



HM 系列运动控制器参考手册

黄石市科威自控有限公司

资料编号：20210901-V14

联系电话：0714-3802747

目 录

第一章 安全注意事项	1
第二章 运动控制器选型	3
2.1 主要功能.....	3
2.2 型号说明.....	4
第三章 运动控制器安装与尺寸	5
3.1 运动控制器安装.....	5
3.1.1 安装场所.....	5
3.1.2 环境条件.....	5
3.1.3 安装方向与间隔.....	5
3.2 运动控制器尺寸.....	6
第四章 运动控制器接口	7
4.1 电源配线.....	7
4.2 状态识别.....	8
4.3 开关量输入.....	8
4.4 开关量输出.....	9
第五章 PLC 功能	11
5.1 软元件介绍.....	11
5.1.1 输入输出继电器[X], [Y].....	12
5.1.2 辅助继电器[M].....	13
5.1.3 状态继电器[S].....	14
5.1.4 定时器[T].....	15
5.1.5 计数器[C].....	17
5.1.6 数据寄存器[D].....	22
5.1.7 扩展寄存器[R].....	24
5.1.8 变址寄存器[V], [Z].....	25
5.1.9 标记指针[P], [I].....	26
5.1.10 常数[K], [H].....	28
5.1.11 掉电保护.....	28
5.2 基本逻辑指令.....	29
5.2.1 [LD], [LDI], [OUT]指令.....	30
5.2.2 [AND], [ANI]指令.....	31
5.2.3 [OR], [ORI]指令.....	32
5.2.4 [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF]指令.....	33
5.2.5 [ORB]指令.....	34
5.2.6 [ANB]指令.....	35
5.2.7 [MPS], [MRD], [MPP]指令.....	36
5.2.8 [MC], [MCR]指令.....	39
5.2.9 [INV]指令.....	41
5.2.10 [ALT]指令.....	42
5.2.11 [PLS], [PLF]指令.....	43

5.2.12	[SET], [RST]指令	44
5.2.13	对应计数器软元件C的[OUT], [RST]指令	45
5.2.14	[NOP], [END]指令	46
5.2.15	编程注意事项	47
5.3	步进顺控指令	48
5.3.1	[STL], [RET]指令	48
5.3.2	分支与汇合	49
5.3.3	循环与跳转	50
5.3.4	步进指令应用示例	50
5.4	常规功能指令	52
5.4.1	条件跳转[CJ]	54
5.4.2	子程序调用[CALL]/子程序返回[SRET]	56
5.4.3	主程序结束[FEND]	57
5.4.4	循环范围开始[FOR]/循环范围结束[NEXT]	58
5.4.5	监视定时器[WDT]	60
5.4.6	开始比较[LD□]	61
5.4.7	并联比较[OR□]	62
5.4.8	串联比较[AND□]	63
5.4.9	数据比较[CMPI]	64
5.4.10	区域比较[ZCP]	65
5.4.11	传送[MOV]	66
5.4.12	成批传送[BMOV]	68
5.4.13	多点传送[FMOV]	70
5.4.14	高低字节交换[SWAP]	72
5.4.15	两个数据交换[XCH]	73
5.4.16	批次复位[ZRST]	74
5.4.17	反向传送[CML]	76
5.4.18	加法[ADD]	77
5.4.19	减法[SUB]	78
5.4.20	乘法[MUL]	79
5.4.21	除法[DIV]	80
5.4.22	加1[INC], 减1[DEC]	82
5.4.23	求平均值[MEAN]	83
5.4.24	逻辑与[WAND], 逻辑或[WOR], 逻辑异或[WXOR]	84
5.4.25	求负[NEG]	87
5.4.26	开方[SQR]	89
5.4.27	循环左移[ROL], 循环右移[ROR]	90
5.4.28	带进位循环左移[RCL], 带进位循环右移[RCR]	92
5.4.29	位左移[SFTL], 位右移[SFTR]	94
5.4.30	字左移[WSFL], 字右移[WSFR]	96
5.4.31	移位读出[SFRD]	99
5.4.32	移位写入[SFWR]	100
5.4.33	整型转浮点[FLT]	101
5.4.34	浮点转整型[INT]	102


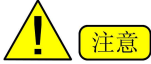
5.4.35	BCD 转二进制[BIN].....	103
5.4.36	二进制转 BCD[BCD].....	104
5.4.37	十六进制转 ASCII[ASCI].....	105
5.4.38	ASCII 转十六进制[HEX].....	106
5.4.39	浮点比较[ECMP].....	107
5.4.40	浮点区域比较[EZCP].....	108
5.4.41	浮点加法[EADD].....	109
5.4.42	浮点减法[ESUB].....	110
5.4.43	浮点乘法[EMUL].....	111
5.4.44	浮点除法[EDIV].....	112
5.4.45	浮点十进制转浮点二进制[EBIN].....	113
5.4.46	浮点二进制转浮点十进制[EBCD].....	114
5.4.47	浮点开方[ESQR].....	116
5.4.48	浮点 SIN 运算[SIN].....	117
5.4.49	浮点 COS 运算[COS].....	118
5.4.50	浮点 TAN 运算[TAN].....	119
5.4.51	浮点 ASIN 运算[ASIN].....	120
5.4.52	浮点 ACOS 运算[ACOS].....	121
5.4.53	浮点 ATAN 运算[ATAN].....	122
5.4.54	浮点 RAD 运算[RAD].....	123
5.4.55	浮点 DEG 运算[DEG].....	124
5.4.56	输入输出刷新[REF].....	125
5.4.57	PID 控制[PID].....	127
5.5	中断系统.....	130
5.5.1	中断资源介绍.....	130
5.5.2	中断执行流程.....	130
5.5.3	中断指令[EI], [IRET], [DI].....	132
5.5.4	中断嵌套.....	133
5.5.5	定时器中断.....	134
5.5.6	外部输入中断.....	135
5.5.7	系统软中断.....	136
5.5.8	高级中断.....	137
5.6	虚拟示波器.....	141
5.6.1	触发方式.....	141
5.6.2	触发单元.....	142
5.6.3	触发门限.....	142
5.6.4	采样周期.....	142
5.6.5	采样点数.....	142
5.6.6	记录起始寄存器序号.....	143
5.6.7	记录区间起始寄存器序号.....	144
5.6.8	记录区间终止寄存器序号.....	144
5.6.9	示波器状态设置/显示.....	145
5.6.10	当前采样点.....	145
5.6.11	通道工作模式.....	145



5.6.12 通道采样寄存器序号.....	146
5.7 特殊辅助继电器[M], 特殊数据寄存器[D].....	147
5.7.1 特殊辅助继电器[M]功能一览表.....	147
5.7.2 特殊辅助寄存器[D]功能一览表.....	154
第六章 通信功能.....	161
6.1 通信原理.....	161
6.1.1 通信方式.....	161
6.1.2 网络系统结构.....	162
6.2 串口初始化[UINIT].....	164
6.3 计算机链接协议.....	166
6.3.1 通信格式.....	166
6.3.2 任务调度配置.....	169
6.3.3 通信示例.....	171
6.4 MODBUS 协议.....	173
6.4.1 通信格式.....	173
6.4.2 软元件通信地址.....	174
6.4.3 MODBUS 指令.....	174
6.4.4 通信示例.....	184
6.5 自由协议.....	185
6.5.1 任务调度配置.....	185
6.5.2 自由协议指令.....	187
6.5.3 通信示例.....	189
6.6 其他协议.....	191
第七章 扩展模块.....	192
7.1 扩展寄存器.....	192
7.2 模拟量模块.....	193
7.2.1 模块规格.....	193
7.2.2 模块读写控制.....	195
7.2.3 输入模拟量.....	196
7.2.4 输出模拟量.....	196
7.3 开关量模块.....	198
7.3.1 模块规格.....	198
7.3.2 模块读写控制.....	200
第八章 运动控制.....	201
8.1 运动控制指令.....	201
8.2 运动轴对象.....	202
8.2.1 运动轴定义.....	202
8.2.2 运动输入轴参数配置.....	203
8.2.3 运动输出轴参数配置.....	204
8.3 单轴基本运动.....	218
8.3.1 回原点[HOME].....	218
8.3.2 点动[JOG].....	222
8.3.3 定位[PSOUT].....	225
8.3.4 制表[PTAB].....	227

8.3.5 表格输出[PTOUT].....	235
8.3.6 插补[IPL].....	238
8.3.7 定速定长输出[DPLSY].....	240
8.3.8 加减速定长输出[DPLSR].....	242
8.3.9 相对定位[DDRVI].....	244
8.3.10 绝对定位[DDRVA].....	246
8.3.11 中断定位[DDVIT].....	248
8.3.12 变速输出[DPLSV].....	250
8.3.13 高速读出[DHMOV].....	252
8.3.14 高速比较置位[DHSCS].....	253
8.3.15 高速比较复位[DHSCR].....	255
8.3.16 高速区间比较[DHSZ].....	257
8.3.17 高速表格比较输出[DHSCT].....	259
8.4 单轴关联运动.....	261
8.4.1 电子齿轮[GEAR].....	261
8.4.2 电子凸轮[CAM].....	264
8.4.3 追剪正程/滚切制表[PCTAB].....	284
8.4.4 追剪返程制表[PBTAB].....	289




第一章 安全注意事项

安全警告和标识

警告标识	含义
	该标识表示若操作错误则可能发生“死亡或重伤”
	该标识表示若操作错误则可能发生“人身伤害或财产损害”

安全标识	含义
	该图形表示“不可实施”的内容
	该图形表示“必须实施”的内容

安全注意事项

	
	切勿在有水的地方、存在腐蚀性、引火性气体的环境内以及可燃物旁使用，容易发生火灾
	不要在振动和冲击激烈的地方使用
	不要用湿手进行接线和设备操作
	切勿将手伸入运动控制器内部
	不要使导线受到损伤或承受过大的外力、重压、受夹
	切勿用裸手检查伺服接线故障，容易引发触电
	应在尘埃较少，不会接触到油、水的地方放置
	请进行正确的接线，否则可能发生火灾或故障
	导线应连接好，通电部位须通过绝缘套做到绝缘
	运动控制器必须实际接地
	外部应设置紧急停止电路，以确保紧急时可及时地停止运转、切断电源
	运动控制器的移动、接线、检查等要在切断电源并确保放电结束后没有触电危险的前提下进行
必须设置过电流保护装置、漏电断路器、温度防护装置以及紧急停止装置	

 注意	
	不要在运动控制器上放置重物
	不要使运动控制器受到较强的冲击
	不要频繁地开、关主电源
	不要对运动控制器进行长时间极限情况测试
	切勿自行改造、分解、修理
	严格按照正确的安装方法安装运动控制器
	确保运动控制器在规定的温度和湿度范围内运行
	发生故障时请清除故障、确保安全后再重新运行
	运动控制器发生故障时请切断电源
	长时间不使用运动控制器时必须切断电源
	运动控制器的保养工作应由 ([专业人员进行
	确保通风散热口的通畅、无障碍
	请在规定电压范围内运行

第二章 运动控制器选型

2.1 主要功能

● 掉电数据保存

掉电保存数据功能，运动控制器可以在断电时自动保存用户数据。

● 丰富的输入输出扩展单元

可接各种输入输出扩展单元。

输入类型：DC 输入（24V）

输出类型：继电器输出，晶体管输出

丰富的特殊扩展单元

模拟量输入单元

模拟量输出单元

热电偶输入单元

热电阻输入单元等

● 运动控制指令群

运动控制器支持简便指令和复杂运动控制指令：

HOME 回原点	JOG 点动	PSOUT 定位	PTAB 制表
PTOUT 表格输出	IPL 插补	DPLS 定速定长输出	DDRVI 相对定位
DDVIT 中断定位	DPLSV 变速输出	DHMOV 高速读出	DHSCS 高速比较置位
DHSCR 高速比较复位	DHSZ 高速区间比较	DHSCT 高速表格比较输出	GEAR 电子齿轮
CAM 电子凸轮	PCTAB 追剪正程/滚切制表	PBTAB 追剪返程制表	

● 高速处理

内装单相 2 点（单点 X0、X2）（MAX：400K），AB 相 2 点（MAX：200K）高速计数器，分辨率为 0.1MS。

脉冲捕捉功能

输入输出刷新功能

梯形图中断功能，6 个外部输入中断，3 个定时器中断，6 个高速计数器中断，8 个高级中断。

6 路 100KHz 高速脉冲输出，单路可达 600KHz。

● 通信

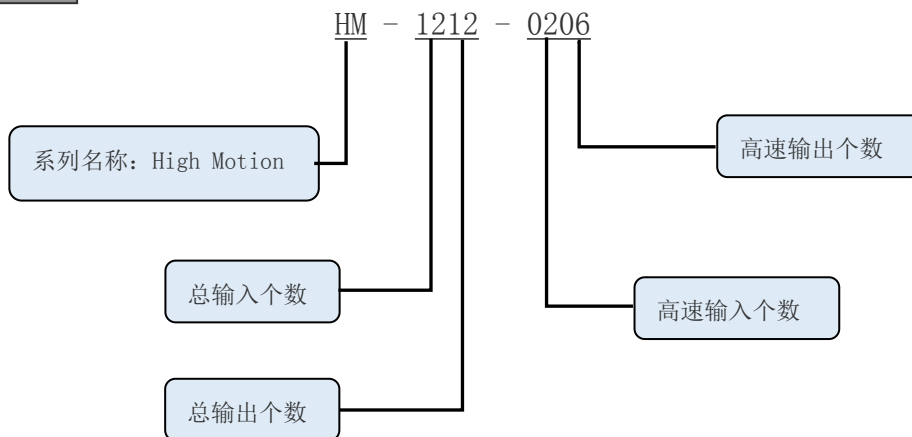
串口 0 和串口 1 通信速率可选择 4800, 9600, 19200, 38400, 57600, 115200，支持的通信协议包含：

编程口协议，MODBUS 协议，主从站链接协议以及自由协议等。

2.2 型号说明

运动控制器
型号

以下是 HM 系列运动控制器的型号命名规则。



第三章 运动控制器安装与尺寸

3.1 运动控制器安装

3.1.1 安装场所

安装注意事项	
不能安装在含有腐蚀性气体的场所	
不能安装在阳光直接照射的场所	
不能安装在温度急剧上下变化的场所	
不能安装在油，水，化合物容易侵入的场所	
不能安装在有大量灰尘的场所	
不能安装在震动大且会造成安装件移动的场所	

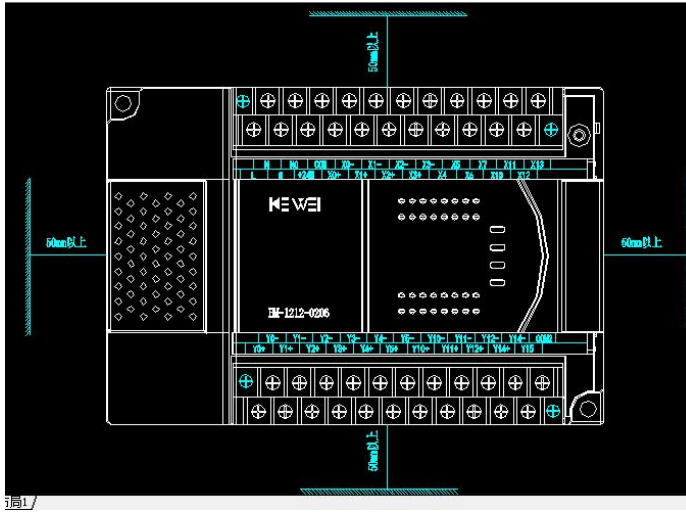
3.1.2 环境条件

项目	条件
使用环境温度	-10℃~55℃不冻结
使用环境湿度	10 ~ 99[RH%]不凝露
存储温度	-20℃~60℃(最高温度保证: 80℃ 72 小时)
存储湿度	10 ~ 99[RH%]不凝露
震动/冲击	4.9 (m/s ²) / 19.6 (m/s ²)
高度	海拔 1000m 以下正常使用, 海拔 1000m 以上降额使用

3.1.3 安装方向与间隔

 注意	
请留足够的空间，以便有效地降温	
为保证控制柜内的温度分布均匀，请安装风扇	
务必使运动控制器良好接地	

为防止机器内部温度过热，请采用正常的壁挂方式安装，要求上下留有 300mm 以上空间作为散热空间。



运动控制器基本单元和其他设备或结构物之间留 50mm 以上的空隙。尽量远离高压线、高压设备和动力设备。安装时有两种安装方法：

DIN 导轨安装：

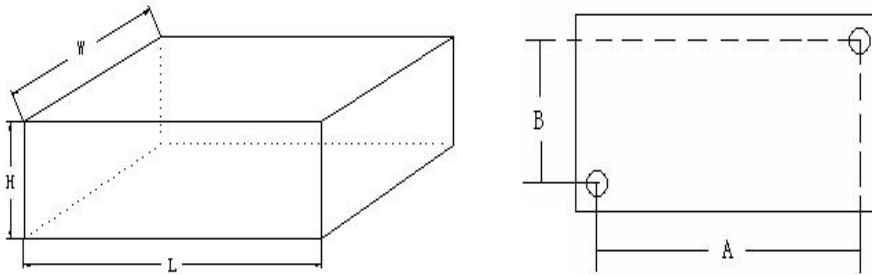
- 将 DIN35mm 槽水平固定于安装背板上
- 将模块底部下方的 DIN 槽卡扣拔出
- 把模块挂在 DIN 上，将卡扣压回原位，锁住模块
- 最后再将模块的两端用 DIN 槽卡档件固定，避免左右滑动

拆卸模块时，请将 DIN 导轨安装挂钩向下拉出。

直接安装时，用 M4 螺钉对准螺孔固定即可。但各个器件之间需要留 1-2mm 的间隙。

3.2 运动控制器尺寸

运动控制器安装尺寸如图所示：



产品型号	L (mm)	W (mm)	H (mm)	A (mm)	B (mm)
HM-1212-0206	140	95	85	130	85

A 表示孔心距。孔的直径是 4.5 mm。

存在较大冲击的场合，建议采用四个 M4 螺钉固定。

第四章 运动控制器接口

接口端子分布

HM-1212-0206 运动控制器的端子分布如下所示:

	N		COM0	X0-	X1-	X2-	X3-	X5	X7	X11	X13
L	G	+24V	X0+	X1+	X2+	X3+	X4	X6	X10	X12	
	Y0-	Y1-	Y2-	Y3-	Y4-	Y5-	Y10-	Y11-	Y12-	Y14	COM1
	Y0+	Y1+	Y2+	Y3+	Y4+	Y5+	Y10+	Y11+	Y12+	Y13	Y15

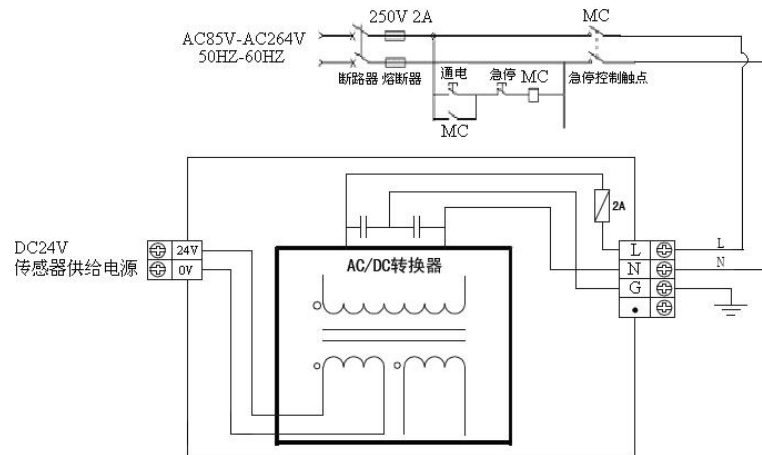
端子名称	说明
L、N	工作电源接入端子，电压范围为 85~264VAC，50/60Hz。
G	接地端子。
+24V、COM0	DC24V 电源输出，额定负载电流为 400mA。
Xn+、Xn-	X000~X003，差分输入，高速开关量输入端子或者 AB 相输入端子。
X4~X13、COM0	X004~X007、X010~X013，集电极输入，开关量输入端子，COM0 为公共端。
Yn+、Yn-	Y000~Y005，差分输出，高速脉冲输出口或普通开关量输出。
Yn+、Yn-	Y010~Y012，差分输出，高速脉冲输出的方向或普通开关量输出。
Yn、COM1	Y013~Y015，集电极输出，高速脉冲输出的方向或普通开关量输出。
RSA、RSB	485 通信端子。

4.1 电源配线

HM 系列运动控制器 220V 电源规格如下表所示:

额定电压	AC220V
电压允许范围	AC85V ~ 264V
额定频率	50/60Hz
允许瞬间断电时间	10ms
电源保险丝	250V 3.15A
冲击电流	最大 40A 5ms 以下/AC100V 最大 50A 5ms 以下/AC200V
最大消耗功率	30W
传感器用电源	24VDC±10% 最大 400mA

- 电源线请用 2mm² 以上的电线，以防止电压下降。
- 即使出现 10ms 以内断电，控制器仍可继续工作。当长时间地断电或异常电压下降时，控制器就停止工作，输出也呈 OFF 状态，当电源恢复供电时，控制器就自动开始运行。

220V 电源配
线

4.2 状态识别

HM 系列运动控制器状态指示 LED 有：POWER、RUN、ERROR。

运动控制器正常运行时，指示 LED POWER 和 RUN 常亮。

- 指示 LED POWER 熄，运动控制器电源不正常。
- 指示 LED RUN 熄，运动控制器处于设置状态。
- 指示 LED ERROR 亮，表示运动控制器运行时出错。在这种情况下，请先检查程序正确与否；若程序正确，再将运动控制器断电再复电，若运动控制器恢复正常，则请检查有无高强度干扰源及运动控制器接地是否确实。若 ERROR 仍亮，则运动控制器可能已经故障，应考虑送修。

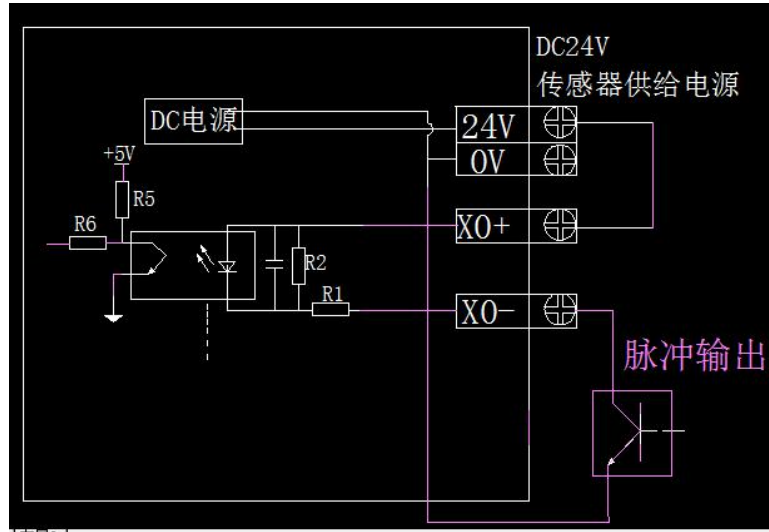
4.3 开关量输入

对于 HM-1212-0206 运动控制器而言，X000~X003 为差分输入电路，X004~X007、X010~X003 为集电极输入电路。HM 系列运动控制器的开关量输入规格表如下所示：

项目	规格	
	X000~X003	X004~X007, X010~X013
输入信号电压	DC24V ± 10%	
输入信号电流	8mA	4.5mA
输入信号阻抗	2kΩ	5.1 kΩ
输入 ON 电流	7mA 以上	2.15mA 以上
输入 OFF 电流	6.3mA 以下	1.2mA 以下
输入响应时间	0.1ms	
输入信号形式	无电压触点，或 NPN 集电极开路晶体管	
电路绝缘	光电耦合器隔离	
输入状态显示	输入 ON 时 LED 灯亮	

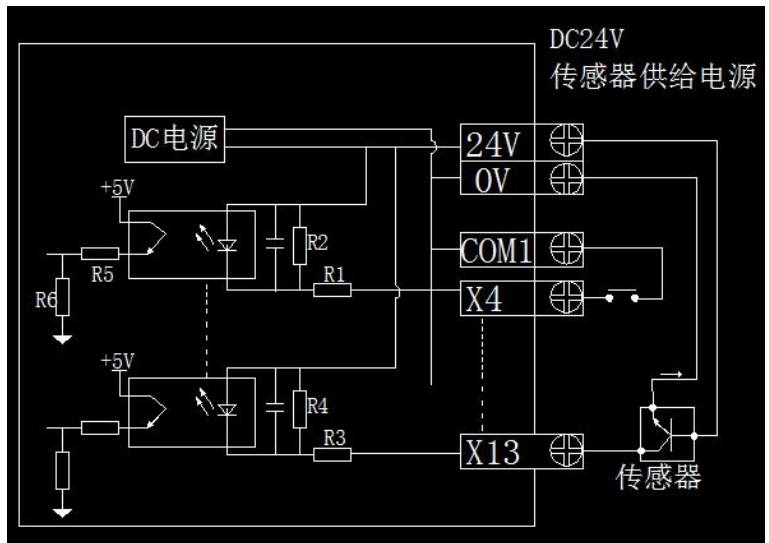
差分输入电路

HM 系列运动控制器的差分输入端子连接电路如下所示：



集电极输入电路

HM 系列运动控制器的集电极输入端子连接电路如下所示：



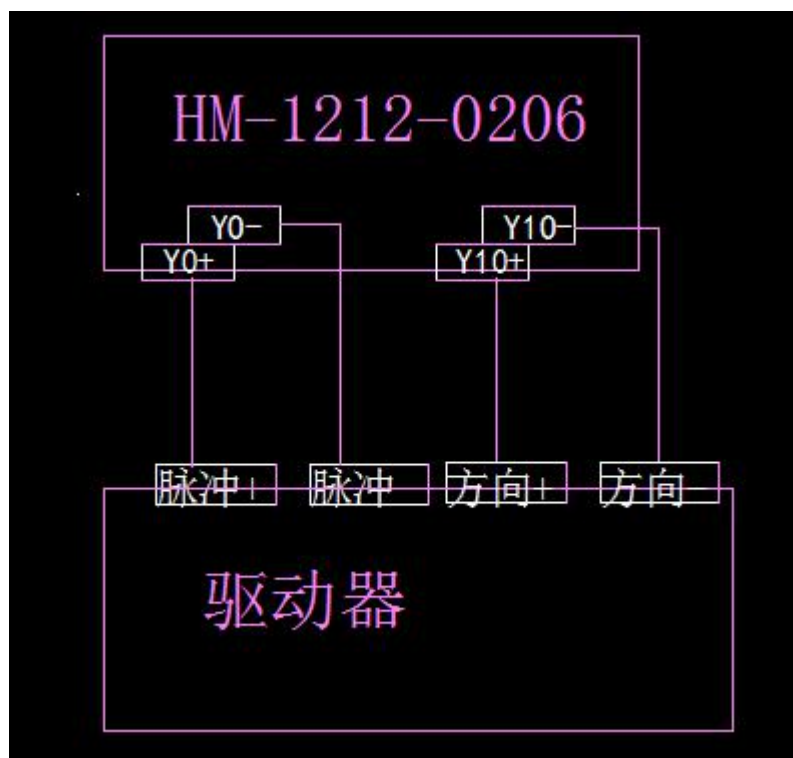
4.4 开关量输出

对于 HM-1212-0206 而言，所有输出为差分输出和晶体管输出。其中 Y0~Y5 为高速脉冲输出，Y10~Y15 为普通输出，作为脉冲的方向。当不作高速输出和方向功能使用时，输出端子可以作一般输出使用。

高速脉冲输出位	Y000~Y005
高速脉冲输出的方向位	Y010~Y015
动作指示	LED 指示灯
脉冲输出最大频率	6 路总共输出 600KHz

高速脉冲输出电路

HM 系列运动控制器的高速脉冲输出端子连接电路如下所示：



第五章 PLC 功能

5.1 软元件介绍

本章节就内置的输入输出继电器、辅助继电器、状态继电器、定时器、计数器、数据寄存器等各种软元件的功能和作用进行说明。

简称	代号及点数范围	说明
输入 X	X000~X377 (8 进制)	输入继电器, 单机点数由运动控制器具体型号而定
输出 Y	Y000~Y377 (8 进制)	输出继电器, 单机点数由运动控制器具体型号而定
辅助 M	M0~M3071	辅助继电器
	M8000~M8255	特殊辅助继电器, 具体定义见第 5.7 章
状态 S	S0~S1023	状态继电器
时间 T	T0~T255	T0~T199, 100ms 时基定时器
		T200~T245, 10ms 时基定时器
		T246~T249, 1ms 时基定时器
		T250~T255, 100ms 时基定时器
计数 C	C0~C255	C0~C199, 16 位增计数器
		C200~C234, 32 位可逆计数器, 计数方向由 M8200~M8234 控制
		C235~C255, 32 位高速计数器 (与高速计数指令配合使用)
寄存器 D	D0~D7999	D0~D79, 内部保留区
		D80~D97, 输出轴脉冲显示区
		D98~D199, 内部保留区
		D200~D299, 软件虚拟输入轴参数区
		D300~D599, 软件虚拟输出轴参数区
		D600~D999, 内部保留区
		D1000~D7999, 用户使用区
D8000~D8255	系统寄存器区, 具体定义见第 5.7 章	
扩展寄存器 R	R0~R99	扩展寄存器区
变址 V, Z	V0~V7, Z0~Z7	与其他软元件配合使用, 如 V0=10, D0V0 的地址为 D10
常数 K, H	带符号整型数范围	K 为 10 进制表达数, H 为 16 进制表达数, 如 H10=K16
程序指针 P	P0~P127	做程序跳转或子程序调用使用
中断指针 I	I000~I999	适用于各类型中断的指示标签
程序容量	2K, 4K, 8K, 12K	运动控制器所适用的梯形图步长控制

* 运动控制器在编写梯形图时, 请严格按照说明书规则进行编程, 否则可能造成 PLC 梯形图运行故障。当运动控制器出现运行故障时, 按以下顺序进行 PLC 的复位。

- 断电后重新上电
- 在上电后的 1~3 秒的时间内, 点击编程视窗的“PLC 停止”按钮
- 出现 PLC 已停止运行的对话框后, 下载正确的梯形图以恢复运动控制器的正常运行

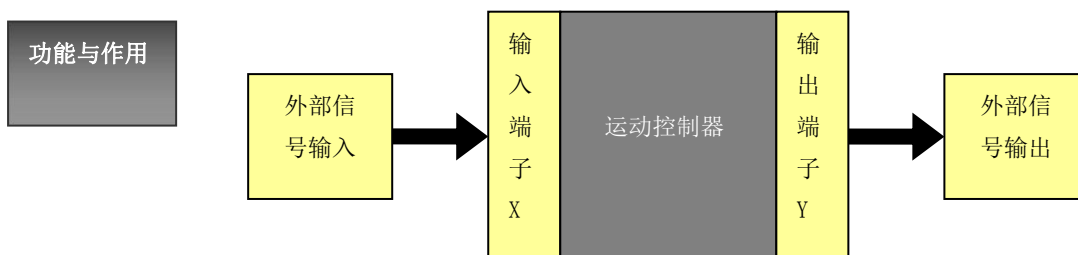
5.1.1 输入输出继电器[X], [Y]

输入输出继电器的编号是与实际输入输出端口对应的映射标识。这些地址号使用 8 进制表达，因此不存在如 8、9 这样的数值。

- 输入继电器 X 作用是接受外部开关量信息，将接受到的外部信息送到 CPU，经过用户程序的处理由输出继电器 Y 控制外部设备，完成控制任务。因此在 PLC 的完整程序中，X、Y 是相互依存的。
- X/Y 的基本功能是作位元件使用，但其组合也可以作为字元件来处理数值（如 K2X0）。
- 输入输出继电器根据接口电路的不同，可分为普通型（5KHz）和高速型（平均 100KHz）。普通型无法做高速型使用，但高速型可做普通型使用。

运动控制器的 X/Y 输入输出继电器使用序号及功能定义如下：

端子类型	端子序号	端子作用
输入端子 X		
高速计数	X000, X002	高速计数端口，可绑定高速计数器 C 进行高速计数。计数分辨率为 0.1MS。
高速计数辅助	X001, X003	高速计数辅助端口，可作为高速计数器 C 的计数方向端子或是 B 相端子。
用户输入	X004~X007	用户输入端口。
高速计数辅助	X010~X013	高速计数辅助端口，可作为高速计数器 C 的启动或复位端子。
用户虚拟输入	X014~X017	虚拟输入端口，用户可通过 M8160~M8163 改变虚拟端口的当前状态。
输出端子 Y		
高速输出	Y000~Y005	高速输出端口，通常绑定运动指令轴进行插补输出。单路可达 600K，6 路一起使用 100K。
用户输出	Y010~Y015	用户输出端口。

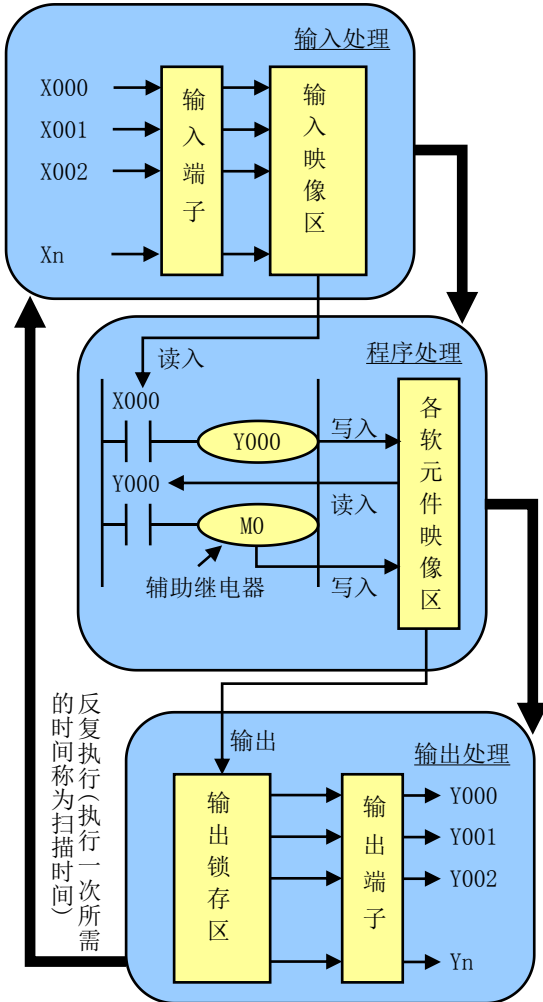


输入端子是 PLC 从外部开关接收信号的窗口。在 PLC 内部，与 PLC 的输入端子相连的输入继电器（X）是一种光绝缘的电子继电器，它有无数的常开触点与常闭触点。这些常开/常闭触点可在 PLC 内随意使用。

输出端子是 PLC 向外部负载发送信号的窗口。

动作与时序

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



● 输入处理

PLC 在执行程序之前，将所有输入端子的 ON/OFF 状态，读入输出映像区。

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

此外，即使输入触点出现 ON→OFF、OFF→ON 的变化时，在判定 ON/OFF 之前，输入滤波器会造成短时间的响应滞后（毫秒级）。

● 程序处理

PLC 根据程序存储器的指令内容，从输入映像区或其他软元件的映像区中读出各软元件的 ON/OFF 的状态，从 0 步开始依次进行运算，然后将结果写入映像区。因此，各软元件的映像区存储区随着程序的执行逐步改变其内容。而且，输出继电器的内部触点根据输出映像存储区的内容执行动作。

● 输出处理

一旦所有指令执行结束，将输出 Y 的映像存储区的 ON/OFF 状态传输至输出锁存区，这为了 PLC 的实际输出。PLC 内的外部输出用触点，按照输出用软元件的响应滞后后实践动作。

5.1.2 辅助继电器[M]

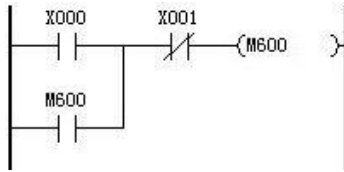
辅助继电器 M 同输出继电器 Y 一样，可以被 PLC 内各种软元件的触点驱动。但 M 不能直接与外部端子相连，仅起辅助逻辑运算的作用。它有无数的常开常闭触点，可被随意使用。

M 在使用时既可作位元件也可作字元件（必须拼成字的形式，如 K2M0）。

- M0~M3071 为用户辅助继电器使用区，可做触点判断及逻辑运算，无使用限制。
- M8000~M8255 为特殊辅助继电器使用区，此区域继电器的状态往往是被系统赋予了特殊的含义，比如 M8002 为上电第一个扫描周期被接通。特殊辅助继电器在一般情况下，只做触点能流导通判断使用，应尽量避免修改此区域辅助继电器的状态，以免影响系统正常运行。

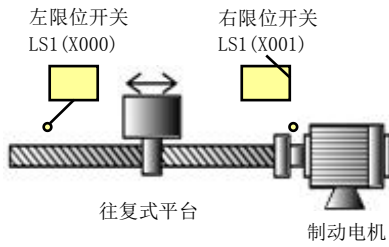
功能与动作示例

PLC 内有许多辅助继电器。这类辅助继电器的线圈与输出继电器一样，由 PLC 内的各种软元件的触点驱动。辅助继电器通常有无数的电子常开触点与常闭触点，在 PLC 内可随意使用。但是，该触点不能直接驱动外部负载，外部负载的驱动要通过输出继电器进行。

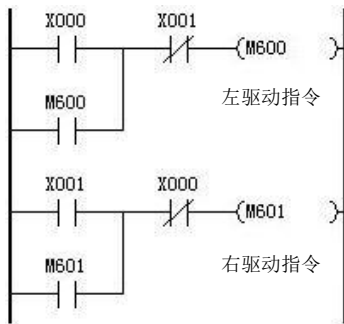


辅助继电器自锁示例

- 左图表示了 M600 动作的自锁示例。在该电路中，如果 X000 接通，M600 动作的话，即使 X000 断开，由于并联触点导通，M600 也会自己保持动作。但是，如果 X001 的常闭触点开路，则 M600 不会动作。



- 下述梯形图的功能：
可通过 M600/M601 的当前状态，控制往复式平台的移动方向。



- X000 = ON (左限位开关) → M600 = ON → 右驱动 →
X001 = ON (右限位开关) → M600 = OFF, M601=ON → 左驱动。

* 特殊辅助继电器含义定义见第 5.7 章

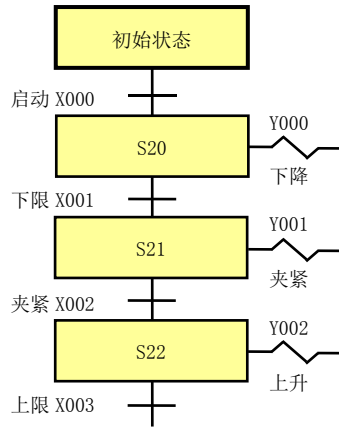
5.1.3 状态继电器[S]

状态继电器 S 的主要作用是在步进顺控(STL)指令中表示工艺状态，以流程的方式，可以使程序变得结构清晰易懂，并且容易修改。

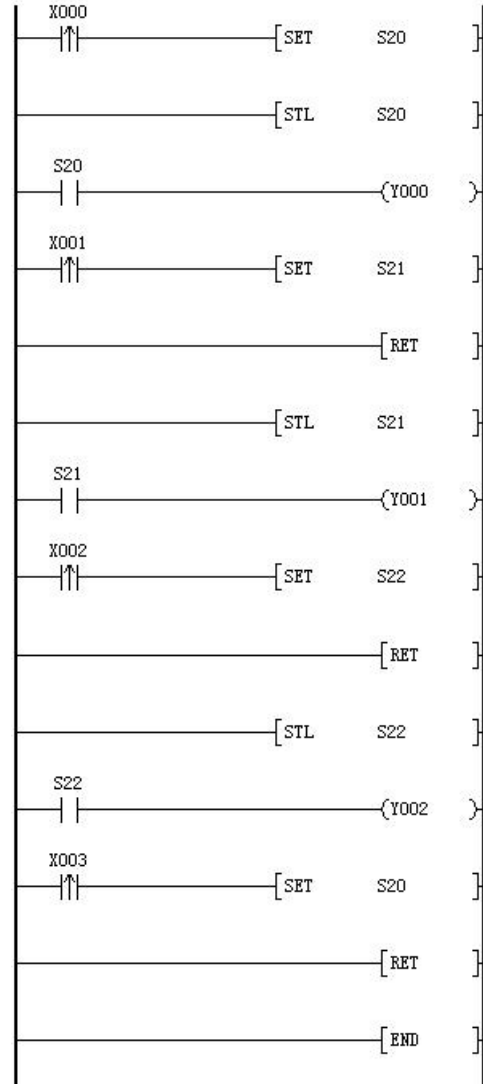
状态继电器 S 与辅助继电器 M 一样，有无数的常开/常闭触点，在顺序程序内可随意使用。在不用于步进梯形图指令时，状态继电器 S 可在触点判断和逻辑运算中使用。

状态继电器 S 的编号为 S0~S1023，一共有 1024 个点位可供使用。

功能与动作
示例



- 状态 S 是对工序步进控制简易编程的重要软元件，经常与步进梯形图指令 STL 结合使用。
- 如上图所示的工序步进控制中，如果启动信号 X000 为 ON，则状态 S20 置位（ON），下降用的电磁阀 Y000 开始动作。其结果是，若下限限位开关 X001 为 ON，则状态 S22 置位（ON），夹紧用的电磁阀 Y001 动作。
- 如果夹紧动作确认的限位开关 X002 为 ON，则状态 S22 置位（ON）。
- 随着状态动作的转移，状态自动返回原状态。
- 右图为示例完整梯形图。



* 关于步进顺控指令 STL 的使用方法，参考对应指令描述说明。

5.1.4 定时器[T]

定时器 T（亦称时间继电器）的主要作用是定时控制。

定时器 T 也可以作为数据寄存器用，通过赋值可以指定定时器当前的值。定时器的值可以用常数 K 直接指定，也可以用数据寄存器间接指定。

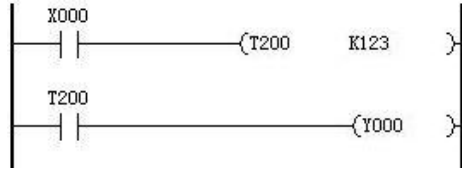
使用 OUT 指令开启定时器，当定时器被指定的值达到时，其触点将闭合，但它的当前值继续变化（增加），满最大值 K32767 时，才停止变化，保持 K32767 的状态。

如果是累计定时器，用 RST 指令复位。

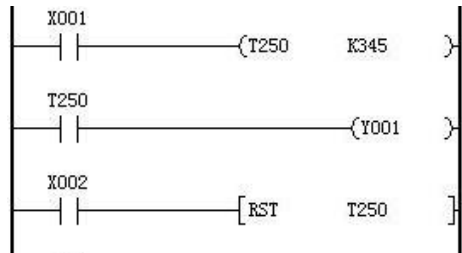
定时器 T 编号	定时时基	累计
T0~T199	100 ms	不累计
T200~T245	10 ms	不累计
T246~T249	1 ms	累计
T250~T255	100 ms	累计

功能与示例

定时器累计 PLC 内的 1ms, 10ms, 100ms 等的时钟脉冲, 当达到所定的设定值时输出触点动作。采用程序存储器内的常数 (K) 作为设定值, 也用数据寄存器 (D) 的内容进行间接指定。

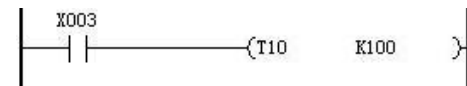


- 如果定时器线圈 T200 的驱动输入 X000 为 ON, T200 用当前值计数累计 10ms 的时钟脉冲。如果该值等于 K123 时, 定时器的输出触点动作。也就是说输出触点在线圈驱动上 1.23 秒后动作。驱动输入 X000 断开, 定时器复位, 输出触点复位。

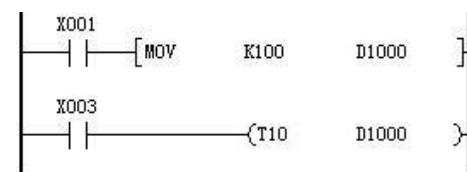


- 如果定时器 T250 的驱动输入 X001 为 ON, 则 T250 用当前值计数器将累积 100ms 的时钟脉冲。如果该值达到设定值 K345 时, 定时器的输出触点动作。在计算过程中, 即使输入 X001 断开, 再次启动后将继续计算, 其累积计算动作时间为 34.5 秒。如果复位输入 X002 为 ON 时, 定时器复位, 输出触点也复位。

设定值的指定方法



- T10 是以 100ms (0.1s) 为单位的定时器。将 100 指定为常数, 则 0.1s x 100 = 10s 的定时器工作。

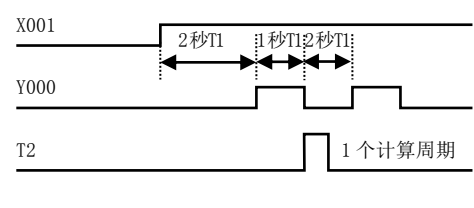
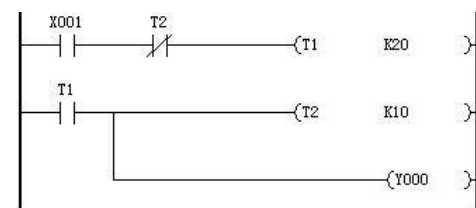
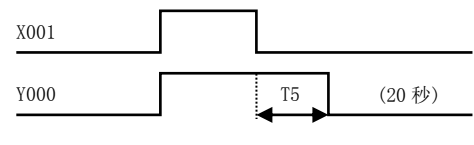
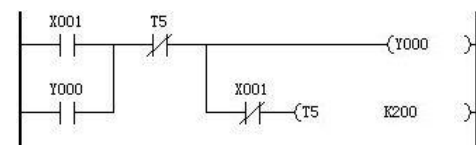


- 将间接指定数据寄存器的内容预先写入程序或通过数值开关输入。

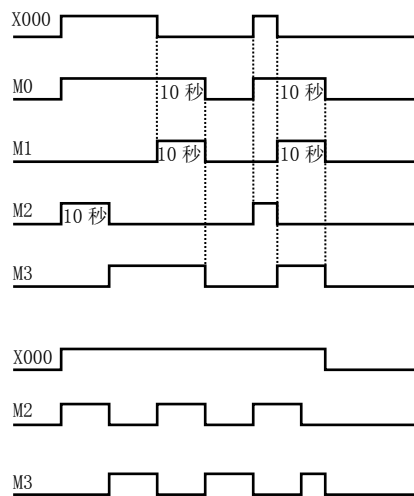
定时器动作与精度

- 除了中断程序和子程序执行的定时器外, 在线圈驱动后, 定时器开始计时。在计时完成后的最初的线圈指令执行时, 输出点动作。
- 定时器精度偏差不会超过 1ms, 但执行动作的时间将受到扫描周期的影响。

应用与示例



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



- 将以 m 指令的值作为以 s 指定的定时器的设定值。此例中为 10 秒。
- M0 为输出延时关断定时器。
- M1 为输入 ON→OFF 后的单脉冲输出定时器。
- M2、M3 为闪烁定时器，其连接方法如图所示。
- 将 M3 如图所示的方法连接后，则 M2、M1 将闪烁输出。
- 如果 X000 断开，则设定时间后 M0、M1 和 M3 断开，T10 也被复位。
- 在这里所使用的定时器不得在其他一般回路中重复使用。

5.1.5 计数器[C]

计数器 C（以下称 C）的主要作用是接收脉冲信号并计数，到达设定值时其触点动作，起控制作用。此外，可作定时器和数据寄存器使用。

根据编号的不同，C 分为 16 位和 32 位两类，它们都由 RST 指令复位。

- C0~C199，16 位增计数器。
- C200~C234，32 位可逆计数器。
- C235~C255，32 位高速计数器。

16 位 C 是加法计数器，计数设定值可用常数项直接设定，亦可用数据寄存器间接设定，计数范围为 1~32767。32 位 C 是环形可逆计数器，可进行增、减计数。计数值到达临界值的变化特点：K-2147483648 减 1 变为 K2147483647；K2147483647 加 1 变为 K-2147483648。

32 位 C 的设定值与 16 位类似，既可用 K 直接设定，又可用数据寄存器间接设定，但必须占用编号连续的两个数据寄存器，例如（D1001、D1000）。

32 位 C 只有一个输入端，要实现增、减计数，可通过以下 M 继电器的 ON/OFF 或对应的 X 方向端子来确定，其与 C200~C255 是一一对应的关系。即 M8200-OFF，C200 增计数；M8200-ON，C200 减计数，其余类推。

计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换	计数器 NO.	方向切换
C200	M8200	C201	M8201	C202	M8202	C203	M8203
C204	M8204	C205	M8205	C206	M8206	C207	M8207
C208	M8208	C209	M8209	C210	M8210	C211	M8211
C212	M8212	C213	M8213	C214	M8214	C215	M8215
C216	M8216	C217	M8217	C218	M8218	C219	M8219
C220	M8220	C221	M8221	C222	M8222	C223	M8223
C224	M8224	C225	M8225	C226	M8226	C227	M8227
C228	M8228	C229	M8229	C230	M8230	C231	M8231
C232	M8232	C233	M8233	C234	M8234	C235	M8235
C236	-	C237	M8237	C238	-	C239	M8239
C240	M8240	C241	M8241	C242	M8242	C243	-
C244	-	C245	-	C246	-	C247	-
C248	-	C249	-	C250	-	C251	-
C252	-	C253	-	C254	-	C255	-

计数器特点

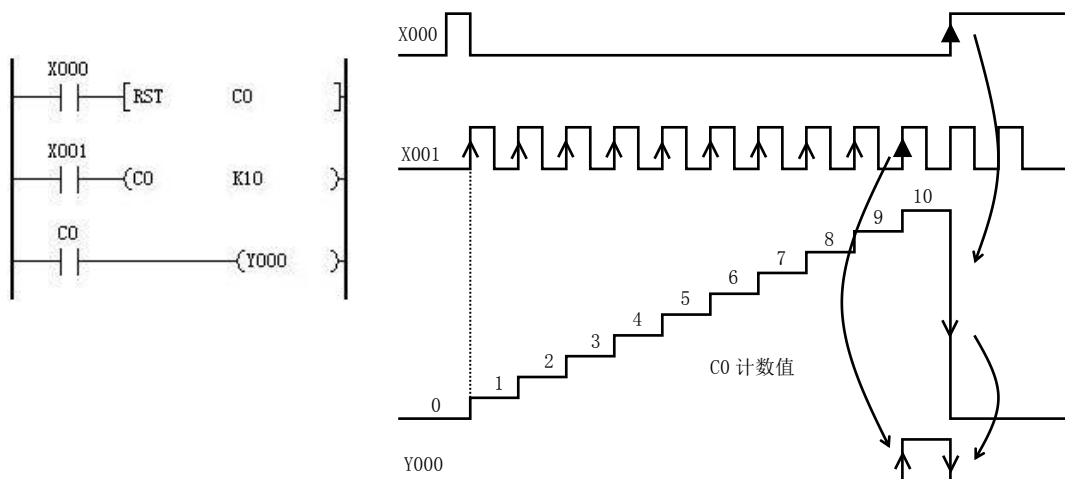
16 位计数器与 32 位计数器的特点如下表所示。可按计数方向的切换与计数范围的使用条件来分开使用。

项目	16 位计数器	32 位计数器
计数方向	顺数	顺/倒可切换使用
设定值	1~32767	-2147483648~+2147483647
指定的设定值	常数 K 或数据寄存器	同左但是将占用连续 2 个数据寄存器
当前值的变化	顺数后不变化	根据条件可变
输出接点	顺数后保持动作	根据条件可变
复位动作	执行 RST 命令时，计数器的当前值位零，输出线圈/触点 OFF。	
当前值寄存器	16 位	32 位

功能与动作示例

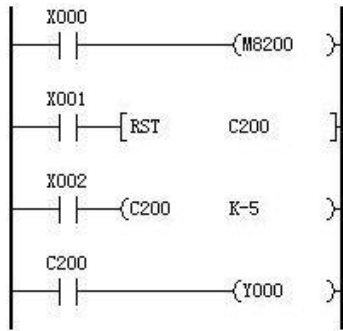
计数器状态的分配，可通过在外围设备改变参数设定进行变更。

16 位 2 进制增计数器，其有效设定值为 K1~K32767（10 进制常数）。设定值 K0 和 K1 具有相同的含义，即在第一次计数开始时输出触点就动作。

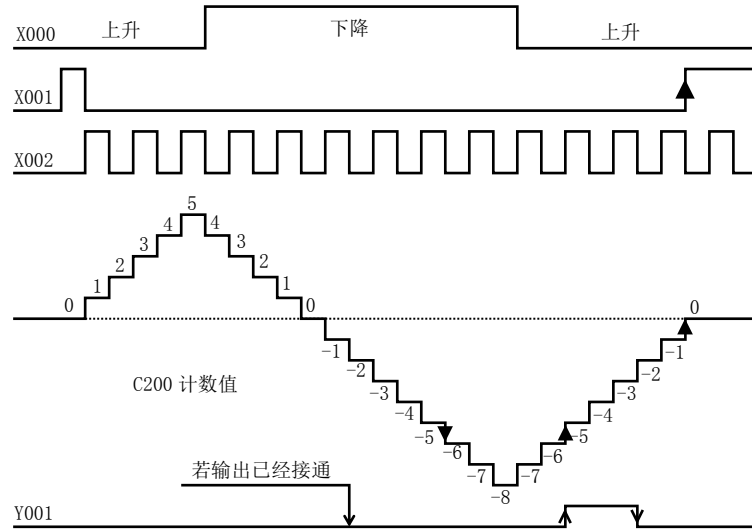


- 计数输入 X001 每驱动 C0 线圈一次，计数器的当前值就增加，在执行第十次的线圈指令时，输出触点动作。以后即使计数输入 X001 再动作，计数器的当前值不变。
- 如果复位输入 X000 为 ON，则执行 RST 指令，计数器的当前值为 0，输出触点复位。
- 计数器的设定值，除用上述常数 K 设定外，还可用数据寄存器编号指定。例如，指定 D1000，如果 D1000 的内容为 123 是一样的。
- 在以 MOV 等指令将设定值以上的数据写入当前值寄存器时，则在下次输入时，输出线圈接通，当前值寄存器变为设定值。

32 位的 2 进制增计数/减计数的设定值有效范围为 -2147483648~2147483647（十进制常数）。利用特殊的辅助继电器 M 指定增计数/减计数的方向。



- 如果对 C2△△驱动 M82△△，则为减计数；不驱动时，则为增计数。
- 根据常数 K 或数据寄存器 D 的内容，设定值可正可负。将连号的数据寄存器的内容视为一对，作为 32 位的数据处理。因此，在指定 D1000 时，D1001 和 D1000 两项作为 32 位设定值处理。

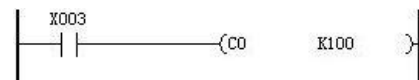


利用计数输入 X002 驱动 C200 线圈时，可增计数或减计数。

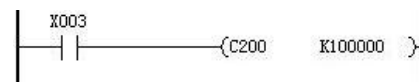
在计数器的当前值由-6→-5 增加时，输出触点置位。在由-5→-6 减少时，输出触点复位。

- 当前值的增减与输出点的动作无关，但是如果从 2147483647 开始增计数，则成为-2147483648。同样，如果从-2147483648 开始减计数，则成为 2147483647。（这类动作被称为环形计数）
- 如果复位输入 X001 为 ON，则执行 RST 指令，计数器的当前值变为 0，输出触点也复位。
- 32 位计数器也可作为 32 位数据寄存器使用。但是，32 位计数器不能作为 16 位应用指令中的软元件。
- 在以 DMOV 指令等把设定值以上的数据写入当前值数据寄存器时，则在以后的计数输入时可继续计数，触点也不变化。

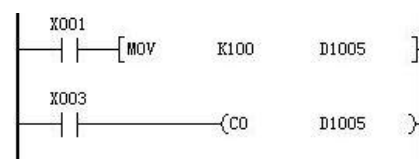
设置值的指定方法



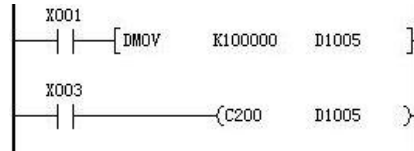
- 常数（10 进制常数）
1~32767 100 次计数。



- 常数（10 进制常数）
-2147483648~2147483647 次计数。



- 间接指定的数据寄存器的内容预先在程序上写入或通过数字开关输入。



- 间接指定用的数据寄存器 2 个为一组 (D1005, D1006)。在使用 32 位命令写入设定值的同时, 请不要在其他的程序上重复该数据寄存器。

计数器的响应速度

计数器对 PLC 的内部信号 X, Y, M, S, C 等触点的动作进行循环扫描并计数。例如, 在将 X001 作为计数输入时, 其接通和断开的持续时间, 必须比 PLC 的扫描时间长(通常在数值 10Hz 以下)。对于这个问题将涉及到后面即将提及的高速计数器通过对特定的输入作中断处理来进行计数, 执行数 KHz 的计数, 而与扫描时间无关。

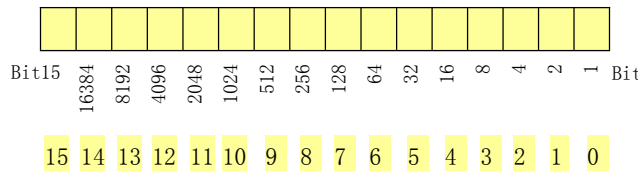
计数器结构与处理

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

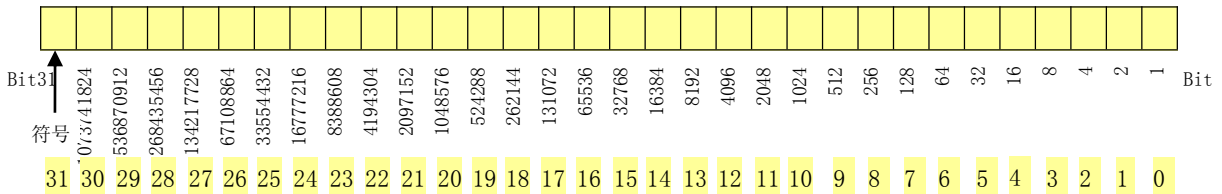
计数器的当前值寄存器的结构如下所示, 对应用指令中的操作数指定计数器编号, 则与数据寄存器一样, 作为 16 位或 32 位数据存储软件元件处理。

- 处理数值的范围 (10 进制)

16 位: 0~32767

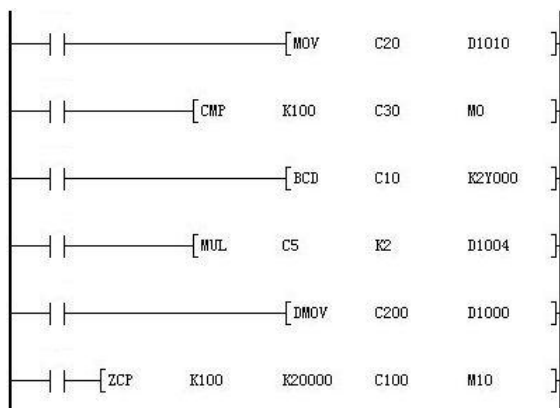


32 位: -2147483648~+2147483647



普通计数器的使用

计数器 C 可以与许多功能指令配合进行运用, 以达到灵活使用的目的。



- C20(当前值)→D1010 传送
- 将 10 进制整数 100 和 C30(当前值)比较, 结果输出到 M0~M2。
- 将 C10(当前值)作 BCD 变换, 输出到 Y000~Y007。
- 将 C5(当前值)翻倍, 向(D1005, D1004)传送。
- 将 C200(当前值)→(D1001, D1000)传送。
- 将 C200(当前值)与 10 进制整数 100~20000 的区间作比较, 结果输出到 M10~M12。

高速计数器的使用

C235~C255 作为 32 位高速计数器,可以与高速计数指令(如 DHSCS)一起配合使用。
对应的 C 计数器序号与对应的 X 输入端子的联动关系如下所示:

	C235	C236	C237	C238	C239	C240	C241	C242	C243
X000	U/D				U/D		U/D		
X001									
X002			U/D			U/D		U/D	
X003									
X010					R		R		
X011						R		R	
X012							S		
X013								S	

	C244	C245	C246	C247	C248	C249
X000	U/D		U/D		U/D	
X001	DIR		DIR		DIR	
X002		U/D		U/D		U/D
X003		DIR		DIR		DIR
X010			R		R	
X011				R		R
X012					S	
X013						S

	C250	C251	C252	C253	C254	C255
X000	A		A		A	
X001	B		B		B	
X002		A		A		A
X003		B		B		B
X010			R		R	
X011				R		R
X012					S	
X013						S

- U/D:计数器增计数/减计数接入端口,可通过对应序号的 M82□□进行计数方向的设置
- DIR:硬件方向口
- R:计数器复位接入端口
- S:计数器启动接入端口
- A:AB 相计数的 A 相接入端口
- B:AB 相计数的 B 相接入端口

5.1.6 数据寄存器[D]

数据寄存器 D 的作用是存储数据，在数据运算、网络通讯等方面尤为重要。其可做为 16 位或 32 位数据寄存器使用（最高位为正、负号位），当作为 32 位寄存器使用时，将连续占用两个相邻地址的 D 寄存器序号。数据寄存器 D 的表现形式多种，可表示整数、二进制浮点数及 BCD 码，作为二进制浮点数使用时将指定为 32 位寄存器使用。

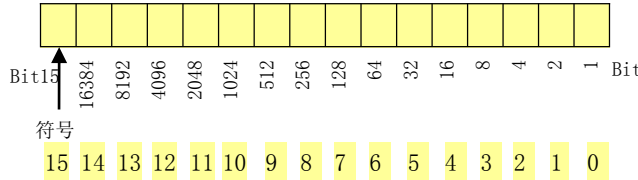
- D0~D999 为系统参数区，一般不允许做中间变量和工艺变量使用。其空间分配如下：
 - ◆ D0~D79 为系统保留区。
 - ◆ D80~D97 为输出轴脉冲显示区域，具体参数定义请参阅第 8.2.3 章内容。
 - ◆ D98~D199 为系统保留区。
 - ◆ D200~D299 为软件虚拟输入轴参数区域，具体参数定义请参阅第 8.2.2 章内容。
 - ◆ D300~D599 为软件虚拟输出轴参数区域，具体参数定义请参阅第 8.2.3 章内容。
 - ◆ D600~D999 为系统保留区。
- D1000~D7999 为用户使用区，无使用限制。
- D8000~D8255 为系统寄存器使用区，此区域寄存器数值被系统赋予了特殊的含义，比如 D8255 为内核版本编号。系统寄存器在一般情况下应尽量避免修改，以免影响系统正常运行。

寄存器结构与处理

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

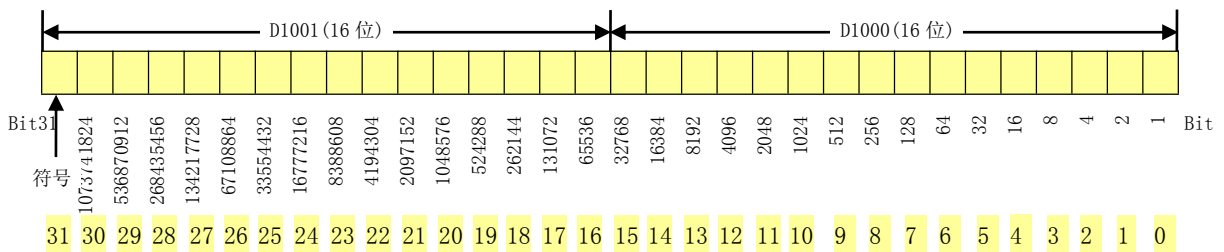
- 处理数值的范围（10 进制）

16 位：-32768~32767

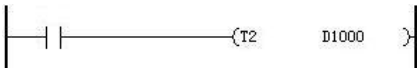

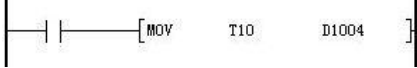
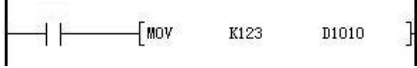
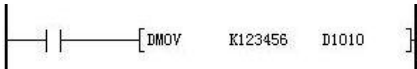

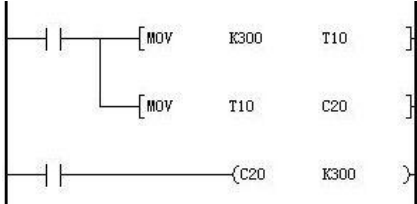


32 位：-2147483648~2147483647

以两个相邻数据寄存器表现 32 位的数据。对于 D1000 作为 32 位寄存器的结构如下(占用 D1001 寄存器)：



数据寄存器 D 在指定 32 位时，如果指定了低位（例如：D1000），则高位为继其之后的编号（例如：D1001）被自动占有。低位可用奇数或偶数的任意一种软元件编号指定。考虑到外围设备的监视功能，建议低位采用偶数软元件编号。

寄存器动作		数据寄存器 D 可以处理各种数值数据，通过利用它进行各种控制。	
			● 作为定时器与计数器的设定值被指定。 计数器与定时器将指定的数据寄存器的内容分别作为设定值执行动作。
			● 改变计数器的当前值。 计数器 C12 的当前值改变为 D1005 的内容。
			● 将定时器与定时器当前值读到数据寄存器。 将计数器 T10 的当前值传送到 D1004 中。
			● 在数据寄存器中存储数据。 向 D1010 传送 123(10 进制整数)。
			● 向(D1011, D1010)传送 123456(10 进制数)。 超过 32767 的数值是 32 位的数值,因此采用 32 位的 DMOV 指令。如果数据寄存器指定为低位 D1010, 则高位 D1011 自动被占用。
			● 向其他的数据寄存器传送数据寄存器的内容。 向 D1020 传送 D1010 的内容。
			● 在程序中没有使用的定时器与计数器可作为 16 位或 32 位的数据存储软元件(计数寄存器)使用。 向 T10 传送 300(10 进制数)。 向 C20 的当前值寄存器传送 T10 的内容。此时 T10 不起定时器的作用, 而作为数据寄存器工作。 在作为 32 位使用时, 与数据寄存器一样, 采用 2 个 16 位软元件(例: C1, C0)可表示 32 位数值。此外, 1 个 32 位计数器(例: C200 等)则可处理 32 位数值。

* 特殊辅助寄存器含义定义见第 5.7.2 章

5.1.7 扩展寄存器[R]

对于复杂系统工艺控制要求，其编程可能需要更多的用户寄存器作为中间单元或缓存区使用。除了传统的数据寄存器 D 以外，提供扩展寄存器 R 软件供用户使用，使之能更方便的构建梯形图程序。

扩展寄存器的使用方法与数据寄存器基本一样，可进行数值数据的读入、写出的数据寄存器。被当作为 32 位寄存器使用时，也将连续占用两个相邻地址的 R 寄存器序号。

- 扩展寄存器 R 编号为：R0~R99。

使用注意 事项

在使用扩展寄存器 R 时，还应注意以下事项。

- R 寄存器无固定初始值，在分配和使用之前应做好其数值的初始化，以免错误的初值导致梯形图工作异常。
- R 寄存器为特有扩展寄存器，在与第三方设备进行连接时(如 HMI)，不能使用编程口协议进行连接，推荐使用 MODBUS 协议进行 R 寄存器的通信。

5.1.8 变址寄存器[V], [Z]

变址寄存器 V 与 Z 两者功能完全一样。

同普通的数据寄存器一样，可进行数值数据的读入、写出的数据寄存器。

在应用指令的操作数中，变址寄存器还可以同其他的软元件编号或数值组合使用，效果为在程序中改变元件编号或数值内容，是一个特殊寄存器。

注意 LD, AND, OUT 等 PLC 的基本顺控指令或步进梯形图指令的软元件编号不能同变址寄存器组合使用。

变址寄存器 V 和 Z 均可做 16 位或 32 位寄存器使用，根据其调用指令的位数决定它的数据类型。当作为 32 位寄存器使用时，将连续占用 2 个相邻地址的变址寄存器。例如 V2(L)，占用 V2/V3 两个 16 位变址寄存器。

- 变址寄存器 V 编号为：V0~V7。
- 变址寄存器 Z 编号为：Z0~Z7。

软元件的变址

可能变址的软元件，其变址内容为：

- 10 进制数的软元件、数值：M, S, T, C, D, KnM, KnS, P, K
例如，V0=K5，执行 D1020V0 时，被执行的软元件的编号为 D1025 (D1020+5)。此外，也可变更常数值。例如，指定 K30V0 时，被执行的是作为 10 进制数的数值 K35 (K30+5)。
- 8 进制数的软元件：X, Y, KnX, KnY
例如，Z1=K8，执行 X000Z1 时，被执行的软元件编号为 X010 (X0+8: 8 进制数加法)。对于软元件编号为 8 进制数的软元件变址来说，V, Z 的内容要被换算成 8 进制数，然后做加法运算。因此，假定 Z1=K10，X000Z1 被指定为 X012，请务必注意此数不是 X010。
- 16 进制的数值：H
例如：以 V5=K30 指定常数 H30V5，则被认为是 H4E (30H+K30)。此外，以 V5=H30 指定常数 H30V5，则被认为是 H60 (30H+30H)。

* 在使用变址修饰软元件的时候，最后得到的修正编号不能超过软元件的编号范围。如 C0V0，若 V0=-1 则系统将会出错。

* 常数变址为环形的，K32767V0，V0=1，K 实际为 -32768。K-32768V0，V0=-1，K 实际为 32767。

* 执行 32 位指令时，非立即数操作数使用变址修饰，以 16 位软元件地址进行变址。比如 D1000V0，当 V0 为 1 时，该操作数指向 D1001 (指向 D1001~D1002 这 2 个 16 位软元件的组合)。

因此当软元件本身属性为 32 位时 (比如 C200)，不建议使用变址修饰，因为它并不会指向下一个软元件序号，而是指向本序号的高 16 位。

* 执行 32 位指令时，立即数操作数使用变址修饰，变址软元件取 32 位进行运算。比如 K1000V0，如果 V0=-1，V1=0，那么 V0 的 32 位为 65535 (取 V0, V1 的组合值，V1 为高 16 位，V0 为低 16 位)，整体计算结果为 66535，而非 K999。

5.1.9 标记指针[P], [I]

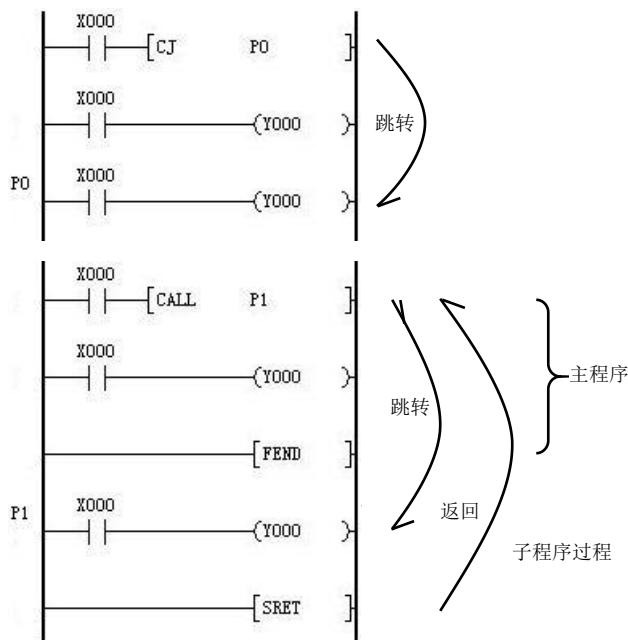
标记指针 P 用于用户调用子程序或程序跳转。

标记指针 I 用于中断程序标号，其对应的中断类型有定时中断，外部中断，计数中断及高级中断。

- 标记指针 P 编号为：P0~P127。
- 标记指针 I 编号为：
 - ◆ I6△△~I8△△：定时中断使用。对应的编号为毫秒定时数。如 I610 对应 10ms 进入一次中断。
 - ◆ I010~I060：计数中断使用。
 - ◆ I△00：外部事件中断使用，其中百位序号对应数值 0~3。其硬件输入端子指向 X4~X7。
 - ◆ I500：1ms 硬件插补中断。
 - ◆ I900~I907：高级中断使用。

程序位置指针 P

- 在控制程序流程时，作为分支标志或子程序名称，在 CJ、CALL 指令中使用。
- P 作指令操作数时可变址修饰。
- P 标识程序位置时（紧靠左母线）不能变址修饰。



- CJ 指令操作 P 软元件。
如果 X000 为 ON, 通过 CJ 指令跳转到指定的 P0 标号位置，执行随后的程序。
- CALL 指令操作 P 软元件。
如果 X000 为 ON, 通过 CALL 指令跳转到指定的 P1 标号位置，执行随后的程序。当执行完毕后（执行 SRET 指令），程序将返回到 CALL 指令的下一条指令，执行随后的程序。

中断位置指针 I

中断用指针有以下四种常用类型，与应用指令 IRET（中断返回），EI（开中断）和 DI（关中断）一起使用。

- 输入中断使用：接受来自特定的输入编号的输入信号，而不受 PLC 的扫描周期的影响。触发该输入型号，执行中断子程序。通过输入中断可处理比扫描周期更短的信号，因而可在顺控过程中作为必要的优先处理或短时脉冲处理控制中使用。
- 定时器中断使用：在各指定的中断循环时间（10ms~99ms）执行中断子程序。在需要有别于 PLC 的运算周期的循环中断处理控制中使用。
- 计数器中断使用：根据 PLC 的高速计数器 C 的比较结果，执行中断子程序。用于利用高速计数器优先处理计数结果的控制。
- 硬件 1MS 中断使用：可在此中断更新脉冲的插补运算。
- 高级中断使用：在运动类控制中, 需要更优化的中断方案。高级中断中支持进程调度语句, 采用高级中断模式可更好的满足高速并发任务的处理。

* 关于中断的具体使用方法，请参阅第 5.5 章节内容。

5.1.10 常数[K], [H]

常数项在可编程逻辑控制器内用以下进制的方式进行划分和说明。

- 十进制数：用 K 进行表示。
 - ◆ 在计数器、定时器设定值时以及在功能指令中作操作数时使用。
 - ◆ K 可变址修饰：如 $V0 = 1100$ ，则 $K100V0 = K1200$ ； $Z0 = -1100$ ， $K1000Z0 = K-100$ 。但在基本逻辑指令中，K 不能变址修饰，例如 $OUT\ T0\ K100$ 指令中 K 不能变址修饰。
 - ◆ 同时类似 $KnM0$ 组合中，K 不能变址修饰，但整体仍可变址修饰，其修饰的对象为软元件编号本身。例如， $V1 = 10$ ， $K4MOV1 = K4M10$ ；若 $Z1 = 1$ ， $V1 = 6$ ，则 $(MOV\ K40Z1\ D1000V1) = (MOV\ K41\ D1006)$ ； $K1M0$ 表示 4 个位元件 ($M0\sim M3$)，同理， $K2M0$ 、 $K3M0$ 、 $K8M0$ 分别表示 8 个、12 个、32 个位元件。
16 位： $n\leq 4$ ，32 位： $n\leq 8$ 。
- 十六进制数：用 H 进行表示。和十进制数一样，用于指定操作数数值和动作时使用，也可用于变址，规则与 K 一致，但不能作为位组合。如 $V = 10$ ，则 $H0010V = H001A = K26$ 。
- 八进制数：只用于输入继电器 X，输出继电器 Y 的编号分配使用。
- 浮点数：在浮点数指令中输出结果将自动转化为浮点型，若输入操作数填写常数项 K，H，系统也将自动转换为浮点型数值参与运算。

5.1.11 掉电保护

设置 $D8020=2222$ ，则开启运动控制器的掉电保护功能。

当产生了掉电保护时，系统将会保存以下的软元件当前状态，下次上电时将上次掉电时的状态进行恢复(即上次掉电时的软元件数值)。

- 辅助继电器 M
- 状态继电器 S
- 定时器 T 寄存器数值
- 计数器 C 寄存器数值
- 数据寄存器 $D1000\sim D1999$
- 扩展寄存器 R
- 变址寄存器 Z/V

即使开启了掉电保护，也不会被保护的软元件状态有：

- 定时器 T 触点，线圈
- 计数器 C 触点，线圈
- 输入触点 X
- 输出线圈 Y
- 数据寄存器 $D2000\sim D7999$

以上不被保护的软元件上电，位元件为复位状态，字元件数值为 0。



注意事项

- 通过软件操作 PLC 从运行到停止，不会诱发掉电保护功能。

5.2 基本逻辑指令

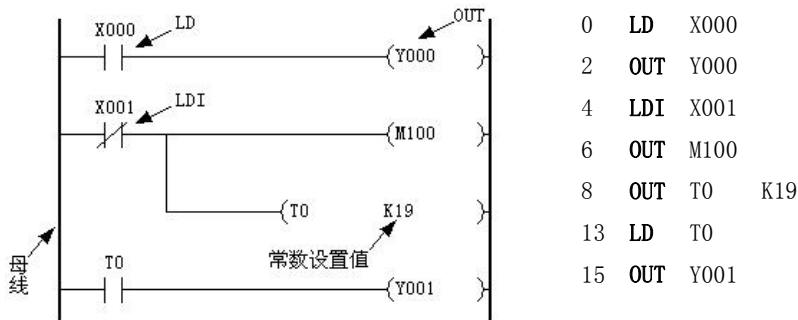
助记符、名称	功能	章节
LD/LDI	运算开始常开/常闭触点	5.2.1
OUT	线圈驱动	
AND/ANI	常开/常闭触点串联连接	5.2.2
OR/ORI	常开/常闭触点并联连接	5.2.3
LDP/LDF	上升/下降沿检出运算开始	5.2.4
ANDP/ANDF	上升/下降沿检出串联连接	
ORP/ORF	上升/下降沿检出并联连接	
ORB	串联回路块的并联连接	5.2.5
ANB	并联回路块的串联连接	5.2.6
MPS	运算存储	5.2.7
MRD	存储读出	
MPP	存储读出与复位	
MC	公共串联点连接线圈	5.2.8
MCR	公共串联点消除	
INV	取反	5.2.9
ALT	位取反	5.2.10
PLS/PLF	上升/下降沿输出	5.2.11
SET	动作保持	5.2.12
RST	清除动作保持, 寄存器清 0	
NOP	空动作	5.2.14
END	扫描周期结束	

5.2.1 [LD], [LDI], [OUT]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	LD 取	运算开始常开触点	 操作元: X, Y, M, S, T, C	2
	LDI 取反	运算开始常闭触点	 操作元: X, Y, M, S, T, C	2
	OUT 输出	线圈驱动	 操作元: Y, M, S, T, C	Y, M, S: 2 C(16位), T: 5 C(32位): 7

- 指令使用说明**
- LD, LDI 指令用于将触点连接到母线上。其他用法也可与后述的 ANB 等指令组合，在分支起点处使用。
 - OUT 指令是对输出继电器、辅助继电器、状态继电器、定时器、计数器的线圈驱动指令。对输入继电器不能使用。
 - 使用 OUT 指令控制定时器或计数器的线圈，同时还必须设定设置值常数 K 或 H，也可以使用数据寄存器 D 编号来间接指定。


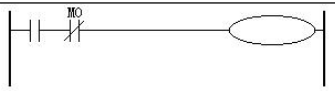
并列的 OUT 命令能多次连续使用，下图中 OUT M100 后紧跟着 OUT T0。



常数 K/H 的设定范围，实际的定时器常数，相对于 OUT 指令的程序步数如下表所示。

定时器, 计数器	常数的设定范围	实际的设定值	步数
1ms 定时器	1~32, 767	0.001~32.767 秒	5
10ms 定时器	1~32, 767	0.01~32.767 秒	5
100ms 定时器		0.1~32.767 秒	5
16 位计数器		1~32, 767	5
32 位计数器	-2, 147, 483, 648 ~ +2, 147, 483, 648	同左	7

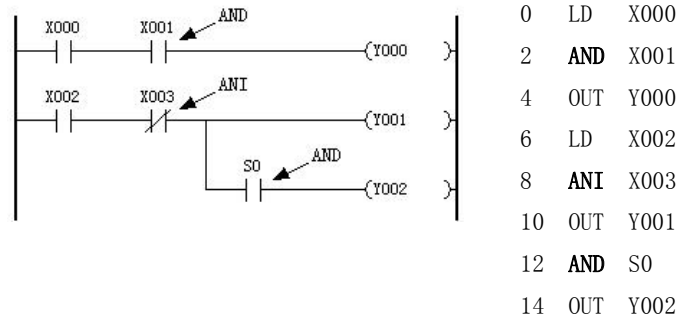
5.2.2 [AND], [ANI]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	AND 与	常开触点串联连接	 操作元: X, Y, M, S, T, C	2
	ANI 与非	常闭触点串联连接	 操作元: X, Y, M, S, T, C	2

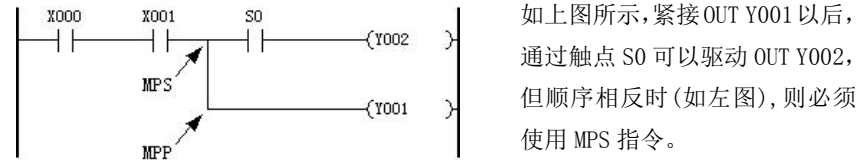
指令使用说明

- 用 AND, ANI 指令可串联连接 1 个触点。
- 串联触点数量不受限制，该指令可多次使用。
- 串联触点数量和纵接输出次数不受限制。

OUT 指令后，通过触点对其他线圈使用 OUT 指令，称之为纵接输出。（下图的 OUT Y001 与 OUT Y002）。这样的纵接输出如果顺序不错，可重复多次。





MPP, MPS 指令关系



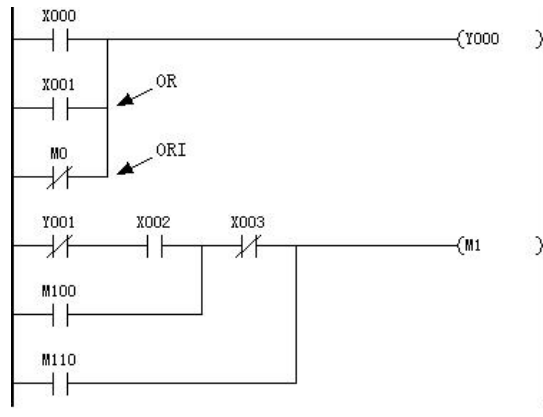
5.2.3 [OR], [ORI]指令

助记符与功能

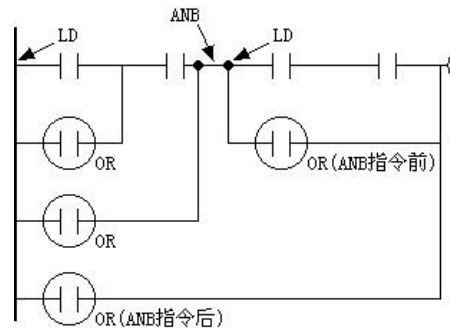
助记符、名称	功能	回路表示和可用软件	程序步
OR 或	常开触点并联连接	 操作元: X, Y, M, S, T, C	2
ORI 或非	常闭触点并联连接	 操作元: X, Y, M, S, T, C	2

指令使用说明

- OR, ORI 被用作一个触点的并联连接指令。如果有两个以上的触点串联连接, 并将这种串联回路块与其他回路并联连接时, 采用后述的 ORB 指令。
- OR, ORI 是指从该指令的步开始, 与前述的 LD、LDI 指令进行并联连接。并联连接的次数不受限制。



ANB 指令关系



使用 OR, ORI 指令所作的并联连接, 原则上是与前述的 LD, LDI 连接。

但在后述的 ANB 指令后, 则可在前面加一条 LD 或 LDI 指令连接。

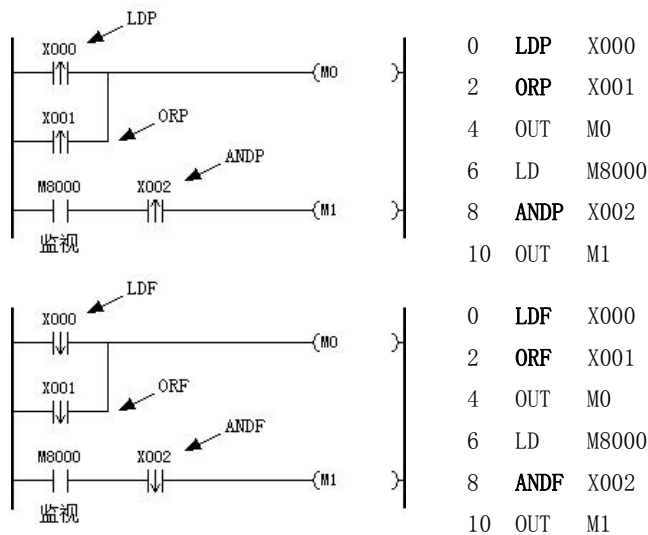
5.2.4 [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF]指令

助记符与功能

助记符、名称	功能	回路表示和可用软件	程序步
LDP 取脉冲上升沿	上升沿检出运算开始	 操作元: X, Y, M, S, T, C	2
LDF 取脉冲下降沿	下降沿检出运算开始	 操作元: X, Y, M, S, T, C	2
ANDP 与脉冲上升沿	上升沿检出串联连接	 操作元: X, Y, M, S, T, C	2
ANDF 与脉冲下降沿	下降沿检出串联连接	 操作元: X, Y, M, S, T, C	2
ORP 或脉冲上升沿	上升沿检出并联连接	 操作元: X, Y, M, S, T, C	2
ORF 或脉冲下降沿	下降沿检出并联连接	 操作元: X, Y, M, S, T, C	2

指令使用说明

- LDP、ANDP、ORP 指令是进行上升沿检出的触点指令，仅在指定位软元件的上升沿时 (OFF→ON 变化时) 接通一个扫描周期。
- LDF、ANDF、ORF 指令是进行下降沿检出的触点指令，仅在指定位软元件的下降沿时 (ON→OFF 变化时) 接通一个扫描周期。



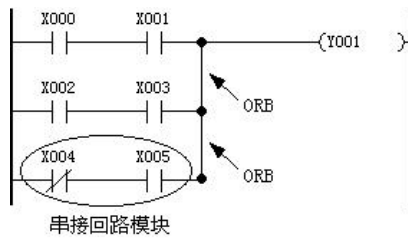
在上图中，X000-X002 由 ON→OFF 时或由 OFF→ON 变化时，M0 或 M1 仅有一个扫描周期接通。

5.2.5 [ORB]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	ORB OR 电路块或	串联回路块的并联连接		1

指令使用说明

- 由两个以上的触点串联连接的回路块被称为串联回路块。
- 将串联回路块并列连接时，分支开始用 LD, LDI 指令，分支结束用 ORB 指令。
- 有多个并联回路时，如对每个回路块使用 ORB 指令，则并联回路没有限制。
- ORB 指令是不带软元件编号的独立指令。
- ORB 指令也可成批使用，但是由于 LD, LDI 指令的重复次数在 8 次以下，因此请务必注意。



正确的程序

```

0 LD X000
2 AND X001
4 LD X002
6 AND X003
8 ORB
9 LDI X004
11 AND X005
13 ORB
14 OUT Y001
    
```

不佳的程序

```

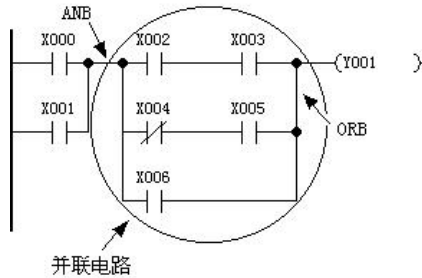
0 LD X000
2 AND X001
4 LD X002
6 AND X003
8 LDI X004
10 AND X005
12 ORB
13 ORB
14 OUT Y001
    
```


5.2.6 [ANB]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	ANB 回路块与	并联回路块的串联连接	 <p>操作元：无</p>	1

指令使用说明

- 当分支回路(并联回路块)与前面的回路串联连接时，使用 ANB 指令。
- 分支的起点用 LD, LDI 指令，并联回路块结束后，使用 ANB 指令与前面的回路串联连接。
- 若多个并联回路块按顺序和前面的回路串联时，ANB 指令使用次数没有限制。
- ANB 指令是不带软元件编号的独立指令。
- ANB 指令也可成批使用，与 ORB 指令一样，请务必注意 LD, LDI 指令的使用次数限制。



```

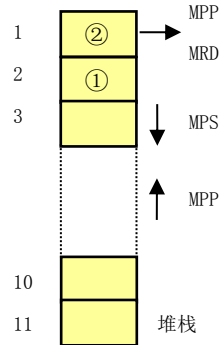
0 LD X000
2 OR X001
4 LD X002 ⇨分支起点
6 AND X003
8 LDI X004 ⇨分支起点
10 AND X005
12 ORB ⇨并联电路块结束
13 OR X006
15 ANB ⇨与前面的电路串联
16 OUT Y001
    
```

5.2.7 [MPS], [MRD], [MPP]指令

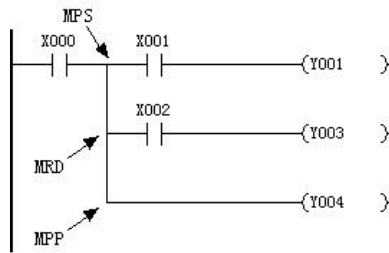
助记符与功能

助记符、名称	功能	回路表示和可用软件	程序步
MPS 进栈	运算存储		1
MRD 读栈	存储读出		1
MPP 出栈	存储读出与复位		1

指令使用说明



- 在 PLC 中有 11 个被称为栈的记忆运算中间结果的存储器。
- 使用一次 MPS 指令就将此时的运算结果送入栈的第一段存储。再次使用 MPS 指令，又将该时刻的运算结果送入第一段存储，将先前送入存储的数据依次移到栈的下一段。
- MPP 指令，使用后各数据按顺序向上移动，将最上端的数据移出，并且该数据从栈中消失。
- MRD 是读出最上端所存的最新数据的专用指令，栈内的数据不会发生移动。
- 这些指令都是不带软件编号的独立指令。

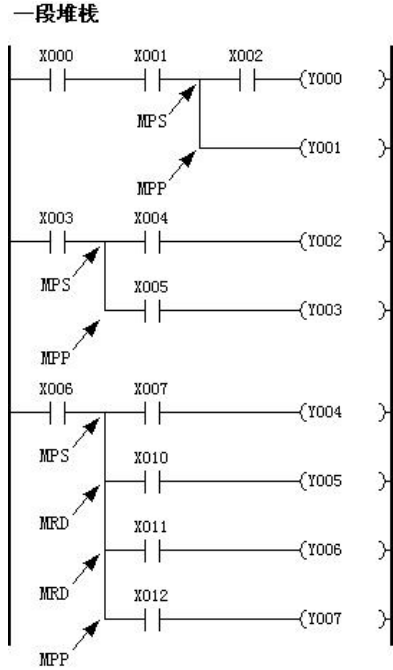


```

0 LD X000
2 MPS
3 AND X001
5 OUT Y001
7 MRD
8 AND X002
10 OUT Y003
12 MPP
13 OUT Y004
    
```

- 这项指令是进行上图所示的分支多重输出回路编程用的方便指令。利用 MPS 指令存储得出的运算中间结果，然后驱动 Y001。用 MRD 指令将该存储读出，再驱动输出 Y003。
- 最终输出回路以 MPP 指令替代 MRD 指令。从而在读出上述存储的同时将它复位。
- MPS 指令也可重复使用，MPS 指令与 MPP 指令的数量差额少于 11，但最终二者的指令数要一样。

编程示例一

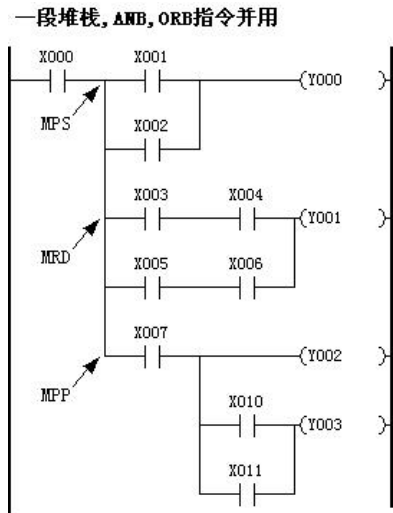


```

0 LD X000
2 AND X001
4 MPS
5 AND X002
7 OUT Y000
9 MPP
10 OUT Y001
12 LD X003
14 MPS
15 AND X004
17 OUT Y002
19 MPP
20 AND X005
22 OUT Y003
24 LD X006
26 MPS
27 AND X007
29 OUT Y004
31 MRD
32 AND X010
34 OUT Y005
36 MRD
37 AND X011
39 OUT Y006
41 MPP
42 AND X012
44 OUT Y007

```

编程示例二



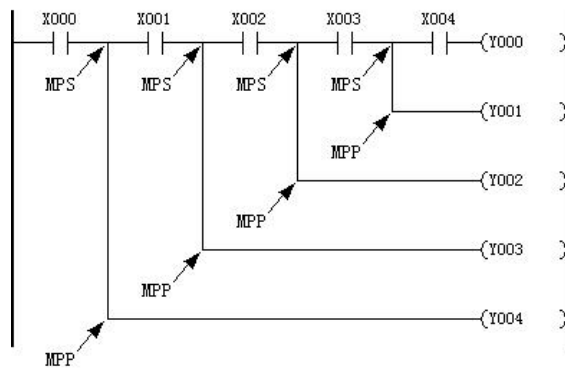
```

0 LD X000
2 MPS
3 LD X001
5 OR X002
7 ANB
8 OUT Y000
10 MRD
11 LD X003
13 AND X004
15 LD X005
17 AND X006
19 ORB
20 ANB
21 OUT Y001
23 MPP
24 AND X007
26 OUT Y002
28 LD X010
30 OR X011
32 ANB
33 OUT Y003

```

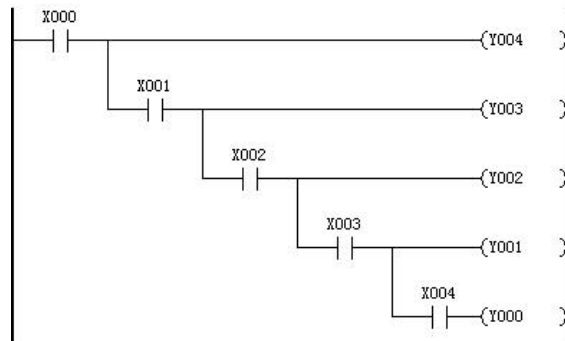
编程示例三

多段堆栈 (4段)



```

0 LD X000      14 OUT Y000
2 MPS         16 MPP
3 AND X001    17 OUT Y001
5 MPS         19 MPP
6 AND X002    20 OUT Y002
8 MPS         22 MPP
9 AND X003    23 OUT Y003
11 MPS        25 MPP
12 AND X004   26 OUT Y004
    
```

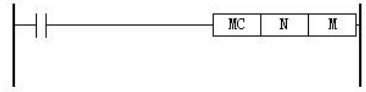



```

0 LD X000      10 OUT Y002
2 OUT Y004     12 AND X003
4 AND X001    14 OUT Y001
6 OUT Y003     16 AND X004
8 AND X002    18 OUT Y000
    
```

前图需采用多重 MPS 指令编程。但是，如果改用后图的回路，则不必采用 MPS 指令，使编程更为方便。

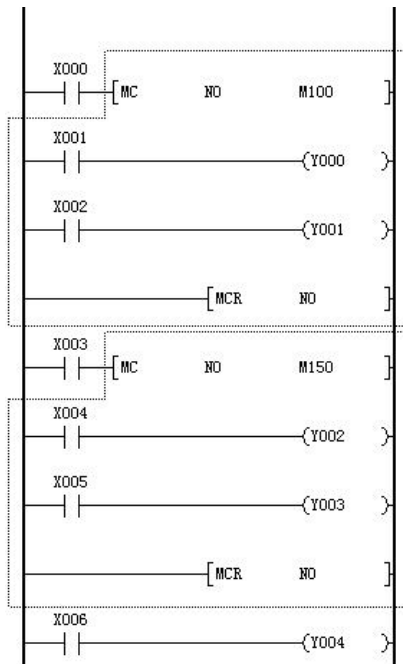
5.2.8 [MC], [MCR]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
助记符与功能	MC 主控	公共串联触点的连接	 操作元: Y, M(特殊 M 除外)	5
	MCR 主控复位	公共串联触点的消除	 操作元: Y, M(特殊 M 除外)	3

指令使用说明

- 在下述的编程示例中，输入 X000 接通时，就执行从 MC 到 MCR 的指令。
- 输入 X000 断开时，成为如下形式。
 - ◆ 现状保持: 累积定时器、计数器、用置位/复位指令驱动的软元件。
 - ◆ 变为断开的软元件: 非累积定时器、用 OUT 指令驱动的软元件。
- 执行 MC 指令后，母线(LD, LDI)向 MC 触点后移动，将其返回到原母线的指令为 MCR。
- 通过更改软元件号 Y, M, 可多次使用主控指令(MC)。但是如果使用同一软元件号，将同 OUT 指令一样，会出现双线圈输出。
- 主控对应的 M 继电器不允许使用变址修饰。

编程示例一



```

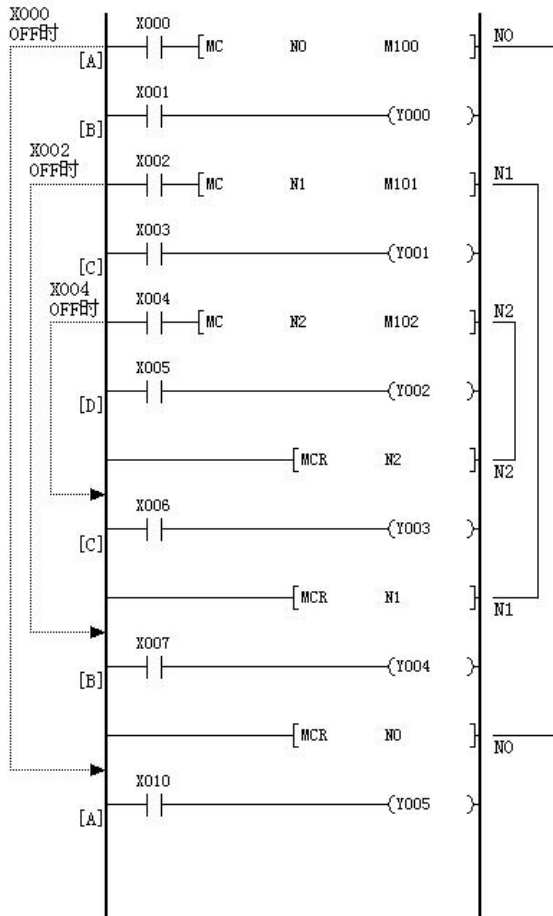
0  LD  X000
2  MC  NO  M100
7  LD  X001
9  OUT Y000
11 LD  X002
13 OUT Y001
15 MCR NO
    
```

⇐母线返回 (NO 为嵌套等级)

⇐在没有嵌套结构时，可再次使用 NO，次数无限制。如有嵌套结构，如下面的编程示例二，嵌套 N 的编号依次增大，最大为 7。

编程示例二

- 在 MC 指令内采用 MC 指令时，嵌套级 N 的编号顺序增大。(N0→N1→N2→N3→N4→N5→N6→N7)。在将该指令返回时，采用 MCR 指令，则从大的嵌套级开始消除(N7→N6→N5→N4→N3→N2→N1→N0)。
- 例如，MCR N6，MCR N7 不编程时，若对 MCR N5 编程，则嵌套级一下子回到 5。
- 嵌套级最大可编写 8 级(N7)。



《嵌套级 N0》
母线 B 在 X000 为 ON 时，呈激活状态。

《嵌套级 N1》
母线 C 在 X000, X002 为 ON 时，呈激活状态。

《嵌套级 N2》
母线 D 在 X000, X002, X004 为 ON 时，呈激活状态。

《嵌套级 N1》
通过 MCR N2，母线返回到 C 的状态。

《嵌套级 N0》
通过 MCR N1，母线返回到 B 的状态。

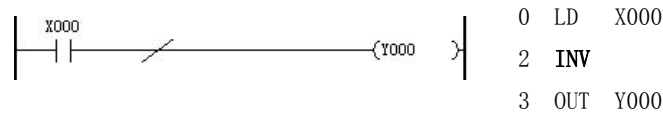
《初始状态》
通过 MCR N0，母线返回到最初 A 的状态。
因此，Y005 的接通/断开只取决于 X010 的接通断开状态，而与 X000, X002, X004 的状态无关。

5.2.9 [INV]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	INV 取反	运算结果反转	 操作元：无	1

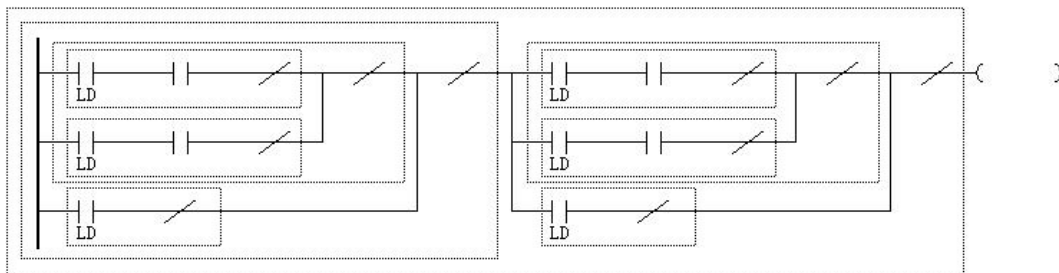
指令使用说明 INV 指令是将 INV 指令执行之前的运算结果反转的指令。不需要指定软元件号。

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



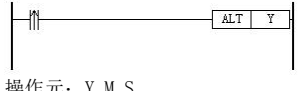
- 在上图中，如果 X000 断开，则 Y000 为 ON；如果 X000 接通，则 Y000 为 OFF。
- 在能输入 AND 或 ANI, ANDP, ANDF 指令步的相同位置处，可以编写 INV 指令。不能像指令表 LD, LDI, LDP, LDF 那样与母线连接，也不能像指令表中 OR, ORI, ORP, ORF 指令那样单独使用。

INV 指令动作范围 在包含 ORB 指令、ANB 指令的复杂回路中编写 INV 指令时，INV 指令的动作范围如下图所示：



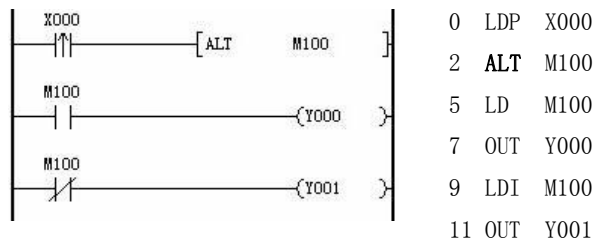
INV 指令的功能是将执行 INV 指令之前存在的 LD、LDI、LDP 和 LDF 指令以后的运算结果反转。因此，如上图所示，在 ORB 指令、ANB 指令中编写程序时，将把从各自的 INV 指令的位置见到的 LD, LDI, LDP, LDF 以后的程序作为 INV 运算的对象。

5.2.10 [ALT]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	ALT 位取反	将指定线圈状态取反	 操作元: Y, M, S	3

指令使用说明 执行 ALT 指令后可以将线圈的状态取反。由原来的 ON 状态改变为 OFF 状态，或由原来的 OFF 状态改变为 ON 状态。

执行 ALT 指令前的线圈状态	执行 ALT 指令后的线圈状态
OFF	ON
ON	OFF



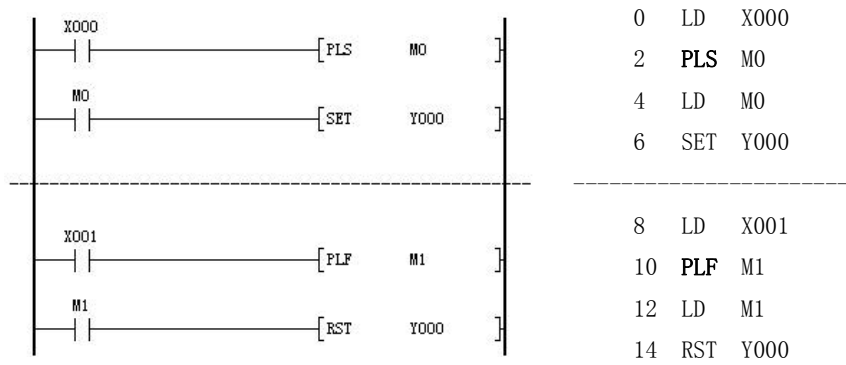
- 在上图中，如果 X000 信号有上升沿变化，则 M100 的当前状态取反。
- 若 M100 为 ON，则输出 Y000；若 M100 为 OFF，则输出 Y001。

5.2.11 [PLS], [PLF]指令

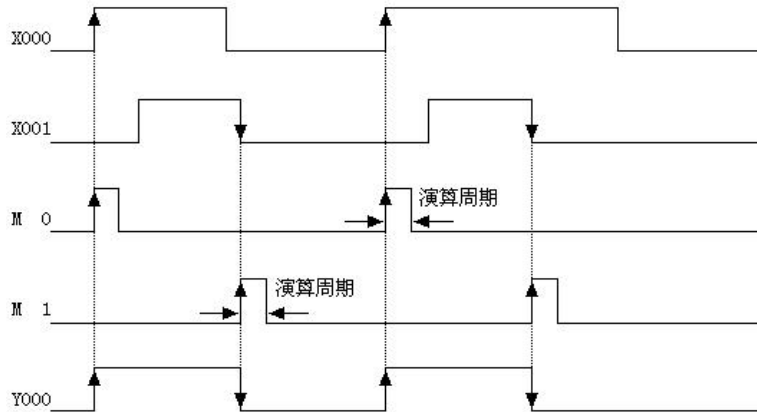
助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	PLS 脉冲	上升沿输出	 操作元: Y, M。特殊 M 除外	2
	PLF 下降沿脉冲	下降沿输出	 操作元: Y, M。特殊 M 除外	2

指令使用说明


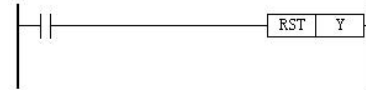
- 使用 PLS 指令时, 仅在驱动输入为 ON 后的一个扫描周期内, 软元件 Y, M 动作。
- 使用 PLF 指令时, 仅在驱动输入为 OFF 后的一个扫描周期内, 软元件 Y, M 动作。



以上示例的时序说明如下所示:

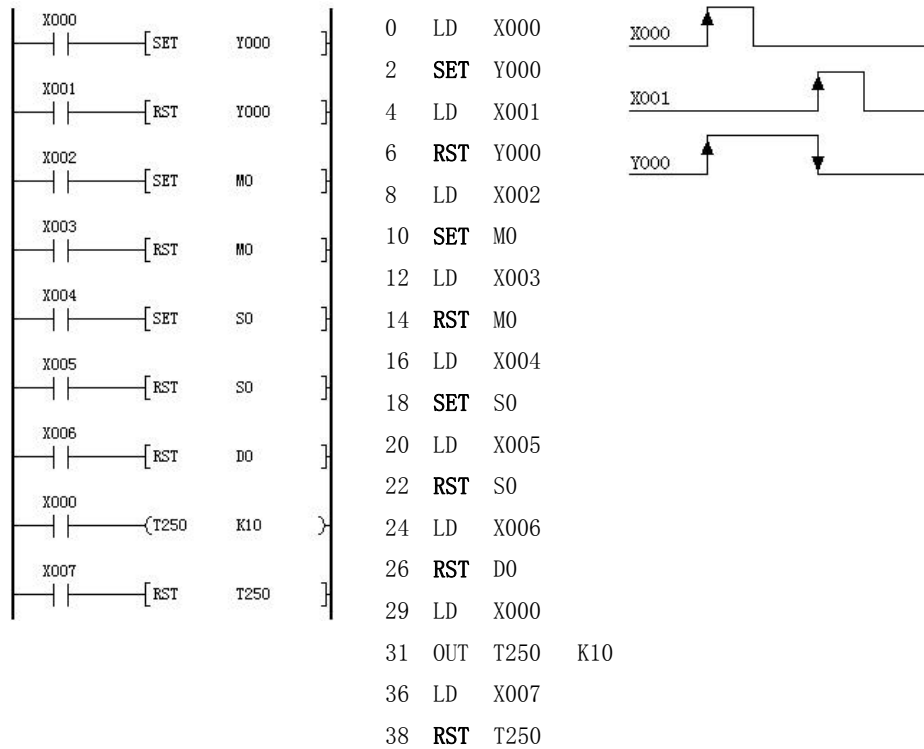


5.2.12 [SET], [RST]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	SET 置位	动作保持	 操作元: Y, M, S	2
	RST 复位	清除动作保持, 当前值及寄存器清零	 操作元: Y, M, S, T, C, D, R, V, Z	Y/M/S:2 T/C/D/R:3

指令使用说明

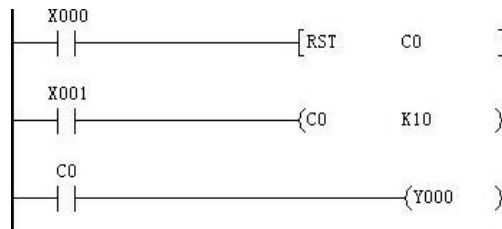
- 在下述程序示例中, X000 一旦接通后, 即使它再断开, Y000 仍继续动作。X001 一旦接通后, 即使它断开, Y000 仍保持不被驱动。对于 M、S 也是如此。
- 对于同一软元件, SET、RST 可多次使用, 顺序也可随意, 但最后执行者有效。
- 此外, 要使数据寄存器(D), 变址寄存器(V)、(Z)的内容清零时, 也可使用 RST 指令。(用常数为 K0 的传送指令也能得到相同的结果)
- 此外, 定时器、计数器当前值的复位以及触点复位也可使用 RST 指令。
- 使用 SET、RST 指令时, 应避免与 OUT 指令同时作用于相同的软元件。



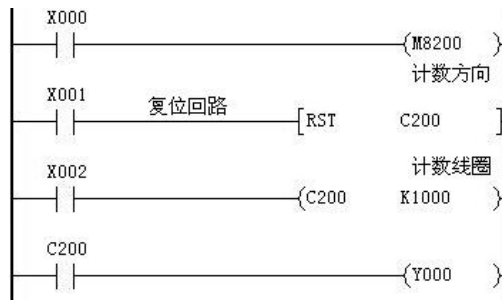
5.2.13 对应计数器软元件 C 的 [OUT], [RST] 指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	OUT 输出	计数线圈的驱动	 操作元:C	C(16位):5 C(32位):7
	RST 复位	输出触点的复位 当前值清零	 操作元:C	3

指令使用说明

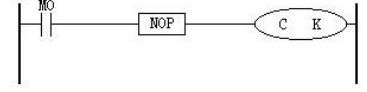



- 计数器 C0 对 X001 的 OFF→ON 次数进行增计数，当 C0 达到设定值 K10 时输出触点 C0 动作，即 Y000 动作。
- 此后，当 X001 继续由 OFF→ON 变化时，计数器的当前值会继续累加，输出触点仍保持动作。
- 令 X000 为接通状态，可清除计数器当前值并使触点复位。



- 在 C200~C234 的单相单输入的计数器中，采用特殊辅助继电器 M8200~M8234 指定计数的方向。
 - ◆ X000:ON 时，C200 为减计数；
 - ◆ X000:OFF 时，C200 为增计数。
- 令 X001 为接通状态，C200 输出触点复位，当前值清零。
- 通过 X002 触点的关断/闭合进行对 C200 计数器的计数，从而得到 X002 端子上的脉冲序列计数值。
- 当 C200 当前数值 ≥ 1000 时，C200 触点得电 (Y000 = ON)，反之触点失电 (Y000 = OFF)。

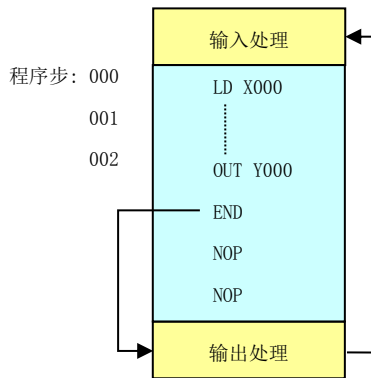
5.2.14 [NOP], [END]指令

助记符与功能	助记符、名称	功能	回路表示和可用软件	程序步
	NOP 空操作	无动作	 <p>操作元:无</p>	1
	END 结束	输入输出处理以及返回到 0 步	 <p>操作元:无</p>	1

NOP 指令使用说明

- 在将程序全部清除时，全部指令成为 NOP。
- 若在普通的指令与指令之间 NOP 指令，则 PLC 将无视其存在继续工作，但可以起到非常短暂的延迟执行效果。
- 若在程序中加入 NOP 指令，则在修改或追加程序时（进行转换操作），可以减少步号的变化，但是程序需要有冗余。
- 此外，若将已写入的指令换成 NOP 指令，则回路会发生变化。请务必注意。

END 指令使用说明



- PLC 反复进行输入处理、程序执行和输出处理。程序的最后必须以 END 指令结束，即使 END 以后仍有程序，也不再执行，而直接进行输出处理。
- 在调试阶段，各程序段插入 END 指令，可依次检出各程序段的动作。在确认前面回路块动作正确后，可以依次删除 END 指令，使程序段连续执行。
- 此外，RUN 开始时首次执行，从执行 END 指令开始。
- 执行 END 指令时，也刷新监视定时器（检查扫描周期是否过长）。

5.2.15 编程注意事项

触点的结构和步数

即使在动作相同的顺控回路中,根据触点的构成方法也可以简化程序与节省程序步数。一般编程的原则是:

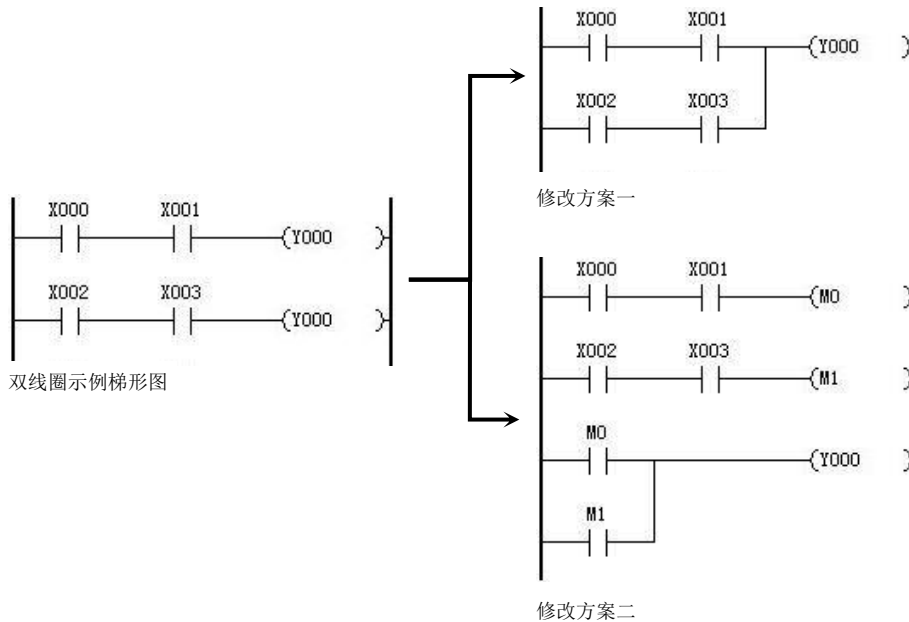
- 将串联触点多的回路写在上方。
- 将并联触点多的回路写在左方。

程序执行顺序

- 对顺控程序作【自上而下】和【自左而右】的处理。
- 顺控指令清单也沿着此流程编码。

双线圈执行策略

- 若在顺控程序中进行线圈的双重输出（双线圈），则后面的动作优先执行。
- 双重输出（双线圈）在程序方面并不违反输出规则，但是由于上述动作十分复杂，因此请按以下示例改变程序。





还有其他的方法，如采用跳转指令、流程指令，不同状态控制同一输出线圈编程的方法。

5.3 步进顺控指令

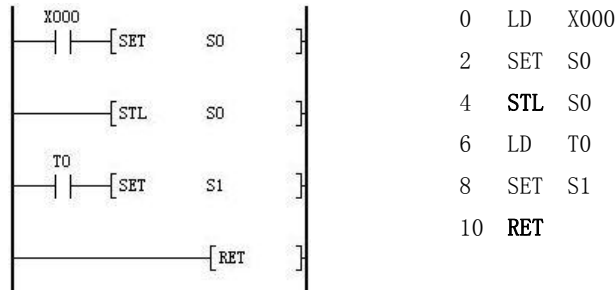
5.3.1 [STL], [RET]指令

助记符与功能

助记符、名称	功能	回路表示和可用软件	程序步
STL 步进梯形图	步进梯形图开始	 操作元:S	2
RET 返回	步进梯形图结束	 操作元:无	1

指令使用说明

- 步进梯形图指令 (STL) 是利用内部软元件状态 (S)，在顺控程序上面进行工序步进形控制的指令。返回 (RET) 是表示状态 (S) 流程的结束。置位指令 SET 是 STL 指令状态开始转移必备条件 (SET 一旦 ON, 不需自保, 解除用 RST)。
- 每一个流程书写都是独立的, 写法上不能嵌套书写。在流程执行时, 不一定要按 S0、S1、S2……的顺序执行, 流程执行的顺序在程序中可以按需求任意指定。可以先执行 S10 再执行 S5, 再执行 S0。
- 执行 RET 指令, 意味着步进梯形图回路的结束, 在希望中断一系列的工序而在主程序编程时, 同样需要 RET 指令, 该指令可多次编程。
- STL 指令和 RET 指令必须配对使用。在一系列 STL 指令之间, 可以不书写 RET 指令, 但在 STL 指令的最后, 若没有编写 RET 指令, 则会导致运行出错。



当 X000 触点闭合, 梯形图执行 S0 顺控指令。当 T0 定时器触点闭合, 程序转向 S1 顺控指令(上图未列出)。

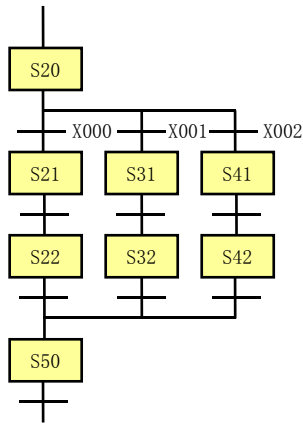
单一流程

- 在同一时刻, 只有一个状态在工作的工艺流程称为单一流程。
- 工序的转移的基本形式也是单一流程控制的。在要求单纯动作的顺序控制中, 只有单一流程已经足够了。但在具有各种输入条件与操作的情况下, 可通过与后述的选择分支及并行分支流程相结合, 从而简便地处理复杂的条件。

5.3.2 分支与汇合

选择性分支与汇合

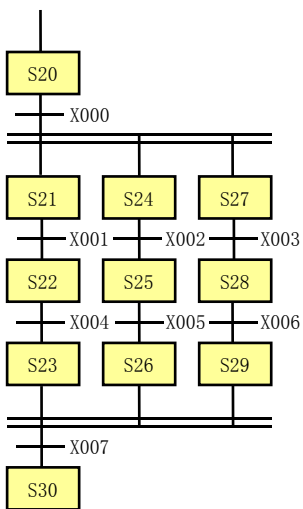
- 在步进顺控指令中，多个条件均可导致状态转移，但这些条件是互斥的：当一个条件成立时，另一个（或多个）条件便不能成立。如果一个条件看作是一个分支的话，我们称这样的分支为选择性分支。这些分支最终要归属到一个共同的状态，该过程就是选择性分支与汇合。



- 从多个流向中选择执行任何一个流向称为选择性分支。
- 左图中 X000、X001、X002 不能同时处于 ON 状态。
- 若 S20 动作且 X000=ON，则动作状态向 S21 移行。而若 S20 不动作，其后即使 X001 或 X002 动作，S31 或 S41 也不动作；
- 合流状态 S50 由 S22、S32、S42 中任意一个驱动。
- 在分支/合流的移行处理程序中，请不要使用 MPS、MRD、MPP、ANB、ORB 命令。在选择合流程序中，首先仅进行合流前状态的驱动处理，之后再依序进行状态的移行处理。

并行分支与汇合

- 能同时进行状态转移的分支称为并列分支。并列分支的状态转移完成后，一定会向一个共同的状态转移，称为汇合。
- 并列分支的状态由一个共同的状态诱发(例如 STL S20 及 LD X000)。
- 并列分支能独立的进行各自的状态转移。
- 各状态首先要进行驱动处理，然后才能进行状态转移，而且所有转移都要顺序进行(即三个条件：驱动、移动指定、移行条件)。
- 并行分支各自的状态转移完毕，都要向一个共同状态转移(如 STL S30)。

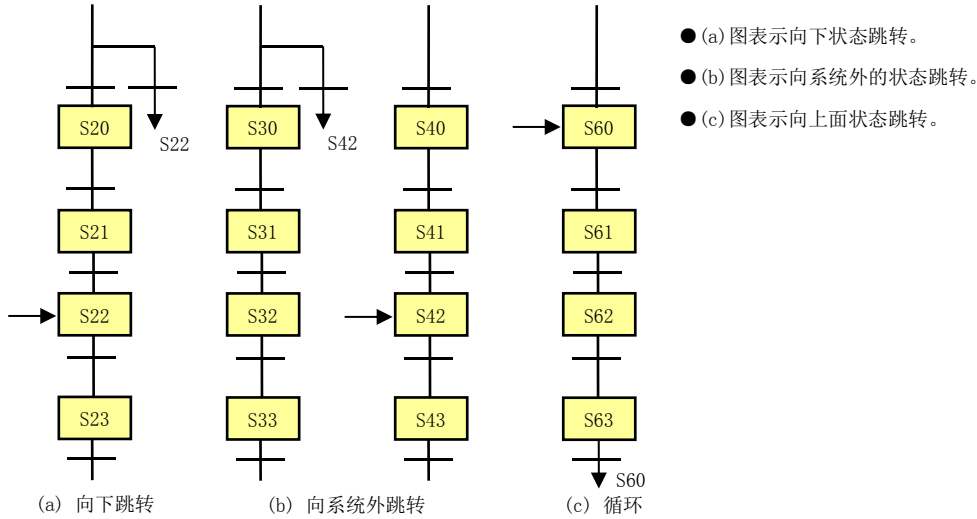


- 左图中，若 S20 动作且 X000=ON 时，则 S21、S24、S27 同时动作，各流向开始动作。
- 各流向动作都终止且 X007=ON 时，合流状态 S30 动作，而移行单元 S23、S26、S29 都不动作。
- 这样的合流又称为待合流。

5.3.3 循环与跳转

循环与跳转

在应用步进顺控指令时，SET Si 是状态转移的首要条件。跳过某些状态直接向下一状态转移或向系统外转移，称为跳转。以符号 ↓ 表示转移的目标状态。向上面状态转移称为重复或循环。同样 ↓ 表示转移的目标状态。



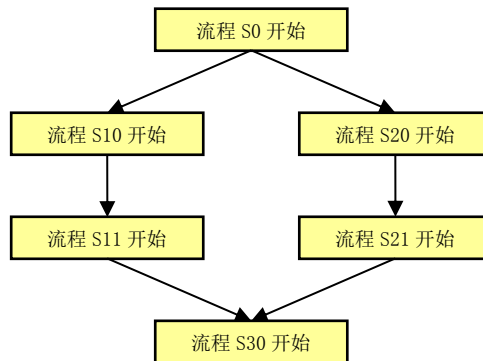
5.3.4 步进指令应用示例

应用编程示例

流程分支运行再合并到一个流程。

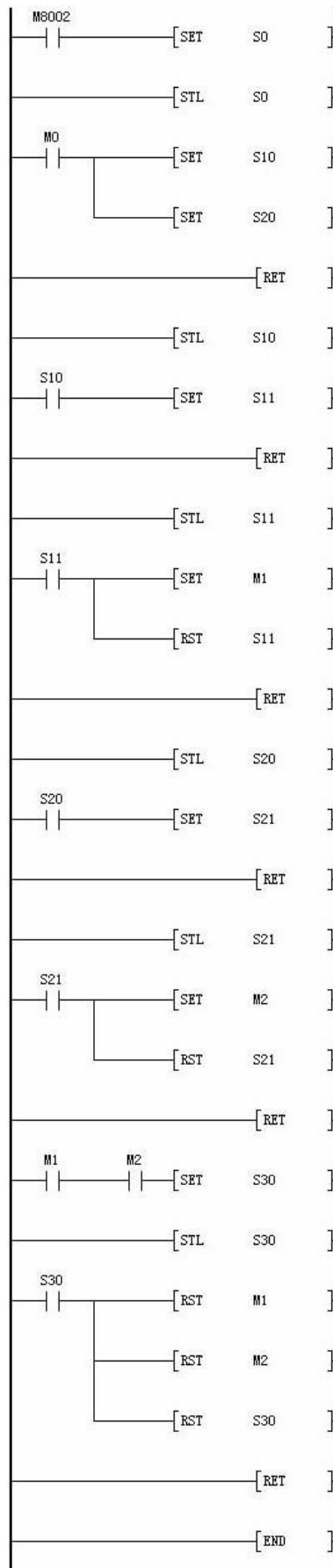
初始开始执行流程 S0，然后分支运行流程 S10 和 S20，直到 S10，S11 这一分支流程运行结束，而且 S20，S21 这一分支流程也运行结束之后，然后再合并运行流程 S30。

● 流程示意图



● 程序流程说明

- ◆ 程序开始运行，由特殊辅助线圈 M8002 置位 S0 线圈，流程 S0 开始运行。当线圈 M0 被置位时，从流程 S0 分支到流程 S10 和 S20。
- ◆ 分支流程：对应 2 个分支流程，从流程 (S10)/(S20) 开始，依次运行流程 (S10, S11)/(S20, S21)。当分支流程运行结束，置位线圈 (M1)/(M2) 作为分支流程完成标志位。
- ◆ 当两部分分支流程都运行结束，然后合并置位线圈 S30，流程 S30 开始运行。当流程 S30 结束则跳出流程，结束流程 S30。



- 特殊辅助继电器 M8002 上电首运行，置位 S0。
- 进入 S0 的 STL 流程。
- 当 M0 的状态为“ON”时，置位 S10, S20。
- 结束 S0 的 STL 流程。
- 进入 S10 的 STL 流程。
- 此时 S10 状态一定为“ON”，置位 S11。
- 结束 S10 的 STL 流程。
- 进入 S11 的 STL 流程。
- 此时 S11 状态一定为“ON”，复位 S11，置位 M1 标志分支流程结束。
- 结束 S11 的 STL 流程。
- 进入 S20 的 STL 流程。
- 此时 S20 状态一定为“ON”，置位 S21。
- 结束 S20 的 STL 流程。
- 进入 S21 的 STL 流程。
- 此时 S21 状态一定为“ON”，复位 S21，置位 M2 标志分支流程结束。
- 结束 S11 的 STL 流程。
- 当两个分支流程结束标志位都被置位时，置位 S30。
- 进入 S30 的 STL 流程。
- 此时 S30 状态一定为“ON”，复位 S30，复位 M1/M2 标志位。
- 结束 S30 的 STL 流程。
- 梯形图结束指令，本次扫描结束。

5.4 常规功能指令

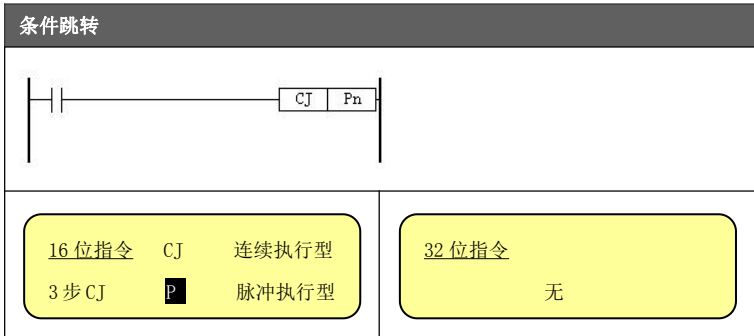
助记符、名称	功能	章节
程序流程		
CJ	条件跳转	5.4.1
CALL	子程序调用	5.4.2
SRET	子程序返回	
FEND	主程序结束	5.4.3
FOR	循环范围开始	5.4.4
NEXT	循环范围结束	
WDT	监视定时器	5.4.5
数据比较		
LD=	开始(S1)=(S2)导通	5.4.6
LD>	开始(S1)>(S2)导通	
LD<	开始(S1)<(S2)导通	
LD<>	开始(S1)≠(S2)导通	
LD>=	开始(S1)≥(S2)导通	
LD<=	开始(S1)≤(S2)导通	
OR=	并联(S1)=(S2)导通	5.4.7
OR>	并联(S1)>(S2)导通	
OR<	并联(S1)<(S2)导通	
OR<>	并联(S1)≠(S2)导通	
OR>=	并联(S1)≥(S2)导通	
OR<=	并联(S1)≤(S2)导通	
AND=	串联(S1)=(S2)导通	5.4.8
AND>	串联(S1)>(S2)导通	
AND<	串联(S1)<(S2)导通	
AND<>	串联(S1)≠(S2)导通	
AND>=	串联(S1)≥(S2)导通	
AND<=	串联(S1)≤(S2)导通	
数据传送		
CMP	数据比较	5.4.9
ZCP	数据区间比较	5.4.10
MOV	传送	5.4.11
BMOV	成批传送	5.4.12
FMOV	多点传送	5.4.13
SWAP	高低字节交换	5.4.14
XCH	两个数据交换	5.4.15
ZRST	批次复位	5.4.16

助记符、名称	功能	章节
CML	反向传送	5.4.17
数据运算		
ADD	加法	5.4.18
SUB	减法	5.4.19
MUL	乘法	5.4.20
DIV	除法	5.4.21
INC	加 1	5.4.22
DEC	减 1	
MEAN	求平均值	5.4.23
WAND	逻辑与	5.4.24
WOR	逻辑或	
WXOR	逻辑异或	5.4.25
NEG	求负	
SQR	开方	5.4.26
数据移位		
ROL	循环左移	5.4.27
ROR	循环右移	
RCL	带进位循环左移	5.4.28
RCR	带进位循环右移	
SFTL	位左移	5.4.29
SFTR	位右移	
WSFL	字左移	5.4.30
WSFR	字右移	
SFRD	移位读出	5.4.31
SFWR	移位写入	5.4.32
数据转换		
FLT	整型转浮点	5.4.33
INT	浮点转整型	5.4.34
BIN	BCD 转二进制	5.4.35
BCD	二进制转 BCD	5.4.36
ASCI	十六进制转 ASCII	5.4.37
HEX	ASCII 转十六进制	5.4.38
浮点运算		
ECMP	浮点比较	5.4.39
EZCP	浮点区间比较	5.4.40
EADD	浮点加法	5.4.41

助记符、名称	功能	章节
ESUB	浮点减法	5. 4. 42
EMUL	浮点乘法	5. 4. 43
EDIV	浮点除法	5. 4. 44
EBIN	浮点十进制转浮点二进制	5. 4. 45
EBCD	浮点二进制转浮点十进制	5. 4. 46
ESQR	浮点开方	5. 4. 47
SIN	浮点 SIN 运算	5. 4. 48
COS	浮点 COS 运算	5. 4. 49
TAN	浮点 TAN 运算	5. 4. 50
ASIN	浮点 ASIN 运算	5. 4. 51
ACOS	浮点 ACOS 运算	5. 4. 52
ATAN	浮点 ATAN 运算	5. 4. 53
RAD	浮点角度转弧度运算	5. 4. 54
DEG	浮点弧度转角度运算	5. 4. 55
专家指令		
REF	输入输出刷新	5. 4. 56
PID	PID 控制	5. 4. 57

5.4.1 条件跳转[CJ]

CJ、CJP 指令作为执行序列指令的一部分，可以缩短梯形图运算周期及使用双线圈。



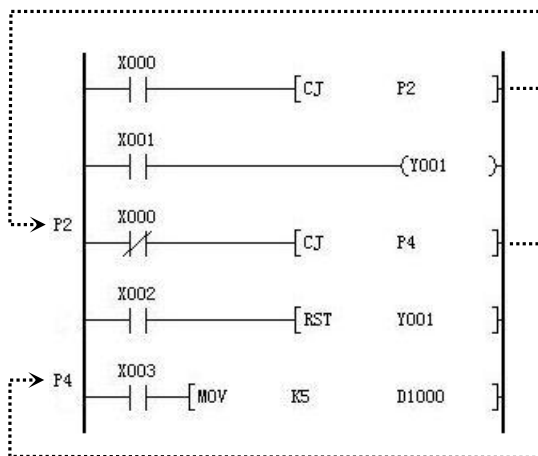
操作数	作用	类型
Pn	跳转到目标标记的指针编号 P(P0~P127)	指针编号

操作数	指针	
Pn	P	I
	●	

功能与动作

在下图的示例中：

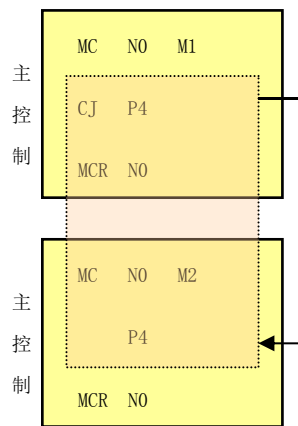
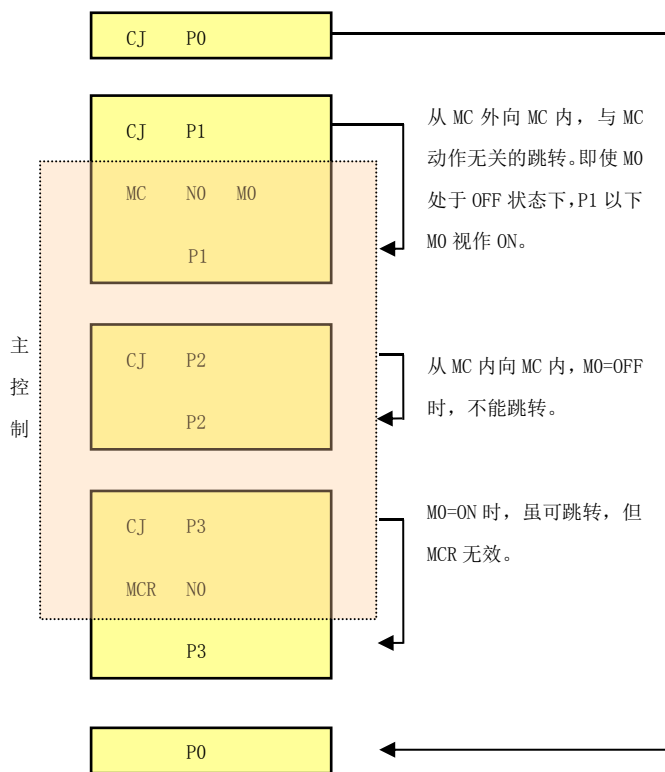
- 如果 X000 为“ON”状态时，程序跳转到 P2，之间语句不执行，其他语句正常执行，直到遇到 END 语句。
- 如果 X000 为“OFF”状态时，程序跳转到 P4，之间语句不执行，其他语句正常执行，直到遇到 END 语句。



- 如左图，Y001 有两条指令进行控制，实际即变成双输出。但是，X000=OFF 时采用 X001 动作，X000=ON 时采用 X002 动作。在一个演算周期内两条指令不会同时执行，即避免双线圈输出。
- CJ 不可以从一个 STL 跳转到另一个 STL。
- 使用跳转指令时注意标号一定要匹配。

编程与动作

主控指令与跳转指令的关系以及动作内容如下图所示。




从 MC 内向其他 MC 内的跳转。如果 M0 处于 ON 状态下，可以跳转。跳转以后的电路可视为与 M2 ON/OFF 无关的“ON”并动作，而且忽略最初的 MCR NO。

5.4.2 子程序调用[CALL]/子程序返回[SRET]

调用要共同处理的程序，可有效的减少程序步，同时也可以提高梯形图的执行效率。

子程序调用




<p>16 位指令 CALL 连续执行型</p> <p>3 步 CALL P 脉冲执行型</p>	<p>32 位指令</p> <p>无</p>
---	------------------------

操作数	作用	类型
Pn	跳转到目标标记的指针编号 P(P0~P127)	指针编号

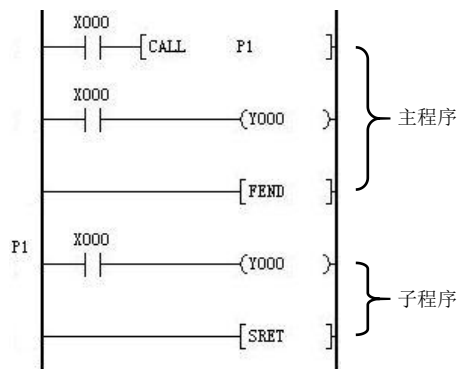
操作数	指针	
Pn	P	I
	●	

子程序返回



<p>16 位指令 SRET 连续执行型</p> <p>1 步</p>	<p>32 位指令</p> <p>无</p>
-------------------------------------	------------------------

功能与动作



- 如左图，如果 X000=ON，则执行调用指令，跳转到标记为 P1 的子程序步，执行子程序后，通过执行 SRET 指令后，返回到原来的主程序步，接着继续执行后续的主程序。
- 在后述的 FEND 指令后对标记编程。
- 调用子程序时，主程序所属的 OUT、PLS、PLF、定时器等均保持。
- 子程序返回时，子程序所属的 OUT、PLS、PLF、定时器等均保持。

编程注意事项

- 子程序调用指令对应的指针编号与子程序返回必须在 FEND 指令之后编写。
- 子程序调用指令必须与对应的指针编号一一对应。
- 子程序调用指令与跳转指令的区别在于：
 - ◆ CALL Pn 一旦执行，首先执行其指针编号 Pn 所指的下一步程序，直至遇到 SRET，再返回子程序调用指令的下一步执行。
 - ◆ CJ Pn 一旦执行，直接执行其指针编号 Pn 所指的下一步程序，并不返回原调用处。
- 子程序可嵌套使用，嵌套层数不要超过 8 层。

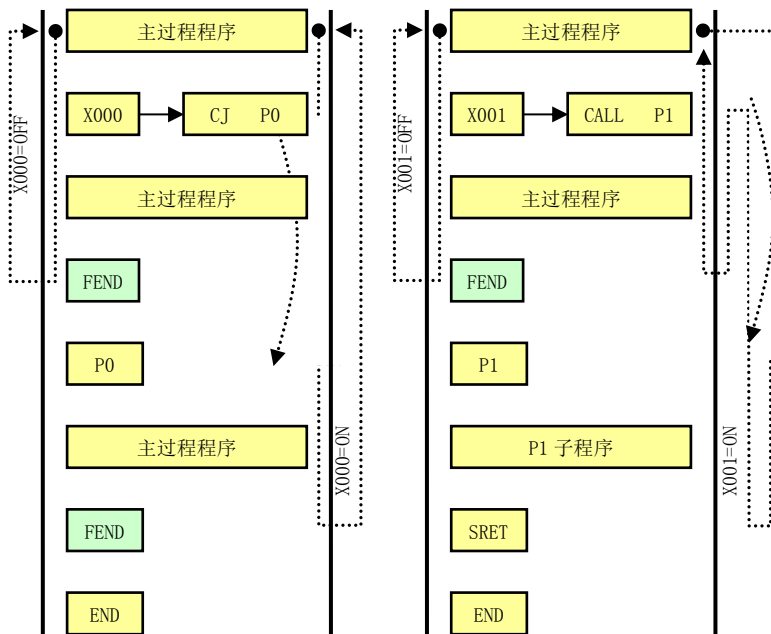
5.4.3 主程序结束[FEND]

主程序结束指令表示梯形图主程序扫描结束。

主程序结束	
16 位指令 FEND 连续执行型 1 步	32 位指令 无

功能与动作

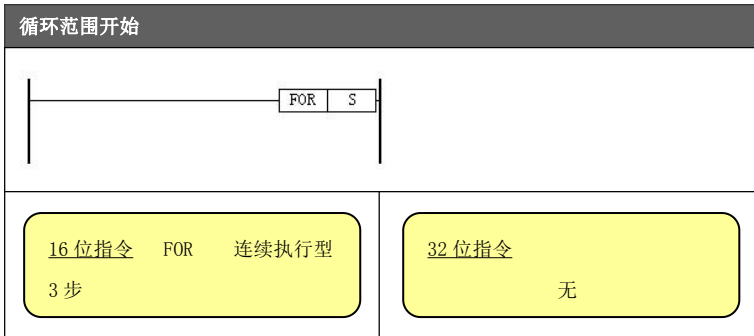
虽然[FEND]指令表示主程序的结束，但若执行此指令，则与 END 指令同样，执行输出处理、输入处理、监视定时器的刷新、向 0 步程序返回。



- CALL、CALLP 指令的标签在 FEND 指令后编程，必须要有 SRET 指令。中断程序也必须在 FEND 指令后编程，必须要有 IRET 指令。
- 在执行 CALL、CALLP 指令后，SRET 指令执行前，如果执行了 FEND 指令；或者在 FOR 指令执行后，NEXT 指令前执行了 FEND 指令，会导致程序运行出错。
- 使用多个 FEND 指令的情况下，必须在最后的 FEND 指令与 END 指令之间编写用户子程序或中断服务程序。

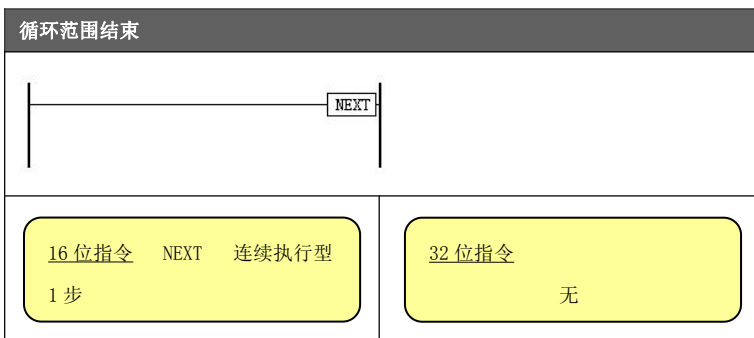
5.4.4 循环范围开始[FOR]/循环范围结束[NEXT]

使用范围循环指令，搭配软元件的变址使用，可有效处理数据区间的运算问题。简化编程的同时减少梯形图步数，但一般情况下不能缩短梯形图演算时间。



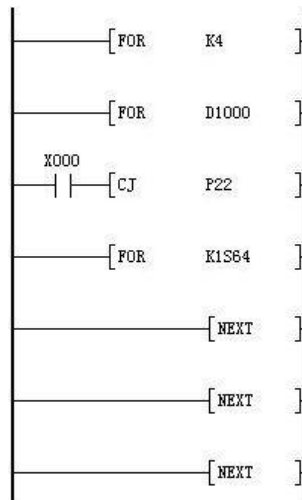
操作数	作用	类型
S	FOR~NEXT 之间程序的循环次数	16 位 BIN

操作数	字型软元件										
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
S	●	●	●								



功能与动作

只在 FOR~NEXT 指令之间的处理（利用源数据指定的次数）执行几次后，才处理 NEXT 指令以后的程序。



- 如左图，是一个三层嵌套的循环示例。假设 D1000 寄存器内容为 6，K1S64 组合内容为 7。
- 若 X000 处于断开状态，那么总计的循环次数为 $4 \times (D1000) 6 \times (K1S64) 7 = 168$ 次。
- 若不想执行 FOR~NEXT 间的程序时，当 X000 触点状态导通，则利用 CJ 指令跳出循环。

**编程注意事
项**

- 循环次数多时扫描周期会延长，可能造成监视器错误。
- 循环次数 1~32767 时有效，指定循环次数小于等于 0 时，被当作 1 处理。
- FOR/NEXT 必须配对使用，可以在循环内再次调用循环，其嵌套层数最多为 5 层。
- NEXT 指令在 FOR 指令之前，或无 NEXT 指令，或在 FEND, END 指令以后出现 NEXT 指令，或 FOR 指令与 NEXT 指令的个数不相等，都会出现错误。

若不想执行 FOR~NEXT 程序，可使用 CJ 指令跳出。

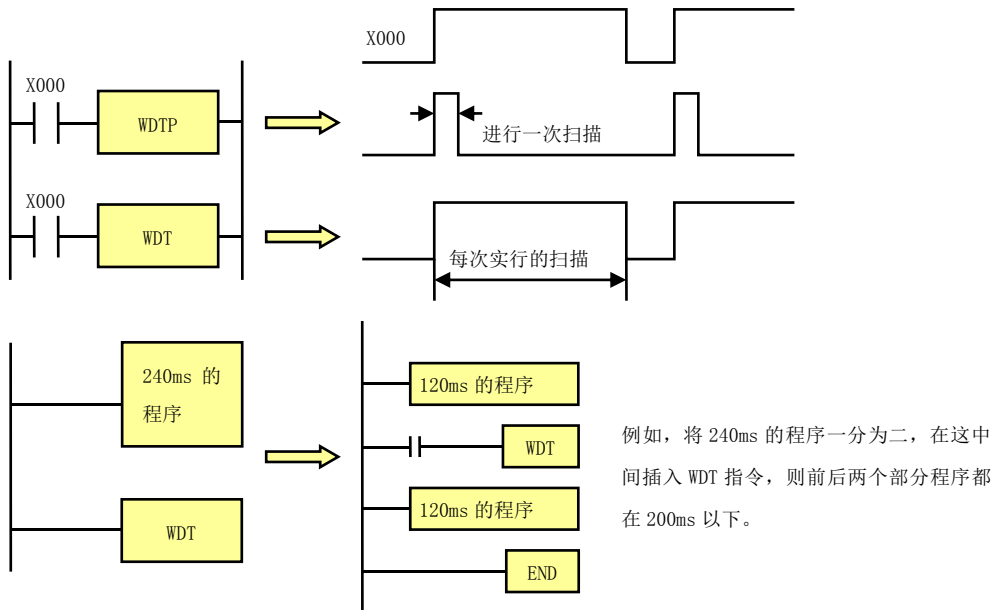
5.4.5 监视定时器[WDT]

在顺控程序中防止梯形图演算时间较长导致系统脱机，可执行监视用定时器刷新指令。

监视定时器	
16 位指令 WDT 连续执行型 1 步 WDT P 脉冲执行型	32 位指令 无

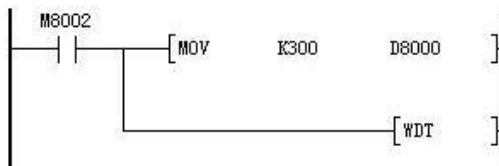
功能与动作

- 当 PLC 的运算周期（O-END 及 FEND 指令执行时间）超过 D8000(监视定时器时间)的设定值时，PLC 出错指示灯点亮同时停止工作，因此需要在编程过程中插入使用该指令。
- 在 CJ 指令标签的步号比 CJ 指令小时，可在标签后编写 WDT 指令。
- 若循环次数较多，可在 FOR/NEXT 指令之间编程时使用该指令。



监视定时器设置

通过改写 D8000(监视定时器时间)的内容，可改变监视定时器的检出时间。通过输入下述的程序，在这之后的 PLC 的程序将采用新的监视定时器时间执行监视。



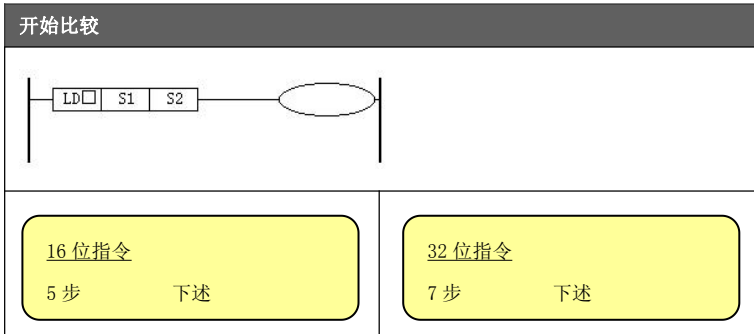
- 初始演算周期，将监视定时器时间设置为 300 毫秒。
- 如果程序内无运行监视定时器时间刷新指令 (WDT)，则在 END 处理时，D8000 的值才有效。

在系统构成中连接有很多台定位，凸轮开关、ID 接口、链接、模拟量等特殊扩展设备时，PLC 运行时进行的缓冲存储器初始化时间会延长，从而使扫描时间延迟。

在这种情况下，监视定时器有可能出错，因此请在起始步附近输入上述程序，将监视定时器时间延长。

5.4.6 开始比较[LD□]

LD□是连接母线的触点比较指令，对源数据内容进行 BIN 比较，对应其结果执行后段的运算。

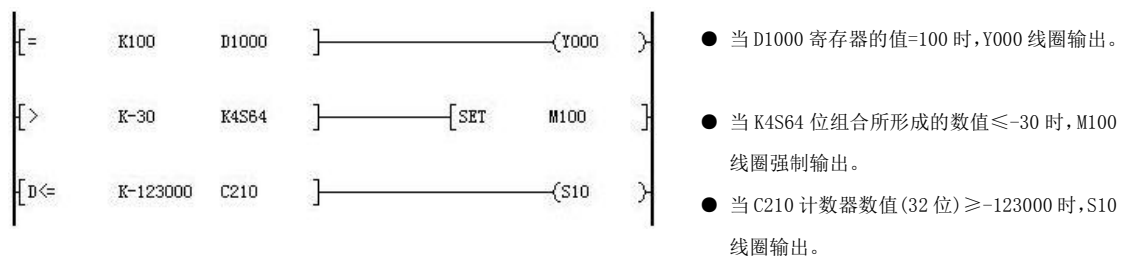


操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位 BIN
S2	指定比较数的数值或软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●

功能与动作

16 位指令	32 位指令	导通条件	非导通条件
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)

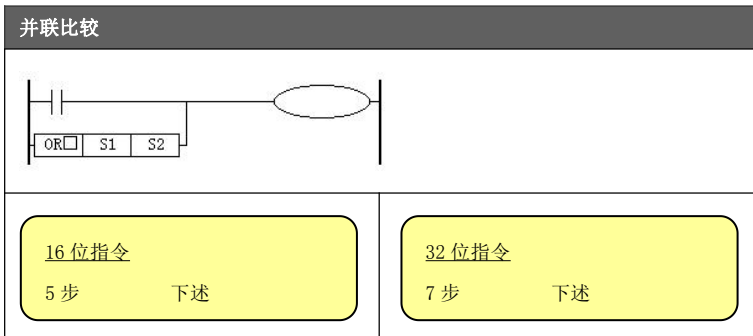


编程注意事项

- 当源数据的最高位（16 位指令：b15，32 位：b31）为 1 时，将该数值作为负数进行比较。
- 32 位计数器（C200~C255）的比较，必须以 32 位指令来进行。若指定为 16 位指令时，可能导致得不到正确的触点比较结果。
- 输入指令可简化指令助记符，比如“LD= K100 D1000”输入为“= K100 D1000”。

5.4.7 并联比较[OR□]

OR□是连接母线的触点比较指令，对源数据内容进行 BIN 比较，对应其结果执行后段的运算。

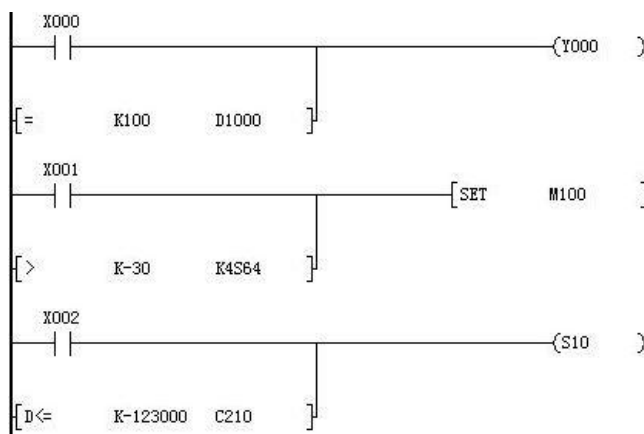


操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位 BIN
S2	指定比较数的数值或软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●

功能与动作

16 位指令	32 位指令	导通条件	非导通条件
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)



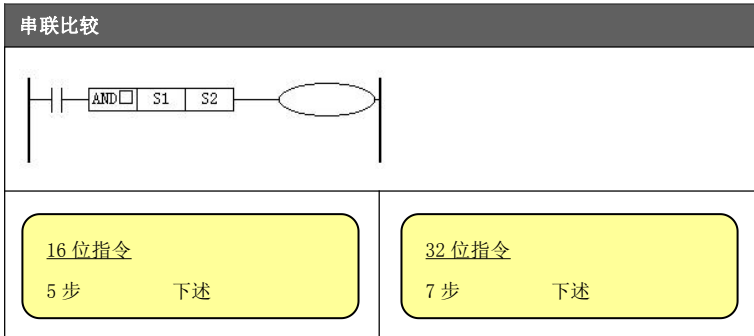
- 当 D1000 寄存器的值=100 或 X000 导通时，Y000 线圈输出。
- 当 K4S64 位组合所形成的数值≤-30 或 X001 导通时，M100 线圈强制输出。
- 当 C210 计数器数值(32 位)≥-123000 或 X002 导通时，S10 线圈输出。

编程注意事项

- 当源数据的最高位(16 位指令: b15, 32 位: b31) 为 1 时，将该数值作为负数进行比较。
- 32 位计数器(C200~C255) 的比较，必须以 32 位指令来进行。若指定为 16 位指令时，可能导致得不到正确的触点比较结果。
- 输入指令可简化指令助记符，比如“OR= K100 D1000”输入为“= K100 D1000”。

5.4.8 串联比较[AND□]

AND□是连接母线的触点比较指令，对源数据内容进行 BIN 比较，对应其结果执行后段的运算。

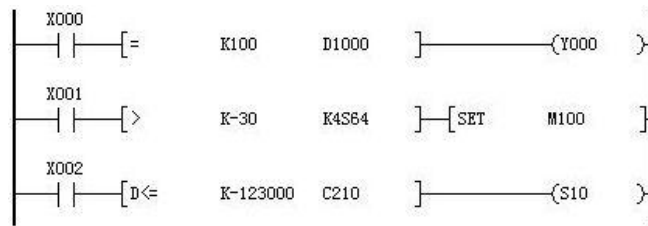


操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位 BIN
S2	指定比较数的数值或软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●

功能与动作

16 位指令	32 位指令	导通条件	非导通条件
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)



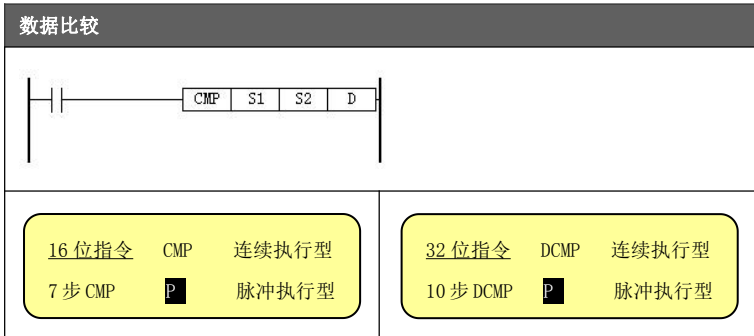
- 当 D1000 寄存器的值=100 且 X000 导通时，Y000 线圈输出。
- 当 K4S64 位组合所形成的数值 ≤ -30 且 X001 导通时，M100 线圈强制输出。
- 当 C210 计数器数值 (32 位) ≥ -123000 且 X002 导通时，S10 线圈输出。

编程注意事项

- 当源数据的最高位 (16 位指令: b15, 32 位: b31) 为 1 时，将该数值作为负数进行比较。
- 32 位计数器 (C200~C255) 的比较，必须以 32 位指令来进行。若指定为 16 位指令时，可能导致得不到正确的触点比较结果。
- 输入指令可简化指令助记符，比如“AND= K100 D1000”输入为“= K100 D1000”。

5.4.9 数据比较[**CMP**]

将指定的两个数据进行大小比较，并输出比较结果。

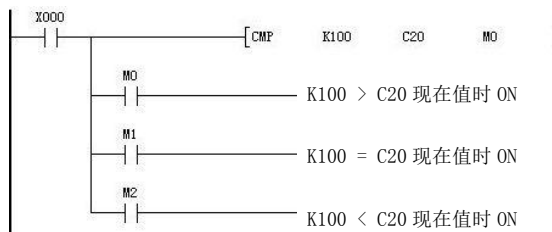


操作数	作用	类型
S1	指定被比较的数据或软元件地址编号	16/32 位 BIN
S2	指定比较源的数据或软元件地址编号	16/32 位 BIN
D	指定比较输出结果的软元件地址编号	位

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●

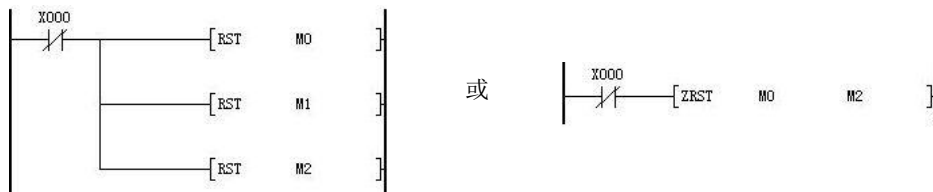
操作数	位型软元件			
D(占 3 点)	X	Y	M	S
		●	●	●

功能与动作



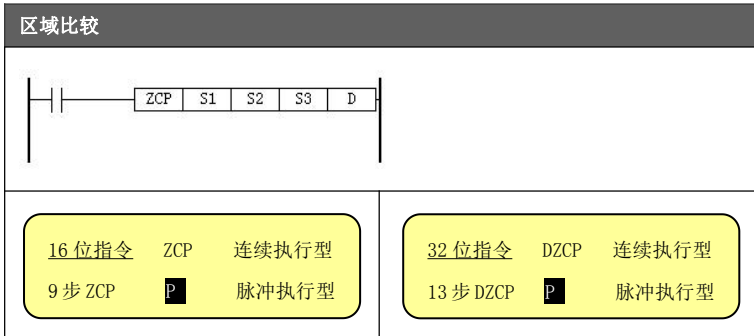
- 比较 K100 和 C20 当前值，根据结果驱动 M0、M1、M2 动作。
- 所有源数据都被看成二进制值处理。
- 作为目标地址假如指定 M0，如左记一样 M0、M1、M2 被自动占有。
- 在执行 **CMP** 指令后，即使 X000 断开，M0、M1、M2 仍然保持之前状态。

指令不执行时，想要清除比较结果的话，可使用复位指令。



5.4.10 区域比较[ZCP]

将 2 点指定数据与当前数据进行大小比较，并输出比较结果。

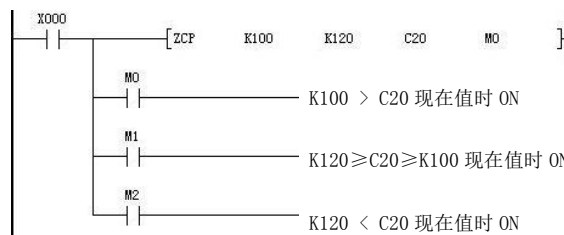


操作数	作用	类型
S1	指定比较基准下限的数据或软元件地址编号	16/32 位 BIN
S2	指定比较基准上限的数据或软元件地址编号	16/32 位 BIN
S3	指定当前数据或软元件地址编号	16/32 位 BIN
D	指定比较输出结果的软元件地址编号	位

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●

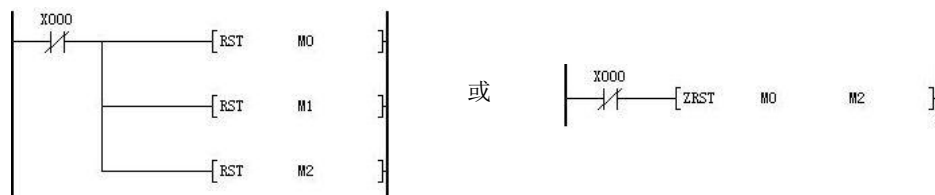
操作数	位型软元件			
D(占 3 点)	X	Y	M	S
		●	●	●

功能与动作



- 比较 K100 与 K120 区间和 C20 当前值，根据结果驱动 M0、M1、M2 动作。
- 所有源数据都被看成二进制值处理。
- 作为目标地址假如指定 M0，如左记一样 M0、M1、M2 被自动占有。
- 在执行 ZCP 指令后，即使 X000 断开，M0、M1、M2 仍然保持之前状态。

指令不执行时，想要清除比较结果的话，可使用复位指令。



5.4.11 传送[MOV]

使指定数据或软元件值传送到其他软元件中。

传送	
16 位指令 MOV 连续执行型 5 步 MOV P 脉冲执行型	32 位指令 DMOV 连续执行型 7 步 DMOV P 脉冲执行型

操作数	作用	类型
S	指定传送源数据或保存数据的软元件地址编号	16/32 位 BIN
D	指定传送目标软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●

功能与动作	指令	说明
		● 将源的内容向目标传送。
		● X000 状态为 OFF 时，数据不变化。
		● 将常数 K100 传送到 D1000。

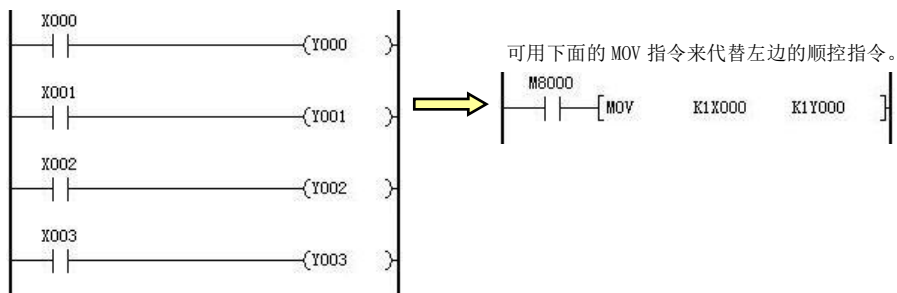
《定时器，计数器的当前值读出示例》

指令	说明
	● 将寄存器 T10 的当前值传送到 D1000。
	● 关于计数器也一样。

《定时器，计数器设定值的间接值读出示例》

指令	说明
	● 将 K10 传送到 D1100。
	● 将 D1100 的值作为 T0 定时器的定时时基(1 秒)。

《位元件的传送》



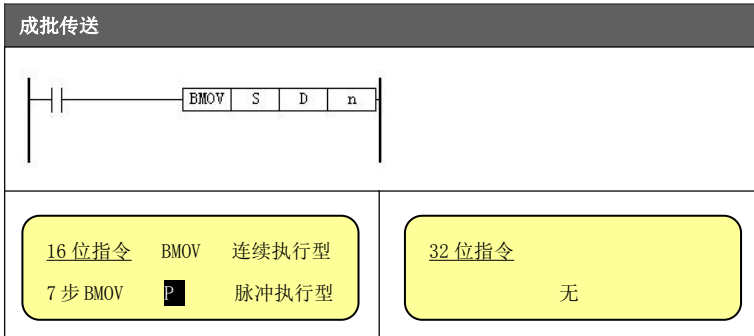
《32 位数据的传送》

运算结果以 32 位输出的应用指令(MUL 等)、32 位数值或 32 位软元件的高速计数器当前值等数据的传送，必须使用 DMOV 指令。



5.4.12 成批传送[BMOV]

使指定软元件区域值传送到其他软元件区域中。

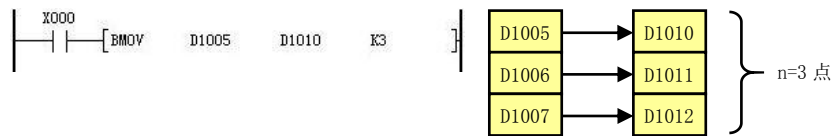


操作数	作用	类型
S	指定传送源软元件地址编号	16 位 BIN
D	指定传送目标软元件地址编号	16 位 BIN
n	指定传送点数的数值	16 位 BIN

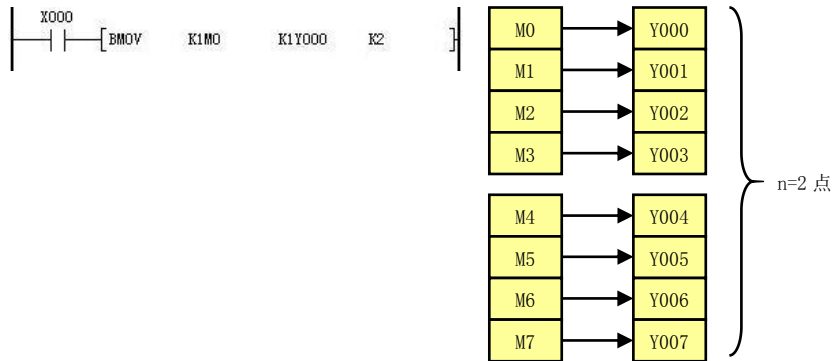
操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●	●	●	●	●	
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

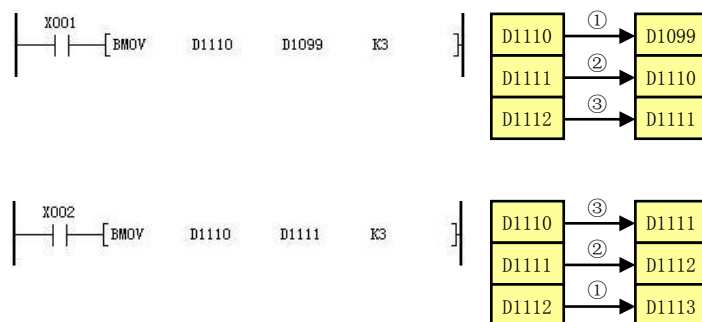
将以源指定软元件为开头的 n 点数据向以目标指定的软元件为开头的 n 点软元件成批传送。



带有位指定的位元件时，源和目标要采用相同的位数。



传送编号范围有重叠时，为了防止输送源数据没传送就改写，根据编号重叠的方法，按①~③的顺序自动传送。

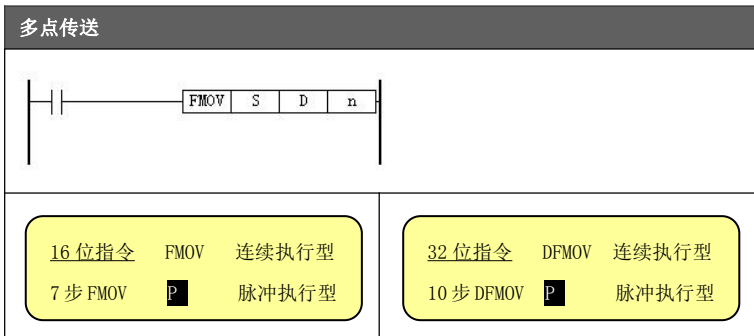


编程注意事
项

- 指令执行的传送点数最少为 1 个点，为提高指令效率，建议不超过 256 个点。
- 当指定的软元件编号加上传送点数后，所指向的软元件地址必须是连续有效的。但排除 C199 到 C200 和 D7999 到 D8000 这两类情况。

5.4.13 多点传送[FMOV]

使指定数据或软元件内容传送到其他软元件区域中。



操作数	作用	类型
S	指定传送源数据或保存数据的软元件地址编号	16/32 位 BIN
D	指定传送目标软元件地址编号	16/32 位 BIN
n	指定传送点数的数值	16/32 位 BIN

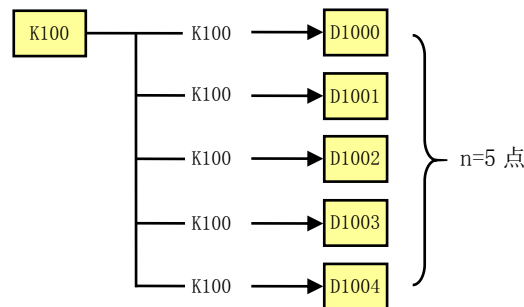
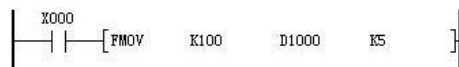
操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

将源指定的数据或软元件的内容向以目标指定的软元件为开头的 n 点软元件进行传送。n 点软元件的内容都一样。

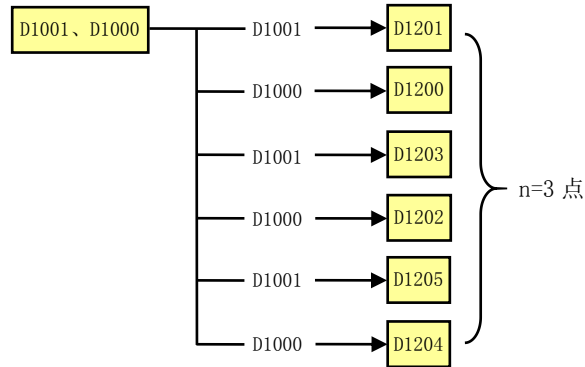
《16 位指令》

将 K100 赋予到 D1000~D1004, 同一数据的多点传送。



《32 位指令》

将 D1001、D1000 的内容赋予到 D1201、D1200；D1203、D1202；D1205、D1204。

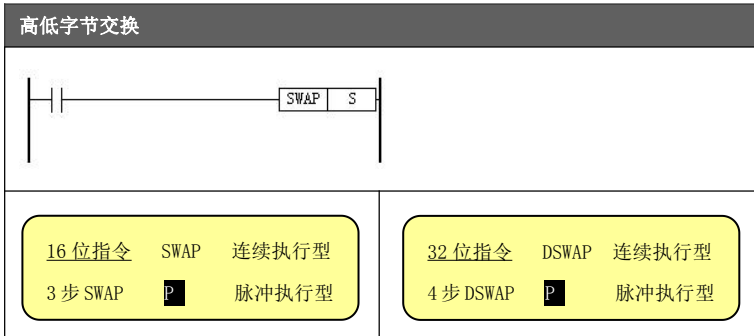


编程注意事
项

- 指令执行的传送点数最少为 1 个点，最多不超过 256 个点。
- 当指定的软元件编号加上传送点数后，所指向的软元件地址必须是连续有效的。但排除 C199 到 C200 和 D7999 到 D8000 这两类情况。

5.4.14 高低字节交换[SWAP]

将指定软元件的高 8 位字节和低 8 位字节进行交换。



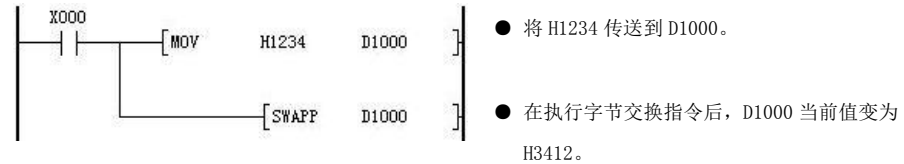
操作数	作用	类型
S	指定高低字节交换的软元件地址编号	16/32 位 BIN

操作数	字型软元件										
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
S		●	●	●	●		●	●	●	●	

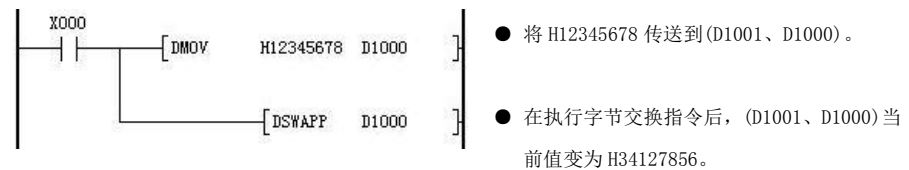
功能与动作

- 为 16 位指令时，交换 1 个字的高低字节。
- 为 32 位指令时，交换 2 个字的高低字节。

《16 位指令》- 为方便说明，立即数采用 16 进制显示。



《32 位指令》- 为方便说明，立即数采用 16 进制显示。



若使用连续型高低字节交换指令，则每个扫描周期均会执行一次高低字节交换。

5.4.15 两个数据交换[XCH]

将两个指定的软元件中的数据进行交换。

两个数据交换

16 位指令	XCH	连续执行型
5 步 XCH	P	脉冲执行型

32 位指令	DXCH	连续执行型
7 步 DXCH	P	脉冲执行型

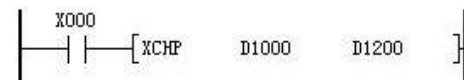
操作数	作用	类型
D1	指定互换的软元件地址编号	16/32 位 BIN
D2	指定互换的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D1		●	●	●	●		●	●	●	●
D2		●	●	●	●		●	●	●	●

功能与动作

目标间的数据交换。

《16 位指令》



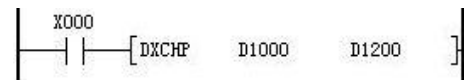
● 执行前 (D1000)=100

(D1200)=101

● 执行后 (D1000)=101

(D1200)=100

《32 位指令》



● 执行前 (D1001、D1000)=123456

(D1201、D1200)=654321

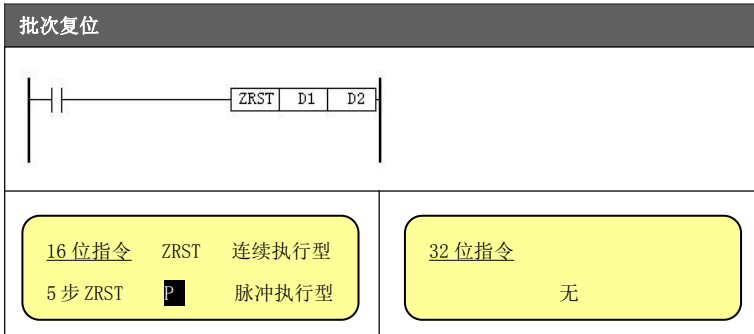
● 执行后 (D1001、D1000)=654321

(D1201、D1200)=123456

若使用连续型两个数据交换指令，则每个扫描周期均会执行一次数据交换。

5.4.16 批次复位[ZRST]

将指定软元件的数据或位清零或复位。



操作数	作用	类型
D1	指定批次复位的起始软元件地址编号	16/32 位 BIN; 位
D2	指定批次复位的结束软元件地址编号	16/32 位 BIN; 位

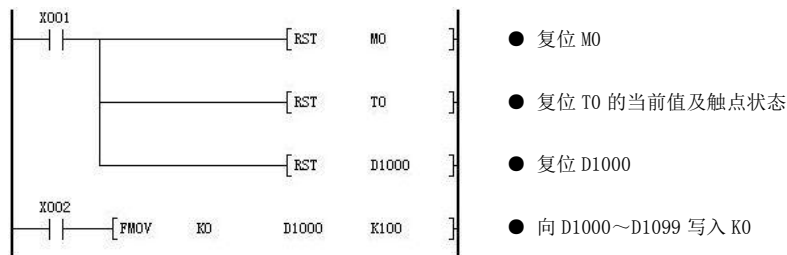
操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●					
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●					

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●
D2	X	Y	M	S
		●	●	●



其他复位指令

- 作为软元件的单独复位指令, 对于位元件 Y, M, S 和字元件 T, C, D 可使用 RST 指令。
- 作为常数 K0 的成批写入指令有 FMOV 指令, 可以把 0 写入 KnY, KnM, KnS, T, C, D 的软元件中。

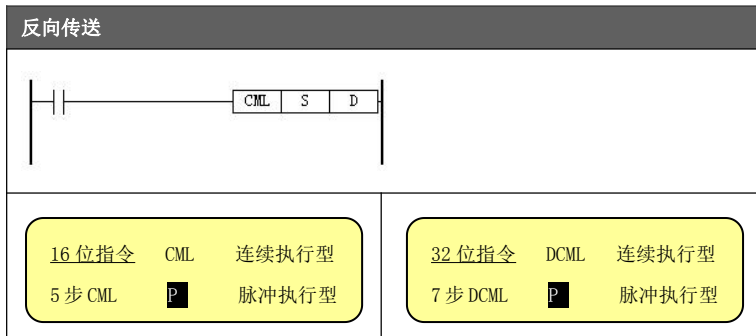


**编程注意
事项**

- D1, D2 必须指定为同一类型的软元件。
- D1 指定的编号必须 \leq D2 指定的编号。
- 软元件对象指定为 C 或者 T 时, 除了清除寄存器数值, 还将复位其触点。
- D1, D2 在指定软元件地址时应注意避免以下设置(非连续物理内存地址):
 - ◆ D1: 指定 \leq M3071; D2 指定 \geq M8000。
 - ◆ D1: 指定 \leq D7999; D2 指定 \geq D8000。

5.4.17 反向传送[CML]

将指定数据或软元件内容进行反向传送。

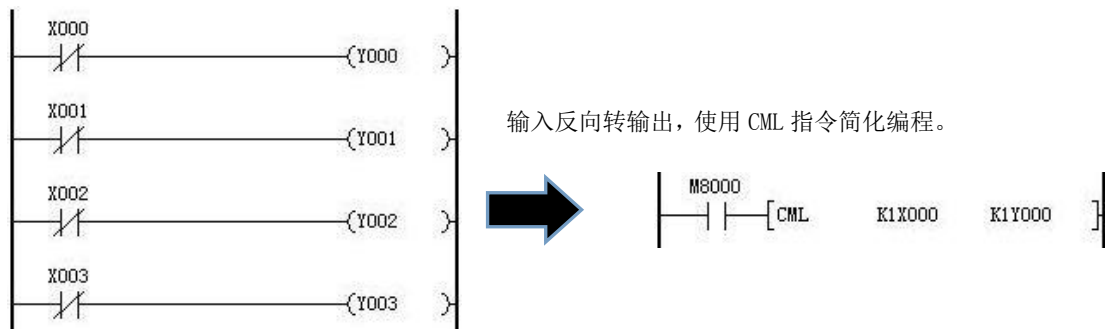
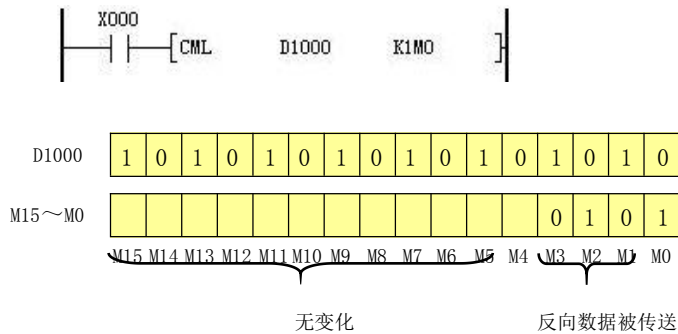


操作数	作用	类型
S	指定源数据值或软元件地址编号	16/32 位 BIN
D	指定保存结果的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
S	●	●	●	●	●	●	●	●	●	●
D		●	●	●	●		●	●	●	●

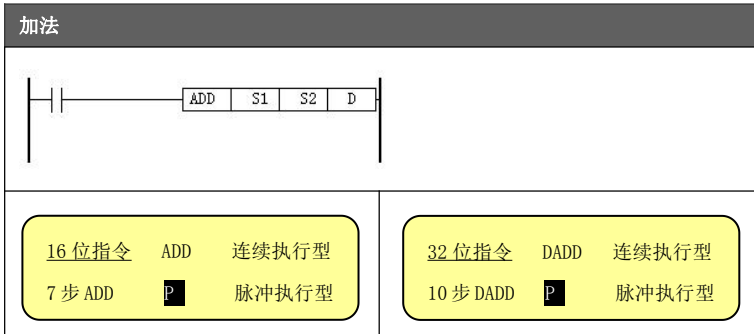
功能与动作

- 将源数据的每位反向 (0→1, 1→0) 后, 传送到目标地址。在源数据中使用常数 K 的话, 能自动地转换成二进制。
- 希望将 PLC 的输出以逻辑反相输出时, 可以使用。



5.4.18 加法[ADD]

使两个数据或软元件内容进行二进制加法运算并将结果存储到其他软元件中。



操作数	作用	类型
S1	指定进行加法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行加法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存加法运算结果的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
S1	●	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●	●
S2	●	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●	●
D	●	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●	●

功能与动作

两个源数据进行二进制加法后传送到目标处。各数据的最高位是正(0)、负(1)的符号位，这些数据以代数形式进行加法运算。

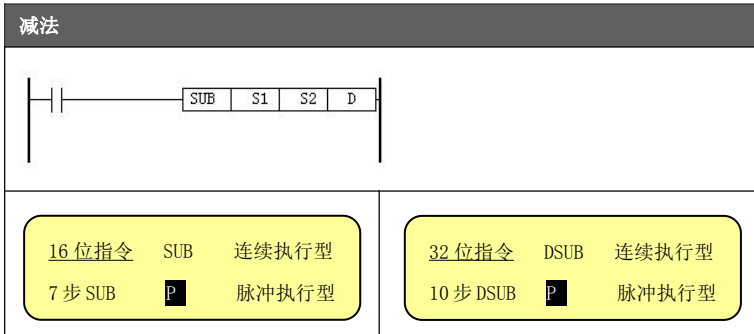


编程注意事项

- 如运算结果向上溢出，即超过 32767（16 位运算）或 2147483647（32 位运算）时，其结果保持为 32767 或 2147483647。
- 如运算结果向下溢出，即超过-32768（16 位运算）或-2147483648（32 位运算）时，其结果保持为-32768 或-2147483648。
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议例如将软元件指定为偶数编号
- 可以将源和目标指定为相同的软元件编号。这种情况下，如使用连续执行型指令，则每个扫描周期的加法运算结果都会变化，请务必注意。

5.4.19 减法[SUB]

使两个数据或软元件内容进行二进制减法运算并将结果存储到其他软元件中。



操作数	作用	类型
S1	指定进行减法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行减法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存减法运算结果的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●

功能与动作

两个源数据进行二进制减法后传送到目标处。各数据的最高位是正(0)、负(1)的符号位，这些数据以代数形式进行减法运算。

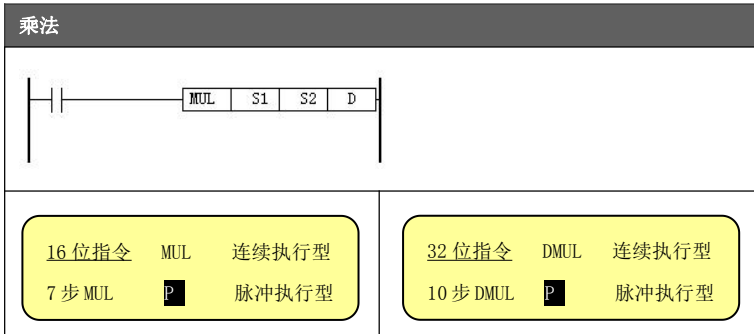


编程注意事项

- 如运算结果向上溢出，即超过 32767（16 位运算）或 2147483647（32 位运算）时，其结果保持为 32767 或 2147483647。
- 如运算结果向下溢出，即超过-32768（16 位运算）或-2147483648（32 位运算）时，其结果保持为-32768 或-2147483648。
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议例如将软元件指定为偶数编号
- 可以将源和目标指定为相同的软元件编号。这种情况下，如使用连续执行型指令，则每个扫描周期的减法运算结果都会变化，请务必注意。

5.4.20 乘法[MUL]

使两个数据或软元件内容进行二进制乘法运算并将结果存储到其他软元件中。

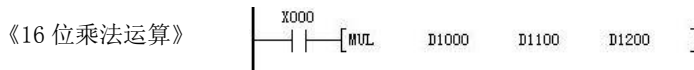


操作数	作用	类型
S1	指定进行乘法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行乘法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存乘法运算结果的软元件地址编号	32/64 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
S1	●	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●	●
S2	●	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●	●
D	●	●	●	●	●	●	●	●	●	●
	●	●	●	●	●	●	●	●	●	●

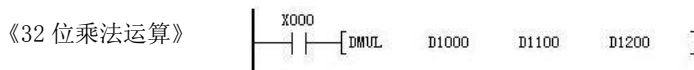
功能与动作

S1 指定软元件的内容是被乘数，S2 指定软元件的内容是乘数，D 指定软元件和其下一个编号的软元件将存入结果的低半部分和高半部分。



$$(D1000) \times (D1100) \rightarrow (D1201, D1200)$$

$$16 \text{ 位} \times 16 \text{ 位} \rightarrow 32 \text{ 位}$$



$$(D1001, D1000) \times (D1101, D1100) \rightarrow (D1203, D1202, D1201, D1200)$$

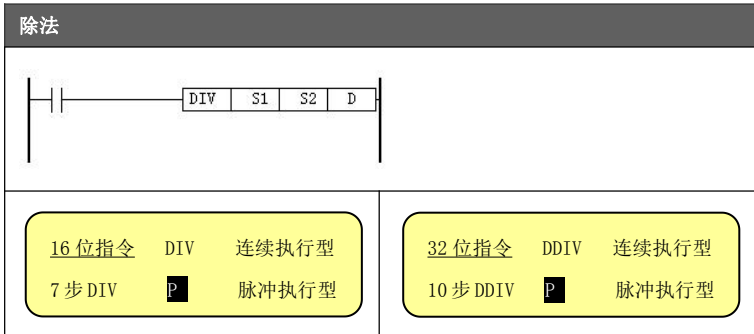
$$32 \text{ 位} \times 32 \text{ 位} \rightarrow 64 \text{ 位}$$

编程注意事项

- 存放结果不要指定软元件的最后一个编号（如 D7999），可能导致高位运算结果丢失。建议指定偶数编号的软元件地址。
- D 是位组合元件时（如 K4M100），可以进行 K1~K8 的位指定。
 - ◆ 16 位运算，指定为 K4 时，只能求得乘积运算的低 16 位。
 - ◆ 32 位运算，目标地址使用位软元件时，只能得到低 32 位的结果，不能得到高 32 位的结果，请向字元件传送后使用字元件进行运算。
- 由于无法监控到 64 位字元件，所以使用 32 位乘法指令时，若存在运算结果超出 32 位的情况，建议使用浮点运算。

5.4.21 除法[DIV]

使两个数据或软元件内容进行二进制除法运算并将结果存储到其他软元件中。

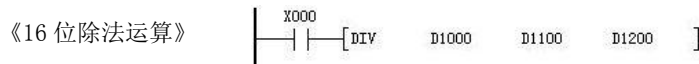


操作数	作用	类型
S1	指定进行除法运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行除法运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存除法运算结果的软元件地址编号	16/32 位 BIN

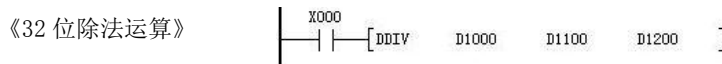
操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●

功能与动作

S1 指定软元件的内容是被除数，S2 指定软元件的内容是除数，D 指定软元件和其下一个编号的软元件将存入商和余数。



被除数 除数 商 余数
 (D1000) ÷ (D1100) → (D1200) …… (D1201)
 16 位 ÷ 16 位 → 16 位 16 位



被除数 除数 商 余数
 (D1001, D1000) ÷ (D1101, D1100) → (D1201, D1200) …… (D1203, D1202)
 32 位 ÷ 32 位 → 32 位 32 位

**编程注意事
项**

- 存放结果不要指定软元件的最后一个编号（如 D7999），可能导致余数结果丢失。建议指定偶数编号的软元件地址。
- D 是位组合元件时（如 K4M100），无法得到余数。
- 商和余数的最高位为正 (0)、负 (1) 的符号位。当被除数或除数中的一方为负数时，商则为负，当被除数为负时余数则为负。
- 除数 S2=0 时：
 - ◆ 被除数 ≥ 0 ，则结果为代数范围最大限值（比如 16 位指令其结果为 K32767）。
 - ◆ 被除数 < 0 ，则结果为代数范围最小限值（比如 16 位指令其结果为 K-32768）。

5.4.22 加1[INC],减1[DEC]

将指定软元件的数据进行自加1/自减1运算。

加 1	
16 位指令 INC 连续执行型 3 步 INC P 脉冲执行型	32 位指令 DINC 连续执行型 4 步 DINC P 脉冲执行型
减 1	
16 位指令 DEC 连续执行型 3 步 DEC P 脉冲执行型	32 位指令 DDEC 连续执行型 4 步 DDEC P 脉冲执行型

操作数	作用	类型
D	指定自加 1/自减 1 的软元件地址编号	16/32 位 BIN

操作数	字型软元件										
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
D		●	●	●	●		●	●	●	●	

功能与动作

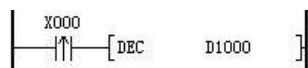
源软元件地址内容自加 1/自减 1 后，其运算结果仍然存放在此地址。

《自加 1 运算》



- X000 每置 ON 一次，D1000 的内容就加 1。
- 16 位运算时，如果当前 D1000=32767，则结果保持不再自加 1。
- 32 位运算时，如果当前 D1000=2147483647，则结果保持不再自加 1。

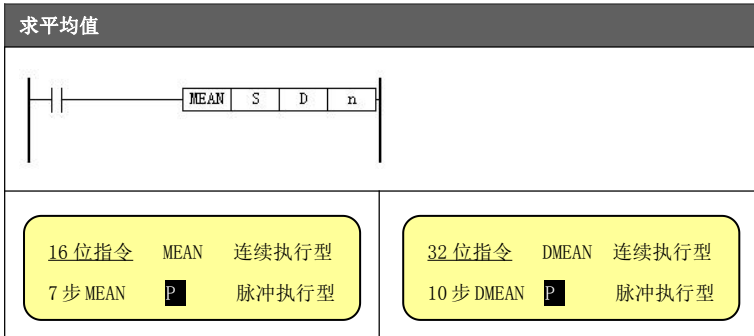
《自减 1 运算》



- X000 每置 ON 一次，D1000 的内容就减 1。
- 16 位运算时，如果当前 D1000=-32768，则结果保持不再自减 1。
- 32 位运算时，如果当前 D1000=-2147483648，则结果保持不再自减 1。

5.4.23 求平均值[MEAN]

将指定数据或软元件进行求平均值的运算。

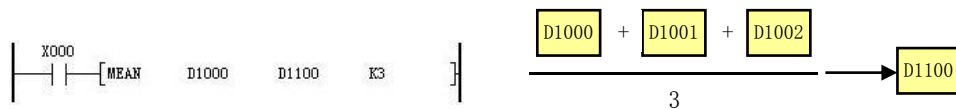


操作数	作用	类型
S	指定源数据的软元件首地址编号	16/32 位 BIN
D	指定存储平均运算结果的软元件地址编号	16/32 位 BIN
n	指定源数据个数的数值	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

- 将 n 点的源数据的平均值（代数和被 n 除）存入目标地址中，余数将舍去。
- 取 n 值时需注意：
S 取 n 后的地址编号不要超过该软元件的地址范围。
对 16 位指令而言，n 取值范围为 1~64。
对 32 位指令而言，n 取值范围为 1~64。



5.4.24 逻辑与[WAND], 逻辑或[WOR], 逻辑异或[WXOR]

将指定数据或软元件的各位进行逻辑与/逻辑或/逻辑异或运算。

逻辑与	
<p>16 位指令 WAND 连续执行型</p> <p>7 步 WAND P 脉冲执行型</p>	<p>32 位指令 DWAND 连续执行型</p> <p>10 步 DWAND P 脉冲执行型</p>
逻辑或	
<p>16 位指令 WOR 连续执行型</p> <p>7 步 WOR P 脉冲执行型</p>	<p>32 位指令 DWOR 连续执行型</p> <p>10 步 DWOR P 脉冲执行型</p>
逻辑异或	
<p>16 位指令 WXOR 连续执行型</p> <p>7 步 WXOR P 脉冲执行型</p>	<p>32 位指令 DXWOR 连续执行型</p> <p>10 步 DXWOR P 脉冲执行型</p>

操作数	作用	类型
S1	指定进行运算的数据或软元件地址编号	16/32 位 BIN
S2	指定进行运算的数据或软元件地址编号	16/32 位 BIN
D	指定保存运算结果的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●

功能与动作

将 2 个指定的数据或软件内容按位进行与/或/异或操作后，结果进行保存。

《逻辑与运算》

位状态	与 (&)					
操作位	0	0	1	1	0	1
操作位	1	1	0	0	0	1
结果位	0	0	0	0	0	1



- 假设 D1000=1234, D1100=5678, 当 X000 触点闭合时, 进行逻辑与运算。
- D1000 二进制 = 0000 0100 1101 0010,
D1100 二进制 = 0001 0110 0010 1110, 按照上表进行计算。
D1200 二进制 = 0000 0100 0000 0010, 十进制表示为 1026。

《逻辑或运算》

位状态	或 ()					
操作位	0	0	1	1	0	1
操作位	1	1	0	0	0	1
结果位	1	1	1	1	0	1



- 假设 D1000=1234, D1100=5678, 当 X000 触点闭合时, 进行逻辑或运算。
- D1000 二进制 = 0000 0100 1101 0010,
D1100 二进制 = 0001 0110 0010 1110, 按照上表进行计算。
D1200 二进制 = 0001 0110 1111 1110, 十进制表示为 5886。

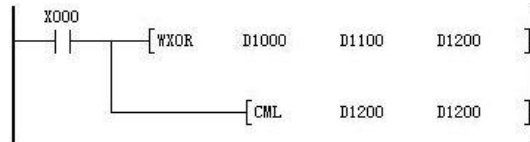
《逻辑异或运算》

位状态	异或(^)					
操作位	0	0	1	1	0	1
操作位	1	1	0	0	0	1
结果位	1	1	1	1	0	0



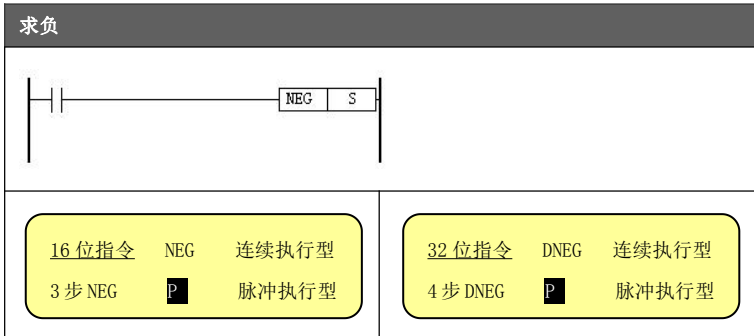
- 假设 D1000=1234, D1100=5678, 当 X000 触点闭合时, 进行逻辑异或运算。
- D1000 二进制 = 0000 0100 1101 0010,
D1100 二进制 = 0001 0110 0010 1110, 按照上表进行计算。
D1200 二进制 = 0001 0010 1111 1100, 十进制表示为 4860。

将逻辑异或指令(WXOR)与取反传送指令(CML)结合使用, 可实现逻辑异或非运算。



5.4.25 求负[NEG]

将指定软元件的内容按位取反后，再加 1 进行保存。



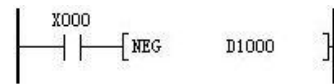
操作数	作用	类型
D	指定源数据的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D		●	●	●	●		●	●	●	●

功能与动作

- 将指定软元件的内容中各位先取反（0→1，1→0），然后再加 1，将其结果再存入原先的软元件中。
- 使用连续执行型指令则在每一个扫描周期执行该运算指令。

《求负运算》



- 假设 D1000=1234，当 X000 触点闭合时，进行求负运算。
- D1000 二进制 = 0000 0100 1101 0010，进行取反操作。
D 取反二进制 = 1111 1011 0010 1101，十进制表示为-1235。
- 将上步结果加 1 送入 D1000，D1000=-1234。

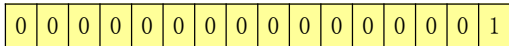
可使用减法指令(SUB)达到上述示例的效果，如下所示：



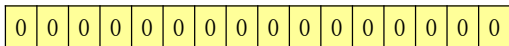
负数表示和绝对值

- 在 PLC 中，负数以 2 的补码形式进行表示。
- 最高位为 1 时表示负数，可使用补码指令求得绝对值。

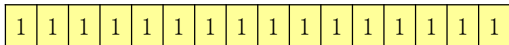
(D1000)=1



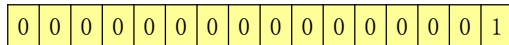
(D1000)=0



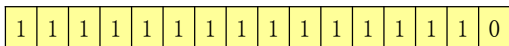
(D1000)=-1



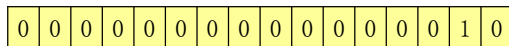
$\overline{(D1000)}+1=1$



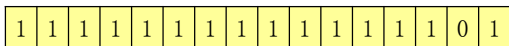
(D1000)=-2



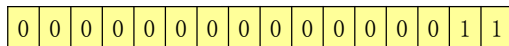
$\overline{(D1000)}+1=2$



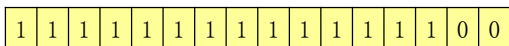
(D1000)=-3



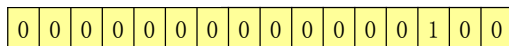
$\overline{(D1000)}+1=3$



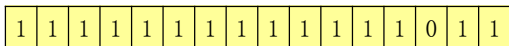
(D1000)=-4



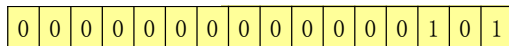
$\overline{(D1000)}+1=4$



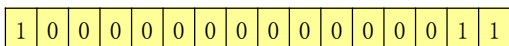
(D1000)=-5



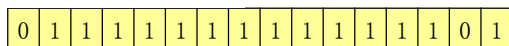
$\overline{(D1000)}+1=5$



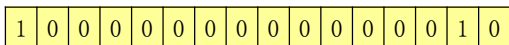
(D1000)=-32765



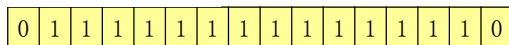
$\overline{(D1000)}+1=32765$



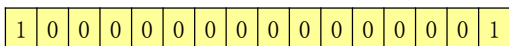
(D1000)=-32766



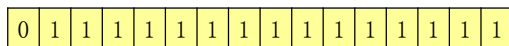
$\overline{(D1000)}+1=32766$



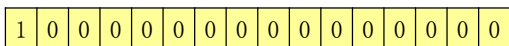
(D1000)=-32767



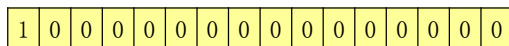
$\overline{(D1000)}+1=32767$



(D1000)=-32768



$\overline{(D1000)}+1=-32768$



绝对值的最大范围只能达到 32767

5.4.26 开方[SQR]

对指定数据或软元件中内容进行开方运算。

开方

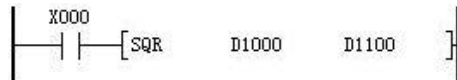
16 位指令	SQR	连续执行型	32 位指令	DSQR	连续执行型
5 步 SQR	P	脉冲执行型	7 步 DSQR	P	脉冲执行型

操作数	作用	类型
S	指定进行开方运算的数据或软元件地址编号	16/32 位 BIN
D	指定存储运算结果的软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

- 进行平方根开方运算。结果保留整数位，舍弃小数位。
- 若开方数是负数，结果为 0。



- X000 触点闭合，进行开平方根的运算。
- D1000=10000，结果 D1100=100，开方无余数。
- D1000=88，结果 D1100=9，开方余数为 7(舍弃)。
- D1000=-100，结果为 0。

5.4.27 循环左移[ROL], 循环右移[ROR]

使指定软元件进行各位数据的循环左移/循环右移。

循环左移

16 位指令 ROL 连续执行型 5 步 ROL P 脉冲执行型	32 位指令 DROL 连续执行型 7 步 DROL P 脉冲执行型
--	--

循环右移

16 位指令 ROR 连续执行型 5 步 ROR P 脉冲执行型	32 位指令 DROR 连续执行型 7 步 DROR P 脉冲执行型
--	--

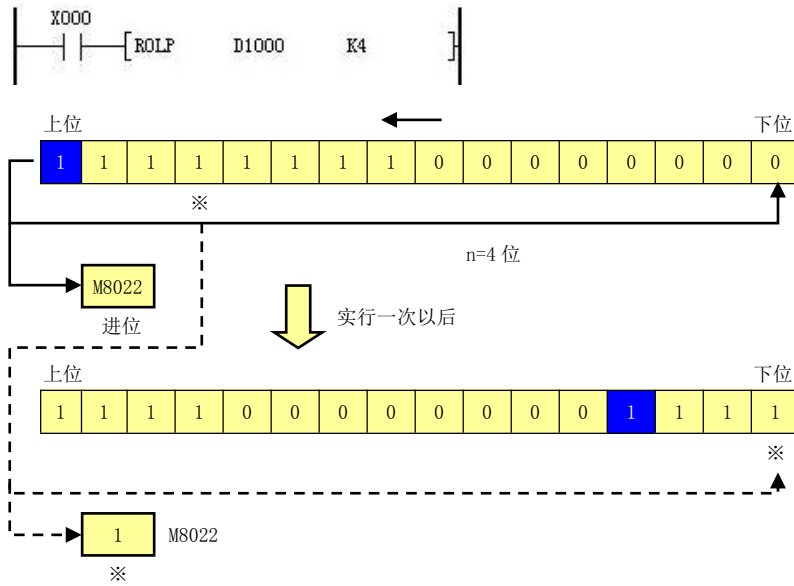
操作数	作用	类型
D	指定源数据的软元件地址编号	16/32 位 BIN
n	指定循环左移/右移的位数	16/32 位 BIN

操作数	字型软元件									
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●								

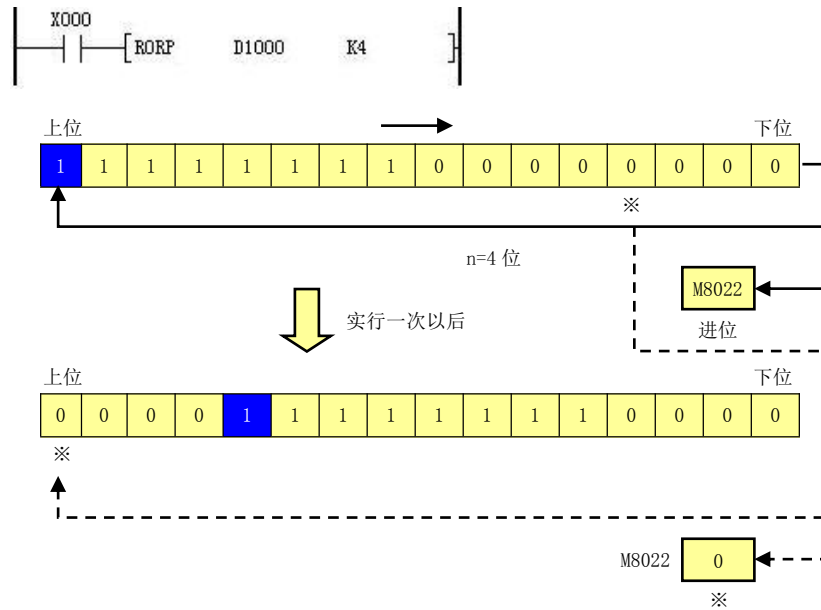
功能与动作

对指定软元件的内容进行 n 位的循环左移/右移操作。最终位被存入进位标志 M8022 中。

《循环左移》



《循环右移》



编程注意事项

- 在指定位组合元件进行移位时，常数修饰项必须指定 K4(16 位指令)或 K8(32 位指令)，例如 K4S100, K8M64。否则运行将出错。
- 16 位指令时指定 $1 \leq n \leq 16$, 32 位指令时指定 $1 \leq n \leq 32$, 否则运行将出错。
- 连续执行型指令每个扫描周期都会进行回转动作，请务必注意。

5.4.28 带进位循环左移[RCL]，带进位循环右移[RCR]

使指定软元件进行各位信息的带进位循环左移/循环右移。

带进位循环左移

16 位指令 RCL 连续执行型 5 步 RCL P 脉冲执行型	32 位指令 DRCL 连续执行型 7 步 DRCL P 脉冲执行型
---	---

带进位循环右移

16 位指令 RCR 连续执行型 5 步 RCR P 脉冲执行型	32 位指令 DRCR 连续执行型 7 步 DRCR P 脉冲执行型
---	---

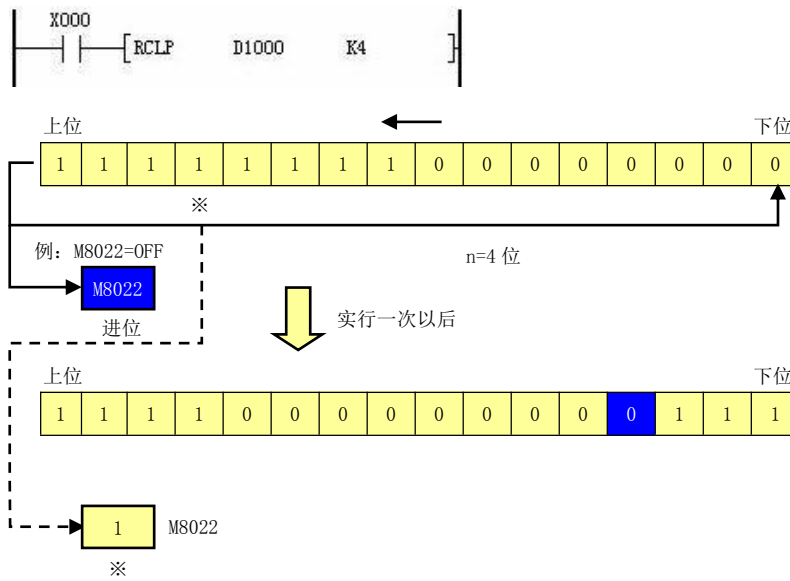
操作数	作用	类型
D	指定源数据的软元件地址编号	16/32 位 BIN
n	指定循环左移/右移的位数	16/32 位 BIN

操作数	字型软元件									
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●								

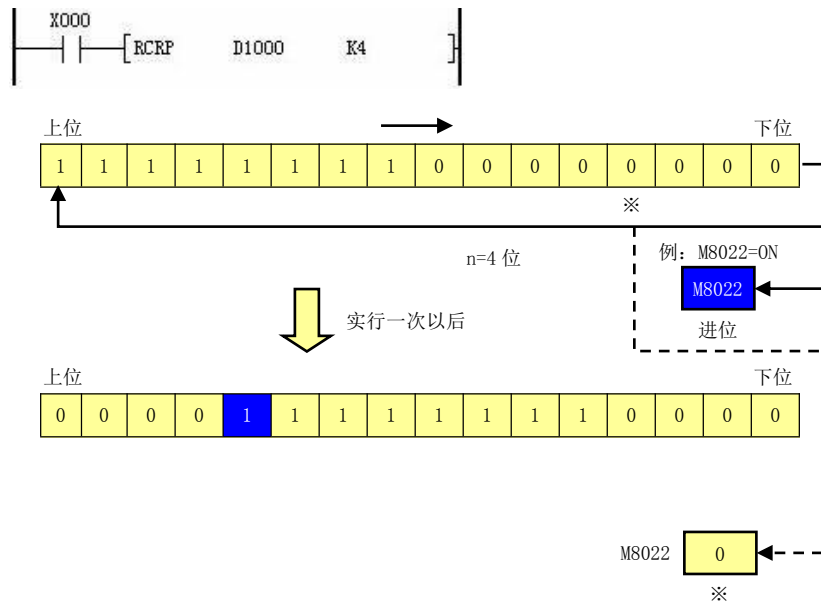
功能与动作

对指定软元件的内容进行 n 位的带进位循环左移/右移操作。起始最高位由进位标志 M8022 填充，最终位被存入进位标志 M8022 中。

《带进位循环左移》



《带进位循环右移》



编程注意事
项

- 在指定位组合元件进行移位时,常数修饰项必须指定K4(16位指令)或K8(32位指令),例如K4S100,K8M64。否则运行将出错。
- 16位指令时指定 $1 \leq n < 16$,32位指令时指定 $1 \leq n < 32$,否则运行将出错。
- 执行带进位循环指令前预先设置M8022状态,可以将其送入目标地址。
- 连续执行型指令每个扫描周期都会进行回转动作,请务必注意。

5.4.29 位左移[SFTL], 位右移[SFTR]

将指定软元件中的数据进行位左移/位右移操作。

位左移

16 位指令 SFTL 连续执行型

9 步 SFTL **P** 脉冲执行型

32 位指令

无

位右移

16 位指令 SFTR 连续执行型

9 步 SFTR **P** 脉冲执行型

32 位指令

无

操作数	作用	类型
S	指定源数据的软元件首地址编号	位
D	指定目标软元件首地址编号	位
n1	指定目标软元件的个数	16/32 位 BIN
n2	指定位左移/位右移每次移动的个数	16/32 位 BIN

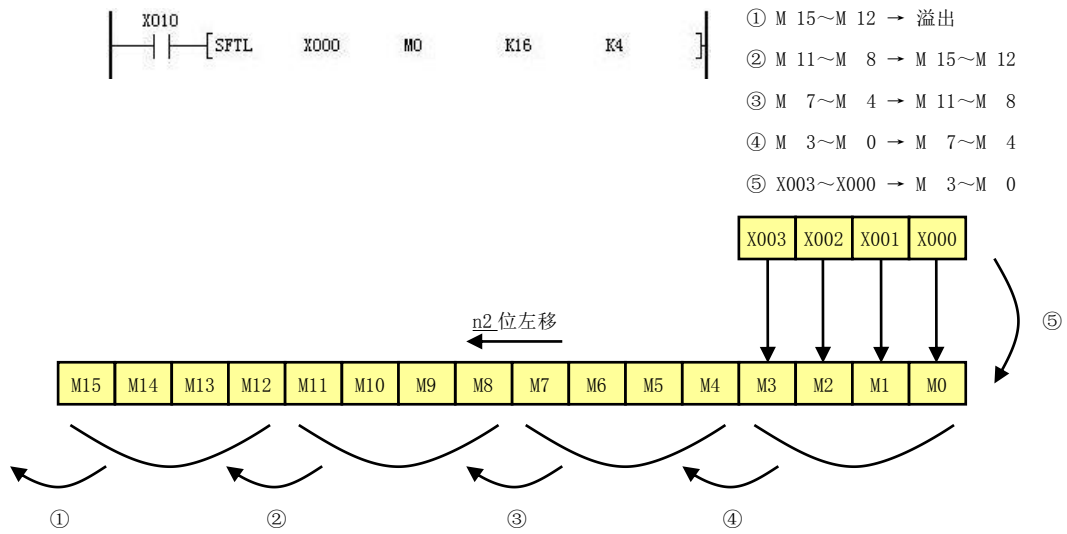
操作数	位型软元件			
S	X	Y	M	S
	●	●	●	●
D	X	Y	M	S
		●	●	●

操作数	字型软元件									
n1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
n2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

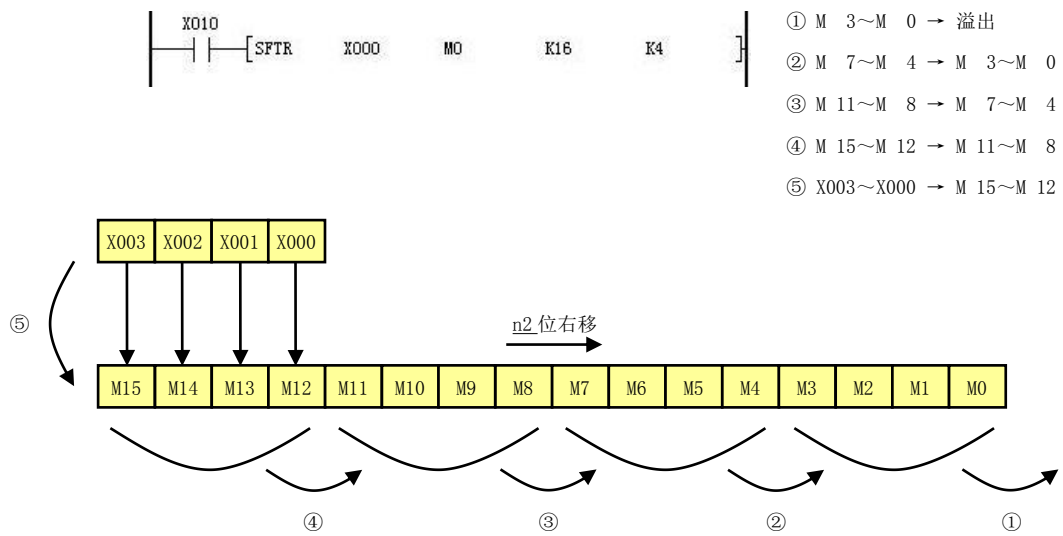
动作与功能

- 对于 n1 位（移动软元件的长度）的位元件进行 n2 的位移动指令（指令执行时执行 n2 位的移位）。
- n2 为 K1 时，每执行一次移位指令，目标软元件移动 1 位。
- 设置 $1 \leq n2 < n1 \leq 1024$ 是有效的。
- 连续执行型指令每个扫描周期都会进行位移动操作，请务必注意。

《位左移》



《位右移》



5.4.30 字左移[WSFL], 字右移[WSFR]

将指定软元件中的数据进行字左移/字右移操作。

字左移

16 位指令 WSFL 连续执行型

9步 WSFL **P** 脉冲执行型

32 位指令

无

字右移

16 位指令 WSFR 连续执行型

9步 WSFR **P** 脉冲执行型

32 位指令

无

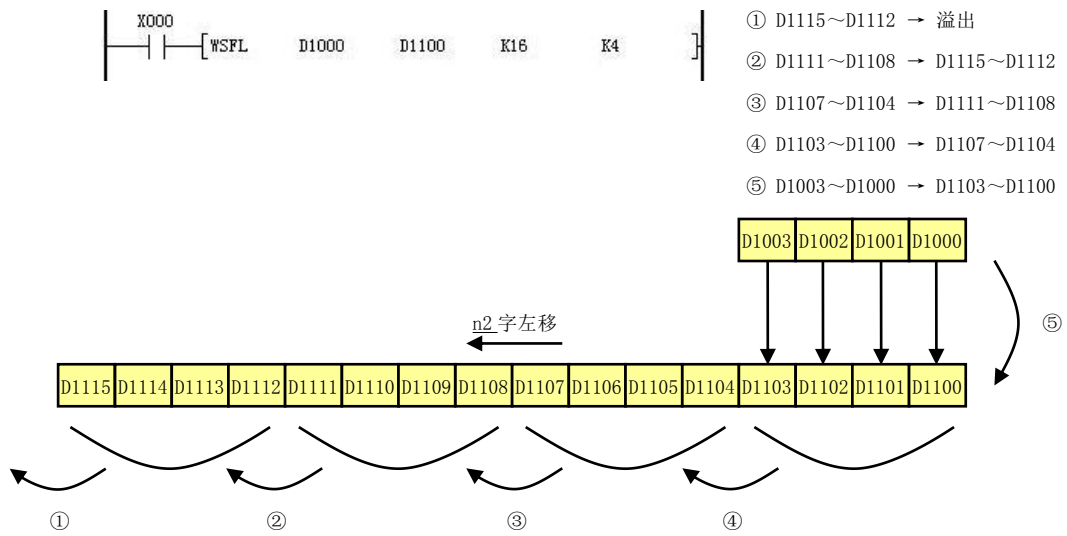
操作数	作用	类型
S	指定源数据的软元件首地址编号	16/32 位 BIN
D	指定目标软元件首地址编号	16/32 位 BIN
n1	指定目标软元件的个数	16/32 位 BIN
n2	指定字左移/字右移每次移动的个数	16/32 位 BIN

操作数	字型软元件										
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●	●	●	●	●	●	●		
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●	●	●		●	●	●		
n1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●									
n2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●									

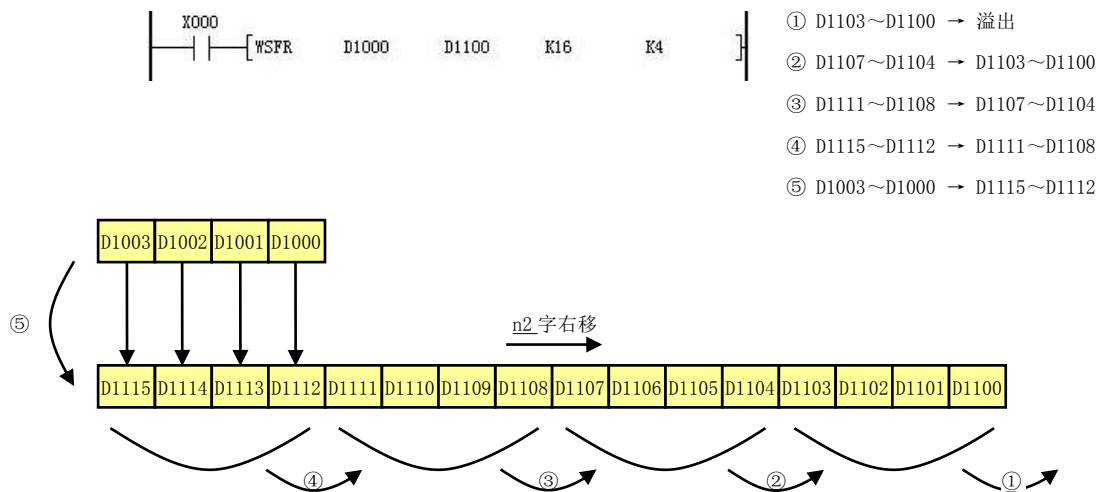
动作与功能

- 对于 n1 字（移动软元件的长度）的字元件进行 n2 的字移动指令（指令执行时执行 n2 字的移字）。
- n2 为 1 时，每执行一次移字指令，目标软元件移动 1 字。
- 对于 D, C, T 字软元件，设置 $1 \leq n2 \leq n1 \leq 512$ 是有效的。
对于位组合元件，设置 $1 \leq n2 \leq n1 \leq 64$ 是有效的。
- 连续执行型指令每个扫描周期都会进行字移动操作，请务必注意。

《字左移》

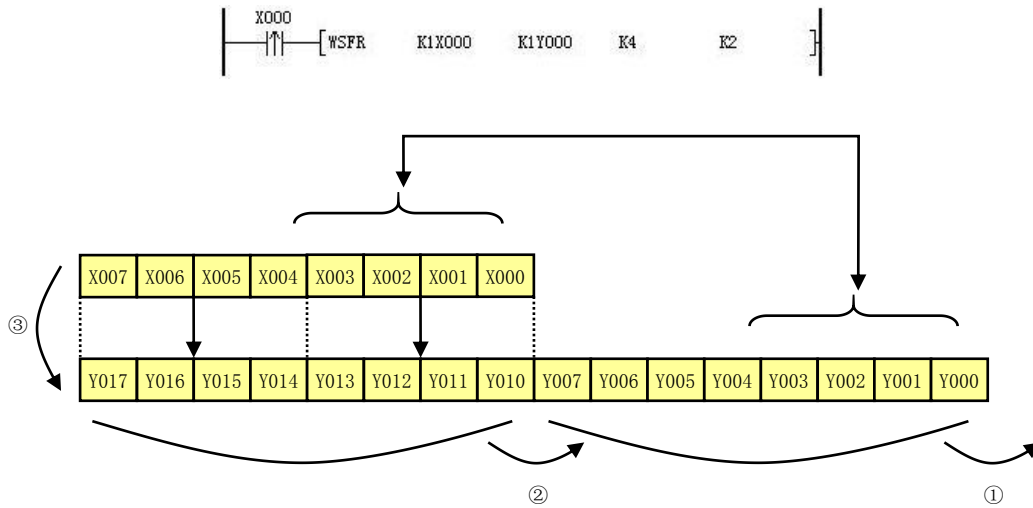


《字右移》



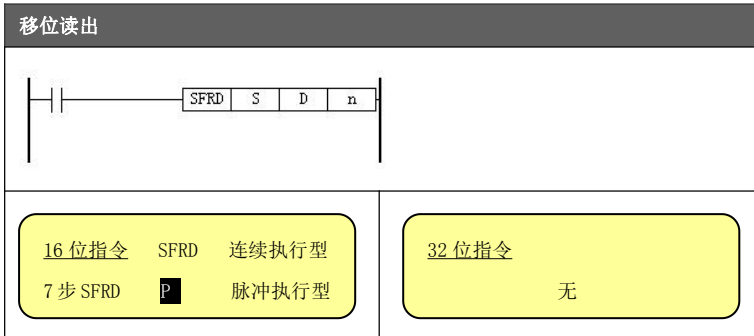
位组合元件
移字

- 当源操作数为字软元件，目的操作数为位组合软元件，且常数修饰项 <4 时(比如K2M100)，系统将位组合补足为16位的字，然后参与移位。
位组合软元件的常数修饰项配置为一致，比如S配置为K4M0，D配置为K4S0。
- 位组合软元件其移动位数总和必须 ≤ 1024 ，否则运行将错误。
 - ◆ S为K4M0，n1为4，移动总位数 $=4 \times 4 \times 4=64$ ，运行正常。
 - ◆ S为K4M0，n1为64，移动总位数 $=4 \times 4 \times 64=1024$ ，运行正常。
 - ◆ S为K4M0，n1为65，移动总位数 $=4 \times 4 \times 65=1040$ ，运行出错。



5.4.31 移位读出[SFRD]

为控制先入先出数据的读出指令。

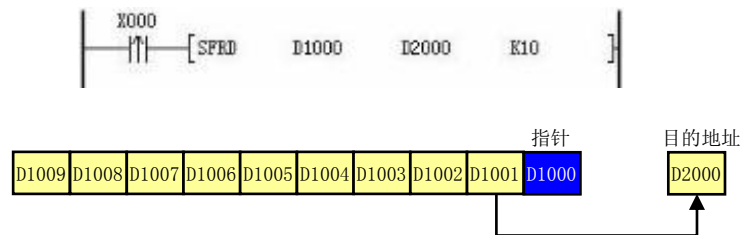


操作数	作用	类型
S	指定移位指针的软元件地址编号	16/32 位 BIN
D	指定读出的软元件地址编号	16/32 位 BIN
n	指定移位读出的点数	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

动作与功能

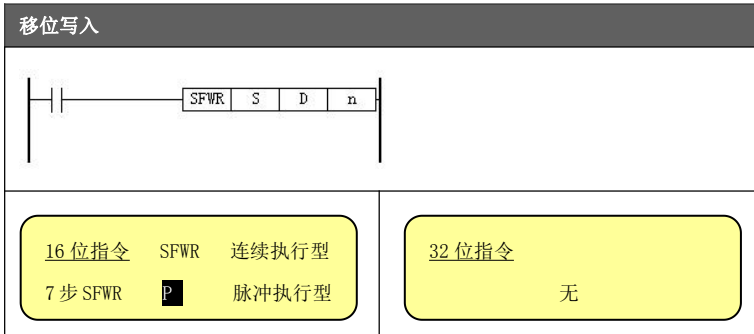
- 对于字元件， $2 \leq n \leq 512$ 是有效的；对于位组合元件， $2 \leq n \leq 64$ 是有效的。
- S 首地址内容超过 n 时，不能执行移位操作，M8022 动作。
- 不足 1 个字的位组合移位，S 和 D 必须同时指定位组合配置，且 K 值相同。
 - ◆ SFRD K1M0 D2000 K10，不合法。
 - ◆ SFRD K2M0 K3S0 K10，不合法。
 - ◆ SFRD K3M0 K3S0 K10，合法。



- 当 X000 从 OFF→ON 变化时，D 首地址编号后面的 1 个软元件地址数值被传送到目的地址，堆栈区当前指针数值个数个地址前移 1 个位置，指针数值自减 1。
 比如指令执行时 D1000=8，那么将 D1001 移动到 D2000，D1009 和 D1008 不变，D1008~D1002 送到 D1007~D1001，然后 D1000 内容变为 7。
- 指针 D1000 的内容为 0 时，则不处理。
- 配置 n 值和 S 首地址时需要考虑在执行过程中不能超出 S 软元件所支持的地址范围。
- 连续执行型指令将在每个扫描周期执行移动。

5.4.32 移位写入[SFWR]

为控制先进先出数据的写入指令。

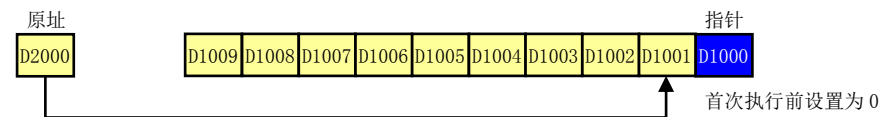


操作数	作用	类型
S	指定写入的软件元件地址编号	16/32 位 BIN
D	指定移位指针的软件元件地址编号	16/32 位 BIN
n	指定移位写入的点数	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

动作与功能

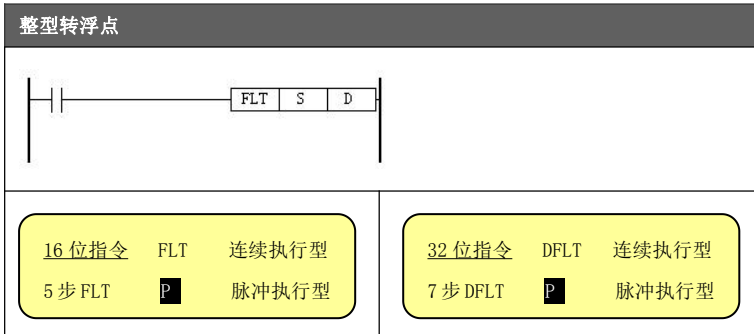
- 对于字元件， $2 \leq n \leq 512$ 是有效的；对于位组合元件， $2 \leq n \leq 64$ 是有效的。
- D 首地址如超过 n，指令不处理，进位标志 M8022 置位。
- 不足 1 个字的位组合移位，S 和 D 必须同时指定位组合配置，且 K 值相同。
 - ◆ SFWR K1M0 D2000 K10，不合法。
 - ◆ SFWR K2M0 K3S0 K10，不合法。
 - ◆ SFWR K3M0 K3S0 K10，合法。



- 首次运行前预先将指针内容 D1000 清零。
- 当 X000 从 OFF→ON 变化时，D2000 的内容被存入 D1001，指针 D1000 的内容变为 1。X000 再次从 OFF→ON 时，D1001 的数值传到 D1002，D2000 的内容被存入 D1001，指针 D1000 的内容变为 2。X000 再次从 OFF→ON 时，D1002 的数值传到 D1003，D1001 的数值传到 D1002，D2000 的内容被存入 D1001，指针 D1000 的内容变为 3。以此类推直到 D1000 变成 10 后，结束整个移位写入过程。
- 配置 n 值和 D 首地址时需要考虑在执行过程中不能超出 D 软元件所支持的地址范围。
- 连续执行型指令将在每个扫描周期执行移动。

5.4.33 整型转浮点[FLT]

将指定数据或软元件内容转换为 32 位二进制浮点数。



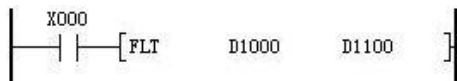
操作数	作用	类型
S	指定源数据或软元件地址编号	16/32 位 BIN
D	指定目标软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

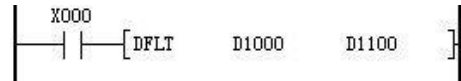
- 二进制整数与二进制浮点值间的转换指令。
- 常数 K/H 在各浮点运算中由系统自动转换，无需 FLT 转成浮点型。
- FLT 的逆变换指令为 INT。

《16 位指令》



(D1000) → (D1101, D1100)
BIN 整数 2 进制浮点值

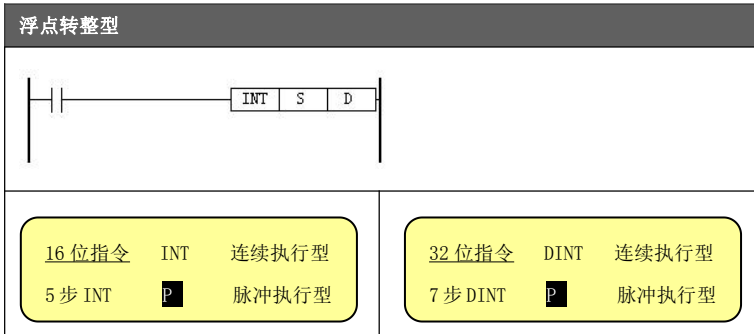
《32 位指令》



(D1001, D1000) → (D1101, D1100)
BIN 整数 2 进制浮点值

5.4.34 浮点转整型[INT]

将指定软元件内容(必须为 32 位二进制浮点数)转换为 2 进制整数。



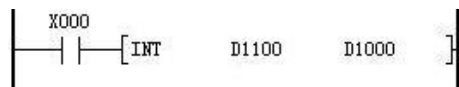
操作数	作用	类型
S	指定源数据或软元件地址编号	32 位二进制浮点
D	指定目标软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

- 32 位二进制浮点值与二进制整数值间的转换指令。
- 转换后结果将舍去小数部分，在转换之前加上 0.5，可实现四舍五入功能。
- 运算结果限值条件：
 - ◆ 16 位指令时：-32768~32767。
 - ◆ 32 位指令时：-2147483648~2147483647。
 - ◆ 源数据超出 32 位二进制浮点范围时：-2147483648。
- INT 的逆变换指令为 FLT。

《16 位指令》



(D1101, D1100) → (D1000)
2 进制浮点值 BIN 整数(小数位舍去)

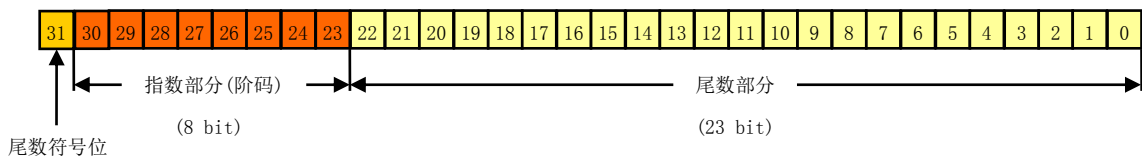
《32 位指令》



(D1101, D1100) → (D1001, D1000)
2 进制浮点值 BIN 整数(小数位舍去)

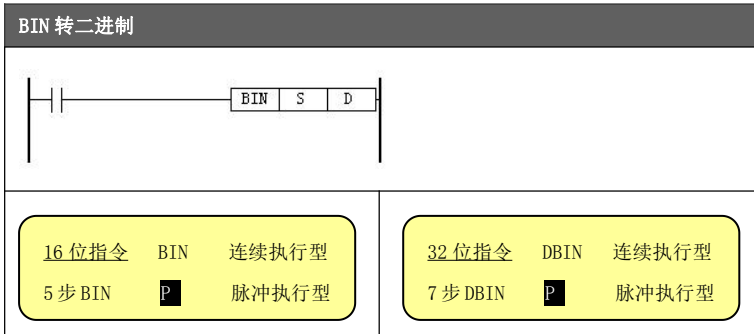
**单精度浮点
数格式**

- 用来表示带有小数部分的实数，一般用于科学计算。
- 占用 4 个字节（32 位）存储空间，包括符号位 1 位，阶码 8 位，尾数 23 位。
- 其数值范围为-3.4E038~3.4E038，单精度浮点数最多有 7 位十进制有效数字，指数用“E”表示。



5.4.35 BCD 转二进制[BIN]

将指定软元件内容(必须为 BCD 码格式)转换为二进制数。

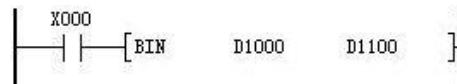


操作数	作用	类型
S	指定源数据的软元件地址编号	BCD
D	指定目标软元件地址编号	16/32 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●

功能与动作

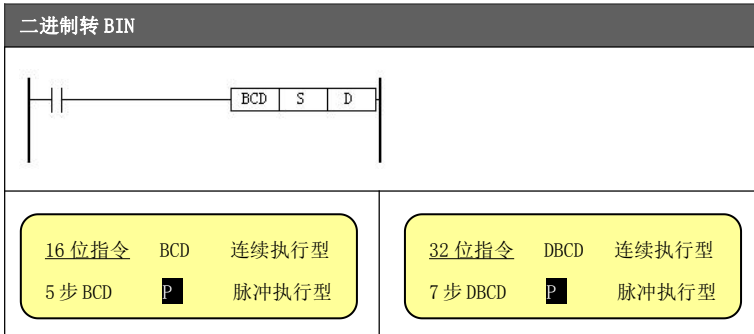
- S 的内容必须为 BCD 码格式，否则不能得到正确的结果。BCD 是用 4 位二进制数来表示 1 位十进制数中的 0~9 这 10 个数的方法。
- 为简便理解，可以视作将十六进制数值(不能有 A, B, C, D, E, F)直接拿出来变为十进制数值(如 H1234 转换后变为 K1234)。
- 由于常数 K/H 项被系统自动转换，所以不能成为适用该指令的软元件。



- X000 触点导通，执行 BCD 码转二进制整数值的指令。
- 若 D1000=4660，因为 4660 其十六进制表为 H1234，所以转换后结果 D1100=1234。

5.4.36 二进制转 BCD[BCD]

将指定软元件内容二进制数转换为 BCD 码格式。

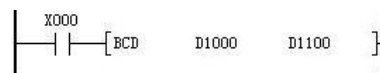


操作数	作用	类型
S	指定源数据的软元件地址编号	16/32 位 BIN
D	指定目标软元件地址编号	BCD

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●	●	●	●	●	●
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●		●	●	●	●

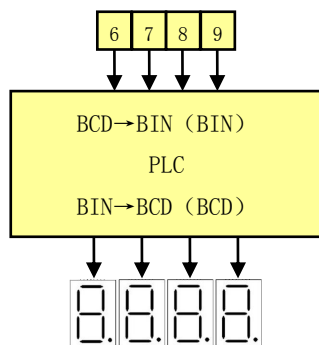
功能与动作

- BCD 是用 4 位二进制数来表示 1 位十进制数中的 0~9 这 10 个数的方法。
- 为简便理解，可以视作将十进制数值拿出来直接变为十六进制数值(如 K1234 转换后变为 H1234)。
- 使用 16 位指令时，S 指定范围为 0~9999。小于 0 时，转换结果为 0；大于 9999 时，转换结果为 H9999。
- 使用 32 位指令时，S 指定范围为 0~99999999。小于 0 时，转换结果为 0；大于 99999999 时，转换结果为 H99999999。
- 将可编程控制器内的二进制数据变为七段显示器用的 BCD 码而向外部输出时使用。



- X000 触点导通，执行二进制整数转 BCD 码的指令。
- 若 D1000=1234，因为 H1234 其十进制表达为 4660，所以转换后结果 D1100=4660。

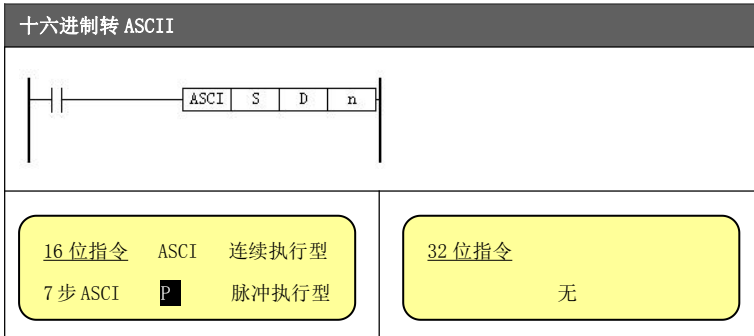
BCD/BIN 输入输出



- 四则运算 (+ - × ÷) 与增量指令、减量指令等 PLC 内的运算都用 BIN 码进行。
- 因此 PLC 获取 BCD 的数字开关信息时要使用 (BCD→BIN) 转换传送指令。
- 另外向 BCD 的七段显示器输出时请使用 (BIN→BCD) 转换传送指令。

5.4.37 十六进制转 ASCII [ASCII]

将指定软元件中的内容转换为 ASCII 码。



操作数	作用	类型
S	指定源数据的软元件地址编号	2 位 HEX
D	指定目标软元件的首地址编号	ASCII
n	指定转换的 ASCII 码字符个数	16 位 BIN

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●						
D(占 n 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●	●						
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

- S 的 HEX 数据的各位转换成 ASCII 码，向 D 的低 8 位进行传送。转换的字符数用 n 指定。
- D 的高 8 位自动变 0，低 8 位存储一个 ASCII 数据。指令执行占用从 D 编号开始后续 n 个寄存器个数。
- 设置 $1 \leq n \leq 256$ 。

ASCII 码转换表(均为十六进制表示):

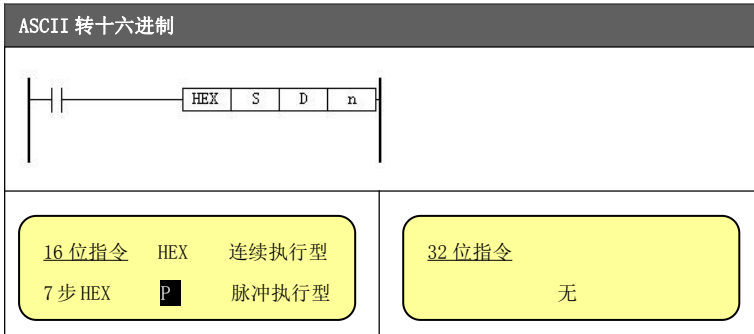
数值	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	HA	HB	HC	HD	HE	HF
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	H41	H42	H43	H44	H45	H46



- X000 触点闭合，执行十六进制转 ASCII 指令。
- 若 D1000=H1234，执行结果：
D1100=H0031，D1101=H0032
D1102=H0033，D1103=H0034

5.4.38 ASCII 转十六进制[HEX]

将指定软元件中的内容(必须为 ASCII 码格式)转换为十六进制数。

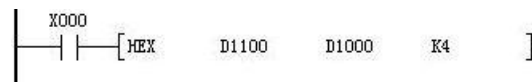


操作数	作用	类型
S	指定源数据的软元件地址编号	ASCII
D	指定目标软元件的首地址编号	2 位 HEX
n	指定转换的 ASCII 码字符个数	16 位 BIN

操作数	字型软元件									
S(占 n 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●					
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●	●					
n	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

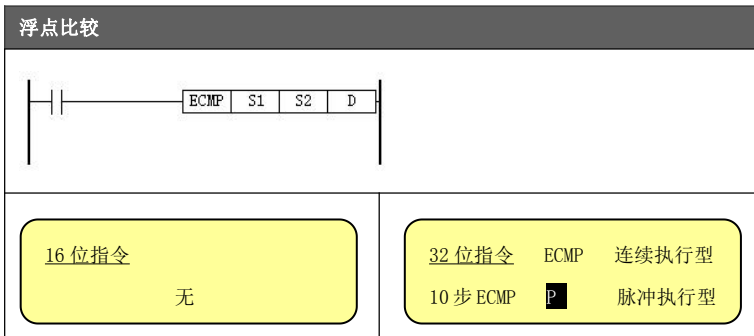
- S 的低 8 位作为 ASCII 码转成 HEX 数据，每 4 位向 D 进行传送。转换的字符数用 n 指定。
- S 内容必须设置高 8 位为 0，低 8 位为上述 ASCII 码转换表的 ASCII 码值（参考 5.4.37 章节），否则转换结果不正确。
- 设置 $1 \leq n \leq 256$ 。



- X000 触点闭合，执行 ASCII 转十六进制指令。
- 若 D1100=H0031，D1101=H0032
D1102=H0033，D1103=H0034
执行结果：D1000=H1234

5.4.39 浮点比较[ECMP]

将指定的两个浮点数进行大小比较，并输出比较结果。



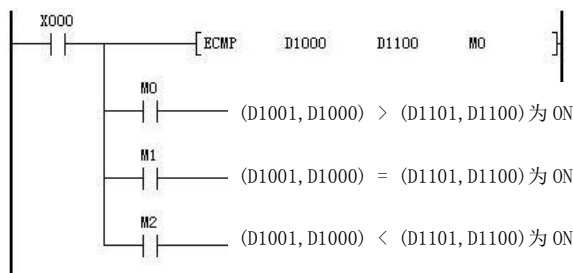
操作数	作用	类型
S1	指定被比较的数据或软元件地址编号	32 位二进制浮点
S2	指定比较源的数据或软元件地址编号	32 位二进制浮点
D	指定比较输出结果的软元件地址编号	位

操作数	字型软元件										
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								

操作数	位型软元件			
D(占 3 点)	X	Y	M	S
			●	●

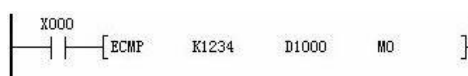
功能与动作

- 比较 S1, S2 内的二进制浮点值，根据大小一致比较结果，对应输出 D 开始的 3 点 ON/OFF 状态。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。



以上均为二进制浮点型数据

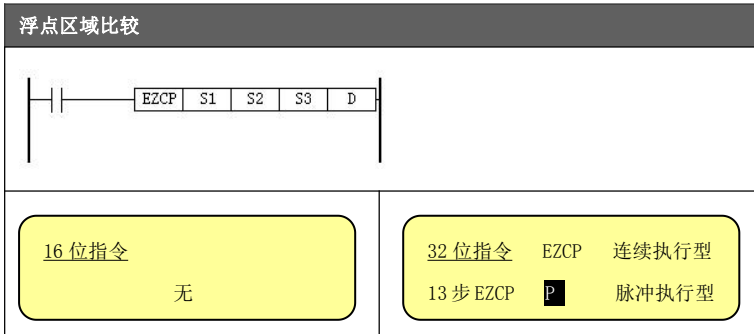
- 比较 (D1001, D1000) 和 (D1101, D1100) 当前值，根据结果驱动 M0、M1、M2 动作。
- 所有源数据都被看成二进制浮点型处理。
- 作为目标地址假如指定 M0，如左记一样 M0、M1、M2 被自动占有。
- 在执行 ECMP 指令后，即使 X000 断开，M0、M1、M2 仍然保持之前状态。



⇒ (K1234) : (D1001, D1000) → (M0, M1, M2)
常数 1234 自动转换成二进制浮点数。

5.4.40 浮点区域比较[EZCP]

将 2 点指定浮点数据与当前浮点数进行大小比较，并输出比较结果。



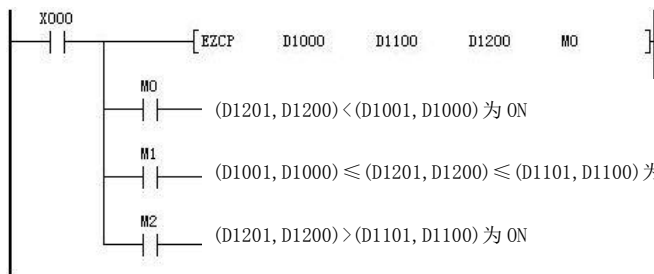
操作数	作用	类型
S1	指定比较基准下限的数据或软元件地址编号	32 位二进制浮点
S2	指定比较基准上限的数据或软元件地址编号	32 位二进制浮点
S3	指定当前数据或软元件地址编号	32 位二进制浮点
D	指定比较输出结果的软元件地址编号	位

操作数	字型软元件										
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								

操作数	位型软元件			
D(占 3 点)	X	Y	M	S
		●	●	●

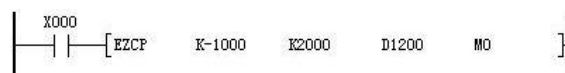
功能与动作

- 比较 S1(下区间比较值)和 S2(上区间比较值)与 S3(当前数据)内的二进制浮点值的大小关系，根据比较结果，对应输出 D 开始的 3 点 ON/OFF 状态。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。



以上均为二进制浮点型数据

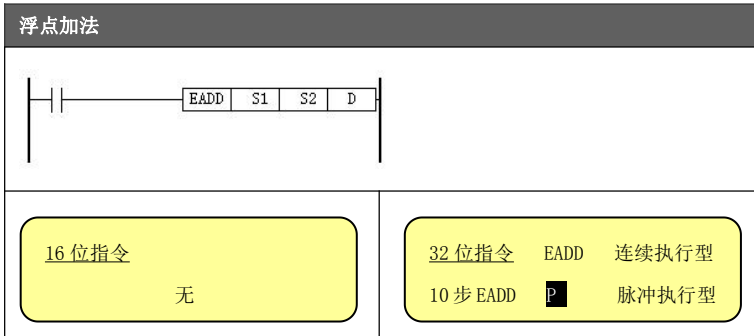
- 比较 (D1001, D1000) 和 (D1101, D1100) 与 (D1201, D1200) 当前值，根据结果驱动 M0、M1、M2 动作。
- 所有源数据都被看成二进制浮点型处理。
- 作为目标地址假如指定 M0，如左记一样 M0、M1、M2 被自动占有。
- 在执行 EZCP 指令后，即使 X000 断开，M0、M1、M2 仍然保持之前状态。



⇒ (K-1000) : (D2001, D2000) : (K2000) → (M0, M1, M2)
常数-1000 和 2000 自动转换成二进制浮点数。

5.4.41 浮点加法[EADD]

使两个数据或软元件内容进行二进制浮点数加法运算并将结果存储到其他软元件中。



操作数	作用	类型
S1	指定进行加法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行加法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存加法运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

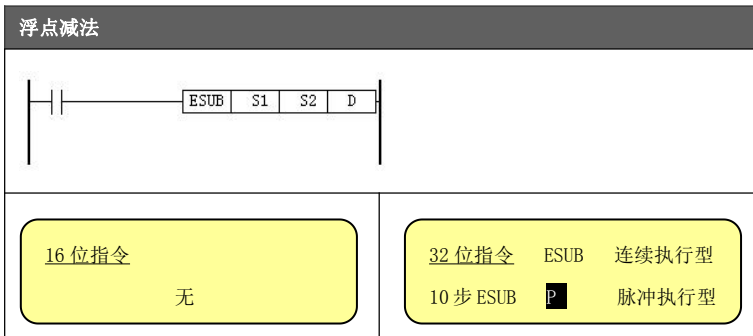
功能与动作

- S1 和 S2 进行二进制浮点加法运算，结果存入 D。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。
- 源数据和目标数据也可指定为同一元件编号，但需注意为连续型指令，若条件满足，在每个扫描周期均会执行一次浮点运算，目标数据将一直变化。



5.4.42 浮点减法[ESUB]

使两个数据或软元件内容进行二进制浮点数减法运算并将结果存储到其他软元件中。



操作数	作用	类型
S1	指定进行减法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行减法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存减法运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

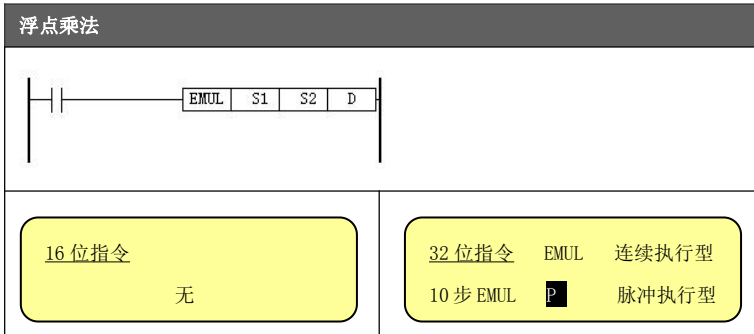
功能与动作

- S1 和 S2 进行二进制浮点减法运算，结果存入 D。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。
- 源数据和目标数据也可指定为同一元件编号，但需注意为连续型指令，若条件满足，在每个扫描周期均会执行一次浮点运算，目标数据将一直变化。



5.4.43 浮点乘法[EMUL]

使两个数据或软元件内容进行二进制浮点数乘法运算并将结果存储到其他软元件中。



操作数	作用	类型
S1	指定进行乘法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行乘法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存乘法运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

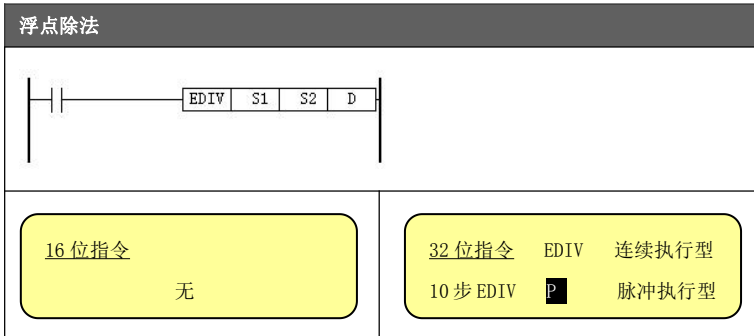
功能与动作

- S1 和 S2 进行二进制浮点乘法运算，结果存入 D。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。
- 源数据和目标数据也可指定为同一元件编号，但需注意为连续型指令，若条件满足，在每个扫描周期均会执行一次浮点运算，目标数据将一直变化。



5.4.44 浮点除法[EDIV]

使两个数据或软元件内容进行二进制浮点数除法运算并将结果存储到其他软元件中。

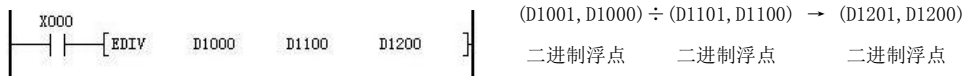


操作数	作用	类型
S1	指定进行除法运算的数据或软元件地址编号	32 位二进制浮点
S2	指定进行除法运算的数据或软元件地址编号	32 位二进制浮点
D	指定保存除法运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

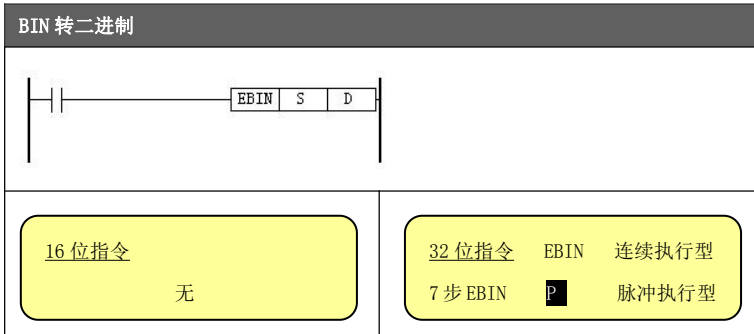
功能与动作

- S1 和 S2 进行二进制浮点除法运算，结果存入 D。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。
- 除数 S2=0 时：
 - ◆ 被除数 > 0，则结果为单精度二进制浮点最大限值 (3.4E038)。
 - ◆ 被除数 ≤ 0，则结果为单精度二进制浮点最小限值 (-3.4E038)。
- 源数据和目标数据也可指定为同一元件编号，但需注意为连续型指令，若条件满足，在每个扫描周期均会执行一次浮点运算，目标数据将一直变化。



5.4.45 浮点十进制转浮点二进制[EBIN]

将指定软元件之内的十进制浮点数转换为二进制浮点数存入目标软元件内。

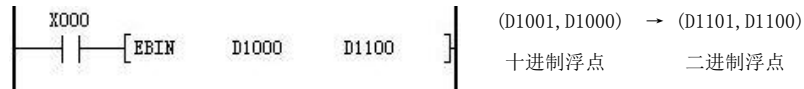


操作数	作用	类型
S	指定源数据的软元件地址编号	32 位十进制浮点
D	指定目标软元件地址编号	32 位二进制浮点

操作数	字型软元件										
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								

功能与动作

把源数据指定的元件内的十进制浮点值转换为二进制浮点值，存入目的地址中。



十进制浮点

指数	尾数
D1001	D1000

 [D1000] × 10[D1001]

二进制浮点

D1101	D1100
-------	-------

 尾数部分 23 位，指数部分 8 位，尾数符号 1 位

指数的输入范围通常为±38，结合尾数部分必须保证其转换值在±3.4E038 范围之内。

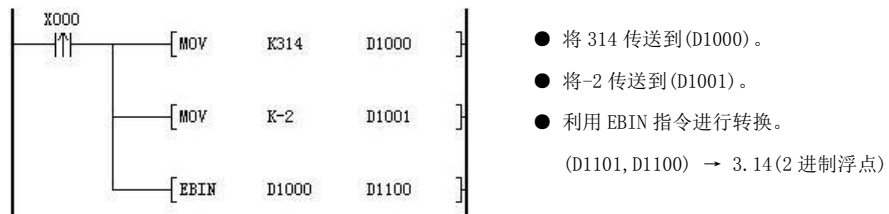
浮点运算在可编程控制器内部全部以二进制浮点值为基础执行。但是，由于二进制浮点值为不易判别的数值，所以可通过能将其转换为十进制浮点值的外部设备很容易地监控。

编程应用示例

使用 EBIN 指令，可以使含有小数点的数值直接转换为二进制浮点数。

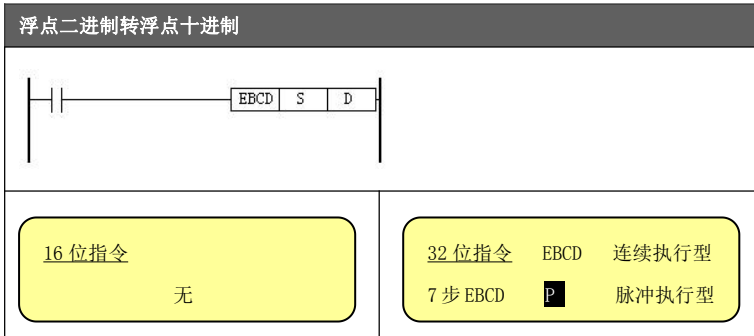
例：3.14 的二进制浮点转换。

3.14=314×10⁻²(十进制浮点)



5.4.46 浮点二进制转浮点十进制[EBCD]

将指定软元件之内的二进制浮点数转换为十进制浮点数存入目标软元件内。



操作数	作用	类型
S	指定源数据的软元件地址编号	32 位二进制浮点
D	指定目标软元件地址编号	32 位十进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

把源数据指定的元件内的二进制浮点值转换为十进制浮点值，存入目的地址中。



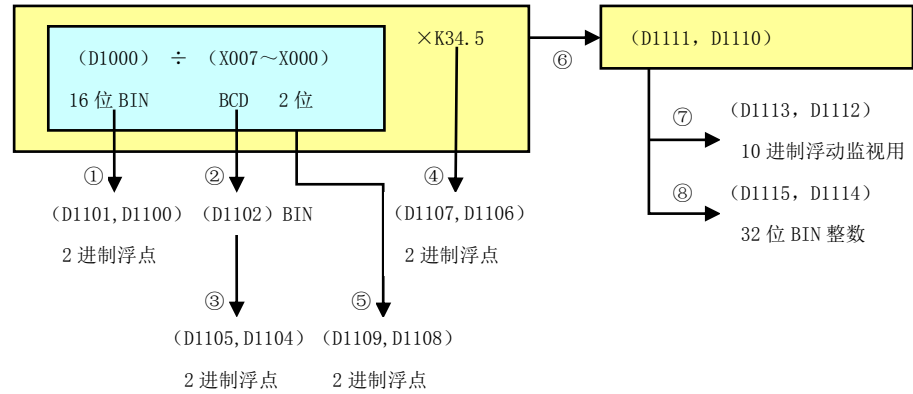
二进制浮点 D1001D1000 尾数部分 23 位，指数部分 8 位，符号 1 位

指数 尾数

十进制浮点 D1101D1100 $[D1100] \times 10 [D1101]$

浮点综合运
算示例

浮点运算综合应用的示例如下所示。

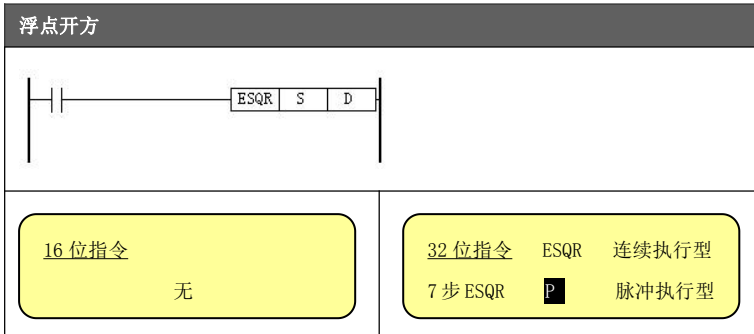


用梯形图实现上述结构图的运算过程：



5.4.47 浮点开方[ESQR]

对指定数据或软元件中内容(二进制浮点数)进行开方运算。

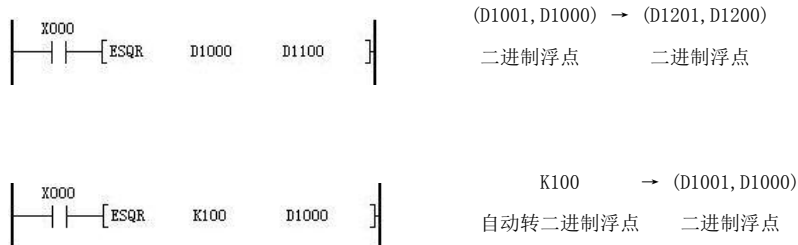


操作数	作用	类型
S	指定进行开方运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

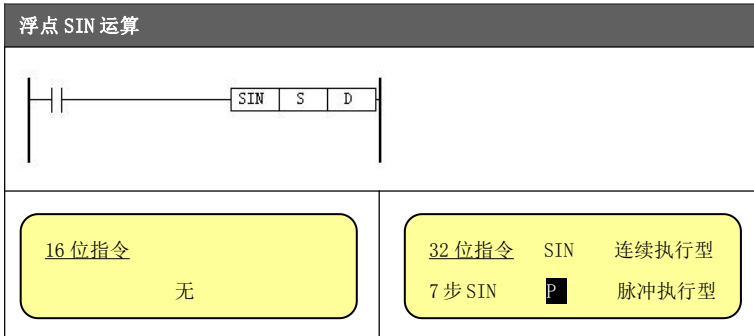
功能与动作

- 进行 S 内二进制浮点值的平方根运算，结果作为二进制浮点数存入 D。
- 常数 K/H 被指定为源数据时，自动转换成二进制浮点值处理。
- 非法使用的情况：
 - ◆ S 不在单精度浮点数取值范围内(-3.4E038~3.4E038)，计算结果为 0。
 - ◆ S < 0，运行将出错。
- 与整数型开方指令 SQR 相比，其运算结果会保留小数位。
- 源数据和目标数据也可指定为同一元件编号，但需注意为连续型指令，若条件满足，在每个扫描周期均会执行一次浮点运算，目标数据将一直变化。



5.4.48 浮点 SIN 运算[SIN]

对指定数据或软元件中内容(二进制浮点数)进行正弦运算。

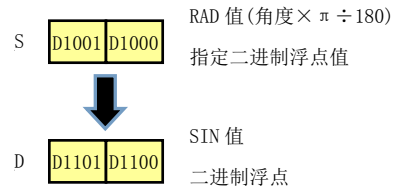


操作数	作用	类型
S	指定进行 SIN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件										
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								

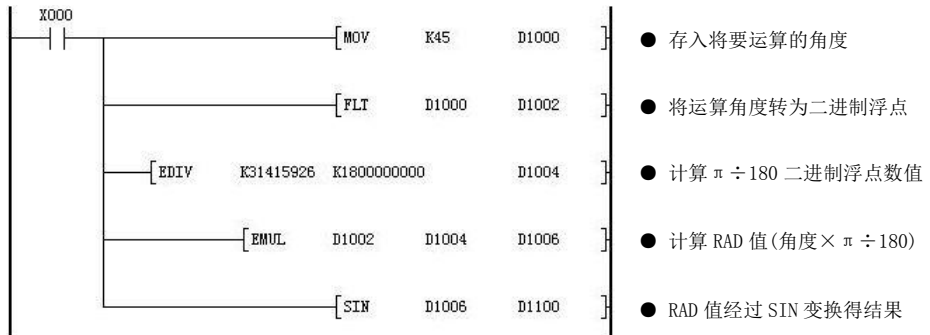
功能与动作

为求源数据指定的弧度(RAD)的 SIN 值,并传送到目的地址中的指令。



SIN 编程示例

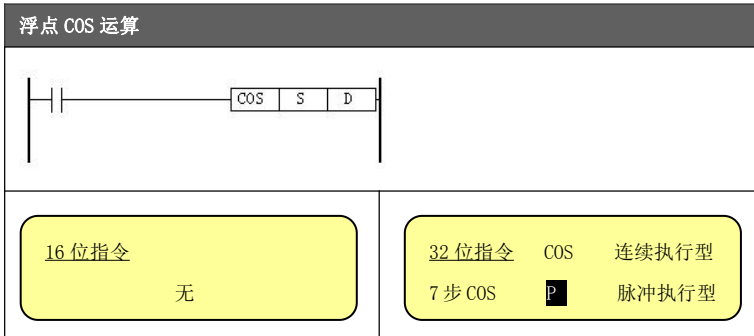
求角度 45° 的 SIN 值梯形图示例。



RAD 值也可以通过 RAD 指令(章节 5.4.54)进行计算。

5.4.49 浮点 COS 运算[COS]

对指定数据或软元件中内容(二进制浮点数)进行余弦运算。

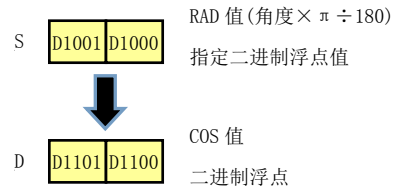


操作数	作用	类型
S	指定进行 COS 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

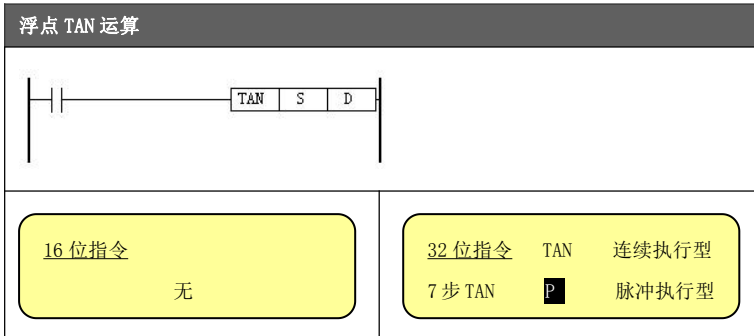
功能与动作

为求源数据指定的弧度(RAD)的 COS 值,并传送到目的地址中的指令。



5.4.50 浮点 TAN 运算[TAN]

对指定数据或软元件中内容(二进制浮点数)进行正切运算。

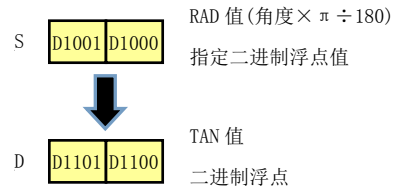


操作数	作用	类型
S	指定进行 TAN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
S	●	●	●							
D		●	●							

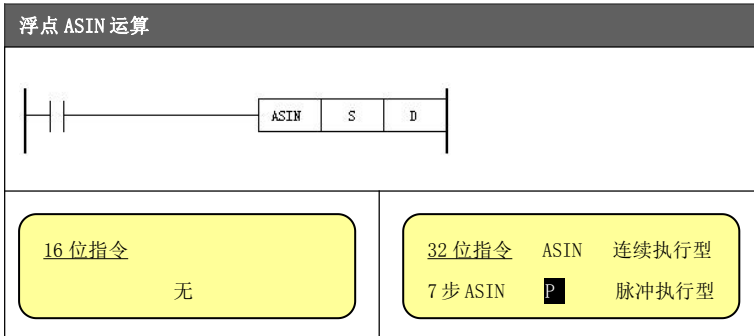
功能与动作

为求源数据指定的弧度(RAD)的 TAN 值，并传送到目的地址中的指令。



5.4.51 浮点 ASIN 运算[ASIN]

对指定数据或软元件中内容(二进制浮点数)进行反正弦运算。



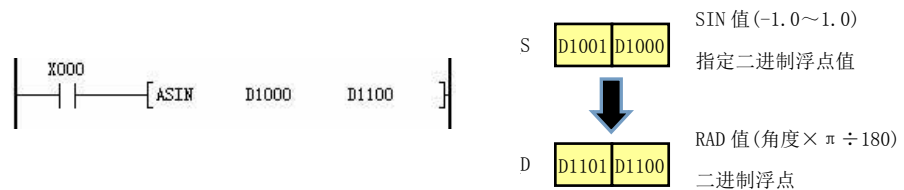
操作数	作用	类型
S	指定进行 ASIN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

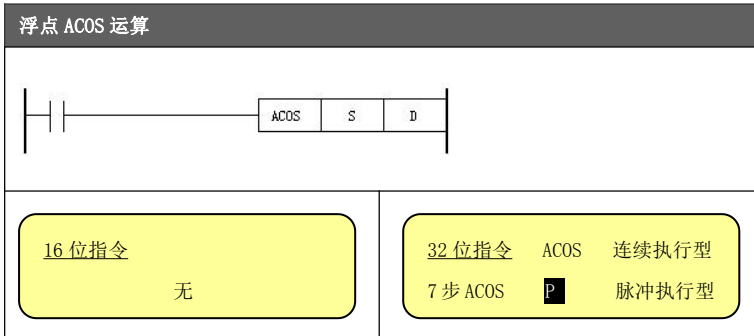
为源数据 S 指定的 SIN 值求出弧度 (RAD)，并传送到目的地址 D 中的指令。

- S 必须指定-1.0~1.0 之间的数值。
- D 中保存的角度(运算结果)是保存弧度 $(-\pi/2) \sim (\pi/2)$ 的值。



5.4.52 浮点 ACOS 运算[ACOS]

对指定数据或软元件中内容(二进制浮点数)进行反余弦运算。



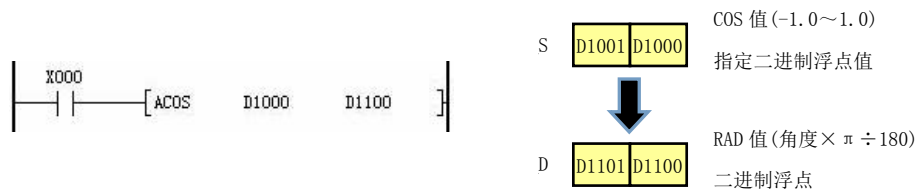
操作数	作用	类型
S	指定进行 ACOS 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件										
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								

功能与动作

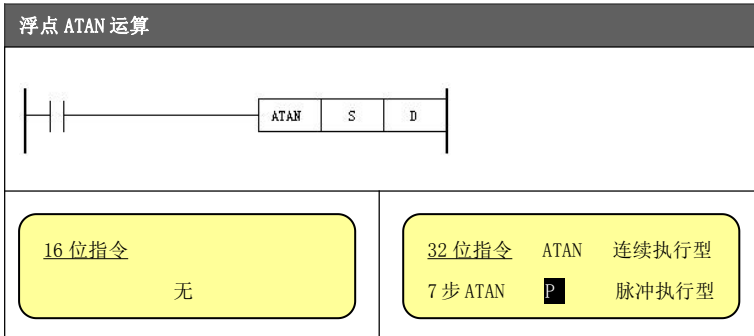
为源数据 S 指定的 COS 值求出弧度 (RAD)，并传送到目的地址 D 中的指令。

- S 必须指定 -1.0~1.0 之间的数值。
- D 中保存的角度 (运算结果) 是保存弧度 $0 \sim \pi$ 的值。



5.4.53 浮点 ATAN 运算[ATAN]

对指定数据或软元件中内容(二进制浮点数)进行反正切运算。



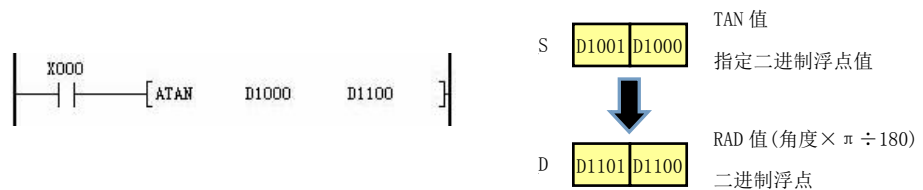
操作数	作用	类型
S	指定进行 ATAN 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

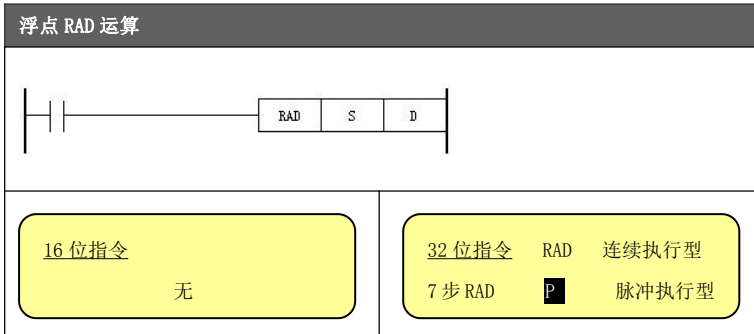
为源数据 S 指定的 TAN 值求出弧度 (RAD)，并传送到目的地址 D 中的指令。

- D 中保存的角度 (运算结果) 是保存弧度 $(-\pi/2) \sim (\pi/2)$ 的值。



5.4.54 浮点 RAD 运算[RAD]

对指定数据或软元件中内容(二进制浮点数)进行角度转弧度运算。



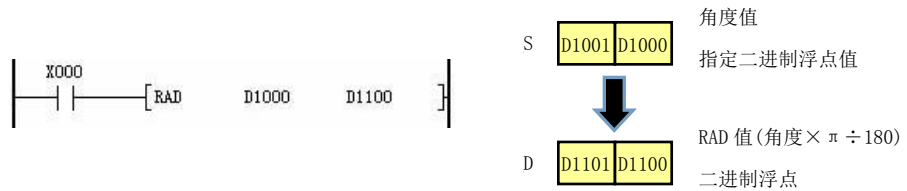
操作数	作用	类型
S	指定进行 RAD 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件									
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

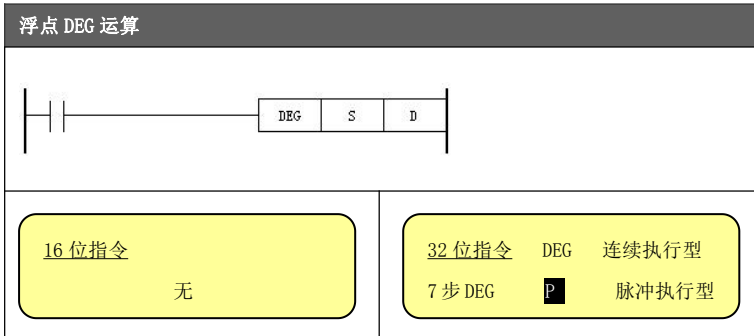
为源数据 S 指定的角度求出弧度(RAD)，并传送到目的地址 D 中的指令。

● $RAD \text{ 值} = \text{角度} \times \pi \div 180。$



5.4.55 浮点 DEG 运算[DEG]

对指定数据或软元件中内容(二进制浮点数)进行弧度转角度运算。



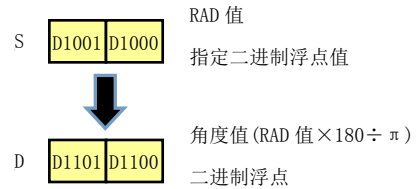
操作数	作用	类型
S	指定进行 DEG 运算的数据或软元件地址编号	32 位二进制浮点
D	指定存储运算结果的软元件地址编号	32 位二进制浮点

操作数	字型软元件										
S	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								

功能与动作

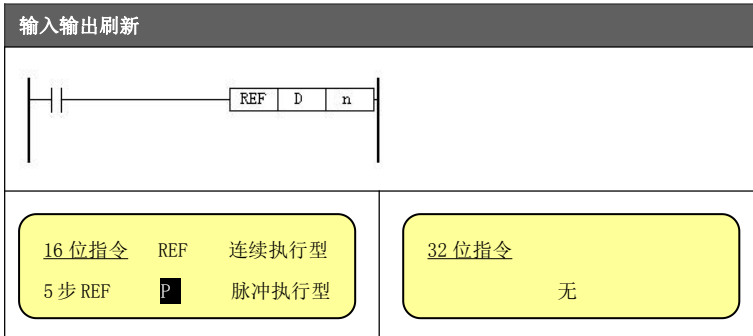
为源数据 S 指定的弧度 (RAD) 求出角度，并传送到目的地址 D 中的指令。

● 角度 = RAD 值 × 180 ÷ π。



5.4.56 输入输出刷新[REF]

对指定输入输出端口的状态立即进行刷新。



操作数	作用	类型
D	指定刷新端口的软件地址编号	16 位 BIN
n	指定刷新位数	16 位 BIN

操作数	位型软元件			
D	X	Y	M	S
	●	●		

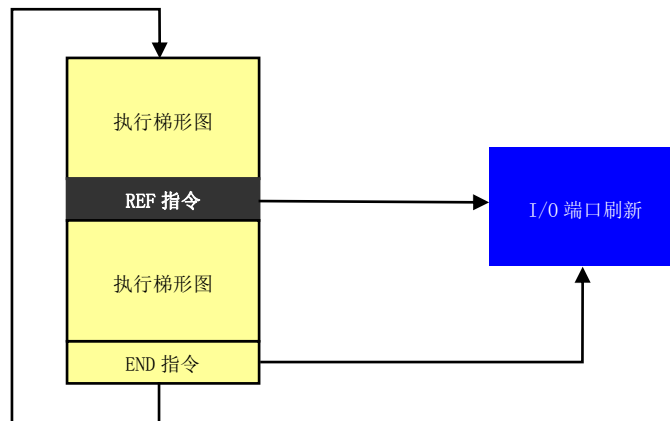
操作数	字型软元件									
n	K/H	R	D	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

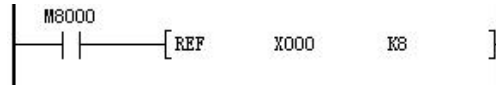
通常情况下，系统对 X/Y 输入输出端口的刷新在梯形图扫描周期结束后执行。但是，在梯形图执行过程中，需要最新的输入信息以及希望立即输出运算结果时，可以使用输入输出刷新指令。

- 对输入端子进行刷新时，D 必须填写 X000 或 X010。
- 对输出端子进行刷新时，D 必须填写 Y000 或 Y010。
- n 值必须填写 8。

输入输出刷新在有输入输出动作的中断处理中使用较多，在中断子程序中进行输入输出的刷新，获取最新的输入信息并且及时输出运算结果。

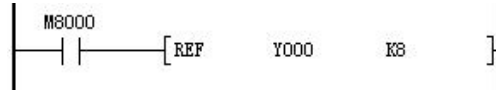


输入刷新



获取 X000~X007 物理端口状态。
对运动控制器而言，这些端口均指向有意义的物理端口。

输出刷新

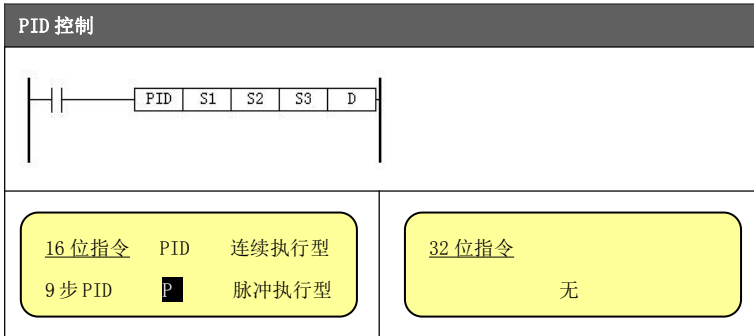


输出 Y000~Y007 状态到物理端口。
对运动运动器而言，Y000~Y005，Y010~Y017 指向有意义的物理端口。
同时需要注意的是，当系统寄存器区 D8180~D8186，D8188~D8195 对应数值不为 0 时，针对该输出端口的刷新无效。

寄存器	数值意义
D8180~D8185	= 0, PLC 控制。
对应 Y000~Y005 的输出方式	= 1, 高速事件输出 (运动指令使用)
D8188~D8195	= 2, 运动轴端口/轴方向指定 (运动指令使用)
对应 Y010~Y017 的输出方式	

5.4.57 PID 控制 [PID]

将指定软元件的内容进行 PID 控制。



操作数	作用	类型
S1	设定目标值的软元件地址编号	16 位 BIN
S2	测定值的软元件地址编号	16 位 BIN
S3	设定控制参数的软元件首地址编号	16 位 BIN
D	输出值的软元件地址编号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
S3(占 25 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

- P-比例调节，I-积分调节，D-微分调节，应用于闭环控制系统。
- S3~S3+24 作为控制参数区被该指令占用，不可用作其他普通数据寄存器。
- 可通过 T 定时器或定时中断来设置 PID 指令的调节周期。
- PID 指令可在定时器中断、子程序、步进梯形图、跳转指令中使用，首次执行前必须清除偏差值 (S3+7)。

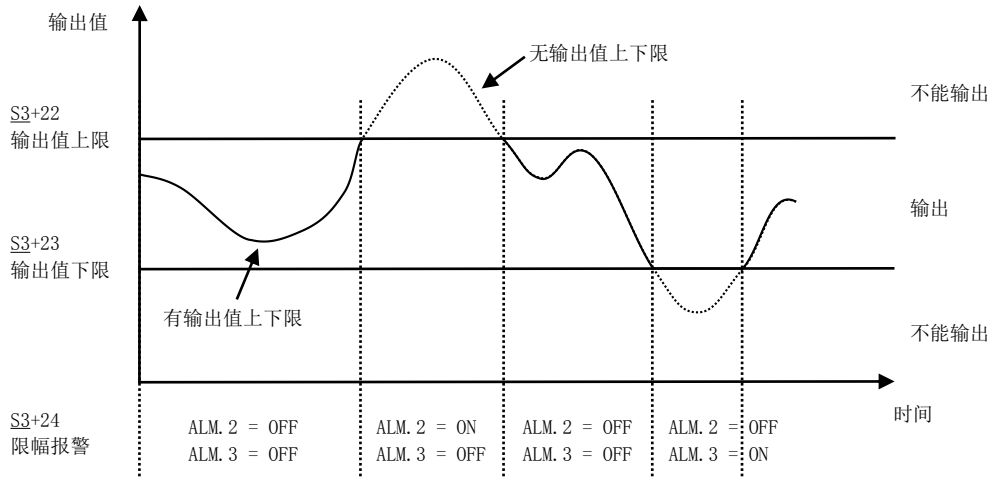
S3 参数设定
定义

PID 指令运算参数设定表。

S3 + 偏移	名称	控制参数设置值	备注	读写
S3 + 0	采样时间	PID 控制采样时间，此参数决定 PID 运算的间隔时间，单位 ms。	梯形图需配合设定的时间执行指令。 例如设置为 100ms 的采样时间，则梯形图可设计 100ms 定时器，定时器触点导通执行一次 PID 运算。	设置值
S3 + 1	运算方向	= 0，逆向计算；= 1，正向计算	推荐设置 1	设置值
S3 + 2	滤波系数	0~99 %	推荐设置 50，设置为 0 则无滤波	设置值
S3 + 3	比例增益	0~32767 %	推荐设置 2000	设置值
S3 + 4	积分时间	0~32767，单位 ms。	推荐设置 500	设置值
S3 + 5	微分增益	0~32767 %	推荐设置 0，无微分计算。	设置值
S3 + 6	微分时间	0~32767，单位 ms。	推荐设置 0，无微分计算。	设置值
S3 + 7(L)	偏差	内部计算使用	首次执行指令前清 0 二进制浮点格式(32 位)	显示值
S3 + 9(L)	偏差一阶倒数	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 11(L)	偏差二阶倒数	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 13(L)	滤波后测定值	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 15(L)	微分调整项	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 17(L)	PID 增量输出	内部计算使用	二进制浮点格式(32 位)	显示值
S3 + 19	PID 输出	内部计算使用	-	显示值
S3 + 20	上限报警	不能超过 32767	推荐设置值 20000	设置值
S3 + 21	下限报警	不能小于 0	推荐设置值 0	设置值
S3 + 22	上限幅	不能超过 32767	推荐设置值 10000	设置值
S3 + 23	下限幅	不能小于 0	推荐设置值 20	设置值
S3 + 24	报警输出	根据输出结果检测报警状态	助记符 ALM，按位显示上下限和上下限幅的报警状态。 ALM.Bit0 = ON，上限报警 ALM.Bit1 = ON，下限报警 ALM.Bit2 = ON，上限幅状态 ALM.Bit3 = ON，下限限幅状态	显示值

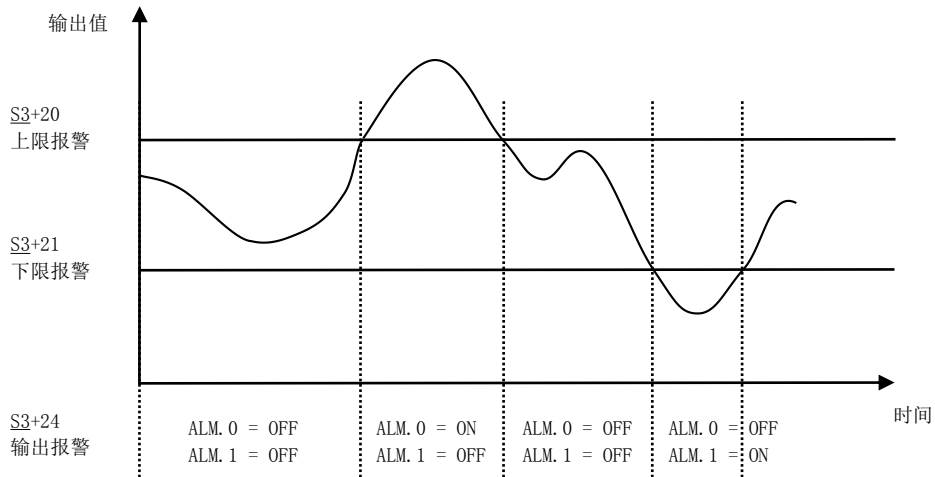
**输出值
上下限**

以下描述 PID 控制对于输出上下限幅的作用示意图。



**报警
上下限**

以下描述 PID 控制对于输出报警上下限的作用示意图。



应用场合

- PID 指令应用十分广泛，例如纺织行业的松紧架控制，供水行业、供气行业、汽车行业、窑炉行业、养殖行业中的恒压供水供气 and 恒温控制等。PLC 内置 PID 控制可以适应各种复杂的控制、多变量的控制。
- 对于运动控制器，由于没有热电偶接入接口和模拟量输出接口，故不能直接应用于控温系统。但可选配带热电偶 A/D 采样控制器或模块，利用通信方式将 A/D 值传输进行 PID 运算，将运算结果通信到控制器或模块进行输出调节。

5.5 中断系统

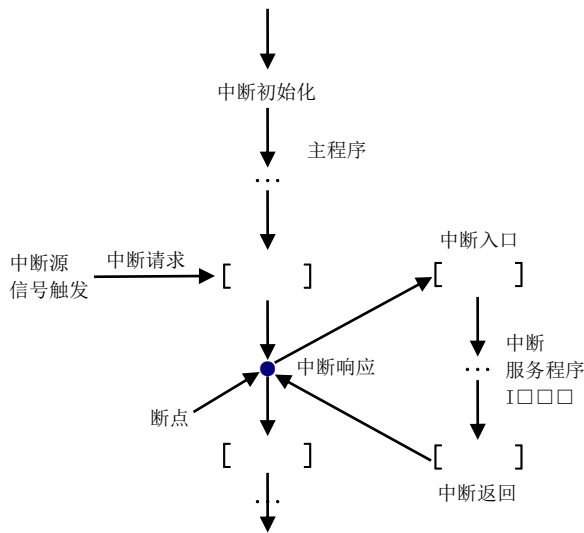
运动控制器具有丰富的中断资源，用户可在程序中调用不同的中断实时处理多种不同类型的任务。综合运用不同的中断，可以实现运动控制器与多个外设同时运行，增强系统实时性，增加系统对故障的处理能力。

5.5.1 中断资源介绍

中断名称	中断标号	说明
外部输入中断	I00□、I10□、I20□、I30□、I40□	外部输入信号 X004~X007 作为中断源，当外部输入信号到来，执行相应中断程序。
定时器中断	I6□□、I7□□、I8□□	定时器定时时间作为中断源，当定时时间到就执行相应中断程序。
系统软中断	I0□0	设置中断申请位执行相应中断程序。
高级中断	I900~I907	用于实现高速并发任务的处理以及插补算法的实现。

5.5.2 中断执行流程

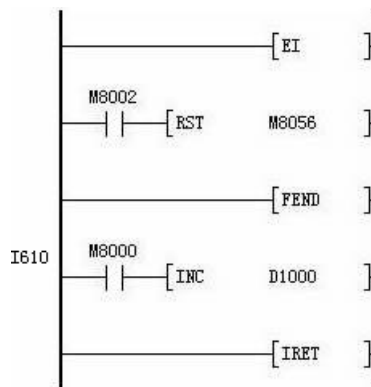
中断就是指某事件的发生引起 CPU 暂停当前程序的运行，转入对所发生事件的处理，处理结束又回到程序被打断处接着执行这样的过程。



- 梯形图是周期扫描型的处理机制，为了能够响应实时性比较强的任务，引入了梯形图中断的概念。
- 可利用在梯形图母线左端添加 I 标号的方式指定一个梯形图中断。
- 系统每扫描完一条指令都会扫描一次所有的中断申请。若中断源满足触发条件且申请有效，则执行相应的中断服务程序。
- 中断执行的流程如左图所示。

应用与示例

以定时器中断为例，列举梯形图的程序对中断的执行流程进行说明。



- EI 开中断
- 清除中断禁止标志位
- 主程序结束
- 中断执行程序
- 中断返回

以上程序执行的过程为：

- 程序中 EI 之后的每条指令，CPU 每扫描完一条指令都会扫描一次中断申请。
- 扫描到定时中断 I610 中断申请后，准备接收 I610 中断源信号。
- 当 10ms 定时时间到，系统中断当前主程序跳转到中断程序，中断程序执行完后 (D1000 寄存器自加 1)，返回主程序，继续执行主程序。
- 当下一个 10ms 定时时间到，系统再次中断当前主程序，跳转到中断程序。中断程序执行完后，返回主程序，继续执行主程序。
- 如此循环，直到进入中断的条件不成立。

中断具体过程

中断的处理过程分为以下五个阶段：



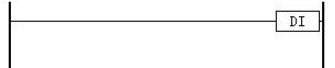
- 中断请求
- 中断仲裁
- 中断响应
- 中断处理
- 中断返回

当中断源向 CPU 发出中断请求时，CPU 通过仲裁决定是否响应此次中断。当 CPU 命中某个中断后，会暂停当前执行的主程序，并将有关的寄存器内容以及标志位状态推入堆栈保留下来，转而执行相应的中断子程序，当中断子程序执行完后，CPU 会恢复进入中断之前的寄存器状态，继续执行主程序。

5.5.3 中断指令[EI], [IRET], [DI]

定时器中断、外部输入信号中断和反馈信号中断的指令主要有以下三条，高级中断的指令将在高级中断中加以介绍。

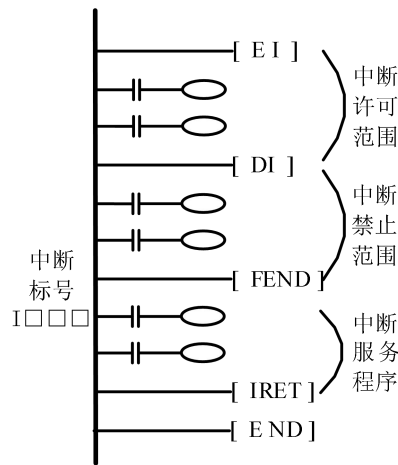
助记符与功能

助记符、名称	功能	回路表示	程序步
EI 开中断	开始中断申请扫描	 操作元：无	1
IRET 中断返回	中断结束返回主程序	 操作元：无	1
DI 中断禁止	禁止中断申请扫描 (禁止中断)	 操作元：无	1

指令使用说明

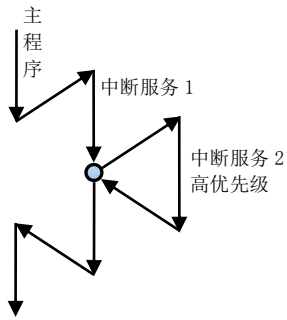
- 以上三条指令都直接与母线连接，中间不能添加其他任何指令；
- EI 和 DI 指令的作用相反，为开/关中断检测的指令。IRET 指令用于中断子程序的返回，每一个 I 标号都必须有一个 IRET 指令进行匹配，否则运行将出错；
- 当程序中有多个中断时，EI 和 DI 只需要使用一次（当不需要禁止中断时，不用 DI 指令）。

应用与示例



- 左侧梯形图中，EI 为开中断，即在此条指令后，系统每扫描完一条指令都会扫描一次中断申请。
- DI 为禁止中断，即在 DI 到 FEND 指令之间区间，禁止接收中断信号，如果中断信号在梯形图运行到此区间内到来，则会延迟到此区间运行结束后进入中断子程序。如果 DI 到 FEND 指令之间的程序运行时间较长，则中断响应有较明显的延迟。
- FEND 表示主程序结束，中断子程序只能在 FEND 指令之后。
- IRET 指令表示中断返回。

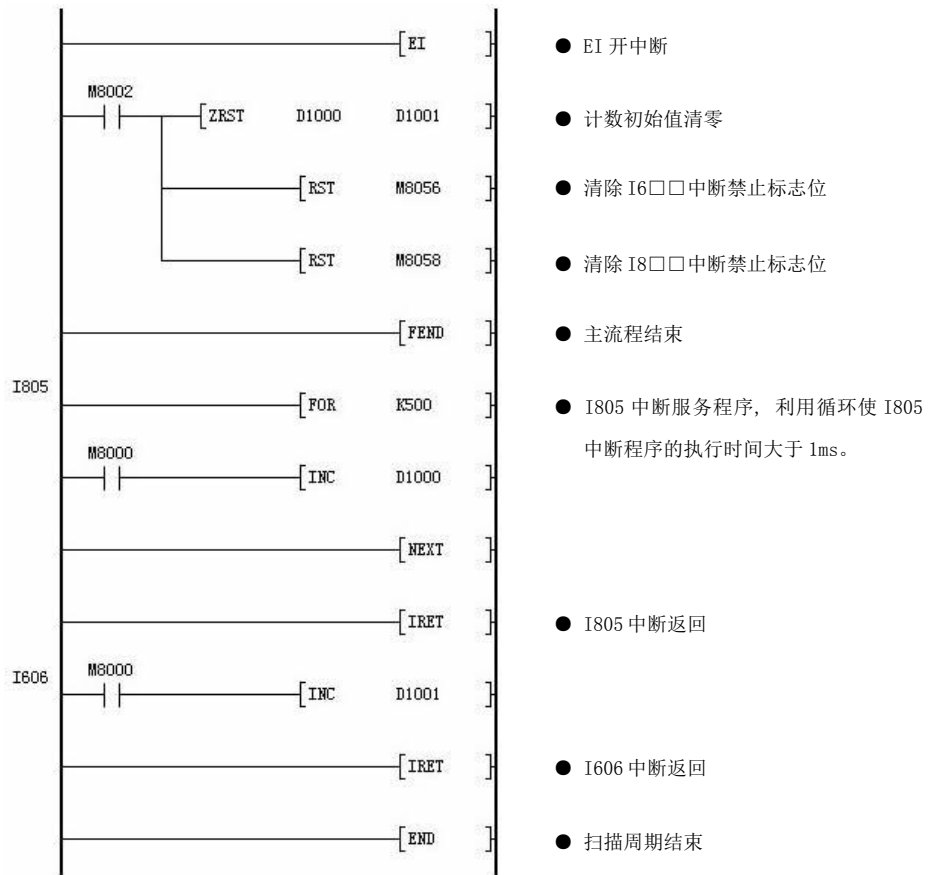
5.5.4 中断嵌套



- 如果在执行一个外部设备的中断服务程序期间允许中断，同时已经建立了一种优先链结构，那么优先级更高的设备就能中断现行的服务程序，这就叫中断嵌套。这种嵌套的结构将使高速设备能进入中断，而暂时挂起较低速设备的中断服务。在完成了较高速设备的服务后，就自动返回去完成较低速设备的服务程序。
- 中断嵌套最多只能 8 层。
- 左图即为中断嵌套的流程。

应用与示例

下面以定时中断组成的嵌套程序为例，对中断嵌套加以说明。

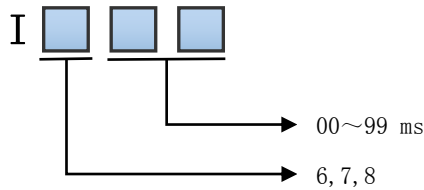


以上程序执行的过程为：

- 程序中 EI 之后的每条指令，CPU 每扫描完一条指令都会扫描一次中断申请；
- 扫描到定时中断 I6□□和 I8□□的中断申请后，准备接收中断源信号；
- 当 5ms 定时时间到，系统中断当前主程序跳转到 I805 中断程序；
- 1ms 后，I606 中断信号到，系统中断 I805 程序的执行跳转到 I606 中断，执行 I606 的程序；
- I606 中断程序执行完后，返回到 I805 中断，继续执行 I805 中断程序；
- I805 中断执行完后，返回主程序继续执行
- 如此循环，直到进入中断的条件不成立。

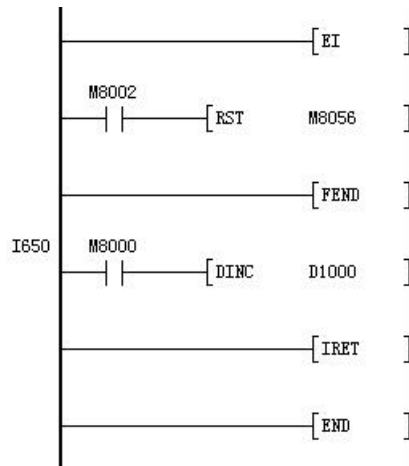
5.5.5 定时器中断

定时中断即是以定时器的定时时间作为中断源的中断，适用于需要以固定的时间执行某些任务的场合。运动控制器提供三个定时中断，中断标号以及对应的控制字如下表：



输入编号	中断周期 (ms)	中断申请标志位	中断禁止标志位
I6□□	在□□中输入 00-99 的整数，表示扫描周期。 如 I650，则表示每隔 50ms 执行一次定时器中断	M8040	M8056
I7□□		M8041	M8057
I8□□		M8042	M8058

应用与示例



- 执行 EI 指令后，允许定时器中断。
- 清除 I6□□中断标识禁止位。
- FEND 表示主程序结束，中断程序必须位于 FEND 指令之后。
- I650 表示使用的中断标号为 I6□□的定时中断，每隔 50ms 执行中断子程序，用 IRET 指令返回主程序。

编程注意事项

- 指针号为 (I6, I7, I8) 的指针不可重复使用。如果重复使用，系统只识别最后一个中断编号。
- 系统中默认为禁止中断状态，即 M8056~M8058 默认处于置位状态，如果需要使用某个定时中断，需要在程序中将对应的控制位进行复位。
- 定时中断的中断程序执行时间应小于或等于定时时间（当等于定时时间时，则系统不能执行其他任务），如果大于定时时间，则程序会出错。
- 若定时中断标号后 2 位为 0（如 I700），则中断定时时间为 100ms。

5.5.6 外部输入中断

外部输入中断即是以运动控制器的外部输入信号作为中断源的中断，适用于当外部信号到来时需要执行某些任务的场合。

运动控制器提供 4 个外部输入事件输入源 (X004~X007)，其对应端口动作特性和中断标号的关联如下所示。

外部输入信号	中断标号	中断特性	中断申请标志位	中断禁止标志位
X004	I000	下降沿	M8032	M8048
-	I001	-	M8033	M8049
X005	I100	下降沿	M8034	M8050
	I101	上升沿	M8035	M8051
X006	I200	下降沿	M8036	M8052
-	I201	-	M8037	M8053
X007	I300	下降沿	M8038	M8054
	I301	上升沿	M8039	M8055

可以通过置位外部输入中断的中断申请位，进入对应的中断服务程序，包括没有对应的 X 输入源的外部输入中断（比如 I001 和 I201）。当系统响应中断申请位进入中断服务程序后，会自动清除中断申请位。

当使用中断申请位调用外部输入中断服务程序时，如果不希望连续进入中断服务程序，不要使用常开或常闭触点去置位中断申请标志位。可以使用 LDP, LDF 边沿指令实现单次置位中断申请标志位。

在复位中断禁止标志位之前，先清除相应的中断申请标志位，以免进入一个非预期的外部中断服务程序。比如 X004 有下降沿动作，M8032 会被置位。那么一旦复位 M8048，就会立即进入 I000 中断。

外部输入中断的中断特性可以通过运动输入轴参数的脉冲捕获寄存器进行配置：

脉冲捕获寄存器	脉冲捕获方式	外部中断事件特性	脉冲捕获寄存器	脉冲捕获方式	外部中断事件特性
D211	H00	X004 下降沿	D261	H00	X006 下降沿
	H10	X004 下降沿 X005 下降沿		H10	X006 下降沿 X007 下降沿
	H20	X004 下降沿 X005 上升沿		H20	X006 下降沿 X007 上升沿
	H11	X005 下降沿		H11	X007 下降沿
	H21	X005 上升沿		H21	X007 上升沿
	H31	无		H31	无

除了重新上电运动控制器更新脉冲捕获方式的配置方式，也可以在高速计数指令运行中改变外部输入中断方式的配置方式，其步骤如下：

- 设置输入轴参数的脉冲捕获方式
- 复位本路对应的高速计数器 RST C2XX，比如 X0 对应 C235，X2 对应 C237。

X0 端子对应输入运动轴的脉冲捕获寄存器为 D211，X2 端子对应的脉冲捕获寄存器为 D261。高速计数器 C2XX 对应的 X 端子参考章节 5.1.5 描述。

举例在线修改 X005 的外部中断方式从下降沿改为上升沿，其步骤如下：

- 设置 D211=H20，配置为 X004 下降沿进中断，X005 上升沿进中断。
- 执行 RST C235。C235 对应的高速计数端子为 X0，而 X0 对应输入运动轴的脉冲捕获寄存器为 D211。

5.5.7 系统软中断

运动控制器提供 5 个系统软中断信号，其对应特性如下所示。

中断标号	中断申请标志位	中断禁止标志位
I010	M8043	M8059
I020	M8044	M8060
I030	M8045	M8061
I040	M8046	M8062
I050	M8047	M8063

I010 为系统 1MS 定时软中断，无需用户设置中断申请位(M8043 由系统自动申请)。

软中断执行 条件

以 I020 系统软中断为例，开启并进入 I020 系统软中断的条件如下：

- 按照梯形图规范编写 I020 中断服务程序
- 设置 EI 指令
- 屏蔽中断禁止标志位(RST M8060)
- 置位中断申请标志位(SET M8044)

5.5.8 高级中断

在运动系统的控制中,一般需要更优化的中断方案。对此,运动控制器采用高级中断模式满足高速并发任务的处理。高级中断中支持进程调度语句。

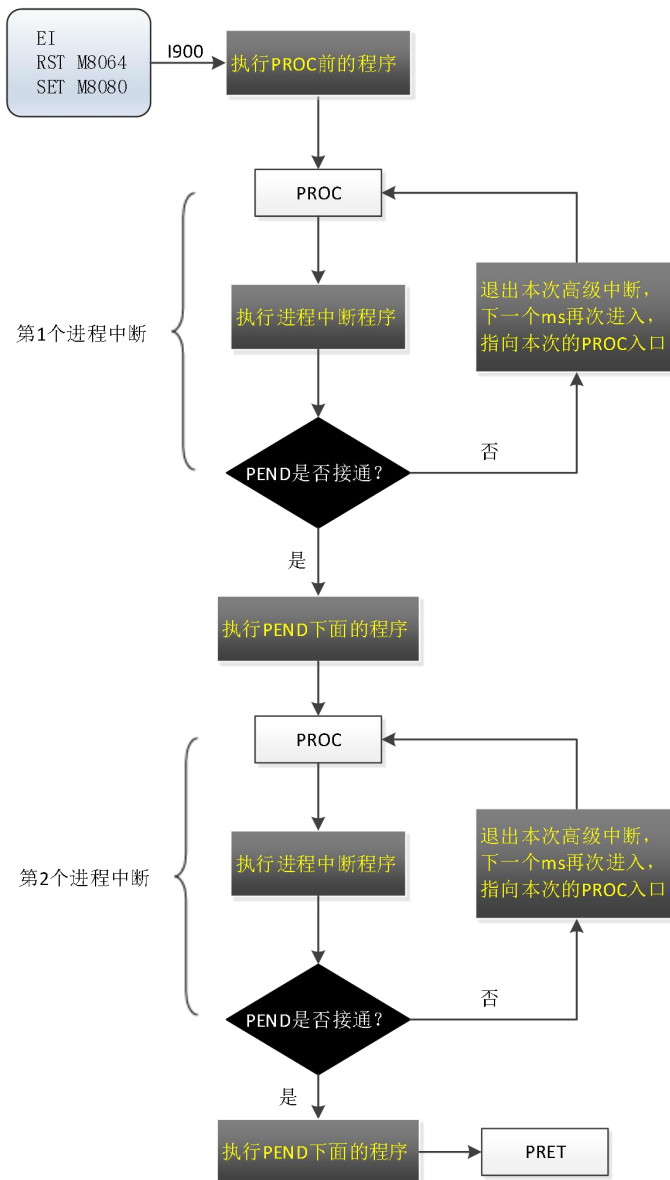
在高级中断中完成一个运动过程需要进行多次插补运算,而插补脉冲输出需要一定的时间,这就可以让系统不断以进程中断的方式进出高级中断,来完成某一运动过程。如果需要完成的运动控制是由多个运动过程组成的,那么可以在高级中断中定义多个进程中断。使用进程调度语句调配这多个进程中断,进程中断的触发信号仅由位置插补周期定时触发。

高级中断执行过程

高级中断的执行过程分两种情况:

- 没有进程调度语句时,其执行过程和低级中断的执行过程一样,每触发一次执行一次。
- 有进程调度语句,碰到 PROC 进程调度语句时,进程中断每触发一次,执行一次。多次出入本次高级中断(进程中断执行时,需多次重复执行,如插补运算)直至 PEND 语句得电,执行下一条梯行图语句。最终碰到 PRET 高级中断返回。

以 I900 为例,假定内含 2 个进程中断。一个完整的高级中断执行流程如下:

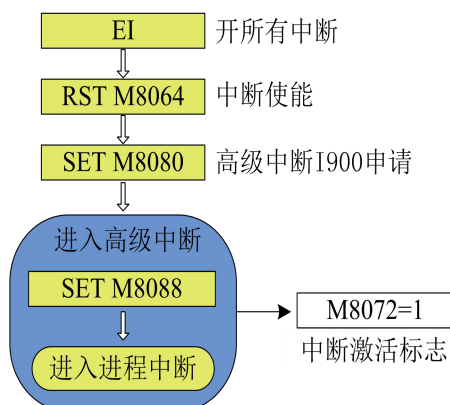


控制字说明

- 运动控制器共有 8 个高级中断，I900~I907 代表 8 个高级中断入口。高级中断以发生时间顺序逐一完成，相互之间不能抢占嵌套。高级中断独立保护断点，不影响低级中断的嵌套层数及功能。
- I900 的进程中中断申请字 M8088 由系统 1ms 定时置位申请。I901~I907 的进程中中断申请字 M8089~M8095 由用户梯形图置位申请，比如遇到光电信号进行置位。进入高级中断后进程中中断申请字系统自动清零。
进程中中断申请字用于再次进入高级中断服务程序(首次进入高级中断仅置位进程中中断申请字即可，不需要置位进程中中断申请字)。
- 进入高级中断流程后，中断激活标志为置位状态。可通过其状态来判断高级中断程序是否已经结束。当高级中断内所有的进程中中断服务程序均已运行结束，高级中断程序自动结束，中断激活标志复位。当需要中途退出高级中断，可对中断激活标志进行复位操作，也可直接结束高级中断。

中断标号	中断禁止标志	中断激活标志	中断申请字	进程中中断申请字
	ON (禁止中断) OFF (使能中断)	ON (进入中断) OFF (未进入中断)	ON (申请应答) OFF (申请不应答)	ON (申请应答) OFF (申请不应答)
I900	M8064	M8072	M8080	M8088
I901	M8065	M8073	M8081	M8089
I902	M8066	M8074	M8082	M8090
I903	M8067	M8075	M8083	M8091
I904	M8068	M8076	M8084	M8092
I905	M8069	M8077	M8085	M8093
I906	M8070	M8078	M8086	M8094
I907	M8071	M8079	M8087	M8095




中断申请字由梯形图置位触发高级中断。例如当 M8080 = 1, M8064 = 0 时，触发 I900 高级中断。进入高级中断后，中断申请字会被系统清零。需要注意的是，中断申请字要采用单次执行指令（如 PLS，STL）进行置位，否则梯形图扫描时会重复执行该指令，造成多次申请中断。



- 以高级中断的 I900 使用为例，在使用高级中断之前，要开中断允许 EI，然后将其中断使能位 M8064 复位，最后采用跳沿指令将 I900 的申请置位（采用跳沿指令是防止多次申请中断，造成多次中断现象，另外，如果在主程序开头申请高级中断，不能使用跳沿指令，因为此时系统刚上电，寄存器的电平状态不稳定，可能造成跳沿指令无法识别）。
- 当进入 I900 高级中断后，中断激活标志位 M8072 置位，进程中中断申请 M8088 是由插补周期进行触发，每个插补周期触发一次（间隔时间为 1ms）。
- 如果高级中断处于进程中中断执行时，要退出高级中断，须复位 M8072 才能跳出中断，此时如果再申请高级中断（SET M8080），可以继续进入高级中断。

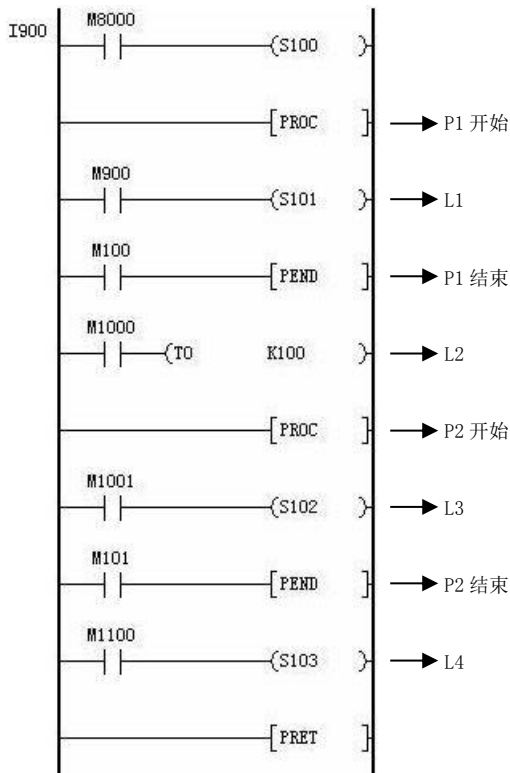
5.5.8.1 高级中断指令[PROC], [PEND], [PRET]

助记符与功能

助记符、名称	功能	回路表示	程序步
PROC 开进程中断	开始中断申请扫描	 操作元: 无	1
PEND 进程结束	结束当前的进程中 断	 操作元: 无	1
PRET 高级中断返 回	整个高级中断结束 返回主程序	 操作元: 无	1

应用与示例

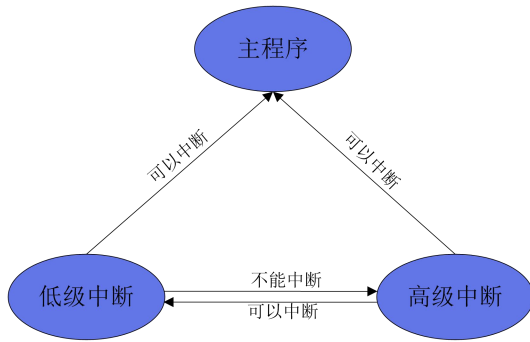
以 I900 为例，对高级中断含进程中断的执行流程进行说明。



- M8064=OFF, M8080=ON 进入 I900
进入后 M8072=ON, M8080=OFF
- 扫描到 P1 结束退出中断
 - ◆ 若 M100=ON, 则下次进程中断到来 (M8072=ON, M8088=OFF) 指向 L2, 程序从 L2 开始执行
 - ◆ 否则, 下次进程中断到来 (M8072=ON, M8088=OFF), 中断指向 L1, 即重复执行 L1~P1 结束
- 当 M8072=ON, M8088=OFF, 系统完成 P1, 从 L2 开始执行后, 系统扫描到 P2 结束退出中断
 - ◆ 若 M101=ON, 则下次进程中断到来 (M8072=ON, M8088=ON) 指向 L4, 程序从 L4 开始执行
 - ◆ 否则, 下次进程中断到来 (M8072=ON, M8088=OFF), 中断指向 L3, 即重复执行 L3~P2 结束
- 系统执行完 P2, 从 L4 进入, 执行到 PRET, M8072=OFF, 高级中断返回

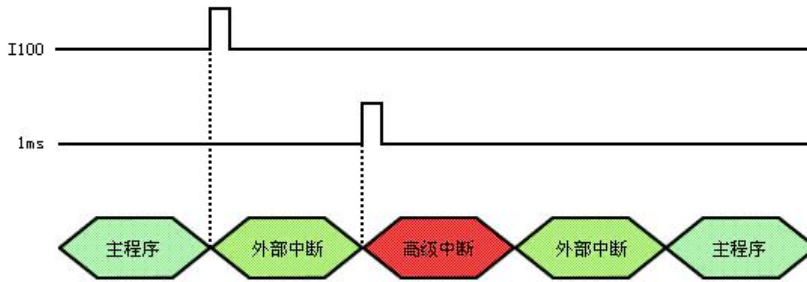
5.5.8.2 高级中断与其他中断的关系

主程序、低级中断和高级中断之间的关系如下图所示，主程序可以被低级中断和高级中断中断，高级中断可以中断低级中断，而低级中断不能中断高级中断。



- 低级中断和高级中断都可以中断主程序，转而执行中断程序，待中断程序执行完后，再返回继续执行主程序。
- 当低级中断正在执行时，高级中断可以中断低级中断，转而执行高级中断，待高级中断执行完后，再返回继续执行低级中断，最后再返回执行主程序。
- 高级中断执行时，低级中断不能中断高级中断。

下面以 I900 高级中断和外部输入中断 I100 为例，说明高级低级中断和主程序之间的时序关系。



5.6 虚拟示波器

HM 系列运动控制器内含一个虚拟示波器，可对寄存器状态的实时变化进行记录，便于用户保存数据或是排查问题。

寄存器地址	参数名称	设定/显示范围	初始推荐设定	单位	生效方式
D8030	触发方式	0~2	0	--	条件生效
D8031	触发单元	0~999	0	--	条件生效
D8032	触发门限	-32768~32767	100	--	条件生效
D8033	采样周期	0~32767	2	ms	条件生效
D8034	采样点数	1~7000	2000	点数	条件生效
D8035	记录起始寄存器序号	1000~7999	1000	--	条件生效
D8036	记录区间起始寄存器序号	1000~7999	1000	--	条件生效
D8037	记录区间终止寄存器序号	1000~7999	4999	--	条件生效
D8038	示波器状态设置/显示	0~4	3	--	立即生效
D8039	当前采样点	采样过程中系统显示当前的已采样点数	0	点数	系统给定
D8040	A 通道工作模式	0, 1, 2, 4	1	--	条件生效
D8041	A 通道采样寄存器序号	0~7999	1000	--	条件生效
D8042	B 通道工作模式	0, 1, 2, 4	1	--	条件生效
D8043	B 通道采样寄存器序号	0~7999	1100	--	条件生效

5.6.1 触发方式

D8030 设定示波器的触发采样模式。

- D8030 = 0，设置示波器处于手动触发状态。

当参数生效后，系统将检测触发单元 D8031 当前数值：

- ◆ D8031 ≠ 0，则立即开始进行数据采样。
- ◆ D8031 = 0，则系统处于等待采样阶段，直到 D8031 ≠ 0 时，立即开始进行数据采样。

所谓“手动”，即 D8031 的数值由用户进行填写，决定什么时候可以启动示波器采样。

D8030 =	D8031 =	示波器开始采样
0	0	×
	非 0, 比如 1	○

- D8030 = 1 或 2，设置示波器处于门限触发状态。
 - ◆ 门限触发状态为触发单元值与触发门限设定值进行比较，根据其大小关系决定是否启动示波器采样。
 - ◇ D8030 = 1，触发单元值 ≥ 触发门限设定值，示波器开始采样。
 - ◇ D8030 = 2，触发单元值 ≤ 触发门限设定值，示波器开始采样。
 - ◆ 触发单元值通过参数触发单元 D8031 进行设定，与手动触发不同的是，门限触发模式下该设置值对应的是寄存器地址单元，而非 1 个常数。
- 例如 D8031 = 12，实际与门限设定值进行比较的是 D12 的当前数值。
- ◆ 触发门限设定值通过参数触发门限 D8032 进行设定。

D8030 =	D8031 = 12	D8032 =	示波器启动采样
1	D12 = 0	100	×
	D12 = 100		○
	D12 = 120		○
2	D12 = 0	100	○
	D12 = 100		○
	D12 = 120		×

- 若示波器已经开始采样，即使触发条件发生改变使启动条件不满足，也不会停止本次的采样过程。

注：若 D8030 填写数值不在 0~2 范围之内，将导致示波器初始化失败。

5.6.2 触发单元

D8031 设定示波器的触发单元。

- 该参数设置的数值意义与参数(触发方式 D8030)的设置相关。

D8030 =	触发方式	D8031 设置方法
0	手动触发	D8031 = 0 时触发关闭，D8031 ≠ 0 时开始触发采样
1 或 2	门限触发	D8031 存放触发寄存器序号，为寄存器地址，而非常数

- 其他参考参数(触发方式 D8030)中描述。
- 若触发方式设置为门限触发，必须保证 D8031 填写数值在 0~999 范围之内。

5.6.3 触发门限

D8032 设定示波器的触发门限。

- 该参数设置的数值意义与参数(触发方式 D8030)和参数(触发单元 D8031)的设置相关。

D8030 =	触发方式	D8031 设置方法
0	手动触发	该参数设定数值与触发采样无关，可随意配置
1 或 2	门限触发	根据 D8031 存放寄存器序号的数值与 D8032 设定值的关系决定是否触发采样

5.6.4 采样周期

D8033 设定示波器采样周期。

- 该参数设置的数值意义与示波器采样周期的换算关系如下所示：

D8033 =	采样周期
0	采样周期 = 0.1 ms，通常采样电流环运行时相关参数时使用。
1	采样周期 = 0.5 ms，通常采样速度环运行时相关参数时使用。
2	采样周期 = 1.0 ms，通常采样位置环运行时相关参数时使用。
N	采样周期 = (N + 1) ms，用户自定义采样周期时使用，N 值必须 > 2。 例如 N = 99，则采样周期 = (99 + 1) = 100.0 ms。

- D8033 填写数值小于 0，采样周期固定为 0.1ms。

5.6.5 采样点数

D8034 设定示波器的采样点数。

- 该参数设置采样过程中所需要的采样次数，每次采样数据按寄存器序号顺序存放在相应寄存器中。存放规则参考参数(A通道工作模式 D8040)中描述。

D8034 与以下所示的参数之间存在着耦合关系，这些参数必须设置正确，否则有可能导致示波器初始化失败。

D8036	记录起始寄存器序号
D8037	记录区间终止寄存器序号
D8040	A通道工作模式
D8042	B通道工作模式

以上参数的配置意义请参考对应的参数描述。

由于用户可自由使用的寄存器序号为 D1000~D7999(D1000 以前资源为系统参数区，一般情况下不允许用户当做自由寄存器使用)，对于以上参数的配置必须注意，存放的采样数据必须在该寄存器区间以内。

参数配置	情况分析	示波器初始化
D8034 = 2000	采样 2000 次，每次采样 2 个字的数据(A/B 通道各占 1 个字)，故采样共需 $2000 \times 2 = 4000$ 个 D 寄存器。 从 D1000 开始存放采样数据，那么采样记录区所占用的寄存器为 D1000~D4999。 由于 $6999 > 4999$ ，故存放没有超限，示波器初始化成功。	○
D8035 = 1000		
D8037 = 6999		
D8040 = 1		
D8042 = 1		
D8034 = 1500	采样 1500 次，每次采样 3 个字的数据(A 通道占 1 个字，B 通道占 2 个字)，故采样共需 $1500 \times 3 = 4500$ 个 D 寄存器。 从 D2000 开始存放采样数据，那么采样记录区所占用的寄存器为 D2000~D6499。 由于 $6999 > 6499$ ，故存放没有超限，示波器初始化成功。	○
D8035 = 2000		
D8037 = 6999		
D8040 = 1		
D8042 = 2		
D8034 = 3000	采样 3000 次，每次采样 2 个字的数据(A 通道占 2 个字，B 通道不采样)，故采样共需 $3000 \times 2 = 6000$ 个 D 寄存器。 从 D1500 开始存放采样数据，那么采样记录区所占用的寄存器为 D1500~D7499。 由于 $6999 < 7499$ ，故存放超限，示波器初始化失败。	×
D8035 = 1500		
D8037 = 6999		
D8040 = 2		
D8042 = 0		
D8034 = 2000	采样 2000 次，每次采样 4 个字的数据(A 通道占 2 个字，B 通道占 2 个字)，故采样共需 $2000 \times 4 = 8000$ 个 D 寄存器。 从 D1000 开始存放采样数据，那么采样记录区最后的寄存器序号应为 8999。用户 D 寄存器最大序号只能到 D7999，但由于 D187 本身设置的不合理，导致没有超限，故示波器仍能初始化成功。 采样数据将覆盖大于 D7999 后的未知区域，可能产生不可预料后果。	○(不推荐)
D8035 = 1000		
D8037 = 10000		
D8040 = 2		
D8042 = 2		

- 注意梯形图中其他工艺流程所用的寄存器不能与示波器采样数据的采样区间重叠，否则可能导致示波器采样数据覆盖用户数据。
- 若 D8034 与其他耦合参数的实际寄存器空间分配超限(如上例描述中的错误情况)，将导致示波器初始化失败。

5.6.6 记录起始寄存器序号

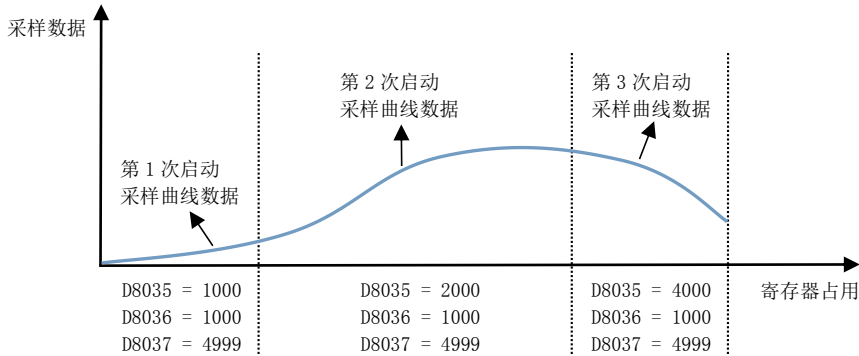
D8035 设定示波器采样数据存放的起始寄存器序号。

- 该参数设置的数值意义为示波器采样数据存放位置，其指向寄存器地址，而非一个常数项。

例如 D8035 = 1000，则采样数据存放寄存器地址从 D1000 开始。

与参数(记录区间起始寄存器序号 D8036)和参数(记录区间终止寄存器序号 D8037)的关系为：

- ◆ D8035 为本次采样实际存放数据的起始地址，D8036 和 D8037 为示波器采样数据用户规划存放的区间地址。必须保证 D8035 指向的寄存器序号在 D8036 和 D8037 之间。
- ◆ 一个完整的曲线数据采样可能分为多次启用示波器采样完成，D8036 规划为整个曲线数据的区间起始点，D8037 规划为整个曲线数据的区间结束点。



- ◆ 若数据曲线采样仅单次执行，可以将 D8035 的设定数值与 D8036 相等，此也为通常情况下的推荐做法。
- 若 D8035 填写的寄存器序号在 1000~7999 范围之外，可能产生不可预料的后果。
- 必须保证满足 $D8037 \geq D8035 \geq D8036$ ，否则将导致示波器初始化失败。

5.6.7 记录区间起始寄存器序号

D8036 设定示波器记录区间起始寄存器序号。

- 该参数设置的数值意义是规划示波器采样数据存放的起始位置，其指向寄存器地址，而非一个常数项。
例如 D8036 = 1000，则规划采样数据存放在 D1000 以上的寄存器区域。
- 该参数与(起始寄存器序号 D8035)和参数(记录区间终止寄存器序号 D8037)之间的关系可在参数 D8035 中查看。
- 若 D8036 填写的寄存器序号在 1000~7999 范围之外，可能产生不可预料的后果。
- 必须保证满足 $D8037 \geq D8035 \geq D8036$ ，否则将导致示波器初始化失败。

5.6.8 记录区间终止寄存器序号

D8037 设定示波器记录区间终止寄存器序号。

- 该参数设置的数值意义是规划示波器采样数据存放的结束位置，其指向寄存器地址，而非一个常数项。
例如 D8037 = 4999，则本次采样数据存放在 D4999 以下的寄存器区域。
- 本次示波器采样的记录终止寄存器序号可通过以下参数算出：

D8034	采样点数
D8035	记录起始寄存器序号
D8040	A 通道工作模式
D8042	B 通道工作模式

记录终止寄存器序号 =

记录起始寄存器序号 + 采样点数 × (A 通道信号数据类型所占字数 + B 通道信号数据类型所占字数) - 1

例如 D8034 = 2000, D8035 = 1000, D8040 = 1, D8042 = 1,

则记录终止寄存器序号 = $1000 + 2000 \times (1 + 1) - 1 = 4999$ ，即本次采样终止寄存器为 D4999。

D8037 的设定数值 \geq 计算出的记录终止寄存器序号即可，若只执行单次采样，推荐设置两者相等。

- 若 D8037 填写的寄存器序号在 1000~7999 范围之外，可能产生不可预料的后果。
- 必须保证满足 $D8037 \geq D8035 \geq D8036$ ，否则将导致示波器初始化失败。

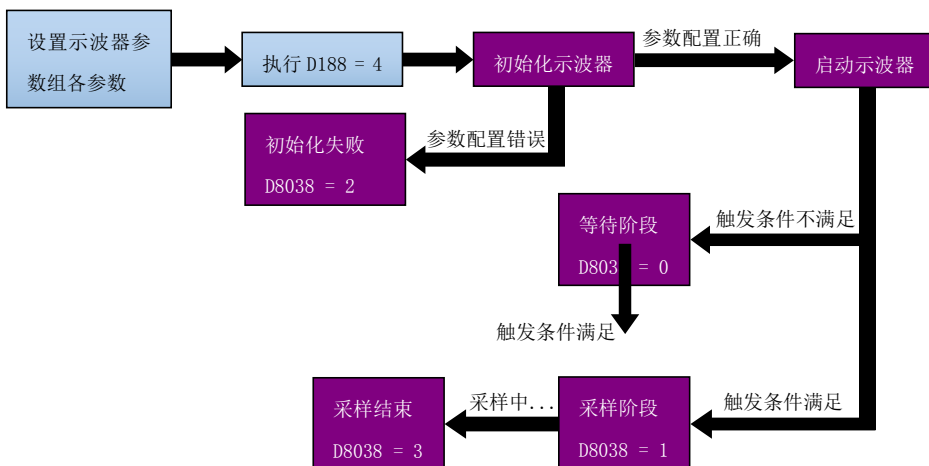
5.6.9 示波器状态设置/显示

D8038 是示波器与用户的交互接口。用户可通过设置该参数申请启动示波器，也可通过其查看当前示波器的状态。

- D8038 为用户设置参数时，仅作为申请启动示波器时使用。

D8038 =	设置/显示值	示波器状态
0	系统给定	启动示波器后，若触发条件不满足，D8038 会被系统置 0，此时示波器处于等待采样阶段。 当触发条件满足后，D8038 会被系统置 1。
1	系统给定	启动示波器后，若触发条件满足，D8038 会被系统置 1，此时示波器处于采样阶段。
2	系统给定	示波器初始化失败，即示波器相关参数设置出错所导致。
3	系统给定	当示波器采样结束，D8038 会被系统置 3。通过观测此数值判断示波器是否已经结束采样。
4	用户设置	示波器申请初始化。 ● 若示波器相关参数设置正确，系统启动示波器。根据参数设置的触发条件决定是否开始采样。 ● 若示波器相关参数设置不正确，系统给定 D8038 = 2。

- 一个采样流程中，D8038 数值的改变顺序如下所示：



- 示波器参数可能出错或不推荐的设置情况参考第 5.6 章节各参数的下划线部分的描述。

5.6.10 当前采样点

D8039 是示波器实时显示当前已采样点数的数值。

- 用户申请示波器初始化(D8038 = 4)后，D8039 由系统赋初值 0。

若触发条件满足开始采样，D8039 的数值随着采样周期变化而递增，直至采样结束。

例如参数(采样点数 D8034) = 2000，参数(采样周期 D8033) = 2。

采样开始后 D8039 数值清零，并 1ms 递增 1 次，直至采样结束。其变化规律为 0, 1, 2, 3, ..., 1998, 1999, 2000。

5.6.11 通道工作模式

示波器可同时支持 2 个通道的数据采集，其中 D8040 为设定示波器 A 通道的工作模式，D8041 为设定示波器 B 通道的工作模式。

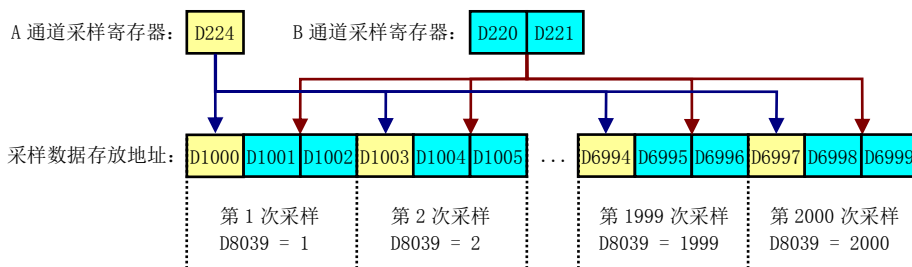
对于 A 通道，其对应的采样寄存器序号通过参数(A 通道采样寄存器序号 D8041)进行设定。

对于 B 通道，其对应的采样寄存器序号通过参数 (B 通道采样寄存器序号 D8043) 进行设定。

D8040/D8042=	A/B 通道工作模式	A/B 工作模式描述	示例寄存器地址
0	A/B 通道关闭	A/B 通道不采样	--
1	A/B 通道字信号	采样 D8041/D8043 所指向的寄存器的字数据	D11, D12, D224, D234, D260...
2	A/B 通道双字信号	采样 D8041/D8043 所指向的寄存器的双字数据	D226, D228, D236, D238...
4	A/B 通道四字信号	采样 D8041/D8043 所指向的寄存器的四字数据	D14, D220, D230... * 以上也可作为双字信号进行采样。

● 示波器采样数据为交替存放规则，以如下示例进行说明。假定示波器的采样参数设置如下：

- ◆ (采样点数 D8034) = 2000
- ◆ (记录起始寄存器序号 D8035) = 1000
- ◆ (A 通道工作模式 D8040) = 1, (A 通道寄存器序号 D8041) = 224
- ◆ (B 通道工作模式 D8042) = 2, (B 通道寄存器序号 D8043) = 220



● 示波器通道的工作模式可填 3 或大于 4 的设定值，只要满足采样结束地址不会超过 (记录区间终止寄存器序号 D8037) 的设定值，示波器仍然可以采样。

例如 D8040 = 10，每次采样将 A 通道采样寄存器序号开始的连续 10 个 D 单元数值进行保存。

但此时所采样的数据为非规则的变量类型，难以利用软件进行分析，故不推荐这样设置。

5.6.12 通道采样寄存器序号

D8041 为示波器 A 通道采样寄存器序号的设定值，D8043 为示波器 B 通道采样寄存器序号的设定值。

● 该参数设置的数值意义是 A/B 通道每次采样的位置，其指向寄存器地址，而非一个常数项。

例如 D8041 = 234，则 A 通道每次采样从 D234 寄存器开始，采样字数通过参数 (A 通道工作模式 D8040) 进行设定。

D8043 = 220，则 A 通道每次采样从 D220 寄存器开始，采样字数通过参数 (B 通道工作模式 D8042) 进行设定。

● 当 D8040 设置为 0 时，即代表 A 通道不参与本次采样，此时 D8041 不参与系统运算，用户可随意配置。
当 D8042 设置为 0 时，即代表 B 通道不参与本次采样，此时 D8043 不参与系统运算，用户可随意配置。

5.7 特殊辅助继电器[M], 特殊数据寄存器[D]

本章节就 PLC 内置的特殊辅助继电器 (M8000~M8255), 特殊数据寄存器 (D8000~D8255) 的功能及作用进行列表说明。

对于未使用或是未作记载和定义的软元件编号, 请勿在程序内进行驱动和写入操作。否则可能出现意料之外的错误。

5.7.1 特殊辅助继电器[M]功能一览表

软元件编号	软元件功能
M8000	PLC 运行 ON, 其他状态 OFF
M8001	PLC 运行 OFF, 其他状态 ON
M8002	PLC 运行首次梯形图扫描周期 ON, 其余扫描周期 OFF
M8003	PLC 运行首次梯形图扫描周期 OFF, 其余扫描周期 ON
M8004	设置 PLC 运行状态 (ON-PLC 运行, OFF-PLC 停止), 操控该位必须满足以下前提条件: ● D8004 = HFACO ● M8005 = OFF
M8005	远程控制设置位 (ON-物理控制, OFF-远程控制)
M8006	梯形图运行错误 ON
M8007	外部 RAM 检测 (ON-外部 RAM 运行正常, OFF-未检测到外部 RAM)
M8008	掉电检测状态 (ON-上电后发生掉电, OFF-上电后未发生掉电)
M8009	暂不开放使用
M8010	暂不开放使用
M8011	暂不开放使用
M8012	暂不开放使用
M8013	暂不开放使用
M8014	暂不开放使用
M8015	暂不开放使用
M8016	暂不开放使用
M8017	暂不开放使用
M8018	暂不开放使用
M8019	暂不开放使用
M8020	零点标志, 加减运算结果为 0 时 ON(保留)
M8021	借位标志, 减法运算结果小于负最大值时 ON(保留)
M8022	进位标志, 进位循环, 移位指令运算使用
M8023	暂不开放使用
M8024	暂不开放使用
M8025	暂不开放使用
M8026	暂不开放使用
M8027	暂不开放使用
M8028	暂不开放使用
M8029	暂不开放使用

软元件编号	软元件功能
M8030	暂不开放使用
M8031	暂不开放使用
M8032	输入中断 I000 申请
M8033	输入中断 I001 申请
M8034	输入中断 I100 申请
M8035	输入中断 I101 申请
M8036	输入中断 I200 申请
M8037	输入中断 I201 申请
M8038	输入中断 I300 申请
M8039	输入中断 I301 申请
M8040	定时中断 I6□□申请
M8041	定时中断 I7□□申请
M8042	定时中断 I8□□申请
M8043	定时中断 I010 申请
M8044	软中断 I020 申请
M8045	软中断 I030 申请
M8046	软中断 I040 申请
M8047	软中断 I050 申请
M8048	输入中断 I000 禁止
M8049	输入中断 I001 禁止
M8050	输入中断 I100 禁止
M8051	输入中断 I101 禁止
M8052	输入中断 I200 禁止
M8053	输入中断 I201 禁止
M8054	输入中断 I300 禁止
M8055	输入中断 I301 禁止
M8056	定时中断 I6□□禁止
M8057	定时中断 I7□□禁止
M8058	定时中断 I8□□禁止
M8059	软中断 I010 禁止
M8060	软中断 I020 禁止
M8061	软中断 I030 禁止
M8062	软中断 I040 禁止
M8063	软中断 I050 禁止
M8064	禁止高级中断 I900 (OFF:DISABLE, ON:ENABLE)
M8065	禁止高级中断 I901 (OFF:DISABLE, ON:ENABLE)
M8066	禁止高级中断 I902 (OFF:DISABLE, ON:ENABLE)
M8067	禁止高级中断 I903 (OFF:DISABLE, ON:ENABLE)
M8068	禁止高级中断 I904 (OFF:DISABLE, ON:ENABLE)
M8069	禁止高级中断 I905 (OFF:DISABLE, ON:ENABLE)

软元件编号	软元件功能
M8070	禁止高级中断 I906 (OFF:DISABLE, ON:ENABLE)
M8071	禁止高级中断 I907 (OFF:DISABLE, ON:ENABLE)
M8072	高级中断 I900 激活 (ON:ACTIVE, OFF:END)
M8073	高级中断 I901 激活 (ON:ACTIVE, OFF:END)
M8074	高级中断 I902 激活 (ON:ACTIVE, OFF:END)
M8075	高级中断 I903 激活 (ON:ACTIVE, OFF:END)
M8076	高级中断 I904 激活 (ON:ACTIVE, OFF:END)
M8077	高级中断 I905 激活 (ON:ACTIVE, OFF:END)
M8078	高级中断 I906 激活 (ON:ACTIVE, OFF:END)
M8079	高级中断 I907 激活 (ON:ACTIVE, OFF:END)
M8080	高级中断 I900 申请 (ON:ASK, OFF:NO)
M8081	高级中断 I901 申请 (ON:ASK, OFF:NO)
M8082	高级中断 I902 申请 (ON:ASK, OFF:NO)
M8083	高级中断 I903 申请 (ON:ASK, OFF:NO)
M8084	高级中断 I904 申请 (ON:ASK, OFF:NO)
M8085	高级中断 I905 申请 (ON:ASK, OFF:NO)
M8086	高级中断 I906 申请 (ON:ASK, OFF:NO)
M8087	高级中断 I907 申请 (ON:ASK, OFF:NO)
M8088	高级中断 0 进程申请字 (1-ASK 0-NO)
M8089	高级中断 1 进程申请字 (1-ASK 0-NO)
M8090	高级中断 2 进程申请字 (1-ASK 0-NO)
M8091	高级中断 3 进程申请字 (1-ASK 0-NO)
M8092	高级中断 4 进程申请字 (1-ASK 0-NO)
M8093	高级中断 5 进程申请字 (1-ASK 0-NO)
M8094	高级中断 6 进程申请字 (1-ASK 0-NO)
M8095	高级中断 7 进程申请字 (1-ASK 0-NO)
M8096	运动指令开关量输入(替代 X000 输入)
M8097	运动指令开关量输入(替代 X001 输入)
M8098	运动指令开关量输入(替代 X002 输入)
M8099	运动指令开关量输入(替代 X003 输入)
M8100	运动指令开关量输入(替代 X004 输入)
M8101	运动指令开关量输入(替代 X005 输入)
M8102	运动指令开关量输入(替代 X006 输入)
M8103	运动指令开关量输入(替代 X007 输入)
M8104	运动指令开关量输入(替代 X010 输入)
M8105	运动指令开关量输入(替代 X011 输入)
M8106	运动指令开关量输入(替代 X012 输入)
M8107	运动指令开关量输入(替代 X013 输入)
M8108	运动指令开关量输入 (替代 X014 输入)
M8109	运动指令开关量输入 (替代 X015 输入)

软元件编号	软元件功能
M8110	运动指令开关量输入（替代 X016 输入）
M8111	运动指令开关量输入（替代 X017 输入）
M8112	暂不开放使用
M8113	暂不开放使用
M8114	暂不开放使用
M8115	暂不开放使用
M8116	暂不开放使用
M8117	暂不开放使用
M8118	暂不开放使用
M8119	暂不开放使用
M8120	暂不开放使用
M8121	暂不开放使用
M8122	暂不开放使用
M8123	暂不开放使用
M8124	暂不开放使用
M8125	暂不开放使用
M8126	暂不开放使用
M8127	暂不开放使用
M8128	暂不开放使用
M8129	暂不开放使用
M8130	暂不开放使用
M8131	暂不开放使用
M8132	暂不开放使用
M8133	暂不开放使用
M8134	暂不开放使用
M8135	暂不开放使用
M8136	暂不开放使用
M8137	暂不开放使用
M8138	暂不开放使用
M8139	暂不开放使用
M8140	暂不开放使用
M8141	暂不开放使用
M8142	暂不开放使用
M8143	暂不开放使用
M8144	暂不开放使用
M8145	暂不开放使用
M8146	暂不开放使用
M8147	暂不开放使用
M8148	暂不开放使用
M8149	暂不开放使用

软元件编号	软元件功能
M8150	暂不开放使用
M8151	暂不开放使用
M8152	暂不开放使用
M8153	暂不开放使用
M8154	暂不开放使用
M8155	暂不开放使用
M8156	暂不开放使用
M8157	暂不开放使用
M8158	暂不开放使用
M8159	暂不开放使用
M8160	VX 虚拟 X14
M8161	VX 虚拟 X15
M8162	VX 虚拟 X16
M8163	VX 虚拟 X17
M8164	暂不开放使用
M8165	暂不开放使用
M8166	暂不开放使用
M8167	暂不开放使用
M8168	暂不开放使用
M8169	暂不开放使用
M8170	暂不开放使用
M8171	暂不开放使用
M8172	暂不开放使用
M8173	暂不开放使用
M8174	暂不开放使用
M8175	暂不开放使用
M8176	暂不开放使用
M8177	暂不开放使用
M8178	暂不开放使用
M8179	暂不开放使用
M8180	暂不开放使用
M8181	暂不开放使用
M8182	暂不开放使用
M8183	暂不开放使用
M8184	暂不开放使用
M8185	暂不开放使用
M8186	暂不开放使用
M8187	暂不开放使用
M8188	暂不开放使用
M8189	暂不开放使用

软元件编号	软元件功能
M8190	暂不开放使用
M8191	暂不开放使用
M8192	暂不开放使用
M8193	暂不开放使用
M8194	暂不开放使用
M8195	暂不开放使用
M8196	暂不开放使用
M8197	暂不开放使用
M8198	暂不开放使用
M8199	暂不开放使用
M8200	内部计数器 C200 方向控制位
M8201	内部计数器 C201 方向控制位
M8202	内部计数器 C202 方向控制位
M8203	内部计数器 C203 方向控制位
M8204	内部计数器 C204 方向控制位
M8205	内部计数器 C205 方向控制位
M8206	内部计数器 C206 方向控制位
M8207	内部计数器 C207 方向控制位
M8208	内部计数器 C208 方向控制位
M8209	内部计数器 C209 方向控制位
M8210	内部计数器 C210 方向控制位
M8211	内部计数器 C211 方向控制位
M8212	内部计数器 C212 方向控制位
M8213	内部计数器 C213 方向控制位
M8214	内部计数器 C214 方向控制位
M8215	内部计数器 C215 方向控制位
M8216	内部计数器 C216 方向控制位
M8217	内部计数器 C217 方向控制位
M8218	内部计数器 C218 方向控制位
M8219	内部计数器 C219 方向控制位
M8220	内部计数器 C220 方向控制位
M8221	内部计数器 C221 方向控制位
M8222	内部计数器 C222 方向控制位
M8223	内部计数器 C223 方向控制位
M8224	内部计数器 C224 方向控制位
M8225	内部计数器 C225 方向控制位
M8226	内部计数器 C226 方向控制位
M8227	内部计数器 C227 方向控制位
M8228	内部计数器 C228 方向控制位
M8229	内部计数器 C229 方向控制位

软元件编号	软元件功能
M8230	内部计数器 C230 方向控制位
M8231	内部计数器 C231 方向控制位
M8232	内部计数器 C232 方向控制位
M8233	内部计数器 C233 方向控制位
M8234	内部计数器 C234 方向控制位
M8235	内部计数器 C235 方向控制位
M8236	暂不开放使用
M8237	内部计数器 C237 方向控制位
M8238	暂不开放使用
M8239	内部计数器 C239 方向控制位
M8240	内部计数器 C240 方向控制位
M8241	内部计数器 C241 方向控制位
M8242	内部计数器 C242 方向控制位
M8243	暂不开放使用
M8244	暂不开放使用
M8245	暂不开放使用
M8246	暂不开放使用
M8247	暂不开放使用
M8248	暂不开放使用
M8249	暂不开放使用
M8250	暂不开放使用
M8251	暂不开放使用
M8252	暂不开放使用
M8253	暂不开放使用
M8254	暂不开放使用
M8255	暂不开放使用

5.7.2 特殊辅助寄存器[D]功能一览表

软元件编号	软元件功能
D8000	扫描周期定时监视器限值，单位为 ms。当梯形图扫描周期超过该值时梯形图报错。
D8001	指令系列号
D8002	程序容量
D8003	驱动系列号
D8004	设置 PLC 远程控制(设置为 HFAC0，则 PLC 处于远程控制模式)
D8005	文件写入操作口令(内部使用)
D8006	梯形图出错时的错误步号
D8007	50 微秒定时器
D8008	毫秒滴答定时器低字
D8009	毫秒滴答定时器高字
D8010	PLC 运行扫描周期实时值
D8011	PLC 运行扫描周期最小值
D8012	PLC 运行扫描周期最大值
D8013	时钟秒数据(保留)
D8014	时钟分数据(保留)
D8015	时钟时数据(保留)
D8016	时钟日数据(保留)
D8017	时钟月数据(保留)
D8018	时钟年数据(保留)
D8019	时钟星期数据(保留)
D8020	数据掉电保护方式，软元件的掉电保护参考章节 5.1.11。 ● =1111 不带停电保护 ● =2222 带停电保护
D8021	掉电电压 AD(实测值)
D8022	暂不开放使用
D8023	暂不开放使用
D8024	暂不开放使用
D8025	暂不开放使用
D8026	主控芯片端口 A 组状态
D8027	暂不开放使用
D8028	主控芯片端口 B 组状态
D8029	暂不开放使用
D8030	虚拟示波器触发方式
D8031	虚拟示波器触发单元
D8032	虚拟示波器触发门限
D8033	虚拟示波器采样周期
D8034	虚拟示波器采样点数
D8035	虚拟示波器记录起始寄存器序号
D8036	虚拟示波器记录区间起始寄存器序号
D8037	虚拟示波器记录区间终止寄存器序号

软件编号	软件功能
D8038	虚拟示波器状态设置/显示
D8039	虚拟示波器当前采样点
D8040	虚拟示波器 A 通道工作模式
D8041	虚拟示波器 A 通道采样寄存器序号
D8042	虚拟示波器 B 通道工作模式
D8043	虚拟示波器 B 通道采样寄存器序号
D8044	50us 时间片内任务时间位置, 数值越大则任务执行时间越短(满值 3999)
D8045	定时中断最大占用时间
D8046	暂不开放使用
D8047	暂不开放使用
D8048	暂不开放使用
D8049	暂不开放使用
D8050	暂不开放使用
D8051	暂不开放使用
D8052	暂不开放使用
D8053	暂不开放使用
D8054	暂不开放使用
D8055	暂不开放使用
D8056	暂不开放使用
D8057	暂不开放使用
D8058	暂不开放使用
D8059	暂不开放使用
D8060	暂不开放使用
D8061	暂不开放使用
D8062	暂不开放使用
D8063	暂不开放使用
D8064	暂不开放使用
D8065	暂不开放使用
D8066	暂不开放使用
D8067	暂不开放使用
D8068	暂不开放使用
D8069	暂不开放使用
D8070	暂不开放使用
D8071	暂不开放使用
D8072	暂不开放使用
D8073	暂不开放使用
D8074	暂不开放使用
D8075	暂不开放使用
D8076	暂不开放使用
D8077	暂不开放使用

软元件编号	软元件功能
D8078	暂不开放使用
D8079	暂不开放使用
D8080	暂不开放使用
D8081	暂不开放使用
D8082	暂不开放使用
D8083	暂不开放使用
D8084	暂不开放使用
D8085	暂不开放使用
D8086	暂不开放使用
D8087	暂不开放使用
D8088	暂不开放使用
D8089	暂不开放使用
D8090	暂不开放使用
D8091	暂不开放使用
D8092	暂不开放使用
D8093	暂不开放使用
D8094	暂不开放使用
D8095	暂不开放使用
D8096	暂不开放使用
D8097	暂不开放使用
D8098	暂不开放使用
D8099	暂不开放使用
D8100	V0 指针寄存器
D8101	V1 指针寄存器
D8102	V2 指针寄存器
D8103	V3 指针寄存器
D8104	V4 指针寄存器
D8105	V5 指针寄存器
D8106	V6 指针寄存器
D8107	V7 指针寄存器
D8108	Z0 指针寄存器
D8109	Z1 指针寄存器
D8110	Z2 指针寄存器
D8111	Z3 指针寄存器
D8112	Z4 指针寄存器
D8113	Z5 指针寄存器
D8114	Z6 指针寄存器
D8115	Z7 指针寄存器
D8116	暂不开放使用
D8117	暂不开放使用

软元件编号	软元件功能
D8118	暂不开放使用
D8119	暂不开放使用
D8120	UART0 协议控制字
D8121	UART0 字节控制字
D8122	UART0 占用 D 寄存器起始单元 (默认 7000)
D8123	UART1 协议控制字
D8124	UART1 字节控制字
D8125	UART1 占用 D 寄存器起始单元 (默认 7020)
D8126	暂不开放使用
D8127	暂不开放使用
D8128	暂不开放使用
D8129	扩展占用 D 起始单元(默认 7100)
D8130	暂不开放使用
D8131	暂不开放使用
D8132	暂不开放使用
D8133	暂不开放使用
D8134	暂不开放使用
D8135	暂不开放使用
D8136	暂不开放使用
D8137	暂不开放使用
D8138	暂不开放使用
D8139	暂不开放使用
D8140	暂不开放使用
D8141	暂不开放使用
D8142	暂不开放使用
D8143	暂不开放使用
D8144	暂不开放使用
D8145	暂不开放使用
D8146	暂不开放使用
D8147	暂不开放使用
D8148	暂不开放使用
D8149	暂不开放使用
D8150	暂不开放使用
D8151	暂不开放使用
D8152	暂不开放使用
D8153	暂不开放使用
D8154	暂不开放使用
D8155	暂不开放使用
D8156	暂不开放使用
D8157	暂不开放使用

软元件编号	软元件功能
D8158	暂不开放使用
D8159	暂不开放使用
D8160	X00 上升沿计数 (5KHz)
D8161	X01 上升沿计数 (5KHz)
D8162	X02 上升沿计数 (5KHz)
D8163	X03 上升沿计数 (5KHz)
D8164	X04 上升沿计数 (5KHz)
D8165	X05 上升沿计数 (5KHz)
D8166	X06 上升沿计数 (5KHz)
D8167	X07 上升沿计数 (5KHz)
D8168	暂不开放使用
D8169	暂不开放使用
D8170	暂不开放使用
D8171	暂不开放使用
D8172	暂不开放使用
D8173	暂不开放使用
D8174	暂不开放使用
D8175	暂不开放使用
D8176	暂不开放使用
D8177	暂不开放使用
D8178	暂不开放使用
D8179	暂不开放使用
D8180	Y0 输出模式 0-PLC 输出;1-保留;2-脉冲;3-保留
D8181	Y1 输出模式 0-PLC 输出;1-保留;2-脉冲;3-保留
D8182	Y2 输出模式 0-PLC 输出;1-保留;2-脉冲;3-保留
D8183	Y3 输出模式 0-PLC 输出;1-保留;2-脉冲;3-保留
D8184	Y4 输出模式 0-PLC 输出;1-保留;2-脉冲;3-保留
D8185	Y5 输出模式 0-PLC 输出;1-保留;2-脉冲;3-保留
D8186	暂不开放使用
D8187	暂不开放使用
D8188	Y10 输出模式 0-PLC 输出;1-高速事件输出;2-无;3-轴方向位
D8189	Y11 输出模式 0-PLC 输出;1-高速事件输出;2-无;3-轴方向位
D8190	Y12 输出模式 0-PLC 输出;1-高速事件输出;2-无;3-轴方向位
D8191	Y13 输出模式 0-PLC 输出;1-高速事件输出;2-无;3-轴方向位
D8192	Y14 输出模式 0-PLC 输出;1-高速事件输出;2-无;3-轴方向位
D8193	Y15 输出模式 0-PLC 输出;1-高速事件输出;2-无;3-轴方向位
D8194	Y16 输出模式 0-PLC 输出;1-保留;2-保留;3-保留
D8195	Y17 输出模式 0-PLC 输出;1-保留;2-保留;3-保留
D8196	暂不开放使用
D8197	暂不开放使用

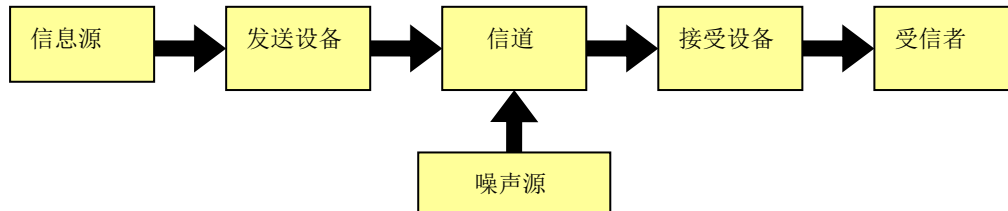
软元件编号	软元件功能
D8198	暂不开放使用
D8199	暂不开放使用
D8200	内部函数号
D8201	内部入口参数地址
D8202	内部出口参数地址
D8203	暂不开放使用
D8204	暂不开放使用
D8205	暂不开放使用
D8206	暂不开放使用
D8207	暂不开放使用
D8208	暂不开放使用
D8209	暂不开放使用
D8210	暂不开放使用
D8211	暂不开放使用
D8212	暂不开放使用
D8213	暂不开放使用
D8214	暂不开放使用
D8215	暂不开放使用
D8216	暂不开放使用
D8217	暂不开放使用
D8218	暂不开放使用
D8219	暂不开放使用
D8220	暂不开放使用
D8221	暂不开放使用
D8222	暂不开放使用
D8223	暂不开放使用
D8224	暂不开放使用
D8225	暂不开放使用
D8226	暂不开放使用
D8227	暂不开放使用
D8228	暂不开放使用
D8229	暂不开放使用
D8230	暂不开放使用
D8231	暂不开放使用
D8232	暂不开放使用
D8233	暂不开放使用
D8234	暂不开放使用
D8235	暂不开放使用
D8236	暂不开放使用
D8237	暂不开放使用

软元件编号	软元件功能
D8238	暂不开放使用
D8239	暂不开放使用
D8240	暂不开放使用
D8241	暂不开放使用
D8242	暂不开放使用
D8243	暂不开放使用
D8244	暂不开放使用
D8245	暂不开放使用
D8246	暂不开放使用
D8247	暂不开放使用
D8248	暂不开放使用
D8249	暂不开放使用
D8250	暂不开放使用
D8251	暂不开放使用
D8252	暂不开放使用
D8253	驱动版本号
D8254	<p>加密控制。PLC 停止状态时写 D8254 为以下数值，并下载一个与运动控制器内相异的梯形图后，系统即进行加密操作。</p> <ul style="list-style-type: none"> ● D8254 = K12345 禁读 ● D8254 = K-11215 禁读禁写 <p>* 加密为不可逆操作，需谨慎操作，后果自负。</p>
D8255	内核版本号

第六章 通信功能

6.1 通信原理

通信的目的是传递消息中所包含的信息，通信系统的一般模型如下图所示。



- 信息源：把各种消息转换成原始电信号，如麦克风。信源可分为模拟信源和数字信源。
- 发送设备：产生适合于在信道中传输的信号。
- 信道：将来自发送设备的信号传送到接收端的物理媒质，分为有线信道和无线信道两大类。
- 噪声源：集中表示分布于通信系统中各处的噪声。
- 接收设备：从受到减损的接收信号中正确恢复出原始电信号。
- 受信者：把原始电信号还原成相应的消息，如扬声器等。

6.1.1 通信方式

- 通信方式分为单工通信、半双工通信和全双工通信。
 - ◆ 单工通信：数据传输是单向的。
 - ◆ 半双工通信：可在一个信号载体的两个方向上传输，但不能同时传送。
 - ◆ 全双工通信：可在信号载体的两个方向同时传输。
- 信息传输方式分为并行传输和串行传输。
 - ◆ 并行传输：将代表信息的数字信号码元序列以成组方式在两条或两条以上的并行信道上同时传输。
 - 优点是节省传输时间，速度快，不需要字符同步措施。
 - 缺点是需要多条通信线路，成本高。
 - ◆ 串行传输：将数字信号码元序列以串行方式一个码元接一个码元地在一条信道上传输。
 - 优点是只需一条通信信道，节省线路铺设费用。
 - 缺点是速度慢，需要外加码组或字符同步措施。
- 串口通信按位(bit)发送和接收字节。串口通信最重要的参数是波特率、数据位、停止位和奇偶校验。对于两个进行通信的端口，这些参数必须匹配。
 - ◆ 波特率

波特率指数据信号对载波的调制速率，它用单位时间内载波调制状态改变次数来表示，其单位为波特(Baud)。波特率与比特率的关系为：比特率 = 波特率 × 单个调制状态对应的二进制位数。

两相调制(单个调制状态对应1个二进制位)的比特率等于波特率；四相调制(单个调制状态对应2个二进制位)的比特率为波特率的两倍；八相调制(单个调制状态对应3个二进制位)的比特率为波特率的三倍；依次类推。

通常情况下，波特率可以理解为1秒钟传输的比特位数。例如9600波特率，则1秒钟可以通信9600个比特位。
 - ◆ 数据位

数据位是衡量通信中实际数据位的参数。当设备发送一个信息包，实际的数据不会是8位的，标准

的值是 5、7 和 8 位，如何设置取决于想传送的信息。例如，标准的 ASCII 码是 0~127 (7 位)。扩展的 ASCII 码是 0~255 (8 位)。如果数据使用简单的文本 (标准 ASCII 码)，那么每个数据包使用 7 位数据。每个包是指一个字节，包括开始/停止位，数据位和奇偶校验位。

◆ 停止位

停止位用于表示单个包的最后一位。常用的值为 1, 1.5 和 2 位。由于数据是在传输线上定时的，并且每一个设备有其自己的时钟，很可能在通信中两台设备间出现了微小的不同步。

因此停止位不仅仅是表示传输的结束，并且提供校正时钟同步的机会。适用于停止位的位数越多，不同时钟同步的容忍程度越大，但是数据传输率同时也越慢。

◆ 奇偶校验位

奇偶校验位是在串口通信中一种简单的检错方式。有四种检错方式:偶、奇、高和低。当然没有校验位也是可以的。对于偶和奇校验的情况，串口会设置校验位 (数据位后面的一位)，用一个值确保传输的数据有偶个或者奇个逻辑高位。

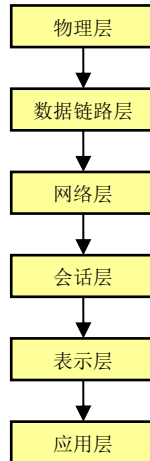
例如，如果数据是 011，那么对于偶校验，校验位为 0，保证逻辑高的位数是偶数个。如果是奇校验，校验位为 1，这样就有 3 个逻辑高位。这样使得接收设备能够知道一个位的状态，有机会判断是否有噪声干扰了通信或者是否传输和接收数据是否不同步。

6.1.2 网络系统结构

网络系统结构——七层参考模型，叫做开放系统互连模型 (Open System Interconnection)。

七层分别是：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

OSI 七层模型通过七个层次化的结构模型使不同的系统不同的网络之间实现可靠的通讯，因此其最主要的功能使就是帮助不同类型的主机实现数据传输。



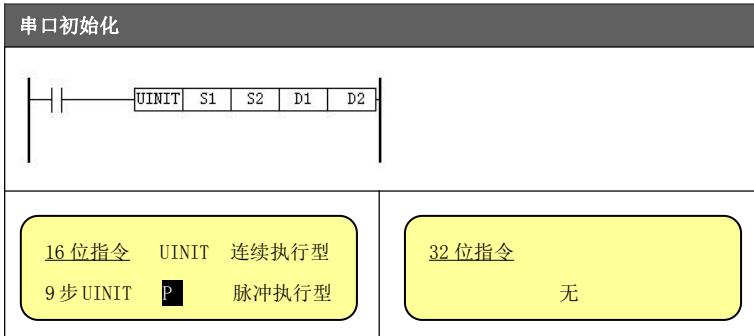
OSI 七层模型介绍

- 第一层到第三层属于 OSI 参考模型的低三层，负责创建网络通信连接的链路；
- 第四层到第七层为 OSI 参考模型的高四层，具体负责端到端的数据通信。
- 每层完成一定的功能，每层都直接为其上层提供服务，并且所有层次都互相支持，而网络通信则可以。
- 自上而下（在发送端）或者自下而上（在接收端）双向进行。
- 并不是每一通信都需要经过 OSI 的全部七层，有的甚至只需要双方对应的某一层即可。
- 物理接口之间的转接，以及中继器与中继器之间的连接就只需在物理层中进行。
- 双方的通信是在对等层次上进行的，不能在不对称层次上进行通信。

OSI 七层模型	简要说明
物理层	<p>OSI 的第一层，它虽然处于最底层，却是整个开放互联模型系统的基础。物理层为设备之间的数据通信提供传输媒体及互连设备，为数据传输提供可靠的环境。物理层的主要功能为数据端设备提供传送数据的通路，数据通路可以是一个物理媒体，也可以是多个物理媒体连接而成。一次完整的数据传输，包括激活物理连接，传送数据，终止物理连接。</p> <p>物理层要形成适合数据传输需要的实体，为数据传送服务。一是要保证数据能在其上正确通过，二是要提供足够的带宽(带宽是指每秒钟内能通过的比特(BIT)数)，以减少信道上的拥塞。传输数据的方式能满足点到点，一点到多点，串行或并行，半双工或全双工，同步或异步传输的需要。</p>
数据链路层	<p>OSI 模型的第二层，它控制网络层与物理层之间的通信。它的主要功能是如何在不可靠的物理线路上进行数据的可靠传递。为了保证传输，从网络层接收到的数据被分割成特定的可被物理层传输的帧。</p> <p>帧是用来移动数据的结构包，它不仅包括原始数据，还包括发送方和接收方的网络地址以及纠错和控制信息。其中的地址确定了帧将发送到何处，而纠错和控制信息则确保帧无差错到达。如果在传送数据时，接收点检测到所传数据中有差错，就要通知发送方重发这一帧。</p>
网络层	<p>OSI 模型的第三层，其主要功能是将网络地址翻译成对应的物理地址，并决定如何将数据从发送方路由到接收方。</p> <p>网络层通过综合考虑发送优先权、网络拥塞程度、服务质量以及可选路由的花费来决定从一个网络中节点到另一个网络中节点的最佳路径。由于网络层处理路由，而路由器因为即连接网络各段，并智能指导数据传送，属于网络层。在网络中，“路由”是基于编址方案、使用模式以及可达性来指引数据的发送。</p>
传输层	<p>OSI 模型中最重要的一层。传输协议同时进行流量控制或是基于接收方可接收数据的快慢程度规定适当的发送速率。除此之外，传输层按照网络能处理的最大尺寸将较长的数据包进行强制分割。</p>
会话层	<p>负责在网络中的两节点之间建立和维护通信。会话层的功能包括：建立通信链接，保持会话过程通信链接的畅通，同步两个节点之间的对话，决定通信是否被中断以及通信中断时决定从何处重新发送。</p>
表示层	<p>应用程序和网络之间的翻译器，在表示层，数据将按照网络能理解的方案进行格式化；这种格式化也因所使用网络的类型不同而不同。</p>
应用层	<p>应用层是开放系统的最高层，是直接为应用进程提供服务的。其作用是在实现多个系统应用进程相互通信的同时，完成一系列任务处理所需的服务。</p>

6.2 串口初始化[UINIT]

通过串口初始化指令指定通信端口、协议、串口参数等内容。



操作数	作用	类型
S1	指定协议控制字	16 位 BIN
S2	指定通信控制字	16 位 BIN
D1	指定资源占用起始 D 寄存器	16 位 BIN
D2	指定串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●								
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●								
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●								
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●								

功能与动作

当执行串口初始化指令时, 填充串口功能控制字, 重新初始化串口, 使之工作于新的通信模式。

一般而言, 串口初始化指令只需要执行一次, 执行完毕后系统将该串口指定为配置的协议。



- 上电首运行触点 M8002 执行串口初始化指令。
- 参数的配备如下:
 - ◆ 计算机链接协议(主站)
 - ◆ 串口通信参数为 9600, n, 8, 1
 - ◆ 通信缓存区首地址指向 D7000
 - ◆ 指定串口 1 为通信对象

协议控制字
S1

协议控制字	数值	说明
高 8 位	0	HEX 传输协议(内部协议, 用户无需使用)
	1	编程口协议(初始默认)
	2	计算机链接主站协议
	3	计算机链接从站协议
	4	MODBUS 协议
	5	自由协议
低 8 位	1~255	指定设备从站号, 只有当选择计算机链接从站协议和 MODBUS 协议时生效。

通信控制字
S2

通信控制字	说明	数值	意义
高 8 位	波特率	0	9600 bps
		1	19200 bps
		2	38400 bps
		3	57600 bps(保留, 用户不能使用)
		4	115200 bps(保留, 用户不能使用)
		>4	9600 bps
第 7~4 位	保留	补 0	
第 3 位	停止位	0	2 位停止位
		1	1 位停止位
第 2~1 位	校验位	00	无奇偶校验
		01	奇校验
		11	偶校验
第 0 位	数据位	0	8 位数据
		1	7 位数据

资源占用字
D1

现有协议中仅计算机链接主站协议和自由协议实际占用 D 寄存器, 其它协议需指定占用 D 寄存器的起始号, 但不使用。

- 选择计算机主站链接协议时, D1 设置参考章节 9.3.2。
- 选择自由协议时, D1 设置参考章节 9.5.1。

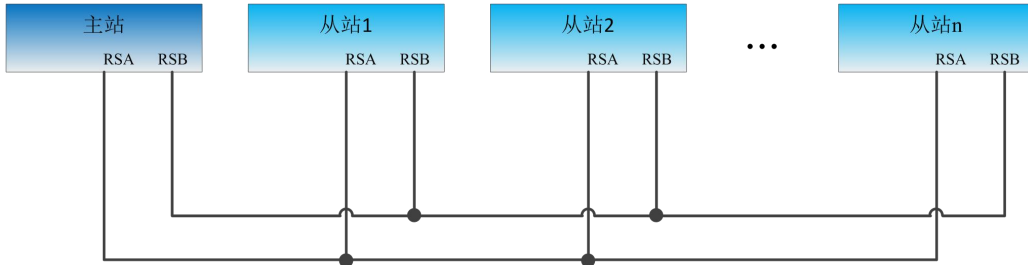
串口号 D2

- 串口号可设置为 0 或 1。
 - ◆ D2 配置为 0 指定串口 0 对象, 对应 RS232 物理接口(参考章节 4.2.1)。
 - ◆ D2 配置为 1 指定串口 1 对象, 对应 RS485 物理接口(参考章节 4.2.2)。
- 将串口 0 对象指定为非编程口协议, 可能导致数据监视功能失效。因此通常来说, 串口初始化指令针对的对象是串口 1 对象, 即 D2 = 1。

6.3 计算机链接协议

运动控制器具备计算机链接协议通信功能，符合 RS485 的标准。用户可通过编写梯形图来实现同其它设备的通信。

计算机链接协议网络构建示意图：



6.3.1 通信格式

本章节对计算机链接协议的主站通信格式、协议控制字、字节控制字通信顺序和报文格式作说明。

计算机链接协议为主从网络，在梯形图中进行通信参数的设置。设置内容包含主从站号、通信格式、任务分配等；硬件上通过 RS485 通信端子（RSA、RSB）连接，构成 485 网络。

主站可以同任何从站通信，从站与从站之间交换数据需要通过主站才能进行。

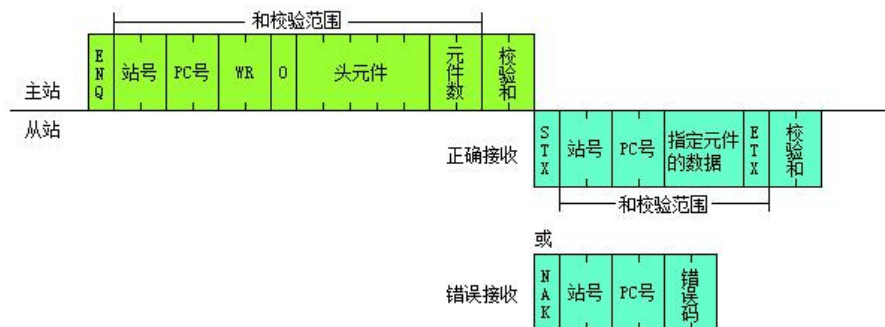
项目	规格	说明
从站数	最多 63 个	从站号 1~63
传送格式	RS485 标准	-
字 节 格 式	波特率	9600/19200/38400
	停止位	1 位
	数据位	8 位
	奇偶校验位	无校验
通信软件元范围	用户数据寄存器 D0~D7999。	

推荐参数配置：
传输速率9600 bps，无校验，数据长度8位，停止位1位。
主从站的通信参数必须配置相同。

读报文协议格式

主站读取从站数据协议格式如下所示。

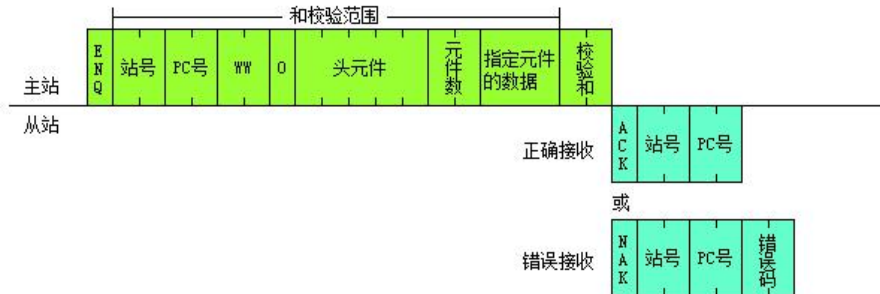
- 横线上方表示主站发送的数据，横线下方表示从站发送的数据。
- 数据发送顺序从左到右。



写报文协议
格式

主站向从站写数据协议格式如下所示。

- 横线上方表示主站发送的数据, 横线下方表示从站发送的数据。
- 数据发送顺序从左到右。



- 站号: 2 字符十六进制数, 范围 1-3F, 对应十进制 1-63. 表示可带 63 个从站。
- PC 号: 规定为 FF。
- 头元件: 通信的软元件首地址, 只能是数据寄存器 D. 用 5 个字符表示, 例如 D1000。
- 元件数: 用 2 位十六进制的字符表示, 例如 0A, 表示 10 个字元件。
- 指定元件数据: 一个元件数据用 4 个十六进制的字符表示, 如 AB96。
- 错误码: 保留使用。
- 校验和 CRC: 将和校验区 ASCII 码字符的十六进制数相加后, 取后两位字符 (00-FF)。

主站报文协议格式解释表:

名称	代码	定义	字符类型	发送方式
ENQ	05H	询问, 不计入和校验区	控制型	字符
站号	01H~3FH	从节点站号 (地址)	数据型	ASCII 码
PC 号	FFH	固定值	数据型	ASCII 码
命令	WR/WW	读/写, 以 1 个字为单位进行操作	数据型	ASCII 码
等待时间	0H	1 字符, 规定为 0H 表示无等待	数据型	ASCII 码
头元件	D□□□□	5 字符表示	数据型	ASCII 码
元件数	00H~40H	需要通信的 D 软元件个数	数据型	ASCII 码
CRC	00H~FFH	检验和 CRC	数据型	ASCII 码

从站报文协议格式解释表:

名称	代码	定义	字符类型	发送方式
ACK	06H	确认	控制型	字符
STX	02H	文本起点, 不计入和校验区	控制型	字符
NAK	15H	不确认	控制型	字符
站号	01H~3FH	从站站号	数据型	ASCII 码
PC 号	FFH	固定值	数据型	ASCII 码
指定元件数据	0H~FH	指定 D 软元件的数据, 1 个 D 软元件需 4 位字符表示	数据型	ASCII 码
ETX	03H	文本结束	控制型	字符
CRC	00H~FFH	检验和 CRC	数据型	ASCII 码

通信报文应用举例

主站与10号从站进行通信的示例报文描述。

读10号站数据：从D7223开始，共5个单元。则发送报文内容如下：

询问协议	和校验区（ASCII码—16进制数）						累加取后两位字符
	站号	PC号	命令	等待时间	头元件	元件数	
ENQ	0A	FF	WR	0	D7223	0 5	CRC
05H	30H 41H	46H 46H	57H 52H	30H	44H 37H 32H 32H 33H	30H 35H	34H 44H

$$\text{CRC} = 30\text{H} + 41\text{H} + 46\text{H} + 46\text{H} + 57\text{H} + 52\text{H} + 30\text{H} + 44\text{H} + 37\text{H} + 32\text{H} + 32\text{H} + 33\text{H} + 30\text{H} + 35\text{H} = 34\text{DH}$$

取最后2个字符4D，4DH用ASCII码表示是34H 44H。

从站回报文内容如下：

询问	和校验区（ASCII码—16进制数）							累加取后两位字符	
	站号	PC号	指定元件的数据（见注释）				文本结束		
STX	0A	FF	0101	0909	FFFF	6565	6D6D	ETX	CRC
02H	30H 41H	46H 46H	30H 31H 30H 31H	30H 39H 30H 39H	46H 46H 46H 46H	36H 35H 36H 35H	36H 44H 36H 44H	03H	37H 36H

$$\text{CRC} = 30\text{H} + 41\text{H} + 46\text{H} + 46\text{H} + 30\text{H} + 31\text{H} + 30\text{H} + 31\text{H} + 30\text{H} + 39\text{H} + 30\text{H} + 39\text{H} + 46\text{H} + 46\text{H} + 46\text{H} + 46\text{H} + 36\text{H} + 35\text{H} + 36\text{H} + 35\text{H} + 36\text{H} + 44\text{H} + 36\text{H} + 44\text{H} + 03\text{H} = 576\text{H}$$

取最后2个字符76，用ASCII码表示是37H 36H。

注：指定元件的数据，每一元件必须用4位字符来表示。本例5个D寄存器软元件，给定的5个4位字符分别是0101、0909、FFFF、6565、6D6D（此为示例值，实际以通信数值为准）。

6.3.2 任务调度配置

运动控制器作为计算机链接主站协议时，需要对通信协议调度区进行配置。

通信协议调度区起始地址由串口初始化指令 UINIT 的操作元 D1 指定，指定调度区 D 寄存器的起始号，调度区将占用从起始号开始的 20 个 D 寄存器。

如 $D1 = 7000$ ，则 $D7000 \sim D7019$ 为该串口的任务调度数据区。

地址单元	说明	推荐初始化数值
D7000	0 级读任务起始单元	用户自定义，一般选取 D7000 以后的单元
D7001	0 级读任务数	用户自定义
D7002	0 级读任务数据长度(字节数)	用户自定义，大于等于读数据个数乘以 2
D7003	0 级写任务起始单元	用户自定义，一般选取 D7000 以后的单元
D7004	0 级写任务数	用户自定义
D7005	0 级写任务数据长度(字节数)	用户自定义，大于等于写数据个数乘以 2
D7006	1 级读任务起始单元	用户自定义，一般选取 D7000 以后的单元
D7007	1 级读任务数	用户自定义
D7008	1 级读任务数据长度(字节数)	用户自定义，大于等于写数据个数乘以 2
D7009	1 级写任务起始单元	用户自定义，一般选取 D7000 以后的单元
D7010	1 级写任务数	用户自定义
D7011	1 级写任务数据长度(字节数)	用户自定义，大于等于写数据个数乘以 2
D7012	00#~15# 设备状态: 0 在线 1 离线	0xFFFF
D7013	16#~31# 设备状态: 0 在线 1 离线	0xFFFF
D7014	32#~47# 设备状态: 0 在线 1 离线	0xFFFF
D7015	48#~63# 设备状态: 0 在线 1 离线	0xFFFF
D7016	从站正确返回数据的报文数	0
D7017	从站不能正确返回数据的报文数	0
D7018~D7019	保留使用	-

多从站任务调度

当主站与多个从站进行通信时，需遵循读/写任务数据长度取大值的原则。

假定主站与从站A和从站B进行通信，根据需求分配读写任务数和任务数据长度。

- 与从站A的实际通信量为读取2个D寄存器数据，写入4个D寄存器数据。

- 与从站B的实际通信量为读取3个D寄存器数据，写入3个D寄存器数据。

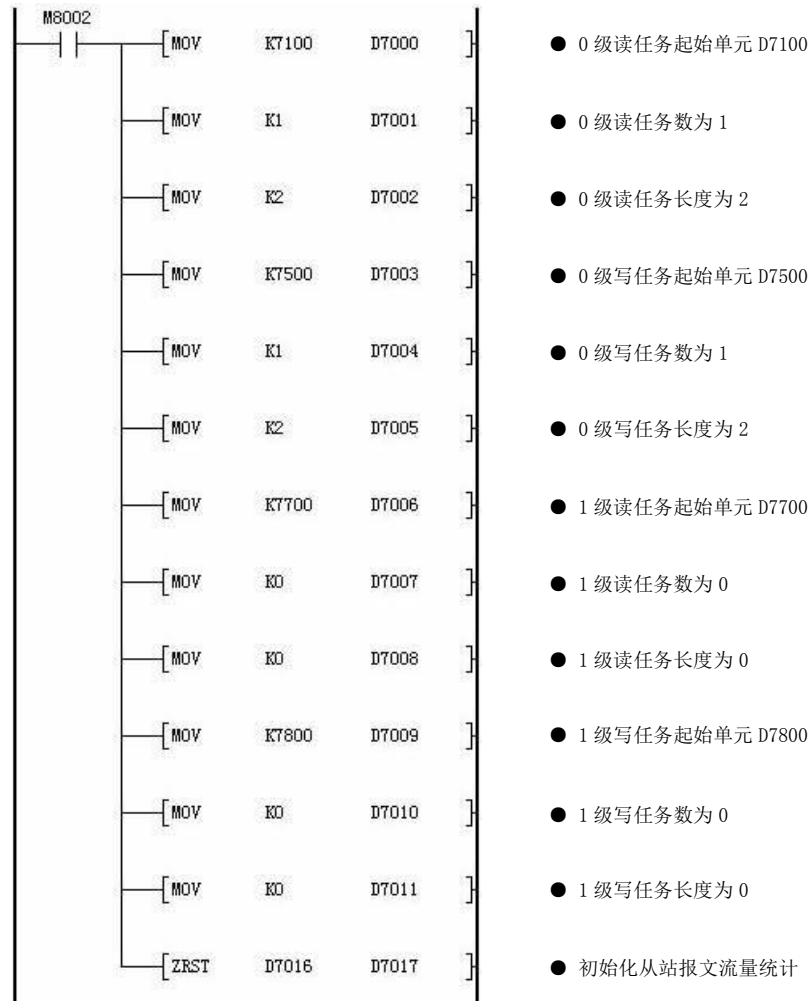
可以分配0级读任务2个，0级写任务2个，用来和从站A以及从站B进行通信。

0级读任务数据长度必须填写6，即取从站中读任务数据长度最长的数值。对于从站A而言，前2个D寄存器作为有效数据应用即可。

0级写任务数据长度必须填写8，即取从站中写任务数据长度最长的数值。对于从站B而言，前3个D寄存器作为有效数据应用即可。

任务调度区的初始化工作必须在 UINIT 指令执行之前完成，通常采用上电第一个扫描周期进行配置。

任务调度区初始化梯形图示例如下 (UINIT 的操作元 $D1=7000$ ，0 级读写任务数为 1，通信数据字节为 2)：



* UINIT 指令介绍请参考章节 9.2 描述。

计算机链接协议规划了两级任务：0 级和 1 级任务。

- 0 级发送任务：发送报文数占发送总报文数的 2/3。
- 1 级发送任务：发送报文数占发送总报文数的 1/3。
- 0 级接收任务：接收报文数占接收总报文数的 2/3。
- 1 级接收任务：接收报文数占接收总报文数的 1/3。

分级任务报文数的遵循规则：

- 在报文数相同的情况下，0 级数据循环比 1 级数据循环快 1 倍。
- 如果 0 级报文数等于 1 级报文数的两倍，则通讯循环次数 0 级和 1 级一样。
- 如果 0 级报文数超出 1 级报文数的两倍以上，则 0 级比 1 级循环次数少。

例如：每秒报文总流量为 30 个/秒，则 0 级每秒报文流量为 20 个/秒，1 级报文流量为 10 个/秒。如果规划 0 级报文个数为 2 个，1 级报文个数为 5 个，则每秒 0 级报文循环次数为 $20/2=10$ 次，1 级报文循环次数为 $10/5=2$ 次，从而满足不同的实时要求。

6.3.3 通信示例

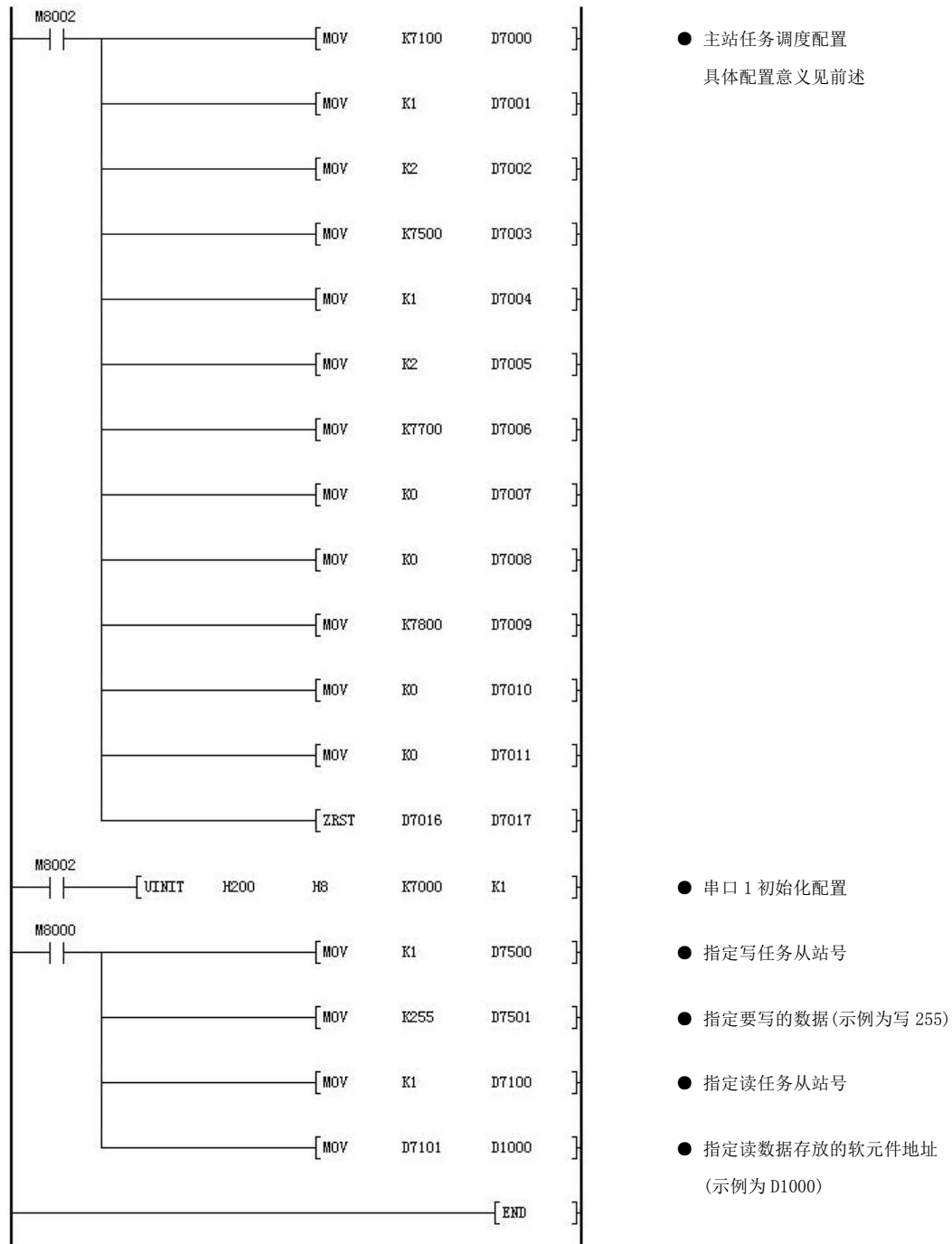
主站：HM 运动控制器

选择计算机链接主站协议，在执行串口初始化指令（UINIT）之前，必须按照前面所述任务调度先配置好任务调度数据（若为从站，则不需要配置任务调度数据）。

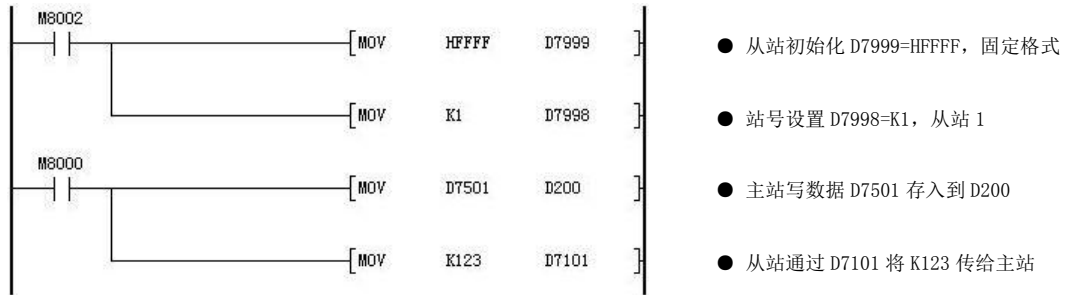
从站：MP1-10M08R 一体机

示例中所选择的通信端口及格式为：COM1、9600bps、无奇偶校验、8 数据位、1 停止位。

主站梯形图（载体 HM）：



从站梯形图（载体 MP1-10M08R）：



梯形图解释

- 主站向从站写数据的过程为：
首先选择从站站号，实例中配置的 0 级写任务起始单元（D1003=K7500）为 D7500，那么给 D7500 赋值 K1 表示向 1 号从站写数据，写任务的起始单元装载的是从站号；D7501 是主站向从站写数据的中间寄存器，主站的数据（K255）装载到 D7501 中，从站中读取 D7501 寄存器的值就获得了主站写给从站的数据（K255）。
- 主站读取从站数据的过程为：
首先选择从站站号，实例中配置的 0 级读任务起始单元（D1000=K7100）为 D7100，那么给 D7100 赋值 K1 表示读 1 号从站的数据，读任务的起始单元装载的是从站号；D7101 是从站向主站传数据的中间寄存器，从站的数据（K123）装载到 D7101 中，主站中读取 D7101 寄存器的值就获得了从站传给主站的数据（K123）。
- 对于 1 级读写从站的数据过程同 0 级任务一样，实例中 1 级读任务的起始单元（D1006=K7700）为 D7700，1 级写任务起始单元（D1009=K7800）为 D7800；读写任务数均为 0 表示没有 1 级读写任务。
- 当读写任务数设置为 0 时，无论读写任务数据长读是否为 0，都不执行读写任务。

6.4 MODBUS 协议

运动控制器具备完整的 MODBUS 协议通信功能，符合国际标准 (RTU 模式)。用户可通过编写梯形图来实现运动控制器同其它设备的通信。

MODBUS 串行链路协议是一个主从协议。在同一时间，总线上只能有一个主站，和一个或多个（最多 247 个）从站。MODBUS 通信总是由主站发起。当从站没有收到来自主站的请求时，不会发送数据，从站之间不能互相通信。主站同时只能启动一个 MODBUS 事务处理。

主站用两种模式向从站发出 MODBUS 请求：

- 单播模式，主站寻址单个从站。从站接收并处理完请求之后，向主站返回一个报文。
在这种模式下，一个 MODBUS 事务处理包含两个报文：一个是主站的请求，另一个是从站的应答。每个从站必须有唯一的地址（1~247），这样才能区别于其他站独立地被寻址。
- 广播模式，主站可以向所有的从站发送请求。
对于主站发送的广播请求没有应答返回；广播请求必须是写命令。所有设备必须接受广播方式的写命令。地址 0 被保留用来识别广播通信。

6.4.1 通信格式

运动控制器与其他设备进行 MODBUS 通信时，以下对通信报文格式、地址域、功能域、数据域、错误校验域进行说明。

MODBUS 通信方式为 RTU 模式，报文所允许的最大长度为 256 个字节，除开地址、功能码和 CRC 校验位，数据部分长度剩下 252 个字节。

项目	规格	说明
传送格式	MODBUS 标准	通信方式为 RTU 模式，报文所允许的最大长度为 256 个字节。报文各项数据所占字节长度为： <ul style="list-style-type: none"> ● 地址(站号)：8-Bits ● 功能码：8-Bits ● 正文数据：N×8-Bits ● CRC 校验：16-Bits
站号地址	0~255	MODBUS 寻址空间由 256 个不同地址组成。主站无需指定地址，每个从站有唯一的一个地址。地址 0 为广播地址，地址 1~247 为从站单个地址，地址 248~255 保留。所有从站能够识别广播地址。 主站通过将要联络的从站的地址放入消息中的地址域来选定从站。当从站发送回应消息时，它把自己的地址放入回应的地址域中，以便主站知道是哪一个从站作出回应。
字节格式	波特率	9600/19200/38400
	停止位	1/2 位
	数据位	7/8 位
	奇偶校验位	奇/偶/无校验
通信软件范围	所有软件均可通信。	

6.4.2 软元件通信地址

元件名称	MODBUS 地址	元件名称	MODBUS 地址
位软元件		字软元件	
M0~M3071	0200H~0DFFH	D0~D7999	0170H~20AFH
X000~X377	0000H~00FFH	D8000~D8255	20B0H~21AFH
Y000~Y377	0100H~01FFH	T0~T255	21B0H~22AFH
S0~S1023	0F00H~0FFFH	C0~C199	22B0H~2377H
M8000~M8255	0E00H~0EFFH	C200~C255 (32 位)	2378H~23E7H
T0~T255 (线圈)	1300H~13FFH	R0~R99	23E8H~244BH
T0~T255 (触点)	1400H~14FFH	对软元件 C200~C255 的访问, 每个元件相当于 2 个字元件。	
C0~C255 (线圈)	1500H~15FFH		
C0~C255 (触点)	1600H~16FFH		

6.4.3 MODBUS 指令

本章节介绍运动控制器支持的 MODBUS 指令的功能及作用。

MODBUS 通信协议下, 当主站向从站发送信息时, 由功能代码向从站说明应执行的动作。如读一组离散式线圈或输入信号的 ON/OFF 状态, 读一组寄存器的数据, 读从站的诊断状态, 写线圈 (或寄存器), 允许下载、记录、确认从机内的程序等。

- 通过串口初始化指令 (UINIT) 将串口对象配置为 MODBUS 主站, 才能使用 MODBUS 指令与从站进行通信。



- 通过串口初始化指令 (UINIT) 将串口对象配置为 MODBUS 从站, 才能与主站进行通信。



功能码定义及运动控制器 MODBUS 通信指令如下表所示。

功能码	通信指令	名称	功能	可操作软元件
01H	MRDC	读线圈	读一组逻辑线圈的当前状态	X Y M S T C
02H	MRDI	读输入线圈	读一组开关输入的当前状态	X Y M S T C
03H	MRDD	读寄存器	读一个或多个寄存器中当前值	D R T C
04H	MRDAI	读输入寄存器	读一个或多个输入寄存器当前值	D R T C
05H	MWRC	写单个线圈	强置一个逻辑线圈的通断状态	Y M S T C
0FH	MWRCS	写多个线圈	强置一串连续逻辑线圈的通断	Y M S T C
06H	MWRD	写单个寄存器	强置一个特定的值到一个寄存器	D R T C
10H	MWRDS	写多个寄存器	强置一系列特定的值到一系列多寄存器	D R T C

* 使用 MODBUS 指令对从站设备通信时, 单次读写从站元件个数不要超过 256 个。

6.4.3.1 读线圈[MRDC], 读输入线圈[MRDI]

读取远程设备中逻辑/开关输入一个或多个线圈的当前状态。

读线圈	
<p>16 位指令 MRDC 连续执行型</p> <p>11 步 MRDC P 脉冲执行型</p>	<p>32 位指令</p> <p>无</p>
读输入线圈	
<p>16 位指令 MRDI 连续执行型</p> <p>11 步 MRDI P 脉冲执行型</p>	<p>32 位指令</p> <p>无</p>

操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软元件首地址	16 位 BIN
S3	通信元件个数	16 位 BIN
D1	主站软元件首地址编号	位
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●

功能与动作

读取远程设备一个或多个线圈/输入开关量的状态,建立远程设备与本地设备之间的读取数据通信功能。

《读线圈》



将站号 1 远程设备中地址 0 开始的 100 个位读入本地设备的 M0~M99, 串口号为 1。

《读输入线圈》



将站号 1 远程设备中地址 0 开始的 10 个位读入本地设备的 Y010~Y017、Y020~Y021, 串口号为 1。

使用注意事项

- 读线圈与读输入线圈两条指令的处理机制相同。
- 运动控制器做从站时,对于软元件通信个数 S3 的范围限定:
 - ◆ M 软元件每次通信个数不超过 124 个。
 - ◆ 其他位软元件每次通信个数不超过 60 个。
- 运动控制器做主站时,使用边沿触点(上升/下降沿)触发 MODBUS 指令,且每次通信间隔时间最好在 1S 以上,否则可能造成报文丢失。
- 当指定通信串口号 D2=1 时,确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 示例中所示远程设备地址为参考值,不代表运动控制器的软元件 MODBUS 地址。

6.4.3.2 读寄存器[MRDD], 读输入寄存器[MRDAI]

读取远程设备中一个或多个寄存器/输入寄存器的当前值。

读寄存器	
	
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #ffffcc;"> <p>16 位指令 MRDD 连续执行型</p> <p>11 步 MRDD P 脉冲执行型</p> </div>	<div style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #ffffcc;"> <p>32 位指令</p> <p style="text-align: center;">无</p> </div>
读输入寄存器	
	
<div style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #ffffcc;"> <p>16 位指令 MRDAI 连续执行型</p> <p>11 步 MRDAI P 脉冲执行型</p> </div>	<div style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #ffffcc;"> <p>32 位指令</p> <p style="text-align: center;">无</p> </div>

操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软元件首地址	16 位 BIN
S3	通信元件个数	16 位 BIN
D1	主站软元件首地址编号	16 位 BIN
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

读取远程设备一个或多个寄存器/输入寄存器中当前值,建立远程设备与本地设备之间的读取数据通信功能。

《读寄存器》



将站号 10 远程设备中地址 1000 开始的 100 个字读入本地设备的 D1000~D1099, 串口号为 1。

《读输入寄存器》



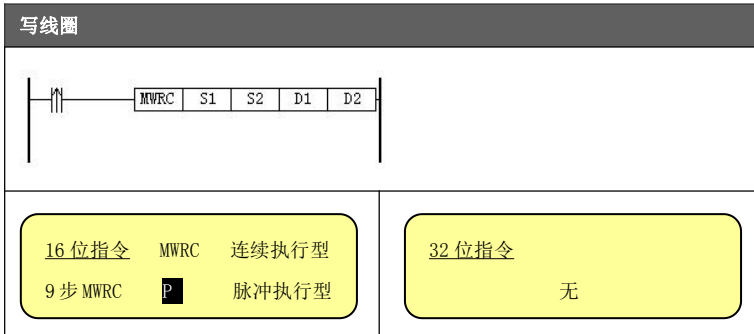
将站号 10 远程设备中地址 2000 开始的 10 个字读入本地设备的 D2000~D2009, 串口号为 1。

使用注意事项

- 读寄存器与读输入寄存器两条指令的处理机制相同。
- 运动控制器做从站时,对于软元件通信个数 S3 的范围限定不超过 125 个。
- 运动控制器做主站时,使用边沿触点(上升/下降沿)触发 MODBUS 指令,且每次通信间隔时间最好在 1S 以上,否则可能造成报文丢失。
- 当指定通信串口号 D2=1 时,确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 示例中所示远程设备地址为参考值,不代表运动控制器的软元件 MODBUS 地址。

6.4.3.3 写线圈[MWRC]

强置远程设备一个逻辑线圈的通断状态。



操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软件地址	16 位 BIN
D1	主站软件编号	16 位 BIN
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●

功能与动作

强置远程设备一个逻辑线圈的通断状态。



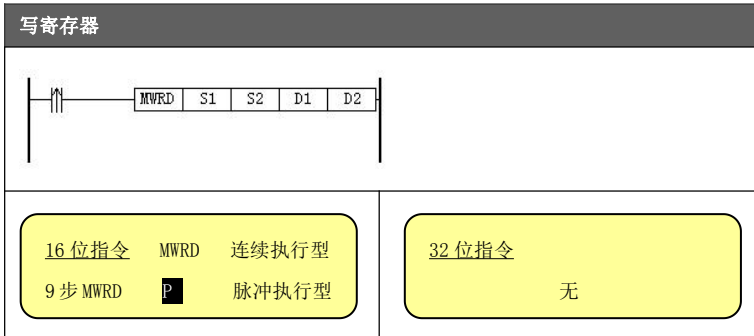
将 M0 软元件状态送入站号 5 远程设备中地址 300 内，串口号为 1。

使用注意事项

- 确认从站的地址 S2 是一个可写入的位地址。
- 运动控制器做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令，且每次通信间隔时间最好在 0.6 秒以上，否则可能造成报文丢失。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 主站软件编号 D1 不能使用变址修饰。
- 示例中所示远程设备地址为参考值，不代表运动控制器的软元件 MODBUS 地址。

6.4.3.4 写寄存器[MWRD]

改变远程设备一个字地址的当前数值。



操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软元件地址	16 位 BIN
D1	主站软元件编号	16 位 BIN
D2	通信串口号	16 位 BIN

操作数	字型软元件										
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								

功能与动作

改变远程设备一个字地址的当前数值。



将 D2000 软元件的内容送入站号 5 远程设备中地址 2000 内，串口号为 1。

使用注意事项

- 确认从站的地址 S2 是一个可写入的字地址。
- 运动控制器做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令，且每次通信间隔时间最好在 1S 以上，否则可能造成报文丢失。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 主站软元件编号 D1 不能使用变址修饰。
- 示例中所示远程设备地址为参考值，不代表运动控制器的软元件 MODBUS 地址。

6.4.3.5 写多个线圈[MWRCS]

强置远程设备一个或多个逻辑线圈的通断状态。

写多个线圈

16 位指令	MWRCS	连续执行型
11 步 MWRCS	P	脉冲执行型

32 位指令	无
--------	---

操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软件地址	16 位 BIN
S3	通信元件个数	16 位 BIN
D1	主站软件编号	位
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

操作数	位型软元件			
D1	X	Y	M	S
		●	●	●

功能与动作

强置远程设备强置一串连续逻辑线圈的通断，建立远程设备与本地设备之间的读取数据通信功能。



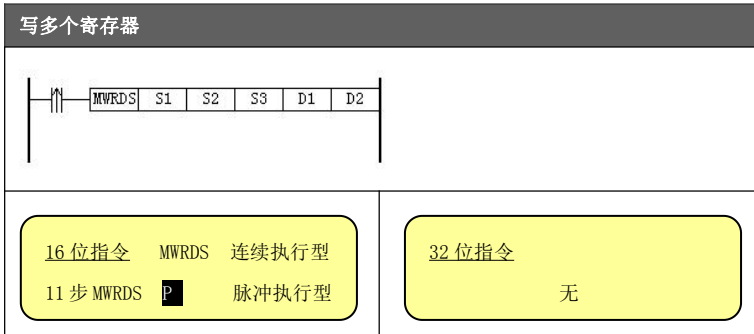
将 M0~M5 软元件状态送入站号 1 远程设备中地址 300~304 内，串口号为 1。

**使用注意
事项**

- 确认从站的地址 S2 是一个可写入的位地址。
- 运动控制器做从站时，对于软元件通信个数 S3 的范围限定：
 - ◆ M 软元件每次通信个数不超过 124 个。
 - ◆ 其他位软元件每次通信个数不超过 60 个。
- 运动控制器做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令，且每次通信间隔时间最好在 1S 以上，否则可能造成报文丢失。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 示例中所示远程设备地址为参考值，不代表运动控制器的软元件 MODBUS 地址。

6.4.3.6 写多个寄存器[MWRDS]

改变远程设备一个或多个寄存器的当前数值。

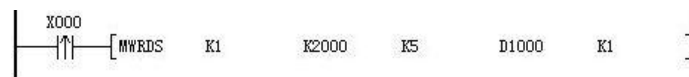


操作数	作用	类型
S1	从站站号	16 位 BIN
S2	从站软元件地址	16 位 BIN
S3	通信元件个数	16 位 BIN
D1	主站软元件编号	16 位 BIN
D2	通信串口号	16 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

强置一系列特定的值到一系列多寄存器，建立远程设备与本地设备之间的读取数据通信功能。



将 D1000~D1004 软元件的内容送入站号 1 远程设备中地址 2000~2004 内，串口号为 1。

使用注意事项

- 确认从站的地址 S2 是一个可写入的字地址。
- 运动控制器做从站时，对于软元件通信个数 S3 的范围限定不超过 123 个。
- 运动控制器做主站时，使用边沿触点（上升/下降沿）触发 MODBUS 指令，且每次通信间隔时间最好在 1S 以上，否则可能造成报文丢失。
- 当指定通信串口号 D2=1 时，确保用串口初始化指令将串口 1 匹配为 MODBUS 协议。
- 示例中所示远程设备地址为参考值，不代表运动控制器的软元件 MODBUS 地址。

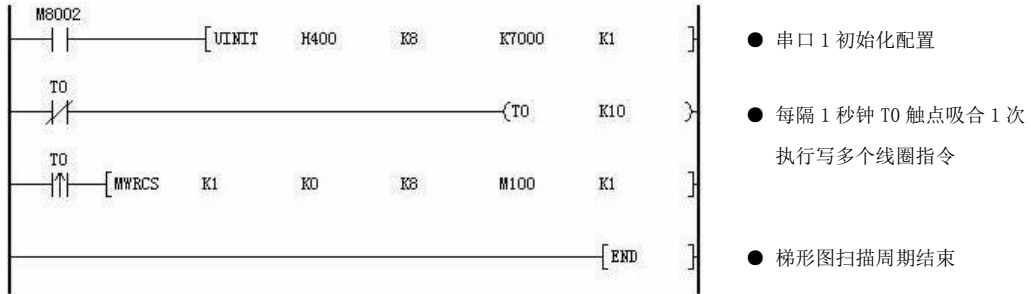
6.4.4 通信示例

主站：HM 运动控制器

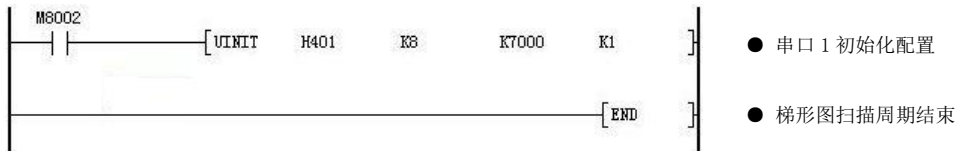
从站：HM 运动控制器

示例中所选择的通信端口及格式为：COM1、9600bps、无奇偶校验、8 数据位、1 停止位。

主站梯形图（载体运动控制器）：



从站梯形图（载体运动控制器）：



梯形图解释

- 此梯形图的作用是将主站的 M100~M107 的状态送到从站 M0~M7 内，每隔 1 秒钟传送 1 次，1 次传送 8 个位状态。
- 主站通过串口初始化指令将串口 1 配置为 MODBUS (站号为 0 即代表主站)。
- 从站通过串口初始化指令将串口 1 配置为 MODBUS 从站，从站号设置为 1。
- 从站软件元件的 0 地址指向的是 M0，MODBUS 软件元件地址参考章节 9.4.2。

6.5 自由协议

自由格式通讯是以数据块的形式进行数据传送，每次发送数据量最大为 255 个字节。

所谓自由格式，即自定义协议通讯。现在市场上很多智能设备都支持 RS232 或者 RS485 通讯，而各家产品所使用的协议不尽相同。例如：三菱 PLC 使用标准编程口协议，伺服变频器通常使用 MODBUS 协议，还有一些温度控制器厂家使用的自定义非标协议。在其产品的使用手册上，基本上都能查询的到其协议的具体格式和收发时序要求。

当知道了这些，就可以用自由协议对其报文格式和收发时序进行模拟。这样即可通讯上第三方设备，对系统的组建起到关键的作用。

自由协议的主从信息地址信息是隐含在编程中，主从信息是在编程时设置主动发送信息还是被动接收信息，地址信息包含规划的报文格式中及解读报文的编程中。

6.5.1 任务调度配置

在运动控制器作为自由协议时，需要对通信协议调度区进行配置。

通信协议调度区起始地址由串口初始化指令 UINIT 的操作元 D1 指定，指定调度区 D 寄存器的起始号，调度区将占用从起始号开始的 20 个 D 寄存器。

如 D1 = 7000，则 D7000~D7019 为该串口的任务调度数据区。

地址单元	说明	推荐初始化数值
D7000	发送起始单元	激活发送指令 USEND 后由系统填写
D7001	待发送字节数	激活发送指令 USEND 后由系统填写
D7002	接收起始单元(发送自动转接收或手动置位接收请求标志位的情况下由用户自定义)	<ul style="list-style-type: none"> ● 激活接收指令 URCV 后由系统填写 ● 用户自定义
D7003	待接收字节数(发送自动转接收或手动置位接收请求标志位的情况下由用户自定义)	<ul style="list-style-type: none"> ● 激活接收指令 URCV 后由系统填写 ● 用户自定义，1~255
D7004	接收起始符，标示报文数据的开始，可视为报文的帧头	用户自定义，0
D7005	接收终止符，标示报文数据的终止，可视为报文的帧尾	用户自定义，0
D7006	接收等待间隔时间，用来判断接收报文结束或发送接收超时	用户自定义，10，单位毫秒
D7007	发送接收控制字(定义下述)	用户自定义，1
D7008	自由协议标志位(定义下述)	由系统填写，也可部分手动设置，0
D7009	实际已接收字节数	由系统填写
D7010~D7019	保留使用	0

发送接收控制字	控制字(按位定义)	对应功能
	Bit0	发送自动转接收 0: 保持 1: 自动转接收
	Bit1	接收起始字符过滤 0: 不过滤 1: 过滤 设置过滤，接收方将过滤掉发送过来的接收起始符之前的所有内容。
	Bit2	接收终止字符过滤 0: 不过滤 1: 过滤 设置过滤，接收方将过滤掉发送过来的接收终止符之后的所有内容。
	Bit15~Bit3	保留使用

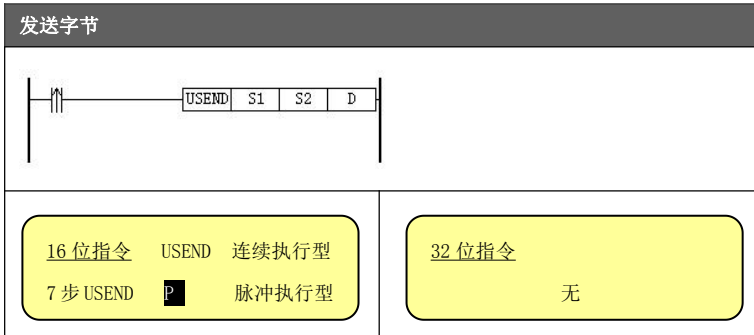
自由协议标志位	控制字(按位定义)	对应功能
	Bit0	发送请求标志 当执行发送指令 USEND 后被置位
	Bit1	发送成功标志 当发送指令 USEND 执行完成后被置位
	Bit2	接收请求标志 <ul style="list-style-type: none"> ● 当发送指令 USEND 执行完成后, 且设置了发送转接收功能时被置位 ● 当接收指令 URCV 被激活时被置位 ● 可手动置位, 直接进入接收状态
	Bit3	接收成功标志 <ul style="list-style-type: none"> ● 当接收指令 URCV 执行完成后被置位 ● 当接收指令 URCV 被激活时被复位 ● 手动置位接收请求标志后, 接收完成时被置位 ● 状态由系统给出, 手动修改无效
	Bit4	接收起始符标志 当接收到接收起始符后被置位
	Bit5	接收终止符标志 当接收到接收终止符后被置位
	Bit6	奇偶校验错标志
	Bit7	发送接收超时标志
	Bit8	D 寄存器设置错误 <ul style="list-style-type: none"> ● 执行串口初始化指令 UINIT 时, 当配置区的起始单元和字符数设置超限时被置位 ● 当发送指令 USEND 或接收指令 URCV 被激活, 且所填软元件的起始单元和字符数设置超限时被置位
	Bit15~Bit9	保留使用

6.5.2 自由协议指令

本章节介绍自由协议下专用的发送接收指令的功能和作用。

6.5.2.1 发送字节[USEND]

通过指定串口发送 1 个或多个字节。



操作数	作用	类型
S1	指定本地发送的软件首地址编号	16/32 位 BIN
S2	指定发送的字符个数	16/32 位 BIN
D	指定串口号	16/32 位 BIN

操作数	字型软元件									
S1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
S2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

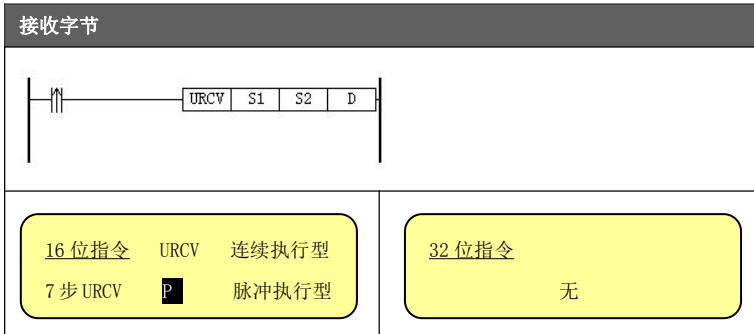
- 通过指定串口发送 1 个或多个字节 ($255 \geq S2 \geq 1$)。
- 指令执行时，发送请求标志位被置位；指令执行完毕，发送完成标志位被置位。
- S1 指定发送寄存器软元件的首地址编号，发送连续 S2 个 D 寄存器的低字节数据。
- 若 S1 和 S2 设置后使得寄存器范围超限（比如 S1 设置为 K7990，S2 设置为 K20，指向的地址区最终将超过 D7999），D 寄存器设置出错标志位被置位。
- 若发送自动转接收标志位被设置，则发送完毕接收请求标志位被置位，系统进入接收状态，这种情况下接收起始单元和字节个数由任务调度配置区指定。
- 发送指令 USEND 不能使用常开或常闭触点激活，必须使用边沿（上升/下降沿）指令触发。



- 当 X000 触点闭合时，向串口 1 对象发送 D4000，D4001，D4002，D4003 的低字节（高字节被自动忽略）。
- 如 D4000=H1122，D4001=H3344，D4002=H5566，D4003=H7788，那么发送到串口的数据为 H22，H44，H66 和 H88 这 4 个字节。

6.5.2.2 接收字节[URCV]

通过指定串口接收 1 个或多个字节。



操作数	作用	类型
S1	指定本地接收的软件首地址编号	16/32 位 BIN
S2	指定接收的字符个数	16/32 位 BIN
D	指定串口号	16/32 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
S1	●	●	●							
S2	●	●	●							
D	●	●	●							

功能与动作

- 通过指定串口接收 1 个或多个字节 ($255 \geq S2 \geq 1$)。
- 指令执行时，接收请求标志位被置位，接收完成标志位被复位；指令执行完毕，接收完成标志位被置位。强制置位接收请求标志位，也可以在不激活接收指令的情况下接收到串口数据，这种情况下接收起始单元和字符数由任务调度区指定。
- S1 指定接收寄存器软元件的首地址编号，存入连续 S2 个 D 寄存器的低字节数据。
- 若 S1 和 S2 设置后使得寄存器范围超限（比如 S1 设置为 K7990，S2 设置为 K20，指向的地址区最终将超过 D7999），D 寄存器设置出错标志位被置位。
- 若接收到字节后一段时间内再未接收到数据（这个时间由任务调度区的接收等待间隔时间设定），即使接收字节数小于设定值，仍然判定接收指令执行完毕，接收完成标志位被置位。
- 接收指令 URCV 不能使用常开或常闭触点激活，必须使用边沿（上升/下降沿）指令触发。



- 当 X001 触点闭合时，串口 1 对象等待接收，接收内容存入 D2000，D2001，D2002，D2003 的低字节。
- 若接收到字节的内容为 H22，H44，H66 和 H88 这 4 个字节，且设置为不过滤模式，那么结果为 D2000=H22，D2001=H44，D2002=H66，D2003=H88，这 4 个寄存器的高字节被自动清 0。

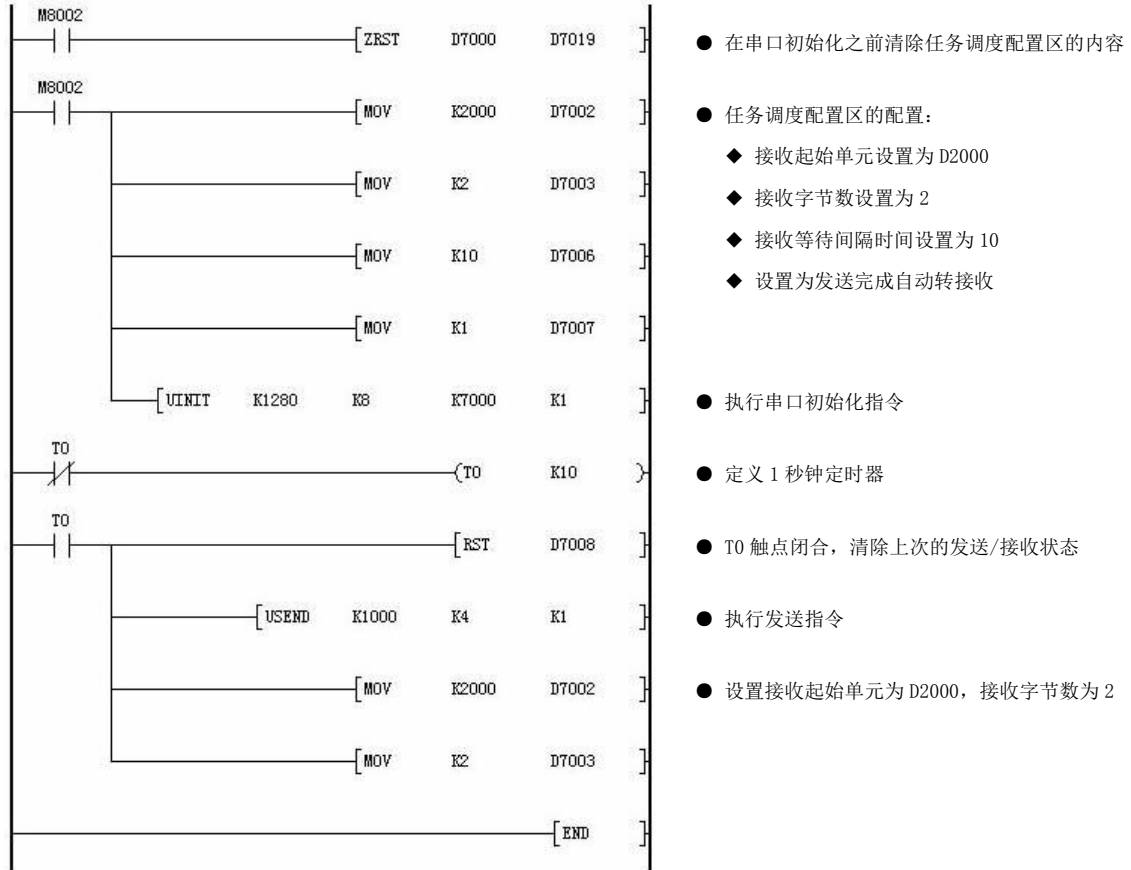
6.5.3 通信示例

通信设备 A: HM 运动控制器

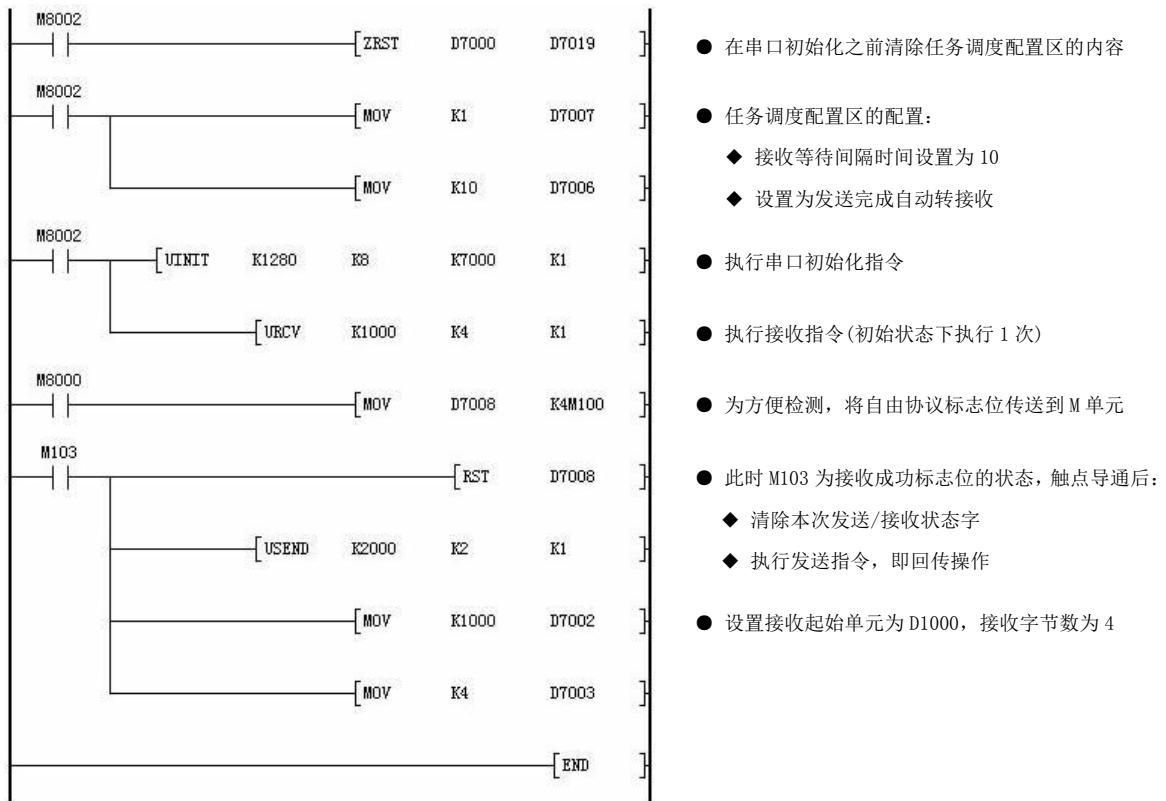
通信设备 B: HM 运动控制器

示例中所选择的通信端口及格式为: COM1、9600bps、无奇偶校验、8 数据位、1 停止位。

通信设备 A 梯形图 (HM 运动控制器):



通信设备 B 梯形图（HM 运动控制器）：



梯形图解释

- 此梯形图的作用是将设备 A 的软元件 D1000~D1003 的内容传送到设备 B 中，将设备 B 的软元件 D2000~D2001 的内容传送到设备 A 中，每隔 1 秒钟执行一次双方通信。
- 改变设备 A 中 D1000~D1003 的数值（低字节），可以发现设备 B 中 D1000~D1003 的数值同样跟着改变；改变设备 B 中 D2000~D2001 的数值（低字节），可以发现设备 A 中 D2000~D2001 的数值同样跟着改变。
- 设备 A 每隔 1 秒钟执行一次发送指令，然后自动转入接收状态。
- 设备 B 初始扫描周期就执行接收指令（仅在此位置执行了一次），然后每个扫描周期检测接收成功标志位，一旦检测到则清除状态标志位，然后执行回传数据的发送指令。再次自动转入接收状态等待。
- 为确保设备 A/B 的通信时序的正确，设置了接收等待间隔时间以确保设备 B 回传时设备 A 发送转接收状态已经稳定。
- 以上示例的接收/发送指令都为常开触点指令触发，但根据梯形图上下文关联，导通状态只能持续一个扫描周期，并不会导致每个扫描周期都触发发送/接收指令。

6.6 其他协议

本章节将要介绍的协议均属于点对点传输的内部协议，这些协议不能用于组建通信网络，所以它们将作为内部协议使用。用户推荐使用前述章节所介绍的协议进行构建通信网络。

HEX 传输协议

- 通常应用于下载批量数据，驱动代码或是应用文件时使用。
- 以页为单位进行传输，每帧报文包含 1 页数据（128 bytes）。
- 以字符型进行传输，一种高速传输的通信方案。通信流程包含下载→比较→固化这 3 个传输过程。
- HEX 传输协议是一个内部的协议，用户无需了解协议结构。

编程口协议

- 梯形图上传/下载以及元件监视的专用协议。
 - 以 ASCII 码型进行传输，每帧报文包含数据可变，但一般不超过 64 个字节。
 - 编程口协议是一个内部的协议，用户无需了解协议结构。
- 编程口协议兼容三菱 FX2N 的编程口协议，在使用第三方设备与运动控制器进行通信时（比如 HMI），也可选择该协议模式对元件进行监视。

第七章 扩展模块

运动控制器通过 SPI 总线方式连接扩展模块，有开关量扩展和模拟量扩展模块，最多可连接六个扩展模块。开关量扩展模块增加 I/O 点数，模拟量模块在温度、流量、液位、压力等过程控制系统有着广泛应用。

7.1 扩展寄存器

运动控制器的 D8129 为扩展寄存器的设置首地址，初始默认值为 7100，即默认 D7100 开始的区域是扩展管理寄存器区。用户可以更改 D8129 修改缺省的起始地址，PLC 重新运行设置生效。

每个扫描周期结束，如有通信任务完成，则进行扩展的数据交换。以 D7100 为首地址进行示例说明，扩展寄存器区定义如下所示：

D7100~D7106	<p>从站 1 分配地址</p> <ul style="list-style-type: none"> ● D7100: 从站的 X 端子在主站上的起始地址 ● D7101: 从站的 Y 端子在主站上的起始地址 ● D7102~D7103: 内部使用 ● D7104: 从站的 AD 值在主站上的起始地址 ● D7105: 从站的 PGA 设置值以及 DA 值在主站上的起始地址 ● D7106: 内部缓存区起始地址
D7107~D7113	从站 2 分配地址，分配地址参考从站 1 分配地址
D7114~D7120	从站 3 分配地址，分配地址参考从站 1 分配地址
D7121~D7127	从站 4 分配地址，分配地址参考从站 1 分配地址
D7128~D7134	从站 5 分配地址，分配地址参考从站 1 分配地址
D7135~D7141	从站 6 分配地址，分配地址参考从站 1 分配地址
D7142	接入的从站个数
D7143	<p>开关量扩展刷新时间设置(默认 60ms)，可在线修改。</p> <ul style="list-style-type: none"> ● 定时小于扫描周期时，以扫描周期通信 ● 定时大于扫描周期时，以扫描周期整数倍通信 ● 开关量扩展通信优先级高于模拟量扩展，故该参数不能设置太小，否则将影响模拟量模块的通信
D7144	<p>模拟量扩展刷新时间设置(默认 60ms)，可在线修改。</p> <ul style="list-style-type: none"> ● 定时小于扫描周期时，以扫描周期通信 ● 定时大于扫描周期时，以扫描周期整数倍通信
D7145	开关量扩展刷新时间实测
D7146	模拟量扩展刷新时间实测
D7147	<p>扩展错误序号</p> <ul style="list-style-type: none"> ● =0 正确 ● =1 接入的扩展分配超限 <ul style="list-style-type: none"> X 扩展输入最大限制: 128 个点 Y 扩展输出最大限制: 128 个点 模拟量接入最大限制: AD/DA 数值及其自动分配占用的 D 寄存器不得超过 384 个。 ● =2 手动分配的地址超限
D7200~D7583	模拟量扩展数据区

扩展接续起始地址

- 开关量扩展 X 输入从 X20 开始
- 开关量扩展 Y 输出从 Y20 开始
- 模拟量扩展地址从 D7200 开始

7.2 模拟量模块

模拟量模块的供电电源为 DC24V。模拟量模块总数不超过 2 个时，可用运动控制器 DC24V 给模块供电。

型号	通道数	范围	分辨率	功能
电压·电流输入				
EX1-4AD	4 通道	电压: DC 0V~10V	0.6 mV (14 位)	电压·电流输入 可混合使用
		电流: DC 0~20mA	1.3uA (14 位)	
电压·电流输出				
EX1-4DA	4 通道	电压: DC 0V~10V	2.5mV (12 位)	电压·电流输出 可混合使用
		电流: DC 0~20mA	5uA (12 位)	
电压·电流输入 / 输出混合				
EX1-2AD/2DA	输入 2 通道	电压: DC 0V~10V	0.6mV (14 位)	电压·电流输入 可混合使用
		电流: DC 0~20mA	1.3uA (14 位)	
	输出 2 通道	电压: DC 0V~10V	2.5mV (12 位)	电压·电流输出 可混合使用
		电流: DC 0~20mA	5uA (12 位)	
温度传感器输入				
EX1-4PT	4 通道	-30 °C~500 °C	0.03 °C (14 位)	支持铂电阻 Pt100
EX1-4TC	4 通道	K 型: 0~1350 °C	0.1 °C (14 位)	支持 K 型、E 型、S 型、B 型热电偶。
		E 型: 0~600 °C	0.03 °C (14 位)	
		S 型: 0~1600 °C	0.1 °C (14 位)	
		B 型: 60~1800 °C	0.12 °C (14 位)	

7.2.1 模块规格

EX1-4AD (含 EX1-2AD/2DA 的 AD 输入) 产品规格:

项目	规格	
环境温度	0~50°C……工作时 -25~75°C……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D 类接地(接地电阻: 100 Ω 以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体; 导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 200mA; EX1-4AD 启动电流 1000mA=1A。(需要在端子排上连接 DC24V 电源供电)。	
模拟量输入范围	电压输入: DC 0V~+10V; (输入电阻 200k Ω)	电流输入: DC 0mA~+20mA; (输入电阻 250 Ω)
最大绝对输入	15V	30mA
AD 值范围	无符号 16 位, 稳定 14 位	无符号 16 位, 稳定 14 位
AD 转换时间	80ms/通道(含滤波); 4 通道 320ms。每通道间存在数字开关切换, 不适用于快速测量。	

EX1-4DA (含 EX1-2AD2DA 的 DA 输出) 产品规格:

项目	规格	
环境温度	0~50℃……工作时 -25~75℃……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D 类接地(接地电阻: 100 Ω 以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体; 导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 200mA; EX1-4DA 启动电流 1000mA=1A。(需要在端子排上连接 DC24V 电源供电) 内部 5V 电源由 DC/DC 转换而来。	
模拟量输出范围	DC 0V~10V (外部负载≥2KΩ)	DC 0mA~20mA (外部负载≤500Ω)
DA 值范围	12 位 二进制	
分辨率	2.5mV (10V/4000)	5uA (20mA/4000)
DA 转换时间	<2ms/通道, 4 通道< 8ms。	

EX1-4TC 产品规格:

项目	规格	
环境温度	0~50℃……工作时 -25~75℃……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D 类接地(接地电阻: 100 Ω 以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体; 导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 50mA; EX1-4TC 启动电流 400mA(需在端子排上连接 DC24V 电源供电)	
模拟量输入范围	0-55 mV	-
最大绝对输入		-
AD 值范围	无符号 16 位, 稳定 14 位	无符号 16 位, 稳定 14 位
AD 转换时间	80ms/通道(含滤波); 4 通道 400ms。每通道间存在数字开关切换, 不适用于快速测量。	

EX1-4PT 产品规格:

项目	规格	
环境温度	0~50℃……工作时 -25~75℃……保存时	
相对湿度	5~85%RH(无结露)……工作时	
接地	D 类接地(接地电阻: 100 Ω 以下)<不可以和强电系统共同接地>	
使用环境	无腐蚀性、可燃性气体; 导电性尘埃(灰尘)不严重	
工作电源	DC24V±10% 50mA; EX-4PT 启动电流 400mA。(需要外部 24V 供电)	
输入信号	3 线式 Pt100	
测温范围	-30~500℃	
AD 值范围	无符号 16 位, 稳定 14 位	
AD 转换时间	80ms/通道(含滤波); 4 通道 320ms。每通道间存在数字开关切换, 不适用于快速测量。	

● 用户可通过修改 D7144 的值来改变读取模拟量模块输入转换值的时间, 设置范围为 1~255, 默认值为 60ms, 即 60ms 读取一次模拟量输入转换值, 设置时应考虑模拟量输入模块的转换速度, 设置值应大于模块的转换速度。

7.2.2 模块读写控制

模拟量模块地址是自动顺序逻辑分配的。

设置值包括扩展模块的 PGA 设置值(不包括 PT100 扩展模块)和模拟量输出 DA 设置值(含 DA 输出的扩展模块)，另外每个设置值在内存里还分配了相应单元的内存作为读写控制缓存区。

输入 PGA 设置

模拟量输入信号有电流信号输入(DC 0~20mA)，也有电压信号输入(DC 0V~10V)，还有不同分度号的热电偶输入，因此必须选择输入类型，从而获取对应类型的转换值。一个通道输入类型选择占用一个数据寄存器。

向寄存器写入输入信号的设置类型，和模拟量输出通道设置输出值，同属写入操作，因此输入类型设置和输出值设置，同一顺序编号。

输入信号类型设置：

输入信号类型	K 分度	E 分度	S 分度	B 分度	电压	电流
PGA	0	1	2	3	5	6

当输入信号为 PT100 时，无需设置信号类型，不占用设置数据寄存器。

以下例说明 PGA 的输入信号设置。序号是从主机向外进行编号的。

型号	输入寄存器		输出寄存器		内部缓存区占用
EX1-4AD	D7200	AD1 标准化值 0~20000	D7204	AD1 类型选择：电压或电流	D7208~D7211
	D7201	AD2 标准化值 0~20000	D7205	AD2 类型选择：电压或电流	
	D7202	AD3 标准化值 0~20000	D7206	AD3 类型选择：电压或电流	
	D7203	AD4 标准化值 0~20000	D7207	AD4 类型选择：电压或电流	
EX1-4PT (0.1 ℃)	D7212	PT1 温度值	不占用输出寄存器	无需设置	无
	D7213	PT2 温度值			
	D7214	PT3 温度值			
	D7215	PT4 温度值			
EX1-4TC (0.1 ℃)	D7216	TC1 温度值	D7220	TC1 设置 (0, 1, 2, 3)	D7224~D7227
	D7217	TC2 温度值	D7221	TC2 设置 (0, 1, 2, 3)	
	D7218	TC3 温度值	D7222	TC3 设置 (0, 1, 2, 3)	
	D7219	TC4 温度值	D7223	TC4 设置 (0, 1, 2, 3)	
EX1-2AD/2DA	D7228	AD1 标准化值 0~20000	D7230	AD1 设置(5, 6)	D7236~D7241
	D7229	AD2 标准化值 0~20000	D7231	AD2 设置(5, 6)	
			D7232	DA1 设置(5, 6)	
			D7233	DA2 设置(5, 6)	
			D7234	DA1 的 DA 输出值	
			D7235	DA2 的 DA 输出值	

输出 PGA 设置

模拟量输出有电流输出(DC 0~20mA)和电压输出(DC 0V~10V)；EX1-4DA 或 EX1-2AD/2DA 均支持电压/电流输出，每路 DA 按设置的信号类型进行输出，因此每路 DA 既有类型选择寄存器，也有 DA 输出寄存器。

输出信号类型	电压	电流
PGA	5	6

以下说明 PGA 的输出信号设置。序号是从主机向外进行编号的。

型号	输入寄存器		输出寄存器		内部缓存区占用
EX1-4DA			D7200	DA1 类型设置 (5, 6)	D7208~D7215
			D7201	DA2 类型设置 (5, 6)	
			D7202	DA3 类型设置 (5, 6)	
			D7203	DA4 类型设置 (5, 6)	
			D7204	DA1 的 DA 输出值 (0~4000)	
			D7205	DA2 的 DA 输出值 (0~4000)	
			D7206	DA3 的 DA 输出值 (0~4000)	
		D7207	DA4 的 DA 输出值 (0~4000)		
EX1-2AD/2DA	D7216	AD1 标准化值 0~20000	D7218	AD1 类型设置 (5, 6)	D7224~D7229
	D7217	AD2 标准化值 0~20000	D7219	AD2 类型设置 (5, 6)	
			D7220	DA1 类型设置 (5, 6)	
			D7221	DA2 类型设置 (5, 6)	
			D7222	DA1 的 DA 输出值 (0~4000)	
			D7223	DA2 的 DA 输出值 (0~4000)	

7.2.3 输入模拟量

模拟量输入模块的标准化值或温度值，经模块转换后，分别对应到 D7200 开始的数据寄存器中。由主机向外读取 AD 标准化值或温度值。在读取输入值前，必须设置输入类型（输入 PGA 设置）。

如系统有三个模拟量输入模块，由主机向外的顺序是 EX1-4AD（电流型输入）、EX1-4PT（蒸汽温度测量）、EX1-4TC（S 型电偶、炉膛温度测量）。则相关通道的输入值对应应在下列数据寄存器中。

型号	输入信号转换值	输入类型设置	内部缓存区占用
EX1-4AD	D7200 = (0~20000)	D7204=6	D7208~D7211
	D7201 = (0~20000)	D7205=6	
	D7202 = (0~20000)	D7206=6	
	D7203 = (0~20000)	D7207=6	
EX1-4PT (0.1 °C)	D7212 = (-300~5000)	PT100 无类型选择	无
	D7213 = (-300~5000)		
	D7214 = (-300~5000)		
	D7215 = (-300~5000)		
EX1-4TC (0.1 °C)	D7216 = (0~16000)	D7220=2	D7224~D7227
	D7217 = (0~16000)	D7221=2	
	D7218 = (0~16000)	D7222=2	
	D7219 = (0~16000)	D7223=2	

7.2.4 输出模拟量

输出模拟量模块可设置成电压输出，也可设置成电流输出，因此，每一个通道的 DA 输出之前，也必须进行输出信号类型设置（输出 PGA 设置）。PGA 设置和 DA 值输出均是用户向 PLC 规定的寄存器写数据，规划到以 D7256 开始的连续区域。对输出而言，先设置类型，后设置输出。对混合模块（有输入和输出）而言，先分配输入类型寄存器，次分配输出类型寄存器，最后分配 DA 输出值寄存器。

假设系统有一个 EX1-4DA（两路电压输出，两路电流输出），则信号类型设置寄存器和 DA 值输出寄存器分配如下：

型号	输出信号类型设置	DA 值输出寄存器	内部缓存区占用
EX1-4DA	D7200=5（电压）	D7204 = (0~4000)	D7208~D7215
	D7201=5（电压）	D7205 = (0~4000)	
	D7202=6（电流）	D7206 = (0~4000)	
	D7203=6（电流）	D7207 = (0~4000)	

输入输出混合

模拟量输出有电流输出 (DC 0~20mA) 和电压输出 (DC 0V~10V)；EX1-4DA 或 EX1-2AD/2DA 均支持电压/电流输出，每路 DA 按设置的信号类型进行输出，因此每路 DA 既有类型选择寄存器，也有 DA 输出寄存器。

假设系统挂有 EX1-4AD（2 路电压输入、2 路电流输入）、EX1-4TC（2 路 K 分度、2 路 S 分度）、EX1-2AD2DA（2 路电压输入、2 路电流输出）、EX1-4DA（4 路均为电流输出）共四个模拟量模块。

数据寄存器分配应符合以下三条：

- ◆ 对每个模块（无论输入输出）而言，必须先设置信号类型（先输入后输出），再读取 AD 值或输出 DA 值。当输入信号是 PT100 时，不进行信号类型设置。
- ◆ 模拟量的地址分配是从 D7200 开始，顺次按照从站类型向后排序。

对上述系统进行数据寄存器分配。

型号	输入类型设置	输出类型设置	AD 值寄存器	DA 值寄存器
EX1-4AD	D7204=5（电压）		D7200=(0~20000)	
	D7205=5（电压）		D7201=(0~20000)	
	D7206=6（电流）		D7202=(0~20000)	
	D7207=6（电流）		D7203=(0~20000)	
EX1-4TC	D7216=0（K）		D7212=(0~13500)	
	D7217=0（K）		D7213=(0~13500)	
	D7218=2（S）		D7214=(0~16000)	
	D7219=2（S）		D7215=(0~16000)	
EX1-2AD2DA	D7222=5（电压）	D7224=6（电流）	D7220=(0~20000)	D7226=(0~4000)
	D7223=5（电压）	D7225=6（电流）	D7221=(0~20000)	D7227=(0~4000)
EX1-4DA		D7234=6（电流）		D7238=(0~4000)
		D7235=6（电流）		D7239=(0~4000)
		D7236=6（电流）		D7240=(0~4000)
		D7237=6（电流）		D7241=(0~4000)

7.3 开关量模块

运动控制器最多可接受总和最多 120 个 X 输入扩展点和 120 个 Y 输出扩展点。

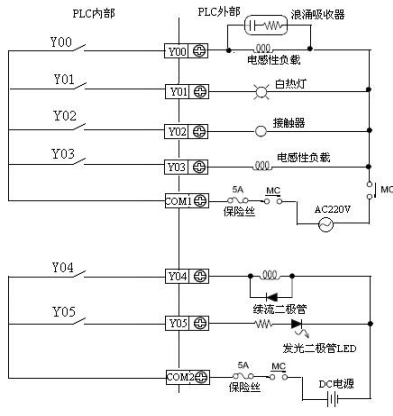
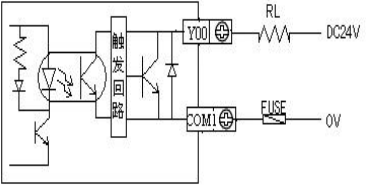
型号	供电电源	合计点数	输入输出特性			
			普通输入	输入电压	普通输出	输出方式
EX1-08M	DC24V	8	8	DC24V	/	/
EX1-08R	总线供电	8	/	/	8	继电器
EX1-08T	总线供电	8	/	/	8	晶体管
EX1-16M	DC24V	16	16	DC24V	/	/
EX1-16R	总线供电	16	/	/	16	继电器
EX1-16T	总线供电	16	/	/	16	晶体管
EX1-16M16R	AC220V	32	16	DC24V	16	继电器
EX1-16M16T	AC220V	32	16	DC24V	16	晶体管

7.3.1 模块规格

开关量输入模块:

项目	规格
	X020~X267
输入电压	DC24V ± 10%
输入电流	4.5mA
输入阻抗	5.1 k Ω
输入 ON 电流	2.3mA 以上
输入 OFF 电流	2.2mA 以下
输入响应时间	约 16 ms
输入信号形式	无电压触点, 或 NPN 集电极开路晶体管
输入电路隔离	光电耦合器隔离
输入状态显示	光电耦合器被驱动时面板 LED 灯亮

开关量输出模块:

项目	继电器输出	晶体管输出
	Y20~Y267	Y20~Y267
外部电源	AC250V; DC30V 以下	DC5~30V
电路绝缘	机械绝缘	光耦绝缘
动作显示	继电器线圈通电时面板 LED 灯亮	光耦驱动时面板 LED 灯亮
最大电阻负载	2A/点 8A/4 点公共端 8A/8 点公共端	0.5A/点; 0.8A/4 点公共 1.6A/8 点公共
最大感性负载	80VA (注 1)	12W/DC24V
最大指示灯负载	100W (注 2)	1.5W/DC24V
开路漏电流	-	0.1 mA /DC30V
最小负载	DC5V 2 mA	-
响应时间 OFF->ON	约 10ms	0.2ms 以下
响应时间 ON->OFF	约 10ms	0.2ms 以下
		<p>负载容量越大，继电器输出接点寿命越短，负载容量<30VA，接点寿命大于 20 万次。</p> <p>因输出接点 OFF 时，没有电流，所以可以直接驱动氖灯等。</p> <p>当继电器输出型驱动直流回路的负载为感性（如继电器线圈）时，用户电路需并联续流二极管。</p> <p>当继电器输出驱动交流回路的负载为感性时，用户电路需并联 RC 浪涌吸收电路，以保护输出继电器触点。</p> <p>原则上继电器输出端口不直接接入容性负载，若有必要，需保证其冲击浪涌电流小于规格说明中的最大电流。</p>
		<p>晶体管输出端口须遵守允许最大电流限制，以保证输出端口的发热限制在允许范围。若有多个晶体管端口输出电流大于 100mA，则应均匀分布于输出端口，不宜安排在相邻的输出端口，以便散热；建议同时为 ON（导通）状态的输出点数不要长时间超过总输出点数的 70%。</p>

7.3.2 模块读写控制

开关量模块和模拟量模块配置在一起时，开关量和模拟量的地址分配相互不影响，各自独立分配。

运动控制器接入开关量模块时，地址是按照 8 进制递推顺序自动分配的，以字节为单位，主机或模块上未
满 8 位的按 8 位算。

比如顺序接入 EX1-08M，EX1-08R，EX1-16M，EX1-16R，那么主机和各模块对应的地址分配如下：

连接顺序	产品型号	输入地址	输出地址
主机	HM	X000~X007 X010~X017	Y000~Y007 Y010~Y015
模块一	EX1-08M	X020~X027	/
模块二	EX1-08R	/	Y020~Y027
模块三	EX1-16M	X030~X037 X040~X047	/
模块四	EX1-16R	/	Y030~Y037 Y040~Y047

第八章 运动控制

运动控制系统是以电机为对象，以电力电子（功率）变换器为弱电控制的媒介，以自动控制理论为分析和设计基础，以电子线路或计算机为控制手段，实现物体位置、速度等物理量控制。

随着技术的发展，在运动控制市场涌现出越来越多的问题解决方案。虽然解决问题的方案变多了，但是各种方案间的不兼容使得运动控制工程变得复杂，终端用户的学习也变得混乱，因而运动控制市场的发展也随之减慢。这时，就需要一个标准化的规则来规范运动控制的市场，从编程语言、方案解决界面等方面加以规范，这样一来就减少了方案对硬件的依赖，同时也增加了软件的可重用性。

运动指令标准

PLCopen 国际组织是独立于厂商的全球性机构，它致力于解决运动控制方面的问题，支持使用该领域国际标准。PLCopen 已成为推动工控变成语言标准化实施的支柱，它颁布的 IEC61131-3 获得了全球控制业界的广泛承认和支持。

IEC61131-3 语言适用于运动控制，它规范了数据类型定义了相关运动功能块，功能块的内容可以使用在任何一种编程语言中。本规范还是一个开放性的框架，没有硬件的依赖性。

8.1 运动控制指令

运动控制器是以 PLCopen 为背景进行的模块化研发，以梯形图为编程语言，以指令的形式进行编程的。这些指令的应用能够解决大多数运动控制系统以及开放式数控系统。

助记符、名称	功能
轴参数配置	
AXPO	轴端口指定
AXPU	轴当量指定
AXSP	轴速指定
AXAS	轴加减速指定
AXEI	轴事件输入点指定
AXOSP	回原点速度指定
AXOX	回原点参数指定
单轴基本运动	
HOME	回原点
JOG	点动
PSOUT	定位
PTAB	制表
PTOUT	表格输出
IPL	插补
DPLSY	定速定长输出
DPLSR	加减速定长输出
DDRVI	相对定位
DDRVA	绝对定位
DDVIT	中断定位
DPLSV	变速输出

助记符、名称	功能
DPTO	多段速输出
CAP	高速读
DHMOV	高速读出
DHSCS	高速比较置位
DHSCR	高速比较复位
DHSZ	高速区间比较
DHSCT	高速表格比较输出
单轴关联运动	
GEAR	电子齿轮
CAM	电子凸轮
PCTAB	追剪正程/滚切制表
PBTAB	追剪返程制表
单轴关联运动	
GEAR	电子齿轮
CAM	电子凸轮
PCTAB	追剪正程/滚切制表
PBTAB	追剪返程制表

8.2 运动轴对象

8.2.1 运动轴定义

对于一个复杂的运动系统用单一坐标系很难直接描述，那么就需要多坐标系即轴组的形式来进行相互之间的解耦，进行运算后，再对多坐标系进行合成，最后进行输出。

通常所说的运动轴一般分为实轴和虚轴。实轴为物理上存在的，如脉冲输出端口；虚轴为数字虚拟轴，可以直接数字输出叠加到实轴上，虚拟中间轴为特殊的软件虚拟轴，只做过渡作用，不实际输出。

运动控制器对于运动轴的划分为输入轴 (AXIN)，输出轴 (AXOUT) 以及软件虚拟轴 (AXV)。而运动指令所针对的轴对象为软件虚拟输出轴，故需把实轴 (即物理轴) 分配到软件虚拟输出轴上，才能通过运动指令关联实轴进行运动控制。

输入轴

输入轴共有 2 个，定义如下：

- AXIN0: 脉冲端口输入 (对应 X0 端口)
- AXIN1: 脉冲端口输入 (对应 X2 端口)

输出轴

输出轴共有 6 个，定义如下：

- AXOUT0: 脉冲端口输出 (对应 Y0 端口)
- AXOUT1: 脉冲端口输出 (对应 Y1 端口)
- AXOUT2: 脉冲端口输出 (对应 Y2 端口)
- AXOUT3: 脉冲端口输出 (对应 Y3 端口)
- AXOUT4: 脉冲端口输出 (对应 Y4 端口)
- AXOUT5: 脉冲端口输出 (对应 Y5 端口)
- AXOUT6: 数字输出轴 0 输出到内存
- AXOUT7: 数字输出轴 1 输出到内存
- AXOUT8: 数字输出轴 2 输出到内存

软件虚拟轴

- 软件虚拟输入轴

运动控制器将相应的 C 寄存器作为软件虚拟输入轴的输入源，其中 C 寄存器地址与输入轴的关系可参考章节 C 计数器描述。

- ◆ AXV6: 绑定硬件 AXIN0
- ◆ AXV7: 绑定硬件 AXIN1

- 软件虚拟输出轴

- ◆ AXV0: 可绑定硬件或数字输出轴 (AXOUT0~AXOUT8)
- ◆ AXV1: 可绑定硬件或数字输出轴 (AXOUT0~AXOUT8)
- ◆ AXV2: 可绑定硬件或数字输出轴 (AXOUT0~AXOUT8)
- ◆ AXV3: 可绑定硬件或数字输出轴 (AXOUT0~AXOUT8)
- ◆ AXV4: 可绑定硬件或数字输出轴 (AXOUT0~AXOUT8)
- ◆ AXV5: 可绑定硬件或数字输出轴 (AXOUT0~AXOUT8)

运动控制器输出轴有 6 个脉冲输出口 (Y0~Y5)，可以分别绑定至不同的软件虚拟输出轴上。如果将一个脉冲输出口绑定多个软件虚拟输出轴，其实就是在这个输出轴上进行运动叠加。如最基础的直线插补就是两个软件虚拟输出轴输出到同一个脉冲输出口上，一个轴走位置，另一个轴必要情况下做脉冲插补。

8.2.2 运动输入轴参数配置

输入轴参数配置指令针对软件虚拟输入轴进行配置，其配置对象为 AXV6~AXV7。

轴参数在 PLC 的内存分配地址如下表所示(每个轴分配 50 个 D 寄存器)：

轴号	对应参数存放的内存首地址
AXV6	D200
AXV7	D250

轴参数配置的方式：

- 梯形图使用 MOV 指令对其寄存器直接进行填写

轴参数定义

软件虚拟输入轴参数的定义如下表所示。地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

轴首地址+偏移	名称	参数描述	单位
轴首地址+0(L)	高速计数绝对值	该虚拟输入轴接入的脉冲计数绝对值。 单位为脉冲。该参数为 32 位数值，占用 2 个 D 寄存器。 该参数无需设置，为显示值。 ● D200 对应 X0 输入端子的高速计数。 ● D250 对应 X2 输入端子的高速计数。	脉冲
轴首地址+2(L)	高速计数绝对值高 32 位	该虚拟输入轴接入的脉冲计数绝对值的高 32 位。 该参数无需设置，为显示值。	脉冲
轴首地址+4(L)	周期计数值	该虚拟输入轴接入的脉冲周期计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+6(L)	外部中断计数值	当产生外部中断时，该虚拟输入轴接入的脉冲计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+8(L)	计数器中断计数值	当产生计数器中断时，该虚拟输入轴接入的脉冲计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+10	脉冲计数方式	该虚拟输入轴的脉冲计数方式。在首次输出(OUT)或切换高速计数器时更新 ● = 0, AB 相计数 ● = 1, 脉冲+方向计数 ● = 2, 增计数 ● = 3, 减计数	-
轴首地址+11	脉冲捕获方式	设置该虚拟输入轴的脉冲捕获方式。 ● H00: D211 对应 X4 下降沿, D261 对应 X6 下降沿。 ● H10: D211 对应 X4 下降沿, X5 下降沿, D261 对应 X6 下降沿, X7 下降沿。 ● H20: D211 对应 X4 下降沿, X5 上升沿, D261 对应 X6 下降沿, X7 上升沿。 ● H11: D211 对应 X5 下降沿, D261 对应 X7 下降沿。 ● H21: D211 对应 X5 上升沿, D261 对应 X7 上升沿。 ● H31: X 端子不对应外部输入中断。	-
轴首地址+12	周期 1MS 计数值	该虚拟输入轴接入的脉冲 1MS 周期计数值。 该参数无需设置，为显示值。	脉冲
轴首地址+13 ~ 轴首地址+49	保留，用户无需配置。		

8.2.3 运动输出轴参数配置

输出轴参数配置指令针对软件虚拟输出轴进行配置，其配置对象为 AXV0~AXV5。软件虚拟输出轴既可以绑定物理输出端口 Y0~Y5，又可以绑定数字输出轴 0~2。

轴参数在 PLC 的内存分配地址如下表所示(每个轴分配 50 个 D 寄存器)：

轴号	对应参数存放的内存首地址
AXV0	D300
AXV1	D350
AXV2	D400
AXV3	D450
AXV4	D500
AXV5	D550

轴参数配置的方式：

- 梯形图使用 MOV 指令对其寄存器直接进行填写
- 梯形图使用轴参数指令进行设置(下述章节介绍)

轴参数定义

软件虚拟输出轴参数的定义如下表所示。地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

轴首地址+偏移	名称	参数描述	单位																					
轴首地址+0	轴输出绑定	<p>指定轴脉冲输出端口 AXOUT0~AXOUT5，对应数值 0~5。</p> <p>默认值为当前轴参数内存地址对应的 Y 端口序号。比如 D300=0，D350=1，... D550=5。</p> <ul style="list-style-type: none"> ● = 0/1/2/3/4/5，绑定 Y00/Y01/Y02/Y03/Y04/Y05 高速脉冲输出端口到该轴 ● = 6/7/8，绑定数字轴 0/数字轴 1/数字轴 2 到该轴 	-																					
轴首地址+1	轴脉冲输出方向使能	<p>使能轴脉冲输出方向端口。默认值为 0。该参数配置方式如下：</p> <ul style="list-style-type: none"> ● = 0，不绑定方向端口到该轴 ● = 1，方向端口绑定情况如下表所示： <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>轴输出绑定 =</th> <th>输出端口</th> <th>脉冲方向口</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Y00</td> <td>Y10</td> </tr> <tr> <td>1</td> <td>Y01</td> <td>Y11</td> </tr> <tr> <td>2</td> <td>Y02</td> <td>Y12</td> </tr> <tr> <td>3</td> <td>Y03</td> <td>Y13</td> </tr> <tr> <td>4</td> <td>Y04</td> <td>Y14</td> </tr> <tr> <td>5</td> <td>Y05</td> <td>Y15</td> </tr> </tbody> </table> <p>* 如果该虚拟轴绑定的是数字轴，则轴脉冲输出方向使能填 0 即可。</p>	轴输出绑定 =	输出端口	脉冲方向口	0	Y00	Y10	1	Y01	Y11	2	Y02	Y12	3	Y03	Y13	4	Y04	Y14	5	Y05	Y15	-
轴输出绑定 =	输出端口	脉冲方向口																						
0	Y00	Y10																						
1	Y01	Y11																						
2	Y02	Y12																						
3	Y03	Y13																						
4	Y04	Y14																						
5	Y05	Y15																						

轴首地址+偏移	名称	参数描述	单位																																								
轴首地址+2	轴时间当量	轴时间当量设置。默认值为 10。用来测量后述速度/加速度参数的计量标准。 比如该值设置为 10，则时间当量 = 10ms(推荐值)。 ● 设置为负数时，则取 16 位无符号数作为计算。比如设置-1，则认为是 65535。	ms																																								
轴首地址+3	轴脉冲当量	该参数保留使用，可随意填写该值，默认值 0。	脉冲 / 轴长度当量																																								
轴首地址+4	轴长度当量	该参数保留使用，可随意填写该值，默认值 0。	mm																																								
轴首地址+5	轴最高限速	轴运行最高速度限制。默认值 1000。该值必须大于 0。 例如设置最高速度为 80KHz，即每 ms 发送 80 个脉冲。假定轴时间当量=10，那么 轴最高限速 = 80 * 10 = 800。	脉冲 / 轴时间当量																																								
轴首地址+6	轴起始速度	轴运行起始速度(起跳速度)。默认值 0。该值必须大于等于 0。 例如设置速度为 200Hz，即每 ms 发送 0.2 个脉冲。假定轴时间当量=10，那么 轴起始速度 = 0.2 * 10 = 2。为保证控制对象的平稳启动，该参数推荐填 0。	脉冲 / 轴时间当量																																								
轴首地址+7	轴最大加减速	轴运行最大加/减速度。默认值 100。该值必须大于 0。 例如设置加速度为 0.5 脉冲/ms ² ，假定轴时间当量 = 10，那么轴最大加减速 = 0.5 * 10 ² = 50。应结合负载变化设置该参数数值。对较轻负载，较大加减速能获得良好的加减速性能；对较重负载，加减速不合理会导致实际输出始终滞后。	脉冲 / 轴时间当量 ²																																								
轴首地址+8	近点信号端口	近点信号输入端口指定（按位设置，数值为 X 端口掩码），回原点需配置该参数。 默认值为 0。 <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>近点信号端口 =</th> <th>选定端口</th> <th>近点信号端口 =</th> <th>选定端口</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>不选定端口</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>X0/M8096</td> <td>256</td> <td>X10/M8104</td> </tr> <tr> <td>2</td> <td>X1/M8097</td> <td>512</td> <td>X11/M8105</td> </tr> <tr> <td>4</td> <td>X2/M8098</td> <td>1024</td> <td>X12/M8106</td> </tr> <tr> <td>8</td> <td>X3/M8099</td> <td>2048</td> <td>X13/M8107</td> </tr> <tr> <td>16</td> <td>X4/M8100</td> <td>4096</td> <td>X14(虚拟)/M8108</td> </tr> <tr> <td>32</td> <td>X5/M8101</td> <td>8192</td> <td>X15(虚拟)/M8109</td> </tr> <tr> <td>64</td> <td>X6/M8102</td> <td>16384</td> <td>X16(虚拟)/M8110</td> </tr> <tr> <td>128</td> <td>X7/M8103</td> <td>-32768</td> <td>X17(虚拟)/M8111</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ● 通过轴事件输入点.Bit13 可使用 M8096~M8111 代替 X00~X17 的控制。 ● 虚拟端口 X14~X17 可通过 M8160~M8163 进行控制。 ● 端口响应频率为 5KHz。 	近点信号端口 =	选定端口	近点信号端口 =	选定端口	0	不选定端口			1	X0/M8096	256	X10/M8104	2	X1/M8097	512	X11/M8105	4	X2/M8098	1024	X12/M8106	8	X3/M8099	2048	X13/M8107	16	X4/M8100	4096	X14(虚拟)/M8108	32	X5/M8101	8192	X15(虚拟)/M8109	64	X6/M8102	16384	X16(虚拟)/M8110	128	X7/M8103	-32768	X17(虚拟)/M8111	-
近点信号端口 =	选定端口	近点信号端口 =	选定端口																																								
0	不选定端口																																										
1	X0/M8096	256	X10/M8104																																								
2	X1/M8097	512	X11/M8105																																								
4	X2/M8098	1024	X12/M8106																																								
8	X3/M8099	2048	X13/M8107																																								
16	X4/M8100	4096	X14(虚拟)/M8108																																								
32	X5/M8101	8192	X15(虚拟)/M8109																																								
64	X6/M8102	16384	X16(虚拟)/M8110																																								
128	X7/M8103	-32768	X17(虚拟)/M8111																																								
轴首地址+9	Z 脉冲信号端口	反馈编码器 Z 脉冲信号输入端口指定，回原点需要配置该参数。默认值为 0。 端口选定设置值参考近点信号端口。该端口只能配置到 X0~X7 端子上。	-																																								
轴首地址+10	左限位端口	运动轴左限位端口指定，回原点需要配置该参数。默认值为 0。 端口选定设置值参考近点信号端口。	-																																								
轴首地址+11	右限位端口	右限位端口指定，回原点需要配置该参数。默认值为 0。 端口选定设置值参考近点信号端口。	-																																								
轴首地址+12	原点回归速度	原点回归速度指定，回原点需要配置该参数。默认值为 0。 计算方法参考轴最高限速参数说明。	脉冲 / 轴时间当量																																								
轴首地址+13	原点爬行速度	原点爬行速度指定，回原点需要配置该参数。默认值为 0。 计算方法参考轴最高限速参数说明。	脉冲 / 轴时间当量																																								

轴首地址+偏移	名称	参数描述	单位																																		
轴首地址+14	轴事件输入点	<p>轴输入点属性指定。默认值为0。按位设置模式功能，其定义如下：</p> <table border="1"> <thead> <tr> <th>回原点模式, Bit</th> <th>定义</th> </tr> </thead> <tbody> <tr> <td>Bit0</td> <td>= 0: 近点信号端口不使能 = 1: 近点信号端口使能</td> </tr> <tr> <td>Bit1</td> <td>= 0: Z脉冲信号端口不使能 = 1: Z脉冲信号端口使能</td> </tr> <tr> <td>Bit2</td> <td>= 0: 左限位端口不使能 = 1: 左限位端口使能</td> </tr> <tr> <td>Bit3</td> <td>= 0: 右限位端口不使能 = 1: 右限位端口使能</td> </tr> <tr> <td>Bit4</td> <td>近点信号端口极性设置。 = 0: 近点信号端口信号进入为 ON, 离开为 OFF = 1: 近点信号端口信号进入为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit5</td> <td>Z脉冲信号端口极性设置。 参考近点信号端口极性设置。</td> </tr> <tr> <td>Bit6</td> <td>左限位信号端口极性设置。 参考近点信号端口极性设置。</td> </tr> <tr> <td>Bit7</td> <td>右限位信号端口极性设置。 参考近点信号端口极性设置。</td> </tr> <tr> <td>Bit8</td> <td>中断信号端口极性设置。 参考近点信号端口极性设置。</td> </tr> <tr> <td>Bit9</td> <td>= 0: 中断信号端口不使能 = 1: 中断信号端口使能</td> </tr> <tr> <td>Bit10</td> <td>急停信号端口极性设置。 参考近点信号端口极性设置。</td> </tr> <tr> <td>Bit11</td> <td>= 0: 急停信号端口不使能 = 1: 急停信号端口使能</td> </tr> <tr> <td>Bit12</td> <td>= 0: 急停产生时立即停止 = 1: 急停产生时减速停止 ● 配置为1时, 必须对应的运动指令支持</td> </tr> <tr> <td>Bit13</td> <td>输入信号源选择。 = 0: 指向 M8096~M8111 = 1: 指向 X00~X17</td> </tr> <tr> <td>Bit14</td> <td>回原点时, 近点信号到开始脉冲计数时的边沿选择。 =0: 下降沿 =1: 上升沿</td> </tr> <tr> <td>Bit15</td> <td>回原点停止方式(过近点信号后生效) = 0: 先计数过近点Z脉冲数个Z脉冲, 然后再计数 过近点脉冲数个脉冲停止回原点流程。 = 1: 计数过近点Z脉冲数个Z脉冲, 或计数过近点 脉冲数个脉冲回原点流程。</td> </tr> </tbody> </table> <p>● 若端口不使能, 则其极性设置和端口指定均不生效。 例如轴事件输入点, Bit2 = 0, 则轴事件输入点, Bit6 和左限位端口设置均无效。</p>	回原点模式, Bit	定义	Bit0	= 0: 近点信号端口不使能 = 1: 近点信号端口使能	Bit1	= 0: Z脉冲信号端口不使能 = 1: Z脉冲信号端口使能	Bit2	= 0: 左限位端口不使能 = 1: 左限位端口使能	Bit3	= 0: 右限位端口不使能 = 1: 右限位端口使能	Bit4	近点信号端口极性设置。 = 0: 近点信号端口信号进入为 ON, 离开为 OFF = 1: 近点信号端口信号进入为 OFF, 离开为 ON	Bit5	Z脉冲信号端口极性设置。 参考近点信号端口极性设置。	Bit6	左限位信号端口极性设置。 参考近点信号端口极性设置。	Bit7	右限位信号端口极性设置。 参考近点信号端口极性设置。	Bit8	中断信号端口极性设置。 参考近点信号端口极性设置。	Bit9	= 0: 中断信号端口不使能 = 1: 中断信号端口使能	Bit10	急停信号端口极性设置。 参考近点信号端口极性设置。	Bit11	= 0: 急停信号端口不使能 = 1: 急停信号端口使能	Bit12	= 0: 急停产生时立即停止 = 1: 急停产生时减速停止 ● 配置为1时, 必须对应的运动指令支持	Bit13	输入信号源选择。 = 0: 指向 M8096~M8111 = 1: 指向 X00~X17	Bit14	回原点时, 近点信号到开始脉冲计数时的边沿选择。 =0: 下降沿 =1: 上升沿	Bit15	回原点停止方式(过近点信号后生效) = 0: 先计数过近点Z脉冲数个Z脉冲, 然后再计数 过近点脉冲数个脉冲停止回原点流程。 = 1: 计数过近点Z脉冲数个Z脉冲, 或计数过近点 脉冲数个脉冲回原点流程。	-
回原点模式, Bit	定义																																				
Bit0	= 0: 近点信号端口不使能 = 1: 近点信号端口使能																																				
Bit1	= 0: Z脉冲信号端口不使能 = 1: Z脉冲信号端口使能																																				
Bit2	= 0: 左限位端口不使能 = 1: 左限位端口使能																																				
Bit3	= 0: 右限位端口不使能 = 1: 右限位端口使能																																				
Bit4	近点信号端口极性设置。 = 0: 近点信号端口信号进入为 ON, 离开为 OFF = 1: 近点信号端口信号进入为 OFF, 离开为 ON																																				
Bit5	Z脉冲信号端口极性设置。 参考近点信号端口极性设置。																																				
Bit6	左限位信号端口极性设置。 参考近点信号端口极性设置。																																				
Bit7	右限位信号端口极性设置。 参考近点信号端口极性设置。																																				
Bit8	中断信号端口极性设置。 参考近点信号端口极性设置。																																				
Bit9	= 0: 中断信号端口不使能 = 1: 中断信号端口使能																																				
Bit10	急停信号端口极性设置。 参考近点信号端口极性设置。																																				
Bit11	= 0: 急停信号端口不使能 = 1: 急停信号端口使能																																				
Bit12	= 0: 急停产生时立即停止 = 1: 急停产生时减速停止 ● 配置为1时, 必须对应的运动指令支持																																				
Bit13	输入信号源选择。 = 0: 指向 M8096~M8111 = 1: 指向 X00~X17																																				
Bit14	回原点时, 近点信号到开始脉冲计数时的边沿选择。 =0: 下降沿 =1: 上升沿																																				
Bit15	回原点停止方式(过近点信号后生效) = 0: 先计数过近点Z脉冲数个Z脉冲, 然后再计数 过近点脉冲数个脉冲停止回原点流程。 = 1: 计数过近点Z脉冲数个Z脉冲, 或计数过近点 脉冲数个脉冲回原点流程。																																				

轴首地址+偏移	名称	参数描述	单位																		
轴首地址+15	过近点脉冲数	设置离开近点信号后需要计数的脉冲数再执行停止回原点流程。默认值为 0。 该参数为 16 位无符号数，超过 32767 将以负数形式设置/显示。	脉冲																		
轴首地址+16	过近点 Z 脉冲数	设置离开近点信号后需要计数的 Z 脉冲数再执行停止回原点流程。默认值为 0。 该参数为 16 位无符号数，超过 32767 将以负数形式设置/显示。	脉冲																		
轴首地址+17	保留，用户无需配置。																				
轴首地址+18(L)	轴相对位置	轴当前的相对位置，为当前轴位置插补增量的累积值。每个插补周期更新 1 次。 单位为脉冲。该参数为 32 位数值，占用 2 个 D 寄存器。 该参数无需设置，为显示值。	脉冲																		
轴首地址+20	轴状态	轴当前的运行状态。其数值为 0 时该轴处于空闲状态，可执行轴运动指令，其他数值情况下执行该轴的运动指令将被阻塞，不会生效，IPL 指令除外。 = 0: 当前无该轴的运动指令运行，可执行该轴的运动指令 = 16: 该轴的运动指令正在执行 = 32: 该轴的运动指令执行完成 = 64: 该轴设置/运行出错，出错代码可查看轴出错代码 = 144: 该轴正在执行缓停 = 160: 该轴处于急停状态 当检测到轴状态等于 32 或 160 时，说明针对该轴的某条运动指令已经执行完毕，此时可断开运动指令。当梯形图扫描到该运动指令断开后，自动将轴状态清 0。 该参数无需设置，为显示值。	-																		
轴首地址+21	轴出错代码	轴参数设置/运行出错代码。 <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>轴出错代码 =</th> <th>定义</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>无错</td> </tr> <tr> <td>1</td> <td>当前轴速度超硬件支持的极限速度</td> </tr> <tr> <td>2</td> <td>未绑定输出轴</td> </tr> <tr> <td>3</td> <td>轴最高限速超硬件支持的极限速度</td> </tr> <tr> <td>4</td> <td>当前轴速度低于轴起始速度</td> </tr> <tr> <td>5</td> <td>当前轴速度超过原点回归速度，回原点时检测</td> </tr> <tr> <td>6</td> <td>当前轴速度低于原点爬行速度，回原点时检测</td> </tr> <tr> <td>7</td> <td>轴起始速度大于轴最高限速</td> </tr> </tbody> </table> <p>该参数无需设置，为显示值。</p>	轴出错代码 =	定义	0	无错	1	当前轴速度超硬件支持的极限速度	2	未绑定输出轴	3	轴最高限速超硬件支持的极限速度	4	当前轴速度低于轴起始速度	5	当前轴速度超过原点回归速度，回原点时检测	6	当前轴速度低于原点爬行速度，回原点时检测	7	轴起始速度大于轴最高限速	-
轴出错代码 =	定义																				
0	无错																				
1	当前轴速度超硬件支持的极限速度																				
2	未绑定输出轴																				
3	轴最高限速超硬件支持的极限速度																				
4	当前轴速度低于轴起始速度																				
5	当前轴速度超过原点回归速度，回原点时检测																				
6	当前轴速度低于原点爬行速度，回原点时检测																				
7	轴起始速度大于轴最高限速																				
轴首地址+22	轴位置插补增量	轴当前插补周期(1ms)的位置增量。每个插补周期更新 1 次。 该参数无需设置，为显示值。	脉冲																		
轴首地址+23	轴急停事件	轴急停事件端口指定，按位进行设置。默认值为 0。 端口选定设置值参考近点信号端口。	-																		
轴首地址+24	轴中断事件	轴中断事件端口指定，按位进行设置。默认值为 0。 端口选定设置值参考近点信号端口。	-																		
轴首地址+25	保留，用户无需配置。																				
轴首地址+26(L)	轴中断位置	轴产生中断事件时的相对位置。 该参数无需设置，为显示值。	脉冲																		
轴首地址+28 ~ 轴首地址+49	保留，用户无需配置。																				

轴参数映射关系

软件虚拟输出轴参数对应 D 寄存器的映射关系为：轴参数起始地址+参数的偏移地址。
比如 AXV3 轴的 轴脉冲输出方向使能 参数，其对应 D 寄存器地址为：450+1=451。

- * 第 1 扫描周期结束时，运动轴参数将被初始化。使用 LD M8002 触点配置轴参数。若梯形图在第 1 扫描周期未填写初始化参数时，系统采用默认轴参数进行初始化。
- * 运动控制器默认将 Y0~Y5 配置到 6 个虚拟输出轴上，这样每个脉冲输出端口可输出的硬件最大频率为 100KHz (依据系统最大输出频率 600KHz 均分)。如果系统用到的轴数小于 6 个 (比如 3 个)，那么可以在第 1 扫描周期配置相应其他 3 个不用的轴参数轴输出绑定为 0，这样每个脉冲口可输出的最大频率为 200KHz。
- * 在配置轴最高限速参数时，不要将其配置为硬件可支持的最大频率，略小一点即可。比如硬件最大频率为 100KHz，轴时间当量为 10，那么轴最高限速可填写 999。
- * 运动控制器配置了 3 个数字轴，也可分配到 6 个虚拟输出轴上。
数字轴不占用硬件最大频率，其 轴最高限速 可配置为系统最大输出频率 600KHz。
- * 从 D80 开始的区域指定 AXOUT 输出轴所发的脉冲数，每个轴占用 2 个 D。

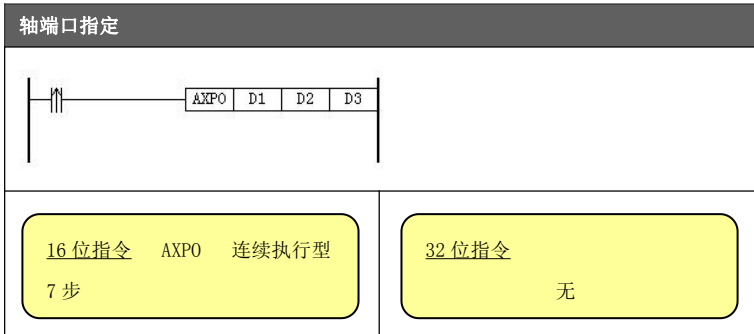
输出轴	输出轴对象	对应脉冲输出存放的地址 (32 位)
AXOUT0	Y00	D80
AXOUT1	Y01	D82
AXOUT2	Y02	D84
AXOUT3	Y03	D86
AXOUT4	Y04	D88
AXOUT5	Y05	D90
AXOUT6	数字轴 0	D92
AXOUT7	数字轴 1	D94
AXOUT8	数字轴 2	D96

软件虚拟输出轴的轴参数 轴相对位置 表示的是软件虚拟输出轴 AXV 所输出的脉冲数，而 D80 区域表示的是输出轴 AXOUT 所输出的脉冲数，两者不一定相等。

比如 AXV0 和 AXV1 的轴参数 轴输出绑定 设置为 1，那么 D318(L) 和 D368(L) 代表的是 AXV0 和 AXV1 对于 Y01 端口分别输出的脉冲数，而 D82(L) 则显示 Y01 输出端口总的输出脉冲数。显然 D82(L) 的数值是 D318(L) 和 D368(L) 之和。

8.2.3.1 轴端口指定[AXPO]

将实际输出轴 (AXOUT) 与软件虚拟输出轴 (AXV) 进行绑定。



操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	实际输出轴号	16 位 BIN
D3	轴方向使能	16 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D1	●									
D2	●									
D3	●	●	●							

功能与动作

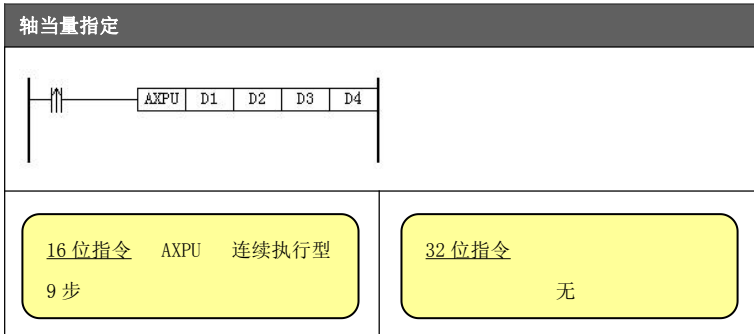
- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定输出轴号 (AXOUT0~AXOUT8)，对应轴参数 轴输出绑定 的赋值。
 - ◆ =0 指定 Y00 脉冲输出口
 - ◆ =1 指定 Y01 脉冲输出口
 - ◆ =2 指定 Y02 脉冲输出口
 - ◆ =3 指定 Y03 脉冲输出口
 - ◆ =4 指定 Y04 脉冲输出口
 - ◆ =5 指定 Y05 脉冲输出口
 - ◆ =6 指定数字输出轴 0
 - ◆ =7 指定数字输出轴 1
 - ◆ =8 指定数字输出轴 2
- D3 指定轴方向使能，对应轴参数 轴脉冲输出方向使能 的赋值。
 - ◆ 当 D3 设置为 0 时，不指定脉冲输出口 Y00~Y05 的方向位。
 - ◆ 当 D3 设置为 1 时，自动占用 Y10~Y15 作为脉冲输出口 Y00~Y05 的方向位。
 - ◆ 当 D2 指定为 6/7/8 时，D3 设置为 0 即可。



将 Y01 输出 (AXOUT1) 绑定到软件虚拟轴 3 (AXV3)，且 Y11 作 Y01 脉冲输出的方向口。

8.2.3.2 轴当量指定[AXPU]

对指定软件虚拟输出轴 (AXV) 的时间当量、脉冲当量和长度当量进行设置。



操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	时间当量	16 位 BIN
D3	脉冲当量	16 位 BIN
D4	长度当量	16 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定当前设置轴的时间当量，对应轴参数 **轴时间当量** 的赋值(推荐填写 10)。设置为负数时，取 16 位无符号数处理。比如设置 -1，则实际值为 65535。
- D3 指定当前设置轴的脉冲当量，对应轴参数 **轴脉冲当量** 的赋值(保留，推荐填 0)。
- D4 指定当前设置轴的长度当量，对应轴参数 **轴长度当量** 的赋值(保留，推荐填 0)。



设置软件虚拟轴 1 (AXV1)

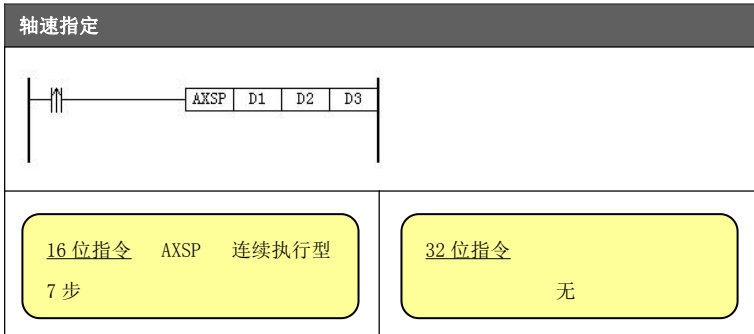
- 时间当量为 10 ms
- 脉冲当量为 0
- 长度当量为 0

编程注意事项

- D2 为当前轴的时间当量，在速度和加速度的计算中需要使用该概念进行耦合计算。需要注意的是，计算结果必须以整数形式存在。所以若时间当量设置的过小，就无法设置比较小的速度和加速度。
比如时间当量设置为 1，如果想要发送 200Hz 的脉冲，那么速度值应该计算为：
 $0.2 * 1 = 0.2$ ，无法写入到寄存器内。
- D3 和 D4 参数保留使用，可随意填写任意数值，推荐填 0。

8.2.3.3 轴速指定[AXSP]

对指定软件虚拟输出轴 (AXV) 的最大限速和起始速度进行设置。



操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	最大轴速	16 位 BIN
D3	起始轴速	16 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定当前设置轴的最大限速，对应轴参数 轴最高限速 的赋值。
- D3 指定当前设置轴的起动速度，对应轴参数 轴起始速度 的赋值。



设置软件虚拟轴 1 (AXV1) 的最大限速和起动速度。

若轴参数 轴时间当量 = 10ms，那么

最大限速 = 500 / 10 = 50 KHz

起动速度 = 10 / 10 = 1 KHz

编程注意事项

- D3 推荐填 0，以保证控制对象平稳启动。
- 轴速的设置应遵循以下限制，否则会导致轴参数 轴错误代码 报错：
 - ◆ 单路不得超过 600KHz
 - ◆ 两路同时运行不得超过 300KHz
 - ◆ 三路同时运行不得超过 200KHz。
 - ◆ 六路同时运行不得超过 100KHz。
- D2 和 D3 不能填写负数，否则会导致 轴错误代码 参数报错。

8.2.3.4 轴加减速指定 [AXAS]

对指定软件虚拟输出轴 (AXV) 的最大加减速度进行设置。

轴加减速指定

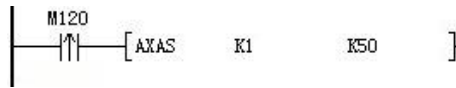
16 位指令 5 步	AXAS 连续执行型	32 位指令 无
----------------------	------------	--------------------

操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	轴最大加减速	16 位 BIN

操作数	字型软元件										
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●										
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定当前设置轴的最大加减速，对应轴参数 轴最大加减速 的赋值 (推荐填写 50)。



设置软件虚拟轴 1 (AXV1) 的最大加减速度。
 若轴参数 轴时间当量 = 10ms，那么
 最大加减速 = 50 / (10 * 10) = 0.5 脉冲/ms²

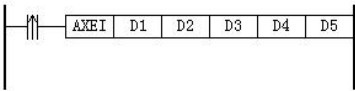
编程注意事项

- 若在执行运动指令的过程中产生机台抖动，可适当减小最大加减速以消除抖动。
- 若控制对象加速太慢，或为跟随轴情况下跟随困难 (凸轮运动)，此时可适当加大最大加减速以加快加速过程。

8.2.3.5 轴事件输入点指定[AXEI]

对指定软件虚拟输出轴 (AXV) 的事件输入点进行设置。

轴事件输入点指定



16 位指令 AXEI 连续执行型 11 步	32 位指令 无
-------------------------------------	--------------------

操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	近点信号输入点掩码	16 位 BIN
D3	编码器 Z 脉冲输入点掩码	16 位 BIN
D4	左限位信号输入点掩码	16 位 BIN
D5	右限位信号输入点掩码	16 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D5	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

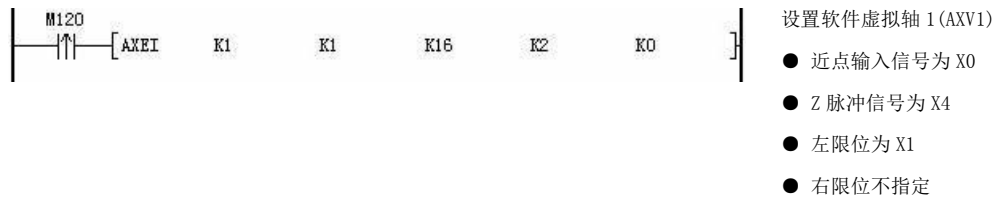
功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定当前设置轴的近点输入信号，对应轴参数 近点信号端口 的赋值。

D2 =	选定端口	D2 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14 (虚拟)/M8108
32	X5/M8101	8192	X15 (虚拟)/M8109
64	X6/M8102	16384	X16 (虚拟)/M8110
128	X7/M8103	-32768	X17 (虚拟)/M8111

通过轴事件输入点.Bit13 可使用 M8096~M8111 代替 X00~X17 的控制。

- 可通过 M8160~M8163 操控虚拟输入端子 X14~X17 的状态。
- D3指定当前设置轴的编码器 Z 脉冲输入信号,对应轴参数 Z 脉冲信号端口 的赋值。
 - ◆ 指定方法同 D2。
- D4指定当前设置轴的左限位开关输入信号,对应轴参数 左限位端口 的赋值。
 - ◆ 指定方法同 D2。
- D5指定当前设置轴的右限位开关输入信号,对应轴参数 右限位端口 的赋值。
 - ◆ 指定方法同 D2。
 - ◆ 由于运动指令并不调用该参数,故推荐填 0

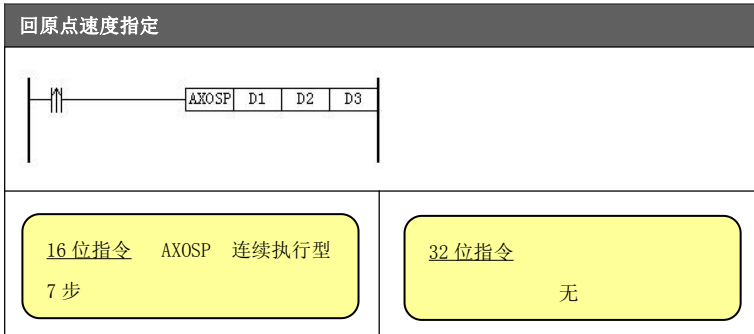


编程注意事项

- 不要指定同一个 X 端口到两个或多个不同的输入信号点上,以免运动指令的动作步序产生混乱。
- 若轴参数轴事件输入点设置不使其输入信号点,则该指令配置的输入端子在实际运动指令中不会调用,可以随意配置。
比如 轴事件输入点 中配置 轴事件输入点.Bit6 = 0,则 D4 数值无意义。
- 右限位指定 D5 数值所对应的输入端子在运动指令中不会调用,推荐填 0。

8.2.3.6 回原点速度指定[AXOSP]

对指定软件虚拟输出轴 (AXV) 的回原点高速和爬行速进行设置。



操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	原点回归速度	16 位 BIN
D3	原点爬行速度	16 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定当前设置轴的原点回归速度，对应轴参数 原点回归速度 的赋值。
- D3 指定当前设置轴的原点爬行速度，对应轴参数 原点爬行速度 的赋值。



设置软件虚拟轴 1 (AXV1) 的回原点高速和回原点爬行速度。

若轴参数 轴时间当量 = 10ms，那么

原点回归速度 = 500 / 10 = 50 KHz

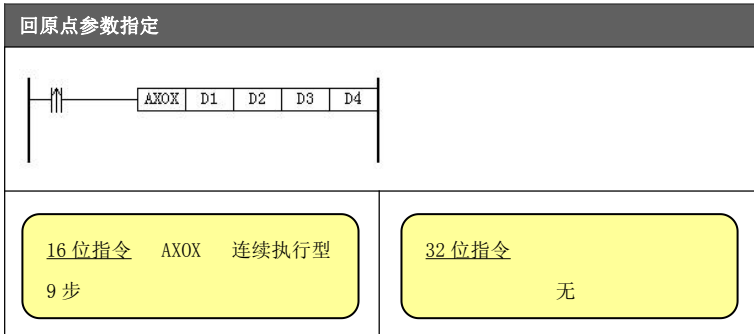
原点爬行速度 = 10 / 10 = 1 KHz

编程注意事项

- 轴速的设置应遵循以下限制，否则会导致轴参数 轴错误代码 报错：
 - ◆ 单路不得超过 600KHz
 - ◆ 两路同时运行不得超过 300KHz
 - ◆ 三路同时运行不得超过 200KHz。
 - ◆ 六路同时运行不得超过 100KHz
- D2 和 D3 不能填写负数，否则会导致 轴错误代码 参数报错。
- 该指令设置的参数在回原点指令 (HOME) 中被系统调用，当该轴对象不需要执行回原点的功能时，可随意配置回原点速度。但仍然需要配置爬行速 D3，此参数在点动指令的某些分支情况下会被耦合调用 (参考章节 8.3.2)。

8.2.3.7 回原点参数指定[AXOX]

对指定软件虚拟输出轴 (AXV) 的回原点方式进行设置。



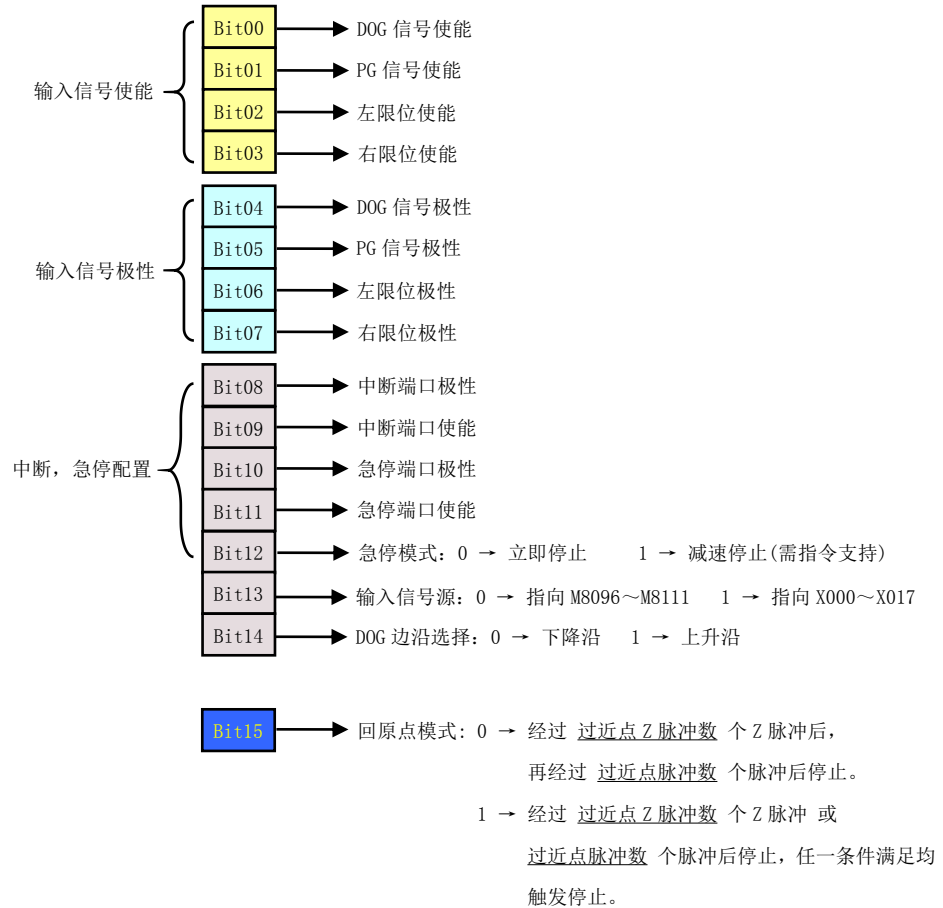
操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	回原点模式	16 位 BIN
D3	PG(Z 脉冲) 信号脉冲数	16 位 BIN
D4	DOG (近点) 信号脉冲数	16 位 BIN

操作数	字型软元件										
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●										
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定当前设置轴的回原点模式，该参数为按位设置方式。设置停止模式、输入信号使能和输入信号的极性选择，对应轴参数 轴事件输入点 的赋值。
- D3 指定当前设置轴的过近点信号后的 Z 脉冲计数，即当前轴经过近点信号后所需要经过的编码器 Z 脉冲计数个数，对应轴参数 过近点 Z 脉冲数 的赋值。
该数值在系统中以 16 位无符号数处理，超过 32767 则以负数形式设置/显示。
- D4 指定当前设置轴的过近点信号后的脉冲计数，即当前轴经过近点信号后所需要经过的脉冲计数个数，对应轴参数 过近点脉冲数 的赋值。
该数值在系统中以 16 位无符号数处理，超过 32767 则以负数形式设置/显示。

回原点模式
D2



设置软件虚拟轴 1 (AXV1)

- 近点/Z 脉冲/左限位/右限位均使能
- 近点/Z 脉冲/左限位/右限位开关极性均设置为 0。
- 回原点模式设置为 1, 即 1 个 Z 脉冲和 600 个脉冲计数值任一条件满足则执行停止。
- 离开近点信号 (DOG) 后, 假定:
 - ◆ 脉冲计数到 300 检测到 Z 相信号, 则直接停止。
 - ◆ 脉冲计数到 600, 还未检测到 Z 相信号, 也直接停止。

编程注意事
项

- 规定电机正转方向为右限位开关方向; 反转方向为左限位方向。
- 信号极性设置为 0: 进入信号 X 端口为 ON, 离开为 OFF;
设置为 1: 进入信号 X 端口为 OFF, 离开为 ON。
- 若不使能相应输入点信号, 则该信号极性选择可随意填写, 且端口指定也可随意填写。
若使能相应输入点信号, 则必须配置对应的端口指定, 否则运动指令运行将异常。
- 右限位信号在运动指令运行时不会被调用, 推荐填写该信号对应的配置数值为 0。
- Z 相信号只会指向对应的硬件 X 端子, 即使输入信号源指向 M 区。只能把 Z 相信号配置到端子 X0~X7 上。
- 该指令设置的参数在回原点指令 (HOME) 中被系统调用, 当该轴对象不需要执行回原点的功能时, 可随意配置该指令对应的参数。

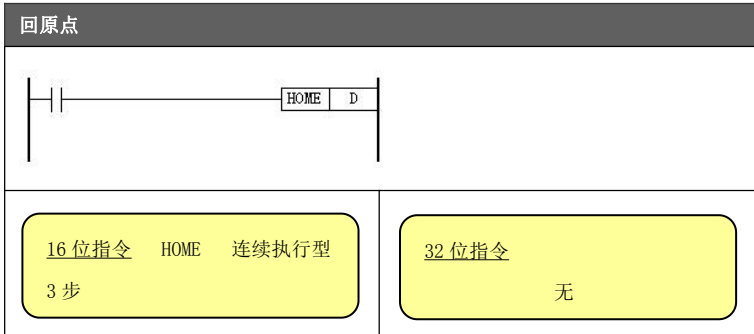
8.3 单轴基本运动

单轴基本运动指令涵盖了回原点、点动、定位、运动规划制表、表格输出和插补输出指令。

在确认对某个软件虚拟输出轴进行运动指令的执行时，请事先确认该轴的相关轴参数已经配置好。

8.3.1 回原点[HOME]

指定对应的软件虚拟输出轴执行回原点功能。



操作数	作用	类型
D	软件虚拟轴轴号	16 位 BIN

操作数	字型软元件										
D	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●										

功能与动作

由于回原点所需要的各项参数均已在轴参数内进行了设置，故回原点指令仅需指定对应的软件虚拟输出轴号即可。

回原点指令所涉及需要配置的轴参数有(配置方法及参数定义参考章节 10.3.2)：

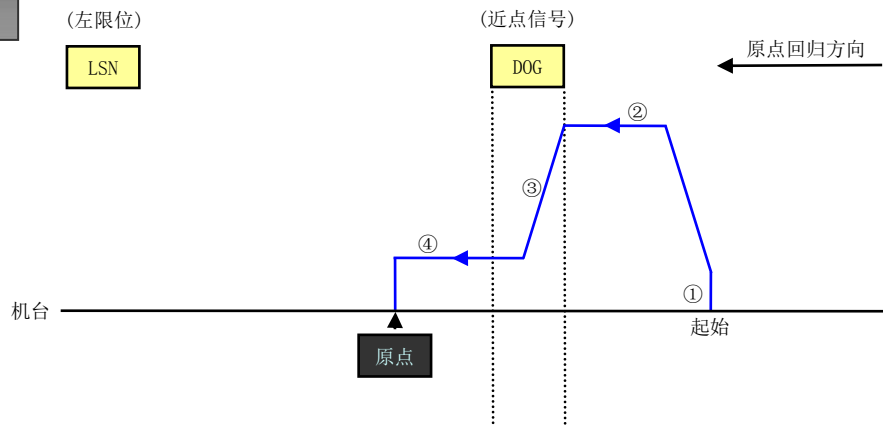
- 近点信号端口：指定近点信号的输入端口
- Z 脉冲信号端口：指定编码器 Z 相脉冲的输入端口
- 左限位端口：指定左限位信号的输入端口
- 右限位端口：指定右限位信号的输入端口(无实际调用)
- 原点回归速度：指定原点回归速度
- 原点爬行速度：指定原点爬行速度
- 回原点模式：指定回原点模式
- 过近点脉冲数：指定离开近点信号后的脉冲计数值。该参数为无符号数，若填写超过 32767 则以负数形式表示。例如填写 K-1000，实际数值为 K64536。
- 过近点 Z 脉冲数：指定离开近点信号后的 Z 相脉冲计数值。该参数为无符号数(同 过近点脉冲数 处理方式一致)。

回原点指令为上升沿触发，但必须使用常开触点导通调用回原点指令，途中断开触点将立即中断回原点流程。

需在执行回原点指令前，根据具体实际工况填写以上参数。若参数填写不合理，可能导致回原点动作错误。

近点使能情况 1

起始位置在近点信号右侧(此情况左限位端子可以不使能)。

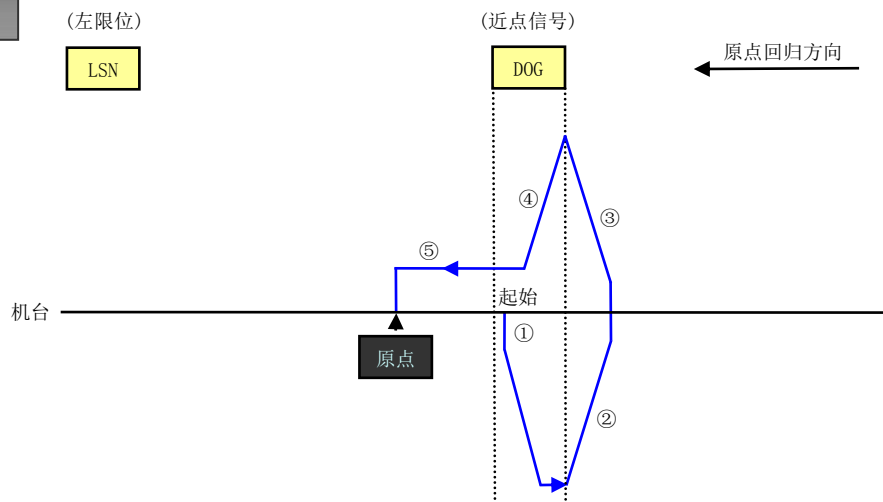


假定信号到位为 ON, 离开为 OFF (端子极性设置为 0)。

- ① 起始位置在 DOG 右端的时候, DOG = OFF。
- ② 以原点回归速度, 向原点回归方向开始移动。
- ③ 检测出 DOG = ON, 减速到原点爬行速度。
- ④ 检测出 DOG = OFF, 再根据轴参数轴事件输入点. Bit15 设置 (Z 相脉冲计数和脉冲计数), 进行精确定位停止。

近点使能情况 2

起始位置在近点信号内(此情况左限位端子可以不使能)。

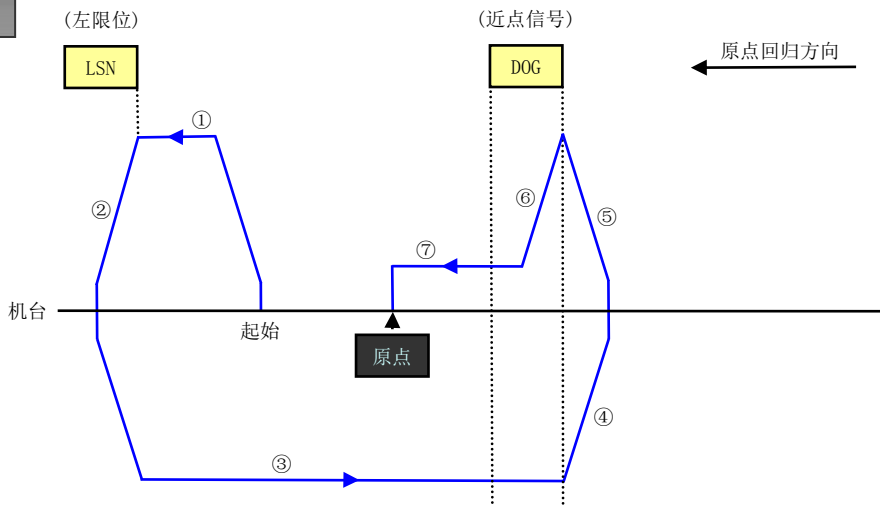


假定信号到位为 ON, 离开为 OFF (端子极性设置为 0)。

- ① 开始位置在 DOG 上, DOG = ON, 向与原点回归相反方向移动。
- ② 检测出 DOG = OFF 后减速停止。
- ③ 以原点回归速度, 向原点回归方向移动。
- ④ 检测出 DOG = ON, 减速到原点爬行速度。
- ⑤ 检测出 DOG = OFF 后, 再根据轴参数轴事件输入点. Bit15 设置 (Z 相脉冲计数和脉冲计数), 进行精确定位停止。

近点使能情况 3

起始位置在近点信号左侧且在左限位开关右侧(此情况左限位端子必须使能)。

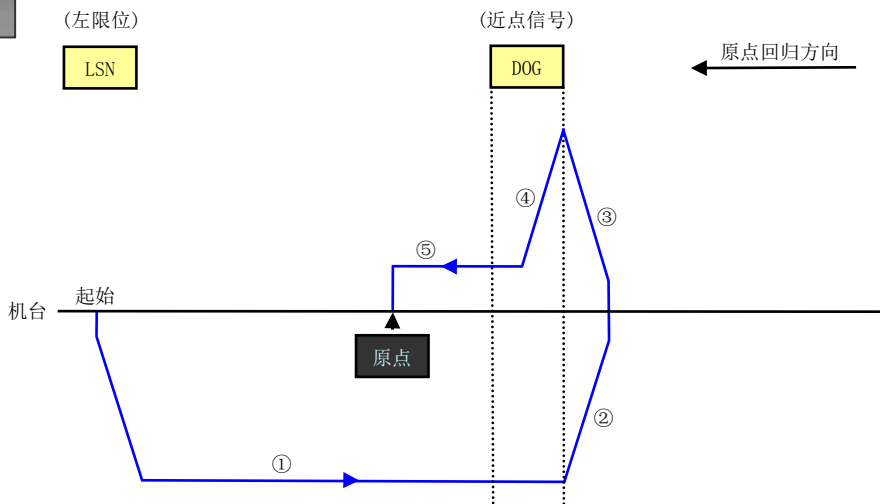


假定信号到位为 ON, 离开为 OFF (端子极性设置为 0)。

- ① 以原点回归速度向原点回归方向移动。
- ② 检测出 LSN = ON 后减速停止。
- ③ 以原点回归速度, 向原点回归相反方向移动。
- ④ 检测出 DOG = OFF 后减速停止。
- ⑤ 以原点回归速度, 向原点回归方向移动。
- ⑥ 检测出 DOG = ON, 减速到原点爬行速度。
- ⑦ 检测出 DOG = OFF, 再根据轴参数轴事件输入点. Bit15 设置 (Z 相脉冲计数和脉冲计数), 进行精确定位停止。

近点使能情况 4

起始位置在左限位开关处(此情况左限位端子必须使能)。



假定信号到位为 ON，离开为 OFF (端子极性设置为 0)。

- ① 以原点回归速度，向原点回归相反方向移动。
- ② 检测出 DOG = OFF 后减速停止。
- ③ 以原点回归速度，向原点回归方向移动。
- ④ 检测出 DOG = ON，减速到原点爬行速度。
- ⑤ 检测出 DOG = OFF，再根据轴参数轴事件输入点. Bit15 设置 (Z 相脉冲计数和脉冲计数)，进行精确定位停止。

近点不使能情况

- 当近点信号不使能的情况下，Z 脉冲信号必须使能。
- 系统将 Z 脉冲信号作为触发停止信号 (类似于近点信号的作用)，当检测到 Z 脉冲信号时，则执行停止流程操作过近点后的脉冲计数阶段，该流程下轴参数 过近点脉冲数 和 过近点 Z 脉冲数 的设置仍然有效。
- 系统将以原点爬行速驱动电机转动，不再加速到原点回归速度，防止无法检测到 Z 相信号。
- 若使能左限位的情况下：
 - ◆ 未检测到 Z 相脉冲，遇到左限位，则反向运动寻找 Z 脉冲，在此之后若再次遇到左限位将被忽略。
 - ◆ 检测到 Z 相脉冲后且系统处于停止流程计数阶段，此时遇到左限位将被忽略。
- 一般情况下，此种设置模式应用于类圆状机台的回原点。

编程注意事项

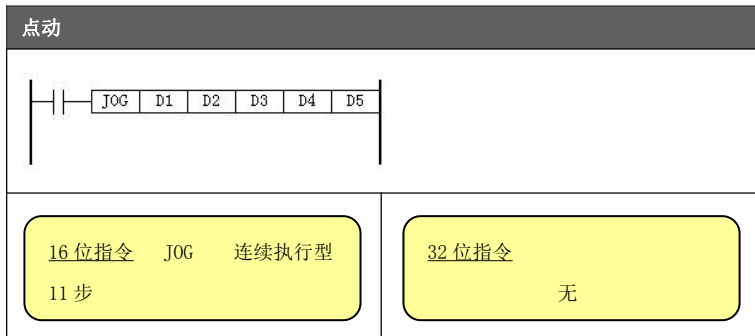
指令执行异常的情况有 (指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 近点信号和 Z 脉冲信号都不使能。
- Z 脉冲信号不使能，轴参数轴事件输入点. Bit15 设 0 且轴参数过近点 Z 脉冲数不为 0。
- 使能左限位但起始位置在左限位之外，或不使能左限位但起始位置在近点信号左侧。
- 输入端口配置不合理。比如将近点信号对应的端子配置到左限位对应端子的右方。
- 轴速度设置不合理导致轴参数 轴错误代码 报错。
- 爬行速设置太快，导致无法检测到开关信号或 Z 脉冲信号 (运动控制器的输入端子检测频率为 5KHz)。
- 使用触点边沿触发执行回原点指令。
- 信号使能但未配置信号输入端口 (比如使能近点信号但轴参数 近点信号端口 填 0)。
- 输入端子极性填反。由于工况不同可能导致进入输入信号端子状态为 OFF，此时应调整该输入信号的极性。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行 (脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.2 点动[JOG]

对指定软件虚拟输出轴 (AXV) 执行点动功能，其实际执行情况分为点动阶段和长动阶段。



操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	点动脉冲数	16 位 BIN
D3	长动速度	16 位 BIN
D4	正向点动/长动输入端口掩码	16 位 BIN
D5	反向点动/长动输入端口掩码	16 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D5	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定点动脉冲数，该参数为无符号数，若填写超过 32767 则以负数形式表示。
比如填写 -32000，实际发送脉冲数值为 $65536 - 32000 = 33536$ 。
 - ◆ D2 不等于 0，指令执行后进入点动阶段。当点动阶段完成后，视情况决定是否进入长动阶段。
 - ◆ D2 等于 0，无点动阶段，视情况决定是否进入长动阶段。
- D3 指定长动速度，与轴参数 轴时间当量 耦合计算。长动速度必须设置大于 0。
如 $D3 = 200$ ，且 轴时间当量 = 10，则点动速度 = $200 / 10 = 20$ KHz。
- 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。

- D4 设置正向点动/长动的输入信号(规定以电机正转为点动正方向)，其指定规则如下：

D4 =	选定端口	D4 =	选定端口
0	不选定端口		
1	X0/M8096	256	X10/M8104
2	X1/M8097	512	X11/M8105
4	X2/M8098	1024	X12/M8106
8	X3/M8099	2048	X13/M8107
16	X4/M8100	4096	X14(虚拟)/M8108
32	X5/M8101	8192	X15(虚拟)/M8109
64	X6/M8102	16384	X16(虚拟)/M8110
128	X7/M8103	-32768	X17(虚拟)/M8111

通过轴事件输入点.Bit13 可使用 M8096~M8111 代替 X00~X17 的控制。

通过 M8160~M8163 代替虚拟输入端子 X14~X17 的控制。

- D5 设置反向点动/长动的输入信号(规定以电机反转为点动反方向)，设置值参考 D4。点动信号决定运动方向及点动是否有效，若点动信号不指定输入端子，则对应方向的点动功能不动作。



设置软件虚拟轴 1 (AXV1)

- 设置点动脉冲数为 1000
 - 假定轴时间当量 = 10，则
长动速度 = 200 / 10 = 20KHz
 - 正向点动指定为 X003 输入端子
 - 反向点动指定为 X004 输入端子
- 当满足 M100 为 ON，且 X003 或 X004 触点闭合，软件虚拟轴 1 开始执行点动功能。

点动 / 长动阶段

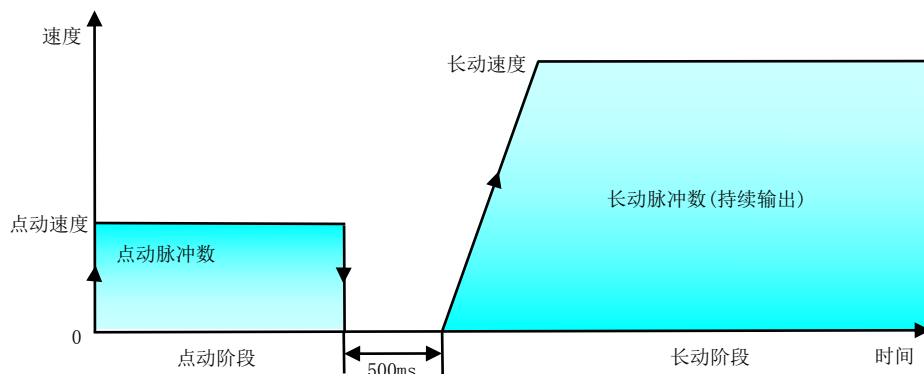
点动阶段	
进入条件	<ul style="list-style-type: none"> ● 点动脉冲数不为 0(必须条件)。 ● 指令启动时，方向信号处于有效状态(必须条件)。
点动速度	<ul style="list-style-type: none"> ● 当长动速度小于轴参数轴起始速度时，以起始速度运行点动脉冲数(无加减速)。 ● 当长动速度大于等于轴参数轴起始速度时，以长动速度运行点动脉冲数(无加减速)。
点动流程	<ul style="list-style-type: none"> ● 一旦进入点动阶段，无论点动方向信号是否失效，系统都将按照起始时的点动方向，以点动速度走完点动脉冲。 ● 点动脉冲走完后，系统延时 500ms。延时后若当前状态满足长动阶段进入的必须条件和可选条件 2，则进入长动阶段，否则点动流程结束。

长动阶段	
进入条件	<ul style="list-style-type: none"> ● 长动速度大于 0(必须条件)。 ● 方向信号处于有效状态(必须条件)。 ● 指令启动时, 点动脉冲数等于 0(可选条件 1)。 ● 点动阶段完成延时结束时, 指令仍然使能且方向信号仍然有效(可选条件 2)。
长动速度	<ul style="list-style-type: none"> ● 当长动速度小于轴参数轴起始速度时, 以长动速度运行(无加减速)。 ● 当长动速度大于等于轴参数轴起始速度时, 以长动速度运行(有加减速)。
长动流程	<ul style="list-style-type: none"> ● 以长动流程起始时的方向持续进行输出, 直到指令条件断开, 点动流程结束。 ● 长动过程中进行正反向信号的切换, 比如正向信号切换到反向信号, 则正向减速到零速后反向加速到长动速度持续输出。 ● 若正/反向信号同时有效, 则以反方向输出。 ● 若当前方向信号失效且运动轴不为零速时, 运动轴将以当前速度匀速运行。 ● 长动速度较大的情况下, 如要停止整个流程, 为安全起见, 应先失效当前的方向信号, 并激活相反的方向信号, 使之减速至接近零速后再执行断开点动指令。

下图为一个完整的点动指令执行的整个流程, 其假定条件有:

- 点动脉冲数不为 0
- 指令一直使能
- 长动速度大于 0
- 点动方向信号一直有效且在执行过程中未切换

系统首先以零速加速到点动速度, 走完点动脉冲数。延时 500ms 后再从零速加速到长动速度持续输出。



编程注意事项

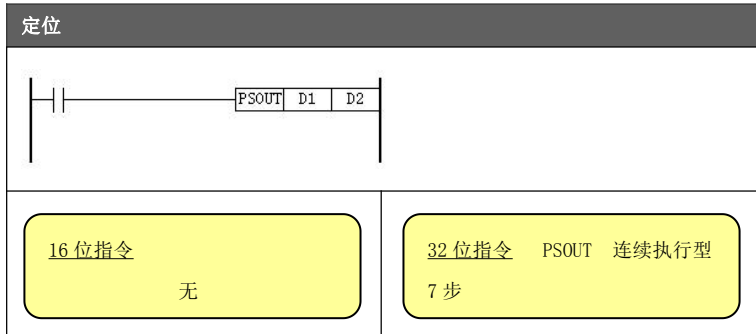
指令执行异常的情况有(指令不动作或动作错误):

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 正向信号和反向信号均未绑定具体 X 输入端口。
- 指令启动时正向信号和反向信号对应的 X 输入端口均为 OFF 状态。
- 长动速度设置 ≤ 0(在期望有长动阶段时)
- 轴速度设置不合理导致轴参数 轴错误代码 报错。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制, 若配置为运动指令占用且指令执行(脉冲口/方向/事件输出), 则对应 Y 软元件不再关联实际端口。

8.3.3 定位[PSOUT]

对指定软件虚拟输出轴 (AXV) 输出一定个数的脉冲数 (带方向)。



操作数	作用	类型
D1	软件虚拟输出轴号	32 位 BIN
D2	定位输出脉冲数	32 位 BIN

操作数	字型软元件										
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●										
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								

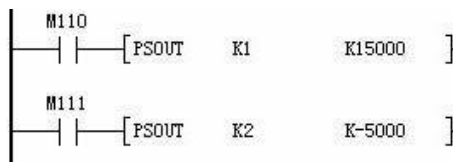
功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定定位输出脉冲数。
 - ◆ D2 = 0，输出 0 个脉冲。
 - ◆ D2 > 0，输出正向 D2 个脉冲。
 - ◆ D2 < 0，输出反向 D2 个脉冲 (必须先使能对应轴参数 轴脉冲输出方向使能)。

软件虚拟输出轴指向 Y0~Y5 脉冲口，则需将控制对象设置为脉冲+方向的模式，系统将通过 Y10~Y15 端口对脉冲输出方向进行控制。

当两个甚至更多的软件虚拟输出轴指定同一个硬件脉冲输出端口，可同时运行多个该指令达到该输出端口的位置综合给定/插补的运动效果。

- 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。



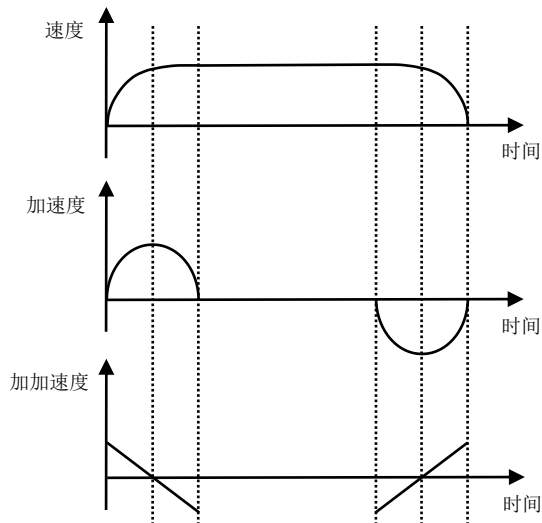
- M110 触点闭合，软件虚拟轴 1 (AXV1) 所绑定的硬件脉冲端口正向输出 15000 个脉冲。
- M111 触点闭合，软件虚拟轴 2 (AXV2) 所绑定的硬件脉冲端口反向输出 5000 个脉冲。

假定 AXV1 和 AXV2 同时绑定 Y4 输出口，且触点 M110 和 M111 相继有效使两条定长指令同一时间段运行，则将对 Y4 脉冲输出进行位置叠加。最后的效果为 Y4 输出 15000 - 5000 = 10000 个正向脉冲。

脉冲输出方法

由于任何物理机械系统都有一定的最大加速度和加加速度的限制，如果超限，不仅会影响加工质量，而且会加速它的损坏。故定位输出指令采用的是软件计算量较小，机械平滑性较高的四次多项式位移曲线。

其加减速曲线的构造需满足速度变化平稳、进给过程中冲击小，过渡过程时间短等基本条件，同时在起始和终止时需满足边界条件。由于位移、速度、加速度和加加速度满足导数关系，所以如果得到了位移曲线，则其它运动特性控制曲线可通过求导依次获得。



四次多项式的速度、加速度、加加速度的关系

编程注意事项

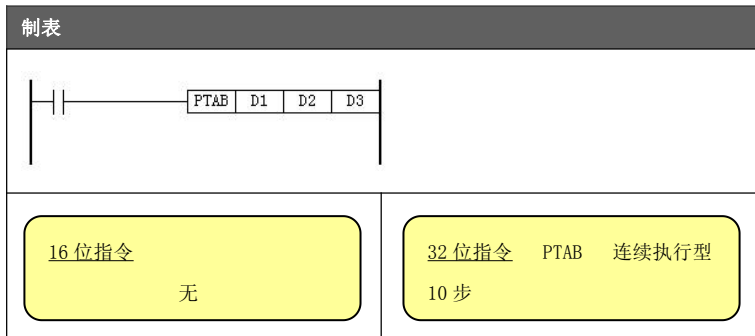
指令执行异常的情况有(指令不动作或动作错误):

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 走反向脉冲时方向信号未使能。
- 轴速度设置不合理导致轴参数 `轴错误代码` 报错。
- 对应轴参数 `轴状态` 不为 0 的情况下执行指令。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.4 制表[PTAB]

运动规划（位置均匀轨迹点表）制表，生成某个连续轨迹曲线中的一段分段曲线。



操作数	作用	类型
D1	段曲线参数配置缓存区起始单元	32 位 BIN
D2	轨迹表格起始单元	32 位 BIN
D3	段曲线表格间隔	32 位 BIN

操作数	字型软元件									
D1 (占 11 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2 (占 n 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- **D1** 指定本段曲线的参数配置缓存区起始寄存器单元，建议为偶数地址。
指令执行时，**D1** 指向寄存器后的连续 11 个寄存器均被占用，这些寄存器应在指令执行前配置好且只配置 1 次 (配置方法后述)，指令执行过程中这些寄存器将作为计算缓存区使用，制表指令导通条件断开这些寄存器将被释放。
也就是说，若为节省寄存器开销，在下一条制表指令执行前重新配置该寄存器区间，就可以重复再次指定。
例：**D1** 填写 D1000，则 D1000~D1010 在指令执行期间均被占用。
- **D2** 指定总曲线轨迹表格的起始寄存器单元，建议为偶数地址。
假定某个曲线轨迹规划 3 个分段曲线拼接而成，则需要用 3 次制表指令完成。这些制表指令所指向 **D2** 填写应一致。
例：**D2** 填写 D5000，总曲线表格将占用 D5000 以后的寄存器区域。
在执行首次制表指令前，需将 **D2** 指向的寄存器数值填 1；若为非接续型曲线，将 **D2+2** 指向的寄存器数值填 0 (均填写 32 位数值)。
- **D3** 指定本段曲线的表格间隔，以时间或主轴位置进行划分。
表格间隔要选取合适的范围，过大则降低曲线细分精度，过小则会造成曲线生成点数过多，使得寄存器开销变大。
表格间隔是求得本段曲线中有多少个点位数据的参数之一，点位数据又称之为型值点。每个型值点均代表从轴的一个位置点信息，通常以脉冲数形式表达。
* 在规划曲线轨迹时，所有分段曲线的表格间隔必须设置为相同数值。

一条连续的轨迹运动应由多个曲线连接而成，由这些曲线的不同时间/主轴位置间隔内的型值点排列成位置轨迹点表。应按照设想轨迹顺序，每调用一次制表指令则生成一个分段曲线数据，调用多次后则所有的分段曲线拼接而成一条连续的轨迹曲线。

位置均匀轨迹点表形成粗插补数据，可以是时间-位置单轴插补方式，也可以位置-位置相对运动（凸轮类运动）插补。因此凸轮类运动也可用位置均匀轨迹点表描述。

轨迹表定义

轨迹曲线表由 N 个制表指令执行生成 ($N \geq 1$)，由用户指定表头寄存器地址 (制表指令的 D2)。表内型值点数据系统自动计算填写，需留足表格数据存放的寄存器区间。

曲线轨迹表内容	内容解释	寄存器示例
表头 - (指定偶地址) 型值点总数(轨迹表量程)	初始手动填 1，之后每调用一次制表指令，由系统更新。	D5000(L)
初始位置点	非接续型曲线，必须在调用制表指令前手动填 0。	D5002(L)
分段曲线 1 型值点 1	第 1 次的制表指令计算生成。	D5004(L)
...		假定 100 个点
分段曲线 1 型值点 n		D5202(L)
分段曲线 2 型值点 1	第 2 次的制表指令计算生成。	D5204(L)
...		假定 200 个点
分段曲线 2 型值点 n		D5602(L)
分段曲线 N 型值点 1	第 N 次的制表指令计算生成。	...
...		
分段曲线 N 型值点 n		
表格总点数和表内型值点的数据类型均为 32 位，占用 2 个寄存器。		
表格占用总寄存器个数 = (表头)2 + 型值点总数 * 2		

分段曲线参数定义

分段曲线配置参数的定义如下表所示。地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

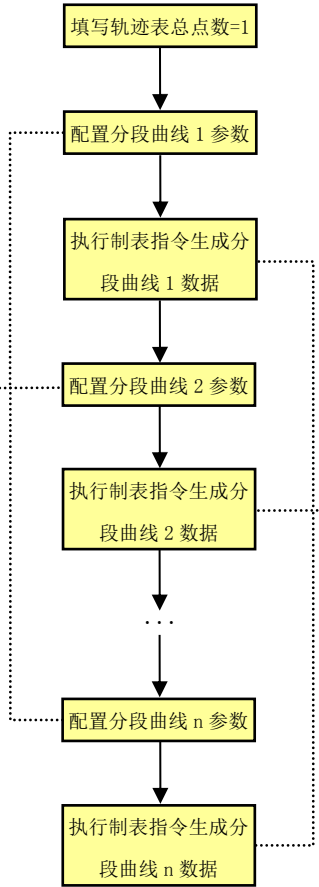
D1+偏移	名称	参数描述	单位
D1+0	曲线类别	指定分段曲线的生成规则。 <ul style="list-style-type: none"> ● = 0: 匀加速曲线 ● = 1: S 曲线(*保留, 若填此数值, 曲线将不计算) ● = 2: 四次多项式曲线 ● = 3: 均匀 B 样条曲线(*保留, 若填此数值, 曲线将不计算) 	-
D1+1	曲线方案	指定分段曲线的计算方案。 <ul style="list-style-type: none"> ● = 0: 相对位移确定, 终点速度未知 ● = 1: 终点速度确定, 相对位移未知 	-
D1+2(L)	相对位移	指定分段曲线相对位移(32 位数据)。 <ul style="list-style-type: none"> ● 当曲线方案设置为 0 的情况下, 必须配置该值。 ● 当曲线方案设置为 1 的情况下, 可随意配置该值, 计算完成后该地址数据由系统刷新。 	脉冲

D1+偏移	名称	参数描述	单位
D1+4	曲线量程	指定分段曲线量程，必须保证能整除制表指令中的 D3(曲线间隔)。 比如 D3 填写 3，曲线量程必须填写 3 的倍数，如 60, 90, 150 等数值。 曲线量程 ÷ D3 = 分段曲线的型值点点数。	-
D1+5	起始速度	分段曲线起始速度。 ● 若为首次制表指令的参数配置，推荐填 0。 ● 生成连续的曲线轨迹，本段分段曲线的起始速度的配置应填写上一段分段曲线的结束速度。	脉冲当量 / 时间当量
D1+6	结束速度	分段曲线结束速度。 ● 当曲线方案设置为 0 的情况下，可随意配置该值，计算完成后该地址数据由系统刷新。 ● 当曲线方案设置为 1 的情况下，必须配置该值。	脉冲当量 / 时间当量
D1+7	最大加减速速度	分段曲线最大加减速速度。生成曲线型值点后该地址数据由系统更新，用户可查看该数值是否在运动轴的极限加减速范围之内，若超过则必须做调整。 该参数无需设置，为显示值。	脉冲当量 / 时间当量 ²
D1+8	时间当量	分段曲线的时间当量，与脉冲当量耦合计算，为分段曲线速度/加减速度的计量单位之一。 规划包含多个分段曲线的连续轨迹曲线制表，推荐所有的分段曲线设置相同的时间当量。	ms
D1+9	脉冲当量	分段曲线的脉冲当量，与时间当量耦合计算，为分段曲线速度/加减速度的计量单位之一。 例如时间当量 = 1，脉冲当量 = 1，则分段曲线的速度单位为 1 脉冲/ms，加速度为 1 脉冲/ms ² 。 规划包含多个分段曲线的连续轨迹曲线制表，推荐所有的分段曲线设置相同的脉冲当量。	脉冲
D1+10	制表状态	制表指令当前状态，系统同一时间只允许 1 条制表指令处于运行状态。 ● = 0：当前无制表指令运行，可执行制表指令 ● = 16：正在生成曲线数据(系统每 1ms 生成 1 个型值点数据) ● = 32：曲线数据生成完毕 ● = 64：曲线生成错误 当检测到制表状态等于 32 时，曲线数据已生成完毕，可断开制表指令。梯形图扫描到指令断开后，自动将制表状态清 0。该数值变 0 后才能执行下一个制表流程。 该参数无需设置，为显示值。	-

轨迹表生成
流程

配置分段曲线的各项参数。

由于有些参数在执行制表过程中会被系统自动作为计算缓冲区使用,所以每条分段曲线的参数配置过程只需执行一次,不要反复配置(可使用边沿触发调用参数配置过程)。



系统根据分段曲线的配置参数,每 1ms 生成一个型值点,直至当前分段曲线数据生成完毕。计算完成后制表状态 = 32,并自动更新曲线轨迹表总点数。

若当前制表指令导通条件未断开,又启动了另一条制表指令,将被忽略。故应通过判断制表状态数值是否结束断开本条制表指令后,且制表状态归 0 再执行下一个制表流程。

曲线量程
与 D3

轨迹表保存从轴位置点的信息,主轴可以为时间轴,也可以为位置轴。设计轨迹表时根据实际情况确定该轨迹表的主轴性质。在表格输出时,主轴当前值必须指定相同性质的寄存器(参考表格输出 PTOUT 指令关于操作数 D3 的描述)。

- 主轴为时间轴: 制表指令的整个曲线量程(所有分段曲线的曲线量程之和)决定曲线的极限执行时间(通常以 ms 形式表达)。表格输出指令指定的主轴实际量程可在此数值基础上进行拉伸,以满足多个以极限曲线向右拉伸的时间簇对应的曲线输出。
- 主轴为位置轴: 制表指令的整个曲线量程(所有分段曲线的曲线量程之和)为主轴一个工艺流程所走的位置(通常以脉冲数形式表达),表格输出指令指定的主轴实际量程必须与曲线量程设置相同数值。

D3 为曲线的间隔,该参数的作用为求该曲线的型值点数,同时也决定了曲线的细分精度。由于表格输出指令的主轴实际量程将针对整个轨迹表进行拉伸,故所有分段曲线的 D3 间隔应该设置为一样。

- 曲线的型值点数 = 曲线量程 ÷ D3,两者必须满足整除关系,且确定了曲线量程的单位,D3 的单位也随之确认。例如曲线量程的单位为 ms,那么 D3 的单位也为 ms。
- 曲线的细分精度与曲线量程成正比,与 D3 间隔成反比。间隔为 1 则曲线细分精度最高,但相对生成点数最多,导致寄存器开销增大。

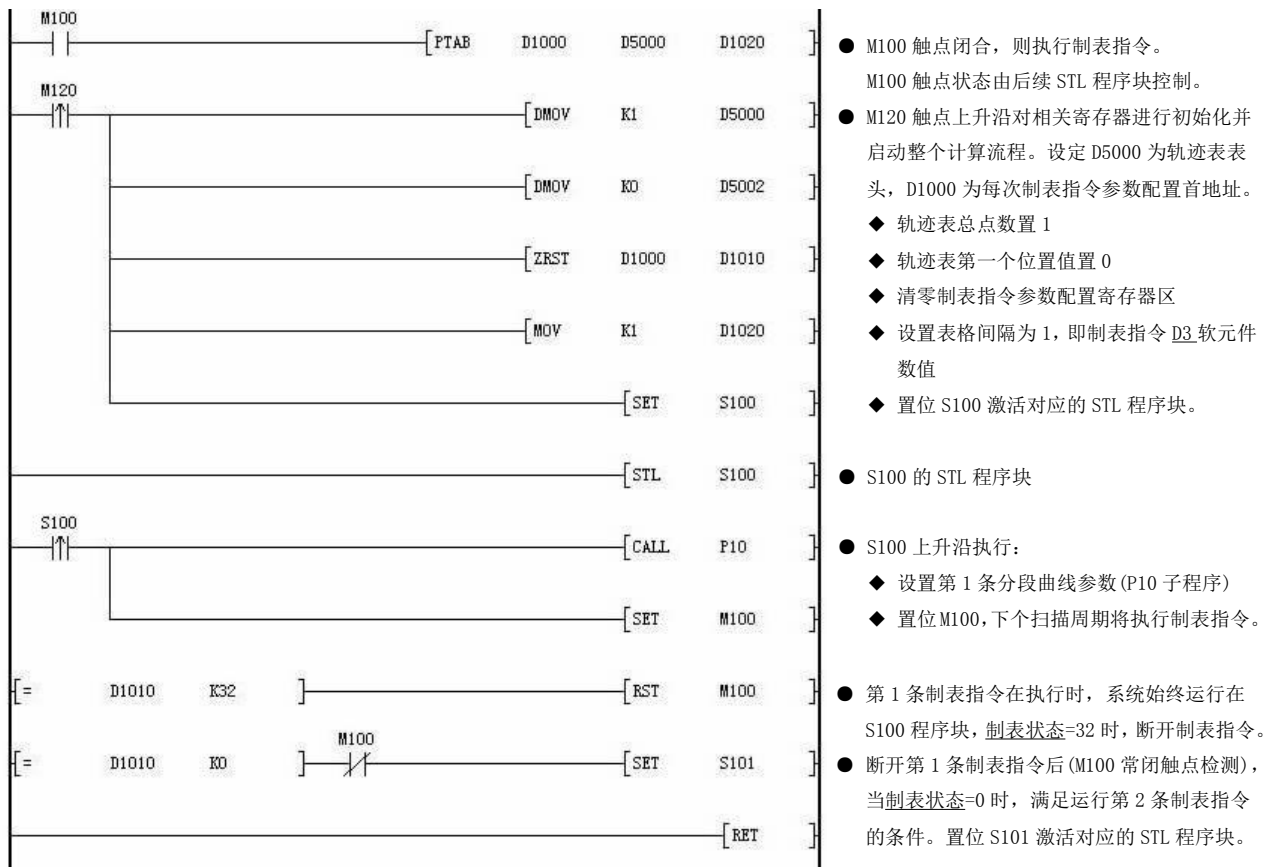
主轴关联举例：

主轴关联	主轴当前值举例	曲线性质	描述
时间轴	D8008(L)：滴答定时器 单位：ms	单轴多段速轨迹表	假定整个曲线量程 = 1000 ms，D3 = 5 ms。 曲线型值点点数 = 1000 ÷ 5 = 200 个点。 表格输出时，主轴实际量程也设置为 1000，则每 5 个 ms 的时间从轴输出到对应型值点的位置。1000ms 时，从轴输出到第 200 个型值点的位置。
位置轴	D200(L)：X0 端子接收脉冲 单位：脉冲	双轴位置关联轨迹表	假定整个曲线量程 = 4000 脉冲，D3 = 2 脉冲。 曲线型值点点数 = 4000 ÷ 2 = 2000 个点。 表格输出时，主轴实际量程也设置为 4000，则主轴每走 2 个脉冲的位移时从轴输出到对应型值点的位置。主轴走完 4000 个脉冲时，从轴输出到第 2000 个型值点的位置。

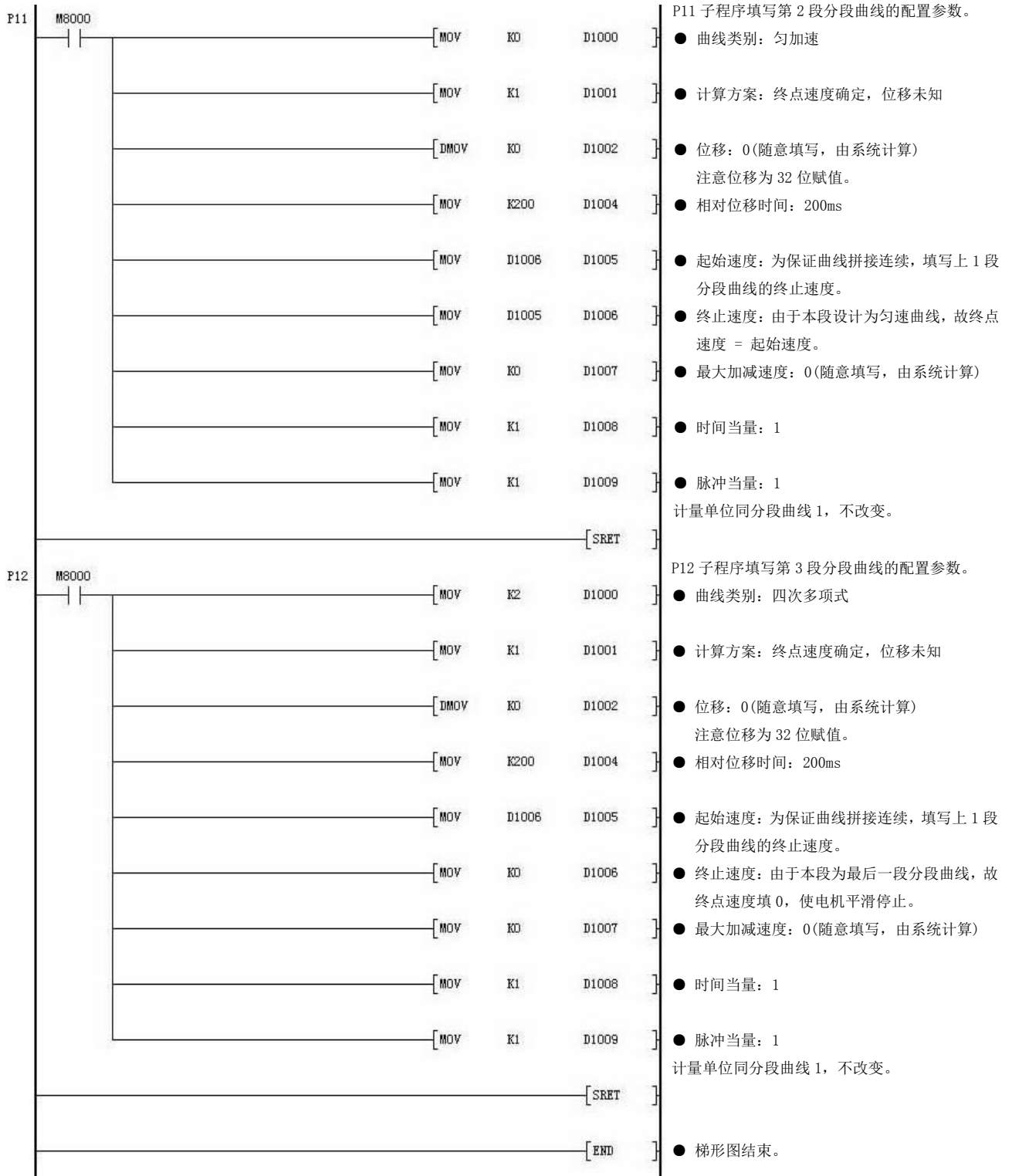
应用举例

设计一个连续的曲线轨迹由 3 段分段曲线完成，应用过程中按照顺序执行 3 次制表指令计算曲线型值点。

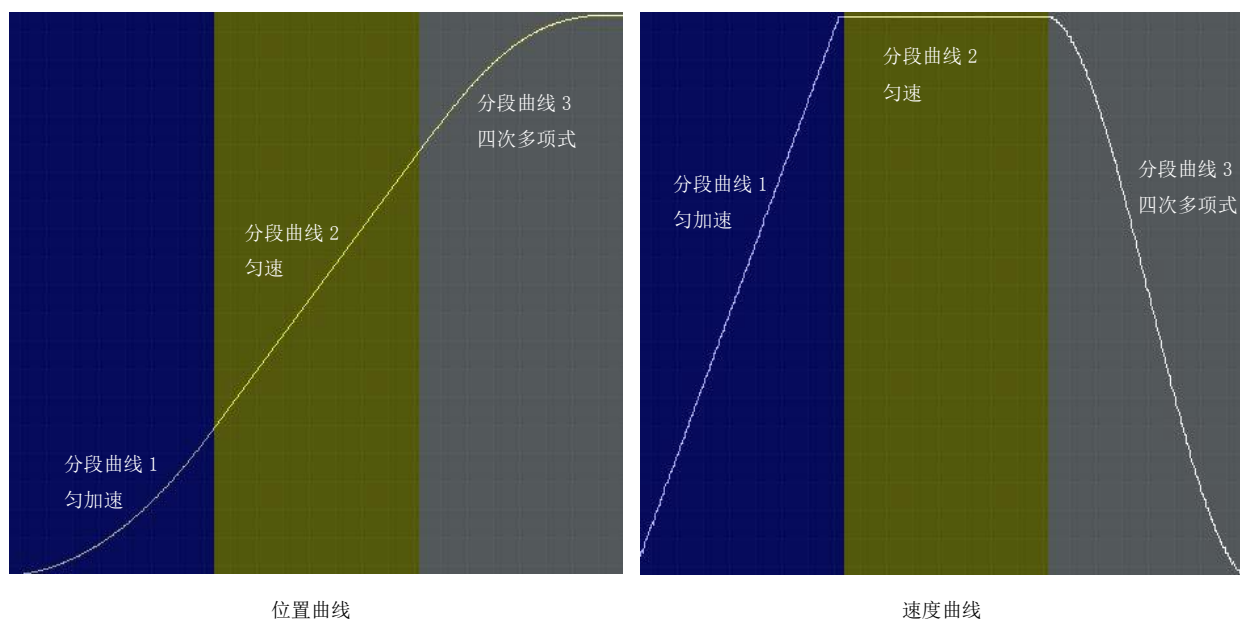
序号	类型
分段曲线 1	匀加速曲线
分段曲线 2	匀速曲线
分段曲线 3	四次多项式曲线





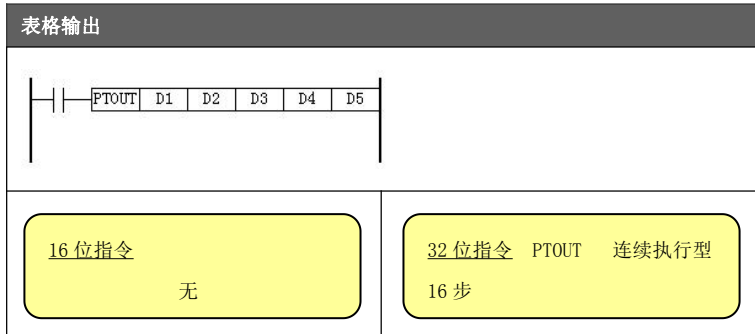


以上示例使用表格输出 (PTOUT) 指令输出的曲线轨迹如下:



8.3.5 表格输出[PTOUT]

将已生成好的轨迹表进行输出(轨迹表可通过制表指令等多种方式生成)。



操作数	作用	类型
D1	软件虚拟输出轴号	32 位 BIN
D2	轨迹表格起始单元	32 位 BIN
D3	主轴(横轴)实际数值	32 位 BIN
D4	主轴(横轴)实际量程	32 位 BIN
D5	输出方式	32 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D5	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

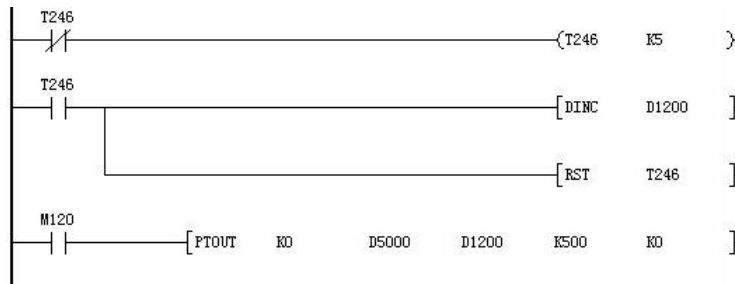
功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定总曲线轨迹表格的起始寄存器单元，建议为偶数地址。
- D3 指定主轴当前实际值，建议为偶数地址。
 - ◆ 该值若对应时间，则表格输出以主轴时间轴进行输出(单轴多段速轨迹表)。
 - ◆ 该值若对应位置，则表格输出以主轴位置轴进行输出(双轴关联凸轮表)。
- D4 指定主轴实际量程。轨迹表通常以实际工艺最大速度/加速度制表，表格输出时若当前主轴速度低于最大值，可将轨迹表总点数增大进行拉伸处理(即可以规划多个以轨迹曲线向右拉伸的曲线簇)，此时加速度/跃度均不会越限。
- D5 指定输出模式。
 - ◆ = 0，到终点结束，轨迹表输出完成后停止输出。
 - ◆ = 1，循环输出，轨迹表输出完成后再次从第 1 个位置点(更新初始点)开始输出。
- 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。

指令以启动时刻或作为循环模式输出的最后 1 个点的位置的 D3 数值作为轨迹表相对零位值, 对其增量变化以及曲线拉伸系数(实际量程 ÷ 表格量程)来寻找轨迹表中的点位, 并输出到该点位的位置值(中间运算结果可能是小数, 经过内部浮点处理输出整数脉冲)。

主轴为时间轴

该功能一般用于单轴多段速度定位输出。一个位置表代表一个多段速输出。当设计这类多段速曲线时, 以设备极限工艺参数(最大工艺速度/加速度)制表。表格的横轴量程将最小。当设备低于最高工艺速度运行时, 采用此表格输出(拉伸轨迹表的横轴量程)可证明加速度/跃度均不会越限。



- M120 触点闭合, 进行表格输出。
- ◆ 指定软件虚拟轴 0 输出。
- ◆ 表头位置为 D5000。
- ◆ 主轴为时间轴, 其实际值 5ms 累加, T246 控制。
- ◆ 输出 500 次位置后本次指令执行完毕, 即执行时间为 $5 \times 500 = 2500$ ms。
- ◆ 本次执行完毕后不再输出。

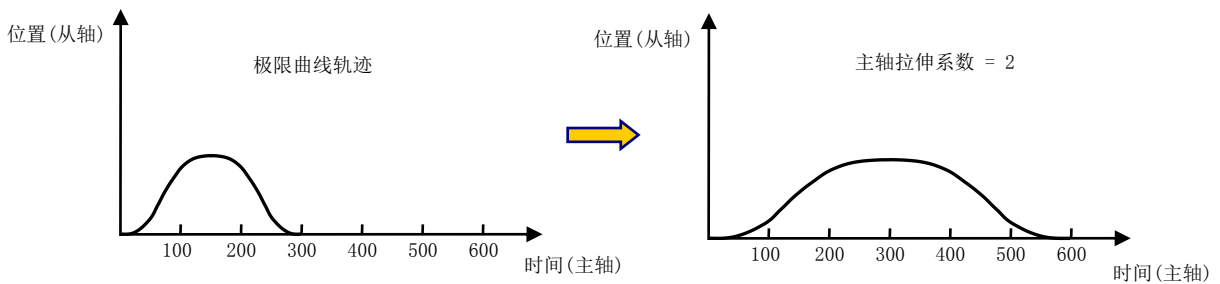
假定表格内总点数为 400 点, 则拉伸系数 = $500 \div 400 = 1.25$ 倍。



对于时间增量为 1ms 的表格输出, 其主轴实际值可填写 D8008。这是一个系统 1ms 的 32 位滴答定时器反馈值。

这样可减少定时器资源的开销。

拉伸系数=2 时, 轨迹表理论曲线与实际输出曲线的对比:



主轴为位置轴

该功能一般用于凸轮曲线从轴输出。一个位置表代表一个凸轮曲线，主轴位置量程与横轴坐标点数形成一定的比例关系。主轴与从轴是位置对位置的关系，与时间无关。轨迹表以主从轴极限工艺参数(最大工艺速度/加速度)制表，如主轴低于最高工艺速度运行，则主轴当前实际值变化变缓，那么从轴对应的位置输出也会变缓。



- M120 触点闭合，进行表格输出。
- ◆ 表头位置为 D2500。
- ◆ 主轴为 D220 数值（32 位）。
- ◆ 假定轨迹表的位置间隔为 10 个脉冲(即制表指令中的操作数 D3 = 10)，轨迹表型值点点数 = 2000。则主轴量程 = 20000 个脉冲。当主轴位置走到 20000 个脉冲时，单次凸轮曲线输出完毕。
- ◆ 本次执行完毕后循环输出(循环工艺)。

衍生应用：以凸轮角度为视角进行轨迹表格输出。

凸轮一圈为 360°，即主轴每 0.5° 角对应从轴 1 个位置输出，输出完所有点位后主轴凸轮正好转动一圈。假定一个工艺流程主轴所走的脉冲数为 N，以 0.5° 为单位进行划分，则从轴(凸轮)每走 0.5° 角，对应主轴所走的脉冲数的增量为 $N \div 720$ 。

规划轨迹表内总点数为 720 点，表格间隔应为 $N \div 720$ ，按照此规则使用制表指令(PTAB)设计轨迹表。

从轴轨迹表型值点序号与主轴实际反馈脉冲数的对应关系如下：

第 1 个凸轮周期				第 2 个凸轮周期				...
主轴反馈 脉冲位置	$(N \div 720) \times 1$	从轴输出型 值点点位	1	主轴反馈 脉冲位置	$(N \div 720) \times 721$	从轴输出型 值点点位	1	...
	$(N \div 720) \times 2$		2		$(N \div 720) \times 722$		2	
	
	$(N \div 720) \times 720$		720		$(N \div 720) \times 1440$		720	

减小角度划分(如以 0.1° 角为划分单位)可显著提高精度，但会带来额外的表格寄存器开销。

现场工艺除了走凸轮曲线外，往往还需要一些附加工艺参数做辅助，比如啮合位移量、事件比较输出等，用单纯的表格输出指令难以满足现场需求。推荐用户使用追剪正程/滚切制表(PCTAB)指令生成凸轮曲线轨迹表，以及使用电子凸轮(CAM)指令对凸轮表格进行输出。这些指令将在后续章节进行介绍。

编程注意事项

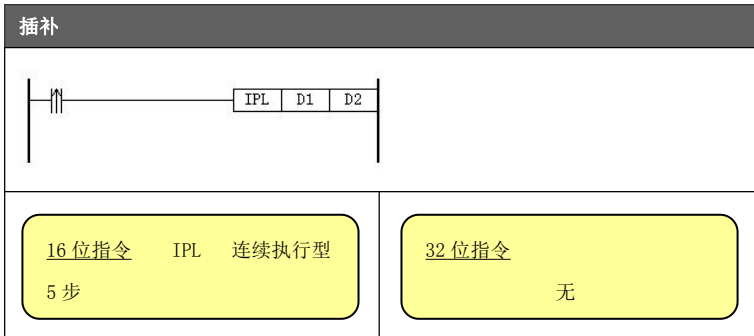
指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 轨迹表起始单元序号或主轴实际值单元序号指定为非偶数寄存器地址。
- 调用制表指令的时序不恰当导致制表指令并未执行，致使轨迹表格型值点生成遗漏的情况。可通过表头单元地址，也即是表格总点数来判断规划的型值点点数是否全部生成。
- 轨迹表格内型值点变化不合理导致轴参数 轴错误代码 报错。
- 轴速度设置不合理导致轴参数 轴错误代码 报错。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.6 插补[IPL]

对指定软件虚拟输出轴 (AXV) 一次性输出一定个数的脉冲数 (带方向), 通常用于消除少量的位置偏差, 比如色标位置偏差。也可用于持续性插补输出。

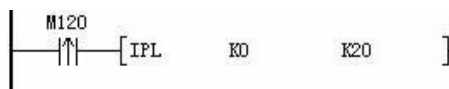


操作数	作用	类型
D1	软件虚拟输出轴号	16 位 BIN
D2	插补脉冲数	16 位 BIN

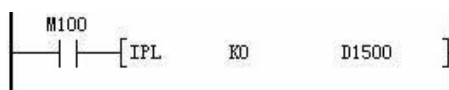
操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5), 对应数值 0~5。
- D2 指定插补输出脉冲数。该数值在下一个插补周期内 (1ms) 一次性给定进行输出, 故不应设置太大, 否则容易引起电机抖动。
推荐插补输出数值在 -100~100 之间 (绝对值越小, 插补越平稳)。
当 D2 填写为负值时, 为反方向脉冲插补输出。



- M120 触点上沿, 向软件虚拟输出轴 0 插补 20 个正向脉冲 (单次插补)。



- M100 触点闭合, 梯形图每次扫描到该条指令, 则向软件虚拟输出轴 0 插补 D1500 寄存器所指向数值的脉冲个数 (持续插补)。

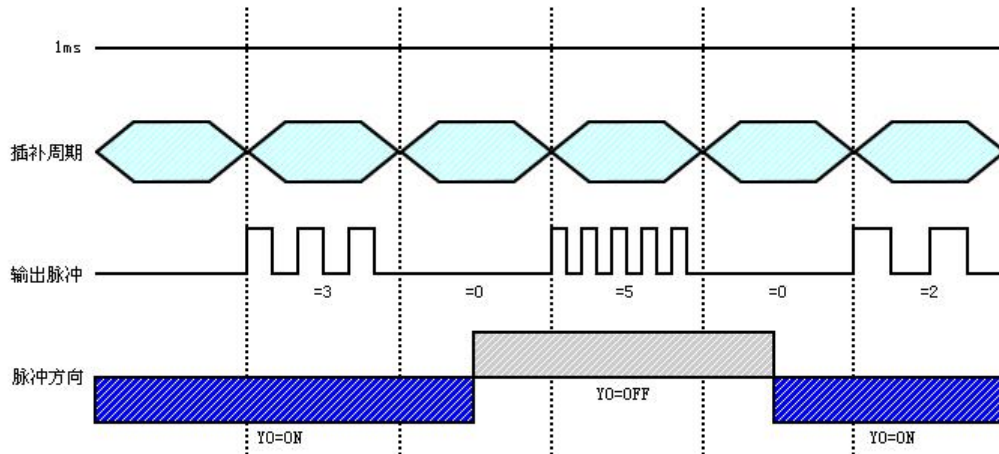
持续插补的程序放置在 1ms 中断内执行, 比如高级中断 I900 内, 以保证插补脉冲输出的连续性。

插补换向时序要求

对于持续性脉冲插补, 若中途对插补脉冲进行换向, 应遵循以下原则:

- 软件虚拟输出轴绑定为 Y0~Y5 脉冲口时, 必须保证在正确的方向上发脉冲, 以满足时序上的要求。
 - ◆ 使能脉冲方向信号 Y10~Y15。
 - ◆ 每次换向前的一个插补周期, 插补脉冲数给 0, 即停止一个插补周期的脉冲输出。若检测到脉冲方向发生变化且上一个插补周期脉冲插补数 > 0, 则本插补周期无论是否填写有插补数值, 都不会输出脉冲。

假定向 Y4 脉冲口进行持续性脉冲插补输出，输出个数为...，-3, 5, -2，... 则其正确的配置时序图如下：



编程注意事项

指令执行异常的情况有(指令不动作或动作错误)：

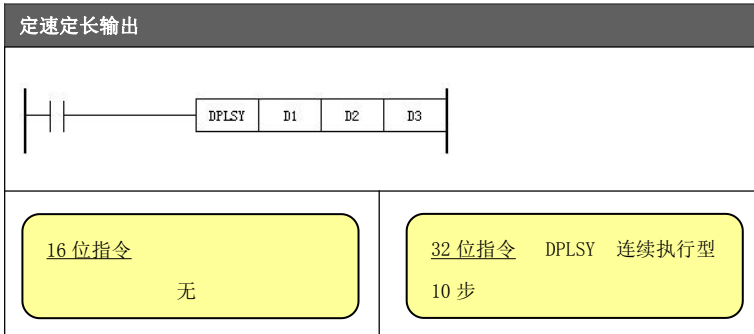
- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 插补数值过大导致电机抖动，推荐插补数值不要超过 100，一般来说插补数值越小则系统变化越平稳。
- 软件虚拟输出轴对应输出轴为 Y0~Y5 时，
 - ◆ 未指定对应方向口 Y10~Y15。
 - ◆ 未遵循上述换向时序要求导致当前插补周期的插补脉冲输出失效。
- 如果想要发送精确的插补脉冲数，IPL 指令最好在 1MS 定时中断里使用。如果放在主流程里，可能导致以下情况的发生：
 - ◆ 当梯形图扫描时间小于 1MS 时，有可能在检测到指令断开时，系统并未发送插补脉冲。
 - ◆ 当梯形图扫描时间大于 1MS 时，比如 5MS，有可能在检测到指令断开时，已发送了 5 次的插补脉冲。
- IPL 指令的插补脉冲输出频率不受轴参数 轴最大限速 的限制，但仍受系统硬件最大限速的限制。

比如 6 个软件虚拟输出轴分别绑定到 Y0~Y5，则硬件最大限速为 $600\text{KHz}/6=100\text{KHz}$ 。IPL 针对该虚拟轴的插补数与同一时刻其他运动指令(如果有的话)叠加到该虚拟轴的脉冲数之和不能高于 100。

PLC 上电运行时 Y0~Y5，Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.7 定速定长输出[DPLSY]

对指定软件虚拟输出轴 (AXV) 按照指定频率输出一定个数或无限个数的脉冲数 (带方向)。



操作数	作用	类型
D1	脉冲频率	32 位 BIN
D2	脉冲数	32 位 BIN
D3	软件虚拟输出轴号	32 位 BIN

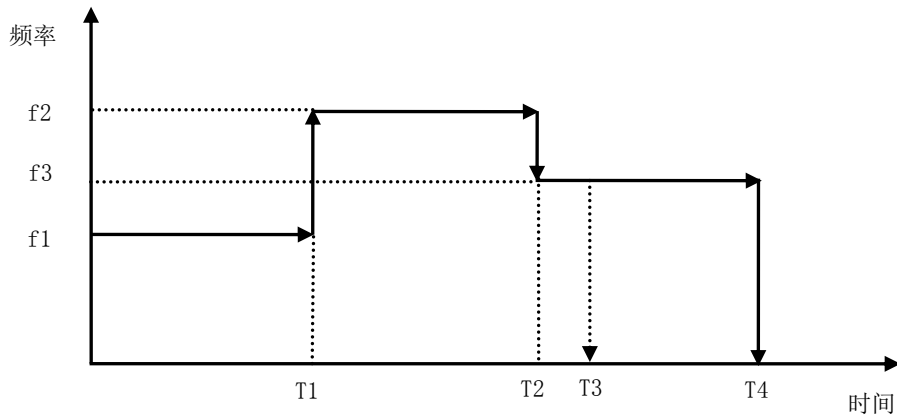
操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D1	●	●	●							
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D2	●	●	●							
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D3	●									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z

功能与动作

- **D1** 指定输出的脉冲频率。实际输出的脉冲频率通过轴参数 轴时间当量 计算得出。计算公式 = $D1 \div (\text{轴时间当量} \div 1000)$ 。
 例如 轴时间当量 = 10ms, D1=1000, 则实际输出的脉冲频率为 100KHz。
 例如 轴时间当量 = 1000ms, D1=1, 则实际输出的脉冲频率为 1Hz。
 起始速度从 0 速开始 (并不以轴参数 轴起始速度 启动)。
 当设置频率大于轴参数 轴最高限速 时, 以 轴最高限速 进行脉冲输出。
- **D2** 指定输出脉冲数。
 - ◆ D2 的取值范围是 -16000000~16000000, 超出该范围可能造成超速报警。取 0 时不发送脉冲。
 当 D2 填写为负值时, 则为反方向脉冲输出。若没有指定方向端口, 则负值不起作用 (按正值输出)。
- **D3** 指定软件虚拟输出轴号 (AXV0~AXV5), 对应数值 0~5。
- 可在线修改输出的脉冲频率。
- 该指令仅支持急停事件, 且急停事件产生时立即停止。
- 当指令停止时, 其停止速度为 0 速。

指令动作时
序

DPLSY 指令动作时序如下图所示：



- 以初始的 f1 频率发送脉冲，f1 为指令指定的频率
- T1 处在线修改了输出频率 f2，以该频率输出脉冲
- T1 处在线修改了输出频率 f3，以该频率输出脉冲
- T3 处若设置了急停事件使能，且相应的事件产生，则立即停止
- T4 处若设置了有限个脉冲输出，且此时输出完毕，则立即停止

编程注意事
项

指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 设置的脉冲频率超过了软件虚拟输出轴的 轴最高限速 配置。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.8 加减速定长输出 [DPLSR]

对指定软件虚拟输出轴 (AXV) 按照指定频率 (带加减速) 输出一定个数的脉冲数 (带方向)。可指定从启动速度到最高速度的加速时间。

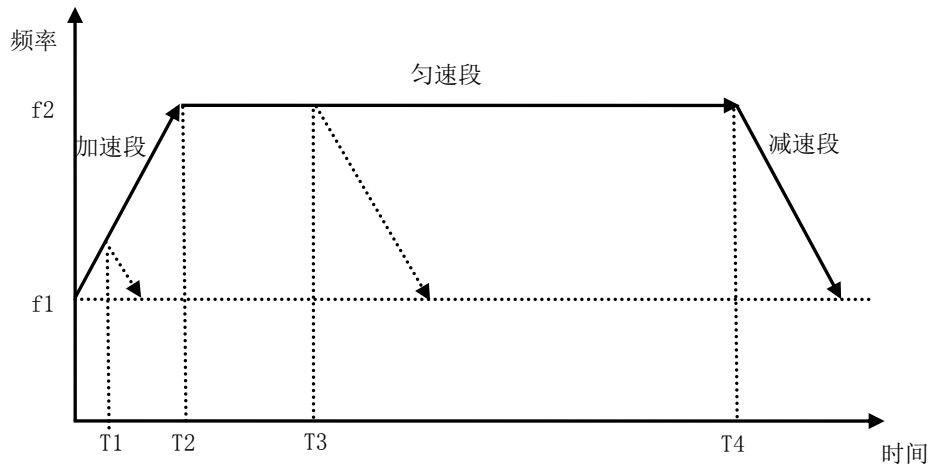
加减速定长输出											
<p>16 位指令</p> <p>无</p>	<p>32 位指令 DPLSR 连续执行型</p> <p>13 步</p>										
操作数	作用	类型									
D1	脉冲频率	32 位 BIN									
D2	脉冲数	32 位 BIN									
D3	加减速时间	32 位 BIN									
D4	软件虚拟输出轴号	32 位 BIN									
操作数	字型软元件										
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●	●	●								
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
	●										

功能与动作

- **D1** 指定输出的脉冲频率。实际输出的脉冲频率通过轴参数 **轴时间当量** 计算得出。计算公式 = $D1 \div (\text{轴时间当量} \div 1000)$ 。脉冲频率必须填写 >0 的数值。
D1 的设置如果小于轴参数 **轴起始速度**。最终速度为轴参数 **轴起始速度**。
 且当 **D1** 的设置大于 **轴最高限速** 时，实际的目标速度不会超过 **轴最高限速**。
- **D2** 指定输出脉冲数。
 ◆ **D2** 的取值范围是 -16000000~16000000，超出该范围可能造成超速报警。
 当 **D2** 填写为负值时，且配置了方向端口，则为反方向脉冲输出。没有指定方向端口，则负值不起作用 (按正值输出)。
- **D3** 指定加减速时间，单位毫秒，对应数值 50~5000。
 由 **D3** 计算得出的加速度小于等于 0 或大于 **轴最大加减速** 时，采用 **轴最大加减速** 所设定的数值。
- **D4** 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- 该指令仅支持急停事件，且急停事件产生时根据轴参数 **轴事件输入点**. Bit12 决定是立即停止还是减速停止。
- 当指令停止时，其停止速度为轴起始速度。

指令动作时
序

DPLSR 指令的动作时序如下图所示：



- 以初始的 f_1 频率发送脉冲进入加速段， f_1 为轴参数 轴起始速度 所指定，加速度由指令指定的加减速时间计算得出。
 - ◆ 当计算得出的加速度小于轴参数 轴最大加减速，那么加速段以轴参数指定的加速度进行加速。
 - ◆ 若使能急停，且 T_1 处产生了该事件，则根据急停模式减速/立即停止。
- T_2 处进入匀速段， f_2 为指令指定的脉冲频率
 - ◆ 当指定的脉冲频率小于轴参数 轴最高限速 或计算得出的匀速段输出频率，则匀速段输出频率为 3 者间最小的数值。
 - ◆ 若使能急停，且 T_3 处产生了该事件，则根据急停模式减速/立即停止。
- T_4 处进入减速段，是加速段的逆过程。
 - ◆ 若使能急停，且在减速段处产生了该事件，急停模式为减速停止则仍然以原减速段规划方式减速停止。否则立即停止。

编程注意事
项

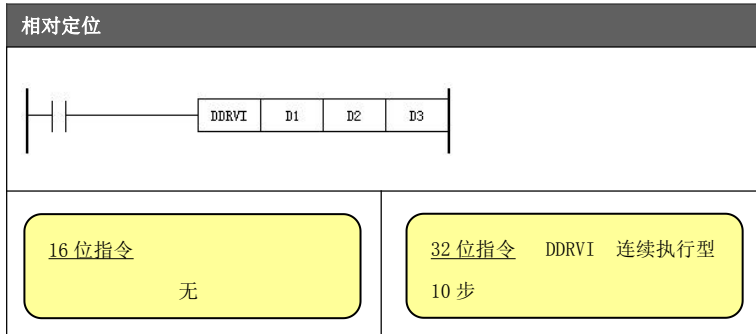
指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 加减速时间，脉冲数，脉冲频率必须满足一定的规则，才能达到上述的标准时序。否则可能造成实际的加减速时间或目标脉冲频率达不到想要设置的数值。但实际输出的脉冲数与设置的脉冲数是相等的。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 $Y_0 \sim Y_5$ ， $Y_{10} \sim Y_{15}$ 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.9 相对定位[DDRVI]

对指定软件虚拟输出轴 (AXV) 按照指定频率 (带加减速) 输出一定个数的脉冲数 (带方向)。



操作数	作用	类型
D1	脉冲数	32 位 BIN
D2	脉冲频率	32 位 BIN
D3	软件虚拟输出轴号	32 位 BIN

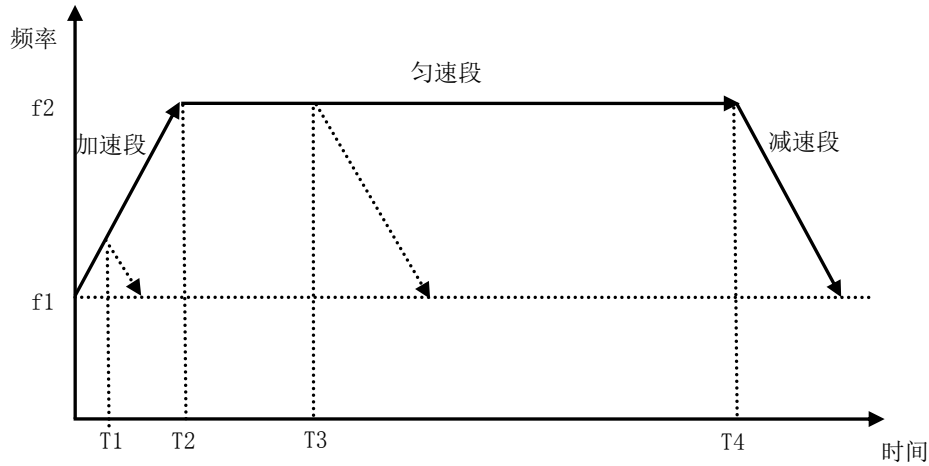
操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D1	●	●	●							
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D2	●	●	●							
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D3	●									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z

功能与动作

- D1 指定输出脉冲数。
 - ◆ D1 的取值范围是-16000000~16000000，超出该范围可能造成超速报警。
 - 当 D1 填写为负值时，且配置了方向端口，则为反方向脉冲输出。没有指定方向端口，则负值不起作用 (按正值输出)。
- D2 指定输出的脉冲频率。实际输出的脉冲频率通过轴参数 轴时间当量 计算得出。计算公式 = $D2 \div (\text{轴时间当量} \div 1000)$ 。
D2 的设置如果小于轴参数 轴起始速度。最终速度为轴参数 轴起始速度。
且当 D2 的设置大于 轴最高限速 时，实际的目标速度不会超过 轴最高限速。
- D3 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- 起始速度由轴参数 轴起始速度 指定，加速度采用 轴最大加减速 所设定的数值。
- 支持急停事件，急停事件产生时根据轴参数 轴事件输入点. Bit12 决定是立即停止还是减速停止。
- 支持左右限位事件，左右限位事件产生时运动轴减速停止。
- 当指令停止时，其停止速度为轴起始速度。

指令动作时
序

DDRVI 指令的动作时序如下图所示：



- 以初始的 f_1 频率发送脉冲进入加速段， f_1 为轴参数 轴起始速度 所指定，加速度为轴参数 轴最大加减速。
 - ◆ 若使能急停，且 T_1 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T_1 处产生了该事件，则减速停止。
- T_2 处进入匀速段， f_2 为指令指定的脉冲频率
 - ◆ 当指定的脉冲频率小于轴参数 轴最高限速 或计算得出的匀速段输出频率，则匀速段输出频率为 3 者间最小的数值。
 - ◆ 若使能急停，且 T_3 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T_3 处产生了该事件，则减速停止。
- T_4 处进入减速段，是加速段的逆过程。
 - ◆ 若使能急停，且在减速段处产生了该事件，急停模式为减速停止则仍然以原减速段规划方式减速停止。否则立即停止。
 - ◆ 若使能左右限位，且在减速段处产生了该事件，仍然以原减速段规划方式减速停止。

编程注意事
项

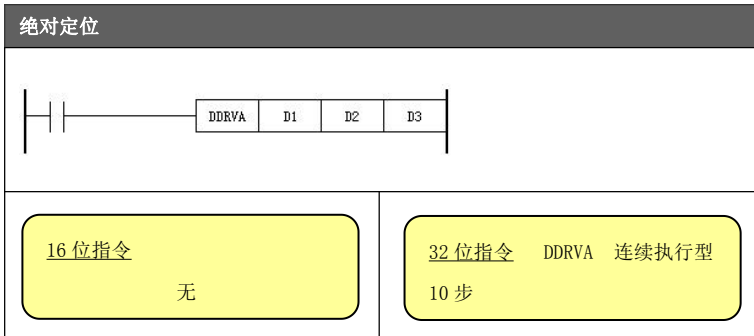
指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 $Y_0 \sim Y_5$, $Y_{10} \sim Y_{15}$ 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.10 绝对定位[DDRVA]

对指定软件虚拟输出轴 (AXV) 按照指定频率 (带加减速) 输出一定个数的脉冲数 (带方向), 运动到指定脉冲数的位置。



操作数	作用	类型
D1	脉冲数	32 位 BIN
D2	脉冲频率	32 位 BIN
D3	软件虚拟输出轴号	32 位 BIN

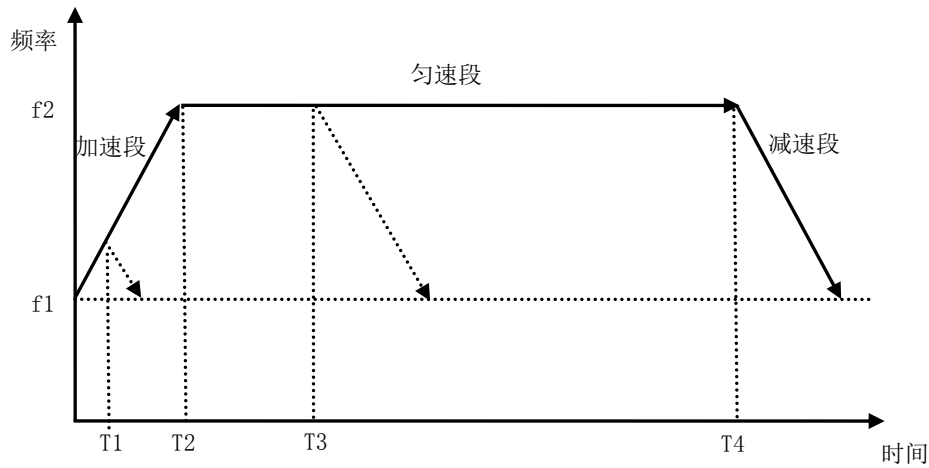
操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

- D1 指定绝对位置脉冲数。
 - ◆ D1 与当前位置差值的取值范围是-16000000~16000000, 超范围可能造成超速报警。当 D1 填写为负值时, 且配置了方向端口, 则为反方向脉冲输出。没有指定方向端口, 则负值不起作用 (按正值输出)。
- D2 指定输出的脉冲频率。实际输出的脉冲频率通过轴参数 轴时间当量 计算得出。计算公式 = $D2 \div (\text{轴时间当量} \div 1000)$ 。
D2 的设置如果小于轴参数 轴起始速度。最终速度为轴参数 轴起始速度。
 且当 D2 的设置大于 轴最高限速 时, 实际的目标速度不会超过 轴最高限速。
- D3 指定软件虚拟输出轴号 (AXV0~AXV5), 对应数值 0~5。
- 实际输出的脉冲数为 D2 减去轴参数 轴相对位置。
- 起始速度由轴参数 轴起始速度 指定, 加速度采用 轴最大加减速 所设定的数值。
- 支持急停事件, 急停事件产生时根据轴参数 轴事件输入点. Bit12 决定是立即停止还是减速停止。
- 支持左右限位事件, 左右限位时间产生时运动轴减速停止。
- 当指令停止时, 其停止速度为轴起始速度。

指令动作时
序

DDRVA 指令的动作时序如下图所示：



- 以初始的 f_1 频率发送脉冲进入加速段， f_1 为轴参数 轴起始速度 所指定，加速度为轴参数 轴最大加减速。
 - ◆ 若使能急停，且 T_1 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T_1 处产生了该事件，则减速停止。
- T_2 处进入匀速段， f_2 为指令指定的脉冲频率
 - ◆ 当指定的脉冲频率小于轴参数 轴最高限速 或计算得出的匀速段输出频率，则匀速段输出频率为 3 者间最小的数值。
 - ◆ 若使能急停，且 T_3 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T_3 处产生了该事件，则减速停止。
- T_4 处进入减速段，是加速段的逆过程。
 - ◆ 若使能急停，且在减速段处产生了该事件，急停模式为减速停止则仍然以原减速段规划方式减速停止。否则立即停止。
 - ◆ 若使能左右限位，且在减速段处产生了该事件，仍然以原减速段规划方式减速停止。

编程注意事
项

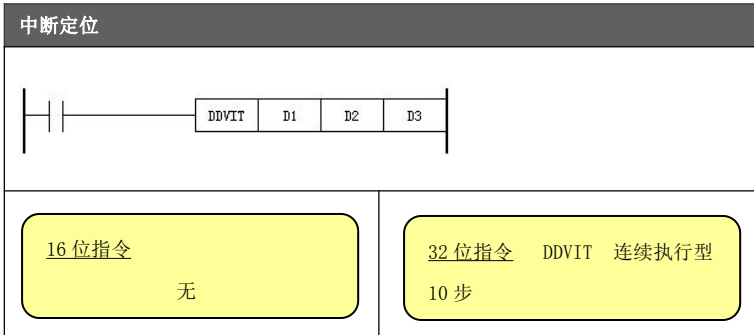
指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
 - 对应轴参数 轴状态 不为 0 的情况下执行指令。
 - 注意操作数所填写的脉冲数即使在 $-16000000 \sim 16000000$ 之间，也有可能
- 导致脉冲数设置超范围的可能。比如当前位置为 -8000000 ，而 D1 填写 8000001 ，那么两者差值为 16000001 。

PLC 上电运行时 $Y_0 \sim Y_5$ ， $Y_{10} \sim Y_{15}$ 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.11 中断定位[DDVIT]

对指定软件虚拟输出轴 (AXV) 按照指定频率 (带加减速) 输出无限个数的脉冲数 (带方向)。当遇到中断事件后, 再输出一定数量的脉冲后减速停止。



操作数	作用	类型
D1	脉冲数	32 位 BIN
D2	脉冲频率	32 位 BIN
D3	软件虚拟输出轴号	32 位 BIN

操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

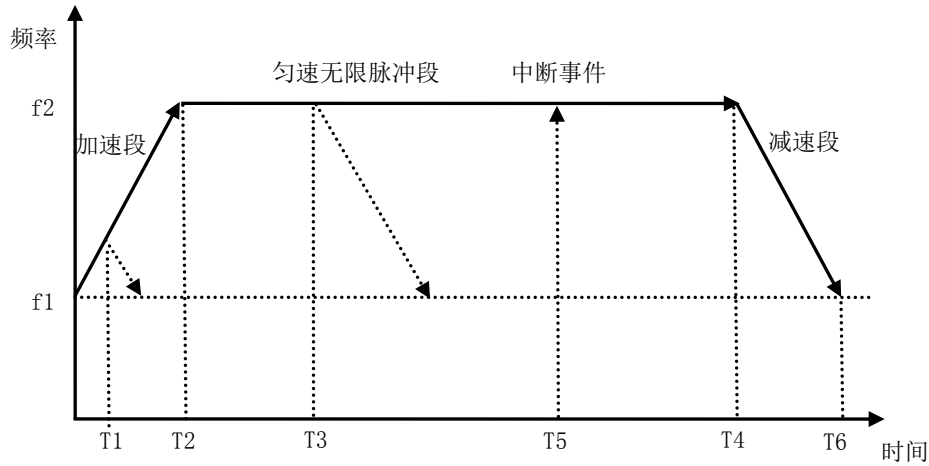
功能与动作

- **D1** 指定遇到中断事件后的输出脉冲数。在中断事件产生之前, 无限发送脉冲。
 - ◆ **D1** 的取值范围是-16000000~16000000, 超出该范围可能造成超速报警。
 当 **D1** 填写为负值时, 且配置了方向端口, 则为反方向脉冲输出。没有指定方向端口, 则负值不起作用 (按正值输出)。
- **D2** 指定输出的脉冲频率。实际输出的脉冲频率通过轴参数 **轴时间当量** 计算得出。计算公式 = $D2 \div (\text{轴时间当量} \div 1000)$ 。该操作数必须填写正数。

指令启动以轴参数 **轴起始速度** 启动, 直到加速到 **D2** 指定输出的脉冲频率进行输出。且当 **D2** 的设置大于 **轴最高限速** 时, 实际的目标速度不会超过 **轴最高限速**。
- **D3** 指定软件虚拟输出轴号 (AXV0~AXV5), 对应数值 0~5。
- 起始速度由轴参数 **轴起始速度** 指定, 加速度采用 **轴最大加减速** 所设定的数值。
- 加速到指定频率后进行无限的脉冲输出, 当检测到中断事件后 (中断事件端口由轴参数 **轴中断事件** 指定), 再输出 **D1** 指定的脉冲数后减速停止。
- 支持急停事件, 急停事件产生时根据轴参数 **轴事件输入点**. Bit12 决定是立即停止还是减速停止。
- 支持左右限位事件, 左右限位时间产生时运动轴减速停止。
- 当指令停止时, 其停止速度为轴起始速度。

指令动作时
序

DDVIT 指令的动作时序如下图所示：



- 以初始的 f_1 频率发送脉冲进入加速段， f_1 为轴参数 轴起始速度 所指定，加速度为轴参数 轴最大加减速。
 - ◆ 若使能急停，且 T_1 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T_1 处产生了该事件，则减速停止。
- T_2 处进入匀速无限脉冲段， f_2 为指令指定的脉冲频率
 - ◆ 当指定的脉冲频率小于轴参数 轴最高限速 或计算得出的匀速段输出频率，则匀速段输出频率为三者间最小的数值。
 - ◆ 若使能急停，且 T_3 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T_3 处产生了该事件，则减速停止。
 - ◆ 当该段产生了中断事件，则进入发送 D1 指定遇到中断事件后的输出脉冲数阶段。
- T_4 处进入减速段，是加速段的逆过程。
 - ◆ 若使能急停，且在减速段处产生了该事件，急停模式为减速停止则仍然以原减速段规划方式减速停止。否则立即停止。
 - ◆ 若使能左右限位，且在减速段处产生了该事件，仍然以原减速段规划方式减速停止。
 - ◆ 指令所指定的 D1 脉冲数是系统从 $T_5 \sim T_6$ 时间段内发送的脉冲数。

编程注意事
项

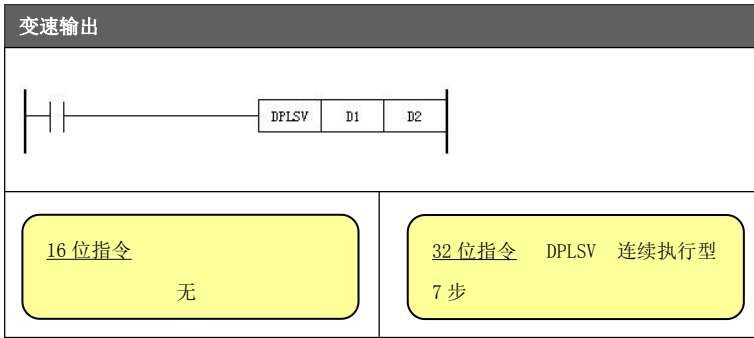
指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。
- 如果设置的 D1 脉冲数不够或轴参数 轴最大加减速 比较低，由于系统要保证最后能够正常的减速到起始速度，可能导致系统匀速段频率无法达到到预设的最大频率。
- 如果 D2 的设置如果与轴参数 轴起始速度 相隔太近，可能造成指令运行出错。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.12 变速输出[DPLSV]

对指定软件虚拟输出轴 (AXV) 按照指定频率 (带加减速) 输出无限个数的脉冲数 (带方向)。在输出过程中可在线修改输出频率。



操作数	作用	类型
D1	脉冲频率	32 位 BIN
D2	软件虚拟输出轴号	32 位 BIN

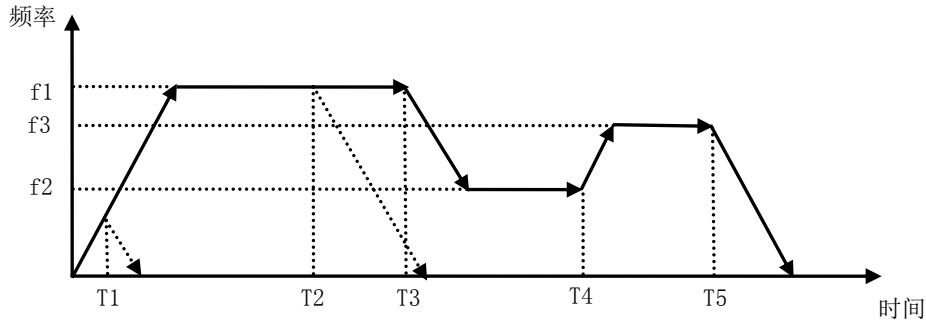
操作数	字型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●									

功能与动作

- **D1** 指定输出的脉冲频率。实际输出的脉冲频率通过轴参数 **轴时间当量** 计算得出。计算公式 = $D1 \div (\text{轴时间当量} \div 1000)$ 。
起始速度从 0 速开始 (并不以轴参数 **轴起始速度** 启动)。
当设置频率大于轴参数 **轴最高限速** 时, 最终以 **轴最高限速** 进行脉冲输出。
- **D2** 指定软件虚拟输出轴号 (AXV0~AXV5), 对应数值 0~5。
- 指令启动后运行到 **D1** 指定输出的脉冲频率后进行无限的脉冲输出, 当检测到 **D1** 发生变化时, 系统通过轴参数 **轴最大加减速** 到达新的频率后, 以当前频率对外输出无限脉冲。
- 支持急停事件, 急停事件产生时根据轴参数 **轴事件输入点. Bit12** 决定是立即停止还是减速停止。
- 支持左右限位事件, 左右限位时间产生时运动轴减速停止。
- 当指令停止时, 其停止速度为 0 速。

指令动作时
序

DPLSV 指令的动作时序示例如下图所示：



- 启动指令，以轴参数 轴最大加减速 加速到 f1 进入匀速段。f1 为指令操作数 D1 指定。
 - ◆ 若使能急停，且 T1 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T1 处产生了该事件，则减速停止。
- f1 匀速段时的事件处理
 - ◆ 若使能急停，且 T2 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T2 处产生了该事件，则减速停止。
 - ◆ T3 处用户在线修改操作数 D1 指定 f2，以轴参数 轴最大加减速 减速到 f2 再次进入匀速段
- f2 匀速段时的事件处理
 - ◆ T4 处用户在线修改操作数 D1 指定 f3，以轴参数 轴最大加减速 加速到 f3 再次进入匀速段
- f3 匀速段时的事件处理
 - ◆ 若使能急停，且 T5 处产生了该事件，则根据急停模式减速/立即停止。
 - ◆ 若使能左右限位，且 T5 处产生了该事件，则减速停止。

编程注意事
项

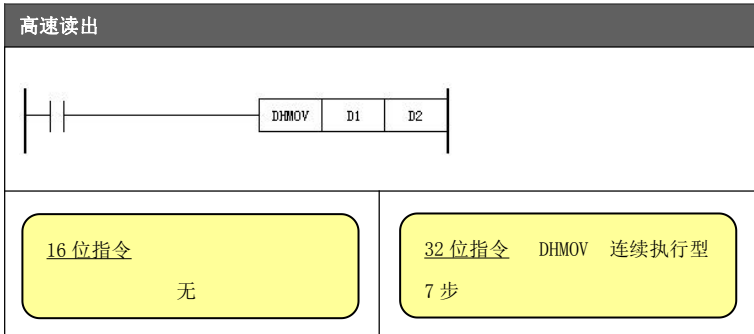
指令执行异常的情况有(指令不动作或动作错误)：

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 对应轴参数 轴状态 不为 0 的情况下执行指令。

PLC 上电运行时 Y0~Y5, Y10~Y15 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.3.13 高速读出 [DHMOV]

对高速计数器或高速输出脉冲的即时读取或即时改写。

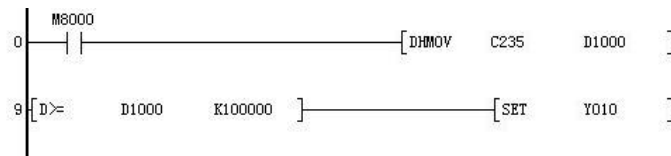


操作数	作用	类型
D1	读出数据地址	32 位 BIN
D2	高速计数器当前值存入的目标寄存器地址	32 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D1		●	●	●						
D2		●	●	●						

功能与动作

- D1 指定需要高速读出的软元件，一般为高速计数器 C235~C255 或脉冲输出的 D 寄存器。
- D2 指定需要存入的目标寄存器地址。
比如 DHMOV C235 D1000，执行结果将当前 C235 的数值存入 D1000 (32 位) 内。
- 常规情况下高速输入输出数值是在本次扫描周期结束时刷新，因此相对程序处理的时刻，信号有些过时（读取）或延时（改写）。本指令为满足更实时要求，在程序处理时对高速计数器或高速输出脉冲的即时读取或即时改写。
- DHMOV 是梯形图用户与底层高速处理的一个便利接口，是用户与底层数据交换的最快的通道。因此读写高速处理的值，通过 DHMOV 指令可以做到即时更新。



- DHMOV 指令执行，将 C235 高速计数器立即刷新，然后将计数器值送入到 D1000 (32 位)。
- 比较 D1000 和一个预设值的大小（这里为 100000），如果 ≥ 该值则置位 Y10。
- 该控制流程与 DHSCS 高速比较置位指令类似，但 DHSCS 可以不受扫描周期的影响，处理结果更为实时。

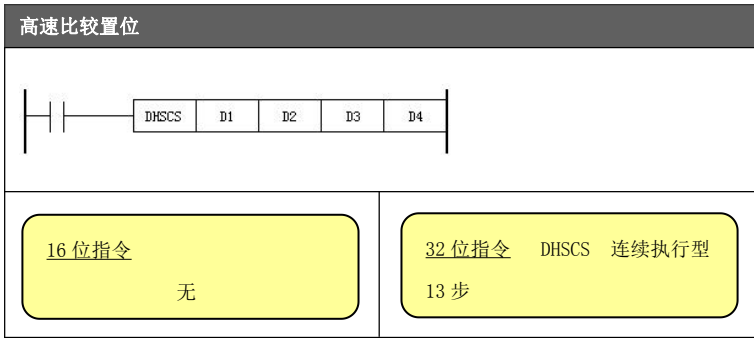
与 DMOV 的区别

针对高速计数器的传送指令区别：

- DHMOV 指令执行，将 D1 指定的计数器立即刷新，然后将计数器当前值传入到 D2。
- DMOV 指令执行，将 D1 指定的计数器当前值传入到 D2。但计数器当前值将在本次扫描周期结束后刷新。

8.3.14 高速比较置位 [DHSCS]

对指定的高速计数值当前值与设定的比较值进行比较，高速计数值当前值与设定的比较值进行比较，由<状态变为≥状态，或由≥状态变为<状态，则置位指定的位元件。



操作数	作用	类型
D1	待比较的脉冲计数值	32 位 BIN
D2	比较计数值对应的寄存器	32 位 BIN
D3	置位的位元件	位
D4	高速数据运算区号	32 位 BIN

操作数	字/位型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●						
D3	X	Y	M	S	I					
		●	●	●						
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定高速计数器待比较的计数值。
- D2 通常指定高速计数器 C235~C255。
- D3 指定比较置位的位元件。
 - ◆ 当高速计数器当前值由<状态变为≥D1 值，或由≥状态变为<D1 值时，比较置位输出的位元件立即置位，不受扫描周期的影响。如果后续条件不满足，该指令也不会执行软元件的复位操作。
- D4 指定指令所占用的高速数据运算区号 0~5。

高速计数器分辨率

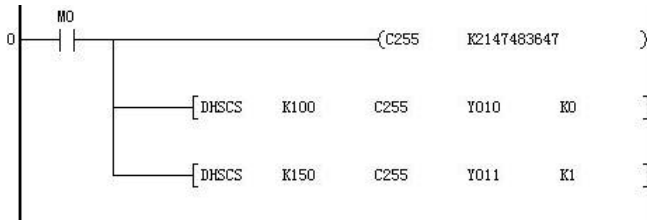
- 高速计数器 C235~C255 的计数分辨率为 0.1ms。
- 数据源为 100KHz 频率脉冲，计数值比较误差 10 个脉冲以内。
 - 数据源小于 10KHz 频率脉冲，计数值比较误差 1 个脉冲以内。

高速数据运算区

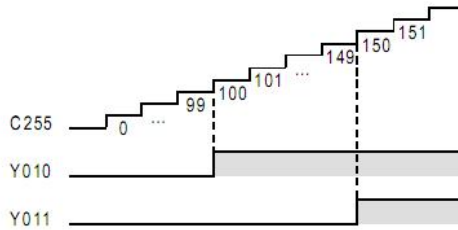
- 高速计数指令 DHSCS, DHSCR, DHSZ 共享 6 个内部的高速数据运算区。
- 高速数据运算区号范围为 0~5。
- 每个高速数据运算区同一时间仅能执行 1 条上述的高速计数指令。使用高速计数指令时需要注意灵活分配运算区。

指令示例一

高速计数器 C255 进行脉冲 AB 相计数，当计数值到 100 时输出 Y10，当计数到 150 时输出 Y11。



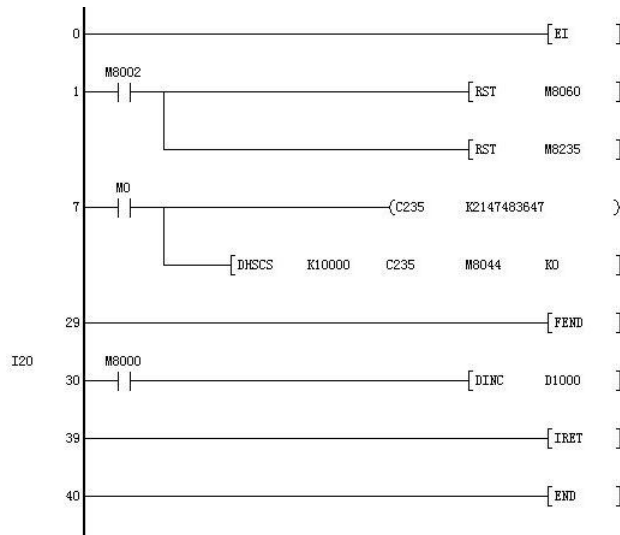
- 要使用 C255 进行高速计数，必须先 OUT 输出，可设置最大计数个数。
- 配置 C255 计数到 100 时输出 Y10，占用高速数据运算区 0。
- 配置 C255 计数到 150 时输出 Y11，占用高速数据运算区 1。



- 当 C255 计数超过 100 时，Y10 由 OFF 变为 ON 状态，若本身为 ON 状态则保持不变。
- 当 C255 计数超过 150 时，Y11 由 OFF 变为 ON 状态，若本身为 ON 状态则保持不变。

指令示例二

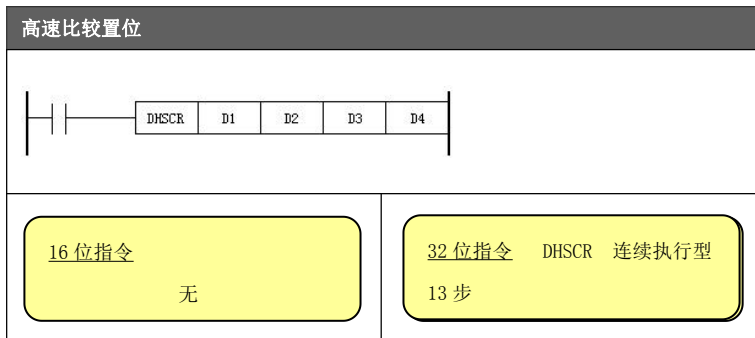
高速计数器 C235 进行增计数，当计数值到 10000 时，进入软中断 I20 处理用户事件。



- EI 打开总中断。
- 复位 M8060，允许 I20；复位 M8235，C235 增计数。
- OUT C255 打开高速计数。高速比较置位计数值超过 10000 时置位 I20 的中断申请位 M8044。
- 主程序结束
- I20 中断服务程序，当 C235 计数值超过 10000 时，进入该中断。
- 中断程序返回
- 程序结束

8.3.15 高速比较复位 [DHSCR]

对指定的高速计数值当前值与设定的比较值进行比较，高速计数值当前值与设定的比较值进行比较，由<状态变为 \geq 状态，或由 \geq 状态变为<状态，则复位指定的位元件。



操作数	作用	类型
D1	待比较的脉冲计数值	32 位 BIN
D2	比较计数值对应的寄存器	32 位 BIN
D3	复位的位元件	位
D4	高速数据运算区号	32 位 BIN

操作数	字/位型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●						
D3	X	Y	M	S						
		●	●	●						
D4	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- D1 指定高速计数器待比较的计数值。
- D2 通常指定高速计数器 C235~C255。
- D3 指定比较复位的位元件。
 - ◆ 当高速计数器当前值由<状态变为 \geq D1 值，或由 \geq 状态变为<D1 值时，比较复位输出的位元件立即复位，不受扫描周期的影响。如果后续条件不满足，该指令也不会执行软元件的置位操作。
- D4 指定指令所占用的高速数据运算区号 0~5。
- 在执行高速比较置位指令之前，需要先 OUT D2 指定高速计数器。

高速计数器分辨率

高速计数器 C235~C255 的计数分辨率为 0.1ms。

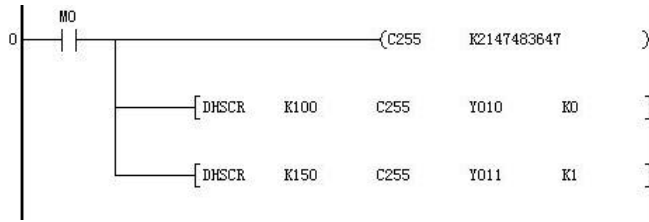
- 数据源为 100KHz 频率脉冲，计数值比较误差 10 个脉冲以内。
- 数据源小于 10KHz 频率脉冲，计数值比较误差 1 个脉冲以内。

高速数据运算区

- 高速计数指令 DHSCS, DHSCR, DHSZ 共享 6 个内部的高速数据运算区。
- 高速数据运算区号范围为 0~5。
- 每个高速数据运算区同一时间仅能执行 1 条上述的高速计数指令。使用高速计数指令时需要注意灵活分配运算区。

指令示例

高速计数器 C255 进行脉冲 AB 相计数，当计数值到 100 时复位 Y10，当计数到 150 时复位 Y11。

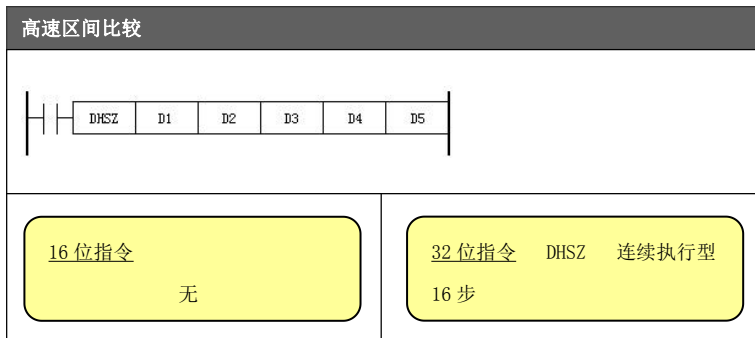


- 要使用 C255 进行高速计数，必须先 OUT 输出，可设置最大计数个数。
- 配置 C255 计数到 100 时复位 Y10，占用高速数据运算区 0。
- 配置 C255 计数到 150 时复位 Y11，占用高速数据运算区 1。

- 当 C255 计数超过 100 时，Y10 由 ON 变为 OFF 状态，若本身为 OFF 状态则保持不变。
- 当 C255 计数超过 150 时，Y11 由 ON 变为 OFF 状态，若本身为 OFF 状态则保持不变。

8.3.16 高速区间比较[DHSZ]

对指定的高速计数值当前值与设定的 2 个比较值(区间)进行数值的比较, 并将比较结果输出至位软元件(3 点)中。



操作数	作用	类型
D1	区间比较值下限	32 位 BIN
D2	区间比较值上限	32 位 BIN
D3	比较计数值对应的寄存器	32 位 BIN
D4	比较输出结果	位
D5	高速数据运算区号	32 位 BIN

操作数	字/位型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●						
D4	X	Y	M	S						
		●	●	●						
D5	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- **D1** 指定高速计数器待比较的计数值区间下限值。
- **D2** 指定高速计数器待比较的计数值区间上限值。
 - ◆ **D2** 的设定值必须大于 **D1** 的设定值。
- **D3** 通常指定高速计数器 C235~C255。
- **D4** 指定比较输出结果的位元件。
 - ◆ 比较输出的位元件满足条件后立即置位, 不受扫描周期的影响。
 - ◆ 指定的 3 个位元件必须处于相同的 16 位地址内。
比如指定 M15 是不正确的, 因为 M15, M16, M17 分别处于不同的 16 位内 (M15 处于 M0~M15 内, M16, M17 处于 M16~M23 内)。
- **D5** 指定指令所占用的高速数据运算区号 0~5。

比较输出动作

以指令 DHSZ K3000 K4000 C235 Y10 K1 举例：

- 区间下限为 3000，区间上限为 4000
- 指定的高速计数器为 C235
- 指令驱动的输出端口是 Y10，Y11，Y12
- 指令占用高速数据运算区号 1

C235 计数值与输出端口的电平状态关系如下表所示：

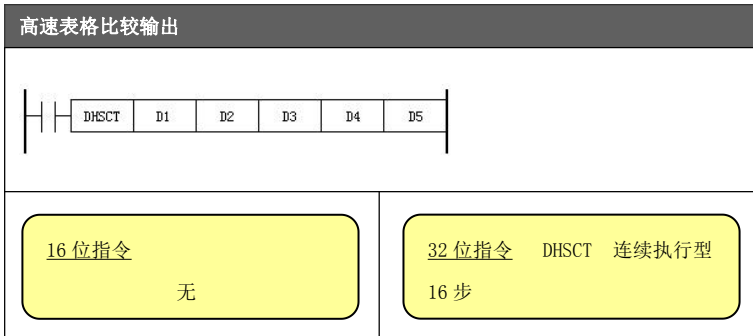
比较条件	Y10 状态	Y11 状态	Y12 状态
$C235 < 3000$	ON	OFF	OFF
$3000 \leq C235 \leq 4000$	OFF	ON	OFF
$C235 > 4000$	OFF	OFF	ON

比较置位 / 复位区别

- DHSZ 区间比较对输出点完全管理其状态，包括置位和复位两种状态。
- DHSCS 高速比较置位和 DHSCR 高速比较复位仅单一管理输出点的状态。置位指令只执行置位操作，复位指令仅执行复位操作。当不满足该条件时，不会对输出点进行反向的操作。

8.3.17 高速表格比较输出 [DHSCT]

对指定的高速计数值当前值与设定的比较表格内当前行的比较值进行比较，满足条件后输出对应的位元件，表格行下移。该过程依次执行，直到表格内所有行的比较结束。



操作数	作用	类型
D1	表格起始单元	32 位 BIN
D2	表格行数	32 位 BIN
D3	比较计数值对应的寄存器	32 位 BIN
D4	输出单元起始地址	位
D5	输出单元使能位	32 位 BIN

操作数	字/位型软元件									
D1	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							
D3	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●	●						
D4	X	Y	M	S						
		●	●	●						
D5	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
	●	●	●							

功能与动作

- **D1** 指定比较表格的起始软元件地址。如 **D1** 指定 D1000，则 D1000 为表格起始单元。
- **D2** 指定比较表格的行数，**D2** 的取值范围为 1~128。
 - ◆ 每个表格行占用三个数据寄存器。其中比较数据 2 个，比较输出值 1 个。
- **D3** 通常指定高速计数器 C235~C255。
- **D4** 指定比较输出结果的位元件起始地址。从起始地址开始指定连续的 16 个位元件。当计数器当前值大于等于表格某行的比较值时，这 16 个位元件输出该行的输出值。
 - ◆ 比较输出的位元件满足条件后立即置位，不受扫描周期的影响。
 - ◆ 指定的位元件起始地址必须被 16 整除。比如 Y0 合法，M3 不合法。
- **D5** 指定比较输出位元件的输出使能位，按位进行设置。
 - ◆ 设置为 1 时，若表格输出值对应位为 1，对应位元件输出。
 - ◆ 设置为 0 时，无论表格输出值是否为 1，对应位元件始终不输出。
 - ◆ 假定比较输出结果的位元件指定 M0，输出使能位为 15。转换成 2 进制为 00000000 00001111。即 M0~M3 输出使能位有效，其他 M 不输出。

表格格式

假定 D1 指定 1000，D2 指定为 5。则其地址分配如下所示：

表格行	比较值	输出值	输出条件
1	D1000 (L)	D1002	计数器当前值 ≥ D1000 (L)
2	D1003 (L)	D1005	计数器当前值 ≥ D1003 (L)
3	D1006 (L)	D1008	计数器当前值 ≥ D1006 (L)
4	D1009 (L)	D1011	计数器当前值 ≥ D1009 (L)
5	D1012 (L)	D1014	计数器当前值 ≥ D1012 (L)

* 比较值必须填写 ≥ 0 的数值。

示例说明

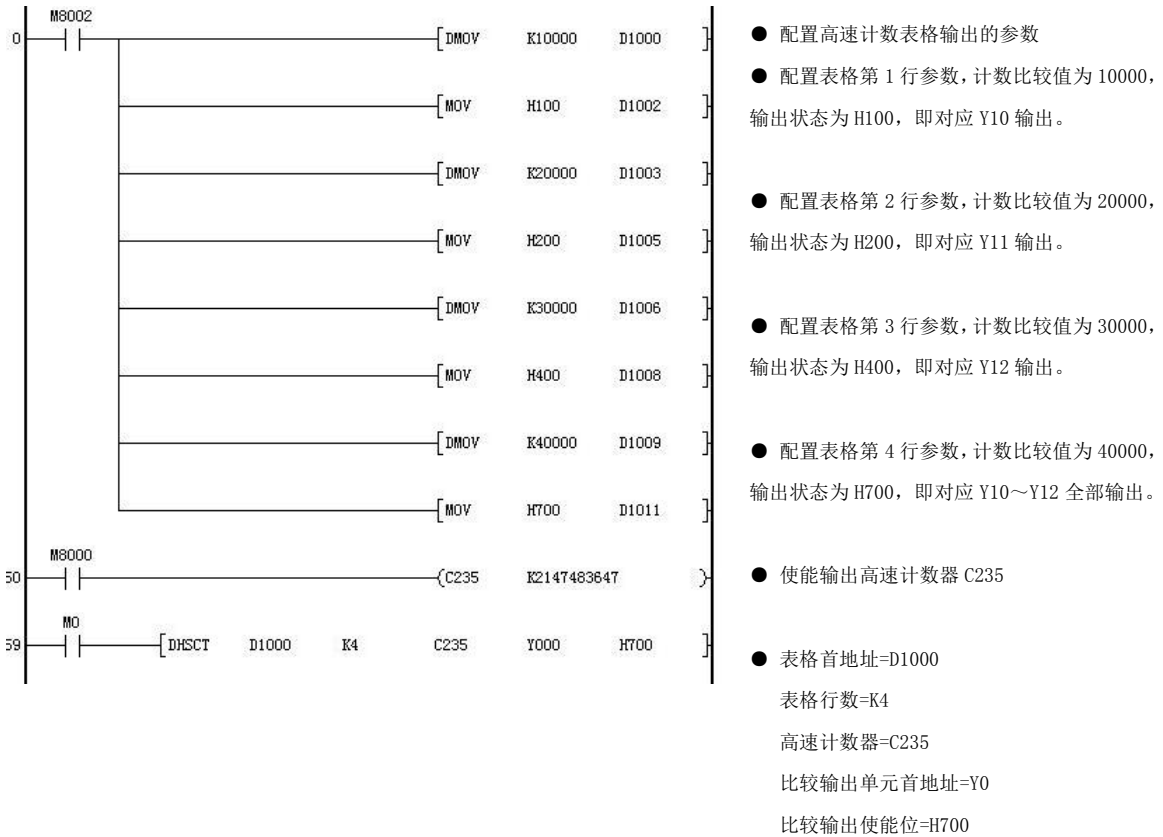
使用计数器 C235，输出元件指定为 Y10~Y12。表格首地址指定为 D1000。C235 的计数值与 Y 端口的输出关系如下表所示：

比较值	输出值
C235 ≥ 10000	Y10=ON, Y11=OFF, Y12=OFF
C235 ≥ 20000	Y10=OFF, Y11=ON, Y12=OFF
C235 ≥ 30000	Y10=OFF, Y11=OFF, Y12=ON
C235 ≥ 40000	Y10=ON, Y11=ON, Y12=ON

上表中共有 4 种计数比较需要输出，故表格行数设计为 4 行。

输出 Y10~Y12，那么首地址应填写 Y0，因为 Y10~Y12 在 Y00~Y15 范围之内。

输出使能的 2 进制数值应为 00000111 00000000，对应 16 进制数为 H0700。

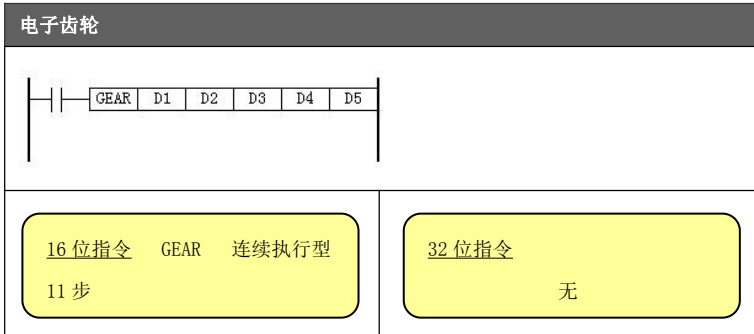


8.4 单轴关联运动

单轴关联运动功能指令指本轴(从轴)与其他轴(主轴)进行运动关联输出。涵盖了电子齿轮、电子凸轮、追剪/滚切制表等指令。

8.4.1 电子齿轮[GEAR]

对指定软件虚拟输出轴 (AXV) 做从轴，与时间/主轴位置成比例关系 (电子齿轮)，比例换算后跟随主轴运动。



操作数	作用	类型
D1	软件虚拟输出轴号(从轴)	16 位 BIN
D2	主轴关联寄存器单元	16 位 BIN
D3	主轴关联寄存器单元变化方式	16 位 BIN
D4	电子齿轮分子	16 位 BIN
D5	电子齿轮分母	16 位 BIN

操作数	字型软元件									
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
D1	●									
D2		●	●							
D3	●	●	●							
D4	●	●	●							
D5	●	●	●							

功能与动作

- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定主轴关联寄存器单元，建议为偶数地址。
- D3 指定主轴关联寄存器单元的变化方式。
 - ◆ = 0，累积型变化。主轴单元增量×电子齿轮比产生的从轴(输出轴)插补量。
适用于累积型主轴，比如软件虚拟输出轴的轴参数 **轴相对位置**。
 - ◆ = 1，增量式变化。主轴单元×电子齿轮比产生的从轴(输出轴)插补量。
适用于增量型主轴，比如软件虚拟输出轴的轴参数 **轴位置插补增量**。
- D4 指定电子齿轮分子，D5 指定电子齿轮分母。D5=0 时，电子齿轮视作 1 处理。
- 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。可在指令导通情况下修改电子齿轮分子或分母，在下一个毫秒周期生效。
- 电子齿轮数值若为负数，则从轴脉冲输出方向与主轴相反。

主轴关联参
考匹配

常见主轴关联	操作数 D2	操作数 D3
软件虚拟输出轴脉冲给定	轴参数 轴相对位置 地址	0
软件虚拟输出轴脉冲给定增量	轴参数 轴位置插补增量 地址	1
系统 1ms 滴答定时器	D8008	0



- 脉冲输出对象为软件虚拟输出轴 1。
- D230 为累积型变化单元(假定)。
- 电子齿轮比为 2:1。

假定软件虚拟输出轴 1 绑定 Y4 脉冲输出口, 则 Y4 输出脉冲为 D230 的脉冲 2 倍频。

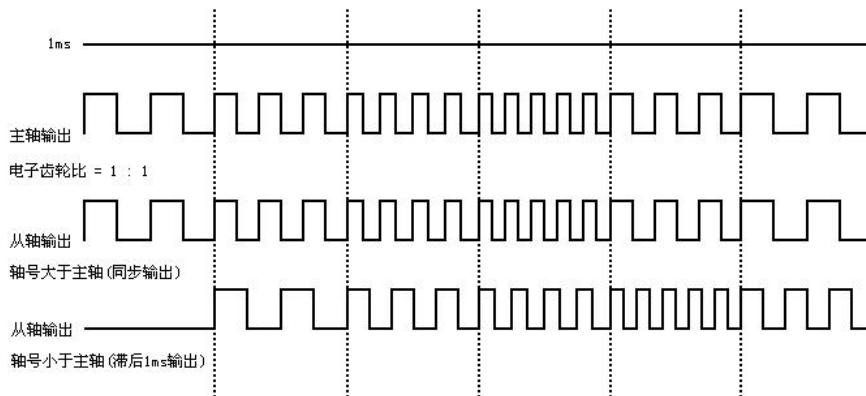


- 脉冲输出对象为软件虚拟输出轴 1。
- 以软件虚拟输出轴 3 的轴参数 轴位置插补增量 为主轴关联单元。
- 轴参数 轴位置插补增量 为增量式变化单元。
- 电子齿轮比为 3:2

假定软件虚拟输出轴 1 绑定 Y4 脉冲输出口, 软件虚拟输出轴 3 绑定数字输出轴 0, 则 Y4 输出脉冲为数字输出轴 0 脉冲数的 1.5 倍频(滞后 1 个插补周期)。

每个插补周期到来时(1ms 中断进入), 系统处理软件虚拟输出轴遵循由小到大的顺序进行脉冲插补处理和更新。假定主轴绑定对应的软件虚拟输出轴号为 AxisM, 从轴对应软件虚拟输出轴号(即 D1 对应数值)为 AxisS。

- AxisM < AxisS, 系统先处理主轴插补计算, 再处理从轴插补计算, 能够保证同插补周期输出。
- AxisM > AxisS, 系统先处理从轴插补计算, 此时主轴所有参数仍处于上一个插补周期的状态, 故从轴将滞后 1 个插补周期输出。



编程注意事项

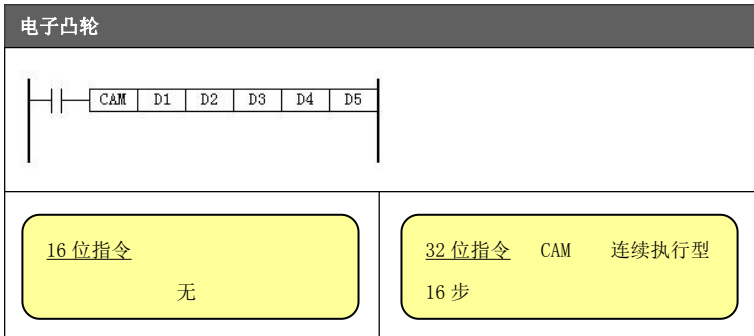
指令执行异常的情况有(指令不动作或动作错误):

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 设置电子齿轮的数值不合理。
 - ◆ 分子/分母设置为 0
 - ◆ 电子齿轮数值太大(从轴输出变化剧烈)造成抖动。
 - ◆ 电子齿轮数值为负数, 且从轴绑定 Y 端口但未绑定方向端口。
- 设置主轴单元寄存器序号为非偶数地址。
- 设置主轴单元变化方式与主轴单元实际变化数值的关系不对应。比如主轴单元填写的是累积型变化数值, 但变化方式填增量型。
- 轴速度设置不合理导致轴参数 `轴错误代码` 报错。
- 在线修改电子齿轮比时, 如果从正电子齿轮比修改到负电子齿轮比, 或由负到正, 系统换向会停发 1 周期的脉冲, 累积到下一周期补发。故容易造成超速报警, 需谨慎使用。
- 对应轴参数 `轴状态` 不为 0 的情况下执行指令。

上电 PLC 运行时 Y0~Y5 端口为梯形图控制, 若配置为运动指令占用且指令执行(脉冲口/方向/事件输出), 则对应 Y 软元件不再关联实际端口。

8.4.2 电子凸轮[CAM]

对指定软件虚拟输出轴 (AXV) 做从轴，与时间/主轴位置成函数关系 (电子凸轮)。主轴变化时，从轴依电子凸轮曲线运动。采用普通制表或专用制表指令产生不同需求的凸轮表 (如追剪/滚切) 表，供凸轮指令使用。



操作数	作用	类型
D1	软件虚拟输出轴号	32 位 BIN
D2	轨迹表格起始单元	32 位 BIN
D3	主轴 (横轴) 实际数值	32 位 BIN
D4	主轴 (横轴) 实际量程	32 位 BIN
D5	凸轮参数缓存区起始单元	32 位 BIN

操作数	字型软元件										
	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
D1											
	●										
D2											
		●	●								
D3											
		●	●								
D4											
	●	●	●								
D5 (占 28 点)											
		●	●								

功能与动作

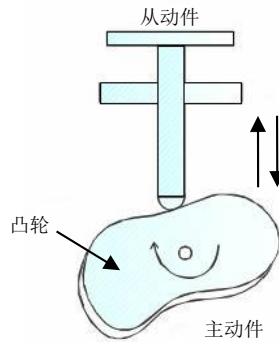
- D1 指定软件虚拟输出轴号 (AXV0~AXV5)，对应数值 0~5。
- D2 指定凸轮曲线轨迹表格的起始寄存器单元，建议为偶数地址。
- D3 指定主轴当前实际值，建议为偶数地址。
 - ◆ 该值若对应时间，则表格输出以主轴时间轴进行输出，可视之为时间凸轮曲线。
 - ◆ 该值若对应位置，则表格输出以主轴位置轴进行输出 (指令推荐用法)。
- D4 指定主轴实际量程。当主轴规划为位置信息时，推荐设置为 1 个凸轮周期内啮合阶段主轴所走的脉冲数，也称之为主轴最大相位。
若主轴实际量程设置与主轴最大相位不相等，两者比例可视作乘以了 1 个电子齿轮的系数，但一般不推荐这样设置。
- D5 指定凸轮中间参数及事件设置的缓存区，相关具体定义后述。
- 必须使用常开触点导通执行指令，断开触点则轴运动立即停止。

电子凸轮功能

传统的机械凸轮机构，是由凸轮、从动件以及机架等三部份组成。

机械凸轮是一种不规则形状的机件，一般为等转速的输入件，可经由直接接触传递运动到从动件，使从动件依设定的规律运动。

从动件为机械凸轮所驱动的被动件，一般为产生不等速、不连续、不规则运动的输出件。机架则是用来支持机械凸轮与从动件的机件。

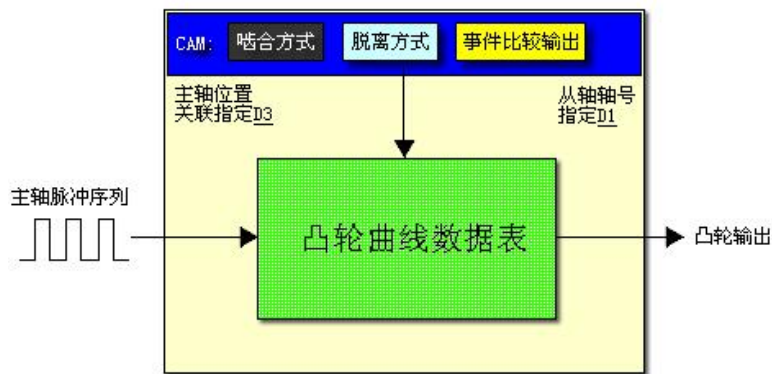


电子凸轮又称 Electronic Cam (E-CAM)，是模拟机械凸轮的一种智能控制器。它通过位置传感器将主轴当前位置反馈给系统，系统将此位置信息进行运算处理，并按设定要求对从轴进行位置输出。

相对传统的方式，使用电子凸轮有着以下的好处：

- 不同的工艺需要不同的凸轮曲线，可通过修改电子凸轮表内的数据，而不用修改机械部件。
- 可以有较高的加减速性能。
- 运行可以较为平稳。

电子凸轮指令功能框图：



- 凸轮曲线数据表：E-CAM 数据表，由制表指令、专用制表指令(追剪/滚切制表)或上位机辅助工具生成。为一个循环凸轮周期中从轴与主轴的位置相对关系的表格。
- 啮合：启动电子凸轮。以啮合点的主轴位置为相对零点，从轴根据主轴的当前位置走 E-CAM 表中的凸轮曲线(输出插补值)。
- 脱离：停止电子凸轮。从轴不再跟随主轴位置的变化而变化。
- 事件比较输出：凸轮在某一相位产生相应的动作，需事件输出驱动(如追剪、滚切)。

凸轮参数缓存区 D5

凸轮参数缓存区一共占用 28 个寄存器单元，如 D5 指定 D1200，则 D1200~D1227 寄存器区域在电子凸轮指令执行时被占用。

地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

D5+偏移	名称	参数描述	单位
D5+0	相位偏差补偿	<p>主轴虚拟相位补偿辅助轴位置值，用于消除长时间运行后由于机械或其他原因造成的主从轴相位偏差。该参数为增量式位置值。</p> <p>凸轮运行时，修改该参数将会把偏差值叠加到当前周期主轴相位上(主轴相位概念解释参考下面的<u>主轴当前相位</u>参数)。在一个插补周期时间内，从轴按照叠加后的主轴相位输出到轨迹表的相应位置。若补偿值较大，建议进行分时插补补偿，否则在一个插补周期内从轴位置变化剧烈可能导致造成轴速过大，引发轴错误。</p> <p>例：运行一段时间后主轴相位偏差了 10 个脉冲，则<u>相位偏差补偿</u>填 10 消除主从轴相位偏差。若再次偏差 5 个脉冲，则<u>相位偏差补偿</u>填 $10 + 5 = 15$。</p> <ul style="list-style-type: none"> ● 该参数填写寄存器的序号，对应占用 2 个 D 寄存器单元。例如填写 K1400，则<u>相位偏差补偿</u>对应寄存器为 D1400(L)，运行时修改该寄存器数值即设置生效。 ● 当主轴当前相位在边界值附近时(即刚进入啮合阶段或离开啮合阶段)，修改<u>相位偏差补偿</u>可能导致主轴当前相位越界。一般应避免该情况发生，可能导致凸轮时序运行异常。 ● 脱离阶段修改<u>相位偏差补偿</u>，且导致主轴相位叠加值进入凸轮啮合区。系统将会进入到啮合阶段，但此次的多余部分脉冲并不会叠加到主轴相位上。 <p>例：主轴只需走 10 个脉冲即可进入啮合区，此时<u>相位偏差补偿</u>从 0 修改为 20，则凸轮将进入啮合阶段，数值 20 作为<u>相位偏差补偿</u>的更新基准值。从轴输出位置为主轴相位为 0 时的轨迹表点位，而非主轴相位为 10 时的轨迹表点位。</p>	-
D5+1	保留参数。		
D5+2(L)	初始前置啮合位移	<p>初始前置啮合位移量，仅首个凸轮周期生效。一般应用于跳过主轴空行程的需求。</p> <ul style="list-style-type: none"> ● 凸轮表格同步区位于周期正中处，可通过该参数调整主/从轴的同步区偏移量。 ● 该参数仅在啮合方式中设置后，方才生效。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 不能设置该参数 < 0。 	脉冲
D5+4(L)	周期前置啮合位移	<p>周期前置啮合位移量，首个凸轮周期流程结束后开始生效。</p> <ul style="list-style-type: none"> ● 凸轮表格同步区位于周期正中处，可通过该参数调整主/从轴的同步区偏移量。 ● 该参数仅在啮合方式中设置后，方才生效。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 不能设置该参数 < 0。 	脉冲
D5+6(L)	相位脱离位移	<p>主轴相位脱离对应位移量，运行到指定的主轴相位处进行脱离。此参数一般应用于单次非循环的凸轮工艺。</p> <ul style="list-style-type: none"> ● 该参数仅在脱离方式中设置后，方才生效。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 不能设置该参数 < 0 或 $>$ CAM 指令操作数 D4 数值(主轴最大相位)。 	脉冲
D5+8~D5+11	保留参数。		

D5+偏移	名称	参数描述	单位																																								
D5+12(L)	主轴当前相位	<p>主轴的位置为累积型脉冲数值，每个凸轮周期开始时将主轴的位置记录为相对零点。主轴相位是指该凸轮周期内主轴的相对位移，即主轴当前周期的实际位置减去该周期的相对零点位置之差。</p> <p>一个凸轮周期主轴相位最大值即为主轴实际量程，此参数由电子凸轮指令的操作数 <u>D4</u> 确定，满足主轴最大相位 > <u>相位脱离位移</u> ≥ 0 的关系。</p> <p>若凸轮脱离后又立即啮合，则当前凸轮周期的主轴最大相位位置点即为下一个凸轮周期的零点。</p> <ul style="list-style-type: none"> ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 该参数无需设置，为显示值。 	脉冲																																								
D5+14	啮合方式	<p>该参数决定凸轮的啮合方式。</p> <ul style="list-style-type: none"> ● 0 - 指令生效啮合(指令得电状态) ● 1 - 输入触发啮合 ● 2 - 前置位移量啮合 <ul style="list-style-type: none"> ◆ 首次周期流程，主轴走初始前置啮合位移后开始啮合。 ◆ 非首次周期流程，主轴走周期前置啮合位移后开始啮合。 ● 3 - 输入触发 + 前置啮合 <ul style="list-style-type: none"> ◆ 首次周期流程，输入触发啮合生效后，主轴经过初始前置啮合位移后开始啮合。 ◆ 非首次周期流程，输入触发啮合生效后，主轴经过周期前置啮合位移后开始啮合。 <p>必须是在脱离状态下，系统才去判断啮合条件。</p> <p>凸轮轨迹表的起始点通常设计为从轴原点(即轨迹表第一个型值点的位置)，在这种情况下，需保证啮合位置点物理上也为从轴的原点位置。</p>	-																																								
D5+15	啮合输入触发	<p>啮合触发输入使能，按位进行设置(掩码)。</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>数值设定 =</th> <th>选定端口</th> <th>数值设定 =</th> <th>选定端口</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>不选定端口</td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>X0/M8096</td> <td>256</td> <td>X10/M8104</td> </tr> <tr> <td>2</td> <td>X1/M8097</td> <td>512</td> <td>X11/M8105</td> </tr> <tr> <td>4</td> <td>X2/M8098</td> <td>1024</td> <td>X12/M8106</td> </tr> <tr> <td>8</td> <td>X3/M8099</td> <td>2048</td> <td>X13/M8107</td> </tr> <tr> <td>16</td> <td>X4/M8100</td> <td>4096</td> <td>X14(虚拟)/M8108</td> </tr> <tr> <td>32</td> <td>X5/M8101</td> <td>8192</td> <td>X15(虚拟)/M8109</td> </tr> <tr> <td>64</td> <td>X6/M8102</td> <td>16384</td> <td>X16(虚拟)/M8110</td> </tr> <tr> <td>128</td> <td>X7/M8103</td> <td>-32768</td> <td>X17(虚拟)/M8111</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ● 通过轴事件输入点.Bit13 可使用 M8096~M8111 代替 X00~X17 的控制。 ● 虚拟端口 X14~X17 可通过 M8160~M8163 进行控制。 ● 该参数仅在啮合方式中设置后，方才生效。 	数值设定 =	选定端口	数值设定 =	选定端口	0	不选定端口			1	X0/M8096	256	X10/M8104	2	X1/M8097	512	X11/M8105	4	X2/M8098	1024	X12/M8106	8	X3/M8099	2048	X13/M8107	16	X4/M8100	4096	X14(虚拟)/M8108	32	X5/M8101	8192	X15(虚拟)/M8109	64	X6/M8102	16384	X16(虚拟)/M8110	128	X7/M8103	-32768	X17(虚拟)/M8111	-
数值设定 =	选定端口	数值设定 =	选定端口																																								
0	不选定端口																																										
1	X0/M8096	256	X10/M8104																																								
2	X1/M8097	512	X11/M8105																																								
4	X2/M8098	1024	X12/M8106																																								
8	X3/M8099	2048	X13/M8107																																								
16	X4/M8100	4096	X14(虚拟)/M8108																																								
32	X5/M8101	8192	X15(虚拟)/M8109																																								
64	X6/M8102	16384	X16(虚拟)/M8110																																								
128	X7/M8103	-32768	X17(虚拟)/M8111																																								

D5+偏移	名称	参数描述	单位																																		
D5+16	啮合输入触发极性	<p>啮合触发输入极性，按位设置选定端口的极性(掩码)。</p> <table border="1" data-bbox="603 315 1249 1709"> <thead> <tr> <th>啮合输入触发极性.Bit</th> <th>定义</th> </tr> </thead> <tbody> <tr> <td>Bit0</td> <td>= 0: 信号进入 X0 端口为 ON, 离开为 OFF = 1: 信号进入 X0 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit1</td> <td>= 0: 信号进入 X1 端口为 ON, 离开为 OFF = 1: 信号进入 X1 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit2</td> <td>= 0: 信号进入 X2 端口为 ON, 离开为 OFF = 1: 信号进入 X2 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit3</td> <td>= 0: 信号进入 X3 端口为 ON, 离开为 OFF = 1: 信号进入 X3 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit4</td> <td>= 0: 信号进入 X4 端口为 ON, 离开为 OFF = 1: 信号进入 X4 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit5</td> <td>= 0: 信号进入 X5 端口为 ON, 离开为 OFF = 1: 信号进入 X5 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit6</td> <td>= 0: 信号进入 X6 端口为 ON, 离开为 OFF = 1: 信号进入 X6 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit7</td> <td>= 0: 信号进入 X7 端口为 OFF, 离开为 ON = 1: 信号进入 X7 端口为 ON, 离开为 OFF</td> </tr> <tr> <td>Bit8</td> <td>= 0: 信号进入 X10 端口为 ON, 离开为 OFF = 1: 信号进入 X10 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit9</td> <td>= 0: 信号进入 X11 端口为 ON, 离开为 OFF = 1: 信号进入 X11 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit10</td> <td>= 0: 信号进入 X12 端口为 ON, 离开为 OFF = 1: 信号进入 X12 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit11</td> <td>= 0: 信号进入 X13 端口为 ON, 离开为 OFF = 1: 信号进入 X13 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit12</td> <td>= 0: 信号进入 X14 端口为 ON, 离开为 OFF = 1: 信号进入 X14 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit13</td> <td>= 0: 信号进入 X15 端口为 ON, 离开为 OFF = 1: 信号进入 X15 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit14</td> <td>= 0: 信号进入 X16 端口为 ON, 离开为 OFF = 1: 信号进入 X16 端口为 OFF, 离开为 ON</td> </tr> <tr> <td>Bit15</td> <td>= 0: 信号进入 X17 端口为 ON, 离开为 OFF = 1: 信号进入 X17 端口为 OFF, 离开为 ON</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ● 通过轴事件输入点, Bit13 可使用 M8096~M8111 代替 X00~X17 的控制。 ● 该参数仅在啮合输入触发中设置后, 方才生效。 	啮合输入触发极性.Bit	定义	Bit0	= 0: 信号进入 X0 端口为 ON, 离开为 OFF = 1: 信号进入 X0 端口为 OFF, 离开为 ON	Bit1	= 0: 信号进入 X1 端口为 ON, 离开为 OFF = 1: 信号进入 X1 端口为 OFF, 离开为 ON	Bit2	= 0: 信号进入 X2 端口为 ON, 离开为 OFF = 1: 信号进入 X2 端口为 OFF, 离开为 ON	Bit3	= 0: 信号进入 X3 端口为 ON, 离开为 OFF = 1: 信号进入 X3 端口为 OFF, 离开为 ON	Bit4	= 0: 信号进入 X4 端口为 ON, 离开为 OFF = 1: 信号进入 X4 端口为 OFF, 离开为 ON	Bit5	= 0: 信号进入 X5 端口为 ON, 离开为 OFF = 1: 信号进入 X5 端口为 OFF, 离开为 ON	Bit6	= 0: 信号进入 X6 端口为 ON, 离开为 OFF = 1: 信号进入 X6 端口为 OFF, 离开为 ON	Bit7	= 0: 信号进入 X7 端口为 OFF, 离开为 ON = 1: 信号进入 X7 端口为 ON, 离开为 OFF	Bit8	= 0: 信号进入 X10 端口为 ON, 离开为 OFF = 1: 信号进入 X10 端口为 OFF, 离开为 ON	Bit9	= 0: 信号进入 X11 端口为 ON, 离开为 OFF = 1: 信号进入 X11 端口为 OFF, 离开为 ON	Bit10	= 0: 信号进入 X12 端口为 ON, 离开为 OFF = 1: 信号进入 X12 端口为 OFF, 离开为 ON	Bit11	= 0: 信号进入 X13 端口为 ON, 离开为 OFF = 1: 信号进入 X13 端口为 OFF, 离开为 ON	Bit12	= 0: 信号进入 X14 端口为 ON, 离开为 OFF = 1: 信号进入 X14 端口为 OFF, 离开为 ON	Bit13	= 0: 信号进入 X15 端口为 ON, 离开为 OFF = 1: 信号进入 X15 端口为 OFF, 离开为 ON	Bit14	= 0: 信号进入 X16 端口为 ON, 离开为 OFF = 1: 信号进入 X16 端口为 OFF, 离开为 ON	Bit15	= 0: 信号进入 X17 端口为 ON, 离开为 OFF = 1: 信号进入 X17 端口为 OFF, 离开为 ON	-
啮合输入触发极性.Bit	定义																																				
Bit0	= 0: 信号进入 X0 端口为 ON, 离开为 OFF = 1: 信号进入 X0 端口为 OFF, 离开为 ON																																				
Bit1	= 0: 信号进入 X1 端口为 ON, 离开为 OFF = 1: 信号进入 X1 端口为 OFF, 离开为 ON																																				
Bit2	= 0: 信号进入 X2 端口为 ON, 离开为 OFF = 1: 信号进入 X2 端口为 OFF, 离开为 ON																																				
Bit3	= 0: 信号进入 X3 端口为 ON, 离开为 OFF = 1: 信号进入 X3 端口为 OFF, 离开为 ON																																				
Bit4	= 0: 信号进入 X4 端口为 ON, 离开为 OFF = 1: 信号进入 X4 端口为 OFF, 离开为 ON																																				
Bit5	= 0: 信号进入 X5 端口为 ON, 离开为 OFF = 1: 信号进入 X5 端口为 OFF, 离开为 ON																																				
Bit6	= 0: 信号进入 X6 端口为 ON, 离开为 OFF = 1: 信号进入 X6 端口为 OFF, 离开为 ON																																				
Bit7	= 0: 信号进入 X7 端口为 OFF, 离开为 ON = 1: 信号进入 X7 端口为 ON, 离开为 OFF																																				
Bit8	= 0: 信号进入 X10 端口为 ON, 离开为 OFF = 1: 信号进入 X10 端口为 OFF, 离开为 ON																																				
Bit9	= 0: 信号进入 X11 端口为 ON, 离开为 OFF = 1: 信号进入 X11 端口为 OFF, 离开为 ON																																				
Bit10	= 0: 信号进入 X12 端口为 ON, 离开为 OFF = 1: 信号进入 X12 端口为 OFF, 离开为 ON																																				
Bit11	= 0: 信号进入 X13 端口为 ON, 离开为 OFF = 1: 信号进入 X13 端口为 OFF, 离开为 ON																																				
Bit12	= 0: 信号进入 X14 端口为 ON, 离开为 OFF = 1: 信号进入 X14 端口为 OFF, 离开为 ON																																				
Bit13	= 0: 信号进入 X15 端口为 ON, 离开为 OFF = 1: 信号进入 X15 端口为 OFF, 离开为 ON																																				
Bit14	= 0: 信号进入 X16 端口为 ON, 离开为 OFF = 1: 信号进入 X16 端口为 OFF, 离开为 ON																																				
Bit15	= 0: 信号进入 X17 端口为 ON, 离开为 OFF = 1: 信号进入 X17 端口为 OFF, 离开为 ON																																				

D5+偏移	名称	参数描述	单位																
D5+17	脱离方式	<p>脱离方式选择</p> <ul style="list-style-type: none"> ● 0 - 周期脱离(运行完一个凸轮周期后自动脱离) ● 1 - 输入触发脱离 ● 2 - 相位脱离, 主轴走到相位脱离位移后脱离 ● 3 - 输入触发 + 相位脱离, 先满足输入触发后, 再走到相位脱离位移移量处, 然后脱离。若输入触发产生时, 主轴相位在相位脱离位移之后, 则凸轮将立即脱离。 <p>例如: 假定主轴最大相位为 1000, 相位脱离位移为 800。若触发输入发生时, 主轴当前相位为 500, 则运行到本周期 800 处脱离; 主轴当前相位为 900, 则直接脱离。</p> <p>脱离方式选择 1/2/3 方式时, 方式 0 仍然有效。即凸轮周期结束时, 无论脱离条件是否满足, 凸轮也将脱离。</p> <p>凸轮指令断开时, 无论凸轮处于什么状态, 立即脱离。</p> <p>必须在啮合状态下, 系统才去判断脱离条件。脱离后若指令仍处于得电状态, 系统开始判断啮合条件, 满足啮合条件后会再次啮合。</p> <p>必须确保主从轴在某一相对安全位置时脱离, 比如追剪系统, 若在同步区进行脱离, 由于事件输出端口(切刀)仍处于输出状态, 可能导致人身安全危险以及机件损坏!</p>	-																
D5+18	脱离输入触发	<p>脱离触发输入使能, 参考啮合输入触发参数设置。</p> <ul style="list-style-type: none"> ● 通常情况下, 不能和啮合输入触发指定同一个 X 端口。 	-																
D5+19	脱离输入触发极性	<p>脱离触发输入极性, 按位设置选定端口极性(掩码)。参考啮合输入触发参数设置。</p> <ul style="list-style-type: none"> ● 该参数仅在脱离输入触发中设置后, 方才生效。 	-																
D5+20	比较事件输出表格地址	<p>指定比较表格的起始软元件地址。</p> <ul style="list-style-type: none"> ● 该参数填写寄存器的序号, 例如填写 K1000, 则比较事件输出表格地址对应寄存器为 D1000, 运行时修改该寄存器数值即设置生效。 ● 比较表格输出表格的每个表格行占用三个数据寄存器。其中比较数据 2 个, 比较输出值 1 个。具体格式参考章节 8.3.17 描述。 	-																
D5+21	比较事件输出表格行数	<p>指定比较表格的输出行数。</p> <p>比较事件的表格输出格式填写方法参考章节 8.3.17 描述。</p>	-																
D5+22	比较事件输出元件	<p>从起始地址开始指定连续的 16 个位元件。对应的数值与位元件的关系如下表所示:</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>比较事件输出元件</th> <th>定义</th> </tr> </thead> <tbody> <tr> <td>0~255</td> <td>Y000~Y□□□, 根据运动控制器型号选填</td> </tr> <tr> <td>1000~1255</td> <td>X000~X□□□, 根据运动控制器型号选填</td> </tr> <tr> <td>2000~2255</td> <td>T0~T255, 触点型</td> </tr> <tr> <td>3000~3255</td> <td>C0~C255, 触点型</td> </tr> <tr> <td>4000~4255</td> <td>M8000~M8255</td> </tr> <tr> <td>5000~5999</td> <td>S0~S999</td> </tr> <tr> <td>10000~13071</td> <td>M0~M3071</td> </tr> </tbody> </table> <p>当主轴当前实际值 ≥ 表格某行的比较值时, 这 16 个位元件输出该行的输出值。</p> <ul style="list-style-type: none"> ● 指定的位元件起始地址必须被 16 整除。 	比较事件输出元件	定义	0~255	Y000~Y□□□, 根据运动控制器型号选填	1000~1255	X000~X□□□, 根据运动控制器型号选填	2000~2255	T0~T255, 触点型	3000~3255	C0~C255, 触点型	4000~4255	M8000~M8255	5000~5999	S0~S999	10000~13071	M0~M3071	-
比较事件输出元件	定义																		
0~255	Y000~Y□□□, 根据运动控制器型号选填																		
1000~1255	X000~X□□□, 根据运动控制器型号选填																		
2000~2255	T0~T255, 触点型																		
3000~3255	C0~C255, 触点型																		
4000~4255	M8000~M8255																		
5000~5999	S0~S999																		
10000~13071	M0~M3071																		

D5+偏移	名称	参数描述	单位																									
D5+23	比较事件输出元件使能	<p>指定比较输出元件的输出使能位，按位进行设置。</p> <ul style="list-style-type: none"> ● 设置为 1 时，若表格输出值对应位为 1，对应位元件输出。 ● 设置为 0 时，无论表格输出值是否为 1，对应位元件始终不输出。 ● 设置事件比较输出元件和对应的轴输出端口不要产生冲突。比如凸轮指令指定 Y0 输出，那么必须关闭 Y0 的脉冲输出使能(配置 D8180=0)。 	-																									
D5+24	凸轮运行状态	<p>凸轮当前运行状态的显示，指令运行过程中进行检测。</p> <ul style="list-style-type: none"> ● = 0: 凸轮脱离状态。 ● = 1: 凸轮啮合状态。 ● 该参数无需设置，为显示值。 	-																									
D5+25	凸轮起始状态	<p>在启动凸轮指令前，必须设置该参数。</p> <p>凸轮指令启动时，凸轮运行状态默认为凸轮起始状态。</p> <ul style="list-style-type: none"> ● 在首次运行凸轮指令时，该参数推荐填 0。在前次凸轮指令运行时，由于意外停止指令。此时应获取结束时的凸轮运行状态，并将其配置到凸轮起始状态，以使本次凸轮指令启动后，会沿用上次凸轮结束时的状态继续运行。 	-																									
D5+26(L)	凸轮起始相位	<p>在启动凸轮指令前，必须设置该参数。</p> <p>该参数与凸轮其他参数的关系如下表所示：</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>凸轮起始状态</th> <th>凸轮起始相位</th> <th>啮合方式</th> <th>后续运行状态</th> </tr> </thead> <tbody> <tr> <td rowspan="6">=0</td> <td rowspan="4">=0</td> <td>=0</td> <td>立即啮合，凸轮起始相位为当前已发生的主轴当前相位</td> </tr> <tr> <td>=1</td> <td>输入触发后啮合</td> </tr> <tr> <td>=2</td> <td>前置位移触发后啮合</td> </tr> <tr> <td>=3</td> <td>输入+前置位移触发后啮合</td> </tr> <tr> <td rowspan="2">>0</td> <td>=0/1</td> <td>立即啮合，凸轮起始相位为当前已发生的主轴当前相位</td> </tr> <tr> <td>=2</td> <td>凸轮起始相位为当前已发生的前置啮合位移 ● 凸轮起始相位不要填写超过前置啮合位移的数值</td> </tr> <tr> <td rowspan="2">=1</td> <td>=0</td> <td rowspan="2">=0/1/2/3</td> <td>输入啮合条件已触发，凸轮起始相位为当前已发生的前置啮合位移 ● 凸轮起始相位不要填写超过前置啮合位移的数值</td> </tr> <tr> <td>>0</td> <td>立即啮合，凸轮起始相位为当前已发生的主轴当前相位</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ● 在首次运行凸轮指令时，该参数推荐填 0。在前次凸轮指令运行时，由于意外停止指令。此时应获取结束时的凸轮运行状态，并根据当前主轴相位，啮合方式等条件填写凸轮起始相位，以使本次凸轮指令启动后，会沿用上次凸轮结束时的状态继续运行。 ● 当凸轮起始相位设置不为 0 时，根据当前主轴相位的数值，在比较输出表格中找到上 1 个小于该数值的输出状态，并进行比较事件输出元件输出。 	凸轮起始状态	凸轮起始相位	啮合方式	后续运行状态	=0	=0	=0	立即啮合，凸轮起始相位为当前已发生的主轴当前相位	=1	输入触发后啮合	=2	前置位移触发后啮合	=3	输入+前置位移触发后啮合	>0	=0/1	立即啮合，凸轮起始相位为当前已发生的主轴当前相位	=2	凸轮起始相位为当前已发生的前置啮合位移 ● 凸轮起始相位不要填写超过前置啮合位移的数值	=1	=0	=0/1/2/3	输入啮合条件已触发，凸轮起始相位为当前已发生的前置啮合位移 ● 凸轮起始相位不要填写超过前置啮合位移的数值	>0	立即啮合，凸轮起始相位为当前已发生的主轴当前相位	脉冲
凸轮起始状态	凸轮起始相位	啮合方式	后续运行状态																									
=0	=0	=0	立即啮合，凸轮起始相位为当前已发生的主轴当前相位																									
		=1	输入触发后啮合																									
		=2	前置位移触发后啮合																									
		=3	输入+前置位移触发后啮合																									
	>0	=0/1	立即啮合，凸轮起始相位为当前已发生的主轴当前相位																									
		=2	凸轮起始相位为当前已发生的前置啮合位移 ● 凸轮起始相位不要填写超过前置啮合位移的数值																									
=1	=0	=0/1/2/3	输入啮合条件已触发，凸轮起始相位为当前已发生的前置啮合位移 ● 凸轮起始相位不要填写超过前置啮合位移的数值																									
	>0		立即啮合，凸轮起始相位为当前已发生的主轴当前相位																									

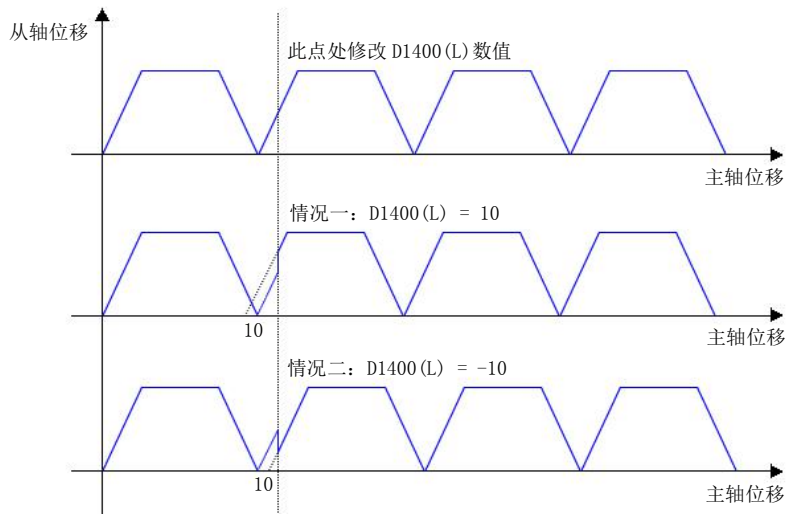
D5+偏移	名称	参数描述	单位										
D5+28	辅助轴前置电子齿轮元件	指定辅助轴前置电子齿轮的元件地址。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>辅助轴前置电子齿轮元件</th> <th>定义</th> </tr> </thead> <tbody> <tr> <td>0~7999</td> <td>对应 D0~D7999</td> </tr> <tr> <td>8000~8255</td> <td>对应 D8000~D8255</td> </tr> <tr> <td>9000~9255</td> <td>对应 C0~C255</td> </tr> <tr> <td>10000~10255</td> <td>对应 T0~T255</td> </tr> </tbody> </table> ● 辅助轴用于在线调整长度。	辅助轴前置电子齿轮元件	定义	0~7999	对应 D0~D7999	8000~8255	对应 D8000~D8255	9000~9255	对应 C0~C255	10000~10255	对应 T0~T255	-
辅助轴前置电子齿轮元件	定义												
0~7999	对应 D0~D7999												
8000~8255	对应 D8000~D8255												
9000~9255	对应 C0~C255												
10000~10255	对应 T0~T255												
D5+29	辅助轴前置电子齿轮分子	用于设置辅助轴前置电子齿轮的分子。	-										
D5+30	辅助轴前置电子齿轮分母	用于设置辅助轴前置电子齿轮的分母，该参数不能设置为 0。	-										

**相位偏差
补偿分析**

在凸轮指令启动时，系统以当前相位偏差补偿数值为基准零点，在运行过程中修改该寄存器，则将修改后的增量数值作为偏差值叠加到主轴相位上，同时从轴（即凸轮轴）也将输出到叠加后主轴相位对应表格的从轴输出点位置。

- 通常情况下对凸轮参数缓存区初始化时应将相位偏差补偿对应的 32 位寄存器数值清零。

假定相位偏差补偿值对应的寄存器为 D1400(L)，初始化数值为 0。其应用效果示意图如下所示：



啮合/脱离
方式分析

凸轮指令分别各有 4 种啮合方式和 4 种脱离方式，一共有 16 种工作模式的选择。如下表所示。

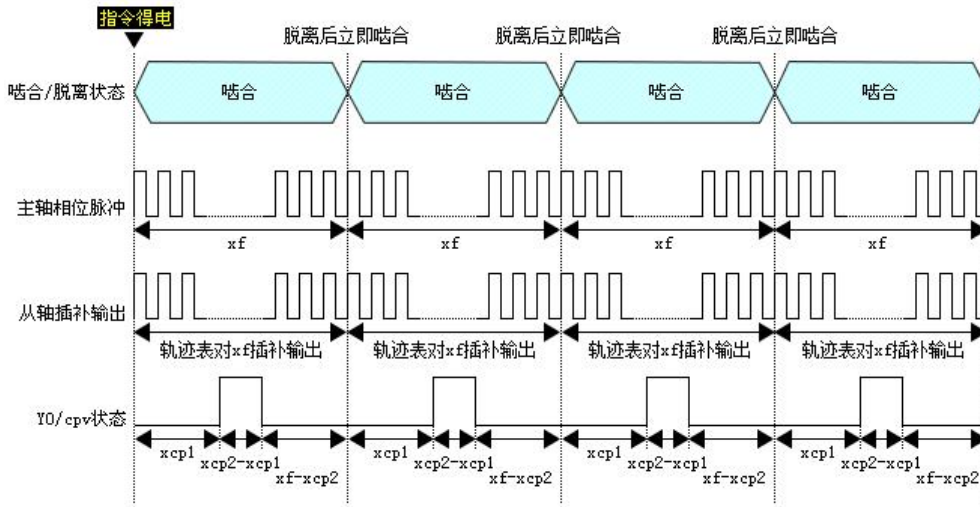
啮合方式 脱离方式	指令生效啮合	输入触发啮合	前置位移量啮合	输入触发+前置位移量啮合
周期脱离	A	B	C	D
输入触发脱离	E	F	G	H
主轴相位脱离	I	J	K	L
输入触发+主轴相位脱离	M	N	O	P

- 无论选择哪种啮合/脱离方式，任意情况下梯形图扫描到断开凸轮指令的操作，凸轮(从轴)立即脱离。
- 无论选择哪种脱离方式，周期脱离始终有效。即凸轮周期结束时，即使设定的脱离条件不满足，凸轮也将立即脱离。
- 凸轮脱离后系统时刻判断啮合条件，若啮合条件满足则立即啮合，故存在脱离后马上啮合的情况(例如方式 A)。由于脱离非常短暂，此种情况下可认为凸轮始终处于啮合状态。
- 假定凸轮输出表格输出元件为 Y0，对应主轴相位变化位移状态变化为 2 个(即表格输出有 2 行)。

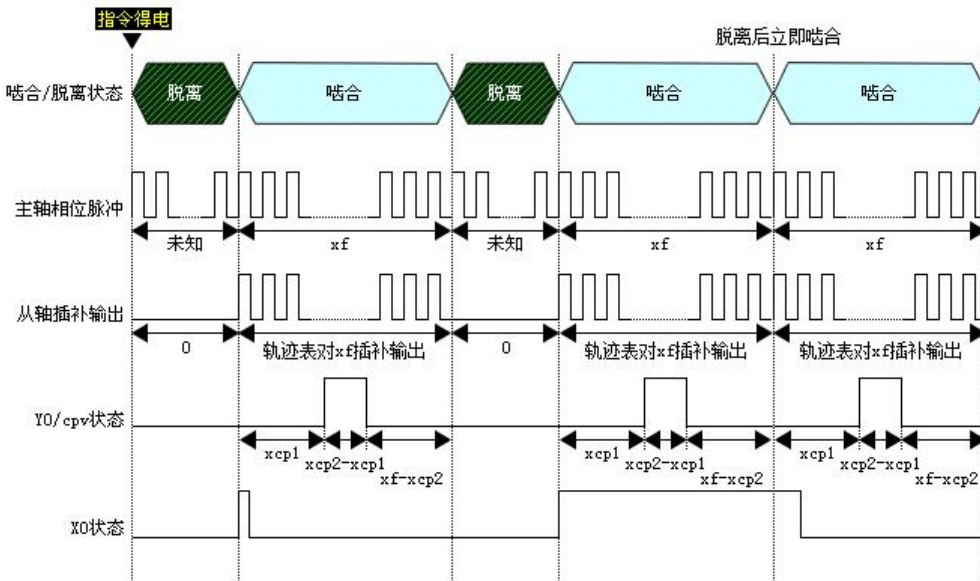
A~P 组合情况的时序分析，其他参数的设置情况以及时序图中所用到的助记符如下所示：

指令运行状态	始终得电(生效)		
输入触发啮合	X0	输入触发啮合极性	信号进入 ON，信号离开 OFF。 时序图上信号来源仅为说明凸轮的工作流程，并不代表实际工况。
输入触发脱离	X1	输入触发脱离极性	信号进入 ON，信号离开 OFF。 时序图上信号来源仅为说明凸轮的工作流程，并不代表实际工况。
比较事件端口	Y0	比较事件端口极性	ON 输出高电平，OFF 输出低电平。
时序图助记符	xf	主轴最大相位(主轴实际量程)，指令操作数 D4 指定。	
	xe	相位脱离位移	
	xn	主轴当前相位	
	xcp1	表格输出第 1 行位移	
	xcp2	表格输出第 2 行位移	
	xs	初始前置啮合位移	
	xs1	周期前置啮合位移	
	cpv	比较事件状态	

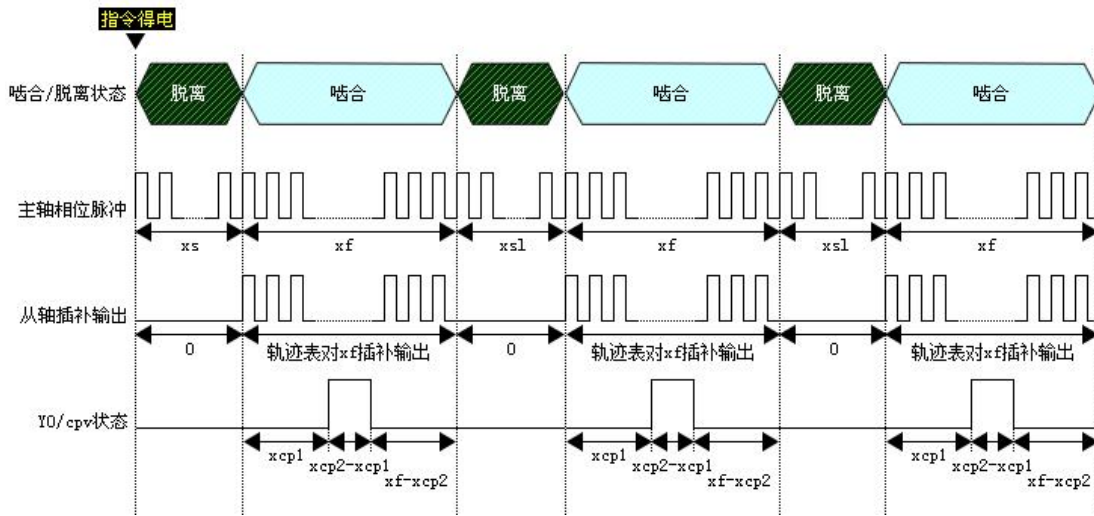
- 组合 A:
 - ◆ 指令生效则啮合。
 - ◆ 周期脱离。



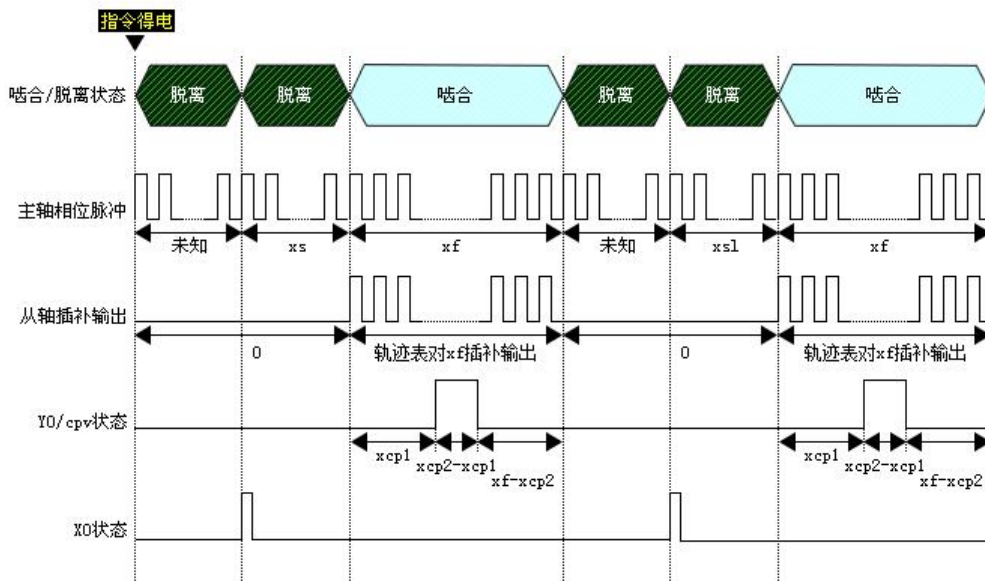
- 组合 B:
 - ◆ 指令生效后, X0 为 ON 则啮合。
 - ◆ 周期脱离。



- 组合 C:
 - ◆ 指令生效后，主轴走前置啮合位移量后则啮合。
 - ◆ 周期脱离。

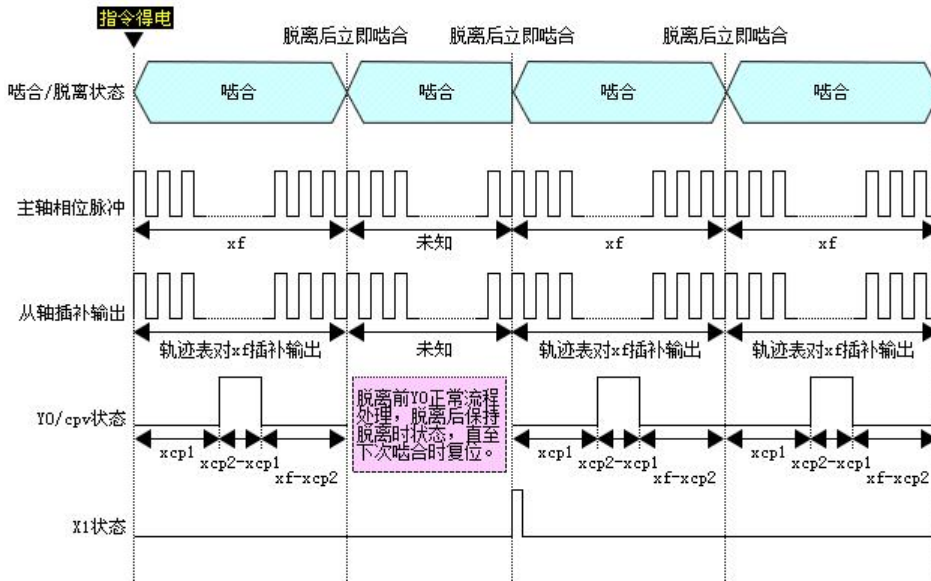


- 组合 D:
 - ◆ 指令生效后，等待 X0 触发为 ON，主轴再走前置啮合位移量后则啮合。
 - ◆ 周期脱离。



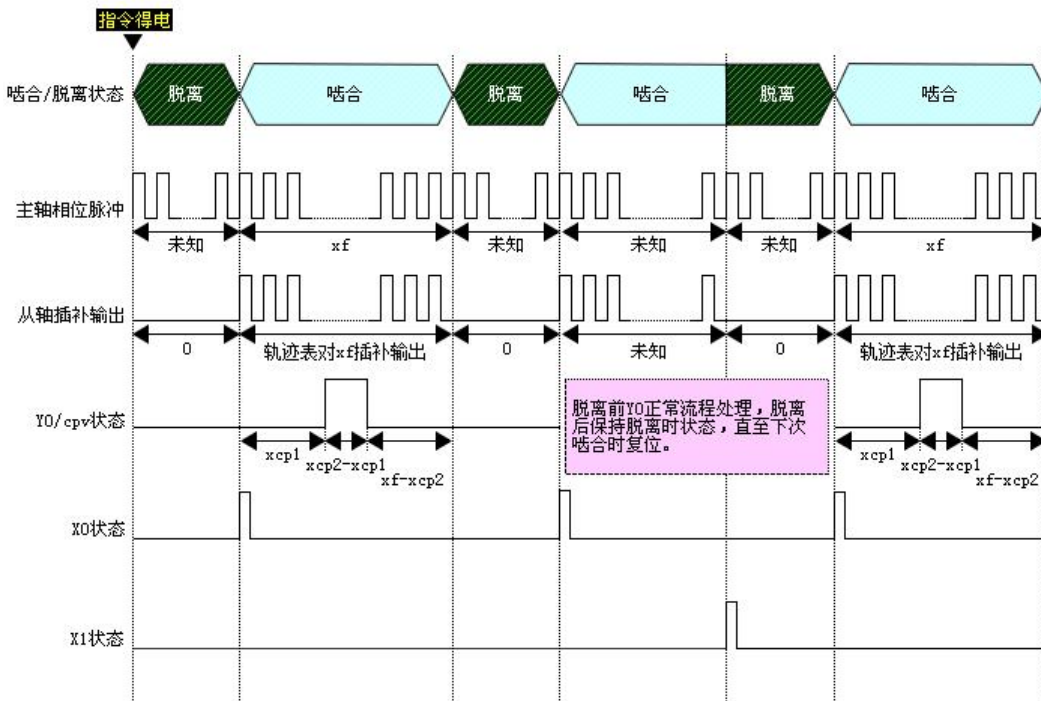
● 组合 E:

- ◆ 指令生效则啮合。
- ◆ X1 触发为 ON 或周期结束，凸轮脱离。



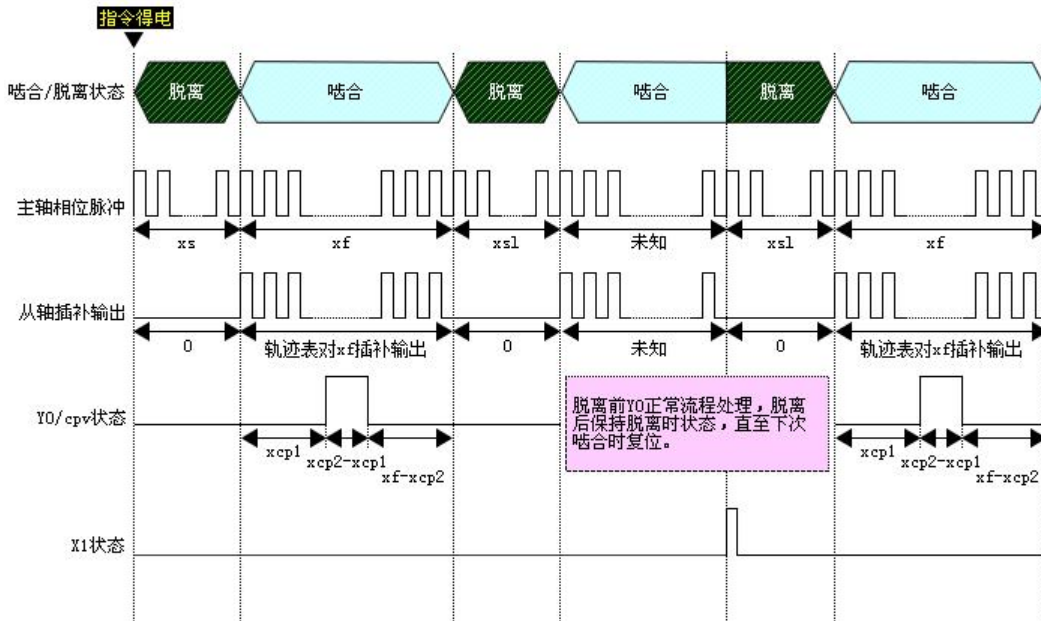
● 组合 F:

- ◆ 指令生效后, X0 为 ON 则啮合。
- ◆ X1 触发为 ON 或周期结束，凸轮脱离。



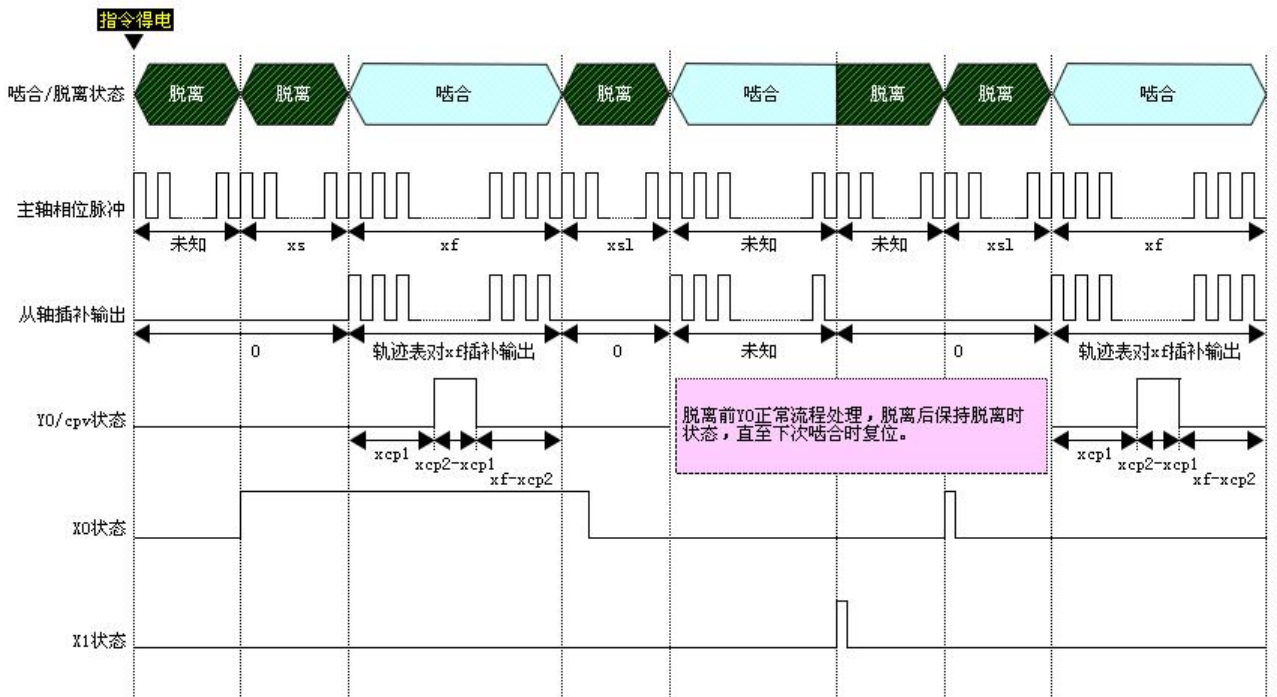
● 组合 G:

- ◆ 指令生效后，主轴走前置啮合位移量后则啮合。
- ◆ X1 触发为 ON 或周期结束，凸轮脱离。

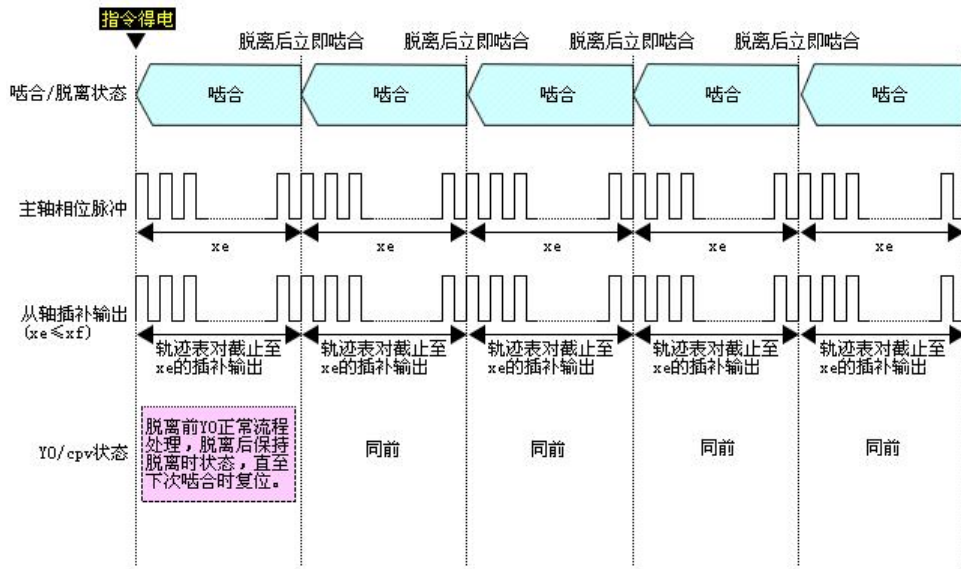


● 组合 H:

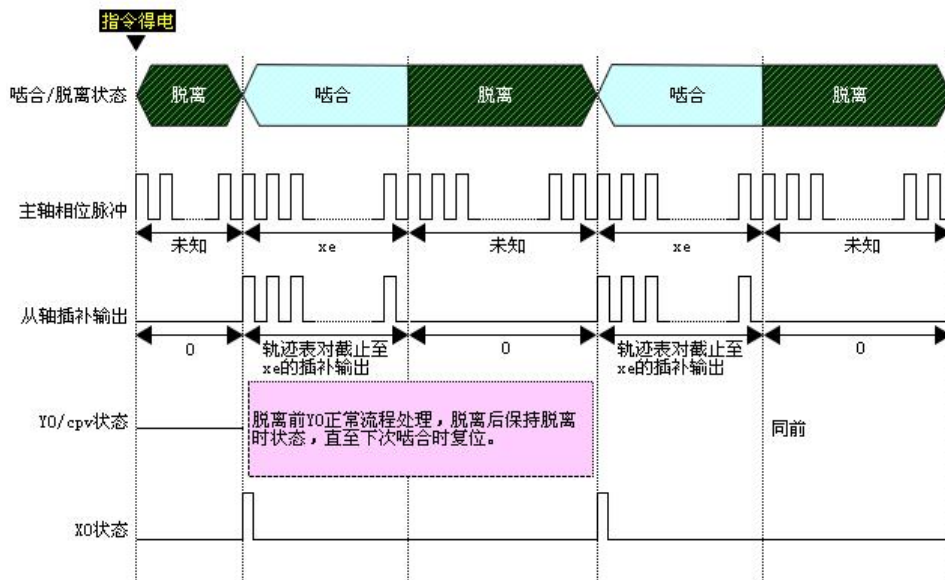
- ◆ 指令生效后，等待 X0 触发为 ON，主轴再走前置啮合位移量后则啮合。
- ◆ X1 触发为 ON 或周期结束，凸轮脱离。



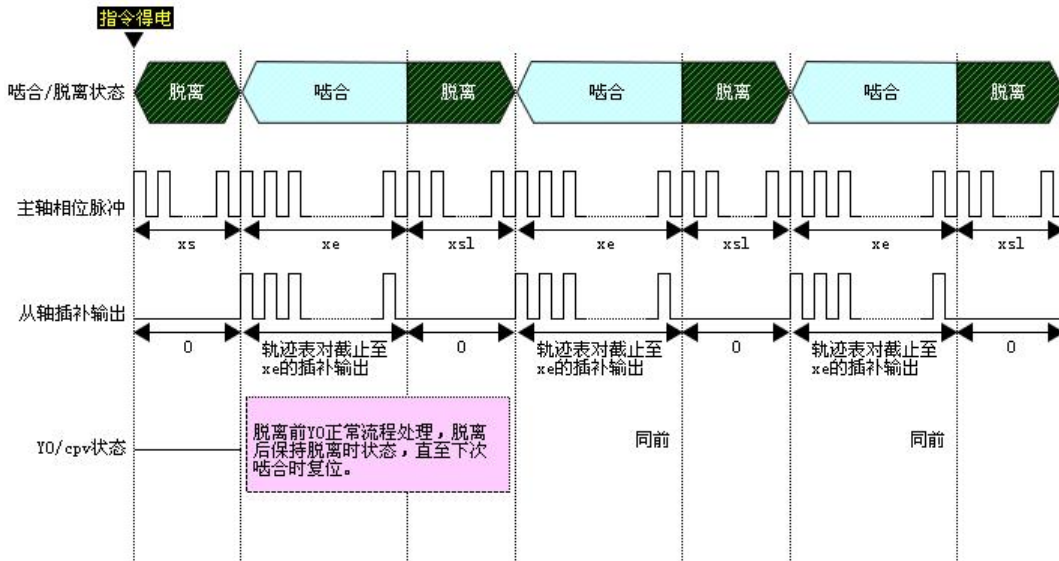
- 组合 I:
 - ◆ 指令生效则啮合。
 - ◆ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



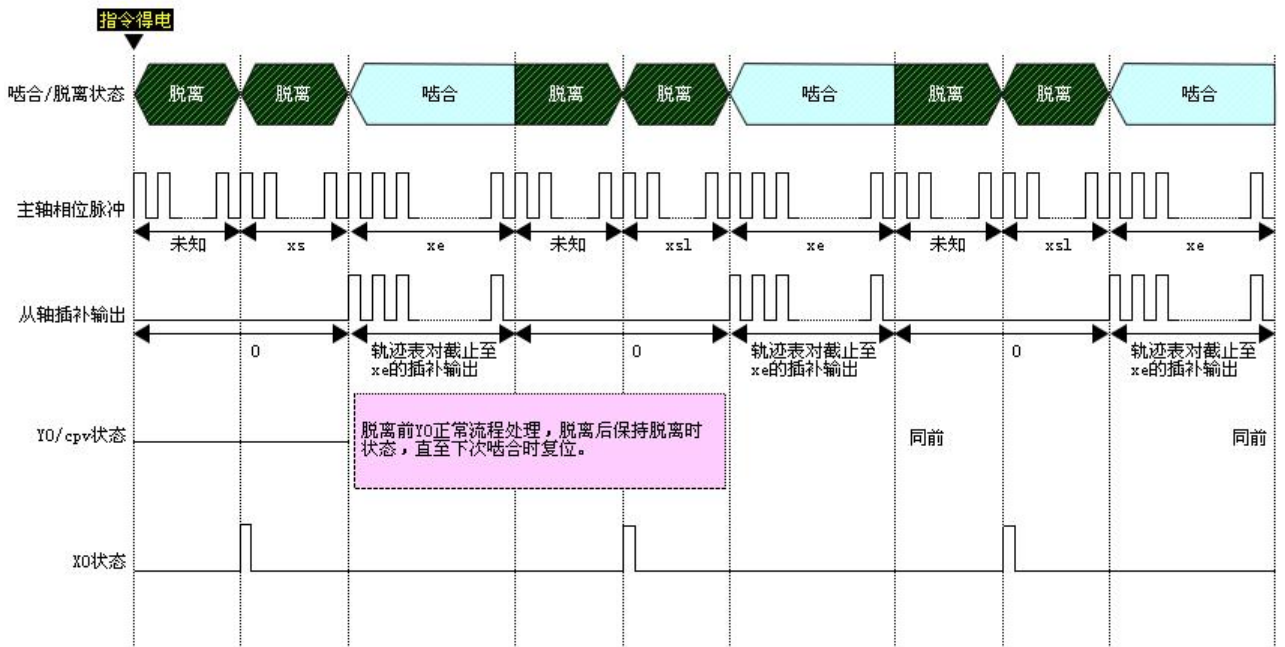
- 组合 J:
 - ◆ 指令生效后，X0 为 ON 则啮合。
 - ◆ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



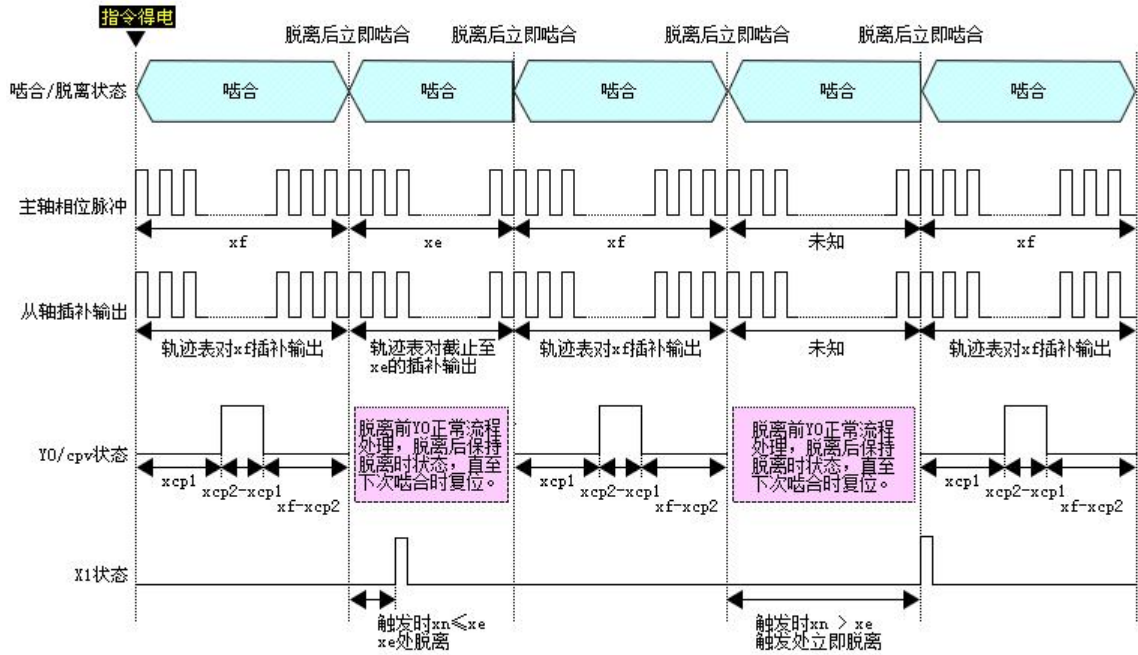
- 组合 K:
 - ◆ 指令生效后，主轴走前置啮合位移量后则啮合。
 - ◆ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



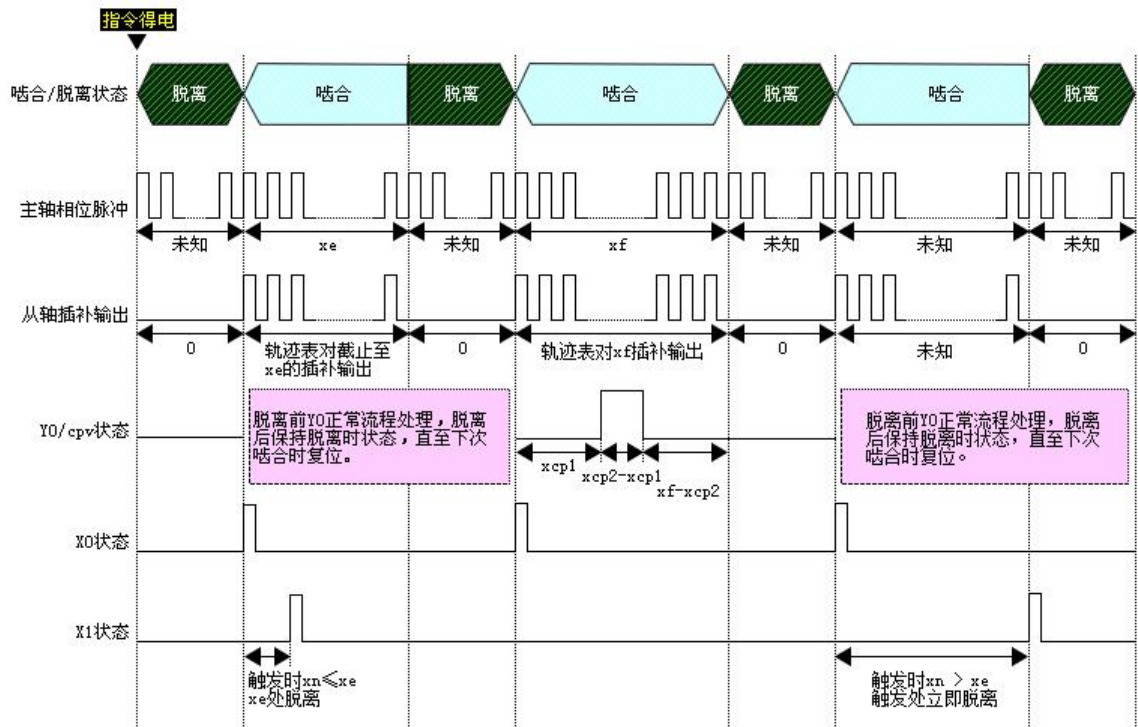
- 组合 L:
 - ◆ 指令生效后，等待 X0 触发为 ON，主轴再走前置啮合位移量后则啮合。
 - ◆ 主轴相位到达主轴相位脱离设置值，凸轮脱离。



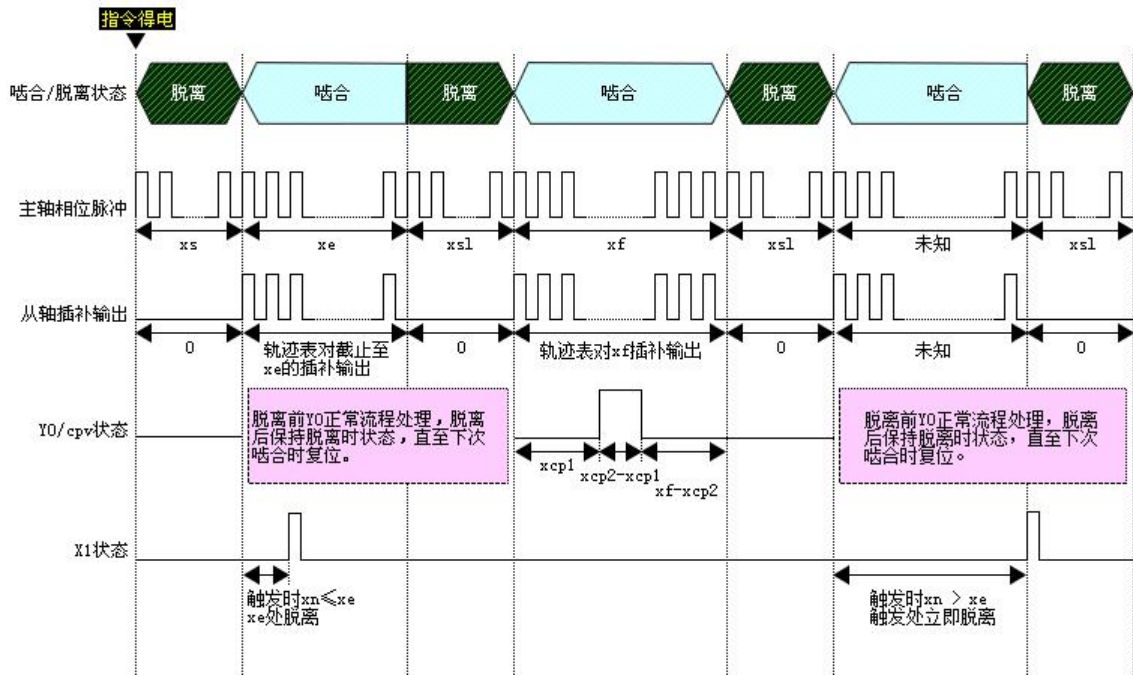
- 组合 M:
 - ◆ 指令生效则啮合。
 - ◆ X1 触发为 ON 后，到达主轴相位脱离点脱离。



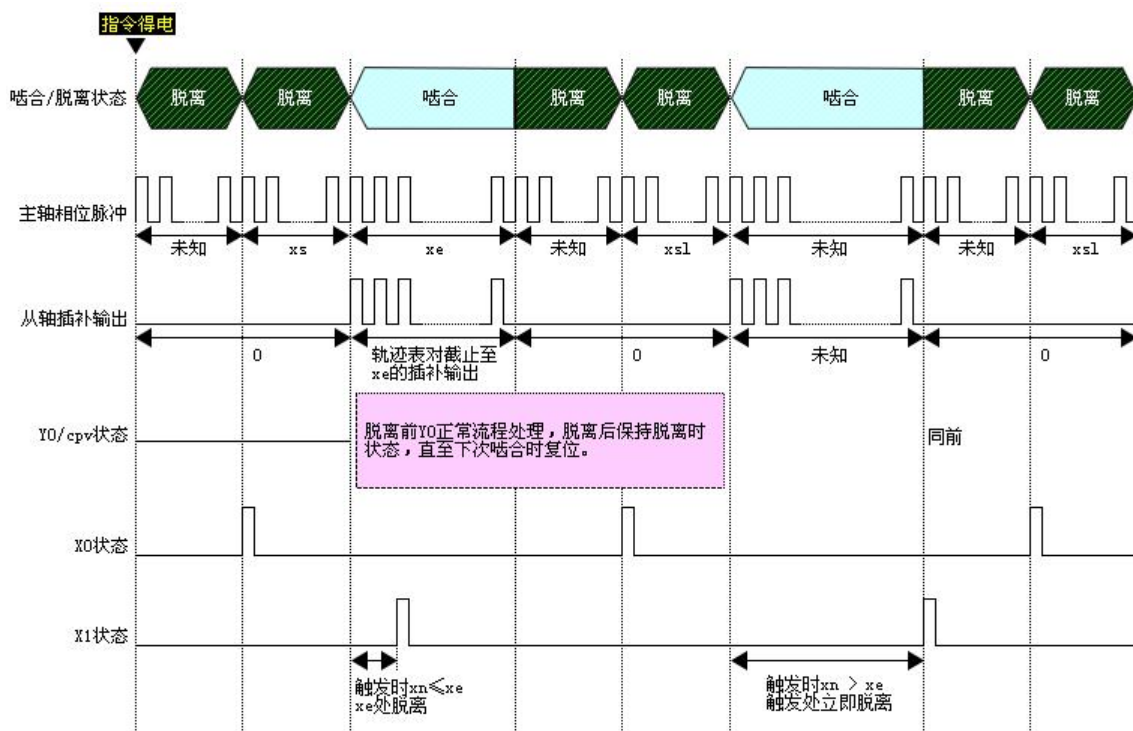
- 组合 N:
 - ◆ 指令生效，X0 为 ON 则啮合。
 - ◆ X1 触发为 ON 后，到达主轴相位脱离点脱离。



- 组合 0:
 - ◆ 指令生效后，主轴走前置啮合位移量后则啮合。
 - ◆ X1 触发为 ON 后，到达主轴相位脱离点脱离。



- 组合 P:
 - ◆ 指令生效后，等待 X0 触发为 ON，主轴再走前置啮合位移量后则啮合。
 - ◆ X1 触发为 ON 后，到达主轴相位脱离点脱离。



追剪/滚切
模式分析

常见追剪/滚切系统通常分为定长和色标两种模式，以下是模式分析。

定长	色标
啮合/脱离方式	
<p>可选立即啮合，若需要前置偏移量，则选择前置啮合。根据实际工况设定初始前置啮合位移和周期前置啮合位移。</p> <p>根据设定的轨迹表曲线量程与工件的加工长度设计脱离方式。若选择触发脱离，其触发点设定为从轴近点或零点信号。</p> <p>由于轨迹表设计起始点一般为零点处，故必须保证啮合位置为从轴的原点处。</p>	<p>色标信号设定为触发啮合输入，若需要前置偏移量，可选触发+前置啮合。根据实际工况设定初始前置啮合位移和周期前置啮合位移。</p> <p>根据设定的轨迹表曲线量程与工件的加工长度设计脱离方式。若选择触发脱离，其触发点设定为从轴近点或零点信号。</p> <p>由于轨迹表设计起始点一般为零点处，故必须保证啮合位置为从轴的原点处。</p>
主轴实际量程(凸轮周期)	
<p>若运行过程中无脱离阶段(周期脱离后立即啮合，可视为主从轴始终啮合)，则应设计主轴周期长度为物料切割长度，以满足工艺周期循环的需要。</p> <p>若设计脱离阶段，相位脱离位移和周期前置啮合位移应相互配合，满足两者之和等于物料的切割长度。</p>	<p>一般接入色标以后，从轴会有脱离等待阶段，设计主轴周期长度应≤物料切割长度。</p> <p>脱离等待阶段的脉冲数等于物料切割长度对应主轴脉冲数减去主轴周期长度脉冲数。</p>
运行精度	
<p>由于主轴相位仅仅通过位置的反馈确定，故长时间运行可能导致主/从轴产生一定的相位偏差，此时应通过调整相位偏差补偿参数消除误差。</p>	<p>每个凸轮周期均有触发啮合输入(色标检测)的过程，故主从轴相位偏差在每个周期开始处将自动校正1次。一般不需要补偿，精度高。</p>

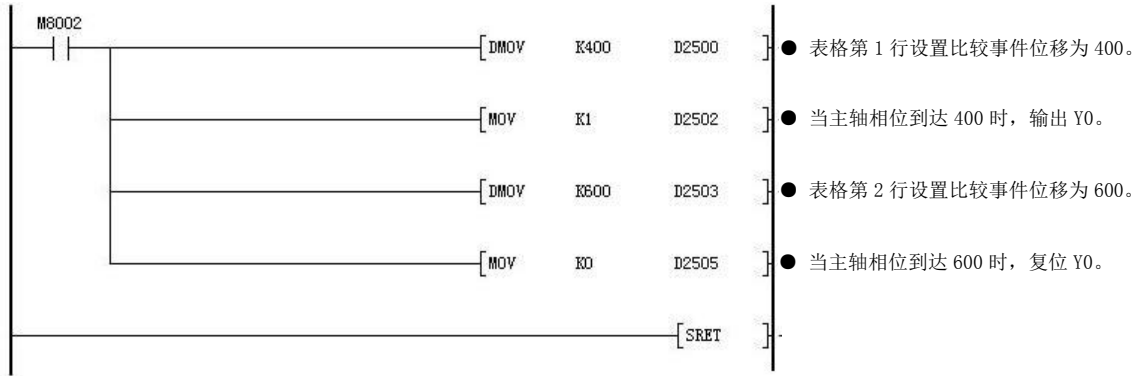
凸轮应用
举例

假定凸轮运行参数如下设置：

- 主轴实际量程：1000
- 初始前置啮合位移：650
- 周期前置啮合位移：150
- 相位脱离位移：850
- 表格比较输出行 1 主轴相位：400
- 表格比较输出行 2 主轴相位：600
- 比较事件输出端口：Y0，极性为 0。
- 输入触发啮合端口：X0，极性为 0。
- 输入触发脱离端口：X1，极性为 0。
- 啮合/脱离方式选择组合 ‘ P ’

设置凸轮输出参数的梯形图示例如下：





编程注意事项

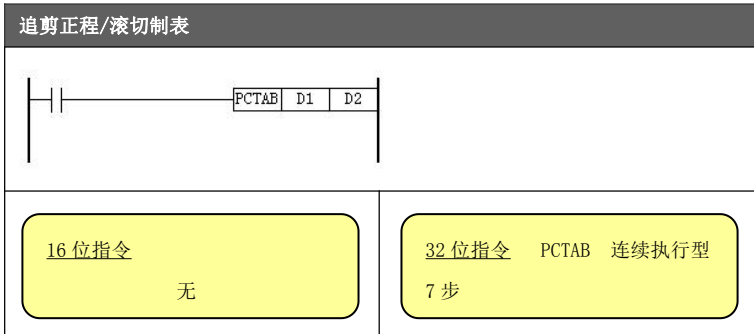
指令执行异常的情况有(指令不动作或动作错误):

- 指定一个不存在的软件虚拟输出轴或该轴未绑定具体脉冲输出口。
- 设置轨迹表起始单元寄存器序号为非偶数地址。
- 设置主轴实际值单元寄存器序号为非偶数地址。
- 啮合/脱离相关设置与轨迹表设计相悖。比如轨迹表起始点规划为原点，但啮合点位置不在原点。
- 当脱离方式选择为触发脱离，且脱离时从轴速度较快，可能导致无法在该 X 端口处脱离，会略微过冲最大不超过当前速度 1 个 ms 的位移。
- 轴速度设置不合理导致轴参数 **轴错误代码** 报错。
- 对应轴参数 **轴状态** 不为 0 的情况下执行指令。

上电 PLC 运行时 Y0~Y5 端口为梯形图控制，若配置为运动指令占用且指令执行(脉冲口/方向/事件输出)，则对应 Y 软元件不再关联实际端口。

8.4.3 追剪正程/滚切制表[PCTAB]

制作追剪正程/滚切的轨迹曲线表。



操作数	作用	类型
D1	曲线参数配置缓存区起始单元	32 位 BIN
D2	轨迹表格起始单元	32 位 BIN

操作数	字型软元件									
D1 (占 16 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							
D2 (占 n 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z
		●	●							

功能与动作

- D1 指定追剪正程/滚切曲线的参数配置缓存区起始寄存器单元，建议为偶数地址。指令执行时，D1 指向寄存器后的连续 16 个寄存器均被占用，这些寄存器应在指令执行前配置好且只配置 1 次 (配置方法后述)，指令执行过程中这些寄存器将作为计算缓存区使用，指令导通条件断开这些寄存器将被释放。
例：D1 填写 D1000，则 D1000~D1015 在指令执行期间均被占用。
- D2 指定总曲线轨迹表格的起始寄存器单元，建议为偶数地址。
D2 指向的寄存器单元 (轨迹表点数)，与 D2+2 指向的寄存器单元 (轨迹表初始位置值) 均在执行指令时由系统初始化，用户可以无需配置。

无论是追剪还是滚切的运动模型，都可以理解为电子凸轮的一种应用。追剪的一个工艺流程周期包括正程 (等待→加速追剪→同步剪切→减速停止) 和返程组成。若不考虑返程流程，则与滚切的工作原理基本一致。故追剪正程/滚切制表指令适用于此类应用。

追剪、滚切 流程异同	追剪		滚切	
	相同点			
<ul style="list-style-type: none"> ● 应用于定长/色标切割，根据物料材质决定选择追剪还是滚切。 ● 主轴为送料轴(一般保持匀速)，从轴控制追剪轴或滚切轴。 ● 在切割时必须与主轴速度保持一致。 				
不同点				
送料轴保持匀速单向送料，刀架伺服电机在切割台上做往复运动。在同步区域内与送料轴速度达到同步，同时完成切割动作，然后反向回到原点，再次追踪同步切割。		滚切轴为圆周形状，始终往同一方向旋转做圆周运动，利用轴面凸出刀口部分接触物料的时间点(同步区域)时进行剪切，期间送料轴可不停断持续等速送料。		
需要控制从轴方向，剪切完毕后回程换向。		无需控制从轴方向，始终同向运动。		
切刀需要程序控制，进入同步区后控制切刀动作，执行切割动作。		切刀安装在凸轮辊轴表面凸出部分，每旋转一周刀口与物料接触一次，切刀无需程序干预。		
物料切割时间长，材质较硬，如木板，管材等。		物料切割时间短，轻薄材料，如纸张，塑料袋等。		

追剪正程/滚切的运动区域可划分为同步区和非同步区：



- 同步区完成工艺操作，需与主轴速度保持一致。比如在切刀接触到物料时一定要与物料速度同步，如果接触时切刀速度大于同步速度，出现对物料一个向前牵扯的力，会照成物料切面不平；如果速度低于物料速度，会出现堵料的现象。
- 非同步区主要保障运动控制器运行平稳，以最小的加速度，最小的变速范围为设计条件。根据实际工艺情况，非同步区又可由加减速区和等待区组合而成。

同步区的规划会影响到实际设备的运行，一个工艺周期中若同步区越大，加减速的时间就越小，表示设备需要在短时间内进行加减速，对于电机、机台、切刀的冲击都很大，而且容易导致伺服过流报警，设备无法正常运行。

一个追剪正程/滚切轨迹点表通常情况下由五段曲线组成，生成的曲线型值点位置变化规律是这些曲线的走向反映。

曲线段	作用区域
起始停止段	等待区，根据配置参数决定是否在该区域。
起始加速/减速段	加速/减速区，根据配置参数决定是否为加速或减速。
匀速段	同步区，与主轴速度一致。
终止加速/减速段	减速/加速区，根据配置参数决定是否为减速或加速。
终止停止段	等待区，根据配置参数决定是否在该区域。

轨迹表定义

同通用制表指令不同，追剪/滚切正程制表指令可以直接生成整个凸轮曲线(不包含追剪返程曲线)，而不需要执行多次制表生成。由用户指定表头寄存器地址(操作数 D2)。表内型值点数据系统根据曲线配置参数自动计算，需留足表格数据存放的寄存器区间。

曲线轨迹表内容	内容解释	寄存器示例
表头 - (指定偶地址)	无需用户初始化。	D5000(L)
型值点总数(轨迹表量程)	调用指令后由系统填 1，计算完成后系统再次更新。	
初始位置点	无需用户初始化。调用指令后系统填 0。	D5002(L)
曲线型值点 1	指令计算生成。	D5004(L)
...		假定 500 个点
曲线型值点 n		D6002(L)
表格总点数和表内型值点的数据类型均为 32 位，占用 2 个寄存器。		
表格占用总寄存器个数 = (表头)2 + 型值点总数 * 2		

曲线参数定义

曲线配置参数缓存区的定义如下表所示。
地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

D1+偏移	名称	参数描述	单位
D1+0(L)	主轴周期位移	指定一个工艺周期内主轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 > 0。	脉冲
D1+2(L)	主轴同步区位移	指定一个工艺周期内主轴同步区所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 > 0 且 ≤ 主轴周期位移。	脉冲
D1+4(L)	从轴周期位移	指定一个工艺周期内从轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 > 0。	脉冲
D1+6(L)	从轴同步区位移	指定一个工艺周期内从轴同步区所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 > 0 且 ≤ 从轴周期位移。	脉冲
D1+8	周期最小时间	指定追剪正程/滚切曲线 1 个凸轮周期的最小时间。 $周期最小时间 \div 表格时间间隔 = 曲线的型值点点数$ ● 作为追剪正程曲线规划，周期最小时间为主轴最大速情况下正程的追击、同步、脱离时间之和，返程时间在追剪返程制表中设定。 ● 作为滚切曲线规划，周期最小时间为主轴最大速情况下整个凸轮周期的时间。主轴最大速度由参数主轴最高限速确认，系统计算得出。 ● 设置该参数 > 0。	ms
D1+9	表格时间间隔	指定追剪正程/滚切曲线的表格时间间隔。 间隔要选取合适的范围，过大则降低曲线细分精度，过小则会造成曲线生成点数过多，使得寄存器开销变大。 ● 设置该参数 > 0，且能被周期最小时间整除。	ms

D1+偏移	名称	参数描述	单位
D1+10	从轴最大速度	从轴运行最大速度。 例如设置速度为 400 KHz，即每 ms 发送 400 个脉冲， 假定其计量单位 = 1 脉冲/ms，则从轴最大速度设置 400。 计量单位参考参数时间当量和脉冲当量描述。 ● 设置该参数 > 0。	脉冲当量 / 时间当量
D1+11	从轴最大加减速	从轴运行最大加/减速度。 例如设置加速度为 1 脉冲/ms ² ， 假定其计量单位 = 0.1 脉冲/ms ² ，则从轴最大加减速设置 10。 计量单位参考参数时间当量和脉冲当量描述。 ● 设置该参数 > 0。	脉冲当量 / 时间当量 ²
D1+12	时间当量	曲线时间当量，与脉冲当量耦合计算，为速度/加减速度的计量单位之一。	ms
D1+13	脉冲当量	曲线脉冲当量，与时间当量耦合计算，为速度/加减速度的计量单位之一。 例如时间当量 = 10，脉冲当量 = 10，则速度单位 = 10 ÷ 10 = 1 脉冲/ms， 加减速度单位 = 10 ÷ 10 ² = 0.1 脉冲/ms ² 。	脉冲
D1+14	制表状态	制表指令当前状态，系统同一时间只允许 1 条制表指令处于运行状态。 ● = 0：当前无制表指令运行，可执行制表指令 ● = 16：正在生成曲线数据(系统每 1ms 生成 1 个型值点数据) ● = 32：曲线数据生成完毕 ● = 64：曲线生成错误 当检测到制表状态等于 32 时，曲线数据已生成完毕，可断开制表指令。梯形图扫描到指令断开后，自动将制表状态清 0。 ● 该参数无需设置，为显示值。	-
D1+15	主轴最高限速	主轴最高限速，制表完成后由系统更新。 根据曲线设置参数不同，系统求得的主轴最高速也不相同。当工艺速度小于主轴最高限速时，满足设计要求。 ● 对于追剪制表而言，正程制表生成的主轴最高限速为中间值，执行返程制表指令后的主轴最高限速将会再次更新。 ● 该参数无需设置，为显示值。	脉冲当量 / 时间当量

主轴最高限速换算

对于追剪/滚切系统，主轴(机台)的速度一般而言以“米/分”为单位。其规定的最大限速需与制表完成后系统生成的主轴最高限速数值进行比较，若该速度大于主轴最高限速，则需要重新调整曲线参数以满足设计要求。

但主轴最高限速的单位是(脉冲当量/时间当量)，与实际物理量关联需要进行量纲转换。
示例：假定系统主轴的最大速度为 30 米/分，脉冲当量为 10 脉冲/mm。

那么主轴最大速度 = 30 * 1000 / (60 * 1000) = 0.5 mm/ms，根据脉冲当量得出主轴最大速度为 5 脉冲/ms。

假定脉冲当量 = 10 脉冲，时间当量 = 10 ms，则(脉冲当量/时间当量)为 1 脉冲/ms。

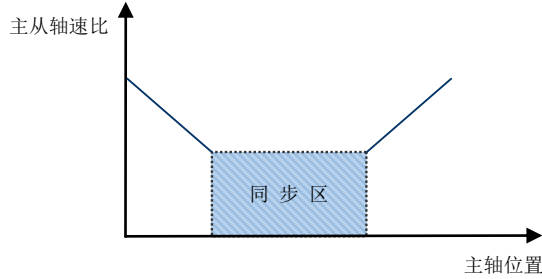
那么主轴最大速度 = 5 (脉冲当量/时间当量)。

凸轮曲线轨迹分析

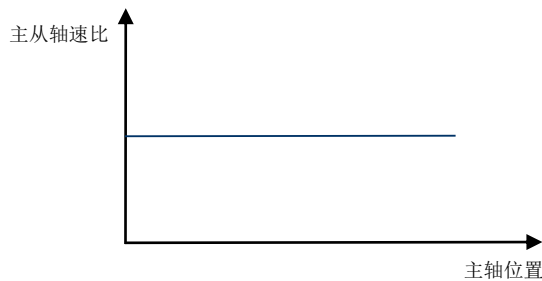
主从轴周期位移量和同步区位移量的设置决定了凸轮曲线的输出方式。

假定从轴同步区位移 ÷ 从轴周期位移 = $K_{从}$ ，主轴同步区位移 ÷ 主轴周期位移 = $K_{主}$ ，
 以下是 $K_{从}$ 与 $K_{主}$ 在不同比例关系下的凸轮曲线轨迹。

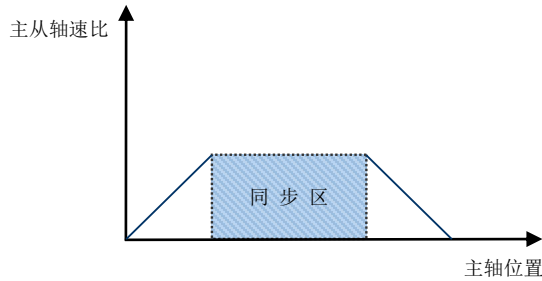
- $K_{从} < K_{主}$ ：在同步区从轴速度与主轴速度同步，过了同步区后，为了赶上下一个凸轮周期，则从轴加速。



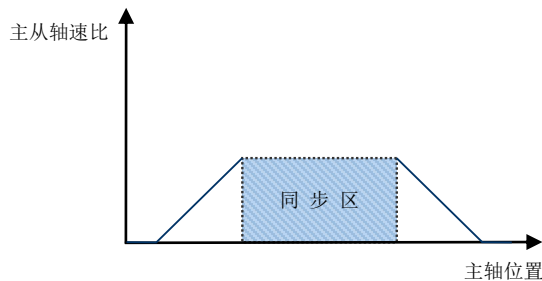
- $K_{从} = K_{主}$ ：从轴与主轴同速比运动。



- $2 \text{倍} K_{主} > K_{从} > K_{主}$ ：同步区结束后从轴减速，在下一个凸轮周期再加速到同步区。



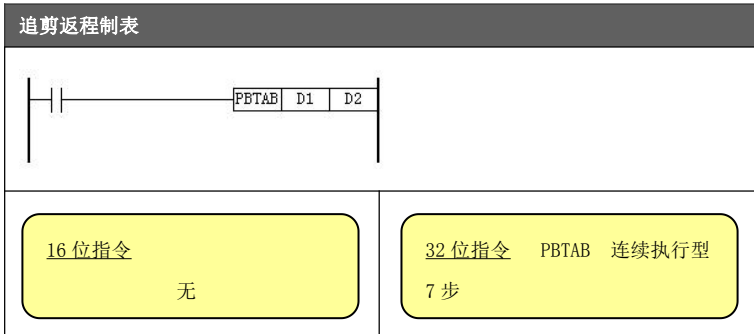
- $K_{从} > 2 \text{倍} K_{主}$ ：最常见的一种情况。在一个凸轮周期中，同步区结束后从轴减速到停止，等待一定主轴脉冲数过去后，再加速到同步区。



* 仅当 $K_{从} > 2 \text{倍} K_{主}$ 时，适用于追剪工艺的正程制表。因为接续追剪返程轨迹表时，从轴速度必须为零速（即处于等待区）。

8.4.4 追剪返程制表[PBTAB]

制作追剪返程的轨迹曲线表。



操作数	作用	类型
D1	曲线参数配置缓存区起始单元	32 位 BIN
D2	轨迹表格起始单元	32 位 BIN

操作数	字型软元件										
D1 (占 12 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								
D2 (占 n 点)	K/H	D	R	C	T	KX	KY	KM	KS	V/Z	
		●	●								

功能与动作

- **D1** 指定追剪返程曲线的参数配置缓存区起始寄存器单元，建议为偶数地址。
指令执行时，**D1** 指向寄存器后的连续 12 个寄存器均被占用，这些寄存器应在指令执行前配置好且只配置 1 次 (配置方法后述)，指令执行过程中这些寄存器将作为计算缓存区使用，指令导通条件断开这些寄存器将被释放。
例：**D1** 填写 D1000，则 D1000~D1011 在指令执行期间均被占用。
- **D2** 指定总曲线轨迹表格的起始寄存器单元，建议为偶数地址。
该起始寄存器单元接续追剪正程表的点数和位置，故不需要对 **D2** 指向的寄存器单元进行初始化 (轨迹表点数)，也不需要初始化 **D2+2** 指向的寄存器单元 (轨迹表初始位置值)。

对于直线追剪系统，其主轴周期长度 = 追剪正程量程 + 追剪返程量程。追剪正程轨迹表由指令 PCTAB 完成 (参考章节 11.4.3)。在执行此指令之前，需确保已完成 PCTAB 指令的制表工作。

一个追剪返程轨迹点表通常情况下由三段曲线组成，生成的曲线型值点位置变化规律是这些曲线的走向反映。

曲线段	作用区域
返程加速段	追剪返程区域的加速段回程，加速到返程最高速进入返程匀速段。
返程匀速段	追剪返程区域的匀速段回程。
返程减速段	追剪返程区域的减速段回程，从返程最高速减速到零速。

* 追剪返程曲线生成轨迹为四次多项式加减速计算方案。

轨迹表定义

追剪返程制表指令直接接续追剪正程轨迹表进行返程轨迹的制表，执行一次制表即可生成。由用户指定表头寄存器地址(操作数 D2)。表内型值点数据系统根据曲线配置参数自动计算，需留足表格数据存放的寄存器区间。

曲线轨迹表内容	内容解释	寄存器示例
表头 - (指定偶地址) 型值点总数(轨迹表量程)	与追剪正程轨迹表指向同一个单元。执行制表指令前不需要初始化。 指令执行完毕由系统更新(即在原有正程制表的轨迹表点数基础上累加返程轨迹表的点数)。	D5000(L)
曲线型值点 1	假定正程已占用 500 个点，则返程型值点生成地址从 D6004 开始。	D6004(L)
...		假定 500 个点
曲线型值点 n	指令计算生成。	D7002(L)
表格总点数和表内型值点的数据类型均为 32 位，占用 2 个寄存器。 表格占用总寄存器个数 = (表头)2 + 正程型值点总数 * 2 + 返程型值点总数 * 2		

曲线参数定义

曲线配置参数缓存区的定义如下表所示。
地址后带(L)标识的参数为 32 位参数，占用 2 个 D 寄存器。

D1+偏移	名称	参数描述	单位
D1+0(L)	主轴回程位移	指定回程期间内主轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数 > 0。	脉冲
D1+2(L)	从轴回程位移	指定回程期间内从轴所走的脉冲数。 ● 该参数为 32 位数值，占用 2 个 D 寄存器单元。 ● 设置该参数为追剪正程制表中参数从轴周期位移的负值。 例如追剪正程制表中参数从轴周期位移 = 60000，则该参数设置-60000。	脉冲
D1+4	返程最小时间	指定追剪返程曲线完成返程的最小时间。 返程最小时间 ÷ 表格时间间隔 = 返程曲线的型值点点数。 ● 设置该参数 > 0。	ms
D1+5	表格时间间隔	指定追剪返程曲线的表格时间间隔。 ● 设置该参数与正程制表中参数表格时间间隔一致，且能被返程最小时间整除。 例如追剪正程制表中参数表格时间间隔 = 1，则该参数设置 1。	ms
D1+6	从轴最大速度	从轴返程最大速度。 ● 设置该参数 < 0。	脉冲当量 / 时间当量
D1+7	从轴最大加减速	从轴返程最大加/减速度。 ● 设置该参数 < 0。	脉冲当量 / 时间当量 ²

D1+偏移	名称	参数描述	单位
D1+8	时间当量	返程曲线时间当量。 ● 设置该参数为追剪正程制表中参数时间当量一致。 例如追剪正程制表中参数时间当量 = 10，则该参数设置 10。	ms
D1+9	脉冲当量	返程曲线脉冲当量。 ● 设置该参数为追剪正程制表中参数脉冲当量一致。 例如追剪正程制表中参数脉冲当量 = 10，则该参数设置 10。	脉冲
D1+10	制表状态	制表指令当前状态，系统同一时间只允许 1 条制表指令处于运行状态。 ● = 0：当前无制表指令运行，可执行制表指令 ● = 16：正在生成曲线数据(系统每 1ms 生成 1 个型值点数据) ● = 32：曲线数据生成完毕 ● = 64：曲线生成错误 当检测到制表状态等于 32 时，曲线数据已生成完毕，可断开制表指令。梯形图扫描到指令断开后，自动将制表状态清 0。 ● 该参数无需设置，为显示值。	-
D1+11	主轴最高限速	主轴最高限速，制表指令执行前填写追剪正程制表后生成的参数 <u>主轴最高限速</u> 数值。制表完成后系统将再次更新该数值。 根据曲线设置参数不同，系统求得的主轴最高速也不相同。当工艺速度小于主轴最高限速时，满足设计要求。 ● 追剪正程制表生成的 <u>主轴最高限速</u> 为中间值，执行返程制表指令后生成的 <u>主轴最高限速</u> 再与 <u>主轴最高限速</u> 进行比较。 ● 该参数在指令执行前需设置初始值，指令执行后为显示值。	<u>脉冲当量</u> / <u>时间当量</u>

主轴最高限速换算

对于追剪/滚切系统，主轴(机台)的速度一般而言以“米/分”为单位。其规定的最大限速需与制表完成后系统生成的主轴最高限速数值进行比较，若该速度大于主轴最高限速，则需要重新调整曲线参数以满足设计要求。

但主轴最高限速的单位是(脉冲当量/时间当量)，与实际物理量关联需要进行量纲转换。

示例：假定系统主轴的最大速度为 30 米/分，脉冲当量为 10 脉冲/mm。

那么主轴最大速度 = $30 * 1000 / (60 * 1000) = 0.5 \text{ mm/ms}$ ，根据脉冲当量得出主轴最大速度为 5 脉冲/ms。

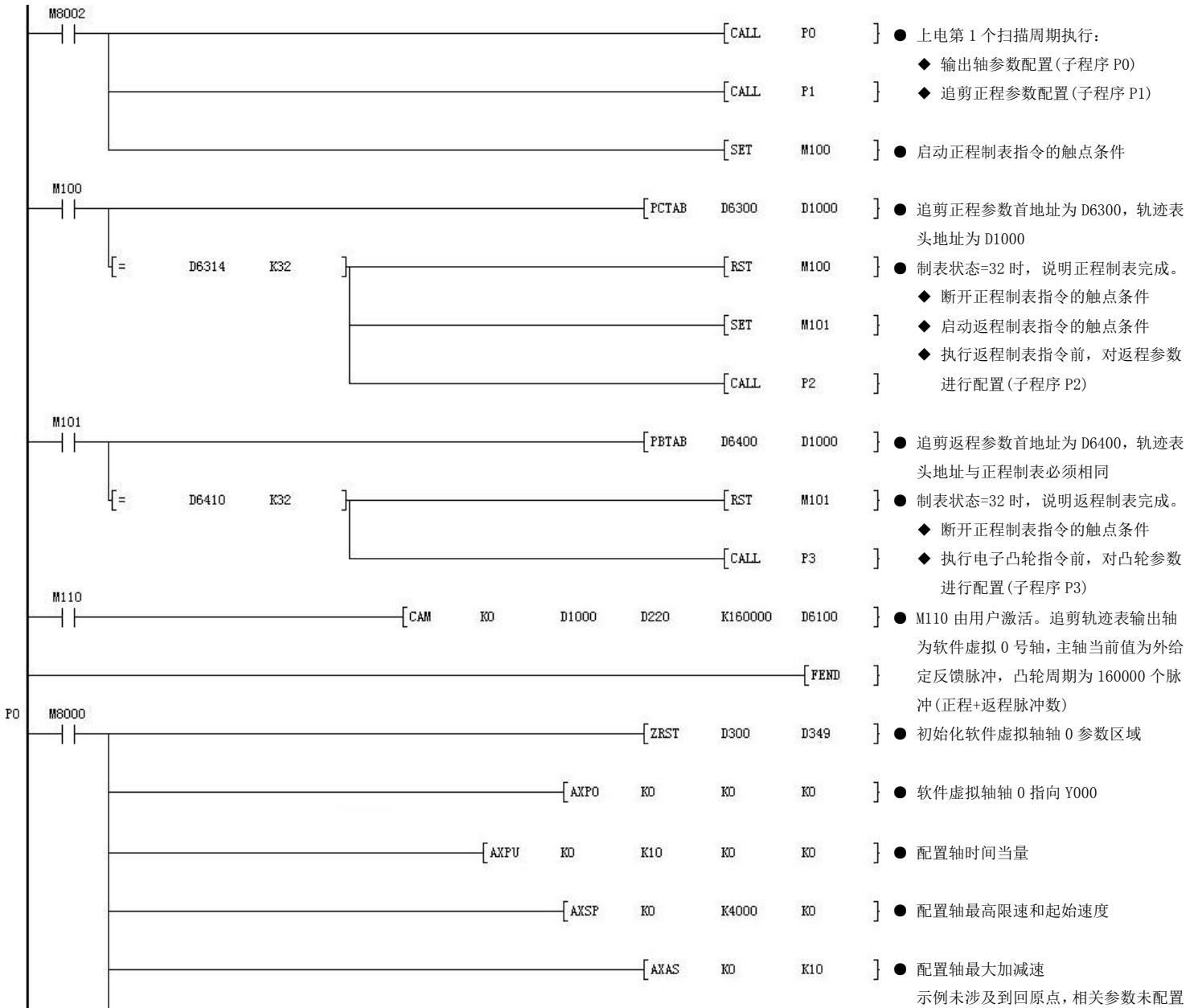
假定脉冲当量 = 10 脉冲，时间当量 = 10 ms，则(脉冲当量/时间当量)为 1 脉冲/ms。

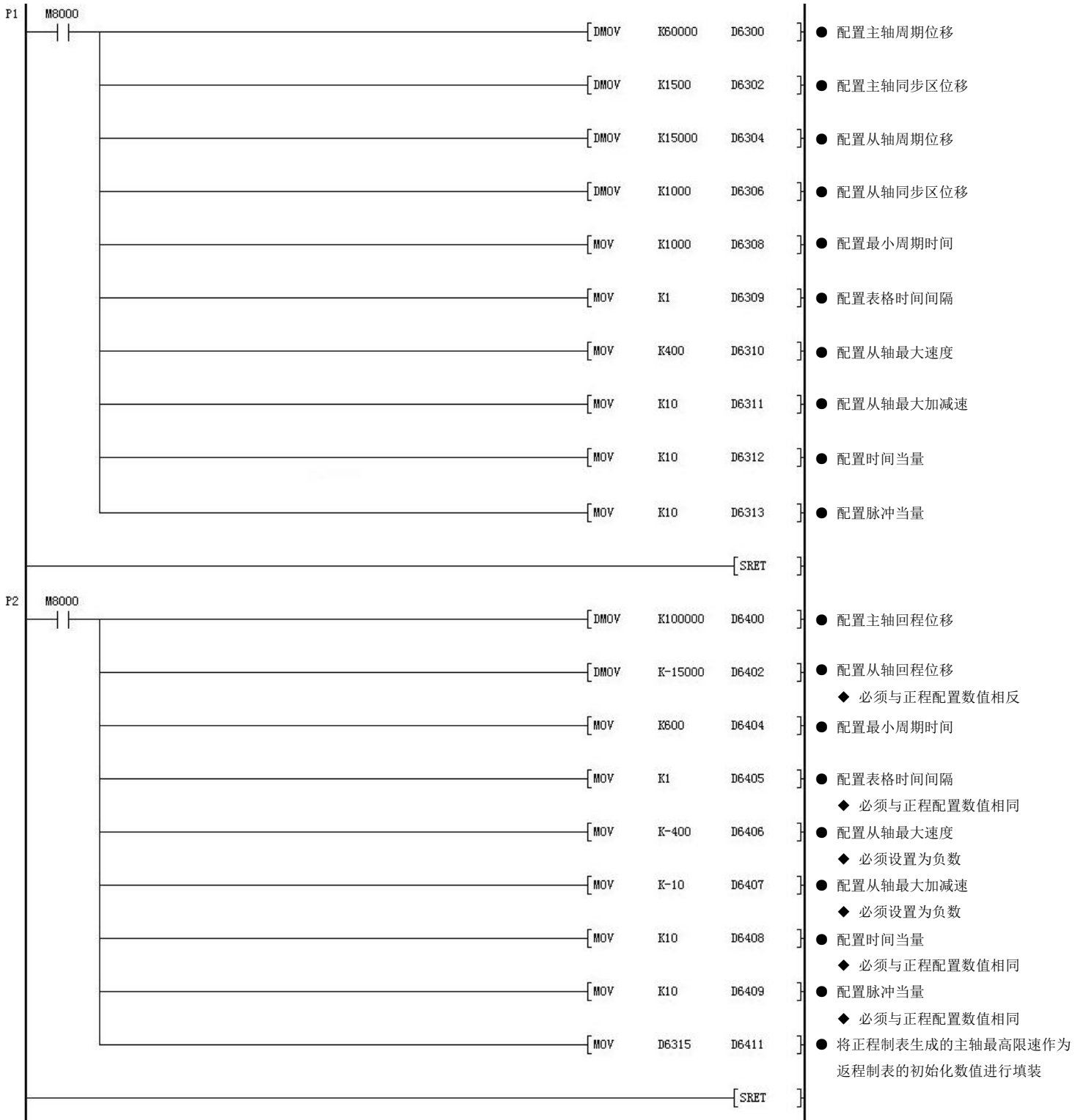
那么主轴最大速度 = 5 (脉冲当量/时间当量)。

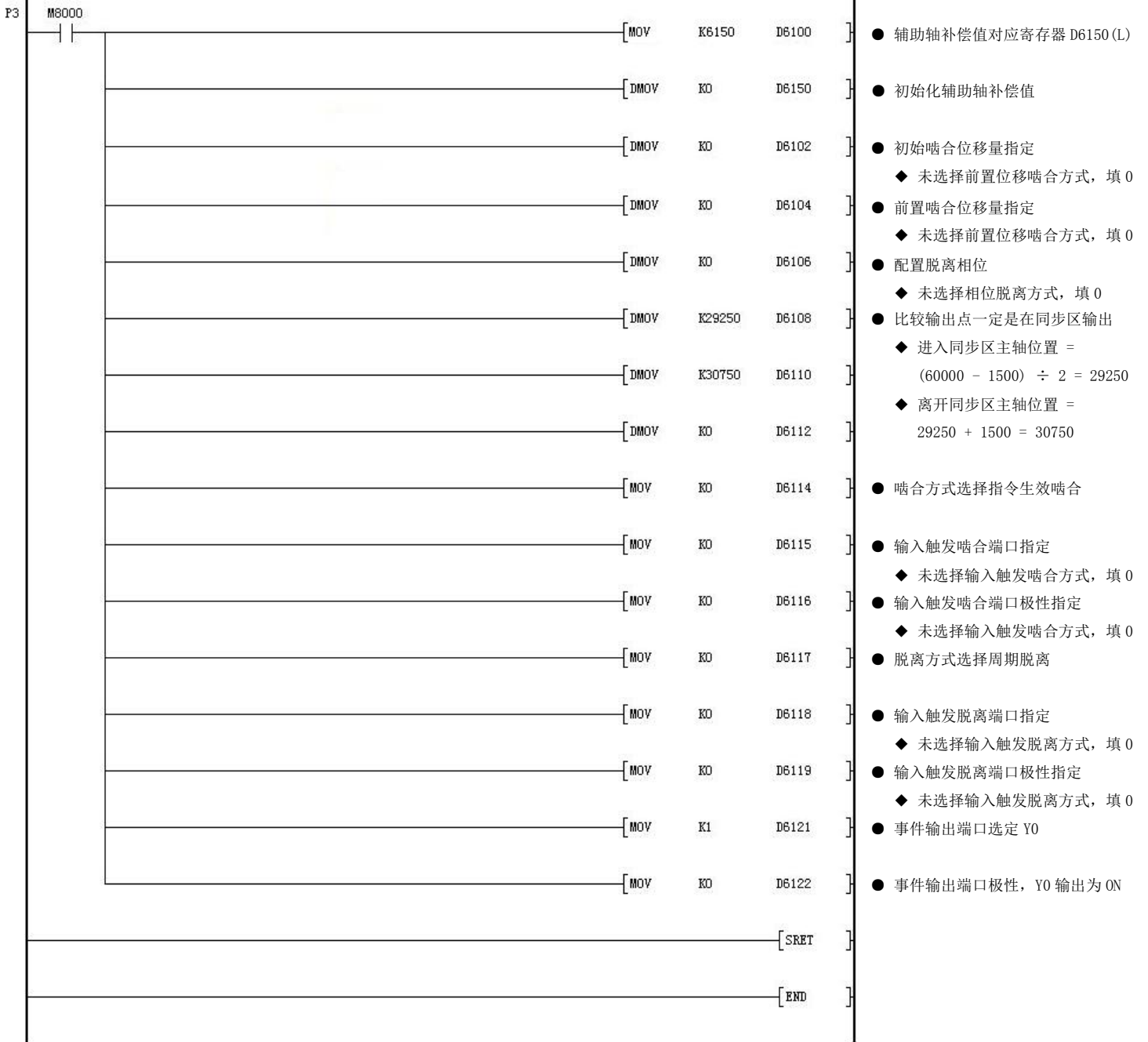
追剪应用举例

完整的追剪工艺实现梯形图示例，包含以下内容：

- 输出轴轴参数配置(输出轴为运动控制器本体，对应软件虚拟轴 0)。
- 追剪正程制表(正程制表梯形图可单独作为追剪正程/滚切制表的示例)。
- 追剪返程制表。
- 轨迹表凸轮输出。
 - ◆ 主轴当前值 D220(L) 假定为外给定脉冲数。主轴实际量程为追剪正程+返程的主轴脉冲数之和。
 - ◆ 示例规划凸轮运行无脱离阶段。啮合方式为指令生效啮合，脱离方式为周期脱离，该模式下前置啮合位移量和脱离相位均可设置为 0，无需指定相关触发端口。







使用方法

- 下载梯形图后 PLC 运行, 自动生成追剪轨迹表数据。
- 置位 M110 后追剪曲线输出。

追剪曲线 轨迹分析

追剪正程制表主从轴参数:

- 主轴周期位移 = 60000 脉冲
- 主轴同步区位移 = 1500 脉冲
- 从轴周期位移 = 15000 脉冲
- 从轴同步区位移 = 1000 脉冲

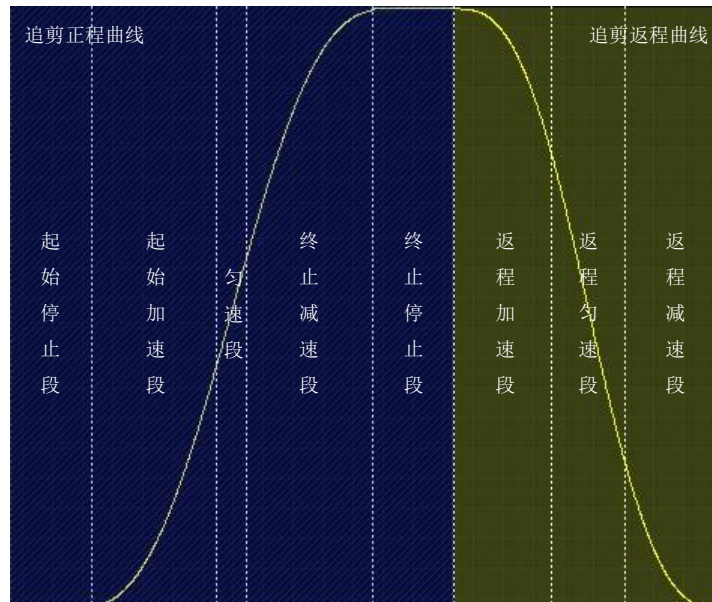
假定从轴同步区位移 ÷ 从轴周期位移 = $K_{从}$ ，主轴同步区位移 ÷ 主轴周期位移 = $K_{主}$ ，
则计算得 $K_{从} = 1000 \div 15000 = 0.067$ ， $K_{主} = 1500 \div 60000 = 0.025$ 。

故 $K_{从} > 2$ 倍 $K_{主}$ ，正程曲线应包含起始等待段、起始加速段、匀速段、终止减速段、
终止停止段这 5 个阶段。

$K_{从}$ 与 $K_{主}$ 的比例关系对应不同的凸轮曲线轨迹，参考章节 10.5.3 描述。

追剪返程输出曲线为四次多项式曲线，分为返程加速段、返程匀速段和返程减速段。

以下是一个完整追剪周期，示例梯形图运行 CAM 指令后，从轴位移变化实测图：



追剪流程从轴位移曲线示意图