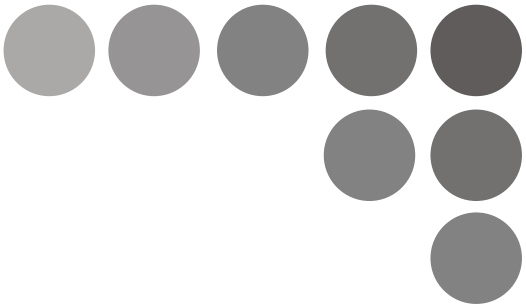


Inovance

H2U Series PLC

programming manual

H2U系列可编程控制器
指令及编程手册

资料编号：X6210031

V1.0

User Manual

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价比控制产品，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 顺序功能图等编程方法。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，部分 MT 型号增加为具有三路高速输出功能，MTQ 版本则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，反馈手册中的错误和遗漏。

公司网页：www.inovance.cn

信息交流：plc@inovance.cn

目　录

前　言i

手册版本信息 - 1 -

指令详解索引 - 1 -

手册快查引导 - 7 -

第一章 梯形图及梯形图程序 - 10 -

1.1 梯形图的编程特点： - 10 -

1.2 梯形图编程时使用的元件符号： - 10 -

1.3 PLC 的执行原理 - 11 -

1.4 PLC 数值的基本知识 - 14 -

第二章 H2U 系列 PLC 的使用方法 - 21 -

2.1 使用 PLC 的软件硬件需求 - 22 -

2.2 编程与用户程序下载 - 23 -

2.3 与 HMI 的配合使用 - 23 -

第三章 软元件说明 - 25 -

3.1 输入继电器 X - 26 -

3.2 输出继电器 Y - 27 -

3.3 辅助继电器 M - 27 -

3.4 状态继电器 S - 28 -

3.5 计时器 T - 28 -

3.6 计数器 C - 30 -

3.7 寄存器 D - 35 -

3.8 子程序与中断指针 P、I - 39 -

3.9 常数 K、H - 41 -

3.10 控制器软元件规格 - 42 -

第四章 指令 - 46 -

4.1 STL/SFC 指令 - 46 -

4.1.1 STL 编程指令 - 46 -

4.1.2 SFC 顺序功能图编程 - 49 -

4.1.2.1 SFC 的编程特点： - 49 -

4.1.2.2 SFC 的编程方法： - 50 -

4.2 应用指令表 - 57 -

4.3 指令解释 - 61 -

4.3.1 基本指令解释 - 61 -

4.3.2 应用指令解释 - 67 -

4.3.2.1 程序流程（00~09） - 67 -

4.3.2.2 传送与比较（10~19） - 78 -

4.3.2.3 四则逻辑运算（20~29） - 90 -

4.3.2.4 循环移位（30~39） - 101 -

4.3.2.5 数据处理（40~49） - 109 -

4.3.2.6 高速处理（50~59） - 120 -

4.3.2.7 方便指令（60~69）..... - 144 -
4.3.2.8 外围设备 I/O（70~79）..... - 167 -
4.3.2.9 外围 SER 设备（80~88）..... - 185 -
4.3.2.10 浮点数（110~147）..... - 207 -
4.3.2.11 定位控制（155~159）..... - 221 -
4.3.2.12 时钟运算（160~169）..... - 233 -
4.3.2.13 外围设备（170~177）..... - 243 -
4.3.2.14 接点比较（224~246）..... - 247 -

附 录..... - 253 -

5.1 系统特殊软元件： ..... - 253 -
5.2 出错信息说明 ..... - 261 -
5.3 错误代码存贮 ..... - 264 -
5.4 H2U 系列 PLC 内置 MODBUS 从站通讯协议说明..... - 264 -
5.5 H2U 系列 3A/6A/6B 扩展卡使用说明..... - 268 -
5.6 H2U 通讯扩展卡说明 ..... - 273 -
5.7 H2U 系列 MTQ 型与 MT 型的软件差别 ..... - 275 -
5.8 H2U 系列高速处理指令的增强功能说明..... - 277 -
5.9 程序流程控制指令的相互关系..... - 282 -
5.10 部分特殊继电器和寄存器功能说明 ..... - 284 -

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
V1.00	2009-10-09	增加 IST 指令说明 修订附录 5.5 节扩展卡的编程说明 增加附录 5.8 节的高速处理增强功能说明 增加附录 5.9 程序流程控制指令的相互关系 增加附录 5.10 部分特殊继电器和寄存器功能说明

指令详解索引

简单逻辑指令

FNC NO.	指令助记符	功能	页码
	LD	加载常开接点	61
	LDI	加载常闭接点	61
	LDP	取脉冲上升沿	61
	LDF	取脉冲下降沿	61
	AND	串联常开接点	61
	ANI	串联常闭接点	61
	ANB	串联回路方块	62
	ANDP	与脉冲上升沿检测串行连接	61
	ANDF	与脉冲(F)下降沿检测串行连接	61
	OR	并联常开接点	62
	ORI	并联常闭接点	62
	ORB	并联回路方块	62
	ORP	或脉冲上升沿检测并行连接	62
	ORF	或脉冲(F)下降沿检测并行连接	62
	OUT	驱动线圈	63
	SET	置位动作保存线圈	63
	RST	接点或缓存器清除	64
	PLS	脉冲上升沿检测线圈	64
	PLF	脉冲下降沿检测线圈	64
	MC	主控公用串行接点用线圈指令	64
	MCR	主控复位公用串行接点解除指令	64
	MPS	存入堆栈	63
	MRD	读出堆栈	63
	MPP	读出堆栈	63
	NOP	无动作	65
	INV	运算结果取反	65
	END	程序结束	65
	P	指针	65
	I	中断插入指针	66

应用指令（以 FNC NO 为序）

分类	FNC NO.	指令助记符	功能	页码
程序流程	00	CJ	条件跳转	67
	01	CALL	子程序调用	68
	02	SRET	子程序返回	68
	03	IRET	中断返回	70
	04	EI	中断许可	70
	05	DI	中断禁止	70
	06	FEND	主程序结束	74
	07	WDT	监控定时器	75
	08	FOR	循环范围开始	76
	09	NEXT	循环范围终了	76
传送与比较	10	CMP	比较	78
	11	ZCP	区域比较	79
	12	MOV	传送	80
	13	SMOV	移位传送	81
	14	CML	倒转传送	83
	15	BMOV	一并传送	85
	16	FMOV	多点传送	86
	17	XCH	交换	87
	18	BCD	BCD 转换	88
	19	BIN	BIN 转换	89
四则逻辑运算	20	ADD	BIN 加法	90
	21	SUB	BIN 减法	91
	22	MUL	BIN 乘法	92
	23	DIV	BIN 除法	93
	24	INC	BIN 加 1	95
	25	DEC	BIN 减 1	96
	26	WAND	逻辑字与	97
	27	WOR	逻辑字或	97
	28	WXOR	逻辑字异或	97
	29	NEG	求补码	99

分类	FNC NO.	指令助记符	功能	页码
循环移位	30	ROR	循环右移	101
	31	ROL	循环左移	101
	32	RCR	带进位循环右移	102
	33	RCL	带进位循环左移	102
	34	SFTR	位右移	103
	35	SFTL	位左移	103
	36	WSFR	字右移	105
	37	WSFL	字左移	105
	38	SFWR	移位写入	107
	39	SFRD	移位读出	108
数据处理	40	ZRST	批次复位	109
	41	DECO	译码	110
	42	ENCO	编码	111
	43	SUM	ON 位数	112
	44	BON	ON 位数判定	113
	45	MEAN	平均值	114
	46	ANS	信号报警置位	115
	47	ANR	信号报警器复位	116
	48	SQR	BIN 开方	117
	49	FLT	整数→浮点数转换	118
高速处理	50	REF	输入输出刷新	120
	51	REFF	滤波器调整	122
	52	MTR	矩阵输入	123
	53	HSCS	比较置位（高速计数器）	125
	54	HSCR	比较复位（高速计数器）	128
	55	HSZ	比较区间（高速计数器）	129
	56	SPD	脉冲密度	134
	57	PLSY	脉冲输出	137
	58	PWM	脉冲调制	140
	59	PLSR	带加减速的脉冲输出	141

分类	FNC NO.	指令助记符	功能	页码
方便指令	60	IST	状态初始化	144
	61	SER	数据查找	150
	62	ABSD	凸轮控制(绝对方式)	152
	63	INCD	凸轮控制(增量方式)	154
	64	TTMR	示教定时器	156
	65	STMR	特殊定时器	157
	66	ALT	交替输出	160
	67	RAMP	斜波信号	161
	68	RTOC	旋转工作台控制	163
	69	SORT	数据排列	165
外围设备I/O	70	TKY	数字键输入	167
	71	HKY	16 键输入	169
	72	DSW	数字式开关	171
	73	SEGD	7 段码译码	173
	74	SEGL	7 段码扫描显示	174
	75	ARWS	方向开关	176
	76	ASC	ASCII 码转换	178
	77	PR	ASCII 码打印输出	180
	78	FROM	BFM 读出	181
	79	TO	BFM 写入	182

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	185
	81	PRUN	8 进制位传送	192
	82	ASCI	HEX-ASCII 转换	193
	83	HEX	ASCII-HEX 转换	195
	84	CCD	校验码	197
	88	PID	PID 运算	199
浮点数	110	ECMP	2 进制浮点数比较	207
	111	EZCP	2 进制浮点数区间比较	208
	118	EBCD	2 进制-10 进制浮点数转换	209
	119	EBIN	10 进制-2 进制浮点数转换	210
	120	EADD	2 进制浮点数加法	211
	121	ESUB	2 进制浮点数减法	212
	122	EMUL	2 进制浮点数乘法	213
	123	EDIV	2 进制浮点数除法	214
	127	ESQR	2 进制浮点数开方	215
	129	INT	2 进制浮点数-BIN 整数转换	216
	130	SIN	浮点数 SIN 运算	217
	131	COS	浮点数 COS 运算	218
	132	TAN	浮点数 TAN 运算	219
	147	SWAP	上下字节变换	220
定位	155	ABS	ABS 位置数读取	221
	156	ZRN	原点回归	223
	157	PLSV	可变脉冲输出	225
	158	DRVI	相对定位	227
	159	DRVA	绝对定位	230

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	233
	161	TZCP	时钟数据区域比较	235
	162	TADD	时钟数据加法	236
	163	TSUB	时钟数据减法	237
	166	TRD	时钟数据读出	238
	167	TWR	时钟数据写入	239
	169	HOUR	计时器	241
外围设备	170	GRY	格雷码变换	243
	171	GBIN	格雷码逆变换	244
	176			
	177			
接点比较	224	LD=	(S1)=(S2)	248
	225	LD>	(S1)>(S2)	248
	226	LD<	(S1)<(S2)	248
	228	LD<>	(S1)<>(S2)	248
	229	LD<=	(S1)<=(S2)	248
	230	LD>=	(S1)>=(S2)	248
	232	AND=	(S1)=(S2)	250
	233	AND >	(S1)>(S2)	250
	234	AND <	(S1)<(S2)	250
	236	AND <>	(S1)<>(S2)	250
	237	AND <=	(S1)<=(S2)	250
	238	AND >=	(S1)>=(S2)	250
	240	OR=	(S1)=(S2)	252
	241	OR >	(S1)>(S2)	252
	242	OR <	(S1)<(S2)	252
	244	OR <>	(S1)<>(S2)	252
	245	OR <=	(S1)<=(S2)	252
	246	OR >=	(S1)>=(S2)	252

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	221
	ABSD	62	凸轮控制(绝对方式)	152
	ADD	20	BIN 加法	90
	ALT	66	交替输出	160
	AND=	232	(S1)=(S2)	250
	AND>	233	(S1)>(S2)	250
	AND<	234	(S1)<(S2)	250
	AND<>	236	(S1)<>(S2)	250
	AND<=	237	(S1)<=(S2)	250
	AND>=	238	(S1)>=(S2)	250
	ANR	47	信号报警复位	116
	ANS	46	信号报警置位	115
	ARWS	75	箭形开关	176
	ASC	76	ASCII 码转换	178
	ASCI	82	HEX→ASCII 转换	193
B	BCD	18	BCD 转换	88
	BIN	19	BIN 转换	89
	BMOV	15	成批转换	85
	BON	44	ON 位数判定	113
C	CALL	01	子程序调用	68
	CCD	84	检验码	197
	CJ	00	条件跳转	67
	CML	14	取反传送	83
	CMP	10	比较	78
	COS	131	浮点数 COS 运算	218
D	DEC	25	BIN 减 1	96
	DECO	41	译码	110
	DI	05	中断禁止	70
	DIV	23	BIN 除法	93
	DRVA	159	绝对定位	230
	DRVI	158	相对定位	227
	DSW	72	数字式开关	171

分类	指令助记符	FNC NO.	功能	页码
E	EADD	120	2 进制浮点数加法	211
	EBCD	118	2 进制-10 进制浮点数转换	209
	EBIN	119	10 进制-2 进制浮点数转换	210
	ECMP	110	2 进制浮点数比较	207
	EDIV	123	2 进制浮点数除法	214
	EI	04	中断许可	70
	EMUL	122	2 进制浮点数乘法	213
	ENCO	42	编码	111
	ESQR	127	2 进制浮点数开方	215
	ESUB	121	2 进制浮点数减法	212
	EZCP	111	2 进制浮点数区间比较	208
F	FEND	06	主程序结束	74
	FLT	49	BIN 整数→2 进制浮点数转换	118
	FMOV	16	多点传送	86
	FOR	08	循环范围开始	76
	FROM	78	BFM 读出	181
G	GBIN	171	格雷码逆变换	244
	GRY	170	格雷码变换	243
H	HEX	83	ASCII-HEX 转换	195
	HKY	71	16 键输入	169
	HOUR	169	计时仪	241
	HSCR	54	比较复位（高速计数器）	128
	HSCS	53	比较置位（高速计数器）	125
	HSZ	55	区间比较（高速计数器）	129

分类	指令助记符	FNC NO.	功能	页码
I	INC	24	BIN 加 1	95
	INCD	63	凸轮控制(增量方式)	154
	INT	129	浮点数-整数转换	216
	IRET	03	中断返回	70
	IST	60	状态初始化	144
L	LD=	224	(S1)=(S2)	248
	LD>	225	(S1)>(S2)	248
	LD<	226	(S1)<(S2)	248
	LD<>	228	(S1)<>(S2)	248
	LD<=	229	(S1)<=(S2)	248
	LD>=	230	(S1)>=(S2)	248
M	MEAN	45	平均值	114
	MOV	12	传送	80
	MTR	52	矩阵输入	123
	MUL	22	BIN 乘法	92
N	NEG	29	求补码	99
	NEXT	09	循环范围终了	76
O	OR=	240	(S1)=(S2)	252
	OR >	241	(S1)>(S2)	252
	OR <	242	(S1)<(S2)	252
	OR <>	244	(S1)<>(S2)	252
	OR <=	245	(S1)<=(S2)	252
	OR >=	248	(S1)>=(S2)	252
P	PID	88	PID 运算	199
	PLSV	157	可变速脉冲输出	225
	PLSY	57	脉冲输出	137
	PLSR	59	有加减速脉冲输出	141
	PR	77	ASCII 键打印输出	180
	PRUN	81	8 进制输送	192
	PWM	58	脉冲幅度调整	140
R	RAMP	67	斜坡信号	161
	RCL	33	带进位的循环左移	102
	RCR	32	带进位的循环右移	102
	REF	50	输入输出刷新	120
	REFF	51	滤波器调整	122
	ROL	31	循环左移	101
	ROR	30	循环右移	101
	ROTC	68	旋转工作台控制	163
	RS	80	串行数据输送	185

分类	指令助记符	FNC NO.	功能	页码
S	SEGD	73	7 段码译码	173
	SEGL	74	7 点码按时间分割显示	174
	SER	61	数据查找	150
	SFRD	39	位移读出	108
	SFTL	35	位左移	103
	SFTR	34	位右移	103
	SFWR	38	移位写入	107
	SIN	130	浮点数 SIN 运算	217
	SMOV	13	移位传送	81
	SORT	69	数据排列	165
	SPD	56	脉冲密度	134
	SQR	48	BIN 开方	117
	SRET	02	子程序返回	68
	STMR	65	特殊定时器	157
	SUB	21	BIN 减法	91
	SUM	43	ON 位数	112
	SWAP	147	上下字节交换	220
T	TADD	162	时钟数据加法	236
	TAN	132	浮点数 TAN 运算	219
	TCMP	160	时钟数据比较	233
	TKY	70	数字键输入	167
	T0	79	BFM 写入	182
	TRD	166	时钟数据读出	238
	TSUB	163	时钟数据减法	237
	TTMR	64	示教定时器	156
	TWR	167	时钟数据写入	239
	TZCP	161	时钟数据区间比较	235

分类	指令助记符	FNC NO.	功能	页码
W	WDT	07	监控定时器	75
	WOR	27	逻辑字或	97
	WSFL	37	字左移	105
	WSFR	36	字右移	105
	WXCR	28	逻辑字异或	97
X	XCH	17	交换	87
Z	ZCP	11	区间比较	79
	ZRN	156	原点回归	223

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 P61～P66 指令详解
2	应用指令的解释	先根据指令名查阅 P1～P6，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 4.1.1 节
10	通讯格式的设置方法	查阅 RS 指令详解
11	各种通讯协议的设置方法	查阅 RS 指令详解
12	如何使用 MODBUS 主站指令	查阅 RS 指令详解
13	如何用 MODBUS 访问 H2U	MODBUS 内置从机协议定义
14	如何使用模拟量扩展卡	查阅 5.5 附录
15	MTQ 的高速功能与使用	参见 5.7 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	增强版的高速处理指令	参见 5.8 附录，增强高速指令
18	部分特殊继电器和寄存器功能说明	参见 5.10 附录

1

梯形图及梯形图程序

第一章 梯形图及梯形图程序

1.1 梯形图的编程特点：

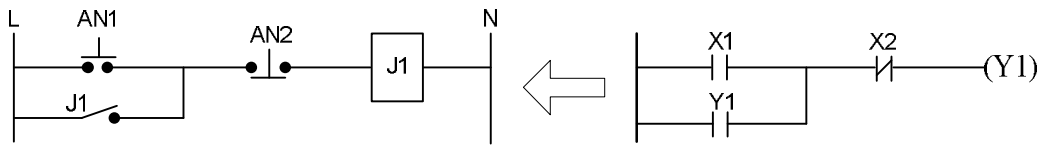
PLC 中梯形图编程方法是仿传统继电器控制系统的电气原理设计的一种设计方法，设计中使用的元件如按钮开关 X、中间继电器 M、时间继电器 T、计数器 C、触点等，都和实际的电气元件的特性相似。

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”，PLC 中同一个继电器的“触点”可被无限次使用，我们可认为一个继电器（无论中间继电器 M、时间继电器 T、还是计数器 C）元件，都具有无限个“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。

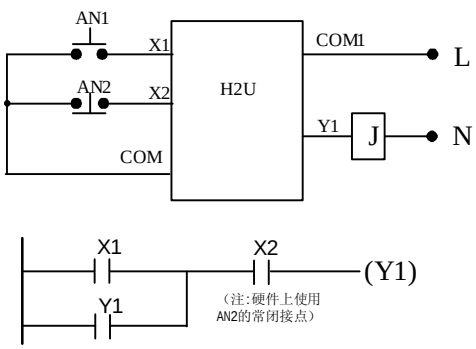
梯形图的设计思想与传统继电器控制系统的设计方法基本相同，以常见的电磁开关的电气原理为例：



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。

若按右图设计 PLC 的信号输入连接和梯形图编程便可实现相同的起停控制功能了。（出于安全的考虑，停止按键一般用常闭型接点。）



用户程序梯形图

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、	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

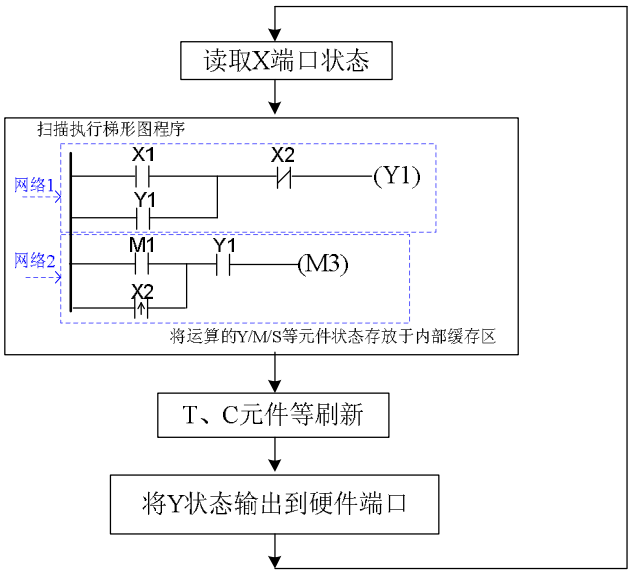
符号	说明	动作特性
	输出 Y 的触点、中间继电器 M、时间继电器 T、计数器 C 的输出触点等。对于 Y、M、T、C 等元件，在未动作状态也为 OFF。	Y：当 Y 继电器的“线圈”得电时为 ON，否则为 OFF。Y 最后状态将对应于 PLC 的输出 Y 端口的状态。
		M：当 M 继电器的“线圈”得电时为 ON，否则为 OFF
		S：当 S 作为普通标志元件使用时，S 继电器的“线圈”得电时为 ON，否则为 OFF
		T：当对应的时间继电器线圈得电，且计时时间达到设定的时间，状态为 ON；否则为 OFF
		C：当对应的计数器的读数达到设定的时间，状态为 ON；否则为 OFF
—|/|—	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	步进指令状态的转移
—( )	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，其常闭型触点动作断开，“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，其常闭型触点动作断开。
		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，其常闭型触点动作断开。清除其“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、

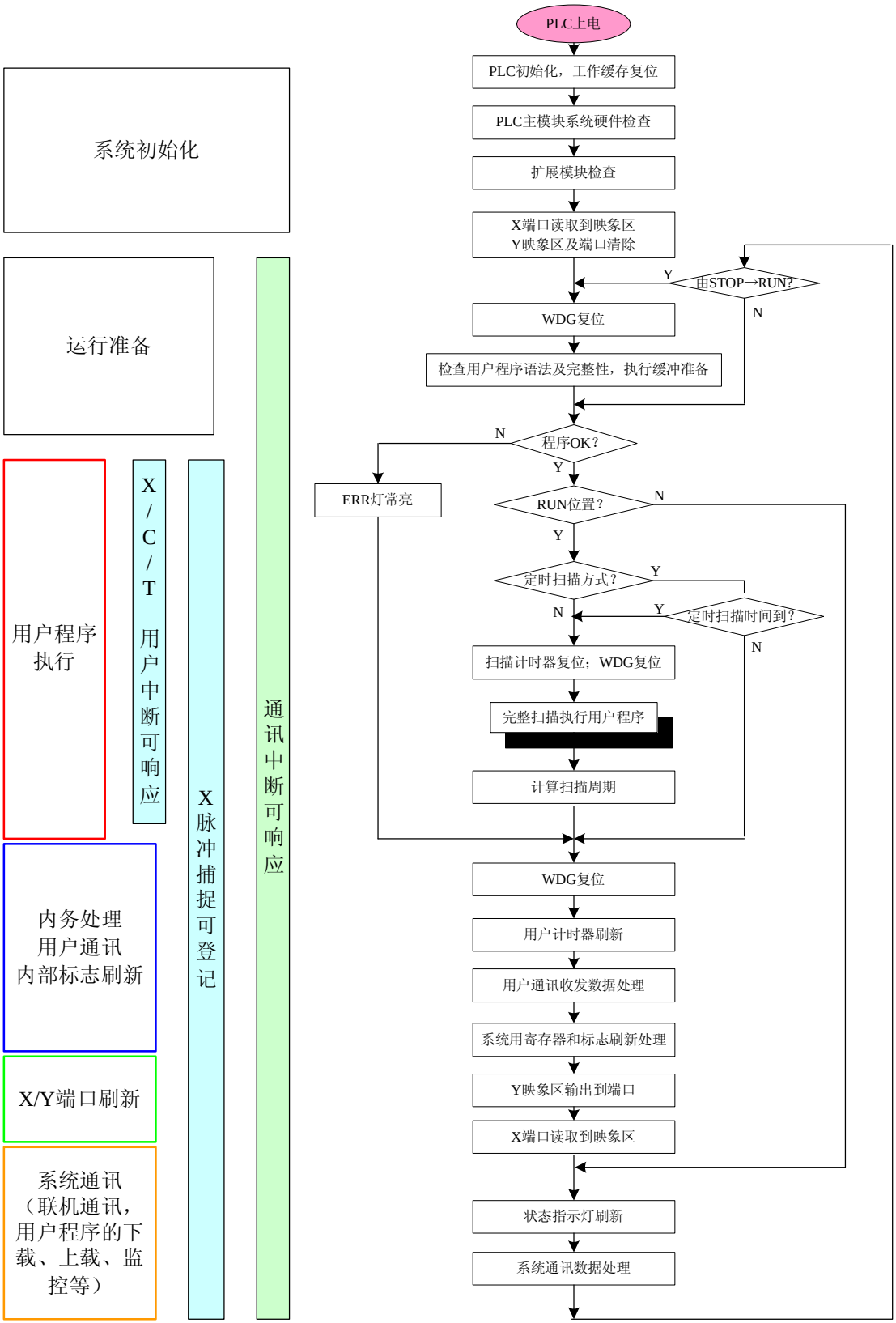
将 Y 状态缓存状态输出到 PLC 的 Y 硬件端口等，这些工作内容周而复始的进行，其中的扫描执行用户程序是 PLC 的核心工作，过程如下例图：



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块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

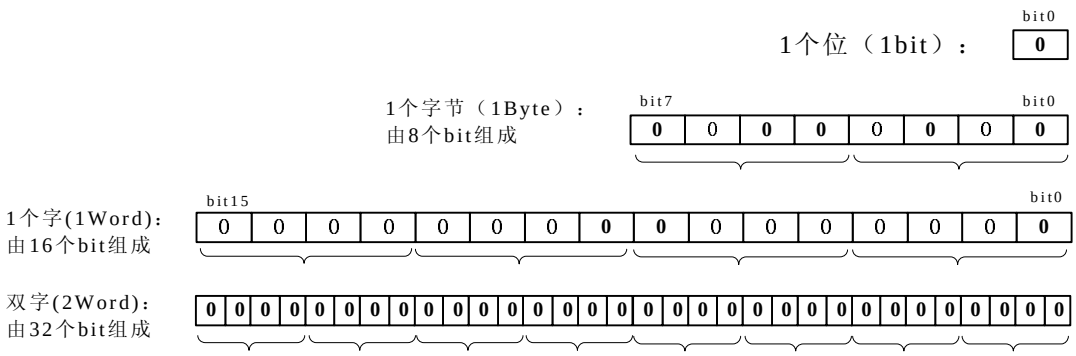
H2U 系列 PLC 内部采用高性能 32bit 作为核心处理器，其工作原理与其他的计算机设备是相似的。所有的 CPU 处理器采用的都是二进制码作为内部处理数据的格式，“数据”在计算机内部是以“信号电平”的方式进行处理的，其中信号电平只有“低”或“高”两个状态，分别对应于二进制数的“0”或“1”，信号电路中不会出现电平误判，可确保处理结果的正确性。

ı 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**

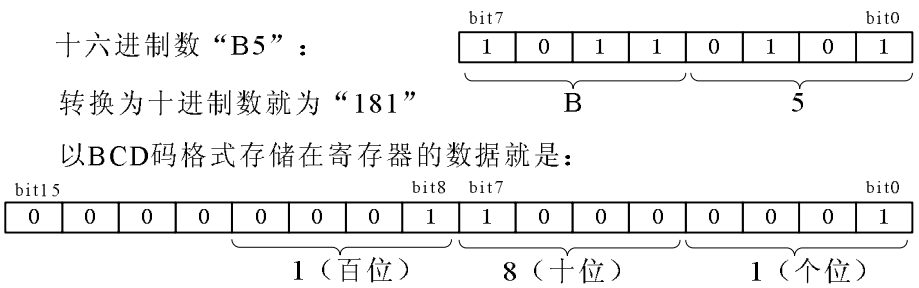
我们生活中习惯使用的数据是采用“十进制”，基本数字为 0～9 共 10 个数，若“9”＋“1”计数，便进位处理得到“10”。

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；…..

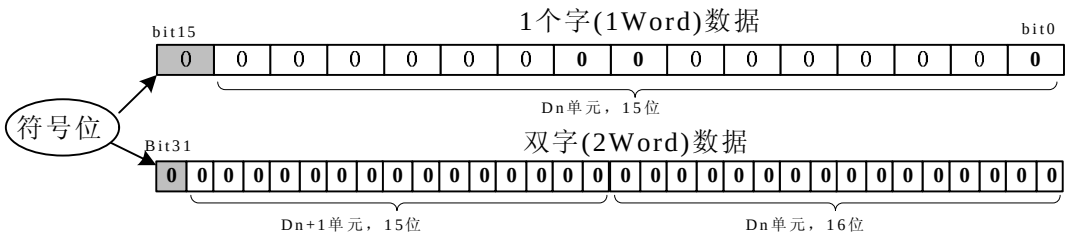
对于本例子中，将“B5”转换为十进制数即为（B×16＋5×1）=（11×16＋5）＝181。

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。

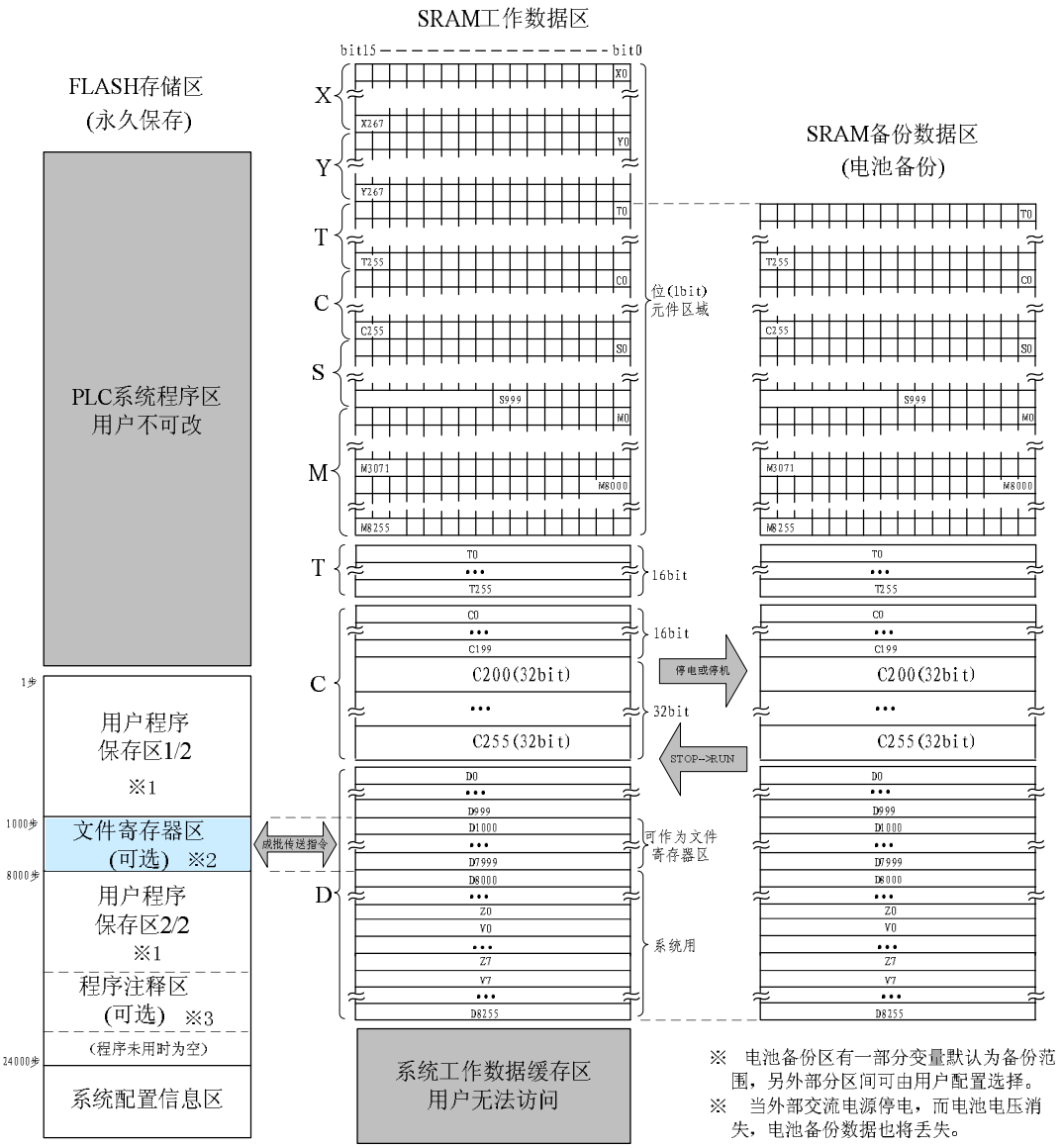
浮点计算是指浮点数参与的运算，这种运算通常伴随着因为无法精确表示而进行的近似或舍入。在 PLC 中有模拟量信号处理和运算时，可能用到浮点数。

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。

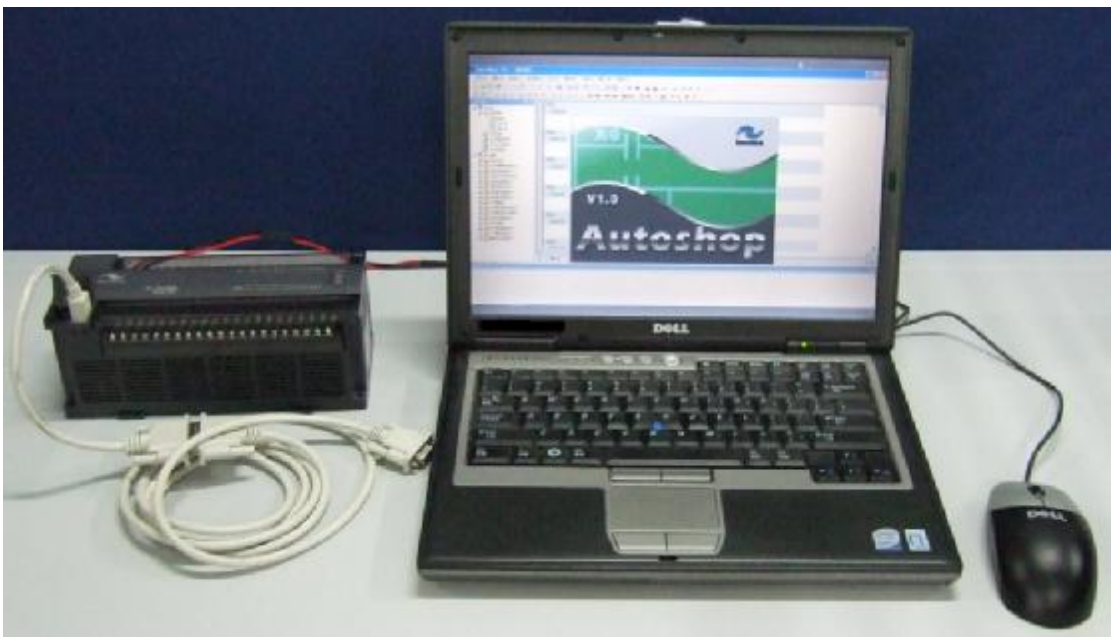
2

H_{2U} 系列 PLC 的使用方法

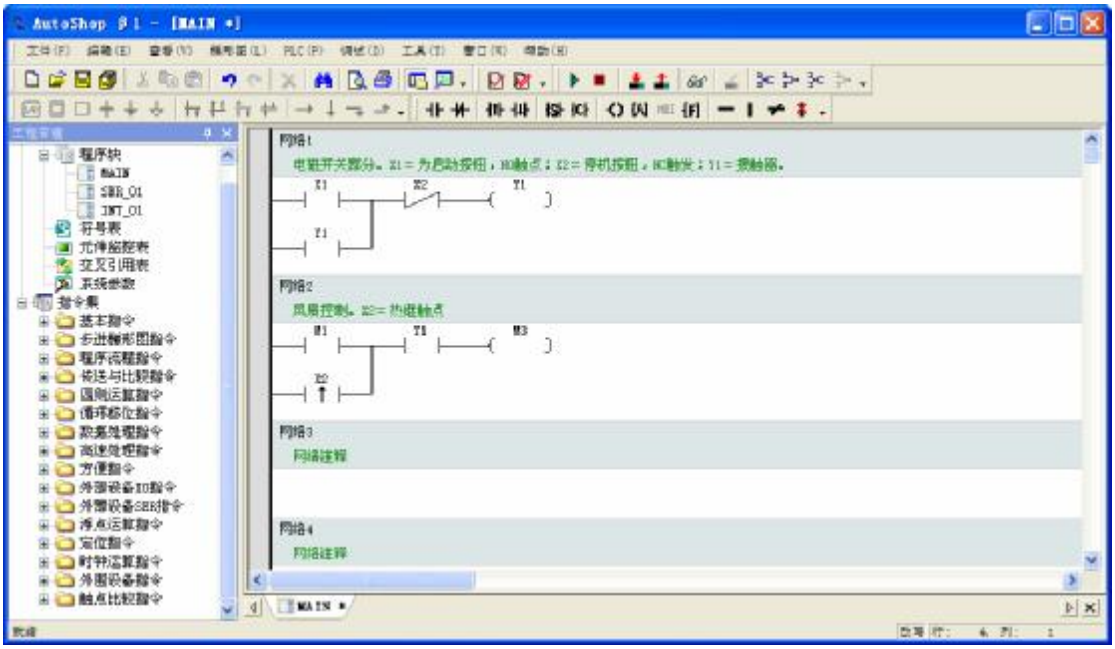
第二章 H2U 系列 PLC 的使用方法

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 以上；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。

2.3 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

第三章 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-X043	Y000-Y027
H2U-3624MTQ	X000-X043	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 X43，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 X44~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的中间变量，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 变量还具有掉电保存特性。

M 数量总计	一般用	停电保持用	停电保持专用	特殊用
3328 点	M0-M499 500 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，可变更成停电保持领域。

※2．停电保持领域。使用参数设定，可变更成非停电保持领域。

※3．停电保持领域，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

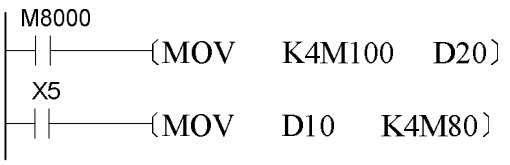
M8033 ：停止时保持输出

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 ※1	S0-S9 （10）点	S500-S899 400 点 ※2	S900-S999 100 点※2

※1：非停电保持领域。通过参数的设定可变更成停电保持的领城。

※2：停电保持领域。通过参数的设定可变更成非停电保持的领城。

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	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：

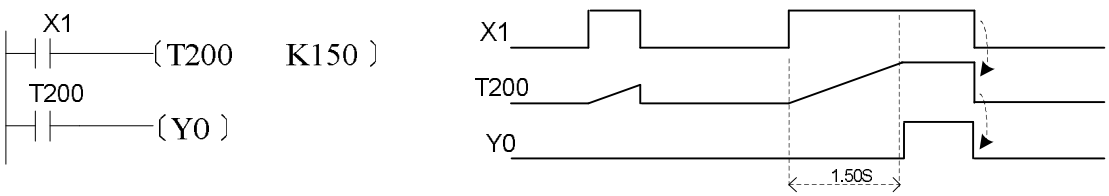
最长的情况为（$T+T_0+a$），其中：T 为设定的定时时间；T_0 为程序扫描执行时间；a 为定时器的计时步长。

最短的的情况为（T－a）。

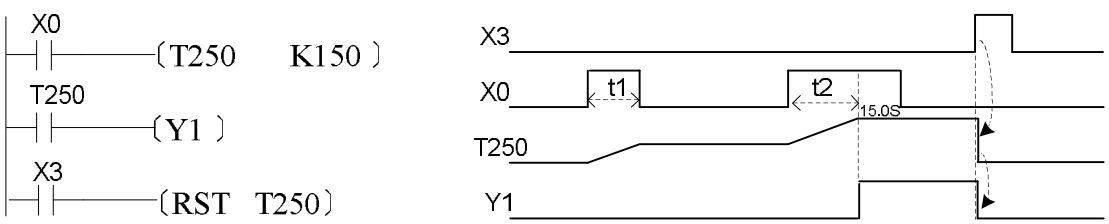
若计时器的触点指令位于线圈指令之前，最不理想的定时长度为（$T+2T_0$）。

- 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。

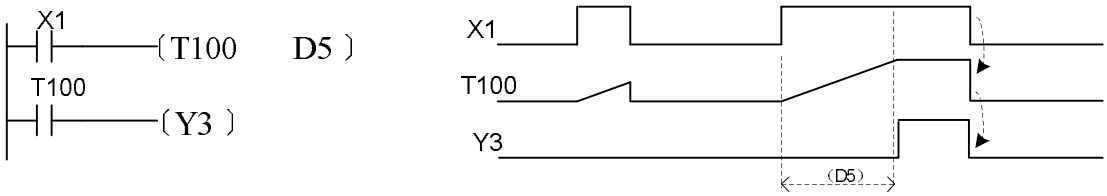
【使用举例 1】：普通计时器 T200 为 10ms 步长的计数器，实际动作延迟为 150×10ms＝1500ms，即 1.50s，动作原理为：



【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	一般用	停电保持用	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 ※2

※1 非停电保持领域。通过设定参数可变更成停电保持领域。

※2 停电保持领域。通过设定参数可变更成非停电保持领域。

※3 停电保持领域。不可通过参数的设定变更。

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：

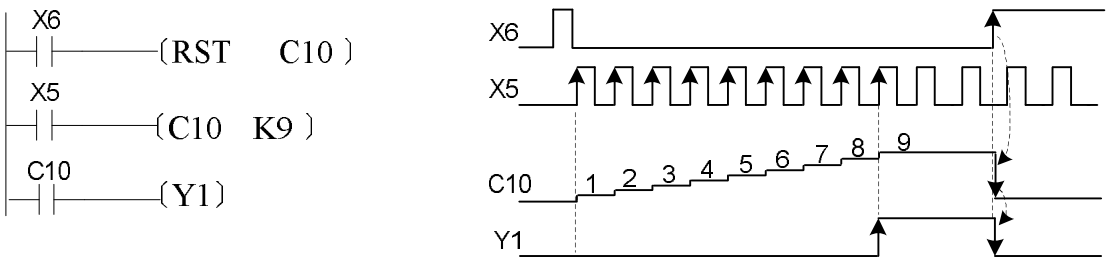
计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换
C200	M8200	C209	M8209	C218	M8218	C226	M8226
C201	M8201	C210	M8210	C219	M8219	C227	M8227
C202	M8202	C211	M8211	—	—	C228	M8228
C203	M8203	C212	M8212	C220	M8220	C229	M8229
C204	M8204	C213	M8213	C221	M8221	C230	M8230
C205	M8205	C214	M8214	C222	M8222	C231	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：

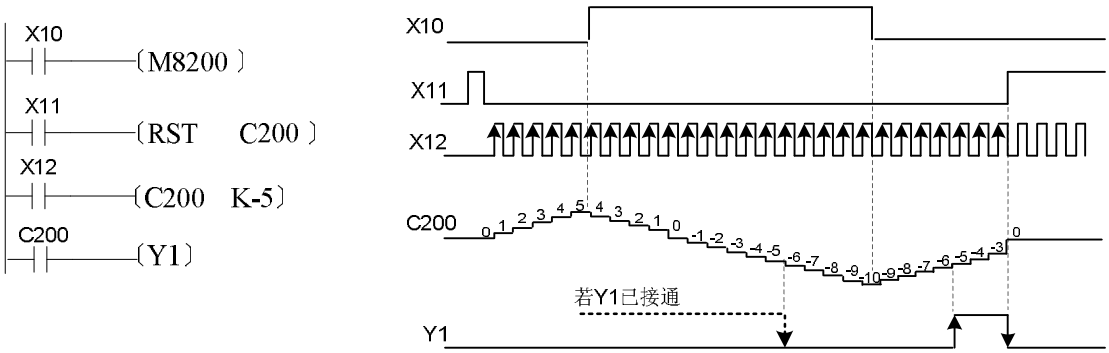


计数输入 X5 每驱动 C10 线圈一次，计数器的当前值就增加，在执行第 9 次的线圈指令时．输出触点动作。以后即使计数输入 X5 再动作，计数器的当前值不变。如果复位输入 X6 为 ON，则执行 RST 指令，计数器的当前值清为 0，输出触点复位。

- 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，因此计数器可接着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。

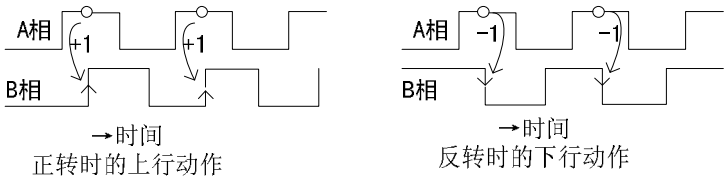
而不作为高速计数器使用的 X 输入端口可在顺控程序内作为普通的输入继电器使用。此外，不作为高速计数器使用的高速计数器编号也可作为数值存储用的 32 位数据寄存器使用。

- 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 监控
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	

分配输入	单相单计数输入										
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245
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 M8I70（脉冲捕捉），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；

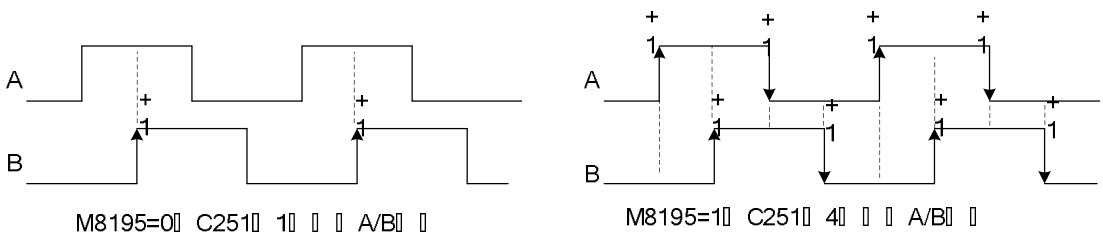
如果使用 C254 计数器，即默认使用了 X0、X1、X2、X6 输入端口，与这些端口相关的计数器、中断口或者脉冲捕捉等，便都不能再使用了。

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 C251～C255 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：



- 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 C251～C255 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存储等，每个寄存器的宽度为 16bit。若采用 32bit 指令，则自动将相邻的 2 个寄存器组成为 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

※1：非停电保持领域。通过设定参数可变更成停电保持领域。

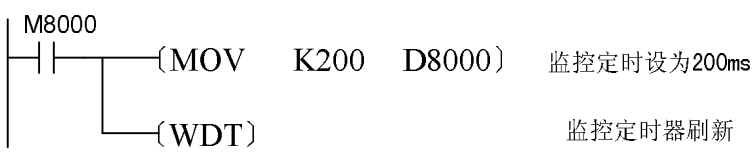
※2：停电保持领域。通过设定参数可变更成非停电保持领域。

※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）被自动占用。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/STOP 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 PLC 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。

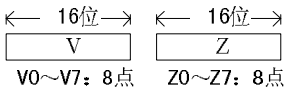
数据寄存器 V、Z

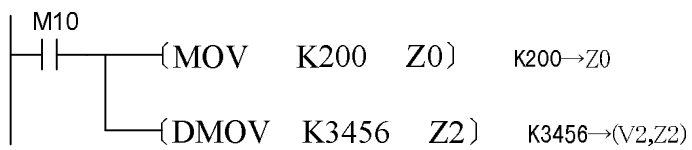
变址寄存器 V 与 Z 同普通的数据寄存器一样，是进行数值数据的读入、写出的 16 位数据寄存器。V0～V7，Z0～Z7 共有 16 个。

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。

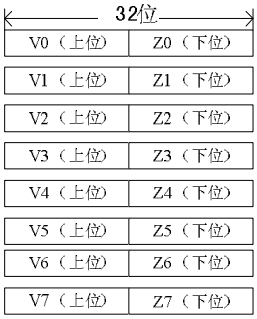
V、Z 寄存器可采用 16bit 和 32bit 方式进行访问，如下图说明：

16bit访问方式时，为独立的16个寄存器



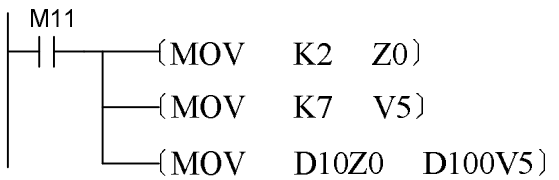


32bit访问方式时，按如下方式组合成8个寄存器



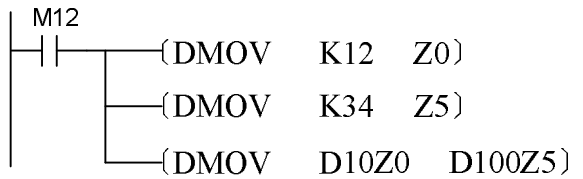
按照惯例，在处理 32 位应用指令中的软元件或处理超过 16 位范围的数值时，（为 32bit 寄存器方式），V（高位）、Z 低位）被同时访问，指定的寄存器名必须为 Z0～Z7。即使指定了 V0~V7 的高位侧，也无法进行变址。

16bit 变址应用举例：



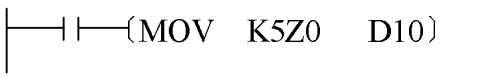
因：D10Z0=D(10+2)=D12
D100V5=D(100+7)=D107
故等效于：
(MOV D12 D107)

32bit 变址应用举例：



因：D10Z0=D[10+(V0,Z0)]=D[10+12]=D22
D100Z5=D[100+(V5,Z5)]=D[100+34]=D134
故等效于：
[DMOV (D23,D22) (D135,D134)]

常数变址的特例：



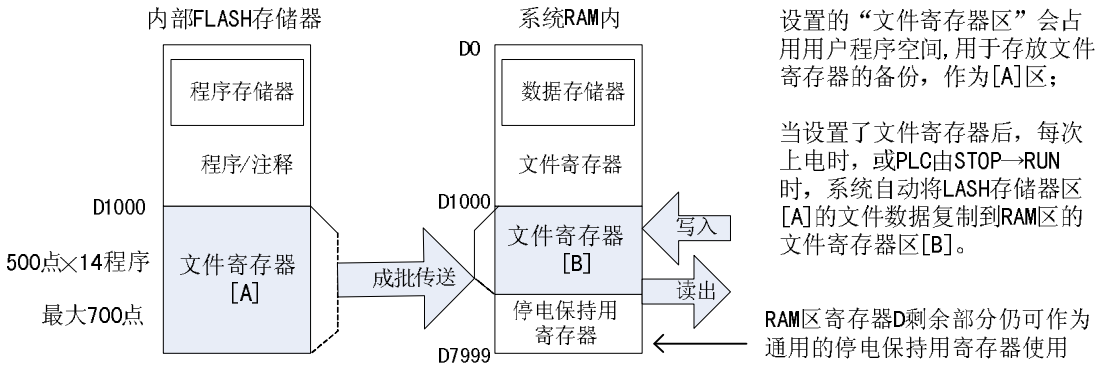
完成的操作是将Z0的值加上K5后，送给D10，Z0的值并不变化，

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。

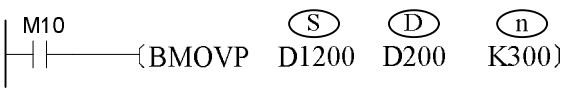
文件寄存器区域定义及处理如下示意图：



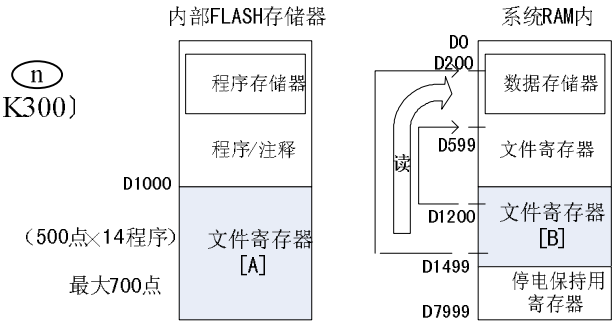
当控制器从 STOP→RUN 时内存中设定的文件寄存器区域[A]，系统程序自动将之批次传送主系统 RAM 中的数据存储区［B］部，数据存储区中已变化的内容将被初始化。此后，除块传送指令 BMOV 外，程序中对元件的操作都将是针对寄存器区域[B]中的元件。

- 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。

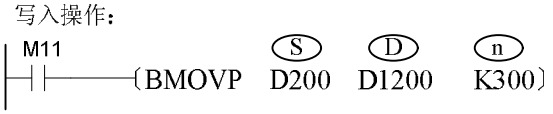
读取操作：



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.

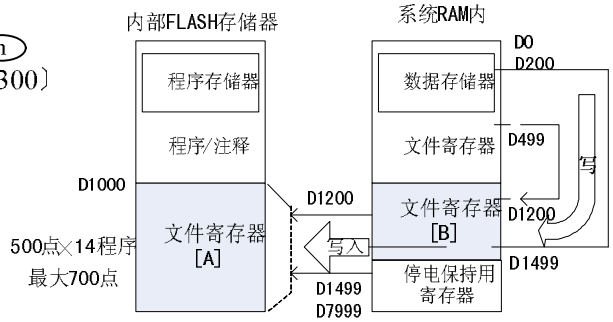


写入操作：



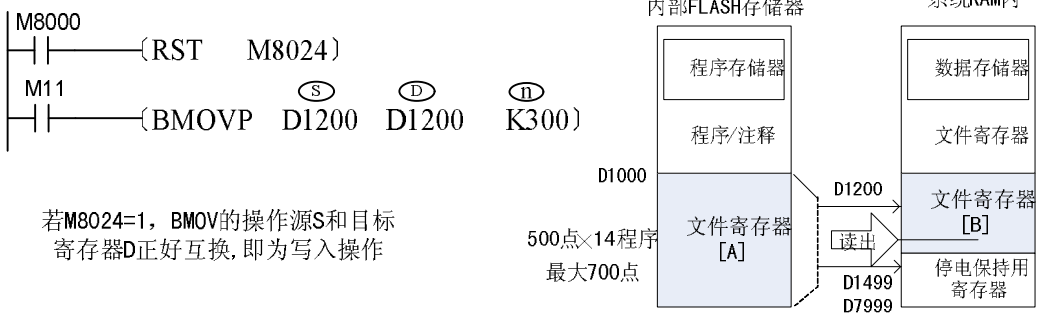
当M10=ON时，会进行如右图的操作；
首先将普通区的数据写入到文件寄存器[B]区，再复制一份到FLASH内存[A]区，确保数据掉电不会丢失。

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：



- 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

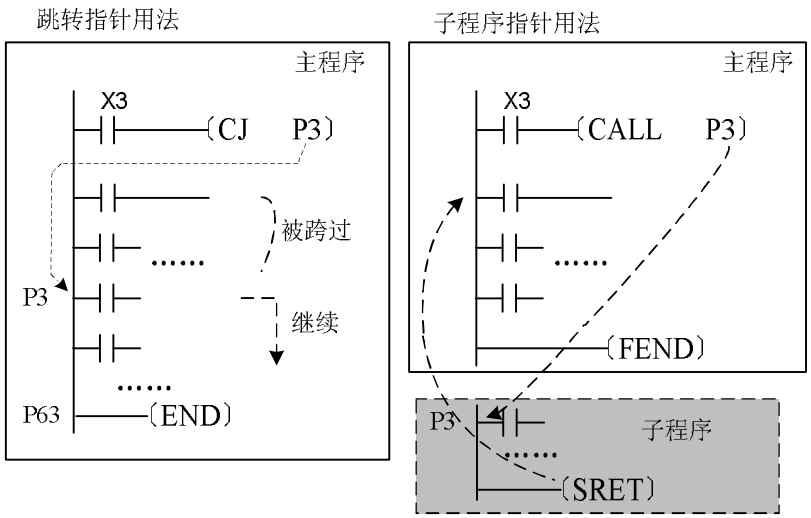
指针（P）用于跳转程序的入口地址和子程序启始地址的标识；指针（I）则用于中断程序的启始地址标识，其编号采用十进制数分配，如下表所示：

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（※注）	I600 I700 I800 共 3 点	I010 I040 I020 I050 I030 I060 共 6 点

（※注：H2U 加强功能版本的允许输入中断数有扩展，请参见附录 5.8 增强功能说明）

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：

I）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；	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		

中断允许/禁止设置			
		1=下降沿中断	
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

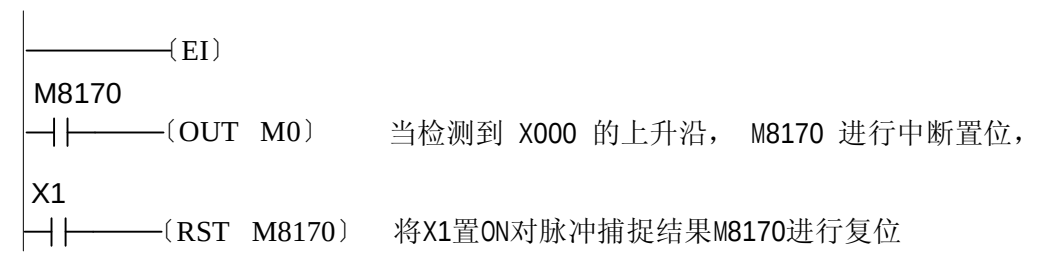
每个中断对应的“中断允许”标志开启后，还需要开启“全局中断允许”，即执行 EI 指令（FNC04）后才最后才能使能中断功能；若执行全局中断禁止 DI 指令（FNC05），则禁止所有的中断的响应。当启用了输入编号的中断允许设定标志，输入信号满足中断设定时，将执行对应的中断子程序。

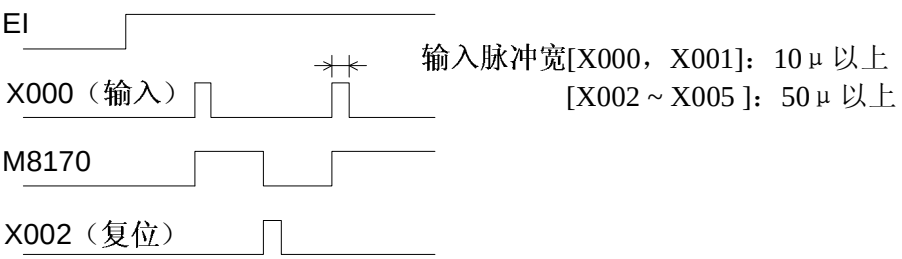
每个中断子程序的末尾均要有 IRET 指令，以表示中断子程序完毕，PLC 执行了该语句后，便会跳回本中断程序开始执行之前的位置。（AutoShop 软件中中断程序不需要写 IRET 指令）

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。

M8170~M8175 的具体使用如下：

执行 FNC04(EI)指令后，当输入继电器 X000~X005 OFF→ON 变化时，特殊辅助继电器 M8170~M8175 置位进行中断处理。为了再次获得输入，必须利用程序对设定的元件进行复位操作。脉冲捕捉动作同个别中断禁止用辅助继电器 M8050~M8055 的动作无关。





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

<table>
<tr><td>输入端口 X</td><td colspan="2">X0～X377，最大可达 256 点；
XY 总和最大 256 点</td><td colspan="2">X0～X5：具有中断功能；
X0～X7：滤波时间可设；</td><td>8 进制命名规则</td></tr>
<tr><td>输出端口 Y</td><td colspan="2">Y0～Y377，最大可达 256 点；
XY 总和最大 256 点</td><td colspan="2">晶体管型具 5 路高速脉冲输出功能，具体规格请看用户手册</td><td>8 进制命名规则</td></tr>
<tr><td>辅助继电器 M</td><td colspan="2">M0～M499
500 点，通用※I</td><td>[M500 ～ M1023]
524 点，保存用※2 继电器</td><td>[M1024 ～ M3071]
2048 点，保存用※3</td><td>M8000～M8255
256 点，特殊用</td></tr>
<tr><td>状态 S</td><td colspan="2">S0～S499 共 500 点※1
初始用 S0～S9</td><td colspan="2">[S500～S899]共 400 点，掉电保存用※2</td><td>[S900 ～ S999] 共 100 点，报警用※2</td></tr>
<tr><td>定时器 T</td><td colspan="2">T0～T199 共 200 点，100ms。
子程序用：T192～T199</td><td>T200～T245
共 46 点，10ms</td><td>[T246 ～ T249]
共 4 点，1ms 累计※3</td><td>[T250～T255]共 6 点，100ms 累计※3</td></tr>
<tr><td>16 位向上计数器 C</td><td colspan="3">C0～C99100 点，通用※1</td><td colspan="2">[C100～C199]100 点，保存用※2</td></tr>
<tr><td rowspan="2">32 位计数器 C</td><td colspan="2">32 位可逆</td><td colspan="3">32 位高速计数可逆</td></tr>
<tr><td>C200 ～ C219</td><td>[C220 ～ C234]</td><td>[C235～C245]
单相单计数输</td><td>[C246～C250]单相双计数输入※2</td><td>[C251～C255]
2 相计数输入</td></tr>
</table>

	20 点，通用※1	15 点，掉电保存用※2	入※2		※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，输人中断指针	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

指　令

第四章 指　令

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 等	1

4.1 STL/SFC 指令

4.1.1 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

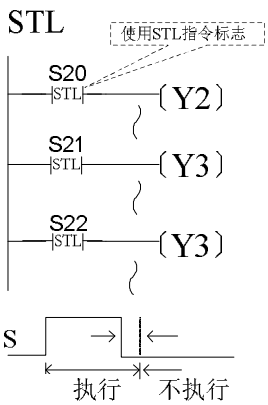
步进梯形图指令（STL，RET）

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：

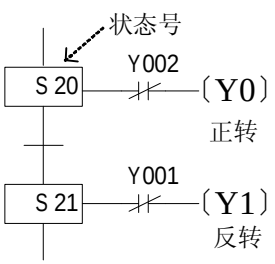
- 步进梯形图程序以 STL 指令开始（注意与普通梯形图中 S 不同），以 RET 指令结束，中间的程序以 S 状态引导，后续该 S 状态的所有操作逻辑，包括条件满足时切换为下一状态的操作。



●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）。

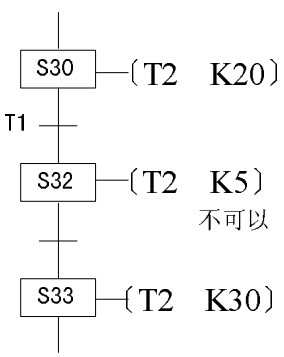
●在不同的状态 S ，可 对 同样的输出软元件（ 如 Y3）。此时，S21或S22接通时，Y3 被输出。但在同一S状态中同样存在“双线圈”处理问题，请注意。

●状态S号编号不可重复使用。



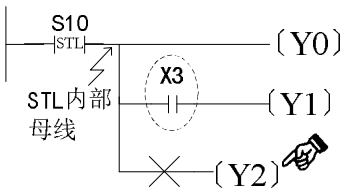
输出的互锁问题：

●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。



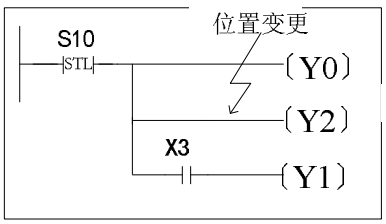
定时器重复使用的问题：

●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。

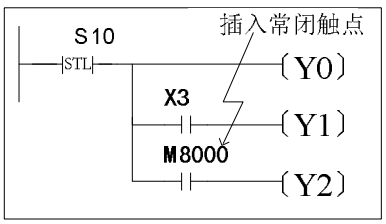


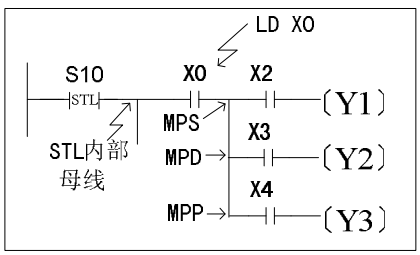
输出驱动方法

●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。



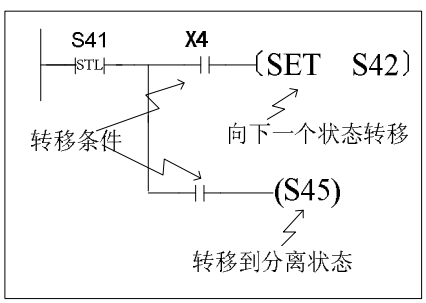
或





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。

4.1.2 SFC顺序功能图编程

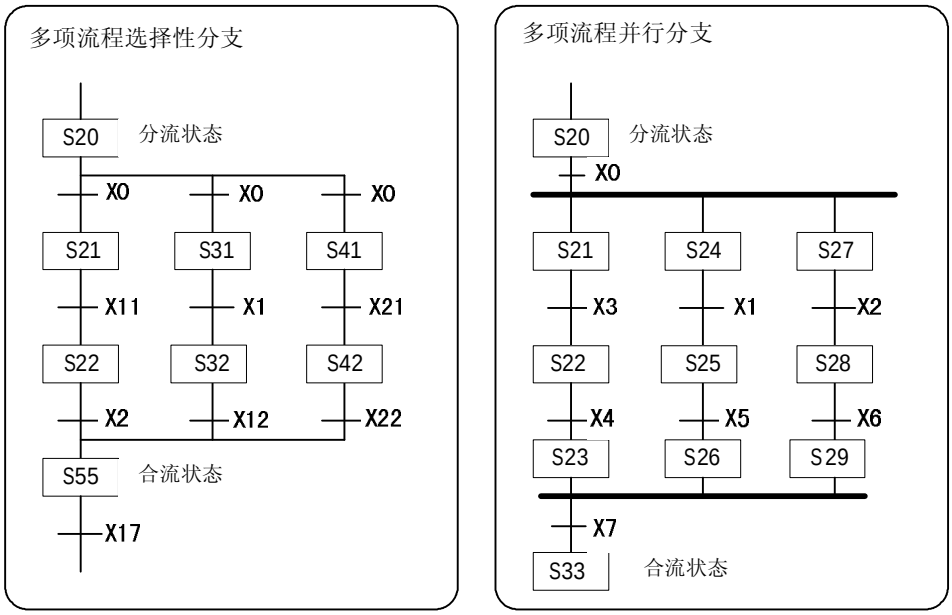
H2U 可编程控制器内置有利用 SFC 图（顺序功能图）的顺控功能，SFC 采用类似流程图的方式，将控制程序按流程图方式直观表述，使得编程调试、维护的大为简化。SFC 图设计时使用的符号定义如下：

	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。

4.1.2.1 SFC 的编程特点：

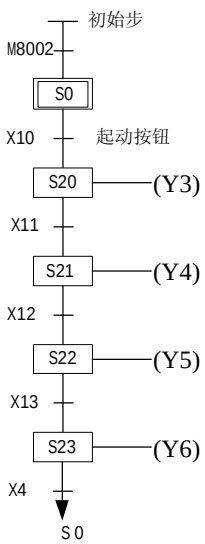
- 在该梯形图块 LAD0 中，采用可编程控制器由 STOP→RUN 转换时，瞬间动作的辅助继电器 M8002，使初始状态 S0 置位（ON）；
- 可编程控制器中 S0-S9 为初始状态软元件；

- 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。



4.1.2.2 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。

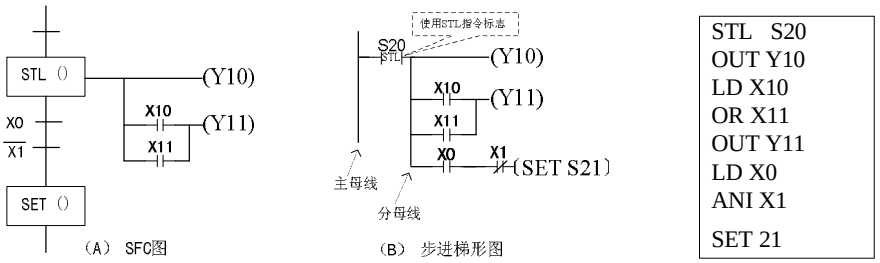
初始状态的作用

- 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。

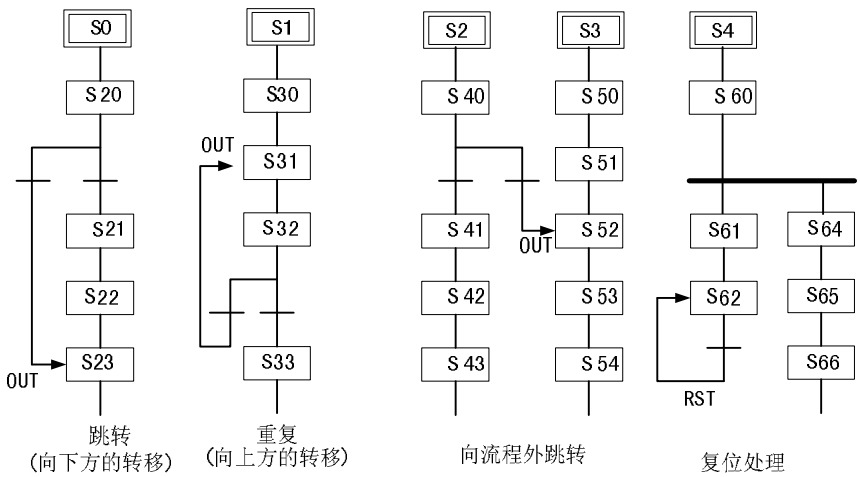
程序用 SFC 图或用步进梯形图均可编写。编程顺序为先进行负载的驱动处理，接着进行转移处理。当然，如果是不需要驱动负载的状态，则不需要进行负载的驱动处理。



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

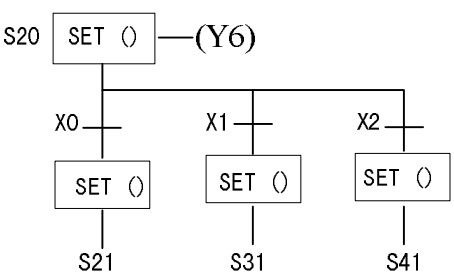


如上图所示，向下方状态的转移（跳转）、向上方状态的转移（重复）、向流程外的转移等的分离状态转移，用 OUT 指令编程。

选择性分支与汇合状态

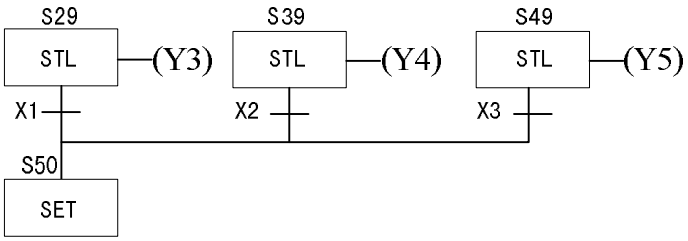
和一般状态的处理相同顺序，首先进行驱动输出处理，然后再进行状态转移处理。

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
```

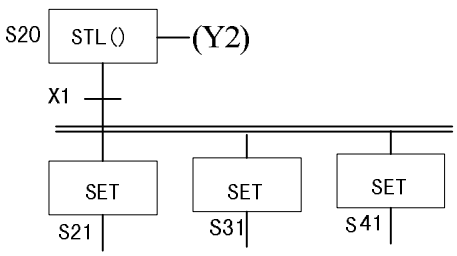
上图例为分支汇合的典型例子，右侧为等效指令表。

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。

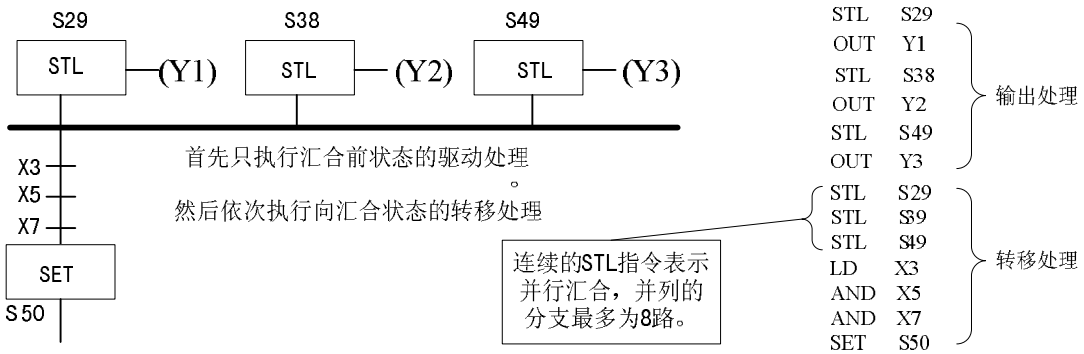
并行分支与汇合状态



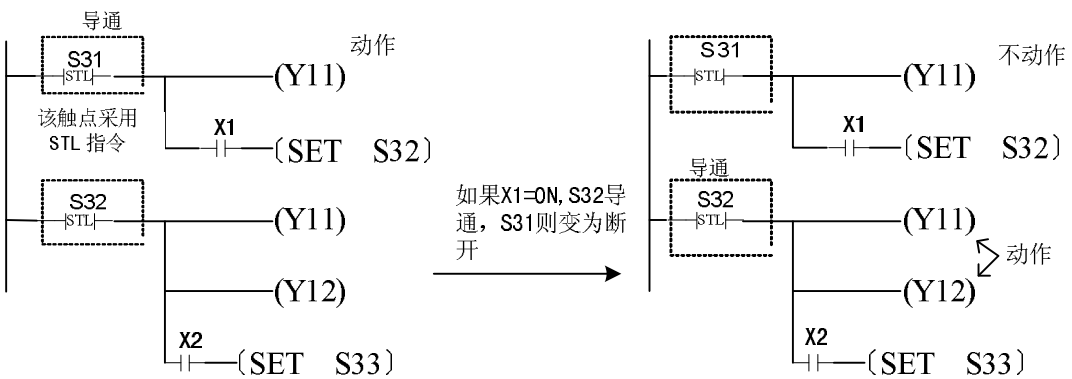
和一般状态的处理相同顺序，首先进行驱动输出处理，然后再进行状态转移处理。

等效指令列表如右：

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

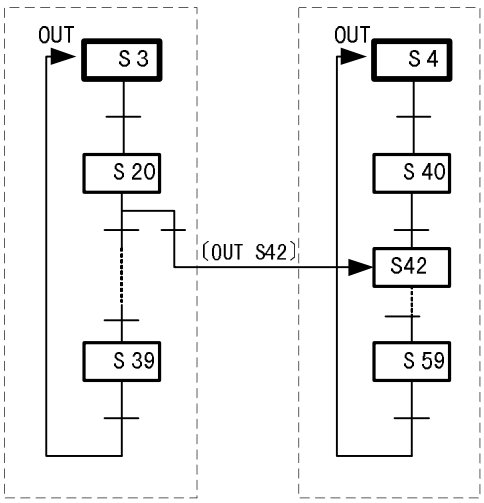


步进梯形图指令及其动作如下图所示：



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。



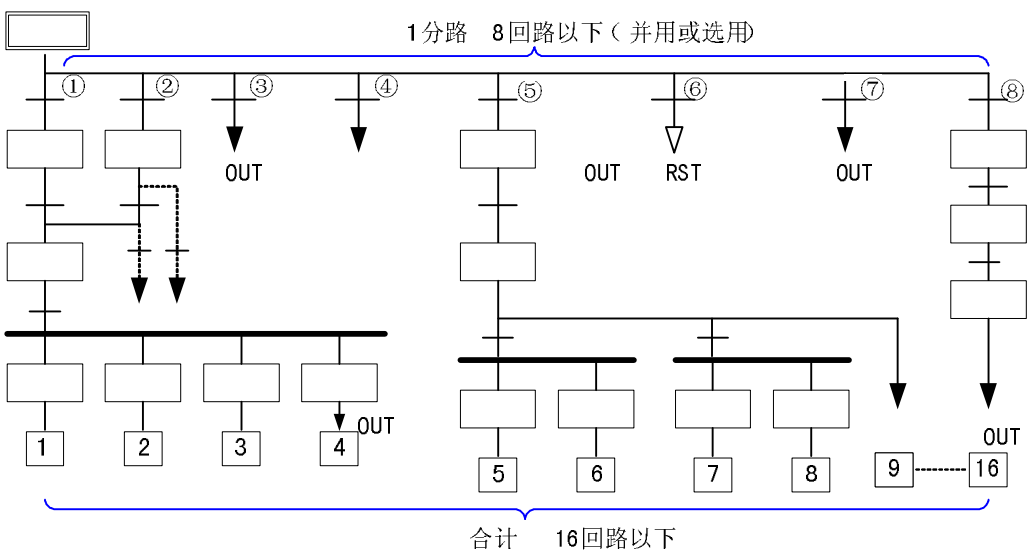
如左图所示，初始状态S3对应 其 STL指令的程序，而初始状态S4 则对应另一程序；

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。

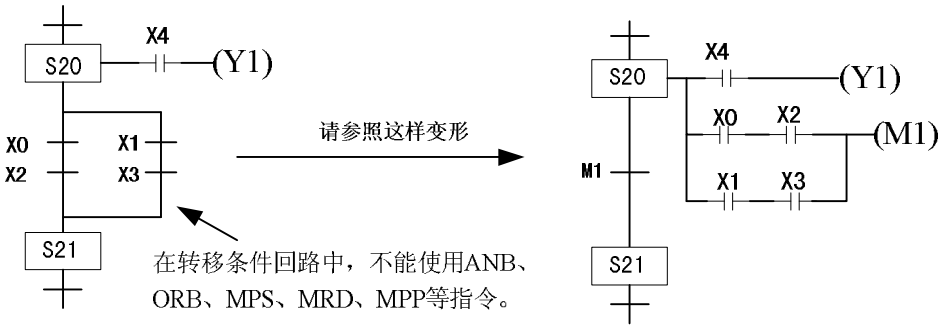
重要的是不可混杂STL指令。

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：



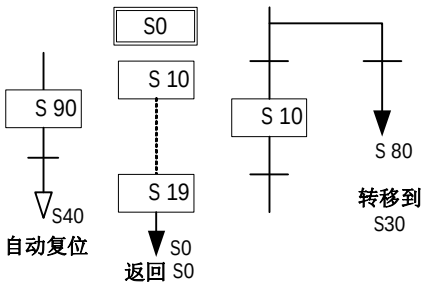
不能进行从汇合线或汇合前的状态开始向分离状态的转移处理或复位处理，应设置空状态，由分支线上向分离状态进行转移与复位处理。

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：

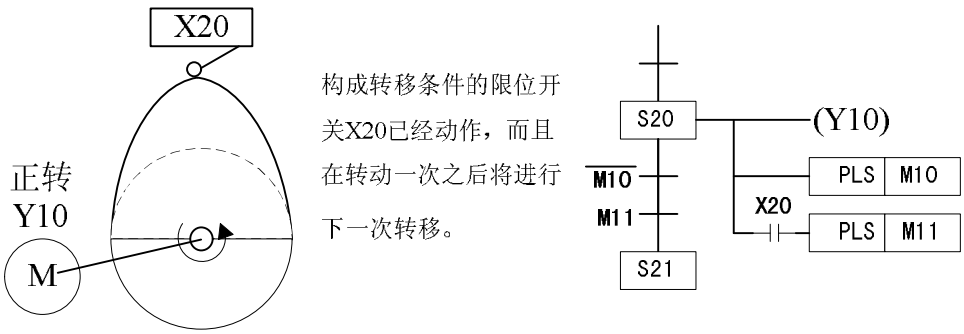
- 在流程中，符号⬇则表示向上面的状态转移重复或向下面的状态转移（跳转），或者向分离的其他流程上的状态转移。符号▽则表示状态的复位处理。
- 状态标志 S 也可以采用 ZRST 指令对一个区间的标志进行批量复位。



- 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。

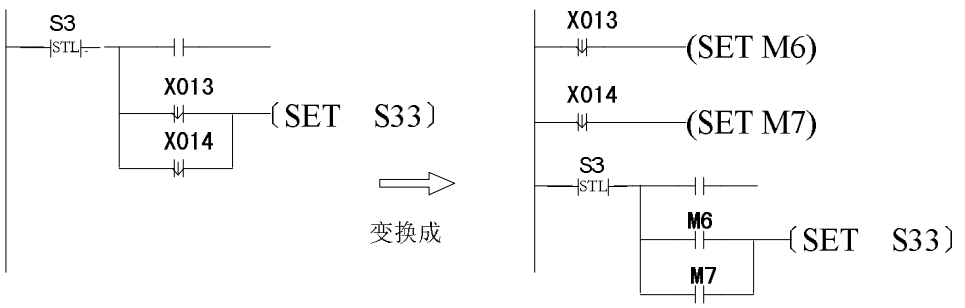


如上图所示的应用中，将转移条件脉冲化，S20 首次动作，通过 M10 使不产生转移。

- 上升沿 / 下降沿检测触点使用时的注意事项：

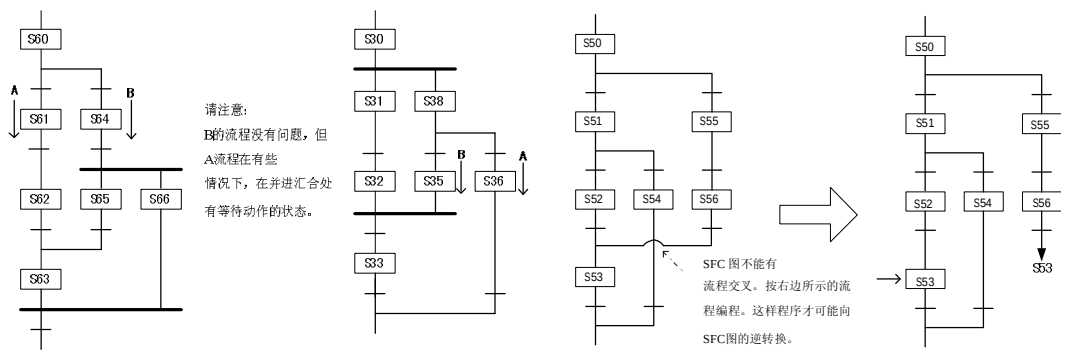
在状态内使用 LDP、LDF、ANDP、ANF、ORP、ORF 的上升沿 / 下降沿检侧触点时. 状态断开时变化的触点，在状态再次接通时被检出。

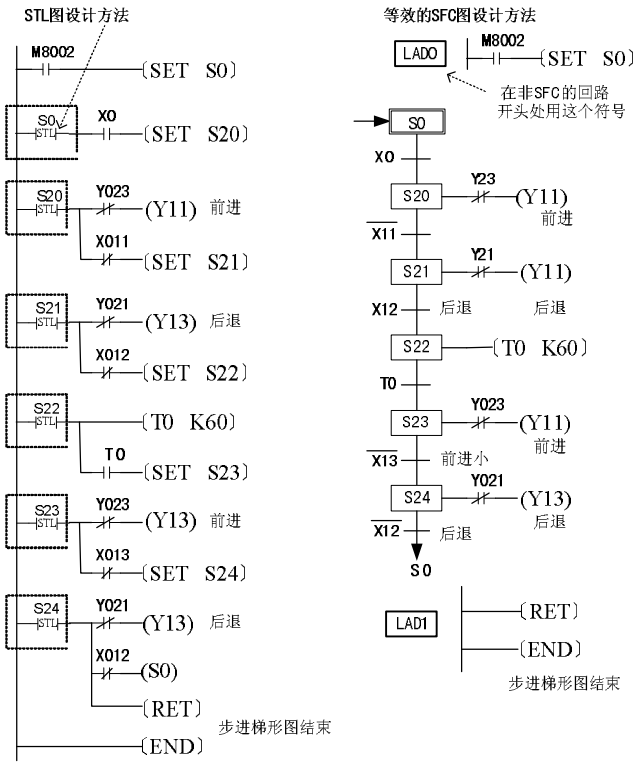
对于状态断开时变化的条件，必需上升沿 / 下降沿检侧时，请按下图所示，修改程序：



通过 X0l3 下降沿向 S33 转移后，若 X0l4 下降. 此时因 S3 断开，X0l4 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




STL图设计方法
M8002
〔SET S0〕
S0
STL
X0
〔SET S20〕
S20
STL
Y023
(Y11) 前进
X011
〔SET S21〕
S21
STL
Y021
(Y13) 后退
X012
〔SET S22〕
S22
STL
〔T0 K60〕
T0
〔SET S23〕
S23
STL
Y023
(Y13) 前进
X013
〔SET S24〕
S24
STL
Y021
(Y13) 后退
X012
(S0)
〔RET〕
〔END〕
步进梯形图结束
等效的SFC图设计方法
LAD0
M8002
〔SET S0〕
在非SFC的回路
开头处用这个符号
S0
X0
S20
Y23
(Y11)
前进
X11
S21
Y21
(Y11)
X12
后退
后退
S22
〔T0 K60〕
T0
S23
Y023
(Y11)
前进
X13
前进小
S24
Y021
(Y13)
后退
X12
后退
S0
LAD1
〔RET〕
〔END〕
步进梯形图结束

4.2 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	FNCNO.	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
	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 I
	25	DEC	✔	✔	BIN 减 I
	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

分类	FNCNO.	指令符号			指令功能
		16bit	32bit	P指令	
	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
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

分类	FNCNO.	指令符号			指令功能
		16bit	32bit	P指令	
	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
	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

分类	FNCNO.	指令符号			指令功能
		16bit	32bit	P指令	
	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)

4.3 指令解释

4.3.1 基本指令解释

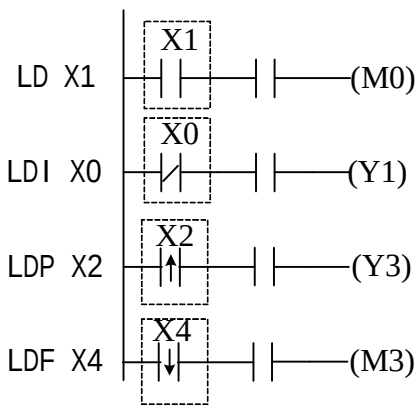
指令	操 作 数						
LD LDI	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
LDP LDF	✔	✔	✔	✔	✔	✔	

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。



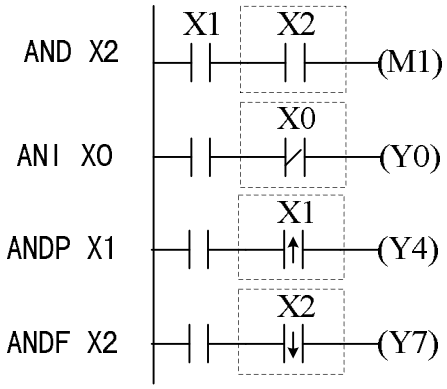
指令	操 作 数						
AND ANI	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
ANDP ANDF	✔	✔	✔	✔	✔	✔	

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；



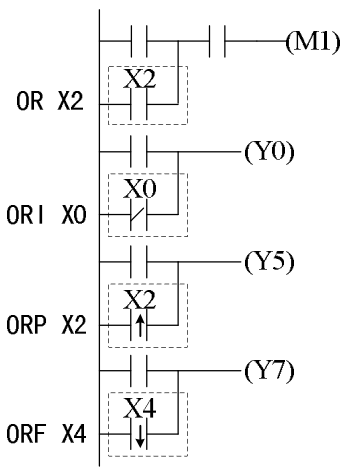
指令	操 作 数						
OR ORI	X0~ X377	Y0~ Y377	M0~M3071 M8000~M8255	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
ORP ORF	✔	✔	✔	✔	✔	✔	

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；

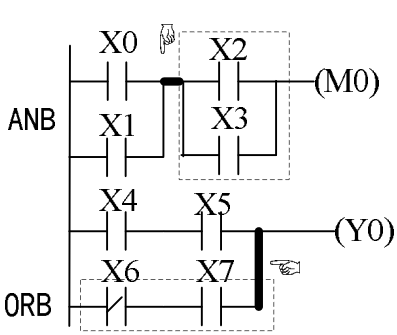
ORP指令是将接点的上升沿跳变状态参与OR运算；

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	无。
ORB	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
```

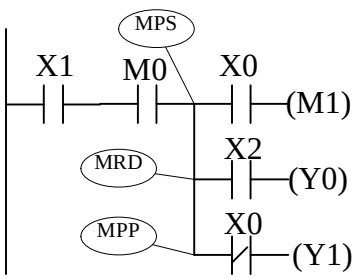
```
☛LD X4
  AND X5
☛LDI  X6
  AND  X7
☞ORB
  OUT Y0
```

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）

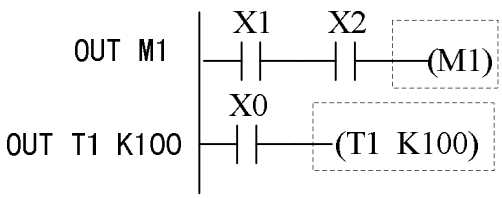



```
LD X1
AND M0
MPS
AND X0
OUT M1
MRD
```

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

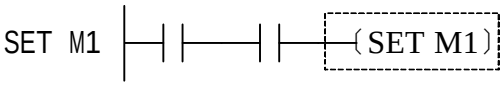
指令	操 作 数						
OUT	**X0~ X377**	**Y0~ Y377**	**M0~M3071 M8000~M8255**	**S0~ S999**	**T0~ T255**	**C0~ C255**	**D0~ D8255**
		✔	✔	✔	✔	✔	

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
		✔	✔	✔			

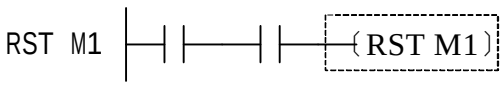
当SET 指令被驱动，其指定的组件被设定为ON，且被设定的组件会维持ON，不管SET 指令是否仍被驱动。可利用RST指令将该组件设为OFF。



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。



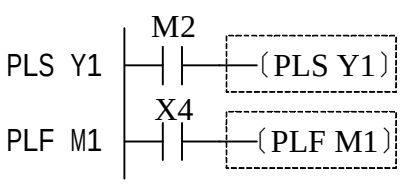
元件	操作结果
S，M，Y	线圈及接点被设定为OFF
T，C	目前计时或计数值会被设为0，且线圈及接点被设定为OFF。
D，V，Z	元件的值清为0

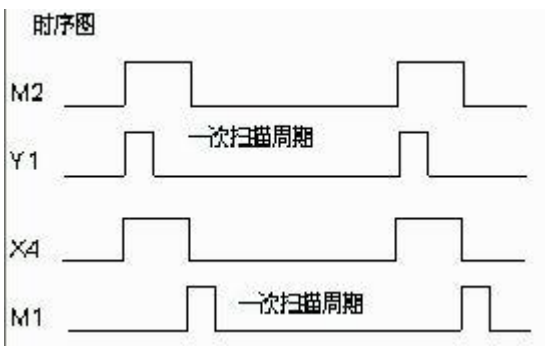
指令	操 作 数						
PLS PLF	X0~ X377	Y0~ Y377	M0~M3071	S0~ S999	T0~ T255	C0~ C255	D0~ D8255
		✔	✔				

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。

指令举例：





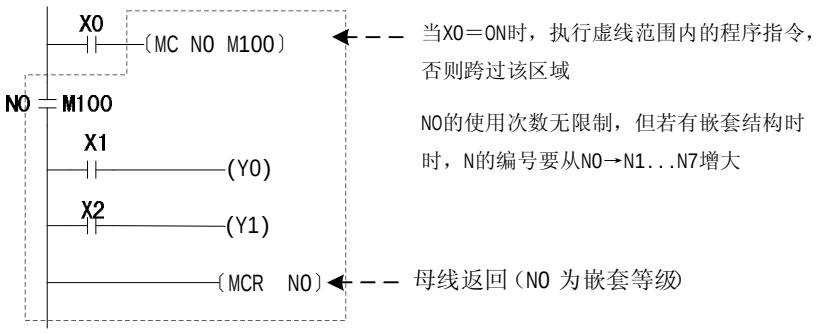
指令	操 作 数
MC MCR	N0~N7

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

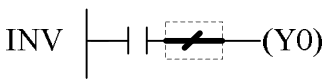
MCR为主控结束指令，置于主控程序最后，在MCR指令之前不可有接点指令。

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。

只在梯形图程序或指令程序最后才加入END指令。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，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

指令	操 作 数
	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。 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用于跳转指令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。

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。

4.3.2 应用指令解释

4.3.2.1 程序流程（00~09）

16bit	32bit	P	FNC00	CJ P*** CJP P***	条件跳转
✔		✔			
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。

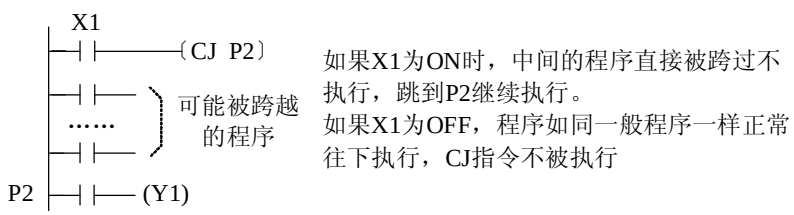
若被跨越的中间地址区的程序中有 TMR 定时器或计数器，且已被驱动，则动作反应为：

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	

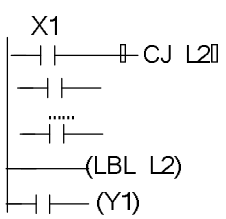
对P***地址指针的要求如下：

- 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；
- 当使用者希望某一部份程序不需要执行时，以缩短扫描周期；或者想使用两个线圈输出时，为避免双线圈的出现。可使用此指令；
- CJ 指令可重复指定同一指针 P，但不可与 CALL 指定同一指针 P，否则出错。

指令举例：



在AutoShop编程环境中，跳转指令用L，上例中格式如下



且子程序和中断程序单独窗口写，故不必注意FEND等事项，CJP63在AutoShop编程环境中为CJEND。

16bit	32bit	P	FNC01	CALL P*** CALLP P***	子程序调用
✔		✔			
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；
- 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~T249 作为定时器。

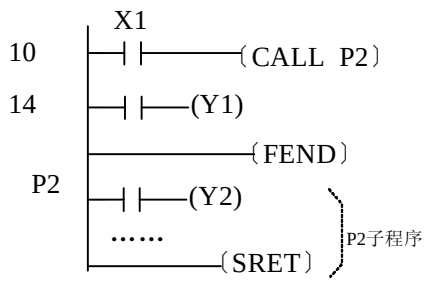
在 AutoShop 编程环境中子程序在单独窗口编写，故没有 FEND、SRET 等指令的问题存在，且子程序名可任意修改（包括中文）。

16bit	32bit	P	FNC02	SRET	子程序完毕。
✔					
1 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。

在AutoShop编程环境中，子程序最后无需编写SRET。

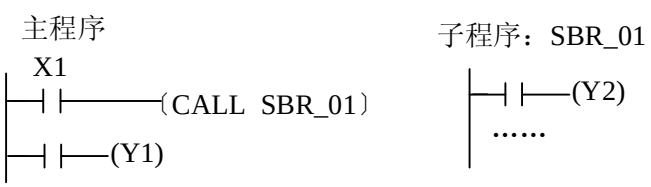
指令举例一：



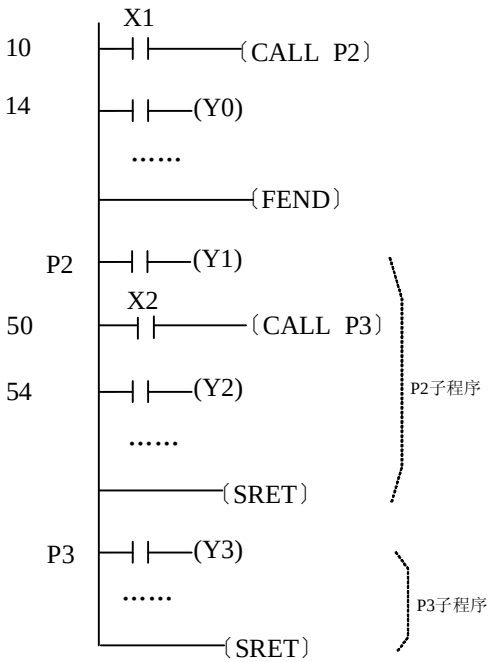
当X1为ON时，执行CALL调用指令，跳到P2指定的子程序执行扫描，当扫描到SRET指令时，返回到CALL指令的下一指令，即地址14继续往下执行.
当X1为OFF时，CALL指令不被执行，不调用P2子程序。

在AutoShop编程环境中“指令举例一”的格式如下：

右击程序块的子程序SBR_01，选择属性。可以修改01为你想要的名称，包括中文）



指令举例二：

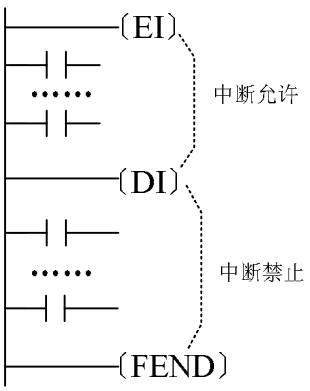


当X1为ON时，执行CALL调用指令，跳到P2指定的子程序执行扫描，分2种情况：如果扫描到地址50时X2为OFF，则继续往下扫描直到SRET指令，返回到主程序的地址14继续往下执行；如果扫描到地址50时X2为ON，则执行CALL P3，转移到P3所指定的子程序进行扫描，当执行到SRET指令时，回到P2子程序地址54继续往下执行

16bit	32bit	P	FNC03	IRET	中断程序完毕	无需触点驱动，无操作数的单独指令。
✔			FNC04	EI	中断允许	
1 步			FNC05	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。在AutoShop编程环境中，中断程序在单独窗口编写，在中断程序最后无需写IRET指令

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。在程序中如果没有中断插入禁止的区间时，可以不使用DI指令。



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1ms～99ms的固定周期发生的中断；

3） 高速计数中断：与FNC53（DHSCS）比较置位指令并用，当高速计数的当前值达到设定值时，产生中断。

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

定时中断指针与设置：

输入编号	中断周期 MS	中断禁止指令
I6□□	在指令的□□中输入1~99的数，如I605为每5MS执行一次定时中断	M8056
I7□□		M8057
I8□□		M8058

高速计数中断指针与设置：

输入编号	中断禁止指令
I010	M8059
I020	
I030	
I040	
I050	
I060	

脉冲输出完成中断指针与设置：（此功能要启动M8090~M8094才能在脉冲输出完毕产生中断）。

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

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；

外部输入中断中对于同一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，触发沿通过指针编号来设定；

外部输入中断：如果对M8050-M8055在程序执行过程中“ON”，则禁止了对应X端口的中断功能。

定时中断：如果对M8056-M8058在程序执行过程中“ON”，则禁止了对应的定时中断功能。

高速计数中断：如果对M8059在程序执行过程中“ON”，则禁止了所有的高速计数中断功能。

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，优先级别高的为优先。

优先级从高到低分别为高速计数器中断、外部中断、时间中断、脉冲输出完成中断。

- 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 1ms 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)。

为了满足在高速计速器运行时，我们另外新增加了 30 个高速计数中断，可以指定任意一个高速中断源产生 30 个中断响应，此功能命名为高速计数器多用户中断。使用设置有如下规律；

标志位	描述
M8084	为 ON 使能高速计数器多用户中断
D8084	为高速计数器序号 C235~C255
D8085	对应的用户中断个数，最大 24 个，从 I507~I530
D8086	对应多个比较点数据的序号，只能为 D 元件，如 200 为 D200 开始的双字

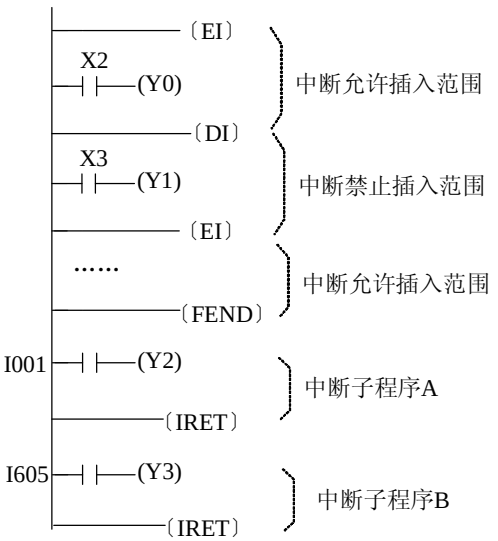
比较点数据存放例子：

D8084=235；D8086=200；D8085=5；M8084=ON；

C235 的数据	记录单元	存放单元数值	对应的用户中断	D8131 的数值
100	D200，D201	=100	I507	0
200	D202，D203	=200	I508	1
300	D204，D205	=300	I509	2
400	D206，D207	=400	I510	3
500	D208，D209	=500	I511	4 0(M8133=ON)

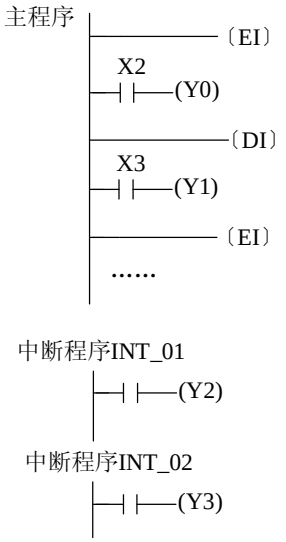
每个中断可以由高速计数器的数值和记录单元的数值产生。

指令举例：



PLC 执行时，当程序扫描到EI 指令到DI 指令间，X0=On 或者定时时间5MS到时，则执行中断插入子程序A或B，而当子程序执行至IRET时，则返回主程序并继续往下执行。

在AutoShop中左图的程序如下：（右击程序块的中断程序INT_01或者INT_02，选择属性。可以修改01或者02为你想要的名称，包括中文）



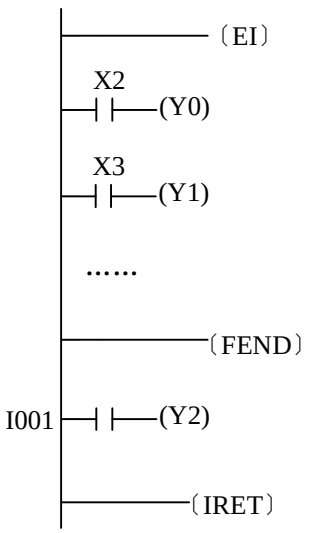
分别右击程序块的中断程序INT_01和INT_02，选择属性-中断事项：分别设为I001和I605

16bit	32bit	P	FNC06	FEND	主程序完毕。
✔					
1 步				无需触点驱动，无操作数的单独指令。	

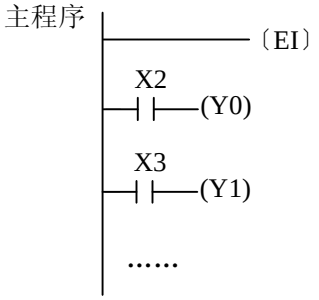
FEND语句位于主程序的结束处，执行了该指令后，即结束了本次用户程序的扫描，向0步程序返回，再次从头对程序进行扫描。

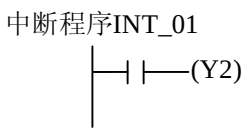
CALL命令调用的子程序必须写在FEND命令后，并且在该子程序结束加上SRET命令；中断子程序也必须写在FEND之后，并在该中断程序结束加上IRET指令，在AutoShop中，子程序或中断程序必须在单独窗口编写，在主程序末尾无需加FEND，在子程序或中断程序的末尾无需加SRET或IRET。

指令举例：



在AutoShop中左图的程序如下：（右击程序块的中断程序INT_01，选择属性。可以修改01为你想要的名称，包括中文）

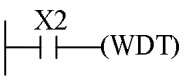




右击程序块的中断程序INT_01，选择属性-中断事项：设为I001

16bit	32bit	P	FNC07	WDT	监视定时器复位。
✔		✔			
1 步				无操作数的单独指令。	

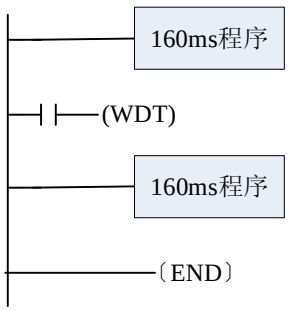
PLC系统内有用于监视用户程序执行一次的时间是否超时的定时器，若超时即会停止用户程序的执行并报警，执行WDT指令即可将该监视定时器复位，让监视定时器重新开始计时，避免超时错误。



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（例如在FOR~NEXT指令之间中插入该指令）；

如果程序扫描时间大于D8000的值（默认200ms），可以在程序间插入WDT指令将每段程序分成扫描时间低于200ms或者根据需要修改D8000的设定值。

指令举例：



此程序扫描时间是320ms，用WDT指令将程序分割为2部分，使得每部分程序扫描时间都在200ms以下。

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)			

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)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

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。

参见NEXT指令的解释与举例。

16bit	32bit	P	FNC09	NEXT	循环范围结束	无需触点驱动，无操作数的单独指令。
✔						
1 步			指令格式：NEXT(无操作数)			

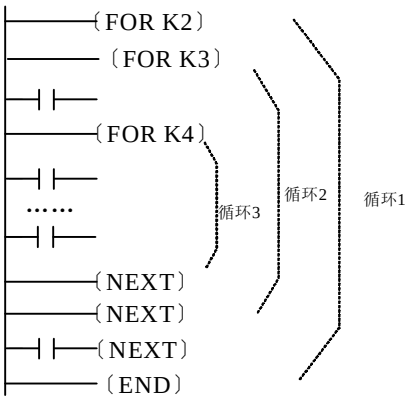
NEXT指令用于指示循环区域的尾部。由FOR指令指定FOR~NEXT循环来回执行N次后跳出FOR~NEXT循环往下继续执行。

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。但需要注意当循环次数过多时，会使PLC扫描周期延长，可能造成逾时监视定时器动作而导致错误产生。可在FOR~NEXT指令之间使用WDT指令来改善。

有下列情况者，都会出错：

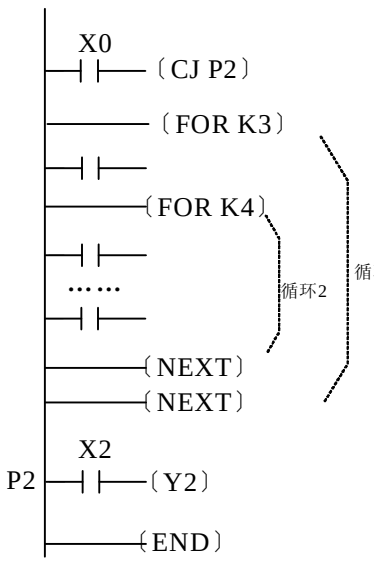
- NEXT 指令在 FOR 指令之前；
- 有 FOR 指令而无 NEXT 指令；
- 在 FEND，END 指令以后有 NEXT 指令；
- FOR 指令与 NEXT 指令个数不一致等。

指令举例一：



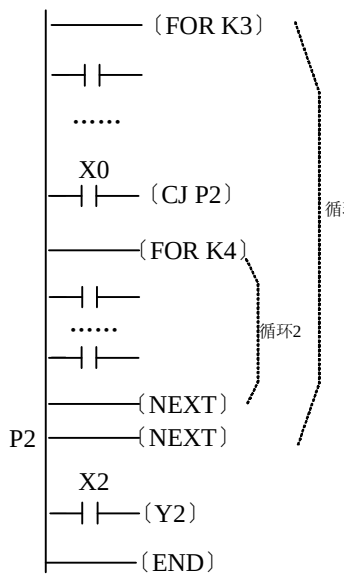
循环1执行2次后在到NEXT指令以后的程序继续执行，而循环1每执行一次循环2会执行3次，而循环2每执行一次循环3又执行4次，所以循环3共执行2*3*4=24次，循环2共执行2*3=6次

指令举例二：



想跳过FOR~NEXT指令时，可用CJ跳转指令实现,范例中当X0为OFF时，执行循环1和循环2，当X0为ON时，CJ指令跳转至P2处，循环1和循环2之间的程序不被执行

指令举例三



想跳过循环内嵌套的FOR~NEXT指令时，也可用CJ跳转指令实现,范例中当X0为OFF时，执行循环1内的循环2，当X0为ON时，CJ指令跳转至P2处，循环1内嵌套的循环2FOR~NEXT被CJ指令跳过不执行。

4.3.2.2 传送与比较（10~19）

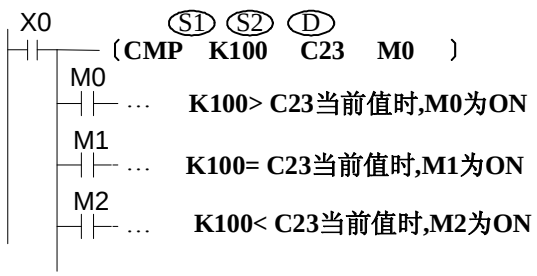
16bit	32bit	P	FNC	CMP	两个数比较
✔	✔	✔	10		
7 步	13 步		指令格式：CMP (S1) (S2) (D)		

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

本指令完成对两个操作变量的大小作比较，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。

其中 (D) 会占用 3 个连续地址的位变量。

指令举例：



当X0=ON时，M0~M2其中之一会ON。
X0由ON变OFF 时，不执行CMP指令，M0~M2仍保持X0 =OFF之前的状态,若要清除M0~M2的比较结果可用RST或者ZRST对M0~M2进行清除。
若需要得到≧、≦、≠的结果时，可将M0~M2串并联即可取得。

16bit	32bit	P	FNC	ZCP	区间比较
✔	✔	✔	11		
9 步	17 步		指令格式：ZCP (S1) (S2) (S) (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
(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。

其中：

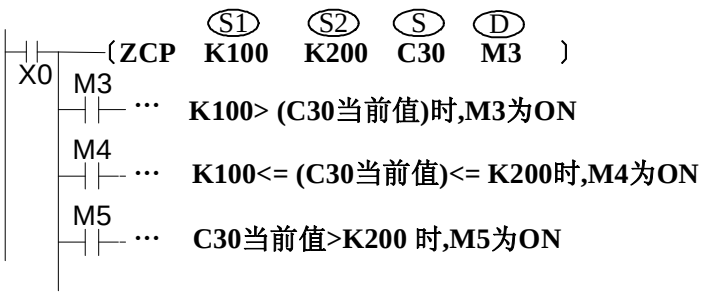
(S1) 为比较区间的下限；

(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。

指令举例：



当X0=ON时，M3~M5其中之一会ON。
X0由ON变OFF 时，不执行ZCP指令，M3~M5仍保持
X0= OFF之前的状态,若要清除M3~M5的比较结果
可用RST或者ZRST对M3~M5进行清除。

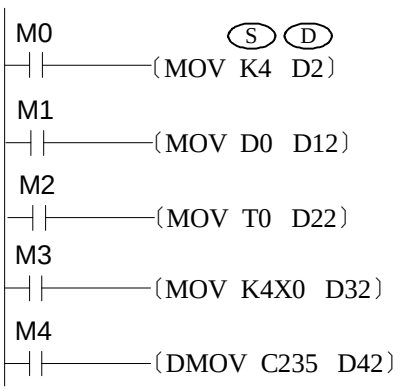
16bit	32bit	P	FNC	MOV	数据移动
✔	✔	✔	12		
5 步	9 步		指令格式：MOV (S) (D)		

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)								✔	✔	✔	✔	✔	✔	✔	✔

需要触点驱动，有 2 个操作变量，将(S)的值复制到(D)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：〔DMOVD1D5〕的操作结果是：D1→D5；D2→D6

指令举例：



当M0为ON时，将K4复制到D2中，当M0由ON ⟹ OFF时，D2保存K4的内容不变，除非用户程序再次将D2的值修改。或者PLC由STOP ⟹ RUN和PLC重新上电，D2的值才会变为0，停电保持寄存器上电或者由停止到运行保持原来的值不变

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

需要触点驱动，最多有 5 个操作变量，其中：

(S) 为待复制的数据源变量；

(m1) 为数据源传送的起始位号，（1～4）范围；

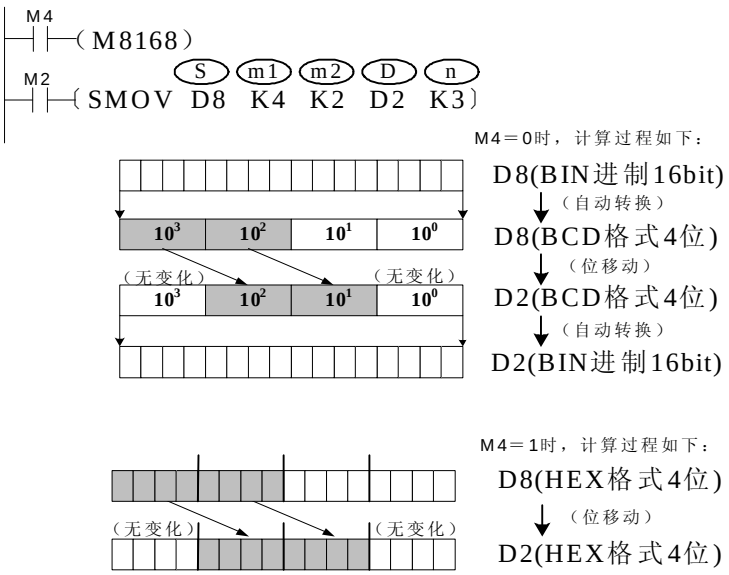
(m2) 为数据源传送的位数，（1～m1）范围；

(D) 为数据源传送的目的变量；

(n) 为数据源传送的目的变量的起始位，（m2～4）范围。

数据位的传送过程与特殊标志M8168的状态有关，当M8168为OFF时是BCD模式（十进制的位），当M8168为ON时是BIN模式，在BIN模式下以4个位作为一个单位作传送（十六进制的位）.

指令举例：



假设D8=K1234，D2=K5678，，则当M8168为OFF时（BCD模式），将M2置ON，则D2的值变为K5128；

当M8168为ON时（BIN模式），此时D8=H04D2=K1234，D2=H162E=K5678，将M2置ON，则D2=H104E=K4174。

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

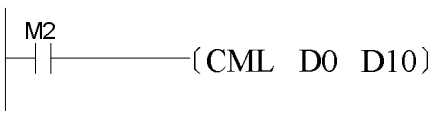
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

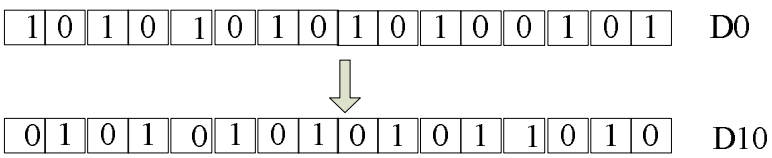
需要触点驱动，有 2 个操作变量，将(S)的 BIN 值逐位取反后复制到(D)中。

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；

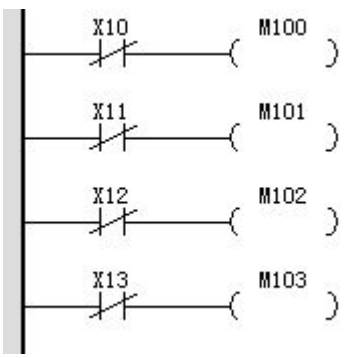
当为32bit指令（**DCML**）时，(S)和(D)都会使用相邻高地址的变量单元参与运算。例如语句：〔**DCML D1 D5**〕的操作结果是：**/D1→D5；/D2→D6**

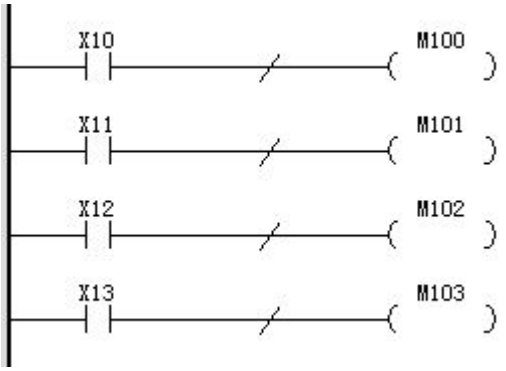
指令举例一：





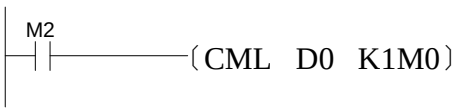
指令举例二：

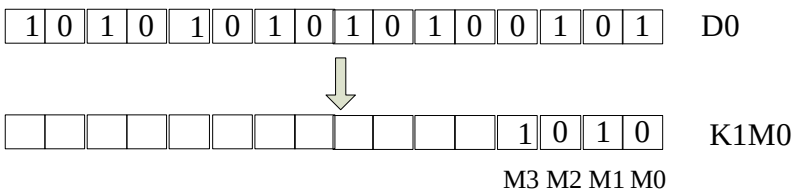




上面2个程序可以用下面的CML指令来实现

指令举例三：





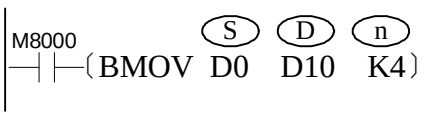
16bit	32bit	P	FNC	BMOV	数据成批传送
✔		✔	15		
7 步		7 步	指令格式：BMOV (S) (D) (n)		

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

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。

其中(n)的取值范围是1～512。

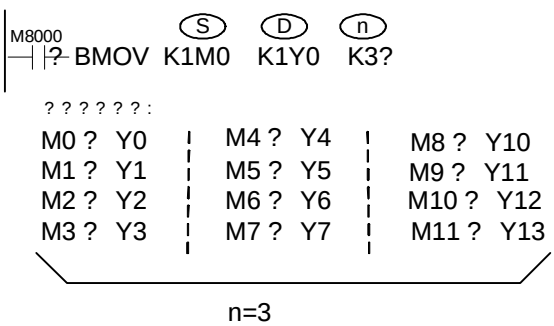
当特殊变量M8024＝1时，成批传送的方向相反，即将由(D)指定起始地址的(n)个变量值复制到由(S)指定起始地址的(n)个单元中。



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。

指令举例：



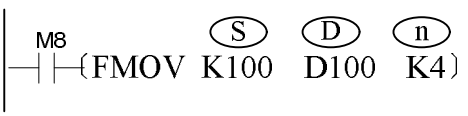
16bit	32bit	P	FNC 16	FMOV	数据多点传送
✔	✔	✔			
7 步	13 步		指令格式：FMOV (S) (D) (n)		

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

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。

其中 (n) 的取值范围是1～512。

指令举例：



当M8置ON时完成的操作是：

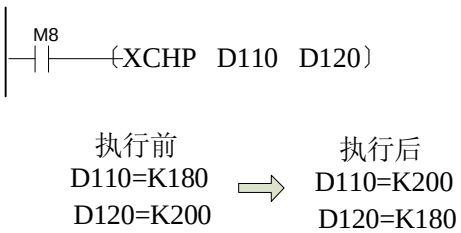
k100 → D100
k100 → D101
k100 → D102
k100 → D103

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

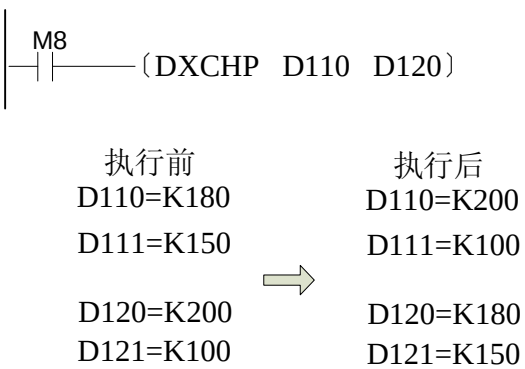
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)								✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		

需要触点驱动，有 2 个操作变量，将(S)和(D)的值彼此交换。

指令举例一：



指令举例二：



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，32位的指令也一样，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。一般用SWAP指令来实现。

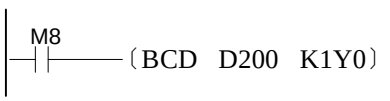
16bit	32bit	P	FNC	BCD	BCD 交换
✔	✔	✔	18		
5 步	9 步		指令格式：BCD (S) (D)		

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)								✔	✔	✔	✔	✔	✔	✔	✔

需要触点驱动，有 2 个操作变量，将(S)（BIN）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。

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。M8067、M8068会置ON，D8067记录错误代码。

指令举例：



将D200的BIN值转换成BCD值后，将结果的个位数存于K1Y0中（Y0~Y3四个bit 元件).
若D200=H000E(十六进制）=K14（十进制），则变换后Y0~Y3=0100(BIN)
若D200=H0028(十六进制）=K40（十进制），则变换后Y0~Y3=0000(BIN)

16bit	32bit	P	FNC	BIN	BIN 交换
✔	✔	✔	19		
5 步	9 步		指令格式：BIN (S) (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)								✔	✔	✔	✔	✔	✔	✔	✔

需要触点驱动，有 2 个操作变量，将(S)(BCD)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。

(S)（BCD）的有效范围，16bit：0~9999；32bit：0~99，999，999

(S)的数据内容不是BCD值（以Hex表示有任一位数不在0~9的范围内）时将会产生运算错误，M8067、M8068会置位。

指令举例：

当M8置位时将K1Y0的BCD值作BIN转换后存入D200中

4.3.2.3 四则逻辑运算（20~29）

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

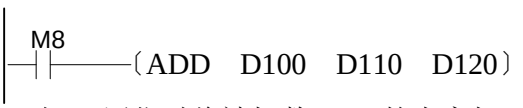
若计算结果为0，则0标志(M8020)会置位；

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(M8022)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(M8021)会置位；

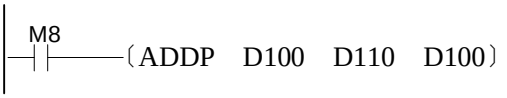
进行32bit运算时，指令中变量地址为为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖。

指令举例一：



当M8置位时将被加数D100的内容加上加数D110的内容后存放到D120中，假如D100=K8;D110=K-12,则D120=8+(-12)=k-4

指令举例二：



当M8置位时将被加数D100的内容加上加数D110的内容后再存放回被加数D100中。

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

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(M8022)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(M8021)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。

指令举例：

```
M8
─┤├──〔SUB  D100  D110  D120〕
```

当M8置位时，将被减数D100的内容减去减数D110的内容后存放到D120中，假如D100=K10;D110=K8,则D120=10-8=K2

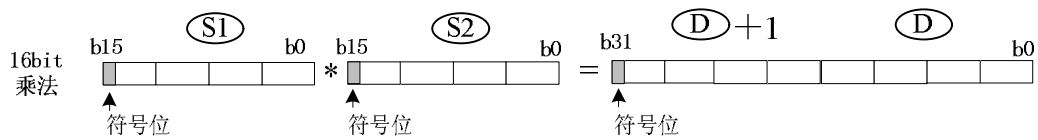
16bit	32bit	P	FNC	MUL	BIN 乘法运算
✔	✔	✔	22		
7 步	13 步		指令格式：MUL (S1) (S2) (D)		

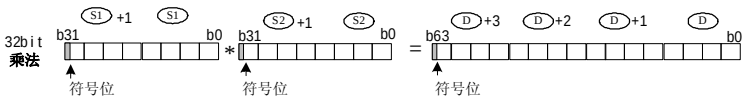
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔		

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。

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。





指令举例：

```
M8
─┤├──〔MUL   D100   D110   D120〕
```

当M8置位时将被乘数D100的内容乘以乘数D110的内容后存放到D120中.
假如D100=K5;D110=K9,则D120=5×9=K45
假如D100=K1234;D110=K5678,则D120,d121=1234×5678=K7006652,需注意此时积大于16bit,需用到D的相邻高位D121,D120

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

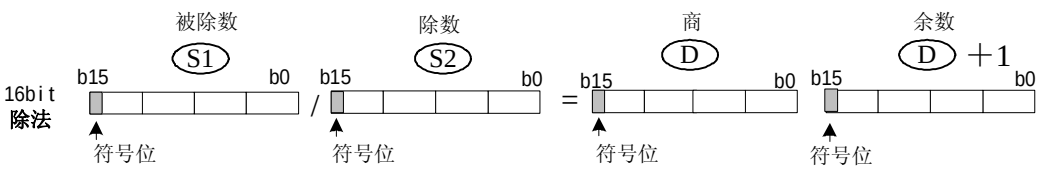
表中V、Z元件仅在16bit运算时可用。

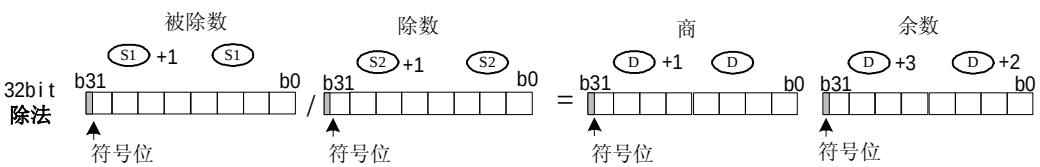
进行32bit运算时，指令中(S1)和(S2)变量地址为低16bit地址，相邻高编号地址单元为高16bit，编程时防止重复或误覆盖；计算所得的商存入(D)、(D)+1所指单元，余数存入(D)+2、(D)+3地址单元中。

若除数(S2)为0，会发生计算错误；

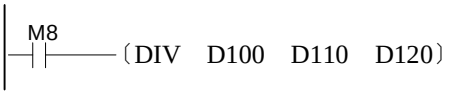
若将位元件（KnX/KnY/KnM/KnS）指定为(D)，不能得到余数；

若被除数为负数，余数即为负数。





指令举例：



当M8置位时将被除数D100的内容除以除数D110的内容后存放到D120中。
假如D100=K5；D110=K2，则D120=K2，商存放于D121，D121=K1。

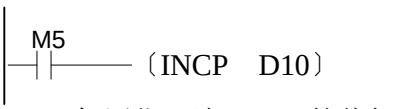
16bit	32bit	P	FNC	INC	BIN 加 1 运算
✔	✔	✔	24		
3 步	5 步		指令格式：INC (D)		

指令每执行一次，(D)中的数值增加 1。

16位运算时，32，767再加1变为-32，768；32位运算时，2，147，483，647再加1变为-2，147，483，648。

本指令对 0 标志、进位、借位标志都不刷新

指令举例：



M5每置位一次，D10的值加1

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

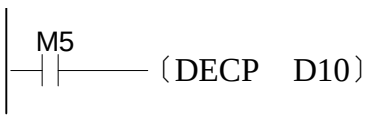
指令每执行一次，(D)的数值减 1。

16位运算时，-32，768再减1变为32，767；32位运算时，-2，147，483，648再减1变为 2，147，483，647。

本指令对 0 标志、进位、借位标志都不刷新。

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。

指令举例：



M5每置位一次，D10的值减1

16bit	32bit	P	FNC	WAND	逻辑与
✔	✔	✔	26		
7 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。

逻辑的‘与’（AND）运算的规则为任一为 0 结果为 0。

1∧1=1　1∧0=0　0∧1=0　0∧0=0

16bit	32bit	P	FNC	WOR	逻辑或
✔	✔	✔	27		
7 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。

逻辑的’或’（OR）运算的规则为任一为 1 结果为 1。

1V1=1　1V0=1　0V1=1　0V0=0

16bit	32bit	P	FNC	WXOR	逻辑异或
✔	✔	✔	28		
7 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。

逻辑的’异或’（XOR）运算的规则为两者相同结果为 0，两者不同结果为 1。

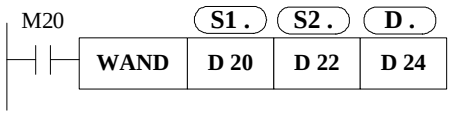
1(1=0　1(0=1　0(1=1　0(0=0

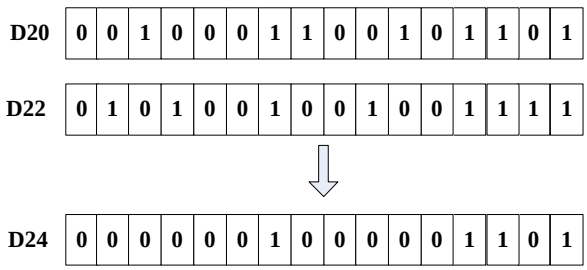
上述三个指令的操作数适用变量类型如下表，当为 32bit 指令时，寄存器变量则占用后续相邻地址的共 2 个单元：

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)								✔	✔	✔	✔	✔	✔	✔	✔

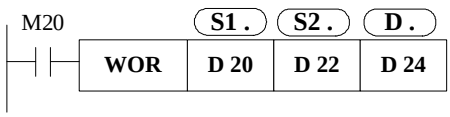
指令举例：

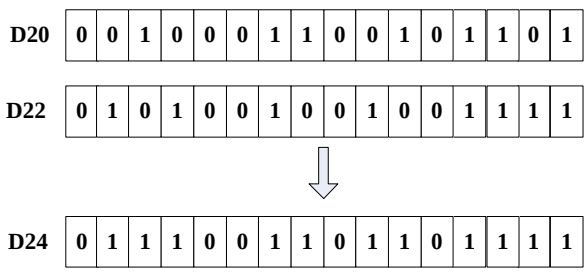
逻辑与


M20
S1 .
S2 .
D .
WAND
D 20
D 22
D 24

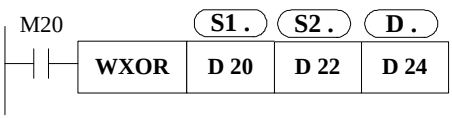

D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1
D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1
D24 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1

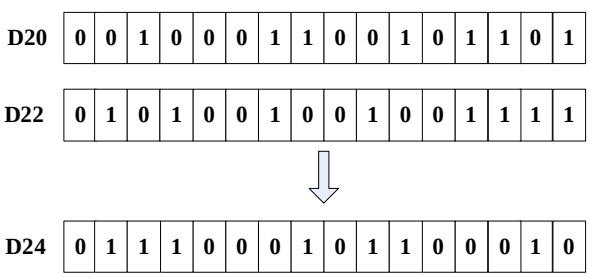
逻辑或


M20
S1 .
S2 .
D .
WOR
D 20
D 22
D 24


D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1
D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1
D24 0 1 1 1 0 0 1 1 0 1 1 0 1 1 1 1

逻辑异或


M20
S1 .
S2 .
D .
WXOR
D 20
D 22
D 24


D20 0 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1
D22 0 1 0 1 0 0 1 0 0 1 0 0 1 1 1 1
D24 0 1 1 1 0 0 0 1 0 1 1 0 0 0 1 0

16bit	32bit	P	FNC	NEG	求补运算
✔	✔	✔	29		
3 步	5 步		指令格式：NEG (D)		

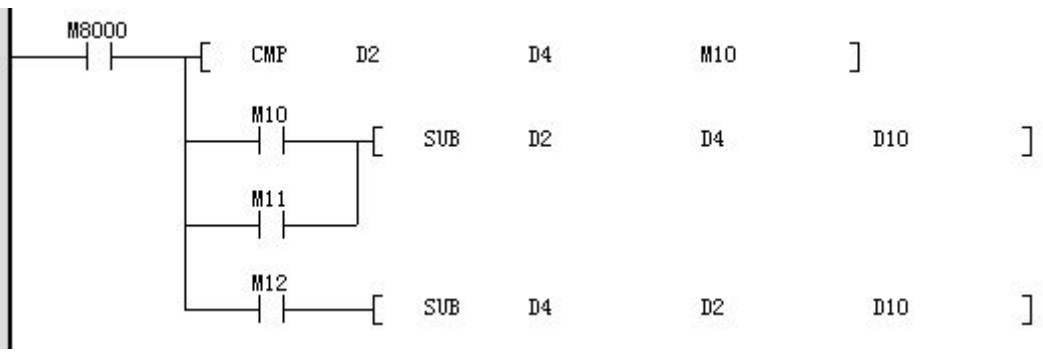
需要触点驱动，有 1 个操作变量。将 (D) 的数值逐位取反、再加 1，存回 (D) 中。

此指令一般用脉冲执行型指令。

使用 NEG 指令，可得到与负的 BIN 值相对应的绝对值。

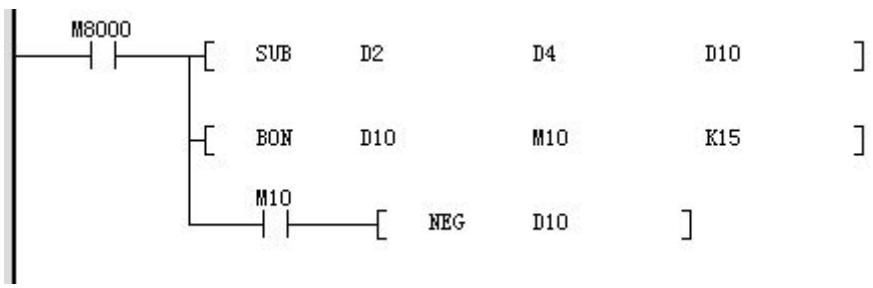
指令举例：

减法运算的差取绝对值



若 D2>D4 时，M10=On。若 D2=D4 时，M11=On。若 D2<D4 时，M12=On。由此可保证 D10 为正值

此程序可用下列的程序来表示



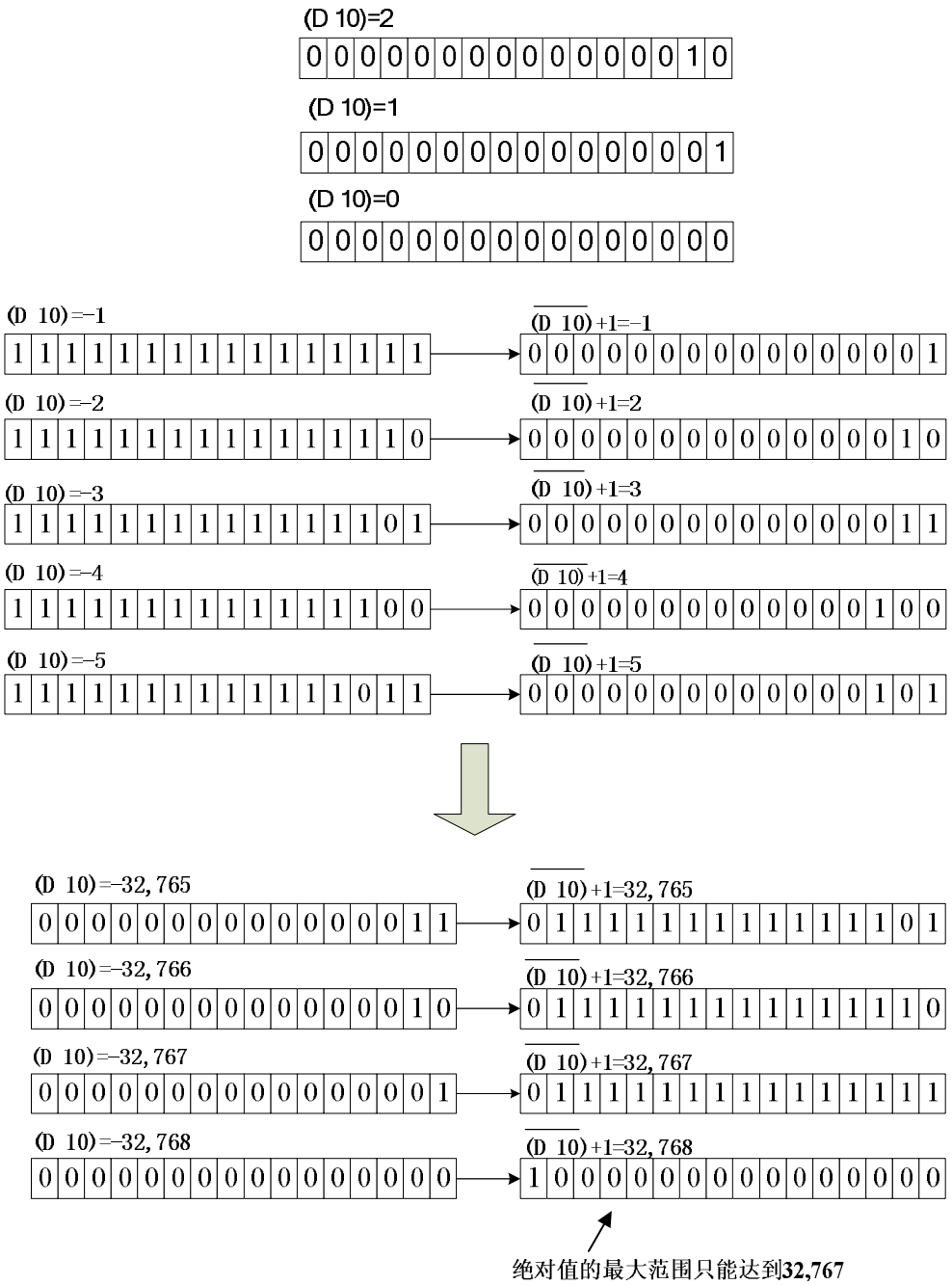
当D10的bit15为“1”时（表示D10为负数），M10=On，用NEG指令将D10取补码可得到D10的绝对值。

上述两例中假如D2=K4，D4=K8；或者D2=K8，D4=K4，D10的结果均为K4

补充说明：负数的表现及绝对值：

1. 正负数是以寄存器最上位（最左边）的位内容来表现，为“0”时为正数、为“1”时为负数。

2. 最高位为1时，可使用NEG指令将它转成绝对值。


(D 10)=2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
(D 10)=1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
(D 10)=0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
(D 10)=−1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
(D 10)+1=−1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
(D 10)=−2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
(D 10)+1=2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
(D 10)=−3
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
(D 10)+1=3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
(D 10)=−4
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
(D 10)+1=4
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
(D 10)=−5
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
(D 10)+1=5
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
(D 10)=−32, 765
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
(D 10)+1=32, 765
0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
(D 10)=−32, 766
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
(D 10)+1=32, 766
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
(D 10)=−32, 767
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
(D 10)+1=32, 767
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
(D 10)=−32, 768
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
(D 10)+1=32, 768
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
绝对值的最大范围只能达到32,767

4.3.2.4 循环移位（30~39）

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

将 (D) 的内容循环右移 (n) 位。

本指令一般使用脉冲执行型指令。

16bit	32bit	P	FNC	ROL	循环左移
✔	✔	✔	31		
5 步	9 步		指令格式：ROL (D) (n)		

将 (D) 的内容循环左移 (n) 位。

本指令一般使用脉冲执行型指令。

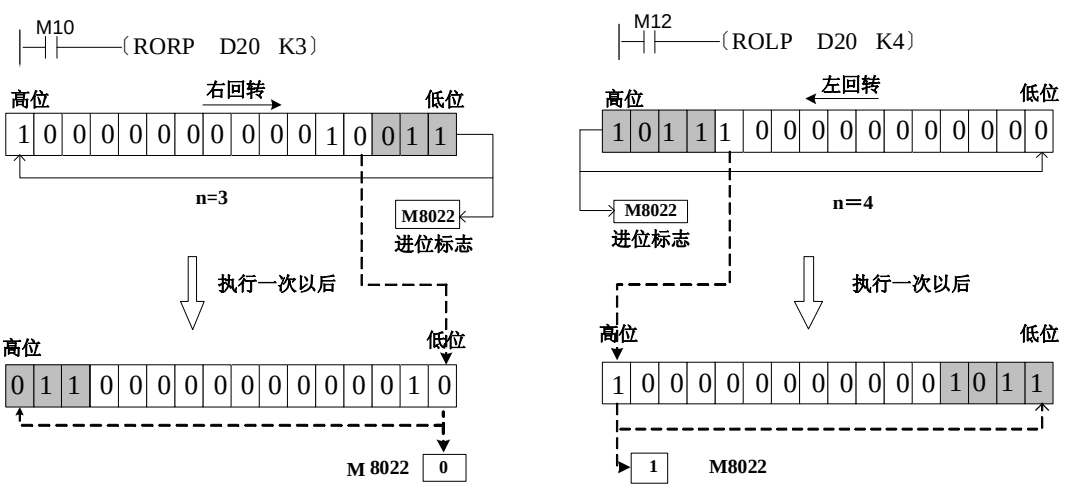
上述两个指令的操作数适用变量类型如下表，当为 32bit 指令时，寄存器变量则占用后续相邻地址的共 2 个单元：

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

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

循环移动的最终位被存入进位标志中。

指令举例：



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

将 (D) 的内容连同进位标志 M8022 循环右移 (n) 位。

本指令一般使用脉冲执行型指令。

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

将 (D) 的内容连同进位标志 M8022 循环左移 (n) 位。

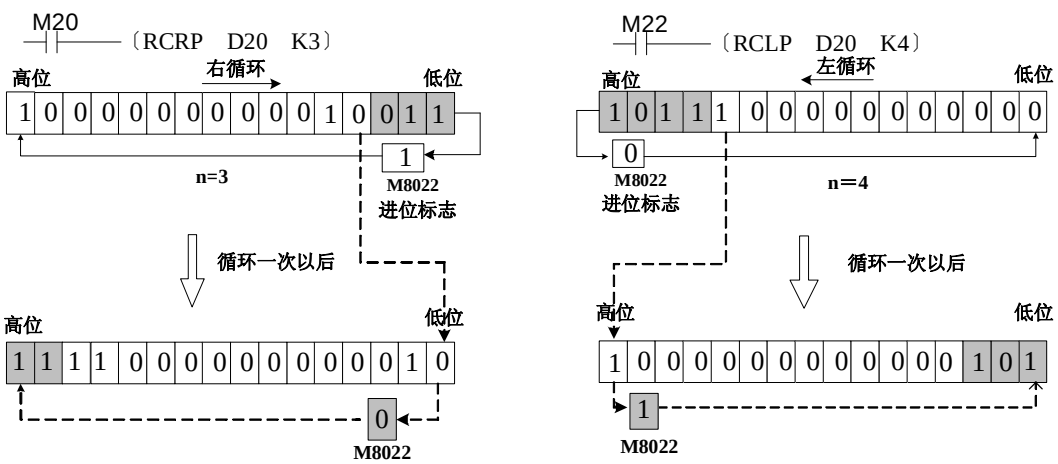
本指令一般使用脉冲执行型指令。

上述两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
(D)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(n)	常数，n=1~16（16bit）；n=1~32（32bit）														

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

指令举例：



16bit	32bit	P	FNC	SFTR	位右移
✔		✔	34		
9 步		9 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。

本指令一般使用脉冲执行型指令。

16bit	32bit	P	FNC	SFTL	位左移
✔		✔	35		
9 步		9 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。

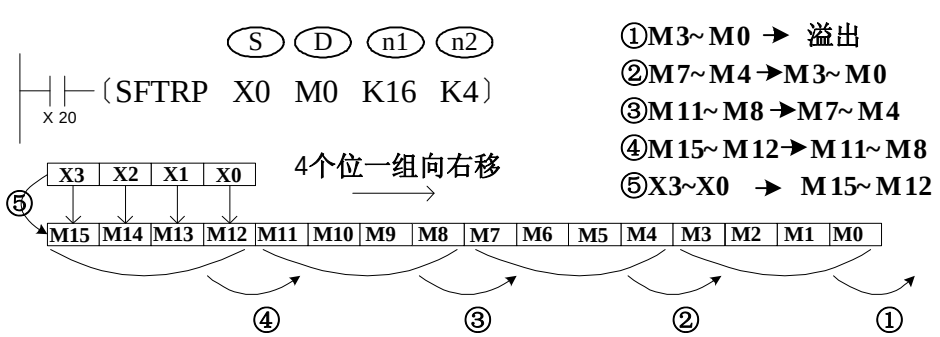
本指令一般使用脉冲执行型指令。

上述两个指令的操作数适用变量类型如下表：

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														

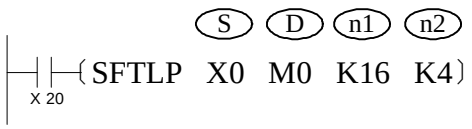
指令举例一：

SFTR命令：

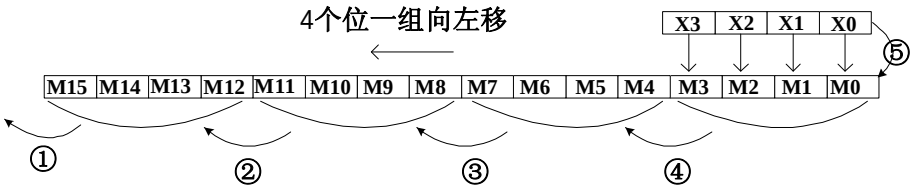


指令举例二：

SFTL命令：



① M15～M12 ➔ 溢出
② M11~M8 ➔ M15~M12
③ M7~M4 ➔ M11~M8
④ M3~M0 ➔ M7~M4
⑤ X3~X0 ➔ M3~M0



16bit	32bit	P	FNC	WSFR	字右移
✔		✔	36		
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

本指令一般使用脉冲执行型指令。

16bit	32bit	P	FNC	WSFL	字左移
✔		✔	37		
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。

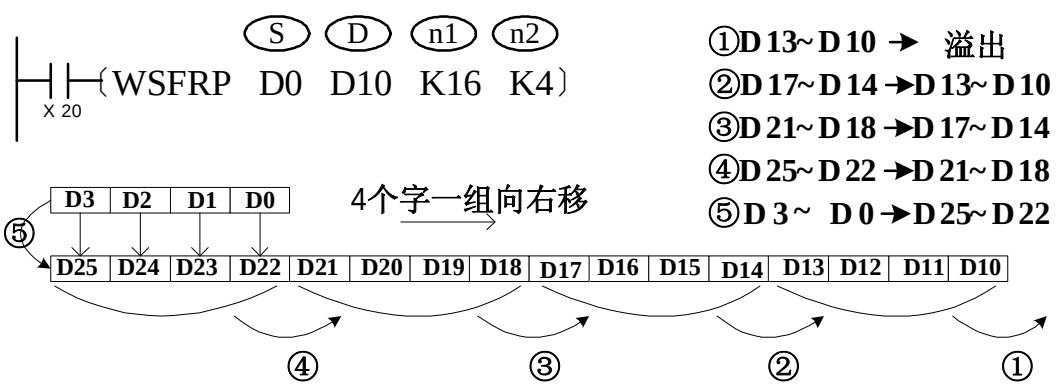
本指令一般使用脉冲执行型指令。

上述两个指令的操作数适用变量类型如下表：

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≤2048														
(n2)	常数，n2≤n1														

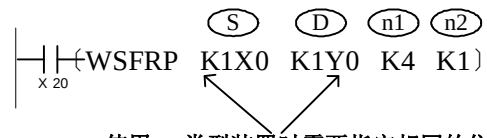
指令举例一：

WSFR命令：

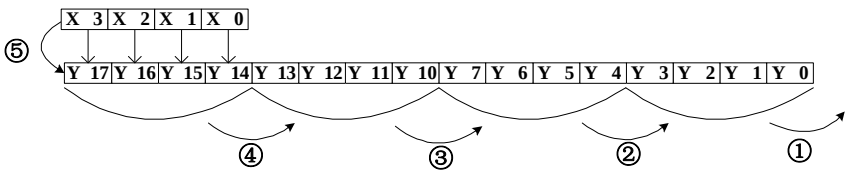


指令举例二：

WSFR命令：



使用Kn类型装置时需要指定相同的位数



扫描一次的位数右移动作依照下列编号1~5 动作。

1：Y3~Y0？ 进位

2：Y17~Y14？ Y13~Y10

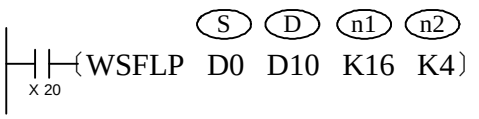
3：Y13~Y10？ Y7~Y4

4：Y7~Y4？ Y3~Y0

5：X3~X00？ Y17~Y14 完成

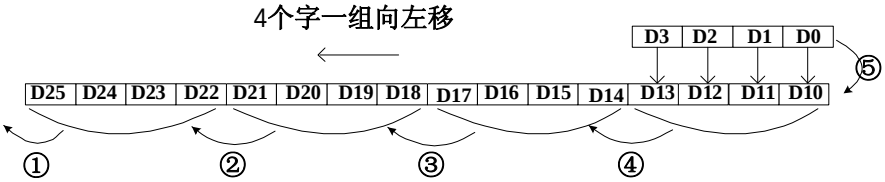
指令举例三：

WSFL命令：



① D25~D22 ➔ 溢出
② D21~D18 ➔ D25~D22
③ D17~D14 ➔ D21~D18
④ D13~D10 ➔ D17~D14
⑤ D3~ D0 ➔ D13~D10

4个字一组向左移



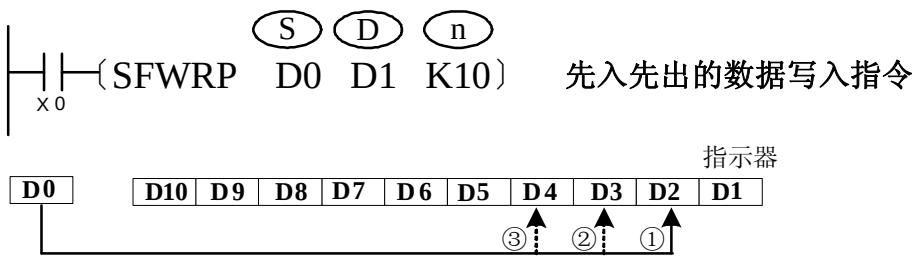
16bit	32bit	P	FNC	SFWR	“先进先出”写入
✔		✔	38		
7 步		7 步	指令格式：SFWR (S) (D) (n)		

操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，2≤n≤2048														

将(S)的值写入由(D)地址起始，个数为(n)的“先进先出”队列中，以第一个编号装置作为指针，当指令执行时，指针内容值先加1，之后**S**所指定的装置其内容值会写入先入先出(D)数据串列中由指针所指定的位置。

本指令一般使用脉冲执行型指令。

指令举例：



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

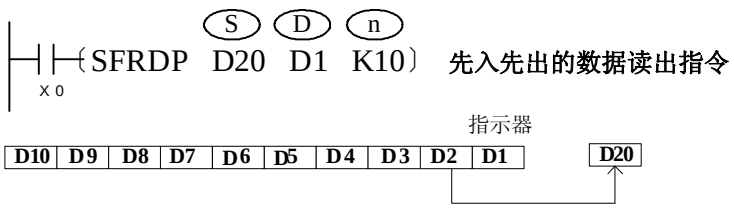
16bit	32bit	P	FNC	SFRD	“先进先出”读出
✔		✔	39		
7 步		7 步	指令格式：SFRD (S) (D) (n)		

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

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。，以第一个编号装置作为指针，当指令执行时，指针内容值先减1，之后**S**所指定的装置其内容值会写入先入先出(D)数据串列中由指针所指定的位置。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

本指令一般使用脉冲执行型指令。

指令举例：



X0由OFF到ON本指令动作依照下列编号 1~3 动作。(D10 内容保持不变)，
1：D2 的内容被读出传送至 D20 当中。
2：D10~D3 全部往右移位一个寄存器 。
3：指针D1 内容减1。

4.3.2.5 数据处理（40~49）

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

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)		✔	✔	✔							✔	✔	✔		

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。

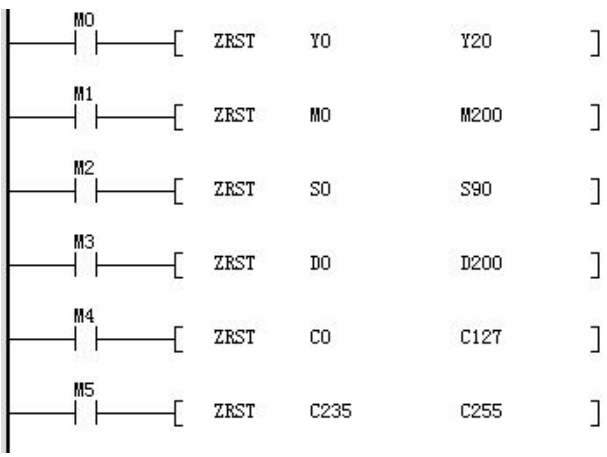
其中要求：

(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；

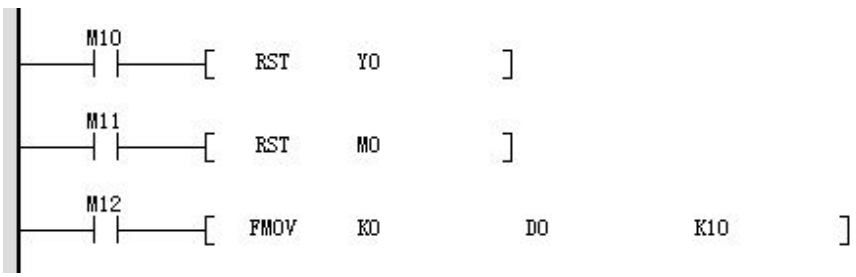
本指令为16bit，但(D1)和(D2)可指定32bit的计数器，此时应同为32bit型或同为16bit型；

指令举例：



补充说明：

位装置Y、M、S和字装置T、C、D也可使用RST指令来单独复位；字装置T、C、D和位寄存器KnY、KnM、KnS也可以用FMOV来多点清除。例如：



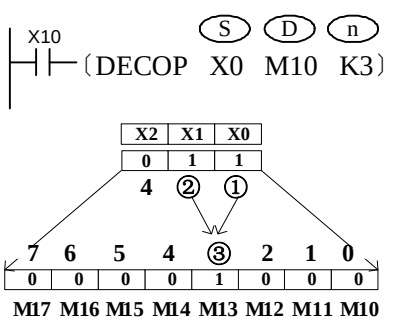
16bit	32bit	P	FNC	DECO	解码
✔		✔	41		
7 步		7 步	指令格式：DECO (S) (D) (n)		

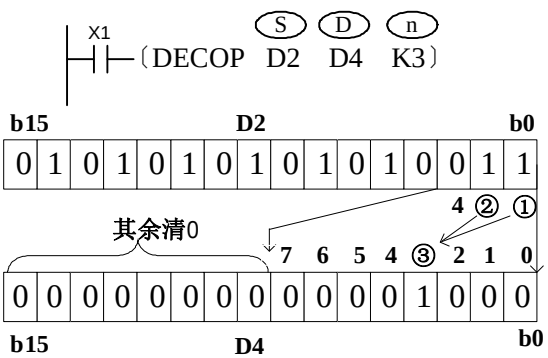
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✔	✔	✔	✔	✔	✔					✔	✔	✔	✔	✔
(D)		✔	✔	✔							✔	✔	✔		
(n)	常数，n=1~8。若 n＝0，指令不执行；其他值则执行出错。														

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。

- 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(D)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。
- 本指令一般使用脉冲执行型指令。

编程举例：





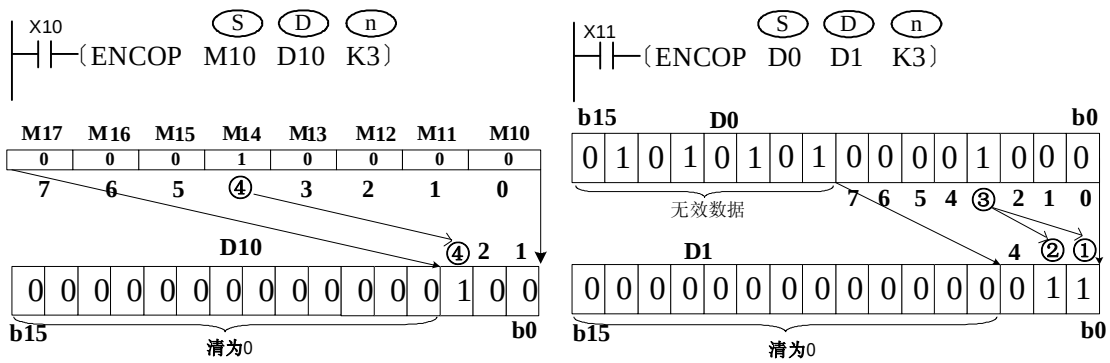
16bit	32bit	P	FNC	ENCO	编码
✔		✔	42		
7 步		7 步	指令格式：ENCO (S) (D) (n)		

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

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

- 源地址内有多个位是 1 时，只计算高位侧的第一个为 1 的位；(S)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(S)如果是位元件，其点数是 256 点。
- 本指令一般使用脉冲执行型指令。

指令举例：



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

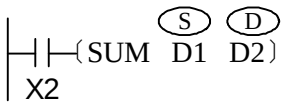
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

计算(S)的BIN进制值中为1的位个数，存入(D)。

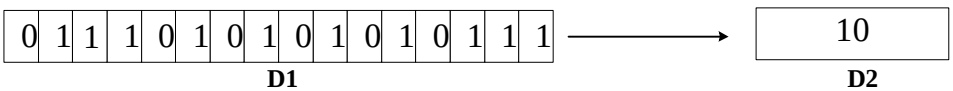
使用D SUM和D SUM P指令的情况下，（(S)+1，(S)）的32位中的1的个数写入(D)，同时(D)+1全部为0。

若(S)中的位全部为零，则零标志位M8020会置ON

指令举例：



将D1 中的位为1的个数总和存入D2中.



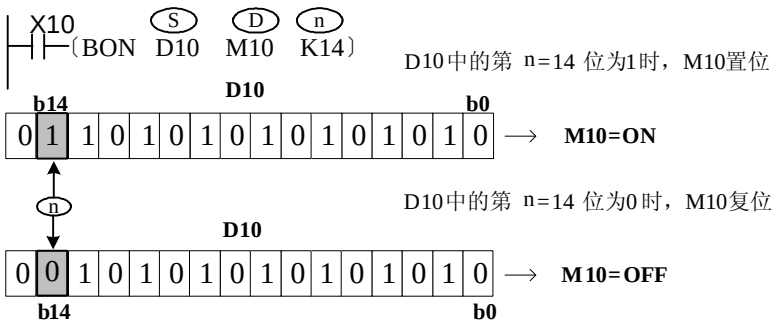
D1 **D2**

16bit	32bit	P	FNC 44	BON	ON 位判断
✔	✔	✔			
5 步	9 步		指令格式：BON (S) (D) (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
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔	✔	✔											
(n)	n=0～15（16bit）；n=0～31（32bit）														

判断(S)的第(n)位的状态，结果存入(D)。

指令举例：



X10由ON变成Off 时，M10 仍保持之前的状态。

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

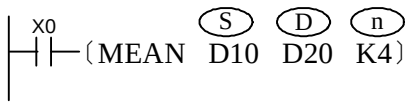
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔	✔	✔
(n)	常数，n=1～64，其他值时，计算会出错。														

求取由(S)开始的(n)个变量的平均值（先求和，再除以n），存入(D)。

若计算中有余数，余数将被舍弃；

当n的值不在1~64的范围时，会计算出错。

指令举例：



(D10+D11+D12+D13)/4=D20
假如D10=K5,D11=K5,D12=K15,D13=K52;则D20=K19.余数1被舍去

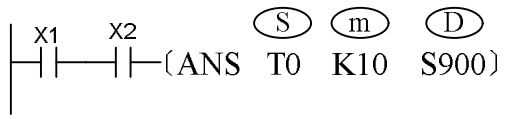
16bit	32bit	P	FNC 46	ANS	报警器置位
✔					
7 步			指令格式：ANS (S) (m) (D)		

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)				✔											
(m)	常数，m=1～32767，（单位为 100ms）。														

驱动信号报警器的方便指令。

其中(S)的范围为T0~T199，(D)的范围为S900~S999。

指令举例：

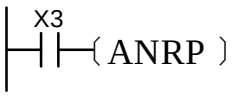


如果 X1 和 X2 同时接通 l 秒以上，则 S900 被置位，以后即使 X1 或 X2 为 OFF 状态，S900 仍保持动作状态（但是 T0 会复位，值变成 0）。若不满 1 秒，X1 或 X2 变为 OFF 时，定时器复位。

如果预先将 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）且当 S900~S999 中任意一个为 ON 时，M8048（报警器动作置 ON）。

16bit	32bit	P	FNC	ANR	报警器复位
✔		✔	47		
1 步		1 步	指令格式：ANR(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，则复位最小编号为 ON 的报警点。

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 ANRP 指令。

指令举例：

M8049为ON，当程序中S900~S999任意一个为ON时，M8048置ON，Y0有报警输出，

假如程序中S910、S911、S912、S913都为ON，则当X5第一次由OFF置ON时，S910被复位，当X5第二次由OFF置ON时，S911被复位，以此类推。

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

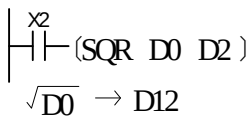
将(S)按BIN值开平方运算，结果存入(D)。

只能指定(S)为正数，如(S)为负数则运算错误标志 M8067 会置 ON，指令不被执行；

运算结果(D)只取整数。舍去小数点，有小数点被舍去时借位标志 M8021 置 ON；

运算结果是0时，零位标志M8020置ON。

指令举例：



假如D0=K100,则X2置ON的时候，D12=K10

假如D0=K110,则X2置ON的时候，D12=K10，小数被舍去

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

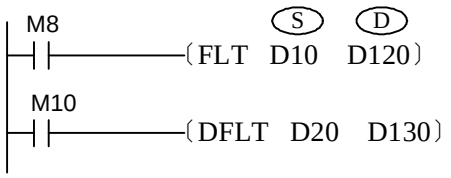
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

将整数(S)转换为浮点数，结果存入(D)和(D)＋1单元。

常数 K、H 在各浮点运算指令中自动转换，因此在本 FLT 指令中不能使用。

这个指令的逆变换指令是 INT（将 2 进浮点数值变换成 BIN 整数）

指令举例一：



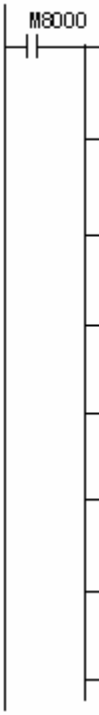
当M8=ON时，将16bit数D10中(16位BIN整数)转换为二进制浮点数后，存放到（D121,D120）

当M10=ON时，将32bit数（D21,D20）中(32位BIN整数)转换为二进制浮点数后，存放到（D131,D130）

指令举例二：

使用指令来完成下列的浮点运算。

D100/K125.5*(X17~X0)=D200



(FLT D100 D110)	将D100(内为BIN整数)变换成D111、D110(2进浮点数值)。
(DEDIV K1255 K10 D120)	将 K1255÷K10 结果存于D121、D120(2 进浮点数值)。
(BIN K4X0 D130)	将X17~X0(BCD值) 变换成D130(16位BIN 整数)。
(FLT D130 D140)	将D130(内为BIN整数)变换成D141、D140(2 进浮点数值)。
(DEDIV D110 D120 D150)	2进浮点数除法(D111、D110)÷(D121、D120)结果存于D151、D150(2 进浮点数值)。
(DEMUL D150 D140 D200)	2进浮点数乘法(D151、D150)×(D141、D140)结果存于D201、D200(2 进浮点数值)。
(DEBCD D200 D160)	2进浮点数值D201、D200 变换成10 进浮点数值D161、D160。(十进制浮点监视用)
(DINT D200 D170)	2进浮点数值D2 0 1、D2 0 0 变换成BIN整数D171、D170。(32位BIN整数)

4.3.2.6 高速处理（50~59）

16bit	32bit	P	FNC	REF	输入输出端口状态刷新
✔		✔	50		
5 步		5 步	指令格式：REF (D) (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
(D)	✔	✔													
(n)	常数，n=8~256，且为 8 的倍数														

将(D)地址开始的(n)个元件状态进行立即更新。

由于PLC访问端口是按字节访问的特性，故要求：

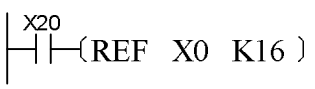
(D)的地址应为X0、X10、…Y0、Y10、...等最低位为0的编号元件；

(n)的值必须是8的倍数（n=8~256）。

正常情况下，输入端口X的状态读取在每次程序开始执行扫描之前进行，输出端口Y的状态刷新则在每次程序执行扫描完毕（执行到END）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。

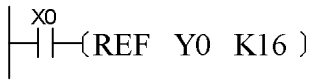
- 可用在 FOR~NEXT 指令之间、CJ 指令之间等。
- 可用于中断子程序中进行输入输出刷新一获取最新的输入信息并及时输出运算结果。
- 实际的输入端口状态变化延迟决定于输入元件的滤波时间，X0～X7 有数字滤波功能，滤波时间在 0～60ms 范围内可设（FNC51（REFF 指令）），其余 IO 端口为硬件滤波，滤波时间约 10ms。具体参数请参考可编程控制器的用户手册。
- 实际的输出端口状态变化延迟决定于输出元件（如继电器）的响应时间。输出刷新中的输出接点将在输出继电器（晶体管）应答时间后动作。继电器输出型的应答滞后时间约为 10ms（最大 20ms），晶体管输出型高速输出口约 10μs、普通点输出口约 0.5ms。具体参数请参考可编程控制器的用户手册。

指令举例一：



执行上述程序时，若 X20 为 ON 状态，会立即读取 X0~X17 的输入点状态，更新输入信号，没有产生输入延迟。

指令举例二：



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，输出信号立即更新。

不必到 END 指令才输出。

16bit	32bit	P	FNC	REFF	输入滤波调整
✔		✔	51		
3 步		3 步	指令格式：REFF (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														

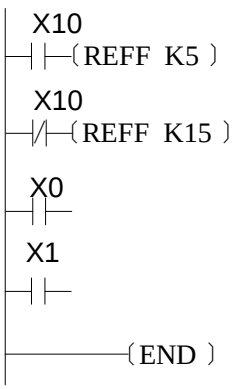
将X0~X7输入端口的滤波时间常数设定为(n)。

可编程控制器中，X0~X7 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 D8020 改变为 0~60ms。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。

亦可用 MOV 指令直接对 D8020 进行赋值改变滤波时间。

指令举例：



X10 为 ON 时，将 X0~X7 的输入滤波时间设为 5ms，X10 为 OFF 时，将 X0~X7 的输入滤波时间设为 15ms；

16bit	32bit	P	FNC	MTR	矩阵输入
✔		✔	52		
9 步		9 步	指令格式：MTR (S) (D1) (D2) (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
(S)	✔														
(D1)		✔													
(D2)		✔	✔	✔											
(n)	常数，n=2~8														

此指令只能用于晶体管输出型PLC，通过将8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)为矩阵扫描输入的硬件X端口的起始地址，要求为X0、X10…等最低位为0的编号元件，占用连续8个；

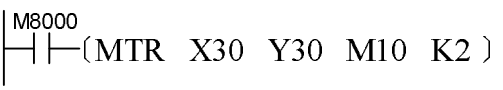
(D1)为矩阵扫描输出的硬件Y端口的起始地址，要求为Y0、Y10…等最低位为0的编号元件，占用连续n个（n=2~8）；

(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等最低位为0的编号元件；

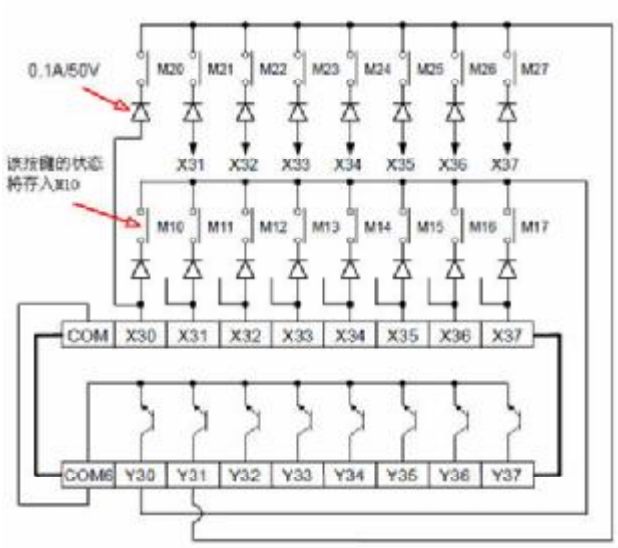
(n)为矩阵扫描的列数，即扫描用Y输出的个数。

本指令的条件接点一般都使用常 On 接点 M8000.

指令举例：



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 ON；

若通过 8 点 X 输入和 8 点晶体管 Y 输出，可获得最大 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)		

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)		✔	✔	✔											

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~C255，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。M8059置ON则禁止了所有的高速计数器中断（I010-I060）。

一般指令Y输出与DHSCS指令Y输出的差异：以（指令举例一）为例

1. 当C255的现在值由99→100变化时，C255接点立即导通，但执行到OUT Y10时，Y10仍会受扫描周期影响，在END后才输出。

2. 当C255的现在值由99→100及101→100变化时，DHSCS指令输出Y10是以中断方式立即输出到外部输出端，与PLC 扫描周期无关。但仍会受输出模

块继电器(10ms)或晶体管(10us)的输出延迟。

使用说明：

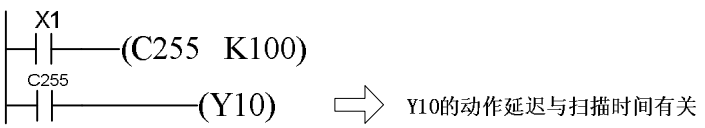
- 使用 HSCS 指令时，应保证所使用的计数器已被启用（见指令举例一），否则该计数器的值将不会有变化；
- 计数器是以中断方式响应计数器的输入信号，及时比较，若本次比较时满足匹配关系，比较输出立即置位。例如指令举例一中，若 C255 的当前值变为 99→100 或 101→100 时，Y10 立即置位，且一直保持该状态，之后即使 C255 与 K100 的比较结果变成不相等，Y10 仍然保持 On 状态，除非有另外的复位指令操作；

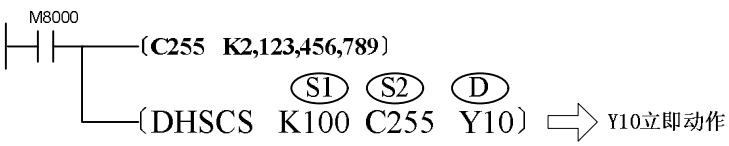
- 指令的比较输出只决定于脉冲输入时的比较结果动作，即使采用 DMOV、DADD 等指令改写高速计数器 C235~C255 的内容，若没有脉冲输入，比较输出也不会变化；单纯的指令驱动能流也不能改变比较结果；
- 指令输出若为 Y 端口，必须为 Y0~Y17 范围，这样才能保证输出得到立即响应；多次驱动 HSCS 指令或与 HSCR、HSZ 指令同时驱动，对象输出 Y 的高 2 位作为同一序号的软元件。例：使用 Y000 时为 Y000~Y007，Y010 时为 Y010~Y017 等；
- 当 HSCS 指令的输出目标为中断 I010～I060 时，每个中断号只能使用 1 次，不可重复。
- HSCS、HSCR、HSZ 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。

高速计数中断指针与设置：

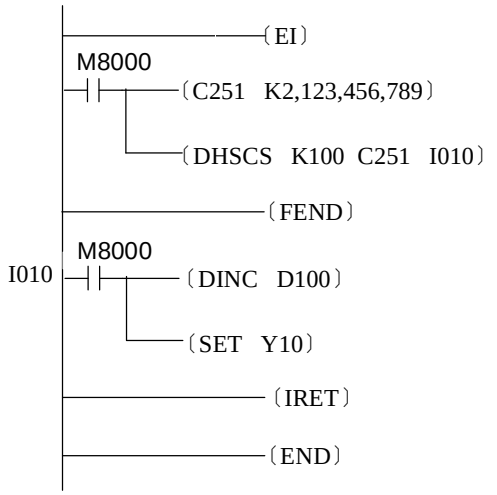
输入编号	中断禁止指令
I010	M8059
I020	
I030	
I040	
I050	
I060	

指令举例一：





指令举例二：



DHSCS指令的D操作数范围也可指定I0□0，□=1~6，作为计数器计数到达时，发生中断，执行该中断服务程序。

如果M8059置ON则禁止了所有的高速计数器中断。

注意此时的D装置用I010和Y、M、S输出点的ON信号区别：

1、用Y输出点:若C251的当前值变为99→100或101 →100时，Y立即置ON，且一直保持ON状态，之后即使C251与K100的比较结果变成不相等，Y仍然保持On 状态,除非有另外的复位指令操作；

2、用I010:若C251的当前值变为99→100或101 →100时，I010只会产生一次中断，不会常ON。

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)		

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)		✔	✔	✔											

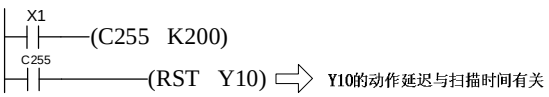
当(S2)计数器的当前值等于设定值(S1)时，立即复位(D)。其中：

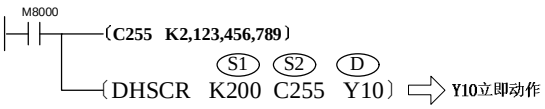
(S1)为设定的比较值，其值的宽度（bit位数）决定于(S2)计数器的位数；

(S2)变量必须为高速计数器C235~C255，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。

指令举例：





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 HSCS 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)		

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)		✔	✔	✔											

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：

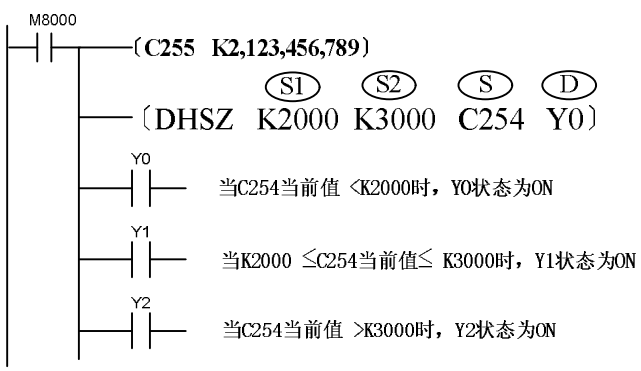
(S1)为设定的比较区间区间下限，其值的宽度（bit位数）决定于(S)计数器的位数，其值必须不大于(S2)，即(S1)≤(S2)；

(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~C255，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。

指令举例：



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 ㊿ ㊿ ㊿ ㊿〕中的㊿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：

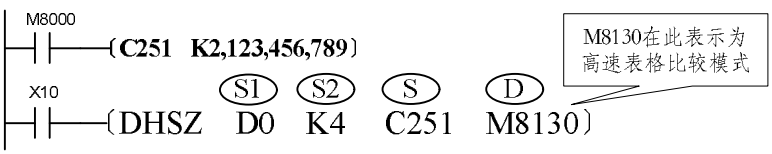
㊿只对应数据寄存器 D 变量，用于表示比较表格的起始地址。可用带 V 或 Z，指令启动后不再受 V 或 Z 的影响；

㊿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；

㊿可以指定为高速计数器 C235~C255；

㊿为 M8130，指明为高速表格比较模式。

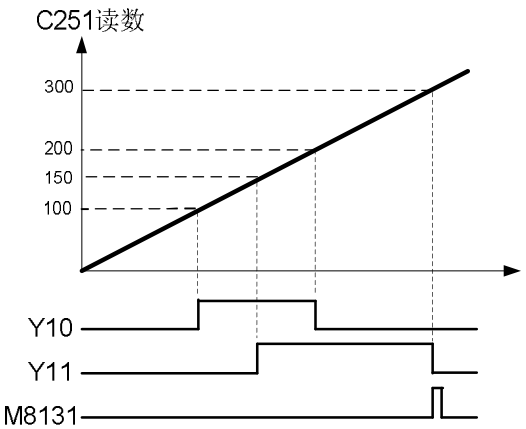
例如：



等效的比较表格为：

㊿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			
㊿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所指定的输出Y被复制成OFF（D3=K0）或是ON（D3=K1）并保持住。而输出Y的动作完全以中断方式来处理。

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，如此的往下顺序执行比较操作，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 Ⓢ1 Ⓢ2 Ⓢ Ⓓ〕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：

Ⓢ1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；

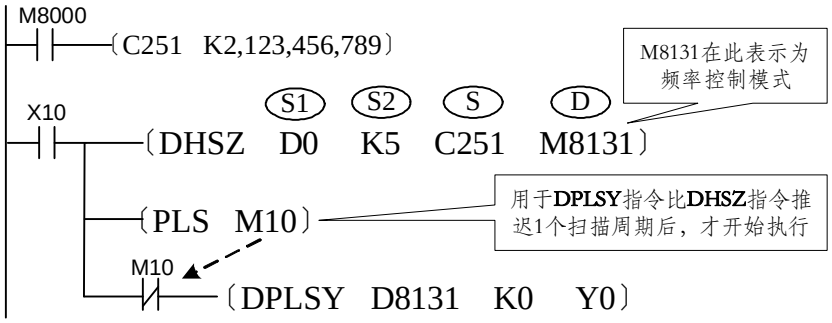
Ⓢ2只可用常数变量 K、H，用于表示表格的行数，被限制为 1≤(K，H)≤128，可用带 V 或 Z，指令启动后不再受 V 或 Z 的影响；

Ⓢ可以指定为高速计数器 C235~C255；

Ⓓ为 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；

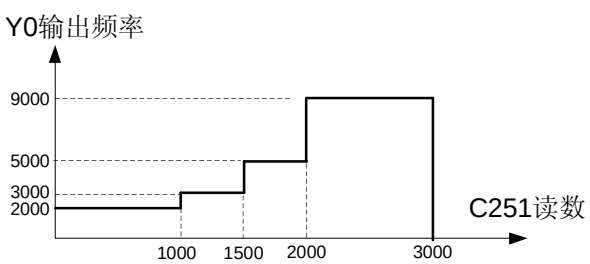
例如：



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：

ⓈS1 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	
ⓈS2 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ⓈS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　MI0〕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	SPD	脉冲密度，转速，线速度
✔			56		
7 步			指令格式：SPD (S1) (S2) (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
(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。

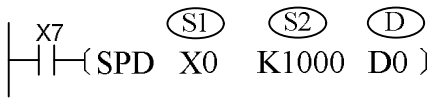
(S1)脉冲信号输入端口，只能为X0~X5；

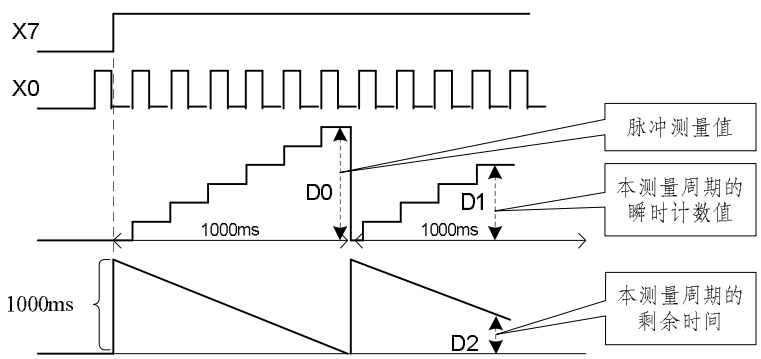
(S2)为设定的脉冲检测时间长度（ms）；

(D)为设定时间长度(S2)内接收的脉冲个数；

被用于SPD指令的X0~X5端口，可同时用于高速计数器或者中断输入中。

指令举例一：





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；

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。

注意：

在新版本的H2U系列PLC中，SPD命令功能有增强，通过置位**[**M8100~M8105**]**标志，可改变计数功能，M8100~M8105标志分别对应X000~X005高速端。(M8100~M8105标志可分别置位，即X0~X5端口的SPD指令功能可分别处理。)

一.：如果功能使能标志为[M8100~M8105]OFF，SPD指令如前面介绍的

(S1)脉冲信号输入端口，只能为X0~X5；

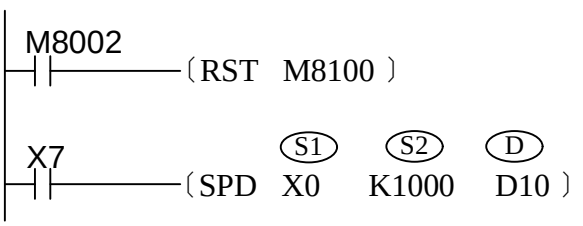
(S2)为设定的脉冲检测时间长度(ms)1~32767；

(D)+0：为在S2时间内的脉冲个数，为16位的数据；

(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。

指令举例二：



D10为设定时间长度1000ms内接收的脉冲个数；D11为实时脉冲计数值；D12为完成本次采样周期的剩余时间。

二：如果功能使能标志M8100~M8105为ON，指令的新功能定义如下：

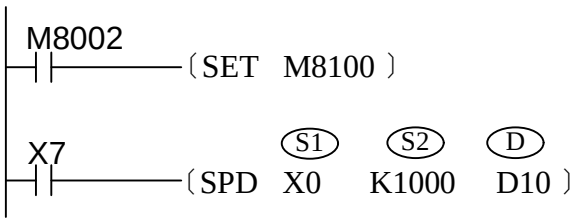
(S1)S1：脉冲信号输入端口，只能为X0~X5；

(S2)S2：为设定的脉冲检测时间长度(ms)1~32767；

(D)+0：为在S2时间内的脉冲个数，为16位的数据；

(D)+1，(D)+2：为每分钟内的脉冲个数，为32位的数据；

指令举例三：



D10 为设定时间长度 1000ms 内接收的脉冲个数；D11，D12 为运行频率=1 分钟脉冲个数*10(0.1 为单位).

16bit	32bit	P	FNC 57	PLSY	脉冲输出
✔	✔				
7 步	13 步		指令格式：PLSY (S1) (S2) (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
(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才适合使用该指令。指令功能是由

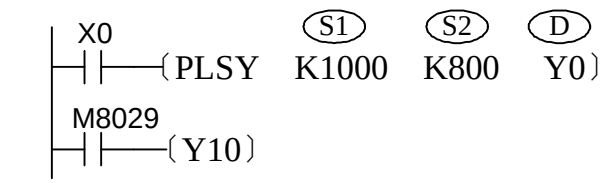
(D)指定的端口，以(S1)的频率，输出(S2)个脉冲，脉冲发送完毕，M8029标志被置位。其中：

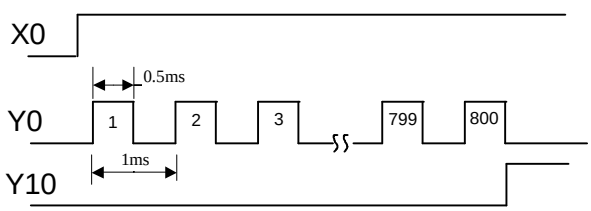
(D)为脉冲输出端口，对于3624MT/2416MT型只能指定Y0或Y1；其他MT型可以指定Y0/Y1/Y2；而MTQ型则可指定Y0/Y1/Y2/Y3/Y4。

(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。

指令举例：





使用 PLSY(16bit 指令)时，(S1)和(S2)都只能是 16bit 宽度；

使用 DPLSY(32bit 指令)时，(S1)和(S2)若为 D、C、T 变量，则按 32bit 宽度处理；

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；（新版本的 H2U 系列 PLC 中，可以在运行中改变输出脉冲数，具体请参考附录 5.8 说明）

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，

PLSY 指令将以当前的参数重新开始脉冲输出。

使用说明：

- PLSY 所用的输出 Y 端口避免与 PWM 或 PLSR 指令所用的 Y 端口重复；
- 40 点 MT、60 点 MT 型 PLC 可以同时使用 2 个 PLSY 指令，或 2 个 PLSR 指令，分别对应 Y0、Y1 端口；其他点数的 MT 型 PLC 可以同时使用 3 个 PLSY 指令，或 3 个 PLSR 指令，分别对应 Y0、Y1、Y2 端口；MTQ 型 PLC 可以同时使用 5 个 PLSY 指令，或 5 个 PLSR 指令，分别对应 Y0、Y1、Y2、Y3、Y4 端口；具体规格请参考可编程控制器硬件用户手册。
- 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		
D8150	低字	PLSY 或 PLSR 指令中设定的输出至 Y2 口的脉冲总数（3624MT/2416MT 没有此端口）	
D8151	高字		
D8152	低字	PLSY 或 PLSR 指令中设定的输出至 Y3 口的脉冲总数（只适用于 MTQ）	
D8153	高字		
D8154	低字	PLSY 或 PLSR 指令中设定的输出至 Y4 口的脉冲总数（只适用于 MTQ）	
D8155	高字		
D8136	低字	已向 Y0 及 Y1 输出的脉冲个数的累计值	
D8137	高字		

注意：

在新版本的 H2U 系列 PLC 中，PLSY 指令的功能有增强，可在 PLSY 指令运行过程中修改脉冲个数、或立即启动下条脉冲输出指令、或实现脉冲输出完成中断等增强功能.。

1. 通过使用特殊位M8135~M8139(分别对应Y0~Y4)为ON，可以实现在运行中更改输

出脉冲个数(可大可小)；

2. 通过使用特殊位M8085~M8089(分别对应Y0~Y4)为ON，可以实现马上启动下条脉冲输出指令，不需要上条能流无效的处理；

3. 通过使用特殊位 M8090~M8094(分别对应 Y0~Y4)为 ON，可以实现脉冲输出完成中断；具体对应如下：

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

具体可参考附录 5.8 中的说明。

16bit	32bit	P	FNC 58	PWM	脉宽调制
✔					
7 步			指令格式：PWM (S1) (S2) (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
(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才适合使用该指令。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；

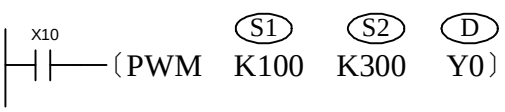
(S2)为设定的脉冲输出周期，必须有(S1)≤(S2)，设定范围为1~32，767ms；

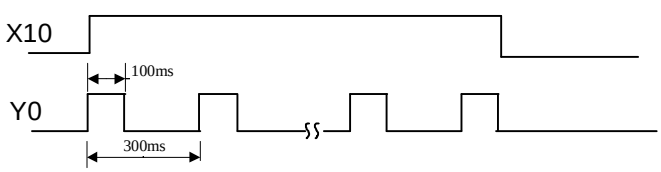
(D)为脉冲输出端口，对于3624MT/2416MT型只能指定Y0或Y1；其他MT型可以指定Y0/Y1/Y2；而MTQ型则可指定Y0/Y1/Y2/Y3/Y4，不要与PLSY，PLSR指令的输出端口重复。

本指令是以中断方式执行，当指令能流为OFF时，输出停止。

(S1)、(S2)可在PWM指令执行时更改。

指令举例：





16bit	32bit	P	FNC 59	PLSR	带加减速脉冲输出
✔	✔				
9 步	17 步		指令格式：PLSR (S1) (S2) (S3) (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
(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才适合使用该指令。该功能是指带加减速功能的固定尺寸传送用脉冲输出指令。其中：

(S1)为设定的输出脉冲的最高频率，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，范围50~5000(ms)，减速时间与加速时间相同，ms单位，设定时请注意：（新版本的H2U系列PLC中减速时间可单独设定，请参考后面的介绍）

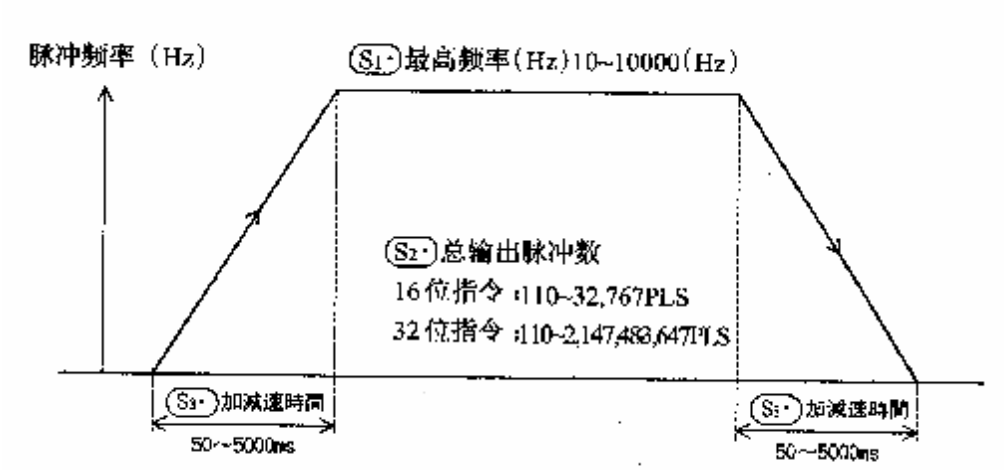
(D)为脉冲输出端口，3624MT/2416MT型只能指定Y0或Y1；1616MT/3232MT型能指定Y0/Y1/Y2；而MTQ型则可选Y0/Y1/Y2/Y3/Y4，不要与PLSY指令的输出端口重复。

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，将减速停止；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；
- 40 点 MT、60 点 MT 型 PLC 可以同时使用 2 个 PLSR 指令，或 2 个 PLSY 指令，分别对应 Y0、Y1 端口；其他点数的 MT 型 PLC 可以同时使用 3 个 PLSR 指令，或 3 个 PLSY 指令，分别对应 Y0、Y1、Y2 端口；MTQ 型 PLC 可以同时使用 5 个 PLSR 指令，或 5 个 PLSY 指令，分别对应 Y0、Y1、Y2、Y3、Y4 端口；具体规格请参考可编程控制器硬件用户手册。
- 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；Y2结束时M8149＝0；Y3结束时M8150＝0；Y4结束时M8151＝0）后，再延迟1个扫描周期，方可再启动该指令（在新版本的H2U系列PLC中通过设置可以不受此限制，具体请参考附录5.8中的说明）；

指令举例：



各输出端口对应的特殊寄存器如下：

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		
D8150	低字	PLSY 或 PLSR 指令中设定的输出至 Y2 口的脉冲总数（3624MT/2416MT 没有此端口）	
D8151	高字		
D8152	低字	PLSY 或 PLSR 指令中设定的输出至 Y3 口的脉冲总数（只适合于 MTQ）	
D8153	高字		
D8154	低字	PLSY 或 PLSR 指令中设定的输出至 Y4 口的脉冲总数（只适合于 MTQ）	
D8155	高字		
D8136	低字	已向 Y0 及 Y1 输出的脉冲个数的累计值	
D8137	高字		

I　本指令的输出频率范围为10~100，000Hz。最高速度或加减速的高速速度转换超出该范围时，将自动转换（上升或下降）至范围内的数值后执行。但是，实际能够输出的输

出频率最低值取决于以公式：

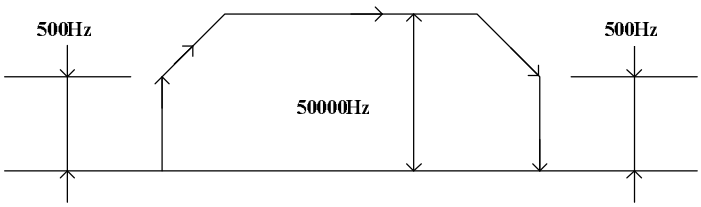
$$\sqrt{\text{最高频率}\ (S1.)\ \text{Hz} \div (2 \times (\text{加减速时间}\ (S3.)\ \text{Ms} \div 1000))} = \text{输出脉冲频率的最低频率数}$$

加速初期和减速末期的频率不得低于上述公式的计算结果。

［例］最高速度：50000Hz 加减速时间：l00ms

$$\sqrt{50000 \div (2 \times (100 \div 1000))} = 500\text{Hz}$$

最高频率 S1.指定为 50000Hz 时。加速初期和减速末期的实际输出频率为 500Hz。



注意：

在新版本的H2U系列PLC中，PLSR，DRVI，DRVA等指令的功能有增强；

1. 通过使用特殊位M8135~M8139(分别对应Y0~Y4)为ON，可以实现在运行中更改输出脉冲个数(可大可小)；同时减速时间分别由如下寄存器定义：

端口号	使用特殊位	修改减速时间用寄存器
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167
Y003	M8138	D8168
Y004	M8139	D8169

2. 通过使用特殊位M8085~M8089(分别对应Y0~Y4)为ON，可以实现如下功能：

在驱动特殊位为ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理；

3. 通过使用特殊位M8090~M8094(分别对应Y0~Y4)为ON，可以实现如下功能：

可以实现脉冲输出完成中断；具体对应如下：

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

具体可参考附录5.8中的说明。

4.3.2.7 方便指令（60~69）

16bit	32bit	P	FNC	IST	状态初始化
✔			60		
7 步			指令格式：IST (S) (D1) (D2)		

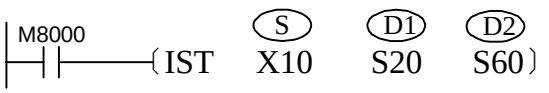
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)	✔	✔	✔												
(D1)				✔											
(D2)				✔											

注：1）该指令在用户程序中只能使用 1 次；

2）(D1) 及 (D2) 只能用 S 变量 S20~S899；且必须 (D1) < (D2)

3）使用该指令时还用到系统专用的 M 变量。

该应用指令可实现一个典型的多个动作循环执行机构的控制状态初始化定义，并指定运行模式的输入信号、动作状态等变量。



(S) 为指定运行模式的输入的起始位变量元件地址，本指令会占用 (S) ～ (S) ＋7的连续8个地址单元，且每个变量的对应特定功能定义，见以下描述；

S　　X10：手动操作　　X14：　连续运行

X11：原点回归　　X15：原点回归启动

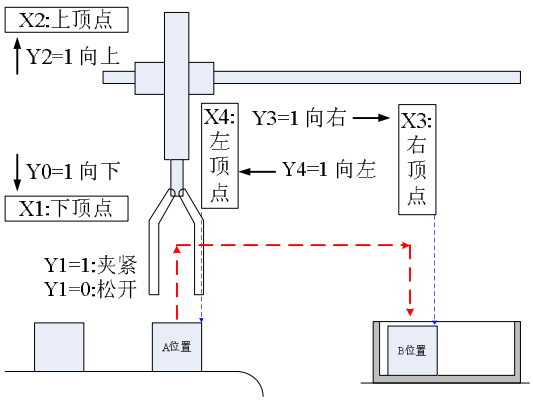
X12：步进　　X16：自动运行启动

X13：一次循环　　X17：停止

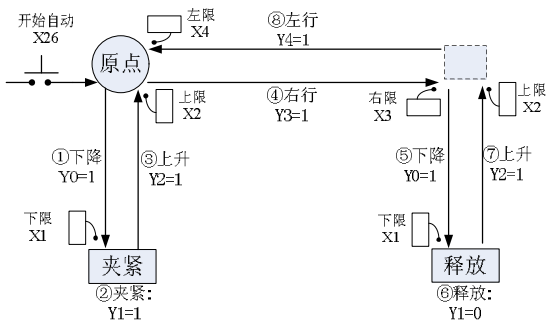
(D1) 为指定自动操作模式中，使用S状态的最小序号；

(D2) 为指定自动操作模式中，使用S状态的最大序号；

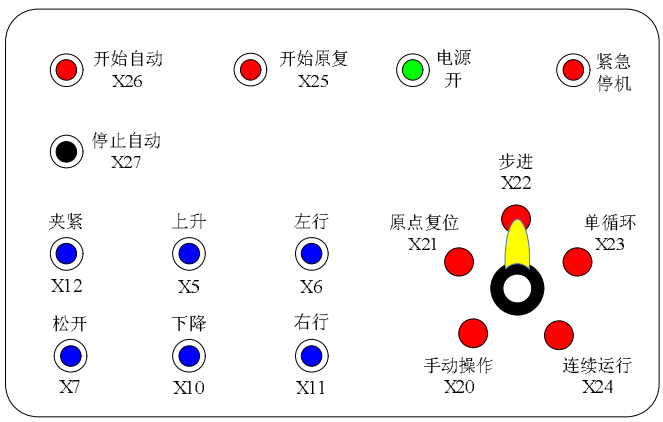
例如下图例系统，执行机构的动作依次是，抓取机构由原点下降到A工件位置，将工件抓取后提升到指定高度，再平移到指定位置后下降，到达指定位置后释放夹具，之后后循原路返回，开始下一个循环动作。IST指令可指定该操作机构的控制信号输入、状态转移的控制等，实现自动控制，还支持手动调试的单步动作、原点复位等。



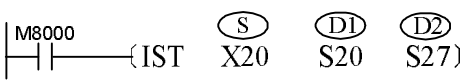
信号端口的分配及动作顺序分析如下：



操作机构还需配备指令按键及状态切换开关，用手动试机、单次动作、循环动作等控制，以下是操作面板示意图，其中有按键端口及其功能分配：



对于如上图的应用，每个完整的循环动作分 8 个步骤（即 8 个状态），可使用如下的指令语句，实现控制系统的状态初始化：



其中Ⓢ指定 X20 为运行模式的起始输入，因此随后地址的 X21～X27 输入端口也将被使用，且功能动作特性将分别定义为：（使用其他 X、M 或 Y 等变量时同理类推）

X20：手动操作模式，用单个按钮接通或切断各控制输出信号；

X21：原点复归模式，按下原点复归用按钮时，使机械自动复归原点

X22：单步操作模式，每次按起动按钮，前进一个工序

X23：循环运行一次模式，在原点位置上按起动按钮时，进行一次循环的自动运行并在原点停止。途中按停止按钮，其工作停止，若再按起动按钮，在此继续动作至原点自动停止。

X24：连续运行模式，在原点位置上按起动按钮，开始连续运行。若按停止按钮，则运转至原点位置后停止。

X25：开始原点复归命令信号

X26：开始自动命令信号

X27：停止自动命令信号

注意，这些端口信号中 X20～X24 决定了控制系统的运行模式，不能同时有为 ON 的状态，故建议采用旋转开关进行信号的选择切换。

D1和D2分别指定自动操作模式中，使用状态的最小和最大序号 S20~S27，共 8 个状态。

还需注意 IST 指令对如下的特殊变量的定义及使用要求：

❙ 如果驱动 IST 指令，下列元件被自动切换控制，用户程序可以进行引用；为了配合 IST 指令的状态切换和控制，用户程序中还需对部分特殊变量进行操作，如下表说明：

IST 指令内部默认使用的变量		用户程序中驱动的变量	
M8040	1＝禁止所有状态的转移。	M8043	1＝原复完毕。在原复模式中，机器返回原点后，通过用户程序将该特殊 M 变量置位。
M8041	1＝转移开始	M8044	1＝原点条件。检测机器的原点条件，驱动该特殊辅助继电器。在全部模式中均置位。
M8042	1＝起动脉冲	M8045	1＝全部输出复位禁止。若在手动、原复、自动模式间进行转换时，机器不在原点位置时，则执行全部输出与动作状态的复位。但若已驱动 M8045，只有动作状态复位。
S0	手动操作初始状态	M8047	1＝STL 监控有效。驱动 M8047 后，现正动作中的状态序号（S0~S899）按从小到大的顺序存入特殊辅助继电器 D8040~D8047 中。由此，可以监控 8 点的动作状态序号。此外，若这些状态的任何一个动作，特殊辅助继电器 M8046 将动作。
S1	原点复归初始状态		
S2	自动运行初始状态		

❙ 在机器“自动运行”模式中，可进行自由转换：单步↔循环一次↔连续运行；

❙ 在机器运行中，在“手动操作”↔“原点复归”↔“自动运行”之间进行转换时，当全部输出复位后，转换后模式有效。（M8045 驱动时不能复位）

❙ 使用 IST 指令时，S10~S19 可作为原点复归用。因此，在编程中请勿将这些状态作为普通状态使用。另外，S0~S9 作为初始状态处理，S0~S2 作为如上述的手动操作，原点复归以及自动运行使用，而 S3~S9 则可以自由地使用。

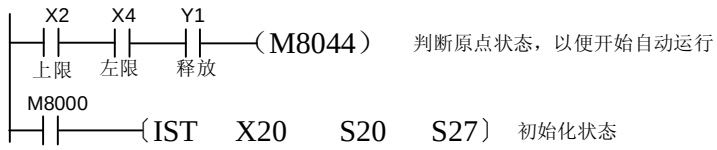
❙ 编程时，IST 指令必须比状态 S0~S2 等一系列的 STL 电路优先编程。

❙ 为了防止其中的 X20~X24 同时为 ON 状态，须用旋转开关。

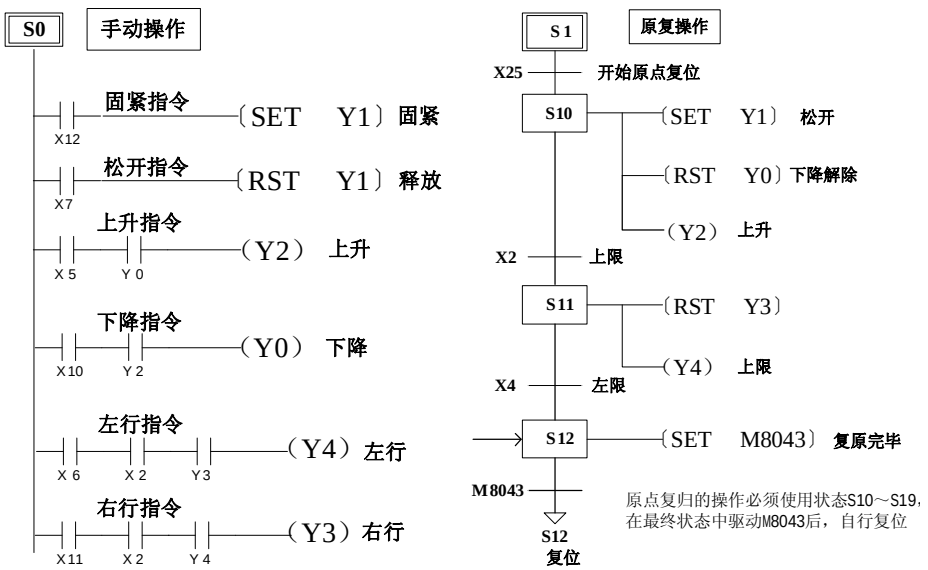
l 原点复归完成（M8043）未动作时，如果在各个（X20），原点复归（X21），自动（X22、X23、X24）之间进行切换时，则所有输出进入 OFF 状态。并且，自动运行在原点复归结束后，才可以再次驱动。

在用 IST 指令进行了控制指令的初始化之后，还需要对执行机构每个状态的动作，以及状态转移的条件进行编程，具体说明如下：

1. 系统初始化：定义原点复位条件；定义 IST 指令中使用的运行模式信号的输入端口、循环动作的状态变量。使用的程序语句如下图。

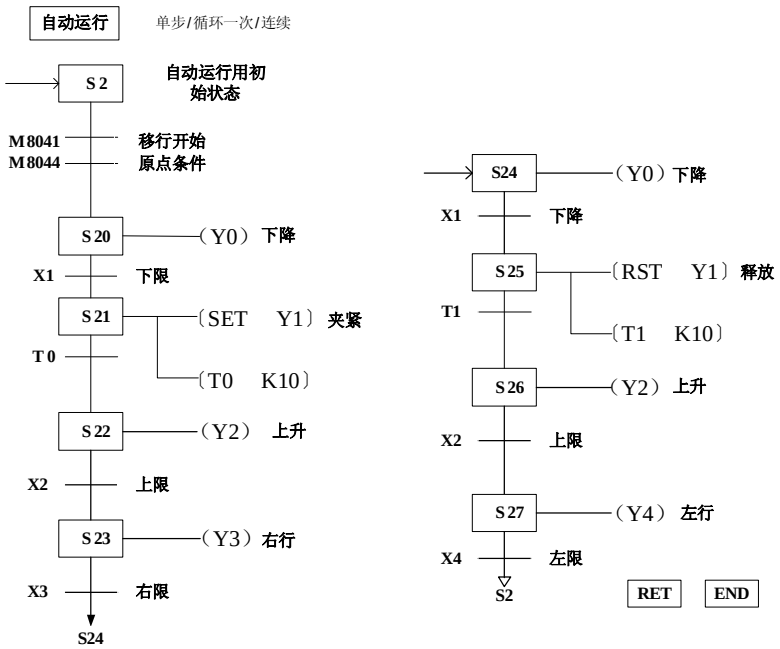


2. 手动操作：由操作盘上定义的命令信号驱动执行，见如下图中 S0 状态的程序语句。若没有手动模式，可以不需编写该部分的程序：



3. 原点复位：根据开始复位的命令信号、复位动作的顺序，设计复位程序，如上右图。

4. 自动运行：根据要求的动作条件和顺序、控制信号输出，编写程序如下图：

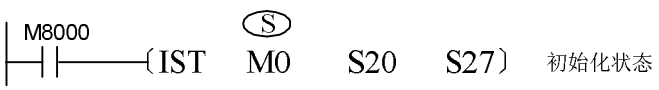


至此，控制系统就可以按前面的动作要求完成循环动作了。以上是为了方便阅读，采用了步进指令进行的编程说明，用户完全可以用等效梯形图来进行编程。

当一个控制系统的“自动运行”模式有不同的状态数，可参考上例编程，修订 D1 及 D2 的设置项，并在“自动运行”状态中进行相应的处理。

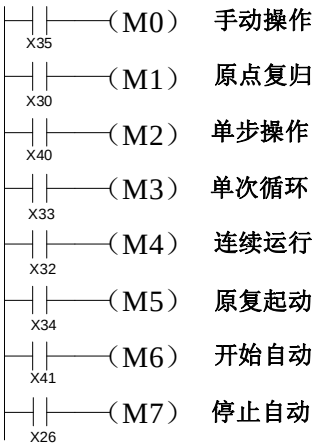
I 对于使用非连续 X 输入端的处理方法：

若需要采用非连续地址的 X 输入端口作为运行模式的给定输入，可采用 M 变量来进行“过渡”传送，即先通过简单的 OUT 指令将非连续的 X 输入端状态，逐个复制到某个连续地址的 M 变量，而使用如下指令：

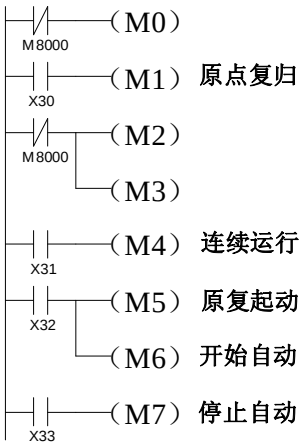


在 IST 中的 S 指定到该连续的 M0~M7 变量区。对于不存在的控制模式，也可用编程指令进行屏蔽，举例如下图中的 X 作为模式输入端与 M 变量的对应关系；对于不需要的模式输入 M 变量，只需将之固定清 0 就可以了：

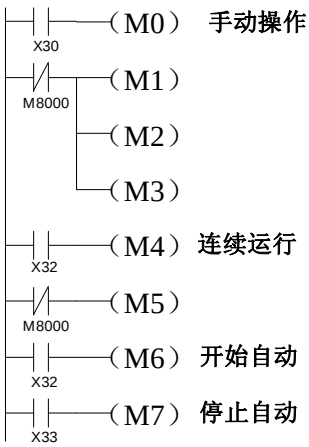
X输入端口不连续，采用连续M变量过渡的处理方法：



没有手动模式的处理方法：



仅有手动/连续模式的处理方法：



16bit	32bit	P	FNC	SER	数据查找
✔	✔	✔	61		
9 步	17 步		指令格式：SER (S1) (S2) (D) (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
(S1)							✔	✔	✔	✔	✔	✔	✔	✔	✔
(S2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔
(n)					✔	✔							✔		
	取值范围：16bit 指令：n＝1～256；32bit 指令：n＝1～128														

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；

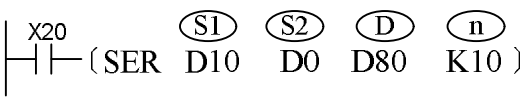
(S2) 为待检索的数据；

(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。

指令举例：



(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 (D0) =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Ⓓ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)		

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；

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；

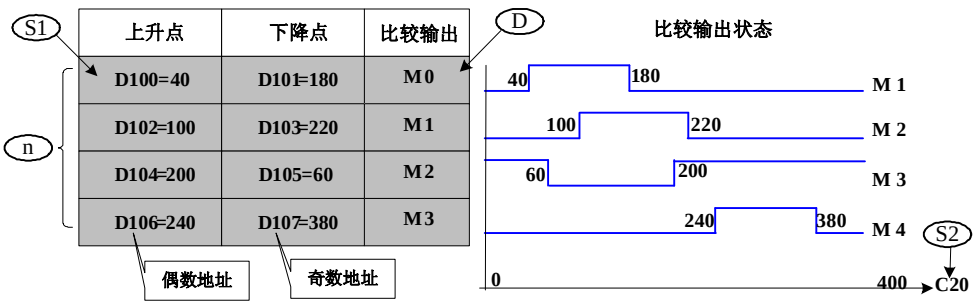
(n) 为多段比较数据的组数。

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。

指令举例：

```
X10                 (S1)   (S2)   (D)    (n)
 ┤├〔ABSD   D100    C20    M0     K4 〕
M8013
 ┤├〔C20    K400〕
X11
 ┤├──────〔RST     C20〕
```

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 (D) 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)		

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；															

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。

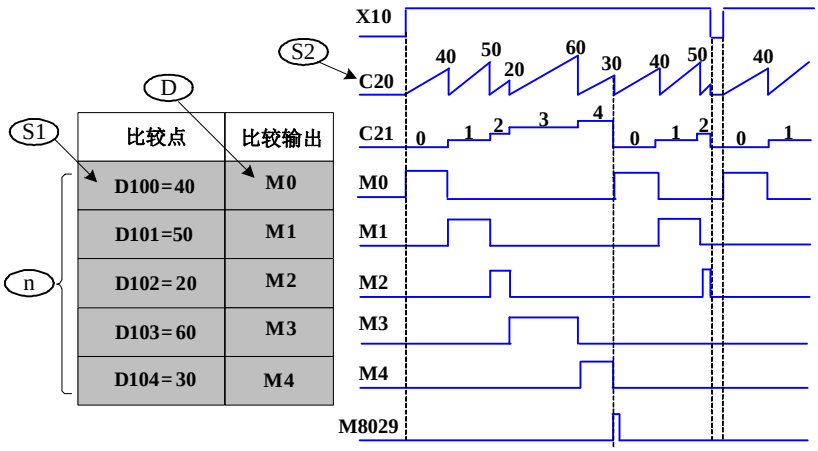
当使用32bit指令时，(S1) (S2) (D) 均指向32bit变量，(n) 也按32bit变量宽度进行计算。

n 的组数比较完成时，指令执行完毕标志M8029会置On。

指令举例：

```
X10                (S1)   (S2)   (D)    (n)
 ┤├──〔INCD    D100    C20    M0     K5 〕
M8013
 ┤├──〔C20     K400〕
```

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：

- 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	TTMR	演示定时器
✔			64		
5 步			指令格式：TTMR (D) (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
(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)为按键保持时间以秒为单位乘以n倍数后的乘积，按键释放后(D)内容没有变化；而(D)＋1单元则用于保存按键的按压时间，按键释放后(D)＋1的内容被复位为0，(D)＋1的时间单位为100ms；

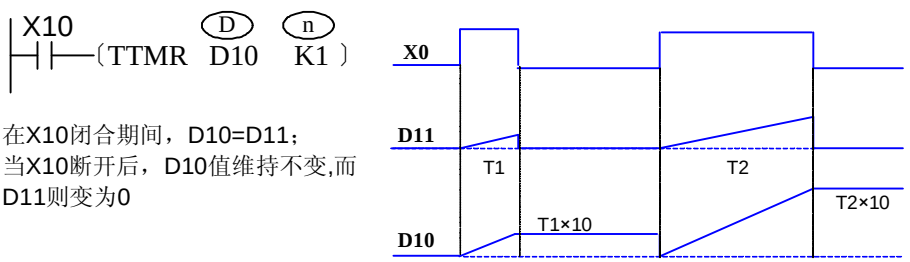
(n)为倍数设定，注意实际倍数为10^n计算方式，(n=0~2)

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。

指令举例一：

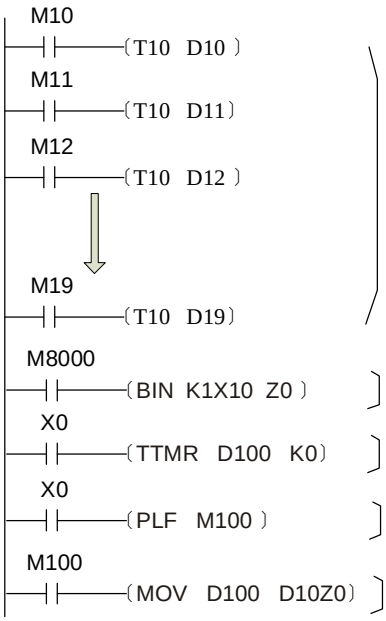


在X10闭合期间，D10=D11；
当X10断开后，D10值维持不变,而D11则变为0

假如X10的按键保持时间为T秒，D10、D11与n三者之间的关系如下：

n	D10	D11(单位：100毫秒)
K0(单位：秒)	1×T	D11=D10×10
K1(单位：100毫秒)	10×T	D11=D10
K2(单位：10毫秒)	100×T	D11=D10/10

指令举例二：



使用TMR指令写入10组设置时间,将设置值预先写入D10~D19,该组定时器为100ms型定时器，因此示教数据的1/10为实际动作时间（秒）

将1位数指拨开关接于X10~X13，使用BIN指令将指拨开关的设置值转换成BIN值并存放于Z0 当中

X0 的On 时间(秒)存放于D100 当中

M100为演示定时器按钮X0释放产生的一次扫描周期脉冲

以指拨开关的设置号码当成间接指定的指针，然后将D100 的内容传送至D10Z0(D10~D19)当中

16bit	32bit	P	FNC 65	STMR	特殊定时器
✔					
7 步			指令格式：STMR (S) (m) (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)											✔				
(m)	常数，m＝1～32767														
(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；

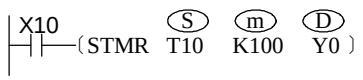
(m)为延时设定值，单位为100ms，设定值范围为K1～K32767；

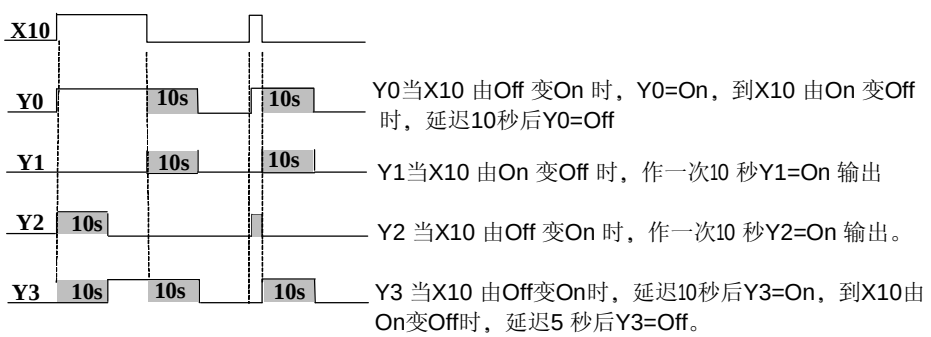
(D)为延时动作输出元件的起始编号，共占用4个连续单元。

使用注意事项：

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。

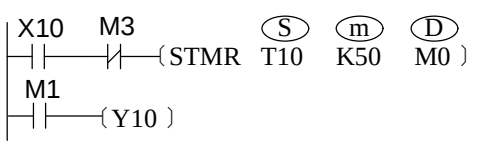
指令举例一：

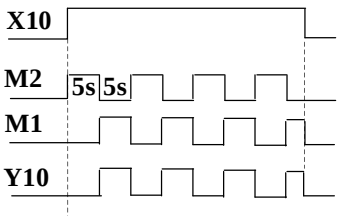




指令举例二：

若在指令能流中引入(D)的元件，可方便地实现振荡器输出（此功能也可以用ALT指令来实现），如下图：


X10
M3
S
m
D
STMR
T10
K50
M0
M1
Y10


X10
M2
5s
5s
M1
Y10

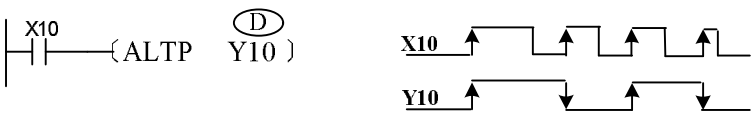
16bit	32bit	P	FNC	ALT	ON/OFF 交替
✔		✔	66		
3 步		3 步	指令格式：ALT (D)		

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

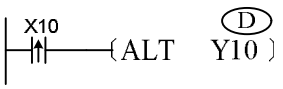
该指令的功能是能流有效时，将(D)元件的状态反转。其中(D)为位变量元件。

一般使用脉冲执行型ALTP指令。

指令举例一：

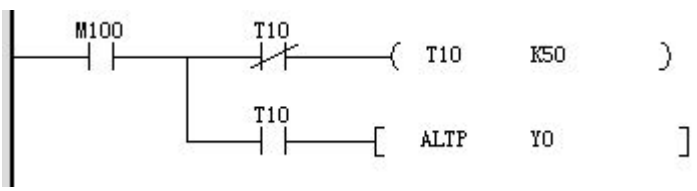


如下指令产生的动作与之相同：



指令举例二：

若在指令能流中引入定时器，可方便地实现振荡器输出（此功能也可以用特殊定时器STMR指令来实现），如下图：



16bit	32bit	P	FNC 67	RAMP	斜坡指令
✔					
9 步			指令格式：RAMP (S1) (S2) (D) (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
(S1)													✔		
(S2)													✔		
(D)													✔		
(n)	常数，1～32767														

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：

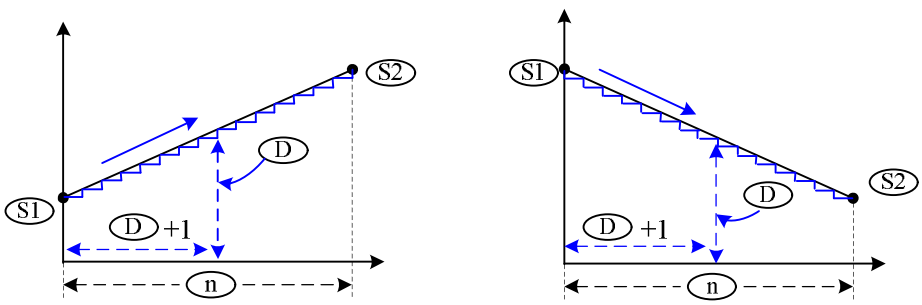
(S1) 斜坡信号的起始值单元；

(S2) 斜坡信号的终点值单元；

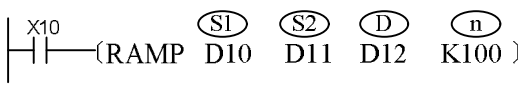
(D) 为线性插值信号的过程值存放单元，而插值次数的计时器存放在 (D) +1单元；

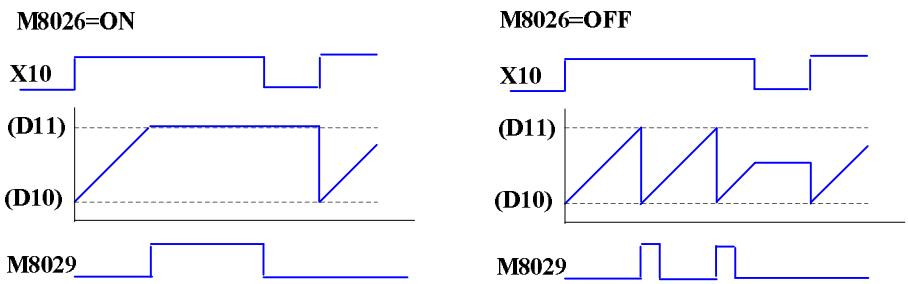
(n) 完成插补过程的程序扫描执行次数。由于插值输出是在正常主循环中进行的，为了保证插值输出的线性，需要将程序执行设置为固定扫描方式（见M8039、D8039说明）。

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ON。执行特点如下例：




M8026=ON
X10
(D11)
(D10)
M8029
M8026=OFF
X10
(D11)
(D10)
M8029

16bit	32bit	P	FNC 68	ROTC	旋转台控制指令
✔					
9 步			指令格式：ROTC (S) (m1) (m2) (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)		✔	✔	✔											
(m1)	2～32767，m1≧m2														
(m2)	0～32767，m1≧m2														

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：

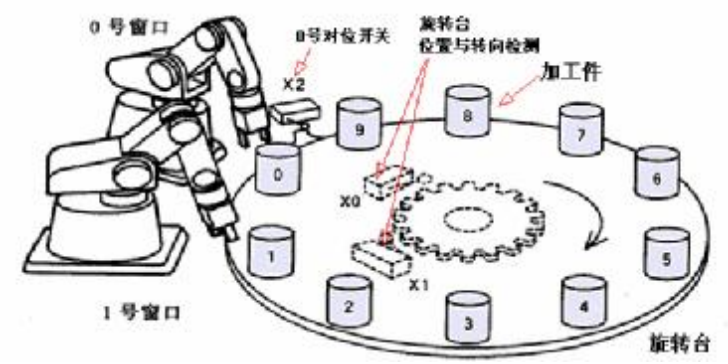
(S) 计数变量的起始单元；

(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)；

(D) 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。

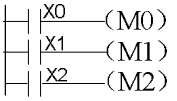
信号配置方式如下图，图中X0、X1分别接AB正交编码器的A相和B相输出信号，也可采用机械开关得到正交相位的信号；X2接用于0号工位的检测输入（当旋转到0号工位时为ON状态），由此3个信号即可检测旋转工作台的当前转动速度和转向和工位。



应用举例：

```
X10
 ┤├──〔ROTC  D200   K10   K2    M0 〕
          (S)   (m1)  (m2)  (D)
```

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	当X10为ON时，可以自动得到M3~M7的结果； 当X10为OFF时，M3~M7均为OFF；
M4	低速正转	
M5	停止	
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
✔					
11 步			指令格式：SORT (S) (m1) (m2) (D) (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
(S)													✔		
(m1)	常数，1～32														
(m2)	常数，1～6														
(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；

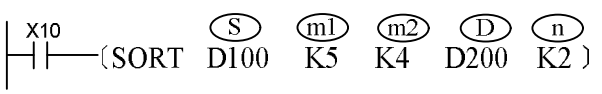
(m2)数组的列数，或称每条记录的栏目数；

(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。

其中(n)的值在1～(m2)范围。

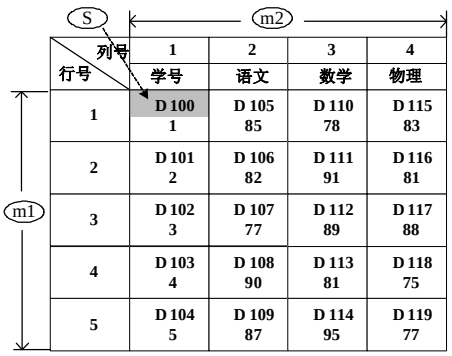
指令举例：



当X10＝ON时，开始排序运算，指令执行完毕，M8029被置ON；

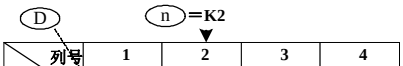
若需要再次排序，需将X10＝OFF一次。

上述指令的等效表格及其数据举例：



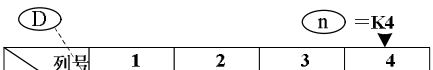
行号 \ 列号	1	2	3	4
	学号	语文	数学	物理
1	D 100 1	D 105 85	D 110 78	D 115 83
2	D 101 2	D 106 82	D 111 91	D 116 81
3	D 102 3	D 107 77	D 112 89	D 117 88
4	D 103 4	D 108 90	D 113 81	D 118 75
5	D 104 5	D 109 87	D 114 95	D 119 77

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

4.3.2.8 外围设备 I/O（70~79）

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)		

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)		✔	✔	✔											

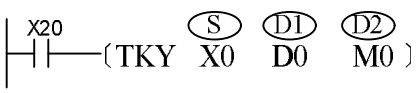
该指令是指定10个连续的位变量单元（如X输入端口），依次代表10进制的0～9按键，当有按键动作（状态为ON）时，以按键的动作顺序，可输入十进制0～9999的4位数值；若采用32bit指令，则可输入十进制0～99，999，999的8位数值。其中：

(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）；

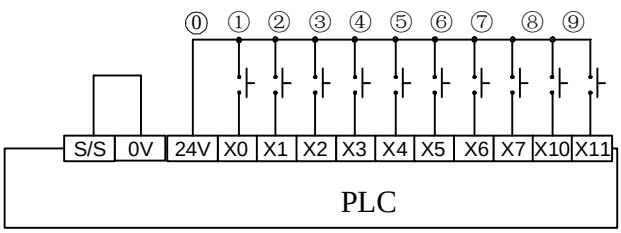
(D1) 为已输入数值的存放单元；

(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。

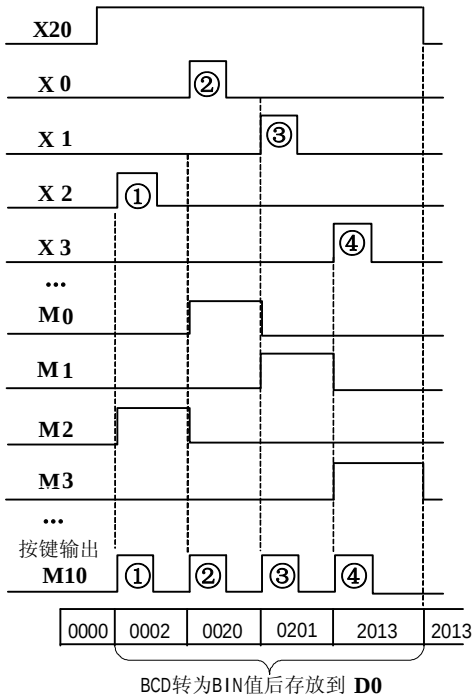
指令举例：



对应的硬件接线如下图：



若要输入数字“2013”，按顺序按键②⓪①③（X2、X0、X1、X3）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 4 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)		

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)										✔	✔	✔	✔	✔	✔
(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：

(S) 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；

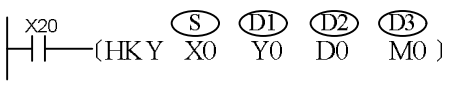
(D1) 为按键的扫描输出Y端口的起始端口号，使用随后的共4个Y端口；

(D2) 为按键输入值存放单元，0~9999；当为32bit指令时可输入0～99，999，999范围内的数值；

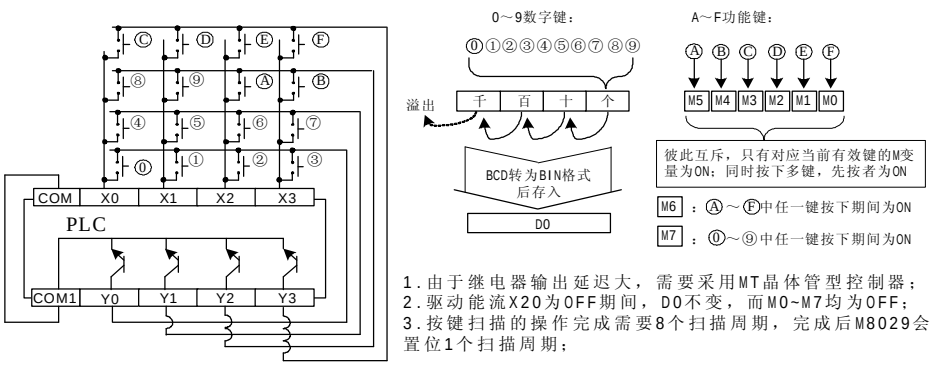
(D3) 为按键输入状态起始单元地址，共占用连续的8个位变量单元。

此指令只能用于晶体管输出型PLC。

指令举例：



对应的接线图及参数响应说明如下：



1.由于继电器输出延迟大，需要采用MT晶体管型控制器；
2.驱动能流X20为OFF期间，D0不变，而M0~M7均为OFF；
3.按键扫描的操作完成需要8个扫描周期，完成后M8029会置位1个扫描周期；

❙ 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)		

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														

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：

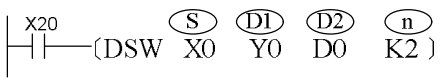
(S)为按键的扫描输入X端口的起始端口号，若(n)=1，使用随后的共4个X端口；若(n)=2，则使用随后的共8个X端口；

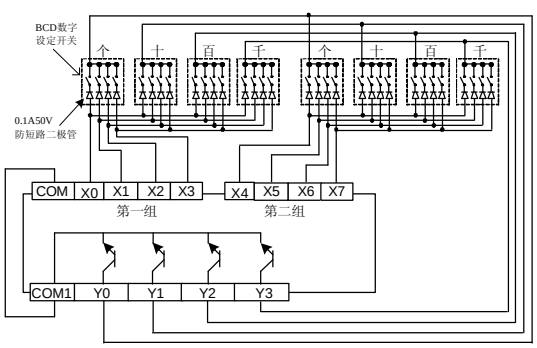
(D1)为按键的扫描输出Y端口的起始端口号，使用随后的共4个Y端口；

(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。

指令举例：



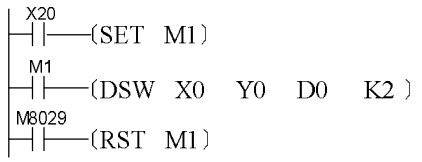


当X20=ON时，执行扫描读取数字开关设定的操作：
1. 第一组数字开关的设定值，经BIN转换后存入D0;
2. 第二组数字开关的设定值，经BIN转换后存入D1;
3. 一次读取循环完毕，M8029会置位一个扫描周期。

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	SEGD	七段码译码
✔		✔	73		
5 步		5 步	指令格式：SEGD (S) (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)								✔	✔	✔	✔	✔	✔	✔	✔

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：

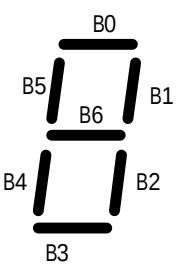
(S)为待译码的数据源（取BIN内容的最低四个位b0~b3）；

(D)为译码后存放7段码的变量。

指令举例：

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 HEX数	数据 BIN数	数码管组合	内部译码表值 B7	B6	B5	B4	B3	B2	B1	B0	译码后字符
0	0000		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

每位对应一个笔段
1＝笔段点亮
0＝笔段熄灭

16bit	32bit	P	FNC 74	SEGL	七段码扫描显示
✔					
7 步			指令格式：SEGL (S) (D) (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
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)		✔													
(n)	常数，n＝0~7														

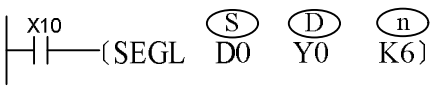
该指令是利用8个或12个Y端口，用于4位或8位七段锁存数码管的显示驱动，显示方式为扫描驱动方式。其中：

(S) 为待显示的数据，其值在BCD转换后才送到数码管进行显示；

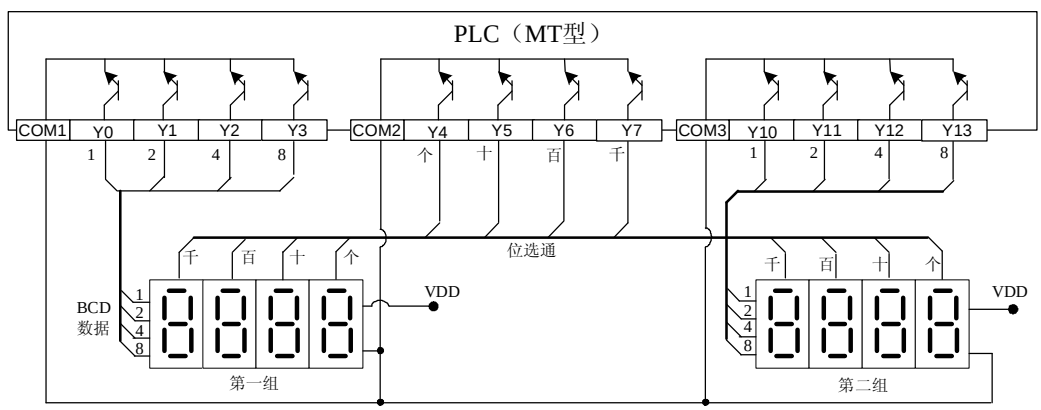
(D) 为显示驱动用的Y端口起始地址号；

(n) 为根据显示数据的组数、信号正负逻辑等相关的设定值，见下面的详细描述。

指令举例：



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

端口会自动循环扫描，每次只有1个端口为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。范例中若D0=K2468，D1=K9753，则第一组将会显示2 4 6 8，第二组显示9 7 5 3。

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ON。

(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：

一组 4 位数的时候 n=0~3。二组 4 位数的时候 n=4~7

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

PLC的晶体管输出极性与7段显示器的输入极性是否相同或者是不同时，可透过参数n的设置值来相互匹配

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。

使用说明：

❙ 由于继电器不适合较高频率的扫描输出动作，晶体管输出型PLC才能使用该指令。

16bit	32bit	P	FNC 75	ARWS	方向开关
✔					
9 步			指令格式：ARWS (S) (D1) (D2) (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
(S)	✔	✔	✔	✔											
(D1)											✔	✔	✔	✔	✔
(D2)		✔													
(n)	常数，n＝0~3														

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：

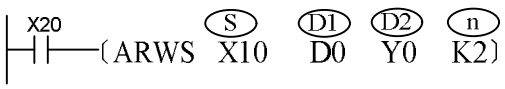
(S)为指定按键输入的起始地址，占用后续的共4个位单元；

(D1)用于被显示及修改的变量，只能显示一个16bit宽度的变量；

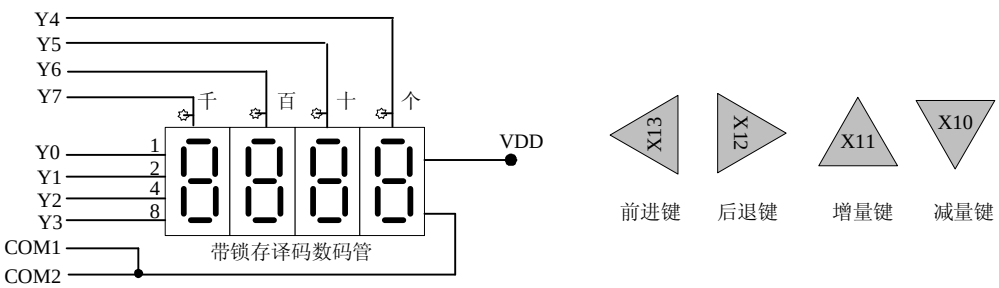
(D2)用作数码管显示驱动的Y端口起始地址，占用后续的共8个Y端口；

(n)为信号逻辑的设定值，参见前面SEGL指令中关于(n)的详细描述。

指令举例：



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

（YO04~YOO7）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。

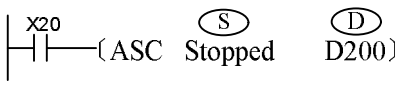
16bit	32bit	P	FNC 76	ASC	ASCII 转换
✔					
11 步			指令格式：ASC (S) (D)		

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)											✔	✔	✔		

(S) 为欲执行ASCII码变换的英文字母，由计算机输入，最大允许长度为8个字符；

(D) 为存放ASCII码的起始单元号，占用随后的共4个（M8161=0）或8个变量单元（M8161=1）。

指令举例：



当X20＝ON时，D200～D2003的赋值如右图

	高字节	低字节
D200	54（t）	53（S）
D201	50（p）	4F（o）
D202	45（e）	50（p）
D203	20	44（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	54（t）
D202	00	4F（o）
D203	00	50（p）
D204	00	50（p）
D205	00	45（e）
D206	00	44（d）
D207	00	20

附：《ASCII代码对照表 》

10 进制位	ASCII (16 进制数)
0	30
1	31
2	32
3	33
4	34
5	35

代码	ASCII (16 进制数)
STX	02
ETX	03

10 进制位	ASCII (16 进制数)
6	36
7	37
8	38
9	39

英语字母	ASCII (16 进制数)	英语字母	ASCII (16 进制数)
A	41	N	4E
B	42	O	4F
C	43	P	50
D	44	Q	51
E	45	R	52
F	46	S	53
G	47	T	54
H	48	U	55
I	49	V	56
J	4A	W	57
K	4B	X	58
L	4C	Y	59
M	4D	Z	5A

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

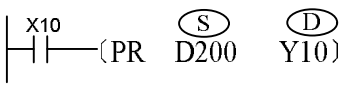
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)											✔	✔	✔		
(D)		✔													

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：

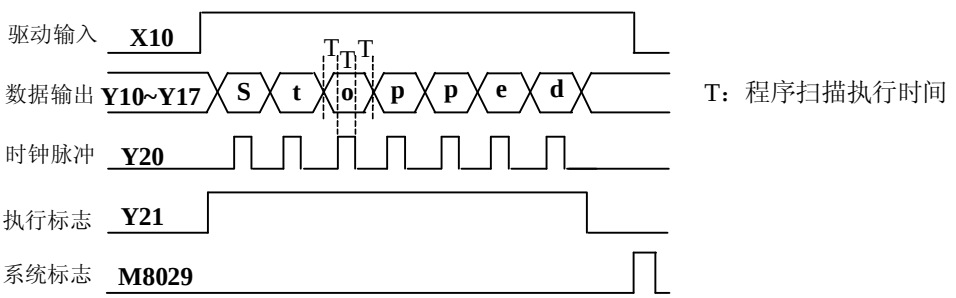
(S)为待输出的变量单元起始地址；

(D)为进行输出打印的的Y端口起始号。

指令举例：



若D200~D203中的ASCII码为“Stopped”，则对应的输出端口信号及其时序如下图：



使用说明：

l 必须使用晶体管输出型 PLC，才能完成该指令功能；

l 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；

l M8027=Off 时，最多可执行 8 个字的串行的输出；当 M8027=On 时，则可执行 1~16 个字的串行输出。

l M8027 为 Off 时：（1）遇到“00”的字符也继续往下执行；（2）能流无效后 M8029 不动作；

l M8027 为 ON 时：（1）打印输出过程中，遇到“00”的字符时，会自动结束打印操作，之后的文字不再处理；（2）M8029 完成标志在驱动能流信号无效后会置 ON；

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		指令格式：FROM (m1) (m2) (D) (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
(D)								✔	✔	✔	✔	✔	✔	✔	✔

m1=0～7；

m2=0～32767；

n=1～32767；

(D) 元件指定时，16bit 指令可用 K1~K4；32bit 指令可用 K1~K8；m1，m2，n 不支持字符装置 D 寄存器

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：

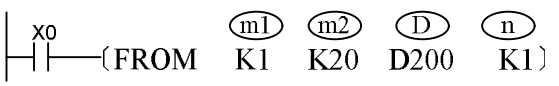
(m1) 为特殊扩展模块的地址编号，取值范围0～7，最靠近主模块的为0，依次编号，最多允许有8个特殊模块；

(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。

指令举例：



表示当X0为ON时，将#1号特殊模块中的第20号地址（16bit宽度）的内容读出到PLC的D200寄存器中，一次读取一笔（n=1）。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	TO	BFM 区写入
✔	✔	✔	79		
9 步	17 步		指令格式：TO (m1) (m2) (D) (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
(D)								✔	✔	✔	✔	✔	✔	✔	✔

m1=0～7；

m2=0～32767；

n=1～32767；

(D) 元件指定时，16bit 指令可用 K1~K4；32bit 指令可用 K1~K8；m1，m2，n 不支持字符装置 D 寄存器

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：

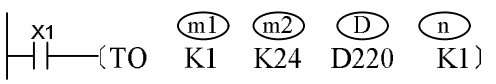
(m1) 为特殊扩展模块的地址编号，取值范围0～7，最靠近主模块的为0，依次编号，最多允许有8个特殊模块；

(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，占用随后的单元；

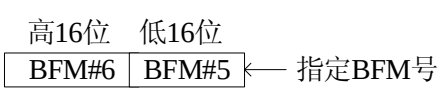
(n) 本次写入参数的笔数，取值范围1～32767，为按寄存器地址依次写入。

指令举例一：

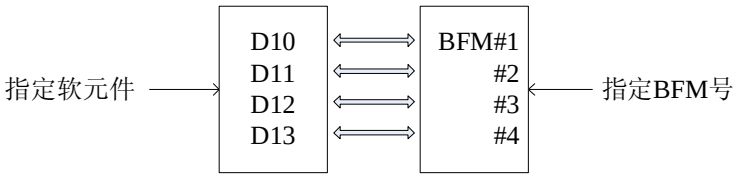


表示当X1为ON时，将PLC的D220寄存器中的数据，写入到#1号特殊模块中的第24号地址中，一次只写入1一笔。当X1为OFF时，不执行操作。

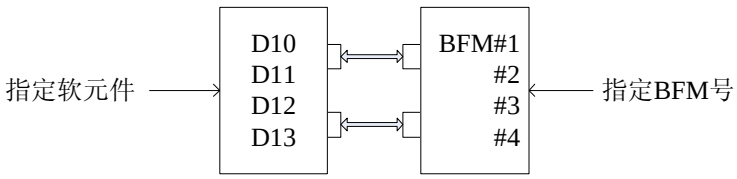
当用32bit指令时，(D) 指定的地址为低16bit地址，(D)＋1为高16bit地址。例如：



(n) 表示操作的参数个数，16bit 指令中，n＝2 表示 2Word；而 32bit 指令中，n＝1 表示 2Word，即 16 位指令的 n=2 与 32 位指令的 n=1 意义相同，请注意。例如：



16位指令 n=4时



32位指令 n=2时

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/T0指令的传送点数可变模式）

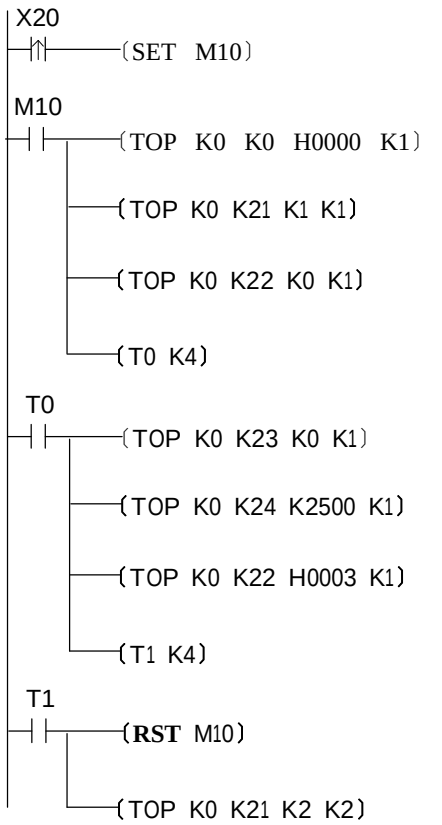
若M8164＝ON，执行FROM/T0指令时，特殊数据寄存器D8164（FROM/T0指令的传送点数指定寄存器）的内容作为传送点数n进行处理；

2. 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，PLC的扫描周期会延长。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WDT指令，或者错开FROM/T0指令的执行时间，或者用脉冲执行型指令。

3. 关于特殊模块的连接方法和输入输出编号等使用方法，请参阅各特殊模块附带的专用手册。

指令举例二：

对一个＃0 编号地址的 H2U-4AD 模块的 CH1 通道作偏移/增益设置操作，编程如下：



X20的闭合时引发4AD模块的偏移/增益调整操作

初始化4AD模块，4个通道均为－10V～10V信号方式

设BFM#21=1，允许偏移/增益调整操作

设BFM#22=0，清除偏移/增益调整

延时400ms

设BFM#23=0，设置偏移

设BFM#24=2500，设置增益

设BFM#22=H0003，将刚才设定的BFM#22/#23值存入CH1的EEPROM设置寄存器中

延时400ms

清除设置操作的命令标志

将BFM#21=2，禁止偏移/增益调整操作

4.3.2.9 外围 SER 设备（80~88）

16bit	32bit	P	FNC 80	RS	串行数据传送
✔					
9 步			指令格式：RS (S) (m) (D) (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
(S)													✔		
(m)					✔	✔							✔		
(D)													✔		
(n)					✔	✔							✔		

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。其中：

(S) 为待发送数据存放的寄存器区的起始地址；

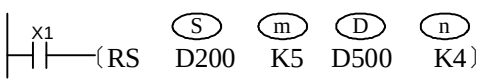
(m) 为待发送数数据的长度（字节数），取值范围0～256；

(D) 为通讯接收数据的存放寄存器区的起始地址；

(n) 本通讯接收的数据长度（字节数），取值范围0～256。

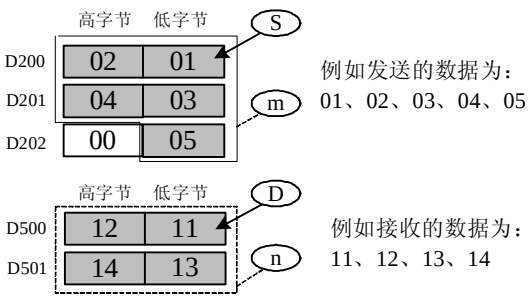
H2U系列对RS指令还有扩展功能，当选择了MODBUS主站协议，此时RS指令即为MOD指令功能，此功能在本指令后面介绍。

指令举例一：

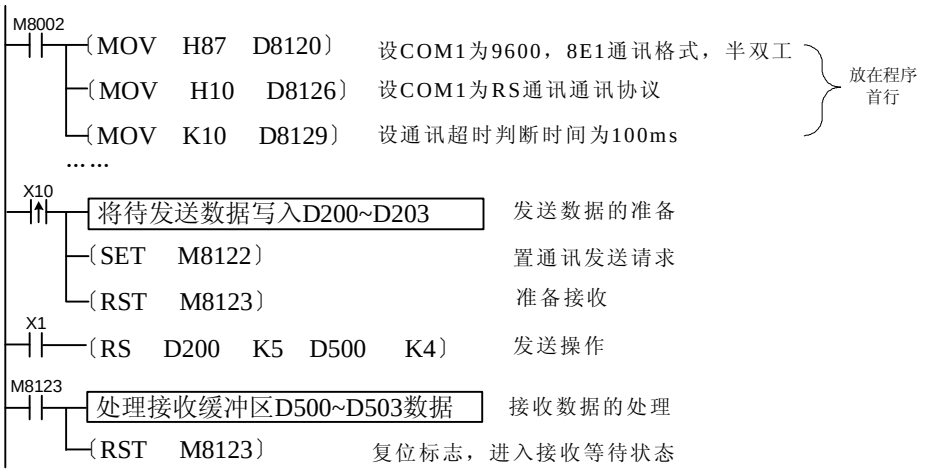


对于H2U系列PLC，RS指令只能用于COM1通讯口；COM0通讯口不支持RS指令。

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-300bps		
		0100b-600bps		
		0101b-1200bps		
		0110b-2400bps		
		0111b-4800bps		
		1000b-9600bps		
		1001b-19200bps		
		1010b-38400bps		
		1011b-57600bps		
		1100b-115200bps		
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；

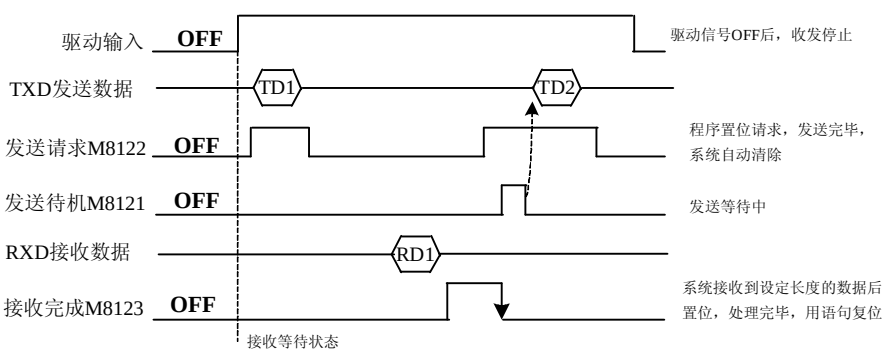
③ 并联从站协议；

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：

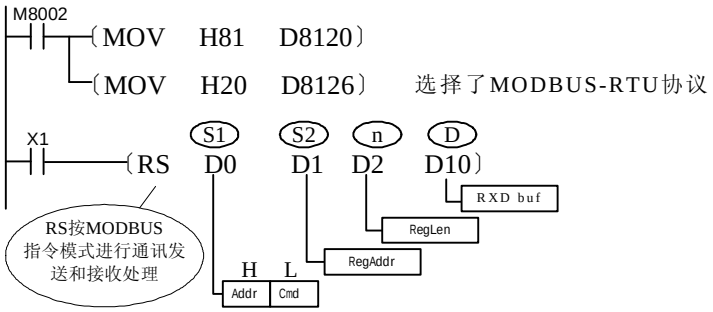
⑤ COM1口的通讯协议由D8126决定。一旦PLC运行后，协议将维持不变；

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（MODBUS主站协议通讯）：

当COM1口的通讯协议配置中选择了MODBUS主站协议（将D8126设定为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)为访问从站的寄存器起始地址号；

(n)欲读或写的数据长度；

(D)为读或写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)									✔		

MODBUS指令有两种，一种符合MODBUS RTU 协议，一种符合MODBUS ASC 协议，用哪种协议由从站所支持的协议格式定，若从站两种协议都支持而用户要求较快速的通信，建议选用RTU协议。两种协议只是通信格式不一样，对用户编程都一样，下面仅就RTU协议做说明。

指令格式：RS(ADDR&CMD，REGADDR，REGLEN，DATABUF)

ADDR&CMD：从机地址和MODBUS功能码，高8位表示从机地址，即目标设备地址，低8位表示

MODBUS功能码，由标准MODBUS协议定义，目前支持0x01，0x02，0x03，0x04，0x05

0x06，0x0f，0x10。具体含义请参照标准MODBUS协议或目标设备MODBUS协议。

REGADDR：所要读或写的从机线圈（1位）或寄存器（16位）地址，取值参考从机MODBUS协议。可为软元件或常数

REGLEN：所要读写的从机线圈或寄存器个数

DATABUF：主机用于存放数据的起始寄存器，即数据缓冲区。缓冲区长度与REGLEN

相关，至少取1。若MODBUS命令为读，指令成功执行完后，把从机数据读到缓冲区中，

若MODBUS命令为写，把缓冲区发送给从机。用户在设计程序时需要计算缓冲区长度，预留足够的寄存器作缓冲区。

相关状态标志

M8122：MODBUS指令执行状态指示，OFF时表示指令执行完毕，ON时为执行中。若M8122为OFF，且指令在一个扫描周期内能流有效，M8122置为ON，系统将会把指令参数记录下来，转入后台执行该指令的通信要求。通信执行完后，当再次运行到此指令的位置时，无论该指令能流是否有效，均会把M8122复位为OFF，立即扫描下一条能流有效的指令，记录指令参数并转入后台执行该指令的通信要求。

M8123：指令通信情况指示，ON表示通信异常，OFF表示通信正常

M8063：指令错误指示，错误码存于D8063。

D8063错误码如下

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

6336：缓冲区溢出

6337：帧错误

指令举例二：

该指令用于具有MODBUS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，指令举例二中，PLC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　　PLC数据缓冲区，本例中读命令通信成功后数据存于D10

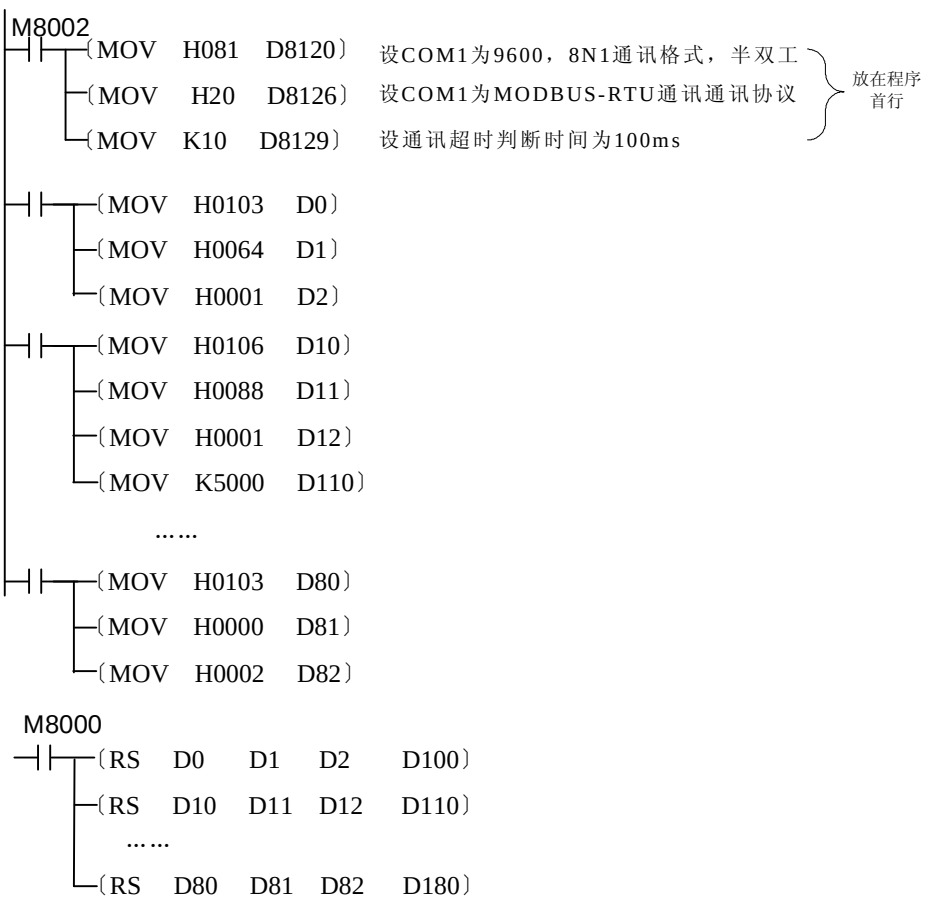
通讯发送、接收过程中不占用用户寄存器缓存，由系统自行处理。

I 使用说明：

1、 程序中可多次使用RS指令，和标准RS不同，MODBUS协议中的RS指令可有多处RS指令被同时使能。

2、 可省略对M8122的置ON，可省略对M8123的复位。

指令举例三：



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

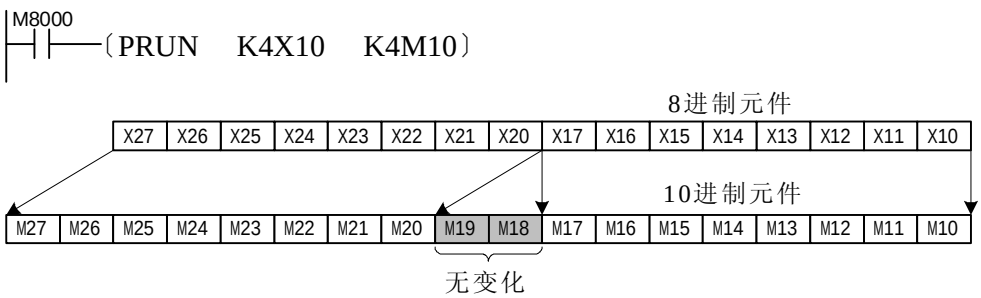
该指令是将(S)起始连续地址的位变量（以8进制为宽度单位），成批复制到(D)起始位变量组中。其中：

(S)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；

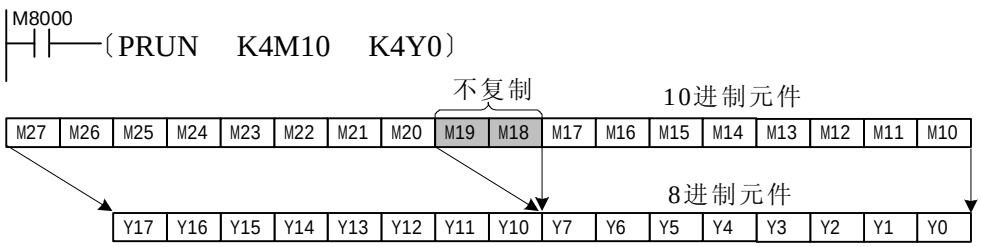
(D)为复制的目的位变量起始地址，同样要求地址的个位必需为0，如M30，Y10等。

其中Kn中，允许n＝1～8。

指令举例一：



指令举例二：



16bit	32bit	P	FNC	ASCI	HEX－ASCII 转换
✔		✔	82		
7 步		7 步	指令格式：ASCI (S) (D) (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
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，n=1~256														

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：

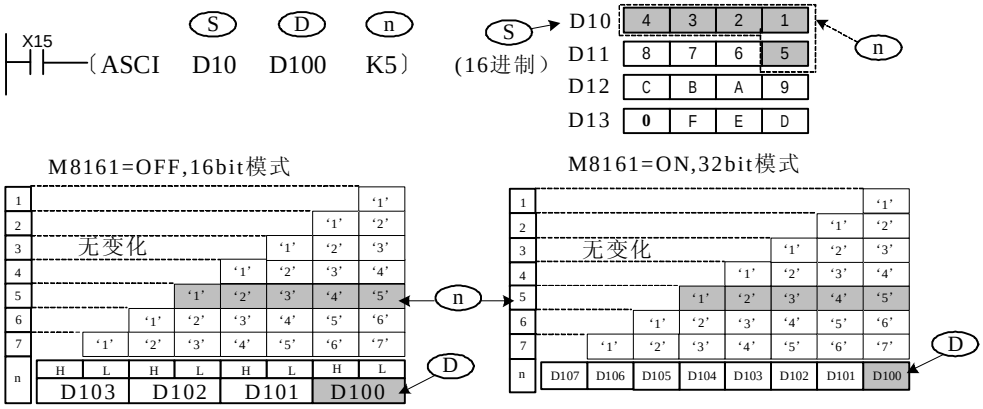
(S)为待转换的变量地址或常数数值；

(D)为转换后ASCII码的存放起始地址；

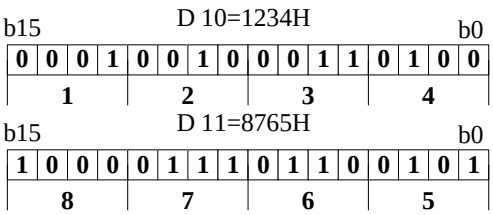
(n)为转换的字符位数。

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。关于HEX和ASCII的对照关系请参考FNC76（ASC）指令后面的附录。

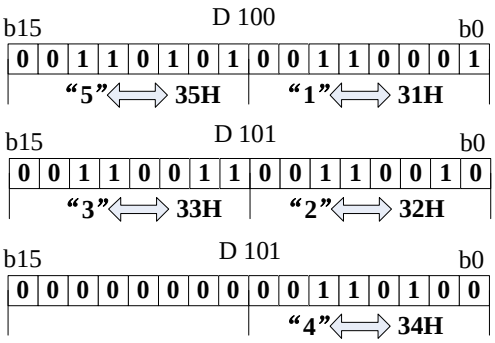
指令举例：



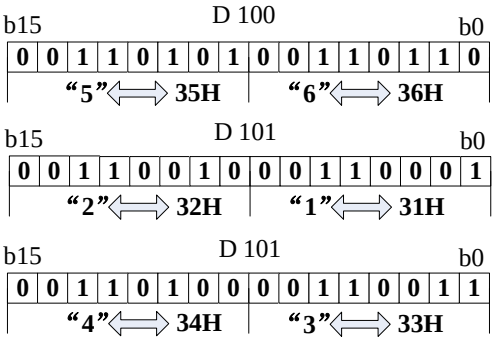
其中，M8161标志决定了计算结果存放目的变量的宽度模式，当M8161=OFF时，为16bit模式，即变量的高字节和低字节分别存储；当M8161=ON时，为8bit模式，只有变量的低字节存储结果，因此实际使用变量区域的长度增加。



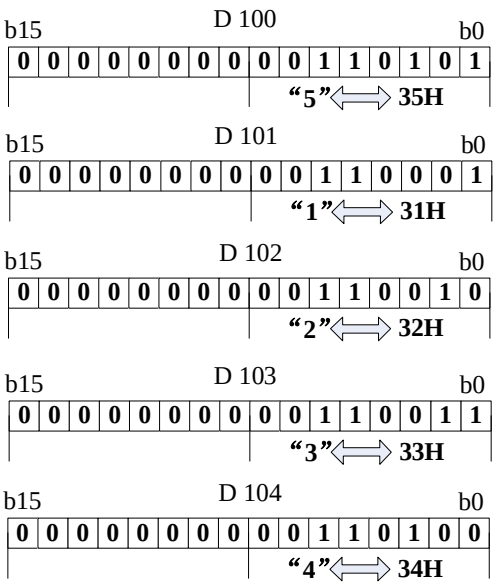
当M8161=OFF、n=5 时，位的组成(D10~D11)转换



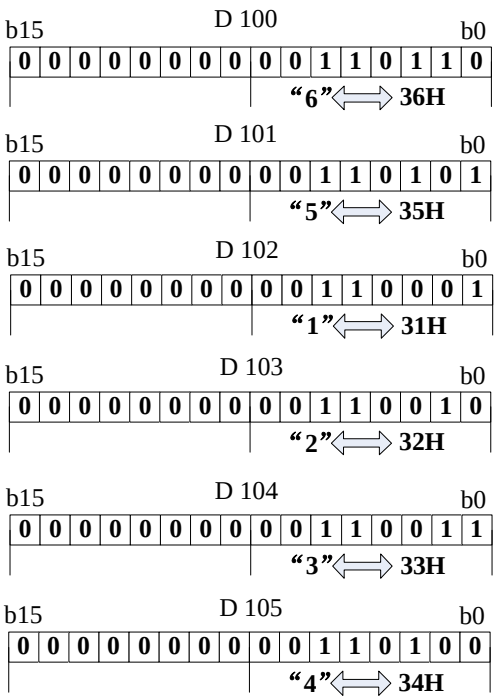
当M8161=OFF、n=6 时，位的组成(D10~D11)转换



当M8161=ON、n=5 时，位的组成(D10~D11)转换



当M8161=ON、n=6 时，位的组成(D10~D11)转换



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

16bit	32bit	P	FNC	HEX	ASCII－HEX 转换
✔		✔	83		
7 步		7 步	指令格式：HEX (S) (D) (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
(S)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔
(n)	常数，n=1~256														

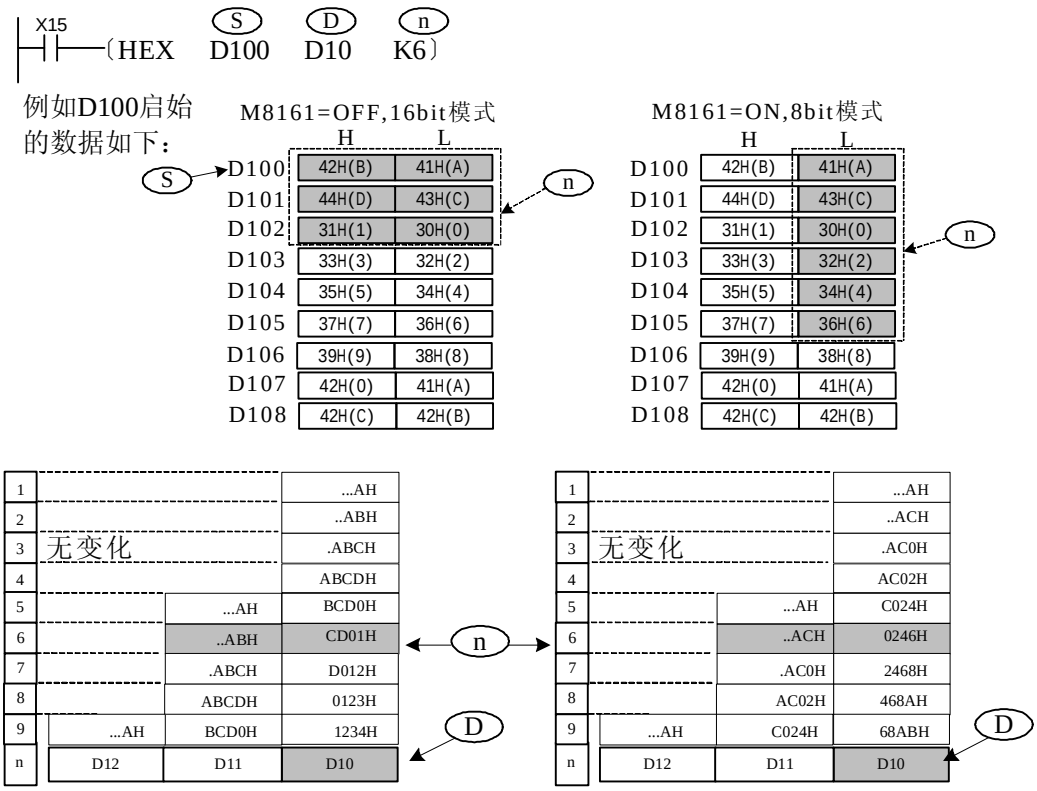
该指令是将(S)起始变量的值转换成ASCII码后，存储到(D)为起始地址的变量中，转换的字符数、存储模式可以设定。其中：

(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。

指令举例：



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。

当M8161=OFF、n=5 时，位的组成
用了D100~D102(高低字节)转换

D 10

1	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0
B				C				D				0			

D 11

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
												A			

当M8161=OFF、n=6 时，位的组成
用了D100~D102(高低字节)转换

D 10

1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1
C				D				0				1			

D 11

0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1
								A				B			

当M8161=ON、n=5 时，位的组成
用了D100~D104(低字节)转换

D 10

1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0
C				0				2				4			

D 11

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
												A			

当M8161=ON、n=6 时，位的组成
用了D100~D105(低字节)转换

D 10

0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0
0				2				4				6			

D 11

0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0
								A				C			

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	CCD	求校验和
✔		✔	84		
7 步		7 步	指令格式：CCD (S) (D) (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
(S)							✔	✔	✔	✔	✔	✔	✔		
(D)								✔	✔	✔	✔	✔	✔		
(n)	常数，n=1~256														

该指令是对(S)启始的(n)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(D)；将逐个异或逻辑运算的结果存于(D)＋1单元中。本指令用于作通信时，为了确保数据传输时的正确性所做的字符串总和检查（SumCheck）。其中：

(S)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；

(D)用于存放校验和；(D)＋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。

指令举例：

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：

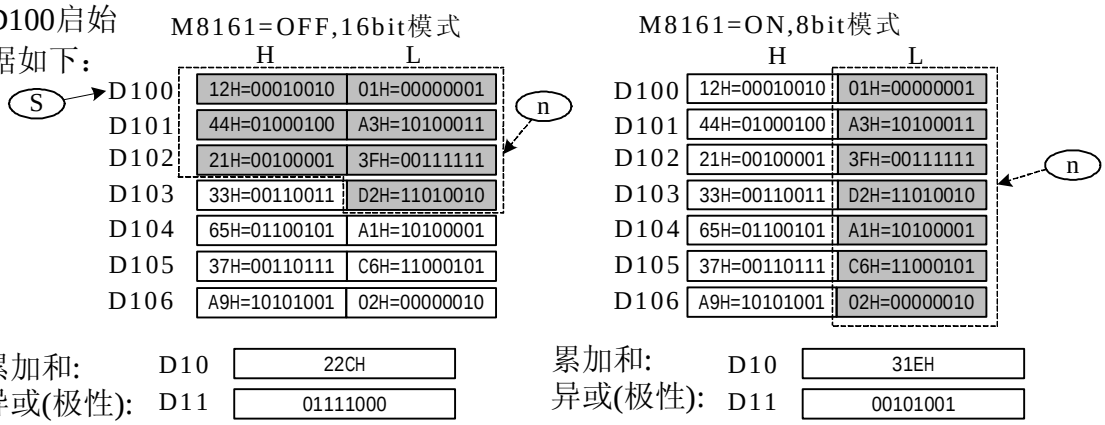
1） 将参与运算的变量都换算为二进制数；

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	PID	PID 运算
✔			88		
9 步			指令格式：PID (S1) (S2) (S3) (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
(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。PID控制在机械设备、气动设备、恒压供水和电子设备等等中有着广泛的应用。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、中间结果保存的缓存区启始地址，占用随后地址的共25个变量单元，取值范围是D0~D7975，最好指定停电保持区，在电源OFF后仍保持设定值，否则首次启动运算前需对缓存区作赋值处理。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，请指定(D)为非电池保持区，否则首次启动运算前需对其作初始化清0处理。

指令举例：

```
X10
 ┤├──〔PID   (S1) D9   (S2) D10   (S3) D200   (D) D130〕
```

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；

D200～D224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表：

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＝自整定不动作；1＝执行自整定 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	输入滤波常数（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） 输出下限设定值-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的理论运算公式如下：

正逻辑

$$\triangle MV=Kp\left|(EVn-\mathbf{EVn-1})+\frac{\mathbf{Ts}}{\mathbf{Ti}}\ EVn+Dn\right|$$

$$\mathbf{EVn=PVnf-SV}$$

$$Dn=\frac{\mathbf{T_D}}{\mathbf{Ts+a_D*T_D}}(-2PVnf\text{-}l+PVnf+PVnf\text{-}2)+\frac{\mathbf{a_D*T_D}}{\mathbf{Ts+a_D*T_D}}*Dn\text{-}1$$

$$MVn=\Sigma\triangle MVn$$

逆逻辑

$$\triangle MV=Kp\left|(EVn-\mathbf{EVn-1})+\frac{\mathbf{Ts}}{\mathbf{Ti}}\ EVn+Dn\right|$$

$$\mathbf{EVn=SV-PVnf}$$

$$Dn=\frac{\mathbf{T_D}}{\mathbf{Ts+a_D*T_D}}(2PVnf\text{-}l-PVnf-PVnf\text{-}2)+\frac{\mathbf{a_D*T_D}}{\mathbf{Ts+a_D*T_D}}*Dn\text{-}1$$

$$MVn=\Sigma\triangle MVn\triangle$$

EVn ：本次采样时的偏差

Dn ：本次的微分项

EVn-1：1 个周期前的偏差

Dn- 1 ：1 个周期前的微分项

SV ：目标值

Kp ：比例增益

PVnf ：本次采样时的测定值（滤波后）

Ts ：采样周期

PVnf-l：l 个周期前的测定值（滤波后）

Tl ：积分常数

PVnf-2：2 个周期前的测定值（滤波后）

TD ：微分常数

△ MV：输出变化量

△ αD ：微分增益

△ MVn ：本次的操作量

……

PVnf 提根据读入的测定值由下列运算式求得的值：

［滤波后的测定值 PVnf］ = PVn + L (PVnf - l - PVn)

PVn ：本次采样时的测定值

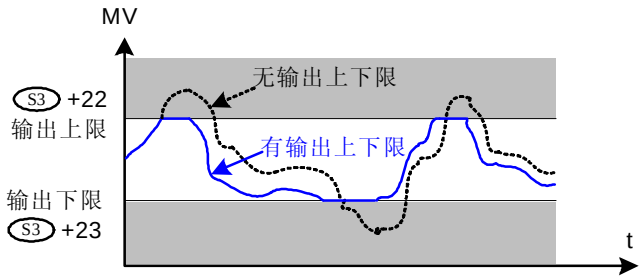
L ：滤波系数

PVnf - 1 ：1 个周期前的测定值（滤波后）

正逻辑也称正方向，如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，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。

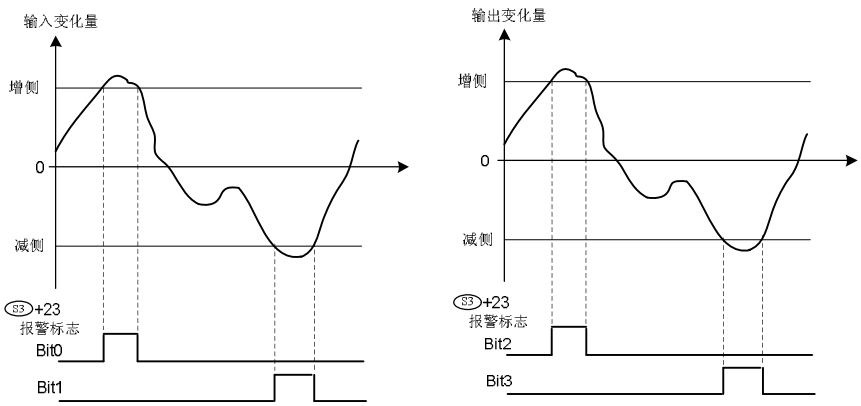


超限报警设定：

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 S3 ＋（20～23）单元中分别设定变化量报警限值，运行中就可在 S3 ＋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，S3 +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。

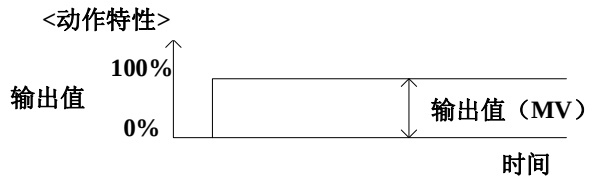
PID的3个常数的求法

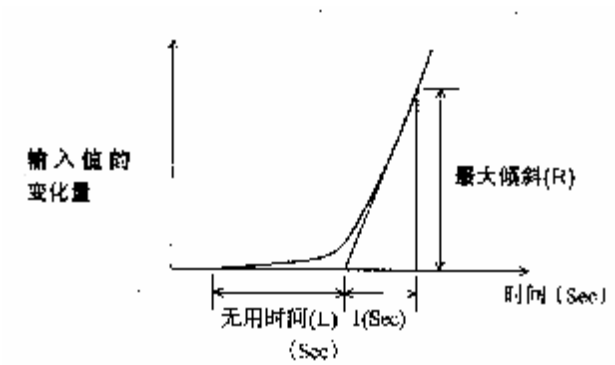
为了执行 PID 控制得到良好的控制结果，必须求得适合于控制对象的各常数（参数）的最佳值。这里必须求得 PID 的 3 个常数 （比例增益（Kp），积分时间（T1） ，微分时间 （TD）的最佳值。

作为其求解方法有阶跃反应法，这里就该阶跃反应法加以说明。

阶跃反应法是对控制系统施加 0 → 100 %※ 1 的阶跃状输出，由输入变化判断动作特性（最大倾斜（R）、无用时间（L ））来求得 PID 的 3 个常数的方法。

※ 1 阶跃状输出也可通过 0→75 %或 0→50%求得。





<动作特性和3个常数>

	比例增益（KP）[%]	积分时间（T1）[×100ms]	微分时间（T1）[×100ms]
仅有比例控制（p动作）	$\frac{1}{RL}$×输出值（MV）	————	————
仅有比例控制（p动作）	$\frac{0.9}{RL}$×输出值（MV）	33L	————
仅有比例控制（p动作）	$\frac{1.2}{RL}$×输出值（MV）	20L	50L

自动调谐功能

使用自动调谐功能为了能得到最佳 PID 控制，用阶跃反应法自动设定重要常数（动作方向（S3＋1 的 bit0）、比例增益 （S3+3）、积分时间（S3+ 4 ）、微分时间（S3+ 6））。使用 FX_{2N} 可编程控制器时，仅适用于 V2 . 00 以上版本。

l 自动调谐方法

① 传送自动调谐用输出值至输出值 D 中

这个自整定用输出值请根据输出设备在输出可能最大值的 50%~100%范围内使用。

② 请设定自整定不能设定的参数（采样时间、输入滤波、微分增益等）以及目标值等。

进行自整定时，若不满足下述要点，则出现不能正确自整定的情况，请特别注意。

③ S3+ l（ACT）的 hit4 置 1 后，则自整定开始。自整定开始时的测定值到目标值的变化量变化 1 / 3 以上，则自整定结束，S3+ l （ACT）的 bit4 自动变为 0 。

要点

O 目标值的设定

自整定开始时的测定值和目标值的差如不是 150 以上则不能正确自整定。因此，若不是 150 以上情况时，先设定自整定用目标值，待自整定完成后，再次设定目标值。

O 采样时间自整定时的采样时间必须在 1 秒（ l 000ms ）以上。另外本采样时间推荐使用大大长于输出变化周期的时间值。

要点

自整定请在系统处于稳定状态时开始。如在不稳定的状态开始，则不能正确进行自整定。

错误代码

控制参数的设定值或 PID 运算中的数据发生错误时，则运算错误 M8067 变为 ON 状态，根据其错误内容 D8067 中存入下述数据。

代码	错误内容	处理状态	处理方法
K6705	应用指令的操作数在对象软元件范围外		
K6706	应用指令的操作数在对象软元件范围外		
K6730	采样时间(TS)在对象软元件范围外(TS<0)		

代码	错误内容	处理状态	处理方法
K6732	输入滤波常数在对象(α)范围外(α<0 或 100≤α)	PID 命令运算停止	请确认控制数据的内容
K6733	比例增益(KP)在对象范围外(KP<0)		
K6734	积分时间(TI)在对象范围外(TI<0)		
K6735	微分增益(KD)在对象范围外(KD<0 或 201≤KD)		
K6736	微分时间(TD)在对象范围外(KD<0)		
K6740	采样时间(Ts)≤运算周期	PID 命令运算继续	
K6742	测定值变化量超过((PV<-32768 或 32767<(PV)		
K6743	偏差超过(EV<-32768 或 32767<EV)		
K6744	积分计算值超过(-32768~32767 以外)		
K6745	由于微分增益(舒）超过微分值超过		
K6746	微分计算值超过(-32768~32767 以外)		
K6747	PID 运算结某超过(-32768~32767 以外)		
K6750	自整定结果不良	自整定结束	自整定开始时的测定值和目标值的差为 150 以下或自整定开始时的测定值和目标值的差的 I/3 以上则结束确认测定值、目标值后，请再次进行自整定。
K6751	自整定动作方向不一致	自整定继续	从自整定开始时的的测定值预测的动作方向和自整定用输出时实际动作方向不一致。请使目标值、自整定用输出值、测定值的关系正确后，再次进行自整定。
K7652	自整定动作不良	自整定	自整定中测定值因上下变化不能正确动作。请使采样时间远远大于输出的变化周期，增大输入滤波常数。设定变更后 请再次进行自整定。

要点

必须在PID运算执行前，将正确的测定值读入PID测定值（PV）中。特别对模拟量输入模块的输入值进行PID运算时，需注意其转换时间。

编程说明：

1、 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(S3)变量区域不

要有重叠；在步进指令、跳转指令、定时中断、子程序中也可以使用，在此情况下执行PID指令时，需事先清除(S3)＋7缓存单元。

2、 采样时间TS的最大误差为-（1运算周期+lms）~+（1运算周期）。如果采样时间TS≤可编程控制器的1个运算周期，则发生下述的PID运算错误（K6740），并以TS＝运算周期执行PID运算。在这种情况下，建议最好使用恒定扫描模式或在定时器中断（16□□～18□□）中使用PID指令。

4.3.2.10 浮点数（110~147）

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)		

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)		✔	✔	✔											

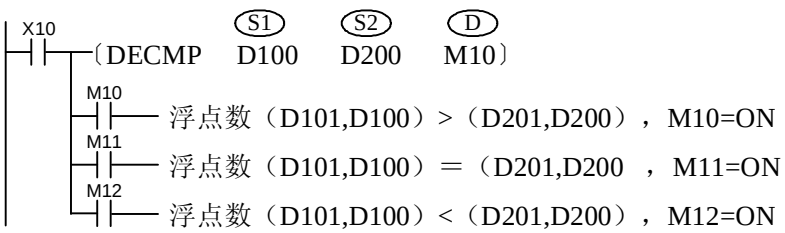
该指令是进行2个浮点数变量的比较，将比较的结果输出到(D)启始的3个变量中。其中：

(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。

指令举例：



当X10=ON时，M10~M12其中之一会ON。
X10由ON变OFF 时，不执行DECP指令，M10~M12仍保持X10=OFF之前的状态，要清除M10~M12的比较结果可用RST或者ZRST对M10~M12进行清除。
若需要得到≧、≦、≠的结果时，可将M10~M12串并联即可取得。

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
	✔	✔			
	17 步	17 步	指令格式：EZCP (S1) (S2) (S) (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
(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：

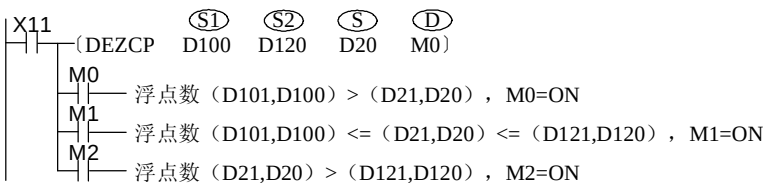
(S1)为二进制浮点变量区间的下限；

(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。

指令举例：



当X11=ON时，M0~M2其中之一会ON。
当X11由ON变OFF时，不执行DEZCP指令，M0/M1/M2保持X11=OFF以前的状态不变。要清除M0~M2的比较结果可用RST或者ZRST对M0~M2进行清除。

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)		

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)													✔		

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。

指令举例：

```
X1
├┤├──〔DEBCD   (S) D2   (D) D10〕
```

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。

其中2 进制浮点数[D3,D2]实数23 位，指数 8 位，符号位 1 位

10 进制浮点数[D 11， D 10]指数(D3)实数(D2),用科学计算式表示为$D2 \times 10^{D3}$

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)		

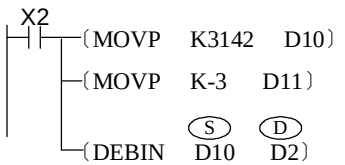
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

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。

指令举例：



将十进制浮点数3.142（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	FNC120	EADD	二进制浮点加法运算
	✔	✔			
	13 步	13 步	指令格式：EADD (S1) (S2) (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
(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被加数和加数；

(D)为二进制浮点加法和的存放单元。

S1或**S2**来源操作数若是常数K或H，会自动将该常数变换成二进制浮点数值来作加法运算；

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令举例：

```
                          (S1)   (S2)   (D)
X10 ─┤├──〔DEADD    D2     D4     D10〕
X11 ─┤├──〔DEADDP   D20    K123    D20〕
```

当 X10＝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 X11 由 OFF 变为 ON 时，二进制浮点数（D21，D20）的值增大 123。这里的常数 K123 在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，此时请使用脉冲执行型指令 DEADDP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

16bit	32bit	P	FNC121	ESUB	二进制浮点减法运算
	✔	✔			
	13 步	13 步	指令格式：ESUB (S1) (S2) (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
(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。

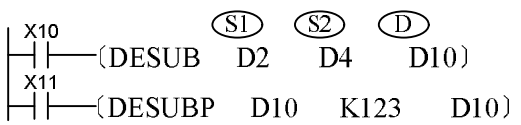
S1或**S2**来源操作数若是常数K或H，会自动将该常数变换成二进制浮点数值来作减法运算；

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令举例：



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。这里的常数 K123 在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，此时请使用脉冲执行型指令 DESUBP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

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

S1或**S2**来源操作数若是常数K或H，会自动将该常数变换成二进制浮点数值来作乘法运算；

若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

指令举例：

```
                          (S1)   (S2)   (D)
X12
 ┤├──〔DEMUL    D2     D4     D10〕
X13
 ┤├──〔DEMULP   D20    K3     D20〕
```

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（D21，D20）的值乘以 3 倍后存回（D21，D20）。这里的常数 K3 在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，此时请使用脉冲执行型指令 DEMULP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

16bit	32bit	P	FNC123	EDIV	二进制浮点除法运算
	✔	✔			
	13 步	13 步	指令格式：EDIV (S1) (S2) (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
(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。

S1或**S2**来源操作数若是常数K或H，会自动将该常数变换成二进制浮点数值来作除法运算；

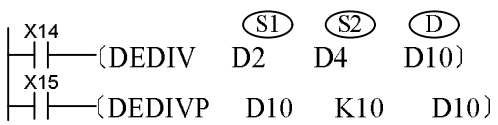
若计算结果为零，则0标志（M8020）会置位。

若运算结果的绝对值大于可表示的最大浮点值，则进位标志（M8022）会置位。

若运算结果的绝对值小于可表示的最小浮点值，则借位标志（M8021）会置位。

除数不得为 0，否则计算出错，M8067、M8068 会置 ON。

指令举例：



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）。这里的常数 K10 在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，此时请使用脉冲执行型指令 DEDIVP，否则若采用连续执行指令，则程序每扫描一次，计算就会被执行一次。

16bit	32bit	P	FNC127	ESQR	二进制浮点开平方运算
	✔	✔			
	9 步	9 步	指令格式：ESQR (S) (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)													✔		

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；

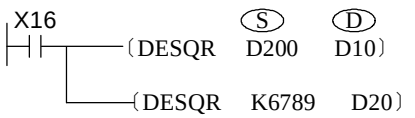
(D)为二进制浮点平方根的存储单元。

操作数(S)若是常数K或H，会自动将该常数变换成二进制浮点数值来作开方运算；

若计算结果为零，则0标志（M8020）会置位。

(S)只有正数有效，如果是负数则计算出错，M8067、M8068 会置 ON。

指令举例：



将二进制浮点数开方结果 $\sqrt{(D201,D200)}$ 存放到→ (D11,D10)
将二进制浮点数K6789做开方，结果存放到？ (D21,D20)，
这里的常数K6789在运算前已自动被调整为二进制浮点数；

16bit	32bit	P	FNC129	INT	二进制浮点→BIN 整数变换
✔	✔	✔			
5 步	9 步		指令格式：INT (S) (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)													✔		

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(D)中。其中：

(S)为待取整变换的二进制浮点数变量；

(D)为变换后BIN整数结果的存储单元。

若计算结果为零，则0标志（M8020）会置位。

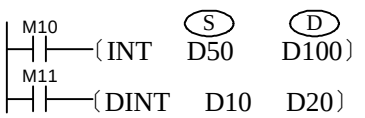
若运算结果有浮点数被舍弃时，则借位标志（M8021）会置位。

若运算结果若超出下列范围时（溢位），则进位标志（M8022）会置位。

16位指令：-32，768~32，767

32 位指令：-2，147，483，648~2，147，483，647

指令举例：



将浮点数(D51,D50)取整后，存放到→ (D100)

将浮点数(D11,D10)取整后，存放到→ (D21,D20)

注意INT和DINT指令存放结果的区别

243

16bit	32bit	P	FNC130	SIN	浮点 SIN 运算
	✔	✔			
	9 步	9 步	指令格式：SIN (S) (D)		

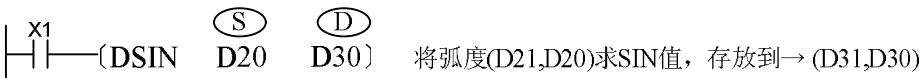
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

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：

(S)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<=2π；

(D)为变换后SIN计算结果的存储单元，二进制浮点数格式。

指令举例一：

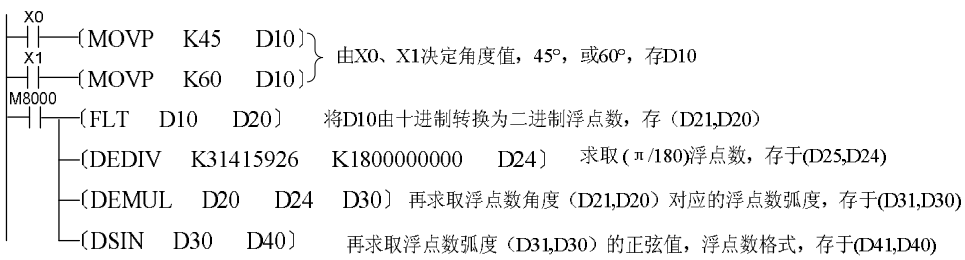


这里计算的源数据、SIN 结果都为二进制浮点数格式。

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

指令举例二：

根据角度值求对应SIN值的程序：



16bit	32bit	P	FNC131	COS	浮点 COS 运算
	✔	✔			
	9 步	9 步	指令格式：COS (S) (D)		

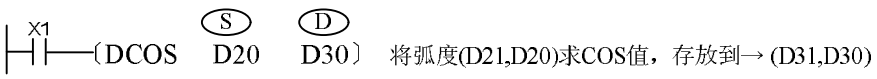
操作数	位元件				字 元 件										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
(S)													✔		
(D)													✔		

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：

(S)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；

(D)为变换后COS计算结果的存储单元，二进制浮点数格式。

指令举例：



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	FNC132	TAN	浮点 TAN 运算
	✔	✔			
	9 步	9 步	指令格式：TAN (S) (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)													✔		

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。

指令举例：

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	FNC147	SWAP	高低字节交换
✔	✔	✔			
3 步	5 步		指令格式：SWAP (S)		

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)								✔	✔	✔	✔	✔	✔	✔	✔

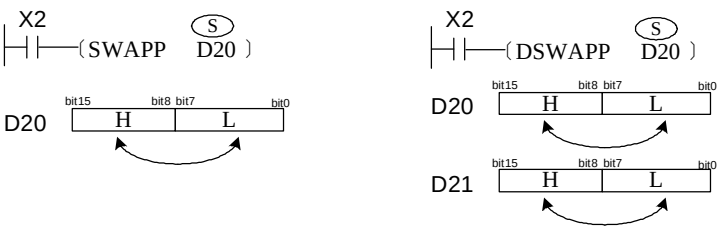
该指令是将指定变量(S)的高低字节的值进行互相交换。

16 位指令时，高8 位与低8 位的值进行互相交换。

32 位指令时，两个寄存器的高 8 位与低 8 位的值各自进行互相交换

注意此指令一般使用脉冲执行型指令，否则若采用连续执行指令，则程序每扫描一次，就会进行一次交换。

指令举例：



左图中将D20 的高8 位与低8 位的值进行互相交换
右图中将D20 的高8 位与低8 位的值进行互相交换，
D21 的高8 位与低8 位的值进行互相交换，

4.3.2.11 定位控制（155~159）

16bit	32bit	P	FNC155	ABS	ABS 位置数据读取
	✔				
	13 步		指令格式：ABS (S) (D1) (D2)		

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)								✔	✔	✔	✔	✔	✔	✔	✔

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：

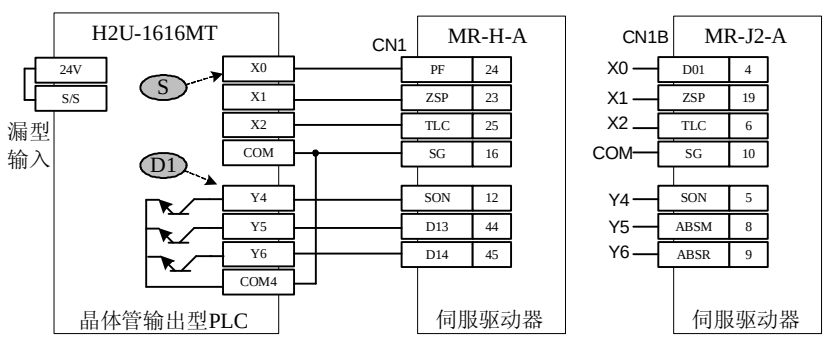
(S) 为读取伺服装置的输入信号，占用后续共3个单元；

(D1) 为传送至伺服装置的控制信号，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，占用 (D2)＋1、(D2) 单元，通常指定D8140。

指令举例：

对应的接线方式如下图，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：

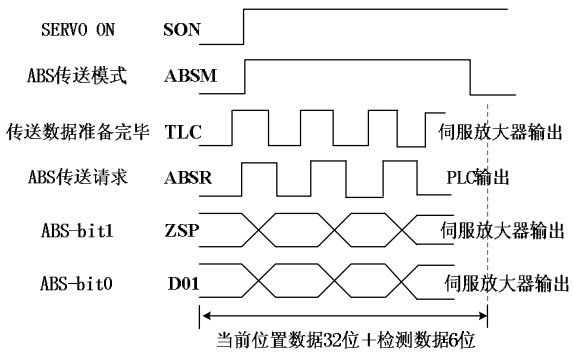


当指令驱动 M10 变为 ON 时开始读取，读取完毕，M8029 标志置为 ON；

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 数据的编程举例如下，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 M21 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	FNC156	ZRN	原点回归
✔	✔				
9 步	17 步		指令格式：ZRN (S1) (S2) (S3) (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
(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；

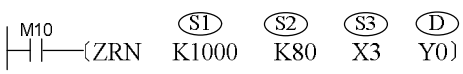
(S2)为原点信号变为ON后的爬行速度，范围是10～32，767Hz；

(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；

(D)为脉冲输出启始地址。对于3624MT/2416MT型只能指定Y0或Y1；其他MT型只能指定Y0/Y1/Y2；而MTQ型则可指定Y0/Y1/Y2/Y3/Y4等；

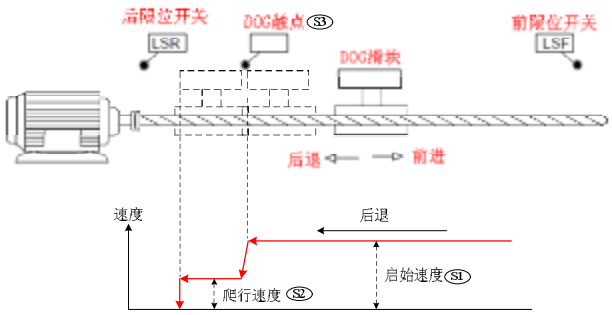
相对位置控制指令 DRVI（FNC158）和绝对位置控制指令 DRVA（FNCl59）在执行时，控制器会计算自身已发出的正转脉冲或反转脉冲数，并将之保存在寄存器[D8141，D8140]（Y000）和[D8l43，D8l42]（Y001)。但该寄存器的数据在断电时会消失，故上电时和初始运行时，必须执行原点回归指令 ZRN，以事先将机械动作的原点位置的数据写入。

指令举例：



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]，Y00l：[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），所以回原点动作必须在 DOG 信号前端开始。

注意事项：

定位指令（ZRN/PLSV/DRVI/DRVA）在程序中可多次使用，但不要对同一高速 Y 输出口同时进行操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需在状态位（Y OOO ： [M8147]，YO01；[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」。

16bit	32bit	P	FNC157	PLSV	可变速度脉冲输出
✔	✔				
7 步	13 步		指令格式：PLSV (S) (D1) (D2)		

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)		✔	✔	✔											

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：

(S)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1～100，000Hz；－1～－100，000Hz。其中负号表示反方向运行的指令信号；

(D1)为脉冲输出端口；3624MT/2416MT型只能指定Y0或Y1；1616MT/3232MT型能指定Y0/Y1/Y2；而MTQ型则可选Y0/Y1/Y2/Y3/Y4等；

(D2)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。

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. D8151（高字节），D8150（低字节）：Y002 输出脉冲数。反转时减少。（使用 32 位）

4. D8153（高字节），D8152（低字节）：Y003 输出脉冲数。反转时减少。（使用 32 位）

5. D8155（高字节），D8154（低字节）：Y004 输出脉冲数。反转时减少。（使用 32 位）

6. M8145：Y000 脉冲输出停止（立即停止）
7. M8146：Y001 脉冲输出停止（立即停止）
8. M8152：Y002 脉冲输出停止（立即停止）
9. M8153：Y003 脉冲输出停止（立即停止）
10. M8154：Y004 脉冲输出停止（立即停止）
11. M8147：Y000 脉冲输出中监控（BUSY/READY）
12. M8148：Y001 脉冲输出中监控（BUSY/READY）
13. M8149：Y002 脉冲输出中监控（BUSY/READY）
14. M8150：Y003 脉冲输出中监控（BUSY/READY）
15. M8151：Y004 脉冲输出中监控（BUSY/READY）

16bit	32bit	P	FNC158	DRVI	相对位置控制
✔	✔				
9 步	17 步		指令格式：DRVI (S1) (S2) (D1) (D2)		

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)		✔	✔	✔											

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；

(S2)为指定的输出脉冲频率，16bit指令时，范围为10～32767Hz；32bit指令时，范围为10～100，000Hz；

(D1)为脉冲输出端口；对于3624MT/2416MT型只能指定Y0或Y1；其他MT型只能指定Y0/Y1/Y2；而MTQ型则可指定Y0/Y1/Y2/Y3/Y4等；

(D2)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。

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[D8l41（高字节），D8l40（低字节）]（使用 32 位）

向［Y00l］输出时，当前寄存器为[D8l43（高字节），D8l42（低字节）]（使用 32 位）

向［Y002］输出时，当前寄存器为[D8l51（高字节），D8l50（低字节）]（使用 32 位）

向［Y003］输出时，当前寄存器为[D8l53（高字节），D8l52（低字节）]（使用 32 位）

向［Y004］输出时，当前寄存器为[D8l55（高字节），D8l54（低字节）]（使用 32 位）

反转时，当前值寄存器的数值减小。

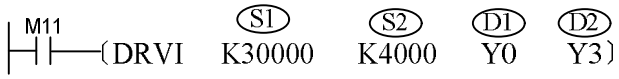
在指令执行过程中，即使改变操作数的内容，也无法在当前运行中表现出来。只在下一次指令执行时才有效。

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；

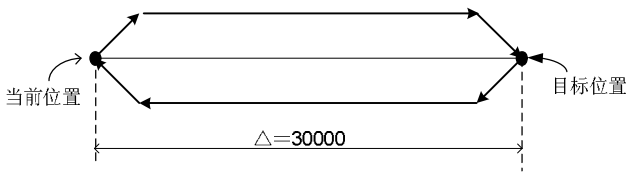
指令驱动接点变为 OFF 后，在脉冲输出中断标志 M8147（Y000）、M8148（Y001）、M8149（Y002）、M8150（Y003）、M8151（Y004）处于 ON 时，将不接受指令的再次驱

动。

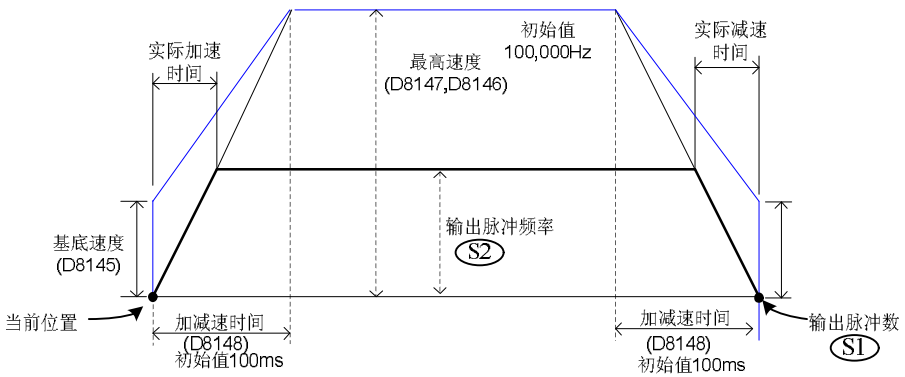
指令举例：



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 √[最高速度 (D8147，D8146)] × (2 × (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 FNC158(DRVI)，FNCl59(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（D8l47，D8l46）的 l/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）

［D8l48］：执行 FNC158(DRVI)，FNCl59(DRVA）指令时的加减速时间。加减速时间表示到达最高速度（D8147，D8146）所需的时间。因此，当输出脉冲频率(S2)低于最高速度（D8147，D8146）时，实际加减速时间会缩短。设定范围：50~5，000(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）

［M8152］：Y002 脉冲输出停止（立即停止）

［M8153］：Y003 脉冲输出停止（立即停止）

［M8154］：Y004 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）

［M8149］：Y002 脉冲输出中监控（BUSY/READY）

［M8150］：Y003 脉冲输出中监控（BUSY/READY）

［M8151］：Y004 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，但不要对同一输出口同时进行输出操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需在状态位（Y 000 ： [M8147]，Y001；[M8148]），Y002；[M8149]），Y003；[M8150]），Y004；[M8151]）]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」。

注意：

在新版本的H2U系列PLC中，PLSR，DRVI，DRVA等指令的功能有增强，请参见附录5.8中的说明。

16bit	32bit	P	FNC159	DRVA	绝对位置控制
✔	✔				
9 步	17 步		指令格式：DRVA (S1) (S2) (D1) (D2)		

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)		✔	✔	✔											

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)为指定的目标位置(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。

若(D1)＝［Y000］，对应[D8l41（高字节），D8l40（低字节）]（使用 32 位）为绝对位置；

若(D1)＝［Y00l］，对应[D8l43（高字节），D8l42（低字节）]（使用 32 位）为绝对位置；

若(D1)＝［Y002］，对应[D8l51（高字节），D8l50（低字节）]（使用 32 位）为绝对位置；

若(D1)＝［Y003］，对应[D8l53（高字节），D8l52（低字节）]（使用 32 位）为绝对位置；

若(D1)＝［Y004］，对应[D8l55（高字节），D8l54（低字节）]（使用 32 位）为绝对位置。

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；对于3624MT/2416MT型只能指定Y0或Y1；其他MT型只能指定Y0/Y1/Y2；而MTQ型则可指定Y0/Y1/Y2/Y3/Y4等；

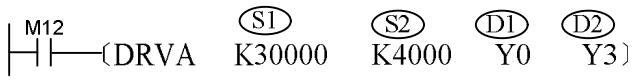
(D2)运行方向输出端口或位变量，可根据(S1)和当前位置的差值决定，输出为ON状态，表示为正向运行；否则为反向运行。

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。

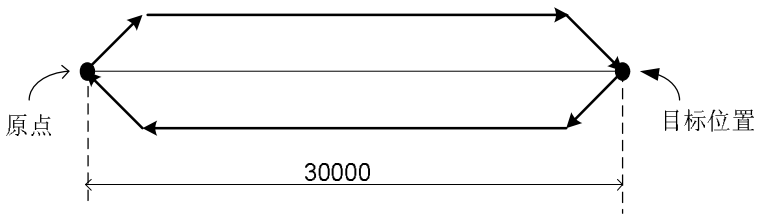
若在指令执行过程中，指令驱动的接点变为 OFF 时，将减速停止。此时执行完成标志 M8029 不会动作；

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。

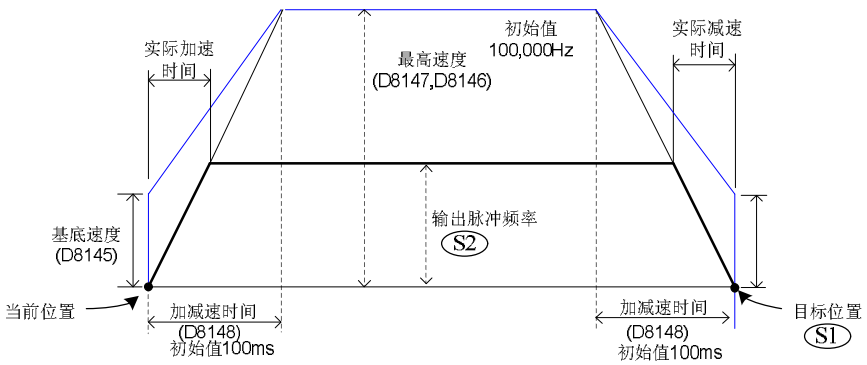
指令举例：



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 $\geq$ [最高速度 (D8147，D8146)]Hz ÷ (2 × [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 FNC158(DRVI)，FNCl59(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（D8l47，D8l46）的 l/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）

［D8l48］：执行 FNC158(DRVI)，FNCl59(DRVA）指令时的加减速时间。加减速时间表示到达最高速度（D8147，D8146）所需的时间。因此，当输出脉冲频率(S2)低于最高

速度（D8147，D8146）时，实际加减速时间会缩短。设定范围：50~5，000(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）

［M8152］：Y002 脉冲输出停止（立即停止）

［M8153］：Y003 脉冲输出停止（立即停止）

［M8154］：Y004 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）

［M8149］：Y002 脉冲输出中监控（BUSY/READY）

［M8150］：Y003 脉冲输出中监控（BUSY/READY）

［M8151］：Y004 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，但不要对同一输出口进行输出操作；

当指令的驱动能流 OFF 之后，再次驱动 ON 时，必需在状态位（Y000 ： [M81471]，Y001；[M8148]），Y002；[M8149]），Y003；[M8150]），Y004；[M8151]）]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」。

注意：

在新版本的H2U系列PLC中，PLSR，DRVI，DRVA等指令的功能有增强，请参见附录5.8中的说明。

4.3.2.12 时钟运算（160~169）

16bit	32bit	P	FNC160	TCMP	时钟数据比较
✔		✔			
11 步		11步	指令格式：TCMP (S1) (S2) (S3) (S) (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
(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；

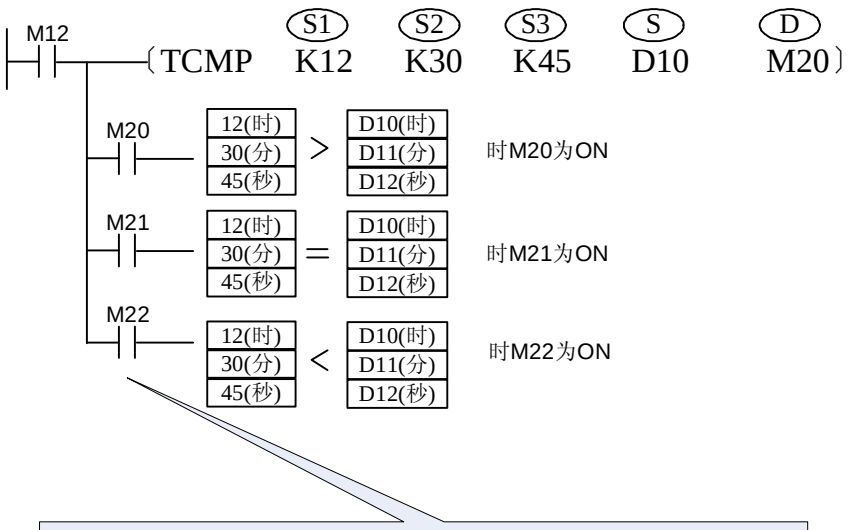
(S2)为指定比较用时间的“分”，范围是0～59；

(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；

指令举例：



当M12 ON时，M20~M22其中之一会ON。
当M12由ON变OFF时，不执行TCMP指令，M20/M21/M22保持M12=OFF以前的状态不变。要清除M20~M22的比较结果可用RST或者ZRST对M20~M22进行清除。
若需要得到≧、≦、≠的结果时，可将M20~M22 串并联即可取得。

16bit	32bit	P	FNC161	TZCP	时钟数据区间比较
✔		✔			
9 步		9 步	指令格式： TZCP	(S1) (S2) (S) (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
(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：

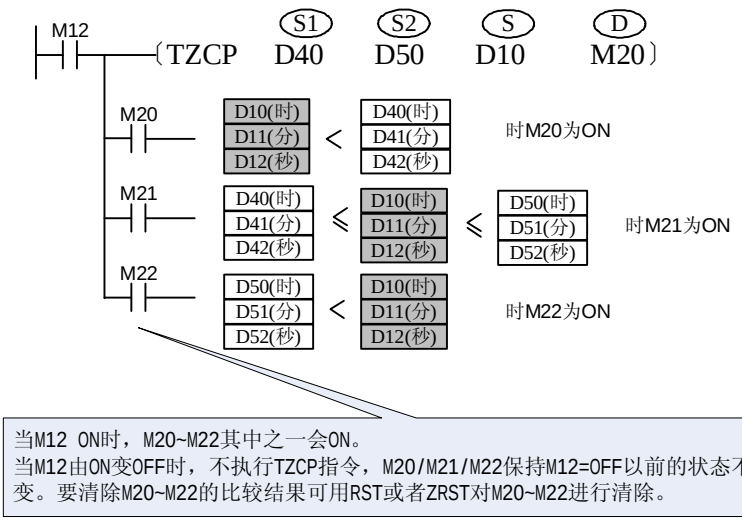
(S1)为设定的时间比较下限，占用 3 个连续的变量单元，依次存储时、分、秒数据：

(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；

指令举例：



16bit	32bit	P	FNC162	TADD	时钟数据加法运算
✔		✔			
7 步		7 步	指令格式：TADD (S1) (S2) (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
(S1)											✔	✔	✔		
(S2)											✔	✔	✔		
(D)											✔	✔	✔		

该指令是将2组时钟数据的时/分/秒对应相加，结果保存于指定的变量中。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：

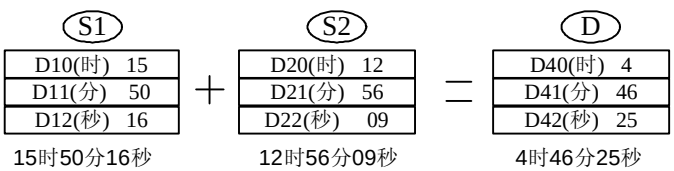
若计算结果超过 24 小时，进位标志 M8022 置 1，实际显示的时间会减去 24：00：00 的数值；

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：

如果加法运算结果超过 24 小时，则进位标志 M8022 置 On。



16bit	32bit	P	FNC163	TSUB	时钟数据减法运算
✔		✔			
7 步		7 步	指令格式：TSUB (S1) (S2) (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
(S1)											✔	✔	✔		
(S2)											✔	✔	✔		
(D)											✔	✔	✔		

该指令是将2组时钟数据的时/分/秒对应相减，结果保存于指定的变量中。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：

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；

若计算结果 00：00：00，零标志 M8020 置 1；

指令举例：

完成的操作如下：

如果加法运算结果为负数时，则借位标志 M8021 置 On。

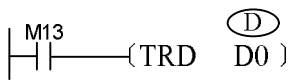
16bit	32bit	P	FNC166	TRD	时钟数据读取
✔		✔			
3 步		3 步	指令格式：TRD (D)		

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

该指令是读取PLC内置的实时时钟的年/月/日/时/分/秒/星期，将该7个数据保存于指定的寄存器中。

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：年、月、日、时、分、秒、星期等数据。

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

注：一般情况下要使用可编程控制器的时钟，先用 TDR 指令将时钟读出来放到 D 寄存器里再使用，不要直接使用 D8012~D8018 的值。

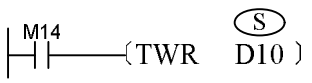
16bit	32bit	P	FNC167	TWR	时钟数据写入
✔		✔			
3 步		3 步	指令格式：TWR (S)		

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)											✔	✔	✔		

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PLC内置的实时时钟数据里。

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：年、月、日、时、分、秒、星期等数据

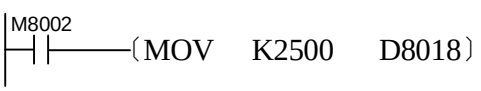
指令举例一：



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

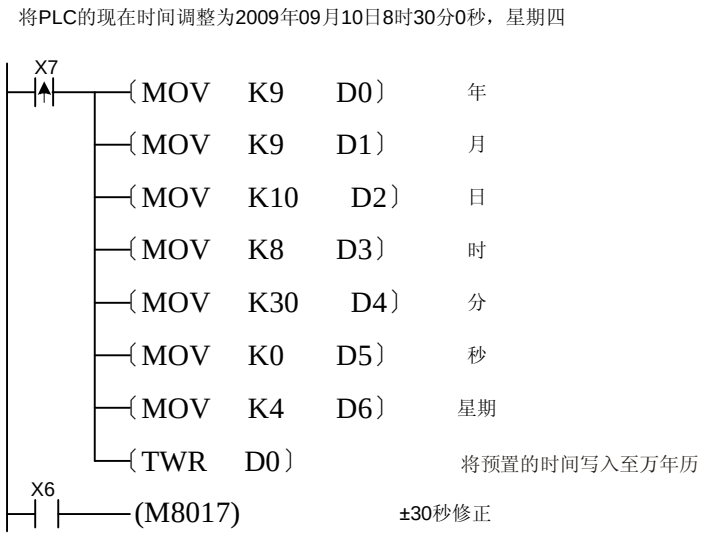
- 注意在写入时钟的时候是7个数据全写入的，在预先设值的时候不能缺少某个变量，比如星期不写，则默认是0，为星期天；如果月不预先赋值，则此时的月变量是0，则PLC认为你提供的月是错误的，此次时钟的修改无效。
- M8017每On一次，PLC内部时钟作±30秒校正动作，这里的校正是指当PLC的内部时钟的秒针于1~29时，会被自动归为“0”秒而分针不变、30~59时，也会被自动归为“0”秒，分针加1分钟。
- M8015置ON可以停止时钟计时。
- 通常的情况下年只显示 2 位数（例：2009 年只显示 09），若需要将“年”采用显示 4 位的格式，仅需在一个扫描周期执行如下语句，就可以切换成 4 位显示格式：



若原来 D8018＝09，切换后 D8018＝2009。

PLC 内部时钟校时方法如下。

指令举例二：



注：一般情况下要修改可编程控制器的时钟，用 TWR 指令将时钟写入到 D8012~D8018，用 MOV 指令直接对 D8012~D8018 进行赋值时需要将 M8015 置位才能写入。

16bit	32bit	P	FNC169	HOUR	计时器
✔	✔				
7 步	13 步		指令格式：HOUR (S) (D1) (D2)		

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)		✔	✔	✔											

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；

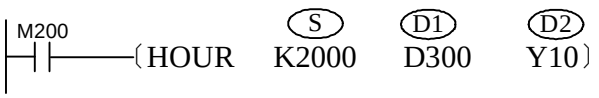
(D2)计时到达告警输出变量单元，当计时到达设定值后，该指定单元状态有效。

16bit时：(D1)设置范围K0~K32，767，单位：小时。(D1)+1为未满1个小时的现在时间值，设置范围K0~K3，599，单位：秒。此时的(D1)共占用2个单元，

32bit 时：(D1)+1、(D1)设置范围 K0~K2，147，483，647，单位：小时。(D1)+2为未满 1 个小时的现在时间值，设置范围 K0~K3，599，单位：秒。此时的(D1)共占用 3 个单元。

指令(D1)计时没有负数，若(D1)制定为非停电保持的寄存器区域，则在 PLC 由 STOP 到 RUN 或者在掉电的时候会将(D1)的值清零。若需要在 PLC 掉电的情况下仍能保持当前值数据，请将(D1)指定为停电保持区域的寄存器。

指令举例：



当M200＝ON时，累计该状态的持续时间，将时记录在D300中，将不满1小时的秒记录在D301中，当D300累计时间达到2000小时后，Y10输出状态为ON。计时条件满足时，到

达Ⓢ指定数值后，累计计时仍继续进行，读数会继续增大；现在时间值D300到达最大数值32，767小时、D301达到3，599秒时会停止计时测量，要重新计时须将现在时间值D300、301清除为0。

4.3.2.13 外围设备（170~177）

16bit	32bit	P	FNC170	GRY	格雷码转换（BIN→GRY）
✔	✔	✔			
5 步	9 步		指令格式：GRY (S) (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)								✔	✔	✔	✔	✔	✔	✔	✔

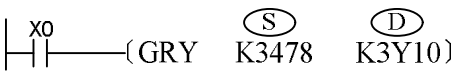
该指令是将BIN数值，转换为格雷码（GRY）。其中：

(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，超出此范围时M8067、M8068会置On，指令不执行；

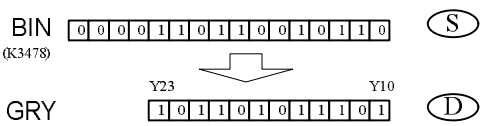
(D) 为转换为格雷码（GRY）后的存放单元。

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)；

指令举例：



执行结果：



16bit	32bit	P	FNC171	GBIN	格雷码逆转换（GRY→BIN）
✔	✔	✔			
5 步	9 步		指令格式：GBIN (S) (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)								✔	✔	✔	✔	✔	✔	✔	✔

该指令是将GRY格雷码，转换为二进制数值。其中：

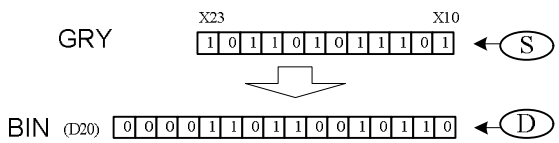
(S)为待转换的GRY码数据源或数据变量单元；16bit指令时范围0～32，767，32bit指令时范围0～2，147，483，647，超出此范围时M8067、M8068会置On，指令不执行；

(D)为转换为BIN后的存放单元。

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）。

指令举例：

执行结果举例：



16bit	32bit	P	FNC176	RD3A	模拟量模块读取
✔		✔			
7 步		7 步	指令格式：RD3A (m1) (m2) (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
(m1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(m2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

本指令提供与三菱 FX0N~3A 型模块的模拟量输入值的读取指令。其中：

(m1) 为特殊模块号；（K0~K7）

(m2) 为模拟量输入通道；（K1~K2）

(D) 为读取回来后的值的存放地址

指令举例：

用 FROM 指令也可以读取 FX0N~3A 的模拟量输入值。

16bit	32bit	P	FNC177	WR3A	模拟量模块写入
✔		✔			
7 步		7 步	指令格式： WR3A (m1) (m2) (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
(m1)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(m2)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
(D)								✔	✔	✔	✔	✔	✔	✔	✔

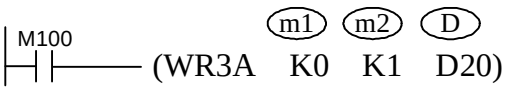
本指令提供与三菱 FX0N~3A 型模块的模拟量输出值的写入指令。其中：

(m1) 为特殊模块号；（K0~K7）

(m2) 为模拟量输入通道；（K1）

(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。

指令举例：



用 TO 指令也可以对 FX0N~3A 的模拟量输出值进行写入。

4.3.2.14 接点比较（224~246）

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	AND<	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)		

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)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

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：

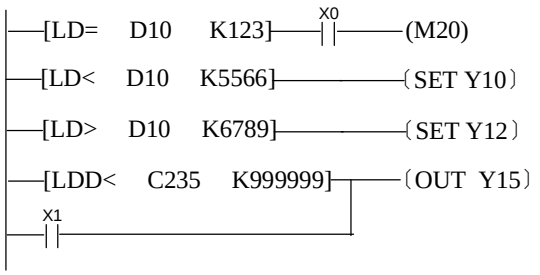
(S1)为待比较的数据源或数据变量单元1；

(S2)为待比较的数据源或数据变量单元2。

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)

指令举例：



当D10 的内容等于K123且X0=On 的时候，M20=On 。

当D10 的内容小于K5566的时候，Y10=On 并保持住。

当D10 的内容大于K6789的时候，Y12=On 并保持住。

当C235 的内容小于K999999，或者X1=On 的时候，Y15=On 。

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。

32位计数器(C200~C255)以本指令作比较时，一定要使用32位指令(LDD☼)。

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)		

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)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

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：

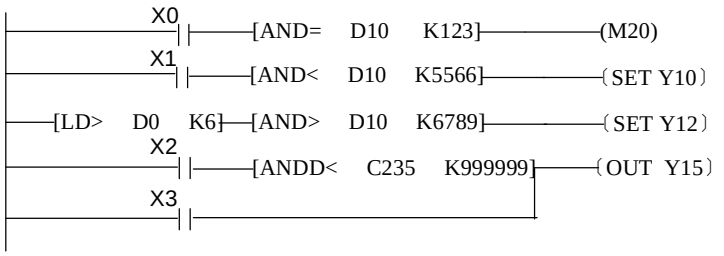
(S1)为待比较的数据源或数据变量单元1；

(S2)为待比较的数据源或数据变量单元2。

AND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=	232	ANDD=	(S1) = (S2)	(S1) ≠ (S2)
AND>	233	ANDD>	(S1) > (S2)	(S1) <= (S2)
AND<	234	ANDD<	(S1) < (S2)	(S1) >= (S2)
16bit 指令	**FNC NO**	**32bit 指令**	**导通条件**	**非导通条件**
AND<>	236	ANDD<>	(S1) <> (S2)	(S1) = (S2)
AND<=	237	ANDD<=	(S1) <= (S2)	(S1) > (S2)
AND>=	238	ANDD>=	(S1) >= (S2)	(S1) < (S2)

指令举例：



当X0=On 且D10 的值又等于K123 时，M20=On。

当X1=On 且D10 的值又小于K5566 时，Y10=On并保持住。

当D0的值大于K6且D10 的值又大于K6789 时，Y12=On并保持住。

当X2=On 且C235 的值又小于K999999，或者X3=ON时，Y15=On并保持住。

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。

32位计数器(C200~C255)以本指令作比较时，一定要使用32位指令(ANDD☼)。

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)		

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)					✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

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：

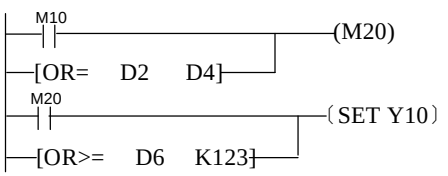
(S1)为待比较的数据源或数据变量单元1；

(S2)为待比较的数据源或数据变量单元2。

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=	240	ORD=	(S1) = (S2)	(S1) ≠ (S2)
OR>	241	ORD>	(S1) > (S2)	(S1) <= (S2)
OR<	242	ORD<	(S1) < (S2)	(S1) >= (S2)
OR<>	244	ORD<>	(S1) <> (S2)	(S1) = (S2)
OR<=	245	ORD<=	(S1) <= (S2)	(S1) > (S2)
OR>=	246	ORD>=	(S1) >= (S2)	(S1) < (S2)

指令举例：



当M10=On，或者是D2 的值等于D4 时，M20=On。

当M20=On，或者是D6 的值等于K123 时，Y10=On 并保持住。

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。

32 位计数器(C200~C255)以本指令作比较时，一定要使用 32 位指令(ORD☼)。

附 录

5.1 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24100H2U=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

M元件	M元件的描述	D元件	D元件的描述
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

M元件	M元件的描述	D元件	D元件的描述
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
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I6□□中断禁止	D8056	保留
M8057	驱动I7□□中断禁止	D8057	保留
M8058	驱动I8□□中断禁止	D8058	保留

M元件	M元件的描述			D元件	D元件的描述
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
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	高速计数器多段中断使能(默认OFF)			D8084	高速计数器多段中断的计数器序号
M8085	Y0端口的输出初始化标志			D8085	多段中断的数据默认为0
M8086	Y1端口的输出初始化标志			D8086	对应的D元件序号
M8087	Y2端口的输出初始化标志			D8087	保留
M8088	Y3端口的输出初始化标志			D8088	保留
M8089	Y4端口的输出初始化标志			D8089	保留
M8090	Y0输出完成中断使能			D8090	保留
M8091	Y1输出完成中断使能			D8091	保留
M8092	Y2输出完成中断使能			D8092	保留
M8093	Y3输出完成中断使能			D8093	保留
M8094	Y4输出完成中断使能			D8094	保留

M元件	M元件的描述	D元件	D元件的描述
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	SPD(X000)具有-脉冲个数/分钟	D8100	保留
M8101	SPD(X001)具有-脉冲个数/分钟	D8101	单板程序版本，如24100H2U=24，100版本V1.00
M8102	SPD(X002)具有-脉冲个数/分钟	D8102	系统提供给用户程序的程序容量
M8103	SPD(X003)具有-脉冲个数/分钟	D8103	保留
M8104	SPD(X004)具有-脉冲个数/分钟	D8104	DRVI，DRVA执行时加速时间[默认100]由M8135决定是否有效[Y0]
M8105	SPD(X005)具有-脉冲个数/分钟	D8105	DRVI，DRVA执行时加速时间[默认100]由M8135决定是否有效[Y1]
M8106	保留	D8106	DRVI，DRVA执行时加速时间[默认100]由M8135决定是否有效[Y2]
M8107	保留	D8107	DRVI，DRVA执行时加速时间[默认100]由M8135决定是否有效[Y3]
M8108	保留	D8108	DRVI，DRVA执行时加速时间[默认100]由M8135决定是否有效[Y4]
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）

M元件	M元件的描述	D元件	D元件的描述
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	为ON时485BD扩展卡有效	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减速时间有效[ON-PLSR，DRVI，DRVA]	D8135	
M8136	Y1减速时间有效[ON-PLSR，DRVI，DRVA]	D8136	Y000&Y001输出脉冲合计数
M8137	Y2减速时间有效[ON-PLSR，DRVI，DRVA]	D8137	
M8138	Y3减速时间有效[ON-PLSR，DRVI，DRVA]	D8138	保留
M8139	Y4减速时间有效[ON-PLSR，DRVI，DRVA]	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	

M元件	M元件的描述	D元件	D元件的描述
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	D8170	保留
M8171	X001脉冲捕捉	D8171	保留
M8172	X002脉冲捕捉	D8172	保留
M8173	X003脉冲捕捉	D8173	本站站号设定状态
M8174	X004脉冲捕捉	D8174	通讯子站设定状态
M8175	X005脉冲捕捉	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

M元件	M元件的描述	D元件	D元件的描述
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

M元件	M元件的描述	D元件	D元件的描述
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

5.2 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(M8069ON 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	
	6203	通信数据的求和不一致	
	6204	数据格式有误	
	6205	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，
	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	

类型	出错代码	出错内容	处理方法
			JP0 若接通，COM0 只能做监控或下载口，且是 RS422 模式
	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配；
	6341	数据帧长度错误	
	6342	地址错误	
	6343	CRC 检验错误	
	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。 ④STL，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	

类型	出错代码	出错内容	处理方法	
		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 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	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。	
	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		
	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
	6732	输入滤波器常数(a)在目标范围外〔a<0 或 100<a)		

类型	出错代码	出错内容	处理方法	
	6733	比例阀(KP)在目标范围外〔KP<0)		PID 运算中产生数据错误。请检查参数。
	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		

5.3 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”。

5.4 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

序号	数据(字节)意义	字节数量	说明
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。

5.5 H2U 系列 3A/6A/6B 扩展卡使用说明

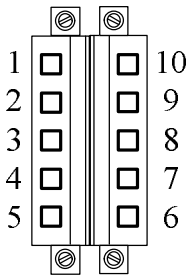
模拟量扩展卡选件可实现在主模块上的模拟量信号的输入和输出功能，端口配置如下：

型号	独立通道数	输入端口		输出端口	
		0～10V 型	0～20mA 型	0～10V 型	0～20mA 型
H2U-3A-BD	2 入 1 出	V1/V2	I1/I2	V1	I1
H2U-6A-BD	4 入 2 出	V1/V2	I3/I4	V1/V2	I1/I2
H2U-6B-BD	4 入 2 出	—	I1/I2/I3/I4	V1/V2	I1/I2

PLC 主模块与扩展卡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

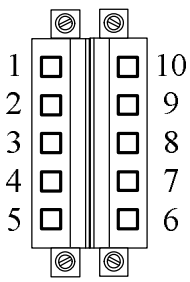
和收发处理，使用时，用户程序只需在特定的系统缓冲区写入需要访问的扩展小板类型、输出值，在特定的寄存器中读取输入端口的检测数据即可。

3A 扩展卡引脚功能：



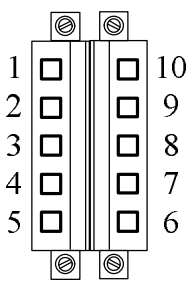
引脚	名称	功能描述	引脚	名称	功能描述
1	V1＋	Ch1 电压信号输入端	10	V2＋	Ch2 电压信号输入端
2	I1＋	Ch1 电流信号采样电阻端	9	I2＋	Ch2 电流信号采样电阻端
3	V1-	Ch1 信号输入参考端	8	V2-	Ch2 信号输入参考端
4	VO＋	AO 电压信号输出端	7	IO＋	AO 电流输出端
5	GND	AO 电压信号参考端	6	NC	功能保留

6A 扩展卡引脚功能：



引脚	信号	功能描述	引脚	信号	功能描述
1	V1＋	Ch1 电压信号输入端	10	VO2＋	Ch2 电压输出端
2	V2＋	Ch2 电压信号输入端	9	IO2＋	Ch2 电流输出端
3	I3＋	Ch1 电流信号采样电阻端	8	V01＋	Ch1 电压输出端
4	I4＋	Ch2 电流信号采样电阻端	7	IO1＋	Ch1 电流输出端
5	GND	输入公共接地端	6	GND	输出公共接地端

6B 扩展卡引脚功能：



引脚	信号	功能描述	引脚	信号	功能描述
1	I1＋	Ch1 电流信号输入端	10	VO2＋	Ch2 电压输出端
2	I2＋	Ch2 电流信号输入端	9	IO2＋	Ch2 电流输出端
3	I3＋	Ch3 电流信号输入端	8	V01＋	Ch1 电压输出端
4	I4＋	Ch4 电流信号输入端	7	IO1＋	Ch1 电流输出端
5	GND	输入公共接地端	6	GND	输出公共接地端

访问扩展卡时的相关寄存器及其定义：

序号	数据名称	PLC 发送到扩展卡	访问 H2U-3A-BD 时的写入值	访问 H2U-6A-BD 时的写入值	访问 H2U-6B-BD 时的写入值
1	控制字	D8220		/	/
2	数据 1	D8221	模拟量输出通道 1 的输出设定值（量程：0-10000）	模拟量输出通道 1 的输出设定值（量程：0-10000）	模拟量输出通道 1 的输出设定值（量程：0-10000）
3	数据 2	D8222	/	模拟量输出通道 2 的输出设定值（量程：0-10000）	模拟量输出通道 2 的输出设定值（量程：0-10000）
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，用户无需关心		

扩展卡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 应答的数据	H2U-6A-BD 应答的数据	H2U-6B-BD 应答的数据
1	控制字	D8230	3A21h	6A42h	6B42h
2	数据 1	D8231	模拟量输入通道 1 的当前值（量程：0-10000）	模拟量输入通道 1 的当前值（量程：0-10000 ） 对应 0-10V	模拟量输入通道 1 的当前值（量程：0-10000）对应 0-20mA
3	数据 2	D8232	模拟量输入通道 2 的当前值（量程：0-10000）	模拟量输入通道 2 的当前值（量程：0-10000 ） 对应 0-10V	模拟量输入通道 2 的当前值（量程：0-10000）对应 0-20mA
4	数据 3	D8233	模拟量输出通道 1 的输出设定值（量程：0-10000）	模拟量输入通道 3 的当前值（量程：0-10000 ） 对应 0-20mA	模拟量输入通道 3 的当前值（量程：0-10000）对应 0-20mA
5	数据 4	D8234	/	模拟量输入通道 4 的当前值（量程：0-10000）对应 0-20mA	模拟量输入通道 4 的当前值（量程：0-10000）对应 0-20mA
6	数据 5	D8235	/	模拟量输出通道 1 的输出设定值（量程：0-10000）对应 0-10V ， 或对应	模拟量输出通道 1 的输出设定值（量程：0-10000）对应 0-10V，或对应 0-20mA

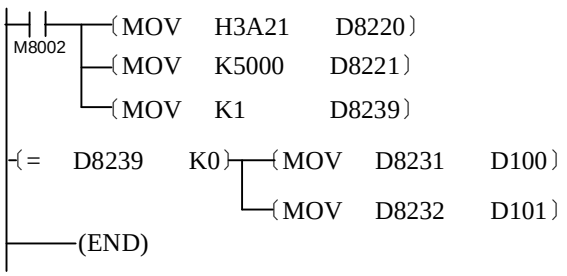
序号	数据名称	PLC 从扩展卡接收	H2U-3A-BD 应答的数据	H2U-6A-BD 应答的数据	H2U-6B-BD 应答的数据
				0-20mA	
7	数据 6	D8236	/	模拟量输出通道 2 的输出设定值（量程：0-10000）对应 0-10V，或对应 0-20mA	模拟量输出通道 2 的输出设定值（量程：0-10000）对应 0-10V，或对应 0-20mA
8	数据 7	D8237	/	/	/
9	数据 8	D8238	/	/	/
10	数据 9	D8239	通信错误计数，若通信正常，自动置 0；非 0，表示通信连续出错次数	通信错误计数，若通信正常，自动置 0；非 0，表示通信连续出错次数	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

由上面表格的说明可知，3A/6A/6B 扩展卡的使用编程方法相同，只需要向系统的专用 D 变量进行读（输入）或写（输出）操作即可，以下以 3A 扩展卡的编程举例来说明使用方法。

应用编程举例 1：

使用 H2U-3A-BD 卡，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 D101。

程序如下：

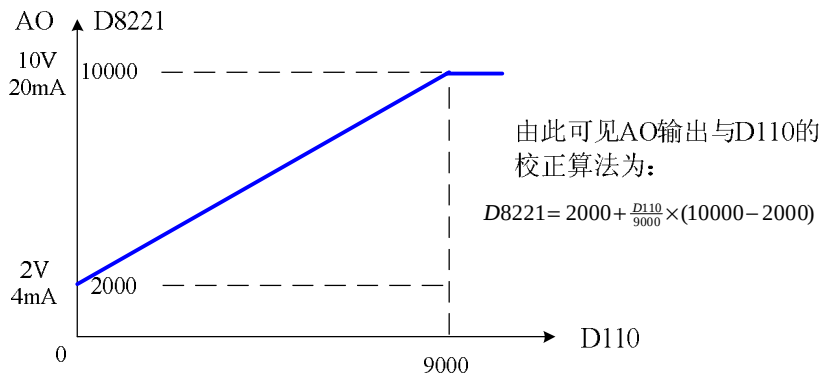


说明：当 D8239 的值非 0 时，表示 PLC 尚未正确读取到扩展小板的值，编程时需注意；AO 通电的输出值 D8221 可根据具体应用在用户程序中进行刷新。

应用编程举例 2：

接上例，使用 H2U-3A-BD 卡，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；但要求其 AO 电流输出与 D110 寄存器当前值相关，当 D110＝0 时，输出 4mA；当 D110＝9000 时，输出 20mA。

分析：AO 的电流输出信号与电压输出信号，分别是〔0～20mA〕或〔0～10V〕对应 D8221 专用寄存器的〔0～10000〕，若要 4～20mA 对应用户的寄存器值，需要用户程序作校正后写入 D8221，才能得到希望的信号电流。（注意：校正后 AO 的电压信号特性也发生了变化。）



程序如下：

```
M8002
──┤├──〔MOV   K1        D8239〕

〔=   D8239   K0〕──〔MOV   D8231   D100〕
                   〔MOV   D8232   D102〕

M8000
──┤├──〔>   D110   K8000〕──〔MOV   K8000   D110 〕
   〔MOV   D110   D0〕          //计算中，乘运算结果会超过1word，需采用32bit运算
   〔MOV   K0     D1〕          //将D110放（D1.D0）32bit临时单元
   〔DMUL  D0     K8000   D2〕  //将（D1.D0）乘8000，结果存（D3.D2）
   〔DDIV  D2     K9000   D4〕  //将（D3.D2）除以9000，结果存（D5.D4）临时单元
   〔ADD   D4     K2000   D8221〕 //将（D4）加2000，写入AO单元

──(END)
```

5.6 H2U 通讯扩展卡说明

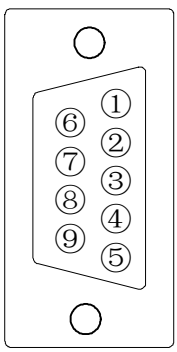
1、H2U-232-BD 扩展卡

H2U-232-BD 扩展卡为 H2U 系列 PLC 主模块用通讯扩展卡，该卡的 DB9 信号插座提供了标准的 RS232 电平，在物理上是 PLC 主模块的 COM1 通讯端口，因此该板的通讯口与 PLC 原配 RS485 电平的 COM1 口不能同时使用，否则会发生通讯冲突。使用该扩展板，能实现 PLC 与 PC 机、HMI、MODEM、PLC、智能仪表等设备的通讯，通讯格式可由编程决定。

产品规格：

端口接头型式	DB9 型，公座
信号标准	RS-232C
传输距离	与波特率有关，最远为 15 米
通讯方式	支持全双工、半双工方式
协议	MODBUS 主、从协议；用于自定义协议
工作电源	5VDC，PLC 内部电源提供
信号隔离	与 PLC 内部逻辑电路不隔离

引脚定义：



引脚号	信号名称	功能描述
1	CD	数据载波检查端（输入）
2	RXD	接收数据端口（输入）
3	TXD	发送数据端口（输出）
4	DTR	发送数据请求（输出）
5	GND	信号地
6	DSR	发送使能（输入）
7，8	—	两引脚内部直接联接
9	—	内部无联接

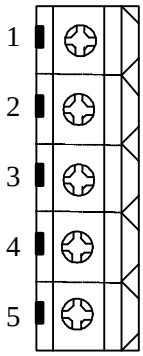
与电脑、触摸屏等的连接方式和设置请参考《H2U 通讯扩展卡用户手册》

2、H2U-485-BD 扩展卡

H2U-485-BD 扩展卡为 H2U 系列 PLC 主模块用扩展通讯卡，该卡提供了 RS485 电平，将 TX 和 RX 信号线分别连接，又可作为 RS422 信号端口使用。在物理上是 PLC 主模块的 COM1 通讯端口，因此该板的通讯口与 PLC 原配 RS485 电平的 COM1 口不能同时使用，否则会发生通讯冲突。使用该扩展板，能实现 PLC 与 PC 机、HMI、PLC、智能仪表等设备的通讯，通讯格式可由编程决定。此卡需要 M8126 为 ON 时才有效。

端口功能定义：

引脚号	信号名称	功能描述
1	RA	485 信号的 RXD 信号＋
2	RB	485 信号的 RXD 信号－
3	TA	485 信号的 TXD 信号＋
4	TB	485 信号的 TXD 信号－
5	GND	信号地



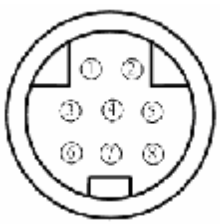
与电脑、触摸屏等的连接方式和设置请参考《H2U 通讯扩展卡用户手册》

3、H2U-422-BD 扩展卡

H2U-422-BD 扩展卡是 H2U 系列 PLC 主模块用通讯扩展卡，该卡的 Mini DIN8 信号插座提供了 RS422 电平，在物理上是 PLC 主模块的 COM1 通讯端口，因此该板的通讯口与 PLC 原配 RS485 电平的 COM1 口不能同时使用，否则会发生通讯冲突。使用该扩展板，能实现 PLC 与 PC 机、HMI、PLC、智能仪表等设备的通讯，通讯格式可由编程决定。H2U 系列 PLC 使用该卡后，可使得 PC 和 HMI 与之同时通讯。

端口功能定义：

引脚号	信号名称	功能描述
1	RB	485 信号的 RXD 信号－
2	RA	485 信号的 RXD 信号＋
3	GND	信号地
4	TB	485 信号的 TXD 信号－
5	VDD	5V＋
6	NC	NC
7	TA	485 信号的 TXD 信号＋
8	NC	NC
金属壳	GND	信号地



该插座引脚的功能定义与
COM0通讯口插座的定义相同

与电脑、触摸屏等的连接方式和设置请参考《H2U 通讯扩展卡用户手册》。

5.7 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	2416MT/3624MT	1616MT/3232MT	MTQ 型
X0	100kHz	100kHz	100kHz
X1	100kHz	100kHz	100kHz
X2	10kHz	100kHz	100kHz
X3	10kHz	100kHz	100kHz
X4	10kHz	100kHz	100kHz
X5	10kHz	100kHz	100kHz
其它 X 端口	指标相同		
Y0	100kHz	100kHz	100kHz
Y1	100kHz	100kHz	100kHz
Y2	—	100kHz	100kHz
Y3	—	—	100kHz
Y4	—	—	100kHz
其它 Y 端口	指标相同		

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
C251～C255	10kHz	30kHz	30kHz

支持指令比较：

指令	2416MT/3624MT 支持的输出端口	1616MT/3232MT 支持的输出端口	MTQ 支持的 输出端口
PLSY	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
PLSR	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
PLSV	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
PWM	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
DRVI	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
DRVA	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4
ZRN	Y0、Y1	Y0、Y1、Y2	Y0、Y1、Y2、Y3、Y4

MT/MTQ 型高速脉冲输出时涉及的系统监控寄存器说明：

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	PLSY&PLSR输出Y003对应的脉冲个数累积值
M8152	Y002脉冲输出停止	D8152	
M8153	Y003脉冲输出停止	D8153	PLSY&PLSR输出Y004对应的脉冲个数累积值
M8154	Y004脉冲输出停止	D8154	
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]

5.8 H2U 系列高速处理指令的增强功能说明

H2U 系列 PLC 高速编程指令除了能兼容 FX1n/FX2n 的指令之外，对其中部分指令功能作了加强，这些加强功能一般通过系统的 M 变量来配合实现。涉及的高速指令功能如下：

指令	配合使用的 M 变量	M 标志为 ON 时指令的加强功能
SPD	M8100~M8105 分别对应 X0～X5 端口	新增 1 分钟脉冲个数（运行速度）计算，该数据是通过实时采样输入端脉冲频率再通过内部运算得出的。
PLSY	M8135~M8139 分别对应 Y0~Y4	可以在运行中更改输出脉冲个数。
	M8085~M8089 分别对应 Y0~Y4	在驱动特殊位为 ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理。
	M8090~M8094 分别对应 Y0~Y4	可以实现脉冲输出完成后产生一次用户中断。
PLSR DRVI DRVA	M8135~M8139 分别对应 Y0~Y4	可以在运行中更改输出脉冲个数； 可以更改减速时间（使加减速时间不同）。
	M8085~M8089 分别对应 Y0~Y4	在驱动特殊位为 ON，可以马上启动下条脉冲输出指令，不需要上条指令能流无效的处理。
	M8090~M8094 分别对应 Y0~Y4	可以实现脉冲输出完成后产生一次用户中断。
高速计速器中断	M8084； D8084~D8086	高速计速器多用户中断(最大支持 24 个：I507~I530)

涉及的加强高速指令使用说明如下：

1.SPD 指令的描述

16bit	32bit	P	FNC56	**SPD**	脉冲密度，转速，线速度
✔					
7 步			指令格式：**SPD** (S1) (S2) (D)		

X000~X005分别使用了功能使能标志M8100~M8105作为增强功能的生效标志，每个标志可单独设置。

1）．如果功能使能标志为[M8100~M8105**]OFF，SPD 为基本功能：**

(S1) ：脉冲信号输入端口，只能为X0~X5；

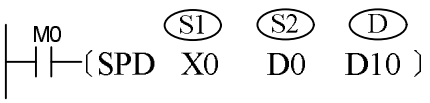
(S2) ：为设定的脉冲检测时间长度(ms)1~32767；

(D)+0：为在S2时间内的脉冲个数，为16位的数据；

(D)+I：本次时间段对脉冲的计数。

(D)+2：用于测定剩余时间(ms)。

指令举例



X0 为脉冲信号输入端口；

D0 为单位时间(ms)；

D10 为 D0 时间段总脉冲个数；

D11 为 D0 时间段正在计的数据；

D12 为时间段剩于的时间(MS)；

2）.如果功能使能标志为[M8100~M8105**]ON，SPD为加强功能：**

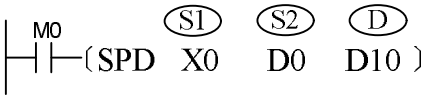
(S1)S1：脉冲信号输入端口，只能为X0~X5；

(S2)S2：为设定的脉冲检测时间长度1~32767(ms)；

(D)+0：为在S2时间内的脉冲个数，为16位的数据；

(D)+1，(D)+2：为每分钟内的脉冲个数，为32位的数据；

指令举例：



X0 为脉冲信号输入端口；

D0 为单位时间 1~32767(ms)；

D10 为 D0 时间段总脉冲个数；

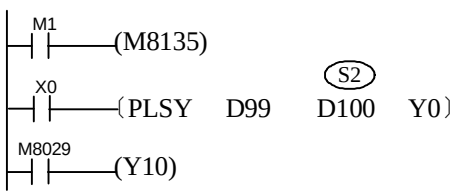
D11，D12 为运行频率=1 分钟脉冲个数*10(0.1 为单位)；

2.PLSY 指令的特殊说明

16bit	32bit	P	FNC57	**PLSY**	脉冲输出
✔	✔				
7 步	13 步		指令格式：**PLSY** S1 S2 D		

1）．通过使用特殊位 M8135~M8139(分别对应 Y0~Y4)为 ON，可以实现如下功能：

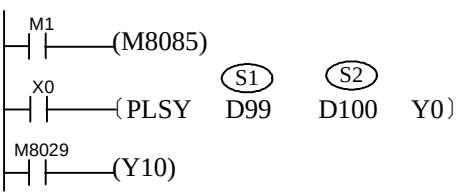
可以在运行中更改输出脉冲个数(可改大，也可改小)。编程举例：



M1 为 ON，该特殊功能有效，此时 X0 导通执行 PLSY 指令，脉冲发送过程中可以更改 D100（脉冲个数），脉冲个数可以增加也可以减少；（注意：更改后的 D100 的数据要大于该指令已经发送的脉冲个数）

2）．通过使用特殊位 M8085~M8089(分别对应 Y0~Y4)为 ON，可以实现如下功能：

在驱动特殊位为 ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理。编程举例：



当 X0=1，PLSY 在指令发脉冲过程中，或脉冲发送完毕，若 M8085 被置位，PLSY 指令就会重启，按设定的频率和脉冲数 S1、S2（即 D99、D100 的当前值）重新开始发送脉冲，而无需指令驱动 X0 由 ON à OFF à ON 变化一次。

3）.通过使用特殊位 M8090~M8094(分别对应 Y0~Y4)为 ON，可以实现如下功能：

可以实现脉冲输出完成后执行一次用户中断；具体对应如下：

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

当特殊功能位（M8090-M8094）为 ON，则在指定脉冲个数发送完毕后，立即执行用户中断 I502～I506。

3.PLSR，DRVI，DRVA 指令的特殊说明

1）．通过使用特殊位 M8135~M8139(分别对应 Y0~Y4)为 ON，可以实现如下功能：

可以在运行中更改输出脉冲个数(可大可小)，同时减速时间分别由如下寄存器定义，时间单位为 ms：

端口号	使用特殊位	修改减速时间 用寄存器
Y000	M8135	D8165
Y001	M8136	D8166
Y002	M8137	D8167
Y003	M8138	D8168
Y004	M8139	D8169

当特殊位(M8135～M8139)为 ON 时，PLSR，DRVI，DRVA 指令执行过程中，可以随时更改脉冲总数(注意：更改后的脉冲总数必须大于当前已经发送的脉冲数量)

加速时间和减速时间可以不同，但是必须在指令执行之前修改好加减速时间（加速时间所在的寄存器不变，减速时间对应寄存器为(D8165～D8169)。

2）．通过使用特殊位 M8085~M8089(分别对应 Y0~Y4)为 ON，可以实现如下功能：

在驱动特殊位为 ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理。

3）.通过使用特殊位 M8090~M8094(分别对应 Y0~Y4)为 ON，可以实现如下功能：

可以实现脉冲输出完成后执行一次用户中断;，具体对应如下：

端口号	使用特殊位	对应的用户中断
Y000	M8090	I502
Y001	M8091	I503
Y002	M8092	I504
Y003	M8093	I505
Y004	M8094	I506

4.高速计速器多用户中断的使用描述

为了满足在高速计速器运行时，支持多高速自由任务，实现了高速计速器多用户中断(最大支持 24 个，均为扩展的中断号)，设定和比较用数据表格的方式定义：

标志位	使用描述
M8084	为 ON 使能高速计速器多用户中断
D8084	为高速计数器序号 235~255
D8085	对应的用户中断个数，最大 24 个，从 I507~I530
D8086	对应多个比较点数据的序号，只能为 D 元件，且为双字宽度，如 200 为 D200 开始的双字

比较点数据存放的例程：

使用高速计数器 C235 时，D8084=235；使用 D200 开头的 32 位寄存器作为数据比较值时，D8086=200；要执行 5 个中断输出时，D8085=5；最后 M8084=ON 该功能有效

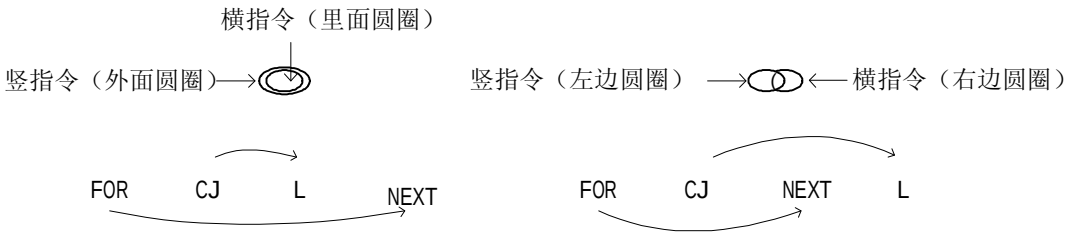
C235 的数据	记录单元	存放单元数值	对应的用户中断	D8131 的数值
100	D200，D201	=100	I507	0
200	D202，D203	=200	I508	1
300	D204，D205	=300	I509	2
400	D206，D207	=400	I510	3
500	D208，D209	=500	I511	4→0(M8133=ON)

每个中断可以由高速计数器的数值和记录单元的数值产生。

注意：该表格执行时，首先从上到下依次执行，只有每一格数据比较成功后才会继续往下一格执行，表格的最后一格执行完成后 M8133=ON，此时数据比较又返回到表格的第 1 格，建议在返回到第 1 格时把 C235 数据清零，这样表格就会循环执行。该功能只有在增计数时才有效！

5.9 程序流程控制指令的相互关系

各种程序流程控制指令之间的相互关系如下所示。下表中 表示包含关系 表示前后区间重复。在 AutoShop 编程环境中子程序和中断程序单独写，故没有 P-SRET、I-IRET、FEND-END、O-FEND、O-END，



竖＼横	MC-MCR	CJ-L	EI-DI	FOR-NEXT	STL-RET
MC-MCR	○ △	○ △	○ ○	○ ×(6607)	○ ×(6605)
CJ-L	○ △	○ △	○ ○	○ △	○ △
EI-DI	○ ○	○ ○	○ ○	○ ○	○ ○
FOR-NEXT	×(6607) ×(6607)	○ △	○ ○	○ ※2	×(6607) ×(6607)
STL-RET	×(6605) ×(6605)	△ △	○ ○	○ ×(6607)	○ ○
P-SRET	×(6606) ×(6608)	○ △	○ ○	○ ×(6607)	×(6606) ×(6605)
I-IRET	×(6606) ×(6606)	○ △	○ ○	○ ×(6607)	×(6606) ×(6606)
FEND-END	○ ×(6608)	○ ×(6701)	○ ※1	○ ×(6607)	△ ×(6605)
O-FEND	○ ×(6608)	○ ○	○ ○	○ ×(6607)	○ ×(6605)
O-END (无FEND)	○ ×(6608)	○ ×(6701)	○ ※1	○ ×(6607)	○ ×(6605)

○ 能够没有问题的互相组合使用。
× 禁止使用的组合，（）之中为错误序号。
△ 虽然不属于禁止范围，但是可能会导致动作的复杂化，建议尽量避免组合使用。

横 / 竖	P-SRET	I-IRET	FEND-END	备注
MC-MCR	◯ × (6608) ◯ × (6606)	◯ × (6608) ◯ × (6606)	◯ × (6608) ◯ × (6608)	※1 为忘记DI的状态。不属于错误。 ※2  按照实线所示方式动作。 ※3 最初的FEND和END有效，不是目的的程序。不属于错误。 除了部分例外，具有包含关系的指令可以进行组合使用。 但是必须注意以下特例情况。 ①在FOR-NEXT,STL-RET, P-SRET,I-RET中无法使用 MC-MCR。 ②在 FOR-NEXT,P-SRET,I-IRET 中无法使用STL-RET。 ③在 MC-MCR,FOR-NEXT,P-SRET,I-IRET 指令间无法使用 I,IRET,SRET,FEND,END 等指令进行屏蔽操作。
CJ-L	◯ △ ◯ △	◯ △ ◯ △	◯ × ◯ ○	
EI-DI	◯ ○ ◯ ○	◯ ○ ◯ ○	◯ ※1 ◯ ○	
FOR-NEXT	◯ × (6607) ◯ × (6701)	◯ × (6607) ◯ × (6607)	◯ × (6607) ◯ × (6607)	
STL-RET	◯ × (6605) ◯ × (6606)	◯ × (6605) ◯ × (6606)	◯ × (6605) ◯ × (6605)	
P-SRET	◯ × (6606) ◯ × (6606)	◯ × (6606) ◯ × (6606)	◯ × (6709) ◯ × (6709)	
I-IRET	◯ × (6606) ◯ × (6606)	◯ × (6606) ◯ × (6606)	◯ × (6606) ◯ × (6606)	
FEND-END	◯ ○ ◯ × (6709)	◯ ○ ◯ × (6709)	◯ ※3 ◯ ※3	
0-FEND	◯ × (6606) ◯ × (6709)	◯ × (6606) ◯ × (6606)	◯ ※3 ◯ ※3	
0-END (无FEND)	◯ × (6606) ◯ × (6709)	◯ × (6606) ◯ × (6706)	◯ ※3 ◯ ※3	

5.10 部分特殊继电器和寄存器功能说明

系统运行状态

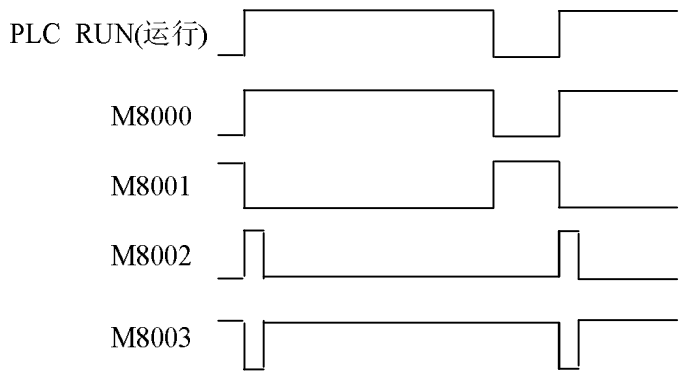
PLC的运行标志 M8000-M8003

1. M8000：M8000为RUN中常时ON接点，即运行监视常开接点（A接点），PLC在RUN的状态下，M8000一直保持为ON。

2. M8001：M8001为RUN中常时OFF接点，即运行监视常闭接点（B接点），PLC在RUN的状态下，M8001一直保持为OFF。

3. M8002：PLC开始RUN的第一次扫描ON，之后保持为OFF。该脉冲的宽度为一次扫描时间，当要作各种初始设置工作时可以使用本接点。

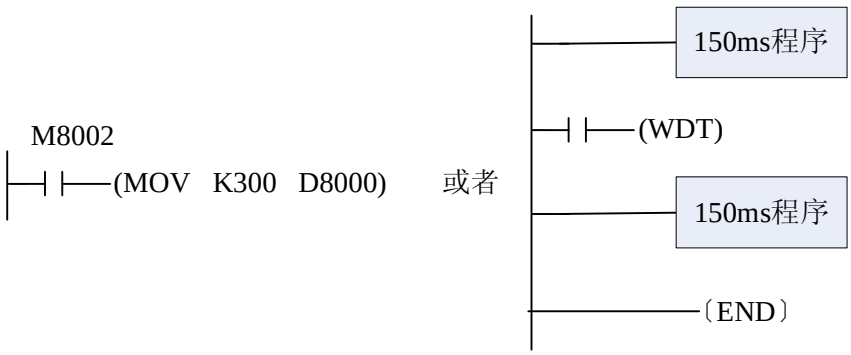
4. M8003：PLC开始RUN的第一次扫描周期OFF，之后一直ON。即启始负向（RUN的瞬间'OFF'）脉冲。



监控定时器 D8000

1、 监控定时器专门用来监视PLC的扫描时间，当扫描时间超过监控定时器的设置时，ERROR红色指示灯长亮，输出全部变成Off。

监控定时器时间的初始值为200ms，当指令运算过于复杂或者是PLC主机连接众多的特殊模块时都会造成扫描时间过长，扫描时间是否超过D8000的设置值，请监视D8010~D8012。此种情况下，可于程序中使用MOV指令来变更监控定时器的设置值，将监控定时器的设置值变更为300ms，也可于PLC程序中加入WDT指令，当CPU执行至WDT指令时，内部监控定时器被清除为零，使得扫描时间不会超过监控定时器的设置时间。。



2、 监控定时器最大可设置至32767ms，但必须注意，监控定时器设置过大时，运算异常发生的检出时机将会跟着被拖慢。因此，若非复杂的运算使得扫描时间超过200ms，一般的情况下请维持在200ms以下较佳。

单板程序版本 D8001

单板程序版本，如D8001=24115的含义是：24为H2U，115为版本V1.15，即版本号为V1.15的H2U型PLC。

程序容量 D8002

程序容量，4K、8K、16K、24K等，H2U型号PLC的容量是24K

语法检查信号 M8004、D8004

M8004：当错误信号M8060~M8067(除了M8062外)中任意一个为ON时，M8004为ON。可用来监控PLC中有无系统错误。

D8004：M8060~M8067的BCD值，初始值为0。

电压电池检测M8005~M8009 D8005~D8009 M8030

1、当电池电压D8005的检测值低于D8006（初始值为2.6V）时，M8005有输出；

2、电池电压低报警有出现（M8005为ON）过，M8006就置ON（锁存），PLC掉电再上电时复位，程序无法对其进行复位；

3、当交流电失电5ms后M8007、M8008动作，失电时间在D8008之内PLC程序继续运行、在D8008之外用户程序不执行，M8000为OFF；

4、扩展模块24V掉电时M8009为ON，D8009记录掉电扩展模块的编号；

5、M8030为ON时，屏蔽电池电压低的警告。

系统时钟

扫描时间监视 D8010~D8012

程序扫描时间的现在值、最小值及最大值被存放在D8010~D8012当中。

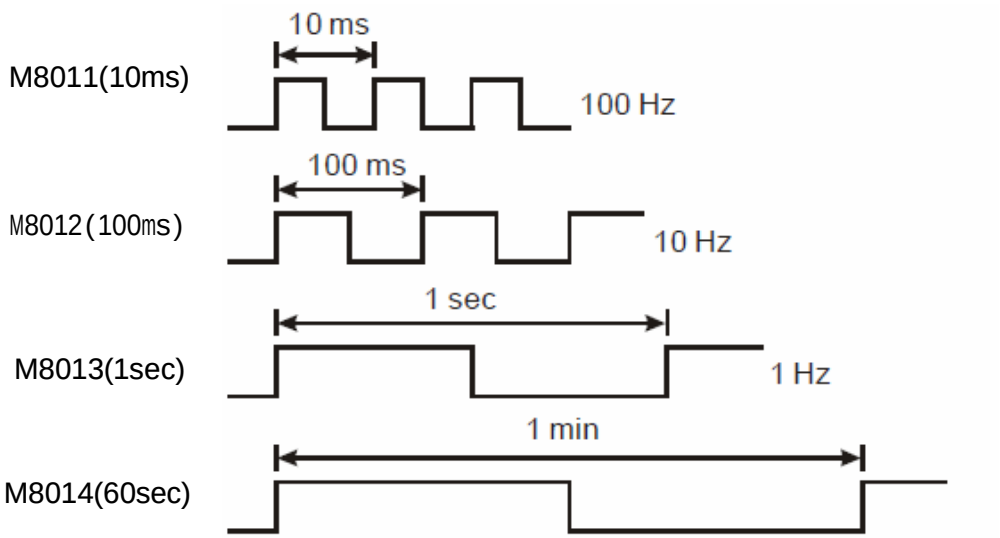
1. D8010：扫描时间的现在值；

2. D8011：扫描时间的最小值；

3. D8012：扫描时间的最大值。

周期振荡时钟 M8011~M8014

PLC主机内部均具备下列4种振荡时钟，只要PLC通上电源，这4种振荡时钟就会自动动作。PLC处于STOP状态下，振荡时钟也会动作，所以振荡时钟启动时序及RUN的启动时序并不会同步。

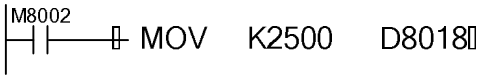


实时时钟 D8013~D8019 M8015~M8019

1、M8015为ON时，时钟计时停止；

2、M8017每On一次，PLC内部时钟作±30秒校正动作，这里的校正是指当PLC的内部时钟的秒针D8013在1~29时，会被自动归为“0”秒而分针不变、D8013在30~59时，也会被自动归为“0”秒，分针加1分钟。

3、通常的情况下年只显示2位数（例：2009年只显示09），若需要将“年”采用显示4位的格式，仅需在一个扫描周期执行如下语句，就可以切换成4位显示格式：



请在可编程控制器每次运行时执行本程序。而且即使传送K2000使显示切换至公历4位，也不会影响当前时间。、

地址号	名称	动作功能
M8015	时钟停止和时间校准	ON 时时钟停止，利用 ON→OFF 的边沿写入时间，执行再动作。
M8016	时间显示停止	ON 时时钟显示停止（计时保持动作）
M8017	±30 秒修正	利用 OFF→ON 的边沿修正秒。（当秒数为 0~29 秒时变为 0 秒。当秒数为 30~59 秒时向分进一位，将秒变为 0 秒）
M8018	安装检侧	常时处于 ON 状态
M8019	RTC 错误	校准时间时，当特殊数据寄存器的数据超过设定范围时变为 ON。

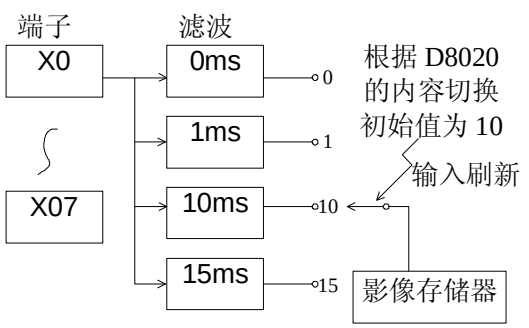
地址号	名称	设定值的范围	动作功能
D8013	秒	0~59	用于写入校准时间的初始值，或读取当前时间。
D8014	分	0~59	
D8015	时	0~23	
D8016	日	0~31	
D8017	月	0~12	
D8018	年	0~99（公历后两位）	
D8019	星期	0~6（对应日~六）	

指令标志

输入滤波调整　D8020

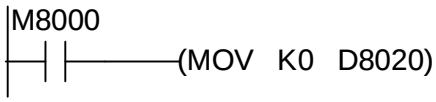
1、X0~X7输入端，可由D8020的内容来设置输入端接收脉冲的反应时间，设置范围0~60ms。默认10ms，

2、当程序中使用高速计数器、中断插入等功能时，则相关端口的滤波时间自动为最短时间，无关的X0~X7端口的滤波时间仍为原设定值D8020。



3、PLC电源Off→On变化时，D8020的内容自动变成10

4、若执行以下程序，滤波常数变为 0ms。但是实际上该输入端设置了 C-R 滤波器硬件，即使是将其指定为 0，但也不会低于 10ms（40 点、60 点 PLC 的 X2~X5 不会低于 50ms）。



运算标志 M8020~M8022

在所有运算中的各种标志的动作如下。

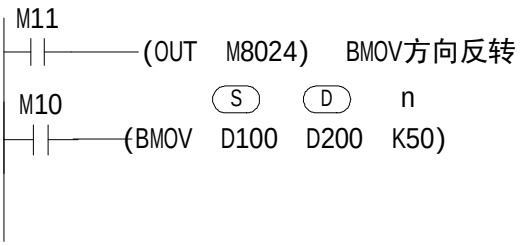
1、零标志：结果值为 0 时，M8020=ON

2、借位标志：结果值超出最小处理单位时，M8021=ON

3、进位标志：结果绝对值超出最大使用范围时，M8022=ON

指令赋值换向 M8024

通过控制BMOV指令的反向标记M8024，可用一个程序指令向两个方向传送数据。

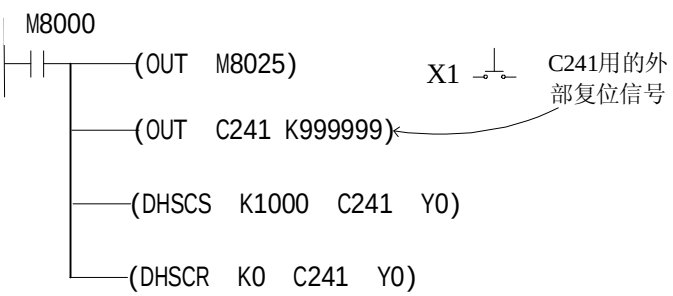


M8024（OFF）：(S) → (D) 读出 D100 D200

M8024（ON）：(S) → (D) 写入 D200 D100

HSC指令模式 M8025

高速计数器的输出接点、FNC53（D HSCS）、FNC54（D HSCR）、FNC55(D HSZ)指令中的比较输出，都随计数输入的当前值寄存器的变化而动作。即使通过传送指令 DMOV 改变当前值（C235~C255），只要没有计数输入，比较输出就不发生变化。这正如前面“注意事项”讲述的那样，关于高速计数器 C241 等，备有外部复位端子（R），通过复位输入信号的上升沿，执行指令、输出比较结果。详见以下内容。外部复位模式 C241 用的外部复位端子。



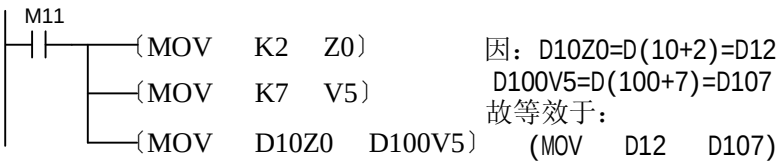
在上例中，当 M8025 为 ON 时，C241 的现在值假如为 2000，此时 Y0 为 ON，如果按下外部复位按钮 X1，C241 的现在值变为 0，即使 X0 没有计数输入，此时 Y0 也复位。

执行完毕标志 M8029

高速脉冲输出、通讯、MTR、HKY、DSW、SEGL、PR等指令执行完毕标志。

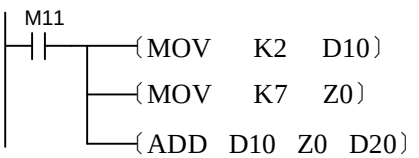
变址寄存器 D8028、D8029 D8182~D8195

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，在程序中我们一般用Z或者V来进行变址修饰：



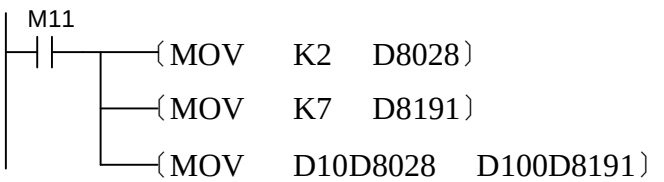
因：D10Z0=D(10+2)=D12
D100V5=D(100+7)=D107
故等效于：
(MOV D12 D107)

亦可以将Z、V当普通寄存器使用



因：D10=2 Z0=7
故D20=2+7=9

Z、V有对应的特殊寄存器，如第一个程序等效于



D8028为Z0寄存器内容；D8029为V0寄存器内容；

D8182为Z1寄存器内容；D8183为V1寄存器内容；

D8184为Z2寄存器内容；D8185为V2寄存器内容；

D8186为Z3寄存器内容；D8187为V3寄存器内容；

D8188为Z4寄存器内容；D8189为V4寄存器内容；

D8190为Z5寄存器内容；D8191为V5寄存器内容；

D8192为Z6寄存器内容；D8193为V6寄存器内容；

D8194为Z7寄存器内容；D8195为V7寄存器内容；

位状态 M8031~M8034

1、 M8031为ON时，清除所有非停电保持的寄存器、继电器等；

Y、一般用M、一般用S接点状态

一般用T的接点及计时线圈

一般用C的接点及计数线圈及复位线圈

一般用D的现在值寄存器

一般用T的现在值寄存器

一般用C的现在值寄存器

2、 M8032为ON时，清除所有停电保持的寄存器、继电器等；

停电保持用M、S的接点状态

累计型定时器T的接点及计时线圈

停电保持用C及高速计数器C的接点、计数线圈

停电保持用D的现在值寄存器

累计型定时器T的现在值寄存器

停电保持用C及高速计数器C的现在值寄存器

3、 M8033为ON时，停机状态所有的软元件保持运行前的状态不变

所有Y、M、S的接点状态

所有T的接点及计时线圈

所有C及高速计数器C的接点、计数线圈

所有D、T、C的现在值寄存器

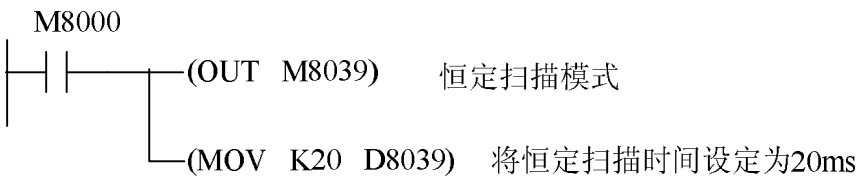
4、 M8034为ON时，所有输出点Y变成OFF。

5、 通过强制RUN/STOP操作使M8035(强制RUN模式)和M8036(强制RUN)变为ON，起动可编程控制器。

6、 将M8037(强制STOP)置于ON，可以停止可编程控制器的运行。

恒定扫描模式　M8039　D8039

驱动辅助继电器M8039，将目标扫描时间（以1ms为单位）预先写入数据寄存器D8039中，则可编程控制器的运算周期不会低于该值。即使运算提早结束时，也在剩余时间内进行等待，扫描时间够D8039的值后才返回0步。如果D8039的内容小于实际上程序的扫描时间时，则以实际上程序的扫描时间为主。



和扫描时间有关的指令RAMP、HKY、SEGL、ARWS、PR，应用上必须用“固定扫描时间模式”或者是和“定时插入中断”搭配使用。特别是HKY指令，它是以4×4矩阵方式作16个数字按钮的输入操作，使用时扫描时间必须固定在20ms以上。D8010~D8012所显示的扫描时间也包括固定的扫描时间。

步进阶梯

步进 M8040~M8049 D8040~D8049

1、驱动M8040后，即使具备了转移条件也无法进行状态之间的转移。而且停止状态内的输出将继续其动作。

2、步进开始，IST指令用标志M8041

3、启动脉冲，IST指令用标志M8042

4、原点回归状态结束，IST指令用标志M8043

5、检测到机械原点动作，IST指令用标志M8044

6、所有输出复位禁止，IST指令用标志M8045

7、当M8047为ON后，S0~S999中任何一个为ON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

8、当M8049为ON后，S900~S999中任何一个为ON时，M8048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

9、D8040~D8047：将S0~S999的最小动作地址号保存在D8040中，其他依次。最大的报警地址号保存在D8047中。

10、当M8049为ON后，D8049保存S900~S999的报警的最小地址号

中断禁止

禁止中断 M8050~M8059

使能了相对应的中断禁止以后，即使有中断信号，此时也不会产生中断，例如M8050为ON后，即使X0有中断脉冲如入，I00□也不会产生。

各个对应的中断定义如下：

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I6口口中断禁止	定时中断0	

中断允许/禁止设置			
M8057	驱动I7口口中断禁止	定时中断1	
M8058	驱动I8口口中断禁止	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

但X0~X5的脉冲捕捉功能不受中断禁止的限制。

联机功能

并联协议 M8070~M8073 M8162 D8070

并联协议，M8070置位为并联协议主站，M8071置位为并联协议从站，M8070与M8071在一台PLC中不能同时置位，若同时置位并联协议无效，若不存在其它优先协议，可通过D8126设置，D8126=50h为并联协议主站，D8126=5h为并联协议从站。

D8070：判断通讯出错的时间设定，默认为500

M8162：高速并联连接模式

	主站发送（从站接收）	从站发送（主站接收）
普通模式 M8162=0	M800~M899 D490~D499	M900~M999 M500~M509
高速模式 M8162=1	D490~D491	D500~D501

具体功能应用设置请看通讯部分并联协议的介绍。

N:N协议 D8076~D8180 M8183~M8191 D8201~D8218

用户只需要设置一台PLC为N：N协议主站，另外多PLC设置为N：N协议从站，且所有使用该协议的PLC串口连接起来，不需要用户程序干预，即可实现多台PLC间互相交换数据。

D8176：站点号，范围0~7，0表示主站点

D8177：从站点的总数，范围1~7，仅主站需要

D8178：刷新范围（模式）设置，范围0~2，仅主站需要

D8179：重试次数设定，仅主站需要

D8180：通信超时设置，*10ms，仅主站需要

M8183～M8190：通信出错标志，M8183对应第0号站点（主站），M8184对应第1号站点，

依次类推，M8190对应第7号站点。

M8191：正在执行数据传送

D8201：监控当前连接扫描时间

D8202：监控最大连接时间

D8203：监控主站通讯错误次数，最大计数到10000次就停止计数，

D8204~D8210：分别监控从站1~7通讯错误次数，最大计数到10000次就停止计数，

D8211：主站通讯错误代码

D8212~D8218：分别为从站1~7通讯错误代码。D8211~D8218的错误代码含义可查出错信息说明。

具体功能应用设置请看通讯部分N：N协议的介绍

特殊功能

输出口初始化 M8085~M8089

通过使用特殊位M8085~M8089(分别对应Y0~Y4)为ON，在PLSY，PLSR，DRVI，DRVA指令中可以实现如下功能：

在驱动特殊位为ON，可以马上启动下条脉冲输出指令，不需要上条能流无效的处理；

输出完成中断 M8090~M8094

通过使用特殊位M8090~M8094(分别对应Y0~Y4)为ON，在PLSY，PLSR，DRVI，DRVA指令中可以实现如下功能：

可以实现脉冲输出完成中断；具体对应如下：

端口号	使用特殊位	对应的用户中断
Y0	M8090	I502
Y1	M8091	I503
Y2	M8092	I504
Y3	M8093	I505
Y4	M8094	I506

具体功能设置请看H2U特殊功能的介绍

加减速时间M8135~M8139 D8104~D8108 D8165~D8169

通过使用特殊位M8135~M8139(分别对应Y0~Y4)为ON，在PLSY，PLSR，DRVI，DRVA指令中可以实现如下功能：

可以在运行中更改输出脉冲个数(可大可小，只能在加速和匀速的时候变，在减速的时候无效)；

同时在PLSR，DRVI，DRVA指令中减速时间分别由如下寄存器定义：

端口号	使用特殊位	对应的寄存器
Y0	M8135	D8165
Y1	M8136	D8166
Y2	M8137	D8167
Y3	M8138	D8168
Y4	M8139	D8169

在M8135为ON的时候，YO的减速时间由D8165决定，默认100ms。依次类推

在DRVI，DRVA指令中加速时间分别由如下寄存器定义：

端口号	使用特殊位	对应的寄存器
Y0	M8135	D8104
Y1	M8136	D8105
Y2	M8137	D8106
Y3	M8138	D8107
Y4	M8139	D8108

在M8135为ON的时候，YO的加速时间由D8104决定，默认100ms。依次类推

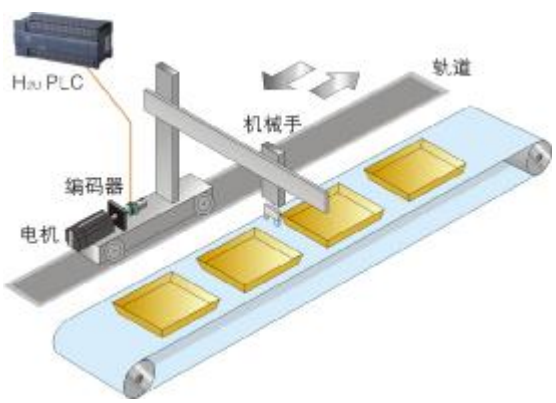
具体功能设置请看H2U特殊功能的介绍

高速计速器多用户中断M8084 D8084~D8086

为了满足在高速计速器运行时，支持多高速自由任务，可以设置任意一个高速计数器（C235~C255）产生多个中断，实现了如下使用规律：此功能命名为高速计速器多用户中断，最大支持24个；

标志位	描述
M8084	为ON使能高速计速器多用户中断
D8084	为高速计数器序号C235~C255
D8085	对应的用户中断个数，最大24个，从I507~I530
D8086	对应多个比较点数据的序号，只能为D元件，如200为D200开始的双字

例如：当行车在轨道上执行装卸任务时，要求运行在不同位置时快速执行不同任务，使用高速计数器多用户中断处理功能，具有响应快操作简单的优势



具体功能设置请看H2U特殊功能的介绍

计米器、测速表功能 M8100~M8105

在SPD指令中，如果功能使能标志M8100~M8105(分别对应X0~X5)为ON，则SPDS1 S2 D写法中的D的定义和原指令有改变，

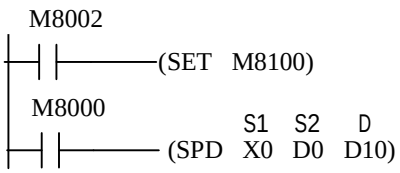
对应的M8100~M8105为OFF时：

D+0：为在S2时间内的脉冲个数，为16位的数据；D+l：本次时间段对脉冲的计数；D+2：用于测定剩余时间(MS)。

对应的M8100~M8105为ON时：

D+0：为在S2时间内的脉冲个数，为16位的数据；D+1，D+2：为每分钟内的脉冲个数，为32位的数据；

例如指令



X0为输入端口；
D0为单位时间(mS)；
D10 为D0 时间段总脉冲个数；
D11,D12 为运行频率=1 分钟脉冲个数*10(0.1为单位)；

具体功能设置请看H2U特殊功能的介绍

输出端口清零 M8140 D8156~D8160

当M8140为ON的时候，用ZRN指令回到原点后D8156~D8160对应的输出点有一个扫描周期的ON输出，用来CLR伺服。

D8156：Y0端口清零信号定义，默认5=Y05

D8157：Y1端口清零信号定义，默认6=Y06

D8158：Y2端口清零信号定义，默认7=Y07

D8159：Y3端口清零信号定义，默认8=Y10

D8160：Y4端口清零信号定义，默认9=Y11

高速环形计数器

高速环形计数器 M8099 D8099

特殊数据寄存器D8099在M8099被驱动后，从下一个扫描周期开始，以0.1ms时钟累计

计算。当D8099内容超过32767时，从0开始重新计算。

利用累积型的 1ms 定时器或特殊数据寄存器 D8099（高速环形计数器），可以测定 1ms 或 0.1ms 为单位的脉冲的宽度

通讯、链接

COM0: D8110~D8119

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：M8121~M8124 M8129 D8110~D8119

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）

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	为ON时485BD扩展卡有效	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）

COM0与COM1的功能区别

COM0硬件为标准RS485和RS422，两者兼容，通过跳线决定。接口端子为8孔鼠标头母座。

COM1硬件为RS485，接口为接线端子。

COM0的485只能做从站，不能做主站，不支持RS指令、联机功能、计算机链接协议等；

COM1的485可做主站或者从站，支持RS指令、联机功能、计算机链接协议等。

定位

定位 M8145~M8154 D8136~D8160

M8145：在发脉冲的时候将M8145置ON。则Y0马上停止脉冲输出，PLSY、DRVI等指令在下一次驱动能流的时候重新从零发脉冲。DRVA指令在下一次驱动能流的时候接着发剩余的脉冲。

M8146：在发脉冲的时候将M8146置ON。则Y1马上停止脉冲输出，PLSY、DRVI等指令在下一次驱动能流的时候重新从零发脉冲。DRVA指令在下一次驱动能流的时候接着发剩余的脉冲。

M8146：在发脉冲的时候将M8146置ON。则Y1马上停止脉冲输出，PLSY、DRVI等指令在下一次驱动能流的时候重新从零发脉冲。DRVA指令在下一次驱动能流的时候接着发剩余的脉冲。

M8152：在发脉冲的时候将M8152置ON。则Y2马上停止脉冲输出，PLSY、DRVI等指令在下一次驱动能流的时候重新从零发脉冲。DRVA指令在下一次驱动能流的时候接着发剩余的脉冲。

M8153：在发脉冲的时候将M8153置ON。则Y3马上停止脉冲输出，PLSY、DRVI等指令在下一次驱动能流的时候重新从零发脉冲。DRVA指令在下一次驱动能流的时候接着发剩余的脉冲。

M8154：在发脉冲的时候将M8154置ON。则Y4马上停止脉冲输出，PLSY、DRVI等指

令在下一次驱动能流的时候重新从零发脉冲。DRVA指令在下一次驱动能流的时候接着发剩余的脉冲。

M8147：在Y0有脉冲输出的时候，M8147为ON，停止时为OFF，可用来做监控；

M8148：在Y1有脉冲输出的时候，M8148为ON，停止时为OFF，可用来做监控；

M8149：在Y2有脉冲输出的时候，M8149为ON，停止时为OFF，可用来做监控；

M8150：在Y3有脉冲输出的时候，M8150为ON，停止时为OFF，可用来做监控；

M8151：在Y4有脉冲输出的时候，M8151为ON，停止时为OFF，可用来做监控。

D8136：Low word；D8137：High word

作为Y0和Y1输出定位指令的当前值数据累加寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8140：Low word；D8141：High word

作为Y0输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8142：Low word；D8143：High word

作为Y1输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8150：Low word；D8151：High word

作为Y2输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8152：Low word；D8153：High word

作为Y3输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8154：Low word；D8155：High word

作为Y4输出定位指令的当前值数据寄存器使用，用DRVI、DRVA指令时，对应旋转方向增减当前值，另外，由于PLSY、PLSR只有脉冲输出没有方向信号的指令亦使用相同的当前值寄存器，因此当执行这些指令时，当前值为脉冲输出数的累加值；

D8145：执行 DRVI、DRVA 等指令时的基底速度。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（D8l47，D8l46）的 l/10 以下。超过该范围时，自动降为最高速度的 1/10 数值运行。

D8146：Low word；D8147：High word

执行DRVI、DRVA等指令时的最高速度。高速输出口输出脉冲的频率必须小于该最高速度。设定范围：10~100，000（Hz）

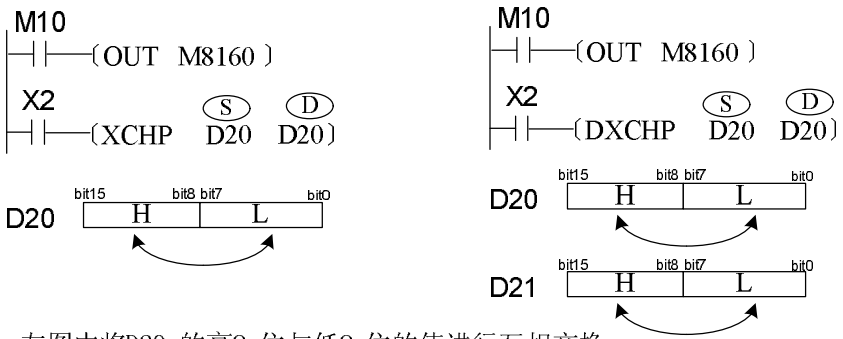
D8l48：DRVI、DRVA、PLSR指令时的加减速时间。加减速时间表示到达最高速度（D8147，D8146）所需的时间。因此，当输出脉冲频率低于最高速度（D8147，D8146）时，实际加减速时间会缩短。设定范围：50~5，000(ms)；

当M8135~M8140

数据处理

XCH的SWAP功能 M8160

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，32 位的指令也一样，完成的操作将是将两个寄存器的高 8 位与低 8 位的值各自进行互相交换。相当于 SWAP 指令的操作，一般用 SWAP 指令来实现.



左图中将D20 的高8 位与低8 位的值进行互相交换
右图中将D20 的高8 位与低8 位的值进行互相交换，
D21 的高8 位与低8 位的值进行互相交换，

8位与16位模式 M8161

M8161标志决定了变量的宽度模式，当M8161=OFF时，为16bit模式；当M8161=ON时，为8bit模式，因此实际使用变量区域的长度增加。

在ASC/RS/ASCII/HEX/CCD中应用。

传送点数可变模式 M8164 D8164

FROM/T0指令的传送点数可变模式：若M8164＝ON，执行FROM/T0指令时，特殊数据寄存器D8164（FROM/T0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；

位切换字功能 M8167

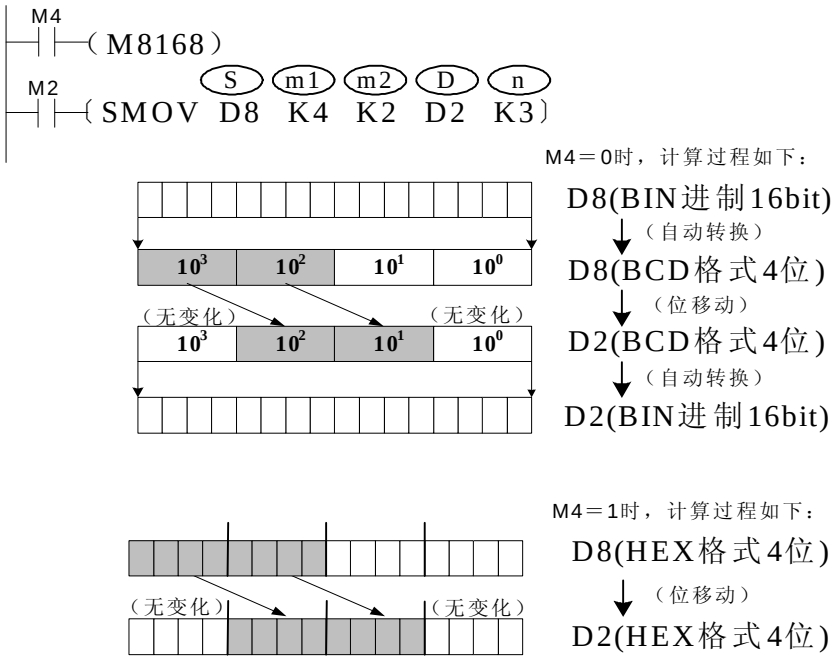
在HEY指令中将M8167置为ON，HEY指令将⓪～ 按键按16进制数据进行存储，保存

到(D2)单元。

例：[123BF]输入后，(D2)中以BIN形式存储[123BF]，即将A~F的功能改变，详细介绍请参考HEY指令。

SMOV十进制与十六进制切换功能 M8168

在SMOV指令中，当M8168为OFF时是BCD模式（十进制的位），当M8168为ON时是BIN模式，在BIN模式下以4个位作为一个单位作传送（十六进制的位）.



假设D8=K1234，D2=K5678，，则当M8168为OFF时（BCD模式），将M2置ON，则D2的值变为K5128；

当M8168为ON时（BIN模式），此时D8=H04D2=K1234，D2=H162E=K5678，将M2置ON，则D2=H104E=K4174

脉冲捕捉

脉冲捕捉功能 M8170~M8175

M8170 X00脉冲捕捉

M8171 X01脉冲捕捉

M8172 X02脉冲捕捉

M8173 X03脉冲捕捉

M8174　X04脉冲捕捉

M8175　X05脉冲捕捉

若需要对出现在X0～X5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。

执行中断允许 EI 指令后，当输入点 X000~X005OFF→ON 变化时，特殊辅助继电器 M8170~M8175 置位进行中断处理。为了再次获得输入，必须利用程序对设定的元件进行复位操作。脉冲捕捉动作同个别中断禁止用辅助继电器 M8050~M8055 的动作无关，M8050~M8055 的置 ON 不能禁止此捕捉功能。

高速输入

表格高速比较模式 M8130 M8131 D8130

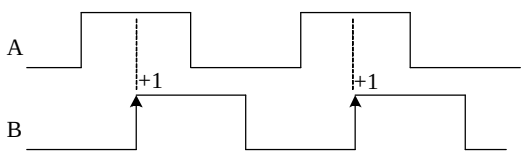
在表格高速比较模式中，第一行的数据一致时，表格计数器 D8130 变为 1，并进入第二行操作。以下同样动作，最后一行操作完毕时，完毕标志 M8l31 动作，回到初始行重复动作。

具体应用请参照：

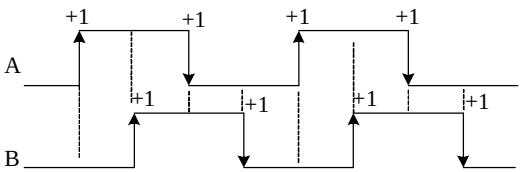
倍频控制　M8195~M8199

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。AB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C251～C255的A/B输入1倍频模式时，最大高速输入频率为50kHz。若C251～C255的A/B输入4倍频模式时，为软件计数模式，最大高速输入频率降为25kHz。

当M8195为OFF的时候，C251的AB相输入为一倍频；当M8195为ON的时候，C251的AB相输入为四倍频，如下图：



M8195=0，C251为1倍频的A/B计数



M8195=1，C251为4倍频的A/B计数

和C251一样：

当M8196为OFF的时候，C252的AB相输入为一倍频；当M8196为ON的时候，C252的AB相输入为四倍频：

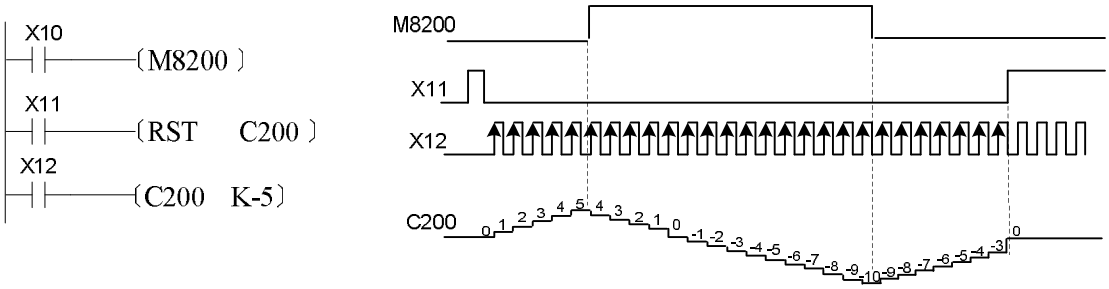
当M8197为OFF的时候，C253的AB相输入为一倍频；当M8197为ON的时候，C253的AB相输入为四倍频：

当M8198为OFF的时候，C254的AB相输入为一倍频；当M8198为ON的时候，C254的AB相输入为四倍频：

当M8199为OFF的时候，C255的AB相输入为一倍频；当M8199为ON的时候，C255的AB相输入为四倍频：

32位计数增减控制　M8200~M8234

对于普通的 32 位增减计数，可利用特殊的辅助继电器 M8200~M8234 指定增计数 / 减计数的方向，如对 C△△△驱动 M8△△△，则为减计数，不驱动时，则为增计数，如下例：



当M8200为OFF时，C200为增计数；当M8200为ON时，C200为减计数

和C200一样：

当M8201为OFF时，C201为增计数；当M8201为ON时，C201为减计数

当M8202为OFF时，C202为增计数；当M8202为ON时，C202为减计数

当M8203为OFF时，C203为增计数；当M8203为ON时，C203为减计数

当M8204为OFF时，C204为增计数；当M8204为ON时，C204为减计数

当M8205为OFF时，C205为增计数；当M8205为ON时，C205为减计数

当M8206为OFF时，C206为增计数；当M8206为ON时，C206为减计数

当M8207为OFF时，C207为增计数；当M8207为ON时，C207为减计数

当M8208为OFF时，C208为增计数；当M8208为ON时，C208为减计数

当M8209为OFF时，C209为增计数；当M8209为ON时，C209为减计数

当M8210为OFF时，C210为增计数；当M8210为ON时，C210为减计数

当M8211为OFF时，C211为增计数；当M8211为ON时，C211为减计数

当M8212为OFF时，C212为增计数；当M8212为ON时，C212为减计数

当M8213为OFF时，C213为增计数；当M8213为ON时，C213为减计数

当M8214为OFF时，C214为增计数；当M8214为ON时，C214为减计数

当M8215为OFF时，C215为增计数；当M8215为ON时，C215为减计数

当M8216为OFF时，C216为增计数；当M8216为ON时，C216为减计数

当M8217为OFF时，C217为增计数；当M8217为ON时，C217为减计数

当M8218为OFF时，C218为增计数；当M8218为ON时，C218为减计数

当M8219为OFF时，C219为增计数；当M8219为ON时，C219为减计数

当M8220为OFF时，C220为增计数；当M8220为ON时，C220为减计数

当M8221为OFF时，C221为增计数；当M8221为ON时，C221为减计数

当M8222为OFF时，C222为增计数；当M8222为ON时，C222为减计数

当M8223为OFF时，C223为增计数；当M8223为ON时，C223为减计数

当M8224为OFF时，C224为增计数；当M8224为ON时，C224为减计数

当M8225为OFF时，C225为增计数；当M8225为ON时，C225为减计数

当M8226为OFF时，C226为增计数；当M8226为ON时，C226为减计数

当M8227为OFF时，C227为增计数；当M8227为ON时，C227为减计数

当M8228为OFF时，C228为增计数；当M8228为ON时，C228为减计数

当M8229为OFF时，C229为增计数；当M8229为ON时，C229为减计数

当M8230为OFF时，C230为增计数；当M8230为ON时，C230为减计数

当M8231为OFF时，C231为增计数；当M8231为ON时，C231为减计数

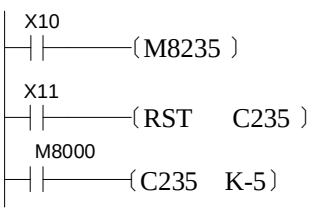
当M8232为OFF时，C232为增计数；当M8232为ON时，C232为减计数

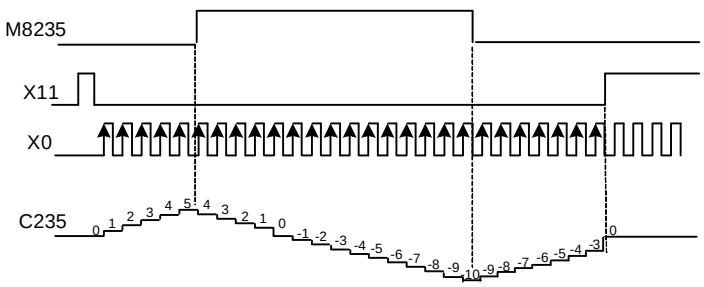
当M8233为OFF时，C233为增计数；当M8233为ON时，C233为减计数

当M8234为OFF时，C234为增计数；当M8234为ON时，C234为减计数

高速单相单计数增减控制　M8235~M8245

单相单计数型高速计数输入，只有1个计数脉冲信号输入端，其计数的增减由程序利用对应的特殊M寄存器决定为增计数或减计数





当M8235为OFF时，C235为增计数；当M8235为ON时，C235为减计数

和C235一样：

当M8236为OFF时，C236为增计数；当M8236为ON时，C236为减计数

当M8237为OFF时，C237为增计数；当M8237为ON时，C237为减计数

当M8238为OFF时，C238为增计数；当M8238为ON时，C238为减计数

当M8239为OFF时，C239为增计数；当M8239为ON时，C239为减计数

当M8240为OFF时，C240为增计数；当M8240为ON时，C240为减计数

当M8241为OFF时，C241为增计数；当M8241为ON时，C241为减计数

当M8242为OFF时，C242为增计数；当M8242为ON时，C242为减计数

当M8243为OFF时，C243为增计数；当M8243为ON时，C243为减计数

当M8244为OFF时，C244为增计数；当M8244为ON时，C244为减计数

当M8245为OFF时，C245为增计数；当M8245为ON时，C245为减计数

C241~C245具有硬件复位输入功能，部分具有硬件起停输入功能

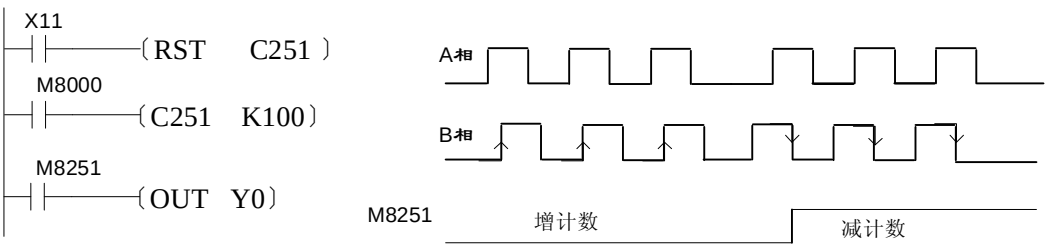
高速计数增减监控 M8246~M8255

单相2输入高速计数和AB相输入高速计数利用外部端口的输入，可自动增计数或减计数。

单相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；

AB相计数型，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M8246-M8255的状态，可监控C246-C255的增计数／减计数状态。如下图C251的监控：



当C251为增计数时，M8251为OFF；当C251为减计数时，M8251为ON。通过监控M8251可以知道C251是增计数还是减计数

和C251一样：

当C246为增计数时，M8246为OFF；当C246为减计数时，M8246为ON。通过监控M8246可以知道C246是增计数还是减计数

当C247为增计数时，M8247为OFF；当C247为减计数时，M8247为ON。通过监控M8247可以知道C247是增计数还是减计数

当C248为增计数时，M8248为OFF；当C248为减计数时，M8248为ON。通过监控M8248可以知道C248是增计数还是减计数

当C249为增计数时，M8249为OFF；当C249为减计数时，M8249为ON。通过监控M8249可以知道C249是增计数还是减计数

当C250为增计数时，M8250为OFF；当C250为减计数时，M8250为ON。通过监控M8250可以知道C250是增计数还是减计数

当C252为增计数时，M8252为OFF；当C252为减计数时，M8252为ON。通过监控M8252可以知道C252是增计数还是减计数

当C253为增计数时，M8253为OFF；当C253为减计数时，M8253为ON。通过监控M8253

可以知道C253是增计数还是减计数

当C254为增计数时，M8254为OFF；当C254为减计数时，M8254为ON。通过监控M8254可以知道C254是增计数还是减计数

当C255为增计数时，M8255为OFF；当C255为减计数时，M8255为ON。通过监控M8255可以知道C255是增计数还是减计数

Inovance

H_{2U} Series PLC

programming manual

深圳市汇川技术股份有限公司
Shenzhen Inovance Technology Co.,Ltd.
地址：深圳市宝安区宝城70区留仙二路鸿威工业区E栋
全国统一服务电话：400-777-1260
传真：(0755)2961 9897
http://www.inovance.cn