



## 指令使用手册（TC 系列主机）

文档发布版本: v2.0.0

对应主机版本:

文档发布时间: 2022-11-23

- 1, 修改旋转轴的周期范围, 将正反周期单元范围修改为 全部是正数周期单元范围;
- 2, 修改 MC\_Home 的运动标志位 SM 的设置说明;
- 3, 修改 SM108.1~SM111.7 为各节点伺服停止方式选择;
- 4, 删除 SM116.5 伺服停止方式选择;
- 5, 删除 SM116.6 直线插补轴选择;
- 6, 增加指令 6.2.4, MUL\_W (整形乘法\_字), 后续排号顺移;
- 7, 增加指令 6.4.1, WToD (字转双字) 指令;
- 8, 增加指令 6.4.2, DToW (双字转字) 指令;
- 9, 把指令 INT (浮点数->整数变换) 移动到 6.4.3;
- 10, 把指令 FLT (整数->浮点数变换) 移动到 6.4.4, 后续排号顺移;
- 11, 修改 MC\_MoveRela 和 MC\_MoveAbso 指令;
- 12, 修改 MC\_ChangeVelo 指令的“Freq”为“Velo”;
- 13, 修改 MC\_MoveVelo 指令的“Freq”为“Velo”, “AcDe\_F”为“AcDe\_V”;
- 14, 增加指令 10.3.7, MC\_MoveRota (旋转轴绝对位移), 后续排号顺移;
- 15, 删除 SM100.0~SM103.7 的电子齿轮比设置标志位, 及相关说明;
- 16, 删除 SM104.0~SM107.7 的旋转轴设置标志位, 及相关说明;
- 17, 删除 SD662~SD721 伺服电机每转位移的机械单元数, 及相关说明;
- 18, 删除 SD722~SD781 伺服电机端周期圈数 (传动比分子) / (传动比分母), 及相关说明;
- 19, 变更直线插补指令为指定以 X 轴的转速;

一	各装置说明	9
1.1	常数 (O)	9
1.2	X/Y 装置	11
1.3	M 装置	12
1.4	SM 装置	12
1.5	T 装置	12
1.6	C 装置	12
1.7	HC 装置	13
1.8	AI/AQ 装置	13
1.9	D 装置	14
1.10	SD 装置	14
1.11	E 装置	14
1.12	ND 装置	14
1.13	KnX, KnY, KnM...Kn 位装置	15
1.15	EL 装置	15
1.16	PLC 装置 Modbus 通讯地址	16
二	SM、SD 使用说明	17
2.1	SM 定义列表	17
2.1.1	通用 SM	17
2.1.2	CC-CAN 总线 SM (使用说明见 10.1 章节)	23
2.2	SD 定义列表	27
2.2.1	通用 SD	27
2.2.2	CC-CAN 总线 SD (使用说明见 10.1 章节)	29
2.3	SM 和 SD 群组功能说明	33
2.3.1	PLC 系统类别	33
2.3.2	RS485 通讯口类别	37
2.3.3	输入口类别	39
2.3.4	输出口类别	39
三	接点开关型指令	40
3.1	基础接点开关	40
3.1.1	LD (常开开关)	40
3.1.2	LDI (常闭开关)	40
3.1.3	LDP (上升沿触发开关)	40
3.1.4	LDF (下降沿触发开关)	41
3.1.5	AND (串联常开接点)	41
3.1.6	ANI (串联常闭接点)	41
3.1.7	ANDP (串联上升沿接点)	41
3.1.8	ANDF (串联下降沿接点)	42
3.1.9	OR (并联常开接点)	42
3.1.10	ORI (并联常闭接点)	42
3.1.11	ORP (并联上升沿接点)	42
3.1.12	ORF (并联下降沿接点)	43
3.2	逻辑接点开关	44
3.2.1	LD&、LD 、LD^ (开始接点, 逻辑运算)	44



3.2.2	AND&、AND 、AND^ (串联接点, 逻辑运算)	45
3.2.3	OR&、OR 、OR^ (并联接点, 逻辑运算)	46
3.2.4	LD=、LD<、LD>、LD<>、LD<=、LD>= (开始接点, 数值比较运算)	47
3.2.5	AND=、AND<、AND>、AND<>、AND<=、AND>= (串联接点, 数值比较运算)	48
3.2.6	OR=、OR<、OR>、OR<>、OR<=、OR>= (并联接点, 数值比较运算)	49
3.3	字装置的位接点开关	50
3.3.1	BLD (常开开始接点, 取字/双字装置位)	50
3.3.2	BLDI (常闭开始接点, 取字/双字装置位)	51
3.3.3	BAND (常开串联接点, 取字/双字装置位)	52
3.3.4	BANI (常闭串联接点, 取字/双字装置位)	53
3.3.5	BOR (常开并联接点, 取字/双字装置位)	54
3.3.6	BORI (常闭并联接点, 取字/双字装置位)	55
3.4	浮点数比较接点开关	56
3.4.1	FLD=、FLD<、FLD>、FLD<>、FLD>=、FLD<= (开始接点, 浮点数比较运算)	56
3.4.2	FAND=、FAND<、FAND>、FAND<>、FAND>=、FAND<= (串联接点, 浮点数比较运算)	57
3.4.3	FOR=、FOR<、FOR>、FOR<>、FOR>=、FOR< (并联接点, 浮点数比较运算)	58
3.5	多接点开关连接指令	59
3.5.1	ANB (块串联)	59
3.5.2	ORB (块并联)	59
3.5.3	MPS (压入堆栈)	59
3.5.4	MRD (读取堆栈)	59
3.5.5	MPP (弹出堆栈)	59
3.5.6	INV (反转状态)	60
3.5.7	NOP (空指令)	60
四	位输出型指令	61
4.1.1	OUT (输出线圈)	61
4.1.2	SET (对位置位)	61
4.1.3	PLS (上升沿检测置位)	61
4.1.4	PLF (下降沿检测置位)	62
4.1.5	BOUT (字符装置位输出)	63
4.1.6	BSET (字符装置位置置位)	64
4.1.7	BRST (字符装置位清0)	65
4.1.8	ALT (On/Off 交替输出)	66
五	基础系统型指令	67
5.1.1	TMR (启动定时器)	67
5.1.2	CNT (启动普通计数器)	69
5.1.3	RST (装置复位)	75
5.1.4	WDT (清空监视定时器)	76
5.1.5	FOR (循环范围开始)	77
5.1.6	NEXT (循环范围结束)	78
5.1.7	JMP (跳转指令)	79
5.1.8	LBL (跳转目的编号指令)	80
5.1.9	EI (中断使能)	81
5.1.10	DI (中断禁用)	82

5.1.11 REF (I/O 状态实时刷新) .....	83
5.1.12 FBx (功能块调用) .....	84
<b>六 应用型指令 .....</b>	<b>85</b>
<b>6.1 传送比较指令 .....</b>	<b>85</b>
6.1.1 CMP (比较设置输出) .....	85
6.1.2 ZCP (区间比较输出) .....	87
6.1.3 MOV (数据搬移) .....	88
6.1.4 SMOV (移码传送) .....	89
6.1.5 CML (反相传送) .....	91
6.1.6 BMOV (范围装置传送) .....	92
6.1.7 FMOV (多点数据传送) .....	93
6.1.8 XCH (数据交换) .....	94
6.1.9 BCD (BIN->BCD 转换) .....	96
6.1.10 BIN (BCD->BIN 转换) .....	97
6.1.11 MVM (指定位搬移) .....	98
6.1.12 MMOV (放大传送) .....	99
6.1.13 CMPT (表格比较指令) .....	100
<b>6.2 整型四则逻辑运算 .....</b>	<b>102</b>
6.2.1 ADD (整型加法) .....	102
6.2.2 SUB (整型减法) .....	103
6.2.3 MUL (整型乘法) .....	104
6.2.4 MUL_W (整型乘法_字) .....	105
6.2.5 DIV (整型除法) .....	106
6.2.6 MOD (整型求余数) .....	107
6.2.7 INC (自加一) .....	108
6.2.8 DEC (自减一) .....	109
6.2.9 WAND/DAND (逻辑与运算) .....	110
6.2.10 WOR/DOR (逻辑或运算) .....	111
6.2.11 WXOR/DXOR (逻辑异或运算) .....	112
6.2.12 NEG (二进制补码) .....	113
<b>6.3 位移指令 .....</b>	<b>114</b>
6.3.1 ROR (右循环移位) .....	114
6.3.2 ROL (左循环移位) .....	115
6.3.3 RCR (附进位标志右循环移位) .....	116
6.3.4 RCL (附进位标志左循环移位) .....	117
6.3.5 SFTR (位右移) .....	118
6.3.6 SFTL (位左移) .....	119
6.3.7 WSFR (字右移) .....	120
6.3.8 WSFL (字左移) .....	121
6.3.9 SFWR (移位写入) .....	122
6.3.10 SFRD (移位读出) .....	124
<b>6.4 数据处理指令 .....</b>	<b>126</b>
6.4.1 WToD (字转双字) .....	126
6.4.2 DToW (双字转字) .....	127

6.4.3	INT (浮点数->整数变换)	128
6.4.4	FLT (整数->浮点数转换)	129
6.4.5	ZRST (批次复位)	130
6.4.6	DECO (译码器)	131
6.4.7	ENCO (编码)	132
6.4.8	SUM (ON 位数量统计)	133
6.4.9	BON (ON 判定)	134
6.4.10	MEAN (计算平均值)	135
6.4.11	SQR (开平方根)	136
6.5	便利指令	137
6.5.1	SER (数据检索)	137
6.5.2	ABSD (绝对方式凸轮控制)	139
6.5.3	INCD (相对方式凸轮控制)	141
6.5.4	TTMR (示教式定时器)	143
6.5.5	STMR (特殊定时器)	145
6.5.6	RAMP (斜坡信号)	146
6.5.7	SORT (资料整理排序)	148
6.6	外部设定显示	150
6.6.1	TKY (十键键盘输入)	150
6.6.2	HKY (十六键键盘输入)	152
6.6.3	SEGD (七段显示器解码)	154
6.6.4	ASC (ASCII 码转换)	155
6.7	浮点运算	156
6.7.1	DECMP (浮点数比较)	156
6.7.2	DEZCP (浮点数区间比较)	157
6.7.3	DMOV (浮点数数据搬移)	159
6.7.4	DRAD (角度->弧度)	160
6.7.5	DDEG (弧度->角度)	161
6.7.6	DEBCD (浮点数->十进制浮点型态)	162
6.7.7	DEBIN (十进制浮点型态->浮点数)	163
6.7.8	DEADD (浮点数加法)	164
6.7.9	DESUB (浮点数减法)	165
6.7.10	DEMUL (浮点数乘法)	166
6.7.11	DEDIV (浮点数除法)	167
6.7.12	DEXP (浮点数取指数)	168
6.7.13	DLN (浮点数取自然对数)	169
6.7.14	DLOG (浮点数取对数)	170
6.7.15	DESQR (浮点数开平方根)	171
6.7.16	DPOW (浮点数权值指令-幂次方)	172
6.7.17	ESCLP (浮点数参数型比例值运算)	173
6.8	三角函数运算	175
6.8.1	DSIN (二进制浮点数 SIN 运算)	175
6.8.2	DCOS (二进制浮点数 COS 运算)	176
6.8.3	DTAN (二进制浮点数 TAN 运算)	177

6.8.4	DASIN (二进制浮点数 ASIN 运算)	178
6.8.5	DACOS (二进制浮点数 ACOS 运算)	179
6.8.6	DATAN (二进制浮点数 ATAN 运算)	180
6.8.7	DSINH (二进制浮点数 SINH 运算)	181
6.8.8	DCOSH (二进制浮点数 COSH 运算)	182
6.8.9	DTANH (二进制浮点数 TANH 运算)	183
6.9	万年历	184
6.9.1	TCMP (万年历数据比较)	184
6.9.2	TZCP (万年历数据区间比较)	186
6.9.3	TADD (万年历数据加算)	188
6.9.4	TSUB (万年历数据减算)	189
6.9.5	TWR (万年历时间写入更新)	190
6.9.6	HOUR (计时仪)	191
6.10	格雷码	192
6.10.1	GRY (BIN->GRAY 码转换)	192
6.10.2	GBIN (GRAY 码->BIN 转换)	193
6.11	矩阵处理	194
6.11.1	MAND (矩阵与运算)	194
6.11.2	MOR (矩阵或运算)	196
6.11.3	MXOR (矩阵异或运算)	198
6.11.4	MXNR (矩阵同或运算)	200
6.11.5	MINV (矩阵求反运算)	202
6.11.6	MCMP (矩阵比较)	203
6.11.7	MBRD (矩阵位读出)	205
6.11.8	MBWR (矩阵位写入)	206
6.11.9	MBS (矩阵位移位)	208
6.11.10	MBR (矩阵位循环移位)	211
6.11.11	MBC (矩阵位状态计数)	213
6.12	特殊功能指令	214
6.12.1	DELAY (延迟指令)	214
6.12.2	GPWM (一般用脉冲波宽调变输出)	215
6.12.3	SWAP (上下字节/字互换)	216
6.12.4	PWD (输入脉宽检测)	217
6.12.5	RTMU (程序执行时间测量开始)	218
6.12.6	RTMD (程序执行时间测量结束)	219
6.12.7	RAND (随机数值产生)	220
6.12.8	SCAL (比例值运算)	221
6.12.9	SCLP (参数型比例值运算)	223
6.12.10	WSUM (求和)	225
6.12.11	MTR (I/O 矩阵输入)	226
6.12.12	HST (高速定时器)	227
6.12.13	PID (PID 运算)	228
6.12.14	FAN (风扇控制)	232
七	高速输入计数指令	233



7.1.1	HCNT (启动高速计数器)	233
7.1.2	HSCS (比较置位-高速计数器)	236
7.1.3	HSCR (比较复位-高速计数器)	238
7.1.4	HSZ (区间比较-高速计数器)	240
7.1.5	SPD (定时脉冲个数检测)	242
八	高速脉冲输出指令	243
九	串行通讯指令	244
9.1	自由通讯协议指令	244
9.1.1	RS (串行数据传送)	244
9.1.2	ASCI (HEX->ASCII 码)	249
9.1.3	HEX (ASCII->HEX 码)	250
9.1.4	CCD (校验码)	251
9.1.5	ABS (取绝对值运算)	254
9.1.6	LRC (LRC 校验码计算)	255
9.2	MODBUS 通讯协议指令	259
9.2.1	MODRD (MODBUS 通讯寄存器数据读取)	259
9.2.2	MODWR (MODBUS 通讯单个寄存器数据写入)	261
9.2.3	MODRW (MODBUS 通讯数据读写)	263
9.3	JSCC 变频器控制指令	266
9.3.1	FWD (变频器正转指令)	266
9.3.2	REV (变频器反转指令)	268
9.3.3	STOP (变频器停止运转指令)	270
9.3.4	FREQ (变频器设置频率指令)	272
9.3.5	RERR (变频器读取故障指令)	274
十	CC-CAN 总线指令	276
10.1	使用总线及节点状态 SM 标志和 SD 寄存器	276
10.2	通用控制指令	278
10.2.1	CC_ReadParam (读取单个参数)	278
10.2.2	CC_WriteParam (写入单个菜单参数)	280
10.2.3	CC_WriteCtl (写入单个控制寄存器)	282
10.3	伺服及手轮运动控制指令	284
10.3.1	MC_ServoOn (伺服使能)	284
10.3.2	MC_Home (原点回归)	286
10.3.3	MC_Stop (停止指令)	288
10.3.4	MC_MoveRela (相对位移)	290
10.3.5	MC_MoveAbso (绝对位移)	292
10.3.6	MC_ChangeVelo (位移速度变更)	295
10.3.7	MC_MoveRota (旋转轴绝对位移)	297
10.3.8	MC_HDWTtoSVO (伺服跟随电子手轮位移)	300
10.3.9	MC_SVOFollow (伺服跟随变量装置位移)	302
10.3.10	MC_SVOFollowD (伺服跟随变量装置位移)	302
10.3.11	MC_MoveVelo (位置模式下速度控制)	304
10.3.12	MC_LinRela (双轴相对直线插补)	307
10.3.13	MC_LinAbso (双轴绝对直线插补)	309

10.3.14	MC_CirRela (双轴相对圆弧插补)	312
10.3.15	MC_CirAbso (双轴绝对圆弧插补)	314
10.4	伺服及手轮通用控制指令	317
10.4.1	SVO_SetHome (伺服原点位置设定)	317
10.4.2	SVO_LimitFWD (伺服正转极限位置设定)	319
10.4.3	SVO_LimitREV (伺服反转极限位置设定)	321
10.4.4	SVO_Mode (伺服运转模式设置)	323
10.4.5	SVO_RunCmd (伺服运转命令)	325
10.4.6	SVO_Reset (伺服故障复位)	327
10.4.7	SVO_Velocity (伺服速度模式转速值设定)	329
10.4.8	SVO_VeloLock (伺服速度模式下锁轴设定)	331
10.4.9	SVO_VelTorLim (伺服速度模式下转矩限制值设定)	333
10.4.10	SVO_Torque (伺服转矩模式下转矩值设定)	335
10.4.11	SVO_TorqLock (伺服转矩模式下锁轴设定)	337
10.4.12	SVO_VeloLimit (伺服转矩模式下转速限制值设定)	339
10.4.13	SVO_GainSel (伺服位置模式下增益组选择)	341
10.4.14	SVO_BreakSet (伺服位置模式下抱闸选择设定)	343
10.4.15	SVO_PosTorLim (伺服位置模式下转矩限制值设定)	345
10.4.16	HDW_ReadMesg (电子手轮位移信息读取)	347
10.5	变频器控制指令	349
10.5.1	VFD_RunCmd (变频器运转命令)	349
10.5.2	VFD_Freq (变频器运转频率设定)	351
10.5.3	VFD_Reset (变频器故障复位)	353
10.6	调速器控制指令	355
10.6.1	SPD_RunCmd (调速器运转命令)	355
10.6.2	SPD_Speed (调速器运转转速设定)	357
10.6.3	SPD_Torque (调速器运转力矩设定)	359
10.6.4	SPD_BreakSet (调速器失电电磁制动器控制)	361
10.6.5	SPD_Reset (调速器故障复位)	363



## 一 各装置说明

### 1.1 常数 (O)

常数	表示符号	16 位范围	32 位范围
二进制	2#	2#16bit	2#32bit
十进制	直接表示	-32768~32767	-2147483648~2147483647
十六进制	16#	16#0000~16#FFFF	16#00000000~16#FFFFFFFF

当数值相加超过对应的位数范围值时,例如 16 位范围的十进制数值 32767,再进行加 1 的运算后,得到的结果将是-32768,十六进制数值 16#FFFF,再加 1 运算后,得到数值 16#0000。

#### (1) 二进制

PLC 内部数值的存储或运算都以二进制的形式进行,二进制数值及相关术语如下:

位 (bit) : 位为二进制的最基本单位,其状态非 0 即 1;

半字节 (nibble) : 是由连续的 4 个位组成 (如 b3~b0), 可用来表示一个位数的十进制数字 0~9 或十六进制数 0~F;

字节 (byte) : 由连续两个半字节组成 (即 8 位, 如 b7~b0), 可表示有符号十进制数-128~127 或十六进制的两个数值 00~FF;

字 (word) : 由连续的两个字节组成 (即 16 位, 如 b15~b0), 可表示有符号十进制数-32768~32767 或十六进制的 4 个数值 0000~FFFF;

双字 (double word) : 由连续的两个字组成 (即 32 位, 如 b31~b0), 可表示有符号十进制数-2147483648~2147483647 或十六进制的 8 个数值 00000000~FFFFFFFF;

#### (2) 十进制

十进制在 PLC 系统应用的时机如下:

- 1, 作为定时器 T, 计数器 C 等的设置值, 例: TMR 50 T0;
- 2, S, M, T, C, D, E 等装置的编号, 例: T10, D10, M5.1;
- 3, 在应用指令中作为操作数使用, 例: MOV 123 D0。

#### (3) BCD

以半个字节或 4 个位来表示一个十进制的数值, 故连续的 16 个位可以表示 4 位数的十进制数值数据。主要用于读取指拨轮数字开关的输入数值或将数值数据输出至七段显示驱动器显示之用。

#### (4) 十六进制

十六进制在 PLC 系统应用的时机如下：在应用指令中作为操作数使用，例：MOV 16#1A2B D0。

各进制对应数值表如下：

二进制								十进制	BCD								十六进制
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3	0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4	0	0	0	0	0	1	0	0	4
0	0	0	0	0	1	0	1	5	0	0	0	0	0	1	0	1	5
0	0	0	0	0	1	1	0	6	0	0	0	0	0	1	1	0	6
0	0	0	0	0	1	1	1	7	0	0	0	0	0	1	1	1	7
0	0	0	0	1	0	0	0	8	0	0	0	0	1	0	0	0	8
0	0	0	0	1	0	0	1	9	0	0	0	0	1	0	0	1	9
0	0	0	0	1	0	1	0	10	0	0	0	1	0	0	0	0	A
0	0	0	0	1	0	1	1	11	0	0	0	1	0	0	0	1	B
0	0	0	0	1	1	0	0	12	0	0	0	1	0	0	1	0	C
0	0	0	0	1	1	0	1	13	0	0	0	1	0	0	1	1	D
0	0	0	0	1	1	1	0	14	0	0	0	1	0	1	0	0	E
0	0	0	0	1	1	1	1	15	0	0	0	1	0	1	0	1	F
0	0	0	1	0	0	0	0	16	0	0	0	1	0	1	1	0	10
0	0	0	1	0	0	0	1	17	0	0	0	1	0	1	1	1	11
.....								.....	.....								.....
0	1	1	0	0	0	1	1	99	1	0	0	1	1	0	0	1	63



## 1.2 X/Y 装置

- 输入/输出接点的编号，小数点后以 8 进制数值编号，小数点前以 10 进制数值编号：
- 对主机而言，输入及输出端的编号固定从 X0.0 及 Y0.0 开始算，编号的多少跟随主机的点数大小而变化。而对 I/O 扩展来说输入及输出端的编号是随着与主机的连接顺序来推算出。

### (1) 输入继电器：X0.0~X31.7

输入继电器最大点数可达 256 点，范围如下：X0.0~X0.7, X1.1~X1.7, ……，X31.0~X31.7；其中 X0, X1, ……，X31 可各自由 8 个输入点，组成一个位组长度的输入组。

### (2) 输出继电器：Y0.0~Y31.7

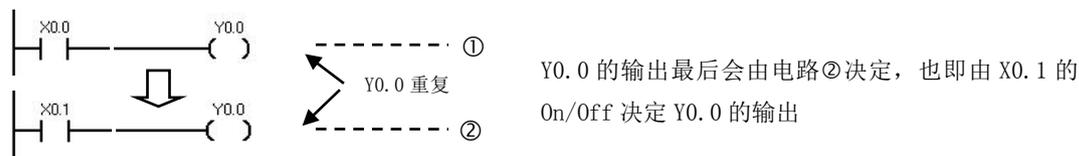
输出继电器最大点数可达 256 点，范围如下：Y0.0~Y0.7, Y1.1~Y1.7, ……，Y31.0~Y31.7；其中 Y0, Y1, ……，Y31 可各自由 8 个输出点，组成一个位组长度的输出组。

### (3) 输入接点 X 的功能：

输入接点 X 与输入装置连接，读取输入信号进入 PLC。每个输入接点 X 的常开或常闭接点于程序中没有使用次数限制。输入接点 X 的 On/Off 只会跟随输入装置的 On/Off 做变化，不可使用通讯使用写线圈方式进行变更输入接点 X 的 On/Off。

### (4) 输出接点 Y 的功能：

输出接点 Y 的任务就是送出 On/Off 信号来驱动连接输出接点 Y 的负载。输出接点分成两种，一为继电器 (Relay)，另一为晶体管 (Transistor)，每一个输出接点 Y 的常开或常闭接点于程序中没有使用次数限制，但输出线圈 Y 的编号，在程序建议仅能使用一次，否则依 PLC 的程序扫描原理，其输出状态的决策权会落在程序中最后的输出 Y 的电路。



### (5) 输入信号再生：

- 1, PLC 在执行程序之前会将外部输入信号的 On/Off 状态一次读入至输入映象区内。
- 2, 在程序执行中若输入信号做 On/Off 变化，但是输入映象区内的状态不会改变，一直到下一次扫描开始时再读入输入信号新的 On/Off 状态。
- 3, 外部信号 On->Off 或 Off->On 变化到程序接点认定为 On/Off 时期间约有 20ms 的延迟（其中延迟时间可以自由设定，同时另外也可能会受程序扫描周期的影响）。

### (6) 程序处理：

PLC 读取输入映象区内各输入信号的 On/Off 状态后开始从地址 0 处依序执行程序中的每一指令，其处理结果即各输出线圈的 On/Off 也逐次存入各装置映象区内。

### (7) 输出再生：

- 1, 当执行到 END 指令，将装置映象区内 Y 的 On/Off 状态送到映象区锁存，而此映象区就是实际上输出继电器的线圈。
- 2, 继电器线圈 On->Off 或 Off->On 变化到接点 On/Off 时期间约有 10ms 的延迟。
- 3, 使用晶体管模块，On->Off 或 Off->On 变化到接点 On/Off 时期间约有 10~20us 的延迟。

### 1.3 M 装置

- 辅助/中间继电器 M 小数点前以 10 进制数编码，小数点后以 8 进制编码：范围 M0.0~M2047.7；
- 辅助继电器 M 与输出继电器 Y 一样有输出线圈及常开常闭接点，而且于程序中无使用次数限制，使用者可利用辅助继电器 M 来组合控制回路，但无法直接驱动外部负载。依其性质可分为两种，其中这两种状态可由用户进行范围设置进行切换：
  - 1, 停电清零辅助继电器：PLC 于运行时若遇到停电，其状态全部被复位为 Off，再送电时其状态仍为 Off。
  - 2, 停电保持辅助继电器：PLC 运行时若遇到停电，其状态将全部被保持，再送电时其状态为停电前状态。

### 1.4 SM 装置

- 特殊用辅助/中间继电器 SM，范围 SM0.0~SM127.7；
- 每一个特殊辅助继电器均有其特定的功用，未定义的特殊用辅助继电器请勿使用。具体详细内容及功能见特殊继电器群组功能说明。

### 1.5 T 装置

定时器分为定时器输出线圈和定时器计时数值，两者的编号，以十进制编号：范围 T0~T255

定时器 T	100ms 一般用	T0~T199, 200 点
	100ms 累计型	T250~T255, 6 点
	10ms 一般用	T200~T239, 40 点
	10ms 累计型	T244~T249, 6 点
	1ms 一般用	T240~T241, 2 点
	1ms 累计型	T242~T243, 2 点

定时器的具体功能及使用方式，参看 TMR 指令及 STMR 指令。

### 1.6 C 装置

计数器分为计数器输出线圈和计数器计数数值，两者的编号，以十进制编号：范围 C0~C255

#### (1) 16 位计数器 C0~C199

- 1, PLC 停电时，若为停电保持型计数器会将停电前的现在值及计数器接点状态存储着，重新上电后会继续累计，否则计数器现在值即被清除；
- 2, 执行 CNT 指令时，只有当 C 当前数值到达设定数值瞬间，C 计数器接点才变成 On (上数) 或 Off (下数)；
- 3, 计数器的设置值可使用常数或任何的 X 装置进行间接设置；
- 4, 其中 C180~C199 可使用对应 SM 进行上/下计数方式的设定；

#### (2) 32 位计数器 C200~C255

- 1, PLC 停电时，若为停电保持型计数器会将停电前的现在值及计数器接点状态存储着，重新上电后会继续累计，否则计数器现在值即被清除；
- 2, 执行 DCNT 指令时，只有当 C 当前数值到达设定数值瞬间，C 计数器接点才变成 On (上数) 或 Off (下数)；
- 3, 计数器的设置值可使用常数或任何的连续的两个 X 装置进行间接设置；
- 4, 其中 C242~C255 可使用对应 SM 进行上/下计数方式的设定。

## 1.7 HC 装置

HC0~HC3 的对应通道由用户自由设定, HC4~HC15 预留。

输入端口		X0.0	X0.1	X0.2	X0.3
功 能 设 置	不可中断单点高速计数	HC0	HC1	HC2	HC3
	可中断单点高速计数	HC0	--	--	--
	AB 相高速计数 1 倍频	HC0		--	--
	AB 相高速计数 2 倍频	HC0		--	--
	AB 相高速计数 4 倍频	HC0		--	--

## 1.8 AI/AQ 装置

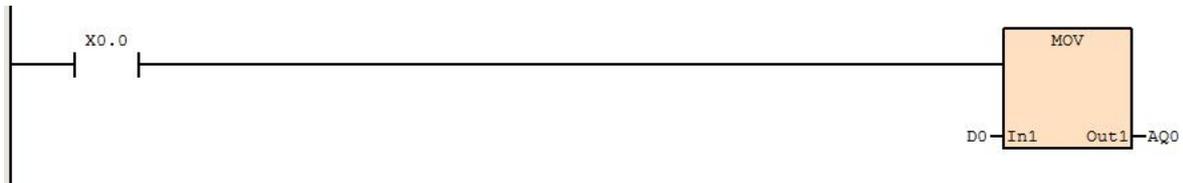
模拟量输入和输出装置以十进制编号, 范围 AI0~AI63, AQ0~AQ63; 排序编号由扩展模块的排接顺序决定。装置长度均为一个字, 16 位。其余性质均与 X/Y 装置相同。

### 例 1:

选用能连接扩展模块的 PLC 主机并连接 EA4AD4DA 扩展模块, 通过软件上的硬件配置, 模拟量配置, 选用通道 AQ0, 设置工作模式为输出电压 0~10V, 对应数值范围 0~4000, 设置完毕后需下载才能生效。

通道	通道类型	工作模式	对应数值范围(上限值: 1~16000)
AQ0	DAQ0	电压0~10V	0 ~ 4000
AQ1	DAQ1	电压0~10V	0 ~ 4000
AQ2	DAQ2	电压0~10V	0 ~ 4000
AQ3	DAQ3	电压0~10V	0 ~ 4000

设置好后, 将 X0.0 置 On, D0 的值被送入 AQ0, 若 D0=2000, 则 AQ0=2000, 对应输出电压为+5V, 即 PLC 中端子 DA0 和 COM0 之间的输出电压为+5V。



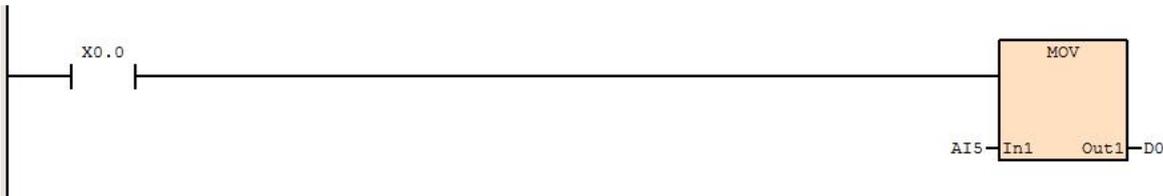
**例 2:**

选用能连接扩展模块的 PLC 主机并连接 EA4TC8YT 或 EA4PT8YT 扩展模块，通过软件上的硬件配置，模拟量配置，选用编号 AI5，设置完毕后需下载才能生效。

模块的热电偶类型: K热电偶

编号	分辨率	偏移修正 (-128~127)	平均采样次数 (1~50)
AI5	<span style="border: 1px solid black; padding: 2px;">每0.5度/1数1</span>	<input type="text" value="0"/>	<input type="text" value="10"/>
AI6	<span style="border: 1px solid black; padding: 2px;">每0.5度/1数1</span>	<input type="text" value="0"/>	<input type="text" value="10"/>
AI7	<span style="border: 1px solid black; padding: 2px;">每0.5度/1数1</span>	<input type="text" value="0"/>	<input type="text" value="10"/>
AI8	<span style="border: 1px solid black; padding: 2px;">每0.5度/1数1</span>	<input type="text" value="0"/>	<input type="text" value="10"/>

设置好后，若测得的环境温度为 25 摄氏度，则 AI5 会显示 50，将 X0.0 置 On，AI5=50 被送入 D0。

**1.9 D 装置**

- 用于储存数值数据，其数据长度为 16 位，最高位为正负号，可储存-32768~32767 的数值数据，也可将两个 16 位寄存器合并成一个 32 位寄存器（D+1，D 编号小的为下 16 位）使用，而其最高位为正负号，可储存-2147483648~2147483647 的数值数据。
- 编号采用十进制，范围 D0~D9999。
- 当寄存器设置为停电保持用寄存器，当 PLC 断电时此区域的寄存器数据不会被清除，仍保持其断电前的数值，否则，断电时会被清除为 0。要清除寄存器内容值，可使用 RST 或 ZRST 或通讯及 MOV 指令等。

**1.10 SD 装置**

- 特殊用寄存器，范围 SD0~SD999；
- 每个特殊用途寄存器均有其特殊定义及用途，主要作为存放系统状态、错误信息、监视状态之用，未定义的特殊用寄存器请勿使用。具体详细内容及功能见特殊寄存器群组功能说明。

**1.11 E 装置**

变址寄存器为 16 位寄存器，十进制编号，范围 E0~E31。可自由的被写入及读出。变址寄存器与一般的操作数相同可用来作为搬移或比较，可用于位装置及字装置，不支持常量。

当变址修饰字装置超出合法范围，若作为输入数据时，数值为 0，若作为输出数据，将不进行输出。

**1.12 ND 装置**

特殊用寄存器，范围 ND0~ND309；

作为精研 PLC 自定义网络专用数据寄存器使用。



### 1.13 KnX, KnY, KnM...Kn 位装置

- Kn 位装置字节编号，可将各位装置组合成字节，字，三字节，双字四种长度的数据进行操作。
- 例如 K2M0，即是将 M0 和 M1 共 16 个点，组合成一个字装置进行数据处理，K3X0，即是将 X0.0~X2.7 共 24 点组成 3 个字节进行数据处理。K1 及 K2 可用于字指令长度的使用，K1，K2，K3 及 K4 均可用于双字指令长度的使用。
- 例如 K1X0 用于字指令时，该数据内容为，高字节数据为 0，低字节数据为 X0.0~X0.7 共 8 点组成的数据；用于双字指令时，高的三个字节数据均为 0，低字节数据为 X0.0~X0.7 共 8 点组成的数据。

### 1.14 L 装置

L 装置为各程序中可建立自由名称和自由数据类型的装置，在各单个程序中，可最多建立 255 个 L 装置（所有数据类型之和），在 PLC 运行过程中，最多可同时使用 500 个 L 装置，超出则会引起系统错误。一共可建立三种数据类型的 L 装置：

- 1, bool 型，即位装置；
- 2, word 型，即单个字装置；
- 3, dword 型，即两个连续的字装置。

建立各 L 装置后，将会按三种数据类型的顺序，1->2->3，按顺序编排各个 L 装置的编号。一旦程序中使用连续装置的指令时，将会在各类型中按装置连续，超出该类型时，输入资料以 0 处理，输出不执行。

### 1.15 EL 装置

EL 装置为功能块中可建立自由名称和自由数据类型以及各自由输入输出类型的装置，只可以在各功能块中建立，可最多建立 8 个 EL 装置（所有数据类型之和）。

EL 可设定为三种数据类型的：

- 1, bool 型，即位装置；
- 2, word 型，即单个字装置；
- 3, dword 型，即两个连续的字装置。

并可定义为三种输入输出类型：

- 1, In 类型，即输入类型装置；
- 2, InOut 类型，即输入输出类型装置；
- 3, Out 类型，即输出类型装置。

建立各 EL 装置后，将会按三种输入输出类型的顺序，1->2->3，按顺序编排各个 EL 装置的编号。程序中使用连续装置的指令时，无法指定 EL 装置，否则系统运算错误。

## 1.16 PLC 装置 Modbus 通讯地址

装置	范围	类别	备注说明	线圈通讯地址 (DEC)	寄存器通讯地址 (DEC)
M	M0.0~M2047.7	Bit	中间继电器区	0~16383	
SM	SM0.0~SM127.7	Bit	用户系统或程序状态标示区	16384~17407	
Y	Y0.0~Y31.7	Bit	外部输出点映射区	17920~18175	
X	X0.0~X31.7	Bit	外部输入点映射区	18176~18431	
T	T0~T255	Bit	定时器触点区	18432~18687	
C	C0~C255	Bit	通用计数器触点区	18688~18943	
HC	HC0~HC15	Bit	高速计数器触点区	18944~18959	
AI	AI0~AI63	Word	模拟量输入数据映射区		1186~1249
AQ	AQ0~AQ63	Word	模拟量输出数据映射区		1250~1313
T	T0~T255	Word	定时器数值区		1314~1569
C	C0~C199	Word	普通计数器数值区		1570~1769
C	C200~C255	DWord	高速计数器数值区		1770~1881
HC	HC0~HC15	DWord	高速计数器数值区		1882~1913
SD	SD0~SD999	Word	用户系统或程序状态数值区		1914~2913
D	D0~D9999	Word	数据存储区		2914~12913
E	E0~E31	Word	变址寄存器 E 区		12914~12945
ND	ND0~ND309	Word	CC-CAN 网络专用数值区		12946~13255



## 二 SM、SD 使用说明

### 2.1 SM 定义列表

属性	功能说明
RO	只读
RW	可读写

#### 2.1.1 通用 SM

编号	点位	名称	属性
0	0	系统错误标志位	RO
	1	系统警告标志位	RO
	2	系统运行常 ON 标志位	RO
	3	系统运行常 OFF 标志位	RO
	4	系统运行正向脉冲接点标志位	RO
	5	系统运行负向脉冲接点标志位	RO
	6	PLC 停机时, 是否将非停电保持区域全部清除	RO
	7	非停电保持区域全部清除开关	RW
1	0	停电保持区域全部清除开关	RW
	1	电池电压标志位	RO
	2	扫描周期超时标志位	RO
	3	复位 PLC 主机控制位	RW
	4	系统运行停止控制软开关是否有效	RO
	5	系统运行停止状态及设置	RW
	7	扫描周期变更有效位	RW
2	0	10ms 时钟脉冲标志位	RO
	1	100ms 时钟脉冲标志位	RO
	2	1s 时钟脉冲标志位	RO
	3	1min 时钟脉冲标志位	RO
10	0	输入中断程序 0 的启用标志位	RO
	1	输入中断程序 1 的启用标志位	RO
	2	输入中断程序 2 的启用标志位	RO
	3	输入中断程序 3 的启用标志位	RO
	4	输入中断程序 4 的启用标志位	RO
	5	输入中断程序 5 的启用标志位	RO
11	0	输入中断程序 0 的边沿设置选择位	RO
	1	输入中断程序 1 的边沿设置选择位	RO
	2	输入中断程序 2 的边沿设置选择位	RO
	3	输入中断程序 3 的边沿设置选择位	RO
20	0	中断程序 0 使能标志位	RO
	1	中断程序 1 使能标志位	RO
	2	中断程序 2 使能标志位	RO

编号	点位	名称	属性
	3	中断程序 3 使能标志位	RO
20	4	中断程序 4 使能标志位	RO
	5	中断程序 5 使能标志位	RO
28	0	模拟量输入通道 0 的数据错误标志位	RO
	1	模拟量输入通道 1 的数据错误标志位	RO
	2	模拟量输入通道 2 的数据错误标志位	RO
	3	模拟量输入通道 3 的数据错误标志位	RO
	4	模拟量输入通道 4 的数据错误标志位	RO
	5	模拟量输入通道 5 的数据错误标志位	RO
	6	模拟量输入通道 6 的数据错误标志位	RO
29	7	模拟量输入通道 7 的数据错误标志位	RO
	0	模拟量输入通道 8 的数据错误标志位	RO
	1	模拟量输入通道 9 的数据错误标志位	RO
	2	模拟量输入通道 10 的数据错误标志位	RO
	3	模拟量输入通道 11 的数据错误标志位	RO
	4	模拟量输入通道 12 的数据错误标志位	RO
	5	模拟量输入通道 13 的数据错误标志位	RO
30	6	模拟量输入通道 14 的数据错误标志位	RO
	7	模拟量输入通道 15 的数据错误标志位	RO
	0	模拟量输入通道 16 的数据错误标志位	RO
	1	模拟量输入通道 17 的数据错误标志位	RO
	2	模拟量输入通道 18 的数据错误标志位	RO
	3	模拟量输入通道 19 的数据错误标志位	RO
	4	模拟量输入通道 20 的数据错误标志位	RO
31	5	模拟量输入通道 21 的数据错误标志位	RO
	6	模拟量输入通道 22 的数据错误标志位	RO
	7	模拟量输入通道 23 的数据错误标志位	RO
	0	模拟量输入通道 24 的数据错误标志位	RO
	1	模拟量输入通道 25 的数据错误标志位	RO
	2	模拟量输入通道 26 的数据错误标志位	RO
	3	模拟量输入通道 27 的数据错误标志位	RO
32	4	模拟量输入通道 28 的数据错误标志位	RO
	5	模拟量输入通道 29 的数据错误标志位	RO
	6	模拟量输入通道 30 的数据错误标志位	RO
	7	模拟量输入通道 31 的数据错误标志位	RO
	0	模拟量输入通道 32 的数据错误标志位	RO
	1	模拟量输入通道 33 的数据错误标志位	RO
32	2	模拟量输入通道 34 的数据错误标志位	RO
	3	模拟量输入通道 35 的数据错误标志位	RO
	4	模拟量输入通道 36 的数据错误标志位	RO
	5	模拟量输入通道 37 的数据错误标志位	RO



编号	点位	名称	属性
	6	模拟量输入通道 38 的数据错误标志位	RO
32	7	模拟量输入通道 39 的数据错误标志位	RO
33	0	模拟量输入通道 40 的数据错误标志位	RO
	1	模拟量输入通道 41 的数据错误标志位	RO
	2	模拟量输入通道 42 的数据错误标志位	RO
	3	模拟量输入通道 43 的数据错误标志位	RO
	4	模拟量输入通道 44 的数据错误标志位	RO
	5	模拟量输入通道 45 的数据错误标志位	RO
	6	模拟量输入通道 46 的数据错误标志位	RO
	7	模拟量输入通道 47 的数据错误标志位	RO
34	0	模拟量输入通道 48 的数据错误标志位	RO
	1	模拟量输入通道 49 的数据错误标志位	RO
	2	模拟量输入通道 50 的数据错误标志位	RO
	3	模拟量输入通道 51 的数据错误标志位	RO
	4	模拟量输入通道 52 的数据错误标志位	RO
	5	模拟量输入通道 53 的数据错误标志位	RO
	6	模拟量输入通道 54 的数据错误标志位	RO
	7	模拟量输入通道 55 的数据错误标志位	RO
35	0	模拟量输入通道 56 的数据错误标志位	RO
	1	模拟量输入通道 57 的数据错误标志位	RO
	2	模拟量输入通道 58 的数据错误标志位	RO
	3	模拟量输入通道 59 的数据错误标志位	RO
	4	模拟量输入通道 60 的数据错误标志位	RO
	5	模拟量输入通道 61 的数据错误标志位	RO
	6	模拟量输入通道 62 的数据错误标志位	RO
	7	模拟量输入通道 63 的数据错误标志位	RO
40	0	C180 的上/下计数方式选择开关	RW
	1	C181 的上/下计数方式选择开关	RW
	2	C182 的上/下计数方式选择开关	RW
	3	C183 的上/下计数方式选择开关	RW
	4	C184 的上/下计数方式选择开关	RW
	5	C185 的上/下计数方式选择开关	RW
	6	C186 的上/下计数方式选择开关	RW
	7	C187 的上/下计数方式选择开关	RW
41	0	C188 的上/下计数方式选择开关	RW
	1	C189 的上/下计数方式选择开关	RW
	2	C190 的上/下计数方式选择开关	RW
	3	C191 的上/下计数方式选择开关	RW
	4	C192 的上/下计数方式选择开关	RW
	5	C193 的上/下计数方式选择开关	RW
	6	C194 的上/下计数方式选择开关	RW

编号	点位	名称	属性
	7	C195 的上/下计数方式选择开关	RW
42	0	C196 的上/下计数方式选择开关	RW
	1	C197 的上/下计数方式选择开关	RW
	2	C198 的上/下计数方式选择开关	RW
	3	C199 的上/下计数方式选择开关	RW
45	0	C242 的上/下计数方式选择开关	RW
	1	C243 的上/下计数方式选择开关	RW
	2	C244 的上/下计数方式选择开关	RW
	3	C245 的上/下计数方式选择开关	RW
	4	C246 的上/下计数方式选择开关	RW
	5	C247 的上/下计数方式选择开关	RW
	6	C248 的上/下计数方式选择开关	RW
46	7	C249 的上/下计数方式选择开关	RW
	0	C250 的上/下计数方式选择开关	RW
	1	C251 的上/下计数方式选择开关	RW
	2	C252 的上/下计数方式选择开关	RW
	3	C253 的上/下计数方式选择开关	RW
	4	C254 的上/下计数方式选择开关	RW
50	5	C255 的上/下计数方式选择开关	RW
	0	SMOV 指令的操作数工作模式	RW
	1	XCH 指令的操作数工作模式	RW
	2	四则运算零标志位	RW
	3	四则运算借位标志位	RW
	4	四则运算进位标志位	RW
	5	CMPT 指令比较模式设定位	RW
	6	编码 ENCO 指令执行失败标志位	RW
51	7	MTR 指令的正在执行标志位	RO
	0	HKY 指令的正在执行标志位	RO
	1	DHKY 指令的正在执行标志位	RO
	2	SORT 指令的排序模式	RW
	3	INCD 指令执行完毕标志信号	RW
	4	RAMP 指令执行完毕标志信号	RW
	5	PWD 指令的工作模式	RW
	6	HST 指令的高速计时正在执行标志	RO
52	7	三角函数操作数模式	RW
	0	浮点数无法正常表达数据标志位	RW
53	0	矩阵比较模式	RW
	1	矩阵位寻找成功标志位	RW
	2	矩阵位寻找结束标志位	RW
	3	矩阵旋转位移输出进位标志位	RW
	4	矩阵旋转位移输入补位标志位	RW



编号	点位	名称	属性
	5	矩阵左右移位模式	RW
53	6	矩阵计数类型模式	RW
	7	万年历正在变更标志位	RW
60	0	PID 通道 0 自整定启动标志位	RW
	1	PID 通道 1 自整定启动标志位	RW
	2	PID 通道 2 自整定启动标志位	RW
	3	PID 通道 3 自整定启动标志位	RW
	4	PID 通道 4 自整定启动标志位	RW
	5	PID 通道 5 自整定启动标志位	RW
	6	PID 通道 6 自整定启动标志位	RW
	7	PID 通道 7 自整定启动标志位	RW
61	0	PID 通道 8 自整定启动标志位	RW
	1	PID 通道 9 自整定启动标志位	RW
	2	PID 通道 10 自整定启动标志位	RW
	3	PID 通道 11 自整定启动标志位	RW
	4	PID 通道 12 自整定启动标志位	RW
	5	PID 通道 13 自整定启动标志位	RW
	6	PID 通道 14 自整定启动标志位	RW
	7	PID 通道 15 自整定启动标志位	RW
62	0	PID 通道 16 自整定启动标志位	RW
	1	PID 通道 17 自整定启动标志位	RW
	2	PID 通道 18 自整定启动标志位	RW
	3	PID 通道 19 自整定启动标志位	RW
	4	PID 通道 20 自整定启动标志位	RW
	5	PID 通道 21 自整定启动标志位	RW
	6	PID 通道 22 自整定启动标志位	RW
	7	PID 通道 23 自整定启动标志位	RW
63	0	PID 通道 24 自整定启动标志位	RW
	1	PID 通道 25 自整定启动标志位	RW
	2	PID 通道 26 自整定启动标志位	RW
	3	PID 通道 27 自整定启动标志位	RW
	4	PID 通道 28 自整定启动标志位	RW
	5	PID 通道 29 自整定启动标志位	RW
	6	PID 通道 30 自整定启动标志位	RW
	7	PID 通道 31 自整定启动标志位	RW
65	0	RS485_1 通讯结束标志位	RO
	1	RS485_2 通讯结束标志位	RO
66	0	RS485_1 正在发送标志位	RO
	1	RS485_2 正在发送标志位	RO
67	0	RS485_1 做主站时的通讯异常标志位	RW
	1	RS485_2 做主站时的通讯异常标志位	RW

编号	点位	名称	属性
68	0	RS485_1 起始字符启用开关	RW
68	1	RS485_2 起始字符启用开关	RW
69	0	RS485_1 结束字符启用开关	RW
	1	RS485_2 结束字符启用开关	RW
70	0	RS485_1 启用 RS 指令开关	RW
	1	RS485_2 启用 RS 指令开关	RW
71	0	RS485_1 使用 RS 指令时处理数据模式	RW
	1	RS485_2 使用 RS 指令时处理数据模式	RW
72	0	RS485_1 广播延时标志位	RO
	1	RS485_2 广播延时标志位	RO
75	0	通讯数据处理类指令的模式	RW
82	0	HC0 正在计数标志位	RO
	1	HC1 正在计数标志位	RO
	2	HC2 正在计数标志位	RO
	3	HC3 正在计数标志位	RO
83	0	HC0 正在使用 HCNT 指令标志位	RO
	1	HC1 正在使用 HCNT 指令标志位	RO
	2	HC2 正在使用 HCNT 指令标志位	RO
	3	HC3 正在使用 HCNT 指令标志位	RO
84	0	HC0 正在使用 SPD 指令标志位	RO
	1	HC1 正在使用 SPD 指令标志位	RO
	2	HC2 正在使用 SPD 指令标志位	RO
	3	HC3 正在使用 SPD 指令标志位	RO
88	0	Y0.0 输出保持标志位	RW
	1	Y0.1 输出保持标志位	RW
	2	Y0.2 输出保持标志位	RW
	3	Y0.3 输出保持标志位	RW
	4	Y0.4 输出保持标志位	RW
	5	Y0.5 输出保持标志位	RW
	6	Y0.6 输出保持标志位	RW
	7	Y0.7 输出保持标志位	RW
89	0	Y1.0 输出保持标志位	RW
	1	Y1.1 输出保持标志位	RW
	2	Y1.2 输出保持标志位	RW
	3	Y1.3 输出保持标志位	RW
	4	Y1.4 输出保持标志位	RW
	5	Y1.5 输出保持标志位	RW
	6	Y1.6 输出保持标志位	RW
	7	Y1.7 输出保持标志位	RW
90	0	Y2.0 输出保持标志位	RW
	1	Y2.1 输出保持标志位	RW



编号	点位	名称	属性
	2	Y2.2 输出保持标志位	RW
90	3	Y2.3 输出保持标志位	RW
	4	Y2.4 输出保持标志位	RW
	5	Y2.5 输出保持标志位	RW
	6	Y2.6 输出保持标志位	RW
	7	Y2.7 输出保持标志位	RW
91	0	Y3.0 输出保持标志位	RW
	1	Y3.1 输出保持标志位	RW
	2	Y3.2 输出保持标志位	RW
	3	Y3.3 输出保持标志位	RW
	4	Y3.4 输出保持标志位	RW
	5	Y3.5 输出保持标志位	RW
	6	Y3.6 输出保持标志位	RW
	7	Y3.7 输出保持标志位	RW

### 2.1.2 CC-CAN 总线 SM (使用说明见 10.1 章节)

编号	点位	名称	属性
108	0	空余	RW
	1	站号 1 伺服停止方式选择	RW
	2	站号 2 伺服停止方式选择	RW
	3	站号 3 伺服停止方式选择	RW
	4	站号 4 伺服停止方式选择	RW
	5	站号 5 伺服停止方式选择	RW
	6	站号 6 伺服停止方式选择	RW
	7	站号 7 伺服停止方式选择	RW
109	0	站号 8 伺服停止方式选择	RW
	1	站号 9 伺服停止方式选择	RW
	2	站号 10 伺服停止方式选择	RW
	3	站号 11 伺服停止方式选择	RW
	4	站号 12 伺服停止方式选择	RW
	5	站号 13 伺服停止方式选择	RW
	6	站号 14 伺服停止方式选择	RW
	7	站号 15 伺服停止方式选择	RW
110	0	站号 16 伺服停止方式选择	RW
	1	站号 17 伺服停止方式选择	RW
	2	站号 18 伺服停止方式选择	RW
	3	站号 19 伺服停止方式选择	RW
	4	站号 20 伺服停止方式选择	RW
	5	站号 21 伺服停止方式选择	RW
	6	站号 22 伺服停止方式选择	RW
	7	站号 23 伺服停止方式选择	RW

编号	点位	名称	属性
111	0	站号 24 伺服停止方式选择	RW
	1	站号 25 伺服停止方式选择	RW
	2	站号 26 伺服停止方式选择	RW
	3	站号 27 伺服停止方式选择	RW
	4	站号 28 伺服停止方式选择	RW
	5	站号 29 伺服停止方式选择	RW
	6	站号 30 伺服停止方式选择	RW
	7	空余	RW
112	0	空余	RW
	1	站号 1 伺服的机械实际位置清 0 标志位	RW
	2	站号 2 伺服的机械实际位置清 0 标志位	RW
	3	站号 3 伺服的机械实际位置清 0 标志位	RW
	4	站号 4 伺服的机械实际位置清 0 标志位	RW
	5	站号 5 伺服的机械实际位置清 0 标志位	RW
	6	站号 6 伺服的机械实际位置清 0 标志位	RW
	7	站号 7 伺服的机械实际位置清 0 标志位	RW
113	0	站号 8 伺服的机械实际位置清 0 标志位	RW
	1	站号 9 伺服的机械实际位置清 0 标志位	RW
	2	站号 10 伺服的机械实际位置清 0 标志位	RW
	3	站号 11 伺服的机械实际位置清 0 标志位	RW
	4	站号 12 伺服的机械实际位置清 0 标志位	RW
	5	站号 13 伺服的机械实际位置清 0 标志位	RW
	6	站号 14 伺服的机械实际位置清 0 标志位	RW
	7	站号 15 伺服的机械实际位置清 0 标志位	RW
114	0	站号 16 伺服的机械实际位置清 0 标志位	RW
	1	站号 17 伺服的机械实际位置清 0 标志位	RW
	2	站号 18 伺服的机械实际位置清 0 标志位	RW
	3	站号 19 伺服的机械实际位置清 0 标志位	RW
	4	站号 20 伺服的机械实际位置清 0 标志位	RW
	5	站号 21 伺服的机械实际位置清 0 标志位	RW
	6	站号 22 伺服的机械实际位置清 0 标志位	RW
	7	站号 23 伺服的机械实际位置清 0 标志位	RW
115	0	站号 24 伺服的机械实际位置清 0 标志位	RW
	1	站号 25 伺服的机械实际位置清 0 标志位	RW
	2	站号 26 伺服的机械实际位置清 0 标志位	RW
	3	站号 27 伺服的机械实际位置清 0 标志位	RW
	4	站号 28 伺服的机械实际位置清 0 标志位	RW
	5	站号 29 伺服的机械实际位置清 0 标志位	RW
	6	站号 30 伺服的机械实际位置清 0 标志位	RW
	7	空余	RW
116	0	CC-CAN 网络错误标志位	RO



编号	点位	名称	属性
116	1	CC-CAN 网络警告标志位	RO
	2	空余	RO
	3	CC-CAN 电子手轮忙碌标志位	RO
	4	CC-CAN 电子手轮信息更新标志位	RO
	5	空余	RO
	6	空余	RO
117	0	空余	RO
	1	站号 1 忙碌标志位	RO
	2	站号 2 忙碌标志位	RO
	3	站号 3 忙碌标志位	RO
	4	站号 4 忙碌标志位	RO
	5	站号 5 忙碌标志位	RO
	6	站号 6 忙碌标志位	RO
118	7	站号 7 忙碌标志位	RO
	0	站号 8 忙碌标志位	RO
	1	站号 9 忙碌标志位	RO
	2	站号 10 忙碌标志位	RO
	3	站号 11 忙碌标志位	RO
	4	站号 12 忙碌标志位	RO
	5	站号 13 忙碌标志位	RO
	6	站号 14 忙碌标志位	RO
119	7	站号 15 忙碌标志位	RO
	0	站号 16 忙碌标志位	RO
	1	站号 17 忙碌标志位	RO
	2	站号 18 忙碌标志位	RO
	3	站号 19 忙碌标志位	RO
	4	站号 20 忙碌标志位	RO
	5	站号 21 忙碌标志位	RO
	6	站号 22 忙碌标志位	RO
120	7	站号 23 忙碌标志位	RO
	0	站号 24 忙碌标志位	RO
	1	站号 25 忙碌标志位	RO
	2	站号 26 忙碌标志位	RO
	3	站号 27 忙碌标志位	RO
	4	站号 28 忙碌标志位	RO
	5	站号 29 忙碌标志位	RO
	6	站号 30 忙碌标志位	RO
121	7	站号 31 忙碌标志位	RO
	0	空余	RO
	1	站号 1 运动控制标志位	RO
	2	站号 2 运动控制标志位	RO

编号	点位	名称	属性
121	3	站号 3 运动控制标志位	RO
	4	站号 4 运动控制标志位	RO
	5	站号 5 运动控制标志位	RO
	6	站号 6 运动控制标志位	RO
	7	站号 7 运动控制标志位	RO
122	0	站号 8 运动控制标志位	RO
	1	站号 9 运动控制标志位	RO
	2	站号 10 运动控制标志位	RO
	3	站号 11 运动控制标志位	RO
	4	站号 12 运动控制标志位	RO
	5	站号 13 运动控制标志位	RO
	6	站号 14 运动控制标志位	RO
123	7	站号 15 运动控制标志位	RO
	0	站号 16 运动控制标志位	RO
	1	站号 17 运动控制标志位	RO
	2	站号 18 运动控制标志位	RO
	3	站号 19 运动控制标志位	RO
	4	站号 20 运动控制标志位	RO
	5	站号 21 运动控制标志位	RO
	6	站号 22 运动控制标志位	RO
124	7	站号 23 运动控制标志位	RO
	0	站号 24 运动控制标志位	RO
	1	站号 25 运动控制标志位	RO
	2	站号 26 运动控制标志位	RO
	3	站号 27 运动控制标志位	RO
	4	站号 28 运动控制标志位	RO
	5	站号 29 运动控制标志位	RO
	6	站号 30 运动控制标志位	RO



## 2.2 SD 定义列表

属性	功能说明
RO	只读
RW	可读写

### 2.2.1 通用 SD

编号	用途说明	属性	默认值
0	系统错误码	RO	0
1	程序容量 (单位: 步)	RO	20000
2	PLC 主机程序主版本编号	RO	-
3	PLC 主机程序子版本编号	RO	-
4	PLC 权限等级设定	RO	0
5	扫描周期监控超时时间 (单位: ms)	RO	200
6	当前扫描时间长度	RO	0
7	最小扫描周期时长	RO	0
8	最大扫描周期时长	RO	0
10	$\pi$ (Pi) 浮点数值的高字存放寄存器	RO	16#0FDA
11	$\pi$ (Pi) 浮点数值的高字存放寄存器	RO	16#4049
25	掉电保持区域 T 装置的起始地址	RO	250
26	掉电保持区域 T 装置的结束地址	RO	255
27	掉电保持区域 16 位 C 装置的起始地址	RO	190
28	掉电保持区域 16 位 C 装置的结束地址	RO	199
29	掉电保持区域 32 位 C 装置的起始地址	RO	250
30	掉电保持区域 32 位 C 装置的结束地址	RO	255
31	掉电保持区域 HC 装置的起始地址	RO	0
32	掉电保持区域 HC 装置的结束地址	RO	3
33	掉电保持区域 M 装置的起始地址	RO	2000
34	掉电保持区域 M 装置的结束地址	RO	2047
35	掉电保持区域 D 装置的起始地址	RO	9000
36	掉电保持区域 D 装置的结束地址	RO	9999
37	掉电保持区域 E 装置的起始地址	RO	30
38	掉电保持区域 E 装置的结束地址	RO	31
40	中断程序 0 的优先级设定	RO	0
41	中断程序 1 的优先级设定	RO	0
42	中断程序 2 的优先级设定	RO	0
43	中断程序 3 的优先级设定	RO	0
44	中断程序 4 的优先级设定	RO	0
45	中断程序 5 的优先级设定	RO	0

编号	用途说明	属性	默认值
60	定时中断程序的时间设置	RO	5
80	扩展模块的错误码	RO	0
100	MTR, HKY 指令的输出扫描间隔时间设定 (单位: ms)	RW	20
101	HST 指令的高速计时值	RO	0
120	万年历秒针实时寄存器	RO	0
121	万年历分针实时寄存器	RO	0
122	万年历时针实时寄存器	RO	0
123	万年历星期实时寄存器	RO	0
124	万年历日期实时寄存器	RO	0
125	万年历月份实时寄存器	RO	0
126	万年历年份实时寄存器	RO	0
130	万年历秒针写入寄存器	RW	0
131	万年历分针写入寄存器	RW	0
132	万年历时针写入寄存器	RW	0
133	万年历星期写入寄存器	RW	0
134	万年历日期写入寄存器	RW	0
135	万年历月份写入寄存器	RW	0
136	万年历年份写入寄存器	RW	0
140	RTM 指令第 0 组的测量时间放置处	RW	0
141	RTM 指令第 1 组的测量时间放置处	RW	0
142	RTM 指令第 2 组的测量时间放置处	RW	0
143	RTM 指令第 3 组的测量时间放置处	RW	0
144	RTM 指令第 4 组的测量时间放置处	RW	0
145	RTM 指令第 5 组的测量时间放置处	RW	0
146	RTM 指令第 6 组的测量时间放置处	RW	0
147	RTM 指令第 7 组的测量时间放置处	RW	0
148	RTM 指令第 8 组的测量时间放置处	RW	0
149	RTM 指令第 9 组的测量时间放置处	RW	0
200	RS485_1 的应用层协议设定	RO	0
201	RS485_2 的应用层协议设定	RO	0
205	RS485_1 的波特率设定	RO	5
206	RS485_2 的波特率设定	RO	5
210	RS485_1 的 MODBUS 站号设定	RO	1
211	RS485_2 的 MODBUS 站号设定	RO	1
215	RS485_1 的 MAC 层格式设定	RO	1
216	RS485_2 的 MAC 层格式设定	RO	1
220	RS485_1 的超时时间设定 (单位: ×10ms)	RO	20



编号	用途说明	属性	默认值
221	RS485_2 的超时时间设定 (单位: $\times 10\text{ms}$ )	RO	20
225	RS485_1 使用 RS 指令时接收到的数据字节数	RO	0
226	RS485_2 使用 RS 指令时接收到的数据字节数	RO	0
230	RS485_1 使用 RS 指令时的起始字符 (低字节有效)	RW	0
231	RS485_2 使用 RS 指令时的起始字符 (低字节有效)	RW	0
235	RS485_1 使用 RS 指令时的结束字符 (低字节有效)	RW	0
236	RS485_2 使用 RS 指令时的结束字符 (低字节有效)	RW	0
240	RS485_1 做从站时延迟响应时间设定 (单位: ms)	RW	0
241	RS485_2 做从站时延迟响应时间设定 (单位: ms)	RW	0
245	RS485_1 做主站时的通讯错误码	RW	0
246	RS485_2 做主站时的通讯错误码	RW	0
300~349	RS485_1 使用 MODBUS 指令的数据处理区域	RW	0
350~399	RS485_2 使用 MODBUS 指令的数据处理区域	RW	0
500~515	X0.0~X3.7 滤波时间设置 (单位: ms, 排序低的在低字节)	RO	16#1414
520	X0.0 的输入功能类型	RO	0
521	X0.1 的输入功能类型	RO	0
522	X0.2 的输入功能类型	RO	0
523	X0.3 的输入功能类型	RO	0

## 2.2.2 CC-CAN 总线 SD (使用说明见 10.1 章节)

### (1) 功能: 伺服电机每转位移的机械单元数 (掉电保持)

编号	用途说明	属性	默认值
602-603	站号 1 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
604-605	站号 2 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
606-607	站号 3 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
608-609	站号 4 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
610-611	站号 5 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
612-613	站号 6 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
614-615	站号 7 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
616-617	站号 8 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
618-619	站号 9 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
620-621	站号 10 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
622-623	站号 11 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
624-625	站号 12 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
626-627	站号 13 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
628-629	站号 14 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
630-631	站号 15 伺服当前的机械实际绝对位置 (掉电保持)	RO	0

编号	用途说明	属性	默认值
632-633	站号 16 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
634-635	站号 17 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
636-637	站号 18 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
638-639	站号 19 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
640-641	站号 20 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
642-643	站号 21 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
644-645	站号 22 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
646-647	站号 23 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
648-649	站号 24 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
650-651	站号 25 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
652-653	站号 26 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
654-655	站号 27 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
656-657	站号 28 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
658-659	站号 29 伺服当前的机械实际绝对位置 (掉电保持)	RO	0
660-661	站号 30 伺服当前的机械实际绝对位置 (掉电保持)	RO	0

(2) 功能：CC-CAN 错误码

编号	用途说明	属性	默认值
800	CC-CAN 网络错误码	RO	0
801	站号 1 错误码	RO	0
802	站号 2 错误码	RO	0
803	站号 3 错误码	RO	0
804	站号 4 错误码	RO	0
805	站号 5 错误码	RO	0
806	站号 6 错误码	RO	0
807	站号 7 错误码	RO	0
808	站号 8 错误码	RO	0
809	站号 9 错误码	RO	0
810	站号 10 错误码	RO	0
811	站号 11 错误码	RO	0
812	站号 12 错误码	RO	0
813	站号 13 错误码	RO	0
814	站号 14 错误码	RO	0
815	站号 15 错误码	RO	0
816	站号 16 错误码	RO	0
817	站号 17 错误码	RO	0
818	站号 18 错误码	RO	0
819	站号 19 错误码	RO	0



编号	用途说明	属性	默认值
820	站号 20 错误码	RO	0
821	站号 21 错误码	RO	0
822	站号 22 错误码	RO	0
823	站号 23 错误码	RO	0
824	站号 24 错误码	RO	0
825	站号 25 错误码	RO	0
826	站号 26 错误码	RO	0
827	站号 27 错误码	RO	0
828	站号 28 错误码	RO	0
829	站号 29 错误码	RO	0
830	站号 30 错误码	RO	0
831	站号 31 错误码	RO	0

## (3) 功能：伺服当前的程序绝对位置

编号	用途说明	属性	默认值
832-833	站号 1 伺服当前的程序绝对位置	RW	0
834-835	站号 2 伺服当前的程序绝对位置	RW	0
836-837	站号 3 伺服当前的程序绝对位置	RW	0
838-839	站号 4 伺服当前的程序绝对位置	RW	0
840-841	站号 5 伺服当前的程序绝对位置	RW	0
842-843	站号 6 伺服当前的程序绝对位置	RW	0
844-845	站号 7 伺服当前的程序绝对位置	RW	0
846-847	站号 8 伺服当前的程序绝对位置	RW	0
848-849	站号 9 伺服当前的程序绝对位置	RW	0
850-851	站号 10 伺服当前的程序绝对位置	RW	0
852-853	站号 11 伺服当前的程序绝对位置	RW	0
854-855	站号 12 伺服当前的程序绝对位置	RW	0
856-857	站号 13 伺服当前的程序绝对位置	RW	0
858-859	站号 14 伺服当前的程序绝对位置	RW	0
860-861	站号 15 伺服当前的程序绝对位置	RW	0
862-863	站号 16 伺服当前的程序绝对位置	RW	0
864-865	站号 17 伺服当前的程序绝对位置	RW	0
866-867	站号 18 伺服当前的程序绝对位置	RW	0
868-869	站号 19 伺服当前的程序绝对位置	RW	0
870-871	站号 20 伺服当前的程序绝对位置	RW	0
872-873	站号 21 伺服当前的程序绝对位置	RW	0
874-875	站号 22 伺服当前的程序绝对位置	RW	0
876-877	站号 23 伺服当前的程序绝对位置	RW	0

编号	用途说明	属性	默认值
878-879	站号 24 伺服当前的程序绝对位置	RW	0
880-881	站号 25 伺服当前的程序绝对位置	RW	0
882-883	站号 26 伺服当前的程序绝对位置	RW	0
884-885	站号 27 伺服当前的程序绝对位置	RW	0
886-887	站号 28 伺服当前的程序绝对位置	RW	0
888-889	站号 29 伺服当前的程序绝对位置	RW	0
890-891	站号 30 伺服当前的程序绝对位置	RW	0

(4) 功能：CC-CAN 电子手轮对应伺服站号设定

编号	用途说明	属性	默认值
901	CC-CAN 电子手轮 1 轴对应伺服站号设定	RW	1
902	CC-CAN 电子手轮 2 轴对应伺服站号设定	RW	2
903	CC-CAN 电子手轮 3 轴对应伺服站号设定	RW	3
904	CC-CAN 电子手轮 4 轴对应伺服站号设定	RW	4
905	CC-CAN 电子手轮 5 轴对应伺服站号设定	RW	5
906	CC-CAN 电子手轮 6 轴对应伺服站号设定	RW	6
907	CC-CAN 电子手轮 7 轴对应伺服站号设定	RW	7



## 2.3 SM 和 SD 群组功能说明

### 2.3.1 PLC 系统类别

#### (1) PLC 系统运行标志位

1, SM0.2: 系统运行常 ON 标志位;

当 PLC 处于运行状态, 该 SM 保持为 ON; 当 PLC 处于停止状态时, 该 SM 为 OFF。

2, SM0.3: 系统运行常 OFF 标志位;

当 PLC 处于运行状态, 该 SM 保持为 OFF; 当 PLC 处于停止状态时, 该 SM 置 ON。

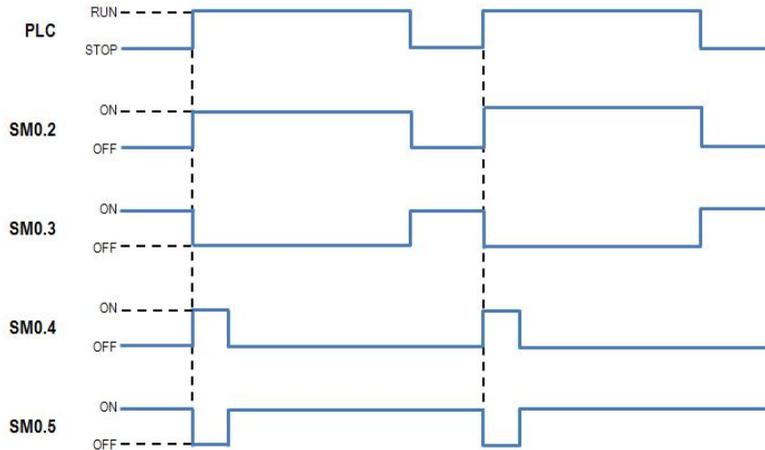
3, SM0.4: 系统运行正向脉冲接点标志位;

PLC 从 STOP->RUN 的第一次扫描周期为 ON, 之后一直为 OFF, 当要做各种初始化设置时可使用本接点。

4, SM0.5: 系统运行负向脉冲接点标志位;

PLC 从 STOP->RUN 的第一次扫描周期为 OFF, 之后一直为 ON。

SM0.2~SM0.5 的关系如下图所示:



5, SM1.4: 系统运行停止控制软开关是否有效;

SM1.5: 系统运行停止状态及设置;

当 SM1.4 为 ON, 表示系统运行停止控制软开关有效, 此时将 SM1.5 置 ON, PLC 处于运行状态; 将 SM1.5 置 OFF, PLC 处于停止状态; 当 SM1.4 为 OFF, 表示系统运行停止控制软开关无效, 此时 SM1.5 接点失效;

## (2) PLC 系统参数及标志位

## 1, SM0.0: 系统错误标志位;

当系统出现错误时, 该标志位置 ON; 若系统正常, 则 SM0.0 置 OFF。

## 2, SD0: 系统错误码。

对应错误码值如下表所示:

错误码	原因
1	PLC 主机中没有用户程序
2	PLC 主机用户程序存储器出错
3	PLC 主机用户掉电保持区域可能损坏
4	PLC 主机用户掉电保持区域出错
5	用户程序逻辑/语法/变量使用错误或低版本无法识别新指令
6	扩展模块错误
7	220V 电源掉电或低电压
8	24V 输出电源过载
9	用户程序运行扫描周期超时
10	用户程序局部变量使用过多
11	用户程序的功能卡嵌套层数过多
12	高温报错, 主机温度高于 65°C 时, 主机停机并报错; 重新低于 60°C 后, 主机恢复正常
13	CC-CAN 网络有错误出现
30	CC-CAN 网络有警告出现

## 3, SM0.1: 系统警告标志位;

当系统出现警告时, SM0.1 标志位置 ON。

## 4, SM1.1: 电池电压标志位;

当 PLC 自带的电池电压过低, 故障或者无电池, 该标志位置 ON; 若自带电池正常, 该标志位置 OFF。

## 5, SD1: 主机程序容量 (单位: 步);

## 6, SD2: PLC 主机程序主版本编号;

SD3: PLC 主机程序子版本编号;

## 7, SD4: PLC 权限等级设置。

记录该主机的权限等级。权限等级如下表所示:

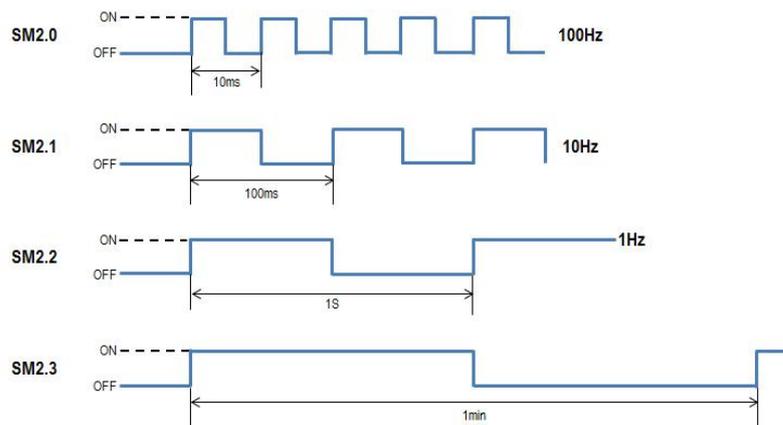
值	权限等级
0	全部权限, 所有 PLC 功能均可使用
1	部分权限, 只能读写数据及上载程序。下载需要密码
2	最小权限, 只能读写 PLC 数据。上载、下载需要密码
3	禁止上载, 只能读写 PLC 数据。下载需要密码, 但禁止上载程序

(3) 扫描周期参数及标志位

- 1, SD6: 当前扫描时间长度;  
 SD7: 最小扫描周期时长;  
 SD8: 最大扫描周期时长;
- 2, SD5: 扫描周期监控超时时间;  
 SM1.2: 扫描周期超时标志位;  
 SD5 默认为 200ms, 当超出设定时间后, SM 置 ON, PLC 主机将停止运行并报错。
- 3, SM1.7: 扫描周期设定变更有效位;  
 设置任何扫描周期相关参数后, 必须将 SM 置 ON 一个扫描周期, 才能使得设置的值生效, 否则设置值无效。

(4) 内部时钟脉冲

- 1, PLC 主机内具备下列四种时钟脉冲, 只要 PLC 通上电源, 这四种时钟脉冲就会自动工作。  
 SM2.0: 10ms 时钟脉冲, 5ms On, 5ms Off;  
 SM2.1: 100ms 时钟脉冲, 50ms On, 50ms Off;  
 SM2.2: 1s 时钟脉冲, 500ms On, 500ms Off;  
 SM2.3: 1min 时钟脉冲, 30s On, 30s Off。



- 2, PLC 处于 STOP 情况下, 时钟脉冲也会动作, 时钟脉冲启动时序及 RUN 的启动时序不会同步。

(5) 停电保持范围

PLC 停电保持范围装置, 依照开始地址编号到结束地址编号的范围为停电保持区域, 该区域为只读数据区域。

装置类型	起始地址所在 SD	结束地址所在 SD
T	SD25	SD26
16 位 C	SD27	SD28
32 位 C	SD29	SD30
HC	SD31	SD32
M	SD33	SD34
D	SD35	SD36
E	SD37	SD38

## (6) 数据清除

- 1, SM0.6: PLC 停机时, 是否将非停电保持区域全部清除;  
将 SM 置 ON, 当 PLC 停机时, 将非停电保持区域全部清除一次;
- 2, SM0.7: 非停电保持区域全部清除开关;  
将 SM 置 ON, 非停电保持区域数据将被全部清除一次, 同时会将该 SM 自动置 OFF;
- 3, SM1.0: 停电保持区域全部清除开关;  
将 SM 置 ON, 停电保持区域数据将被全部清除一次, 同时会将该 SM 自动置 OFF;

## (7) 中断程序功能参数及标志位

- 1, SM10.0~SM10.5: 输入中断程序 0~5 的启用标志位;  
若存在中断程序时, 对应的 SM 将置 ON; 若不存在中断程序时, 对应的 SM 将置 OFF。
- 2, SM11.0~SM11.3: 输入中断程序 0~3 的边沿设置选择位;  
若 SM 为 ON, 对应的输入中断程序为上升沿触发; 若 SM 为 OFF, 对应的输入中断程序为下降沿触发。
- 3, SM20.0~SM20.5: 中断程序 0~5 使能标志位;  
若 SM 为 ON, 对应中断程序为使能状态; 若 SM 为 OFF, 对应中断程序为禁用状态。
- 4, SD40~SD45: 中断程序 0~5 的优先级设定;  
SD 可设定值为 0~2; 数值 0 优先级为高; 数值 1 优先级为中; 数值 2 优先级为低;
- 5, SD60: 定时中断程序的时间设置;  
单位为 ms;

## (8) 扩展模块

- 1, SD80: 扩展模块错误码:  
对应错误码值如下表所示:

错误码	原因
1	没有连接扩展模块
11~20	扩展模块 N 没有回应
21~30	扩展模块 N 型号与配置不符
31~40	扩展模块 N 的电源出错

## (9) 模拟量输入

- 1, SM28.0~SM35.7: 模拟量输入通道 0~63 的数据错误标志位;  
当模拟量输入通道的数据出错时, 如一个 4~20mA 的输入信号, 若检测到输入信号远小于 4mA 或者远大于 20mA, 则对应的输入通道数据错误标志位 SM 置 ON。

## (10) $\pi$ (PI)值

- 1, 利用 SD10、SD11 组合成的 32 位数据寄存器来存放  $\pi$  (PI) 的数值。
- 2, 浮点数值=16#40490FDA。

### (11) 万年历

1, SD120~SD126: 万年历实时寄存器, 实时刷新; SD130~SD136: 万年历写入寄存器;

其中, 万年历星期实时寄存器 SD123 及万年历星期写入寄存器 SD133 的值 0~6 分别对应星期日至星期六; 万年历年份实时寄存器 SD126 及万年历年份写入寄存器 SD136 范围为 2000~2099;

SD	功能定义	读写权限
120	万年历秒针实时寄存器	RO
121	万年历分针实时寄存器	RO
122	万年历时针实时寄存器	RO
123	万年历星期实时寄存器	RO
124	万年历日期实时寄存器	RO
125	万年历月份实时寄存器	RO
126	万年历年份实时寄存器	RO

SD	功能定义	读写权限
130	万年历秒针写入寄存器	RW
131	万年历分针写入寄存器	RW
132	万年历时针写入寄存器	RW
133	万年历星期写入寄存器	RW
134	万年历日期写入寄存器	RW
135	万年历月份写入寄存器	RW
136	万年历年份写入寄存器	RW

2, SM53.7: 万年历正在变更标志位;

当正在进行万年历写入操作时, 万年历正在变更标志位 SM53.7 置 ON; 写入操作完成时, SM53.7 置 OFF。

### 2.3.2 RS485 通讯口类别

#### (1) 硬件设定

1, SD200~SD201: RS485\_1~RS485\_2 的应用层协议设定;

SD 值	应用层协议
16#0000	MODBUS-RTU
16#0001	MODBUS-ASCII

2, SD205~SD206: RS485\_1~RS485\_2 的波特率设定;

SD 值	波特率设定
16#0000	1200bps
16#0001	2400bps
16#0002	4800bps
16#0003	9600bps
16#0004	19200bps
16#0005	38400bps
16#0006	57600bps
16#0007	115200bps

3, SD210~SD211: COM1~COM2 的 MODBUS 站号设定;

站号设定范围为 16#0001~16#00F7;

4, SD215~SD216: COM1~COM2 的 MAC 层格式设定;

SD 值	MAC 层格式
16#0000	8,N,2
16#0001	8,E,1
16#0002	8,O,1
16#0003	7,N,2 (RTU 模式下不可设定)
16#0004	7,E,1 (RTU 模式下不可设定)
16#0005	7,O,1 (RTU 模式下不可设定)

5, SD220~SD221: COM1~COM2 的超时时间设定 (单位: ×10ms) ;

当 COM 做主站时, 即主动发送, 无论是 RS 指令还是 MODBUS 协议; 发送数据后, 如果需要等待从站回复数据, 那么将从发送数据完成时开始计时, 超出 SD 设定的时间范围时, 将会报错, 并停止等待接收数据;

## (2) 程序使用

程序使用相关 SM 和 SD 在通讯指令中有详细使用说明。以下表格是 SD245~SD246, 即做主站时通讯错误码:

SD 值	原因
1	非法功能, 不支持的功能码
2	非法数据地址, 对 MODBUS 地址不支持
3	非法数据, 对 MODBUS 地址操作数据不支持
4	从站设备故障, 在执行请求的操作中出现异常, 导致无法执行
5	请求的执行需要长时间, 设备正在执行, 主站应等待时间后发轮询报文询问
6	设备正在忙, 无法响应请求, 主站可等待时间后再继续发送请求
10	主站等待回复超时
12	主站接送数据个数超出范围
13	主站接送受到受干扰的异常数据
14	主站使用 MODBUS 协议通讯时, 接收到的 LRC/CRC 校验码不正确
15	主站使用 MODBUS-ASCII 通讯时接收到错误的结束符
16	总线有数据干扰或冲突, 数据发送失败



### 2.3.3 输入口类别

#### (1) 硬件设定

- 1, SD500~SD515: X0.0~X3.7 滤波时间设置 (单位: ms, 排序低的在低字节);  
通过该 SD 来设置输入端接收输入的过滤时间, 设置范围 0~50ms, 默认值为 20ms。
- 2, SD520~SD523: X0.0~X0.3 的输入功能类型;

输入口功能类型	SD 值			
	X0.0	X0.1	X0.2	X0.3
普通输入	16#0000	16#0000	16#0000	16#0000
不可中断单点高速计数	16#0001	16#0001	16#0001	16#0001
可中断单点高速计数 (HC0)	16#0002			
AB 相高速计数 1 倍频	16#0003	16#0003		
AB 相高速计数 2 倍频	16#0004	16#0004		
AB 相高速计数 4 倍频	16#0005	16#0005		

#### (2) 程序使用

程序使用相关 SM 和 SD 在通讯指令中有详细使用说明。

### 2.3.4 输出口类别

#### (1) 硬件设定

- 1, SM88~SM91: Y0~Y3 输出保持标志位;  
若 SM 置 ON, 对应的 Y 装置输出保持。

#### (2) 程序使用

程序使用相关 SM 和 SD 在通讯指令中有详细使用说明。

### 三 接点开关型指令

#### 3.1 基础接点开关

##### 3.1.1 LD（常开开关）

助记符	功能							
LD	常开开关，逻辑开始							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明：**

LD 指令用于左母线开始的常开接点或一个接点回路块开始的常开接点，它的作用是把当前内容保存，同时把取来的接点状态存入累加器内。

##### 3.1.2 LDI（常闭开关）

助记符	功能							
LDI	常闭开关，逻辑开始							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明：**

LDI 指令用于左母线开始的常闭接点或一个接点回路块开始的常闭接点，它的作用是把当前内容保存，同时把取来的接点状态存入累加器内。

##### 3.1.3 LDP（上升沿触发开关）

助记符	功能							
LDP	逻辑开始，上升沿触发开关							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明：**

LDP 指令用法与 LD 相同，但动作不同，它的作用是指当前内容保存，同时把取来的接点上升沿检出状态存入累加器内。

### 3.1.4 LDF (下降沿触发开关)

助记符	功能							
LDF	逻辑开始, 下降沿触发开关							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

LDF 指令用法与 LD 相同, 但动作不同, 它的作用是指当前内容保存, 同时把取来的接点下降沿检出状态存入累加器内。

### 3.1.5 AND (串联常开接点)

助记符	功能							
AND	串联常开接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

AND 指令用于常开接点的串联连接, 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, 并将结果存入累加器内。

### 3.1.6 ANI (串联常闭接点)

助记符	功能							
ANI	串联常闭接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

ANI 指令用于常闭接点的串联连接, 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, 并将结果存入累加器内。

### 3.1.7 ANDP (串联上升沿接点)

助记符	功能							
ANDP	串联上升沿接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

ANDP 指令用于接点上升沿检出的串联连接。

### 3.1.8 ANDF (串联下降沿接点)

助记符	功能							
ANDF	串联下降沿接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

ANDF 指令用于接点下降沿检出的串联连接。

### 3.1.9 OR (并联常开接点)

助记符	功能							
OR	并联常开接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

OR 指令用于常开接点的并联连接，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“或”(OR)的运算，并将结果存入累加器内。

### 3.1.10 ORI (并联常闭接点)

助记符	功能							
ORI	并联常闭接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

ORI 指令用于常闭接点的并联连接，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“或”(OR)的运算，并将结果存入累加器内。

### 3.1.11 ORP (并联上升沿接点)

助记符	功能							
ORP	并联上升沿接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

ORP 指令用于接点上升沿检出的并联连接。

### 3.1.12 ORF (并联下降沿接点)

助记符	功能							
ORF	并联下降沿接点							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
	■	■	■	■	■	■	■	■

**指令说明:**

ORF 指令用于接点下降沿检出的并联连接。

### 3.2 逻辑接点开关

#### 3.2.1 LD&、LD|、LD^ (开始接点, 逻辑运算)

类型	助记符	P	逻辑块开始, 接点型态逻辑运算																				
字	LD#		In1(W)										In2(W)										
双字	DLD#		In1(D)										In2(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■

**指令说明:**

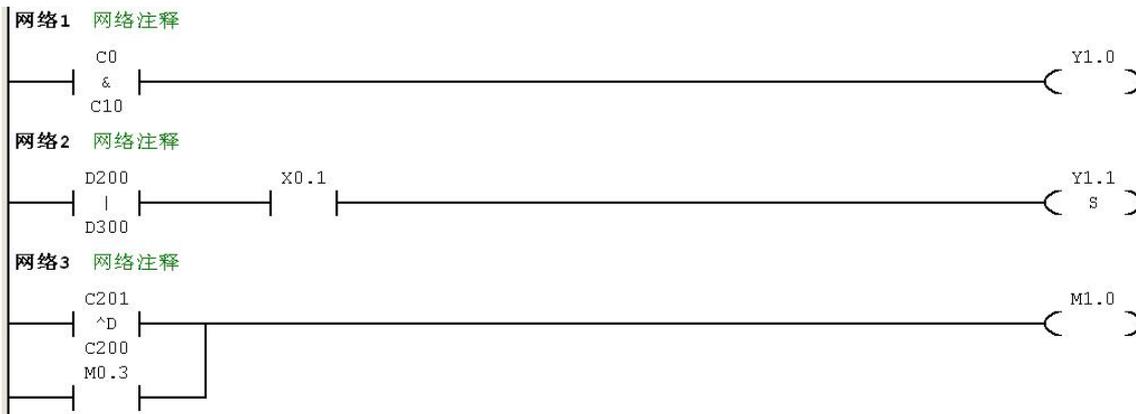
- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作逻辑与 (LD&), 或 (LD|), 异或 (LD^ ) 运算, 若结果不为 0, 该接点导通, 若运算结果为 0, 则该接点不导通。

16-bit 指令	32-bit 指令	导通条件	非导通条件
LD&	DLD&	In1 & In2 ≠ 0	In1 & In2 = 0
LD	DLD	In1   In2 ≠ 0	In1   In2 = 0
LD^	DLD^	In1 ^ In2 ≠ 0	In1 ^ In2 = 0

**程序范例:**

**例:**

- C0 与 C10 的内容做逻辑的‘与’ (AND) 运算不等于 0 时, Y1.0=On。  
 D200 与 D300 的内容做逻辑的‘或’ (OR) 运算不等于 0 时, 而且 X0.1=On 的时候, Y1.1=On 并保持住。  
 C201 与 C200 的内容做逻辑的‘异或’ (XOR) 运算不等于 0 时或是 M0.3=On 的时候, M1.0=On。



### 3.2.2 AND&、AND|、AND^ (串联接点, 逻辑运算)

类型	助记符	P	AND 接点型态, 逻辑运算																							
字	AND#		In1(W)												In2(W)											
双字	DAND#		In1(D)												In2(D)											
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/	E	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/		
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	
In2										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	

**指令说明:**

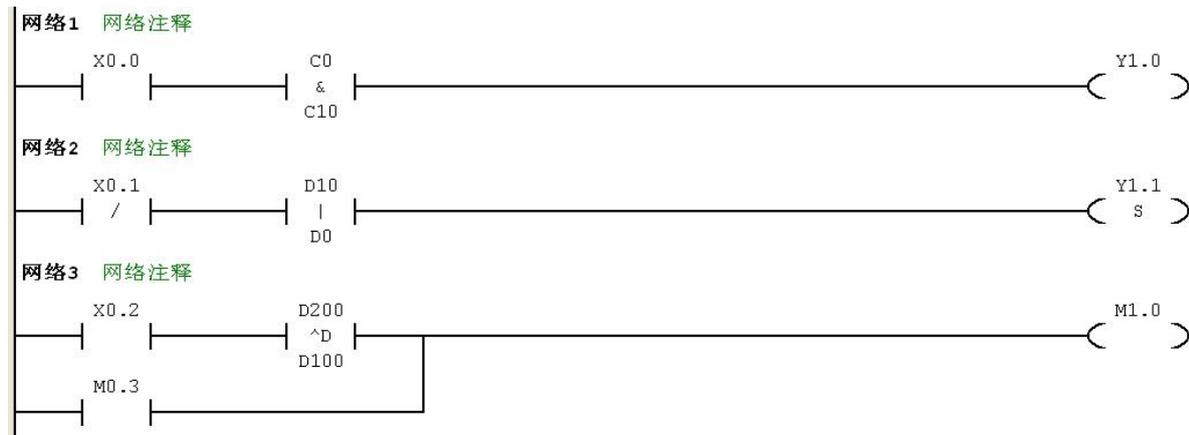
- 1, In1: 数据来原装置 1; In2: 数据来原装置 2;
- 2, In1 与 In2 的内容作逻辑与 (AND&), 或 (AND|), 异或 (AND^ ) 运算, 若结果不为 0, 该接点导通, 若运算结果为 0, 则该接点不导通。

16-bit 指令	32-bit 指令	导通条件	非导通条件
AND&	DAND&	In1 & In2 ≠ 0	In1 & In2 = 0
AND	DAND	In1   In2 ≠ 0	In1   In2 = 0
AND^	DAND^	In1 ^ In2 ≠ 0	In1 ^ In2 = 0

**程序范例:**

**例:**

- 当 X0.0=On 时且 C0 与 C10 的内容做逻辑的‘与’ (AND) 运算不等于 0 时, Y1.0=On。
- 当 X0.1=Off 时且 D10 与 D0 的内容做逻辑的‘或’ (OR) 运算不等于 0 时, Y1.1=On 并保持住。
- 当 X0.2=On 时且 32 位寄存器 D200(D201)与 32 位寄存器 D100(D101)的内容做逻辑的‘异或’ (XOR) 运算不等于 0 时或者 M0.3=On 的时候, M1.0=On。



### 3.2.3 OR&、OR|、OR^（并连接点，逻辑运算）

类型	助记符	P	OR 接点型态, 逻辑运算																						
字	OR#		In1(W)										In2(W)												
双字	DOR#		In1(D)										In2(D)												
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■

**指令说明:**

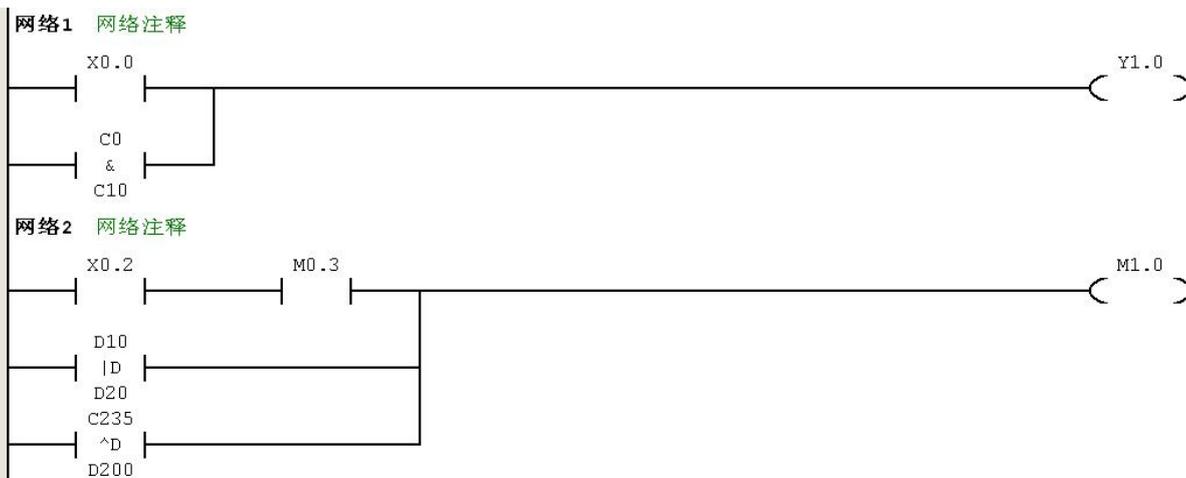
- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作逻辑与 (OR&) , 或 (OR|) , 异或 (OR^ ) 运算, 若结果不为 0, 该接点导通, 若运算结果为 0, 则该接点不导通。

16-bit 指令	32-bit 指令	导通条件	非导通条件
OR&	DOR&	In1 & In2 ≠ 0	In1 & In2 = 0
OR	DOR	In1   In2 ≠ 0	In1   In2 = 0
OR^	DOR^	In1 ^ In2 ≠ 0	In1 ^ In2 = 0

**程序范例:**

**例:**

- 当 X0.0=On 时或 C0 与 C10 的内容做逻辑的‘与’ (AND) 运算不等于 0 时, Y1.0=On。
- 当 X0.2 及 M0.3 都等于 On 的时候, 或者是 32-bit 寄存器 D10(D11)与 32 位寄存器 D20(D21)的内容做逻辑的‘或’ (OR)
- 运算不等于 0 时, 或者是 32 位计数器 C235 与 32 位寄存器 D200(D201)的内容做逻辑的‘异或’ (XOR) 运算不等于 0 时, M1.0=On。



### 3.2.4 LD=、LD<、LD>、LD<>、LD<=、LD>= (开始接点, 数值比较运算)

类型	助记符	P	LD 接点型态, 数值比较运算																							
字	LD*		In1(W)												In2(W)											
双字	DLD*		In1(D)												In2(D)											
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/		
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	
In2										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	

**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作比较运算, 根据指令 (=, <, >, <>, <=, >=), 如果 In1 符合对 In2 的比较, 则该接点导通, 如果不符合, 则该接点不导通。

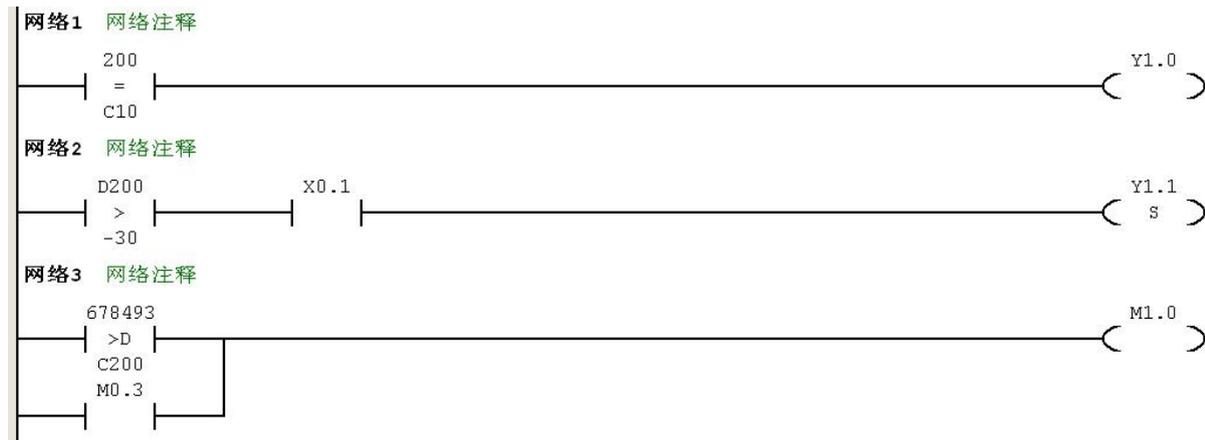
**程序范例:**

**例:**

C10 的内容等于 200 时, Y1.0=On。

当 D200 的内容大于-30, 而且 X0.1=On 的时候, Y1.1=On 并保持住。

C200 的内容小于 678,493 或者是 M0.3=On 的时候, M1.0=On。



### 3.2.5 AND=、AND<、AND>、AND<>、AND<=、AND>=（串联接点，数值比较运算）

类型	助记符	P	AND 接点型态，数值比较运算																						
字	AND*		In1(W)										In2(W)												
双字	DAND*		In1(D)										In2(D)												
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L		X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■

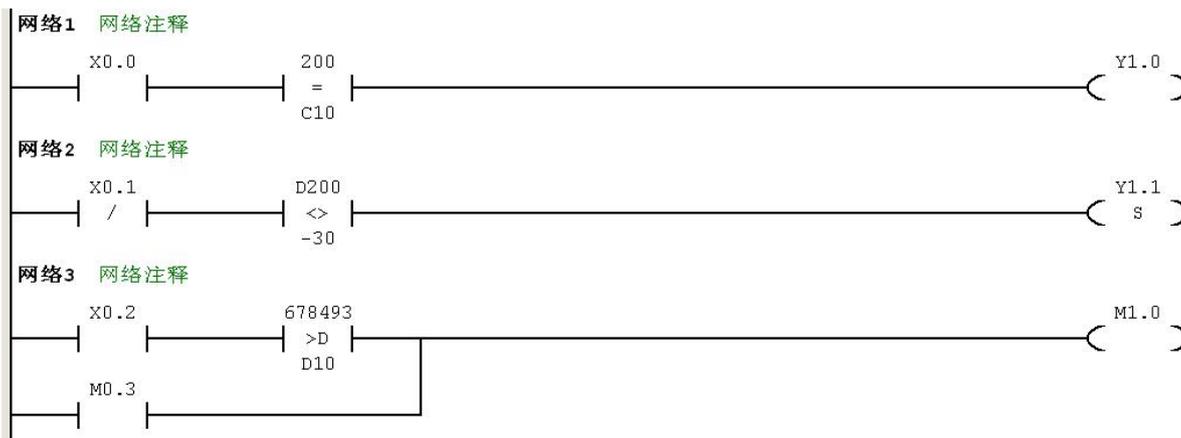
**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作比较运算, 根据指令 (=, <, >, <>, <=, >=), 如果 In1 符合对 In2 的比较, 则该接点导通, 如果不符合, 则该接点不导通。

**程序范例:**

**例:**

- 当 X0.0=On 时且 C10 的现在值又等于 200 时, Y1.0=On。
- 当 X0.1=Off 而寄存器 D200 的内容又不等于-30 的时候, Y1.1=On 并保持住。
- 当 X0.2=On 而且 32 位寄存器 D10(D11)的内容又小于 678,493 的时候或 M0.3=On 时, M1.0=On。



### 3.2.6 OR=、OR<、OR>、OR<>、OR<=、OR>=（并联接点，数值比较运算）

类型	助记符	P	OR 接点型态, 数值比较运算																				
字	OR*		In1(W)										In2(W)										
双字	DOR*		In1(D)										In2(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■

**指令说明:**

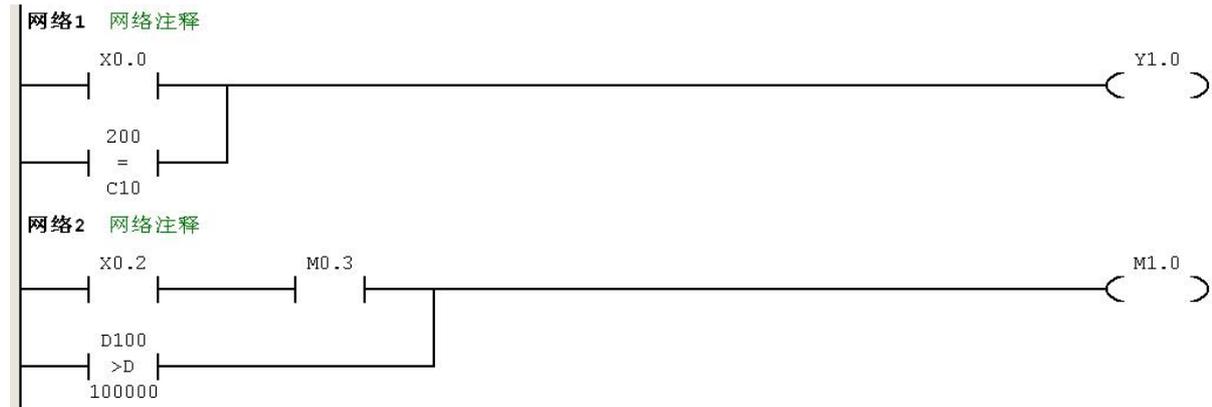
- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作比较运算, 根据指令 (=, <, >, <>, <=, >=), 如果 In1 符合对 In2 的比较, 则该接点导通, 如果不符合, 则该接点不导通。

**程序范例:**

**例:**

当 X0.0=On 时, 或者是 C10 的现在值等于 200 时, Y1.0=On。

当 X0.2 及 M0.3 都等于 On 的时候, 或者是 32 位寄存器 D100(D101)的内容大于或等于 100,000 时, M1.0=On。



### 3.3 字装置的位接点开关

#### 3.3.1 BLD (常开开始接点, 取字/双字装置位)

类型	助记符	P	逻辑块常开接点开始, 接点型态字符取装置位																					
字	BLD		In1(W)										In2(W)											
双字	DBLD		In1(D)										In2(W)											
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**指令说明:**

- 1, In1: 数据目的装置; In2: 欲取装置之位;
- 2, 一个逻辑块的常开接点开始, 取指定装置位的值作为该接点的逻辑状态;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

**例:**

加载 D0 Bit3 的常开接点来驱动 Y0.0 线圈。





### 3.3.2 BLDI (常闭开始接点, 取字/双字装置位)

类型	助记符	P	逻辑块常闭开始, 接点型态字符取装置位																							
字	BLDI		In1(W)												In2(W)											
双字	DBLDI		In1(D)												In2(W)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		

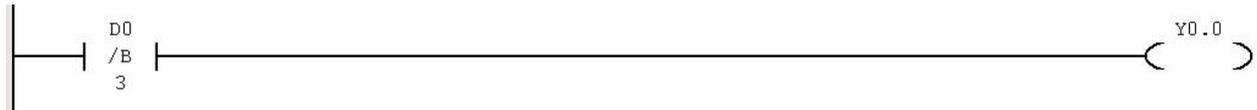
**指令说明:**

- 1, In1: 数据目的装置; In2: 欲取装置之位;
- 2, 一个逻辑块的常闭接点开始, 取指定装置位的值作为该接点的逻辑状态;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

加载 D0 Bit3 的常闭接点来驱动 Y0.0 线圈。



### 3.3.3 BAND（常开串联接点，取字/双字装置位）

类型	助记符	P	AND 接点型态字符取装置位																				
字	BAND		In1(W)										In2(W)										
双字	DBAND		In1(D)										In2(W)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■

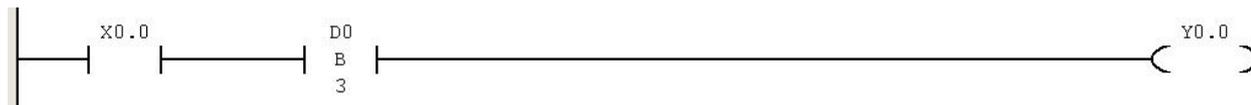
**指令说明:**

- 1, In1: 数据目的装置; In2: 欲取装置之位;
- 2, 取指定装置位的值作为该接点的逻辑状态, 然后与 BAND 之前的逻辑状态作与运算, 作为之后的逻辑状态;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

**例:**

串联 D0 Bit3 的常开接点来驱动 Y0.0 线圈。





### 3.3.4 BANI (常闭串联接点, 取字/双字装置位)

类型	助记符	P	ANI 接点型态字符取装置位																							
字	BANI		In1(W)												In2(W)											
双字	DBANI		In1(D)												In2(W)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		

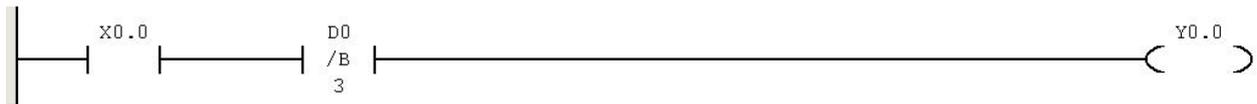
**指令说明:**

- 1, In1: 数据目的装置; In2: 欲取装置之位;
- 2, 取指定装置位的值取反逻辑后作为该接点的逻辑状态, 然后与 BANI 之前的逻辑状态作与运算, 作为之后的逻辑状态;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

串联 D0 Bit3 的常闭接点来驱动 Y0.0 线圈。



### 3.3.5 BOR (常开并联接点, 取字/双字装置位)

类型	助记符	P	OR 接点型态字符取装置位																				
字	BOR		In1(W)										In2(W)										
双字	DBOR		In1(D)										In2(W)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■

**指令说明:**

- 1, In1: 数据目的装置; In2: 欲取装置之位;
- 2, 取指定装置位的值作为该接点的逻辑状态, 然后与 BOR 之前的逻辑状态作或运算, 作为之后的逻辑状态;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

并联 D0 Bit3 的常开接点来驱动 Y0.0 线圈。





### 3.3.6 BORI (常闭并联接点, 取字/双字装置位)

类型	助记符	P	ORI 接点型态字符取装置位																							
字	BORI		In1(W)												In2(W)											
双字	DBORI		In1(D)												In2(W)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		

**指令说明:**

- 1, In1: 数据目的装置; In2: 欲取装置之位;
- 2, 取指定装置位的值取反逻辑后作为该接点的逻辑状态, 然后与 BORI 之前的逻辑状态作或运算, 作为之后的逻辑状态;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

并联 D0 Bit3 的常闭接点来驱动 Y0.0 线圈。



### 3.4 浮点数比较接点开关

#### 3.4.1 FLD=、FLD<、FLD>、FLD<>、FLD>=、FLD<=（开始接点，浮点数比较运算）

类型	助记符	P	LD 接点型态，浮点数数值比较运算																				
字																							
双字	FLD*		In1(D)										In2(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

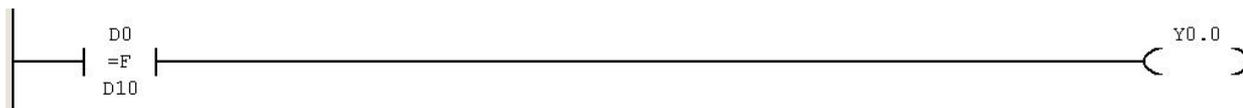
**指令说明：**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作二进制浮点数比较运算，根据指令 (=, <, >, <>, <=, >=)，如果 In1 符合对 In2 的比较，则该接点导通，如果不符合，则该接点不导通。

**程序范例：**

**例：**

当寄存器 D0(D1)的浮点数内容等于寄存器 D10(D11)的浮点数内容，则 Y0.0=On。





### 3.4.2 FAND=、FAND<、FAND>、FAND<>、FAND>=、FAND<=（串联接点，浮点数比较运算）

类型	助记符	P	AND 接点型态，浮点数数值比较运算																							
字																										
双字	FAND*		In1(D)												In2(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			

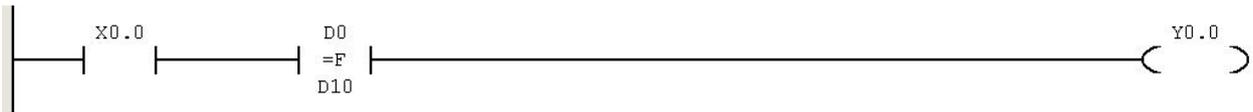
**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作二进制浮点数比较运算, 根据指令 (=, <, >, <>, <=, >=), 如果 In1 符合对 In2 的比较, 则该接点导通, 如果不符合, 则该接点不导通。

**程序范例:**

**例:**

当 X0.0=On, 并且寄存器 D0(D1) 的浮点数内容等于寄存器 D10(D11) 的浮点数内容时, Y0.0=On。



### 3.4.3 FOR=、FOR<、FOR>、FOR<>、FOR>=、FOR< (并联接点, 浮点数比较运算)

类型	助记符	P	OR 接点型态, 浮点数数值比较运算																				
字																							
双字	FOR*		In1(D)										In2(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

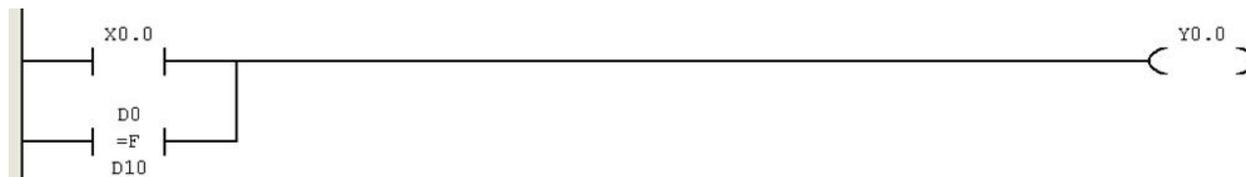
**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2;
- 2, In1 与 In2 的内容作二进制浮点数比较运算, 根据指令 (=, <, >, <>, <=, >=), 如果 In1 符合对 In2 的比较, 则该接点导通, 如果不符合, 则该接点不导通。

**程序范例:**

**例:**

当 X0.0=On, 或者寄存器 D0(D1) 的浮点数内容等于寄存器 D10(D11) 的浮点数内容时, Y0.0=On。





## 3.5 多接点开关连接指令

### 3.5.1 ANB (块串联)

助记符	功能
ANB	块串联
操作数	无

**指令说明:**

ANB 是将前一保存的逻辑结果与目前累加器的内容作“与” (AND) 的运算。

### 3.5.2 ORB (块并联)

助记符	功能
ORB	块并联
操作数	无

**指令说明:**

ORB 是将前一保存的逻辑结果与目前累加器的内容作“或” (OR) 的运算。

### 3.5.3 MPS (压入堆栈)

助记符	功能
MPS	压入堆栈
操作数	无

**指令说明:**

将目前累加器的内容存入堆栈。(堆栈指针加一)

### 3.5.4 MRD (读取堆栈)

助记符	功能
MRD	读取堆栈
操作数	无

**指令说明:**

读取堆栈内容存入累加器。(堆栈指针不动)

### 3.5.5 MPP (弹出堆栈)

助记符	功能
MPP	弹出堆栈
操作数	无

**指令说明:**

自堆栈取回前一保存的逻辑运算结果, 存入累加器。(堆栈指针减一)

### 3.5.6 INV (反转状态)

助记符	功能
INV	反转状态
操作数	无

**指令说明:**

将 INV 指令之前的逻辑运算结果反相存入累加器内。

### 3.5.7 NOP (空指令)

助记符	功能
NOP	空指令
操作数	无

**指令说明:**

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，使用时机如下：想要删除某一指令，而又不想改变程序长度，则可以 NOP 指令取代。



## 四 位输出型指令

### 4.1.1 OUT (输出线圈)

助记符	功能							
OUT	输出线圈							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
		■	■	■				■

#### 指令说明:

- 1, 将 OUT 指令之前的逻辑运算结果输出至指定的组件;
- 2, 线圈接点动作:

运算结果	OUT 指令		
	线圈	接点	
		常开接点	常闭接点
FALSE	OFF	不导通	导通
TRUE	ON	导通	不导通

### 4.1.2 SET (对位置位)

助记符	功能							
SET	对位置位							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
		■	■	■				■

#### 指令说明:

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。

### 4.1.3 PLS (上升沿检测置位)

助记符	功能							
PLS	上升沿检测置位							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
		■	■	■				■

#### 指令说明:

上升沿检出指令, 当连接能流 OFF 到 ON (上升沿触发) 时, PLS 指令被执行, 对应线圈输出一个扫描周期。

#### 4.1.4 PLF（下降沿检测置位）

助记符	功能							
PLF	下降沿检测置位							
操作数	位装置							
	X	Y	SM	M	T	C	HC	L/EL
		■	■	■				■

**指令说明：**

下降沿检出指令，当连接能流 ON 到 OFF（下降沿触发）时，PLF 指令被执行，对应线圈输出一个扫描周期。



### 4.1.5 BOUT (字符装置位输出)

类型	助记符	P	字符装置位输出																							
字	BOUT		In1(W)												InOut1(W)											
双字	DBOUT		In1(W)												InOut1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
InOut1										■	■	■	■	□	□		■	■	■	■				■		

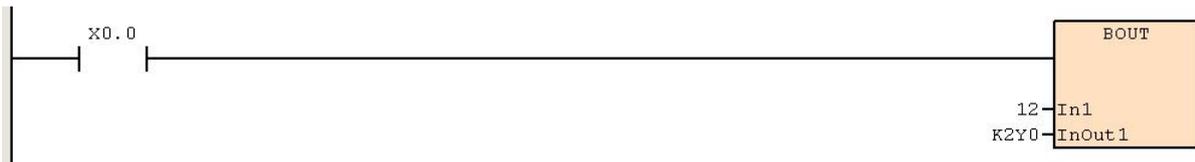
**指令说明:**

- 1, In1: 欲输出之位; InOut1: 数据目的装置;
- 2, 将 BOUT 指令之前的逻辑运算结果输出至指定的位;
- 3, In1 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

当 X0.0=On 时, 输出 Y1.4。



### 4.1.6 BSET (字符装置位置位)

类型	助记符	P	字符装置位置位																				
字	BSET		In1(W)										InOut1(W)										
双字	DBSET		In1(W)										InOut1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■
InOut1										■	■	■	■	□	□		■	■	■	■			■

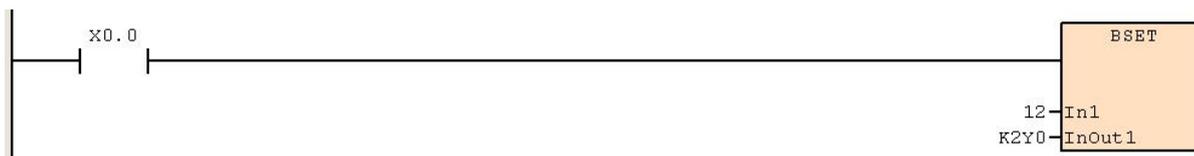
**指令说明:**

- 1, In1: 欲置位之位; InOut1: 数据目的装置;
- 2, 若 BSET 指令之前的运算结果为 ON, 则置位对应的位;
- 3, In1 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

当 X0.0=On 时, 置位 Y1.4 并保持住。





### 4.1.7 BRST (字符装置位清 0)

类型	助记符	P	字符装置位清 0																							
字	BRST		In1(W)												InOut1(W)											
双字	DBRST		In1(W)												InOut1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
InOut1										■	■	■	■	□	□		■	■	■	■				■		

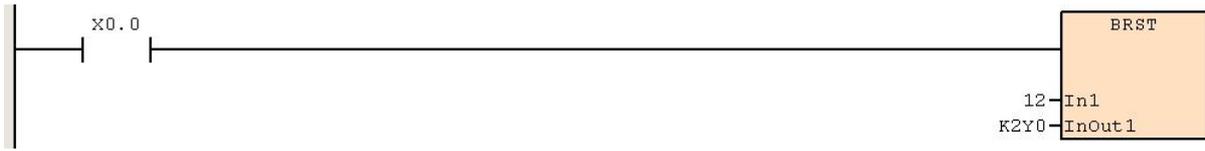
**指令说明:**

- 1, In1: 欲清 0 之位; InOut1: 数据目的装置;
- 2, 若 BRST 指令之前的运算结果为 ON, 则对应的位清 0;
- 3, In2 的取值范围, 16 位指令时为 0~15, 32 位指令时为 0~31, 超出不执行。

**程序范例:**

例:

当 X0.0=On 时, 将 Y1.4 清 0, 即 Y1.4=off。



### 4.1.8 ALT (On/Off 交替输出)

类型	助记符	P	On/Off 交替输出																						
字	ALT	■	InOut1(b)																						
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	AI	A	D	S	E	N	O	L/	
InOut1		■	■	■				■																	

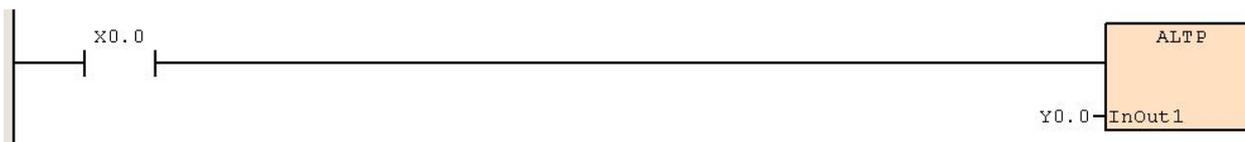
**指令说明:**

- 1, InOut1: 目的装置;
- 2, ALT 指令执行时, InOut1 On/Off 交替输出。

**程序范例:**

**例:**

当第一次 X0.0 从 Off->On 时, Y0.0 = On; On->Off 时, Y0.0= On 保持不变。第二次 X0.0 从 Off->On 时, Y0.0 = Off; On->Off 时, Y0.0= Off 保持不变。





## 五 基础系统型指令

### 5.1.1 TMR (启动定时器)

类型	助记符	P	启动定时器																						
字	TMR		In1(W)											InOut1(b)											
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■		
InOut1				■																					

#### 参数说明:

- 1, In1: 指定的定时值数值, 单位根据对应的 T 装置而定;
- 2, InOut1: 选择的 T 定时装置, 不可使用变址装置;

#### 指令执行注意点:

- 1, 该指令需要能流持续, 当能流断开时, 对应的 T 装置将停止计时, 同时非累积型 T 装置, 将会清空当前的计时值;
- 2, TMR 和 STMR 指令不可指定同一个 T 装置, 后指定的不执行;

#### 指令功能说明:

- 1, 当 TMR 指令执行时, 其所指定的定时器线圈受电, 定时器开始计时, 当到达所指定的定时值 (计时值 $\geq$ 设定值), 其接点动作如下:

NO(Normal Open) 接点	导通
NC(Normal Close) 接点	不导通

- 2, 定时器操作数 T 装置的单位及属性参考规格表。

定时器 T	100ms 一般用	T0~T199, 200 点
	100ms 累计型	T250~T255, 6 点
	10ms 一般用	T200~T239, 40 点
	10ms 累计型	T244~T249, 6 点
	1ms 一般用	T240~T241, 2 点
	1ms 累计型	T242~T243, 2 点

#### 定时器的功能:

- 1) 定时器是以 1ms、10ms、100ms 为一个计时单位, 计时方式采用上数计时, 当定时器现在值大于等于设置值时, 输出线圈导通, 设置值可为常数, 也可使用数据寄存器当成设置值。
- 2) 定时器在 TMR 指令执行时开始计时, 每次扫描到该指令执行时判断计时是否到达, 若计时到达, 则输出线圈导通。
- 3) 定时器的时机设置时间 = 计时单位\*设置值。

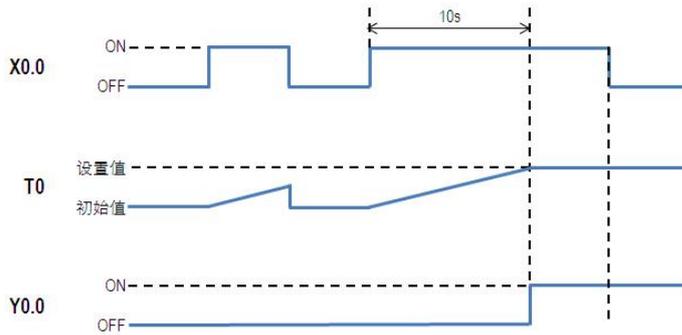
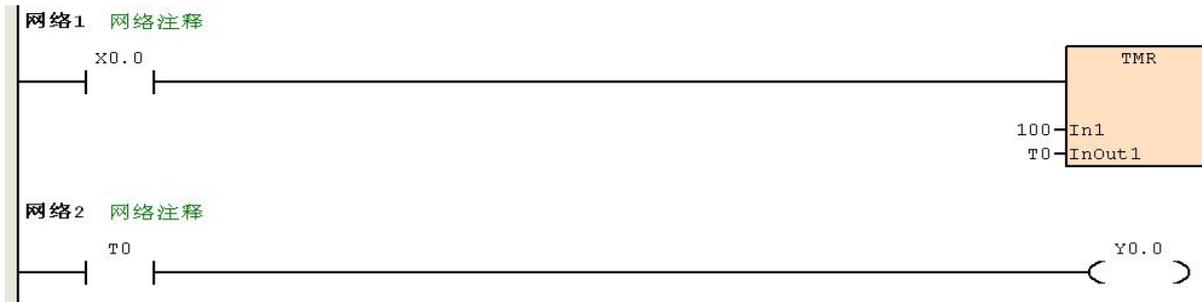
依其性质可区分为下列两种:

- 1) 一般用定时器: 当执行到 TMR 指令时, 能流状态为 on, 则保持计时; 若为 off, 停止计时并同时 will 将计时数值清零。
- 2) 累计型定时器: 当执行到 TMR 指令时, 能流状态为 on, 则保持计时; 若为 off, 停止计时但保持当前计时数值不变。

程序范例：

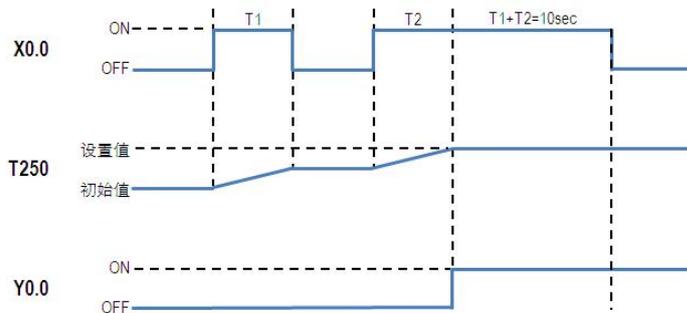
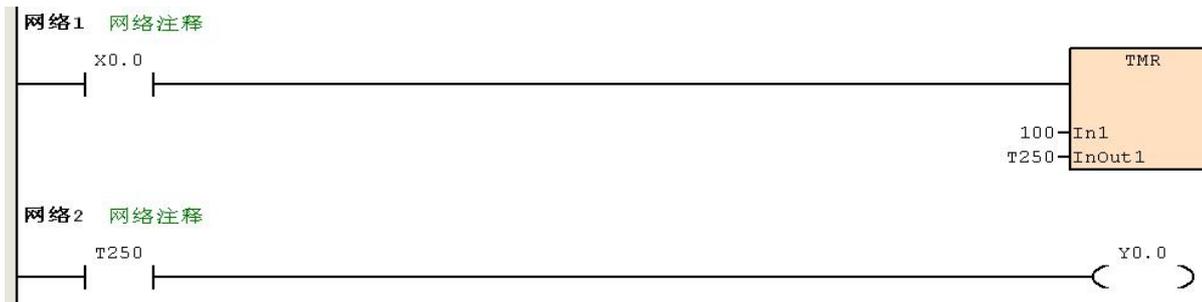
例 1：一般用定时器

当 X0.0=On 时，定时器 T0 的现在值以 100ms 上数计时，当定时器现在值=设置值 100 时，输出线圈 T0=On，并保持住。  
 当 X0.0=Off 时，定时器 T0 的现在值不变，输出线圈 T0=On 仍保持住。  
 当停电时，定时器 T0 的现在值清为 0，输出线圈 T0 变为 Off。



例 2：累计型定时器

- 当 X0.0=On 时，定时器 T250 的现在值以 100ms 上数计时，当定时器现在值=设置值 100 时，输出线圈 T250=On。
- 当计时中若 X0.0=Off 或停电时，定时器 T250 暂停计时，现在值不变，待 X0.0 再 On 时，继续计时，其现在值往上累加直到定时器现在值=设置值 100 时，输出线圈 T250=On。





### 5.1.2 CNT (启动普通计数器)

类型	助记符	P	启动普通计数器																							
字	CNT		In1(W)												InOut1(b)											
双字	DCNT		In1(D)												InOut1(b)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
InOut1						□																				

#### 参数说明:

- 1, In1: 指定的计数数值;
- 2, InOut1: 指定的 C 计数器, 不可使用变址装置;

#### 指令执行注意点:

- 1, 若要重新计数或作清除的动作, 请利用 RST 指令; 若使用 MOV 指令或通讯方式将一个大于设置值的数值传送到 C 现在值寄存器时, 在下次执行 CNT 时, C 计数器接点即变成 On, 同时现在值内容变成与设置值相同。
- 2, DCNT 为 32 位计数器 C200~C255; 其中使用 C200~C241 时, 计数动作同 C0~C179 一致; 使用 C242~C255 时, 计数动作同 C180~C199 一致, 对应的设定模式为 SM45.0~SM46.5。
- 3, 当主机 stop 后, 即相当于能流断开。

#### 指令功能说明:

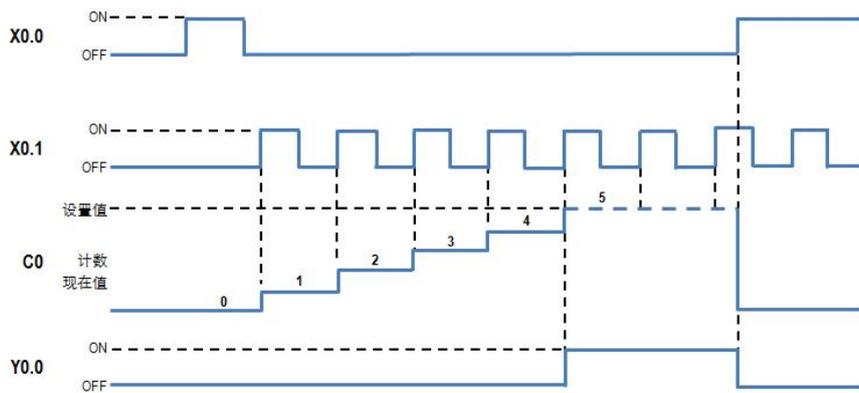
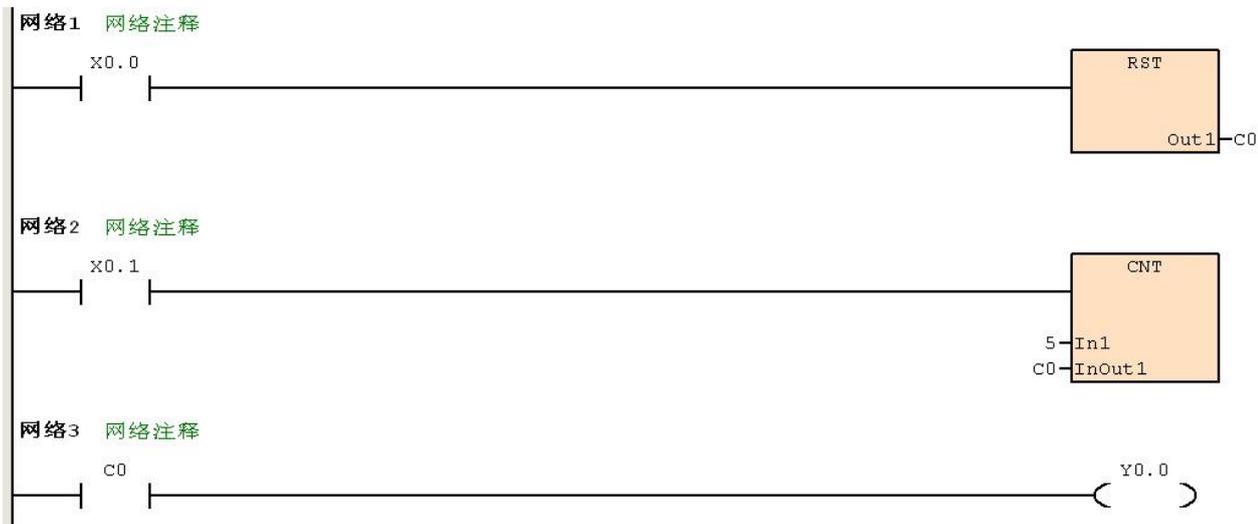
- 1, CNT 为 16 位计数器 C0~C199。其中 C0~C179 每一次扫描到 CNT 指令由 Off->On 时, 当计数到达所指定的计数值 (计数值 = 设定值), 其 C 接点接通, 当计数到达之后, 若再有计数脉冲输入, 其接点及计数值保持不变;
- 2, 当使用 C180~C199, 将依据 SM40.0~SM42.3 的设定模式, 计数器将执行单次上数或下数的动作; 上数时到达计数设定值, 对应的 C 接点将置 ON, 同时可继续正常计数; 下数时, 当计数当前值到达计数设定值后, 再一次计数时, 对应的 C 接点将会置 OFF, 同时可继续正常计数。

项目	16 位计数器		32 位计数器	
	C0~C179	C180~C199	C200~C241	C242~C255
计数方向	上数	上、下数	上数	上、下数
设置值的范围	-32768~32767		2147483648~2147483647	
设置值的指定	常数或各字寄存器		常数或各双字寄存器	
现在值的变化	计数到达设置值 就不再计数	继续计数	计数到达设置值 就不再计数	继续计数
输出接点	计数值到达设置值, 接点导通并保持 (下数越过设置值, 接点断开)			
复位动作	RST 指令被执行时, 现在值归零, 接点被复位成 Off			
接点动作	在执行 CNT 时动作		在执行 DCNT 时动作	

程序范例：

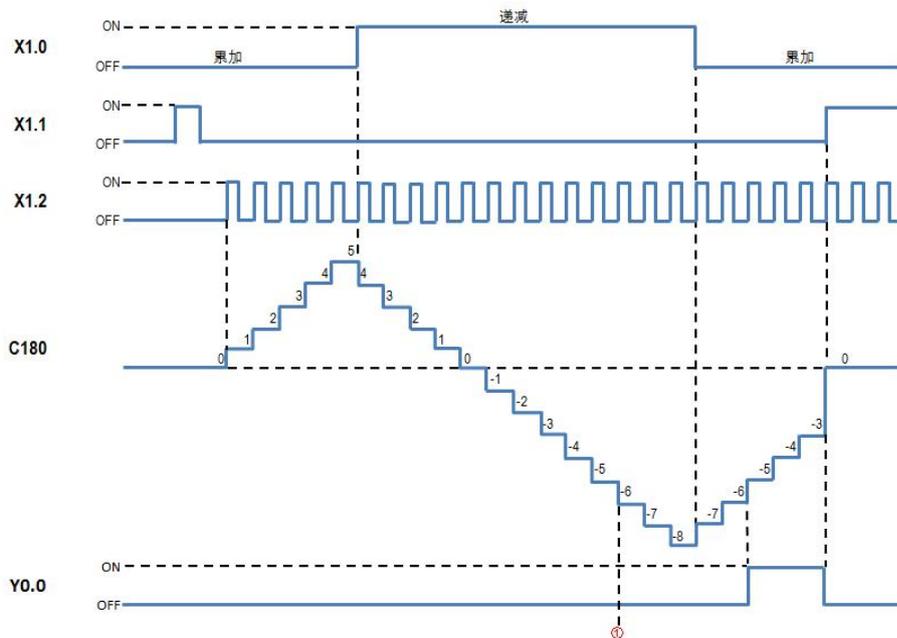
例 1：16 位上数计数器

- 1, 当 X0.0=On 时 RST 指令被执行, C0 的现在值归零, 输出接点被复位为 Off。
- 2, 当 X0.1 由 Off→On 时, 计数器的现在值将执行上数 (加一) 的动作。
- 3, 当计数器 C0 计数到达设置值 5 时, C0 现在值 = 设置值=5, 接点导通。之后的 X0.1 触发信号 C0 完全不接受, C0 现在值保持在 5 处。



**例 2: 16 位上、下计数器**

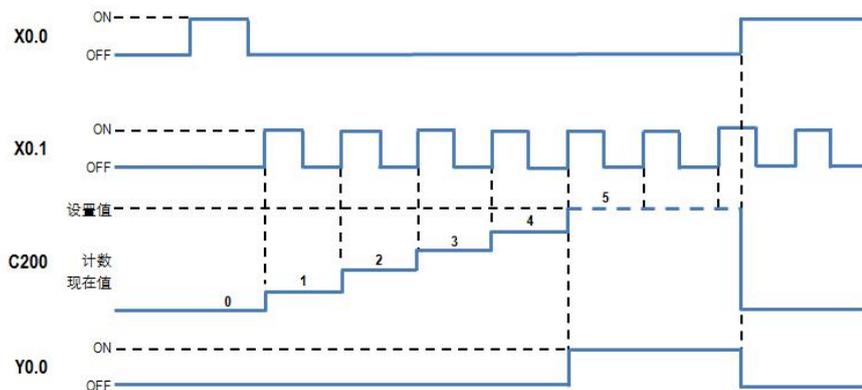
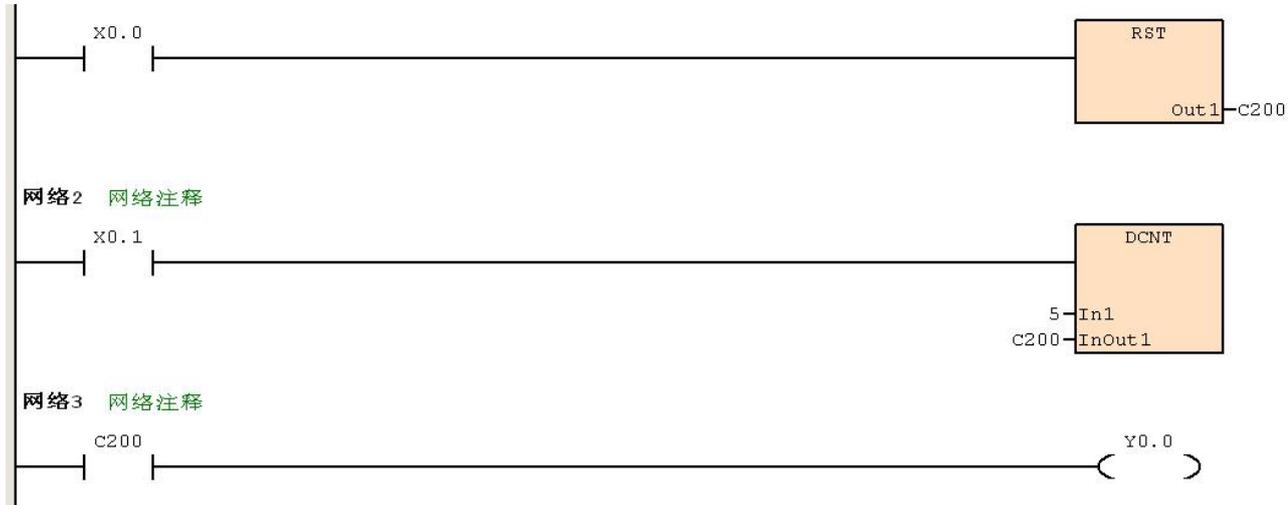
- 1, 由 X1.0 驱动 SM40.0 来决定 C180 为加算或减算。
- 2, 当 X1.1 由 Off→On 时, RST 指令执行, C180 的现在值被清为 0, 且接点变为 Off。
- 3, 当 X1.2 由 Off→On 时, 计数器的现在值将执行上数 (加一) 的动作或下数 (减一) 的动作。
- 4, 当计数器 C180 的现在值从-6→-5 变化时, C180 接点由 Off→On。当计数器 C180 的现在值从-5→-6 变化时, C180 接点由 On→Off。



①: 当 Y0.0 原状态为 ON 时, 在此处就会变为 OFF

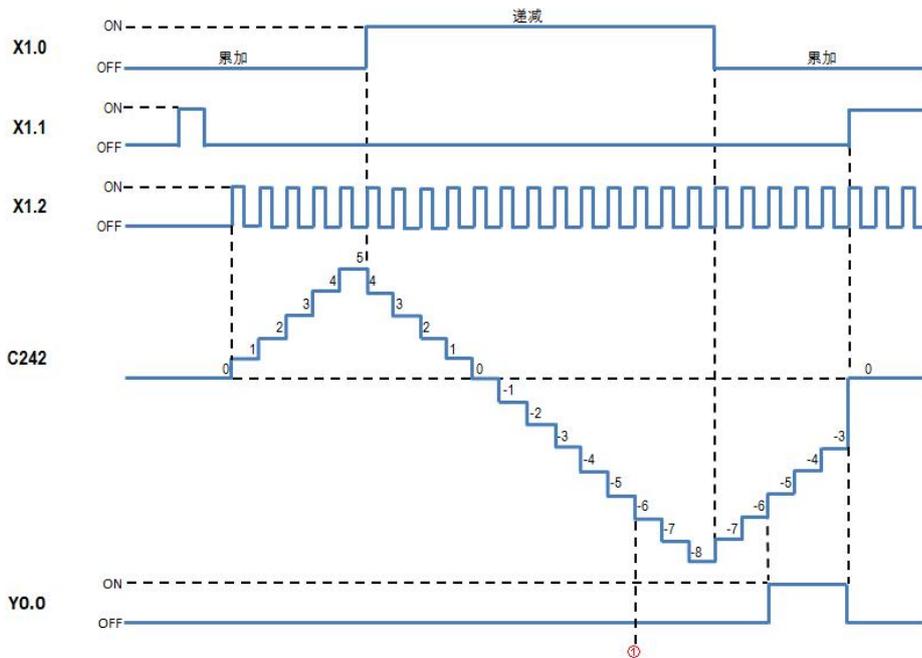
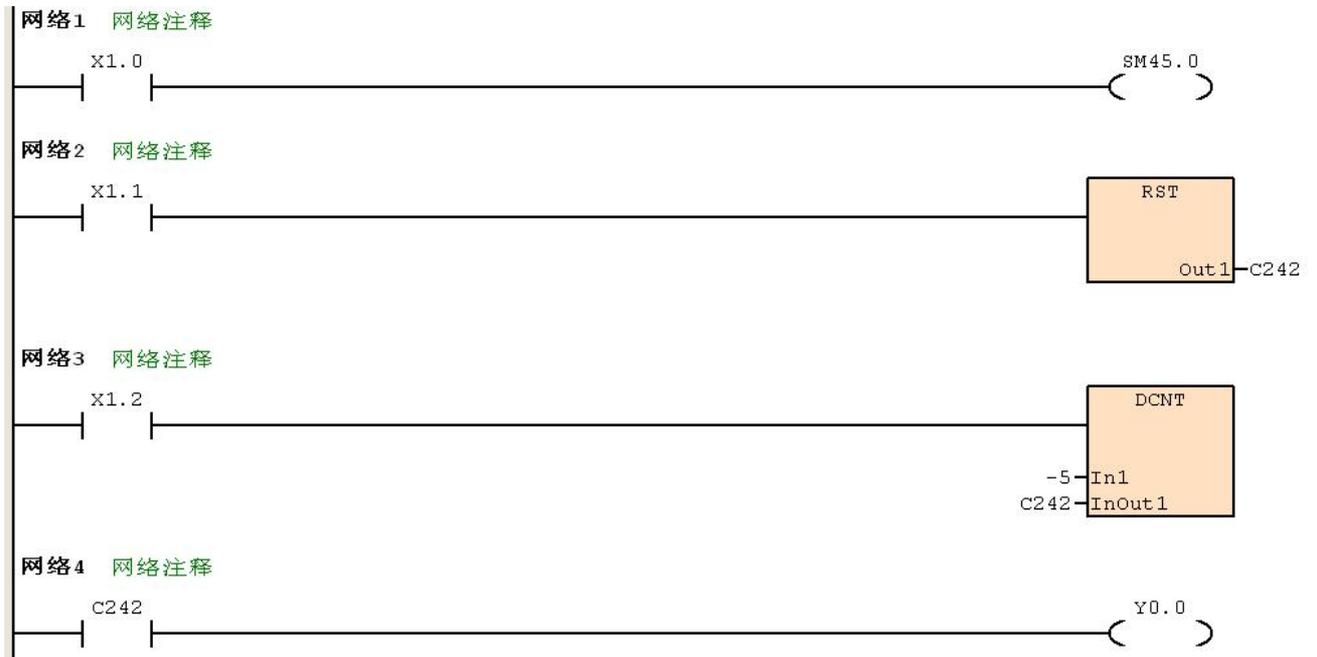
例 3: 32 位上数计数器

- 1, 当 X0.0=On 时 RST 指令被执行, C200 的现在值归零, 输出接点被复位为 Off。
- 2, 当 X0.1 由 Off→On 时, 计数器的现在值将执行上数 (加一) 的动作。
- 3, 当计数器 C200 计数到达设置值 5 时, C200 接点导通, C200 现在值 = 设置值=5。之后的 X0.1 触发信号 C200 完全不接受, C200 现在值保持在 5 处。



**例 4: 32 位上、下计数器**

- 1, 由 X1.0 驱动 SM45.0 来决定 C242 为加算或减算。
- 2, 当 X1.1 由 Off→On 时, RST 指令执行, C242 的现在值被清为 0, 且接点变为 Off。
- 3, 当 X1.2 由 Off→On 时, 计数器的现在值将执行上数 (加一) 的动作或下数 (减一) 的动作。
- 4, 当计数器 C242 的现在值从-6→-5 变化时, C242 接点由 Off→On。当计数器 C242 的现在值从-5→-6 变化时, C242 接点由 On→Off。



①: 当 Y0.0 原状态为 ON 时, 在此处就会变为 OFF

补充对应 SM 说明:

SM	点数	对应功能	读写
40	0	C180 的上/下计数方式选择开关	RW
	1	C181 的上/下计数方式选择开关	RW
	2	C182 的上/下计数方式选择开关	RW
	3	C183 的上/下计数方式选择开关	RW
	4	C184 的上/下计数方式选择开关	RW
	5	C185 的上/下计数方式选择开关	RW
	6	C186 的上/下计数方式选择开关	RW
	7	C187 的上/下计数方式选择开关	RW
41	0	C188 的上/下计数方式选择开关	RW
	1	C189 的上/下计数方式选择开关	RW
	2	C190 的上/下计数方式选择开关	RW
	3	C191 的上/下计数方式选择开关	RW
	4	C192 的上/下计数方式选择开关	RW
	5	C193 的上/下计数方式选择开关	RW
	6	C194 的上/下计数方式选择开关	RW
	7	C195 的上/下计数方式选择开关	RW
42	0	C196 的上/下计数方式选择开关	RW
	1	C197 的上/下计数方式选择开关	RW
	2	C198 的上/下计数方式选择开关	RW
	3	C199 的上/下计数方式选择开关	RW
45	0	C242 的上/下计数方式选择开关	RW
	1	C243 的上/下计数方式选择开关	RW
	2	C244 的上/下计数方式选择开关	RW
	3	C245 的上/下计数方式选择开关	RW
	4	C246 的上/下计数方式选择开关	RW
	5	C247 的上/下计数方式选择开关	RW
	6	C248 的上/下计数方式选择开关	RW
	7	C249 的上/下计数方式选择开关	RW
46	0	C250 的上/下计数方式选择开关	RW
	1	C251 的上/下计数方式选择开关	RW
	2	C252 的上/下计数方式选择开关	RW
	3	C253 的上/下计数方式选择开关	RW
	4	C254 的上/下计数方式选择开关	RW
	5	C255 的上/下计数方式选择开关	RW



### 5.1.3 RST (装置复位)

类型	助记符	P	装置复位																							
字	RST	■	Out1(b)																							
双字	RST		Out1(W)																							
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
Out1		■	■	■	■	■	■	■		■	■	■					■	■	■	■			■			

**指令说明:**

- 1, Out1: 指定的被执行装置;
- 2, 当 RST 指令被执行, 其指定的装置的动作如下:

组件	状态
Y,M	线圈及接点都会被设定为 off
T,C,HC	目前计时或计数值会被设为 0, 且线圈及接点都会被设定为 off。
SM	部分为只读的 SM 不会改变状态, 其余会被设定为 off
AQ,D,E	内容值会被设为 0
SD	部分为只读的 SD 不会改变值, 其余会被设定为 0

- 3, 若 RST 指令没有被执行, 其指定组件的状态保持不变。

**程序范例:**

例:

当 X0.0=On 时, Y0.5 被清 0, 即 Y0.5=off。



### 5.1.4 WDT (清空监视定时器)

类型	助记符	P	清空监视定时器
字	WDT	■	无
双字			

**指令执行注意点:**

1, 在每个周期执行后, 会将当前的扫描时间长度, 以及最大和最小的扫描周期时长放入 SD6, SD7 和 SD8 中。当 PLC 的扫描时间超过扫描周期监控超时时间 SD5 时, **扫描周期超时标志位 SM1.2** 将会置起。

**指令功能说明:**

- 1, PLC 系统中有一监视定时器(watchdog timer), 用来监视 PLC 系统的正常运转。
- 2, WDT 可用来清除 PLC 中监视定时器的计时时间。当 PLC 的扫描时间 (由位址 0 至 END 指令执行时间) 超过设定的扫描周期时间时, ERR 指示灯会亮起, 系统进入停机状态, 使用者必须将 PLC 重新上电, 或使用软件重新下载程序并运行;
- 3, 令监视定时器动作的时机:
  - 1) PLC 系统发生异常;
  - 2) 程序执行时间太长, 造成扫描周期大于扫描周期监控超时时间 SD5 的内容值。可以下列 2 种方法来解决:
    - A、使用 WDT 指令。
    - B、可由 SD5 (出厂值设置为 200ms) 的设置值改变监视时间。

**程序范例:**

**例:** 若程序扫描周期为 300ms, 此时, 可将程序分割为 2 部份, 并在中间放入 WDT 指令, 使得前半及后半程序都在 200ms 以下。



### 5.1.5 FOR (循环范围开始)

类型	助记符	P	循环范围开始																							
字	FOR		In1(W)																							
双字	DFOR		In1(D)																							
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		

**指令执行注意点：**

- 1, 该指令必须直接接在左母线，单独一个网络段；
- 2, 具体的说明在 NEXT 指令中，**两者必须配合使用。**

**指令功能说明：**

- 1, 操作数 In1 表示回路重复执行的次数，取值范围为不小于 0，当指定次数范围小于 0 时，都视为 0；

### 5.1.6 NEXT (循环范围结束)

类型	助记符	P	循环范围结束
字	NEXT		无
双字			

**指令执行注意点：**

- 1, 该指令必须直接接在左母线，单独一个网络段。
- 2, 指定次数范围要大于等于 0，当指定次数范围小于 0 时，都视为 0；
- 3, FOR~NEXT 回路最多可嵌套使用 8 层，但要注意嵌套次数过多时，会使 PLC 扫描周期增加有可能造成监视定时器动作，而导致错误产生。可使用 WDT 指令来解决；
- 4, 下列情形会产生错误：
  - 1) NEXT 指令在 FOR 指令之前；
  - 2) 有 FOR 指令没有 NEXT 指令；
  - 3) FOR~NEXT 指令个数不同。

**指令功能说明：**

- 1, 由 FOR 指令指定 FOR~NEXT 循环来回执行 In1 次后跳出 FOR~NEXT 循环往下继续执行，当为 0 时，直接不执行；
- 2, 若 In1 使用非立即数时，在执行过程中，如果改变循环次数，则根据最新的次数来执行。例如，D0 初始值为 2，之后执行以下指令，当循环到第二次时 X0.0 闭合，D0 将赋值为 5，则循环次数变更为 5 次，那么将继续执行循环 5-2=3 次。如果变更后的循环次数值小于已经执行循环次数的值，那么将直接跳出循环。

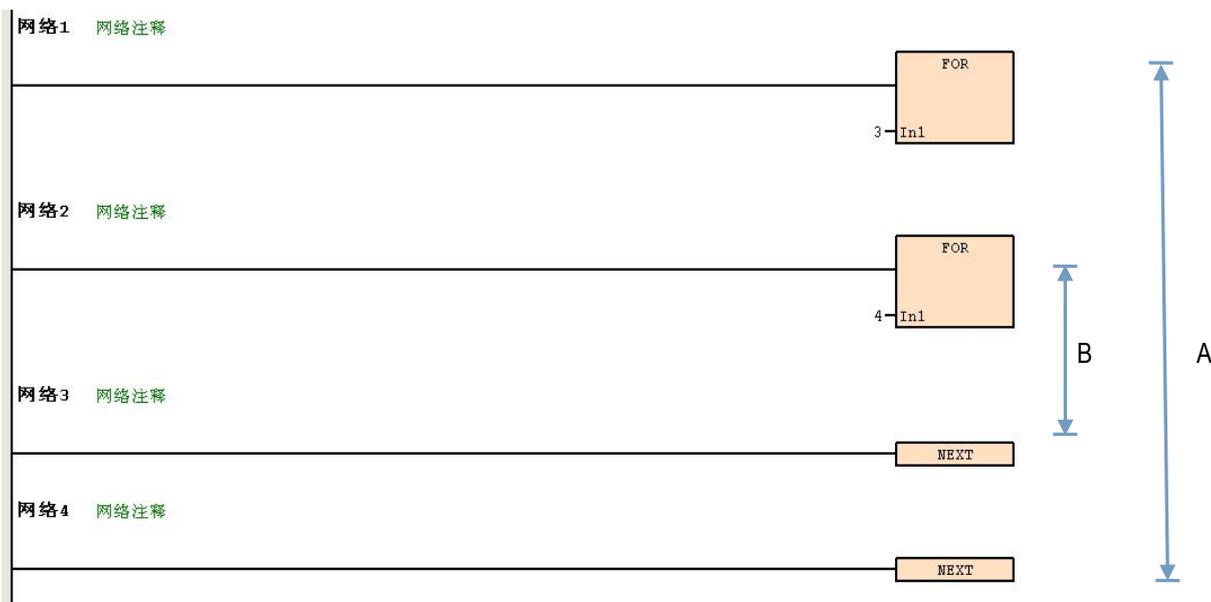
```

FOR   D0
LD    X0.0
MOV   5    D0
NEXT
    
```

**程序范例：**

例：

A 程序执行 3 次后在到 NEXT 指令以后的程序继续执行。而 A 程序每执行一次 B 程序会执行四次，所以 B 程序合计共执行 3 × 4 = 12 次。



### 5.1.7 JMP (跳转指令)

类型	助记符	P	跳转指令																						
字	JMP	■	In1(W)																						
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■		

**参数说明:**

1, In1: 条件跳转目的编号, 即 LBL 指令所指编号。

**指令执行注意点:**

- 1, 跳转目的地所指的程序若在 JMP 指令之前, 需注意会发生扫描周期超时错误, PLC 停止运转, 请注意使用;
- 2, JMP 指令可重复指定同一目的编号;
- 3, 转移执行中的各种装置保持之前状态;
- 4, 若指定的目的编号不存在, 则不进行跳转;
- 5, 主程序和功能块以及中断程序的跳转点互相独立。

**指令功能说明:**

1, 当使用者希望 PLC 程序中的某一部分不需要执行时, 以缩短扫描周期, 以及使用于双重输出时, 可使用 JMP 或 JMPP 指令, 需要配合 LBL 指令使用。

### 5.1.8 LBL (跳转目的编号指令)

类型	助记符	P	跳转目的编号指令																					
字	LBL		In1(W)																					
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1																							■	

**参数说明:**

1, In1: 跳转入的目的编号, 必须只能为常数。

**指令执行注意点:**

- 1, 该指令必须直接接在左母线, 单独一个网络段;
- 2, 主程序可编号 0~255, 各功能块和中断程序编号为 0~7;
- 3, 不可位于 FOR 和 NEXT 指令之间。

**程序范例:**

例:

当 X0.0=On 时, 程序自动从网络 1 转移至网络 3 继续执行, 网络 2 跳过不执行, 即此时 X0.1=On, Y0.0=Off。

当 X0.0=Off 时, 程序如同一般程序由网络 1 继续往下执行, 此时 JMP 指令不被执行, 即此时 X0.1=On, Y0.0=On。



### 5.1.9 EI (中断使能)

类型	助记符	P	中断使能																							
字	EI	■	In1(W)																							
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		

**参数说明:**

1, In1: 指定使能的中断编号。

**指令执行注意点:**

- 1, 使能中断前, 请确认是否设置了中断, 否则将无法正常使用;
- 2, 中断编号 0~5, 对应着 0~3 外部输入中断, 4 定时中断, 5 高速计数中断;
- 3, 对应的当前中断程序上电使能标志位 SM20.0~SM20.5。

### 5.1.10 DI (中断禁用)

类型	助记符	P	中断禁用 (失能)																					
字	DI	■	In1(W)																					
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

1, In1: 要禁用使用的中断编号。

**指令执行注意点:**

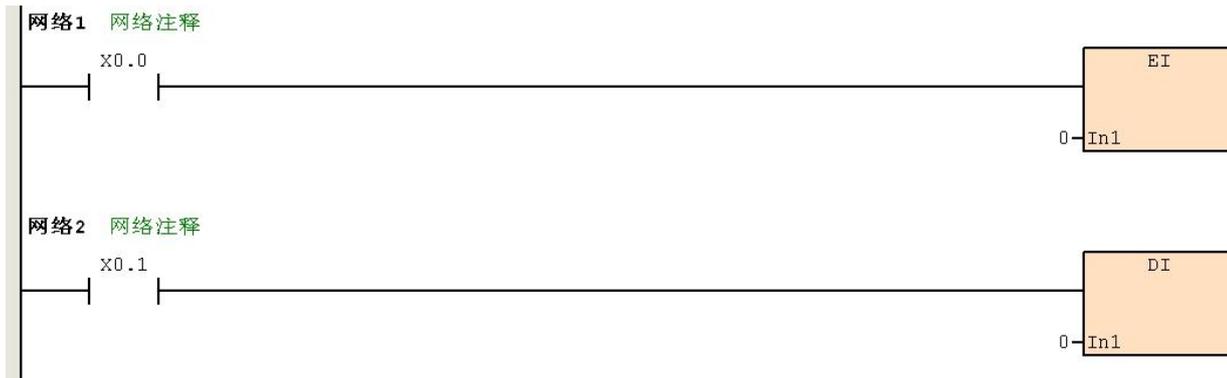
- 1, 中断编号 0~5, 对应着 0~3 外部输入中断, 4 定时中断, 5 高速计数中断;
- 2, 对应的当前**中断是否使能标志位** SM20.0~SM20.5。

**程序范例:**

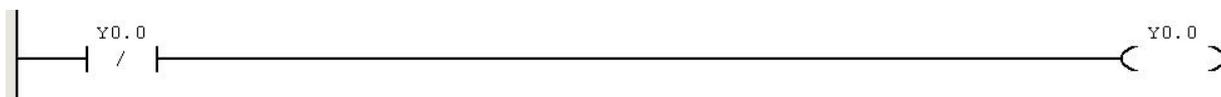
例:

中断程序 INT0 为外部输入中断, 中断来源是 X0.4, 触发方式是上升沿。当 X0.0 置 On 后, 中断程序 INT0 使能, X0.4 每一次置 On, Y0.0 在 On、Off 之间交替输出。当 X0.1 置 On 后, 中断程序 INT0 禁用, 此时不论 X0.4 如何变化, Y0.0 都保持之前的状态不变。

主程序:



中断程序 INT0:





### 5.1.11 REF (I/O 状态实时刷新)

类型	助记符	P	I/O 状态实时刷新																							
字	REF	■	In1(W)												Out1(b)											
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
Out1	■	■																								

**参数说明:**

- 1, In1: I/O 更新处理数目;
- 2, Out1: I/O 更新处理起始装置。

**指令执行注意点:**

- 1, Out1 操作数必须指定在本机的输入/输出端子范围内, 超出部分将不处理。

**指令功能说明:**

- 1, PLC 的输入/输出端子的状态全部为程序扫描至 END 后, 才作状态的更新, 其中输入点的状态是在程序开始扫描时, 自外部输入点的状态读入存在输入点内存中, 而输出端子在 END 指令后, 才将输出点内存内容送至输出装置。因此在运算过程中需要最新的输入/输出数据时, 则可利用本指令。

**程序范例:**

例:

当 X0.0=On 时, Y0.0~Y0.4 的 5 点输出信号实时被送至输出端, 输出信号立即更新, 不必到 END 指令才输出。



### 5.1.12 FBx (功能块调用)

助记符	功能
FBx	功能块调用
操作数	用户自定义

**指令说明:**

- 1, 在调用功能块时, 性质和普通应用指令类似。只有在能流值为 1 时, 才执行, 否则跳过该功能块内的所有内容;
- 2, 不可在功能块和中断中使用上升/下降沿触发型开关;
- 3, 每次进入功能块和中断都等同于第一次进入, 上升沿触发型指令等同于常执行指令;
- 4, 一共可建立和调用 60 个功能块, 嵌套调用最多 8 层;
- 5, 可建立最多合计 8 个的输入, 输出, 输入输出外部局部变量类型;
- 6, 单个功能块最多可定义 16 个内部局部变量。

## 六 应用型指令

### 6.1 传送比较指令

#### 6.1.1 CMP (比较设置输出)

类型	助记符	P	比较设置输出																						
字	CMP	■	In1(W)									In2(W)									Out1(b)				
双字	DCMP		In1(D)									In2(D)									Out1(b)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■		
Out1		■	■	■																					

**参数说明:**

- 1, In1: 比较值;
- 2, In2: 被比较值;
- 3, Out1: 比较结果。

**指令执行注意点:**

- 1, 大小比较是以有符号数进行比较。

**指令功能说明:**

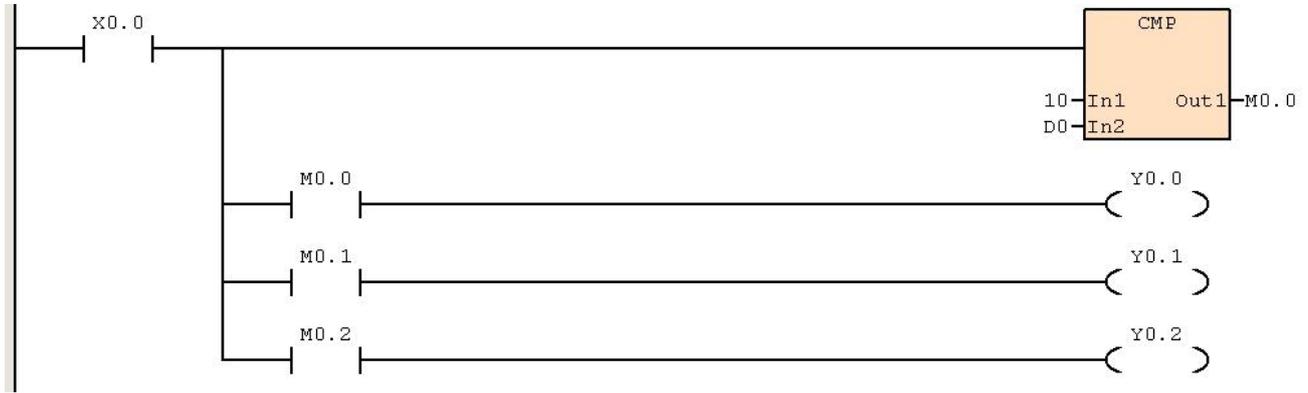
- 1, 将操作数 In1 和 In2 的内容作大小比较, 其比较结果在 Out1 作表示;
- 2, 以 Out1 开始的连续三个位装置来分别表示三个比较结果, 大于, 等于或小于; 例如, Out1 指定装置 M0.0, 那么当 In1 大于 In2 时, M0.0 置起, 两者相等时, M0.1 置起, In1 小于 In2 时, M0.2 置起, 对应装置置起同时, 将会置 off 另两个装置。

**程序范例:**

**例:**

- 指定装置为 M0.0, 则自动占用 M0.0, M0.1 及 M0.2;
- 当 X0.0=On 时, CMP 指令执行, M0.0, M0.1 及 M0.2 其中之一会 On, 当 X0.0=Off 时, CMP 指令不执行, M0.0, M0.1 及 M0.2 状态保持在 X0.0=Off 之前的状态;
- 若需要得到  $\geq$ 、 $\leq$ 、 $\neq$  的结果时, 可将 M0.0~M0.2 串并联即可取得。  
 $10 > D0$  现在值时 Y0.0 为 ON;  
 $10 = D0$  现在值时 Y0.1 为 ON;  
 $10 < D0$  现在值时 Y0.2 为 ON。

(程序图见下页)



### 6.1.2 ZCP (区间比较输出)

类型	助记符	P	区间比较输出																					
			In1(W)							In2(W)							In3(W)						Out1(b)	
字	ZCP	■	In1(D)							In2(D)							In3(D)						Out1(b)	
双字	DZCP		位装置										字/双字装置											
变量	X	Y	S	M	T	C	H	L/	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	
			M				C	C	E			M			C	C	I	Q	D	D	E	D		L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In3									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Out1		■	■	■																				

**参数说明:**

1, In1: 区间比较的下限值; In2: 区间比较的上限值; In3: 被比较值; Out1: 比较结果。

**指令执行注意点:**

- 1, 当下限 In1>上限 In2 时, 则指令即为下限 In1 与被比较值 In3 进行 CMP 指令比较;
- 2, 大小比较是以有符号数进行比较。

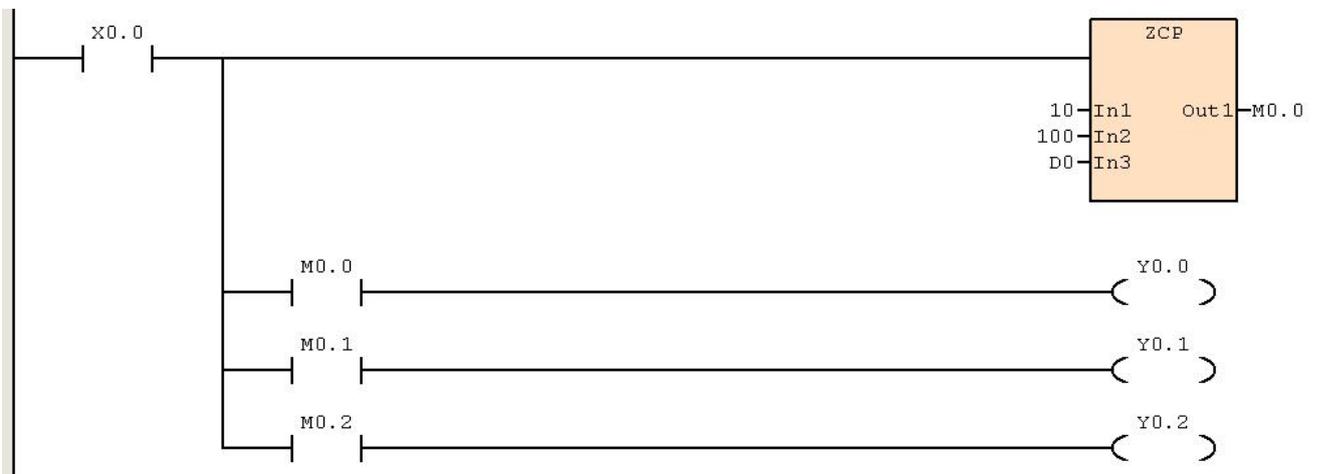
**指令功能说明:**

1, 被比较值 In3 与下限 In1 及上限 In2 作比较, 比较结果在 Out1 连续的三个位装置表示; 例如 Out1 指定为 M0.0, In1 大于 In3 时, M0.0 置起, In1<=In3<=In2 时, M0.1 置起, In2<In3 时, M0.2 置起, 对应装置置起同时, 将会置 Off 另两个装置。

**程序范例:**

**例:**

- 指定装置为 M0.0, 则自动占有 M0.0, M0.1 及 M0.2。
- 当 X0.0=On 时, ZCP 指令执行, M0.0, M0.1 及 M0.2 其中之一会 On, 当 X0.0=Off 时, ZCP 指令不执行, M0.0, M0.1 及 M0.2 状态保持在 X0.0=Off 之前的状态。
- 10>D0 现在值时 Y0.0 为 ON;
- 10<=D0 现在值<=100 时 Y0.1 为 ON;
- D0 现在值>100 时 Y0.2 为 ON。



### 6.1.3 MOV (数据搬移)

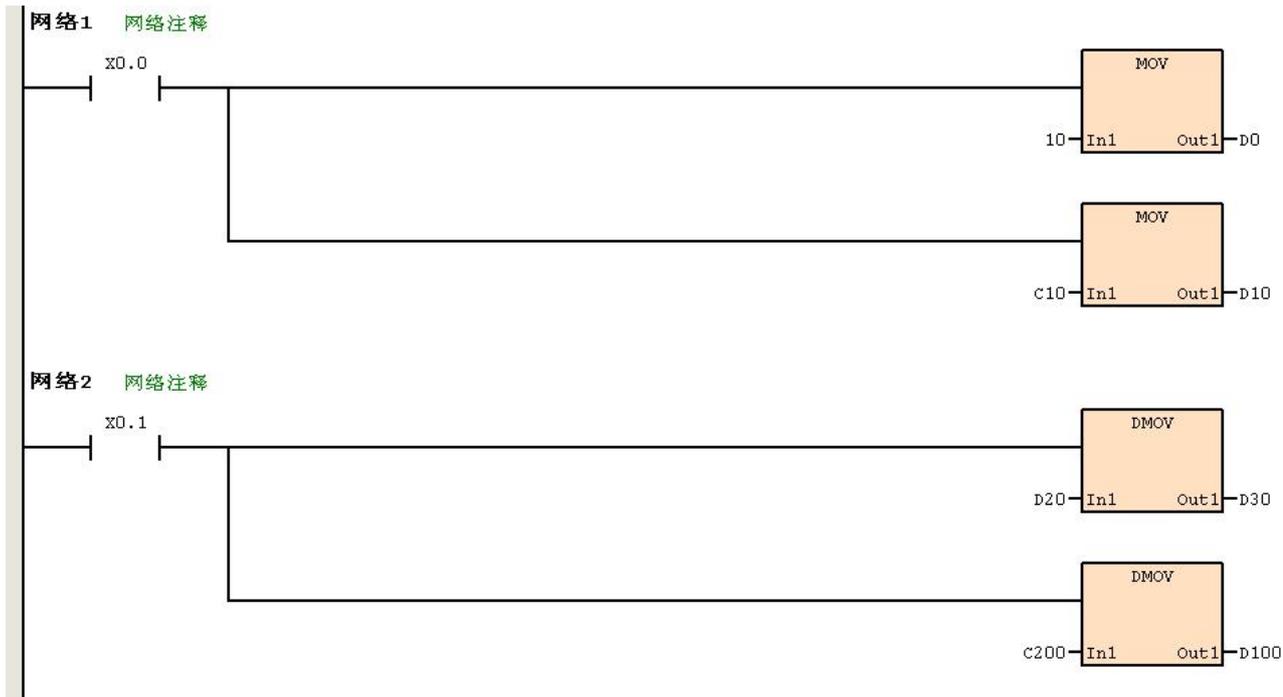
类型	助记符	P	数据搬移																				
字	MOV	■	In1(W)										Out1(W)										
双字	DMOV		In1(D)										Out1(D)										
变量	位装置										字/双字装置												
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1								■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Out1									■	■	■	■	□	□		■	■	■	■				■

**指令说明:**

- 1, In1: 资料来源。Out1: 数据搬移目的装置。
- 2, 当该指令执行时, 将 In1 的内容直接搬移至 Out1 内。当指令不执行时, Out1 内容不会变化。
- 3, 若运算结果为 32 位输出时 (如应用指令 MUL 等) 和 32 位装置高速计数器的现在值数据搬动则必须要用 **DMOV 指令**。

**程序范例:**

- 例:**
- 16 位数据搬移, 须使用 MOV 指令。
  - 当 X0.0=Off 时, D0 内容没有变化, D10 内容没有变化, 若 X0.0=On 时, 将数值 10 传送至 D0 数据寄存器内, 将 C10 现在值传送至 D10 数据寄存器内。
  - 32 位数据搬移, 须使用 DMOV 指令。
  - 当 X0.1=Off 时, (D31、D30)、(D101、D100)内容没有变化, 若 X0.1=On 时, 将(D21、D20)现在值传送至(D31、D30)数据寄存器内。同时, 将 C200 现在值传送至(D101、D100)数据寄存器内。



### 6.1.4 SMOV (移码传送)

类型	助记符	P	移码传送																								
			In1(W)					In2(W)					In3(W)					In4(W)					InOut1(W)				
字	SMOV	■	In1(D)					In2(W)					In3(W)					In4(W)					InOut1(D)				
双字	DSMOV	■	In1(D)					In2(W)					In3(W)					In4(W)					InOut1(D)				
变量	位装置										字/双字装置																
	X	Y	S	M	T	C	H	L/			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/		
			M				C	H	E			M					C	I	Q		D				L		
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In3										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In4										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
InOut1											■	■	■	■	□	□		■	■	■	■	■			■		

**参数说明:**

- 1, In1: 资料来源;
- 2, In2: 数据来源传送起始码数;
- 3, In3: 数据来源传送码的个数;
- 4, In4: 传送的目的起始码数;
- 5, InOut1: 传送的目的装置。

**指令执行注意点:**

- 1, 使用 16 位指令时, In2 的操作数范围为 1~4, In3 的操作数范围为 1~In2, In4 的操作数范围为 In3~4;  
 使用 32 位指令时, In2 的操作数范围为 1~8, In3 的操作数范围为 1~In2, In4 的操作数范围为 In3~8。
- 2, SMOV 有个**工作模式选择位** SM50.0, 当 SM50.0 为 OFF 时, 为 BCD 值模式, 将会将 BIN 值自动转化为 BCD 值来进行传送, 再将结果自动转化为 BIN 值; 当 SM50.0 为 ON 时, 为 BIN 值模式。16 位指令, 当 BCD 值超过 10 进制的 0~9999 的范围时, 以及 32 位指令, BCD 数值超过 10 进制的 0~99999999, 指令不执行。

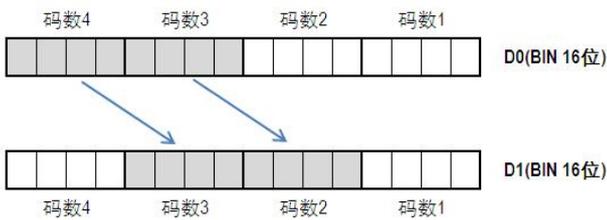
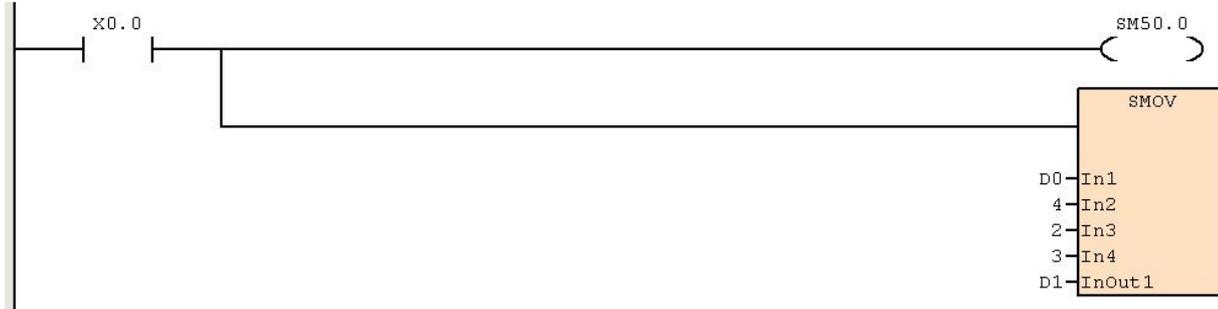
**指令功能说明:**

- 1, 此指令可将数据重新分配或合成。该指令执行时, 指定 In1 的第 In2 码数开始往低位计算的共 In3 个码内容传送到 InOut1 的第 In4 码数开始往低位计算的共 In3 个码中。

程序范例:

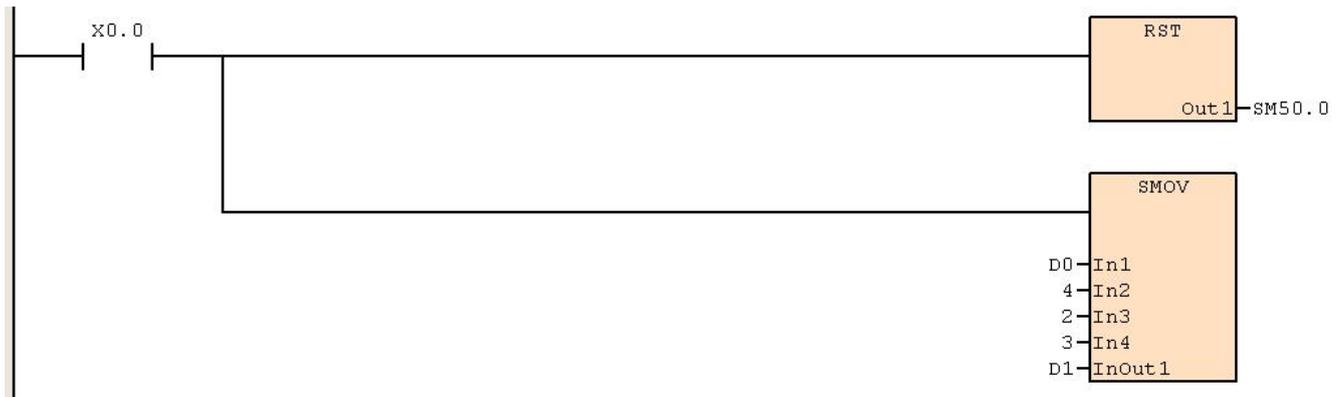
例 1:

D0 = 16#1234, D1 = 16#5678, X0.0=On 时, SM50.0 为 On, 执行该指令将会把 D0 中的第 4 码以及第 3 码两个码值, 即 16#12, 移入 D1 的第 3 码和第 2 码两个码中, 执行后, D1 = 16#5128。



例 2:

当 SM50.0=Off 时(BCD 模式), X0.0=On, 指定 D0 的 10 进制数值的第 4 位数(即千位数)开始往低位计算的 2 位数内容传送至 D1 的 10 进制数值的第 3 位数(也即百位数)开始往低位计算的 2 位数中。而 D1 的  $10^3$  及  $10^0$  于本指令被执行后内容没有变化。



若执行前 D0=1,234, D1=5,678, 执行完毕后, D0 不变, D1=5,128。

### 6.1.5 CML (反相传送)

类型	助记符	P	反相传送																							
字	CML	■	In1(W)												Out1(W)											
双字	DCML		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□	□		■	■	■				■			

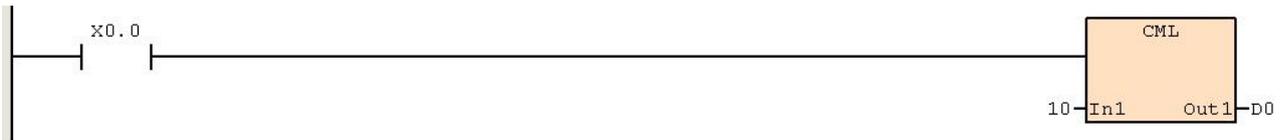
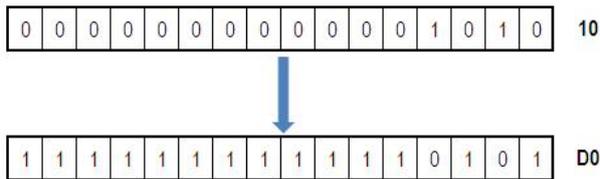
**指令说明:**

- 1, In1: 传送的数据来源; Out1: 传送的目的装置;
- 2, 作反相输出时, 使用本指令。将 In1 的内容进行二进制位数全部反相(0->1、1->0)传送至 Out1 当中。如果内容为常数时, 此常数自动被转换成 BIN 值。

**程序范例:**

**例:**

当 X0.0=On 时, 将 10 的 b0~b15 内容反相后传送到 D0 中。(D0=-11, 最高位为符号位)



### 6.1.6 BMOV (范围装置传送)

类型	助记符	P	范围装置传送																						
字	BMOV	■	In1(W)							In2(W)							Out1(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	L
In1										■	■	■	■	■	□		■	■	■	■	■	■			
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Out1											■	■	■	■	□			■	■	■	■				

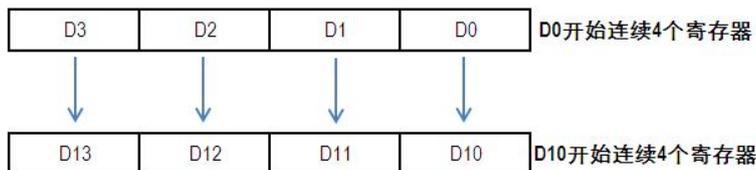
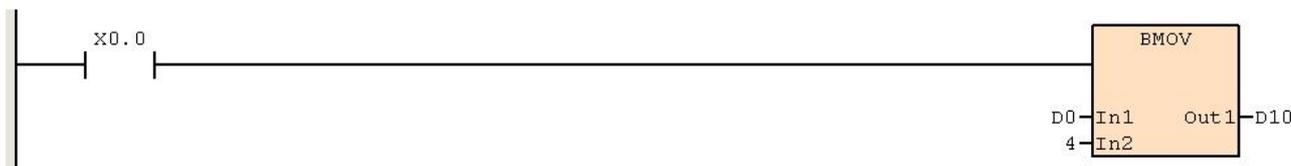
**指令说明:**

- 1, In1: 来源装置起始; In2: 传送装置的个数; Out1: 目的装置起始;
- 2, In2 传送装置的个数取值范围为 1~2000, 超出范围指令不执行;
- 3, In1 所指定的装置起始号码开始算 In2 个寄存器的内容被传送至 Out1 所指定的装置起始号码开始算 In2 个寄存器当中。

**程序范例:**

例:

当 X0.0=On 时, D0~D3 的 4 个寄存器的内容被传送至 D10~D13 的 4 个寄存器内。



### 6.1.7 FMOV (多点数据传送)

类型	助记符	P	多点数据传送																						
			In1(W)							In2(W)							Out1(W)								
字	FMOV	■	In1(D)							In2(W)							Out1(D)								
双字	DFMOV			In1(D)							In2(W)							Out1(D)							
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/		X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
Out1										■	■	■	■	□	□		■	■	■	■	■				

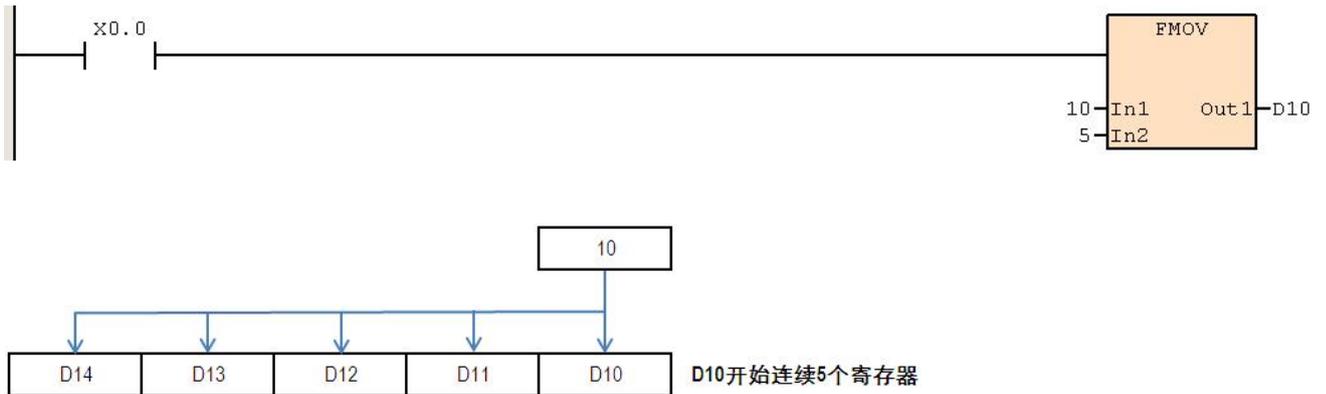
**指令说明:**

- 1, In1: 资料来源; In2: 传送数据的个数; Out1: 目的装置的起始。
- 2, In2 传送装置的个数取值范围为 1~2000, 超出范围指令不执行;
- 3, In1 的内容被传送至 Out1 所指定的装置起始号码开始算 In2 个数据长度当中。

**程序范例:**

**例:**

当 X0.0=On 时, 10 被传送到由 D10 开始的连续 5 个寄存器中。



### 6.1.8 XCH (数据交换)

类型	助记符	P	数据交换																				
字	XCH	■	InOut1(W)										InOut2(W)										
双字	DXCH		InOut1(D)										InOut2(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
InOut1										■	■	■	■	□	□		■	■	■	■			■
InOut2										■	■	■	■	□	□		■	■	■	■			■

**参数说明:**

- 1, InOut1: 将互相交换的数据 1;
- 2, InOut2: 将互相交换的数据 2。

**指令执行注意点:**

- 1, 当 XCH 指令的操作数工作模式 SM50.1 为 Off, 将 InOut1 和 InOut2 所指定的装置内容值互相交换; 当 SM50.1 为 On, 将 InOut1 和 InOut2 的装置内容的上下 8 位 (16 位指令) 或上下 16 位 (32 位指令) 互相交换后, 再将两装置内容互相交换。

**指令功能说明:**

- 1, 该指令为将 InOut1 和 InOut2 所指定的装置内容值互相交换。

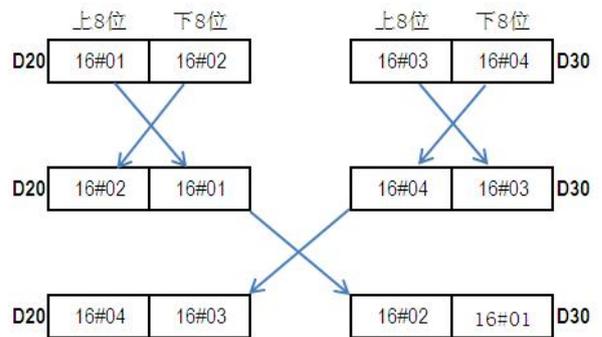
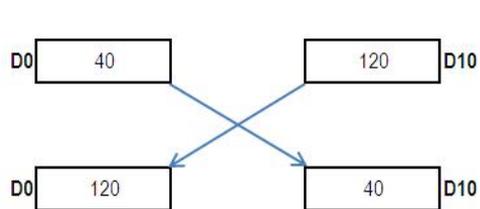
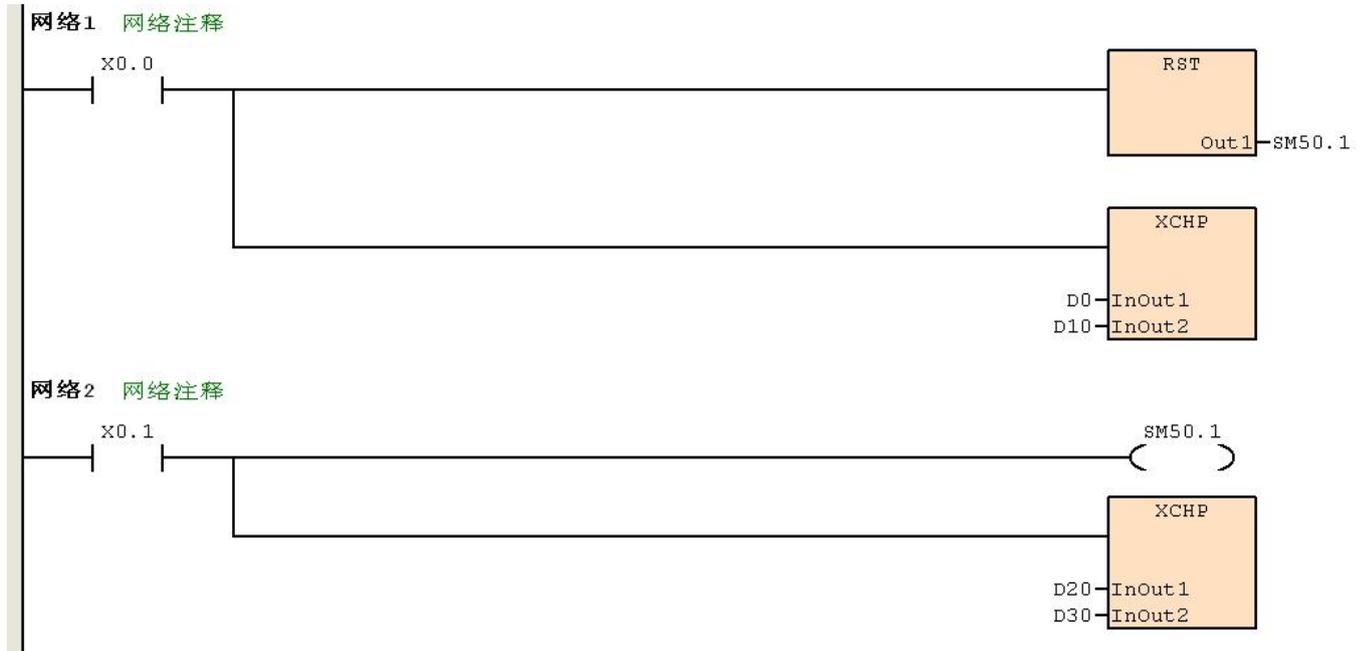


程序范例:

例:

X0.0=Off→On 时, SM50.1=Off, D0 与 D10 的内容互相交换。

X0.1=Off→On 时, SM50.1=On, D20 与 D30 的内容上下 8 位互相交换后, 再将两装置内容互相交换。



### 6.1.9 BCD (BIN->BCD 转换)

类型	助记符	P	BIN->BCD 转换																				
字	BCD	■	In1(W)										Out1(W)										
双字	DBCDCD		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■

**参数说明:**

- 1, In1: 资料来源;
- 2, Out1: 变换的结果。

**指令执行注意点:**

- 1, BCD 变换结果若超过 0~9999, 则取 9999;
- 2, DBCD 变换结果超过 0~99999999, 则取 99999999;

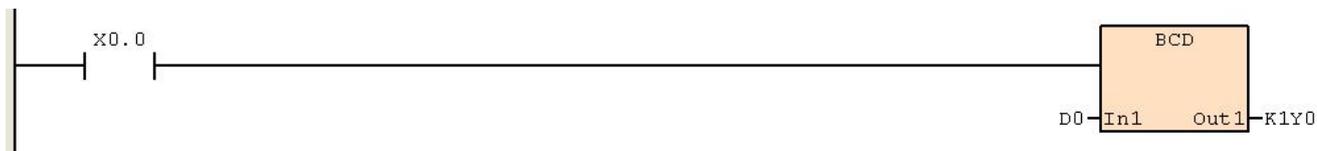
**指令功能说明:**

- 1, 数据来源 In1 的内容 (BIN 值) 作 BCD 的转换, 存于 Out1 中;
- 2, PLC 内的四则运算, 以及 INC、DEC 指令都是以 BIN 方式来执行。所以在应用方面, 当要看到 10 进制数值的显示器时, 用 BCD 转换即可将 BIN 值变为 BCD 值输出。

**程序范例:**

例:

当 X0.0=On 时, D0 的 BIN 值被转换成 BCD 值后, 将结果的个位和十位数存于 K1Y0 (Y0~Y7) 八个 bit 组件。Y0.0~Y0.3 为低位, Y0.4~Y0.7 为高位, 若 D0=100011=35, 则 Y0.0=On, Y0.2=On, Y0.4=On, Y0.5=On。



### 6.1.10 BIN (BCD->BIN 转换)

类型	助记符	P	BCD->BIN 转换																							
字	BIN	■	In1(W)												Out1(W)											
双字	DBIN		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/		X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/		
			M				C	C	E	L			M				C	C	I	Q	D	D	E	D	O	E
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	■
Out1											■	■	■	■	□	□		■	■	■	■					■

**参数说明:**

- 1, In1: 资料来源;
- 2, Out1: 变换的结果。

**指令执行注意点:**

- 1, 数据来源 In1 的内容有效数值范围: BCD (0~9999) , DBCD (0~99999999) ;
- 2, 当 In1 的资料内容并非为 BCD 值 (以 Hex 表示有任一位数不在 0~9 的范围内) , 则该指令不执行;
- 3, 常数会自动转换为 BIN, 故不需运用此指令。

**指令功能说明:**

- 1, 资料来源 In1 的内容 (BCD: 0~9999) 作 BIN 的转换, 存于 Out1 中。

**程序范例:**

例:

令 X0.0=On, X0.2=On, X0.4=On, X0.5=On, 当 X1.0=On 时, 指令执行, K1X0 的 BCD 值被转换成 BIN 值后, 将结果存于 D0 中, D0=35=100011。X0.0~X0.3 为低位, X0.4~X0.7 为高位。



### 6.1.11 MVM (指定位搬移)

类型	助记符	P	指定位搬移																						
字	MVM	■	In1(W)									In2(W)									InOut1(W)				
双字	DMVM		In1(D)									In2(D)									InOut1(D)				
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L	E	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	
In1										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	
In2										■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	
InOut1											■	■	■	■	□	□		■	■	■	■			■	

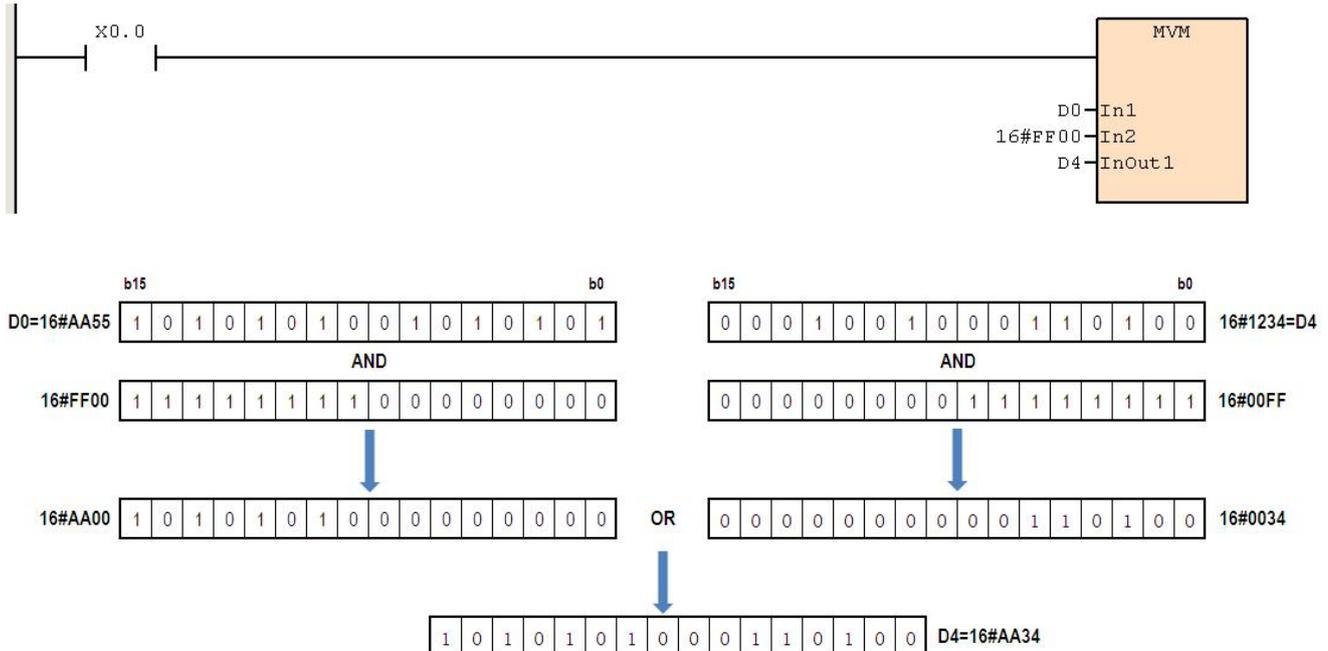
**指令说明:**

- 1, In1: 来源装置; In2: 指定遮蔽(off)的 bit; InOut1: 搬移目的装置 = (In1 & In2) | (InOut1 & ~In2);
- 2, 该指令, 将 In1 中的资料, 根据 In2 所指定的位(on 位为指定位), 搬移到 InOut1 所指示装置中的对应指定位中;
- 3, 让 In1 与 In2 做逻辑“与”(AND)运算, In2 的取反值在与 InOut1 做逻辑的“与”(AND)运算, 再将前 2 者运算结果作逻辑的“或”(OR) 运算存入 InOut1 中;
- 4, 逻辑的“与”(AND)运算之规则为任一为 0 结果为 0;
- 5, 逻辑的“或”(OR)运算之规则为任一为 1 结果为 1。

**程序范例:**

例:

当 X0.0 = On 时, 16 位 D0 与 16#FF00 做逻辑“与”(AND) 运算, D4 与 16#00FF (16#FF00 的取反值) 做逻辑“与”(AND) 运算, 再将前 2 者运算结果做逻辑的“或”(OR) 运算, 将结果存入 D4。



### 6.1.12 MMOV (放大传送)

类型	助记符	P	放大传送																							
字	MMOV	■	In1(W)												Out1(D)											
双字																										
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	L	
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■
Out1											■	■	■	■	□	□		■	■	■	■					■

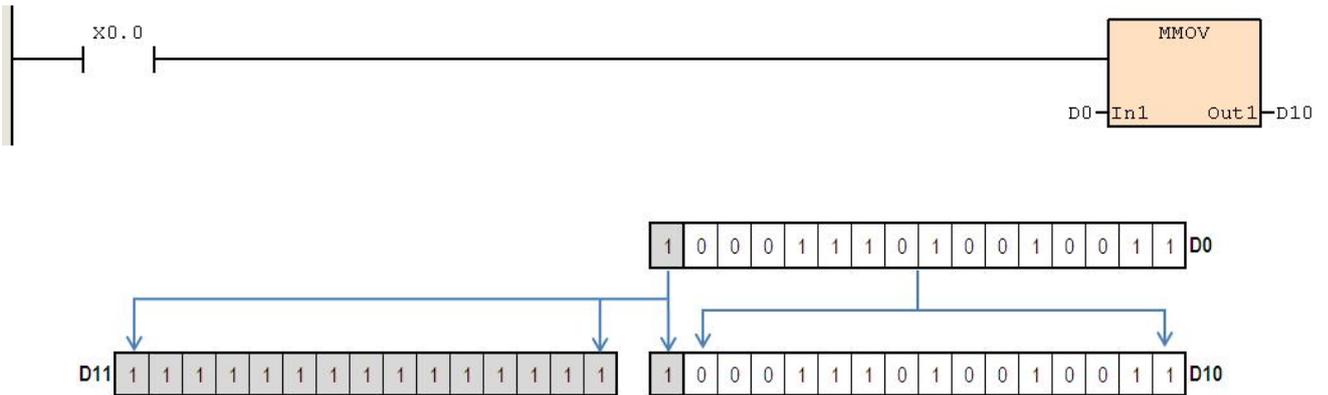
**指令说明:**

- 1, In1: 数据来源 (16 位) ; Out1: 数据搬移目的地 (32 位) ;
- 2, 将 16 位装置 In1 的有符号数传送到 32 位的装置 Out1 中, 保持数值大小不变。

**程序范例:**

例:

当 X0.0 为 On 时, D0 的数据传送到 D10 和 D11 中, 数值大小保持不变。



### 6.1.13 CMPT（表格比较指令）

类型	助记符	P	表格比较指令																							
字	CMPT	■	In1(W)						In2(W)						In3(W)						Out1(D)					
双字	DCMPT		In1(D)						In2(D)						In3(W)						Out1(D)					
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■					
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■				
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
Out1										■	■	■	■	■	■		■	■	■	■	■		■			

**参数说明：**

- 1, In1：比较来源起始装置；
- 2, In2：被比较来源起始装置；
- 3, In3：比较数据长度与比较功能设定；
- 4, Out1：目标装置。

**指令执行注意点：**

- 1, In3 操作数之低 8 位表示比较之长度设定，指令范围为 1~32；比数值 1 还小以 1 执行，比 32 还大以 32 执行；
- 2, In3 操作数之高 8 位表示比较条件设定， 0: In1 等于 In2； 1: In1 小于 In2； 2: In1 小于等于 In2； 3: In1 大于 In2； 4: In1 大于等于 In2；当数值均不在 0~4 范围时，默认为 0，即等于条件；
- 3, 比较模式设置，当使用 DCMPT 指令时，如果**比较模式设定位** SM50.5 置为 On，则比较数值将以浮点数进行比较，SM50.5 置位为 Off 时，以有符号 32 位数据进行比较。

**指令功能说明：**

- 1, 对比较结果符合的，将会按顺序置起 Out1 装置的对应位(bit)，Out1 固定占用 32 位数据装置，未比较满 32 位的，高位以 0 填充。



**程序范例:**

**例:**

当 X0.0=On 时, 将位于 D0~D7 与 D20~D27 内的 16 位数值做“等于”比较, 比较结果存于 D100 内。



其中 D0 的字符串内容如下表:

编号	D0	D1	D2	D3	D4	D5	D6	D7
数值	10	20	30	40	50	60	70	80

其中 D20 的字符串内容如下表:

编号	D20	D21	D22	D23	D24	D25	D26	D27
数值	12	20	33	44	50	66	70	88

经由 CMPT 指令比较之后, 比较相同的数值其所对应之 bit 将会被设定为 1, 其余不相同数值所对应之 bit 都会被清除为 0, 故 D100 所得到的内容如下表所示:

D100	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8~15
	0	1	0	0	1	0	1	0	0...0
16#0052(82)									

## 6.2 整型四则逻辑运算

### 6.2.1 ADD (整型加法)

类型	助记符	P	整型加法																					
字	ADD	■	In1(W)						In2(W)						Out1(W)									
双字	DADD		In1(D)						In2(D)						Out1(D)									
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■				■

**参数说明:**

- 1, In1: 被加数;
- 2, In2: 加数;
- 3, Out1: 和。

**指令执行注意点:**

- 1, **加法相关标识变化:** 在指令运算前将 SM50.2、SM50.3、SM50.4 置 OFF。
  - 16 位整型加法:
    - 1) 运算结果为 0 时, **零标志**(Zero flag)SM50.2 为 On;
    - 2) 运算结果小于-32768 时, **借位标志**(Borrow flag)SM50.3 为 On;
    - 3) 运算结果大于 32767 时, **进位标识**(Carry flag)SM50.4 为 On。
  - 32 位整型加法:
    - 1) 运算结果为 0 时, **零标志**(Zero flag)SM50.2 为 On;
    - 2) 运算结果小于-2147483648 时, **借位标识**(Borrow flag)SM50.3 为 On;
    - 3) 运算结果大于 2147483647 时, **进位标识**(Carry flag)SM50.4 为 On。

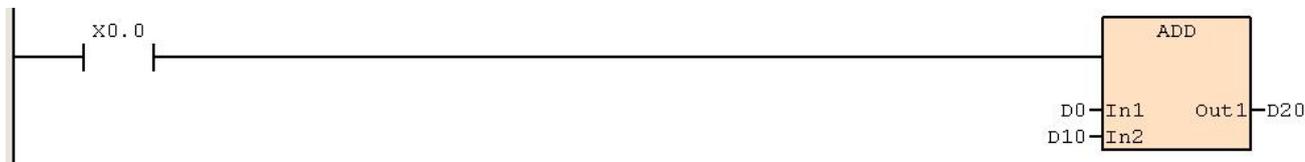
**指令功能说明:**

- 1, 将两个数据源: In1 及 In2 以整型方式相加, 结果存于 Out1 中。

**程序范例:**

例:

16 位整型加法: 当 X0.0=On 时, 被加数 D0 内容加上加数 D10 的内容, 将结果存在 D20 的内容当中。



### 6.2.2 SUB (整型减法)

类型	助记符	P	整型减法																						
字	SUB	■	In1(W)									In2(W)									Out1(W)				
双字	DSUB		In1(D)									In2(D)									Out1(D)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□	□		■	■	■	■			■		

**参数说明:**

- 1, In1: 被减数;
- 2, In2: 减数;
- 3, Out1: 差。

**指令执行注意点:**

1, **减法相关标志变化:** 在指令运算前将 SM50.2、SM50.3、SM50.4 置 OFF。

16 位整型减法:

- 1) 运算结果为 0 时, **零标志**(Zero flag)SM50.2 为 On;
- 2) 运算结果小于-32768 时, **借位标志**(Borrow flag)SM50.3 为 On;
- 3) 运算结果大于 32767 时, **进位标识**(Carry flag)SM50.4 为 On。

32 位整型减法:

- 1) 运算结果为 0 时, **零标志**(Zero flag)SM50.2 为 On;
- 2) 运算结果小于-2147483648 时, **借位标识**(Borrow flag)SM50.3 为 On;
- 3) 运算结果大于 2147483647 时, **进位标识**(Carry flag) SM50.4 为 On。

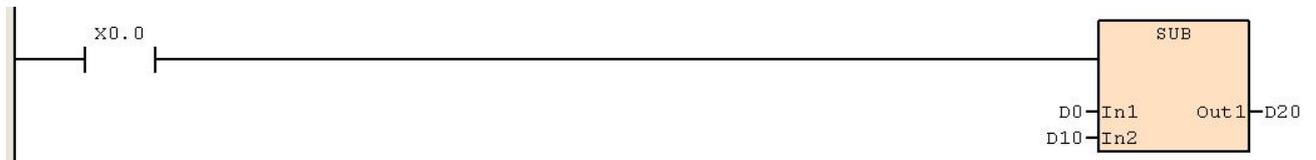
**指令功能说明:**

1, 将两个数据源 In1 及 In2 以整型方式相减, 结果存于 Out1 中。

**程序范例:**

**例:**

16 位 BIN 减法: 当 X0.0=On 时, 将 D0 内容减掉 D10 内容, 差值存在 D20 的内容中。



### 6.2.3 MUL (整型乘法)

类型	助记符	P	整型乘法																					
字	MUL	■	In1(W)					In2(W)					Out1(D)											
双字	DMUL		In1(D)					In2(D)					Out1(D)											
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■	

**参数说明:**

- 1, In1: 被乘数;
- 2, In2: 乘数;
- 3, Out1: 积。

**指令执行注意点:**

- 1, 16 位和 32 位整型乘法运算的结果都是占用 32 位长度, 即一个双字装置。

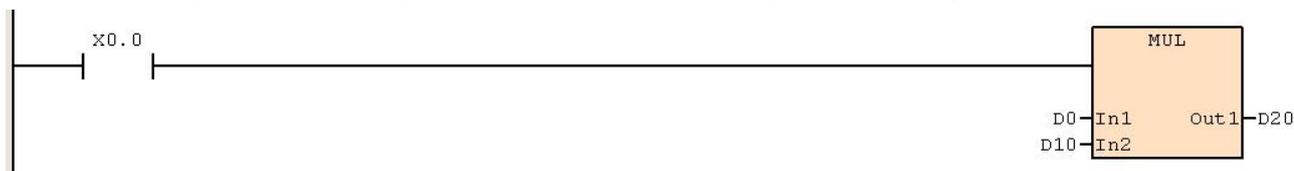
**指令功能说明:**

- 1, 将两个数据源: In1 及 In2 以有符号整型数相乘后, 积存于 Out1 中。

**程序范例:**

例:

16 位 D0 乘上 16 位 D10 其结果是 32 位的积, 高 16 位存于 D21, 低 16 位存于 D20 内。





### 6.2.4 MUL\_W (整型乘法\_字)

类型	助记符	P	整型乘法																					
字	MUL_W	■	In1(W)							In2(W)							Out1(W)							
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■	

**参数说明:**

- 4, In1: 被乘数;
- 5, In2: 乘数;
- 6, Out1: 积。

**指令执行注意点:**

- 1, 该指令的得数, 是 16 位有符号整型, 若数值超出范围, 将会得到不正常的数值。

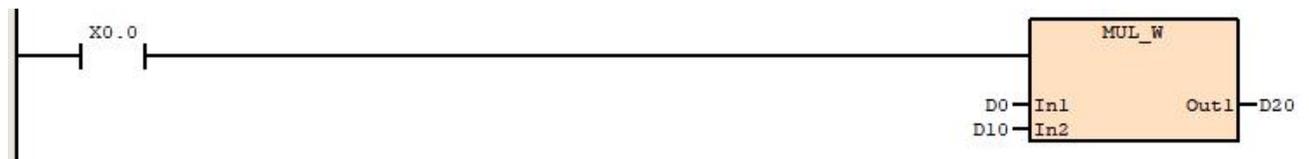
**指令功能说明:**

- 1, 将两个数据源: In1 及 In2 以有符号整型数相乘后, 积存于 Out1 中。

**程序范例:**

例:

16 位 D0 乘上 16 位 D10 其结果是 16 位的积, 存于 D20 内。



### 6.2.5 DIV (整型除法)

类型	助记符	P	整型除法																						
字	DIV	■	In1(W)							In2(W)							Out1(W)								
双字	DDIV		In1(D)							In2(D)							Out1(D)								
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■	
Out1										■	■	■	■	□	□		■	■	■	■			■		

**参数说明:**

- 1, In1: 被除数;
- 2, In2: 除数;
- 3, Out1: 商。

**指令执行注意点:**

- 1, 除数为 0 时, 指令不执行。

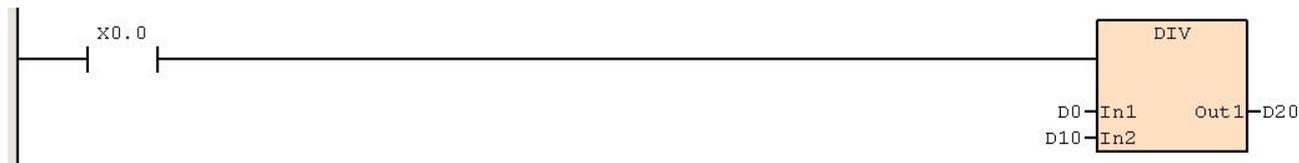
**指令功能说明:**

- 1, 将两个数据源 In1 及 In2 以有符号整型数相除后, 商存于 Out1 中。

**程序范例:**

**例:**

当 X0.0=On 时, 被除数 D0 除以除数 D10, 结果商放于 D20 内。





### 6.2.6 MOD (整型求余数)

类型	助记符	P	整型求余数																						
字	MOD	■	In1(W)									In2(W)									Out1(W)				
双字	DMOD		In1(D)									In2(D)									Out1(D)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□	□		■	■	■	■			■		

**参数说明:**

- 4, In1: 被除数;
- 5, In2: 除数;
- 6, Out1: 余数。

**指令执行注意点:**

- 1, 除数为 0 时, 指令不执行。

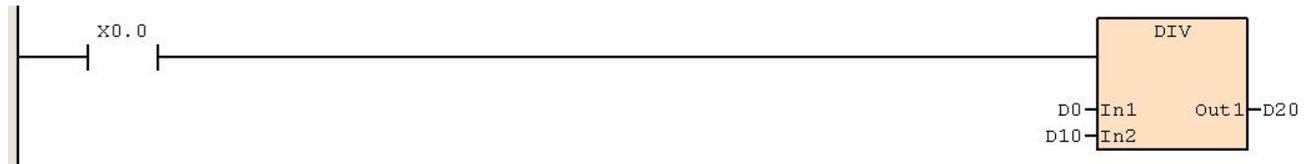
**指令功能说明:**

- 1, 将两个数据源 In1 及 In2 以有符号整型数相除后, 余数存于 Out1 中。

**程序范例:**

**例:**

当 X0.0=On 时, 被除数 D0 除以除数 D10, 结果余数放于 D20 内。



### 6.2.7 INC (自加一)

类型	助记符	P	自加一																				
字	INC	■	InOut1(W)																				
双字	DINC		InOut1(D)																				
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
InOut1										■	■	■	■	□	□		■	■	■	■			■

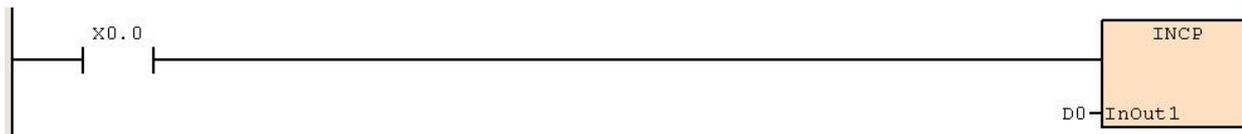
**指令说明:**

- 1, InOut1: 目的装置;
- 2, 若指令不是脉冲执行型, 则当指令执行时, 程序每次扫描周期被指定的装置 InOut1 的内容都会加一;
- 3, 本指令一般都是使用脉冲执行型指令 (INCP, DINCP);
- 4, 16 位运算时, 32767 再加 1 则变成-32768; 32 位运算时, 2147483647 再加 1 则变成-2147483648。

**程序范例:**

例:

当 X0.0=Off→On 时, D0 内容自动加 1。





### 6.2.8 DEC (自减一)

类型	助记符	P	自减一																				
字	DEC	■	InOut1(W)																				
双字	DDEC		InOut1(D)																				
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
InOut1										■	■	■	■	□	□		■	■	■	■			■

**指令说明:**

- 1, InOut1: 目的装置;
- 2, 若指令不是脉冲执行型, 则当指令执行时, 程序每次扫描周期被指定的装置 InOut1 的内容都会减一;
- 3, 本指令一般都是使用脉冲执行型指令 (DECP, DDECP) ;
- 4, 16 位运算时, -32768 再减 1 则变成 32767; 32 位运算时, -2147483648 再减 1 则变成 2147483647。

**程序范例:**

例:

当 X0.0=Off→On 时, D0 内容自动减 1。



### 6.2.9 WAND/DAND (逻辑与运算)

类型	助记符	P	逻辑与 (AND)运算																				
字	WAND	■	In1(W)										In2(W)					Out1(W)					
双字	DAND		In1(D)										In2(D)					Out1(D)					
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■

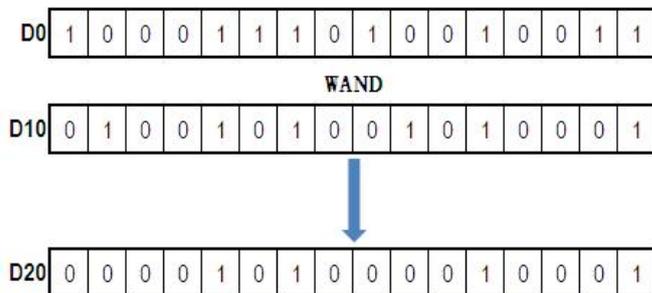
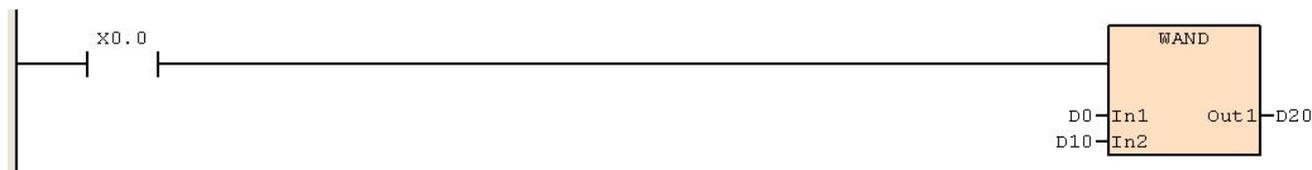
**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2; Out1: 运算结果;
- 2, 两个数据源 In1 及 In2 作按位做逻辑的“与” (AND) 运算并将结果存于 Out1 中;
- 3, 逻辑的“与” (AND) 运算的规则为任一为 0 结果为 0。

**程序范例:**

例:

当 X0.0=On 时, 16 位 D0 与 D10 作 WAND, 逻辑与(AND)运算, 将结果存于 D20 中。



### 6.2.10 WOR/DOR (逻辑或运算)

类型	助记符	P	逻辑或 (OR)运算																							
字	WOR	■	In1(W)								In2(W)								Out1(W)							
双字	DOR		In1(D)								In2(D)								Out1(D)							
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■			
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□	□		■	■	■	■			■			

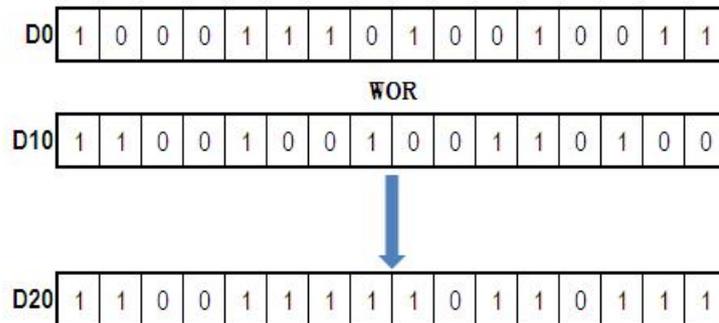
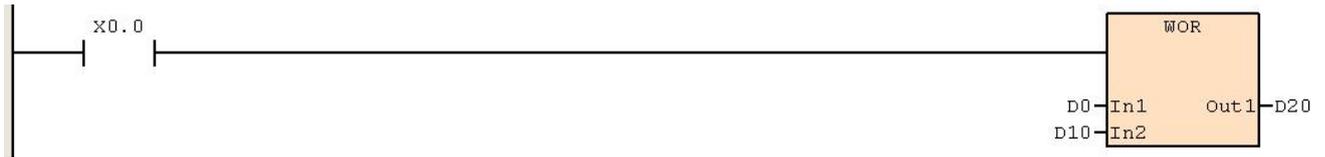
**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2; Out1: 运算结果;
- 2, 两个数据源 In1 及 In2 作按位做逻辑的“或” (OR) 运算并将结果存于 Out1 中;
- 3, 逻辑的“或” (OR) 运算的规则为任一为 1 结果为 1。

**程序范例:**

**例:**

当 X0.0=On 时, 16 位 D0 与 D10 作 WOR, 逻辑或(OR)运算, 将结果存于 D20 中。



### 6.2.11 WXOR/DXOR (逻辑异或运算)

类型	助记符	P	逻辑异或 (XOR)运算																					
字	WXOR	■	In1(W)							In2(W)							Out1(W)							
双字	DXOR		In1(D)							In2(D)							Out1(D)							
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■	■

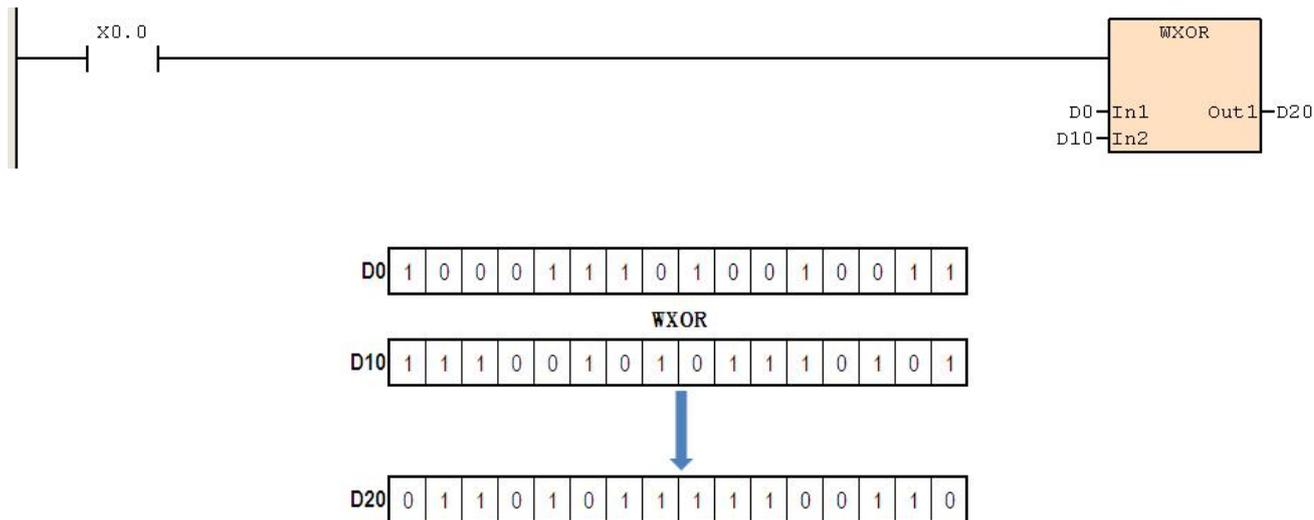
**指令说明:**

- 1, In1: 数据来源装置 1; In2: 数据来源装置 2; Out1: 运算结果;
- 2, 两个数据源 In1 及 In2 作按位做逻辑的“异或” (XOR) 运算并将结果存于 Out1 中;
- 3, 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。

**程序范例:**

例:

当 X0.0=On 时, 16 位 D0 与 D10 作 WXOR, 逻辑异或(XOR)运算, 将结果存于 D20 中。





### 6.2.12 NEG (二进制补码)

类型	助记符	P	二进制补码																							
字	NEG	■	InOut1(W)																							
双字	DNEG		InOut1(D)																							
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
InOut1										■	■	■	■	□	□		■	■	■	■				■		

**指令说明:**

- 1, InOut1: 求补码的装置。
- 2, 本指令一般都是使用脉冲执行型指令 (NEGP, DNEGP) ;
- 3, 功能效果为将 InOut1 装置中的内容全部反相后, 再加 1, 将得到的结果再放入 InOut1 装置中。

**程序范例:**

**例:**

当 X0.0=Off→On 时, D0 内容的各位全部反相 (0→1、1→0) 后再加 1 存放于原寄存器 D0 当中。



### 6.3 位移指令

#### 6.3.1 ROR (右循环移位)

类型	助记符	P	右循环移位																				
字	ROR	■	In1(W)									InOut1(W)											
双字	DROR		In1(W)									InOut1(D)											
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■
InOut1										■	■	■	■	□	□		■	■	■	■			■

**参数说明:**

- 1, In1: 一次循环的位数;
- 2, InOut1: 将循环的装置。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (RORP, DRORP) ;
- 2, 最后一个从最低位移到最高位的位内容, 将被传送到进位标志 SM50.4 中;
- 3, 如果循环位数为负值, 则视为 0。

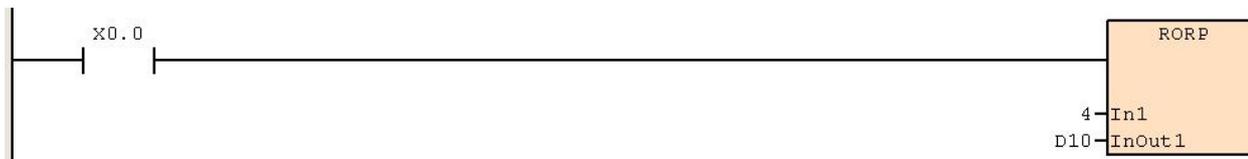
**指令功能说明:**

- 1, 将 InOut1 所指定的装置内容一次向右循环 (即高位向低位移位, 最低位再移到最高位) In1 个位。

**程序范例:**

**例:**

当 X0.0 从 Off→On 变化时, D10 的 16 个位依次向右移动四次, 如下图所示, 有颜色的位内容被传送至进位标志信号 SM50.4。



### 6.3.2 ROL (左循环移位)

类型	助记符	P	左循环移位																							
字	ROL	■	In1(W)												InOut1(W)											
双字	DROL		In1(W)												InOut1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
InOut1										■	■	■	■	□	□		■	■	■	■			■			

**参数说明:**

- 1, In1: 一次循环的位数;
- 2, InOut1: 将循环的装置。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (ROLP, DROLP) ;
- 2, 最后一个从最高位移到最低位的位内容, 将被传送到进位标志 SM50.4 中;
- 3, 如果循环位数为负值, 则视为 0。

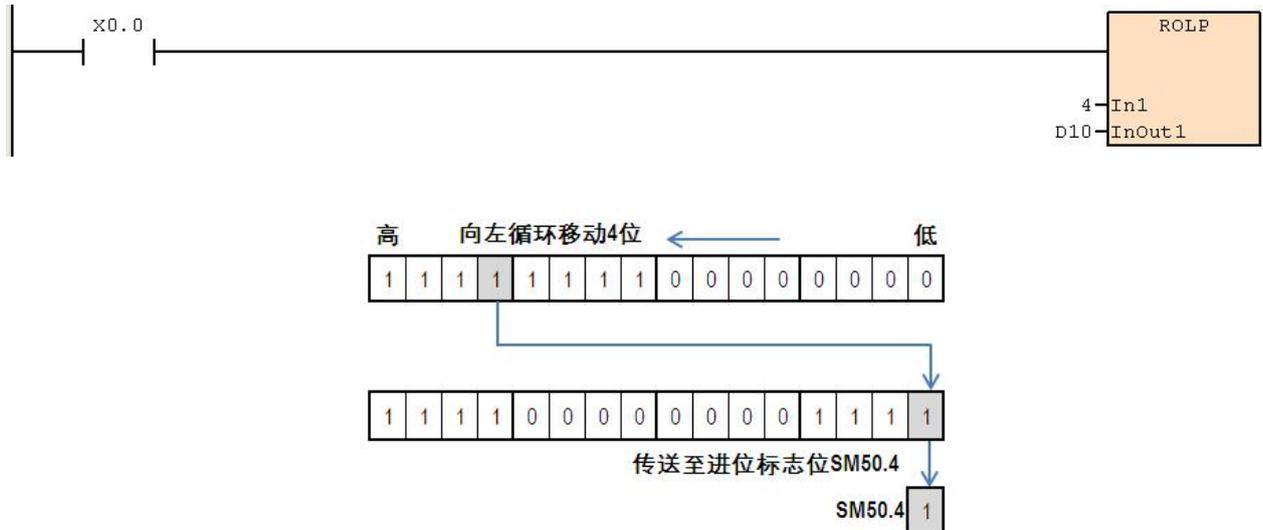
**指令功能说明:**

- 1, 将 InOut1 所指定的装置内容一次向左循环 (即低位向高位移位, 最高位再移到最低位) In1 个位。

**程序范例:**

例:

当 X0.0 从 Off→On 变化时, D10 的 16 个位依次向左移动四次, 如下图所示有颜色的位内容被传送到进位标志信号 SM50.4。



### 6.3.3 RCR (附进位标志右循环移位)

类型	助记符	P	附进位标志右循环移位																				
字	RCR	■	In1(W)										InOut1(W)										
双字	DRCR		In1(W)										InOut1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■
InOut1										■	■	■	■	□	□		■	■	■	■			■

**参数说明:**

- 1, In1: 一次循环的位数;
- 2, InOut1: 将循环的装置。

**指令执行注意点:**

- 3, 本指令一般都是使用脉冲执行型指令 (RCRP, DRCRP) ;
- 4, 最低位向右移位一次将被传送到进位标志 SM50.4 中, 同时原进位标志 SM50.4 中的内容将会被传送到最高位。
- 5, 如果循环位数为负值, 则视为 0。

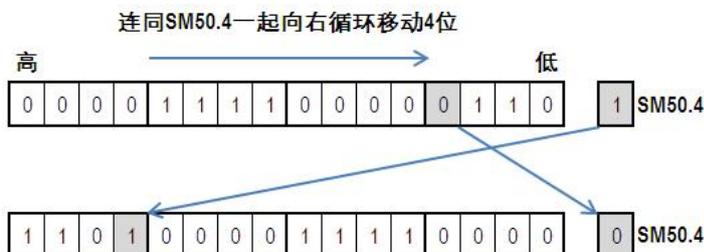
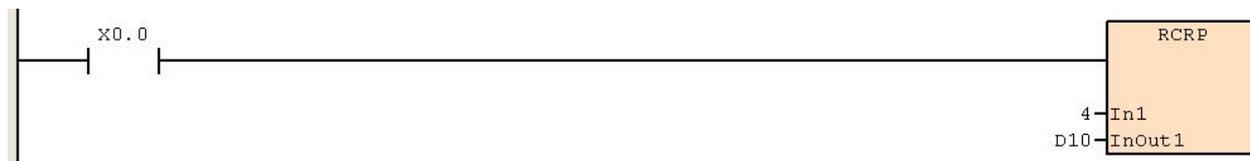
**指令功能说明:**

- 1, 将 InOut1 所指定的装置内容同时将进位标志附加在最后位, 若为 16 位指令时共 17 位数据, 若为 32 位指令时共 33 位数据, 进行一次向右循环 (即高位向低位移位, 最低位再移到最高位) In1 个位;

**程序范例:**

**例:**

当 X0.0 从 Off→On 变化时, D10 的 16 个位连同进位标志 SM50.4 共 17 个位依次向右移动四次, 如下图所示有颜色的位内容 0 被传送至进位标志信号 SM50.4 内。





### 6.3.4 RCL (附进位标志左循环移位)

类型	助记符	P	附进位标志左循环移位																				
字	RCL	■	In1(W)								InOut1(W)												
双字	DRCL		In1(W)								InOut1(D)												
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■
InOut1										■	■	■	■	□	□		■	■	■				■

#### 参数说明:

- 1, In1: 一次循环的位数;
- 2, InOut1: 将循环的装置。

#### 指令执行注意点:

- 1, 本指令一般都是使用脉冲执行型指令 (RCLP, DRCLP) ;
- 2, 最高位向左移位一次将被传送到进位标志 SM50.4 中, 同时原进位标志 SM50.4 中的内容将会被传送到最低位。
- 3, 如果循环位数为负值, 则视为 0。

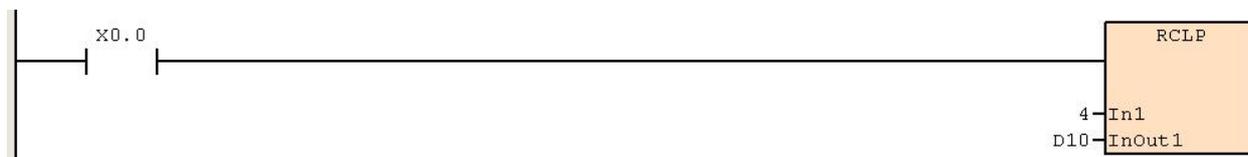
#### 指令功能说明:

- 1, 将 InOut1 所指定的装置内容同时将进位标志附加在最高位, 若为 16 位指令时共 17 位数据, 若为 32 位指令时共 33 位数据, 进行一次向左循环 (即低位向高位移位, 最高位再移位到最低位) In1 个位。

#### 程序范例:

##### 例:

当 X0.0 从 Off→On 变化时, D10 的 16 个位连同进位标志 SM50.4 共 17 个位依次向左移动四次, 如下图所示有颜色的位内容 1 被传送至进位标志信号 SM50.4 内。



### 6.3.5 SFTR (位右移)

类型	助记符	P	位右移																					
			In1(b)					In2(W)					In3(W)					InOut1(b)						
字	SFTR	■																						
双字																								
变量	位装置									字/双字装置														
	X	Y	S	M	T	C	H	L/	E/	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/
In1	■	■	■	■	■	■	■																	
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
InOut1		■	■	■																				

**参数说明:**

- 1, In1: 移位装置的起始编号;
- 2, In2: 将移位的数据长度;
- 3, In3: 一次移位的位数;
- 4, InOut1: 将移位装置的起始编号。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (SFTRP) ;
- 2, In2 操作数范围 = 1~1024; In3 操作数范围 = 1~In2, 以及装置要在位范围内。

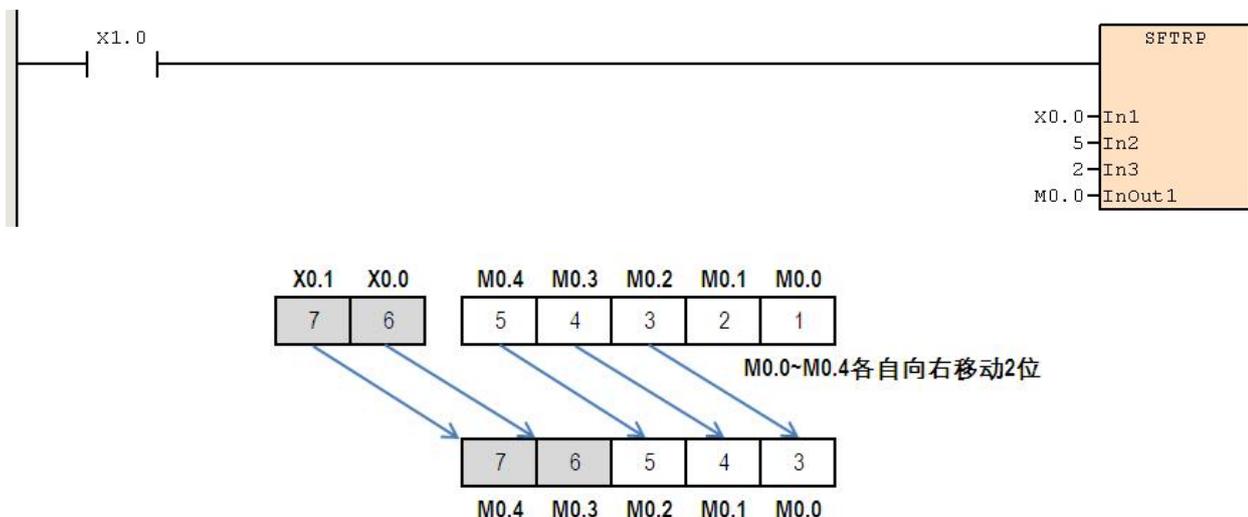
**指令功能说明:**

1, 将 InOut1 开始的起始编号, 具有 In2 个数字 (移位寄存器长度) 的位装置, 以 In3 位个数来右移, 而 In1 开始起始编号以 In3 位个数移入 InOut1 中来填补位空位。

**程序范例:**

例:

以下程序, 执行的内容为, 将 M0.0~M0.4 组成的 5 位数, 向右位移 2 位, 即 M0.2->M0.0, M0.3->M0.1, M0.4->M0.2, 之后将 X0.0~X0.1 放入空白中, 即 X0.0->M0.3, X0.1->M0.4, 组成新的 M0.0~M0.4 的 5 个位数。



### 6.3.6 SFTL (位左移)

类型	助记符	P	位左移																								
			In1(b)					In2(W)					In3(W)					InOut1(b)									
字	SFTL	■	位装置										字/双字装置														
双字			X	Y	S	M	T	C	H	L/	E	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	E
In1	■	■	■	■	■	■	■																				
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	
In3										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	
InOut1		■	■	■																							

**参数说明:**

- 1, In1: 移位装置的起始编号;
- 2, In2: 将移位的数据长度;
- 3, In3: 一次移位的位数;
- 4, InOut1: 将移位装置的起始编号。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (SFTLP) ;
- 2, In2 操作数范围 = 1~1024; In3 操作数范围 = 1~In2, 以及装置要在位范围内。

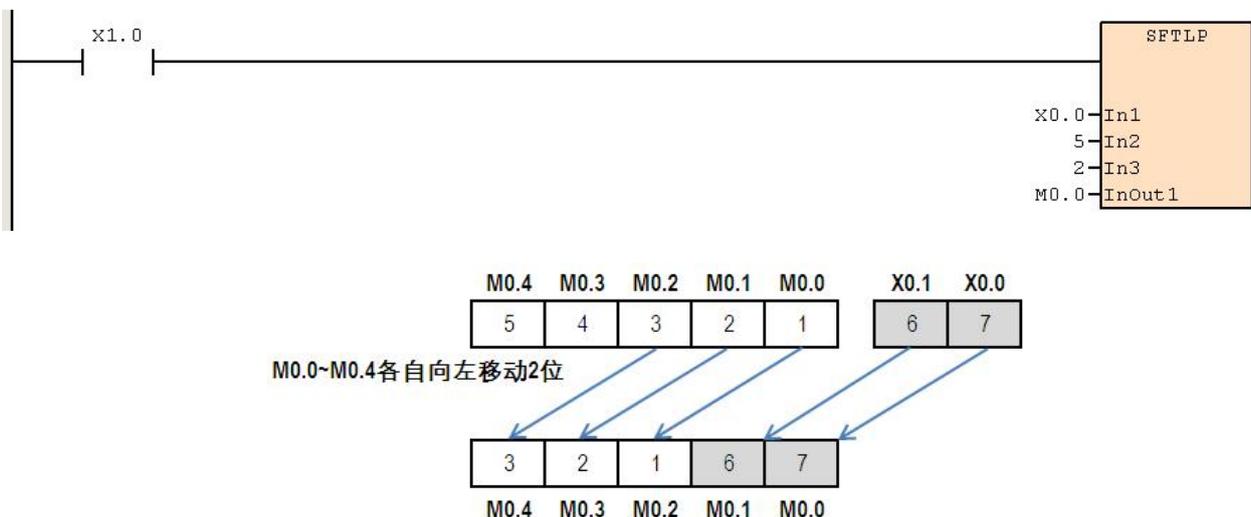
**指令功能说明:**

1, 将 InOut1 开始的起始编号, 具有 In2 个数字 (移位寄存器长度) 的位装置, 以 In3 位个数来左移, 而 In1 开始起始编号以 In3 位个数移入 InOut1 中来填补位空位。

**程序范例:**

例:

以下程序, 执行的内容为, 将 M0.0~M0.4 组成的 5 位数, 向左移位 2 位, 即 M0.2->M0.4, M0.1->M0.3, M0.0->M0.2, 之后将 X0.0~X0.1 放入空白中, 即 X0.1->M0.1, X0.0->M0.0, 组成新的 M0.0~M0.4 的 5 个位数。



### 6.3.7 WSFR (字右移)

类型	助记符	P	字右移																					
			In1(W)					In2(W)					In3(W)					InOut1(W)						
字	WSFR	■																						
双字																								
变量	位装置									字/双字装置														
	X	Y	S	M	T	C	H	L/	E/	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/
			M			C	C	L			M				C	C	I	Q		D		D		
In1									■	■	■	■	■	□			■	■	■	■	■	■		
In2									■	■	■	■	■	□			■	■	■	■	■	■	■	■
In3									■	■	■	■	■	□			■	■	■	■	■	■	■	■
InOut1										■	■	■	■	□				■	■	■	■			

**参数说明:**

- 1, In1: 移位装置的起始编号;
- 2, In2: 将移位的数据长度;
- 3, In3: 一次移位的字数;
- 4, InOut1: 将移位装置的起始编号。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (WSFRP) ;
- 2, In2 操作数范围 = 1~512; In3 操作数范围 = 1~In2, 以及装置要在字范围内。

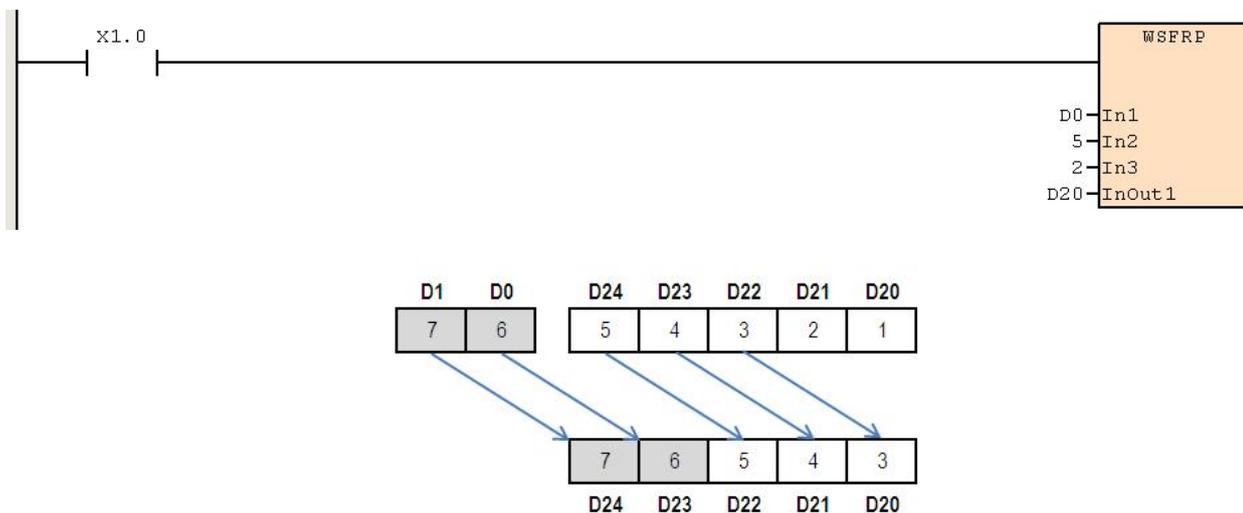
**指令功能说明:**

1, 将 InOut1 开始的起始编号, 具有 In2 个数字 (移位寄存器长度) 的字装置, 以 In3 字数来右移, 而 In1 开始起始编号以 In3 字数移入 InOut1 中来填补字空位。

**程序范例:**

例:

以下程序, 执行的内容为, 将 D20~D24 组成的 5 个字数据, 向右移动 2 个字, 即 D22->D20, D23->D21, D24->D22, 之后将 D0~D1 放入空白中, 即 D0->D23, D1->D24, 组成新的 D20~D24 的 5 个字资料。



### 6.3.8 WSFL (字左移)

类型	助记符	P	字左移																						
字	WSFL	■	In1(W)					In2(W)					In3(W)					InOut1(W)							
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	L
In1										■	■	■	■	■	□		■	■	■	■	■	■			
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In3										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
InOut1											■	■	■	■	□			■	■	■	■				

**参数说明:**

- 1, In1: 移位装置的起始编号;
- 2, In2: 将移位的数据长度;
- 3, In3: 一次移位的字数;
- 4, InOut1: 将移位装置的起始编号。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (WSFLP) ;
- 2, In2 操作数范围 = 1~512; In3 操作数范围 = 1~In2, 以及装置要在字范围内。

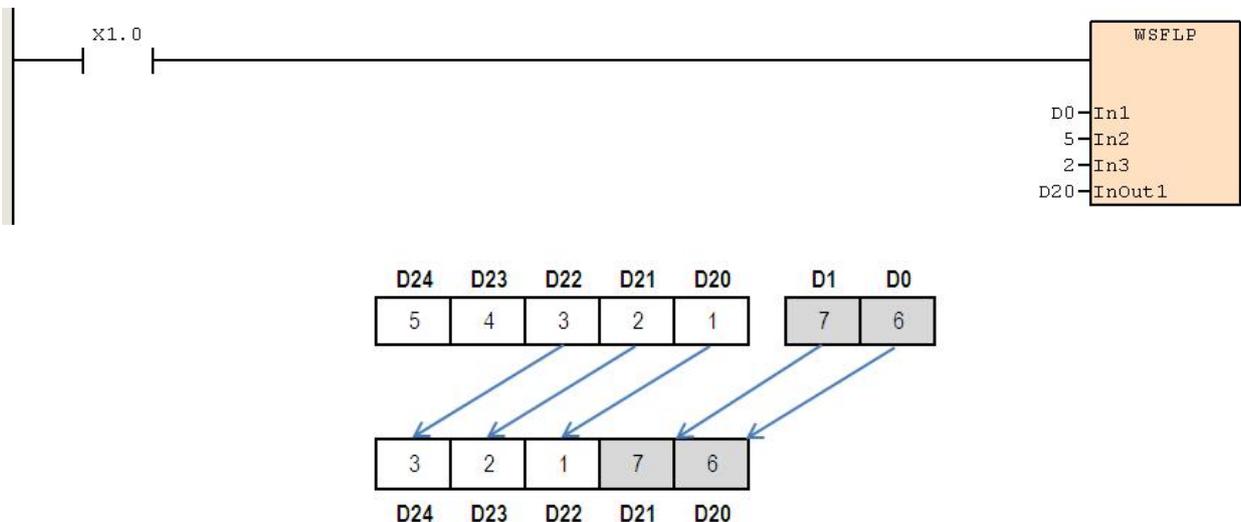
**指令功能说明:**

1, 将 InOut1 开始的起始编号, 具有 In2 个数字 (移位寄存器长度) 的字装置, 以 In3 字数来左移, 而 In1 开始起始编号以 In3 字数移入 InOut1 中来填补字空位。

**程序范例:**

例:

以下程序, 执行的内容为, 将 D20~D24 组成的 5 个字数据, 向左移动 2 个字, 即 D22->D24, D21->D23, D20->D22, 之后将 D0~D1 放入空白中, 即 D1->D21, D0->D20, 组成新的 D20~D24 的 5 个字资料。



### 6.3.9 SFWR (移位写入)

类型	助记符	P	移位写入																							
字	SFWR	■	In1(W)									In2(W)									InOut1(W)					
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
InOut1										■	■	■	■	□			■	■	■	■						

**参数说明:**

- 1, In1: 移位写入资料的个数;
- 2, In2: 移位写入数据的装置;
- 3, InOut1: 写入数据的装置起始编号。

**指令执行注意点:**

- 1, In1 的取值范围为 2~512, 若 InOut1 的内容值为负值, 则自动将 InOut1 的值设置为 0;
- 2, 本指令一般都是使用脉冲执行型指令 (SFWRP)。

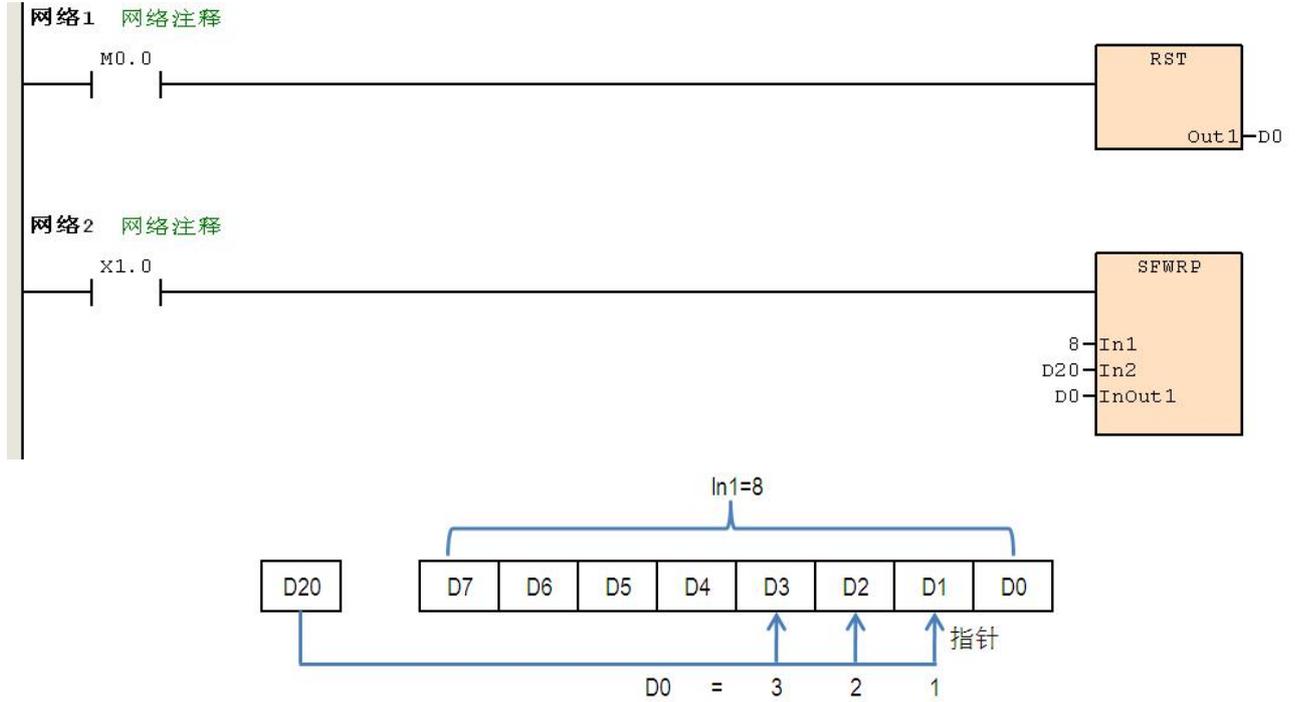
**指令功能说明:**

- 1, 将 InOut1 起始编号开始 In1 个字装置的数据队列定义为先入先出数据队列, 以第一个编号装置作为计数指针, 当指令执行时, 会先判断计数指标内容是否超过 In1-1, 如果超过则不再处理, 同时**进位标志信号 SM50.4**置起; 如果没超过, 则将计数指标内容加 1, 并将 In2 所指定的装置数值写入先入先出数据队列中由计数指针内容指定的位置。

**程序范例:**

**例:**

SFWRP 8 D20 D0, 先将 D0 的内容值清 0, 之后执行该指令时, 先判断 D0 是否大于等于  $(8-1) = 7$ ,  $D0=0 < 7$ , 则进行指令处理, 将 D0 内容值加 1, 则  $D0 = 1$ , 后将 D20 的内容值放入  $D(0+D0)$ 中, 即  $D0 = 1$ , 则放入  $D(0+1) = D1$  中, 再下一次执行该指令时, 同样, 将 D0 内容值加 1, 则  $D0 = 2$ , 后将 D20 的内容值放入  $D(0+D0) = D2$  中, 一直到  $D0 \geq 7$  为止。即先将 D0 内容清 0, 当 X1.0 从 Off→On 变化七次后指令不再执行, D20 的值被依次送入 D1 到 D7 共  $8-1=7$  个数中, 而最后 D0 的值为 7。



### 6.3.10 SFRD (移位读出)

类型	助记符	P	移位读出																						
字	SFRD	■	In1(W)							InOut1(W)							Out1(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■		
InOut1										■	■	■	■	□			■	■	■	■					
Out1										■	■	■	■	□			■	■	■	■					

**参数说明:**

- 1, In1: 数据队列长度;
- 2, InOut1: 数据队列起始编号;
- 3, Out1: 数据队列移位读出装置。

**指令执行注意点:**

- 1, In1 的取值范围为 2~512, 若 InOut1 的内容值为负值, 则自动将 InOut1 的值设置为 0;
- 2, 本指令一般都是使用脉冲执行型指令 (SFRDP)。

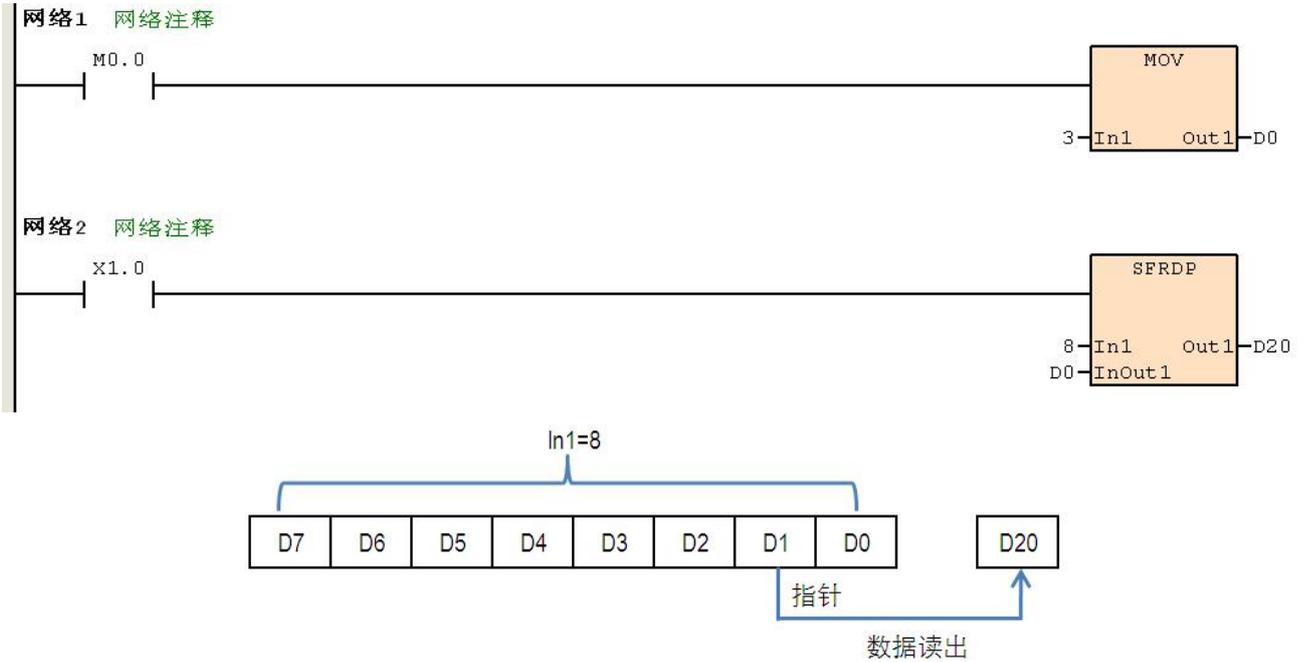
**指令功能说明:**

- 1, 将 InOut1 起始编号开始 In1 个字符装置的数据队列定义为先入先出数据队列, 执行该指令时, 先判断 InOut1 指定的内容是否等于 0, 若为 0, 本指令不再处理数据读出的动作, **零标志信号** SM50.2 置起; 若不为 0, 将 InOut1 内容值减 1 之后, (InOut1+1) 所指定的装置其内容值会写入 Out1 所指定的位置, 接着 (InOut1+In1-1) ~ (InOut1+2) 全部右移一个寄存器, (InOut1+In1-1) 的内容清 0。

**程序范例:**

**例:**

SFRDP 8 D0 D20, 先设置 D0 的内容值为 3, 之后执行该指令时, 先判断 D0 是否为 0, 非 0 则进行指令处理, 将 D0 内容值减 1, 则 D0 = 2, 后将 D (0+1) =D1 的内容值放入 D20 中, 然后再将 D(0+8-1)=D7~D2 全部右移一个字, D7 内容清 0, 再下一次执行该指令时, 同样, 将 D0 内容值减 1, 则 D0 = 1, 后将 D1 的内容值放入 D20 中, 然后再将 D(0+8-1)=D7~D2 全部右移一个字, D7 内容清 0, 一直到 D0=0 为止。即设置 D0 的内容值为 3, 当 X1.0 从 Off→On 变化三次后指令不再执行, D1~D3 的值依次送入 D20, D4~D7 的值全部右移 3 位, D5~D7 的值全部清 0。



## 6.4 数据处理指令

### 6.4.1 WToD (字转双字)

类型	助记符	P	批次复位																							
字	WToD	■	In1(W)									Out1(D)														
双字																										
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	L	
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■
Out1											■	■	■				■	■	■	■					■	

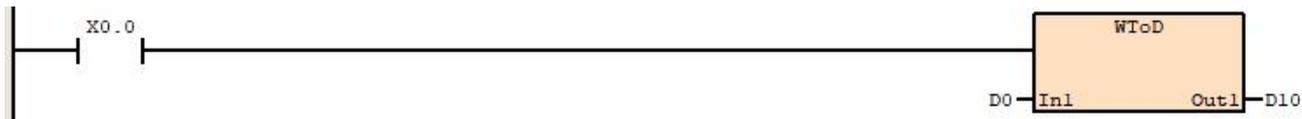
#### 指令说明:

- 1, In1: 要转换的字装置;
- 2, Out1: 得到转换后的双字装置;
- 3, 该指令功能, 将 In1 指定的字装置数据, 转换到 Out1 指定的双字装置中。

#### 程序范例:

##### 例:

当 X0.0=On 时, 将 D0 中 16 位值转换为 32 位值, 高 16 位存于 D11, 低 16 位存于 D10 内。





### 6.4.2 DToW (双字转字)

类型	助记符	P	批次复位																					
字	DToW	■	In1(D)										Out1(W)											
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■				■	■	■	■				■	

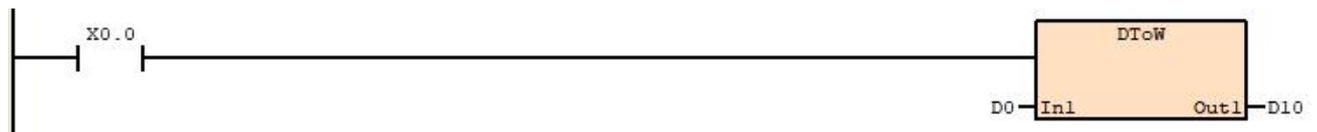
**指令说明:**

- 1, In1: 要转换的双字装置;
- 2, Out1: 得到转换后的字装置;
- 3, 该指令功能, 将 In1 指定的双字装置数据, 转换到 Out1 指定的字装置中;
- 4, 如果双字装置中的数据超出字装置的范围, 将得到不确定的异常数据。

**程序范例:**

例:

当 X0.0=On 时, 将 D0、D1 中 32 位值转换为 16 位值, 存于 D10 内。



### 6.4.3 INT (浮点数->整数变换)

类型	助记符	P	浮点数->整数变换																				
字	INT	■	In1(D)										Out1(W)										
双字	DINT		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

- 1, In1: 欲变换的来源装置;
- 2, Out1: 变换的结果。

**指令执行注意点:**

- 1, 本指令动作与 FLT 指令刚好相反;
- 2, 变换结果为 0 时, **零标志位** SM50.2 置起;
- 3, 变换结果有小数被舍弃时, **借位标志** SM50.3 置起;
- 4, 变换结果若超出 BIN 整数范围时, **进位标志** SM50.4 置起。

**指令功能说明:**

- 1, In1 所指定的寄存器内容以二进制浮点数值态被变换成整数寄存于 Out1 所指定的寄存器当中, 浮点数中小数被舍弃。

**程序范例:**

例:

当 X0.0=On 时, 将二进制浮点数(D1, D0) 变换成整数将结果存放至(D10)当中, 浮点数中的小数被舍弃。



### 6.4.4 FLT (整数->浮点数转换)

类型	助记符	P	BIN 整数->浮点数转换																							
字	FLT	■	In1(W)												Out1(D)											
双字	DFLT		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
	In1								■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■				■			

**指令说明:**

- 1, In1: 变换来源装置; Out1: 存放变换结果的装置;
- 2, 将 16 位 BIN 整数或 32 位 BIN 整数变换成浮点数值;
- 3, 若转换结果为 0, 则零标志位 SM50.2 置起。

**程序范例:**

**例:**

当 X0.0=On 时, 将 D0(内为 BIN 整数)变换成 D11、D10(二进浮点数值)。  
 若 D0=10, 则 X0.0=On, 转换后浮点数的 32-bit 数值为 16#41200000, 存于 32-bit 寄存器 D10(D11)内。



### 6.4.5 ZRST (批次复位)

类型	助记符	P	批次复位																				
字	ZRST	■	Out1(b)										Out2(b)										
双字	ZRST		Out1(W)										Out2(W)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
Out1		■	■	■	■	■	■			■	■	■					■	■	■	■			
Out2		■	■	■	■	■	■			■	■	■					■	■	■	■			

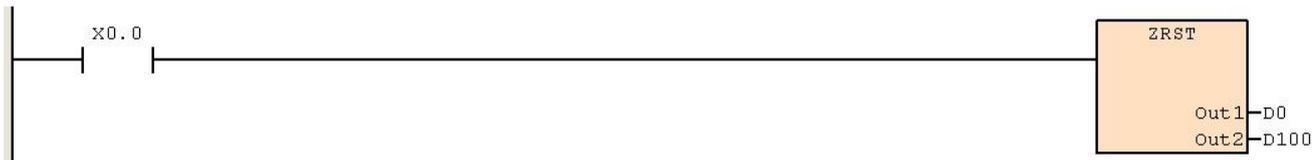
**指令说明:**

- 1, Out1: 批次复位起始装置; Out2: 批次复位结束装置;
- 2, Out1 操作数编号 <= Out2 操作数编号, 若 Out1 编号 > Out2 编号, 只有 Out2 编号被清除;
- 3, Out1 和 Out2 操作数必须**同时指定**位装置或**同时指定**字装置。

**程序范例:**

**例:**

当 X0.0 为 On 时, 资料寄存器 D0 ~ D100 资料被清除为 0。



### 6.4.6 DECO (译码器)

类型	助记符	P	译码器																				
字	DECO	■	In1(W)							In2(W)							Out1(b)						
双字																							
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■
Out1		■	■	■																			

**参数说明:**

- 1, In1: 译码来源装置;
- 2, In2: 译码位长度;
- 3, Out1: 存放译码结果装置。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (DECOP) ;
- 2, In2 的范围为 1~8, 当 In2 = 8 时, 可做最大解码 256 点。

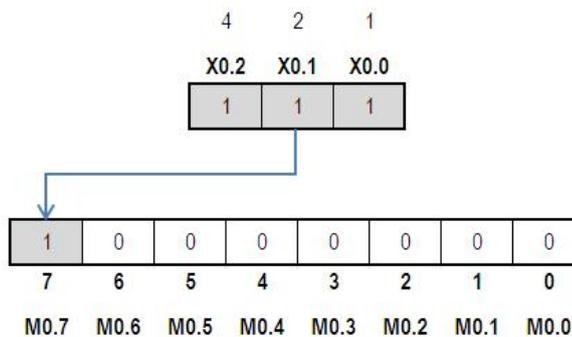
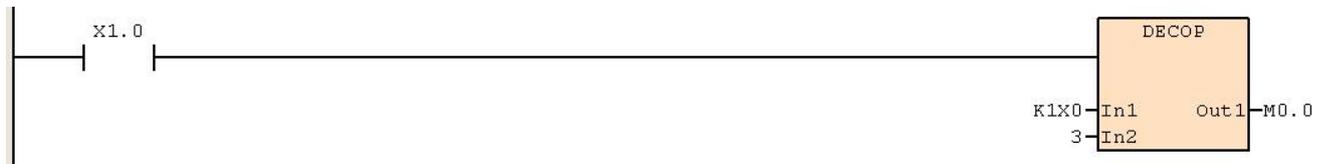
**指令功能说明:**

- 1, 来源装置 In1 的低位连续 In2 位作译码, 并将其 2 的 In2 次方长度的结果存于 Out1 起始编号装置中。

**程序范例:**

**例:**

- 指令 DECOP K1X0 3 M0.0, 执行时, 将 X0 组中的低三位 X0.0~X0.2 组成的 3 位 BIN 值 7 (假设 X0.0~X0.2 均为 ON), 则 M0.0~M0.7 连续 8 个点, 分别视为 0~7 的 BIN 数值, 执行结果是, 将 M0.0~M0.6 置为 OFF, M0.7 置位 ON。
- X1.0 = Off→On 时, DECO 指令将 X0.0~X0.2 的内容值译码到 M0.0~M0.7。
- 当 DECO 指令执行过后, 而 X1.0 变为 Off, 已经做译码输出者照常动作。



### 6.4.7 ENCO (编码)

类型	助记符	P	编码																						
字	ENCO	■	In1(b)							In2(W)							Out1(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/		X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	
			M			C	C	E	L			M				C	C	I	Q		D				L
In1	■	■	■	■	■	■	■																		
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	
Out1											■	■	■	■	□			■	■	■	■			■	

**参数说明:**

- 1, In1: 编码来源装置;
- 2, In2: 编码位长度;
- 3, Out1: 存放编码结果装置。

**指令执行注意点:**

- 1, 本指令一般都是使用脉冲执行型指令 (ENCOP) ;
- 2, In2 的范围为 1~256;
- 3, 取 In1 的最高 bit 位的第一个为 1 的位做 BIN 值; 若都没有位为 1 时, 则 ENCO 指令执行失败标志位 SM50.6 置起。

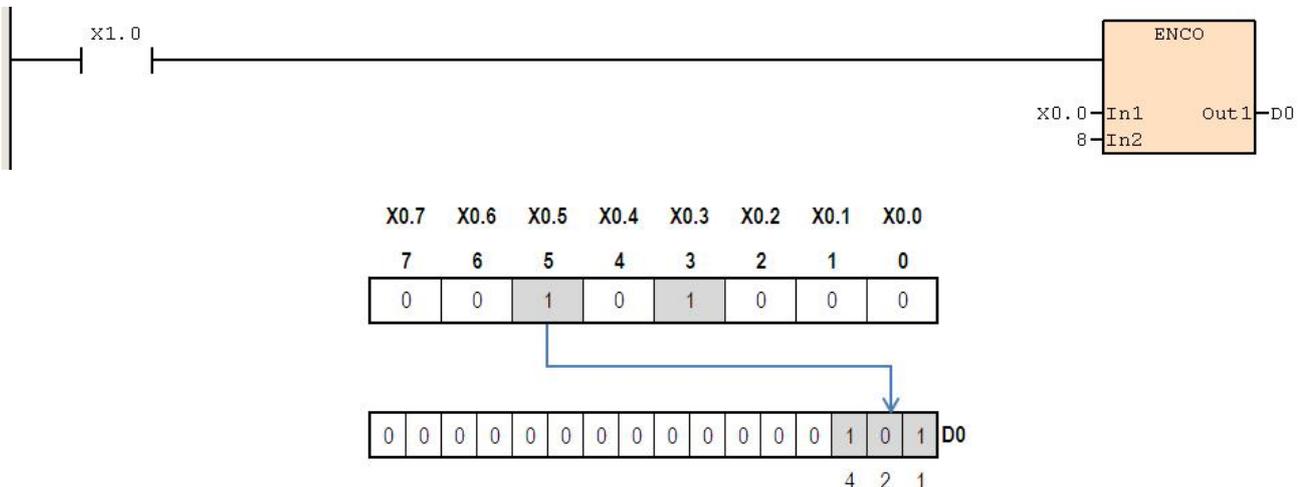
**指令功能说明:**

- 1, 将编码来源装置 In1 位的装置数作编码, 结果存于装置 Out1 中。

**程序范例:**

**例:**

- 指令 ENCOP X0.0 8 D0, 执行时, 将以 X0.0为起始编号的, 连续8位数目的位装置, 即 X0.0~X0.7的状态, 分别视为0~7的 BIN 数值, 其中若 X0.5, X0.3为1, 取最高位为1的 BIN 值, 即取 X0.5对应 BIN 数值5, 存放入 D0, 即 D0 = 5。
- 当 X1.0 = Off→On 时, ENCO 指令将 2^3 位数据 (X0.0~X0.7) 编码存放于 D0 的下位 3 位(b2~b0)内, D0 中未被使用的位(b15~b3)全部变为 0。
- 当 ENCO 指令执行过后, 而 X1.0 变为 Off 后, D0 内数据不变。



### 6.4.8 SUM (ON 位数量统计)

类型	助记符	P	ON 位数量统计																							
字	SUM	■	In1(W)												Out1(W)											
双字	DSUM		In1(D)												Out1(W)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□			■	■	■				■			

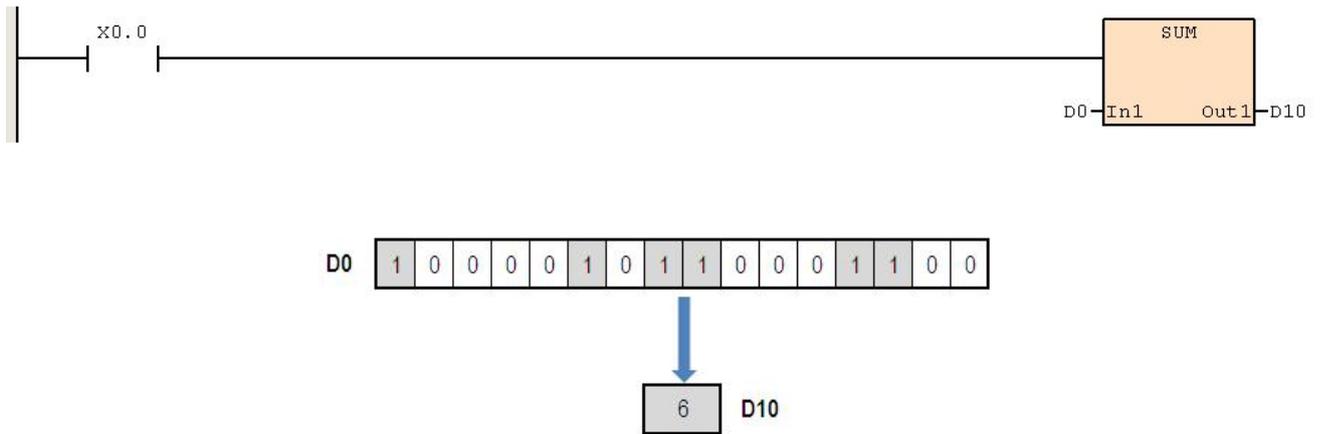
**指令说明:**

- 1, In1: 来源装置; Out1: 存放计数值的目的装置;
- 2, 该指令会将来源装置中所有为 ON 的位数统计, 并放入 Out1 中;
- 3, 如果来源装置 In1 的全部位都为 off 时, 零标志信号 SM50.2 置起 ON。

**程序范例:**

**例:**

当 X0.0 为 On 时, D0 的 16 个位中, 内容为“1”的位总数被存于 D10 当中。



### 6.4.9 BON (ON 判定)

类型	助记符	P	ON 判定 (取特定位)																						
字	BON	■	In1(W)							In2(W)							Out1(b)								
双字	DBON		In1(D)							In2(W)							Out1(b)								
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1								■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In2								■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
Out1		■	■	■			■																		

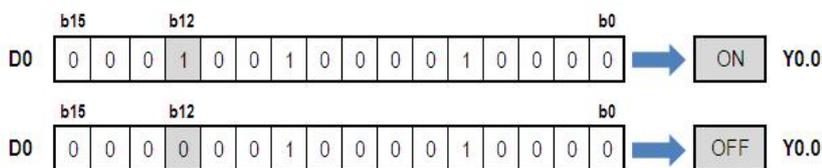
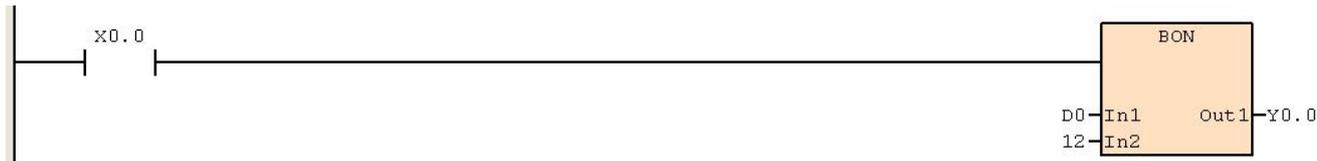
**指令说明:**

- 1, In1: 来源装置; In2: 指定判定的位 (自 0 开始编号); Out1: 存放判定结果的装置;
- 2, 该指令可将来源装置 In1 中, 指定的 In2 位, 读取出来并放入 Out1 所指定的装置中;
- 3, 16 位指令 In2 的范围为 0~15, 32 位指令为 0~31。

**程序范例:**

**例:**

当 X0.0=On 时, 若是 D0 的第 12 个位为“1”时, Y0.0=On, 为“0”时, Y0.0=Off。  
X0.0 变成 Off 时, Y0.0 仍保持之前的状态。





### 6.4.10 MEAN (计算平均值)

类型	助记符	P	计算平均值																						
字	MEAN	■	In1(W)									In2(W)									Out1(W)				
双字	DMEAN		In1(D)									In2(W)									Out1(D)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□	□	■	■	■	■	■	■				
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□	□		■	■	■	■			■		

**参数说明:**

- 1, In1: 将取平均值的起始装置;
- 2, In2: 取平均值的装置个数;
- 3, Out1: 存放平均值的装置。

**指令执行注意点:**

- 1, 如果计算中出现余数时, 余数会被舍去;
- 2, 如果指定的装置号码超过字装置范围时, 超出部分各装置值按 0 计算;
- 3, In2 的取值范围是 1~64。

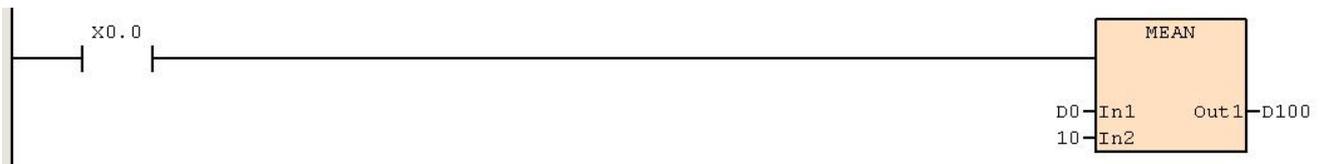
**指令功能说明:**

- 1, 将 In1 起始的 In2 个装置内容值相加后取平均值存入 Out1 中。

**程序范例:**

**例:**

当 X0.0=On 时, D0 开始算的 10 个(In2 = 10)寄存器的内容全部相加, 相加之后再除以 10 以求得平均值并存储于指定的 D100 当中, 余数被舍去。



### 6.4.11 SQR（开平方根）

类型	助记符	P	开平方根																				
字	SQR	■	In1(W)										Out1(W)										
双字	DSQR		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■

**参数说明：**

- 1, In1: 将开平方根的来源装置；
- 2, Out1: 存放结果的装置。

**指令执行注意点：**

- 1, In1 只可以指定正数，若指定负数时，指令不执行；
- 2, 运算结果 Out1 只求整数，小数点被舍弃，有小数点被舍弃时，借位标志信号 SM50.3 置起；
- 3, 运算结果 Out1 为 0 时，零标志信号 SM50.2 置起。

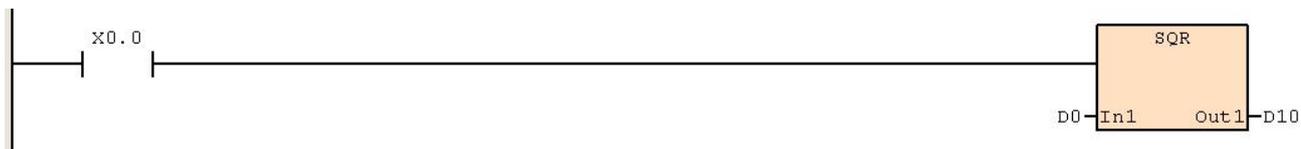
**指令功能说明：**

- 1, 将 In1 所指定的装置内容值开平方根后，存放于 Out1 所指定的装置中。

**程序范例：**

**例：**

当 X0.0=On 时，将 D0 内容值开平方根后，存放于 D10 内。



## 6.5 便利指令

### 6.5.1 SER (数据检索)

类型	助记符	P	数据检索																						
字	SER	■	In1(W)						In2(W)						In3(W)						Out1(W)				
双字	DSER		In1(D)						In2(D)						In3(W)						Out1(W)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1								■	■	■	■	■	□	□	■	■	■	■	■	■					
In2								■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In3								■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
Out1									■	■	■	■	□			■	■	■	■						

**参数说明:**

- 1, In1: 被检索的数据的起始装置;
- 2, In2: 欲检索的数据;
- 3, In3: 被检索的数据的装置个数;
- 4, Out1: 存放检索结果的起始装置。

**指令执行注意点:**

- 1, 其中 In3 的取值范围是 1~256; Out1 连续占用 5 个寄存器装置;
- 2, Out1 中, 数据的编号从 0 开始计算, 例如共 10 个数字, 则编号分别为 0~9; 检索结果中, 所有相等值的数据数量记录在 Out1 装置中, 第一个相等值的编号记录在 Out1+1 装置中, 最后一个相等值的编号记录在 Out1+2 装置中, 最小值的编号记录在 Out1+3 装置中, 最大值的编号记录在 Out1+4 装置中。

**指令功能说明:**

- 1, 以 In2 所指定的数据对 In1 起始的共 In3 个被检索的数据进行检索, 检索结果存放于 Out1 所指定的连续 5 个寄存器当中。

程序范例:

例:

- 当 X0.0=On 时, 由 D0~D9 组成的数据区块与 100 作比较, 结果存放在 D20~D22 中, 当相等值不存在时, D20~D22 的内容全部为 0。
- 大小比较以代数型态进行。(-10 < 2)
- 所有比较数据的最小值编号记录在 D23, 最大值编号记录在 D24。当最小值最大值不只一个时, 会记录编号大者。



In1	内容值	比较数据	数据编号	结果
D0	88	In2  100	0	
D1	100		1	相等
D2	110		2	
D3	150		3	
D4	100		4	相等
D5	300		5	
D6	100		6	相等
D7	5		7	最小
D8	100		8	相等
D9	500	9	最大	

Out1	内容值	说明
D20	4	相等值的数据个数
D21	1	第一个相等值的编号
D22	8	最后一个相等值的编号
D23	7	最小值的编号
D24	9	最大值的编号



### 6.5.2 ABSD (绝对方式凸轮控制)

类型	助记符	P	绝对方式凸轮控制																							
字	ABSD		In1(W)						In2(W)						In3(W)						Out1(b)					
双字	DABSD		In1(D)						In2(D)						In3(W)						Out1(b)					
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■					
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■			
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
Out1		■	■	■																						

#### 参数说明:

- 1, In1: 比较表起始装置;
- 2, In2: 欲比较的数值数据;
- 3, In3: 多段比较的组数;
- 4, Out1: 比较结果的起始装置。

#### 指令执行注意点:

- 1, DASBD 指令的 In2 也可指定高速计数器。但是, 在这种使用方式下, 高速计数器的现在值与设置值作比较时, 还是会受到扫描周期的影响而无法作实时性的输出;
- 2, In3 的取值范围在 0~64。

#### 指令功能说明:

- 1, ABSD 指令为对应计数器的现在值产生多个输出波形的指令, 通常用来做绝对方式凸轮控制;
- 2, 例如, 指令 ABSD D0 C0 3 M0.0, 正常时, 将以 Out1 装置偏置为偶数的为比较下限, 偏置为奇数的为比较上限, 即 D0, D2, D4 为下限, D1, D3 及 D5 为上限进行比较。

如果下限值小于等于上限值时:

若 C0 在第一组对应范围内, 即  $D0 \leq C0 \leq D1$ , 则 M0.0 置 ON, 否则置 OFF;

第二组和第三组同理, 在范围内 M0.1, M0.2 则 ON, 否则置 OFF。

如果上限值小于下限值时:

例如第一组  $D0 > D1$ , 那么若 C0 在第一组比较时,  $D1 \leq C0 \leq D0$  时, M0.0 为 OFF, 反之则置 ON;

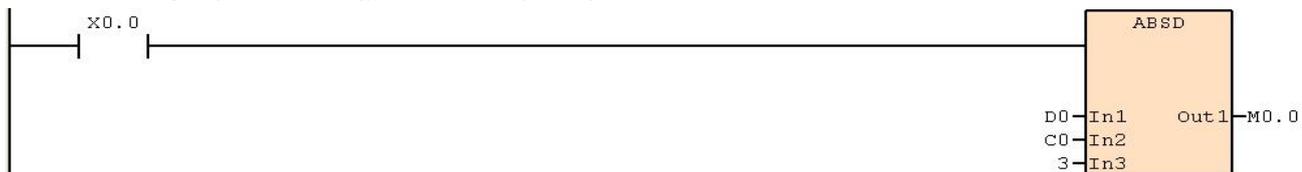
第二组第三组同理, 在范围内 M0.1, M0.2 置 OFF, 反之则置 ON。

程序范例:

例:

当 X0.0=On 时, 计数器 C0 的现在值与 D0~D5 等 3 组上下限值作区间比较, 结果分别反应在 M0.0~M0.2。

X0.0=Off 时, 原 M0.0~M0.2 的 On/Off 状态不会变化。



在大于等于下限值且小于等于上限值范围内对应的 M0.0~M0.2 会 On。

下限值	上限值	C0 现在值	输出
D0=30	D1=100	$30 \leq C0 \leq 100$	M0.0=On
D2=110	D3=250	$110 \leq C0 \leq 250$	M0.1=On
D4=10	D5=80	$10 \leq C0 \leq 80$	M0.2=On

若下限值大于上限值时, 则小于上限值( $C0 < 60$ )或大于下限值( $C0 > 140$ )时, M0.2=On。

下限值	上限值	C0 现在值	输出
D0=30	D1=100	$30 \leq C0 \leq 100$	M0.0=On
D2=110	D3=250	$110 \leq C0 \leq 250$	M0.1=On
D4=140	D5=60	$60 \leq C0 \leq 140$	M0.2=Off

### 6.5.3 INCD (相对方式凸轮控制)

类型	助记符	P	相对方式凸轮控制																							
字	INCD		In1(W)						In2(W)						InOut1(W)						Out1(b)					
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	AI	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
InOut1														□												
Out1		■	■	■																						

**参数说明:**

- 1, In1: 比较表起始装置;
- 2, In2: 多段比较的组数;
- 3, InOut1: 计数器编号;
- 4, Out1: 比较结果的起始装置。

**指令执行注意点:**

- 1, InOut1 的装置只可取范围 C0~C198, 连续占用两个计数器, InOut1 需使用 CNT 指令驱动, InOut1+1 不要使用 CNT 指令;
- 2, In2 的取值范围在 1~64;
- 3, In2 的组数比较完成后, **指令执行完毕标志** SM51.3 会置 On, 至下一个该指令时置 Off;
- 4, 该指令的参数可随时修改, 能流 Off 时, 参数保持不变, 需要用户自行进行复位。

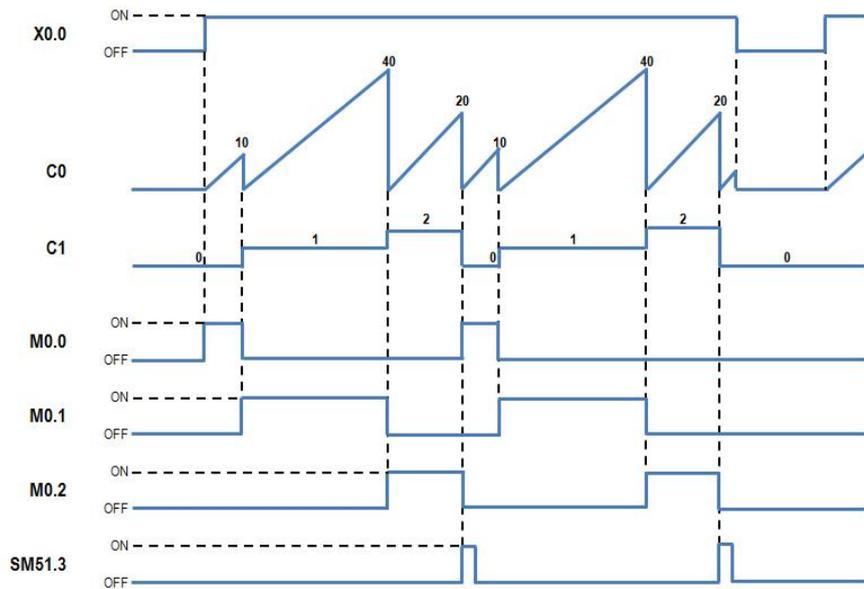
**指令功能说明:**

- 1, INCD 指令为用一对计数器产生多个输出波形的指令, 通常用来做相对方式凸轮控制;
- 2, InOut1 的现在值与 In1 的设置值作比较, 每到达一组设置值, InOut1 的现在值被复位为 0 重新计数。而目前执行的段数被暂存于 InOut1+1 的计数器当中。

程序范例：

例：

- D0~D2 表示比较目标值；C0 表示现在值；C1 表示已经执行的组数；M0.0~M0.2 对应 D0~D2 表示正在执行的组，当任意一组正在执行时，M0.0~M0.2 当中一点对应 On。
- 其中 D0 = 10, D1 = 40, D2 = 20, 那么指令执行时，C0 被复位为 0, M0.0 被置 ON, M0.1 及 M0.2 为 OFF, 当到达第一组数值即 D0 = 10 时，C0 被复位为 0, 同时 M0.0 被置 OFF, 而 M0.1 被置 On, 同时 C1 = 1, 表明已执行完一组比较；当 C0 继续执行到达第二组数值即 D1 = 40 时，C0 被复位为 0, 同时 M0.0 和 M0.1 被置 OFF, 而 M0.2 被置 ON, C1 = 2；继续执行，到达第三组数据 D2 = 20 时，C0、C1 被复位为 0, M0.0、M0.1 和 M0.2 都被置 OFF, SM51.3 置 On。以此循环执行；如果能流中断，那么停止执行，并且 C0, C1, M0.0, M0.1 和 M0.2 都将保持不变。



### 6.5.4 TTMR (示教式定时器)

类型	助记符	P	示教式定时器																							
字	TTMR		In1(W)								In2(W)								Out1(W)							
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	AI	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□			■	■	■	■						

**参数说明:**

1, In1: 倍数设置; In2: 该指令的使用组编号; Out1: 储存按钮开关 On 时间的装置编号;

**指令执行注意点:**

- In2: 该指令最多可使用 8 组编号, 取值 0~7, 若编号重复, 则两个指令是共同使用, 以后 On 的为计时, 先 Off 的为关闭;
- 倍数设置:
  - In1 = 0, Out1 以秒为单位;
  - In1 = 1, Out1 乘以 10 倍以 100ms 为单位;
  - In1 = 2, Out1 乘以 100 倍以 10ms 为单位。
- 当能流 Off 时, 该指令停止执行, Out1+1 内容复位为 0, Out1 内容没有变化。

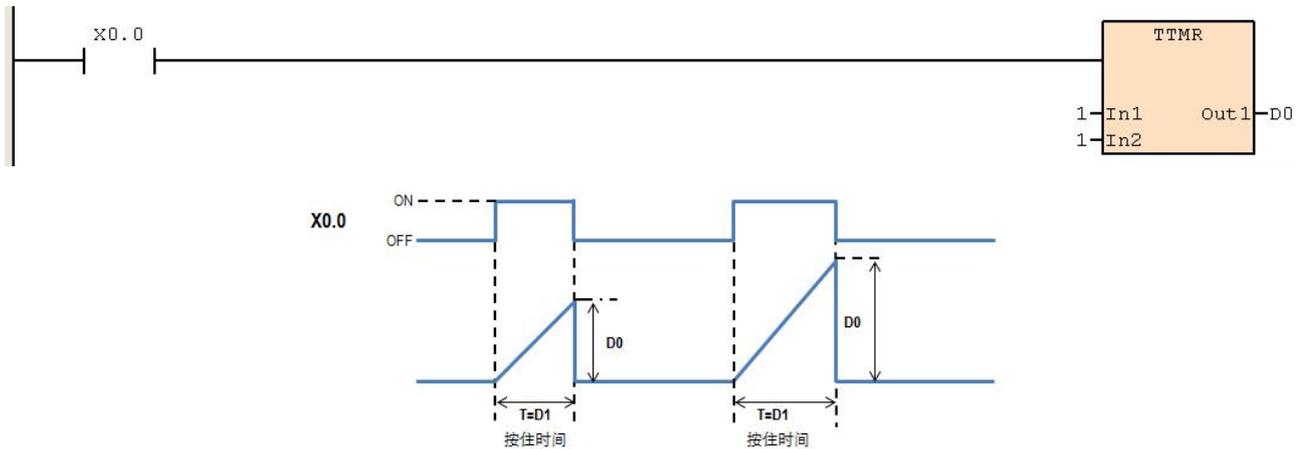
**指令功能说明:**

- 将外接的按钮开关 On 的时间以 100ms 为单位存放于 Out1+1 的装置内, 而开关 On 时间以秒为单位乘以 In1 倍数后存放于 Out1 装置内。

程序范例：

例：

按钮开关 X0.0 被按住时间 (X0.0 的 On 时间) 被存入于 D1 当中，由 In1 来指定该时间的倍数，并将位数时间存入于 D0 当中。由此，可使用按钮开关来调整定时器的设置值。当 X0.0 变成 Off 时，D1 的内容被复位为 0，但是 D0 内容没有变化。



假设 X0.0 的 On 时间为 T 秒，而 D0、D1 与 In1 之间的关系如下表。

In1	D0	D1(单位: 100 ms)
0 (单位: s)	$1 \times T$	$D1 = D0 \times 10$
1 (单位: 100ms)	$10 \times T$	$D1 = D0$
2 (单位: 10ms)	$100 \times T$	$D1 = D0 / 10$

### 6.5.5 STMR (特殊定时器)

类型	助记符	P	特殊定时器																							
字	STMR		In1(W)								InOut1(b)								Out1(b)							
双字																										
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/	L/	X	Y	S	M	T	C	H	AI	A	D	S	E	N	O	L/		
			M				C	E	L			M				C		Q		D						
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■		
InOut1					■																					
Out1		■	■	■																						

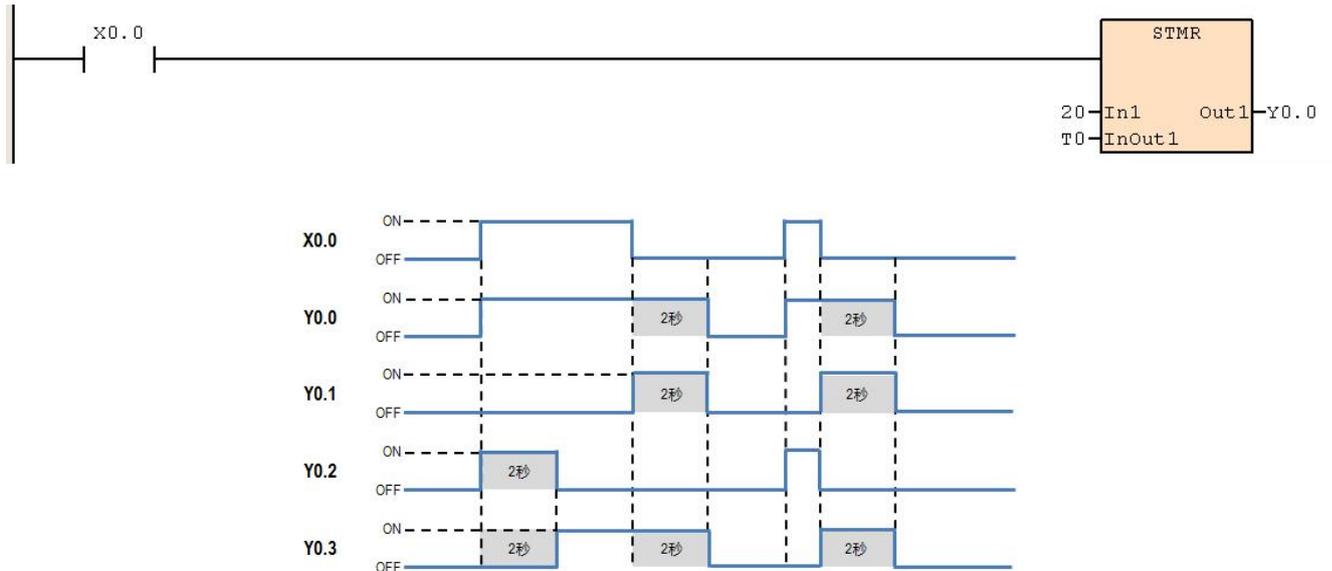
**指令说明:**

- 1, In1: 定时器设置值, 单位以定时器为准; InOut1: 定时器编号, 不可使用变址; Out1: 输出装置的起始编号;
- 2, STMR 指令, 用来产生 Off 延迟, 一次触发及闪烁回路的专用指令;
- 3, STMR 指令与 TMR 指令不可重复使用同一个定时器, 为上/下边沿触发指令, 计入边沿指令;
- 4, Out1 占用连续 4 个装置。

**程序范例:**

例:

LD X0.0, STMR 20 T0 Y0.0, 则 Y0.0 为 Off 延迟接点, 当 X0.0 由 Off 变成 On 时, Y0.0 = On, 到 X0.0 由 On 变成 Off 时, 延迟 2 秒后 Y0.0 = Off; Y0.1 于 X0.0 由 On 变成 Off 时, 做一次 2 秒 Y0.1 = On 输出; Y0.2 于 X0.0 由 Off 变成 On 时, 做一次 2 秒 Y0.2 = On 输出; Y0.3 于 X0.0 由 Off 变成 On 时, 延迟 2 秒后 Y0.3 = On, 到 X0.0 由 On 变成 Off 时, 延迟 2 秒后 Y0.3 = Off。



### 6.5.6 RAMP (斜坡信号)

类型	助记符	P	斜坡信号																						
字	RAMP		In1(W)							InOut1(W)							InOut2(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/		X	Y	S	M	T	C	H	AI	A	D	S	E	N	O	L/	
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
InOut1											■	■	■	■	□			■	■	■	■				■
InOut2											■	■	■	■	□			■	■	■	■				■

**参数说明:**

- 1, In1: 倾斜信号的终点设置;
- 2, InOut1: 扫描次数设定;
- 3, InOut2: 倾斜信号的移动点装置。

**指令执行注意点:**

- 1, InOut1: 扫描次数设定为 0~32767, 当设定为 0 或小于 0 时, 指令不执行; 本指令是一个求斜率的指令, 斜率与扫描周期有绝对的关系, 因此使用本指令时, 通常必须预先将扫描周期加以固定;
- 2, 当指令能流 Off 时, 指令执行停止, InOut1 和 InOut2 保持不变;
- 3, 当指令执行完毕, 即扫描次数为 0 或设定小于 0 时, RAMP 指令**执行完毕标志** SM51.4 置起直到下一该指令扫描。

**指令功能说明:**

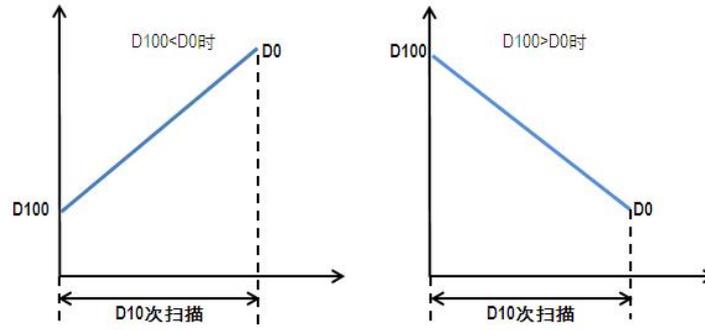
- 1, 预先将倾斜信号的起点设置值写入 InOut2 中, 倾斜信号的终点设置值写入 In1 中, 当指令执行时 InOut2 的设置值朝 In1 增加或减少, 每经过一次扫描, InOut1 数值减 1, 减到 0 时, 到达 In1 设定值。

**程序范例:**

例:

当 X0.0=On 时, D100 的值朝 D0 增加或减少, 每经过一次扫描, 扫描次数 D10 数值减 1, 减到 0 时, 到达 D0 设定值。





### 6.5.7 SORT (资料整理排序)

类型	助记符	P	资料整理排序																					
			In1(W)							In2(W)					In3(W)				In4(W)				Out1(W)	
字	SORT	■	位装置							字/双字装置														
双字			X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O
In1									■	■	■	■	■	□		■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In4									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□			■	■	■	■	■			

**参数说明:**

- 1, In1: 原始数据区块的起始装置;
- 2, In2: 每组数据的字数;
- 3, In3: 被排序的数据组数;
- 4, In4: 数据排序的组编号;
- 5, Out1: 存放排序结果数据区块的起始装置。

**指令执行注意点:**

- 1, 排序结果显示在, Out1 所指定的起始号码开始算起的 In2 \* In3 个寄存器当中, 因此 Out1 与 In1 指定同一个寄存器的话, 排序结果将与会替换原来被排序的数据 In1 内容;
- 2, In2 的取值范围为 1~32, In3 的取值范围为 1~6, In4 的取值范围为 1~In3。

**指令功能说明:**

- 1, 当 In4 为 1~In3 时, 对对应的字段进行排序, **排序模式**根据 SM51.2 进行安排, SM51.2 为 Off 时, 从小到大排序, On 时从大到小排序。

程序范例:

例:

In2 为行数, In3 为列数, In4 为指定要整理排序的列



排序数据构成例

	1	2	3	4
1	(D0)1	(D5)90	(D10)75	(D15)66
2	(D1)2	(D6)55	(D11)65	(D16)54
3	(D2)3	(D7)80	(D12)98	(D17)89
4	(D3)4	(D8)70	(D13)60	(D18)99
5	(D4)5	(D9)95	(D14)79	(D19)75

D30=3 时的排序后数据

	1	2	3	4
1	(D50)1	(D55)90	(D60)60	(D65)66
2	(D51)2	(D56)55	(D61)65	(D66)54
3	(D52)3	(D57)80	(D62)75	(D67)89
4	(D53)4	(D58)70	(D63)79	(D68)99
5	(D54)5	(D59)95	(D64)98	(D69)75

D30=4 时的排序后数据

	1	2	3	4
1	(D50)1	(D55)90	(D60)75	(D65)54
2	(D51)2	(D56)55	(D61)65	(D66)66
3	(D52)3	(D57)80	(D62)98	(D67)75
4	(D53)4	(D58)70	(D63)60	(D68)89
5	(D54)5	(D59)95	(D64)79	(D69)99

## 6.6 外部设定显示

### 6.6.1 TKY (十键键盘输入)

类型	助记符	P	十键键盘输入																					
字	TKY		In1(b)							InOut1(W)							InOut2(b)							
双字	DTKY		In1(b)							InOut1(D)							InOut2(b)							
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	AI	A Q	D	S D	E	N D	O	L/ E L	
In1	■	■	■	■	■	■																		
InOut1										■	■	■	■	□	□		■	■	■	■				■
InOut2		■	■	■																				

#### 参数说明:

- 1, In1: 按键输入起始装置;
- 2, InOut1: 按键输入值存放处;
- 3, InOut2: 按键输出信号。

#### 指令执行注意点:

- 1, In1 操作数会占用连续 10 点, InOut2 操作数会占用连续 11 点;
- 2, 当对应按键被按下后, 在别的按键被按之前, InOut2 对应的点位将会一直保持 ON;
- 3, 任何一个按键被按下时, InOut2 中的第 11 点都将 ON, 并保持到所有按键松开;
- 4, 当能流值为 Off 时, InOut1 内的值不会变化, InOut2 的第 11 点将会 Off。

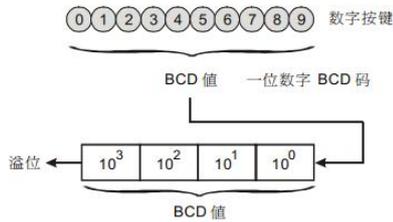
#### 指令功能说明:

- 1, 本指令指定由 In1 开始的 10 个输入点, 依序代表 BCD 值数字的 0~9。这 10 个输入点分别接上 10 个按键, 依据这 10 个按键被压下的先后顺序可输入 4 位 BCD 值数字 0~9999 (16 位指令) 或 8 位 BCD 值数字 0~99999999 (32 位指令), 并将输入的数值存放在 InOut1, 而 InOut2 则存放键盘的按键情形, 其中 InOut1 的超出 9999 或 99999999 的数值将会溢出;

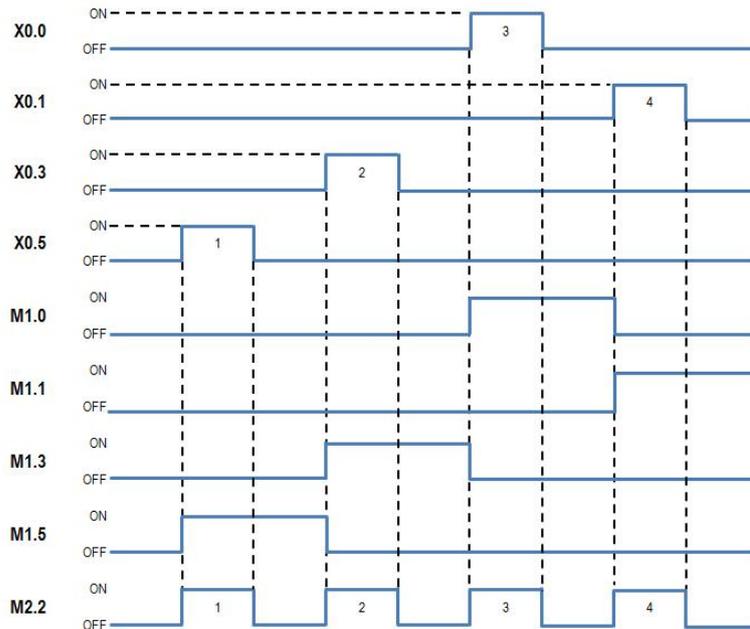
**程序范例:**

**例:**

指令指定 X0.0 开始的 10 个输入端与 0~9 的 10 个按键连接，当 X2.0=On 时，指令执行，将键盘输入的数值以 BCD 值的形态存入 D0 中，而按键的情况则放在 M1.0~M2.1。



- 如下列时序图所示，连接于数字键盘 X0.5、X0.3、X0.0、X0.1 的 4 个按键以 1、2、3、4 的顺序作打入的动作，结果为 16#5,301 被暂存于 D0 当中。当 X0.0~X1.1 当中任何一个按键被按下时，M1.0~M2.1 当中一点对应 On。任何一个按键被按下时，M2.2=On。
- 当条件接点 X2.0 变成 Off 时，D0 之前的值无变化，但是，M1.0~M2.2 全部变成 Off。



### 6.6.2 HKY (十六键键盘输入)

类型	助记符	P	十六键键盘输入																				
			In1(b)				InOut1(W)				InOut2(b)				Out1(b)								
字	HKY		In1(b)				InOut1(D)				InOut2(b)				Out1(b)								
双字	DHKY		In1(b)				InOut1(D)				InOut2(b)				Out1(b)								
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1	■																						
InOut1										■	■	■	□	□		■	■	■	■				
InOut2		■	■	■																			
Out1		■																					

**参数说明:**

- 1, In1: 按键扫描输入起始装置;
- 2, InOut1: 按键输入值存放处;
- 3, InOut2: 按键输出信号;
- 4, Out1: 按键扫描输出起始装置。

**指令执行注意点:**

- 1, In1 操作数会占用连续 4 点, Out1 操作数会占用连续 4 点, InOut2 操作数会占用连续 17 点;
- 2, 当对应按键被按下后, 在别的按键被按之前, InOut2 对应的点位将会一直保持 On;
- 3, 任何一个按键被按下时, InOut2 中的第 17 点都将 On, 并保持到所有按键松开;
- 4, 该指令需要能流维持, 当能流值为 Off 时, 指令停止, InOut1 内的值不会变化, InOut2 的第 17 点将会 Off;
- 5, 该指令为每 20ms 扫描一行, 因此按下按钮的时间必须不小于 80ms, 扫描时间可以在 SD100 中进行设置, 默认为 20ms;
- 6, 该指令分别在所有程序**只可出现一次**, 否则不可正常使用, 使用时对应 HKY 正在使用标志 SM51.0 或 DHKY **正在使用标志** SM51.1 置 On, 当执行完毕或能流 Off 时, SM51.0 或 SM51.1 置 Off。

**指令功能说明:**

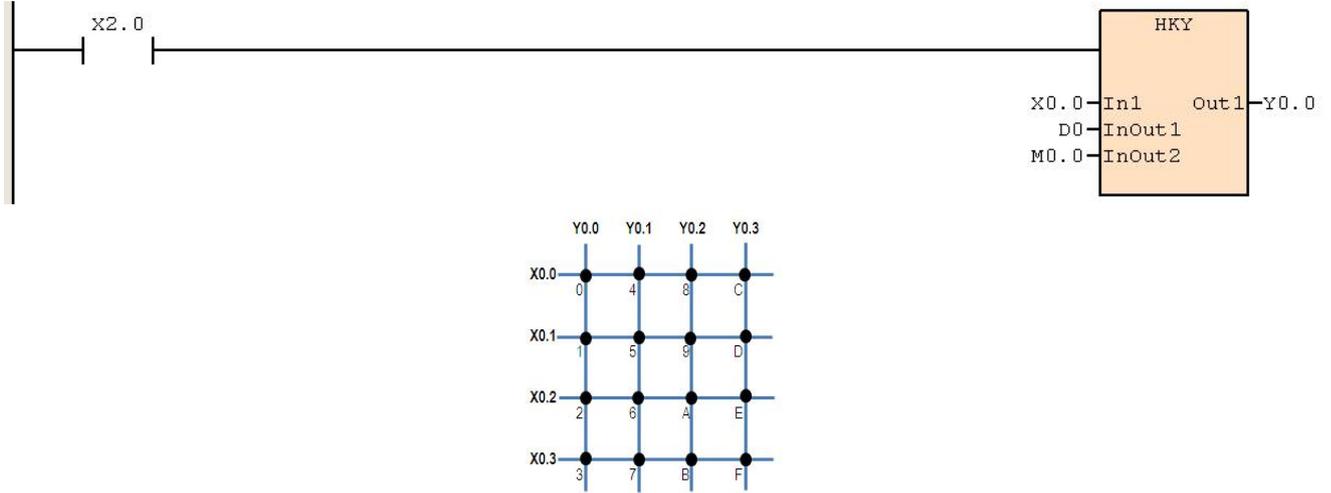
- 1, 本指令指定由 In1 开始的 4 个外部输入点以及由 Out1 开始的连续 4 个外部输出点以矩阵扫描的方式构成 16 键的键盘。键盘输入的数值存放在 InOut1 中, 而 InOut2 则存放上一次键盘的按键情形, 如果有数个按键同时被按下时, 以先按者优先;
- 2, 由数字键盘锁打入的值被暂存于 InOut1 当中, 使用 16 位指令 HKY 时, InOut1 最大可容纳十六进制数值 FFFF, 超过 4 位数时, 最前面的位数溢出。使用 32 位指令 DHKY 时, InOut1 最大可容纳十六进制数值 FFFFFFFF, 超过 8 位数时, 最前面的位数溢出。



**程序范例:**

**例:**

- 指令指定 X0.0~X0.3 等 4 个输入端与 Y0.0~Y0.3 等 4 个输出端构成扫描 16 键的键盘。当 X2.0=On 时,指令执行,Y0.0~Y0.3 输出端按顺序持续输出,若外部数字键盘锁打入值为 6,则 Y0.1=On, X0.2=On,将数值 6 存放在 D0 中,且 M0.6=On。
- 当条件接点 X2.0 变成 Off 时,D0 之前的输入值无变化,但是 M0.0~M1.6 全部变成 Off。



### 6.6.3 SEGD (七段显示器解码)

类型	助记符	P	七段显示器解码																						
字	SEGD	■	In1(W)										Out1(b)												
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	L
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Out1		■	■	■																					

**指令说明:**

- 1, In1: 欲译码的来源装置; Out1: 译码后的输出起始装置;
- 2, 本指令将 In1 的低 4 位 (b0~b3) 的内容 (0~F: 16 进制) 被解码成 7 段显示器输出, 解码的结果存放于 Out1 开始的连续 7 位当中;
- 3, Out1 操作数会占用连续 7 点。

**程序范例:**

**例:**

- 当 X0.0=On 时, D0 的下位 4 个位 (b0~b3) 的内容 (0~F: 16 进制) 被解码成 7 段显示器输出, 解码的结果暂存于 Y0.0~Y0.6 当中。若指定数据超出 4 个位, 仍取下位 4 个位的内容解码。
- Y0.0~Y0.6 分别对应 a~g。当 X0.0=On, D0=2 时, abdeg 置 On, 即 Y0.0=On, Y0.1=On, Y0.3=On, Y0.4=On, Y0.6=On。



7 段显示器译码表:

16进制	位组合	七段显示器的构成	各节点状态							显示值
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	7
8	1000		ON	8						
9	1001		ON	ON	ON	ON	OFF	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	b
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	c
D	1101		OFF	ON	ON	ON	ON	OFF	ON	d
E	1110		ON	OFF	OFF	ON	ON	ON	ON	e
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	f

### 6.6.4 ASC (ASCII 码转换)

类型	助记符	P	ASCII 码转换																							
字	ASC	■	In1(W)												Out1(W)											
双字	DASC		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1																							■			
Out1										■	■	■	■	□	□		■	■	■	■				■		

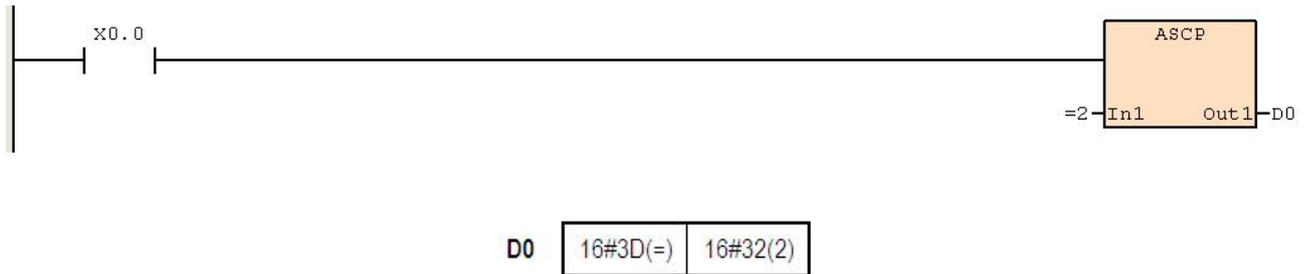
**指令说明:**

- 1, In1: 欲执行 ASCII 码变换的字符; Out1: 存放 ASCII 码转换后的装置;
- 2, In1 强制认为是字符信息, 例如填写 D10, 那么分辨为三个字符'D', '1'和'0';
- 3, 当使用 16 位指令 ASC 时, In1 最多只可填入 2 个字符, 使用 32 位指令 DASC 时, In1 最多只可填入 4 个字符, 最先输入的字符转换后放于 Out1 中的最高字节中;
- 4, 若要放入字符串, 必须多次使用才可完成目的。

**程序范例:**

例:

当 X0.0=On 时, 将字符“=2”变换成 ASCII 码后暂存于 D0 当中。



## 6.7 浮点运算

### 6.7.1 DECMP (浮点数比较)

类型	助记符	P	浮点数比较																					
字																								
双字	DECMP	■	In1(D)						In2(D)						Out1(b)									
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1		■	■	■																				

**参数说明:**

1, In1: 二进制浮点数比较值; In2: 二进制浮点数被比较值; Out1: 比较结果, 占用连续 3 点。

**指令执行注意点:**

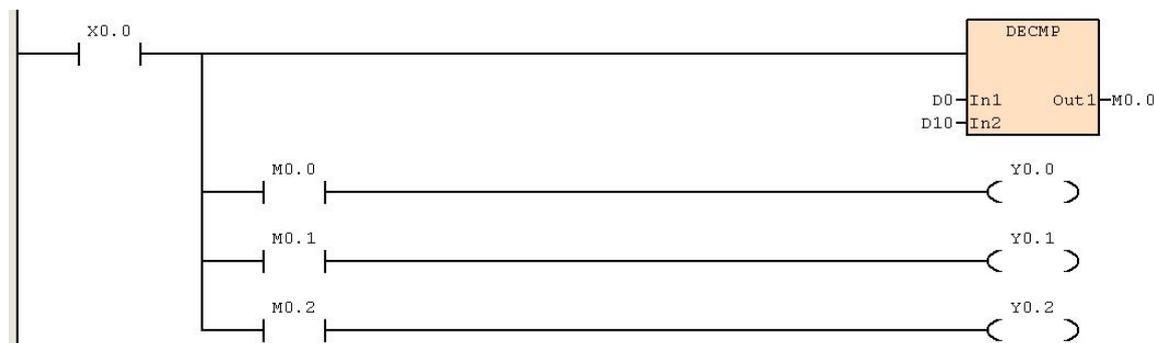
- 1, In1 或 In2 来源操作数若是指定常数, 指令会将该常数转换成二进制浮点数值作比较;
- 2, 如果浮点数无法正常表达, 则**浮点数无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

**指令功能说明:**

1, 二进制浮点数值 In1 与 In2 作比较, 比较结果 (>、=、<) 在 Out1 作表示。

**程序范例:**

- 例:**
- 指定装置为 M0.0, 则自动占有 M0.0~M0.2。
  - 当 X0.0=On 时, DECMP 指令执行, M0.0~M0.2 其中之一会 On, 当 X0.0=Off 时, DECMP 指令不执行, M0.0~M0.2 状态保持在 X0.0=Off 之前的状态。
  - 若需要得到 ≥、≤、≠的结果时, 可将 M0.0~M0.2 串并联即可取得。



- (D1, D0)>(D11, D10)时 Y0.0 为 On
- (D1, D0)=(D11, D10)时 Y0.1 为 On
- (D1, D0)<(D11, D10)时 Y0.2 为 On



### 6.7.2 DEZCP (浮点数区间比较)

类型	助记符	P	浮点数区间比较																														
字		■																															
双字	DEZCP		In1(D)								In2(D)								In3(D)								Out1(b)						
变量	位装置									字/双字装置																							
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L										
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■									
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■									
In3									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■									
Out1		■	■	■																													

#### 参数说明:

1, In1: 区间比较的二进制浮点数下限值; In2: 区间比较的二进制浮点数上限值; In3: 二进制浮点数比较值; Out1: 比较结果, 占用连续 3 点。

#### 指令执行注意点:

- 1, In1 或 In2 来源操作数若是指定常数的话, 指令会将该常数转换成二进制浮点数值作比较;
- 2, 当二进制浮点数下限值 In1 大于上限值 In2 时, 则指令以下限值 In1 作为上下限值进行比较。
- 3, 如果浮点数无法正常表达, 则**浮点数无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

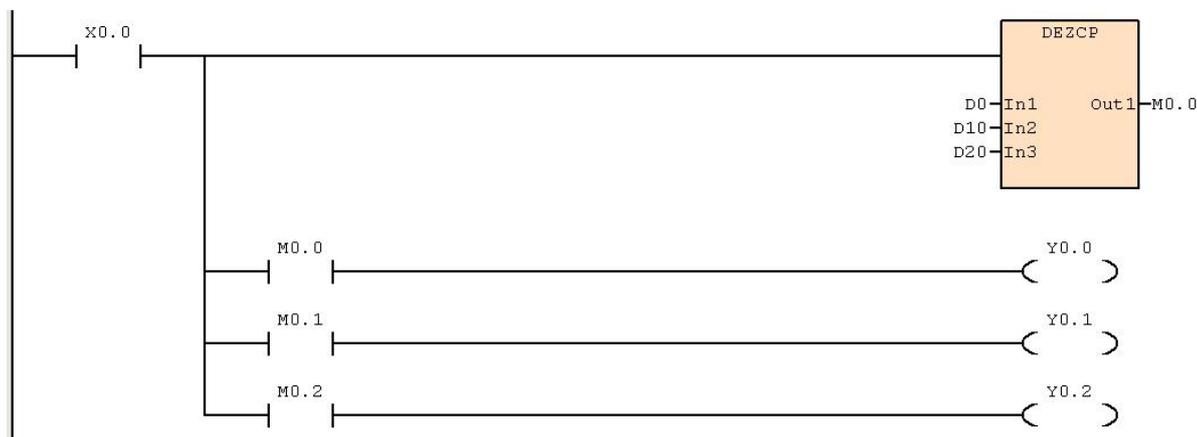
#### 指令功能说明:

1, 二进制浮点数比较值 In3 与二进制浮点数下限值 In1 及二进制浮点数上限值 In2 作比较, 其比较结果在 Out1 作表示, 例如 Out1 被指定为 M0.0, In1 大于 In3 时, M0.0 置起,  $In1 \leq In3 \leq In2$  时, M0.1 置起,  $In2 < In3$  时, M0.2 置起, 对应装置置起同时, 将会置 off 另两个装置。

程序范例:

例:

- 指定装置为 M0.0, 则自动占有 M0.0~ M0.2。
- 当 X0.0=On 时, DEZCP 指令执行, M0.0~M0.2 其中之一会 On, 当 X0.0=Off 时, DEZCP 指令不执行, M0.0~M0.2 状态保持在 X0.0=Off 之前的状态。



$(D1, D0) > (D21, D20)$  时 Y0.0 为 On

$(D1, D0) \leq (D21, D20) \leq (D11, D10)$  时 Y0.1 为 On

$(D21, D20) > (D11, D10)$  时 Y0.2 为 On

### 6.7.3 DMOVR (浮点数数据搬移)

类型	助记符	P	浮点数数据搬移																							
字		■																								
双字	DMOVR		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
Out1										■	■	■	■	■	■		■	■	■				■			

**参数说明:**

- 1, In1: 浮点数资料来源;
- 2, Out1: 数据传送目的地。

**指令执行注意点:**

- 1, 当该指令执行时, 将 In1 的内容直接搬移至 Out1 内。当指令不执行时, Out1 内容不会变化;
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

**指令功能说明:**

- 1, 当 In1 输入为常数时, 指令会将该常数转换成二进制浮点数值作处理。

**程序范例:**

**例:**

32 位浮点数值数据搬移, 须使用 DMOVR 指令。

当 X0.0=Off 时, (D1、D0) 内容没有变化, 若 X0.0=On, 将 2222 自动转换成的浮点数值传送至 (D1、D0) 数据寄存器。



### 6.7.4 DRAD (角度->弧度)

类型	助记符	P	角度->弧度																					
字		■																						
双字	DRAD		In1(D)										Out1(D)											
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■				■

**参数说明:**

- 1, In1: 资料来源 (角度);
- 2, Out1: 变换的结果 (弧度)。

**指令执行注意点:**

- 1, 若转换结果为 0, 则**零标志位** SM50.2 置起。
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

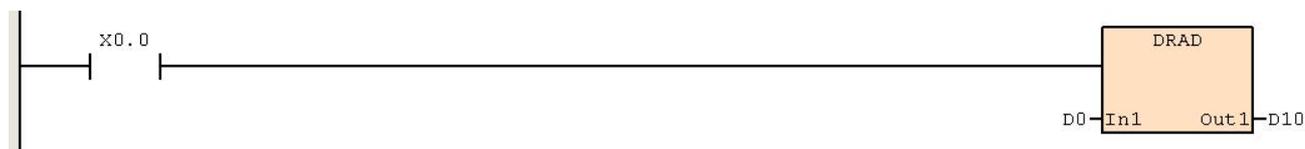
**指令功能说明:**

- 1, 使用下列公式将角度转换成弧度:  $\text{弧度} = \text{角度} * (\text{Pi}/180)$ ,  $\text{Pi} = 3.1415926$ 。

**程序范例:**

**例:**

当 X0.0=On 时, 指定二进制浮点数(D1, D0)的角度值, 将角度转换成弧度值后存于(D11, D10)当中, 内容为二进制浮点数。



### 6.7.5 DDEG (弧度->角度)

类型	助记符	P	弧度->角度																							
字		■																								
双字	DDEG		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■				■			

**参数说明:**

- 1, In1: 资料来源 (弧度);
- 2, Out1: 变换的结果 (角度)。

**指令执行注意点:**

- 1, 若转换结果为 0, 则**零标志位** SM50.2 置起。
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

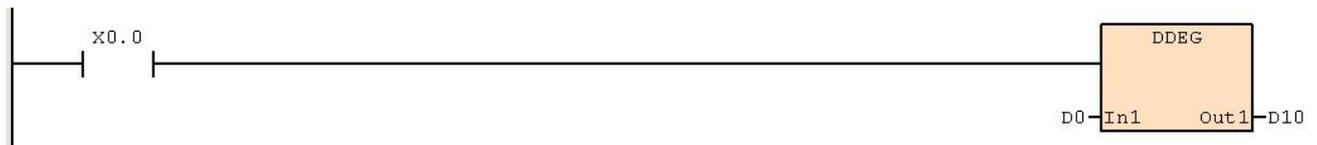
**指令功能说明:**

- 1, 使用下列公式将弧度转换成角度:  $角度 = 弧度 * (180 / \pi)$ ,  $\pi = 3.1415926$ 。

**程序范例:**

**例:**

当 X0.0=On 时, 指定二进制浮点数(D1, D0)的角度值, 将弧度值转换成角度后存于(D11, D10)当中, 内容为二进制浮点数。



### 6.7.6 DEBCD (浮点数->十进制浮点型态)

类型	助记符	P	浮点数->十进制浮点型态																				
字		■																					
双字	DEBCD		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

- 1, In1: 资料来源;
- 2, Out1: 变换的结果。

**指令执行注意点:**

- 1, PLC 是以二进制浮点数作正常浮点数运算, DEBCD 指令就是用来将二进制浮点数变换成十进制浮点型态的专用指令;
- 2, 若转换结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 二进制浮点数不便于使用者直观的察看数据, 可使用十进制浮点型态来察看。十进制浮点形态的表达格式为: 例如, 使用 D1 和 D0 连续两个装置来表达十进制的数-432100 = -4321\*10<sup>2</sup>, 那么将实数部分 -4321 放入 D0 中, 将指数部分 2 放入 D1 再例如表达十进制数据 0.01234 = 1234\*10<sup>-5</sup>, 那么 D0 = 1234, D1 = -5。
- 4, 其中十进制浮点数的范围是, 实数部分-32768~32767。

**指令功能说明:**

- 1, 将 In1 所指定的寄存器以二进制浮点数被变换成十进制浮点型态寄存于 Out1 所指定的连续两个寄存器当中。

**程序范例:**

**例:**

当 X0.0=On 时, D1, D0 内的二进制浮点数被变换成十进制浮点数寄存于 D11, D10 当中。



### 6.7.7 DEBIN (十进制浮点型态->浮点数)

类型	助记符	P	十进制浮点型态->浮点数																							
字		■																								
双字	DEBIN		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
Out1										■	■	■	■	■		■	■	■					■			

**参数说明:**

- 1, In1: 资料来源;
- 2, Out1: 变换的结果。

**指令执行注意点:**

- 1, DEBIN 指令就是用来将十进制浮点型态变换成二进制浮点数的专用指令;
- 2, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

**指令功能说明:**

- 1, 将 In1 所指定的寄存器以十进制浮点数被变换成二进制浮点数寄存于 Out1 所指定的寄存器当中。

**程序范例:**

例:

当 X0.0=On 时, D1、D0 内十进制浮点数被变换成二进制浮点数寄存于 D11、D10 当中。



### 6.7.8 DEADD (浮点数加法)

类型	助记符	P	浮点数加法																						
字		■																							
双字	DEADD		In1(D)							In2(D)							Out1(D)								
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■	■			■		

**参数说明:**

- 1, In1: 被加数;
- 2, In2: 加数;
- 3, Out1: 和。

**指令执行注意点:**

- 1, In1 或 In2 来源操作数若是指定常数的话, 指令会将该常数转换成二进制浮点数值来作加算;
- 2, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

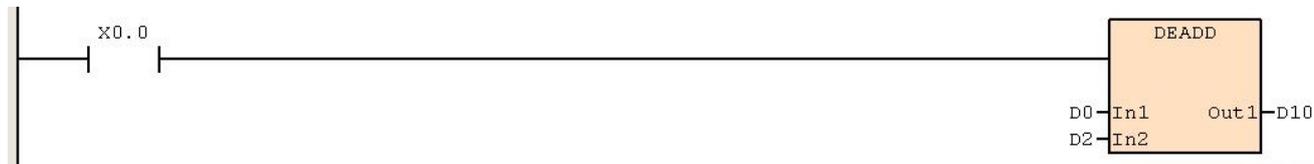
**指令功能说明:**

1, In1 所指定的寄存器内容加上 In2 所指定的寄存器内容, 和被放置于 Out1 所指定的寄存器当中, 加算的动作全部以单精度二进制浮点数值型态进行。

**程序范例:**

例:

当 X0.0=On 时, 将单精度浮点数(D1, D0)+单精度浮点数(D3, D2), 结果存放在(D11, D10)中。



### 6.7.9 DESUB (浮点数减法)

类型	助记符	P	浮点数减法																						
字		■																							
双字	DESUB		In1(D)											In2(D)						Out1(D)					
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■	■			■		

**参数说明:**

- 1, In1: 被减数;
- 2, In2: 减数;
- 3, Out1: 差。

**指令执行注意点:**

- 1, In1 或 In2 来源操作数若是指定常数, 指令会将该常数转换成二进制浮点数值来作减算;
- 2, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

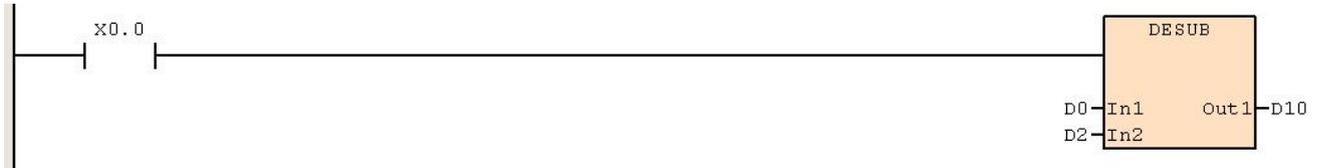
**指令功能说明:**

- 1, In1 所指定的寄存器内容减掉 In2 所指定的寄存器内容, 差被存放至 Out1 所指定的寄存器当中, 减算的动作全部以二进制浮点数型态进行。

**程序范例:**

**例:**

当 X0.0=On 时, 将二进制浮点数(D1, D0)减去二进制浮点数(D3, D2), 结果存放在(D11, D10)中。



### 6.7.10 DEMUL (浮点数乘法)

类型	助记符	P	浮点数乘法																						
字		■																							
双字	DEMUL		In1(D)							In2(D)							Out1(D)								
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■	■			■		

**参数说明:**

- 1, In1: 被乘数;
- 2, In2: 乘数;
- 3, Out1: 积。

**指令执行注意点:**

- 1, In1 或 In2 来源操作数若是指定常数, 指令会将该常数转换成二进制浮点数值来作乘算;
- 2, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

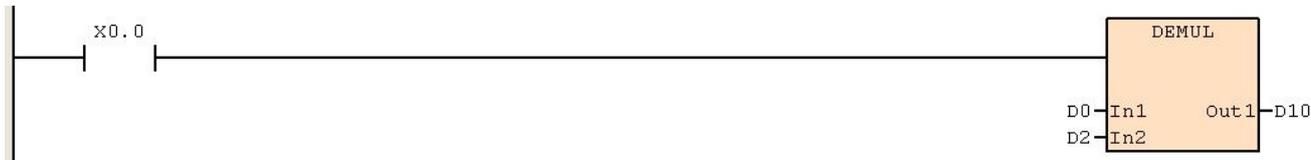
**指令功能说明:**

1, In1 所指定的寄存器内容乘以 In2 所指定的寄存器内容, 积被存放至 Out1 所指定的寄存器当中, 乘算的动作全部以二进制浮点数型态进行。

**程序范例:**

例:

当 X0.0=On 时, 将二进制浮点数(D1, D0)乘上二进制浮点数(D3, D2)将积存放至(D11, D10)所指定的寄存器当中。



### 6.7.11 DEDIV (浮点数除法)

类型	助记符	P	浮点数除法																						
字		■																							
双字	DEDIV		In1(D)									In2(D)						Out1(D)							
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■	■			■		

**参数说明:**

- 1, In1: 被除数;
- 2, In2: 除数;
- 3, Out1: 商。

**指令执行注意点:**

- 1, In1 或 In2 来源操作数若是指定常数, 指令会将该常数转换成二进制浮点数值来作除算;
- 2, 若 In2 的数值为 0, 则指令不执行;
- 3, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 4, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

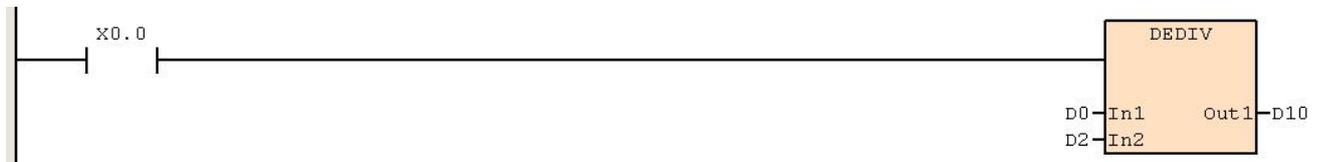
**指令功能说明:**

- 1, In1 所指定的寄存器内容除以 In2 所指定的寄存器内容, 商被存放至 Out1 所指定的寄存器当中, 除算的动作全部以二进制浮点数值型态进行。

**程序范例:**

例:

当 X0.0=On 时, 将二进制浮点数(D1, D0)除以二进制浮点数(D3, D2)将商存放至(D11, D10)所指定的寄存器当中。



### 6.7.12 DEXP (浮点数取指数)

类型	助记符	P	浮点数取指数																				
字		■																					
双字	DEXP		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

- 1, In1: 运算来源装置;
- 2, Out1: 运算结果装置。

**指令执行注意点:**

- 1, In1 内容正负数都有效, 运算时均以浮点数方式执行;
- 2, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

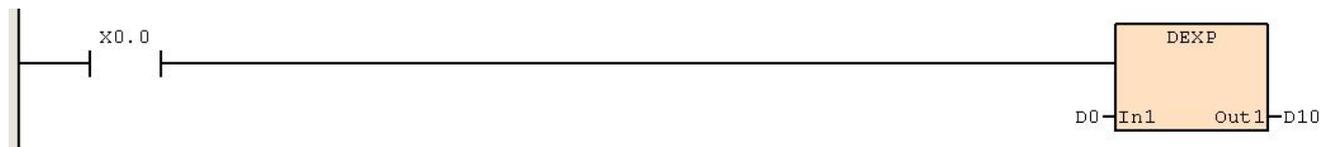
**指令功能说明:**

- 1, 以  $e = 2.71828$  为底数, 运算来源装置 In1 为指数做 EXP 运算:  $Out1 = e^{In1}$ 。

**程序范例:**

例:

当 X0.0 为 On 时, (D1, D0)为指数做 EXP 运算, 其值为二进制浮点数值并存放于(D11, D10)寄存器中。





### 6.7.13 DLN (浮点数取自然对数)

类型	助记符	P	浮点数取自然对数																							
字		■																								
双字	DLN		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■		■	■	■					■			

#### 参数说明:

- 1, In1: 运算来源装置;
- 2, Out1: 运算结果装置。

#### 指令执行注意点:

- 1, In1 内容只有正数有效, 运算时均以浮点数方式执行;
- 2, 若运算结果为 0, 则**零标志位** SM50.2 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

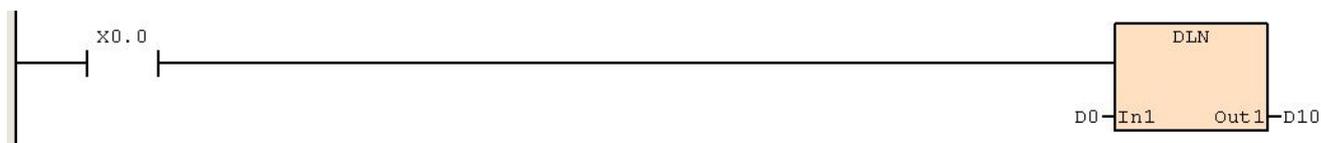
#### 指令功能说明:

- 1, 以运算来源装置 In1 为操作数做自然对数 Ln 运算:  $Out1 = \ln(In1)$ 。

#### 程序范例:

##### 例:

当 X0.0 为 On 时, 将(D0, D1)寄存器为真数做 ln 运算, 其值为二进制浮点数并存放于(D10, D11)寄存器中。



### 6.7.14 DLOG (浮点数取对数)

类型	助记符	P	浮点数取对数																												
字		■																													
双字	DLOG		In1(D)										In2(D)										Out1(D)								
变量	位装置										字/双字装置																				
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L								
In1								■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■								
In2								■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■								
Out1									■	■	■	■	■	■		■	■	■	■			■									

**参数说明:**

- 1, In1: 运算底数装置;
- 2, In2: 运算真数装置;
- 3, Out1: 运算结果装置。

**指令执行注意点:**

- 1, In1 的取值范围必须为正数且不等于 1, In2 内容必须为正数, 运算时均以二进制浮点数方式执行;
- 2, 若运算结果为 0, 则**零标志位** SM50.2 置起;
- 3, 如果浮点数无法正常表达, 则**浮点数无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

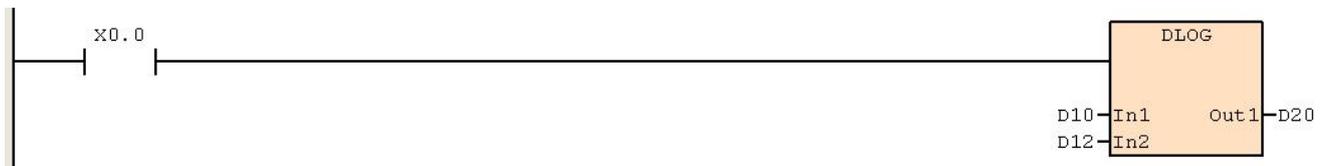
**指令功能说明:**

1, 将 In1 内容及 In2 内容为操作数做 Log 运算:  $Out1 = \log_{In1} In2 = \ln(In2)/\ln(In1)$ 。

**程序范例:**

例:

当 X0.0 为 On 时, 将(D10, D11)及(D12, D13)32 位寄存器二进制浮点数值做 log 运算并将结果存于(D20, D21) 32 位寄存器。



### 6.7.15 DESQR (浮点数开平方根)

类型	助记符	P	浮点数开平方根																							
字		■																								
双字	DESQR		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■		■	■	■					■			

**参数说明:**

- 1, In1: 欲开平方根来源装置;
- 2, Out1: 开平方根的结果。

**指令执行注意点:**

- 1, 开平方的动作全部以二进制浮点数型态进行, In1 若为常数, 则会将该常数转换成二进制浮点数值来作运算;
- 2, 若开平方根的结果为 0 时, **零标志位 SM50.2** 置起;
- 3, 来源运算装置只有正数有效;
- 4, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

**指令功能说明:**

- 1, In1 所指定的寄存器内容被开平方, 所得的结果寄存于 Out1 所指定的寄存器中。

**程序范例:**

**例:**

当 X0.0=On 时, 将二进制浮点数(D1, D0)取开平方根, 将结果存放至(D13, D12)所指定的寄存器当中。



### 6.7.16 DPOW (浮点数权值指令-幂次方)

类型	助记符	P	浮点数权值指令 (幂次方)																						
字		■																							
双字	DPOW		In1(D)							In2(D)							Out1(D)								
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Out1										■	■	■	■	■	■		■	■	■	■			■		

**参数说明:**

- 1, In1: 底数装置;
- 2, In2: 次幂数装置;
- 3, Out1: 运算结果装置。

**指令执行注意点:**

- 1, In1 的内容值只有非负数有效, 运算时均以浮点数方式执行, 常数均被自动转换成二进制浮点数;
- 2, 若运算结果为 0, 则**零标志位 SM50.2** 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

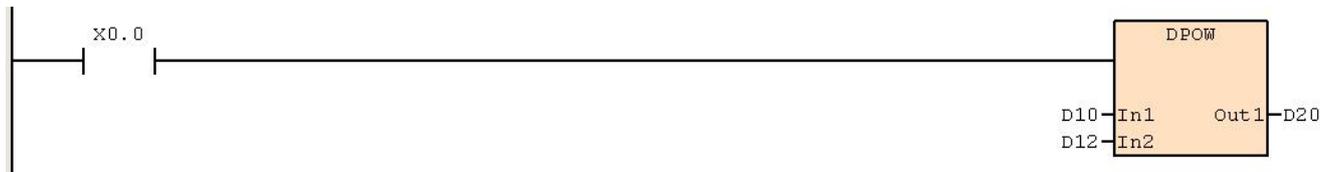
**指令功能说明:**

- 1, 将二进制浮点数据 In1 及 In2 以次幂数相乘后存放于 Out1 中:  $Out1 = In1 ^ In2$ 。

**程序范例:**

**例:**

当 X0.0 为 On 时, 将(D11, D10)及(D13, D12)32 位寄存器二进制浮点数做 pow 运算并将结果存于(D21, D20) 32 位寄存器中。





### 6.7.17 ESCLP (浮点数参数型比例值运算)

类型	助记符	P	浮点数参数型比例值运算																						
字		■																							
双字	ESCLP		In1(D)									In2(D)									Out1(D)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■	■		■	■	■	■			■		

#### 参数说明:

- 1, In1: 来源数值资料;
- 2, In2: 参数;
- 3, Out1: 计算目的装置。

#### 指令执行注意点:

- 1, In2 操作数将连续占用 4 个双字装置, 参数设置内容如下:

装置编号	参数名称说明
In2	来源数值最大值
In2+1	来源数值最小值
In2+2	目的数值最大值
In2+3	目的数值最小值

- 2, 来源数值和目的数值运算关系:

$y = kx + b$  式中:

$y =$  目的数值(Out1)

$k =$  斜率 = (目的数值最大值 - 最小值) ÷ (来源数值最大值 - 最小值)

$x =$  来源数值 (In1)

$b =$  偏移量 = 目的数值最小值 - 来源数值最小值 × 斜率

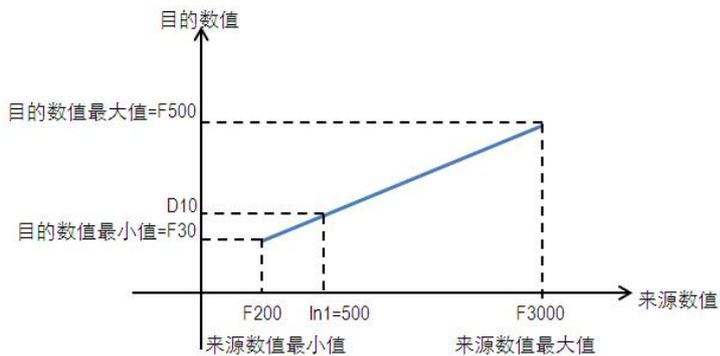
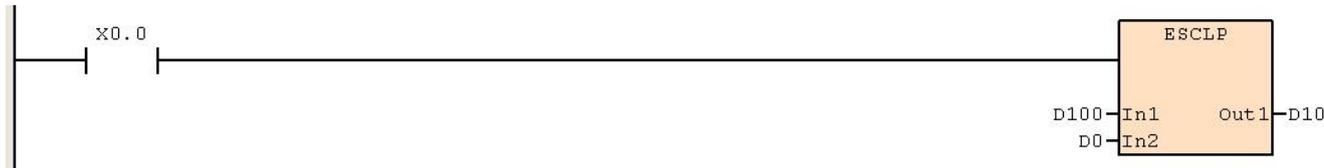
#### 指令功能说明:

- 1, 指令内部运算公式为  $Out1 = [(In1 - 来源数值最小值) \times (目的数值最大值 - 目的数值最小值)] \div (来源数值最大值 - 来源数值最小值) + 目的数值最小值$ 。

程序范例：

例：

- 已知 In1 数值资料来源 D100 为 F500，来源数值最大值 D0 为 F3000，来源数值最小值 D2 为 F200，目的数值最大值 D4 为 F500，目的数值最小值 D6 为 F30，当 X0.0=On 时，可在 D10 得到所要求的比例值。
- 运算方式： $D10 = [(F500 - F200) \times (F500 - F30)] \div (F3000 - F200) + F30 = F80.35$ 。





## 6.8 三角函数运算

### 6.8.1 DSIN (二进制浮点数 SIN 运算)

类型	助记符	P	二进制浮点数 SIN 运算																				
字		■																					
双字	DSIN		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■		■	■	■	■				■

#### 参数说明:

1, In1: 指定的来源值; Out1: 取 SIN 值结果;

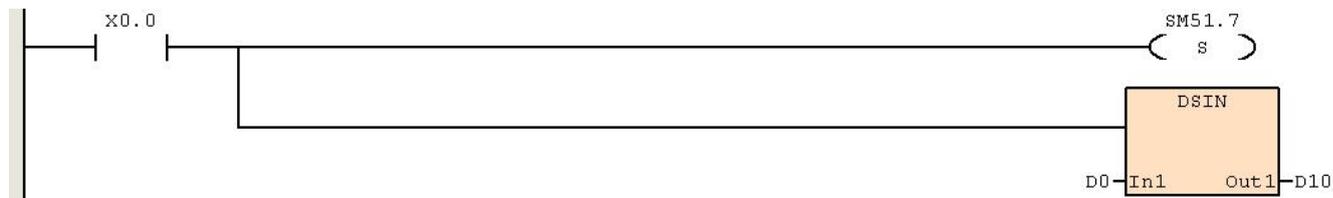
#### 指令执行注意点:

- 1, In1 所指定的来源可指定为弧度或角度, 由**三角函数操作数模式 SM51.7** 决定;
- 2, 当 SM51.7=OFF 时, 指定为角度模式, 角度范围:  $0^\circ \leq \text{角度值} < 360^\circ$ ;
- 3, 当 SM51.7=ON 时, 指定为弧度模式, 弧度(RAD)值等于 (角度 \* Pi/180) ;
- 4, 当计算结果为 0, 则**零标志位 SM50.2** 置起;
- 5, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位 SM52.0** 置起; 如果大于可表示的最大浮点数值, **进位标志位 SM50.4** 置起; 如果小于可表示的最小浮点数值, **借位标志位 SM50.3** 置起。

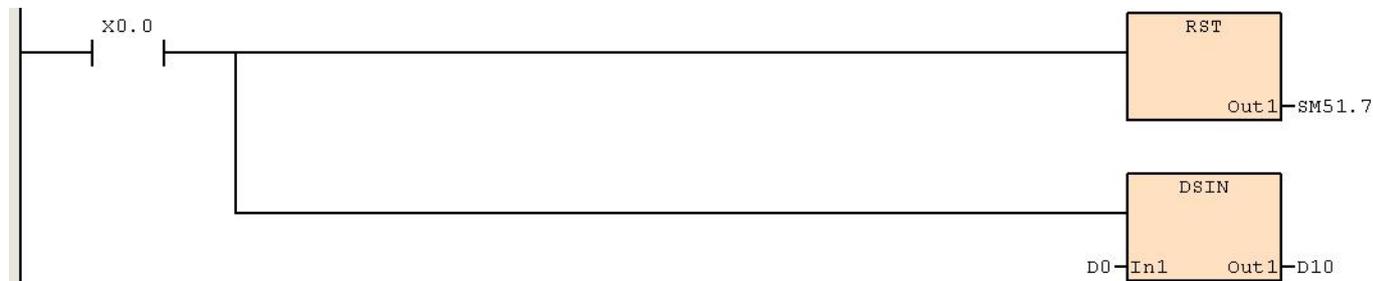
#### 程序范例:

##### 例:

SM51.7 = ON, 指定为弧度模式, 当 X0.0=On 时, 指定二进制浮点数(D1, D0)的弧度(RAD)值求取 SIN 值后存于(D11, D10) 当中, 内容为二进制浮点数。



SM51.7 = OFF, 指定为角度模式, 当 X0.0=On 时, 指定(D1, D0) 的角度值, 角度范围:  $0^\circ \leq \text{角度值} < 360^\circ$ 。求取 SIN 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.2 DCOS (二进制浮点数 COS 运算)

类型	助记符	P	二进制浮点数 COS 运算																				
字		■																					
双字	DCOS		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

1, In1: 指定的来源值; Out1: 取 COS 值结果。

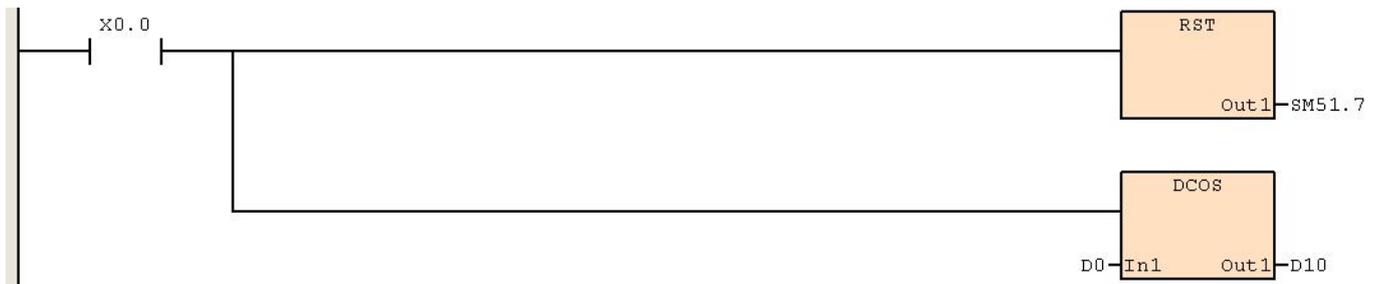
**指令执行注意点:**

- 1, In1 所指定的来源可指定为弧度或角度, 由三角函数操作数模式 SM51.7 决定;
- 2, 当 SM51.7=OFF 时, 指定为角度模式, 角度范围:  $0^\circ \leq \text{角度值} < 360^\circ$ ;
- 3, 当 SM51.7=ON 时, 指定为弧度模式, 弧度(RAD)值等于 (角度 \* Pi/180) ;
- 4, 当计算结果为 0, 则零标志位 SM50.2 置起;
- 5, 如果浮点数无法正常表达, 则浮点数无法正常表达数据标志位 SM52.0 置起; 如果大于可表示的最大浮点数值, 进位标志位 SM50.4 置起; 如果小于可表示的最小浮点数值, 借位标志位 SM50.3 置起。

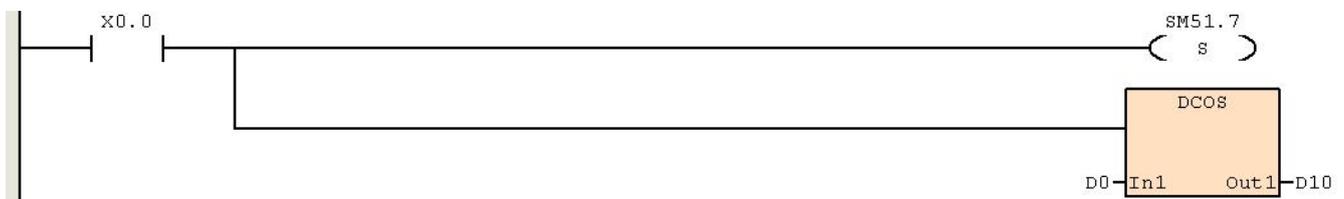
**程序范例:**

例:

SM51.7 = OFF, 指定为角度模式, 当 X0.0=On 时, 指定(D1, D0) 的角度值, 角度范围:  $0^\circ \leq \text{角度值} < 360^\circ$ 。求取 COS 值后存于(D11, D10) 当中, 内容为二进制浮点数。



SM51.7=ON, 指定为弧度模式, 当 X0.0=On 时, 指定二进制浮点数(D1, D0)的弧度(RAD)值求取 COS 值后存于(D11, D10) 当中, 内容为二进制浮点数。





### 6.8.3 DTAN (二进制浮点数 TAN 运算)

类型	助记符	P	二进制浮点数 TAN 运算																							
字		■																								
双字	DTAN		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■			■	■	■				■			

#### 参数说明:

1, In1: 指定的来源值; Out1: 取 TAN 值结果。

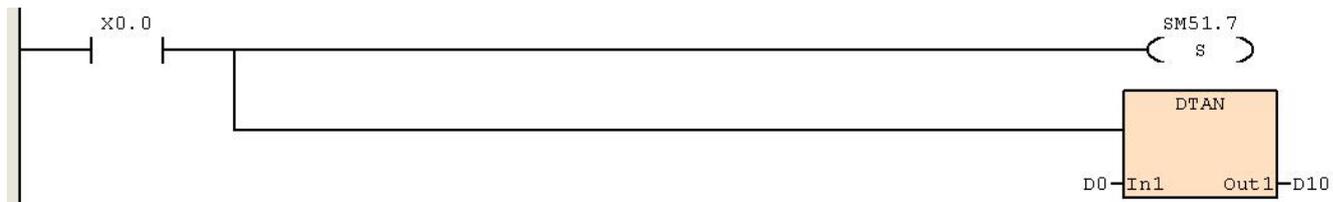
#### 指令执行注意点:

- 1, In1 所指定的来源可指定为弧度或角度, 由三角函数操作数模式 SM51.7 决定;
- 2, 当 SM51.7=OFF 时, 指定为角度模式, 角度范围:  $0^\circ \leq \text{角度值} < 360^\circ$ ;
- 3, 当 SM51.7=ON 时, 指定为弧度模式, 弧度(RAD)值等于 (角度 \* Pi/180) ;
- 4, 当计算结果为 0, 则零标志位 SM50.2 置起;
- 5, 如果浮点数无法正常表达, 则浮点数无法正常表达数据标志位 SM52.0 置起; 如果大于可表示的最大浮点数值, 进位标志位 SM50.4 置起; 如果小于可表示的最小浮点数值, 借位标志位 SM50.3 置起。

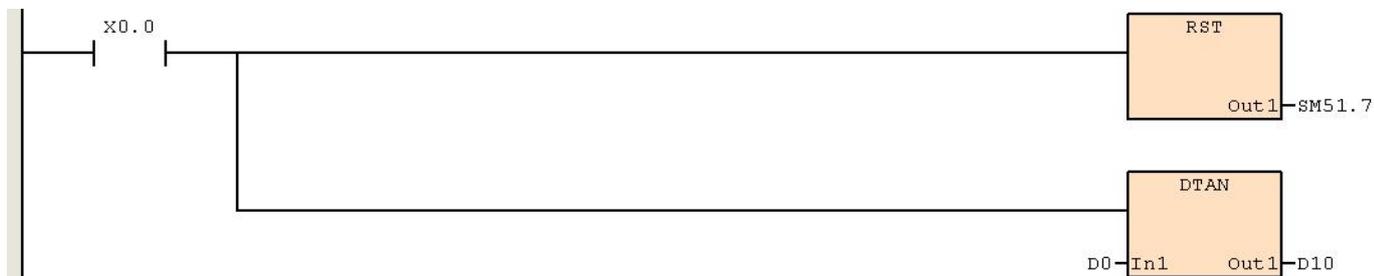
#### 程序范例:

##### 例:

SM51.7=ON, 指定为弧度模式, 当 X0.0=On 时, 指定二进制浮点数(D11, D10)的弧度(RAD)值求取 TAN 值后存于(D11, D10) 当中, 内容为二进制浮点数。



SM51.7=OFF, 指定为角度模式, 当 X0.0=On 时, 指定(D11, D10) 的角度值, 角度范围:  $0^\circ \leq \text{角度值} < 360^\circ$ 。求取 TAN 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.4 DASIN (二进制浮点数 ASIN 运算)

类型	助记符	P	二进制浮点数 ASIN 运算																				
字		■																					
双字	DASIN		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

1, In1: 指定的来源; Out1: 取 ASIN 值 (即  $\sin^{-1}$ ) 结果。

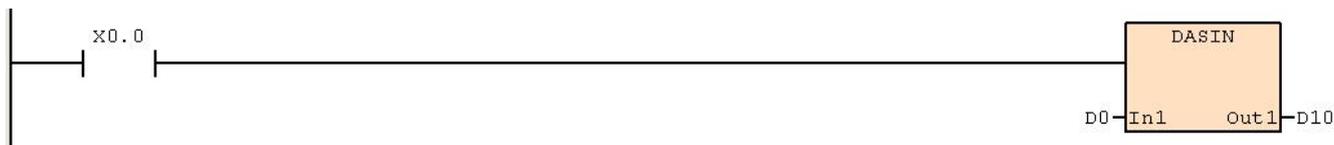
**指令执行注意点:**

- 1, In1 操作数指定的正弦值数值只能介于-1.0~1.0 之间, 若不在此范围内, 则指令不执行;
- 2, 若转换结果为 0, 则**零标志位** SM50.2 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

**程序范例:**

例:

当 X0.0=On 时, 指定二进制浮点数(D11, D0)求取 ASIN 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.5 DACOS (二进制浮点数 ACOS 运算)

类型	助记符	P	二进制浮点数 ACOS 运算																						
字		■																							
双字	DACOS		In1(D)											Out1(D)											
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■		■	■	■					■		

**参数说明:**

1, In1: 指定的来源; Out1: 取 ACOS 值 (即  $\cos^{-1}$ ) 结果。

**指令执行注意点:**

- 1, In1 操作数指定的余弦值数值只能介于-1.0~1.0 之间, 若不在此范围内, 则指令不执行;
- 2, 若转换结果为 0, 则**零标志位** SM50.2 置起;
- 3, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

**程序范例:**

**例:**

当 X0.0=On 时, 指定二进制浮点数(D11, D10)求取 ACOS 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.6 DATAN (二进制浮点数 ATAN 运算)

类型	助记符	P	二进制浮点数 ATAN 运算																				
字		■																					
双字	DATAN		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

1, In1: 指定的来源; Out1: 取 ATAN 值 (即  $\tan^{-1}$ ) 结果。

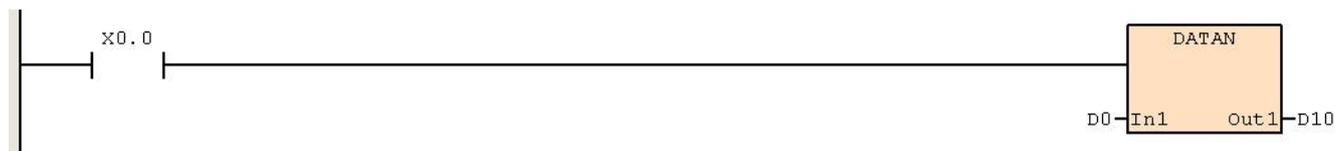
**指令执行注意点:**

- 1, 若转换结果为 0, 则**零标志位** SM50.2 置起;
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

**程序范例:**

例:

当 X0.0=On 时, 指定二进制浮点数(D1, D0)求取 TAN 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.7 DSINH (二进制浮点数 SINH 运算)

类型	助记符	P	二进制浮点数 SINH 运算																							
字		■																								
双字	DSINH		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■		■	■	■					■			

**参数说明:**

- 1, In1: 指定的来源;
- 2, Out1: 取 SINH 值结果。

**指令执行注意点:**

- 1, 若转换结果为 0, 则**零标志位** SM50.2 置起;
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

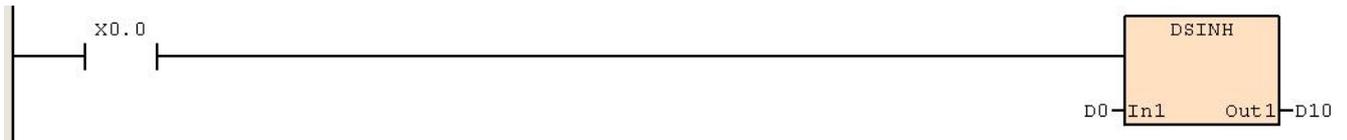
**指令功能说明:**

- 1, 双曲三角函数  $\sinh = (e^{In1} - e^{-In1}) / 2$ 。

**程序范例:**

**例:**

当 X0.0=On 时, 指定二进制浮点数(D1, D0)求取 SINH 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.8 DCOSH (二进制浮点数 COSH 运算)

类型	助记符	P	二进制浮点数 COSH 运算																				
字		■																					
双字	DCOSH		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Out1										■	■	■	■	■	■		■	■	■	■			■

**参数说明:**

- 1, In1: 指定的来源;
- 2, Out1: 取 COSH 值结果。

**指令执行注意点:**

- 1, 若转换结果为 0, 则**零标志位** SM50.2 置起;
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

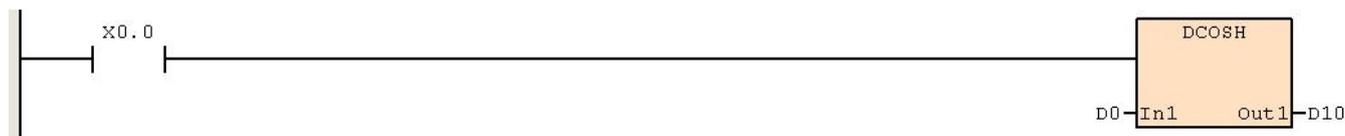
**指令功能说明:**

1, 双曲三角函数  $\cosh = (e^{In1} + e^{-In1}) / 2$ 。

**程序范例:**

例:

当 X0.0=On 时, 指定二进制浮点数(D1, D0)求取 COSH 值后存于(D11, D10) 当中, 内容为二进制浮点数。



### 6.8.9 DTANH (二进制浮点数 TANH 运算)

类型	助记符	P	二进制浮点数 TANH 运算																							
字		■																								
双字	DTANH		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	■		■	■	■					■			

**参数说明:**

- 1, In1: 指定的来源;
- 2, Out1: 取 TANH 值结果。

**指令执行注意点:**

- 1, 若转换结果为 0, 则**零标志位** SM50.2 置起;
- 2, 如果浮点数无法正常表达, 则浮点数**无法正常表达数据标志位** SM52.0 置起; 如果大于可表示的最大浮点数值, **进位标志位** SM50.4 置起; 如果小于可表示的最小浮点数值, **借位标志位** SM50.3 置起。

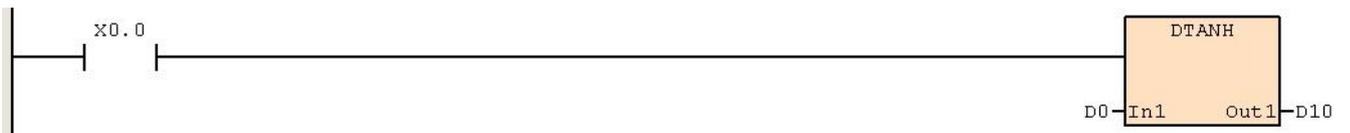
**指令功能说明:**

- 1, 双曲三角函数  $\tanh = (e^{In1} - e^{-In1}) / (e^{In1} + e^{-In1})$ 。

**程序范例:**

**例:**

当 X0.0=On 时, 指定二进制浮点数(D1, D0)求取 TANH 值后存于(D11, D10) 当中, 内容为二进制浮点数。



## 6.9 万年历

### 6.9.1 TCMP (万年历数据比较)

类型	助记符	P	万年历数据比较																				
字	TCMP	■	In1(W)										Out1(b)										
双字																							
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■		
Out1		■	■	■																			

**参数说明:**

- 1, In1: 占用连续 3 个字装置, 分别为 In1 设置比较时间的“时”, 设置范围为 0~23;
- 2, In1+1: 设置比较时间的“分”, 设置范围为 0~59;
- 3, In1+2: 设置比较时间的“秒”, 设置范围为 0~59;
- 4, Out1: 比较结果。

**指令执行注意点:**

- 1, 所设置值的范围超过范围, 则指令不执行;
- 2, 万年历现在时间存在 SD120~SD126 中, SD120~SD126 实时刷新, 万年历 SD 具体分配见 SD 列表;
- 3, 若比较时间小于现在时间, 则 Out1+2 置起, 若等于, 则 Out1+1 置起, 若大于, 则 Out1 置起。

**指令功能说明:**

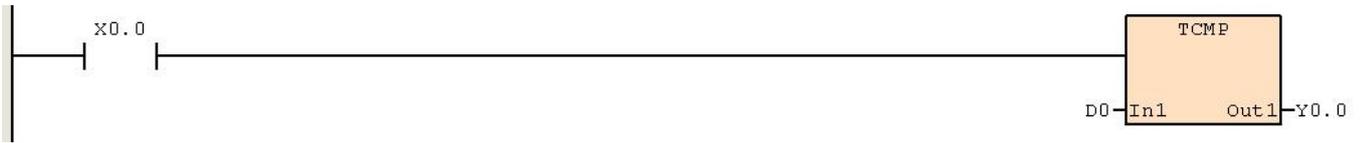
- 1, 将 In1 所指定的时、分、秒设置值与 SD120~SD126 中所指的万年历时、分、秒现在值做比较, 其比较结果在 Out1 作表示。



**程序范例:**

**例:**

- 当 X0.0=On 时, 指令执行, 将 SD120~SD126 万年历现在时间与设置值 D0~D2 做比较, 将结果显示到 Y0.0~Y0.2。当 X0.0 由 On→Off 变化时, 指令不被执行, 但是 Y0.0~Y0.2 之前的 On/Off 状态仍被保持住。
- 若需要得到  $\geq$ 、 $\leq$ 、 $\neq$  的结果时, 可将 Y0.0~Y0.2 串并联即可取得。



D0 (时)	>	SD122 (时)	的时候, Y0.0 为 ON
D1 (分)		SD121 (分)	
D2 (秒)		SD120 (秒)	

D0 (时)	=	SD122 (时)	的时候, Y0.1 为 ON
D1 (分)		SD121 (分)	
D2 (秒)		SD120 (秒)	

D0 (时)	<	SD122 (时)	的时候, Y0.2 为 ON
D1 (分)		SD121 (分)	
D2 (秒)		SD120 (秒)	

### 6.9.2 TZCP (万年历数据区间比较)

类型	助记符	P	万年历数据区间比较																						
字	TZCP	■	In1(W)							In2(W)							Out1(b)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□		■	■	■	■	■	■				
In2									■	■	■	■	■	□		■	■	■	■	■	■				
Out1		■	■	■																					

**参数说明:**

- 1, In1: 设置比较时间的下限值;
- 2, In2: 设置比较时间的上限值;
- 3, Out1: 比较结果。

**指令执行注意点:**

- 1, In1、In1+1、In1+2: 设置比较时间下限值的“时”、“分”、“秒”;
- 2, In2、In2+1、In2+2: 设置比较时间上限值的“时”、“分”、“秒”;
- 3, 若设定值超过允许的合法范围, 则指令不执行;
- 4, 万年历现在时间存在 SD120~SD126 中, SD120~SD126 实时刷新, 万年历 SD 具体分配见 SD 列表;
- 5, 当现在时间小于下限值时, 则 Out1 为 On, 当现在时间大于上限值时, 则 Out1+2 为 On, 其余状态则 Out1+1 为 On;
- 6, 当设定的下限值 In1 大于上限值 In2 时, 上下限值都取 In2 作为比较值。

**指令功能说明:**

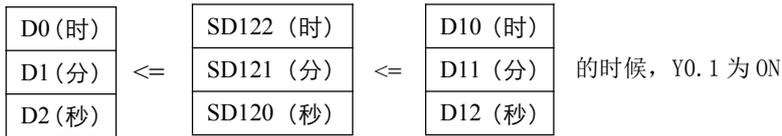
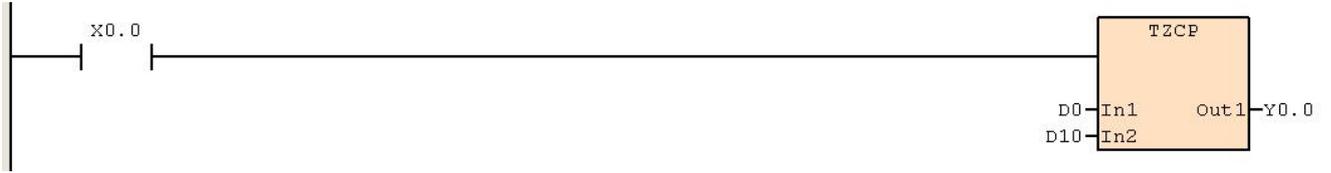
- 1, 将由 SD120~SD126 区域所指定的万年历现在时间时、分、秒值与 In1 所指定设置比较时间的下限值及 In2 所指定设置比较时间的上限值做区间比较, 其比较结果在 Out1 作表示。



**程序范例:**

**例:**

当 X0.0=On 时, TZCP 指令执行, Y0.0~Y0.2 其中的一个会 On, 当 X0.0=Off 时, TZCP 指令不执行, Y0.0~Y0.2 状态保持在 X0.0=Off 之前的状态。



### 6.9.3 TADD (万年历数据加算)

类型	助记符	P	万年历数据加算																				
字	TADD	■	In1(W)										Out1(W)										
双字																							
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■		
Out1										■	■	■	■	□			■	■	■	■			

**参数说明:**

- 1, In1: 时间加数;
- 2, Out1: 时间和。

**指令执行注意点:**

- 1, 万年历现在时间存在 SD120~SD126 中, SD120~SD126 实时刷新, 万年历 SD 具体分配见 SD 列表;
- 2, 加算结果若大于等于 24 小时的话, **进位标志 SM50.4** 置起, Out1 显示加算总值减掉 24 小时所得的结果;
- 3, 加算结果若是等于 0 (0 时 0 分 0 秒), **零标志位 SM50.2** 置起;
- 4, 设定值必须在允许的合法范围内, 超出则不执行。

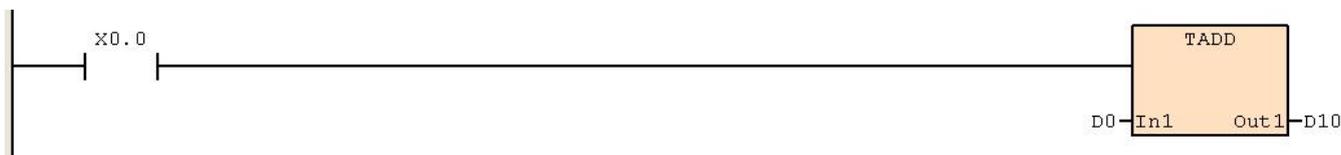
**指令功能说明:**

- 1, 将 SD120~SD126 中所指定的万年历数据时、分、秒加上 In1 所指定的万年历数据时、分、秒, 所得到的结果存于指定 Out1 所指定的寄存器时、分、秒当中。

**程序范例:**

**例:**

当 X0.0=On 时, TADD 指令执行, 将 D0~D2 所指定的万年历数据时、分、秒与 SD120~SD126 所指定的万年历数据时、分、秒相加, 所得到的结果存于 D10~D12 所指定的寄存器中得到加总后的时、分、秒。



D0 (时)	+	SD122 (时)	=	D10 (时)
D1 (分)		SD121 (分)		D11 (分)
D2 (秒)		SD120 (秒)		D12 (秒)

50 分 26 秒

8 时 10 分 20 秒

加算结果若超过 24 小时的话, 进位标志 SM50.4=On。

D0 (时)	+	SD122 (时)	=	D10 (时)
D1 (分)		SD121 (分)		D11 (分)
D2 (秒)		SD120 (秒)		D12 (秒)

10 分 38 秒

11 时 30 分 8 秒

### 6.9.4 TSUB (万年历数据减算)

类型	助记符	P	万年历数据减算																							
字	TSUB	■	In1(W)												Out1(W)											
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
Out1										■	■	■	■	□			■	■	■	■						

**参数说明:**

1, In1: 时间减数; Out1: 时间差。

**指令执行注意点:**

- 1, 万年历现在时间存在 SD120~SD126 中, SD120~SD126 实时刷新, 万年历 SD 具体分配见 SD 列表;
- 2, 减算结果若为负数的话, **借位标志 SM50.3** 置起, Out1 显示减算总值加上 24 小时所得的结果;
- 3, 减算结果若是等于 0 (0 时 0 分 0 秒), **零标志位 SM50.2** 置起;
- 4, 设定值必须在允许的合法范围内, 超出则不执行。

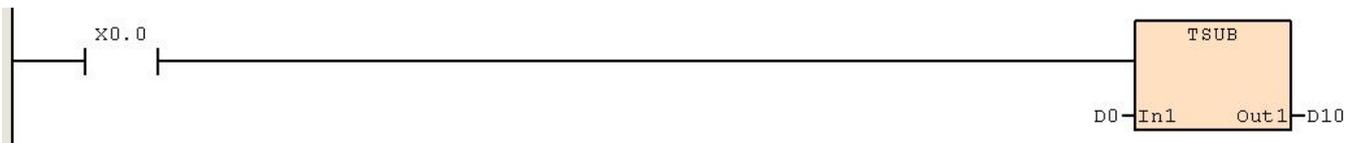
**指令功能说明:**

1, 将 SD120~SD126 中所指定的万年历数据时、分、秒减去 In1 所指定的万年历数据时、分、秒, 所得到的结果存于指定 Out1 所指定的寄存器时、分、秒当中。

**程序范例:**

**例:**

当 X0.0=On 时, TSUB 指令执行, 将 SD120~SD126 所指定的万年历数据时、分、秒与 D0~D2 所指定的万年历数据时、分、秒相减, 所得到的结果存于 D10~D12 所指定的寄存器中得到相减后的时、分、秒。



SD122 (时)		D0 (时)	=	D10 (时)
SD121 (分)	—	D1 (分)	=	D11 (分)
SD120 (秒)		D2 (秒)		D12 (秒)
20 时 20 分 7 秒		14 时 30 分 5 秒		14 时 30 分 5 秒      5 时 50 分 2 秒

减算结果若为负数的话, 借位标志 SM50.3=On。

SD122 (时)		D0 (时)	=	D10 (时)
SD121 (分)	—	D1 (分)	=	D11 (分)
SD120 (秒)		D2 (秒)		D12 (秒)
19 时 11 分 15 秒		5 时 20 分 30 秒		19 时 11 分 15 秒      10 时 9 分 15 秒

### 6.9.5 TWR (万年历时间写入更新)

类型	助记符	P	万年历时间写入更新
字	TWR	■	无
双字			

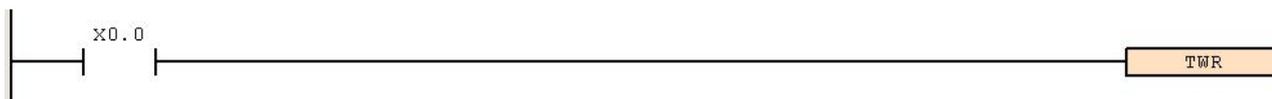
**指令说明:**

- 1, PLC 主机内建万年历时钟, 要调整万年历时钟的时候, 可使用本指令将要调整的现在时间写入至内建的万年历时钟当中;
- 2, 本指令执行后, 新的设置时间, 即 SD130~SD136 中的当前时间设置立刻被写入至 PLC 内部的万年历时钟当中, 因此执行本指令时, 请注意所写入的新设置时间, 与写入当时现在时间是否吻合, 万年历 SD 具体分配见 SD 列表。
- 3, 当正在进行万年历写入操作时, 万年历**正在变更标志位** SM53.7 置 ON; 写入操作完成时, SM53.7 置 OFF。

**程序范例:**

**例:**

当 X0.0=On 时, 将 SD130~SD136 中的当前时间写入至 PLC 内部的万年历时钟当中。



SD	功能定义	读写权限
130	万年历秒针写入寄存器	RW
131	万年历分针写入寄存器	RW
132	万年历时针写入寄存器	RW
133	万年历星期写入寄存器	RW
134	万年历日期写入寄存器	RW
135	万年历月份写入寄存器	RW
136	万年历年份写入寄存器	RW

其中万年历星期写入寄存器 SD133 的值 0~6 分别对应星期日至星期六; 万年历年份写入寄存器 SD136 范围为 2000~2019。



## 6.9.6 HOUR (计时仪)

类型	助记符	P	计时仪																					
字	HOUR		In1(W)						In2(W)						InOut1(W)						Out1(b)			
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
InOut1										■	■	■	■	□			■	■	■	■				
Out1		■	■	■				■																

### 参数说明:

- 1, In1: 执行指令的编号;
- 2, In2: 设置导通时间, 单位: 小时;
- 3, InOut1: 测量中的现在时间值, 单位: 小时;
- 4, Out1: 输出装置。

### 指令执行注意点:

- 1, In1 的取值范围为 0~3, 共可使用 4 次分别计时, 编号不可相同;
- 2, In2 为设置导通时间, 设置范围为 1~32767, 单位: 小时;
- 3, InOut1 为测量中的现在时间值, InOut1 范围 0~32767, 单位: 小时; InOut1+1 范围为 0~3599, 单位: 秒;
- 4, 当输出装置导通后, 定时器会继续计时;
- 5, 当现在值时间到达最大数值, 会停止计时测量, 要重新计时须清除为 0;
- 6, 请将 InOut1 设置为停电保持区域, 这样可以保证在停电后, 依然能够保持计时值, 下次上电后可继续计时而不被清 0。

### 指令功能说明:

- 1, 将输入接点导通时间做计时, 当到达设置时间 (以小时为单位), 会将输出装置导通。可提供使用者管理机械的运作计时或维修。

### 程序范例:

#### 例:

使用执行指令编号为 0, 当 X0.0=On 时开始计时, 当到达 10 小时 Y0.0 导通, 而 D0 会记录测量中的现在时间值(单位: 小时), D1 会记录测量中不足 1 小时的现在时间值 0~3599(单位: 秒)。



## 6.10 格雷码

### 6.10.1 GRY (BIN->GRAY 码转换)

类型	助记符	P	BIN->GRAY 码转换																				
字	GRY	■	In1(W)										Out1(W)										
双字	DGRY		In1(D)										Out1(D)										
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■

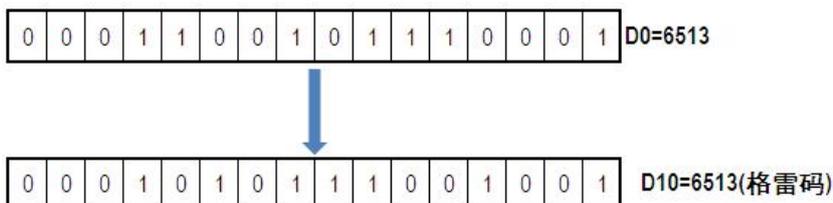
**指令说明:**

- 1, In1: 来源装置; Out1: 存放 GRAY 码的装置;
- 2, 将 In1 所指定装置的内容值 (BIN 值) 变换格雷码 (GRAY CODE) 后存放 to Out1 所指定的装置中。

**程序范例:**

**例:**

当 X0.0=On 时, 将 D0 所指定装置的内容值 (BIN 值) 变换格雷码 (GRAY CODE) 后存放 to D10 所指定的装置中。



### 6.10.2 GBIN (GRAY 码->BIN 转换)

类型	助记符	P	GRAY 码->BIN 转换																							
字	GBIN	■	In1(W)												Out1(W)											
双字	DGBIN		In1(D)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□	□		■	■	■				■			

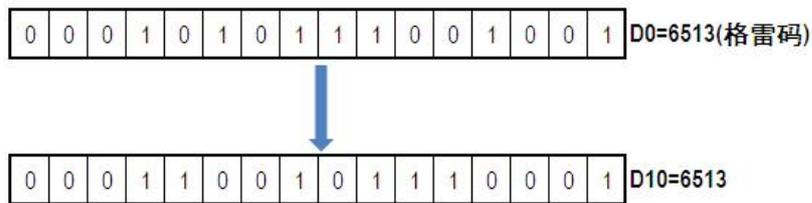
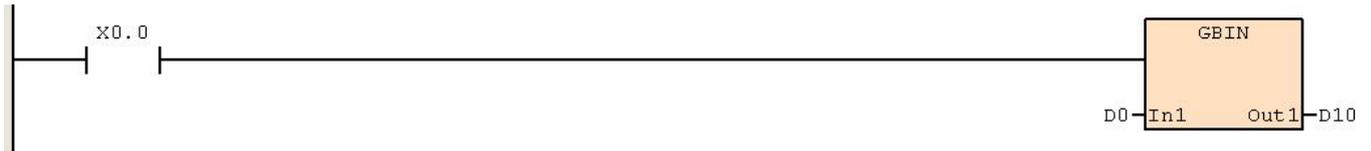
**指令说明:**

- 1, In1: 来源装置; Out1: 存放 GBIN 码的装置;
- 2, 将 In1 所指定装置的内容值 (GRAY CODE 值) 变换成自然二进制数值 (BIN) 后存放到 Out1 所指定的装置中。

**程序范例:**

**例:**

当 X0.0=On 时, 将 D0 所指定装置的内容值 (GRAY CODE 值) 变换成自然二进制数值 (BIN) 后存放到 D10 所指定的装置。



## 6.11 矩阵处理

### 6.11.1 MAND (矩阵与运算)

类型	助记符	P	矩阵与 (AND) 运算																							
字	MAND	■	In1(W)								In2(W)								In3(W)				Out1(W)			
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■					
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□			■	■	■	■						

**参数说明:**

- 1, In1: 矩阵来源装置 1;
- 2, In2: 矩阵来源装置 2;
- 3, In3: 数组长度, 取值范围为 1~256;
- 4, Out1: 运算结果。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In3, 一个矩阵共有 16\*In3 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In3 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

**指令功能说明:**

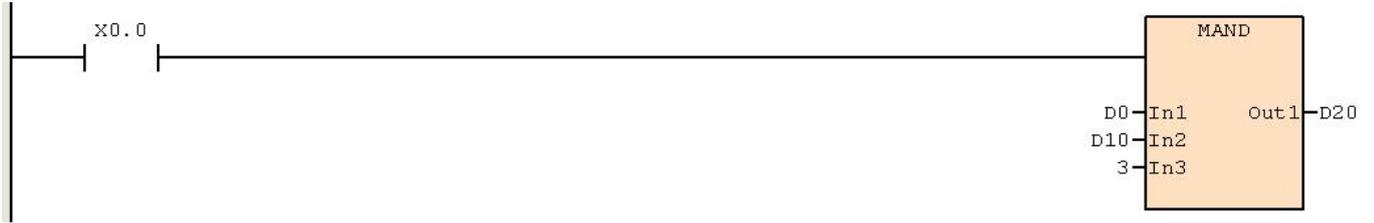
- 1, 两个矩阵来源依数组长度 In3 将 In1 及 In2 作矩阵的“与” (AND) 运算并将结果存于 Out1 中; 矩阵的“与” (AND) 运算的规则为两位均为 1 结果为 1, 否则为 0。



**程序范例:**

**例:**

当 X0.0=On, 16 位寄存器 D0~D2 共三行与 16 位 D10~D12 共三行作 MAND, 矩阵与(AND)运算, 将结果存于 D20~D22 中。



D0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D2	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1

MAND

D10	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0
D11	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0
D12	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0



D20	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
D21	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
D22	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

### 6.11.2 MOR (矩阵或运算)

类型	助记符	P	矩阵或 (OR) 运算																															
字	MOR	■	In1(W)								In2(W)								In3(W)								Out1(W)							
双字																																		
变量	位装置									字/双字装置																								
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L											
In1									■	■	■	■	■	□		■	■	■	■	■	■													
In2									■	■	■	■	■	□		■	■	■	■	■	■													
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■											
Out1										■	■	■	■	□			■	■	■	■														

**参数说明:**

1, In1: 矩阵来源装置 1; In2: 矩阵来源装置 2; In3: 数组长度, 取值范围为 1~256; Out1: 运算结果。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In3, 一个矩阵共有 16\*In3 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In3 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

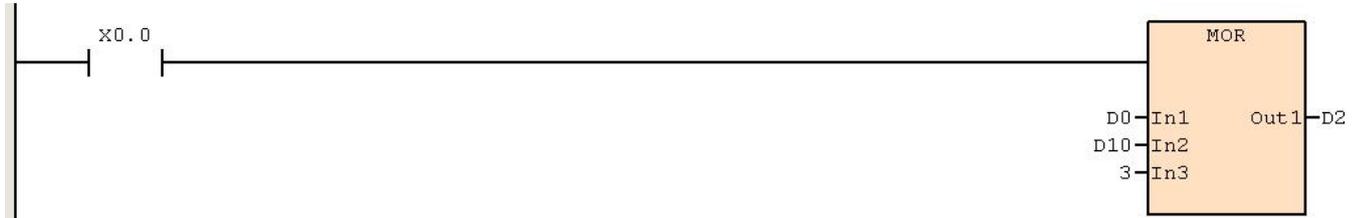
**指令功能说明:**

- 1, 两个矩阵来源依数组长度 In3 将 In1 及 In2 作矩阵的“或” (OR) 运算并将结果存于 Out1 中; 矩阵的“或” (OR) 运算的规则为两位中只要有一个为 1 结果为 1, 否则为 0。

**程序范例:**

**例:**

当 X0.0=On 时, 16 位寄存器 D0~D2 共三行与 16 位 D10~D12 共三行作 MOR, 矩阵或(OR)运算, 将结果存于 D20~D22 三行。



D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

MOR

D10	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
D11	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
D12	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1



D20	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1
D21	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1
D22	0	1	0	1	1	1	1	1	1	1	1	0	1	0	1

### 6.11.3 MXOR (矩阵异或运算)

类型	助记符	P	矩阵异或 (XOR) 运算																				
字	MXOR	■	In1(W)							In2(W)							In3(W)				Out1(W)		
双字			位装置							字/双字装置													
变量	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■		
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■
Out1										■	■	■	■	□			■	■	■	■			

**参数说明:**

- 1, In1: 矩阵来源装置 1;
- 2, In2: 矩阵来源装置 2;
- 3, In3: 数组长度, 取值范围为 1~256;
- 4, Out1: 运算结果。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In3, 一个矩阵共有 16\*In3 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In3 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

**指令功能说明:**

- 1, 两个矩阵来源依数组长度 In3 将 In1 及 In2 作矩阵的“异或” (XOR) 运算并将结果存于 Out1 中; 矩阵的“异或” (XOR) 运算的规则为两位不同结果为 1, 否则为 0。

**程序范例:**

**例:**

当 X0.0=On 时, 16 位寄存器 D0~D2 共三行与 16 位寄存器 D10~D12 共三行作 MXOR, 矩阵异或 (XOR) 运算, 将结果存于 16 位寄存器 D20~D22 共三行中。



D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D2	0	1	0	1	0	1	0	1	0	1	0	1	0	1

**MXOR**

D10	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
D11	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
D12	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1



D20	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0
D21	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0
D22	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

### 6.11.4 MXNR (矩阵同或运算)

类型	助记符	P	矩阵同或 (XNR) 运算																				
字	MXNR	■	In1(W)							In2(W)							In3(W)				Out1(W)		
双字			位装置							字/双字装置													
变量	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■		
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■
Out1										■	■	■	■	□			■	■	■	■			

**参数说明:**

- 1, In1: 矩阵来源装置 1;
- 2, In2: 矩阵来源装置 2;
- 3, In3: 数组长度, 取值范围为 1~256;
- 4, Out1: 运算结果。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In3, 一个矩阵共有 16\*In3 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In3 个矩阵位当做一连串单点的集合, 自此集合中指定某一单点运作, 而不将装置当数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

**指令功能说明:**

- 1, 两个矩阵来源依数组长度 In3 将 In1 及 In2 作矩阵的“同或” (XNR) 运算并将结果存于 Out1 中; 矩阵的“同或” (XNR) 运算的规则为两位相同结果为 1, 否则为 0。



**程序范例:**

**例:**

当 X0.0=On 时, 16 位寄存器 D0~D2 共三行与 16 位寄存器 D10~D12 共三行作 MXNR, 矩阵 XNR 运算, 将结果存于 16 位寄存器 D20~D22 共三行中。



D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D2	0	1	0	1	0	1	0	1	0	1	0	1	0	1

**MXNR**

D10	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
D11	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1
D12	0	0	0	0	1	1	1	1	1	0	1	0	0	1	0	1



D20	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	1
D21	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	1
D22	1	0	1	0	0	1	0	1	0	0	0	0	1	1	1	1

### 6.11.5 MINV (矩阵求反运算)

类型	助记符	P	矩阵求反运算																			
			In1(W)						In2(W)						Out1(W)							
字	MINV	■																				
双字																						
变量	位装置									字/双字装置												
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O
In1								■	■	■	■	■	□		■	■	■	■	■	■		
In2								■	■	■	■	■	□		■	■	■	■	■	■	■	■
Out1									■	■	■	■	□			■	■	■	■			

**参数说明:**

1, In1: 矩阵来源装置; In2: 数组长度, 取值范围为 1~256; Out1: 运算结果;

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In3, 一个矩阵共有 16\*In3 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In3 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

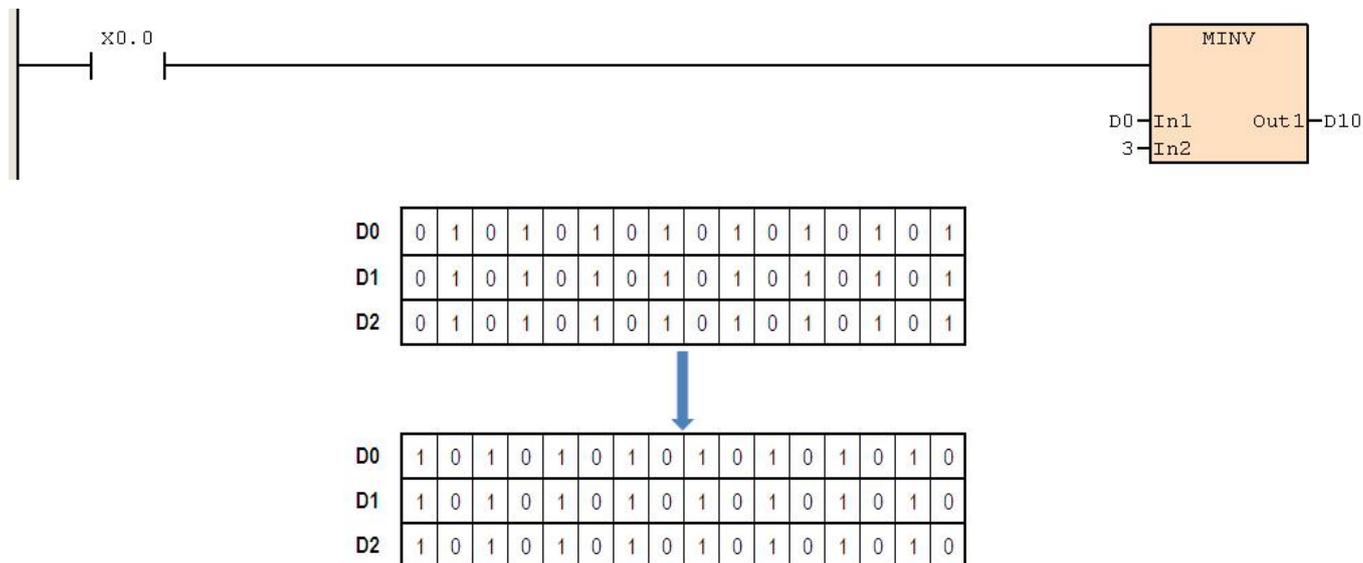
**指令功能说明:**

1, 将矩阵来源依数组长度 In2 将 In1 矩阵的“取反”(INV) 运算并将结果存于 Out1 中; 矩阵的“取反”(INV) 运算的规则为将 0 转 1, 1 转 0。

**程序范例:**

例:

当 X0.0=On 时, 16 位寄存器 D0~D2 共三行作 MINV, 矩阵反相运算, 将结果存于 16 位寄存器 D10~D12 共三行中。





### 6.11.6 MCMP (矩阵比较)

类型	助记符	P	矩阵比较																																												
字	MCMP	■	In1(W)									In2(W)									In3(W)									In4(W)									Out1(W)								
双字																																															
变量	位装置												字/双字装置																																		
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L																								
In1									■	■	■	■	■	□		■	■	■	■	■	■																										
In2									■	■	■	■	■	□		■	■	■	■	■	■																										
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■																								
In4									■	■	■	■	■	□		■	■	■	■	■	■	■	■																								
Out1										■	■	■	■	□			■	■	■	■	■		■																								

#### 参数说明:

- 1, In1: 矩阵来源装置 1;
- 2, In2: 矩阵来源装置 2;
- 3, In3: 数组长度, 取值范围为 1~256;
- 4, In4: 比较开始的指针位置, 不可大于  $16 * In3 - 1$ ;
- 5, Out1: 指针 Pr, 用于存放目标的位置值。

#### 指令执行注意点:

- 1, 由**矩阵比较模式标志位** SM53.0 决定比较相同值 (SM53.0 为 On) 还是比较不同值 (SM53.0 为 Off), 当比较到达时立即停止比较动作, **矩阵位寻找成功标志位** SM53.1=1; 当比较完最后一个 bit 时, **矩阵位寻找结束标志位** SM53.2 标志置 On; 当比较完最后一个 bit 还没找到不同 (或相同) 值时, Out1 将存放的值为  $16 * In3$ ;
- 2, 指针 Pr, 即目标的位置值, 为  $0 \sim 16 * In3 - 1$ , 即如果是在第二个数组的 bit3 找到目标, 那么  $Out1 = 19$ ;
- 3, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数为矩阵的长度, 一个矩阵共有  $16 * In3$  个位 (点), 其运算单位一次只有一个位 (点)。矩阵指令是将  $16 * In3$  个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点作运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 4, 矩阵的宽度固定为 16 位 (bits)。

#### 指令功能说明:

- 1, 每次比较两个矩阵来源依数组长度 In3, 将 In1 及 In2 两个矩阵中的每一个 bit 从位置值 In4 开始作比较, 找出值不同 (或相同) 的位置, 再将此位置值存放到 Out1 中, 完成此次比较。



### 6.11.7 MBRD (矩阵位读出)

类型	助记符	P	矩阵位读出																						
字	MBRD	■	In1(W)							In2(W)							In3(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■			
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■

**参数说明:**

1, In1: 矩阵来源装置; In2: 数组长度, 取值范围为 1~256; In3: 指针 Pr, 用于指定目标的位置值。

**指令执行注意点:**

- 1, 矩阵的指针 Pr, 由使用者于指令中指定, 其有效范围为 0~16\*In2-1, 若超出范围, 则指令不执行;
- 2, 指针 Pr, 即目标的位置值, 为 0~16\*In2-1, 即如果目标位置是在第二个数组的 bit3 处, 那么 In3 = 19;
- 3, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In2, 一个矩阵共有 16\*In2 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In2 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点作运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 4, 矩阵的宽度固定为 16 位 (bits)。

**指令功能说明:**

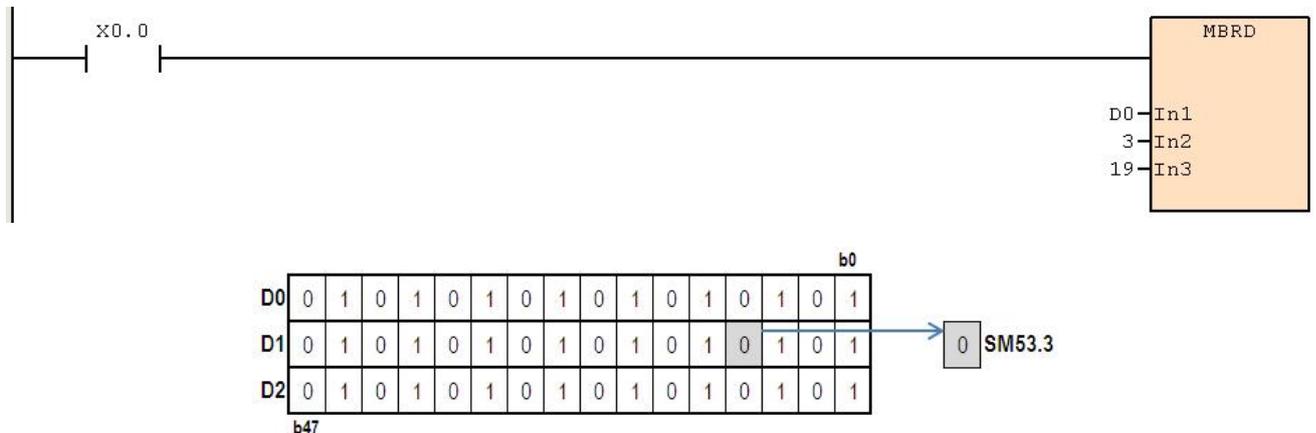
1, 当指令执行时, 将 In3 指定的位置矩阵位状态读出存放于**矩阵旋转位移输出进位标志位 SM53.3** 中。

**程序范例:**

**例:**

当 X0.0 由 Off→On 时, 将矩阵 D0 的第 19 位置矩阵位状态读出存放于矩阵旋转位移输出进位标志位 SM53.3 中, 此时 SM53.3=0。

第一行从右往左为 b0~b15, 第二行从右往左为 b16~b31, 第三行从右往左为 b32~b47。



### 6.11.8 MBWR (矩阵位写入)

类型	助记符	P	矩阵位写入																						
字	MBWR	■	In1(W)									In2(W)							InOut1(W)						
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L		X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
InOut1											■	■	■	■	□			■	■	■	■				

**参数说明:**

1, In1: 数组长度, 取值范围为 1~256; In2: 指针 Pr, 用于指定目标的位置值; InOut1: 矩阵来源和写入装置。

**指令执行注意点:**

- 1, 矩阵的指针 Pr, 由使用者于指令中指定, 其有效范围为 0~16\*In1-1, 若超出范围, 则指令不执行;
- 2, 指针 Pr, 即目标的位置值, 为 0~16\*In1-1, 即如果目标位置是在第二个数组的 bit3 处, 那么 In2 = 19;
- 3, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In1, 一个矩阵共有 16\*In1 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In1 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点作运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 4, 矩阵的宽度固定为 16 位 (bits)。

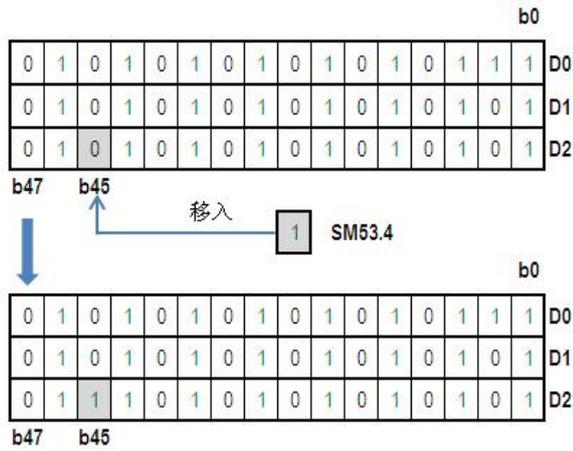
**指令功能说明:**

- 1, 当指令执行时, 将**矩阵旋转位移输入补位标志位 SM53.4** 放入 In2 指定的位置矩阵位。

**程序范例:**

**例:**

当 X0.0 由 Off→On 时, 将矩阵旋转位移输入补位标志位 SM53.4=1 放入矩阵 D0 的第 45 位中。  
 第一行从右往左为 b0~b15, 第二行从右往左为 b16~b31, 第三行从右往左为 b32~b47。



### 6.11.9 MBS (矩阵位移位)

类型	助记符	P	矩阵位移位																				
字	MBS	■	In1(W)							In2(W)							Out1(W)						
双字			位装置							字/双字装置													
变量	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■
Out1										■	■	■	■	□			■	■	■	■			

**参数说明:**

- 1, In1: 矩阵来源装置;
- 2, In2: 数组长度, 取值范围为 1~256;
- 3, Out1: 运算结果存放装置。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In2, 一个矩阵共有 16\*In2 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In2 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点作运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

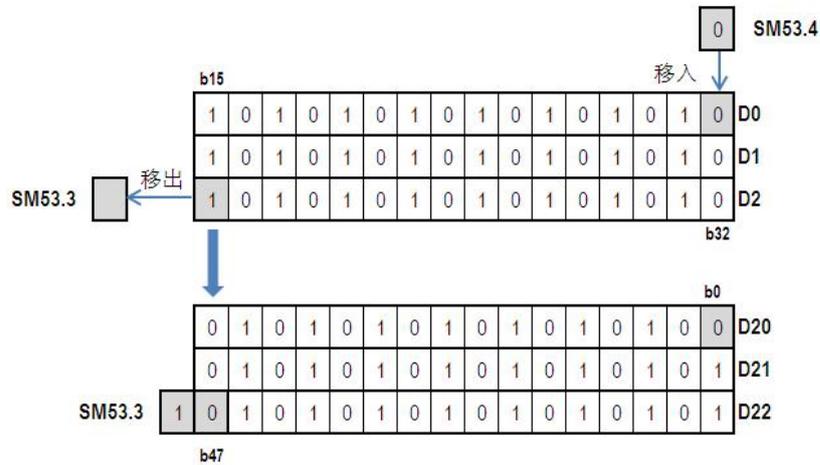
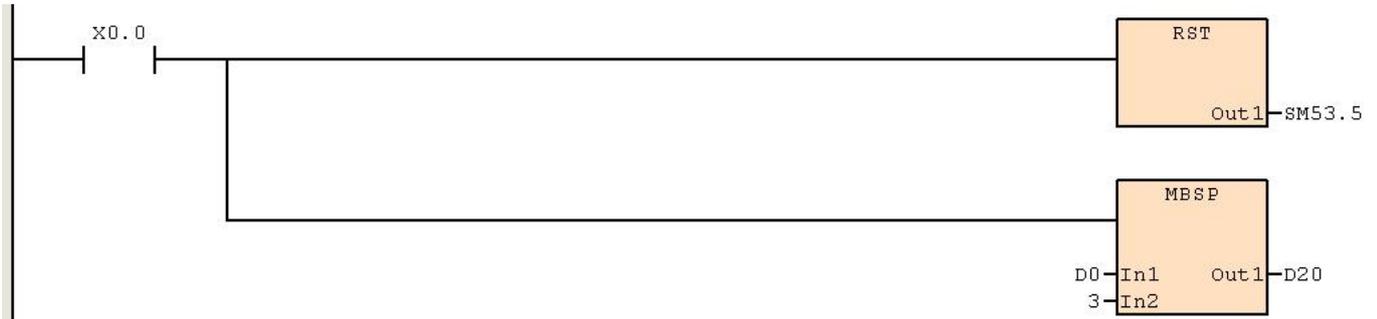
**指令功能说明:**

- 1, 矩阵来源依数组长度 In2 将 In1 矩阵位做左右移位控制, **矩阵左右移位模式** SM53.5= 0 决定矩阵位左移, SM53.5 = 1 矩阵位右移。每次移动一位, 因位移而腾出的空位 (左移时为 b0, 右移时为 b16\*In2-1) 则以**补位标志位** SM53.4 的状态填补。而因位移而挤出的位 (左移时为 b16\*In2 - 1, 右移时为 b0) 状态则送到**进位标志位** SM53.3 中去, 然后将结果存入 Out1。

程序范例:

例 1:

- 当 X0.0=On 时 SM53.5 = 0 作矩阵左移。设补位标志位 SM53.4=0, 16 位寄存器 D0~D2 矩阵作左移, 将结果存于 16 位寄存器 D20~D22 矩阵中, 进位标志 SM53.3=1。
- 第一行从右往左为 b0~b15, 第二行从右往左为 b16~b31, 第三行从右往左为 b32~b47。





### 6.11.10 MBR (矩阵位循环移位)

类型	助记符	P	矩阵位循环移位																							
字	MBR	■	In1(W)								In2(W)								Out1(W)							
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□			■	■	■	■	■					

**参数说明:**

- 1, In1: 矩阵来源装置;
- 2, In2: 数组长度, 取值范围为 1~256;
- 3, Out1: 运算结果存放装置。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In2, 一个矩阵共有 16\*In2 个位(点), 其运算单位一次只有一个位(点)。矩阵指令是将 16\*In2 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点作运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右。
- 2, 矩阵的宽度固定为 16 位 (bits)。

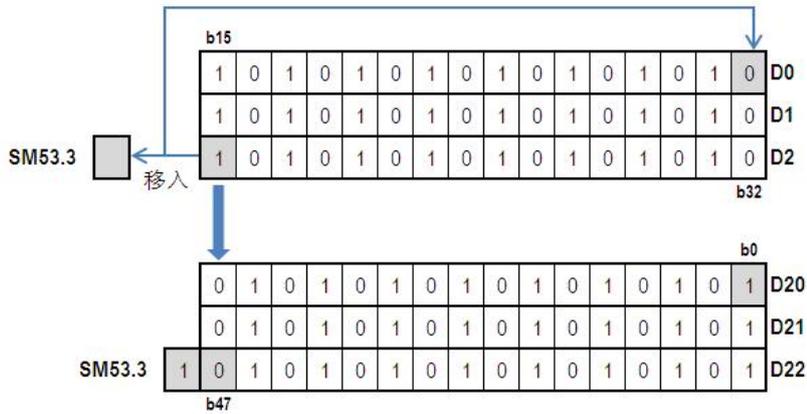
**指令功能说明:**

- 1, 矩阵来源依数组长度 In2 将 In1 矩阵位做左右移位控制, **左右移位标志位** SM53.5=0 决定矩阵位左移, SM53.5=1 矩阵位右移。每次移动一位, 因位移而腾出的空位(左移时为 b0, 右移时为 b16\*In2-1)则以因位移而挤出的位(左移时为 b16\*In2-1, 右移时为 b0) 状态补进, 同时也将挤出的位送到**进位标志** SM53.3 中去, 然后将结果存入 Out1。

程序范例:

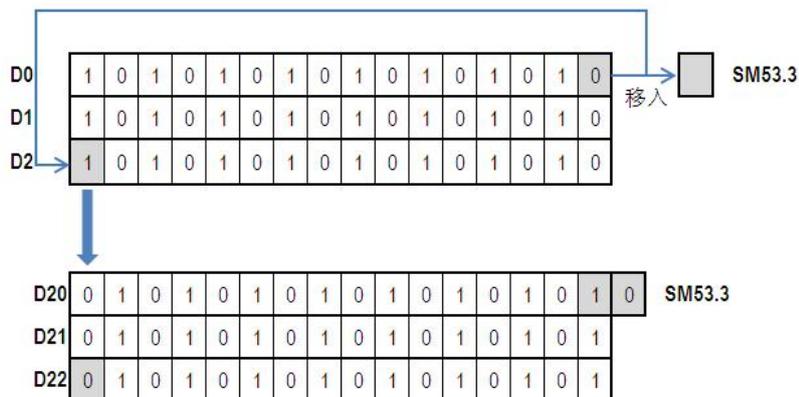
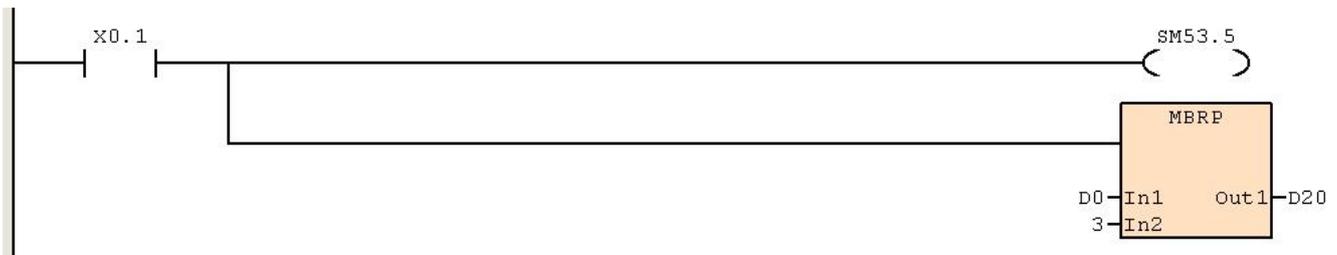
例 1:

- 当 X0.0=On 时, SM53.5=0 作矩阵左循环移位。16 位寄存器 D0~D2 矩阵作左循环移位, 将结果存于 16 位寄存器 D20~D22 矩阵中, 进位标志位 SM53.3=1。
- 第一行从右往左为 b0~b15, 第二行从右往左为 b16~b31, 第三行从右往左为 b32~b47。



例 2:

- 当 X0.1=On 时, SM53.5 = 1 作矩阵右循环移位, 16 位寄存器 D0~D2 矩阵作右循环移位, 将结果存于 16 位寄存器 D20~D22 矩阵中, 进位标志位 SM53.3=0。
- 第一行从右往左为 b0~b15, 第二行从右往左为 b16~b31, 第三行从右往左为 b32~b47。



### 6.11.11 MBC (矩阵位状态计数)

类型	助记符	P	矩阵位状态计数																							
字	MBC	■	In1(W)								In2(W)								Out1(W)							
双字																										
变量	位装置									字/双字装置																
	X	Y	S	M	T	C	H	L/	E/	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/		
			M				C	H	L			M				C	H	A	A	D	S	E	N	O	L	
In1										■	■	■	■	■	□		■	■	■	■	■	■				
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
Out1											■	■	■	■	□			■	■	■	■	■			■	

**参数说明:**

1, In1: 矩阵来源装置; In2: 数组长度, 取值范围为 1~256; Out1: 运算结果存放装置。

**指令执行注意点:**

- 1, 矩阵是 1 个以上的连续的 16 位寄存器所组成, 组成矩阵的寄存器个数称为矩阵的长度 In2, 一个矩阵共有 16\*In2 个位(点), 其运算单位一次只有一个位 (点)。矩阵指令是将 16\*In2 个矩阵位当做一连串单点的集合, 而自此集合中指定某一单点作运作, 而不将装置当做数值看待; 矩阵运作中有左, 右位移或旋转, 定义高编号者为左, 低编号者为右;
- 2, 矩阵的宽度固定为 16 位 (bits)。

**指令功能说明:**

1, 依数组长度 In2 计算 In1 矩阵中所有位为 1 或为 0 的个数, 并将数目存于 Out1 中; 当**计数类型模式** SM53.6 = 1 时, 计算矩阵位为 1 的个数, 当 SM53.6 = 0 时, 计算矩阵位为 0 的个数。

**程序范例:**

**例:**

当 X0.0 为 On 时, D0~D2 的矩阵中, 当 SM53.6 = 1 时计算矩阵位为 1 的位总数被存于 D10 当中。当 SM53.6 = 0 时计算矩阵位为 0 的位总数被存于 D10 当中。



1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	D0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	D1
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	D2

D10  SM53.6=0, D10存入0的个数

D10  SM53.6=1, D10存入1的个数

## 6.12 特殊功能指令

### 6.12.1 DELAY (延迟指令)

类型	助记符	P	延迟指令																				
字	DELAY	■	In1(W)																				
双字																							
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■

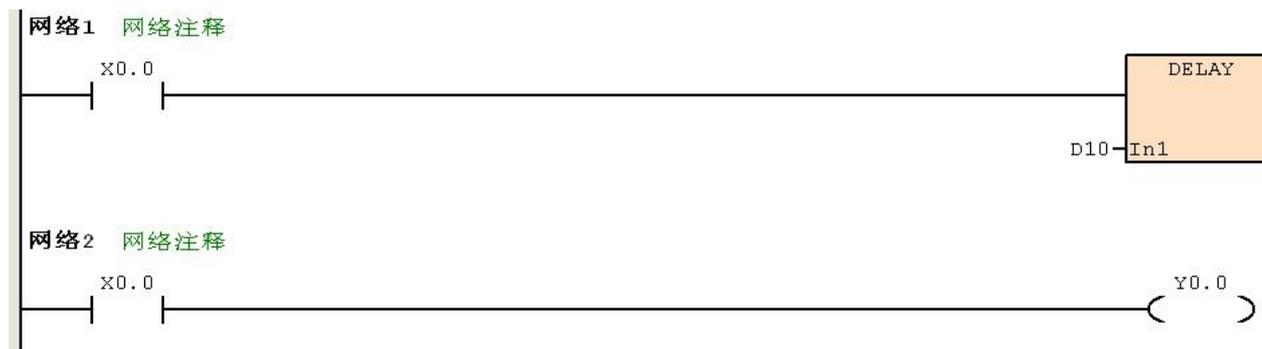
#### 指令说明:

- 1, In1: 延迟时间, 单位 0.1ms, 范围为 0~32767;
- 2, 在每次扫描至 DELAY 指令后面的程序执行会依使用者指定的时间作延迟。

#### 程序范例:

##### 例:

设定 D10=100, 当 X0.0 由 Off→On 时, Y0.0 依据 DELAY 所设定的时间延迟 10ms 后置 On。



### 6.12.2 GPWM (一般用脉冲波宽调变输出)

类型	助记符	P	一般用脉冲波宽调变输出																						
字	GPWM		In1(W)							In2(W)					In3(W)				Out1(b)						
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	E	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	E
			M				C	L				M					I	Q		D					
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In3										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Out1		■	■	■				■																	

**参数说明:**

1, In1: 脉冲输出宽度; In2: 脉冲输出周期; In3: 指令执行编号; Out1: 脉冲输出装置。

**指令执行注意点:**

- 1, In1 脉冲输出宽度指定 t = 0~32767ms;
- 2, In2 脉冲输出周期指定 T = 1~32767ms;
- 3, In3 指令执行编号为 0~31, 若指定同一个编号, 则共享该指令, 任一能流停止时都停止;
- 4, 当 In1 ≤ 0 时, 脉冲输出装置无输出, 当 In1 ≥ In2 时, 脉冲输出装置一直为 On;
- 5, 参数均可在 GPWM 执行时更改;
- 6, 该指令需要能流维持, 当能流为 Off 时, 指令停止执行, 该指令为周期性扫描执行, 只有在扫描到该指令时, 才更新输出, 因此会有一个扫描周期的误差。

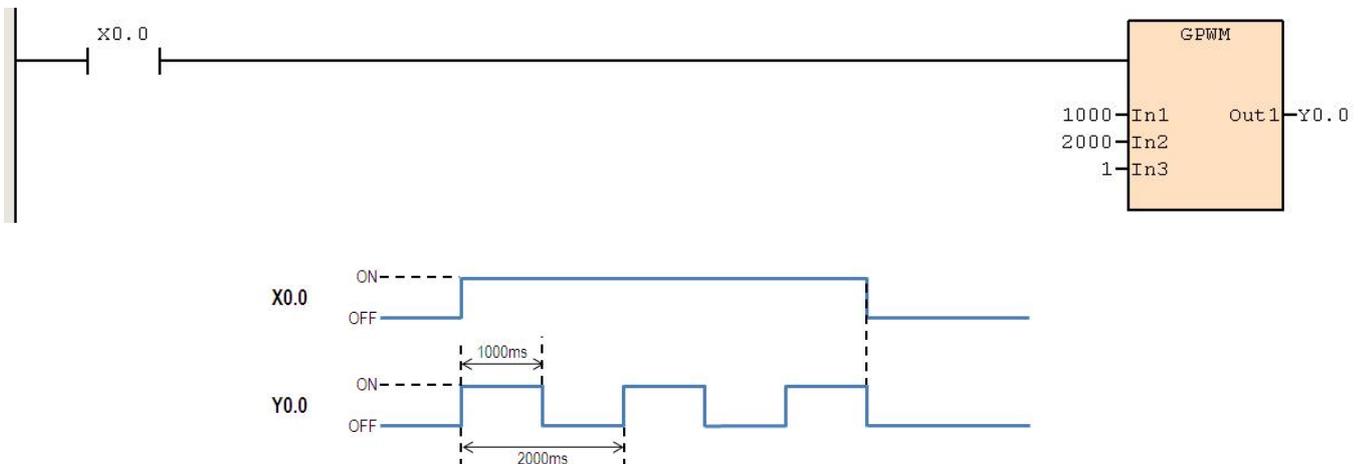
**指令功能说明:**

1, GPWM 指令执行时, 指定 In1 脉冲输出宽度与 In2 脉冲输出周期由 Out1 脉冲输出装置输出。

**程序范例:**

**例:**

当 X0.0=On 时, 以 1,000ms 脉冲输出宽度与 2,000ms 脉冲输出周期由 Y0.0 输出以下脉冲, 当 X0.0=Off 时, Y0.0 输出也变成 Off。



### 6.12.3 SWAP (上下字节/字互换)

类型	助记符	P	上下字节/字互换																				
字	SWAP	■	InOut1(W)																				
双字	DSWAP		InOut1(D)																				
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
InOut1										■	■	■	■	□	□		■	■	■	■			■

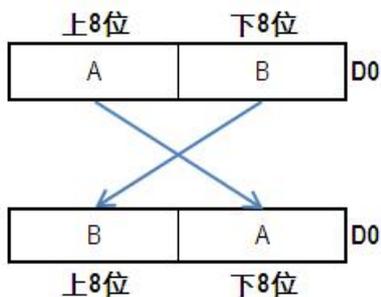
**指令说明:**

- 1, In1: 欲执行上下字节/字互换的装置;
- 2, 16 位指令时, 对 InOut1 所指定的装置进行上下字节互换;
- 3, 32 位指令时, 对 InOut1 所指定的装置进行上下字互换。

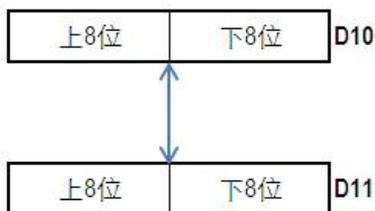
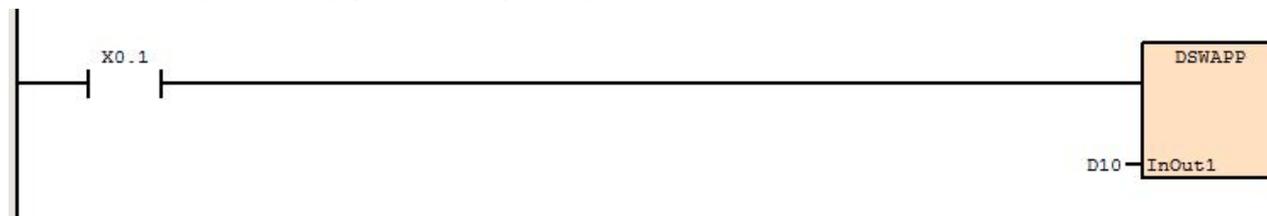
**程序范例:**

例:

当 X0.0=On 时, 将 D0 的上位 8 位与下位 8 位的内容互相交换。



当 X0.1=On 时, 将 D10 的内容与 D11 的内容互相交换。



### 6.12.4 PWD (输入脉宽检测)

类型	助记符	P	输入脉宽检测																							
字	PWD		In1(b)												Out1(W)											
双字	DPWD		In1(b)												Out1(D)											
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1	■																									
Out1										■	■	■	■	□	□		■	■	■	■				■		

**参数说明:**

- 1, In1: 指定的输入来源装置;
- 2, Out1: 存放检测结果的装置。

**指令执行注意点:**

- 1, Out1 单位为 0.1ms;
- 2, 该指令需要能流维持, 当能流断开时, 指令停止。

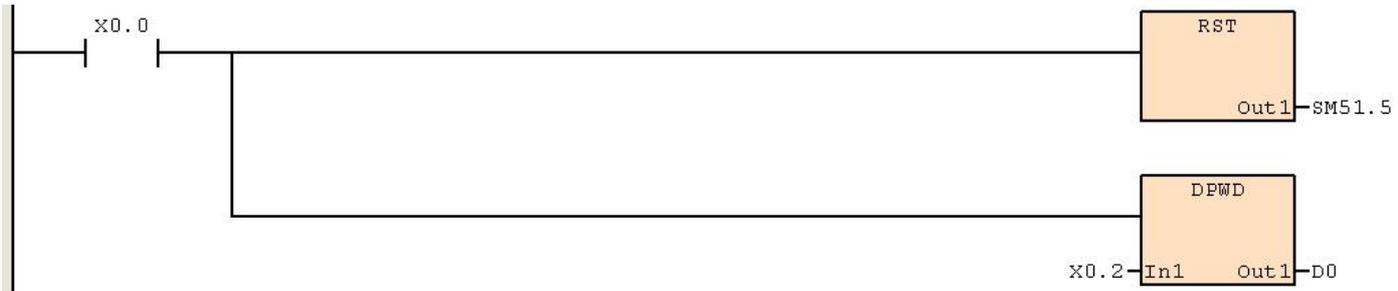
**指令功能说明:**

- 1, PWD 执行为检测 X0.0~X0.7 八个输入点信号的时间长度, 有效测量时间范围长度是大于一个扫描周期, 当**指令工作模式标志位 SM51.5** 为 Off 时, 为检测输入点信号连续上升沿与下降沿的时间长度, 当标志位 SM51.5 为 On 时, 为检测输入点信号, 连续两个上升沿的时间长度。

**程序范例:**

例:

当 X0.0=On 时, 将输入点 X0.2 上升沿与下降沿的时间长度记录后存放放到 D1、D0 中。



### 6.12.5 RTMU (程序执行时间测量开始)

类型	助记符	P	程序执行时间测量开始																						
字	RTMU		In1(W)																						
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1																							■		

**参数说明:**

1, In1: 该指令执行的编号。

**指令执行注意点:**

- 1, In1 的编号范围可设定为 0~9, 对应指定的 10 个 SD140~SD149 寄存器, 最多可同时嵌套测量 10 个中断/功能块, 若有重复使用编号的, 将测量最后一次使用该指令的时间;
- 2, 测量的时间单位为 0.1ms, 不足 0.1ms 的以 0.1ms 计算;
- 3, 该指令为直接接在左母线上, 单独一个网络段。

**指令功能说明:**

1, 该指令配合 RTMD 指令, 只有当执行 RTMD 指令时, 才会关闭对应的定时器, 同时把计算出来的时间指定给 RTMD 所指定的 SD140~SD149 中。

### 6.12.6 RTMD (程序执行时间测量结束)

类型	助记符	P	程序执行时间测量结束																						
字	RTMD		In1(W)																						
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1																							■		

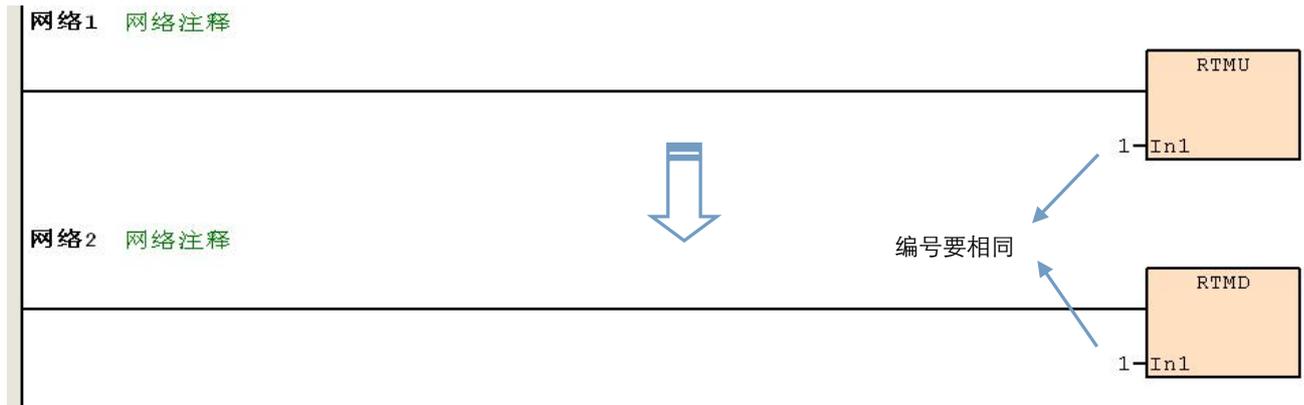
**指令说明:**

- 1, In1: 该指令执行的编号;
- 2, In1 的编号范围可设定为 0~9, 配合 RTMU 指令使用, 将测量 RTMU 开始到 RTMD 所经过的时间;
- 3, 提供开发者用于初期对程序时间参数的测定参考用;
- 4, 该指令为直接接在左母线上, 单独一个网络段。

**程序范例:**

例:

将 RTMU 到 RTMD 所经过的时间放入 SD141。



### 6.12.7 RAND (随机数值产生)

类型	助记符	P	随机数值产生																						
字	RAND	■	In1(W)							In2(W)							Out1(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
Out1										■	■	■	■	□			■	■	■	■				■	

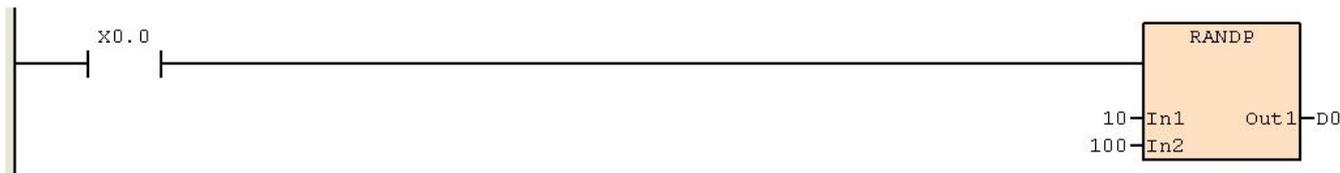
**指令说明:**

- 1, In1: 随机数产生的范围下限; In2: 随机数产生的范围上限; Out1: 随机数产生的结果;
- 2, In1 与 In2 的范围为 0~32767, 小于 0 以 0 计算;
- 3, 若使用者输入 In1 > In2, 则指令不执行。

**程序范例:**

例:

当 X0.0=On 时, RAND 指令产生介于范围下限 10 与范围上限 100 的随机数, 将结果存放 D0 内。



### 6.12.8 SCAL (比例值运算)

类型	助记符	P	比例值运算																						
字	SCAL	■	In1(W)							In2(W)					In3(W)					Out1(W)					
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
Out1										■	■	■	■	□			■	■	■	■	■			■	

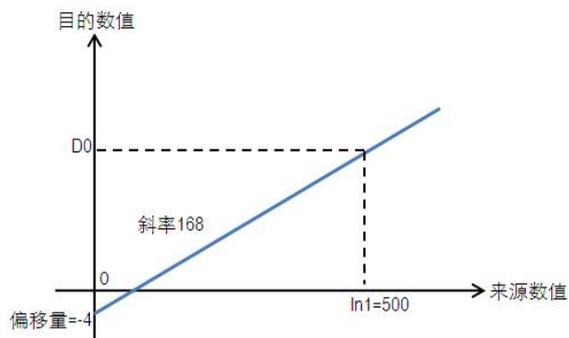
**指令说明:**

- 1, In1: 来源数值资料; In2: 斜率; In3: 偏移量; Out1: 计算目的装置;
- 2, 计算结果将会把小数点 4 舍 5 入, 取整后放入 Out1 装置中;
- 3, In1, In2 与 In3 的取值范围为-32768~32767, 其中 In2 的输入单位为 0.001;
- 4, 指令内部运算公式为  $Out1 = (In1 * In2) / 1000 + In3$ 。

程序范例：

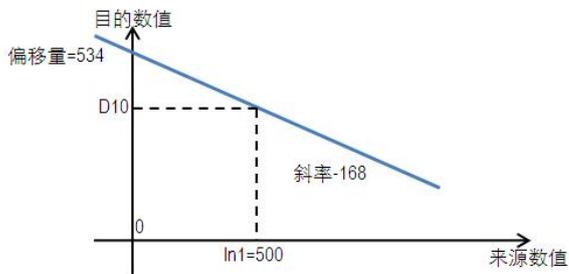
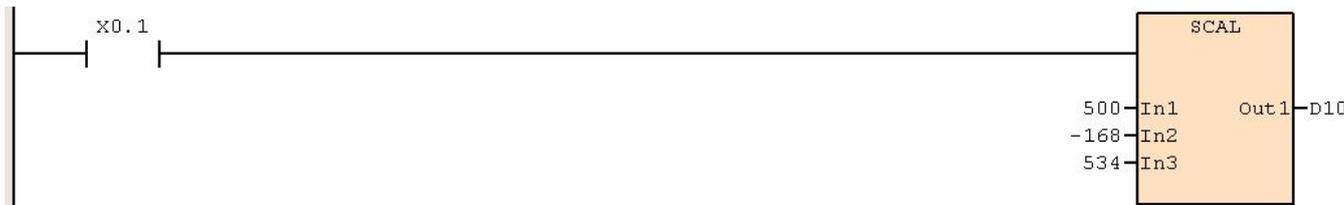
例 1：

- 已知 In1 数值资料来源为 500，In2 斜率为 168，In3 偏移量为-4，当 X0.0=On 时，SCAL 指令执行，可在 D0 得到所要求的比例值。
- 运算方式： $D0 = (500 \times 168) \div 1,000 + (-4) = 80$



例 2：

- 已知 In1 数值资料来源为 500，In2 斜率为-168，In3 偏移量为 534，当 X0.1=On 时，SCAL 指令执行，可在 D10 得到所要求的比例值。
- 运算方式： $D10 = (500 \times -168) \div 1,000 + 534 = 450$





### 6.12.9 SCLP (参数型比例值运算)

类型	助记符	P	参数型比例值运算																							
字	SCLP	■	In1(W)								In2(W)								Out1(W)							
双字	DSCLP		In1(D)								In2(D)								Out1(D)							
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■				
Out1										■	■	■	■	□	□		■	■	■	■				■		

#### 参数说明:

- 1, In1: 来源数值资料;
- 2, In2: 参数;
- 3, Out1: 计算目的装置。

#### 指令执行注意点:

- 1, 16 位指令 In2 操作数将连续占用 4 个字装置, 参数设置内容如下:

装置编号	参数名称说明	设置范围
In2	来源数值最大值	-32768~32767
In2+1	来源数值最小值	-32768~32767
In2+2	目的数值最大值	-32768~32767
In2+3	目的数值最小值	-32768~32767

- 2, 32 位指令 In2 操作数将连续占用 4 个双字装置, 参数设置内容如下:

装置编号	参数名称说明	设置范围
In2, In2+1	来源数值最大值	-2,147,483,648~2,147,483,647
In2+2, In2+3	来源数值最小值	--2,147,483,648~2,147,483,647
In2+4, In2+5	目的数值最大值	-2,147,483,648~2,147,483,647
In2+6, In2+7	目的数值最小值	-2,147,483,648~2,147,483,647

- 3, 计算结果将会把小数点 4 舍 5 入, 取整后放入 Out1 装置中。

#### 指令功能说明:

- 1, 指令内部运算公式为  $Out1 = [(In1 - \text{来源数值最小值}) * (\text{目的数值最大值} - \text{目的数值最小值})] / (\text{来源数值最大值} - \text{来源数值最小值}) + \text{目的数值最小值}$ ;
- 2, 来源数值和目的数值运算关系:

$y = kx + b$  式中:

$y =$  目的数值(Out1)

$k =$  斜率 = (目的数值最大值 - 最小值) / (来源数值最大值 - 最小值)

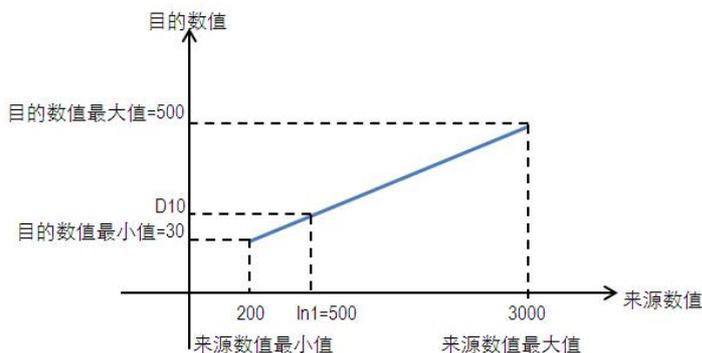
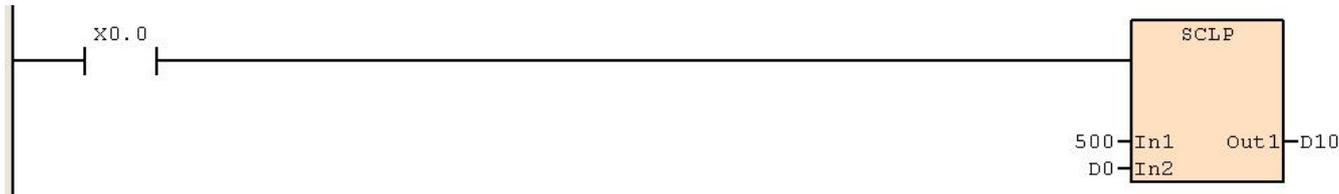
$x =$  来源数值 (In1)

$b =$  偏移量 = 目的数值最小值 - 来源数值最小值 \* 斜率

程序范例:

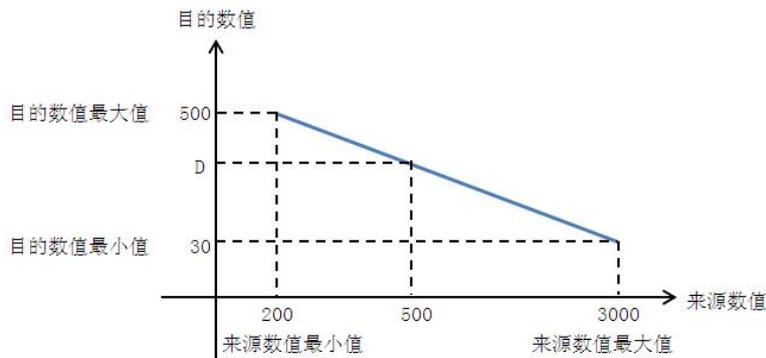
例 1:

- 已知 In1 数值资料来源为 500，来源数值最大值 D0=3,000，来源数值最小值 D1=200，目的数值最大值 D2=500，目的数值最小值 D3=30，当 X0.0=On 时，SCLP 指令执行，可在 D10 得到所要求的比例值。
- 运算方式： $D10 = [(500 - 200) \times (500 - 30)] \div (3,000 - 200) + 30 = 80.35$ ，取整数，D10 = 80。



例 2:

- 已知 In1 数值资料来源为 500，来源数值最大值 D0=3,000，来源数值最小值 D1=200，目的数值最大值 D2=30，目的数值最小值 D3=500 当 X0.0=On 时，SCLP 指令执行，可在 D10 得到所要求的比例值。
- 运算方式： $D10 = [(500 - 200) \times (30 - 500)] \div (3,000 - 200) + 500 = 449.64$ 。四舍五入取整数，D10 = 450。





### 6.12.10 WSUM (求和)

类型	助记符	P	求和																				
字	WSUM	■	In1(W)										In2(W)					Out1(W)					
双字	DWSUM		In1(D)										In2(W)					Out1(D)					
变量	位装置										字/双字装置												
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■
Out1										■	■	■	■	□	□		■	■	■	■			■

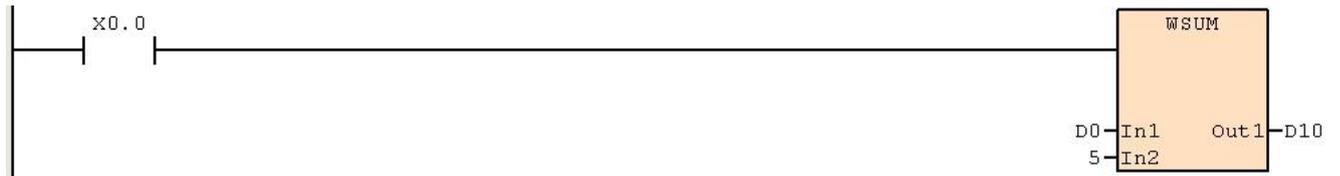
**指令说明:**

- 1, In1: 欲求和之起始装置; In2: 求和的装置个数; Out1: 计算目的装置;
- 2, In2 的取值范围为 1~64;
- 3, 该指令是以整型方式, 将 In1 起始的 In2 个装置内容相加后放入 Out1 中, 结果溢出不处理。

**程序范例:**

**例:**

当 X0.0 = On 时, D0 开始算的 5 个(In2=5)寄存器的内容全部相加, 相加之后存于指定的 D10 当中。



### 6.12.11 MTR (I/O 矩阵输入)

类型	助记符	P	I/O 矩阵输入																									
			In1(b)								In2(W)				Out1(b)				Out2(b)									
字	MTR																											
双字																												
变量	位装置										字/双字装置																	
	X	Y	S	M	T	C	H	L/	E	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	E	L	
In1	■																											
In2										■	■	■	■	■	□			■	■	■	■	■	■	■	■	■	■	■
Out1		■																										
Out2		■	■	■																								

**参数说明:**

1, In1: 矩阵扫描输入起始装置; In2: 矩阵扫描的行数; Out1: 矩阵扫描输出起始装置; Out2: 矩阵扫描所对应起始装置。

**指令执行注意点:**

- 1, In1 指定矩阵所有连接输入端的起始号码, 一旦指定后, 该号码开始算连续 8 点为矩阵输入端;
- 2, Out1 指定哪一个晶体管输出 Y 为矩阵扫描的起始号码;
- 3, 使用本指令时, In2 的取值范围是 1~8, 即最大可将 8 个输入开关并接 8 行, 可得 64 个输入点, X 和 Y 必须在本机点数范围之内;
- 4, 每一行的读取时间可在 SD100 中设置, 默认值为 20ms, 因此 On/Off 速度在 20\*In2 ms 内的输入信号不适用在矩阵输入;
- 5, 本指令需要能流维持, 正在执行时**指令执行标志** SM50.7 置起直到该指令执行完毕则 SM50.7 置 OFF; 当能流断开时, SM50.7 复位; 重新开始时, 则从头开始执行;
- 6, 本指令在所有程序中只能出现一次, 否则不可正常使用;
- 7, Out2 不可使用变址。

**指令功能说明:**

1, 本指令由 In1 起始的连续 8 个输入端, 以 Out1 起始的 In2 个外部输出点用矩阵扫描的方式读取 In2 行的 8 个开关状态, 变成 8\*In2 的多点矩阵输入点。并将扫描读取的开关状态反映在 Out2 起始的装置中。

**程序范例:**

当 X3.0=On 时, MTR 指令开始执行, 外部 2 行共 16 个开关的状态被顺序读取并存放在内部继电器 M0.0~M0.7, M1.0~M1.7。  
 当 X3.0=On 时, Y0.0 与 Y0.1 持续扫描; 当 X0.1=On, Y0.0=On, 则 M0.1=On; 当 X0.1=On, Y0.1=On, 则 M1.1=On。





### 6.12.12 HST (高速定时器)

类型	助记符	P	高速定时器
字	HST		无
双字			

**指令说明:**

- 1, 当能流持续 On 时, 启动高速定时器, 同时 HST 指令的高速计时 **正在执行标志位** SM51.6 置起 On, 高速定时器开始计时并将计时值记录在 SD101 中, SD101 的最小计时单位为 100us;
- 2, SD101 的计时范围是 0~32767, 当计时到 32767 时, 下一个计时是从 0 再开始, 每次开始执行, SD101 都将从当前值开始;
- 3, 当能流断开, 关闭高速定时器, 同时 SM51.6 置 OFF, SD101 立刻停止计时;
- 4, SD101 的更新时机是在扫描到执行 HST 指令时刻, 因此要使用 SD101 的数值前, 建议加入一个 HST 指令。

**程序范例:**

**例:**

- 当 X0.0 = On 时 SM51.6=On, 启动高速定时器开始计时并将计时值记录在 SD101 中。
- 当 X0.0 = Off 时 SM51.6=Off, 关闭高速定时器, 并保持当前值不变。
- 当 X0.0 再次置 On, SD101 从当前值开始计时。



### 6.12.13 PID (PID 运算)

类型	助记符	P	PID 运算																									
字	PID		In1(W)						In2(W)						In3(W)						InOut1(W)						Out1(W)	
双字	DPID		In1(W)						In2(D)						In3(D)						InOut1(D)						Out1(D)	
变量	位装置									字/双字装置																		
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L					
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■				
In2									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■				
In3									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■				
InOut1										■	■	■	■	□	□		■	■	■	■								
Out1										■	■	■	■	□	□		■	■	■	■				■				

**参数说明:**

- 1, In1: PID 使用的通道组;
- 2, In2: 设定目标值 (SV) ;
- 3, In3: 现在值来源 (PV) ;
- 4, InOut1: PID 调节的参数设定;
- 5, Out1: 输出值 (MV) 。

**指令执行注意点:**

- 1, PID 运算控制的专用指令, 当所有的参数设置完毕后, 在采样时间到达后的该次扫描才执行 PID 运算动作, 执行结果得到一个输出值 (MV), 该输出值的实际应用, 若是在数字开关上, 则配合 GPWM 指令用作占空比数值, 若是在模拟量输出上, 则可赋予 AQ 数值, 其中的 GPWM 周期和 AQ 上限设定要与 PID 的输出上限设定一致; 同时若使用 GPWM 输出, 建议将采样时间与 GPWM 周期时间设定一致;
- 2, 该指令需要能流维持, 能流断开时, 指令停止执行;
- 3, In1: 范围 0~31, 分别为通道 0~通道 31; 主机目前共支持 32 通道 PID 同时运算进行, 该参数用于定义当前 PID 运算使用的是第几通道进行运算控制, 请勿重复使用;
- 4, In3: 现在值来源如果是本机或扩展模拟量, 可直接使用 AI 装置;
- 5, InOut1: 连续占用 8 个字装置或双字装置, 具体内容安排如下

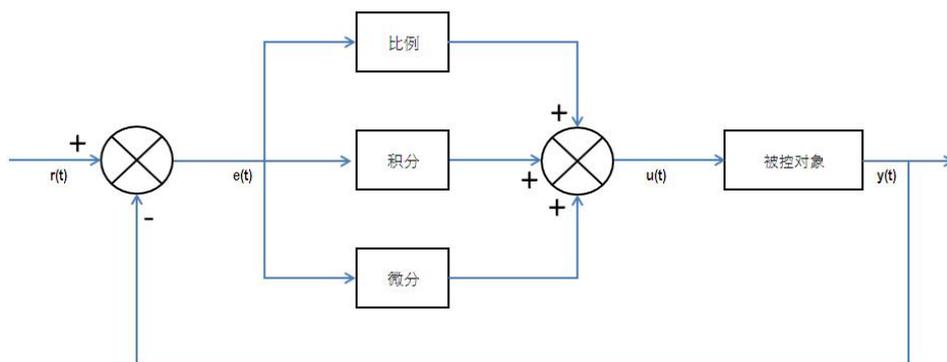
装置编号	功能	设定范围	备注说明
InOut1	动作方向	0: 正向动作 (E = SV - PV) 1: 逆向动作 (E = PV - SV) 其它无效	在使用自整定方式时, 该参数需要手动定义设置
InOut1+1	采样时间	16 位和 32 位均为: 1~32767 (单位: 10ms)	自整定时, 该值需要进行设定; 为本指令每多少时间计算更新输出值 (MV) 一次。当小于一次扫描周期时, PID 指令以一次扫描周期来执行, 非自整定时若设定为 0 时, 不动作。
InOut1+2	比例增益 (Kp)	16 位为: 1~32767 (%) 32 位为: 1~2147483647 (%)	SV-PV 间的误差放大比例值, 设定为小于等于 0 时, 不动作。



装置编号	功能	设定范围	备注说明
InOut1+3	积分时间 (Ti)	16 位和 32 位均为: 0~32767 (单位: 100ms)	小于等于 0 时作为无积分处理
InOut1+4	微分时间 (Td)	16 位和 32 位均为: 0~32767 (单位: 10ms)	小于等于 0 时作为无微分处理
InOut1+5	稳态偏差量 (E) 不作用范围	16 位: 0~32767 32 位: 0~2147483647	自整定时, 该值需要进行设定; 例如, 设置为 10, 那么当 PV-SV 的数值在 -10~10 之间时, 当前输出值 (MV) 将为输出值下限的值
InOut1+6	输出值 (MV) 设定上限	16 位: -32768~32767 32 位: -2147483648~2147483647	自整定时, 该值需要进行设定; 上限值必须大于等于下限值, 否则上下限值互换;
InOut1+7	输出值 (MV) 设定下限	16 位: -32768~32767 32 位: -2147483648~2147483647	自整定时, 该值需要进行设定;

### 指令功能说明:

1, PID 的计算公式如下



模拟 PID 控制系统原理图

该系统由模拟PID控制器和被控对象组成。

图中,  $r(t)$ 是给定值,  $y(t)$ 是系统的实际输出值, 给定值与实际输出值构成控制偏差 $e(t)$

$$e(t) = r(t) - y(t)$$

$e(t)$ 作为PID控制的输入,  $u(t)$ 作为PID控制器的输出和被控对象的输入。所以PID控制器的控制规律为

$$u(t) = K_p[e(t) + 1/T_i \int e(t)dt + (T_d)de(t)/dt]$$

其中:  $K_p$  —— 控制器的比例系数

$T_i$  —— 控制器的积分时间, 也称积分系数

$T_d$  —— 控制器的微分时间, 也称微分系数

运算结果  $u(t)$ 得到一个数值, 对其的应用:

(1) 模拟量输出: 将输出设定上下限值对应 AQ 的上下限值, 将  $u(t)$ 的当前值对应 AQ 的当前值即可;

(2) 开关点输出: 配合 GPWM 指令, 将输出上下限制对应周期长度值, 将  $u(t)$ 的当前值对应当前占空比数值即可。

2, 对温度控制一开始不知如何设定比例增益 ( $K_p$ ), 积分时间 ( $T_i$ ) 及微分时间 ( $T_d$ ), 可使用对应通道 SM60.0~SM63.7 的自整定启动标志位, 在设定好控制方向, 稳态偏差量 E 以及采样时间和输出值上下限后, 置位对应的 SM 标志位, 即可进行自整定运算, 其中, 如果输出值上限设定为小于等于 0 时, 输出值上限将默认被设定为 4000 (前提是输出值下限小于等于 0); 采样时间设定为小于等于 0 时, 采样时间将被设定为 4 秒; 当自整定运算完成, 对应的 SM 标志位会自动复位, 用户可使用下降沿来判断是否自整定完成; 当完成自整定后, 用户可继续手动进行参数微调, 来达到更佳的控制效果; 自整定前, 请保证系统维持在无输出的稳定状态下。

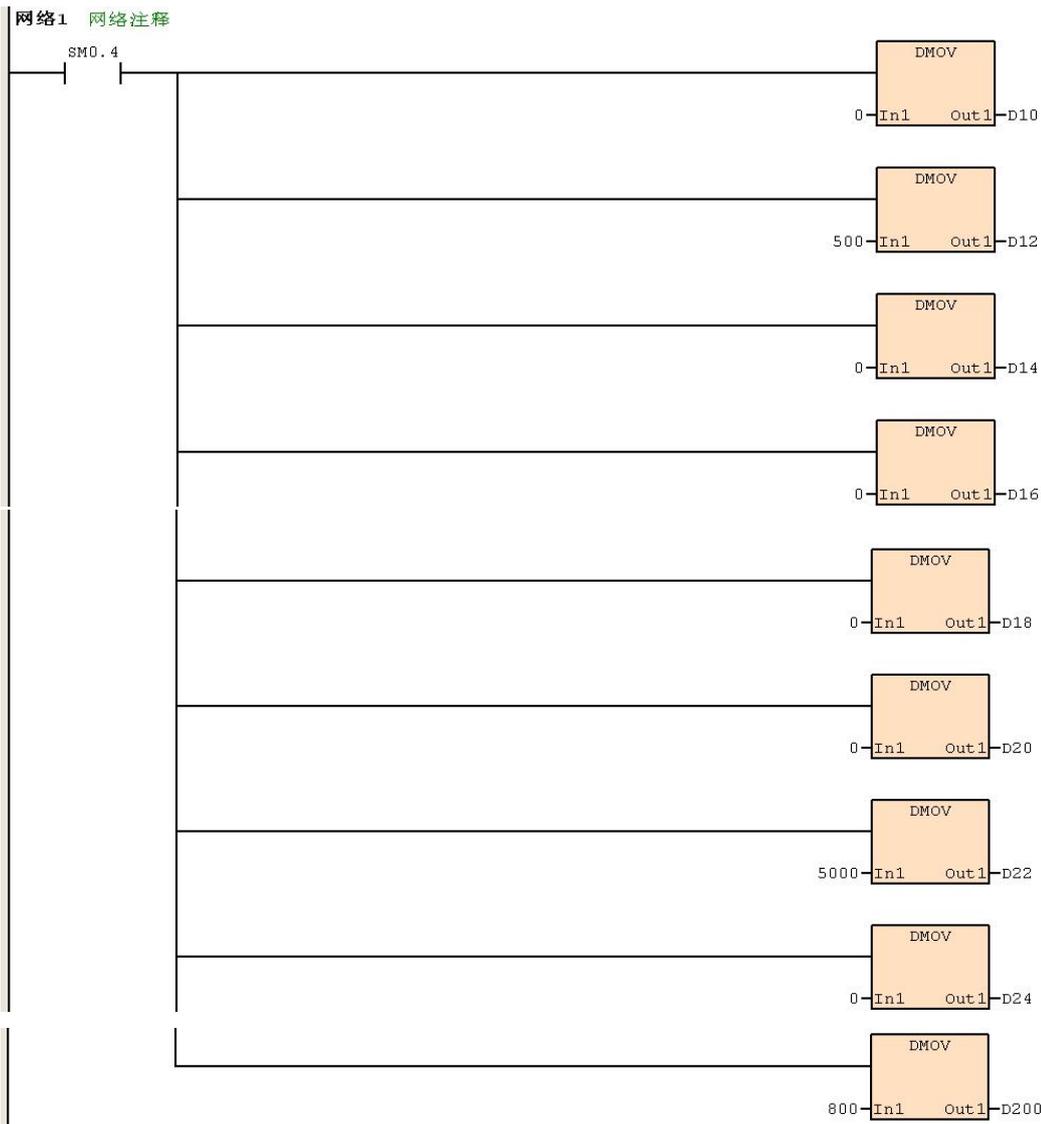
3, 自整定模式适用的控制对象为: 温度; 不适用的控制对象: 液位及流量等; 对于其它的应用, 可使用经验法进行参数设定, 一般使用的经验设定值如下表:

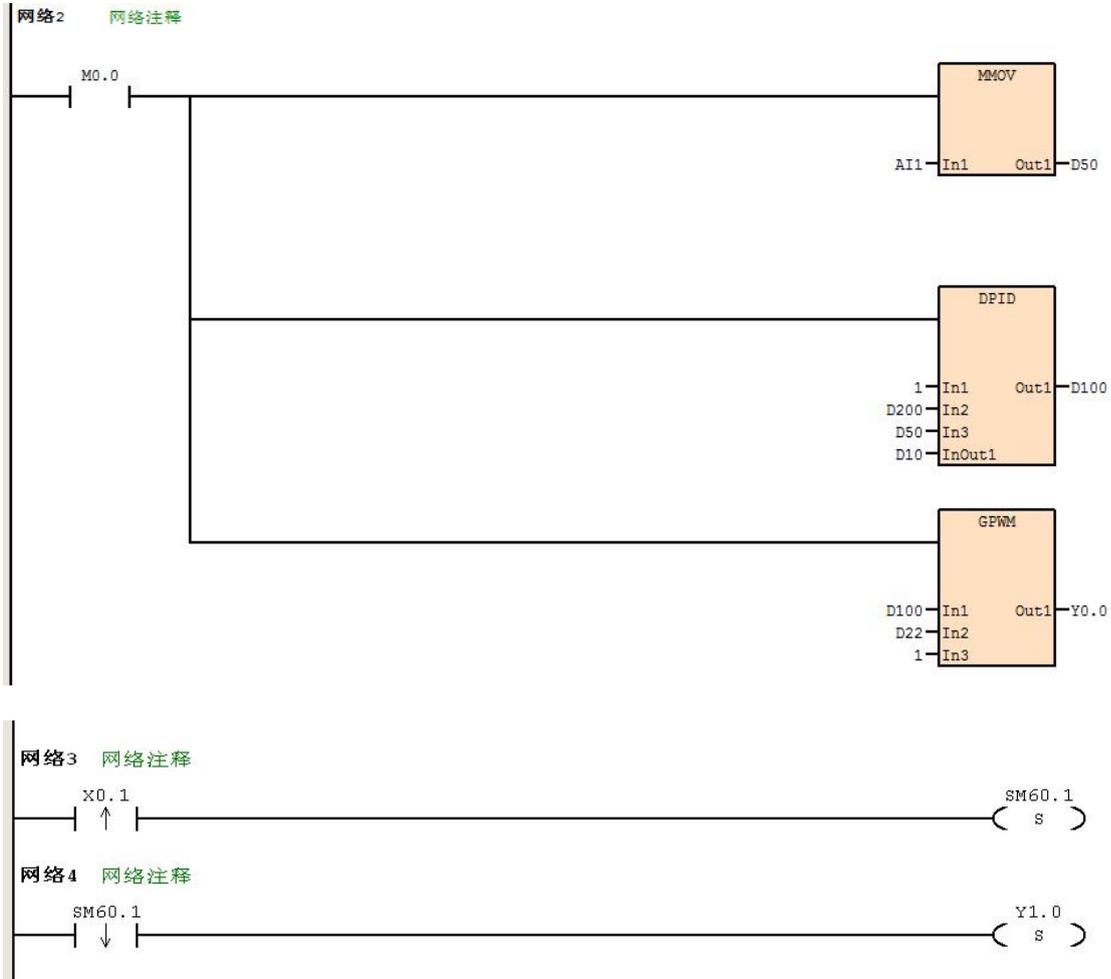
	Kp (%)	Ti (*100ms)	Td (*10ms)
温度系统	2000 ~ 60000	1800 ~ 6000	3000 ~ 18000
流量系统	4000 ~ 10000	60 ~ 600	0
压力系统	3000 ~ 7000	240 ~ 1800	0
液位系统	2000 ~ 8000	600 ~ 3000	0

**程序范例:**

例: 温度控制

- 程序运行时, 使用 PID 通道 1 将需要设定的参数一一赋值到相应的寄存器中, 其中动作方向设定为正方向, 采样时间设定为 5 秒, 稳态偏差 E 设定为 0, 输出值上限设定为 5000, 输出值下限设定为 0。
- 此时将 M0.0 置 On, 并将 D200 的值设定为所需要的温度值, 就开始进入等待自整定阶段了。
- 将 X0.1 置 On, 开始 PID 自整定, 此时 SM60.1 处于 On 状态。当自整定结束后, SM60.1 会自动复位, 此时 Y1.0 会被置 ON, 自整定完成后, 会将自整定计算出来的 Kp、Ti、Td 值赋值到相应的寄存器当中, 此时, PID 会根据计算出来的 Kp、Ti、Td 值来进行 PID 温度控制。





6.12.14 FAN (风扇控制)

类型	助记符	P	风扇控制																						
字	FAN		Out1(b)																						
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
Out1		■	■	■				■																	

指令说明:

- 1, Out1: 指定对应的输出对象;
- 2, 当能流持续 On 时, 启动风扇控制, 当检测到 PLC 机体温度大于 45°C 时, 输出 Out1 为 ON; 当检测到 PLC 机体温度小于 40°C 时, Out1 停止输出。

程序范例:

例:

- 当 X0.0 = On 时, 启动风扇控制, 当检测到 PLC 机体温度大于 45°C 时, Y0.0 输出 ON; 当检测到 PLC 机体温度小于 40°C 时, Y0.1 输出 OFF。
- 当 X0.0 = Off 时, 关闭风扇控制。





## 七 高速输入计数指令

### 7.1.1 HCNT (启动高速计数器)

类型	助记符	P	启动高速计数器																				
字																							
双字	HCNT		In1(D)										InOut1(b)										
变量	位装置										字/双字装置												
	X	Y	S	M	T	C	H	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L
In1									■	■	■	■	■	□	□	■	■	■	■	■	■	■	■
InOut1							■																

#### 参数说明:

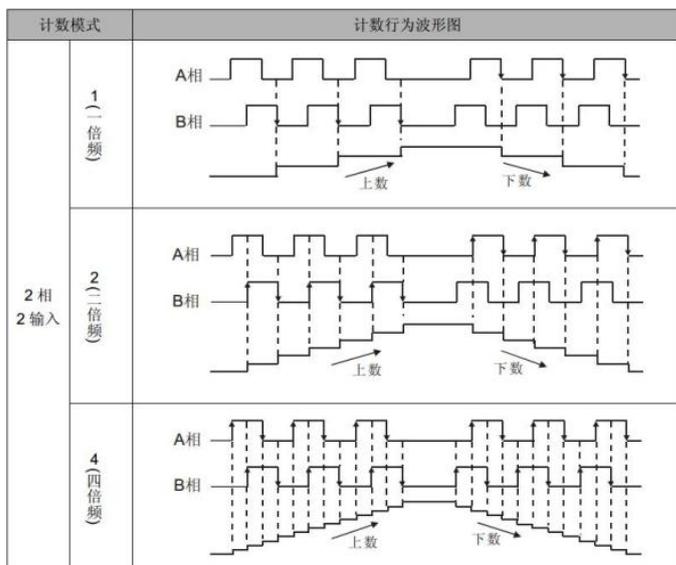
- 1, In1: 指定的计数数值;
- 2, InOut1: 指定的 HC 计数器, 不可使用变址装置。

#### 指令执行注意点:

- 1, InOut1 选择 HC0、HC1 时, 最高输入频率为 50kHz, 选择 HC2、HC3 时, 最高输入频率为 2.5kHz;
- 2, 当 HCNT 能流为 off 时, 该计数器停止计数, 但原有计数值不会被清除, 可使用指令 RST 清除计数值及其接点;
- 3, 当 HCNT 指令能流为 On 时, 对应的 HC0~HC3 **正在计数标志位** SM82.0~SM82.3 及 HC0~HC3 正在使用 HCNT 指令标志位 SM83.0~SM83.3 置 On, 当 HCNT 指令能流为 Off 时, 对应的 HC0~HC3 正在计数标志位 SM82.0~SM82.3 及 HC0~HC3 正在使用 HCNT 指令标志位 SM83.0~SM83.3 置 Off。

#### 指令功能说明:

- 1, 当 HCNT 能流为 On 时, 对应的输入点输入高速计数脉冲, 则执行计数动作, 到达设定数值后, InOut1 置 On, 同时停止计数, InOut1 保持为设定数值不变;
- 2, AB 相计数方式如图所示:



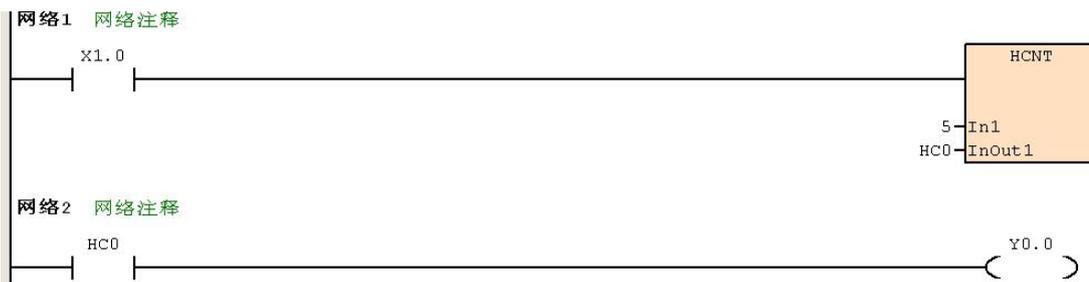
程序范例:

例 1: 单点高速计数输入

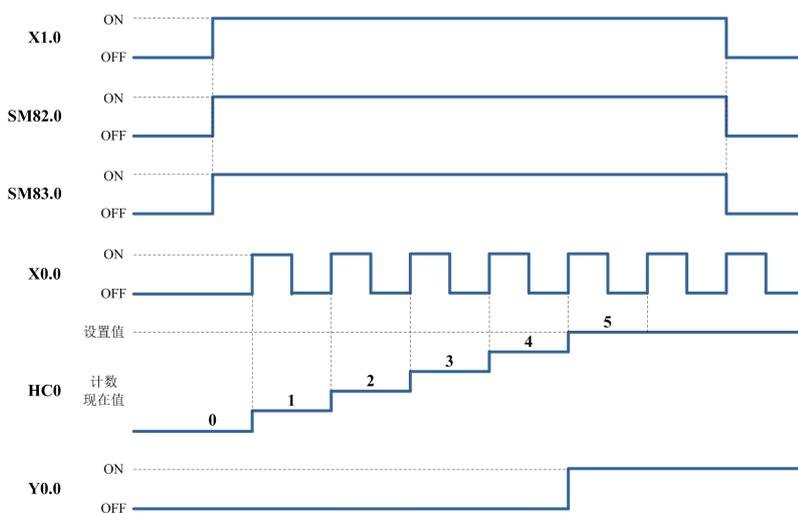
在硬件配置中的高速输入点功能设定如下图所示:

	X0.0	X0.1	X0.2	X0.3
普通输入	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
不可中断单点高速计数	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
可中断单点高速计数 (HC0)	<input type="radio"/>			
AB相高速计数1倍频	<input type="radio"/>	<input type="radio"/>		
AB相高速计数2倍频	<input type="radio"/>	<input type="radio"/>		
AB相高速计数4倍频	<input type="radio"/>	<input type="radio"/>		

说明: HC0和HC1最高100K, HC2和HC3最高2.5K Hz.



1. 当 X1.0=On, X0.0 由 Off→On 时, 计数器的现在值将执行上数 (加一) 的动作。
2. 当高速计数器 HC0 计数到达设置值 5 时, HC0 接点导通, HC0 现在值 = 设置值=5。之后的 X0.0 触发信号 HC0 完全不接受, HC0 现在值保持在 5 处。

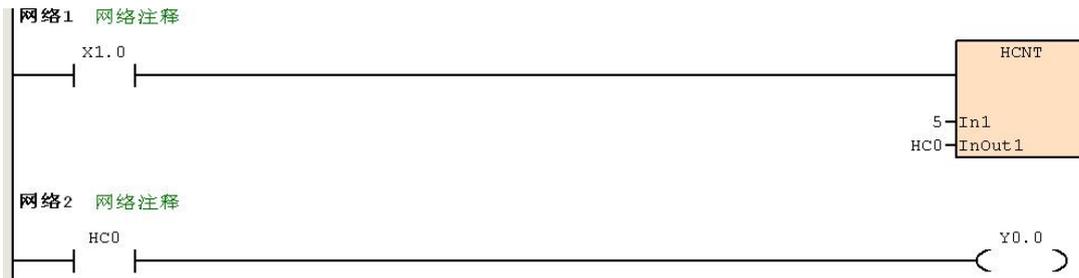


**例 2：AB 相高速计数输入时：**

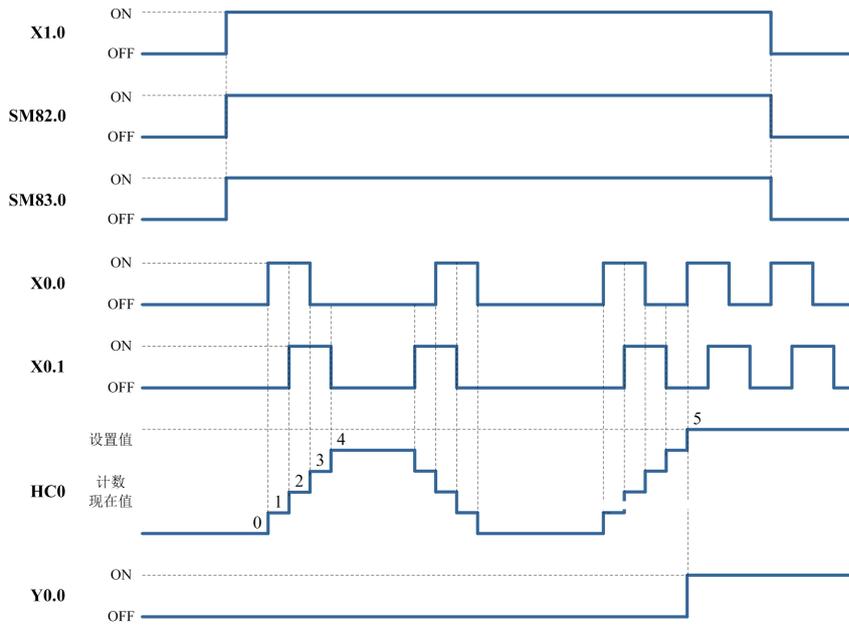
在硬件配置中的高速输入点功能设定如下图所示：

	X0.0	X0.1	X0.2	X0.3
普通输入	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
不可中断单点高速计数	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
可中断单点高速计数 (HCO)	<input type="radio"/>			
AB相高速计数1倍频	<input type="radio"/>	<input type="radio"/>		
AB相高速计数2倍频	<input type="radio"/>	<input type="radio"/>		
AB相高速计数4倍频	<input checked="" type="radio"/>	<input checked="" type="radio"/>		

说明：HCO和HC1最高100K，HC2和HC3最高2.5K Hz。



- 1, HC0 在 X1.0=On 时，接受 X0.0 输入端 A 相来的计数信号及 X0.1 输入端 B 相来的计数信号，计数器的现在值以 4 倍频执行上数（加一）或下数（减一）的动作。
- 2, 当计数器 HC0 计数到达设置值 5 时，HC0 接点导通，HC0 现在值 = 设置值=5。之后的 X0.0、X0.1 触发信号 HC0 完全不接受，HC0 现在值保持在 5 处。



### 7.1.2 HSCS (比较置位-高速计数器)

类型	助记符	P	比较置位 (高速计数器)																						
字																									
双字	HSCS		In1(D)									In2(D)									Out1(b)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L		X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1										■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
In2																	■								
Out1		■	■	■																					

**参数说明:**

- 1, In1: 比较值;
- 2, In2: 高速计数器编号;
- 3, Out1: 比较结果。

**指令执行注意点:**

- 1, 若 Out1 所指定的装置为 Y 装置时, 当比较值与高速计数器当前值相等时, 会实时输出到外部的 Y 输出端口中; 同样其它的装置也将实时的变更当前状态, 不受扫描周期影响;
- 2, 该指令在 HCx 设置为非中断高速计数器类型时, 可有效使用, 若设置成中断类型或者没有设置高速计数器, 将可能无法正常使用;
- 3, 在使用该指令之前, **必须先使用 HCNT 指令**将对应的 HCx 高速计数器启动计数;
- 4, 该指令执行需要能流维持, 当能流断开, 则该指令停止;
- 5, Out1 不可使用变址。

**指令功能说明:**

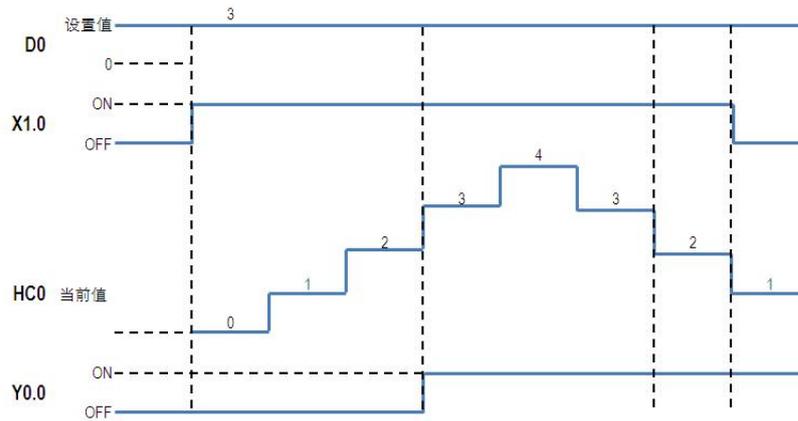
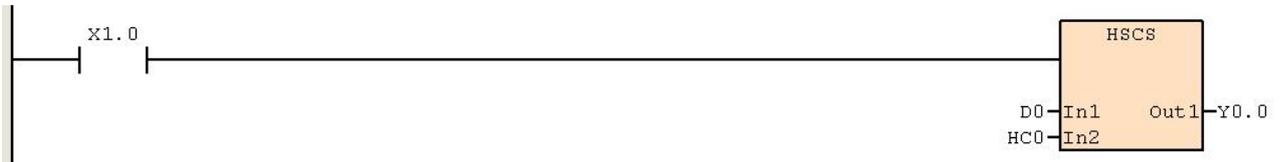
- 1, 当由 HSCS 指令所指的 In2 高速计数器产生加 1 或者减 1 的变化时, HSCS 指令会立即做比较动作, 当计数器当前值等于设定的 In1 的比较值时, 由 Out1 所指定的装置会变成 On, 之后即使比较结果变成不相等, 该装置依然保持 On 的状态。



**程序范例:**

**例:**

当 PLC 执行 RUN 指令后,若 X1.0=On,HSCS 指令执行,当 HC0 的现在值由(D0-1)→D0 或(D0+1)→D0 变化时,Y0.0=On,实时输出到外部 Y0.0 输出端,且一直保持为 On。



### 7.1.3 HSCR (比较复位-高速计数器)

类型	助记符	P	比较复位 (高速计数器)																						
字																									
双字	HSCR		In1(D)									In2(D)									Out1(b)				
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2																■									
Out1		■	■	■																					

**参数说明:**

- 1, In1: 比较值;
- 2, In2: 高速计数器编号;
- 3, Out1: 比较结果。

**指令执行注意点:**

- 1, 若 Out1 所指定的装置为 Y 装置时, 当比较值与高速计数器当前值相等时, 会实时输出到外部的 Y 输出端口中; 同样其它的装置也将实时的变更当前状态, 不受扫描周期影响;
- 2, 该指令在 HCx 设置为非中断高速计数器类型时, 可有效使用, 若设置成中断类型或者没有设置高速计数器, 将可能无法正常使用;
- 3, 在使用该指令之前, **必须先使用 HCNT 指令**将对应的 HCx 高速计数器启动计数;
- 4, 该指令执行需要能流维持, 当能流断开, 则指令停止;
- 5, Out1 不可使用变址。

**指令功能说明:**

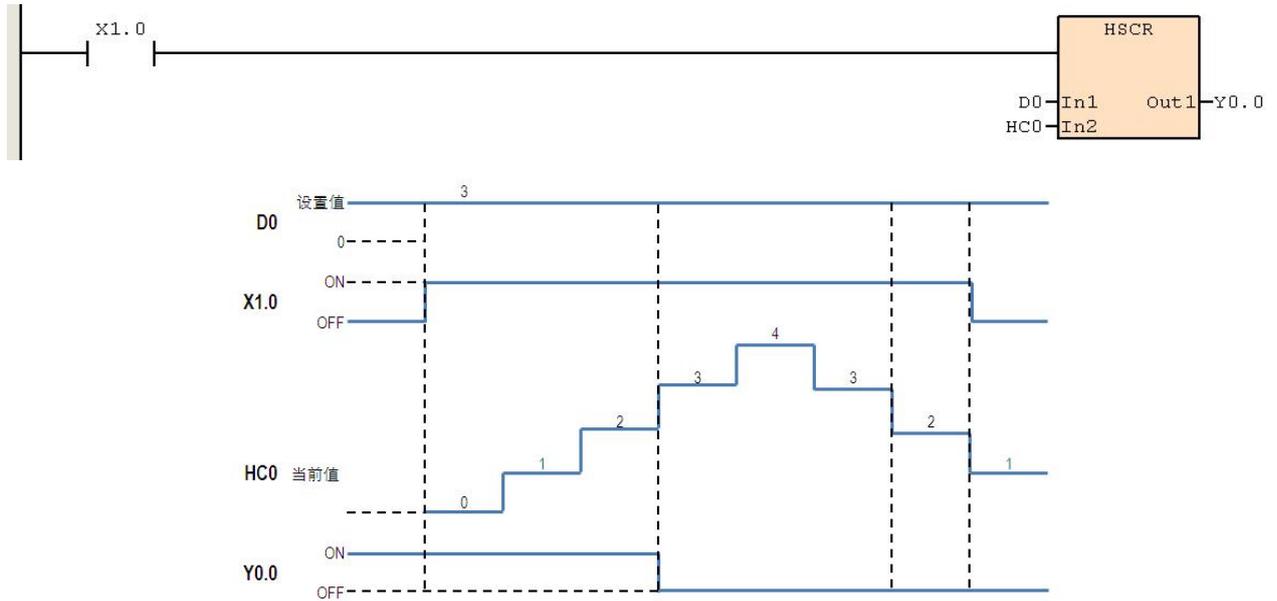
- 1, 当由 HSCR 指令所指的 In2 高速计数器产生加 1 或者减 1 的变化时, HSCR 指令会立即做比较动作, 当计数器当前值等于设定的 In1 的比较值时, 由 Out1 所指定的装置会变成 Off, 之后即使比较结果变成不相等, 该装置依然保持 Off 的状态。



**程序范例:**

**例:**

当 PLC 执行 RUN 指令后,若 X1.0=On,HSCR 指令执行,当 HC0 的现在值由(D0-1)→D0 或(D0+1)→D0 变化时,Y0.0=On 会被清除 Off。



### 7.1.4 HSZ (区间比较-高速计数器)

类型	助记符	P	区间比较 (高速计数器)																						
字																									
双字	HSZ		In1(D)							In2(D)							In3(D)							Out1(b)	
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L		
In1									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
In3															■										
Out1		■	■	■																					

**参数说明:**

- 1, In1: 区间比较下限值;
- 2, In2: 区间比较上限值;
- 3, In3: 高速计数器编号;
- 4, Out1: 比较结果。

**指令执行注意点:**

- 1, 如果 In1 > In2 的设置值, 那么将取 In1 做为上下限值;
- 2, Out1 将连续占用三个装置, 若 Out1 所指定的装置为 Y 装置时, 当比较值与高速计数器当前值相等时, 会实时输出到外部的 Y 输出端口中; 同样其它的装置也将实时的变更当前状态, 不受扫描周期影响;
- 3, 该指令在 HCx 设置为非中断高速计数器类型时, 可有效使用, 若设置成中断类型或者没有设置高速计数器, 将可能无法正常使用;
- 4, 在使用该指令之前, **必须先使用 HCNT 指令**将对应的 HCx 高速计数器启动计数;
- 5, 该指令执行需要能流维持, 当能流断开, 则指令停止;
- 6, Out1 不可使用变址。

**指令功能说明:**

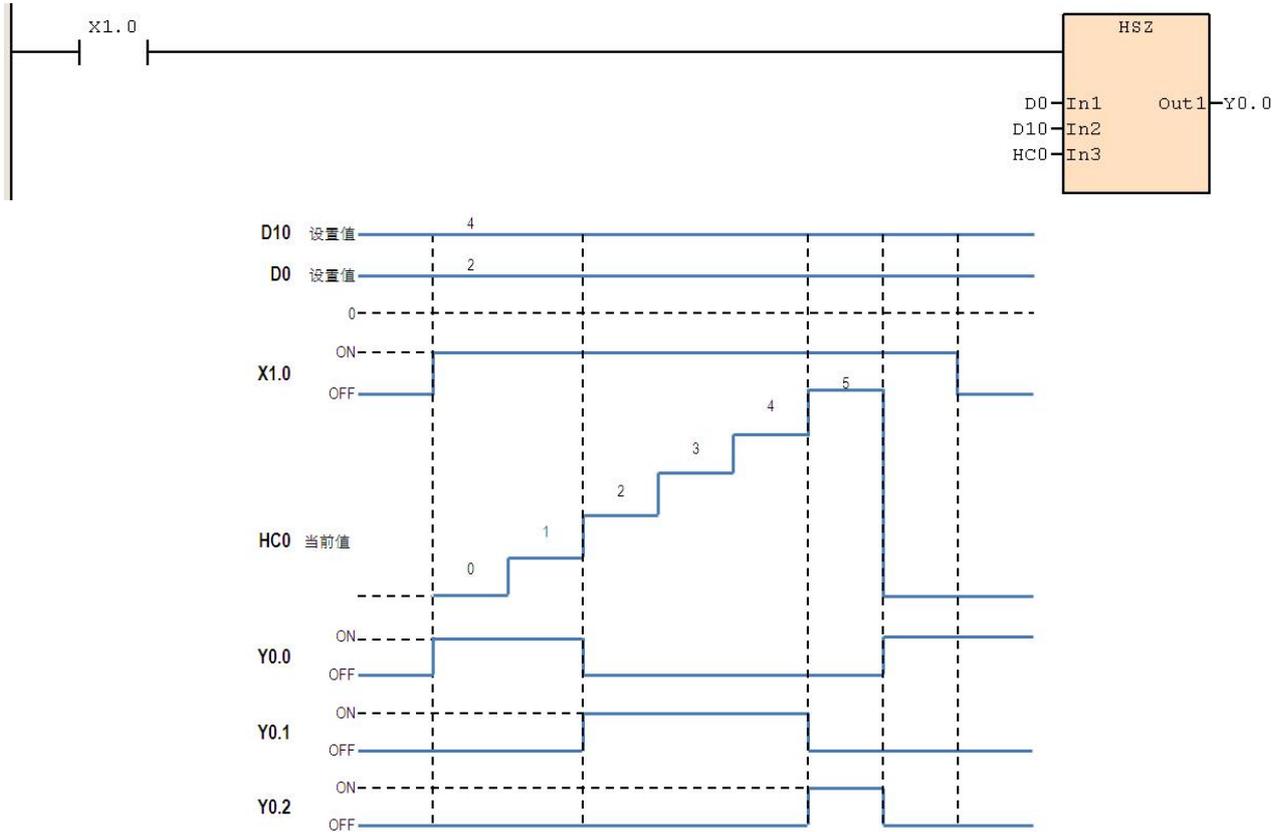
- 1, 当由 HSZ 指令所指的 In3 高速计数器产生加 1 或者减 1 的变化时, HSZ 指令会立即做比较动作:
  - 1) 当计数器当前值小于设定的 In1 的区间比较下限值时, 由 Out1 所指定的装置会变成 On, 而 Out1+1 及 Out1+2 所指定的装置会变成 Off;
  - 2) 当计数器当前值大于等于 In1 并且小于等于 In2, 则 Out1+1 所指定的装置会变成 On, 而 Out1 及 Out1+2 所指定的装置会变成 Off;
  - 3) 当计数器当前值大于 In2, 则 Out1+2 所指定的装置会变成 On, 而 Out1 及 Out1+1 所指定的装置会变成 Off。



**程序范例:**

**例:**

指定装置为 Y0.0, 则自动占有 Y0.0~Y0.2。当 HSZ 指令执行时, 高速计数器 HC0 在有计数输入时, 到达上下限值, Y0.0~Y0.2 其中的一个会 On。HC0 现在值 < D0 时, Y0.0 为 On;  $D0 \leq HC0$  现在值  $\leq D10$  时, Y0.1 为 On; HC0 现在值 > D10 时, Y0.2 为 On。



### 7.1.5 SPD (定时脉冲个数检测)

类型	助记符	P	定时脉冲个数检测																						
			In1(W)							InOut1(b)							InOut2(W)								
字	SPD																								
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	E/	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	
			M			C	C	L			M					C	I	Q		D		D		L	
In1									■	■	■	■	■	□			■	■	■	■	■	■	■	■	
InOut1																									
InOut2										■	■	■	■	□			■	■	■	■					

**参数说明:**

- 1, In1: 接收脉冲时间, 范围 1~32767ms, 计时数值在执行中不可变更, 在下一次执行时才有效;
- 2, InOut1: 高速输入通道选择;
- 3, InOut2: 检测结果, 连续占用 3 个字装置。

**指令执行注意点:**

- 1, 高速输入通道选择, 必须已经对外部输入端口进行功能设置, 则将会按照所对应的设置功能, 进行一段时间内的脉冲计数;
- 2, 如果检测的是 AB 相脉冲, 则如果得到结果是正值, 表明 A 相领先 B 相, 反之是 B 相领先 A 相;
- 3, SPD 指令和 HCNT 指令对应的 HC 只可同时执行一个;
- 4, 该指令需要能流持续, 当执行过程中, 能流 off, 则指令停止;
- 5, 当 SPD 指令能流为 On 时, 对应的 HC0~HC3 **正在计数标志位** SM82.0~SM82.3 及 HC0~HC3 **正在使用 SPD 指令标志位** SM84.0~SM84.3 置 On, 当 SPD 指令能流为 Off 时, 对应的 HC0~HC3 **正在计数标志位** SM82.0~SM82.3 及 HC0~HC3 **正在使用 HCNT 指令标志位** SM84.0~SM84.3 置 Off;
- 6, 若指令结束, 在下一个扫描周期执行该指令时, 如果能流依然为 On, 重新开始执行该指令, HC 和存放结果数值清 0;
- 7, InOut1 和 InOut2 不可使用变址装置。

**指令功能说明:**

1, 该指令的功能即是, 在 In1 指定的时间 (单位 ms) 内, 计算 InOut1 对应的输入端所接收脉冲个数, 计时结束后, 输入个数结果放置于 InOut2 所指定的连续两个字装置 (即双字装置) 中, InOut2+2 所指定的装置将存放计时剩余时间。

**程序范例:**

**例:**

在 10000ms 内, 计算 HC0 对应的输入端所接收脉冲个数, 计时过程中, D2 将存放计时剩余时间; 计时结束, 检测的脉冲个数放置于 D0(D1)双字装置中, 脉冲个数存放一个扫描周期, 下一个扫描周期将会重新置 0。





## 八 高速脉冲输出指令

## 九 串行通讯指令

### 9.1 自由通讯协议指令

#### 9.1.1 RS (串行数据传送)

类型	助记符	P	串行数据传送																																		
字	RS	■	In1(W)							In2(W)							In3(W)							In4(W)							Out1(W)						
双字																																					
变量	位装置											字/双字装置																									
	X	Y	S	M	T	C	H	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L														
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■														
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■														
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■														
In4									■	■	■	■	■	□		■	■	■	■	■	■	■	■														
Out1										■	■	■	■	□			■	■	■	■																	

**参数说明:**

- 1, In1: PLC 主机的通讯口, 所指定的值必须为 1 或者 2, 其它任何值无效;
- 2, In2: 传送数据的起始装置;
- 3, In3: 传送数据的个数, 数据长度为 0~50 个;
- 4, In4: 接收数据的个数, 数据长度为 0~50 个;
- 5, Out1: 接收数据的起始装置。

**指令执行注意点:**

- 1, 若不需要传送数据时, 可将 In3 指定为数据 0, 若不需要接收数据时, 可将 In4 指定为数据 0;
- 2, 使用前必须先置 ON 启用 RS 指令开关 SM70.x 启用 RS 指令;
- 3, 通讯指令完全结束后, 通讯结束标志位 SM65.x, 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断通讯异常标志位 SM67.x 是否置起, 及读取错误码 SD245~SD246;
- 4, 当数据发送开始, 正在发送标志位 SM66.x 置位为 On, 数据发送完毕, 正在发送标志位 SM66.x 复位为 Off;
- 5, 如果通讯成功无异常, 数据存放指定的 Out1 装置中, 且通讯异常标志位 SM67.x 为 OFF; 若有异常, 则通讯异常标志位 SM67.x 置 On, 同时通讯错误码放于对应 SD245~SD246 中, 数据存放中的数据无效;
- 6, 当 RS 指令的处理数据模式 SM71.x 为 On 时, 将以每个装置的低位字节进行发送/接收, 高位字节的数据不去处理, SM71.x 为 Off 时, 先发送/接收该装置的高字节的数据, 之后发送/接收该装置的低位字节的数据, 随后发送/接收下一个装置的高字节数据, 之后再低位字节数据, 依次直到发送/接收长度完毕;
- 7, 该指令为自由协议通讯方式, 不受 MODBUS 的 ASCII 或 RTU 格式影响, 也不做任何校验动作;
- 8, 完整的接收数据的个数 (单位字节), 放置于使用 RS 指令接收到的数据字节数装置 SD225~SD226 中;
- 9, 发送和接收时可使用 RS 指令起始字符启用开关 SM68.x、RS 指令起始字符 SD230~SD231 的低字节指定起始字符是否使用, RS 指令结束字符启用开关 SM69.x、RS 指令结束字符 SD235~SD236 的低字节指定结束字符是否使用:
  - 1) 当使用起始字符, 则在接收起始字符后, 开始接收有效数据, 起始字符不计算在接收长度内; 在发送数据时, 自动在第一个字节填充起始字符, 起始字符不计算在发送长度内;
  - 2) 当使用结束字符时, 忽略接收字符长度 (但是仍不可大于 40), 同时以接收到结束字符完成接收动作; 在发送数据时, 自动在最后一字节填充结束字符, 结束字符不计算在发送长度内。

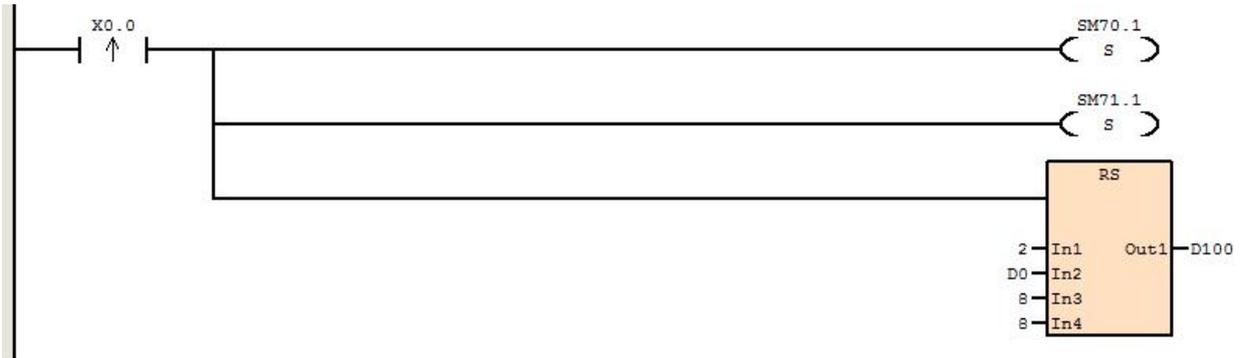
**指令功能说明:**

1, 此指令专为主机使用 RS485 串行通讯接口所提供的便利指令,只要在 In2 来源数据寄存器事先存入字资料并设置长度 In3, 并设置接收数据寄存器 Out1 及长度 In4, 即可在指定 In1 通讯口中输出指定的串行数据。

**程序范例:**

**例 1:**

- RS485\_2 的 RS 指令处理数据模式 SM71.1 为 On 时, 进行 8 位处理模式。(外部机器为 RTU 模式)
- 先将发送数据预先写入 D0 开始的寄存器内, 当 X0.0 由 Off 到 On 时, PLC 执行 RS 指令即进入等待传送、接收数据的状态。开始执行 D0 开始连续 8 笔发送数据送出, 如果通讯成功无异常, 回复数据则存入由 D100 开始的连续寄存器内。



PLC 传送数据寄存器

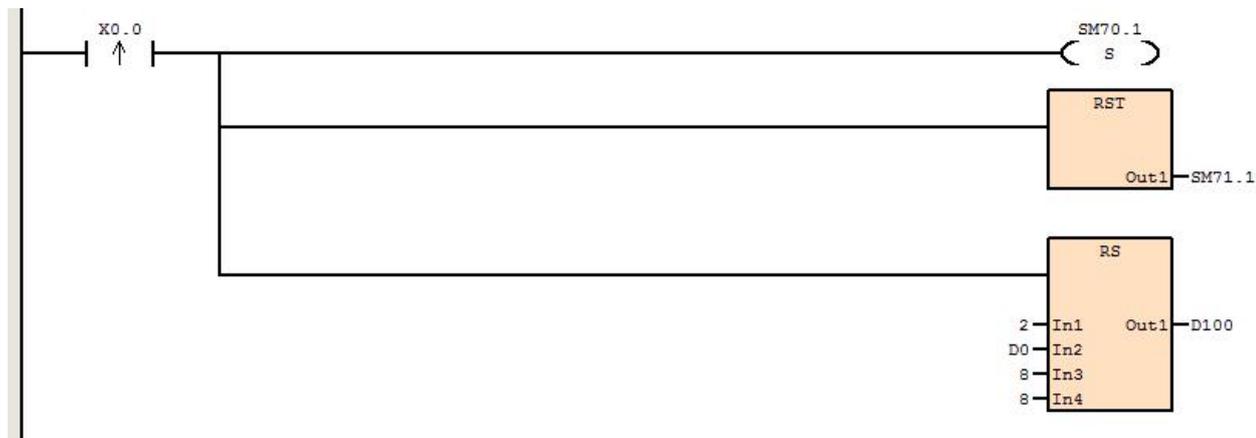
寄存器	数据	说明
D0 下	16#01	地址
D1 下	16#06	功能码
D2 下	16#10	数据地址
D3 下	16#00	
D4 下	16#00	数据内容
D5 下	16#12	
D6 下	16#0D	CRC CHK Low
D7 下	16#07	CRC CHK High

PLC 接受数据寄存器

寄存器	数据	说明
D100 下	16#01	地址
D101 下	16#06	功能码
D102 下	16#10	数据地址
D103 下	16#00	
D104 下	16#00	数据内容
D105 下	16#12	
D106 下	16#0D	CRC CHK Low
D107 下	16#07	CRC CHK High

例 2:

- COM2 的 RS 指令处理数据模式 SM71.1 为 Off 时，进行 16 位处理模式。（外部机器为 RTU 模式）
- 先将发送数据内容预先写入 D0 开始的寄存器内，当 X0.0=On 时，PLC 执行 RS 指令即进入等待传送、接收数据的状态。开始执行 D0 开始连续 8 笔发送数据送出，当所有过程结束后，如果通讯成功无异常，回复数据则存入由 D100 开始的连续寄存器内。



PLC 传送数据寄存器

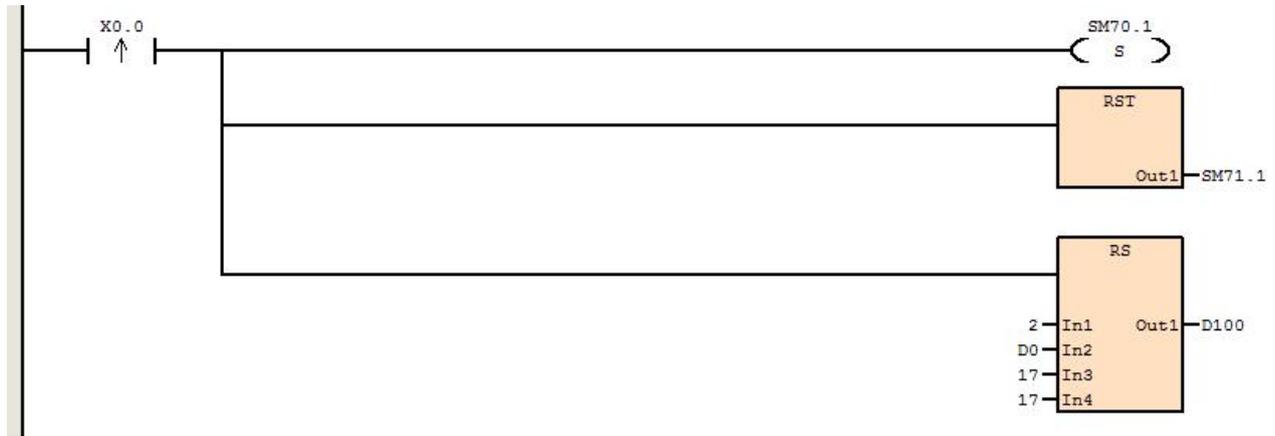
寄存器	数据	说明
D0 上	16#01	地址
D0 下	16#06	功能码
D1 上	16#10	数据地址
D1 下	16#00	
D2 上	16#00	数据内容
D2 下	16#12	
D3 上	16#0D	CRC CHK Low
D3 下	16#07	CRC CHK High

PLC 接受数据寄存器

寄存器	数据	说明
D100 上	16#01	地址
D100 下	16#06	功能
D101 上	16#10	数据地址
D101 下	16#00	
D102 上	16#00	数据内容
D102 下	16#12	
D103 上	16#0D	CRC CHK Low
D103 下	16#07	CRC CHK High

例 3:

- COM2 的 RS 指令处理数据模式 SM71.1 为 Off 时，进行 16 位处理模式。（外部机器为 ASCII 模式）
- 先将发送数据内容预先写入 D0 开始的寄存器内，当 X0.0=On 时，RS 指令执行 PLC 即进入等待传送、接收数据的状态。开始执行 D0 开始连续 17 笔发送数据送出，当所有过程结束后，如果通讯成功无异常，回复数据则存入由 D100 开始的连续寄存器内。



PLC 传送数据寄存器

寄存器	数据	说明
D0 上	‘:’ 16#3A	STX (头码)
D0 下	‘0’ 16#30	地址
D1 上	‘1’ 16#31	
D1 下	‘0’ 16#30	功能码
D2 上	‘6’ 16#36	
D2 下	‘1’ 16#31	数据地址
D3 上	‘0’ 16#30	
D3 下	‘0’ 16#30	
D4 上	‘0’ 16#30	
D4 下	‘0’ 16#30	数据内容
D5 上	‘0’ 16#30	
D5 下	‘1’ 16#31	
D6 上	‘2’ 16#32	
D6 下	‘D’ 16#44	LRC CHK
D7 上	‘7’ 16#37	LRC CHK
D7 下	CR 16#0D	END (后缀)
D8 上	LF 16#0A	

PLC 接收数据寄存器

寄存器	数据		说明
D100 上	‘:’	16#3A	STX (头码)
D100 下	‘0’	16#30	地址
D101 上	‘1’	16#31	
D101 下	‘0’	16#30	功能码
D102 上	‘6’	16#36	
D102 下	‘1’	16#31	数据地址
D103 上	‘0’	16#30	
D103 下	‘0’	16#30	
D104 上	‘0’	16#30	
D104 下	‘0’	16#30	数据内容
D105 上	‘0’	16#30	
D105 下	‘1’	16#31	
D106 上	‘2’	16#32	
D106 下	‘D’	16#44	LRC CHK
D107 上	‘7’	16#37	LRC CHK
D107 下	CR	16#0D	END (后缀)
D108 上	LF	16#0A	



### 9.1.2 ASCII (HEX->ASCII 码)

类型	助记符	P	HEX->ASCII 码																							
			In1(W)								In2(W)								Out1(W)							
字	ASCII	■																								
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□			■	■	■	■	■					

#### 参数说明:

- 1, In1: 数据来源起始装置;
- 2, In2: 变换的字符码个数, 取值范围为 1~256;
- 3, Out1: 存放 ASCII 码转换后的起始装置。

#### 指令执行注意点:

- 1, 通讯数据处理类指令的模式 SM75.0 为 Off 时, In1 取装置的 16 位从高到低转换, Out1 开始的存放数据装置的低位和高字节都将用来存储转换结果; SM75.0 为 On 时, In1 只取装置的低位字节转换, 高字节不转换, Out1 开始的装置只有低位字节用于存储转换结果, 高字节用 0 填充;

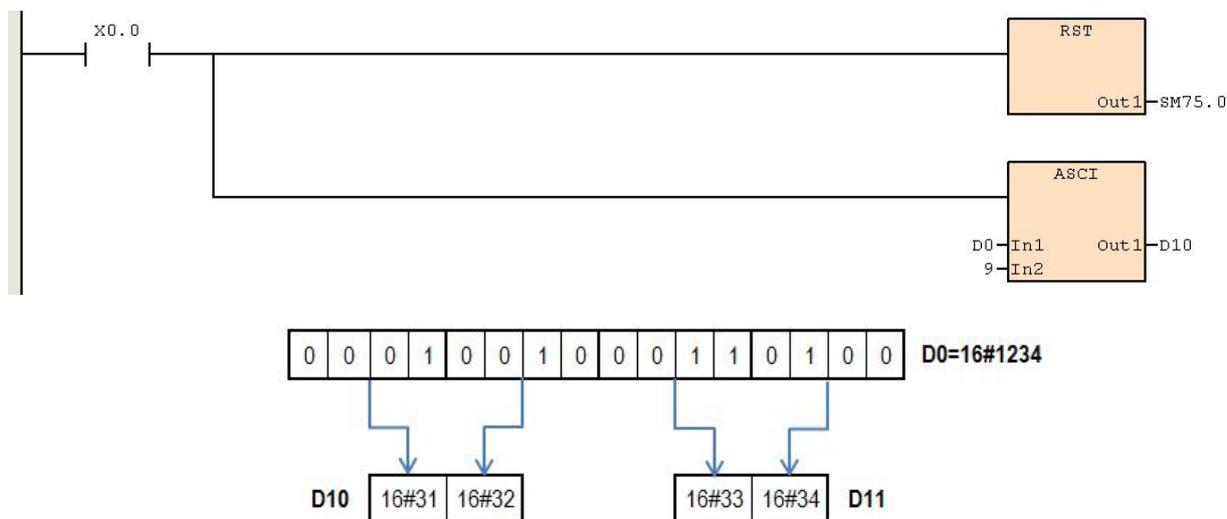
#### 指令功能说明:

- 1, 该指令是将 In1 开始的装置内的数据, 以 16 进制的格式, 依次从高位开始将 In2 个数 (大于 4 或 2 时, 依次取 In1+1 装置的数据转换) 转换成 ASCII 码放入 Out1 开始的连续的装置中。

#### 程序范例:

##### 例:

D0 = 16#1234, D1 = 16#5678, D2 = 16#9ABC, 且 SM75.0 为 Off, 当 X0.0 为 On 时, 指令执行, 将 D0 中的 16#1 转换成 ASCII 码 16#31, 放入 D10 的高 8 位中, D0 中的 16#2 转换成 ASCII 码 16#32, 放入 D10 的低 8 位中, 即 D10 = 16#3132, 同理, 则 D11 = 16#3334, D12 = 16#3536, D13 = 16#3738, D14 = 16#3900。



### 9.1.3 HEX (ASCII->HEX 码)

类型	助记符	P	ASCII->HEX 码																						
			In1(W)						In2(W)						Out1(W)										
字	HEX	■																							
双字																									
变量	位装置									字/双字装置															
	X	Y	S	M	T	C	H	L/	L/	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	
			M			C	C	E	L			M				C	C	I	Q	D	D	E	D	O	E
In1										■	■	■	■	■	□		■	■	■	■	■	■			
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Out1											■	■	■	■	□			■	■	■	■				

**参数说明:**

- 1, In1: 数据来源起始装置;
- 2, In2: 变换的个数, 取值范围为 1~256;
- 3, Out1: 存放 HEX 码转换后的起始装置。

**指令执行注意点:**

- 1, 通讯数据处理类指令的模式 SM75.0 为 Off 时, In1 取装置的高 8 位和低 8 位来转换, Out1 的 16 位从高位开始的高码存储转换结果; SM75.0 为 On 时, In1 取装置的低 8 位来转换, Out1 的从低 8 位的高码开始存储转换结果, 高 8 位用 0 填充;
- 2, 该指令只能转换 ASCII 数值在 16#30~39, 以及 16#41~46 之间, 有任何一个数值超出该范围, 则指令不执行。

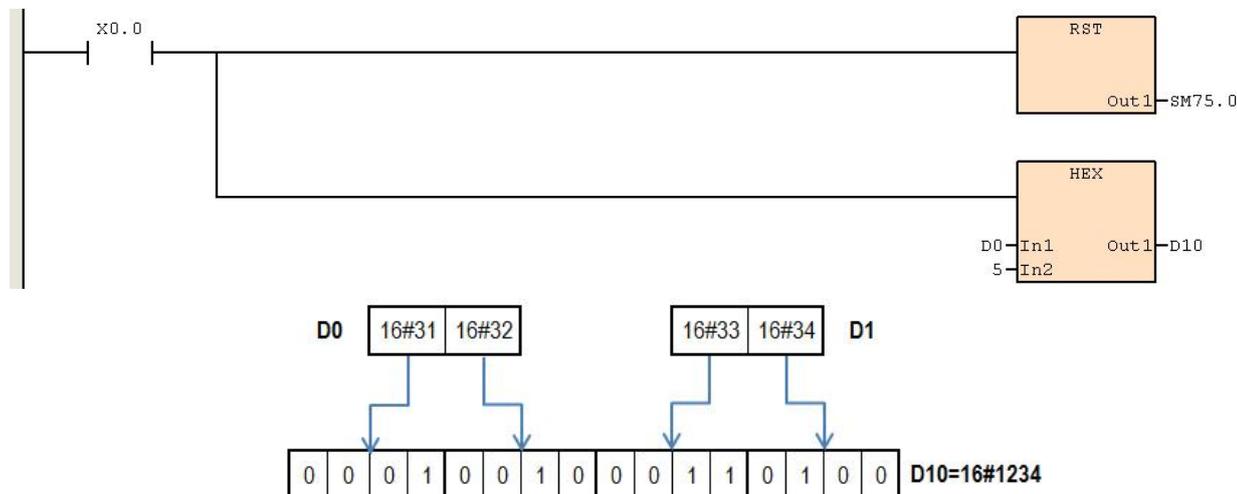
**指令功能说明:**

- 1, 该指令是将 In1 开始的装置内的数据, 以 16 进制的格式, 依次从高位开始 (每 2 个装置为一组) 将 In2 个数 (大于 2 时, 依次取 In1+1 装置的数据转换) 转换成 HEX 码放入 Out1 开始的连续的装置中。

**程序范例:**

**例:**

D0 = 16#3132, D1 = 16#3334, D2 = 16#3536, 且 SM75.0=Off, 当 X0.0 为 On 时, 指令执行, 将 D0 中的 16#3132 转换成 HEX 码 16#12, 放入 D10 的高 8 位中, D1 中的 16#3334 转换成 HEX 码 16#34, 放入 D10 的低 8 位中, 即 D10 = 16#1234, 同理, 则 D11 = 16#5000。



### 9.1.4 CCD (校验码)

类型	助记符	P	校验码																							
字	CCD	■	In1(W)									In2(W)									Out1(W)					
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□			■	■	■	■						

**参数说明:**

- 1, In1: 数据来源起始装置;
- 2, In2: 数据的个数, 取值范围为 1~256;
- 3, Out1: 存放校验检查结果的装置。

**指令执行注意点:**

- 1, 通讯数据处理类指令的模式 SM75.0= Off, 字转换模式, In2 数据个数以字节 (低字节和高字节都有效) 为单位作加总;  
 通讯数据处理类指令的模式 SM75.0= On, 字节转换模式, In2 数据个数以字节 (只有低字节有效) 为单位作加总。

**指令功能说明:**

- 1, 将 In1 所指定寄存器起始号码开始算的 In2 个数据内容作加总, 加总结果存放在 Out1 所指定的寄存器当中, 而将各个数据按位做异或运算, 存放在 Out1+1 的寄存器当中
- 2, 该指令用来作通信时, 为了确保数据传输时的正确性所做的字符串总和检查 (Sum Check) 。

程序范例:

例 1:

- 当 SM75.0= Off 时, 指定为 16 位转换模式。
- 当 X0.0=On 时, 将 D0 所指定寄存器起始号码开始算的 6 个数据 (以 8 位为单位 In2=6 代表指定 D0~D2) 内容作加总, 加总结果存放在 D100 所指定的寄存器当中, 将各个数据按位做异或运算, 结果存放在 D101 的寄存器当中。



In1	数据的内容	
D0 下	100 = 0 1 1 0 0 1 0 0	
D0 上	111 = 0 1 1 0 1 1 1 1	
D1 下	120 = 0 1 1 1 1 0 0 0	
D1 上	202 = 1 1 0 0 1 0 1 0	
D2 下	123 = 0 1 1 1 1 0 1 1	
D2 上	211 = 1 1 0 1 0 0 1 1	
D100	867	合计
D101	0 0 0 1 0 0 0 1	异或结果

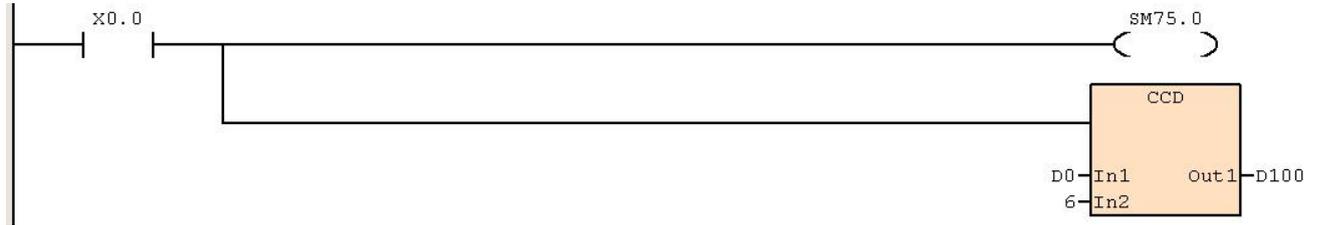
D100      0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1

D101      0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1



例 2:

- 当 SM75.0=On 时, 指定为 8 位转换模式。
- 当 X0.1=On 时, 将 D0 所指定寄存器起始号码开始算的 6 个数据 (以 8 位为单位 In2=6 代表指定 D0~D5) 内容作加总, 加总结果存放在 D100 所指定的寄存器当中, 将各个数据按位做异或运算, 结果存放在 D101 的寄存器当中。



In1	数据的内容
D0 下	100 = 0 1 1 0 0 1 0 0
D1 下	111 = 0 1 1 0 1 1 1 1
D2 下	120 = 0 1 1 1 1 0 0 0
D3 下	202 = 1 1 0 0 1 0 1 0
D4 下	123 = 0 1 1 1 1 0 1 1
D5 下	211 = 1 1 0 1 0 0 1 1
D100	867
D101	0 0 0 1 0 0 0 1

合计  
异或结果

D100    0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 1

D101    0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1

### 9.1.5 ABS (取绝对值运算)

类型	助记符	P	取绝对值运算																				
字	ABS	■	InOut1(W)																				
双字	DABS		InOut1(D)																				
变量	位装置									字/双字装置													
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L
InOut1										■	■	■	■	□	□		■	■	■	■			■

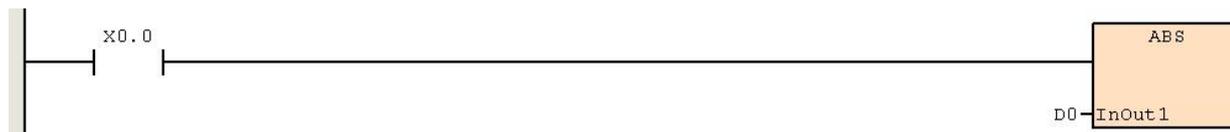
**指令说明:**

- 1, InOut1: 欲取绝对值的装置;
- 2, 当该指令执行时, 被指定的装置 InOut1 取绝对值。

**程序范例:**

例:

当 X0.0=Off→On 时, D0 内容取绝对值。



### 9.1.6 LRC (LRC 校验码计算)

类型	助记符	P	LRC 校验码计算																							
字	LRC	■	In1(W)								In2(W)								Out1(W)							
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
Out1										■	■	■	■	□			■	■	■	■	■			■		

**参数说明:**

- 1, In1: 待计算校验码运算的起始装置;
- 2, In2: 运算个数, 取值范围为 1~256;
- 3, Out1: 存放运算结果的装置。

**指令执行注意点:**

- 1, **通讯数据处理类指令的模式 SM75.0 = Off**, 将 In1 起始装置将其 16 进位数据区分为上 8 位和下 8 位, 将各个位数做 LRC 校验码运算, 传送到 Out1 的下 8 位中, 运算的个数以 In2 来设置;
- 2, **通讯数据处理类指令的模式 SM75.0 = On** 时, 将 In1 起始装置将其各个数据的低 8 位做 LRC 校验码运算, 高 8 位无效, 传送到 Out1 的下 8 位中, 运算的位数以 In2 来设置。

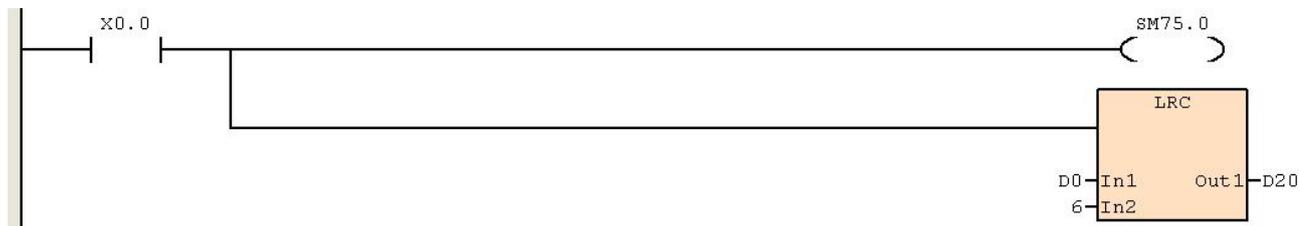
**指令功能说明:**

- 1, 该指令即将各运算数据以 8 位相加得 8 位的和后, 取 2 的补码 (即取反加 1), 该指令用于通讯格式为 ASCII 时的校验码计算;

程序范例:

例 1:

SM75.0= On 时, 启动 8 位模式, 当 X0.0=On 时, 将 D0~D5 的各低 8 位做 LRC 运算, 将结果传送到 D20 的下 8 位中。



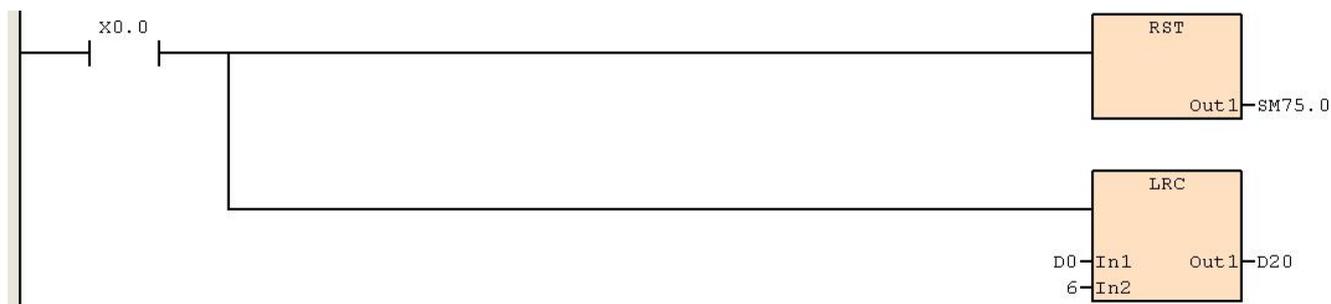
寄存器	数据	说明
D0 下	16#01	地址
D1 下	16#03	功能码
D2 下	16#21	起始数据地址
D3 下	16#01	
D4 下	16#00	数据个数
D5 下	16#06	
D20 下	16#D4	LRC

LRC 校验码:  $16\#01+16\#03+16\#21+16\#01+16\#00+16\#06=16\#2C$ , 然后取 2 的补码=16#D4。

此时, 将 16#D4 放入 D20 的下 8 位中。

例 2:

SM75.0= Off 时, 启动 16 位模式, 当 X0.0=On, 将 D0~D2 的各上、下 8 位做 LRC 运算, 将结果传送到 D20 的下 8 位。



寄存器	数据	说明
D0 上	16#01	地址
D0 下	16#03	功能码
D1 上	16#21	起始数据地址
D1 下	16#01	
D2 上	16#00	数据个数
D2 下	16#06	
D20 下	16#D4	LRC

LRC 校验码:  $16\#01+16\#03+16\#21+16\#01+16\#00+16\#06=16\#2C$ , 然后取 2 的补码=16#D4。

此时, 将 16#D4 放入 D20 的下 8 位中。



### 9.1.7 CRC (CRC 校验码计算)

类型	助记符	P	CRC 校验码计算																							
字	CRC	■	In1(W)								In2(W)								Out1(W)							
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■					
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■			
Out1										■	■	■	■	□		■	■	■	■	■						

#### 参数说明:

- 1, In1: 待计算校验码运算的起始装置;
- 2, In2: 运算个数, 取值范围为 1~256;
- 3, Out1: 存放运算结果的装置。

#### 指令执行注意点:

- 1, 通讯数据处理类指令的模式 SM75.0 = Off 时, 将 In1 起始装置将其 16 位数据区分为高 8 位和低 8 位, 将各个 8 位数据做 CRC 校验码运算, 传送到 Out1 的装置位中, 运算的个数以 In2 来设置;
- 2, 通讯数据处理类指令的模式 SM75.0 = On 时, 将 In1 起始装置将其 16 位数据区分为高 8 位 (无效数据) 和低 8 位, 将各低 8 位做 CRC 校验码运算, 传送到 Out1 的下 8 位和 Out1+1 的下 8 位中, 上 8 位都为 0, 运算的个数以 In2 来设置。

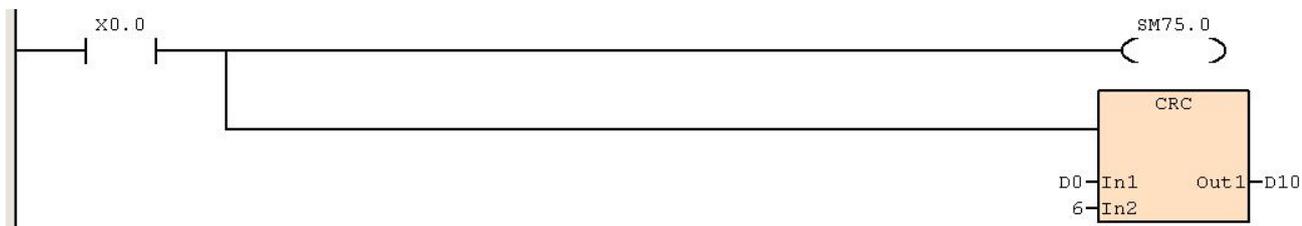
#### 指令功能说明:

- 1, 该指令用于通讯格式为 RTU 时的 CRC 校验码计算;
- 2, CRC 校验码运算规则如下:
  - ① 令 16 位寄存器 (CRC 寄存器) = 16#FFFF
  - ② 将第一个 8 位字节的数据与 CRC 寄存器的低 8 位做异或运算, 将结果存入 CRC 寄存器内;
  - ③ 右移一位 CRC 寄存器, 将 0 填入高位处;
  - ④ 检查右移的值, 如果是 0 则将步骤三的新值存入 CRC 寄存器内, 否则将 CRC 寄存器与数值 16#A001 做异或运算, 将结果存入 CRC 寄存器内;
  - ⑤ 重复步骤三~步骤四, 将 8 位数据全部运算完成;
  - ⑥ 重复步骤二~步骤五, 取下一个 8 位的信息指令, 直到所有信息指令运算完成。最后, 得到的 CRC 寄存器的值, 即是 CRC 的校验码。值得注意的是, CRC 的校验码必须高低字节交换放置于数据信息的校验码中。

程序范例：

例 1：

SM75.0=On, 启动 8 位模式, 当 X0.0=On, 将 D0~D5 的各低 8 位做 CRC 运算, 将结果传送到 D10、D11 的下 8 位。

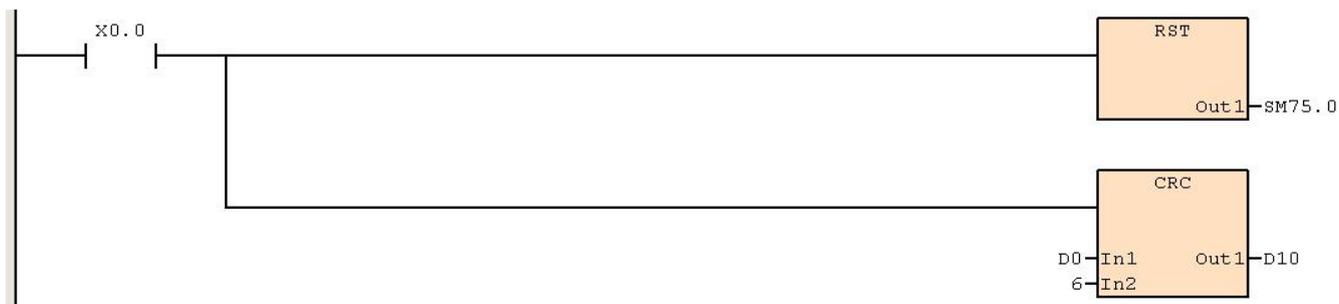


寄存器	数据	说明
D0 下	16#01	地址
D1 下	16#06	功能
D2 下	16#20	数据地址
D3 下	16#00	
D4 下	16#00	数据内容
D5 下	16#12	
D10 下	16#07	CRC
D11 下	16#02	CRC

CRC 校验码：此时，16#07 存于 D10 下 8 位内，16#02 存于 D11 下 8 位内。

例 2：

SM75.0=Off 时, 启动 16 位模式, 当 X0.0=On 时, 将 D0~D2 的各上、下 8 位做 CRC 运算, 将结果传送到 D10 的上、下 8 位中。



寄存器	数据	说明
D0 上	16#01	地址
D0 下	16#06	功能
D1 上	16#20	数据地址
D1 下	16#00	
D2 上	16#00	数据内容
D2 下	16#12	
D10 上	16#07	CRC
D10 下	16#02	CRC

CRC 校验码：此时，16#07 存于 D10 上 8 位内，16#02 存于 D10 下 8 位内。



## 9.2 MODBUS 通讯协议指令

### 9.2.1 MODRD (MODBUS 通讯寄存器数据读取)

类型	助记符	P	MODBUS 通讯寄存器数据读取																							
字	MODRD	■	In1(W)						In2(W)						In3(W)						In4(W)					
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In4									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		

#### 参数说明:

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站站地址, 操作数范围为 1~255;
- 3, In3: 欲读取数据的 MODBUS 地址;
- 4, In4: 读取寄存器的个数, 范围为 1~50。

#### 指令执行注意点:

- 1, 使用前必须先确保 **RS 指令开关** SM70.x 为 OFF;
- 2, 执行该指令, 将会读取指定地址的值放入指定的 SD300~SD349 或 SD350~SD399 中;
- 3, 通讯指令完全结束后, **通讯结束标志位** SM65.x, 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位** SM67.x 是否置起, 及读取错误码 SD245~SD246;
- 4, 如果通讯成功无异常, 数据存放 SD300~SD349 或 SD350~SD399 中, **通讯异常标志位** SM67.x 为 OFF; 若有异常, **通讯异常标志位** SM67.x 置起, 同时错误码放于对应 SD245~SD246 中, 存放的数据无效;
- 5, 无论使用 ASCII 协议或是 RTU 协议, 使用者所填写和看到的 SD300~SD349 或 SD350~SD399 中的数据, 均为 RTU 格式数据, 发送和接收时 ASCII 格式被 PLC 主机自动转换; 接收数据时, 先接收 SD300 或 SD350, 依次接收到 SD349 或 SD399; 各 SD 中, 先接收高 8 位的字节, 后接收低 8 位的字节;
- 6, 同一时间, 一个通讯口只有一个通讯请求会被处理, 请注意判断**通讯结束标志位** SM65.x。

#### 指令功能说明:

- 1, 该指令遵从 MODBUS 协议的寄存器数据读取通讯规则, 可利用该指令方便快捷的对通讯从站的数据进行读取。

程序范例:

例:

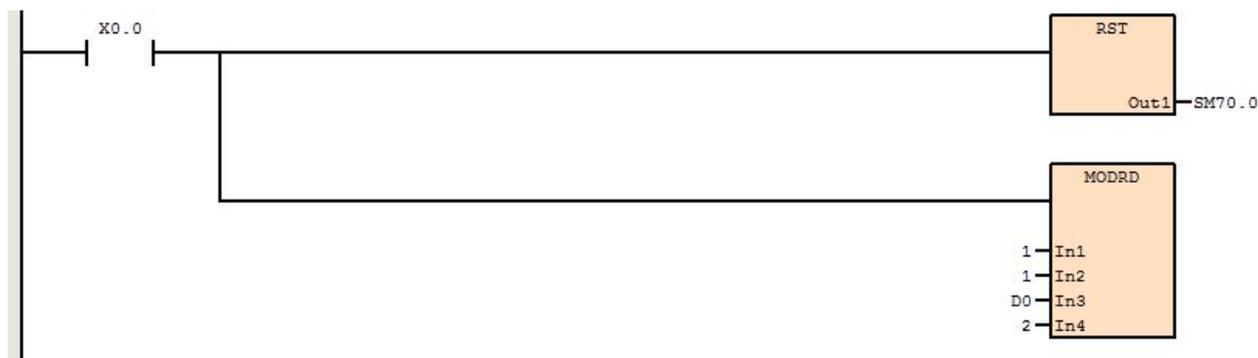
设置 PLC 主机通讯口为 1，读取数据的 MODBUS 地址 D0=16#2102，读取寄存器的个数设置为 2 个。

设置 PLC 通讯模式为 RTU，波特率为 38400，数据协议为 8,E,1，超时时间为 200ms，具体设置如下图所示:



通讯的设置也可以通过 SD 来查看，具体参见 SD 的说明。

设置好后,将 X0.0 置 On,MODRD 指令执行,将会读取 MODBUS 地址 16#2102、16#2103 的值放入指定的 SD300~SD349。



PLC 接收数据寄存器

寄存器	数据	说明
SD300 上	16#17	地址 16#2102 的内容
SD300 下	16#70	
SD301 上	16#00	地址 16#2103 的内容
SD301 下	16#00	



## 9.2.2 MODWR (MODBUS 通讯单个寄存器数据写入)

类型	助记符	P	MODBUS 通讯单个寄存器数据写入																							
字	MODWR	■	In1(W)									In2(W)					In3(W)					In4(W)				
双字																										
变量	位装置									字/双字装置																
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L			
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In3									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		
In4									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■		

### 参数说明:

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站地址, 操作数范围为 0~255, 其它地址无效不执行, 0 表示对所有站地址进行广播控制, 发送结束后会自动延时 200ms 后, 才可继续发送下一笔数据, 在延时过程中, **广播延时标志位 SM72.0~SM72.1** 会置 On;
- 3, In3: 欲写入数据的 MODBUS 地址;
- 4, In4: 欲写入的数据。

### 指令执行注意点:

- 1, 使用前必须先确保 **RS 指令开关 SM70.x** 为 OFF;
- 2, 执行该指令, 将会把欲写入的数据传送到指定的从站的地址中;
- 3, 通讯指令完全结束后, **通讯结束标志位 SM65.x**, 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位 SM67.x** 是否置起, 及读取错误码 SD245~SD246;
- 4, 如果通讯成功无异常, **通讯异常标志位 SM67.x** 为 OFF; 若有异常, 则**通讯异常标志位 SM67.x** 置起, 同时错误码放于对应 SD245~SD246 中;
- 5, 若使用地址 0, 即广播发送, 在 200ms 后, **通讯结束标志位 SM65.x** 才会置 ON 一个扫描周期, 并不去处理任何回送数据;
- 6, 同一时间, 一个通讯口只有一个通讯请求会被处理, 请注意判断**通讯结束标志位 SM65.x**。

### 指令功能说明:

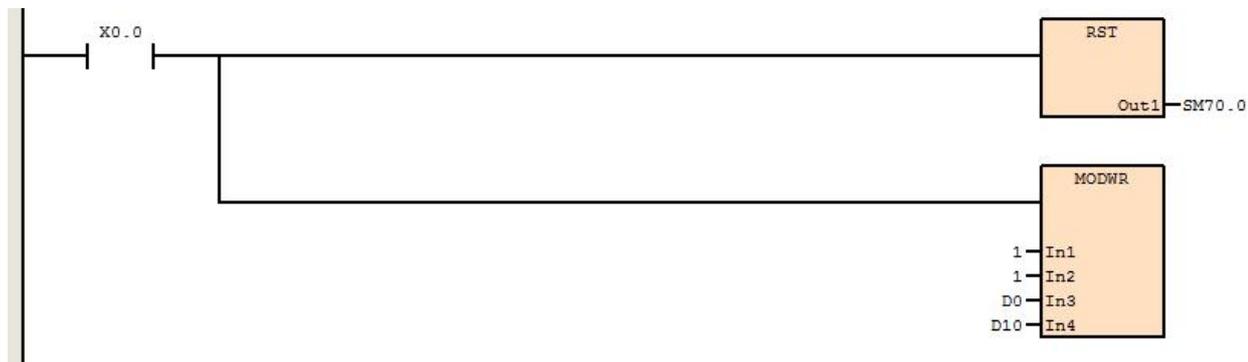
- 1, 指令遵从 MODBUS 协议的单个寄存器数据写入通讯规则, 可利用该指令方便快捷的对通讯从站的指定地址进行数据写入。

程序范例：

例：

设置 PLC 主机通讯口为 1，写入数据的 MODBUS 地址 D0=16#2000，欲写入的资料 D10=16#12。

设置好后，将 X0.0 置 On，MODWR 指令执行，将会把写入数据 16#12 通过通讯传送到指定的从站的地址 16#2000 中。





### 9.2.3 MODRW (MODBUS 通讯数据读写)

类型	助记符	P	MODBUS 数据读写																								
字	MODRW	■	In1(W)					In2(W)					In3(W)					In4(W)					In5(W)				
双字																											
变量	位装置										字/双字装置																
	X	Y	S	M	T	C	H	L/	E/	L	X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L/	E/	L
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	■
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	■
In3										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	■
In4										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	■
In5										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	■	■

#### 参数说明:

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站地址, 范围为 0~255, 0 表示对所有站地址进行广播控制, 发送结束后会自动延时 200ms 后, 才可继续发送下一笔数据, 在延时过程中, **广播延时标志位** SM72.0~SM72.1 会置 On;
- 3, In3: 通讯功能码, 只取低 8 位;
- 4, In4: 欲读写数据的 MODBUS 地址;
- 5, In5: 读写数据个数, 数据个数为 1~50 个。

#### 指令执行注意点:

- 1, 使用前必须先确保 **RS 指令开关** SM70.x 为 OFF;
- 2, 执行该指令, 将会读取指定地址的值放入指定的 SD300~SD349 或 SD350~SD399 中或将该区域中的数值写入指定地址,
- 3, 通讯指令完全结束后, **通讯结束标志位** SM65.x, 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位** SM67.x 是否置起, 及读取错误码 SD245~SD246;
- 4, 如果通讯成功无异常, 则数据存放 SD300~SD349 或 SD350~SD399 中, **通讯异常标志位** SM67.x 为 OFF; 若有异常, 则**通讯异常标志位** SM67.x 置起, 错误码放于对应 SD245~SD246 中, 数据存放的 SD300~SD349 或 SD350~SD399 中数据无效;
- 5, 若使用地址 0, 即广播发送, 在 200ms 后, **通讯结束标志位** SM65.x 才会置 ON 一个扫描周期, 并不去处理任何回送数据;
- 6, 无论使用 ASCII 协议或是 RTU 协议, 使用者所填写和看到的 SD300~SD349 或 SD350~SD399 中的数据, 均为 RTU 格式数据, 发送和接收时 ASCII 格式被 PLC 主机自动转换; 发送/接收数据时, 先发送/接收 SD300 或 SD350, 依次发送/接收到 SD349 或 SD399; 各 SD 中, 先发送/接收高 8 位的字节, 后发送/接收低 8 位的字节;

#### 指令功能说明:

- 1, 该指令遵从 MODBUS 协议的寄存器数据读写通讯规则, 可利用该指令方便快捷的与通讯从站的数据进行读写, 支持的通讯功能码包括:

16#01	读取多笔位 (bit) 装置
16#05	单笔位 (bit) 装置状态写入
16#0F	多笔位 (bit) 装置状态写入
16#10	多笔字符 (word) 装置数据写入

程序范例：

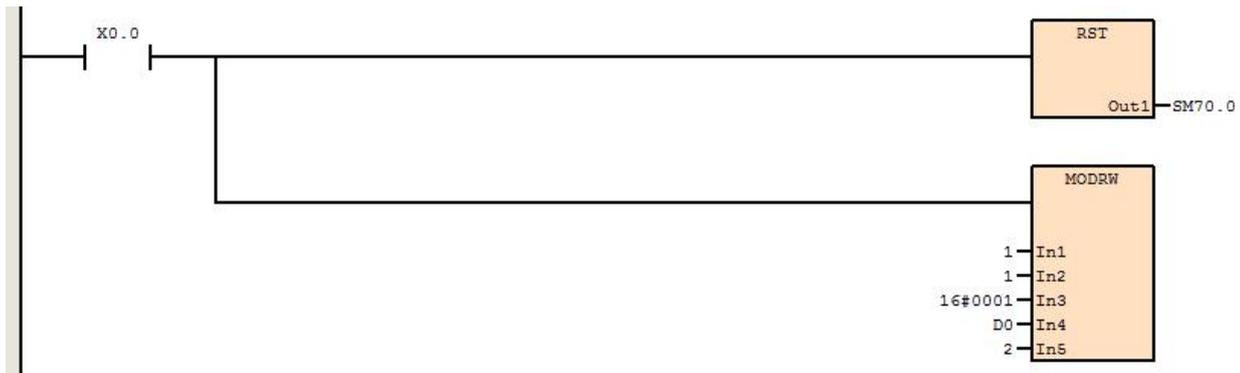
例 1: 16#10 多笔字符 (word) 装置数据写入

- 设置 PLC 主机通讯口为 1, 写入数据的 MODBUS 地址 D0=16#2000, 欲写入的资料 SD300=16#12, SD301=16#1770。
- 设置好后, 将 X0.0 置 On, MODRW 指令执行, 将会把写入数据 16#12、16#1770 通过通讯传送到指定的从站的地址 16#2000、16#2001 中。



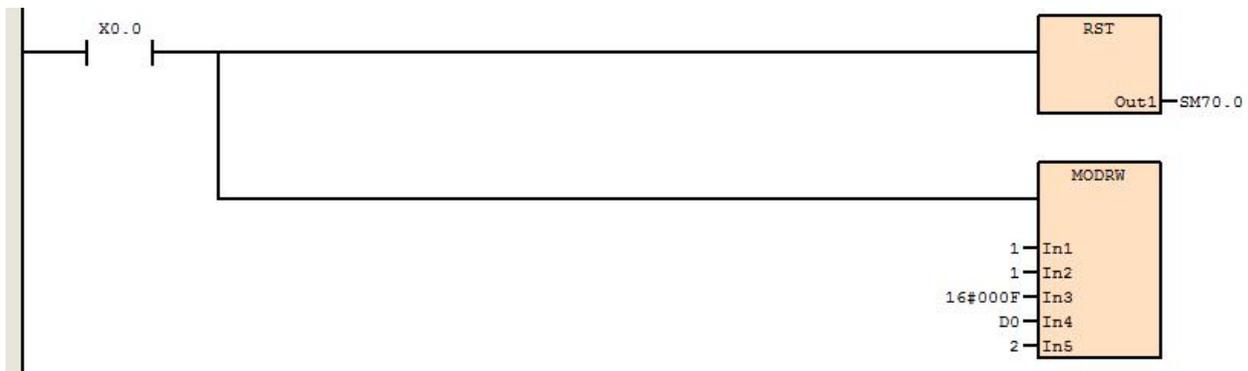
例 2: 16#01 读取多笔位 (bit) 装置

- 设置 PLC 主机通讯口为 1, 读取数据的 MODBUS 地址 D0=16#0500, 读取位装置的个数设置为 6 个。
- 设置好后, 将 X0.0 置 On, MODRW 指令执行, 将会读取 MODBUS 地址 16#0500, 16#0501, 16#0502, 16#0503, 16#0504, 16#0505 位状态值放入指定的 SD300 中。



例 3: 16#0F 多笔位 (bit) 装置状态写入

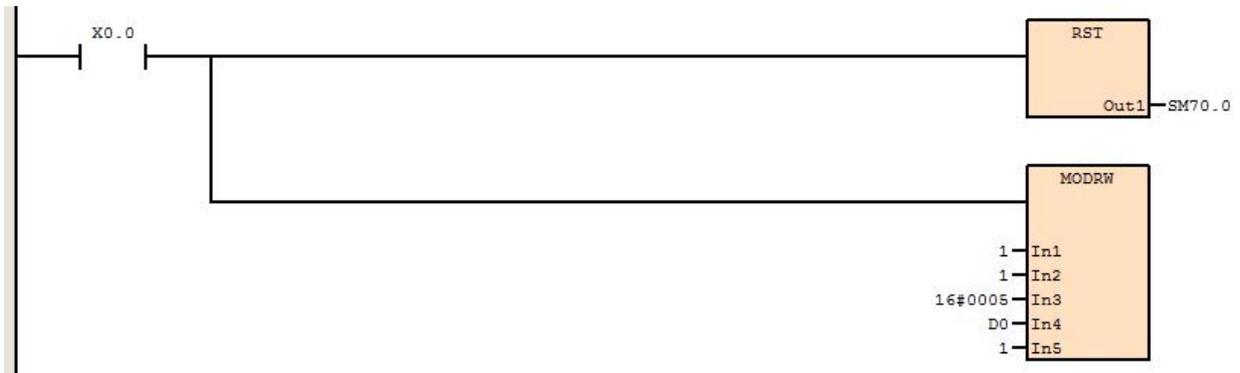
- 设置 PLC 主机通讯口为 1, 写入数据的 MODBUS 地址 D0=16#0500, 欲写入的资料 SD300=16#07。
- 设置好后, 将 X0.0 置 On, MODRW 指令执行, 将会把写入数据 16#07 通过通讯传送到指定的 6 个位地址, 其数据值通过位状态表示, 即 16#0500=1, 16#0501=1, 16#0502=1, 16#0503=0, 16#0504=0, 16#0505=0。





例 4: 16#05 单笔位 (bit) 装置状态写入

设置 PLC 主机通讯口为 1, 写入数据的 MODBUS 地址 D0=16#0500, 欲写入的位装置状态为 On (SD300 不等于 0 时设置为 On, SD300 等于 0 时设置为 Off)。



### 9.3 JSCC 变频器控制指令

#### 9.3.1 FWD（变频器正转指令）

类型	助记符	P	变频器正转指令																					
字	FWD	■	In1(W)										In2(W)											
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明：**

- 1, In1: PLC 主机的通讯口，值必须为 1 或者 2；
- 2, In2: 通讯从站站地址，范围为 0~255，0 表示对所有站地址进行广播控制，发送结束后会自动延时 200ms 后，才可继续发送下一笔数据，在延时过程中，**广播延时标志位 SM72.0~SM72.1** 会置 On。

**指令执行注意点：**

- 1, 使用前必须先确保 **RS 指令开关 SM70.x** 为 OFF；
- 2, 执行该指令，将会把正转传送到指定的从站变频器中；
- 3, 通讯指令完全结束后，**通讯结束标志位 SM65.x**，会置 ON 一个扫描周期，建议使用下降沿来判断该标志位，同时注意判断**通讯异常标志位 SM67.x** 是否置起，及读取错误码 SD245~SD246；
- 4, 当所有过程结束后，如果通讯成功无异常，则通讯异常标志位 SM67.x 为 OFF；若有异常，则**通讯异常标志位 SM67.x** 置起，同时错误码放于对应 SD245~SD246 中；
- 5, 若使用地址 0，即广播发送，在 200ms 后，**通讯结束标志位 SM65.x** 才会置 ON 一个扫描周期，并不去处理任何回送数据；
- 6, 同一时间，一个通讯口只有一个通讯请求会被处理，请注意判断**通讯结束标志位 SM65.x**。

**指令功能说明：**

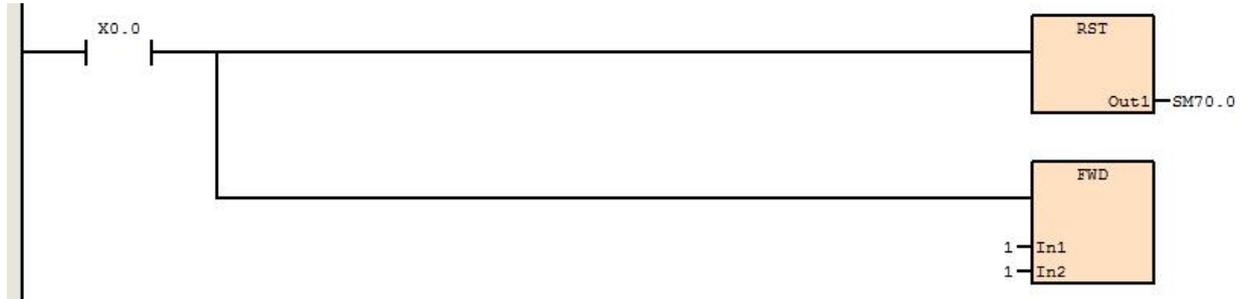
- 1, 该指令遵从 MODBUS 协议的通讯规则，可利用该指令方便快捷的对变频器进行正转指令控制。



**程序范例:**

**例:**

- 设置 PLC 主机通讯口为 1。
- 在指令执行前必须确保启用 RS 指令开关 SM70.x 为 OFF 状态，否则指令不执行。设置好后，将 X0.0 置 On，FWD 指令执行，将会把通讯数据传送到指定的从站的地址中。



### 9.3.2 REV (变频器反转指令)

类型	助记符	P	变频器反转指令																					
字	REV	■	In1(W)										In2(W)											
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站站地址, 范围为 0~255, 其它地址无效不执行, 0 表示对所有站地址进行广播控制, 发送结束后会自动延时 200ms 后, 才可继续发送下一笔数据, 在延时过程中, **广播延时标志位 SM72.0~SM72.1** 会置 On。

**指令执行注意点:**

- 1, 使用前必须先确保 **RS 指令启用开关 SM70.x** 为 OFF;
- 2, 执行该指令, 将会把反转传送到指定的从站变频器中;
- 3, 通讯结束后, **通讯结束标志位 SM65.x** 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位 SM67.x** 是否置起, 及读取错误码 SD245~SD246;
- 4, 当所有过程结束后, 如果通讯成功无异常, 则通讯异常标志位 SM67.x 为 OFF, 若有异常, 则**通讯异常标志位 SM67.x** 置起, 同时错误码放于对应 SD245~SD246 中;
- 5, 若使用地址 0, 即广播发送, 在 200ms 后, **通讯结束标志位 SM65.x** 才将置 ON 一个扫描周期, 并不去处理任何回送数据;
- 6, 同一时间, 一个通讯口只有一个通讯请求会被处理, 请注意判断**通讯结束标志位 SM65.x**。

**指令功能说明:**

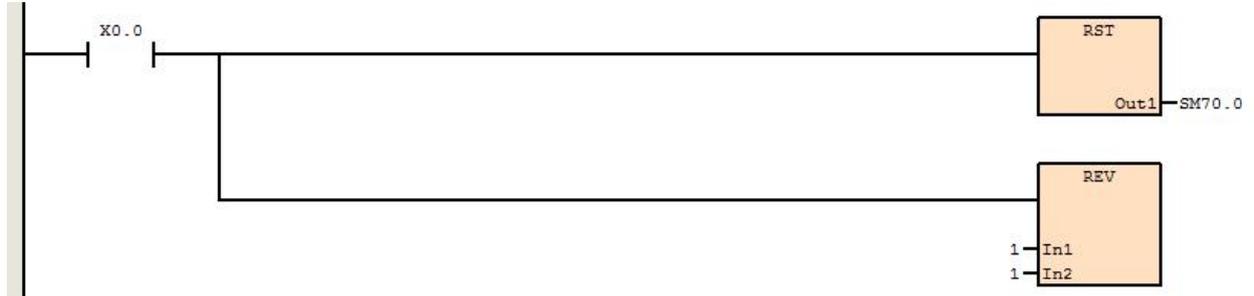
- 1, 该指令遵从 MODBUS 协议的通讯规则, 可利用该指令方便快捷的对变频器进行反转指令控制。



**程序范例:**

**例:**

- 设置 PLC 主机通讯口为 1。
- 在指令执行前必须确保 RS 指令启用开关 SM70.x 为 OFF 状态，否则指令不执行。设置好后，将 X0.0 置 On，REV 指令执行，将会把通讯数据传送到指定的从站的地址中。



### 9.3.3 STOP (变频器停止运转指令)

类型	助记符	P	变频器停止运转指令																					
字	STOP	■	In1(W)										In2(W)											
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站站地址, 范围为 0~255, 0 表示对所有站地址进行广播控制, 发送结束后会自动延时 200ms 后, 才可继续发送下一笔数据, 在延时过程中, **广播延时标志位** SM72.0~SM72.1 会置 On。

**指令执行注意点:**

- 1, 使用前必须先确保 **RS 启用指令开关** SM70.x 为 OFF;
- 2, 执行该指令, 将会把停止命令送到指定的从站变频器中;
- 3, 通讯结束后, **通讯结束标志位** SM65.x 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位** SM67.x 是否置起, 及读取错误码 SD245~SD246;
- 4, 当所有过程结束后, 如果通讯成功无异常, 则通讯异常标志位 SM67.x 为 OFF, 若有异常, 则**通讯异常标志位** SM67.x 置起, 同时错误码放于对应 SD245~SD246 中;
- 5, 若使用地址 0, 即广播发送, 在 200ms 后, **通讯结束标志位** SM65.x 才将置 ON 一个扫描周期, 并不去处理任何回送数据;
- 6, 同一时间, 一个通讯口只有一个通讯请求会被处理, 请注意判断**通讯结束标志位** SM65.x。

**指令功能说明:**

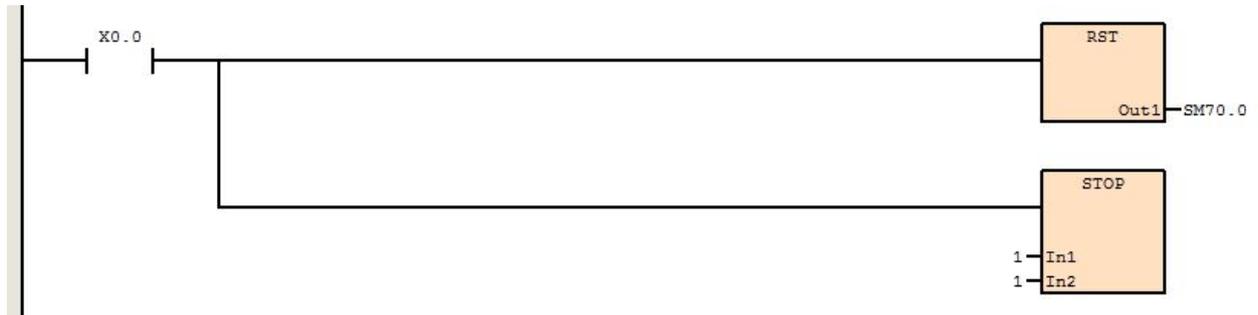
- 1, 该指令遵从 MODBUS 协议的通讯规则, 可利用该指令方便快捷的对变频器进行停止运转指令控制。



**程序范例:**

**例:**

- 设置 PLC 主机通讯口为 1。
- 在指令执行前必须确保 RS 指令启用开关 SM70.x 为 OFF 状态，否则指令不执行。设置好后，将 X0.0 置 On，STOP 指令执行，将会把通讯数据传送到指定的从站的地址中。



### 9.3.4 FREQ (变频器设置频率指令)

类型	助记符	P	变频器设置频率指令																						
字	FREQ	■	In1(W)							In2(W)							In3(W)								
双字																									
变量	位装置									字/双字装置															
	X	Y	S M	M	T	C	H C	L/ E L		X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In3										■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站站地址, 范围为 0~255, 0 表示对所有站地址进行广播控制, 发送结束后会自动延时 200ms 后, 才可继续发送下一笔数据, 在延时过程中, **广播延时标志位** SM72.0~SM72.1 会置 On;
- 3, In3: 欲设置的运转频率, 单位是 0.1HZ, 即如果 In3 等于 1000, 即频率设置为 100.0HZ。

**指令执行注意点:**

- 1, 使用前必须先确保 **RS 启用指令开关** SM70.x 为 OFF;
- 2, 执行该指令, 将会把设置频率到指定的从站变频器中;
- 3, 通讯结束后, **通讯结束标志位** SM65.x 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位** SM67.x 是否置起, 及读取错误码 SD245~SD246;
- 4, 当所有过程结束后, 如果通讯成功无异常, 则通讯异常标志位 SM67.x 为 OFF, 若有异常, 则**通讯异常标志位** SM67.x 置起, 同时错误码放于对应 SD245~SD246 中;
- 5, 若使用地址 0, 即广播发送, 在 200ms 后, **通讯结束标志位** SM65.x 才将置 ON 一个扫描周期, 并不去处理任何回送数据;
- 6, 同一时间, 一个通讯口只有一个通讯请求会被处理, 请注意判断**通讯结束标志位** SM65.x。

**指令功能说明:**

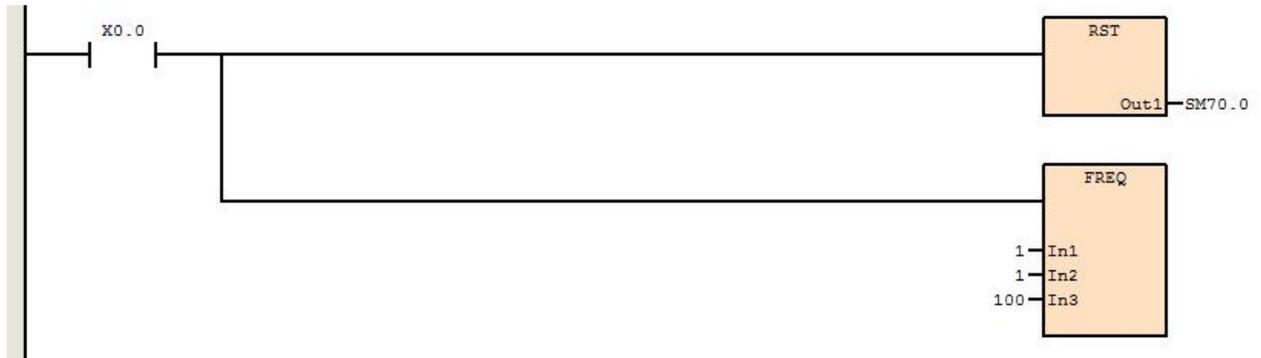
- 1, 该指令遵从 MODBUS 协议的单个寄存器数据写入通讯规则, 可利用该指令方便快捷的对变频器进行运转频率的设置。



**程序范例:**

**例:**

- 设置 PLC 主机通讯口为 1。
- 在指令执行前必须确保 RS 指令启用开关 SM70.x 为 OFF 状态，否则指令不执行。设置好后，将 X0.0 置 On，FREQ 指令执行，将会把通讯数据传送到指定的从站的地址中。



### 9.3.5 RERR (变频器读取故障指令)

类型	助记符	P	变频器读取故障指令																					
字	RERR	■	In1(W)										In2(W)											
双字																								
变量	位装置									字/双字装置														
	X	Y	S M	M	T	C	H C	L/ E L	X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L/ E L	
In1									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
In2									■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, In1: PLC 主机的通讯口, 值必须为 1 或者 2;
- 2, In2: 通讯从站站地址, 范围为 1~255。

**指令执行注意点:**

- 1, 使用前必须先确保 **RS 启用指令开关 SM70.x** 为 OFF;
- 2, 执行该指令, 将会读取变频器故障的值放入 SD300 或 SD350 中;
- 3, 无论使用 ASCII 协议或是 RTU 协议, 使用者所填写和看到的 SD300 或 SD350 中的数据, 均为 RTU 格式数据, 发送和接收时 ASCII 格式被 PLC 主机自动转换; 接收数据时, 先接收高 8 位的字节, 后接收低 8 位的字节;
- 4, 通讯结束后, **通讯结束标志位 SM65.x** 会置 ON 一个扫描周期, 建议使用下降沿来判断该标志位, 同时注意判断**通讯异常标志位 SM67.x** 是否置起, 及读取错误码 SD245~SD246;
- 5, 当所有过程结束后, 如果通讯成功无异常, 则通讯异常标志位 SM67.x 为 OFF, 若有异常, 则**通讯异常标志位 SM67.x** 置起, 同时错误码放于对应 SD245~SD246 中;
- 6, 若使用地址 0, 即广播发送, 在 200ms 后, **通讯结束标志位 SM65.x** 才将置 ON 一个扫描周期, 并不去处理任何回送数据;
- 7, 同一时间, 一个通讯口只有一个通讯请求会被处理, 请注意判断**通讯结束标志位 SM65.x**。

**指令功能说明:**

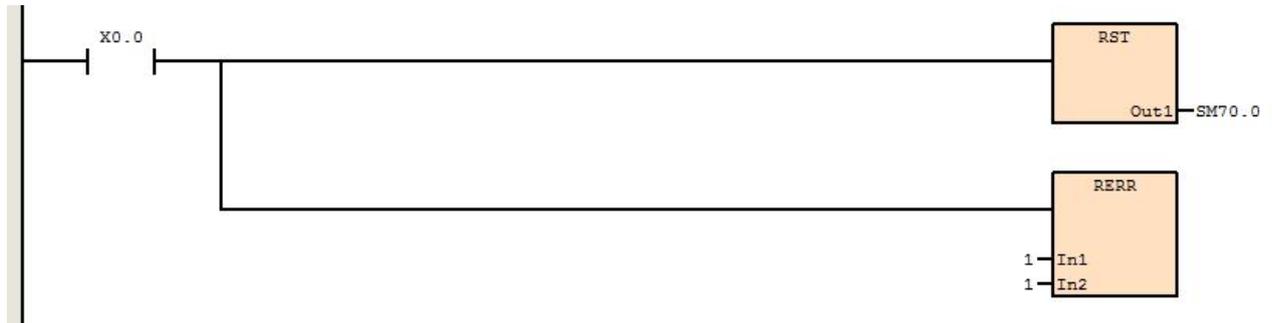
- 1, 该指令遵从 MODBUS 协议的寄存器数据读取通讯规则, 可利用该指令方便快捷的对变频器进行读取故障指令控制。



**程序范例:**

**例:**

- 设置 PLC 主机通讯口为 1。
- 在指令执行前必须确保 RS 指令启用开关 SM70.0 为 OFF 状态，否则指令不执行。设置好后，将 X0.0 置 On，RERR 指令执行，将会读取指定地址的值放入指定的 SD300 中。



## 十 CC-CAN 总线指令

### 10.1 使用总线及节点状态 SM 标志和 SD 寄存器

#### 1, 总线通讯忙碌状态及错误

(1) SM117.1~SM120.7: 各节点正在忙碌标志位, 通讯相关指令使用前须确认对应站号的忙碌标志位为 OFF;

(2) SM116.0: CC-CAN 总线错误标志位, 用于标志能够引起系统错误停机的错误标志, 原因有:

- 1) 总线上不存在节点;
- 2) 总线干扰太大, 导致所有节点无法通讯;
- 3) 有节点出现掉线或设备不一致, 而节点的错误处理为报错;

(3) SM116.1: CC-CAN 总线警告标志位, 用于标志能够引起系统警告的警告标志, 原因有:

有节点出现掉线或设备不一致, 而节点的错误处理为警告;

(4) SD800: CC-CAN 总线错误码, 用于记录最近一次引起总线报错的原因:

SD 值	原因
0	总线无错误
1~31	节点 1~31 出现掉线或设备不一致或设备报错
40	总线通讯无法进行 (不存在节点或干扰太大)

(5) SD801~SD831: CC-CAN 总线节点错误码, 用于记录各节点的错误原因:

SD 值	原因
0	总线无错误
1	掉线
2	设备类型不一致
3	寄存器读取失败
4	写入地址超出范围
5	写入数值超出范围
6	写入寄存器为只读类型
7	当前设备状态, 无法写入该寄存器

#### 2, 电子手轮相关功能

(1) SM116.3: 电子手轮正在忙碌标志位, 用于指示电子手轮是否正在读取位移信息;

(2) SM116.4: 电子手轮更新信息标志位, 当电子手轮完成一次数据更新, 该标志位将会置 ON 一个扫描周期;

(3) SD901~SD907: CC-CAN 电子手轮 1~7 轴对应伺服站号 M 设定; 默认是 1 轴对应站号 1 的伺服, 第 N 轴对应站号 N 的伺服; 用于“伺服跟随电子手轮位移”指令, 可提前设定好轴号对应伺服站号, 自动输出。

#### 3, 伺服通用功能

(1) SM121.1~SM124.6: 各节点伺服正在运动控制标志位;

(2) SM108.1~SM111.6: 各节点伺服停止方式选择, 用于选择当 PLC 停机时或使用“停止指令”时, 正常运转伺服的停止方式:

OFF: 按照设定的加减速时间进行减速停止;

ON: 不减速, 直接马上停止。

(3) SM116.6: CC-CAN 直线插补指令的设定目标频率方式选择;

OFF: 设定的目标频率为转速最大轴 (即脉冲个数更多的轴), 另一轴自动设定目标频率;

ON: 设定的目标频率为 X 轴的频率, Y 轴根据插补算法自动设定目标频率。

(4) SD832~SD891: 站号 N 伺服当前的程序绝对位置, 绝对位置型指令依据此 SD 值进行坐标定位, 使用时需要在回原点后, 将 SD 绝对位置设置为 0, 之后使用绝对位置型指令运动; 其中每个站号, 占用两个 SD, 例如站号 1 的伺服, SD832~SD833 组成 32 位的站号 1 的当前绝对位置, 后边站号的伺服以此类推。

#### 4, 运动控制机械位置相关 (以下相关的设置变更后, 必须复位一次 PLC)

##### (1) 机械实际绝对位置

SD602~SD661: 站号 N 伺服当前的机械实际绝对位置 (掉电保持), 双字装置;

SM112.1~SM115.6: 站号 N 伺服的机械实际位置清 0 标志位; 应用于更换 PLC 时, 伺服原点还在, 而 PLC 需要重新校对机械实际位置原点; 正常使用; 置 ON 后会将 SD602~SD661 清 0, 同时自动复位为 OFF;

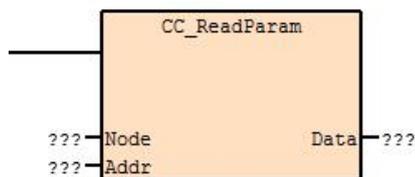
**说明:** 当使用 SVO\_SetHome 指令或使用软件中的原点设置功能时, SD602~SD661 会被设置为 0, 同时会发送设置原点命令给伺服, 会清空原先伺服的正反极限设置 (若希望 SD602~SD661 清 0, 而不设置伺服, 请使用 SM112.1~SM115.6 清 0); 具有掉电保持功能 (亦不用担心掉电后, 适量手动转动电机问题), 可用于原点回归功能以及机械当前实际单元位置的显示和参数设置, 也可使用于转矩模式下的位置判断;

①数值根据上位机的设定电机单元数值, 换算成实际单元位置;

②若上位机设置伺服做为旋转轴 (周期型), 系统根据传动比 (电机端及机械端周期圈数) 设置, 机械每转过一圈, 会自动把 SD602~SD661 的位置数值重置在机械终端周期单元数值范围内, 可用于显示真实的角度位置。

## 10.2 通用控制指令

### 10.2.1 CC\_ReadParam (读取单个参数)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	EL
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Addr	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Data	Word	Output		■	■	■	■	□			■	■	■	■				

#### 参数说明:

- 1, Node: 目标节点站号, 范围 1~31;
- 2, Addr: 目标节点的数据地址;
- 3, Data: 读取回复的目标节点目标地址的数据;

#### 指令执行注意点:

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

#### 指令功能说明:

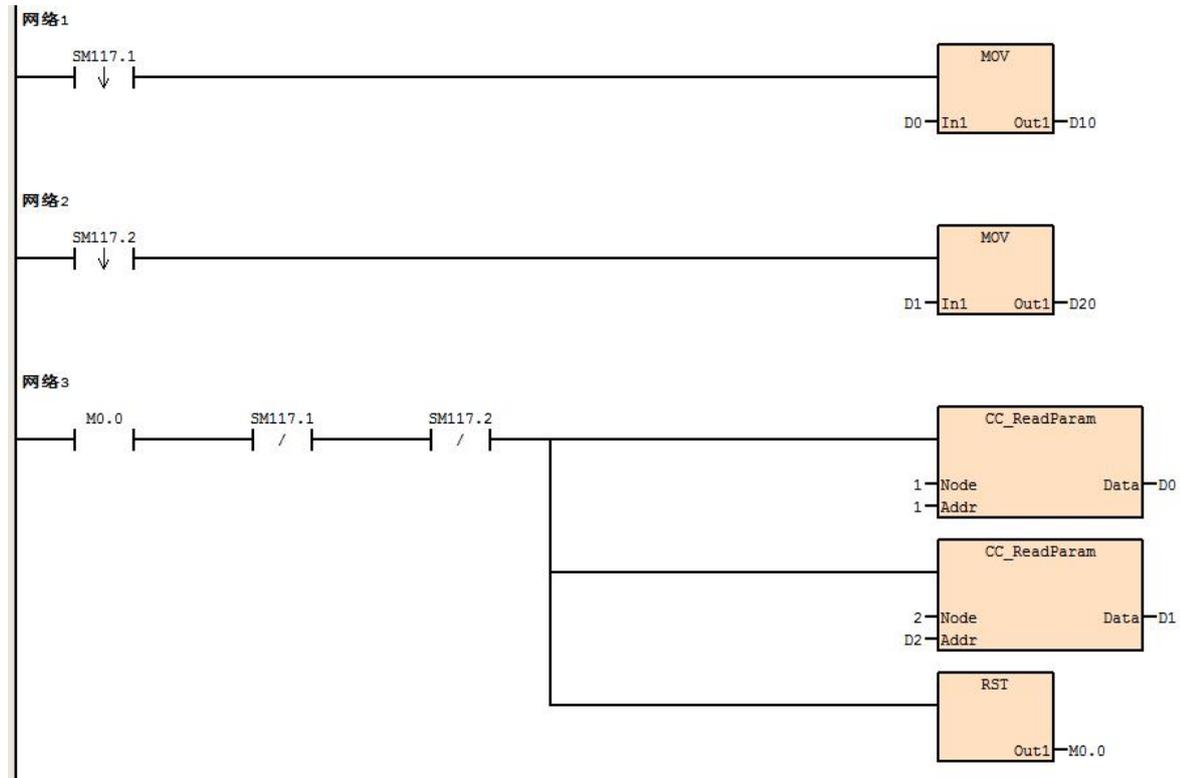
- 1, 该指令的功能是, 从目标节点站号 Node 的读取地址为 Addr 的数据, 放入 Data 装置中。

#### 程序范例:

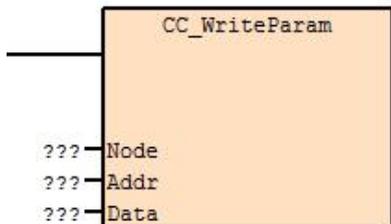
(见下页)

例:

- 当站号 1 和站号 2 的从站都空闲时, 即 SM117.1 和 SM117.2 均为 OFF, (SM117.1 对应站号 1 的忙碌标志, SM117.2 对应站号 2 的忙碌标志), 置位 M0.0 为 ON, 即可去读取站号 1 地址为 1 的数据放入 D0 装置中, 同时去读取站号 2 的地址为装置 D2 的数据中的数据放入 D1 装置中, 读取过程中站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 置 ON;
- 当站号 1 完成数据读取时, 即 SM117.1 忙碌标志从 ON->OFF 时, 将读取回来的数据从 D0 搬移到 D10 装置中; 同样, 当站号 2 完成数据读取时, 即 SM117.2 忙碌标志从 ON->OFF, 将读取回来的数据从 D1 搬移到 D20 装置中;
- 完成上述动作后, 再将 M0.0 置为 ON, 即可重复上一次的读取动作。



### 10.2.2 CC\_WriteParam (写入单个菜单参数)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Addr	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Data	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标节点站号, 范围 1~31;
- 2, Addr: 目标节点的数据地址;
- 3, Data: 想要写入目标节点目标地址的数据;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

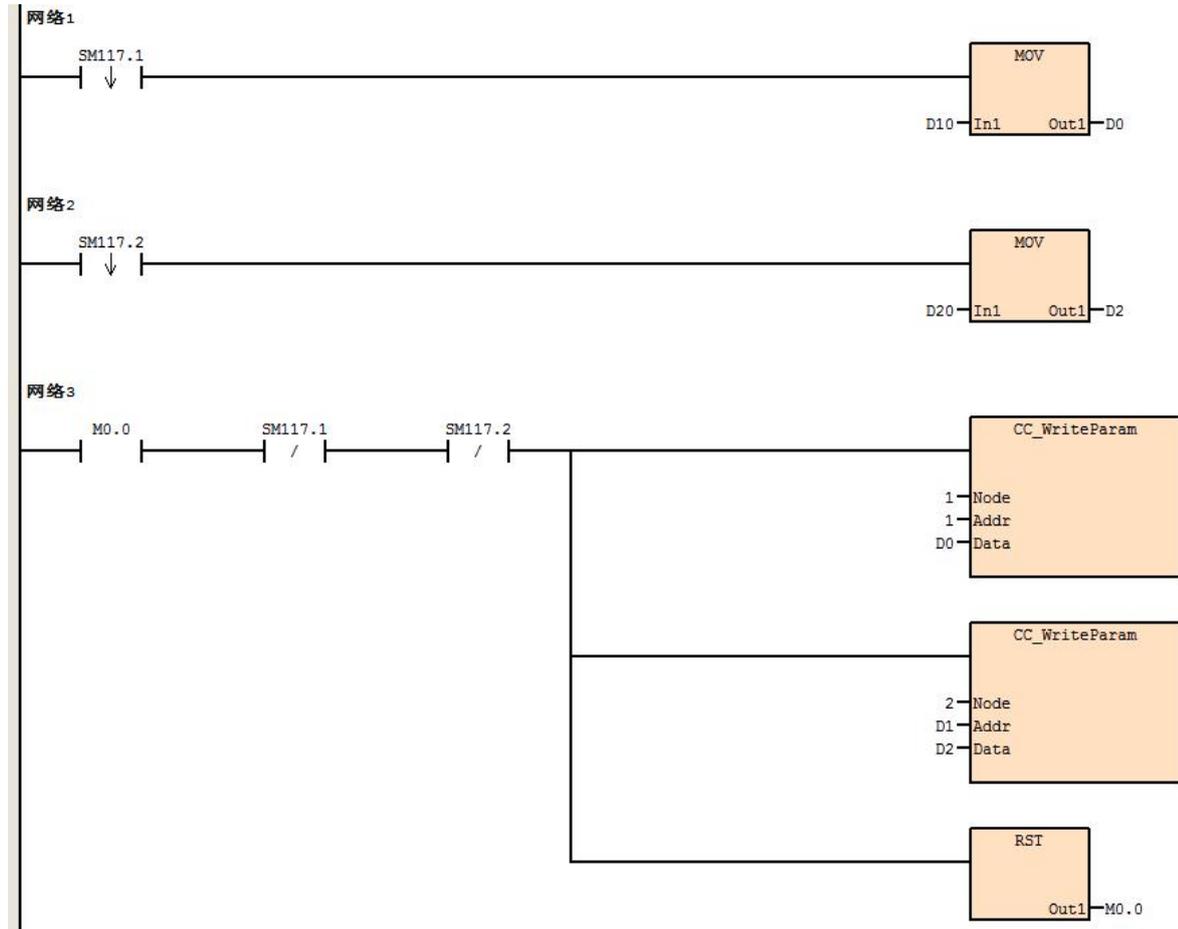
- 1, 该指令的功能是, 将 Data 装置中的数据, 写入目标节点站号 Node 的寄存器地址 Addr 的菜单中。

**程序范例:**

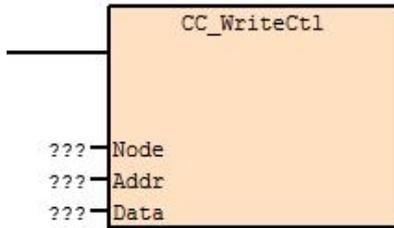
(见下页)

例:

- 当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即可对站号 1 地址为 1 的写入 D0 装置中的数据，同时对站号 2 的装置 D1 指定的地址写入 D2 装置中的数据，写入过程中站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 置 ON；
- 当站号 1 完成写入数据时，即 SM117.1 忙碌标志从 ON->OFF 时，将 D10 中的数据搬运到 D0 中；同样，当站号 2 完成写入数据时，即 SM117.2 忙碌标志从 ON->OFF 时，将 D20 中的数据搬运到 D2 中；
- 完成上述动作后，再将 M0.0 置为 ON，即可重复上一次的写入动作。



### 10.2.3 CC\_WriteCtl (写入单个控制寄存器)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Addr	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Data	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标节点站号, 范围 1~31;
- 2, Addr: 目标节点的数据地址;
- 3, Data: 想要写入目标节点目标地址的数据;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

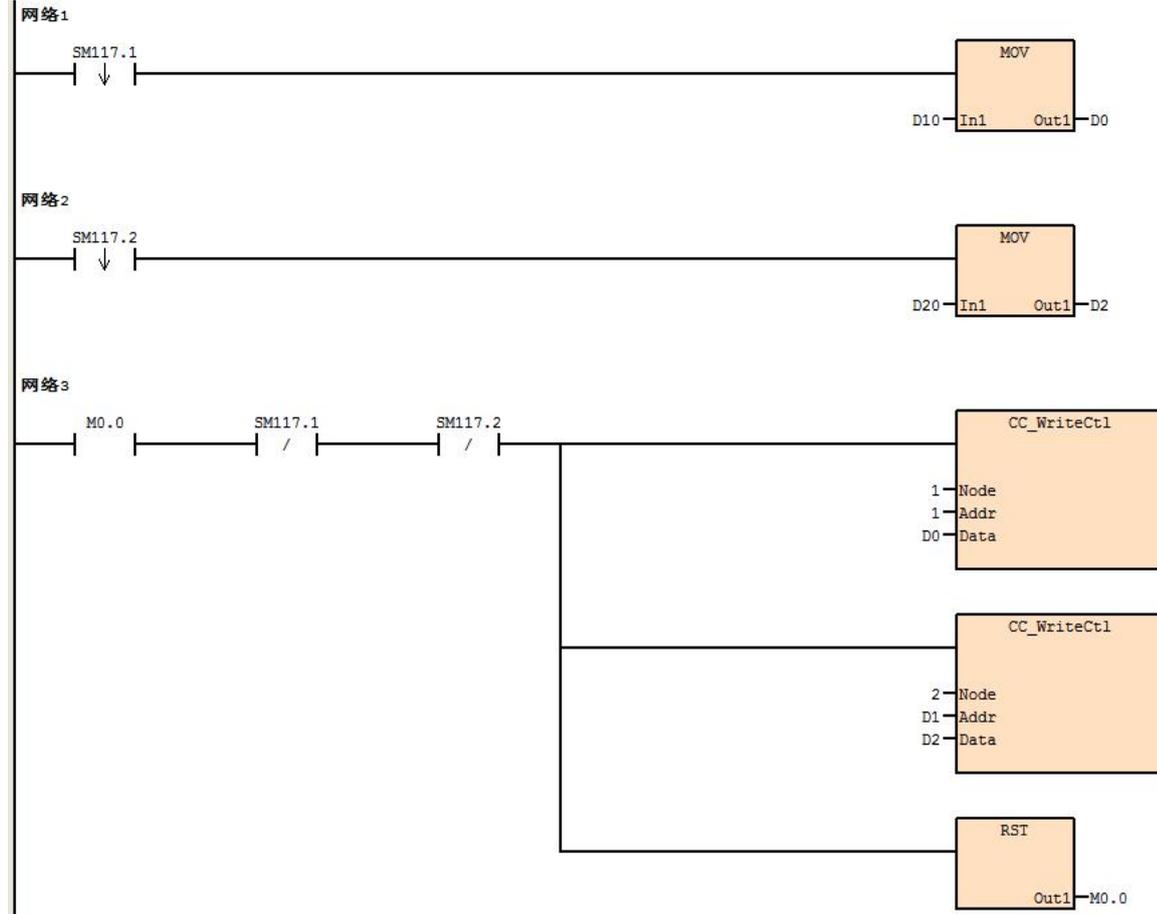
- 1, 该指令的功能是, 将 Data 装置中的数据, 写入目标节点站号 Node 的寄存器地址 Addr 的控制寄存器中。

**程序范例:**

(见下页)

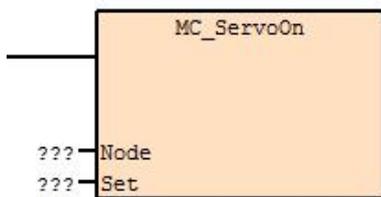
例:

- 当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即可对站号 1 地址为 1 的写入 D0 装置中的数据，同时对站号 2 的装置 D1 指定的地址写入 D2 装置中的数据，写入过程中站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 置 ON；
- 当站号 1 完成写入数据时，即 SM117.1 忙碌标志从 ON->OFF 时，将 D10 中的数据搬运到 D0 中；同样，当站号 2 完成写入数据时，即 SM117.2 忙碌标志从 ON->OFF 时，将 D20 中的数据搬运到 D2 中；
- 完成上述动作后，再将 M0.0 置为 ON，即可重复上一次的写入动作。



### 10.3 伺服及手轮运动控制指令

#### 10.3.1 MC\_ServoOn (伺服使能)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	EL							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 伺服使能设置位, 置 ON 使能伺服, 置 OFF 伺服失能;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

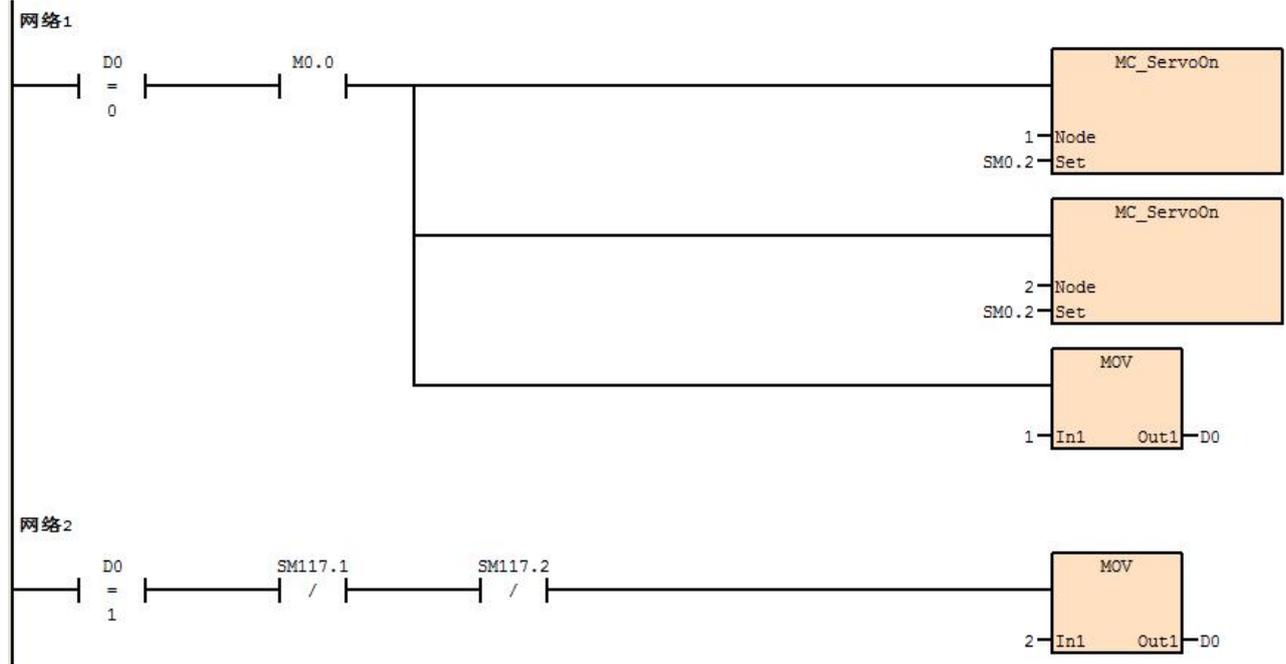
- 1, 该指令的功能是, 控制目标伺服使能或失能;
- 2, 伺服是否处于使能状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

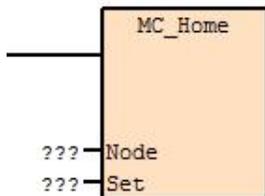
(见下页)

例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），设置 M0.0 为 ON，即可对站号 1 和站号 2 的伺服发送“伺服使能”命令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“伺服使能”命令，即 SM117.1 和 SM117.2 都是 OFF 时，将数值 2 搬移到 D0 中，进入步骤 2。



### 10.3.2 MC\_Home (原点回归)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	EL							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 伺服原点回归控制位, 置 ON 开始原点回归, 置 OFF 伺服停止原点回归;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 原点回归开始, 置 ON 对应站号的**运动控制标志 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到原点回归成功, **运动控制标志 SM** 置 OFF; 注意若伺服故障或未使能, 请使用停止原点回归命令或 MC\_Stop 指令, 使 MC\_Home 指令结束;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿触发执行运动控制指令。
- 4, 该指令是利用伺服自身的回原点功能, 在使用该指令之前, 请确认已在伺服菜单中设置完成原点位置和正反转极限位置。

**指令功能说明:**

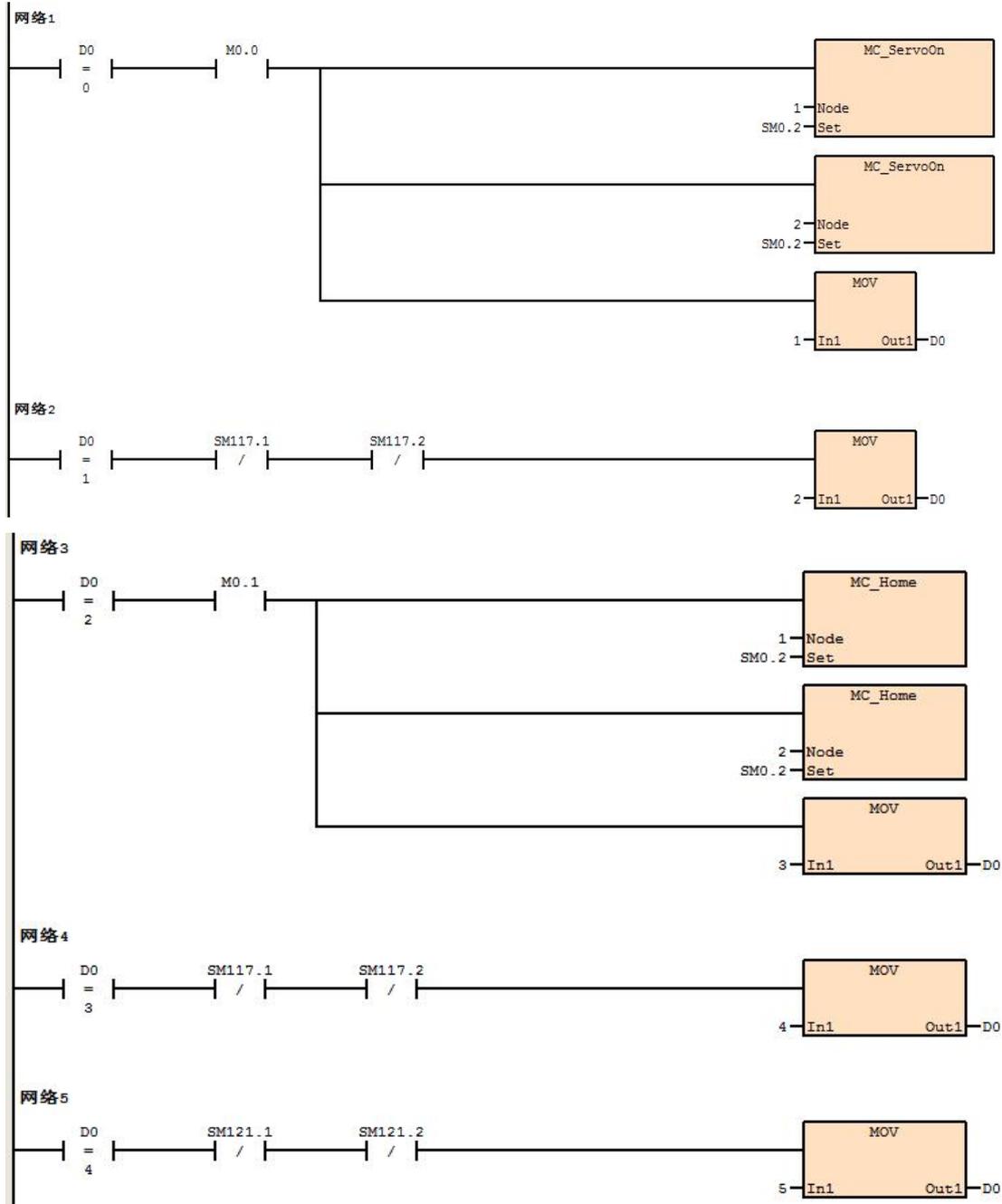
- 1, 该指令的功能是, 控制目标伺服开始或停止原点回归运转;
- 2, 当伺服成功完成原点回归时, 将会把对应站号的 SD832~SD891: 伺服 N 当前的程序绝对位置 数值复位为 0。

**程序范例:**

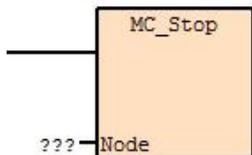
(见下页)

例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF, 设置 M0.0 为 ON, 即对站号 1 和站号 2 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服使能”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 2;
- 在步骤 2 中, 置位 M0.1 为 ON, 即对站号 1 和站号 2 的伺服发送“原点回归”命令, 进入步骤 3;
- 在步骤 3 中, 等待站号 1 和站号 2 都接收到“原点回归”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 4;
- 在步骤 4 中, 等待站号 1 和站号 2 确认完成了“原点回归”运转过程, 即运动控制标志 SM121.1 和 SM121.2 都是 OFF 时, 进入步骤 5。



### 10.3.3 MC\_Stop (停止指令)



变量名	数据类型	输入输出	字/双字装置																
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L	
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	

**参数说明:**

1, Node: 目标伺服站号, 范围 1~30;

**指令执行注意点:**

1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;

**指令功能说明:**

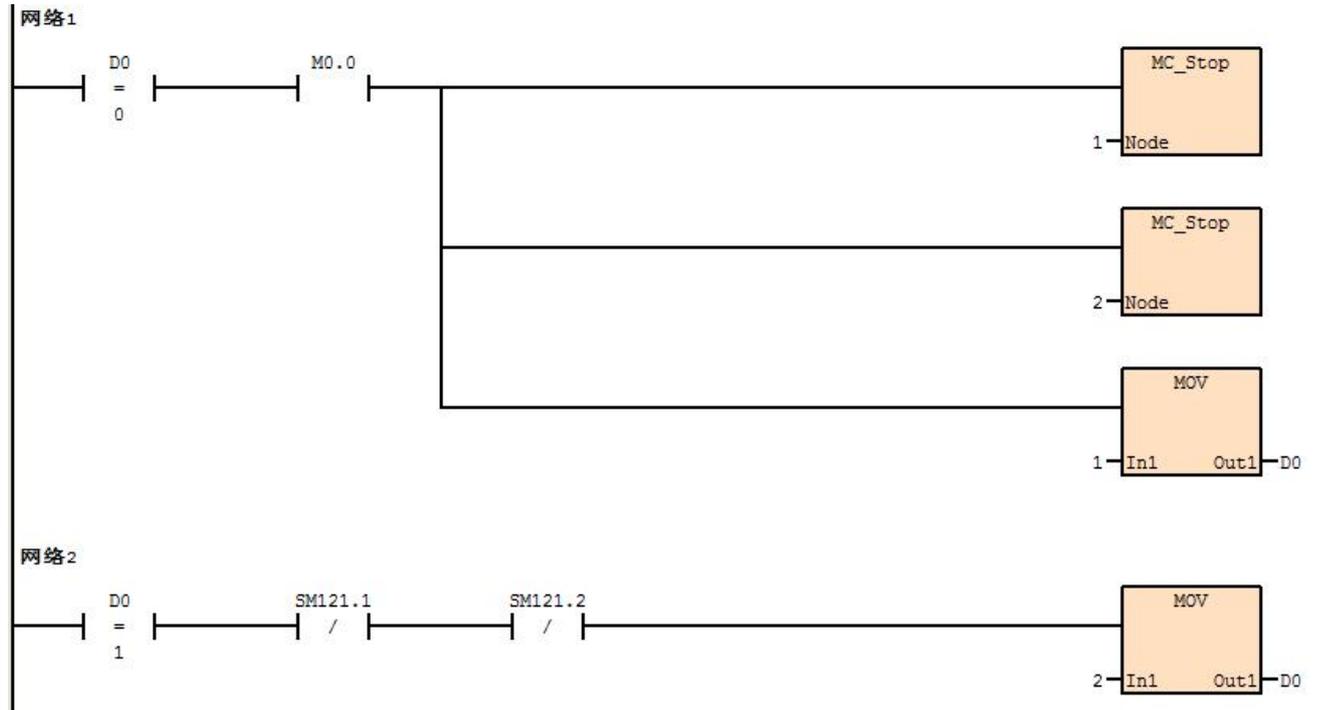
- 1, 该指令的功能是, 控制目标伺服从位置控制中的位移指令中, 停止执行;
- 2, 当伺服正在执行位移指令时, 可以使用该指令让伺服停止继续执行, 剩余的未执行位移距离不再继续执行, 若是使用多轴位移指令, 则关联的其他伺服也将停止执行;
- 3, 使用 SM116.5 伺服停止方式选择, 可以用来选择伺服的停止方式是减速停止, 还是不加减速直接停止; 当 SM116.5 为 OFF 时, 伺服将根据对应轴设定的加减速时间进行减速停止; 当 SM116.5 为 ON 时, 对应轴不加减速直接停止。

**程序范例:**

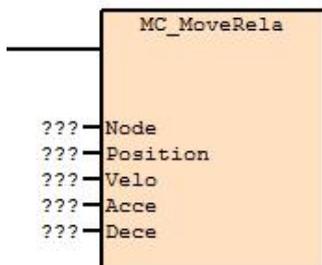
(见下页)

例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的正在运转时, 即**运动控制标志位** SM121.1 和 SM121.2 均为 ON, (SM121.1 对应站号 1 的运动控制标志, SM121.2 对应站号 2 的运动控制标志), 置位 M0.0 为 ON, 即可对站号 1 和站号 2 的伺服做“伺服停止”命令; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服停止”命令, 即 SM121.1 和 SM121.2 都是 OFF 时, 将数值 2 搬移到 D0 中, 进入步骤 2.



### 10.3.4 MC\_MoveRela (相对位移)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Position	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Acce	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Dece	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Position: 相对当前位置位移的目标单元个数; 数值为正, 伺服正转; 数值为负, 伺服反转;
- 3, Velo: 伺服运转的目标速度, 单位: Unit/s, 设定范围 1~10,000,000;
- 4, Acce: 加速度, 0~10,000,000, 单位: Unit/s<sup>2</sup>, 设定为 0 表示没有加速直接到达目标速度;
- 5, Dece: 减速度, 0~10,000,000, 单位: Unit/s<sup>2</sup>, 设定为 0 表示没有减速直接停止或到达变更后的速度。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令。

**指令功能说明:**

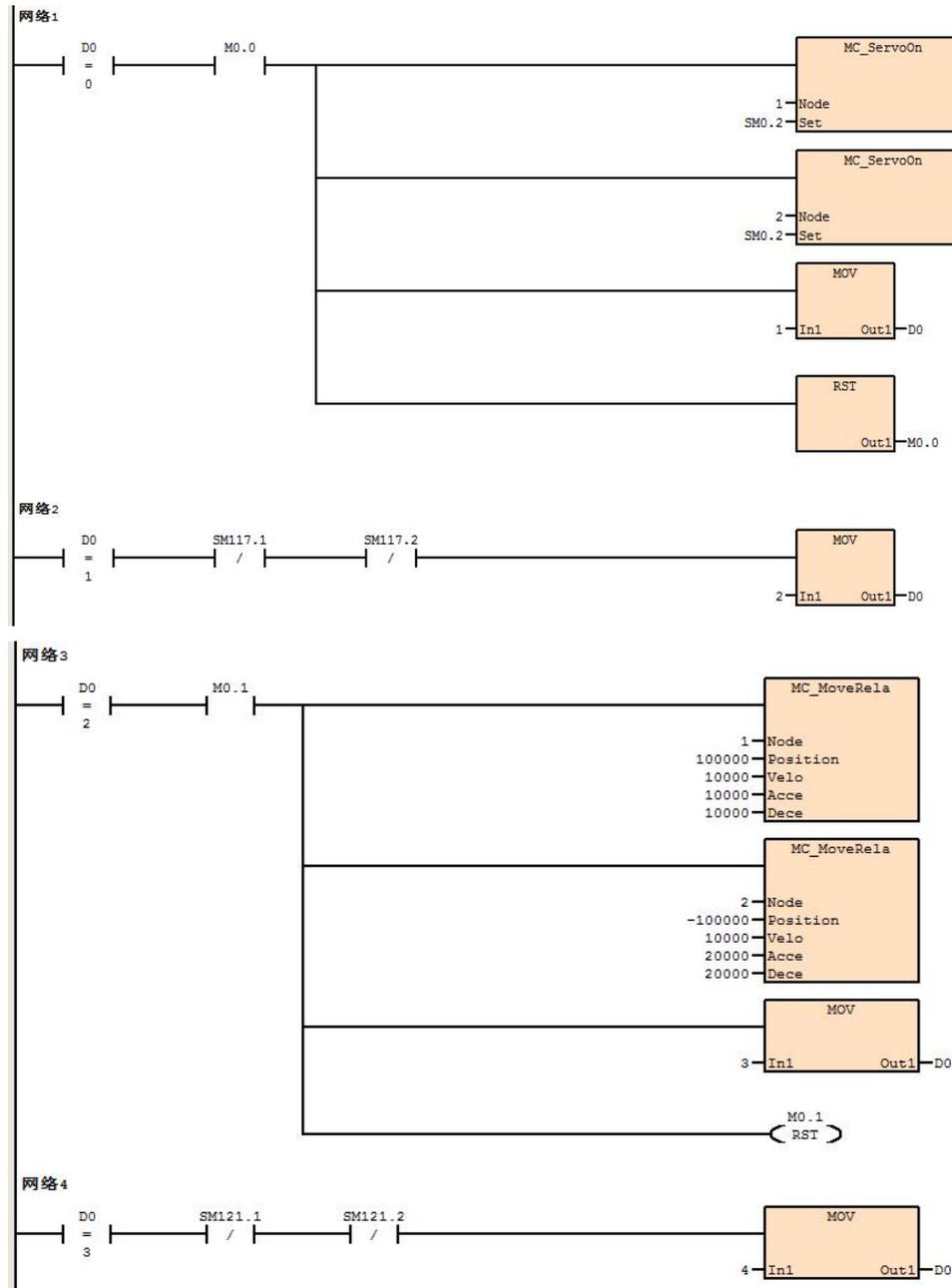
- 1, 该指令的功能是, 控制目标伺服以目标转速, 运转相对当前位置的单元个数;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行;
- 3, 可使用 SD662~SD721 查看机械当前真实的绝对位置 (对应伺服原点)。

**程序范例:**

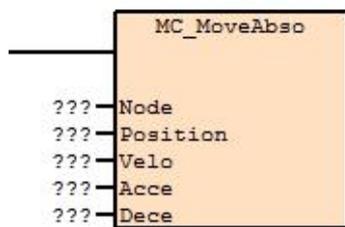
(见下页)

例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF，置位 M0.0 为 ON，即对站号 1 和站号 2 的伺服发送“伺服使能”命令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“伺服使能”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，当站号 1 和站号 2 的**运动控制标志位 SM121.1 和 SM121.2** 均为 OFF，置位 M0.1 为 ON，即对伺服 1 进行**正转** 100000 单元，最大速度 10000 Unit/s，加减速时间 1 秒运转控制，对伺服 2 进行**反转** 100000 单元，最大速度 10000 Unit/s，加减速时间 0.5 秒运转控制；伺服运转过程中**运动控制标志位 SM121.1 和 SM121.2** 置 ON；同时将数值 3 搬移到 D0 装置中，进入步骤 3；
- 在步骤 3 中，等待站号 1 和站号 2 运转完成，即 SM121.1 和 SM121.2 都是 OFF 时，将数值 4 搬移到 D0，进入步骤 4。



### 10.3.5 MC\_MoveAbso (绝对位移)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Position	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Acce	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Dece	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Position: 位移的目标绝对位置数值, 该轴的绝对位置是 32 位数据, 基于程序绝对位置 SD832~SD891 位移;
- 3, Velo: 伺服运转的目标速度, 单位: Unit/s, 设定范围 1~10,000,000;
- 4, Acce: 加速度, 0~10,000,000, 单位: Unit/s<sup>2</sup>, 设定为 0 表示没有加速直接到达目标速度;
- 5, Dece: 减速度, 0~10,000,000, 单位: Unit/s<sup>2</sup>, 设定为 0 表示没有减速直接停止或到达变更后的速度。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令。

**指令功能说明:**

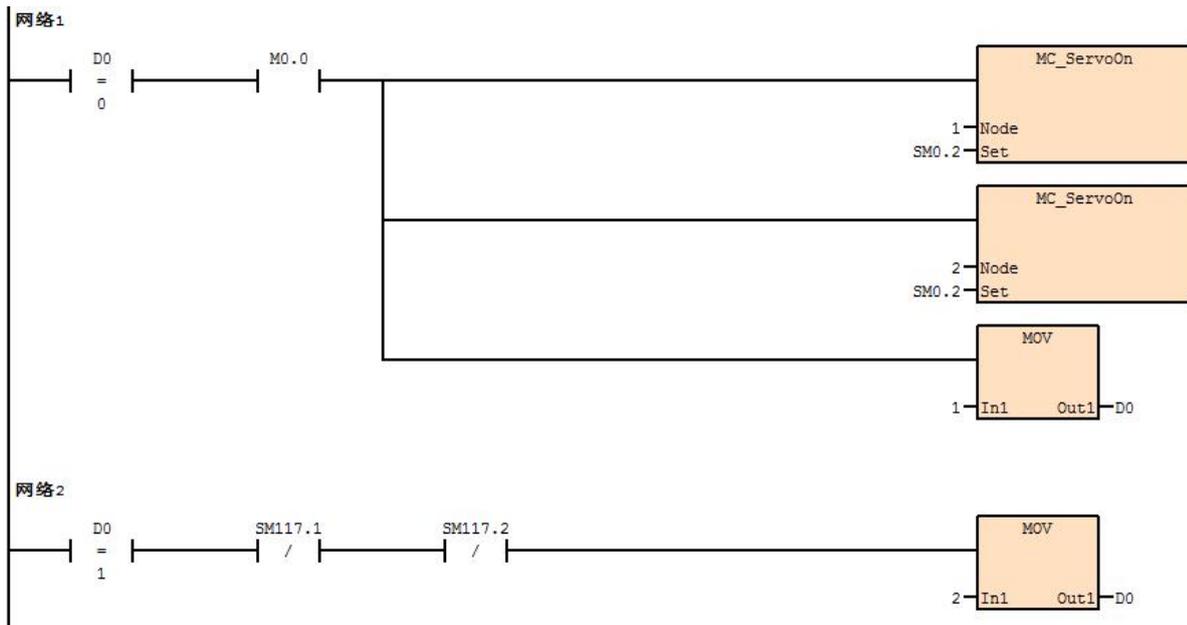
- 1, 该指令的功能是, 控制目标伺服以目标转速, 运转到目标绝对位置;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行;
- 3, 绝对位置的依据 SD832~SD891 对应位移, 例如站号 1 的伺服当前绝对位置是 SD832~SD833 组成, 当前数值是 800, 那么填入目标绝对位置为 0 时, 指令执行时, 即伺服向反方向运转 800 个单元位置;
- 4, 可使用 SD662~SD721 查看机械当前真实的绝对位置 (对应伺服原点)。

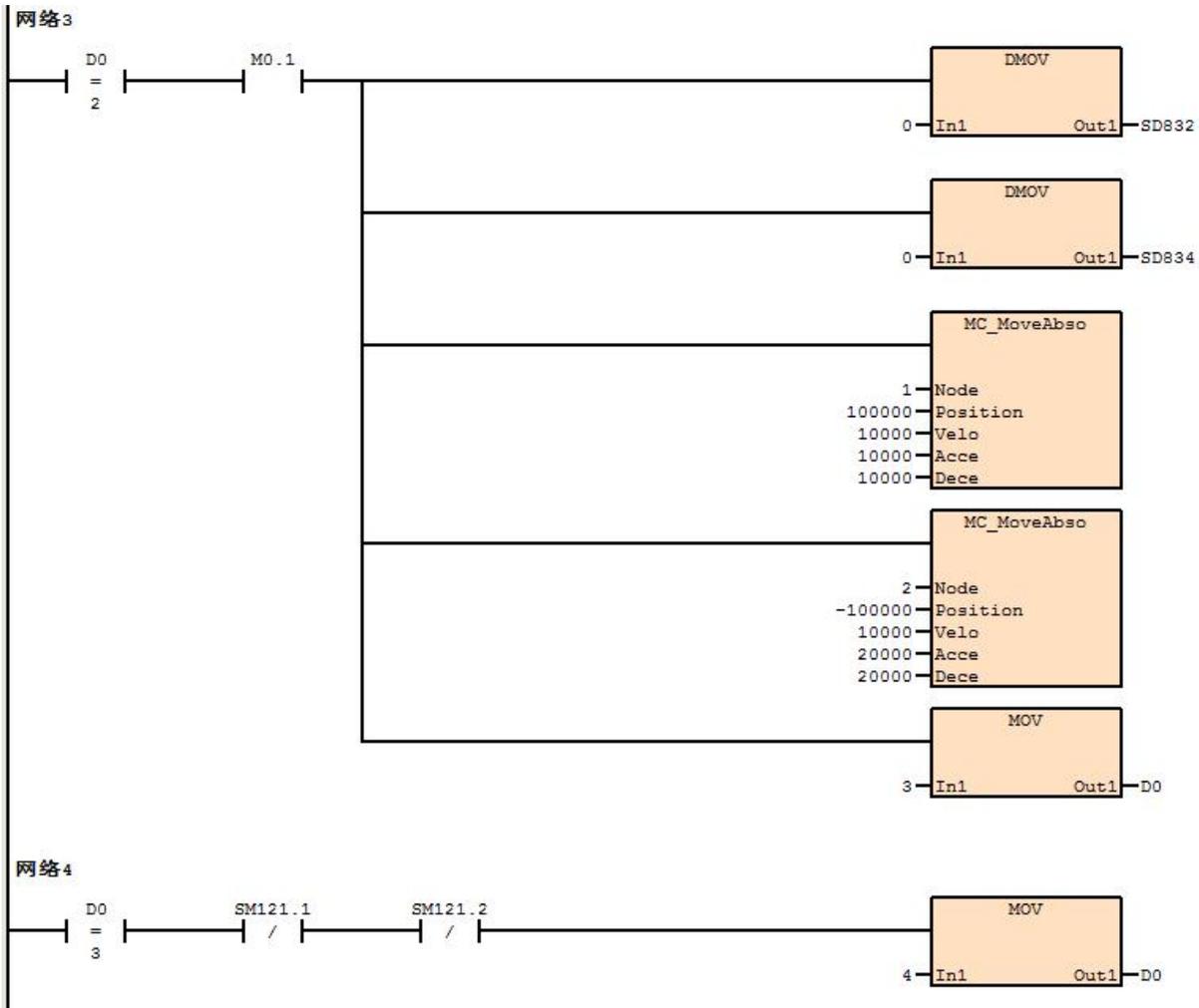
**程序范例:**

(见下页)

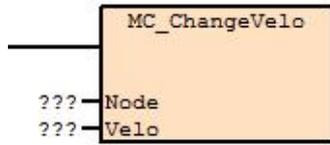
例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF, 置位 M0.0 为 ON, 即对站号 1 和站号 2 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服使能”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 2;
- 在步骤 2 中, 当站号 1 和站号 2 的**运动控制标志位 SM121.1 和 SM121.2** 均为 OFF, 置位 M0.1 为 ON, 先将伺服 1 和伺服 2 的当前绝对位置都设置为 0 (使用将数值 0 搬移到 SD832~SD833 以及 SD834~SD835 实现), 那么 MC\_MoveAbs 指令中设置的运转目标位置, 即是对应当前位置 0 的数值; 即对伺服 1 进行**正转** 100000 单元, 最大转速 10000 unit/s, 加减速时间 1 秒运转控制, 对伺服 2 进行**反转** 100000 单元, 最大转速 10000 Unit/s, 加减速时间 0.5 秒运转控制; 伺服运转过程中运动控制标志位 **SM121.1 和 SM121.2** 置 ON; 同时将数值 3 搬移到 D0 装置中, 进入步骤 3;
- 在步骤 3 中, 等待站号 1 和站号 2 运转完成, 即 **SM121.1 和 SM121.2** 都是 OFF 时, 将数值 4 搬移到 D0, 进入步骤 4.





### 10.3.6 MC\_ChangeVelo (位移速度变更)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Velo: 伺服运转的目标转速, 单位: Unit/s, 设定范围 0~10,000,000;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令指定对象, 须正在使用“绝对位移 (MC\_MoveAbso)”或“相对位移 (MC\_MoveRela)”指令, 对其他指令无效。

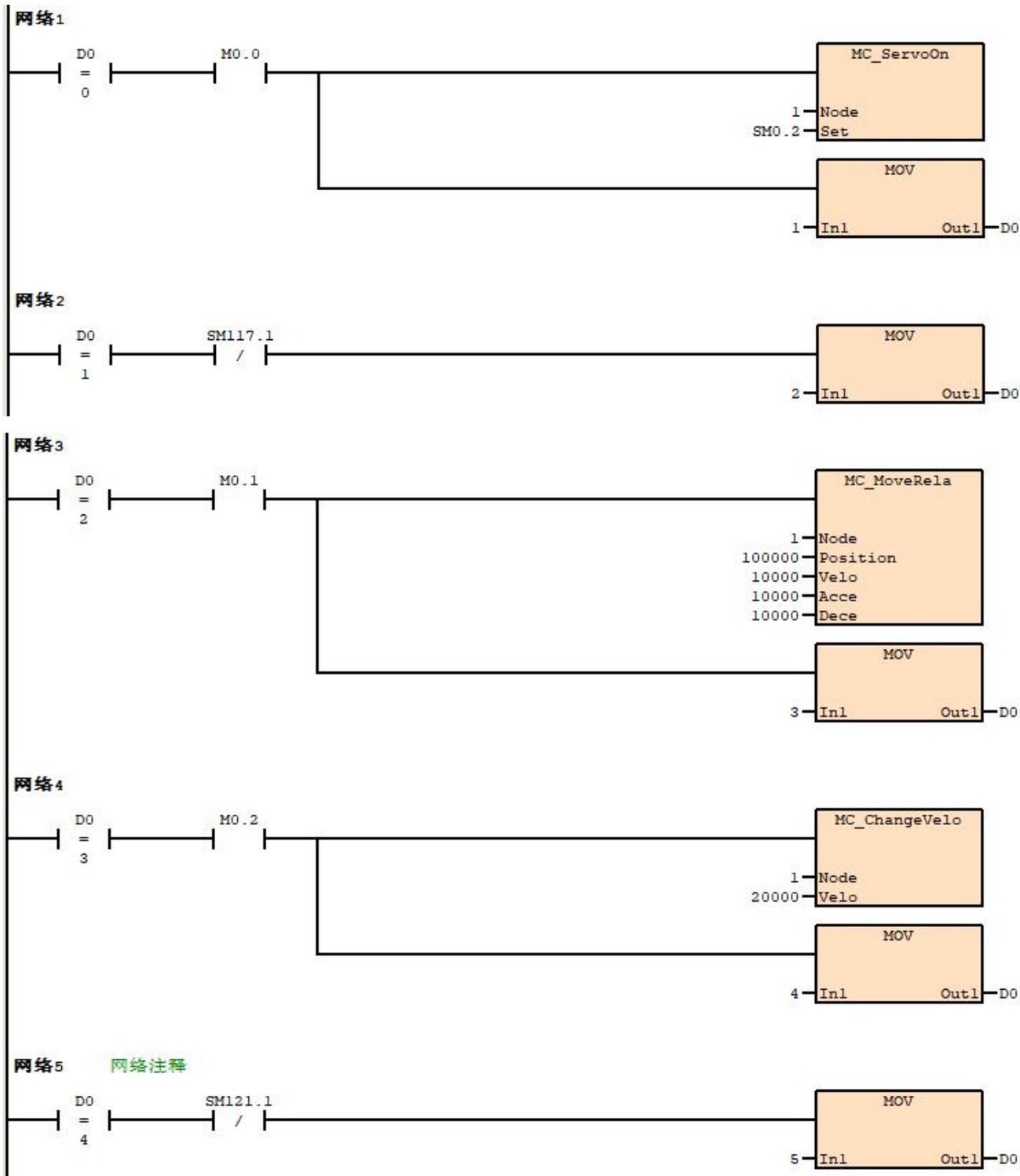
**指令功能说明:**

- 1, 该指令的功能是, 对正在使用“绝对位移 (MC\_MoveAbso)”或“相对位移 (MC\_MoveRela)”指令的目标伺服, 根据位移指令中的加减速设定, 变更到目标转速 (转速设为 0 即暂停运转)。

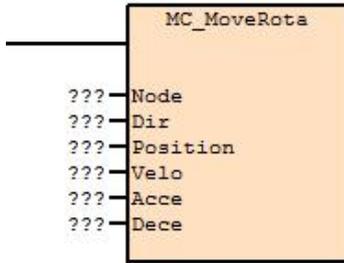
**程序范例:**

**例:**

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 的忙碌标志位 SM117.1 为 OFF, 置位 M0.0 为 ON, 即对站号 1 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 完成“伺服使能”命令, 即 SM117.1 是 OFF 时, 进入步骤 2;
- 在步骤 2 中, 当站号 1 的**运动控制标志位 SM121.1** 为 OFF, 置位 M0.1 为 ON, 即对伺服 1 进行**正转** 100000 单元, 最大速度 10000 Unit/s, 加减速时间 1 秒运转控制; 伺服运转过程中**运动控制标志位 SM121.1** 置 ON; 同时将数值 3 搬移到 D0 装置中, 进入步骤 3;
- 在步骤 3 中, 置位 M0.2 为 ON, 即对伺服 1 进行加减速时间为 1 秒, 目标速度为 20000 Unit/s 的位移速度变更;
- 在步骤 4 中, 等待站号 1 运转完成, 即 **SM121.1** 是 OFF 时, 将数值 5 搬移到 D0, 进入步骤 5。



### 10.3.7 MC\_MoveRota (旋转轴绝对位移)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Dir	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Position	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Acce	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Dece	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Dir: 旋转方向, 0: 最近距离方向旋转; 1: 正转方向旋转; 2: 反转方向旋转; 其他数值无效不执行指令;
- 3, Position: 位移的目标绝对位置, 范围: 0~ 机械旋转一圈单元数, 超出不执行; 基于程序绝对位置 SD832~SD891 位移;  
 (若需要在旋转轴设置下位移多圈, 可使用 MC\_MoveAbso 或 MC\_MoveRela 指令, 当成线性轴控制);  
 如果目标绝对位置和程序绝对位置相同; 使用最近距离方向旋转, 则不运转; 使用其他方向, 则运转一圈。
- 4, Velo: 伺服运转的目标速度, 单位: Unit/s, 设定范围 1~10,000,000;
- 5, Acce: 加速度, 0~10,000,000, 单位: Unit/s<sup>2</sup>, 设定为 0 表示没有加速直接到达目标速度;
- 6, Dece: 减速度, 0~10,000,000, 单位: Unit/s<sup>2</sup>, 设定为 0 表示没有减速直接停止或到达变更后的速度。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令;
- 4, 当该指令触发执行, 对应站号的程序绝对位置 SD832~SD891 若在旋转周期范围外, 则会自动对周期单元数取余数并赋予程序绝对位置 SD832~SD891, 再去计算绝对位移数值和方向;

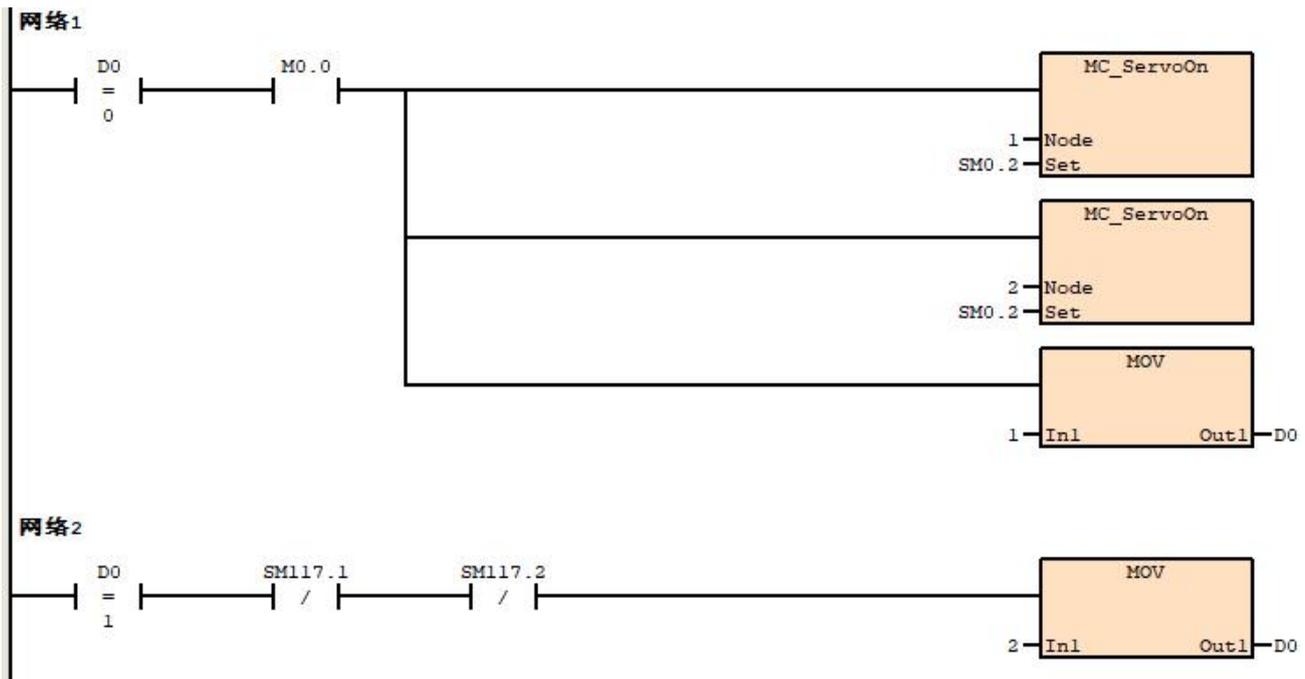
**指令功能说明:**

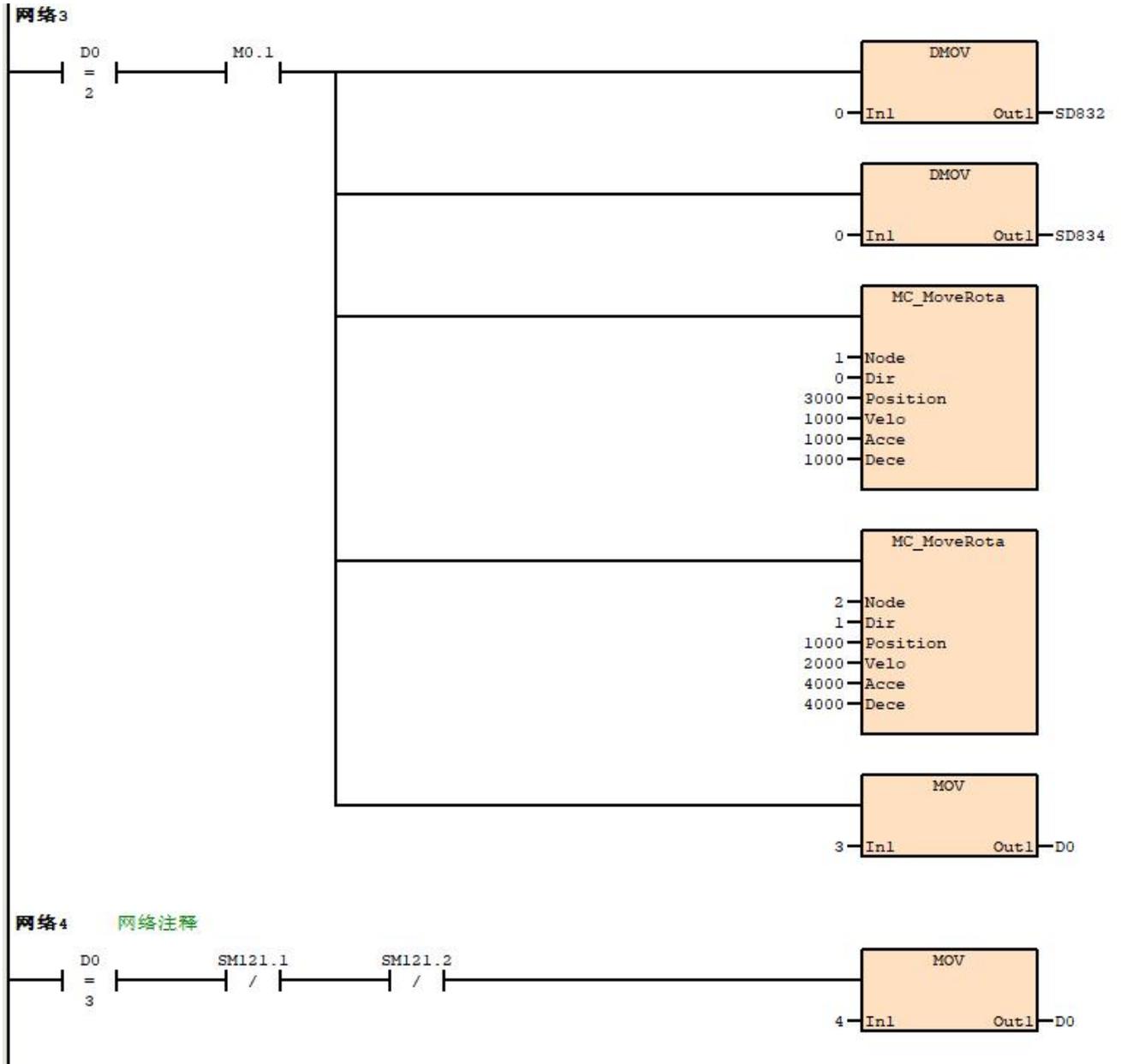
- 1, 该指令的功能是, 控制目标设定为旋转轴的伺服以目标转速, 以设定的方向运转到目标绝对位置;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行;
- 3, 绝对位置的依据 SD832~SD891 对应位移, 例如站号 1 的伺服当前绝对位置是 SD832~SD833 组成, 当前数值是 20, 机械旋转一周单元数是 360, 那么填入目标绝对位置为 350 时, 方向设定为 0: 最近方向旋转, 指令执行时, 即伺服向反方向运转 30 个单元;
- 4, 使用 SD662~SD721 查看机械实际绝对位置 (对应伺服原点), 用来确认当前实际机械位置和程序绝对位置是否一致。

程序范例:

例:

- 在例程中，轴类型为**旋转轴**，伺服电机转一圈对应机械终端行程为 3600；D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF，置位 M0.0 为 ON，即对站号 1 和站号 2 的伺服发送“伺服使能”命令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“伺服使能”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，当站号 1 和站号 2 的**运动控制标志位 SM121.1 和 SM121.2** 均为 OFF，置位 M0.1 为 ON，先将伺服 1 和伺服 2 的当前绝对位置都设置为 0（使用将数值 0 搬移到 SD832~SD833 以及 SD834~SD835 实现），那么 MC\_MoveRota 指令中设置的运转目标位置，即是对应当前位置 0 的数值；即对伺服 1 进行**反转** 600 单元（反转至目标绝对位置的行程相较于正转的行程更短），最大转速 1000 uint/s，加减速时间 1 秒运转控制，对伺服 2 进行**正转** 1000 单元，最大转速 2000 Unit/s，加减速时间 0.5 秒运转控制；伺服运转过程中运动控制标志位 **SM121.1 和 SM121.2** 置 ON；同时将数值 3 搬移到 D0 装置中，进入步骤 3；
- 在步骤 3 中，等待站号 1 和站号 2 运转完成，即 **SM121.1 和 SM121.2** 都是 OFF 时，将数值 4 搬移到 D0，进入步骤 4。





### 10.3.8 MC\_HDWToSVO (伺服跟随电子手轮位移)

MC HDWToSVO

#### 指令执行注意点：

- 1, 该指令是需要能流维持导通执行, 在能流导通期间周期读取, 在能流断开后, 该指令不再执行;
- 2, 使用前先确认 SM116.3 电子手轮忙碌标志位为 OFF, 指令执行时, SM116.3 置 ON;
- 3, 该指令在一个项目中只能编写一次, 否则会出现使用异常。

#### 指令功能说明：

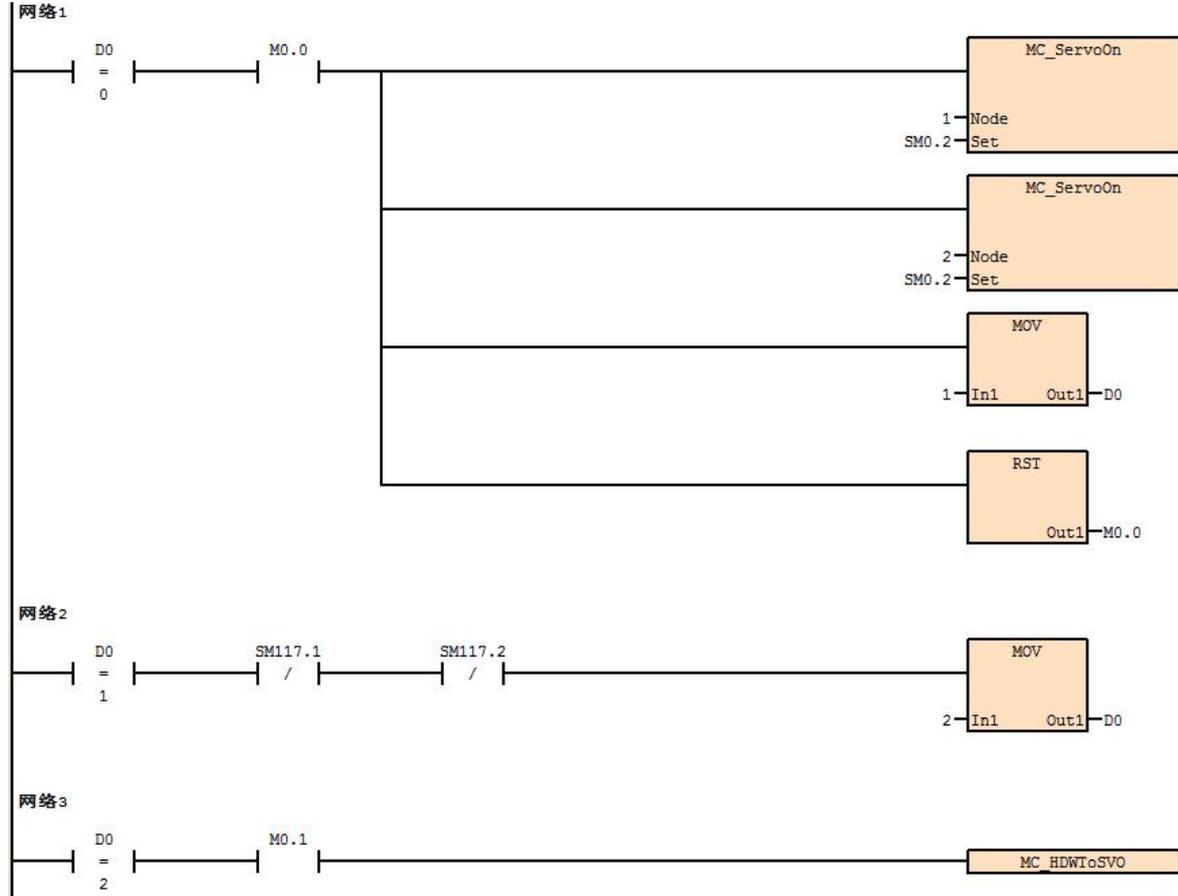
- 1, 该指令的功能是, 自动读取电子手轮的位移及选定轴, 输出到根据 SD901~SD907 的设定的对应伺服站号;
- 2, 电子手轮当前的状态信息, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看;
- 3, SD901 到 SD907 分别对应电子手轮切换选定轴 1~7 对应的伺服站号设定, 默认是 1 轴对应站号 1, 轴 7 对应站号 7, 使用者可根据实际伺服站号, 进行修改设定。

#### 程序范例：

(见下页)

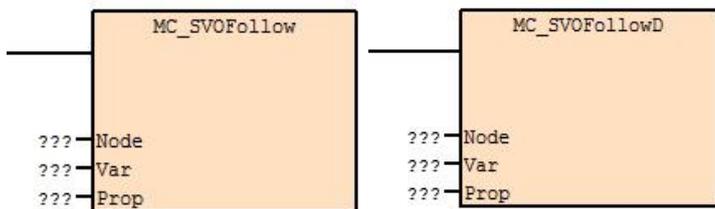
例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF, 置位 M0.0 为 ON, 即对站号 1 和站号 2 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服使能”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 2;
- 在步骤 2 中, SD901 和 SD902 使用默认数值 1 和 2, 即表明电子手轮中的 X 轴对应伺服 1, Y 轴对应伺服 2; 置位 M0.1 为 ON, “伺服跟随电子手轮位移”指令开始执行, 当电子手轮选中了 X 轴, 伺服 1 将会跟随用户手摇轮的倍率和脉冲数进行完全的跟随位移; 电子手轮切换到 Y 轴后, 伺服 2 将会跟随用户手摇轮的倍率和脉冲数进行完全的跟随位移。



### 10.3.9 MC\_SVOfollow (伺服跟随变量装置位移)

### 10.3.10 MC\_SVOfollowD (伺服跟随变量装置位移)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Var	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■		
	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■			
Prop	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Var: 跟随的变量装置选择, MC\_SVOfollow 指令的变量装置是字装置类型, MC\_SVOfollowD 指令的变量装置是双字装置类型;
- 3, Prop: 跟随变量的比例设定, 设定范围 1~1000。

**指令执行注意点:**

- 1, 同一个目标伺服站号一个项目中只能编写一次, 否则会出现使用异常;
- 2, 该指令是需要能流维持导通执行, 在能流导通期间周期读取, 在能流断开后, 该指令不再执行;
- 3, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 4, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流去触发执行运动控制指令。

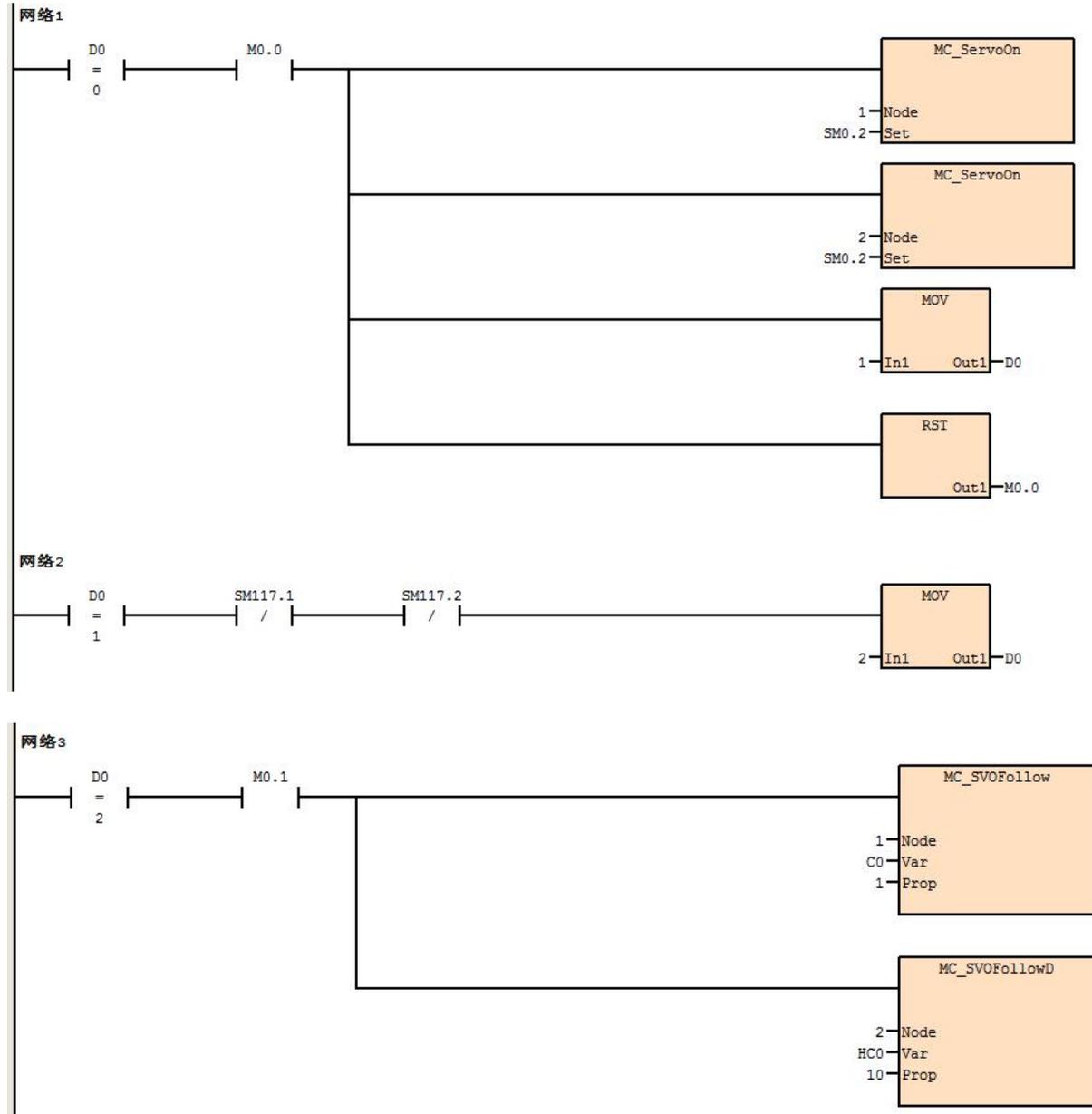
**指令功能说明:**

- 1, 该指令的功能是, 控制目标伺服跟随变量装置数值变化, 以设定比例进行位移, 例如变量装置变化为 1, 设定比例为 5, 那么伺服将位移 5;

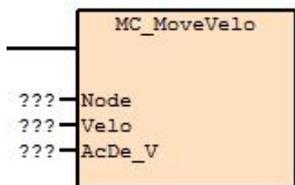
**程序范例:**

例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF, 置位 M0.0 为 ON, 即对站号 1 和站号 2 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服使能”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 2;
- 在步骤 2 中, 当站号 1 和站号 2 的**运动控制标志位 SM121.1 和 SM121.2** 均为 OFF, 置位 M0.1 为 ON, 伺服 1 开始完全跟随 C0 的变化量进行运转, 伺服 2 开始跟随 HC0 变化量的 10 倍进行运转; 伺服运转过程中运动控制标志位 **SM121.1** 和 **SM121.2** 置 ON; 当 M0.1 置 OFF 时, 两个指令都停止执行, 运动控制标志位 **SM121.1** 和 **SM121.2** 置 OFF。



### 10.3.11 MC\_MoveVelo (位置模式下速度控制)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
AcDe_V	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Velo: 伺服运转的目标速度, 单位: Unit / S, 设定范围-10,000,000~10,000,000; 其中负值表示反转, 数值为 0 停止运转;
- 3, AcDe\_V: 从当前运转速度到目标速度的加/减速; 单位 Unit / S<sup>2</sup>; 范围 0~100,000,000, 设置为 0 表示直接到达目标速度。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 该指令执行时, 不可执行其他运动控制类型指令; 但是可以继续触发执行指令, 用于修改转速和转向;
- 4, 超过设定范围时, 指令不执行。

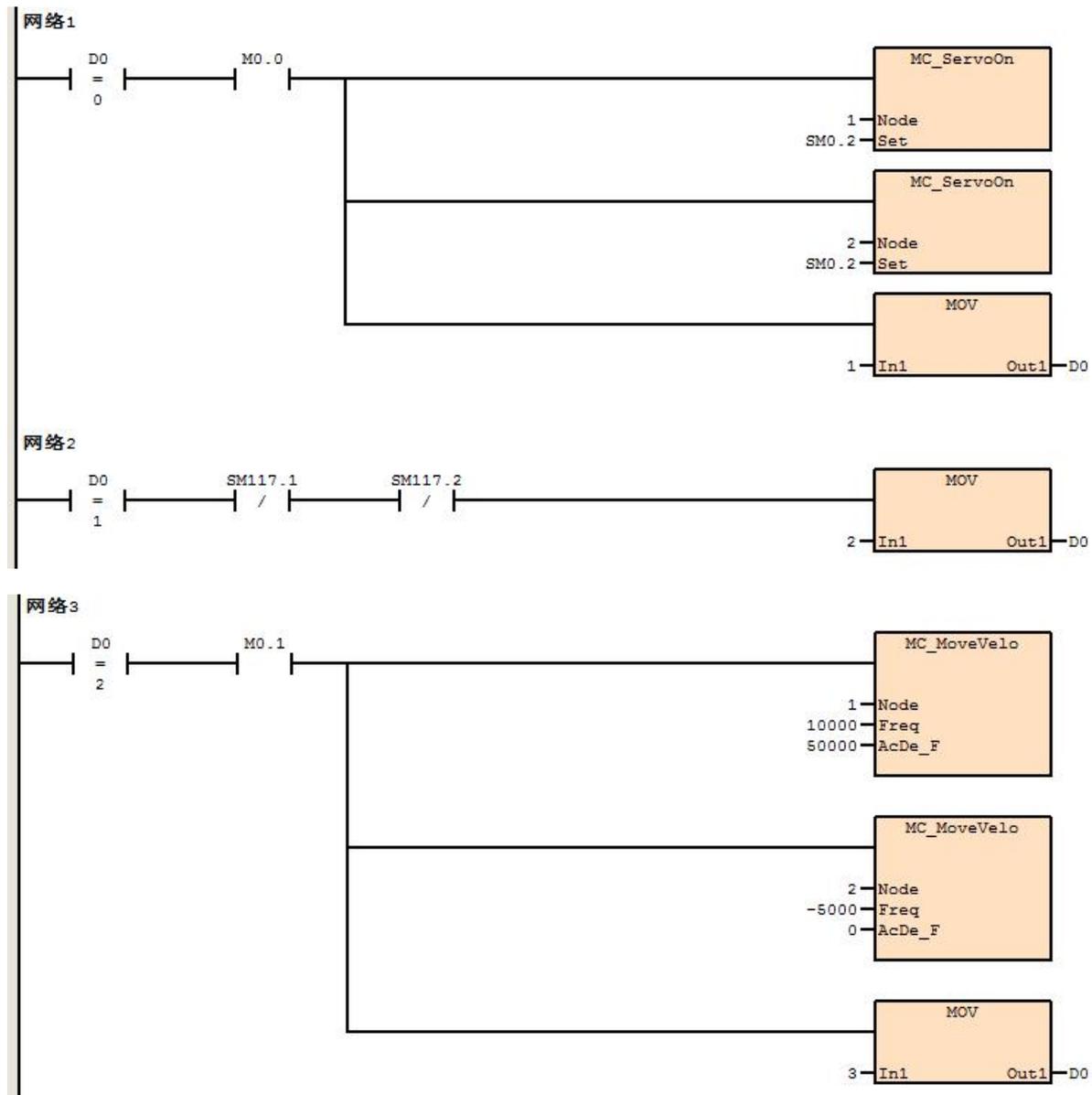
**指令功能说明:**

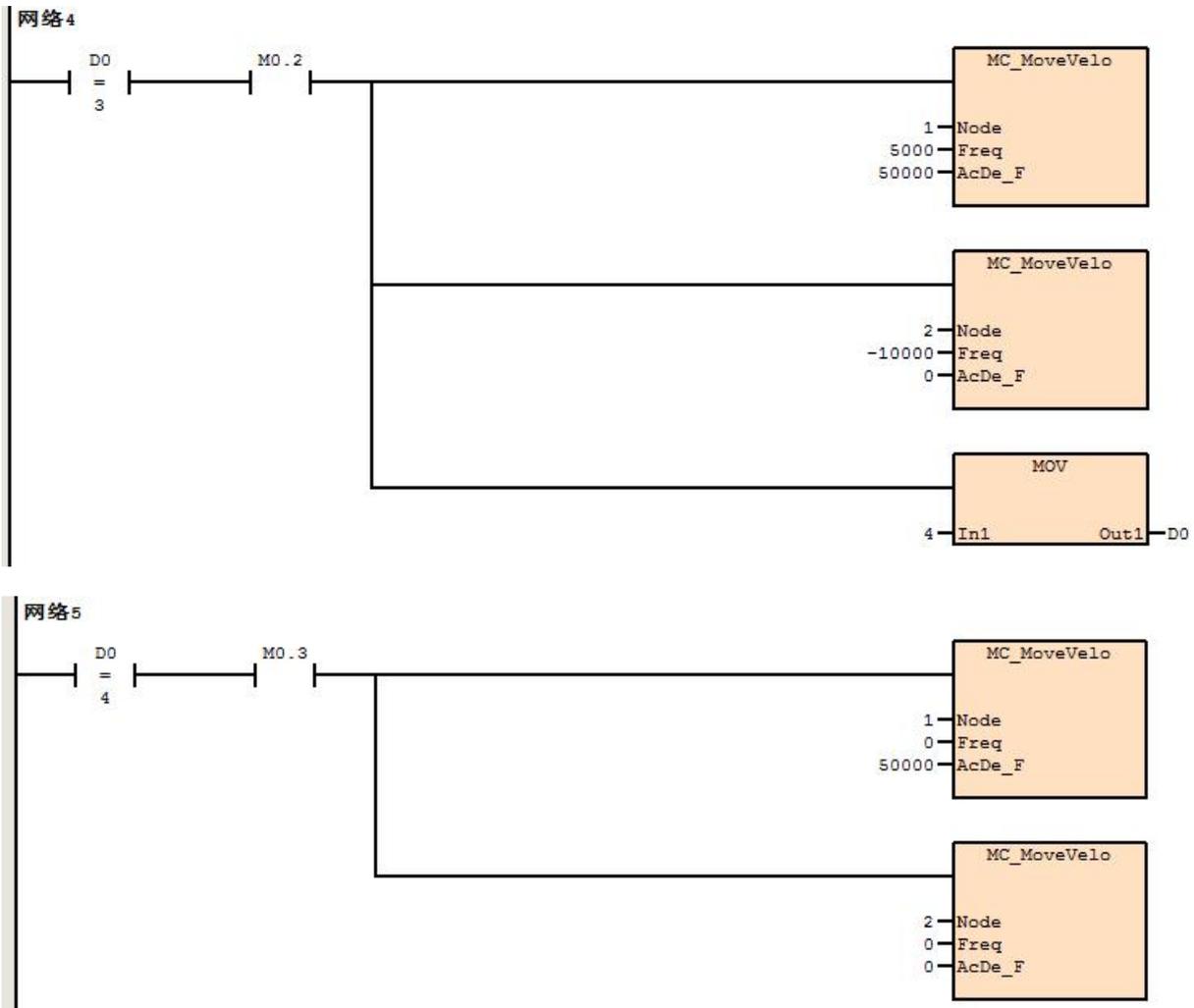
- 1, 该指令的功能是, 在伺服位置模式下, 进行速度控制, 可控制目标转速, 转向以及加减速;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行, 也可以将 Velo 设定为 0, 再上升沿触发执行一次该指令。

**程序范例:**

例:

- 在例程中，伺服的运转模式是默认为位置模式，不需要进行更改；
- D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF，置位 M0.0 为 ON，即对站号 1 和站号 2 的伺服发送“伺服使能”命令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“伺服使能”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，置位 M0.1 为 ON，伺服 1 和伺服 2 开始同步运转，其中伺服 1 的加速度设置为 50,000 Unit/s<sup>2</sup>，从当前停止运转经过 0.2s 的加速到达设定目标转速正转 10,000 Unit/s，并持续运转；伺服 2 未通过加速直接反转运转到目标转速 5,000 Unit/s 并持续运转。之后进入步骤 3；
- 在步骤 3 中，置位 M0.2 为 ON，伺服 1 的减速度设置为 50,000 Unit/s<sup>2</sup>，从当前正转 10,000 Unit/s 的转速，经过 0.1s 减速到设定目标转速 5,000 Unit/s，并持续运转；伺服 2 未通过加速直接反转运转到目标转速 10,000 Unit/s 并持续运转。进入步骤 4；
- 在步骤 4 中，置位 M0.3 为 ON，伺服 1 的减速度设置为 50,000 Unit/s<sup>2</sup>，从当前正转 5,000 Unit/s 的转速，经过 0.1s 的时间减速到停止运转；伺服 2 不经过减速，直接停止运转。





### 10.3.12 MC\_LinRela (双轴相对直线插补)

变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
NodeX	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
NodeY	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
PosX	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
PosY	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
AcDe_T	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, NodeX: X 轴目标伺服站号, 范围 1~30;
- 2, NodeY: Y 轴目标伺服站号, 范围 1~30;
- 3, PosX: X 轴伺服相对当前位置位移的目标单元数; 数值为正, 伺服正转; 数值为负, 伺服反转;
- 4, PosY: Y 轴伺服相对当前位置位移的目标单元数; 数值为正, 伺服正转; 数值为负, 伺服反转;
- 5, Velo: X 轴的目标转速, Y 根据插补算法自动设定目标转速, 单位: Unit/s, 设定范围 1~10,000,000;
- 6, AcDe\_T: 加减速时间, 从 0 加速到目标转速的时间, 也是从目标转速减速到 0 的时间; 范围 0~2000, 单位: 0.01s。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令。

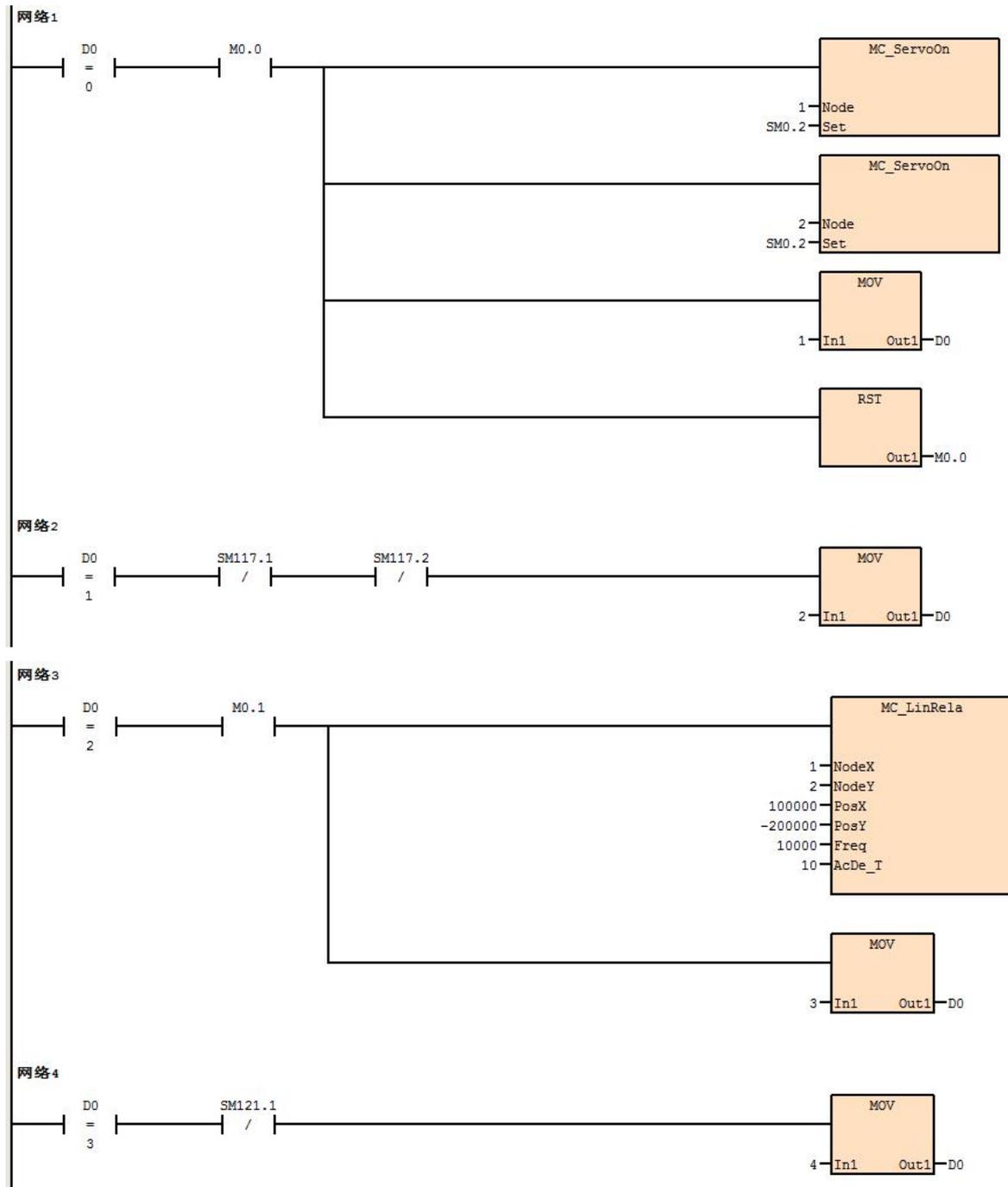
**指令功能说明:**

- 1, 该指令的功能是, 控制两轴目标伺服以设定 X 轴转速, 进行相对位置的直线插补;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行, 停止指令的 Node 可以选择两轴中任意一轴。

**程序范例:**

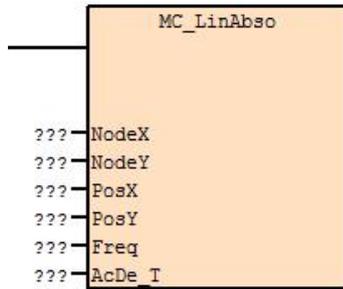
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF，置位 M0.0 为 ON，即对站号 1 和站号 2 的伺服发送“伺服使能”命令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“伺服使能”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，当站号 1 和站号 2 的运动控制标志位 SM121.1 和 SM121.2 均为 OFF，置位 M0.1 为 ON，伺服 1 和伺服 2 开始同步运转，伺服 1 以 1 秒时间加速到最大速度 10000Unit/s，伺服 2 以 1 秒时间加速到 20000Unit/s，做直线插补运动；伺服运转过程中运动控制标志位 SM121.1 和 SM121.2 置 ON；进入步骤 3；
- 在步骤 3 中，等待运动控制标志 SM121.1 和 SM121.2 置 OFF（选一个即可，指令执行完成会同时 OFF），进入步骤 4。





### 10.3.13 MC\_LinAbso (双轴绝对直线插补)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
NodeX	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
NodeY	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
PosX	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
PosY	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Velo	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
AcDe_T	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

#### 参数说明:

- 1, NodeX: X轴目标伺服站号, 范围 1~30;
- 2, NodeY: Y轴目标伺服站号, 范围 1~30;
- 3, PosX: X轴伺服位移的目标绝对位置, 该轴的绝对位置是 32 位数据, 基于 SD832~SD891 位移;
- 4, PosY: Y轴伺服位移的目标绝对位置, 该轴的绝对位置是 32 位数据, 基于 SD832~SD891 位移;
- 5, Velo: X轴的目标转速, Y轴根据插补算法自动设定目标频率, 单位: Unit/s, 设定范围 1~10,000,000;
- 6, AcDe\_T: 加减速时间, 从 0 加速到目标转速的时间, 也是从目标转速减速到 0 的时间; 范围 0~2000, 单位: 0.01s。

#### 指令执行注意点:

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令。

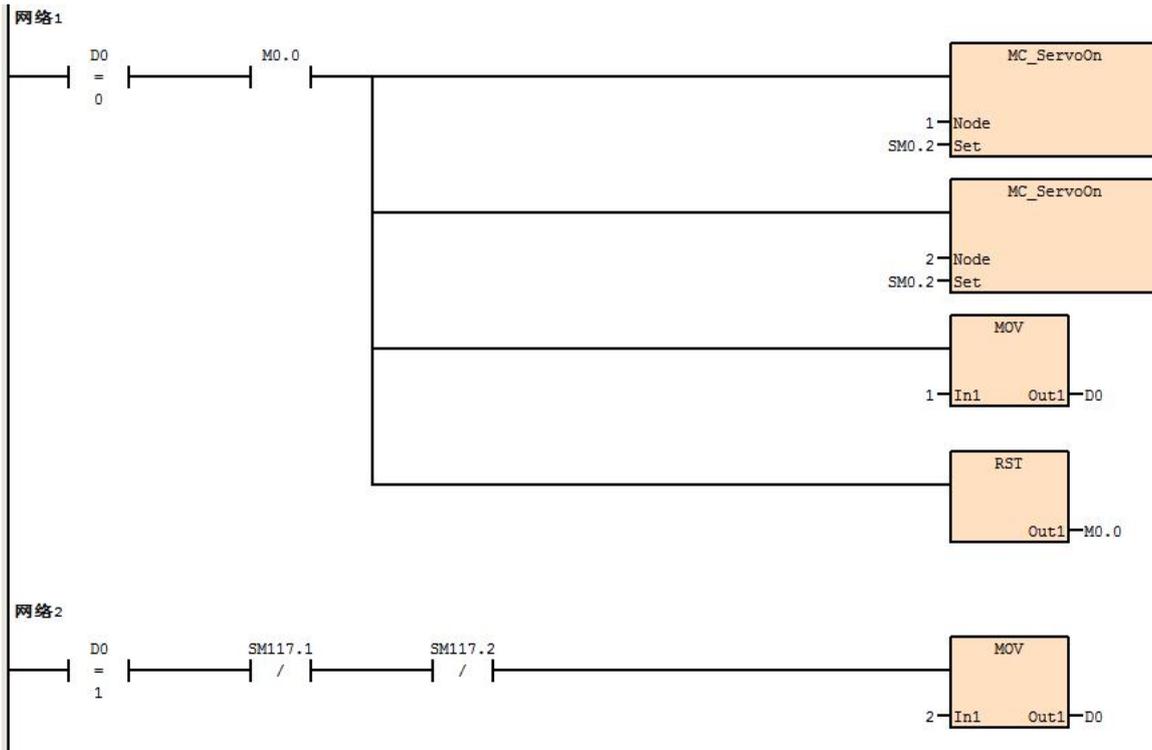
#### 指令功能说明:

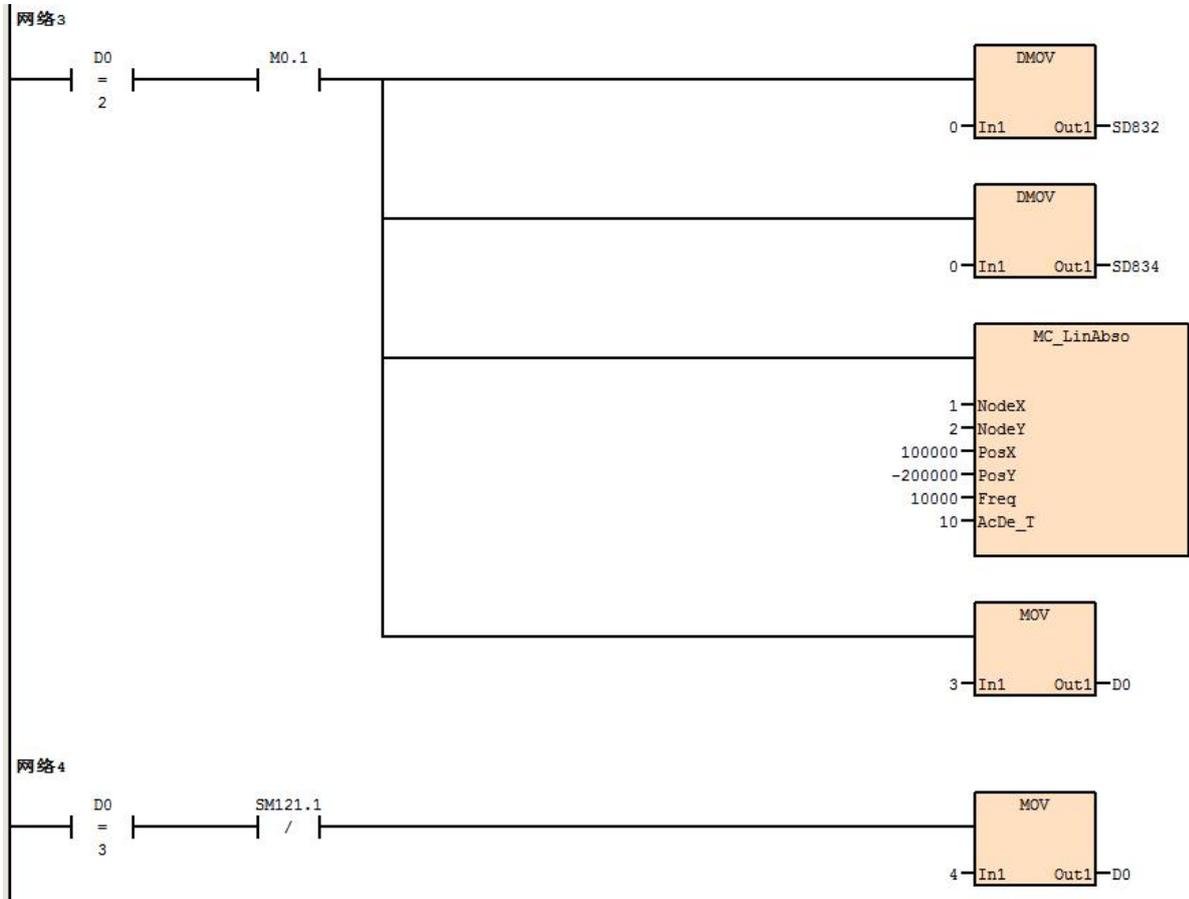
- 1, 该指令的功能是, 控制两轴目标伺服以 X轴目标转速, 进行绝对位置的直线插补;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行, 停止指令的 Node 可以选择两轴中任意一轴;
- 3, 绝对位置的依据 SD832~SD891 对应位移, 例如站号 1 的伺服当前绝对位置是 SD832~SD833 组成, 当前数值是 800, 那么填入目标绝对位置为 0 时, 指令执行时, 即伺服向反方向运转 800 个脉冲位置。

#### 程序范例:

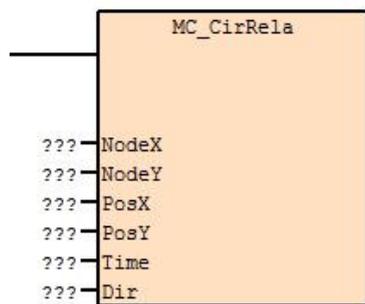
例：

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF，置位 M0.0 为 ON，即对站号 1 和站号 2 的伺服发送“伺服使能”命令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“伺服使能”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，当站号 1 和站号 2 的运动控制标志位 **SM121.1 和 SM121.2** 均为 OFF，置位 M0.1 为 ON，先将伺服 1 和伺服 2 的当前绝对位置都设置为 0（使用将数值 0 搬运到 SD832~SD833 以及 SD834~SD835 实现），那么 MC\_LinAbso 指令中设置的运转目标位置，即是对应当前位置 0 的数值；之后伺服 1 和伺服 2 开始同步运转，其中伺服 2 反转 200000 脉冲大于伺服 1 的正转 100000 脉冲，那么伺服 2 以 1 秒时间加速到最大频率 10000Hz，伺服 1 以 1 秒时间加速到 5000Hz，做直线插补运动；伺服运转过程中运动控制标志位 **SM121.1 和 SM121.2** 置 ON；进入步骤 3；
- 在步骤 3 中，等待运动控制标志 **SM121.1 和 SM121.2** 置 OFF（选一个即可，指令执行完成两个会同时 OFF），进入步骤 4。





### 10.3.14 MC\_CirRela (双轴相对圆弧插补)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
NodeX	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
NodeY	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
PosX	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
PosY	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Time	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S	M	T	C	H	L	EL							
Dir	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, NodeX: X 轴目标伺服站号, 范围 1~30;
- 2, NodeY: Y 轴目标伺服站号, 范围 1~30;
- 3, PosX: X 轴伺服相对当前位置位移的目标脉冲个数; 数值为正, 伺服正转; 数值为负, 伺服反转;
- 4, PosY: Y 轴伺服相对当前位置位移的目标脉冲个数; 数值为正, 伺服正转; 数值为负, 伺服反转;
- 5, Time: 完成圆弧运转所需要的时间, 时间单位: 0.1s, 设定范围: 1~6000;
- 6, Dir: 圆弧运转方向, 设定为 OFF, 顺时针画圆弧; 设定为 ON, 逆时针画圆弧。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令。

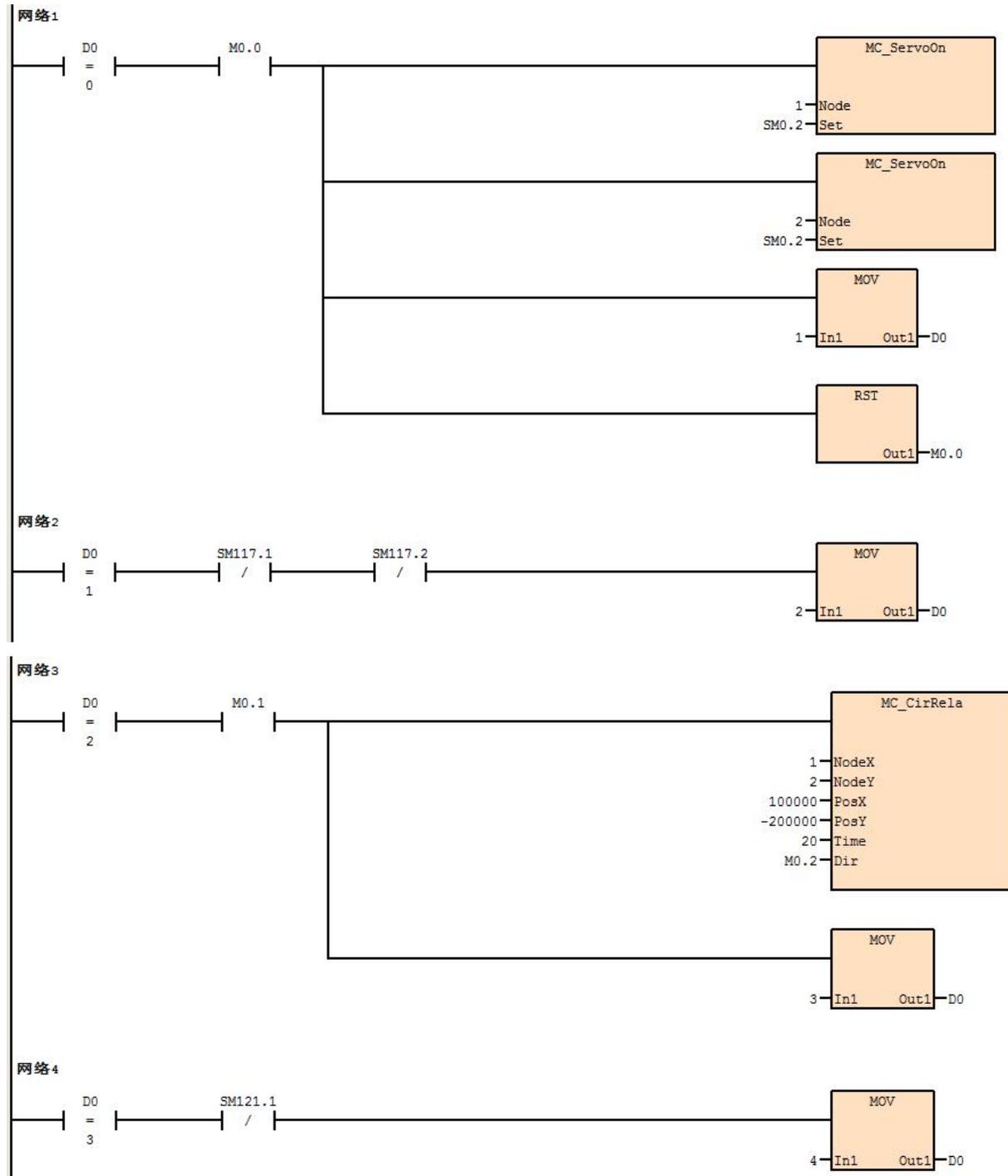
**指令功能说明:**

- 1, 该指令的功能是, 控制两轴目标伺服以在设定时间内, 画出设定方向的相对位置的 90 度圆弧, 该圆弧可以不是正圆弧;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行, 停止指令的 Node 可以选择两轴中任意一轴;
- 3, 圆弧插补指令, 没有加减速设定, 指令一旦执行, 会自动按照设定时间分配圆弧每一段伺服的运转转速。

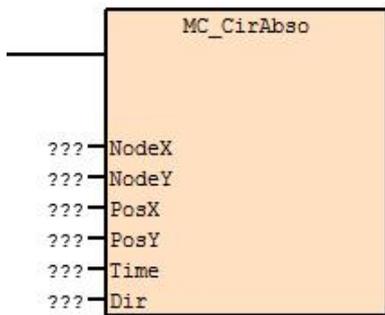
**程序范例:**

例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF, 置位 M0.0 为 ON, 即对站号 1 和站号 2 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服使能”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 2;
- 在步骤 2 中, 当站号 1 和站号 2 的运动控制标志位 SM121.1 和 SM121.2 均为 OFF, 置位 M0.1 为 ON, 伺服 1 和伺服 2 开始同步运转, 根据 M0.2 为 OFF, 根据 Time 参数 20 (对应时间 2 秒), 顺时针画出圆弧, 伺服 1 的正转 100000 脉冲, 伺服 2 反转 200000 脉冲; 伺服运转过程中运动控制标志位 SM121.1 和 SM121.2 置 ON; 进入步骤 3;
- 在步骤 3 中, 等待运动控制标志 SM121.1 和 SM121.2 置 OFF (选一个即可, 指令执行完成会同时 OFF), 进入步骤 4。



### 10.3.15 MC\_CirAbso (双轴绝对圆弧插补)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
NodeX	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
NodeY	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
PosX	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
PosY	DWord	Input	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Time	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S	M	T	C	H	L	EL							
Dir	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, NodeX: X 轴目标伺服站号, 范围 1~30;
- 2, NodeY: Y 轴目标伺服站号, 范围 1~30;
- 3, PosX: X 轴伺服位移的目标绝对位置, 该轴的绝对位置是 32 位数据, 基于 SD832~SD891 位移;
- 4, PosY: Y 轴伺服位移的目标绝对位置, 该轴的绝对位置是 32 位数据, 基于 SD832~SD891 位移;
- 5, Time: 完成圆弧运转所需要的时间, 时间单位: 0.1s, 设定范围: 1~6000;
- 6, Dir: 圆弧运转方向, 设定为 OFF, 顺时针画圆弧; 设定为 ON, 逆时针画圆弧。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的**运动控制标志位 SM** (站号 1~30 分别对应 SM121.1~SM124.6), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个运动控制指令, 必须等待其他运动控制指令完成后, 使用者才可以置起能流上升沿去触发执行运动控制指令。

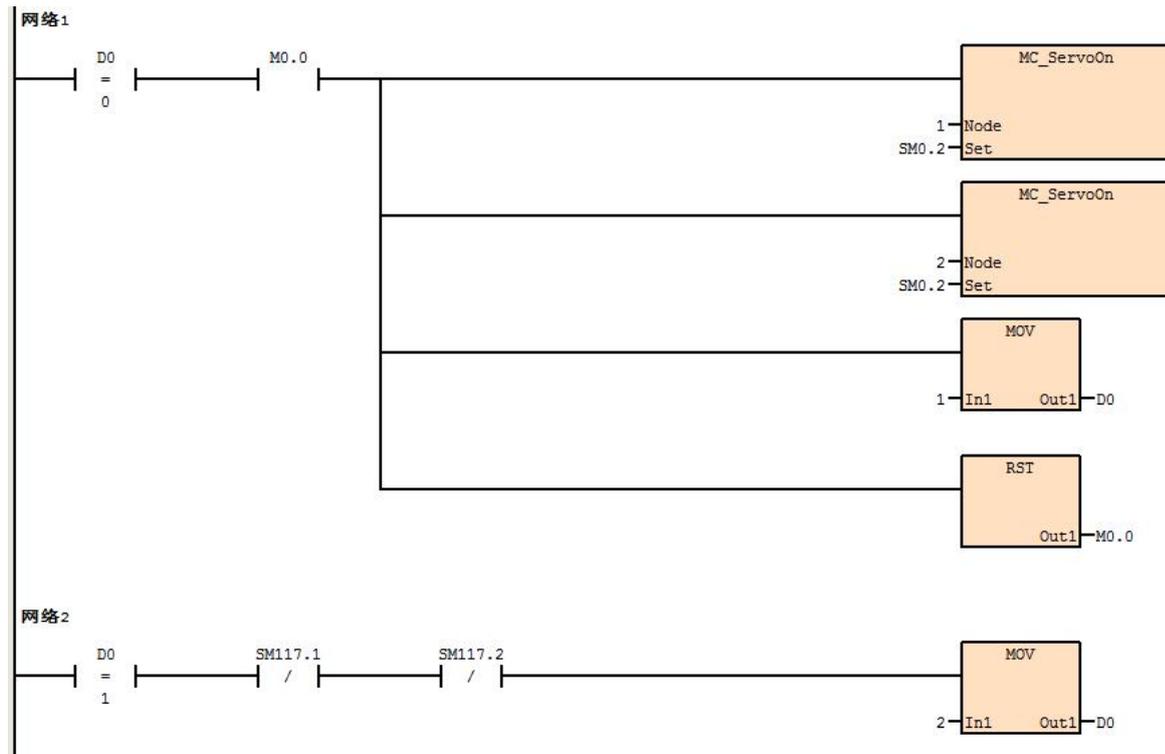
**指令功能说明:**

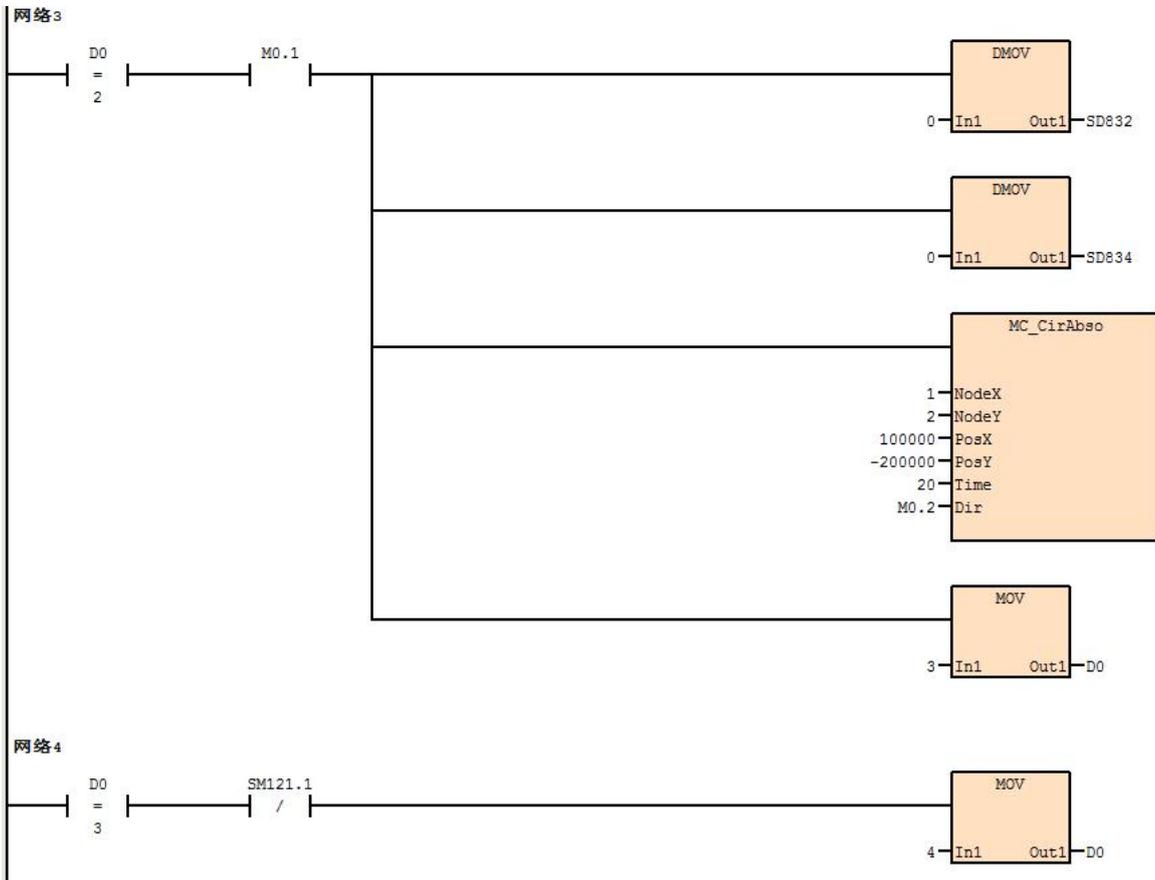
- 1, 该指令的功能是, 控制两轴目标伺服以在设定时间内, 画出设定方向的绝对位置的 90 度圆弧, 该圆弧可以不是正圆弧;
- 2, 伺服运转后, 可以使用“停止指令”让该指令停止执行, 停止指令的 Node 可以选择两轴中任意一轴;
- 3, 圆弧插补指令, 没有加减速设定, 指令一旦执行, 会自动按照设定时间分配圆弧每一段伺服的运转转速;
- 4, 绝对位置的依据 SD832~SD891 对应位移, 例如站号 1 的伺服当前绝对位置是 SD832~SD833 组成, 当前数值是 800, 那么填入目标绝对位置为 0 时, 指令执行时, 即伺服向反方向运转 800 个脉冲位置。

**程序范例:**

例:

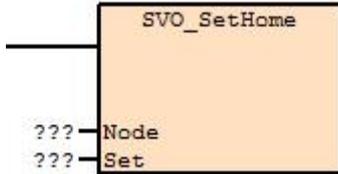
- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的忙碌标志位 SM117.1 和 SM117.2 均为 OFF, 置位 M0.0 为 ON, 即对站号 1 和站号 2 的伺服发送“伺服使能”命令, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬运到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都完成“伺服使能”命令, 即 SM117.1 和 SM117.2 都是 OFF 时, 进入步骤 2;
- 在步骤 2 中, 当站号 1 和站号 2 的运动控制标志位 **SM121.1 和 SM121.2** 均为 OFF, 置位 M0.1 为 ON, 先将伺服 1 和伺服 2 的已发送脉冲绝对位置都设置为 0 (使用将数值 0 搬运到 SD832~SD833 以及 SD834~SD835 实现), 那么 MC\_CirAbso 指令中设置的运转目标位置, 即是对应当前位置 0 的数值; 伺服 1 和伺服 2 开始同步运转, 根据 M0.2 为 OFF, 根据 Time 参数 20 (对应位时间 2 秒), 顺时针画出圆弧, 伺服 1 的正转 100000 脉冲, 伺服 2 反转 200000 脉冲; 伺服运转过程中运动控制标志位 **SM121.1 和 SM121.2** 置 ON; 进入步骤 3;
- 在步骤 3 中, 等待运动控制标志 **SM121.1 和 SM121.2** 置 OFF (选一个即可, 指令执行完成两个会同时 OFF), 进入步骤 4。





## 10.4 伺服及手轮通用控制指令

### 10.4.1 SVO\_SetHome (伺服原点位置设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

#### 参数说明:

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 设置选择; ON: 设置当前位置为伺服机械原点位置; OFF: 取消已设置的伺服机械原点位置。

#### 指令执行注意点:

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

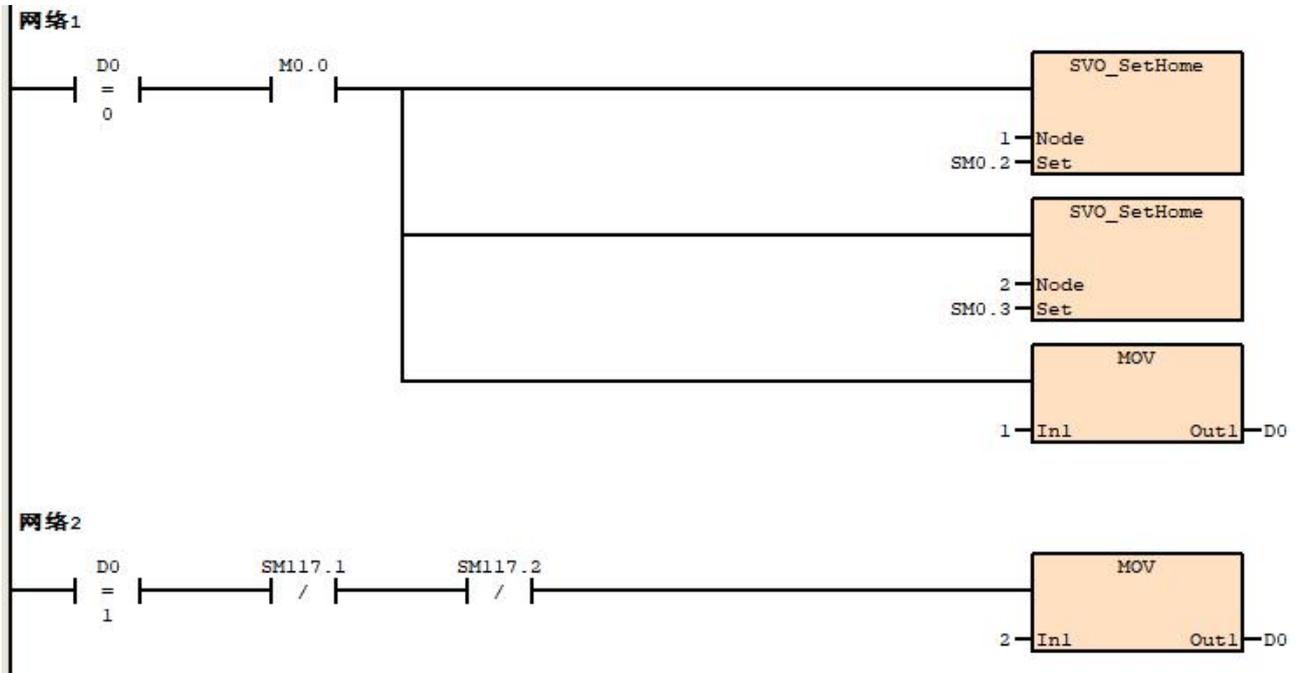
#### 指令功能说明:

- 1, 该指令的功能是, 设定当前位置为伺服的机械原点位置;
- 2, 使用该指令的同时, 会让对应轴的“伺服当前的机械实际绝对位置 SD602~SD661”以及“伺服当前的程序绝对位置 SD832~SD891”复位为 0。

#### 程序范例:

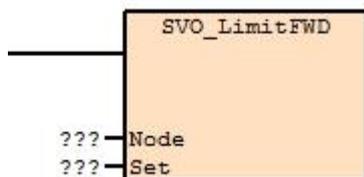
##### 例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的从站都空闲时, 即 SM117.1 和 SM117.2 均为 OFF, (SM117.1 对应站号 1 的忙碌标志, SM117.2 对应站号 2 的忙碌标志), 置位 M0.0 为 ON, 即对站号 1 设置当前位置为伺服机械原点位置, 对站号 2 取消已设置的伺服机械原点位置, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都指令执行完成, 即 SM117.1 和 SM117.2 都是 OFF 时, 将数值 2 搬移到 D0 中, 进入步骤 2。





## 10.4.2 SVO\_LimitFWD (伺服正转极限位置设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

### 参数说明:

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 设置选择; ON: 设置当前位置为正转极限位置; OFF: 取消已设置的正转极限位置。

### 指令执行注意点:

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

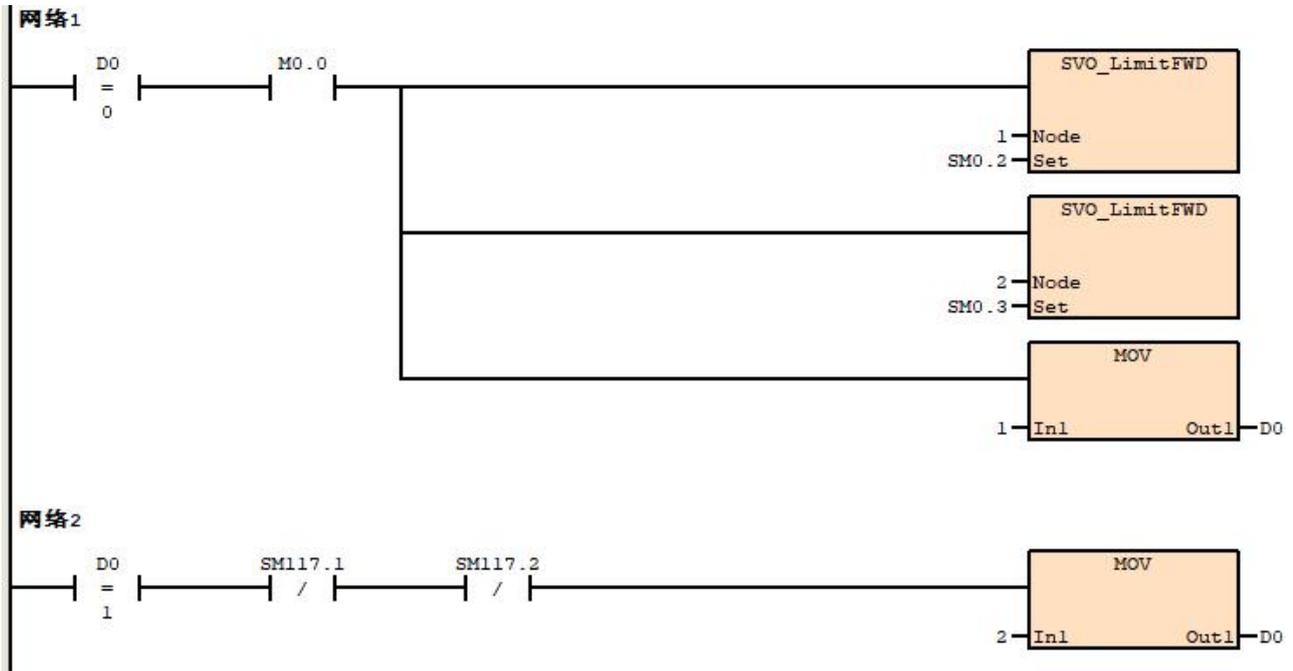
### 指令功能说明:

- 1, 该指令的功能是, 设定当前位置为伺服的正转极限位置 (硬极限)。

### 程序范例:

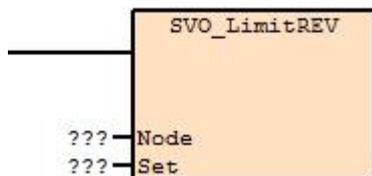
#### 例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的从站都空闲时, 即 SM117.1 和 SM117.2 均为 OFF, (SM117.1 对应站号 1 的忙碌标志, SM117.2 对应站号 2 的忙碌标志), 置位 M0.0 为 ON, 即对站号 1 设置当前位置为正转极限位置, 对站号 2 取消已设置的正转极限位置, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都指令执行完成, 即 SM117.1 和 SM117.2 都是 OFF 时, 将数值 2 搬移到 D0 中, 进入步骤 2。





### 10.4.3 SVO\_LimitREV (伺服反转极限位置设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

#### 参数说明:

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 设置选择; ON: 设置当前位置为反转极限位置; OFF: 取消已设置的反转极限位置。

#### 指令执行注意点:

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

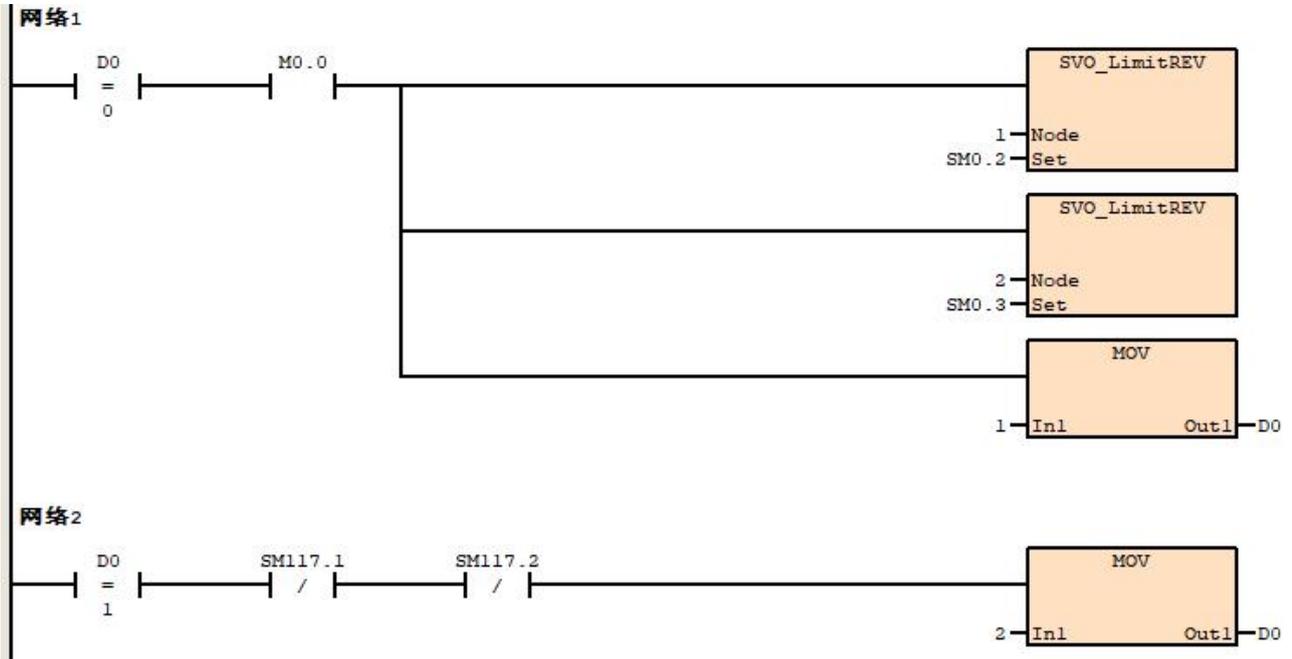
#### 指令功能说明:

- 1, 该指令的功能是, 设定当前位置为伺服的反转极限位置 (硬极限)。

#### 程序范例:

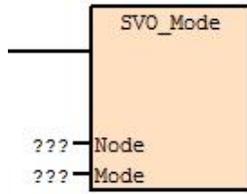
##### 例:

- 在例程中, D0 用于表示程序步骤, 步骤 0 (即 D0=0) 当站号 1 和站号 2 的从站都空闲时, 即 SM117.1 和 SM117.2 均为 OFF, (SM117.1 对应站号 1 的忙碌标志, SM117.2 对应站号 2 的忙碌标志), 置位 M0.0 为 ON, 即对站号 1 设置当前位置为反转极限位置, 对站号 2 取消已设置的反转极限位置, 发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON; 同时将数值 1 搬移到 D0 装置中, 表明进入步骤 1;
- 在步骤 1 中, 等待站号 1 和站号 2 都指令执行完成, 即 SM117.1 和 SM117.2 都是 OFF 时, 将数值 2 搬移到 D0 中, 进入步骤 2。





### 10.4.4 SVO\_Mode (伺服运转模式设置)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Mode	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Mode: 伺服运转模式设置, 0: 位置模式 (默认);  
 1: 速度模式;  
 2: 转矩模式;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

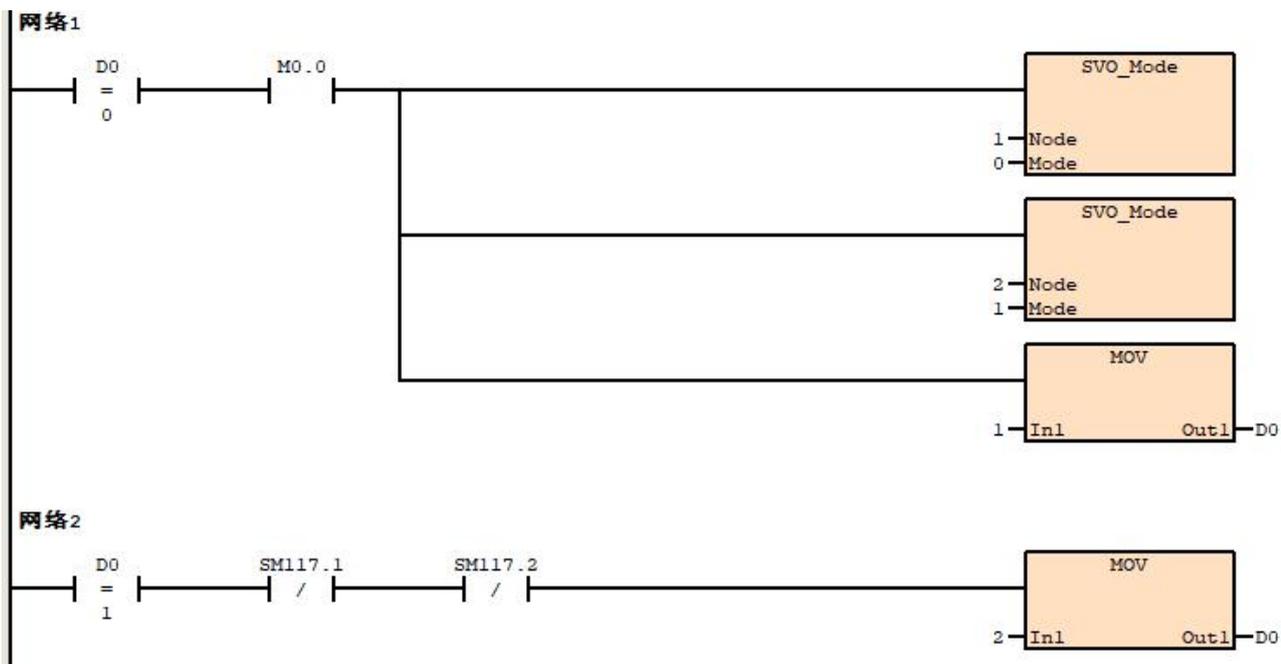
- 1, 该指令的功能是, 设置伺服的运转模式, 在伺服没有运转时可以设置;
- 2, 伺服当前所处的运转模式, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

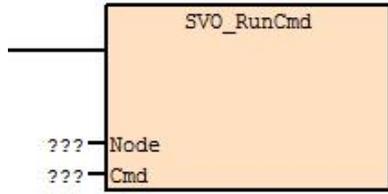
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1“设置运转模式”为“位置模式”，对站号 2“设置运转模式”为“速度模式”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“设置运转模式”命令，即 SM117.1 和 SM117.2 都是 OFF 时，将数值 2 搬移到 D0 中，进入步骤 2。





### 10.4.5 SVO\_RunCmd (伺服运转命令)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Cmd	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Cmd: 伺服速度模式和转矩模式下的运转命令, 0: 停止;  
 1: 正转;  
 2: 反转;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

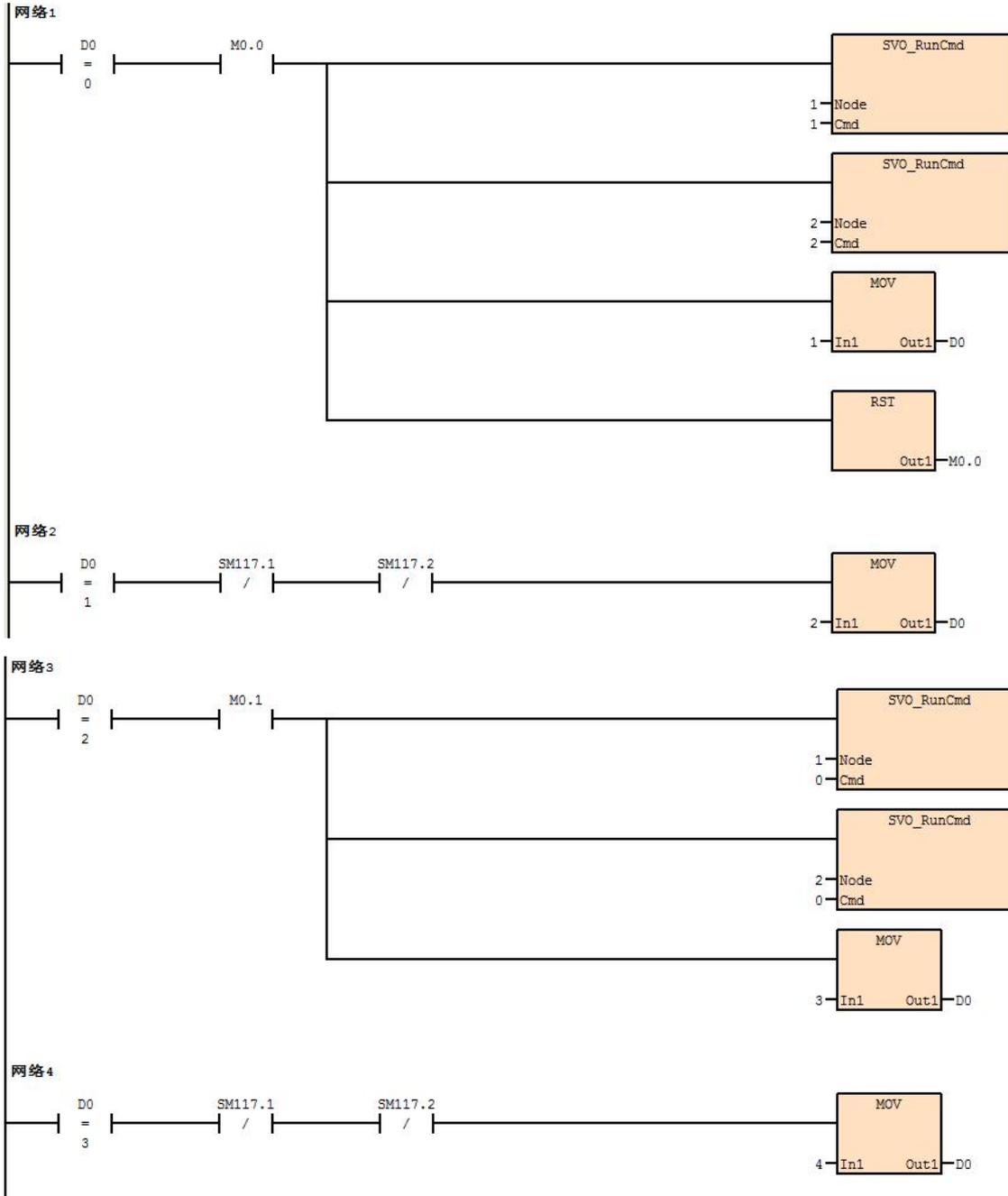
- 1, 该指令的功能是, 当伺服处于速度模式或转矩模式时, 控制伺服的运转;
- 2, 伺服当前所处的运转状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

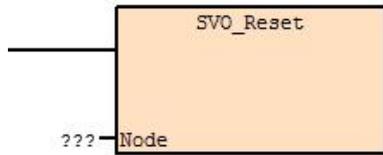
例:

- 在例程中，伺服 1 处于“速度模式”，伺服 2 处于“转矩模式”；D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1“运转命令”设置为“正转”，对站号 2“运转命令”为“反转”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“设置运转”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，置位 M0.0 为 ON，即对站号 1 和站号 2 的“运转命令”都设置为“停止”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；进入步骤 3；
- 在步骤 3 中，等待站号 1 和站号 2 都完成“设置运转停止”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 4。





### 10.4.6 SVO\_Reset (伺服故障复位)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

1, Node: 目标伺服站号, 范围 1~30;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

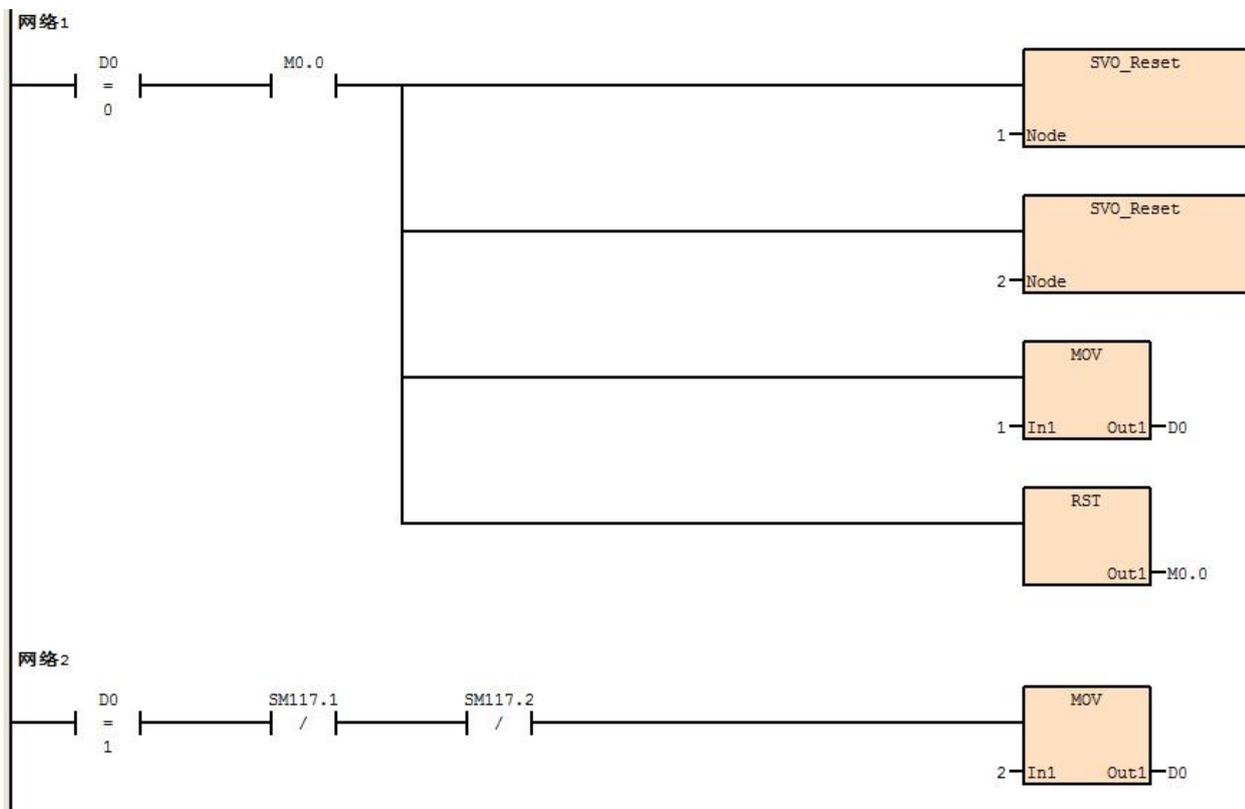
- 1, 该指令的功能是, 当伺服处于故障状态时, 使用该指令可将故障状态清除;
- 2, 伺服当前所处的运转状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

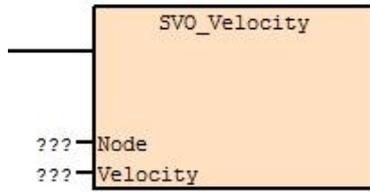
例：

- 在例程中，伺服 1 和伺服 2 处于“报错状态”中；D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1 和站号 2 发送“清除故障状态”指令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“清除故障状态”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.4.7 SVO\_Velocity (伺服速度模式转速值设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Velocity	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Velocity: 伺服运转的目标转速, 范围 0~4500, 单位: r/min (即直接填入电机目标转速: 转/分钟);

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

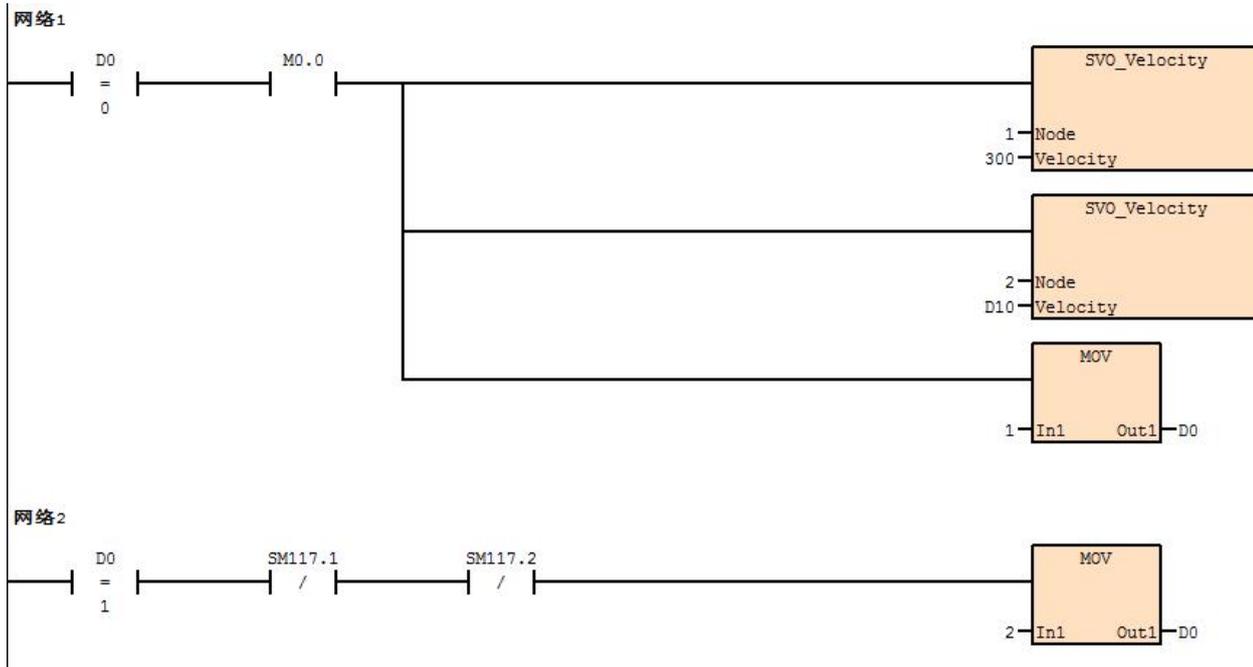
- 1, 该指令是当伺服处于速度模式时, 设定伺服的运转目标转速, 伺服将以速度模式菜单中设置的加减速时间进行运转;
- 2, 伺服当前转速, 最高转速, 以及转矩限制值, 是否锁轴, 均可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

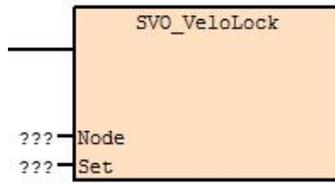
例:

- 在例程中，伺服 1 和伺服 2 都处在“速度模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对伺服 1 设置“运转转速”为 300 r/min，对伺服 2 设置“运转转速”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“运转转速”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2.





### 10.4.8 SVO\_VeloLock (伺服速度模式下锁轴设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 运转停止时锁轴是否生效, ON 为锁轴, OFF 为不锁轴 (默认);

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

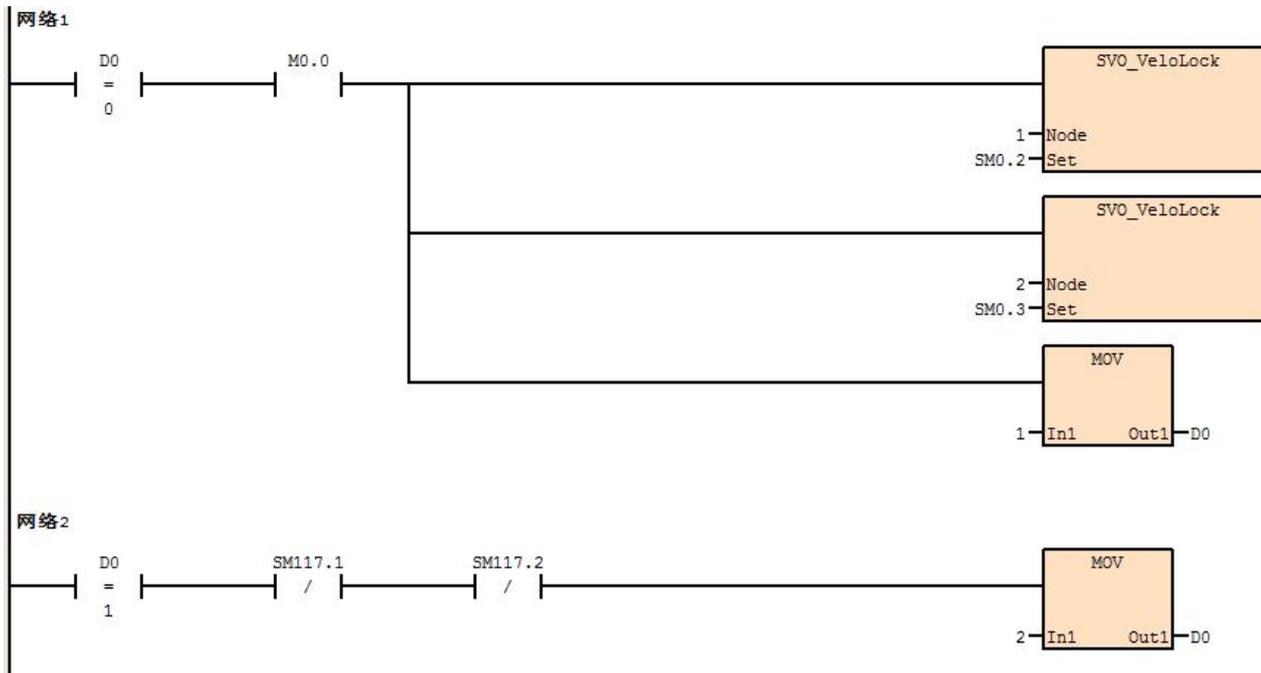
- 1, 该指令的功能是, 当伺服处于速度模式时, 进行锁轴控制;
- 2, 伺服当前是否锁轴, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

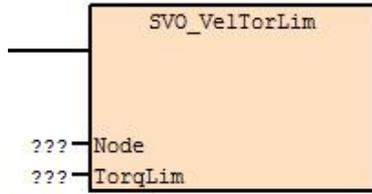
例:

- 在例程中，伺服 1 和伺服 2 都处在“速度模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON,即对伺服 1 设置“锁轴设定”为生效(SM0.2 为常 ON 标志位),对伺服 2 设置“锁轴设定”为不生效(SM0.3 为常 OFF 标志位)，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“锁轴设定”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2.





### 10.4.9 SVO\_VelTorLim (伺服速度模式下转矩限制值设定)



变量名	数据类型	输入输出	字/双字装置																
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E	
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
TorqLim	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, TorqLim: 转矩限制值设定, 设定范围 10~400, 单位: %, 设备默认设定是 300%;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

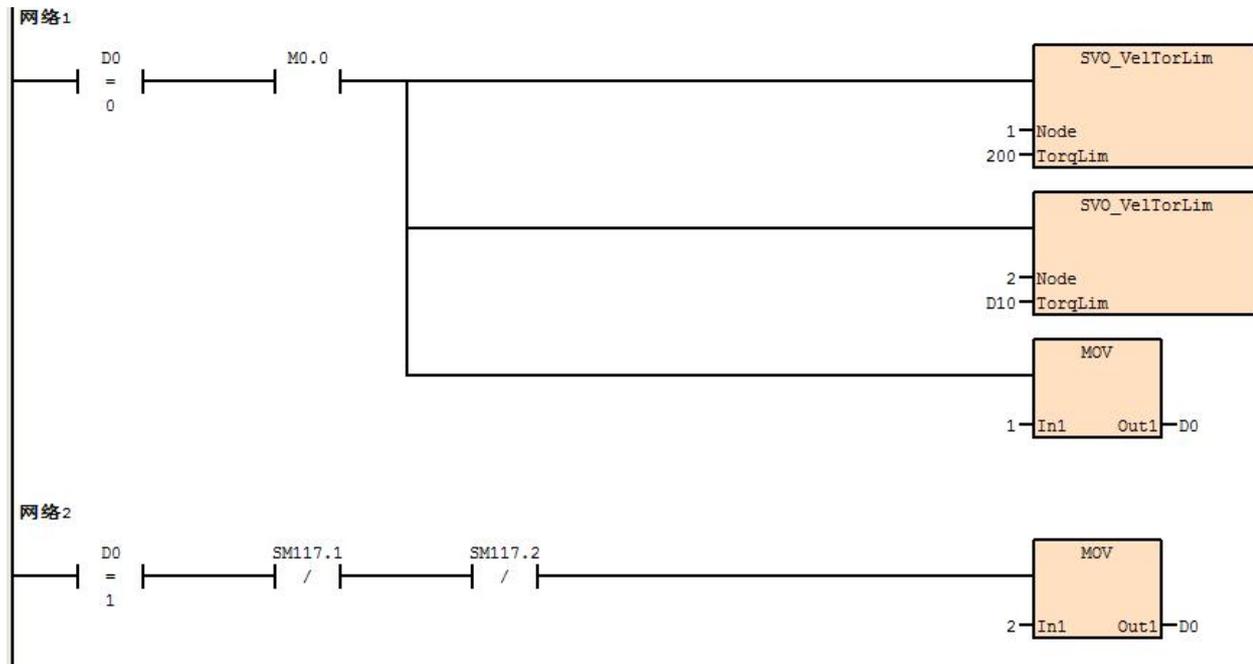
- 1, 该指令的功能是, 当伺服处于速度模式时, 进行转矩限制值设定;
- 2, 伺服运转时是否达到转矩限制值, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

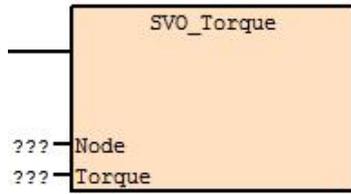
例：

- 在例程中，伺服 1 和伺服 2 都处在“速度模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对伺服 1 设置“速度模式下转矩限制”为 200%，对伺服 2 设置“速度模式下转矩限制”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 完成设置“速度模式下转矩限制”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.4.10 SVO\_Torque (伺服转矩模式下转矩值设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Torque	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Torque: 转矩值设定, 设定范围 0~200, 单位: %, 设备默认设定是 50%;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

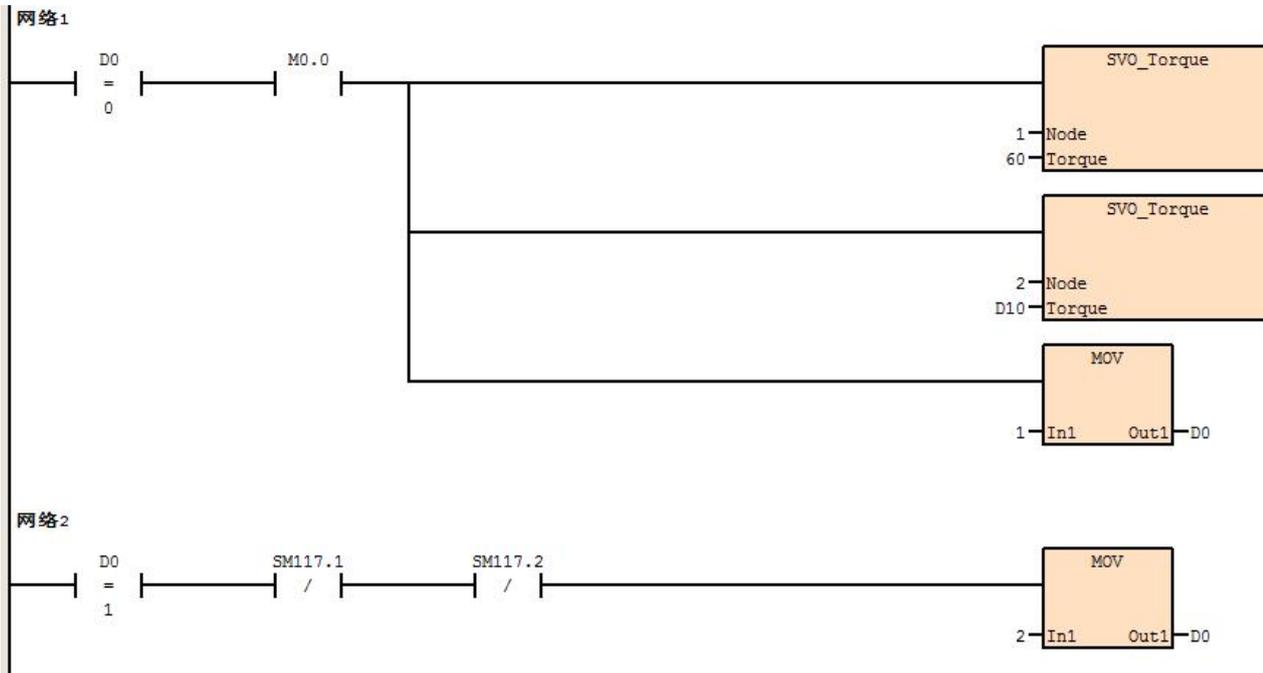
- 1, 该指令的功能是, 当伺服处于转矩模式时, 进行输出转矩值设定;
- 2, 伺服当前的输出转矩值, 最大转矩值, 转速限制值, 均可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

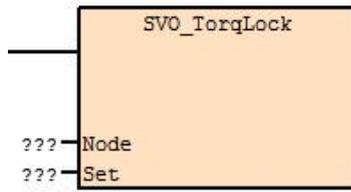
(见下页)

例:

- 在例程中，伺服 1 和伺服 2 都处在“转矩模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对伺服 1 设置“输出转矩”为 60%，对伺服 2 设置“输出转矩”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“输出转矩”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。



### 10.4.11 SVO\_TorqLock (伺服转矩模式下锁轴设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 运转命令停止时锁轴是否生效, ON 为锁轴, OFF 为不锁轴 (默认);

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

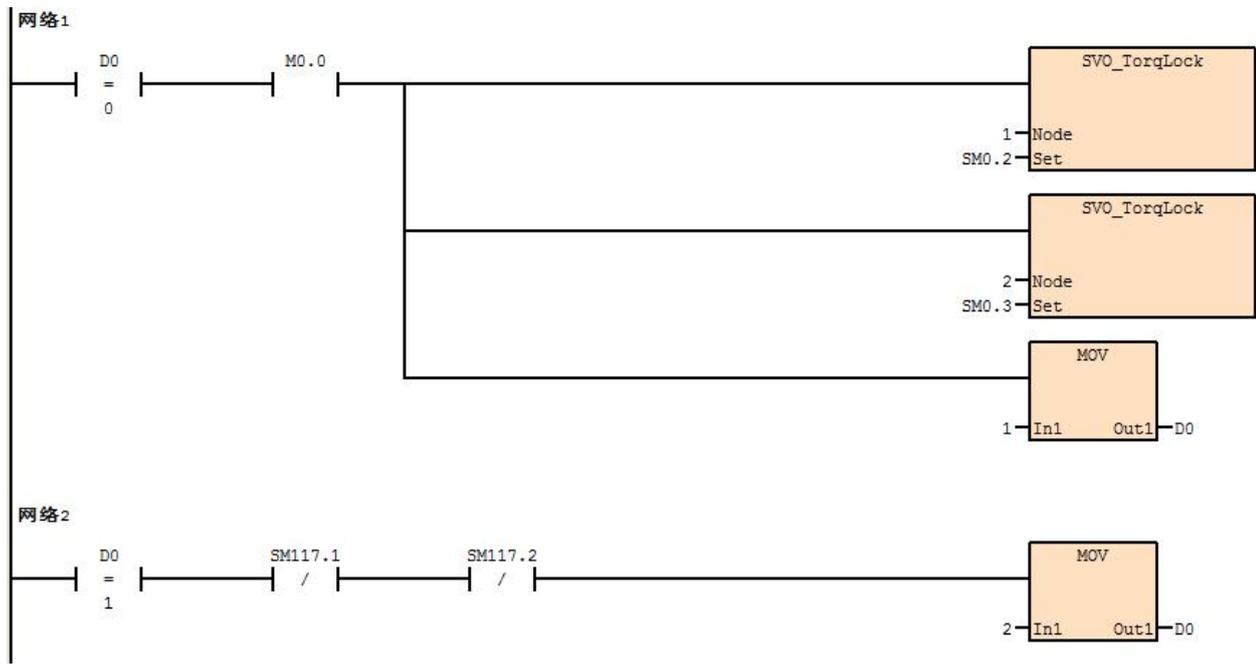
- 1, 该指令的功能是, 当伺服处于转矩模式时, 进行锁轴控制;
- 2, 伺服当前是否锁轴, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

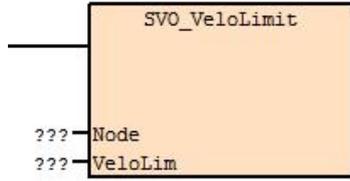
例:

- 在例程中，伺服 1 和伺服 2 都处在“转矩模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON,即对伺服 1 设置“锁轴设定”为生效(SM0.2 为常 ON 标志位),对伺服 2 设置“锁轴设定”为不生效(SM0.3 为常 OFF 标志位)，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“锁轴设定”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.4.12 SVO\_VeloLimit (伺服转矩模式下转速限制值设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
VeloLim	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, VeloLim: 转速限制值设定, 设定范围 10~4500, 单位: r/min, 设备默认设定是额定转速;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

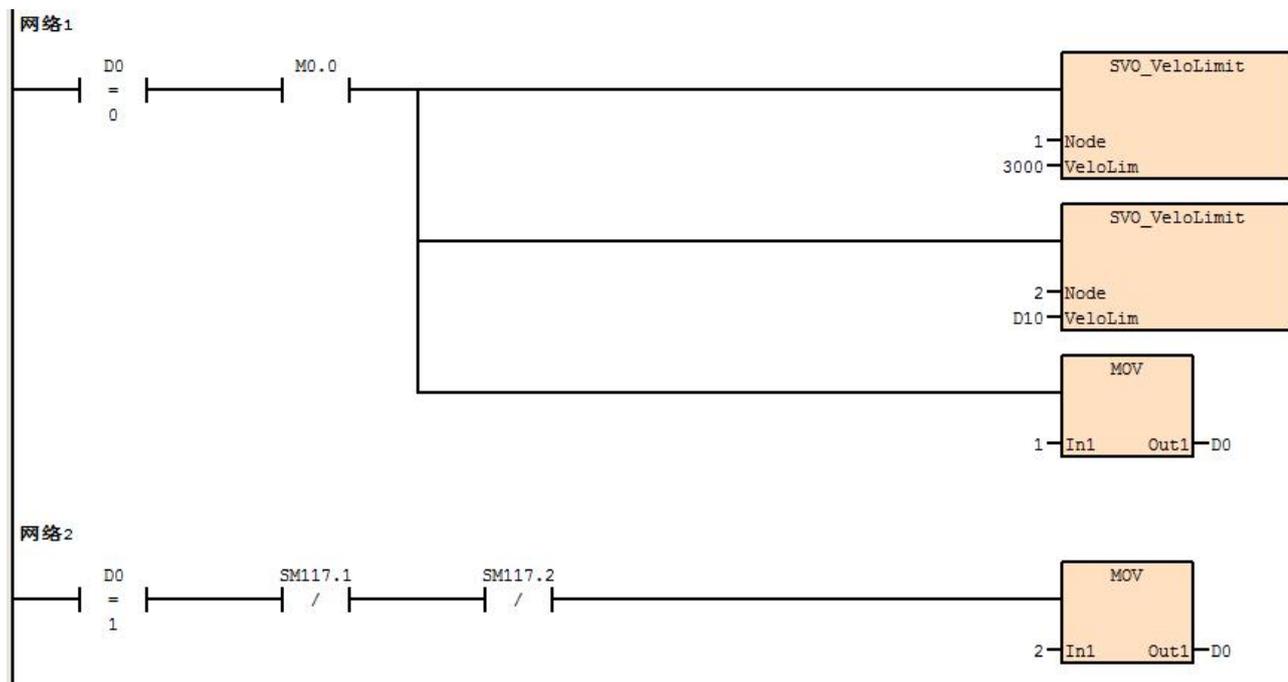
- 1, 该指令的功能是, 当伺服处于转矩模式时, 进行转速限制值设定;
- 2, 伺服是否达到转速限制值, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

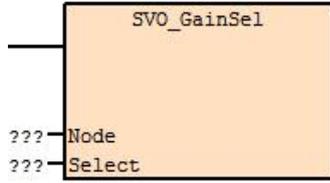
(见下页)

例：

- 在例程中，伺服 1 和伺服 2 都处在“转矩模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对伺服 1 设置“转矩模式下转速限制”为 3000 r/min，对伺服 2 设置“转矩模式下转速限制”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 完成设置“转矩模式下转速限制”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。



### 10.4.13 SVO\_GainSel (伺服位置模式下增益组选择)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Select	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Select: 增益组选择, 可以根据设定的增益组, 选择使用, 0: 第一增益组; 1: 第二增益组。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

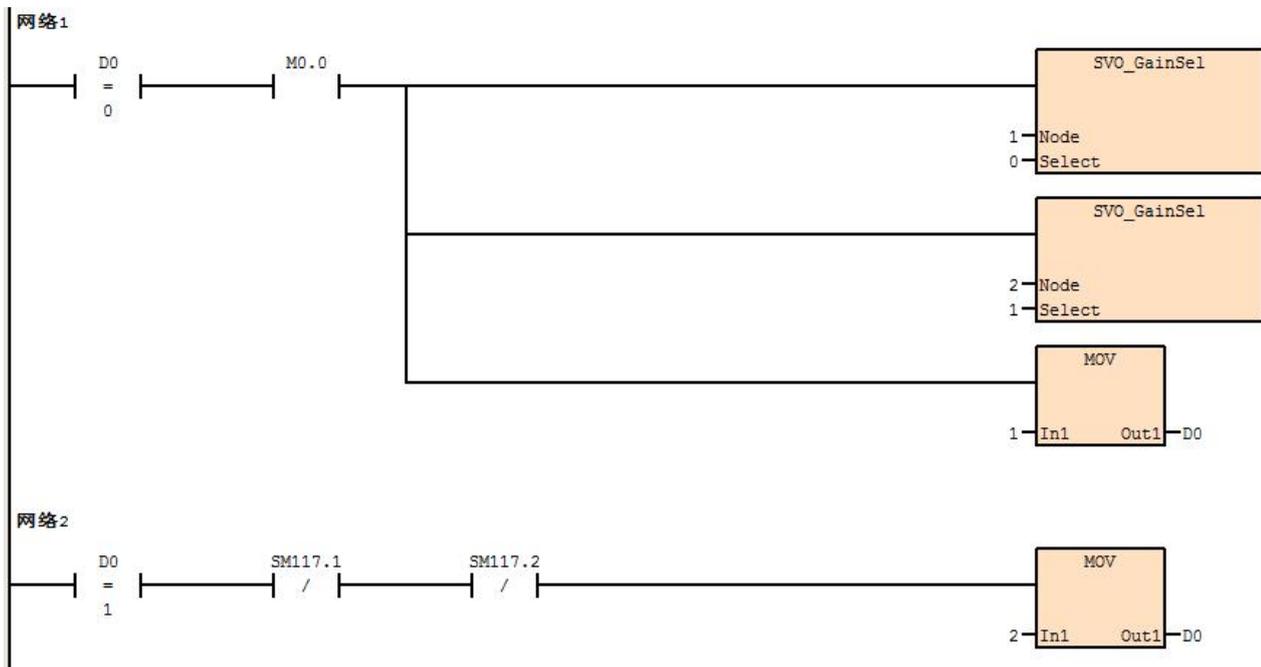
- 1, 该指令的功能是, 当伺服处于位置模式时, 进行增益组的选择使用;
- 2, 伺服当前选择的增益组, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

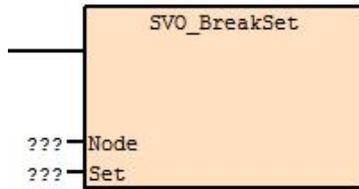
(见下页)

例:

- 在例程中，伺服 1 和伺服 2 都处在“位置模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对伺服 1 设置“增益组选择”为 0：第一增益组，对伺服 2 设置“增益组选择”为 1：第二增益组，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“增益组选择”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。



### 10.4.14 SVO\_BreakSet (伺服位置模式下抱闸选择设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, Set: 该指令功能还取决于伺服菜单 F0-09 的选择:
  - 选择 4, 外部机械抱闸允许选择, OFF: 不允许外部机械抱闸; ON: 允许外部机械抱闸;
  - 选择 5, 停止时失电制动器抱闸, OFF: 不进行抱闸; ON: 失电制动器抱闸。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

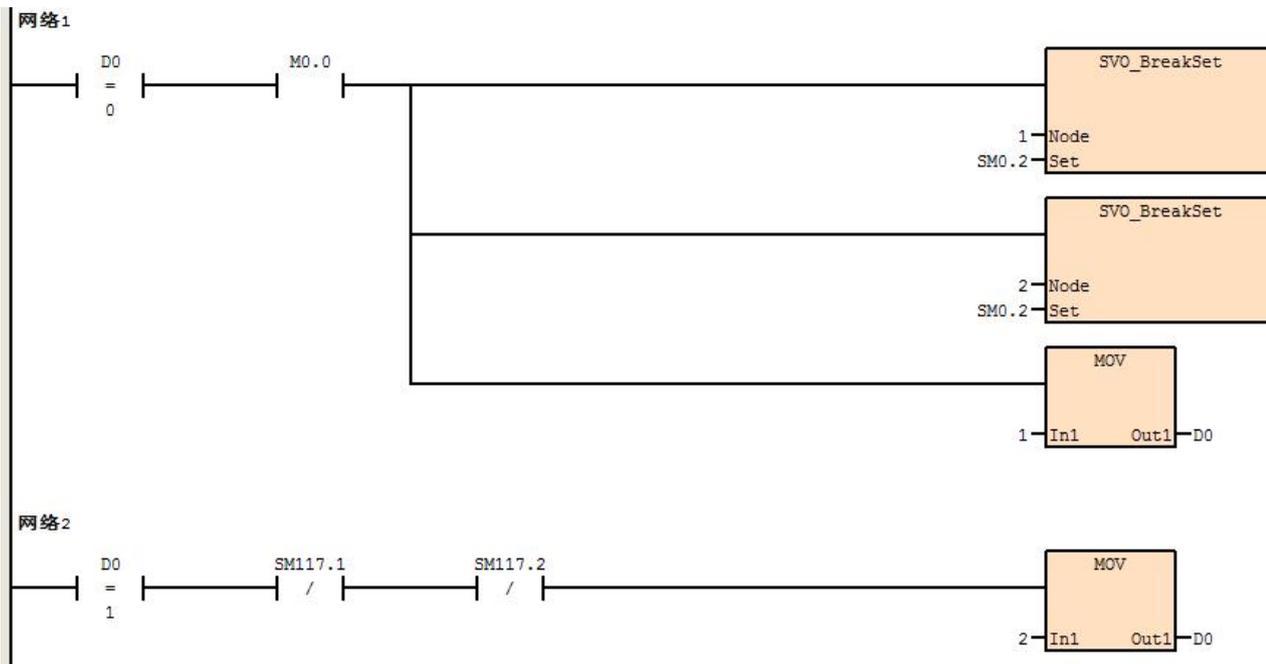
- 1, 该指令的功能是, 当伺服处于位置模式时, 当伺服停止运转时, 是否允许给予外部机械抱闸或失电制动器抱闸设定;
- 2, 伺服当前是否允许外部机械抱闸, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

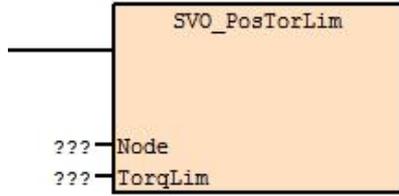
例:

- 在例程中，伺服 1 和伺服 2 都处在“位置模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，伺服 1 是外部机械抱闸，即设置“外部抱闸”为允许（SM0.2 为常 ON 标志位），伺服 2 是失电制动器抱闸，即设置“失电制动器抱闸”为生效（SM0.2 为常 ON 标志位），发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“外部抱闸允许或失电制动器抱闸”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.4.15 SVO\_PosTorLim (伺服位置模式下转矩限制值设定)



变量名	数据类型	输入输出	字/双字装置																
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E	
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	
TorqLim	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■	

**参数说明:**

- 1, Node: 目标伺服站号, 范围 1~30;
- 2, TorqLim: 转矩限制值设定, 设定范围 10~400, 单位: %, 设备默认设定是 300%;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

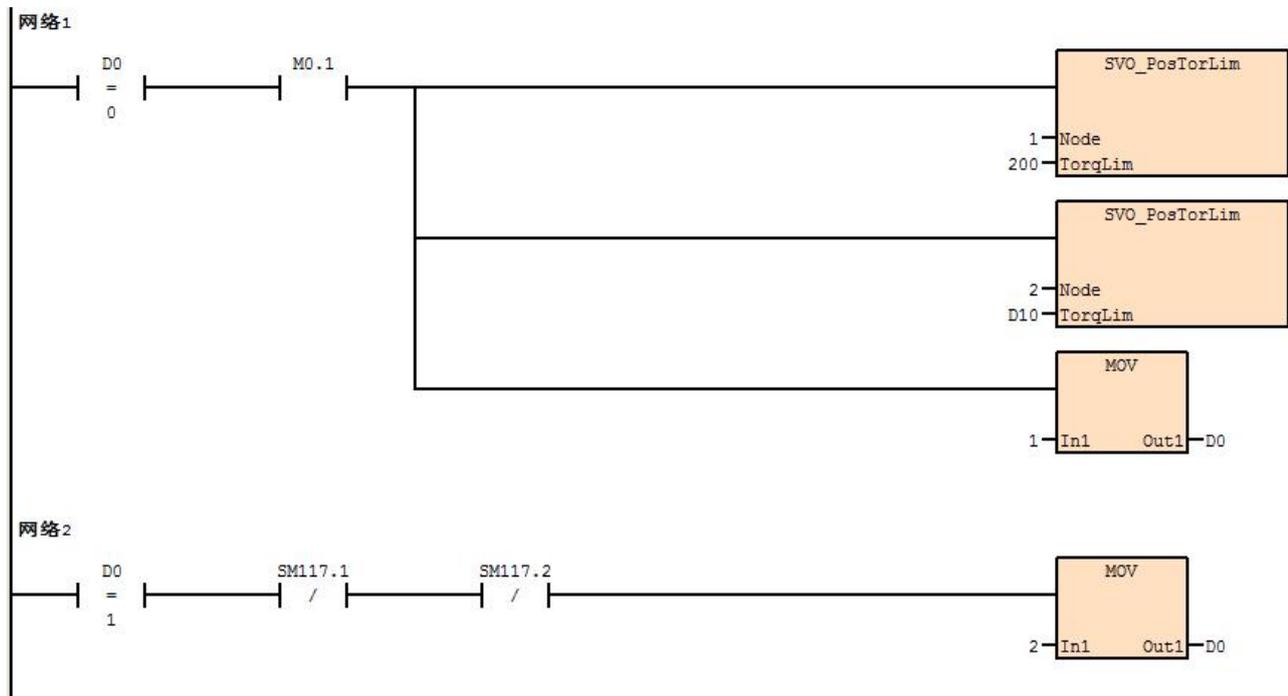
- 1, 该指令的功能是, 当伺服处于位置模式时, 进行转矩限制值设定;
- 2, 伺服运转时是否达到转矩限制值, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

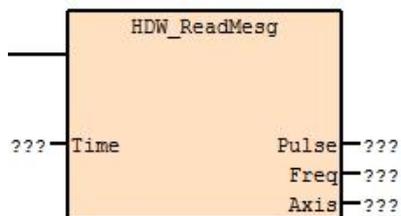
例：

- 在例程中，伺服 1 和伺服 2 都处在“位置模式”中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对伺服 1 设置“位置模式下转矩限制”为 200%，对伺服 2 设置“位置模式下转矩限制”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 完成设置“位置模式下转矩限制”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.4.16 HDW\_ReadMesg (电子手轮位移信息读取)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Time	Word	Input	■	■	■	■	■	□	□	■	■	■	■	■	■	■	■	■
Pulse	DWord	Output		■	■	■	■	□			■	■	■	■				
Freq	DWord	Output		■	■	■	■	□			■	■	■	■				
Axis	Word	Output		■	■	■	■	□			■	■	■	■				

#### 参数说明:

- 1, Time: 读取电子手轮位移信息的间隔时间, 手轮回复的将会是在这段时间内的位移脉冲数, 范围: 1~20, 单位: 0.1s;
- 2, Pulse: 存放脉冲个数装置, 双字数据类型, 间隔时间内, 手轮经过倍频计算后的位移脉冲数, 正数为正转, 负数为反转;
- 3, Freq: 存放脉冲频率装置, 双字数据类型, 间隔时间内, 手轮经过倍频计算后的位移脉冲频率;
- 4, Axis: 当前电子手轮选定的输出轴, 数值为 1~N; (N 为电子手轮最大支持的切换轴数)。

#### 指令执行注意点:

- 1, 该指令是需要能流维持导通执行, 在能流导通期间周期读取, 在能流断开后, 该指令不再执行;
- 2, 使用时请先确认 SM116.3 电子手轮忙碌标志位为 0;
- 3, 在每次更新完成新的信息读取时, SM116.4 电子手轮更新信息标志位会置 ON 一个周期, 可使用该标志位判断电子手轮是否完成新的数据更新;
- 4, 该指令在一个项目中只能编写一次, 否则会出现使用异常。

#### 指令功能说明:

- 1, 该指令的功能, 在用户设定的间隔时间, 周期的读取电子手轮该时间内的位移脉冲数及频率, 正反转方向, 及输出轴选择;
- 2, 电子手轮当前的状态信息, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看;
- 3, 能流维持期间, 可以修改 Time 的数值, 但是在修改后的第一次执行时, 读回的位移脉冲为 0。

#### 程序范例:

(见下页)

例:

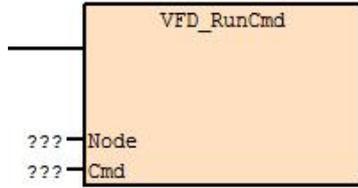
- 在例程中，当 M0.0 置位为 ON，开始每间隔 100ms 将电子手轮的间隔时间内的位移脉冲数，频率以及选定轴分别放于 D0, D2 和 D4 装置中，电子手轮正在使用标志位 SM116.3 置 ON。
- 当 M0.0 置为 OFF，停止采集电子手轮当前的信息，SM116.3 置 OFF。





## 10.5 变频器控制指令

### 10.5.1 VFD\_RunCmd (变频器运转命令)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S	M	T	C	H	A	A	D	S	E	N	O	L	E
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Cmd	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标变频器站号, 范围 1~30;
- 2, Cmd: 变频器的运转命令, 0: 停止;  
1: 正转;  
2: 反转;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

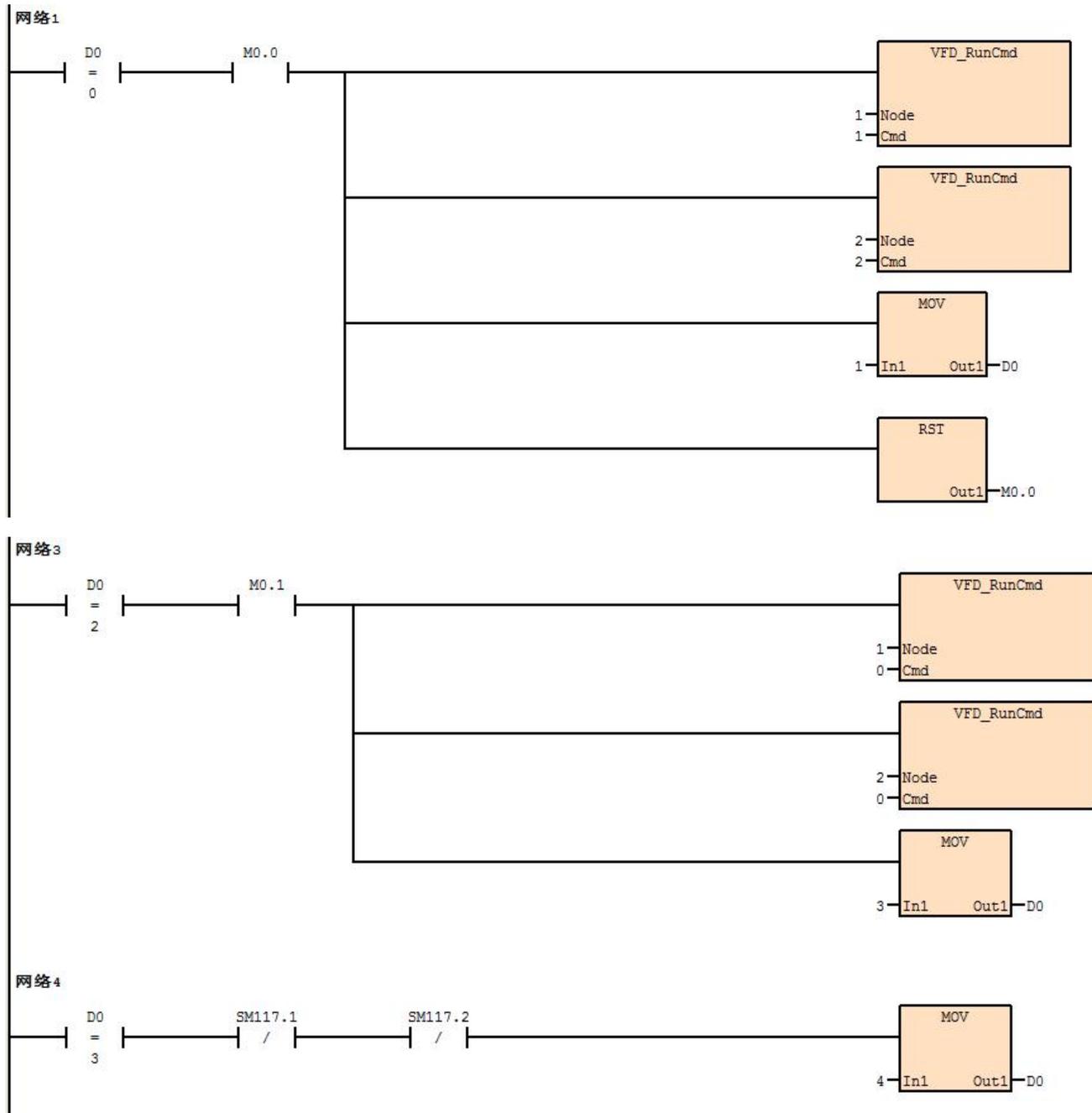
- 1, 该指令的功能是, 控制变频器的运转;
- 2, 变频器当前所处的运转状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

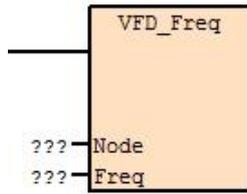
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1“运转命令”设置为“正转”，对站号 2“运转命令”为“反转”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“设置运转”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，置位 M0.0 为 ON，即对站号 1 和站号 2 的“运转命令”都设置为“停止”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；进入步骤 3；
- 在步骤 3 中，等待站号 1 和站号 2 都完成“设置运转停止”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 4。





### 10.5.2 VFD\_Freq (变频器运转频率设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Freq	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标变频器站号, 范围 1~30;
- 2, Freq: 变频器运转的目标频率, 范围 0~9999, 单位: 0.1Hz;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

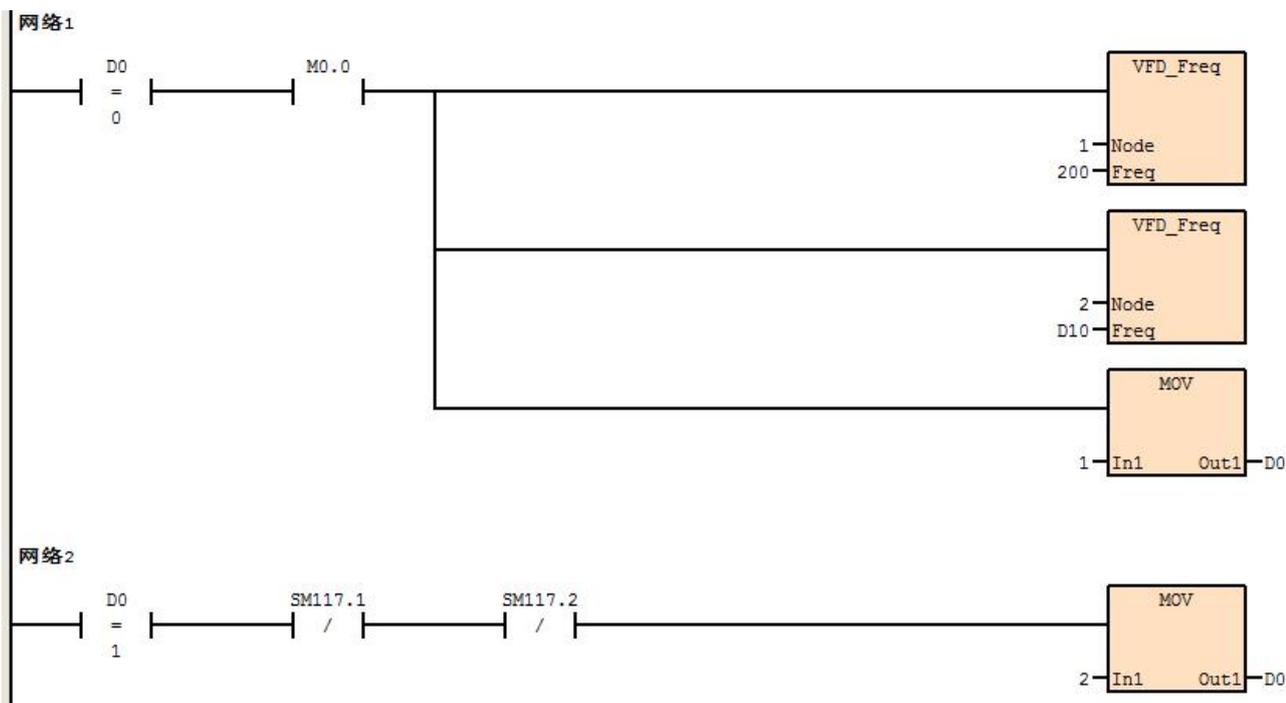
- 1, 该指令的功能是, 设定变频器的运转目标频率, 变频器将以菜单中设置的加减速时间进行运转;
- 2, 变频器当前输出频率可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

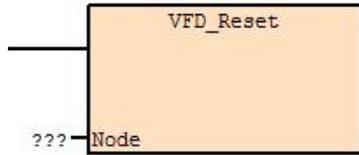
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1“设置运转频率”为 20.0Hz，对站号 2“设置运转频率”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“设置运转频率”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.5.3 VFD\_Reset (变频器故障复位)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

1, Node: 目标变频器站号, 范围 1~30;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

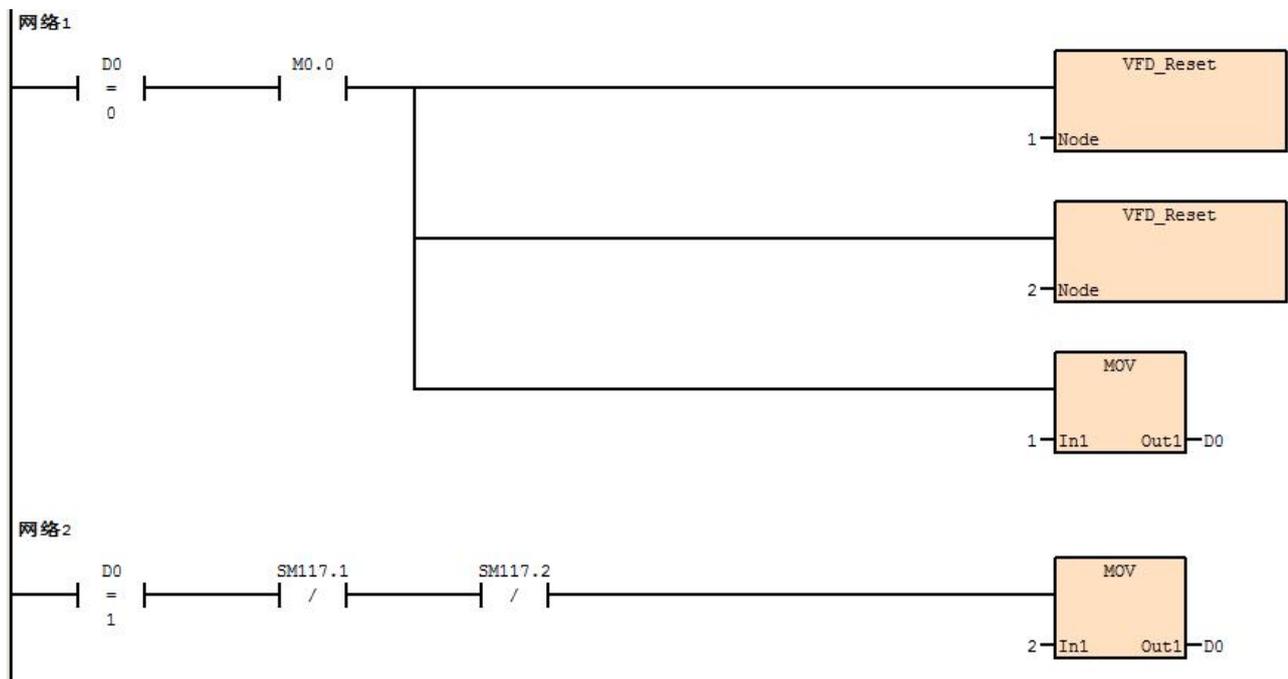
- 1, 该指令的功能是, 当变频器处于故障状态时, 使用该指令可将故障状态清除;
- 2, 变频器当前所处的运转状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

例:

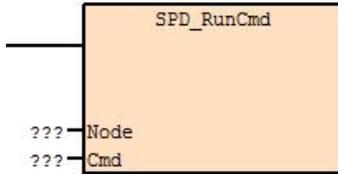
- 在例程中，变频器 1 和变频器 2 处于“报错状态”中；D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1 和站号 2 发送“清除故障状态”指令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“清除故障状态”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





## 10.6 调速器控制指令

### 10.6.1 SPD\_RunCmd (调速器运转命令)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Cmd	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标调速器站号, 范围 1~30;
- 2, Cmd: 调速器的运转命令, 0: 停止;  
1: 正转;  
2: 反转;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

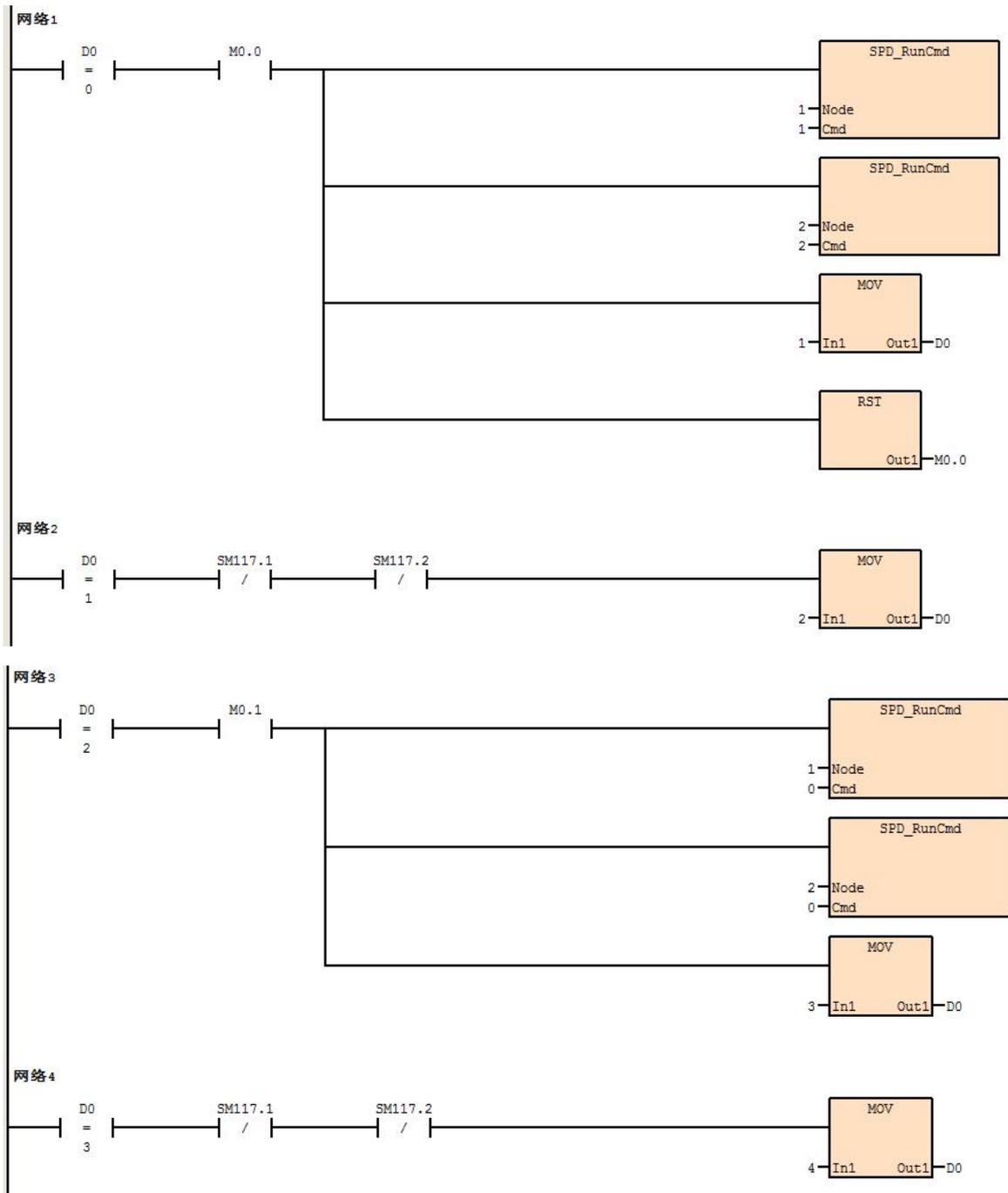
- 1, 该指令的功能是, 控制调速器的运转;
- 2, 调速器当前所处的运转状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

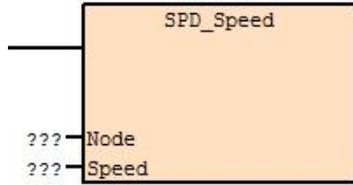
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1“运转命令”设置为“正转”，对站号 2“运转命令”为“反转”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“设置运转”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2；
- 在步骤 2 中，置位 M0.0 为 ON，即对站号 1 和站号 2 的“运转命令”都设置为“停止”，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；进入步骤 3；
- 在步骤 3 中，等待站号 1 和站号 2 都完成“设置运转停止”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 4。





### 10.6.2 SPD\_Speed (调速器运转转速设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Speed	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标调速器站号, 范围 1~30;
- 2, Speed: 调速器运转的目标转速, 电源频率为 50Hz 时, 转速设定范围是 0~1400, 单位: r/min;  
 电源频率是 60Hz 时, 转速设定范围是 0~1600, 单位: r/min。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

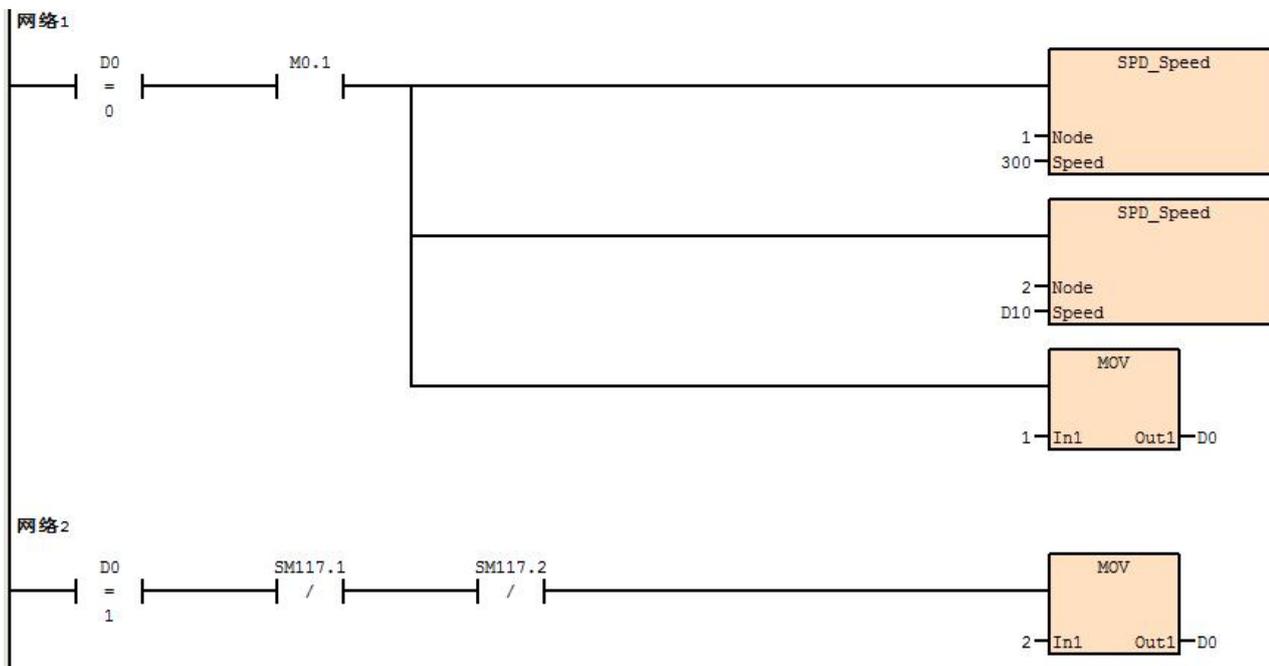
- 1, 该指令的功能是, 设定调速器的运转转速, 该指令仅对 YT 系列调速电机和 YF 系列调速电磁制动电机有效, 当电机转速小于 90r/min 时, 电机停止。

**程序范例:**

(见下页)

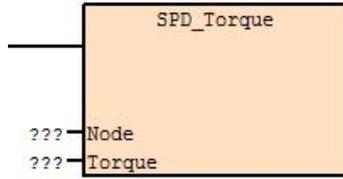
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1 设置“运转转速”为 300 r/min，对调速器 2 设置“运转转速”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“运转转速”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.6.3 SPD\_Torque (调速器运转力矩设定)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
Torque	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

- 1, Node: 目标调速器站号, 范围 1~30;
- 2, Torque: 调速器运转的目标输出力矩值, 设定范围是 0~100, 单位: 1%, 此值应小于菜单中设置的最大力矩。

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

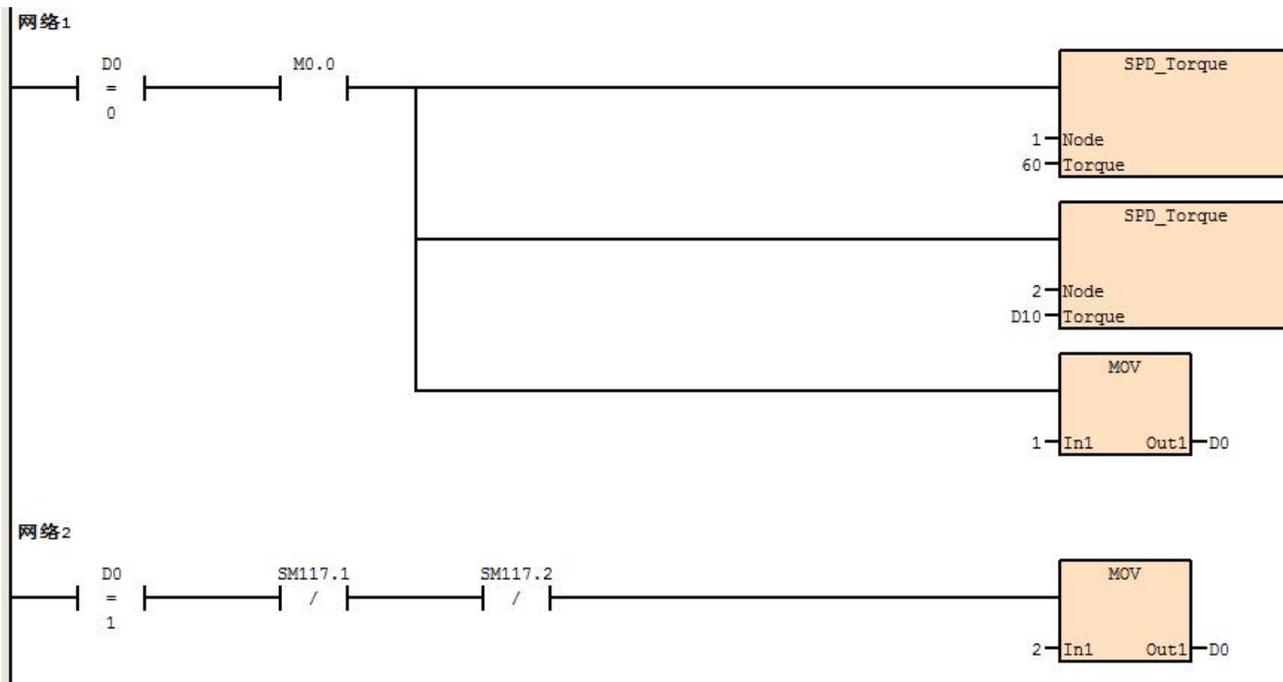
- 1, 该指令的功能是, 设定调速器的运转输出力矩值, 该指令仅对 TP 系列力矩电机有效。

**程序范例:**

(见下页)

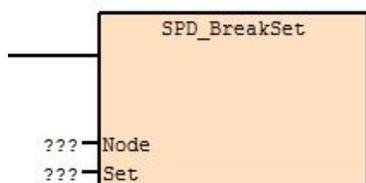
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对调速器 1 设置“输出转矩”为 60%，对调速器 2 设置“输出转矩”为 D10 装置内的数值，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬移到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成设置“输出转矩”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.6.4 SPD\_BreakSet (调速器失电电磁制动器控制)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■
变量名	数据类型	输入输出	位装置															
			X	Y	S M	M	T	C	H C	L	E L							
Set	Bit	Input	■	■	■	■	■	■	■	■	■							

#### 参数说明:

- 1, Node: 目标调速器站号, 范围 1~30;
- 2, Set: 失电电磁制动器控制, OFF: 松闸; ON: 抱闸。

#### 指令执行注意点:

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

#### 指令功能说明:

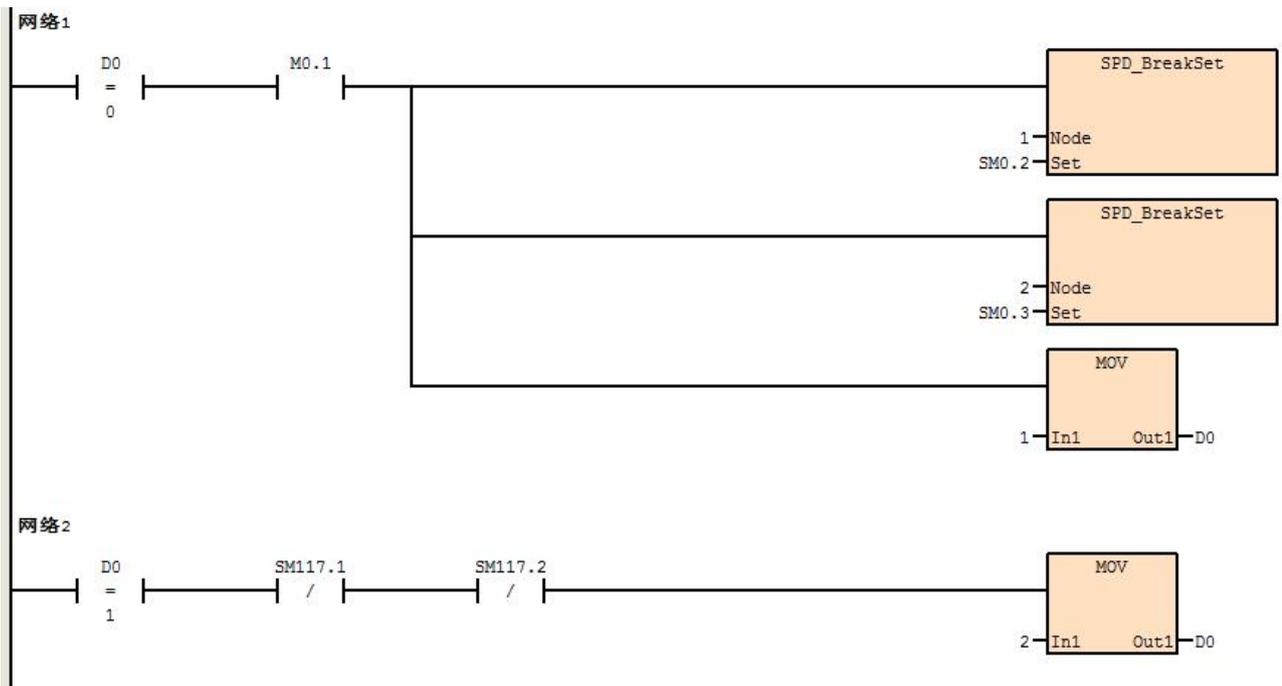
- 1, 该指令的功能是, 在电机停止运转状态下, 控制调速器失电电磁制动器抱闸和松闸动作, 该指令仅对 YF 系列调速电磁制动电机和 YB 系列电磁制动电机有效。

#### 程序范例:

(见下页)

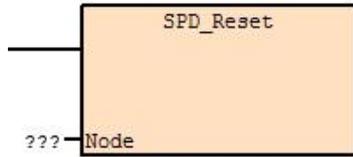
例:

- 在例程中，D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即设置调速器 1“失电电磁制动器”为抱闸（SM0.2 为常 ON 标志位），设置调速器 2“失电电磁制动器”为松闸（SM0.3 为常 OFF 标志位），发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“设置失电制动器”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。





### 10.6.5 SPD\_Reset (调速器故障复位)



变量名	数据类型	输入输出	字/双字装置															
			X	Y	S M	M	T	C	H C	A I	A Q	D	S D	E	N D	O	L	E L
Node	Word	Input	■	■	■	■	■	□		■	■	■	■	■	■	■	■	■

**参数说明:**

1, Node: 目标调速器站号, 范围 1~30;

**指令执行注意点:**

- 1, 该指令是能流上升沿触发执行, 若要再次执行, 需要将能流失电一个扫描周期后, 再次接通能流;
- 2, 指令执行时, 会置 ON 对应站号的忙碌标志位 SM (站号 1~31 分别对应 SM117.1~SM120.7), 直到执行完成置 OFF;
- 3, 同一时间, 同一个从站只能执行一个总线指令, 必须等待其他指令完成后, 才能接通能流执行其他指令, 请使用忙碌标志位 SM 辅助判断。

**指令功能说明:**

- 1, 该指令的功能是, 当调速器处于故障状态时, 使用该指令可将故障状态清除;
- 2, 调速器当前所处的运转状态, 可以通过对应站号配置的 ND (CC-CAN 设备状态信息自动读取装置) 来查看。

**程序范例:**

(见下页)

例:

- 在例程中，调速器 1 和调速器 2 处于“报错状态”中；D0 用于表示程序步骤，步骤 0（即 D0=0）当站号 1 和站号 2 的从站都空闲时，即 SM117.1 和 SM117.2 均为 OFF，（SM117.1 对应站号 1 的忙碌标志，SM117.2 对应站号 2 的忙碌标志），置位 M0.0 为 ON，即对站号 1 和站号 2 发送“清除故障状态”指令，发送过程中忙碌标志位 SM117.1 和 SM117.2 置 ON；同时将数值 1 搬运到 D0 装置中，表明进入步骤 1；
- 在步骤 1 中，等待站号 1 和站号 2 都完成“清除故障状态”命令，即 SM117.1 和 SM117.2 都是 OFF 时，进入步骤 2。

