Raynen 睿能®

X5S/X5 可编程逻辑控制器用户手册软件篇

CE TÜV

Raynen 睿能®

股票代码：603933

目录

目录 1
一、X 系列内存详解 10
1.1. X5S/X5 资源配 10
1.2. X5S/X5 特殊内存配置 12
1.2.1 特殊辅助寄存器 SM8 12
1.2.2 特殊数据寄存器 SD8 19
二 、指令篇 25
2.1 基本指令 34
2.1.1 LD、LDI/取、取反指令 36
2.1.2 OUT/输出指令 38
2.1.3 AND、ANI/与、与反转指令 40
2.1.4 OR、ORI/或、或反转指令 41
2.1.5 LDP、LDF、ANDP、ANDF、ORP、ORF/沿触发类指令 42
2.1.6 ORB/回路块或指令 44
2.1.7 ANB/回路块与指令 45
2.1.8 MPS、MRD、MPP/堆栈类指令 46
2.1.9 MC、MCR/主控、主控复位指令 47
2.1.10 INV/反转指令 49
2.1.11 MEP、MEF/运算结果脉冲化指令指令 50
2.1.12 PLS、PLF/脉冲输出指令 50
2.1.13 SET、RST/置位、复位指令 51
2.1.14 NOP/空指令 53
2.1.15 END/程序结束指令 53
2.2 程序流程指令 53
2.2.1 CJ/程序跳转指令 53
2.2.2 CALL/子程序调用指令 55
2.2.3 SRET/子程序返回指令 58
2.2.4 IRET/中断返回指令 58

2.2.5 EI/允许中断指令..... 58
2.2.6 DI/禁止中断指令..... 59
2.2.7 FEND/主程序结束指令..... 59
2.2.8 WDT/看门狗定时器指令..... 59
2.2.9 FOR/循环范围开始指令..... 60
2.2.10 NEXT/循环范围结束指令..... 62
2.3 传送比较指令..... 62
2.3.1 CMP/比较指令..... 62
2.3.2 ZCP/区间比较指令..... 64
2.3.3 MOV/传送指令..... 65
2.3.4 SMOV/位移动指令..... 66
2.3.5 CML/反转指令..... 68
2.3.6 BMOV/成批传送指令..... 69
2.3.7 FMOV/多点传送指令..... 71
2.3.8 XCH/交换指令..... 73
2.3.9 BCD/BCD 转换指令 74
2.3.10 BIN/BIN 转换指令 75
2.4 四则逻辑运算指令..... 76
2.4.1 ADD/加法运算指令..... 76
2.4.2 SUB/减法运算指令..... 78
2.4.3 MUL/乘法运算指令..... 79
2.4.4 DIV/除法运算指令..... 81
2.4.5 INC/加一指令..... 82
2.4.6 DEC/减一指令..... 83
2.4.7 WAND/逻辑与指令..... 84
2.4.8 WOR/逻辑或指令..... 85
2.4.9 WXOR/逻辑异或指令..... 87
2.4.10 NEG/补码指令..... 88
2.5 循环移位指令..... 89
2.5.1 ROR/循环右移指令..... 89

2.5.2 ROL/循环左移指令..... 91
2.5.3 RCR/带进位循环右移指令..... 92
2.5.4 RCL/带进位循环左移指令..... 94
2.5.5 SFTR/位右移指令..... 95
2.5.6 SFTL/位左移指令..... 97
2.5.7 WSFR/字右移指令..... 99
2.5.8 WSFL/字左移指令..... 102
2.5.9 SFWR/移位写入指令..... 105
2.5.10 SFRD/移位读出指令..... 106
2.6 数据处理指令..... 108
2.6.1 ZRST/成批复位指令..... 108
2.6.2 DECO/译码指令..... 109
2.6.3 ENCO/编码指令..... 111
2.6.4 SUM/计算 ON 位数指令..... 113
2.6.5 BON/ON 位的判定指令 114
2.6.6 MEAN/平均值指令..... 116
2.6.7 ANS/信号报警器置位指令..... 117
2.6.8 ANR/信号报警器复位指令..... 119
2.6.9 SQR/开方运算指令..... 120
2.6.10 FLT/BIN 整数→2 进制浮点数转换指令..... 121
2.7 高速处理指令..... 122
2.7.1 REF/输入输出刷新指令..... 122
2.7.2 REFF/带滤波器的输入刷新指令..... 124
2.7.3 HSCS/高速比较置位指令..... 125
2.7.4 HSCR/高速比较复位指令..... 128
2.7.5 HSZ/高速区间比较指令..... 129
2.7.6 SPD/脉冲密度指令..... 135
2.8 通信指令..... 136
2.8.1 RS2/串行数据传送 2 指令..... 136
2.8.2 ADPRW/MODBUS 通信主站用指令 136

2.9 浮点运算指令..... 137
2.9.1 ECMP/浮点数比较指令..... 137
2.9.2 EZCP/2 进制浮点数区间比较指令 138
2.9.3 EMOV/2 进制浮点数数据传送指令 139
2.9.4 ESTR/2 进制浮点数→字符串的转换指令 140
2.9.5 EVAL/字符串→2 进制浮点数的转换指令 144
2.9.6 EBCD/2 进制浮点数→10 进制浮点数的转换指令..... 147
2.9.7 EBIN/10 进制浮点数→2 进制浮点数的转换指令..... 148
2.9.8 EADD/2 进制浮点数加法运算指令 149
2.9.9 ESUB/2 进制浮点数减法运算指令 151
2.9.10 EMUL/2 进制浮点数乘法运算指令 152
2.9.11 EDIV/2 进制浮点数除法运算指令 153
2.9.12 EXP/2 进制浮点数指数运算指令 154
2.9.13 LOGE/2 进制浮点数自然对数运算指令 155
2.9.14 LOG10/2 进制浮点数常用对数运算指令 156
2.9.15 ESQR/2 进制浮点数开方运算指令 157
2.9.16 ENEG/2 进制浮点数符号翻转指令 158
2.9.17 INT/2 进制浮点数→BIN 整数的转换指令..... 159
2.9.18 SIN/2 进制浮点数 SIN 运算指令 161
2.9.19 COS/2 进制浮点数 COS 运算指令 162
2.9.20 TAN/2 进制浮点数 TAN 运算指令 163
2.9.21 ASIN/2 进制浮点数 SIN-1 运算指令 164
2.9.22 ACOS/2 进制浮点数 COS-1 运算指令 165
2.9.23 ATAN/2 进制浮点数 TAN-1 运算指令 166
2.9.24 RAD/2 进制浮点数角度→弧度的转换指令 167
2.9.25 DEG/2 进制浮点数弧度→角度的转换指令 168
2.10 数据处理 2 指令..... 170
2.10.1 WSUM/算出数据合计值指令..... 170
2.10.2 WTOB/字节单位的数据分离指令..... 171
2.10.3 BTOW/字节单位的数据结合指令..... 173

2.10.4 UNI/16 位数据的 4 位结合指令 175
2.10.5 DIS/16 位数据的 4 位分离指令 176
2.10.6 SWAP/高低字节互换指令..... 177
2.10.7 SORT2/数据排序 2 指令..... 178
2.11 时钟运算指令..... 182
2.11.1 TCMP/时钟数据比较指令..... 182
2.11.2 TZCP/时钟数据区间比较指令..... 184
2.11.3 TADD/时钟数据加法运算指令..... 185
2.11.4 TSUB/时钟数据减法运算指令..... 186
2.11.5 HTOS/时、分、秒数据的秒转换指令..... 188
2.11.6 STOH/秒数据的（时、分、秒）转换指令..... 189
2.11.7 TRD/读出时钟数据指令..... 190
2.11.8 TWR/写入时钟数据指令..... 191
2.11.9 HOUR/计时表指令..... 193
2.12 格雷码指令..... 195
2.12.1 GRY/格雷码的转换指令..... 195
2.12.2 GBIN/格雷码的逆转换指令..... 196
2.13 数据块指令..... 197
2.13.1 BK+/数据块的加法运算指令..... 197
2.13.2 BK-/数据块的减法运算指令..... 199
2.13.3 BKCMP=、>、<、<>、<=、>=/数据块的比较指令..... 201
2.14 字符串控制指令..... 205
2.14.1 STR/BIN→字符串的转换指令..... 205
2.14.2 VAL/字符串→BIN 的转换指令 208
2.14.3 $+/字符串的结合指令..... 212
2.14.4 LEN/检测出字符串的长度指令..... 214
2.14.5 RIGHT/从字符串的右侧开始取出指令..... 215
2.14.6 LEFT/从字符串的左侧开始取出指令..... 217
2.14.7 MIDR/从字符串中的任意取出指令..... 218
2.14.8 MIDW/从字符串的任意替换指令..... 220

2.14.9 INSTR/字符串的检索指令..... 223
2.14.10 $MOV/字符串的传送指令..... 224
2.15 数据处理 3 指令..... 226
2.15.1 FDEL/数据表的数据删除指令..... 226
2.15.2 FINS/数据表的数据插入指令..... 227
2.15.3 POP/读取后入的数据指令..... 228
2.15.4 SFR/16 位数据 n 位右移（带进位）指令 230
2.15.5 SFL/16 位数据 n 位左移（带进位）指令 232
2.16 触点比较指令..... 233
2.16.1 LD=、>、<、<>、<=、>=/触点比较指令..... 233
2.16.2 AND=、>、<、<>、<=、>=/触点比较指令..... 236
2.16.3 OR=、>、<、<>、<=、>=/触点比较指令..... 239
2.17 数据表处理指令..... 241
2.17.1 LIMIT/上下限位控制指令..... 241
2.17.2 BAND/死区控制指令..... 242
2.17.3 ZONE/区域控制指令..... 244
2.17.4 SCL/定坐标（不同点坐标数据）指令..... 246
2.17.5 DABIN/10 进制 ASCII→BIN 的转换指令 248
2.17.6 BINDA/BIN→10 进制 ASCII 的转换指令 250
2.17.7 SCL2/定坐标 2（X/Y 坐标数据）指令..... 252
2.18 扩展文件寄存器指令..... 255
2.18.1 LOADR/读出扩展文件寄存器指令..... 256
2.18.2 SAVER/成批写入扩展文件储存器指令..... 257
2.18.3 INITR/扩展寄存器的初始化指令..... 259
2.18.4 LOGR/登录到扩展文件寄存器指令..... 260
2.18.5 RWER/扩展文件寄存器的写入指令..... 262
2.18.6 INITER/扩展文件寄存器的初始化指令..... 263
2.19 运动控制指令..... 265
概述..... 265
指令介绍..... 265

轴端子映射..... 268
特殊寄存器..... 268
轴错误说明..... 272
参数类型说明..... 274
轴参数说明..... 277
旋切工艺..... 278
追剪工艺..... 279
2.19.1 MC_Power/启用/禁用轴指令..... 280
2.19.2 MC_MoveAbsolute/绝对定位指令..... 281
2.19.3 MC_MoveRelative/相对定位指令..... 285
2.19.4 MC_MoveVelocity/速度控制指令..... 288
2.19.5 MC_Stop/停止指令..... 291
2.19.6 MC_Home/原点返回指令..... 293
2.19.7 MC_MoveFollow/跟随指令..... 303
2.19.8 MC_GroupEnable/启用轴组指令..... 305
2.19.9 MC_GroupDisable/关闭轴组指令..... 307
2.19.10 MC_GroupAbsLinear_V101/绝对直线插补指令..... 307
2.19.11 MC_GroupRelLinear_V101/相对直线插补指令..... 311
2.19.12 MC_GroupAbsCircular_V101/绝对圆弧插补指令..... 314
2.19.13 MC_GroupRelCircular_V101/相对圆弧插补指令..... 319
2.19.14 MC_GroupStop/轴组停止指令..... 323
2.19.15 MC_FlyingShearInit/追剪初始化指令..... 325
2.19.16 MC_FlyingShear/追剪指令..... 327
2.19.17 MC_FlyingShearOut/追剪脱离指令..... 330
2.19.18 MC_RotaryCutInit/旋切初始化指令..... 330
2.19.19 MC_RotaryCutIn/旋切耦合指令..... 332
2.19.20 MC_RotaryCutOut/旋切脱离指令..... 333
2.19.21 MC_GearIn/电子齿轮耦合指令..... 335
2.19.22 MC_GearOut/电子齿轮脱离指令..... 338
2.19.23 MC_SetPosition/当前位置更改指令..... 340

2.19.24 MC_AxisConfig/轴配置指令..... 342
2.19.25 MC_GearInPos/电子齿轮耦合位置指令..... 343
2.19.26 MC_TouchProbe/探针指令..... 346
2.19.27 MC_Phasing/主轴相位补偿指令..... 348
2.19.28 MC_MoveSuperimposed/位移叠加指令..... 350
2.19.29 MC_CamTableSelect/凸轮表选择指令..... 352
2.19.30 MC_CamIn/凸轮表啮合指令..... 355
2.19.31 MC_CamOut/电子凸轮脱离指令..... 357
2.19.32 MC_CamWritePoint/电子凸轮表关键点设置指令..... 359
2.19.33 MC_GetTappetValue/读取挺杆点状态指令..... 361
BufferMode 和连续插补功能说明 363
2.20 方便指令..... 369
2.20.1 ABSD/凸轮控制绝对方式指令..... 369
2.20.2 INCD/凸轮控制相对方式指令..... 371
2.20.3 ALT/交替输出指令..... 372
2.20.4 SORT/数据排序指令..... 373
三、应用篇..... 377
3.1 子程序..... 377
3.2 中断程序..... 377
3.3 高速计数..... 383
3.4 串口通讯部分..... 387
3.4.1 监控下载 387
3.4.2 RS2 通信 388
3.4.3 串口 MODBUS RTU 通信..... 394
3.5 密码功能..... 404
3.6 以太网功能..... 411
3.7 扩展模块的使用..... 419
3.7.1 扩展模块的 ID 419
3.7.2 数字量扩展模块..... 421
3.7.3 模拟量扩展模块..... 421

3.7.4 模块寄存器参照表..... 422
3.7.5 模块配置使能开关..... 424
3.7.6 总线报错和模拟量模块指示灯..... 425
3.7.7 断线恢复..... 425
3.7.8 模块配置举例..... 425
3.7.9 模块配置注意事项..... 429
3.8 BD 板模块的使用 429
3.8.1 BD 板的 ID 及版本说明 429
3.8.2 数字量 BD 板使用说明..... 430
3.8.3 模拟量 BD 板使用说明..... 430
3.8.4 模拟量 BD 板地址对照表..... 431
四、附录..... 433
4.1 特殊报错寄存器错误码内容..... 433

一、X 系列内存详解

1.1. X5S/X5 资源配

名称	内容	软元件类型	内容
X	256 点	位元件	X0 到 X377（8 进制）
Y	256 点	位元件	Y0 到 Y377（8 进制）
M	7680 点	位元件	M0 到 M7679
SM8	512 点	位元件	SM8000~SM8511
SM	1024 点（X5S） 或 1360 点（X5）	位元件	SM0~SM1023（X5S） 或 SM0~SM1359（X5）
S	4096 点	位元件	S0~S4095
V	8 点	字元件	V0~V7
Z	8 点	字元件	Z0~Z7
T	512 点	字元件	T0~T511
C	200 点	16 位字元件	C0~C199
C（32 位）	56 点	32 位字元件	C200~C255
D	8000 点	字元件	D0~D7999
SD8	512 点	字元件	SD8000~SD8511
SD	1024 点（X5S） 或 1360 点（X5）	字元件	SD0~SD1023（X5S） 或 SD0~SD1359（X5）
R	32768 点	字元件	R0~R32767
输入中断	8 点		I00X~I07X
定时器中断	3 点		I1XX~I3XX
计数器中断	6 点		I100~I600
主控用嵌套	8 点		N0~N7
指针 P	4096 点		P0~P4095

常数 K			16 位范围-32768~+32767 32 位范围 -2147483648~+2147483647
常数 H			16 位范围 0~FFFF 32 位范围 0~FFFFFFFF
实数 E			-1.0×2^{128}～ -1.0×2^{-126}, 0, 1.0×2^{-126}～ 1.0×2^{128}（单精度浮点数范围）
字符串			以“”框起来的半角字符 最长 32 个字符

特别说明：

（1）定时器分类

100ms 型 0.1~3276.7s	10ms 型 0.01~327.67s	1ms 累计型 0.001~32.767s	100ms 累计型 0.1~3276.7s	1ms 型 0.001~32.767s
T0~T199 200 点	T200~T245 46 点	T246~T249 4 点 执行中断 保持用	T250~T255 6 点 保持用	T256~T511 256 点
T192~T199 子程序专用				

其中子程序专用和执行中断保持用的用法比较特殊，普通定时器仅在执行子程序时才计时，如果不执行子程序，计时会中断。在子程序中启动了专用定时器，即使子程序不执行了。定时器仍然继续计时，这样就保证了计时的准确性。

（2）V、Z 寄存器

当 32 位指令的操作数进行变址操作时，V 为高 16 位，Z 为低 16 位；对 X，Y 软元件进行变址时，要将 V，Z 值转换成 8 进制后进行加法运算。

（3）掉电保持范围

	符号	进制	点数	起始	结束	锁存起始	锁存结束	锁存设置范围
辅助继电器	M	10	7680	0	7679	500	1023	0-1023

状态	S	10	4096	0	4095	500	999	0-999
定时器	T	10	512	0	511	T246~T255 掉电保持		
计数器（16 位）	C	10	200	0	199	100	199	0-199
计数器（32 位）	C	10	56	200	255	220	255	200-255
数据寄存器	D	10	8000	0	7999	200	511	0-511
扩展寄存器	R	10	32768	0	32767	R0~R32767 掉电保持		

其中：

- 锁存设置范围没有设定时：默认掉电不保持。
- 锁存起始~锁存结束：可以选择，选择将不能保持的变为保持的。
- 锁存结束~结束地址：默认掉电保持。
- X、Y 默认不保存，T 是默认只保持 T246~T255，R 寄存器都保持。

（4）字软元件的位指定 Dx.y

指定字软元件的位，可以将其作为位数据使用，指定字软元件的位时，请使用字软元件编号和位编号(16 进制数)进行设定，其中 x=0~7999，y=0~F。 当做位软元件使用，Dx.y 不支持变址操作，只有基本指令支持 Dx.y 的使用。

1.2. X5S/X5 特殊内存配置

1.2.1 特殊辅助寄存器 SM8

寄存器	功能	说明
SM8000	RUN 时一直为 ON	
SM8001	RUN 时一直为 OFF	
SM8002	仅第一个扫描周期为 ON	
SM8003	仅第一个扫描周期为 OFF	
SM8005	电池电压过低	
SM8006	电池电压过低锁存	
SM8011	10ms 时钟	
SM8012	100ms 时钟	
SM8013	1s 时钟	

SM8014	1min 时钟	
SM8015	停止计时以及预置实时时钟用	
SM8016	时间读出后的显示被停止实时时钟用	
SM8017	±30 秒的修正实时时钟用	
SM8018	检测出安装(一直为 ON)实时时钟用	
SM8019	实时时钟(RTC)错误实时时钟用	
SM8028	指令执行异常结束	
SM8029	指令动作结束标志	
SM8031	非保持内存全部清除	
SM8032	保持内存全部清除	
SM8033	STOP 输出保持	
SM8034	禁止所有输出	
SM8035	强制 RUN	
SM8036	强制 RUN	
SM8037	强制 STOP 指令	
SM8039	恒定扫描模式	
SM8040	STL 状态动作（在执行 END 指令时处理）	
SM8041	STL 监控有效（在执行 END 指令时处理）	
SM8048	信号报警器动作	
SM8049	信号报警器有效	
SM8060	错误发生标志位	
SM8061	PLC 硬件错误	
SM8064	参数错误	
SM8065	语法错误	
SM8066	回路错误	
SM8067	运算错误	
SM8068	运算出错锁存	
SM8080	零标志位	
SM8081	借位标志位	

SM8082	进位标志位	
SM8083	零位，乘除运算结果为 0 时，置 ON	
SM8084	进位，除法运算结果溢出时，置 ON	
SM8090	指定 BMOV 方向	
SM8091	XCH 的 SWAP 功能	
SM8092	SORT2(FNC 149)指令降序排列	
SM8093	SMOV(FNC 13)处理 HEX 数据的功能	
SM8094	BKCMP 指令 块比较信号	
SM8095	BINDA 指令输出字符数切换信号	
SM8110	X000 脉冲捕捉	
SM8111	X001 脉冲捕捉	
SM8112	X002 脉冲捕捉	
SM8113	X003 脉冲捕捉	
SM8114	X004 脉冲捕捉	
SM8115	X005 脉冲捕捉	
SM8116	X006 脉冲捕捉	
SM8117	X007 脉冲捕捉	
SM8120	X000 中断重定向到 X010	
SM8121	X001 中断重定向到 X011	
SM8122	X002 中断重定向到 X012	
SM8123	X003 中断重定向到 X013	
SM8124	X004 中断重定向到 X014	
SM8125	X005 中断重定向到 X015	
SM8126	X006 中断重定向到 X016	
SM8127	X007 中断重定向到 X017	
SM8130	禁止 X000 中断 I00X	
SM8131	禁止 X001 中断 I01X	
SM8132	禁止 X002 中断 I02X	
SM8133	禁止 X003 中断 I03X	

SM8134	禁止 X004 中断 I04X	
SM8135	禁止 X005 中断 I05X	
SM8136	禁止 X006 中断 I06X	
SM8137	禁止 X007 中断 I07X	
SM8139	外部中断延时标志	
SM8140	禁止定时器中断 I1XX	
SM8141	禁止定时器中断 I2XX	
SM8142	禁止定时器中断 I3XX	
SM8150	禁止计数器中断 I100	
SM8151	禁止计数器中断 I200	
SM8152	禁止计数器中断 I300	
SM8153	禁止计数器中断 I400	
SM8154	禁止计数器中断 I500	
SM8155	禁止计数器中断 I600	
SM8180	HSZ 指令表格使能	
SM8181	HSZ 指令表格比较执行结束标志位	
SM8187	高速环形计数器（0.1ms 单位，16 位）动作	
SM8188	1ms 的环形计数器（32 位 SD8188，SD8189）动作	
SM8191	C251 的 4 倍频使能	
SM8192	C252 的 4 倍频使能	
SM8193	C253 的 4 倍频使能	
SM8194	C254 的 4 倍频使能	
SM8195	C255 的 4 倍频使能	
SM8196	C233 的 4 倍频使能	
SM8197	C234 的 4 倍频使能	
SM8200	C200 增减计数模式控制	
SM8201	C201 增减计数模式控制	
SM8202	C202 增减计数模式控制	
SM8203	C203 增减计数模式控制	

SM8204	C204 增减计数模式控制	
SM8205	C205 增减计数模式控制	
SM8206	C206 增减计数模式控制	
SM8207	C207 增减计数模式控制	
SM8208	C208 增减计数模式控制	
SM8209	C209 增减计数模式控制	
SM8210	C210 增减计数模式控制	
SM8211	C211 增减计数模式控制	
SM8212	C212 增减计数模式控制	
SM8213	C213 增减计数模式控制	
SM8214	C214 增减计数模式控制	
SM8215	C215 增减计数模式控制	
SM8216	C216 增减计数模式控制	
SM8217	C217 增减计数模式控制	
SM8218	C218 增减计数模式控制	
SM8219	C219 增减计数模式控制	
SM8220	C220 增减计数模式控制	
SM8221	C221 增减计数模式控制	
SM8222	C222 增减计数模式控制	
SM8223	C223 增减计数模式控制	
SM8224	C224 增减计数模式控制	
SM8225	C225 增减计数模式控制	
SM8226	C226 增减计数模式控制	
SM8227	C227 增减计数模式控制	
SM8228	C228 增减计数模式控制	
SM8229	C229 增减计数模式控制	
SM8230	C230 增减计数模式控制	
SM8231	C231 增减计数模式控制	
SM8232	C232 增减计数模式控制	

SM8233	C233 增减计数模式控制	
SM8234	C234 增减计数模式控制	
SM8235	C235 增减计数模式控制	
SM8236	C236 增减计数模式控制	
SM8237	C237 增减计数模式控制	
SM8238	C238 增减计数模式控制	
SM8239	C239 增减计数模式控制	
SM8240	C240 增减计数模式控制	
SM8241	C241 增减计数模式控制	
SM8242	C242 增减计数模式控制	
SM8243	C243 增减计数模式控制	
SM8244	C244 增减计数模式控制	
SM8245	C245 增减计数模式控制	
SM8246	C246 增减计数状态指示	
SM8247	C247 增减计数状态指示	
SM8248	C248 增减计数状态指示	
SM8249	C249 增减计数状态指示	
SM8250	C250 增减计数状态指示	
SM8251	C251 增减计数状态指示	
SM8252	C252 增减计数状态指示	
SM8253	C253 增减计数状态指示	
SM8254	C254 增减计数状态指示	
SM8255	C255 增减计数状态指示	
SM8301	[COM0]串口通信出错锁存	
SM8302	[COM0]超时的判断标志	
SM8303	[COM0]MODBUS 通信中	
SM8304	[COM0]MODBUS 通信发生错误	
SM8305	[COM0]主站发生重发事件的判断标志	
SM8306	[COM0]MODBUS 通信出错标志	

SM8310	[COM0]RS2 发送请求控制	
SM8311	[COM0]RS2 接收结束标志	
SM8312	[COM0]RS2 发送等待指示	
SM8321	[COM1]串口通信出错锁存	
SM8322	[COM1]超时的判断标志	
SM8323	[COM1]MODBUS 通信中	
SM8324	[COM1]MODBUS 通信发生错误	
SM8325	[COM1]主站发生重发事件的判断标志	
SM8326	[COM1]MODBUS 通信出错标志	
SM8330	[COM1]RS2 发送请求控制	
SM8331	[COM1]RS2 接收结束标志	
SM8332	[COM1]RS2 发送等待指示	
SM8341	[COM2]串口通信出错锁存	
SM8342	[COM2]超时的判断标志	
SM8343	[COM2]MODBUS 通信中	
SM8344	[COM2]MODBUS 通信发生错误	
SM8345	[COM2]主站发生重发事件的判断标志	
SM8346	[COM2]MODBUS 通信出错标志	
SM8350	[COM2]RS2 发送请求控制	
SM8351	[COM2]RS2 接收结束标志	
SM8352	[COM2]RS2 发送等待指示	
SM8400	以太网功能禁用标志	
SM8401	MODBUS TCP 功能禁用标志	
SM8402	MODBUS TCP 通信时从站不在线标志位	
SM8403	MODBUS TCP 通信时主站网线未连接或断开标志位	
SM8420	模块总线错误标志	
SM8430	模块离线配置清除标志	
SM8431	[MD1]配置使能	
SM8432	[MD2]配置使能	

SM8433	[MD3]配置使能	
SM8434	[MD4]配置使能	
SM8435	[MD5]配置使能	
SM8436	[MD6]配置使能	
SM8437	[MD7]配置使能	
SM8438	[BD1]配置使能	
SM8439	[BD2]配置使能	
SM8441	[MD1]命令使能	
SM8442	[MD2]命令使能	
SM8443	[MD3]命令使能	
SM8444	[MD4]命令使能	
SM8445	[MD5]命令使能	
SM8446	[MD6]命令使能	
SM8447	[MD7]命令使能	
SM8448	[BD1]命令使能	
SM8449	[BD2]命令使能	

1.2.2 特殊数据寄存器 SD8

寄存器	功能	说明（默认值）
SD8000	看门狗定时器（默认 200，单位 ms）	200
SD8001	PLC 类型及系统版本	4100
SD8002	内存容量 ，固定为 16	16
SD8003	IO 口描述	0x1010
SD8004	PLC 补充版本号	暂时无效
SD8005	电池电压，单位 0.1v	33
SD8006	检测出电池电压低的等级，单位 0.1v	27
SD8010	当前扫描时间（0.1ms）	
SD8011	最小扫描时间（0.1ms）	
SD8012	最大扫描时间（0.1ms）	

SD8013	实时时钟（秒）	
SD8014	实时时钟（分）	
SD8015	实时时钟（时）	
SD8016	实时时钟（日）	
SD8017	实时时钟（月）	
SD8018	实时时钟（年）	
SD8019	实时时钟（星期）	
SD8020	输入滤波器值（单位 ms）	10
SD8021	FPGA 固件版本	
SD8039	恒定扫描时间设定（单位 ms）	
SD8040	ON 状态编号 1	状 态 S0-S899,S1000-S4095 中正在动作的状态的最小编号保存到 SD8040 中。其次为 ON 的状态保存到紧接着的寄存器，最大八点
SD8041	ON 状态编号 2	
SD8042	ON 状态编号 3	
SD8043	ON 状态编号 4	
SD8044	ON 状态编号 5	
SD8045	ON 状态编号 6	
SD8046	ON 状态编号 7	
SD8047	ON 状态编号 8	
SD8049	ON 状态最小编号	SM8049 为 ON 时，保存信号报警继电器 S900～S999 中为 ON 的状态的最小编号
SD8060	PLC 错误类型	存储对应的 SM 编号
SD8061	PLC 硬件错误的错误代码编号	
SD8062	语法、回路、运算错误步（低位）	
SD8063	语法、回路、运算错误步（高位）	
SD8064	参数错误的错误代码编号	
SD8065	语法错误的错误代码编号	

SD8066	回路错误的错误代码编号	
SD8067	运算错误的错误代码编号	
SD8068	运算错误的错误代码编号的锁存	
SD8070	发生运算出错的步编号的锁存（低位）	
SD8071	发生运算出错的步编号的锁存（高位）	
SD8080	Z0 寄存器的内容	
SD8081	V0 寄存器的内容	
SD8082	Z1 寄存器的内容	
SD8083	V1 寄存器的内容	
SD8084	Z2 寄存器的内容	
SD8085	V2 寄存器的内容	
SD8086	Z3 寄存器的内容	
SD8087	V3 寄存器的内容	
SD8088	Z4 寄存器的内容	
SD8089	V4 寄存器的内容	
SD8090	Z5 寄存器的内容	
SD8091	V5 寄存器的内容	
SD8092	Z6 寄存器的内容	
SD8093	V6 寄存器的内容	
SD8094	Z7 寄存器的内容	
SD8095	V7 寄存器的内容	
SD8139	外部中断延时寄存器	
SD8180	HSZ 比较表格计数器	
SD8187	16 位递增环形计数器	0~32767（0.1ms 单位 16 位）递增
SD8188	32 位递增环形计数器（低位）	0~2147483647（1ms 单位）递增
SD8189	32 位递增环形计数器（高位）	
SD8300	[COM0]通信格式设定	详情见通信口配置寄存器表格

SD8301	[COM0]错误代码编号	
SD8302	[COM0]设定响应超时时间	
SD8303	[COM0]协议	
SD8304	[COM0]显示主站当前的重发次数	
SD8305	[COM0]设定进行重发的次数	
SD8306	[COM0]请求间延迟	
SD8307	[COM0]从站本站号	从栈可设
SD8308	[COM0]MODBUS 通信错误寄存器	显示当前最新错误
SD8309	[COM0]MODBUS 通信错误锁存寄存器	锁存第一次的错误
SD8310	[COM0]RS2 发送剩余计数器	
SD8311	[COM0]RS2 接收计数器	
SD8312	[COM0]RS2 报头 1、2 设定	
SD8313	[COM0]RS2 报头 3、4 设定	
SD8314	[COM0]RS2 报尾 1、2 设定	
SD8315	[COM0]RS2 报尾 3、4 设定	
SD8316	[COM0]RS2 接收和校验	
SD8317	[COM0]RS2 接收和校验计算结果	
SD8318	[COM0]RS2 发送和校验	
SD8320	[COM1]通信格式设定	详情见通信口配置寄存器表格
SD8321	[COM1]错误代码编号	
SD8322	[COM1]设定响应超时时间	
SD8323	[COM1]协议	
SD8324	[COM1]显示主站当前的重发次数	
SD8325	[COM1]设定进行重发的次数	
SD8326	[COM1]请求间延迟	
SD8327	[COM1]从站本站号	从栈可设
SD8328	[COM1]MODBUS 通信错误寄存器	显示当前最新错误
SD8329	[COM1]MODBUS 通信错误锁存寄存器	锁存第一次的错误

SD8330	[COM1]RS2 发送剩余计数器	
SD8331	[COM1]RS2 接收计数器	
SD8332	[COM1]RS2 报头 1、2 设定	
SD8333	[COM1]RS2 报头 3、4 设定	
SD8334	[COM1]RS2 报尾 1、2 设定	
SD8335	[COM1]RS2 报尾 3、4 设定	
SD8336	[COM1]RS2 接收和校验	
SD8337	[COM1]RS2 接收和校验计算结果	
SD8338	[COM1]RS2 发送和校验	
SD8340	[COM2]通信格式设定	详情见通信口配置寄存器表格
SD8341	[COM2]错误代码编号	
SD8342	[COM2]设定响应超时时间	
SD8343	[COM2]协议	
SD8344	[COM2]显示主站当前的重发次数	
SD8345	[COM2]设定进行重发的次数	
SD8346	[COM2]请求间延迟	
SD8347	[COM2]从站本站号	从栈可设
SD8348	[COM2]MODBUS 通信错误寄存器	显示当前最新错误
SD8349	[COM2]MODBUS 通信错误锁存寄存器	锁存第一次的错误
SD8350	[COM2]RS2 发送剩余计数器	
SD8351	[COM2]RS2 接收计数器	
SD8352	[COM2]RS2 报头 1、2 设定	
SD8353	[COM2]RS2 报头 3、4 设定	
SD8354	[COM2]RS2 报尾 1、2 设定	
SD8355	[COM2]RS2 报尾 3、4 设定	
SD8356	[COM2]RS2 接收和校验	
SD8357	[COM2]RS2 接收和校验计算结果	
SD8358	[COM2]RS2 发送和校验	

SD8400	显示 PLC 当前分配协议连接个数	
SD8401	显示当前 TCP 连接数量	
SD8402	不在线从站对应的 IP（第四段）	
SD8403	显示 MODBUS TCP 错误码	
SD8404	设定 MODBUS TCP 主站接收超时时间	ms 单位
SD8405	保存当前最新的超时 IP 及超时数据所在配置表中的位置	
SD8406	尝试重连时间间隔	ms 单位
SD8409	用户不可用	MAC 地址烧录用
SD8410	本机 IP 第一段	
SD8411	本机 IP 第二段	
SD8412	本机 IP 第三段	
SD8413	本机 IP 第四段	
SD8420	模块总线错误代码编号	
SD8421	[MD1]ID	模块 1 数据寄存器在 D7500
SD8422	[MD2]ID	模块 2 数据寄存器在 D7550
SD8423	[MD3]ID	模块 3 数据寄存器在 D7600
SD8424	[MD4]ID	模块 4 数据寄存器在 D7650
SD8425	[MD5]ID	模块 5 数据寄存器在 D7700
SD8426	[MD6]ID	模块 6 数据寄存器在 D7750
SD8427	[MD7]ID	模块 7 数据寄存器在 D7800
SD8428	[BD1]ID 寄存器	BD1 数据寄存器在 D7900
SD8429	[BD2]ID 寄存器	BD2 数据寄存器在 D7950
SD8431	[MD1]版本寄存器	
SD8432	[MD2]版本寄存器	
SD8433	[MD3]版本寄存器	
SD8434	[MD4]版本寄存器	
SD8435	[MD5]版本寄存器	
SD8436	[MD6]版本寄存器	

SD8437	[MD7]版本寄存器	
SD8438	[BD1]版本寄存器	
SD8439	[BD2]版本寄存器	

二 、指令篇

一条指令一般由指令头、源操作数、目的操作数组成，源操作数和目的操作数可以没有，

注意：

（1）程序中出现未实现的指令会报停机错误 6506。

（2）指令后面+P 代表边沿触发，指令前面+D 代表 32 位指令。

X5S/X5 支持使用的指令集如下表：

基本指令	
指令	指令说明
LD	装载
LDI	装载取反
LDP	取脉冲上升沿
LDF	取脉冲下降沿
AND	与
ANI	与非
ANDP	与脉冲上升沿
ANDF	与脉冲下降沿
OR	或
ORI	或非
ORP	或脉冲上升沿
ORF	或脉冲下降沿
MEP	运算结果上升沿脉冲化指令
MEF	运算结果下降沿脉冲化指令

ANB	回路块与
ORB	回路块或
MPS	存储进栈
MRD	存储读栈
MPP	存储出栈
INV	取反
OUT	输出
SET	置位
RST	复位
PLS	上升沿脉冲检测
PLF	下降沿脉冲检测
MC	主控
MCR	主控复位
NOP	空操作
END	程序结束
STL	步进梯形图开始
RET	步进梯形图结束
程序流程指令	
指令	说明
CJ	条件跳转
CALL	子程序调用
SRET	子程序返回
IRET	中断返回
EI	允许中断
DI	禁止中断
FEND	主程序结束
WDT	看门狗定时器
FOR	循环范围的开始
NEXT	循环范围的结束

传送比较指令	
指令	说明
CMP	比较
ZCP	区间比较
MOV	传送
SMOV	移位
CML	取反传送
BMOV	成批传送
FMOV	多点传送
XCH	交换
BCD	BCD 转换
BIN	BIN 转换
四则逻辑运算指令	
指令	说明
ADD	BIN 加法
SUB	BIN 减法
MUL	BIN 乘法
DIV	BIN 除法
INC	BIN 加 1
DEC	BIN 减 1
WAND	逻辑与
WOR	逻辑或
WXOR	逻辑异或
NEG	求补码
循环移位指令	
指令	说明
ROR	循环右移
ROL	循环左移
RCR	带进位循环右移

RCL	带进位循环左移
SFTR	位右移
SFTL	位左移
WSFR	字右移
WSFL	字左移
SFWR	移位写入
SFRD	移位读出
数据处理指令	
指令	说明
ZRST	成批复位
DECO	译码
ENCO	编码
SUM	ON 位总数
BON	ON 的判定
MEAN	平均值
ANS	信号报警器置位
ANR	信号报警器复位
SQR	BIN 开方
FLT	BIN 整数->2 进制浮点数转换
高速处理指令	
指令	说明
REF	输入输出刷新
REFF	滤波器调整
HSCS	比较置位（高速计数器）
HSCR	比较复位（高速计数器）
HSZ	区间比较（高速计数器）
SPD	脉冲密度
通信指令	
RS2	自由口通信

ADPRW	用于建立 MODBUS 主站的通信
浮点运算指令	
指令	说明
ECMP	2 进制浮点数比较
EZCP	2 进制浮点数区间比较
EMOV	2 进制浮点数数据传送
ESTR	2 进制浮点数→字符串的转换
EVAL	字符串→2 进制浮点数的转换
EBCD	2 进制浮点数→10 进制浮点数转换
EBIN	10 进制浮点数→2 进制浮点数转换
EADD	2 进制浮点数加法运算
ESUB	2 进制浮点数减法运算
EMUL	2 进制浮点数乘法运算
EDIV	2 进制浮点数除法运算
EXP	2 进制浮点数指数运算
LOGE	2 进制浮点数自然对数运算
LOG10	2 进制浮点数常用对数运算
ESQR	2 进制浮点数开方运算
ENEG	2 进制浮点数符号翻转
INT	2 进制浮点数→BIN 整数的转换
SIN	2 进制浮点数 SIN 运算
COS	2 进制浮点数 COS 运算
TAN	2 进制浮点数 TAN 运算
ASIN	2 进制浮点数 SIN-1 运算
ACOS	2 进制浮点数 COS-1 运算
ATAN	2 进制浮点数 TAN-1 运算
RAD	2 进制浮点数角度→弧度的转换
DEG	2 进制浮点数弧度→角度的转换
数据处理 2	

WSUM	算出数据合计值
WTOB	字节单位的数据分离
BTOW	字节单位的数据结合
UNI	16 位数据的 4 位结合
DIS	16 位数据的 4 位分离
SWAP	高低字节互换
SORT2	数据排序 2
时钟运算	
指令	说明
TCMP	时钟数据比较
TZCP	时钟数据区间比较
TADD	时钟数据加法运算
TSUB	时钟数据减法运算
HTOS	时、分、秒数据的秒转换
STOH	秒数据的[时、分、秒]转换
TRD	时钟数据的读出
TWR	时钟数据的读入
HOUR	长时间计时
格雷码	
GRY	格雷码的转换
GBIN	格雷码的逆转换
数据块指令	
BK+	数据块的加法运算
BK-	数据块的减法运算
BKCMP=	数据块比较
BKCMP>	
BKCMP<	
BKCMP<>	
BKCMP<=	

BKCMP>=	
字符串控制指令	
STR	BIN→字符串的转换
VAL	字符串→BIN 的转换
$+	字符串的结合
LEN	检测出字符串的长度
RIGHT	从字符串的右侧开始取出
LEFT	从字符串的左侧开始取出
MIDR	从字符串中的任意取出
MIDW	字符串中的任意替换
INSTR	字符串的检索
$MOV	字符串的传送
数据处理指令 3	
FDEL	数据表的数据删除
FINS	数据表的数据插入
POP	读取后入的数据[先入后出控制用]
SFR	16 位数据 n 位右移(带进位)
SFL	16 位数据 n 位左移(带进位)
触点比较指令	
指令	说明
LD=	(S1)=(S2)
LD>	(S1)>(S2)
LD<	(S1)<(S2)
LD<>	(S1)!=(S2)
LD<=	(S1)<=(S2)
LD>=	(S1)>=(S2)
AND=	(S1)=(S2)
AND>	(S1)>(S2)
AND<	(S1)<(S2)

AND<>	(S1)!=(S2)
AND<=	(S1)<=(S2)
AND>=	(S1)>=(S2)
OR=	(S1)=(S2)
OR>	(S1)>(S2)
OR<	(S1)<(S2)
OR<>	(S1)!=(S2)
OR<=	(S1)<=(S2)
OR>=	(S1)>=(S2)
数据表处理	
LIMIT	上下限限位控制
BAND	死区控制
ZONE	区域控制
SCL	定坐标(不同点坐标数据)
DABIN	10 进制 ASCII→BIN 的转换
BINDA	BIN→10 进制 ASCII 的转换
SCL2	定坐标 2(X/Y 坐标数据)
扩展文件寄存器	
LOADR	读出扩展文件寄存器
SAVER	成批写入扩展文件寄存器
INITR	扩展寄存器的初始化
LOGR	登录到扩展寄存器
RWER	扩展文件寄存器的删除・写入
INITER	扩展文件寄存器的初始化
运动控制指令	
MC_Home	原点回归
MC_Stop	控制指定轴减速到停止
MC_MoveAbsolute	控制指定轴依照指定的运动方式移动到用户设定的绝对目标位置
MC_MoveRelative	控制指定轴依照指定的运动方式移动到用户设定的相对目标位置

MC_MoveVelocity	控制指定轴在位置模式下依照指定的运动方式及速度做均速运动
MC_Power	控制指定轴启动或关闭
MC_GroupEnable	使能轴组
MC_GroupDisable	失能轴组
MC_GroupStop	控制指定轴减速到停止
MC_GroupAbsLinear	控制轴组执行直线插补，使用绝对位置
MC_GroupRelLinear	控制轴组执行直线插补，使用相对位置
MC_GroupAbsCircular	控制轴组执行圆弧，使用绝对位置
MC_GroupRelCircular	控制轴组执行圆弧，使用相对位置
MC_FlyingShearInit	用于追剪功能执行之前的参数初始化
MC_FlyingShear	用于建立追剪关系
MC_FlyingShearOut	用于解除主轴与从轴之间已经建立的追剪关系
MC_RotaryCutInit	用于旋切功能执行之前的参数初始化
MC_RotaryCutIn	用于建立旋切关系
MC_RotaryCutOut	用于解除旋切轴与进给轴之间已经建立的旋切关系
MC_GearIn	建立主从轴间的齿轮关系
MC_GearOut	解除主从轴间已经建立的齿轮关系
MC_MoveFollow	跟随指令，可用作编码器轴
MC_SetPosition	改变坐标系统中轴的当前位置
MC_AxisConfig	轴配置指令
MC_GearInPos	电子齿轮耦合位置指令
MC_CamTableSelect	凸轮表选择指令
MC_CamIn	凸轮表啮合指令
MC_CamOut	电子凸轮脱离指令
MC_CamWritePoint	电子凸轮表关键点设置指令
MC_TouchProbe	探针指令
MC_Phasing	主轴相位补偿指令
MC_MoveSuperimposed	位移叠加指令

方便指令	
ABSD	凸轮控制绝对方式
INCD	凸轮控制相对方式
ALT	交替输出
SORT	数据排序

2.1 基本指令

以梯形图方式编程时，有些基本指令是不可见的，只有转换成指令列表才可以查看。

基本指令大致分五类：

1、触点指令：一般放在梯形图中紧接母线的地方，根据触发方式的不同来作为后面指令执行的一个动作开关。

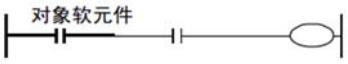
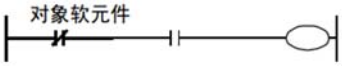
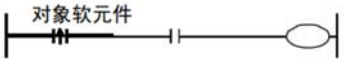
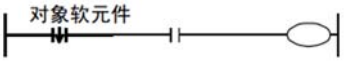
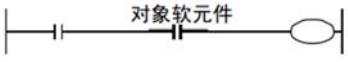
2、结合指令：连接梯形图形成多个执行逻辑。

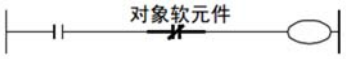
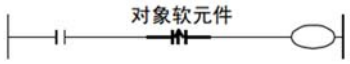
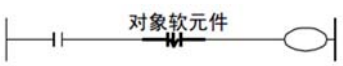
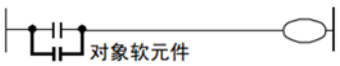
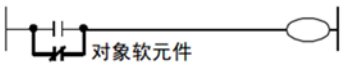
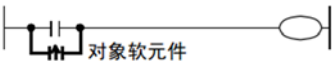
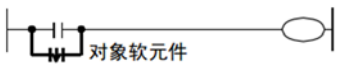
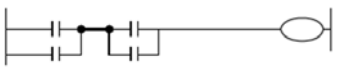
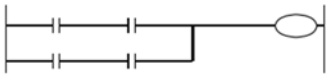
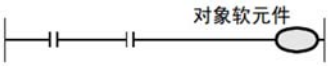
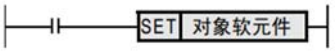
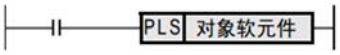
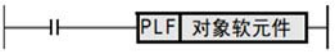
3、输出指令：根据输出方式给对象软元件赋初值。

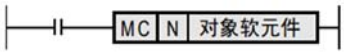
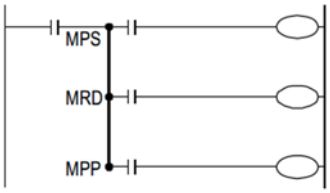
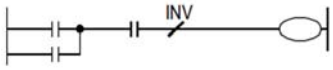
4、主控指令：控制一个程序块的开启关闭。

5、结束指令：END，放于梯形图末尾；FEND 用于主程序结束，会执行与 END 指令相同的输出处理、输入处理、看门狗定时器的刷新，然后返回到 0 步的程序，在编写子程序和中断程序时需要使用 FEND 指令。

基本指令表如下（a 触点：常开触点；b 触点：常闭触点）：

记号	称呼	符号	功能	对象软元件
LD	取		a 触点的逻辑运算开始	X,Y,M,S,T,C,Dx.y
LDI	取反		b 触点的逻辑运算开始	X,Y,M,S,T,C,Dx.y
LDP	取脉冲上升沿		检测到上升沿运算开始	X,Y,M,S,T,C,Dx.y
LDF	取脉冲下降沿		检测到下降沿运算开始	X,Y,M,S,T,C,Dx.y
AND	与		串联 a 触点	X,Y,M,S,T,C,Dx.y

ANI	与反转		串联 b 触点	X,Y,M,S,T,C,Dx.y
ANDP	与脉冲上升沿		上升沿检出的串联连接	X,Y,M,S,T,C,Dx.y
ANDF	与脉冲下降沿		下降沿检出的串联连接	X,Y,M,S,T,C,Dx.y
OR	或		并联 a 触点	X,Y,M,S,T,C,Dx.y
ORI	或反转		并联 b 触点	X,Y,M,S,T,C,Dx.y
ORP	或脉冲上升沿		上升沿检出的并联连接	X,Y,M,S,T,C,Dx.y
ORF	或脉冲下降沿		下降沿检出的并联连接	X,Y,M,S,T,C,Dx.y
ANB	回路块与		回路块的串联连接	-
ORB	回路块或		回路块的并联连接	-
OUT	输出		线圈驱动	Y,M,S,T,C,Dx.y
SET	置位		动作保持	Y,M,S,Dx.y
RST	复位		解除保持的动作，清除当前值及寄存器	Y,M,S,T,C,,Dx.y D,R,V,Z
PLS	脉冲		上升沿微分输出	Y,M
PLF	下降沿脉冲		下降沿微分输出	Y,M
MEP	运算结果上升沿脉冲化指令		运算结果的上升沿时为 ON	-

MEF	运算结果下降沿脉冲化指令		远算结果的下降沿时为 ON	-
MC	主控		连接到公共触点	Y,M
MCR	主控复位		解除连接到公共触点	-
MPS	存储进栈		压入堆栈	-
MRD	存储读栈		读取堆栈	-
MPP	存储出栈		弹出堆栈	-
INV	反转		运算结果的反转	-
NOP	空操作		无处理	-
END	结束		程序结束以及输入输出处理和返回 0 步	-

2.1.1 LD、LDI/取、取反指令

1、概要

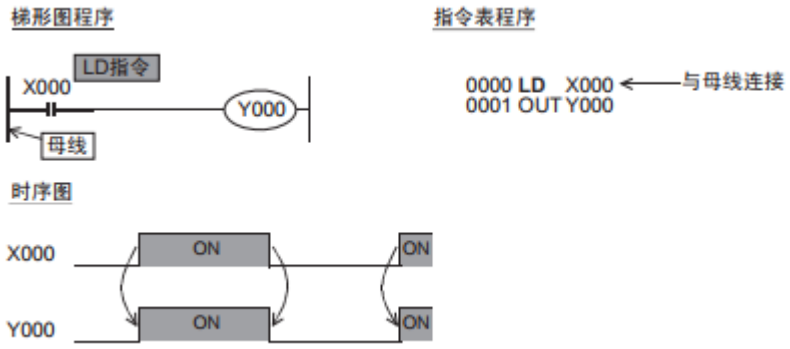
与母线相连的常开触点和常闭触点。

2、操作数可用元件类型

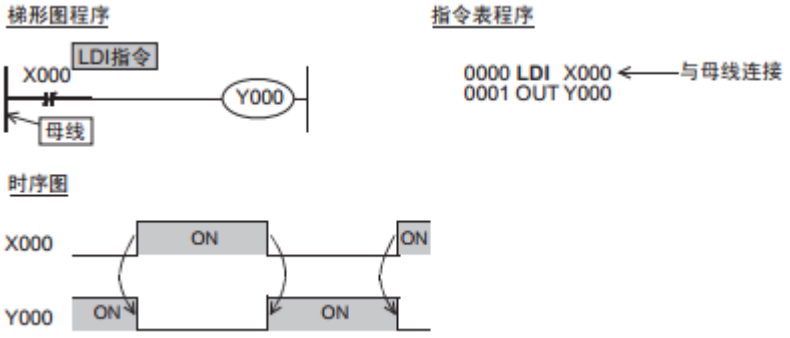
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
LD	*	*	*	*	*	*	*											*					
LDI	*	*	*	*	*	*	*											*					

3、功能和动作说明

（1）LD 指令（a 触点的逻辑运算开始）

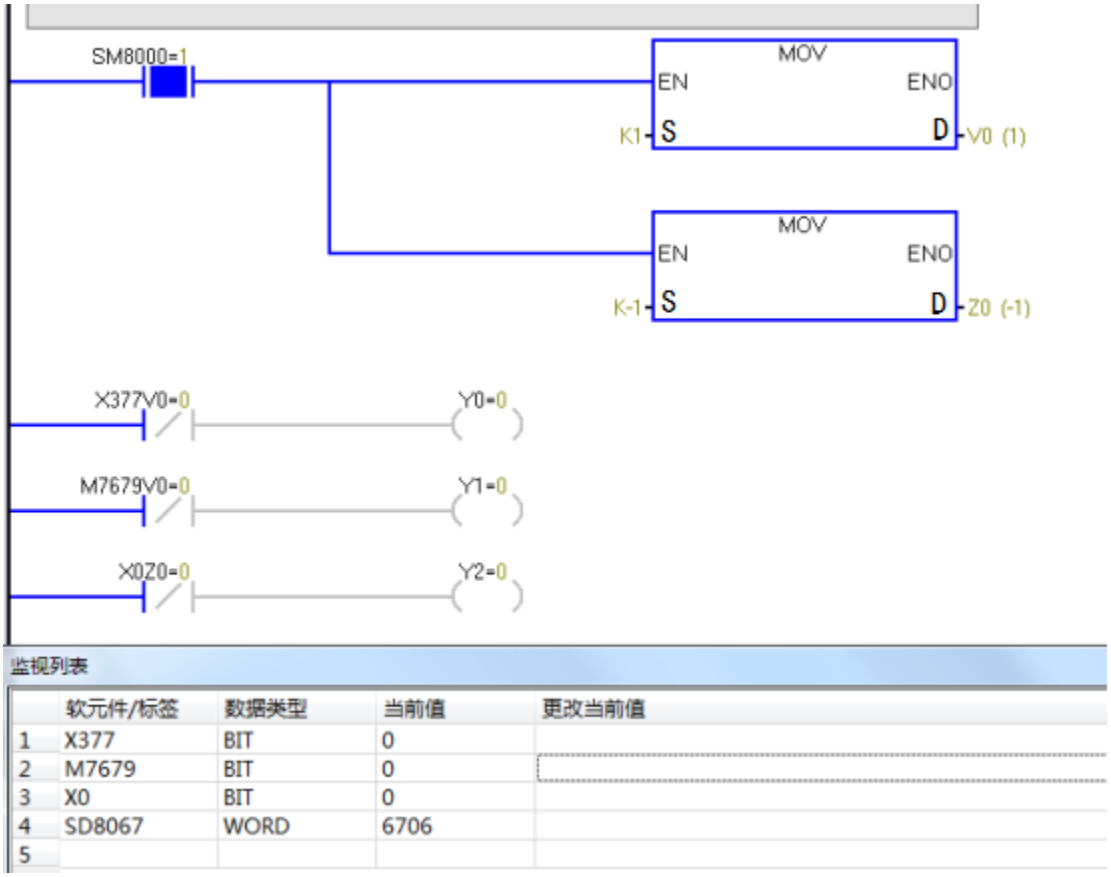


（2）LDI 指令（b 触点的逻辑运算开始）



4、注意事项

使用触点类指令，要注意变址修饰越界的情况，以 LDI 为例，LDI+位软元件变址操作，当位软元件变址操作越界后，XPLC 监控返回 0,但在执行 LDI 时报错返回，不执行取反操作。（XPLC 对于指令中出现的错误问题，只要出错则指令不执行，即输出 0）



监视列表

	软元件/标签	数据类型	当前值	更改当前值
1	X377	BIT	0	
2	M7679	BIT	0	
3	X0	BIT	0	
4	SD8067	WORD	6706	
5				

此梯形图为变址越界时的执行情况，输出 Y 没有点亮，正常情况下 Y 会点亮。

2.1.2 OUT/输出指令

1、概要

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

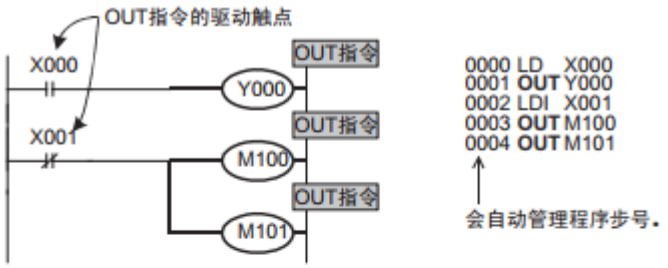
2、操作数可用元件类型

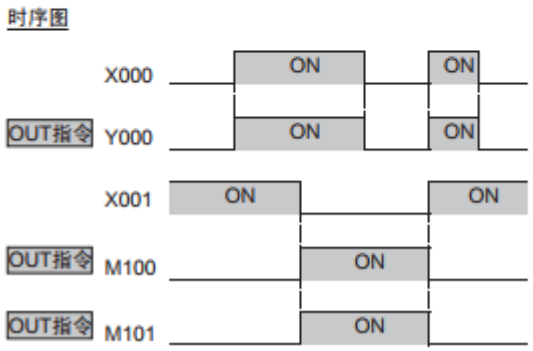
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
OUT		*	*	*	*	*	*											*					
设定值														*	*			*	*				

3、功能和动作说明

（1）使用位元件时

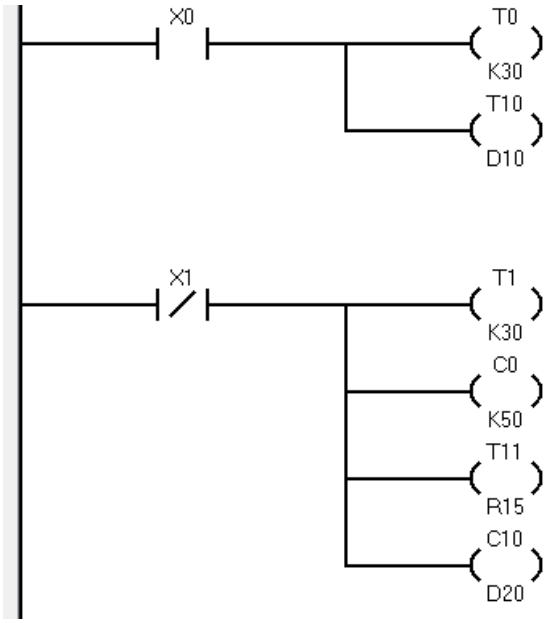
用 OUT 指令编写的软元件，根据驱动触点的状态执行 ON/OFF。





（2）使用定时器和计数器

在针对定时器的计时线圈和计数器的计数线圈的 OUT 指令后需要加上设定值。设定值可以使用 10 进制数(K)直接指定，也可以使用数据寄存器(D)或扩展寄存器(R)间接指定。



4、注意事项

（1）定时器和计数器的设定值的设定范围以及实际的定时器常数、OUT 指令的程序步数(包含设定值)参照 1.1 章节的资源配置表。

（2）当变址后对应的软元件不存在时报错 6706，且变址后的软元件不能为对应的高速计数器。

2.1.3 AND、ANI/与、与反转指令

1、概要

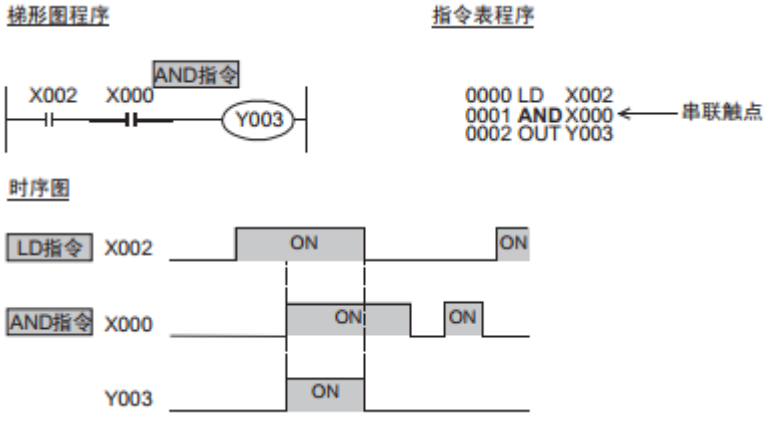
AND、ANI 指令用于串联连接触点。串联触点的数量没有限制，该指令可以连续多次使用。

2、操作数可用元件类型

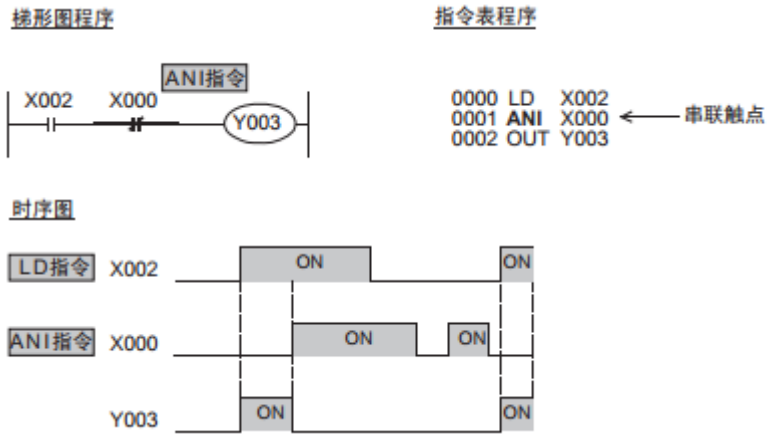
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
AND	*	*	*	*	*	*	*											*					
ANI	*	*	*	*	*	*	*											*					

3、功能和动作说明

（1）AND 指令（串联 a 触点）



（2）ANI 指令（串联 b 触点）



4、注意事项

要注意变址修饰越界的情况，参考 LDI 指令。

2.1.4 OR、ORI/或、或反转指令

1、概要

OR、ORI 指令用于并联连接触点。

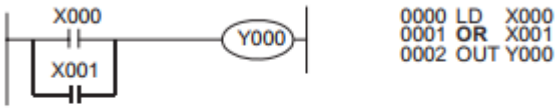
2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
OR	*	*	*	*	*	*	*											*					
ORI	*	*	*	*	*	*	*											*					

3、功能和动作说明

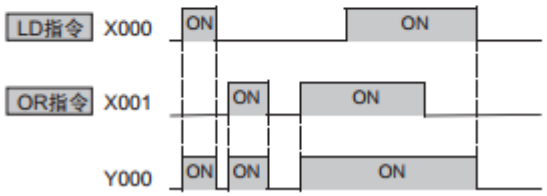
（1）OR 指令（并联 a 触点）

梯形图程序　　指令表程序



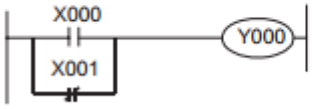

```
0000 LD   X000
0001 OR   X001
0002 OUT Y000
```

时序图



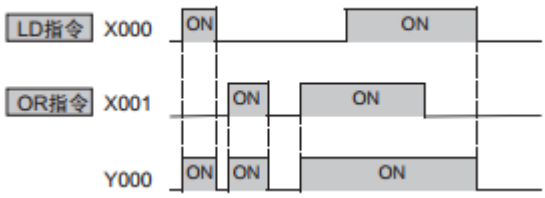
（2）ORI 指令（并联 b 触点）

梯形图程序　　指令表程序




```
0000 LD   X000
0001 OR   X001
0002 OUT Y000
```

时序图



4、注意事项

要注意变址修饰越界的情况，参考 LDI 指令。

2.1.5　LDP、LDF、ANDP、ANDF、ORP、ORF/沿触发类指令

1、概要

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

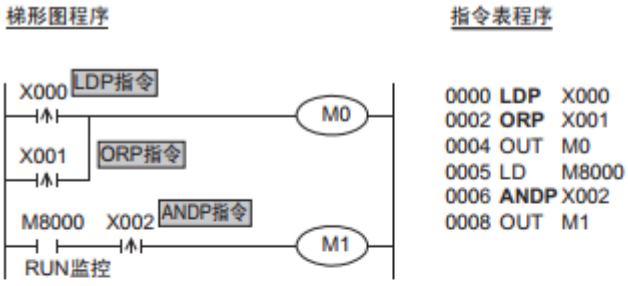
LDF、ANDF、ORF 指令是检测下降沿的触点指令，仅在指定位软元件的下降沿(从 ON 改变到 OFF)时，接通 1 个运算周期。

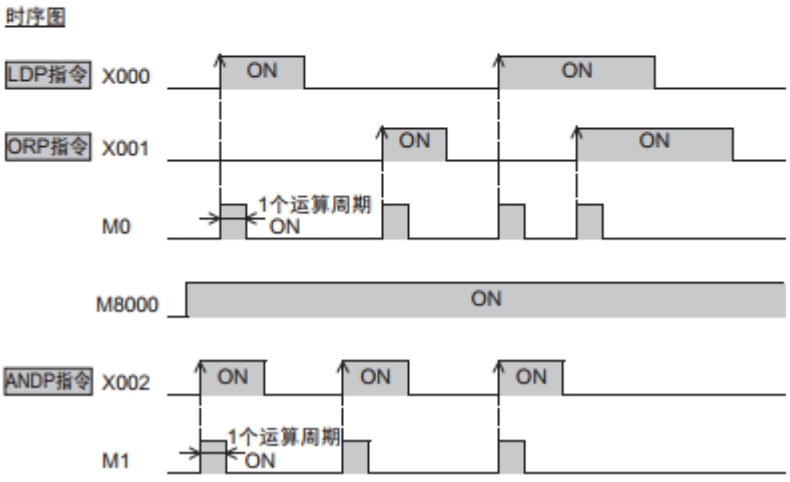
2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
LDP	*	*	*	*	*	*	*																
LDF	*	*	*	*	*	*	*																
ANDP	*	*	*	*	*	*	*																
ANDF	*	*	*	*	*	*	*																
ORP	*	*	*	*	*	*	*																
ORF	*	*	*	*	*	*	*																

3、功能和动作说明

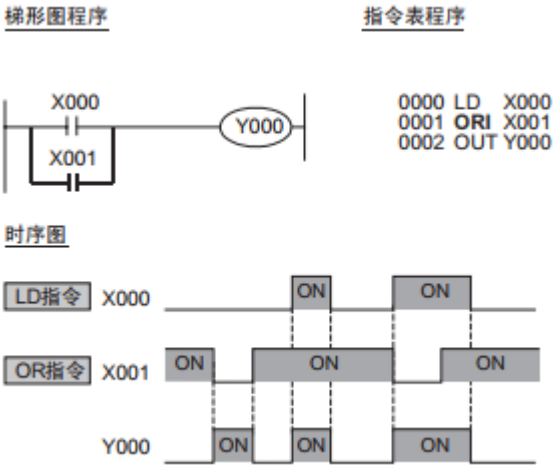
（1）LDP、ANDP、ORP 指令（检测到下升沿时运算开始、串联连接、并联连接）





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

（2）LDF、ANDF、ORF 指令（检测到上升沿时运算开始、串联连接、并联连接）



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

4、注意事项

（1）要注意变址修饰越界的情况，参考 LDI 指令。

（2）脉冲触发类指令一般不适用于中断子程序中。

（3）脉冲触发指令一般不能左右连续使用。

2.1.6 ORB/回路块或指令

1、概要

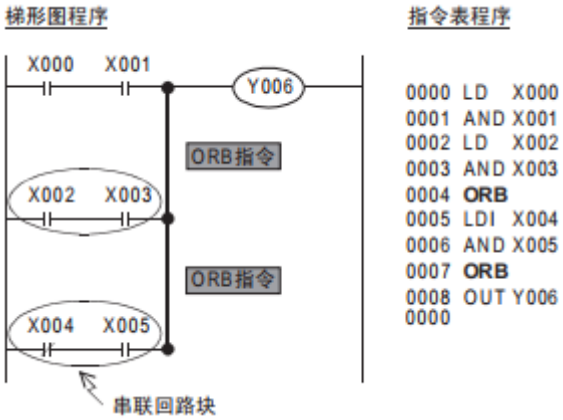
回路块的并联连接。

2、操作数可用元件类型

无。

3、功能和动作说明

并联连接串联回路块时，分支的起点使用 LD、LDI 指令，分支的结束使用 ORB 指令。ORB 指令，与后述的 ANB 指令等相同，都是不带软元件编号的独立指令。有多个并联回路时，在每个回路块中使用 ORB 指令，从而连接。



4、注意事项

ORB 指令可以无限使用，一般都是和 LD、LDI 连用，但是 LD、LDI 指令重复次数在 8 次以下，因此要注意。

2.1.7 ANB/回路块与指令

1、概要

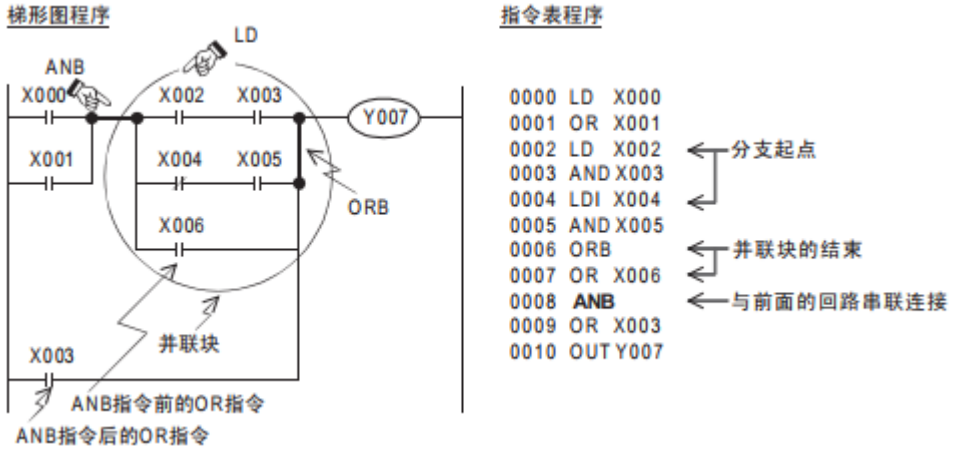
回路块的串联连接。

2、对象软元件

无。

3、功能和动作说明

分支的起点使用 LD、LDI 指令，并联回路块结束后，可以使用 ANB 指令和前面的回路串联连接。有多个并联回路的时候，对每个回路块使用 ANB 指令，从而连接。



4、注意事项

ANB 指令可以无限使用，一般都是和 LD、LDI 连用，但是 LD、LDI 指令重复次数在 8 次以下，因此要注意。

2.1.8 MPS、MRD、MPP/堆栈类指令

1、概要

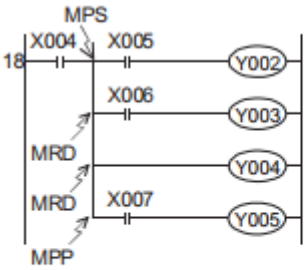
堆栈的内存，用于记忆运算的中间结果(ON 或 OFF)。三个指令分别表示压入堆栈、读取堆栈、弹出堆栈。

2、操作数可用元件类型

略。

3、功能和动作说明

这些指令是用于编写多重分支输出回路的便捷指令。




```
0018 LD   X004
0019 MPS
0020 AND  X005
0021 OUT  Y002
0022 MRD
0023 AND  X006
0024 OUT  Y003
0025 MRD
0026 OUT  Y004
0027 MPP
0028 AND  X007
0029 OUT  Y005
0030 END
```

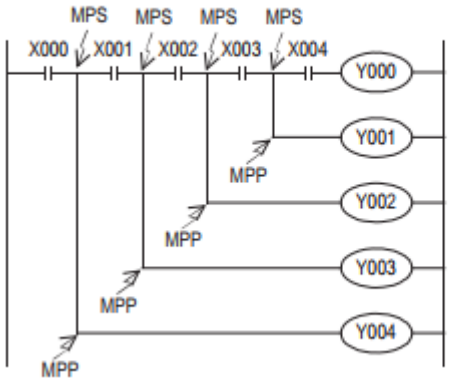
①使用 MPS 指令存储运算的中间结果后，驱动输出 Y002。
②使用 MRD 指令读取该存储内容后，驱动输出 Y003，MRD 指令可以多次编程。
③在最终输出回路中使用 MPP 指令替代 MRD 指令，从而在读出上述存储内容的同时将其复位。

4、注意事项

使用堆栈指令时要注意梯形图的执行顺序，一般情况下（无程序跳转指令和中断发生）都是从上到下，从左到右的顺序执行梯形图。

比如：

梯形图程序



指令表程序

```
0000 LD   X000
0001 MPS
0002 AND  X001
0003 MPS
0004 AND  X002
0005 MPS
0006 AND  X003
0007 MPS
0008 AND  X004
```

当 X1 置 ON 时，如果此时置位 X0，则应该 OUT Y3 先执行，OUT Y4 后执行。

2.1.9 MC、MCR/主控、主控复位指令

1、概要

执行 MC 指令后，母线(LD、LDI 点)移动到 MC 触点之后，开始执行 MC 和 MCR 之间的程序。使用 MCR 指令，可以将其返回原来的母线位置。

2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
MC		*	*																				

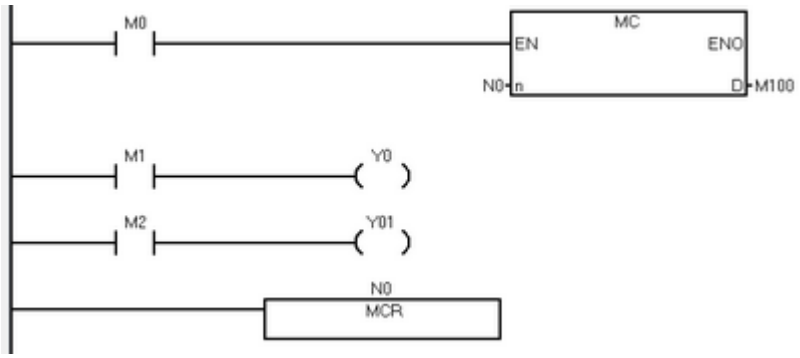
3、功能和动作说明

执行 MC 指令后，母线移动到 MC 触点之后。MC 触点后的母线上连接的驱动指令，只在 MC 指令执行时才执行各个动作，不执行 MC 指令时，无论驱动指令之前的触点状态如何，均为 OFF 执行(与触点 OFF 时的动作相同)。

在使用了主控的回路中，触点指令中如果有不需要触发条件的指令(FOR ～NEXT,EI,DI 指令等)，则执行该指令，而与 MC 指令的执行指令无关。在如下所示的程序举例中，当输入 X000 为 ON 时，则执行从 MC 到 MCR 的指令，但是当 X000 为 OFF 时，各个驱动软元件的动作如下。

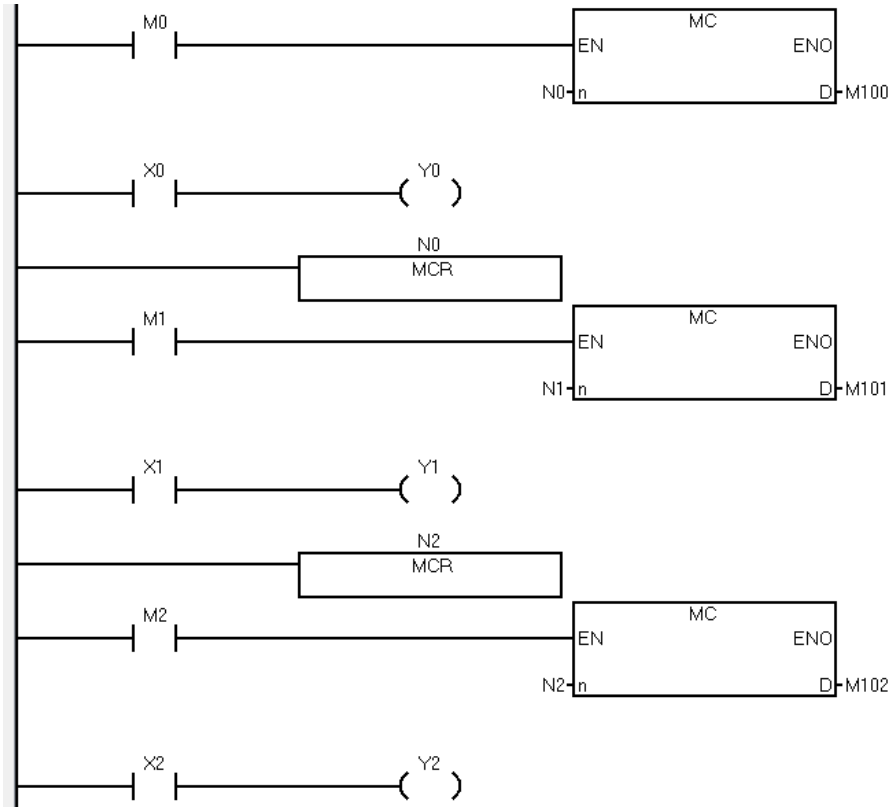
变为 OFF 的软元件 ： 定时器(累计定时器除外)，用 OUT 指令驱动的软元件

保持状态的软元件: 累计定时器、计数器、用 SET/RST 指令驱动的软元件



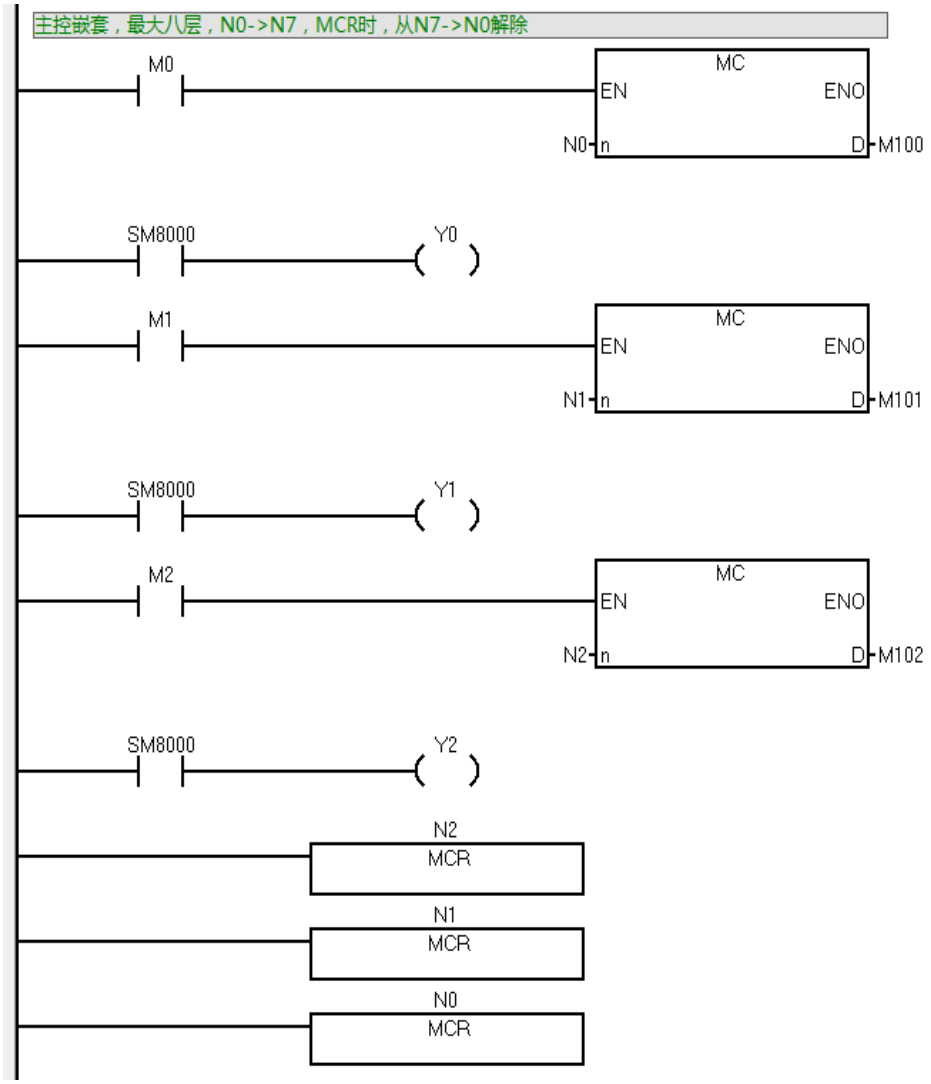
4、注意事项

（1）主控的无嵌套使用



无嵌套时 N 的编号可以任意排列，且使用次数无限制。

（2）主控的嵌套使用



有嵌套时最大允许 8 层嵌套，执行 MC 指令时，嵌套等级 N 编号依次增大，N0->N7，返回时，执行 MCR 指令，从大的嵌套等级开始解除，N7->N0。

2.1.10 INV/反转指令

1、概要

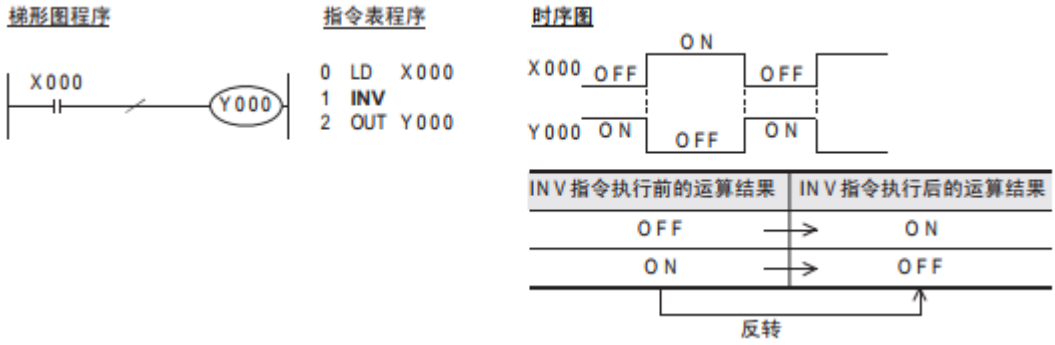
将 INV 指令执行前的运算结果反转的指令，无需指定软元件编号。

2、操作数可用元件类型

无。

3、功能和动作说明

运算结果反转。



4、注意事项

注意 INV 指令的作用范围，是将前面程序块的输出结果进行反转。

2.1.11 MEP、MEF/运算结果脉冲化指令指令

1、概要

MEP、MEF 指令是使运算结果脉冲化的指令，不需要指定软元件编号。

2、对象软元件

无。

3、功能和动作说明

（1）MEP 指令（运算结果的上升沿时为 ON）



（2）MEF 指令（运算结果的下升沿时为 ON）



4、注意事项

注意指令的作用范围，是根据前面程序块的运算结果而动作的。

2.1.12 PLS、PLF/脉冲输出指令

1、概要

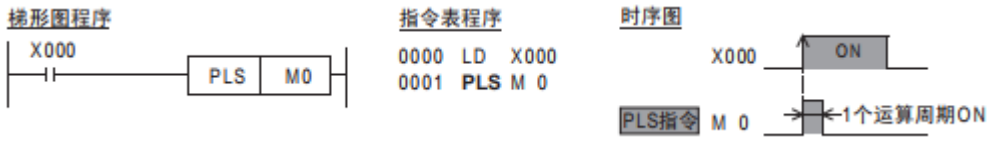
使用 PLS 指令后，仅在驱动输入 ON 以后的 1 个运算周期内，对象软元件动作。
使用 PLF 指令后，仅在驱动输入 OFF 以后的 1 个运算周期内，对象软元件动作。

2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
PLS		*	*															*					
PLF		*	*															*					

3、功能和动作说明

（1）PLS 指令（上升沿的微分输出）



X000 从 OFF 变为 ON 时，只有一个运算周期的 M0 为 ON。

（2）PLF 指令（下升沿的微分输出）



X000 从 ON 变为 OFF 时，只有一个运算周期的 M1 为 ON。

4、注意事项

PLS、PLF 指令不能同时在出梯形图中编程。

2.1.13 SET、RST/置位、复位指令

、概要

（1）SET 置位位元件

（2）RST 复位位元件

（3）RST 清除字元件的当前值

2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
SET		*	*			*	*											*					
RST		*	*	*	*	*	*					*	*	*	*	*	*	*					

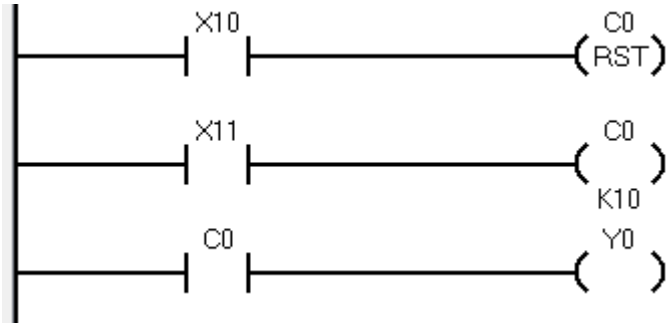
3、功能和动作说明

（1）SET 指令

对输出继电器(Y)、辅助继电器(M)、状态(S)以及数据寄存器(D)的指定位进行线圈驱动的指令。

（2）RST 指令

①复位普通定时器和累计型计数器



由 C0 对 X011 的 OFF→ON 的次数增计数，计数结果达到设定值 K10 的时候，输出触点 C0 动作。此后，即使 X011 从 OFF 变为 ON，计数器的当前值也不改变，输出触点也保持 动作。为了清除这些，恢复输出触点，要使 X010 为 ON。OUT C0 指令后面，需要指定常 数 K 或间接设定用的数据寄存器编号。停电保持(保持)用计数器的场合，即使停电也 能保持当前值以及输出触点的动作状态和复位状态。

②复位高速计数器

高速计数器的复位一般不由 RST 控制，高速有增减计数，当增计数达到设定值时置位 高速计数器，减计数到 0 时则复位高速计数器。

③复位普通为元件或者字元件清 0

4、注意事项

操作特殊寄存器时，指令会执行，但结果不一定会置位或复位，比如 RST SM8000，SM8000 仍然为 ON。

2.1.14 NOP/空指令

1、概要

空操作的指令。

2、操作数可用元件类型

无。

3、功能和动作说明

在一般的指令和指令之间加入 NOP 时，可编程控制器会无视其存在而继续运行。如在程序的中间加入 NOP，当需要更改、增加程序的时候，只需要很小地变动步编号就能实现，但是要求程序有余量。此外，若将已经写入的指令换成 NOP 指令，则回路会发生变化，请务必注意。

4、注意事项

在编程时，NOP 会占用程序步，但是不会影响程序运行。

2.1.15 END/程序结束指令

1、概要

表示程序结束的指令，在梯形图程序最后面，无显示。

2、操作数可用元件类型

无。

3、功能和动作说明

（1）程序结束处理

可编程控制器重复执行输入处理->执行程序->输出处理，若在程序的最后写入 END 指令，则不执行此后剩余的程序步，而直接进行输出处理，FEND 同 END。

（2）重要参数刷新

执行 END 时会刷新看门狗时间，刷新报错寄存器。

4、注意事项

UNISYS 编程时，会隐藏 END 指令，用户不必添加。

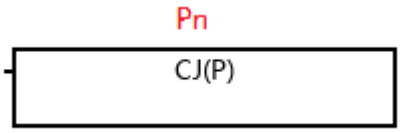
2.2 程序流程指令

使用流程指令可以使梯形图按实际需求进行程序流转。

2.2.1 CJ/程序跳转指令

普通程序跳转，跳转至对应的 P 程序位置。

1、指令格式



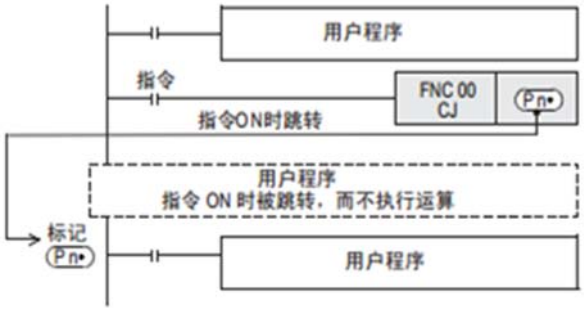
操作数的数据类型：

操作数	内容	类型
D	跳转目标标记的指针编号(P)	指针编号

操作数为指针标号 P0~P4095，其中 P63 为 END 所在步序，不需标记。指针标号允许用变址寄存器修改。CJ 和 CJP 都占 3 个程序步，指针标号占 1 步。

2、功能和动作说明

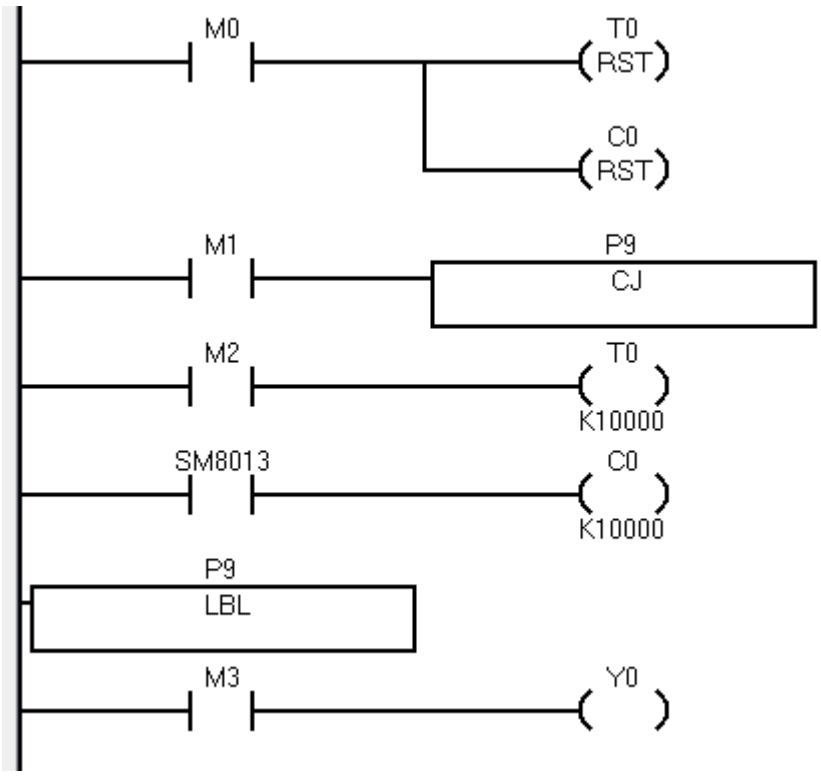
当指令输入为 ON 时，执行指定标记(指针编号)的程序。



3、程序举例

条件跳转指令 CJ 的步骤：

如图所示，当 M1 接通时，则由 CJ　P9 指令跳到标号为 P9 的指令处开始执行，跳过了程序的一部分，减少了扫描周期。如果 M1 断开，跳转不会执行，则程序按原顺序执行。



4、注意事项

（1）在一个程序中一个标号只能出现一次，否则将出错；

（2）在跳转执行期间，即使被跳过程序的驱动条件改变，但其线圈（或结果）仍保持跳转前的状态，因为跳转期间根本没有执行这段程序。

（3）如果在跳转开始前定时器和计数器已在工作，则在跳转执行期间它们将停止工作，到跳转条件不满足后又继续工作。但对于正在工作的高速计数器 C235~C255 或者子程序专用定时器不管有无跳转仍连续工作。

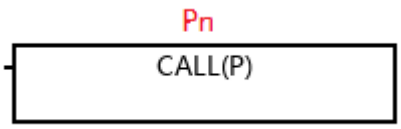
（4）若积算定时器和计数器的复位（RST）指令在跳转区外，即使它们的线圈被跳转，但对它们的复位仍然有效。

（5）跳转指令可以根据特定的需求减少程序周期，跳过不需要运行的程序段。

2.2.2 CALL/子程序调用指令

用于子程序跳转（调用）。

1、指令格式



操作数的数据类型如下表

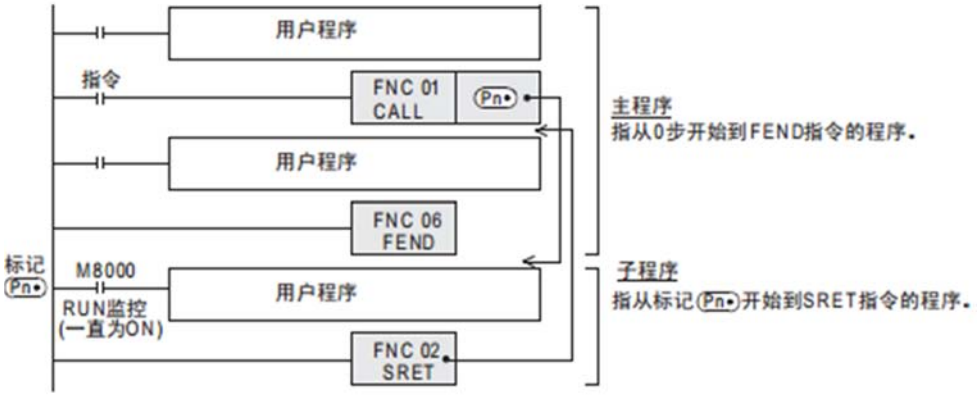
操作数	内容	类型
D	跳转目标标记的指针编号(P)	指针编号

操作数为指针标号 P0~P4095

2、功能和动作说明

当指令输入为 ON 时，执行 CALL 指令，向标记 Pn 的步跳转。接着，执行标记 Pn 的子程序。

执行 SRET 指令后，返回到 CALL 指令的下一步。

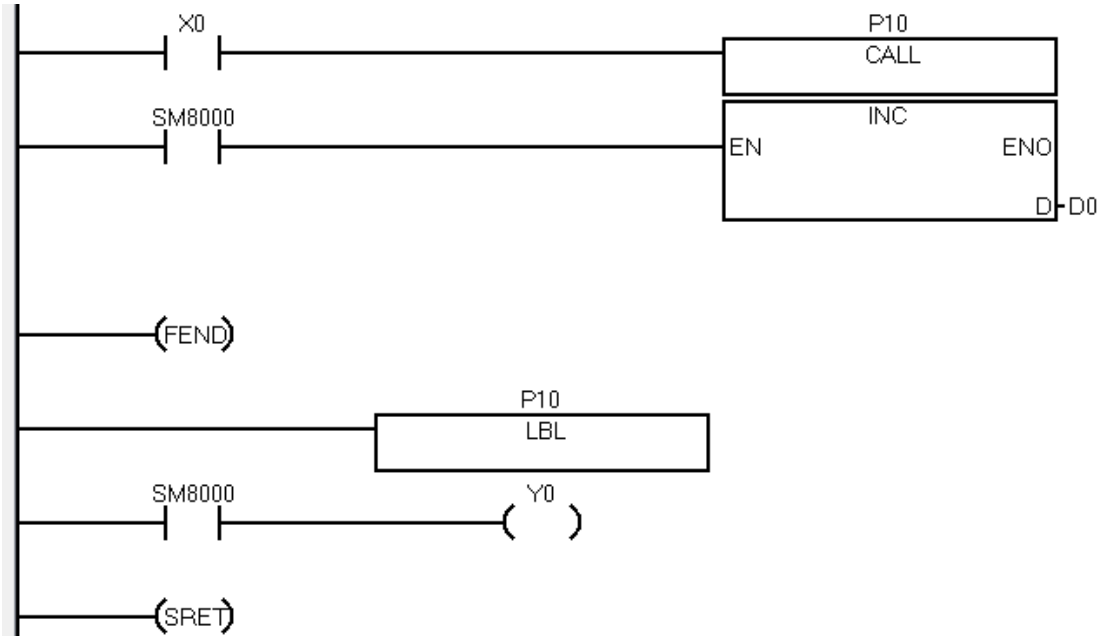


3、程序举例

（1）CALL 无嵌套的使用

主程序：X0 为 ON，则向标记 P10 的步跳转

子程序：执行子程序后，通过执行 SRET 指令，可以返回到原来的步+1 的位置。



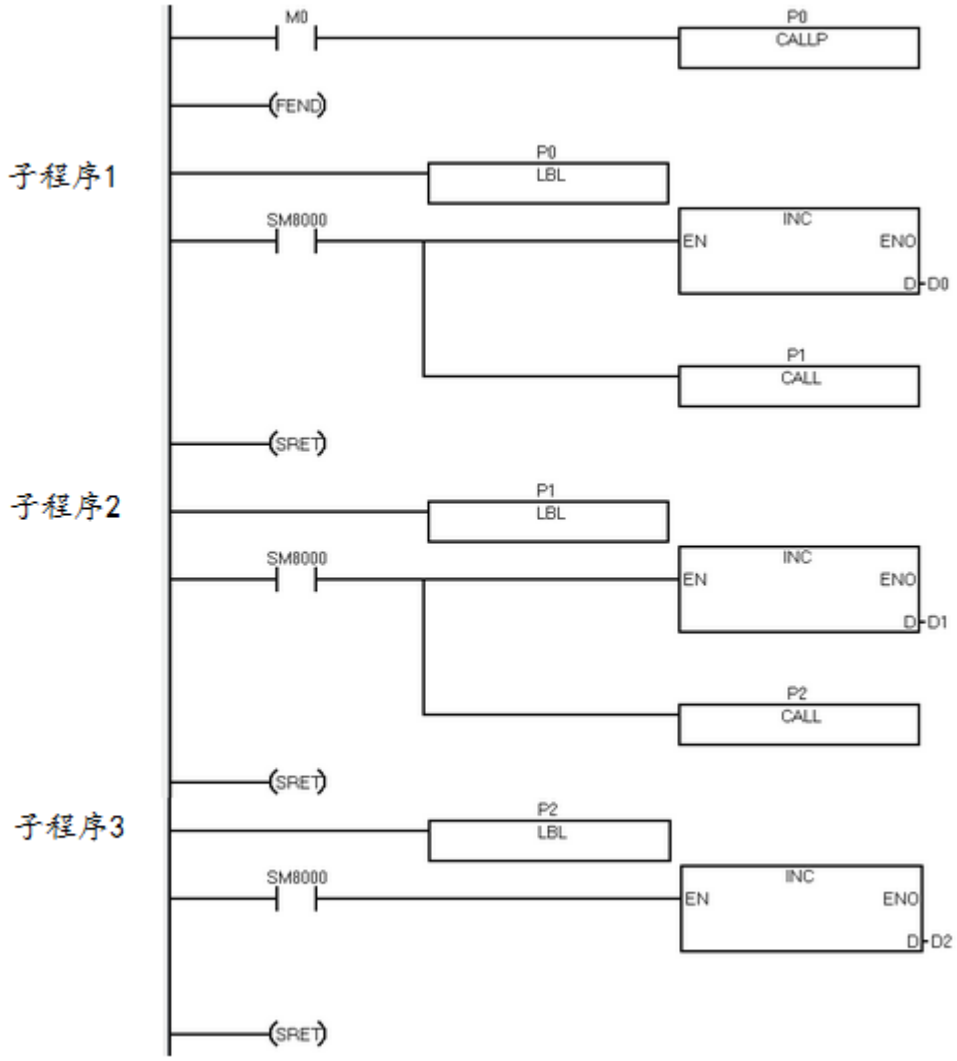
（2）CALL 多层嵌套的使用

主程序：M0 每 OFF→ON 一次，向标记 P0（子程序 1）跳转一次。

子程序 1:执行 INC 指令后 D0 加 1，然后向标记 P1（子程序 2）的步跳转。

子程序 2:执行 INC 指令后 D1 加 1，然后向标记 P2（子程序 3）的步跳转。

子程序 3:执行 INC 指令后 D2 加 1，然后 SRET 返回到子程序 2，然后再返回到子程序 1，然后再返回到 CALLP P0 下方。



4、注意事项

（1）在主程序的最后用 FEND 指令编程，没有 FEND 指令运行报 6637 停机错误。

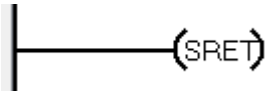
（2）CALL 指令用的标记(P)，在 FEND 指令后编程，否则运行报 6637 停机错误。

（3）CALL 嵌套最多允许 6 层，超过报 6634 停机错误。

2.2.3 SRET/子程序返回指令

使用 CALL 指令调用的子程序，子程序结束后使用 SRET 返回。

1、指令格式



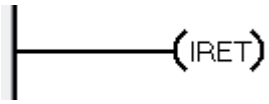
2、功能和动作说明

主要用于 CALL 指令调用后返回，用法参照 CALL 指令中例程。

2.2.4 IRET/中断返回指令

中断程序结束后用 IRET 返回到主程序调用位置的下一步位置。

1、指令格式



2、功能和动作说明

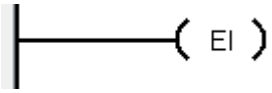
在处理主程序过程中如果产生中断(输入、定时器、计数器)，则跳转到中断(I)程序，然后使用 IRET 指令返回主程序。

用法说明参照 3.2 章节。

2.2.5 EI/允许中断指令

中断允许控制用，在需要使用中断的位置编写此指令。

1、指令格式



2、功能和动作说明

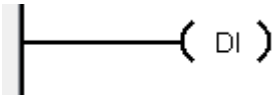
可编程控制器通常为禁止中断状态。使用 EI（FNC4）指令，可以使可编程控制器变为允许中断的状态。使用输入中断和定时器中断，计数器中断功能的时候，请使用该指令。

用法说明参照 3.2 章节。

2.2.6 DI/禁止中断指令

中断禁止控制用，在不需要中断响应的位置编写此指令，使得中断在 EI 允许的位置处执行。

1、指令格式



2、功能和动作说明

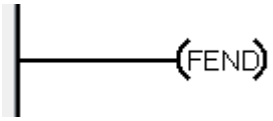
在改为允许中断后，使用 EI 指令可以再次更改为禁止中断。

用法说明参照 3.2 章节。

2.2.7 FEND/主程序结束指令

主程序结束专用指令。

1、指令格式



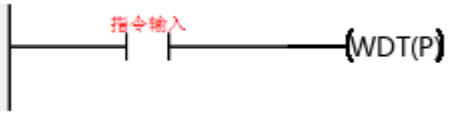
2、功能和动作说明

FEND 指令是主程序结束命令，功能同 END 指令，只是作为主程序专用的结束指令，在编写子程序和中断程序时需要使用 FEND 指令。

2.2.8 WDT/看门狗定时器指令

通过此指令刷新看门狗时间。

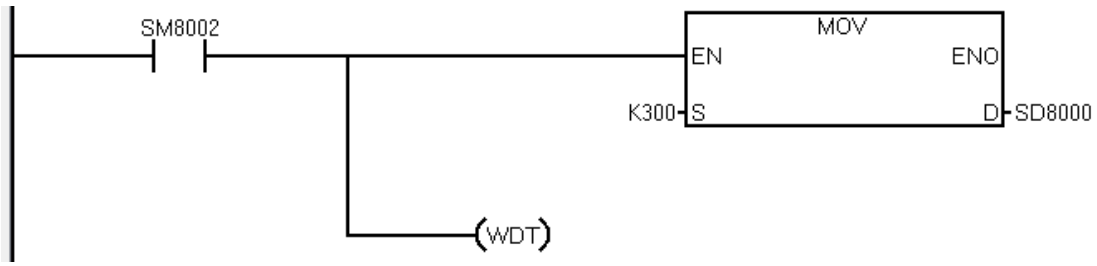
1、指令格式



2、功能和动作说明

可编程控制器的运算周期(0～END 或 FEND 指令的执行时间)如要超出 200ms 时，可编程控制器会出现看门狗定时器错误(检测出运算异常)，然后 ERROR(ERR) LED 灯亮后停止。类似这样的运算周期较长的情况，在程序中间插入 WDT 指令，可以避免出现这样的错误。

3、程序举例



设置看门狗定时器时间 300ms；

看门狗定时器刷新：没有编写 WDT 指令的时候，END 处理时 SD8000 的值变为有效。

4、注意事项

避免看门狗定时器超时的方法：

（1）控制梯形图运行周期。

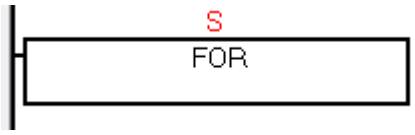
（2）梯形图中 FOR、NEXT 这类流程指令使用次数不要太多。

（3）当梯形图中有运行起来比较耗时的程序时，提前刷新看门狗定时器或者更改看门狗定时器的时间。

2.2.9 FOR/循环范围开始指令

用于程序的重复执行操作。

1、指令格式



操作数说明

操作数	内容	类型
S	FOR~NEXT 指令之间的重复次数(1~32767,-32768~0 时，作为 1 处理)	BIN16 位

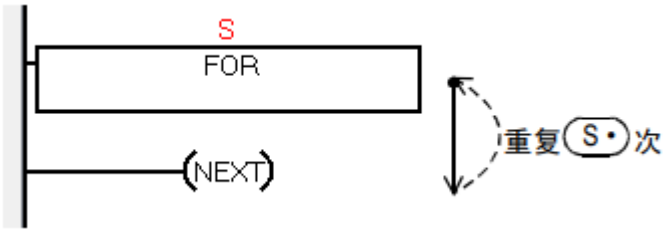
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D								*	*	*	*	*	*	*	*	*	*	*	*	*			

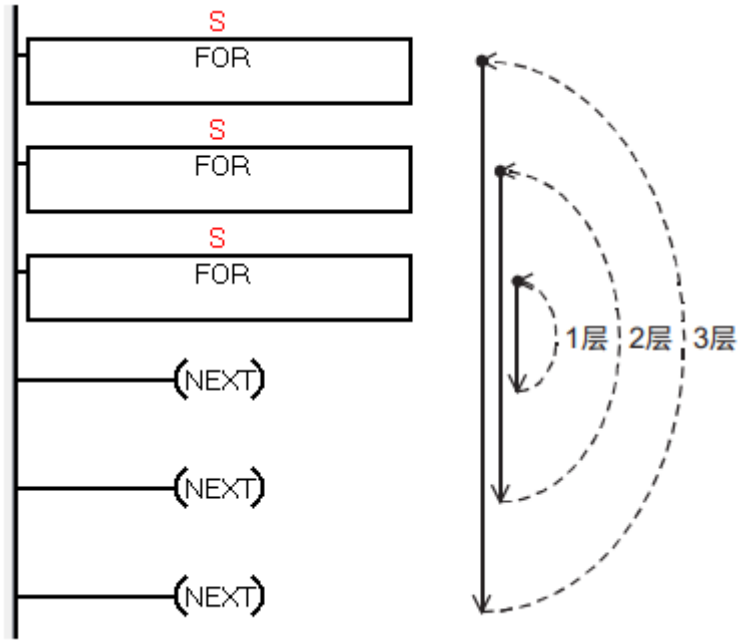
2、功能和动作说明

从 FOR(FNC 08)指令开始到 NEXT 指令之间的程序按指定次数重复运行。

（1）单层 FOR-NEXT



（2）多层 FOR-NEXT 嵌套



3、程序举例

参考上面的功能动作说明。

4、注意事项

（1）FOR-NEXT 嵌套使用最多允许 5 层，超出报 6620 停机错误。

（2）使用多层嵌套时注意看门狗超时时间，可能会引发超时，必要时要刷新看门狗时间。

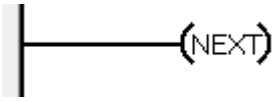
（3）FOR-NEXT 对应个数不匹配会报 6619 停机错误。

（4）FOR 后面的次数是指一个梯形图扫描周期改循环执行的次数。

2.2.10 NEXT/循环范围结束指令

用于 FOR 循环的程序段结束。

1、指令格式



2、功能和动作说明

和 FOR 指令结合使用，作为循环的结束，具体使用方法参考 FOR 指令。

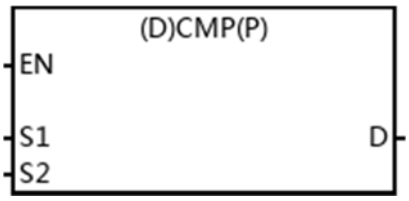
2.3 传送比较指令

这一章节中的指令是使用应用指令时最为重要的数据传送和比较等基本的数据操作指令。

2.3.1 CMP/比较指令

比较 2 个值，根据结果置位输出位元件。

1、指令格式



操作数说明

操作数	内容	类型
S1	比较值	BIN16/32 位
S2	比较源	BIN16/32 位
D	输出比较结果的起始位元件	位

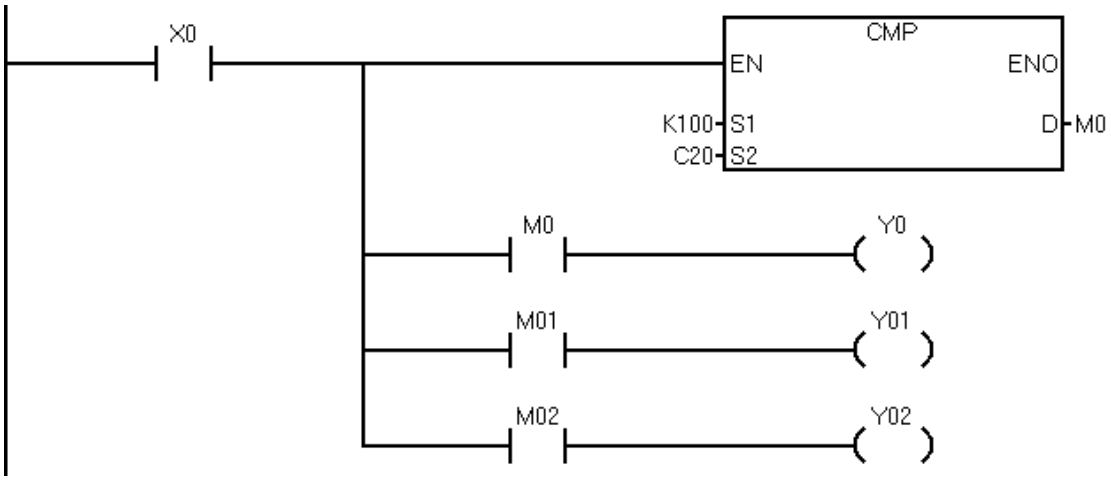
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*	*			*												*					

2、功能和动作说明

对比较值 S1 和比较源 S2 的内容进行比较，根据其结果(小、一致、大)，使 D、D+1、D+2 中的一个为 ON。

3、程序举例



C20 当前值 < 100 时，M0 置 ON，Y0 点亮；

C20 当前值 = 100 时，M1 置 ON，Y1 点亮；

C20 当前值 > 100 时，M2 置 ON，Y2 点亮。

4、注意事项

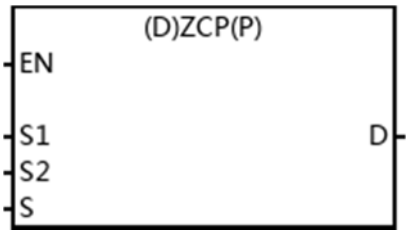
（1）D 中指定的软元件个数是连续的三个，注意不要与其他正在使用的软元件重复。

（2）即使是指令输入为 OFF,CMP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

2.3.2 ZCP/区间比较指令

针对 2 个值构成的区间，将比较源的值与此区间比较，根据结果(区间左侧、区间内，区间右侧)置位输出位软元件。

1、指令格式



操作数说明

操作数	内容	类型
S1	左区间的比较值	BIN16/32 位
S2	右区间的比较值	BIN16/32 位
S	比较源	BIN16/32 位
D	输出比较结果的起始位软元件	位

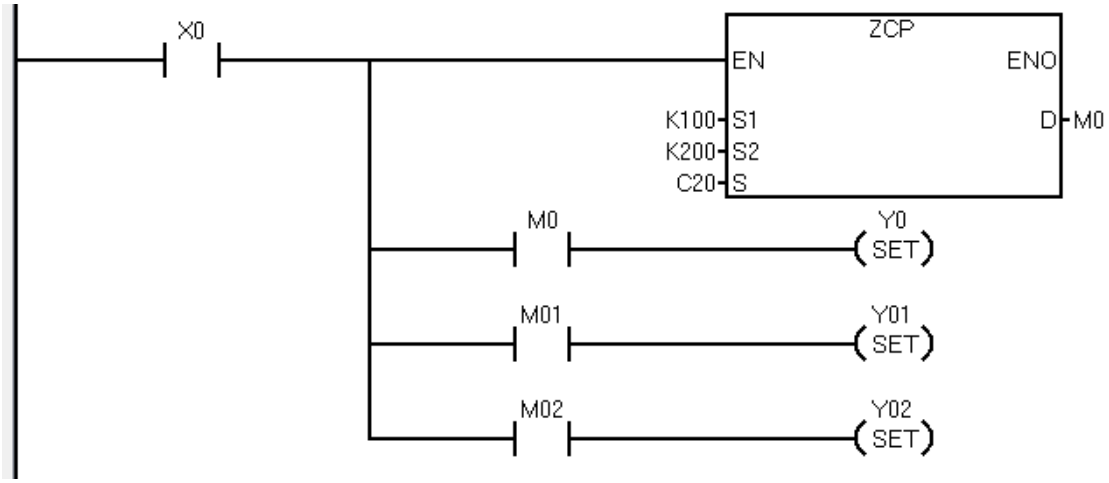
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*	*			*												*					

2、功能和动作说明

将比较源 S 的内容与左区间比较值 S1 和右区间比较值 S2 进行比较，根据其结果(区间左侧、区间内，区间右侧)，使 D、D+1、D+2 其中一个为 ON。

2、程序举例



C20 当前值 < 100 时，M0 置 ON，Y0 点亮；

100 ≤ C20 当前 ≤ 120 时，M1 置 ON，Y1 点亮；

C20 当前值 > 120 时，M2 置 ON，Y2 点亮。

3、注意事项

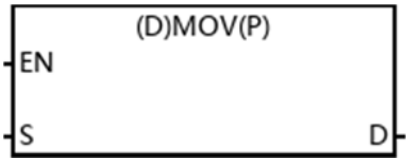
（1）D 中指定的软元件个数是连续的三个，注意不要与其他正在使用的软元件重复。

（2）即使是指令输入为 OFF,CMP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

2.3.3 MOV/传送指令

将源软元件的内容传送(复制)到目的软元件中的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	传送源	BIN16/32 位
D	传送目标	BIN16/32 位

操作数可用元件类型

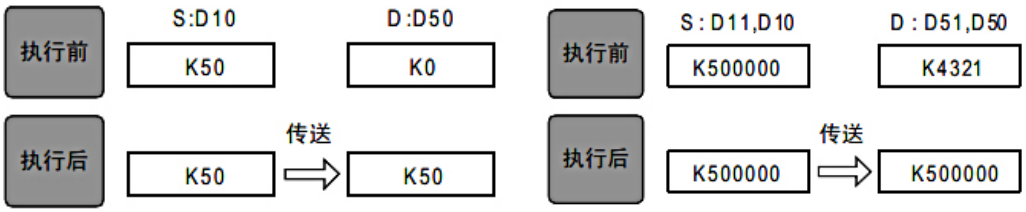
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将传送源 S 的 1 点数据内容传送给目标 D。

3、程序举例

（1）传字软元件



（2）传组合位元件



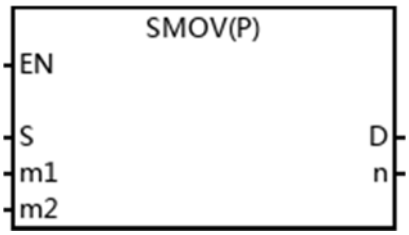
4、注意事项

如果传送的是组合位元件，16 位指令最多传送 16 个（4 的倍数）位软元件，32 位指令最多传送 32 个（4 的倍数）位软元件。

2.3.4 SMOV/位移动指令

以位数为单位(4 位)进行数据分配合成的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	进行位移动的数据	BIN16 位
m1	移动的起始位的位置	BIN16 位
m2	移动的位的个数	BIN16 位
D	位移动后的目标数据	BIN16 位
n	移动目标的起始位的位置	BIN16 位

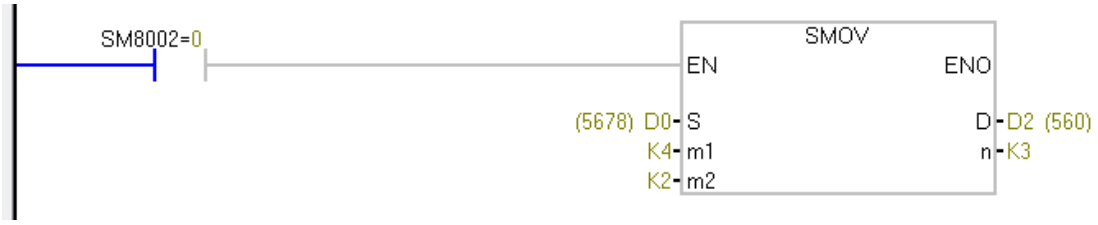
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*					
m1																			*	*			
M2																			*	*			
D									*	*	*	*	*	*	*	*	*	*					
n																			*	*			

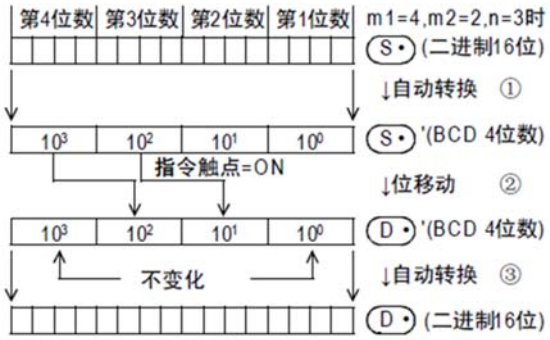
2、功能和动作说明

传送源 S 和传送目标 D 的内容转换(0000～9999)成 4 位数的 BCD，m1 位数起的低 m2 位数部分被传送(合成)到传送目标 D 的 n 位数起始处，然后转换成 BIN，保存在传送目标 D 中。

3、程序举例



D0	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	5678
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	560



① (S·)从BIN转换为BCD。

②从第 m1 位数起的低 m2 位数部分的数据，被传送（合成）到(D·)'的第 n 位数起始 m2 位数。
(D·)'的10^3位数，被10^0位数在执行来自(S·)'的传送时不受任何影响。

③合成的数据(BCD)转换成BIN后，保存到(D·)中。

4、注意事项

（1）原数据取值范围是 0~9999。

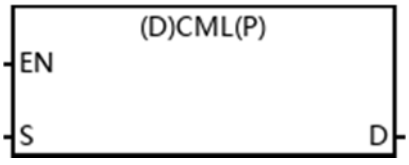
（2）m1，m2，n 的范围是 1~4。

（3）将 SM8090 置 ON 后，执行 SMOV 指令时，则不能进行 BIN→BCD 转换。位移动以 4 位为单位执行。

2.3.5 CML/反转指令

将源数据的所有位取反然后进行传送的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	执行反转的原数据	BIN16/32 位
D	反转后目标数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S 中指定的软元件的各位反转(0→1,1→0)后，传送至 D。

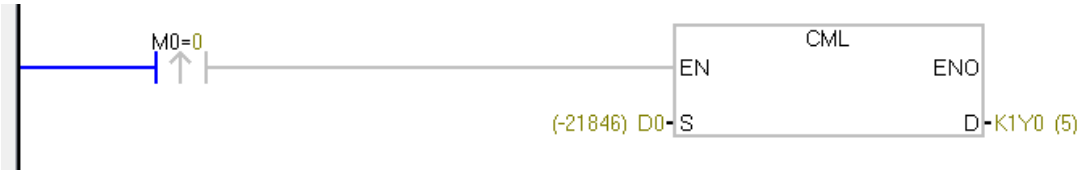
3、程序举例



D0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1	12345
D1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0	-12346

4、注意事项

指定的是位组合元件时，根据实际的位数来反转传送，但是这个位数不能超过指令位数且为 4 的倍数，下面的例子是指定位数为 4 的情况。



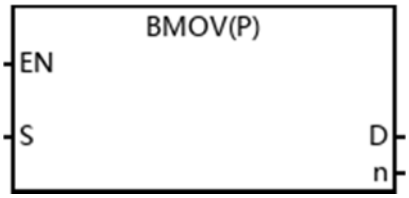
软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	-21846

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
Y0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5

2.3.6 BMOV/成批传送指令

对指定点数的多个数据进行成批传送(复制)。

1、指令格式



操作数说明

操作数	内容	类型
S	传送源	BIN16
D	传送目标	BIN16
n	传送点数[1≤n≤512]	BIN16

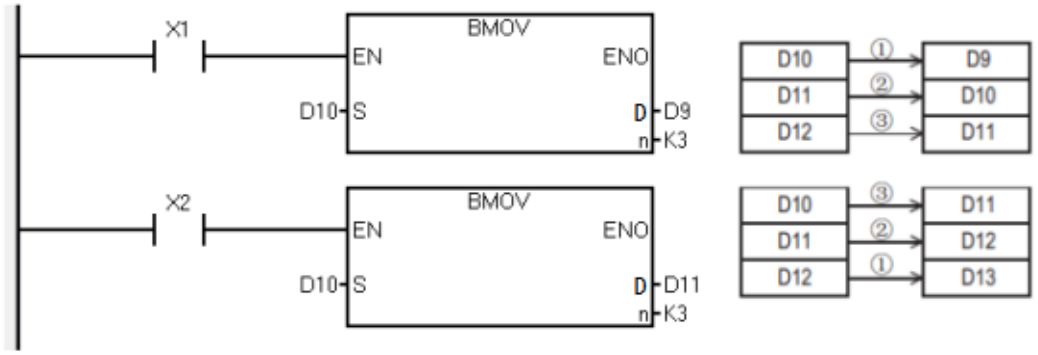
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n														*					*	*			

2、功能和动作说明

将 S 开始的 n 点的数据成批传送到 D 开始的 n 点中。超出软元件编号范围时，在可能的范围内传送。

3、程序举例

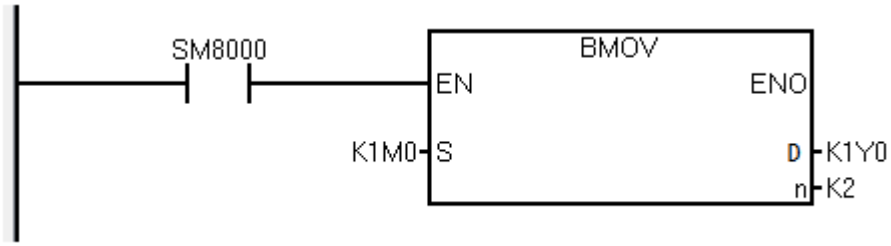


4、注意事项

（1）无论传送源的数据有无传送，为了防止数据源没有传送就被改写，采用编号重叠的方法，按上面的①～③的顺序自动传送。

（2）双向传送功能，当 SM8090 标志位置 ON 时，传送方向变为 D->S。

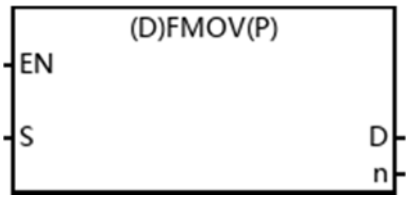
（3）带有位数指定的位软元件的情况下，S 和 D 要采用相同的位数，例如



2.3.7 FMOV/多点传送指令

将同一数据传送到多点软元件中的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	传送源	BIN16/32 位
D	传送目标的起始字软元件	BIN16/32 位
n	传送点数[1≦n≦512]	BIN16

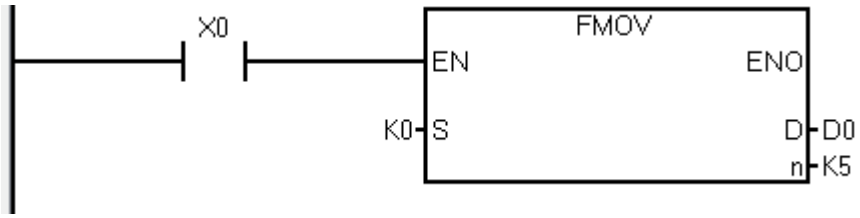
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*			*					
n																			*	*			

2、功能和动作说明

将 S 的内容传送到以 D 起始的 n 点的软元件中，n 点的软元件内容都相同，以 n 指定的个数超出了软元件编号范围时，在可能的范围内传送。

3、程序举例



执行前

D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3
D3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4
D4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5

执行后

D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4、注意事项

n 指定的个数超出了软元件编号范围时，在可能的范围内传送。

2.3.8 XCH/交换指令

在 2 个软元件之间进行数据交换

1、指令格式



操作数说明

操作数	内容	类型
D1	要交换数据	BIN16/32 位
D2	要交换数据	BIN16/32 位

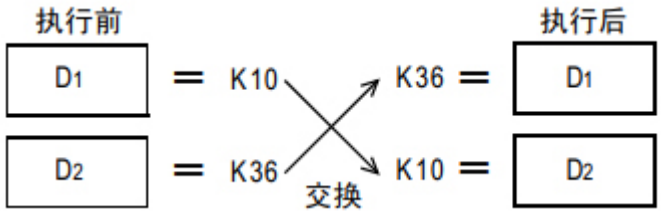
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D1									*	*	*	*	*	*	*	*	*	*					
D2									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

D1 和 D2 相互之间进行数据交换。

3、



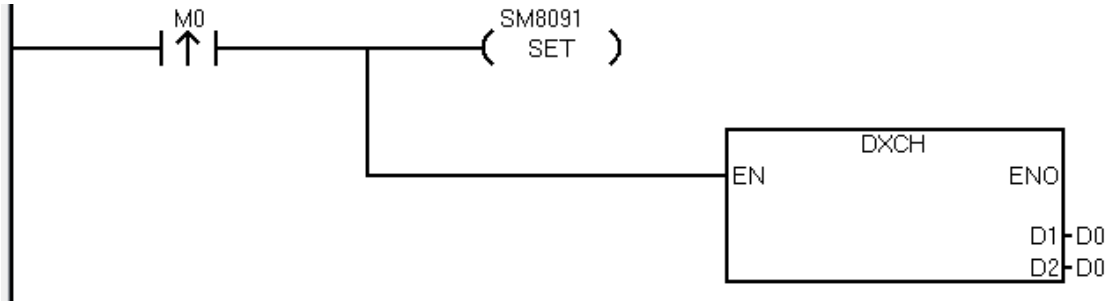
程序举例

略。

4、注意事项

（1）在 SM8091 为 ON 的状态下执行指令时，交换字软元件的高 8 位(字节)和低 8 位(字节)。该动作和 SWAP 指令为相同的动作。

（2）32 位运算时，交换各个字软元件的高 8 位(字节)和低 8 位(字节)。



执行前

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	5678
D1	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1234

执行后

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	7856
D1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	3412

（3）SM8091 为 ON 时，D1 和 D2 的软元件编号不一致的时候会报 6706 运算错误。

2.3.9 BCD/BCD 转换指令

将 BIN(2 进制数)转换成 BCD(10 进制数)后传送的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	源(2 进制数)数据	BIN16/32 位
D	目标(10 进制数)数据	BIN16/32 位

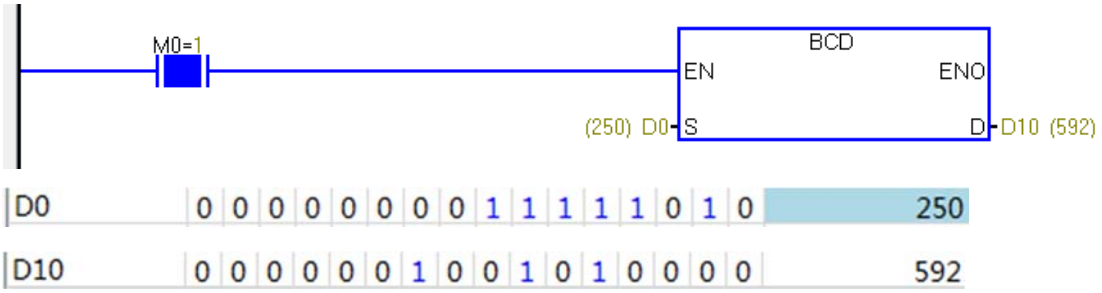
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*					
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S 的 BIN(2 进制数)数据转换成 BCD(10 进制数)数据后传送到 D 中。16 位指令时 S 的数据可以转换成 K0~K9999 的 BCD(10 进制数)，32 位指令时 S 的数据可以转换成 K0~K99999999 的 BCD(10 进制数)。

3、程序举例



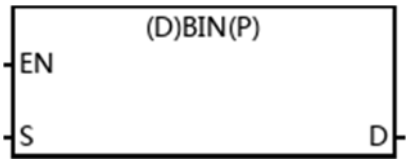
4、注意事项

注意源操作数的取值范围。

2.3.10 BIN/BIN 转换指令

将 10 进制数(BCD)转换成 2 进制数(BIN)的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	源数据(10 进制数)	BIN16/32 位

D	目标数据(2 进制数)	BIN16/32 位

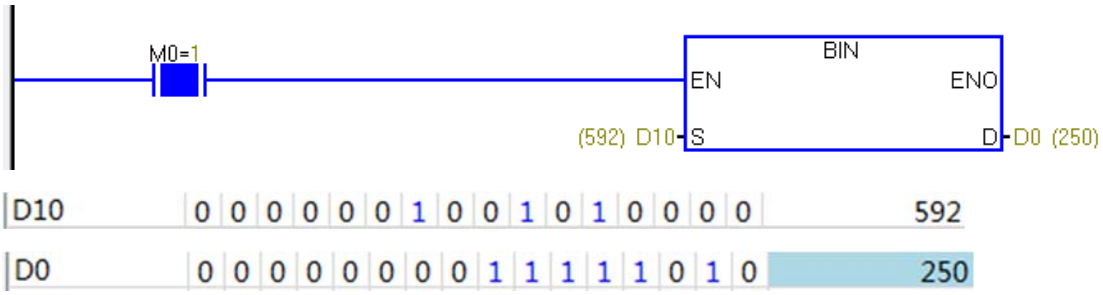
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*					
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S 的 BCD(10 进制数)数据转换成 BIN(2 进制数)数据后传送到 D 中。16 位指令时 S 可以在 0～9999(BCD)的范围内转换，32 位指令时 S 可以在 0～99999999(BCD)的范围内转换。

3、程序举例



D10 写入数据 592，592 对应的是 250 的 BCD 数据，250 转换为 2 进制数为 1111 1010。

4、注意事项

注意源操作数的取值范围。

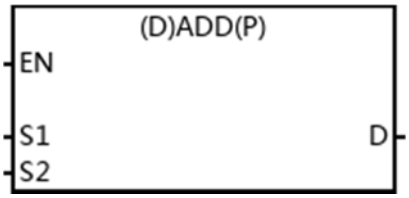
2.4 四则逻辑运算指令

用于基本的加减乘除、逻辑运算。

2.4.1 ADD/加法运算指令

将两个数据进行加法运算的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	进行加法运算的数据 1	BIN16/32 位
S2	进行加法运算的数据 2	BIN16/32 位
D	加法运算结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S1 和 S2 的内容进行二进制加法运算后传送到 D 中。

3、程序举例

无

4、注意事项

标志位的动作及数值的正负的关系：

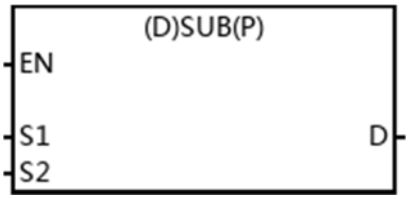
软元件	名称	内容
SM8080	零位	ON：运算结果为 0 时 OFF：运算结果为 0 以外时
SM8081	借位	ON：运算结果小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时，借位标志位动作。

		OFF：运算结果不小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时
SM8082	进位	ON：运算结果大于 32,767(16 位运算) 或者 2,147,483,647(32 位运算) 时，进位标志位动作。 OFF：运算结果不大于 32,767(16 位运算)或者 2,147,483,647(32 位运算)时

2.4.2 SUB/减法运算指令

将两个数据进行减法运算的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	进行减法运算的数据 1	BIN16/32 位
S2	进行减法运算的数据 2	BIN16/32 位
D	减法运算结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S1 和 S2 的内容进行二进制减法运算后传送到 D 中。

3、程序举例

略。

4、注意事项

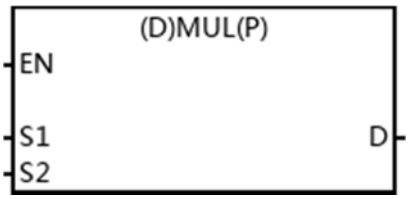
标志位的动作及数值的正负的关系：

软元件	名称	内容
SM8080	零位	ON：运算结果为 0 时 OFF：运算结果为 0 以外时
SM8081	借位	ON：运算结果小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时，借位标志位动作。 OFF：运算结果不小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时
SM8082	进位	ON：运算结果大于 32,767(16 位运算) 或者 2,147,483,647(32 位运算) 时，进位标志位动作。 OFF：运算结果不大于 32,767(16 位运算)或者 2,147,483,647(32 位运算)时

2.4.3 MUL/乘法运算指令

将两个数据进行乘法运算的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	进行乘法运算的数据 1	BIN16/32 位
S2	进行乘法运算的数据 2	BIN16/32 位
D	乘法运算结果	BIN32/64 位

操作数可用元件类型

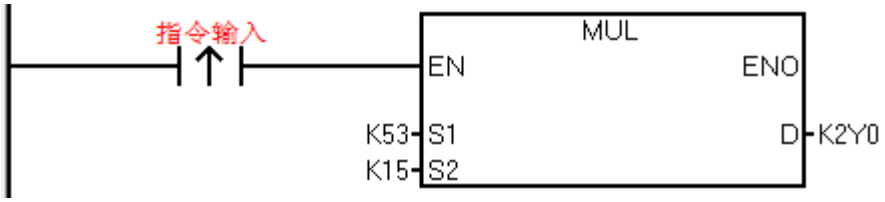
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

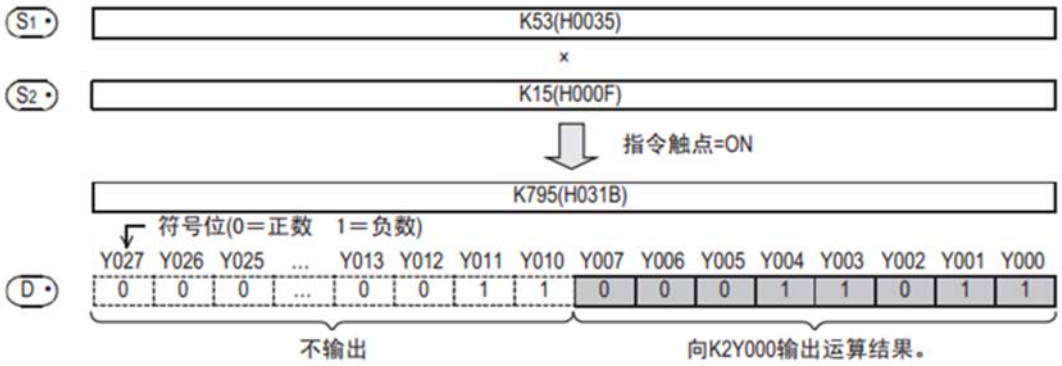
2、功能和动作说明

16 位指令时，将 S1 和 S2（S1、S2 均为单字）的内容进行二进制乘法运算后传送到 D 开始的 32 位数据中。

32 位指令时，将 S1 和 S2（S1、S2 均为双字）的内容进行二进制乘法运算后传送到的 64 位(四个连续的字软元件)中。

3、程序举例





（D+1，D）指定位数(K1～8)时，例如，指定 K2 时，只能得到乘积(32 位)中的低 8 位。

4、注意事项

（1）使用组合位时，得到的远算结果为位数 x4。

（2）当运算结果为 0 时，SM8083 置 ON，其他结果为 OFF。

（3）进行 32 位指令运算时，目的操作数不能为 Z 寄存器。

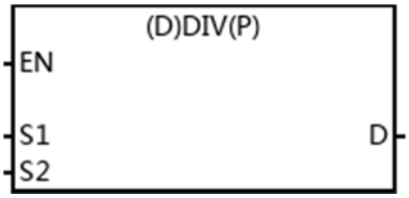
（4）32 位指令计算出来得到的 64 位数据上位机并不能直接监控，只能通过多个寄存器自己组合出结果，要想更直观的显示，建议使用浮点数运算。

（5）注意 16 位指令和 32 位指令，操作数代表的实际寄存器范围

2.4.4 DIV/除法运算指令

将两个数据进行除法运算的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	被除数	BIN16/32 位
S2	除数	BIN16/32 位
D	除法运算结果	BIN32/64 位

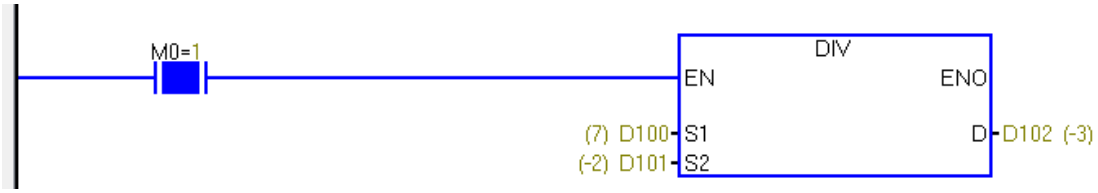
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S1 和 S2 的内容进行二进制除法运算后，商传送到 D 中，余数传送到 D+1 中。16 位指令时，D 占用 2 个字元件，32 位指令时，D 占用 4 个字元件。

3、程序举例



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D100	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	7
D101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-2
D102	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	-3
D103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

4、注意事项

（1）有关运算结果：商和余数的最高位显示位正(0)、负(1)的符号。当被除数或除数其一为负时，商为负。当被除数为负时，余数为负。

（2）通过指定位数来指定位软元件时，不能得出余数。使用 32 位运算(DDIV、DDIVP)时，不能指定使用 Z 寄存器。

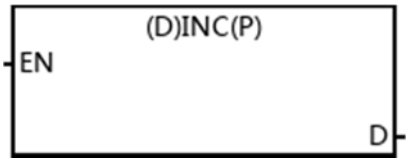
（3）除数 S2 为 0 时，发生运算错误，运算结果超出指令位数对应的值时，也会报错，且进位标志置 ON。

软元件	名称	内容
SM8083	零位	ON：运算结果为 0 时 OFF：运算结果为 0 以外时
SM8084	进位	ON ：运算结果超过 32,767(16 位运算)或者 2,147,483,647(32 位运算)时，进位标志位动作。 OFF ：运算结果为 32,767(16 位运算)或者 2,147,483,647(32 位运算)以下时

2.4.5 INC/加一指令

数据自增一指令。

1、指令格式



操作数说明

操作数	内容	类型

D	进行自增一的数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 D 的内容加一后传送到 D 中。

3、程序举例

略。

4、注意事项

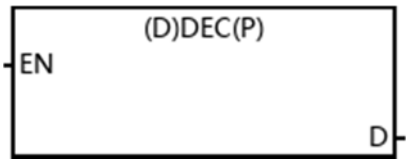
（1）16 位运算：+32,767 上加 1 后，变为-32,768，但是标志位(零、借位、进位)不动作。

（2）32 位运算：+2,147,483,647 上加 1 后，变为-2,147,483,648，但是标志位(零、借位、进位)不动作。

2.4.6 DEC/减一指令

数据自减一指令。

1、指令格式



操作数说明

操作数	内容	类型
D	进行自减一的数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

D 的内容减一运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

（1）16 位运算：-32,768 上减一后，变为+32,767，但是标志位(零、借位、进位)不动作。

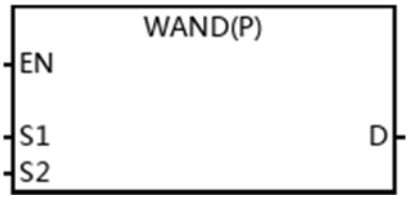
（2）32 位运算：-2,147,483,648 上减一后，变为+2,147,483,647，但是标志位(零、借位、进位)不动作。

2.4.7 WAND/逻辑与指令

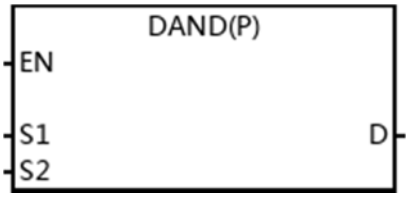
两个数据进行逻辑与运算的指令。

1、指令格式

16 位形式：



32 位形式：



操作数说明

操作数	内容	类型

S1	进行逻辑与运算的数据 1	BIN16/32 位
S2	进行逻辑与运算的数据 2	BIN16/32 位
D	逻辑与结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

S1 和 S2 的内容以各位为单位进行逻辑与运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

逻辑与运算以位位单位，如下表中(1∧1=1　0∧1=0　1∧0=0　0∧0=0)所示变化。

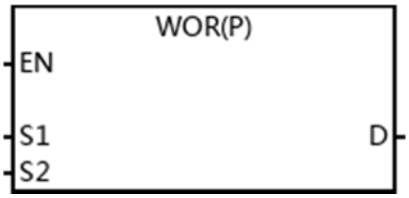
	S1	S2	D
位单位的逻辑运算	0	0	0
	1	0	0
	0	1	0
	1	1	1

2.4.8 WOR/逻辑或指令

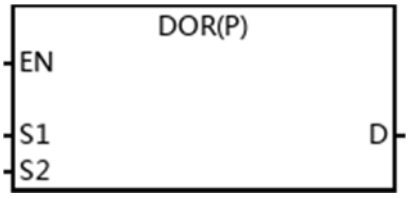
两个数据进行逻辑或运算的指令。

1、指令格式

16 位形式：



32 位形式：



操作数说明

操作数	内容	类型
S1	进行逻辑或运算的数据 1	BIN16/32 位
S2	进行逻辑或运算的数据 2	BIN16/32 位
D	逻辑或结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

S1 和 S2 的内容以各位为单位进行逻辑或运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

逻辑或运算以位为单位，如下表中(1∨1=1　0∨1=1　0∨0=0　1∨0=1)所示变化。

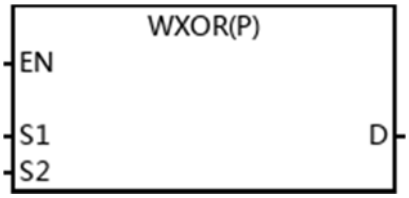
	S1	S2	D
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	1

2.4.9 WXOR/逻辑异或指令

两个数据进行逻辑异或运算的指令。

1、指令格式

16 位形式：



32 位形式：



操作数说明

操作数	内容	类型
S1	进行逻辑异或运算的数据 1	BIN16/32 位
S2	进行逻辑异或运算的数据 2	BIN16/32 位
D	逻辑异或结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

S1 和 S2 的内容以各位为单位进行逻辑异或运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

逻辑异或运算以位为单位，如下表中(1∀1=0　0∀0=0　1∀0=1　0∀1=1)所示变化。

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

2.4.10 NEG/补码指令

求出数值的 2 进制补码(各位反转+1 后的值)的指令。使用这个指令后，可以反转数值的符号。

1、指令格式



操作数说明

操作数	内容	类型
D	求补码的数据或运算结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

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

3、程序举例

略。

4、注意事项

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

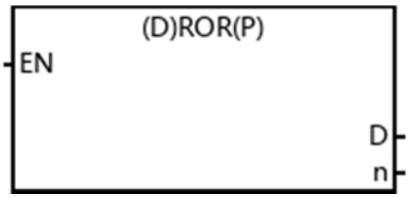
2.5 循环移位指令

使位数据或字数据按指定的方向或位数进行移动。

2.5.1 ROR/循环右移指令

使指定位数部分的位信息右移、循环的指令，不带进位标志。

1、指令格式



操作数说明

操作数	内容	类型
D	进行循环右移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

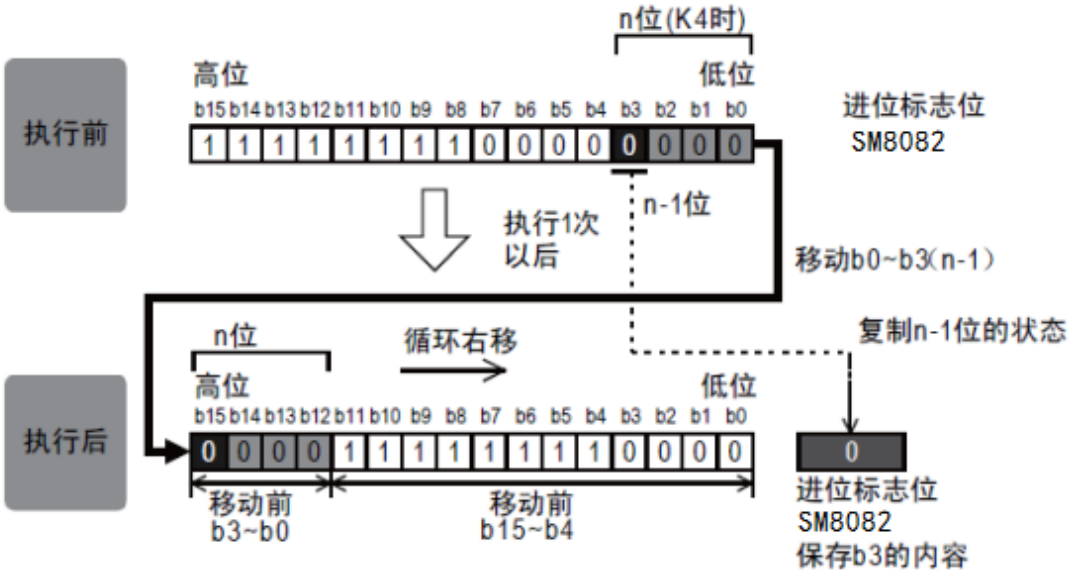
2、功能和动作说明

D 的所有位数中的 n 位循环右移。最后的位保存在进位标志位(SM8082)中。

3、程序举例

将 D0 = -256 = 0xFF00 进行 4 位循环右移





4、注意事项

（1）相关软元件：SM8082，进位标志，最后从最低位移出的位为 1 时置 ON。

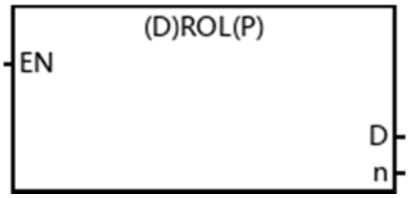
（2）连续执行型(ROR、DROR)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3）在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0) 。

2.5.2 ROL/循环左移指令

使指定位数部分的位信息左移、循环的指令，不带进位标志。

1、指令格式



操作数说明

操作数	内容	类型
D	进行循环左移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

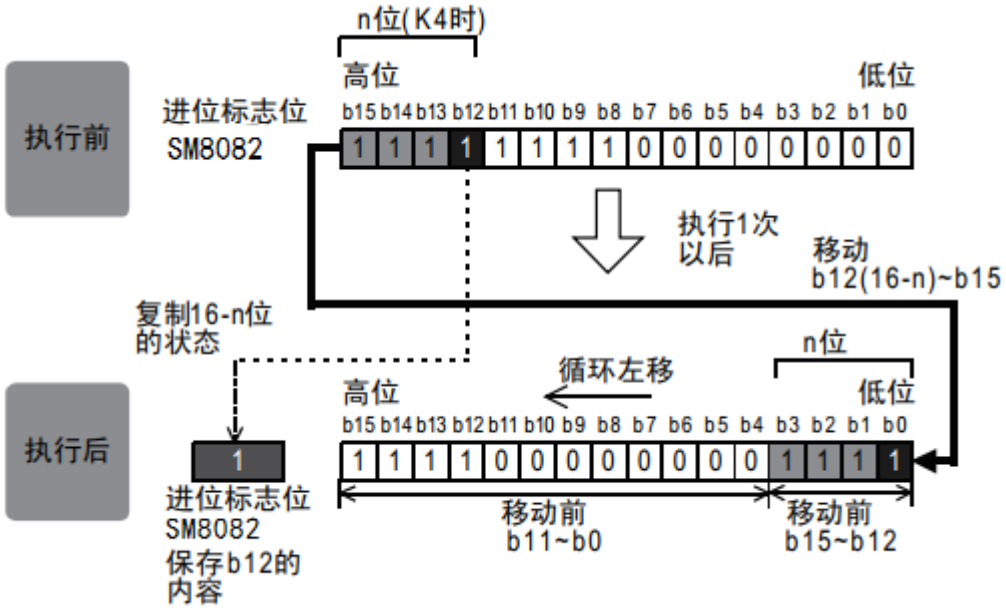
2、功能和动作说明

D 的所有位数中的 n 位循环左移。最后的位保存在进位标志位(SM8082)中。

3、程序举例

将 D0 = -256 = 0xFF00 进行 4 位循环左移





4、

注意事项

（1）相关软元件：SM8082，进位标志，最后从最高位移出的位为 1 时置 ON。

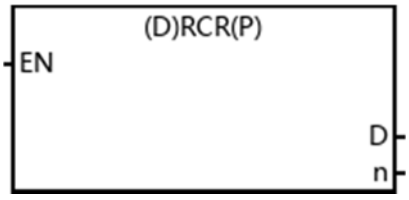
（2）连续执行型(ROL、DROL)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3）在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0) 。

2.5.3 RCR/带进位循环右移指令

使指定位数部分的位信息右移、循环的指令，带进位标志。

1、指令格式



操作数说明

操作数	内容	类型

D	进行循环右移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

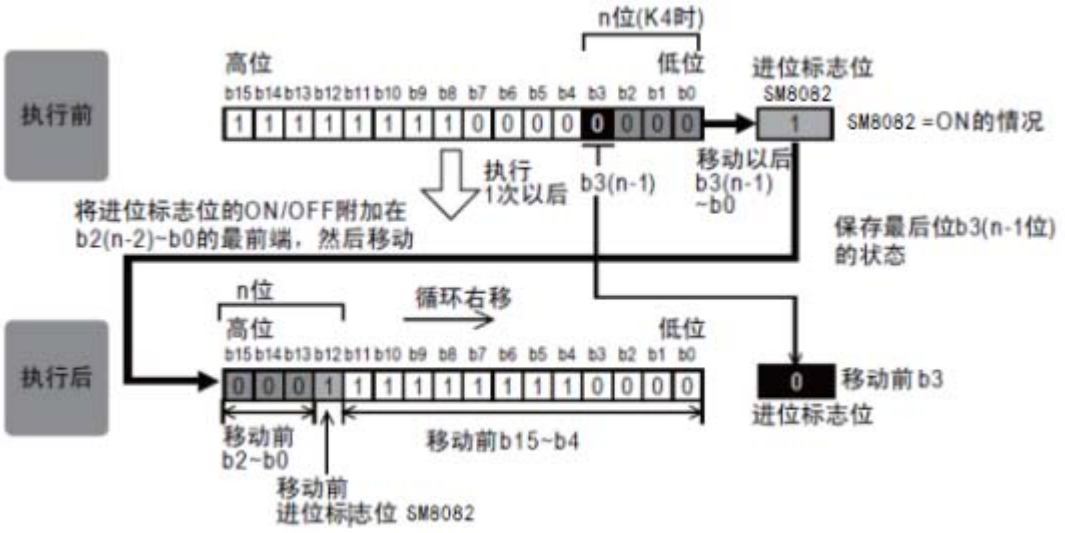
2、功能和动作说明

D 的所有位+1 位(进位标志位 SM8082)向右移动 n 位。因为循环回路中有进位标志位，所以如果执行循环移位指令之前 SM8082 就先 ON 或 OFF，则会被送入目标操作数中。

3、程序举例

将 D0 = -256 = 0xFF00 进行带进位的 4 位循环右移





4、注意事项

（1）相关软元件：SM8082，进位标志，最后从最低位移出的位为 1 时置 ON。

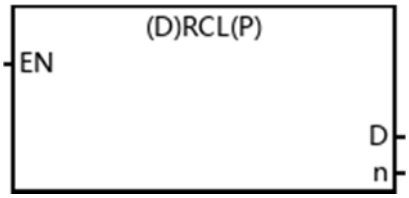
（2）连续执行型(RCR、DRCR)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3）在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0) 。

2.5.4 RCL/带进位循环左移指令

使指定位数部分的位信息左移、循环的指令，带进位标志。

1、指令格式



操作数说明

操作数	内容	类型
D	进行循环左移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

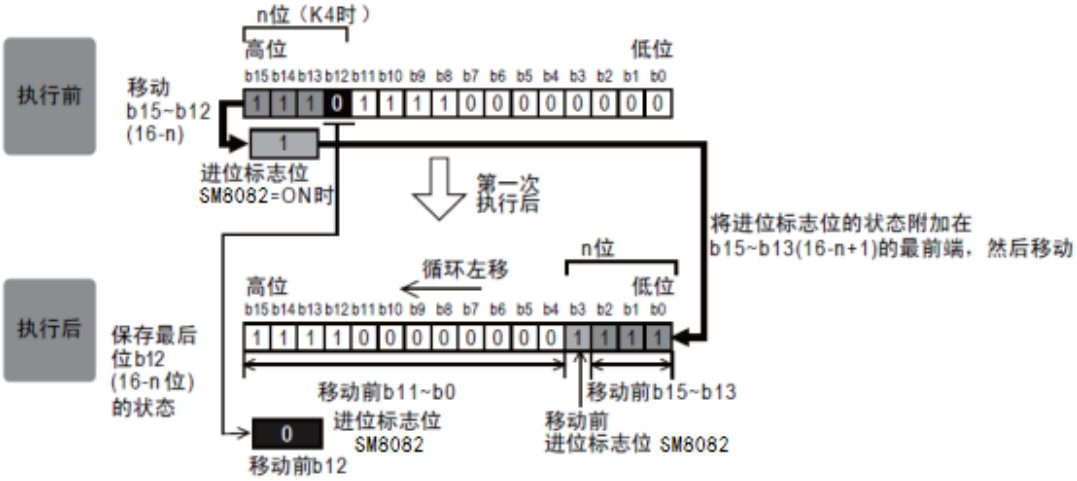
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

2、功能和动作说明

D 的所有位+1 位(进位标志位 SM8082)向左移动 n 位。因为循环回路中有进位标志位，所以如果执行循环移位指令之前 SM8082 就先 ON 或 OFF，则会被送入目标操作数中。

3、程序举例





4、注意事项

（1）相关软元件：SM8082，进位标志，最后从最高位移出的位为 1 时置 ON。

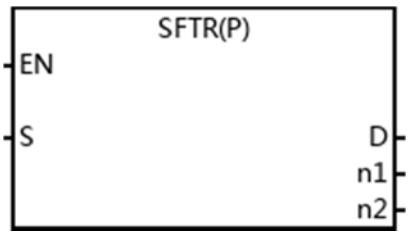
（2）连续执行型(RCL、DRCL)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3）在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0)。

2.5.5 SFTR/位右移指令

右移指定的位长度的指令。移动后，从最高位开始传送 n2 点长度的 S 位软元件。

1、指令格式



操作数说明

操作数	内容	类型
S	传送的位数据起始位元件	位

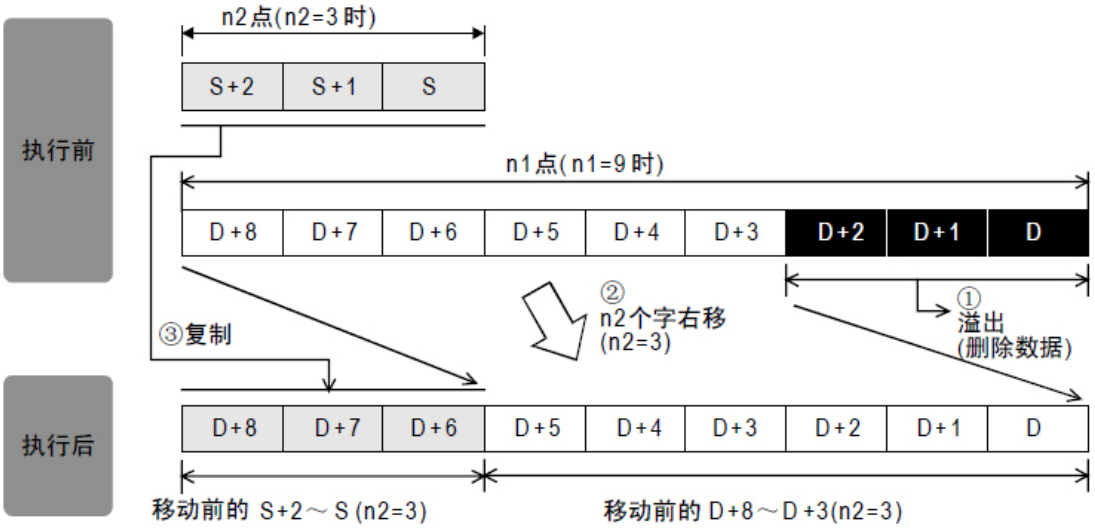
D	右移位的数据起始位软元件	位
n1	右移数据的位长度 n2≦n1≦1024	BIN16
n2	右移的位数 n2≦n1≦1024	BIN16

操作数可用元件类型

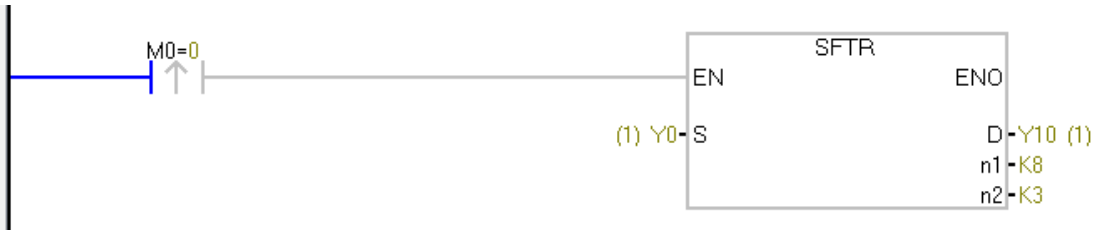
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*	*											*					
D		*	*			*												*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

对于以 D 起始的 n1 位(移位寄存器的长度)数据，右移 n2 位(下记的①、②)。移位后，将 S 开始 n2 位数据传送（下记的③）到从 D+n1-n2 开始的 n2 位中。



3、程序举例



将 Y10~Y17 这 8 点位元件右移 3 位后，将 Y0~Y2 传入高位 Y15~Y17。

执行前

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	0	0	0	1	1	1	1	1

执行后

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	1	0	1	0	0	0	1	1

4、注意事项

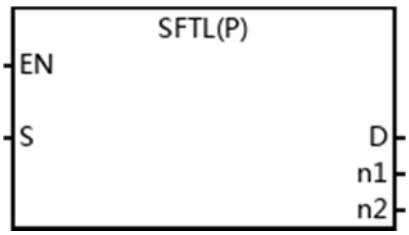
（1）每次当指令输入从 OFF 变为 ON 时，执行 n2 位移位，指令输入状态一直为 ON 时，每个扫描周期(运算周期)都执行移位。

（2）传送源 S 和移位数据 D 重复时，报错 6710。

2.5.6 SFTL/位左移指令

左移指定的位长度的指令。移动后，从最低位开始传送 n2 点长度的 S 位软元件。

1、指令格式



操作数说明

操作数	内容	类型
S	传送的位数据起始位元件	位
D	左移位的数据起始位软元件	位
n1	左移数据的位长度 n2≦n1≦1024	BIN16

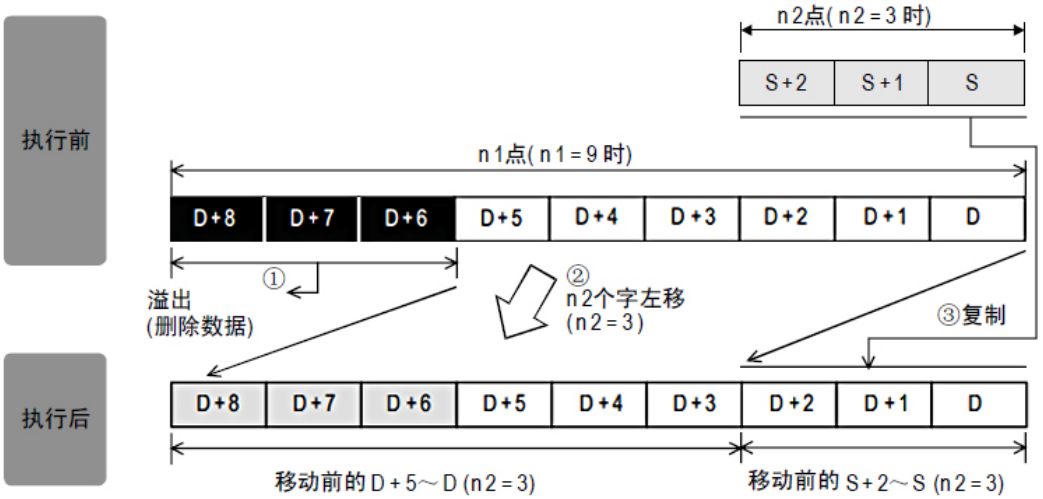
n2	左移的位数 n2≦n1≦1024	BIN16

操作数可用元件类型

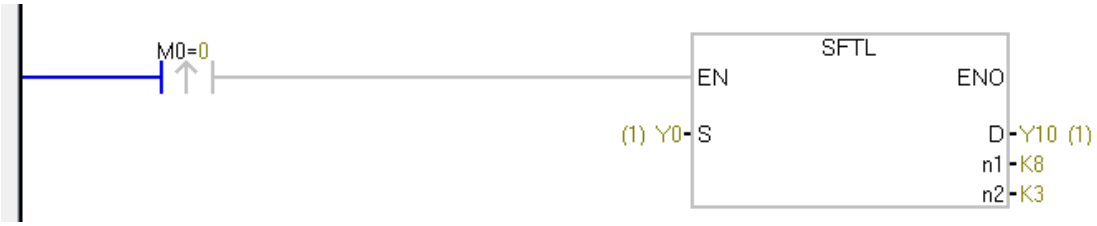
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*	*											*					
D		*	*			*												*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

对于以 D 起始的 n1 位(移位寄存器的长度)数据，左移 n2 位(下记的①、②)。移位后，将 S 开始 n2 位数据传送（下记的③）到从 D+n1-n2 开始的 n2 位中。



3、程序举例



将 Y10~Y17 这 8 点位元件左移 3 位后，将 Y0~Y2 传入低位 Y10~Y12。

执行前

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	1	1	1	1	1	0	0	0

执行后

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	1	1	0	0	0	1	0	1

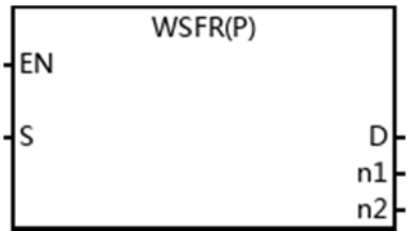
4、注意事项

SFTLP 指令中，每次当指令输入从 OFF 变为 ON 时，执行 n2 位移位，但是请注意 SFTL 指令中，每个扫描周期(运算周期)都执行移位。

2.5.7 WSFR/字右移指令

将 n1 个字长的字软元件右移 n2 个字的指令。

1、指令格式



操作数说明

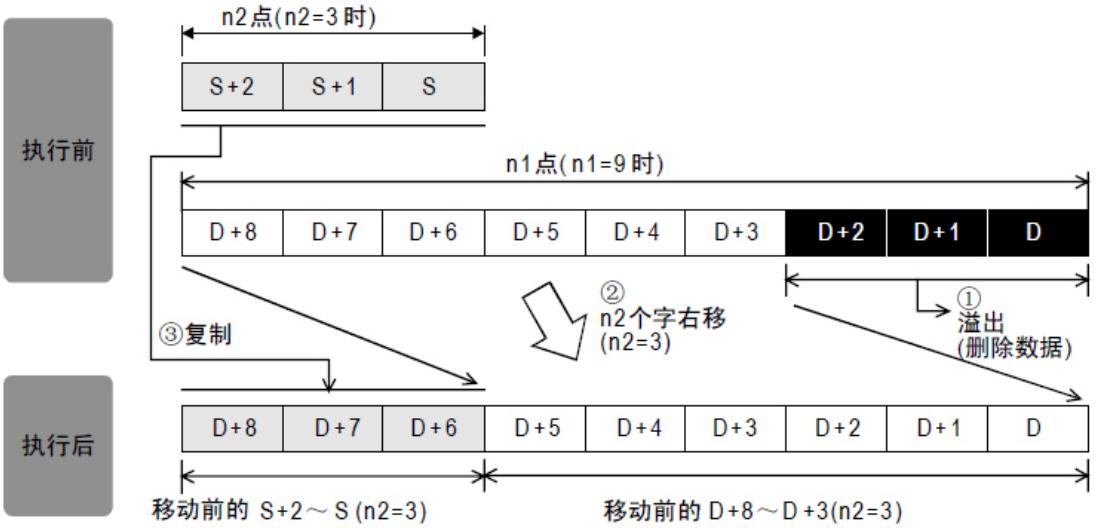
操作数	内容	类型
S	传送的字数据的起始字元件	BIN16
D	右移数据的起始字软元件	BIN16
n1	右移数据的字长度 n2≦n1≦512	BIN16
n2	右移的字数 n2≦n1≦512	BIN16

操作数可用元件类型

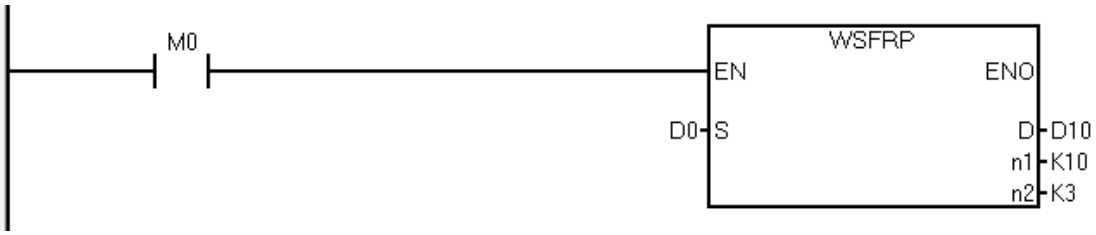
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

对于以 D 起始的 n1 个字软元件，右移 n2 个字(下面的①、②)。移位后，将 S 开始的 n2 点数据传送(下面的③)到从（D+n1-n2）开始的 n2 点中。



3、程序举例



将 D10~D19 这 10 点位元件右移 3 位后，将 D0~D2 传入 D17~D19。

执行前

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	12
D13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16
D17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	17
D18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	18
D19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	19

执行后

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16
D14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	17
D15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	18
D16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	19
D17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2

4、注意事项

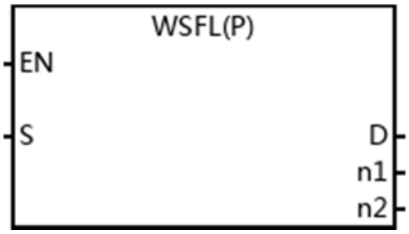
（1）WSFRP 指令中，每次当指令输入从 OFF 变为 ON，就执行 n2 个字的移位，但是请注意 WSFR 指令中，每个运算周期都执行移位。

（2）传送源 S 和移位软元件 D 重复的时候，发生运算错误。(错误代码：K6710)

2.5.8 WSFL/字左移指令

将字数据信息左移指定字个数的指令。

1、指令格式



操作数说明

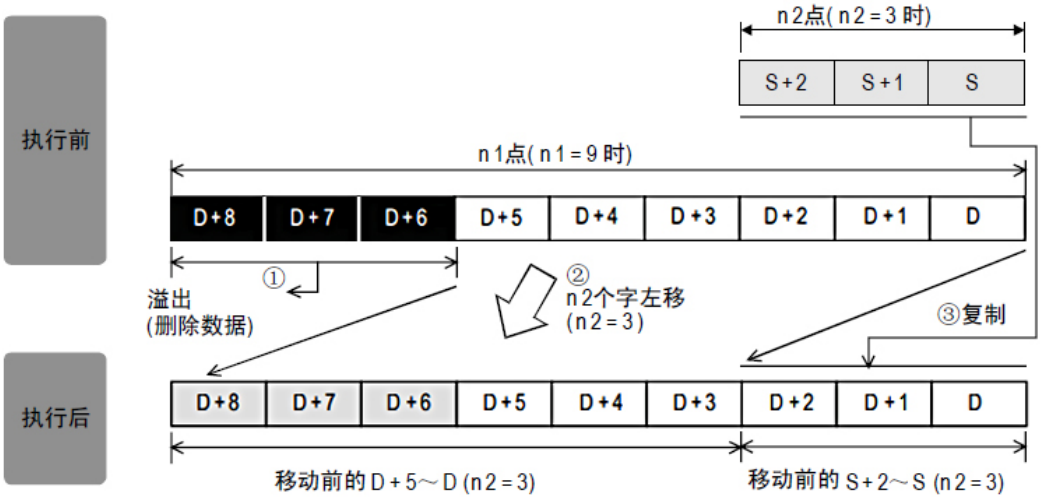
操作数	内容	类型
S	传送的字数据起始字元件	BIN16
D	左移数据的起始字软元件	BIN16
n1	左移数据的字长度 n2≦n1≦512	BIN16
n2	左移的字数 n2≦n1≦512	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

将以 D 起始的 n1 个字软元件，左移 n2 个字(下面的①、②)。移位后，将 S 开始的 n2 点传送(下面的③)到从 D 开始的 n2 点中。



3、程序举例



将 D10~D19 这 10 点位元件左移 3 位后，将 D0~D2 传入 D10~D12。

执行前

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	12
D13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16
D17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	17
D18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	18
D19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	19

执行后

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10
D14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11
D15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	12
D16	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D17	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D18	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16

4、注意事项

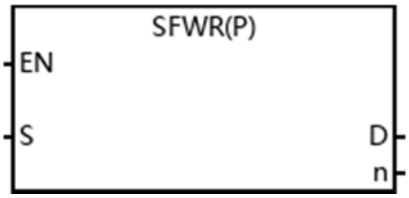
（1）WSFLP 指令中，每次当指令输入从 OFF 变为 ON，就执行 n2 个字的移位，但是请注意 WSFL 指令中，每个运算周期都执行移位。

（2）传送源 S 和移位软元件 D 重复的时候，发生运算错误。(错误代码：K6710)

2.5.9 SFWR/移位写入指令

为实现先入先出或先入后出控制功能，先将准备的数据写入的指令，可用 SFRD 或 POP 指令读出数据。

1、指令格式



操作数说明

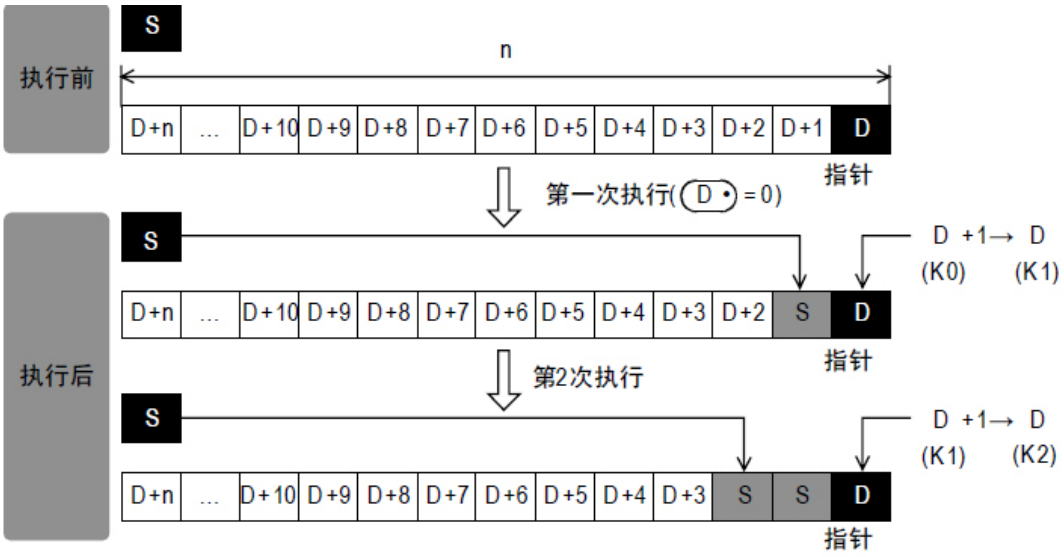
操作数	内容	类型
S	写入的数据	BIN16
D	保存数据并移位的起始字软元件 (D 保存数据点数，D+1 开始保存写入的数据)	BIN16
n	保存的数据点数+1 的值 2≦n≦512	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*			*					
n																			*	*			

2、功能和动作说明

在 D+1 开始的 n－1 点中依次写入 S 的内容，并对 D 中保存的数据数+1。



（1）X000 从 OFF 变为 ON 时， S 的内容被保存到 D+1 中， D+1 的内容变为 S 的值。

（2）S 的内容变化后再次执行输入从 OFF 变为 ON 后， S 的内容被保存到 D+2 中，D+2 的内容变为 S。(由于用连续执行型指令 SFWR，每个运算周期都依次被保存，因此请用脉冲执行型指令 SFWRP 编程。)

（3）从右端依次执行，在指针 D 的内容中表示数据的保存点数。

3、程序举例

略。

4、注意事项

（1）SM8082 进位标志位的动作，指针 D 的内容超过 n－1 时，变为无处理(不写入)，且进位标志位 SM8082 置 ON。

（2）请注意每个扫描周期(运算周期)都会依次保存(改写)。

2.5.10 SFRD/移位读出指令

先入先出形式的数据读出指令。

1、指令格式



操作数说明

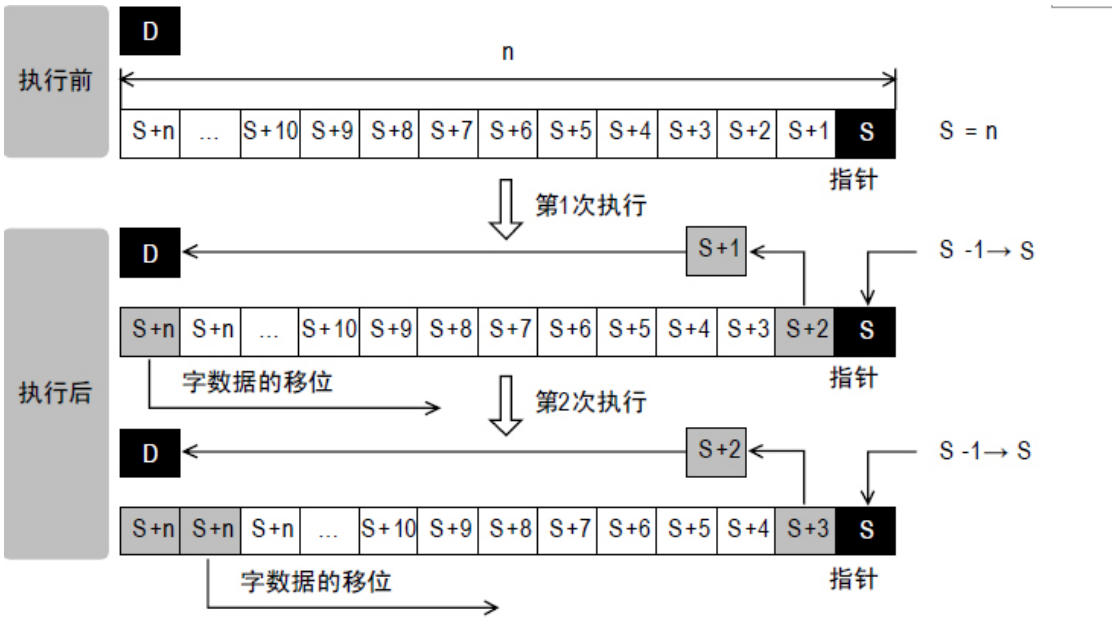
操作数	内容	类型
S	要读出的数据的起始字元件 (S 是数据点数，数据是从 S+1 开始的)	BIN16
D	读出的数据	BIN16
n	保存的数据的点数+1 的值。2≦n≦512	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S									*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					
n																			*	*			

2、功能和动作说明

使用 SFRD(FNC 38)指令被依次写入的 S1+1 传送(读出)到 D 中后，从 S1+1 开始的 n-1 点逐字右移。S 中保存的数据数-1。



（1）指令触点为 ON 时， S+1 的内容传送(读出)到 D 中。

（2） 与此同时，指针 S 的内容减少，左侧的数据逐字右移。

3、程序举例

略。

4、注意事项

（1）SM8080 零位标志位的动作，数据的读出，通常从 S+1 开始执行，但是指针 S 的内容为 0 时，零标标志位 SM8080 动作。

（2）S+n 的内容不会因为读出而变化。

（3）每个扫描周期(运算周期)都会执行依次读出，但 S+n 的内容不会变化。

（4）S 为 0 时，不执行指令。

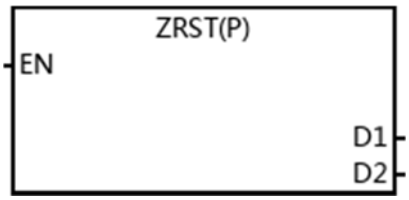
2.6 数据处理指令

处理复杂的数据或满足特殊的数据用途。

2.6.1 ZRST/成批复位指令

成批数据的复位（清零）指令。

1、指令格式



操作数说明

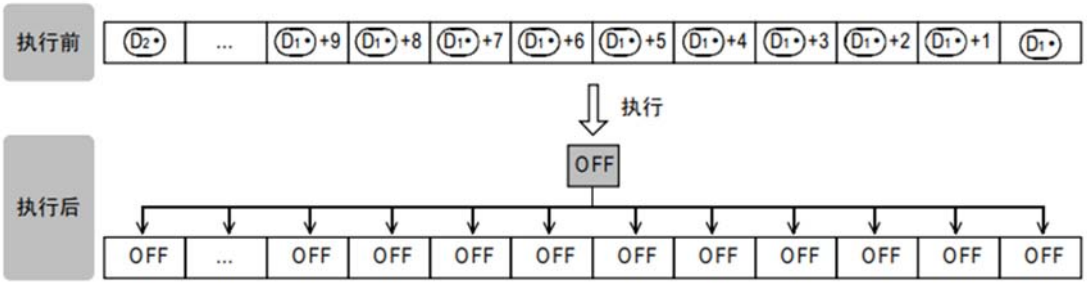
操作数	内容	类型
D1	成批复位的开始元件	位/BIN16 位
D2	成批复位的结束元件	位/BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D1		*	*			*						*	*	*	*			*					
D2		*	*			*						*	*	*	*			*					

2、功能和动作说明

将同一种类的 D1~D2 全部复位。



3、程序举例

略。

4、注意事项

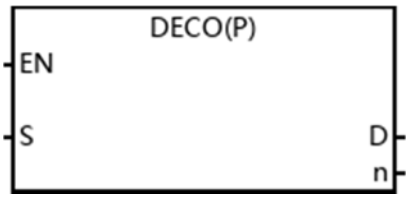
（1）D1，D2 必须指定为同一种类的软元件，且编号 D1≦D2 编号，当 D1 编号大于 D2 时,D1 中指定的软元件只复位 1 点。

（2） ZRST 作为 16 位处理指令。也可以在 D1，D2 中指定 32 位计数器，但是不能出现 D1 和 D2 的寄存器位数不一样的情况。

2.6.2 DECO/译码指令

将数字数据中任意一个转换成 1 点的 ON 位的指令。根据 ON 位的位置可以将位编号读成数值。

1、指令格式



操作数说明

操作数	内容	类型
S	要译码的数据	BIN16 位
D	译码结果	BIN16 位
n	保存译码结果的软元件的位点数(n=1～8)(n=0 时为不处理)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*						*	*	*	*	*	*	*	*	*			
D		*	*			*						*	*	*	*			*					
n																			*	*			

2、功能和动作说明

S 的值相对应的 D～D+2ˆn-1 中的 1 个置 ON。

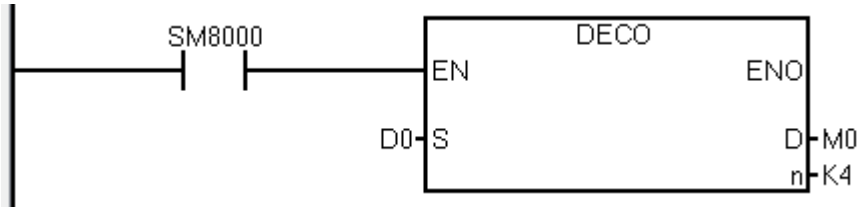
D 为软元件时：1≦n≦8，S 中指定的软元件的 n 位数(1≦n≦8)，在 D 中被译码。S 都为 0 时，位软元件 D 为 ON。n=8 时， 最大 2ˆ8=256 点。

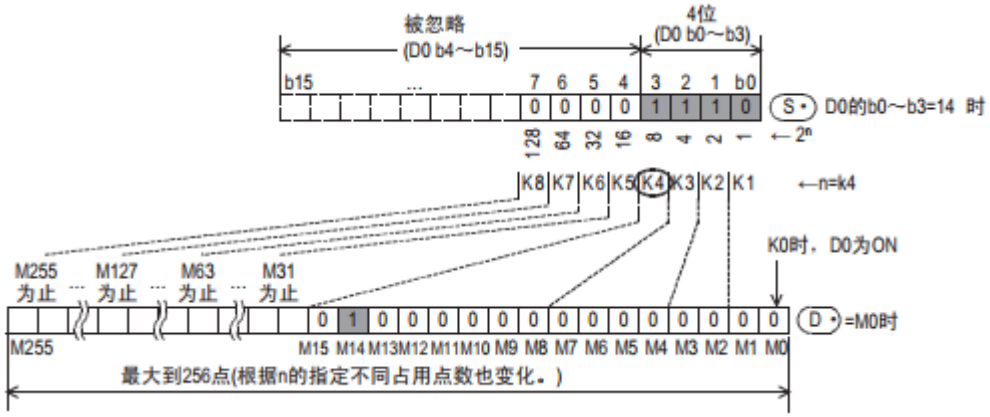
D 为字软元件时：1≦n≦4，S 的低 n 位在 D 中被译码。S 都为 0 时，位软元件 D 的 b0 为 ON。n≦3 时 D 的高字节都为 0。

3、程序举例

（1）D 为软元件

D0 的值(当前值取 14)在 M0～M15 中译码。



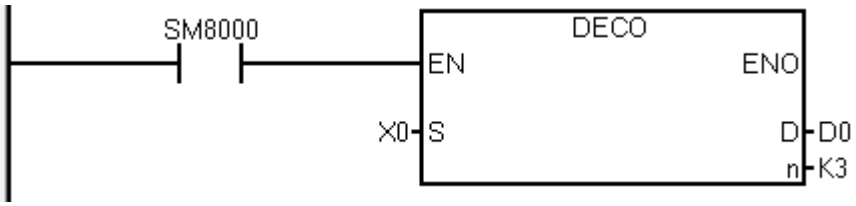


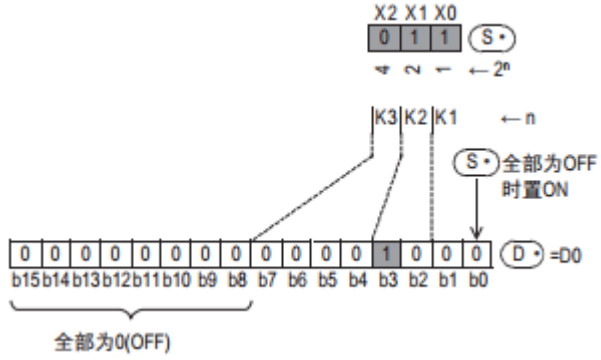
- D0 的 b0～b3 的值为 14(0+2+4+8)时，M0 开始的第 15 号的 M14 为 1(ON)。
- D0=0 时，M0 为 1(ON)。

- 使 n=K4，根据 D0(0～15)的数值，M0～M15 中任意一个为 1 点(ON)。
- 若使 n 在 K1～K8 之间变化，D0 就可以对应 0～255 的数值。

（2）D 为字软元件

X000～X002 的值(X000、X001 为 ON，X002 为 OFF)在 D0 中译码。





- X000～X002 的值为 3(1+2+0)时，b0 开始的第 4 号的 b3 为 1(ON)。
- X000～X002 都为 0(OFF)时，b0 为 1(ON)。

4、注意事项

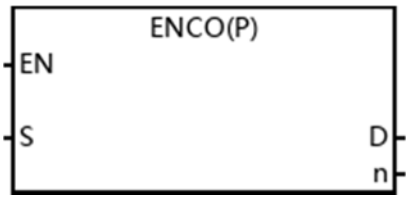
（1）n 在 K1～K8 之间变化时，注意目的寄存器 D 为位软元件的范围不能与其他控制重复。

（2）n 为 0 时，指令不处理。

2.6.3 ENCO/编码指令

求出在数据中 ON 位的位置的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	要编码的数据	BIN16 位

D	编码结果	BIN16 位
n	保存编码结果的软元件的位点数(n=1～8)(n=0 时为不处理)	BIN16 位

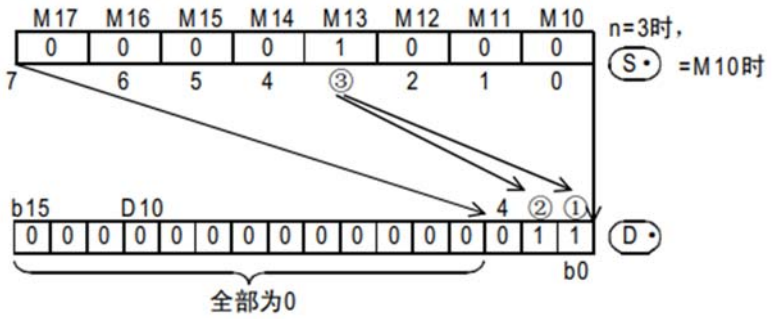
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*						*	*	*	*	*	*	*					
D												*	*	*	*	*	*	*					
n																			*	*			

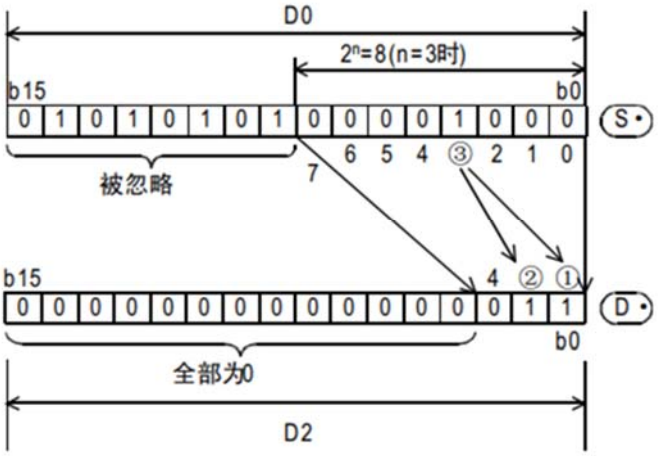
2、功能和动作说明

在 D 中保存 S 的 2^n 位编码后的值，编码就是将 ON 位的位置转换成 BIN 数据。

（1）S 为位软元件：1≦n≦8，S 开始 2^n 个的 ON 位位置，在 D 中编码。n=8 时，S 为最大 256 点。D 的编码结果从高位到低位 n 位，全部为 0(OFF)。



（2）S 为字软元件：1≦n≦4，到 S 中指定的软元件的位 2^n 个为止的 ON 位位置，都在 D 中编码。D 的编码结果从高位到低位 n 位，全部为 0(OFF)。



3、程序举例

略。

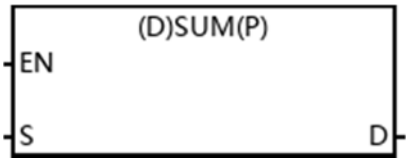
4、注意事项

S 的数据中多个位为 ON 的情况下，忽略低位侧，对高位侧的 ON 位进行编码。

2.6.4 SUM/计算 ON 位数指令

计算在指定的软元件的数据中有多少个为“1”(ON)的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	源数据	BIN16/32 位
D	目标数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

（1）16 位指令

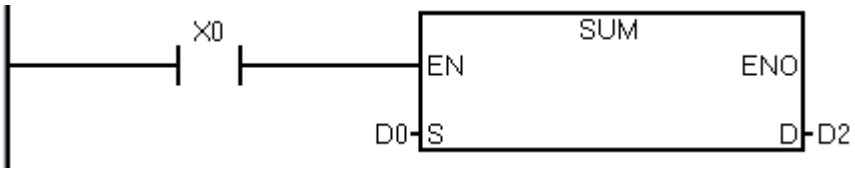
对 S 中为 ON 的位进行计数后保存到 D 中。S 都为 0（OFF 时），零位标志位 SM8080 为 ON。

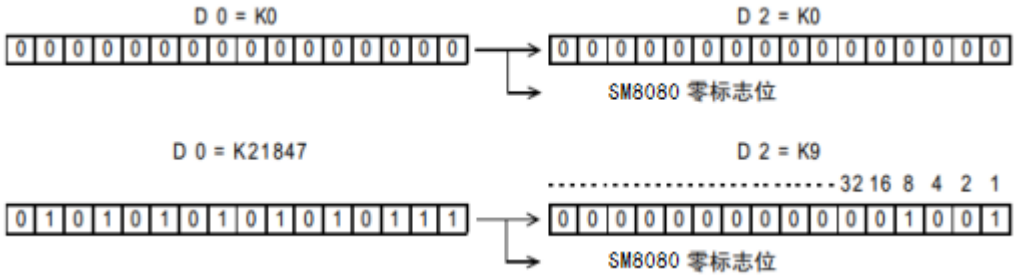
（2）32 位指令

对（S+1,S ）中为 ON 的位进行计数后保存到 D 中。D 中保存 ON 为的点数，D+1 中保存 K0。(S+1,S)都为 0（OFF 时），零位标志位 SM8080 为 ON。

3、程序举例

X000 为 ON 时，对 D0 中 ON 的位数计数后保存到 D2 中





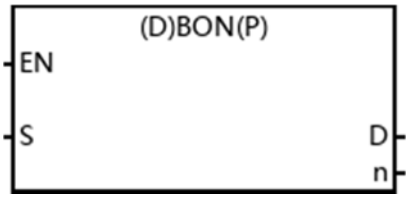
4、注意事项

相关标志位 SM8080 的动作。

2.6.5 BON/ON 位的判定指令

检查软元件中指定位的位置为 ON 还是 OFF 的指令

1、指令格式



操作数说明

操作数	内容	类型
S	源数据	BIN16/32 位
D	目标数据	位
n	要判定的位位置[n：0～15(16 位指令)，n：0～31(32 位指令)]	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*	*			*												*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

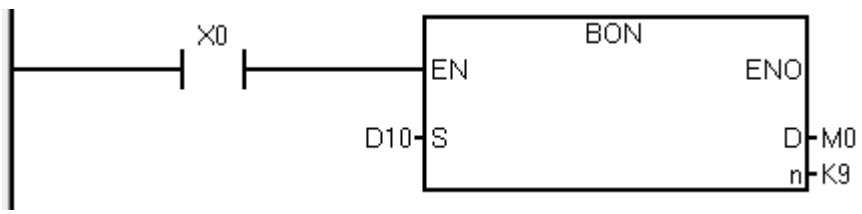
S 的 n 位的状态（ON/OFF）输出到 D。

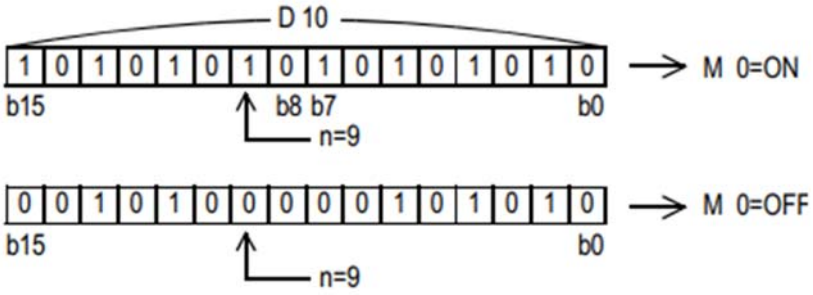
（2）32 位指令

（S+1，S）中的 n 位的状态（ON/OFF）输出到 D。

3、程序举例

D10 中的 n=第 9 位为 1(ON)时，M0 为 1(ON)。





4、注意事项

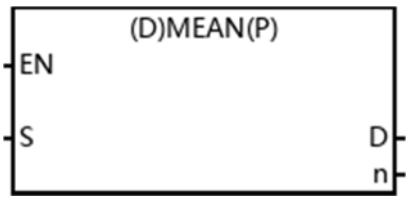
将 D、R 指定为 32 位指令的 n 时，[n+1,n]的 32 位值便生效，敬请注意。

DBON DO MO RO 时，则 n=[R1, RO]

2.6.6 MEAN/平均值指令

求数据的平均值的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	求平均值的源数据	BIN16/32 位
D	平均值	BIN16/32 位
n	求平均值数据个数(n=1～64)	BIN16/32 位

操作数可用元件类型

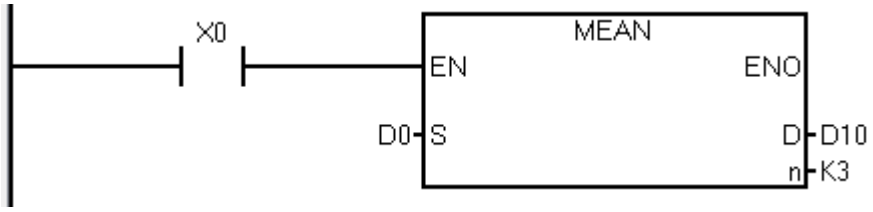
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

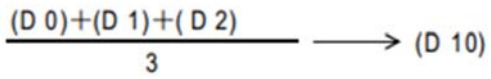
2、功能和动作说明

将 S 开始的 n 个 16 位或 32 位（根据指令类型）数据的平均值保存到 D 中，余数舍去。

3、程序举例

将 D0、D1、D2 的数据相加，除以 3 后求得的值保存到 D10 中





4、注意事项

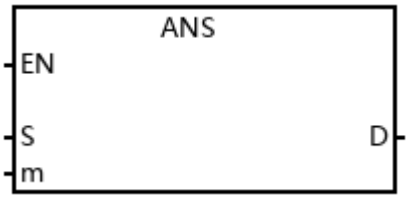
（1）软元件编号溢出时，在可能的范围内将 n 作为较小的值处理。

（2）注意 n 的范围(n=1～64)，超出报错。

2.6.7 ANS/信号报警器置位指令

对信号报警器用的状态(S900～S999)进行置位用的指令。

1、指令格式



操作数说明

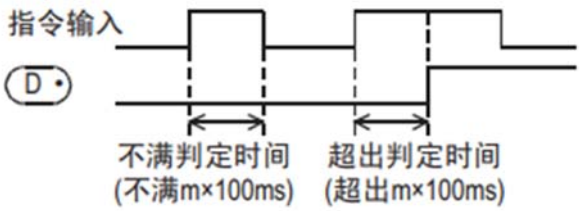
操作数	内容	类型
S	计时定时器	BIN16/32 位
m	判定时间[m=1～32,767(100ms 单位)]	BIN16/32 位
D	信号报警器	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*						*					
m														*	*				*	*			
D						*												*					

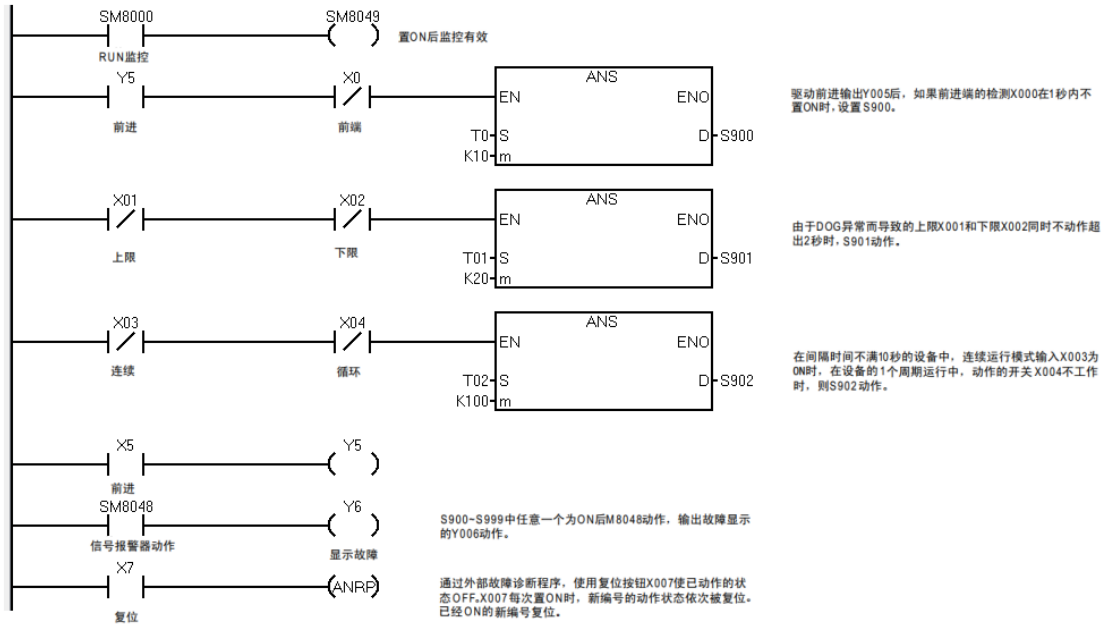
2、功能和动作说明

16 位运算指令，指令输入超出判定时间[m×100ms，定时器 S]以上持续为 ON 时，D 置位。指令输入在不满判定时间[m×100ms]就已 OFF 的情况下，复位判定用定时器 S 的当前值,不置位 D。此外，指令输入 OFF 后，判定用定时器复位。



3、程序举例

如下所示，编写诊断外部故障用的程序，如监控 SD8049(ON 状态最小编号)的内容时，会显示 S900～S999 中为 ON 的状态的最小编号。同时发生多个故障时，排除了最小编号的故障后可以得知下一个故障编号 。



4、注意事项

（1）S 为 T 软元件时，范围为 T00～T199，报警用状态软元件范围是 S900～S999。

（2）注意与 S900～S999 相关的特殊寄存器的动作。

软元件	名称	内容
SM8049	信号报警器有效	SM8049 置 ON 后，下面的 SM8048 和 SD8049 工作。
SM8048	信号报警器动作	SM8049 为 ON，状态 S900～S999 中任一动作的时候，SM8048 置 ON。
SD8049	ON 状态最小编号	保存 S900～S999 中动作的最小编号。

2.6.8 ANR/信号报警器复位指令

对信号报警器(S900～S999)中已经置 ON 的小编号进行复位。

1、指令格式



2、功能和动作说明

16 位运算指令，ANR 指令输入为 ON 后，将信号报警器 S900～S999 中运行中的状态复位。如有多个状态动作时，复位编号最小的一个状态。再次使指令输入为 ON 后，在动作的信号报警器用状态(S900～S999)中，下一个最小的编号被复位。

软元件	名称	内容
SM8049	信号报警器有效	SM8049 置 ON 后，下面的 SM8048 和 SD8049 工作。
SM8048	信号报警器动作	SM8049 为 ON，状态 S900～S999 中任一动作的时候，SM8048 置 ON。
SD8049	ON 状态最小编号	保存 S900～S999 中动作的最小编号。

3、程序举例

可参考 ANS 指令。

4、注意事项

（1）相关寄存器的动作:

软元件	名称	内容
SM8049	信号报警器有效	SM8049 置 ON 后，下面的 SM8048 和 SD8049 工作。
SM8048	信号报警器动作	SM8049 为 ON，状态 S900～S999 中任一动作的时候，SM8048 置 ON。
SD8049	ON 状态最小编号	保存 S900～S999 中动作的最小编号。

（3）使用 ANR 指令时，每个运算周期都依次复位最小标号，ANPR 指令只执行 1 个运算周期，根据实际需求来选择对应的指令以复位对应的报警器。

2.6.9 SQR/开方运算指令

求平方根(开根号)的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	源数据	BIN16/32 位

D	目标数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*	*	*			
D														*	*			*					

2、功能和动作说明

计算出 S 的数据的平方根后，保存到 D 中。

3、程序举例

无

4、注意事项

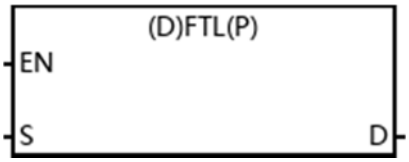
（1）当运算结果为小数时，舍弃小数部分，将整数结果填入 D，SM8081 置 ON。

（2）结果为 0 时，SM8080 置 ON。

2.6.10 FLT/BIN 整数→2 进制浮点数转换指令

将 BIN 整数值转换成 2 进制浮点数(实数)的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	BIN 整数	BIN16/32 位
D	2 进制浮点数(实数)	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

将 S 的 BIN 整数值数据转换成 2 进制浮点数(实数)值后，保存在（D +1，D）中。

3、程序举例

无

4、注意事项

16 位和 32 位指令的目的操作数都是占用 2 个字软元件。

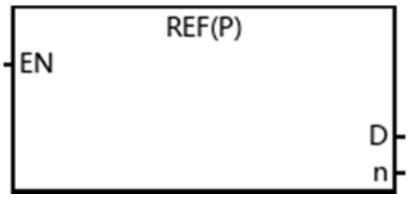
2.7 高速处理指令

可以按最新的输入输出信息进行程序控制，并能有效利用数据高速处理能力进行中断处理。

2.7.1 REF/输入输出刷新指令

在顺控程序扫描过程中，想要获得最新的输入(X)信息时，以及将输出(Y)扫描结果立即输出的指令。

1、指令格式



操作数说明

操作数	内容	类型
D	刷新的位软元件(X、Y)	位
n	刷新点数(8～256 中 8 的倍数)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D	*	*																					
n																			*	*			

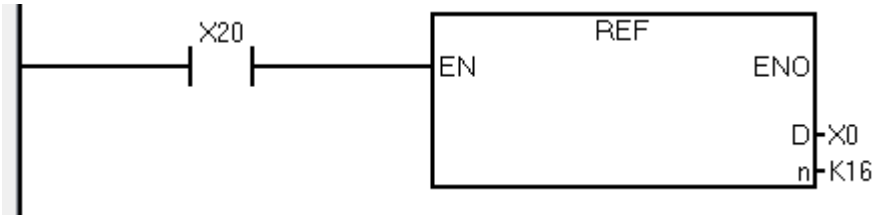
2、功能和动作说明

将 D 地址开始的 n 个元件状态进行立即更新，由于 PLC 访问端口是按字节访问的特性，故要求 S 的地址为 X0、X10、.....Y0、Y10、.....等最低位为 0 的编号元件。且 n 的值必须是 8 的整数倍（n=8~256）。

正常情况下，输入端口 X 的状态读取在每次程序开始执行扫描之前，实际的输入端口状态变化延迟决定于输入元件的滤波时间，X0~X7，X10~X17 有数字滤波功能，滤波时间在 0~60ms 可设（REFF 指令，默认滤波时间为 10ms）；输出端口 Y 的状态刷新则在每次程序执行扫描完毕（执行到 END）之后批次执行，这样 IO 处理会有一定的延迟。若应用中需要最新的输入信号以及希望立即输出运算结果时，可以使用立即刷新指令 REF。

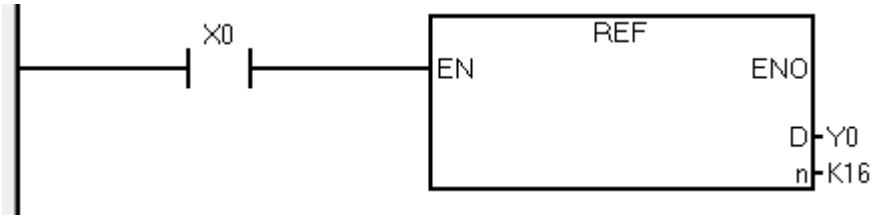
3、程序举例

（1）立即刷新输入信号



X20 为 ON 时，会立即读取 X0~X17 的输入点状态，更新输入信号，没有产生输入延迟。

（2）立即刷新输出信号



X0 为 ON 时，会立即将 Y0~Y17 的状态进行刷新，输出信号立即变更，不必到 END 指令才输出。

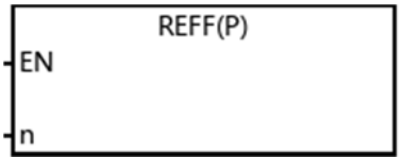
4、注意事项

根据实际的应用需求应用此指令，可在当前位置立即刷新输入输出，而不是按照梯形图的运行流程来进行输入输出的刷新。

2.7.2 REFF/带滤波器的输入刷新指令

输入 X0~X17 的输入滤波器为数字式滤波器，采用 REFF 指令和 SD8020 可以更改滤波器的时间。

1、指令格式



操作数说明

操作数	内容	类型
n	滤波器的时间[0~60ms]	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
n														*	*				*	*			

2、功能和动作说明

根据输入的数字式滤波器时间[n×1ms]刷新输入 X000～X017 的 16 点映像存储区。

关于滤波器时间在不同的使用情境下，会有所区别，当前以 X5S 系列为例说明：

当 n 设定为 0 时，实际对应的输入滤波器滤波时间为最小滤波时间。

3、最小滤波时间

输入端子	最小滤波时间	
	X5S-32/X5-32	X5-60
X00~X07	1.5us	1.5us

X10~X13	400us	1.5us
其他本体输入端子	400us	

用于高速或中断的输入端子，在高速或中断程序运行时，一律按端子对应的最小的滤波时间进行滤波。

4、程序举例

无。

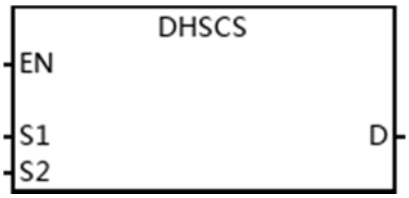
5、注意事项

注意高速中使用的输入和输入中断功能中指定的中断指针的输入，此情景中 REFF 指令刷新 SD8020 的滤波器时间不能正常处理，而是按上述说明进行滤波器设定。

2.7.3 HSCS/高速比较置位指令

每次执行时，都将高速计数器的计数值和指定值做比较，如果两个值相等，立即置位比较输出的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	高速比较设定值	BIN32 位
S2	高速计数器[C235～C255]	BIN32 位
D	被置位的对象元件	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*		*	*	*	*			
S2													*					*					
D		*	*			*												*					*

2、功能和动作说明

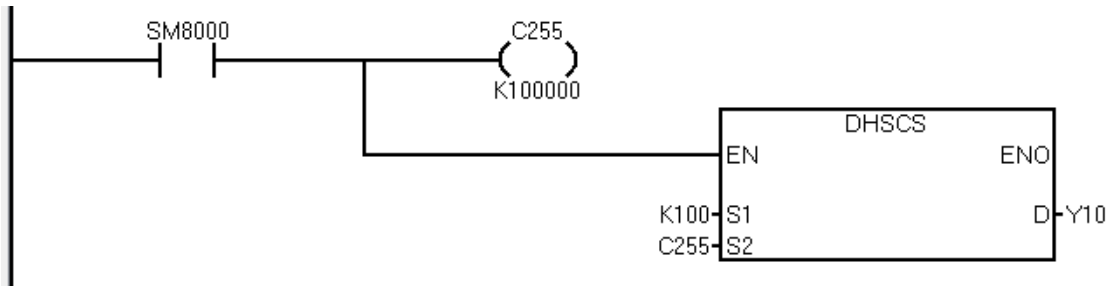
（1）当 S2 计数器的当前值等于设定值 S1 时，立即置位 D。S2 变量必须为高速计数器 C235~C255，因为涉及到的计数器均为 32 位，故必须采用 32 位指令 DHSCS。

（2）D 为普通位元件时：当为 Y0~Y17 范围端口时为立即输出；当为 Y20 以后的端口时，会等本次用户程序扫描完毕才会输出；当为 M、S 变量时，立即刷新。

（3）D 为调用计数中断子程序指针时：当 D 为 I100~I600 时，即为调用高速计数器中断 0~5 的子程序。当然必须写好相应的中断子程序、开启相应中断允许标志和全局中断允许标志等，才能正常相应计数器中断。SM8150~SM8150 置 ON 则禁止 I100~I600 中断。

3、程序举例

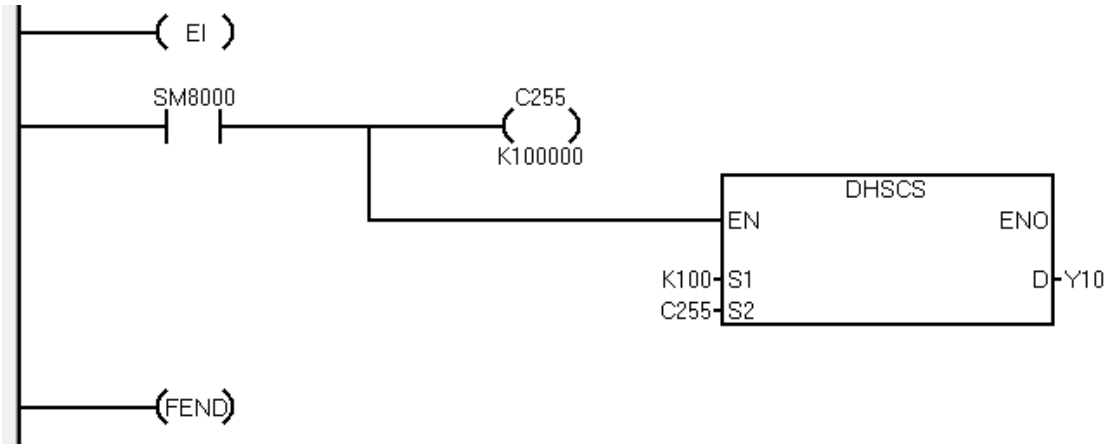
（1）置位普通位元件



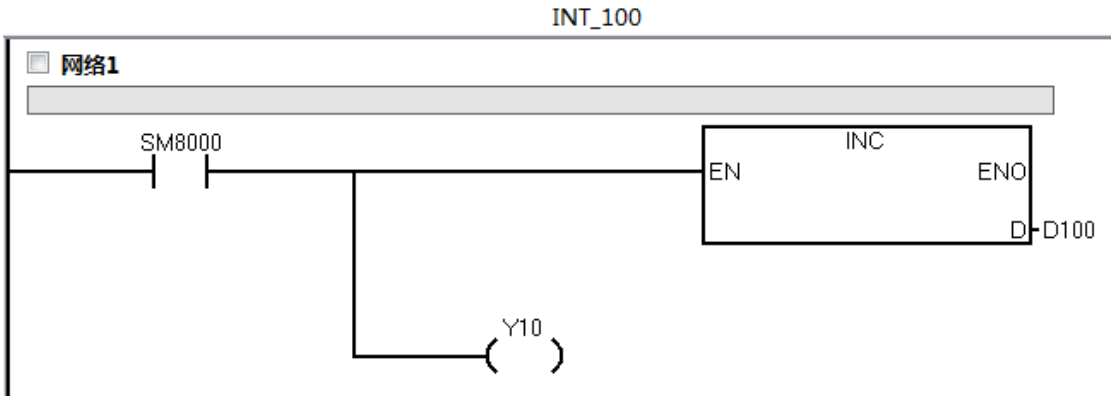
当 C255 计数到达 100 时，Y10 立即置位。

（2）调用计数中断子程序

主程序：



I100 中断子程序：



当 C252 计数到达 100 时，会执行 I100 对应的中断子程序，D100 值加 1，Y10 置位。

4、注意事项

（1）使用 HSCS 指令时，应保证所使用的计数器已被启用，否则计数器的值将不会变化；

（2）计数器是以中断方式响应计数器的输入信号，及时比较，若本次比较时满足匹配关系，比较输出立即置位。例如上面例一中，若 C255 的当前值变为 99->100 或 101->100 时，Y10 立即置位，且一直保持该状态，之后即使 C255 与 K100 的比较结果变成不相等，Y10 仍然保持 ON 状态，除非有另外的复位操作。

（3）指令的比较输出结果只决定脉冲于输入时的比较结果，即使采用 DMOV、DADD 等指令改写 C235~C255 的内容，若没有脉冲输入，比较输出也不会变化。

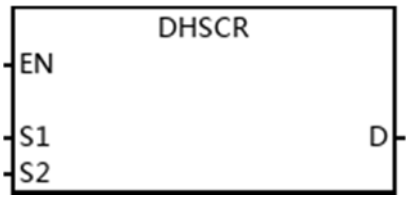
（4）当 HSCS 指令的输出目标为计数器中断 I100~I600 时，每个中断号只能使用 1 次，不可重复。

（5）HSCS、HSCR、HSZ 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在 6 个指令一下。HSZ 指令特殊模式（高速表格比较模式）仅能同时驱动 1 个指令。

2.7.4 HSCR/高速比较复位指令

每次执行时，都将高速计数器的计数值和指定值做比较，如果两个值相等，立即复位比较输出的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	高速比较设定值	BIN32 位
S2	高速计数器[C235～C255]	BIN32 位
D	被复位的位元件	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*		*	*	*	*			
S2													*					*					
D		*	*			*							*					*					

2、功能和动作说明

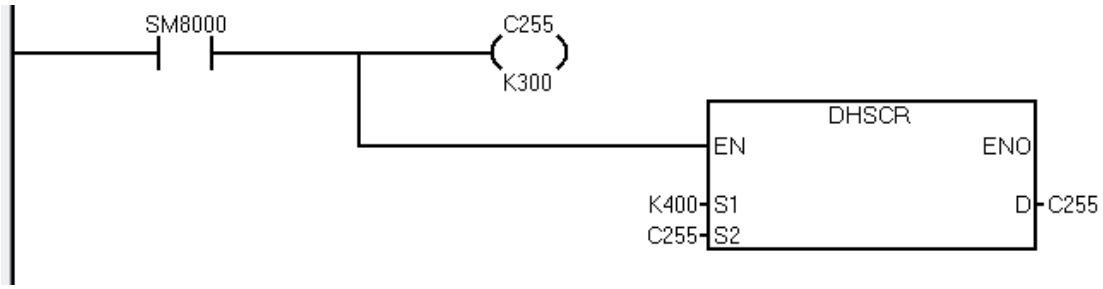
（1）当 S2 计数器的当前值等于设定值 S1 时，立即复位 D。S2 变量必须为高速计数器 C235~C255，因为涉及到的计数器均为 32 位，故必须采用 32 位指令 DHSCR。

（2）D 为普通位元件时：当为 Y0~Y17 范围端口时为立即输出；当为 Y20 以后的端口时，会等本次用户程序扫描完毕才会输出；当为 M、S 变量时，立即刷新。

（3）D 为计数器时：可以复位任意计数器，常用于复位本体高速计数器。

3、程序举例

自我复位的梯形图实例 C255 的当前值变为 400 后，立即执行 C255 的复位，当前值为 0，输出触点为 OFF。



4、注意事项

（1）使用 HSCSR 指令时，应保证所使用的计数器已被启用，否则计数器的值将不会变化；

（2）计数器是以中断方式响应计数器的输入信号，及时比较，若本次比较时满足匹配关系，比较输出立即复位。

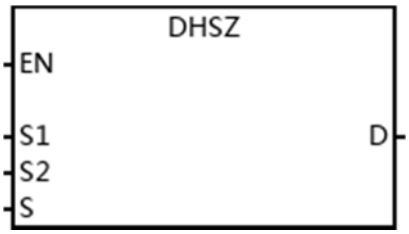
（3）指令的比较输出结果只决定脉冲于输入时的比较结果，即使采用 DMOV、DADD 等指令改写 C235~C255 的内容，若没有脉冲输入，比较输出也不会变化。

（4）HSCS、HSCR、HSZ 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在 6 个指令一下。HSZ 指令特殊模式（高速表格比较模式）仅能同时驱动 1 个指令。

2.7.5 HSZ/高速区间比较指令

将高速计数器的当前值和 2 个值(区间)进行比较，并将比较结果输出(刷新)位软元件(3 点)中。也可将该指令设置成高速表格比较模式。

1、指令格式



操操作数说明

操作数	内容	类型
S1	比较区间下限	BIN32 位
S2	比较区间上限	BIN32 位
S	高速计数器[C235～C255]	BIN32 位
D	输出比较结果	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*		*	*	*	*			
S2								*	*	*	*	*	*	*	*		*	*	*	*			
S													*					*					
D		*	*			*																	

2、功能和动作说明

（1）普通模式

• S1 为区间下限，S2 为区间上限，满足 S1<=S2；

• S 必须为 C235~C255；

• D 占用三个连续地址单元，当为 Y0~Y17 端口范围时，立即输出，其他 Y 端口在本次用户程序扫描完毕才输出；当为 M、S 变量时，为立即刷新。

当 S1 ＞ S 时，D 置位；

当 S1 ≤ S ＜ S2 时，D+1 置位；

当 S ≥ S2 时，D+2 置位；

（2）高速表格比较模式

当参数 D 为特殊辅助继电器 SM8180 时，即指定高速表格比较模式，此模式下，注意操作数的意义。

• S1 表示比较表格的起始地址，只能是数据寄存器 D；

• S2 表示表格的行数（仅 K、H）限制为 K1~K128/H1~H80；

• S 必须为 C235~C255；

• D 为 SM8180，指定高速表格比较模式。

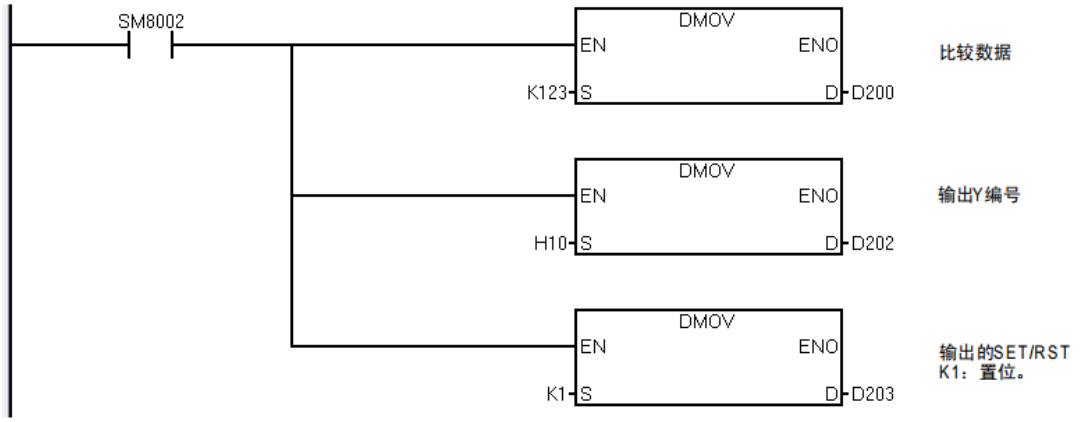
比较表格的形式如下：

比较数据（32 位）	输出 Y 编号	SET/RST	表格计数器（SD8180）
S1+1，S1	S1+2	S1+3	0 ↓

S1+5，S1+4	S1+6	S1+7	1 ↓
S1+9，S1+8	S1+10	S1+11	2 ↓
···	···	···	···
S1+(S2*4-3),S1+(S2*4-4)	S1+(S2*4-2)	S1+(S2*4-1)	S2-1 ↓ 从 0 开始重复

其中：每一行表格需要占用 4 个软元件；输出 Y 编号用 16 进制数指定，Y0~Y,7 对应 H0~H7，Y10~Y17 对应 H10~H17.....Y370~Y377 对应 H370~H377；通过设定的内容 0/1 控制对应表格中的输出 Y 的 ON/OFF。

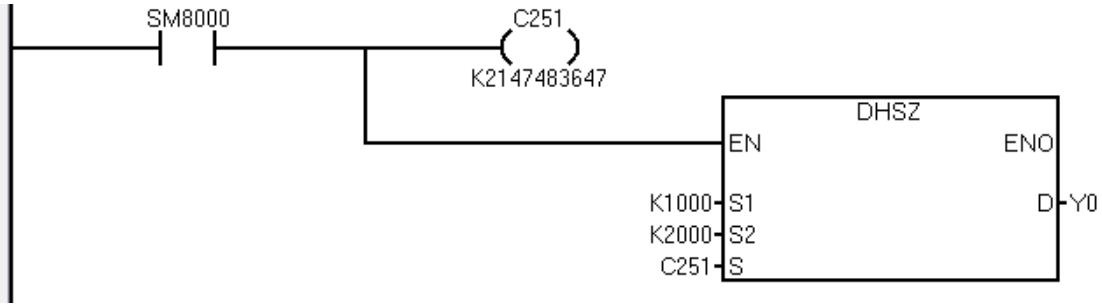
一般通过数据传送指令来构建表格，比如



3、程序举例

（1）普通模式

高速计数器 C251 的当前值如下变化(计数)时，在 Y000、Y001、Y002 的任意一个中输出比较结果。

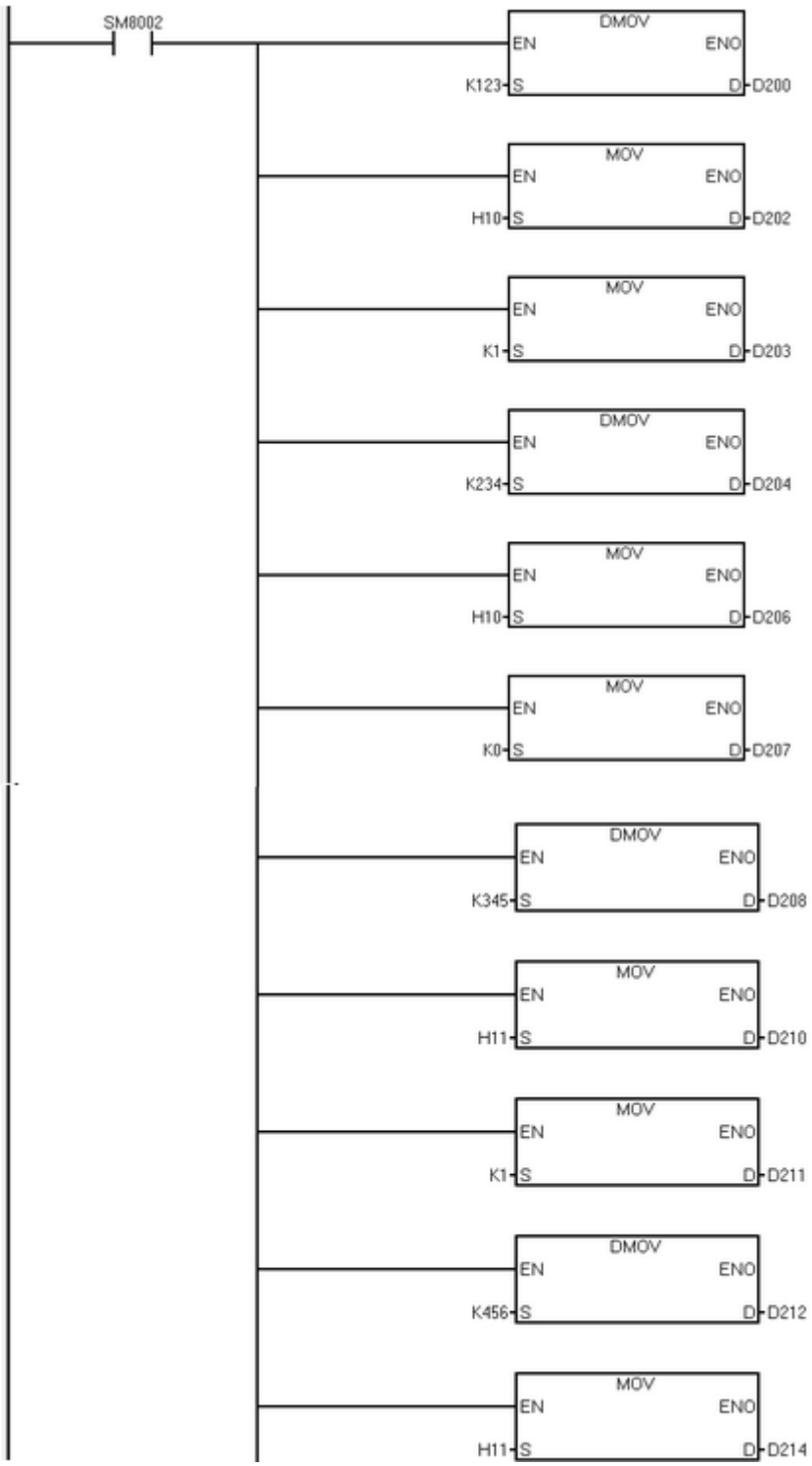


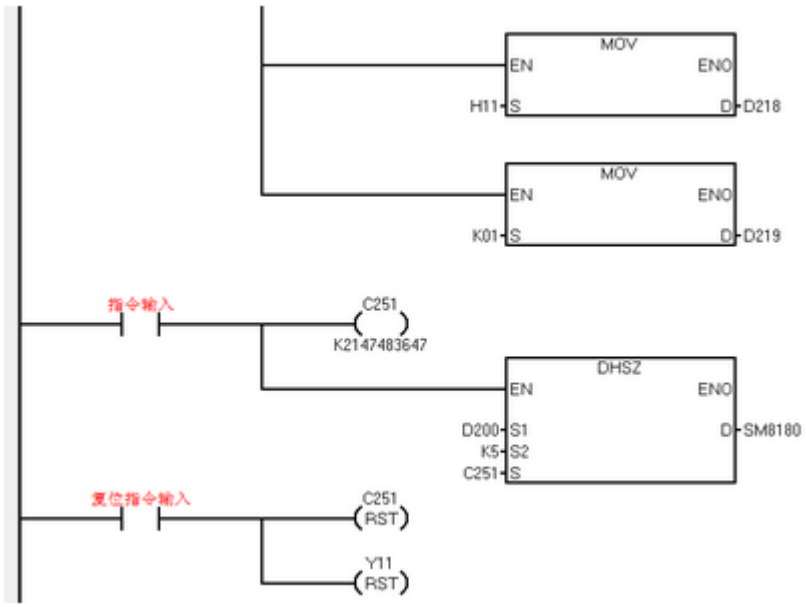
当 C251 < 1000 时，Y0 为 ON；

当 1000 <= C251 < 2000 时，Y1 为 ON；

当 C251 > 2000 时，Y2 为 ON。

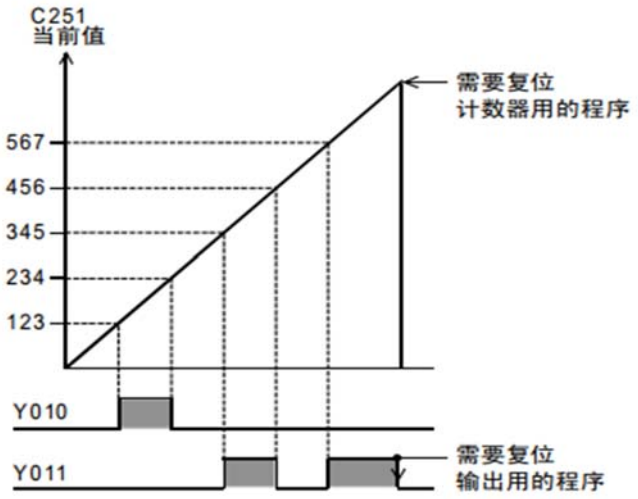
（2）高速表格比较模式





梯形图转换成比较表格：

比较数据（32 位）	输出 Y 编号	SET/RST	表格计数器（SD8180）
D201,D200 K123	D202 H10	D203 K1	0 ↓
D205,D204 K234	D206 H1	D207 K0	1 ↓
D209,D208 K345	D210 H11	D211 K1	2 ↓
D213,D212 K456	D214 H11	D215 K0	3 ↓
D217，D216 K567	D218 H11	D219 K1	4 ↓ 从 0 开始重复



①执行该指令后，表格数据的最上一行的表格被设置成比较对象数据。
②高速计数器 C251 的当前值和比较对象的表格数据一致时，比较对象的表格数据中指定的输出 Y 编号被执行 SET/RST。并且，这个输出 Y 不等到执行 END 指令的输出刷新，直接执行输出处理。
③表格计数器 SD8180 的当前值+1。
④比较对象的表格数据移到下一行的表格。
⑤在表格计数器 SD8180 的当前值变为 4 之前，重复②～③的动作。当前值变为 4 时，返回到动作①，表格计数器复位(SD8180=0)。此时，结束标志位 SM8181 ON。
⑥指令触点断开(指令执行 OFF)后，中止执行指令，表格计数器 SD8180 复位(SD8180=0)。

4、注意事项

（1）HSCS、HSCR、HSZ 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在 6 个指令一下。HSZ 指令特殊模式（高速表格比较模式）仅能同时驱动 1 个指令。

（2）由于在执行 END 时，表格才会制作完成，所以特殊模式下的动作是在第二个周期开始动作。高速表格模式下，第二个扫描周期才开始表格比较置位或复位输出。

（3）普通模式下，即使高速计数器没有打开也会进行区间比较置位对应软元件，特殊模式下不会进行比较置位。

（4）相关的特殊寄存器

SM8180	HSZ 高速表格比较模式
SM8181	高速表格比较执行结束标志
SD8180	HSZ 高速表格比较模式下的表格计数器

2.7.6 SPD/脉冲密度指令

采用中断输入方式对指定时间内的输入脉冲进行计数的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	输入(X)脉冲的软元件编号	位
S2	时间(ms)数据或是保存数据的字软元件编号	BIN16/32 位
D	保存脉冲密度数据的起始字软元件编号	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1	*																						
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D												*	*	*	*	*	*	*					

2、功能和动作说明

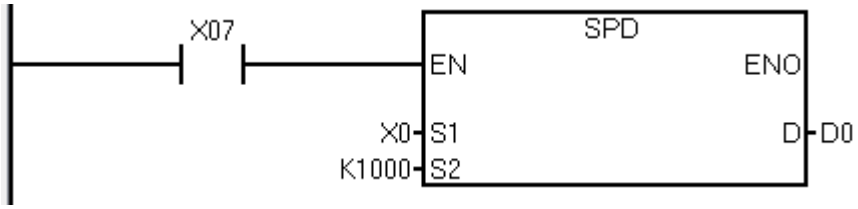
（1）16 位指令

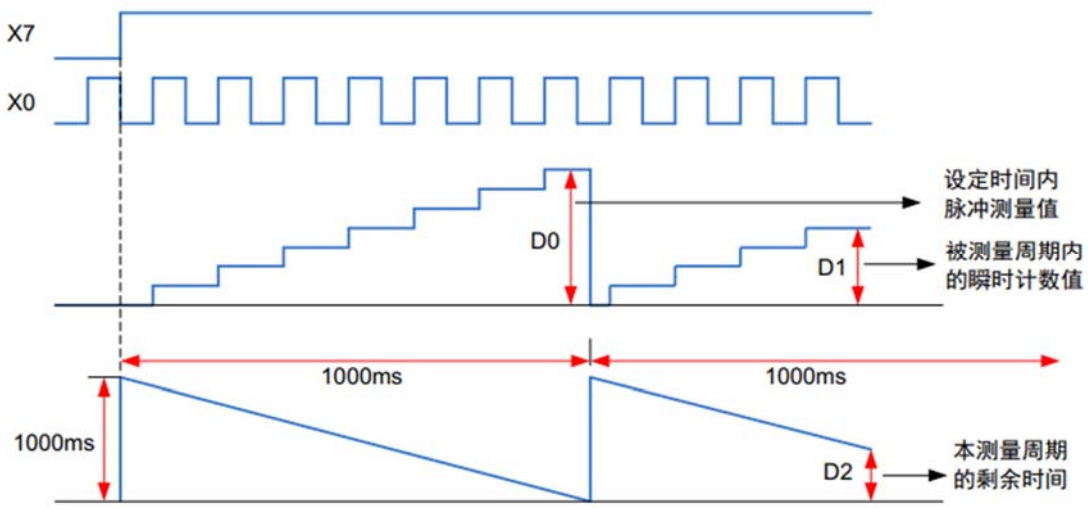
只在 S2×1ms 时间内对输入 S1 的脉冲进行计数，测定值保存到 D，当前值保存到 D+1，剩余时间保存到 D+2 中。重复这个操作，可以在测量值 D 中，得到脉冲密度(也就是与转速成比例的值)。

（2）32 位指令

只在（S2+1，S2）×1ms 时间内对输入 S1 的脉冲进行计数，测定值保存到（D+1，D），当前值保存到（D+3，D+2），剩余时间保存到（D+5，D+4）中。重复这个操作，可以在测量值（D+1，D）中，得到脉冲密度(也就是与转速成比例的值)。

3、程序举例





当 x7 置 ON 时，D1 对 x0 的 OFF->ON 动作计数，1000ms 后将其结果存入 D0，随之 D1 复位，再次对 X0 的动作计数。D2 用于测定剩余时间。，因此根据 D0 和 S2 的设定值可以求得脉冲频率。若脉冲信号取自旋转编码器，可求得转速。

4、注意事项

（1）指定的 S1 输入的 X0~X3 不能同时使用于高速计数器和输入中断。

（2）每个输入点只能使用一条该指令。

2.8 通信指令

两条通信指令用于方便组建特殊的通信网络，分别是 RS2（串行数据通信指令）、ADPRW（MODBUS 主站通信用指令）。

2.8.1 RS2/串行数据传送 2 指令

参考 3.4.2 章节。

2.8.2 ADPRW/MODBUS 通信主站用指令

参考 3.4.3 章节。

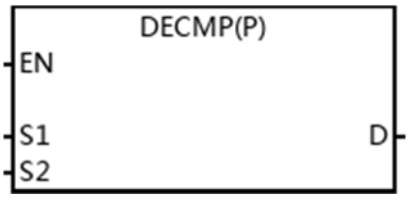
2.9 浮点运算指令

用于对精度要求比较高的浮点数数据分析计算，包含基本的浮点数四则运算、浮点数转换、三角函数等。

2.9.1 ECMP/浮点数比较指令

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

1、指令格式



操作数说明

操作数	内容	类型
S1	比较值 1	实数(2 进制)
S2	比较值 2	实数(2 进制)
D	比较结果	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2														*	*			*			*		
D		*	*			*												*					

2、功能和动作说明

32 位运算指令，将比较值（S1+1，S1）和比较源（S2+1，S2）作为浮点数数据进行比较，然后根据比较的结果(小于、等于、大于)将 D、D+1、D+2 中的任意一位置 ON。（S1 +1，S1）、（S2 +1，S2）中指定了常数(K、H)时，会自动将数值从 BIN 转换成 2 进制浮点数后再处理。

3、程序举例

可参考 CMP 指令。

4、注意事项

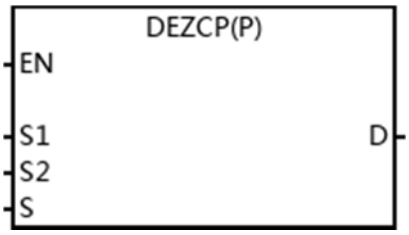
（1）D 占用 3 点，（D，D+1，D+2），注意不要与其他用途的软元件重复。

（2）即使指令输入 OFF，不执行 DECMP 指令，D～D+2 也能保持指令输入 OFF 之前的状态。

2.9.2 EZCP/2 进制浮点数区间比较指令

将上下 2 点的比较范围和数据(2 进制浮点数)进行比较，根据其结果输出到位软元件(3 点)中的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	左区间浮点数	实数(2 进制)
S2	右区间浮点数	实数(2 进制)
S	浮点数比较源	实数(2 进制)
D	输出结果的起始位软元件编号(占用 3 点)	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2														*	*			*			*		
S														*	*			*			*		
D		*	*			*												*					

2、功能和动作说明

32 位运算指令，将比较值(S1+1，S1)、(S2+1，S2)和比较源(S+1，S)作为浮点数数据进行比较，然后根据比较的结果(小于、等于、大于)将 D、D+1、D+2 中的任意一位置 ON。

3、程序举例

可参考 ZCP 指令。

4、注意事项

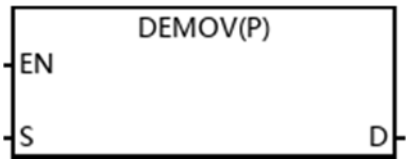
（1）比较数据的大小关系请设置为(S1+1,S)≦(S2+1，S2)。(S1+1,S) ＞(S2+1，S2) 的情况时，(S2+1，S2) 的数值视为与(S1+1,S)相同，从而进行比较。

（2）即使指令输入 OFF，不执行 DECMP 指令，D～D+2 也能保持指令输入 OFF 之前的状态。

2.9.3 EMOV/2 进制浮点数数据传送指令

传送 2 进制浮点数数据的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	传送源	实数(2 进制)
D	目标数据	实数(2 进制)

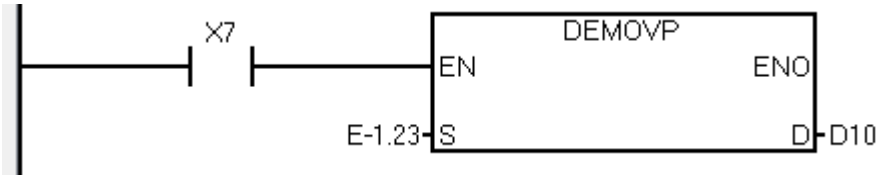
操作数可用元件类型

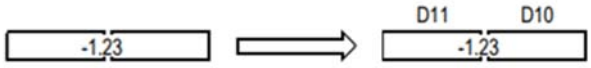
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将传送源（S+1，S）的内容(2 进制浮点数数据)传送到（D+1,D）中。3、程序举例

X007 为 ON 时，将实数-1.23 保存到 D11、D10 中的程序。





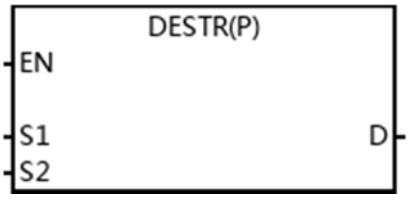
4、注意事项

注意与普通的 MOV 指令的区分。

2.9.4 ESTR/2 进制浮点数→字符串的转换指令

该指令用于将 2 进制浮点数数据转换成指定位数的字符串(ASCII 码)。

1、指令格式



操作数说明

操作数	内容	类型

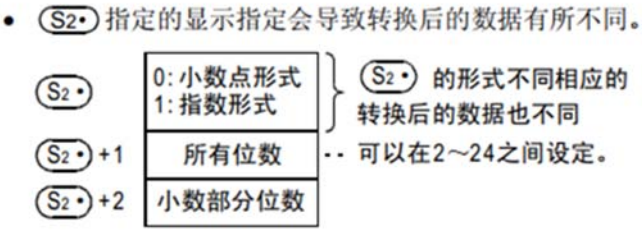
S1	传送源（2 进制浮点数）	实数(2 进制)
S2	指定转换形式的软元件起始编号	BIN16 位
D	目标数据（存储转换后的字符串）	字符串

操作数可用元件类型

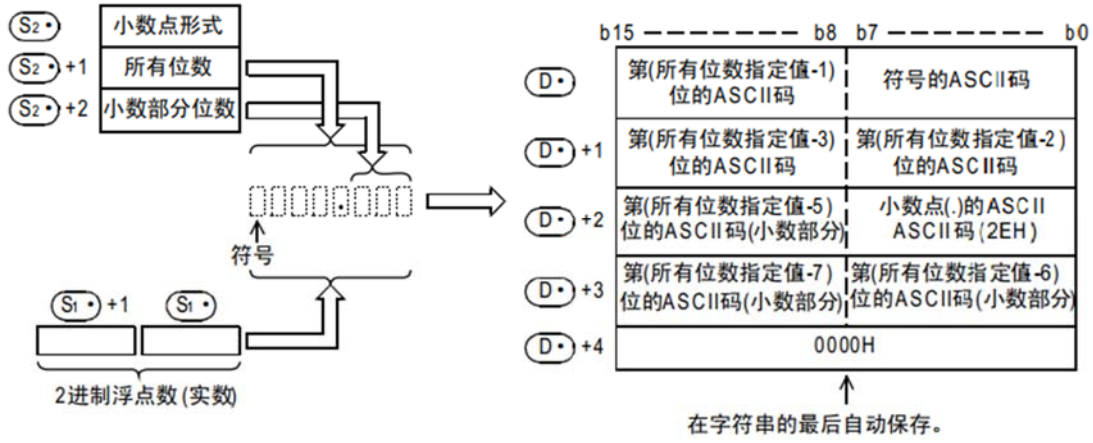
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

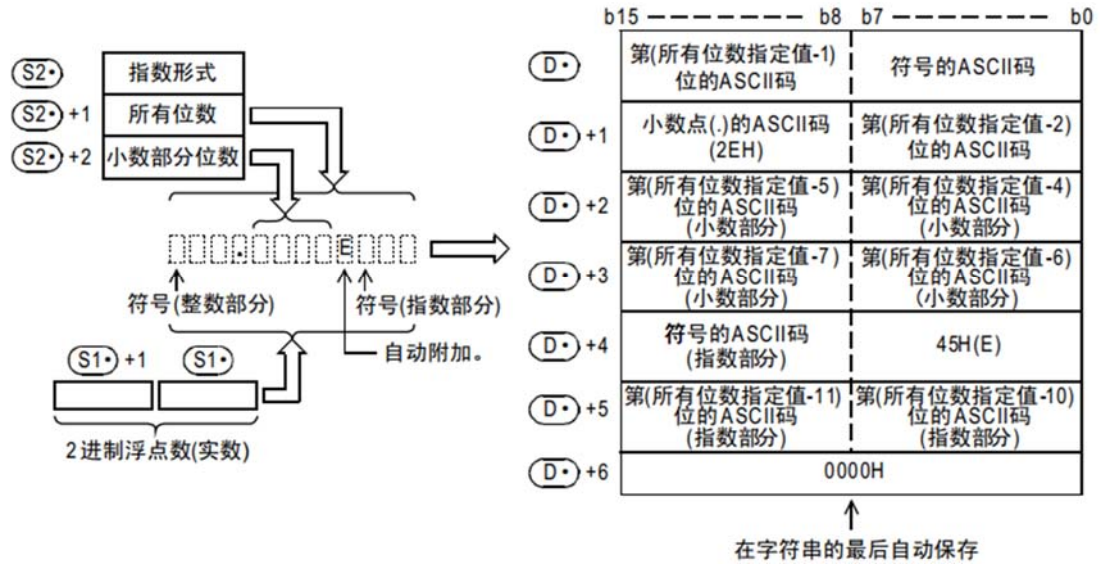
32 位运算指令，根据 S2,S2+1,S2+2 中指定的内容，将（S1+1，S1）的内容(2 进制浮点数数据）转换成字符串，并保存至 D 开始的软元件中。



（1）小数点形式时



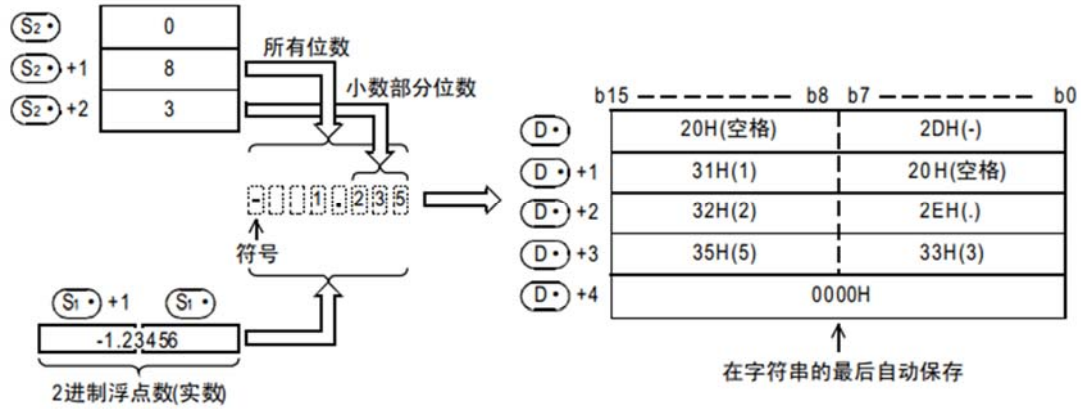
（2）指数形式时



3、程序举例

（1）小数点形式

例如，所有位数为 8，小数部分位数为 3，指定-1.23456 时，D 开始的软元件中会如下所示被保存。



• 字符中，2 进制浮点数数据为正时保存“20H”(空格)，为负时保存“2DH”(-)。

• 小数部分位数的范围中不能容纳 2 进制浮点数数据的小数部分时，小数低位部分被四舍五入。

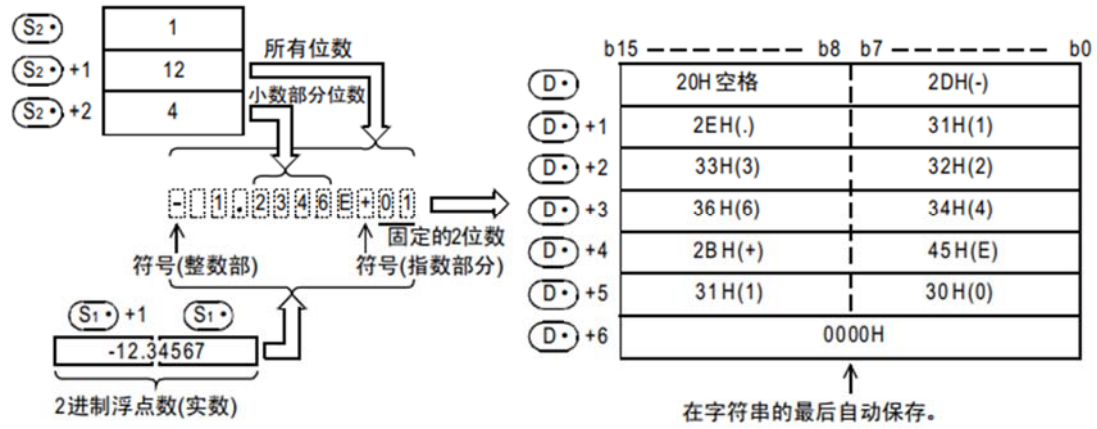
• 小数部分位数设定为“0”以外数字时，会自动将“2EH”(.)保存到指定的小数部分位数+1 的位中。但是，小数部分位数为"0"时，不保存"2EH"(.)。

• 当从所有位数中去除符号、小数点、小数部分的位数后，其位数比 2 进制浮点数数据的整数部分大的时候，在符号和整数部分之间保存“20H”(空格)。

• 在转换的字符串最后，自动保存“00H”或“0000H”。

（2）指数形式

例如，所有位数为 12，小数部分位数为 4，指定-12.34567 时，在 D 开始的软元件中会如下所示被保存。



• 在整数部分的符号中，2 进制浮点数数据为正时保存“20H”(空格)，为负时保存“2DH”(-)。

• 整数部分固定为 1 位数。在整数部分和符号之间保存“20H”(空格)。

• 在小数部分位数的范围中不能容纳 2 进制浮点数数据的小数部分时，小数低位部分被四舍五入。

• 小数部分位数设定为“0”以外数字时，会自动将“2EH”(.)保存到指定的小数部分位数+1 的位中。但是，小数部分位数为“0”时，不保存“2EH”(.)。

• 在指数部分的符号中，指数为正时保存“2BH”(+)，指数为负时保存“2DH”(-)。

• 指数部分固定为 2 位数。当指数部分为 1 位数时，在其与指数部分的符号之间中保存“30H”(0)。

• 在转换的字符串最后，自动保存“00H”或“0000H”。

4、注意事项

以下情况会出现运算错误：

（1）S1 不在范围 0，±2^-126≤S1＜±2^128 中时。

（2）S2 中指定的形式为 0、1 以外时。

（3）S2+1 中指定的所有位数指定不符合下面的范围时

小数点形式：

小数部分的位数为“0”时，所有位数≧2；

小数部分的位数为“0”以外的数字时，所有位数≧(小数部分位数+3)。

指数形式：

小数部分的位数为“0”时，所有位数≧6；

小数部分的位数为“0”以外的数字时，所有位数≧(小数部分位数+7)。

（4）S2+2 中指定的小数部分的位数指定不符合下面范围时

小数点形式时：小数点位数≦(所有位数-3)。

指数形式时：小数点位数≦(所有位数-7)。

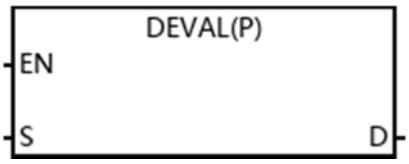
（5）当保存指定字符串的软元件 的范围，超出相应软元件的范围时。

（6）转换结果超出已指定的所有位数时。

2.9.5 EVAL/字符串→2 进制浮点数的转换指令

用于将字符串(ASCII 码)转换成 2 进制浮点数数据的指令。

1、指令格式



操作数说明

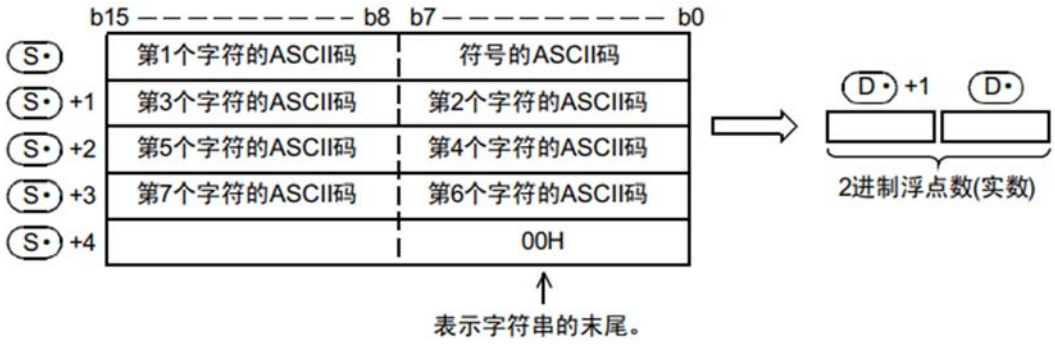
操作数	内容	类型
S	传送源（字符串数据）	字符串
D	目标数据（2 进制浮点数数据）	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D														*	*			*					

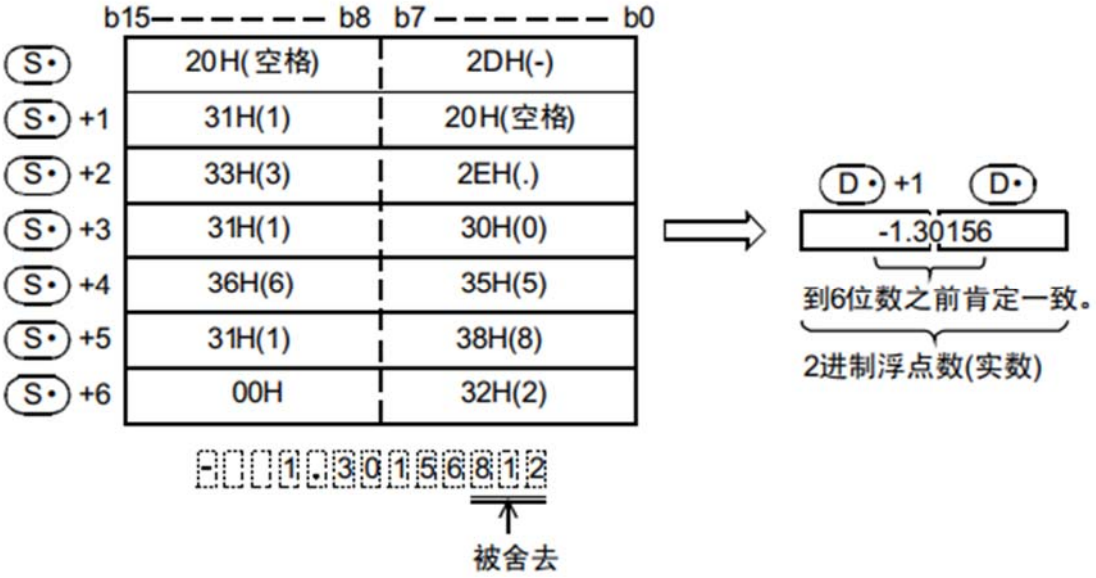
2、功能和动作说明

32 位运算指令，将 S 开始的软元件中保存的字符串转换成 2 进制浮点数数据后，保存到（D+1，D）中。指定的字符串无论是小数点形式，还是指数形式，都可以转换成 2 进制浮点数数据 。

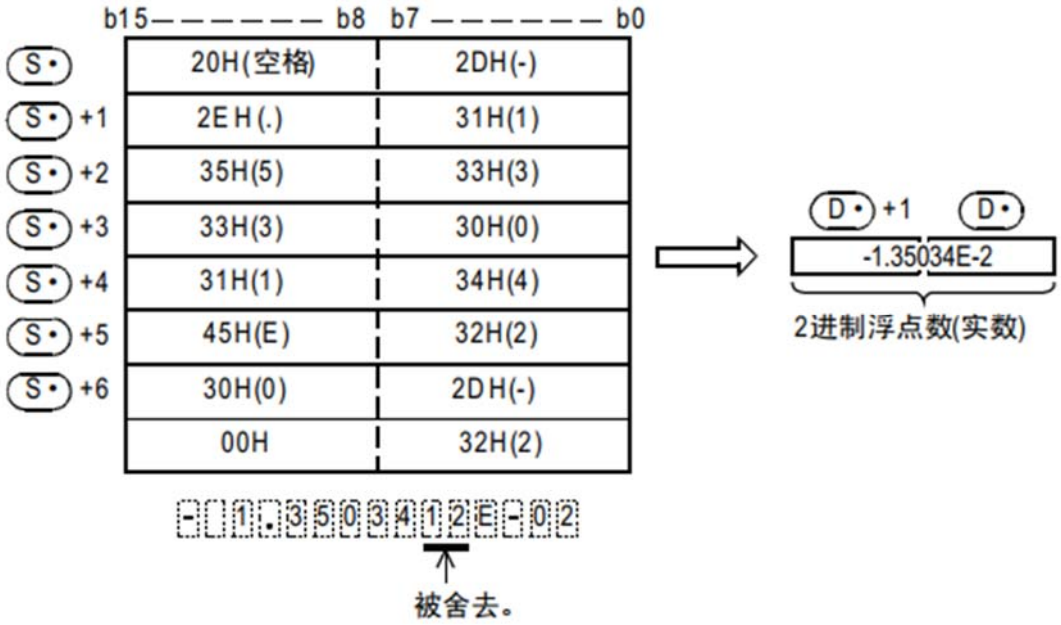


3、程序举例

（1）小数点形式



（2）指数形式



• S 中指定的，要转换成 2 进制浮点数的字符串，去除了符号、小数点、指数部分仍然有 7 位数以上时，舍去第 7 位数以后的数。

• 在小数点形式中，将符号指定为“2BH”(+)或省略符号，则作为正值转换。此外，将符号指定为"2DH"(-)则作为负的值转换。

• 用指数形式在指数部分的符号内指定"2BH"(+)，省略符号后作为正的指数转换。将指数部分的符号指定为"2DH"(-)则作为负的指数转换。

• 指定的字符串中，在最初的“0”以外的数值之间如果存在“20H”(空格)或是“30H”(0)时，会忽略“20H”、“30H”而进行转换。

• 在指数形式的字符串中，“E”和数值之间如果存在“30H”(0)，则忽略“30H”而进行转换。

•字符串最大可以设定到 24 个字符。字符串中的“20H”(空格)、“30H”(0)也作为一个字符来计算。

4、注意事项

（1）相关软元件

软元件	名称	内容	
		条件	动作
SM8080	零位	转换结果真的为零 (尾数部分为“0”时)	零位标志位(SM8080)为 ON。
SM8081	借位	转换结果的绝对值＜2^-126	D 的值小于 32 位实数的最小值(2^-126)部分被舍去，借位 标志位(SM8081)为 ON。

SM8082	进位	转换结果的绝对值≧2^128	的值大于 32 位实数的最大值(2^128)部分被舍去，进位 标志位(SM8082)为 ON。

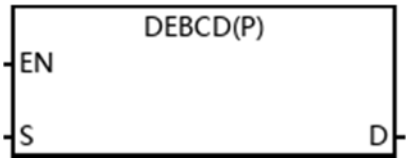
（2）以下情况会出现运算错误，报错 6706

• 整数部分、小数部分中存在“30H”(0)～“39H”(9)以外的字符时。
• S 指定的字符串中存在 2 个以上的“2EH”(.)时。
• 指数部分中有“45H”(E)、“2BH”(+)、“2DH”(-)以外的字符或是有多个指数部分时。
• S 开始的相应软元件范围内没有“00H”时。
• S 以后的字符数为 0 或是超出了 24 个字符时。

2.9.6 EBCD/2 进制浮点数→10 进制浮点数的转换指令

将软元件中的 2 进制浮点数转换成→10 进制浮点数的指令。

1、指令格式



操作数说明

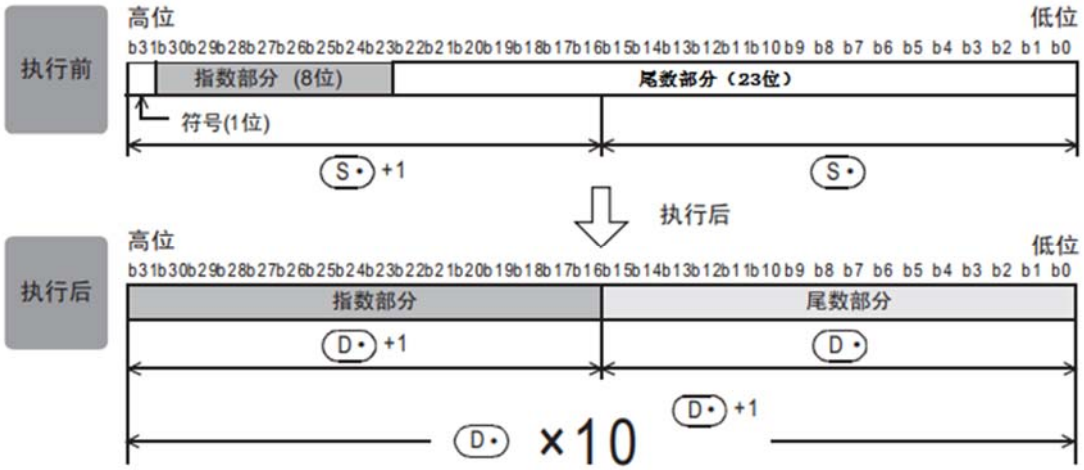
操作数	内容	类型
S	源数据（2 进制数据）	实数(2 进制)
D	目标数据（10 进制浮点数数据）	实数(10 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 2 进制浮点数转换成 10 进制浮点数后，保存到（D+1，D）中。



3、程序举例

略。

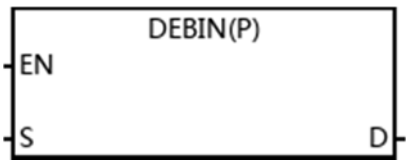
4、注意事项

在浮点数运算中，都以 2 进制浮点数执行。但是，由于 2 进制浮点数本身是不易理解的数值(专用的监控方法)，所以转换成 10 进制浮点数运算后，可以便于在外围设备上进行监控等。

2.9.7 EBIN/10 进制浮点数→2 进制浮点数的转换指令

将软元件中的 10 进制浮点数转换成 2 进制浮点数的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存 10 进制浮点数数据的数据寄存器编号	实数(10 进制)

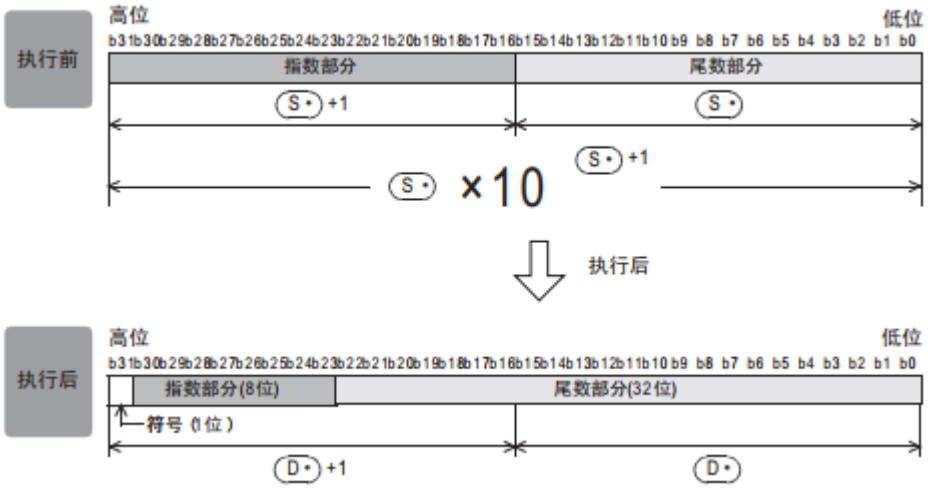
D	保存被转换的 2 进制浮点数数据的数据寄存器编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 10 进制浮点数转换成 2 进制浮点数后，保存到（D+1，D）中。



3、程序举例

无

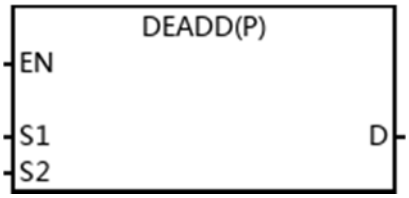
4、注意事项

注意 2 进制浮点数的存储方式与 10 进制的区别。

2.9.8 EADD/2 进制浮点数加法运算指令

2 个 2 进制浮点数加法运算的指令。

1、指令格式



操操作数说明

操作数	内容	类型
S1	保存进行加法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行加法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存加法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将（S1+1，S1）和（S2+1，S2)的 2 进制浮点数数据进行加法运算，并将其运算结果以 2 进制浮点数形式传送到(D+1，D)中。

[(S1·) +1, (S1·)]+[(S2·) +1, (S2·)]→[(D·) +1, (D·)]
2进制浮点数(实数)2进制浮点数(实数)2进制浮点数(实数)

3、程序举例

无

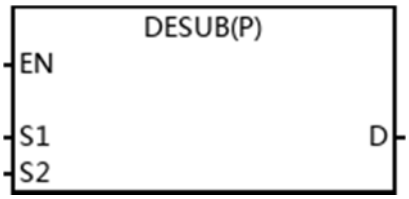
4、注意事项

（S1+1，S1）以及（S2+1，S2）和(D+1,D)中可以指定同一软元件，但是使用连续性指令 DEADD，则每个周期结果都会改变。

2.9.9 ESUB/2 进制浮点数减法运算指令

2 个 2 进制浮点数减法运算的指令。

1、指令格式



操作数说明

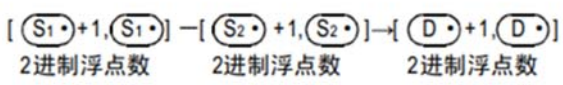
操作数	内容	类型
S1	保存进行减法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行减法的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存减法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将[S1+1，S1]减去[S2+1，S2]的 2 进制浮点数数据，并将其运算结果以 2 进制浮点数形式传送到[D+1，D]中。



3、程序举例

无

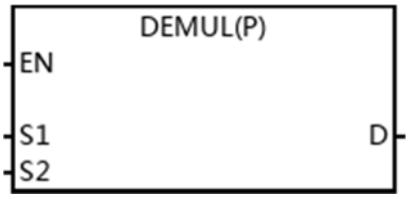
4、注意事项

同 EADD。

2.9.10 EMUL/2 进制浮点数乘法运算指令

2 个 2 进制浮点数乘法运算的指令。

1、指令格式



操作数说明

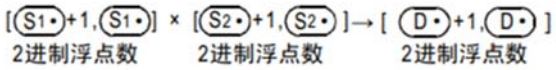
操作数	内容	类型
S1	保存进行乘法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行乘法的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存乘法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2														*	*			*			*		
D														*	*			*					

2、功能和动作说明

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



3、程序举例

无

4、注意事项

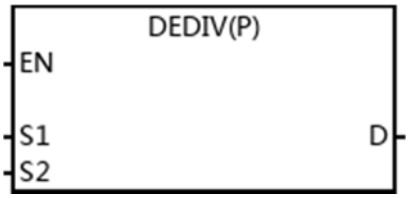
运算结果超过单精度浮点数范围时，对应的标志位置位

软元件	名称	内容	
		条件	动作
SM8080	零位	转换结果真的为零 (尾数部分为“0”时)	零位标志位(SM8080)为 ON。
SM8081	借位	转换结果的绝对值＜2^-126	D 的值小于 32 位实数的最小值(2^-126)部分被舍去，借位 标志位(SM8081)为 ON。
SM8082	进位	转换结果的绝对值≧2^128	的值大于 32 位实数的最大值(2^128)部分被舍去，进位 标志位(SM8082)为 ON。

2.9.11 EDIV/2 进制浮点数除法运算指令

2 个 2 进制浮点数除法运算的指令。

1、指令格式



操作数说明

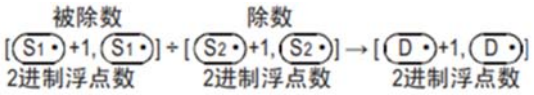
操作数	内容	类型
S1	保存进行除法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行除法的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存除法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2														*	*			*			*		
D														*	*			*					

2、功能和动作说明

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



3、程序举例

无

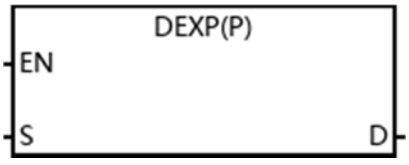
4、注意事项

运算结果超过单精度浮点数范围时，对应的标志位 SM8080，SM8081，SM8082 的动作。

2.9.12 EXP/2 进制浮点数指数运算指令

该指令是以 e(2.71828)为底的指数运算指令。

1、指令格式



操作数说明

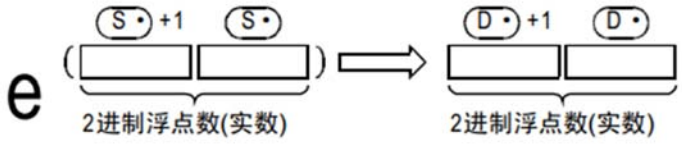
操作数	内容	类型
S	保存执行指数运算的 2 进制浮点数数据的软元件起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，以[S+1，S1]为指数做运算，并将其运算结果保存到[D+1，D]中。



3、程序举例

无

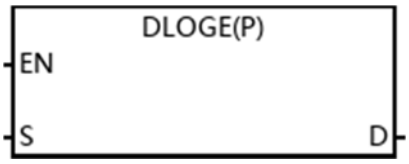
4、注意事项

结果不在 2^-126≦｜运算结果｜＜2^128 范围内，报运算错误 6706。

2.9.13 LOGE/2 进制浮点数自然对数运算指令

该指令执行自然对数运算。

1、指令格式



操作数说明

操作数	内容	类型
S	保存进行自然对数运算的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)

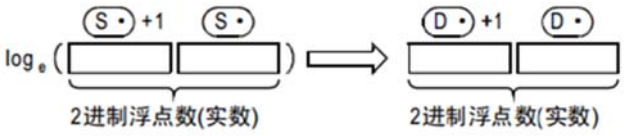
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，执行[S+1，S]的自然对数[以 e(2.17828) 为底时的对数] 运算,并将其运算结果保存到[D+1，D]中。



3、程序举例

无

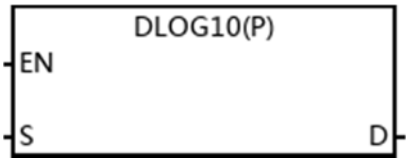
4、注意事项

（S+1，S）中指定的值只可以设定正数，指定为负数或 0 时报错 6706。

2.9.14 LOG10/2 进制浮点数常用对数运算指令

该指令执行常用对数运算。

1、指令格式



操作数说明

操作数	内容	类型

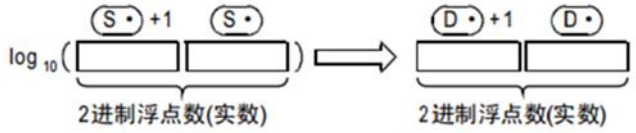
S	保存进行常用对数运算的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，执行[S+1，S]的常用对数(10 为底时的对数)运算,并将其运算结果保存到[D+1，D]中。



3、程序举例

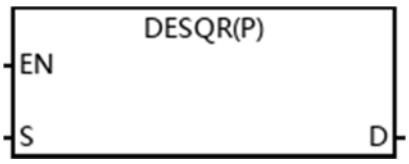
4、注意事项

（S+1，S）中指定的值只可以设定正数，指定为负数或 0 时报错 6706。

2.9.15 ESQR/2 进制浮点数开方运算指令

2 进制浮点数开方(开根号)运算的指令。

1、指令格式



操作数说明

操作数	内容	类型

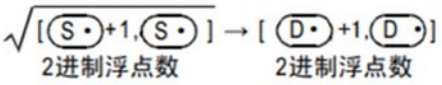
S	保存执行开方运算的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，执行[S+1，S]进行开方(开根号)运算(2 进制浮点数运算)后,将其运算结果保存到[D+1，D]中。



3、程序举例

略。

4、注意事项

（1）（S+1，S）值为正数有效，为负数时报错 6706。

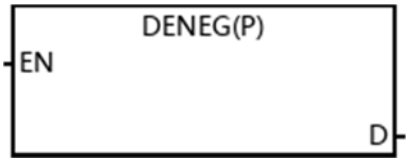
（2）运算结果为 0 时 SM8080 置位。

软元件	名称	内容
SM8080	零位	运算结果真为 0 时 ON

2.9.16 ENEG/2 进制浮点数符号翻转指令

使 2 进制浮点数(实数)数据的符号翻转的指令。

1、指令格式



操作数说明

操作数	内容	类型
D	保存要执行符号翻转的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)

操作数可用元件类型

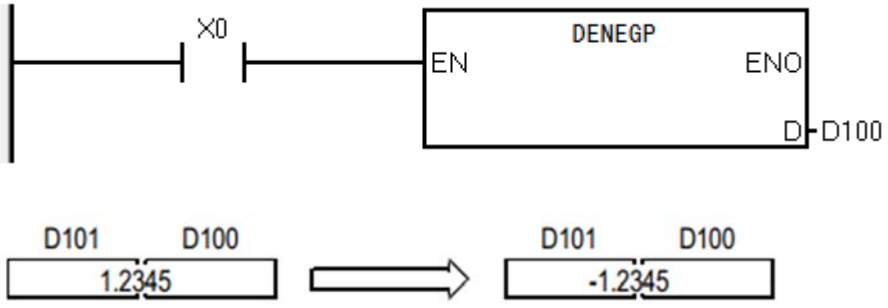
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D														*	*			*					

2、功能和动作说明

32 位运算指令，[D+1，D]的 2 进制浮点数数据的符号翻转，保存在[D+1，D]中。

3、程序举例

X000 为 ON 时，将 D100、D101 的 2 进制浮点数数据的符号翻转，并且保存到 D100、D101 中的程序。



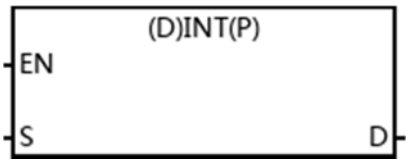
4、注意事项

无

2.9.17 INT/2 进制浮点数→BIN 整数的转换指令

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

1、指令格式



操作数说明

操作数	内容	类型
S	保存要转换成 BIN 整数的 2 进制浮点数数据的数据寄存器编号	实数(2 进制)
D	保存转换后的 BIN 整数的数据寄存器编号	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

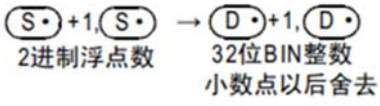
（1）16 位指令

[S+1，S]的 2 进制浮点数转换成 BIN 整数后，传送到 D 中。



（2）32 位指令

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



3、程序举例

无

4、注意事项

（1）相关标志位的动作

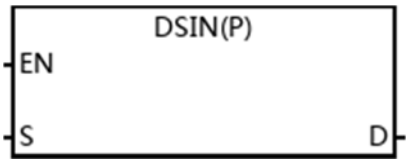
软元件	名称	内容
SM8080	零位	运算结果真为 0 时 ON。
SM8081	借位	位转换时，有因不满 1 而被舍去的情况发生时，置 ON。
SM8082	进位	运算结果超出-32,768～32,767(16 位运算时)，或是-2,147,483,583～2,147,483,583(32 位运算时)的范围而产生溢出时为 ON。(运算结果不反映。)

（2）小数点以后的值被舍去

2.9.18 SIN/2 进制浮点数 SIN 运算指令

求角度(RAD)的 SIN 值的指令。

1、指令格式



操作数说明

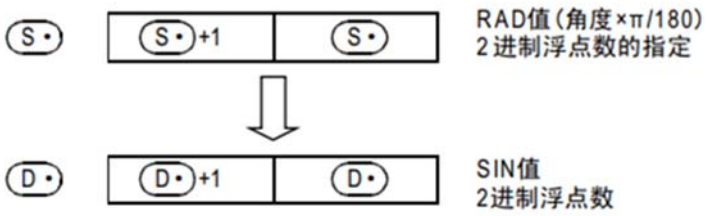
操作数	内容	类型
S	保存 2 进制浮点数的 RAD(角度)的软元件编号	实数(2 进制)
D	保存 2 进制浮点数的 SIN 值的软元件编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

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



3、程序举例

无

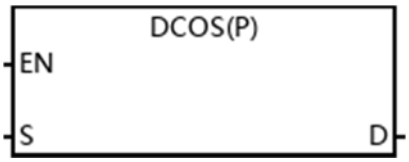
4、注意事项

运算结果为 0 时，SM8080 置位。

2.9.19 COS/2 进制浮点数 COS 运算指令

求角度(RAD)的 COS 值的指令。

1、指令格式



操作数说明

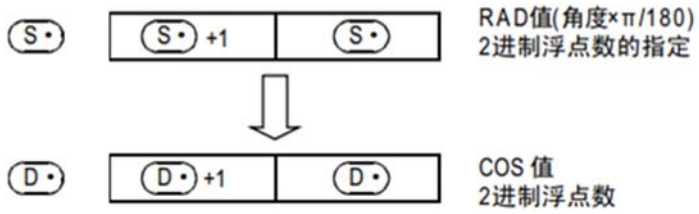
操作数	内容	类型
S	保存 2 进制浮点数的 RAD(角度)的软元件编号	实数(2 进制)
D	保存 2 进制浮点数的 COS 值的软元件编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

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



3、程序举例

无

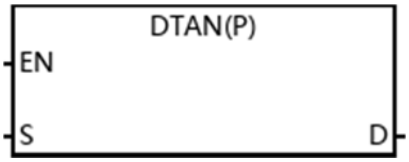
4、注意事项

无

2.9.20 TAN/2 进制浮点数 TAN 运算指令

求角度(RAD)的 TAN 值的指令。

1、指令格式



操作数说明

操作数	内容	类型

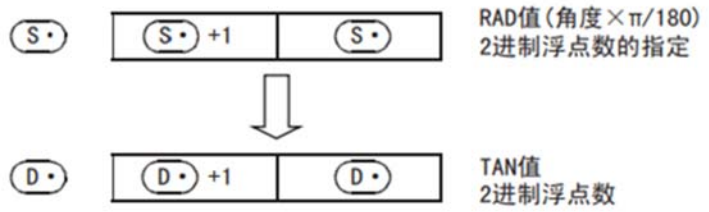
S	保存 2 进制浮点数的 RAD(角度)的软元件编号	实数(2 进制)
D	保存 2 进制浮点数的 TAN 值的软元件编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

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



3、程序举例

无

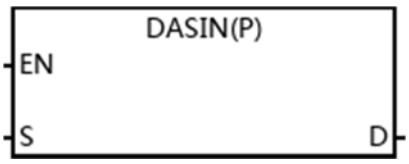
4、注意事项

运算结果为 0 时，SM8080 置位。

2.9.21 ASIN/2 进制浮点数 SIN-1 运算指令

执行 SIN-1 运算的指令。

1、指令格式



操作数说明

操作数	内容	类型

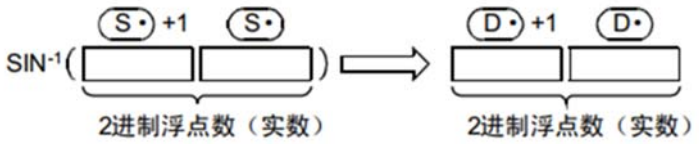
S	保存执行 SIN^{-1}(反正弦)运算的 SIN 值的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,[S+1，S]的 SIN 值求出角度，将运算结果保存到[D+1，D]中。此外，可以在 S 中直接指定实数。



3、程序举例

无

4、注意事项

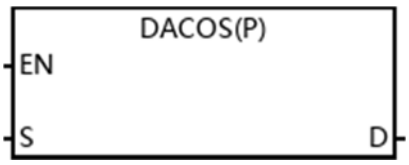
（1）（S+1，S）的 SIN 值，设定在-1.0~1.0 范围，不在此范围报错 6706。

（2）（D+1，D）中保存的角度是弧度值（-π/2）~（π/2）。

2.9.22 ACOS/2 进制浮点数 COS-1 运算指令

执行 COS-1 运算的指令。

1、指令格式



操作数说明

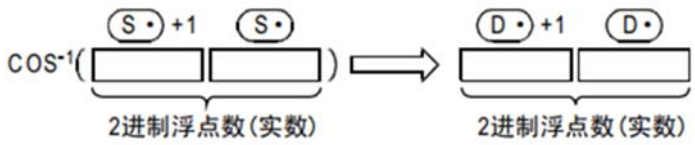
操作数	内容	类型
S	保存执行 COS^{-1}(反余弦)运算的 COS 值的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,[S+1，S]的 COS 值求出角度，将运算结果保存到[D+1，D]中。此外，可以在 S 中直接指定实数。



3、程序举例

无

4、注意事项

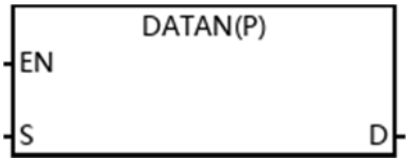
（1）（S+1，S）的 COS 值，设定在-1.0~1.0 范围，不在此范围报错 6706。

（2）（D+1，D）中保存的角度是弧度值 0~π。

2.9.23 ATAN/2 进制浮点数 TAN-1 运算指令

执行 TAN-1 运算的指令。

1、指令格式



操作数说明

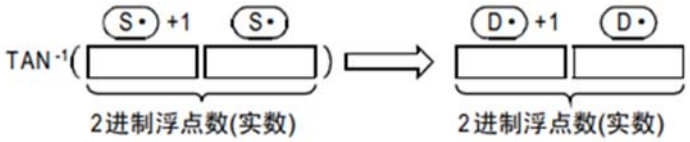
操作数	内容	类型
S	保存执行 TAN^{-1}(反正切)运算的 TAN 值的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,[S+1，S]的 TAN 值求出角度，将运算结果保存到[D+1，D]中。此外，可以在 S 中直接指定实数。



3、程序举例

无

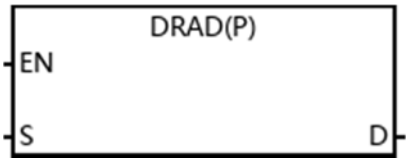
4、注意事项

（D+1，D）中保存的角度是弧度值>(-π/2)或<(π/2)。

2.9.24 RAD/2 进制浮点数角度→弧度的转换指令

将角度单位的值转换成弧度单位的指令。

1、指令格式



操作数说明

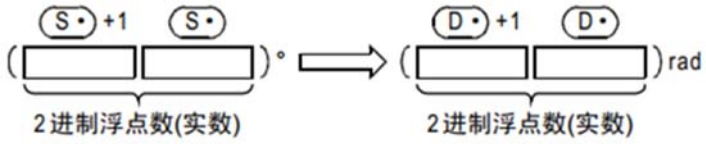
操作数	内容	类型
S	保存要转换成弧度单位的角度的软元件起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,[S+1，S]的单位从角度单位转换成弧度单位 ，将运算结果保存到[D+1，D]中。此外，可以在 S 中直接指定实数。弧度单位 = 角度单位*π/180。



3、程序举例

无

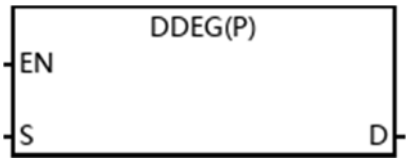
4、注意事项

无

2.9.25 DEG/2 进制浮点数弧度→角度的转换指令

将弧度单位的值转换成角度(DEG)单位的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存要转换成角度单位的弧度的软元件起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,[S+1，S]的单位从弧度单位转换成角度单位 ，将运算结果保存到[D+1，D]中。



3、程序举例

无

4、注意事项

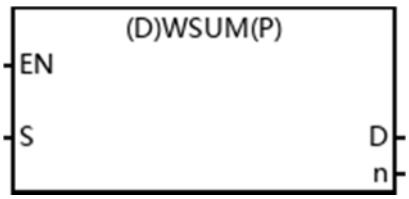
无

2.10 数据处理 2 指令

2.10.1 WSUM/算出数据合计值指令

该指令可计算出连续的 16 位或是 32 位数据的合计值。

1、指令格式



操作数说明

操作数	内容	类型
S	保存要算出合计值的数据的软元件起始编号	BIN16/32 位
D	保存合计值的软元件起始编号	BIN32/64 位
n	数据个数(0＜n)	BIN16/32 位

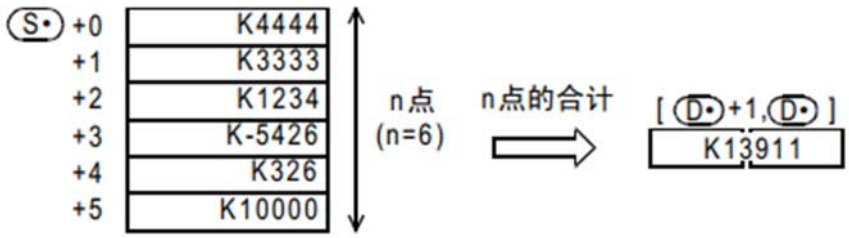
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

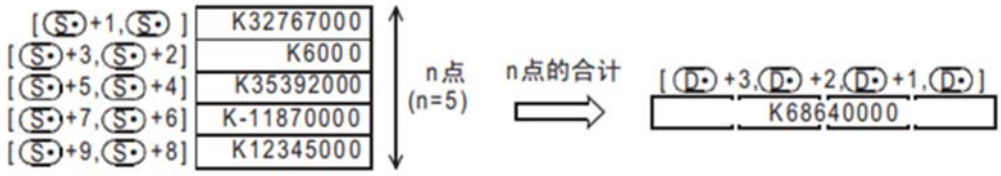
（1）16 位指令

将 S 开始的 n 点 16 位数据的合计值，以 32 位数据形式保存在[D+1，D]中。



（2）32 位指令

将（S+1，S）开始的 n 点 32 位数据的合计值，以 64 位数据形式保存在（D+3，D+2，D+1，D）中。



3、程序举例

无

4、注意事项

（1）数据个数设定≤0 时，报错 6706。

（2）注意运算结果，16 位指令时，超过 16 位数的情况下，查看 D+1，D 分别是运算结果的高 16 位、低 16 位，未超过查看 D 即可；32 位指令时，超过的情况下，查看（D+3，D+2），（D+1，D）分别是运算结果的高 32 位、低 32 位，未超过时查看（D+1,D ）即可。

2.10.2 WTOB/字节单位的数据分离指令

将连续的 16 位数据按照字节(8 位)单位进行分离的指令。

1、指令格式



操作数说明

操作数	内容	类型

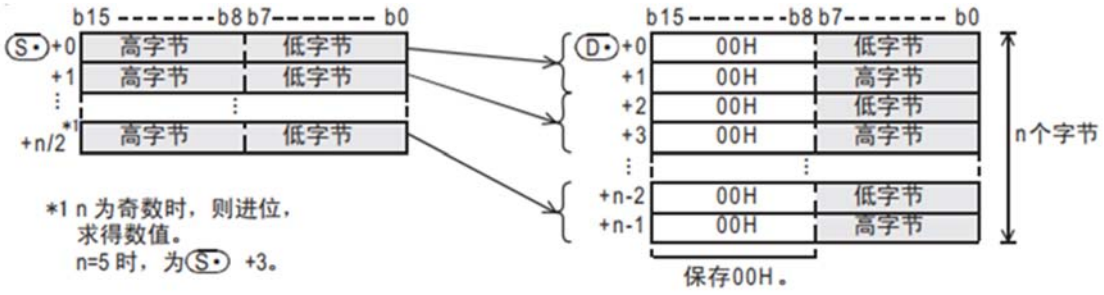
S	保存要按照字节单位进行分离的数据的软元件起始编号	BIN16 位
D	保存已经按照字节单位分离的结果的软元件起始编号	BIN16 位
n	要分离的字节数据个数(0≦n)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

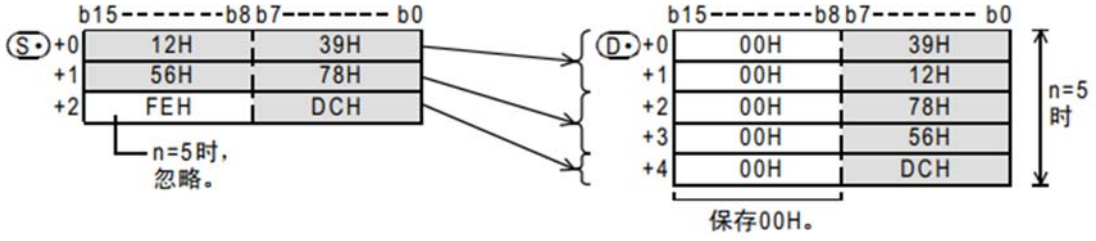
2、功能和动作说明

（1）16 位指令，将 S 开始的 n/2 个软元件中保存的 16 位数据分离成 n 个字节，如下所示保存到以 D 开始的 n 点软元件中。



（2）在保存分离后字节数据的软元件(D 以后)的高字节(8 位)中，保存 00H。

（3）n 为奇数时，如下图所示，在分离源的最终数据中，只有低字节(8 位)为对象数据。



3、程序举例

无

4、注意事项

（1）当分离源软元件的 S～(S+n/2)超出了指定软元件的软元件范围时。 n 为奇数时，需要占用进位后数值的个数部分的软元件，报错 6706。

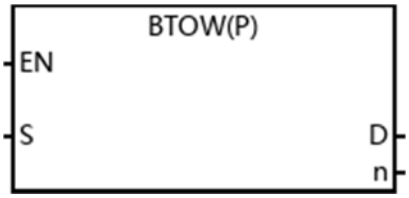
（2）当保存已分离的数据的软元件 D～(D+n-1)超出了指定软元件的软元件范围时，报错 6706。

（3）n=0 时，不执行指令。

2.10.3 BTOW/字节单位的数据结合指令

将连续的 16 位数据的低 8 位(低字节)结合在一起的指令。

1、指令格式



操作数说明

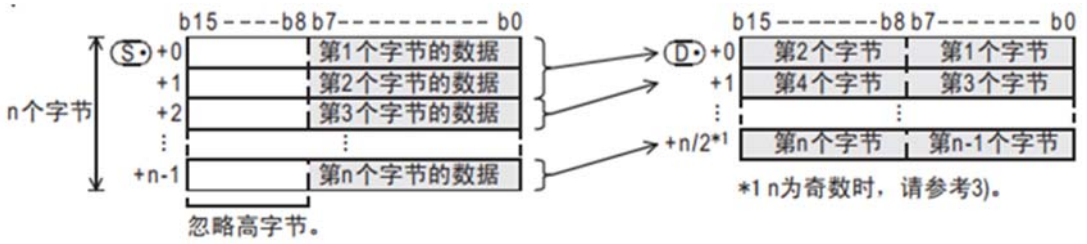
操作数	内容	类型
S	保存要按照字节单位结合的数据的软元件起始编号	BIN16 位
D	保存已经按照字节单位结合的结果的软元件起始编号	BIN16 位
n	要结合的字节数据个数(0≦n)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

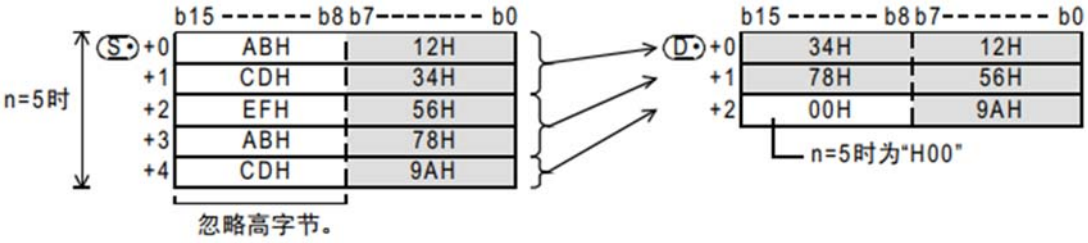
2、功能和动作说明

（1）16 位指令，将 S 开始的 n 点 16 位数据的低字节(8 位)结合在一起后的 16 位数据，如下所示保存到以 D 开始的 n/2 点软元件中。



（2）结合源的 16 位数据(S 以后)的高字节(8 位)被忽略。

（3）n 为奇数时，如下图所示，最终结合后的数据的高字节(8 位)为 00H。



3、程序举例

无

4、注意事项

（1）结合源的软元件的 S～(S+n-1)中指定的软元件超出了该软元件范围时，报错 6706。

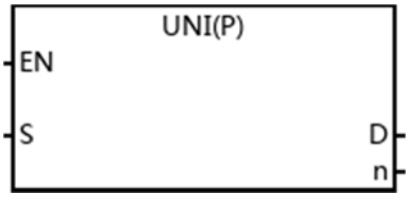
（2）当保存已结合数据的软元件 D～(D +n/2) 超出了指定软元件的软元件范围时。当 n 为奇数时，需要占用进位后数值的个数部分的软元件，报错 6706。

（3）n=0 时，不执行指令。

2.10.4 UNI/16 位数据的 4 位结合指令

将连续的 16 位数据的低 4 位结合在一起的指令。

1、指令格式



操作数说明

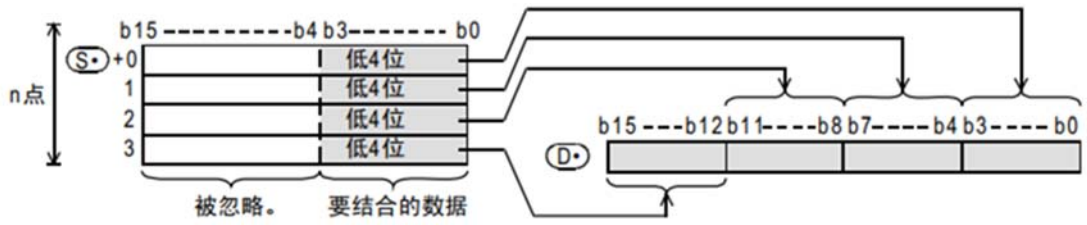
操作数	内容	类型
S	保存要结合的数据的软元件起始编号	BIN16 位
D	保存已结合的数据的软元件编号	BIN16 位
n	结合数(0～4，n=0 时不处理)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

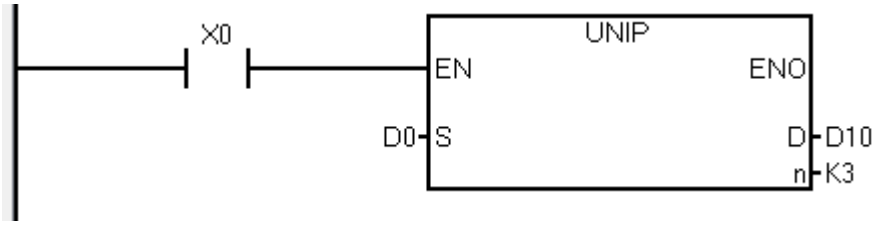
2、功能和动作说明

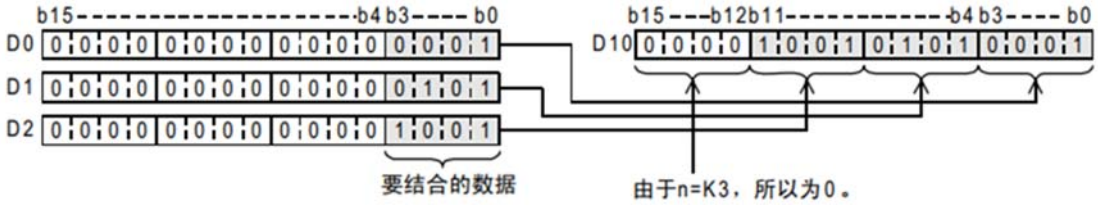
16 位指令，将 S 开始的 n 点 16 位数据的低 4 位结合后的 16 位数据，如下所示保存到 D 中。



3、程序举例

X000 为 ON 后，将 D0~D2 的低 4 位结合后，保存到 D10 中。





4、注意事项

注意连续执行指令时会一直执行结合。

2.10.5 DIS/16 位数据的 4 位分离指令

将连续的 16 位数据的低 4 位结合在一起的指令。

1、指令格式



操作数说明

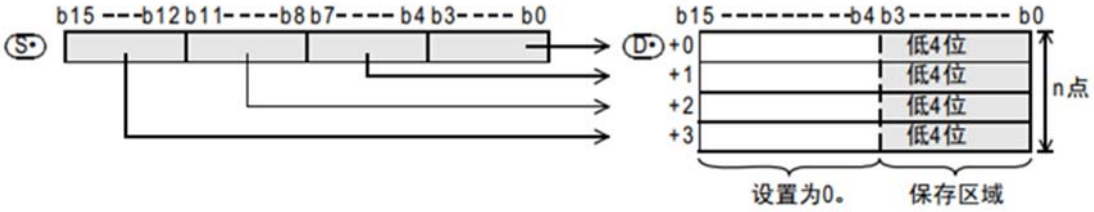
操作数	内容	类型
S	保存要分离的数据的软元件起始编号	BIN16 位
D	保存已分离的数据的软元件编号	BIN16 位
n	分离数(0～4，n=0 时不处理)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令，将 S 开始的 16 位数据以 4 位结合后为单位分离，如下所示保存到 D 中。



（2）n 范围(0～4，n=0 时不处理)

（3）D 开始的 n 点软元件的高 12 位设置为 0。

3、程序举例

略。

4、注意事项

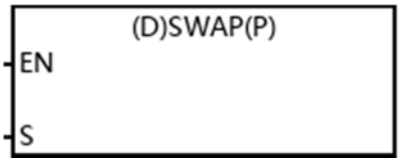
（1）当 D 开始的 n 点软元件超出了知道软元件的范围时，报错 6706。

（2）n 指定的 0~4 意外数字报错 6706。

2.10.6 SWAP/高低字节互换指令

互换字数据的高 8 位和低 8 位的指令。

1、指令格式



操作数说明

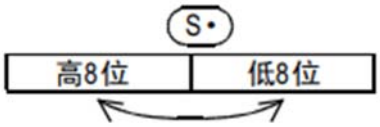
操作数	内容	类型
S	高低字节互换的字软元件	BIN16 位/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

（1）16 位指令



（2）32 位指令



3、程序举例

略。

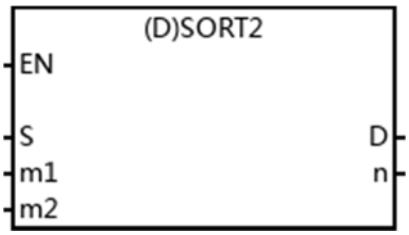
4、注意事项

即使是 32 位指令，也是执行各自的低 8 位和高 8 位的互换。与 XCH 指令的扩展功能相同。

2.10.7 SORT2/数据排序 2 指令

以指定的群数据(列)为基准，以行为单位，将由数据(行)和群数据(列)构成的数据表进行升序/降序重新排序的指令。在这个指令中，由于是在连续的软元件中保存数据(行方向)，所以便于增加数据(行)。

1、指令格式



操作数说明

操作数	内容	类型
S	保存数据表格的软元件起始编号[占用 m1×m2 点]	BIN16 位/32 位
m1	数据(行)数[1～32]	BIN16 位/32 位
m2	群数据(列)数[1～6]	BIN16 位/32 位
D	保存运算结果的软元件起始编号[占用 m1×m2 点]	BIN16 位/32 位
n	作为排序标准的群数据(列)的列编号[1～m2]	BIN16 位/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*								
m1														*	*				*	*			
m2																			*	*			
D														*	*								
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

针对 开始的(m1×m2)点的数据表格(排序前)，以 n 列的群数据为基准，将数据行进行升序或是降序的排列，然后保存到从 开始的(m1×m2)点的数据表格(排序后)中。

（2）32 位指令

针对(S+1,S)开始的(m1×m2)点的数据表格(排序前)，以 n 列的群数据为基准，将数据行进行升序或是降序的排列，然后保存到从(D+1,D)开始的(m1×m2)点的数据表格(排序后)中。

3、程序举例

在“n=K2(列号 2)”和“n=K3(列号 3)”的情况下，对如下所示的排序前的数据进行排序，动作如下所示。下面例举了 16 位运算的动作例子。执行 32 位运算时，请使用 BIN32 位构成数据表格。此外，如果先在第 1 列中输入管理编号等连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前数据：

列号 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	S	S+1	S+2	S+3
		1	150	45	20
	2	S+4	S+5	S+6	S+7
		2	180	50	40
	3	S+8	S+9	S+10	S+11
		3	160	70	30
	4	S+12	S+13	S+14	S+15
		4	100	20	8
	5	S+16	S+17	S+18	S+19
		5	150	50	45

以 n=K2(列号 2)为基准执行指令时的排序结果(升序的情况)

列号 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+1	D+2	D+3
		4	100	20	8
	2	D+4	D+5	D+6	D+7
		1	150	45	20

	3	D+8	D+9	D+10	D+11
		5	150	50	45
	4	D+12	D+13	D+14	D+15
		3	160	70	30
	5	D+16	D+17	D+18	D+19
		2	180	50	40

以 n=K3(列号 3)为基准执行指令时的排序结果(降序的情况)

列号 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+1	D+2	D+3
		3	160	70	30
	2	D+4	D+5	D+6	D+7
		2	180	50	40
	3	D+8	D+9	D+10	D+11
		5	150	50	45
	4	D+12	D+13	D+14	D+15
		1	150	45	20
	5	D+16	D+17	D+18	D+19
		4	100	20	8

4、注意事项

（1）相关软元件的动作

SM8029：指令结束标志，排序结束时置位；

SM8092：置 ON 时降序排列，OFF 时升序排列。

（2）以下操作会引起错误排序，请注意

- 动作过程中，请勿使操作数和数据的内容变化。
- 再次执行时，请将指令输入 OFF 一次。

- 指令的使用次数的限制在程序中最多可同时驱动 2 次。
- 原来的数据和排序替换后的数据，请错开，不要重叠。

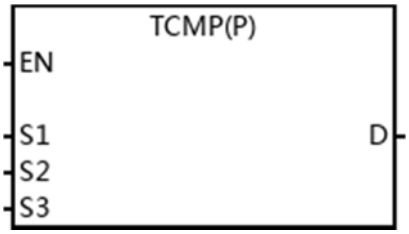
2.11 时钟运算指令

针对时钟数据进行运算、比较的指令，还可以修改内置时钟时间。

2.11.1 TCMP/时钟数据比较指令

将比较基准时间和时间数据进行大小比较，根据比较的结果控制位软元件 ON/OFF。。

1、指令格式



操作数说明

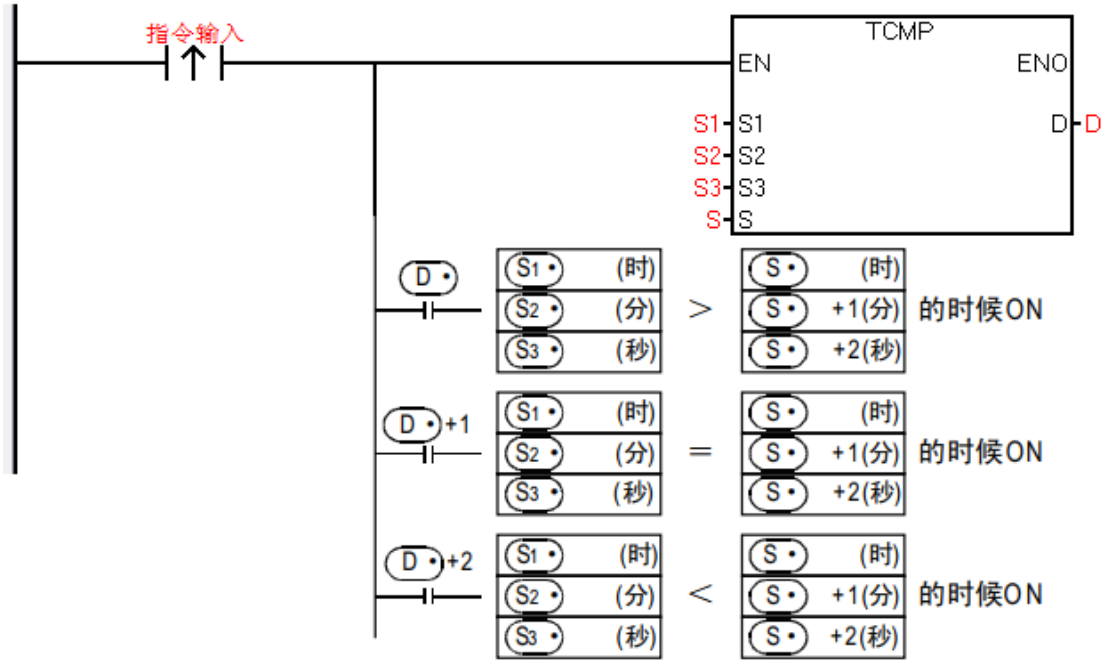
操作数	内容	类型
S1	指定比较基准时间的“时”。[设定范围: 0～23]	BIN16 位
S2	指定比较基准时间的“分”。[设定范围: 0～59]	BIN16 位
S3	指定比较基准时间的“秒”。[设定范围: 0～59]	BIN16 位
S	指定时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	根据比较结果 ON/OFF 位软元件。(占用 3 点)	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S3								*	*	*	*	*	*	*	*	*	*	*	*	*			
D1												*	*	*	*			*					
D2		*	*			*												*					

2、功能和动作说明

将比较基准时间(时、分、秒)[S1，S2 ，S3] 的时间与时间数据(时、分、秒)[S ，S+1，S+2]进行大小比较，根据其大小一致的结果 ON/OFF D 开始的 3 点。



3、程序举例

略。

4、注意事项

（1）即使是指令输入为 OFF,TCMP 指令不执行时，D ～D +2 也会保持当指令输入

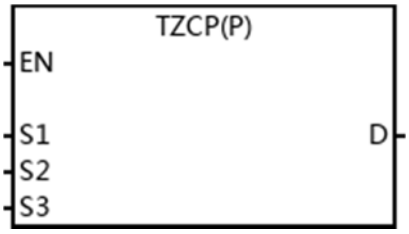
从 ON 变为 OFF 之前的状态。

（3）使用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时，请使用 TRD(FNC 166)指令，读出特殊数据寄存器的值以后，在各个操作数中指定其字软元件。

2.11.2 TZCP/时钟数据区间比较指令

将上下 2 点的比较基准时间和时间数据进行大小比较，根据比较的结果控制指定位软元件的 ON/OFF。

1、指令格式



操作数说明

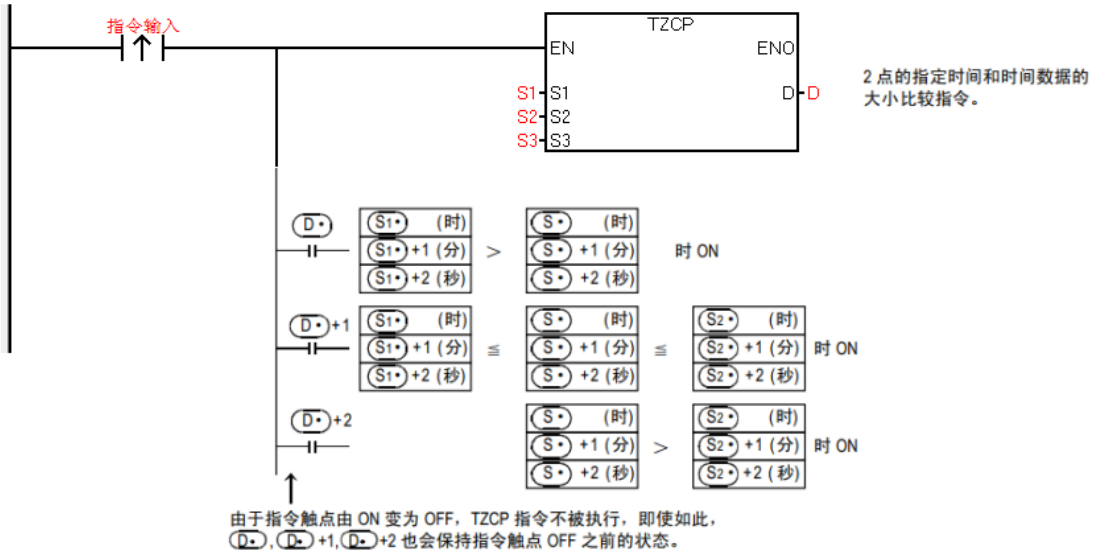
操作数	内容	类型
S1	指定比较下限时间(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S2	指定比较上限时间(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S3	指定时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	根据比较结果 ON/OFF 位软元件。(占用 3 点)	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S3								*	*	*	*	*	*	*	*	*	*	*	*	*			
D2		*	*			*												*					

2、功能和动作说明

将上下 2 点比较基准时间(时、分、秒)与以 S3 开头的 3 点时间数据(时、分、秒)进行比较，根据比较的结果 ON/OFF 从 D 开始的 3 点软元件。



3、程序举例

略。

4、注意事项

用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时，请使用 TRD(FNC 166)指令，读出特殊数据寄存器的值以后，在各个操作数中指定其字软元件。

2.11.3 TADD/时钟数据加法运算指令

将 2 个时间数据进行加法运算后，保存在字软元件中。

1、指令格式



操作数说明

操作数	内容	类型
S1	指定进行加法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S2	指定进行加法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位

D	保存两个时间数据(时、分、秒)加法运算的结果。(占用 3 点)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*					
D												*	*	*	*			*					

2、功能和动作说明

将[S1，S1+1，S1+2] 的时间数据(时、分、秒) 与[S2，S2+1，S2+2] 的时间数据(时、分、秒)进行加法运算，其结果保存到[D，D+1，D+2](时、分、秒)中。

“时”范围：0~23

“分”范围：0~59

“秒”范围：0~59

3、程序举例

略。

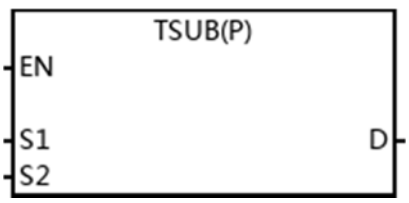
4、注意事项

主要相关标志位的动作：当运算结果超出 24 小时时，进位标志位变为 ON，从单纯的加法运算值中减去 24 个小时后将该时间作为运算。运算结果为 0(0 时 0 分 0 秒)时，零位标志位变为 ON。

2.11.4 TSUB/时钟数据减法运算指令

将 2 个时间数据进行减法运算后，保存在字软元件中。

1、指令格式



操作数说明

操作数	内容	类型
S1	指定进行减法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S2	指定进行减法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	保存两个时间数据(时、分、秒)减法运算的结果。(占用 3 点)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*					
D																		*					

2、功能和动作说明

从[S1，S1+1，S1+2] 的时间数据(时、分、秒) 中减去[S2，S2+1，S2+2] 的时间数据(时、分、秒)，其结果保存到[D，D+1，D+2](时、分、秒)中。

“时”范围：0~23

“分”范围：0~59

“秒”范围：0~59

3、程序举例

略。

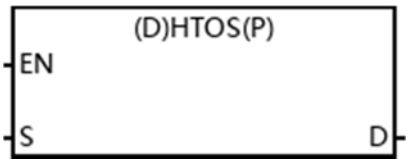
4、注意事项

相关标志位的动作：当运算结果小于 0 时，借位标志位变为 ON，从单纯的减法运算值中加上 24 个小时后，将该时间作为运算结果被保存。运算结果为 0(0 时 0 分 0 秒)时，零位标志位变为 ON。

2.11.5 HTOS/时、分、秒数据的秒转换指令

将[时、分、秒]单位的时间(时刻)数据转换成秒单位的数据的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存转换前的时间(时刻)数据(时、分、秒)的软元件的起始编号	BIN16 位
D	保存转换后的时间(时刻)数据(秒)的软元件编号。	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

（1）16 位指令

将[S，S+1，S+2]的时间(时刻)数据(时、分、秒)换算成秒后，将结果保存到 D 中。

S 范围：0~9

S+1 范围：0~59

S+2 范围：0~59

（2）32 位指令

将[S，S+1，S+2]的时间(时刻)数据(时、分、秒)换算成秒后，将结果保存到（D+1，D）中。

S 范围：0~32767

S+1 范围：0~59

S+2 范围：0~59

3、程序举例

略。

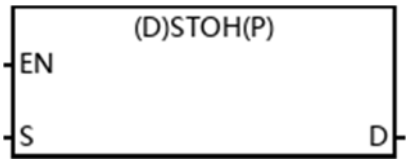
4、注意事项

S、S+1、S+2 的数据超出与指令对应的可设定范围时，报错 6706。

2.11.6 STOH/秒数据的（时、分、秒）转换指令

将秒单位的时间(时刻)数据转换成[时、分、秒]单位的数据的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存转换前的时间(时刻)数据(秒)的软元件编号	BIN16/32 位
D	保存转换后的时间(时刻)数据(时、分、秒)的软元件起始编号	BIN16 位

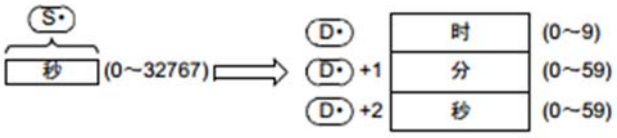
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

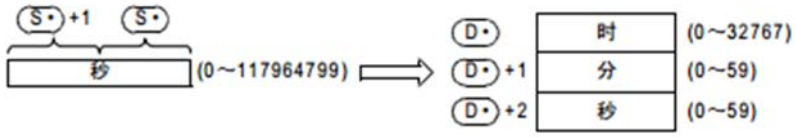
（1）16 位指令

将 S 的秒数据换算成时、分、秒，其结果保存到[D，D+1，D+2](时、分、秒)中。



（2）32 位指令

将（S+1，S）的秒数据换算成时、分、秒，其结果保存到[D，D+1，D+2](时、分、秒)中。



3、程序举例

略。

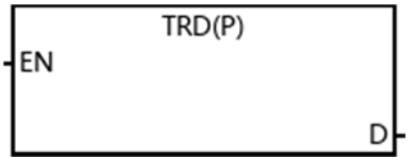
4、注意事项

要转换的数据超出与指令对应的可设定范围时，报错 6706。

2.11.7 TRD/读出时钟数据指令

将可编程控制器内置实时时钟的时钟数据(SD8013～SD8019)按照下面的格式读出 D～D+6 中。

1、指令格式



操作数说明

操作数	内容	类型
D	指定保存读出时间数据的起始软元件编号。(占用 7 点)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D												*	*	*	*			*					

2、功能和动作说明

16 位指令，将可编程控制器内置实时时钟的时钟数据(SD8013～SD8019)按照下面的格式读出到 D ～ D+6 中。

	软元件	项目	时钟数据		软元件	项目
特殊数据寄存器	SD8018	年（公历）	0~99（公历后 2 位数）	→	D0	年（公历）
	SD8017	月	1~12	→	D1	月
	SD8016	日	1~31	→	D2	日
	SD8015	时	0~23	→	D3	时
	SD8014	分	0~59	→	D4	分
	SD8013	秒	0~59	→	D5	秒
	SD8019	星期	0（日）~6（六）	→	D6	星期

3、程序举例

略。

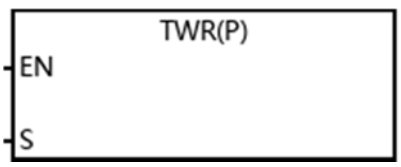
4、注意事项

D 占用 7 点软元件，请注意不要与机器其他控制中使用的软元件重复。

2.11.8 TWR/写入时钟数据指令

向可编程控制器内置实时时钟写入时钟数据的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	指定写入时间数据的源地址的起始软元件编号。（占用 7 点）	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					

2、功能和动作说明

将设定的时钟数据 S ～ S+6 写入可编程控制器内置实时时钟的时钟数据(SD8013～SD8019)中。

	软元件	项目	时钟数据		软元件	项目	
设定时间用的数据	S	年（公历）	0~99（公历后 2 位数）	→	SD8018	年（公历）	特殊数据寄存器
	S+1	月	1~12	→	SD8017	月	
	S+2	日	1~31	→	SD8016	日	
	S+3	时	0~23	→	SD8015	时	
	S+4	分	0~59	→	SD8014	分	
	S+5	秒	0~59	→	SD8013	秒	
	S+6	星期	0（日）~6（六）	→	SD8019	星期	

3、程序举例

略。

4、注意事项

（1）执行 TWR(FNC 167)指令后，实时时钟的时钟数据即刻被更改。因此，请先将快几分钟的时钟数据传送到 S～S+6 中，等到变成正确的时间时才执行指令。

（2）使用这个指令设定时钟数据(时间校准)时，不需要控制特殊辅助继电器 SM8015(时间停止以及时间校准)。

（3）设定了不可能显示的日期时间数值时，不执行时钟数据的变更。此时，请设定正确的时钟数据后再次写入。

（4）星期(SD8019)与写入数值无关,是根据日期内容自动补正。

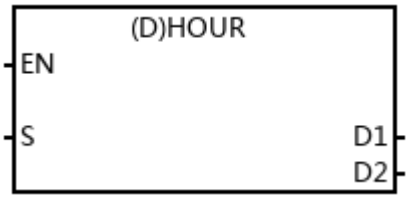
（5）占用 S 开始的连续 7 点软元件，请注意不要与机器其他控制中使用的软元件重复。

（6）设定时间时，设定快 1 分钟的时间，等到达正确时间时使 TWR 指令前面的触发开关置位，以使适中数据更新。

2.11.9 HOUR/计时表指令

以 1 个小时为单位，对输入触点持续 ON 的时间进行累加检测的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	使 D2 为 ON 的时间(以 1 个小时为单位设定)	BIN16/32 位
D1	以 1 个小时为单位的当前值(指定停电保持用数据寄存器)	BIN16/32 位
D2	报警输出的起始编号	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D1														*	*			*					
D2		*	*			*												*					

2、功能和动作说明

（1）16 位指令

当指令输入的累计 ON 时间超出了 S 时，D2 变为 ON。D1+1 中不满 1 个小时的当前值，以 1 秒单位被保存。

S：D2 变 ON 为止的时间以 1 个小时为单位指定。
D1：以 1 个小时为单位的当前值
D1+1 ：不满 1 个小时的当前值(1 秒单位)
D2：报警输出目标地址编号，当前值 D1 在 S 的指定时间以上时置 ON。

（2）32 位指令

（S+1，S）：D2 变为 ON 为止的时间指定，用 S1+1(高位)，S1(低位)指定。
（D1+1，D1）：以 1 个小时为单位的当前值保存在 D1+1(高位)，D1(低位)中。
D1+2 ：不满 1 个小时的当前值(1 秒单位)
D2：报警输出的指定，当前值 D1、D1+1 在 S 的指定时间以上时置 ON。

3、程序举例

略。

4、注意事项

（1）由于即使断开可编程控制器的电源后，也可以继续使用当前值数据，所以请在 D1 中指定停电保持用的数据寄存器。使用一般的数据寄存器时，由于可编程控制器的电源 OFF 和 STOP→RUN 的操作，当前值会被清除。

（2）报警输出 D2 为 ON 以后，测量仍能继续。再次开始进行测量的情况下，请清除 D1～D1+1(32 位指令为 D1～D1+2)的当前值。清除当前值时后，报警输出将变为 OFF。

（3）当前值 D1 达到最大值时停止测量。要继续测量时，请清除 D1～D1+1(32 位指令为 D1～D1+2)的当前值。

（4）D1 占用 2 个(16 位运算)或者 3 个(32 位运算)软元件。请注意不要与机器其他控制中使用的软元件重复。

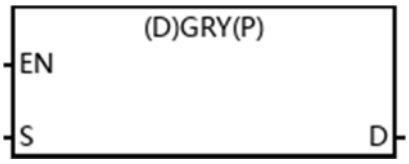
2.12 格雷码指令

主要用于格雷码数据的转换操作。

2.12.1 GRY/格雷码的转换指令

将 BIN 值转换成格雷码后进行传送的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	转换源数据，或是保存转换源数据的字软元件	BIN16/32 位
D	保存转换后数据的字软元件	BIN16/32 位

操作数可用元件类型

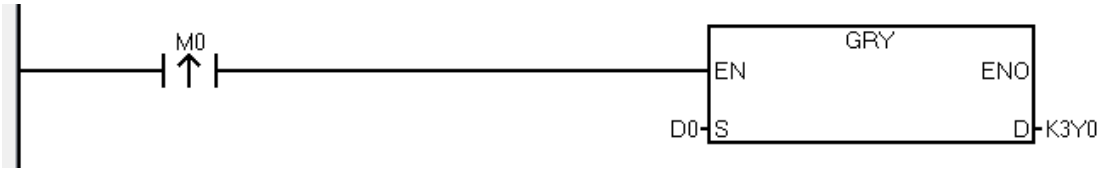
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

源数据(BIN)→目标数据(GRY)的转换传送指令。格雷码的值只需要在原来的二进制的基础上右移一位再异或原来的二进制值即可得到。

3、程序举例

S 为 D0 = 3456, D 为 K3Y0 时



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	3456

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
Y0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	2880

4、注意事项

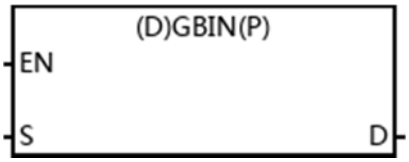
（1）数据的转换速度取决于可编程控制器的扫描时间。

（2）16 位指令时，S 范围为 0~32767；32 位指令时，S 范围为 0~2147483647。

2.12.2 GBIN/格雷码的逆转换指令

将格雷码转换成 BIN 值后进行传送的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	转换源数据，或是保存转换源数据的字软元件	BIN16/32 位
D	保存转换后数据的字软元件	BIN16/32 位

操作数可用元件类型

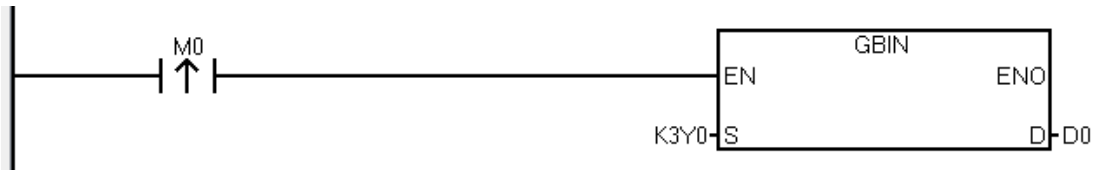
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

源数据(GRY)→目标数据(BIN)的转换传送指令。二进制格雷码转换成自然二进制码,其法则是保留格雷码的最高位作为自然二进制码的最高位，而次高位自然二进制码为高位自然二进制码与次高位格雷码相异或，而自然二进制码的其余各位与次高位自然二进制码的求法相类似。

3、程序举例

S 为 K3Y0, D 为 D0 时，指令执行后，D0 结果如下。



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	1	0	1	0	1	1	1	0	1	1

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1234

4、注意事项

16 位指令时，S 范围为 0~32767；32 位指令时，S 范围为 0~2147483647。

2.13 数据块指令

对批量数据同时进行加法、减法、比较运算操作的指令。

2.13.1 BK+/数据块的加法运算指令

数据块的 BIN 加法运算的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	保存执行加法运算的数据的软元件起始编号	BIN16/32 位

S2	执行加法运算的常数，或是保存执行加法运算的数据的软元件起始编号	BIN16/32 位
D	保存运算结果的软元件起始编号	BIN16/32 位
n	数据的个数	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*	*	*			
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

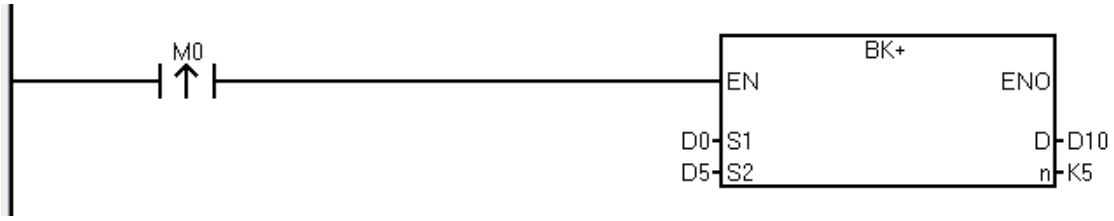
将 S1 开始 n 点 16 位数据和 S2 开始的 n 点 16 位数据(BIN)进行加法运算后，将运算结果保存到 D 开始的 n 点中。

（2）32 位指令

将（S1+1，S1）开始 2n 点 32 位数据和（S2+1，S2）开始的 2n 点 32 位数据(BIN)进行加法运算后，将运算结果保存到（D+1，D）开始的 2 n 点中。

3、程序举例

以 16 位指令运算为例，指令执行后 D10~D14 的结果如下。



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1234
D1	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	5678
D2	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	-2000
D3	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	-1234
D4	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	4000
D5	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	4000
D6	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1234
D7	1	1	1	1	1	0	1	1	0	0	1	0	1	1	1	0	-1234
D8	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	5000
D9	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	4321
D10	0	0	0	1	0	1	0	0	0	1	1	1	0	0	1	0	5234
D11	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	6912
D12	1	1	1	1	0	0	1	1	0	1	0	1	1	1	1	0	-3234
D13	0	0	0	0	1	1	1	0	1	0	1	1	0	1	1	0	3766
D14	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	8321

4、注意事项

（1）运算结果中产生数据上溢出、下溢出时，进位标志不置位。

（2）出现以下情况会报错 6706。

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

2.13.2 BK-/数据块的减法运算指令

数据块的 BIN 减法运算的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	保存执行减法运算的数据的软元件起始编号	BIN16/32 位
S2	执行减法运算的常数，或是保存执行减法运算的数据的软元件起始编号	BIN16/32 位
D	保存运算结果的软元件起始编号	BIN16/32 位

n	数据的个数	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*	*	*			
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

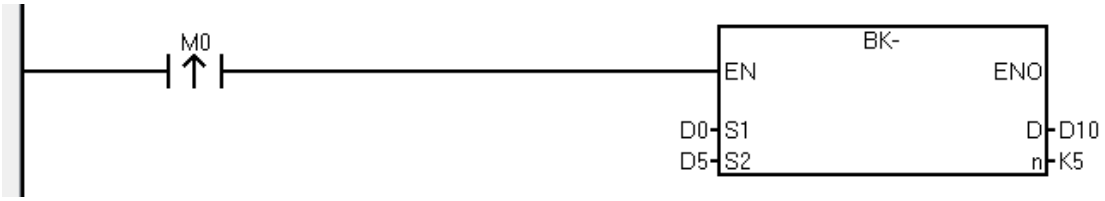
将 S1 开始 n 点 16 位数据和 S2 开始的 n 点 16 位数据(BIN)进行减法运算后，将运算结果保存到 D 开始的 n 点中。

（2）32 位指令

将（S1+1，S1）开始 2n 点 32 位数据和（S2+1，S2）开始的 2n 点 32 位数据(BIN)进行减法运算后，将运算结果保存到（D+1，D）开始的 2 n 点中。

3、程序举例

以 16 位指令运算为例，指令执行后，D10~D14 结果如下。



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	1	8765
D1	0	0	1	0	0	0	1	0	1	0	1	1	1	0	0	0	8888
D2	0	0	1	0	0	1	0	0	0	1	1	0	1	1	0	1	9325
D3	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	5000
D4	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	4352
D5	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1234
D6	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	5678
D7	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0	0	9876
D8	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	4321
D9	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	4000
D10	0	0	0	1	1	1	0	1	0	1	1	0	1	0	1	1	7531
D11	0	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0	3210
D12	1	1	1	1	1	1	0	1	1	1	0	1	1	0	0	1	-551
D13	0	0	0	0	0	0	1	0	1	0	1	0	0	1	1	1	679
D14	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	352

4、注意事项

（1）运算结果中产生数据上溢出、下溢出时，进位标志不置位。

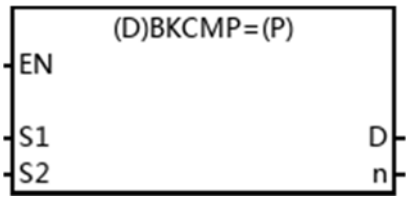
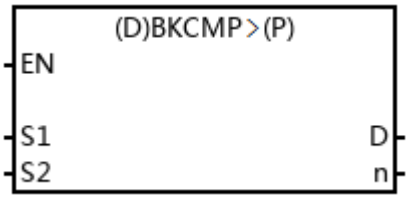
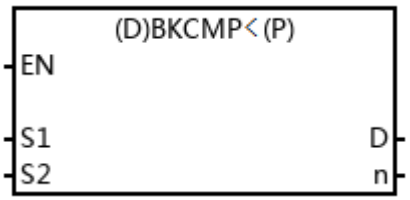
（2）出现以下情况会报错 6706。

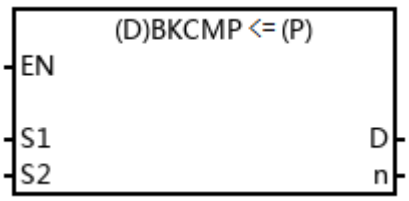
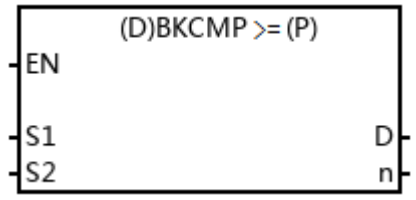
•S1、S2、D 开始的 n 点(32 位运算时为 2n 点)软元件超出相应的软元件范围时。
•S1 开始的 n 点软元件和 D 开始的 n 点软元件重复时(32 位运算时为 2n 点)。
•S2 开始的 n 点软元件和 D 开始的 n 点软元件重复时(32 位运算时为 2n 点)。

2.13.3 BKCMP=、>、<、<>、<=、>=/数据块的比较指令

这些指令按照各个指令的比较条件来比较数据块。

1、指令格式

指令位数	指令分类	指令格式
16 位	BKCMP=	
	BKCMP=P	
32 位	DBKCMP=	
	DBKCMP=P	
16 位	BKCMP>	
	BKCMP>P	
32 位	DBKCMP>	
	DBKCMP>P	
16 位	BKCMP<	
	BKCMP<P	
32 位	DBKCMP<	
	DBKCMP<P	
16 位	BKCMP<>	
	BKCMP<>P	
32 位	DBKCMP<>	
	DBKCMP<>P	

<table>
<tr><td rowspan="2">16 位</td><td>BKCMP<=</td><td rowspan="2">
</td></tr>
<tr><td>BKCMP<=P</td></tr>
<tr><td rowspan="2">32 位</td><td>DBKCMP<=</td></tr>
<tr><td>DBKCMP<=P</td></tr>
<tr><td rowspan="2">16 位</td><td>BKCMP>=</td><td rowspan="2">
</td></tr>
<tr><td>BKCMP>=P</td></tr>
<tr><td rowspan="2">32 位</td><td>DBKCMP>=</td></tr>
<tr><td>DBKCMP>=P</td></tr>
</table>

操作数说明

操作数	内容	类型
S1	比较值或是保存比较值的软元件编号	BIN16/32 位
S2	保存比较源数据的软元件起始编号	BIN16/32 位
D	保存比较结果的软元件起始编号	位
n	要比较的数据数	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*	*	*			
S2												*	*	*	*			*					
D		*	*			*												*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

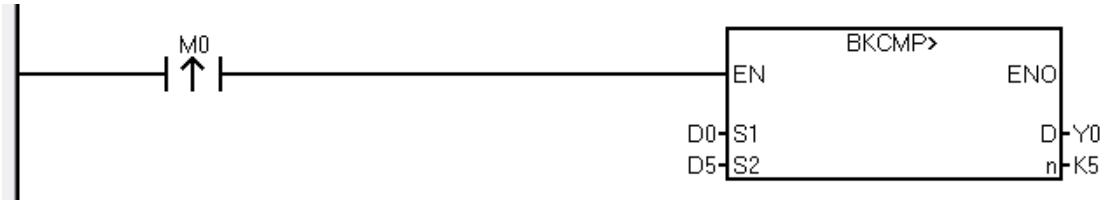
将 S1 开始 n 点 16 位数据和 S2 开始的 n 点 16 位数据(BIN)进行比较后，将比较结果保存到 D 开始的 n 点中。

（2）32 位指令

将（S1+1，S1）开始 2n 点 32 位数据和（S2+1，S2）开始的 2n 点 32 位数据(BIN)进行比较后，将比较结果保存到（D+1，D）开始的 2 n 点中。

3、程序举例

以 16 位指令 BKCMP>指令为例，当指令执行后 Y0~Y4 的结果如下。



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1234
D1	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	5678
D2	0	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	5000
D3	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	7777
D4	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	1	4321
D5	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	1	5321
D6	0	0	0	0	1	1	0	1	0	1	0	0	0	1	1	1	3399
D7	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	5678
D8	0	0	0	1	1	0	0	1	1	0	0	0	1	1	1	1	6543
D9	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1200

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

4、注意事项

（1）指令比较结果如下

16 位指令	32 位指令	比较结果 ON 的条件	比较结果 OFF 的条件
BKCMP=	DBKCMP=	S1=S2	S1≠S2
BKCMP>	DBKCMP>	S1>S2	S1≤S2
BKCMP<	DBKCMP<	S1<S2	S1≥S2
BKCMP<>	DBKCMP<>	S1≠S2	S1=S2
BKCMP<=	DBKCMP<=	S1≤S2	S1>S2

BKCMP>=	DBKCMP>=	S1≥S2	S1<S2

（2）SM8094 块比较信号的动作：数据块指令的比较结果都为 ON(1)时变为 ON。

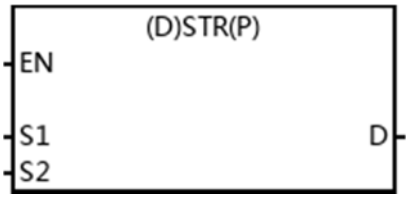
2.14 字符串控制指令

针对字符串进行控制的相关操作指令，包含对字符串的转换、替换、检索、取出等。

2.14.1 STR/BIN→字符串的转换指令

将 BIN 数据转换成字符串(ASCII 码)的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	保存要转换数值的位数的软元件起始编号	BIN16/32 位
S2	保存要转换的 BIN 数据的软元件编号	BIN16/32 位
D	保存已转换的字符串的软元件起始编号	字符串

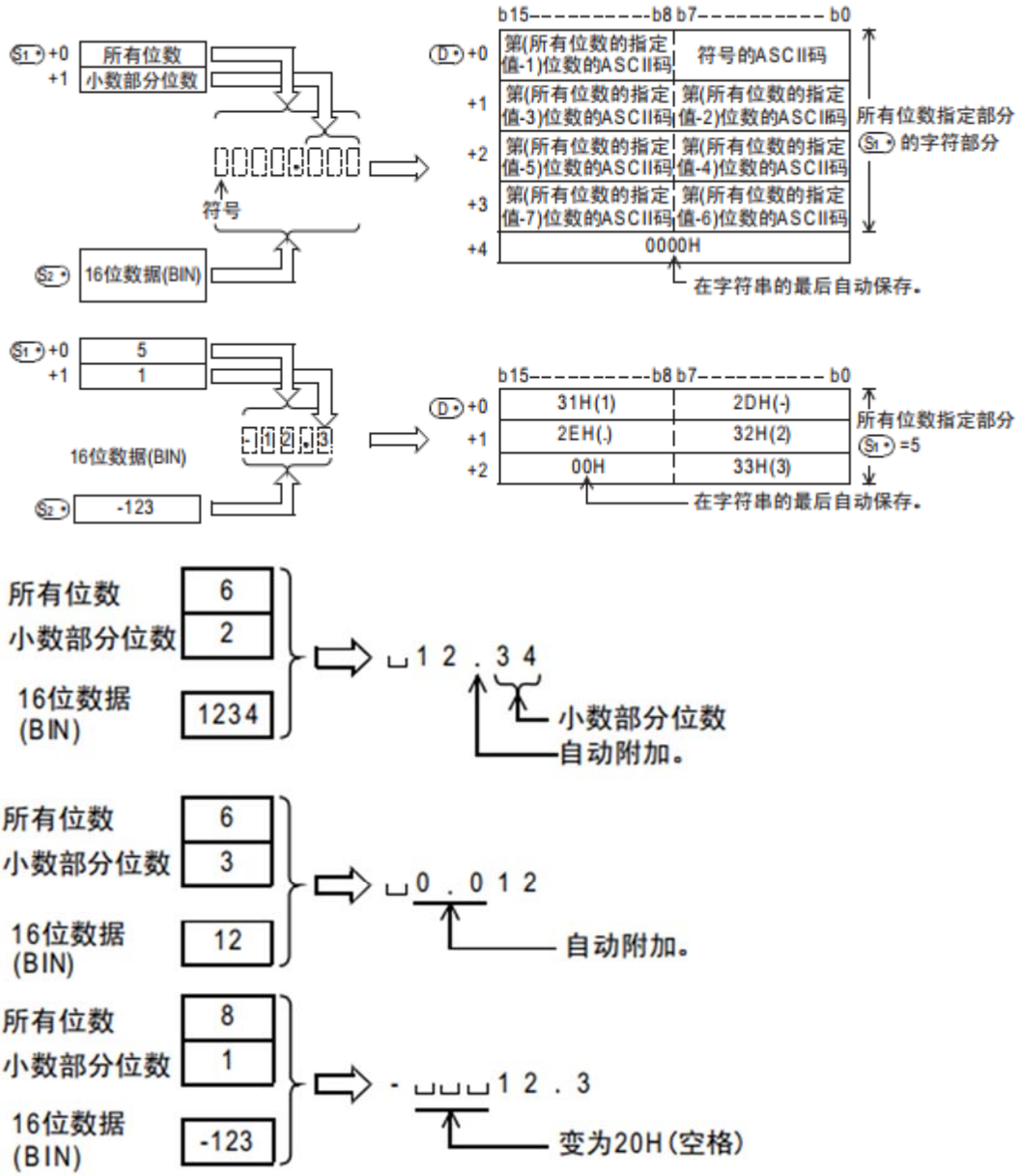
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D												*	*	*	*			*					

2、功能和动作说明

（1）16 位指令

将 S2 的 16 位数据(BIN)，在所有位数(S1)、小数点部分位数(S1 +1) 指定的位置中加上小数点后，转换成字符串，保存到 D 开始的软元件中。



转换后的字符串数据会如下所示地保存到 D 开始的软元件编号中：

• 在符号中，16 位数据(BIN)S2 为正时保存“空格(20H)”，为负时保存“-(2DH)”。

• 在小数部分位数 S1+1 中设定为“0”以外的数字时，会自动在小数部分位数+1 位数的位置上加上小数点“. (2EH)”。小数部分位数 +1 为“0”时，不附加小数点。

• 与 S2 的 16 位数据(BIN)的位数相比，S1+1 的小数部分位数较多时，会自动向右对齐，在左边附加“0(30H)”后进行转换。

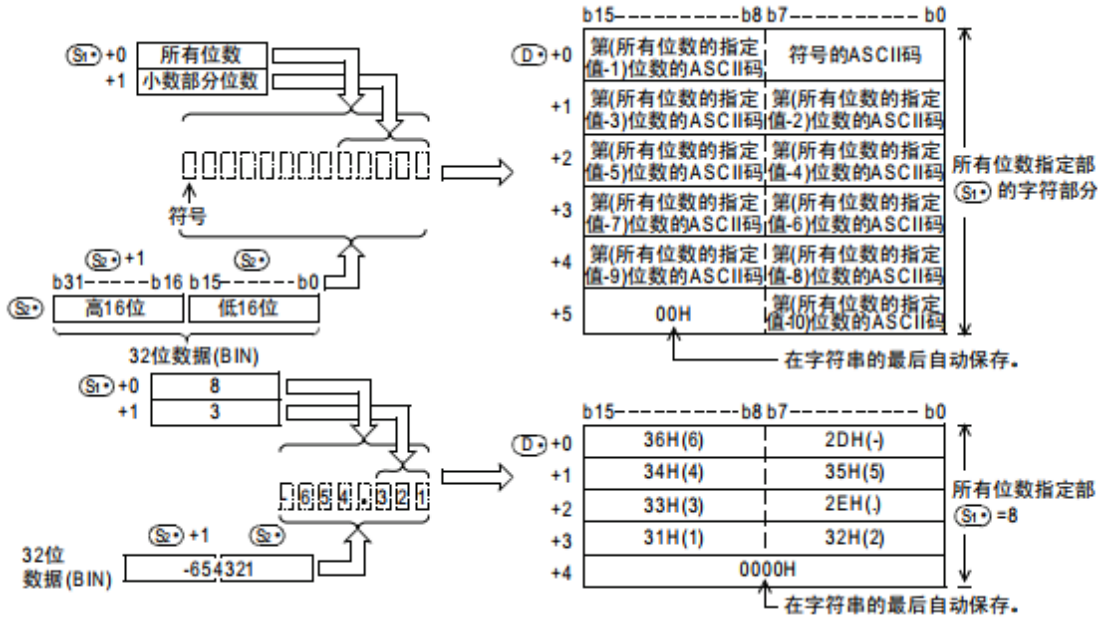
• 如果除去小数点和符号以外，所有位数 S1 的位数多于 S2 的 16 位数据(BIN)的位数时，在符号和数值之间保存“空格(20H)”。此外，S2 的 16 位数据(BIN)的位数较多时，会错误。

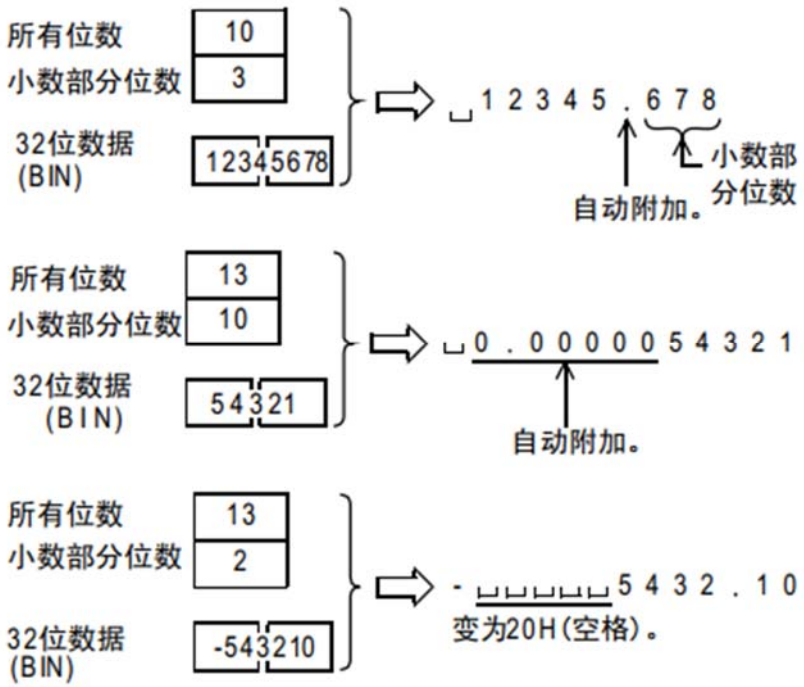
• 在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。总位数为偶数

位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8 位)中保存“00H”。

（2）32 位指令

将（S2+1，S2）的 32 位数据(BIN)，在所有位数(S1)、小数点部分位数(S1 +1) 指定的位置中加上小数点后，转换成字符串，保存到 D 开始的软元件中。





转换后的字符串数据会如下所示地保存到 D 开始的软元件编号中：

- 在符号中，32 位数据(BIN)S2 为正时保存“空格(20H)”，为负时保存“-(2DH)”。
- 在小数部分位数 S1+1 中设定为“0”以外的数字时，会自动在小数部分位数+1 位数的位

置上加上小数点“. (2EH)”。小数部分位数 +1 为“0”时，不附加小数点。
• 与 S2 的 32 位数据(BIN)的位数相比，S1+1 的小数部分位数较多时，会自动向右对齐，在左边附加“0(30H)”后进行转换。
• 如果除去小数点和符号以外，所有位数 S1 的位数多于 S2 的 32 位数据(BIN)的位数时，在符号和数值之间保存“空格(20H)”。此外，S2 的 32 位数据(BIN)的位数较多时，会错误。
• 在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。总位数为偶数位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8 位)中保存“00H”。

3、程序举例

略。

4、注意事项

以下情况会报错 6706：

• 所有位数 S1 为如下所示的范围以外时。

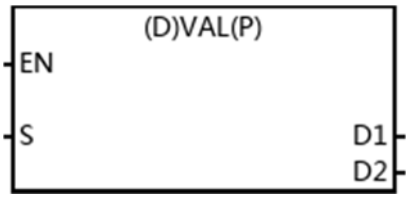
16 位运算：2~8；32 位运算 2~13。
• 所有位数 S1+1 为如下所示的范围以外时。

16 位运算：0~5；32 位运算 0~10。
• 所有位数 S1 和小数部分位数 S1+1 的关系非如下所示的范围时。
(所有位数-3)≧小数部分位数
• 所有位数 S1+符号、小数点的位数比 S2 的 BIN 数据位数少时。
• 保存字符串的 D 以后的软元件超出了相应的软元件范围时。

2.14.2 VAL/字符串→BIN 的转换指令

将字符串(ASCII 码)转换成 BIN 数据的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存要转换成BIN数据的字符串的软元件起始编号	字符串
D1	保存已经转换的 BIN 数据位数的软元件起始编号	BIN16 位

D2	保存已经转换的 BIN 数据的软元件起始编号	BIN16/32 位

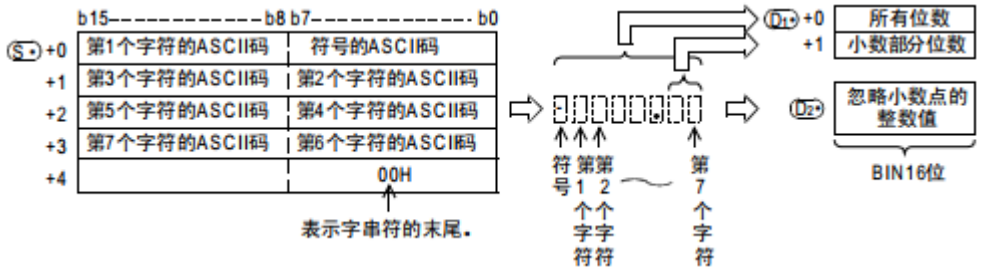
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D1												*	*	*	*			*					
D2									*	*	*	*	*	*	*			*					

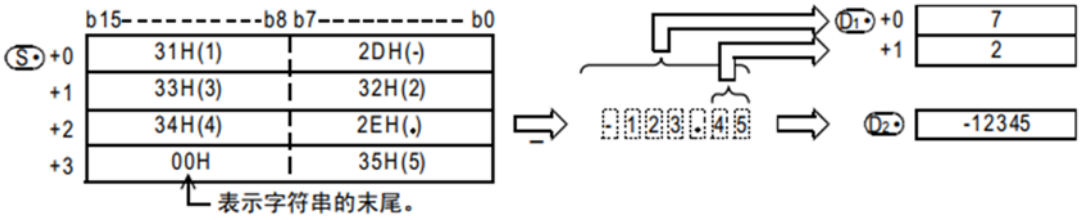
2、功能和动作说明

（1）16 位指令

将 S 开始的软元件中保存的字符串转换成 16 位数据（BIN），然后将所有位数保存到 D1 中、将小数部分位数保存到 D1+1 中、将 BIN 数据保存到 D2 中。
从字符串转换成 BIN 时，以字节为单位将 S 开始到保存“00H” 的软元件编号为止的数据，作为字符串进行处理。



例如，在 S 开始的软元件中指定了“-123.45”的字符串时，D1、D2 中如下所示保存。



要转换的字符串数据：

• 字符串的字符数，忽略小数点时的数值范围

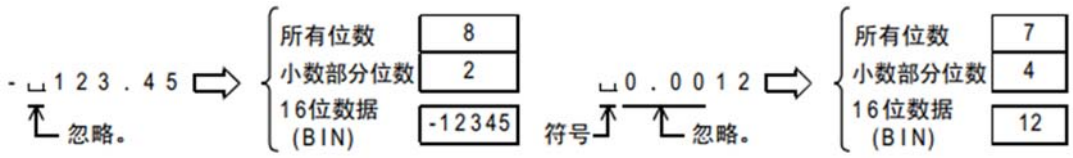
	内容
所有字符位数	2~8 个字符

小数部分的字符位数	0~5 个字符，但是位数在（所有位数-3）以下
忽略小数点的数值范围	-32768~32767 （例“123.45”-> “12345”）

• 在要转换的字符中使用的字符种类

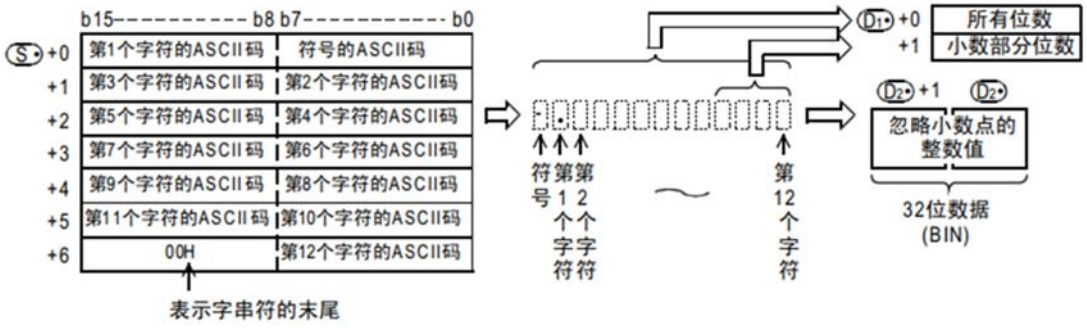
		字符的种类
符号	正的数值	“空格（20H）”
	负的数值	“-（2DH）”
小数点		“.(2EH)”
3 数字		“0(30H)”～“9（39H）”

• D1 中保存所有位数。所有位数，就是所有的字符数(包括数字、符号、小数点)。
• D1 +1 中保存小数部分的位数。小数部分的位数为小数点“. (2EH)”以后的字符数。
• 在 D2 中，无视小数点，将字符串转换成 16 位的数据(BIN)。但是，在 S 开始的字符串中，符号和最初的“0”以外的数字之间的“空格(20H)”或是“0(30H)”被忽略，而转换成 16 位数据(BIN)。

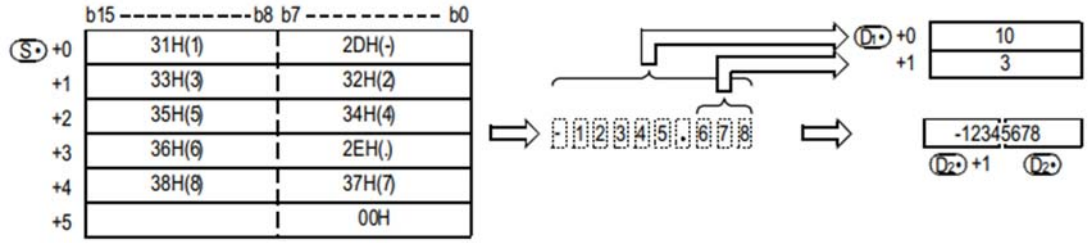


（2）32 位指令

将 S 开始的软元件中保存的字符串转换成 32 位数据 (BIN)，然后将所有位数保存到 D1 中、将小数部分位数保存到 D1+1 中、将 BIN 数据保存到(D2+1,D2)中。
从字符串转换成 BIN 时，以字节为单位将 S 开始到保存“00H” 的软元件编号为止的数据，作为字符串进行处理。



例如，在 S 开始的软元件中指定了“-12345.678”的字符串时，D1、D2 中如下所示保存。



要转换的字符串数据：

• 字符串的字符数，忽略小数点时的数值范围

	内容
所有字符位数	2~13 个字符
小数部分的字符位数	0~10 个字符，但是位数在（所有位数-3）以下
忽略小数点的数值范围	-2147483648~2147483647 （例“12345.678”-> “12345678”）

• 在要转换的字符中使用的字符种类

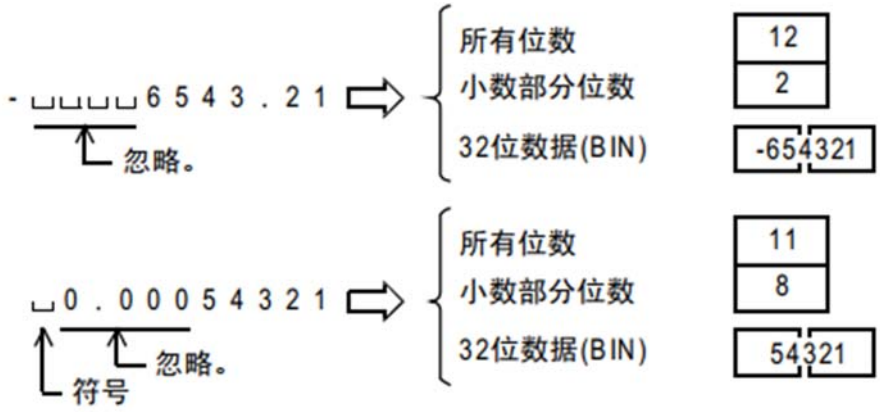
		字符的种类
符号	正的数值	“空格（20H）”
	负的数值	“-（2DH）”
小数点		“.(2EH)”
3 数字		“0(30H)”～“9（39H）”

• D1 中保存所有位数。所有位数，就是所有的字符数(包括数字、符号、小数点)。

• D1 +1 中保存小数部分的位数。小数部分的位数为小数点“. (2EH)”以后的字符数。

• 在(D2+1,D2)中，无视小数点，将字符串转换成 16 位的数据(BIN)。但是，在 S 开始的字符串中，符号和最初的“0”以外的数字之间的“空格(20H)”或是“0(30H)”被忽略，而

转换成 32 位数据(BIN)。



3、程序举例

略。

4、注意事项

以下情况会报错 6706：

• 要转换的字符串（S 以后）的字符数为如下所示的范围以外时。

16 位运算：2~8；32 位运算 2~13。

• 要转换的字符串（S 以后）的小数部分的字符为如下所示的范围以外时。

16 位运算：0~5；32 位运算 0~10。

• 要转换的字符串（S 以后）的所有字符数，和小数部分的字符数之间的关系非如下所示的范围时。

(所有字符数-3)≧小数部分字符数。

• 在符号中设定了“空格(20H)”、“-(2DH)”以外的 ASCII 码时。

• 各数字的位数中设定了“0(30H)”～“9(39H)”，以及小数点“. (2EH)”以外的 ASCII 码时。

• 要转换的字符串(S 以后)中设定了多个小数点“. (2EH)”时。

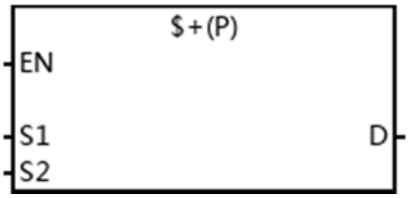
• 转换后的 BIN 数据超出了如下所示的范围时。

16 位运算：-32768~32767；32 位运算：-2147483648~2147483647。

2.14.3 $+/字符串的结合指令

连接字符串与字符串的指令。

1、指令格式



操作数说明

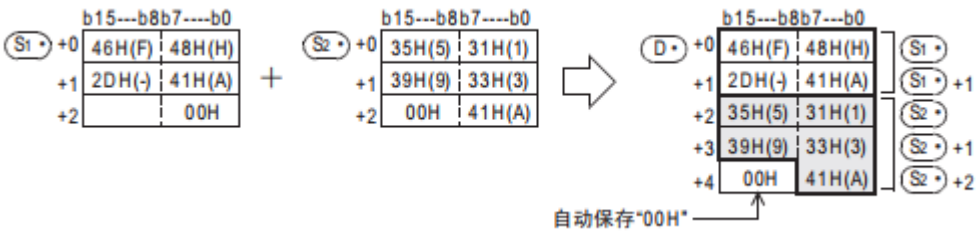
操作数	内容	类型
S1	字符串 1	字符串
S2	字符串 2	字符串
D	连接后的字符串	字符串

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*			*		
S2								*	*	*	*	*	*	*	*			*			*		
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

在 S1 开始的字符串数据后面连接 S2 开始的字符串数据，然后保存到 D 以后的软元件中。S1 和 S2 的字符串，是指以字节为单位从被指定的软元件开始到检测到第一个[00H]的位置为止的数据。



字符串的结合，就是指忽略 S1 中表示指定字符串末尾的“00H”，接着 S1 的最后字符连接 S2 中指定的字符串。此外，执行字符串的结合后会自动将“00H”附加在最后。

- 连接后的字符数为奇数时，在保存最后字符的软元件的高字节中保存“0000H”。
- 连接后的字符数为偶数时，在保存最后字符的软元件的下一个软元件中保存“0000H”。

3、程序举例

略。

4、注意事项

（1）直接指定字符串时，可以指定(输入)的字符数最多为 32 个字符。 但是 S1 和 S2 中被指定了字软元件时，字符数没有限制。

（2）S1、S2 中任何一个的值从 00H 开始时，在 D 中保存“0000H”。

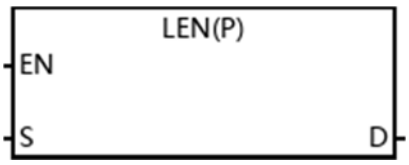
（3）以下情况报错 6706：

• D 中指定的软元件编号开始的软元件数，比保存所有已经结合的字符串所需的软元件数量更少时。(在所有的字符串和最终字符后面不能保存[00H])
• 当 S1 和 S2 中指定的保存字符串的软元件和 D 中指定的保存字符串的软元件编号重复时。
• 当 S1 和 S2 指定的软元件开始的相应软元件范围中[00H]未被设定时。

2.14.4 LEN/检测出字符串的长度指令

检测出指定字符串的字符数(字节数)的指令。

1、指令格式



操作数说明

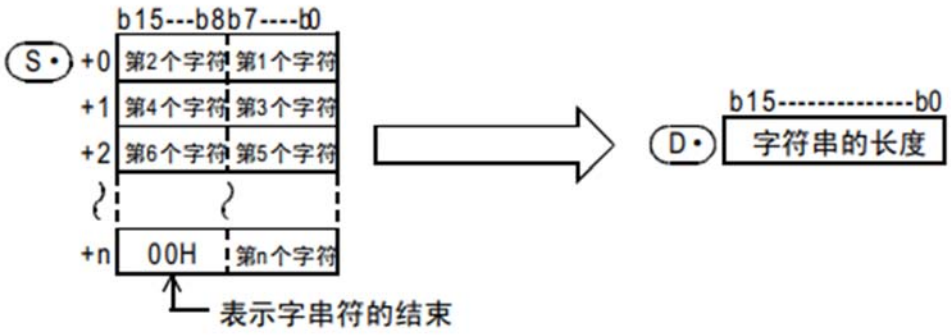
操作数	内容	类型
S	保存要检测出字符数的字符串的软元件起始编号	字符串
D	保存已检测出的字符串的长度(字节数)的软元件编号	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

检测出以 S 开头的字符串的长度，将字符串的长度保存到 D 中。S 以字节为单位，将从开始到第 1 个保存有[00H]的软元件编号为止的数据，作为字符串进行处理。



3、程序举例

略。

4、注意事项

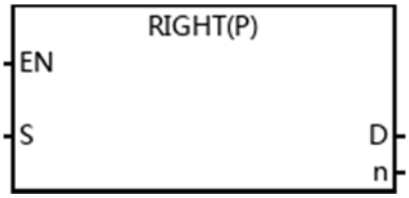
以下情况会报错 6706：

- 在 S 指定的软元件编号开始的相应软元件范围内没有设定[00H]时。
- 检测出字符数超过 32768 个时。

2.14.5 RIGHT/从字符串的右侧开始取出指令

从指定的字符串的右侧取出指定字符数的字符的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存字符串的软元件起始编号	字符串
D	保存被取出的字符串的软元件起始编号	字符串
n	要取出的字符数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n														*	*				*	*			

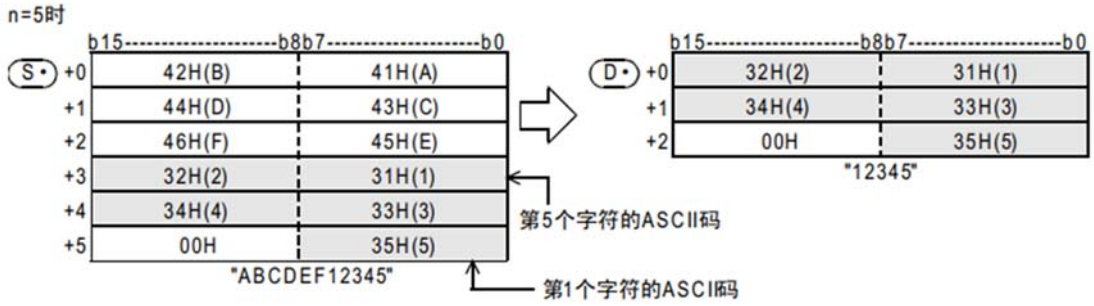
2、功能和动作说明

从 S 开始的软元件中保存的字符串数据的右侧(字符串的末尾)取出 n 个字符的数据，保存到 D 开始的软元件中。但是，n 中指定的字符数为“0”时，D 中保存 NULL 代码(0000H)。此外，取出字符串时，会在最后自动附加“00H”。

- 要取出的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数为偶数时， 在保存最后字符的软元件的下一个软元件中保“0000H”。

3、程序举例

将 S 为起始的字符串数据“ABCDEF12345”的右 5 位字符存到 D 中：



以 S 开头的字符串，就是指从被指定的软元件开始，以字节为单位，到检测出第 1 个[00H]的位置为止的数据。

4、注意事项

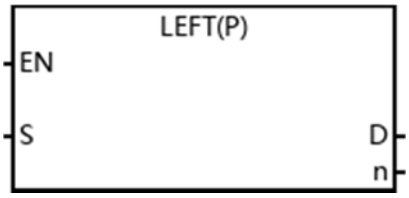
以下情况会报错 6706：

- S 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
- n 超出了 S 中指定的字符数时。
- D 中指定的软元件编号开始的软元件数，比保存已经取出的字符串(n 个字符)所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])
- n 为负值时。

2.14.6 LEFT/从字符串的左侧开始取出指令

从指定的字符串的左侧取出指定字符数的字符的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存字符串的软元件起始编号	字符串
D	保存被取出的字符串的软元件起始编号	字符串
n	要取出的字符数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n														*	*				*	*			

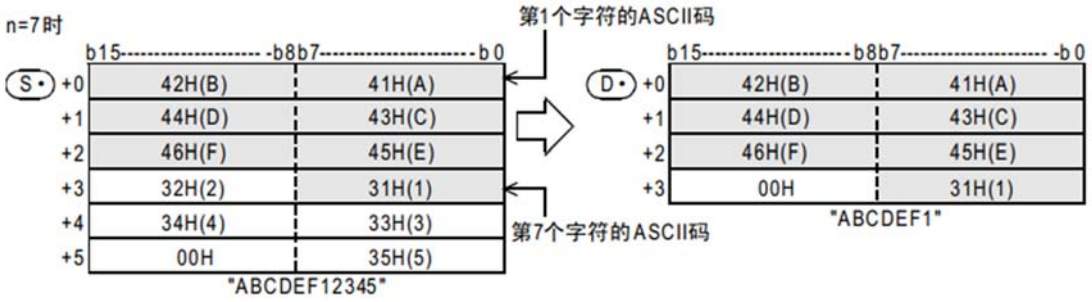
2、功能和动作说明

从 S 开始的软元件中保存的字符串数据的左侧(字符串的开头)取出 n 个字符的数据，保存到 D 开始的软元件中。但是，n 中指定的字符数为“0”时，D 中保存 NULL 代码(0000H)。此外，取出字符串时，会在最后自动附加“00H”。

- 要取出的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数为偶数时， 在保存最后字符的软元件的下一个软元件中保“0000H”。

3、程序举例

将 S 为起始的字符串数据“ABCDEF12345”的左 7 位字符存到 D 中：



以 S 开头的字符串，就是指从被指定的软元件开始，以字节为单位，到检测出第 1 个[00H]的位置为止的数据。

4、注意事项

以下情况会报错 6706：

- S 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
- n 超出了 S 中指定的字符数时。
- D 中指定的软元件编号开始的软元件数，比保存已经取出的字符串(n 个字符)所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])
- n 为负值时。

2.14.7 MIDR/从字符串中的任意取出指令

取出指定的字符串中任意位置上的字符串的指令。

1、指令格式



操作数说明

操作数	内容	类型

S1	保存字符串的软元件起始编号	字符串
D	保存被取出的字符串的软元件起始编号	字符串
S2	指定要取出的字符的起始位置以及字符数的软元件起始编号 S2：起始字符位置 S2+1：字符数	BIN16 位

操作数可用元件类型

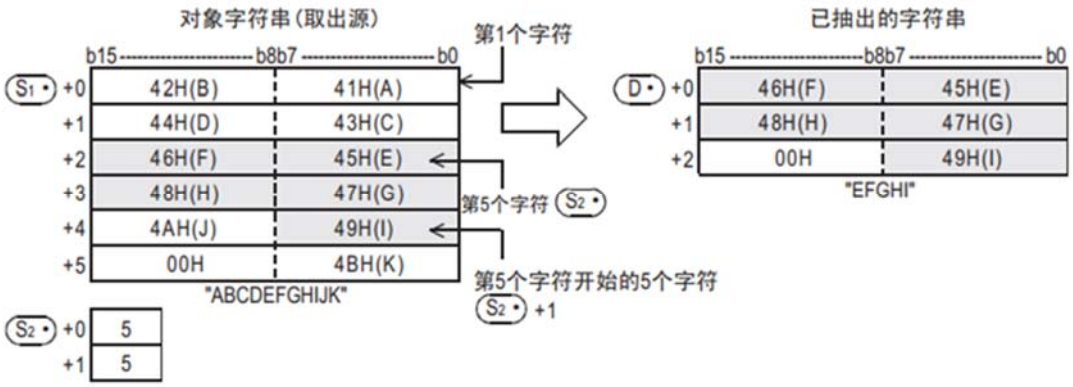
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
S2								*	*	*	*	*	*	*	*			*					

2、功能和动作说明

从 S1 开始的软元件中保存的字符串数据的左侧(字符串的开头)起第 S2 个字符开始，取出（S2+1）个字符的数据，保存到 D 开始的软元件中。

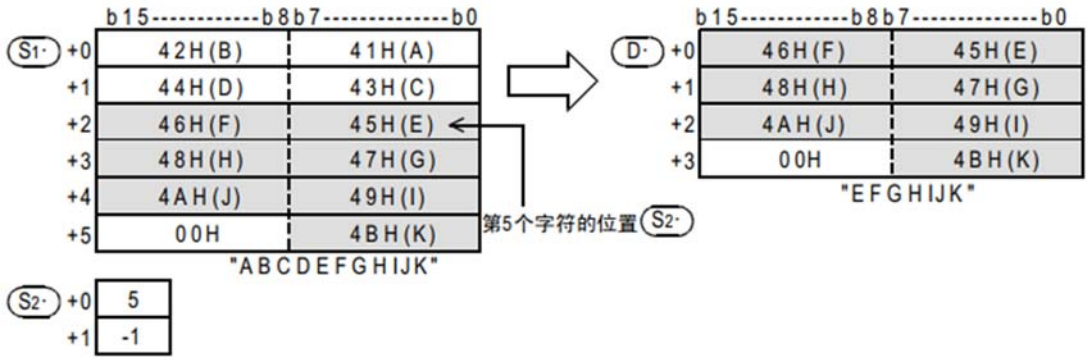
此外，取出字符串时，会在最后自动附加“00H”。

- 要取出的字符数（S2+1）为奇数时，在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数(S2+1)为偶数时， 在保存最后字符的软元件的下一个软元件中保存“0000H”。



- S1 中指定的字符串(数据)，就是指从被指定的软元件开始到检测到第 1 个[00H]为止的数据。
- S2+1 中指定的要取出的字符数为“0”时不执行处理。

• S2+1 中指定的要取出的字符数为“-1”时，到 S1 指定的字符串的最终字符数据为止的内容被保存在中。



3、程序举例

略。

4、注意事项

以下情况会报错 6706：

• S1 中指定的软元件开始的相应软元件范围内没有设定[00H]时。

• S2 的值超出了 S1 中指定的字符串的字符数时。

• D 开始的（S2+1）的字符数超出了 D 的软元件范围时。

• D 中指定的软元件编号开始的软元件数，比保存已取出的字符串 （S2+1）个字符所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])

• S2 为负值时。

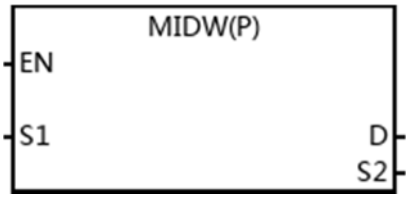
• S2+1 为-2 以下的值时。

• S2+1 超出了 S1 的字符数时。

2.14.8 MIDW/从字符串的任意替换指令

用指定的字符串中任意位置上的字符串去替换指定的字符串的指令。

1、指令格式



操作数说明

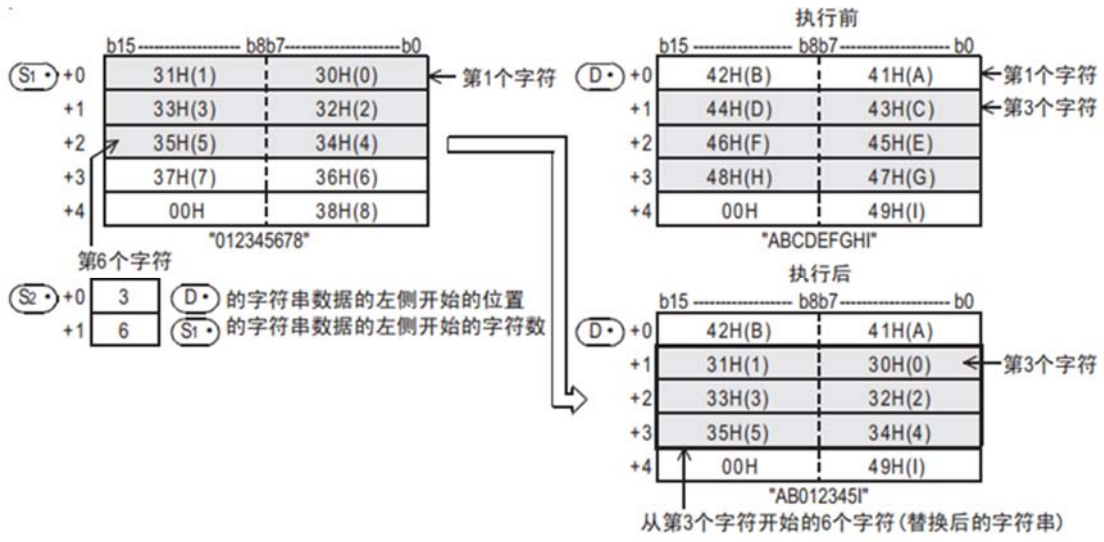
操作数	内容	类型
S1	保存字符串的软元件起始编号	字符串
D	保存替换后的字符串的软元件起始编号	字符串
S2	指定要替换的字符的起始位置以及字符数的软元件起始编号 S2：被替换的字符串的起始字符位置 S2+1：要替换的字符数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
S2								*	*	*	*	*	*	*	*			*					

2、功能和动作说明

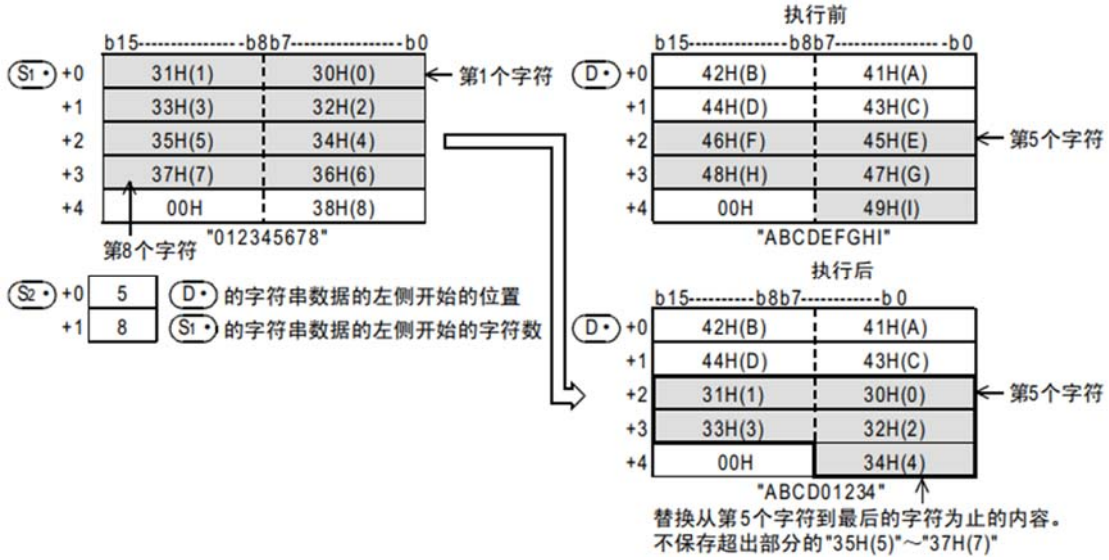
针对 S1 起始的软元件中保存的字符串数据，将其左侧(字符串的开头) 开始的字符数据，(字符数据数为 S2+1 中指定的字符数)保存到 D 起始软元件所保存的字符串中，保存位置从字符串数据左起数 S2 指定的位置开始。



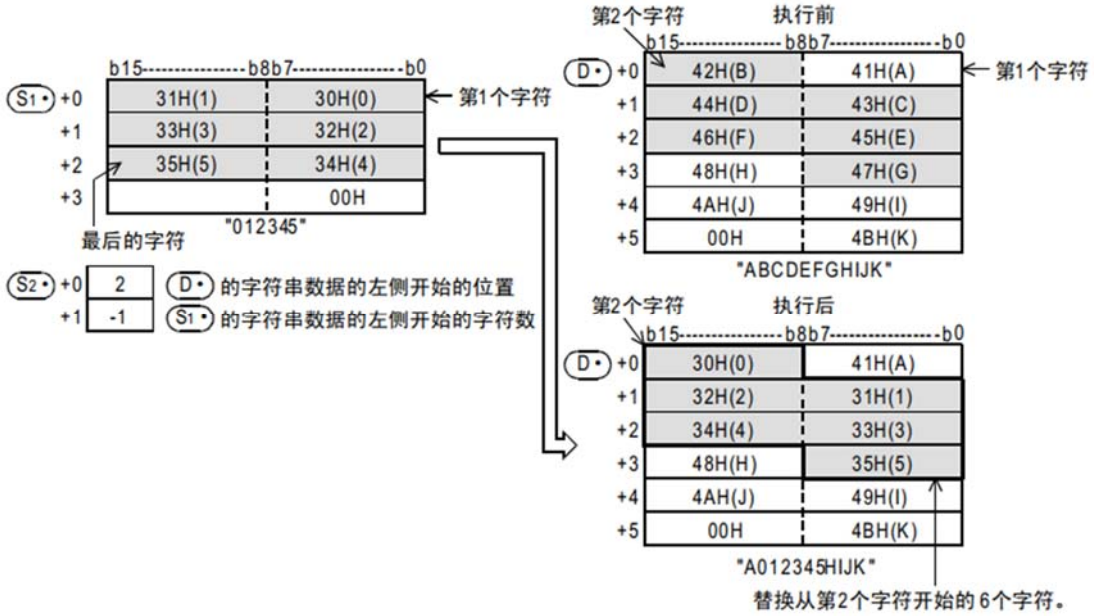
• S1 中 D 被指定的字符串(数据)，就是指从指定的软元件开始到检测开头到第 1 个[00H]为止的数据。

• S2+1 中指定的要替换的字符数为"0"时不执行处理。

• S2+1 中指定的要替换的字符数，一旦超出了 D 开始的字符串数据的最后字符时，则保存到最后字符为止的数据。



• S2+1 中指定的字符数为“-1”时，到 S1 指定的最终字符数据为止的内容都被保存到 D 指定的软元件以后中。



3、程序举例

略。

4、注意事项

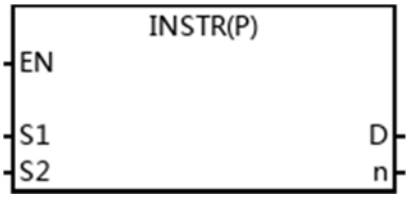
以下情况报错 6706：

- S1 和 D 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
- S2 的值超出了 D 的字符数时。
- S2 为负值时。
- S2+1 为-2 以下的值时。
- S2+1 超出了 S1 的字符数时。

2.14.9 INSTR/字符串的检索指令

从指定的字符串中检索指定字符串的指令。

1、指令格式



操作数说明

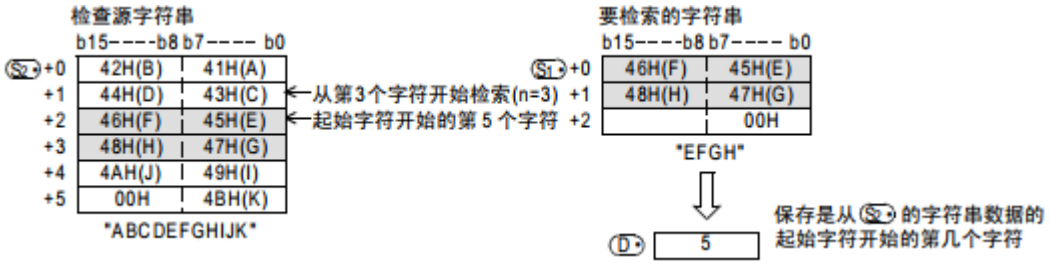
操作数	内容	类型
S1	保存要检索的字符串的软元件起始编号	字符串
S2	保存检索源字符串的软元件起始编号	字符串
D	保存检索结果的软元件起始编号	BIN16 位
n	开始检索的位置	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*				*	
S2												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）从 S2 起始的软元件中保存的检索源字符串的左起(起始字符)第 n 个字符开始，检索与保存在 S1 起始的软元件中的字符串相同的字符串，然后将检索结果的字符串位置信息保存到 D 中。检索的结果，就是检索源字符串左起(起始字符)的第几个字符的字符位置信息。



（2）不存在一致的字符串时，在 D 中保存“0”。

（3）开始检索的位置 n 为负数或是“0”不执行处理。

（4）在要检索的字符串 S1 中，可以直接指定字符串。

3、程序举例

略。

4、注意事项

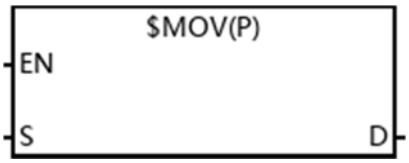
以下情况报错 6706：

- 开始检索的位置 n 超过了 S2 的字符数时。
- S1 开始的相应软元件的软元件范围内没有 00H(NULL)时。
- S2 开始的相应软元件的软元件范围内没有 00H(NULL)时。

2.14.10 $MOV/字符串的传送指令

传送字符串数据的指令。

1、指令格式



操作数说明

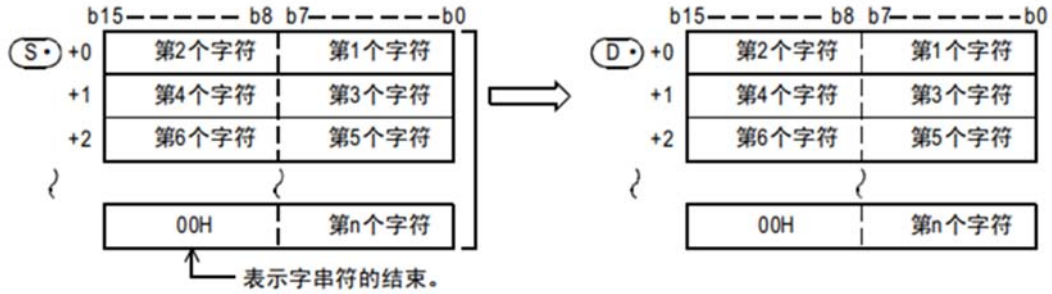
操作数	内容	类型
S	源字符串(最大 32 个字符)，	字符串
D	目的字符串	字符串

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*				*	
D									*	*	*	*	*	*	*			*					

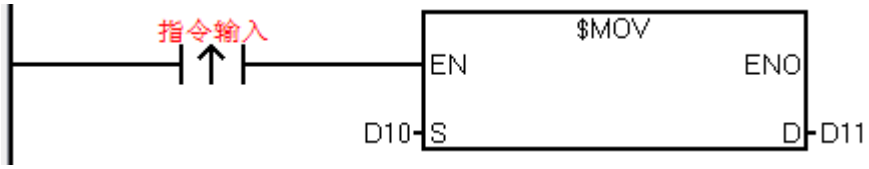
2、功能和动作说明

将 S 指定的软元件编号起始的软元件中保存的字符串数据，传送到 D 的软元件编号起始的软元件中。在传送字符串过程中，从 S 指定的软元件编号开始，一直到之后的软元件中，其高字节或是低字节中包含“00H”的软元件为止，都一次进行传送。



3、程序举例

将 D10～D13 中保存的字符串传送到 D11～D14 中



指令执行后，可看到传送后 D10 内的字符串没有变动。

显示格式： 16I 32I 32E ASC 10 16

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D10	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	12
D11	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	12
D12	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	34
D13	0	0	1	1	0	1	1	0	0	0	1	1	0	1	0	1	56

保存要传送的字符串数据的软元件范围（S～S+n），以及保存被传送的字符串数据的软元件范围（D~D+n/2）使用重复时，也能执行传送。

4、注意事项

（1）S+n 的低字节中保存了“00H”时，D+n 的高字节、低字节中都保存“00H”。

（2）以下情况报错 6706：

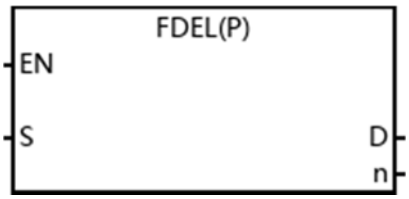
• S 中指定的软元件编号～指定软元件的最终编号之间不存在“00H”时。
• 在 D 中指定的软元件编号～指定软元件的最终编号之间的点数中，不能保存所有指定的字符串时。

2.15 数据处理 3 指令

2.15.1 FDEL/数据表的数据删除指令

删除数据表格中任意数据的指令。

1、指令格式



操作数说明

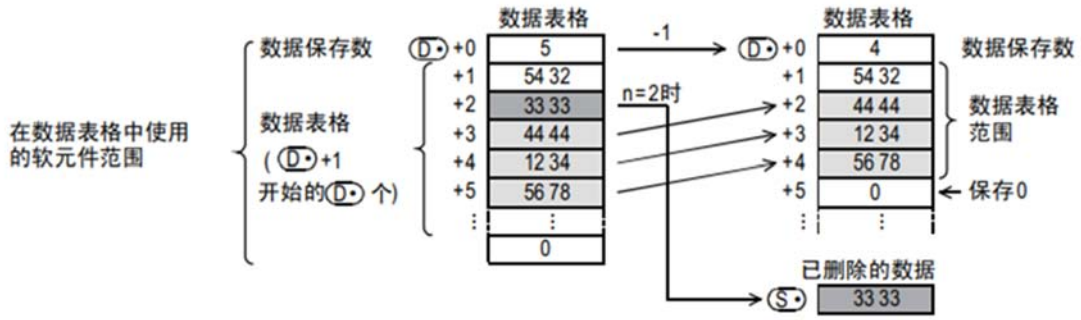
操作数	内容	类型
S	保存被删除的数据	BIN16 位
D	数据表格起始软元件	BIN16 位
n	要删除的数据在表格的位置	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

删除数据表格(D 起始)的第 n 个数据，将删除的数据保存到 S 中。数据表格的第 n+1 个开始的数据逐个向前靠拢，数据保存数减 1。



3、程序举例

略。

4、注意事项

（1）数据表格的范围，为数据保存数 D 的下一个软元件(D+1)开始的 D 个。

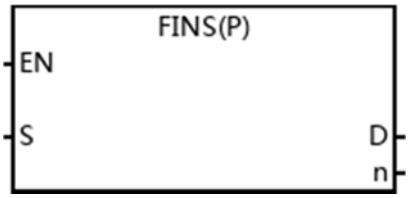
（2）以下情况报错 6706：

- 要删除的数据的表格位置 n 比数据保存数更大时。
- n 的值超出了数据表格 D 的软元件范围时。
- 在 n≦0 的情况执行了指令时。
- 数据保存数 D 的值为 0 时。
- 数据表格的范围超出了相应的软元件范围时。

2.15.2 FINS/数据表的数据插入指令

在数据表格中的任意位置处插入数据的指令。

1、指令格式



操作数说明

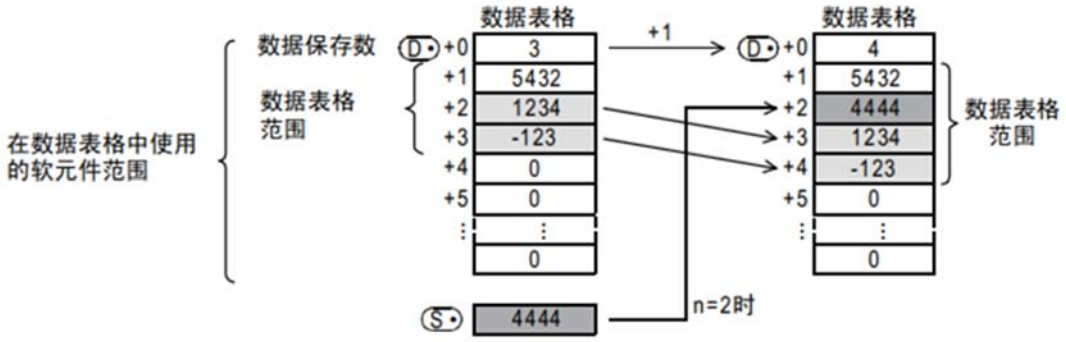
操作数	内容	类型
S	保存插入数据的软元件编号	BIN16 位
D	数据表格的起始软元件编号	BIN16 位
n	插入数据在表格的位置	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

将 16 位数据 S 插入到数据表格(D 以后)的第 n 号中。数据表格的第 n 号以后的数据逐个后移，数据保存数加 1。



3、程序举例

略。

4、注意事项

（1）数据表格的范围，为数据保存数 D 的下一个软元件(D+1)开始的 D 个。

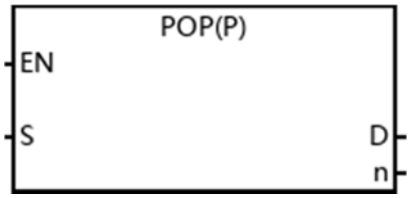
（2）以下情况报错 6706：

- 要插入的数据的表格位置 n 比数据保存数+1 更大时。
- n 的值超出了数据表格 D 的软元件范围时。
- 在 n≦0 的情况执行了指令时。
- 数据表格的范围超出了相应的软元件范围时。

2.15.3 POP/读取后入的数据指令

先入后出形式的数据读出。

1、指令格式



操作数说明

操作数	内容	类型
S	要读出的数据的起始字元件 (S 是数据点数，数据是从 S+1 开始的)	BIN16 位
D	读出的数据	BIN16 位
n	保存的数据点数+1（2≦n≦512）	BIN16 位

操作数可用元件类型

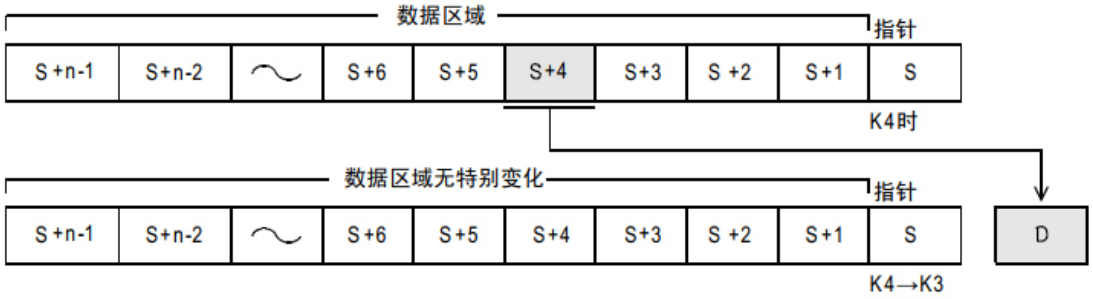
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S									*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					
n																			*	*			

2、功能和动作说明

先入后出控制用数据

值	内容
S	指针数据（被保存的数据个数）
S+1	数据区域（使用唯一写入指令 SFWR 被先入的数据）
S+2	
···	
S+n-2	
S+n-1	

对于（S～S+n-1）的字软元件，每次执行指令时，读出（S +指针数据 S）的软元件保存到 D 中（使用先入先出控制用的移位写入指令 (SFWR)写入的最后的数据读出到 D 中），执行一次指针数据 S 的值减 1。



3、程序举例

略。

4、注意事项

（1）SM8080 的动作，当指针 S=0 时，执行指令后变为 ON，且不处理指令。

（2）指针的当前值为 1 时，S 中被写入 0，SM8080 置 ON。

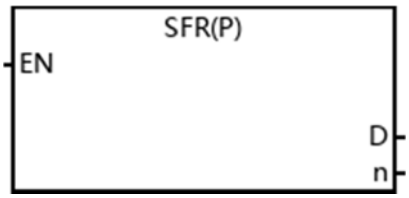
（3）以下情况报错 6706：

- S ＞ n-1 时。
- S ＜ 0 时。

2.15.4 SFR/16 位数据 n 位右移（带进位）指令

使字软元件中的 16 位向右移动 n 位的指令

1、指令格式



操作数说明

操作数	内容	类型
D	保存要移动的数据的起始字软元件编号	BIN16 位
n	移动的次数 0≦n≦15	BIN16 位

操作数可用元件类型

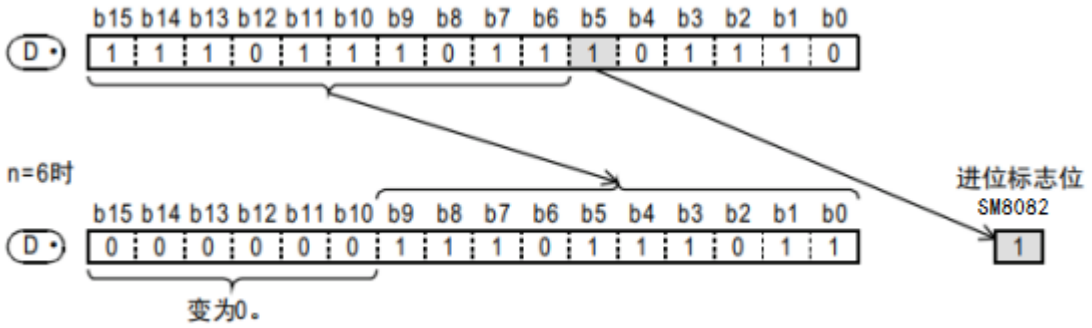
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n								*	*	*	*	*	*	*	*	*	*		*	*			

2、功能和动作说明

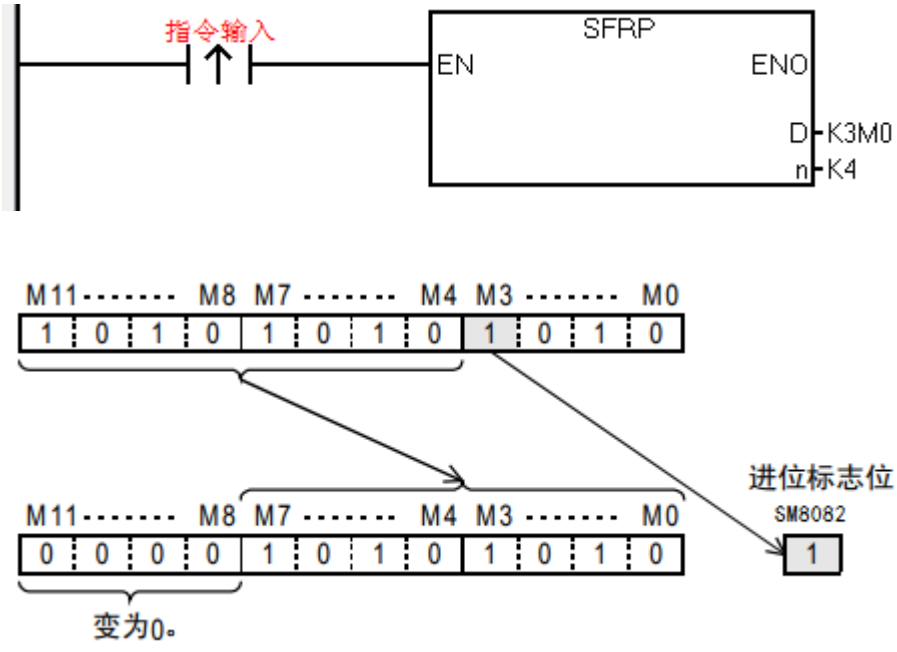
（1）字软元件 D 中的 16 位右移 n 位。n 指定 0～15 的数字。当 n 中指定了 16 以上的数值时，根据 n/16 的余数移动。例如，如 n=18 时，18/16=1 余 2，所以右移 2 位。

（2） 将字软元件 D 中的第 n 位(n-1 位)的 ON(1)/OFF(0)状态转移到进位标志 SM8082 中。

（3） 最高位开始的 n 位变为 0。



（4）通过位数指定来指定位软元件时，将指定位数(4×K□)的数据进行移位。



3、程序举例

略。

4、注意事项

（1）SM8082 进位标志的动作，根据移动 n-1 的状态 ON/OFF。

（2）n 为负值时报错 6706。

2.15.5 SFL/16 位数据 n 位左移（带进位）指令

使字软元件中的 16 位向左移动 n 位的指令

1、指令格式



操作数说明

操作数	内容	类型
D	保存要移动的数据的起始字软元件编号	BIN16 位
n	移动的次数 0≦n≦15	BIN16 位

操作数可用元件类型

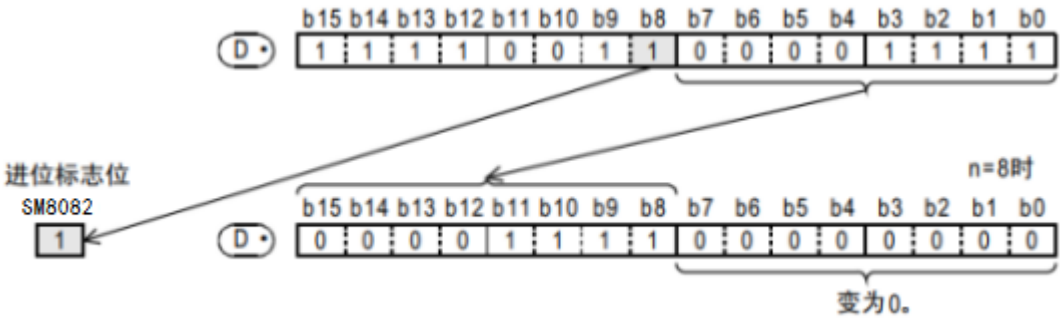
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n								*	*	*	*	*	*	*	*	*	*		*	*			

2、功能和动作说明

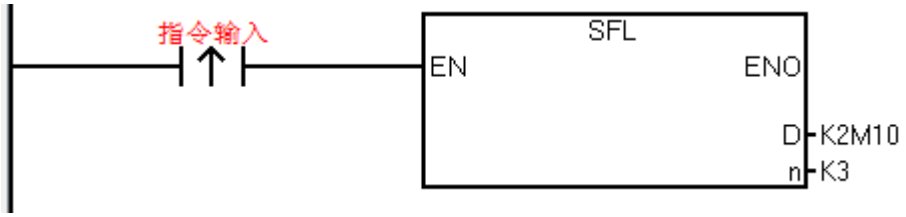
（1）字软元件 D 中的 16 位左移 n 位。n 指定 0～15 的数字。当 n 中指定了 16 以上的数值时，根据 n/16 的余数移动。例如，如 n=18 时，18/16=1 余 2，所以右移 2 位。

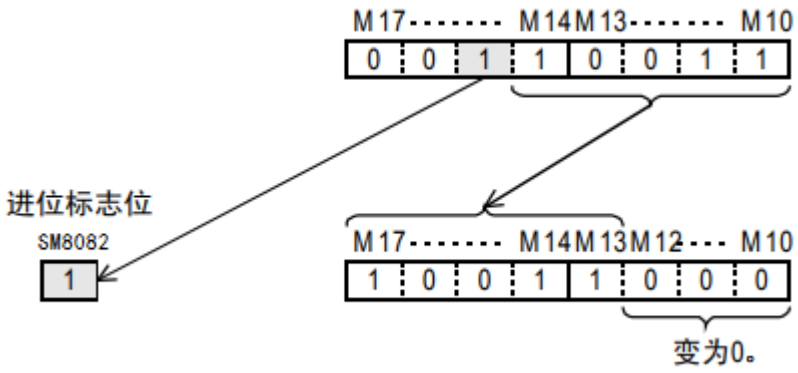
（2） 将字软元件 D 中的第 n+1 位(n 位)的 ON(1)/OFF(0)状态转移到进位标志 SM8082 中。

（3） 最低位开始的 n 位变为 0。



（4）通过位数指定来指定位软元件时，将指定位数(4×K□)的数据进行移位。





3、程序举例

略。

4、注意事项

（1）SM8082 进位标志的动作，根据移动 n 的状态 ON/OFF。

（2）n 为负值时报错 6706。

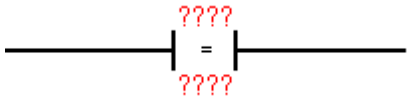
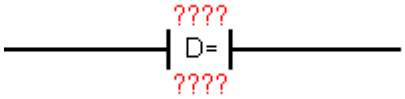
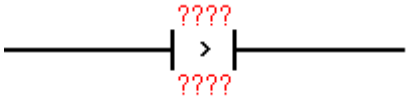
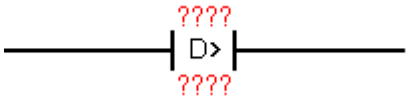
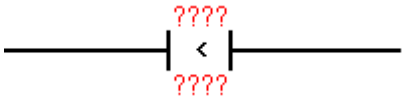
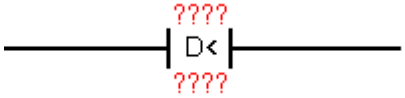
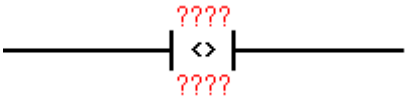
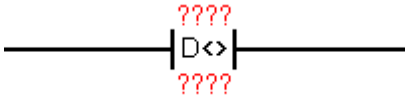
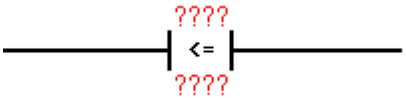
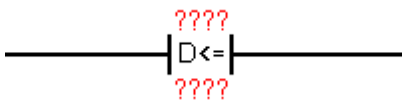
2.16 触点比较指令

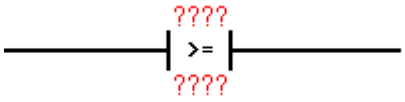
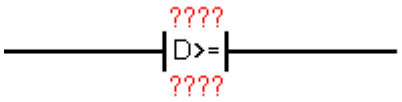
提供使用触点符号进行数据比较的指令，在 UNISYS 中显示出来的是触点比较符号。

2.16.1 LD=、>、<、<>、<=、>=/触点比较指令

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

1、指令格式

指令位数	指令分类	指令格式
16 位	LD=	
32 位	LDD=	
16 位	LD>	
32 位	LDD>	
16 位	LD<	
32 位	LDD<	
16 位	LD<>	
32 位	LDD<>	
16 位	LD<=	
32 位	LDD<=	

16 位	LD>	
32 位	LDD>=	

操作数说明

操作数	内容	类型
S1	比较数据 1	BIN16/32 位
S2	比较数据 2	BIN16/32 位

操作数可用元件类型

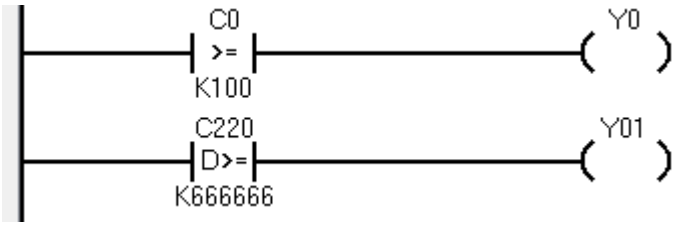
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			

2、功能和动作说明

连接在母线上的触点比较指令。对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

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

3、程序举例



C0 值>=100 时，Y0 被驱动，C220 值>=66666 时 Y1 被驱动。

4、注意事项

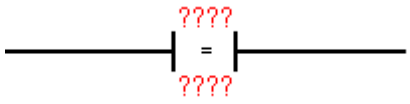
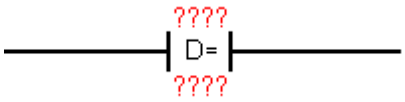
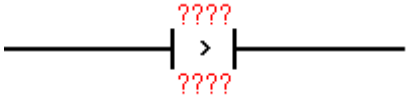
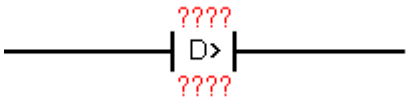
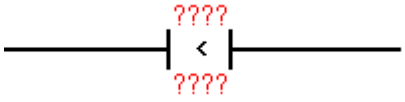
（1）注意当比较源是 32 位计数器（C200~C255）的时候选择 32 位指令。

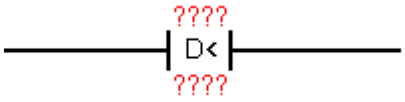
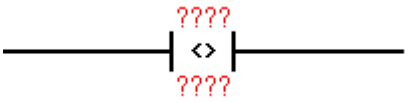
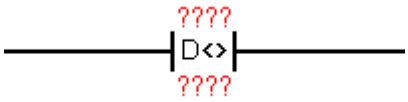
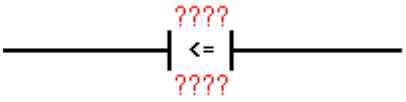
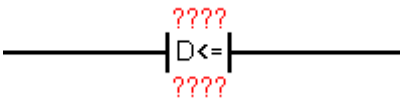
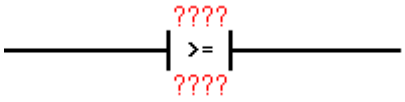
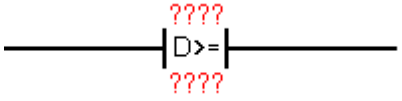
（2）注意指令的书写方式。

2.16.2 AND=、>、<、<>、<=、>=/触点比较指令

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

1、指令格式

指令位数	指令分类	指令格式
16 位	AMD=	
32 位	ANDD=	
16 位	AND>	
32 位	ANDD>	
16 位	AND<	

32 位	ANDD<	
16 位	AND<>	
32 位	ANDD<>	
16 位	AND<=	
32 位	ANDD<=	
16 位	AND>	
32 位	ANDD>=	

操作数说明

操作数	内容	类型
S1	比较数据 1	BIN16/32 位
S2	比较数据 2	BIN16/32 位

操作数可用元件类型

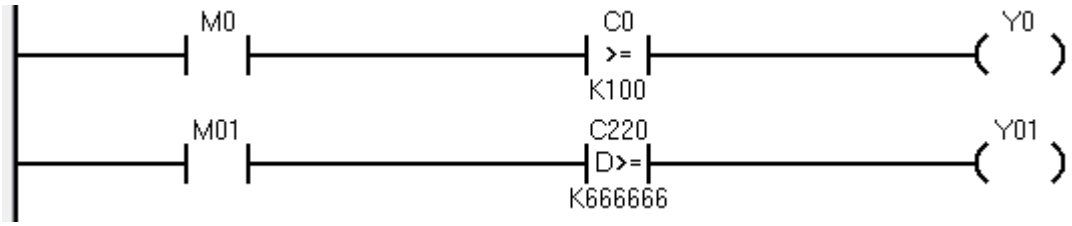
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			

2、功能和动作说明

与其他触点串联的触点比较指令。对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

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

3、程序举例



M0 为 ON 且 C0 值>=100 时，Y0 被驱动；M1 为 ON 且 C220 值>=66666 时 Y1 被驱动。

4、注意事项

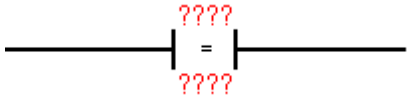
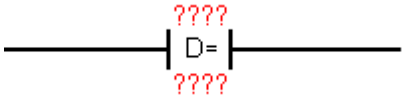
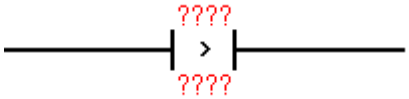
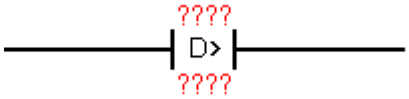
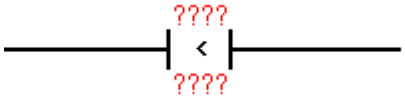
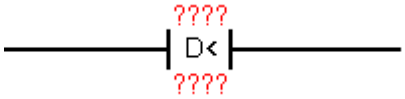
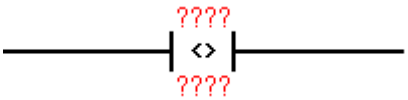
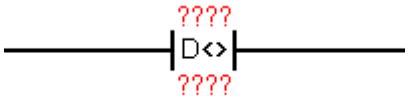
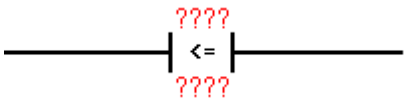
（1）注意当比较源是 32 位计数器（C200~C255）的时候选择 32 位指令。

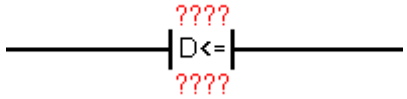
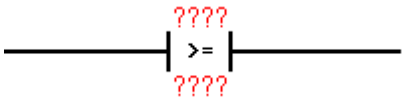
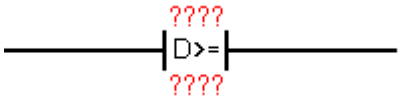
（2）注意指令的书写方式。

2.16.3 OR=、>、<、<>、<=、>=/触点比较指令

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

1、指令格式

指令位数	指令分类	指令格式
16 位	OR=	
32 位	ORD=	
16 位	ORD>	
32 位	ORD>	
16 位	OR<	
32 位	ORD<	
16 位	OR<>	
32 位	ORD<>	
16 位	OR<=	

32 位	ORD<=	
16 位	OR>	
32 位	ORD>=	

操作数说明

操作数	内容	类型
S1	比较数据 1	BIN16/32 位
S2	比较数据 2	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			

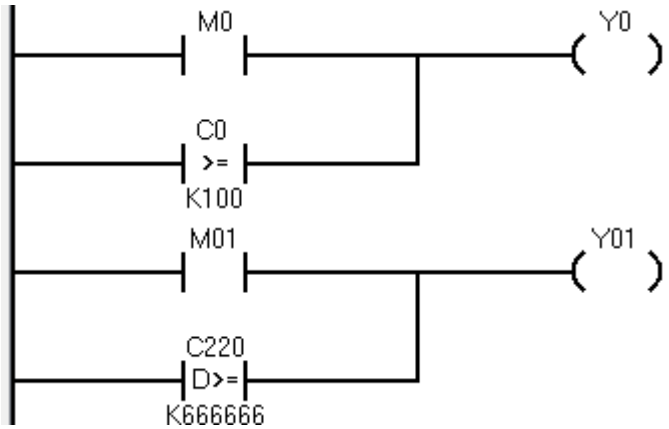
2、功能和动作说明

与其他触点并联的触点比较指令。对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

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

OR>=	ORD>=	S1≥S2	S1<S2

3、程序举例



M0 为 ON 或 C0 值>=100 时，Y0 被驱动；M1 为 ON 或 C220 值>=66666 时 Y1 被驱动。

4、注意事项

（1）注意当比较源是 32 位计数器（C200~C255）的时候选择 32 位指令。

（2）注意指令的书写方式。

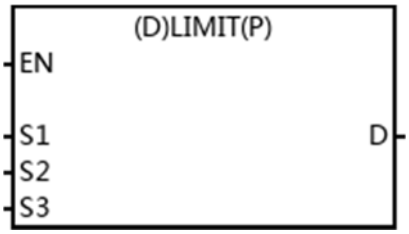
2.17 数据表处理指令

主要用于输入数据通过群数据时对输出数据的控制。

2.17.1 LIMIT/上下限位控制指令

设置输入数值的上限值/下限值然后输出的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	下限限位值	BIN16/32 位
S2	上限限位值	BIN16/32 位

S3	要通过上下限限位控制的输入值	BIN16/32 位
D	经过上下限限位控制的输出值	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2								*	*	*	*	*	*	*	*			*	*	*			
S3								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

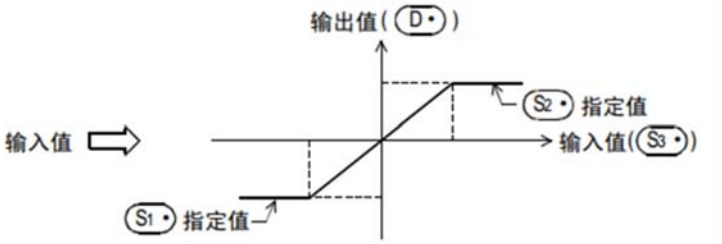
2、功能和动作说明

通过判断 S3 中指定的输入值，是否在 S1、S2 指定的上下限值的范围内，以此控制保存在 D 指定的软元件中的输出值。

下限值 > 输入值时，输出值 = 下限值；

上限值 < 输入值时，输出值 = 上限值；

下限值 ≤ 输入值 ≤ 上限值时，输出值 = 输入值。



3、程序举例

略。

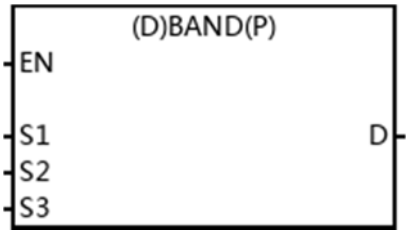
4、注意事项

执行指令后，若设定的下限值 > 上限值时报错 6706。

2.17.2 BAND/死区控制指令

通过判断输入值是否在指定的死区的上下限范围内，从而来控制输出值的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	死区(无输出区域)的下限值	BIN16/32 位
S2	死区(无输出区域)的上限值	BIN16/32 位
S3	要通过死区控制的输入值	BIN16/32 位
D	经过死区控制的输出值	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2								*	*	*	*	*	*	*	*			*	*	*			
S3								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

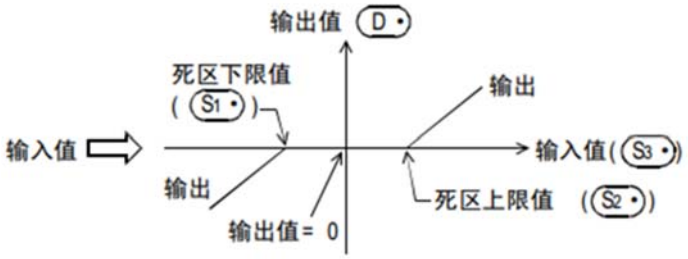
2、功能和动作说明

通过判断 S3 指定的输入值是否在 S1、S2 指定的死区范围内，以此来控制保存在 D 指定的软元件中的输出值。

下限值 > 输入值时，输出值 = 输入值 - 下限值；

上限值 < 输入值时，输出值 = 输入值 -上限值；

下限值 ≤ 输入值 ≤ 上限值时，输出值 = 0。



3、程序举例

略。

4、注意事项

（1）输出值溢出时，输出值 = 输入值 - 死区下限值

16 位运算：运算结果超出-32768~32767 时

例：S1 = 10，S3 = -32768，输出 D = -32768 - 10 = 8000H - AH = 7FF6H = 32758

32 位运算：运算结果超出-2147483648～2147483647 时

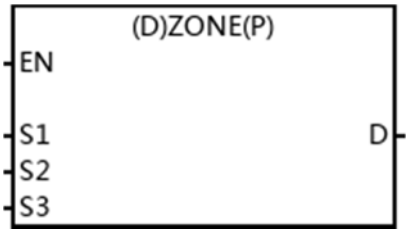
例：（S1+1,S1）= 1000，（S3+1，S3） = -2147483648，输出（D+1，D）= -2147483648 - 1000 = 80000000H - 000003E8H = 7FFFFC18H = 2147482648

（3）指令运行时，如果死区下限值 > 死区上限值时报错 6706

2.17.3 ZONE/区域控制指令

根据输入值是正数还是负数，用指定的偏差值来控制输出值的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	负偏差值	BIN16/32 位
S2	正偏差值	BIN16/32 位

S3	要通过区域控制的输入值	BIN16/32 位
D	通过区域控制的输出值	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2								*	*	*	*	*	*	*	*			*	*	*			
S3								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

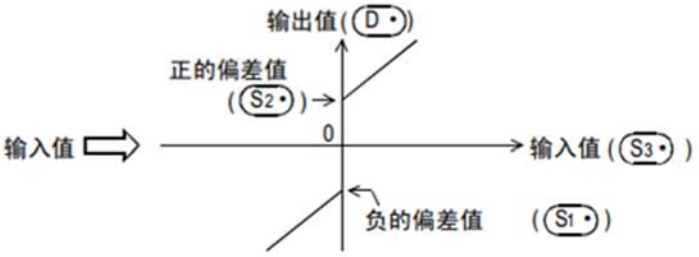
2、功能和动作说明

在 S3 指定的输入值上加上 S1 或 S2 指定的偏差值，然后保存到 D 指定的软元件编号中。

输入值 < 0 时，输出值 = 输入值 + 负的偏差值；

输入值 = 0 时，输出值 = 0；

输入值 > 0 时，输出值 = 输入值 + 正的偏差值。



3、程序举例

略。

4、注意事项

（1）输出值溢出时，输出值 = 输入值 + 负的偏差值

16 位运算：运算结果超出-32768~32767 时

例：S1 = -100，S3 = -32768，输出 D = -32768 + （-100） = 8000H + FF9CH = 7F9CH = 32668

32 位运算：运算结果超出-2147483648～2147483647 时

例：（S1+1,S1）= -1000，（S3+1，S3） = -2147483648，输出（D+1，D）= -2147483648 + （-1000） = 80000000H - FFFFFC18H = 7FFFFC18H = 2147482648

（2）指令运行时，如果死负的偏差值 > 正的偏差值时报错 6706

2.17.4 SCL/定坐标（不同点坐标数据）指令

根据指定的数据表格，对输入值执行定坐标后输出的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	X 坐标点	BIN16/32 位
S2	组成坐标系的数据表格	BIN16/32 位
D	输出 Y 坐标点	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2														*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

（1）16 位指令

根据指定的转换特性，对 S1 指定的输入值执行定坐标，然后保存到 D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2 指定的软元件开始的数据表格执行的。但是，输出数据不是整数值时，小数第 1 位四舍五入后输出。

（2）32 位指令

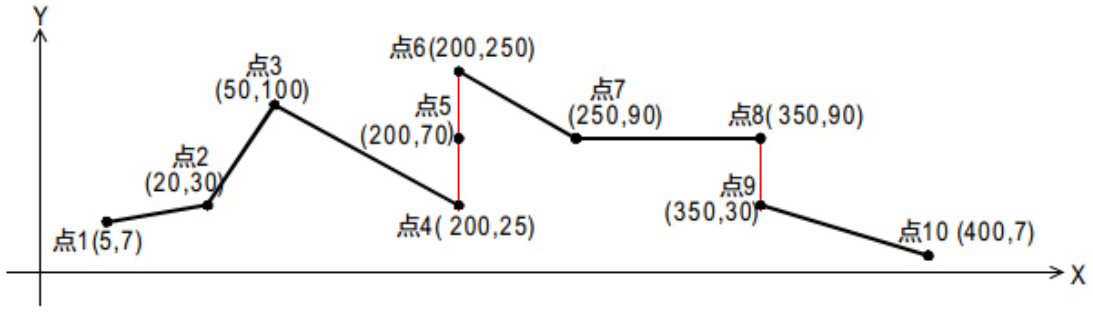
根据指定的转换特性，对（S1+1，S1）指定的输入值执行定坐标，然后保存到（D+1，D)指定的软元件编号中。定坐标用的转换，是依据保存在（S2+1,S2）指定的软元件开始的数据表格执行的。但是，输出数据不是整数值时，小数第 1 位四舍五入后输出。

3、程序举例

（1）定坐标用转换表格的设定

设定项目		设定数据表格的软元件分配	
		16 位运算	32 位运算
坐标点数		S2	（S2+1，S2）
点 1	X 坐标	S2+1	（S2+3，S2+2）
	Y 坐标	S2+2	（S2+5，S2+4）
点 2	X 坐标	S2+3	（S2+7，S2+6）
	Y 坐标	S2+4	（S2+9，S2+8）
···	···	···	···
点 n	X 坐标	S2+2n-1	（S2+4n-1，S2+4n-2）
	Y 坐标	S2+2n	（S2+4n+1，S2+4n）

（2）以 16 位运算为例，执行 32 位运算时按 32 位数据格式设定表格数据



设定项目	设定软元件及设定内容			备注
	S2 中指定了 R0 时		设定内容	
坐标点数	S2	R0	K10	

点 1	X 坐标	S2+1	R1	K5	
	Y 坐标	S2+2	R2	K7	
点 2	X 坐标	S2+3	R3	K20	
	Y 坐标	S2+4	R4	K30	
点 3	X 坐标	S2+5	R5	K50	
	Y 坐标	S2+6	R6	K100	
点 4	X 坐标	S2+7	R7	K200	如果象这样指定 3 点的坐标，则输出值为中间值。这个例子中，将点 5 的 y 坐标指定为输出值(中间值)。 此外，3 点以上的 X 坐标相同时，也输出第 2 点的数值 数值
	Y 坐标	S2+8	R8	K25	
点 5	X 坐标	S2+9	R9	K200	
	Y 坐标	S2+10	R10	K70	
点 6	X 坐标	S2+11	R11	K200	
	Y 坐标	S2+12	R12	K250	
点 7	X 坐标	S2+13	R13	K250	如果象这样指定 2 点的坐标，则输出值取后一个点的 y 坐标值。 这个例子中，将点 9 的 y 坐标指定为输出值
	Y 坐标	S2+14	R14	K90	
点 8	X 坐标	S2+15	R15	K350	
	Y 坐标	S2+16	R16	K90	
点 9	X 坐标	S2+17	R17	K350	
	Y 坐标	S2+18	R18	K30	
点 10	X 坐标	S2+19	R19	K400	
	Y 坐标	S2+20	R20	K7	

4、注意事项

（1）当 X 坐标或 Y 坐标相同时，按上图表格中所写方式输出结果。

（2）以下情况报错 6706：

• 数据表格的 Xn 数据没有按照升序排列时。但是，由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。

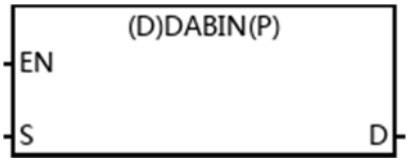
• S1 在数据表格设定的范围以外时。

• 运算过程中的数值超出了 32 位数据的范围时。此时，请确认各点之间的距离没有超出 65535 以上。如果超出 65535 时，请缩短各点之间的距离。

2.17.5 DABIN/10 进制 ASCII→BIN 的转换指令

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

1、指令格式



操作数说明

操作数	内容	类型
S	保存要转换成 BIN 值的数据(ASCII 码)的软元件起始编号	字符串
D	保存转换结果的软元件编号	BIN16/32 位

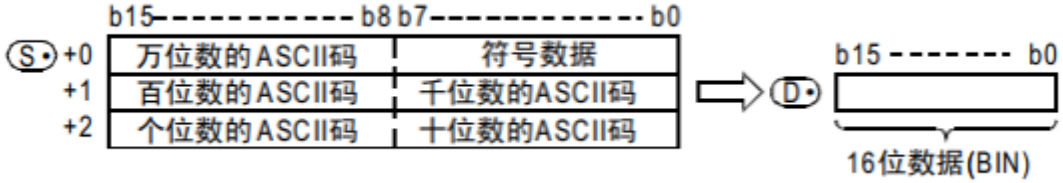
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

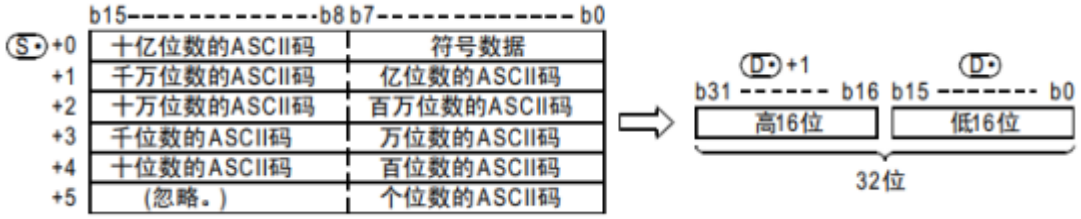
（1）16 位指令

将以 10 进制数字的 ASCII 码（30H ～ 39H）显示的 S~S+2 的数据转换成 16 位数据(BIN) 后，保存到 D 中。



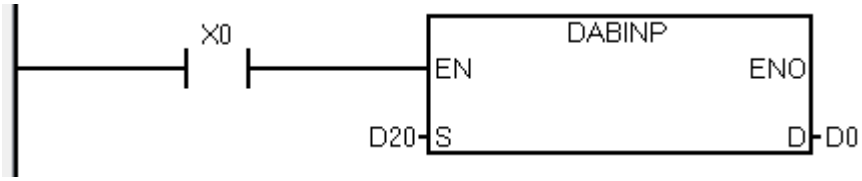
（2）32 位指令

将以 10 进制数字的 ASCII 码（30H ～ 39H）显示的 S~S+5 的数据转换成 32 位数据(BIN) 后，保存到（D+1，D）中。



3、程序举例

当 X000 为 ON 时，将 D20～D22 中设定的符号以及 5 位 10 进制数字的 ASCII 码数据转换成 BIN 值后，保存到 D0 中的程序。



显示格式： 16I 32I 32E ASC 10 16

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D20	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	-
D21	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	2
D22	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	34

4、注意事项

（1）各位数的 ASCII 码位“20H”、“00H（NULL）”时，作为“30H”处理。

（2）以下情况报错 6706：

• 符号数据(S 的低位字节)中，要转换的数据为正时设定“20H(空格)”，为负时设定"2DH(-)"，设定其它以外值报错。

• S～S+2(5)的各位数的 ASCII 码为“30H”～“39H”、“20H(空格)”“00H(NULL)”以外的值时。

• S～S+2(5)的数值范围为下述的范围以外时：

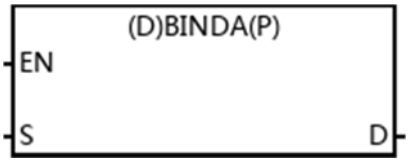
16 位运算：-32768~32767；32 位运算：-2147483648~2147483647。

• S～S+2(5)超出软元件范围时。

2.17.6 BINDA/BIN→10 进制 ASCII 的转换指令

将 BIN 数据转换成 ASCII 码(30H～39H)的指令。

1、指令格式



操作数说明

操作数	内容	类型
S	保存要转换成 ASCII 码的 BIN 数据的软元件编号	BIN16/32 位
D	保存转换结果的软元件起始编号	字符串

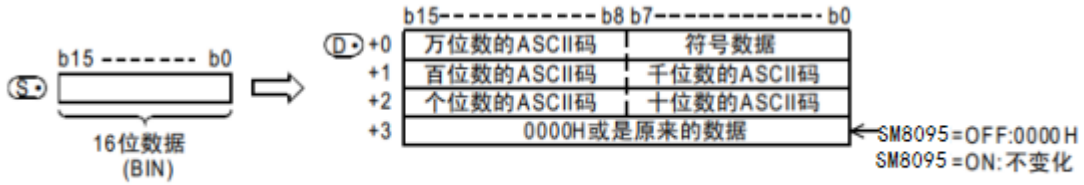
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D												*	*	*	*			*					

2、功能和动作说明

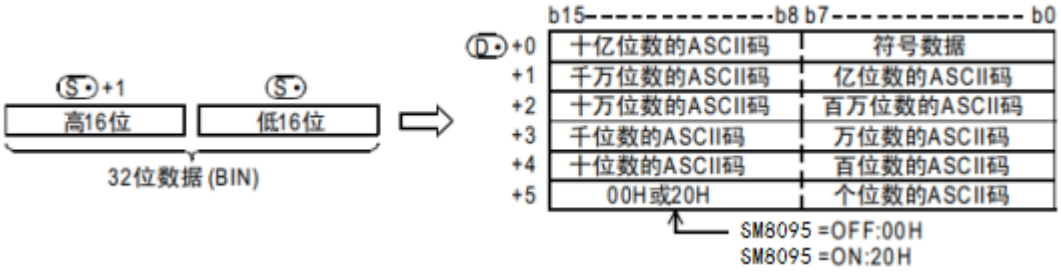
（1）16 位指令

将 S 的 16 位数据(BIN)按照 10 进制的各个位数转换成 ASCII 码(30H～39H)，然后保存到 D 中。



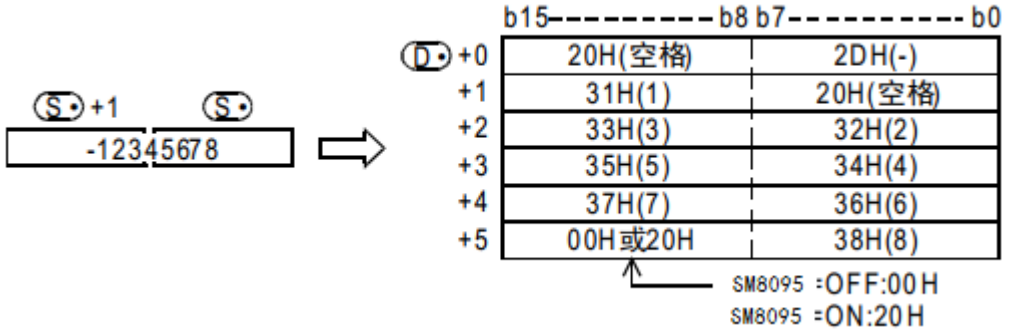
（2）32 位指令

将（S+1，S）的 32 位数据(BIN)按照 10 进制的各个位数转换成 ASCII 码(30H～39H)，然后保存到 D 开始的软元件中。



3、程序举例

例如，（S+1，S）为-12,345,678 时，D 开始的软元件中如下所示地进行保存。



4、注意事项

（1）相关软元件的动作：

软元件	名称	内容	
SM8095	输出字符数切换信号	16 位运算	32 位运算
		SM8095=OFF 时，D+3 为 0000H（NULL）。 SM8095=ON 时，D+3 不变化。	SM8095=OFF 时，D+5 高字节为 00H（NULL）。 SM8095=ON 时，D+5 高字节为 20H（空格）。

要注意 16 位运算时 SM8095 为 OFF 的情况，要占用 4 个字软元件。

（2）符号数据(D 的低位字节)中，要转换的数据为正时保存“20H(空格)”，为负时保存"2DH(-)"。

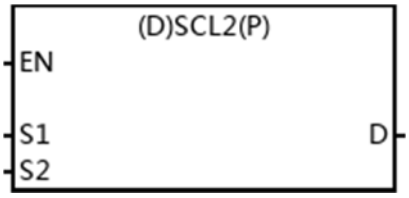
（3）当 S 中数据的总位数低于 D 能存储的总位数时，有效位数的左侧保存“20H（空格）”，如上面的例子。

（4）D 的占用点数不能超出改软元件的范围，超出报错 6706。

2.17.7 SCL2/定坐标 2（X/Y 坐标数据）指令

根据指定的数据表格，对输入值执行定坐标后输出的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	X 坐标点	BIN16/32 位
S2	组成坐标系的数据表格	BIN16/32 位
D	输出 Y 坐标点	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2														*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

（1）16 位指令

根据指定的转换特性，对 S1 指定的输入值执行定坐标，然后保存到 D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2 指定的软元件开始的数据表格执行的。但是，输出数据不是整数值时，小数第 1 位四舍五入后输出。

（2）32 位指令

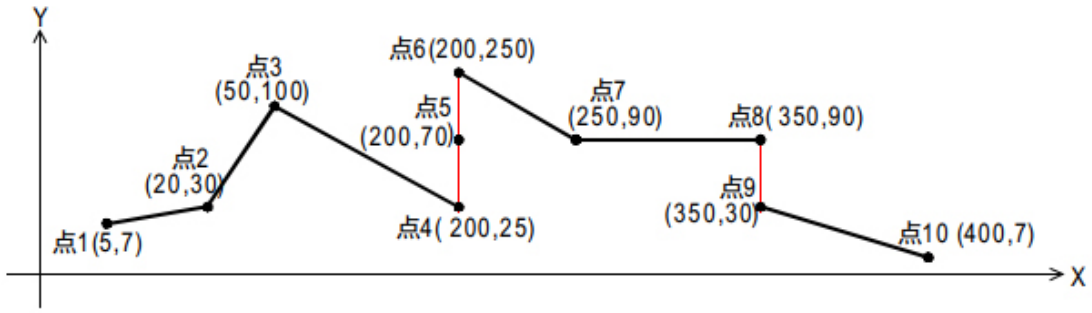
根据指定的转换特性，对（S1+1，S1）指定的输入值执行定坐标，然后保存到（D+1，D)指定的软元件编号中。定坐标用的转换，是依据保存在（S2+1,S2）指定的软元件开始的数据表格执行的。但是，输出数据不是整数值时，小数第 1 位四舍五入后输出。

3、程序举例

（1）定坐标用转换表格的设定

设定项目		设定数据表格的软元件分配	
		16 位运算	32 位运算
坐标点数		S2	（S2+1，S2）
X 坐标	点 1	S2+1	（S2+3，S2+2）
	点 2	S2+2	（S2+5，S2+4）
	···	···	···
	点 n（最后）	S2+n	（S2+2n+1，S2+2n）
Y 坐标	点 1	S2+n+1	（S2+2n+3，S2+2n+2）
	点 2	S2+n+2	（S2+2n+5，S2+2n+4）
	···	···	···
	点 n（最后）	S2+2n	（S2+4n+1，S2+4n）

（2）以 16 位运算为例，执行 32 位运算时按 32 位数据格式设定表格数据



设定项目		设定软元件及设定内容			备注
		S2 中指定了 R0 时		设定内容	
坐标点数		S2	R0	K10	
X 坐标	点 1	S2+1	R1	K5	
	点 2	S2+2	R2	K20	
	点 3	S2+3	R3	K50	
	点 4	S2+4	R4	K200	*1
	点 5	S2+5	R5	K200	
	点 6	S2+6	R6	K200	
	点 7	S2+7	R7	K250	
	点 8	S2+8	R8	K350	*2
	点 9	S2+9	R9	K350	
	点 10	S2+10	R10	K400	

Y 坐标	点 1	S2+11	R11	K7	
	点 2	S2+12	R12	K30	
	点 3	S2+13	R13	K100	
	点 4	S2+14	R14	K25	*1
	点 5	S2+15	R15	K70	
	点 6	S2+16	R16	K250	
	点 7	S2+17	R17	K90	
	点 8	S2+18	R18	K90	*2
	点 9	S2+19	R19	K30	
	点 10	S2+20	R20	K7	

*1. 如果像点 4、5、6 那样指定 3 点的坐标时则输出值为中间值。在这个例子中，将点 5 的 y 坐标指定为输出值(中间值)。此外，即使 3 点以上的 X 坐标相同时，输出第 2 点的数值。

*2. 如果像点 8、9 那样指定 2 点的坐标，则输出值取后一个点的 y 坐标的值。在这个例子中，将点 9 的 y 坐标指定为输出值。

4、注意事项

（1）当 X 坐标或 Y 坐标相同时，按例程中所写方式输出结果。

（2）以下情况报错 6706：

• 数据表格的 Xn 数据没有按照升序排列时。但是，由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。

• S1 在数据表格设定的范围以外时。

• 运算过程中的数值超出了 32 位数据的范围时。此时，请确认各点之间的距离没有超出 65535 以上。如果超出 65535 时，请缩短各点之间的距离。

2.18 扩展文件寄存器指令

ER 寄存器存储在外部 Flash 对于一些想长期保存的数据可以使用 ER 寄存器保存。

段编号与起始软元件编号对照表：

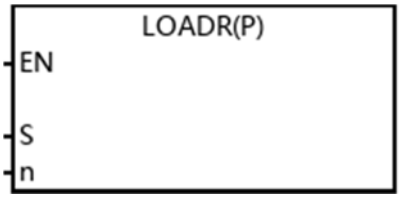
段编号	起始软元件编号	写入软元件范围	段编号	起始软元件编号	写入软元件范围
段 0	R0	ER0~ER2047	段 8	R16384	ER16384~ER18431
段 1	R2048	ER2048~ER4095	段 9	R18432	ER18432~ER20479

段 2	R4096	ER4096~ER6143	段 10	R20480	ER20480~ER22527
段 3	R6144	ER6144~ER8191	段 11	R22528	ER22528~ER24575
段 4	R8192	ER8192~ER10239	段 12	R24576	ER24576~ER26623
段 5	R10240	ER10240~ER12287	段 13	R26624	ER26624~ER28671
段 6	R12288	ER12288~ER14335	段 14	R28672	ER28672~ER30719
段 7	R14336	ER14336~ER16383	段 15	R30720	ER30720~ER32767

2.18.1 LOADR/读出扩展文件寄存器指令

将扩展文件寄存器 ER 的当前值读到 R 寄存器中。

1、指令格式



操作数说明

操作数	内容	类型
S	保存数据的扩展寄存器的软元件编号(作为数据传送源的扩展文件寄存器与 S 的编号相同)	BIN16 位
n	读出点数（0≦n≦32767)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n														*					*	*			

2、功能和动作说明

将存储在外部 FLASH 的 ER 寄存器读出到对应的 R 寄存器，以点为单位进行读出，最多可以读出 32768 点。

3、程序举例

略。

4、注意事项

当 n=0 时，按 n=32768 执行指令。

2.18.2 SAVER/成批写入扩展文件储存器指令

将任意点数的扩展寄存器(R)的当前值 1 段(2048 点)写入扩展文件寄存器(ER)中时使用。

1、指令格式



操作数说明

操作数	内容	类型
S	保存数据的扩展寄存器的软元件编号，但是，只可以指定扩展寄存器的段的起始软元件编号	BIN16 位
n	每个周期写入点数（0≤n≤2048）	BIN16 位
D	记录已经写入的点数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n																			*	*			
D														*				*					

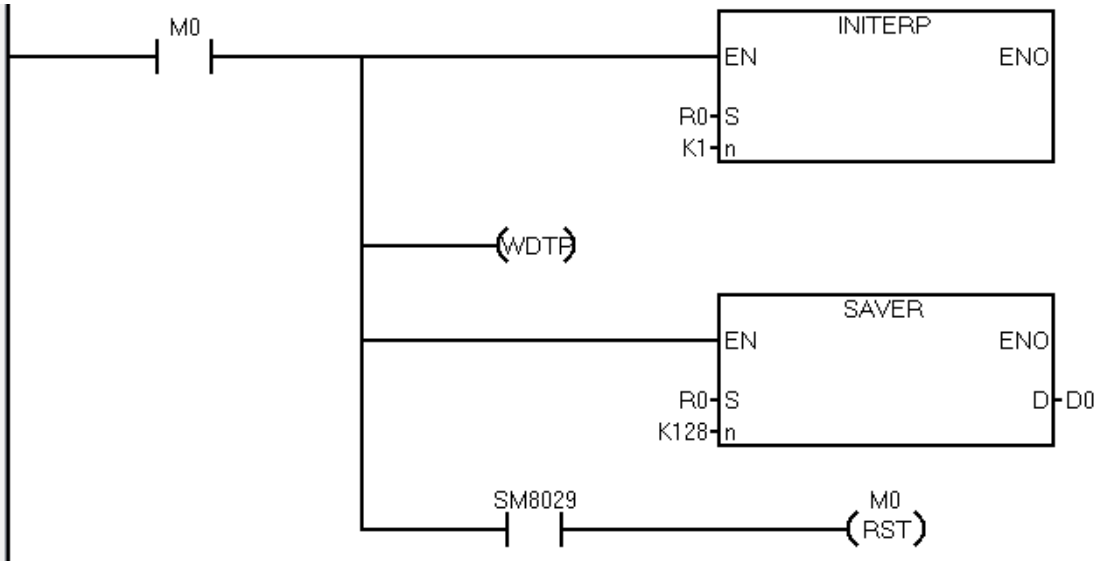
2、功能和动作说明

通过 2048/n 个运算周期(如有余数时，为+1 个扫描)将 S～S+2047 的扩展寄存器（R）的内容(当前值)写入具有相同编号的扩展文件寄存器(ER)中。指令执行过程中，在 D 中保存已经写入的点数。

- 每个运算周期写入 n 点。
- 以段为单位(2048 点)执行写入。各段的起始软元件编号对应 2.19 章节的表格。
- 在 n 中指定了 0 的值时，作为 n=2048 执行指令。
- 当 2048 点的写入(传送)结束时，指令执行结束，指令执行结束标志位 SM8029 变为 ON。

3、程序举例

当 X000 为 ON 时，将设定数据用的扩展寄存器 R10～R19(0 段)的变更内容，反映到扩展文件寄存器(ER)中的程序。(每个运算周期写入 128 点)



①先执行 INITER 指令初始化扩展文件寄存器（ER）

②使用 SAVER 指令每个周期写入 128 点，将 R0~R2047 的值都会写入 ER0~ER2047。

4、注意事项

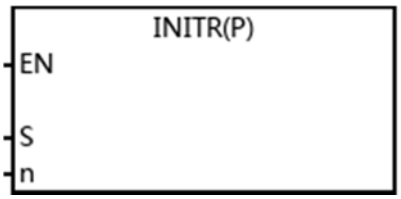
（1）SAVER 指令执行之前必须先执行 INITER（FNC 295）指令或者 INITR（FNC 292）指令进行 ER 数据的初始化，否则报错 6770。

（2）为防止 SAVER 指令执行写入时间过长造成看门狗超时，在此之前刷新看门狗时间或更改比较大的时间保证能完成写入。

2.18.3 INITR/扩展寄存器的初始化指令

对扩展寄存器(R)和存储器盒(闪存)中的扩展文件寄存器(ER)进行初始化(HFFFF<K-1>)时使用本指令。

1、指令格式



操作数说明

操作数	内容	类型
S	要初始化的扩展寄存器和扩展文件寄存器的软元件编号，但是，仅可以指定扩展寄存器的段的起始软元件编号	BIN16 位
n	要初始化的扩展寄存器和扩展文件寄存器的段数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n																			*	*			

2、功能和动作说明

对与 S 指定的扩展寄存器编号相同的扩展文件寄存器开头的 n 个段，执行初始化(写入初始值 HFFFF(K-1))，按段执行初始化。

3、程序举例

无。

4、注意事项

（1）INITR 指令同时初始化相同段中的 2048 个 R 寄存器和 ER 寄存器。

（2）多个段初始化时注意看门狗时间，防止超时。

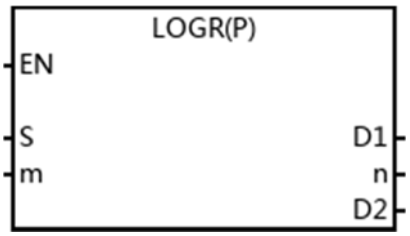
（3）n 的范围为 1~16，为其他数字时报错 6706。

（4）要初始化的软元件编号超出 32767 时，到 R32767（ER32767）为止的软元件执行初始化，并报错 6706。

2.18.4 LOGR/登录到扩展文件寄存器指令

这个指令用于执行指定软元件的登录，并可以将已经登录的数据保存到扩展寄存器(R)以及扩展文件寄存器(ER)中。

1、指令格式



操作数说明

操作数	内容	类型
S	执行登录的对象软元件起始编号	BIN16 位
m	登录点数 1≦m≦8000	BIN16 位
D1	登录中使用的软元件起始编号	BIN16 位
n	登录段数 1≦n≦16	BIN16 位
D2	已登录的数据数	BIN16 位

操作数可用元件类型

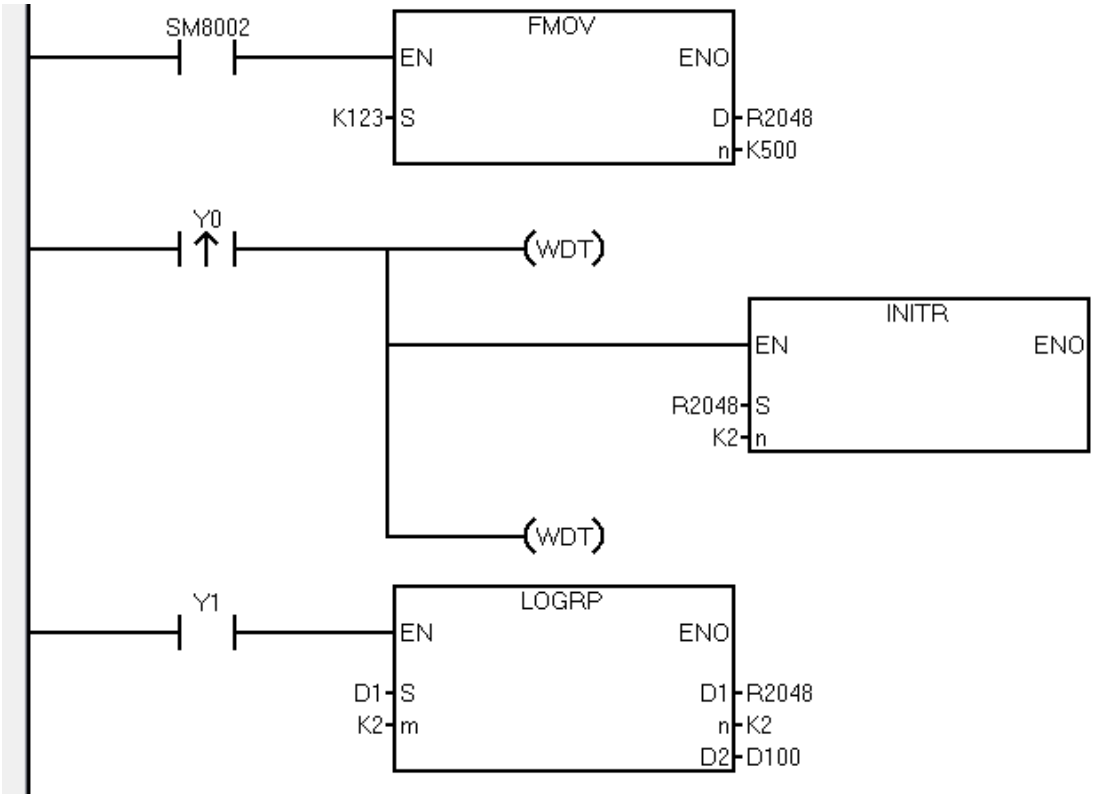
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*				*					
m														*					*	*			
D1															*								
n																			*	*			
D2														*				*					

2、功能和动作说明

在指令被驱动期间，到 D1 开头的扩展寄存器(R)以及扩展文件寄存器(ER)的 n 段的区域被填满为止，一直登录 S 开始的 m 点的软元件。此外，将保存的登录数据的数据数存放在 D2 中。

3、程序举例

每次 Y001 为 ON 时，在 R2048～R6143、ER2048～ER6143 的区域中登录 D1~D2 数据的程序。



第 1 次登陆后：R2048 = D1，R2049 = D2，ER2048 = D1，ER2049 = D2，D100 = 2；

第 2 次登陆后：R2050 = D1，R2051 = D2，ER2050 = D1，ER2051 = D2，D100 = 4；

...

第 2048 次登陆后：R6142 = D1，R6143 = D2，ER6142 = D1，ER6143 = D2，D100 = 4096。

至此，R2048～R6143 这两段区域以及 ER2048～ER6143 这两段区域的数据登录完毕。

4、注意事项

（1）登录数据之前，先使用 INITR 指令，以段为单位进行区域初始化，未初始化就执行该指令，报错 6770。NITR 中的两个操作数要与 LOGR 指令中的 D1，n 对应。

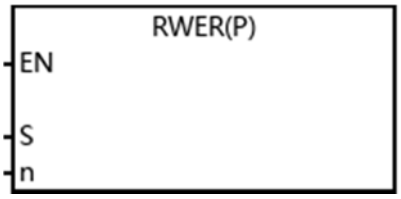
（2）LOGR 指令连续执行时，在每个运算中都执行登录，通过 1 次输入执行 1 次登录时，请使用脉冲执行型。

（3）当登录的点数超过可登录的最大点数时，登录到 R32767（ER32767）为止，报错 6706。

2.18.5 RWER/扩展文件寄存器的写入指令

将扩展寄存器(R)的任意点数的当前值写入扩展文件寄存器(ER)中时使用。

1、指令格式



操作数说明

操作数	内容	类型
S	扩展寄存器段起始地址（只能是段起始地址）	BIN16 位
n	写入点数（0 或 2048 整数倍）	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n														*					*	*			

2、功能和动作说明

S 开始的 n 点扩展寄存器(R)的当前值写入相同编号的扩展文件寄存器(ER)中。ER 寄存器内容存储在外部 FLASH，因此无法监控，只能通过 LOADR 或 LOGR 指令读出，其中 R 的偏移只能是起始段位置，n 只能是 2048 的整数倍或 0。

3、程序举例

略。

4、注意事项

（1）传送点数较多时，注意看门狗超时。

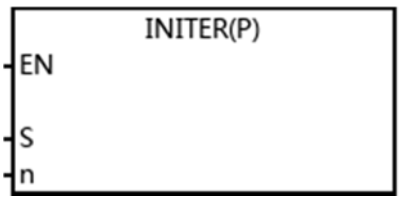
（2）n=0 时，n=32768

（3）要传送的软元件的末尾软元件编号超出了 32767 时,对到末尾软元件编号的 R32767 为止的软元件执行写入。

2.18.6 INITER/扩展文件寄存器的初始化指令

用于在执行 SAVER 指令前，对扩展文件寄存器(ER)进行初始化(HFFFF<K-1>)的情况时使用。

1、指令格式



操作数说明

操作数	内容	类型

S	与要执行初始化的扩展文件寄存器的软元件编号相同的扩展寄存器的软元件编号，但是，仅可以指定扩展寄存器的段的起始软元件编号。	BIN16 位
n	要初始化的扩展文件寄存器的段数（1≤n≤16）	BIN16 位

操作数可用元件类型

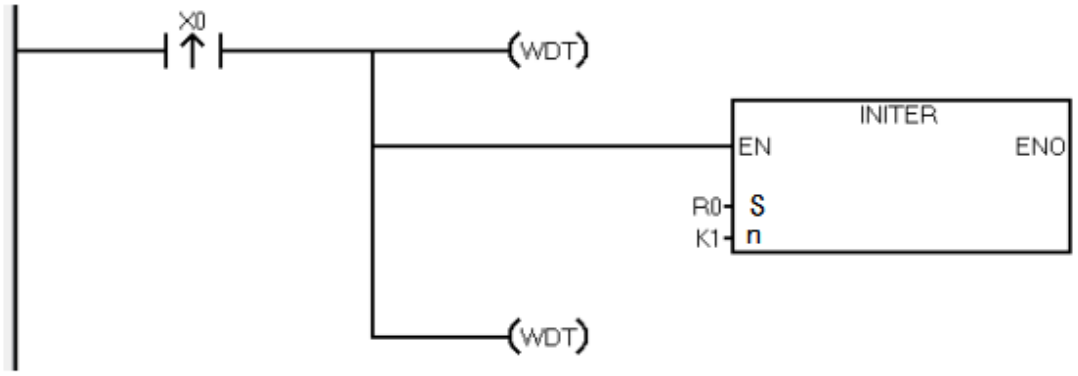
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n																			*	*			

2、功能和动作说明

对与 S 编号相同的扩展文件寄存器(ER) 开头的 n 个段，执行初始化(写入初始值 HFFFF(K-1))，按段执行初始化

3、程序举例

对段 0 中的扩展文件寄存器 ER0～ER2047 进行初始化的程序。



扩展文件寄存器(ER) 内容的变化如下，当然这个寄存器的值在上位机上是观察不到的，因为存储在外部 FLASH，只能通过对应的读写文件寄存器的命令读出来观察。

软元件编号	当前值	
	执行前	执行后
ER0	H1234	HFFFF
ER1	H5678	HFFFF

...		HFFFF
ER2047	HCDEF	HFFFF

4、注意事项

（1）对多段数据进行初始化时，注意看门狗超时。

（2）当 S 设定了 ER 段的起始软元件编号以外的数字时报错 6706。

（3）要初始化的软元件的软元件编号超出了 32767 时,初始化到 32767 为止。

2.19 运动控制指令

概述

X5S 运动控制功能支持点位运动、速度运动、直线插补、圆弧插补、追剪、旋切、随动等，X5 在此基础上还支持电子齿轮、电子凸轮等高级高级运动控制功能。根据实际应用选择对应的指令组合，指令使用过程中，注意相关的寄存器使用方法，还有对应参数类型可选择的数据范围及实际意义。

指令介绍

分类	指令	内容
单轴	MC_Home	原点回归
	MC_Stop	控制指定轴减速到停止
	MC_MoveAbsolute	控制指定轴依照指定的运动方式移动到用户设定的绝对目标位置
	MC_MoveRelative	控制指定轴依照指定的运动方式移动到用户设定的相对目标位置
	MC_MoveVelocity	控制指定轴在位置模式下依照指定的运动方式及速度做匀速运动
	MC_Power	控制指定轴开启或关闭
	MC_MoveFollow	跟随指令，可用作编码器轴
	MC_SetPosition	改变坐标系统中轴的当前位置。
轴组	MC_GroupEnable	使能轴组

	MC_GroupDisable	失能轴组
	MC_GroupStop	控制指定轴组减速到停止
	MC_GroupAbsLinear	控制轴组执行直线插补，使用绝对位置
	MC_GroupRelLinear	控制轴组执行直线插补，使用相对位置
	MC_GroupAbsCircular	控制轴组执行圆弧插补，使用绝对位置
	MC_GroupRelCircular	控制轴组执行圆弧插补，使用相对位置
扩展	MC_FlyingShearInit	此指令用于追剪功能执行之前的参数初始化工作，必须首先执行该功能块，才能继续执行追剪其他相关的功能块。
	MC_FlyingShear	此命令用于建立追剪关系，可以根据应用需求指定主轴与从轴的轴号。当指令执行时，指令的输出状态能指示出追剪所处区间。
	MC_FlyingShearOut	此指令用于解除主轴与从轴之间已经建立的追剪关系。此指令会让从轴会立即停止，主轴不受影响。
	MC_RotaryCutInit	此指令用于旋切功能执行之前的参数初始化工作，必须首先执行该功能块，才能继续执行旋切其他相关的功能块。
	MC_RotaryCutIn	建立旋切关系，根据实际的应用需求，指定旋切轴和进给轴，当指令执行成功后，旋切轴按照指定的旋切曲线进行运动。
	MC_RotaryCutOut	此指令用于解除旋切轴与进给轴之间已经建立的旋切关系。关系解除后，旋切轴的刀头会停在进入点，且旋切轴不

		再跟随进给轴。此指令不会影响进给轴的运动。
	MC_GearIn	此指令用于建立主从轴间的齿轮关系。建立齿轮关系时，可设定齿轮比等参数。齿轮关系建立后，从轴会以给定的比例关系跟随主轴运动，实现主从轴同步控制。主从轴可以是实轴，也可以是虚轴，或者外部编码器主轴等。
	MC_GearOut	此指令用于解除主从轴间已经建立的齿轮关系。齿轮关系解除后，从轴会以脱离前的速度继续运行。
	MC_GearInPos	此指令用于建立主从轴间的齿轮关系。建立齿轮关系时，可设定齿轮比，主轴从轴的同步位置等参数。齿轮关系建立后，从轴会以给定的比例关系跟随主轴运动，实现主从轴同步控制。主从轴可以是实轴，也可以是虚轴，或者外部编码器主轴等。
	MC_TouchProbe	探针功能，根据触发信号记录轴的位置值
	MC_Phasing	执行同步控制中主轴的相位补偿
	MC_MoveSuperimposed	位移叠加指令
仅适用于 X5PLC	MC_CamTableSelect	此指令用于选择相应的凸轮表
	MC_CamIn	此指令用于执行凸轮表啮合的功能
	MC_CamOut	此指令用于解除主从轴间已经建立的凸轮关系。凸轮关系解除后，从轴会以脱离前的速度继续运行。

	MC_CamWritePoint	此指令用于修改凸轮表关键点。
	MC_GetTappetValue	此指令用于读取挺杆的状态值

轴端子映射

型号	X5S-32、X5-32		X5-60	
轴编号	脉冲端子	方向端子	脉冲端子	方向端子
1	Y0	Y4	Y0	Y6
2	Y1	Y5	Y1	Y7
3	Y2	Y6	Y2	Y10
4	Y3	Y7	Y3	Y11
5	/	/	Y4	Y12
6	/	/	Y5	Y13

特殊寄存器

（1）SM0-SM99,SD0-SD99

SM 偏移	说明	SD 偏移	说明
0-15	**专用滤波器**[*1] 使能位	0-15	专用滤波器设定值，1us 为单位

***1 专用滤波器**

不同于 SD8020 通用滤波器，SD8020 作用于每一个本体输入端子，专用滤波器仅作用于单个对应的端子。当专用滤波器生效时，SD8020 将不再影响该端子；SD0-SD15 为专用滤波器，当 SM0-SM15 置位时生效，最多支持 X0-X17(本体输入点数小于 16 时以本体输入点数为准)，影响普通端子功能、原点回归以及限位开关，目前不会影响高速计数器和中断的滤波；1us 为单位，小于最小滤波时间时以最小滤波时间为准(高速端子最小 1.5us,普通端子最小 400us)。

（2）轴相关的寄存器

轴	对应 SM 范围	对应 SD 范围
Axis1	SM100~SM199	SD100~SD199
Axis2	SM200~SM299	SD200~SD299
Axis3	SM300~SM399	SD300~SD399
Axis4	SM400~SM499	SD400~SD499
Axis5	SM500~SM599	SD500~SD599
Axis6	SM600~SM699	SD600~SD699
Axis7	SM700~SM799	SD700~SD799
Axis8	SM800~SM899	SD800~SD899
Axis9	SM900~SM999	SD900~SD999
Axis10	SM1000~SM1099	SD1000~SM1099
Axis11	SM1100~SM1199	SD1100~SD1199
Axis12	SM1200~SM1299	SD1200~SD1299

（3）SM 寄存器说明

注意：偏移是相对于轴对应的 SM 范围的起始位置而言。

SM 偏移	作用	读写权限	说明
0	轴使能(POWER)指示	R	
1			
2	轴错误标志	R	
3			
4	默认正向限位标志	RW	
5	默认反向限位标志	RW	
6			
7			
8			
9			

20	轴形态指示	R	TRUE 为实轴
21	轴样式指示	R	FALSE 为直线轴，TRUE 为旋转轴
22	加减速模式指示	R	FALSE 为梯形，TRUE 为 S 型
40	轴正向限位取反指示	R	TRUE 为轴正向限位取反
41	轴反向限位取反指示	R	TRUE 轴反向限位取反
42	原点/参考信号取反指示	R	TRUE 为信号取反
43	DOG 信号取反指示	R	TRUE 为信号取反

（4）SD 寄存器说明

注意：偏移是相对于轴对应的 SD 范围的起始位置而言。

SD 偏移	作用	读写权限	数据类型	说明
0				
1	所在轴组 ID	R	WORD(AXIS_REF_AXISGROUPID)	
2	轴错误码	R	WORD	
3				
4	轴实际位置(用户单位)	R	REAL	
5				
6				
7				
8	轴实际速度(用户单位/秒)	R	REAL	
9				
10	轴实际加	R	REAL	

11	速度（用户单位/秒 2）			
20 21	电机转动一圈所需脉冲数	R	DWORD	
22 23	电机转动一圈移动的距离指示	R	REAL	
24 25	轴最高速度指示	R	REAL	
26 27	轴最大加速度指示	R	REAL	
28 29	轴最大减速度指示	R	REAL	
30 31	轴最大加加速度指示	R	REAL	
32 33	旋转轴最大位置指示	R	REAL	
34 35	旋转轴最小位置指示	R	REAL	
36	轴正向限位软元件类型指示	R	WORD	H11：X H13：M H14：S H15：SM
37	轴正向限位软元件下标指示	R	WORD	对应软元件下标，M1000 这里就是 1000
38	轴反向限	R	WORD	H11：X

	位软元件类型指示			H13：M H14：S H15：SM
39	轴反向限位软元件下标指示	R	WORD	对应软元件下标，M1000 这里就是 1000

轴错误说明

根据轴错误寄存器里面的错误码来查看具体错误。

错误码	内容
0	轴或轴组无错误
1	轴或轴组分配错误
1000	轴错误
1001	不允许的轴命令
1002	轴编号错误
1003	轴状态错误
1004	轴命令处理不存在
1005	轴命令最大使用个数错误
1006	轴组合无效
1008	轴速度错误
1009	轴加速度错误
1010	轴减速度错误
1011	轴加加速度错误
1012	轴方向错误
1013	轴功能块调用冲突
1014	轴运动许可错误
1015	轴正限位溢出
1016	轴负限位溢出
1017	轴状态繁忙
1018	轴比例系数分母为零错误

1019	轴位置超差警告
1020	凸轮表 ID 无效
1021	凸轮表内存溢出
1022	凸轮表点数错误
1023	运行速度超过最大设定速度
1024	旋切 ID 错误
1025	旋切轴位置错误
1026	同步长度错误
1027	追剪返回的速度系数设置错误
1028	追剪 ID 无效
1029	从轴位置错误
1030	主轴位置错误
1031	轴梯形加减速错误
1032	轴 S 加减速错误
1033	轴强制结束
1034	开关模式(MC_SWITCH_MODE)无效
1035	原点回归方向无效
1036	对应轴未 Power
1037	无效的 DOG 信号
1038	原点回归失败
1039	位置修改不被允许，只能在静止时修改位置
1040	凸轮表主轴相位错误
1041	探针触发的模式错误
2000	轴组错误
2001	轴组命令不存在
2002	轴组 BufferMode 设置错误
2003	轴组状态错误
2004	轴组命令不允许
2005	轴组功能块调用冲突

2006	轴组速度错误
2007	轴组加速度错误
2008	轴组减速度错误
2009	轴组加加速度错误
2010	轴组方向错误
2011	轴组命令超过最大数错误
2012	轴组圆弧参数错误
2013	轴组繁忙
2014	轴组创建错误
2015	轴组删除错误
2016	轴组梯形加减速错误
2017	轴组 S 加减速错误
2018	轴组未使能
2019	轴组 ID 无效

参数类型说明

注意：运动控制中的参数请不要使用组合位(K)，且不支持常量变址（KxxV(Z)）。

（1）BOOL，WORD，DWORD，REAL，REAL[n]

BOOL 对应位软元件，对于输入参数建议使用 SM8000(True)/SM8001(False),也可以指定 M/S/X/Y/SM8。对于输出参数请使用 M/S/Y/SM8。

WORD 和 DWORD 分别对应单字和双字。

REAL 实际占用内存相当于 DWORD，值为浮点数，需要注意的是运动控制指令的 REAL 类型参数请不要填写奇数地址，如 D1，D3，D5...等，请填写 D0，D2，D4...。

REAL[n]就是连续 n 个 REAL，连续占用 n 个 DWORD。

（2）枚举类型

运动控制指令中除了 BOOL，WORD，DWORD，REAL 以外的类型均为枚举类型，实际占用内存相当于 WORD，枚举的取值在下表中限制。

注意：在上位机上编写指令时只能输入对应的常量。

AXIS_REF_AXISID	X5S/X5-32 最多有 8 个轴，Axis1-Axis4 为实轴,Axis5-Axis8 为虚轴；X5-60 最多有 12 个轴，Axis1-Axis6 为实轴,Axis6-

	Axis12 为虚轴；	
内容	对应常量	作用
Axis1	K1	
Axis2	K2	
...	...	

AXIS_REF_AXISGROUPID	X5S/X5-32 最多有 4 个轴组；X5-60 最多有 6 个轴组	
内容	对应常量	作用
AxisGroup1	K1	
AxisGroup2	K2	
...	...	

MC_DIRECTION	运动方向	
内容	对应常量	作用
mcPositiveDirection	K0	正方向
mcShortestWay	K1	最小位移方向
mcNegativeDirection	K2	负方向
mcCurrentDirection	K3	当前方向

MC_HOME_DIRECTION	原点回归方向	
内容	对应常量	作用
mcHomePositiveDirection	K0	指定原点回归为正向(仅用作旋转轴)
mcHomeNegativeDirection	K1	指定原点回归为反向(仅用作旋转轴)
mcHomeSwitchPositive	K2	指令生效时如果原点信号为 FALSE，则运动方向为 mcPositiveDirection，反之为 mcNegativeDirection
mcHomeSwitchNegative	K3	与 mcHomeSwitchPositive 效果相反

MC_BUFFER_MODE	指令缓冲模式

内容	对应常量	作用
mcBM_Aborting	K0	打断模式
mcBM_Buffered	K1	缓冲模式
mcBM_BlendingPrevious	K3	连续插补模式上一段速度过度
mcBM_BlendingNext	K4	连续插补模式下一段速度过度

MC_SWITCH_MODE	开关模式	
内容	对应常量	作用
mcRisingEdge	K3	上升沿有效
mcFallingEdge	K4	下降沿有效

MC_SOURCE	数据源	
内容	对应常量	来源
mcSetValue	K0	设定值
mcActualValue	K1	实际值

MC_CIRC_MODE	圆弧模式	
内容	对应常量	作用
mcBordef	K0	圆上任意一点
mcCenter	K1	圆心模式
mcRadius	K2	半径模式

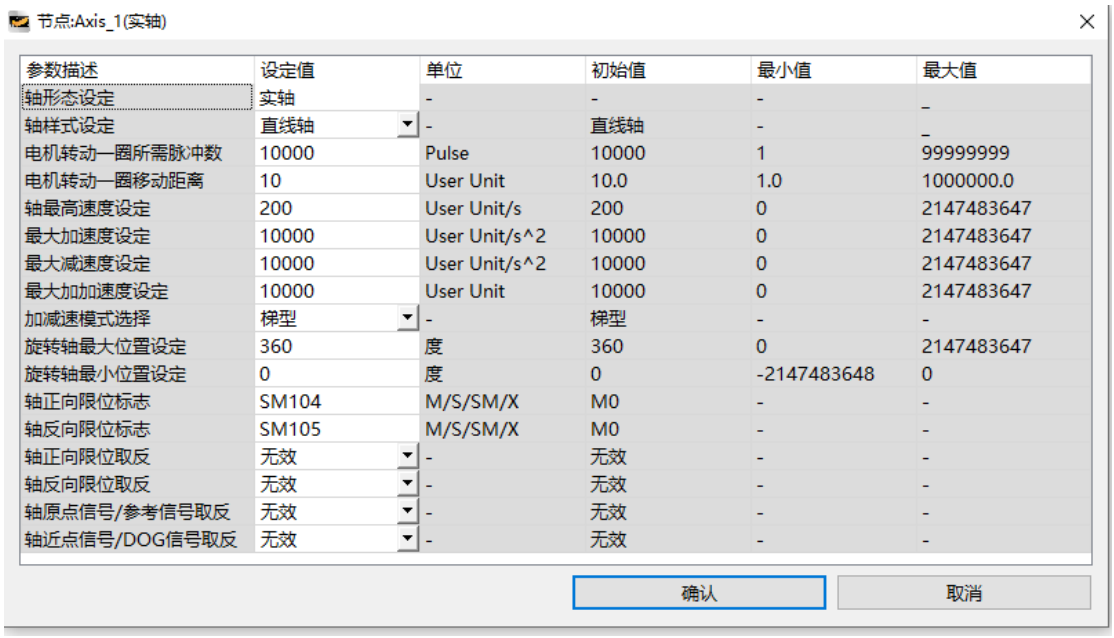
MC_CIRC_PATHCHOICE	顺圆或逆圆	
内容	对应常量	作用
mcCW	K0	顺圆
mcCCW	K1	逆圆

MC_TRANSITION_MODE	插补过度模式	
内容	对应常量	作用

mcTMNone	K0	不使用圆弧过度
mcTMComerSuperimposed	K1	使用圆弧过度

轴参数说明

轴参数采用上位机进行配置下载，配置界面如下图：



默认配置参数数入下表：

参数描述	参数范围	默认值（N 为轴编号 1-8）	单位
轴形态设定	实轴/虚轴	*1	/
轴样式设定	直线轴/旋转轴	直线轴	/
电机转一圈所需脉冲数	1-99999999	10000	Pulse
电机转一圈移动距离	1.0-1000000.0	10.0	User Unit
轴最高速度设定	0-2147483647	200	User Unit/s
最大加速度设定	0-2147483647	10000	User Unit/s^2
最大减速度设定	0-2147483647	10000	User Unit/s^2
最大加加速度设定	0-2147483647	10000	User Unit/s^3
加速模式选择	梯形/S 型	梯形	/
旋转轴最大位置	2147483647	360	度
旋转轴最小位置	0	0	度
轴正向限位标志	M/S/SM/X	SMN04,N 用轴号代替	/
轴反向限位标志	M/S/SM/X	SMN05,N 用轴号代替	/
轴正向限位取反	FALSE/TRUE	FALSE	/
轴反向限位取反	FALSE/TRUE	FALSE	/
轴原点信号/参考信号取反	FALSE/TRUE	FALSE	/

轴近点信号/DOG 信号取反	FALSE/TRUE	FALSE	/

注：

*1：

X5S/X5-32TD 最多支持 8 个轴，其中 4 个实轴，4 个虚轴。
X5-60TD 最多支持 12 个轴，其中 6 个实轴，6 个虚轴。

实轴：可以通过端子发脉冲

虚轴：不具有发脉冲功能，但是可以在寄存器中观察相关状态。常用于跟随/凸轮/电子齿轮等相关功能中。

直线轴：位置无边界（实际取值为单精度浮点数范围）

旋转轴：位置成环形，旋转轴最大位置和旋转轴最小位置重合

轴最高速度设定：实轴速度受最大脉冲频率限制，需要满足如下条件：
0 < 实轴最高速度 <= 200K /（电机转动一圈所需脉冲数/电机转一圈的位移）
200K 是 UX5S/X5 输出端子允许的最大脉冲频率

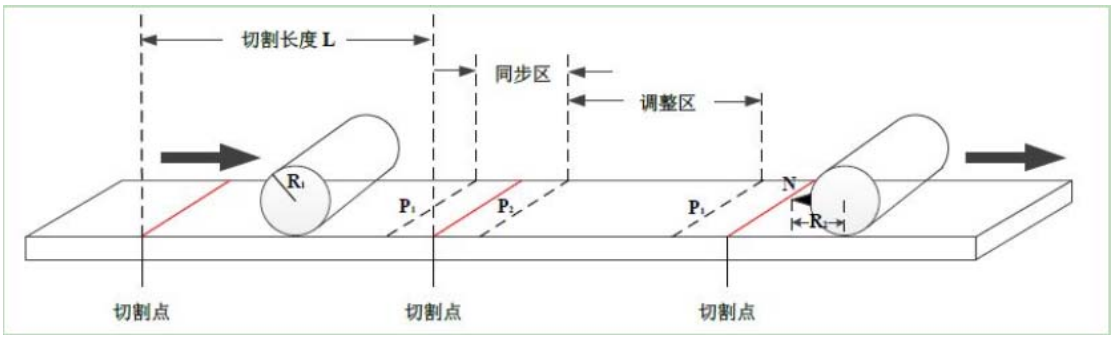
最大加加速度设定：加速模式设为 S 型时有效

信号取反：正向限位信号/反向限位信号/参考信号(原点信号)/DOG 信号支持信号取反，即将有效信号从高电平改为低电平。需要注意的是在原点回归中，上升沿触发将由对应输入信号从 0 变化 1 改为对应输入信号从 1 变化为 0

旋切工艺

旋切是指对进给轴上传输的材料进行垂直方向的切割工艺，旋切轴根据剪切材料的长度变化，动态地调整自己运行的速度，保证在切割时刀头和被切割材料的线速度一致。

旋切功能的参数说明如下图所示：



图中参数	实际含义	对应指令参数
L	材料的剪切长度	CutLength

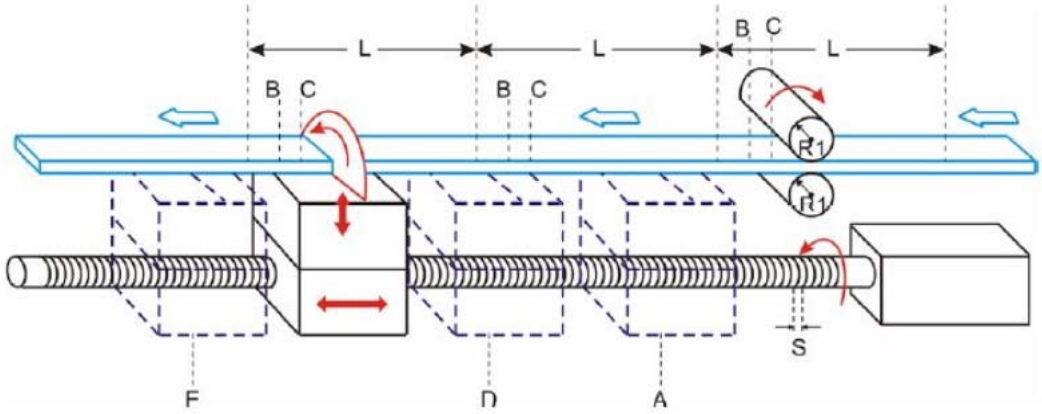
R1	进给轴半径	FeedAxisRadius
R2	旋切轴半径	RotaryAxisRadius
N	旋切轴上切刀的数量	KnifeNum
P1	同步区的起始位置	SyncStartPos
P2	同步区的结束位置	SyncStopPos

注意：指令参数可在实际使用的指令当中找到，并在运动控制参数说明表中找到可以填写的数据值。

追剪工艺

追剪是对运动中的材料进行垂直切割的工艺。从轴由等待位置开始加速，当速度达到与主轴速度一样的时候，丝杠从动件与材料的运动速度相等，两者相对静止，同步位置位，触发刀轴控制切刀向上切割。

追剪工艺功能的参数说明如下图所示：



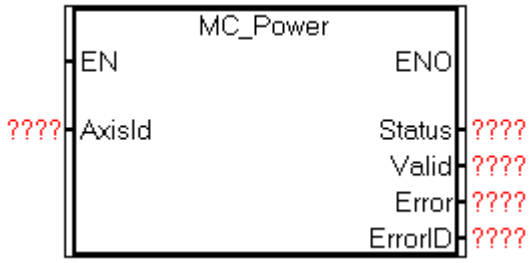
相关参数	实际含义	对应指令参数
R1	主轴半径	无，在轴配置中设置周期
S	从轴丝杆导程	无，在轴配置中设置导程
A	从轴的等待位置 当追剪功能启动后，从站自动运行到此为止	SlaveWaitPos
B	主轴开始位置 当主轴到达此位置时，从轴由等待位置追赶主轴，以实现速度同步	MasterStartPos

C	主轴同步开始位置 同步区开始的时候,对应的主轴位置	MasterSyncPosi
D	从轴同步开始位置 同步区开始的时候,对应的从轴位置	SlaveSyncPos
E	从轴同步结束位置 同步区结束的时候,对应的从轴位置	SlaveEndPos
L	剪切长度 材料的切割长度	CutLength

注意：指令参数可在实际使用的指令当中找到，并在运动控制参数说明表中找到可以填写的数据值。

2.19.1 MC_Power/启用/禁用轴指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制指定轴开启或关闭</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisId</td><td>轴 ID</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>输出参数</td><td></td><td></td></tr>
<tr><td>Status</td><td>当运动轴启动后为 True</td><td>BOOL</td></tr>
<tr><td>Valid</td><td>当指令生效时为 True</td><td>BOOL</td></tr>
<tr><td>Error</td><td>当指令错误发生时为 True</td><td>BOOL</td></tr>
<tr><td>ErrorID</td><td>当指令错误发生时纪录错误码</td><td>WORD</td></tr>
</table>

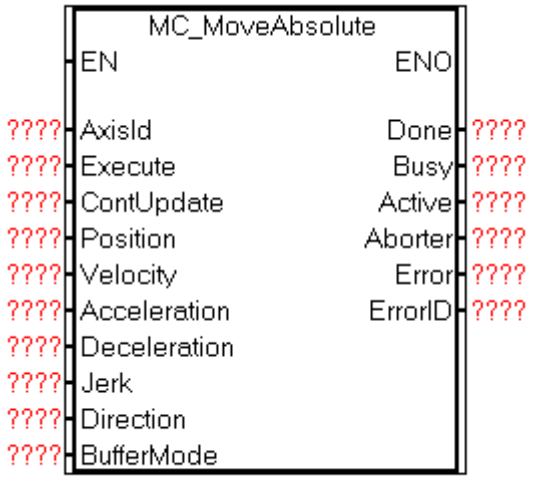
2、功能描述

MC_Power 指令只在 EN 上升沿和下降沿动作，如下图，打开 M1000 时轴 1 被使能，关闭 M1000 时轴 1 失效。



2.19.2 MC_MoveAbsolute/绝对定位指令

1、指令详解



控制指定轴依照指定的运动方式移动到用户设定的绝对目标位置

输入参数	解释	参数类型
AxisId	轴 ID	AXIS_REF_AXISID
Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
ContUpdate	为 True 时，持续更新目标速度	BOOL
Position	绝对位置（用户单位）	REAL
Velocity	目标速度（用户单位/秒）	REAL
Acceleration	加速度（用户单位/秒 2）	REAL
Deceleration	减速度（用户单位/秒 2）	REAL
Jerk	加加速度；跃度（用户单位/秒 3）	REAL

Direction	指定运动轴伺服电机运转方向。在使用旋转轴时有效	MC_DIRECTION K0 正方向 K1 最小位移方向 K2 负方向 K3 当前方向
BufferMode	指定此功能块指令的缓冲行为模式	MC_BUFFER_MODE K0 打断模式 K1 缓冲模式
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisId 的值。

在 EN 的上升沿锁存 BufferMode。

当 EN 复位时输出参数都会被复位。

在 Execute 输入的上升沿，指令锁存 Position、Velocity，Acceleration，Deceleration，Jerk，Direction 并触发绝对定位功能。

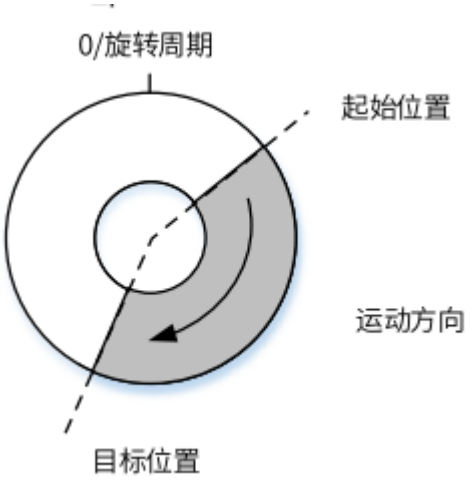
在 ContUpdate 为 True 时持续更新 Position、Velocity，Acceleration，Deceleration，Jerk，Direction。

线性模式下，Position 用于设定定位的目标位置。如果当前位置小于目标位置，轴将正向运动，
最后到达 Position 设定的位置，如果当前位置大于目标位置，轴将反向运动，最后到达 Position 设定的位置。

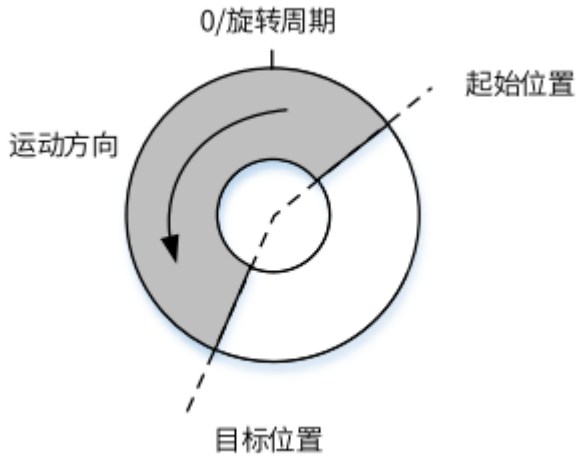
在上位机的轴配置中设置加减速类型为 T 形或 S 形，运动过程中轴的速度将按照 Acceleration 和 Deceleration 设定的值做加速或减速运动。

在环形模式下，指令首先将 Position 对旋转周期求模，得到一个旋转周期内的绝对位置 Position_p,轴实际的运动方向由以下 4 中情况决定：

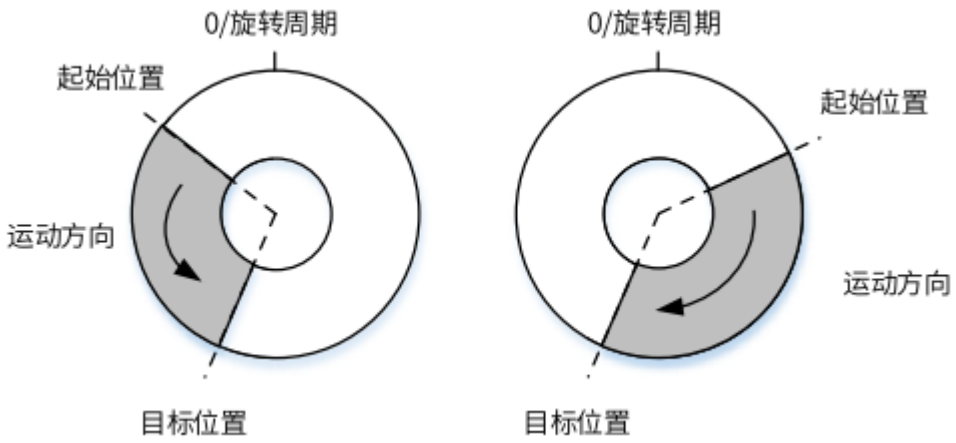
（1） Direction = 0 表示正向运动（目标速度大于 0）。如果当前速度大于 0，则继续朝当前方向运行，然后到 Position_p 设定的位置停止；如果当前速度小于 0，则先做减速运动，减速到 0 后开始速度反向运动，直到 Position_p 设定的位置停止；如果 Position_p 和当前位置相等，则不运动。



（2）Direction = 2 表示反向运动（目标速度小于 0）。如果当前速度小于 0，则继续朝当前方向运行，然后到 Position_p 设定的位置停止；如果当前速度大于 0，则先做减速运动，减速到 0 后开始速度反向运动，直到 Position_p 设定的位置停止；如果 Position_p 和当前位置相等，则不运动



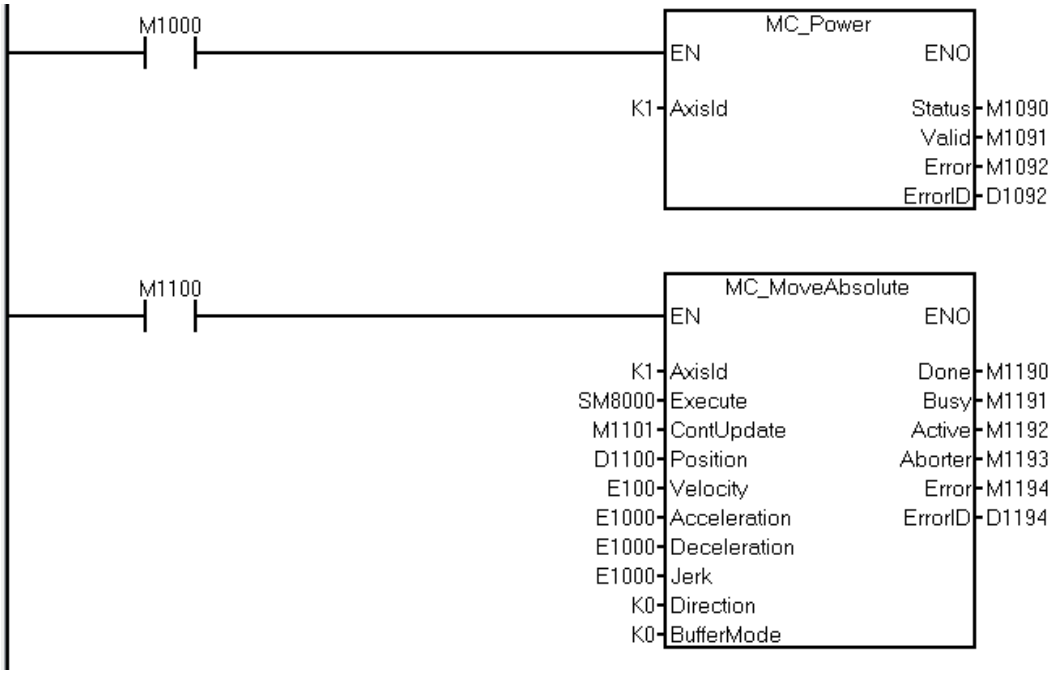
（3）Direction = 1 表示最短距离，在 Execute 的上升沿，轴将记录当前位置，并假设当前速度是 0，推算出从 0 速启动轴正向运动到 Position_p 的位移 Distance，如果 Distance 小于等于 0.5*旋转一周位移，则正向运动，如果大于 0.5*旋转一周位移则反向运动；如果 Position_p 和当前位置相等，则不运动。



（4）Direction = 3 表示沿当前方向，在 Execute 的上升沿，轴将按照 Execute 上升沿到达之前最近的一次运动方向运动，直到 Position_p 设定的位置停止，如果是第一次上电，则正向运动（目标速度大于 0）。如果 Position_p 和当前位置相等，则不运动。

3、程序举例

从位置 0，到目标位置 100



梯形图执行前初始状态如下:（其中 D1100 为目标点位置，SD104 为轴 1 当前实际位置）

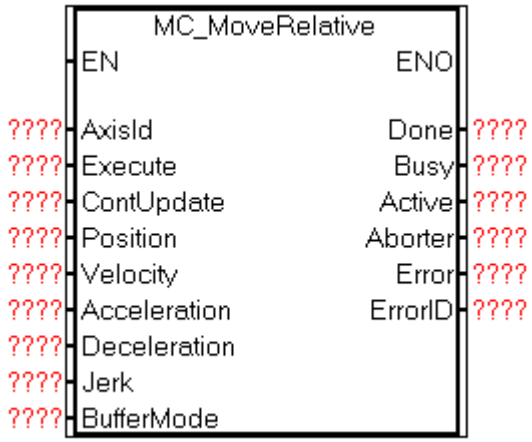
软元件/标签	数据类型	当前值
D1100	FLOAT	100.0
M1000	BIT	0
M1100	BIT	0
SD104	FLOAT	0.0
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0

依次打开 M1000 和 M110 后变化如下：

软元件/标签	数据类型	当前值
D1100	FLOAT	100.0
M1000	BIT	1
M1100	BIT	1
SD104	FLOAT	100.0
M1190	BIT	1
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0

2.19.3 MC_MoveRelative/相对定位指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制指定轴依照指定的运动方式移动到用户设定的相对目标位置</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisId</td><td>轴 ID</td><td>AXIS_REF_AXISID</td></tr>
</table>

Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
ContUpdate	为 True 时，持续更新目标速度	BOOL
Position	相对位置（用户单位）	REAL
Velocity	目标速度（用户单位/秒）	REAL
Acceleration	加速度（用户单位/秒 2）	REAL
Deceleration	减速度（用户单位/秒 2）	REAL
Jerk	加加速度；跃度 （用户单位/秒 3）	REAL
BufferMode	指定此功能块指令的缓冲行为模式	MC_BUFFER_MODE K0 打断模式 K1 缓冲模式
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisId 的值。

在 EN 的上升沿锁存 BufferMode。

当 EN 复位时输出参数都会被复位。

在 Execute 输入的上升沿，指令锁存 Position、Velocity，Acceleration，Deceleration，Jerk 并触发相对定位功能。

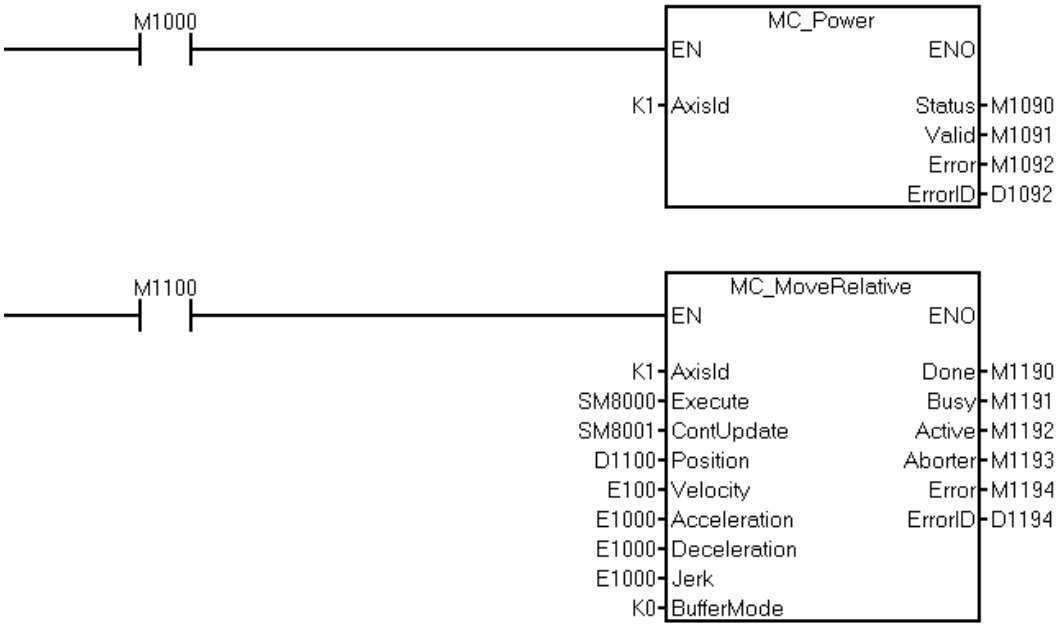
在 ContUpdate 为 True 时持续更新 Position、Velocity，Acceleration，Deceleration，Jerk。

Position 用于设定相对定位的距离。不管在线性模式还是环形模式下，如果 Distance 为正数，轴正向运行 Distance 指定的距离，如果 Distance 为负，则轴负向运行|Distance|指定的距离。

在上位机的轴配置中设置加减速类型为 T 形或 S 形，运动过程中轴的速度将按照 Acceleration 和 Deceleration 设定的值做加速或减速运动。

3、程序举例

从当前位置 0 往正方向走 100 个单位



相关寄存器初始值如下（其中 D1100 为设定位移，SD104 为轴 1 当前实际位置）

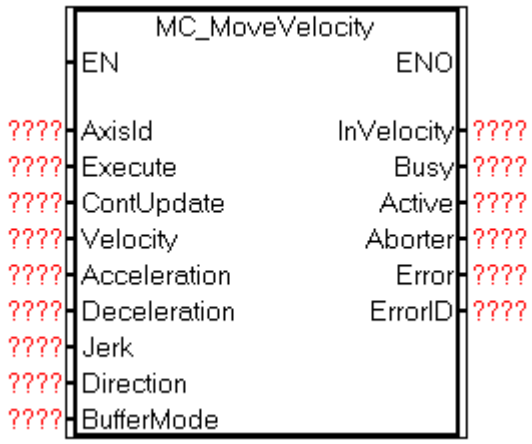
软元件/标签	数据类型	当前值
D1100	FLOAT	100.0
M1000	BIT	0
M1100	BIT	0
SD104	FLOAT	200.0
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0

打开 M1000 和 M1100 后变化如下

软元件/标签	数据类型	当前值
D1100	FLOAT	100.0
M1000	BIT	1
M1100	BIT	1
SD104	FLOAT	300.0
M1190	BIT	1
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0

2.19.4 MC_MoveVelocity/速度控制指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制指定轴在位置模式下依照指定的运动方式及速度做均速运动</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisId</td><td>轴 ID</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Execute</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>ContUpdate</td><td>为 True 时，持续更新目标速度</td><td>BOOL</td></tr>
<tr><td>Velocity</td><td>目标速度（用户单位/秒）</td><td>REAL</td></tr>
<tr><td>Acceleration</td><td>加速度（用户单位/秒 2）</td><td>REAL</td></tr>
<tr><td>Deceleration</td><td>减速度（用户单位/秒 2）</td><td>REAL</td></tr>
<tr><td>Jerk</td><td>加加速度；跃度
（用户单位/秒 3）</td><td>REAL</td></tr>
<tr><td>Direction</td><td>指定运动轴伺服电机运转方向</td><td>MC_DIRECTION
K0 正方向</td></tr>
</table>

		K2 负方向 K3 当前方向
BufferMode	指定此功能块指令的缓冲行为模式	MC_BUFFER_MODE K0 打断模式 K1 缓冲模式
输出参数		
InVelocity	当指定速度达到目标速度时为 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令

在 EN 后请不要改变 AxisId 的值

在 EN 的上升沿锁存 BufferMode

当 EN 复位时输出参数都会被复位。

在 Execute 输入的上升沿，指令锁存 Velocity，Acceleration，Deceleration，Jerk，Direction 并触发受控运动

在 ContUpdate 为 True 时持续更新 Velocity，Acceleration，Deceleration，Direction，Jerk。允许在运行中修改 ContUpdate

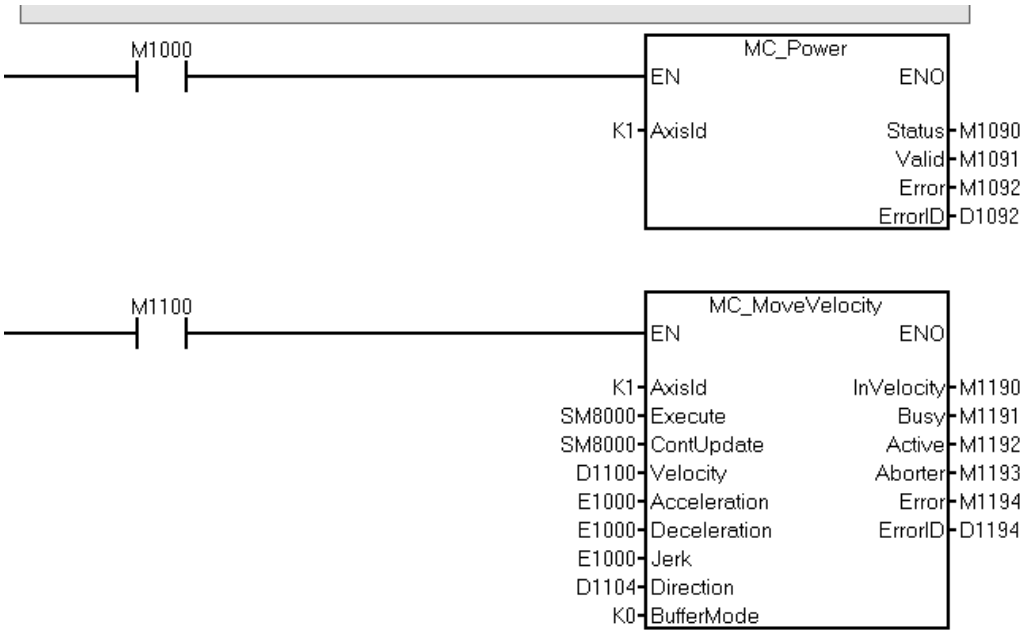
本指令执行中其他指令启动说明

对其他指令的 BufferMode 指定为中断模式时，会立刻结束该指令动作其它指令。

对其他指令的 BufferMode 指定为缓存模式时，如果本指令的输出“InVelocity(达到目标速度)”变为 TRUE，则其他指令开始动作。

3、程序举例

以固定速度 100 往正方向运动



相关寄存器初始值如下（其中 D1100 为设定速度，SD104 为轴 1 当前实际位置，SD108 为轴 1 当前实际速度）

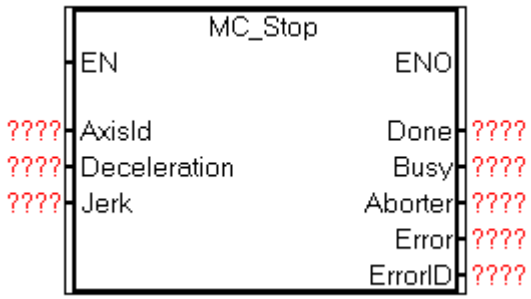
软元件/标签	数据类型	当前值
D1100	FLOAT	100.0
D1104	WORD	0
M1000	BIT	0
M1100	BIT	0
SD104	FLOAT	29589.199
SD108	FLOAT	0.0
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0

打开 M1000 和 M1100 后变化如下：

软元件/标签	数据类型	当前值
D1100	FLOAT	100.0
D1104	WORD	0
M1000	BIT	1
M1100	BIT	1
SD104	FLOAT	29995.1
SD108	FLOAT	100.0
M1190	BIT	1
M1191	BIT	1
M1192	BIT	1
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0

2.19.5 MC_Stop/停止指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制指定轴减速到停止</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisId</td><td>轴 ID</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Deceleration</td><td>减速度（用户单位/秒 2）</td><td>REAL</td></tr>
<tr><td>Jerk</td><td>加加速度；跃度
（用户单位/秒 3）</td><td>REAL</td></tr>
<tr><td>输出参数</td><td></td><td></td></tr>
<tr><td>Done</td><td>动作完成时置 True</td><td>BOOL</td></tr>
<tr><td>Busy</td><td>当指令被触发执行时为 True</td><td>BOOL</td></tr>
<tr><td>Aborted</td><td>当指令被中断时为 True</td><td>BOOL</td></tr>
<tr><td>Error</td><td>当指令错误发生时为 True</td><td>BOOL</td></tr>
<tr><td>ErrorID</td><td>当指令错误发生时记录错误码</td><td>WORD</td></tr>
</table>

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令

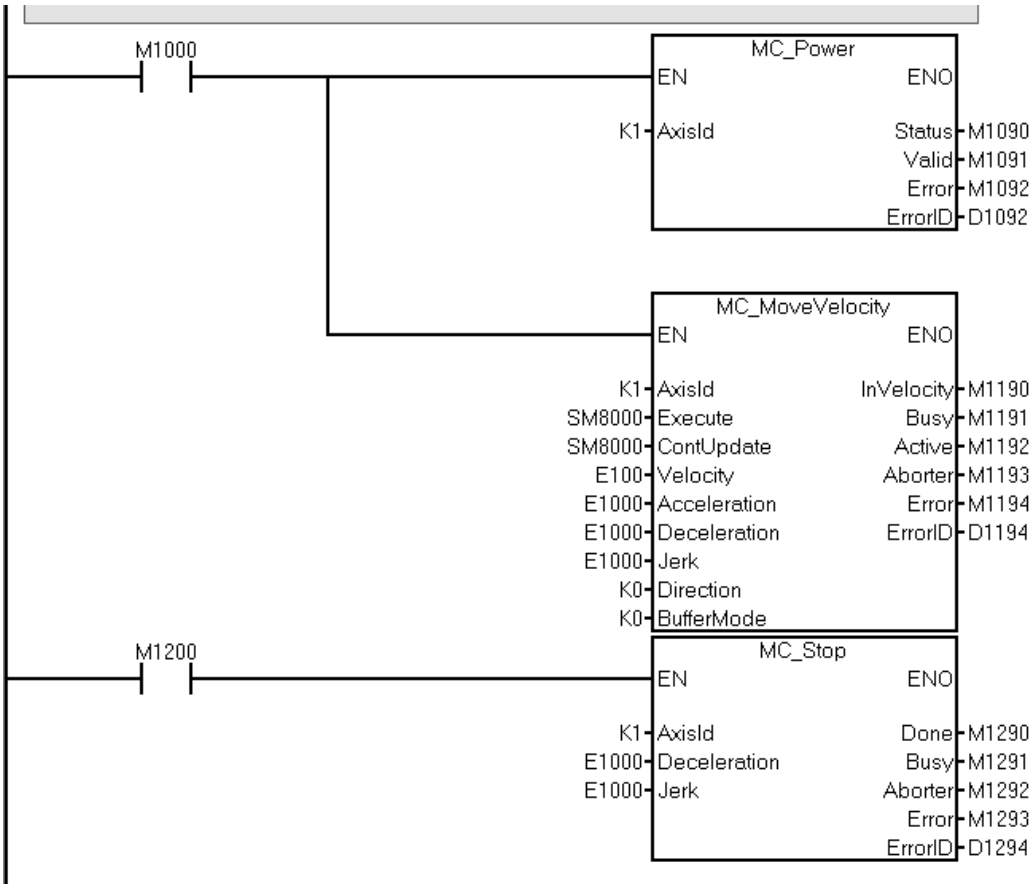
在 EN 后请不要改变 AxisId 的值

当 EN 复位时输出参数都会被复位。

MC_Stop 会结束正在运行的指令，该指令默认采用打断模式，会清空该轴的整个任务队列。

3、程序举例

使用 MC_Stop 指令停止 MC_MoveVelocity 指令



打开 M1000 后相关寄存器如下（启动 SD108 是轴 1 当前实际速度）

软元件/标签	数据类型	当前值
M1000	BIT	1
SD104	FLOAT	699.4
SD108	FLOAT	100.0
M1190	BIT	1
M1191	BIT	1
M1192	BIT	1
M1193	BIT	0
D1194	WORD	0
M1290	BIT	0
M1291	BIT	0
M1292	BIT	0
M1293	BIT	0
D1294	WORD	0

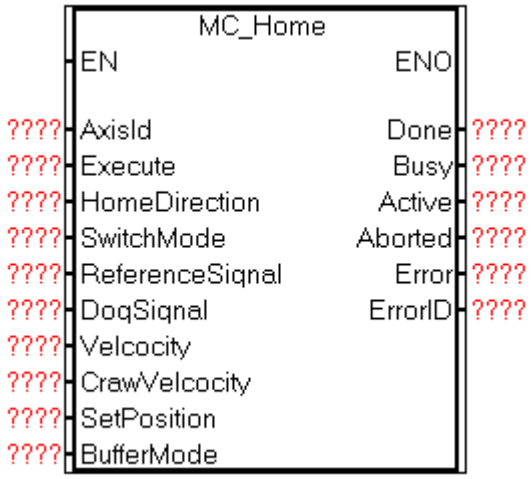
打开 M1200 后：

软元件/标签	数据类型	当前值
M1000	BIT	1
SD104	FLOAT	8653.2
SD108	FLOAT	0.0
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	1
D1194	WORD	0
M1290	BIT	1
M1291	BIT	0
M1292	BIT	0
M1293	BIT	0
D1294	WORD	0
M1200	BIT	1

速度指令的停止标志 M1193 置位，且当前速度变为 0。

2.19.6 MC_Home/原点返回指令

1、指令详解

		原点回归
输入参数	解释	参数类型
AxisId	轴 ID	AXIS_REF_AXISID
Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
HomeDirection	原点回归方向	MC_HOME_DIRECTION K0 指定原点回归为正向(仅用作旋转轴) K1 指定原点回归为反向(仅用作旋转轴) K2 指令生效时如果原点信号为 FALSE，则运动方向为正方向，反之为负方向 K3 与 K2 效果相反
SwitchMode	原点信号的触发方式	MC_SWITCH_MODE K3 上升沿有效 K4 下降沿有效
ReferenceSignal	原点信号（由 SwitchMode 控制触发方式）	BOOL
DogSignal	Dog 信号(遇到时为 True)	BOOL
Velocity	回归速度	REAL
CrawVelocity	遇到 Dog 信号后的爬行速度	REAL

SetPosition	回归完成后的位置设定	REAL
BufferMode	指定功能块的缓冲行为	MC_BUFFER_MODE K0 打断模式 K1 缓冲模式
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令。

在 EN 后请不要更改右侧输入参数的值。

原点会归会在 Execute 的上升沿执行。

当 EN 复位时输出参数都会被复位。

HomeDirection 指定原点回归的方向:

HomeDirection 为 mcHomePositiveDirection(K0) 或 mcHomeNegativeDirection（K1）仅能在旋转轴上使用，且固定一个方向运动。

HomeDirection 为 mcHomeSwitchPositive(K2)时，若启动时原点信号为 False，则往正方向运动；若启动时原点信号为 True 则往负方向运动。

HomeDirection 为 mcHomeSwitchNegative(K3)时，若启动时原点信号为 False，则往负方向运动；若启动时原点信号为 True 则往正方向运动。

SwitchMode 指定原点信号的触发方式，为 mcRisingEdge(K3)时为上升沿(ReferenceSiqnal 由 False 变为 True)触发，为 mcFallingEdge(K4)时为下降沿(ReferenceSiqnal 由 True 变为 False)触发。

ReferenceSiqnal 指定原点信号(零点信号)。

DogSignal 指定近点信号(DOG 信号)，当运动中遇到 DOG 信号（DogSignal 为 True）且运动方向与 HomeDirection 相同时 Dog 信号生效，此时改变速度为 CrawVelocity。DogSignale 可以省略，此时请指定与 ReferenceSiqnal 相同的软元件。

Velocity 指定原点回归的速度，原点回归的加减速度固定为为上位机配置的轴最大加速度

和轴最大减速度。当没有 Dog 信号时，请适当将 Velocity 调小。

CrawVelocity 为 Dog 信号生效后的爬行速度。

SetPosition 用作原点回归结束修改当前位置为 SetPosition。

原点回归过程中一旦遇到限位开关，就会自动改变运动方向往相反的方向运动。

原点回归结束的条件

1）直线轴有 DOG 信号时原点回归结束条件如下:

①DOG 生效，即经过 DOG 且运动方向与 HomeDirection 方向相同;

②遇到原点信号的有效边沿(SwitchMode 指定)且运动方向与 HomeDirection 方向相同;

2）直线轴无 DOG 信号(ReferenceSiqnal 和 DogSignal 指定相同软元件)是原点回归结束条件如下:

①遇到原点信号的有效边沿(SwitchMode 指定)且运动方向与 HomeDirection 方向相同;

3）旋转轴原点回归结束条件如下:

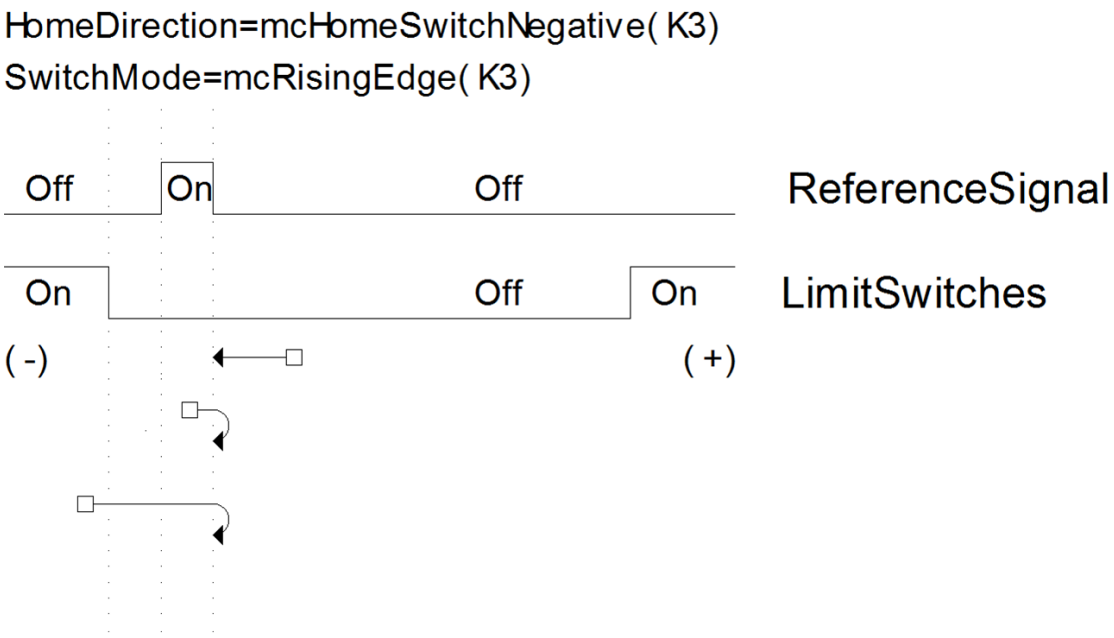
①运动方向与 HomeDirection 方向相同且遇到原点信号的有效边沿(SwitchMode 指定);

原点回归轨迹示意图

1. 直线轴原点回归轨迹

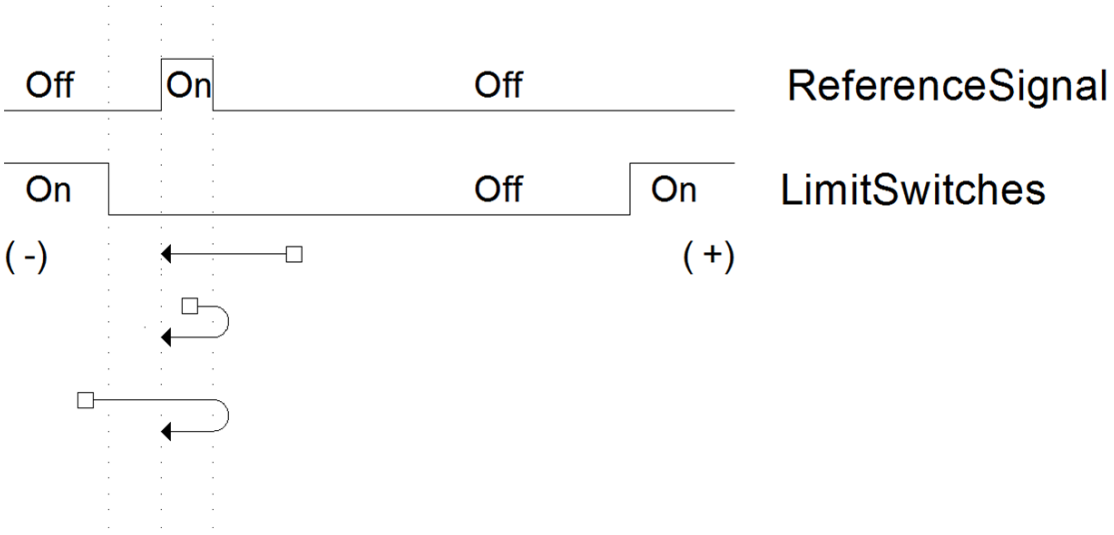
1）不带 DOG 信号，HomeDirection=mcHomeSwitchNegative(K3)

①SwitchMode=mcRisingEdge(K3)



②SwitchMode=mcFallingEdge(K4)

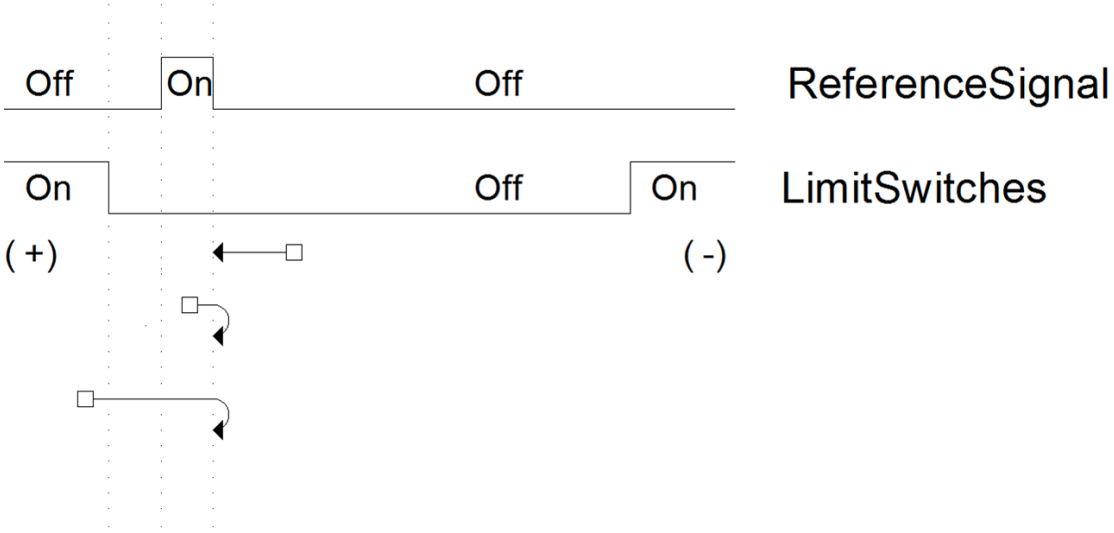
HomeDirection=mcHomeSwitchNegative(K3)
SwitchMode=mcFallingEdge(K4)



2）不带 DOG 信号，HomeDirection=mcHomeSwitchPositive(K2)

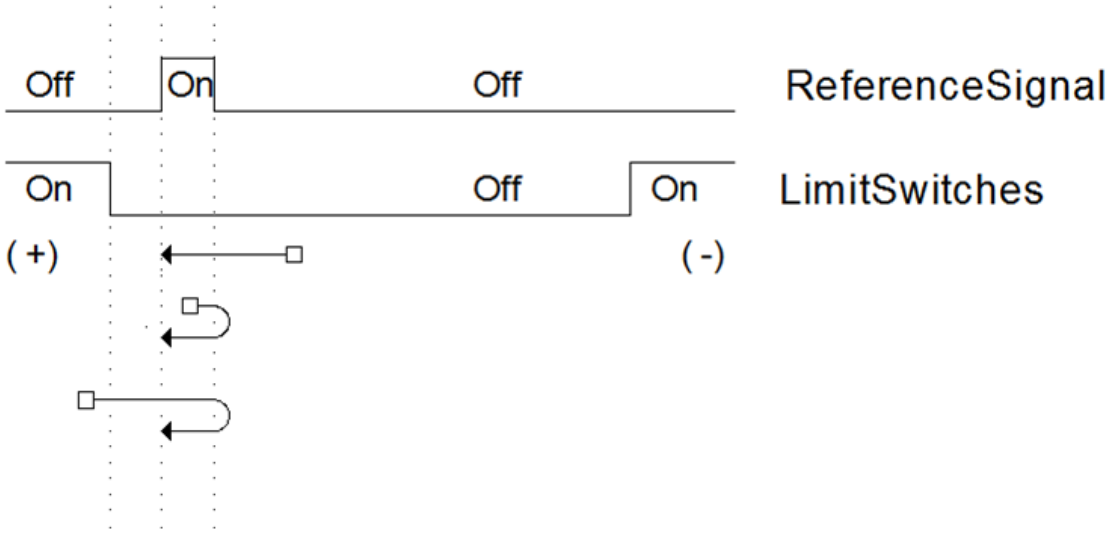
①SwitchMode=mcRisingEdge(K3)

HomeDirection=mcHomeSwitchPositive(K2)
SwitchMode=mcRisingEdge(K3)



②SwitchMode=mcFallingEdge(K4)

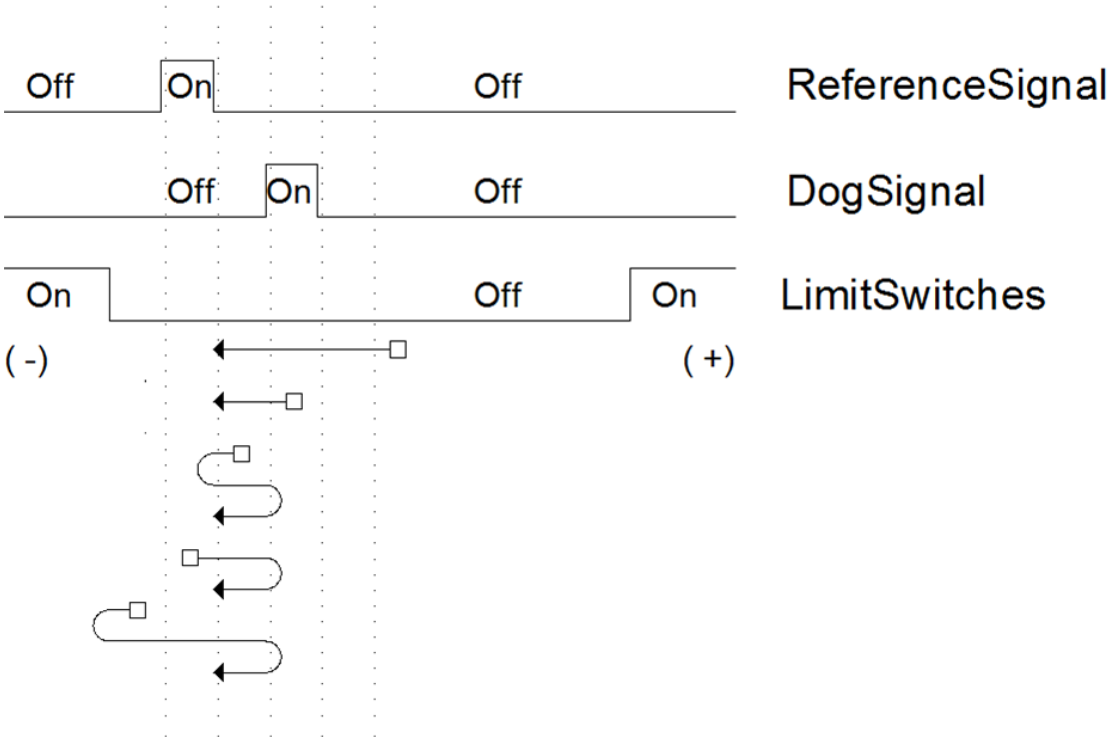
HomeDirection=mcHomeSwitchPositive(K2)
SwitchMode=mcFallingEdge(K4)



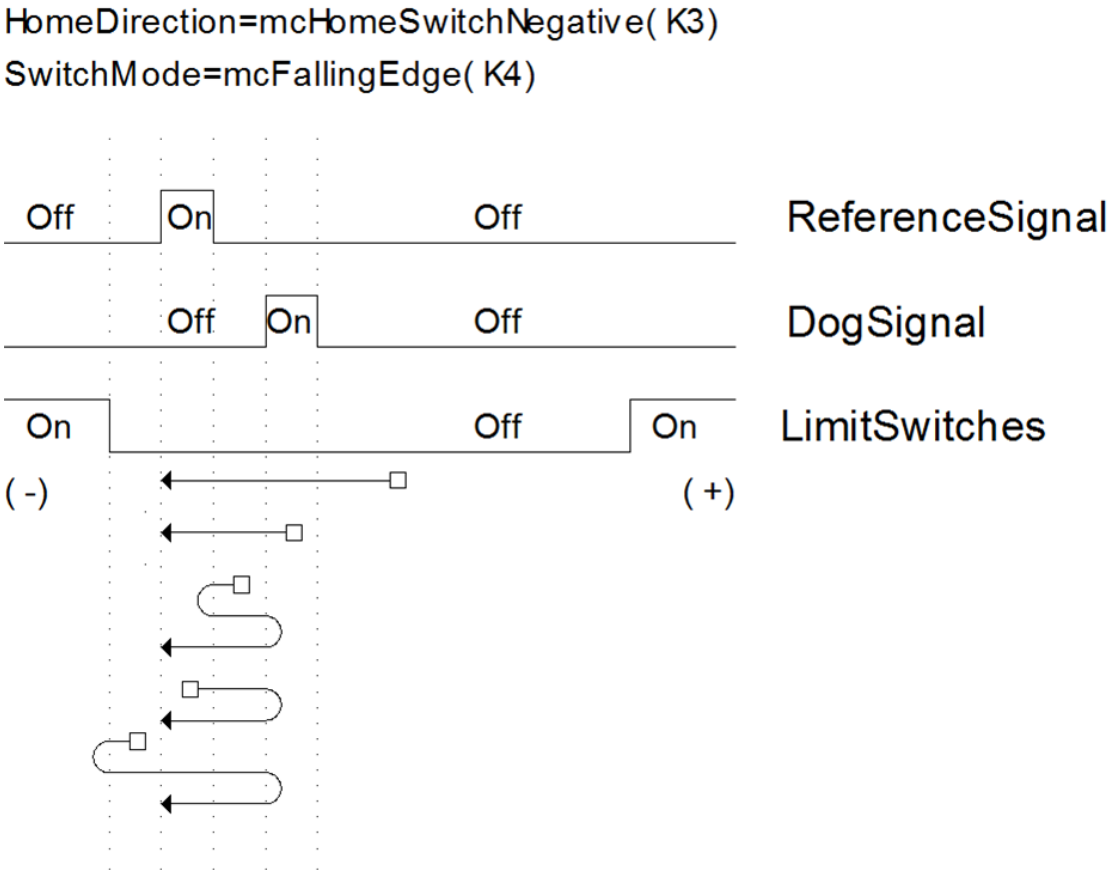
3）带 DOG 信号，HomeDirection=mcHomeSwitchNegative(K3)

①SwitchMode=mcRisingEdge(K3)

HomeDirection=mcHomeSwitchNegative(K3)
SwitchMode=mcRisingEdge(K3)



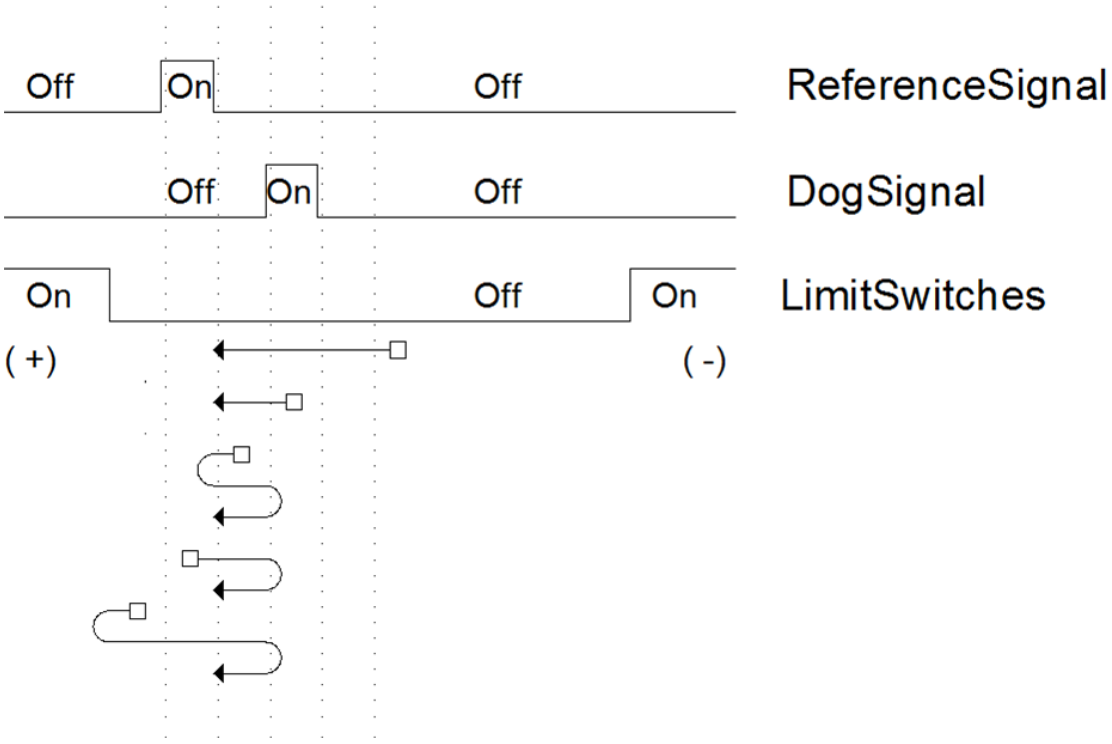
②SwitchMode=mcFallingEdge(K4)



4）带 DOG 信号，HomeDirection=mcHomeSwitchPositive(K2)

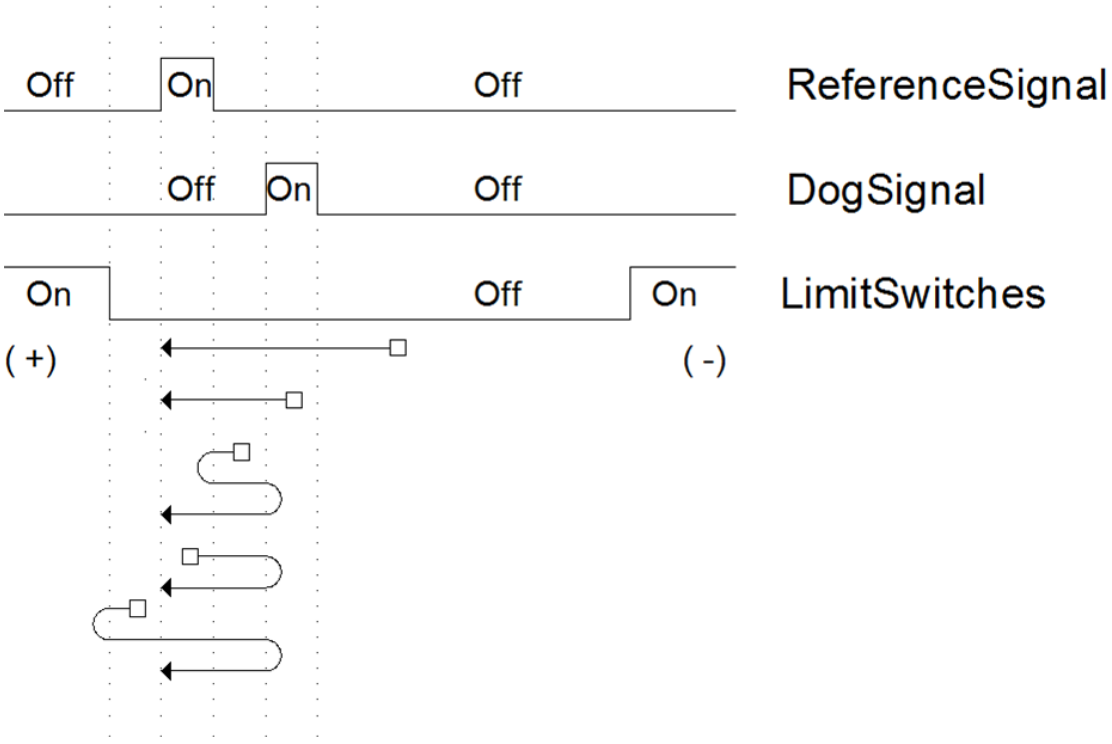
①SwitchMode=mcRisingEdge(K3)

HomeDirection=mcHomeSwitchPositive(K2)
SwitchMode=mcRisingEdge(K3)

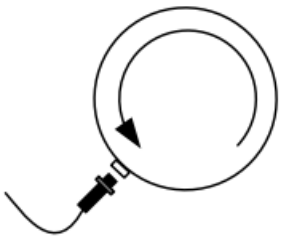


②SwitchMode=mcFallingEdge(K4)

HomeDirection=mcHomeSwitchPositive(K2)
SwitchMode=mcFallingEdge(K4)

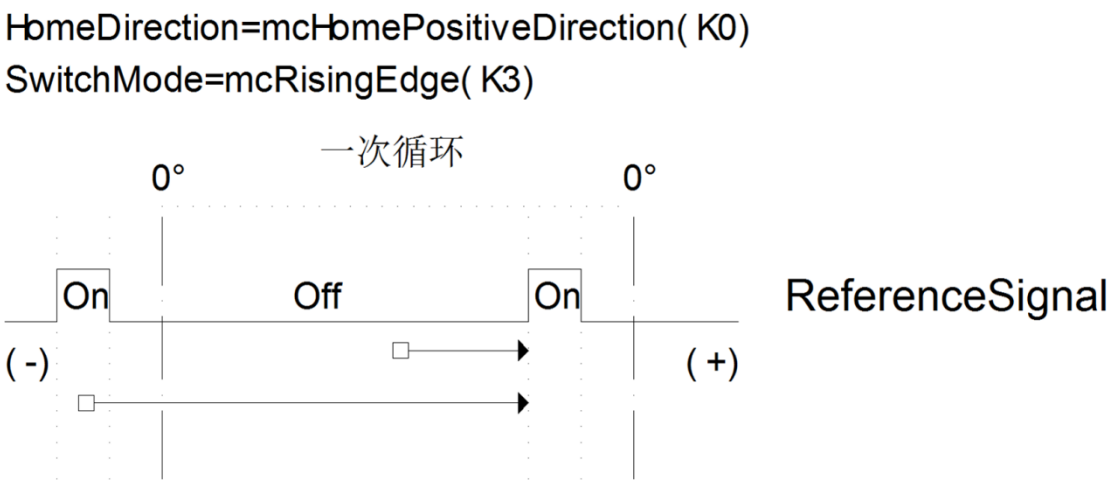


2. 旋转轴原点回归轨迹

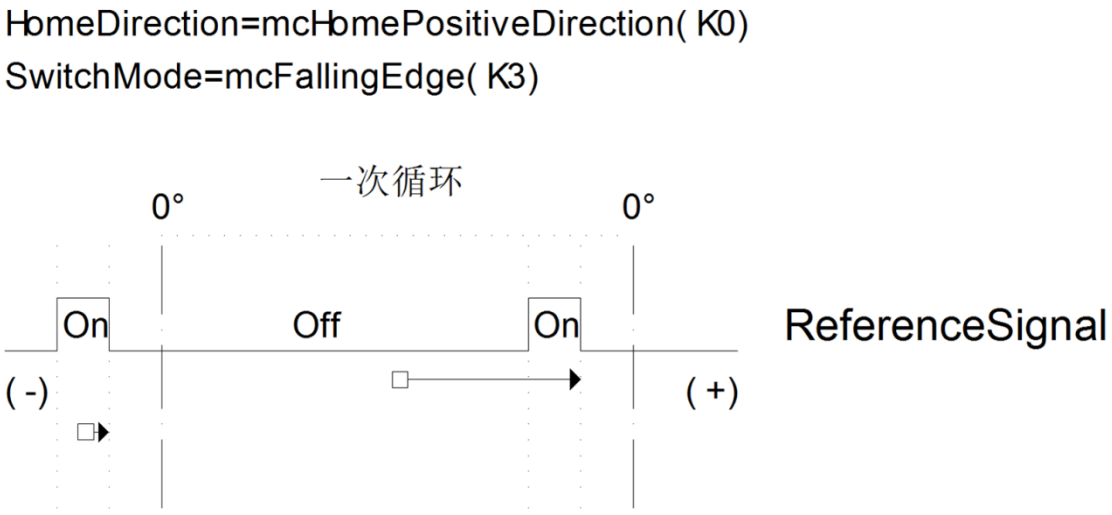


1）HomeDirection=mcHomePositiveDirection(K0)

①SwitchMode=mcRisingEdge(K3)



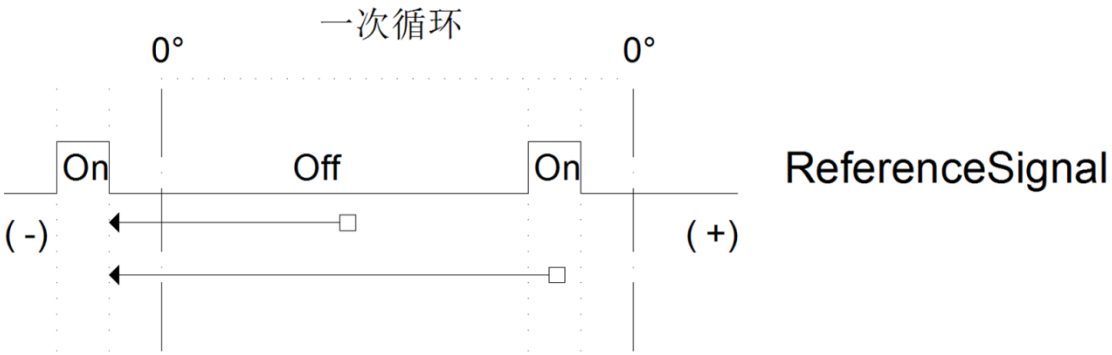
②SwitchMode=mcFallingEdge(K4)



2）HomeDirection=mcHomeNegativeDirection(K1)

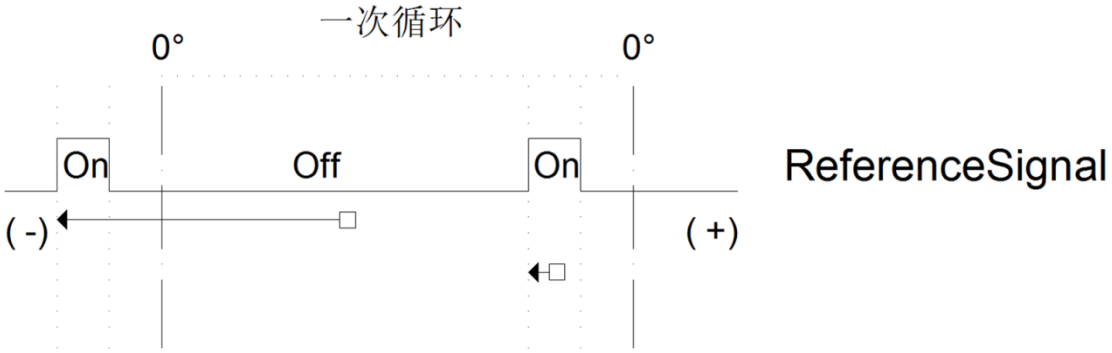
①SwitchMode=mcRisingEdge(K3)

HomeDirection=mcHomeNegativeDirection(K1)
SwitchMode=mcRisingEdge(K3)



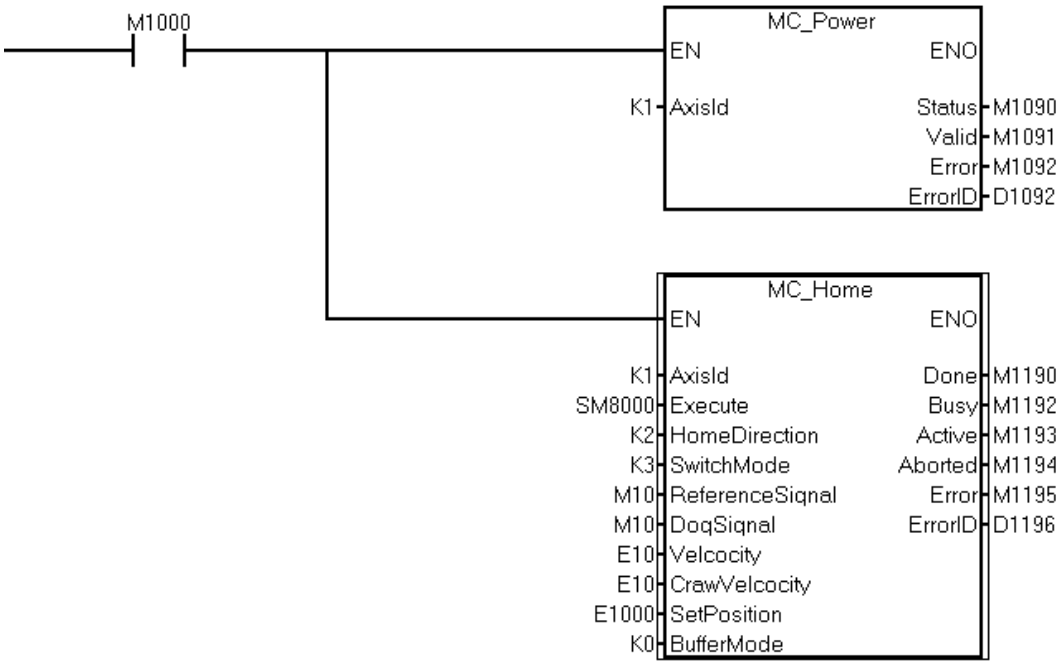
②SwitchMode=mcFallingEdge(K4)

HomeDirection=mcHomeNegativeDirection(K1)
SwitchMode=mcFallingEdge(K4)



3、程序举例

无 DOG 信号的快速原点回归，检测原点信号下降沿



HomeDirection=2 代表指令生效时如果原点信号为 FALSE，则运动方向为 mcPositiveDirection，反之为 mcNegativeDirection；

SwitchMode=3 代表原点信号下降沿有效(即碰到原点信号的下降沿且当前运动方向为正方向完成回归)；

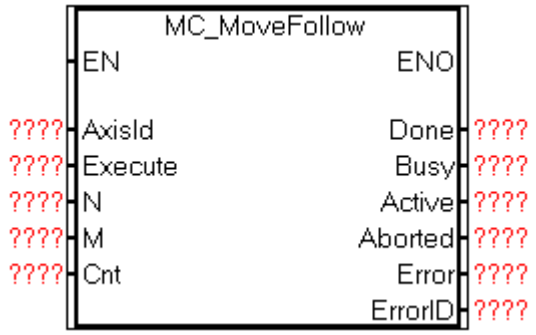
RefenenceSignal 指示原点信号（原点），这里使用 M10 手动控制；

DogSignal 指示近点信号(DOG)，遇到 DOG 信号后会使用爬行速度（CrawVelcocity）运动，直到找到原点信号（远点），这里 DogSignal 和 RefenenceSignal 都指定 M10 代表省略 DOG（这种情况应将 Velococity 适当调小）；

SetPosition 代表完成回归后设定当前位置为该值。

2.19.7 MC_MoveFollow/跟随指令

1、指令详解

		跟随指令，可用作编码器轴
输入参数	解释	参数类型
AxisId	轴 ID	AXIS_REF_AXISID
Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
N	分子	DWORD
M	分母	DWORD
Cnt	关联的 32 位高速计数器	DWORD
输出参数		
Done	保留	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令。

当 EN 复位时输出参数都会被复位

Cnt 可指定 C235-C255

$$\frac{N}{M} = \frac{\text{轴速度（User Unit/s）}}{\text{C235 频率/单位脉冲个数}}$$

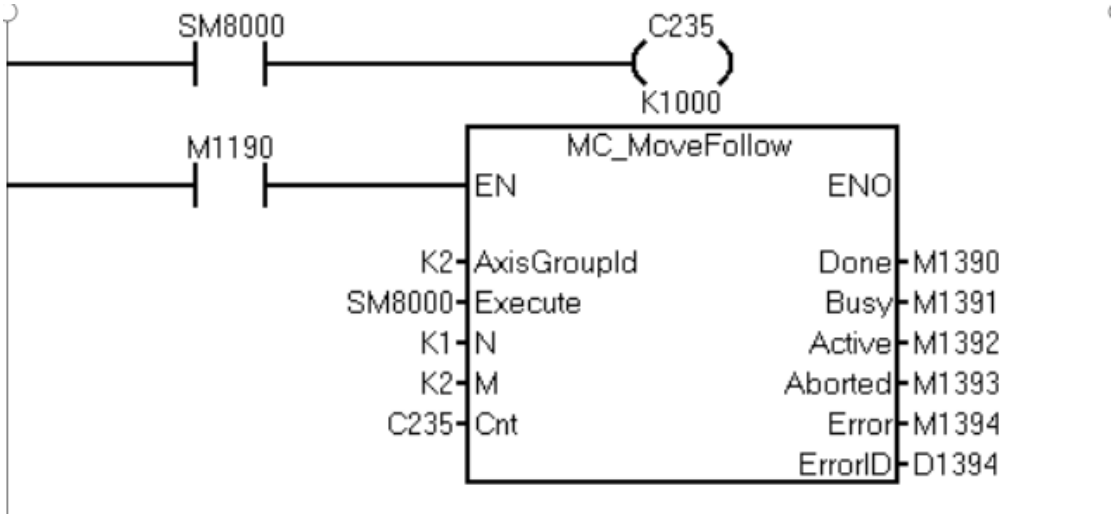
注:

单位脉冲个数=电机转一圈所需脉冲个数/电机转一圈的位移

3、程序举例

轴 2 跟随 C235 运动

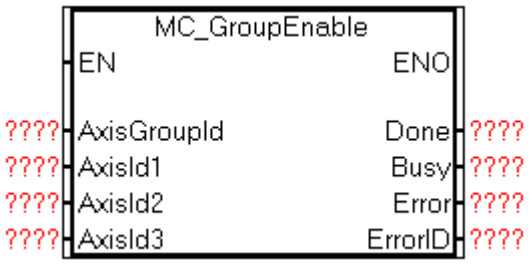
注：图中省略了轴 2Power



当 M1190 打开时轴 2 将跟随 C235 的计数而运动，脉冲对应关系可以通过轴 2 配置参数调整，速度比可以通过 N 和 M 的值调整，轴 2 速度与 C235 输入频率的比例关系为 N:(M*单位脉冲个数)。

2.19.8 MC_GroupEnable/启用轴组指令

1、指令详解

	使轴组成立

输入参数	解释	参数类型
AxisGroupId	轴组 ID	AXIS_REF_AXISGROUPID
AxisId1	轴 1ID，不可省略	AXIS_REF_AXISID
AxisId2	轴 2ID，不可省略	AXIS_REF_AXISID
AxisId3	轴 3ID，写 K0 省略	AXIS_REF_AXISID
输出参数		
Done	动作完成时置 True	BOOL

Busy	当指令被触发执行时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

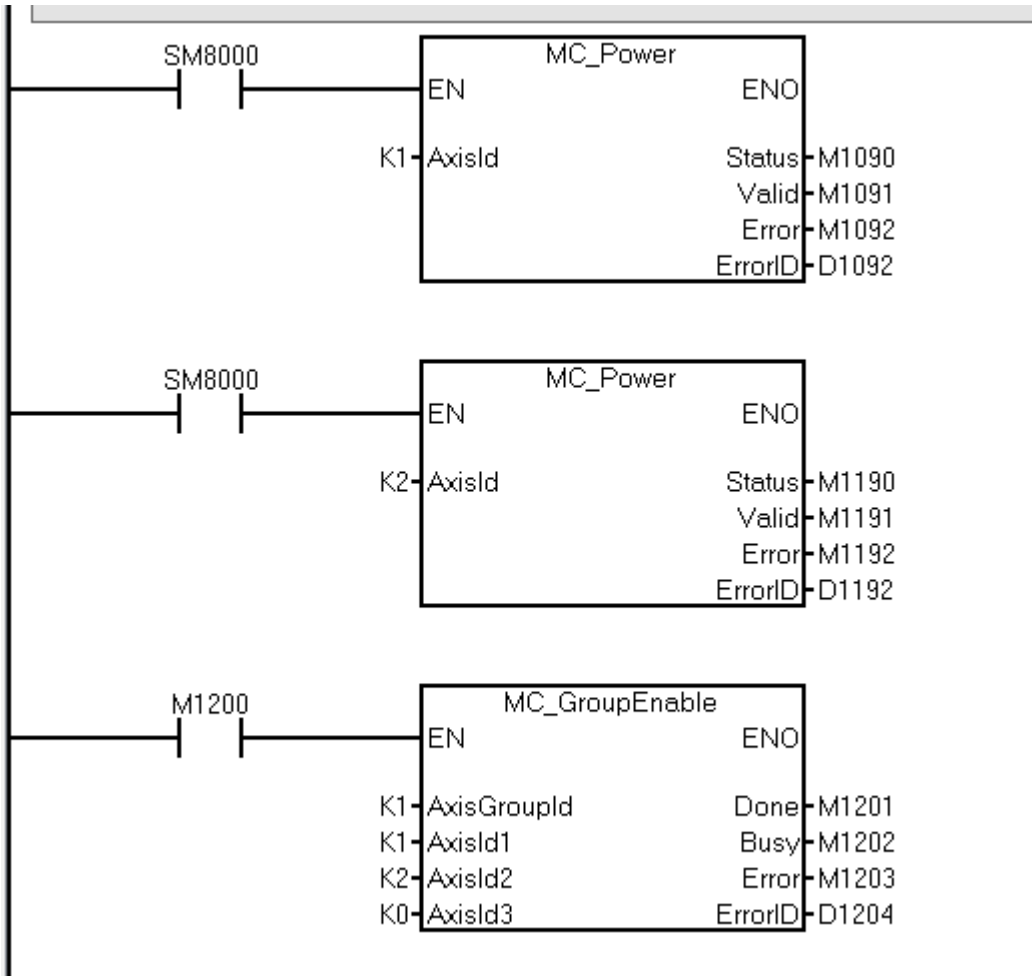
添加轴组的轴必须先使用 MC_Power 使能，且处于静止状态

EN 生效时将 AxisId1、AxisId2、AxisId3（可写 K0 省略）加入轴组 AxisGroupId

当 EN 复位时输出参数都会被复位

3、程序举例

添加轴 1 和轴 2 到轴组 1



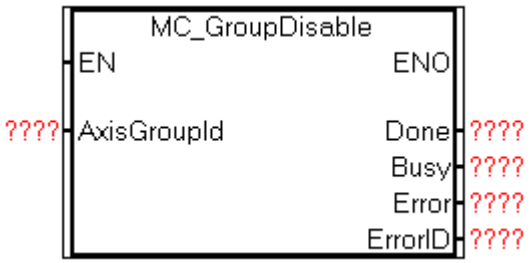
使用轴组前需要先 power 相应的轴，最多可以添加三个轴到一个轴组中（三轴只支持直线插补，圆弧只能选两轴）。AxisId1，AxisId2，AxisId3 分别指定 X、Y、Z 轴的 ID，AxisId3=0 表示 Z 轴不存在。

SD101 和 SD201 分别指示轴 1 和轴 2 所在轴组编号，打开 M1200 后可以查看：

软元件/标签	数据类型	当前值
M1200	BIT	1
M1201	BIT	1
SD101	WORD	1
SD201	WORD	1

2.19.9 MC_GroupDisable/关闭轴组指令

1、指令详解

		失能轴组
输入参数	解释	参数类型
AxisGroupId	轴组 ID	AXIS_REF_AXISGROUPID
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_GroupEnable 指令将轴组切换到使能状态才可以调用本指令，且必须在轴组无动作(没有其他相关指令执行)时使用

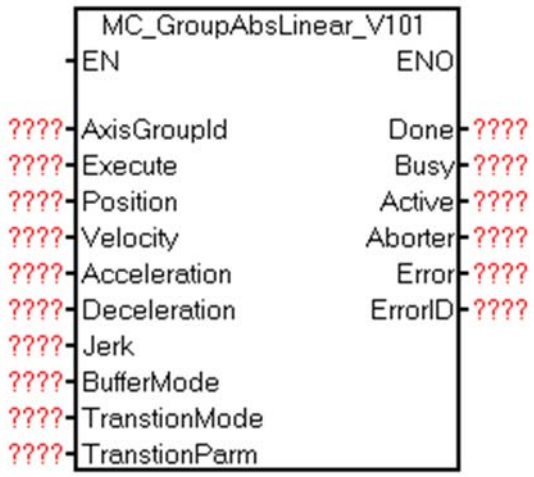
当 EN 复位时输出参数都会被复位

执行成功后会把 SDx01（轴组所在 ID）的值清空

如果轴组正在执行运动指令，可以先使用 MCGroupStop 打断后再执行 MC_GroupDisable。

2.19.10 MC_GroupAbsLinear_V101/绝对直线插补指令

1、指令详解

	控制轴组执行直线插补，使用绝对位置

输入参数	解释	参数类型
AxisGroupId	轴组 ID	AXIS_REF_AXISGROUPID
Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Position	绝对位置（用户单位）	REAL[2]或 REAL[3]
Velocity	目标速度（用户单位/秒）	REAL
Acceleration	加速度（用户单位/秒^2）	REAL
Deceleration	减速度（用户单位/秒^2）	REAL
Jerk	加加速度；跃度 （用户单位/秒^3）	REAL
BufferMode	指定此功能块指令的缓冲行为模式	MC_BUFFER_MODE K0 打断模式 K1 缓冲模式 K3 连续插补以前一个速度过渡 K4 连续插补以后一个速度过渡
TranstionMode	连续插补过渡模式	MC_TRANSITION_MODE K0 不用圆弧过渡 K1 使用圆弧过渡
TranstionPam	连续插补圆弧过渡半径	REAL

输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_GroupEnable 指令将轴组切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisGroupId 的值

在 EN 的上升沿锁存 BufferMode

当 EN 复位时输出参数都会被复位。

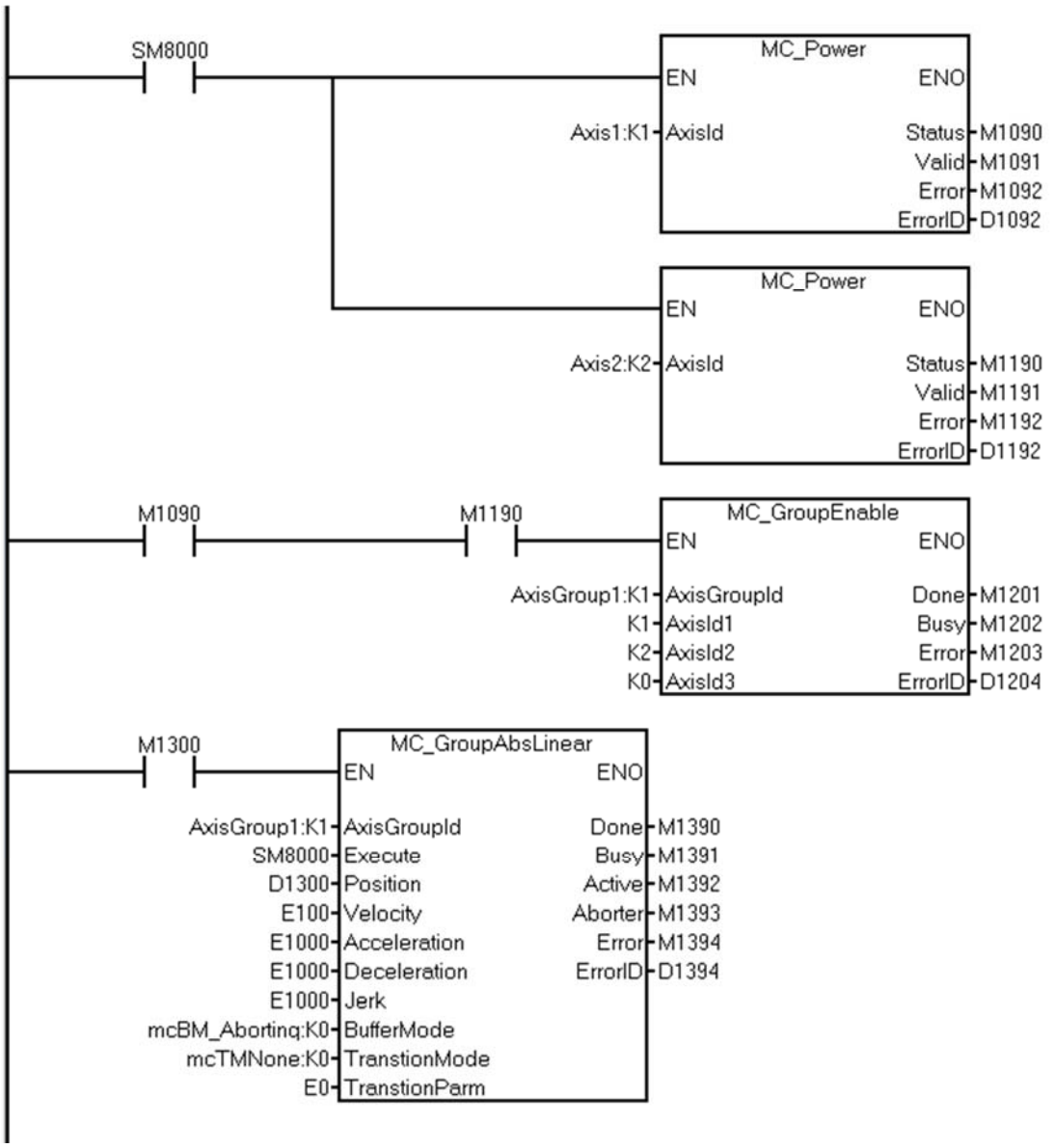
在 Execute 输入的上升沿，指令锁存 Position，Velocity，Acceleration，Deceleration，Jerk 并触发绝对直线插补

Position 会占用 2(2 轴)或 3(3 轴)个连续的 DWORD

BufferMode=3 和 4 时会触发连续插补功能，详情参考 BufferMode 和连续插补功能说明

3、程序举例

从位置（0,0）走绝对位移到位置(100,100)



Position 参数会占用 2 个 DWORD（3 轴是 3 个），从 D1300 到 D1303，组成两个 32 位浮点数，用来指示 XY 轴的坐标；

Velocity 参数指定轴 1 轴 2 合成速度；

相关寄存器初始状态如下：

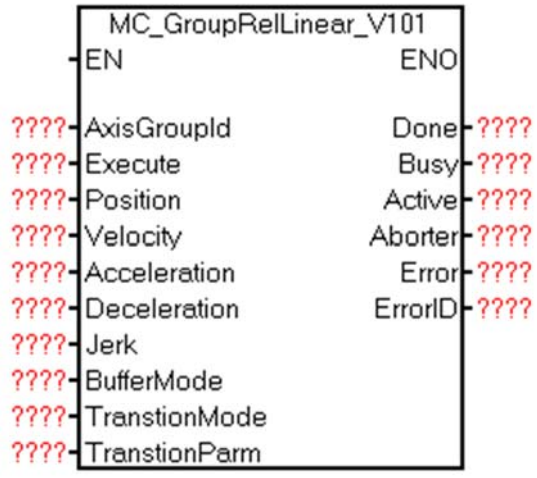
软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	0
D1300	FLOAT	100.0
D1302	FLOAT	100.0
SD104	FLOAT	0.0
SD204	FLOAT	0.0
M1390	BIT	0

打开 M1300 后发生如下改变：

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	1
D1300	FLOAT	100.0
D1302	FLOAT	100.0
SD104	FLOAT	100.0
SD204	FLOAT	100.0
M1390	BIT	1

2.19.11 MC_GroupRelLinear_V101/相对直线插补指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制轴组执行直线插补，使用相对位置</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisGroupId</td><td>轴组 ID</td><td>AXIS_REF_AXISGROUPID</td></tr>
</table>

Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Position	相对位置（用户单位）	REAL[2]或 REAL[3]
Velocity	目标速度（用户单位/秒）	REAL
Acceleration	加速度（用户单位/秒^2）	REAL
Deceleration	减速度（用户单位/秒^2）	REAL
Jerk	加加速度；跃度 （用户单位/秒 ^3）	REAL
BufferMode	指定此功能块指令的缓冲行为模式	MC_BUFFER_MODE K0 打断模式 K1 缓冲模式 K3 连续插补以前一个速度过渡 K4 连续插补以后一个速度过渡
TranstionMode	连续插补过渡模式	MC_TRANSITION_MODE K0 不用圆弧过渡 K1 使用圆弧过渡
TranstionPam	连续插补圆弧过渡半径	REAL
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_GroupEnable 指令将轴组切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisGroupId 的值

在 EN 的上升沿锁存 BufferMode

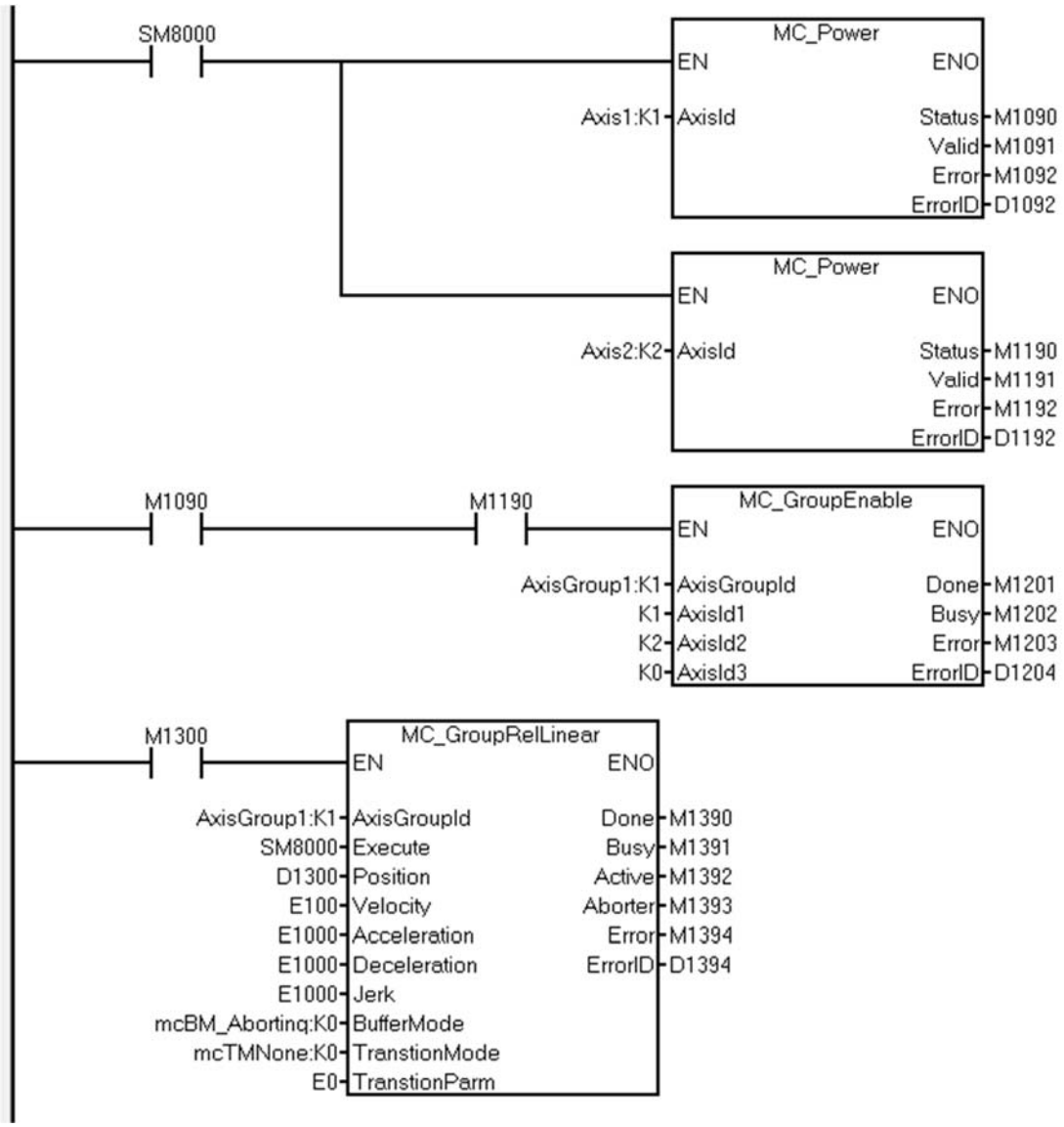
当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿，指令锁存 Position，Velocity，Acceleration，Deceleration，Jerk 并触发相对直线插补

Position 会占用 2(2 轴)或 3(3 轴)个连续的 DWORD

3、程序举例

从位置（100,100）位移（100,100）



初始状态如下：

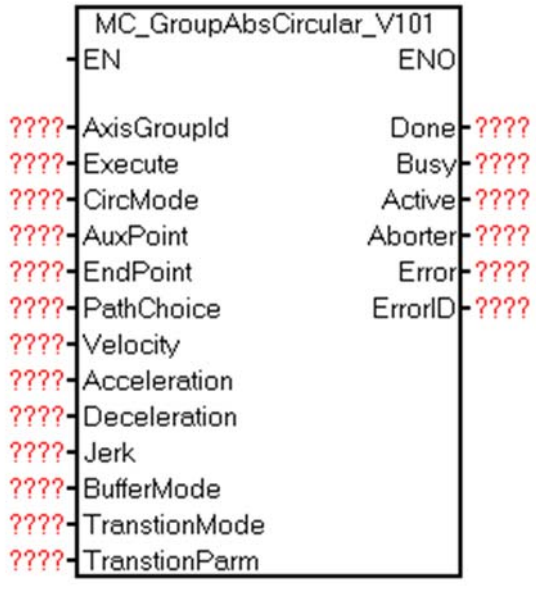
软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	0
D1300	FLOAT	100.0
D1302	FLOAT	100.0
SD104	FLOAT	100.0
SD204	FLOAT	100.0
M1390	BIT	0

打开 M1300 后发生如下变化：

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	1
D1300	FLOAT	100.0
D1302	FLOAT	100.0
SD104	FLOAT	200.0
SD204	FLOAT	200.0
M1390	BIT	1

2.19.12 MC_GroupAbsCircular_V101/绝对圆弧插补指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制轴组执行圆弧，使用绝对位置</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisGroupId</td><td>轴组 ID</td><td>AXIS_REF_AXISGROUPID</td></tr>
</table>

Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
CircMode	圆弧模式	MC_CIRC_MODE K0　圆上任意一点 K1　圆心模式 K2　半径模式
AuxPoint	辅助点	REAL[2]
EndPoint	绝对目标位置	REAL[2]
PathChoice	顺圆或逆圆	MC_CIRC_PATHCHOICE K0 顺圆 K1 逆圆
Velocity	目标速度（用户单位/秒）	REAL
Acceleration	加速度（用户单位/秒^2）	REAL
Deceleration	减速度（用户单位/秒^2）	REAL
Jerk	加加速度；跃度 （用户单位/秒 ^3）	REAL
BufferMode	指定此功能块指令的缓冲行为模式	MC_BUFFER_MODE K0　打断模式 K1　缓冲模式 K3　连续插补以前一个速度过渡 K4　连续插补以后一个速度过渡
TranstionMode	连续插补过渡模式	MC_TRANSITION_MODE K0　不用圆弧过渡 K1　使用圆弧过渡
TranstionPam	连续插补圆弧过渡半径	REAL
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL

Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_GroupEnable 指令将轴组切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisGroupId 的值

在 EN 的上升沿锁存 BufferMode

当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿,指令锁存 CircMode,CircMode,EndPoint,Velocity,Acceleration,Deceleration，Jerk 并触发绝对圆弧插补

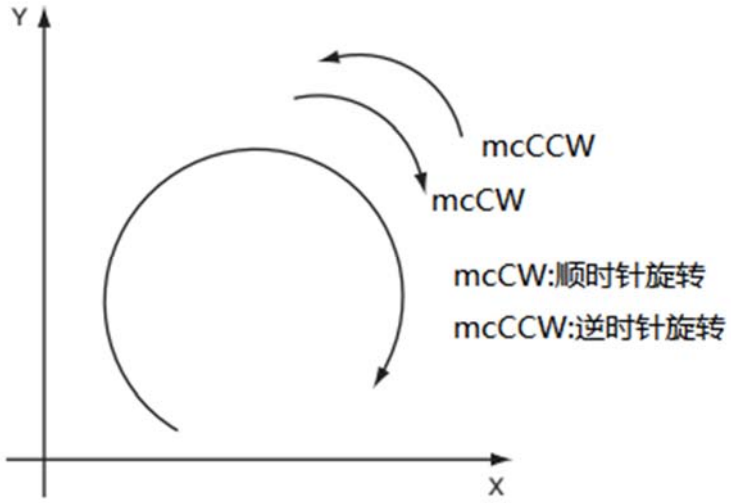
AuxPoint 和 EndPoint 会占用 2 个连续的 DWORD

AuxPoint 或 EndPoint 在指定位置时代表绝对位置

当起始位置和目标点位置相同时代表画一个整圆

BufferMode=3 和 4 时会触发连续插补功能，详情参考 BufferMode 和连续插补功能说明

圆弧插补使用的轴为 X 轴，Y 轴。

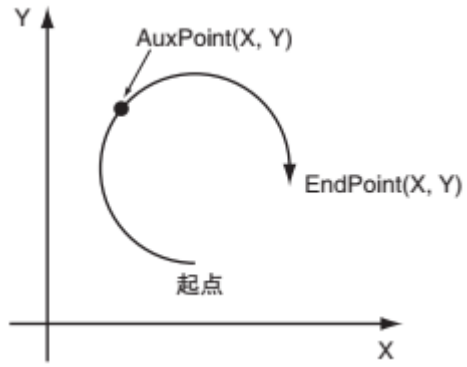


CircMode（圆弧插补模式）

圆弧插补方式有“圆上任意一点指定/中心点指定/半径指定”三种，通过 CircMode 指定。

（1）任意圆上一点指定

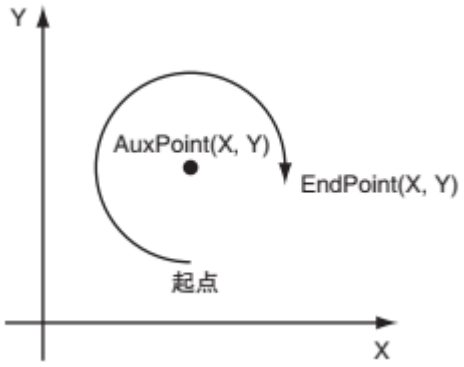
以当前位置为起点，圆上一点位置 AuxPoint(X,Y)和终点 EndPoint(X,Y)的圆弧插补。



起点，圆上一点位置与终点在同一条直线上，或者圆上一点位置与终点为同一点，以及起点和圆上一点位置为同一点时，会发生异常报错；起点和终点为同一点时，也会发生异常报错。

（2）中心点指定

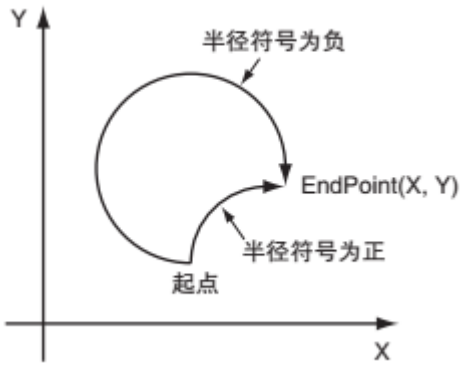
以当前位置为起点，执行以通过中心位置 AuxPoint(X,Y)指定的圆弧连接终点 EndPoint(X,Y)的圆弧插补。圆弧的旋转方向通过 PathChoice 指定。起点，终点为同一点时，绘制正圆。



（3）半径指定

以当前位置为起点，执行以通过半径 AuxPoint(X,Y)指定的圆弧连接终点 EndPoint(X,Y)的圆弧插补。半径通过 AuxPoint(X,Y)的第一个元素来指定，不使用第二个元素。例如，半径为 100 时，AuxPoint(X,Y)指定为 AuxPoint(100,0)。半径符号为负数时，绘制出较长的

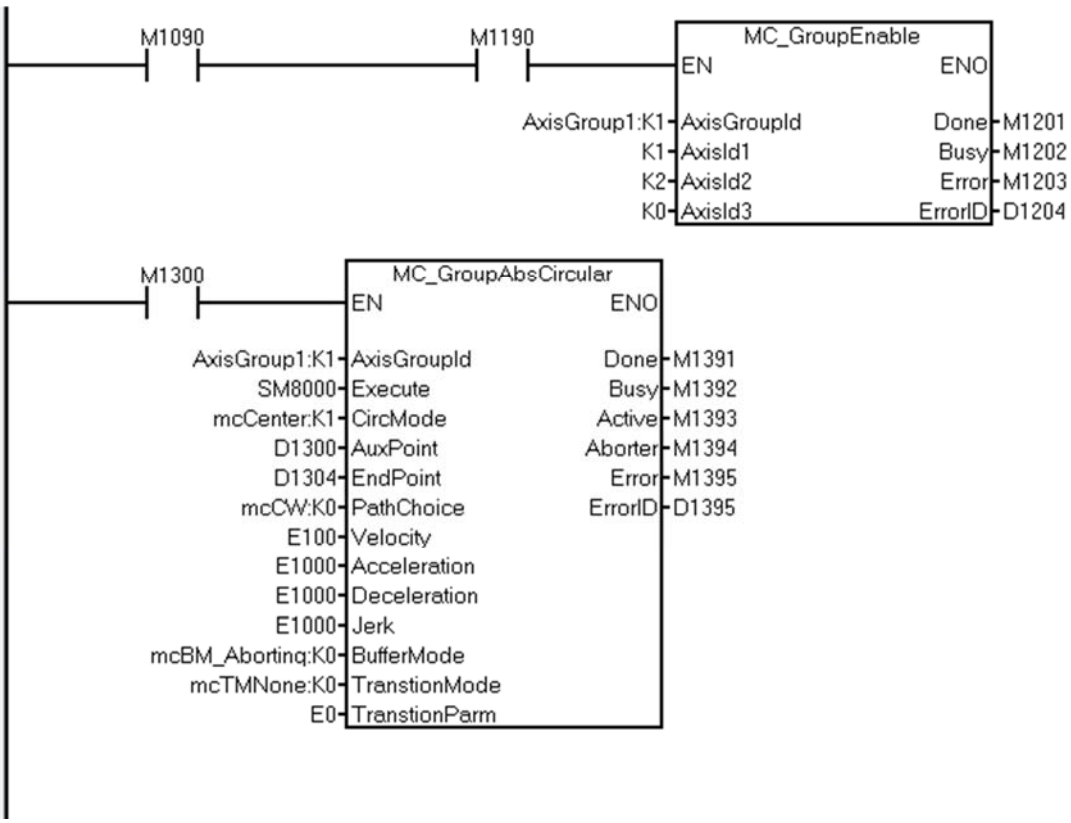
圆弧，半径符号为正时，绘制出较短的圆弧。圆弧的旋转方向通过 PathChoice 指定。指定的半径小于起点和终点间距离的 1/2 时，无法绘制圆弧，将出现异常。



3、程序举例

从当前位置（0,0），以（50,0）为半径走绝对圆弧到目标点（100，0）

注：图中省略了轴 1 轴 2Power



CicMode=1 表示采用圆心模式画圆

PathChoice=0 表示顺时针画圆

相关寄存器初始状态如下：

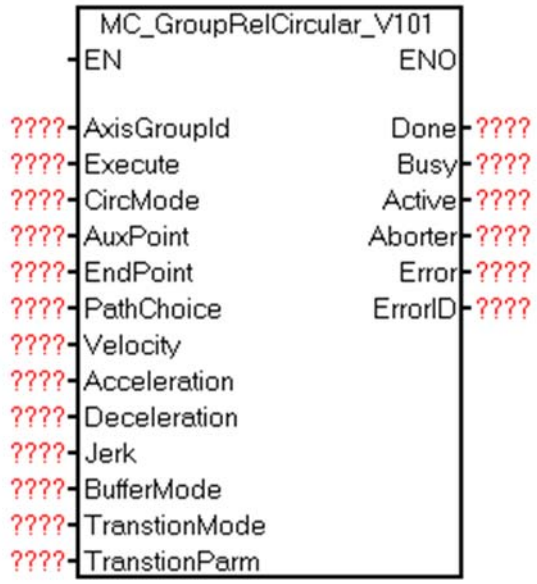
软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	0
D1300	FLOAT	50.0
D1302	FLOAT	0.0
D1304	FLOAT	100.0
D1306	FLOAT	0.0
M1391	BIT	0
SD108	FLOAT	0.0
SD208	FLOAT	0.0
SD104	FLOAT	0.0
SD204	FLOAT	0.0

打开 M1300 圆弧插补执行完后相关寄存器状态如下

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	1
D1300	FLOAT	50.0
D1302	FLOAT	0.0
D1304	FLOAT	100.0
D1306	FLOAT	0.0
M1391	BIT	1
SD108	FLOAT	0.0
SD208	FLOAT	0.0
SD104	FLOAT	100.0
SD204	FLOAT	0.0

2.19.13 MC_GroupRelCircular_V101/相对圆弧插补指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>控制轴组执行圆弧，使用相对位置</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisGroupId</td><td>轴组 ID</td><td>AXIS_REF_AXISGROUPID</td></tr>
<tr><td>Execute</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>CircMode</td><td>圆弧模式</td><td>MC_CIRC_MODE
K0 圆上任意一点
K1 圆心模式
K2 半径模式</td></tr>
<tr><td>AuxPoint</td><td>辅助点</td><td>REAL[2]</td></tr>
<tr><td>EndPoint</td><td>相对目标位置</td><td>REAL[2]</td></tr>
<tr><td>PathChoice</td><td>顺圆或逆圆</td><td>MC_CIRC_PATHCHOICE
K0 顺圆
K1 逆圆</td></tr>
<tr><td>Velocity</td><td>目标速度（用户单位/秒）</td><td>REAL</td></tr>
<tr><td>Acceleration</td><td>加速度（用户单位/秒^2）</td><td>REAL</td></tr>
<tr><td>Deceleration</td><td>减速度（用户单位/秒^2）</td><td>REAL</td></tr>
<tr><td>Jerk</td><td>加加速度；跃度
（用户单位/秒 ^3）</td><td>REAL</td></tr>
<tr><td>BufferMode</td><td>指定此功能块指令的缓冲行</td><td>MC_BUFFER_MODE</td></tr>
</table>

	为模式	K0 打断模式 K1 缓冲模式 K3 连续插补以前一个速度过渡 K4 连续插补以后一个速度过渡
TranstionMode	连续插补过渡模式	MC_TRANSITION_MODE K0 不用圆弧过渡 K1 使用圆弧过渡
TranstionPam	连续插补圆弧过渡半径	REAL
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

只有使用 MC_GroupEnable 指令将轴组切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisGroupId 的值

在 EN 的上升沿锁存 BufferMode

当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿,指令锁存 CircMode,CircMode,EndPoint,Velocity,Acceleration, Deceleration, Jerk 并触发相对圆弧插补

AuxPoint 和 EndPoint 会占用 2 个连续的 DWORD

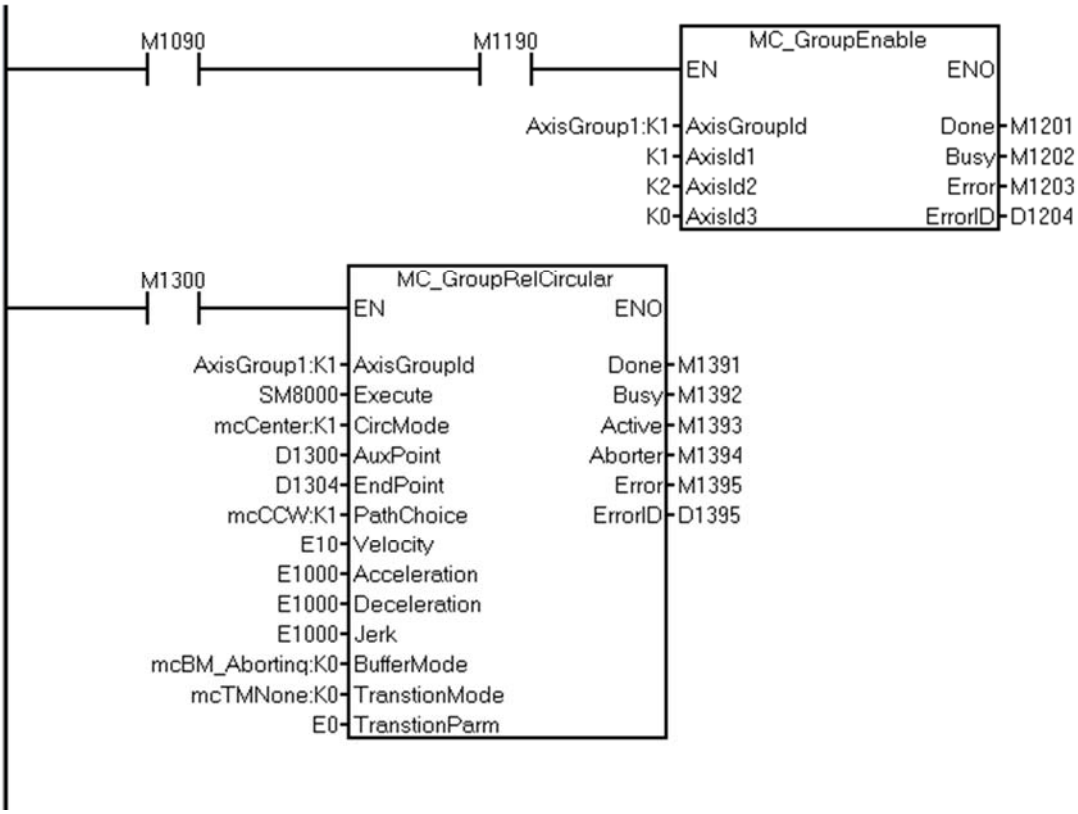
AuxPoint 或 EndPoint 在指定位置时代表相对位置

当目标点为(0,0)时代表画一个整圆

3、程序举例

以位置（100,0）起始，相对圆心（100,0），走相对圆弧插补到相对目标点（0,0）

注：图中省略了轴 1 轴 2Power



PathChoice=1 表示逆时针画圆

相关寄存器初始状态如下

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	0
D1300	FLOAT	50.0
D1302	FLOAT	0.0
D1304	FLOAT	0.0
D1306	FLOAT	0.0
M1391	BIT	1
SD108	FLOAT	0.0
SD208	FLOAT	0.0
SD104	FLOAT	100.0
SD204	FLOAT	0.0

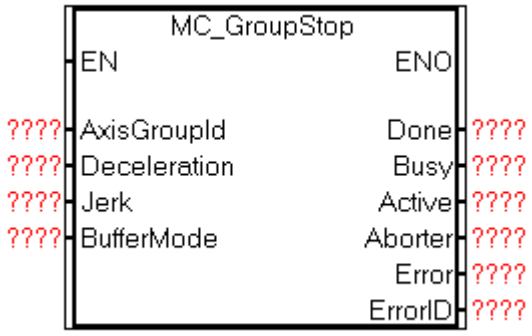
终点相对坐标设为(0,0)表示画一个整圆。

打开 M1300 圆弧插补完成后发生如下变化：

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	1
D1300	FLOAT	50.0
D1302	FLOAT	0.0
D1304	FLOAT	0.0
D1306	FLOAT	0.0
M1391	BIT	0
SD108	FLOAT	0.0
SD208	FLOAT	0.0
SD104	FLOAT	100.0
SD204	FLOAT	0.0

2.19.14 MC_GroupStop/轴组停止指令

1、指令详解

		控制指定轴组减速到停止
输入参数	解释	参数类型
AxisGroupId	轴组 ID	AXIS_REF_AXISGROUPID
Deceleration	减速度（用户单位/秒^2）	REAL
Jerk	加加速度；跃度 （用户单位/秒 ^3）	REAL
BufferMode	保留参数	MC_BUFFER_MODE
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当运动轴受控制时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL

Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

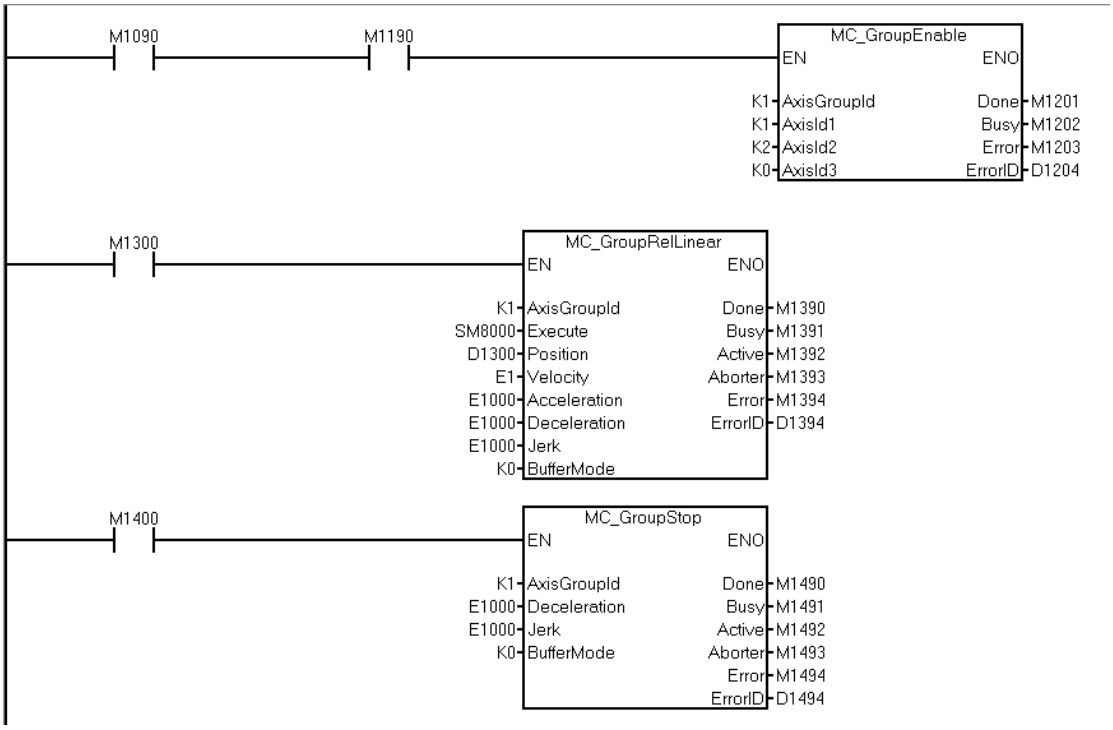
只有使用 MC_GroupEnable 指令将轴组切换到使能状态才可以调用本指令。

当 EN 复位时输出参数都会被复位

MC_GroupStop 默认使用打断模式，会清空改轴组的整个任务队列。

3、程序举例

使用 MC_GroupStop 指令结束相对位移插补指令



运行梯形图并打开 M1300：

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	1
D1300	FLOAT	-100.0
D1302	FLOAT	-100.0
SD108	FLOAT	-1.0
SD208	FLOAT	-1.0
SD104	FLOAT	-4.063
SD204	FLOAT	-4.063
M1390	BIT	0
M1400	BIT	0
M1391	BIT	1

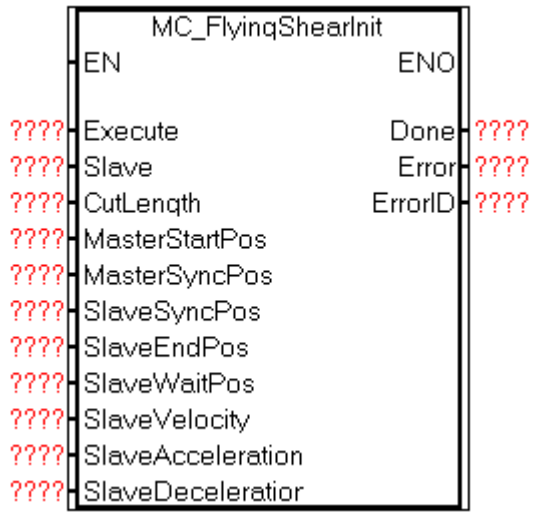
打开 M1400 时，相关寄存器发生如下变化：

软元件/标签	数据类型	当前值
M1090	BIT	1
M1190	BIT	1
M1201	BIT	1
M1300	BIT	1
D1300	FLOAT	-100.0
D1302	FLOAT	-100.0
SD108	FLOAT	0.0
SD208	FLOAT	0.0
SD104	FLOAT	-9.268
SD204	FLOAT	-9.268
M1390	BIT	0
M1400	BIT	1
M1391	BIT	0

轴组 1 停止，轴 1 轴 2 速度变为 0。

2.19.15 MC_FlyingShearInit/追剪初始化指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于追剪功能执行之前的参数初始化工作，必须首先执行该功能块，才能继续执行追剪其他相关的功能块。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Slave
从轴号</td><td>指定的从轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>CutLength
剪切长度</td><td>材料的切割长度</td><td>REAL</td></tr>
<tr><td>MasterStartPos
主轴开始位置</td><td>开始位置
当主轴到达此位置时，从轴由等待位置追赶主轴，以实现速度同步</td><td>REAL</td></tr>
<tr><td>MasterSyncPos
主轴同步开始位置</td><td>同步区开始的时候，对应的主轴位置</td><td>REAL</td></tr>
<tr><td>SlavesyncPos
从轴同步开始位置</td><td>同步区开始的时候，对应的从轴位置</td><td>REAL</td></tr>
<tr><td>SlaveEndPos
从轴同步结束位置</td><td>同步区结束时，对应的从轴位置</td><td>REAL</td></tr>
<tr><td>SlaveWaitPos
从轴等待位置</td><td>从轴的等待位置。
当追剪功能启动后，从轴自动运行到此位置</td><td>REAL</td></tr>
</table>

SlaveVelocity 从轴速度	从轴终端执行机构的运行速度，此参数总为正，单位：单元/秒	REAL
SlaveAcceleration 从轴加速度	从轴终端执行机构的加速度，此参数总为正，单位：单元/秒^2	REAL
SlaveDeceleration 从轴减速度	从轴终端执行机构的减速度，此参数总为正，单位：单元/秒^2	REAL
输出参数		
Done 完成位	当初始化指令执行完成时，Done 位被置位	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

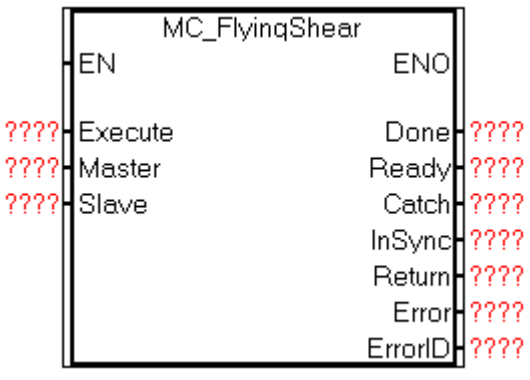
只有使用 MC_Power 指令将轴 Slave 切换到使能状态才可以调用本指令。

在 EN 后请不要改变 Slave 的值。

当 EN 复位时输出参数都会被复位

2.19.16 MC_FlyingShear/追剪指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此命令用于建立追剪关系,可以根据应用需求指定主轴与从轴的轴号。当指令执行时，指令的输出装置能指示出追剪所处区间。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
</table>

输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Master 主轴轴号	指定的主轴轴号	AXIS_REF_AXISID
Slave 从轴轴号	指定的从轴轴号	AXIS_REF_AXISID
输出参数		
Done 完成位	当追剪进入完成时，Done 位被置位	BOOL
Ready 准备位	从轴到达等待位置后被置位，当进入追赶区时被复位。	BOOL
Catch 追赶区标位	追赶区开始时被置位,追赶区结束后被复位	BOOL
Insync 同步区标位	同步区开始时被置位,同步区结束后被复位	BOOL
Return 返回区标位	返回区开始时被置位,返回区结束时被复位	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

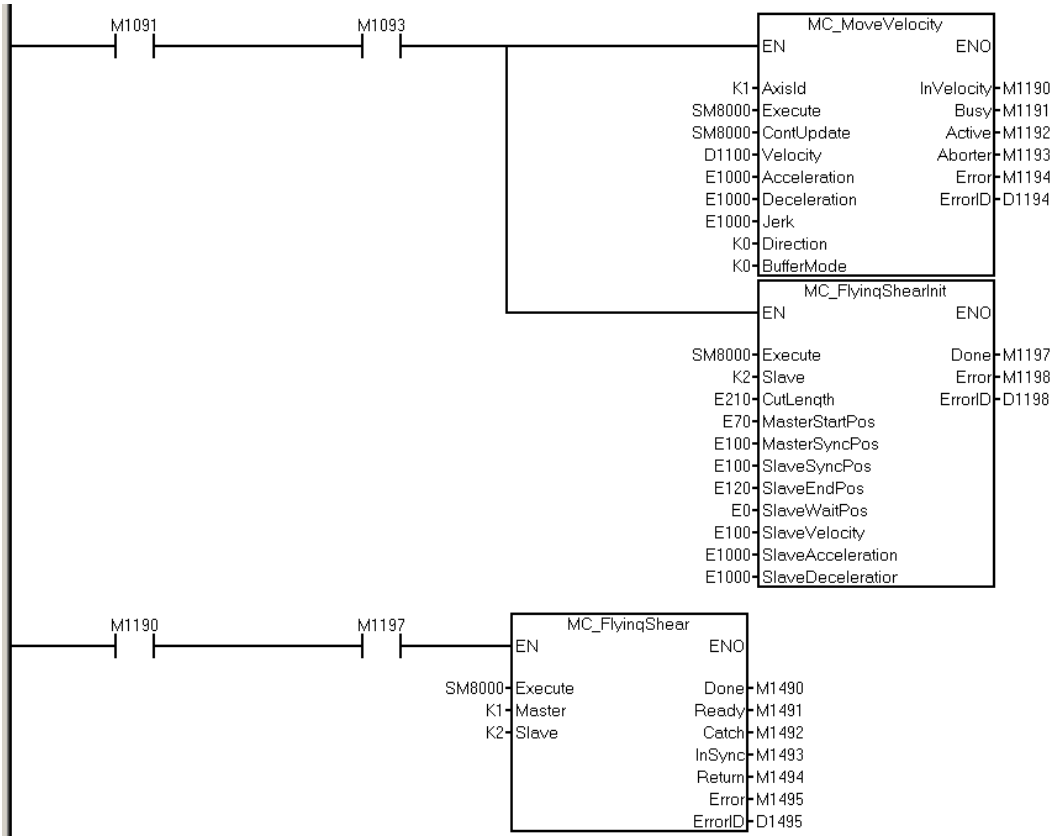
当轴 Slave 使用 MC_FlyingShearInit 指令后才能使用 MC_FlyingShear 指令

在 EN 后请不要改变 Master 和 Slave 的值。

当 EN 复位时输出参数都会被复位

3、程序举例

注：图中省略了轴 1 轴 2Power



梯形图下载后轴 1 开始做匀速运动，并对追剪的参数进行初始化，当 MC_MoveVelocity 的输出 InVelocity 置位后开始执行追剪指令，此时监视列表如图

监视列表

	软元件/标签	数据类型	当前值	更改当前值
1	M1190	BIT	1	
2	M1197	BIT	1	
3	M1490	BIT	1	
4	M1295	BIT	0	
5	M1496	BIT	0	

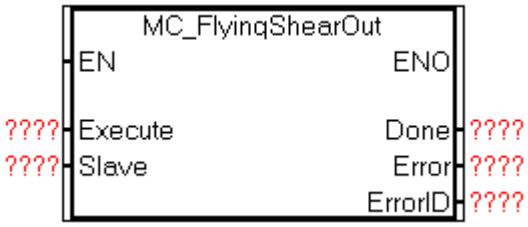
手动置位 M1295 后执行追剪脱离指令，追剪轴立即停止，主轴不受影响继续匀速运行，监视列表如图

监视列表

	软元件/标签	数据类型	当前值	更改当前值
1	M1190	BIT	1	
2	M1197	BIT	1	
3	M1490	BIT	0	
4	M1295	BIT	1	
5	M1496	BIT	1	

2.19.17 MC_FlyingShearOut/追剪脱离指令

1、指令详解

		此指令用于解除主轴与从轴之间已经建立的追剪关系。此指令会让从轴回到从轴等待位置，主轴不受影响。
参数名称	参数解释	参数类型
输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Slave 从轴轴号	指定的从轴编码	AXIS_REF_AXISID
输出参数		
Done 完成位	当追剪脱离完成时 ，Done 位被置位	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

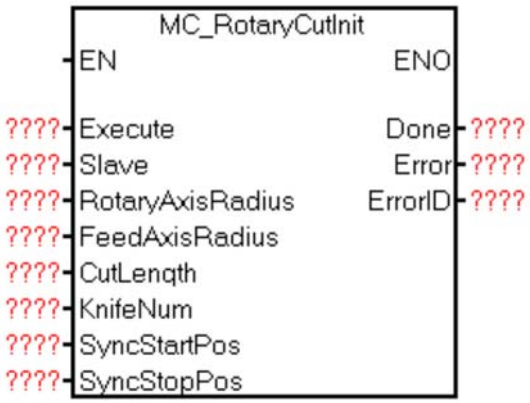
当轴 Slave 使用 MC_FlyingShear 指令进入同步状态后才能使用 MC_FlyingShearOut 指令

在 EN 后请不要改变 Slave 的值

当 EN 复位时输出参数都会被复位

2.19.18 MC_RotaryCutInit/旋切初始化指令

1、指令详解

		此指令用于旋切功能执行之前的参数初始化工作，必须首先执行该功能块，才能继续执行旋切其他相关的功能块。
参数名称	参数解释	参数类型
输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Slave 从轴号	指定的从轴编码	AXIS_REF_AXISID
RotaryAxisRadius 旋切轴半径	旋切轴半径。即旋切棍圆心到刀尖的距离。	REAL
FeedAxisRadius 进给轴半径	进给轴半径。即进给辊半径长度	REAL
CutLength 剪切长度	材料的剪切长度	REAL
KnifeNum 刀头数量	旋切轴上切刀的数量	DWORD
SyncStartPos 同步区的起始位置	同步区的起始位置。即同步区开始时对应的进给轴位置	REAL
SyncStopPos 同步区的结束位置	同步区的结束位置。即同步区结束时对应的进给轴位置	REAL
输出参数		
Done 完成位	当初始化指令执行完成时，Done 位被置位	BOOL
Error	检测到有错误，Error 位被置	BOOL

错误位	位。	
ErrorID 错误代码	错误代码。	WORD

2、功能描述

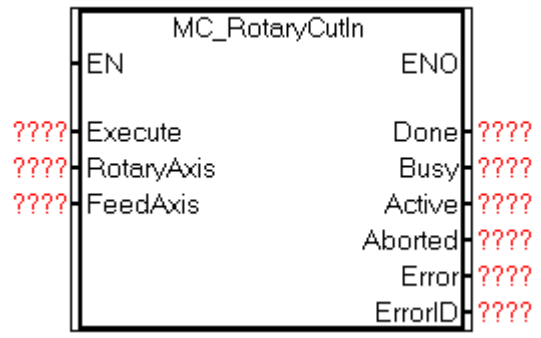
当轴 Slave 使用 MC_Power 指令后才能使用该指令

当 EN 复位时输出参数都会被复位

旋切耦合指令运行时，若修改旋切参数后需要重新复位下该模块触点，然后在使能运行该模块，参数在下一个旋切周期中生效

2.19.19 MC_RotaryCutIn/旋切耦合指令

1、指令详解

		建立旋切关系 根据实际的应用需求，指定旋切轴和进给轴，当指令执行成功后，旋切轴按照指定的旋切曲线进行运动。
参数名称	参数解释	参数类型
输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
RotaryAxis 旋切轴轴号	指定的旋切轴编码	AXIS_REF_AXISID
FeedAxis 进给轴轴号	指定的进给轴编码	AXIS_REF_AXISID
输出参数		
Done 完成位	当旋切进入完成时，Done 位被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL

Active 控制中	开始指令有效，则输出 ON。	BOOL
Aborted 执行中断	指令被其他指令中断时,则输出 ON。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

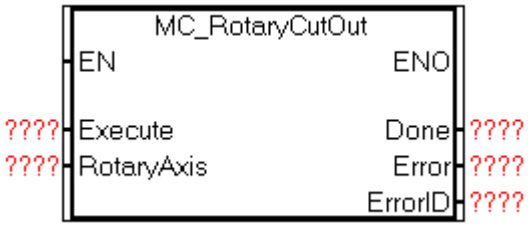
当轴 RotaryAxis 使用 MC_RotaryCutInit 指令初始化后才能使用该指令

在 EN 后请不要改变 RotaryAxis 和 FeedAxis 的值

当 EN 复位时输出参数都会被复位

2.19.20 MC_RotaryCutOut/旋切脱离指令

1、指令详解

		此指令用于解除旋切轴与进给轴之间已经建立的旋切关系。关系解除后，旋切轴的刀头会停在进入点,且旋切轴不再跟随进给轴。此指令不会影响进给轴的运动。
参数名称	参数解释	参数类型
输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
RotaryAxis 旋切轴轴号	指定的旋切轴编码	AXIS_REF_AXISID
输出参数		
Done 完成位	当旋切脱离完成时 ，Done 位被置位	BOOL
Error	检测到有错误，Error 位被置位。	BOOL

错误位		
ErrorID 错误代码	错误代码。	WORD

2、功能描述

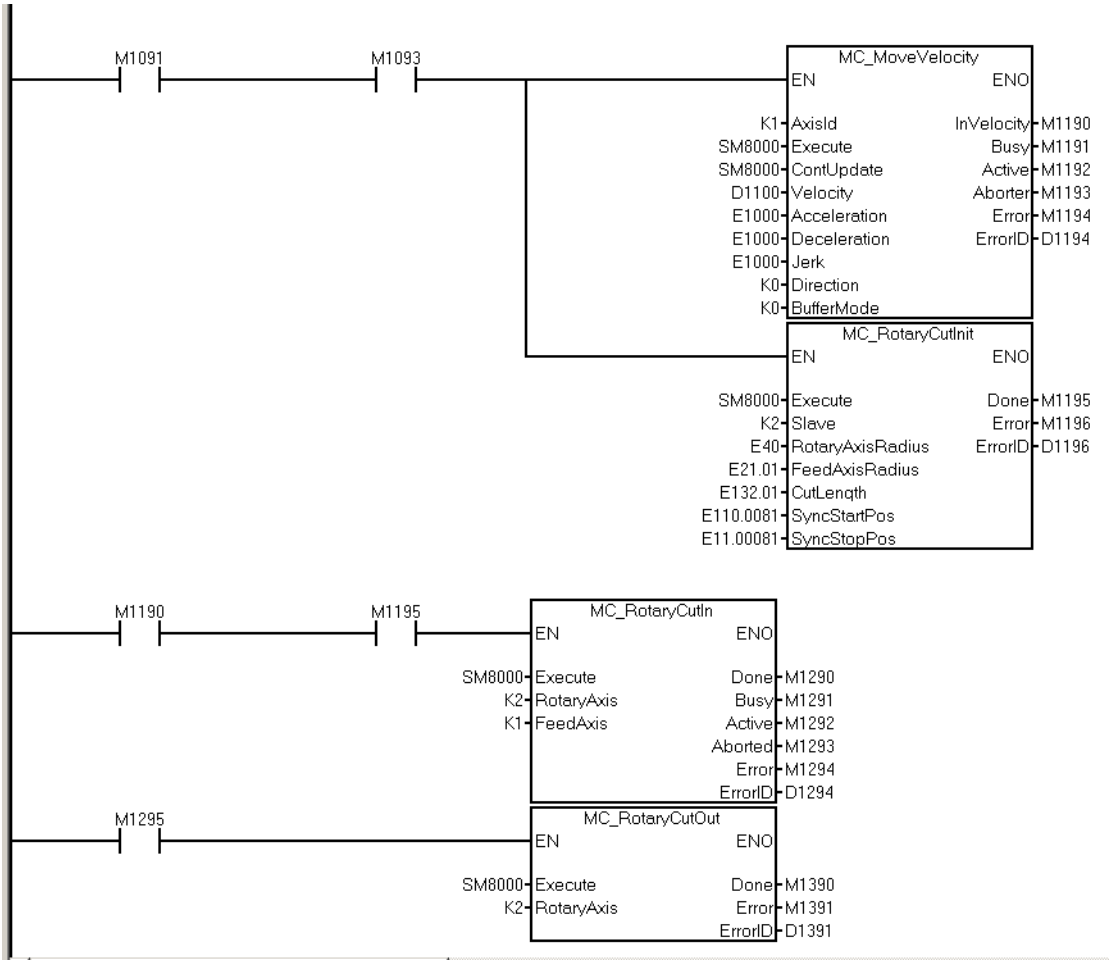
当轴 RotaryAxis 使用 MC_RotaryCutIn 指令进入同步状态后才能使用该指令

在 EN 后请不要改变 RotaryAxis 的值

当 EN 复位时输出参数都会被复位

3、程序举例

注：图中省略了轴 1 轴 2Power



梯形图下载后轴 1 开始做匀速运动，并对旋切的参数进行初始化，当 MC_MoveVelocity 的输出 InVelocity 置位后开始执行旋切指令，此时监视列表如图

监视列表

	软元件/标签	数据类型	当前值	更改当前值
1	M1295	BIT	0	
2	M1190	BIT	1	
3	M1195	BIT	1	
4	M1290	BIT	1	
5	M1390	BIT	0	

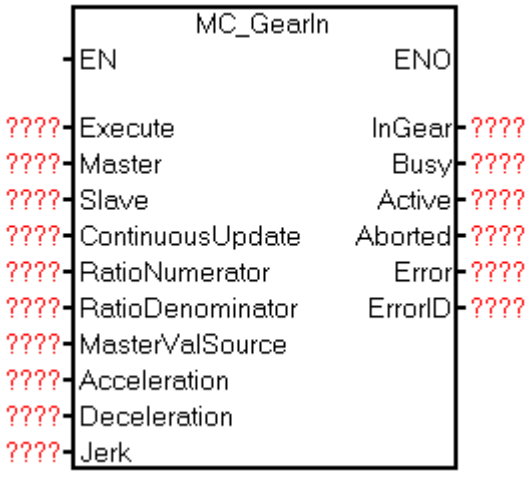
手动置位 M1295 后执行旋切脱离指令，旋切轴停在开始的位置，主轴不受影响继续匀速运行，监视列表如图

监视列表

	软元件/标签	数据类型	当前值	更改当前值
1	M1295	BIT	1	
2	M1190	BIT	1	
3	M1195	BIT	1	
4	M1290	BIT	0	
5	M1390	BIT	1	

2.19.21 MC_GearIn/电子齿轮耦合指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于建立主从轴间的齿轮关系。建立齿轮关系时，可设定齿轮比等参数。齿轮关系建立后，从轴会以给定的比例关系跟随主轴运动，实现主从轴同步控制。主从轴可以是实轴，也可以是虚轴，或者外部编码器主轴等。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Master
主轴号</td><td>指定的主轴编码</td><td>AXIS_REF_AXISID</td></tr>
</table>

Slave 从轴号	指定的从轴编码	AXIS_REF_AXISID
ContinuousUpdate 连续更新	当执行条件由 OFF 变 ON 时，该指令的输入参数的刷新将被写入	BOOL
RatioNumerator 齿轮分子	电子齿轮的分子(此参数不能为 0)，齿轮比为负时表示主从轴运动方向相反。齿轮比代表电子齿轮的齿数比。	DWORD
RatioDenominator 齿轮分母	电子齿轮的分母(此参数必须大于 0)	DWORD
MasterValSource 同步源选择	指定主轴的同步位置值的源头。	MC_SOURCE K0　设定值 K1　实际值
Acceleration 加速度	电子齿轮耦合时从轴对应终端执行机构的加速度,此参数总为正。单位：单元/秒 2。	REAL
Deceleration 减速度	电子齿轮耦合时从轴对应终端执行机构的减速度,此参数总为正。单位：单元/秒^2。	REAL
Jerk 阶跃	终端轴的加加速度,此参数总为正。单位：单元/秒^3.	REAL
输出参数		
InGear 同步信号	当主轴和从轴成功建立电子齿轮关系后，InGear 位被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Active 控制中	开始指令有效，则输出 ON。	BOOL
Aborted 执行中断	指令被其他指令中断时,则输出 ON。	BOOL

Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

只有对主轴和从轴分别使用 MC_Power 指令将各轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 Master，Slave 的值。

当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿，指令锁存 RatioNumerator、RatioDenominator，MasterValSource，Acceleration,Deceleration，Jerk 并触发电子齿轮功能。

在 ContinuousUpdate 为 True 时持续更新 RatioNumerator、RatioDenominator，MasterValSource，Acceleration,Deceleration，Jerk。

通过 Slave(从轴)指定动作对象轴，指定 RatioNumerator(齿轮比分子)、RatioDenominator(齿轮比分母) 、MasterValSource(位置类型选择) 、Acceleration(加速度) 、Deceleration(减速度)，进行齿轮动作。

可对 Master(主轴) 指定理论设定值和实际运行值。

开始动作后，Slave(从轴)以 Master(主轴)速度乘以齿轮比得到的速度为目标速度，进行加减速动作。

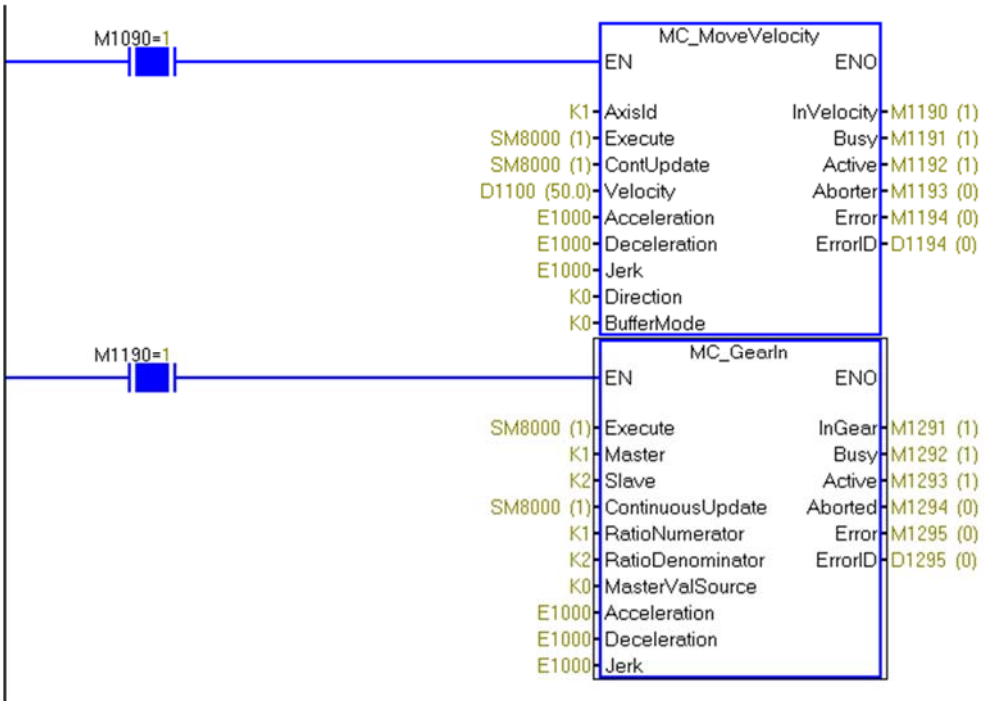
达到目标速度之前称为 Catching phase(追赶中)、达到后称为 InGear phase(齿轮同步中)。

MasterValSource 可分为 mcSetValue(K0)和 mcActualValue(K1)。mcSetValue 是使用理论计算的指令位置，mcSetValue 是使用实际计算的指令位置。

3、程序举例

轴 2 以轴 1 速度的一半同步运动

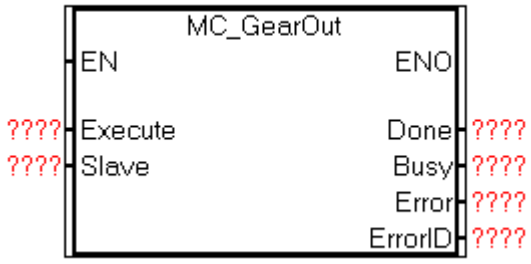
注：图中省略了轴 1 轴 2Power



电子齿轮功能块中以轴 1 为主轴，主轴以 50unit/s 速度匀速运动，轴 2 执行电子齿轮功能块，速度是主轴的速度乘以电子齿轮比即 50*1/2=25。

2.19.22 MC_GearOut/电子齿轮脱离指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于解除主从轴间已经建立的齿轮关系。齿轮关系解除后，从轴会以脱离前的速度继续运行。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Slave
从轴号</td><td>指定的从轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>输出参数</td><td></td><td></td></tr>
<tr><td>Done</td><td>当取消齿轮关联指令执行完</td><td>BOOL</td></tr>
</table>

完成位	成时 ，Done 位被置位	
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

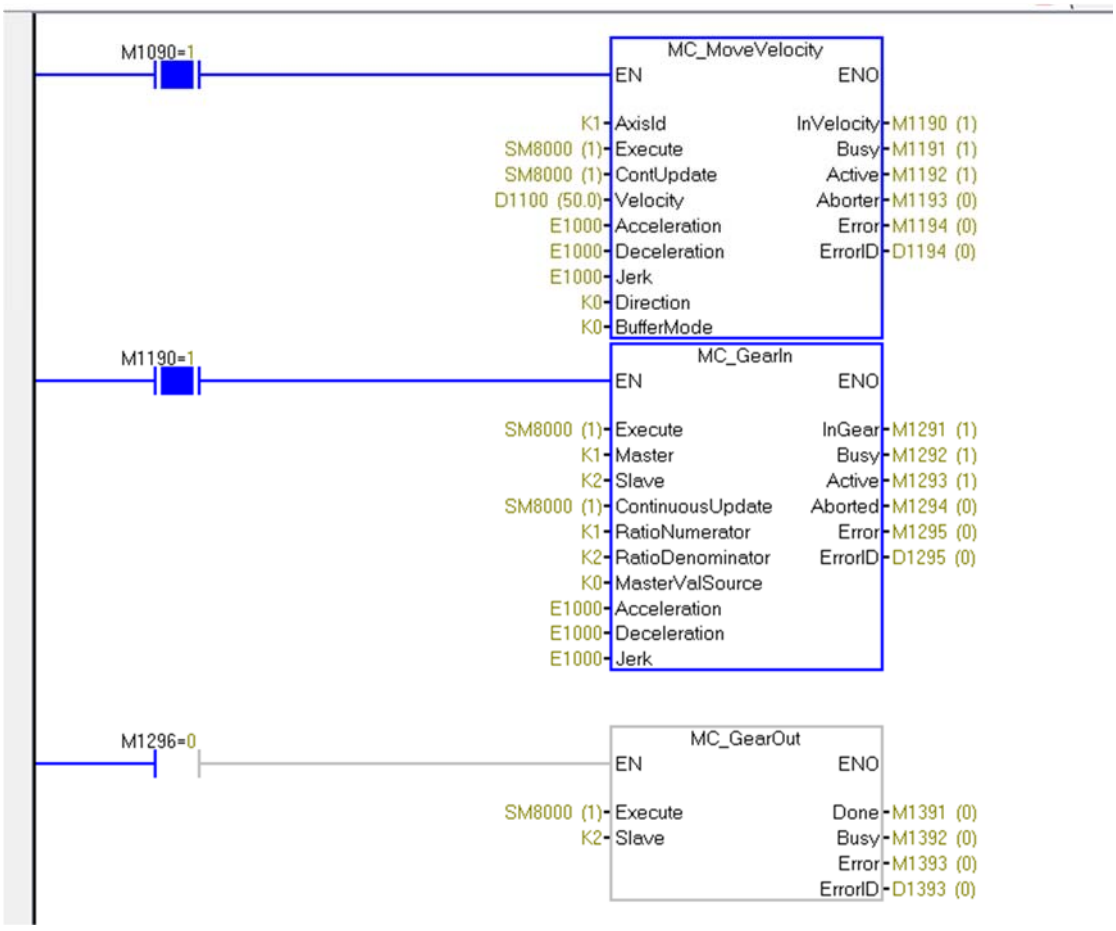
当轴 Slave 使用 MC_GearIn 指令进入同步状态后才能使用该指令

当 EN 复位时输出参数都会被复位

3、程序举例

轴 2 解除同步状态

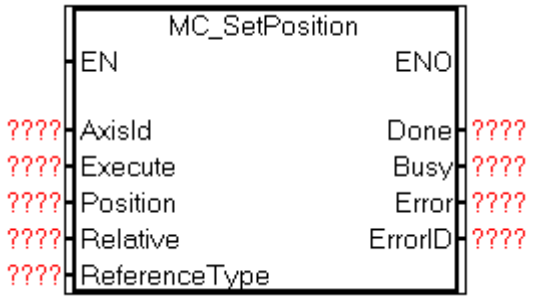
注：图中省略轴 1 轴 2Power



将 M1296 置位，执行 MC_GearOut 功能块，完成后解除轴 2 的同步状态，轴 2 以最后的解除速度做匀速运动。

2.19.23 MC_SetPosition/当前位置更改指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>改变坐标系统中轴的当前位置。</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisId</td><td>轴 ID</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Execute</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Position</td><td>设定位置（用户单位）</td><td>REAL</td></tr>
<tr><td>Relative</td><td>指定相对位置或绝
对位置
【FALSE:指定绝对位置
TRUE:指定相对位置】</td><td>BOOL</td></tr>
<tr><td>ReferenceType</td><td>保留参数</td><td>MC_SOURCE</td></tr>
<tr><td>输出参数</td><td></td><td></td></tr>
<tr><td>Done</td><td>动作完成时置 True</td><td>BOOL</td></tr>
<tr><td>Busy</td><td>当指令被触发执行时为 True</td><td>BOOL</td></tr>
<tr><td>Error</td><td>当指令错误发生时为 True</td><td>BOOL</td></tr>
<tr><td>ErrorID</td><td>当指令错误发生时记录错误码</td><td>WORD</td></tr>
</table>

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisId 的值。

当 EN 复位时输出参数都会被复位

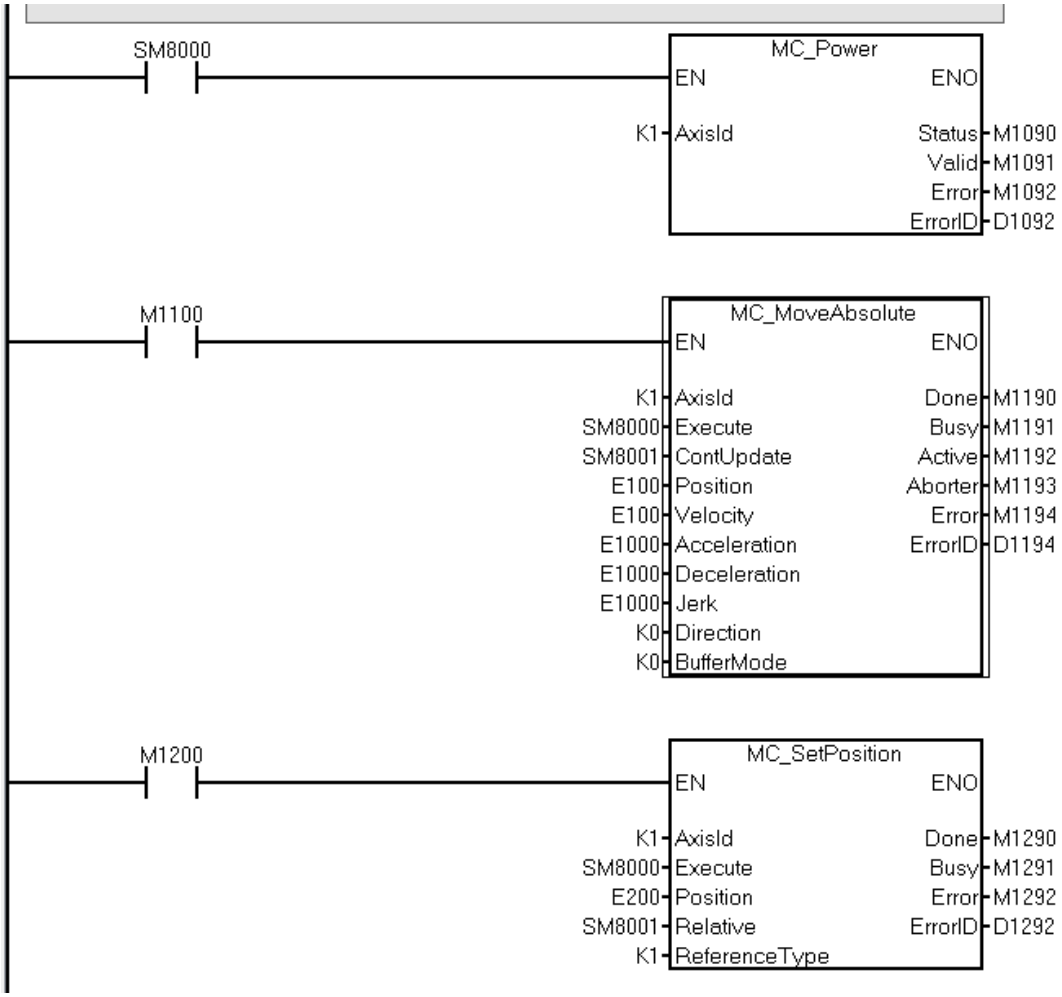
在 Execute 输入的上升沿，指令锁存 Position、Relative 并触发位置设定功能。

Relative=0 指定绝对位置，Relative=1 时指定相对位置

使用 MC_SetPosition 指令时需要保证轴处于静止状态，否则可能造成位置错乱。

3、程序举例

轴 1 走一段绝对位移后修改位置为 200



运行梯形图并打开 M1100 执行绝对位移后相关寄存器状态如下图所示:

软元件/标签	数据类型	当前值
M1100	BIT	1
M1200	BIT	0
SD104	FLOAT	100.0
M1190	BIT	1
M1290	BIT	0

打开 M1200，重新设置当前位置：

软元件/标签	数据类型	当前值
M1100	BIT	1
M1200	BIT	1
SD104	FLOAT	200.0
M1190	BIT	1
M1290	BIT	1

2.19.24 MC_AxisConfig/轴配置指令

1、指令详解

		修改轴配置
输入参数	解释	参数类型
AxisId	轴 ID	AXIS_REF_AXISID
Rotary	True 表示旋转轴，False 为直线轴	BOOL
Impulse	电机转一圈需要的脉冲个数	DWORD
Road	电机转一圈移动距离【User Unit】	REAL
MaxVelocity	最大速度【User Unit/s】	REAL
MaxAcceleration	最大加速度【User Unit/s^2】	REAL
MaxDeceleration	最大减速度【User Unit/s^2】	REAL
MaxJerk	最大加加速度【User Unit/s^3】	REAL
SCurve	加速曲线，True 表示 S 型，False 表示 T 型	BOOL
AxisModMax	旋转轴最大位置【度】	REAL
AxisModMin	旋转轴最小位置【度】	REAL
SwLimitPos	正向限位	BOOL
SwLimitNeg	反向限位	BOOL
SwLimitPosReversed	正向限位取反标志	BOOL
SwLimitNegReversed	反向限位取反标志	BOOL
ReferenceSignalReversed	参考/原点信号取反标志	BOOL
DogSignalReversed	DOG 信号取反标志	BOOL
输出参数		

Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

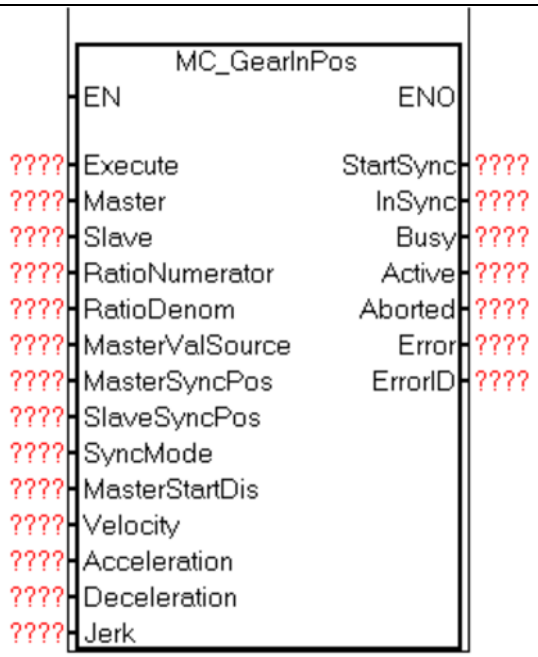
2、功能描述

只有在对应轴 MC_Power 指令断开时才可以调用本指令。

该指令配置的参数**不会**掉电保持。

2.19.25 MC_GearInPos/电子齿轮耦合位置指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于建立主从轴间的齿轮关系。建立齿轮关系时，可设定齿轮比，主轴从轴的同步位置等参数。齿轮关系建立后，从轴会以给定的比例关系跟随主轴运动，实现主从轴同步控制。主从轴可以是实轴，也可以是虚轴，或者外部编码器主轴等。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Master
主轴号</td><td>指定的主轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Slave
从轴号</td><td>指定的从轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>RatioNumerator</td><td>电子齿轮的分子(此参数不能</td><td>DWORD</td></tr>
</table>

齿轮分子	为 0），齿轮比为负时表示主从轴运动方向相反。齿轮比代表电子齿轮的齿数比。	
RatioDenominator 齿轮分母	电子齿轮的分母(此参数必须大于 0）	DWORD
MasterValSource 同步源选择	指定主轴的同步位置值的源头。	MC_SOURCE K0 设定值 K1 实际值
MasterSyncPos 主轴同步的起始位置	指定主轴与从轴 SyncIn 时刻的主轴位移	REAL
SlaveSyncPos 从轴同步的起始位置	指定主轴与从轴 SyncIn 时刻的从轴位移	REAL
SyncMode 同步的方式	采用何种方式进行同步	MC_SyncMode K0 最短距离 K1 加速 K2 减速
MasterStartDis 主轴跟踪距离	当主轴的位移大于 MasterSyncPos 与 MasterStartDis 的差值时从轴开始运行	REAL
Velocity 从轴运行的最大速度	指定从轴运行的最大速度	REAL
Acceleration 加速度	电子齿轮耦合时从轴对应终端执行机构的加速度,此参数总为正。单位：单元/秒^2。	REAL
Deceleration 减速度	电子齿轮耦合时从轴对应终端执行机构的减速度,此参数总为正。单位：单元/秒^2。	REAL
Jerk 阶跃	终端轴的加加速度,此参数总为正。单位：单元/秒^3.	REAL
输出参数		
StartSync	从轴开始运行的时刻置位该	BOOL

开始跟踪信号	信号	
InSync 同步信号	当主轴和从轴成功建立电子齿轮关系后，InSync 位被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Active 控制中	开始指令有效，则输出 ON。	BOOL
Aborted 执行中断	指令被其他指令中断时，则输出 ON。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

只有对主轴和从轴分别使用 MC_Power 指令将各轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 Master，Slave 的值。

当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿，指令锁存 RatioNumerator、RatioDenominator，MasterValSource，
MasterSyncPos,SlaveSyncPos,SyncMode,MasterStartDis,Acceleration,Deceleration，Jerk 并触发电子齿轮位移功能。

通过 Slave(从轴)指定动作对象轴，指定 RatioNumerator(齿轮比分子)、RatioDenominator(齿轮比分母) 、MasterValSource(位置类型选择) 、MasterSyncPos（主轴同步时位移）、SlaveSyncPos（从轴同步时位移）、SyncMode（同步模式）、MasterStartDis（从轴开始运行时主轴位移）、Acceleration(加速度) 、Deceleration(减速度)，进行齿轮动作。

可对 Master(主轴) 指定理论设定值和实际运行值。

开始动作后，Slave(从轴)以 Master(主轴)速度乘以齿轮比得到的速度为目标速度，进行加减速动作。

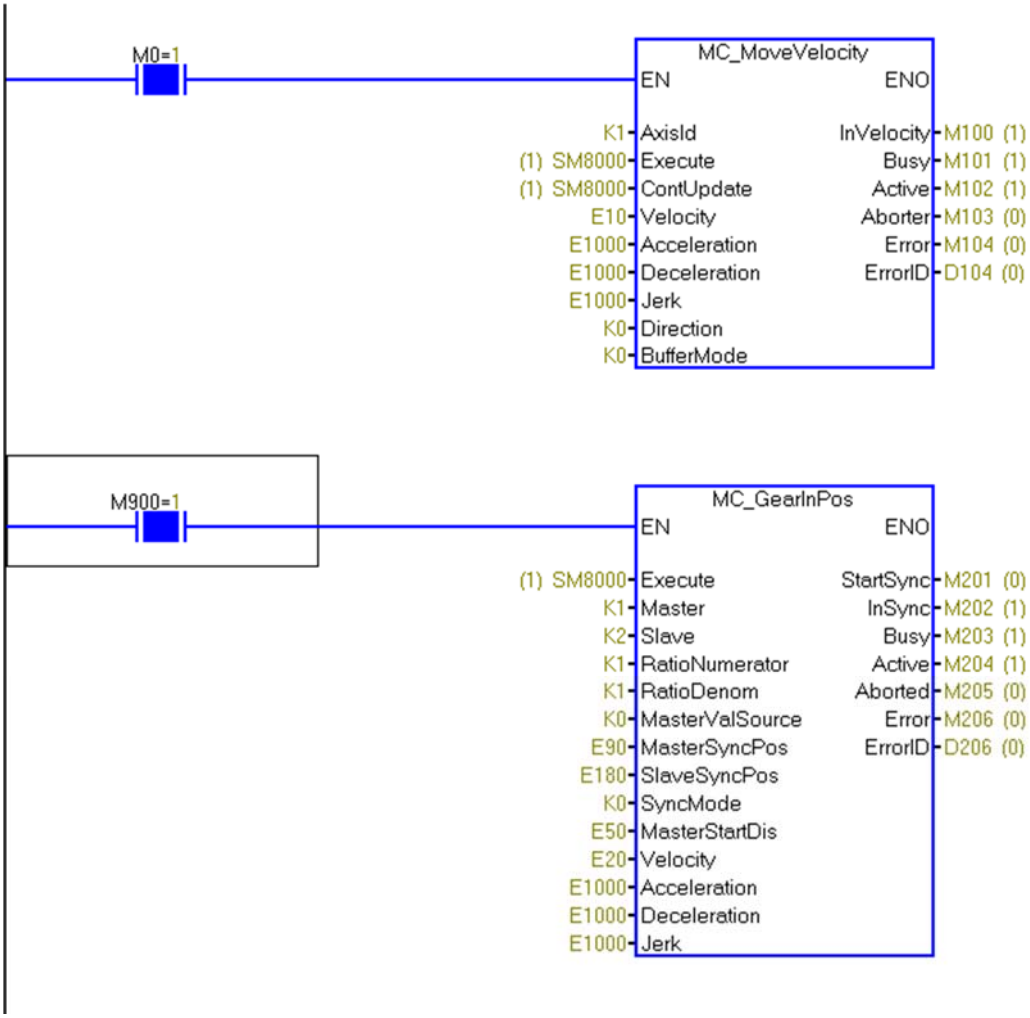
从轴开始运行的时候 StartSync(开始同步)、位置和速度达到后称为 InSync(齿轮同步中)。

MasterValSource 可分为 mcSetValue(K0)和 mcActualValue(K1)。mcSetValue 是使用理论计算的指令位置，mcSetValue 是使用实际计算的指令位置。此指令要求主轴位移必须是正数。

3、程序举例

轴 2 以轴 1 相同速度的同步运动

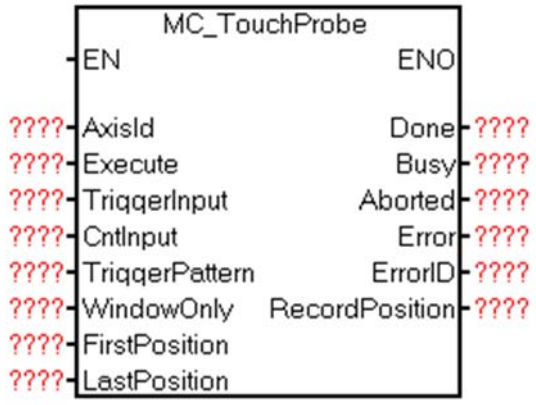
注：图中省略了轴 1 轴 2Power



电子齿轮位移功能块中以轴 1 为主轴，主轴以 10unit/s 速度匀速运动，轴 2 执行电子齿轮位移功能块，速度是主轴的速度乘以电子齿轮比即 10*1/1=10，当主轴位移大于等于 40(MasterSyncPos-MasterStartDis)时从轴开始运动，此时 StartSync 置位，当主轴位置为 90（MasterSyncPos）从轴位置为 180（SlaveSyncPos）时达到同步状态，此时 Insync 置位。

2.19.26 MC_TouchProbe/探针指令

1、指令详解

		此指令根据外部 IO 的触发信号记录轴当时的位置值
参数名称	参数解释	参数类型
输入参数		
AxisId	轴 ID	AXIS_REF_AXISID
Execute	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
TriggerInput*	对应的输入 IO 口	BOOL
CntInput	对应的计数器（C235-C255）	DWORD
TriggerPattern	触发方式选择	TriggerPattern K0 上升沿 K1 下降沿
WindowsOnly	阈值窗口开启	BOOL
FirstPosition	开始阈值	REAL
LastPosition	结束阈值	REAL
输出参数		
Done	触发后记录轴位置后置位该信号	BOOL
Busy	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Aborted	指令被其他指令中断时，则输出 ON。	BOOL
Error	检测到有错误，Error 位被置位。	BOOL
ErrorID	错误代码。	WORD

RecordPosition	记录触发时刻的轴位置值	REAL

*TriggerInput：只能填 X0~X7，使用 X10~X17 时需要在 EI 指令前打开对应的中断重定向功能，参考 3.2.4。此时 X0~X7 的 MC_TouchProbe 功能就会重定向到 X10~X17（TriggerInput 依然填 X0~X7）。

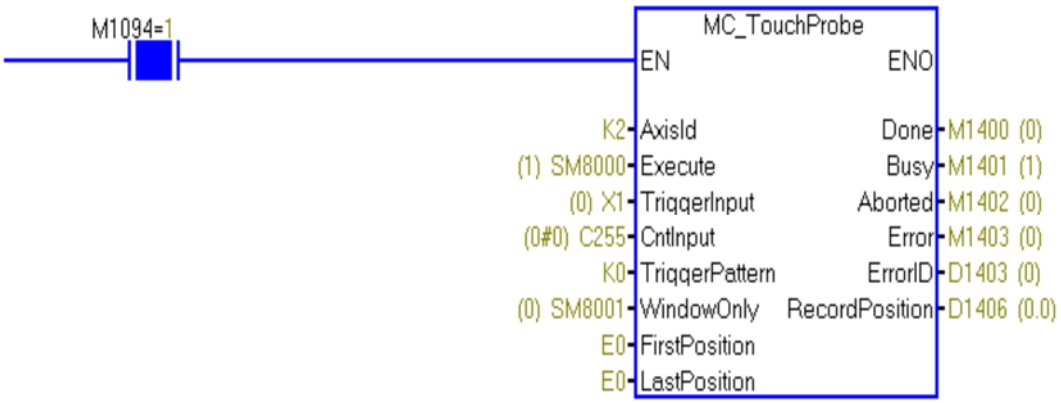
2、功能描述

此功能块可以通过外部 IO 信号来记录轴当前的位置值，在 WindowsOnly 无效的时候，直接记录触发时刻的位置值，在 WindowsOnly 有效的时候，仅在阈值的开始及结束数值之间的时候触发，特别提示输出参数 RecordPosition 的软元件必须是偶数地址。

3、程序举例

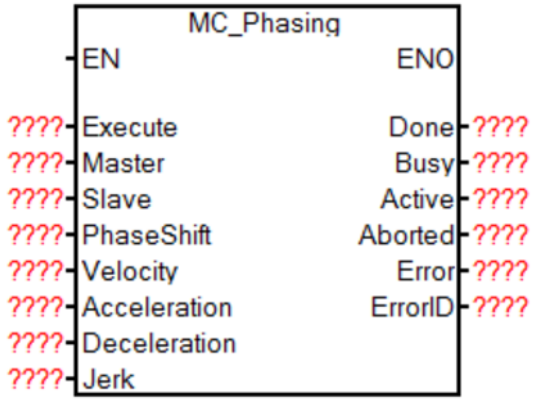
轴 2 的探针功能

注：图中省略了轴 2Power



2.19.27 MC_Phasing/主轴相位补偿指令

1、指令详解



控制指定轴依照 PhaseShift 相位补偿量进行偏移

输入参数	解释	参数类型
Execute	当执行条件由 OFF 变 ON 时，执	BOOL

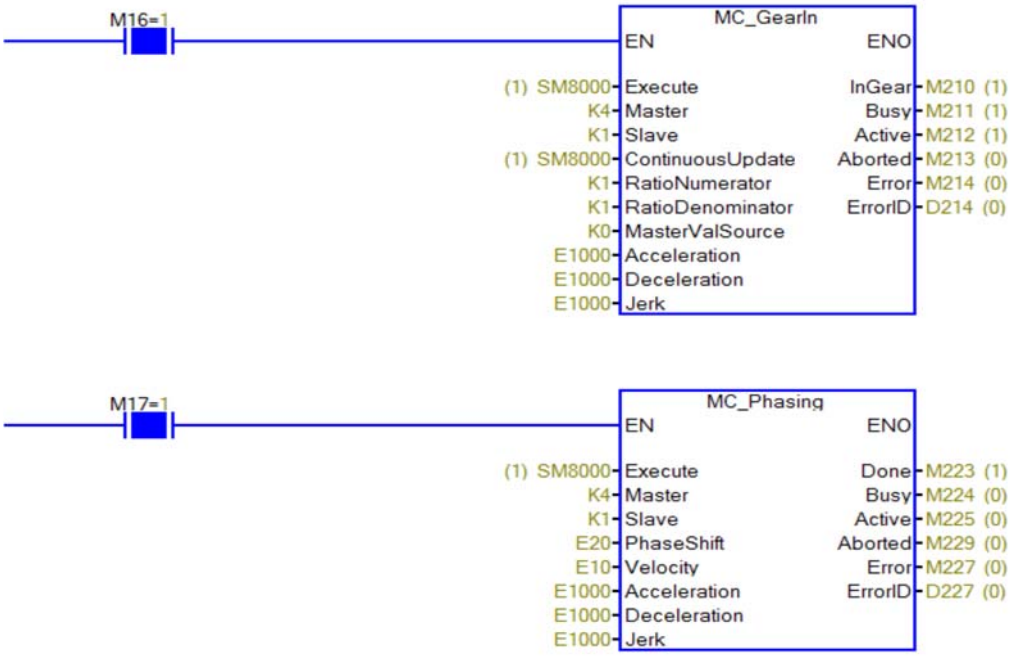
	行该指令	
Master	指定的主轴编码	AXIS_REF_AXISID
Slave	指定的从轴编码	AXIS_REF_AXISID
PhaseShift	相位补偿量	REAL
Velocity	目标速度（用户单位/秒）	REAL
Acceleration	加速度（用户单位/秒 2）	REAL
Deceleration	减速度（用户单位/秒 2）	REAL
Jerk	加加速度；跃度 （用户单位/秒 3）	REAL
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Active	当指令运行时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD

2、功能描述

该指令仅在 MC_GearIn,MC_GearInPos,MC_CamIn 功能块中配合使用

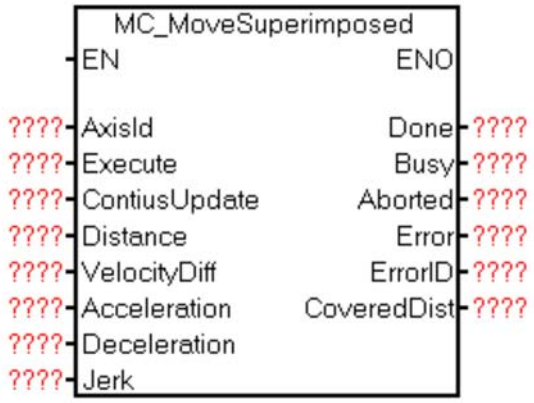
3、程序举例

注：图中省略了轴 1 轴 4Power



2.19.28 MC_MoveSuperimposed/位移叠加指令

1、指令详解

<table>
<tr><td>
</td><td colspan="2">控制指定轴在当前运动状态下按照给定的运行参数叠加一段给定位移</td></tr>
<tr><td>输入参数</td><td>解释</td><td>参数类型</td></tr>
<tr><td>AxisId</td><td>轴 ID</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Execute</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>ContinusUpdate</td><td>为 True 时，持续更新目标速度</td><td>BOOL</td></tr>
<tr><td>Distance</td><td>叠加位移（用户单位）</td><td>REAL</td></tr>
<tr><td>Velocity</td><td>叠加速度（用户单位/秒）</td><td>REAL</td></tr>
<tr><td>Acceleration</td><td>加速度（用户单位/秒 2）</td><td>REAL</td></tr>
</table>

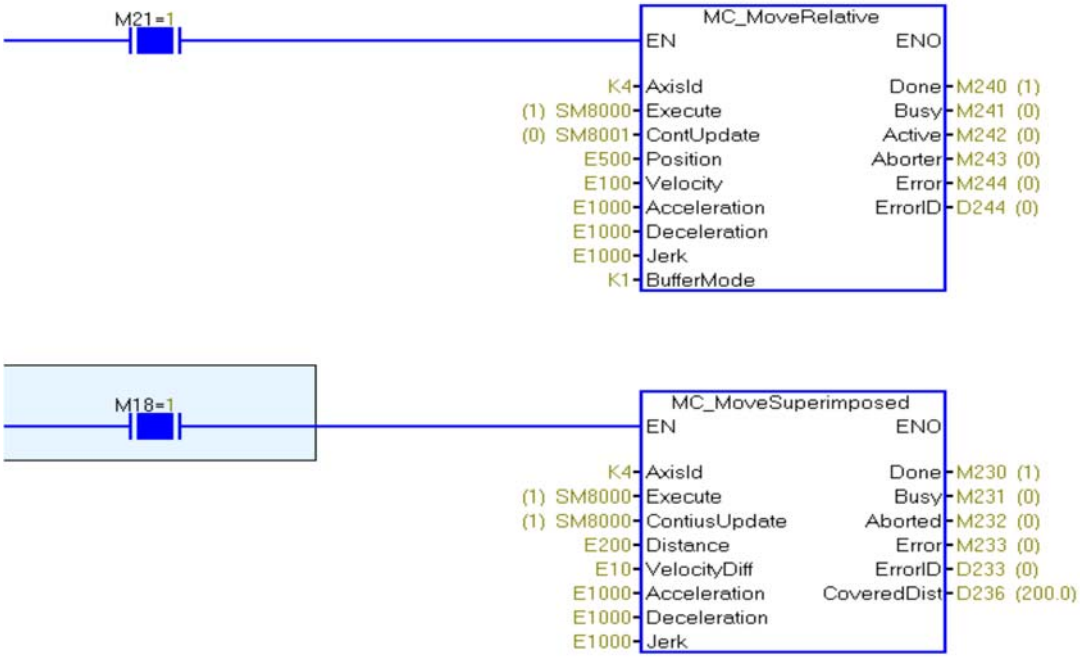
Deceleration	减速度（用户单位/秒 2）	REAL
Jerk	加加速度；跃度 （用户单位/秒 3）	REAL
输出参数		
Done	动作完成时置 True	BOOL
Busy	当指令被触发执行时为 True	BOOL
Aborted	当指令被中断时为 True	BOOL
Error	当指令错误发生时为 True	BOOL
ErrorID	当指令错误发生时记录错误码	WORD
CoveredDist	已经叠加运动的位移	REAL

2、功能描述

只有使用 MC_Power 指令将轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 AxisId 的值，运行此指令的时候，不会终止前一个指令执行，两条指令同时执行，距离，速度，加减速会实时叠加。当其中一个指令执行完毕，将不再叠加速度，加减速，另一个指令仍然独立运行。

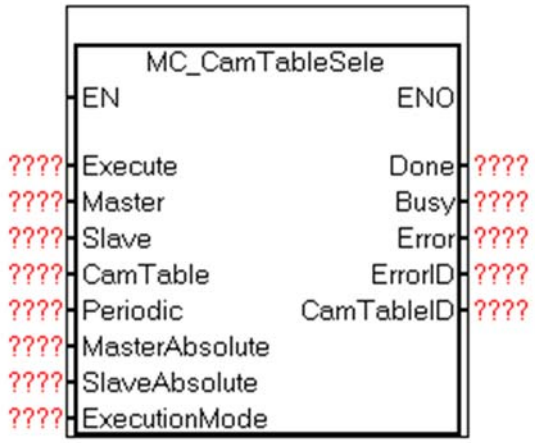
3、程序举例



运行完成后的 SD404 寄存器数值为 700，即 relative 模块的 500 与 superimposed 模块的 200 之和。

2.19.29 MC_CamTableSelect/凸轮表选择指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于选择相应的凸轮表</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Master
主轴号</td><td>指定的主轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Slave
从轴号</td><td>指定的从轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>CamTable
凸轮表属性</td><td>设置凸轮表的 ID，点数，主轴最大位移等属性</td><td>DWORD</td></tr>
<tr><td>Periodic
周期运行</td><td>凸轮表是否周期运行，FALSE 表示只运行一次，TRUE 表示周期运行</td><td>BOOL</td></tr>
<tr><td>MasterAbsolute
主轴起始位置方式指定</td><td>指定主轴位移按绝对位置还是相对位置，TRUE 代表按绝对位置，FALSE 代表按相对位置</td><td>BOOL</td></tr>
<tr><td>SlaveAbsolute
从轴起始位置方式指定</td><td>指定从轴位移按绝对位置还是相对位置，TRUE 代表按绝对位置，FALSE 代表按相对位置</td><td>BOOL</td></tr>
</table>

ExecutionMode 执行的方式	采用何种方式执行	MC_EXECUTION_MODE K0 立即执行 K1 缓冲执行
输出参数		
Done 完成信号	当凸轮表建立成功后，Done 被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD
CamTableID 凸轮表 ID 号	输出该凸轮表的 ID 号	DWORD

2、功能描述

只有对主轴和从轴分别使用 MC_Power 指令将各轴切换到使能状态才可以调用本指令。

在 EN 后请不要改变 Master，Slave 的值。

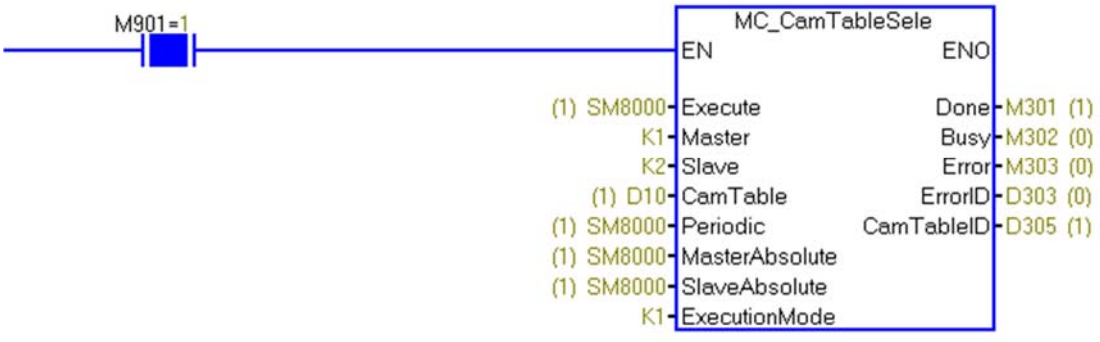
当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿，指令锁存 CamTable 、Period 、MasterAbsolute 、SlaveAbsolute 、ExecutionMode 并触发凸轮表选择功能。

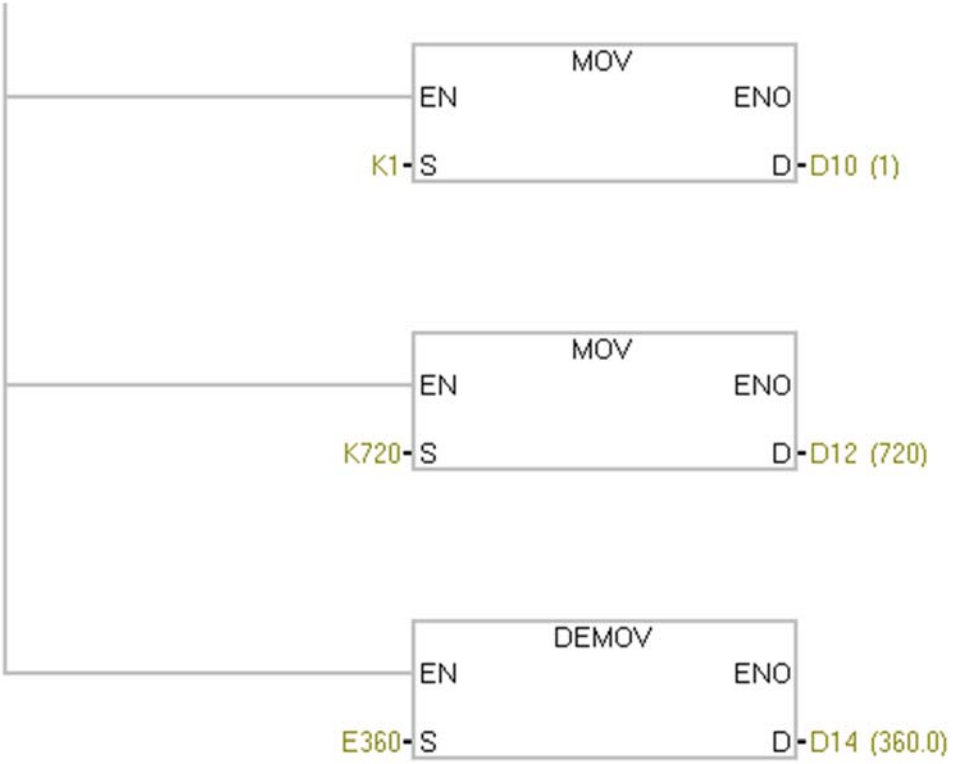
功能运行成功后置位 Done 信号，同时输出对应的凸轮表 ID。

3、程序举例

注：图中省略了轴 1 轴 2Power



注意输入参数 Camtable 是占用三个连续的 DWORD 地址，分别设置凸轮表的 ID 号，凸轮表的总点数，凸轮表中主轴的最大位移，实例如下



，这些参数也是根据用户自己在 E-CAM 编辑器中添加的 ECAM 表所得，上图对应的 ECAM 表如下

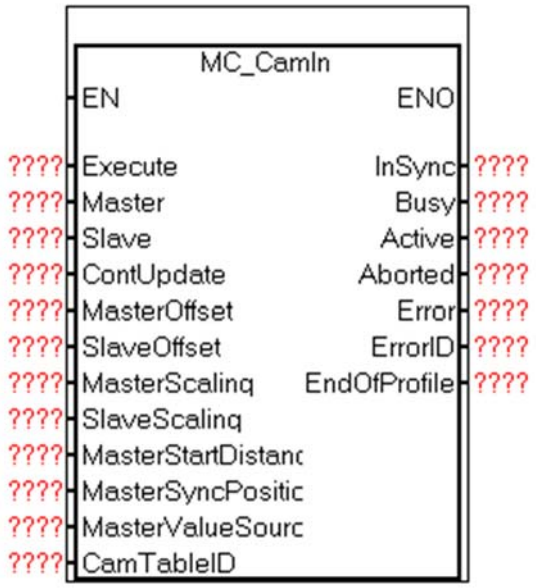


	主轴	从轴	速度	加速度	节点	相节距
1	0.0	0.0				
2	50.0000	80.0000	☑ 10.0000	☐ 0.0000	五阶多项式	0.50
3	100.0000	130.0000	☐ 18.0000	☐ 0.0000	直线	0.50
4	150.0000	210.0000	☑ 0.0000	☐ 0.0000	五阶多项式	0.50
5	360.0000	0.0000	☐ 0.0000	☐ 0.0000	五阶多项式	0.50

D12 对应的凸轮表总点数是 720，表示主轴最大的位移 360 除以相节距 0.5，每张凸轮表总点数不得超过 1024 个点。

2.19.30 MC_CamIn/凸轮表啮合指令

1、指令详解

		此指令用于执行凸轮表啮合的功能
参数名称	参数解释	参数类型
输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Master 主轴号	指定的主轴编码	AXIS_REF_AXISID
Slave 从轴号	指定的从轴编码	AXIS_REF_AXISID
ContinuousUpdate 连续更新	当执行条件由 OFF 变 ON 时，该指令的输入参数的刷新将被写入	BOOL
MasterOffset 主轴偏移	以指定的偏移值移动主轴的相位	REAL
SlaveOffset 从轴偏移	以指定的偏移值移动从轴的相位	REAL
MasterScaling 主轴比例	以指定的比例放大/缩小主轴的相位	REAL

SlaveScaling 从轴比例	以指定的比例放大/缩小从轴的相位	REAL
MasterStartDistance 主轴跟踪距离	当主轴的位移大于 MasterSyncPosition 与 MasterStartDistance 的差值时从轴开始运行	REAL
MasterSyncPosition 主轴同步位移	从轴开始同步的时候对应主轴位移	REAL
MasterValueSource 同步源选择	指定主轴的同步位置值的源头。	MC_SOURCE K0 设定值 K1 实际值
CamTableID 凸轮表 ID 号	凸轮表 ID 号	DWORD
输出参数		
InSync 同步信号	当主轴和从轴成功建立电子凸轮关系后，InSync 位被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Active 控制中	开始指令有效，则输出 ON。	BOOL
Aborted 执行中断	指令被其他指令中断时,则输出 ON。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD
EndOfProfile 凸轮表结束	凸轮表结束标志	BOOL

2、功能描述

只有对主轴和从轴分别使用 MC_Power 指令将各轴切换到使能状态，同时必须先执行凸轮表选择指令，有可用的凸轮表 ID 后才可以调用本指令。

在 EN 后请不要改变 Master，Slave 的值。

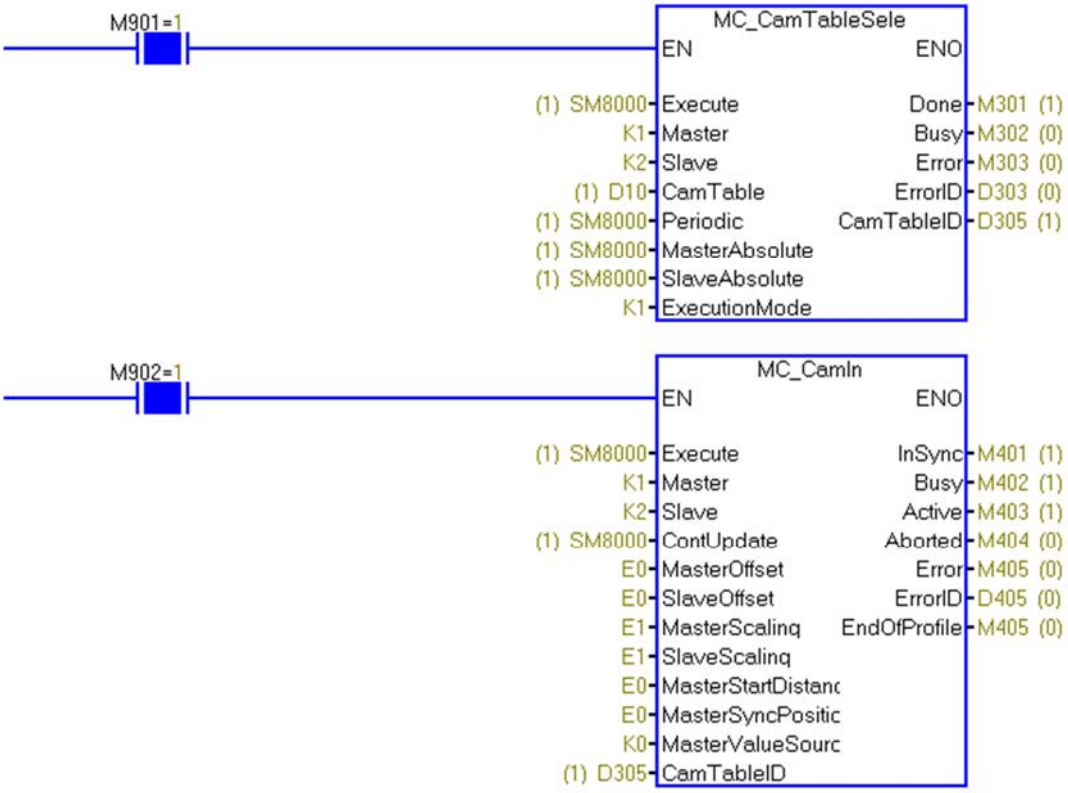
当 EN 复位时输出参数都会被复位

在 Execute 输入的上升沿，指令锁存 MasterOffset、SlaveOffset、MasterScaling、SlaveScaling 、MasterStartDistance 、MasterSyncPosition 、MasterValueSource,CamTableID 并触发凸轮啮合功能。

功能运行成功后置位 Insync 信号，运行到凸轮表末尾的时候会置位 EndOfProfile 信号。

4、程序举例

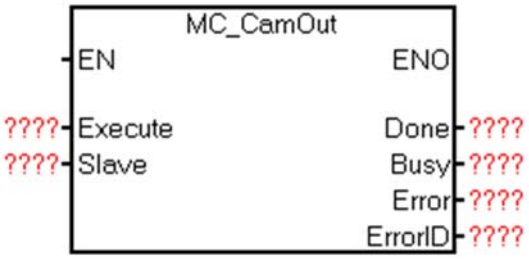
注：图中省略了轴 1 轴 2Power



必须先执行 MC_CamTableSelect 指令，然后执行 MC_CamIn

2.19.31 MC_CamOut/电子凸轮脱离指令

1、指令详解

		此指令用于解除主从轴间已经建立的凸轮关系。凸轮关系解除后，从轴会以脱离前的速度继续运行。
参数名称	参数解释	参数类型
输入参数		
Execute 启动信号	当执行条件由 OFF 变 ON 时，执行该指令	BOOL
Slave 从轴号	指定的从轴编码	AXIS_REF_AXISID
输出参数		
Done 完成位	当取消齿轮关联指令执行完成时 ，Done 位被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

当轴 Slave 使用 MC_CamIn 指令进入同步状态后才能使用该指令

当 EN 复位时输出参数都会被复位

3、程序举例

轴 2 解除同步状态

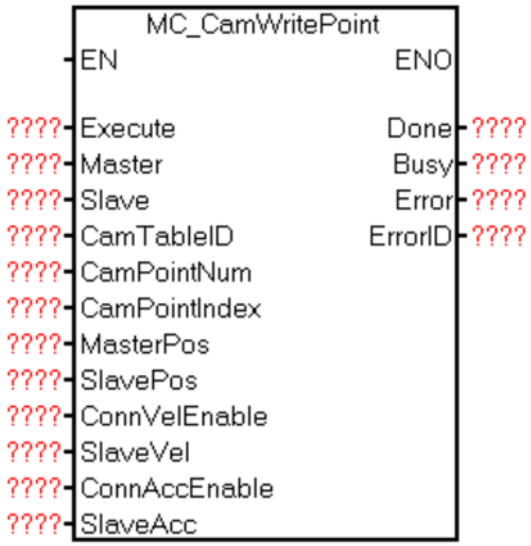
注：图中省略轴 1 轴 2Power



将 M904 置位，执行 MC_CamOut 功能块，完成后解除轴 2 的同步状态，轴 2 以最后的解除速度做匀速运动。

2.19.32 MC_CamWritePoint/电子凸轮表关键点设置指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于修改凸轮表关键点。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>Master
主轴号</td><td>指定的主轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>Slave
从轴号</td><td>指定的从轴编码</td><td>AXIS_REF_AXISID</td></tr>
<tr><td>CamTableID
凸轮表 ID</td><td>凸轮表 ID 号</td><td>tMC_CAM_ID</td></tr>
</table>

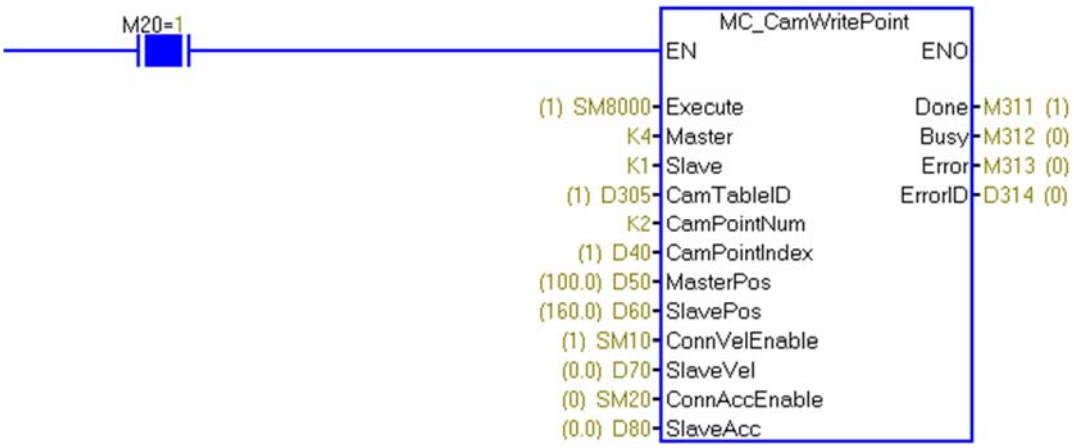
CamPointNum 修改的总关键点个数	需要修改的关键点个数	DWORD
CamPointIndex 关键点序号	第几个关键点，默认从 1 开始，不计算（0,0）	DWORD[]
MasterPos 主轴位置	设置的关键点中主轴位置	REAL[]
SlavePos 从轴位置	设置的关键点中从轴位置	REAL[]
ConnectingVelEnable	关键点是否使用连接速度	BOOL[]
SlaveVel 从轴速度	关键点连接速度大小	REAL[]
ConnectingAccEnable	关键点是否使用连接加速度	BOOL[]
SlaveAcc 从轴加速度	关键点连接加速度大小	REAL[]
输出参数		
Done 完成位	当更新指令执行完成时,Done 位被置位	BOOL
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD

2、功能描述

当 EN 复位时输出参数都会被复位，CamPointNum 设置的数值必须与后面的数组个数一致，否则凸轮的轨迹会不正确。当曲线类型选择直线的时候，不能选择连接速度和连接加速度；当曲线类型选择三次多项式的时候，只能选择连接速度，不能选择连接加速度；当曲线类型选择五次多项式的时候，可以选择连接速度，也可以选择连接加速度。

3、程序举例

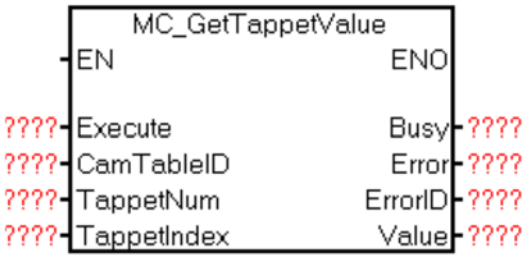
注：图中省略轴 1 轴 2Power



当从轴正在执行一个凸轮表的时候，使能 M20,将会改变凸轮表的两个关键点数据，然后以更改后的数据进行同步运动。

2.19.33 MC_GetTappetValue/读取挺杆点状态指令

1、指令详解

<table>
<tr><td colspan="2">
</td><td>此指令用于获取凸轮挺杆点的状态。</td></tr>
<tr><td>参数名称</td><td>参数解释</td><td>参数类型</td></tr>
<tr><td>输入参数</td><td></td><td></td></tr>
<tr><td>Execute
启动信号</td><td>当执行条件由 OFF 变 ON 时，执行该指令</td><td>BOOL</td></tr>
<tr><td>CamTableID
凸轮表 ID</td><td>凸轮表 ID 号</td><td>tMC_CAM_ID</td></tr>
<tr><td>TappetNum
挺杆点个数</td><td>需要读取的挺杆点总数</td><td>DWORD</td></tr>
<tr><td>TappetIndex</td><td>需要读取的挺杆点序号</td><td>DWORD[]</td></tr>
</table>

挺杆点的序号		
输出参数		
Busy 执行忙位	模块正在工作，则输出 ON，模块空闲输出 OFF。	BOOL
Error 错误位	检测到有错误，Error 位被置位。	BOOL
ErrorID 错误代码	错误代码。	WORD
Value 挺杆状态	挺杆的状态	BOOL[]

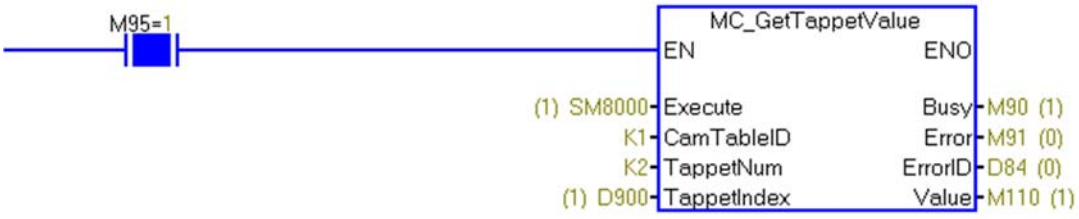
3、功能描述

当 EN 复位时输出参数都会被复位，TappetNum 设置的数值必须与后面的数组个数一致，输出的 Value 数组的元素也应和 TappetNum 对应。

挺杆点在建立凸轮表的时候可以根据需求自由设定，选取合适的挺杆状态，在主轴正向的位移到达设定挺杆点坐标，执行对应的状态。凸轮转一圈通过终点的时候会将输出状态清除，现仅支持主轴为旋转轴正向旋转的情况。

3、程序举例

注：图中省略轴 1 轴 2Power

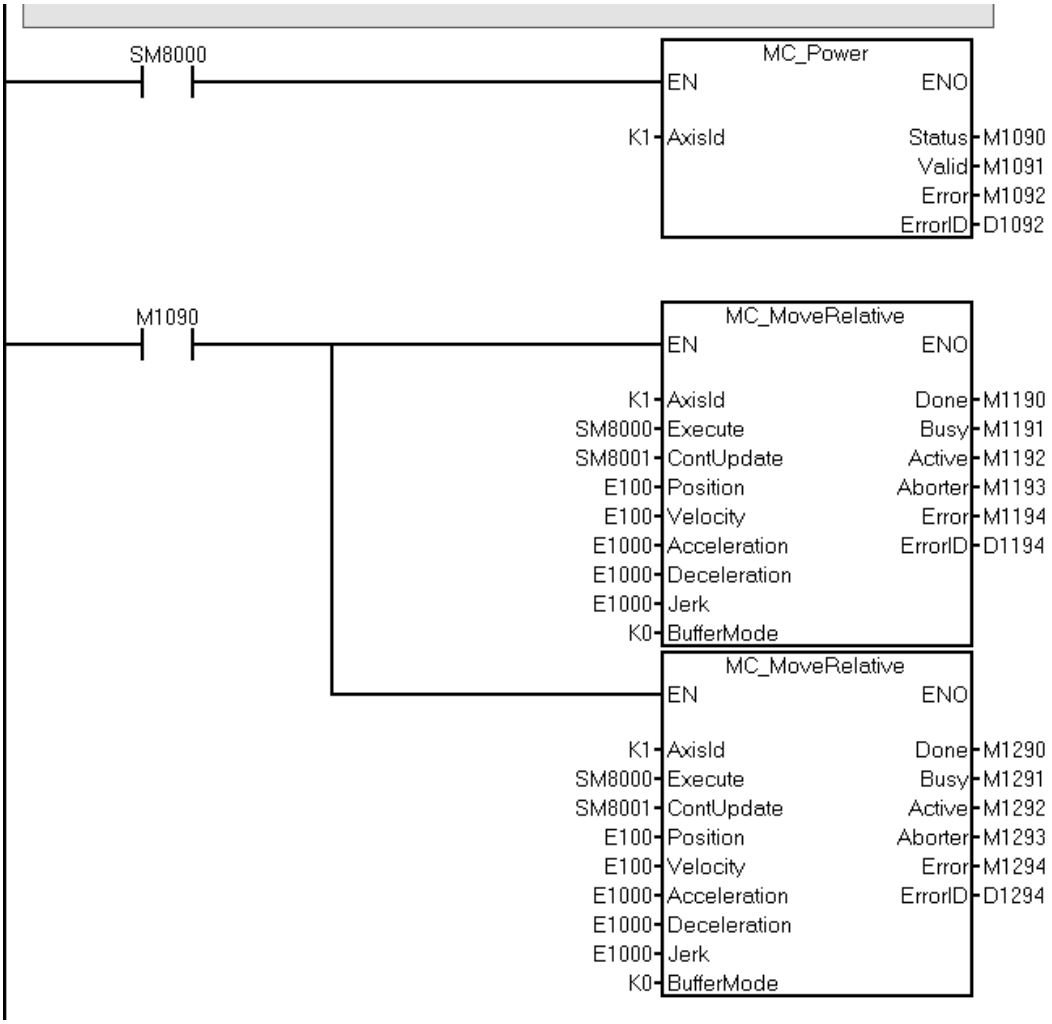


BufferMode 和连续插补功能说明

①BufferMode=0 打断模式

该模式会停止正在运行的任务立刻执行新任务，如果存在缓冲任务则会把缓冲任务全部清空。

示例：轴 1 使用相对位移打断前一段相对位移



运行前相关寄存器初始状态如下：

软元件/标签	数据类型	当前值
SM8000	BIT	0
M1090	BIT	0
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0
M1290	BIT	0
M1291	BIT	0
M1292	BIT	0
M1293	BIT	0
M1294	BIT	0
D1294	WORD	0
SD104	FLOAT	0.0

运行后相关寄存器变化如下:

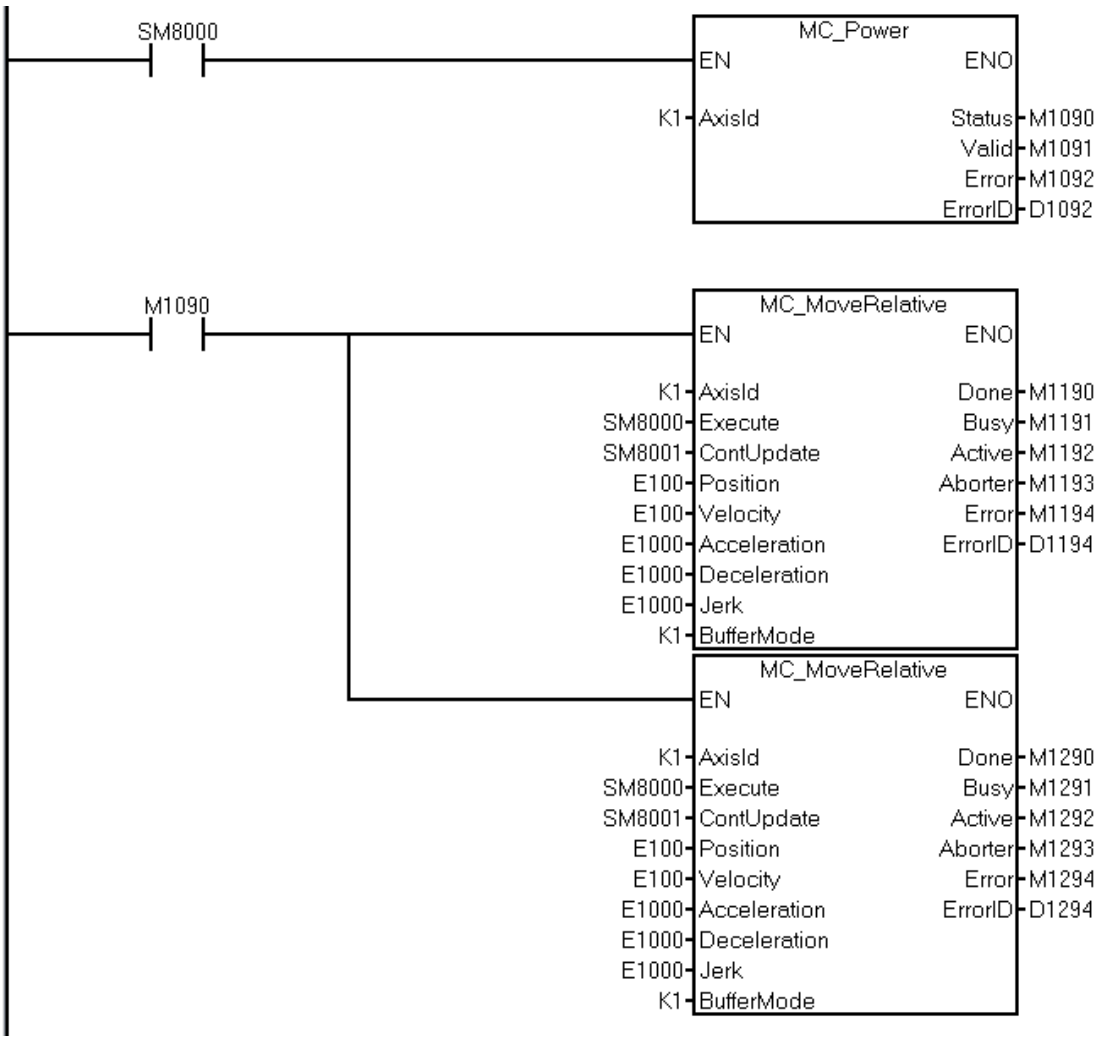
软元件/标签	数据类型	当前值
SM8000	BIT	1
M1090	BIT	1
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	1
M1194	BIT	0
D1194	WORD	0
M1290	BIT	1
M1291	BIT	0
M1292	BIT	0
M1293	BIT	0
M1294	BIT	0
D1294	WORD	0
SD104	FLOAT	100.0

第一条 MC_MoveRelative 指令被打断，停止状态标志（M1193）被置位，完成了第二条 MC_MoveRelative，完成标志（M1290）置位。

②BufferMode=1 缓冲模式

该模式会将新任务添加到等待队列中，等到前面添加的任务执行完成才会执行。执行后会从队列中清除。

示例：轴 1 连续添加两条相对位移指令到任务队列中



运行前相关寄存器初始状态如下：

软元件/标签	数据类型	当前值
SM8000	BIT	0
M1090	BIT	0
M1190	BIT	0
M1191	BIT	0
M1192	BIT	0
M1193	BIT	1
M1194	BIT	0
D1194	WORD	0
M1290	BIT	0
M1291	BIT	0
M1292	BIT	0
M1293	BIT	0
M1294	BIT	0
D1294	WORD	0
SD104	FLOAT	0.0

运行后相关寄存器变化如下：

软元件/标签	数据类型	当前值
SM8000	BIT	1
M1090	BIT	1
M1190	BIT	1
M1191	BIT	0
M1192	BIT	0
M1193	BIT	0
M1194	BIT	0
D1194	WORD	0
M1290	BIT	1
M1291	BIT	0
M1292	BIT	0
M1293	BIT	0
M1294	BIT	0
D1294	WORD	0
SD104	FLOAT	200.0

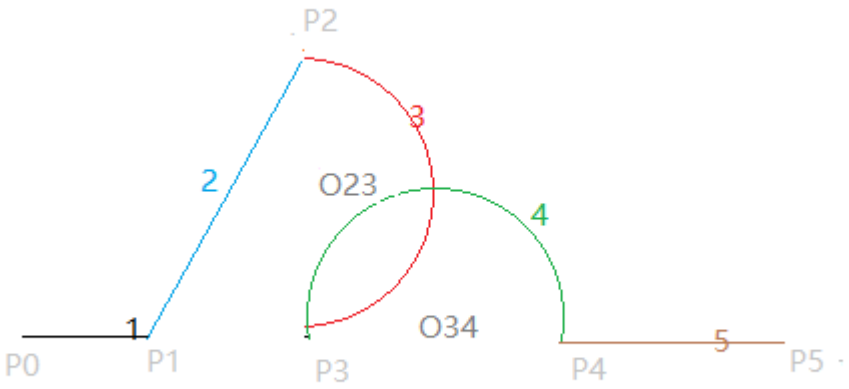
两条指令都执行完成（M1190 和 M1290 都置位），当前位置 SD104 的值位两段位移之和。

③BufferMode=3 连续插补模式以前一个插补模块的速度过渡

④BufferMode=4 连续插补模式以后一个插补模块的速度过渡

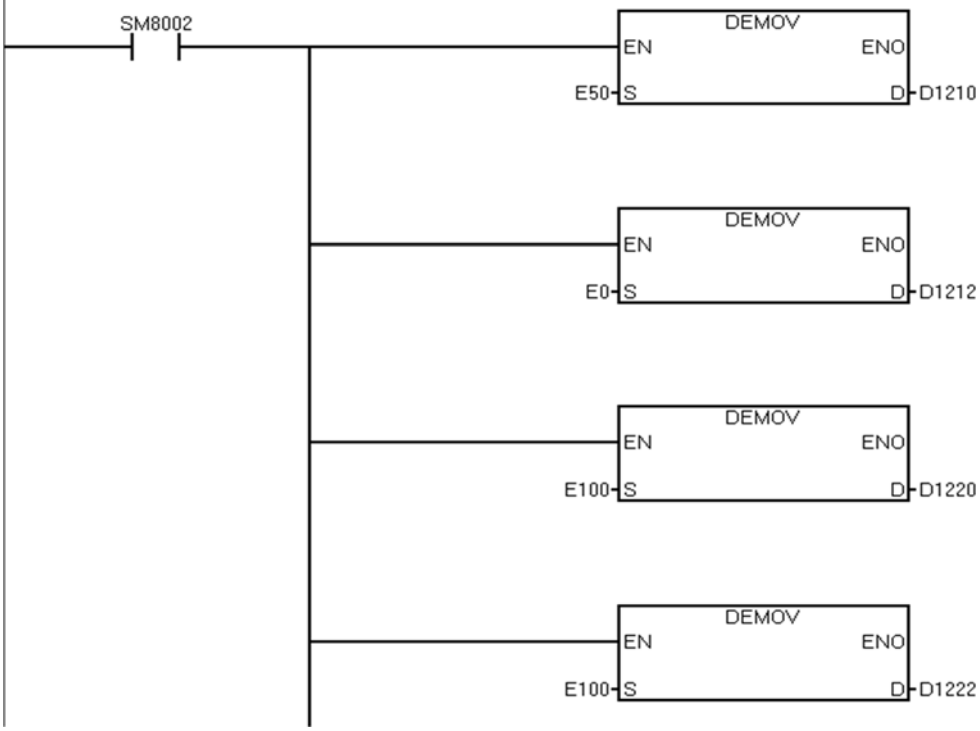
备注：当连续插补的时候前后两个轨迹导致电机反向的时候，推荐使用圆弧过渡模式，且仅有绝对直线插补指令和绝对圆弧插补指令的圆心模式支持圆弧过渡，圆弧过渡的时候会根据实际情况（不一定会添加成功）在两段位移间加入过度曲线，当第一个插补模块设置的过渡方式是 MOVE_BM_BLENDINGPREVIOUS 时候会以当前本身模块设定的速度过渡。相对直线插补指令和相对圆弧插补指令不支持连续插补，仅支持平面 2 轴连续插补。

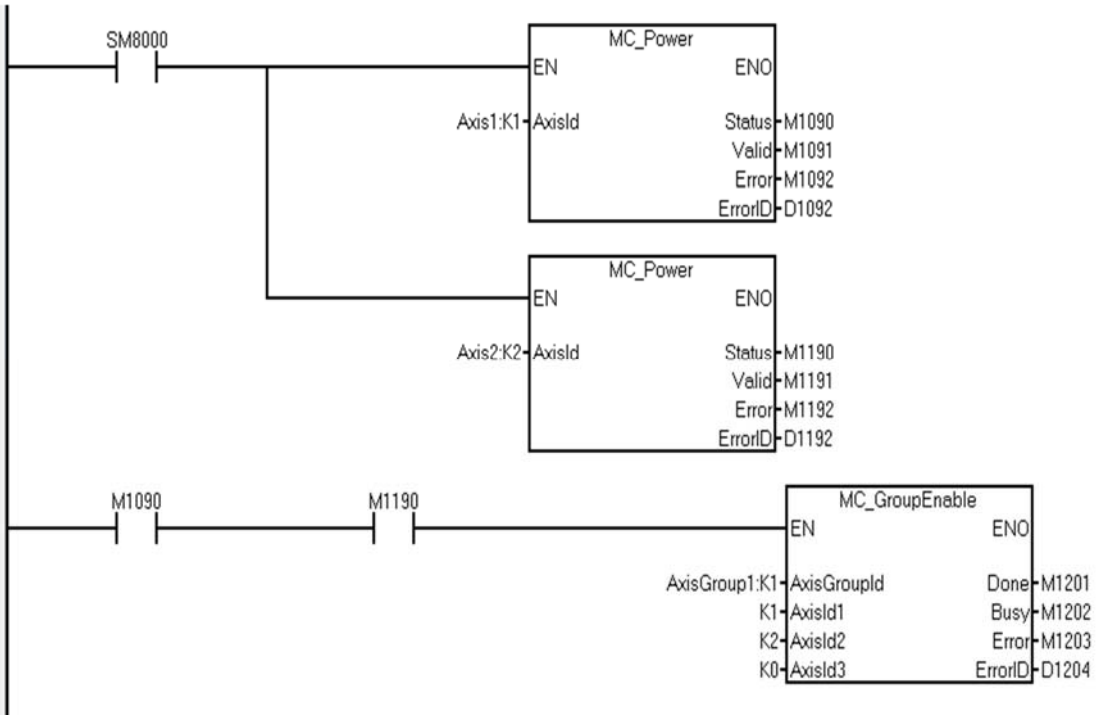
示例：使用连续插补完成下图所示轨迹（省略了部分赋值语句）

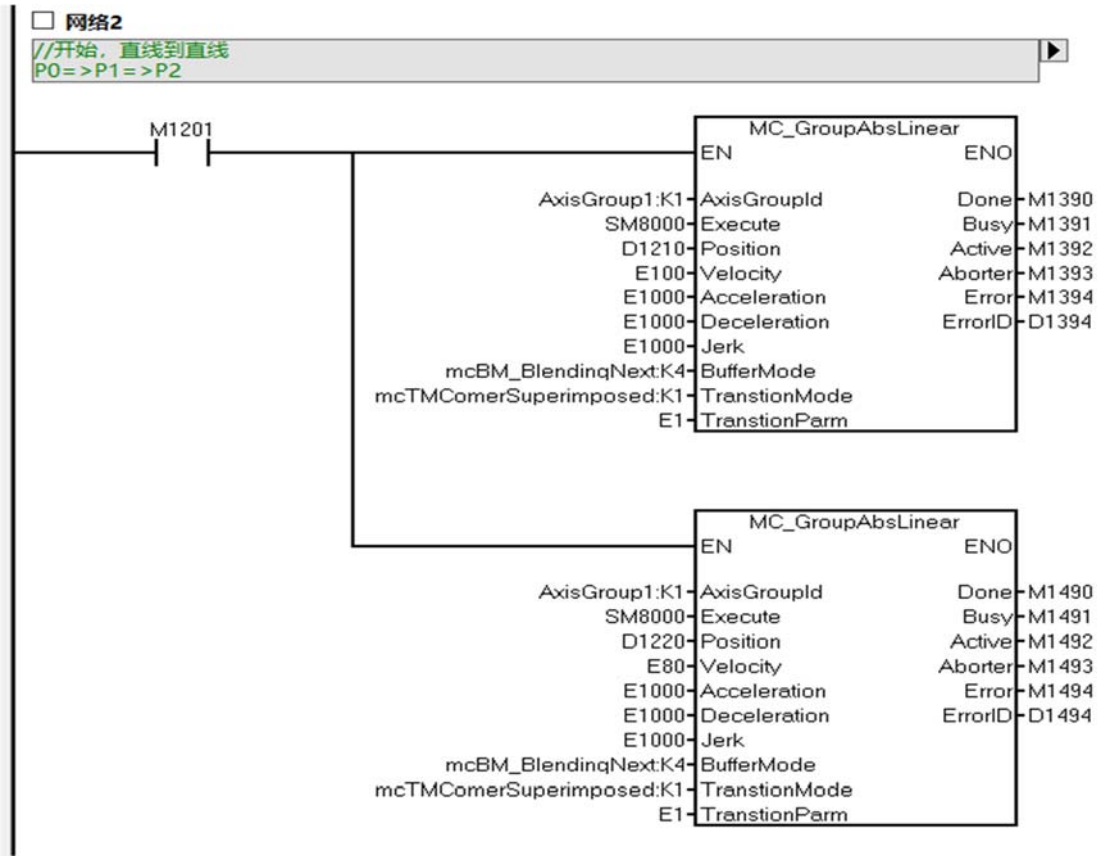


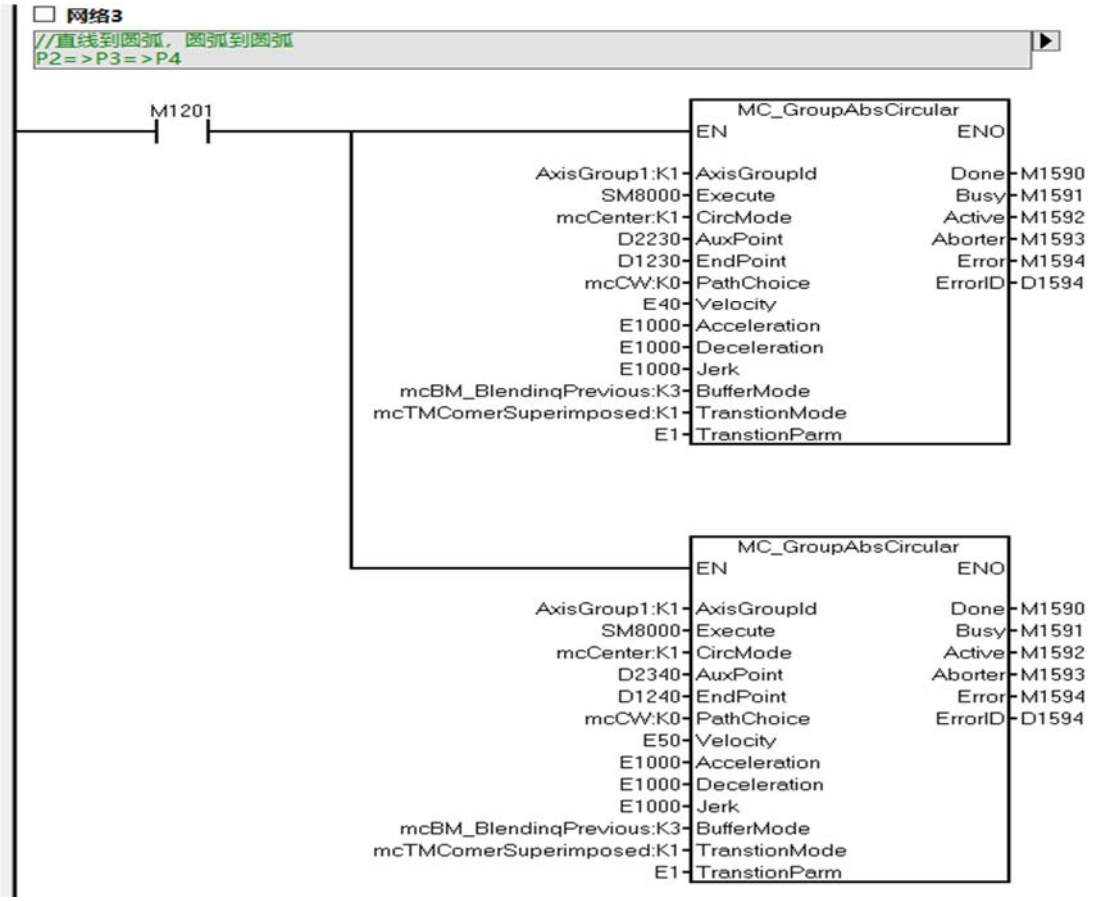
网络1

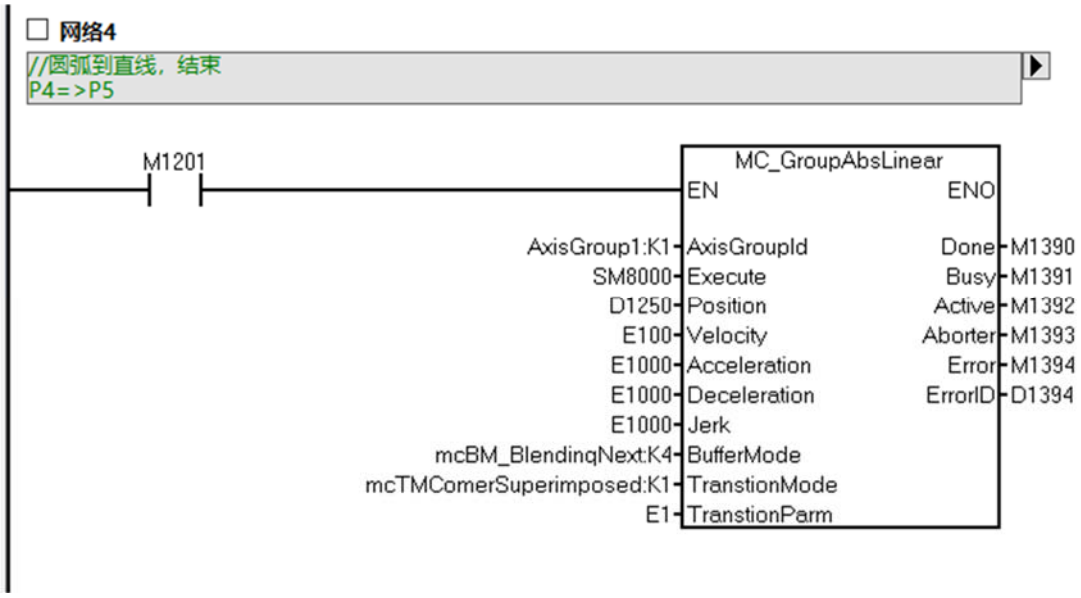
```
//点位初始化
P0 = (0,0)
P1 = (50,0) = (D1210,D1212)
P2 = (100,100) = (D1220,D1222)
P3 = (100,0) = (D1230,D1232)
P4 = (200,0) = (D1240,D1242)
P5 = (300,0) = (D1250,D1252)
O23 = (100,50) = (D2230,D2232)
O34 = (150,0) =  (D2340，D2342)
```






网络2
//开始，直线到直线
P0=>P1=>P2
M1201
MC_GroupAbsLinear
EN
ENO
AxisGroup1:K1 AxisGroupId
SM8000 Execute
D1210 Position
E100 Velocity
E1000 Acceleration
E1000 Deceleration
E1000 Jerk
mcBM_BlendingNext:K4 BufferMode
mcTMCornerSuperimposed:K1 TransitionMode
E1 TransitionParm
Done M1390
Busy M1391
Active M1392
Aborter M1393
Error M1394
ErrorID D1394
MC_GroupAbsLinear
EN
ENO
AxisGroup1:K1 AxisGroupId
SM8000 Execute
D1220 Position
E80 Velocity
E1000 Acceleration
E1000 Deceleration
E1000 Jerk
mcBM_BlendingNext:K4 BufferMode
mcTMCornerSuperimposed:K1 TransitionMode
E1 TransitionParm
Done M1490
Busy M1491
Active M1492
Aborter M1493
Error M1494
ErrorID D1494


网络3
//直线到圆弧，圆弧到圆弧
P2=>P3=>P4
M1201
MC_GroupAbsCircular
EN
ENO
AxisGroup1:K1 AxisGroupId
SM8000 Execute
mcCenter:K1 CircMode
D2230 AuxPoint
D1230 EndPoint
mcCW:K0 PathChoice
E40 Velocity
E1000 Acceleration
E1000 Deceleration
E1000 Jerk
mcBM_BlendingPrevious:K3 BufferMode
mcTMCornerSuperimposed:K1 TransitionMode
E1 TransitionParm
Done M1590
Busy M1591
Active M1592
Aborter M1593
Error M1594
ErrorID D1594
MC_GroupAbsCircular
EN
ENO
AxisGroup1:K1 AxisGroupId
SM8000 Execute
mcCenter:K1 CircMode
D2340 AuxPoint
D1240 EndPoint
mcCW:K0 PathChoice
E50 Velocity
E1000 Acceleration
E1000 Deceleration
E1000 Jerk
mcBM_BlendingPrevious:K3 BufferMode
mcTMCornerSuperimposed:K1 TransitionMode
E1 TransitionParm
Done M1590
Busy M1591
Active M1592
Aborter M1593
Error M1594
ErrorID D1594

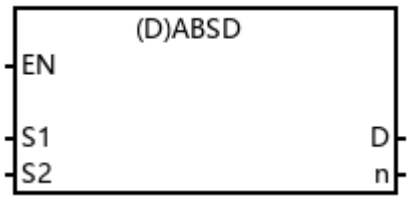


2.20 方便指令

2.20.1 ABSD/凸轮控制绝对方式指令

根据计数器的当前值，产生多个输出模式的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	数据表起始软元件	BIN16
S2	计数器	BIN16
D	输出起始位元件	位
n	输出的位元件点数/数据表数据组数（1≤n≤64）	BIN16

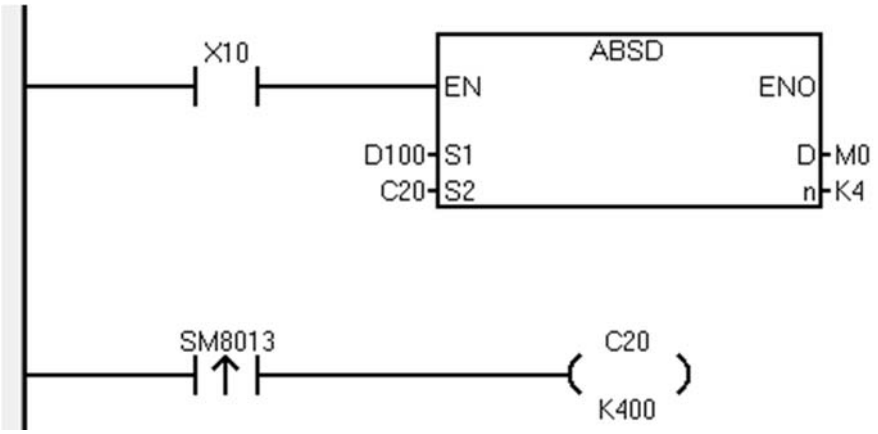
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
S2													*					*					
D		*	*			*												*					
n																			*	*			

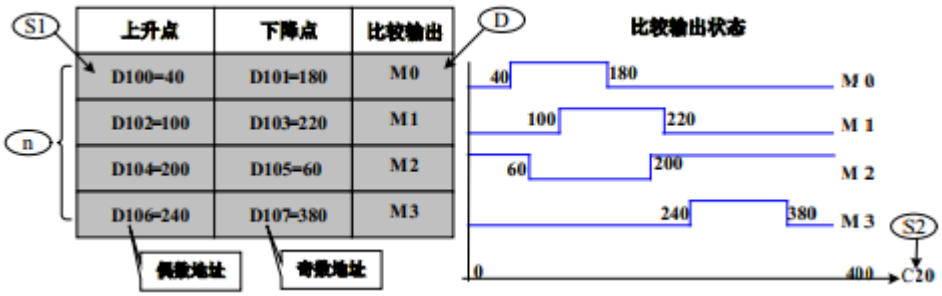
2、功能和动作说明

将 S1 为起始地址存储的数据表格数据与计数器 S2 中的当前数据进行比较，根据比较结果对 D 为起始地址的位元件进行 ON/OFF 控制。该指令完成的操作是多区段比较，用于实现凸轮控制，比较用的表格、计数器等均按绝对方式设置。

3、程序举例



当 X10 为 ON 时，执行结果如下：



4、注意事项

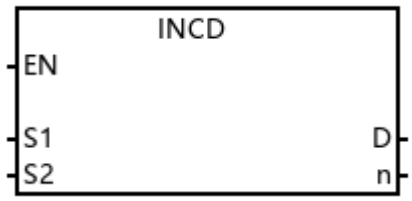
（1）该指令是主程序中扫描执行，比较结果受扫描时间的滞后影响。

（2）当 DABSD 指令中采用了高速计数器，比较输出 D 也受用户程序扫描时间的滞后影响，对于需要及时响应的应用，可以采用 HSZ 高速比较指令。

2.20.2 INCD/凸轮控制相对方式指令

使用连续的两个计数器，产生多个输出模式的指令。

1、指令格式



操作数说明

操作数	内容	类型
S1	数据表起始软元件	BIN16
S2	计数器，占用 2 个，相邻的 S2+1 对应的计数器用于计算比较匹配后计数器复位的次数	BIN16
D	输出起始位元件	位
n	输出的位元件点数/数据表数据组数（1≤n≤64）	BIN16

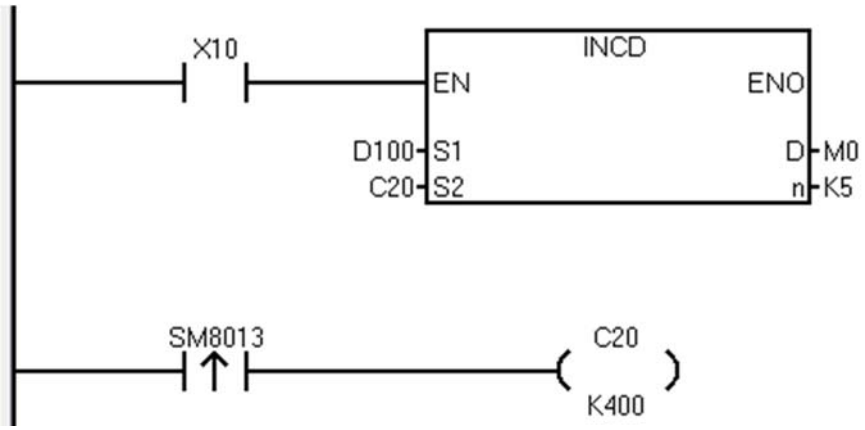
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
S2													*					*					
D		*	*			*												*					
n																			*	*			

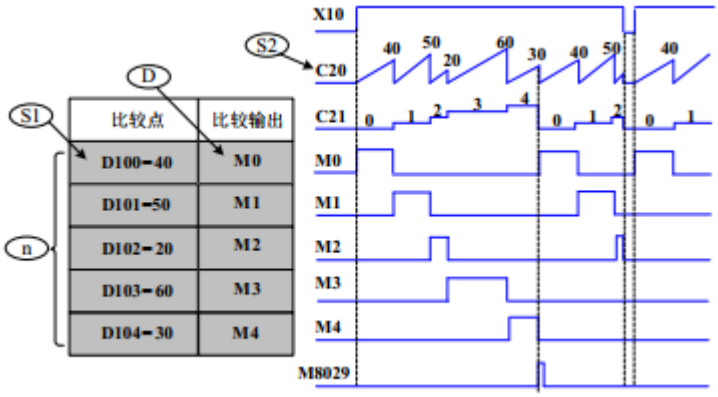
2、功能和动作说明

将 S1 为起始地址存储的数据表格数据与计数器 S2 中的当前数据进行比较，根据比较结果对 D 为起始地址的位元件进行 ON/OFF 控制。该指令完成的操作是多区段比较，用于实现凸轮控制，比较用的表格、计数器等均按增量方式设置。

3、程序举例



当 X10 为 ON 时，执行结果如下：



4、注意事项

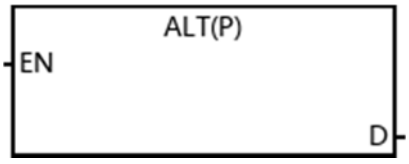
（1）该指令是主程序中扫描执行，比较结果收扫描时间的滞后影响，对于需要及时响应的应用，可采用 HSZ 高速比较指令。

（2）n 的组数比较完毕后，指令执行完毕标志 SM8029 会置位。

2.20.3 ALT/交替输出指令

输入为 ON 时，使位软元件状态反转用的指令。

1、指令格式



操作数说明

操作数	内容	类型
D	交替输出的位软元件	位

操作数可用元件类型

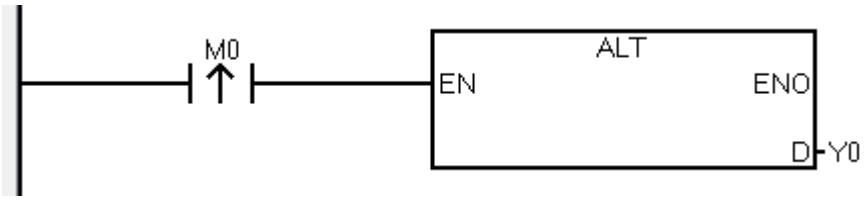
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D		*	*			*												*					

2、功能和动作说明

指令输入每次从 OFF 变为 ON 时，D 中指定的位软元件就执行状态的反转。

4、程序举例

每次 M0 的上升沿触发，Y0 的状态都会反转



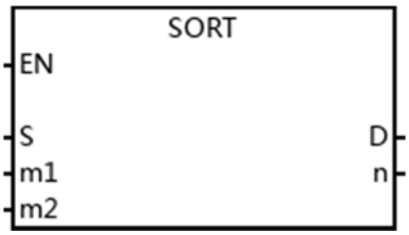
4、注意事项

使用 ALT 指令编程时，每个运算周期都执行反转动作，使用触点指令来控制输出的状态反转。

2.20.4 SORT/数据排序指令

该指令是用于将数据(行)和群数据(列)构成的数据表格，以指定的群数据(列)为标准，按照行单位将数据表格重新升序排列。在这个指令中，群数据(列)被保存在连续的软元件中。此外，数据(行方向)被保存在连续的软元件中。还有便于增加数据(行)，支持升序/降序排列的 SORT2(FNC 149)指令。

1、指令格式



操作数说明

操作数	内容	类型
S	源数据表格[占用 m1×m2 点]	BIN16 位
m1	行数[1～32]	
m2	列数[1～6]	
D	保存排序后的表[占用 m1×m2 点]	
n	作为排序标准的列编号[1～m2]	

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*								
m1														*	*				*	*			
m2														*	*				*	*			
D																							
n														*	*				*	*			

2、功能和动作说明

在 S 开始的(m1×m2)点的数据表格(排序前)中，以 n 列的群数据为标准，按照升序重新排列数据行，然后保存在从 D 开始的(m1×m2)点的数据表格(排序后)中。

3、程序举例

在“n=K2(列号 2)”和“n=K3(列号 3)”的情况下，对如下所示的排序前的数据进行排序，动作如下所示。如果先在第 1 列中输入管理编号等连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前数据：

行号 \ 列号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	S	S+5	S+10	S+15
		1	150	45	20
	2	S+1	S+6	S+11	S+16
		2	180	50	40
	3	S+2	S+7	S+12	S+17
		3	160	70	30
	4	S+3	S+8	S+13	S+18
		4	100	20	8
	5	S+4	S+9	S+14	S+19
		5	150	50	45

（1）以 n=K2(列号 2)为基准执行指令时的排序结果(升序的情况)

行号 \ 列号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+5	D+10	D+15
		4	100	20	8
	2	D+1	D+6	D+11	D+16
		1	150	45	20
	3	D+2	D+7	D+12	D+17
		5	150	50	45
	4	D+3	D+8	D+13	D+18
		3	160	70	30
	5	D+4	D+9	D+14	D+19
		2	180	50	40

（2）以 n=K3(列号 3)为基准执行指令时的排序结果(降序的情况)

列号 / 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+5	D+10	D+15
		4	100	20	8
	2	D+1	D+6	D+11	D+16
		1	150	45	20
	3	D+2	D+7	D+12	D+17
		2	180	50	40
	4	D+3	D+8	D+13	D+18
		5	150	50	45
	5	D+4	D+9	D+14	D+19
		3	160	70	30

4、注意事项

（1）相关软元件的动作

M8029：指令结束标志，排序结束时置位；

（2）以下操作会引起错误排序，请注意

- 动作过程中，请勿使操作数和数据的内容变化。
- 再次执行时，请将指令输入 OFF 一次。
- 程序中仅可使用 1 次。
- 原来的数据和排序替换后的数据存放的位置，请错开，不要重叠。

三、应用篇

3.1 子程序

动作说明：当指令输入为 ON 时，执行 CALL 指令，向标记 P 的步跳转。接着，执行标记 P 的子程序，执行 SRET 后，返回到 CALL 指令的下一步，子程序的调用返回程序可以参考 2.2.2 章节。

注意事项：

（1）在主程序的最后用 FEND 指令编程；

（2）CALL 指令用的标记 P，必须在 FEND 指令后编程；

（3）最多允许 6 层嵌套，超出报 6634 停机错误；

（4）子程序没有 SRET 指令，XPLC 报 6637 停止运行。

关于流程类的梯形图 XPLC 在自检的时候就会把错误找出来并停机错误。

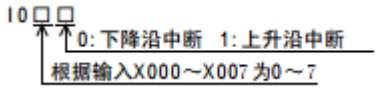
3.2 中断程序

动作说明：EI 是中断开放指令，DI 是中断禁止指令。中断源触发中断后，PLC 就跳转到其中断用指针编号指定的子程序执行。IRET 是中断返回指令，它是中断处理程序的结束指令，其作用是使 PLC 返回到被中断时的下一条指令继续执行。 若没有中断处理子程序，虽然触发了中断，但没有处理任何事情。

1、中断事件分类

（1）外部输入中断：8 点

外部中断指针标号如下：



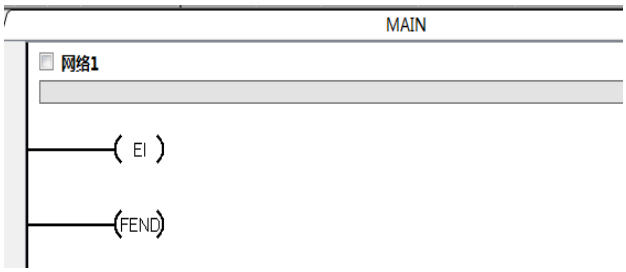
在不受可编程控制器运算周期的影响下，接收来自特定的输入编号的输入信号。触发该输入信号，执行中断子程序。由于输入中断可以处理比运算周期更短的信号，因此可在顺控过程中作为需要优先处理或者短时间脉冲处理控制时使用。

输入	输入中断用指针		禁止中断标志位
	上升沿中断	下降沿中断	
X00	I001	I000	SM8130
X01	I011	I010	SM8131
X02	I021	I020	SM8132
X03	I031	I030	SM8133
X04	I041	I040	SM8134
X05	I051	I050	SM8135
X06	I061	I060	SM8136
X07	I071	I070	SM8137

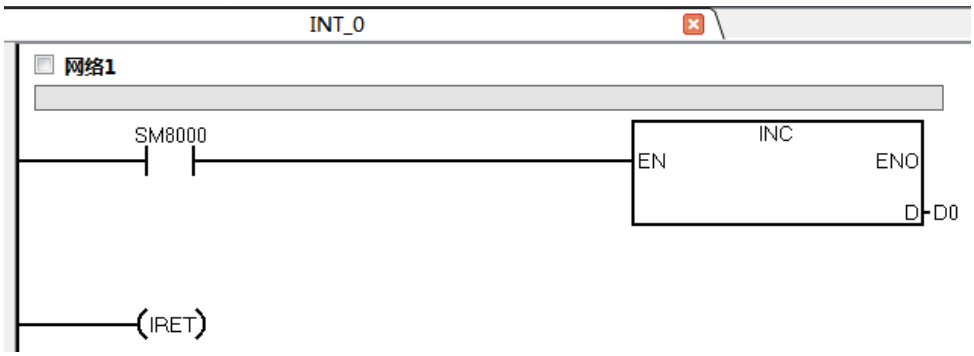
同一个梯形图中同一个输入的上升沿和下降沿都可以触发对应的外部中断。

程序举例:

主程序：

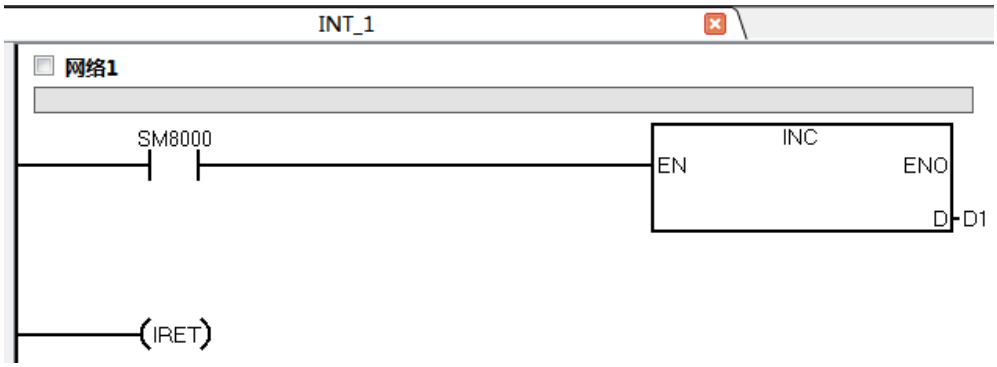


I000 中断子程序：



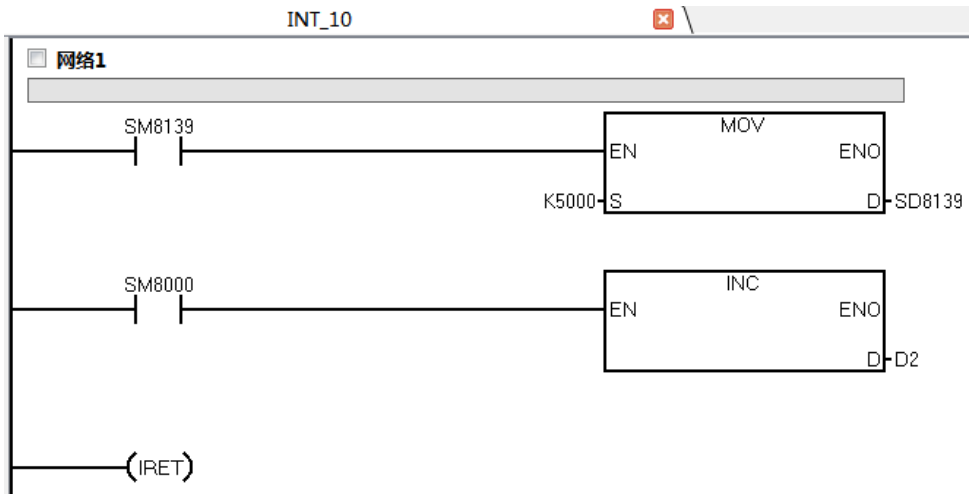
中断说明：当 X0 发生下降沿中断时，D0 计数立即加 1。

I001 中断子程序：



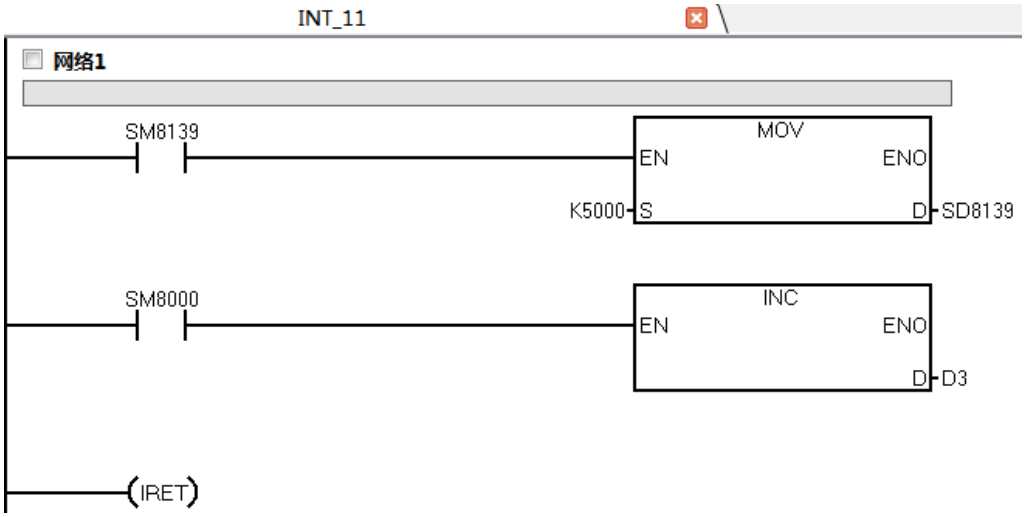
中断说明：当 X0 发生上升沿中断时，D1 计数加立即 1。

I010 中断子程序：



中断说明：当 X1 发生下降沿中断时，由于 SM8139 外部中断延时的作用，会有 SD8139 的延迟，这里是 5S，然后 D2 计数加 1。

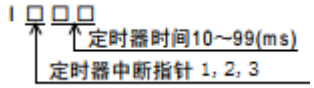
I011 中断子程序：



中断说明：当 X1 发生上升沿中断时，由于 SM8139 外部中断延时的作用，会有 SD8139 的延迟，这里是 5S，5S 后 D3 计数加 1。

（2）定时器中断：3 点

定时器中断指针标号如下：

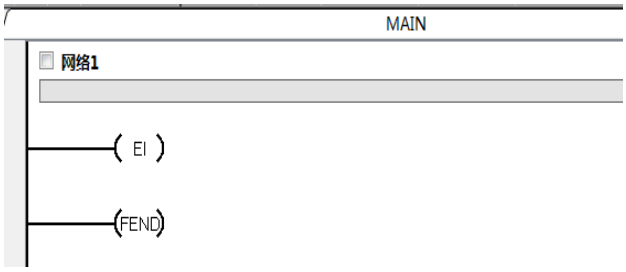


每隔指定的中断循环时间(10ms～99ms)，执行中断子程序。在可编程控制器的运算周期以外，需要循环中断处理的控制中使用。

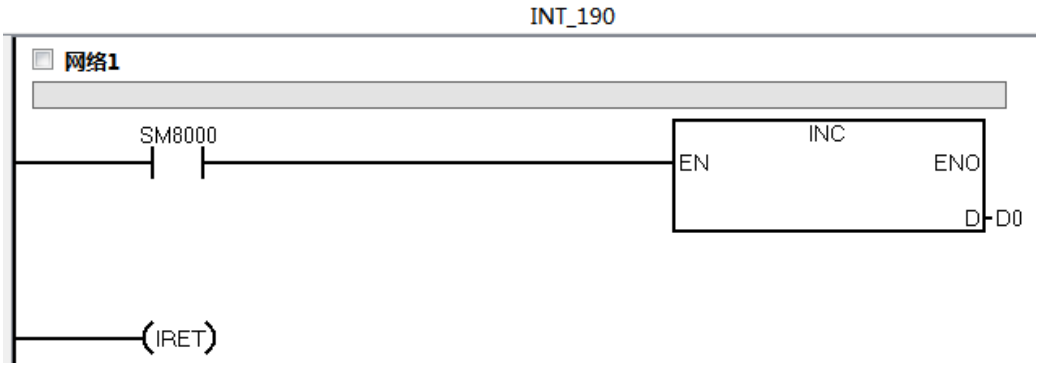
输入编号	中断周期（ms）	中断禁止标志位
I1XX	在指针名的 XX 中，输入 10~99 的整数。 例如:I110=每 10ms 的定时器中断	SM8140
I2XX		SM8141
I3XX		SM8142

程序举例：

主程序：

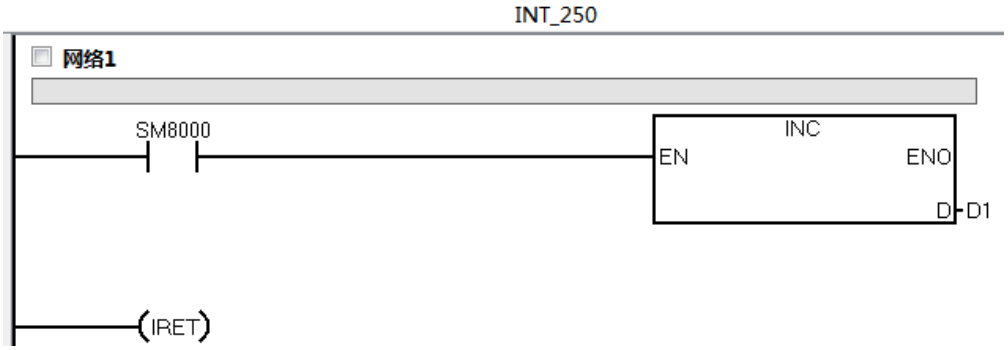


I190 中断子程序：



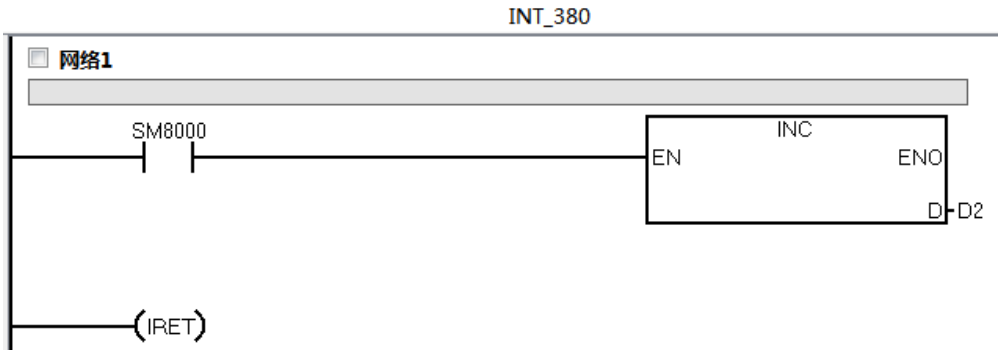
中断说明：定时器 1 计时 90ms 后，发生定时器中断，D0 计数立即加 1。

I250 中断子程序：



中断说明：定时器 2 计时 50ms 后，发生定时器中断，D1 计数立即加 1。

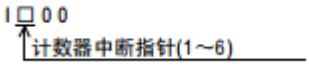
I380 中断子程序：



中断说明：定时器 3 计时 80ms 后，发生定时器中断，D2 计数立即加 1。

（3）计数器中断：6 点

计数器中断指针标号如下：

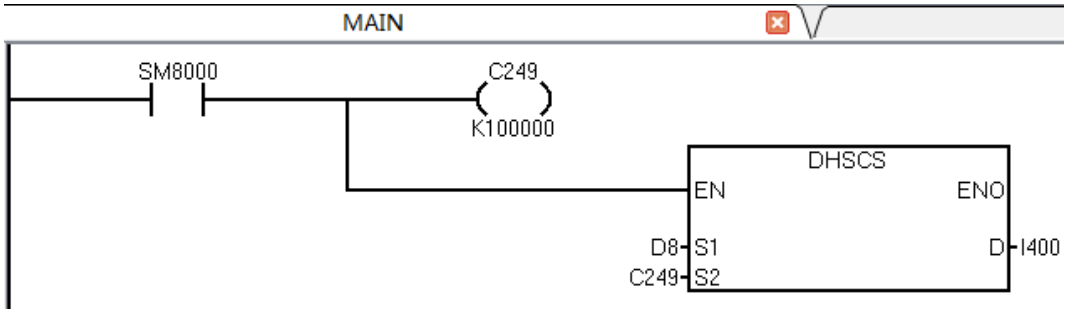


根据高速计数器用比较置位指令(DHSCS 指令)的比较结果，执行中断子程序。用于使用高速计数器优先处理计数结果的控制。

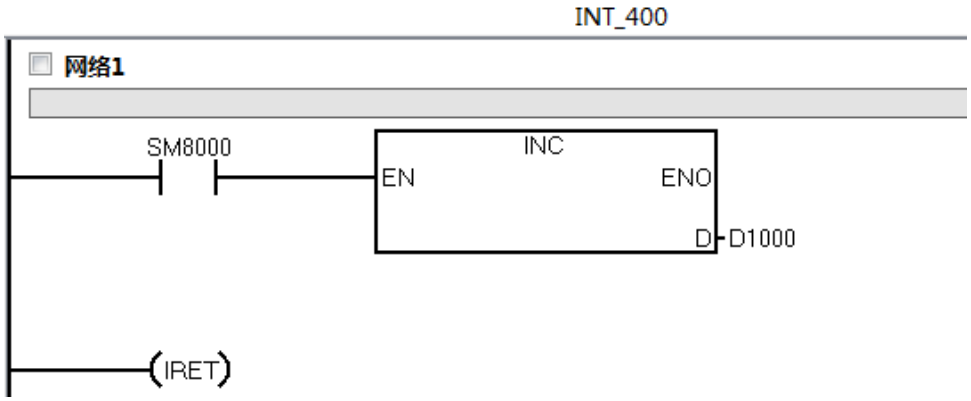
指针编号	中断禁止标志位
I100	SM8150
I200	SM8151
I300	SM8152
I400	SM8153
I500	SM8154
I600	SM8155

程序举例：

主程序：



I400 中断子程序：



中断说明：当高速计数器 C249 计数值与 D8 相当时，会立即触发 I400 这个计数器中断，然后 D1000 值加 1。

2、中断优先级

外部输入中断 > 定时器中断 > 计数器中断

I000 > I010 > I020 > I030 > I040 > I050 > I060 > I070 > I001 > I011 > I021 > I031 > I041 > I051 > I061 > I071 > I1xx > I2xx > I3xx > I100 > I200 > I300 > I400 > I500 > I600

3、注意事项

（1）如果对应的中断禁止标志置位，即使产生了中断也不会执行。

（2）注意 SM8139 寄存器的动作，中断程序后面紧接一个 LD SM8139，表示中断产生后不立即动作，而是延迟 SD8139 的延时时间才动作。

4、中断重定向*1

当输入端子 X00-X07 被其他功能占用时，可以使用 SM8120~SM8127 将中断功能重定向到输入端子 X10~X17。重定向即可将输入信号接到 X10~X17 使用对应功能。

原始端子	重定向后的端子	使能开关
X00	X10	SM8120
X01	X11	SM8121
X02	X12	SM8122
X03	X13	SM8123
X04	X14	SM8124
X05	X15	SM8125
X06	X16	SM8126
X07	X17	SM8127

受重定向影响的功能有：外部输入中断、输入捕获、MC_TouchProbe。

1. 重定向后信号响应能力可能受到影响，各个端子中断滤波滤波时间参考 2.7.2 的“3、最小滤波时间”

3.3 高速计数

根据不同点数 PLC，有不同的适用范围，目前只支持 32 点。

点数	规格
16 点	3 个 AB 相 3 个单相（最多使用 3 个计数器）
24 点	4 个 AB 相 4 个单相（最多使用 4 个计数器）
32 点	4 个 AB 相 4 个单相（最多使用 4 个计数器）
40 点	4 个 AB 相 4 个单相（最多使用 4 个计数器）
60 点	6 个 AB 相 6 个单相（最多使用 6 个计数器）

根据每个点数的端子，再从下表选择端子对应模式的计数器。

(1) 支持使用的高速计数器分类：

U：增计数输入；D：减计数输入；R 复位输入；S：启动输入；

A：A 相脉冲输入；B：B 相脉冲输入

	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254	C255	C227	C228	C229	C230	C231	C232	C233	C234
X0	U/D								U/D			U			U		A	A										
X1		U/D							R			D			D		B	B										
X2			U/D							U/D			U		R		R		A									
X3				U/D						R			D		S				B									
X4					U/D					S				U		U				A									
X5						U/D					U/D			D		D				B									
X6							U/D				R					R					A								
X7								U/D			S					S					B								

X10																					U/D				U			
X11																						U/D			D	U	A	
X12																							U/D			D	B	A
X13																								U/D				B

（2）控制特殊寄存器：

单相单计数增/减计数控制特殊 SM8 元件：

	单向单计数														
	C235	C236	C237	C238	C239	C240	C241	C242	C243	C244	C245	C227	C228	C229	C230
增减计数控制	SM8235	SM8236	SM8237	SM8238	SM8239	SM8240	SM8241	SM8242	SM8243	SM8244	SM8245	SM8227	SM8228	SM8229	SM8230

单相双计数增/AB 相计数增减状态特殊 SM8 元件：

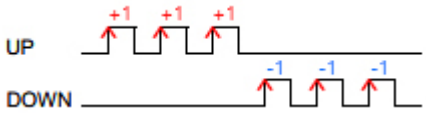
	单向单计数							AB 相计数						
	C246	C247	C248	C249	C250	C231	C232	C251	C252	C253	C254	C255	C233	C234
增减计数状态	SM8246	SM8247	SM8248	SM8249	SM8250	SM8231	SM8232	SM8251	SM8252	SM8253	SM8254	SM8255	SM8233	SM8234

倍频控制特殊 SM8 元件：

	AB 相计数						
	C251	C252	C253	C254	C255	C233	C234
四倍频使能	SM8191	SM8192	SM8193	SM8194	SM8195	SM8196	SM8197

（3）计数模式说明

计数模式	输入信号形式	计数方向
单相单计数		通过 SM8235～SM8245 的 ON/OFF 来指定增计数或是减计数。 ON：减计数 OFF：增计数

单相双计数			通过 SM8246～SM8250 的 ON/OFF 来查看增计数或是减计数状态。 ON：减计数状态 OFF：增计数状态
双相双计数	1 倍频		通过 SM8251～SM8255 的 ON/OFF 来查看增计数或是减计数状态。 ON：减计数状态 OFF：增计数状态 通过置位 SM8195～SM8199 可使 C251～C255 转换为 4 倍频计数模式
	4 倍频		

（4）高速指令的使用，请参考 2.7 章节。

注意：断电后，高速计数器的复位状态会被保存，梯形图中无 RST 指令复位此高速计数器的情况下，使用该高速计数器不能正常计数。另外，使用 RST 指令复位高速计数器时，RST 指令前面不能使用常闭开关，如 SM8000，否则，会导致计数器处于复位状态而不能正常计数。

3.4 串口通讯部分

X 系列 PLC 包含三路可用串口，232 口、485 口，还有一路 BD 板口，当 BD 板时 485BD 板时，其功能与另外两个串口功能相同，都是以梯形图形式来实现 232 口和 485 口的 RS2 或 MODBUS 通信。COM0、COM1、COM2 接 485BD 板时,都可以进行监控下载、RS2 通信、MODBUS 通信。COM0、COM1、COM2 是相对于 PLC 板子的三个串口通道的顺序，上位机显示的则是实际电脑检测出来的 COM 标号，不能混淆。

3.4.1 监控下载

其中 RS485 接口和 RS232 接口支持下载监控协议。

通过 USB-485 通信转换器或 USB-232 通信转换器，将 PLC 对应端口与 PC 相连接，在 PC 上打开 UniSYS 软件，即可通过 UniSYS 软件对 PLC 进行下载梯形图和监控操作。值得说明的是，在通信开始之前，需要在 UniSYS 软件上对 PLC 进行通信测试，依次点击菜单栏工具->通信配置，选择对应端口，设置波特率，点击测试，若弹框显示通信测试成功，则说明 UniSYS 软件可与 PLC 进行正常的通信操作，若显示其他弹框，则说明通信存在故障，UniSYS 软件无法与 PLC 进行正常通信，此时需查找故障原因（查看接线是否正确，PLC 是否正常）。下图为正常通信状态显示弹框。



下载监控协议是一种主从协议，其中 UniSYS 软件作为主机，PLC 作为从机。从机一般不主动发送通信数据，在主机有通信请求时，从机会向主机发送通信数据作为应答。

下载监控协议是一类协议命令的统称，此协议包含 UniSYS 软件对 PLC 的大部分操作，包括下载，上载，校验，监控，强制等功能。

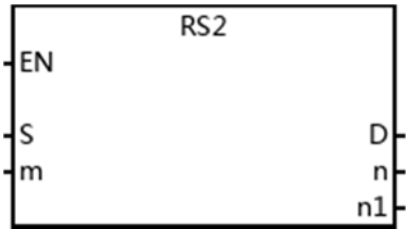
下载通信协议可以设置多种波特率，最常见的有 9600baud，19.2kbaud 和 115.2kbaud。

USB-485 或 USB-232 通信转换器是一种即插即用设备，可支持 USB V1.1 及 USB2.0 的 PC，此设备提供 PC 端 USB 端口的数据向 RS485 或 RS232 类型数据的转换，以便 PLC 端对应的端口可以收到正常的通信数据。必要时，使用此设备需要在 PLC 端安装对应的驱动程序，可使用驱动安装软件进行自动检测，并一键安装相应的驱动。

3.4.2 RS2 通信

通过 XPLC 本体上的 COM0/COM1/COM2(接 485BD 板)口进行自由口通信，从而执行数据的发送和接收的指令。

1、RS2 指令格式



操作数说明

操作数	内容	类型
S	保存发送数据的数据寄存器的起始软元件	BIN16 位
m	发送数据的字节数[设定范围：1～512]	BIN16 位
D	数据接收结束时，保存接收数据的数据寄存器的起始软元件	BIN16 位
n	接收数据的字节数[设定范围：1～512]	BIN16 位
n1	使用通道编号[设定内容：K0：COM0； K1：COM1；K2：COM2]	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
m														*	*				*	*			
D														*	*			*					
n														*	*				*	*			
n1																			*	*			

2、功能和动作说明

该指令用于 RS-232C 或 RS-485 串行通信口的无协议通信，从而执行数据的发送和接收。

使用该指令首先得使用梯形图配置通信相关的寄存器。

（1）通信格式寄存器配置表

通信格式配置	Bit15~13	Bit12~11	Bit10~7	Bit6~4	Bit3	Bit2~1	Bit0
协议	协议类型	硬件类型	RS2 的帧格式	波特率	停止位	校验位	数据位
RS2	010：RS2	00:232 口 01:485 口	0001~1100	001:4800 010:9600 011:19200 100:38400	0:1bit 停止位 1:2bit 停止位	00:无校验 01:奇校验 10:偶校验	0:7bit 数据位 1:8bit 数据位

通过配置 RS2 帧格式，选择要在通信中使用的报文帧：

0001	数据
0010	数据+CR+LF
0011	数据+报尾
0100	数据+报尾+CR+LF
0101	数据+报尾+和校验
0110	数据+报尾+和校验+CR+LF
0111	报头+数据
1000	报头+数据+CR+LF
1001	报头+数据+报尾
1010	报头+数据+报尾+CR+LF
1011	报头+数据+报尾+和校验
1100	报头+数据+报尾+和校验+CR+LF

（2）COM0（232 口）使用 RS2 通信时的相关软元件

位软元件	名称
SM8301	串行通信错误
SM8302	超时的判断标志位
SM8310	发送请求
SM8311	接收结束标志位
SM8312	发送等待标志位

字软元件	名称
SD8300	设定通信格式
SD8301	串行通信错误代码
SD8302	设定响应超时时间（单位 10ms），设定为 0 时，默认 100ms
SD8310	发送数据的剩余点数
SD8311	接收点数的监控
SD8312	报头 1、2
SD8313	报头 3、4
SD8314	报尾 1、2
SD8315	报尾 3、4
SD8316	接收和校验(接收数据)
SD8317	接收和校验(计算结果)
SD8318	发送和校验

（3）COM1（485 口）使用 RS2 通信时的相关软元件

位软元件	名称
SM8321	串行通信错误
SM8322	超时的判断标志位
SM8330	发送请求
SM8331	接收结束标志位
SM8332	发送等待标志位

字软元件	名称
SD8320	设定通信格式
SD8321	串行通信错误代码
SD8322	设定响应超时时间（单位 10ms），设定为 0 时，默认 100ms
SD8330	发送数据的剩余点数
SD8331	接收点数的监控

SD8332	报头 1、2
SD8333	报头 3、4
SD8334	报尾 1、2
SD8335	报尾 3、4
SD8336	接收和校验(接收数据)
SD8337	接收和校验(计算结果)
SD8338	发送和校验

（4）COM2（接 485BD 板）使用 RS2 通信时的相关软元件

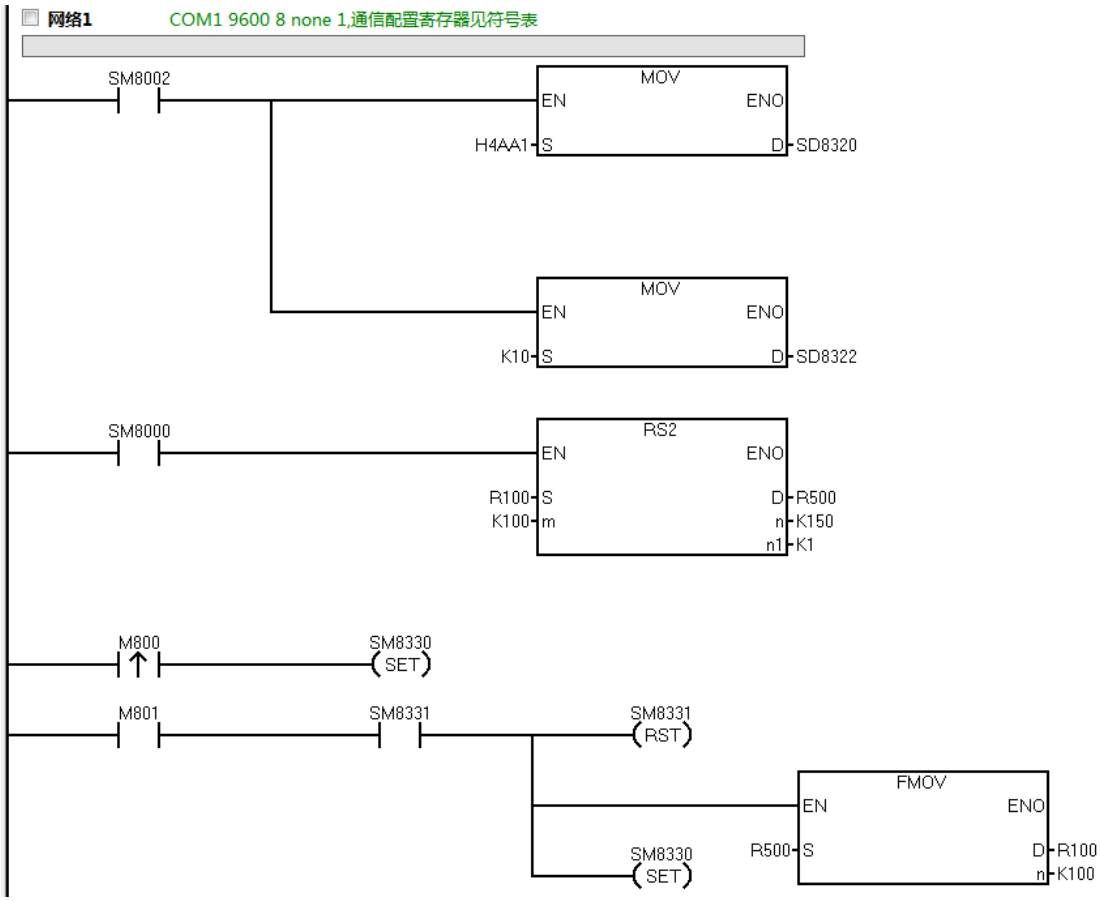
位软元件	名称
SM8341	串行通信错误
SM8342	超时的判断标志位
SM8350	发送请求
SM8351	接收结束标志位
SM8352	发送等待标志位

字软元件	名称
SD8340	设定通信格式
SD8341	串行通信错误代码
SD8342	设定响应超时时间（单位 10ms），设定为 0 时，默认 100ms
SD8350	发送数据的剩余点数
SD8351	接收点数的监控
SD8352	报头 1、2
SD8353	报头 3、4
SD8354	报尾 1、2
SD8355	报尾 3、4
SD8356	接收和校验(接收数据)
SD8357	接收和校验(计算结果)

SD8358	发送和校验

其中 SD8300/SD8320/SD8340 用于设定 COM0/COM1/COM2 的通信格式，具体设定方法见：注意 RS2 通信的波特率最大可设置成 38400。

3、程序举例



当与其通信的设备串口配置与此一致时，梯形图中置位 SM8300（发送请求），plc 将发送 R100~R149 中的 100 个字节的数据。当 SM8331（接收结束标志）置位时，监控可以发现 R500 以后的 150 个字节保存的是通信设备发送过来的数据，此时再次置位 M801，plc 会将接收的数据中的前 100 个字节发送给通信设备。

4、注意事项

（1）报头：最多 4 个报头，放于对应的特殊寄存器，00H 之前的数据有效，按低位、高位、低位、高位的顺序排布。比如 COM1 的报头设定为 SD8332 = 1234H，SD8333 = 5600H，则实际报头为 H34（SD8332 低位） H12（SD8332 高位）。

（2）报尾：最多四个报尾，使用方法类似报头。

（3）和校验：选择和校验时必须设定有报尾。发送数据时，计算数据+报尾的和，附加在发送的数据中；接收数据时，检查接收到的和，是否与 PLC 中计算的相同。

（4）CR+LF：通信格式设定中选择了 CR+LF，会在发送的数据后加上 CR+LF 的字符代码，接收数据时如果连续接收到 CR+LF，则表示接收结束。

（5）接收数据的字节点数 n 应该大于等于实际接收的数据长度加上 8。

（6）一般情况下，接收完毕最好不要立即置位发送请求标志，给一点延时时间等待串口状态转换。

3.4.3 串口 MODBUS RTU 通信

1、梯形图中配置通信需要的参数说明

（1）通信格式配置表

通信格式配置	Bit15~13	Bit12~11	Bit10~7	Bit6~4	Bit3	Bit2~1	Bit0
协议	协议类型	硬件类型	主从站	波特率	停止位	校验位	数据位
MODBUS	011：MODBUS	00:232 口 01:485 口	0000：从站 0001：主站	001:4800 010:9600 011:19200 100:38400 101:57600 110:115200	0:1bit 停止位 1:2bit 停止位	00:无校验 01:奇校验 10:偶校验	0:7bit 数据位 1:8bit 数据位

RTU 模式数据传输方式：

编码系统：8 位二进制，十六进制 0-9，A-F

每个字节的位：1 个起始位；8 个数据位；带校验时 1 位停止位；无校验时 2 位停止位

错误校验区：循环冗余校验（CRC）

（2）通信协议配置表

位	名称	内容	
		0（bit=OFF）	1（bit=ON）
b0	选择协议	其他通信协议	MODBUS 协议

b1~b3	不可使用		
b4	主从站设定	MODBUS 主站	MODBUS 从站
b5~b7	不可使用		
b8	RTU/ASCII 模式	RTU	ASCII（不支持）
b9~b15	不可使用		

（3）SD8300/SD8320/SD8340 允许配置的通信格式

主站			从站			通信
数据位	校验位	停止位	数据位	校验位	停止位	
8bit	无	1bit	8bit	无	1bit	是
8bit	无	2bit	8bit	无	2bit	是
8bit	奇	1bit	8bit	奇	1bit	是
8bit	奇	2bit	8bit	奇	2bit	是
8bit	偶	1bit	8bit	偶	1bit	是
8bit	偶	2bit	8bit	偶	2bit	是

2、相关特殊寄存器及错误码说明

（1）MODBUS 相关特殊寄存器

其中：COM0 是 232 口，COM1 是 485 口，COM2 是 485BD 板口，下一次通信是指下一条 ADPRW 指令执行或轮询下一行 MODBUS 配置表格，正常情况下 PLC STOP 时所有相关寄存器都会清除。

特殊数据寄存器			有效站	详细内容
COM0	COM1	COM2		
SM8029	SM8029	SM8029	主站	指令结束标志（MODBUS 中用于 ADPRW 指令）
SM8301	SM8321	SM8341	主站/从站	串口通信出错锁存
SM8302	SM8322	SM8342	主站	超时的判断标志，当到达重发次数时，超时标志才会置位，下一次通信时清除
SM8303	SM8323	SM8343	主站/从站	MODBUS 通信中
SM8304	SM8324	SM8344	主站/从站	MODBUS 通信发生错误，下一次通信时清除
SM8305	SM8325	SM8345	主站	主站发生重发事件的判断标志，下一次通信

				时清除
SM8306	SM8326	SM8346	主站/从站	MODBUS 通信出错锁存
SD8300	SD8320	SD8340	主站/从站	通信格式设定
SD8301	SD8321	SD8341	主站/从站	错误代码编号，MODBUS 通信出错时，里面写入 6321
SD8302	SD8322	SD8342	主站	设定响应超时时间（单位 ms），填 0 时默认 3s
SD8303	SD8323	SD8343	主站/从站	协议
SD8304	SD8324	SD8344	主站/从站	显示主站当前的重发次数
SD8305	SD8325	SD8345	主站	设定进行重发的次数（0~20），超过 20 为 20
SD8306	SD8326	SD8346	主站	请求间延迟（帧间延迟，单位 ms），波特率大于 19200，延迟时间默认 4ms，其他为 10ms，从站中不可更改，当主站连接的从站数比较多时，建议设定>=10ms
SD8307	SD8327	SD8347	从站	从站本站号（范围 1~247）
SD8308	SD8328	SD8348	主站/从站	MODBUS 通信错误寄存器，显示当前最新错误
SD8309	SD8329	SD8349	主站/从站	MODBUS 通信错误锁存寄存器，锁存第一次的错误

（2）MODBUS 错误码说明

错误码	出错内容	有效站	相关元件动作	解决方式
202	MODBUS 参数设定错误	主站/从站	（1）使用 COM0 时：SM8301 、 SM8304 、SM8306 会置 ON，当前的错误码存在 SD8308 中，SD8309 会锁存第一次的错误码，SD8301 存入 6321 （2）使用 COM1 时：SM8321 、 SM8324 、SM8326 会置 ON，当前的错误码存在 SD8328 中，SD8329 会锁存第一次的错误码，SD8321 存入 6321	检查 MODBUS 相关通信配置梯形图或配置参数表是否正确
205	CRC/LCR 校验码错误	主站/从站		检查数据协议帧的校验是否正确
207	响应文本长度不正确	主站/从站		确定接收的响应文本的长度是否正确
210	非法软元件地址读写	主站/从站		确认主站是否访问了有效的软元件范围（地址是否是允许范围内）； 确认从站的 MODBUS 地址分配是否正确或者包含了读写的

			（3）使用 COM2 时：SM8341 、 SM8344 、SM8346 会置 ON，当前的错误码存在 SD8348 中，SD8349 会锁存第一次的错误码，SD8341 存入 6321	地址范围
212	异常文本	主站/从站		确定主站读取的地址+点数是否超出规定范围； 检测 ADPRW 指令参数设定是否正确
214	无此功能码	主站/从站		确定 MODBUS 是否支持该功能码
217	ADPRW 指令不正确使用	主站/从站		确认主站中的 MODBUS 通信否配置成功； 确认从站中是否使用了 ADPRW 指令，如果有则删除
211	从站响应超时	主站	除上述动作外，对应的串口超时标志位（分别是 SM8302、SM8322、SM8342）会置位	确定从站站号和通信参数是否正确； 确定通信线是否连接稳定； 确定从站是否正常启动

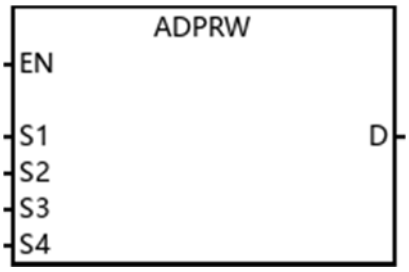
3、MODBUS 标准功能支持一览表

功能代码	子功能代码	功能名	文本可访问点数	该功能是否允许
0x01		线圈读出	1~2000 点	√
0x02		输入读出	1~2000 点	√
0x03		保持寄存器读出	1~125 点	√
0x04		输入寄存器读出	1~125 点	√
0x05		1 线圈写入	0 固定	√
0x06		1 寄存器写入	0 固定	√
0x0F		批量线圈写入	1~1968 点	√
0x10		批量寄存器写入	1~123 点	√
0x17		批量寄存器读出/写入	读出：1~125 点 写入：1~121 点	√

4、主站与从站通信专用指令 ADPRW

用于和 MODBUS 主站所对应从站进行通信(数据的读出/写入)的指令，指令对应的参数可参照上表进行配置。

指令格式见下表



操作数说明

操作数	内容	类型
S1	S1 指定站号和通信串口号，从站本站号（1~247），S1+1 为串口号（范围 0~2），在程序运行的第一个周期务必先将串口号填写完毕	BIN16 位
S2	功能代码	BIN16 位
S3	MODBUS 地址（0H~FFFFH）	BIN16 位
S4	读写 MODBUS 地址点数或子功能数据	BIN16 位
D	读/写数据缓存区	位/BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*					
S2														*	*			*	*	*			
S3														*	*			*	*	*			
S4														*	*			*	*	*			
D	*	*	*			*								*	*			*	*	*			

5、MODBUS 软元件地址分配

（1）位软元件

MODBUS 软元件		XPLC 软元件
输入（读出专用）	线圈（读出/写入）	

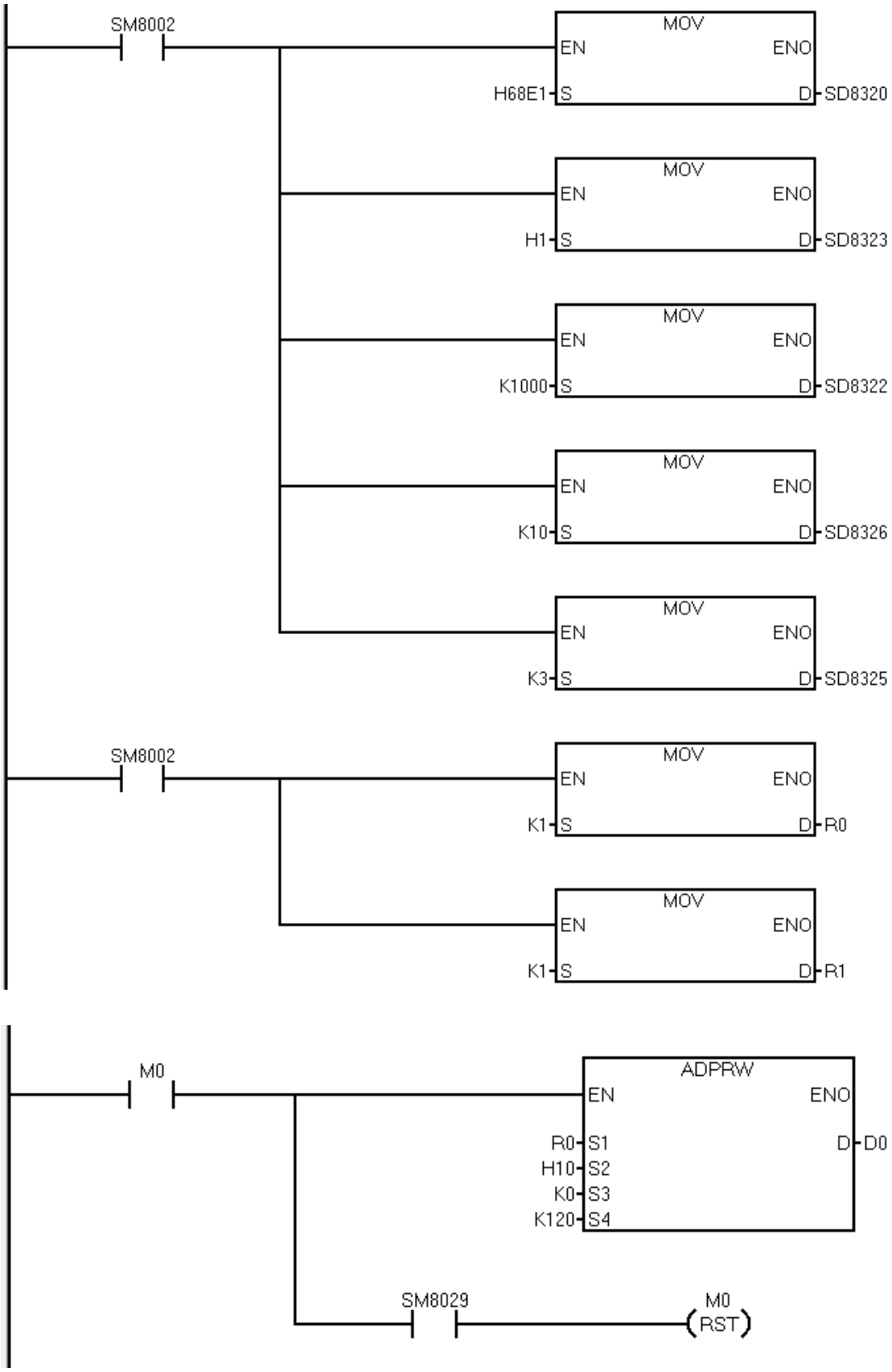
0x0000～0x1DFF	0x0000～0x1DFF	M0～M7679
0x1E00～0x1FFF	0x1E00～0x1FFF	SM8000～SM8511
0x2000～0x2FFF	0x2000～0x2FFF	S0～S4095
0x3000～0x31FF	0x3000～0x31FF	TS0～TS511
0x3200～0x32FF	0x3200～0x32FF	CS0～CS255
0x3300～0x33FF	0x3300～0x33FF	Y0～Y377
0x3400～0x34FF	-	X0～X377

（2）字软元件

MODBUS 软元件		XPLC 软元件
输入寄存器（读出专用）	保持寄存器（读出/写入）	
0x0000～0x1F3F	0x0000～0x1F3F	D0～D7999
0x1F40～0x213F	0x1F40～0x213F	SD8000～SD8511
0x2140～0xA13F	0x2140～0xA13F	R0～R32767
0xA140～0xA33F	0xA140～0xA33F	TN0～TN511
0xA340～0xA407	0xA340～0xA407	CN0～CN199
0xA408～0xA477	0xA408～0xA477	CN200～CN255
0xA478～0xA657	0xA478～0xA657	M0～M7679
0xA658～0xA677	0xA658～0xA677	SM8000～SM8511
0xA678～0xA777	0xA678～0xA777	S0～S4095
0xA778～0xA797	0xA778～0xA797	TS0～TS511
0xA798～0xA7A7	0xA798～0xA7A7	CS0～CS255
0xA7A8～0xA7B7	0xA7A8～0xA7B7	Y0～Y377
0xA7B8～0xA7C7	-	X0～X377

6、梯形图配置 MODBUS 通信举例

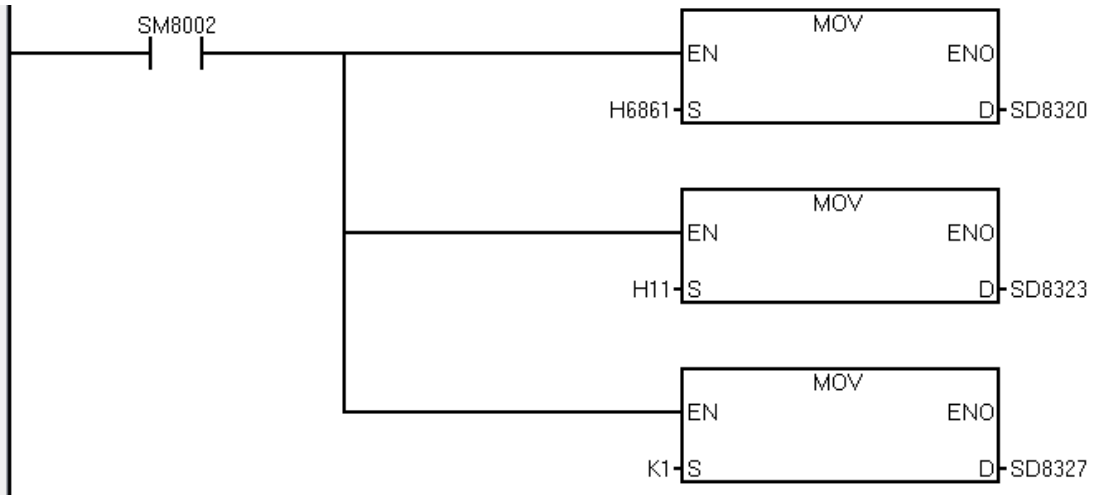
（1）COM1 主站配置举例


SM8002
MOV
EN
ENO
H68E1
S
D
SD8320
MOV
EN
ENO
H1
S
D
SD8323
MOV
EN
ENO
K1000
S
D
SD8322
MOV
EN
ENO
K10
S
D
SD8326
MOV
EN
ENO
K3
S
D
SD8325
SM8002
MOV
EN
ENO
K1
S
D
R0
MOV
EN
ENO
K1
S
D
R1
M0
ADPRW
EN
ENO
R0
S1
D
D0
H10
S2
K0
S3
K120
S4
SM8029
M0
RST

其中：COM1 的主站通信配置为波特率 115200，8 位数据位，无检验，1 位停止位，超时时间 1s，帧间延迟 10ms，重试次数 3 次。当 M0 置位时，执行批量寄存器写入功能，将 D0~D199 的 120 寄存器数据写到从站。

注意：①ADPRW 参数的实际意义，S1，S1+1 参数要在第一个周期进行初始化；②一般的，主站根据串口号对应的寄存器配置 5 个寄存器即可完成通信配置；③SM8029 的作用周期，一条 ADPRW 指令结束置位，下一条于 ADPRW 指令开始时复位。可通过此标志进行 MODBUS 的轮询操作。

（2）COM1 从站配置举例



其中：COM1 的从站通信配置为波特率 115200，8 位数据位，无检验，1 位停止位，站号为 1。

注意：一般的，从站根据串口号对应的寄存器配置 3 个寄存器即可完成通信配置。

7、配置表法配置 MODBUS 通信

（1）协议配置

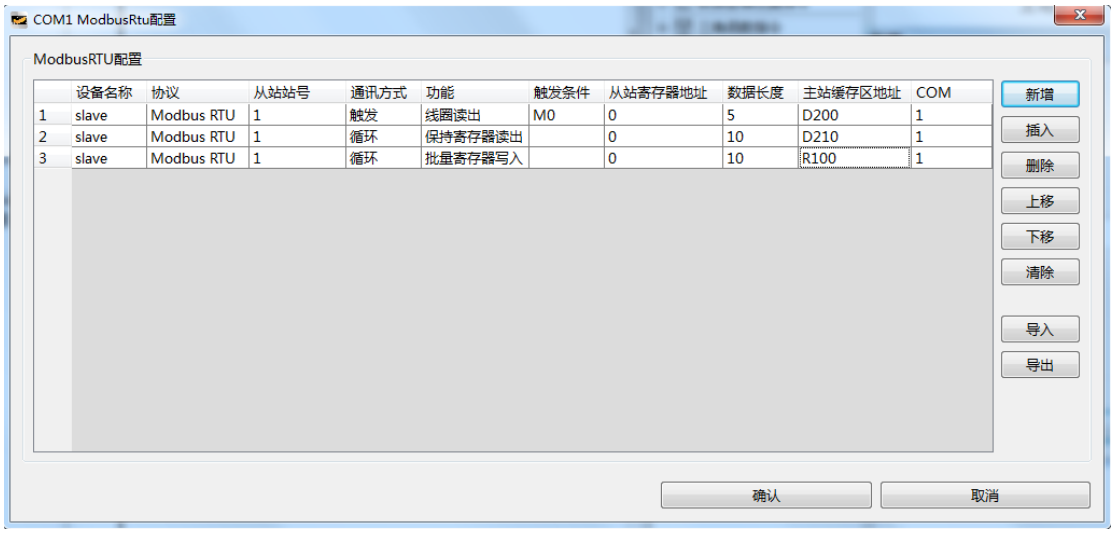
主站配置：


COM1通信协议配置
协议选择
MODBUS-RTU主站
H/W 类型
RS485
协议配置
波特率:
9600
数据长度:
8位
通信超时:
0
ms(0~32767)
校验位:
无
帧间延迟:
0
ms(0~3000)
停止位:
2位
重试次数:
1
次(1~20)
确认
取消

从站配置：


COM1通信协议配置
协议选择
MODBUS-RTU从站
H/W 类型
RS485
协议配置
波特率:
9600
站号：
1
(1~247)
数据长度:
8位
校验位:
无
停止位:
2位
确认
取消

（2）通信配置



（3）使用说明

①配置 MODBUS 通信时，可以直接采用配置表方式，将配置表和程序一起下载到 PLC，即可执行 MODBUS 通信，不需要再去编写梯形图来配置通信协议，配置表法配置通信的优先级高于梯形图配置。

②MODBUS RTU 配置表，主要用于主站通信，分为触发通讯方式和循环通讯方式。

③循环方式和触发方式的使用建议如下

• 循环通信:需要反复、尽快读写从站内变化比较快的数据，例如读取变频器的运行频率、运行状态、输入端口状态;在过程控制应用系统中，实时通信改写变频器的运行频率、输出端口的状态，等等，可选择“循环”通信方式。PLC 在执行用户程序时，会反复扫描执行通信配置表中的所有“循环” 配置项。

• 触发通信:需要定时读或写从站的刷新速度较慢的数据，例如读取变频器的输出电流、输出功率、当前故障告警信息，等等，可选择“触发”通信方式，在用户程序中,每置位触发条件 1 次，即会引起通信配置表中对应的通信项的 1 次通信操作;用户程序中定时置位该触发条件，即可实现所需频度的通信读写操作。

• 通信方式设定的建议:根据所需要交互参数刷新的特性要求进行合理选择，可使得通信表现大为改善，不要为了编程简单，将所有通信项都设置为“循环”项，这样可能因循环项太多，而使得数据交互及时性降低，影响系统的控制效果。将一些不重要的数据访问，安排为“触发”项，根据轻重缓急来触发通信，可大为改善通信及时性效果。

④配置表法中可用功能只有 4 个，线圈读出、批量线圈写入、保持寄存器读出、批量寄存器写入。

8、特别说明

（1）XPLC 的通信格式配置务必参考此文档进行配置。

（2）MODBUS 通信只有在 PLC 为 RUN 的状态下才通信，STOP 状态 PLC 恢复为监控协议。

（3）XPLC 目前支持 RTU 模式传输数据，不支持 ASCII 模式。

（4）站号可设定范围为 1-247，为保证通信稳定最大允许连接 32 台从站，当主站连接的从站数比较多时，建议设定>=10ms。

（5）支持同时使用多个 COM 口进行 MODBUS 通信，MODBUS 通信是开始于第二个周期的，故 PLC 运行的第一个周期请在 ADPRW 指令之前初始化通信配置参数以及 ADPRW 指令使用到的串口参数。

（6）不允许中途更改通信参数配置和 ADPRW 指令中的串口参数，如果更改了，只有下一次 PLC 由停止变为运行时才会生效。

（7）主站不支持播放功能。

（8）MODBUS RTU 配置表的使用优先级高于梯形图配置的寄存器内容和 ADPRW 通信指令。

3.5 密码功能

分为初级加密和高级加密。初级加密主要是通过密码对程序的读写进行限制操作，防止使用者随意上载或下载或查看程序。高级加密主要是通过秘钥来限制该程序的使用权限，只有给 PLC 注入了秘钥，才能使用与之匹配的扩展程序，扩展程序被加密后始终具有不可查看和不可更改属性，可以很好的保护自己的知识产权。

3.5.1 PLC 初级加密功能

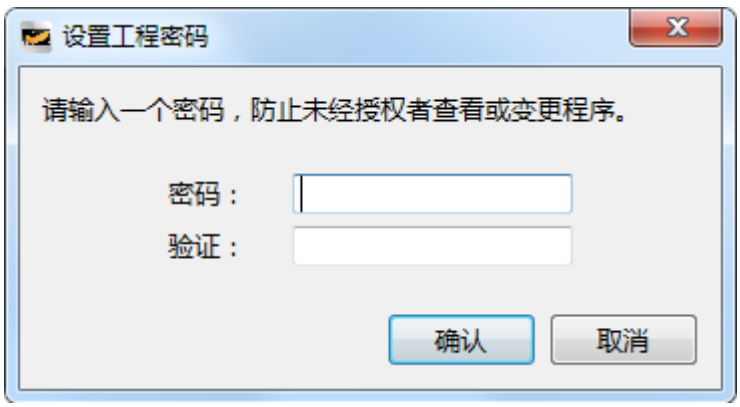
上位机通过 PLC 密码设置可以设置 PLC 程序的读写（修改）权限，初级加密分为工程加密、程序加密两大块，程序加密又分为禁止上载、禁止下载、屏蔽主程序、屏蔽子程序。

1、工程加密

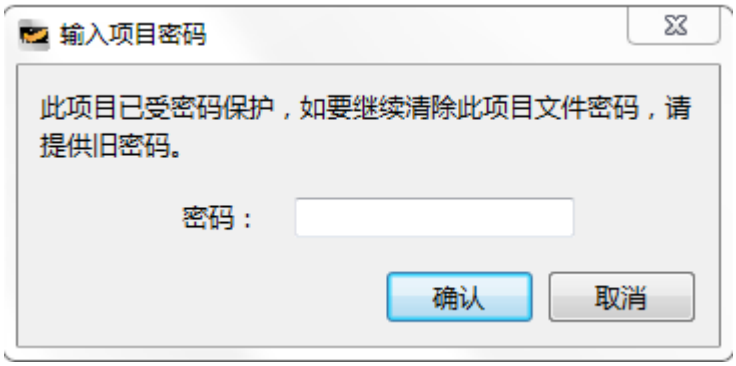
防止未授权者查看和变更程序，只允许下载，下载后，PLC 中存储此密码，此程序将不再允许上载，只有在 PLC 的密码设置中清除密码才能上载；当加密的工程下载至 PLC 后，可以允许其它程序的下载，但是下载后其它程序也会变更为禁止上载状态，只有进行存储器清除或在 PLC 的密码设置中清除密码才能正常上载，这个功能主要用于保护客户原程序不被上载。

操作方法及效果如下：

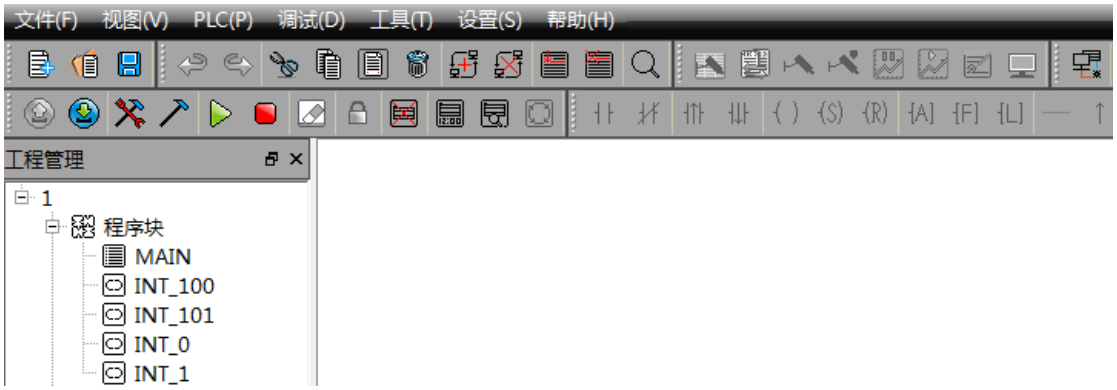
①点击 UNISYS 上位机中的文件->设置工程密码，密码设定后，再次打开这个工程，这个工程会被加密。



②打开加密的工程文件时，如果输入正确的工程密码，则可查看和修改程序内容。



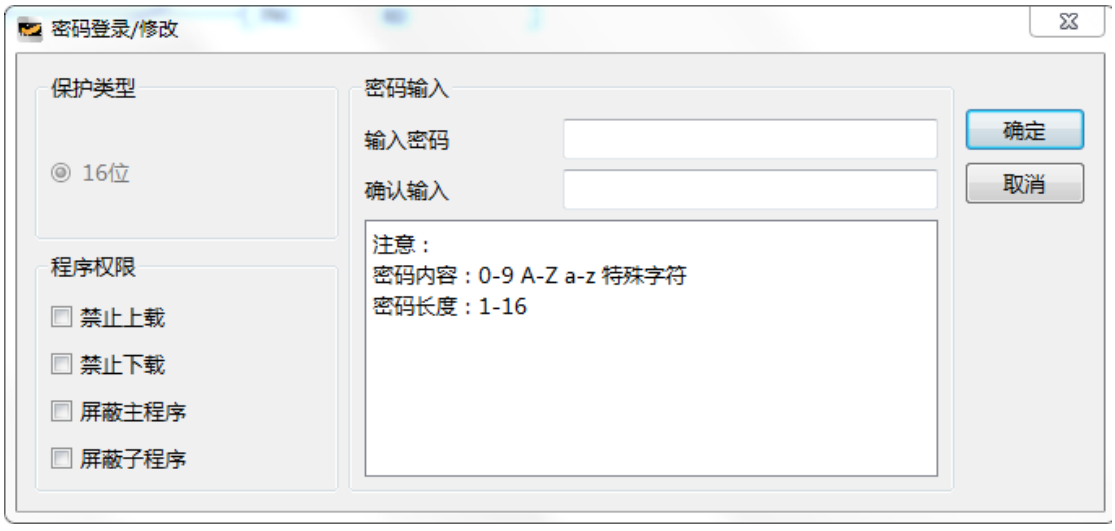
③跳过上面的密码验证过程，或者不进行文件->清除工程密码，则程序块部分中的程序全部被屏蔽，不可查看和更改，效果如下。



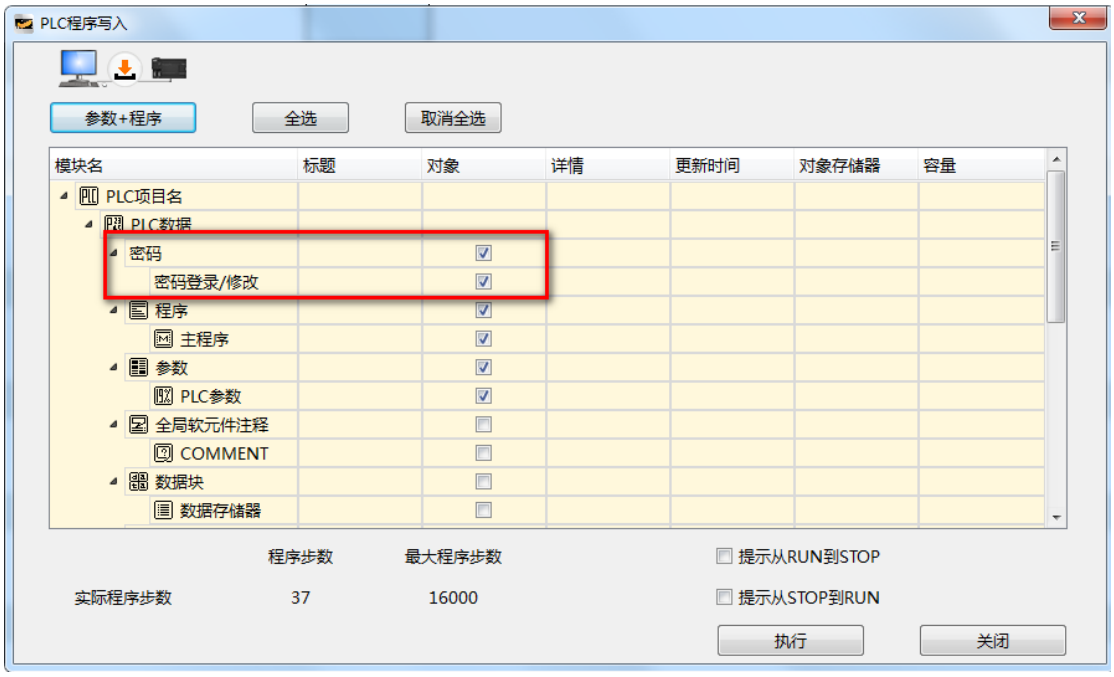
④项目工程未解密或者清除密码的情况下，将此工程下载至 PLC，PLC 会将此项目密码保存起来，并变更程序的权限为禁止上载。

2、程序加密

对用户程序进行相应权限的加密，可防止重要部分的梯形图程序被查看或变更，分为禁止上载、禁止下载、屏蔽主程序、屏蔽子程序。

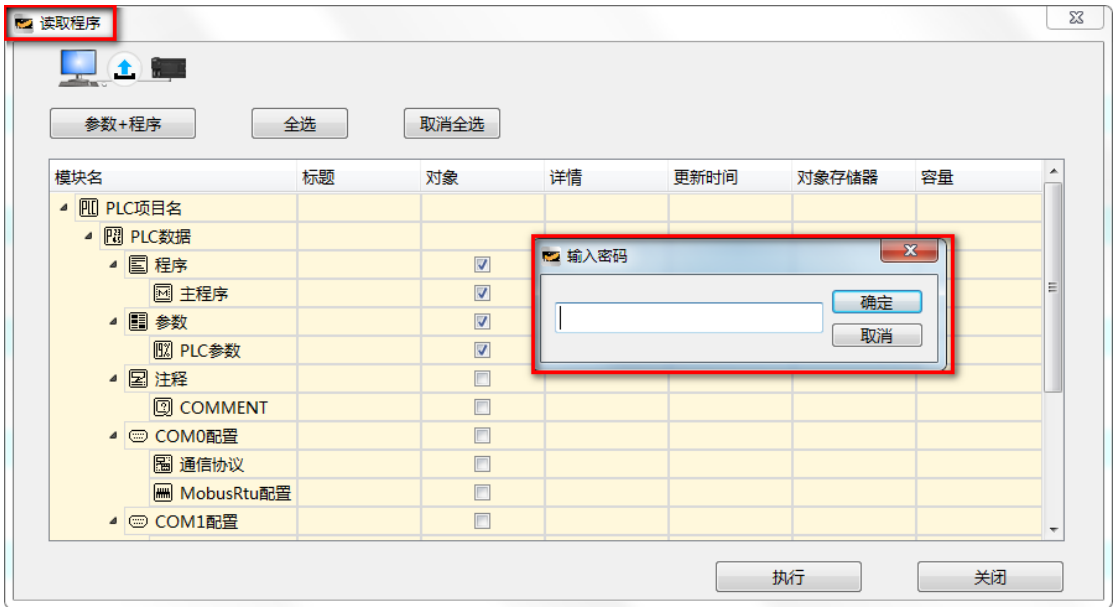


勾选对应的权限以限制程序的可查看或可变更权限，在下载时跟随程序、配置等一起将密码下载到 PLC，如果 PLC 有密码的情况下则需要先输入原密码，然后 PLC 才会变更为当前下载的密码。

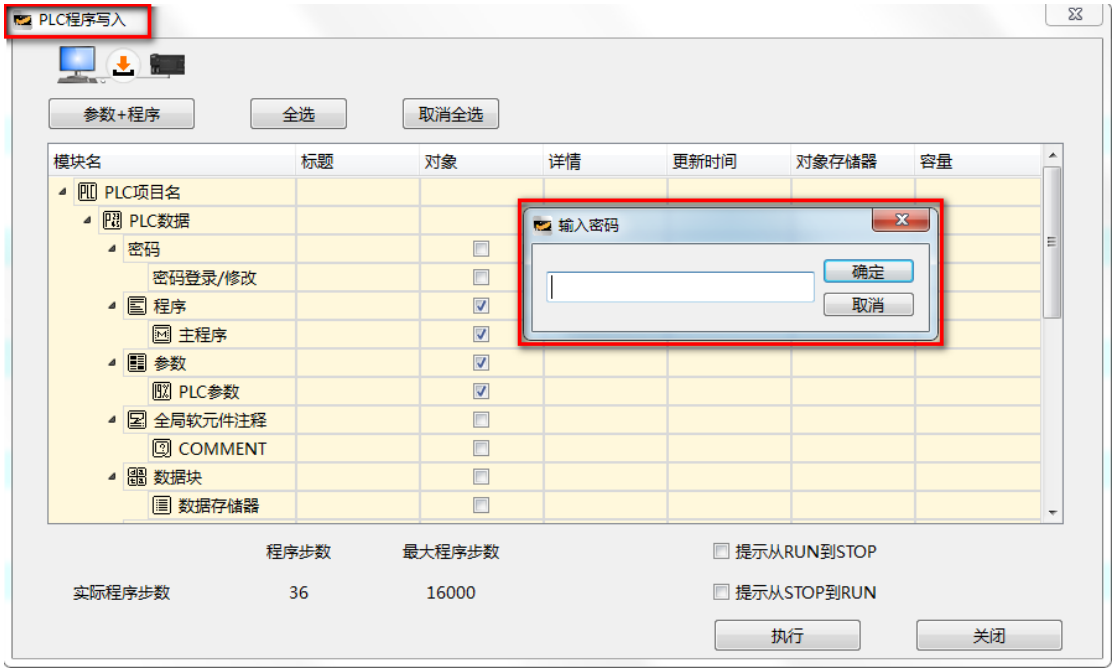


效果如下（前提是 PLC 已经进行了相应的程序权限加密）：

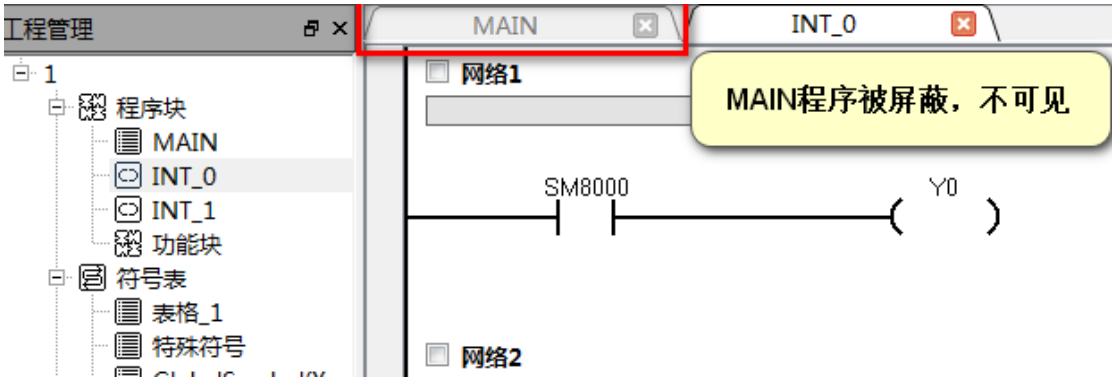
（1）禁止上载：上载时



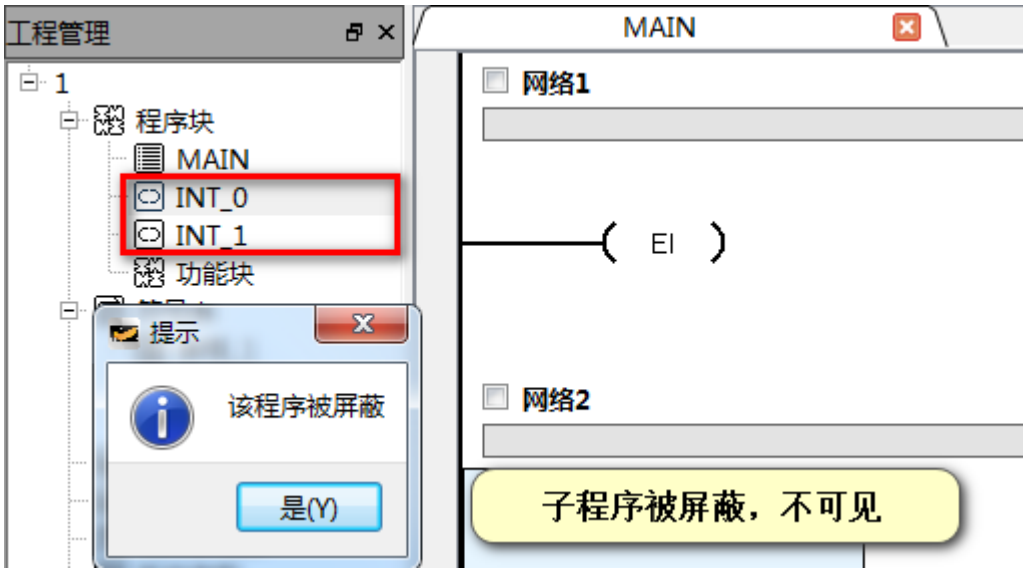
（2）禁止下载：下载时需要输入密码才可下载程序



（3）屏蔽主程序：主程序部分不可见、不可更改，子程序部分可以更改

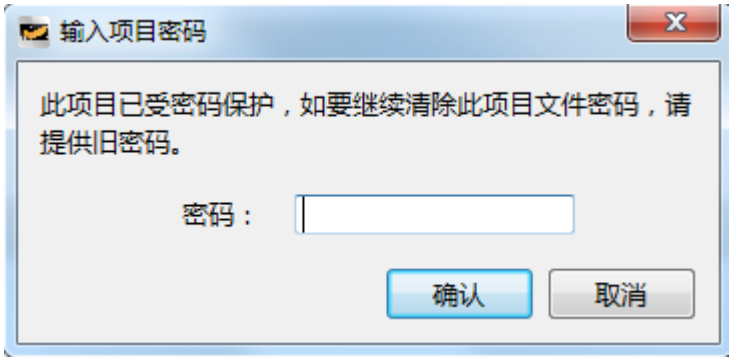


（4）屏蔽子程序：

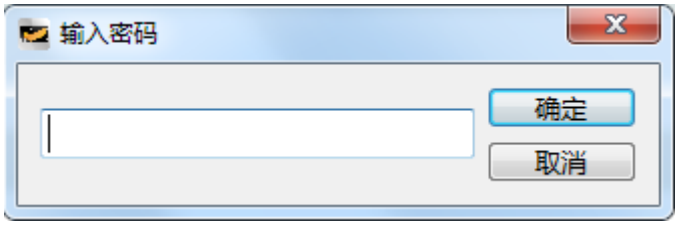


3、密码清除

（1）清除工程密码：点击 UNISYS 中文件->清除工程密码，清除密码时需要输入正确密码，清除后，项目中所有程序变为可见和可变更状态。



（2）清除 PLC 的密码：点击 UNISYS 中 PLC->PLC 密码设置->清除，清除密码时需要输入正确密码，密码清除后则存储在 PLC 中的程序权限将消失。另外 PLC 存储器清除时会清除 PLC 中存储的密码。



4、注意事项

（1）禁止上载权限不能同时与屏蔽主程序权限或者屏蔽子程序权限共同存在，其它的权限可以同时存在。

（2）密码功能主要是限制程序的读写权限，当忘记密码时，可使用存储器清除功能清除掉密码，但此时 PLC 存储的程序也会被清除。

3.5.2 PLC 高级加密功能

应用场景 1

A 和 B 合作开发一个梯形图，并各自保护自己的知识产权，A 将加密了的扩展程序工程给 B 使用（前提是 A 将秘钥下载到 PLC），但 B 无法查看和变更此部分的梯形图。同时 B 可以在 A 的工程的基础上编辑自己的梯形图且也可做相应的加密操作（将 B 的扩展程序的秘钥下载到 PLC 并加密 B 的扩展程序），这样 A 和 B 均可保护自己的知识产权。

应用场景 2

A 事先将秘钥下载到 PLC，将 PLC 给 B 使用，然后 B 就可以使用 A 提供的加密了的扩展程序（B 中带有这个扩展程序对应的秘钥），但是 B 却不能将此工程给 C 的 PLC 使用，因为 C 的 PLC 没有此秘钥。

以上场景的应用都可通过生成扩展程序块和 32 位秘钥来对梯形图进行保护。上位机通过右键程序块可以添加扩展程序块，预使该扩展程序生效，必须将对应的 32 位秘钥通过上位机->PLC 下的秘钥注入操作注入到 PLC 中，然后该扩展程序下载至 PLC 后才会生效，可以通过这个 32 位秘钥有效保护扩展程序。扩展程序块可以编写三块，附加功能有扩展程序加密、密码清除、秘钥生成导出、秘钥导入、资源使用表。

1、扩展程序加密

如果使用高级加密功能并且允许上位机随机生成秘钥，加密前需要先执行秘钥生成导出操作，然后再加密。如果使用自己用字符组合成的秘钥，加密前先执行秘钥导入。主要用于保护扩展程序块，使其不可更改和查看。

2、密码清除

主要用于开发者查看和更改扩展程序块，密码清除后扩展程序变为可见和可更改状态。

3、秘钥生成导出

主要用于生成与扩展程序块匹配的 32 位秘钥，再将这个秘钥注入到 PLC，该扩展程序块下载到 PLC 才能被识别使用。

4、秘钥导入

主要用于开发者自己编写自己的秘钥。

5、资源使用表

可编辑可保存，主要用于开发者提供资源使用情况，方便后面的开发者在有扩展程序的工程中继续编程，当然也可以不提供，不让其他客户知道初次开发者使用了哪些资源。

6、具体使用方法

扩展程序中生成导出（或导入）秘钥后，该扩展程序只能被带有此秘钥的 PLC 使用，通过下载此秘钥到 PLC，使 PLC 获得该扩展程序的使用权限。

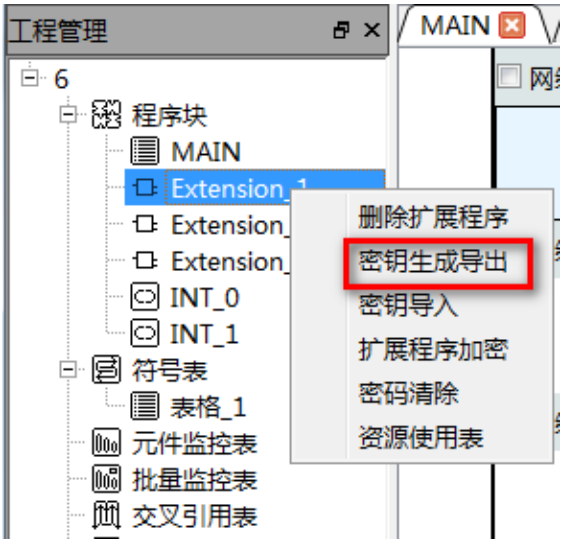
初次开发者按如下流程编写扩展程序块，并将 32 位秘钥注入到 PLC，如此该 PLC 才能使对应的扩展程序块生效。

（1）编写扩展程序

打开 UNISYS 中程序块，右键程序块插入扩展程序，使用几块就插入几块，最多三块，右键扩展程序块可插入子程序或中断子程序，所有程序的总容量大小是与系统参数设定相关的，编程规则与 MAIN 程序一致。

（2）秘钥生成导出

用上位机给扩展程序匹配随机的 32 位秘钥，并导出到文本文件*.txt，方便开发者查看具体秘钥信息。

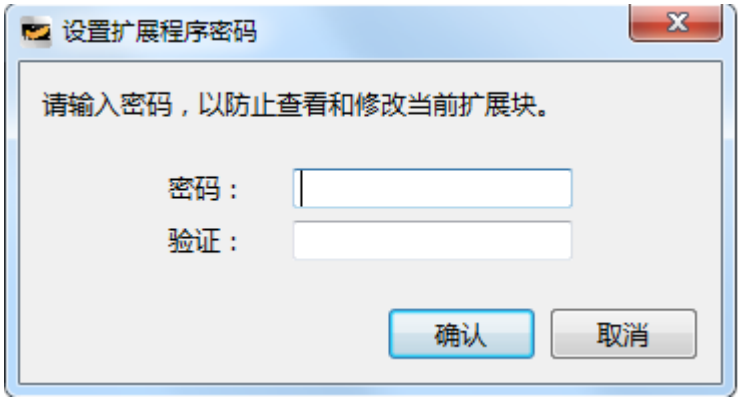


（3）秘钥导入

采用自己组合的 32 位秘钥，则跳过（2）步骤，开发者创建一个.txt 文件，将 32 位秘钥信息写入（有效字符 0~9，a~z，A~Z），可以在扩展程序中使用自己的秘钥。

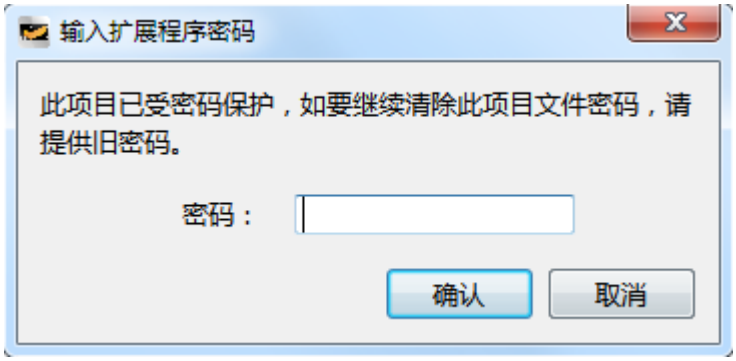
（4）加密扩展程序块

扩展程序块加密后才能被下载到 PLC 中，否则上位机不会下载扩展程序，加密后，扩展程序的屏蔽功能会生效，将不让查看和修改。



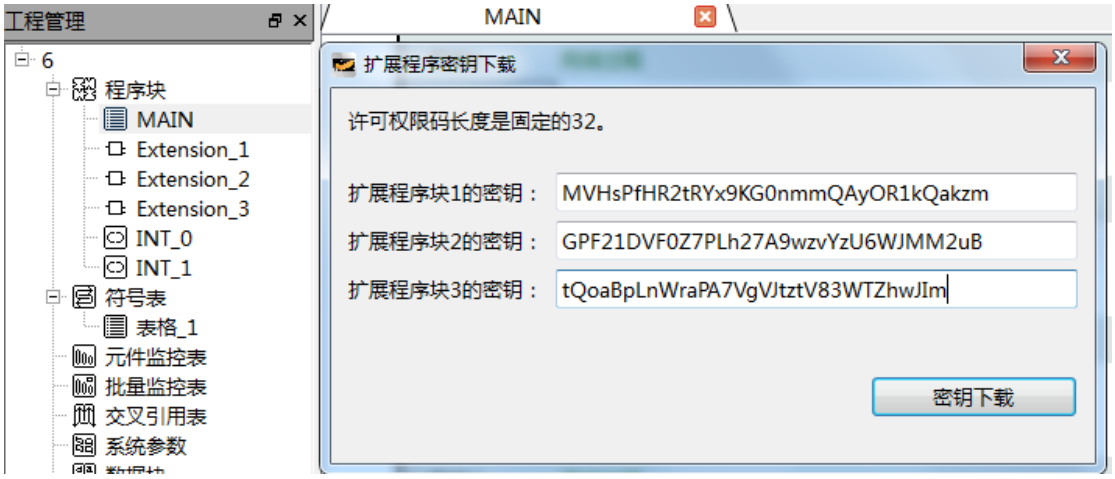
（5）密码清除

方便开发者查看和更改扩展程序。



（6）PLC 秘钥注入

使用扩展程序块之前，通过上位机->PLC 下的扩展程序秘钥下载操作先将对应的秘钥注入到 PLC，这个秘钥是开发者提前下载到 PLC 的与扩展程序匹配的秘钥。



（8）勾选程序下载，会将扩展程序块对应的工程也下载到 PLC 中

7、注意事项

（1）扩展程序块是从属于程序块的，编写梯形图是编写整个程序块，这里只是把要加密的内容放到了扩展程序块，但编程原则还是和单独编程 MAIN 程序是一致的，只是相当于把以前的梯形图程序划分为多块而已，MAIN 程序是前部分的梯形图，扩展程序则是后部分的梯形图，当然可以选择只编写扩展程序，这样就相当于加密整个梯形图数据。

（2）扩展程序编写完毕一定要先对其进行秘钥生成导出或秘钥导入后再加密，以便获得该扩展程序的使用秘钥，这个秘钥是用来下载到 PLC，使 PLC 对该扩展程序块生效用的。

（3）当 PLC 有扩展程序块时，对应扩展程序块标号的新的秘钥注入不会生效，即不允许有扩展程序块时更改秘钥，需进行存储器清除清除或程序清除时才能下载新的秘钥。

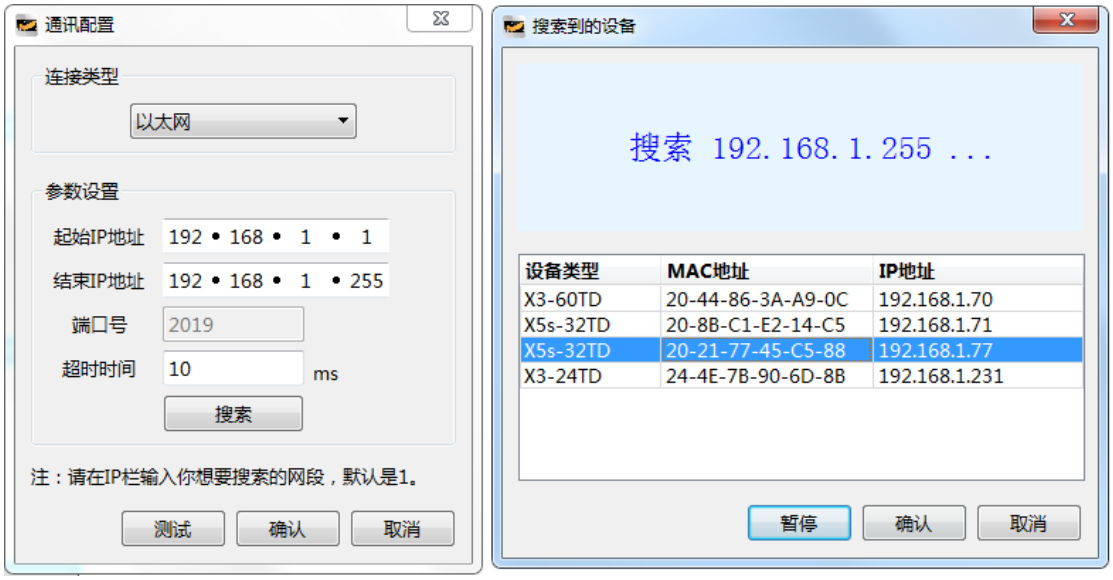
（4）当扩展程序里面的秘钥与 PLC 中的秘钥不匹配时，扩展程序会下载失败，PLC 中将只生效 MAIN 程序和下载成功的扩展程序。

（5）删除扩展程序块时需要输入密码验证，正确才能删除，同时删除里面的程序、秘钥、原密码。

3.6 以太网功能

1、以太网模式下的监控下载

选择上位机的通信配置，连接类型更改为以太网，显示一个默认的 IP 地址，点击搜索可搜索出同网段连接了以太网的 PLC IP 地址及其对应的 MAC 地址，更改网段可搜索其他网段的 PLC IP 地址。然后选择要连接的 PLC IP，点击测试，测试成功后即可进行监控下载了。支持多个安装了 UNISYS 软件的电脑监控下载或一台电脑上使用多个 UNISYS 监控下载，但是个数有限制，具体看下面的连接协议分配介绍。

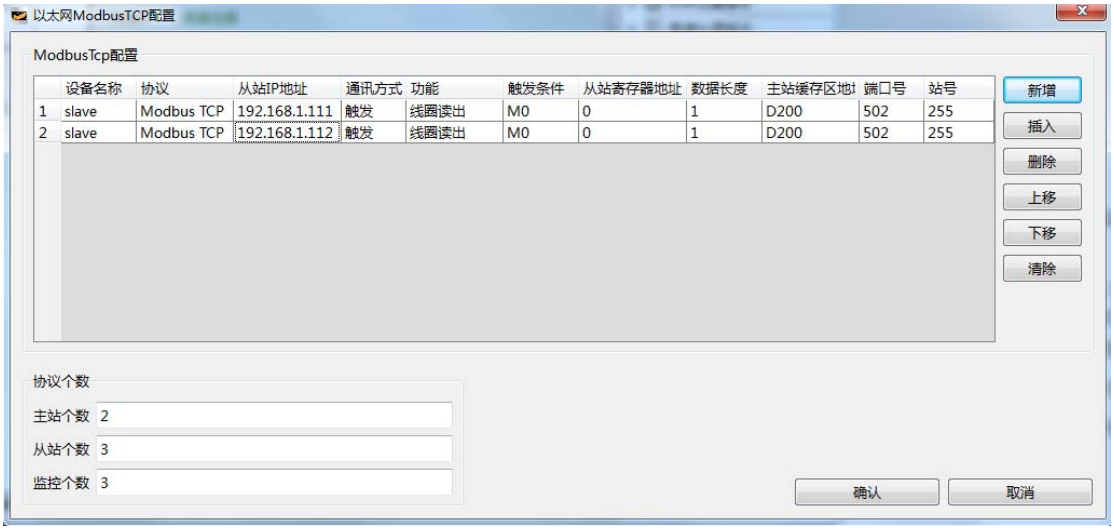


2、上下载 PLC 的 IP 的地址

上位机可以重新配置 PLC 的 IP 地址参数，PLC 默认 IP 是 192.168.1.254



3、以太网通信协议配置



（1）协议个数

根据配置表来分配连接个数，主站个数与 IP 有关，从站和监控个数可自行分配，总的连接个数不能超过八个,即最多允许 8 个不同 IP。

① 主站个数：根据配置参数中的不同 IP 来分配，有几个 IP 就分配几个主站连接个数，因为 PLC 满足八个功能码，所以允许同一个 IP 配置参数可以有八行配置表格，且在输入表格时可以任意选择配置参数，但同一 IP 的总配置表格行数不能超过 8 行

② 从站个数：表示允许多少个主站（IP 不同）来连接当前的 PLC

③ 监控个数：表示允许多少个上位机软件对该 PLC 进行监控下载

（2）MODBUS TCP 通信配置

① 序列号：对应配置表所在行数

② 协议：MODBUS TCP

③ 设备名称：用于备注 PLC 连接的从站设备名称

④ 从站 IP 地址：即 PLC 连接的服务器的 IP 地址

⑤ 通讯方式：触发表示主站必须将触发条件置位才能进行数据发送来访问从站；循环表示主站一直发送数据访问从站

⑥ 功能：包括线圈读出,输入读出,保持寄存器读出,输入寄存器读出,单个线圈写入,单个（保持）寄存器写入,批量线圈写入,批量（保持）寄存器写入

⑦ 触发条件：在通讯模式为触发的情况下，需要将对应的条件标志位置位才能通信，通信超时或者通信完毕后会复位触发条件

⑧ 从站寄存器地址：主站要访问的从站对应的软元件地址，地址分配与串口 MODBUS 软元件地址分配一致

⑨ 数据长度：根据选择的功能来限制数据长度。线圈读出、输入读出：1~2000；保持寄存器读出、输入寄存器读出：1~125；单个线圈写入、单个（保持）寄存器写入：固定0；批量线圈写入：1~1968；批量（保持）寄存器写入：1~123

⑩ 主站缓冲区地址：存储从站回传的数据信息

11 端口号：MODBUS TCP 通信用，固定 502

（3）特别说明

① PLC 掉电时会保存上次的 MODBUS TCP 参数，下次会根据配置进行通信；

② 当分配的监控连接个数为 0 且将该分配连接下载至下位机时，上位机的以太网模式下将不能进行监控下载功能；

③ MODBUS TCP 参数配置表最大可配置 64 行，只允许 8 个不同 IP，且每个 IP 配置表不能超过 8 行，主站连接个数要与配置表对应，否则不允许这样的配置

④ 不同从站（IP 不同）的触发条件必须不同，上位机会进行限制

⑤ 使用 PLC 存储器清除功能时会清除 PLC 中存储的 MODBUS TCP 参数配置表

4、以太网相关特殊寄存器

元件	功能
SM8400（可读写）	以太网功能控制位 0：开启以太网功能 1：禁用以太网功能
SM8401（可读写）	MODBUS TCP 功能控制位 0：开启 MODBUS TCP 功能 1：禁用 MODBUS TCP 功能
SM8402（可读写）	从站不在线标志 0：从站都在线，此时会复位 SM8402 1：有从站不在线，任意从站不在线时置位
SM8403（可读写）	PLC 网线未连接或断开 0：网线连接正常 1：网线未连接或断开
SD8400（只读）	显示 PLC 当前分配协议连接个数 0：连接个数分别为 3 个监控，4 个从站，0 个主站

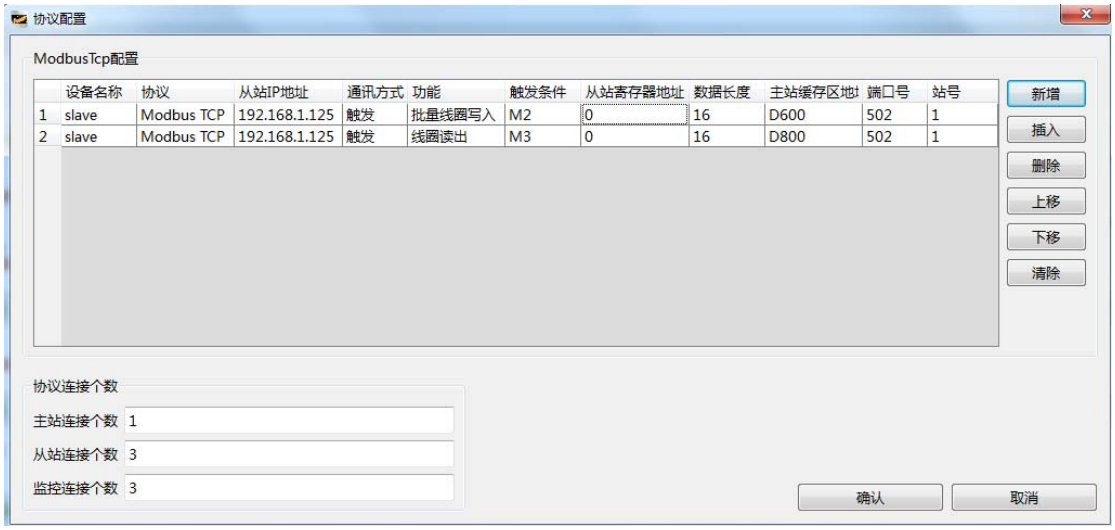
	16 进制数：11-8bit，7-4bit，3-0bit 组成的四位数分别代表监控、从站、主站连接个数 （SD8400 = H340 表示监控连接 3 个，从站连接 4 个，主站连接 0 个）
SD8401（只读）	显示当前 TCP 连接数量（主+从+监控连接的总个数）
SD8402（可读写）	不在线从站对应的 IP（第四段），当 SM8402 置位时查看，此时需要重新打开服务器（从设备）才能再次建立通信，通信建立成功后会清除该寄存器，如果值不为 0，可能是还有其他的设备未正常连接或者该设备重启失败，默认值 0
SD8403（可读写）	显示 MODBUS TCP 错误码 0 表示无错误 -1 表示 SD8400 值设置不正确，此时会使用默认连接配置 PLC 作从站时的报错（错误码组成：高八位为站号，低八位为出错类型）：此情况表示主站发送给 PLC 的数据帧错误 出错类型： 0x01 表示非法功能 0x02 表示非法数据地址，从地址设置错误 0x03 表示非法数据值，数据长度设置错误或者主站缓冲区数据设置错误 PLC 作主站时的报错（错误码组成：高八位为 IP，低八位为出错类型）：此情况表示从站返回给 PLC 的数据帧错误 出错类型： 0x11 表示非法功能 0x12 表示非法数据地址 0x13 表示非法数据值
SD8404（可读写）	设定超时时间（ms 单位），默认 0 时代表设定超时时间为 1000ms

SD8405（可读写）	保存当前最新的超时 IP 及超时数据所在配置表中的位置（高八位显示 IP，低八位显示该 IP 具体超时的数据在配置表中的位置）
SD8406（可读写）	尝试重连时间间隔（从站掉线后每次主站尝试重连从站的时间间隔，ms 单位），为 0 时，默认为 100ms
SD8409（不允许使用）	烧录 mac 地址专用寄存器，为-1 时，表示需要烧写 mac 地址，
SD8410~SD8413	本机 IP1~4 段

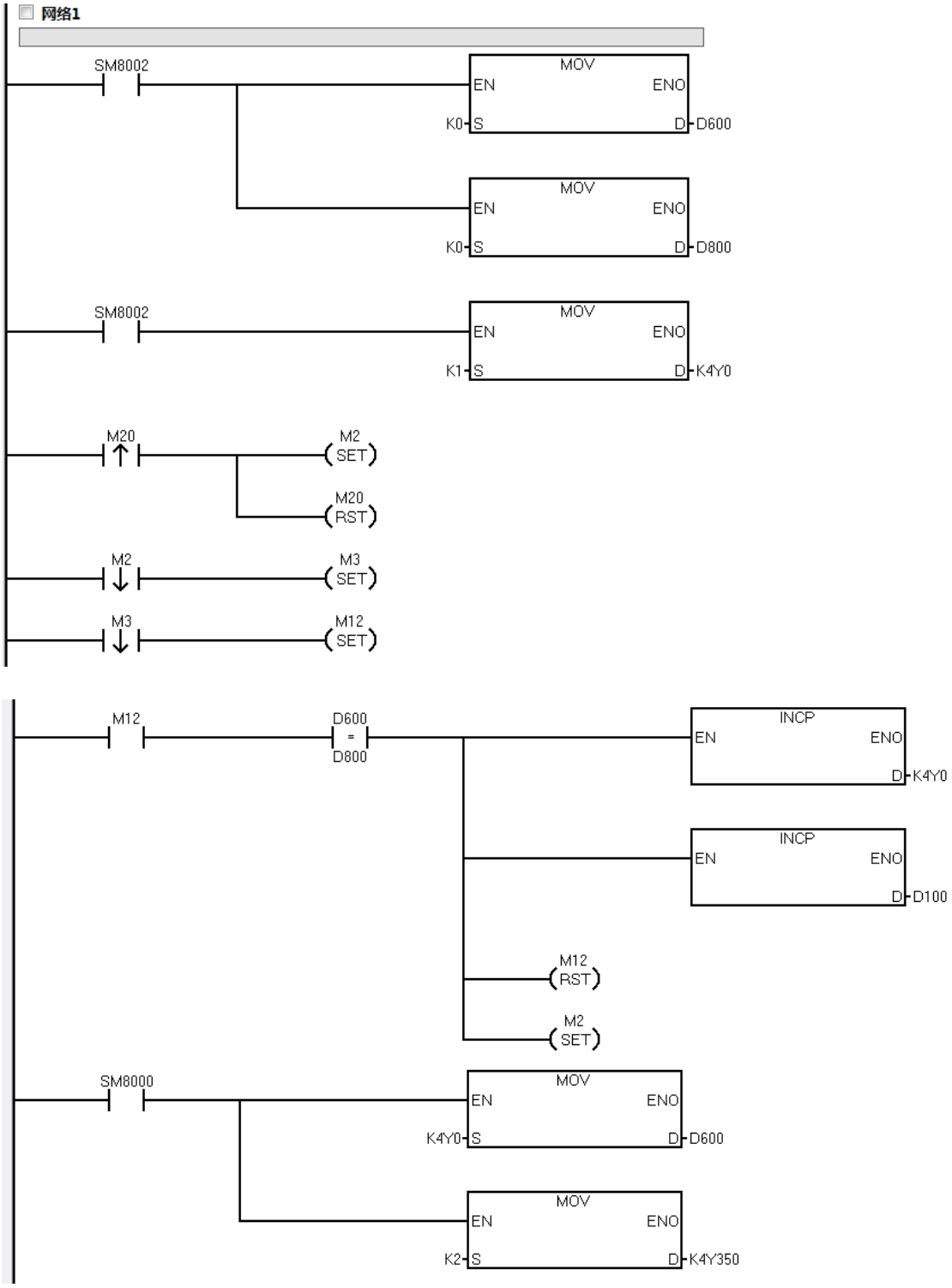
5、MODBUS TCP 触发方式协议配置举例

用触发轮询方式对 IP 为 192.168.1.125 的从站进行线圈读写操作。

（1）下载协议配置



（2）编写梯形图操作触发条件，使其可以按照程序流程进行通信



（3）启动 M20，可看到主站 PLC 上的输出 LED 开启跑马灯模式。

（4）通信流程解析

①PLC 会根据配置表中的配置功能一行行顺次执行，执行完整个表后就再次从第一行配置顺次执行，

②哪个触发条件使能了就执行哪一条配置行的功能，也是遵循①中的执行顺序

③触发功能执行完了或者发生了超时现象后会复位触发位，可根据此触发位的复位来判断当前配置行执行结束，然后用沿跳变方式来执行下一个功能配置行，按此方式可完成整个表的触发轮询

④如果配置表中选择循环功能时，PLC 会一直按配置表顺次执行，一直进行通信功能，触发方式只是可以根据用户的流程需求来选择什么时候进行通信。

6、以太网功能使用注意事项

（1）当 PLC 之间不能正常 MODBUS TCP 通信时，首先检查 PLC 的 IP 问题，是否有 IP 冲突或 IP 配置不正确，再检查网线是否正常、网络是否稳定，再检查配置表是否配置出错

（2）PLC 之间的网线直连通信，中间通过交换机连接或者采用交叉网线方式，交叉网线接线方式如下



（3）循环方式和触发方式的使用建议如下

• 循环通信:需要反复、尽快读写从站内变化比较快的数据，例如读取变频器的运行频率、运行状态、输入端口状态;在过程控制应用系统中，实时通信改写变频器的运行频率、输出端口的状态，等等，可选择“循环”通信方式。PLC 在执行用户程序时，会反复扫描执行通信配置表中的所有“循环” 配置项。

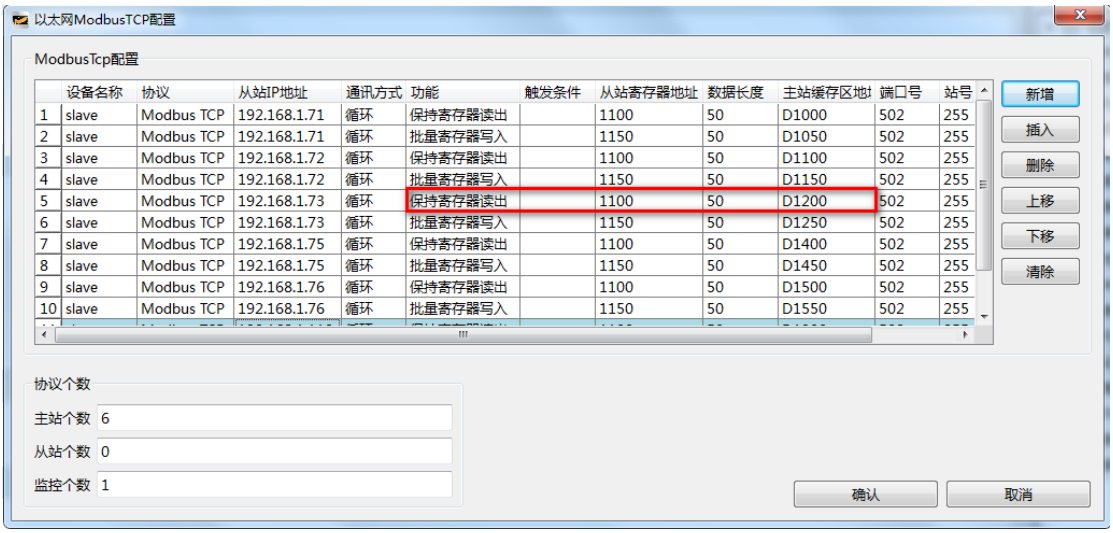
• 触发通信:需要定时读或写从站的刷新速度较慢的数据，例如读取变频器的输出电流、输出功率、当前故障告警信息，等等，可选择“触发”通信方式，在用户程序中,每置位触发条件 1 次，即会引起通信配置表中对应的通信项的 1 次通信操作;用户程序中定时置位该触发条件，即可实现所需频度的通信读写操作。

• 通信方式设定的建议:根据所需要交互参数刷新的特性要求进行合理选择，可使得通信表现大为改善，不要为了编程简单，将所有通信项都设置为“循环”项，这样可能因循环项太多，而使得数据交互及时性降低，影响系统的控制效果。将一些不重要的数据访问，安排为“触发”项，根据轻重缓急来触发通信，可大为改善通信及时性效果。

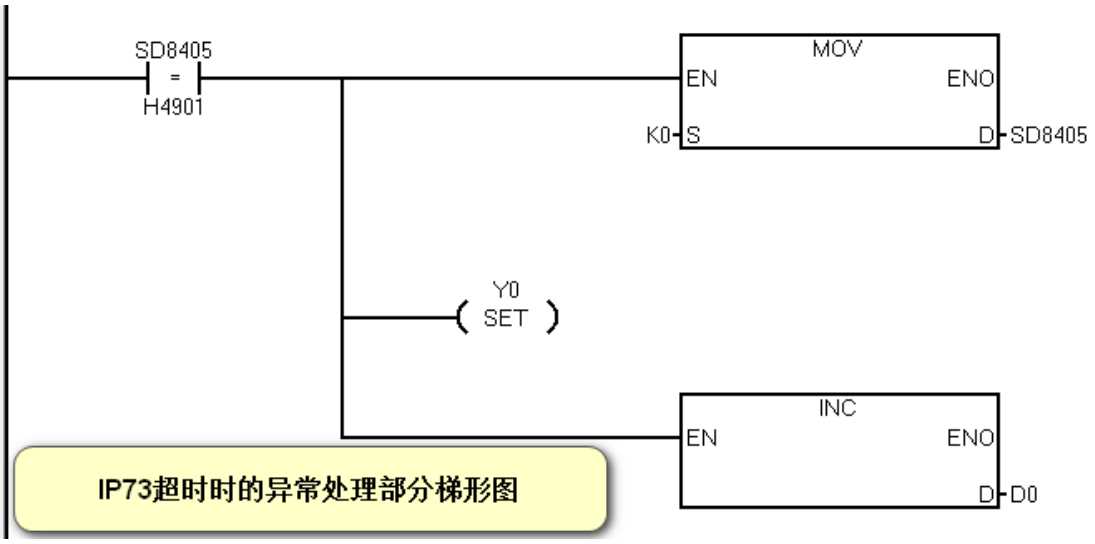
（4）通信超时时的异常处理

是通信就会出错导致超时，如果使用者使用以太网通信会发生的错误会对设备运行产生重大事故，举例说明规避方法：

IP73 的第一个功能保持寄存器读出功能读出的数据是特别重要的数据，不允许超时导致数据不刷新而造成设备运行故障，则处理方法如下。



IP73 对应的是第三个 IP，超时时 SD8405 高八位显示 IP，低八位显示超时的功能数据所在该 IP 下的位置，SD8405 是实时刷新的。



3.7 扩展模块的使用

3.7.1 扩展模块的 ID

1、模块 ID 组成规则

字节 1(H)								字节 2							
t	t	t	t	N	N	A	A	I	I	I	I	Q	Q	Q	Q
t：模块类型标识								I：输入标识							

0000 - 普通模块 0001 - 模块(RTD) 0010 - 模块(TC) 0011 - 模块(PID) ... 其他 - 保留 N： 禁用 A：I/O 类型标识 00 - 无 01 - 数字 10 - 模拟 11 - （没有混合模块）	0000 - 无输入 0001 - 1AI 或 4DI 0010 - 2AI 或 8DI 0011 - 4AI 或 16DI 0100 - 8AI 或 32DI 0101 - 16AI 或 64DI Q：输出标识 0000 - 无输出 0001 - 1AO 或 4DO 0010 - 2AO 或 8DO 0011 - 4AO 或 16DO 0100 - 8AO 或 32DO 0101 - 16AO 或 64DO

2、模块分类及对应的 ID

模块	ID（16 进制）	说明
8DI	0120	8 点输入数字量
16DI	0130	16 点输入数字量
32DI	0140	32 点输入数字量
8DO	0102	8 点输出数字量
16DO	0103	16 点输出数字量
32DO	0104	32 点输出数字量
2AI	0220	2 通道输入模拟量
4AI	0230	4 通道输入模拟量
8AI	0240	8 通道输入模拟量
2AO	0202	2 通道输出模拟量
4AO	0203	4 通道输出模拟量
8AO	0204	8 通道输出模拟量
8DI/8DO	0122	已有，8 点输入/8 点输出数字量

16DI/16DO	0133	16 点输入/16 点输出数字量
4AI/2AO	0232	已有，4 通道输入/2 通道输出模拟量
4RTD	1230	4 输入热电阻模拟量模块
8TC	2240	TC 模块，8 通道热电偶输入模拟量
8TC-PID	3240	特殊模块，8 通道热电偶输入模拟量，PID 模块
...	...	...

3、与模块 ID 相关的特殊寄存器

SD8421	SD8422	SD8423	SD8424
模块 1 的 ID	模块 2 的 ID	模块 3 的 ID	模块 4 的 ID
SD8425	SD8426	SD8427	
模块 5 的 ID	模块 6 的 ID	模块 7 的 ID	

ID 显示：

正常情况下显示模块 ID，异常时，从未连接显示 H0000(0),断线显示 HFFFF(-1),组态错误显示 HFFFE(-2)。

总线总是尝试获取存在过的最大连接数，即当上电后存在过 7 个模块时，移除 7 就会认为 7 断线。

3.7.2 数字量扩展模块

数字量模块不足一个字节的（如 4DI）按照一个字节算。且扩展模块的端口编号总是从 8 进制为 0 开始。XPLC 挂接模块的数字量输入输入从本体后面的地址开始计算，最多到 X367/Y367(X370~X377 预留给 BD 板)。上电的时候根据 ID 可以得到每个槽对应的数字量字节数。从而可以知道各个槽对应的 X/Y 的编号。

3.7.3 模拟量扩展模块

模拟量扩展模块在 PLC 内的存储区域，固定放在 D7500~D7999 里面。模拟量的数据包括状态寄存器、数据寄存器、配置寄存器、附加寄存器（一般情况下数字量没有配置寄存器和附加寄存器），配置寄存器用于配置模块各个通道的量程，增益，极性等信息。附加寄存器用于一些特殊用途，如：配置保存、校准等。每个模块占用 50 个 D（多的为预留）寄存器。且每组第一个为状态寄存器。具体地址如下：

状态寄存器	R 区（模拟量输入数据区）	W 区（模拟量输出数据区）	C 区（配置寄存器）	A 区（附加寄存器）
D7500	D7501~D7508（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7509~D7510	D7511
D7550	D7551~D7558（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7559~D7560	D7561
D7600	D7601~D7608（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7609~D7610	D7611
D7650	D7651~D7658（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7659~D7660	D7661
D7700	D7701~D7708（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7709~D7710	D7711
D7750	D7751~D7758（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7759~D7760	D7761
D7800	D7801~D7808（①没有 R 区 ②没有 W 区 ③R、W 都有）		D7809~D7810	D7811

注意：

(1) 当使用模块时，对应模块中使用的寄存器请勿在梯形图中再次使用，否则会出现问题。

(2) 数据区的 R 区、W 区要根据模块类型及输入输出点数来区分，包含表中描述的三种情况。

①比如 8AO，W 区有 8 个，没有 R 区；②比如 8AI，R 区有 8 个，没有 W 区；③比如 4AI2AO，则 R 区有 4 个寄存器，W 区有 2 个寄存器。

3.7.4 模块寄存器参照表

寄存器对照表：

偏移	作用	说明
0	状态寄存器	参照状态寄存器说明
1	AI0	AI 通道 0
2	AI1	AI 通道 1
...	...	...
n	AIn	AI 通道 n
n+1	AO0	AO 通道 0
n+2	AO1	AO 通道 1
...	...	...
8	AOx	AO 通道 x
9	配置	根据 AI 通道数来选择使用 1 个或 2 个配置寄存器，4 个通道占用 1 个配置寄存器
10		
11	附加	校准及参数保存

注：1 个状态寄存器，8 个数据寄存器（1~n 表示 AI 通道数据，n+1~8 表示 AO 通道数据），2 个配置寄存器，1 个附加寄存器

状态寄存器说明如下：

bit	内容
Bit15-Bit8	/
Bit7	内部断线或底板断线
Bit6-Bit4	/
Bit3	未校准
Bit2	命令错误
Bit1	配置错误
Bit0	内部通讯错误

注意:该状态字对 8DI8DO 模块无效

配置寄存器详解：

bit	作用	说明
15	AI(x+3)极性控制	0：双极性；1：单极性
14-12	AI(x+3)增益控制	000、001、010 分别对应 1、2、4 倍增益
11	AI(x+2)极性控制	0：双极性；1：单极性
10-8	AI(x+2)增益控制	000、001、010 分别对应 1、2、4 倍增益
7	AI(x+1)极性控制	0：双极性；1：单极性
6-4	AI(x+1)增益控制	000、001、010 分别对应 1、2、4 倍增益
3	AIx 极性控制	0：双极性；1：单极性
2-0	AIx 增益控制	000、001、010 分别对应 1、2、4 倍增益

注意：

（1）配置寄存器根据 AI 通道数分配，如果是 4AI2AO 模块，只有 1 个配置寄存器，则 x = 0，配置寄存器就是 AI3-AI0 的配置；如果是 8AI 模块，有两个配置寄存器，x 分别为 0,4，第一个配置寄存器就是 AI3-AI0 的配置，第二个寄存器就是 AI7-AI4 的配置。

（2）写完配置寄存器后需要使能配置开关

附加寄存器详解：

bit	作用	说明
15-12	附加命令	见附加命令说明表
11-8	/	/
7-4	/	/
3-0	通道选择	/

注意：写完需要使能配置开关

附加命令说明：

值	作用	说明
0x0	/	/
0x1	校准初始化	清除校准状态
0x6	AI 0 值校准	/
0x7	AI +值校准	/
0x8	AI -值校准	
0xA	参数保存	保存参数到 FLASH

R 区数据寄存器数据说明：

寄存器值-32000~+32000

增益倍数	对应电压
1	-10V~10V
2	-5V~5V
4	-2.5V~2.5V

AI 电压校准说明：

校准系数	对应电压
0	0V
+	+10V
-	-10V

W 区数据寄存器数据说明：

寄存器值-32000~+32000，对应-10V~10V

3.7.5 模块配置使能开关

address	模块 1	模块 2	模块 3	模块 4	模块 5	模块 6	模块 7
配置使能	SM8431	SM8432	SM8433	SM8434	SM8435	SM8436	SM8437

命令使能	SM8441	SM8442	SM8443	SM8444	SM8445	SM8446	SM8447

使能后，发送字的值形式为 0XDCBA，其中 D 表示命令，C 表示量程，B 表示增益，A 表示通道，写配置和读配置共用一个寄存器。配置标志使能后，禁止读取模块，只将配置寄存器的值发往模块，再立即自动把配置标志清除，然后等待模块稳定后解除禁止读取，最后 run 的时候读上来的配置与设置的一样，说明配置成功。按照配置 > 校准 > 输出的优先级，三个动作只能执行一个。配置成功后会立刻将配置参数写入 FLASH。

将命令码 HA000 写入对应的寄存器，可以保存所有量程的校准参数，校准时切换到对应的量程下操作即可，命令码 H1000 会清除全部校准参数,

AI 校准时只需要通道 0 接线，四个通道校准系数相同(零值、满值、负满值对应命令码 H6000，H7000，H8000)。AI 输入不再限制在 32000

3.7.6 总线报错和模拟量模块指示灯

错误码寄存器 SD8420,错误标志 SM8420

错误码形式 A00E

A: 地址编号，例如模块 7 断线就是 7003

E: 错误码(2：连接错误，3：断线，4：组态错误，B：配置不匹配)

模拟量模块指示灯:

PW: 电源

COM: 模块内部通信错误

ERR: 读写异常(不正确的命令或参数配置)

3.7.7 断线恢复

暂不支持热插拔，更换模块，必须断电重启。

3.7.8 模块配置举例

例 1（手动配置模块配置寄存器）：

下面以 4AI2AO 配置为例说明模块配置方法，假设 4AI2AO 模块是第 1 个模块，该模块有 4 个模拟量输入通道（对应 AI3-AI0），2 个模拟量输出通道，对应的配置使能标志位 SM8431，命令使能标志是 SM8441。

模块相关操作使能开关

address	模块 1	模块 2	模块 3	模块 4	模块 5	模块 6	模块 7
配置使能	SM8431	SM8432	SM8433	SM8434	SM8435	SM8436	SM8437
命令使能	SM8441	SM8442	SM8443	SM8444	SM8445	SM8446	SM8447

（1）状态寄存器 D7500 可观察当前模块状态

bit	内容
Bit15-Bit8	/
Bit7	内部断线
Bit3	未校准
Bit2	命令错误
Bit1	配置错误
Bit0	内部通讯错误

（2）数据寄存器 D7501~D7504 可观察当前的 4 个模拟量输入值，D7505~D7506 可观察当前的 2 个模拟量输出值

默认的配置情况下（双极性、1 倍增益）-32000~32000 对应-10V~+10V

增益倍数	对应电压
1	-10V~10V
2	-5V~5V
4	-2.5V~2.5V

（3）配置寄存器 D7509

D7609 对应 AI3-AI0 的极性和增益控制，4bit 对应一个通道

bit	作用	说明
15	AI3 极性控制	0：双极性；1：单极性

14-12	AI3 增益控制	000、001、010 分别对应 1、2、4 倍增益
11	AI2 极性控制	0：双极性；1：单极性
10-8	AI2 增益控制	000、001、010 分别对应 1、2、4 倍增益
7	AI1 极性控制	0：双极性；1：单极性
6-4	AI1 增益控制	000、001、010 分别对应 1、2、4 倍增益
3	AI0 极性控制	0：双极性；1：单极性
2-0	AI0 增益控制	000、001、010 分别对应 1、2、4 倍增益

如果要将 AI0 配置成双极性 2 倍增益，其他通道默认，则将 D7509 的 3-0bit 写为 0001，其他位写为 0000 0000 0000 即将 H1 写入 D7509，然后将配置使能标志置位，这里是置位 SM8431，可看到输入的电压 5V 对应的数据 D7501 = 16000 会变为 32000，变为 2 倍。

注意：使能配置以后，新的配置会自动保存起来，写完配置寄存器要使能对应的配置使能标志。

（4）写输出

操作 W 区，模拟量直接操作 W 区寄存器既可，4AI2AO 模块的 AO0 和 AO1 对应 D7505 和 D7506。如果是数字量模块，则写输出通过 Y 继电器来实现。

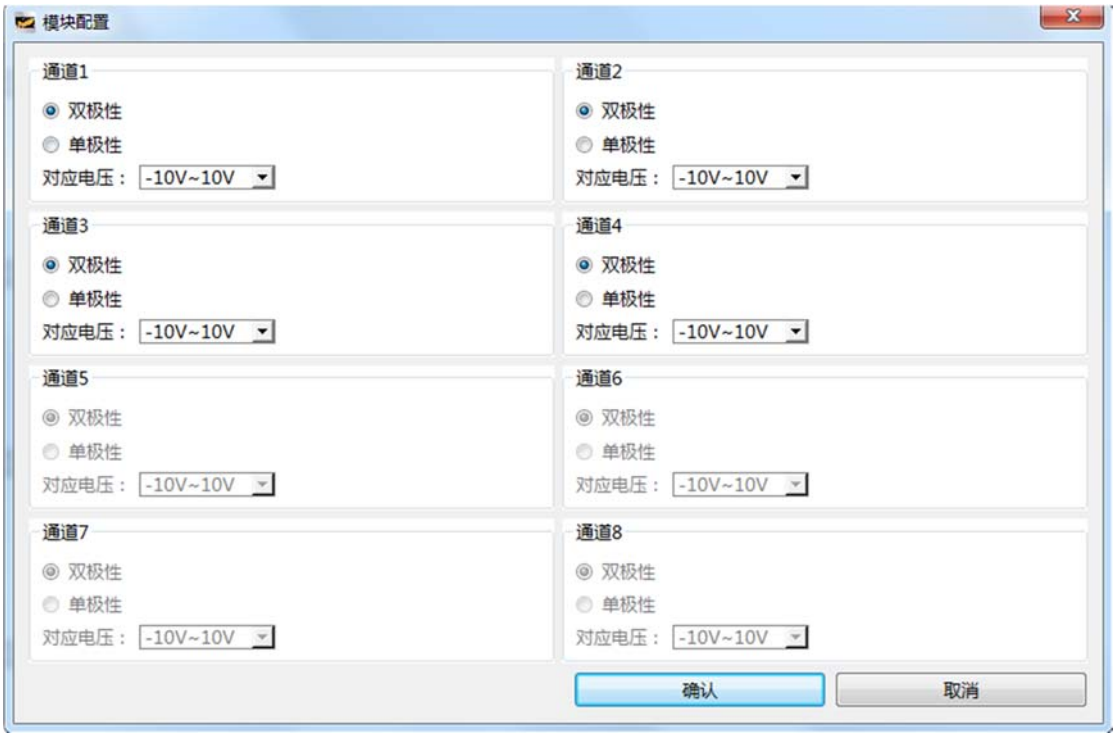
例 2（上位机 UNISYS 模块离线配置）

先将此配置下载到 PLC，然后再去连接模块，或者根据已经连接到 PLC 的模块来进行相对应的配置，统称为模块离线配置。

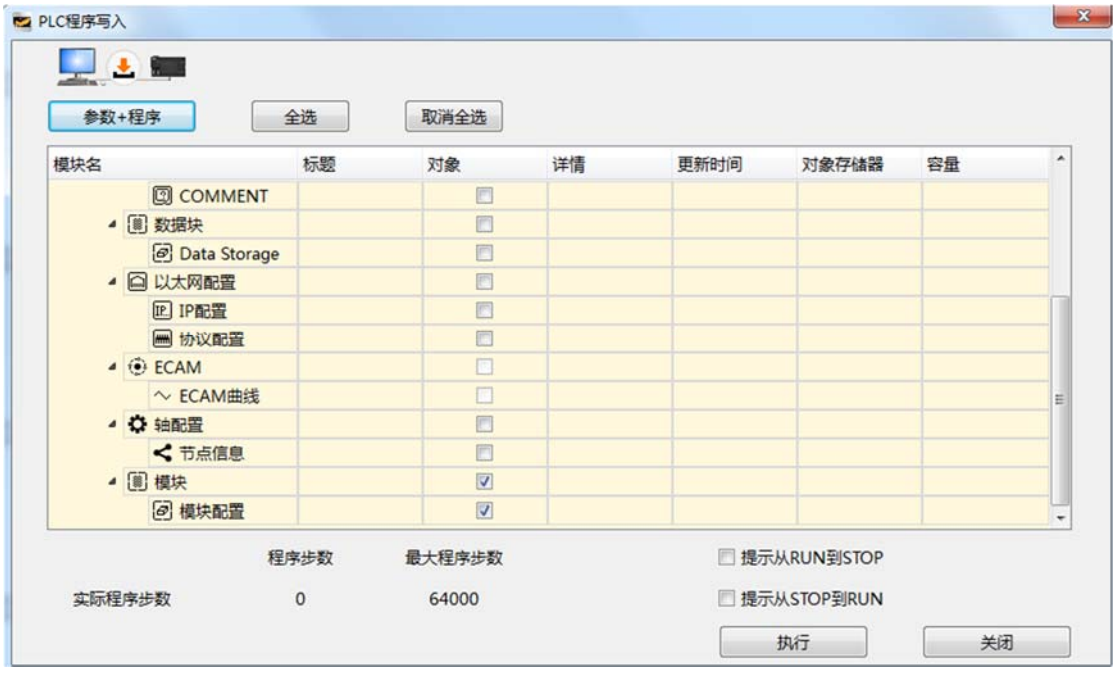
（1）打开上位机 UNISYS 中的模块配置，选择将要连接到 PLC 的模块



（2）双击对应模块编辑参数，这里双击第一行配置

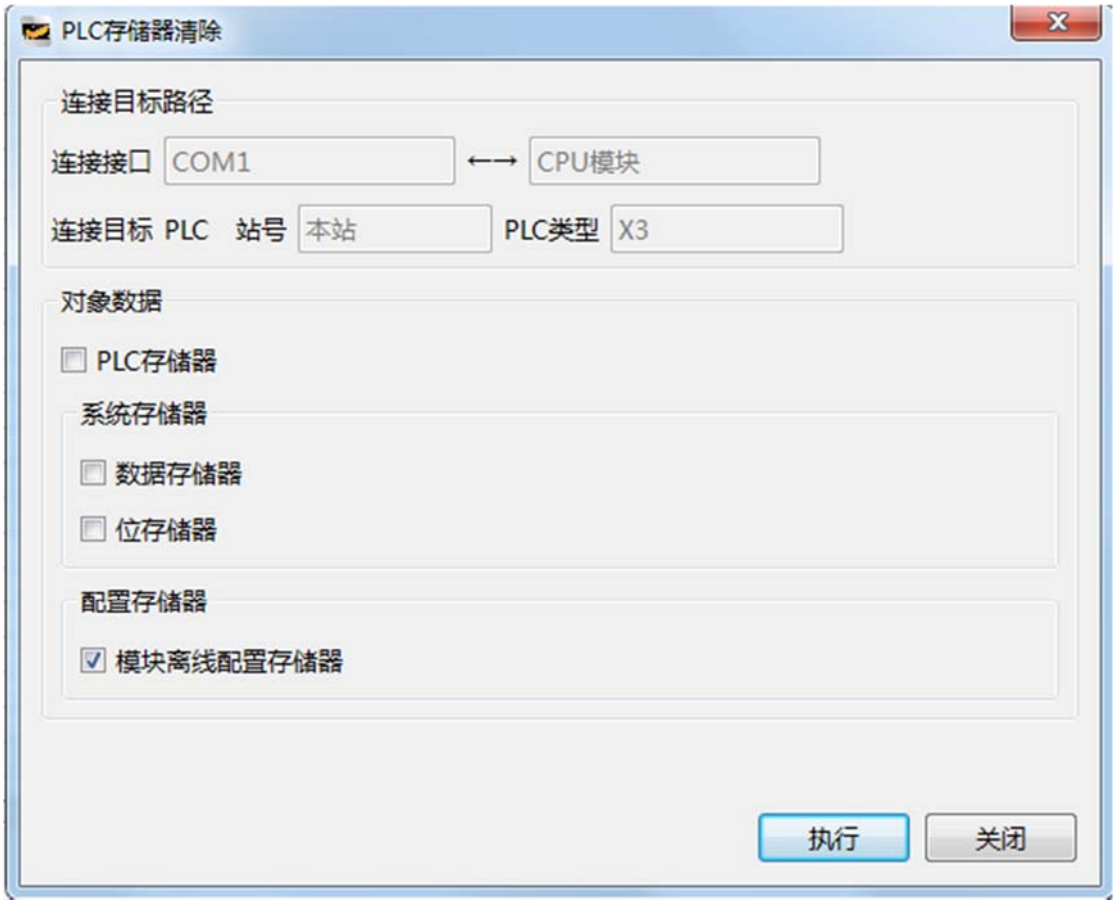


（3）下载模块配置到 PLC，即可完成模块离线配置



（4）以上全部操作成功，即可正常使用模块。

（5）当通过上位机形式对模块进行配置后，如果想再次写模块寄存器的方式进行手动配置，需要先清除当前的离线配置。



（6）UNISYS 同时具有在线模块读出、离线模块读出功能，在线读出的是实际连接在 PLC 上的模块，离线读出的是上位机离线配置下载的配置。

（7）如果离线组态配置和当前接在模块上的组态不匹配，会报错（SD8420 的错误码部分为 B：配置不匹配），然后当前模块的配置则采用之前存储的配置。

3.7.9 模块配置注意事项

（1）在线配置是指梯形图或寄存器方式直接更新配置到模块 FLASH 中；离线配置保存在 PLC 中,在下载后或重新上电时更新配置到模块 FLASH 中。

（2）注意在线配置和离线配置的优先级，不建议同时使用在线配置和离线配置。离线配置在上电时或离线配置下载后更新模块配置，在线配置在配置时更新模块配置。具体配置以最后一次更新的为准。

3.8 BD 板模块的使用

3.8.1 BD 板的 ID 及版本说明

1、目前有 4 款可用 BD 板模块，BD 板分类及对应 ID 如下。

特殊寄存器	数值	说明
SD8428	H1240	数字量，2DI4DO
	H1600	数字量，6DI
	H1060	数字量，6DO
	H2401	模拟量（电流），4AI
	H2202	热电阻，2RTD
	H2210	模拟量（电压），2AI1AO
	H230A	（电流+RTD），1AI1RTD
	H2020	模拟量 2AO

2、BD 板版本寄存器

SD8438	100	10 进制 100 表示当前 BD 板软件的版本号为 V1.00

3.8.2 数字量 BD 板使用说明

数字量的 DI0-DI7 点对应 PLC 的 X370-X377，DO0-DO7 点对应 PLC 的 Y370-Y377。上电的时候根据 ID 可以得到各个槽对应的 X/Y 的编号。

DI0-DI7	DO0-DO7
X370-X377	Y370-Y371

3.8.3 模拟量 BD 板使用说明

1、BD 板配置命令说明

命令形式：0xDCBA		说明
15~12bit	0-1	0000：配置第2通道为两线制/四线制 0001：配置第2通道三线制接法
11~8bit	0-1	0000：配置第1通道为两线制/四线制 0001：配置第1通道三线制接法

7~4bit	0-3	0000：配置第2通道为PT100类型 0001：配置第2通道为PT200四种类型 0010：配置第2通道为PT1000四种类型 0011：配置第2通道为Ni1000四种类型
3~0bit	0-3	0000：配置第1通道为PT100类型 0001：配置第1通道为PT200四种类型 0010：配置第1通道为PT1000四种类型 0011：配置第1通道为Ni1000四种类型

2、BD 板状态寄存器说明

地址	分别表示两个BD板的运行状态
bit0	1: 通信错误　0:正常
bit1	1：配置错误　0:正常
bit2	1: 命令错误　0:正常
bit3	1: 未校准　　0:正常
Bit4~15	预留

3.8.4 模拟量 BD 板地址对照表

1、4AI BD 板的地址对应表

地址	说明
SD8428	H2401，表示 4AI BD 板
SD8438	100，表示版本号 V1.00
D7900	BD 板状态
D7901	显示输入第 1 通道数值
D7902	显示输入第 2 通道数值
D7903	显示输入第 3 通道数值
D7904	显示输入第 4 通道数值

D7905	校准寄存器
SM8448	校准命令使能

2、2RTD BD 板的地址对应表

地址	说明
SD8428	H2202，表示 2RTD BD 板
SD8438	100，表示版本号 V1.00
D7900	BD 板状态
D7901	显示输入第 1 通道数值
D7902	显示输入第 2 通道数值
D7903	配置寄存器
D7904	校准寄存器
SM8438	配置命令使能
SM8448	校准命令使能

3、2AI1AO BD 板的地址对应表

地址	说明
SD8428	H2210，表示 2AI1AO BD 板
SD8438	100，表示版本号 V1.00
D7900	BD 板状态
D7901	显示输入第 1 通道数值
D7902	显示输入第 2 通道数值
D7903	显示输出第 1 通道数值

4、1AI1RTD BD 板的地址对应表

地址	说明
SD8428	H230A，表示 1AI1RTD BD 板
SD8438	100，表示版本号 V1.00
D7900	BD 板状态

D7901	显示输入第 1 通道数值
D7902	显示 RTD 输入第 1 通道数值
D7903	配置寄存器
D7904	配置命令使能

四、附录

4.1 特殊报错寄存器错误码内容

报错处理流程：

（1）先读取 SD8060 查找错误类型，根据错误类型可以判断错误级别，error 或者 warning，通信类错误(串行通信、模块总线灯)不会写入。

（2）根据错误类型查看对应错误寄存器里面的错误码，查看对应错误的错误步，只有运算错误才有错误步。

（3）根据错误寄存器里面的错误码可以显示错误描述或者诊断信息。

（4）具体错误问题见下表。

错误类型	错误类型编码	错误代码编号	错误描述	错误级别	诊断信息
PLC 硬件错误	8061	6101	RAM 错误	Error	可能是上电时外部 flash 读写失败，或损坏
		6105	看门狗定时超时错误		采样(运算时间)超出了 SD8000 的值。请确认程序。
		6116	栈错误		程序中 P 标签的存储出现问题
参数错误	8064	6402	内存容量的设定错误		检查系统参数区设定与当前 PLC 支持的参数范围是否一致
		6403	保持区域的设定错误		
		6404	注释区域的设定错误		
		6409	RUN 端子设定错误		
语法错误	8065	6501	指令-软元件符号-软元件的编号组合错误		出现此错误是指令的使用方法不正确，请

		6504	标签编号重复		检查指令的使用方法，也有可能是程序下载过程中突然断电导致程序不完整
		6505	软元件编号超出范围		
		6506	使用了未定义的指令		
		6508	中断输入（I）的定义错误		
		6509	程序错误		
		6510	MC 的嵌套编号的大小关系错误		
回路/流程错误	8066	6610	LD、LDI 连续使用次数超过 9 次		作为回路块整体的指令组合方法有不正确的地方时，或者成对的指令的关系不正确时，会发生这样的错误，检查指令的相互关系
		6617	应该从母线开始的指令没有连接在母线上		
		6618	只能在主程序中使用的指令在主程序以外		
		6619	FOR-NEXT 之间有不能使用的指令：STL、RET、MC、MCR、I、IRET		
		6620	FOR-NEXT 嵌套超出（最大 6 层）		
		6621	FOR NEXT 数量关系错误		
		6623	无 MC 指令		
		6624	无 MCR 指令		
		6625	STL 的连续使用次数超出 9 次		
		6626	STL-RET 之间有不能使用的指令：MC、MCR、I、SRET、IRET		
		6627	无 STL 指令		
		6629	无 P、I		
		6634	CALL 的嵌套超出 6 个		
		6637	分支错误		

运算错误	8067	6701	①没有 CJ、CALL 的跳转目标地址②变址修饰的结果越界或不正确	Warning	这些是在运算执行过程中产生的错误，请修改程序或是检查应用指令的操作数的内容。 即使没有发生语法、回路错误，例如，软元件变址超过其最大范围
		6705	应用指令中的操作数在规定之外的软元件		
		6706	应用指令的操作数的软元件编号范围或数据的值超出		
		6710	参数之间的不匹配		在移位指令等中，存在源操作数和目标操作数重复的情况
		6730	采样时间(Ts)为对象范围以外(Ts≦0)		在控制参数的设定值或 PID 运算中出现数据错误，请检查参数的内容。
		6732	输入滤波器常数(α)为对象范围以外(α<0 或 100≦α)		
		6733	比例增益(KP)为对象范围以外(KP<0)		
		6734	积分时间(TI)为对象范围以外(TI<0)		
		6735	微分增益(KD)为对象范围以外(KD<0 或 201≦KD)		
		6736	微分时间(TD)为对象范围以外(TD<0)		
		6742	测量值变化量超出(ΔPV<-32768 或是 32767<ΔPV)		各参数在最大值或最小值下继续运行
		6743	偏差超出(EV<-32768 或是 32767<EV)		
		6747	PID 运算结果超出(-32768～32767 以外)		
		6748	PID 输出上限设定值<输出下限设定值		请确认对象的设定内容是否正确

		6749	PID 输入变化量报警设定值、输出变化量报警设定值异常(设定值<0)		
		6764	输入或输入端子被其他功能占用		可能是其他计数器或脉冲输出指令占用了端子
		6770	文件寄存器指令写入错误		使用文件寄存器写入指令之前没有对其进行初始化导致，先使用相关指令初始化
		6774	对应的高速计数器未使能		
		6775	一个高速计数器最多可以使用 6 条比较指令		
		6776	每种表格比较指令只能使用一个		
串行通信错误	/	6201	串行通信奇偶校验、溢出、帧错误		
		6203	串行通信数据校验错误		
		6204	串行通信数据格式错误		
模块总线错误	/	N*1000+2(N 为模块地址)	模块 N 连接错误		可能是 PLC 无法为模块分配内存或者 PLC 不支持该模块
		N*1000+3(N 为模块地址)	模块 N 连接断线		可能是工作环境干扰太大，或者供电不稳定
		N*1000+4(N 为模块地址)	模块 N 组态错误		当前模块连接状态与离线配置内容不符合