	指令格式： **EDIV** (S1) (S2) (D)		

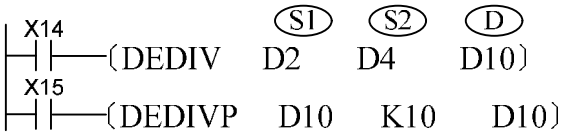
该指令是进行二进制浮点的除法运算。其中：

(S1)和(S2)分别为二进制浮点的被除数和除数；

(D)为二进制浮点除法商的存放单元启始地址。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔							✔		
(S2)					✔	✔							✔		
(D)													✔		

例如指令：



当 X14＝ON 时，二进制浮点数（D3,D2）除以二进制浮点数（D5,D4）后，二进制浮点数商存放于（D11,D10）；

当 X15 由 OFF 变为 ON 时，二进制浮点数（D11,D10）的值除以 10 后存回（D11,D10）。这里常数 K10 运算前已自动被调整为二进制浮点数；

商的存放单元可以与除数数或被除数为同一单元，若采用连续执行指令，则每程序每扫描一次，计算就会被执行一次。

除数不得为 0，否则计算出错。

16bit	32bit	P	FNC 127	**ESQR**	二进制浮点开平方运算
	✔	✔			
	9 步	9 步	指令格式：　**ESQR**　(S)　(D)		

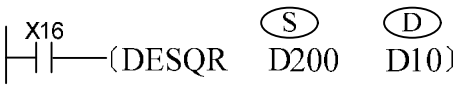
该指令是进行二进制浮点的开平方运算，即求二进制浮点数的平方根。其中：

(S)为待求平方根的二进制浮点数变量；

(D)为二进制浮点平方根的存储单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔							✔		
(D)													✔		

例如指令：



将二进制浮点数开方结果 $\sqrt{(D201,D200)}$ 存放到→ (D11,D10)

若Ⓢ为常数 K 或 H，开方运算前系统会自动将之转换成二进制浮点数。

16bit	32bit	Ⓟ	FNC 129	**INT**	二进制浮点→BIN 整数变换
✔	✔	✔			
5 步	9 步	9 步	指令格式： **INT** Ⓢ Ⓓ		

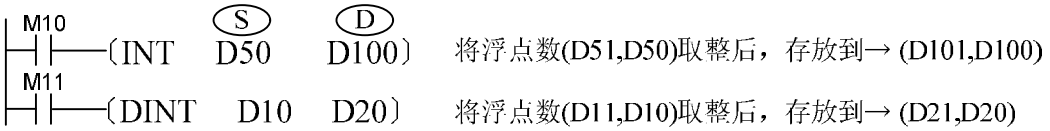
该指令是进行二进制浮点的取整运算，丢弃小数部分，将二进制结果存于Ⓓ中。其中：

Ⓢ为待取整变换的二进制浮点数变量；

Ⓓ为变换后BIN整数结果的存储单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ													✔		
Ⓓ													✔		

例如指令：



运算中，若结果为 0，零标志置 ON；转换时不满 1，有小数丢弃时，借位标志置 ON；

运算结果超过如下范围，发生溢出时，进位标志置 ON：

对于 16bit 指令：-32,768~32,767

对于 32 bit 指令：-2,147,483,648~2,147,483,647

16bit	32bit	Ⓟ	FNC 130	**SIN**	浮点 SIN 运算
	✔	✔			
	9 步	9 步	指令格式： **SIN** Ⓢ Ⓓ		

该指令是求指定角度（RAD，弧度）的SIN（正弦）值，变量为二进制浮点存储格式。其中：

Ⓢ为待求正弦值的角度变量，RAD单位，以二进制浮点数表示。取值范围0<= α <=2 π ；

Ⓓ为变换后SIN计算结果的存储单元，二进制浮点数格式。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ													✔		
Ⓓ													✔		

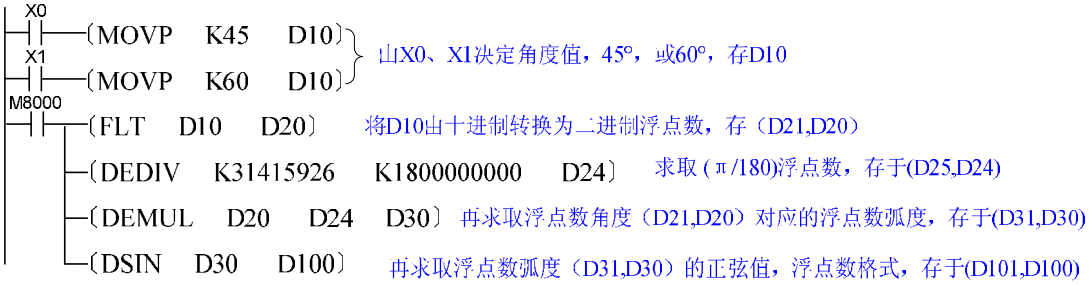
例如指令：

将弧度(D21,D20)求SIN值，存放到→ (D31,D30)

这里计算的源数据、SIN 结果都为二进制浮点数格式。

RAD(弧度)值＝角度×π/180°，如角度 360°对应的弧度＝360°×π/180°＝2π。

根据角度值求对应 SIN 值的程序举例：



16bit	32bit	P	FNC 131	**COS**	浮点 COS 运算
	✔	✔			
	9 步	9 步	指令格式：　**COS**　Ⓢ　Ⓓ		

该指令是求指定角度（RAD，弧度）的COS（余弦）值，变量为二进制浮点存储格式。其中：

Ⓢ为待求余弦值的角度变量，RAD单位，以二进制浮点数表示。取值范围0<= α <=2π；

Ⓓ为变换后COS计算结果的存储单元，二进制浮点数格式。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ													✔		
Ⓓ													✔		

例如指令：

将弧度(D21,D20)求COS值，存放到→ (D31,D30)

这里计算的源数据、COS 结果都为二进制浮点数格式。

RAD(弧度)值＝角度×π/180°，如角度 360°对应的弧度＝360°×π/180°＝2π。

关于以角度求取 COS 值的编程语句，可参考 SIN 指令中的举例。

16bit	32bit	P	FNC 132	**TAN**	浮点 TAN 运算
	✔	✔			
	9 步	9 步	指令格式： **TAN** (S) (D)		

该指令是求指定角度（RAD，弧度）的TAN（正切）值，变量为二进制浮点存储格式。其中：

(S)为待求正切值的角度变量，RAD单位，以二进制浮点数表示。取值范围0<=α<2π；

(D)为变换后TAN计算结果的存储单元，二进制浮点数格式。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

例如指令：

将弧度(D21,D20)求TAN值，存放到→(D31,D30)

这里计算的源数据、TAN 结果都为二进制浮点数格式。

RAD(弧度)值＝角度×π/180°，如角度 360°对应的弧度＝360°×π/180°＝2π。

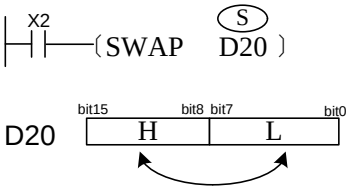
关于以角度求取 TAN 值的编程语句，可参考 SIN 指令中的举例。

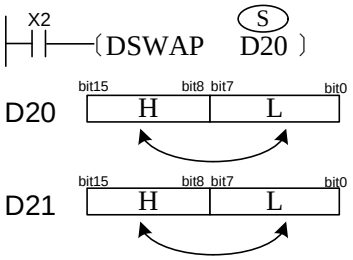
16bit	32bit	P	FNC 147	**SWAP**	高低字节交换
✔	✔	✔			
3 步	5 步	5 步	指令格式： **SWAP** (S)		

该指令是将指定变量(S)的高低字节的值进行交换。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)								✔	✔	✔	✔	✔	✔	✔	✔

例如指令：





16bit	32bit	P	FNC 155	**ABS**	ABS 位置数据读取
	✔				
	9 步		指令格式： **ABS** (S) (D1) (D2)		

该指令是通过高速输入端口读取伺服驱动器中电机的绝对位置（ABS）数据。其中：

(S)为读取伺服信号的启始位变量，占用后续共3个单元；

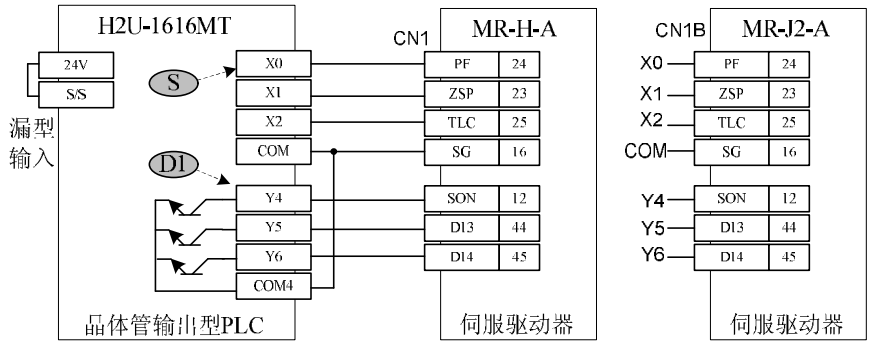
(D1)为读取伺服信号用的时钟脉冲及握手信号输出启始位变量，占用后续共3个单元；

(D2)则是从伺服读取数据的存储单元，32bit宽度，即还占用(D2)＋1单元，通常指定D8140。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✔	✔	✔	✔											
(D1)		✔	✔	✔											
(D2)								✔	✔	✔	✔	✔	✔	✔	✔

例如指令：

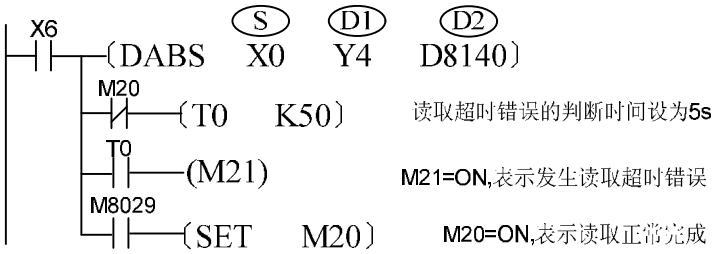
对应的硬件连接方法如下图，其中伺服驱动器为市售三菱公司产品，配合有绝对位置检测的编码器的伺服马达：



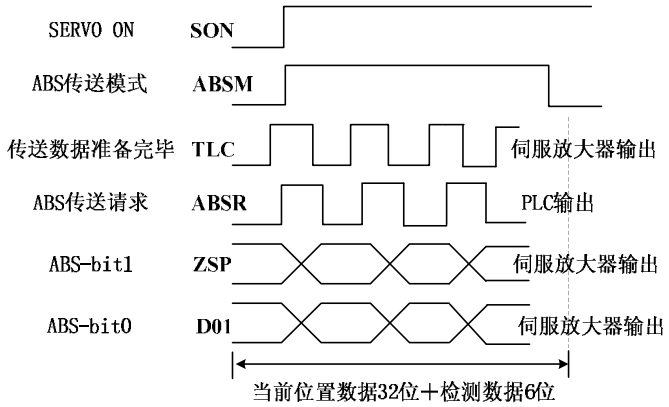
当指令驱动标志 M10 变为 ON 时开始读取，读取完毕，M8029 标志置为 ON；

当指令执行过程中，驱动标志变为 OFF，读取操作即被中断；

读取 ABS 数据的编程操作语句如下例，当 X6 端子闭合时才读取 ABS 数据，若 5s 内没有完成读取操作，超时标志 M20 被置位，编程如下：



ABS 读操作的信号时序如下图，指令执行时，PLC 会按该实现自动完成与伺服驱动器的访问操作：



❙ 该指令只能以 32bit 方式执行，即指令为 DABS。

16bit	32bit	P	FNC 156	**ZRN**	原点回归
	✔				
	9 步		指令格式： **ZRN** (S1) (S2) (S3) (D)		

该指令是PLC与伺服驱动器配合工作时，用指定脉冲速度和脉冲输出端口，让执行机构向动作原点（DOG）移动，直到遇到原点信号满足条件为止。其中：

(S1)为原点回归动作的启始速度。16bit指令时，范围是10～32,767Hz；32bit指令时，范围是10～100,000Hz；

(S2)为原点信号变为ON后的爬行速度，范围是10～32,767Hz；

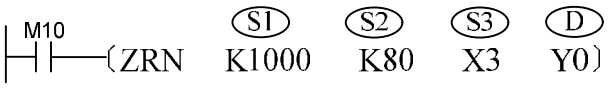
(S3)原点信号（DOG）输入，虽然XYMS信号都可以，但只有X信号的及时性最好；

Ⓓ为脉冲输出启始地址，MT型只能为Y0或Y1，MTQ型则可选Y0、Y1、Y2、Y3、Y4等。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ1					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Ⓢ2					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Ⓢ3	✔	✔	✔	✔											
Ⓓ		✔													

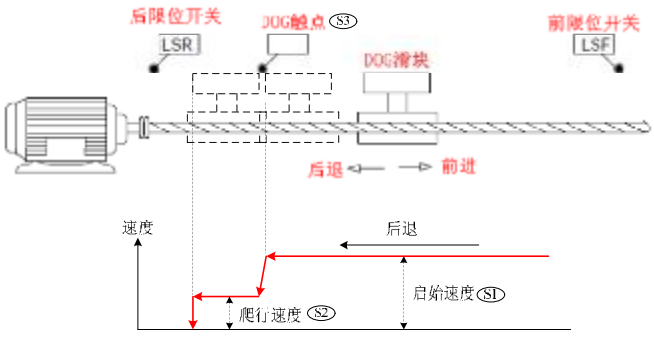
相对位置控制指令 DRVI（FNC158）和绝对位置控制指令 DRVA（FNC159）在执行时，控制器会计算自身已发出的正转脉冲或反转脉冲数，并将之保存在寄存器[D8141，D8140]（Y000）和[D8143，D8142]（Y001)。但该寄存器的数据在断电时会消失，故上电时和初始运行时，必须执行原点回归指令 ZRN，以事先将机械动作的原点位置的数据写入。

例如指令：



本指令的动作是，当 M10 变为 ON 后，PLC 从 Y0 高速输出端口，开始以 1000Hz 发出脉冲，令步进电机向原点作后退，当遇到 DOG 信号变为 ON 时（此时 DOG 滑块刚好碰到 DOG 触点），输出频率降为 80Hz，作低速爬行，直到 DOG 信号再变为 OFF 时，Y0 停止输出脉冲，向当前值寄存器（Y000：[D8141，D8140]，Y001：[D8143，D8142]）中写入 0 。另外， M8140（清零信号输出功能）ON 时，同时输出清零信号。随后，当执行完成标志（M8029）置为 ON 的同时，脉冲输出中监控（Y000 ：[M8147]，Y001：[M8148]）变为 OFF。

参见下图：



本指令执行中，涉及的系统变量有：

1．　D8141（高位），D8140（低位）]：Y000 输出的当前值寄存器（使用 32 位）

2．　D8143（高位），D8142（低位）]：Y001 输出的当前值寄存器（使用 32 位）

3．M8145 ：Y000 脉冲输出停止（立即停止）

4．M8146 ：Y001 脉冲输出停止（立即停止）

5．M8147 ：Y000 脉冲输出中监控（BUSY/READY）

6．M8148 ：Y001 脉冲输出中监控（BUSY/READY）

由于伺服驱动器对位置信息具有掉电保持功能，该指令并不需要每次上电时都需进行；指令执行中，仅能单方向运动（后退方向），故执行本指令前 DOG 滑块必需处于离开 DOG 点的位置。

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一的高速 Y 输出口进行操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需满足前一次驱动的定位指令所使用的[脉冲输出中监控（Y 000 ： [M81471]，Y001；[M8148]）]OFF 后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有 1 个周期以上的 OFF 时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」，再次扫描到该指令（运算）时，才开始再驱动需要的脉冲的输出。

16bit	32bit	P	FNC 157	**PLSV**	可变速度脉冲输出
✔	✔				
9 步	17 步		指令格式： **PLSV** Ⓢ ⒹⅠ ⒹⅡ		

该指令是按指定的端口、频率和运行方向输出脉冲频率，没有加减速过程，当驱动能流无效时，输出脉冲直接停止。只有晶体管输出PLC才能使用该指令。其中：

Ⓢ为指定的输出脉冲频率。16bit指令时，范围是1～32,767Hz；－1～－32,768Hz ；32bit指令时，范围是1～200,000Hz；－1～－200,000Hz。其中负号表示反方向运行的指令信号；

ⒹⅠ为脉冲输出端口；MT型只能指定Y0或Y1，MTQ型则可选Y0、Y1、Y2、Y3、Y4等；

ⒹⅡ运行方向输出端口或位变量，输出为ON状态，表示为正向运行；否则为反向运行。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
ⒹⅠ		✔													
ⒹⅡ		✔	✔	✔											

指令举例：

语句表示，当 M1 为 ON 时，由 Y1 端口输出 10kHz 频率的脉冲，Y4 用于控制运行方向，若 Y4=ON 表示为正方向。

本指令执行中，涉及的系统变量有：

1. D8141（高字节），D8140（低字节）：Y000 输出脉冲数。反转时减少。（使用 32 位）
2. D8143（高字节），D8142（低字节）：Y001 输出脉冲数。反转时减少。（使用 32 位）
3. M8145：Y000 脉冲输出停止（立即停止）
4. M8146：Y001 脉冲输出停止（立即停止）
5. M8147 ：Y000 脉冲输出中监控（BUSY/READY）
6. M8148 ：Y001 脉冲输出中监控（BUSY/READY）

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一的高速 Y 输出口进行操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需满足前一次驱动的定位指令所使用的[脉冲输出中监控（Y 000 ： [M81471]，Y001；[M8148]）]OFF 后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有 1 个周期以上的 OFF 时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」，再次扫描到该指令（运算）时，才开始再驱动需要的脉冲的输出。

16bit	32bit	P	FNC 158	**DRVI**	相当位置控制
✔	✔				
9 步	17 步		指令格式： **DRVI** (S1) (S2) (D1) (D2)		

该指令是按指定的端口、频率和运行方向输出指定的脉冲数，令伺服执行机构在当前位置的基础上作给定偏移量的运动。只有晶体管输出PLC才能使用该指令。其中：

(S1)为指定的本次输出脉冲数。16bit指令时，范围是－32768～32,767；32bit指令时，范围是-2,147,483,648～2,147,483,647。其中负号表示反方向；

Ⓢ2为指定的输出脉冲频率，范围为10～200,000Hz；

Ⓓ1为脉冲输出端口；MT型主模块只能指定Y0或Y1，MTQ型主模块则可选Y0、Y1、Y2、Y3、Y4等端口；

Ⓓ2运行方向输出端口或位变量，输出为ON状态，表示为正向运行；否则为反向运行。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
Ⓢ1					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Ⓢ2					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
Ⓓ1		✔													
Ⓓ2		✔	✔	✔											

输出脉冲数，是相对于下面的当前值寄存器作为相对位置：

向［Y000］输出时，当前寄存器为[D8141（高字节），D8140（低字节）]（使用 32 位）

向［Y001］输出时，当前寄存器为[D8143（高字节），D8142（低字节）]（使用 32 位）

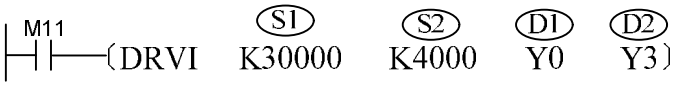
反转时，当前值寄存器的数值减小。

在指令执行过程中，即使改变操作数的内容，也无法在当前运行中表现出来。只在下一次指令执行时才有效。

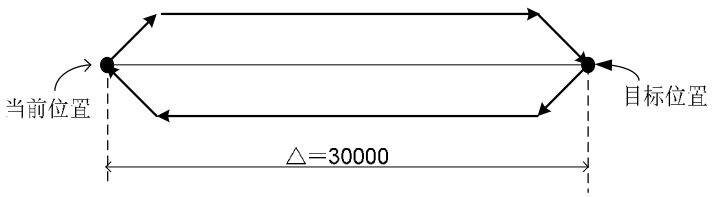
若在指令执行过程中，指令驱动的接点变为 OFF 时，将减速停止。此时执行完成标志 M8029 不会动作；

指令驱动接点变为 OFF 后，在脉冲输出中断标志 M8147（Y000）、M8148（Y001）处于 ON 时，将不接受指令的再次驱动。

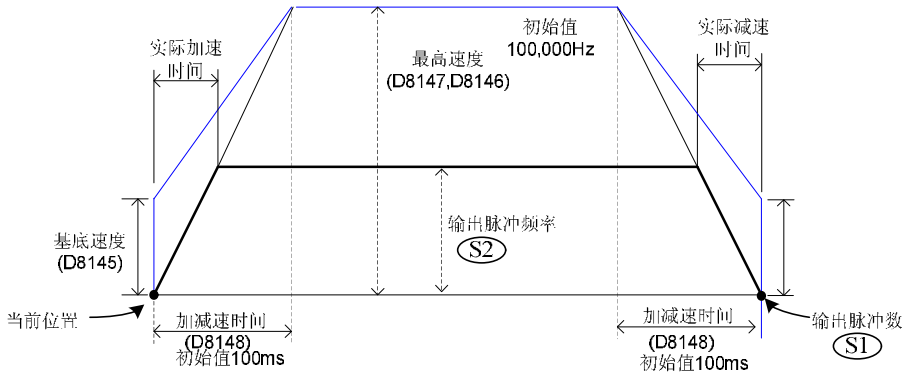
指令举例：



表示以 4kHz 的频率、由 Y0 端口输出 30000 个脉冲，令外部伺服执行机构运行，方向则有 Y3 决定。



脉冲输出过程中，频率会按预定值有加减速：



实际能够输出的输出脉冲频率的最低频率数，根据以下公式所决定：

输出脉冲频率的最低频率数＝$\sqrt{\text{最高速度 [D8147 , D8146] H z} \div (2 \times \text{加减速时间 [D8148]ms} + 1000)}$

即使指定了低于上面计算结果的数值，仍将输出计算值的频率。 加速初期和减速最终部分的频率也不可低于上述计算结果.

指令执行中涉及的系统变量如下：

[D8145]： 执行 FNC 158 (DRVI)，FNCI59 (DRVA ）指令时的基底速度。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（D8I47，D8I46 ）的 I / 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 （高字节） ，D8146 （低字节）]： 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 ~100 ，000 （Hz）

[D8I48]：执行 FNC158 (DRVI) ，FNCI59 (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（D 8147 ，D8146）时，实际加减速时间会缩短。设定范围： 50 ~ 5，000 (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）

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一的高速 Y 输出口进行操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需满足前一次驱动的定位指令所使用的[脉冲输出中监控（Y 000 ： [M81471]，Y001；[M8148]）]OFF 后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有 1 个周期以上的 OFF 时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」，再次扫描到该指令（运算）时，才开始再驱动需要的脉冲的输出。

16bit	32bit	P	FNC 159	**DRVA**	绝对位置控制
✔	✔				
9 步	17 步		指令格式： **DRVA** (S1) (S2) (D1) (D2)		

该指令是按指定的端口、频率和运行方向输出脉冲，令伺服执行机构运动到指定目的点。只有晶体管输出PLC才能使用该指令。其中：

(S1)为指定的目的点绝对位置。16bit 指令时，范围是－32768～32,767；32bit 指令时，范围是-2,147,483,648～2,147,483,647。该设定值将被置入相应的绝对目标位置寄存器：

若(D1)＝［Y000］，写入[D8141（高字节），D8140（低字节）]　（使用 32 位）；

若(D1)＝［Y001］，写入[D8143（高字节），D8142（低字节）]　（使用 32 位）。

其中负号表示反方向。反转时，当前值寄存器的数值减小。

(S2)为指定的输出脉冲频率，范围为10～32,767Hz（16bit指令）；或10～100,000Hz（32bit指令）；

(D1)为脉冲输出端口；MT型只能指定Y0或Y1，MTQ型则可选Y0、Y1、Y2、Y3、Y4等；

(D2)运行方向输出端口或位变量，可根据(S1)和当前位置的差值决定，输出为ON状态，表示为正向运行；否则为反向运行。

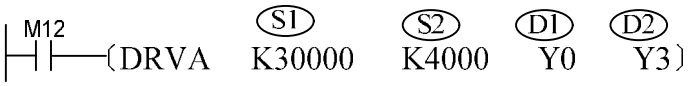
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D1)		✔													
(D2)		✔	✔	✔											

在指令执行过程中，即使改变操作数的内容，也无法在当前运行中表现出来。只在下一次指令执行时才有效。

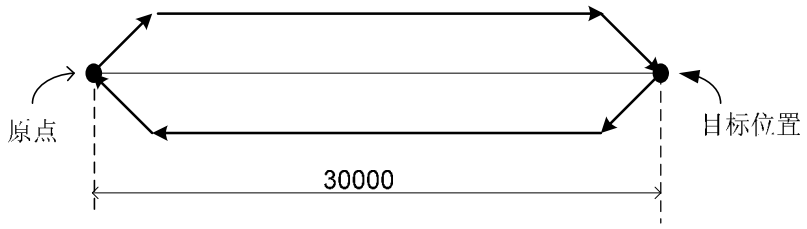
若在指令执行过程中，指令驱动的接点变为 OFF 时，将减速停止。此时执行完成标志 M8029 不会动作；

指令驱动接点变为 OFF 后，在脉冲输出中断标志 M8147（Y000）、M8148（Y001）处于 ON 时，将不接受指令的再次驱动。

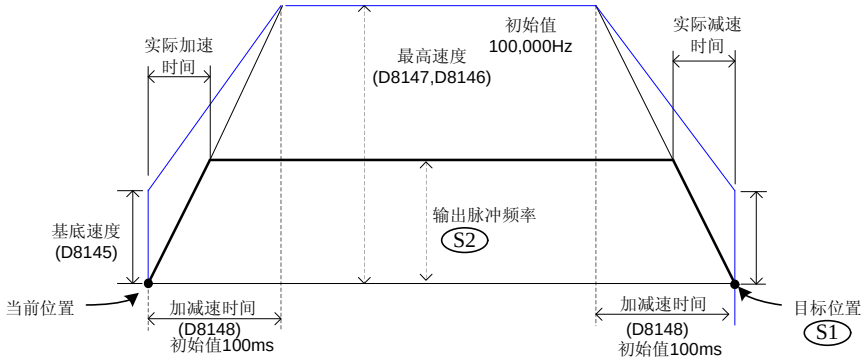
指令举例：



该指令是由指定原点向目标点运行的控制方式，



脉冲输出过程中，频率会按预定值有加减速：



实际能够输出的输出脉冲频率的最低频率数，根据以下公式所决定：

输出脉冲频率的最低频率数＝$\sqrt{\text{最高速度 [D8147 , D8146] H z} \div (2 \times \text{加减速时间 [D8148]ms} + 1000)}$

即使指定了低于上面计算结果的数值，仍将输出计算值的频率。 加速初期和减速最终部分的频率也不可低于上述计算结果.

指令执行中涉及的系统变量如下：

[D8145]： 执行 FNC 158 (DRVI)，FNC159 (DRVA ）指令时的基底速度。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 （高字节） , D8146 （低字节)]: 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 ~100 , 000 （Hz）

[D8148]：执行 FNC158 (DRVI) , FNCI59 (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（D 8147 , D8146）时，实际加减速时间会缩短。设定范围： 50 ~ 5，000 (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）

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一的高速 Y 输出口进行操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需满足前一次驱动的定位指令所使用的[脉冲输出中监控（Y 000 : [M81471]，Y001；[M8148]）]OFF 后，经过一个运算周期后，方能再次驱动。

定位指令再次驱动时，必须有 1 个周期以上的 OFF 时间，若以比上述条件更短的时间内执行再驱动时，将在最初指令执行（运算）时发生「运算错误」，再次扫描到该指令（运算）时，才开始再驱动需要的脉冲的输出。

16bit	32bit	P	FNC 160	**TCMP**	时钟数据比较
✔	✔				
9 步	17 步		指令格式： **TCMP** (S1) (S2) (S3) (S) (D)		

该指令是将指定的时、分、秒数值，与内部实时时钟进行比较，输出比较结果。其中：

(S1) 为指定比较用时间的“时”，范围是 0～23：

(S2) 为指定比较用时间的“时”，范围是0～59；

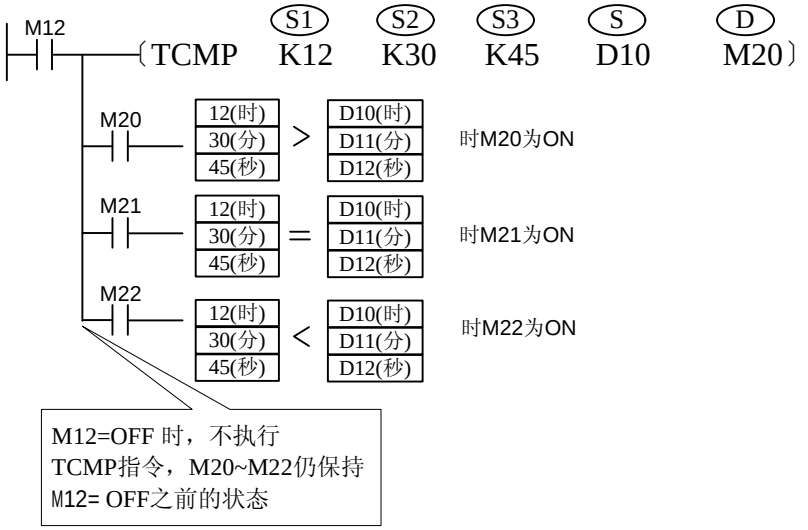
(S3) 为指定比较用时间的“秒”，范围是0～59；

(S) 为实时时钟的时间寄存器的启始地址，通常为时钟读取TRD或MOV指令读取后的存放单元；

(D)为比较结果的存放变量启始地址，占用后续共3个变量单元；

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S3)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S)											✔	✔	✔		
(D)		✔	✔	✔											

指令举例：



16bit	32bit	P	FNC 161	**TZCP**	时钟数据区间比较
✔		✔			
11 步		11 步	指令格式：　**TZCP**　(S1) (S2) (S) (D)		

该指令是将内置实时时钟数据与指定的两组时/分/秒预设值进行区间比较，输出比较结果。

其中：

(S1)为设定的时间比较下限，占用 3 个连续的变量单元，依次存储时、分、秒数据：

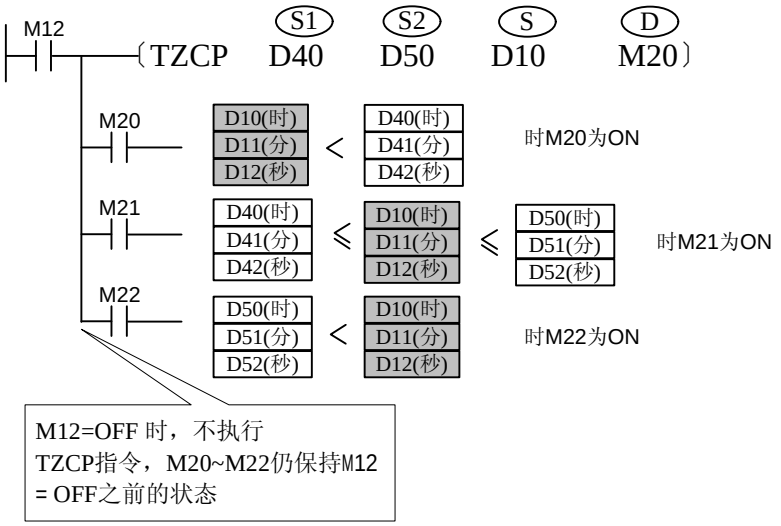
(S2)为设定的时间比较上限，占用3个连续的变量单元，依次存储时、分、秒数据：

(S)为实时时钟的时间寄存器的启始地址，通常为时钟读取TRD或MOV指令读取后的存放单元；

(D)为比较结果的存放变量启始地址，占用后续共3个变量单元；

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)											✔	✔	✔		
(S2)											✔	✔	✔		
(S)											✔	✔	✔		
(D)		✔	✔	✔											

指令举例：



16bit	32bit	P	FNC 162	**TADD**	时钟数据加法运算
✔		✔			
7 步		7 步	指令格式： **TADD** (S1) (S2) (D)		

该指令是将2组时钟数据的时/分/秒对应相加，结果保持于指定的变量中。其中：

(S1) 为时间被加数，占用 3 个连续的变量单元，依次存储时、分、秒数据：

(S2) 为时间加数，占用3个连续的变量单元，依次存储时、分、秒数据：

(D) 为时间相加和，存储单元，占用3个连续的变量单元，依次存储时、分、秒数据：

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)											✔	✔	✔		
(S2)											✔	✔	✔		
(D)											✔	✔	✔		

若计算结果超过 24 小时，进位标志 M8022 置 1，实际显示的时间会减去 24:00:00 的数值；

若计算结果为 00:00:00，零标志 M8020 置 1。

指令举例：

完成的操作如下：

16bit	32bit	P	FNC 163	**TSUB**	时钟数据减法运算
✔		✔			
7 步		7 步	指令格式： **TSUB** (S1) (S2) (D)		

该指令是将2组时钟数据的时/分/秒对应相减，结果保持于指定的变量中。其中：

(S1) 为时间被减数，占用 3 个连续的变量单元，依次存储时、分、秒数据：

(S2) 为时间减数，占用3个连续的变量单元，依次存储时、分、秒数据：

(D) 为时间相减差，存储单元，占用3个连续的变量单元，依次存储时、分、秒数据：

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)											✔	✔	✔		
(S2)											✔	✔	✔		
(D)											✔	✔	✔		

若计算结果 00:00:00，零标志 M8020 置 1；

若计算结果为负，借位标志 M8021 置 1，实际显示的时间会加上 24:00:00 的数值；

指令举例：

完成的操作如下：

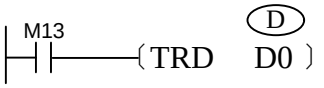
16bit	32bit	P	FNC 166	**TRD**	时钟数据读取
✔		✔			
3 步		3 步	指令格式： **TRD** (D)		

该指令是读取内置的实时时钟的年/月/日/时/分/秒/星期，将该7个数据保持于指定的寄存器中。

操作数	位元件				字　元　件									
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(D)											✔	✔	✔		

其中：(D)为保存读取时间的启始存储单元，占用共7个连续的变量单元，地址由小到大依次存储：星期、秒、分、时、日、月、年等数据。

指令举例：



操作如下：

项目	系统变量		操作后(D)
年(0~99)	D8018	→	D0
月(1~12)	D8017	→	D1
日(1~31)	D8016	→	D2
时(0~23)	D8015	→	D3
分(0~59)	D8014	→	D4
秒(0~59)	D8013	→	D5
星期[0(日)~6]	D8012	→	D6

16bit	32bit	P	FNC 167	**TWR**	时钟数据写入
✔		✔			
3 步		3 步	指令格式： **TWR** (S)		

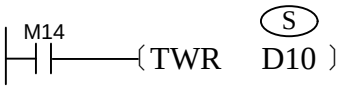
该指令是将指定时钟数据(S)（含年/月/日/时/分/秒/星期）的7个数据写入内置的实时时钟。

操作数	位元件				字　元　件									
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)											✔	✔	✔		

其中：(S)为保存读取时间的启始存储单元，占用共 7 个连续的变量单元，地址由小到大

依次存储：星期、秒、分、时、日、月、年等数据

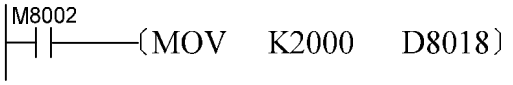
指令举例：



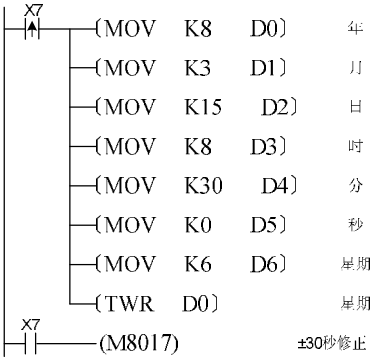
操作如下：

项目	数据源 (D)		系统变量
年(0~99)	D0	→	D8018
月(1~12)	D1	→	D8017
日(1~31)	D2	→	D8016
时(0~23)	D3	→	D8015
分(0~59)	D4	→	D8014
秒(0~59)	D5	→	D8013
星期[0(日)~6]	D6	→	D8012

若需要将“年”采用 4 位的格式，可执行如下语句，切换成 4 位格式：



若原来 D8018＝08，切换后 D8018＝2008。校时方法如下：



16bit	32bit	P	FNC 169	**HOUR**	计时器
✔	✔				
7 步	7 步		指令格式： **HOUR** (S) (D1) (D2)		

该指令是记录驱动条件满足的累加时间，当达到设定的时间后，令指定输出有效。其中：

(S)为设定时间，单位为“小时”，当累加时间达到该设定值后，令输出有效；

(D1)累计时间启始单元，16bit指令，共占用2个单元；32bit指令，共占3个单元。对于需要掉电保持的计时用途，可选择掉电保持的D变量单元；

(D2)计时到达告警输出变量单元，当计时到达设定值后，该指定单元状态有效。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D1)													✔		
(D2)		✔	✔	✔											

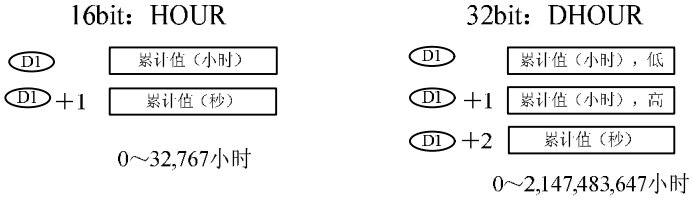
程序举例：

当 M200＝ON 时，累计该状态的持续时间，记录在 D3010 中，当累计时间达到 2000 小时后，Y10 输出状态为 ON。

到达(S)指定数值后，累计计时仍继续进行，计时条件满足时，读数会继续增大；

若(D1)指定为非掉电保持的单元，PLC 每次由 OFF→ON 时读数会清 0；

指令占用的(D1)变量单元及计时范围如下图，计时没有负数：



16bit	32bit	P	FNC 170	**GRY**	格雷码转换（BIN→GRY）
✔	✔	✔			
5 步	9 步	9 步	指令格式：　**GRY**　(S)　(D)		

该指令是将BIN数值，转换为格雷码（GRY）。其中：

(S)为待转换的BIN的数据源或数据变量单元；16bit指令时范围0～32,767，32bit指令时范围0～2,147,483,647；

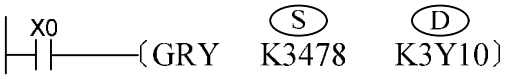
(D)为转换为格雷码（GRY）后的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

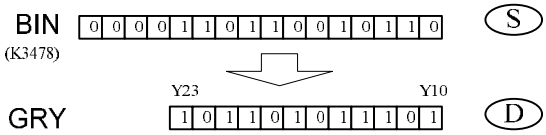
(D)								✔	✔	✔	✔	✔	✔	✔	✔

BIN→GRY 数学算法：从最右边一位起，依次将每一位与左边一位异或(XOR)，作为对应 GRY 该位的值，最左边一位不变(相当于左边是 0)；

程序举例：



执行结果：



16bit	32bit	P	FNC 171	**GBIN**	格雷码逆转换（GRY→BIN）
✔	✔	✔			
5 步	9 步	9 步	指令格式： **GBIN** (S) (D)		

该指令是将GRY格雷码，转换为二进制数值。其中：

(S)为待转换的GRY码数据源或数据变量单元；

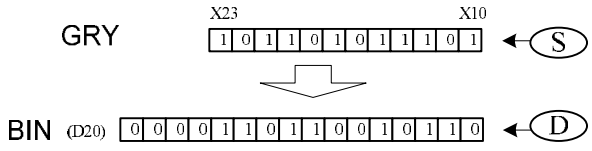
(D)为转换为BIN后的存放单元。

操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

GRY→BIN 数学算法：从左边第二位起，将每位与左边一位解码后的值异或，作为该位解码后的值（最左边一位依然不变）。

程序举例：

执行结果举例：



FNC NO.	指令助记符	《指令名称》
224	LD =	触点比较指令运算开始（S1）=（S2）时导通
225	LD>	触点比较指令运算开始（S1）>（S2）时导通
226	LD <	触点比较指令运算开始（S1）<（S2）时导通
228	LD< >	触点比较指令运算开始（S1）≠（S2）时导通
229	LD≤	触点比较指令运算开始（S1）≤（S2）时导通
230	LD≥	触点比较指令运算开始（S1）≥（S2）时导通
232	AND =	触点比较指令串联连接（S1）=（S2）时导通
233	AND >	触点比较指令串联连接（S1）>（S2）时导通
234	ANO<	触点比较指令串联连接（S1）<（S2）时导通
236	AND < >	触点比较指令串联连接（S1）≠（S2）时导通
237	AND≤	触点比较指令串联连接（S1）≤（S2）时导通
238	AND≥	触点比较指令串联连接（S1）≥（S2）时导通
240	OR =	触点比较指令并联连接（S1）=（S2）时导通
241	OR >	触点比较指令并联连接（S1）>（S2）时导通
242	OR<	触点比较指令并联连接（S1）<（S2）时导通
244	OR< >	触点比较指令并联连接（S1）≠（S2）时导通
245	OR ≥	触点比较指令并联连接（S1）≥（S2）时导通
246	OR≤	触点比较指令并联连接（S1）≤（S2）时导通

16bit	32bit	P	FNC	**LD** ☼	触点型比较指令
✔	✔		224～230		
5 步	9 步		指令格式： **LD** (S1) (S2)		

该指令将两个操作数进行比较，将比较结果以逻辑状态输出，参与比较的变量都按有符号数处理，☼比较符中有=、>、<、>=、<=、<>等。其中：

(S1)为待比较的数据源或数据变量单元1；

(S2)为待比较的数据源或数据变量单元2。

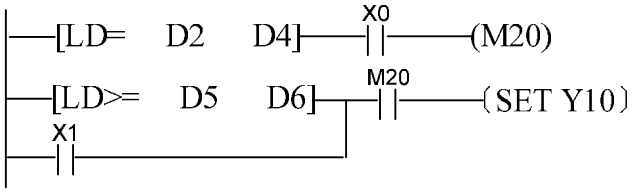
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

LD触点型比较方式有：

16bit 指令	FNC NO	32bit 指令	导通条件	非导通条件
LD=	224	LDD=	(S1) = (S2)	(S1) ≠ (S2)

LD>	225	LDD>	(S1) > (S2)	(S1) <= (S2)
LD<	226	LDD<	(S1) < (S2)	(S1) >= (S2)
LD<>	228	LDD<>	(S1) <> (S2)	(S1) = (S2)
LD<=	229	LDD<=	(S1) <= (S2)	(S1) > (S2)
LD>=	230	LDD>=	(S1) >= (S2)	(S1) < (S2)

程序举例：



对于参与比较的数为 32bit 宽度的计数器，应使用 32bit 指令 LDD*，否则出错。

16bit	32bit	P	FNC	AND ☼	接点型比较指令
✔	✔		232～238		
5 步	9 步		指令格式： **AND** (S1) (S2)		

该指令之前已有其他逻辑操作。该指令将两个操作数进行比较，将比较结果以逻辑状态形式参与程序能流的运算，指令中参与比较的变量都按有符号数处理，☼比较符中有=、>、<、>=、<=、<>等。其中：

(S1)为待比较的数据源或数据变量单元1；

(S2)为待比较的数据源或数据变量单元2。

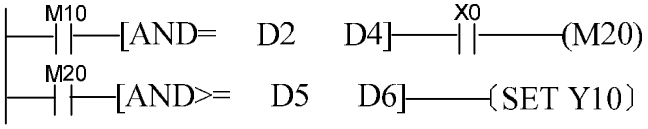
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

LD触点型比较方式有：

16bit 指令	FNC NO	32bit 指令	导通条件	非导通条件
AND=	224	ANDD=	(S1) = (S2)	(S1) ≠ (S2)

AND >	225	ANDD>	(S1) > (S2)	(S1) <= (S2)
AND <	226	ANDD<	(S1) < (S2)	(S1) >= (S2)
AND <>	228	ANDD<>	(S1) <> (S2)	(S1) = (S2)
AND <=	229	ANDD<=	(S1) <= (S2)	(S1) > (S2)
AND >=	230	ANDD>=	(S1) >= (S2)	(S1) < (S2)

程序举例：



对于参与比较的数为 32bit 宽度的计数器，应使用 32bit 指令 ANDD*，否则出错。

16bit	32bit	P	FNC	OR ☼	并联接点型比较指令
✔	✔		240～246		
5 步	9 步		指令格式： **OR** (S1) (S2)		

该指令将两个操作数进行比较，将比较结果以逻辑状态形式参与程序能流的或运算，指令中参与比较的变量都按有符号数处理，☼比较符中有=、>、<、>=、<=、<>等。其中：

(S1) 为待比较的数据源或数据变量单元1；

(S2) 为待比较的数据源或数据变量单元2。

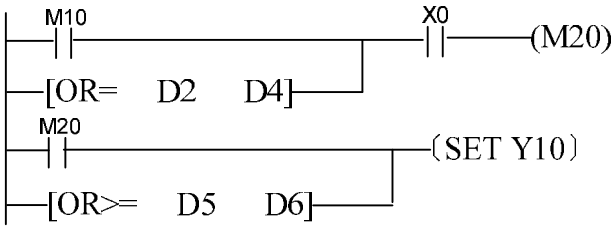
操作数	位元件				字　元　件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

LD触点型比较方式有：

16bit 指令	FNC NO	32bit 指令	导通条件	非导通条件
OR=	224	OR D=	(S1) = (S2)	(S1) ≠ (S2)
OR >	225	OR D>	(S1) > (S2)	(S1) <= (S2)
OR <	226	OR D<	(S1) < (S2)	(S1) >= (S2)

OR <>	228	OR D<>	(S1) <> (S2)	(S1) = (S2)
OR <=	229	OR D<=	(S1) <= (S2)	(S1) > (S2)
OR >=	230	OR D>=	(S1) >= (S2)	(S1) < (S2)

程序举例：



对于参与比较的数为 32bit 宽度的计数器，应使用 32bit 指令 ORD*，否则出错。

8. 附录

8.1 系统特殊软元件：

M8000～M8255，D8000～D8255 被定义为特殊元件种类，及其功能如下表所述。

M元件	M元件的描述	D元件	D元件的描述
系统运行状态			
M8000	用户程序运行时置为ON状态	D8000	用户程序运行的监视定时器
M8001	M8000状态取反	D8001	单板程序版本，如24100 H2U=24,100版本V1.00
M8002	用户程序开始运行的第一个周期为ON	D8002	程序容量，4K,8K,16K等
M8003	M8002状态取反	D8003	固定为0X10，为可编程控制器内存储
M8004	当M8060~M8067[除M8062]中任意一个处于ON，则M8004有效	D8004	错误的M8060~M8067的BCD值，正常为0。
M8005	电池电压过低时动作	D8005	电池电压的BCD当前值
M8006	电池电压低有出现就动作[锁存]	D8006	电池电压过低的检测值，初始值为2.6V
M8007	当交流失电5mS后M8007&M8008动作，但在D8008之内程序继续运行	D8007	保存M8007动作的次数，当失电时该单元清0处理
M8008	当D8008时间内都失电,当M8008由ON→OFF时，用户程序不运行。M8000为OFF	D8008	交流失电检测时间，默认为10mS
M8009	扩展单元24V掉电时动作	D8009	扩展单元24V失电的模块号
系统时钟			
M8010	保留	D8010	当前扫描时间,从用户程序0步开始(0.1mS)
M8011	10mS时钟周期的振荡时钟	D8011	扫描时间的最小值(0.1mS)
M8012	100mS时钟周期的振荡时钟	D8012	扫描时间的最大值(0.1mS)
M8013	1S时钟周期的振荡时钟	D8013	时钟秒(0~59)
M8014	1分钟时钟周期的振荡时钟	D8014	实时时钟分(0~59)
M8015	时钟停止和预置	D8015	实时时钟小时(0~23)
M8016	时钟读取显示停止	D8016	实时时钟日(1~31)
M8017	±30秒修正	D8017	实时时钟月(1~12)
M8018	安装检测	D8018	实时时钟公历年(2000~2099)
M8019	实时时钟(RTC)出错	D8019	实时时钟星期
指令标志			
M8020	运算零标志	D8020	X000~X007的输入滤波常数0~60[默认10mS]
M8021	运算借位标志	D8021	保留
M8022	运算进位标志	D8022	保留
M8023	保留	D8023	保留
M8024	BMOV指令的方向	D8024	保留
M8025	HSC指令模式	D8025	保留
M8026	RAMP指令模式	D8026	保留
M8027	PR模式	D8027	保留
M8028	保留	D8028	和Z0同一变量地址
M8029	部分指令（PLSR等）指令执行完成	D8029	和V0同一变量地址
系统模式			
M8030	为ON时，屏蔽电池低告警	D8030	保留
M8031	为ON时，清除所有非保存存储器	D8031	保留

M元件	M元件的描述	D元件	D元件的描述
M8032	为ON时，清除所有保存存储器	D8032	保留
M8033	为ON时，停机状态所有的软元件不变	D8033	保留
M8034	为ON时，PLC所有的输出都为OFF状态	D8034	保留
M8035	强制运行命令1	D8035	保留
M8036	强制运行命令2	D8036	保留
M8037	强制停止命令	D8037	保留
M8038	通讯参数设定标志	D8038	保留
M8039	恒定扫描控制	D8039	恒定扫描时间，默认0，以mS为单位
步进阶梯			
M8040	转移禁止	D8040	将S0~S899的最小动作地址号保存在D8040中，其它依次，最大的地址号保存在D8047中。
M8041	转移开始	D8041	
M8042	对应启动输入的脉冲输出	D8042	
M8043	原点回归状态的结束标志	D8043	
M8044	检测到机械原点动作	D8044	
M8045	所有输出复位禁止	D8045	
M8046	M8047动作后，当S0~S999中任何一个为ON，M8046为ON	D8046	
M8047	STL监视有效[D8040~D8047有效]	D8047	
M8048	M8049=ON，S900~S999任何一个有效，M8048有效	D8048	保留
M8049	信号报警有效，[D8049有效]	D8049	保存S900~S999的报警最小地址号
中断禁止		保留	
M8050	驱动 I00□中断禁止	D8050	保留
M8051	驱动 I10□中断禁止	D8051	保留
M8052	驱动 I20□中断禁止	D8052	保留
M8053	驱动 I30□中断禁止	D8053	保留
M8054	驱动 I40□中断禁止	D8054	保留
M8055	驱动 I50□中断禁止	D8055	保留
M8056	驱动 I60□中断禁止	D8056	保留
M8057	驱动 I70□中断禁止	D8057	保留
M8058	驱动 I80□中断禁止	D8058	保留
M8059	驱动 计数器中断禁止	D8059	保留
系统错误检测			
元件	名称	错误灯	运行		
M8060	I/O构成错误[]	OFF	RUN	D8060	I/O构成错误的未安装I/O起始地址号
M8061	PLC硬件错误	闪烁	STOP	D8061	PLC硬件错误的错误代码序号
M8062	PLC通讯错误	OFF	RUN	D8062	PLC通讯错误代码
M8063	联机错误/一般通讯错误	OFF	RUN	D8063	并行联机错误代码
M8064	参数错误	闪烁	STOP	D8064	参数错误代码
M8065	语法错误	闪烁	STOP	D8065	语法错误的代码
M8066	回路错误	闪烁	STOP	D8066	回路错误的代码
M8067	运算错误	OFF	RUN	D8067	运算错误的代码
M8068	运算错误锁存	OFF	RUN	D8068	锁存运算错误程序的步号
M8069	保留			D8069	M8065~M8067的错误发生的步号
联机功能					
M8070	联机主站驱动			D8070	并联联机错误时间 500mS

M元件	M元件的描述	D元件	D元件的描述
M8071	联机从站驱动	D8071	保留
M8072	并联连接运行中为ON	D8072	保留
M8073	并联连接M8070/M8071设定不良	D8073	保留
采样跟踪			
M8074	保留	D8074	采样剩余次数
M8075	取样跟踪准备开始指令	D8075	采样次数的设定(1~512)
M8076	取样跟踪准备完成，执行开始指令	D8076	采样周期
M8077	取样跟踪 执行中监控	D8077	触发指定
M8078	取样跟踪 执行完成监控	D8078	触发条件元件地址号设定
M8079	取样跟踪超过D8075的数据	D8079	采样数据指针
M8080	保留	D8080	位元件地址号No.0
M8081	保留	D8081	位元件地址号No.1
M8082	保留	D8082	保留
M8083	保留	D8083	保留
M8084	保留	D8084	保留
M8085	保留	D8085	保留
M8086	保留	D8086	保留
M8087	保留	D8087	保留
M8088	保留	D8088	保留
M8089	保留	D8089	保留
M8090	保留	D8090	保留
M8091	保留	D8091	保留
M8092	保留	D8092	保留
M8093	保留	D8093	保留
M8094	保留	D8094	保留
M8095	保留	D8095	保留
M8096	保留	D8096	字元件地址号No.0
M8097	保留	D8097	字元件地址号No.1
M8098	保留	D8098	字元件地址号No.2
高速环形计数器			
M8099	高速环形计数器计数启动	D8099	[0~32767]上升动作环形计数器(0.1mS)
其它功能使用			
M8100	保留	D8100	保留
M8101	保留	D8101	单板程序版本，如24100 H2U=24,100版本V1.00
M8102	保留	D8102	系统提供给用户程序的程序容量
M8103	保留	D8103	保留
M8104	保留	D8104	保留
M8105	保留	D8105	保留
M8106	保留	D8106	保留
M8107	保留	D8107	保留
M8108	保留	D8108	保留
M8109	输出刷新错误	D8109	输出刷新错误的输出地址编号

COM0 通讯.链接			
M8110	保留	D8110	通讯格式，界面配置设定，默认为0
M8111	发送等待中（RS指令）	D8111	站号设置，界面配置设定，默认为1
M8112	发送标志（RS指令） 指令执行状态（MODBUS）	D8112	传送剩余数据数量（仅对RS指令）
M8113	接收完成标志（RS） 通讯错误标志（MODBUS）	D8113	接收到的数据数量（仅对RS指令）
M8114	接收中（仅对RS指令）	D8114	起始字符STX（仅对RS指令）
M8115	保留	D8115	终止字符ETX（仅对RS指令）
M8116	保留	D8116	通讯协议设定，界面配置设定，默认为0
M8117	保留	D8117	计算机链接协议接通要求数据起始地址号
M8118	保留	D8118	计算机链接协议接通要求发送数据数量
M8119	超时判断	D8119	通讯超时时间判断，界面配置设定，默认为10（100ms）
COM1 通讯.链接			
M8120	保留	D8120	通讯格式，界面配置设定，默认为0
M8121	发送等待中（RS指令）	D8121	站号设置，界面配置设定，默认为1
M8122	发送标志（RS指令） 指令执行状态（MODBUS）	D8122	传送剩余数据数量（仅对RS指令）
M8123	接收完成标志（RS） 通讯错误标志（MODBUS）	D8123	接收到的数据数量（仅对RS指令）
M8124	接收中（仅对RS指令）	D8124	起始字符STX（仅对RS指令）
M8125	保留	D8125	终止字符ETX（仅对RS指令）
M8126	保留	D8126	通讯协议设定，界面配置设定，默认为0
M8127	保留	D8127	计算机链接协议接通要求数据起始地址号
M8128	保留	D8128	计算机链接协议接通要求发送数据数量
M8129	超时判断	D8129	通讯超时时间判断，界面配置设定，默认为10（100ms）
高速&定位			
M8130	HSZ指令平台的控制模式	D8130	HSZ高速比较平台使用(记录号)
M8131	和M8130联合使用	D8131	HSZ&PLSY速度模型使用(记录号)
M8132	HSZ&PLSY速度模式	D8132	HSZ&PLSY速度模型频率使用
M8133	和M8132联合使用	D8133	
M8134	保留	D8134	HSZ&PLSY速度模型比较脉冲数使用
M8135	Y0减速时间和脉冲输出量可变有效[ON]	D8135	
M8136	Y1减速时间和脉冲输出量可变有效[ON]	D8136	Y000&Y001输出脉冲合计数
M8137	Y2减速时间和脉冲输出量可变有效[ON]	D8137	
M8138	Y3减速时间和脉冲输出量可变有效[ON]	D8138	保留
M8139	Y4减速时间和脉冲输出量可变有效[ON]	D8139	保留
M8140	ZRN的CLR信号输出功能有效	D8140	PLSY&PLSR输出Y000对应的脉冲个数累积值
M8141	保留	D8141	
M8142	保留	D8142	PLSY&PLSR输出Y001对应的脉冲个数累积值
M8143	保留	D8143	
M8144	保留	D8144	
M8145	Y000脉冲输出停止	D8145	DRVI,DRVA执行时的偏置速度
M8146	Y001脉冲输出停止	D8146	DRVI,DRVA执行时的最高速度[默认100,000]
M8147	Y000脉冲输出监控	D8147	

M8148	Y001脉冲输出监控	D8148	DRVI,DRVA执行时加减速时间[默认100]
M8149	Y002脉冲输出监控	D8149	保留
M8150	Y003脉冲输出监控	D8150	PLSY&PLSR输出Y002对应的脉冲个数累积值
M8151	Y004脉冲输出监控	D8151	
M8152	Y002脉冲输出停止	D8152	PLSY&PLSR输出Y003对应的脉冲个数累积值
M8153	Y003脉冲输出停止	D8153	
M8154	Y004脉冲输出停止	D8154	PLSY&PLSR输出Y004对应的脉冲个数累积值
M8155	保留	D8155	
M8156	保留	D8156	Y0端口清零信号定义(ZRN)[默认5=Y005]
M8157	保留	D8157	Y1端口清零信号定义(ZRN)[默认6=Y006]
扩展功能			
M8158	保留	D8158	Y2端口清零信号定义(ZRN)[默认7=Y007]
M8159	保留	D8159	Y3端口清零信号定义(ZRN)[默认8=Y010]
M8160	(XCH)的SWAP功能	D8160	Y4端口清零信号定义(ZRN)[默认9=Y011]
M8161	ASC/RS/ASCII/HEX/CCD的位处理模式	D8161	保留
M8162	高速并联连接模式	D8162	保留
M8163	保留	D8163	保留
M8164	(FROM/TO)传送点数可变模式	D8164	(FROM/TO)传送点数指定模式
M8165	保留	D8165	PLSR,DRVI,DRVA执行时减速时间[默认100]由M8135决定是否有效[Y0]
M8166	保留	D8166	PLSR,DRVI,DRVA执行时减速时间[默认100]由M8136决定是否有效[Y1]
M8167	(HEY)HEX数据处理功能	D8167	PLSR,DRVI,DRVA执行时减速时间[默认100]由M8137决定是否有效[Y2]
M8168	(SMOV)HEX数据处理功能	D8168	PLSR,DRVI,DRVA执行时减速时间[默认100]由M8138决定是否有效[Y3]
M8169	保留	D8169	PLSR,DRVI,DRVA执行时减速时间[默认100]由M8139决定是否有效[Y4]
脉冲捕捉		通讯.链接	
M8170	X000脉冲捕捉，上升沿置位，R/W	D8170	保留
M8171	X001脉冲捕捉，上升沿置位，R/W	D8171	保留
M8172	X002脉冲捕捉，上升沿置位，R/W	D8172	保留
M8173	X003脉冲捕捉，上升沿置位，R/W	D8173	本站站号设定状态
M8174	X004脉冲捕捉，上升沿置位，R/W	D8174	通讯子站设定状态
M8175	X005脉冲捕捉，上升沿置位，R/W	D8175	刷新范围设定状态
M8176	保留	D8176	本站站号设定
M8177	保留	D8177	通讯子站数设定
M8178	保留	D8178	刷新范围设定
M8179	保留	D8179	重试次数设定
M8180	保留	D8180	通信超时设置

通讯.链接		变址寻址	
M8181	保留	D8181	保留
M8182	保留	D8182	位元件地址号No.2/Z1寄存器内容
M8183	数据传送主站出错	D8183	位元件地址号No.3/V1寄存器内容
M8184	数据传送从站1出错	D8184	位元件地址号No.4/Z2寄存器内容
M8185	数据传送从站2出错	D8185	位元件地址号No.5/V2寄存器内容
M8186	数据传送从站3出错	D8186	位元件地址号No.6/Z3寄存器内容
M8187	数据传送从站4出错	D8187	位元件地址号No.7/V3寄存器内容
M8188	数据传送从站5出错	D8188	位元件地址号No.8/Z4寄存器内容
M8189	数据传送从站6出错	D8189	位元件地址号No.9/V4寄存器内容
M8190	数据传送从站7出错	D8190	位元件地址号No.10/Z5寄存器内容
M8191	数据传送进行中	D8191	位元件地址号No.11/V5寄存器内容
M8192	保留	D8192	位元件地址号No.12/Z6寄存器内容
M8193	保留	D8193	位元件地址号No.13/V6寄存器内容
M8194	保留	D8194	位元件地址号No.14/Z7寄存器内容
M8195	C251倍频控制	D8195	位元件地址号No.15/V7寄存器内容
M8196	C252倍频控制	D8196	保留
M8197	C253倍频控制	D8197	保留
M8198	C254倍频控制	D8198	保留
M8199	C255倍频控制	D8199	保留
	计数器增/减控制或状态	通讯.链接	
M8200	C200控制	D8200	保留
M8201	C201控制	D8201	当前连接扫描时间
M8202	C202控制	D8202	最大连接时间
M8203	C203控制	D8203	主站通讯错误次数
M8204	C204控制	D8204	从站1通讯错误次数
M8205	C205控制	D8205	从站2通讯错误次数
M8206	C206控制	D8206	从站3通讯错误次数
M8207	C207控制	D8207	从站4通讯错误次数
M8208	C208控制	D8208	从站5通讯错误次数
M8209	C209控制	D8209	从站6通讯错误次数
M8210	C210控制	D8210	从站7通讯错误次数
M8211	C211控制	D8211	主站通讯错误代码
M8212	C212控制	D8212	从站1通讯错误代码
M8213	C213控制	D8213	从站2通讯错误代码
M8214	C214控制	D8214	从站3通讯错误代码
M8215	C215控制	D8215	从站4通讯错误代码
M8216	C216控制	D8216	从站5通讯错误代码
M8217	C217控制	D8217	从站6通讯错误代码
M8218	C218控制	D8218	从站7通讯错误代码
M8219	C219控制	D8219	保留
M8220	C220控制	D8220	保留
M8221	C221控制	D8221	保留
M8222	C222控制	D8222	保留
M8223	C223控制	D8223	保留
M8224	C224控制	D8224	保留

M8225	C225控制	D8225	保留
M8226	C226控制	D8226	保留
M8227	C227控制	D8227	保留
M8228	C228控制	D8228	保留
M8229	C229控制	D8229	保留
M8230	C230控制	D8230	保留
M8231	C231控制	D8231	保留
M8232	C232控制	D8232	保留
M8233	C233控制	D8233	保留
M8234	C234控制	D8234	保留
M8235	C235控制	D8235	保留
M8236	C236控制	D8236	保留
M8237	C237控制	D8237	保留
M8238	C238控制	D8238	保留
M8239	C239控制	D8239	保留
M8240	C240控制	D8240	保留
M8241	C241控制	D8241	保留
M8242	C242控制	D8242	保留
M8243	C243控制	D8243	保留
M8244	C244控制	D8244	保留
M8245	C245控制	D8245	保留
M8246	C246状态	D8246	保留
M8247	C247状态	D8247	保留
M8248	C248状态	D8248	保留
M8249	C249状态	D8249	保留
M8250	C250状态	D8250	保留
M8251	C251状态	D8251	保留
M8252	C252状态	D8252	保留
M8253	C253状态	D8253	保留
M8254	C254状态	D8254	保留
M8255	C255状态	D8255	保留

8.2 出错信息说明

特殊数据寄存器 D8060～D8067，存储的错误代码和内容如下表所示。

类型	出错代码	出错内容	处理方法
I/O 结构出错 M8060(D8060)继续运行	例 1020	没有装 I/O 起始元件号“1020”时：1=输 X (0＝输出 Y)，020=元件号	还没有装的输入继电器，输出继电器的编号被编入程序。可编程控制器可以继续运行，若是程序员，请进行修改。
PC 硬件出错 M8061(D8061) 停止运行	0000	无异常	检查扩展电线的连接是否正确。
	6101	RAM 出错	
	6102	运算电路出错	
	6103	I/O 总线出错(M8069 驱动时)	
	6104	扩展设备 24V 以下(M8069 ON 时)	
	6105	监视定时器出错	运算时间超过 D8000 的值，检查程序。
PC/PP 通信出错 M8062(D8062) 继续运行	0000	无异常	程序面板(PP)或程序连口连接的设备与可编程控制器(PC)间的连接是否正确。
	6201	奇偶出错 超过出错 成帧出错	
	6202	通信字符有误	
	6202	通信数据的求和不一致	
	6203	数据格式有误	
	6204	指令有误	
并行连接 通信出错 M8063(D8063) 继续运行	0000	无异常	检查双方的可编程控制器的电源是否为 ON，适配器和控制器之间，以及适配器之间连接是否正确。
	6301	奇偶出错 超过出错 成帧出错	
	6302	通信字符有误	
	6303	通信数据的和数不一致	
	6304	数据格式有误	
	6305	指令有误	
	6306	监视定时器溢出	
	6307~ 6311	无	
	6312	并行连接字符出错	
	6313	并行连接和数出错	
	6314	并行连接格式出错	
	6330	MODBUS 从站地址设置错误	COM0 通讯出错 请检查 COM0 的通讯电缆是否正确连接；检查通讯双方通讯格式是否匹配；检查通讯协议是否匹配；检查是否开机，COM0 只能在开机状态才可能做自由口，若关机只能做监控或下载口；检查 JP0 跳线是否插入，COM0 只能在跳线断开时作为 RS485 自由口，JP0 若接通，COM0 只能做监控或下载口，且是 RS422 模式
	6331	数据帧长度错误	
	6332	地址错误	
	6333	CRC 检验错误	
	6334	不支持的命令码	
	6335	接收超时	
	6336	数据错误	
	6337	缓冲区溢出	
	6338	帧错误	
	6340	MODBUS 从站地址设置错误	COM1 通讯出错，请检查 COM1 的通讯电缆是否正确连接； 检查通讯双方通讯格式是否匹
	6341	数据帧长度错误	
	6342	地址错误	

类型	出错代码	出错内容	处理方法
	6343	CRC 检验错误	配；
	6344	不支持的命令码	
	6345	接收超时	
	6346	数据错误	
	6347	缓冲区溢出	
	6348	帧错误	
参数出错 M8064(D8064) 停止运行	0000	无异常	停止可编程控制器的运行，用参数方式设定正确值。
	6401	程序的求和不一致	
	6402	存储的容量设定有误	
	6403	保存区域设定有误	
	6404	指令区的设定有误	
	6405	文件寄存器的区设定有误	
语法出错 M8065(D8065) 停止运行	0000	无异常	检查编程时对各个指令的使用是否正确？产生错误时请用程序模式进行修改。
	6503	①OUT T，OUT C 之后无设定值 ②应用指令操作数数量不足	
	6504	①标号重复	
	6505	元件号范围溢出	
	6506	使用了未定义指令	
	6507	卷标编号(P)定义出错	
	6508	中断输入(I)的定义出错	
	6511	中断输入和高速计数器输入重复	
电路出错 M8066(D8066) 停止运行	0000	无异常	对整个电路而言，当指令组合不对时，对指令关系有错时都能产生错误。在程序中要修改指令的相互关系，使之正确无误。
	6605	①MPS 的连续使用次数在 9 次以上 ②在 STL 内有 MC，MCR,I(中断)，SRET ③在 STL 外有 RET，没有 RET	
	6606	①没有 P(指针)，I(中断) ②没有 SRET，IRET ③I(中断)，SRET，IRET 在主程序中。 ④STC，RET，MC，MCR 在子程序和中断子程序中	
	6607	①FOR 和 NEXT 关系有错误，嵌套在 6 次以上 ②在 FOR-NEXT 之间有 STL，RET，MC，MCR，IRET，SRET，FEND，END	
	6608	①MC 和 MCR 的关系有错误 ②MCR 没有 NO ③MC-MCR 间有 SRET，IRET，I(中断)	
	6618	只能在主程序中使用的的指令却在主程序之外(中断，子程序等)。	
	6619	FOR-NEXT 之间使用了不能用的指令。 STL,RET,MC,MCR,I，IRET	
	6620	FOR～NEXT 间嵌套溢出	
	6621	FOR～NEXT 数的关系有错误	
	6622	没有 NEXT 指令	
	6623	没有 MC 指令	
	6624	没有 MCR 指令	
	6625	STL 的连续使用次数在 9 次以上	
	6626	在 STL～RET 之间有不能用的指令	

类型	出错代码	出错内容	处理方法	
		MC,MCR,I,SRET,IRET		
	6627	没有 RET 指令		
	6628	在主程序中有不能用的指令 I,SRET,IRET		
	6629	无 P,I		
	6630	没有 SRET,IRET 指令		
	6631	SRET 位于不能用的场所		
	6632	FEND 位于不能用的场所		
	6635	高速输入和高速输出使用硬件端口超过限制		
运算出错 M8067 (D8067) 继续运行	0000	没有异常		
	6701	①CJ,CALL 没有跳转地址 ②在 END 指令后面有卷标 ③在 FOR～NEXT 间或子程序之间有单独的卷标	运算过程中产生错误，以及程序的修改或应用指令的操作数的内容是否有错误。既使语法、电路没有出错，卜述原因也可能产生运算错误。（例）T200Z 虽没有错但运算结果 Z=100 时，T=300，这样，元件编号则溢出。	
	6702	CALL 的嵌套级在 6 层以上		
	6704	FOR-NEXT 的嵌套级在 6 层以上		
	6705	应用指令的操作数在目标元件以外		
	6706	应用指令的操作数的元件号范围和数据值溢出		
	6707	因没有设定文件寄存器的参数而存取了文件寄存器		
	6708	FROM～TO 指令出错		
	6709	其它(IRET,SRET 忘记，FOR～NEXT 关系有错误等)		
	6730	取样时间(TS)在目标范围外（TS=0）	PID 运算停止	产生控制参数的设定值和 PID 运算中产生数据错误。请检查参数。
	6732	输入滤波器常数(a)在目标范围外〔a<0 或 100<a)		
	6733	比例阀(KP)在目标范围外〔KP<0)		
	6734	积分时间(TI)在目标范围外〔TI< 0)		
	6735	微分阀(KD)在目标范围外〔KD<0 或 201<KD)		
	6736	微分时间在目标范围外(TD<0)		
	6740	取样时间(TS)<运算周期	将运算数据作 MAX 值，继续运算。	
	6742	测定值变量溢出(△PV<32768 或 <△PV)		
	6743	偏差溢出(EV<-32768 或 32767<EV〕		
	6744	积分计算值溢出(-32768～32767 以外)		
	6745	因微分阀(KP)溢出，产生微分值溢出		
	6746	微分计算值值溢出(-32768～32767 以外)		
	6747	PID 运算结果溢出(-32768～32767 以外)		
	6760	高速指令(DHSZ 等)超过 6 条限制		

8.3 错误代码存贮

H2U 的错误按下述定时检查，把前项的出错代码存入特殊数据寄存器 D8060～D8067。错误代码存储寄存器

出错项目	电源 OFF→ON	电源 ON 后初次 STOP→RUN 时	其它
M8060 I/O 地址号构成出错	检查	检查	运算中
M8061 PC 硬件出错	检查	-	运算中
M8062 PC/PP 通信出错	-	-	从 PP 接收信号时
M8063 连接模块通值出错	-	-	从对方接受信号时
M8064 参数出错 M8065 语法出错 M8066 电路出错	检查	检查	程序变更时(STOP) 程序传送时(STOP)
M8087 运算出错 M8088 运算出错锁存	-	-	运算中(RUN)

D8060～D8067 各存一个出错内容，同一出错项目产生多次出错时，每当消除出错原因时，仍存储发生中的出错代码。无出错时存入“0”。

8.4 H2U 系列 PLC 内置 MODBUS 从站通讯协议说明

概述：

支持 MODBUS 协议功能码 0x01，0x03，0x05，0x06，0x0f，0x10；通过这些功能码，可读写的线圈有 M，S，T，C，X（只读），Y 等变量；寄存器有 D，T，C。

1．MODBUS **帧格式**

1.1 **功能码** 0x01（01）：**读线圈**

请求帧格式：从机地址+0x01+线圈起始地址+线圈数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x01（功能码）	1 个字节	读线圈
3	线圈起始地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈数量	2 个字节	高位在前，低位在后（N）
5	CRC 校验	2 个字节	高位在前，低位在后

响应帧格式：从机地址+0x01+字节数+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x01（功能码）	1 个字节	读线圈
3	字节数	1 个字节	值：[（N+7）/8]
4	线圈状态	[（N+7）/8] 个字节	每 8 个线圈合为一个字节，最后一个若不足 8 位，未定义部分填 0。前 8 个线圈在第一个字节，最地址最小的线圈在最低位。依次类推
5	CRC 校验	2 个字节	高位在前，低位在后

错误响应：见错误响应帧

1.2 功能码 0x03（03）：读寄存器

请求帧格式：从机地址+0x03+寄存器起始地址+寄存器数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x03（功能码）	1 个字节	读寄存器
3	寄存器起始地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器数量	2 个字节	高位在前，低位在后（N）
5	CRC 校验	2 个字节	高位在前，低位在后

响应帧格式：从机地址+0x03+字节数+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x03（功能码）	1 个字节	读寄存器
3	字节数	1 个字节	值：N*2
4	寄存器值	N*2 个字节	每两字节表示一个寄存器值，高位在前低位在后。寄存器地址小的排在前面
5	CRC 校验	2 个字节	高位在前，低位在后

错误响应：见错误响应帧

1.3 功能码 0x05（05）：写单线圈

请求帧格式：从机地址+0x05+线圈地址+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x05（功能码）	1 个字节	写单线圈
3	线圈地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈状态	2 个字节	高位在前，低位在后。非 0 即为有效
5	CRC 校验	2 个字节	高位在前，低位在后

响应帧格式：从机地址+0x05+线圈地址+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x05（功能码）	1 个字节	写单线圈
3	线圈地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈状态	2 个字节	高位在前，低位在后。非 0 即为有效
5	CRC 校验	2 个字节	高位在前，低位在后

错误响应：见错误响应帧

1.4 功能码 0x06（06）：写单个寄存器

请求帧格式：从机地址+0x06+寄存器地址+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x06（功能码）	1 个字节	写单寄存器
3	寄存器地址	2 个字节	高位在前，低位在后，见寄存器值编址
4	寄存器值	2 个字节	高位在前，低位在后。非 0 即为有效
5	CRC 校验	2 个字节	高位在前，低位在后

响应帧格式：从机地址+0x06+寄存器地址+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x06（功能码）	1 个字节	写单寄存器
3	寄存器地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器值	2 个字节	高位在前，低位在后。非 0 即为有效
5	CRC 校验	2 个字节	高位在前，低位在后

错误响应：见错误响应帧

1.5 功能码 0x0f（15）：写多个线圈

请求帧格式：从机地址+0x0f+线圈起始地址+线圈数量+字节数+线圈状态+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x0f（功能码）	1 个字节	写多个单线圈
3	线圈起始地址	2 个字节	高位在前，低位在后，见线圈编址
4	线圈数量	2 个字节	高位在前，低位在后。N，最大为 1968
5	字节数	1 个字节	值：值：[（N+7）/8]
6	线圈状态	[（N+7）/8] 个字节	每 8 个线圈合为一个字节，最后一个若不足 8 位，未定义部分填 0。前 8 个线圈在第一个字节，最地址最小的线圈在最低位。依次类推
7	CRC 校验	2 个字节	高位在前，低位在后

响应帧格式：从机地址+0x05+线圈起始地址+线圈数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x0f（功能码）	1 个字节	写多个单线圈
3	线圈起始地址	2 个字节	高位在前，低位在后，见线圈编址

4	线圈数量	2 个字节	高位在前，低位在后。
5	CRC 校验	2 个字节	高位在前，低位在后

错误响应：见错误响应帧

1.6 功能码 0x10（16）：写多个寄存器

请求帧格式：从机地址+0x10+寄存器起始地址+寄存器数量+字节数+寄存器值+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x10（功能码）	1 个字节	写多个寄存器
3	寄存器起始地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器数量	2 个字节	高位在前，低位在后。N，最大为 120
5	字节数	1 个字节	值：N*2
6	寄存器值	N*2（N*4）	
7	CRC 校验	2 个字节	高位在前，低位在后

响应帧格式：从机地址+0x05+线圈起始地址+线圈数量+CRC 检验

序号	数据(字节)意义	字节数量	说明
1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	0x10（功能码）	1 个字节	写多个寄存器
3	寄存器起始地址	2 个字节	高位在前，低位在后，见寄存器编址
4	寄存器数量	2 个字节	高位在前，低位在后。N，最大为 120
5	CRC 校验	2 个字节	高位在前，低位在后

错误响应：见错误响应帧

1.7 错误响应帧

错误响应：从机地址 +（功能码+0x80）+ 错误码 + CRC 校验

序号	数据(字节)意义	字节数量	说明

1	从机地址	1 个字节	取值 1~247，由 D8121 设定
2	功能码+0x80	1 个字节	错误功能码
3	错误码	1 个字节	1~4
4	CRC 校验	2 个字节	高位在前，低位在后

2 变量编址

2.1 线圈编址

线圈：指位变量，只有两种状态 0 和 1。在本 PLC 中包含 M，S，T，C，X，Y 等变量。

变量名称	起始地址	线圈数量	说明
M0~3071	0（0）	3072	
M8000~M8256	0x1F40（8000）	256	
S0~S999	0xE000（57344）	1000	
T0~T256	0xF000（61440）	256	
C0~C255	0xF400（62464）	256	
X0~X255	0xF800（63488）	256	
Y0~Y255	0xFC00（64512）	256	

2.2 寄存器编址

寄存器:指 16 位或 32 位变量,在本 PLC 中,16 位变量包含 D,T,C0~199;32 位变量为 C200~255

变量名称	起始地址	寄存器数量	说明
D0~D8255	0（0）	8256	
T0~T255	0xF000（61440）	256	
C0~C199	0xF400（62464）	200	
C200~C255	0xF700（63232）	56	32 位寄存器

说明：

通过 MODBUS 访问 C200~C255 段 32 位寄存器时，一个寄存器作两寄存器看待，一个 32 位寄存器占用两个 16 寄存器空间。比如用户要读或写 C205~C208 这 4 个寄存器，MODBUS 地址为 0xF70A（0xF700+10），寄存器数量 8（4*2）。

32 位寄存器不支持写单个寄存器（0x06）功能码。

8.5 H2U 系列 3A 扩展小板使用说明

扩展小板选件 H2U-3A-BD 小板用于 2 路模拟量的输入和 1 路模拟量输出。

PLC 主模块与 H2U-3A-BD 小板之间采用了通讯帧的方式进行数据交互，通讯帧由 PLC 自动组织和收发处理，使用时，用户程序只需在特定的系统缓冲区写入需要访问的扩展小板类型、输出值，在特定的寄存器中读取输入端口的检测数据即可。

访问 3A 扩展小板时的相关寄存器及其含义：

序号	数据名称	PLC 发送到扩展卡	访问 H2U-3A-BD 时的写入值举例
1	控制字	D8220（扩展卡 ID＋命令码）	3A21h 3A 2 1 分别表示此单元是 3A 扩展单元，2 个模拟量输入通道，1 个模拟量输出通道。
2	数据 1	D8221	模拟量输出通道 1 的输出设定值（量程：0-10000）
3	数据 2	D8222	/
4	数据 3	D8223	/
5	数据 4	D8224	/
6	数据 5	D8225	/
7	数据 6	D8226	/
8	数据 7	D8227	/
9	数据 8	D8228	/
10	CRC 校验码	D8229	PLC 系统自动计算，用户程序不能更改

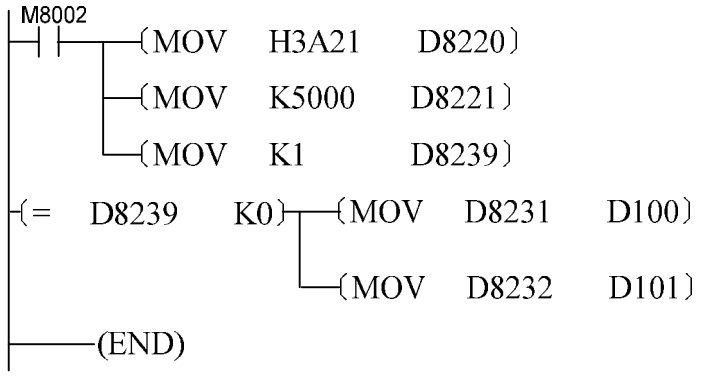
H2U-3A-BD 应答的数据及存放地址：

序号	数据名称	PLC 从扩展卡接收	H2U-3A-BD 答复的数据
1	控制字	D8230	3A21h
2	数据 1	D8231	模拟量输入通道 1 的当前值（量程：0-10000）
3	数据 2	D8232	模拟量输入通道 2 的当前值（量程：0-10000）
4	数据 3	D8233	模拟量输出通道 1 的输出设定值（量程：0-10000）
5	数据 4	D8234	/
6	数据 5	D8235	/
7	数据 6	D8236	/
8	数据 7	D8237	/
9	数据 8	D8238	/
10	数据 9	D8239	通信错误计数，若通信正常，自动置 0；非 0，表示通信连续出错次数

使用编程举例：

使用 H2u 和 3A 小板，要求 AO 输出通道输出 5V；将 AI1 输入通道的采样值放入 D100，AI2 输入通道的采样值放入 D102。

程序如下：



说明：当 D8239 的值非 0 时，表示 PLC 尚未正确读取到扩展小板的值，编程时需注意；AO 通电的输出值 D8221 可根据具体应用在用户程序中进行刷新。

8.6 H2U 系列 MTQ 型与 MT 型的软件差别

H2U 系列的 MTQ 型 PLC，具有 6 路高速输入,5 路高速输出，高速信号处理频率提升，以满足具有多路高速应用的需求。

硬件响应特性比较：

输入端口	MT 型	MTQ 型
X0	100kHz	100kHz
X1	100kHz	100kHz
X2	100kHz ※	100kHz
X3	10kHz	100kHz
X4	10kHz	100kHz
X5	10kHz	100kHz
其它 X 端口	指标相同	
Y0	100kHz	100kHz
Y1	100kHz	100kHz
Y2	100kHz ※	100kHz
Y3	2kHz	100kHz
Y4	2kHz	100kHz
其它 Y 端口	指标相同	

※H2U-2416MR/T 和 H2U-3624MR/T 型号为 10kHz。

高速计数器最高频率指标比较：

AB 相计数器	H2U-2416MR/T H2U-3624MR/T	H2U-1616MR/T H2U-3232MR/T H2U-4040MR/T H2U-6464MR/T	H2U-3232MTQ
C253	10kHz	30kHz	30kHz
C255	10kHz	30kHz	30kHz

支持指令比较：

指令	MT 支持的 输出端口	MTQ 支持的 输出端口
PLSY	Y0、Y1	Y0、Y1、Y2、Y3、Y4
PLSR	Y0、Y1	Y0、Y1、Y2、Y3、Y4
PLSV	Y0、Y1	Y0、Y1、Y2、Y3、Y4
PWM	Y0、Y1	Y0、Y1、Y2、Y3、Y4
DRVI	Y0、Y1	Y0、Y1、Y2、Y3、Y4
DRVA	Y0、Y1	Y0、Y1、Y2、Y3、Y4
ZRN	Y0、Y1	Y0、Y1、Y2、Y3、Y4

MTQ 型高速脉冲输出时涉及的系统监控寄存器说明：

M8146	Y001脉冲输出停止	D8146	DRVI,DRVA 执行时的最高速度[默认100,000]
M8147	Y000脉冲输出监控	D8147	
M8148	Y001脉冲输出监控	D8148	DRVI,DRVA执行时加减速时间[默认100]
M8149	Y002脉冲输出监控	D8149	保留
M8150	Y003脉冲输出监控	D8150	PLSY&PLSR输出Y002对应的脉冲个数累积值
M8151	Y004脉冲输出监控	D8151	
M8152	Y002脉冲输出停止	D8152	PLSY&PLSR输出Y003对应的脉冲个数累积值
M8153	Y003脉冲输出停止	D8153	
M8154	Y004脉冲输出停止	D8154	PLSY&PLSR输出Y004对应的脉冲个数累积值
M8155	保留	D8155	
M8156	保留	D8156	Y0端口清零信号定义(ZRN)[默认5=Y005]
M8157	保留	D8157	Y1端口清零信号定义(ZRN)[默认6=Y006]
M8158	保留	D8158	Y2端口清零信号定义(ZRN)[默认7=Y007]
M8159	保留	D8159	Y3端口清零信号定义(ZRN)[默认8=Y010]
M8160	(XCH)的SWAP功能	D8160	Y4端口清零信号定义(ZRN)[默认9=Y011]
M8161	ASC/RS/ASCII/HEX/CCD 的位处理模式	D8161	保留

M8162	高速并联连接模式	D8162	保留
M8163	保留	D8163	保留
M8164	(FROM/TO)传送点数可变模式	D8164	(FROM/TO)传送点数指定模式
M8165	保留	D8165	PLSR,DRVI,DRVA执行时减速时间[默认100] 由M8135决定是否有效[Y0]
M8166	保留	D8166	PLSR,DRVI,DRVA执行时减速时间[默认100] 由M8136决定是否有效[Y1]
M8167	(HEY)HEX数据处理功能	D8167	PLSR,DRVI,DRVA执行时减速时间[默认100] 由M8137决定是否有效[Y2]
M8168	(SMOV)HEX数据处理功能	D8168	PLSR,DRVI,DRVA执行时减速时间[默认100] 由M8138决定是否有效[Y3]
M8169	保留	D8169	PLSR,DRVI,DRVA执行时减速时间[默认100] 由M8139决定是否有效[Y4]