COTRUST

Magicworks PLC 指令库

手册

手册版本 V0.3
发布日期 09/2021

权利声明

版权声明

商标声明

 ®、TrustPLC®、MagicWorks®、COTRUST®、MagiCampus®、COMOTION®、  ®均为深圳市合信自动化技术有限公司或其关联公司的注册商标，受法律保护，侵权必究。未经深圳市合信自动化技术有限公司或其关联公司的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、永久下载、修改、传播、抄录或与其他产品捆绑销售，或注册为其域名或无线网址名称，或域名或无线网址的主要部分。

免责声明

本手册依据现有信息制作，其内容如有更改，恕不另行通知。深圳市合信自动化技术有限公司在编写该手册的时候已尽最大努力保证其内容准确可靠，但深圳市合信自动化技术有限公司不对本手册中的遗漏、不准确或印刷错误导致的损失和损害承担责任。

前 言

使用我公司 Magicworks PLC 软件中相关库之前，请您仔细阅读本手册，以便更清楚地掌握每个库的使用要求以及特性。

服务支持

深圳市合信自动化技术有限公司建立了售后维修服务中心，并提供电话热线服务。客户在产品使用过程中遇到问题时，可随时与深圳市合信自动化技术有限公司各地的服务支持联系。各地服务支持热线电话请到 http://www.co-trust.com/Company/Contact/index.html 获取。此外，客户还可通过深圳市合信自动化技术有限公司网站 http://www.co-trust.com 或官方微信及时了解最新产品动态，以及下载需要的技术文档。

版本修订记录

发行日期	修订后版本	修订内容
2021 年 9 月	V0.3	**编程软件 Magicworks PLC V2.22 指令详解** • 首版发布

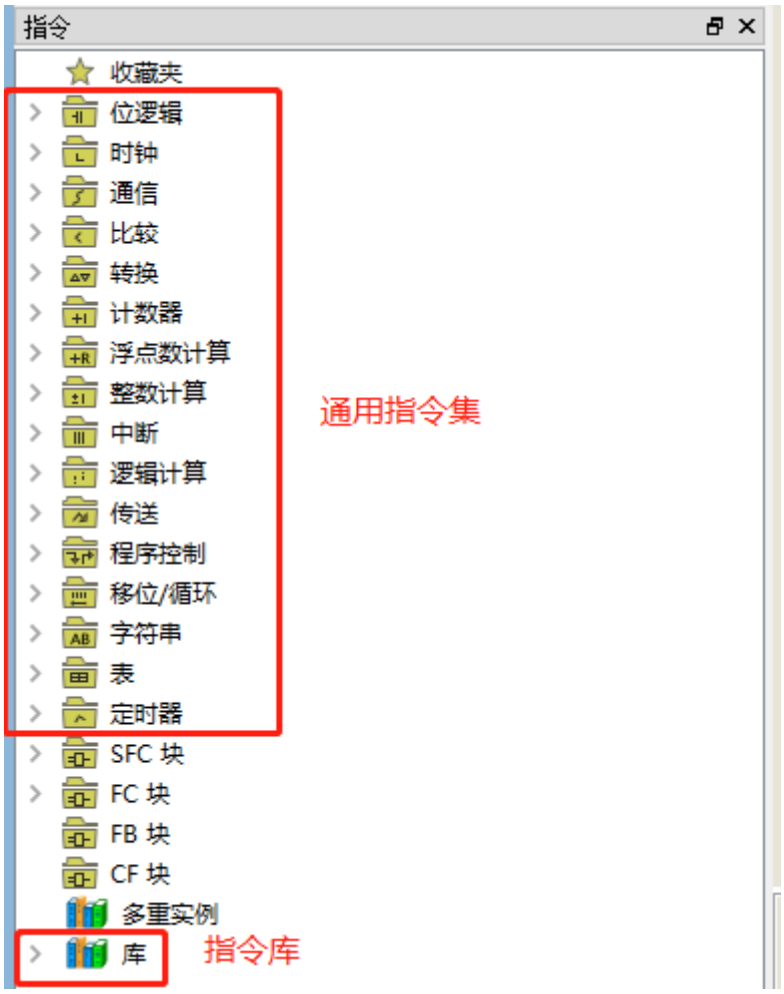
目 录

概述

1

本章主要对 MagicWorks PLC 指令集进行描述。

Magicworks PLC 指令主要由通用指令集和指令库组成。



通用指令集

通用指令集包含位逻辑、时钟、通信、比较、转换、计数器、浮点计数器、整数计算、中断、逻辑计算、传送、程序控制、移位/循环、字符串、表、定时器等指令。

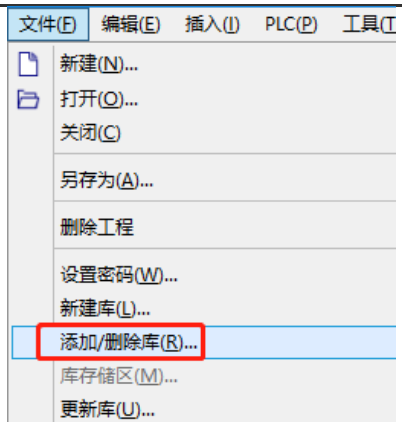
指令库

指令库主要分为通信指令库、运控库、称重库、扩展数据库、高速计数指令库、脉冲指令库。

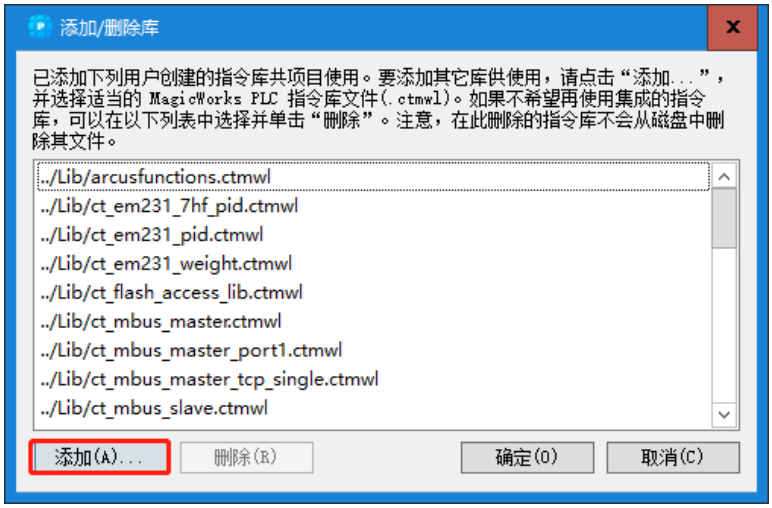
1.1 添加库文件

库文件的添加步骤如下

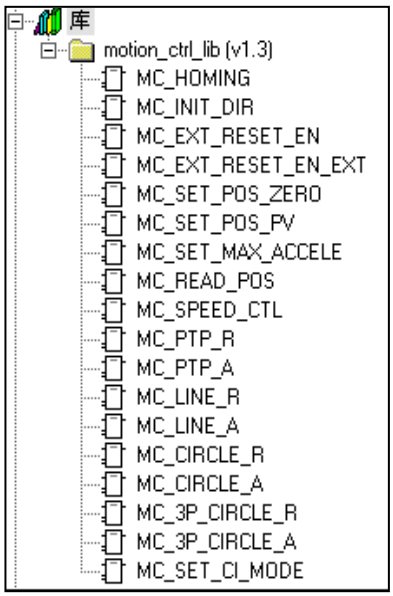
选择 MagicWorks PLC 的菜单项“文件”→“添加/删除库”，如下图所示：



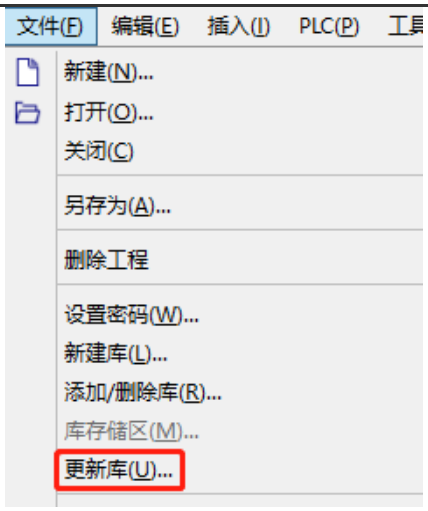
点击“添加(A)...”，浏览至该库文件的保存目录。选择需添加的库文件文件，并点击“确认”。



库添加成功之后，查看指令树中的库节点，即可看到库（例如 motion_ctrl_lib(v1.3).mwl）已经被添加到指令树中。



另外，用户也可选择在编程软件里在线更新库，选择“文件”→“更新库”，即可看到库文件自动更新情况。



1.2 PLC 型号集合

CTH200 系列 PLC	
基本型	
型号	订货号
H224	CTH2 214-1BA33-0X24
H226L	CTH2 216-1BA33-0X40
标准型	
型号	订货号
H224	CTH2 214-1AD33-0X24
	CTH2 214-1BD33-0X24
H226L	CTH2 216-2AD33-0X40
	CTH2 216-2BD33-0X40
H226M	CTH 216-1AD33-0X24
	CTH 216-1BD33-0X24
老平台高性能型	
型号	订货号
H224X	CTH2 214-1AX33-0X24
	CTH2 214-1BX33-0X24
H226XL	CTH2 216-2AX33-0X40
	CTH2 216-2BX33-0X40
H228XL	CTH2 218-3BX33-0X60
V5 高性能型升级版	
型号	订货号
H224X	CTH2 214-1AX35-0X24
	CTH2 214-1BX35-0X24
H226XM	CTH2 216-1AX35-0X24
	CTH2 216-1BX35-0X24
H226XL	CTH2 216-2AX35-0X40
	CTH2 216-2BX35-0X40
I6 高性能型升级版	
型号	订货号

H226IM	CTH2 216-1AD46-0X24
	CTH2 216-1BD46-0X24
H226IL	CTH2 216-2AD46-0X40
	CTH2 216-2BD46-0X40
H226IH	CTH2 216-1AH46-2B24

CTSC200 系列 PLC	
V5 平台	
型号	**订货号**
224+	CTS7 214-1AD35-0X24
	CTS7 214-1BD35-0X24
224I	CTS7 214-1AD45-0X24
	CTS7 214-1BD45-0X24
224XP	CTS7 214-2AD45-0324
	CTS7 214-2BD45-0324
226I	CTS7 216-2AD45-0X40
	CTS7 216-2BD45-0X40
226M	CTS7 216-1AD35-0X24
	CTS7 216-1BD35-0X24
226L	CTS7 216-2AD35-0X40
	CTS7 216-2BD35-0X40
226H	CTS7 216-1AH35-0B24
	CTS7 216-1AH35-2B24
老平台	
型号	**订货号**
224E	CTS7 214-1AE33-0X24
224+	CTS7 214-1AD33-0X24
	CTS7 214-1BD33-0X24
224I	CTS7 214-1AD41-0X24
	CTS7 214-1BD41-0X24
224XP	CTS7 214-2AD41-0324
	CTS7 214-2BD41-0324
226I	CTS7 216-2AD41-0X40
	CTS7 216-2BD41-0X40
226M	CTS7 216-1BD33-0X24
	CTS7 216-1AD33-0X24
226L	CTS7 216-2AD33-0X40
	CTS7 216-2BD33-0X40
226M-CAN	CTS7 216-1AC33-0X24
226H	CTS7 216-1AH34-0B24
	CTS7 216-1AH34-1B24
	CTS7 216-1AH34-2B24

CTH300-H 系列 PLC	
型号	订货号
H35-00	CTH3 H35-000S1
H36-01	CTH3 H36-001S2
H31-00	CTH3 H31-000S1
H32-01	CTH3 H32-001S2
H52-10	CTH3 H52-100S2
H56-10	CTH3 H56-100S2

CTMC 系列 PLC	
型号	订货号
M228IL	CTMC 218-3AI35-0X60
M228ML	CTMC 218-3AM35-0X60
M228SL	CTMC 218-3AS35-0X60

通用指令集

2

本章主要对 MagicWorks PLC 指令集进行描述。

2.1 通用指令一览表

类别	STL	指令名称
位逻辑指令	LD	常开触点载入指令
	A	常开触点与指令
	O	常开触点或指令
	LDN	常闭触点载入指令
	AN	常闭触点与指令
	ON	常闭触点或指令
	LDI	常开触点立即指令
	AI	常开触点与立即指令
	OI	常开触点或立即指令
	LDNI	常闭触点立即指令
	ANI	常闭触点与立即指令
	ONI	常闭触点或立即指令
	NOT	逻辑栈顶取反指令
	EU	上升沿检测指令
	ED	下降沿检测指令
	ALD	与载入指令
	OLD	或载入指令
	LPS	逻辑进栈指令
	LDS	载入堆栈指令
	LRD	逻辑读取指令
	LPP	逻辑出栈指令
	=	输出指令
	=I	立即输出指令
	S	置位指令
	SI	立即置位指令
	R	重设指令
	RI	立即重设指令
	AENO	功率流与指令
	NOP	空操作指令

类别	STL	指令名称
比较指令	LDBx	字节比较（=,<,>,<=,>=,<>）载入指令
	LDWx	整数比较（=,<,>,<=,>=,<>）载入指令
	LDDx	长整数比较（=,<,>,<=,>=,<>）载入指令
	LDRx	浮点数比较（=,<,>,<=,>=,<>）载入指令
	ABx	字节操作数比较（=,<,>,<=,>=,<>）与指令
	AWx	整数比较（=,<,>，<=,>=,<>）与指令
	ADx	双整数比较（=,<,>，<=,>=,<>）与指令
	ARx	浮点数比较（=,<,>，<=,>=,<>）与指令
	OBx	字节操作数比较（=，<,>,<=,>=,<>）或指令
	OWx	整数比较（=,<,>,<=，>=,<>）或指令
	ODx	双整数比较（=,<,>，<=,>=,<>）或指令
	ORx	浮点数比较（=,<,>，<=,>=,<>）或指令
	LDSx	字符串比较（=,<>）载入指令
	ASx	字符串比较（=,<>）与指令
	OSx	字符串比较（=,<>）或指令
传送指令	MOVB	字节移动指令
	MOVW	字移动指令
	MOVD	双字移动指令
	MOVR	浮点数移动指令
	BMB	块移动字节指令
	BMW	块移动字指令
	BMD	块移动双字指令
	SWAP	高低字节交换指令
	BIR	移动字节立即读指令
	BIW	移动字节立即写指令
	SRCP	存储配方到存储卡
	LRCP	从存储卡装载配方
	SDL	存储数据记录到存储卡
整数计算指令	+I	整数加法指令
	-I	整数减法指令
	*I	整数乘法指令
	/I	整数除法指令
	+D	双整数加法指令
	-D	双整数减法指令
	*D	双整数乘法指令
	/D	双整数除法指令
	MUL	整数与双整数相乘
	DIV	整数与双整数相除

类别	STL	指令名称
整数计算指令	INCB	字节递增指令
	DECB	字节递减指令
	INCW	整数递增指令
	DECW	整数递减指令
	INCD	长整数递增指令
	DECD	长整数递减指令
浮点数运算令	+R	实数加法指令
	-R	实数加法指令
	*R	实数乘法指令
	/R	实数除法指令
	SQRT	平方根指令
	SIN	正弦运算指令
	COS	余弦运算指令
	TAN	正切运算指令
	LN	自然对数运算指令
	EXP	自然指数运算指令
	PID	PID 环路运算指令
转换指令	BTI	字节至整数转换
	ITB	整数至字节转换
	ITD	整数至双整数转换
	ITS	整数转换为字符串
	DTI	双整数至整数转换
	DTR	双整数至实数转换
	DTS	双整数至字符串
	ROUND	进位取整指令
	TRUNC	截位取整指令
	RTS	实数至字符串转换
	BCDI	BCD 至整数转换
	IBCD	整数至 BCD 转换
	ITA	整数至 ASCII 转换
	DTA	双整数至 ASCII 转换
	RTA	实数至 ASTII 码转换指令
	ATH	ASCII 至16进制转换
	HTA	16进制至 ASCII 转换
	STI	字符串至整数转换
	STD	字符串至双整数转换
	STR	字符串至实数转换
	DECO	解码指令
	ENCO	编码指令
	SEG	段指令

类别	STL	指令名称
实时时钟	TODR	读取实时时钟
	TODW	设置实时时钟
	TODRX	读取扩展的实时时钟
	TODWX	设置扩展的实时时钟
移位循环指令	SLB	向左移位字节
	SLW	向左移位字
	SLD	向左移位双字
	SRB	向右移位字节
	SRW	向右移位字
	SRD	向右移位双字
	RLB	向左旋转字节
	RLW	向左旋转字
	RLD	向左旋转双字
	RRB	向右旋转字节
	RRW	向右旋转字
	RRD	向右旋转双字
	SHRB	移位寄存器位指令
逻辑计算指令	INVB	反转字节指令
	INVW	反转字指令
	INVD	反转双字指令
	ANDB	与字节指令
	ANDW	与字指令
	ANDD	与双字指令
	ORB	或字节指令
	ORW	或字指令
	ORD	或双字指令
	XORB	异或字节指令
	XORW	异或字指令
	XORD	异或双字指令
计数器指令	CTU	向上计数指令
	CTD	向下计数指令
	CTUD	向上/向下计数指令
	HDEF	高速计数器定义
	HSC	高速计数器指令
	PLS	脉冲输出指令
	PTO	脉冲输出指令
定时器指令	TON	打开延时计时器
	TONR	保留性打开延时计时器
	TOF	关闭延时计时器
	BITIM	读取开始时间计时器
	R_UTIM	读取当前微秒值指令
	CITIM	计算间隔时间计时器

类别	STL	指令名称
字符串指令	SLEN	取字符串长度指令
	SCPY	复制字符串指令
	SSCPY	从字符串复制子字符串指令
	SCAT	并置字符串指令
	SFND	在字符串中查找字符串指令
	CFND	在字符串中查找一个字符指令
表指令	FILL	内存填充指令
	ATT	增加至表格指令
	FNDx	表格查找（=,<,>,<=，>=,<>）指令
	FIFO	先入先出指令
	LIFO	后入先出指令
中断指令	CRETI	从中断程序有条件返回指令
	ENI	启用中断指令
	DISI	禁用中断指令
	ATCH	附加中断指令
	DTCH	分离中断指令
	CEVNT	清除中断事件指令
程序控制指令	FOR	FOR-NEXT 循环开始
	NEXT	FOR-NEXT 循环结束
	JMP	跳转至标签指令
	LBL	标签指令
	WDR	监视器重设指令
	DLED	诊断 LED 指令
	CALL	调用子例行程序指令
	LSCR	载入顺序控制中继
	SCRE	顺序控制中继结束
	SCRT	顺序控制中继转换
	CSCRE	有条件顺序控制中继结束
	CRET	从子程序有条件返回指令
	END	有条件结束
	STOP	停机指令
通信指令	CANW	CANopen SDO 写指令
	CANR	CANopen SDO 读指令
	XMT	自由口传输指令
	RCV	自由口接收指令
	NETR	网络读取指令
	NETW	网络写入指令
	UDPNETR	以太网网络读功能
	UDPNETW	以太网网络写入功能
	MBSNDMSG	发送 Modbus 消息
	SPA	设置端口指令
	GPA	获得端口地址指令
	EBUSR	读取模块信息指令

	EBUSW	写入模块信息指令
	EBUSGETDIA	获取模块内部诊断信息指令
	EBUSSNDCMD	发送模块操作命令指令

2.2 通用指令详解

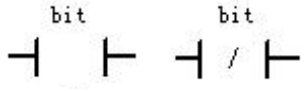
位逻辑指令

标准触点指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

```
LD      bit
A       bit
O       bit
LDN     bit
AN      bit
ON      bit
```

梯形图：



输入输出参数	数据类型	适用操作数
位	BOOL	I, Q, M, SM, T, C, V, S, L

操作数说明：如果参数 bit 的数据类型为 I 或 Q，标准触点指令从内存或过程映象寄存器读取引用值。

功能说明：

当参数位变量 bit 为 1 时，常开触点（LD、A、O）接通。

当参数位变量 bit 为 0 时，常闭触点（LDN、AN、ON）接通。

在 STL 中，常开触点用载入（LD）、与（A）、或（O）指令表示，常开指令将地址位数值置于栈顶。

常闭触点由负装载（LDN）、与非（AN）、或非（ON）指令表示，常闭指令将地址位数值的逻辑非置于栈顶。

在 LAD 中，常开和常闭指令用触点表示。

示例程序：

STL：

ORGANIZATION_BLOCK 主程序：OB1

Network 1

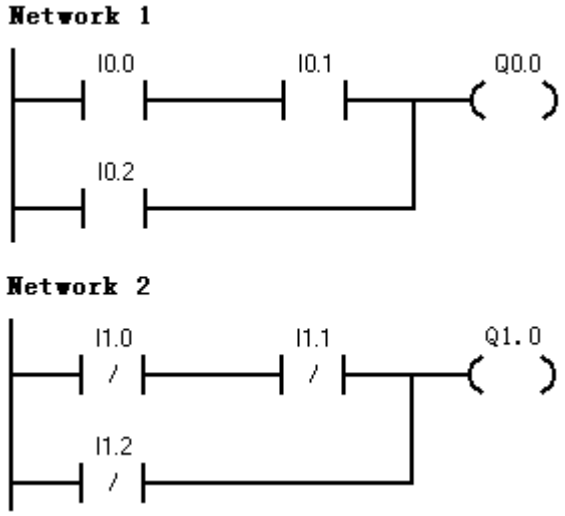
```
LD      I0.0
A       I0.1
O       I0.2
=       Q0.0    //如果 I0.0 与 I0.1 同时为 1，或者 I0.2 为 1，则 Q0.0 点亮
```

Network 2

```
LDN     I1.0
AN      I1.1
ON      I1.2
=       Q1.0    //如果 I1.0 与 I1.1 同时为 0，或者 I1.2 为 0，则 Q1.0 点亮
```

梯形图：

ORGANIZATION_BLOCK 主程序：OB1

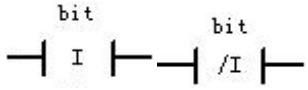


立即触点指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

```
LDI     bit
AI      bit
OI      bit
LDNI    bit
ANI     bit
ONI     bit
```

梯形图：



输入输出参数	数据类型	适用操作数
位	BOOL	I

功能说明：

当实际输入点为 1 时，立即常开触点（LDI、AI、OI）接通。

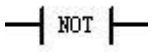
当实际输入点为 0 时，立即常闭触点（LDNI、ANI、ONI）接通。

立即指令执行时获取实际输入值，不更新 CPU 进程映象寄存器。立即触点的参数值会立即更新，而不依赖于 CPU 的扫描周期。

在 STL 中，立即常开触点用立即载入（LDI）、立即与（AI）和立即或（OI）指令表示，立即常开指令立即将实际输入值装载、与、或至栈顶。

立即常闭触点用立即载入非（LDNI）、立即与非（ANI）和立即或非（ONI）指令表示，立即常闭指令立即将实际输入点数值的逻辑非装载、与、或至栈顶。

在 LAD 中，立即常开和立即常闭指令用触点表示。

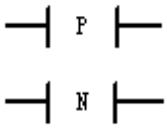
取反指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： NOT		
梯形图： 		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明：

取反（NOT）触点将输入的使能位状态取反。当使能位为 1，取反（NOT）触点断开，当使能位为 0，取反（NOT）触点接通。

在 STL 中，取反（NOT）指令将堆栈顶部的数值从 0 改变为 1，或从 1 改变为 0。

在 LAD 中，取反（NOT）指令用触点表示。

边沿检测指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： EU ED		
梯形图： 		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明：

上升沿检测（EU）指令比较本次扫描与上次扫描的输入能流的变化，如能流有上升沿变化时（OFF→ON），将能流置为 ON 状态一个扫描间隔；其他情况下将能流置为 OFF 状态。

下降沿检测（ED）指令比较本次扫描与上次扫描的输入能流的变化，如能流有下降沿变化时（ON→OFF），将能流置为 ON 状态一个扫描间隔；其他情况下将能流置为 OFF 状态。

在 STL 中，上升沿检测指令一旦在堆栈顶部数值中检测到 0 至 1 转换时，则将堆栈顶值设为 1；否则，将其设为 0。

下降沿检测指令一旦在堆栈顶部数值中检测到 1 至 0 转换时，则将堆栈顶值设为 1；否则，将其设为 0。

对于运行时编辑操作，您必须为边沿检测指令输入一个参数。有关详情，请参阅在运行模式中进行程序编辑。

在 LAD 中，边沿检测指令用触点表示。

提示

因为边沿检测指令要求执行“打开至关闭”或“关闭至打开”转换，所以无法在首次扫描时检测上升

沿或下降沿。在首次扫描中，CTSC-200 设置由这些指令指定的位状态。在其后的扫描中，这些指令才能检测到指定位的转换。

逻辑堆栈指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

ALD

OLD

LPS

LDS n

LRD

LPP

梯形图： 梯形图无此指令

输入输出参数	数据类型	适用操作数
LDS 指令操作数 n	无	常数（0～8）

功能说明：

与载入指令（ALD）采用逻辑与操作将堆栈第一级和第二级中的数值组合，并将结果载入堆栈顶部。执行 ALD 后，堆栈深度减 1。

或载入指令（OLD）采用逻辑或操作将堆栈第一级和第二级中的数值组合，并将结果载入堆栈顶部。执行 OLD 后，堆栈深度减 1。

逻辑进栈指令（LPS）复制堆栈中的顶值并使该数值进栈。堆栈底值被推出栈并丢失。

逻辑出栈指令（LPP）将堆栈中的一个数值出栈。第二个堆栈数值成为堆栈新顶值。

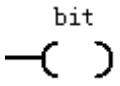
逻辑读取指令（LRD）将第二个堆栈数值复制至堆栈顶部。不执行进栈或出栈，但旧堆栈顶值被复制破坏。

载入堆栈指令（LDS）复制堆栈中的堆栈位 n，并将该数值置于堆栈顶部。堆栈底值被推出栈并丢失。

输出指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： = bit

梯形图：



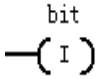
输入输出参数	数据类型	适用操作数
输出位 bit	BOOL	I, Q, M, SM, T, C, V, S, L
使能位（LAD）	BOOL	使能位

功能说明：

输出指令（=）将输出位的新数值写入过程映像寄存器。

在 STL，位于堆栈顶端的数值被复制至指定的位。

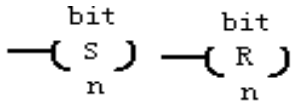
在 LAD 中，当输出指令被执行时，CPU 将过程映像寄存器中的输出位打开或关闭，指定的位被设为等于使能位。

立即输出指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： =I bit		
梯形图： 		
输入输出参数	数据类型	适用操作数
输出位 bit	BOOL	Q
使能位（LAD）	BOOL	使能位

功能说明：

执行指令时，立即输出指令（=I）将新值写入实际输出和对应的过程映像寄存器位置。执行“立即输出”指令时， 实际输出点的值被立即设为等于使能位。"I"表示立即输出；执行指令时，新值被写入实际输出和对应的过程映像寄存器位置。这与非立即输出不同，非立即输出仅将新值写入过程映像寄存器而不马上更新实际输出。

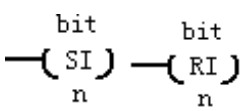
对于 STL，立即输出指令将位于堆栈顶端的数值复制至指定的实际输出位 bit。

置位复位指令	设置 ENO = 0的错误条件	0006（间接地址）、 0091（操作数超出范围）
	影响的特殊内存位	无
指令列表： S bit, n R bit, n		
梯形图： 		
输入输出参数	数据类型	适用操作数
位 bit	BOOL	I, Q, M, SM, T, C, V, S, L
位数 n	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD

功能说明：

当能流有效时，置位打开或复原从指定地址 bit 开始的 n 个点，最多可以置位和复位 255 个点。

如果“复位”指令指定一个定时器位（T）或计数器位（C），指令将复位定时器或计数器位，并清除定时器或计数器的当前值。

立即置位复位指令	设置 ENO = 0的错误条件	0006（间接地址）、0091（操作数超出范围）
	影响的特殊内存位	无
指令列表： SI bit, n RI bit, n		
梯形图： 		
输入输出参数	数据类型	适用操作数

位 bit	BOOL	Q
位数 n	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *AC, *LD

功能说明：能流有效时，立即置位指令（SI）和立即复位指令（RI）将立即置位打开或复位清零从指定地址 bit 开始的 n 个点，可以置位或复位 1 至 128 个点。执行立即置位复位指令时，新值被写入实际输出点和相应的过程映象寄存器位置。这与非立即指令不同，非立即指令只将新值写入过程映象寄存器而不会在扫描周期中立即更新物理输出点。

功率流与指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

AENO

梯形图： 梯形图无此指令

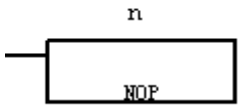
输入输出参数	数据类型	适用操作数
无	无	无

功能说明：AENO 是 ENO 位的逻辑与运算，运算的结果放入栈顶。ENO 是 LAD 中指令盒的布尔输出。如果指令盒在 EN 输入位置被使能而且执行时无错，则 ENO 输出向下一个指令盒传递能流。ENO 位用于堆栈顶部，影响其后指令执行的使能位。STL 指令没有 EN 输入；堆栈顶值必须为逻辑 1，有条件指令才能执行，STL 中也没有 ENO 输出，但采用 AENO 指令可存取 ENO 输出位。AENO 可用于生成与方框 ENO 位相同的效果。

空操作指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： NOP n

梯形图：



输入输出参数	数据类型	适用操作数
n	字节	常数（0～255）

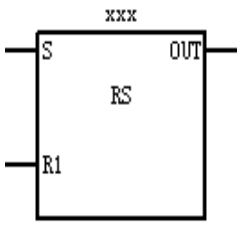
功能说明：

空操作指令（NOP）对用户程序执行空操作，可以用在用户程序需要短暂延时的地方。

复位主双稳态触发器	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： STL 中不存在此指令

梯形图：



输入输出参数	数据类型	适用操作数
S	BOOL	Power Flow
R1	BOOL	Power Flow
xxx	BOOL	I, Q, M, V, S

OUT	BOOL	Power Flow

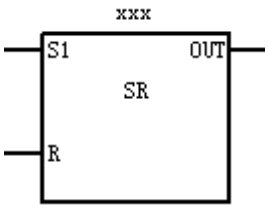
功能说明：复原主双稳态触发器（RS）是一种复原主要位的锁存器。如果设置（S）和复原（R）信号均为真实，则输出（OUT）为虚假。“位”参数指定被设置或复原的布尔参数。供选用输出反映位参数的信号状态。

设置主双稳态触发器	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

STL 中不存在此指令

梯形图：



输入输出参数	数据类型	适用操作数
S1	BOOL	Power Flow
R	BOOL	Power Flow
xxx	BOOL	I, Q, M, V, S
OUT	BOOL	Power Flow

功能说明：设置主双稳态触发器（SR）是一种设置主要位的锁存器。如果设置（S1）和复原（R）信号均为真实，则输出（OUT）为真实。“位”参数指定被设置或复原的布尔参数。供选用输出反映位参数的信号状态。

比较指令

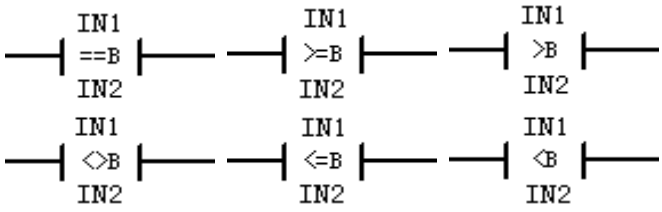
字节比较指令	设置 ENO = 0的错误条件	非法间接地址
	影响的特殊内存位	无

指令列表：

LDB= IN1, IN2
AB= IN1, IN2
OB= IN1, IN2
LDB<> IN1, IN2
AB<> IN1, IN2
OB<> IN1, IN2
LDB>= IN1, IN2
AB>= IN1, IN2
OB>= IN1, IN2
LDB<= IN1, IN2
AB<= IN1, IN2
OB<= IN1, IN2
LDB> IN1, IN2
AB> IN1, IN2

OB>　IN1, IN2
LDB< IN1, IN2
AB<　IN1, IN2
OB<　IN1, IN2

梯形图：



输入输出参数	数据类型	适用操作数
IN1	字节	IB, QB, MB, SMB, VB, SB, LB, AC, 常数, *VD, *LD, *AC
IN2	字节	IB, QB, MB, SMB, VB, SB, LB, AC, 常数, *VD, *LD, *AC

功能说明：

无论使能位状态如何，"比较"指令均会执行。字节比较指令用于比较两个字节型的值，字节比较不带符号。

在 STL 中，比较结果为真时，栈顶为 1。

在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTSC-200 立即停止执行程序：

- 非法间接地址
- 比较实数指令中存在非法实数（例如，NAN）

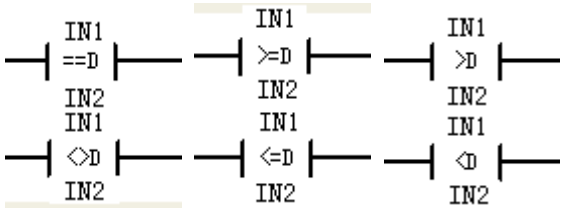
为防止出现以上错误，请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

双整数比较指令	设置 ENO = 0的错误条件	非法间接地址
	影响的特殊内存位	无

指令列表：

LDD= IN1, IN2
AD=　IN1, IN2
OD=　IN1, IN2
LDD<> IN1, IN2
AD<>　IN1, IN2
OD<>　IN1, IN2
LDD>= IN1, IN2
AD>=　IN1, IN2
OD>=　IN1, IN2
LDD<= IN1, IN2
AD<=　IN1, IN2
OD<=　IN1, IN2
LDD> IN1, IN2
AD>　IN1, IN2
OD>　IN1, IN2
LDD< IN1, IN2
AD<　IN1, IN2
OD<　IN1, IN2

梯形图：



输入输出参数	数据类型	适用操作数
IN1	双整数	ID, QD, MD, SD, SMD, VD, LD, HC, AC, 常数, *VD, *LD, *AC
IN2	双整数	ID, QD, MD, SD, SMD, VD, LD, HC, AC, 常数, *VD, *LD, *AC

功能说明：

无论使能位状态如何，“比较”指令均会执行。双整数比较指令用于比较两个双整型的值，双整数比较带符号（如 16#7FFFFFFF > 16#80000000，因为 16#80000000 为负）。

在 STL 中，比较结果为真时，栈顶为 1。

在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTSC-200 立即停止执行程序：

- 非法间接地址
- 比较实数指令中存在非法实数（例如，NAN）

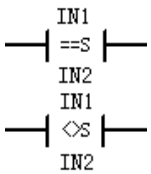
为防止出现以上错误，请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

字符串比较指令	设置 ENO = 0的错误条件	非法间接地址 遇到长度超过254个字符的字符串 字符串的起始地址和长度无法放入一个指定的内存区
	影响的特殊内存位	无

指令列表：

```
LDS= IN1, IN2
AS=  IN1, IN2
OS=  IN1, IN2
LDS<> IN1, IN2
AS<>  IN1, IN2
OS<>  IN1, IN2
```

梯形图：



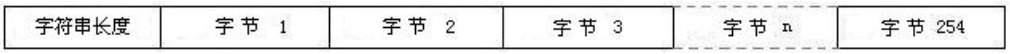
输入输出参数	数据类型	适用操作数
IN1	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
IN2	字符串	VB, LB, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的

格式(一个方格代表一个字节)：



功能说明：

无论使能位状态如何，“比较”指令均会执行。字符串比较指令用于比较两个字符串是否相同。

在 STL 中，比较结果为真时，栈顶值为 1。

在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTSC-200 立即停止执行程序：

- 非法间接地址
- 长度超过 255 个字符的字符串
- 字符串的起始地址和长度无法放入一个指定的内存区

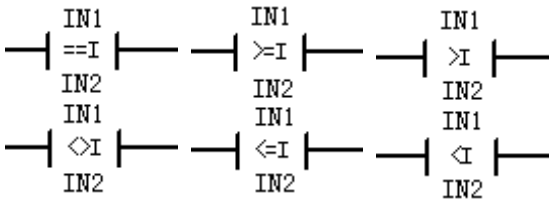
为了防止出现以上错误，请确保在使用比较指令前指针和存放 ASCII 字符串的内存位置已正确初始化并且为 ASCII 字符串保留的缓冲区可完全放置在指定的内存区中。

整数比较指令	设置 ENO = 0的错误条件	非法间接地址
	影响的特殊内存位	无

指令列表：

```
LDW= IN1, IN2
AW=  IN1, IN2
OW=  IN1, IN2
LDW<> IN1, IN2
AW<>  IN1, IN2
OW<>  IN1, IN2
LDW>= IN1, IN2
AW>=  IN1, IN2
OW>=  IN1, IN2
LDW<= IN1, IN2
AW<=  IN1, IN2
OW<=  IN1, IN2
LDW> IN1, IN2
AW>  IN1, IN2
OW>  IN1, IN2
LDW< IN1, IN2
AW<  IN1, IN2
OW<  IN1, IN2
```

梯形图：



输入输出参数	数据类型	适用操作数
IN1	整数	IW, QW, MW, SW, SMW, T, C, VW, LW, AIW, AC, 常数, *VD, *LD,*AC

IN2	整数	IW, QW, MW, SW, SMW, T, C, VW, LW, AIW, AC, 常数, *VD, *LD,*AC

功能说明：

无论使能位状态如何，"比较"指令均会执行。整数比较指令用于比较两个整数型的值，整数比较带符号（如 16#7FFF > 16#8000，因为 16#8000 为负）。

在 STL 中，比较结果为真时，栈顶为 1。
在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTSC-200 立即停止执行程序：

- 非法间接地址
- 比较实数指令中存在非法实数（例如，NAN）

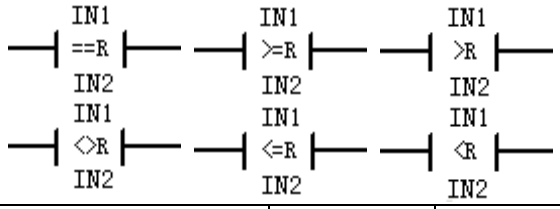
为防止出现以上错误，请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

实数比较指令	设置 ENO = 0的错误条件	非法间接地址 非法实数
	影响的特殊内存位	无

指令列表：

```
LDR= IN1, IN2
AR=  IN1, IN2
OR=  IN1, IN2
LDR<> IN1, IN2
AR<>  IN1, IN2
OR<>  IN1, IN2
LDR>= IN1, IN2
AR>=  IN1, IN2
OR>=  IN1, IN2
LDR<= IN1, IN2
AR<=  IN1, IN2
OR<=  IN1, IN2
LDR> IN1, IN2
AR>  IN1, IN2
OR>  IN1, IN2
LDR< IN1, IN2
AR<  IN1, IN2
OR<  IN1, IN2
```

梯形图：



输入输出参数	数据类型	适用操作数
IN1	实数	ID, QD, MD, SD, SMD, VD, LD, AC, 常数, *VD, *LD, *AC
IN2	实数	ID, QD, MD, SD, SMD, VD, LD, AC, 常数, *VD, *LD, *AC

功能说明：

无论使能位状态如何，"比较"指令均会执行。实数比较指令用于比较两个实数的值，实数比较带符号。

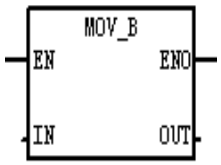
在 STL 中，比较结果为真时，栈顶值为 1。
在 LAD 中，比较结果为真时，触点打开。

<注意> 执行比较指令时如出现以下错误为致命错误，会使 CTSC-200 立即停止执行程序：

- 非法间接地址
- 比较实数指令中存在非法实数（例如，NAN）

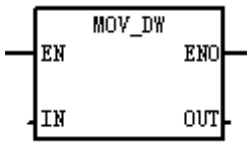
为防止出现以上错误，请确保在使用比较指令前指针已正确初始化并且参数中不存在非法实数。

传送指令

字节赋值指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOVB IN, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

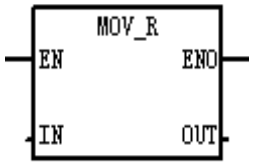
功能说明：字节赋值指令（MOVB）将输入字节 IN 的值拷贝赋给输出字节型变量 OUT。

双字赋值指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOVD IN OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, &SMB, &AIW, &AQW AC, 常数, *VD, *LD, *AC
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：双字赋值指令（MOVD）将输入双字 IN 的值拷贝赋给输出双字型变量 OUT。

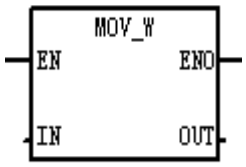
浮点数赋值指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOVR IN OUT		
梯形图：		



输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SD, SMD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：浮点数赋值指令（MOVR）将输入实数 IN 的值拷贝赋给输出浮点型变量 OUT。

字赋值指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： MOVW IN, OUT		
梯形图：		



输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, 常数, AC, *VD, *AC, *LD
OUT	字	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD

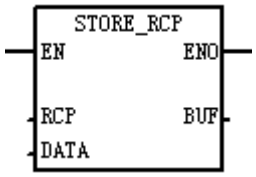
功能说明：字赋值指令（MOVW）将输入字 IN 的值拷贝赋给输出字型变量 OUT。

存储数据记录到存储卡	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： SDL LOG, BUF		
梯形图：		

输入输出参数	数据类型	适用操作数
LOG	字节	常数 0~3
BUF	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：将当前项目中的数据记录 LOG 拷贝到存储卡中。

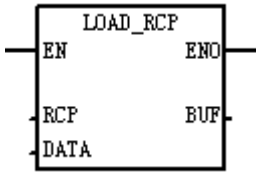
存储配方到存储卡	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： SRCP RCP, DATA, BUF		
梯形图：		



输入输出参数	数据类型	适用操作数
RCP	字节	常数 0~3
DATA	字	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
BUF	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：将当前项目中的配方 RCP 拷贝到存储卡中。

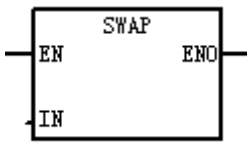
从存储卡装载配方	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： LRCP RCP, DATA, BUF		
梯形图：		



输入输出参数	数据类型	适用操作数
RCP	字节	常数 0~3
DATA	字	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD
BUF	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：将存储卡上的配方 RCP 拷贝到以 BUF 指定的地址为起始地址的内存中。

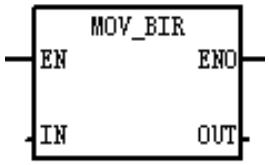
字节交换指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： SWAP IN		
梯形图：		



输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD

功能说明：字节交换指令（SWAP）交换输入字 IN 的高位字节和低位字节。

赋值字节立即读取指令	设置 ENO = 0的错误条件	0006 间接地址 无法存取扩充模块（仅限立即指令）
	影响的特殊内存位	无
指令列表： BIR IN, OUT		
梯形图：		



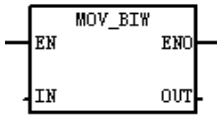
输入输出参数	数据类型	适用操作数
IN	字节	IB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

功能说明：赋值字节立即读取指令（BIR）读取物理端口输入字节 IN 的值，并将结果拷贝赋给 OUT 指定的地址中，不更新进程映像寄存器。

赋值字节立即写入指令	设置 ENO = 0的错误条件	0006 间接地址 无法存取扩充模块（仅限立即指令）
	影响的特殊内存位	无

指令列表： BIW IN, OUT

梯形图：



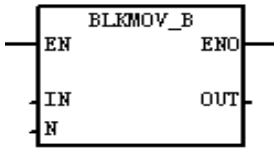
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
OUT	字节	QB, *VD, *LD, *AC

功能说明：赋值字节立即写入指令（BIW）从读取输入字节 IN 的值，并将值立即写入指定的物理端口 OUT 和对应的进程映像寄存器。

成块赋值字节指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无

指令列表： BMB IN, OUT, N

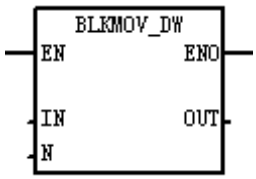
梯形图：



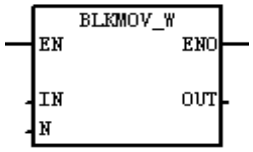
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD

功能说明：成块赋值字节指令（BMB）将从起始地址 IN 开始的 N 个字节的值拷贝赋给 从输出地址 OUT 开始的 N 个字节，N 的范围为 1 至 255。

成块赋值双字指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无

指令列表： BMD IN, OUT, N		
梯形图：  		
输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC，常数, *VD, *AC, *LD

功能说明：成块赋值双字指令（BMD）将从起始地址 IN 开始的 N 个双字的值拷贝赋给 从输出地址 OUT 开始的 N 个双字，N 的范围为 1 至 255。

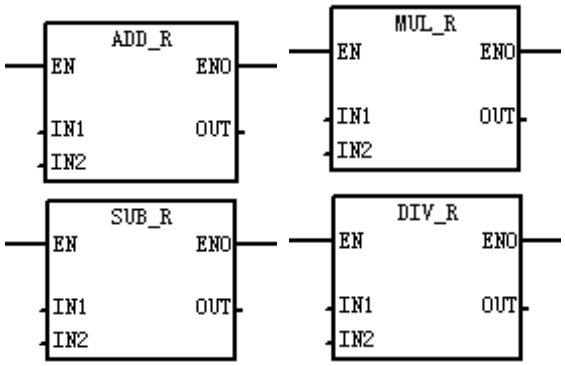
成块赋值字指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
指令列表： BMW IN, OUT, N		
梯形图：  		
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, *VD, *LD, *AC
OUT	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC，常数, *VD, *AC, *LD

功能说明：成块赋值字指令（BMW）将从起始地址 IN 开始的 N 个字的值拷贝赋给 从输出地址 OUT 开始的 N 个字，N 的范围为 1 至 255。

浮点数运算

浮点数加减乘除指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0(/R)
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0(/R)
指令列表： +R IN1 OUT -R IN1 OUT *R IN1 OUT /R IN1 OUT		

梯形图：



输入输出参数	数据类型	适用操作数
IN1	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
IN2	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：

浮点数加法指令（+R）将两个 32 位实数相加，并将计算结果（32 位实数）放入 OUT 指定的地址中。

浮点数减法指令（-R）将两个 32 位实数相减，并将计算结果（32 位实数）放入 OUT 指定的地址中。

浮点数乘法指令（*R）将两个 32 位实数相乘，并将乘积（32 位实数）放入 OUT 指定的地址中。

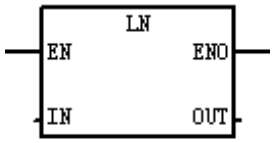
浮点数除法指令（/R）将两个 32 位实数相除，并将商（32 位实数）放入 OUT 指定的地址中。

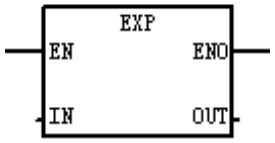
在 LAD 中：IN1 + IN2 = OUT
IN1 - IN2 = OUT
IN1 * IN2 = OUT
IN1 / IN2 = OUT

在 STL 中：IN1 + OUT = OUT
OUT - IN1 = OUT
IN1 * OUT = OUT
OUT / IN1 = OUT

SM1.1 指示溢出错误和非法数值。如果在除法运算中设置 SM1.3，则其他数学状态位不变，而且原始输入操作数也不变。SM1.1 用于表示溢出错误和非法数值。如果设置 SM1.1，则 SM1.0 和 SM1.2 状态无效，原始输入操作数不变。如果在除法运算过程中未设置 SM1.1 和 SM1.3，则说明数学运算已完成，得出有效结果，而且 SM1.0 和 SM1.2 包含有效状态。

自然对数自然指数指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果
指令列表： LN IN, OUT EXP IN, OUT		
梯形图：		





输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：

自然对数指令（LN）对 IN 中的数值进行自然对数计算，并将结果置于 OUT 中。欲从自然对数获得以 10 为底数的对数，用自然对数除以 2.302585（约等于 10 的自然对数）。欲将任何实数提升为另一个实数的乘幂，包括分数指数：将"自然指数"指令与"自然对数"指令相结合。例如，欲将 X 提升为 Y 乘幂，输入以下指令：EXP (Y * LN (X))。

自然指数指令（EXP）进行 e 的 IN 次方指数计算，并将结果置于 OUT 中。欲将任何实数提升为另一个实数的乘幂，包括分数指数：将"自然指数"指令与"自然对数"指令相结合。例如，欲将 X 提升为 Y 乘幂，输入以下指令：

EXP (Y * LN (X))。

举例：

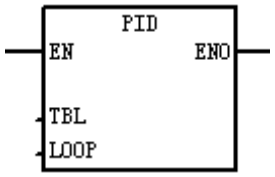
5 的立方 = 5^3=EXP(3*LN(5))=125

125 的立方根 = 125^(1/3)=EXP(1/3)*LN(125))=5

5 的立方的平方根 = 5^(3/2)=EXP(3/2*LN(5))=11.18034

SM1.1 用于指示溢出错误和非法数值。如果设置 SM1.1，则 SM1.0 和 SM1.2 状态无效，且原来的输入操作数不改动。如果未设置 SM1.3，则数学操作完成，并产生有效的结果，且 SM1.0 和 SM1.2 包含有效状态。

PID 控制指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.1 溢出
指令列表： PID TBL LOOP		
梯形图：		



输入输出参数	数据类型	适用操作数
TBL	字节	VB
LOOP	字节	常数(0 to 7)

功能说明：

本指令有两个操作数：表示回路表起始地址的 TBL 地址和 0 至 7 常数的"回路"号码。程序中可使用八条

PID 指令。如果两条或多条 PID 指令使用相同的回路号码（即使它们的表格地址不同），PID 计算会互相干扰，结果难以预料。回路表存储用于控制和监控回路运算的参数，包括程序变量、设置点、输出、增益、样本时间、整数时间（重设）、导出时间（速率）以及整数和（偏差）的当前值及先前值。

如果回路表起始地址或指令中指定的 PID 回路号码操作数超出范围，CPU 编译器将生成一则错误（范围错误），编译将会失败。

PID 指令不对某些回路表输入值进行范围检查。您必须保证程序变量和设置点（以及作为输入的偏差和先前程序变量）是 0.0 和 1.0 之间的实数。

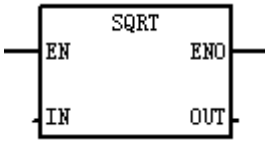
如果进行 PID 计算的数学运算时遇到错误，将设置 SM1.1（溢出或非法数值）并终止 PID 指令的执行。（对回路表中的输出数值的更新可能不完整，因此您应当忽略这些数值，并在执行下一个回路 PID 指令之前纠正引起数学错误的输入值。）

取样速率：

欲按要求的样品速率进行 PID 计算，必须按计时器的控制速率从定时中断程序或从主程序执行 PID 指令。样品时间必须通过回路表作为 PID 指令输入提供。

使用 PID 向导：

MagicWorks PLC 提供 PID 向导，指导您为闭路控制程序定义 PID 算法。选择菜单命令工具 > 指令向导，并从指令向导窗口选择 PID。

平方根指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果
指令列表： SQRT IN OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

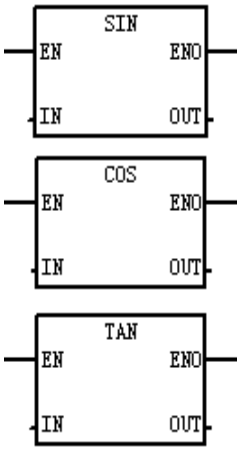
功能说明：

平方根指令（SQRT）对 32 位实数（IN）取平方根，产生一个 32 位实数结果并将结果放入 OUT 中，如以下等式所示：

$\sqrt{IN} = OUT$

SM1.1 用于表示溢出错误和非法数值。如果设置 SM1.1，则 SM1.0 和 SM1.2 状态无效，原始输入操作数不变。如果未设置 SM1.1，则说明数学运算已完成，得出有效结果，而且 SM1.0 和 SM1.2 包含有效状态。

正弦余弦正切指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果

		SM1.1 溢出 SM1.2 负结果
指令列表： SIN IN OUT COS IN OUT TAN IN OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, 常数, *VD, *LD, *AC
OUT	浮点数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：

正弦指令（SIN）对角度值 IN 进行三角运算，并将结果放置在 OUT 中。输入角以弧度为单位。欲将输入角从角度转换成弧度，用角度乘以 1.745329E-2（约等于 pi/180）。

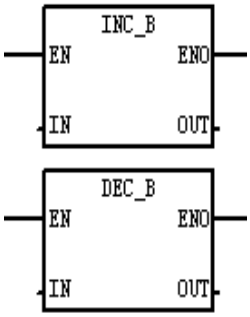
余弦指令（COS）对角度值 IN 进行三角运算，并将结果放置在 OUT 中。输入角以弧度为单位。欲将输入角从角度转换成弧度，用角度乘以 1.745329E-2（约等于 pi/180）。

正切指令（TAN）对角度值 IN 进行三角运算，并将结果放置在 OUT 中。输入角以弧度为单位。欲将输入角从角度转换成弧度，用角度乘以 1.745329E-2（约等于 pi/180）。

SM1.1 用于指示溢出错误和非法数值。如果设置 SM1.1，则 SM1.0 和 SM1.2 状态无效，且原来的输入操作数不改动。如果未设置 SM1.3，则数学操作完成，并产生有效的结果，且 SM1.0 和 SM1.2 包含有效状态。

整数计算指令

字节自加自减指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出
指令列表： INCB OUT DECB OUT		
梯形图：		



输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：

字节自加指令（INCB）在输入字节 IN 上加 1，并将加 1 的结果放入 OUT 指定的地址中。
字节自减指令（DECB）在输入字节 IN 上减 1，并将减 1 的结果放入 OUT 指定的地址中。

字节自加和自减运算不带符号。

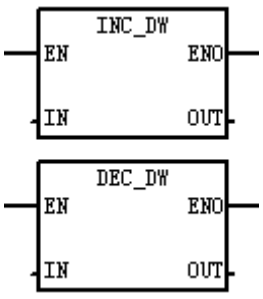
在 LAD 中：IN + 1 = OUT
IN - 1 = OUT

在 STL 中：OUT + 1 = OUT
OUT - 1 = OUT

双整数自加自减指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果

指令列表：
INCD OUT
DECD OUT

梯形图：



输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, HC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：

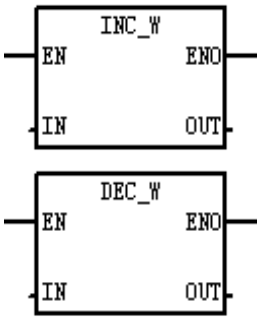
双整数自加指令（INCD）在输入双字 IN 上加 1，并将加 1 的结果放入 OUT 指定的地址中。

双整数自减指令（DECD）在输入双字 IN 上减 1，并将减 1 的结果放入 OUT 指定的地址中。

双整数自加和自减运算带符号（16#7FFFFFFF > 16#80000000）。

在 LAD 中： IN + 1 = OUT
IN - 1 = OUT

在 STL 中： OUT + 1 = OUT
OUT - 1 = OUT

整数自加自减指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果
指令列表： INCW OUT DECW OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, AC, AIW, LW, T, C, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, *VD, *LD, *AC 整数

功能说明：

整数自加指令（INCW）在输入字 IN 上加 1，并将加 1 的结果放入 OUT 指定的地址中。

整数自减指令（DECW）在输入字 IN 上减 1，并将减 1 的结果放入 OUT 指定的地址中。

整数自加和自减运算带符号（16#7FFF > 16#8000）。

在 LAD 中：IN + 1 = OUT
IN - 1 = OUT

在 STL 中：OUT + 1 = OUT
OUT - 1 = OUT

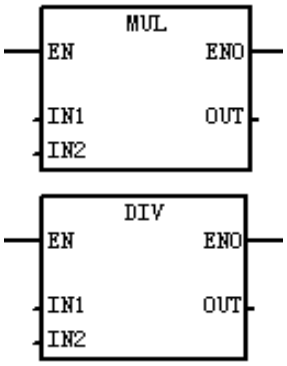
整数与双整数的乘除指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0

指令列表：

MUL IN1 OUT

DIV IN1 OUT

梯形图：



输入输出参数	数据类型	适用操作数
IN1	整数	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, 常数, *VD, *LD, *AC
IN2	整数	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：

在 STL 乘法指令中，32 位 OUT 的低位字（16 位）被用作乘数之一。在 STL 除法指令中，32 位 OUT 的低位字（16 位）被用作除数。特殊内存（SM）位指示错误和非法数值。如果在除法运算过程中设置 SM1.3（除以零），则其他数学状态位被保留不变。否则，在完成数学运算时，所有受支持的数学状态位均包含有效状态。

在 LAD 中：IN1 * IN2 = OUT
IN1 / IN2 = OUT

在 STL 中：IN1 * OUT = OUT
OUT / IN1 = OUT

双整数加减乘除指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0(/I)
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0(/I)

指令列表：

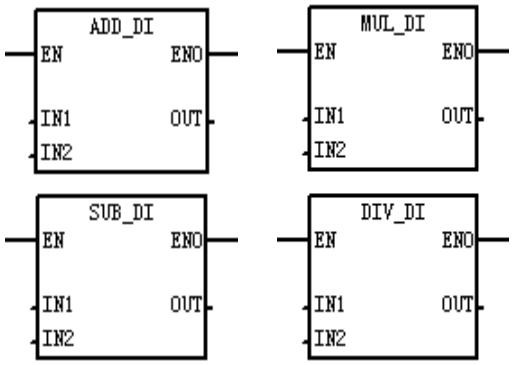
+D IN1 OUT

-D IN1 OUT

*D IN1 OUT

/D IN1 OUT

梯形图：



输入输出参数	数据类型	适用操作数
IN1	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, HC, 常数, *VD, *LD, *AC
IN2	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, HC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：

双整数加法运算指令（+D）将两个 32 位整数相加，并将计算结果（32 位整数）放入 OUT 指定的地址中。如果结果超出 32 位长度表示范围，则设置溢出位。

双整数减法运算指令（-D）将两个 32 位整数相减，并将计算结果（32 位整数）放入 OUT 指定的地址中。如果结果超出 32 位长度表示范围，则设置溢出位。

双整数乘法运算指令（*D）将两个 32 位整数相乘，并将乘积结果（32 位整数）放入 OUT 指定的地址中。如果结果超出 32 位长度表示范围，则设置溢出位。

双整数除法运算指令（/D）将两个 32 位整数相除，并将商（32 位整数）放入 OUT 指定的地址中，不保留余数。如果结果超出 32 位长度表示范围，则设置溢出位。

在 LAD 中：IN1 + IN2 = OUT
IN1 - IN2 = OUT
IN1 * IN2 = OUT
IN1 / IN2 = OUT

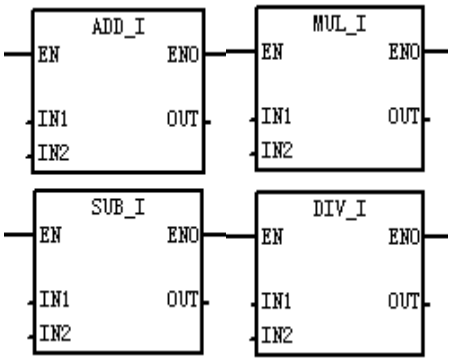
在 STL 中：IN1 + OUT = OUT
OUT - IN1 = OUT
IN1 * OUT = OUT
OUT / IN1 = OUT

SM1.1 指示溢出错误和非法数值。如果设置 SM1.1，则 SM1.0 和 SM1.2 状态无效，且原来的输入操作数不改动。如果未设置 SM1.3，则数学操作完成，并产生有效的结果，且 SM1.0 和 SM1.2 包含有效状态。如果在除法运算过程中设置 SM1.3，则其他数学状态位将保持不变。

整数加减乘除指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出 SM1.3 除数为0(/I)
	影响的特殊内存位	SM1.0 零结果 SM1.1 溢出 SM1.2 负结果 SM1.3 除数为0(/I)
指令列表： +I IN1 OUT -I IN1 OUT		

*I IN1 OUT
/I IN1 OUT

梯形图：



输入输出参数	数据类型	适用操作数
IN1	整数	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, 常数, *VD, *LD, *AC
IN2	整数	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *LD, *AC

功能说明：

整数加法运算指令（+I）将两个 16 位整数相加，并将计算结果（16 位整数）放入 OUT 指定的地址中。如果结果超出 16 位长度表示范围，则设置溢出位。

整数减法运算指令（-I）将两个 16 位整数相减，并将计算结果（16 位整数）放入 OUT 指定的地址中。如果结果超出 16 位长度表示范围，则设置溢出位。

整数乘法运算指令（*I）将两个 16 位整数相乘，并将乘积结果（16 位整数）放入 OUT 指定的地址中。如果结果超出 16 位长度表示范围，则设置溢出位。

整数除法运算指令（/I）将两个 16 位整数相除，并将商（16 位整数）放入 OUT 指定的地址中，不保留余数。如果结果超出 16 位长度表示范围，则设置溢出位。

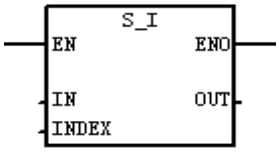
在 LAD 中：IN1 + IN2 = OUT
IN1 - IN2 = OUT
IN1 * IN2 = OUT
IN1 / IN2 = OUT

在 STL 中：IN1 + OUT = OUT
OUT - IN1 = OUT
IN1 * OUT = OUT
OUT / IN1 = OUT

SM1.1 指示溢出错误和非法数值。如果设置 SM1.1，则 SM1.0 和 SM1.2 状态无效，且原来的输入操作数不改动。如果未设置 SM1.3，则数学操作完成，并产生有效的结果，且 SM1.0 和 SM1.2 包含有效状态。如果在除法运算过程中设置 SM1.3，则其他数学状态位将保持不变。

转换指令

字符串至整数转换指令	设置 ENO = 0的错	0006 间接地址

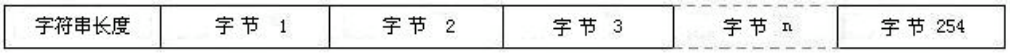
	误条件	0091 操作数范围 009B 非法指数(指定起始位置值0的字符串操作) SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1(溢出或非法值)
指令列表： STI IN, INDEX, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, AC, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC,*VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



STI 指令用以下形式转换字符串：

[spaces] [+ or -] [digits 0-9]

INDEX 值通常设为 1，从字符串的第一个字符开始转换。可将该值设为其他值，在字符串中的不同点开始转换。当输入字符串包含不属于需要转换数字一部分的文本时，可采用此种方法。例如，如果输入字符串是"value:16000 "，您可以将 INDEX 设为值 7，跳过字符串开始的字"value:"。

当达到字符串结束时或遇到第一个无效字符时，转换终止。无效字符是数字（0-9）以外的任何字符。

每当转换产生一个对于输出值过大的整数值时，则设置溢出错误（SM1.1）。例如，如果输入字符串产生一个大于 32767 或小于-32768 的值时，STI 指令设置溢出错误。

如果当输入字符串未包含有效值而无法执行转换时，也会设置溢出错误（SM1.1）。例如，如果输入字符串包含"A123"，转换指令会设置 SM1.1（溢出），输出值保持不变。

下表是有效和无效字符串转换为整数/双整数/实数时的举例（INDEX = 1）：

输入字符串	字符串有效性	转换为整数	转换为双整数	转换为实数
"65535"	有效	65535	65535	65535.0
"-0032768"	有效	-32768	-32768	-32768.0
"+32767.999 "	有效	32767	32767	32767.0
"00065535FFFF"	有效	65535	65535	65535.0
"65536"	有效	65536(SM1.1=1, 溢出)	65536	65536.0
"-002147483648"	有效	-2147483648(SM1.1=1,溢出)	-2147483648	-2147483648.0
"+002147483648"	有效	+2147483648(S	+2147483648(	+2147483648.0

		M1.1=1,溢出)	SM1.1=1,溢出)	
"F255"	无 效 (SM1.1=1, 无效)			
"++255"	无 效 (SM1.1=1, 无效)			
" "	无 效 (SM1.1=1, 无效)			

功能说明：字符串至整数转换指令将字符串数值 IN 转换为整数值，并将转换结果存储在 OUT 指定的地址中，从字符串的第 INDEX 个字符位置开始转换。

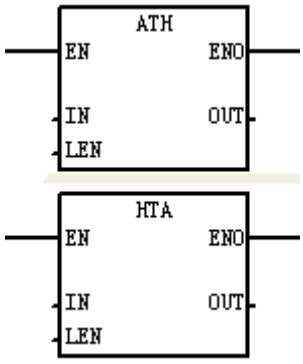
ASCII 码与十六进制数字的转换指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 SM1.7 非法 ASCII 值(ATH)
	影响的特殊内存位	SM1.7（非法 ASCII）

指令列表：

ATH IN, OUT, LEN

HTA IN, OUT, LEN

梯形图：



输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD
LEN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

操作数说明：

有效 ASCII 输入字符为：

字母数字字符　　0～9　和　A～F

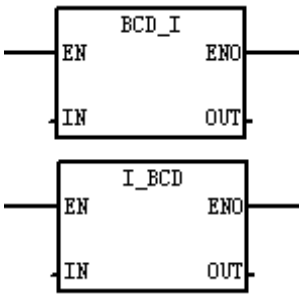
对应十六进制代码值　30～39 和 41～46

功能说明：

ASCII 至 HEX 转换指令将从内存位置 IN 开始的长度为 LEN 的 ASCII 字符号码转换十六进制数字，并放入 OUT 指定的地址。ASCII 字符串的最大长度为 255 字符。

HEX 至 ASCII 转换指令将从输入字节 IN 开始的十六进制数字转换成从 OUT 开始的 ASCII 字符。欲转换的十六进制数字位数由长度 LEN 指定，可转换的最大十六进制数字位数为 255。

整数与 BCD 码转换指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.6 无效 BCD 数值

	影响的特殊内存位	SM1.6（无效 BCD）
指令列表： BCDI OUT IBCD OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *AC, *LD
OUT	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC

操作数说明：

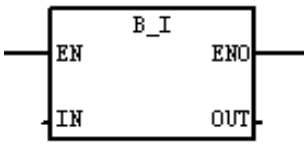
BCD 格式的有效范围：

数据类型	最小值			最大值		
字	十进制	十六进制	BCD 码	十进制	十六进制	BCD 码
字	0	0000	0000 0000 0000 0000	9999	9999	1001 1001 1001 1001

功能说明：

BCD 至整数指令将二进制编码的十进制值 IN 转换成整数值，并将结果载入 OUT 指定的变量中。IN 的有效范围是 0 至 9999 BCD。

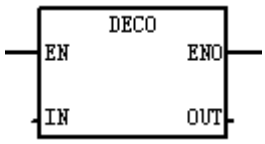
整数至 BCD 指令将输入整数值 IN 转换成二进制编码的十进制数，并将结果载入 OUT 指定的变量中。IN 的有效范围是 0 至 9999 BCD。例如，您可以将双整值转换为实数。您还可以在整数和 BCD 格式之间转换。对于 STL，IN 和 OUT 参数使用相同的地址。

字节至整数转换指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： BTI IN, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *AC, *VD, *LD
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *LD, *AC

功能说明：字节至整数转换指令将字节数值 IN 转换成整数值，并将结果写入 OUT 指定的变量中。因为字节不带符号，所以无符号扩展。

解码指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无

指令列表： DECO IN, OUT		
梯形图：  		
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SMB, LB, SB, AC, 常数, *VD, *LD, *AC
OUT	字	VW, IW, QW, MW, SMW, LW, SW, AQW, T, C, AC, *VD, *AC, *LD

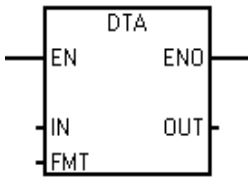
功能说明：解码指令设置输出字 OUT 中与输入字节 IN 低半字节（低 4 位）表示的位数相对应的位为 1，设置输出字 OUT 的所有其它位均为 0。

示例程序：

```
LD SM0.0
DECO 2#10101111, VW0                    //设置 VW0 的第 15(2#1111)位为 1，其它位为 0
```

运行结果：VW0 = 2#1000 0000 0000 0000

双整数至 ASCII 码转换指令	设置 ENO = 0的错误条件	0006 间接地址 FMT 位 > 0（用于 FMT 值最高的4个位） nnn > 5
	影响的特殊内存位	无

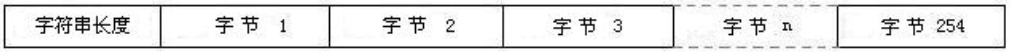
指令列表： DTA IN, OUT, FMT		
梯形图：  		
输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, 常数, AC, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

操作数说明：

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



操作数 FMT 定义：

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c：整数与小数部分的分隔符，c = 1:使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符

nnn：小数部分的位数,有效范围为 0～5，输出字符串的长度始终为 8 个字符。nnn 等于 0 会使转换结果显示为不带小数点，当 nnn 值大于 5 时，输出显示为 12 个 ASCII 空格字符的字符串。

整数至 ASCII 码的转换遵循下列规则：

1）正值写入输出缓冲区，不带符号。

2）负值写入输出缓冲区，带起始负号(-)。

3）小数点左侧的起首零（与小数点相邻的数字除外）被压缩。

4）输出缓冲区中的数值右对齐。

下表显示几个使用小数点（c = 0）和小数点右面四位小数（nnn = 100）格式的值范例。

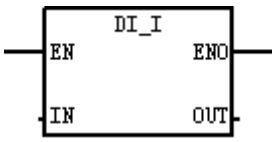
	OUT	**字节 OUT**										
		+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11
IN=123							0	.	0	1	2	3
IN=-12345						-	1	.	2	3	4	5
IN=1234567					1	2	3	.	4	5	6	7

功能说明：整数至 ASCII 转换指令将整数值 IN 转换成 ASCII 字符数组。格式 FMT 指定转换的精度，以及是将小数点显示为逗号还是点号。转换结果放入从 OUT 开始的 12 个连续字节中。ASCII 字符数组总是 12 个字符。

双整数至整数转换指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出或非法数值
	影响的特殊内存位	SM1.1（溢出）

指令列表： DTI IN, OUT

梯形图：



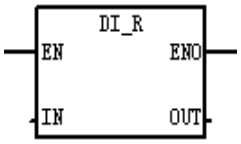
输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *LD, *AC

功能说明：双整数至整数转换指令将输入的双整数值 IN 转换成整数值，并将结果放入 OUT 指定的变量中。如果转换的值过大，则无法在输出中表示，会导致溢出错误，输出不受影响。

双整数至实数转换指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无

指令列表： DTR IN, OUT

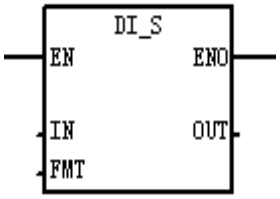
梯形图：



输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *AC, *LD

OUT	实数	VD, ID, QD, MD, SD, SMD, LD, HC, AC, *VD, *AC, *LD

功能说明：双整数至实数转换指令将 32 位带符号的整数 IN 转换成 32 位实数，并将结果放入 OUT 指定的变量中。

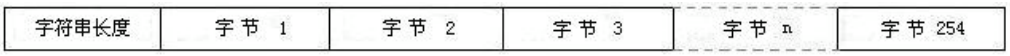
双整数至字符串转换指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 非法格式（nnn > 5）
	影响的特殊内存位	无
指令列表： DTS IN, OUT, FMT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	双整数	VD, ID, QD, MD, SD, SMD, LD, HC, 常数, AC, *VD, *AC, *LD
OUT	字符串	VB, *VD, LB, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



操作数 FMT 定义：

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c：整数与小数部分的分隔符，c = 1:使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符。

nnn：小数部分的位数,有效范围为 0～5，输出字符串的长度始终为 12 个字符。nnn 等于 0 会使转换结果显示格式为不带小数点，当 nnn 值大于 5 时，输出显示为 12 个 ASCII 空格字符的字符串。

下表显示几个使用小数点（c = 0）和小数点右面四位小数（nnn = 100）格式的值范例。位于 OUT 位置的值是字符串长度。

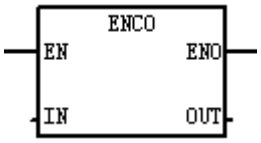
	OUT	**字节 OUT**											
		+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12
IN=123	12							0	.	0	1	2	3
IN=12345	12							1	.	2	3	4	5
IN=1234567	12					1	2	3	.	4	5	6	7

功能说明：双整数至字符串转换指令将输入的双整数 IN 转换为长度为 12 个字符的 ASCII 字符串。格式（FMT）指定小数点右面的转换精度，小数点是显示为逗号还是句点，转换结果字符串写入从 OUT 开始

的 13 个连续字节中。

提示：根据以下规则制定格式的输出字符串：

- 整数不带符号写入输出缓冲区。
- 负数带起始减号（-）写入输出缓冲区。
- 小数点左面的起始零（除紧靠小数点的数字外）被压缩。
- 小数点右面的数值被取整，使之符合指定的小数点右面的位数。
- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

编码指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： ENCO IN, OUT		
梯形图：  		

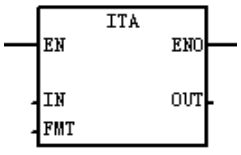
输入输出参数	数据类型	适用操作数
IN	字	VW, IW, QW, MW, SMW, LW, SW, AIW, T, C, AC, 常数, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SMB, LB, SB, AC, *VD, *LD, *AC

功能说明：编码指令将输入字 IN 位值为 1 的最低位的位数作为输出字节 OUT 的低半字节，写入输出字节的最低 4 位中。

示例程序：

```
LD          SM0.1
MOVB        0, VB0
LD          SM0.0
ENCO        2#1100000000, VB0        //2#1100000000 最低不为 0 的位是第 8 位，因此 VB0 的低半字节为 2#1000
```

运行结果：VB0 = 2#0000 1000

整数至 ASCII 码转换指令	设置 ENO = 0的错误条件	0006 间接地址 FMT 位 > 0（用于 FMT 值最高的4个位） nnn > 5
	影响的特殊内存位	无
指令列表： ITA IN, OUT, FMT		
梯形图：  		

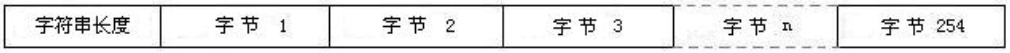
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

OUT	字节	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



操作数 FMT 定义：

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c：整数与小数部分的分隔符，c = 1:使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符。

nnn：小数部分的位数,有效范围为 0～5，输出字符串的长度始终为 8 个字符。nnn 等于 0 会使转换结果显示为不带小数点，当 nnn 值大于 5 时，输出显示为 8 个 ASCII 空格字符的字符串。

整数至 ASCII 码的转换遵循下列规则：

1）正值写入输出缓冲区，不带符号。

2）负值写入输出缓冲区，带起始负号(-)。

3）小数点左侧的起首零（与小数点相邻的数字除外）被压缩。

4）输出缓冲区中的数值右对齐。

下表显示几个使用小数点（c = 0）和小数点右面三位小数（nnn = 011）格式的值范例。

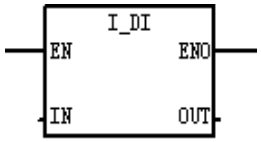
	OUT	字节 OUT						
		+1	+2	+3	+4	+5	+6	+7
IN=12				0		0	1	2
IN=-123				0		1	2	3
IN=1234				1		2	3	4
IN=-12345			1	2		3	4	5

功能说明：整数至 ASCII 转换指令将整数值 IN 转换成 ASCII 字符数组。格式 FMT 指定转换的精度，以及是将小数点显示为逗号还是点号。转换结果放入从 OUT 开始的 8 个连续字节中。ASCII 字符数组总是 8 个字符。

整数至字节转换指令	设置 ENO = 0的错误条件	0006 间接地址 溢出或非法值
	影响的特殊内存位	SM1.1（溢出）
指令列表： ITB IN, OUT		
梯形图：		

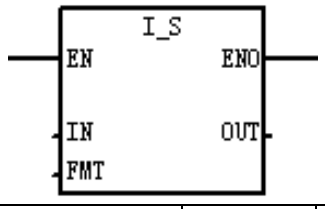
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

功能说明：整数至字节转换指令将输入的整数 IN 转换成字节值，并将结果放入 OUT 指定的变量中。数值 0 至 255 被转换。超过此范围的输入值将导致溢出，输出不受影响。

整数至双整数转换指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： ITD IN, OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：整数至双整数转换指令将输入的整数值 IN 转换成双整数值，并将结果放入 OUT 指定的变量中。符号被扩展。

整数至字符串转换指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 非法格式（nnn > 5）
	影响的特殊内存位	无
指令列表： ITS IN, OUT, FMT		
梯形图： 		

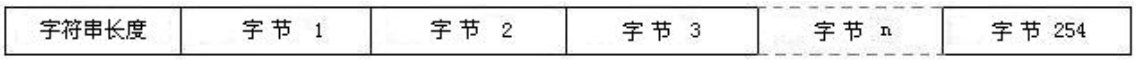
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, 常数, AC, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的

格式（一个方格代表一个字节）：



操作数 FMT 定义：

位号	7	6	5	4	3	2	1	0
值	0	0	0	0	c	n	n	n

c：整数与小数部分的分隔符，c = 1：使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符。

nnn：小数部分的位数，有效范围为 0～5，输出字符串的长度始终为 8 个字符。nnn 等于 0 会使转换结果显示格式为不带小数点，当 nnn 值大于 5 时，输出显示为 8 个 ASCII 空格字符的字符串。

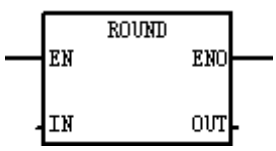
下表显示几个使用小数点（c = 0）和小数点右面三位小数（nnn = 011）格式的值范例。位于 OUT 位置的值是字符串长度。

	OUT	字节 OUT							
		+1	+2	+3	+4	+5	+6	+7	+8
IN=12	8				0	.	0	1	2
IN=-123	8				0	.	1	2	3
IN=1234	8				1	.	2	3	4
IN=-12345	8			1	2	.	3	4	5

功能说明：整数至字符串转换指令将输入的整数 IN 转换为长度为 8 个字符的 ASCII 字符串。格式（FMT）指定小数点右面的转换精度，小数点是显示为逗号还是句点，转换结果字符串写入从 OUT 开始的 9 个连续字节中。

提示：根据以下规则制定格式的输出字符串：

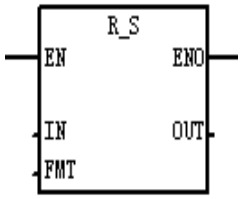
- 整数不带符号写入输出缓冲区。
- 负数带起始减号（-）写入输出缓冲区。
- 小数点左面的起始零（除紧靠小数点的数字外）被压缩。
- 小数点右面的数值被取整，使之符合指定的小数点右面的位数。
- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

取整指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1（溢出）
指令列表： ROUND IN, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, AC, 常数, *VD, *LD, *AC

OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：取整指令将输入的实数 IN 转换成双整数值，并将结果存入 OUT 指定的变量中。小数部分采取四舍五入的方法取整。

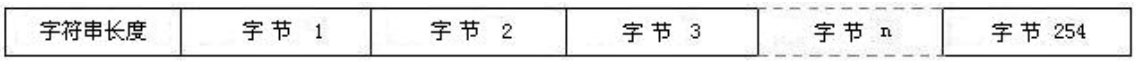
实数至字符串转换指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 非法格式： nnn > 5 ssss < 3 ssss < 要求的字符数
	影响的特殊内存位	无
指令列表： RTS IN, OUT, FMT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, 常数, AC, *VD, *LD, *AC
OUT	字符串	VB, LB, *VD, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



操作数 FMT 定义：

位号	7	6	5	4	3	2	1	0
值	s	s	s	s	c	n	n	n

ssss：输出字符串长度，小于 3 无效。

c：整数与小数部分的分隔符，c = 1:使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符。

nnn：小数部分的位数,有效范围为 0～5，nnn 等于 0 会使转换结果显示格式为不带小数点，当 nnn 值大于 5 时或当指定的输出字符串长度太小无法存储转换的值时，输出字符串用 ASCII 空格字符填充。

下表显示几个使用小数点（c = 0）和小数点右面有一位数（nnn = 001）以及输出字符串长度为 7 个字符（ssss = 0111）格式的值范例。位于 OUT 位置的值是字符串长度。

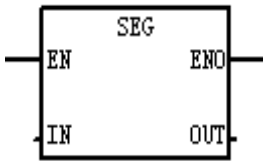
	OUT	**字节 OUT**						
		+1	+2	+3	+4	+5	+6	+7
IN=12345.6	7	1	2	3	4	5	.	6
IN=0.0005	7					0	.	0

IN=-3.6544444	7				-	3	.	7

功能说明：实数至字符串转换指令将实数值 IN 转换为 ASCII 字符串。格式（FMT）指定输出字符串的长度、转换精度以及小数点是显示为逗号还是句点。转换结果放入以 OUT 开始的字符串中。结果字符串长度在格式中指定，可以是 3 至 15 个字符。CTSC-200 使用的实数格式最多可支持 7 个高位数字。设置 7 个以上高位数字会产生取整错误。

提示：根据以下规则制定格式的输出字符串：

- 整数不带符号写入输出缓冲区。
- 负数带起始减号（-）写入输出缓冲区。
- 小数点左面的起始零（除紧靠小数点的数字外）被压缩。
- 小数点右面的数值被取整，使之符合指定的小数点右面的位数。
- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

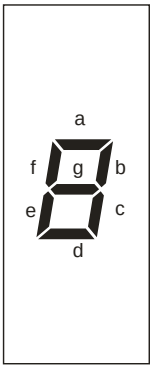
段指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无
指令列表： SEG IN, OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SMB, LB, AC, *VD, *AC, SB, *LD

操作数说明：

下图显示了段指令中用到的七段显示器段编码。

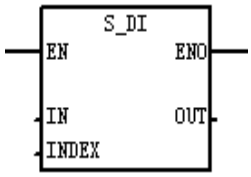
(进) LSD	段显示	(出) -gfe dcba
0	0	0011 1111
1	1	0000 0110
2	2	0101 1011
3	3	0100 1111
4	4	0110 0110
5	5	0110 1101
6	6	0111 1101
7	7	0000 0111



(进) LSD	段显示	(出) -gfe dcba
8	8	0111 1111
9	9	0110 0111
A	A	0111 0111
B	b	0111 1100
C	c	0011 1001
D	d	0101 1110
E	E	0111 1001
F	F	0111 0001

功能说明：段（SEG）指令允许您生成照明七段显示段的位格式。

字符串至双整数转换指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 009B 非法指数(指定起始位置值0的字符串操作) SM1.1 溢出或非法值

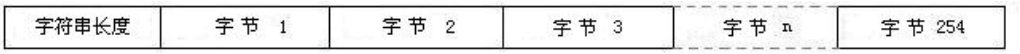
	影响的特殊内存位	SM1.1(溢出或非法值)
指令列表： STD IN, INDEX, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC,*VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



STD 指令用以下形式转换字符串：

[spaces] [+ or -] [digits 0-9]

INDEX 值通常设为 1，从字符串的第一个字符开始转换。可将该值设为其他值，在字符串中的不同点开始转换。当输入字符串包含不属于需要转换数字一部分的文本时，可采用此种方法。例如，如果输入字符串是"value:16000 "，您可以将 INDEX 设为值 7，跳过字符串开始的字"value:"。

当达到字符串结束时或遇到第一个无效字符时，转换终止。无效字符是数字（0-9）以外的任何字符。

每当转换产生一个对于输出值过大的整数值时，则设置溢出错误（SM1.1）。例如，如果输入字符串产生一个大于+2147483647 或小于-+2147483648 的值时，STD 指令设置溢出错误。

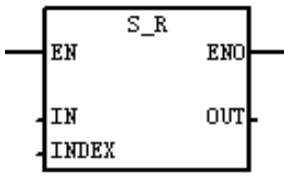
如果当输入字符串未包含有效值而无法执行转换时，也会设置溢出错误（SM1.1）。例如，如果输入字符串包含"A123"，转换指令会设置 SM1.1（溢出），输出值保持不变。

下表是有效和无效字符串转换为整数/双整数/实数时的举例（INDEX = 1）：

输入字符串	字符串有效性	转换为整数	转换为双整数	转换为实数
"65535"	有效	65535	65535	65535.0
"-0032768"	有效	-32768	-32768	-32768.0
"+32767.999 "	有效	32767	32767	32767.0
"00065535FFFF"	有效	65535	65535	65535.0
"65536"	有效	65536(SM1.1=1,溢出)	65536	65536.0
"-002147483648"	有效	-2147483648(SM1.1=1,溢出)	-2147483648	-2147483648.0
"+002147483648"	有效	+2147483648(SM1.1=1,溢出)	+2147483648(SM1.1=1,溢出)	+2147483648.0
"F255"	无效(SM1.1=1,无效)			

"++255"	无效(SM1.1=1, 无效)			
" "	无效(SM1.1=1, 无效)			

功能说明：字符串至双整数转换指令将字符串数值 IN 转换为双整数值，并将转换结果存储在 OUT 指定的地址中，从字符串的第 INDEX 个字符位置开始转换。

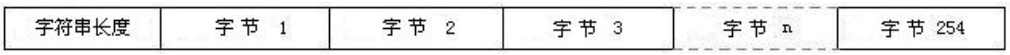
字符串至实数转换指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 009B 非法指数(指定起始位置值0的字符串操作) SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1(溢出或非法值)
指令列表： STR IN, INDEX, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	实数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC,*VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



STI 指令用以下形式转换字符串：

[spaces] [+ or -] [digits 0-9]

INDEX 值通常设为 1，从字符串的第一个字符开始转换。可将该值设为其他值，在字符串中的不同点开始转换。当输入字符串包含不属于需要转换数字一部分的文本时，可采用此种方法。例如，如果输入字符串是"value:16000 "，您可以将 INDEX 设为值 7，跳过字符串开始的字"value:"。

字符串至实数转换指令不会使用实数的科学计数法或指数形式转换字符串。该指令不会产生溢出错误（SM1.1），但只会将字符串转换为直至指数的实数，然后终止转换。例如，字符串"1.234E6"将转换为实数值 1.234，而不产生错误讯息。

当达到字符串结束时或遇到第一个无效字符时，转换终止。无效字符是数字（0-9）以外的任何字符。

每当转换产生一个对于输出值过大的整数值时，则设置溢出错误（SM1.1）。

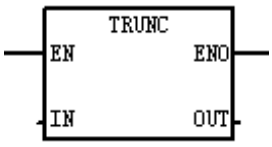
如果当输入字符串未包含有效值而无法执行转换时，也会设置溢出错误（SM1.1）。例如，如果输入字

符串包含"A123"，转换指令会设置 SM1.1（溢出），输出值保持不变。

下表是有效和无效字符串转换为整数/双整数/实数时的举例（INDEX = 1）：

输入字符串	字符串有效性	转换为整数	转换为双整数	转换为实数
"65535"	有效	65535	65535	65535.0
"-0032768"	有效	-32768	-32768	-32768.0
"+32767.999 "	有效	32767	32767	32767.0
"00065535FFFF"	有效	65535	65535	65535.0
"65536"	有效	65536(SM1.1=1, 溢出)	65536	65536.0
"-002147483648"	有效	-2147483648(SM1.1=1,溢出)	-2147483648	-2147483648.0
"+002147483648"	有效	+2147483648(SM1.1=1,溢出)	+2147483648(SM1.1=1,溢出)	+2147483648.0
"F255"	无 效 (SM1.1=1, 无效)			
"++255"	无 效 (SM1.1=1, 无效)			
" "	无 效 (SM1.1=1, 无效)			

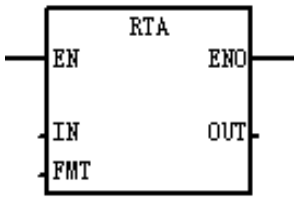
功能说明：字符串至实数转换指令将字符串数值 IN 转换为实数值，并将转换结果存储在 OUT 指定的地址中，从字符串的第 INDEX 个字符位置开始转换。

截断指令	设置 ENO = 0的错误条件	0006 间接地址 SM1.1 溢出或非法值
	影响的特殊内存位	SM1.1（溢出）
指令列表： TRUNC IN, OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, AC, 常数, *VD, *LD, *AC
OUT	双整数	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD

功能说明：截断指令将 32 位实数 IN 转换成 32 位双整数，并将结果放入 OUT 指定的变量中。只有实数的整数部分被转换，小数部分被丢弃。如果输入的实数值过大或无效，将会导致溢出，输出不受影响。

实数至 ASTII 码转换指令	设置 ENO = 0的错误条件	0006 间接地址 nnn > 5 ssss < 3 ssss < OUT 中的字符数
	影响的特殊内存位	无
指令列表： RTA IN, OUT, FMT		
梯形图：		

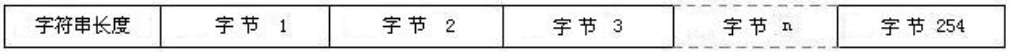


输入输出参数	数据类型	适用操作数
IN	实数	VD, ID, QD, MD, SD, SMD, LD, 常数, AC, *VD, *LD, *AC
OUT	字符串	VB, LB, *VD, *AC, *LD
FMT	字节	VB, IB, QB, MB, SB, SMB, LB, 常数, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



操作数 FMT 定义：

位号	7	6	5	4	3	2	1	0
值	s	s	s	s	c	n	n	n

ssss：输出字符串长度，小于 3 无效。

c：整数与小数部分的分隔符，c = 1:使用逗号作为整数与小数部分的分隔符；c = 0:使用小数点作为整数与小数部分的分隔符。

nnn：小数部分的位数,有效范围为 0～5，nnn 等于 0 会使转换结果显示格式为不带小数点，当 nnn 值大于 5 时或当指定的输出字符串长度太小无法存储转换的值时，输出字符串用 ASCII 空格字符填充。

下表显示几个使用小数点（c = 0）和小数点右面有一位数（nnn = 001）以及输出字符串长度为 7 个字符（ssss = 0111）格式的值范例。位于 OUT 位置的值是字符串长度。

	OUT	字节 OUT					
		+1	+2	+3	+4	+5	+6
IN=12345.6	1	2	3	4	5	.	6
IN=0.0005					0	.	0
IN=-3.6544444				-	3	.	7

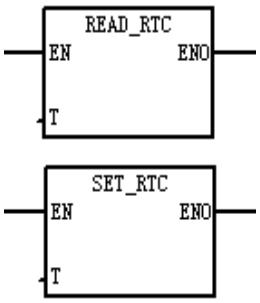
功能说明：实数至 ASCII 码转换指令将实数值 IN 转换为 ASCII 字符。格式（FMT）指定输出字符串的长度、转换精度以及小数点是显示为逗号还是句点。转换结果放入以 OUT 开始的字符串中。结果字符串长度在格式中指定，可以是 3 至 15 个字符。

提示：根据以下规则制定格式的输出字符串：

- 整数不带符号写入输出缓冲区。
- 负数带起始减号（-）写入输出缓冲区。
- 小数点左面的起始零（除紧靠小数点的数字外）被压缩。
- 小数点右面的数值被取整，使之符合指定的小数点右面的位数。

- 输出字符串的大小最小必须比小数点右面的位数大三个字节。
- 输出字符串中的数值必须右对齐。

实时时钟

读取和设置实时时钟指令	设置 ENO = 0的错误条件	0006 间接地址 0007 TOD 数据错误(TODW) 000C 不存在时钟
	影响的特殊内存位	SM4.3（对此时钟有两个同时访问，非重要错误0007）
指令列表： TODR T TODW T		
梯形图：  		

输入输出参数	数据类型	适用操作数
时钟 T	字节	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD

操作数说明：所有日期和时间值必须采用 BCD 格式编码（例如，16#97 代表 2002 年）。请参阅下表：

8 个字节时间缓冲区格式（T）

T 字节	说明	字节数据
0	年 (0-99)	当前年份后两位 (BCD 码)
1	月 (1-12)	当前月份 (BCD 码)
2	日期 (1-31)	当前日期 (BCD 码)
3	小时 (0-23)	当前小时 (BCD 码)
4	分钟 (0-59)	当前分钟 (BCD 码)
5	秒 (0-59)	当前秒 (BCD 码)
6	00	保留字节，始终设置为00
7	星期几 (1-7)	当前是星期几，1=星期日 (BCD 码)

长时间掉电或内存丢失后，实时时钟会被初始化为以下日期和时间：

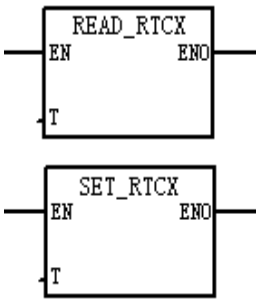
日期： 90 年 1 月 1 日
时间： 00:00:00
星期： 星期日

提示：CTSC-200 CPU 无法根据日期和星期是否正确，所以在输入时应当确保日期正确。请勿在主程序和中断程序中使用 TODR/TODW 指令。当另一条 TODR/TODW 指令正在执行时，若尝试执行中断程序中的 TODR/TODW 指令，则该指令不会被执行，同时 SM4.3 被置位。

功能说明：

读取实时时钟指令（TODR）从硬件时钟读取当前时间和日期，并将其写入以地址 T 起始的 8 个字节的缓冲区。

设置实时时钟指令（TODW）设置当前硬件时钟为以 T 起始的 8 个字节的缓冲区的内容。

<table>
<tr><td rowspan="2">读取和设置扩展实时时钟指令</td><td>设置 ENO = 0的错误条件</td><td>0006 间接地址
0007 TOD 数据错误(TODW)
000C 不存在时钟
0091 范围错误</td></tr>
<tr><td>影响的特殊内存位</td><td>SM4.3（对此时钟有两个同时访问，非重要错误0007）</td></tr>
<tr><td colspan="3">指令列表：
TODRX T
TODWX T</td></tr>
<tr><td colspan="3">梯形图：
 
</td></tr>
<tr><td>输入输出参数</td><td>数据类型</td><td>适用操作数</td></tr>
<tr><td>时钟 T</td><td>字节</td><td>VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD</td></tr>
</table>

操作数说明：所有日期和时间值必须采用 BCD 格式编码（例如，16#97 代表 2002 年）。请参阅下表：

8 个字节时间缓冲区格式（T）

T 字节	说明	字节数据
0	年 (0-99)	当前年份后两位 (BCD 码)
1	月 (1-12)	当前月份 (BCD 码)
2	日期 (1-31)	当前日期 (BCD 码)
3	小时 (0-23)	当前小时 (BCD 码)
4	分钟 (0-59)	当前分钟 (BCD 码)
5	秒 (0-59)	当前秒 (BCD 码)
6	00	保留字节，始终设置为00
7	星期几 (1-7)	当前是星期几，1=星期日 (BCD 码)
8	模式(00H-03H, 08H, 10H-13H，FFH)	修正模式： 00H = 修正已禁用 01H = 欧盟 (相对于 UTC 的时区调整 = 0小时) 02H = 欧盟 (相对于 UTC 的时区调整 = +1小时) 03H = 欧盟 (相对于 UTC 的时区调整 = +2小时) 04H-07H = 保留 08H = 欧盟 (相对于 UTC 的时区调整 = -1小时) 09H-0FH = 保留 10H = 美国 11H = 澳大利亚 12H = 澳大利亚 (塔斯马尼亚)

		13H = 新西兰 14H-FDH = 保留 FEH = 保留
9	修正小时数 (0-23)	修正数量，小时 (BCD 码)
10	修正分钟数 (0-59)	修正数量，分钟 (BCD 码)
11	开始月份 (1-12)	夏时制的开始月份 (BCD 码)
12	开始日期 (1-31)	夏时制的开始日期 (BCD 码)
13	开始小时 (0-23)	夏时制的开始小时 (BCD 码)
14	开始分钟 (0-59)	夏时制的开始分钟 (BCD 码)
15	结束月份 (1-12)	夏时制的结束月份 (BCD 码)
16	结束日期 (1-31)	夏时制的结束日期 (BCD 码)
17	结束小时 (0-23)	夏时制的结束小时 (BCD 码)
18	结束分钟 (0-59)	夏时制的结束分钟 (BCD 码)

欧盟常规：在三月最后一个星期日的 UTC 时间凌晨一点将时间向前调一小时。在十月最后一个星期日的 UTC 时间凌晨两点将时间往回调一小时。(当做出修正时，当地时间因相对于 UTC 的时区调整而不同)

美国常规：在四月第一个星期日的当地时间凌晨两点将时间向前调一小时。在十月最后一个星期日的当地时间凌晨两点将时间往回调一小时。

澳大利亚常规：在十月最后一个星期日的当地时间凌晨两点将时间向前调一小时。在三月最后一个星期日的当地时间凌晨三点将时间往回调一小时。

澳大利亚(塔斯马尼亚)常规：在十月第一个星期日的当地时间凌晨两点将时间向前调一小时。在三月最后一个星期日的当地时间凌晨三点将时间往回调一小时。

新西兰常规：在十月第一个星期日的当地时间凌晨两点将时间向前调一小时。在三月十五日或之后的第一个星期日的当地时间凌晨三点将时间往回调一小时。

长时间掉电或内存丢失后，实时时钟会被初始化为以下日期和时间：

日期：90 年 1 月 1 日
时间：00:00:00
星期：星期日

提示：CTSC-200 CPU 无法根据日期和星期是否正确，所以在输入时应当确保日期正确。请勿在主程序和中断程序中使用 TODRX/TODWX 指令。当另一条 TODRX/TODWX 指令正在执行时，若尝试执行中断程序中的 TODRX/TODWX 指令，则该指令不会被执行，同时 SM4.3 被置位。

功能说明：

读取扩展实时时钟指令（TODRX）从硬件时钟读取当前时间、日期及夏时制配置，并将其写入以地址 T 起始的 19 个字节的缓冲区。

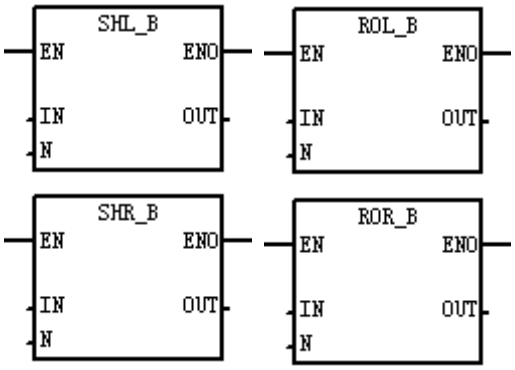
设置扩展实时时钟指令（TODWX）设置当前硬件时钟为以 T 起始的 19 个字节的缓冲区的内容。

移位/循环指令

字节移位/循环指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 如果移位结果是零，则设置零位 SM1.1 为移出的最后一个位设置溢出位
指令列表：		

SLB OUT, N
SRB OUT, N
RLB OUT, N
RRB OUT, N

梯形图：



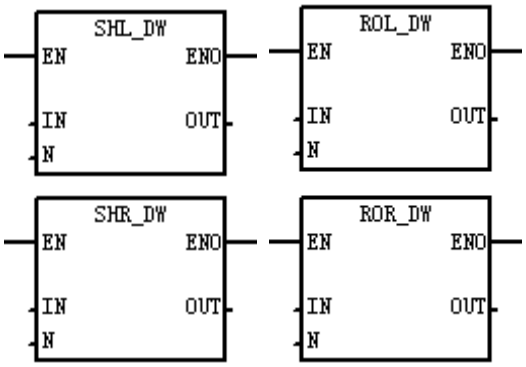
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

功能说明：

右移字节（SRB）和左移字节（SLB）指令将输入数值（IN）根据移位计数（N）向右或向左移动，并将结果载入输出字节（OUT）。移位指令对每个移出位补 0。如果移位数目（N）大于或等于 8，则数值最多被移位 8 次。如果移位数目大于 0，溢出内存位（SM1.1）采用最后一次移出位的数值。如果移位操作结果为 0，设置 0 内存位（SM1.0）。右移和向左移字节操作不带符号。

循环右移字节（RRB）和循环左移字节（RLB）指令将输入字节数值（IN）向右或向左旋转 N 位，并将结果载入输出字节（OUT）。旋转具有循环性。如果移位数目（N）大于或等于 8，执行旋转之前先对位数（N）进行模数 8 操作，从而使位数在 0 至 7 之间。如果移动位数为 0，则不执行旋转操作。如果执行旋转操作，旋转的最后一位数值被复制至溢出位（SM1.1）。如果移动位数不是 8 的整倍数，旋转出的最后一位数值被复制至溢出内存位（SM1.1）。如果旋转数值为 0，设置 0 内存位（SM1.0）。循环右移和循环左移字节操作不带符号。

双字移位/循环指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 如果移位结果是零，则设置零位 SM1.1 为移出的最后一个位设置溢出位
指令列表： SLD OUT, N SRD OUT, N RLD OUT, N RRD OUT, N		
梯形图：		



输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SD, SMD, LD, AC, HC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SMB, SB, LB, AC, 常数, *VD, *LD, *AC
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC

功能说明：

右移双字（SRD）和左移双字（SLD）指令将输入双字数值（IN）向右或向左移动 N 位，并将结果载入输出双字（OUT）。移位指令对每个移出位补 0。如果移位数目（N）大于或等于 32，则数值最多被移位 32 次。如果移位数目大于 0，溢出内存位（SM1.1）采用最后一次移出位数值。如果移位操作结果为 0，设置 0 内存位（SM1.0）。请注意当您使用带符号数据类型时，符号位被移位。

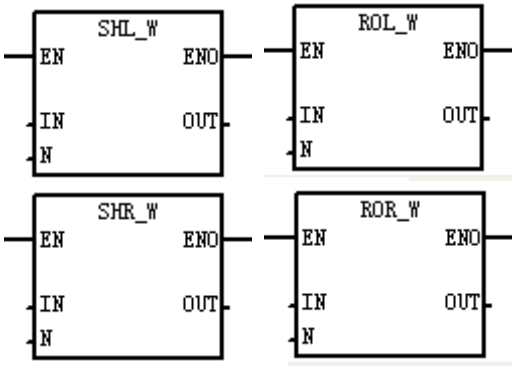
循环右移双字（RRD）和循环左移双字（RLD）指令将输入双字数值（IN）向右或向左旋转 N 位，并将结果载入输出双字（OUT）。旋转具有循环性。如果移位数目（N）大于或等于 32，执行旋转之前在移动位数（N）上执行模数 32 操作。从而使位数在 0 至 31 之间。如果移动位数为 0，则不执行旋转操作。如果执行旋转操作，旋转的最后一位数值被复制至溢出位（SM1.1）。如果移动位数不是 32 的整倍数，旋转出的最后一位数值被复制至溢出内存位（SM1.1）。如果旋转数值为 0，设置 0 内存位（SM1.0）。循环右移和循环左移双字操作不带符号。

字移位/循环指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 如果移位结果是零，则设置零位 SM1.1 为移出的最后一个位设置溢出位

指令列表：

```
SLW OUT, N
SRW OUT, N
RLW OUT, N
RRW OUT, N
```

梯形图：



输入输出参数	数据类型	适用操作数

IN	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, Constant, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SMB, SB, LB, AC，常数, *VD, *LD, *AC
OUT	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC

功能说明：

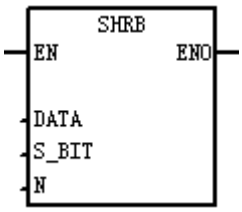
右移字（SRW）和左移位字（SLW）指令将输入字（IN）数值向右或向左移动 N 位，并将结果载入输出字（OUT）。移位指令对每个移出位补 0。如果移位数目（N）大于或等于 16，则数值最多被移位 16 次。如果移位数目大于 0，溢出内存位（SM1.1）采用最后一次移出位数值。如果移位操作结果为 0，设置 0 内存位（SM1.0）。请注意当您使用带符号的数据类型时，符号位被移位。

循环右移字（RRW）和循环左移字（RLW）指令将输入字数值（IN）向右或向左旋转 N 位，并将结果载入输出字（OUT）。旋转具有循环性。如果移动位数（N）大于或等于 16，在旋转执行之前的移动位数（N）上执行模数 16 操作。从而使移动位数在 0 至 15 之间。如果移动位数为 0，则不执行旋转操作。如果执行旋转操作，旋转的最后一位数值被复制至溢出位（SM1.1）。如果移动位数不是 16 的整倍数，旋转出的最后一位数值被复制至溢出内存位（SM1.1）。如果旋转数值为 0，设置 0 内存位（SM1.0）。循环右移和循环左移字操作不带符号。

移位寄存器指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围 0092 计数域错误
	影响的特殊内存位	SM1.1 为移出的最后一个位设置溢出位

指令列表： SHRB DATA, S_BIT, N

梯形图：



输入输出参数	数据类型	适用操作数
DATA	布尔	I, Q, M, SM, T, C, V, S, L
S_BIT	布尔	I, Q, M, SM, T, C, V, S, L
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC，常数, *VD, *LD, *AC

功能说明：

移位寄存器位（SHRB）指令将 DATA 数值移入移位寄存器。S_BIT 指定移位寄存器的最低位。N 指定移位寄存器的长度和移位方向（移位加 = N，移位减 = -N）。SHRB 指令移出的每个位被放置在溢出内存位（SM1.1）中。该指令由最低位（S_BIT）和由长度（N）指定的位数定义。

使用以下等式计算"移位寄存器"最高位地址（MSB.b）：

MSB.b = [(S_BIT 字节) + ([N] - 1 + (S_BIT 位)) / 8] . [被 8 除的余数]

例如：如果 S_BIT 是 V33.4 和 N is 14，以下计算显示 MSB.b 是 V35.1。

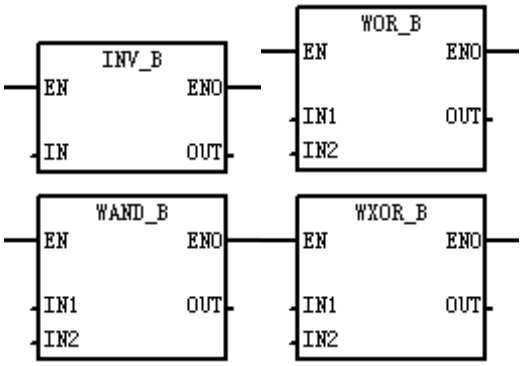
MSB.b = V33 + ([14] - 1 +4)/8
= V33 + 17/8
= V33 + 2 ，余数为 1
= V35.1

在"移位减"（用长度（N）的负值表示）中，输入数据移入移位寄存器的最高位中，并移出最低位（S_BIT）。移出的数据被放置在溢出内存位（SM1.1）中。

在"移位加"（用长度（N）的正值表示）中，输入数据（DATA）移入移位寄存器的最高位中（由 S_BIT 指定），并移出移位寄存器的最高位。移出的数据被放置在溢出内存位（SM1.1）中。

移位寄存器的最大长度为 64 位（无论正负）。

逻辑计算指令

字节取反、与、或、异或运算指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 操作结果等于零
指令列表： INVB OUT ANDB IN1 OUT ORB IN1 OUT XORB IN1 OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN1	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
IN2	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *AC, *LD
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

功能说明：

字节取反指令（INVB）对输入字节 IN 执行求补操作，并将结果放入 OUT 指定的地址中。

字节与运算指令（ANDB）对两个输入字节（IN1 和 IN2）的对应位执行与运算操作，并将结果放入 OUT 指定的地址中。

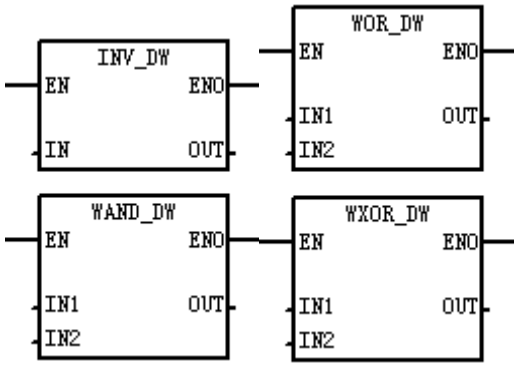
字节或运算指令（ORB）对两个输入字节（IN1 和 IN2）的对应位执行或运算操作，并将结果放入 OUT 指定的地址中。

字节异或运算指令（XORB）对两个输入字节（IN1 和 IN2）的对应位执行异或运算操作，并将结果放入 OUT 指定的地址中。

双字取反、与、或、异或运算指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 操作结果等于零
指令列表：		

INVD OUT
ANDD IN1, OUT
ORD IN1, OUT
XORD IN1, OUT

梯形图：



输入输出参数	数据类型	适用操作数
IN1	双字	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *AC, *LD
IN2	双字	VD, ID, QD, MD, SD, SMD, LD, HC, AC, 常数, *VD, *AC, *LD
OUT	双字	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD

功能说明：

双字取反指令（INVD）对输入双字 IN 执行求补操作，并将结果放入 OUT 指定的地址中。

双字与运算指令（ANDD）对两个输入双字（IN1 和 IN2）的对应位执行与运算操作，并将结果放入 OUT 指定的地址中。

双字或运算指令（ORD）对两个输入双字（IN1 和 IN2）的对应位执行或运算操作，并将结果放入 OUT 指定的地址中。

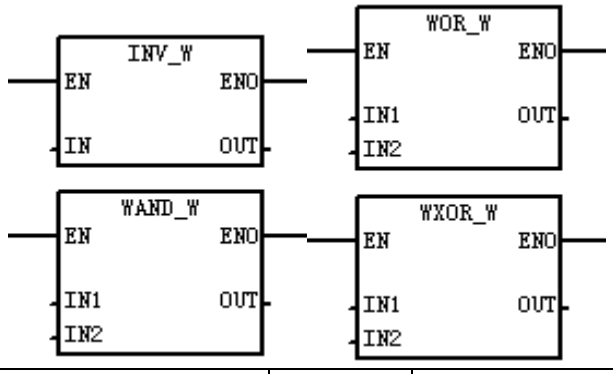
双字异或运算指令（XORD）对两个输入双字（IN1 和 IN2）的对应位执行异或运算操作，并将结果放入 OUT 指定的地址中。

整数取反、与、或、异或运算指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SM1.0 操作结果等于零

指令列表：

INVW OUT
ANDW IN1, OUT
ORW IN1, OUT
XORW IN1, OUT

梯形图：



输入输出参数	数据类型	适用操作数

IN1	字	VW, IW, QW, MW, SW, SMW, T, C, AIW, LW, AC, 常数, *VD, *AC, *LD
IN2	字	VW, IW, QW, MW, SW, SMW, T, C, AIW, LW, AC, 常数, *VD, *AC, *LD
OUT	字	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD

功能说明：

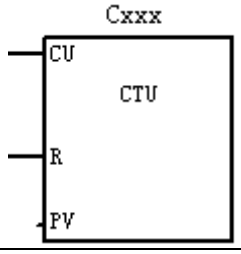
字取反指令（INVW）对输入字 IN 执行求补操作，并将结果放入 OUT 指定的地址中。

字与运算指令（ANDW）对两个输入字（IN1 和 IN2）的对应位执行与运算操作，并将结果放入 OUT 指定的地址中。

字或运算指令（ORW）对两个输入字（IN1 和 IN2）的对应位执行或运算操作，并将结果放入 OUT 指定的地址中。

字异或运算指令（XORW）对两个输入字（IN1 和 IN2）的对应位执行异或运算操作，并将结果放入 OUT 指定的地址中。

计数器指令

加计数器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： CTU Cxxx, PV		
梯形图：  		

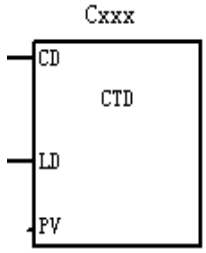
输入输出参数	数据类型	适用操作数
Cxxx	字	常数 (C0～C255)
CU	布尔	使能位
R	布尔	使能位
PV	整数	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, 常数, *VD, *AC, *LD, SW

功能说明：每次加计数输入端 CU 从 OFF 变为 ON 时，加计数器指令 CTU 从当前值计数加 1。当前值（Cxxx）大于或等于预设值（PV）时，计数器位（Cxxx）打开。复原（R）输入打开或执行"复原"指令时，计数器被复原。达到最大值（32767）时，计数器停止计数。计数器范围： Cxxx=C0 至 C255 在 STL 中，CTU 复原输入是堆栈顶值，加计数输入是装载在第二个堆栈位置的值。

<注释> 因为每个计数器有一个当前值，请勿将相同的计数器号码设置给一个以上计数器。（号码相同的加计数器、加 / 减计数器和减计数器存取相同的当前值。）

减计数器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： CTD Cxxx, PV		

梯形图：



输入输出参数	数据类型	适用操作数
Cxxx	字	常数 (C0～C255)
CD	布尔	使能位
R	布尔	使能位
PV	整数	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, 常数, *VD, *AC, *LD, SW

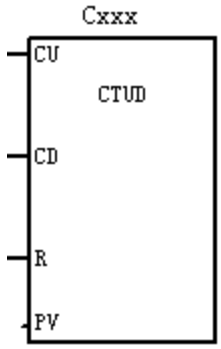
功能说明：每次减计数输入端 CD 从 OFF 变为 ON 时，减计数器指令 CTD 从当前值计数减 1。当前值 Cxxx 等于 0 时，计数器位（Cxxx）打开。载入输入（LD）打开时，计数器复原计数器位（Cxxx）并用预设值（PV）载入当前值。当前值达到零时，减计数器停止计数，计数器位 Cxxx 打开。计数器范围：Cxxx=C0 至 C255 在 STL 中，CTD 载入输入是堆栈顶值，而减计数输入是装载在第二个堆栈位置的数值。

<注释> 因为每个计数器有一个当前值，请勿将相同的计数器号码设置给一个以上计数器。（号码相同的加计数器、加 / 减计数器和减计数器存取相同的当前值）

加减双向计数器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： CTUD Cxxx, PV

梯形图：

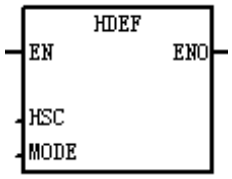


输入输出参数	数据类型	适用操作数
Cxxx	字	常数 (C0～C255)
CU	布尔	使能位
CD	布尔	使能位
R	布尔	使能位
PV	整数	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, 常数, *VD, *AC, *LD, SW

功能说明：每次加计数输入端 CU 从 OFF 变为 ON 时，加减双向计数器指令 CTUD 从当前值计数加 1。每次减计数输入端 CD 从 OFF 变为 ON 时，加减双向计数器指令 CTUD 从当前值计数减 1。计数器的当前值 Cxxx 保持当前计数。每次执行计数器指令时，预设值 PV 与当前值进行比较。达到最大值

（32767），位于加计数输入位置的下一个上升沿使当前值返转为最小值（-32,768）。在达到最小值（-32,768）时，位于减计数输入位置的下一个上升沿使当前计数返转为最大值（32,767）。当当前值 Cxxx 大于或等于预设值 PV 时，计数器位 Cxxx 打开。否则，计数器位关闭。当"复原"（R）输入打开或执行"复原"指令时，计数器被复原。达到 PV 时，CTUD 计数器停止计数。计数器范围：Cxxx=C0 至 C255 在 STL 中，CTUD 复原输入是堆栈顶值，减计数输入是装载在第二个堆栈位置的值，加计数输入是装载在第三个堆栈位置的值。

<注释> 因为每个计数器有一个当前值，请勿将相同的计数器号码设置给一个以上计数器。（号码相同的加计数器、加 / 减计数器和减计数器存取相同的当前值。）

高速计数器定义指令	设置 ENO = 0的错误条件	0003 输入点冲突 0004 中断中的非法指令 000A HSC 重新定义
	影响的特殊内存位	无
指令列表： HDEF HSC, MODE		
梯形图：  		

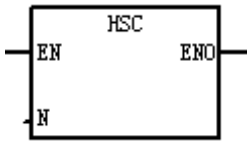
输入输出参数	数据类型	适用操作数
HSC	字节	常数（0,1,2,3,4或5）
MODE	字节	常数（0,1,2,3,4,5,6,7,8,9,10或11）

操作数说明：

HSC：高速计数器的 ID，目前 CPU224 和 CPU226 都支持 6 个高速计数器，因此 HSC 的范围为常数 0～5.

MODE：高速计数器的模式，高速计数器有 12 种模式，MODE 的范围为常数 0～11.

功能说明：高速计数器定义指令 HDEF 选择特定的高速计数器（HSCx）的操作模式。模式选择定义高速计数器的时钟、方向、起始和复原功能。您可以为每台高速计数器使用一条"高速计数器定义"指令。

高速计数器指令	设置 ENO = 0的错误条件	0001 HSC 在 HDEF 之前 0005 HSC/PLS 同步
	影响的特殊内存位	无
指令列表： HSC N		
梯形图：  		

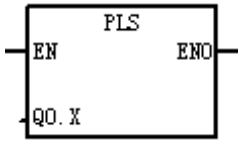
输入输出参数	数据类型	适用操作数
N	字	常数（0,1,2,3,4或5）

操作数说明：

N：高速计数器的 ID，目前 CPU224 和 CPU226 都支持 6 个高速计数器，因此 N 的范围为常数 0～5.

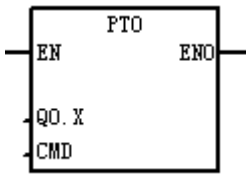
功能说明：高速计数器指令 HSC 根据 HSC 特殊内存位的状态配置和控制高速计数器。参数 N 指定高速计数器的号码。高速计数器最多可配置为十二种不同的操作模式。每台计数器在功能受支持的位置有专用时钟、方向控制、复原和起始输入。对于双相计数器，两个时钟均可按最高速度运行。在正交模式

中，您可以选择一倍\（1x）或四倍（4x）的最高计数速率。所有的计数器按最高速率运行，而不会相互干扰。

脉冲输出指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： PLS N		
梯形图：  		

输入输出参数	数据类型	适用操作数
N	字	常数0（Q0.0输出脉冲）或1(Q0.1输出脉冲)

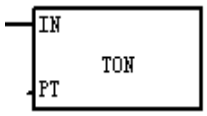
功能说明：脉冲输出（PLS）指令被用于控制在高速输入（Q0.0 和 Q0.1）中提供的"脉冲串输出"（PTO）和 "脉宽调制"（PWM）功能。PTO 提供方波（50%占空比）输出，配备周期和脉冲数用户控制功能。PWM 提供连续性变量占空比输出，配备周期和脉宽用户控制功能。

脉冲输出指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： PTO N, CMD		
梯形图：  		

输入输出参数	数据类型	适用操作数
N	字节	常数0（Q0.0输出脉冲）或1(Q0.1输出脉冲)
CMD	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

功能说明：脉冲输出（PLS）指令被用于控制在高速输入（Q0.0 和 Q0.1）中提供的"脉冲串输出"（PTO）和"脉宽调制"（PWM）功能。PTO 提供方波（50%占空比）输出，配备周期和脉冲数用户控制功能。PWM 提供连续性变量占空比输出，配备周期和脉宽用户控制功能。

定时器指令

接通延时定时器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： TON Txxx, PT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	布尔	使能位

Txxx	字	常数(0～255)
PT	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

操作数说明：

定时器类型	分辨率	最大值	定时器 ID
TON/TOF	1ms	32767*1ms	T32, T96
	10ms	32767*10ms	T33-T36, T97-T100
	100ms	32767*100ms	T37-T63, T101-T255
TONR	1ms	32767*1ms	T0, T64
	10ms	32767*10ms	T1-T4, T65-T68
	100ms	32767*100ms	T5-T31, T69-T95

功能说明：接通延时定时器（TON）指令在启用输入为“打开”时，开始计时。当前值（Txxx）大于或等于预设时间（PT）时，定时器位为“打开”。启用输入为“关闭”时，接通延时定时器当前值被清除。达到预设值后，定时器仍继续计时，达到最大值 32767 时，停止计时。TON、TONR 和 TOF 定时器有三种分辨率。分辨率由定时器 ID 决定，详细情况见操作数说明里的表格。

LAD 和 FBD 定时器选择

1）点击定时器号码域，然后键入定时器号码。

2）倘若您键入的定时器号码无效，则时间基准值继续为"???"。

3）将光标放在定时器框内稍等片刻，即可看到定时器工具提示。请查看此类定时器的有效号码列表。

4）一旦键入有效定时器号码，时间基准值就会在定时器框内显示，例如"10 ms"。

<注释>

▪ TOF 及 TON 不能共享相同的定时器号码。例如，不能有 TON T32 和 TOF T32。

▪ 您可以将 TON 用于单间隔计时。

▪ 可用"复原"（R）指令复原任何定时器。"复原"指令执行下列操作：

▪ 定时器位 = 关闭，定时器当前值 = 0

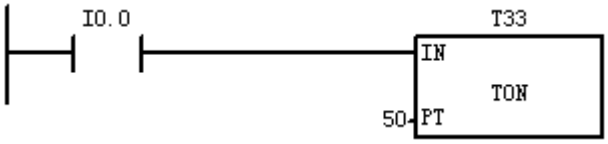
示例程序：

STL：

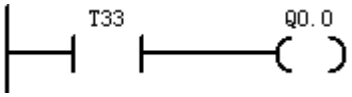
```
ORGANIZATION_BLOCK 主程序：OB1

Network 1

LD          I0.0
TON         T33, +50      // 如果 I0.0 为打开状态，在 50*10ms 后定时器 T33 超时，定时器的位为 1
                          // I0.0 打开 T33 启动开始计时，I0.0 关闭 T33 复位并且停止计时

Network 2

LD          T33           // 如果 T33 位打开，则点亮 Q0.0，反之关闭 Q0.0
=           Q0.0
```

梯形图：

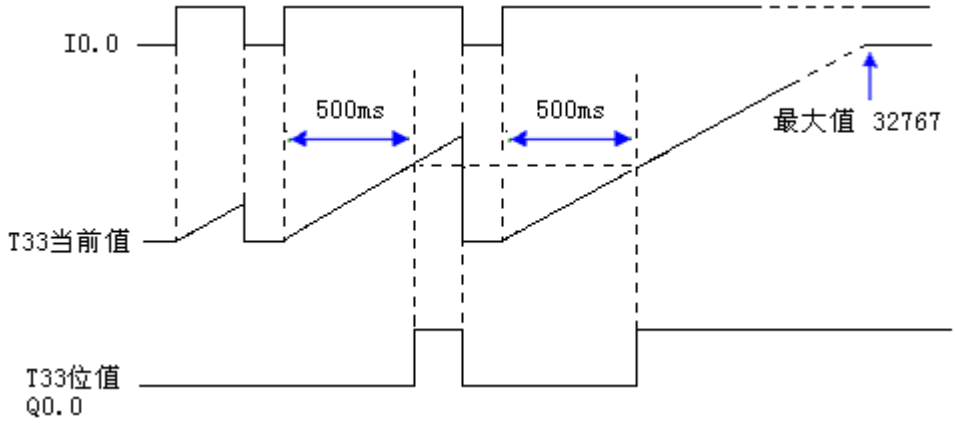
```
ORGANIZATION_BLOCK 主程序：OB1

Network 1
```

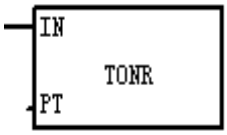


Network 2



时序图：



掉电保持接通延时定时器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： TONR Txxx, PT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	布尔	使能位
Txxx	字	常数(0～255)
PT	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

操作数说明：

定时器类型	分辨率	最大值	定时器 ID
TON/TOF	1ms	32767*1ms	T32, T96
	10ms	32767*10ms	T33-T36, T97-T100
	100ms	32767*100ms	T37-T63, T101-T255
TONR	1ms	32767*1ms	T0, T64
	10ms	32767*10ms	T1-T4, T65-T68
	100ms	32767*100ms	T5-T31, T69-T95

功能说明：掉电保护性接通延时定时器（TONR）指令在启用输入为"打开"时，开始计时。当前值（Txxx）大于或等于预设时间（PT）时，计时位为"打开"。当输入为"关闭"时，保持保留性延迟定时器当前值。您

可使用保留性接通延时定时器为多个输入"打开"阶段累计时间。使用"复原"指令（R）清除保留性延迟定时器的当前值。达到预设值后，定时器继续计时，达到最大值 32767 时，停止计时。TON、TONR 和 TOF 定时器有三种分辨率。分辨率由定时器 ID 决定，详细情况见操作数说明里的表格。

LAD 和 FBD 定时器选择

1）点击定时器号码域，然后键入定时器号码。

2）倘若您键入的定时器号码无效，则时间基准值继续为"???"。

3）将光标放在定时器框内稍等片刻，即可看到定时器工具提示。请查看此类定时器的有效号码列表。

4）一旦键入有效定时器号码，时间基准值就会在定时器框内显示，例如 "10 ms"。

<注释>

您可以将 TONR 用于累积多个计时间隔。

可用"复原"（R）指令复原任何定时器。"复原"指令执行下列操作：

定时器位 = 关闭，定时器当前值 = 0

只能用"复原"指令复原 TONR 定时器。

示例程序：

STL：

ORGANIZATION_BLOCK 主程序：OB1

Network 1

```
LD      I0.0
TONR    T0, +500    // 如果 I0.0 为打开状态，在 500*1ms 后定时器 T0 超时，定时器的位为 1
                    // I0.0 打开 T0 启动开始计时，I0.0 关闭定时器当前值保持，下次 I0.0 打开累加
```

Network 2

```
LD      T0          // 如果 T0 位打开，则点亮 Q0.0，反之关闭 Q0.0
=       Q0.0
```

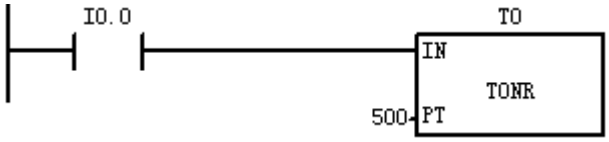
Network 3

```
LD      I1.0        // I1.0 打开时 T0 复位
R       T0, 1
```

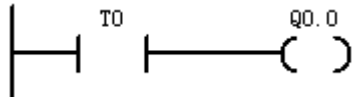
梯形图：

ORGANIZATION_BLOCK 主程序：OB1

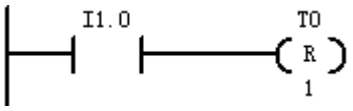
Network 1



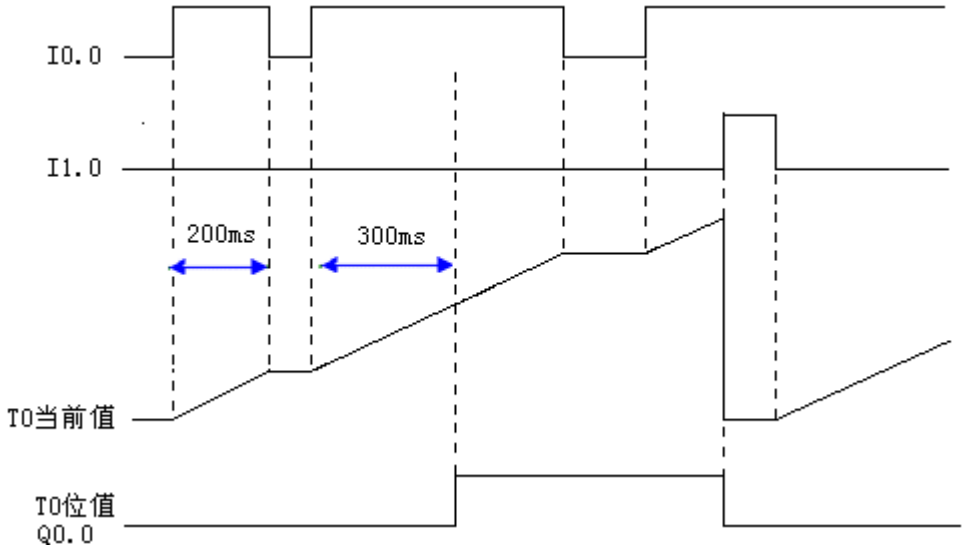
Network 2

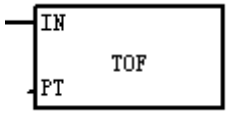


Network 3



时序图：



断开延时定时器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： TOF Txxx, PT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	布尔	使能位
Txxx	字	常数(T0～T255)
PT	整数	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, 常数, *VD, *LD, *AC

操作数说明：

定时器类型	分辨率	最大值	定时器 ID
TON/TOF	1ms	32767*1ms	T32, T96
	10ms	32767*10ms	T33-T36, T97-T100
	100ms	32767*100ms	T37-T63, T101-T255
TONR	1ms	32767*1ms	T0, T64
	10ms	32767*10ms	T1-T4, T65-T68
	100ms	32767*100ms	T5-T31, T69-T95

功能说明：断开延时定时器（TOF）用于在输入关闭后，延迟固定的一段时间再关闭输出。启用输入打开时，定时器位立即打开，当前值被设为 0。输入关闭时，定时器继续计时，直到消逝的时间达到预设时间。达到预设值后，定时器位关闭，当前值停止计时。如果输入关闭的时间短于预设数值，则定时器位仍保持在打开状态。TOF 指令必须遇到从"打开"至"关闭"的转换才开始计时。如果 TOF 定时器位于 SCR 区域内部，而且 SCR 区域处于非现用状态，则当前值被设为 0，计时器位被关闭，而且当前值不计时。TON、TONR 和 TOF 定时器有三种分辨率。分辨率由定时器 ID 决定，详细情况见操作数说明里的表格。

LAD 和 FBD 定时器选择

1）点击定时器号码域，然后键入定时器号码。

2）倘若您键入的定时器号码无效，则时间基准值继续为"???"。

3）将光标放在定时器框内稍等片刻，即可看到定时器工具提示。请查看此类定时器的有效号码列表。

4）一旦键入有效定时器号码，时间基准值就会在定时器框内显示，例如"10 ms"。

<注释>

TOF 及 TON 不能共享相同的定时器号码。例如，不能有 TON T32 和 TOF T32。

您可以将 TOF 用于延长时间以超过关闭（或假）条件，例如在电机关闭后使电机冷却。

可用"复原"（R）指令复原任何定时器。"复原"指令执行下列操作：

定时器位 = 关闭，定时器当前值 = 0

复原后，TOF 定时器要求启用输入从"打开"转换为"关闭"，以便重新启动。

示例程序：

STL：

ORGANIZATION_BLOCK 主程序：OB1

Network 1

```
LD          I0.0
TOF         T32, +500     // 如果 I0.0 为打开状态，在 500*1ms 后定时器 T32 超时，定时器的位为 0
                          // I0.0 下降沿 T32 启动开始计时，I0.0 上升沿 T32 复位并且停止计时
```

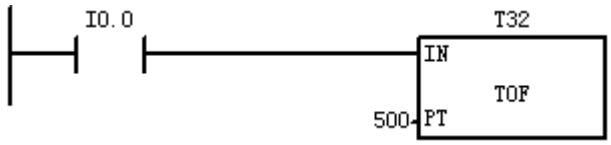
Network 2

```
LD          T32           // 如果 T32 位打开，则点亮 Q0.0，反之关闭 Q0.0
=           Q0.0
```

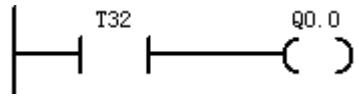
梯形图：

ORGANIZATION_BLOCK 主程序：OB1

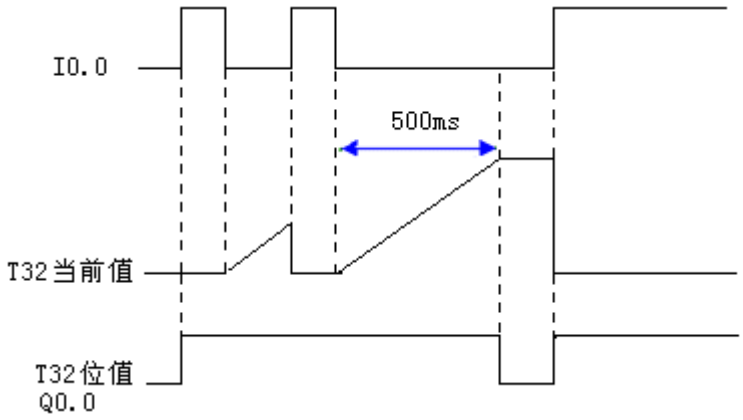
Network 1

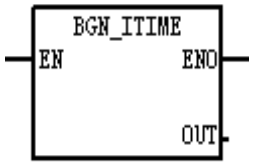


Network 2



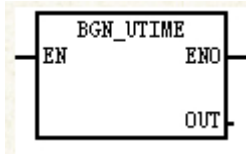
时序图：



开始间隔时间指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： BITIM OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
OUT	双字	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：读取内置 1 毫秒计数器的当前值，并将该值存储于 OUT。双字毫秒值的最大计时间隔为 2 的 32 次方，即 49.7 日。

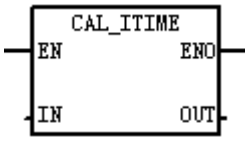
取当前微秒值指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表：R_UTIM OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
OUT	双字	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：读取内置 1 毫秒计数器的当前值，并将该值存储于 OUT，单位为毫秒，双字毫秒值的最大计时间隔为 2 的 32 次方，即 49.7 日。

使用示例：R_UTIM VD0

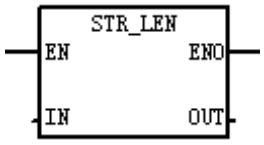
计算间隔时间指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： CITIM IN, OUT		
梯形图：		



输入输出参数	数据类型	适用操作数
IN	双字	VD, ID, QD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC
OUT	双字	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

功能说明：计算当前时间与 IN 所提供时间的时差，将该时差存储于 OUT。双字毫秒值的最大计时间隔为 2 的 32 次方，即 49.7 日。取决于 BGN_ITIME 指令的执行时间，CAL_ITIME 指令将自动处理发生在最大间隔内的一毫秒定时器翻转。

字符串指令

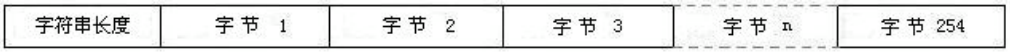
字符串长度指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围
	影响的特殊内存位	无
指令列表： SLEN IN, OUT		
梯形图： 		

输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

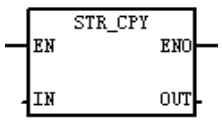
ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



功能说明：字符串长度指令返回 IN 指定的字符串长度。常数字符串参数最长为 255 个字节。

复制字符串指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围
	影响的特殊内存位	无
指令列表： SCPY IN, OUT		
梯形图： 		

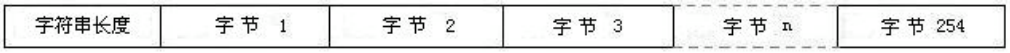
输入输出参数	数据类型	适用操作数

IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：

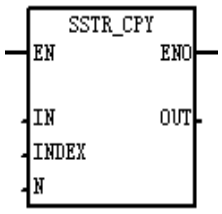


功能说明：复制字符串指令将 IN 指定的字符串复制至 OUT 指定的字符串。常数字符串参数最长为 255 个字节。

从字符串复制子字符串指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 009B 非法索引（指定起始位置0的字符串操作）
	影响的特殊内存位	无

指令列表： SSCPY IN, INDEX, N, OUT

梯形图：

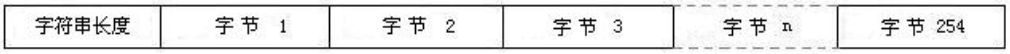


输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
INDEX	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *LD, *AC

ASCII 常数字符串数据类型的格式：

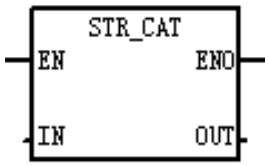
字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



功能说明：从字符串复制子字符串指令将（从索引（INDEX）开始）IN 指定的具体字符串数目复制至 OUT 指定的字符串。

字符串连接指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围

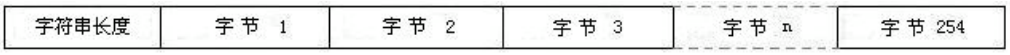
	影响的特殊内存位	无
指令列表： SCAT IN, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字符串	VB, *VD, LB, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束 ("字符串常数")。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式(一个方格代表一个字节)：



功能说明：字符串连接指令将 IN 指定的字符串附加至 OUT 指定的字符串之后。常数字符串参数最长为 255 个字节。

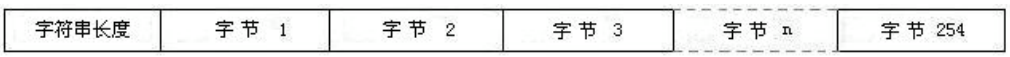
查找字符串指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 009B 非法索引（指定起始位置0的字符串操作）
	影响的特殊内存位	无
指令列表： SFND IN1, IN2, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN1	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
IN2	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

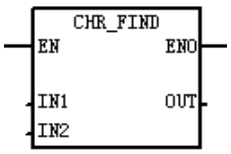
ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



功能说明：在字符串内查找字符串指令在字符串 IN1 中搜索首次出现的字符串 IN2。搜索从 OUT 起始位置开始。如果找到一个与字符串 IN2 完全符合的字符系列，该系列的第一个字符位置被写入 OUT。如果在字符串 IN1 中未找到字符串 IN2，OUT 被设为 0。单个常数字符串最长为 126 个字节，两个常数字符串综合最长为 240 个字节。

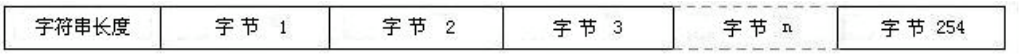
在字符串内查找字符指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数范围 009B 非法索引（指定起始位置0的字符串操作）
	影响的特殊内存位	无
指令列表： CFND IN1, IN2, OUT		
梯形图：  		

输入输出参数	数据类型	适用操作数
IN1	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
IN2	字符串	VB, 常数字符串, LB, *VD, *LD, *AC
OUT	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC

ASCII 常数字符串数据类型的格式：

字符串是一系列字符和对应的内存地址，每个字符作为一个字节存储。字符串的第一个字节是定义字符串长度（即字符数）的整数。如果常数字符串被直接输入程序编辑器或数据块，那么该字符串必须用双引号字符起始和结束（"字符串常数"）。

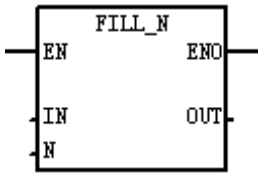
字符串的最大长度是 255 个字节（254 个字符加上长度字节）。下面的内存图显示了字符串数据类型的格式（一个方格代表一个字节）：



功能说明：在字符串中查找字符指令在字符串 IN1 中搜索字符串 IN2 中描述的字符集中的任何字符首次出现的位置。搜索从起始位置 OUT 开始。如果找到一个相符的字符，该字符位置被写入 OUT。如果未找到相符的字符，OUT 被设为 0。 单个常数字符串最长为 126 个字节，两个常数字符串综合最长为 240 个字节。

表指令

内存填充指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无
指令列表： FILL IN, OUT, N		
梯形图：		



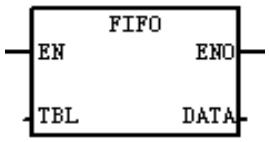
输入输出参数	数据类型	适用操作数
IN	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
N	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
OUT	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC

功能说明：内存填充（FILL）指令用包含在地址 IN 中的字值写入 N 个连续字，从地址 OUT 开始。N 的范围是 1 至 255。

先进先出指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围 SM1.5 空表
	影响的特殊内存位	SM1.5 空表

指令列表： FIFO TBL, DATA

梯形图：



输入输出参数	数据类型	适用操作数
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC
DATA	整数	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *LD, *AC

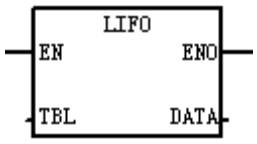
功能说明：先入先出（FIFO）指令通过移除表格（TBL）中的第一个条目，并将数值移至 DATA 指定位置的方法，移动表格中的最早（或第一个）条目。表格中的所有其他条目均向上移动一个位置。每次执行指令时，表格中的条目数减 1。

提示：欲建立表格，首先为最大表条目数建立一个条目。如果您没有这样做，则无法在表格中建立任何条目。此外，所有的表格读取和表格写入指令必须用边缘触发器指令激活。

后进先出指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围 SM1.5 空表
	影响的特殊内存位	SM1.5 空表

指令列表： LIFO TBL, DATA

梯形图：



输入输出参数	数据类型	适用操作数
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC
DATA	整数	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *LD, *AC

功能说明：后入先出（LIFO）指令将表格中的最新（或最后）一个条目移至输出内存地址，方法是移除表格（TBL）中的最后一个条目，并将数值移至 DATA 指定的位置。每次执行指令时，表格中的条目数

减 1。

提示：欲建立表格，首先为最大表条目数建立一个条目。如果您没有这样做，则无法在表格中建立任何条目。此外，所有的表格读取和表格写入指令必须用边缘触发器指令激活。

表格查找指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围
	影响的特殊内存位	无

指令列表：

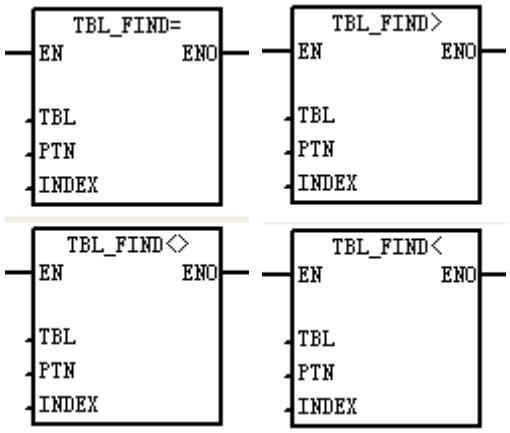
FND= TBL, PIN, INDEX

FND<> TBL, PIN, INDEX

FND< TBL, PIN, INDEX

FND> TBL, PIN, INDEX

梯形图：



输入输出参数	数据类型	适用操作数
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC
PIN	整数	VW, IW, QW, MW, SW, SMW, AIW, LW, T, C, AC, 常数, *VD, *LD, *AC
INDEX	字	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC

功能说明：表格查找（TBL）指令在表格（TBL）中搜索与某些标准相符的数据。"表格查找"指令搜索表，从 INDEX 指定的表格条目开始，寻找与 CMD 定义的搜索标准相匹配的数据数值（PTN）。命令参数（CMD）被指定一个 1 至 4 的数值，分别代表 =、<>、<, and >。如果找到匹配条目，则 INDEX 指向表格中的匹配条目。欲查找下一个匹配条目，再次激活"表格查找"指令之前必须在 INDEX 上加 1。如果未找到匹配条目，INDEX 的数值等于条目计数。一个表格最多可有 100 个条目，数据项目（搜索区域）从 0 排号至最大值 99。

<注释> 当您在用 ATT、LIFO 和 FIFO 指令生成的表格中使用"表格查找"指令时，条目计数与数据条目直接对应。"表格查找"指令并不要求 ATT、LIFO 和 FIFO 所要求的最大条目数。因此，如下所示，您应当将"查找"指令的 SRC 操作数设为一个高于对应的"增加至表格"、"后入先出"或"先入先出"指令 TBL 操作数的一个字地址（两个字节）。

提示：欲建立表格，首先为最大表条目数建立一个条目。如果您没有这样做，则无法在表格中建立任何条目。此外，所有的表格读取和表格写入指令必须用边缘触发器指令激活。

追加至表格指令	设置 ENO = 0的错误条件	0006 间接地址 0091 操作数超出范围 SM1.4 表溢出

	影响的特殊内存位	SM1.4 表溢出
指令列表： ATT DATA, TABLE		
梯形图：  		

输入输出参数	数据类型	适用操作数
DATA	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, 常数, *VD, *LD, *AC
TBL	字	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, , *LD *AC

功能说明：增加至表格（ATT）指令向表格（TBL）中加入字值（DATA）。表格中的第一个数值是表格的最大长度（TL）。第二个数值是条目计数（EC），指定表格中的条目数。新数据被增加至表格中的最后一个条目之后。每次向表格中增加新数据后，条目计数加 1。表格最多可包含 100 个条目，不包括指定最大条目数和实际条目数的参数。

提示：欲建立表格，首先为最大表条目数建立一个条目。如果您没有这样做，则无法在表格中建立任何条目。此外，所有的表格读取和表格写入指令必须用边缘触发器指令激活。

中断指令

从中断程序有条件返回指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： CRETI		
梯形图：  		

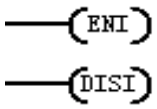
输入输出参数	数据类型	适用操作数
无	无	无

功能说明：从中断指令有条件返回（CRETI）指令可根据先前逻辑条件用于从中断返回。欲增加中断程序，在 Project Manager 主界面点菜单编辑（Edit）>插入（Insert）>Organization Block，。

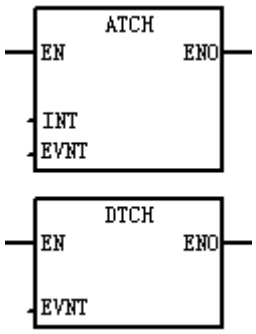
<注释> MagicWorks PLC 自动为每个中断程序增加一个无条件返回。

您应当优化中断程序，执行具体任务，然后使控制返回主例行程序。保持中断程序简明扼要，可加快执行速度，并且不会长时间延迟其他过程。否则，无法预测的条件会引起主程序控制的装置操作异常。限制在中断程序中不能使用 DISI、ENI、HDEF、LSCR、END 指令。

中断使能和禁止指令	设置 ENO = 0的错误条件	0004 尝试在中断程序中执行 ENI、DISI 或 HDEF 指令
	影响的特殊内存位	无
指令列表： ENI		

DISI		
梯形图：  		
输入输出参数	数据类型	适用操作数
无	无	无

功能说明：中断允许（ENI）指令全局性启用所有附加中断事件进程。中断禁止（DISI）指令全局性禁止所有中断事件进程。转换至 RUN（运行）模式时，中断开始时被禁止。一旦进入 RUN（运行）模式，您可以通过执行全局中断允许指令，启用所有中断进程。执行中断禁止指令会禁止处理中断；但是现用中断事件将继续入队等候。

中断连接与分离指令	设置 ENO = 0的错误条件	0002 配置对 HSC 的输入赋值
	影响的特殊内存位	无
指令列表： ATCH INT, EVENT DTCH EVENT		
梯形图：  		
输入输出参数	数据类型	适用操作数
INT	字节	OBx(x 为常数 2-127)
EVENT	字节	常数 0-33

功能说明：

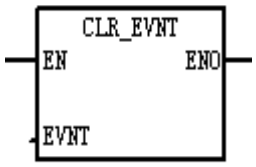
中断连接（ATCH）指令将中断事件（EVNT）与中断程序号码（INT）相联系，并启用中断事件。

中断分离（DTCH）指令取消中断事件（EVNT）与所有中断程序之间的关联，并禁用中断事件。

在激活中断程序之前，必须在中断事件和您希望在事件发生时执行的程序段之间建立关联。使用“中断连接”指令将中断事件（由中断事件号码指定）与程序段（由中断程序号码指定）联系在一起。您可以将多个中断事件附加在一个中断程序上，但一个事件不能同时附加在多个中断程序上。当您将一个中断事件附加在一个中断程序上是，会自动启用中断。如果您用全局禁用中断指令禁用所有的中断，则每次出现的中断事件均入队等候，直至使用全局启用中断指令或中断队列溢出重新启用中断。您可以使用“中断分离”指令断开中断事件与中断程序之间的关联，从而禁用单个中断事件。“中断分离”指令使中断返回至非现用或忽略状态。

清除中断事件指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： CEVNT EVENT		

梯形图：



输入输出参数	数据类型	适用操作数
EVENT	字节	常数 0-33

功能说明：清除中断事件指令会删除中断队列中所有类型为 EVNT 的中断事件。此指令用于清除不必要的中断，后者可能由假传感器输出暂态造成。

程序控制指令

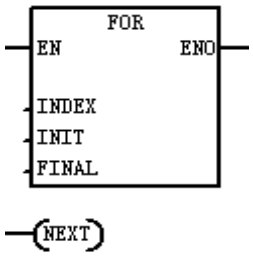
循环指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	无

指令列表：

FOR INDEX, INIT, FINAL

NEXT

梯形图：



输入输出参数	数据类型	适用操作数
INDEX	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC
INIT	整数	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, 常数, *VD, *LD, *AC
FINAL	整数	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, 常数, *VD, *LD, *AC

功能说明：FOR（FOR）指令执行 FOR 和 NEXT 之间的指令。您必须指定索引值或当前循环计数（INDEX）、起始值（INIT）和结束值（FINAL）。NEXT（NEXT）指令标记 FOR 循环结束，并将堆栈顶值设为 1。使用 FOR/NEXT 指令描述为指定计数重复的循环。每条 FOR 指令要求一个 NEXT 指令。您可以复原 FOR/NEXT 循环（在 FOR/NEXT 循环中放置一个 FOR/NEXT 循环），深度可达八。例如，假定 INIT 值等于 1，FINAL 值等于 10，FOR 与 NEXT 之间的指令被执行 10 次，INDEX 值递增：1、2、3、...10。如果起始值大于结束值，则不执行循环。每次执行 FOR 和 NEXT 之间的指令后，INDEX 值递增，并将结果与结束值比较。如果 INDEX 大于结束值，循环则终止。

<注释>

- 如果您启用 FOR/NEXT 循环，则将继续循环程序，直至结束反复操作，除非您从循环内部改变结束值。您可以在 FOR/NEXT 处于循环过程时改变数值。
- 再次启用循环时，它将初始值复制至索引值（当前循环次数）。下次被启用时，FOR/NEXT 指令复原。

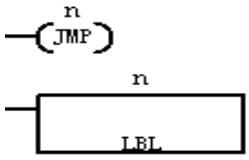
跳转指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

JMP n

LBL n

梯形图：



输入输出参数	数据类型	适用操作数
n	字节	常数　0～255

功能说明：

跳转至标签（JMP）指令对程序中的指定标签（n）执行分支操作。跳转接受时，堆栈顶值始终为逻辑 1。

标签（LBL）指令标记跳转目的地（n）的位置。

您可以在主程序、子程序或中断程序中使用"跳转"指令。"跳转"及其对应的"标签"指令必须始终位于相同的代码段中（主程序、子程序或中断程序）。您不能从主程序跳转至子程序或中断程序中的标签，与此相似，您也不能从子程序或中断程序跳转至该子程序或中断程序之外的标签。您可以在 SCR 段中使用"跳转"指令，但对应的"标签"指令必须位于相同的 SCR 段内。

顺序控制继电器指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表：

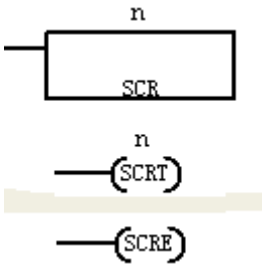
LSCR n

SCRT n

SCRE

CSCRE

梯形图：



输入输出参数	数据类型	适用操作数
n	布尔	S

功能说明：SCR 指令为您提供一种可自然纳入 LAD、FBD 或 STL 程序的简单、强有力的状态控制编程技术。每当应用程序包含一系列必须重复执行的操作时，SCR 可用于为程序安排结构，以便使之直接与应用程序相对应。因而您能够更快速、更方便地编程和调试应用程序。载入顺序控制继电器（LSCR）指令用指令（N）引用的 S 位数值载入 SCR 和逻辑堆栈。SCR 段被 SCR 堆栈的结果数值激励或取消激励。SCR 堆栈数值被复制至逻辑堆栈的顶端，以便方框或输出线圈可直接与左电源杆连接，无须插入触点。顺序控制继电器转换（SCRT）指令识别要启用的 SCR 位（下一个要设置的 n 位）。当使能位进入线圈或 FBD 方框时，打开引用 n 位，并关闭 LSCR 指令（启用该 SCR 段）的 n 位顺序控制继电器结束（SCRE）指令标记 SCR 段的结束。一旦将电源应用于输入，有条件顺序控制继电器结束（CSCRE）指令即标记

SCR 段结束。CSCRE 只有在 STL 编辑器中才能使用。

<注释> CSCRE 只有在第二代（22x）CPU（从 1.20 版开始）才能使用。

下列说明适用于顺序控制继电器指令。

- "载入 SCR"指令（LSCR）标记 SCR 段的开始，"SCR 结束"指令（SCRE）标记 SCR 段的结束。"载入 SCR"和"SCR 结束"指令之间的所有逻辑执行取决于 S 堆栈数值。"SCR 结束"和下一条"载入 SCR"指令之间的逻辑不取决于 S 堆栈数值。
- "SCR 转换"指令（SCRT）提供一种从现用 SCR 段向另一个 SCR 段转换控制的方法。当"SCR 转换"指令有使能位时，该指令会复原当前现用段的 S 位，并设置被引用段的 S 位。在"SCR 转换指令执行时，复原现用段的 S 位不会影响 S 堆栈。因此，SCR 段在退出前保持激励状态。
- “有条件 SCR 结束”指令（CSCRE）提供一种无须执行“有条件 SCR 结束”和“SCR 结束”指令之间的指令即可退出现用 SCR 段的方法。“有条件 SCR 结束”指令不会影响任何 S 位，亦不会影响 S 堆栈。

使用 SCR 的限制

- 您不能在一个以上例行程序中使用相同的 S 位。例如，如果您在主程序中使用 S0.1，则不能在子程序中再使用。
- 您不能在 SCR 段中使用 JMP 和 LBL 指令。这表示不允许跳转入或跳转出 SCR 段，亦不允许在 SCR 段内跳转。您可以使用跳转和标签指令在 SCR 段周围跳转。
- 您不能在 SCR 段中使用"结束"指令。

用 SCR 段控制程序流

CTSC-200 PLC 中的主程序包含每次 PLC 扫描即按顺序执行一次的指令。对于很多应用程序，可能需要以逻辑方式将主程序分为一系列操作步骤，以便反映控制进程中的步骤（例如，一系列机器操作）。

以逻辑方式将程序分为多个步骤的一种方法是使用 SCR 段。通过使用 SCR 段，可以将程序分为一个连续步骤流，或分为可以同时现用的多个步骤流。可以将一个步骤流有条件地分为多个步骤流，或将多个步骤流有条件地重新组合为一个步骤流。

看门狗复位指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： WDR		
梯形图： —(WDR)		
输入输出参数	数据类型	适用操作数
无	无	无

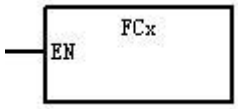
功能说明：

看门狗复原（WDR）指令重新触发 CTSC-200 CPU 的系统监视程序定时器，扩展扫描允许使用的时间，而不会出现看门狗错误。使用"看门狗复原"指令时应当小心。如果使用循环指令阻止扫描完成或严重延迟扫描完成，下列程序只有在扫描周期完成后才能执行：

- 通讯（自由端口模式除外）
- I/O 更新（立即 I/O 除外）
- 强迫更新
- SM 位更新（不更新 SM0、SM5 至 SM29）
- 运行时间诊断程序

- 10 毫秒和 100 毫秒定时器对于超过 25 秒的扫描不能正确地累计时间
- 用于中断程序时的 STOP（停止）指令
- 配备离散输出的扩充模块还包括看门狗定时器，如果模块未被 CTSC-200 写入，监视程序定时器会关闭输出。对每个配备离散输出的扩充模块使用立即写入，在扩展扫描时间期间使正确的输出保持打开。

<注释> 如果您预计扫描时间将超过 500 毫秒，或者您预计会发生大量中断活动，可能阻止返回主扫描超过 500 毫秒，您应当使用 WDR 指令，重新触发看门狗定时器。每次使用“看门狗复原”指令时，您还应当使用对每个离散扩充模块中的一个输出字节（QB）使用立即写入，复原每个扩充模块看门狗。如果您使用"看门狗复原"指令允许执行要求很长扫描时间的程序，将模式开关改变为 STOP（停止）位置会使 CTSC-200 在 1.4 秒内转换为 STOP（停止）模式。

调用	设置 ENO = 0的错误条件	0008 功能嵌套数超过最大
	影响的特殊内存位	无
指令列表： CALL Function Name, parameter 1, ... parameter n (最多允许16个参数)		
梯形图：  		
输入输出参数	数据类型	适用操作数
FCx	功能	功能(x = 0 - 127)
使能位	布尔	使能位
输入参数	布尔，字节，字，整数，双字，双整数，字符串	V, I, Q, M, SM, S, T, C, L, VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C, &SB, &AI, &AQ, &SMB, *VD, *LD, *AC, constant
输入/输出参数	布尔，字节，字，整数，双字，双整数	V, I, Q, M, SM, S, T, C, L, VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC
输出参数	布尔，字节，字，整数，双字，双整数	V, I, Q, M, SM, S, T, C, L, VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC

功能说明：调用功能（CALL）指令将控制转换给功能（FCx）。您可以使用带参数或不带参数的 "调用功能" 指令。在功能完成执行后。控制返回至 "调用功能" 之后的指令。每个功能调用的输入 / 输出参数最大限制为 16。如果您尝试下载的程序超过此一限制，会返回一则错误信息。如果您为功能指定一个符号名，例如 USR_NAME，该符号名会出现在指令树的 "功能" 文件夹中。

将参数值指定给功能中的局部内存时应遵守下列规则：

1）参数值指定给局部内存的顺序由 CALL 指定，参数从 L.0 开始。

2）一至八个连续位参数值被指定给从 Lx.0 开始持续至 Lx.7 的单字节。

3）字节、字和双字数值被指定给局部内存，位于字节边界（LBx、LWx 或 LDx）位置。在带参数的"调用功能"指令中，参数个数、类型与顺序必须与功能局部变量表中定义的变量完全匹配。参数顺序必须以

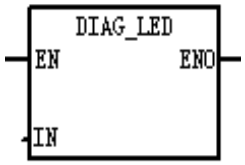
输入参数开始，其次是输入 / 输出参数，然后是输出参数。

<注释> 位于指令树中的功能名称的工具提示显示每个参数的名称。

诊断 LED 指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： DLED IN

梯形图：



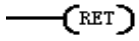
输入输出参数	数据类型	适用操作数
IN	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC

功能说明：

如果输入参数 IN 的数值为零，则诊断 LED 会被设置为不发光。如果输入参数 IN 的数值大于零，则诊断 LED 会被设置为发光 (黄色)。标记为 SF/ DIAG 的 CPU 发光二极管 (LED) 能够配置为： 当在系统块内指定的条件为真或当 DIAG_LED 指令以非零 IN 参数得到执行时发黄光。系统块 (配置 LED) 复选框选项：1) SF/ DIAG LED 会在 CPU 中的某个项被强制时发黄光 2) SF/ DIAG LED 会在某模块有 I/O 错误时发黄光取消对两个“配置 LED”选项的复选，就会让 DIAG_LED 指令独自控制 SF/ DIAG 黄光。CPU 系统故障 (SF) 用红光表示。

从子程序有条件返回指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： CRET

梯形图： 

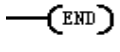
输入输出参数	数据类型	适用操作数

功能说明：

从子程序有条件返回（CRET）指令根据前一个逻辑终止子程序。从子程序有条件返回指令是供选用指令。MagicWorks PLC 会自动增加要求使用的从子程序无条件返回指令，且不在程序编辑器的“子程序 POU”标记显示的程序逻辑中显示。

有条件结束指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无

指令列表： END

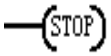
梯形图： 

输入输出参数	数据类型	适用操作数

功能说明：

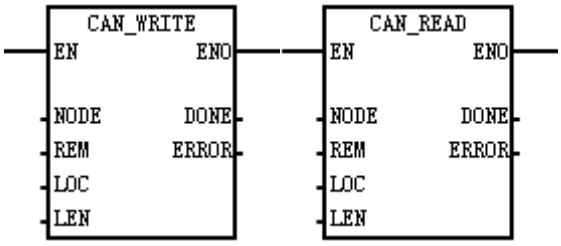
有条件结束（END）指令根据前一个逻辑条件终止主用户程序。

<注释> 您可以在主程序中使用“有条件结束”指令，但不能在子程序或中断程序中使用。MagicWorks PLC 自动在主用户程序中增加无条件结束。

有条件停机指令	设置 ENO = 0的错误条件	无
	影响的特殊内存位	无
指令列表： STOP		
梯形图： 		
输入输出参数	数据类型	适用操作数

功能说明：在条件满将 CPU 切换至 STOP 状态。

通信指令

CANopen SDO 读写指令	设置 ENO = 0的错误条件	0006 间接地址
	影响的特殊内存位	SMB500~SMB532(显示 CANopen 通信的状态信息)
指令列表： CANW NODE,REM,LOC,LEN,DONE,ERROR CANR NODE,REM,LOC,LEN,DONE,ERROR		
梯形图： 		

输入输出参数	数据类型	适用操作数
NODE	字节	常数（1～127）
REM	字节	VB, *VD, *LD, *AC
LOC	字节	VB, *VD, *LD, *AC
LEN	字节	常数（1～255）
DONE	布尔	Q, M, V
ERROR	双字	VD, QD, MD, AC, *VD, *LD, *AC

备注：CTH300-H 系列 PLC 不支持 CANopen SDO 读写指令

SDO 错误信息说明：

错误码	说明
0x00000000	指令执行无错误
0x00000001	主站不处于 Operational 状态，因此不能发送
0x00000002	目标节点故障，因此不发送
0x00000003	目标节点不存在，因此不发送
0x00000004	上一次发送的相同命令尚未得到回应
0x00000005	用户输入的指令参数有错误
0x00000006	指令缓冲区已满
0x00000007	超时没有收到回应
0x00000008	收到的回应报文的数据长度有错误
0x00000009	收到的回应报文不是期望的回应报文

功能说明：

CANopen SDO 通信指令（CANW 和 CANR）是用于实现 CANopen 通信的 SDO 传输。

CANW 将本地 LOC 开始长度为 LEN 字节的 V 内存的值写到 NODE ID 为 NODE 的远程站中，远程站中的写入地址为 REM 开始长度为 LEN 的 V 内存。

CANR 是读入 NODE ID 为 NODE 的远程站的一块 V 内存的值，读取数据的来源为 REM 开始长度为 LEN 的 V 内存，读取数据在本地的存放地址为 LOC 开始长度为 LEN 的 V 内存。

DONE 是完成标志位，读写未完成时为 0，每次读写完成后 DONE 被置 1。

ERROR 是错误标志，用于显示错误信息，ERROR=0 时表示无错误。

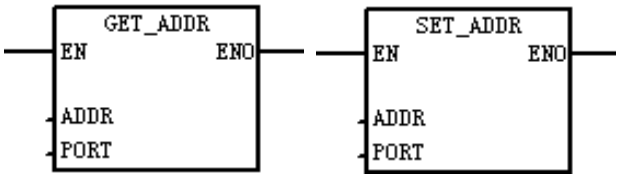
读取设置端口地址指令	设置 ENO = 0的错误条件	0006 间接地址 0004 尝试在中断程序中执行 SPA(SPA)
	影响的特殊内存位	无

指令列表：

GPA ADDR, PORT

SPA ADDR, PORT

梯形图：



输入输出参数	数据类型	适用操作数
ADDR(GPA)	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC
ADDR(SPA)	字节	VB, IB, QB, MB, SB, SMB, LB, AC, 常数, *VD, *LD, *AC
PORT	字节	常数（0用于 CPU224；0或1用于 CPU 226/226M/226L）

功能说明：

获得端口地址指令（GPA）读取端口号为 PORT 的通信端口站地址，并将数值置于 ADDR 中指定的地址内。

设置端口地址指令（SPA）将端口号为 PORT 的通信端口的站地址设为 ADDR 中指定的数值，设定的新地址不会被永久性保存。电源重新上电后，设置的端口会返回至最后的地址（系统块里设置并下载的地址）。

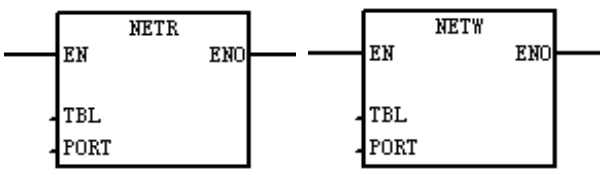
网络读写指令	设置 ENO = 0的错误条件	0006 间接地址 表状态字节 E 位为1
	影响的特殊内存位	无

指令列表：

NETR TABLE, PORT

NETW TABLE, PORT

梯形图：



输入输出参数	数据类型	适用操作数
TBL	字节	VB, MB, *VD, *LD, *AC

PORT	字节	常数（0用于 CPU224；0或1用于 CPU 226/226M/226L）

功能说明：

网络读取指令（NETR）开始一项通讯操作，通过指定的端口 PORT 根据表格 TBL 定义从远程设备收集数据。

网络写入指令（NETW）开始一项通讯操作，通过指定的端口 PORT 根据表格 TBL 定义向远程设备写入数据。

网络读写指令最多可从远程站读取或往远程站写入 16 字节信息，在程序中允许有任意数目的 NETR/NETW 指令，但在任何时间最多只能有 8 条被激活，即激活的 NETR 条数加 NETW 条数之和最多为 8 条。

用户还可以使用“网络读取 / 网络写入指令向导”配置通信程序。选择菜单命令“工具”-> “NETR/NETW 指令向导”即可启动此向导。

TBL 参数定义

n7	n6	n5	n4	n3	n2	n1	n0
远程站地址							
指针指向							
数据区域							
远程站							
(I/Q/M/V)							
数据长度							
数据字节0							
数据字节1							
...							
数据字节15							

n7：完成（功能完成）0 = 未完成 1 = 完成

n6：现用（功能入队）0 = 非现用 1 = 现用

n5：错误 0 = 无错 1 = 错误

n4：保留，始终为 0

远程站址：存取数据的 PLC 地址

数据指针：指向 PLC 中数据的间接指针

数据长度：存取的数据字节数目（1-16）

接收或传输数据区域：如下所示，为数据保留的 1-16 个字节（由数据长度指定）。

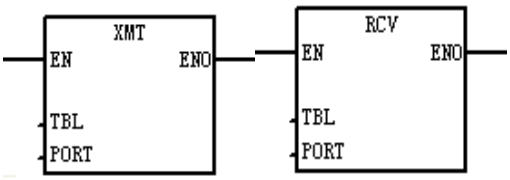
对于 NETR，该数据区域指执行 NETR 之后从远程站读取的数值存储的区域。

对于 NETW，该数据区域指执行 NETW 之前发送至远程站的数值存储的区域。

n3~n0：错误代码，含义如下：

错误代码	说明
0	无错
1	超时错误；远程站不应答
2	接收错误；应答中存在校验、帧或校验和错误
3	脱机错误；重复站址或故障硬件引起的冲突
4	队列溢出错误；8 个以上 NETR/NETW 方框被激活
5	违反协议；未启用 SMB30 中的 PPI+即尝试执行 NETR/NETW
6	非法参数；NETR/NETW 表格包含一个非法或无效数值
7	无资源；远程站繁忙（正在上载或下载序列）
8	第 7 层错误；违反应用程序协议

9	信息错误：数据地址错误或数据长度不正确

传送接收指令	设置 ENO = 0的错误条件	0006 间接地址 0009 端口0中同时执行 XMT/RCV 000B 端口1中同时执行 XMT/RCV 接收状态字节显示接收有错误
	影响的特殊内存位	SMB86和 SMB186（接收用户的禁止命令终止接收信息、输入参数错误或无起始或结束条件、收到结束字符、超时、超出最大字符数或奇偶校验错误） SMB3（自由口接收优先级错误） SMB4（中断队列溢出）
指令列表： XMT TABLE, PORT RCV TABLE, PORT		
梯形图：  		

输入输出参数	数据类型	适用操作数
TBL	字节	VB, IB, QB, MB, SB, SMB, *VD, *LD, *AC
PORT	字节	常数（0用于 CPU224；0或1用于 CPU 226/226M/226L）

功能说明：

传送指令（XMT）在自由端口模式中使用，通过通讯端口传送数据。

接收指令（RCV）开始或终止“接收信息”服务。您必须指定一个开始条件和一个结束条件，“接收”方框才能操作。通过指定端口（PORT）接收的信息存储在数据缓冲区（TBL）中。数据缓冲区中的第一个条目指定接收的字节数目。

示例程序：

STL：

```
ORGANIZATION_BLOCK 主程序：OB1
// 用串口工具写入以 S 开头 E 结束的字符串正确返回，Q0.0/Q1.0 亮
Network 1
LD          SM0.1
MOVB        0, VB500
MOVB        0, VB1000
MOVB        16#09, SMB30     //初始化 PORT0 为自由口，波特率 9.6k，8 位数据位，无校验
MOVB        16#E0, SMB87    //允许接收，起始结束检测
MOVB        'S', SMB88       //起始字符‘S’
MOVB        'E', SMB89       //结束字符‘E’
MOVB        200, SMB94       //接收缓冲区为 200 个字节
ATCH        INT0, 23         //启用中断，接收完成
ATCH        INT1, 9          //发送完成
ENI
RCV         VB100, 0
INTERRUPT_BLOCK INT_0：INT0

Network 1
LD          SM0.0
XMT         VB100, 0         //将收到的字符发送给 Port 0
```

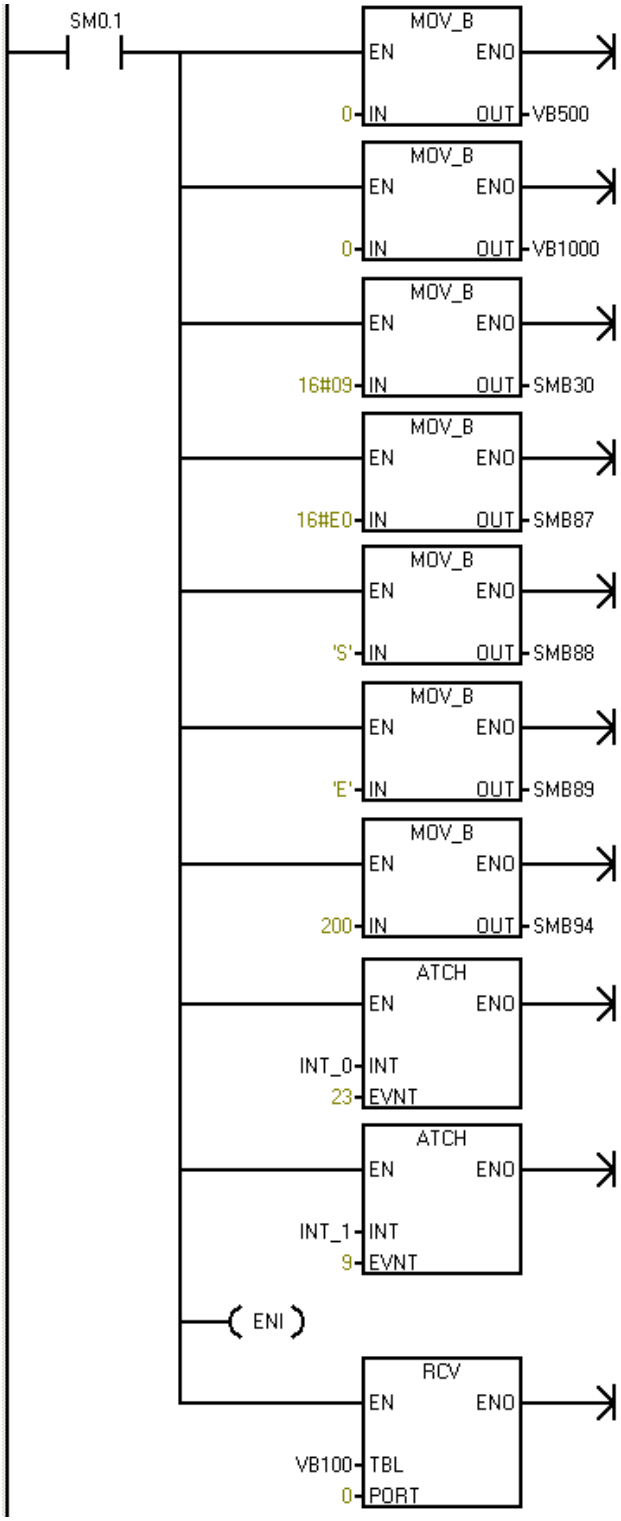
```
AENO
=           Q0.0
INCB        VB1000
Network 2
LD          SM0.0
CRETI
INTERRUPT_BLOCK INT_1：INT1

Network 1
LD          SM0.0
RCV         VB100, 0        //发送完成后接收新的字符
AENO
=           Q0.1
INCB        VB500

Network 2
LD          SM0.0
CRETI
```

梯形图：

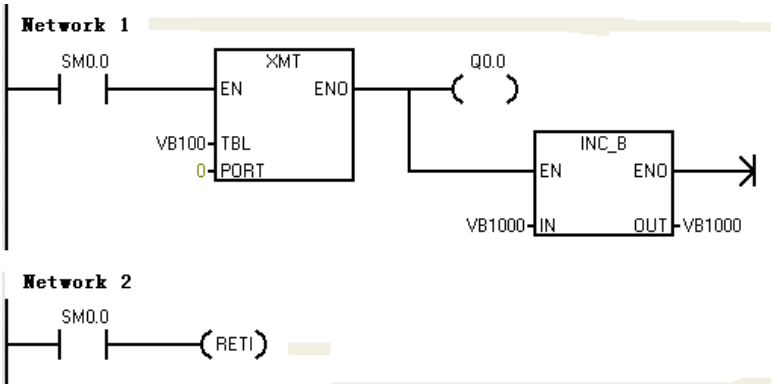
ORGANIZATION_BLOCK 主程序：OB1

// 用串口工具写入以 S 开头 E 结束的字符串正确返回，Q0.0/Q1.0 亮

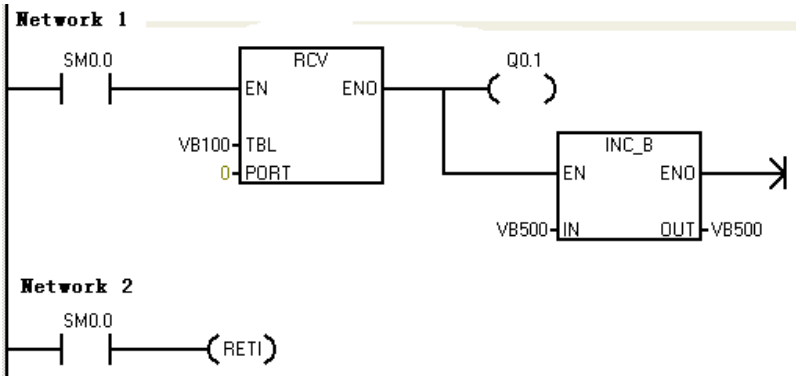
Network 1

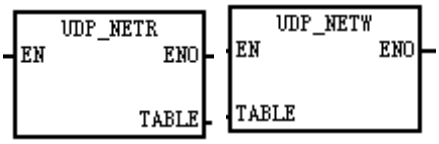


INTERRUPT_BLOCK INT_0：INT0



INTERRUPT_BLOCK INT_1：INT1



以太网 PPI 网络读写指令 UDP_NETR/UDP_NETW		
指令列表：UDP_NETR/UDP_NETW EN, TABLE, ENO		
梯形图： 		
参数	**类型**	**意义**
EN	BOOL	命令使能位，上升沿触发
TABLE	BYTE	包含状态位、IP 地址、指针、数据长度、数据字节，具体描述参考表“UDP_NETR/UDP_NETW 中参数 TABLE 的描述”。

功能说明：以太网口网络读取指令（UDPNETR）开始一项通讯操作，通过 CPU 上的 Ethernet 通信口，根据表格 TBL 定义从远程设备收集数据。

以太网口网络写入指令（UDPNETW）开始一项通讯操作，通过 CPU 上的 Ethernet 通信口，根据表格 TBL 定义远程设备写入数据。

网络读写指令最多可从远程站读取或从远程站写入 200 字节信息，在程序中允许有任意数目的 NETR/NETW 指令，但在任何时间最多只能有 8 条指令被激活。

TABLE 参数使用说明

字节偏移量　　7　　0

D	A	E	n4	错误代码
远程站 IP 地址第 1 个字节				
远程站 IP 地址第 2 个字节				
远程站 IP 地址第 3 个字节				

远程站 IP 地址第 4 个字节
端口号高字节
端口号低字节
指向远程站目标区域指针<I，Q，M，V，DB>（共 4 个字节）第 1 个字节
指向远程站目标区域指针第 2 个字节
指向远程站目标区域指针第 3 个字节
指向远程站目标区域指针第 4 个字节
数据长度
数据字节 0
数据字节 1
...
数据字节 199

D：命令完成位，0=未完成，1=完成

A：激活位，0=指令不在执行中，1=指令正在执行中

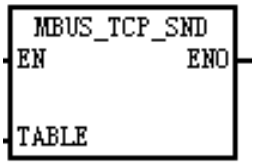
E：错误，0=无错，1=错误

n4：保留，始终为 0

TABLE 参数中错误代码的描述

错误代码	说明
0	无错误
1	超时错误，远程站不应答
2	接收错误，应答中存在校验、帧或校验和错误
3	脱机错误，重复站地址或故障硬件引起的冲突
4	队列溢出错误，8 个以上 NETR/NETW 指令被激活
6	非法参数，NETR/NETW 表格包含一个非法或无效数值
7	无资源，远程站繁忙（正在上载或下载序列）
8	层 7 错误，违反应用程序协议
9	信息错误，数据地址或数据长度不正确
10	已达到最大连接数
129	内部错误，表指针为空

注释：支持同时 8 个不同 IP 地址（即 8 个连接）的读写，每个 IP 地址可有 32 条读写数据信息，每个信息读写的数据长度最大为 200 字节。

以太网 MODBUS 网络读写指令 MBUS_TCP_SND		
指令列表：MBUS_TCP_SND EN, TABLE, ENO		
梯形图：  		
参数	**类型**	**意义**
EN		MBUS_TCP_SND 使能位
TABLE	BYTE	包含状态位、IP 地址、端口号、读写类型、指针、数据长度、数据字节，具体描述参考下表 TABLE 参数说明。

功能说明：MODBUS_TCP 网络读写指令开始一项通讯操作，通过 MODBUS_TCP 协议与指定的远程设备交换数据。用户还可以使用“MODBUS_TCP 指令向导”配置通信程序。

TABLE 参数说明

字节偏移量 7 0

D	A	E	n4	错误代码
远程站 IP 地址第 1 个字节				
远程站 IP 地址第 2 个字节				
远程站 IP 地址第 3 个字节				
远程站 IP 地址第 4 个字节				
端口号高字节				
端口号低字节				
消息读写类型 0：读 1：写				
读写地址第 1 个字节（高字节）				
读写地址第 2 个字节				
读写地址第 3 个字节				
读写地址第 4 个字节（低字节）				
保留				
单元 ID				
数据长度高字节				
数据长度低字节				
数据字节 0				
...				
数据字节 239				

D：读写功能完成位，0=未完成，1=完成

A：激活位，1=指令正在执行中，0=指令不在执行中

E：错误状态位，0=无错；1=有错误

n4：保留，始终为 0

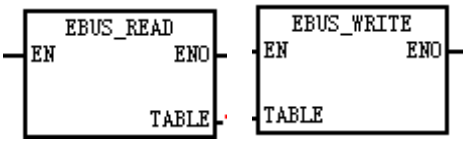
TABLE 参数中错误代码的描述（只有在 Done 位为 1 时，错误代码才有效）

错误代码	说明
0	无错误
1	响应校验错误
2	未用
3	接收超时（从站无响应）
4	请求参数错误（slave address，Modbus address，count， RW）
5	Modbus/自由口未使能
6	Modbus 正在忙于其它请求
7	响应错误（响应不是请求的操作）
8	响应 CRC 校验和错误
101	从站不支持请求的功能
102	从站不支持数据地址
103	从站不支持此种数据类型
104	从站设备故障
105	从站接受了信息，但是响应被延迟
106	从站忙，拒绝了该信息

107	从站拒绝了信息
108	从站存储器奇偶错误

注释：

- 同时可有多个从站消息读写，但是每个从站只能同时有一条消息读写。
- TCP_MBUS_MSG 收到应答之后或者出错的时候 Done 才打开。
- TCP MODBUS 最大支持 32 个连接，每条消息一次最大可读写 240 字节。

外部总线读/写指令		
EBUSR TBL EBUSW TBL		
梯形图 		
输入输出参数	数据类型	适用操作数
TBL	字节	VB, MB, *VD, *LD, *AC

功能说明：从模块读取 nBytes 到 TBL 中指向的地址/将 TBL 中指定的 nBytes 写入到模块。

TABLE 参数表定义：

错误码 ErrCode							
R3	R2	R1	R0	S3	S2	S1	S0
偏移量 Offset0							
偏移量 Offset1							
数据长度 Len							
数据字节 Byte[0]							
数据字节 Byte[1]							
...							
数据字节 Byte[Len-1]							

ErrCode：0－成功、其它－失败；

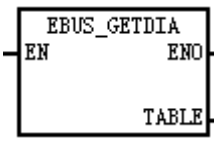
R3 R2 R1 R0：机架号 RackID

S3 S2 S1 S0：槽号 SlotID

Offset0 Offset1：模块参数的偏移量，与总线通信应用层协议表中的偏移量对应

Len：要读取或写入的数据字节数

Byte：读取或写入的数据

获取外部总线的诊断信息
EBUSGETDIA TBL
梯形图 

功能说明：获取模块的诊断信息

TABLE 参数表定义：

错误码 ErrCode							
R3	R2	R1	R0	S3	S2	S1	S0
数据长度 Len							
数据字节 Byte[0]							
数据字节 Byte[1]							
...							
数据字节 Byte[Len-1]							

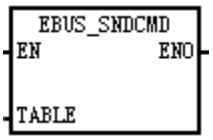
ErrCode：0－成功、其它－失败

R3 R2 R1 R0：机架号 RackID

S3 S2 S1 S0：槽号 SlotID

Len：读取的诊断信息的字节数

Byte：读取的诊断信息内容

外部总线发送命令指令		
EBUSSNDCMD　TBL		
梯形图  		
输入输出参数	数据类型	适用操作数
TABLE	字节	VB, MB, *VD, *LD, *AC

功能说明：向指定模块发送一条命令。

TABLE 参数表定义：

D	E	ErrCode5	ErrCode4	ErrCode3	ErrCode2	ErrCode1	ErrCode0
R3	R2	R1	R0	S3	S2	S1	S0
超时时间（ms）Timeout0							
超时时间（ms）Timeout1							
输入数据长度 LenIn[0]							
输入数据长度 LenIn[1]							
输入数据字节 ByteIn[0]							
输入数据字节 ByteIn[1]							
...							
输入数据字节 ByteIn[LenIn-1]							
输出数据长度 LenOut[0]							
输出数据长度 LenOut[1]							
输出数据字节 ByteOut[0]							
输出数据字节 ByteOut[1]							
...							
输出数据字节 ByteOut[LenOut-1]							

D：完成位（功能完成）　　0 = 未完成，1 = 完成

E：错误位　　0 = 无错，1 = 错误

错误代码	说明
0x00	没有错误
0x01	发送超时
0x02	模块类型不对
0x03	没有多余的发送控制块
0x04	模块忙
0x05	等待标志超时
0x06	总线标志错误
0x07	软件错误

R3 R2 R1 R0：机架号 RackID

S3 S2 S1 S0：槽号 SlotID

Timeout0 Timeout1：指令执行的超时时间

LenIn：输入数据的字节数

ByteIn：输入数据的内容，格式根据具体模块来定

LenOut：输出数据的字节数

ByteOut：输出数据的内容，格式根据具体模块来定

通信指令库

3

本章主要介绍通信指令库

3.1 CAN 通信指令库 canfree_lib（v1.3）

3.1.1 库支持的 PLC 型号

CTH200 高性能系列 CPU、充电桩专用控制器可通过调用 CAN 通信指令库来进行 CAN 通信，完成 CAN 数据收发和滤波，本节主要介绍该库的指令应用。

CTH200 系列	H224X（V5 高性能型升级版） H226XL（V5 高性能型升级版） H226XM（V5 高性能型升级版） H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版）
CTMC 系列	M228IL M228ML M228SL
CTSC 系列	224+（V5 平台） 224I（V5 平台） 224XP（V5 平台） 226I V5（V5 平台） 226M（V5 平台） 226L（V5 平台） 226H（V5 平台）
CTH200 系列充电桩专用控制器	H226XC

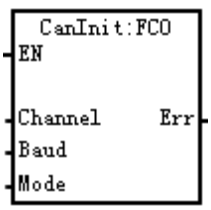
3.1.2 指令详解

CANInt（CAN 初始化）

① 指令名称：CANInt

② 功能：初始化 CAN 自由口；

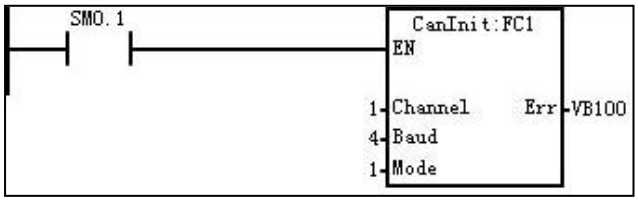
③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-1	CAN 通道选择 0：通道 0 1：通道 1 扩展板均为通道 1
	Baud	IN	BYTE	0-7	波特率 0(1MHz)；1（800kHz）；2（500kHz）；3（250kHz）；4（125kHz）；5（50kHz）；6（20kHz）；7（10kHz）
	Mode	IN	BYTE	0-1	帧格式 0：CAN_A 标准帧 1：CAN_B 扩展帧

	Err	OUT	BYTE	0-4	错误码，参见 Err 错误码定义。 0：无错 4：参数错误

④示例

初始化 CAN 口参数为：CAN 小板通道，125kHz 波特率，帧格式为扩展帧格式

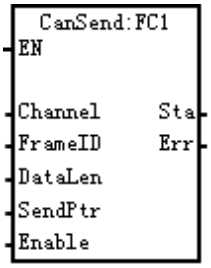


CANSend（CAN 发送数据）

① 指令名称：CANSend

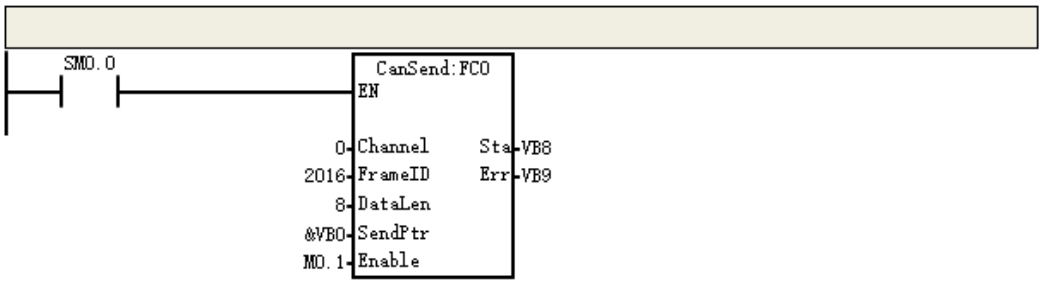
② 功能：CAN 发送数据；启用该指令，Enable 置 1 则发送一次

③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-1	CAN 通道选择 0：通道 0 1：通道 1 扩展板均为通道 1
	FrameID	IN	DWORD	--	发送数据的帧 ID
	DataLen	IN	BYTE	--	发送数据的帧长度
	SendPtr	IN	DWORD	--	发送数据缓冲区指针
	Enable	IN	BOOL	--	使能位
	Sta	OUT	BYTE	0-1	发送状态（0-未完成，1-完成） 发送完成置 1，自动清 0
	Err	OUT	BYTE	0-4	错误码（参见 Err 错误码定义） 本地 CAN0/CAN1 通道：0-无错；1-发送超时；4-参数错误 CAN 扩展板通道：0-无错，2-发送缓冲区已满

④示例

通过 CAN 扩展板通道发送 8 字节数据，帧 ID 为 2016（通过 SM0.0 调用，通过使能位 Enable 触发）

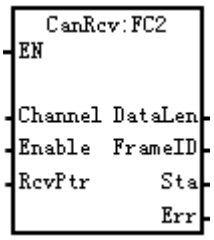


CANRcv（CAN 接收数据）

① 指令名称：CANRcv；

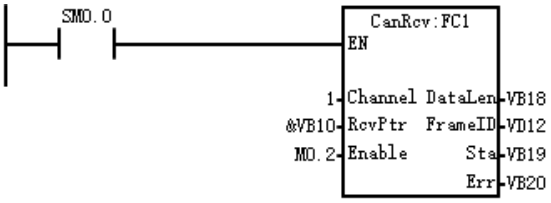
② 功能：CAN 接收数据；收到数据后会把数据填到接收缓冲区，接收指令用于查询接收缓冲区是否有数据。Enable 置 1 一次，就查询一下接收缓冲区。

③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-1	CAN 通道选择 0：通道 0 1：通道 1 扩展板均为通道 1
	Enable	IN	BOOL		使能位
	RcvPtr	IN	DWORD	--	接收数据缓冲区指针
	FrameID	OUT	DWORD	--	接收到数据的帧 ID
	DataLen	OUT	BYTE	--	接收到数据的帧长度
	St	OUT	BYTE	0-1	接收状态：0-接收未完成 1-接收完成
	Err	OUT	BYTE	0-4	错误码（参见 Err 错误码定义） 0：无错； 3：接收缓冲区已空

④示例

通过 CAN 扩展板通道接收指针所指存储器中的数据，存放帧 ID 为 2016（通过 SM0.0 调用，通过使能位 Enable 触发）

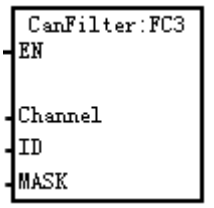


CANFilter（CAN 接收数据报文过滤）

① 指令名称：CANFilter；

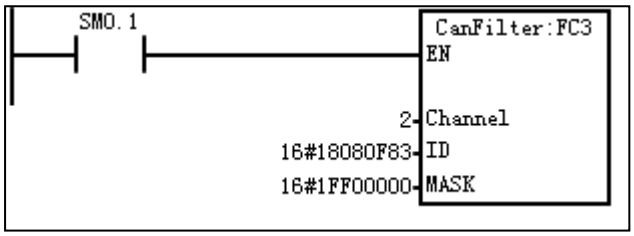
② 功能：CAN 接收数据报文过滤；如果只接收特定帧 ID 的 CAN 报文，可以调用过滤指令。如果程序不调用过滤指令，默认过滤 ID 和掩码都是 0，即接收任何 CAN 报文。

③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-1	CAN 通道选择 0：通道 0 1：通道 1 扩展板均为通道 1
	ID	IN	DWORD		要接收的数据报文 ID
	MASK	IN	DWORD	--	对 ID 进行掩码控制

④示例

通过 CAN 扩展板接收帧 ID 为 0x180xxxxx 的报文

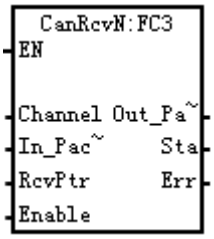


CanRcvN（接收多帧数据）

① 指令名称：CanRcvN；

② 功能：读取若干帧 CAN 数据，输出实际数量帧 CAN 数据，顺序存放到缓冲区内存，13 字节偏移一帧

③参数说明

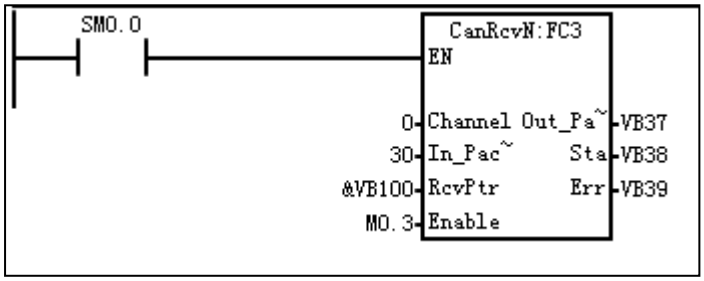
库函数	参数名称	输入输出属性	类型	数值范围	说明
	EN	IN	BOOL	--	--
	Channel	IN	BYTE	0-1	CAN 通道选择 0：通道 0 1：通道 1 扩展板均为通道 1
	In_packets	IN	BYTE		设定一次接收的帧数
	RcvPtr	IN	DWORD		接收数据内存地址
	Enable	IN_OUT	BOOL	--	使能位
	Out_packets	OUT	BYTE		实际接收到的帧数
	Sta	OUT	BYTE	0-1	接收状态：0-未完成；1-完成
	Err	OUT	BYTE	0-4	错误码(0 无错误，3 缓冲区已空，4 参数错误)

CAN 数据结构（共 13 字节）

帧 ID	帧长 len<=8	帧数据 Data
4bytes	1byte	8bytes

④ 示例

通过 CAN0 口一次接收 30 帧数据存储在 VB100 中，将实际的输出帧数存放到缓冲区地址 VB37 中。



Err 错误码定义

错误编码	定义
0	无错误

1	发送超时
2	缓冲区已满（发送）
3	缓冲区已空（接收）
4	参数错误
5	Canopen 协议复用冲突

3.2 MODBUS 主站和从站库

一共有 4 个库，分别是 PORT0 口的主从站库，PORT1 口的主从站库。

MODBUS 功能块主要针对 MODBUS 功能块占用 CPU 大量程序空间和数据空间而提供的一组内嵌的简单易用的 MODBUS 协议库。MODBUS 功能块是集成在 CPU 内部，不占用用户数据空间，作为一组库函数提供给用户使用。

提示
如需该库请前往合信网站免费下载，网址：http://www.co-trust.com

3.2.1 库支持的 PLC 型号

CTH200 系列	H224（基本型、标准型） H226L（基本型、标准型） H226M（标准型） H224X（老平台高性能型，V5 高性能型升级版） H226XM（老平台高性能型，V5 高性能型升级版） H226XL（老平台高性能型，V5 高性能型升级版） H228XL（老平台高性能型） H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版）
CTMC 系列	M228IL M228ML M228SL
CTSC 系列	224+（老平台） 224+（V5 升级版） 224E（老平台） 224I（V5 升级版） 224XP（老平台） 224XP（V5 升级版） 226I（老平台） 226I（V5 升级版） 226M（老平台） 226M（V5 升级版） 226L（老平台） 226L（V5 升级版） 226H（老平台） 226M-CAN（老平台） 226H（V5 升级版）
CTH300-H 系列	H36-01 H32-01 H35-00 H31-00 H56-10 H52-10

3.2.2 库功能说明及指令详解

【Modbus 地址】

Modbus 地址通常是包含数据类型和偏移量的 5 个或 6 个字符值。第一个或前两个字符决定数据类型，最后的四个字符是符合数据类型的一个适当的值。Modbus 主站则将这个地址对应到正确的功能上。

对于 CTH200 系列和 CTSC200 系列的 CPU 而言，Modbus 从站指令支持以下地址：00001 至 00128 是实际输出，对应 Q0.0--Q15.7；10001 至 10128 是实际输入，对应 I0.0—I15.7；30001 至 30032 是模拟输入寄存器，对应 AIW0 至 AIW62；40001 至 4XXXX 是保持寄存器，对应 V 内存区；

Modbus 地址与 CTH200 系列 CPU 地址对应关系

Modbus地址	CTH200系列CPU
000001	Q0.0
000002	Q0.1
000003	Q0.2
……	……
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
……	……
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
……	……
030032	AIW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
……	……
04xxxx	HoldStart+2 x (xxxx-1)

对于 CTH300-H 系列的 CPU 而言，Modbus 从站指令支持以下地址：00001 至 02048 是实际输出，对应 Q0.0~Q255.7；10001 至 12048 是实际输入，对应 I0.0~I255.7；30001 至 30512 是模拟输入寄存器，对应 AIW0 至 AIW1022；40001 至 4XXXX 是保持寄存器，对应 V 内存区。

所有 Modbus 地址都是从 1 开始编号的。下表所示为 Modbus 地址与 CPU 地址的对应关系。Modbus 从站协议允许您对 Modbus 主站可访问的输入、输出、模拟输入和保持寄存器（V 区）的数量进行限定。

Modbus 地址与 CTH300-H 系列 CPU 地址对应关系

Modbus地址	CPU地址
000001	Q0.0
000002	Q0.1
000003	Q0.2
……	……

002047	Q255.6
002048	Q255.7
010001	I0.0
010002	I0.1
010003	I0.2
……	……
012047	I255.6
012048	I255.7
030001	AIW0
030002	AIW2
030003	AIW4
……	……
030512	AIW1022
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
……	……
04xxxx	HoldStart+2 x (xxxx-1)

【使用 Modbus 从站协议指令】

※ CT_MODBUS 从站协议指令占用 CPU 的资源

1）根据使用不同的 Modbus 协议库占用自由口 PORT0 或者 PORT1 作为 Modbus 从站协议通讯。当 PORT0 或者 PORT1 作为 Modbus 协议通讯时，它不能再作为其它任何目的使用，包括与 MagicWorks PLC、SETP7-Micro/WIN 通讯、PLC 通讯、自由口通讯。MBUS_INIT 指令控制 Port 的设定是 Modbus 协议还是 PPI。

2）与选用 Port 自由口通讯相关的所有的 SM。

3）占用 92 个字节程序空间。

※ 在程序中使用 Modbus 从站协议指令遵循的步骤

1）在您的程序中插入 MBUS_INIT 指令并且只在一个循环周期中执行该指令，MBUS_INIT 指令可用于对 Modbus 通讯参数的初始化或修改。当您插入 MBUS_INIT 指令时，几个隐藏的子程序和中断服务程序会自动地添加到您的程序中。

2）在您的程序中只使用一个 MBUS_SLAVE 指令。该指令在每个循环周期中执行，为接收到的所有请求提供服务。

3）用通讯电缆将 CTH200 CPU 通讯口与 Modbus 主站连接起来。

※ Modbus 从站协议指令所支持的功能

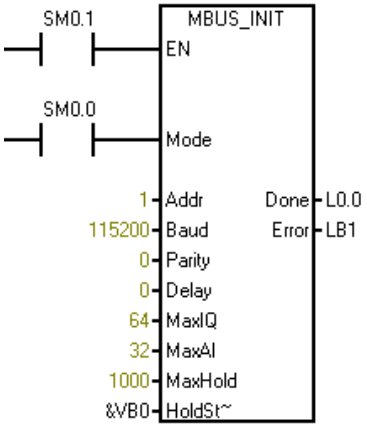
Modbus 从站协议指令支持 Modbus RTU 协议。这些指令使用 CPU 的自由口功能，支持大部分常用 Modbus 功能。以下是所支持的 Modbus 功能：

功能	描述
1	读单个/多个线圈（实际输出）状态。功能1返回任意数量输出点的接通/断开状态（Q）
2	读单个/多个接触器（实际输入）状态。功能2返回任意数量的输入点的接通/断开状态（I）

3	读单个/多个保持寄存器。功能3返回V存储器的内容，保持寄存器在Modbus下是字类型，在一个请求中最多可读120个字
4	读单个/多个输入寄存器。功能4返回模拟输入值
5	写单个线圈（实际输出）。功能5将实际输出点设置为指定值。该输出点不是被强制，用户程序可以重写由Modbus的请求而写入的值
6	写单个保持寄存器。功能6写一个单个保持寄存器的值到CPU的V存储区
15	写多个线圈（实际输出）。功能15写多个实际输出值到CPU的Q映像区。起始输出点必须是一个字节的开始（如，Q0.0或Q2.0），并且要写的输出的数量是8的倍数。这是Modbus从站协议指令的限定。这些点不是被强制，用户程序可以重写由Modbus的请求而写入的值
16	写多个保持寄存器。功能16写多个保存寄存器到CPU的V区。在一个请求中最多可写120字

MBUS_INIT

①指令名称：MBUS_INIT；



②功能：MBUS_INIT 指令用于使能和初始化或禁止 Modbus 通讯。MBUS_INIT 指令必须无错误的执行，然后才能够使用 MBUS_SLAVE 指令。在继续执行下一条指令前，MBUS_INIT 指令必须执行完并且 Done 位被立即置位。MBUS_INIT 指令应该在每次通讯状态改变时只执行一次。因此，EN 输入端应使用边沿检测元素以脉冲触发，或只在第一个循环周期内执行一次。

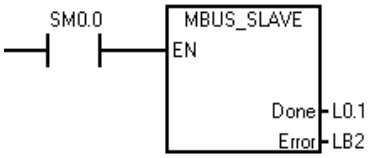
③参数说明

参数地址	说明	类型	数值范围	备注
Mode	选择通讯协议：输入1值将Port定义为Modbus协议并使能该协议，输入值为0将Port定义为PPI并禁止Modbus协议	位		
Addr	设置本站地址	字节	1~247	
Baud	设置波特率(bps)	双字	1200、2400、4800、9600、19200、38400、57600、115200	
Parity	设置校验	字节	0--无校验 1--奇校验 2--偶校验	所有设置使用一个停止位。
Delay	通过为标准Modbus信息超时增加指定数量的毫秒，扩展标准Modbus信息结束超时条件	整型	0~32767	单位：毫秒
MaxIQ	设置可使用的I点和Q点数	整型	0~128。	建议MaxIQ取值128，即

			数值为0则禁止对输入和输出的读写。	允许访问所有I/Q点。
MaxAI	设置可使用的字输入寄存器（AI）的个数	整型	0~32。 为0时则禁止读模拟输入。	MaxAI的建议值： H224/H224X/H226L/H226M/H226XL为32 H226XM/H228XL为194
MaxHold	设置可以使用的V存储区字保持寄存器的个数	整型	0~32767	单位：字
HoldStart	设置可以使用的V存储区的保持寄存器的起始地址	双字	指向V存储区的指针。	
Done	当MBUS_INIT指令完成时，Done输出接通	位		
Error	Error输出字节包含该指令的执行结果	字节		

MBUS_SLAVE

①指令名称：MBUS_SLAVE；



②功能：MBUS_SLAVE 指令用于服务来自 Modbus 主站的请求，必须在每个循环周期都执行，以便检查和响应 Modbus 请求。当 EN 输入接通时，该指令在每一循环周期内执行。MBUS_SLAVE 指令无输入参数。

③参数说明

参数地址	说明	类型	数值范围	备注
Done	当MBUS_SLAVE指令响应Modbus请求时Done输出接通。如果没有服务的请求，Done输出会断开	位		
Error	输出包含该指令的执行结果	字节	参见错误代码表	该输出只有Done接通时才有效。如果Done断开，错误代码不会改变。

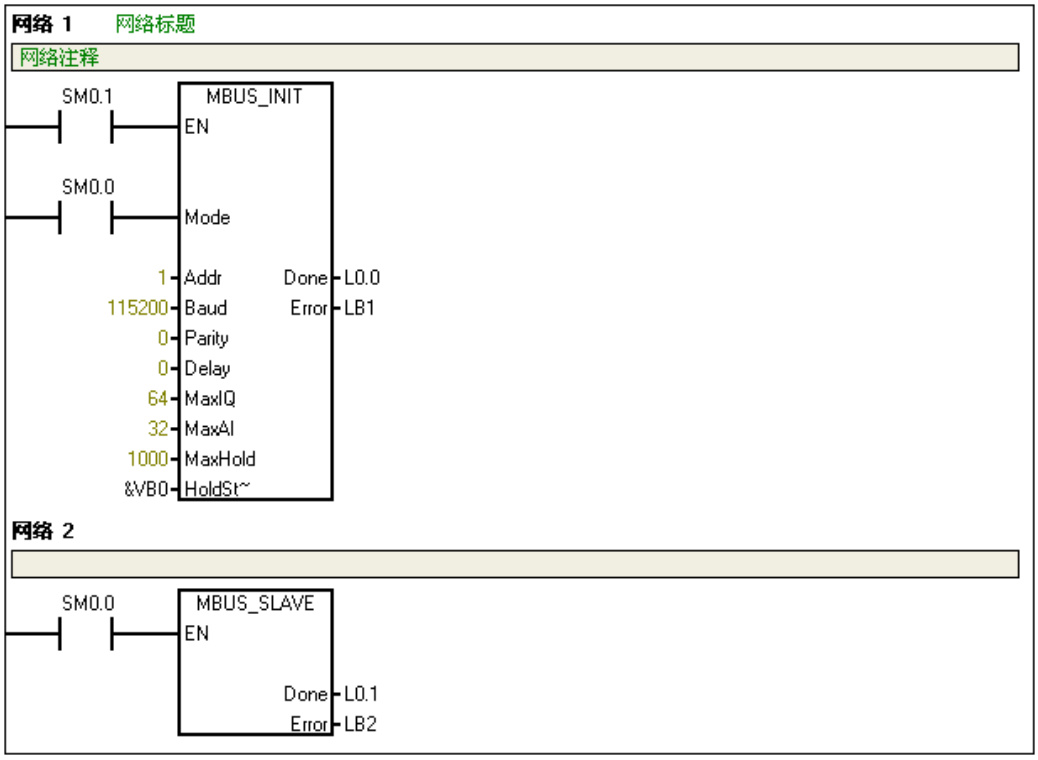
错误代码表：

0	无错误
1	存储区范围错误
2	非法波特率或校验
3	非法从站地址
4	Modbus参数的非法值
5	保持寄存器与Modbus从站符号地址重叠
6	接收校验错误
7	接收CRC错误
8	非法功能请求/不支持的功能
9	请求中有非法存储区地址

10	从站功能未使能

④Modbus 从站协议指令使用实例

下面的梯形图程序就是建立了一个从站地址为 1，波特率为 115200，无校验的 MODBUS 从站：



MBUS_INIT 参数配置说明

Addr	设置从站地址为 1
Boud	设置通信波特率为115200
Party	设置奇偶校验为无校验
Delay	设置延迟时间为0毫秒
MaxIQ	设置最大可使用64个I点和64个Q点（即000001-0000064和010001-010064）
MaxAI	设置最大可读32路模拟量输入（即030001-030032）
MaxHold	设置最大可使用的V区字保存寄存器的个数（以字为单位）
StartHold	Modbus主站可以访问S7-200 V区保存寄存器的起始地址（如&VB0）

【使用 Modbus 主站协议指令】

※ Modbus 主站协议指令占用 CTH200 的 CPU 资源

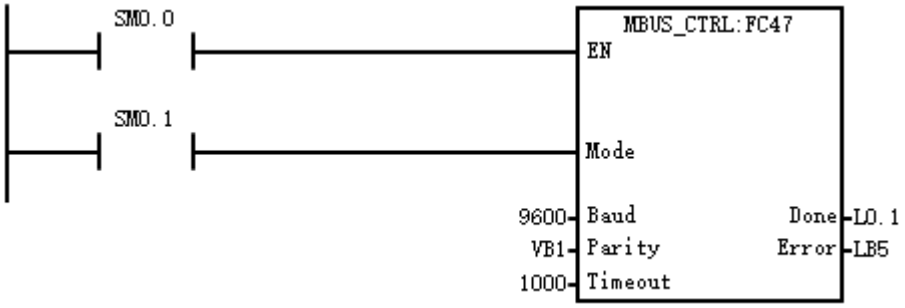
1)根据使用不同的 Modbus 协议库占用自由口 Port 0 或者 Port 1 作为 Modbus 从站协议通讯。当 Port 0 或者 Port 1 作为 Modbus 协议通讯时，它不能再作为其它任何目 的使用，包括与 MagicWorks PLC 或 SETP7-Micro/WIN 通讯，自由口通讯。MBUS_INIT 指令控制 Port 的设定是 Modbus 协议还是 PPI。

2）与选用 Port 自由口通讯相关的所有的 SM。

3）占用 119 个字节程序空间。

MBUS_CTRL

①指令名称：MBUS_CTRL；



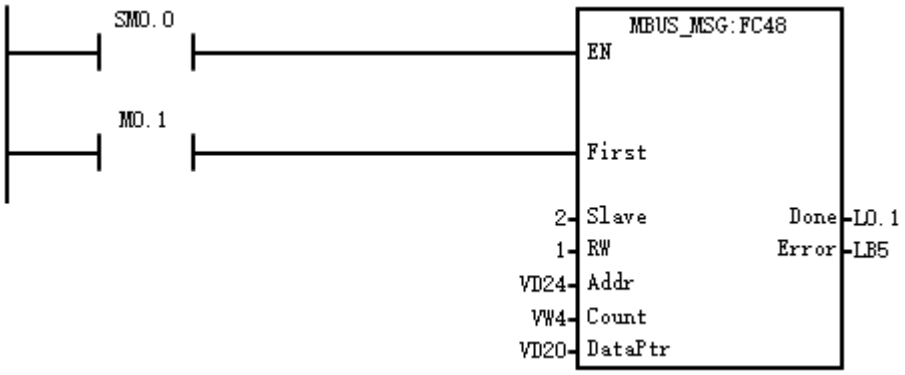
②功能：使用 SM0.0 调用 MBUS_CTRL 指令完成主站的初始化，并启动其功能控制。

③参数说明

参数地址	说明	类型	数值范围	备注
Mode	设置通讯模式:为 1 时，使能 Modbus 协议功能；为 0 时恢复系统为 PPI 协议	位		
Baud	设置波特率(bps)	双字	11200，2400，4800，9600，19200，38400，57600，115200	
Parity	设置校验	字节	0--无校验 1--奇校验 2--偶校验	所有设置使用一个停止位。
Timeout	主站等待从站响应的时间，以毫秒为单位	整型	1~32767	典型的设置值为 1000 毫秒（1 秒）。
Done	完成位，初始化完成，此位会自动置 1。	位		
Error	初始化错误代码	字节	0--无错误 1--校验选择非法 2--波特率选择非法 3--模式选择非法	只有在 Done 位为 1 时有效。

MBUS_MSG

①指令名称：MBUS_ MSG



②功能：使用 SM0.0 调用 Modbus RTU 主站读写子程序 MBUS_MSG 指令，First 接通发送一个 Modbus 请求。同一时刻只能有一个读写功能（即 MBUS_MSG）使能。

③参数说明

参数地址	说明	类型	数值范围	备注
First	读写请求位	位		每一个新的读写请求必须使用脉冲触发。
Slave	设置从站地址	字节	1~247	
RW	操作命令	字节	0~读 1~写	
Addr	选择读写的数据类型	双字	00000 至 0xxxx--开关量输出 10000 至 1xxxx--开关量输入 30000 至 3xxxx--模拟量输入 40000 至 4xxxx--保持寄存器	
Count	通讯的数据个数（位或字的个数）	整型		Modbus 主站每一个 MBUS_MSG 指令可读/写的最大数据量为 120 个字。
DataPtr	数据指针，如果是读指令，读回的数据放到这个数据区中；如果是写指令，要写出的数据放到这个数据区中	双字		
Done	完成位，读写功能完成位	位		
Error	错误代码		参见错误代码表	只有在 Done 位为 1 时，错误代码才有效。

错误代码表：

0	无错误
1	响应校验错误
2	未用
3	接收超时（从站无响应）
4	请求参数错误
5	Modbus/自由口未使能
6	Modbus正在忙于其它请求
7	响应错误（响应不是请求的操作）
8	响应CRC校验和错误
101	从站不支持请求的功能
102	从站不支持数据地址
103	从站不支持此种数据类型
104	从站设备故障
105	从站接收了信息，但是响应被延迟
106	从站忙，拒绝了该信息
107	从站拒绝了信息
108	从站存储器奇偶错误

3.3 ETHERNET_SET(V1.2)库

以太网设置指令库 ETHERNET_SET(V1.2)用于设置 PLC 以太网口本地通信参数，无需停止 CPU 即可设置、获取远程以太网 PLC 的 IP 地址、MAC 地址和设备名称。

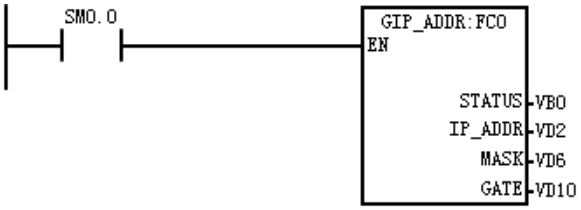
3.3.1 库支持的 PLC 型号

CTH200 系列	H224（基本型、标准型） H226L（基本型、标准型） H226M（标准型） H224X（老平台高性能型、V5 高性能型升级版） H226XL（老平台高性能型、V5 高性能型升级版） H226XM（V5 高性能型升级 H228XL（老平台高性能型） H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版）
CTMC 系列	M228IL M228ML M228SL
CTSC 系列	224I（V5 平台） 226I（V5 平台） 224XP（V5 平台）
CTH300-H 系列	H36-01 H32-01 H35-00 H31-00 H56-10 H52-10

3.3.2 指令详解

GIP_ADDR（获取 IP 地址）

①指令名称：GIP_ADDR



② 功能：获取 IP 地址

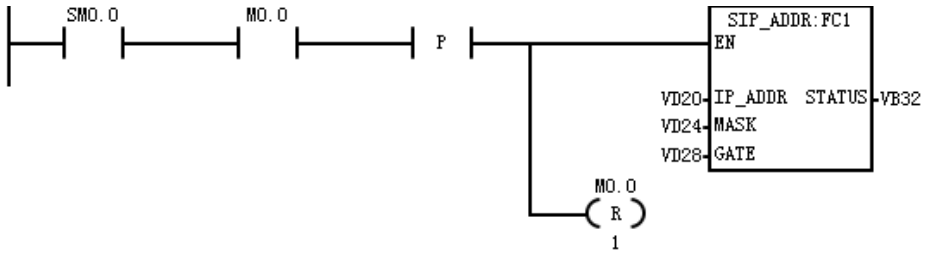
③ 参数说明

参数名称	输入输出属性	类型	说明
EN	IN	BOOL	使能端，允许 SM0.0 调用
STATUS	OUT	BYTE	状态字 bit0=1 表示获取成功 bit1=1 表示获取失败 其他位未使用
IP_ADDR	OUT	DWORD	IP 地址，共四个字节，每个字节由低到高分别表示 IP 地址的对应四个数值。

MASK	OUT	DWORD	子网掩码，共四个字节，每个字节由低到高分别表示子网掩码的对应四个数值。
GATE	OUT	DWORD	网关，共四个字节，每个字节由低到高分别表示网关的对应四个数值。

SIP_ADDR（设置 IP 地址）

①指令名称：SIP_ADDR



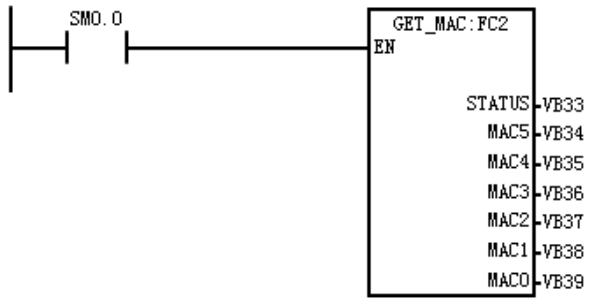
② 功能：设置 IP 地址；

③ 参数说明

参数名称	输入输出属性	类型	说明
EN	IN	BOOL	使能端，用沿触发，设置指令不能循环调用，因为设置 IP 和 EPPROM 一样有次数限制，且不停写以太网将无法通讯
IP_ADDR	IN	DWORD	P 地址，共四个字节，每个字节由低到高分别表示 IP 地址的对应四个数值。
MASK	IN	DWORD	子网掩码，共四个字节，每个字节由低到高分别表示子网掩码的对应四个数值。
GATE	IN	DWORD	网关，共四个字节，每个字节由低到高分别表示网关的对应四个数值。
STATUS	OUT	BYTE	状态字 bit0=1 表示设置成功 bit1=1 表示非法 IP 地址 bit2=1 表示 IP 与掩码不匹配 Bit3=1 表示 IP 与网关不匹配 其他位未使用

GET_MAC（获取 MAC 地址）

①指令名称：GET_MAC



② 功能：获取 MAC 地址；

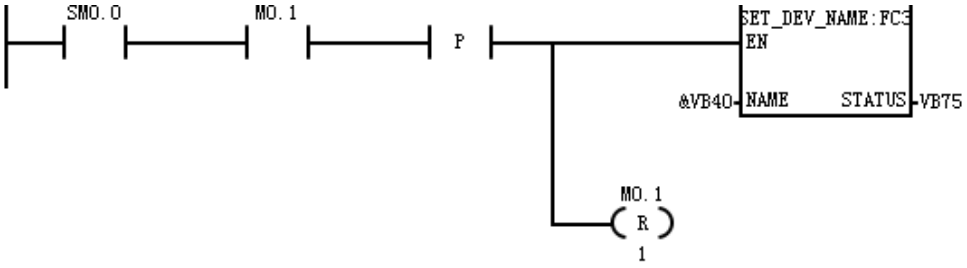
③参数说明

参数名称	输入输出属性	类型	说明
EN	IN	BOOL	使能端，允许 SM0.0 调用

STATUS	OUT	BYTE	状态字 bit0=1 表示获取成功 其他位未使用
MAC0~ MAC5	OUT	BYTE	MAC 地址 xx:xx:xx:xx:xx:xx MAC5------MAC0

SET_DEV_NAME（设置设备名）

①指令名称：SET_DEV_NAME



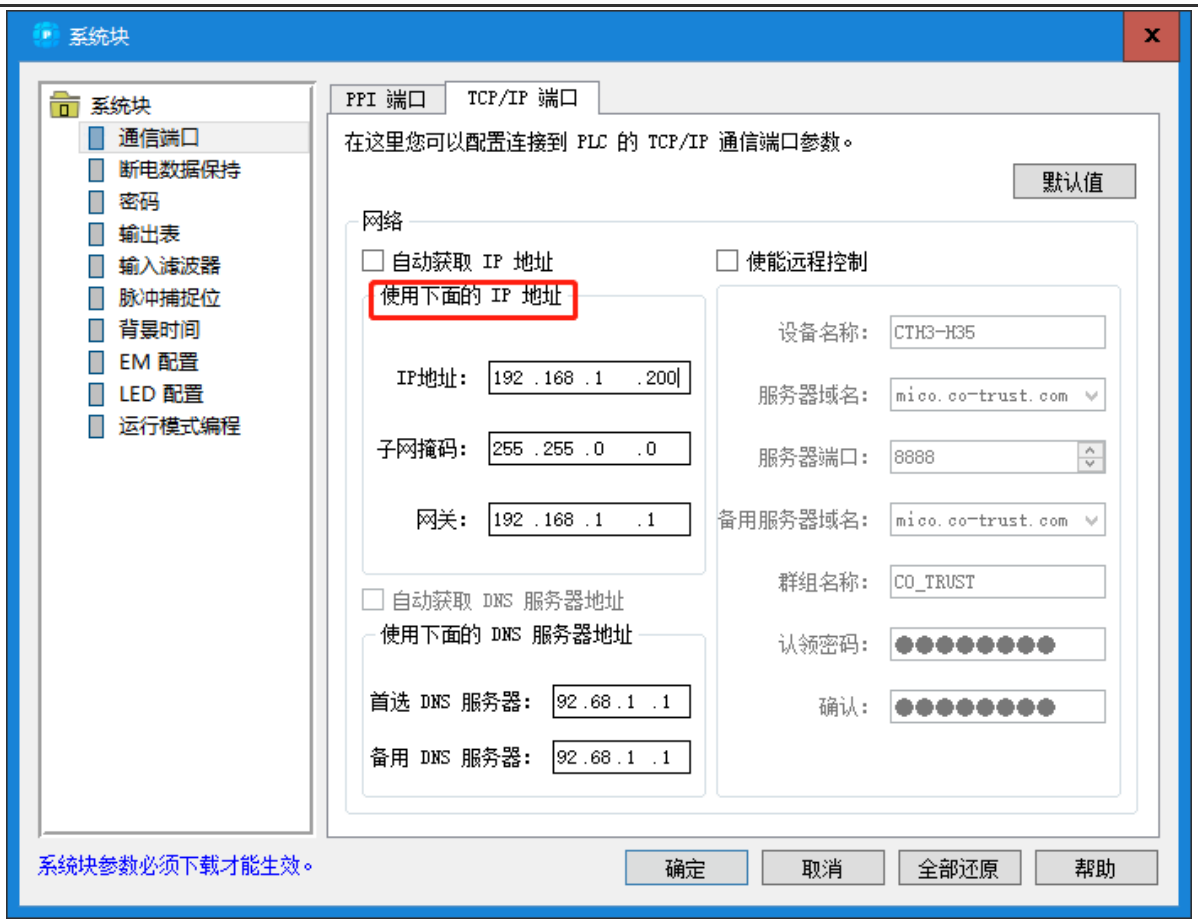
② 功能：设置设备名称；

③参数说明

参数名称	输入输出属性	类型	说明
EN	IN	BOOL	使能端，用沿触发，设置指令不能循环调用，因为设置 IP 和 EPPROM 一样有次数限制，且不停写以太网将无法通讯
NAME	IN	DWORD	设备名指针，指向区域的第一个字节表示长度。整个字符串包含长度字节最大 32bytes。
STATUS	OUT	BYTE	状态字 bit0 为 1 表示设置成功； bit1 为 1 表示长度错误； bit2 为 1 表示保存失败； bit3 表示非法字符

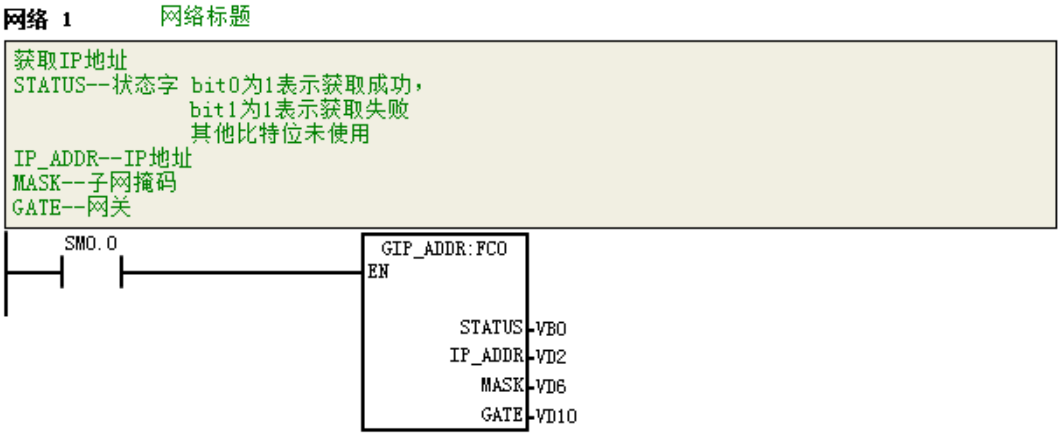
注意事项：

除了库，也可以用 TD4S（V2.05 以上版本）设置 IP。若采用 TD4S 文本屏设置 IP，请在系统块 TCP/IP 端口设置中选择使用静态 IP，不能选择使用自动获取 IP 地址模式，否则设置无法生效。而使用库设置的话就不需要此操作。



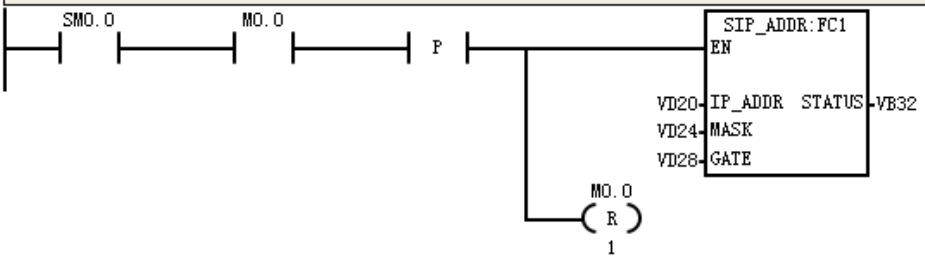
3.3.3 应用示例

以太网指令设置库的应用示例程序如下所示：



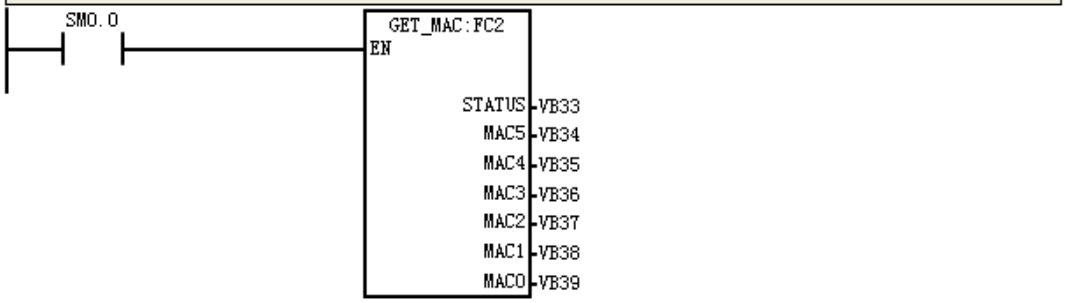
网络 2

```
程序注释:
设置IP地址
IP_ADDR--IP地址
MASK--子网掩码
GATE--网关
STATUS--状态字 bit0为1表示设置完成
               bit1为1表示非法IP地址
               bit2为1表示IP与掩码不匹配
               bit3为1表示IP与网关不匹配
               其他比特位未使用

注意：此设置指令不能循环调用，因为设置IP和EPPROM一样有次数限制，用沿触发
```



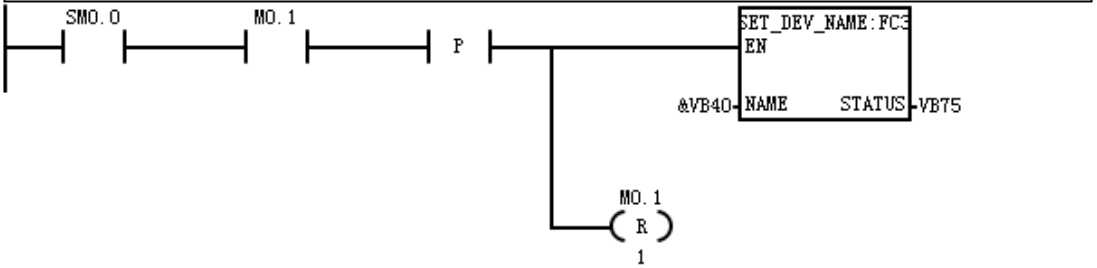
网络 3

```
程序注释
获取MAC地址
STATUS--状态字 bit0为1表示获取成功，其他比特位未使用
MAC0~5--MAC地址
        xx:xx:xx:xx:xx:xx
        MAC5---------MAC0
```



网络 4

```
程序注释
NAME  设备名指针，指向区域的第一个字节表示长度。整个字符串包含长度字节最大32bytes。
STATUS bit0为1表示设置成功；bit1为1表示长度错误；bit2为1表示保存失败；bit3表示非法字符。

注意：此设置指令不能循环调用，因为设置IP和EPPROM一样有次数限制,用沿触发
```



3.4 以太网“socket”通信库

3.4.1 库支持的 PLC 型号

备注：高性能升级版型 H224X/H226XM/H226XL 的固件版本从 V2.29 或以上支持 SOCKET 功能，标准型 H224/H226M/H226L 的固件版本从 V1.29 或以上支持 SOCKET 功能。

CTH200 系列	**ct_socket（v1.1）：** H224（基本型、标准型） H226M（标准型） H226L（标准型） **ct_socket（v1.2）：** H224X（老平台高性能型，V5 高性能型升级版） H226XM（老平台高性能型，V5 高性能型升级版） H226XL（老平台高性能型，V5 高性能型升级版） H228XL（老平台高性能型） H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版）
CTMC 系列	**ct_socket（v1.2）：** M228IL M228ML M228SL
CTSC 系列	**ct_socket（v1.2）：** 224+（V5 平台） 224I V5（V5 平台） 224XP（V5 平台） 226I（V5 平台） 226L（V5 平台） 226M（V5 平台） 226H（V5 平台）
CTH300-H 系列	H36-01 H32-01 H35-00 H31-00 H56-10 H52-10

目前 CPU 支持 2 个 UDP 连接和 2 个 TCP 客户端连接，提供的指令 SOCK_Open、SOCK_Send、SOCK_Recv、SOCK_Close，共 4 条。

约定项	范围	说明
连接号（SockID）	0~255	255 为无效连接号
连接类型（SockType）	0，1	0为UDP连接，1为TCP客户端连接
端口号（Lport/Rport）	1~65535	本地端口号要选用不被其他连接占用的端口号，如系统块里默认端口号20000为PLC监控连接使用。TCP客户端的本地端口号由底层分配，Lport可设置为任何值（UDP模式下不能设为65535），Rport需与服务器端口号一致。

套接字错误码约定：

错误码	说明
0	无错误
1	连接已关闭
2	端口号错误。常用，本地端口号被占用，或端口号超出范围报错
3	设备断线
4	UDP 建立连接失败。常用，创建连接失败，返回连接号 255
5	TCP 客户端连接远方服务端失败
6	TCP 服务端监听失败
7	数据长度错误，最大数据长度为 512 字节
8	数据区为空错误
9	连接号错误
10	接收数据错误（没接收到数据）

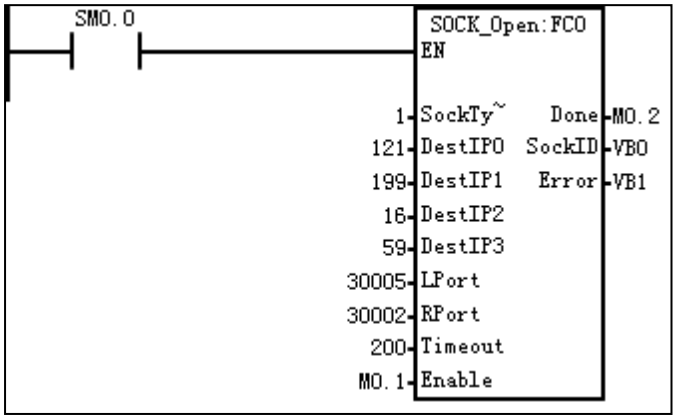
11	连接类型错误
12	IP 信息错误
13	超时错误
14	TCP 客户端需要重连标志。常用，TCP 重连

3.4.2 指令详解

SOCK_Open（创建连接）

① 指令名称：SOCK_Open

② 功能：创建一个连接



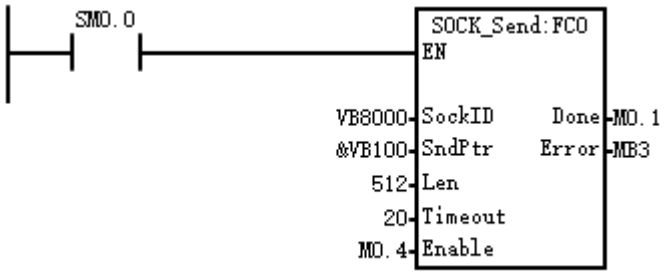
③参数说明

参数地址	说明	变量类型	数据类型	备注
SockTy	链接类型，UDP(0)和 TCP-CLIENT(1)	IN	BYTE	不支持 TCP 服务器连接。
DestIP0	目标IP地址的第1个字节。	IN	BYTE	例如目标地址为192.168.1.202，第一个字节为192。
DestIP1	目标IP地址的第2个字节。	IN	BYTE	例如目标地址为192.168.1.202，第一个字节为168。
DestIP2	目标IP地址的第3个字节。	IN	BYTE	例如目标地址为192.168.1.202，第一个字节为1。
DestIP3	目标IP地址的第4个字节。	IN	BYTE	例如目标地址为192.168.1.202，第一个字节为202。
Lport	本地端口，UDP使用	IN	WORD	TCP-CLIENT模式，LPort由底层随机分配
Rport	目标端口	IN	WORD	与远程服务器端口号一致。
Timeout	超时时间	IN	BYTE	单位：100ms。
Enable	使能位	I/O	BOOL	
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
SockID	输出连接号	OUT	BYTE	底层分配，应用要提供一个全局内存保存连接号。
Error	错误代码	OUT	BYTE	错误码。0：无错

SOCK_Send（发送数据）

① 指令名称：SOCK_Send

② 功能：发送数据

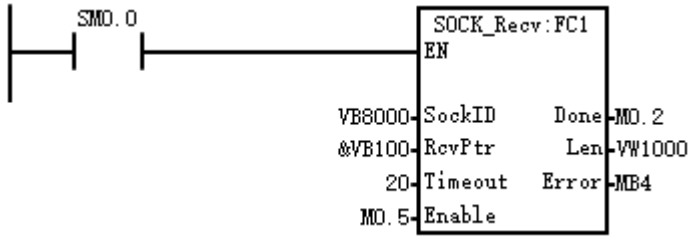


参数地址	说明	变量类型	数据类型	备注
SockID	输出连接号	OUT	BYTE	底层分配，应用要提供一个全局内存保存连接号。
SndPtr	数据缓冲区	IN	DWORD	指向待发送数据的指针，可以指向I、Q、M或V存储器的指针（如&VB100）
Len	待发送字节数	IN	WORD	范围为1~512字节。
Timeout	超时时间	IN	BYTE	单位：100ms。
Enable	使能位	I/O	BOOL	使能
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
Error	错误代码	OUT	BYTE	错误码。0：无错

SOCK_Recv（接收数据）

① 指令名称：SOCK_Recv

② 功能：接收数据



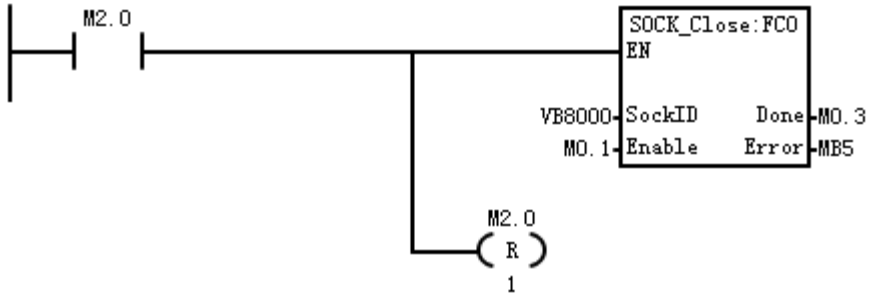
③参数说明

参数地址	说明	变量类型	数据类型	备注
SockID	输出连接号	OUT	BYTE	底层分配，应用要提供一个全局内存保存连接号。
RcvPtr	数据缓冲区	IN	DWORD	指向接收数据存储位置的指针，可以指向I、Q、M或V存储器的指针（如&VB100）
Timeout	超时时间	IN	BYTE	单位：100ms。
Enable	使能位	I/O	BOOL	使能
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
Len	实际接收字节数	IN	WORD	范围1~512字节。
Rport	目标端口	IN	WORD	与远程服务器端口号一致。
Error	错误代码	OUT	BYTE	错误码。0：无错

SOCK_Close（关闭连接）

① 指令名称：SOCK_Close

② 功能：关闭连接

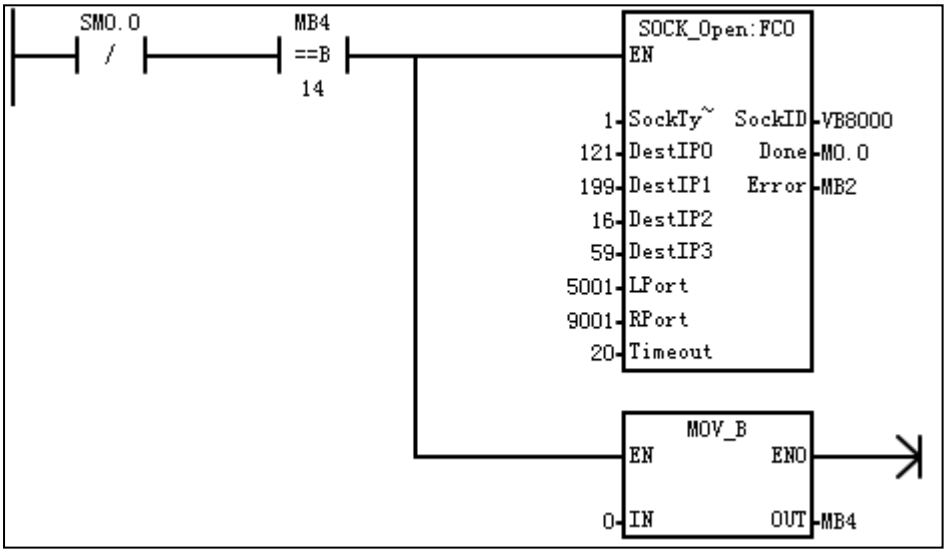


③参数说明

参数地址	说明	变量类型	数据类型	备注
SockID	输出连接号	OUT	BYTE	底层分配，应用要提供一个全局内存保存连接号。
Done	读写功能完成位	OUT	BOOL	使能后自动清零。
Error	错误代码	OUT	BYTE	错误码。0：无错
Enable	使能位	I/O	BOOL	使能

3.4.3 应用示例

TCP 客户端掉线重连，通过发送出错标志来确定是否重新打开连接：重连标志 14



3.5 tcp_server_lib 库

3.5.1 库支持的 PLC 型号

CTH200 系列	H224X（V5 高性能型升级版） H226XM（V5 高性能型升级版） H226XL（V5 高性能型升级版） <备注> PLC 固件版本为 V2.80 以上
CTMC 系列	M228IL M228ML M228SL <备注> PLC 固件版本为 2.80 以上

CTH300-H 系列	H36-01 H32-01 H35-00 H31-00 H56-10 H52-10

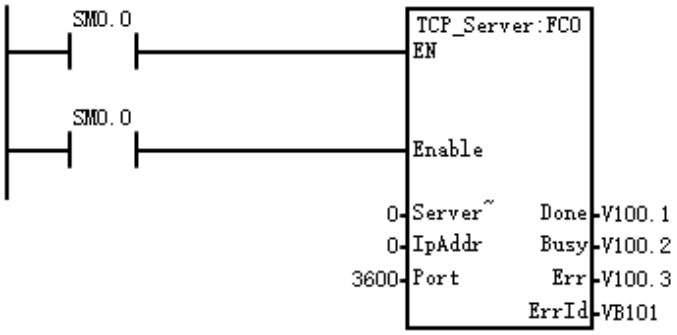
tcp_server_lib 库文件共包含 4 个指令，分别是：TCP_server、TCP_connect、TCP_send、TCP_recv。

<备注> 您可从合信官网免费下载 CT_tcp_server_lib 库：http://www.co-trust.com。

3.5.2 指令详解

TCP_Server

① 指令名称：TCP_Server



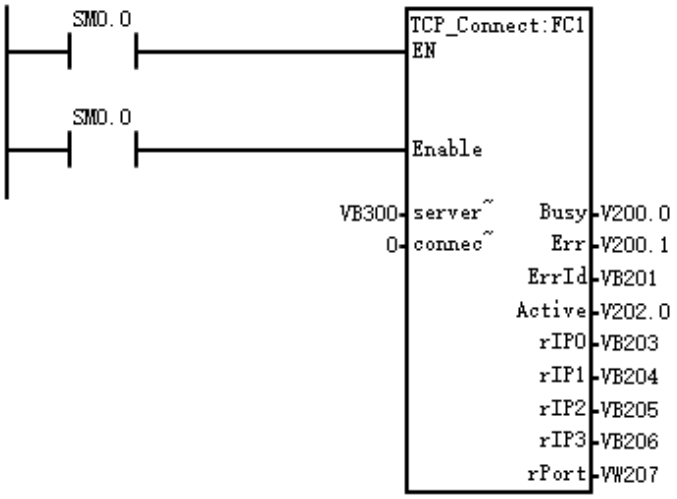
② 功能：此指令用于启动 TCP Server。

③ 参数说明

参数名	输入输出属性	参数描述	类型	备注
Enable	IN	指令执行条件	BOOL	当Enable为1时启动Tcp server Enable为0时停止Tcp server
ServerId	IN	Tcp Server Id号	BYTE	提供2个TCP Server，Service Id为（0，1）
IpAddr	IN	远程连接地址	BYTE	暂时只能取0
Port	IN	Tcp Serve监听的端口号	WORD	
Done	OUT	指令完成位	BOOL	当指令完成或出错时Done置位。 当指令的执行条件Off时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令的执行条件Off时，Busy位被复位。
Err	OUT	指令错误位	BOOL	当指令出错时。Err位置位。 当指令的执行条件Off 时，Err位被复位。
ErrId	OUT	错误代码	BYTE	0. 无错 1. ServerId 错误 2. IpAddr 错误 3. 端口号被占用 当指令的执行条件Off 时，ErrId清0。

TCP_Connect

① 指令名称：TCP_Connect



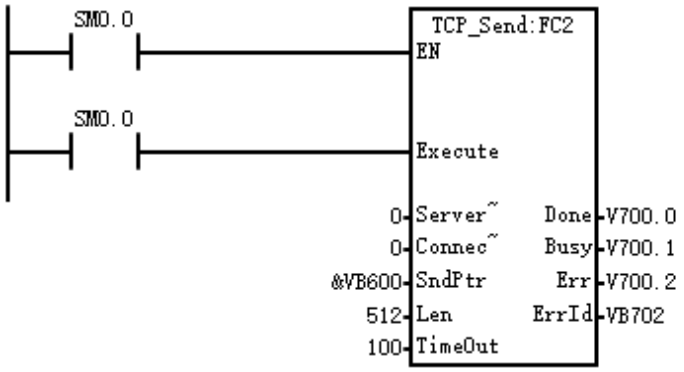
② 功能：此指令用于得到 TCP 连接的状态。

③ 参数说明

参数名	输入输出属性	参数描述	类型	备注
Enable	IN	连接使能位	BOOL	当Enable为1时得到连接的状态。 当Enable下降沿，且处于连接状态，则断开连接。
ServerId	IN	Tcp Server Id号	BYTE	取值0，1
ConnectId	IN	连接Id号	BYTE	每个 Server 最多支持4个连接，connect Id 为（0，1，2，3）。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令的执行条件Off时，连接成功断开，Busy位被复位。
Err	OUT	指令错误位	BOOL	当指令出错时。Err位置位。
ErrId	OUT	错误代码	BYTE	0：无错 1：ServerId 错误 4：ConnectId 错误
Active	OUT	连接状态位	BOOL	当连接建立时，Active置位。 当连接断开时，Active复位。
rIP0	OUT	远程连接的IP地址0	BYTE	
rIP1	OUT	远程连接的IP地址1	BYTE	
rIP2	OUT	远程连接的IP地址2	BYTE	
rIP3	OUT	远程连接的IP地址3	BYTE	
rPort	OUT	远程连接的端口号	WORD	

TCP_Send

① 指令名称：TCP_Send



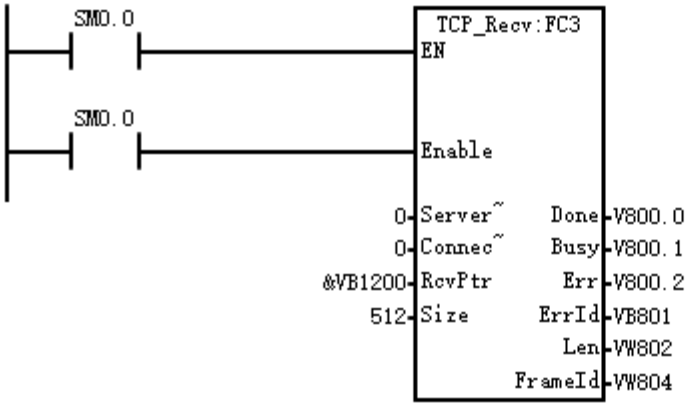
② 功能：此指令用于 Tcp 发送指令，此指令用于向 TCP 连接发送数据。

③ 参数说明

参数名	输入输出属性	参数描述	类型	备注
Excute	IN	指令触发位	BOOL	当Excute上升沿时，触发发送指令。
ServerId	IN	Tcp Server Id号	BYTE	取值0，1
ConnectId	IN	连接Id号	BYTE	每个 Server 最多支持 4 个连接，connect Id 为（0，1，2，3）。
SndPtr	IN	发送指针位	DWORD	
Len	IN	发送数据长度	WORD	最多512字节
Timeout	IN	发送超时时间ms	WORD	
Done	OUT	发送完成位	WORD	数据发送完成时，Done置位。 当指令的执行条件Off时，Done位被复位。 在Execute为0时，发送完成，Done将置位一个周期。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令的执行条件Off时,Busy位被复位。
Err	OUT	指令错误位	BOOL	当指令出错时。Err位置位。
ErrId	OUT	错误代码	BYTE	0：无错 1：ServerId 错误 4：ConnectId 错误 5：Server没有打开 6：连接状态错误 7：发送超时

TCP_Recv

① 指令名称：TCP_Recv



② 功能：此指令用于从 Tcp 连接中接收数据。

③ 参数说明

参数名	输入输出属性	参数描述	类型	备注
Enable	IN	接收使能位	BOOL	Enable为1时，指令处于接收数据的状态。 Enable为0时，指令不接收数据。
ServerId	IN	Tcp Server Id号	BYTE	取值0，1
ConnectId	IN	连接Id号	BYTE	每个 Server 最多支持 4 个连接，connect Id 为（0，1，2，3）。
RcvPtr	IN	接收数据指针位	DWORD	
Size	IN	接收数据最大长度	WORD	最多512字节
Done	OUT	接收完成位	WORD	接收到数据时Done置位。 当指令的执行条件Off时,Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令的执行条件Off 时,Busy位被复位。
Err	OUT	指令错误位	BOOL	当指令出错时。Err位置位。
ErrId	OUT	错误代码	BYTE	0：无错 1：ServerId 错误 4：ConnectId 错误 5：Server没有打开 6：连接状态错误 7：发送超时
FrameId	OUT	接收到数据的帧ID	WROD	每当接到一帧数据，FrameId会增加1，用户可以依照这个来判断是否有新的数据到来。

3.6 S7_Protocol(v1.0)库

3.6.1 库支持的 PLC 型号

CTH200 系列	H224X（V5 高性能型升级版） H226XM（V5 高性能型升级版） H226XL（V5 高性能型升级版）

	H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版） <备注> PLC 固件版本为 V2.80 以上）
CTMC 系列	M228IL M228ML M228SL
CTSC 系列	224I（V5 平台） 224XP（V5 平台） 226I（V5 平台）
CTH300-H 系列	H36-01 H32-01 H56-10 H52-10

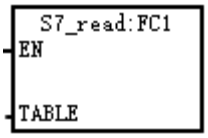
S7 协议库共有以下指令：

- S7_read
- S7_write

3.6.2 指令详解

S7_read

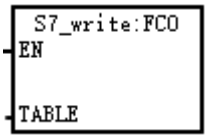
① 指令名称：S7_read



② 功能：S7 主站进行网络读操作

S7_write

① 指令名称：S7_write



② 功能：S7 主站进行网络写操作

③ 参数说明

S7_read/S7_write 指令的 TABLE 参数表：

D	A	E	0	错误代码	0
远程站 IP 第一个字节					1
远程站 IP 第二个字节					2
远程站 IP 第三个字节					3
远程站 IP 第四个字节					4
保留（必须设为 0）					5

保留（必须设为 0）	6
指向远程站目标区域指针的第 1 个字节	7
指向远程站目标区域指针的第 2 个字节	8
指向远程站目标区域指针的第 3 个字节	9
指向远程站目标区域指针的第 4 个字节	10
长度（不大于 200）	11
指向本地站目标区域指针的第 1 个字节	12
指向本地站目标区域指针的第 2 个字节	13
指向本地站目标区域指针的第 3 个字节	14
指向本地站目标区域指针的第 4 个字节	15

D:完成位，0 未完成，1 完成

A:激活位，指令正在执行中为 1，未执行为 0

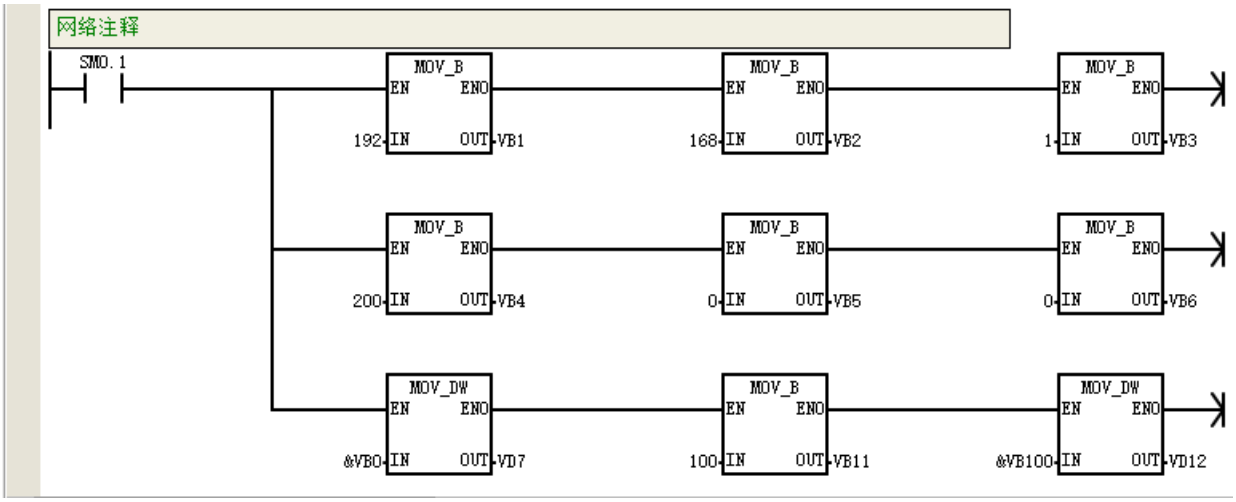
E:位，0 无错，1 出错

错误代码表

代码	说明
0	无错
1	超时错误：远程站不应答
2	保留
3	脱机错误：重复站址或故障硬件引起的冲突
4	队列溢出错误：8 个 S7_write/S7_read 方框被激活
5	接收错误：收到的回复有错误
6	非法参数：S7_write/S7_read 表格包含一个非法或无效数值
7	保留
8	保留
9	信息错误：数据长度不正确
10	连接数超过 8 个
11	网线没插

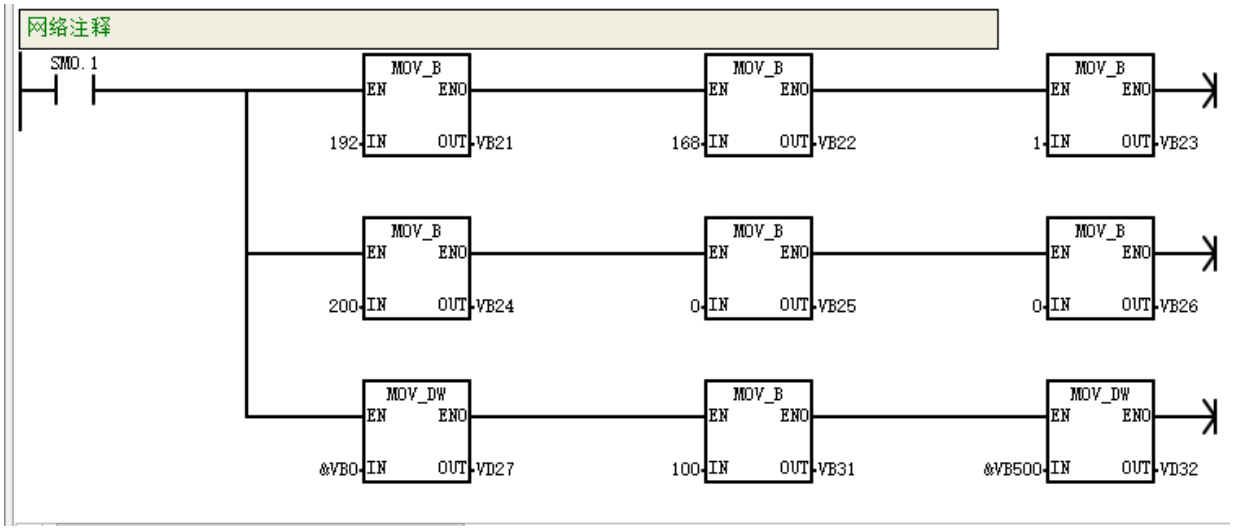
3.6.3 应用示例

网络 1：根据 S7_read/S7_write 指令的 TABLE 参数表将远程站的 IP 设为 192.168.1.200，远程站目标区域指针设为&VB0（一定要用指针）,读的长度设置为 100 字节,本地站目标区域指针设为&VB100（一定要用指针），配置好参数表。

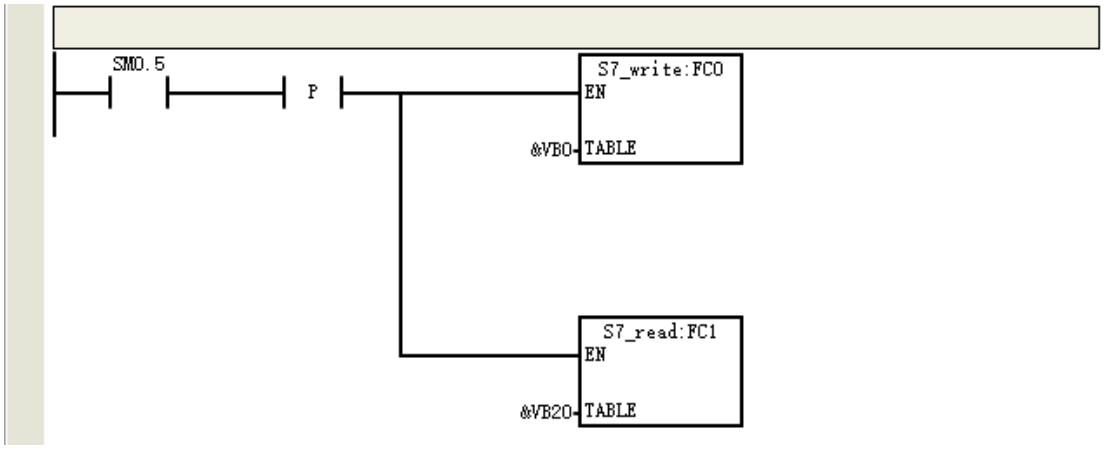


网络 2：根据 S7_read/S7_write 指令的 TABLE 参数表，配置好参数表。将远程站的 IP 设为 192.168.1.200，

远程站目标区域指针设为&VB0（一定要用指针）,写的长度设置为 100 字节,本地站目标区域指针设为&VB100（一定要用指针）。



网络 3：调用 S7_read/S7_write 指令，通过参数表的参数进行读写数据，指令的 TABLE 参数务必使用指针。



3.7 canopen_lib 库

本库中提供 SDO 的读写指令，仅用于对 CTH300-H 系列 PLC 本机的 CAN 口，对 CAN1M 模块不起作用。

使用 SDO 指令可以对从站用参数进行读写操作，由于 SDO 传输效率低，频繁调用 SDO 会增加 CAN 传输的负载，建议用于不频繁改变的参数。

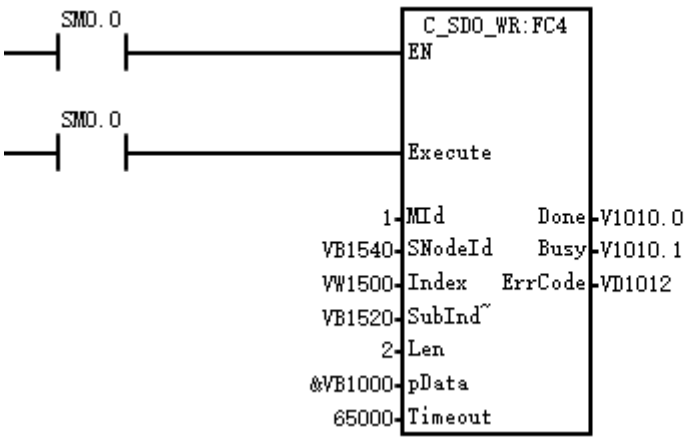
3.7.1 库支持的 PLC 型号

CTH200 系列	H224X（V5 高性能型升级版） H226X（V5 高性能型升级版） H226XL（V5 高性能型升级版） H226IM（I6 高性能型升级版，固件版本为 V2.15 以上） H226IL（I6 高性能型升级版，固件版本为 V2.15 以上） H226IH（I6 高性能型升级版，固件版本为 V2.15 以上）
CTMC 系列	M228IL M228ML M228SL
CTSC 系列	224+（V5 平台） 224I（V5 平台） 224XP（V5 平台） 226I（V5 平台） 226M（V5 平台）） 226L（V5 平台） 226H（V5 平台） <备注> （PLC 固件版本为 V2.15 以上）
CTH300-H 系列	H36-01 H32-01 H56-10 H52-10

3.7.2 指令详解

C_SDO_WR (CanOpen SDO 写指令)

① 指令名称：C_SDO_WR



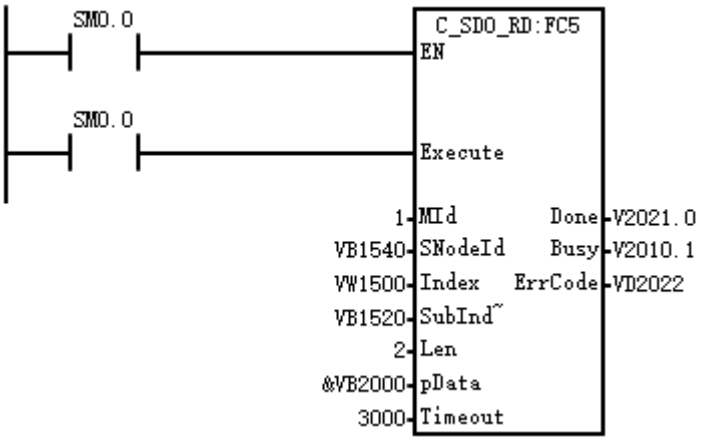
② 功能：此指令使用 CanOpen 的 SDO 功能对从站的参数进行写操作。

参数说明

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
MId	IN	主站Id	BYTE	必须为1
SNodeId	IN	从站的节点Id	BYTE	取值范围1-127
Index	IN	参数索引	WORD	
SubIndex	IN	子索引	BYTE	
Len	IN	长度（字节）	BYTE	
pData	IN	数据指针	DWORD	
Timeout	IN	协议超时时间（100us）	WORD	当设为0时，使用默认时间
Done	OUT	指令完成位	BOOL	当指令完成或出错时Done置位。 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
ErrCode	OUT	中止代码	DWORD	指令中止代码。 见附表1

C_SDO_RD (CanOpen SDO 读指令)

① 指令名称：C_SDO_RD



② 功能：此指令使用 CanOpen 的 SDO 功能对从站的参数进行读操作 。

③ 参数说明

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
MId	IN	主站Id	BYTE	必须为1

SNodeId	IN	从站的节点Id	BYTE	取值范围1-127
Index	IN	参数索引	WORD	
SubIndex	IN	子索引	BYTE	
Len	IN	长度（字节）	BYTE	
pData	IN	数据指针	DWORD	
Timeout	IN	协议超时时间（100us）	WORD	当设为0时，使用默认时间
Done	OUT	指令完成位	BOOL	当指令完成或出错时Done置位。 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
ErrCode	OUT	中止代码	DWORD	指令中止代码。 见附表1

附表 1

中止代码	
0	无错误
05 03 00 00	触发位没有交替改变
05 04 00 00	SDO 协议超时
05 04 00 01	非法或未知的 Client/Server 命令字
05 04 00 02	无效的块大小（仅 Block Transfer 模式）
05 04 00 03	无效的序号（仅 Block Transfer 模式）
05 03 00 04	CRC 错误（仅 Block Transfer 模式）
05 03 00 05	内存溢出
06 01 00 00	对象不支持访问
06 01 00 01	试图读只写对象
06 01 00 02	试图写只读对象
06 02 00 00	对象字典中对象不存在
06 04 00 41	对象不能够映射到 PDO
06 04 00 42	映射的对象的数目和长度超出 PDO 长度
06 04 00 43	一般性参数不兼容
06 04 00 47	一般性设备内部不兼容
06 06 00 00	硬件错误导致对象访问失败
06 06 00 10	数据类型不匹配，服务参数长度不匹配
06 06 00 12	数据类型不匹配，服务参数长度太大
06 06 00 13	数据类型不匹配，服务参数长度太短
06 09 00 11	子索引不存在
06 09 00 30	超出参数的值范围（写访问时）
06 09 00 31	写入参数数值太大
06 09 00 32	写入参数值太小
06 09 00 36	最大值小于最小值
08 00 00 00	一般性错误
08 00 00 20	数据不能传送或保存到应用

08 00 00 21	由于本地控制导致数据不能传送或保存到应用
08 00 00 22	由于当前设备状态导致数据不能传送或保存到应用
08 00 00 23	对象字典动态产生错误或对象字典不存在

运控指令库

4

4.1 motion_ctrl_lib(v1.5)运控库

motion_ctrl_lib(v1.5)运控库无需复杂编程，只需要调用和设置一些简单的参数就可以使用该控制库。支持该库的 PLC 型号均具有 6 轴独立控制功能，轴 0~3 为 1 组，任意两轴可进行插补输出，组内同时只能进行一条插补指令；轴 4、5 为 2 组，4、5 轴可单独进行插补输出。两组配置相互独立，可以同时运行。

4.1.1 库指令中运动轴与 CPU 的 I/O 对应关系

普通 IO	Q0.0	Q0.1	Q0.2	Q0.3	Q0.4	Q0.5	Q0.6	Q0.7	Q1.0	Q1.1	Q1.2	Q1.3
CTMC 系列：M228IL	Pulse_0	Dir_0	Pulse_1	Dir_1	Pulse_2	Dir_2	Pulse_3	Dir_3	Pulse_4	Dir_4	Pulse_5	Dir_5
CTSC 系列：226H **CTH200 系列：H226IH**	Pulse_0	Dir_0	Pulse_1	Dir_1	Pulse_2	Dir_2	Pulse_3	Dir_3	--			
CTSC 系列：224E 晶体管型 **226M-CAN** 晶体管型 **224I** 晶体管型 **226I** 晶体管型 **CTH200 系列：H224X** **H226XL** **H228XL**	Pulse_0	Pulse_1	Dir_0	Dir_1	--							
CTH200 系列：H224 、H226L **H226M**	Pulse_0	Pulse_1	Pulse_2	--								
CTH200 系列：H226XM **H226IM** **H226IL**	Pulse_0	Pulse_1	Pulse_2	Dir_0	Dir_1	Dir_2	--					

提示

1）Pulse_0 --------0 轴脉冲输出； Dir_0 ------0 轴方向输出；
Pulse_1 --------1 轴脉冲输出； Dir_1 ------1 轴方向输出；
Pulse_2 --------2 轴脉冲输出； Dir_2 -------2 轴方向输出；
Pulse_3 --------3 轴脉冲输出； Dir_3 -------3 轴方向输出；

2）对于支持 PTO 的型号，Q0.0 和 Q0.1 在使用运控库时，不能同时使用 PTO 功能

3)为了避免 IO 中断与高速计数器的冲突,我司将 CTSC 系列 226H 的 IO 输入中断由原来的 I0.0、I0.1、I0.2、I0.3 改为相应的 I0.0、I0.1、I1.1、I1.5。

4.1.2 中断事件表

CTH200、CTSC 支持的中断事件

优先级别群组	中断事件号码	优先级别组	说明
通讯和诊断事件（最高优先级）	8	0	端口0：接收字符
	9	0	端口0：传输完成
	23	0	端口0：接收信息完成
	24	0	端口1：接收信息完成
	25	0	端口1：接收字符

	26	0	端口1：传输完成
IO 中断+高速计数器中断（中等优先级）	0	1	上升边缘，I0.0
	2	1	上升边缘，I0.1
	4	1	上升边缘，I0.2
	6	1	上升边缘，I0.3
	1	1	下降边缘，I0.0
	3	1	下降边缘，I0.1
	5	1	下降边缘，I0.2
	7	1	下降边缘，I0.3
	12	1	HSC0 CV=PV
	27	1	HSC0方向改变
	28	1	HSC0外部复原 / Zphase
	13	1	HSC1 CV=PV
	14	1	HSC1方向改变
	15	1	HSC1外部复原
	16	1	HSC2 CV=PV
	17	1	HSC2方向改变
	18	1	HSC2外部复原
	32	1	HSC3 CV=PV
	29	1	HSC4 CV=P
	30	1	HSC4方向改变
	31	1	HSC4外部复原 / Z phase
	33	1	HSC5 CV=PV
PTO 中断（中等优先级）	19	1	PTO 0 完成中断
	20	1	PTO 1 完成中断
定时（最低优先级）	10	2	定时中断0
	11	2	定时中断1
	21	2	定时器 T32 CT=PT 中断
	22	2	定时器 T96 CT=PT 中断

CTMC 支持的中断事件

中断号	中断名	触发条件	特别说明
0	I0.0 上升沿中断	I0.0 检测到上升沿	
1	I0.0 下降沿中断	I0.0 检测到下降沿	
2	I0.1 上升沿中断	I0.1 检测到上升沿	
3	I0.1 下降沿中断	I0.1 检测到下降沿	
4	I0.2 上升沿中断	I0.2 检测到上升沿	
5	I0.2 下降沿中断	I0.2 检测到下降沿	
6	I0.3 上升沿中断	I0.3 检测到上升沿	
7	I0.3 下降沿中断	I0.3 检测到下降沿	
8	PORT0 接收数据	PORT0 收到一个字节数据	
9	PORT0 发送完成	PORT0 发送完缓冲区的数据	
10	定时中断 0	给定的定时 0 信号到达	
11	定时中断 1	给定的定时 1 信号到达	
12	HSC0 设定值中断	计数值=设定值	
13	HSC1 设定值中断	计数值=设定值	
14	HSC1 方向中断	计数方向改变	
15	HSC1 外部复位中断	外部复位信号有效	
16	HSC2 设定值中断	计数值 = 设定值	
17	HSC2 方向中断	计数方向改变	
18	HSC2 外部复位中断	外部复位信号有效	
19	PTO 0 输出完成	PTO 给定脉冲输出完成	不支持
20	PTO 1 输出完成	PTO 给定脉冲输出完成	不支持
21	T32	T32 时间到	
22	T96	T96 时间到	
23	PORT0 信息接收完成	PORT0 按设定条件完成接收	

24	PORT1 信息接收完成	PORT1 按设定条件完成接收	
25	PORT1 字符接收	PORT1 收到一个字节数据	
26	PORT1 发送完成	PORT1 发送完缓冲区的数据	
27	HSC0 方向中断	计数方向改变	
28	HSC0 外部复位中断	外部复位信号有效	
29	HSC4 设定值中断	计数值 = 设定值	
30	HSC4 方向中断	计数方向改变	
31	HSC4 外部复位中断	外部复位信号有效	
32	HSC3 设定值中断	计数值=设定值	
33	HSC5 设定值中断	计数值=设定值	
38	HSC3 方向中断	计数方向改变	新增
39	HSC5 方向中断	计数方向改变	新增
40	HSC3 外部复位中断	外部复位信号有效	新增
41	HSC5 外部复位中断	外部复位信号有效	新增
42	HSC0 捕获中断	外部捕获信号有效	新增
43	HSC1 捕获中断	外部捕获信号有效	新增
44	HSC2 捕获中断	外部捕获信号有效	新增
45	HSC3 捕获中断	外部捕获信号有效	新增
46	HSC4 捕获中断	外部捕获信号有效	新增
47	HSC5 捕获中断	外部捕获信号有效	新增

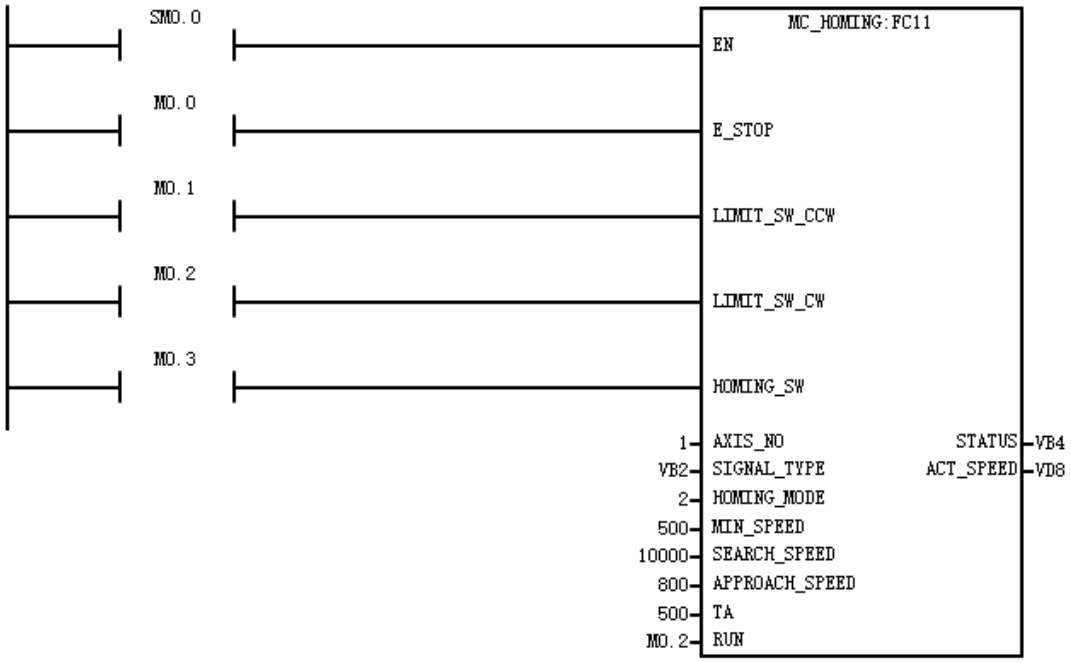
4.1.3 库支持的 PLC 型号

指令类别	函数名	指令名	适用的PLC型号
通用指令	MC_HOMING	回原指令	**CTH200 系列：**H226IH、H224X、H226XL、H226XM、H228XL、H226IM、H226IL。 **CTMC 系列：**M228IL **CTSC系列：**224E 晶体管型、224I 晶体管型、226M-CAN晶体管型、226I 晶体管型、226H (CTS7 216- 1AH34-0B24、CTS7 216-1AH35-0B24、CTS7 216- 1AH34-1B24、CTS7 216- 1AH34-2B24、CTS7 216- 1AH35-2B24) **备注：**H224、H226L、H226M 仅支持 MC_PTP_R 和 MC_PTP_A 224E 晶体管型、224I 晶体管型、226M-CAN 晶体管型、226I 晶体管型、H224X、H226XL、H226XM、H228XL H226IM、H226IL 不支持 MC_SET_POS_PV
	MC_INIT_DIR	配置电机方向指令	
	MC_EXT_RESET_EN	外部复位坐标使能指令	
	MC_EXT_RESET_EN_EXT	外部复位坐标使能指令 II	
	MC_SET_POS_ZERO	软件清零指令	
	MC_SET_POS_PV	设置目标位置指令	
	MC_SET_MAX_ACCELE	设置最大加速度指令	
	MC_READ_POS	读位置指令	
	MC_SPEED_CTL	速度控制指令	
	MC_PTP_R	单轴相对运动指令	
	MC_PTP_A	单轴绝对运动指令	
	MC_STOP_CTL	控制停止指令	
直线插补	MC_LINE_R	两轴直线插补相对运动指令	**CTH200 系列** H226IH **CTMC 系列** M228IL H226 CTS7 216- 1AH34-1B24 CTS7 216- 1AH34-2B24 CTS7 216- 1AH35-2B24
	MC_LINE_A	两轴直线插补绝对运动指令	
圆弧插补	MC_CIRCLE_R	两轴圆弧插补相对运动指令	**CTH200 系列** H226IH **CTMC 系列** M228IL H226 CTS7 216- 1AH34-2B24、CTS7 216-1AH35-2B24
	MC_CIRCLE_A	两轴圆弧插补绝对运动指令	
	MC_3P_CIRCLE_R	两轴三点画弧插补相对运动指令	
	MC_3P_CIRCLE_A	两轴三点画弧插补绝对运动指令	
	MC_SET_CI_MODE	设置连续插补功能指令	
专用指令	MC_SET_PWM	脉冲宽度调制指令	该指令仅有 CTS7 216-1AH35-0B24、CTS7 216- 1AH35-2B24（固件版本为 2.80 以上）支持。

4.1.4 指令详解

MC_HOMING（回原指令）

① 函数名：MC_HOMING



② 功能：通过设置回原模式等参数，可寻找设备原点。

轴号与外部复位 IO 信号的对应关系如下

CPU 型号	轴 0	轴 1	轴 2	轴 3	轴 4	轴 5
CTMC 系列： M228IL	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I2.0（HSC3，SM137.0）	I0.5（HSC4，SM147.0）	I2.4（HSC5，SM157.0）
CTH200 系列： H226IH H224X H226XL H226XM H228XL H226IM H226IL **CTSC 系列：** **224E** 晶体管型 **226M-CAN** 晶体管型 **224I** 晶体管型 **226I** 晶体管型 **CTSC 系列：** **226H**	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I0.5（HSC4，SM147.0）	--	--

若回原模式以原点开关为参考时（回原模式 3 或 4），必须将原点开关信号（指令的 HOMING_SW 参数）接至上述对应点，否则无法找到原点。

Z pulse 即为 HSC 复位信号，详见高速计数器说明。

高速计数器说明

项目		说明
计数器总数		6
单相计数器		6×200KHz
双相计数器		6×100KHz
电流（大于3mA 且电压低于30V）	单相	200KHz
	双相	100KHz

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效，0：无效	BOOL	0~1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当 E_STOP 为 1 时，RUN 内部复位。
LIMIT_SW_CCW	IN	CCW 逆时针行程限位输入	BOOL	0~1	
LIMIT_SW-CW	IN	CW 顺时针行程限位输入	BOOL	0~1	
HOMING_SW	IN	原点信号输入	BOOL	0~1	
AXIS_NO	IN	轴号	BYTE	0~3	该参数在运行过程中不可修改。
SIGNAL_TYPE	IN	7 6 5 4 3 2 1 0 信号类型 Bit0：逆时针行程限位输入信号类型 0—高电平 1—低电平 Bit1：顺时针行程限位输入信号类型 0—高电平 1—低电平 Bit2：原点开关信号类型 0—高电平 1—低电平	BYTE	0~255	
HOMING_MODE	IN	回原模式	BYTE	1~14	模式说明详见章节回原功能
MIN_SPEED	IN	最小速度。 单位：Hz	DWORD	0~200000	1、当速度小于 5Hz 时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、搜索速度不应太大，接近速度应尽量小。
SEARCH_SPEED	IN	原点搜索速度。 单位：Hz	DWORD	0~200000	
APPROACH_SPEED	IN	原点接近速度。 单位：Hz	DWORD	0~200000	
TA	IN	加减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、具体指令的加速度见**提示 3**。
RUN	IN	运行使能位 1：有效	BOOL	0~1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。
STATUS	OUT	7 6 5 4 3 2 1 0 输出状态字节： Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行 0—不运行 Bit2：完成标志 1—完成，指令执行完毕	BYTE	0~255	Bit0： 1、只对轴参数配置错误和回原模式超范围进行判断。 2、其它参数不作报错，会自动设置成一个最接近的合理值。 3、若 TA=0 若没有设置最大加速度，则报参数故障；TD 亦然。

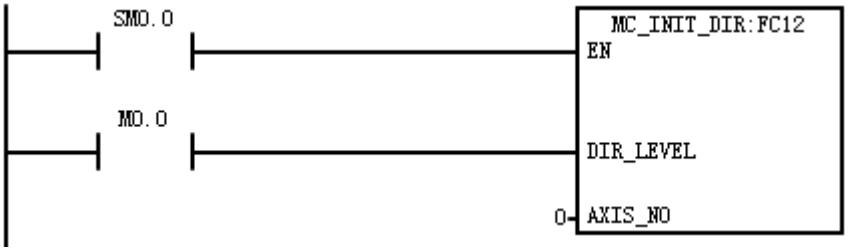
		0—未完成，指令未执行 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或执行已完成。 Bit4：预留。 Bit5：是否找到原点 1—找到原点 0—没找到原点			
ACT_SPEED	OUT	当前速度	DWORD	0~200000	

④ 说明

程序对各输入的检测以扫描方式实现（Z 相信号不受此影响），故当开关量变化时处理不及时，可能有些延迟。若回原速度（包括搜索速度和接近速度）太大时，这个处理延迟被放大，导致回原不准。

MC_INIT_DIR（配置电机方向指令）

① 函数名：MC_INIT_DIR



② 功能：配置电机的方向。

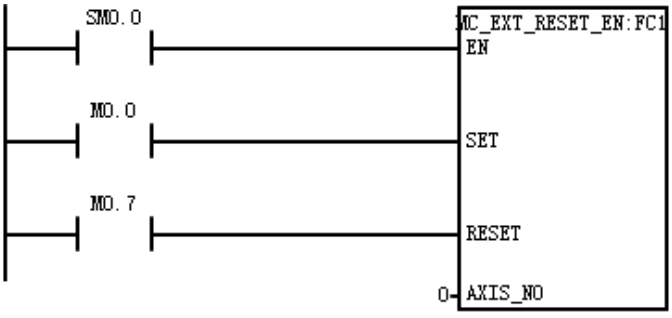
建议此指令只在 CPU 上电第一个扫描周期执行一次。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
DIR_LEVEL	IN	配置方向信号为正向时的有效电平。 DIR_LEVEL 为 1 时，设置对应方向轴输出“1”时为电机正转。 DIR_LEVEL 为 0 时，设置对应方向轴输出“0”时为电机正转。	BOOL	0~1	默认值：1，即默认方向轴输出为“1”时为电机正转。
AXIS_NO	IN	设置轴号	BYTE	0~3	

MC_EXT_RESET_EN（外部复位坐标使能指令）

① 函数名：MC_EXT_RESET_EN



② 功能：当调用该指令，设置是否使能外部 IO 复位绝对坐标值。

轴号与外部复位信号的对应关系：

CPU 型号	轴 0	轴 1	轴 2	轴 3	轴 4	轴 5
CTMC 系列：M228IL	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I2.0（HSC3，SM137.0）	I0.5（HSC4，SM147.0）	I2.4（HSC5，SM157.0）

CTH200 系列： H226IH H224X H226XL H226XM H228XL H226IM H226IL **CTSC 系列：** **224E** 晶体管型 **226M-CAN** 晶体管型 **224I** 晶体管型 **226I** 晶体管型 **CTSC 系列：** **226H**	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I0.5（HSC4，SM147.0）	--	--

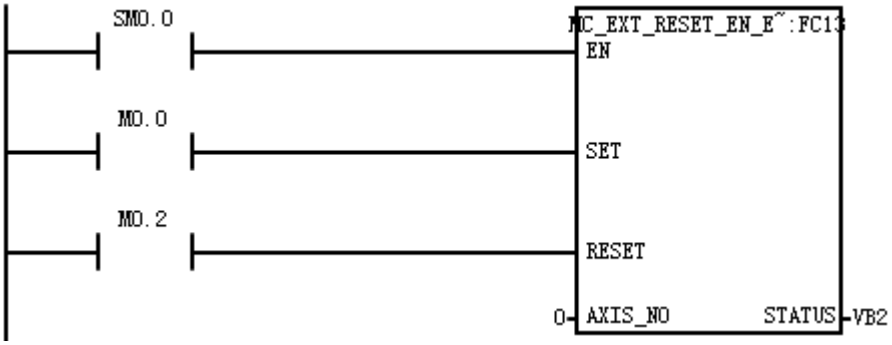
各轴的复位信号输入引脚与对应高速计数器的复位引脚一致，各轴的复位信号有效电平由对应高速计数器的复位有效电平控制位决定，若没有配置对应高速计数器，则默认复位有效电平为高电平。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围
SET	IN	SET 上升沿，设置外部复位使能，每次调用时，SET 应先复位，然后再置 1。	BOOL	0~1
RESET	IN	RESET 上升沿，禁止外部复位使能，每次调用时，RESET 应先复位，然后再置 1。	BOOL	0~1
AXIS_NO	IN	设置轴号	BYTE	0~3

MC_EXT_RESET_EN_EXT（外部复位坐标使能指令 II）

① 函数名：MC_EXT_RESET_EN_EXT



② 功能：当调用该指令，设置是否使能外部 IO 复位绝对坐标值，并查看指令执行状态。

轴号与外部复位信号的对应关系：

CPU 型号	轴 0	轴 1	轴 2	轴 3	轴 4	轴 5
CTMC 系列： M228IL	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I2.0（HSC3，SM137.0）	I0.5（HSC4，SM147.0）	I2.4（HSC5，SM157.0）

CTH200 系列： H226IH H224X H226XL H226XM H228XL H226IM H226IL **CTSC 系列：** **224E** 晶体管型 **226M-CAN** 晶体管型 **224I** 晶体管型 **226I** 晶体管型 **CTSC 系列：** **226H**	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I0.5（HSC4，SM147.0）	--	--

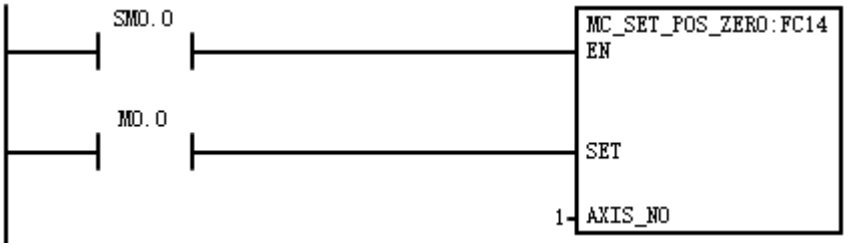
各轴的复位信号输入引脚与对应高速计数器的复位信号引脚一致；各轴的复位信号有效电平由对应高速计数器的复位有效电平控制位决定，若没有配置对应高速计数器，则默认复位有效电平为高电平。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围
SET	IN	SET上升沿，设置外部复位使能，每次调用时，SET应先复位，然后再置1。	BOOL	0~1
RESET	IN	RESET上升沿，禁止外部复位使能，每次调用时，RESET 应先复位，然后再置1。	BOOL	0~1
AXIS_NO	IN	设置轴号	BYTE	0~3
STATUS	OUT	状态位： 7 6 5 4 3 2 1 0 Bit0：复位状态标志位 1—复位完成 0—复位未完成 Bit1~Bit7：预留	BYTE	0~1

MC_SET_POS_ZERO（软件清零指令）

① 函数名：MC_SET_POS_ZERO



② 功能：将绝对坐标复位。

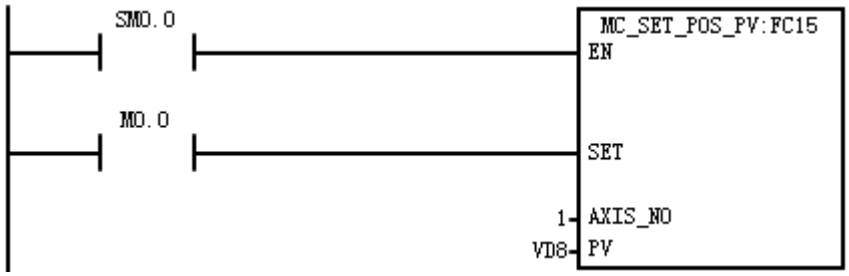
当机器运动到某一位置时，调用该指令，相当于把该轴的原点设定在该位置。那么以后每次调用“读绝对坐标”命令，就能得到相对于该点的坐标值。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围
SET	IN	清零功能使能位。 在SET上升沿把绝对坐标清0，每次调用时，SET应先置0，然后再置1。	BOOL	0~1
AXIS_NO	IN	设置轴号。	BYTE	0~3

MC_SET_POS_PV（设置目标位置指令）

① 函数名：MC_SET_POS_PV



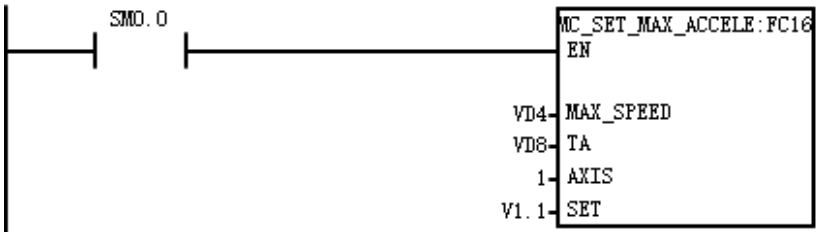
② 功能：此指令用于将机器所处的绝对位置写入到模块。比如，机器运行到某一位置时断电，可将其此时所处的位置保存下来，等下次上电时，将此位置写回到模块，则模块绝对位置计数起点与机器实际起点位置一致，而机器不需回到原点；若此位置刚好为原点，则此指令与 MC_SET_POS_ZERO 效果相同。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围
AXIS_NO	IN	设置轴号。	BYTE	0~3
SET	IN	SET 上升沿，指令使能，每次调用时，SET 应先复位，然后再置 1。	BOOL	0~1
PV	IN	设定的目标位置，分正负。输出正脉冲表示沿 X 轴的正方向，负脉冲数表示沿着 X 轴的负方向。	DINT	-2147483648 ~ +2147483647

MC_SET_MAX_ACCELE（设置最大加速度指令）

① 函数名：MC_SET_MAX_ACCELE



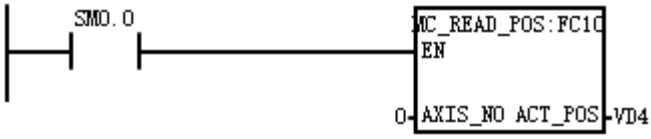
② 功能：设置最大加速度（= MAX_SPEED/TA）（TA≠0）（若没有调用此指令，则认为没有设置最大加速度）

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。 单位：Hz	DWORD	0~200000	运行过程中可以修改。
TA	IN	加速/减速时间， 单位：ms	DWORD	0~10000	运行过程中可以修改；若 TA=0，则认为没有设置最大加速度。
AXIS	IN	设置轴号	BYTE	0~3	此指令无错误状态输出，轴号必须设置正确。
SET	IN	在以上参数确定后，给 SET 一个上升沿以使设置生效。	BOOL	0~1	

MC_READ_POS（读位置指令）

① 函数名：MC_READ_POS



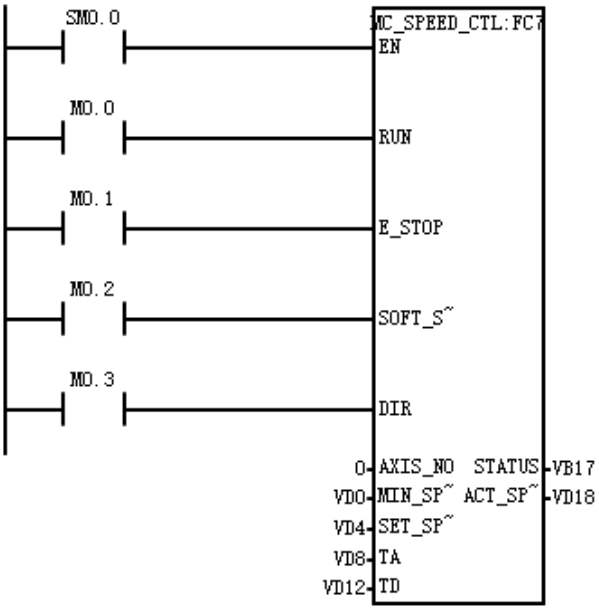
② 功能：读取每轴的绝对坐标值。一旦设定原点坐标后，那么该值会根据输出的脉冲和方向的 关系进行代数计算：正转输出一个脉冲：＋1，反转输出一个脉冲：－1。最后得到的是一个以设定点为原点的绝对坐标。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
AXIS_NO	IN	设置轴号	BYTE	0~3	
ACT_POS	OUT	当前轴的绝对坐标（1个脉冲代表1个单位坐标）	DINT	-2147483648 ~ +2147483647	此指令无错误状态输出，轴号必须设置正确。

MC_SPEED_CTL（速度控制指令）

① 函数名：MC_SPEED_CTL



② 功能：控制单轴输出脉冲的频率，可任意时候改变输出脉冲的频率（速度）。当接收到软停止命令时，会自动减速停止。当收到紧急停止命令时，会马上停止脉冲输出，不经过减速。

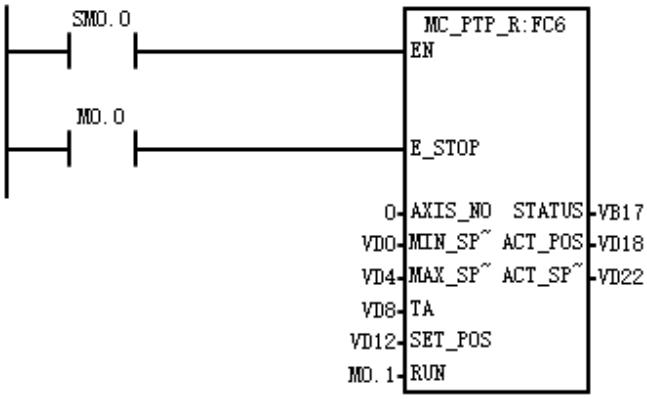
③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
RUN	IN	运行使能位。 1：有效， 0：无效。	BOOL	0/1	只有RUN=1与E_Stop=0且SOFT_STOP=0时才能运行。
E_STOP	IN	紧急停止位。1：有效，0：无效。 当收到有效紧急停止命令后，输出脉冲会马上停止，不经过减速。	BOOL	0/1	只有RUN=1与E_Stop=0且SOFT_STOP=0时才能运行。
SOFT_STOP	IN	软停止位。1：有效，0：无效。 当收到有效软停止命令时，输出脉冲会减速停止	BOOL	0/1	只有RUN=1与E_Stop=0且SOFT_STOP=0时才能运行。
DIR	IN	脉冲的方向位	BOOL	0/1	该参数在运行过程中能修改。
AXIS_NO	IN	设置轴号	BYTE	0~3	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200 000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出； 2、该参数在运行过程中可以修
SET_SP	IN	设定速度，在收到停止命	DWORD	0~200	

EED		令前，输出脉冲会加速或减速到此速度。单位：Hz		000	改。 3、将设定速度写0，可实现软停功能。
TA	IN	加速时间，从最小速度到设定速度的加速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改； 2、加速度只在启动时和TA/TD变化时计算，计算方法详见提示3；
TD	IN	减速时间，从设定速度到最小速度的减速时间。单位：ms	DWORD	0~10000	
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且未执行完。 0—不运行，该因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行没执行或者指令正在执行中但没完成 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用 0—忙标志无效，指令正在执行或此执行完成 Bit4~Bit7：预留	BYTE	0~255	Bit0： • 只对轴参数配置错误进行判断； • MIN_SPEED/SET_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
ACT_SPEED	OUT	当前速度（频率）输出	DWORD	0~200000	

MC_PTP_R（单轴相对运动指令）

① 函数名：MC_PTP_R



② 功能：用作单轴点对点控制（单轴定长驱动）。调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

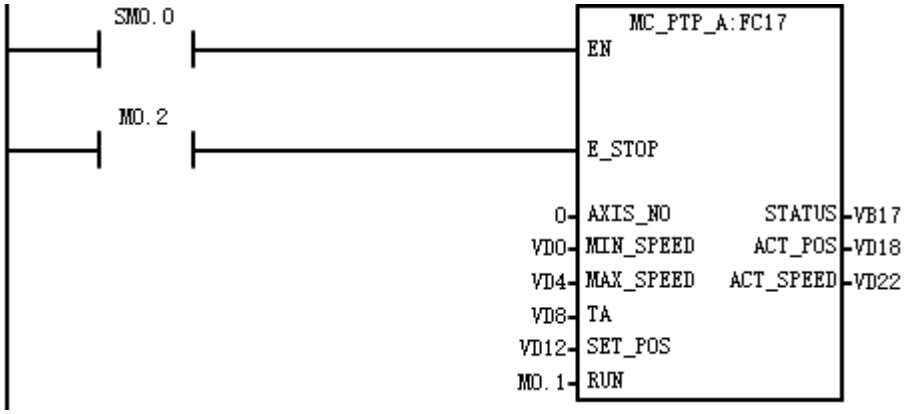
③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有Run=1与E_Stop=0时才能运行。 2、E_STOP 为1时，RUN 内部复位。
AXIS_NO	IN	设置轴号	BYTE	0~3	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。 单位：Hz	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出； 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	最大速度，即运行中的最大速度。 单位：Hz	DWORD	0~200000	
TA	IN	加速 / 减速时间。 单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算，计算方法详见提示3。
SET_POS	IN	输出的脉冲数，分正负。 正脉冲数表示沿 X 轴的正方向，负脉冲数表示沿着 X 轴的负方向（此为相对坐标）	DINT	-2147483648 ~ +2147483647	该参数在运行过程中可以修改，当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。当新设定值小于已输出脉冲数，则会马上停止脉冲输出。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有RUN=1与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当E_STOP为1时，RUN内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没	BYTE	0~255	Bit0 注： 1、只对轴参数进行判断； 2、MIN_SPEED/ MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。

		得以运行；或者指令已经运行完毕 Bit2：完成标志 1—完成，指令执行完毕 0—未完成，指令未执行或指令正在执行中但未完成 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用 0—忙标志无效，指令正在执行或此执行已完成 Bit4~Bit7：预留			
ACT_POS	OUT	当前的相对坐标或本指令已输出的脉冲数	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度	DWORD	0~200000	

MC_PTP_A（单轴绝对运动指令）

① 函数名：MC_PTP_A



② 功能：用作单轴点对点控制（非定长，而是定点）。调用一次可在原脉冲数基础上输出脉冲至指定脉冲数，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

③ 参数说明

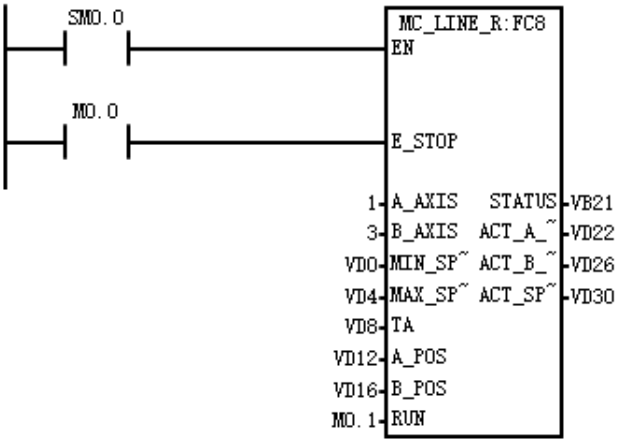
参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有 Run =1 与 E_Stop=0 时才能运行。 2、当E_STOP为1时，RUN 内部复位。
AXIS_NO	IN	设置轴号	BYTE	0~3	该参数在运行过程中不能修改。

MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：Hz。	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议MIN_SPEED 不要小于500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为500）。 4、将MAX_SPEED写0，可实现软停功能（即输出脉冲减速停止），此时使能位RUN不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	
TA	IN	加速/减速时间，单位ms	DWORD	0~10000	1、该参数在运行过 程中可以修改。 2、加速度只在启动时和TA/TD变化时计算方法见提示3。
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿X轴的正方向，负脉冲数表示沿着 X 轴的负方向（此为绝对坐标）	DINT	-2147483648 ~ +2147483647	该参数在运行过程中可以修改，当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。当新设定值小于已输出脉冲数，则会马上停止脉冲输出。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1）只 有 RUN=1 与E_STOP=0 时才能运行。 2、当运行完成后，RUN内部复位。 3、当E_STOP 为1时，RUN 内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误。 0—参数配置正常。 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4~Bit7：预留	BYTE	0~255	Bit0： 1、只对轴参数进行判断。 2）MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。

ACT_POS	OUT	当前的绝对坐标	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度	DWORD	0~200000	

MC_LINE_R（两轴直线插补相对运动指令）

① 函数名：MC_LINE_R



② 功能：可在任意两轴之间、平面上任意区域内进行直线插补功能(设置点为相对坐标)。

③ 参数

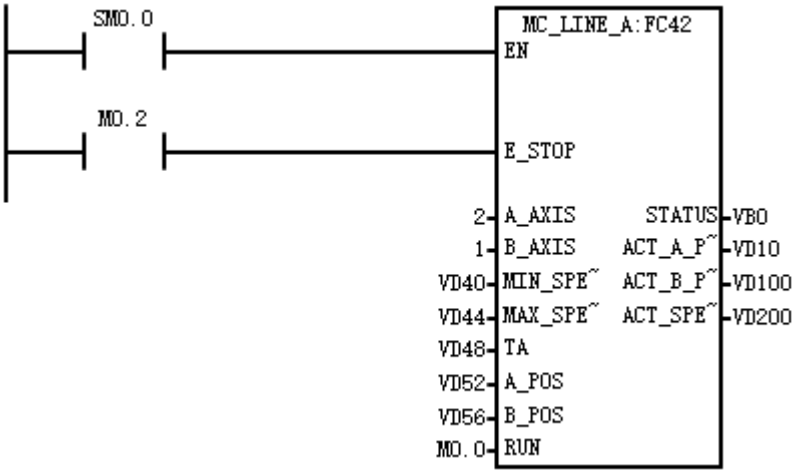
参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0时才能运行。 2、当 E_STOP为 1 时，RUN内部复位。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到相应轴上。	BYTE	0~3	该参数在运行过程中不可修改。
B_AXIS	IN	插补B轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的0、1、2、3轴上。该参数即可设定B轴映射到相应的轴上。	BYTE	0~3	
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议MIN_SPEED不要小于500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为500）。 4、将MAX_SPEED 写0，可实现软停功能（即输出

MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	脉冲减速停止），此时使能位RUN不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
TA	IN	加速/减速时间。单位：ms	DWORD	0~10000	• 该参数在运行过程中可以修改。 • 加速度只在启动时和TA/TD变化时计算方法见提示3。
A_POS	IN	虚拟 A 轴的终点(相对)坐标	DINT	-2147483648 ~ +2147483647	• 该参数在运行过程中不能修改； 2、单位：脉冲，与实际位移转换计算方法详见提示2。
B_POS	IN	虚拟 B 轴的终点(相对)坐标	DINT	-2147483648 ~ +2147483647	
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0时才能运行。 2、当运行完成后，RUN内部复位。 3、当E_STOP为 1时，RUN内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误。 0—参数配置正常。 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，该因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行没执行或者指令正在执行中但没完成。 Bit3：忙标志 1：忙标志有效，直线插补模块或相应的轴被其指令占用。 0：忙标志无效，指令正在执行或此执行完成。 Bit4~Bit7：预留	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 2、MIN_SPEED /MAX_SPEED 等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
ACT_A_POS	OUT	A 轴的当前位置(相对坐标，本次调用实际输出脉冲数)，如果A轴配给0轴，那么该值就表示0轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B轴的当前位置(相对坐标，本次调用实际输出脉冲数)，如果B轴配给1轴，	DINT	-2147483648 ~	

		那么该值就表示1轴的相对坐标。		+2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_LINE_A（两轴直线插补绝对运动指令）

① 函数名：MC_LINE_A



② 功能：可在任意两轴之间、平面上任意区域内进行直线插补功能（设置点为绝对坐标）。

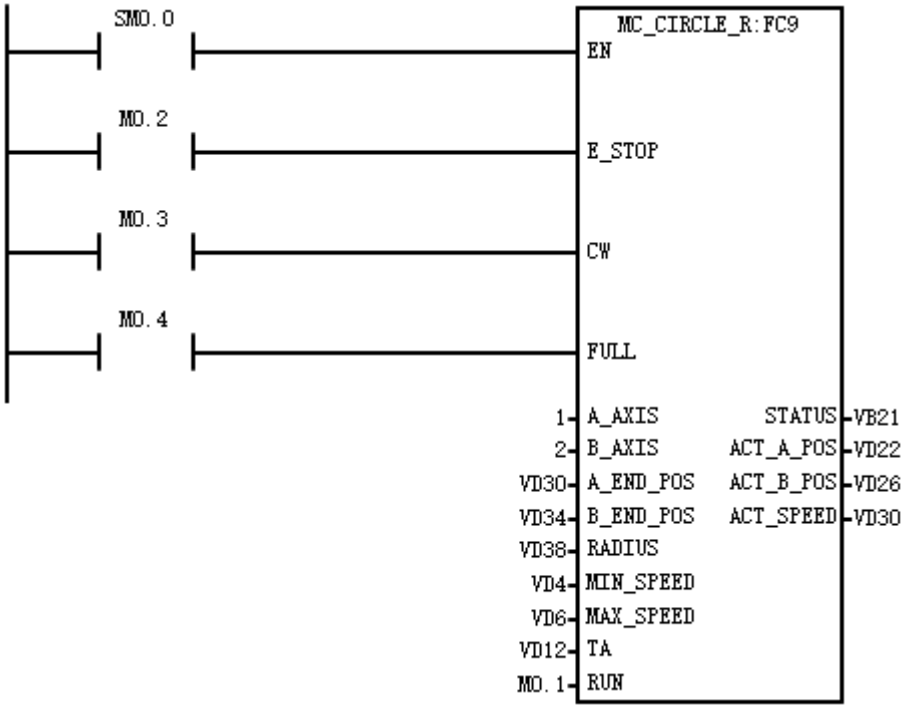
③ 参数说明

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP =0时才能运行。 2、当 E_STOP为 1 时，RUN内部复位。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到相应轴上。	BYTE	0~3	该参数在运行过程中不可修改。
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到相应的轴上。	BYTE	0~3	
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	

TA	IN	加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算。
A_POS	IN	虚拟 A 轴的终点（绝对）坐标	DINT	-2147483648 ~ +2147483647	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算方法详见提示3。（绝对指令必须以脉冲增量来计算）。
B_POS	IN	虚拟 B 轴的终点（绝对）坐标	DINT	-2147483648 ~ +2147483647	
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与E_STOP =0时才能运行。 2、当运行完成后，RUN内部复位。 3、当 E_STOP为 1 时，RUN内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，该因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行没执行或者指令正在执行中但没完成 Bit3：忙标志 1：忙标志有效，直线插补模块或相应的轴被其指令占用 0：忙标志无效，指令正在执行或此执行完成 Bit4~Bit7：预留	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 • MIN_SPEED /MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
ACT_A_POS	OUT	A 轴的当前位置（绝对坐标），如果 A 轴配给0轴，那么该值就表示0轴的绝对坐标	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置（绝对坐标），如果 B 轴配给1轴，那么该值就表示1轴的绝对坐标	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_CIRCLE_R（两轴圆弧插补相对运动指令）

① 函数名：MC_CIRCLE_R



② 功能：可在任意两轴之间进行圆弧插补(设置点为相对坐标)。

③ 参数

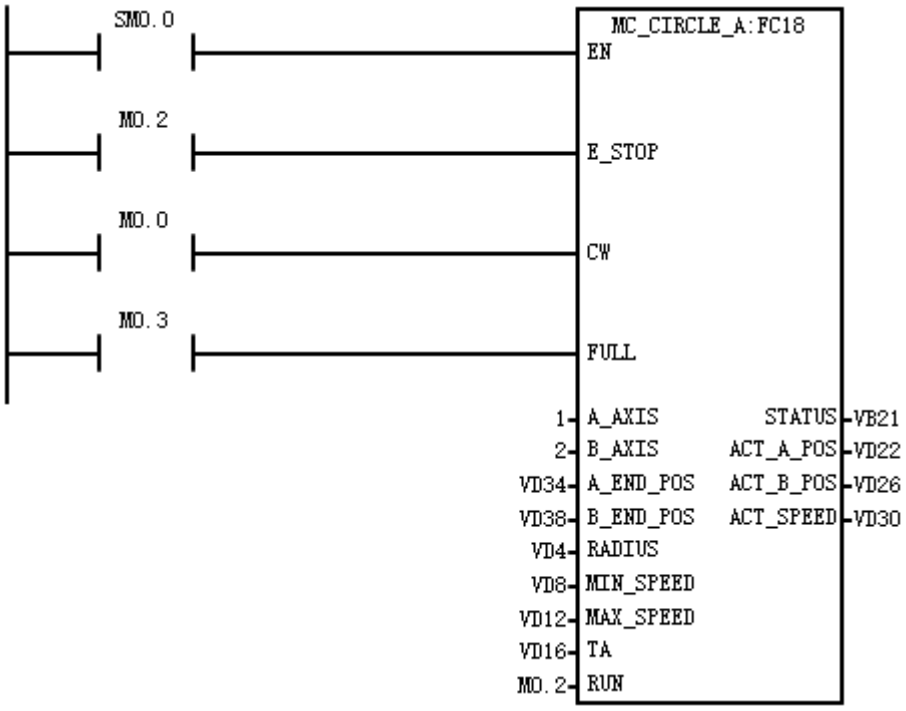
参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1与E_STOP=0时才能运行。 2、当 E_STOP为 1时，RUN内部复位。
CW	IN	顺时针或逆时针插补标志。 1：顺时针 0：逆时针	BOOL	0/1	该参数在运行过程中不能修改。
FULL	IN	全圆标志。 1：全圆 0：圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴，即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴，即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
A_END_POS	IN	如果 FULL 为 0，表示虚拟 A 轴的终点（相对起点）坐标； 如果FULL为1，那么该点只表示圆弧上不同于起点的另一点的（相对起点）坐标，并非终点坐标。	DINT	-2147483648 ~ +2147483647	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算见提示 2
B_END_POS	IN	如果 FULL 为 0，表示虚拟	DINT	-2147483648	

		B 轴的终点（相对起点）坐标； 如果FULL为1，那么该点只表示圆弧上不同于起点的另一点的（相对起点）坐标，并非终点坐标。		~ +2147483647	
RADIUS	IN	圆弧的半径。 1、分正、负。正数：表示走弧度小于 180 的圆弧轨迹。负数：表示走弧度大于 180 度的圆弧轨迹。 2、RADIUS 的绝对值表示圆弧的半径大小。	DINT	|R|<3x10^6	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算见见提示2
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、由于处理器的局限，当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、建议 MIN_SPEED 不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	
TA		加速/减速时间。 单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1与E_STOP=0时才能运行。 2、当运行完成后，RUN内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，因公共资源被其他指令占用，所以指令还未运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行未执行或者指令正在执行但没完成。 Bit3：忙标志 1—忙标志有效，圆弧插补模	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 ● MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 ● 若TA=0若没有设置最大加速度，则报参数故障；TD亦然。

		块或相应的轴被其他指令占用。 0—忙标志无效，指令正在执行或此指令执行完成。 Bit4~Bit7：预留			
ACT_A_POS	OUT	A 轴的当前位置（相对坐标，本次调用实际输出脉冲数），如果 A轴配给 0 轴，那么该值就表示 0轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置（相对坐标，本次调用实际输出脉冲数），如果B轴配给 1 轴，那么该值就表示 1轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_CIRCLE_A（两轴圆弧插补绝对运动指令）

① 函数名：MC_CIRCLE_A



② 功能：可在任意两轴之间进行圆弧插补（设置点为绝对坐标）。

③ 参数说明

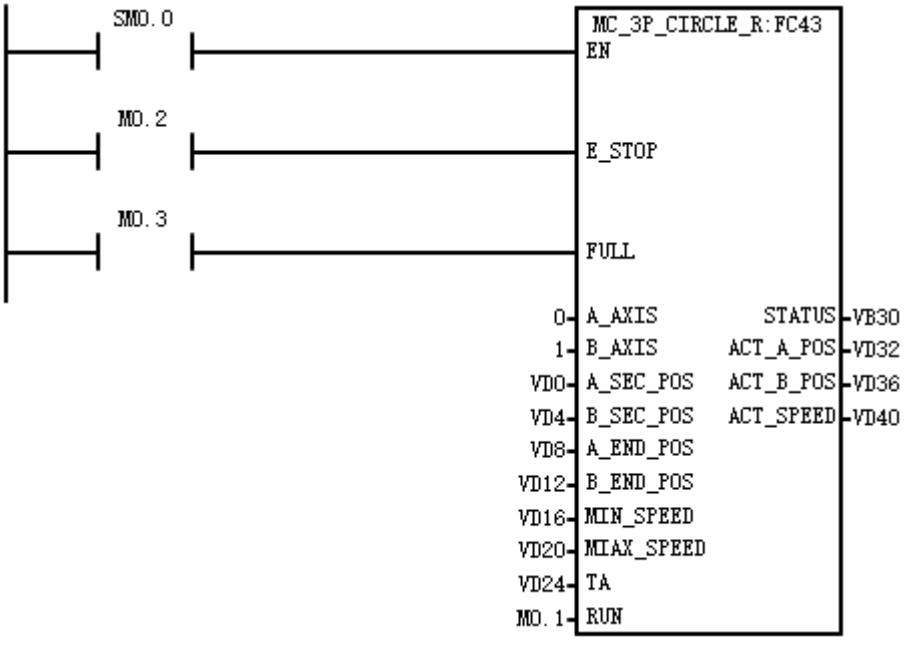
参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1与E_STOP =0时才能运行。 2、E_STOP为1时，RUN内部复位。
CW	IN	顺时针或逆时针插补标志。 1：顺时针 0：逆时针	BOOL	0/1	该参数在运行过程中不能修改。
FULL	IN	全圆标志。 1：全圆 0：圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补 A 轴的轴号。 插补需要两个轴，即虚拟的	BYTE	0~3	该参数在运行过程中不能修改。

		A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定A轴映射到那一轴上。			
B_AXIS	IN	插补 B 轴的轴号。 插补需要两个轴，即虚拟的 A 轴和 B 轴。我们需要映射到实际输出的 0、1、2、3 轴上。该参数即可设定B轴映射到那一轴上。	BYTE	0~3	
A_END_POS	IN	如果FULL为0，表示虚拟 A 轴的终点（绝对）坐标； 如果FULL为1，那么该点只表示圆弧上不同于起点的另一点的（绝对）坐标，并非终点坐标。	DINT	-2147483648 ~ +2147483647	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算方法详见见提示 2。（绝对指令必须以脉冲增量来计算）
B_END_POS	IN	如果 FULL为0，表示虚拟 B 轴的终点（绝对）坐标； 如果FULL为1，那么该点只表示圆弧上不同于起点的另一点的（绝对）坐标，并非终点坐标。	DINT	-2147483648 ~ +2147483647	
RADIUS	IN	圆弧的半径。 1、分正、负。正数：表示走弧度小于180 的圆弧轨迹。负数：表示走弧度大于 180 度的圆弧轨迹。 2、RADIUS 的绝对值表示圆弧的半径大小。	DINT	|R| <3 x 10^6	1、该参数在运行过程中不能修改。 2、单位：脉冲，与实际位移转换计算方法详见见提示2。
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、MIN_SPEED 建议不要小于 500，否则在脉冲输出完成减速结束时最小速度受限（通常限制为 500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	
TA		加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算，计算方法详见提示3。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与E_STOP=0时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1 时，RUN 内部复位。

STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，因公共资源被其他指令占用，所以指令还未运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行未执行或者指令正在执行但没完成。 Bit3：忙标志 1—忙标志有效，圆弧插补模块或相应的轴被其他指令占用。 0—忙标志无效，指令正在执行或此指令执行完成。 Bit4~Bit7：预留	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 2、MIN_SPEED / MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
ACT_A_POS	OUT	A 轴的当前位置（绝对坐标），如果 A轴配给 0 轴，那么该值就表示 0轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置（绝对坐标），如果 B轴配给 1 轴，那么该值就表示 1轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_3P_CIRCLE_R（两轴三点画弧插补相对运动指令）

① 函数名：MC_3P_CIRCLE_R



② 功能：可在任意两轴之间进行圆弧插补（设置点为相对坐标）。

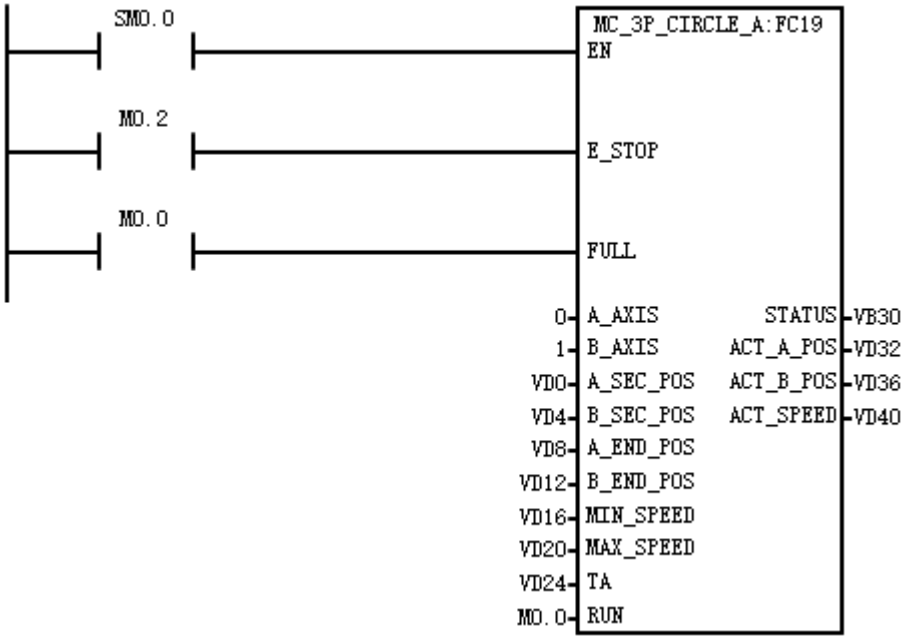
③ 参数说明

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有RUN=1与E_STOP=0时才能运行。 2、E_STOP为1时，RUN内部复位。
FULL	IN	全圆标志。 1：全圆 0：圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补A轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的0、1、2、3轴上。该参数即可设定A轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
B_AXIS	IN	插补B轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的0、1、2、3轴上。该参数即可设定B轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
A_SEC_POS	IN	决定圆弧的第二点的A轴坐标(此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第2点)（此为相对坐标）	DINT	-2147483648 ~ +2147483647	1、插补的起点坐标即为该圆弧的第1点。（相对指令起点坐标为（0，0））。 2、该参数在运行过程中不能修改。 ● 单位：脉冲，与实际位移转换计算方法见提示2。 ● 插补的起点坐标默认为（0,0）。
B_SEC_POS	IN	决定圆弧的第二点的B轴坐标(此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第2点)（此为相对坐标）	DINT	-2147483648 ~ +2147483647	
A_END_POS	IN	决定圆弧的第三点的A轴坐标，若FULL为0，即也是终点坐标(此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第3点)（此为相对坐标）	DINT	-2147483648 ~ +2147483647	
B_END_POS	IN	决定圆弧的第三点的B轴坐标，若FULL为0，即也是终点坐标(此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第3点)（此为相对坐标）	DINT	-2147483648 ~ +2147483647	
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、MIN_SPEED 建议不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。 4、将 MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使
MAX_SPEE	IN	长轴最大速度，即运行中	DWO	0~200000	

D		的最大速度。单位：Hz	RD		能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
TA		加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可以修改。 2、加速度只在启动时和TA/TD变化时计算，计算方法详见提示3。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有 RUN =1 与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、E_STOP为1时，RUN内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，因公共资源被其他指令占用，所以指令还未运行；或指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行未执行或者指令正在执行但没完成。 Bit3：忙标志 1—标志有效，圆弧插补模块或相应轴被其他指令占用。 0—标志无效，指令正在执行或此指令执行完成。 Bit4~Bit7：预留	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 2、MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。
ACT_A_POS	OUT	A轴的当前位置（相对坐标，本次调用实际输出脉冲数），如果 A轴配给0轴，那么该值就表示0轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B轴的当前位置（相对坐标，本次调用实际输出脉冲数），如果B轴配给1轴，那么该值就表示1轴的相对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_3P_CIRCLE_A（两轴三点画弧插补绝对运动指令）

① 函数名：MC_3P_CIRCLE_A



② 功能：可在任意两轴之间进行圆弧插补（设置点为绝对坐标）。

③ 参数说明

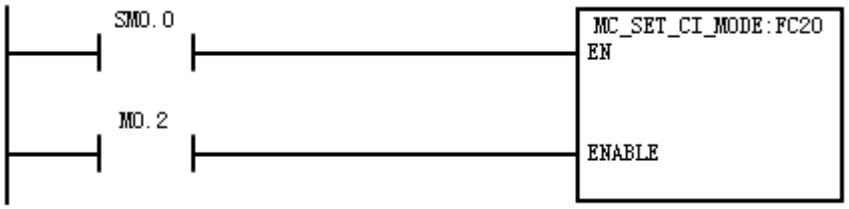
参数名	输入输出属性	参数描述	数据类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有RUN=1与E_STOP=0时才能运行。 2、当E_STOP为1时，RUN内部复位。
FULL	IN	全圆标志。 1：全圆 0：圆弧	BOOL	0/1	该参数在运行过程中不能修改。
A_AXIS	IN	插补A轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的0、1、2、3轴上。该参数即可设定A轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
B_AXIS	IN	插补B轴的轴号。 插补需要两个轴，即虚拟的A轴和B轴。我们需要映射到实际输出的0、1、2、3轴上。该参数即可设定B轴映射到那一轴上。	BYTE	0~3	该参数在运行过程中不能修改。
A_SEC_POS	IN	决定圆弧的第二点的A轴坐标(此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第2点)(此为绝对坐标)	DINT	-2147483648 ~ +2147483647	1、插补的起点坐标即为该圆弧的第1点（绝对指令起点坐标为即为当前点的绝对坐标）。 2、该参数在运行过程中不能修改。 3、单位：脉冲，与实际位移转换计算方法见提示2。（绝对指令必须以脉冲增量来计算）。
B_SEC_POS	IN	决定圆弧的第二点的B轴坐标(此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第2点)(此为绝对坐标)	DINT	-2147483648 ~ +2147483647	
A_END_POS	IN	决定圆弧的第三点的A轴坐标，若FULL为0，即也是终点坐标(此指令用于三点确定一圆弧或圆时，此	DINT	-2147483648 ~ +2147483647	

		参数为运行方向上的第3点)（此为绝对坐标）		7	
B_END_POS	IN	决定圆弧的第三点的B轴坐标，若FULL为0，即也是终点坐标（此指令用于三点确定一圆弧或圆时，此参数为运行方向上的第3点）（此为绝对坐标）	DINT	-2147483648 ~ +2147483647	
MIN_SPEED	IN	长轴最小速度，即启动时或停止时的速度。单位：Hz	DWORD	0~200000	1、当速度小于5Hz时，脉冲输出关闭，即无输出。 2、该参数在运行过程中可以修改。 3、MIN_SPEED 建议不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）。 4、MAX_SPEED 写 0，可实现软停功能（即输出脉冲减速停止），此时使能位 RUN 不复位；若将速度写回大值，可继续输出脉冲直至输出完成。
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。单位：Hz	DWORD	0~200000	
TA		加速/减速时间。单位：ms	DWORD	0~10000	1、该参数在运行过程中可修改。 2、加速度只在启动时和TA/TD变化时计算，计算方法详见提示3
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	1、只有RUN =1 与E_STOP=0时才能运行。 2、运行完成后，RUN内部复位。 3、当E_STOP为1时，RUN内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，并且还没执行完。 0—不运行，因公共资源被其他指令占用，所以指令还未运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，执行未执行或者指令正在执行但没完成。 Bit3：忙标志 1—忙标志有效，圆弧插补模块或相应的轴被其他指令占用。	BYTE	0~255	Bit0： 1、只对轴参数配置错误进行判断。 2、MIN_SPEED /MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、若TA=0若没有设置最大加速度，则报参数故障；TD亦然。

		0—忙标志无效，指令正在执行或此指令执行完成。 Bit4~Bit7：预留			
ACT_A_POS	OUT	A 轴的当前位置（绝对坐标），如果A轴配给0轴，那么该值就表示0轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_B_POS	OUT	B 轴的当前位置（绝对坐标），如果 B轴配给1轴，那么该值就表示1轴的绝对坐标。	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前的实际速度	DWORD	0~200000	

MC_SET_CI_MODE（设置连续插补指令）

① 函数名：MC_SET_CI_MODE



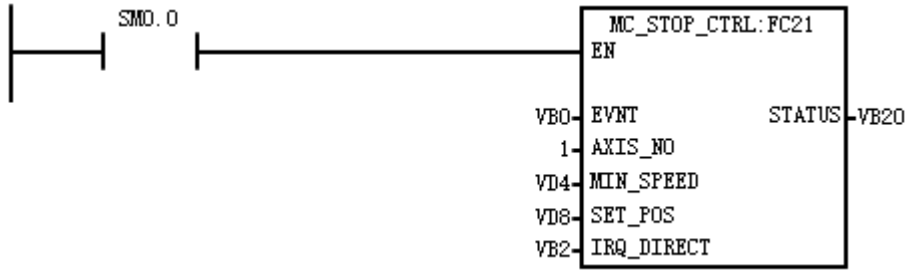
② 功能：设置是否使能连续插补功能。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
ENABLE	IN	0：关闭连续插补功能 1：使能连续插补功能	BOOL	0~1	建议连续插补运行过程中不要修改。

MC_STOP_CTRL（控制停止指令）

① 函数名：MC_STOP_CTRL



② 功能：通过一个中断事件，触发一个轴停止，使得正在运行的轴按设定参数执行停止过程。支持位置指令和速度指令，任何插补指令均不支持这种停止方式。仅支持 IO 中断号，不支持通信类中断号。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
EVNT	IN	中断事件号	BOOL	M228IL：0~47 226H：0~33	除掉中间的通信中断、定时中断以及PTO中断，详情参见中断事件表
AXIS_NO	IN	设置轴号	BYTE	M228IL：0~5 226H ：0~3	
MIN_SPEE	IN	最小速度，启动或停止时的速度	BYTE	0～200000	单位：HZ，建议最小速度不要小于 500，否则在脉冲输出将完成减速结束时最小速度受限（通常限制为 500）

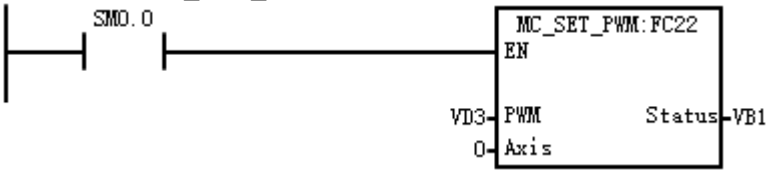
SET_POS	IN	设定输出脉冲数（相对坐标	DWORD	0～+2147483647	
IRQ_DIRECT	IN	该参数仅仅对中断号为27，14，17，30起作用，当中断号设置为以上四个值中的一个时可以给IRQ_DIRECT配置为0，表示高速计数器从增计数变为减计数，然后开始执行电机缓停，配置为1表示为电机从减计数变为增计数，然后执行电机缓停，如果设置的非1则默认使用下降沿中断。	DWORD		
STATUS	OUT	脉冲输出状态	BYTE		具体见下表

顺序：bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0

Bit7	保留
Bit6	保留
Bit5	保留
Bit4	保留
Bit3	1：缓停指令个数超限； 0：缓停指令个数正常。
Bit2	1：执行完缓停； 0：没有执行缓停或者正在缓停中。
Bit1	1：电机正在缓停； 0：执行完缓停或者是没有启动缓停；
Bit0	轴号状态位。1：轴号设置错误；0：轴号设置正确

MC_SET_PWM（脉冲宽度调制指令）

① 函数名：MC_SET_PWM



② 功能：该指令用于调节运控输出脉冲的占空比，只有 V5 平台 PLC 硬件支持该功能，固件版本 V2.73 以上支持该功能。该指令只用于 PLC 226EH。
该指令执行立即生效，如果此时脉冲正在输出，则占空比在调用指令之后立即修改。

③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
PWM	IN	输出占空比	REAL	0.0~1.0	范围0.0至1.0，对应0%至100%占空比。
Axis	IN	轴号，需要调节占空比的轴号范围跟PLC型号最大支持的轴数相同。	BYTE		
Status	OUT	状态字，0-初始状态，1-完成，2-占空比超范围，3-轴号错误	BYTE		

提示

提示 ❶ 当某轴调用运动控制指令后，那么该轴只有在重新上电复位或 CPU 停机时，才可以恢复普通 IO 的功能。

提示 ❷ 直线插补指令（A_POS、B_POS），圆弧插补指令中（A_END_POS、B_END_POS、RADIUS）计算方法说明如下：

符号	说明	单位
POS	插补指令中需填入的脉冲个数	

L	坐标上实际点的绝对值	mm
S	运动轴上丝杆导程	mm
M	步进驱动器的细分或伺服驱动的分辨率（即电机转动一圈需要的脉冲个数）	
计算公式：POS = L*M÷S (单位：脉冲数)		

提示 ❸ 当 TA≠0，加速度=（MAX_SPEED-MIN_SPEED）/TA（若设置有最大加速度，则受限于最大加速度）；若 TA=0，则采用指令 MC_SET_MAX_ACCELE 设置的最大加速度，若没有设置最大加速度，则报参数故障；TD 亦然。对于双轴指令，若两轴均设置了最大加速度，则采用其中的较小值作为系统加速度。

4.1.5 运控功能应用示例

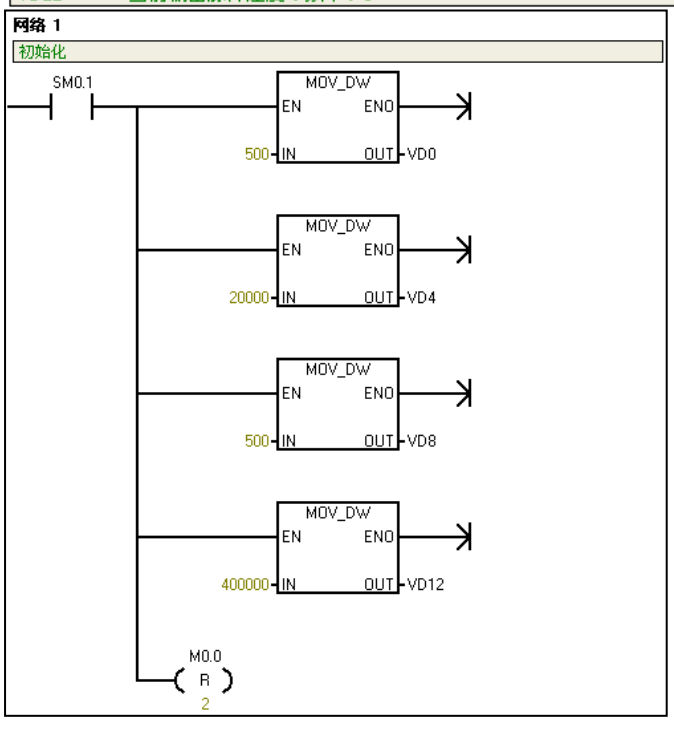
CTMC 运动控制器集成强大的运动控制功能，可实现多种运动控制方式，本节将结合具体实例对这些运动控制功能加以介绍。

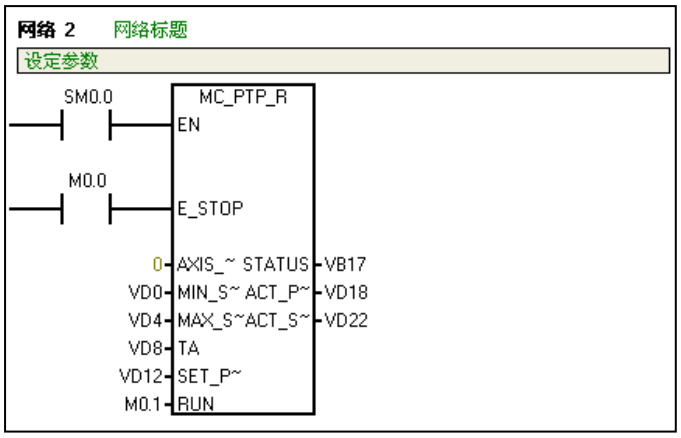
单轴相对运动控制

位置控制是指控制某个物体完成从当前位置运动到另一个位置的指定位移，一般称为点对点运动或定长运动。设置最大/最小速度、目标位置等参数后执行位置控制指令，当前位移等于设定位移时，CPU 将减速停止。单轴相对运动指令详细说明参见 MC_PTP_R（单轴相对运动指令）。

应用程序示例

```
程序注释
功能：用作单轴点对点控制（单轴定长驱动）。
调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

M0.0 ------------紧急停止位；
轴号为0-------Q0.0脉冲输出、Q0.1方向输出；
VD0 -------------启动/停止速度
VD4 -------------加速完成后的正常速度；
VD8 -------------加速时间（ms）；
VD12------------要输出的脉冲数；
VB17------------输出状态字节；
VD18------------输出脉冲个数；
VD22------------当前输出脉冲速度（频率）。
```



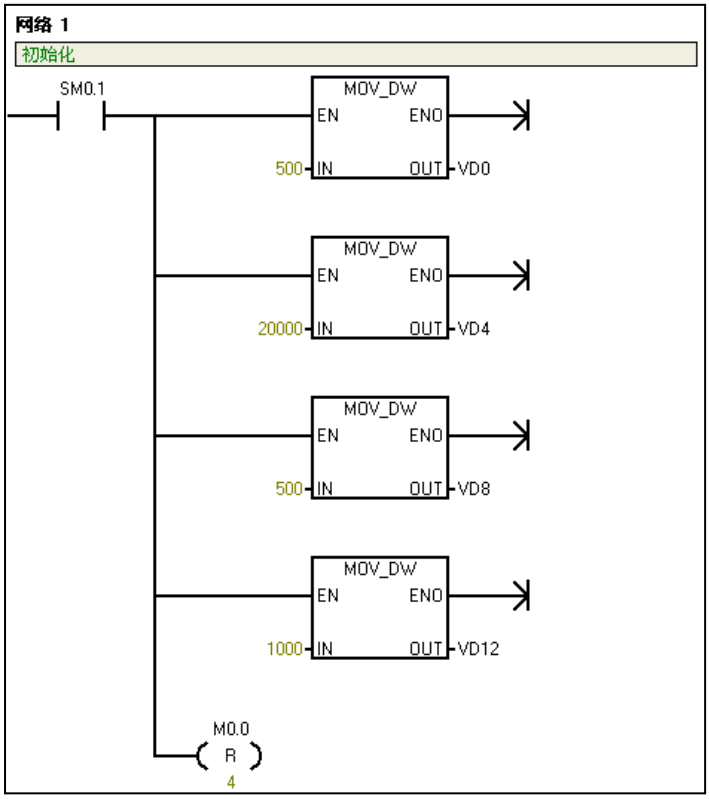


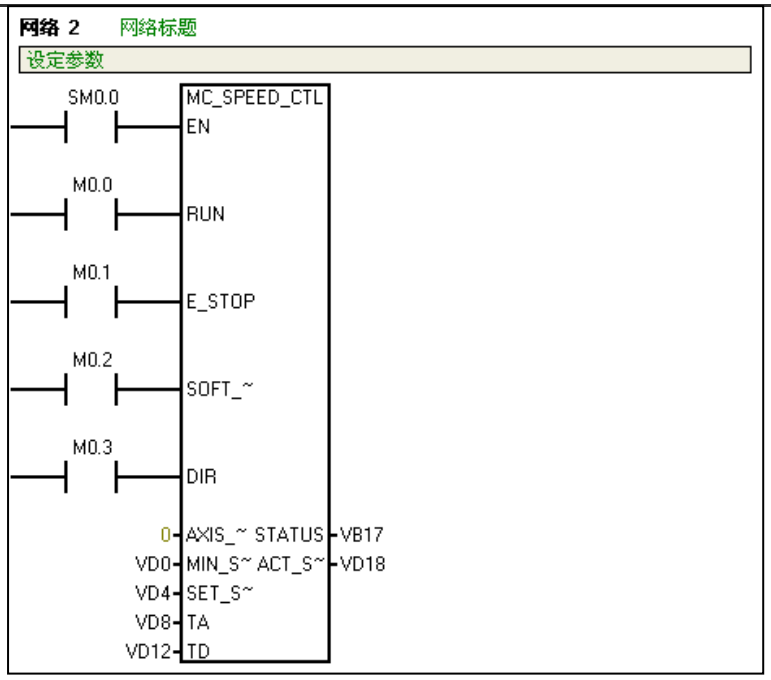
单轴速度运动控制

速度控制是指电机从起始速度开始运行，加速至指定速度连续运动。只有当接收到停止命令后，才减速直至停止（也可设为立即停止）。该模式下不控制运动距离，它的主要用途是：寻找机械原点、速度控制。速度控制运动指令详细说明参见 MC_SPEED_CTL（速度控制指令）。

应用程序示例：

```
程序注释
功能：控制单轴输出脉冲的频率，可任意时候改变输出脉冲的频率（速度）。
当接收到软停止命令时，会自动减速停止。当收到紧急停止命令时，会马上停止脉冲输出，不经过减速。

M0.0 ---------- 运行使能位；
M0.1 ---------- 紧急停止位；
M0.2 ---------- 软停止位；
M0.3 ---------- 脉冲方向位（0为反方向，1为正方向）；
轴号为0------Q0.0脉冲输出、Q0.1方向输出；
VD0 ----------启动/停止速度
VD4 ---------- 加速完成后的正常速度；
VD8 ---------- 加速时间（ms）；
VD12---------- 减速时间（ms）；
VB17---------- 输出状态字节；
VD18---------- 当前输出脉冲速度（频率）。
```



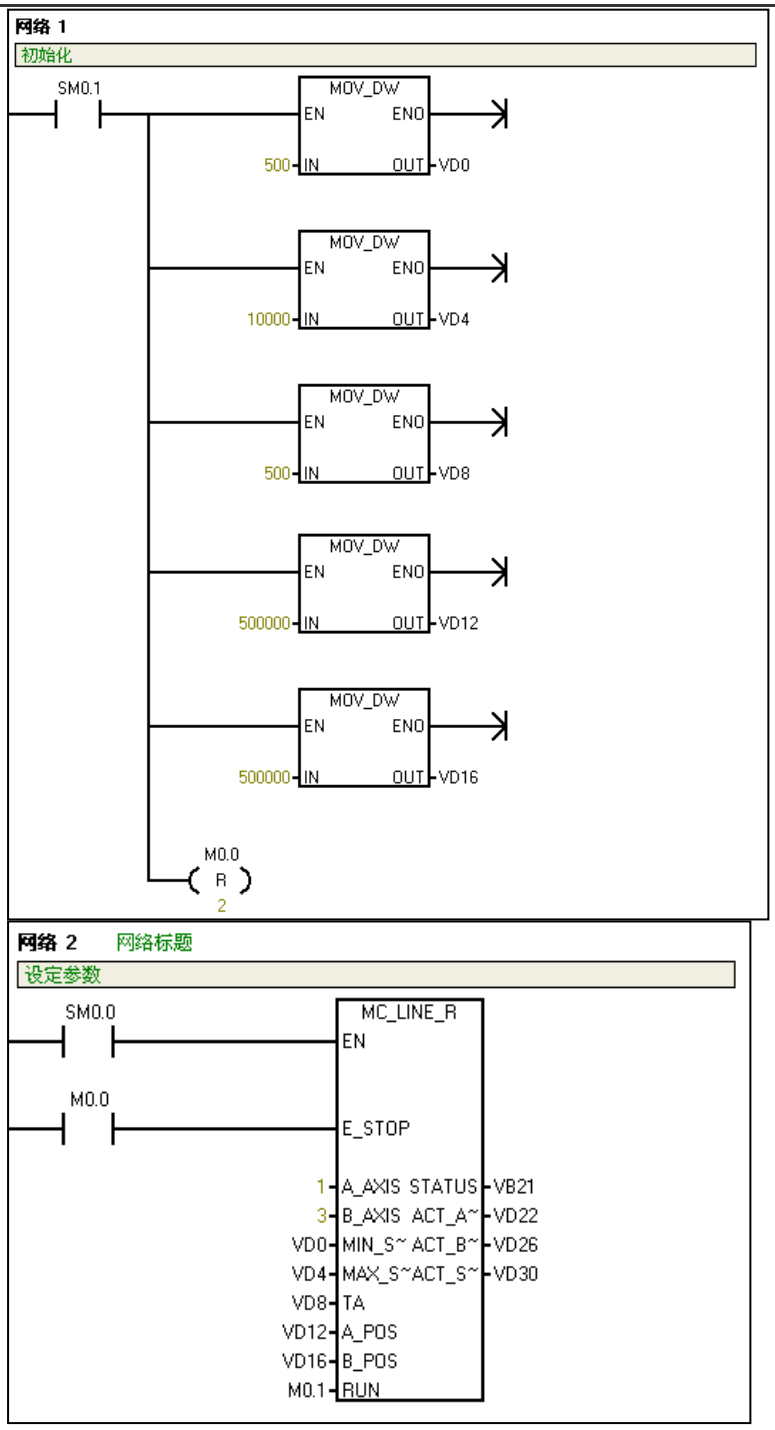


两轴直线插补相对运动

位置控制中，以直线为轨迹对实现群组化的 X 轴、Y 轴这 2 轴、或者包括 Z 轴在内的 3 轴的电机动作进行控制的插补控制。指定方法包括合成速度指定、长轴速度指定两种。两轴直线插补相对运动指令参见 MC_LINE_R（两轴直线插补相对运动指令）

应用程序示例

```
程序注释
功能：可在任意两轴之间、平面上任意区域内进行直线插补功能。

M0.0 ----------紧急停止位；
轴号为1------Q0.2脉冲输出、Q0.3方向输出；
轴号为3------Q0.6脉冲输出、Q0.7方向输出；
VD0 ----------长轴启动/停止速度
VD4 ----------长轴加速完成后的正常速度；
VD8 ----------加减速时间；
VD12----------虚拟A轴的终点(相对)坐标；
VD16----------虚拟B轴的终点(相对)坐标；
M0.1 ----------运行使能位；
VB21----------输出状态字节；
VD22----------A轴的当前位置（相对坐标，本次调用实际输出脉冲数）；
VD26----------B轴的当前位置（相对坐标，本次调用实际输出脉冲数）；
VD30 ----------当前的实际速度（频率）。
```



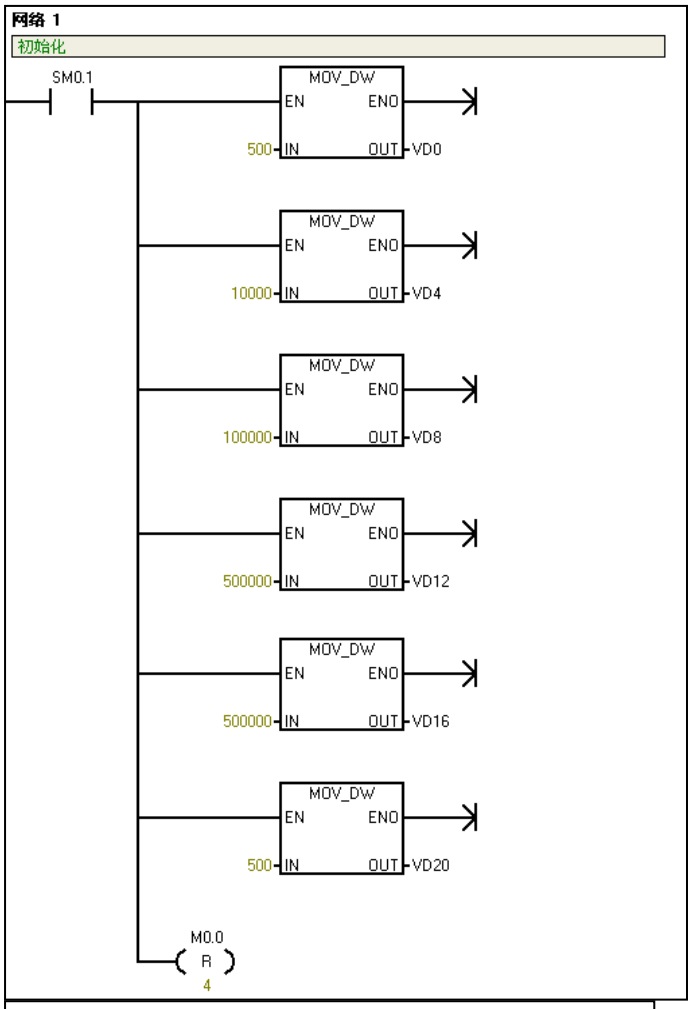
两轴圆弧插补相对运动

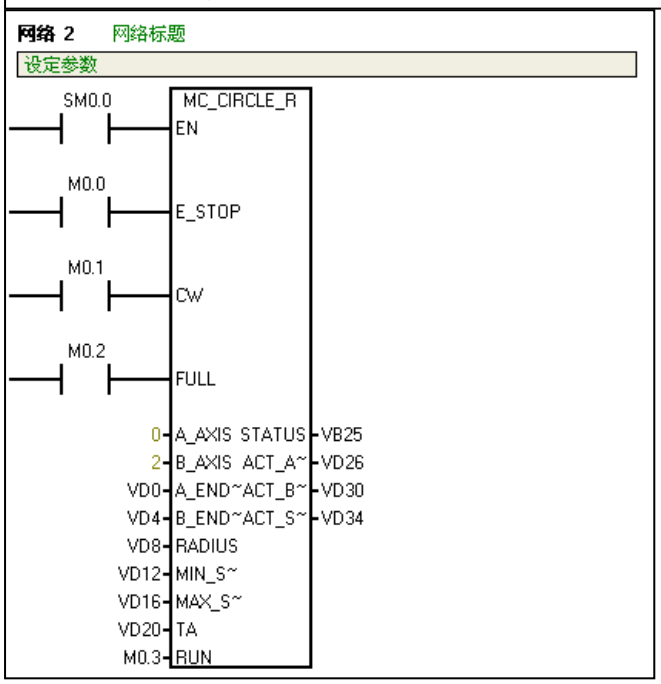
圆弧插补从当前位置开始，根据所指定的圆心、终点位置、圆弧半径（正数或负数）及插补的方向（按顺时针或逆时针）来进行。

位置控制中，以圆弧为轨迹对实现群组化的 X 轴、Y 轴这 2 轴的电机动作进行控制的插补控制。圆弧的指定方法包括中心点指定、通过点指定两种。两轴圆弧插补相对运动指令详细介绍参见 MC_CIRCLE_R（两轴圆弧插补相对运动指令）。

应用程序示例

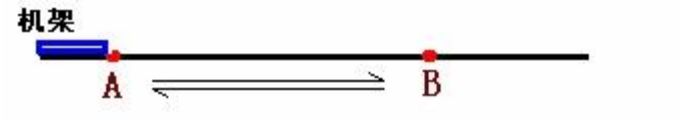
```
程序注释
功能：可在任意两轴之间、平面上任意区域内进行圆弧插补功能。

M0.0 ----------紧急停止位；
M0.1 ----------顺时针或逆时针插补标志位（1、顺时针、0、逆时针）；
M0.2 ---------- 全圆标志位（1、全圆、 0- 圆弧）
轴号为0-------Q0.0脉冲输出、Q0.1方向输出；
轴号为2-------Q0.4脉冲输出、Q0.5方向输出；
VD0 ----------长轴启动/停止速度
VD4 ---------- 长轴加速完成后的正常速度；
VD8 ---------- 圆弧的半径；
VD12---------- 虚拟A轴的终点相对坐标；
VD16---------- 虚拟B轴的终点相对坐标；
VD20---------- 加减速时间；
M0.3 ----------运行使能位；
VB25----------输出状态字节；
VD26---------- A轴的当前位置（相对坐标，本次调用实际输出脉冲数）；
VD30----------B轴的当前位置（相对坐标，本次调用实际输出脉冲数）；
VD34 ----------当前的实际速度（频率） 。
```





混合运动

控制步进电机从 A 点到 B 点往返运动，步进电机细分 1000，丝杆导程 5mm，A 到 B 的位移 L 为 2000mm。

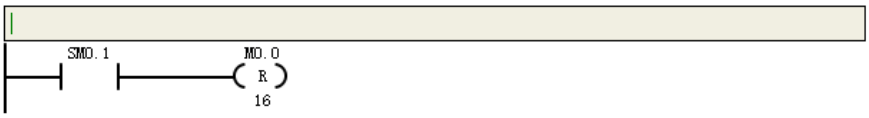


系统说明：

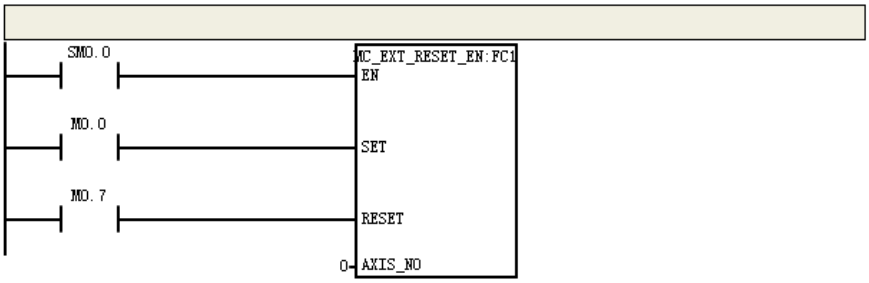
本例设置 CPU M228IL 第 0 轴做点到点运动的参数。主要调用 MC_PTP_R 来设定控制参数。

I0.2 为 A 点硬件归零复位点（此点为行程开关量输入，设此点为机械原点）；I2.0 为系统急停输入；Q0.0 为脉冲输出，Q0.1 方向输出。

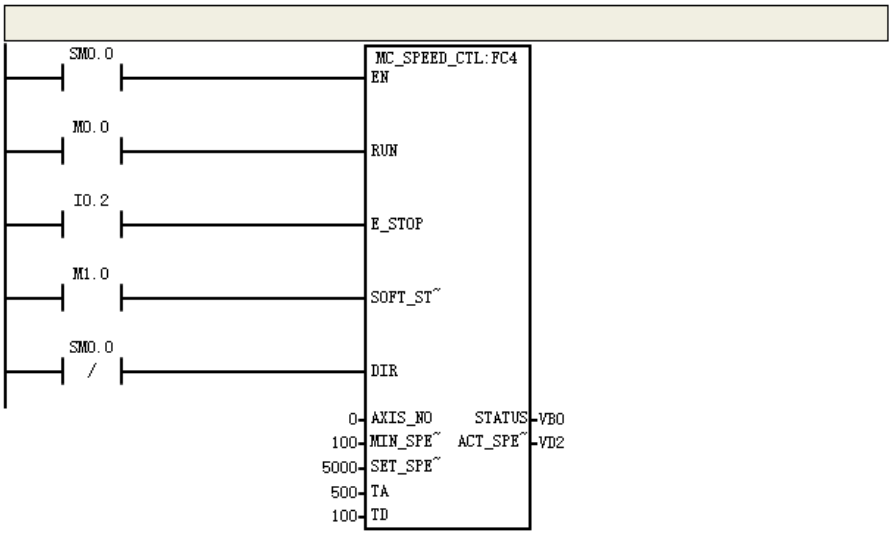
网络 1



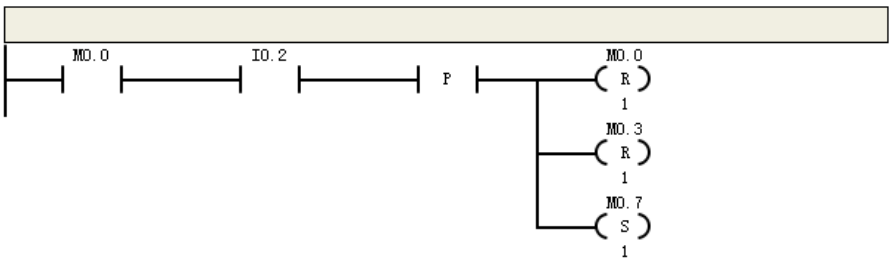
网络 2



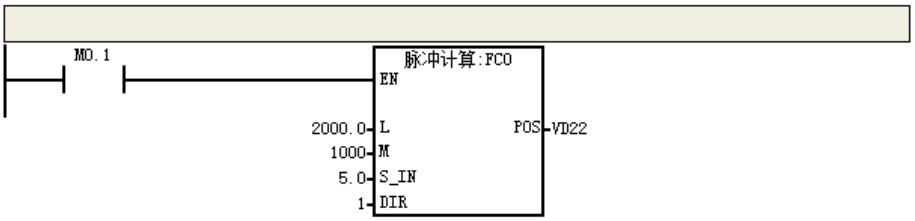
网络 3



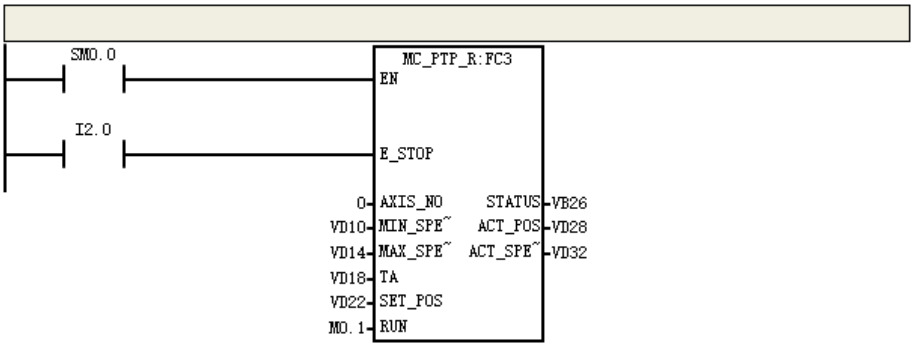
网络 4



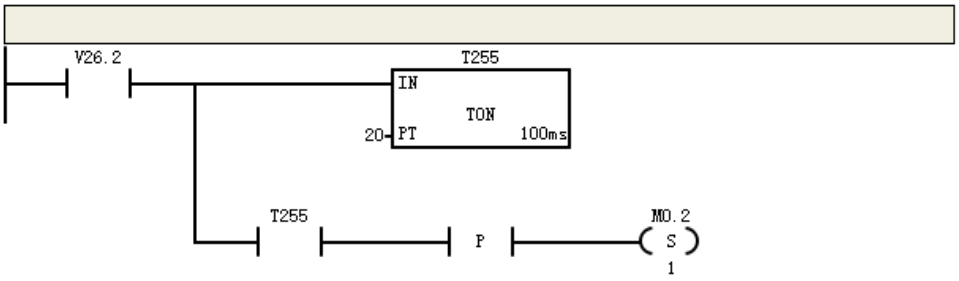
网络 5


M0.1
脉冲计算:FC0
EN
2000.0
L
POS
VD22
1000
M
5.0
S_IN
1
DIR

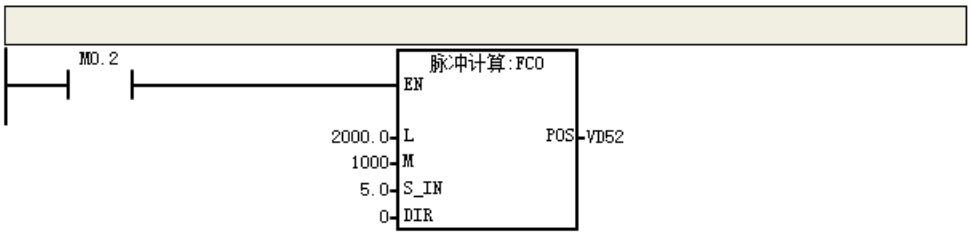
网络 6


SM0.0
MC_PTP_R:FC3
EN
I2.0
E_STOP
0
AXIS_NO
STATUS
VB26
VD10
MIN_SPE~
ACT_POS
VD28
VD14
MAX_SPE~
ACT_SPE~
VD32
VD18
TA
VD22
SET_POS
M0.1
RUN

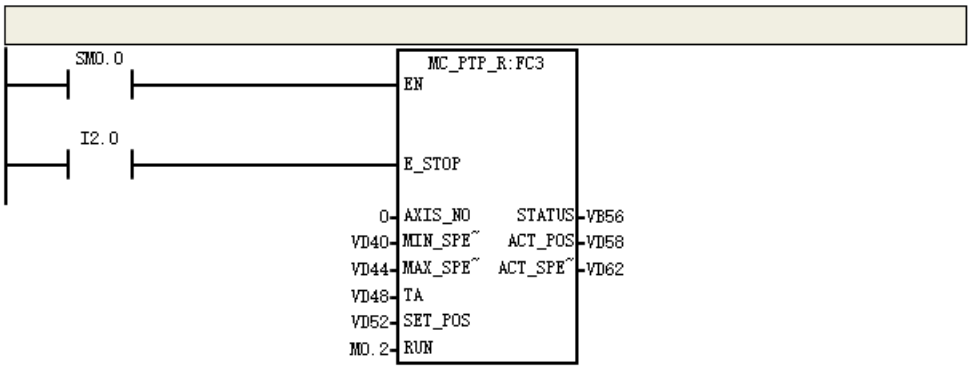
网络 7


V26.2
T255
IN
TON
20
PT
100ms
T255
P
M0.2
S
1

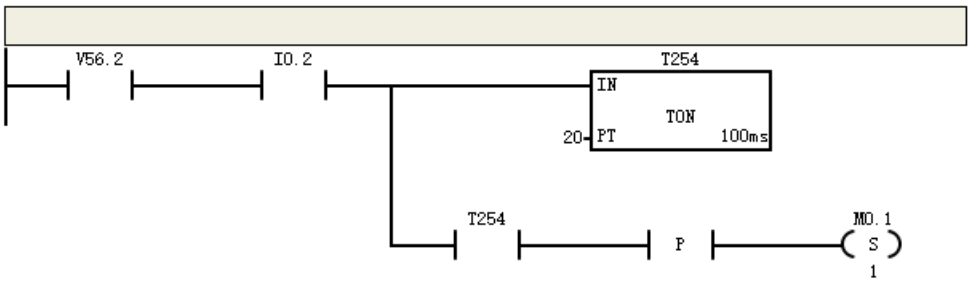
网络 8



网络 9



网络 10



注：应用程序也可于合信公司网站下载。网址：http://www.co-trust.com/Download/index.html

回原功能

CTMC 系列 CPU 可使用以下 14 种回原模式，用户可根据对精度的要求及实际应用需求来选择。

回原模式	说明
1	参考负向原点开关和 Z 相信号的原点模式
2	参考正向原点开关和 Z 相信号的原点模式
3	只参考负向原点开关的原点模式
4	只参考正向原点开关的原点模式
5	只参考 Z 相信号的原点模式（负向回原）
6	只参考 Z 相信号的原点模式（正向回原）
7	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）
8	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）
9	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）
10	参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）

11	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）
12	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）
13	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）
14	参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）

回原模式说明

轴号与外部复位 IO 信号的对应关系如下（Z pulse 即为 HSC 复位信号，详见高速计数器说明）：

CPU 型号	轴 0	轴 1	轴 2	轴 3	轴 4	轴 5
CTMC 系列： M228IL	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I2.0（HSC3，SM137.0）	I0.5（HSC4，SM147.0）	I2.4（HSC5，SM157.0）
CTH200 系列： H226IH H224X H226XL H226XM H228XL H226IM H226IL **CTSC 系列：** **224E** 晶体管型 **226M-CAN** 晶体管型 **224I** 晶体管型 **226I** 晶体管型 **CTSC 系列：** **226H**	I0.2（HSC0，SM37.0）	I1.0（HSC1，SM47.0）	I1.4（HSC2，SM57.0）	I0.5（HSC4，SM147.0）	--	--

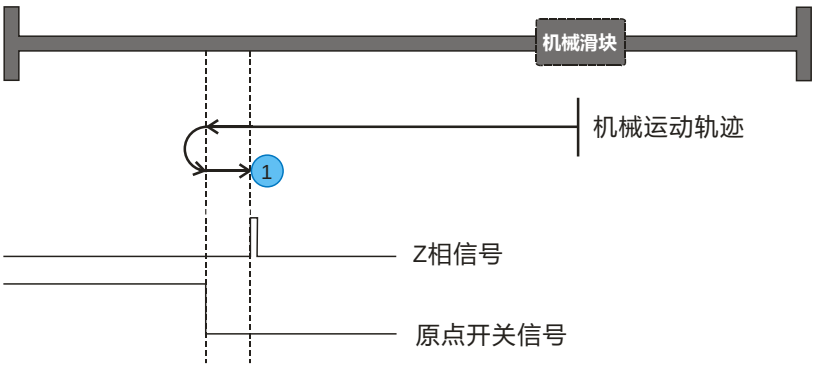
若回原模式以原点开关为参考时（回原模式 3 或 4），必须将原点开关信号（指令的 HOMING_SW 参数）接至 CPU 的上述对应点，否则无法找到原点。

无论机械初始处于什么位置，当设备（原点开关、正向行程限位开关、负向行程限位开关）安装完好，伺服所寻找的设备原点总是唯一的。以下各模式示意图中的竖线“|”代表机械初始位置，圆圈“⊗”代表原点位置。

回原模式 1：DI 触发回原

将外部 DI 引脚配置成“原点开关信号”，触发此信号（上升沿），则复位 P216（用户位置坐标）为原点 0（此模式为特殊回原模式，不输出回原完成信号）。

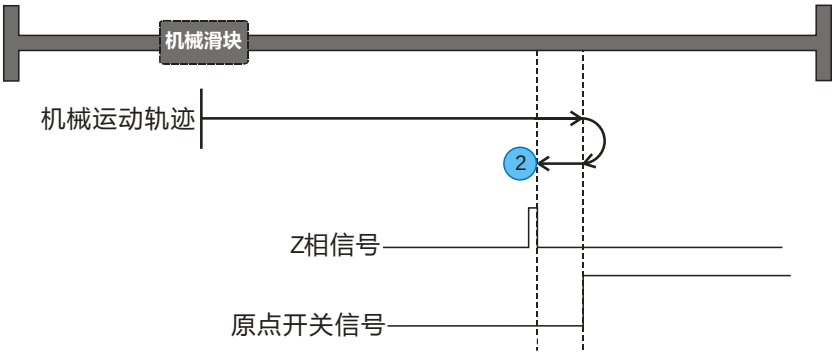
回原模式 1：参考负向原点开关和 Z 相信号的原点模式



原点开关位于机械负方向。机械往原点开关方向运动，在检测到原点开关后减速停止，再反转退出原点

开关，找电机的下一个 Z 相信号并将该位置记录为原点，电机立即停止。

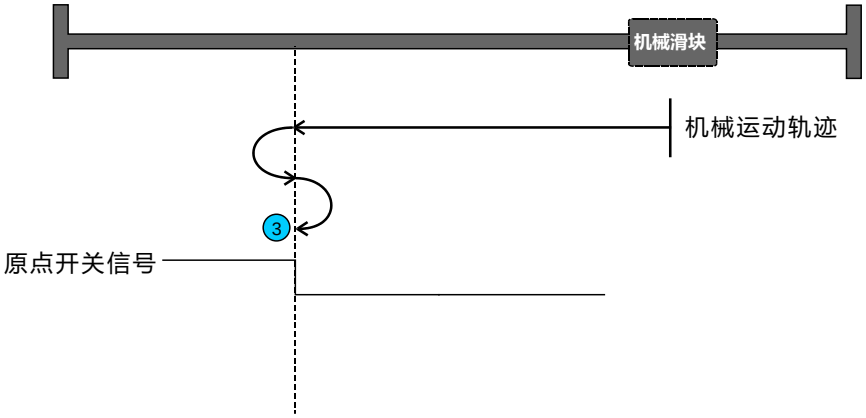
回原模式 2：参考正向原点开关和 Z 相信号的原点模式



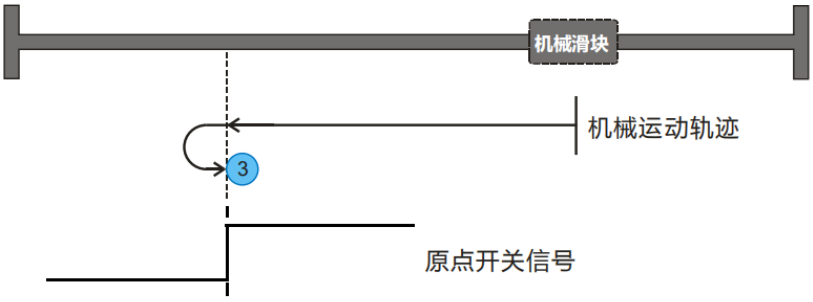
原点开关位于机械正方向。机械往原点开关方向运动，在检测到原点开关后减速停止，再反转退出原点开关，找电机的下一个 Z 相信号并将该位置记录为原点，电机立即停止。

回原模式 3：参考负向原点开关的原点模式（请将原点开关信号接至轴对应的 Z 相信号点）

1）当原点开关信号类型设定为高电平有效时。原点位于机械负方向，机械往原点开关方向运动，在检测到原点开关时，先减速至停止，再反转退出原点开关，在检测到原点开关下降沿时，再次减速停止并换向，从而再次接近原点开关，在开关产生上升沿时停止即找到原点，如下图：

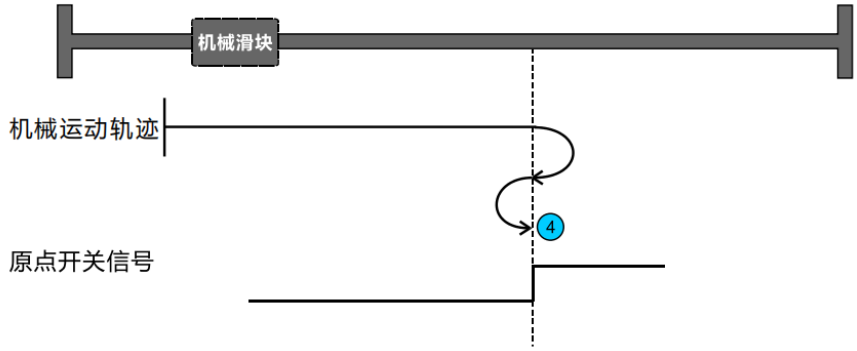


2）当原点开关信号类型设定为低电平有效时。原点开关位于机械负方向，机械往原点开关方向运动，在检测到原点开关时先减速至停止，反转退出原点开关，在退出原点开关产生上升沿时停止即找到原点，如下图：

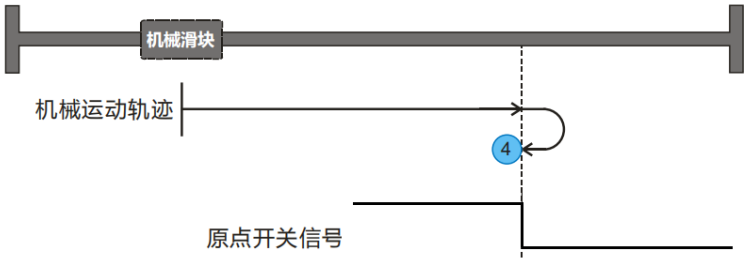


回原模式 4：参考正向原点开关的原点模式（请将原点开关信号接至轴对应的 Z 相信号点）

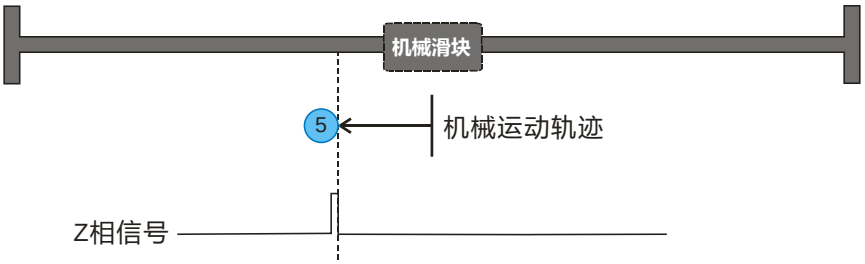
当原点开关信号类型设定为高电平有效时。原点开关位于机械正方向，机械往原点开关方向运动，在检测到原点开关时先减速停止，再反转退出原点开关，在出现原点开关下降沿时，再次减速停止并换向，从而再次接近原点开关，在开关产生上升沿时停止即找到原点，如下图：



当原点开关信号类型设定为低电平有效时。原点开关位于机械正方向，机械往原点开关方向运动，在检测到原点开关时先减速停止，再反转退出原点开关，在退出原点开关产生上升沿时停止即找到原点，如下图：

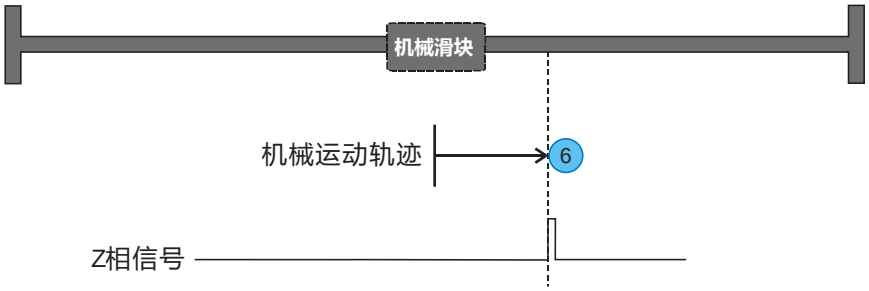


回原模式 5：参考 Z 相信号的原点模式（负向回原）



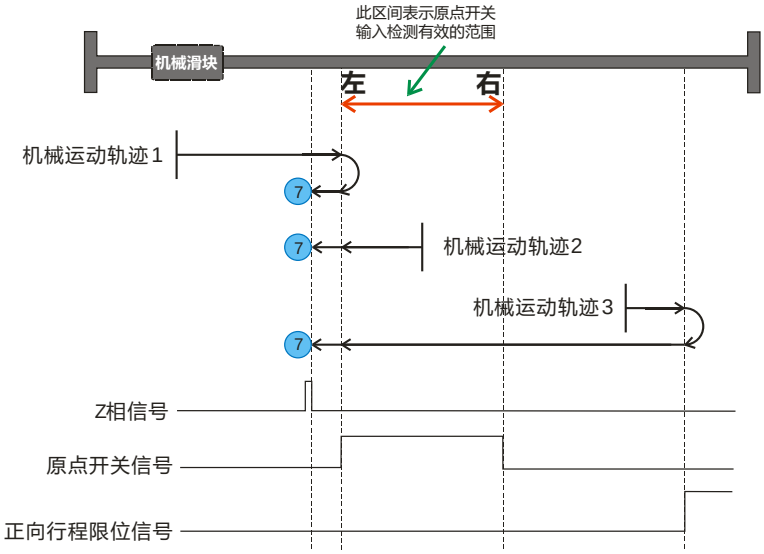
电机从当前位置往负方向运动，找到下一个 Z 相信号时将该位置记录为原点。

回原模式 6：参考 Z 相信号的原点模式（正向回原）



电机从当前位置往正方向运动，找到下一个 Z 相信号时将该位置记录为原点。

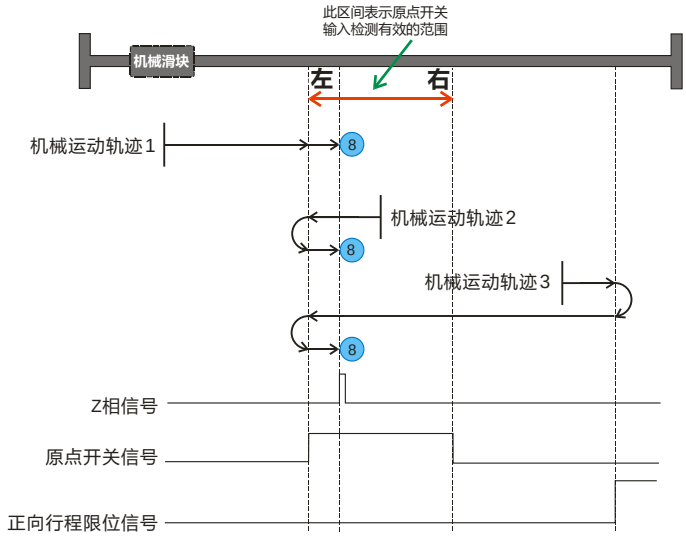
回原模式 7：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号左边沿以左的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

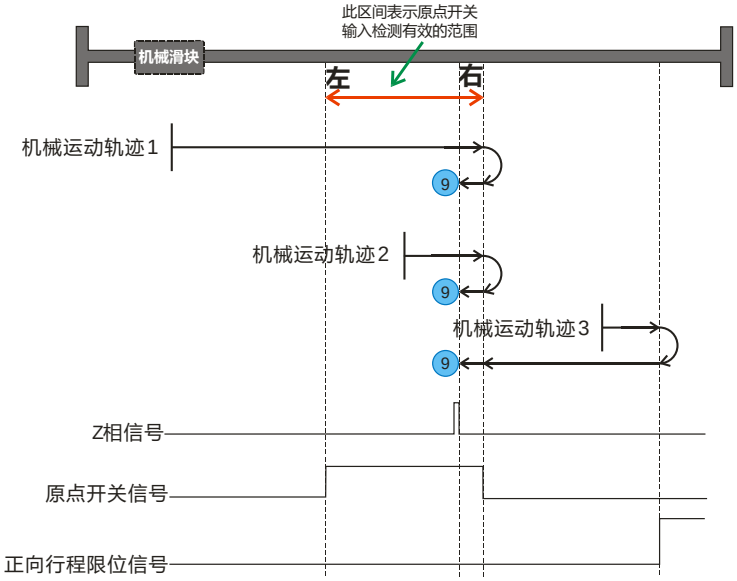
回原模式 8：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号左边沿以右的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

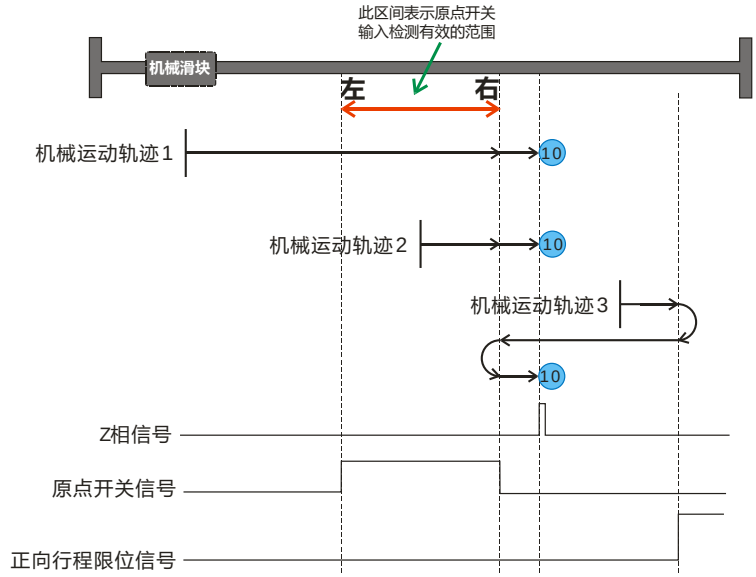
回原模式 9：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号右边沿以左的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

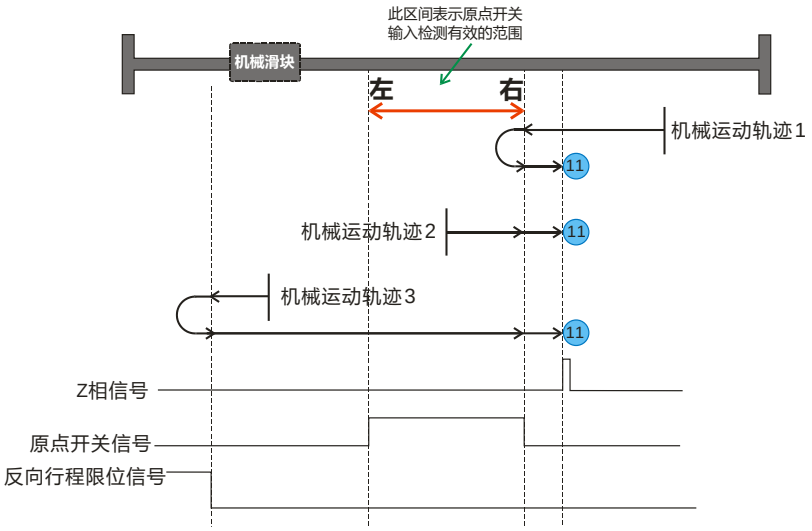
回原模式 10：参考原点开关、Z 相信号和正限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）



如上图所示，机械滑块往正限位方向（正方向）滑行，Z 相信号处于原点开关信号右边沿以右的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（正方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

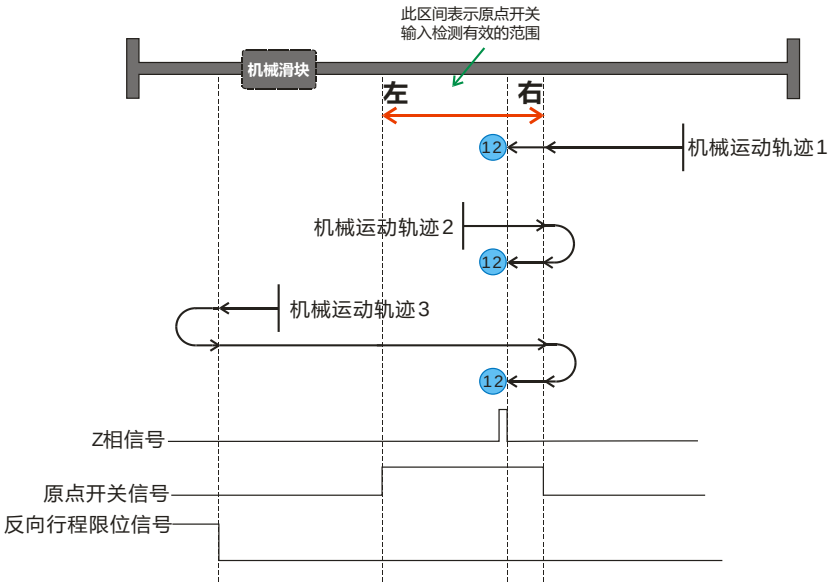
回原模式 11：参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以右的 Z 相信号）



如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号右边沿以右的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

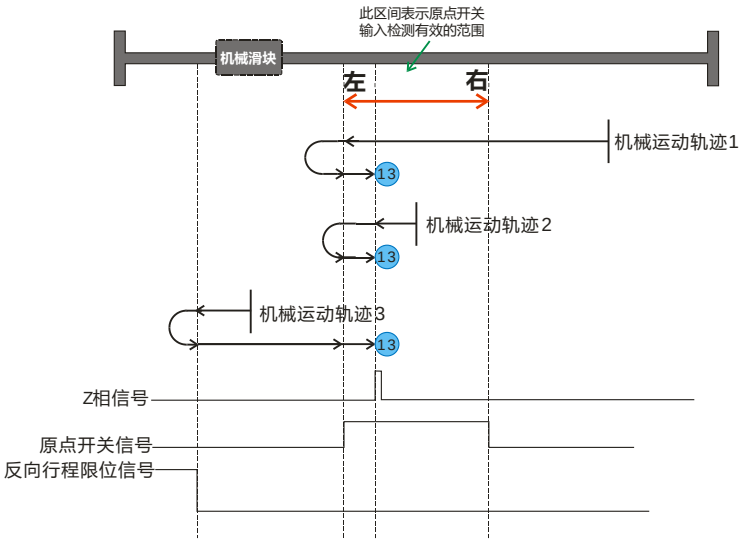
回原模式 12：参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关右边沿以左的 Z 相信号）



如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号右边沿以左的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往正方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

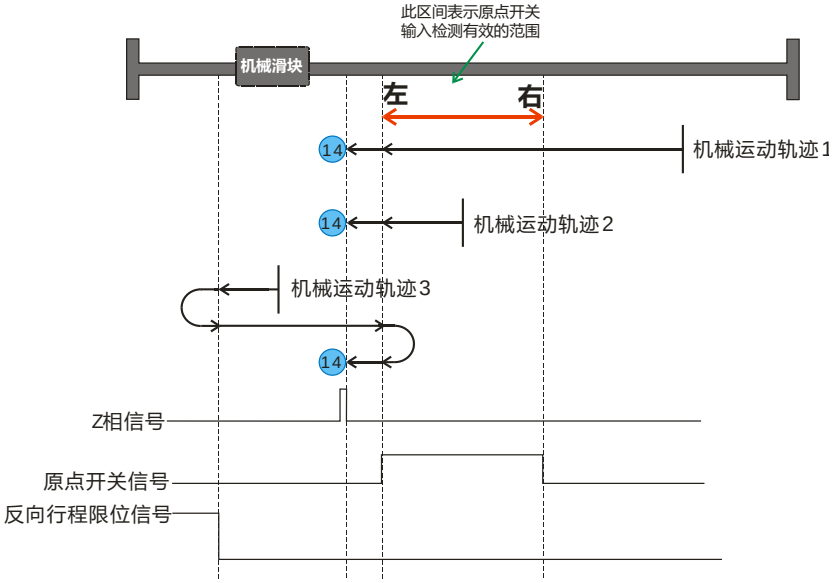
回原模式 13：参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以右的 Z 相信号）



如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号左边沿以右的位置，即原点开关信号有效范围内。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

回原模式 14：参考原点开关、Z 相信号和负限位的原点模式（采正向原点开关左边沿以左的 Z 相信号）

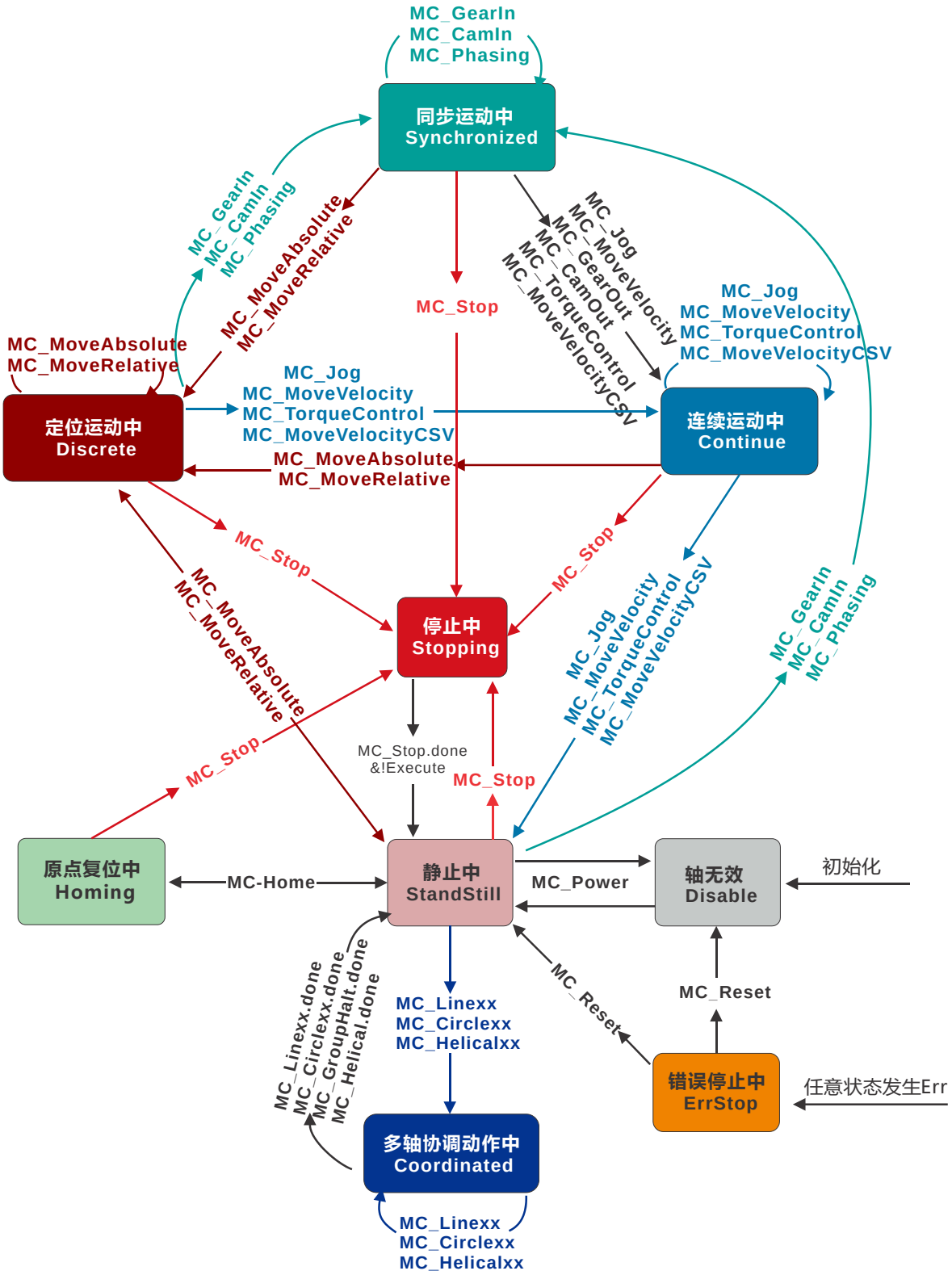


如上图所示，机械滑块往负限位方向（负方向）滑行，Z 相信号处于原点开关信号左边沿以左的位置，即原点开关信号有效范围外。

当机械处于原点开关范围内（机械运动轨迹 2），则直接往负方向运行即可寻原点；当机械处于原点开关范围外（机械运动轨迹 1 和机械运动轨迹 3），机械往限位开关方向恒定运行（负方向），根据检测到原点开关与限位开关的先后可知运动轨迹，从而可寻原点。

4.2 轴指令“plcopen_lib（v2.3）”

Axis 指令依据 PLCOpen 标准设计，轴状态指令流程图如下图所示：



提示

MagicWorks PLC(V2.22 及以上版本)支持 ct_plcopen_lib(v2.3)指令库；

脉冲轴不能与 ct_hsp300_lib 库里 MC_PTP_R、MC_PTP_A、MC_SPEED_CTRL 同时使用。

plcopen_lib(v2.3)指令分四类，分别是单轴指令、多轴指令、轴组指令和追飞剪指令，以下各章节将介绍

这四类轴指令。

4.2.1 单轴控制指令

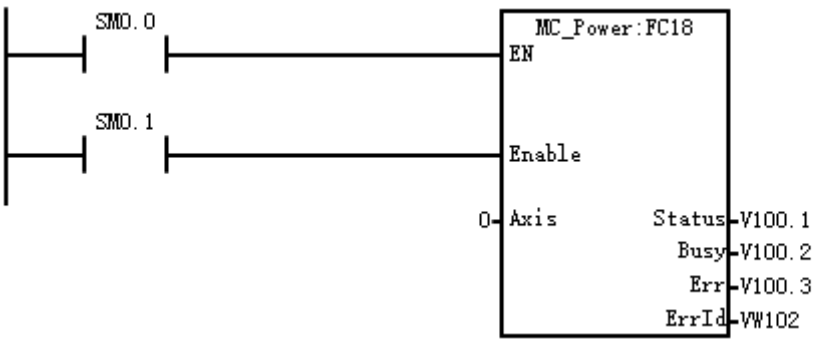
单轴控制指令一般用于定位控制，即伺服电机拖动外部机构运动到指定位置。
单轴控制常用到的 MC 功能块如下：

函数名	指令名称	适用的 PLC 型号	
		CTMC 系列 M228ML、M228SL	CTH300 系列 H31-00、H32-01、H35-00、H36-01、H52-10、H56-10
MC_Power	轴使能指令	支持	支持
MC_MoveAbsolute	绝对位移指令	支持	支持
MC_MoveRelative	相对位移指令	支持	支持
MC_MoveVelocity	速度指令	支持	支持
MC_Stop	停止指令	支持	支持
MC_Halt	可打断的停止指令	支持	支持
MC_Home	原点回归指令	支持	支持
SMC_Home	特殊原点回归指令	不支持	支持
MC_Reset	复位错误指令	支持	支持
MC_MoveFeed	中断事件触发运动指令	支持	支持
MC_MoveJog	点动指令	支持	支持
MC_SetPosition	设置当前位置指令	支持	支持
SMC_CtrlByActPos	位置跟随指令	不支持	支持
MC_ReadStatus	读取轴状态指令	支持	支持
MC_ReadSetPos	读取轴设定位置指令	支持	支持
MC_WriteParameter	写参数指令	支持	支持
MC_ReadParameter	读取参数指令	支持	支持
MC_TouchProbe	探针指令	不支持	支持
MC_TorqueControl	力矩控制指令	不支持	支持
MC_MoveVelocityCSV	速度指令（CSV 模式）	不支持	支持
MC_ReadVelPos	读取轴速度和位置指令	不支持	支持

<备注> MC_Home--驱动器回原方式、SMC_Home--控制器回原方式。

MC_Power（轴使能指令）

函数名：MC_Power



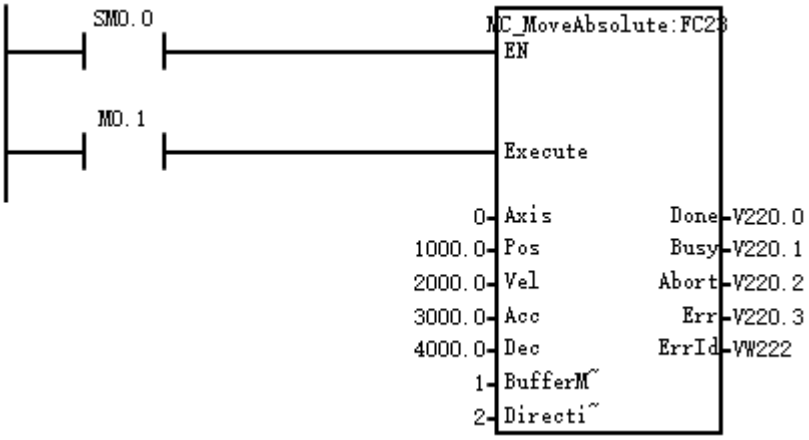
功能：此指令用于对相应的轴使能或解除使能。
参数说明：

参数名	输入输出属性	参数描述	类型	备注

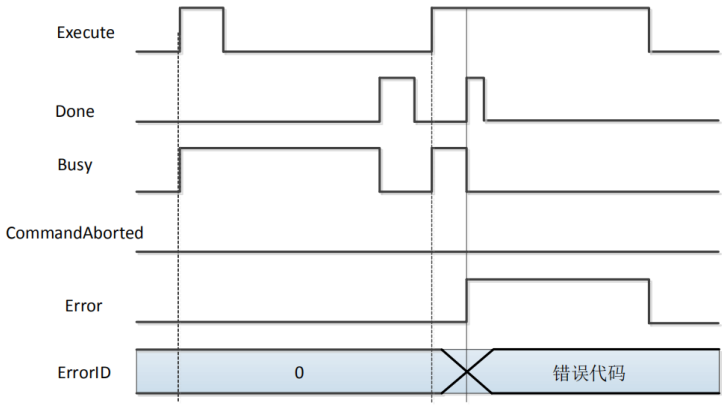
Enable	IN	轴使能	BOOL	设为TRUE则进入可运行状态，设为FALSE则解除可运行状态。
Axis	IN	轴ID号	BYTE	
Status	OUT	状态位	BOOL	进入可运行状态时变为 TRUE。 进入不可运行状态变为 FALSE。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件由On变Off时，Error位被复位。
ErrId	OUT	错误码	WORD	错误码，详见章节4.3.5 错误代码。

MC_MoveAbsolute（绝对位移指令）

函数名：MC_MoveAbsolute



功能：此指令用于控制终端执行机构按照给定速度，加减速移动到相对于零点的目标位置，此指令在运行过程中一旦被指令终止，剩余未完成的距离将被丢弃，执行新的指令功能。



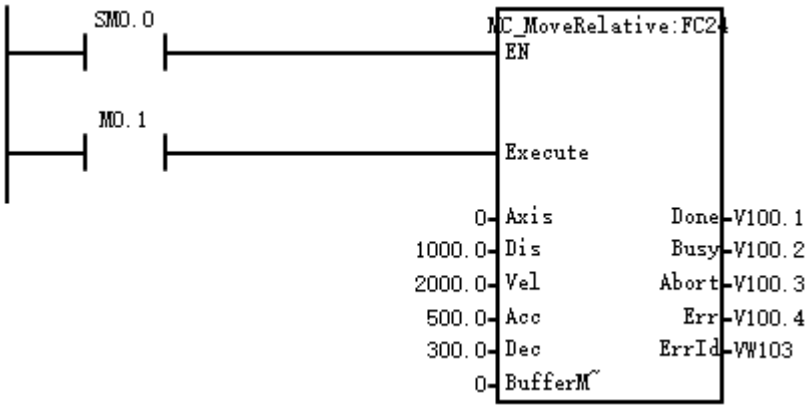
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Pos	IN	位置	REAL	终端执行机构相对零点的目标位置，单位：单元
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。（单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。（单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。（单位：

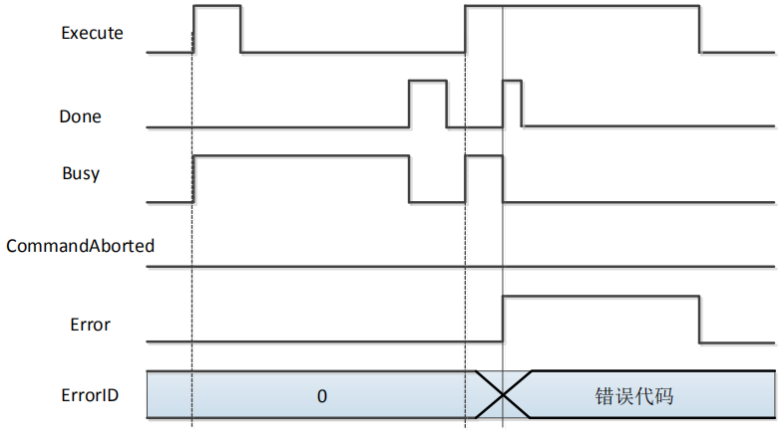
				单元/秒/秒）
BufferMode	IN	中止模式	BYTE	定义轴的动作： 0：Abort (直接打断正在进行的指令) 其他：非法
Direction	IN	方向	INT	运转方向： 0：距离最短的方向； 1：正方向； -1：反方向； 2：延续当前的方向； 该参数针对旋转轴时生效，对直线轴无效。
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位置位； 当指令的执行条件Off时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	指令执行过程中被终止时，Abort位置位； 当指令的执行条件Off时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码

MC_MoveRelative（相对位移指令）

函数名：MC_MoveRelative



功能：此指令用于控制终端执行机构以当前位置为参考点，按照给定的速度，加减速移动给定的距离，此指令在运行过程中一旦被终止，剩余未完成的距离将被丢弃，执行新的指令功能。

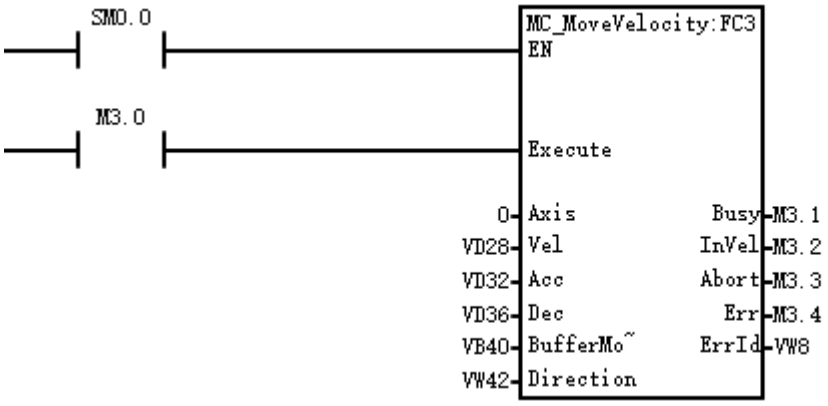


参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Dis	IN	距离	REAL	以当前位置为参考，终端执行机构要移动的目标距离，该值设置为负数代表轴将会反转。 单位：单元。
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveVelocity（速度指令）

函数名：MC_MoveVelocity



功能：此指令用于控制终端执行机构按给定的加减速运动至给定速度，并匀速运行。当终端机构到达给定速度后，此指令执行完成，但终端机构仍以此速度继续运行。

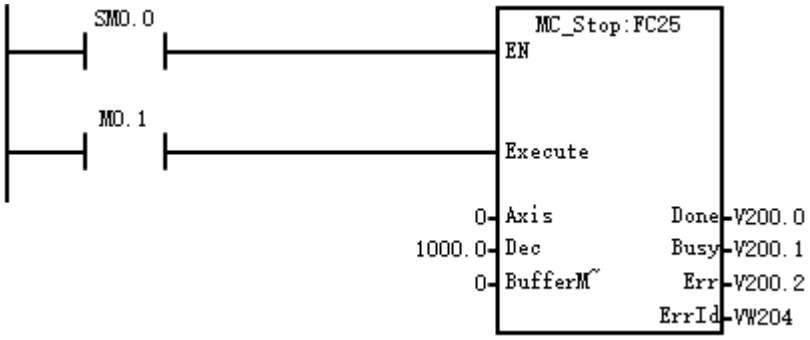
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。

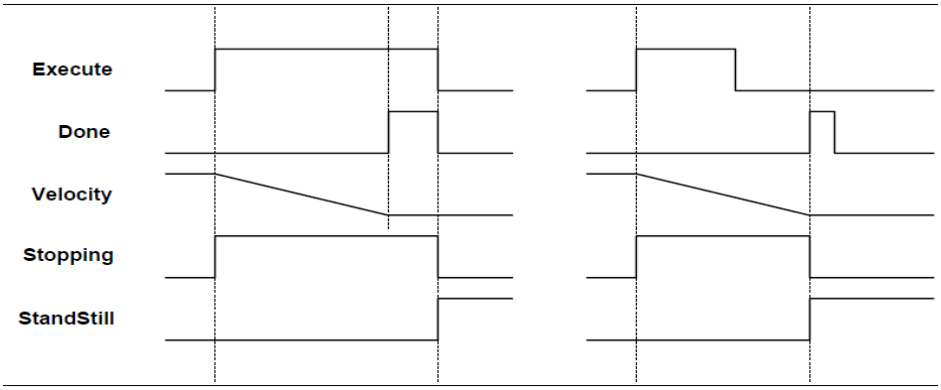
Axis	IN	轴ID号	BYTE	
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。（单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。（单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。（单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Direction	IN	方向	INT	1：正方向 -1：负方向
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Invel	OUT	速度到达位	BOOL	到达目标速度后，Invelocity位被置位；当指令的执行条件由On变Off 时，Invelocity位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_Stop（停止指令）

函数名：MC_Stop



功能：此指令用于控制终端执行机构按给定的减速度减速，直到停止。此指令运行时，不能被其它任何指令终止。速度降为 0 后，轴变 Stopping 状态，直到 Execute 变为 0，才会切到 StandStill（静止状态）。



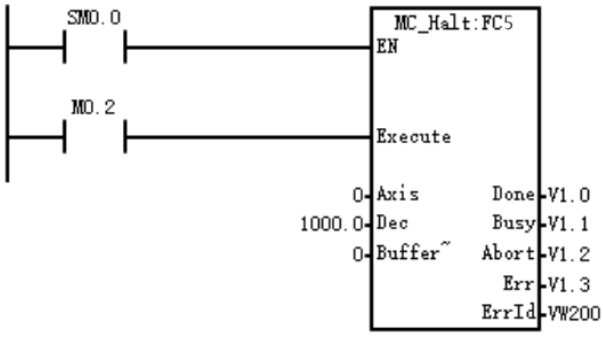
参数说明：

参数名	输入输出属性	参数描述	类型	备注

Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Done	OUT	完成位	BOOL	速度降为0后，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_Halt（可打断的停止指令）

函数名：MC_Halt



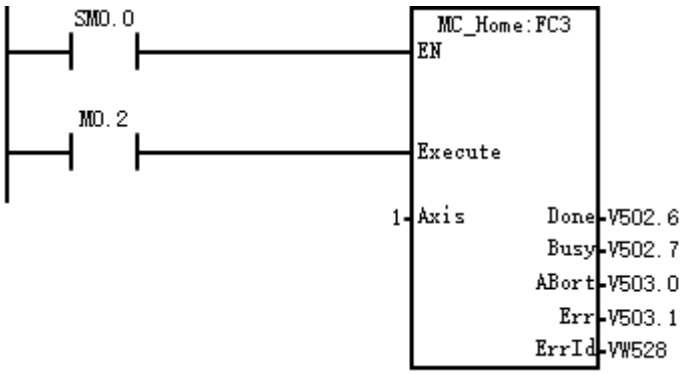
功能：此指令用于控制终端执行机构按给定的减速度减速，直到停止。此指令运行时，能被其它指令终止。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Done	OUT	完成位	BOOL	速度降为0后，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_Home（原点回归指令）

函数名：MC_Home



功能：此指令用于控制轴按轴参数给定的模式和速度执行回原点动作，回原点模式和速度在轴参数设置界面中设定。

注意：脉冲轴与通信轴的配置有所不同

• 脉冲轴是在参数设置界面设定回原模式和回原速度，详情见章节 3.2.7 轴配置 脉冲轴配置。

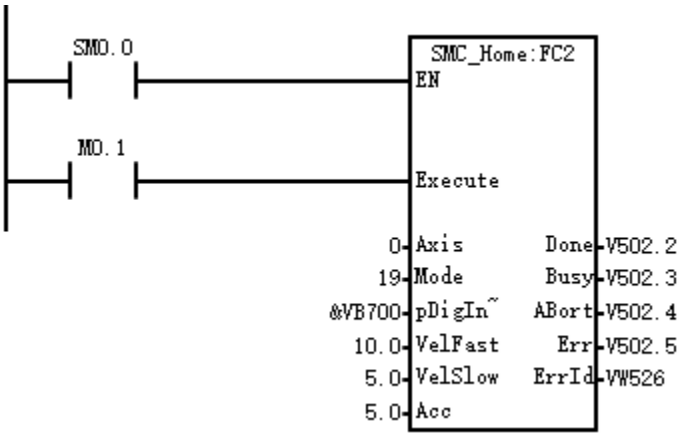
• 通信轴，则是要改写回原模式 16#6098:0 的值以及配置回原速度 16#6099:0，16#6099:1；具体操作见章节 3.2.7 轴配置 通信轴配置。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Done	OUT	完成位	BOOL	速度降为0后，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

SMC_Home（特殊原点回归指令）

函数名：SMC_Home



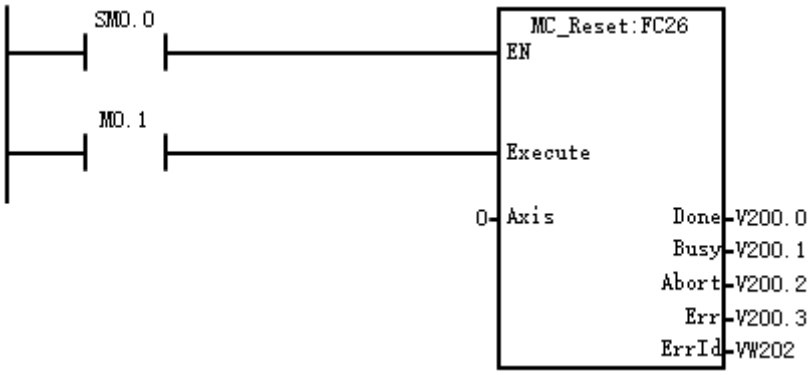
功能：此指令由控制器控制轴按参数给定的模式和速度执行回原点动作，回原模式和速度在指令中设定；注意，一般情况下，轴的原点开关/限位信号是接至 PLC 系统。回原完成后，控制器会偏置轴的坐标为0.0，轴本身的 16#6064:0(position actual value)不会因为操作指令而发生变化。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Mode	IN	回零模式	BYTE	参见脉冲轴回零示例1。
pDigInput	IN	输入信号指针	指针	原点开关，正向，反向限位开关位号指向的内存地址。 内存里 bit0：负向开关 bit1：正向开关 bit2：原点开关 开关信号1为有效，0为无效。 例如，pDigInput等于 &IB0时， I0.0：负向开关 I0.1：正向开关 I0.2：原点开关。
VelFast	IN	快速回零速度	REAL	查找原点时的快速速度，此参数总为正。 （单位：单元/秒）
VelSlow	IN	慢速回零速度	REAL	查找原点时的慢速速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Done	OUT	完成位	BOOL	回零完成后，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见4.3.5 错误代码

MC_Reset（复位错误指令）

函数名：MC_Reset



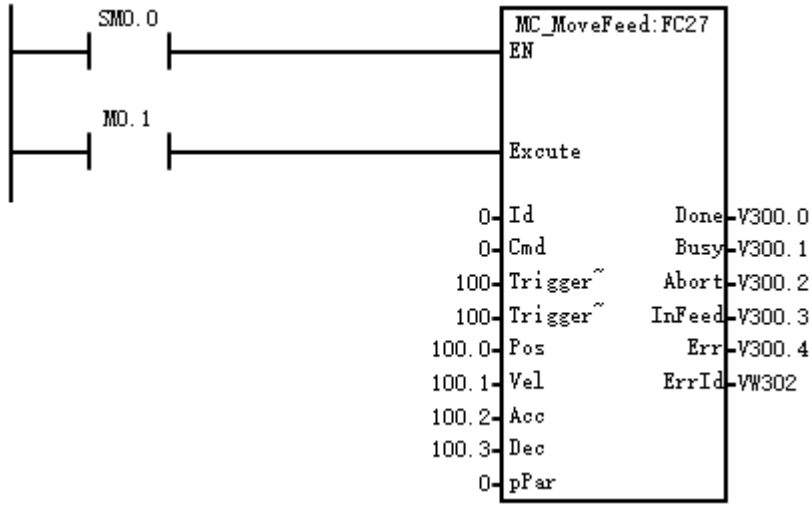
功能：用于解除轴的异常，解除异常后，轴状态将从 ErrorStop 状态转换到 StandStill 状态。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Done	OUT	完成位	BOOL	当轴状态被复位至准备执行状态后，Done 位被置位； 当指令的执行条件由 On 变 Off 时，Done 位被复位
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveFeed（中断事件触发运动指令）

函数名：MC_MoveFeed



功能：指定输入中断事件发生后，轴进行 Cmd 指定的动作。在指定中断发生之前，Execute 变为 0，可取消当前指令。运动完成后，需要再次触发才能再次响应中断。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Id		轴或轴组Id	BYTE	见下表2
CMD	IN	发生的动作	BYTE	0：停止。 1：相对运动。 2：绝对运动。 3：速度控制。 16#80：轴组停止
TriggerInput	IN	中断事件	BYTE	见下附表1
TriggerVar	IN	中断事件参数	DWORD	
Pos	IN	位置	REAL	见下附表2
Vel	IN	速度	REAL	
Acc	IN	加速度	REAL	
Dec	IN	减速度	REAL	
pPar	IN	其它参数指针	DWORD	预留。当上述参数不足描述参数时，将参数放置于pPar指示的内存里面。
Done	IN	完成位	BOOL	当轴状态被复位至准备执行状态后，Done 位被置位； 当指令执行条件由 On 变 Off 时，Done 位复位
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
InFeed	OUT	运动进行位	BOOL	当中断发生,并触发运动时，InFeed被置位。 当指令完成进InFeed被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

附表 1 中断触发事件

TriggerInput	意义	TriggerVar	
		CTMC系列M228ML、M228SL	CTH300系列H32-01、H31-00、H36-01、H35-00、H56-10、H52-10
0	上升沿，I0.0	不适用，默认为 0	发生中断事件的模块 TriggerVar=机架号*256 + 槽号 如模块放置于机架号为 1，槽号为 3 的位置， 则 TriggerVar =16#0103
2	上升沿，I0.1		
4	上升沿，I0.2		
6	上升沿，I0.3		
1	下降沿，I0.0		
3	下降沿，I0.1		
5	下降沿，I0.2		
7	下降沿，I0.3		
12	HSC0 CV=PV		
27	HSC0 方向改变		
28	HSC0 外部复原 / Zphase		

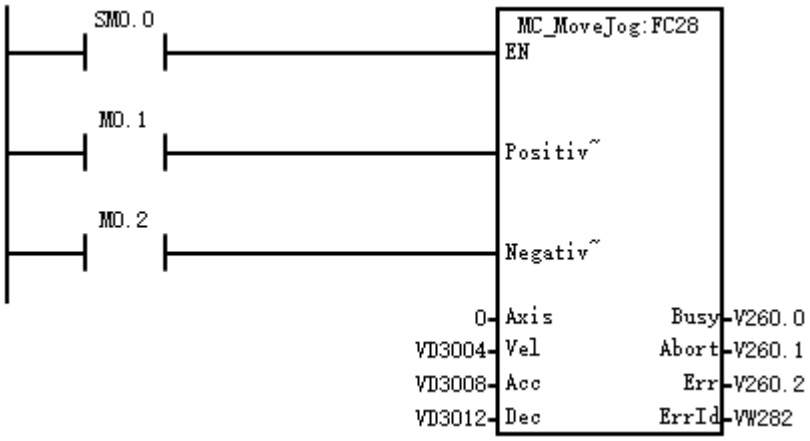
13	HSC1CV=PV		
14	HSC1 方向改变		
15	HSC1 外部复原 / Zphase		
128	V 内存上升沿中断	TriggerVar =内存偏移地址* 8 + 内存的位。如 V0.0 此值为 0；V127.7 此值为 1023（127*8+7）。	
129	V 内存下降沿中断		

附表 2 轴运动

CMD	参数	详细描述
0：停止。	Id：轴 ID 号 Dec：减速度	相当于中断触发后执行 MC_Stop 指令。
1：相对运动。	Id：轴 ID 号 Vel：速度 Acc：加速度 Dec：减速度 Pos：运动的距离	相当于中断触发后执行 MC_MoveRelative
2：绝对运动。	Id：轴 ID 号 Vel：速度 Acc：加速度 Dec：减速度 Pos：位置 当轴为旋转轴时： VEL > 0 方向为正向旋转 VEL < 0 方向为反向旋转	相当于中断触发后执行 MC_MoveAbsolute
3：速度运动。	Id：轴 ID 号 Vel：速度 Acc：加速度 Dec：减速度 Vel > 0.0 正向 Vel < 0.0 反向	相当于中断触发后执行 MC_MoveVelocity
16#80：轴组停止	Id：轴组 ID 号 Dec：减速度	相当于中断触发后执行 MC_GroupHalt

MC_MoveJog（点动指令）

函数名：MC_MoveJog



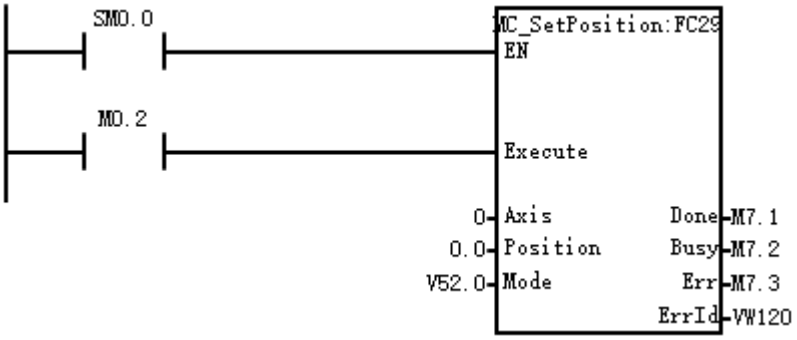
功能：根据指定 Velocity(目标速度)执行微动移动。需要正方向微动移动时，将 PositiveEnable(正方向有效)设为 TRUE，需要负方向微动移动时，将 NegativeEnable(负方向有效)设为 TRUE。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
PositiveEnable	IN	正方向有效	BOOL	设为TRUE，则开始正方向移动。设为FALSE，则结束移动。
NegativeEnable	IN	负方向有效	BOOL	设为 TRUE，则开始负方向移动。设为 FALSE，则结束移动。
Axis	IN	轴ID号	DWORD	轴号
Vel	IN	目标速度	REAL	指定目标速度。单位为［指令单位/秒］。
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒） 当设为0时，实现紧急停止。
Busy	OUT	忙位	BOOL	指令正在执行为TRUE，执行完成为FLASE。
Abort	OUT	中止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_SetPosition（设置当前位置指令）

函数名：MC_SetPosition



功能：此指令设定当前坐标的值(单元)。

注意：本指令不能在轴运动时使用，当轴作为同步运动的主轴，也不可以使用，不然可能发生不可预知的后果。

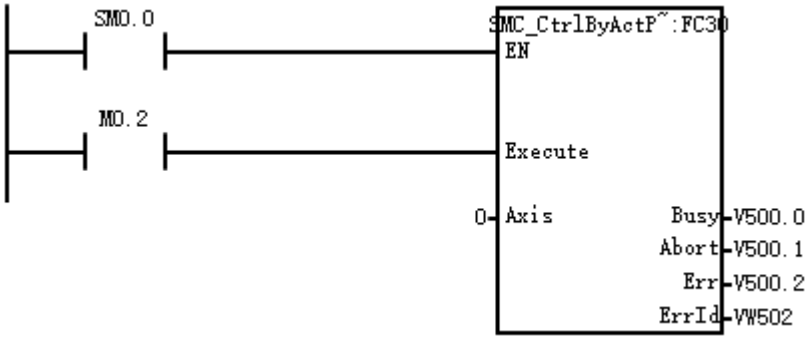
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	轴号
Position	IN	指定的坐标位置	REAL	指定的坐标位置（单元） 取值-1E+13<=Position<=1E+13
Mode	IN	位置模式	BOOL	0：绝对位置模式.指定为此模式，则轴的绝对位置为Position 1：相对位置模式.指定为此模式，则轴的绝对位置为当前位置加Position
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位； 当指令的执行条件 Off 时，Done 位被复位。
Busy	OUT	忙位	BOOL	指令正在执行为TRUE，执行完成为FLASE
Error	OUT	错误位	OBOL	如果检测到有错误，Error位被置位；

				当指令的执行条件由On变Off时，Error位被复位。
ErrId	OUT	错误码	WORD	错误代码，详见章节4.3.5 错误代码。

SMC_CtrlByActPos（位置跟随指令）

函数名：SMC_CtrlByActPos



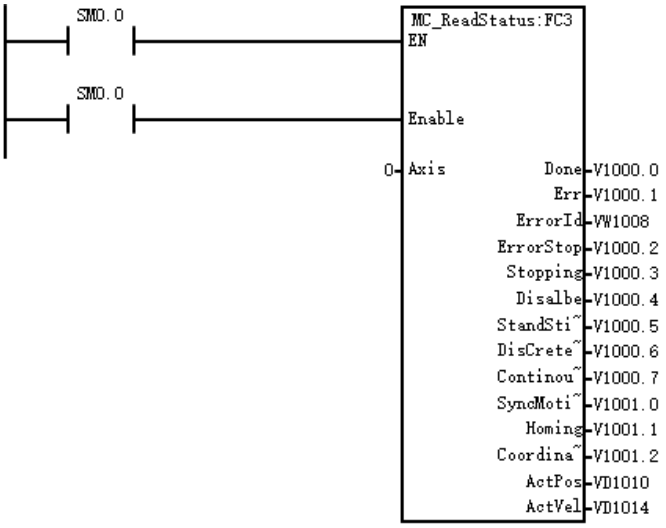
功能：调用此指令后，轴位置将由外部指令控制，轴设定位置将跟随当前位置。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由 Off 变 On 时，执行该指令。 当执行条件由 ON 变 Off 时，取消当前指令
Axis	IN	主轴 ID 号	BYTE	轴号
Busy	OUT	指令处理中	BOOL	True，指令正在处理中
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort 位被置位； 当指令的执行条件 Off 时，Abort 位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error 位被置位； 当指令的执行条件 Off 时，Error 位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节 4.3.5 错误代码

MC_ReadStatus（读取轴状态指令）

函数名：MC_ReadStatus



功能：此指令用于读取相应轴状态参数的值。

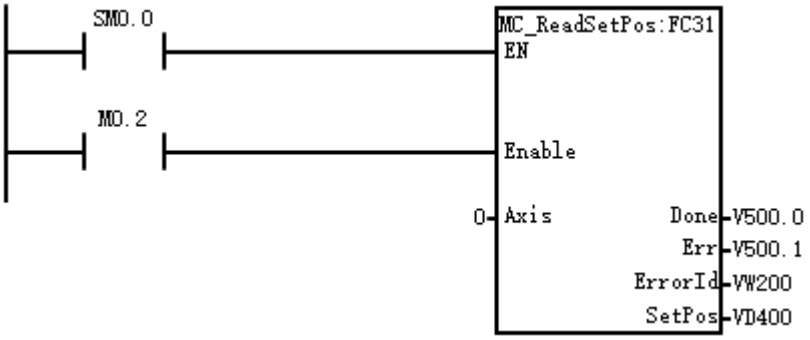
参数说明：

参数名	输入输出属性	参数描述	类型	备注

enable	IN	指令执行条件	BOOL	当为On时，执行该指令。
Axis	IN	轴Id号	BYTE	轴号
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位； 当指令的执行条件 Off 时，Done 位被复位。
Error	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件由On变Off时，Error位被复位。
ErrId	OUT	错误码	WORD	错误码 当Error为TRUE时，ErrId为指令的错误。 当ErrStop为TRUE时，ErrId为轴当前错误。
ErrStop	OUT	状态指示位	BOOL	TRUE，如果轴位于状态ErrStop
Stopping	OUT	状态指示位	BOOL	TRUE，如果轴位于状态Stopping
Disabled	OUT	状态指示位	BOOL	TRUE，如果轴位于状态Disabled
StandStill	OUT	状态指示位	BOOL	TRUE，如果轴位于状态StandStill
DisCreteMotion	OUT	状态指示位	BOOL	TRUE，如果轴位于状态DisCreteMotion
Continous Motion	OUT	状态指示位	BOOL	TRUE，如果轴位于状态ContinousMotion
SyncMotion	OUT	状态指示位	BOOL	TRUE，如果轴位于状态SyncMotion
Homing	OUT	状态指示位	BOOL	TRUE，如果轴位于状态Homing
Coordinate	OUT	状态指示位	BOOL	TRUE，如果轴位于状态Coordinate
ActPos	OUT	当前坐标	REAL	单元
ActVel	OUT	当前速度	REAL	单元/秒

MC_ReadSetPos（读取轴设定位置指令）

函数名：MC_ReadSetPos



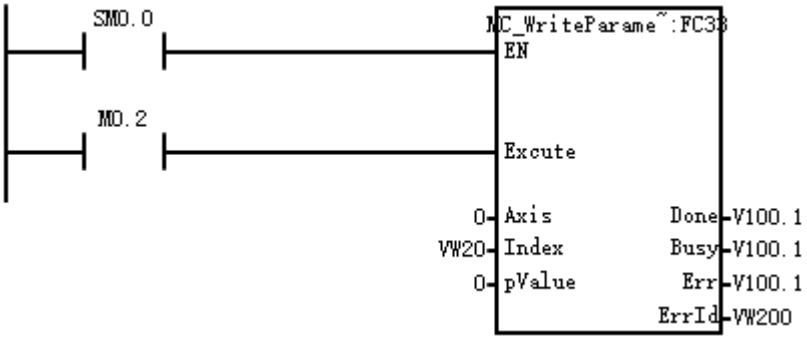
功能：此指令用于读取轴的当前设定位置。

参数说明

参数名	输入输出属性	参数描述	类型	备注
enable	IN	指令执行条件	BOOL	当为 On 时，执行该指令。
Axis	IN	轴 ID 号	BYTE	轴号
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位； 当指令的执行条件 Off 时，Done 位被复位。
Error	OUT	错误位	BOOL	如果检测到有错误，Error 位被置位； 当指令的执行条件由 On 变 Off 时，Error 位被复位。
ErrId	OUT	错误码	WORD	当 Error 为 TRUE 时，ErrId 为指令的错误。 当 ErrStop 为 TRUE 时，ErrId 为轴当前错误。
Setpos	OUT	当前设定位置	REAL	单元

MC_WriteParameter（写参数指令）

函数名：MC_WriteParameter

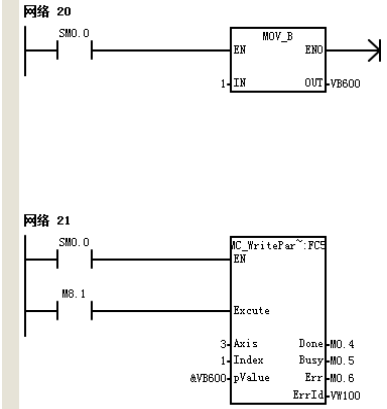


功能：此指令用于写入相应的轴的配置参数的值，写入成功后，相当于修改了硬件组态，永久有效。只能在轴未使能的情况下运行该指令。

参数说明

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Index	IN	参数的序号	WORD	见下表**参数描述表**
pValue	IN	参数值的指针	DWORD	参数指针
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位； 当指令的执行条件 Off 时，Done 位被复位。
Busy	Out	忙位	BOOL	指令正在执行为 TRUE，执行完成为 FLASE
Error	OUT	错误位	BOOL	如果检测到有错误，Error位被置位；当指令的执行条件由On变Off时，Error位被复位。
ErrId	OUT			错误码

参数描述表

Index	参数	pValue	说明
0	量纲转换	pValue 为指向量纲描述的指针。量纲描述见下附表 3。	
1	加减速方式	pValue：为指向加减速方式的指针。加减速方式为一个 BYTE 加速模式，0：梯形；1：S 形；2：Sin 加减速	轴 3 加减速方式为 S 形加减速 
2	速度	pValue：为指向速度设定的指针。指向结构见附表 4	

附表 3 纲量转换数据结构说明

名称	类型	说明
ScalPulseInc	REAL	量纲转换，脉冲增量

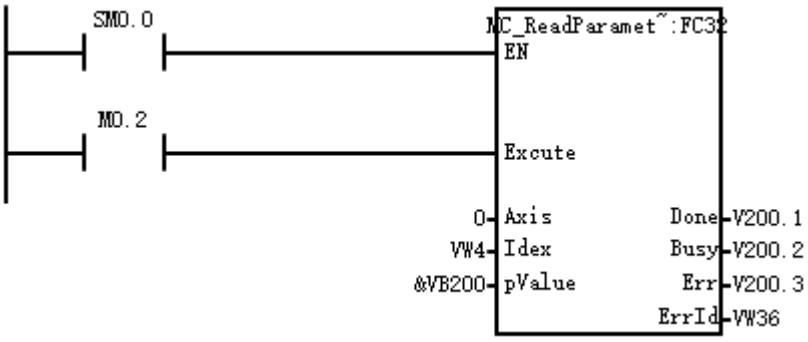
		（硬件组态量纲转换左 1）
ScalMotorTurns	REAL	量纲转换，电机圈数 （硬件组态量纲转换右 1）
ScalMotorTurnsInc	REAL	量纲转换，电机圈数增量 （硬件组态量纲转换左 2）
ScalGearTurnsInc	REAL	量纲转换，齿轮圈数增量 （硬件组态量纲转换右 2）
ScalGearTurnsInc	REAL	量纲转换，齿轮圈数增量 （硬件组态量纲转换左 3）
ScalAppUnitsc	REAL	量纲转换，工程单位 （硬件组态量纲转换左 3）

附表 4 速度设定说明

名称	类型	说明
VelMax	REAL	最大速度
VelMin	REAL	最小速度
VelStartStop	REAL	启动停止速度
AccMax	REAL	最大加速度
AccMin	REAL	最小加速度

MC_ReadParameter（读取参数指令）

函数名：MC_ReadParameter



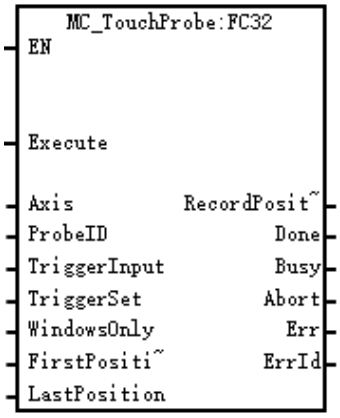
功能：此指令用于读取相应的轴的配置参数的值。

参数说明

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Index	IN	参数的序号	WORD	见参数描述表
pValue	OUT	参数指针	DWORD	
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位； 当指令的执行条件 Off 时，Done 位被复位。
Busy	OUT	忙位	BOOL	指令正在执行为 TRUE，执行完成为 FLASE
Error	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件由On变Off时，Error位被复位。
ErrId	OUT	错误码		错误码

MC_TouchProbe（探针指令）

函数名：MC_TouchProbe



功能：记录发生触发事件时的实际位置。

此指令如果选择通信轴作为来源，PDO 中需要增加探针功能的对象字典，探针 1 为 0x60B8，10x60B9，0x60BA，0x60BB；探针 2 为 0x60B8，0x60B9，0x60BC ，0x60BD。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
Axis	IN	轴ID号	BYTE	轴号
ProbeID	IN	探针ID	BYTE	0：PLC中断 1：通讯轴Probe1 2：通讯轴Probe2
TriggerInput	IN	探针来源	BYTE	对PLC中断：中断事件号吗，详情见下表-- **PLC中断探针来源** 对通讯轴： • 通讯轴Probe1：使能指令后此设置会映射到驱动器16#60B8：00的2-3位 0：DI输入 1：Z相 2，3：驱动器自定 • 通讯轴Probe2：使能指令后此设置会映射到驱动器16#60B8：00的10-11位） 0：DI输入 1：Z相 2，3：驱动器自定
TriggerSet	IN	探针设定	BYTE	• 对PLC中断：触发中断的模块位置，如：模块放在机架号1，槽号5，则地址为16#15。 • 对通讯轴：使能指令后此设置会映射到驱动器16#60B8：00的4-5位 0：上升沿 1：下降沿
WindowsOnly	IN	窗口有效	BOOL	指定窗口 1 为有效，0 为无效。
FirstPosition	IN	起始位置	REAL	指定接收触发的开始位置。

LastPosition	IN	终止位置	REAL	指定接收触发的结束位置。
Recordposition	OUT	触发记录位置	REAL	触发发生时当前的位置
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位； 当指令的执行条件 Off 时，Done 位被复位。
Busy	OUT	忙位	BOOL	指令正在执行为TRUE，执行完成为FLASE
Abort	OUT	中止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Error	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件由On变Off时，Error位被复位。
ErrId	OUT	错误码	WORD	错误代码，详见章节14.3.4 错误代码。

以下是探针指令中的 TriggerInput 参数为 PLC 中断时的中断事件号码表、WindowsOnly 窗口有效、驱动器 16#60B8 bit 位定义的说明。

1、PLC 中断探针来源：中断事件号码表

中断事件号码	优先级别组别	说明
0	1	上升边沿，I0.0
2	1	上升边沿，I0.1
4	1	上升边沿，I0.2
6	1	上升边沿，I0.3
48	1	上升边沿，I0.4
50	1	上升边沿，I0.5
52	1	上升边沿，I0.6
54	1	上升边沿，I0.7
56	1	上升边沿，I1.0
58	1	上升边沿，I1.1
1	1	下降边沿，I0.0
3	1	下降边沿，I0.1
5	1	下降边沿，I0.2
7	1	下降边沿，I0.3
49	1	下降边沿，I0.4
51	1	下降边沿，I0.5
53	1	下降边沿，I0.6
55	1	下降边沿，I0.7
57	1	下降边沿，I1.0
59	1	下降边沿，I1.1
12	1	HSC0 CV=PV
27	1	HSC0方向改变
28	1	HSC0外部复原/Zphase
13	1	HSC1 CV=PV
14	1	HSC1方向改变
15	1	HSC1外部复原/Zphase

中断事件号码	优先级别组别	说明
16	1	HSC2 CV=PV
17	1	HSC2方向改变
18	1	HSC2外部复原/Zphase（不支持）
32	1	HSC3 CV=PV
38	1	HSC3方向改变
40	1	HSC3外部复原/Zphase（不支持）
29	1	HSC4 CV=PV
30	1	HSC4方向改变（不支持）
31	1	HSC4外部复原/Zphase（不支持）
33	1	HSC5 CV=PV
39	1	HSC5方向改变（不支持）
41	1	HSC5外部复原/Zphase（不支持）
19	1	PTO 0完成中断（不支持）
20	1	PTO 1完成中断（不支持）

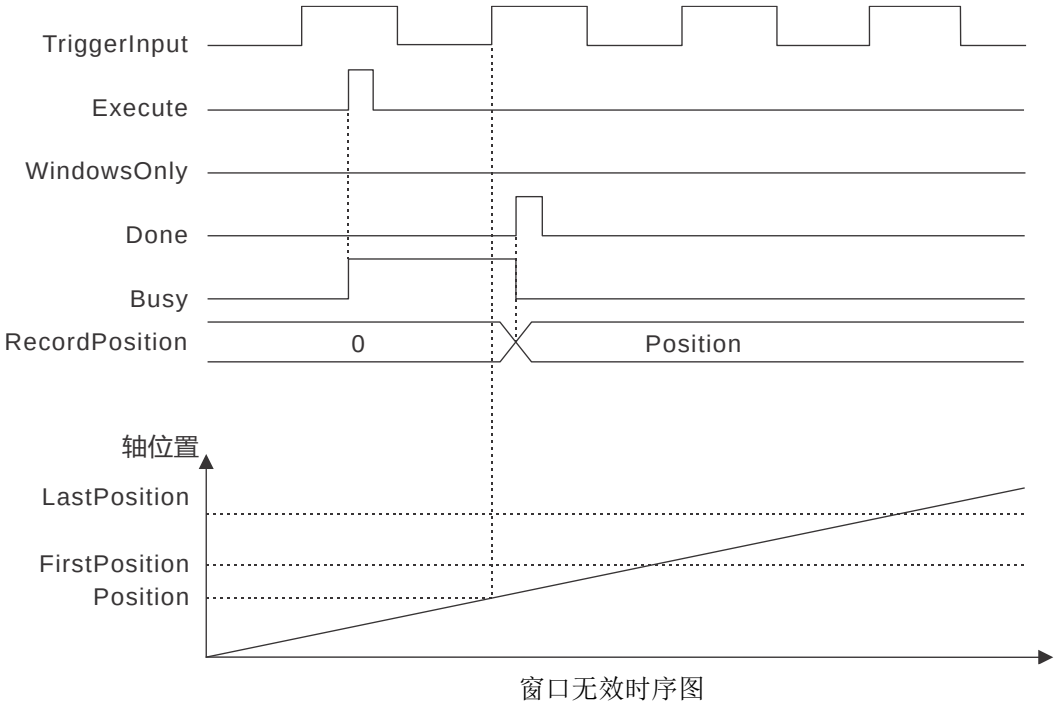
2、WindowsOnly

- 在 WindowsOnly 中，指定窗口 1 为有效，0 为无效。
- 指定 Disable 时，在所有轴位置检出触发。
- 指定 Enable 时，仅轴位置在 FirstPosition（起始位置）和 LastPosition（终止位置）的范围内检出触发。

<备注> WindowsOnly 由 FALSE 变化为 TRUE 的瞬间以及锁定功能启动之间的时间无法锁定。
锁定功能启动需要时间，因此 WindowsOnly 的有效范围极短时无法锁定，可锁定有效范围的临界值取决于伺服驱动器和编码器输入终端的性能和 EtherCAT 通信。

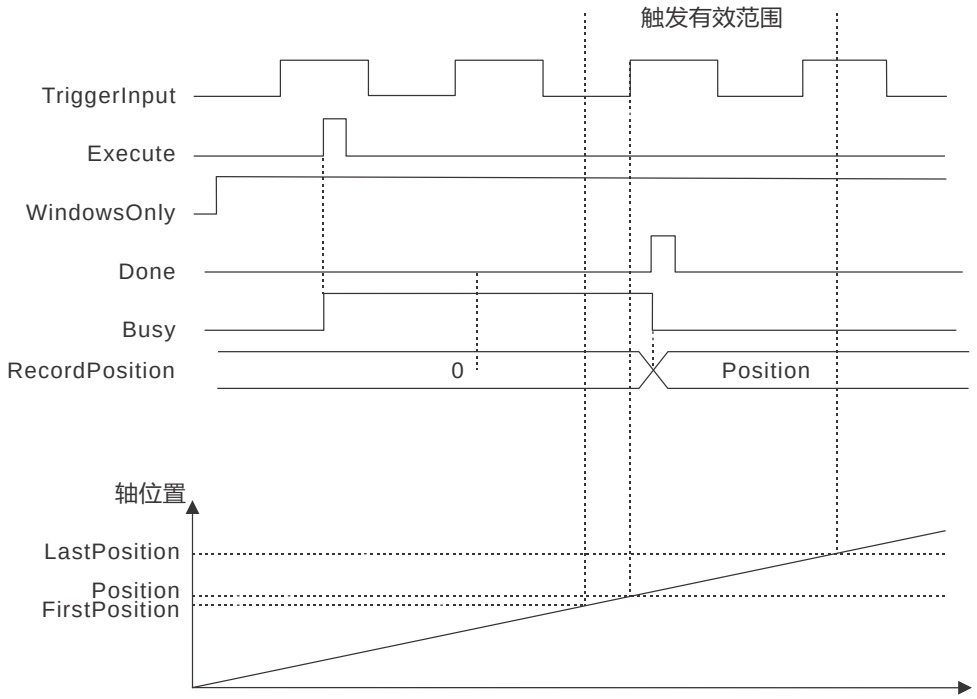
1） WindowsOnly 的指定不同，动作也不同，如下时序图所示。

• WindowsOnly 指定窗口为 0 时，将 Excute（执行条件）变为 TRUE 后，最初触发的轴位置输出到 Recordposition（锁定位置）。



窗口无效时序图

• WindowsOnly 指定窗口为 1，仅在窗口的范围内检出触发输入，获取轴位置



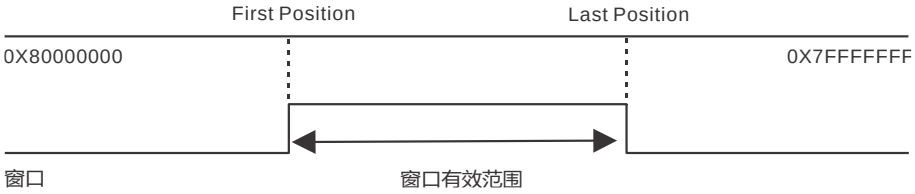
窗口有效时序图

2） 不同计数模式下 FirstPosition（起始位置）和 LastPosition（终止位置）的范围

线性模式

线性模式时的窗口有效范围如下图所示：

窗口有效范围包含 FirstPosition（起始位置）和 LastPosition（终止位置）



<备注> 当 FirstPostion > LastPostion 时，LastPostion <=RecordedPostion<= firstPostion 窗口有效。

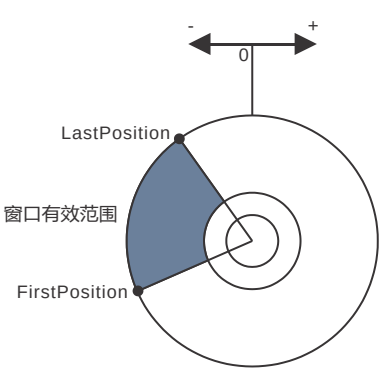
旋转模式

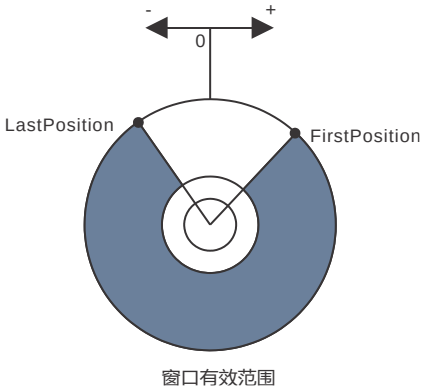
• FirstPosition（起始位置）≤ LastPosition（终止位置），FirstPosition（起始位置）> LastPosition（终止位置）两者均可指定。

• FirstPosition（起始位置）> LastPosition（终止位置）时，设定值跨越环计数器的上下限位置。

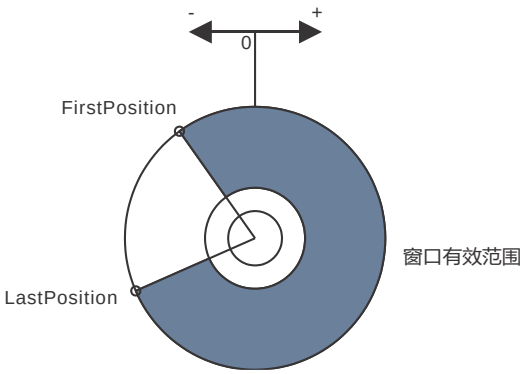
• 超过环计数器上下限范围指定时，会发生异常。

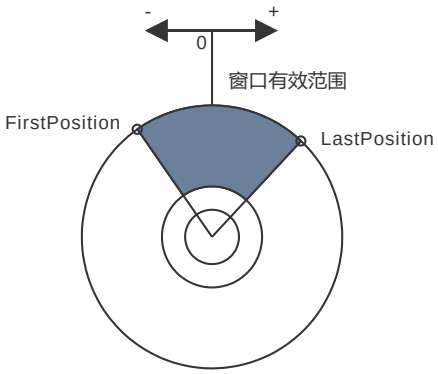
FirstPosition（起始位置）≤ LastPosition（终止位置）





FirstPosition（起始位置）> LastPosition（终止位置）





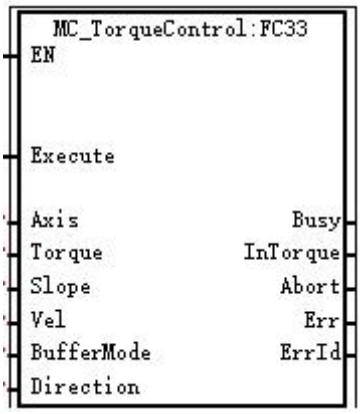
<备注>： FirstPostion和LastPostion设定超过旋转轴模值范围时，将被修正到模值范围之内。

3、驱动器 16#60B8 bit 位定义：

Bit 位	描述	
0	探针 1 使能 0：探针 1 不使能 1：探针 1 使能	Bit0~Bit5：探针 1 设置
1	探针 1 触发模式 0：单次触发，只在触发信号第一次有效时触发。 1：连续触发	
2	探针 1 触发信号选择 0：DI8 输入信号 1：Z 信号	
3	NA	
4	探针 1 上降沿使能 0：上降沿不锁存 1：上降沿锁存	
5	探针 1 下降沿使能 0：下降沿不锁存 1：下降沿锁存	
6~7	NA	
8	探针 2 使能 0：探针 2 不使能 1：探针 2 使能	Bit8~Bit13：探针 2 设置
9	探针 2 触发模式 0：单次触发，只在触发信号第一次有效时触发。 1：连续触发	
10	探针 2 触发信号选择 0：DI9 输入信号 1：Z 信号	
11	NA	
12	探针 2 上降沿使能 0：上降沿不锁存 1：上降沿锁存	
13	探针 2 下降沿使能 0：下降沿不锁存 1：下降沿锁存	
14~15	NA	

MC_TorqueControl（力矩控制指令）

函数名：MC_TorqueControl



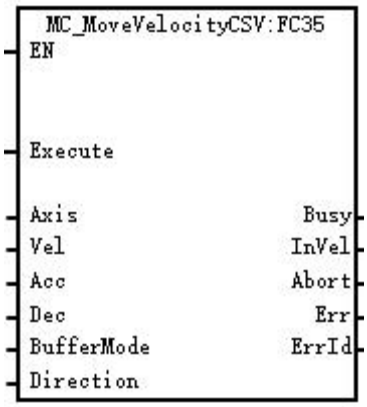
功能：矩控制指令，本指令通过力矩模式来控制标准的 402 轴做转矩运动，PDO中需要增加 0x6071，0x6060 和 0x6071 三个对象字典；此指令只用于通信轴,不可用于脉冲轴与虚拟轴。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Torque	IN	目标转矩	REAL	以“0.1%”为单位指定向伺服驱动器输出的目标转矩。 以额定转矩为“100.0%”时的比率进行指定
Slope	IN	转矩加速度	REAL	从当前转矩到目标转矩的转矩变化率，单位为%/s
Vel	IN	速度限制	REAL	限制速度，单位：单元/秒
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Direction	IN	方向	INT	1：正方向 -1：负方向
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off时，Busy位被复位。
InTorque	OUT	转矩到达位	BOOL	到达目标转矩后，InTorque位被置位；当指令的执行条件由On变Off时，InTorque位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节14.3.4 错误代码。

MC_MoveVelocityCSV（速度指令（CSV 模式））

函数名：MC_MoveVelocityCSV



功能：此指令用于控制终端执行机构按给定的加减速运动至给定速度，并匀速运行。当终端机构到达给定速度后，此指令执行完成，但终端机构仍以此速度继续运行。

此指令的功能与 MC_MoveVelocity 相同，但采用周期性同步速度模式用于控制伺服轴。PDO 中需要增加 0x6060，0x6061 和 0x60ff 三个对象字典，此指令只用于通信轴。

需要注意的是当调用该指令时，不能调用 MC_MoveSuperImosed 指令进行运动叠加动作。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Axis	IN	轴ID号	BYTE	
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。（单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。（单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。（单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Direction	IN	方向	INT	1：正方向 -1：负方向
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Invel	OUT	速度到达位	BOOL	到达目标速度后，Invel位被置位；当指令的执行条件由On变Off时，Invel位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位；当指令的执行条件Off时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节14.3.4 错误代码。

MC_ReadVelPos（读取轴速度和位置指令）

函数名：MC_ReadVelPos

功能：此指令用于读取轴的速度和位置。

参数说明

参数名	输入输出属性	参数描述	类型	备注
Axis	IN	轴 ID 号	BYTE	轴号
Err	OUT	错误位	BOOL	如果检测到有错误，Err 位被置位；
ErrId	OUT	错误码	WORD	当 Err 为 TRUE 时，ErrId 为指令的错误。
Actpos	OUT	当前实际位置	REAL	单元
ActVel	OUT	当前实际速度	REAL	单元/秒
Setpos	OUT	当前设定位置	REAL	单元
SetVel	OUT	当前设定速度	REAL	单元/秒

4.2.2 同步控制指令

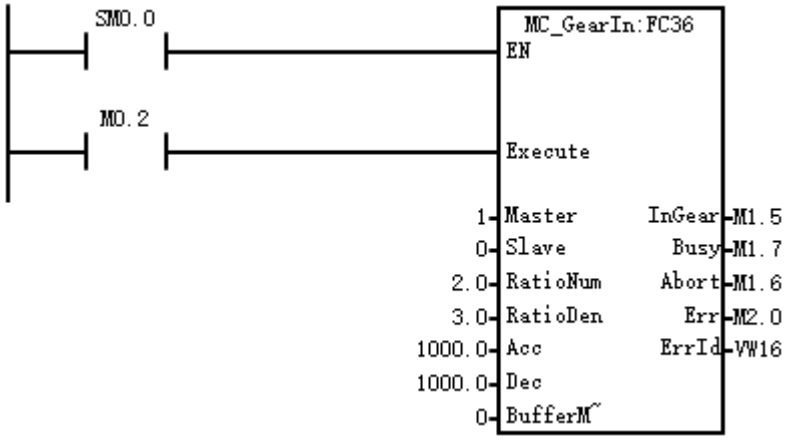
同步控制一般用于电子齿轮、电子凸轮、相位偏移指令等。

同步控制指令常用到的 MC 功能块如下：

函数名	指令名称	适用的 PLC 型号	
		CTMC 系列 M228ML、M228SL	300-H 系列 H31-00、H32-01、H35-00、H36-01、H52-10、H56-10
MC_GearIn	电子齿轮耦合指令	支持	支持
MC_GearOut	电子齿轮脱离指令	支持	支持
MC_CamTableSelect	凸轮表选择指令	支持	支持
MC_CamIn	电子凸轮关联指令	支持	支持
MC_CamOut	解除电子凸轮关联指令	支持	支持
MC_Phasing	相位偏移指令	支持	支持
MC_MoveSuperImposed	叠加相对位移指令	支持	支持

MC_GearIn（电子齿轮耦合指令）

函数名：MC_GearIn



功能：此指令用于建立主从轴间的齿轮关系。建立齿轮关系时，可设定齿轮比等参数。齿轮关系建立后从轴以给定的比例关系跟随主轴运动，实现主从轴同步控制。

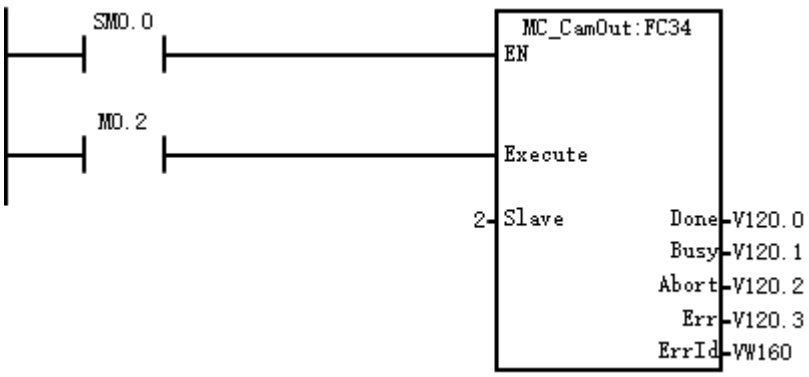
同步完成之后从轴速度=主轴速度 * RatioNum/RatioDen。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
Master	IN	主轴ID号	BYTE	
Slave	IN	从轴ID号	BYTE	
RatioNum	IN	电子齿轮的分子	REAL	非0
RetioDen	IN	电子齿轮的分母	REAL	非0
Acc	IN	加速度	REAL	
Dec	IN	减速度	REAL	
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
InGear	OUT	耦合完成位	BOOL	耦合完成变InGear置位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_GearOut（电子齿轮脱离指令）

函数名：MC_GearOut



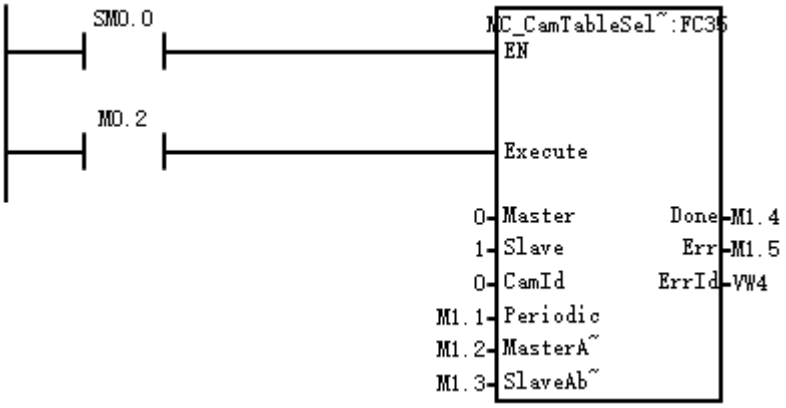
功能：此指令用于解除主从轴间的齿轮关系，关系解除后，从轴会以脱离前的速度继续运行。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Slave	IN	从轴ID号	BYTE	
Done	OUT	完成位	BOOL	当取消齿轮关联指令执行成功时，Done位被置位。 当指令执行条件Off时，Done位复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位；
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_CamTableSelect（凸轮表选择指令）

函数名：MC_CamTableSelect



功能：此指令用于选择凸轮曲线，同时指定主从轴关联时的模式。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Master	IN	主轴ID号	BYTE	
Slave	IN	从轴ID号	BYTE	
CamId	IN	凸轮表的ID号	BYTE	
Periodic	IN	周期	BOOL	此参数为1时，从轴会将周期性地执行凸轮运动。 此参数为0时，从轴只执行一个周期的凸轮运动。
MasterAbsolute	IN	主轴绝对位置模式	BOOL	当此参数为1时，主轴为绝对位置模式。 当此参数为0时，主轴为相对位置模式
SlaveAbsolute	IN	从轴绝对位置模式	BOOL	此参数为1时，从轴为绝对位置模式。 此参数为0时，从轴为相对位置模式。
Done	OUT	完成位	BOOL	当凸轮参数设置成功时，Done位被置位。 当指令执行条件Off时，Done位复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

在基本配置中指定了凸轮数据的起始地址，配置之后将生成数据块（功能暂时未实现），或指定（V 内存，符号表）起始地址。起始地址之后的一块内存存放 CAM 表数据。

当凸轮表选择多项式方式时，数据如下：

序号	名称	类型	注释
1		STRUCT	
2	xStart	REAL	X 轴坐标起点
3	xEnd	REAL	X 轴坐标终点
4	yStart	REAL	Y 轴坐标起点
5	yEnd	REAL	Y 轴坐标终点
6	nElements	WORD	元素个数
7	dX0	REAL	第0个点 X 坐标
8	dY0	REAL	第0个点 Y 坐标
9	dV0	REAL	第0个点速度
10	dA0	REAL	第0个点加速度
11	dX1	REAL	第1个点 X 坐标
12	dY1	REAL	第1个点 Y 坐标
13	dV1	REAL	第1个点速度
14	dA1	REAL	第1个点加速度
15	dX2	REAL	第2个点 X 坐标
	...	...	...
		END_STRUCT	

用户可以通过修改此内存里数据，从而改变凸轮表数据，修改数据后，调用 MC_CamTableSelect,通知 PLC 凸轮表数据改变，如果此时系统处于循环凸轮表运行之中，改变后的数据将在下一个凸轮周期生效，需要立即生效则在调用 MC_CamTableSelect 后，立即调用 MC_CamIn 指令。

凸轮偏移 Offset 和缩放比例 Scaling：

主轴输入转换的位置是根据以下公式进行的，并且使用转化后的 X 作为作为凸轮的输出：

计算公式：**X=MasterScaling*MasterPosition+MasterOffset**

因此，如果主轴的比例大于 1，所述凸轮将会运行在一个更高的速率，如果比例值小于 1，

速率将会随之降低。

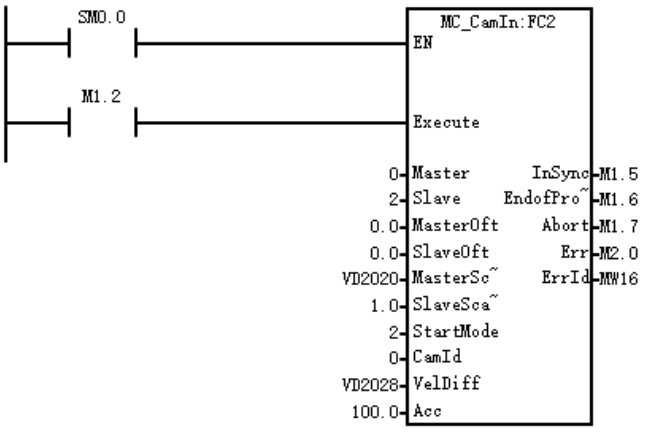
从轴 SlaveOffset，SlaveScaling：

计算公式：**Y=SlaveScaling*CAM(X)+SlaveOffset**

如果 SlaveScaling>1 导致凸轮效果的拉伸，从轴的范围将会增加；如果 SlaveScaling<1 将会导致一个收缩。

MC_CamIn（电子凸轮关联指令）

函数名：MC_CamIn



功能：此指令用于建立主从轴间的凸轮关系，建立凸轮关系时，可根据应用需求指定主从轴的偏移值，缩放比和启动模式。当指令执行完毕，从轴根据凸轮曲线跟随主轴运动。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。

Master	IN	主轴ID号	BYTE	
Slave	IN	从轴ID号	BYTE	
MasterOffset	IN	主轴偏移	REAL	主轴凸轮位置偏移。单位：单元
SlaveOffset	IN	从轴偏移	REAL	从轴凸轮位置偏移。单位：单元
MasterScaling	IN	主轴缩放	REAL	主轴缩放配置参数。此参数不为0
SlaveScaling	IN	从轴缩放	REAL	从轴缩放配置参数。此参数不为0
StartMode	IN	启动模式	BYTE	0：正向立即跳变 1：加速（走最短距离）跳变 2：斜坡正转 3：斜坡反转 注：正转反转只是针对从轴是旋转轴的情况，直线轴StartMode为0，1时为跳变，2，3为斜坡启动。 该参数的四种模式只是针对从站来说的，从轴是绝对还是相对模式由MC_CamTableSelect指令配置。
CamId	IN	凸轮表的ID号	BYTE	
VelDiff	IN	凸轮表耦合速度	REAL	
Acc	IN	凸轮表加速度	REAL	
InSync	OUT	凸轮关系建立	BOOL	主轴和从轴建立关系后，InSync被置位， 当指令的执行条件Off 时，InSync位被复位。
EndOfProfile	OUT	凸轮关系结束	BOOL	如果MC_CamTableSelect指令执行时，Perioc参数为0，当凸轮曲线执行完成一次后，EndOfProfile被置位。 Perioc参数为1，当凸轮曲线执行完成一次后，EndOfProfile被置位，一周期后复位。 当指令执行条件Off时，EndOfProfile位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码

凸轮偏移 Offset 和缩放比例 Scaling：

主轴输入转换的位置是根据以下公式进行的，并且使用转化后的 X 作为作为凸轮的输出：

X=MasterScaling*MasterPosition+MasterOffset

因此，如果主轴的比例大于 1，所述凸轮将会运行在一个更高的速率，如果比例值小于 1，速率将会随之降低。

从轴 SlaveOffset,SlaveScaling：

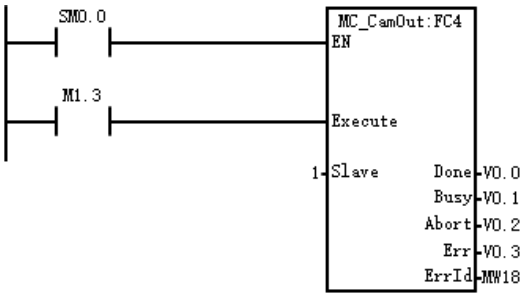
计算公式：

Y=SlaveScaling*CAM(X)+SlaveOffset

如果 SlaveScaling>1 导致凸轮效果的拉伸，从轴的范围将会增加；如果 SlaveScaling<1 将会导致一个收缩。

MC_CamOut（解除电子凸轮关联指令）

函数名：MC_CamOut



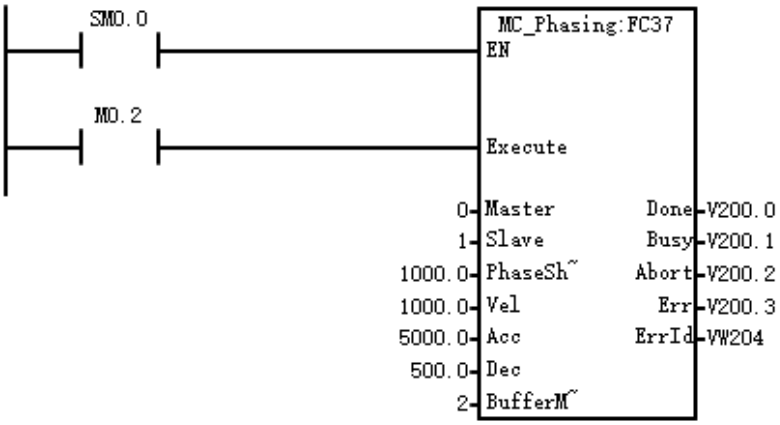
功能：此指令用于解除主从轴间的凸轮关系，关系解除后，从轴会以脱离前的速度继续运动。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Slave	IN	从轴ID号	BYTE	
Done	OUT	完成位	BOOL	当取消凸轮关联指令执行成功时，Done位被置位。 当指令执行条件Off时，Done位复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位；
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

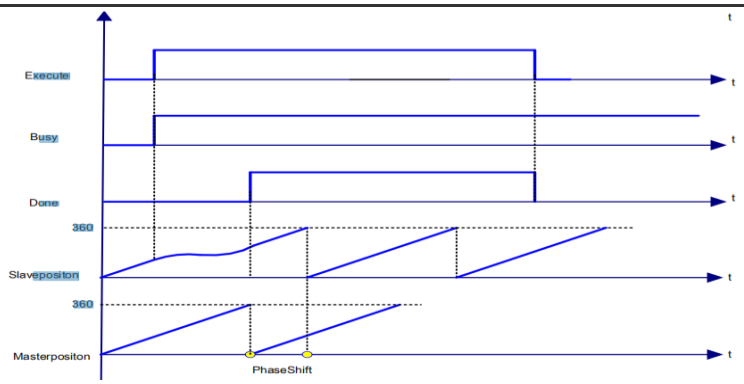
MC_Phasing（相位偏移指令）

函数名：MC_Phasing



功能：此指令用于调整主轴和从轴的相位差，当指令完调用完成之后，主轴和从轴之间相位差为PhaseShift，即从轴位置=主轴位置-PhaseShift。

如下图所示，主从轴都按 360 周期运动，Execute 信号上升沿执行调整，调整完成后从轴与主轴之间相位偏差为 PhaseShift 设定的值。

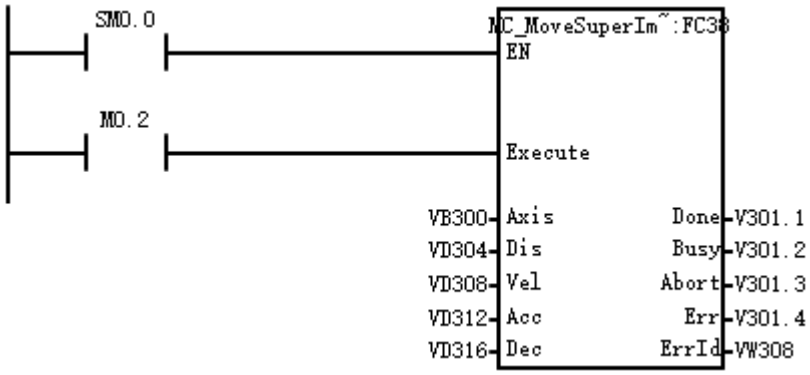


参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
Master	IN	主轴ID号	BYTE	
Slave	IN	从轴ID号	BYTE	
PhaseShift	IN	相位偏移	REAL	相位偏移
Vel	IN	速度	REAL	相位调整速度。
Acc	IN	加速度	BOOL	相位调整加速度。
Dec	IN	减速度	BOOL	相位调整减速度。
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Done	OUT	相位偏移调整完成	BOOL	相位调整完成Done被置位。 当指令的执行条件Off 时，Done位被复位
Busy	OUT	指令处理中	BOOL	True，指令正在处理中
Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveSuperImposed（叠加相对位移指令）

函数名：MC_MoveSuperImposed



功能：此指令用于控制终端执行机构在当前运动状态下按给定速度，加减速独立的追加一段给定距离，此指令执行时，不会终止前一个指令执行，两条指令同时执行，距离、速度、加减速会实时叠加。当其中一个指令执行完毕，将不再叠加其速度、加减速，另一指令仍然独立运行。

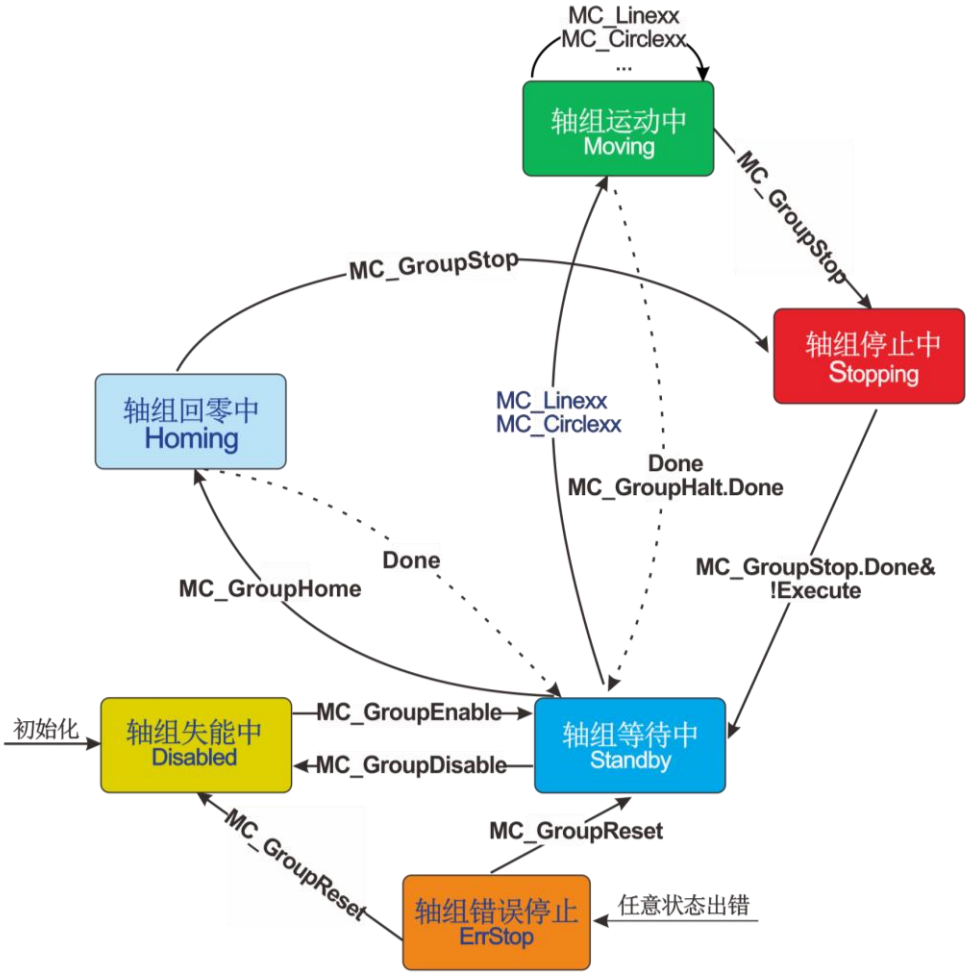
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行	BOOL	当执行条件由Off 变On时，执行该指令。

		条件		
Axis	IN	轴ID号	BYTE	
Dis	IN	距离	REAL	以当前位置为参考，终端执行机构要移动的目标距离，该值设置为负数代表轴将会反转。 单位：单元。
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见4.3.5 错误代码

4.2.3 轴组指令

轴组的运动控制指令，一般用于多轴的直线插补、圆弧插补指令等。轴组指令状态流程图如下图所示：



提示

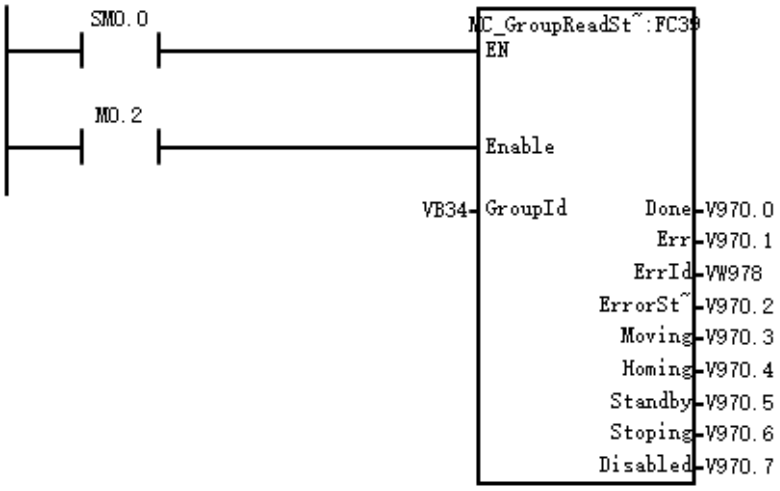
图中 MC_GroupHome，GroupStop 等函数还没有实现。因此 Homing 和 Stopping 两种轴组状暂时保留。

轴组指令常用到的 MC 功能块如下：

函数名	指令名称	适用的 PLC 型号	
		CTMC 系列 M228ML、M228SL	300-H 系列 H31-00、H32-01、H35-00、H36-01、H52-10、H56-10
MC_GroupReadState	读取轴组状态指令	支持	支持
MC_GroupEnable	轴组有效指令	支持	支持
MC_GroupDisable	轴组无效指令	支持	支持
MC_GroupReset	轴组错误复位指令	支持	支持
MC_GroupHalt	轴组停止指令	支持	支持
MC_MoveLinerRelative	相对位移直线插补指令	支持	支持
MC_MoveLinerAbsolute	绝对位移直线插补指令	支持	支持
MC_MoveCircularRalative	相对位移圆弧插补指令	支持	支持
MC_MoveCircularAbolute	绝对位移圆弧插补指令	支持	支持
MC_Helical	螺旋插补指令	不支持	支持

MC_GroupReadState（读取轴组状态指令）

函数名：MC_GroupReadState



功能：此指令用于读取相应的轴组状态参数的值。

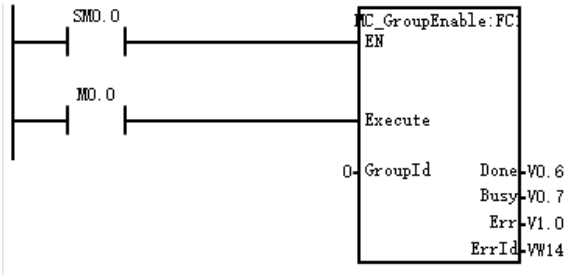
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Enable	IN	指令执行条件	BOOL	当为On时，执行该指令。
GroupId	IN	轴组Id号	BYTE	轴组号
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位；
Err	OUT	错误位	BOOL	如果检测到有错误，Err位被置位； 当指令的执行条件由On变Off时，Err位被复位。
ErrId	OUT	错误码	WORD	当Err为TRUE时，ErrId为指令的错误。 当ErrStop为TRUE时，ErrId为轴组当前错误。
ErrStop	OUT	状态指示位	BOOL	TRUE，如果轴组位于状态ErrStop
Stopping	OUT	状态指示位	BOOL	TRUE，如果轴组位于状态Stopping
Moving	OUT	状态指示位	BOOL	TRUE，如果轴组位于状态Moving
Standby	OUT	状态指示位	BOOL	TRUE，如果轴组位于状态Standby
Homing	OUT	状态指示位	BOOL	TRUE，如果轴组位于状态Homing

Disabled	OUT	状态指示位	BOOL	TRUE，如果轴组位于状态Disabled

MC_GroupEnable（轴组有效指令）

函数名：MC_GroupEnable



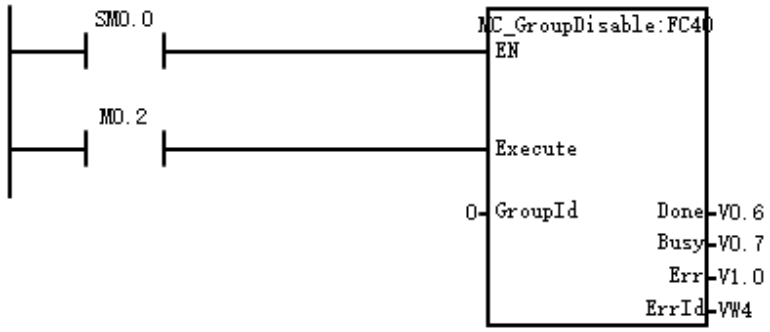
功能：使轴组有效。轴组状态从 GroupDisalbed 转变为 GroupStandby。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
GroupId	IN	轴组Id号	BYTE	轴组号
Busy	OUT	忙位	BOOL	指令正在执行为TRUE，执行完成为FLASE
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位；
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_GroupDisable（轴组无效指令）

函数名：MC_GroupDisable



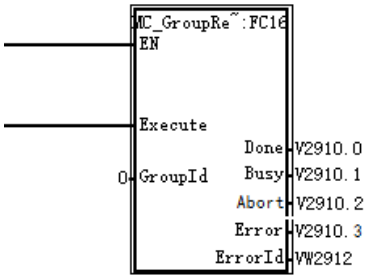
功能：使轴组无效。将轴组状态由 GroupStandby 转变为 GroupDisalbed。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
GroupId	IN	轴组ID号	BYTE	轴组号
Done	OUT	状态位	BOOL	指令执行成功后，Done 位被置位；
Busy	OUT	忙位	BOOL	指令正在执行为TRUE，执行完成为FLASE
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_GroupReset（轴组错误复位指令）

函数名：MC_GroupReset



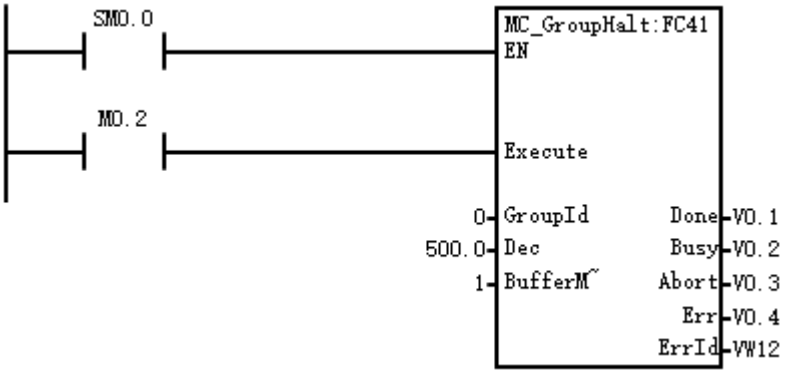
功能：轴组处于 ErrStop 状态时，调用本指令，清除轴组和轴的错误，并将轴组状态切换到 Disabled（出错前轴组未使能）或 Standby。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
GroupId	IN	轴组Id号	BYTE	轴组号
Done	IN	指令完成位	BOOL	完成位
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Abort	OUT	命令终止位	BOOL	保留接口，不会出现
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_GroupHalt（轴组停止指令）

函数名：MC_GroupHalt



功能：使插补动作中的所有轴减速停止。

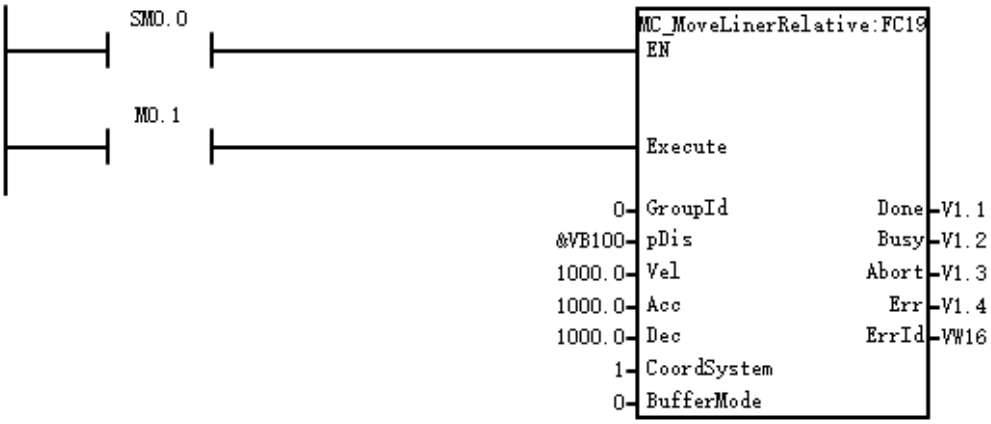
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
GroupId	IN	轴组Id号	BYTE	轴组号
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。（单位：单元/秒），当设为0时，实现紧急停止
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令) 其他：非法
Done	OUT	完成位	BOOL	指令执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Abort	OUT	中止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。

				当指令的执行条件Off 时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveLinerRelative（相对位移直线插补指令）

函数名：MC_MoveLinerRelative



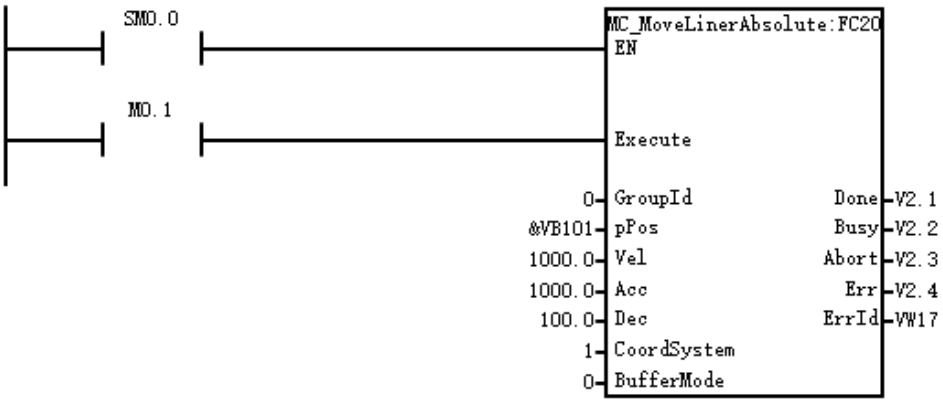
功能：此指令按照相对的位移进行直线插补

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
GroupId	IN	轴组ID号	BYTE	
pDis	IN	相对位置指针	DWORD	指向描述目标距离的指针。 目标距离为描述为6个REAL类型的数据。代表（MCS下的空间位移） X：REAL; Y：REAL; Z：REAL; A：REAL; B：REAL: C：REAL, 或（ACS下的每个轴的位移） A0：REAL A1：REAL; A2：REAL; A3：REAL: A4：REAL; A5：REAL; （单位：单元）
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令，禁用过渡) 1：Buffered(前一段减速完成开始执行缓冲的功能块) 2：BendingPrevious(以当前速度走到前一段结束并按照前一段的速率开始执行下一段，禁用过渡)

				16#12：TMCorner（附加角过渡，在前一段开始执行减速时以附加角过渡到下一段）；详情见BufferMode连续插补具体介绍。
CoordSystem	IN	坐标系统	BYTE	0：ACS 1：MCS (暂时只有这种) 2：WCS 3：PCS_1 4：PCS_2
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveLinerAbsolute（绝对位移直线插补指令）



函数名：MC_MoveLinerAbsolute

功能：此指令按照绝对的位移进行直线插补

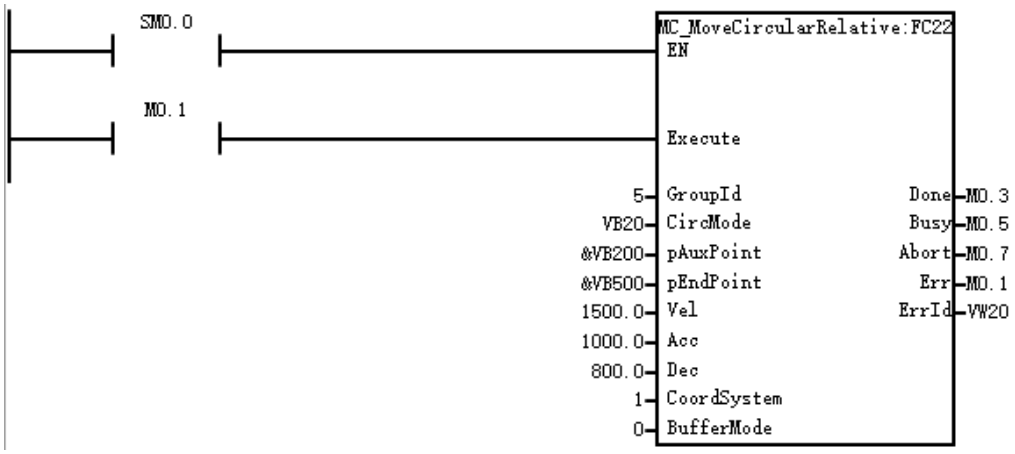
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
GroupId	IN	轴组ID号	BYTE	
pPos	IN	绝对位置指针	DWORD	指向描述目标位置的指针。 目标位置为描述为6个REAL类型的数据。代表（MCS下的空间位移） X：REAL; Y：REAL; Z：REAL; A：REAL;（保留） B：REAL: (保留) C：REAL,（保留） 或（ACS下的每个轴的位移） A0：REAL A1：REAL; A2：REAL; (对两轴保留) A3：REAL:（保留）

				A4：REAL;（保留） A5：REAL:（保留） （单位：单元）
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。（单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令，禁用过渡) 1：Buffered(前一段减速完成开始执行缓冲的功能块) 2: BendingPrevious(以当前速度走到前一段结束并按照前一段的速率开始执行下一段，禁用过渡) 16#12：TMCorner（附加角过渡，在前一段开始执行减速时以附加角过渡到下一段）；详情见BufferMode连续插补具体介绍。
CoordSystem	IN	坐标系统	BYTE	0：ACS 1：MCS (暂时只有这种) 2：WCS 3：PCS_1 4：PCS_2
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveCircularRalative（相对位移圆弧插补指令）

函数名：MC_MoveCircularRalative



功能：此指令按照相对的位移进行圆弧插补。

参数说明：

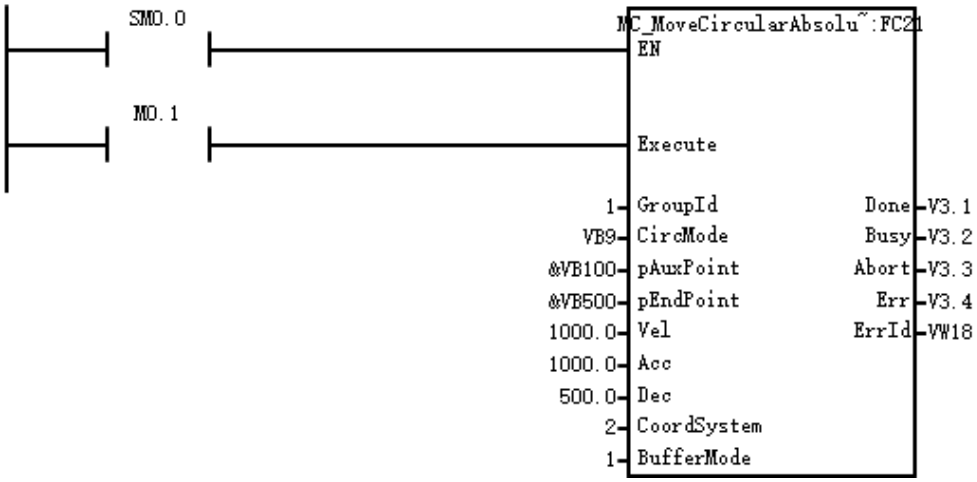
参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。

GroupId	IN	轴组ID号	BYTE	
CircMode	IN	圆弧插补方式	BYTE	Bit0~Bit3 0：三点式 1：中心点式。 8：三点式全圆 9：中心点式全圆 Bit4：插补方向（中心点式和全圆有效） 0：为反转(CCW) 1：正转(CW) 圆弧的方向是根据右手定则选择的：CCW方向是沿着手指弯曲的方向。 当为中心点式时，如果中心点与起点和终点的距离不完全相同（这通常是由于编程中心点的精度有限），则将其投影到起点和终点的垂直平分线上。一旦中心点离得太远（超过半径的1%），就会返回一个错误。
pAuxPoint	IN	参考位置指针	DWORD	指向描述参考位置的指针。 三点式时，pAuxPostion指的是经过的参考点。 当为中心点式时，pAuxPostion指的是圆心。 参考位置为描述为6个REAL类型的数据。代表（MCS下的空间位移） X：REAL Y：REAL Z：REAL A：REAL（保留） B：REAL（保留) C：REAL（保留） （单位：单元）
pEndPoint	IN	目标位置指针	DWORD	指向描述目标位置的指针。 目标位置为描述为6个REAL类型的数据。代表（MCS下的空间位移） X：REAL Y：REAL Z：REAL A：REAL（保留） B：REAL（保留） C：REAL（保留）
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令，禁用过渡) 1：Buffered(前一段减速完成开始执行缓冲的功能块) 2：BendingPrevious(以当前速度走到前一段结束并按照前一段的速率开始执行下一段，禁用过渡) 16#12：TMCorner（附加角过渡，在前一段开始执行减速时以附加角过渡到下一段）；详情见BufferMode连续插补具体介绍。
CoordSystem	IN	坐标系统	BYTE	0：ACS 1：MCS (暂时只有这种) 2：WCS 3：PCS_1 4：PCS_2
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。

Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_MoveCircularAbolute（绝对位移圆弧插补指令）

函数名: MC_MoveCircularAbolute



功能：此指令按照绝对的位移进行圆弧插补。

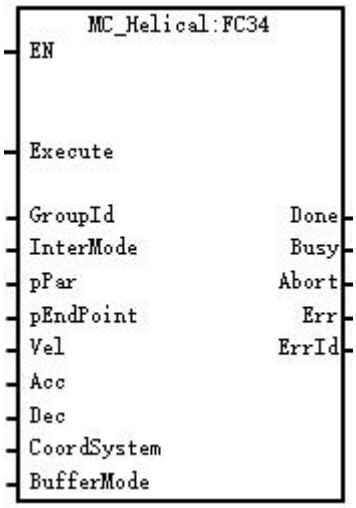
参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off 变On时，执行该指令。
GroupId	IN	轴组ID号	BYTE	
CircMode	IN	圆弧插补方式	BYTE	Bit0-Bit3 0：三点式 1：中心点式。 8：三点式全圆 9：中心点式全圆 Bit4: 插补方向（中心点式有效） 0：为反转(CCW) 1：正转(CW) 圆弧的方向是根据右手定则选择的：CCW方向是沿着手指弯曲的方向。 当为中心点式时， 如果中心点与起点和终点的距离不完全相同（这通常是由于编程中心点的精度有限），则将其投影到起点和终点的垂直平分线上。一旦中心点离得太远（超过半径的1%），就会返回一个错误。

pAuxPoint	IN	参考位置指针	DWORD	指向描述参考位置的指针。 当三点式时，pAuxPostion指的是经过的参考点。 当为中心点式时，pAuxPostion指的是圆心。 参考位置距离为描述为6个REAL类型的数据。代表（MCS，PCS，WCS下的） X：REAL Y：REAL Z：REAL A：REAL（保留） B：REAL（保留） C：REAL（保留） （单位：单元）
pEndPoint	IN	结束位置指针	DWORD	指向描述结束位置的指针。 结束位置距离为描述为6个REAL类型的数据。 X：REAL Y：REAL Z：REAL A：REAL（保留） B：REAL（保留） C：REAL（保留）（单位：单元）
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令，禁用过渡) 1：Buffered(前一段减速完成开始执行缓冲的功能块) 2：BendingPrevious(以当前速度走到前一段结束并按照前一段的速率开始执行下一段，禁用过渡) 16#12：TMCorner（附加角过渡，在前一段开始执行减速时以附加角过渡到下一段）；详情见 BufferMode 连续插补具体介绍。
CoordSystem	IN	坐标系统	BYTE	0：ACS；1：MCS (暂时只有这种)；2：WCS 3：PCS_1；4：PCS_2
Done	OUT	完成位	BOOL	绝对位移动作执行完成时，Done位被置位； 当指令的执行条件Off 时，Done位被复位。
Abort	OUT	命令终止位	BOOL	当该指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off 时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off 时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

MC_Helical（螺旋插补指令）

函数名：MC_Helical

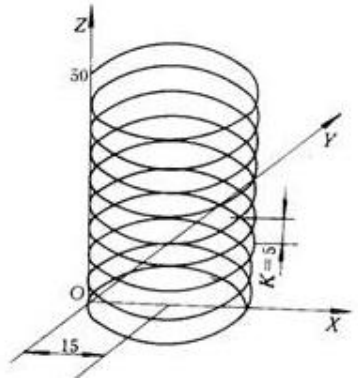


功能：此指令按照螺旋线的方式进行插补。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	当执行条件由Off变On时，执行该指令。
GroupId	IN	轴组ID号	BYTE	
InterMode	IN	圆弧插补方式	BYTE	0：xy平面逆圆，相对位置，忽略终点的XY坐标； 1：xy平面顺圆，相对位置，忽略终点的XY坐标； 0x10：xy平面逆圆，相对位置，终点的XY坐标不在螺旋线上时报错； 0x11：xy平面顺圆，相对位置，终点的XY坐标不在螺旋线上时报错；
pPar	IN	参数指针	DWORD	指向描述参数的指针 pPar指向的是螺旋线插补的参数，描述为6个REAL类型 0：圆心在圆弧平面的横坐标 1：圆心在圆弧平面的纵坐标 2：螺旋线的导程。螺旋线绕圆柱体转一圈，沿圆柱体轴线方向移动的距离。 3、4、5：预留 （单位：单元）
pEndPoint	IN	目标位置指针	DWORD	指向描述目标位置的指针。 目标位置为描述为6个REAL类型的数据。代表（MCS下的空间位移（相对）） X：REAL Y：REAL Z：REAL A：REAL（保留） B：REAL（保留） C：REAL（保留）
Vel	IN	速度	REAL	终端执行机构的运转速度，此参数总为正。 （单位：单元/秒）
Acc	IN	加速度	REAL	终端执行机构的加速度，此参数总为正。 （单位：单元/秒/秒）
Dec	IN	减速度	REAL	终端执行机构的减速度，此参数总为正。 （单位：单元/秒/秒）
BufferMode	IN	中止模式	BYTE	0：Abort (直接打断正在进行的指令，禁用过渡) 1：Buffered(前一段减速完成开始执行缓冲的功

				能块) 2：BendingPrevious(以当前速度走到前一段结束并按照前一段的速率开始执行下一段，禁用过渡) 16#12：TMCorner（附加角过渡，在前一段开始执行减速时以附加角过渡到下一段）
CoordSystem	IN	坐标系统	BYTE	0：ACS 1：MCS (暂时只有这种) 2：WCS 3：PCS_1 4：PCS_2
Done	OUT	完成位	BOOL	动作执行完成时，Done位被置位； 当指令的执行条件Off时，Done位被复位。
Abort	OUT	命令终止位	BOOL	当指令执行过程中被终止时，Abort位被置位； 当指令的执行条件Off 时，Abort位被复位。
Busy	OUT	指令执行位	BOOL	当指令执行时。Busy位被置位。 当指令完成Busy复位。 当指令的执行条件Off时，Busy位被复位。
Err	OUT	错误位	BOOL	如果检测到有错误，Error位被置位； 当指令的执行条件Off时，Error位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

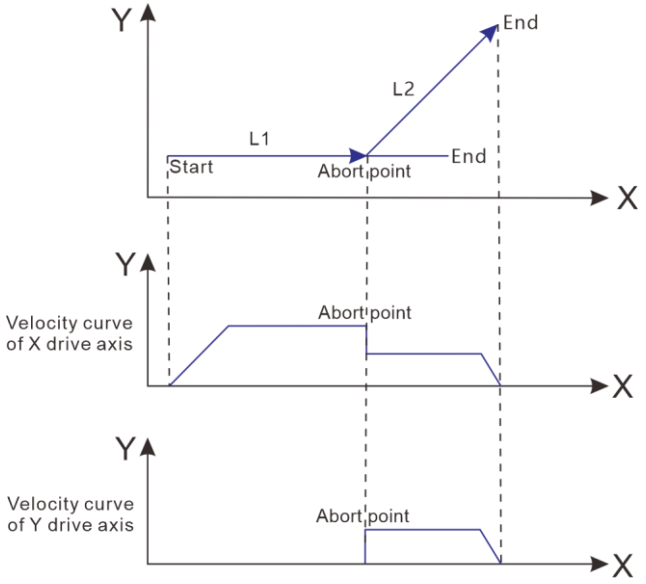


BufferMode 连续插补具体介绍

取值	描述	详细
0	Abort	直接打断正在进行的指令，禁用过渡
1	Buffered	前一段减速完成，开始执行缓冲的指令
2	BendingPrevious	以前一段速度走到前一段结束并按照前一段的速率开始执行下一段，禁用过渡
16#12	TMCorner	附加角过渡，在前一段开始执行减速时以附加角过渡到下一段

Abort

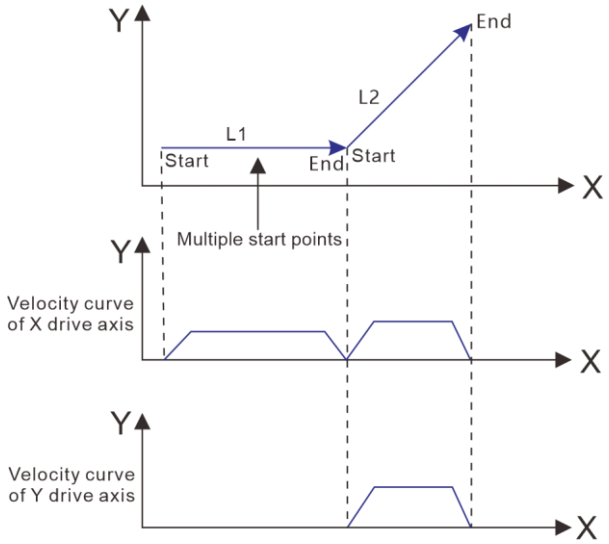
直线 L1 先执行第一条插补指令，在 L1 走完之前触发第二条插补指令，如果第二条插补指令的缓冲模式设置为“Abort“，则第二条插补指令立即打断第一条插补指令并开始执行新的插补曲线。



Buffered

直线 L1 首先执行第一条插补指令，在 L1 走完之前触发第二条插补指令，如果第二条插补指令的缓冲模式设置为“Buffered”，插补器将继续执行第一条插补指令。

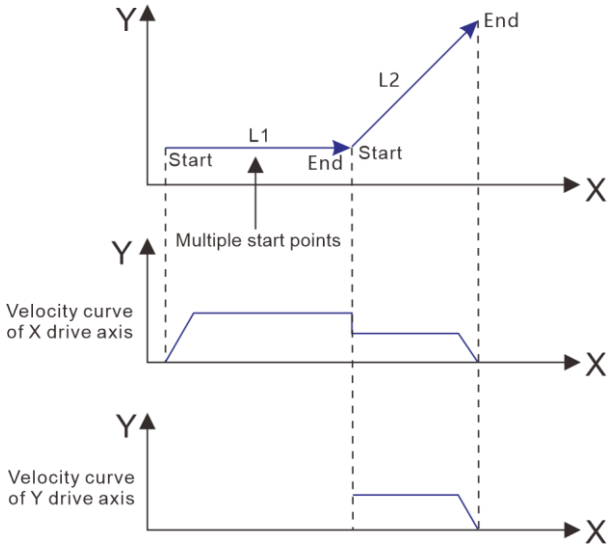
等第一条插补指令执行完，Done 信号输出有效后，第二条插补指令开始执行，如下图所示：



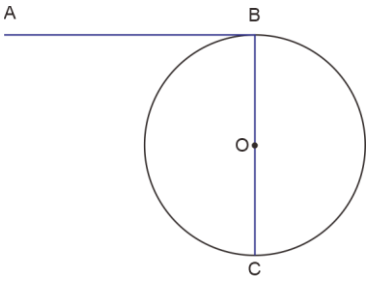
BendingPrevious

直线 L1 先执行第一条插补指令，在 L1 走完之前触发第二条插补指令。如果第二条插补指令的缓冲模式设置为“BendingPrevious”，插补器将尝试保持第一条指令的目标速度走完整条直线。

等待第一条插补指令执行完，Done 信号输出有效后，第二条插补指令开始执行，切换点将保持速率不变，坐标轴的分速度将重新分配。如下图所示：

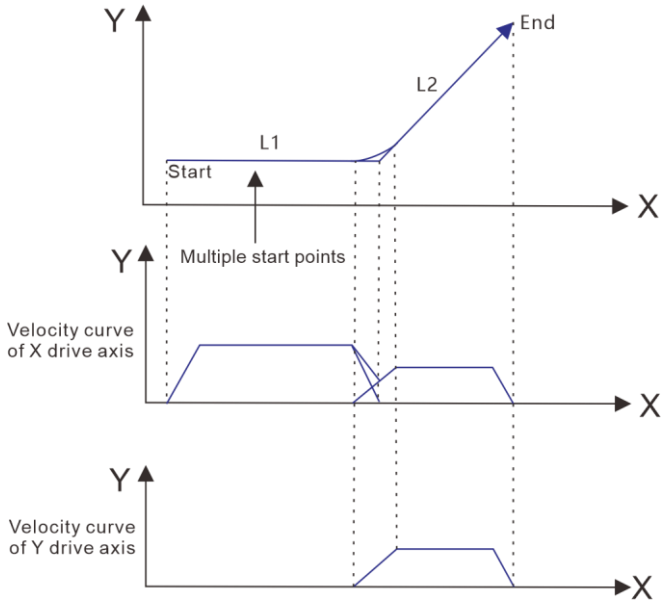


该模式适合在直线与圆弧相互切换且直线位于圆弧的切线上，采用该模式可以保持插补曲线的速度连续。



TMCorner

直线 L1 首先执行第一条插补指令，在 L1 走完之前触发第二条插补指令。如果第二条插补指令的缓冲模式设置为“TMCorner”，插补器检测到第一条直线 L1 开始执行减速时，启动第二条插补指令，以附加角的方式过渡到第二条插补，保持各方向上速度连续，如下图所示：



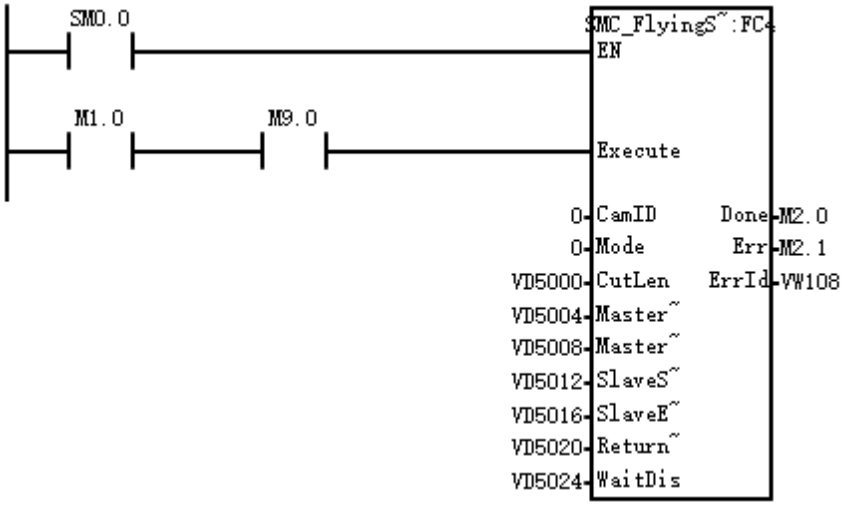
4.2.4 追飞剪指令

追飞剪指令适用于 M228SL，追飞剪包含四个功能块，如下表所示：

函数名	指令名称	备注
SMC_FlyingShearSet	追剪设置指令	仅适用于 CTMC 系列 M228ML、M228SL
SMC_RotatingCutset	飞剪设置指令	
SMC_GetPosByEvent	位置抓取指令	
SMC_PosToPulse	位置转换指令	

SMC_FlyingShearSet（追剪设置指令）

函数名：SMC_FlyingShearSet



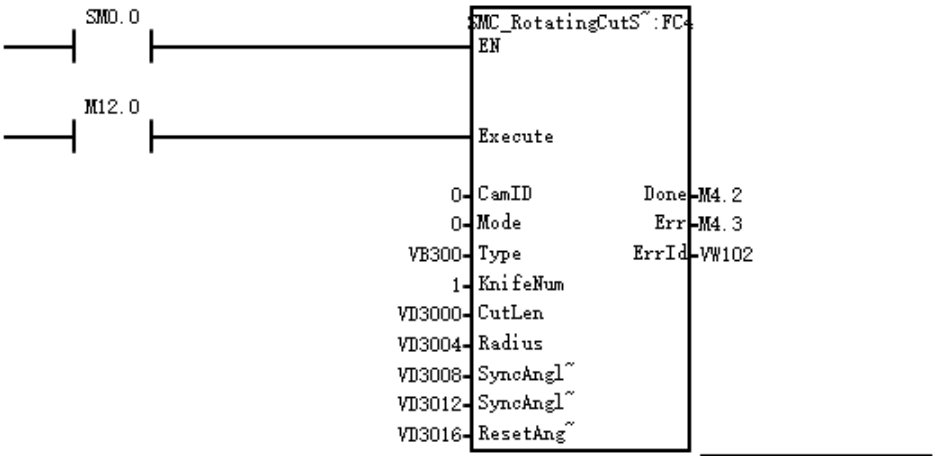
功能：该指令用于修改一个正在运行的追剪凸轮相关参数，调用之后，会重新计算凸轮曲线。重新计算好的曲线会在下一个凸轮周期开始时生效。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	上升沿执行
CamID	IN	追剪凸轮ID	BYTE	
Mode	IN	模式	BYTE	预留，默认参数为0
CutLen	IN	裁剪长度	REAL	定标模式下通过标的位置测定给出
MasterStartPos	IN	主轴起始位置	REAL	
MasterSyncPos	IN	主轴同步位置	REAL	
SlaveSyncPos	IN	从轴同步位置	REAL	
SlaveEndPos	IN	从轴同步结束位置	REAL	
ReturnDis	IN	从轴返回对应主轴的位移	REAL	
WaitDis	IN	从轴等待时主轴的位移	REAL	
Done	OUT	完成位	BOOL	
Err	OUT	错误位	BOOL	若检测到有错误，Error 位被置位；当指令的执行条件为 Off 时，Error 位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

SMC-RotatingCutSet（飞剪设置指令）

函数名：SMC_RotatingCutset



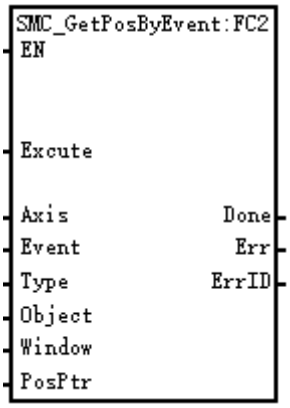
功能：该指令用于修改飞剪凸轮（旋转切割）用到的相关参数，调用之后，会重新计算凸轮曲线。重新计算好的曲线会在下一个凸轮周期开始时生效。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	上升沿执行
CamID	IN	凸轮编号	BYTE	飞剪凸轮ID
Mode	IN	模式	BYTE	预留参数，默认为0。
Type	IN	曲线类型	BYTE	0：工作曲线 1：启动曲线 2：停止曲线
KnifeNum	IN	切刀个数	BYTE	从轴上的切刀数目，切刀要求均匀角度安装，两两切刀之间的角度相同。
CutLen	IN	裁剪长度	REAL	定标模式下通过标的位置测定给出
Radius	IN	刀轮半径	REAL	
SyncAngelFront	IN	前同步角	REAL	
SyncAngelBack	IN	后同步角	REAL	
ResetAngel	IN	刀轮复位角	REAL	指复位之后，切刀从原点按运行方向运行，当前同步角边沿到达切点经过的角度值。
Done	OUT	完成位	BOOL	
Err	OUT	错误位	BOOL	若检测到有错误，Error 位被置位；当指令的执行条件为 Off 时，Error 位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

SMC_GetPosByEvent（位置抓取指令）

函数名：SMC_GetPosByEvent



功能：该指令绑定一个 IO 中断，当中断发生时读取某个轴的输出脉冲计数（HSC 或映射内存值），并按该轴的量纲配置，将这个脉冲值转换为坐标值，输出到 PosPtr 指针指定的内存中。

注意：

1、只需要调用一次即可绑定中断事件。

2、必须先调用 ATCH 指令（中断连接指令），再调用 ENI 指令（中断使能指令），该指令中断才能产生。

3、该指令功能块在 ATCH 指令绑定的 OB 块之前执行，速度快于 OB 块内读取的值。

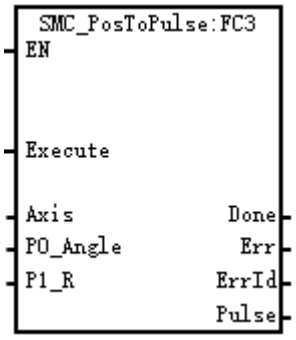
4、如果是 HSC 必须在该指令前初始化，才能读到变化的脉冲值。

参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	上升沿执行
Axis	IN	轴号	BYTE	
Event	IN	事件中断号	BYTE	仅支持本机IO中断
Type	IN	类型	BYTE	0：HSC，中断发生时，读取HSC当前值，如果是HSC捕获中断，将会读取HSC硬件捕获寄存器。 1：内存映射，中断发生时，读取映射的内存值。
Object	IN	读取对象	DWORD	如果Type参数是0，则该值表示的是HSC的ID，范围为0~5。 如果Type参数是1，则该值是指向内存的指针。
Window	IN	检测窗口	REAL	连续两次转换的最小间隔位置
PosPtr	IN	转换结果	REAL	转换结果存放的指针
Done	OUT	完成位	BOOL	0：指令未完成 1：指令完成
Err	OUT	错误位	BOOL	若检测到有错误，Error 位被置位；当指令的执行条件为 Off 时，Error 位被复位。
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。

SMC_PosToPulse（位置转换指令）

函数名：SMC_PosToPulse



功能：该指令将两个输入点按轴的量纲，转换为实际的脉冲数。参数说明：

参数名	输入输出属性	参数描述	类型	备注
Execute	IN	指令执行条件	BOOL	上升沿执行
Axis	IN	轴号	BYTE	需要用到量纲参数和类型
P0_Angle	IN	事件中断号	REAL	位置参数 0，如果是直线轴，则就是位置 0，如果是旋转轴，则是角度。
P1_R	IN	类型	REAL	位置参数1，如果是直线轴，就是位置1，如果是旋转轴，则是半径。
Done	OUT	完成位	BOOL	0：指令未完成 1：指令完成

Err	OUT	错误位	BOOL	
ErrId	OUT	错误代码	WORD	错误代码，详见章节4.3.5 错误代码。
Pulse	OUT	转换之后的脉冲个数	DWORD	

4.2.5 错误代码

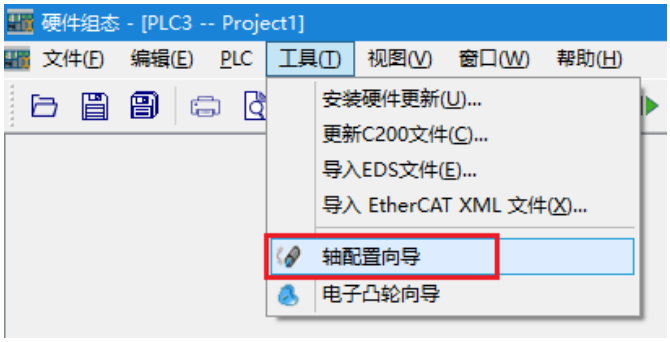
错误代码说明：

错误码	说明	错误码	说明
0	没有错误	53	运动过程中断使能
1	轴号取值错误	60	RetioNumerator 取值错误
2	轴状态错误	61	RetioMenominator 取值错误
3	指令丢失	65	凸轮表和 MC_TableSelect 不一致
4	轴忙	66	凸轮表 ID 错误（或没有创建）
5	速度取值太小（小于最小速度）	67	凸轮表数据错误
6	速度取值太大（大于最大速度）	68	MasterScaling 取值错误
7	加速度取值太小	69	SlaveScaling 取值错误
8	加速度取值太大	70	StartMode 取值错误
9	减速度取值太小	71	目标转矩（16#6071：0）没有映射
10	减速度取值太大	72	操作模式（16#6060：0）没有映射
11	方向取值错误	73	当前转矩（16#6077：0）没有映射
12	HSP 轴取值错误	74	SLOP 设定错误
13	HSP 模块号错误	75	ProbeID 设定错误
15	轴未准备好	76	Touch probe function（16#60B8：0）没有映射
16	复位错误失败	77	Touch probe staus（16#60B9：0）没有映射
17	GroupId 错误	78	Touch probe value(16#60BA:00 ~60BD)没有映射
19	距离指针错误	79	窗口设定错误
20	BufferMode 取值错误	80	达到左限位开关
21	CoordSys 取值错误	81	达到右限位开关
22	内部软件错误	82	到达负向软件限位
23	内存申请错误	83	到达正向软件限位
24	CircMode 设置错误	90	cmd 取值错误
25	圆弧三点成一线	91	TriggerInput 取值错误
26	两点到圆心距离不相等	92	TriggerVar 取值错误
27	interMode 设置错误	93	MC_Feed 执行失败
30	回原模式缺少 Z 相配置	94	MC_Feed 指令正在执行
31	回原模式缺少正向开关配置	95	模块中没有此中断，或中断没有使能
32	回原模式缺少负向开关配置	96	设定力矩错误
33	回原模式缺少原点开关配置	100	轴组中的轴错误
34	回原模式取值错误	101	轴组状态设置错误
35	伺服内部回原错误	102	轴组中有轴已处于别的轴组控制之中
43	Position 参数设置过大	110	超过最大缓冲指令条数
44	Position 参数设置过小	1000	此配置将来才会支持
45	主轴坐标超范围	1600	没有资源
46	运行超速（运行速度超过最大速度）	1601	事件中断 ID 不支持

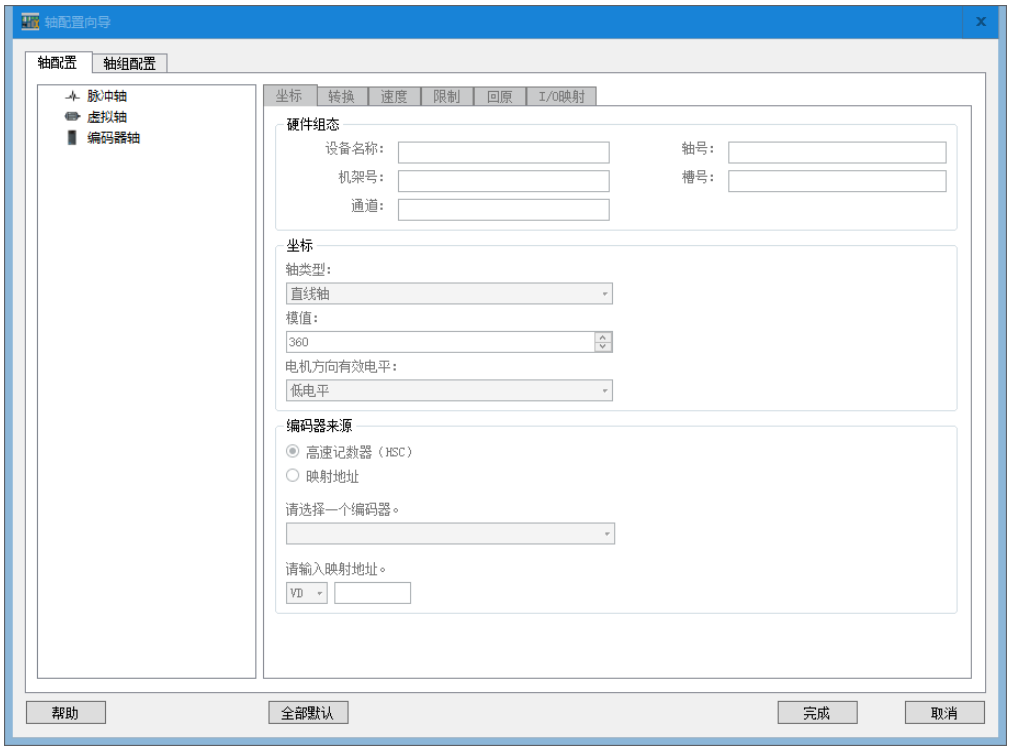
47	导程设置错误	1602	追剪参数错误
48	终点不在曲线上	1603	飞剪参数错误
49	圆心坐标设定错误	1604	飞剪角度设置错误
50	通讯错误	2001~2022	模块错误
51	伺服内部错误	3000	组态中存在错误
52	没有错误需要清除 MC_Reset		

4.2.6 轴配置向导

在硬件组态界面选择“工具”→“轴配置向导”即可进行轴配置，如下图所示。



轴配置向导界面如下所示：双击“脉冲轴”、“虚拟轴”或“编码器轴”即可添加所需的轴。



提示

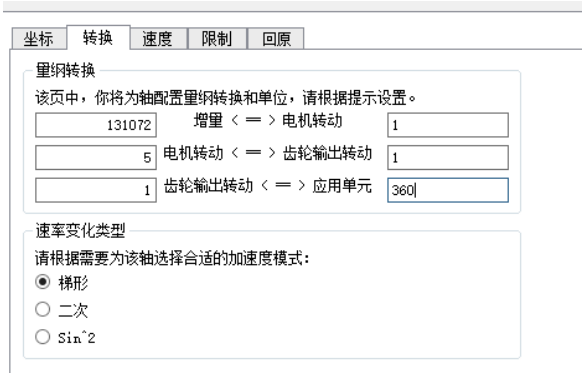
1、轴配置中所有轴所配置轴个数总和上限为 64，单类轴个数上限也为 64。

2、PLC 程序中可通过指令 MC_Readstatus 获取轴当前状态。

【坐标】

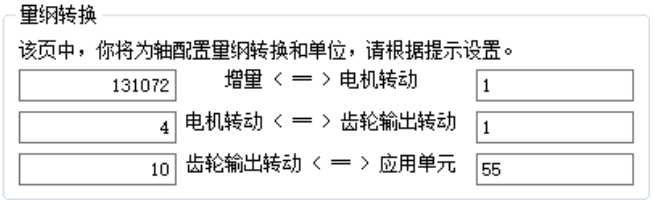
- 硬件组态：“设备名称”为添加该轴时的轴名称；“轴号”为该轴添加的顺序 ID，不可修改，随轴的添加依次递增，若删除某条轴，则新添加的轴轴号为上一次删除轴轴号；“机架号”为 0（CTMC 系列硬件组态），“槽号”为该轴对应槽号；“通道号”为该轴对应的通道号，范围 0~5。
- 坐标：“轴类型”可选择“直线轴”或“旋转轴”，对于丝杆类型的往复运行机构，其行程是有限，而需要知道其在丝杆行程范围内的绝对位置，此时选择“直线轴”较好；若是单方向运转类型的旋转轴，采用线性模式容易出现位置计数溢出，从而计算错误，故选择“旋转轴”较好。其余选项均为默认不可选；“用户坐标来源”固定可选“指令输出”；“电机方向有效电平”可选择“低电平”或“高电平”。

【转换】

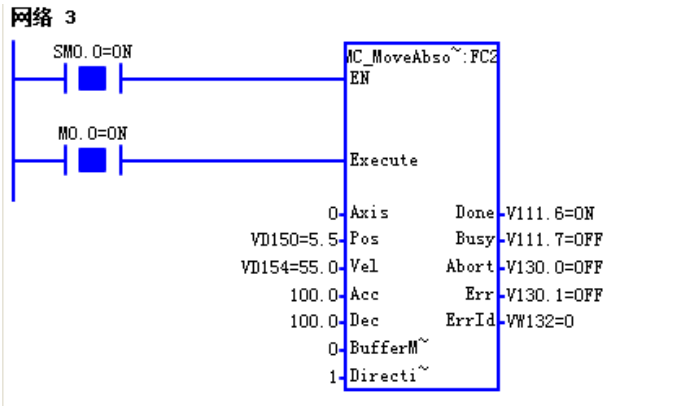


在“转换”界面可根据提示设置纲量转换和速率变化类型。将上图中的“应用单元”设为 360，当程序指令要求伺服运行 1 个单位时，齿轮端将会选择 1/360 圈（轴运动 1 度），伺服电机端会旋转 5*1/360 圈（电机旋转 5 度），以此类推。按照实际机械结构设定对应参数（即电子齿轮比）之后，就可按照应用系统的运动距离物理单位输入距离命令，使参数控制直观易懂。

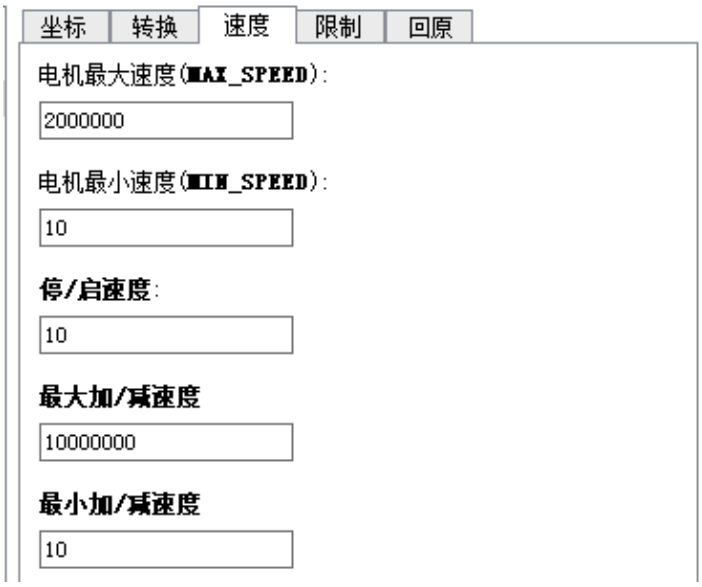
举例：伺服电机经过减速机 4:1 的机械减速后，驱动导程为 5.5mm 的丝杆，即丝杆转动一圈，丝杆滑块运动 5.5mm。量纲转换设置如下：



设置后的量纲即可作为 MC 控制指令的物理参数量纲，将上述设置下载到程序块中，编程使得工件运动到 5.5mm 坐标处，位置命名单位即为设备物理坐标单位，程序示例如下：



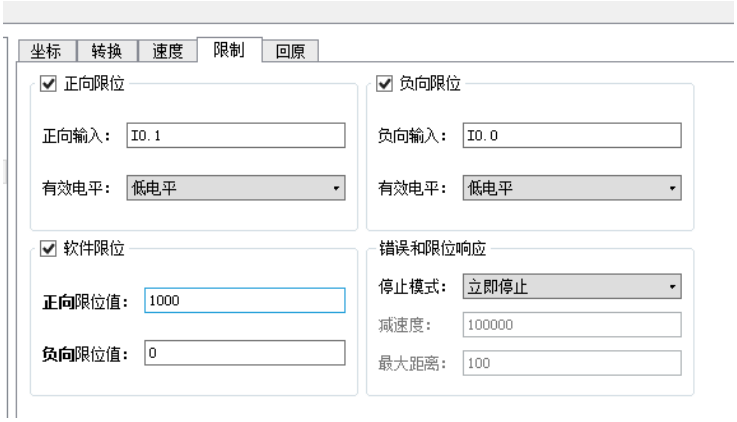
【速度】



在此页面可设置电机的最大/最小速度、停止/启动速度以及最大和最小加/减速度。程序运行过程中，若设定速度超过组态设置，库指令将报警。错误代码如下：

Errorcode	意义
5	速度取值太小（小于最小速度）
6	速度取值太大（大于最大速度）
7	加速度取值太小
8	加速度取值太大
9	减速度取值太小
10	减速度取值太大

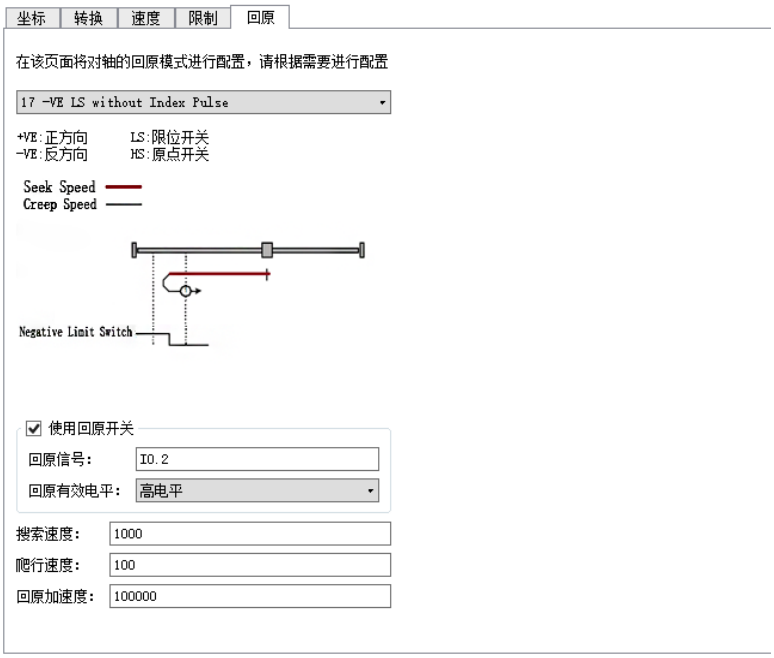
【限制】



勾选“正向限位”和“负向限位”可对正/负向输入进行修改设置，“有效电平”可选择“高电平”或“低电平”；勾选“软件限位”可对正/负向限位值进行修改设置；“错误和限位响应”的“停止模式”可选

“立即停止”或“减速停止”。

【回原】

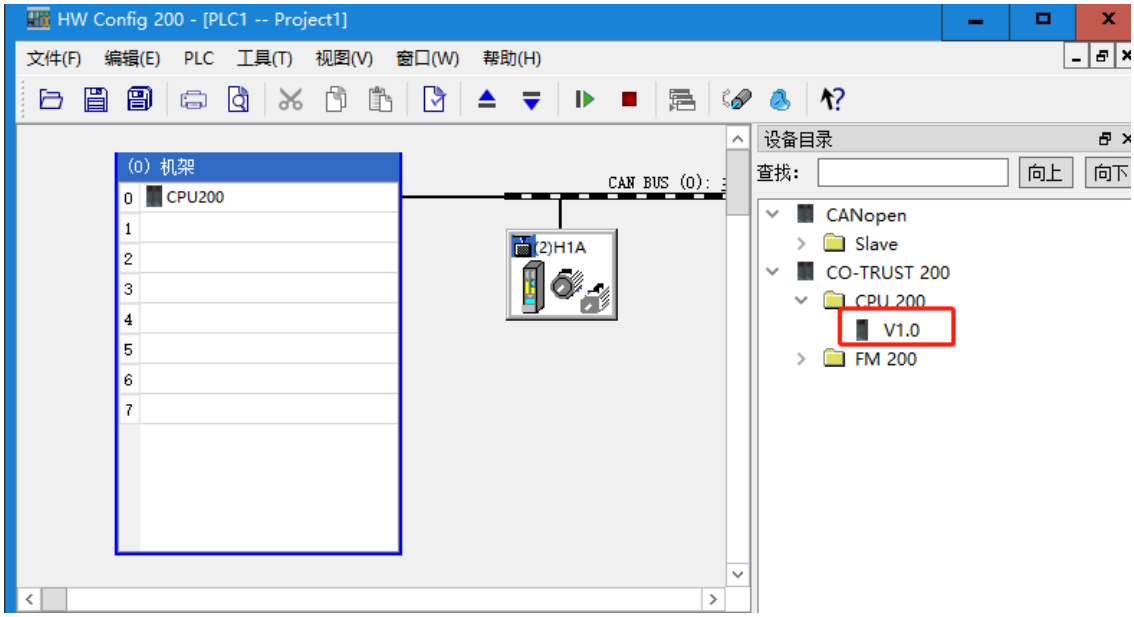


在此页面可对轴的回原模式进行配置，此配置仅脉冲轴支持，选择下拉选项可查看回原模式，共有 17 种回原模式可选。

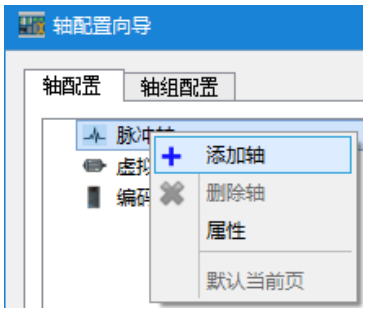
4.2.7 轴配置

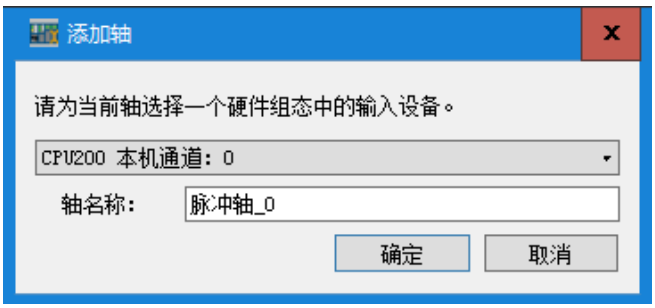
1、脉冲轴配置

配置脉冲轴前，需先在机架中添加“CPU200”，如下图所示：

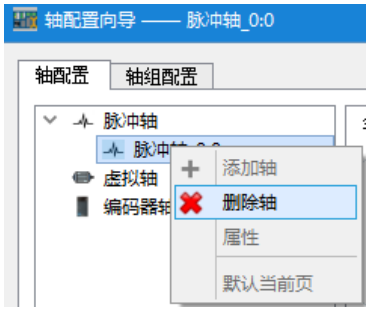


完成以上配置后在“工具”选项中选择“轴向导配置”，在轴向导配置界面鼠标右键点击脉冲轴即可选择添加轴，轴名称可自由更改，点击“确定”即成功添加轴，如下所示：

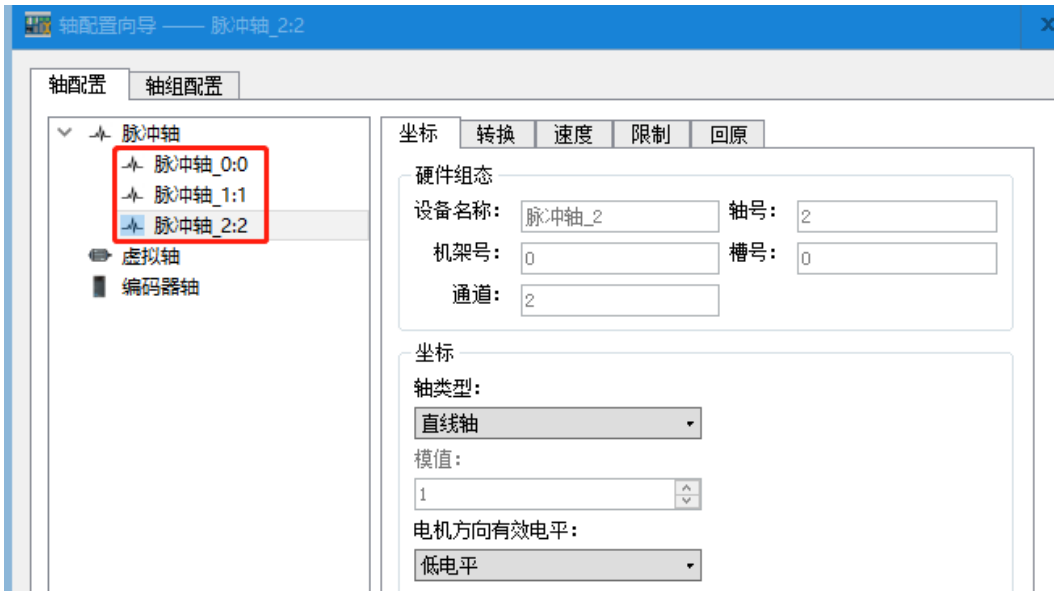




右键选中添加的轴，可选择删除轴，如下图：

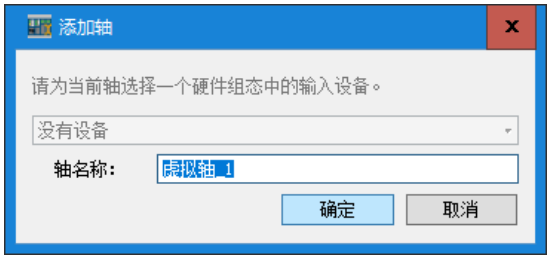


添加的脉冲轴界面如下所示：

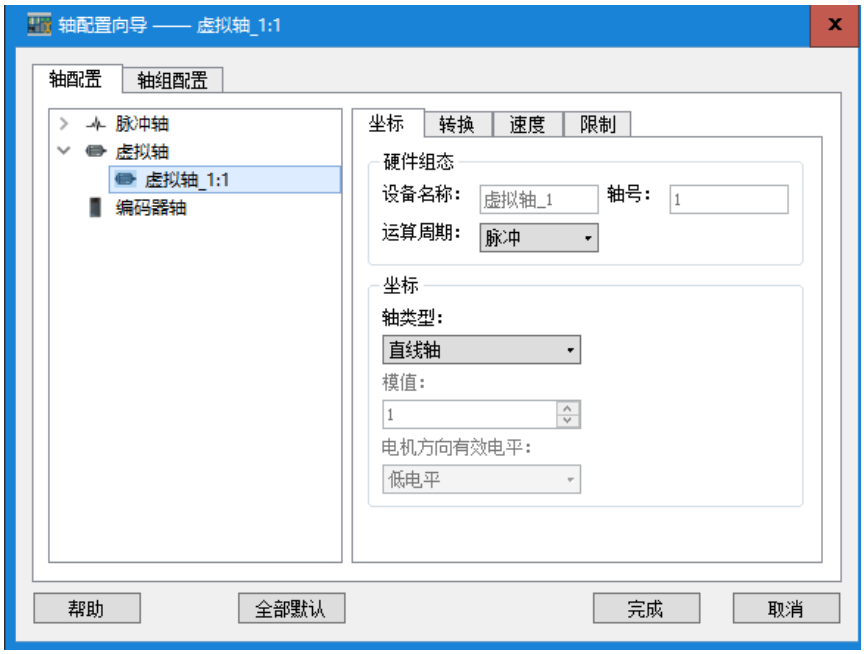


2、虚拟轴配置

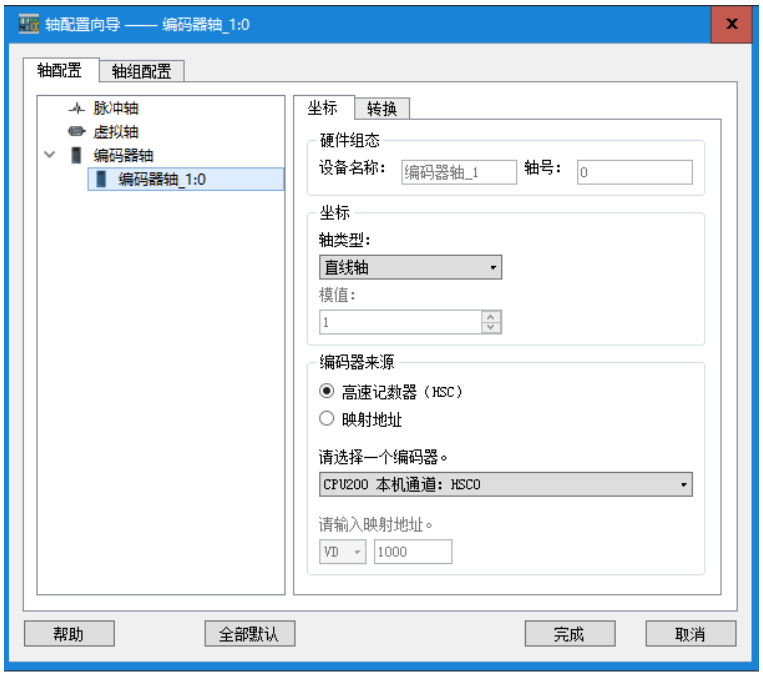
在轴配置向导界面，鼠标右键点击“虚拟轴”，选择添加轴，在弹出的以下界面可选择修改轴名称，点击“确定”完成虚拟轴添加。



添加的虚拟轴界面如下所示：



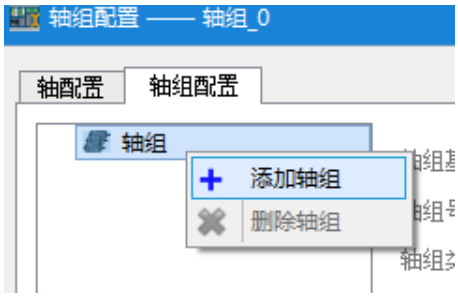
3、编码器轴配置



在上述界面选中“编码器轴”，右键可选择添加新的编码器轴。在“编码器来源”选项下为该轴选择对应编码器，可选择输入相应映射地址（默认 V 区）确定编码器。

4.2.8 轴组配置

轴组配置即将配置好的轴两个或三个一组以轴组的形式自由组合，在“轴组配置”界面，右键点击“添加轴组”进行轴组添加，如下图所示：



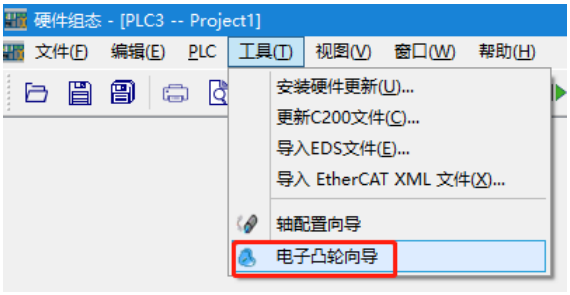
添加成功的轴组界面如下所示：

- “轴组基本配置”：“轴组号”不可更改，“轴组类型”下拉选项可选择“2-轴正交”或“3-轴正交”，分别对应可选两组或三组“构成轴”；“运算周期”可选“脉冲”；“加速模式”可选“梯形”，“sin*2”和“二次”。
- “逻辑轴”：默认六组，不可更改。
- “构成轴”：“2-轴正交”对应两个构成轴，“3-轴正交”对应三个构成轴，下拉选项只可选择“脉冲轴”或“虚拟轴”。

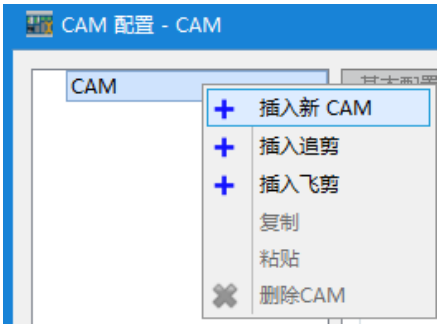
4.2.9 电子凸轮向导

在电子凸轮编辑器中可以通过图形或列表的方式实现电子凸轮盘(或者凸轮开关功能)。当根据相关的应用程序产生代码时，将创建各种全局数据结构(CAM 数据)，这些数据结构能够被 IEC 程序访问。

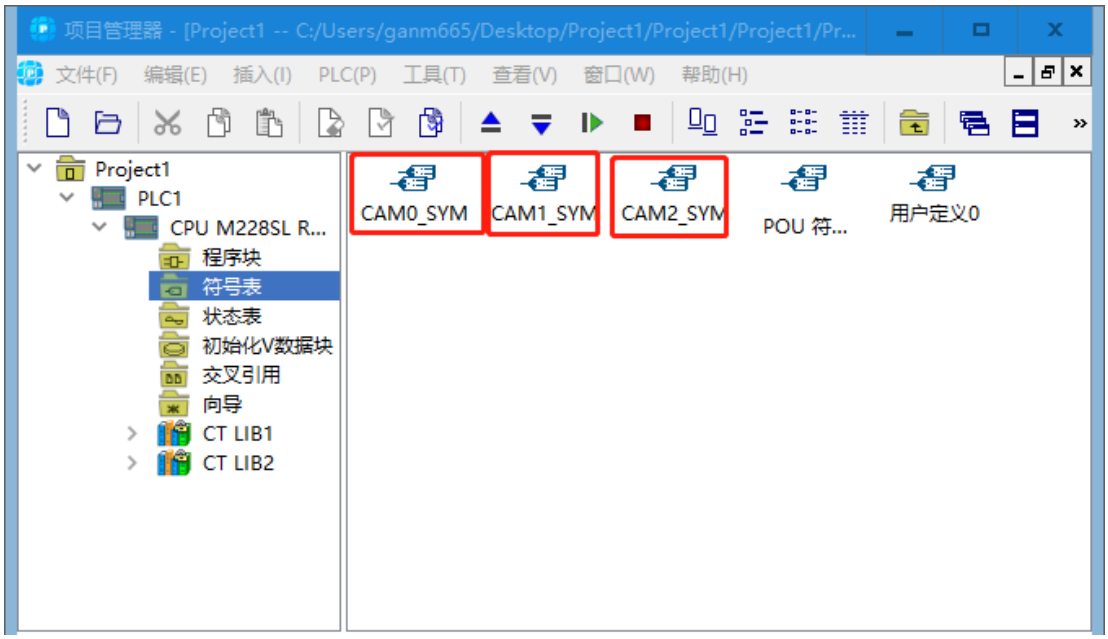
1、在硬件组态界面选择“工具”→“电子凸轮向导”即可进行电子凸轮配置。



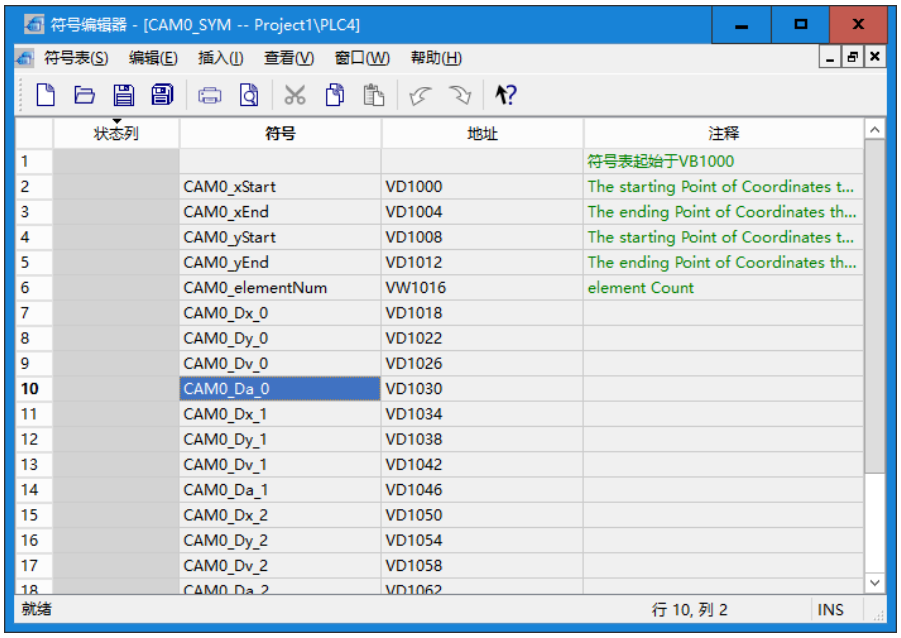
2、电子凸轮向导配置界面如下，选中“CAM”右键可插入新的 CAM、追剪、飞剪：



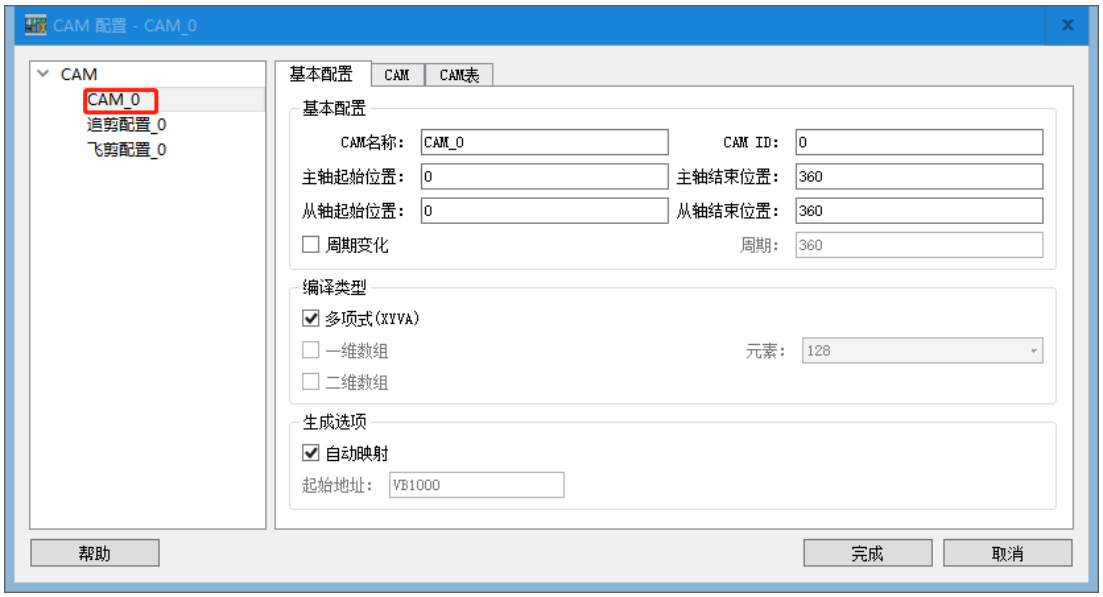
成功插入 CAM、追剪、飞剪后，在项目管理器界面选择“符号表”，可看到新增的 CAM。



3、点击“CAM”出现以下界面，可查看对应 CAM 的符号地址。



4、插入 CAM 后，配置界面如下所示。

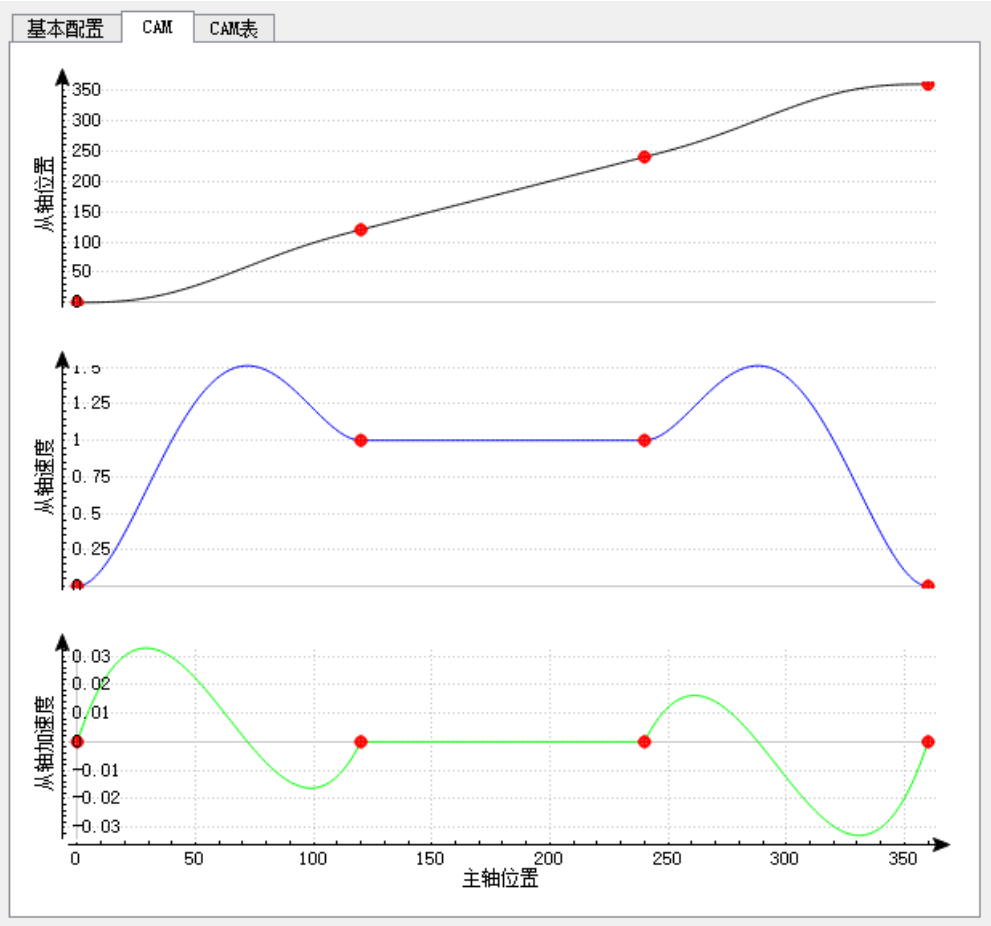


【基本配置】

- “基本配置”：用户可手动更改“CAM 名称”以及主从轴的起始和结束位置（主轴位置对应 CAM 曲线图中的横坐标，从轴位置对应 CAM 曲线图中的纵坐标），勾选“周期变化”，CAM 中曲线图将呈周期变化，主轴结束位置即为周期。
- “编译类型”：勾选“多项式(XYVA)”，CAM 在状态表中将以 x,y,a,v 的符号显示横纵坐标、速度与加速度状态，目前暂不支持“一维数组”和“二维数组”的编译类型。
- “生成选项”：勾选“自动映射”，将默认起始地址为 VB1000，取消勾选，可对起始地址进行修改，起始地址对应符号表中的起始地址。

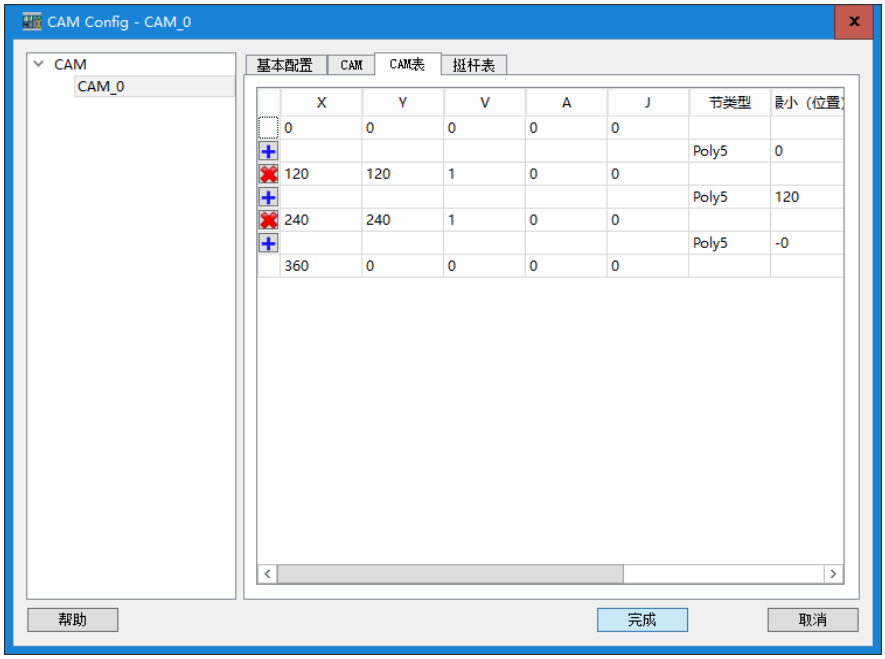
【CAM】

此页面以曲线图的形式分别显示“主轴位置”与“从轴位置”、“从轴速度”以及“从轴加速度”的关系。横坐标“主轴位置”最大值对应“基本配置”界面的“主轴结束位置”设置值，纵坐标“从轴位置”最大值对应“基本配置”界面“从轴结束位置”的设置值。点击红色节点，可将其拖动。

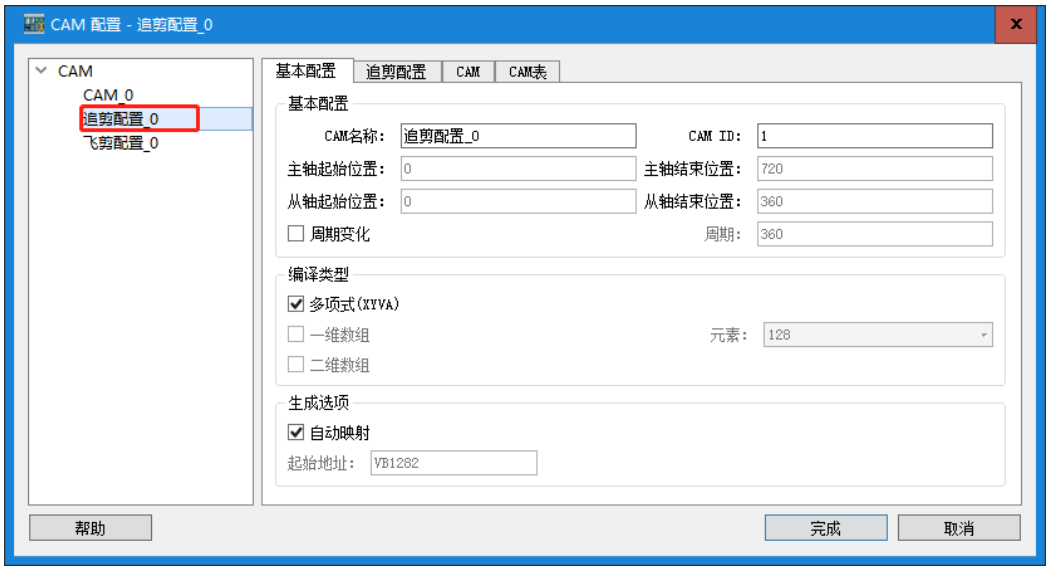


【CAM 表】

此页面对应显示 CAM 页面曲线图中红色节点的信息，一个红色节点对应一组数据，点击“”可添加新节点，点击“”可删除已添加的节点。双击“X、Y、V、A”对应的数值，可进行手动修改。



5、插入追剪后，配置界面如下所示。

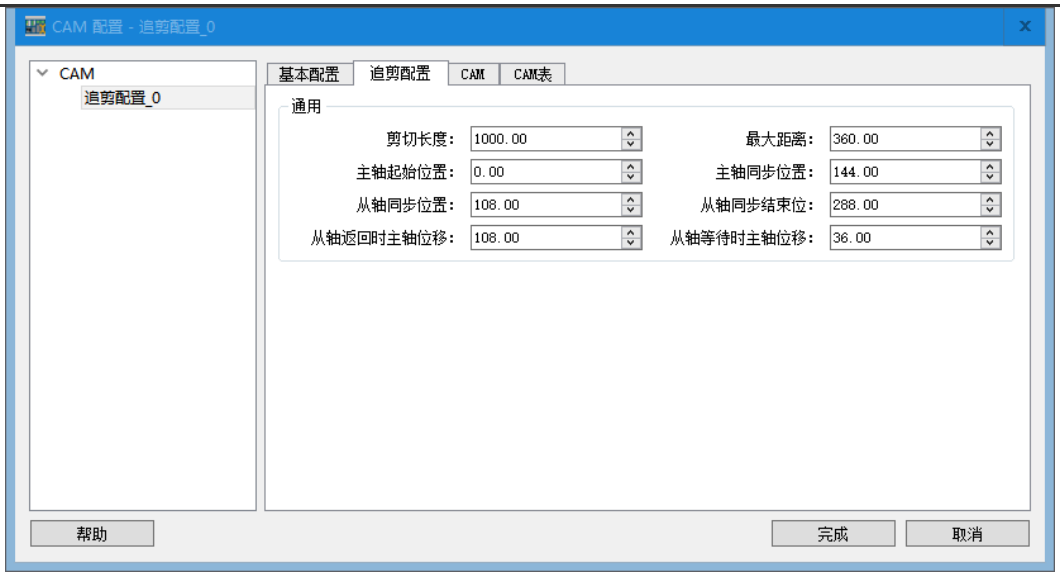


【基本配置】

- “基本配置”：用户可手动更改“CAM 名称”以及 CAM ID。主轴位置对应 CAM 曲线图中的横坐标，从轴位置对应 CAM 曲线图中的纵坐标，勾选“周期变化”，CAM 中曲线图将呈周期变化，主轴结束位置即为周期。
- “编译类型”：勾选“多项式(XYVA)”，CAM 在状态表中将以 x, y, a, v 的符号显示横纵坐标、速度与加速度状态，目前暂不支持“一维数组”和“二维数组”的编译类型。
- “生成选项”：勾选“自动映射”，将默认起始地址为 VB1000，取消勾选，可对起始地址进行修改，起始地址对应符号表中的起始地址。

注意：自动映射的 V 区地址不适用于动态变化的曲线，追剪飞剪指令修改曲线时会导致曲线点数的增加或减少从而导致内存冲突。当使用追剪飞剪指令时，推荐用户自定义起始地址，并预留足够的内存范围。

【追剪配置】

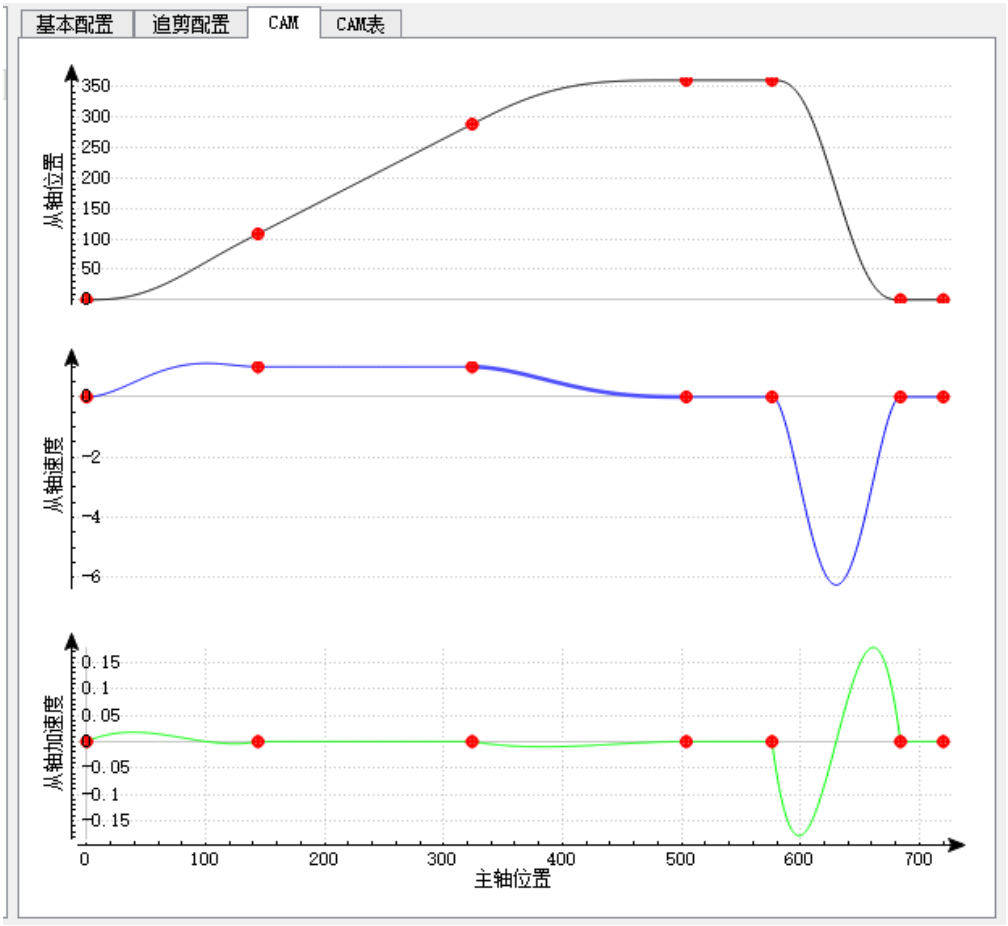


追剪配置中可以设置主轴和从轴的参数以及物料剪切长度。

剪切长度：即物料需要进行剪切的长度。
最大距离：即从轴可运动的最大距离。
主轴起始位置：主轴开始将物料往前传送的位置。
主轴同步位置：主轴同步开始的位置，此时主轴速度与从轴速度保持一致。
从轴同步位置：从轴同步开始的位置，从轴与主轴速度一致。
从轴同步结束位置：从轴同步结束的位置。
从轴返回主轴位移：从轴走到设定的最大距离后，往反方向返回原点期间主轴运动的位移。
从轴等待时主轴位移：从轴回到原点等待期间主轴运动的位移。

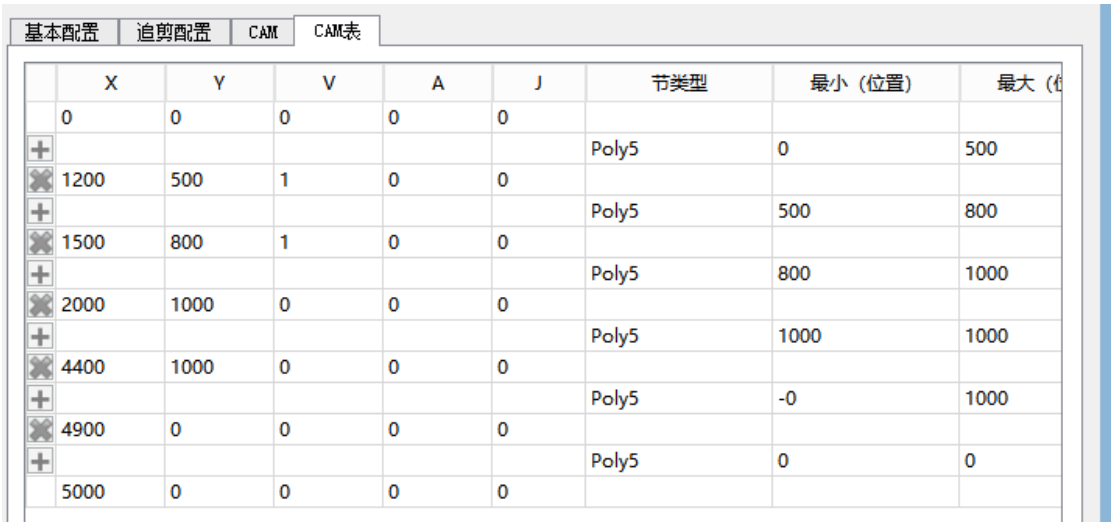
【CAM】

此页面以曲线图的形式分别显示“主轴位置”与“从轴位置”、“从轴速度”以及“从轴加速度”的关系。横坐标“主轴位置”最大值对应“基本配置”界面的“主轴结束位置”设置值，纵坐标“从轴位置”最大值对应“基本配置”界面“从轴结束位置”的设置值。追剪飞剪的 CAM 曲线上红色节点不可拖动。



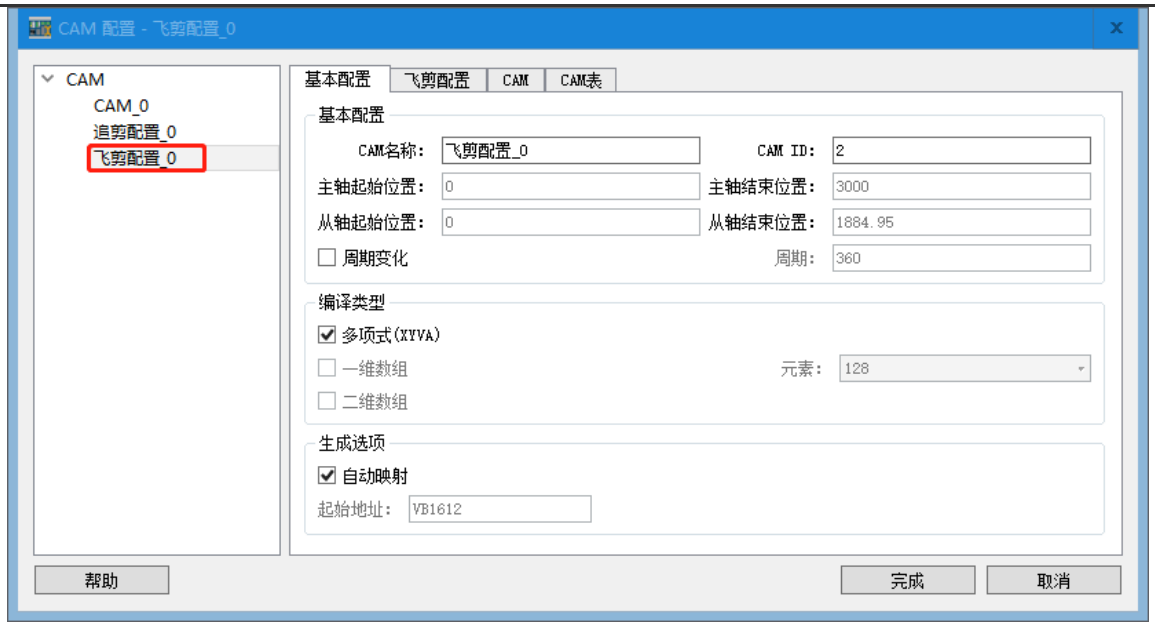
【CAM 表】

追剪中的 CAM 表参数不可修改，此页面对应显示 CAM 页面曲线图中红色节点的信息，一个红色节点对应一组数据。



X	Y	V	A	J	节类型	最小（位置）	最大（
0	0	0	0	0			
					Poly5	0	500
1200	500	1	0	0			
					Poly5	500	800
1500	800	1	0	0			
					Poly5	800	1000
2000	1000	0	0	0			
					Poly5	1000	1000
4400	1000	0	0	0			
					Poly5	-0	1000
4900	0	0	0	0			
					Poly5	0	0
5000	0	0	0	0			

6、插入飞剪后，配置界面如下所示。

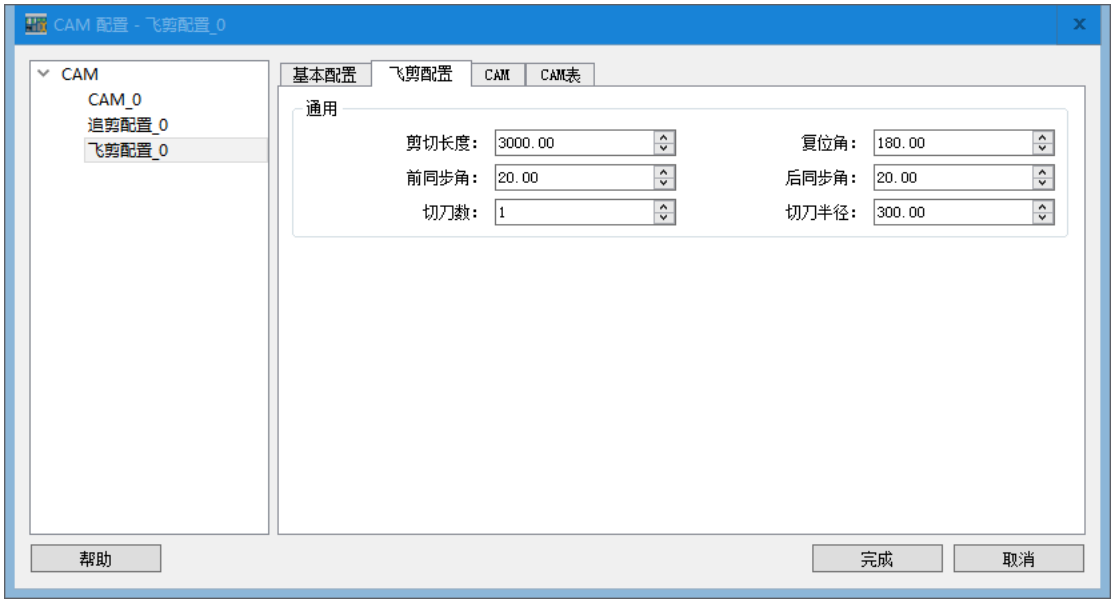


- “基本配置”：用户可手动更改“CAM 名称”以及 CAM ID。主轴位置对应 CAM 曲线图中的横坐标，从轴位置对应 CAM 曲线图中的纵坐标，勾选“周期变化”，CAM 中曲线图将呈周期变化，主轴结束位置即为周期。
- “编译类型”：勾选“多项式(XYVA)”，CAM 在状态表中将以 x, y, a, v 的符号显示横纵坐标、速度与加速度状态，目前暂不支持“一维数组”和“二维数组”的编译类型。
- “生成选项”：勾选“自动映射”，将默认起始地址为 VB1000，取消勾选，可对起始地址进行修改，起始地址对应符号表中的起始地址。

注意:自动映射的 V 区地址不适用于动态变化的曲线，追剪飞剪指令修改曲线时会导致曲线点数的增加或减少从而导致内存冲突。当使用追剪飞剪指令时，推荐用户自定义起始地址，并预留足够的内存范围。

【飞剪配置】

飞剪配置中可以设置从轴的参数以及物料剪切长度。



剪切长度：即物料需要剪切的长度。

复位角：指复位之后，切刀从原点按运行方向运行，当前同步角边沿到达切点经过的角度值。

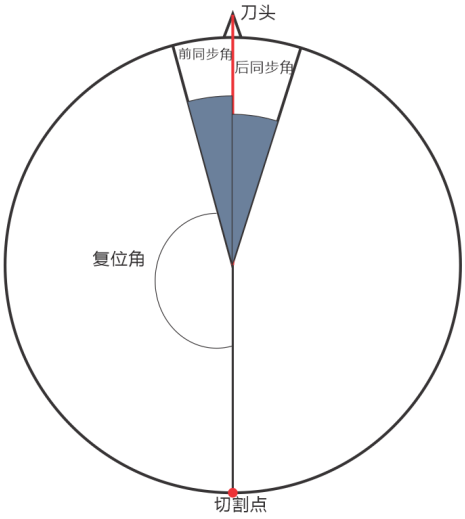
前同步角：物料先经过的同步区域角度为前同步角。

后同步角：物料后经过的同步区域角度为后同步角，目前规定前后同步角的大小要一致。

切刀数：从轴上的切刀数目。若切刀数目为两个或两个以上，要求切刀均匀角度安装，两两切刀之间的角度相同。

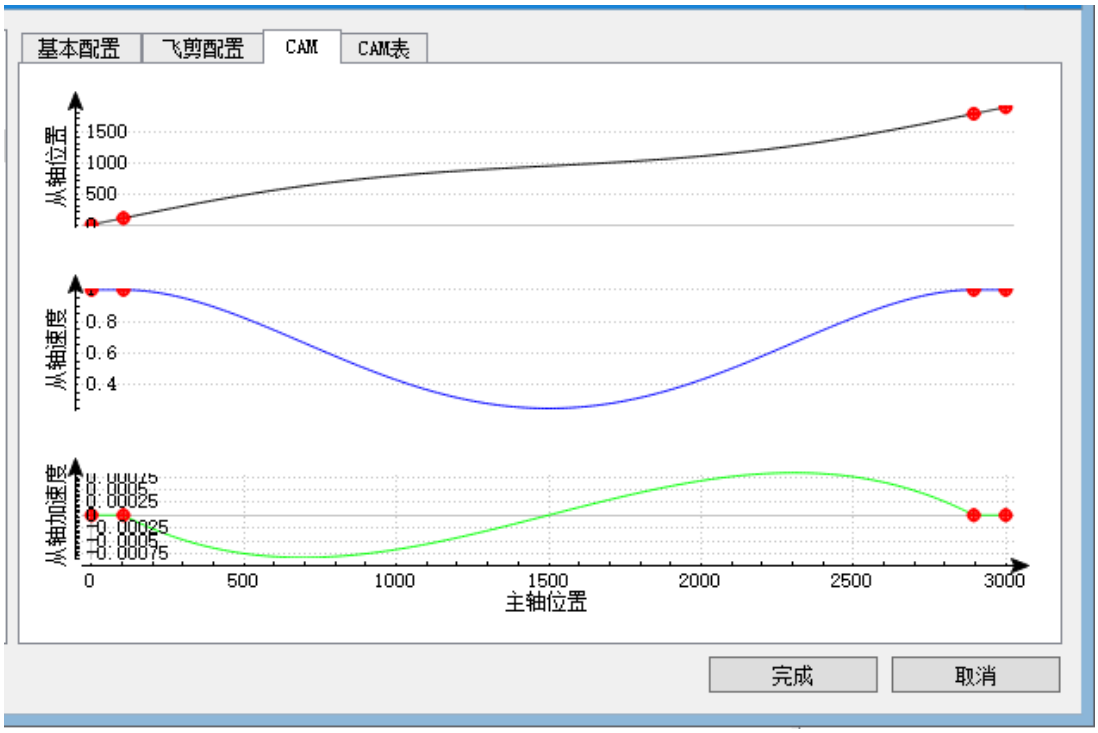
切刀半径：从轴圆心到刀头的距离即为切刀半径。

从轴结构示意图如下：



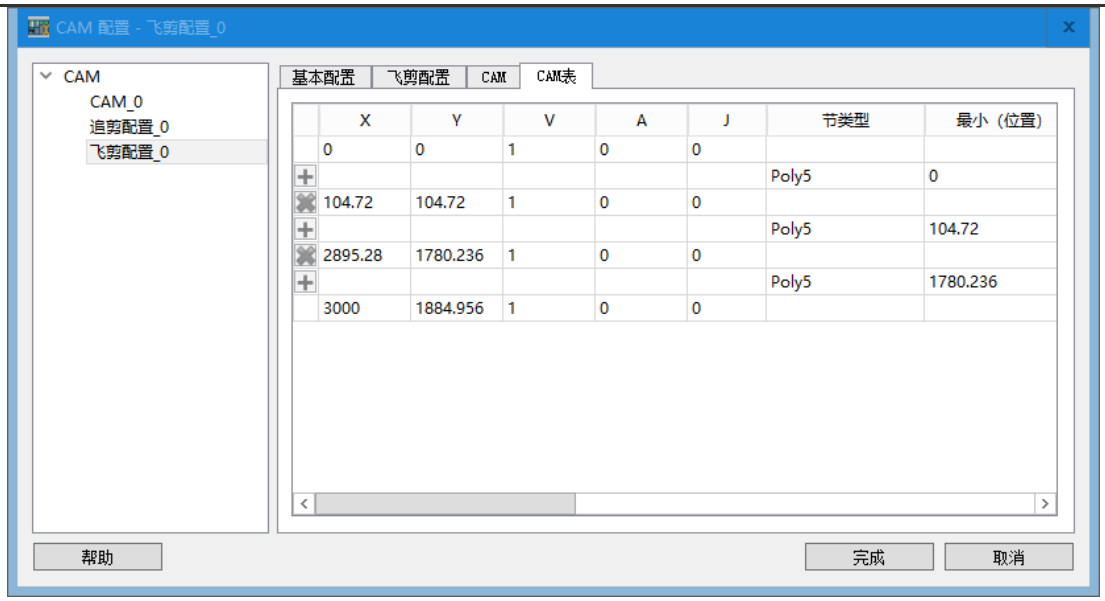
【CAM】

此页面以曲线图的形式分别显示“主轴位置”与“从轴位置”、“从轴速度”以及“从轴加速度”的关系。横坐标“主轴位置”最大值对应“基本配置”界面的“主轴结束位置”设置值，纵坐标“从轴位置”最大值对应“基本配置”界面“从轴结束位置”的设置值。与普通凸轮不同，追剪飞剪的 CAM 曲线红色节点不可拖动。



【CAM 表】

飞剪中的 CAM 表参数不可修改。此页面对应显示 CAM 页面曲线图中红色节点的信息，一个红色节点对应一组数据。

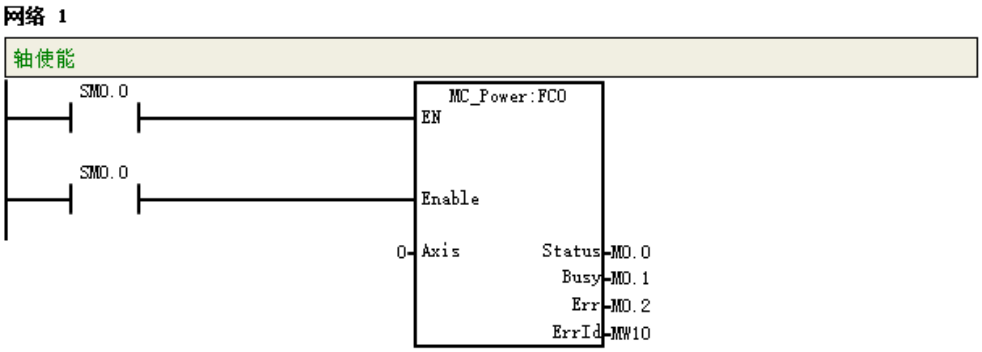


4.2.10 电子凸轮程序示例

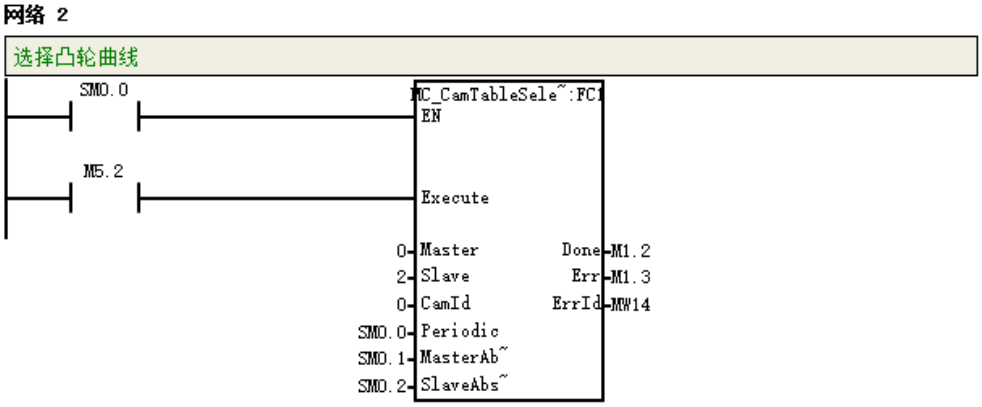
以下程序为 CTMC 系列 CPU 使用轴指令 ct_plcopen_lib(v2.0)实现电子凸轮功能的示例。

程序实现的网络示意图

网络 1：对轴进行使能操作。



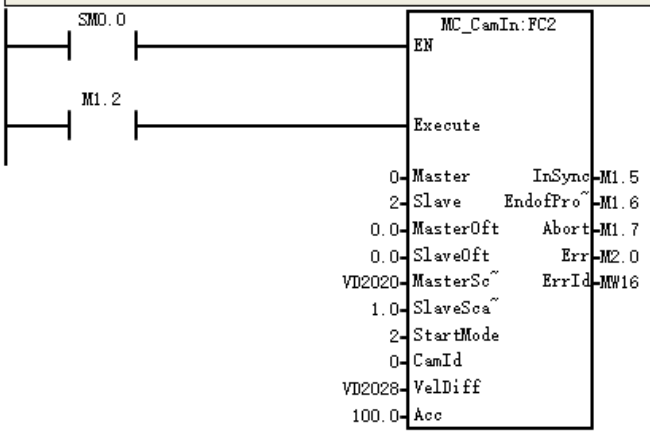
网络 2：选择凸轮曲线，设置主站、从站信息、状态信息等参数。



网络 3：建立主从轴间的凸轮关系，建立凸轮关系时，根据应用需求指定主从轴的偏移值，缩放比和启动模式。当指令执行完毕，从轴根据凸轮曲线跟随主轴运动。

网络 3

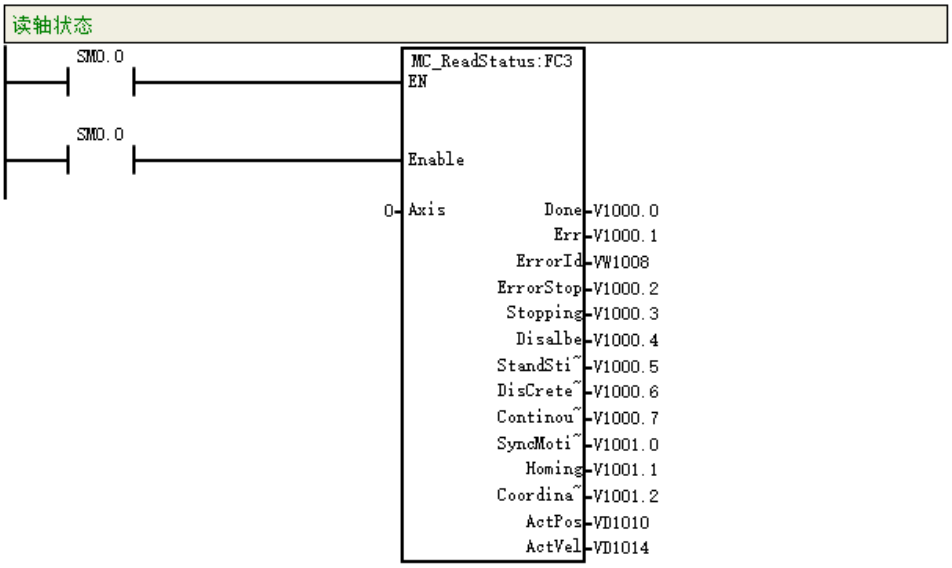
建立主从轴间凸轮关系



网络 4：读取相应轴状态。

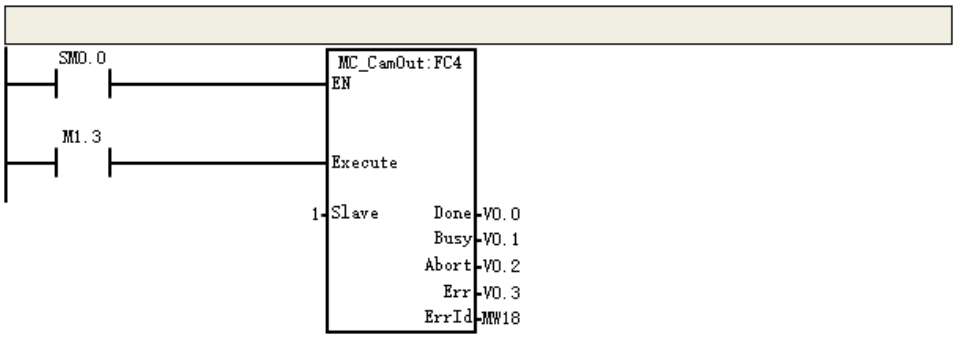
网络 4

读轴状态

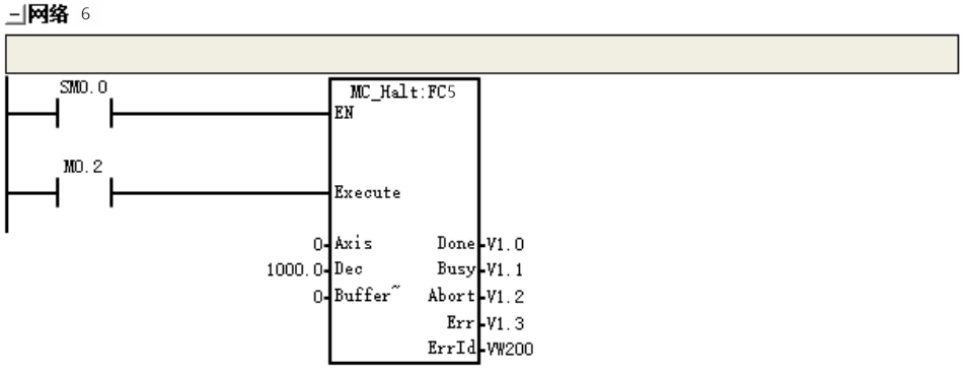


网络 5：解除主从轴间的凸轮关系，关系解除后，从轴会以脱离前的速度继续运动。

网络 5



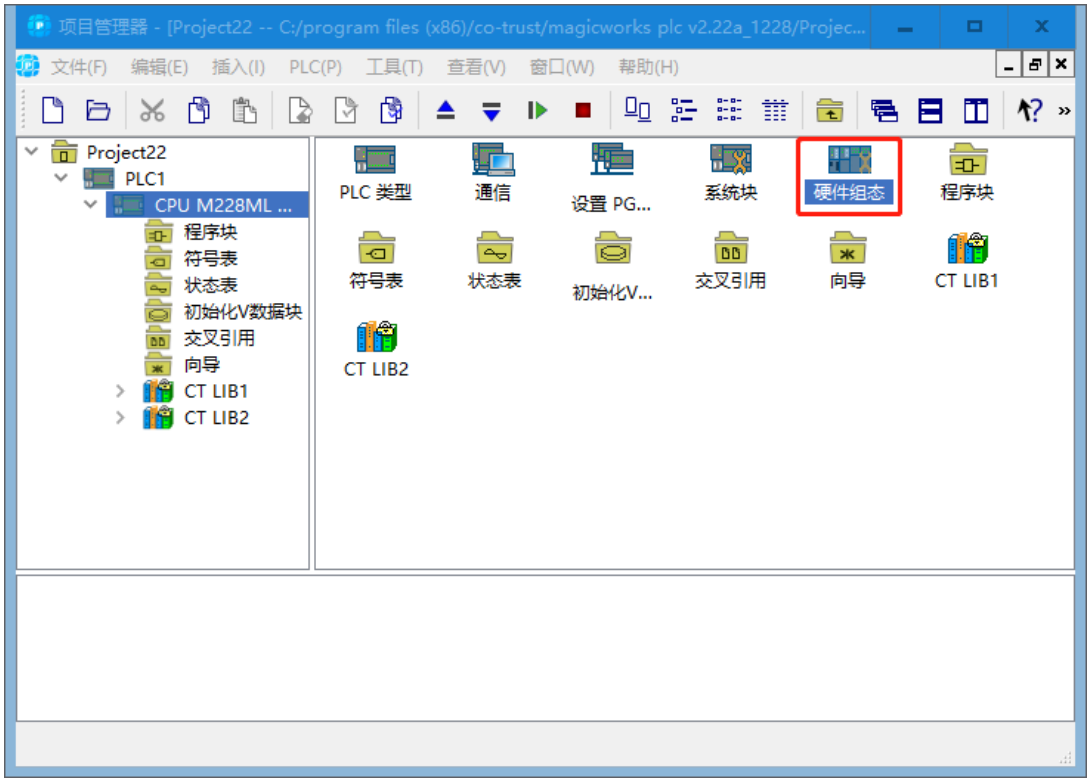
网络 6：轴按给定的减速度减速，直到停止。



4.2.11 连续插补功能应用举例

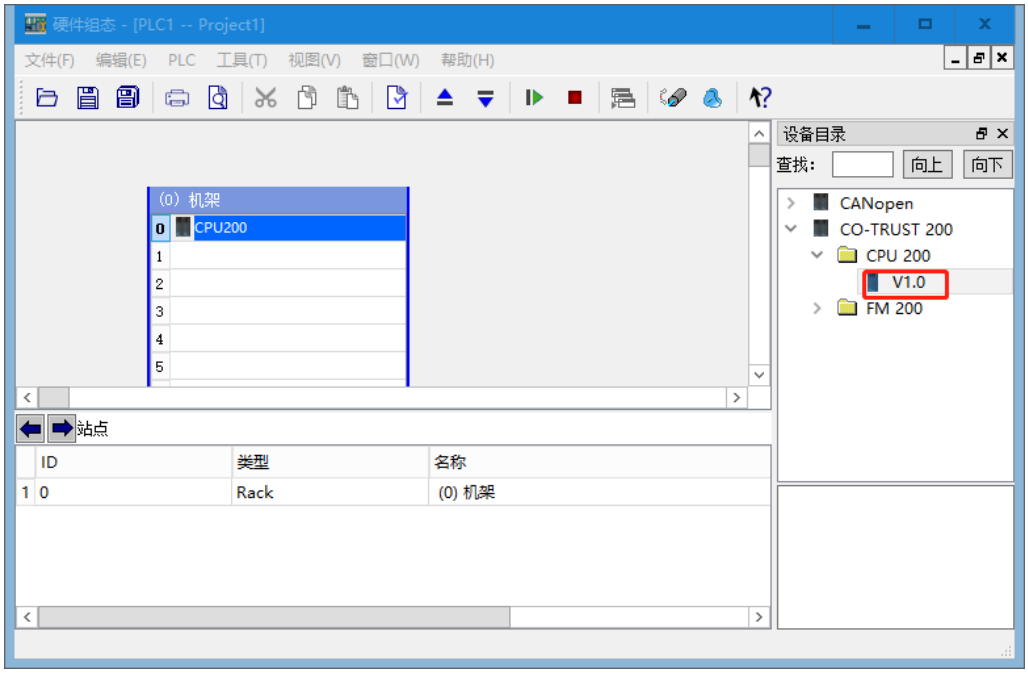
下面以 CPU M228ML 为例子对连续插补功能进行具体操作介绍。

在 M228ML 站点的项目管理器界面选择“硬件组态”，进行硬件组态，然后在主程序中调用插补指令进行编程。本例调用的插补指令为相对位移直线插补指令。



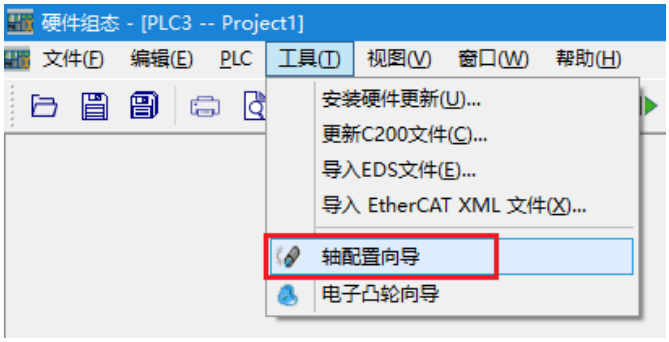
步骤一：添加 CPU 模块至 CPU 上。

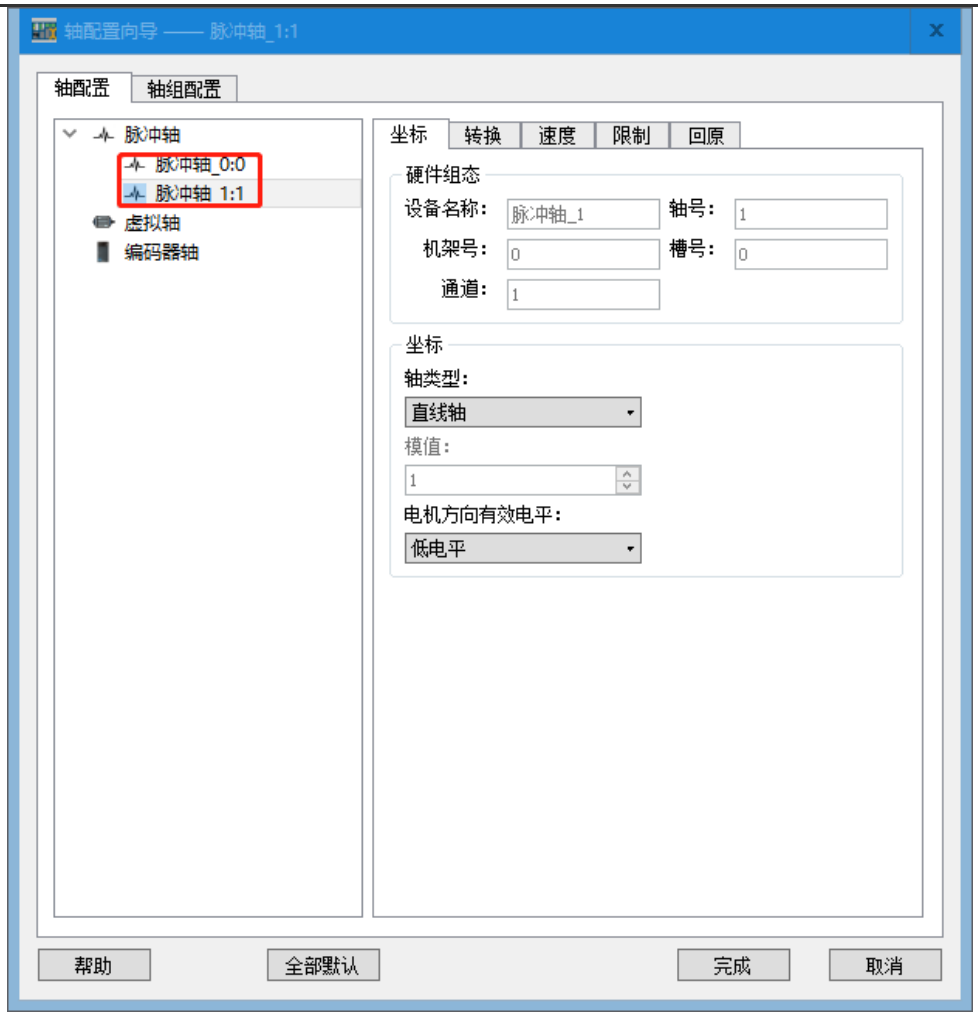
硬件组态界面中，在“CO-TRUST 200”项目树下展开“CPU 200”文件夹，添加 CPU 模块，CPU 只能放至机架的 0 号槽内，如下图所示。



步骤二：轴配置

在硬件组态界面选择“工具”→“轴配置向导”，添加两个脉冲轴，如下图所示。

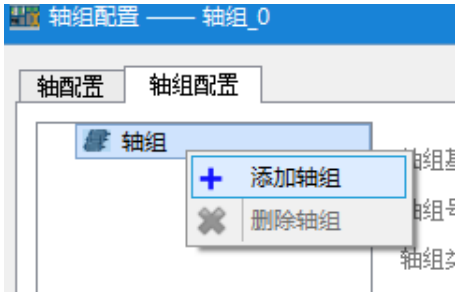




<备注> 使用插补功能需要配置两个或两个以上的脉冲轴。

步骤三：配置轴组

轴组配置即将配置好的轴两个或三个一组以轴组的形式自由组合，本例配置的轴有两个；在“轴组配置”界面，右键点击“添加轴组”进行轴组添加，如下图所示：

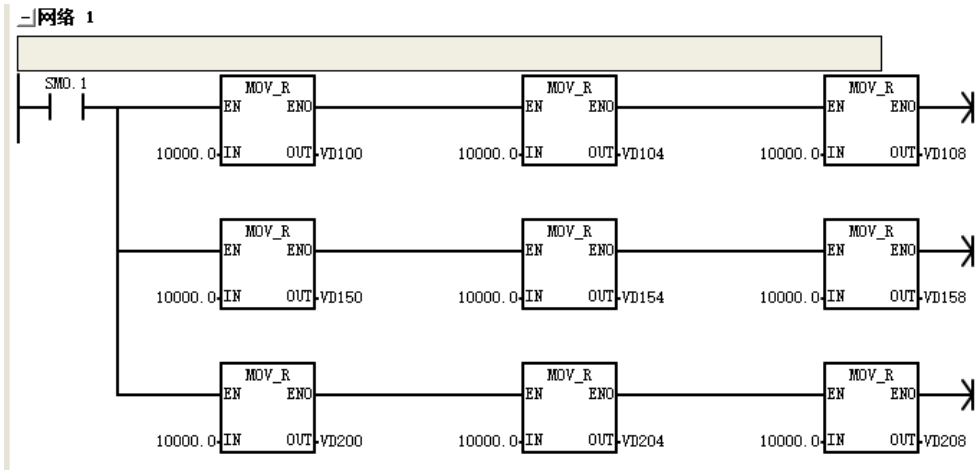


添加成功的轴组界面如下所示：

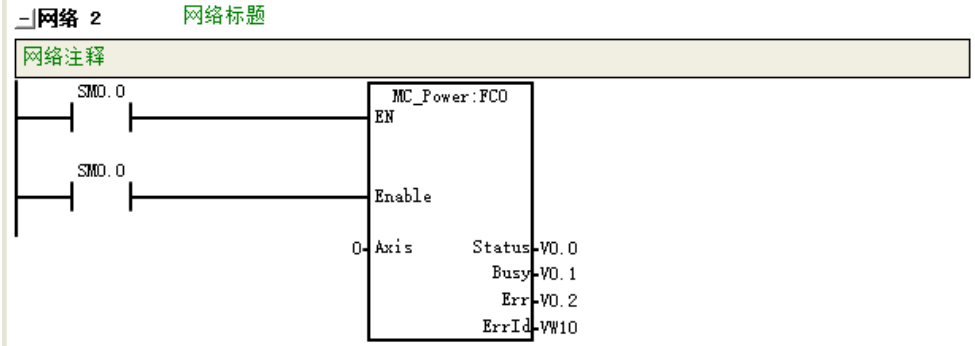
- “轴组基本配置”：“轴组号”不可更改，“轴组类型”下拉选项可选择“2-轴正交”或“3-轴正交”，分别对应可选两组或三组“构成轴”；“运算周期”可选“脉冲”；“加速模式”可选“梯形”，“sin*2”和“二次”。
- “逻辑轴”：默认六组，不可更改。
- “构成轴”：“2-轴正交”对应两个构成轴，“3-轴正交”对应三个构成轴，下拉选项可选择“脉冲轴”或“虚拟轴”。

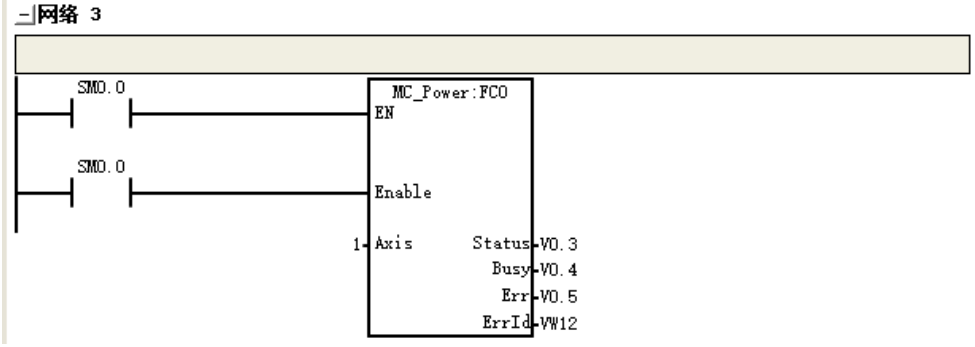
步骤四：在主程序中调用插补指令进行编程。

网络 1：设置插补指令的速度

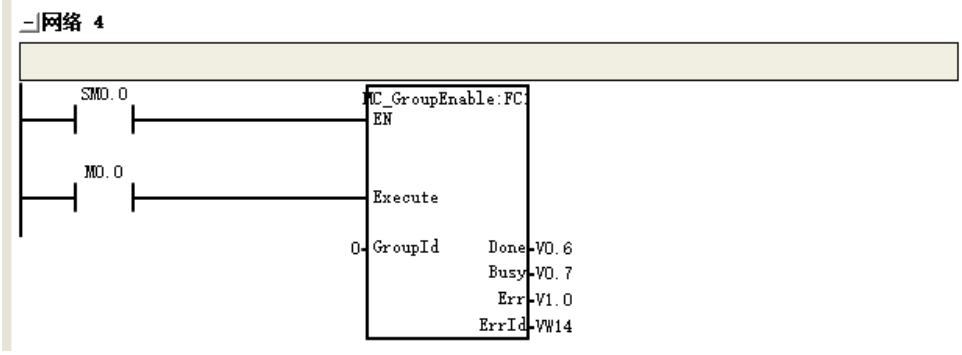


网络 2、网络 3：分别对两个脉冲轴进行使能操作。

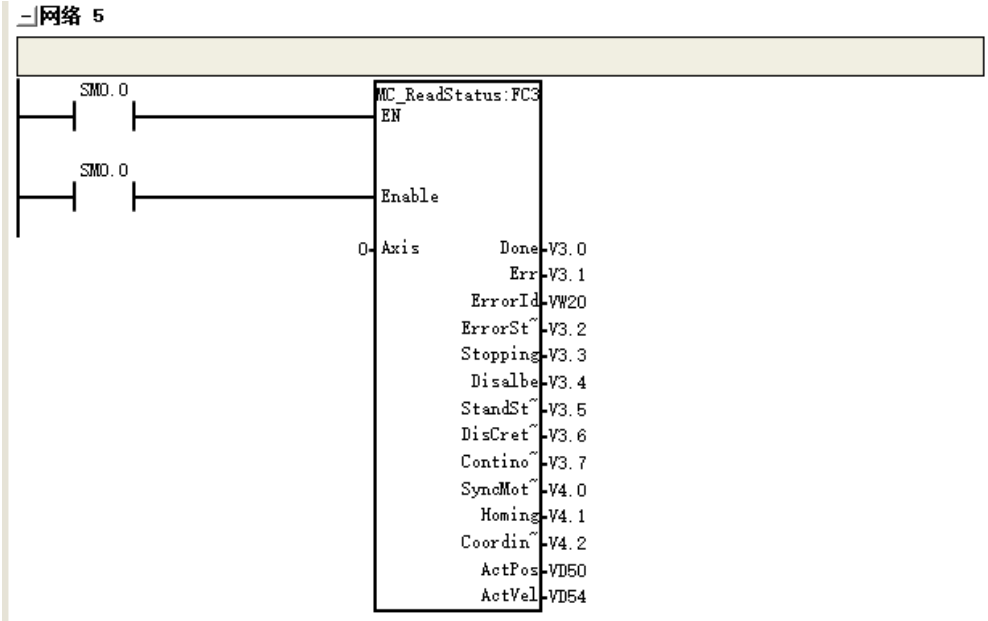


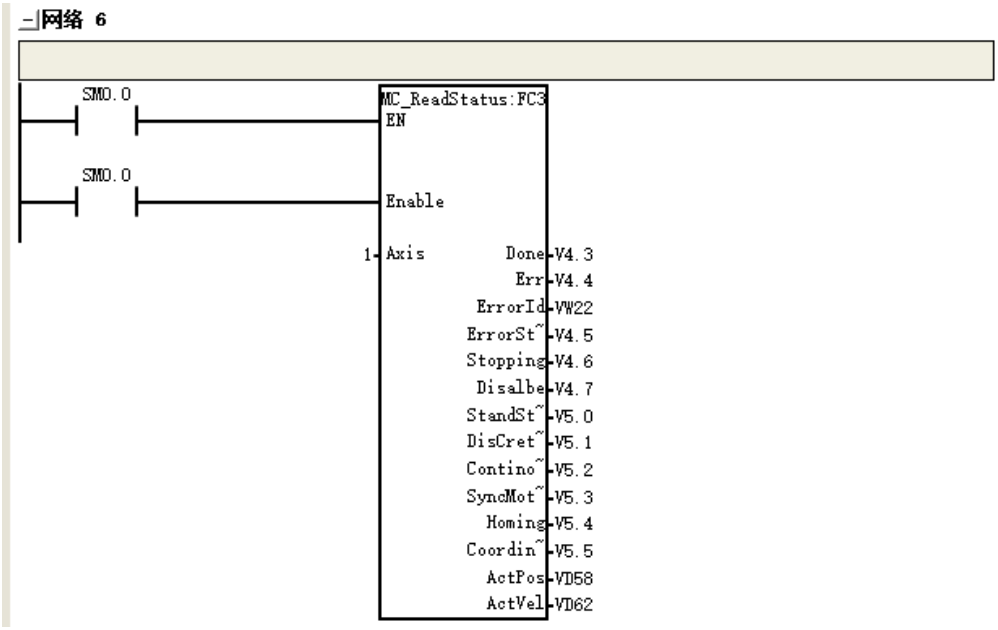


网络 4：调用轴组有效指令，使轴组有效



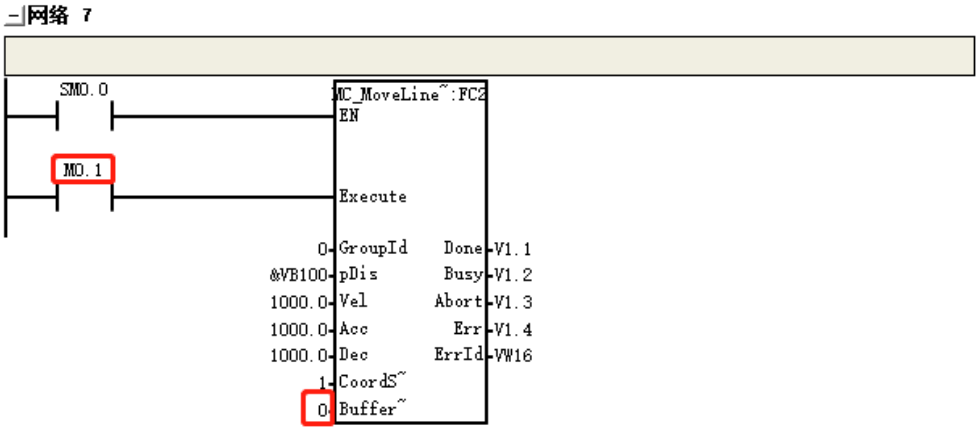
网络 5、网络 6：读取相应的轴状态。

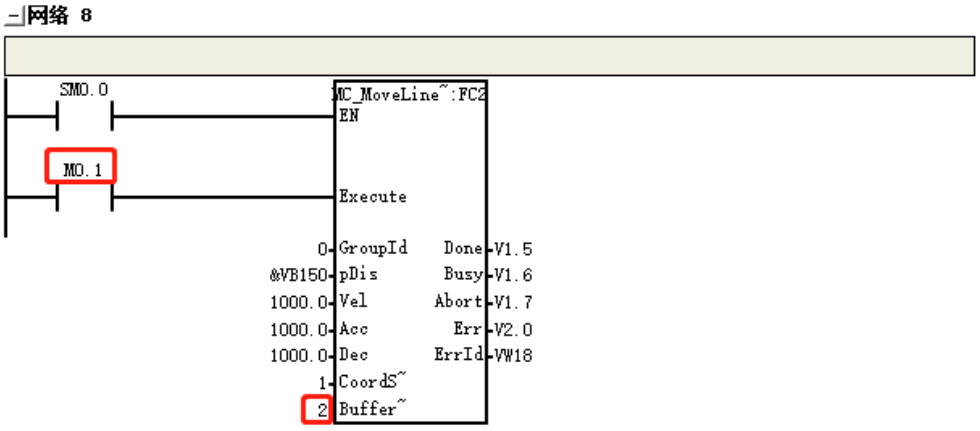


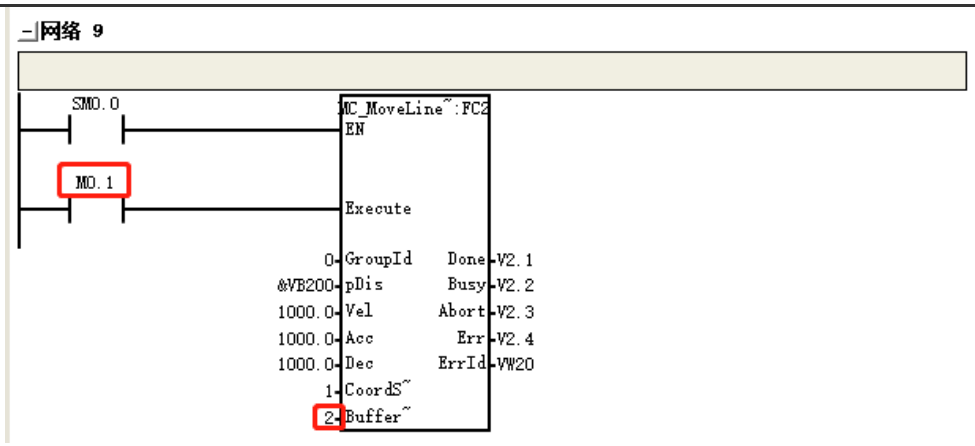


网络 7、8、9:设置运行速度、加速度、减速度，插补模式等参数，指令使能位可以用同一个位来同时触发（程序中缓存 2 条连续插补指令），然后按照指令先后顺序进行连续直线插补运动。

<备注>： 连续插补一次最大缓存指令条数为 8 条，如果想缓存第 9 条，当第 1 条指令运行完成后再使能缓存第 9 条。





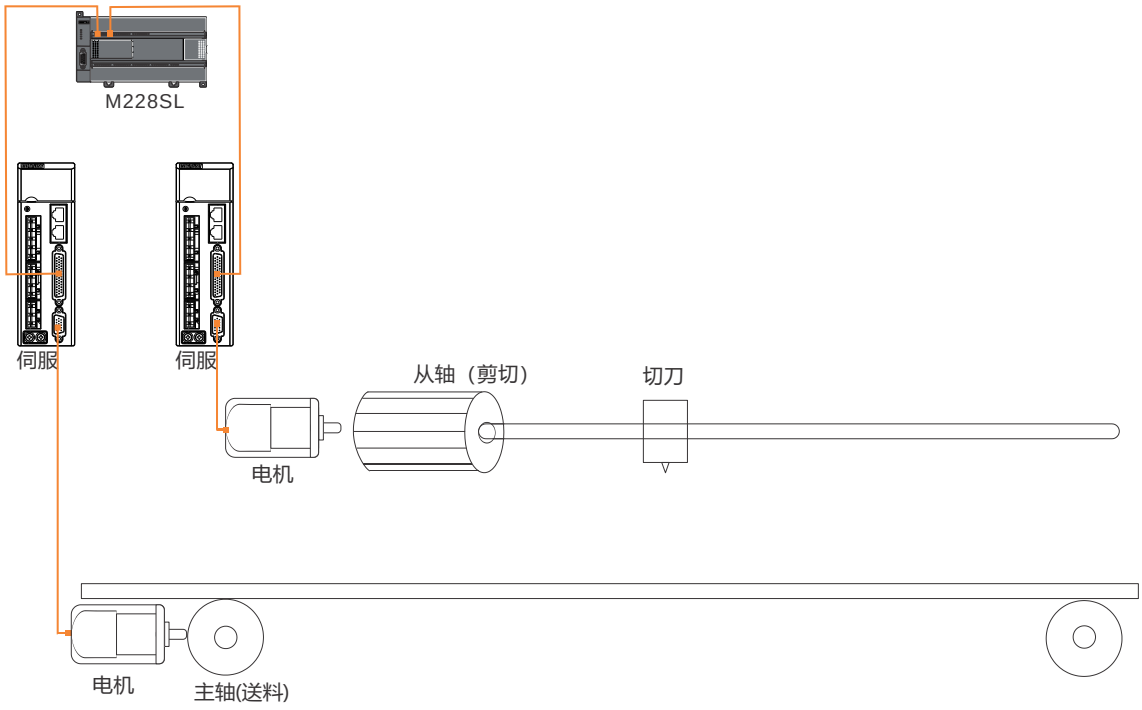


4.2.12 追剪功能介绍

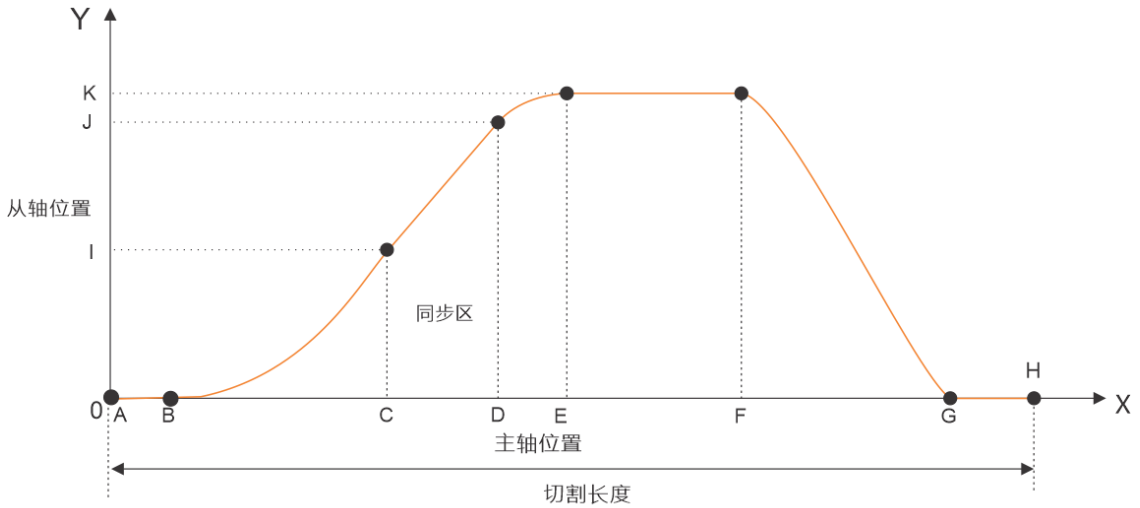
追剪工艺介绍

CTMC 系列运动控制器 M228SL 支持追剪飞剪功能。追剪是指对运动中的物料进行切割的工艺。追剪执行机构与物料平行，当追剪执行机构与物料同步后，追剪执行机构上的刀头对物料进行切割，切割动作完成后刀头返回原点位置。

追剪架构示意图如下：



从轴与主轴位置关系图如下：



	注释	对应的指令参数
A	起始位置	--
B	主轴起始位置	MasterStartPos
C	主轴同步位置	MasterSyncPos
D	主轴同步结束位置	--
F	从轴开始减速返回原点位置	--
G	从轴返回到原点位置	--
I	从轴同步位置	SlaveSyncPos
J	从轴同步结束位置	SlaveEndPos
K	从轴可运动的最大距离位置	--

A→B：从轴等待触发信号。

B→C：调整区，此区域内从轴调整速度。

C→D：同步区，同步区内主轴与从轴速度大小一致，切割动作在同步区内完成。

D→E：从轴离开同步区，减速运行到设定的最大距离停止。

E→F：从轴停在设定的最大距离位置等待，这一段可以是 0，不等待。

F→G：从轴做反向减速运动，回到等待位置。

G→H：从轴回到原点位置后，等待进入下一个周期的等待距离，这一段可以是 0。

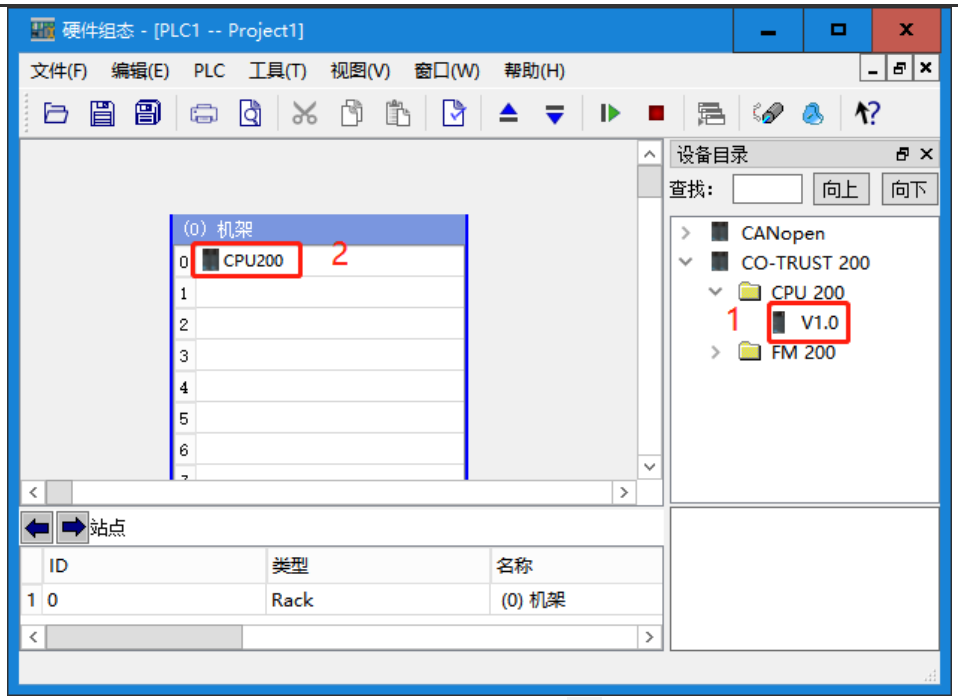
追剪功能应用

追飞剪功能的使用场景多应用于灌装行业，旋盖行业，瓦楞行业，枕式包装行业，瓶类丝网印刷行业，剪切行业、定长切纸行业，绕线行业，追标行业，同步盖印行业，追锯追剪行业等。

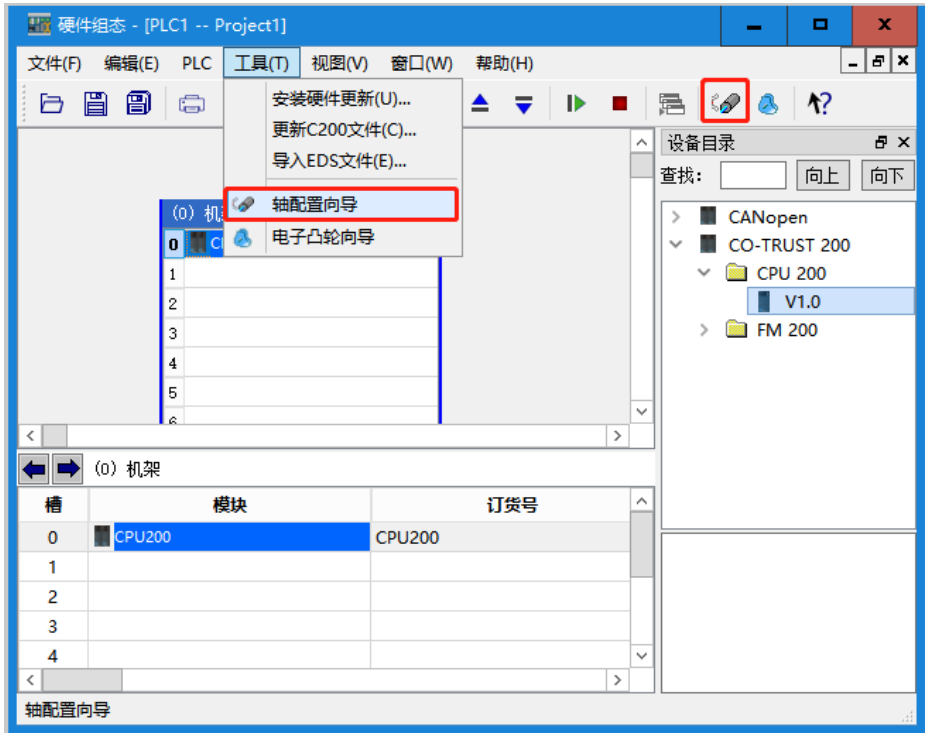
Magicworks PLC V2.22 版本集成追剪及飞剪功能，可在硬件组态界面对追剪飞剪进行组态，设置追剪飞剪相关参数；通过 CAM 曲线对主轴位置与从轴位置、从轴速度、从轴加速度之间的关系进行预览。

1、新建追剪

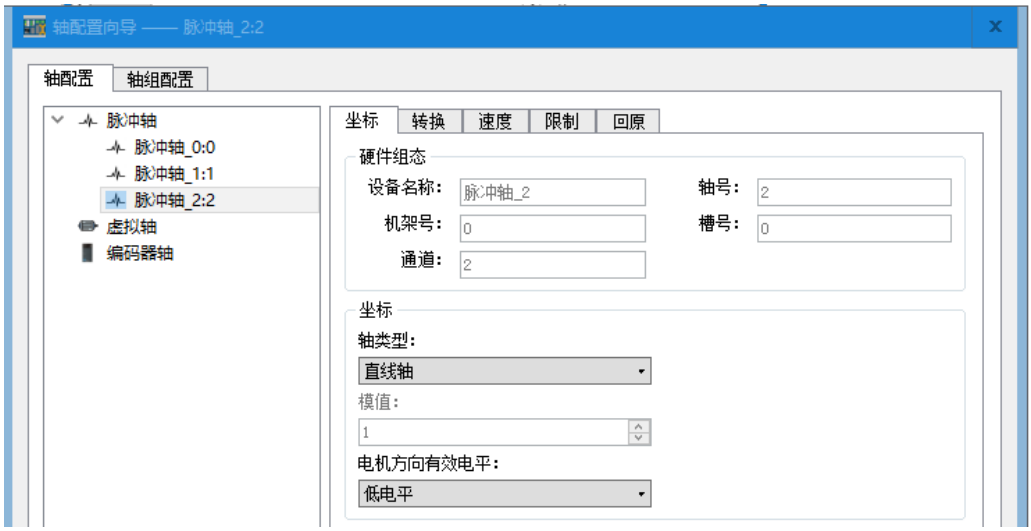
打开 Magicworks PLC 并新建工程，在硬件组态中进行组态，选择 CPU200，将 CPU200 拖到机架上，CPU200 只能放在机架 0 上。



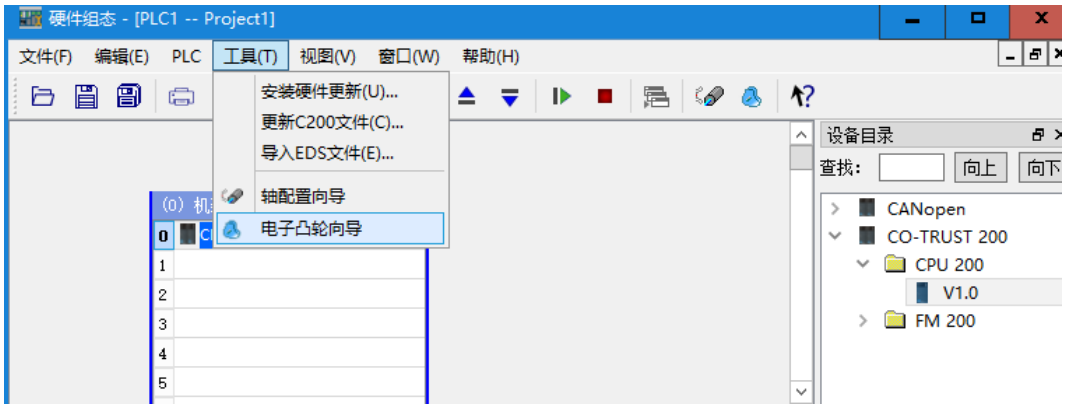
点击“工具”→“轴配置向导”或直接点击图标即可进行轴配置。



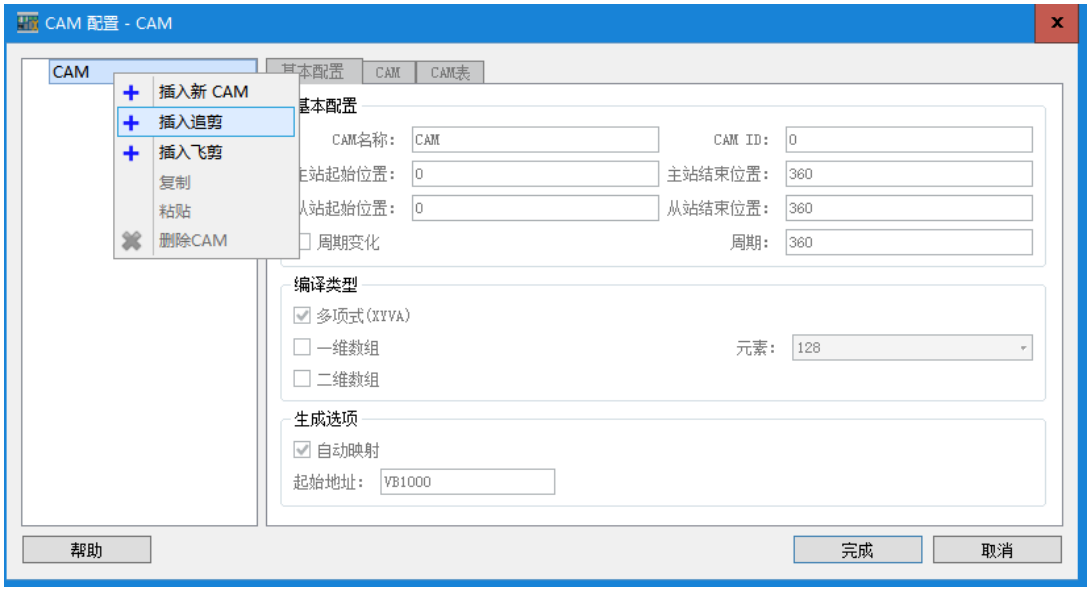
选择轴配置后，可任意添加脉冲轴、虚拟轴、编码器轴；选择“轴类型”及“电机方向有效电平”，点击“完成”。



点击“工具”→“电子凸轮向导”即可进行电子凸轮配置。

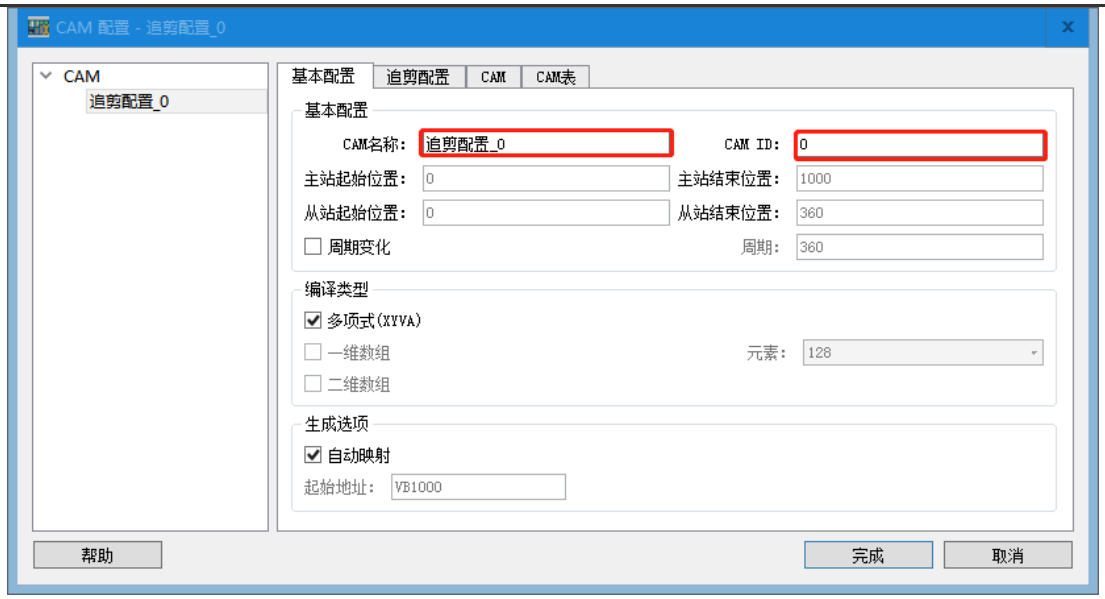


右键点击“CAM”，即可插入追剪。

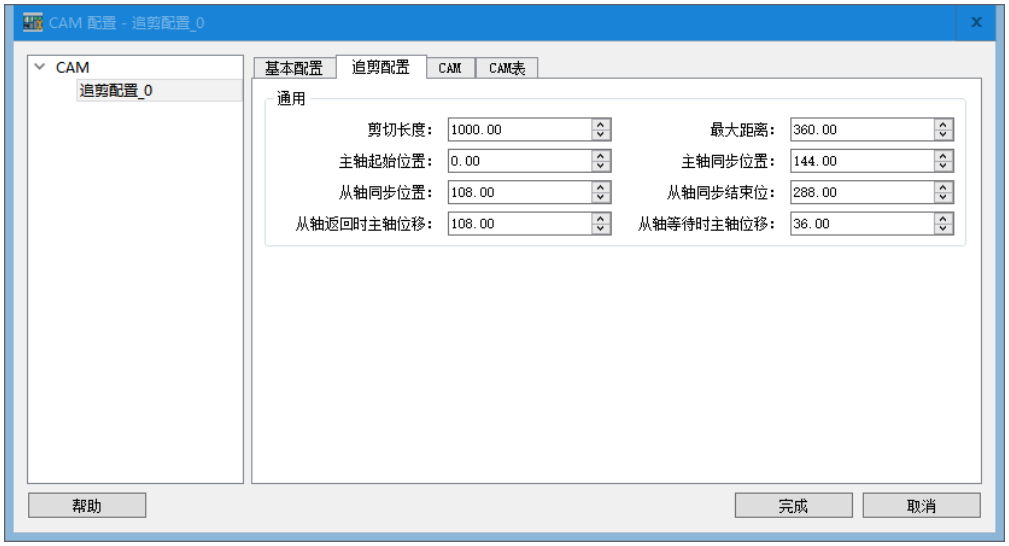


2、追剪参数设置

基本配置中，可以修改 CAM 名称以及 CAM ID



追剪配置中可以设置主轴和从轴的参数以及物料剪切长度。



剪切长度：即物料需要进行剪切的长度。

最大距离：即从轴可运动的最大距离。

主轴起始位置：主轴开始将物料往前传送的位置。

主轴同步位置：主轴同步开始的位置，此时主轴速度与从轴速度保持一致。

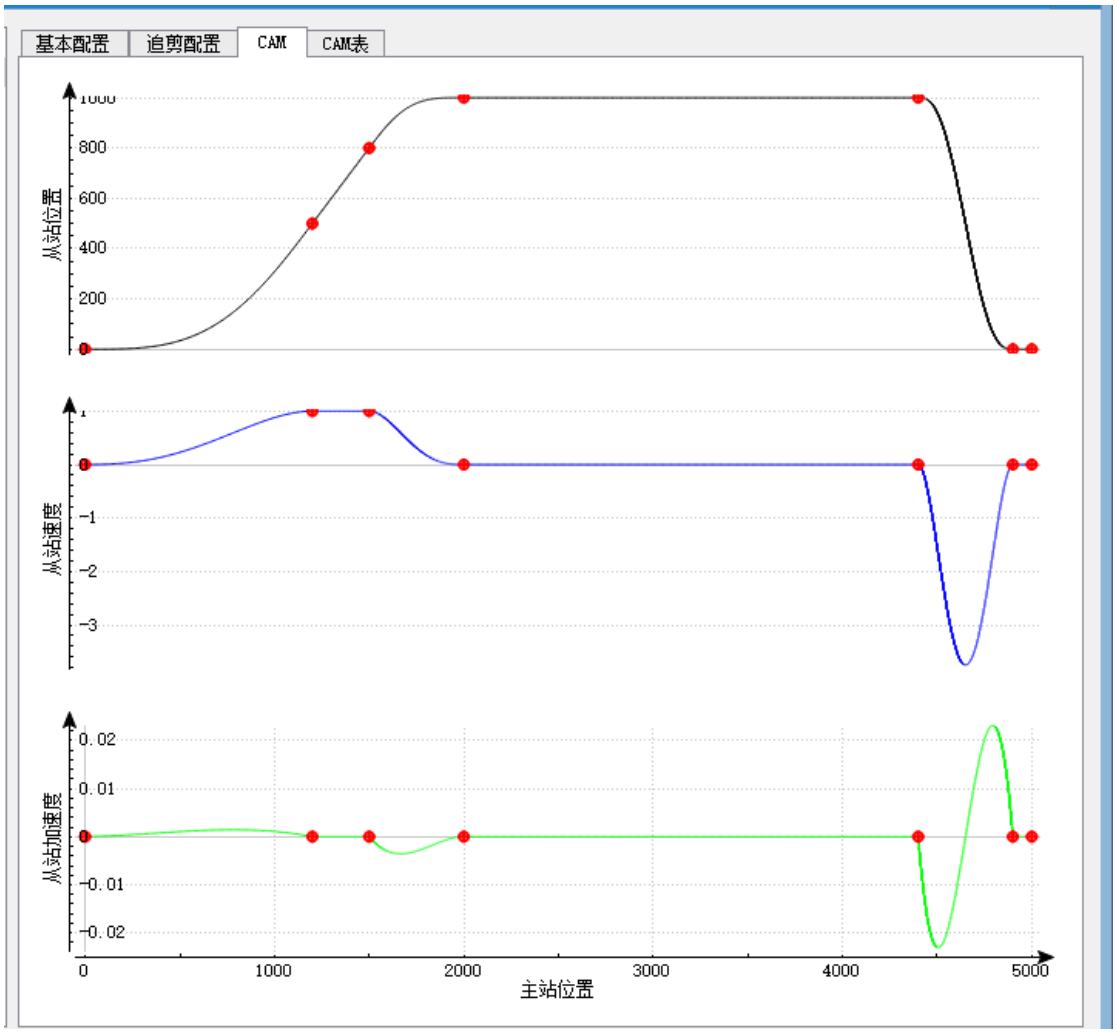
从轴同步位置：从轴同步开始的位置，从轴与主轴速度一致。

从轴同步结束位置：从轴同步结束的位置。

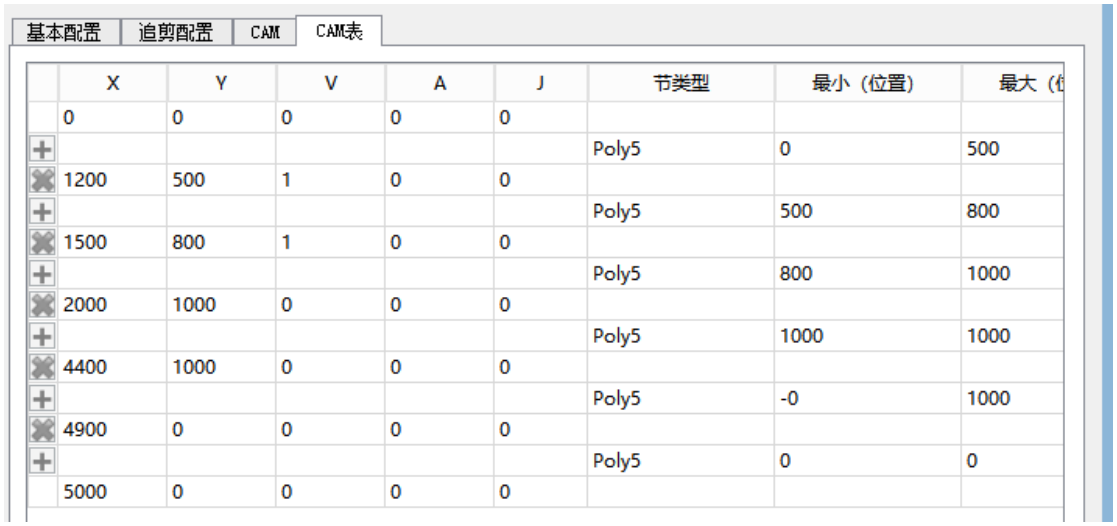
从轴返回主轴位移：从轴走到设定的最大距离后，往反方向返回原点期间主轴运动的位移。

从轴等待时主轴位移：从轴回到原始等待期间主轴运动的位移。

CAM 曲线可以预览主轴位置与从轴位置、从轴速度以及从轴加速度三者的关系。



与普通电子凸轮不同，CAM 表中的参数不可修改。此页面对应显示 CAM 曲线图中红色节点的信息，一个红色节点对应一组数据。



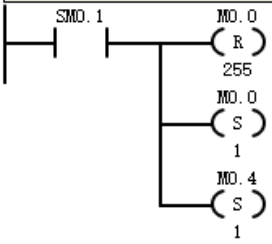
X	Y	V	A	J	节类型	最小（位置）	最大（
0	0	0	0	0			
					Poly5	0	500
1200	500	1	0	0			
					Poly5	500	800
1500	800	1	0	0			
					Poly5	800	1000
2000	1000	0	0	0			
					Poly5	1000	1000
4400	1000	0	0	0			
					Poly5	-0	1000
4900	0	0	0	0			
					Poly5	0	0
5000	0	0	0	0			

3、追剪程序示例：

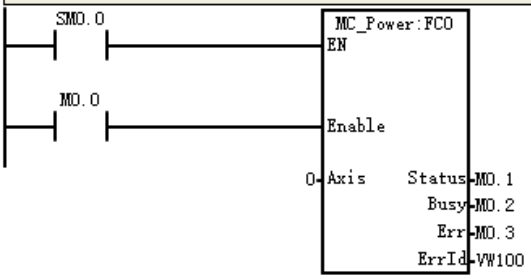
程序注释

网络 1　　初始化

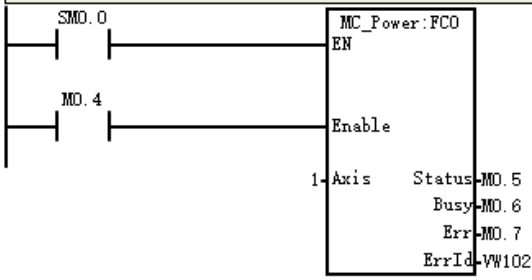
网络注释



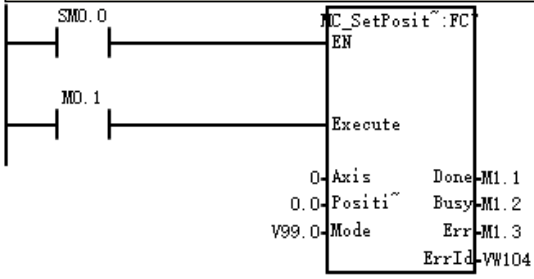
网络 2　　主轴--0轴使能



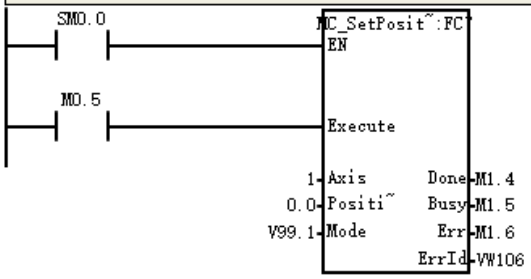
网络 3　　从轴--1轴使能



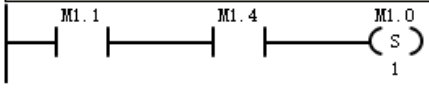
网络 4　　轴0复位，实际系统按实际情况实现



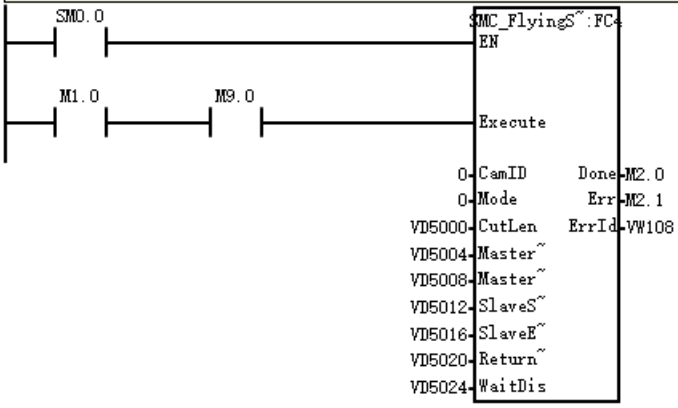
网络 5　从轴复位，实际系统按实际情况实现



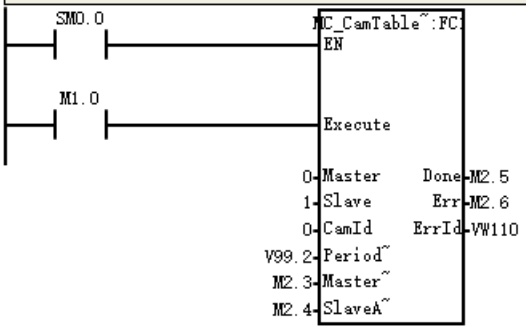
网络 6　两个轴复位完成，可以启动追剪凸轮



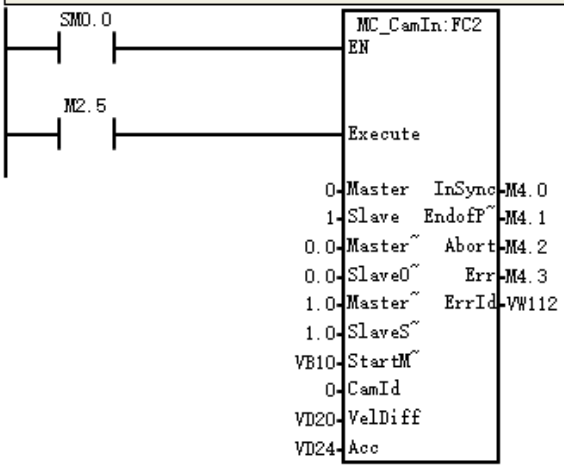
网络 7　追剪配置指令，调用指令会根据给定的参数，修改凸轮的曲线，并在凸轮的下一个周期生效



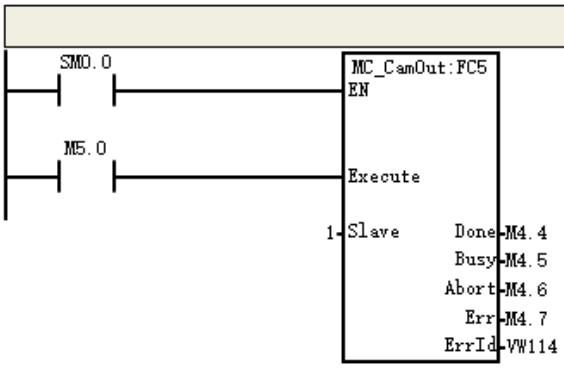
网络 8　绑定主从轴和追剪凸轮关系



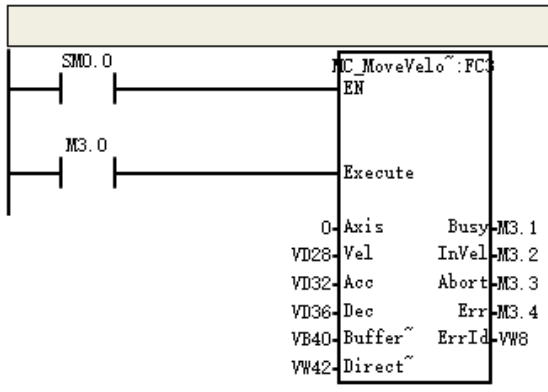
网络 9　　启动追剪凸轮

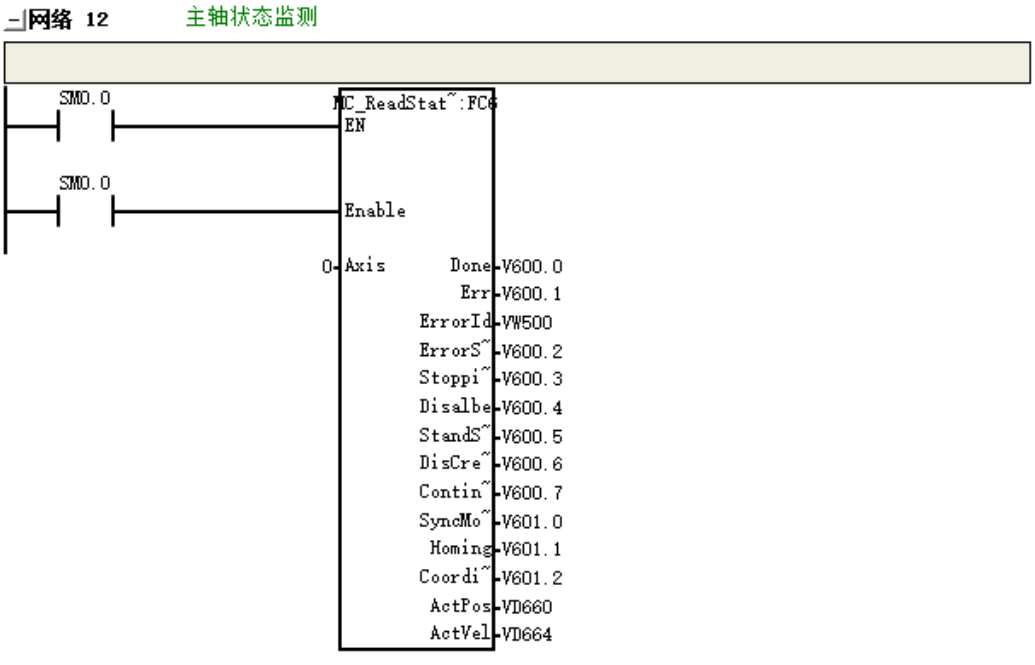


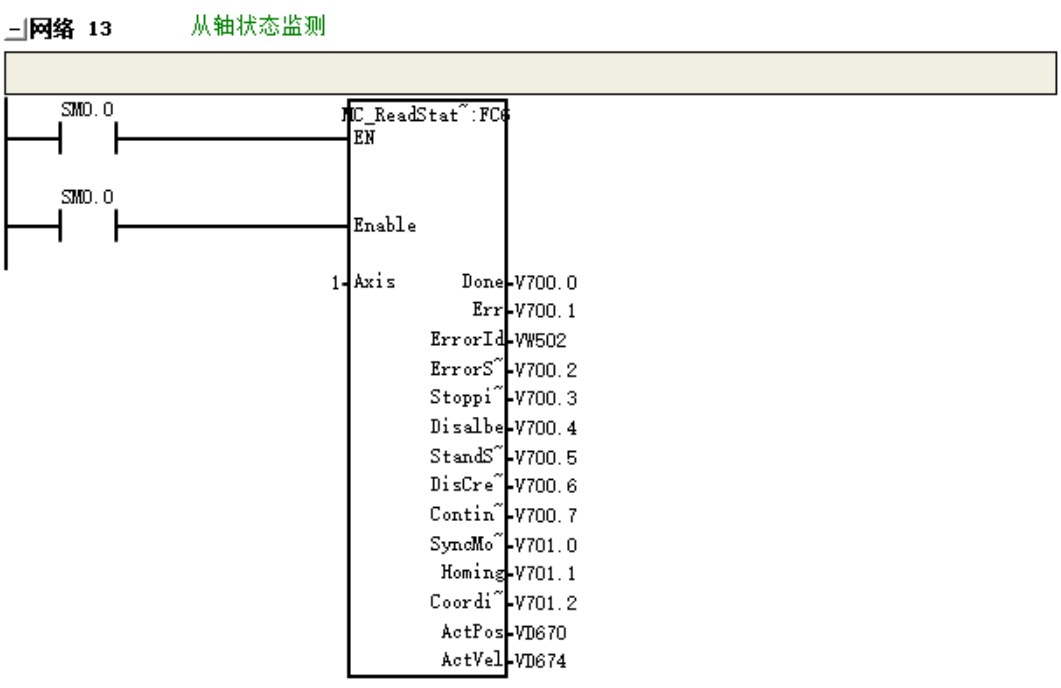
网络 10　　运行结束，退出凸轮控制，此时从轴按最后状态匀速运行，用其他指令处理停止过程。



网络 11　　主轴用速度指令匀速运行

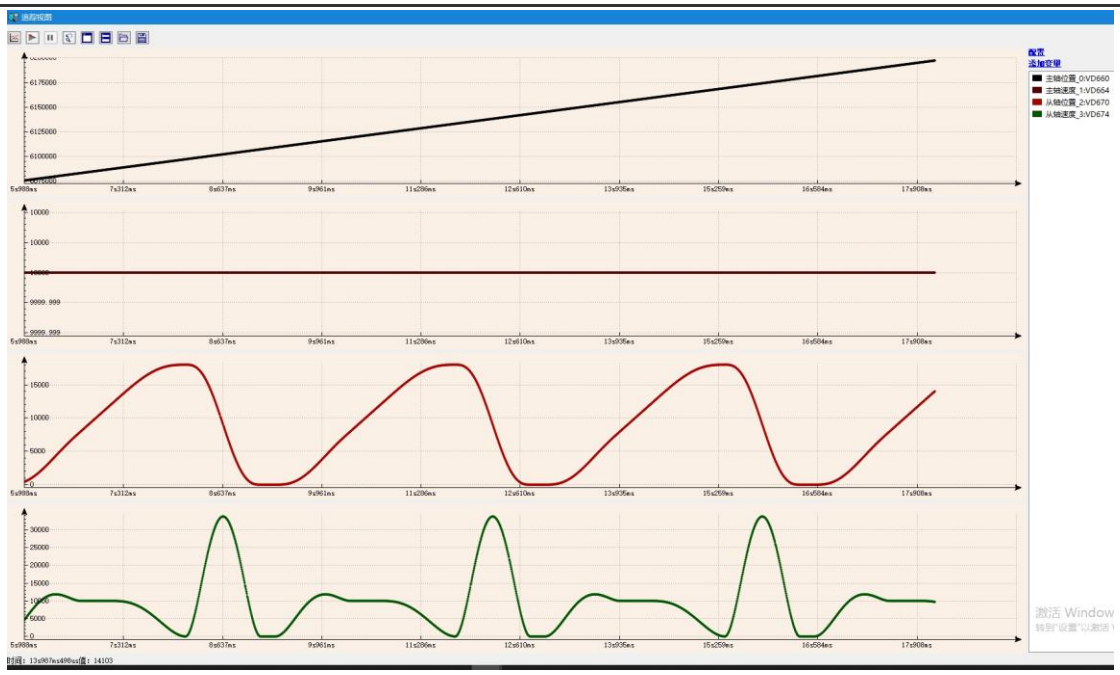






程序编写完进行编译，若编译无误后即可将程序下载到目标机。

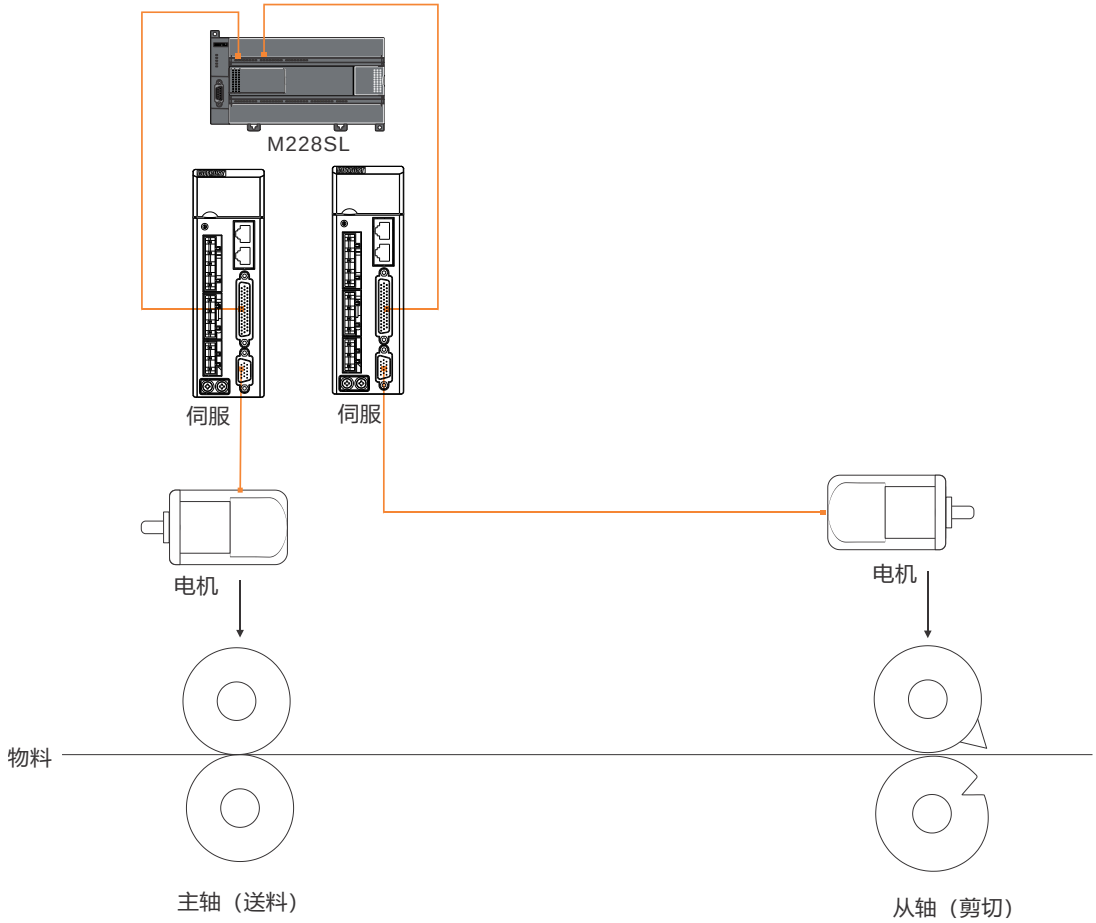
在应用程序运行时，可以在 MagicWorks PLC 的“追踪视图”中查看跟踪变量的值曲线记录。要求是设置跟踪配置，将跟踪配置传输到 PLC，并开始跟踪记录，追踪配置具体操作见 E Trace 追踪功能。追剪运行曲线如下所示：



4.2.13 飞剪功能介绍

飞剪是指对运动中的物料进行垂直切割的工艺，飞剪机构做周期性圆周运动，当切刀的线速度与物料速度一致时进行切割。

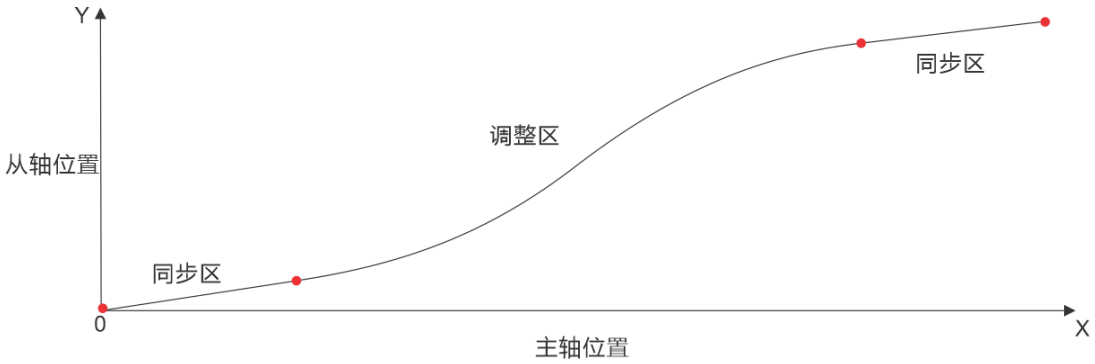
飞剪架构示意图如下：



根据物料的剪切长度，飞剪功能曲线可分为以下三种：

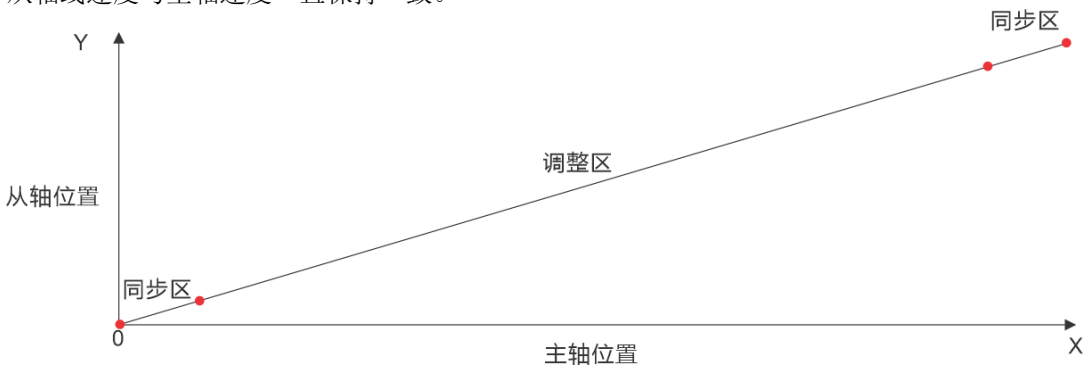
1）物料剪切长度小于刀轮走过的周长

同步区内从轴线速度与主轴速度相等，调整区内从轴进行速度调整，先加速后减速直到同步主轴速度。



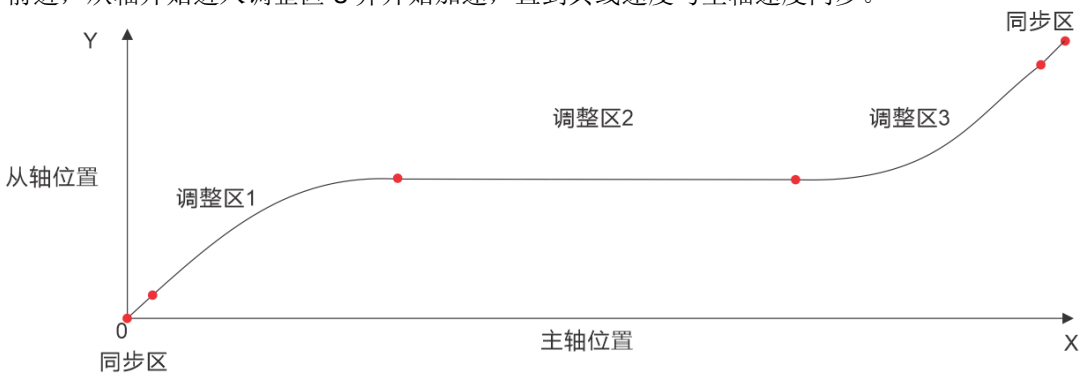
2）物料剪切长度等于刀轮走过的周长

从轴线速度与主轴速度一直保持一致。

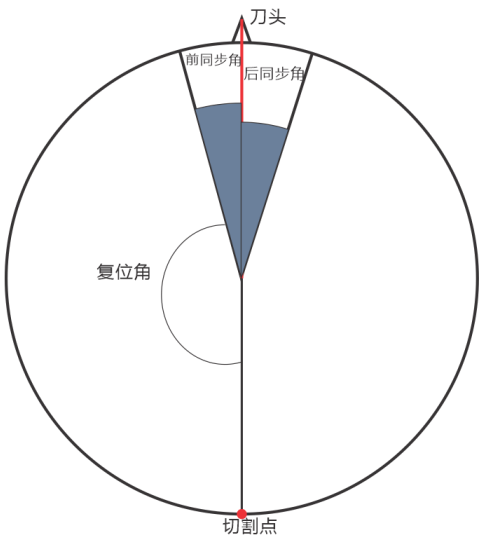


3）物料剪切长度大于刀轮走过的周长

首先从轴在调整区 1 中进行加速，由于剪切的物料过长，进入调整区 2 后从轴减速为 0；此时主轴持续前进，从轴开始进入调整区 3 并开始加速，直到其线速度与主轴速度同步。



从轴结构示意图如下：



剪切长度：即裁剪的长度。

复位角：指复位之后，切刀从原点按运行方向运行，前同步角边沿到达切割点经过的角度值。

前同步角：进入同步点，到切刀所在点之间的夹角。

后同步角：切刀点到退出同步区的点之间的夹角。

切刀数：从轴上的切刀数目。若切刀数目为两个或两个以上，要求切刀均匀角度安装，两两切刀之间的角度相同。

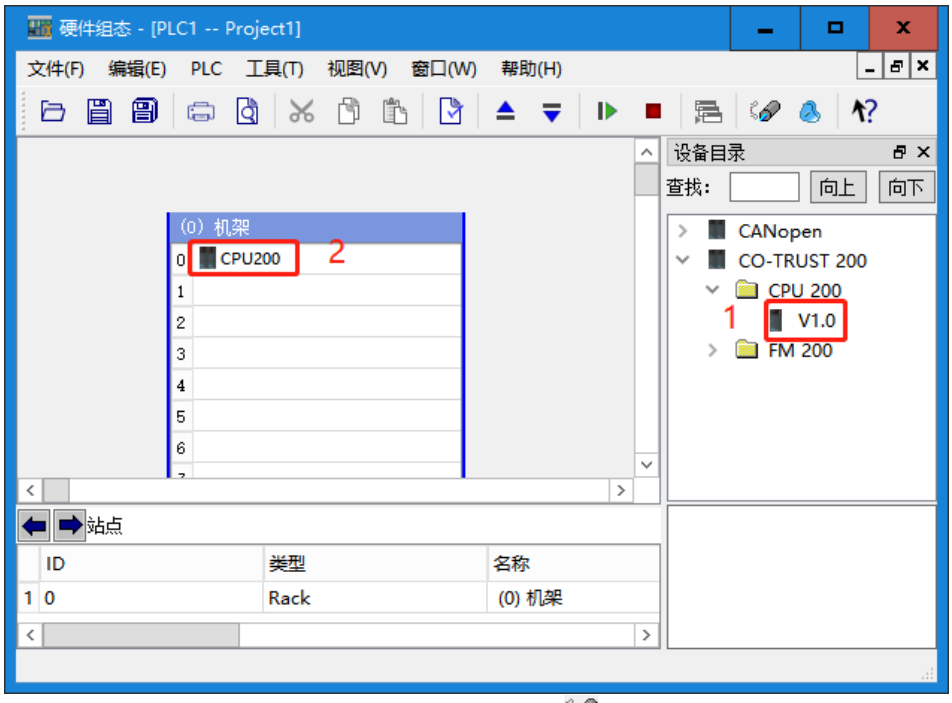
切刀半径：从轴圆心到刀头的距离即为切刀半径。

飞剪功能应用

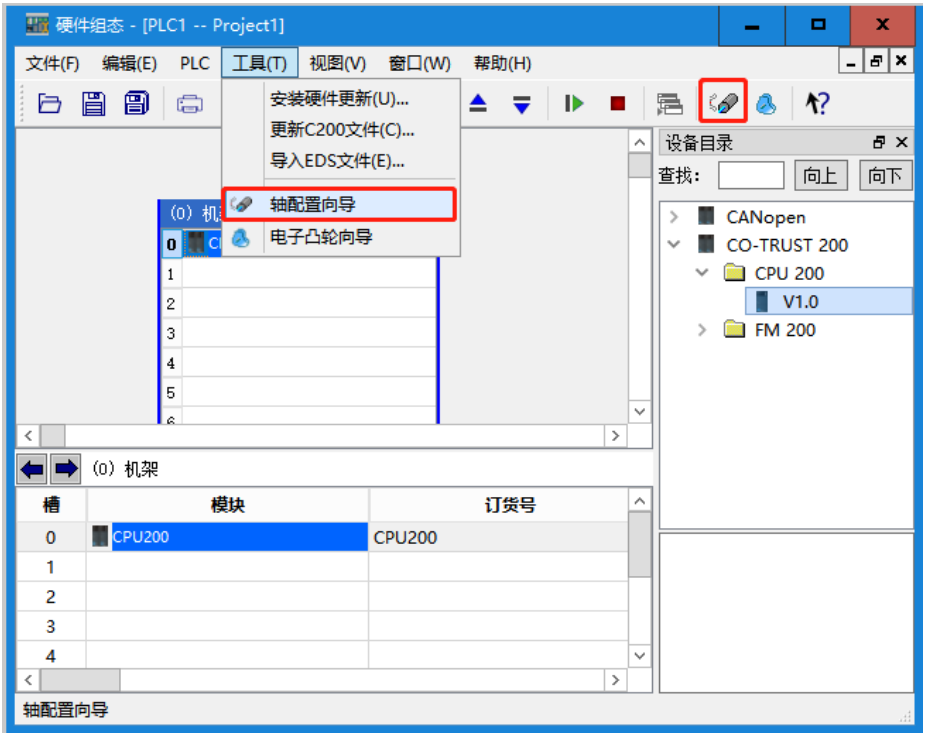
追飞剪功能的使用场景多应用于灌装行业，旋盖行业，瓦楞行业，枕式包装行业，瓶类丝网印刷行业，剪切行业、定长切纸行业，绕线行业，追标行业，同步盖印行业，追锯追剪行业等。Magicworks PLC V2.22 版本集成追剪及飞剪功能，可在硬件组态界面对追剪飞剪进行组态，设置追剪飞剪相关参数；通过 CAM 曲线对主轴位置与从轴位置、从轴速度、从轴加速度的关系进行预览。

1、新建飞剪

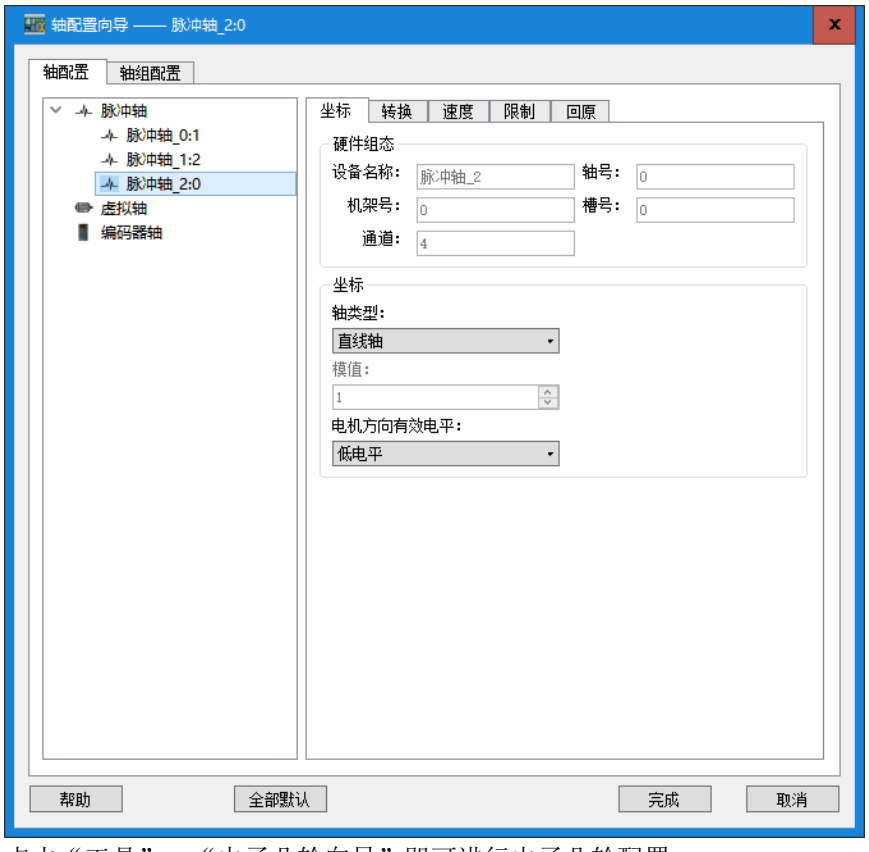
新建工程，在硬件组态中进行组态，选择 CPU200，将 CPU200 拖到机架上，CPU200 只能放在机架 0 上。



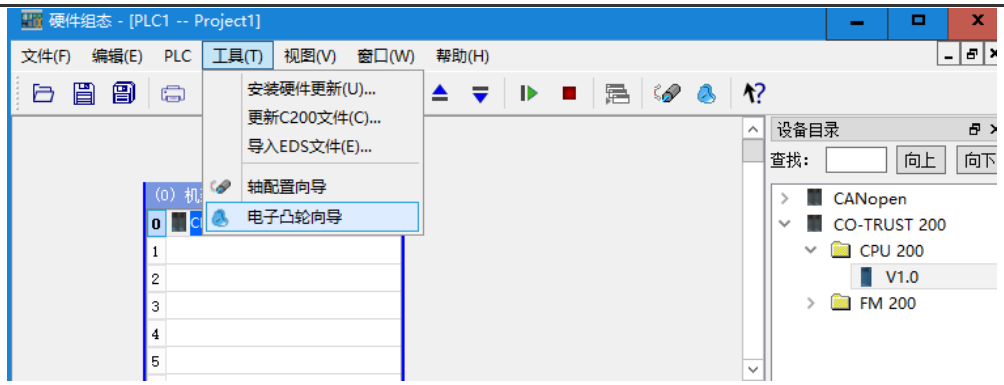
点击“工具”→“轴配置向导”或直接点击图标即可进行轴配置。



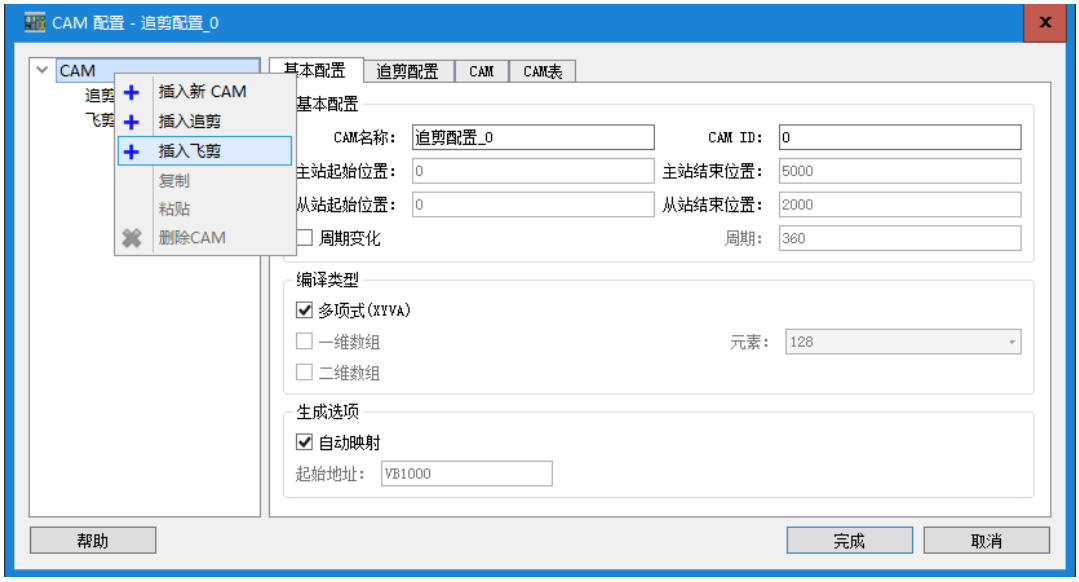
选择轴配置后，可任意添加脉冲轴、虚拟轴、编码器轴；选择“轴类型”及“电机方向有效电平”，点击“完成”。



点击“工具”→“电子凸轮向导”即可进行电子凸轮配置。

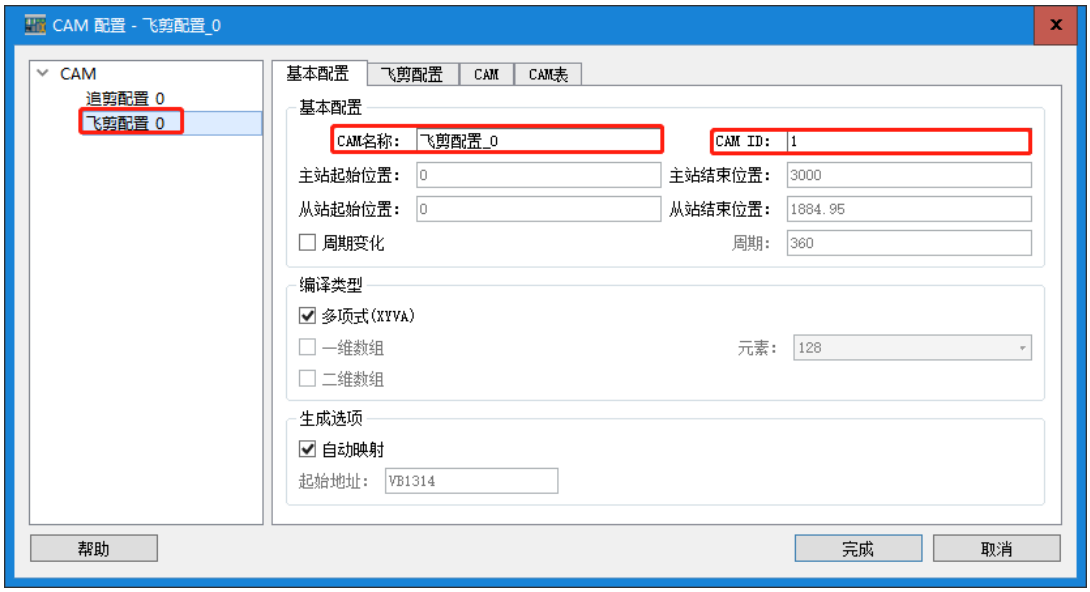


右键点击“CAM”，即可插入飞剪。



2、飞剪参数设置

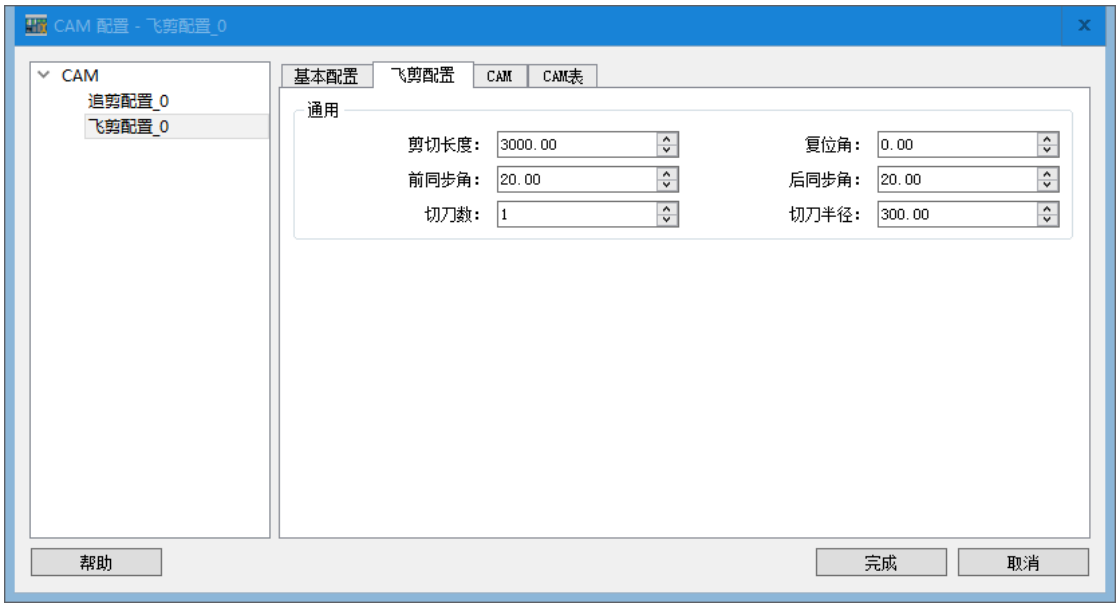
基本配置中，可以修改 CAM 名称以及 CAM ID



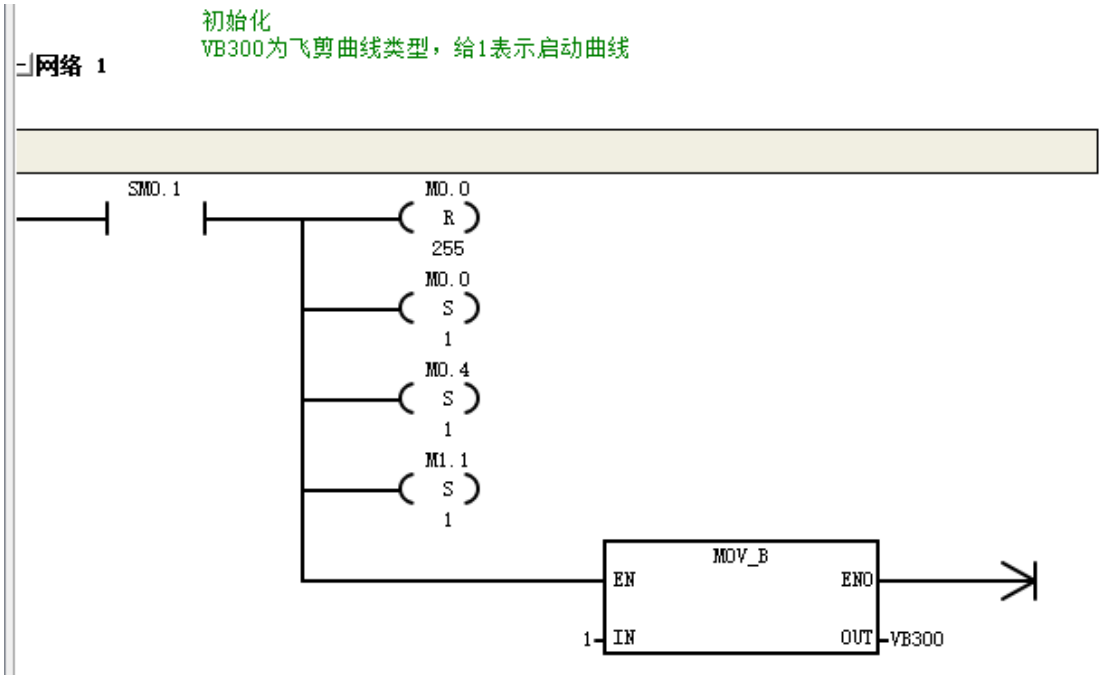
飞剪配置中可以设置物料剪切长度以及飞剪参数。

注意：复位角必须大于前同步角与后同步角的和。

物料后经过的同步区域角度为后同步角，目前规定前后同步角的大小要一致。

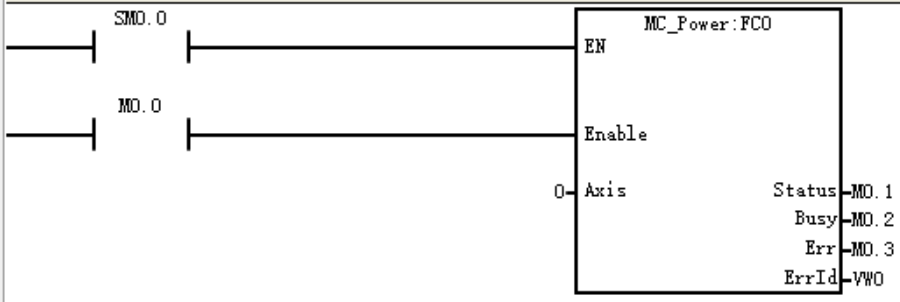

CAM 配置 - 飞剪配置_0
CAM
追剪配置_0
飞剪配置_0
基本配置
飞剪配置
CAM
CAM表
通用
剪切长度:
3000.00
复位角:
0.00
前同步角:
20.00
后同步角:
20.00
切刀数:
1
切刀半径:
300.00
帮助
完成
取消

3、程序示例：

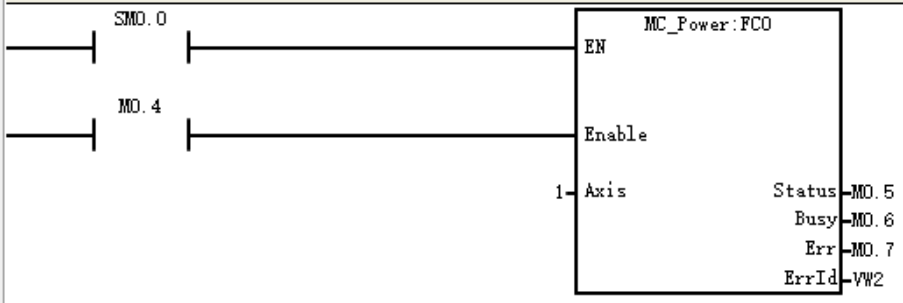

初始化
VB300为飞剪曲线类型，给1表示启动曲线
网络 1
SM0.1
M0.0
R
255
M0.0
S
1
M0.4
S
1
M1.1
S
1
MOV_B
EN
ENO
1
IN
OUT
VB300

网络 2　　主轴，轴0使能

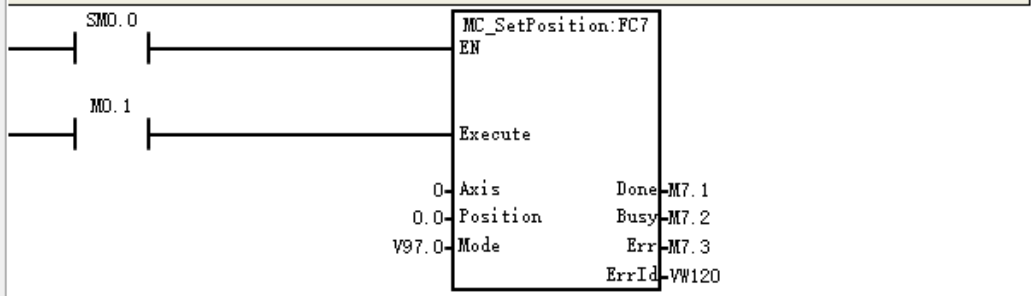
网络注释



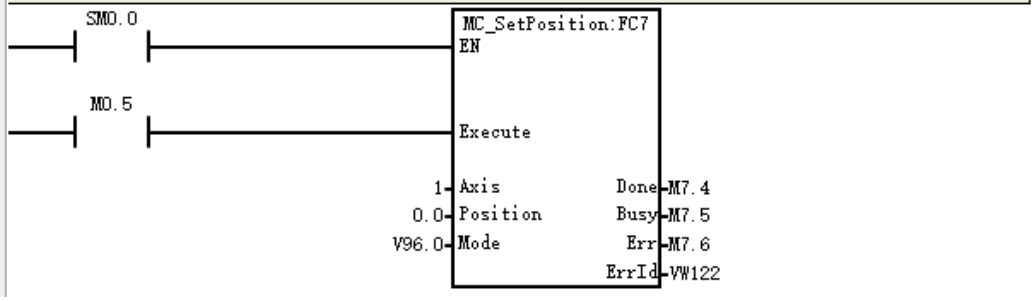
网络 3　　从轴，轴1使能



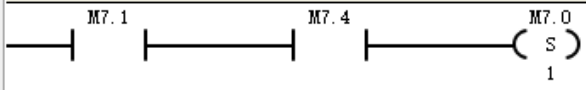
网络 4　　轴0复位，实际系统中需要根据原点传感器和回原指令完成这个过程



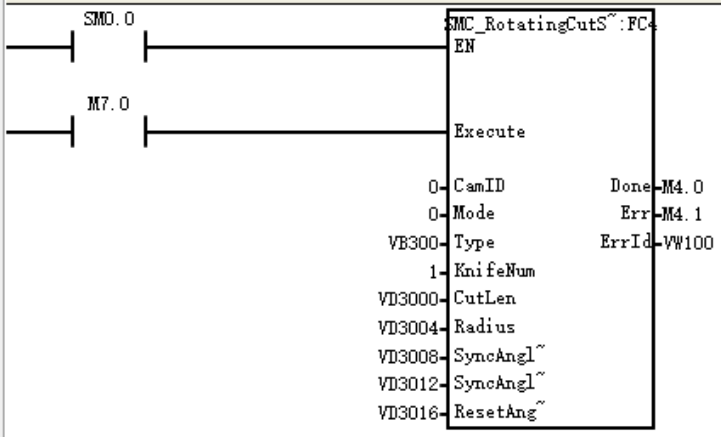
网络 5　　轴1复位，实际系统中需要根据原点传感器和回原指令完成这个过程



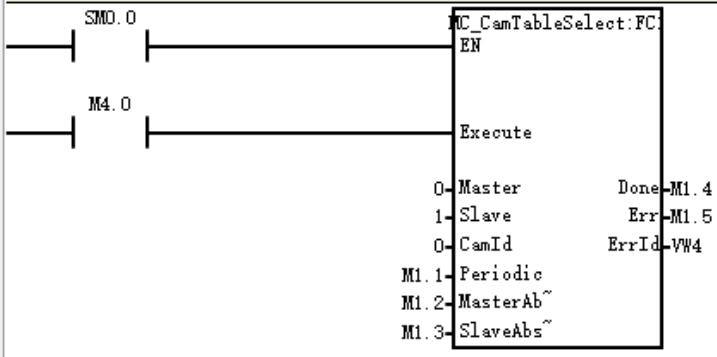
网络 6　当轴0复位完成，启动曲线使能，生成一条启动曲线
实际系统中，不一定是轴复位完成作为修改曲线的条件，根据实际情况确定



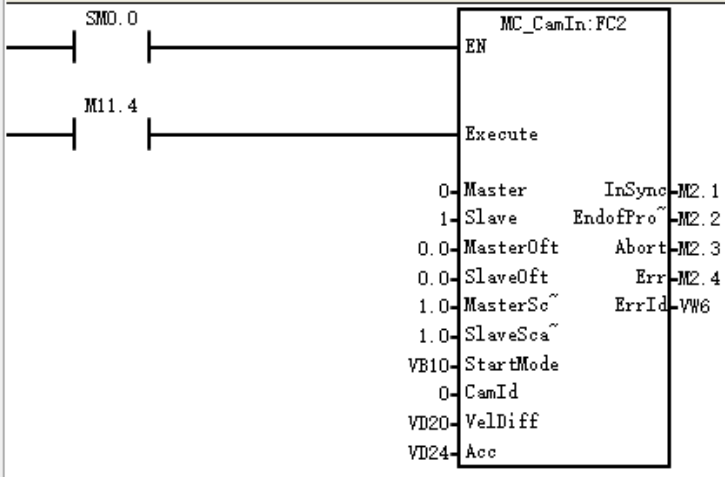
网络 7　调用旋切配置指令，指令会根据给定的参数，修改凸轮0的曲线
第一次运行VB300为1，生成一条启动曲线



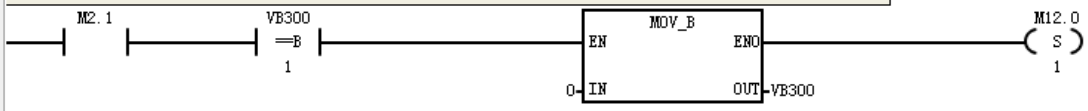
网络 8　绑定主从轴和凸轮的关系



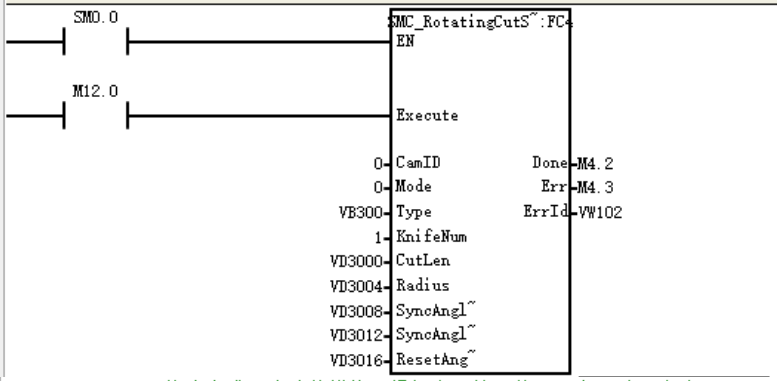
网络 9　启动凸轮，一般要求主轴达到匀速运行之后才执行该指令。
实际系统需要根据机器条件来执行。



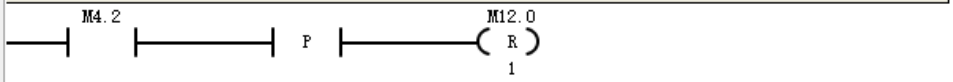
网络 10 启动曲线进入同步区之后，切换工作曲线，将曲线类型VB300改为0，表示工作曲线
当启动曲线运行完成之后会自动切换到工作曲线



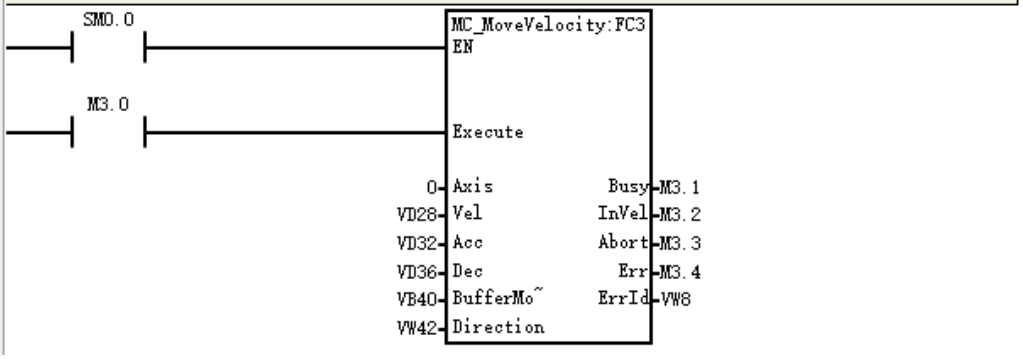
网络 11 调用旋切配置指令，将会根据VB300的值和新的参数生成一条新的曲线
并且新的曲线会在凸轮完成当前周期之后自动切换到这条曲线上。

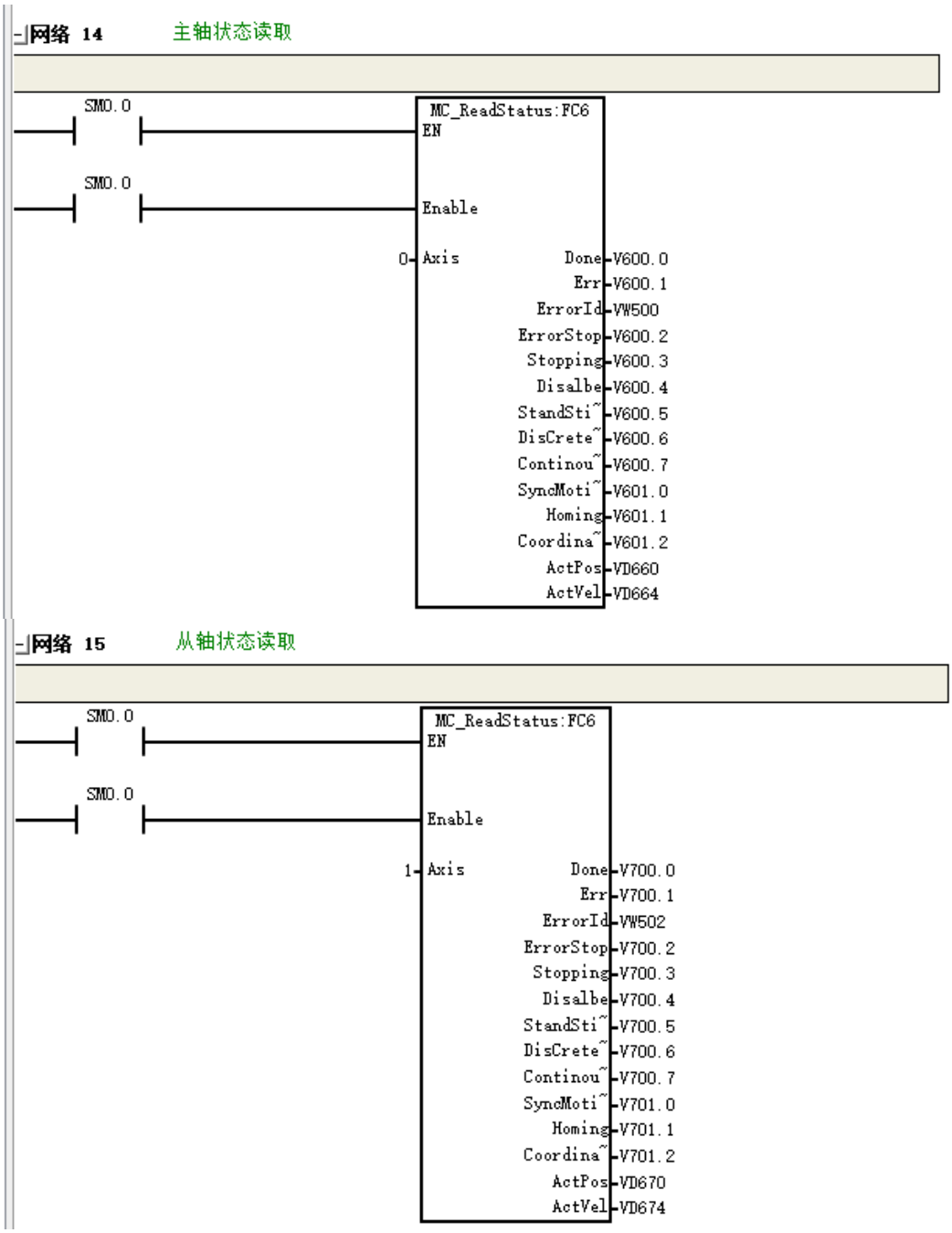


网络 12 修改完成，清除使能位，根据实际情况使用，也可以不清除。



网络 13 主轴按速度模式运行，首先启动主轴，达到匀速之后启动旋切功能

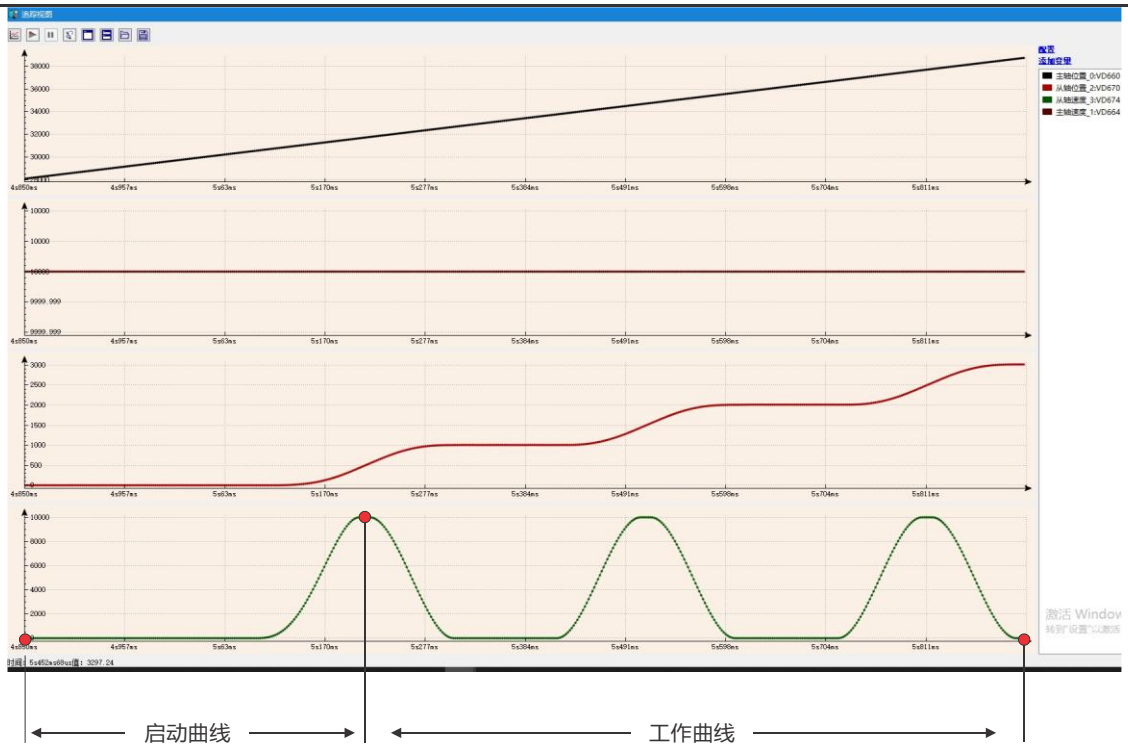




程序编写完进行编译，若编译无误后即可将程序下载到目标机。

在应用程序运行时，可以在 MagicWorks PLC 的“追踪视图”中查看跟踪变量的值曲线记录。要求是设置跟踪配置，将跟踪配置传输到 PLC，并开始跟踪记录，追踪配置具体操作见《MagicWorks PLC 用户手册》，下载地址为 http://www.co-trust.com。

飞剪运行曲线如下所示：



4.2.14 回原功能介绍

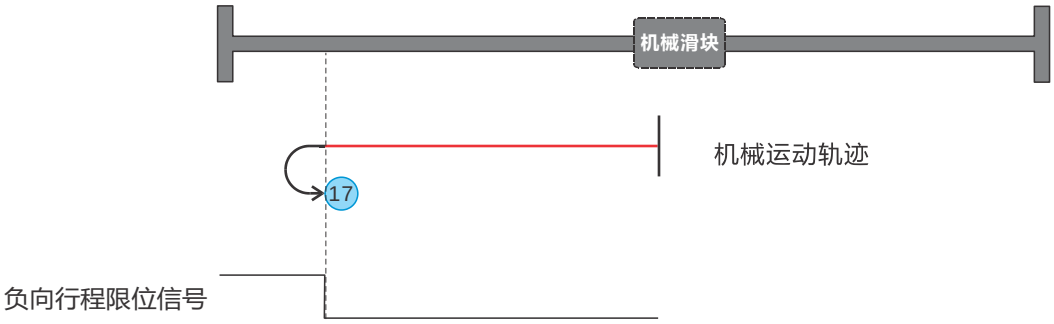
回原模式原理介绍

用户可根据各自身对精度的要求及实际应用需求从以下 15 种回原模式进行选择。

模式	说明
17	参考负向行程限位信号
18	参考正向行程限位信号
19	参考正向原点信号开关
20	参考正向原点信号开关
21	参考负向原点信号开关
22	参考负向原点信号开关
23	参考原点开关和正限位的原点模式
24	参考原点开关和正限位的原点模式
25	参考原点开关和正限位的原点模式
26	参考原点开关和正限位的原点模式
27	参考原点开关和负限位的原点模式
28	参考原点开关和负限位的原点模式
29	参考原点开关和负限位的原点模式
30	参考原点开关和负限位的原点模式
35	设置当前位置为原点

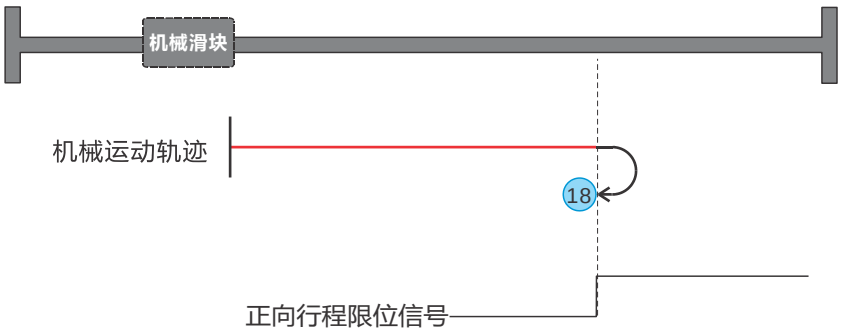
无论机械初始处于什么位置，当设备（原点开关、正向行程限位开关、负向行程限位开关）安装完好，伺服所寻找的设备原点总是唯一的。以下各模式示意图中的竖线“|”代表机械初始位置，圆圈“Ⓧ”代表原点位置。“—”代表搜索速度，“—”代表爬行速度

回原模式 17：参考负限位下降沿



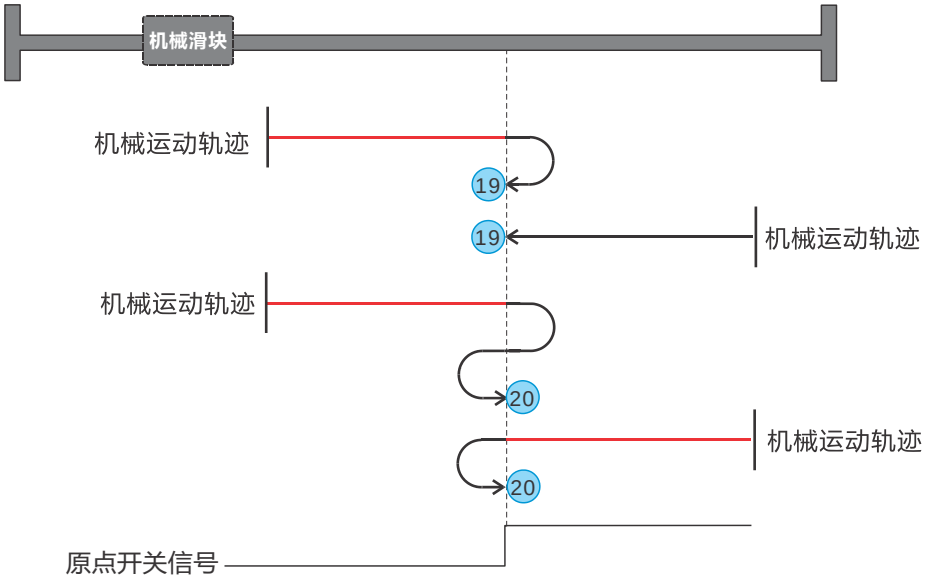
回原开始时，负向行程限位信号为 0，反向高速进行回原，遇到行程限位信号上升沿后，进行减速并反向，低速运行直到遇到行程限位信号下降沿停机，此位置即为原点位置。

回原模式 18：参考正限位下降沿



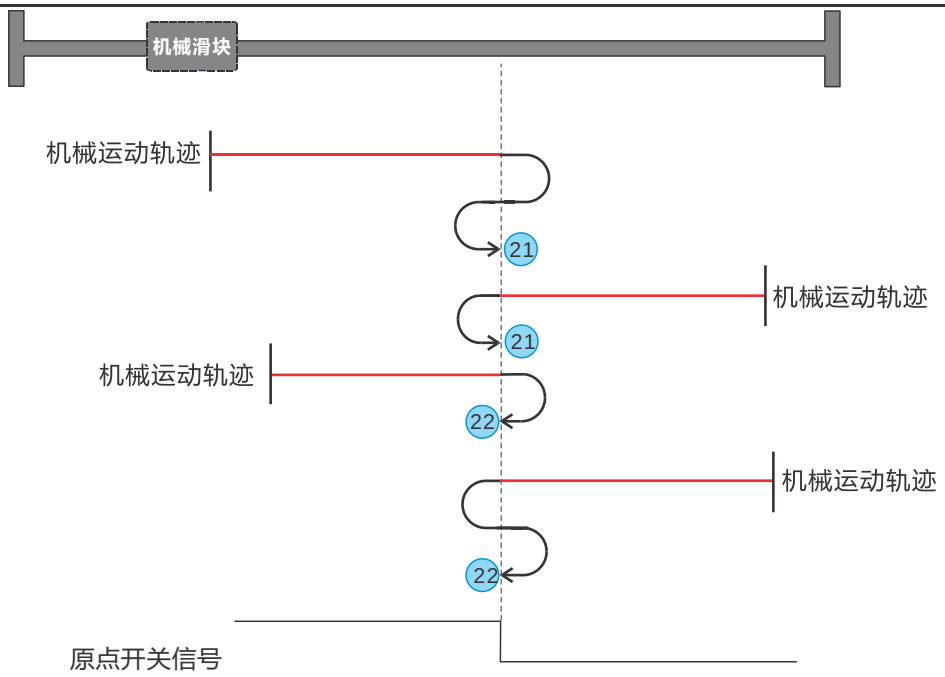
回原开始时，正向行程限位信号为 0，正向高速进行回原，遇到行程限位信号上升沿后，进行减速并反向，并低速运行，直到遇到行程限位信号下降沿停机，此位置即为原点位置。

回原模式 19、20：参考正向原点信号开关



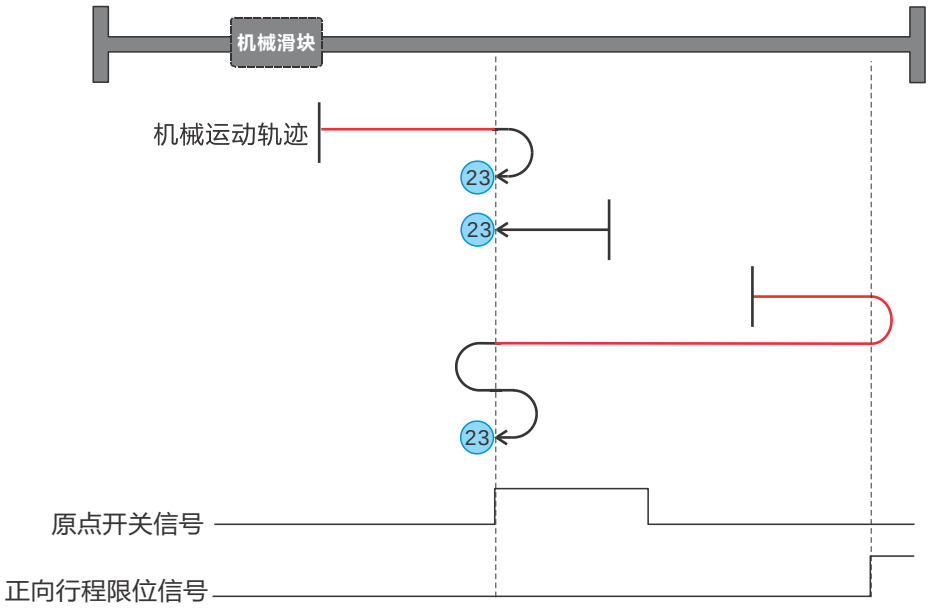
起始运动方向取决于原点开关信号的高位与低位状态，模式 19 的原点在原点开关信号由高位变为低位的下降沿位置；模式 20 的原点在原点开关信号由低位变为高位的上升沿位置。

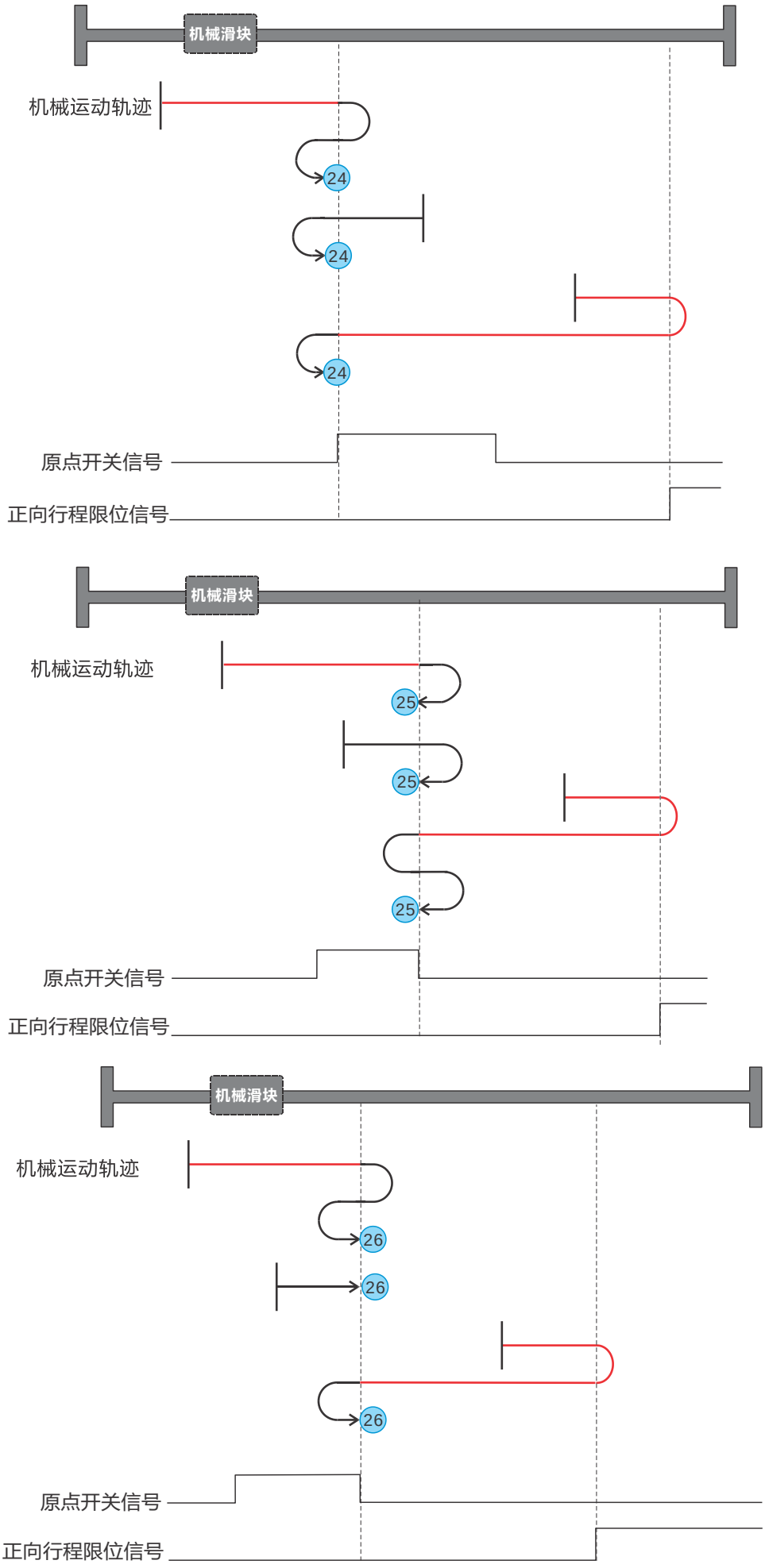
回原模式 21、22：参考负向原点信号开关



起始运动方向取决于原点开关信号的高位与低位状态，模式 21 的原点在原点开关信号由高位变为低位的下降沿位置；模式 22 的原点在原点开关信号由低位变为高位的上升沿位置。

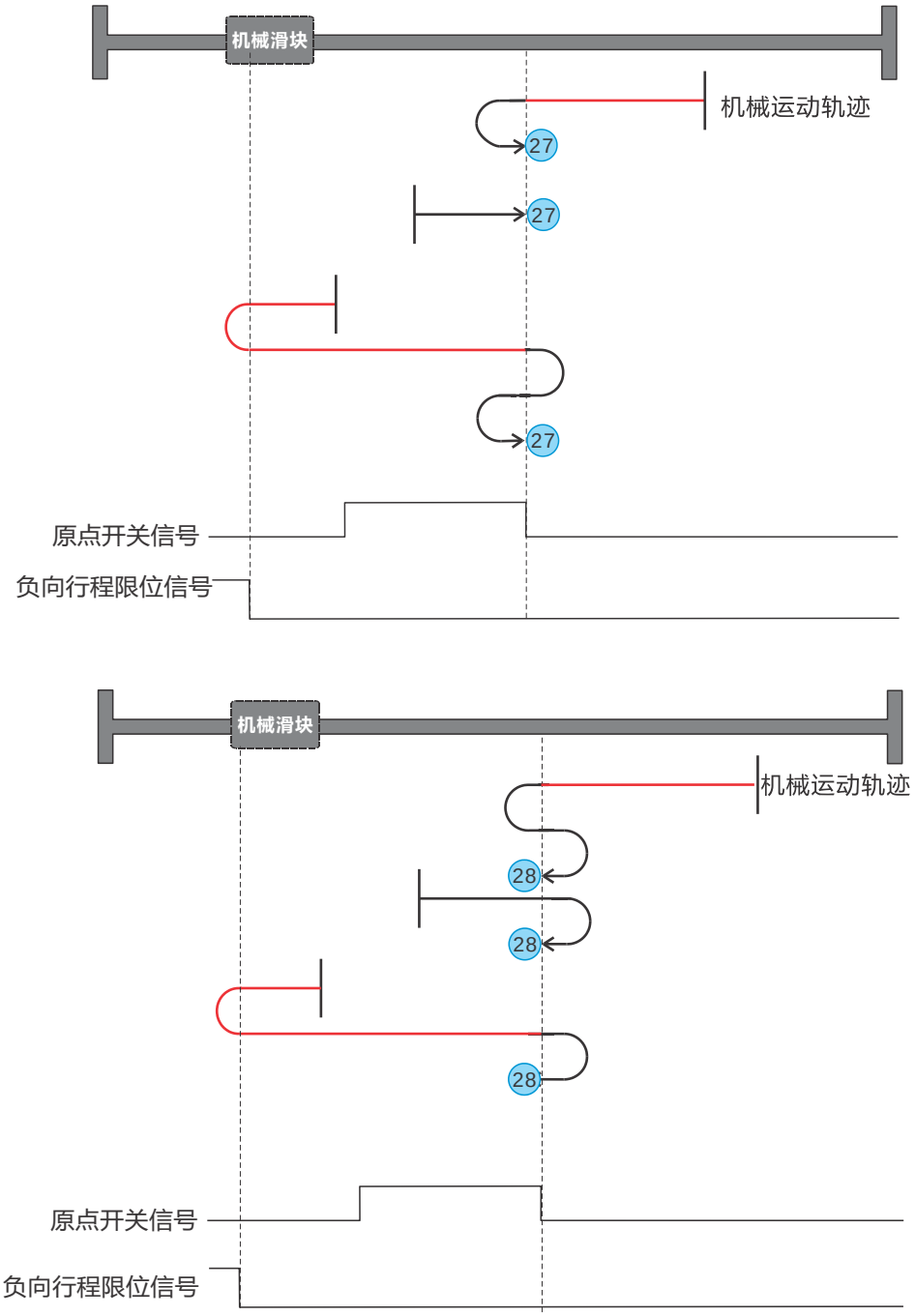
回原模式 23~26：参考原点信号开关和正向行程限位

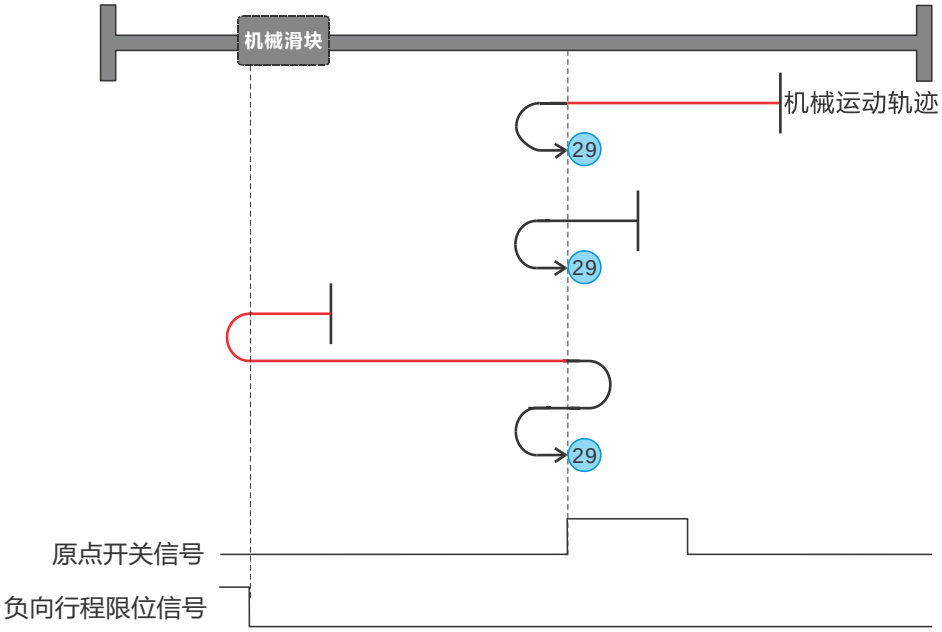


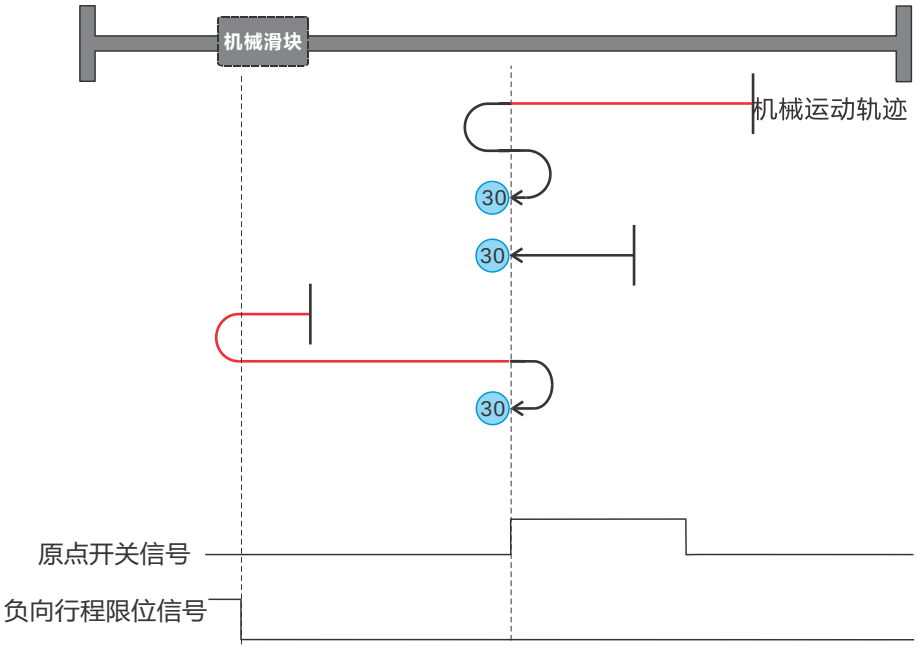

机械滑块
机械运动轨迹
24
24
24
原点开关信号
正向行程限位信号
机械滑块
机械运动轨迹
25
25
25
原点开关信号
正向行程限位信号
机械滑块
机械运动轨迹
26
26
26
原点开关信号
正向行程限位信号

起始运动方向取决于原点开关信号和正向行程限位信号的状态，模式 23 和 26 的原点在原点开关信号由高位变为低位的下降沿位置；模式 24 和 25 的原点在原点开关信号由低位变为高位的上升沿位置。

回原模式 27~30：参考原点信号开关和负向行程限位







起始运动方向取决于原点开关信号和负向行程限位信号的状态，模式 27 和 30 的原点在原点开关信号由高位变为低位的下降沿位置；模式 28 和 29 的原点在原点开关信号由低位变为高位的上升沿位置。

回原模式 35：以当前位置为原点

电机停在当前位置不动，给定回原信号即完成回原。

回原功能应用举例

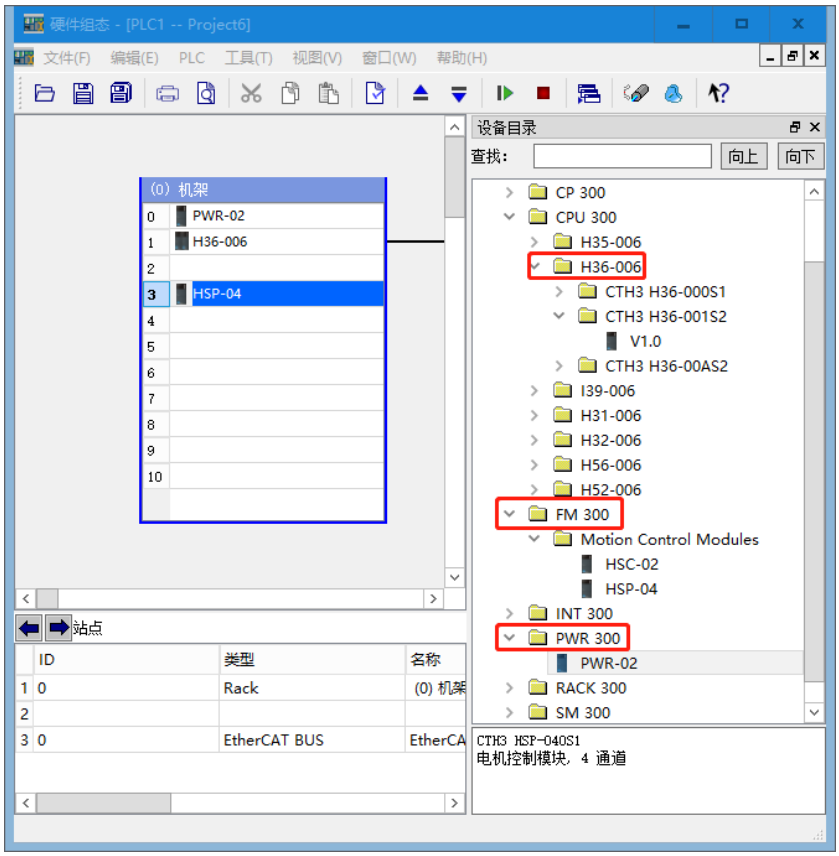
下面以 H36-006 CPU 为例对回原功能进行具体介绍。

示例 1：脉冲轴原点回归示例

在 H36-006 CPU 站点的项目管理器界面选择“硬件组态”，进行硬件组态。

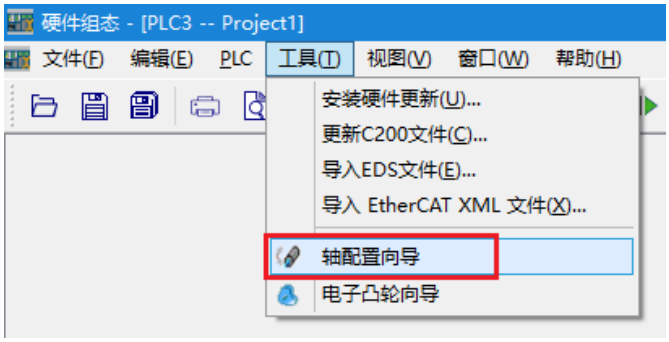
步骤一：添加电源模块、CPU 和 HSP 模块

硬件组态界面中，在“CO-TRUST 300”项目树下顺序展开“PWR 300”、“CPU 300”和“FM 300”文件夹，添加电源模块、CPU 和 HSP 模块，电源模块只能放至机架的 0 号槽内，CPU 只能放至机架的 1 号槽内，HSP 模块可放在 3-10 号卡槽内，如下图所示。

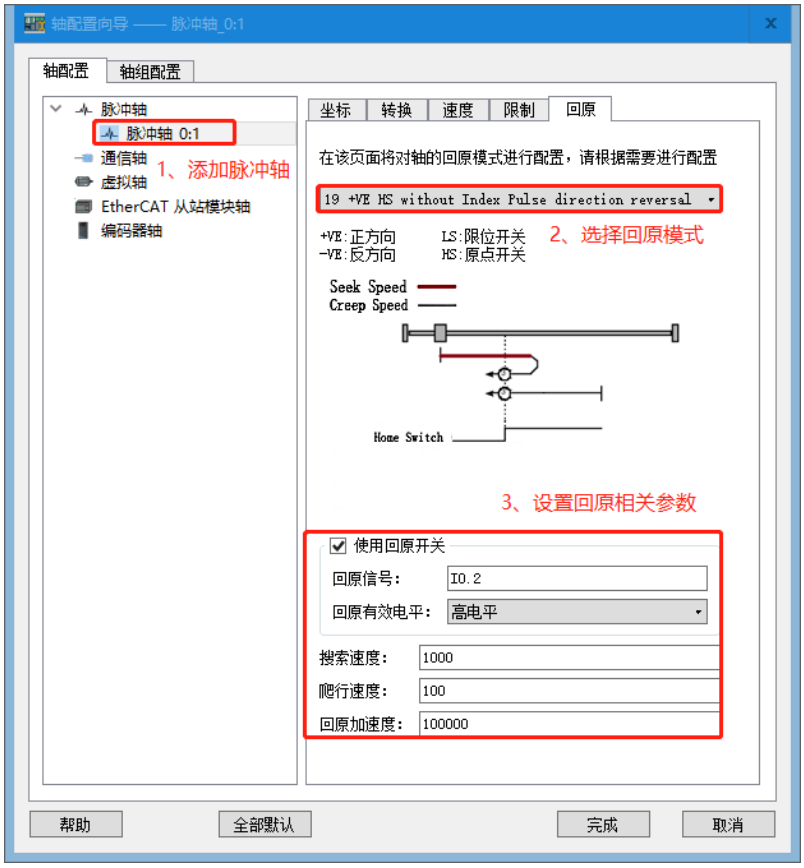


步骤二：配置轴

在硬件组态界面选择“工具”→“轴配置向导”，添加脉冲轴，即可进行轴配置，如下图所示。

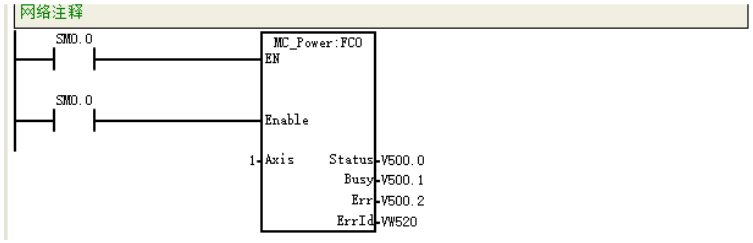


点击“回原”，在此页面可对轴的回原模式进行配置，此配置仅脉冲轴支持，选择下拉选项可查看回原模式，共有 17 种回原模式可选。

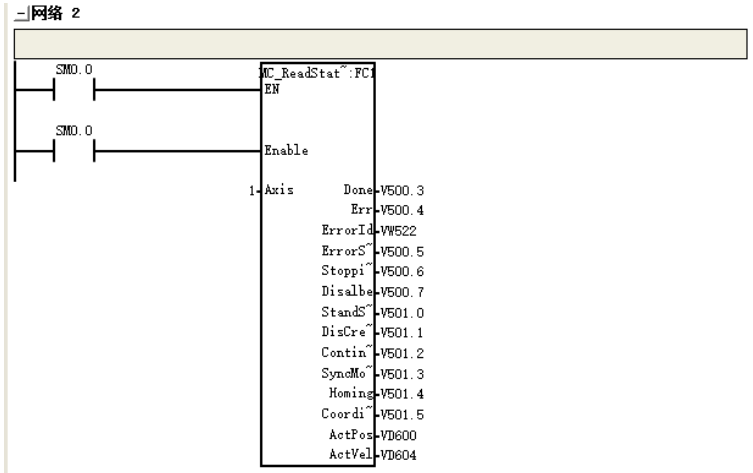


步骤三：在主程序中调用 MC_Home 指令（原点回归指令）进行编程。

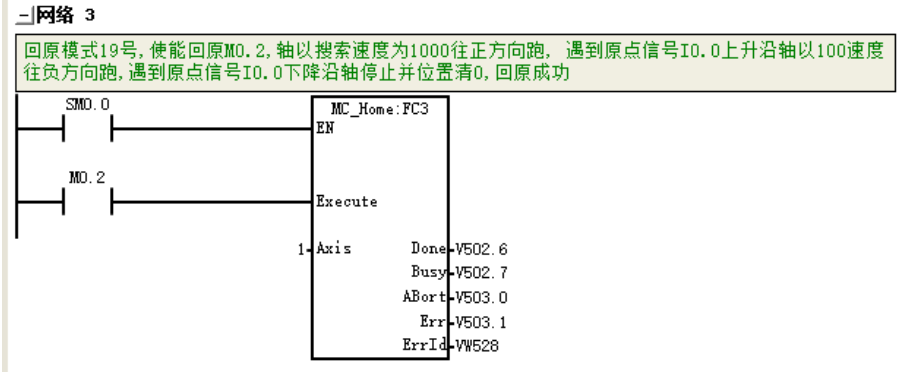
网络 1：对轴进行使能操作，轴 ID 号 Axis 设置为 1。



网络 2：读取轴状态



网络 3：回原模式为 19 号，回原信号为 I0.0，轴的搜索速度为 1000。

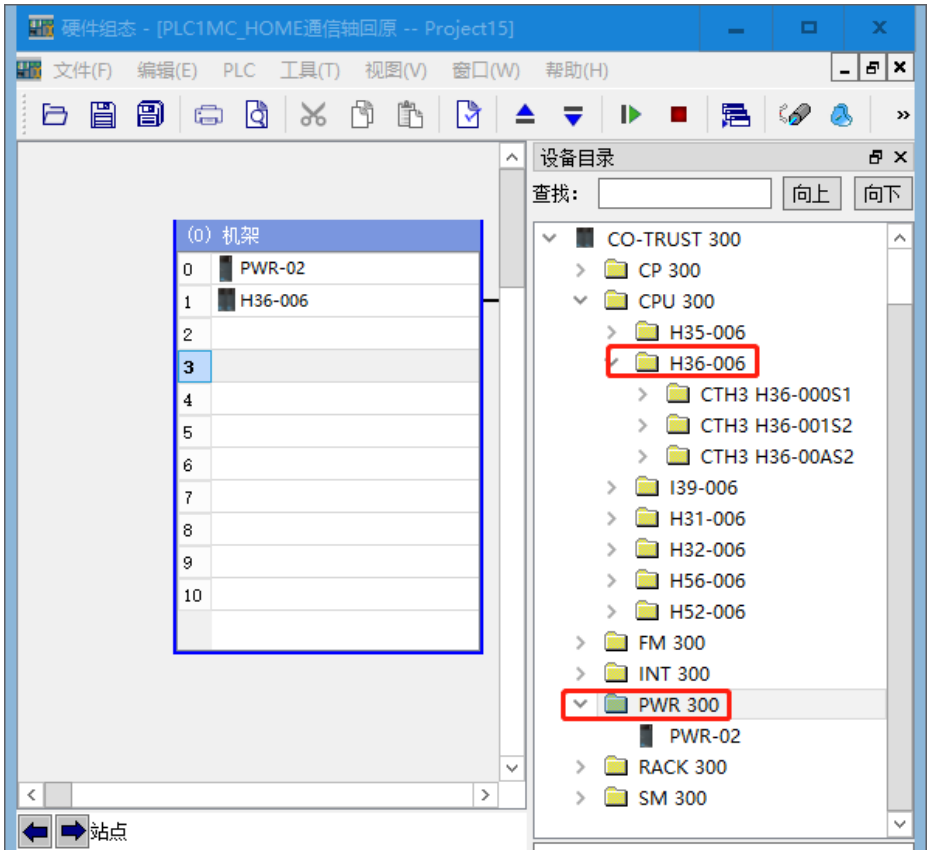


示例 2：通信轴原点回归示例

在 H36-006 CPU 站点的项目管理器界面选择“硬件组态”，进行硬件组态。

步骤一：添加电源模块、CPU

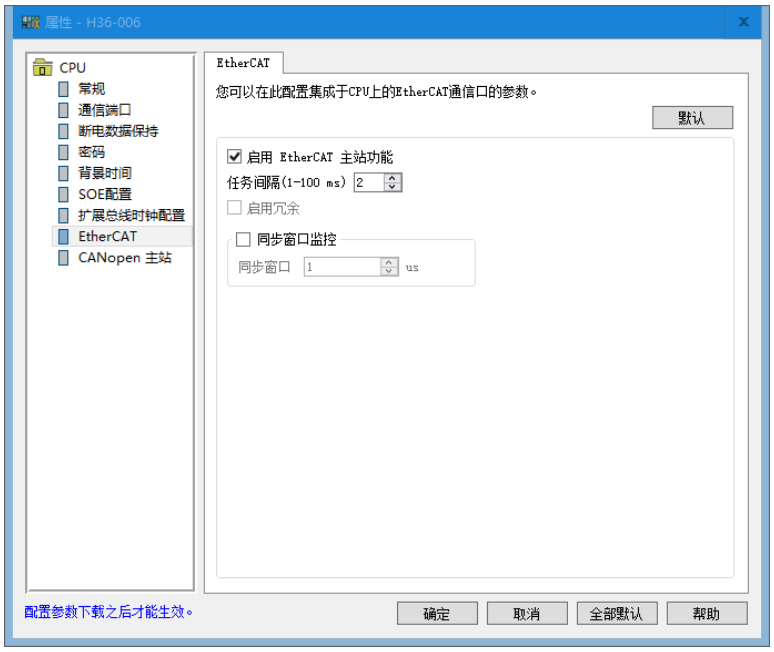
硬件组态界面中，在“CO-TRUST 300”项目树下顺序展开“PWR 300”和“CPU 300”文件夹，添加电源模块、CPU，电源模块只能放至机架的 0 号槽内，CPU 只能放至机架的 1 号槽内，如下图所示。



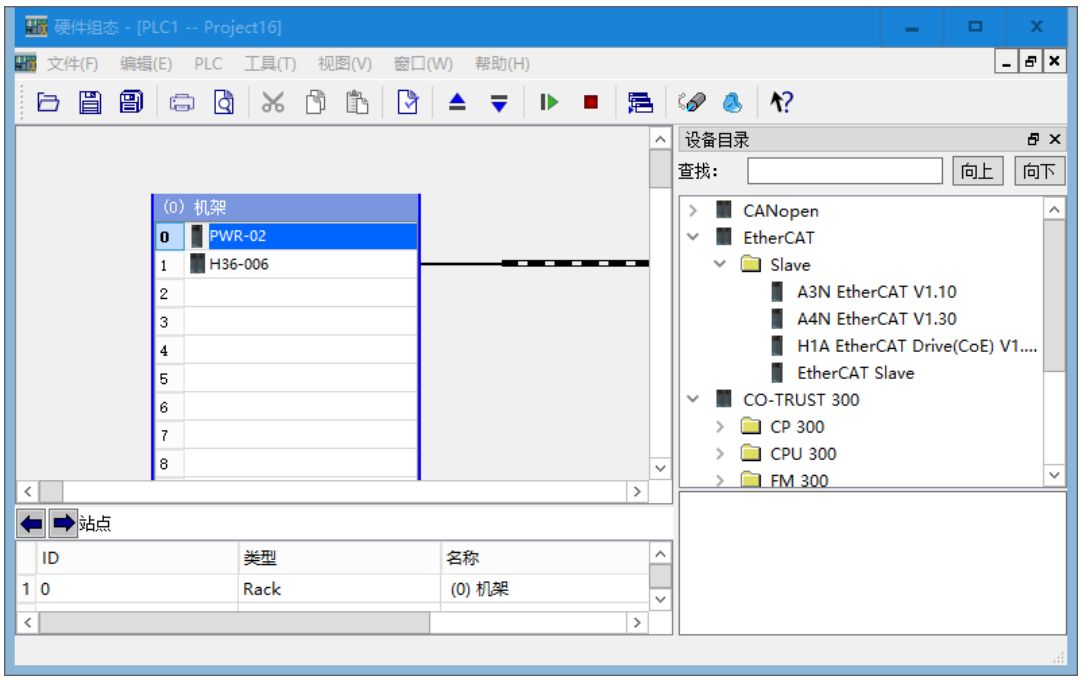
步骤二：配置轴

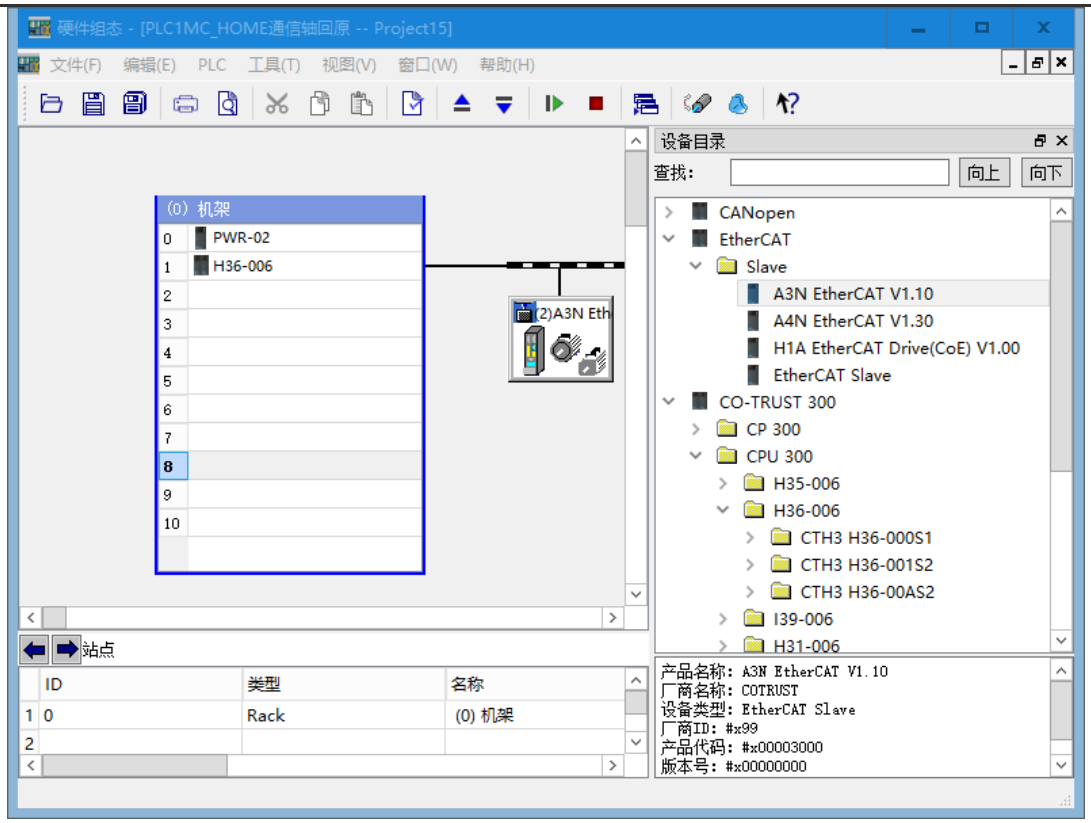
添加通信轴前需在硬件组态界面添加相应的 EtherCAT 从站，否则将无法添加通信轴。具体操作如下：

1）右击机架上的 H36，选择“属性”→ “EtherCAT”，点击页面中的“启用 EtherCAT 主站功能”。

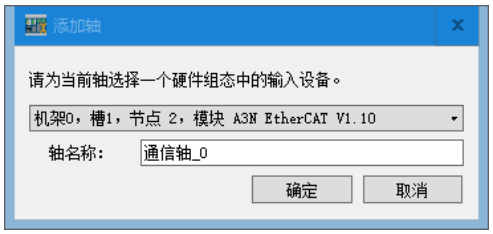


2）确定后可看到硬件组态界面的“EtherCAT BUS”，点击右边项目树下的“EtherCAT”，选择从站并将从站拖到“EtherCAT BUS”下。



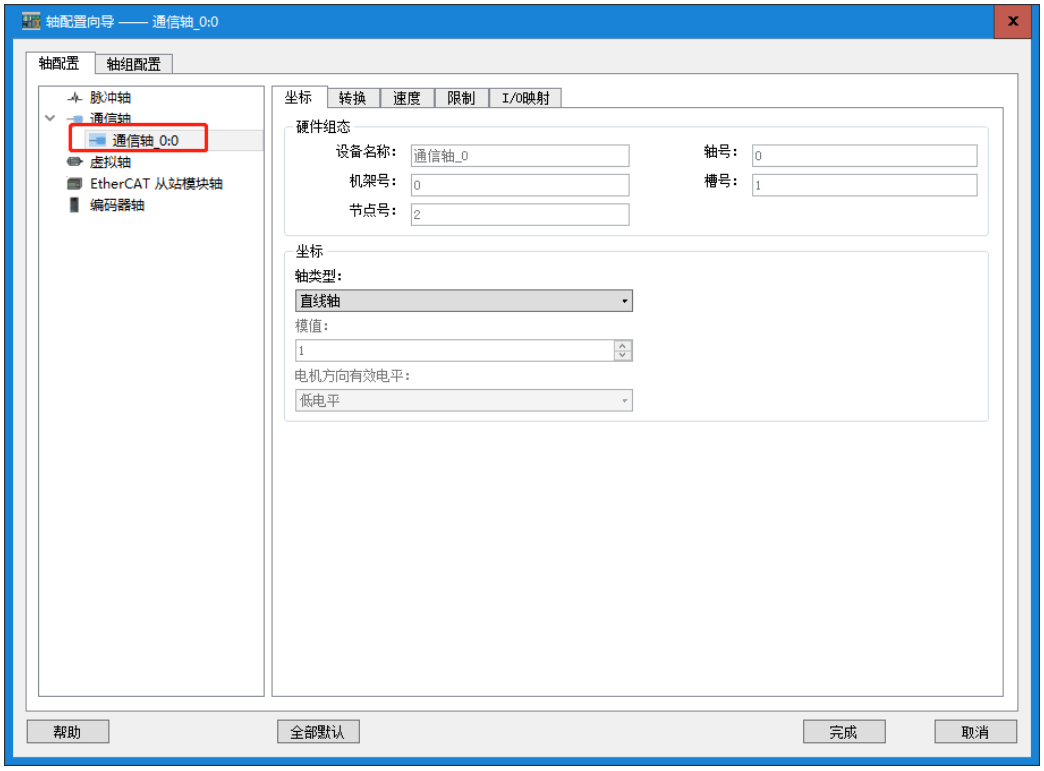


3）完成以上配置后在“工具”选项中选择“轴向导配置”，在轴向导配置界面双击“通信轴”，在弹出的“添加轴”界面中可为该轴选择输入设备，可选设备为硬件组态中所添加的从站。“轴名称”可修改，点击“确定”完成添加。



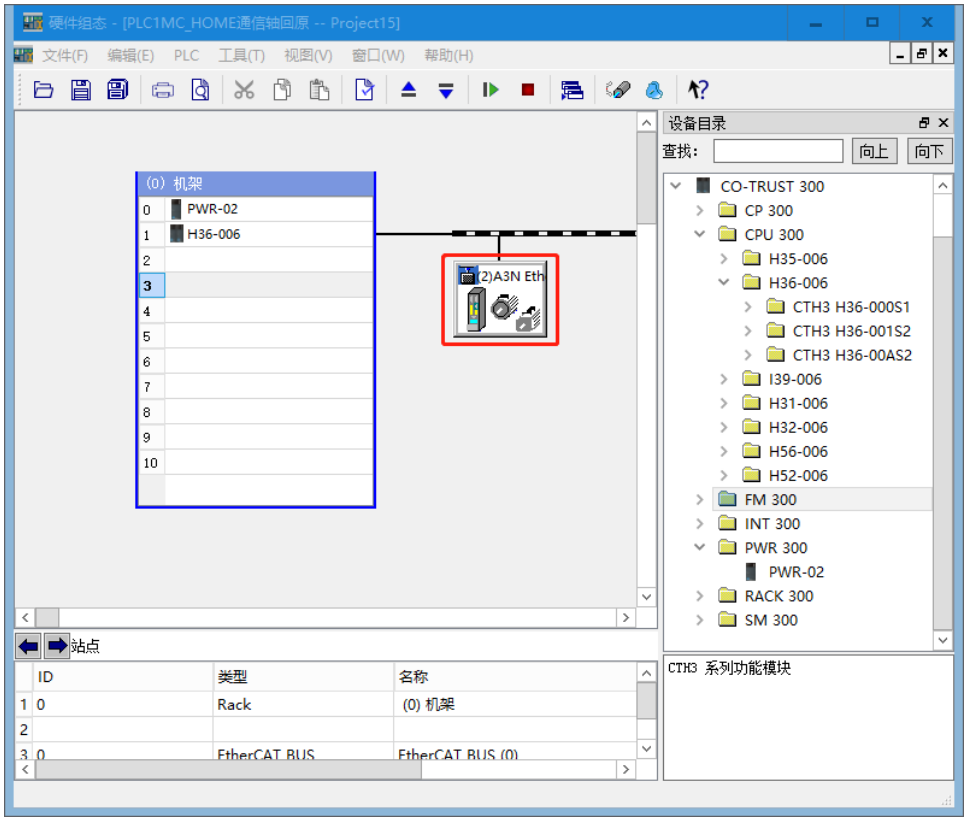
注意：一个从站只可对应添加一个通信轴。

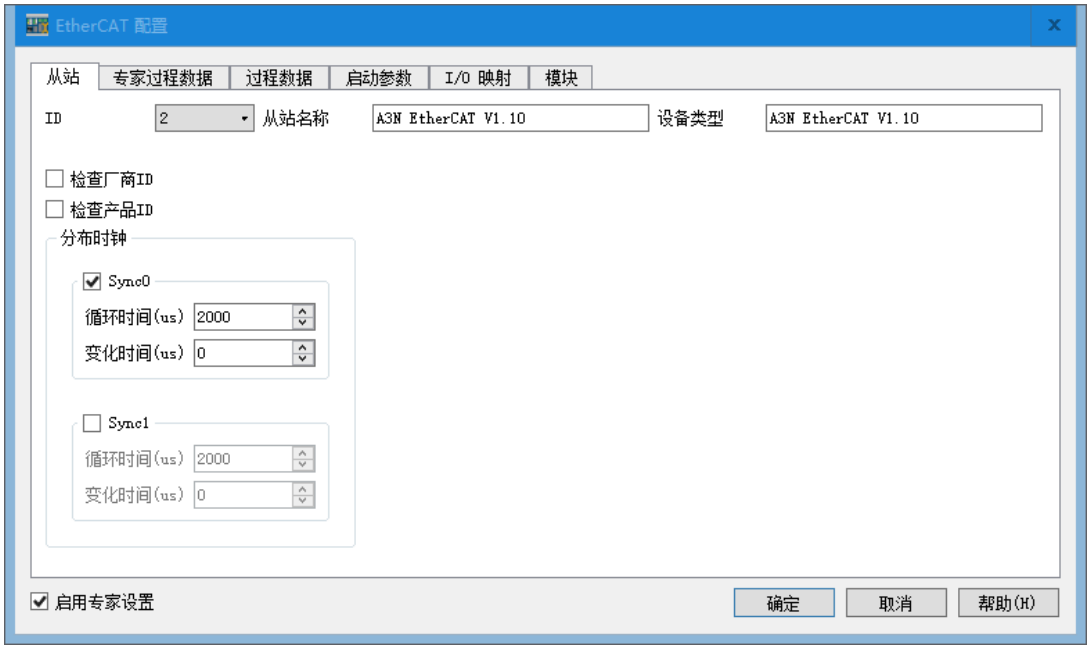
4）添加的通信轴界面如下所示：



5）使用驱动器回原（MC_Home 指令），在配置通信轴回原功能时，需配置 16#6098:0（回原模式）、16#6099:1（回原搜索速度）、16#6099:2（回原爬行速度）；操作如下：

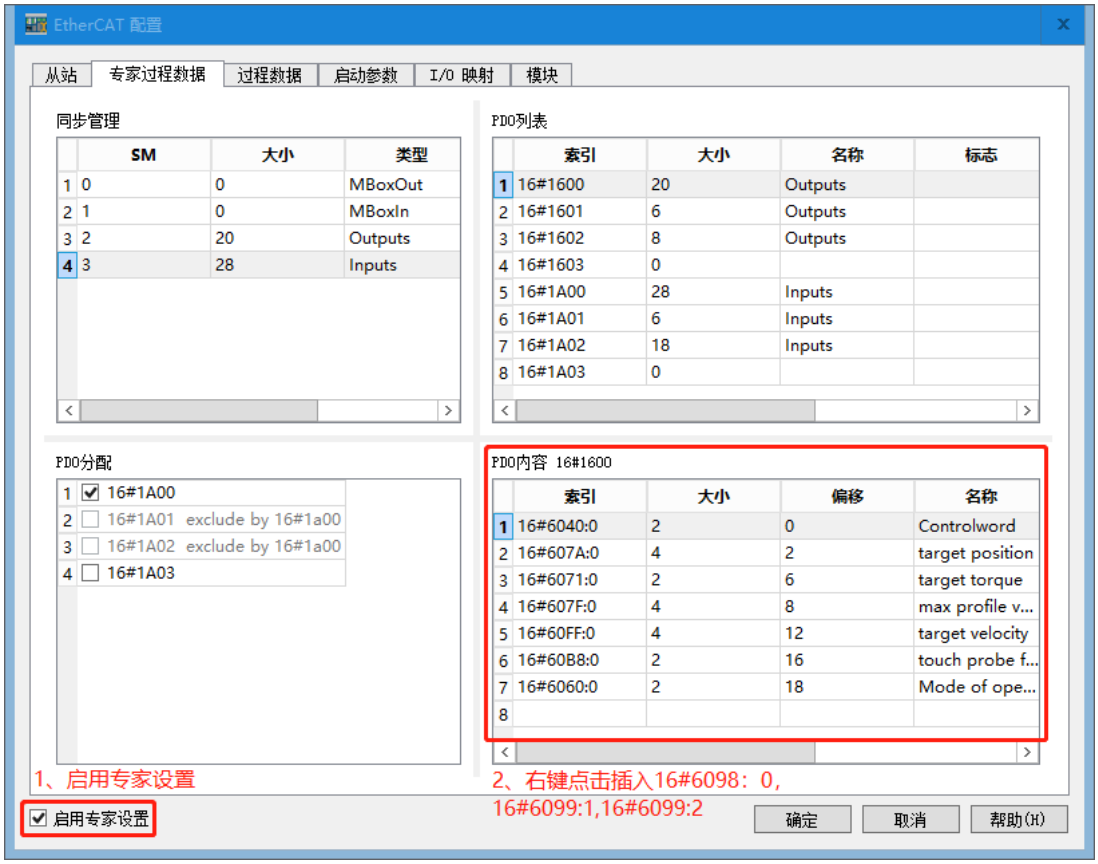
首先双击挂接在“EtherCAT BUS”下的从站，出现 EtherCAT 配置界面。





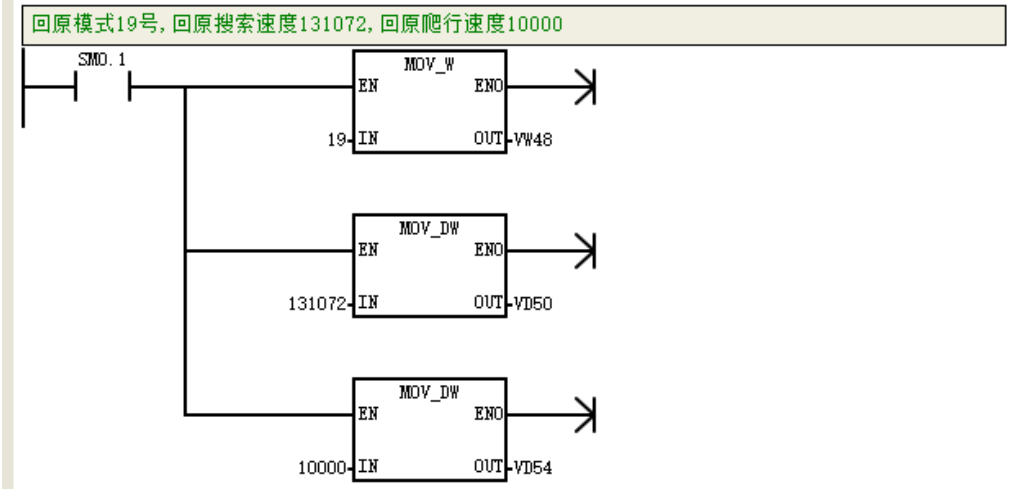
6）点击“启用专家设置”，右键点击后插入 16#6098:0（回原模式）、16#6099:1（回原搜索速度）、16#6099:2（回原爬行速度）。

<备注> 回原爬搜索速度与爬行速度的指令单位有两种，分别为：pps(脉冲数/每秒)、rpm(转数/每分)请参阅驱动器厂商说明书，具体请查阅驱动器厂商说明书。

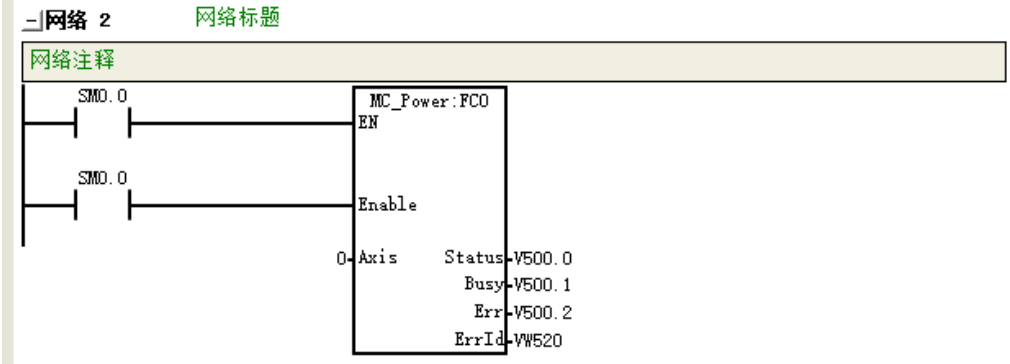


步骤三：在主程序中调用 MC_Home 指令（原点回归指令）进行编程。

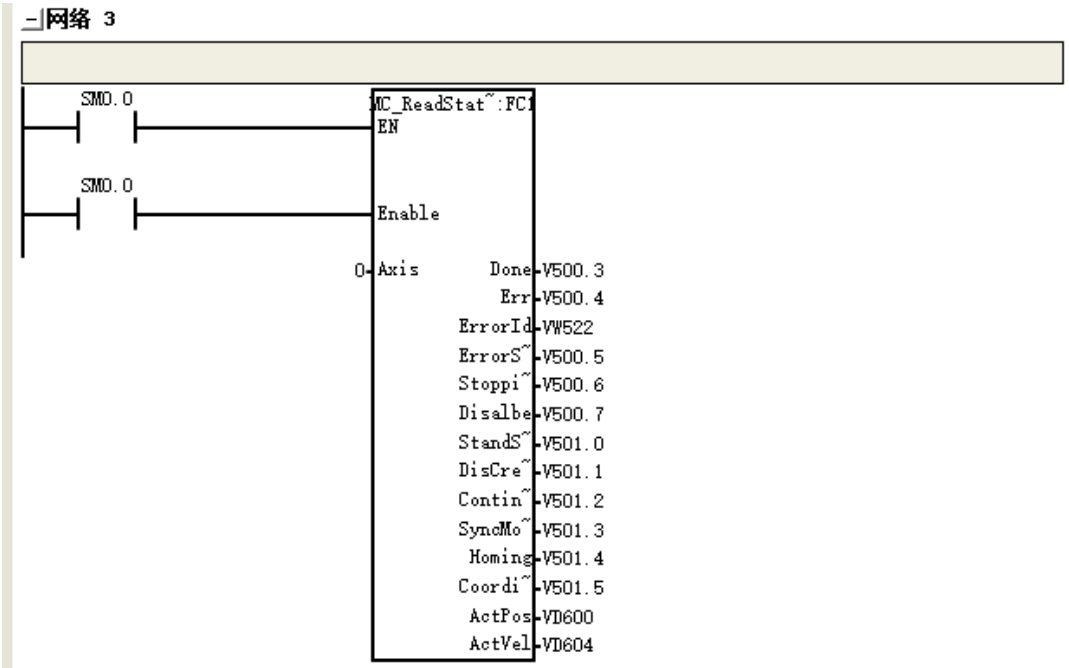
网络 1：回原模式为 19 号，回原搜索速度为 131072，回原爬行速度为 1000。



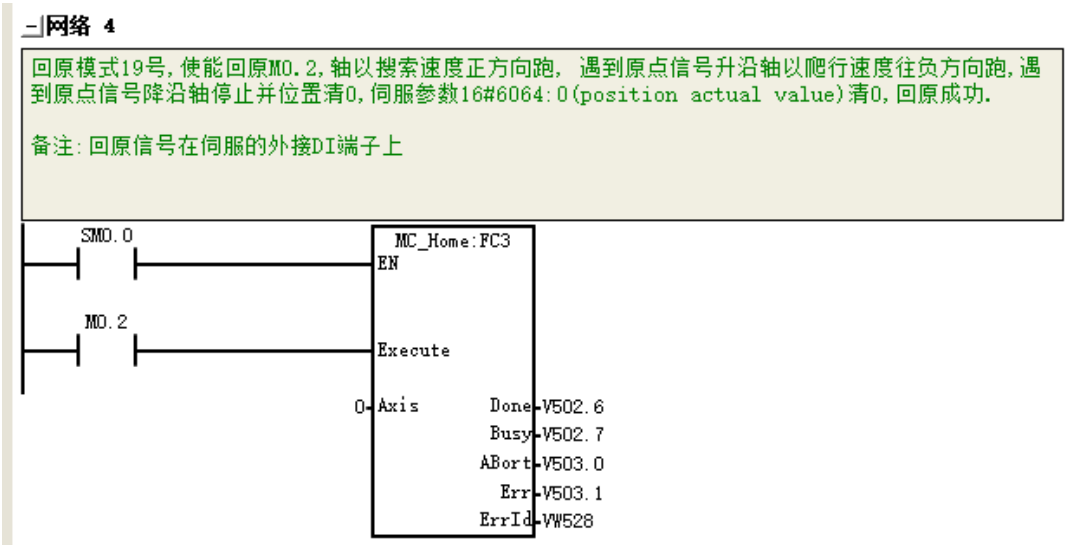
网络 2：使轴进行使能操作。



网络 3：读取轴状态



回原模式为 19 号，回原信号 I0.2，轴以搜索速度正方向跑，遇到原点信号升沿轴以爬行速度往负方向跑，遇到原点信号降沿轴停止并位置清 0，伺服参数 16#6064:0(position actual value)清 0。
备注：回原信号在伺服的外接 DI 端子上。

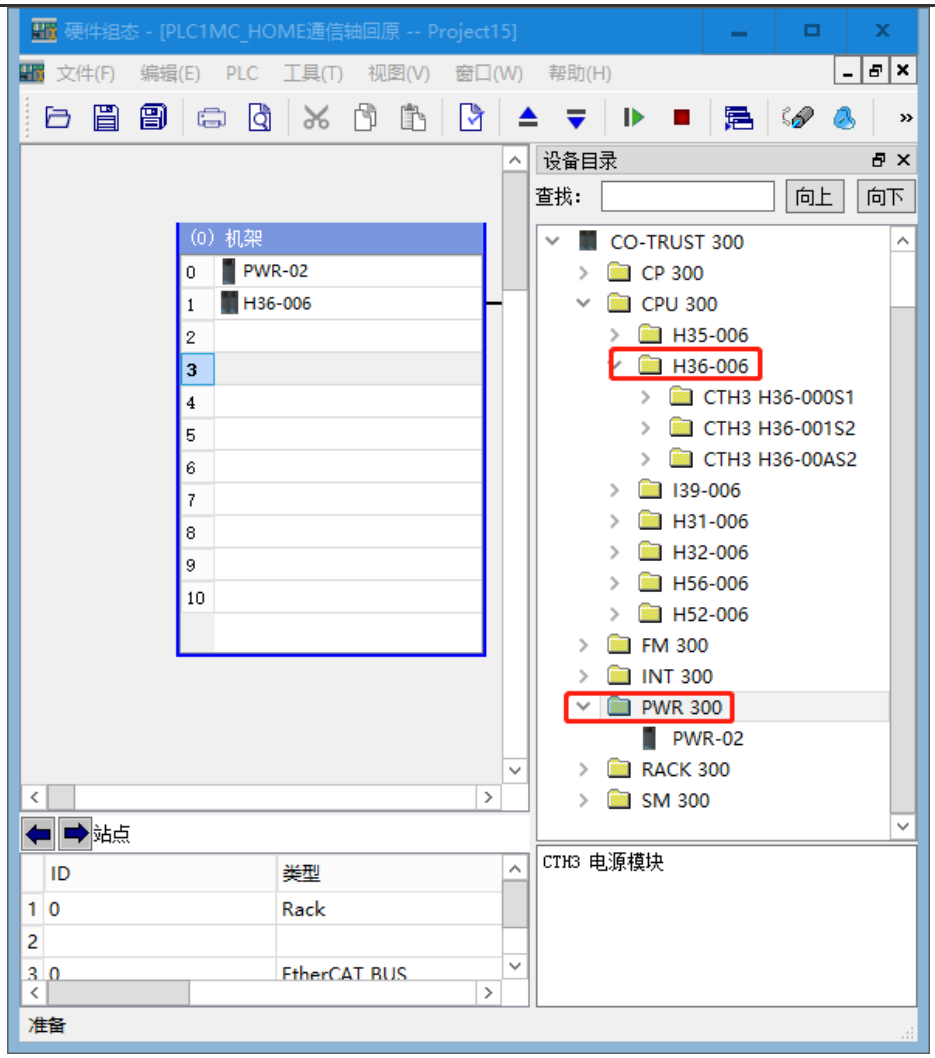


示例 3：通信轴特殊原点回归示例

在 H36-006 CPU 站点的项目管理器界面选择“硬件组态”，进行硬件组态。

步骤一：添加电源模块、CPU

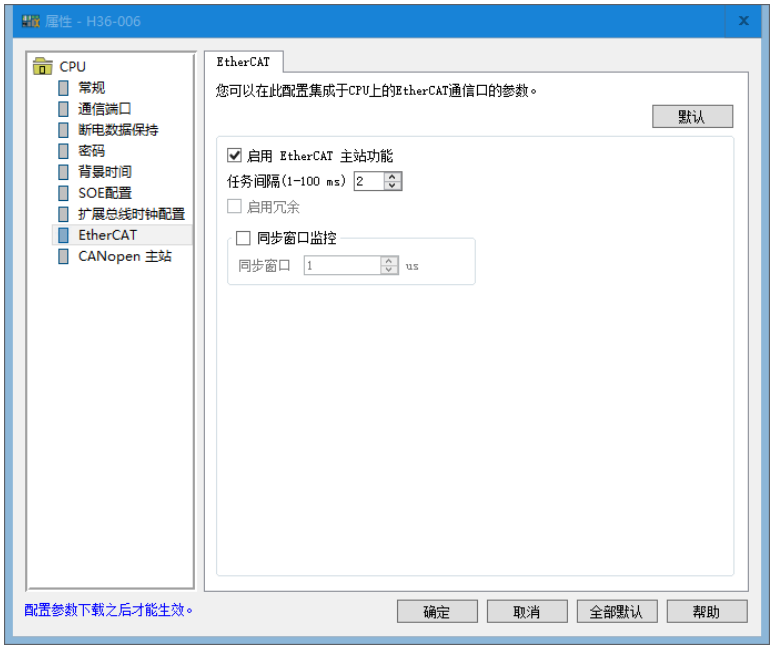
硬件组态界面中，在“CO-TRUST 300”项目树下顺序展开“PWR 300”和“CPU 300”文件夹，添加电源模块、CPU，电源模块只能放至机架的 0 号槽内，CPU 只能放至机架的 1 号槽内，如下图所示。



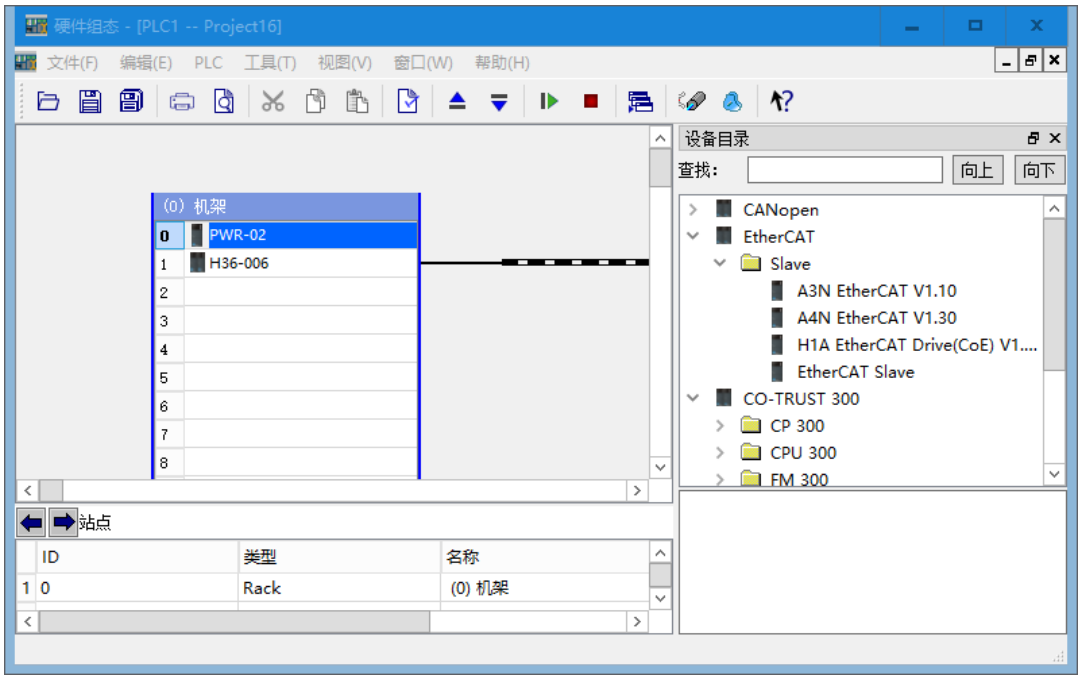
步骤二：配置轴

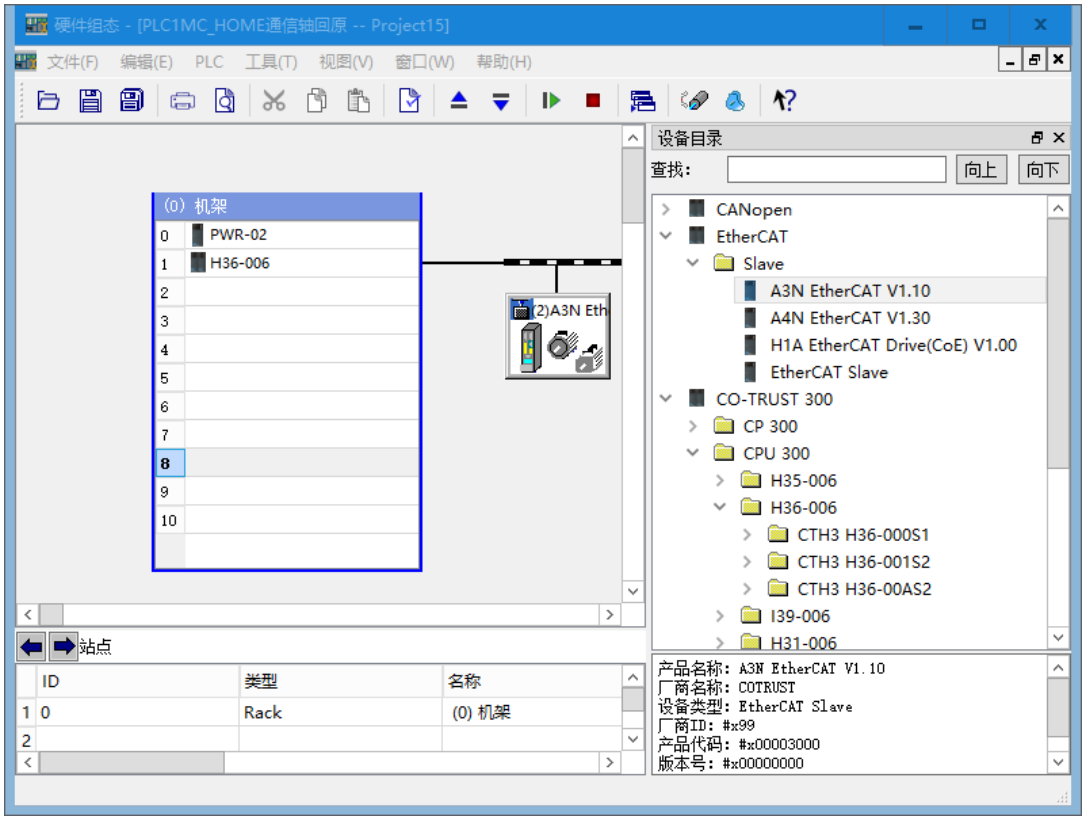
添加通信轴前需在硬件组态界面添加相应的 EtherCAT 从站，否则将无法添加通信轴。具体操作如下：

1）右击机架上的 H36，选择“属性”→“EtherCAT”，点击页面中的“启用 EtherCAT 主站功能”。

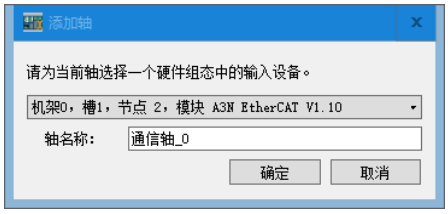


2）确定后可看到硬件组态界面的“EtherCAT BUS”，点击右边“EtherCAT”，选择从站并将从站拖到“EtherCAT BUS”下。



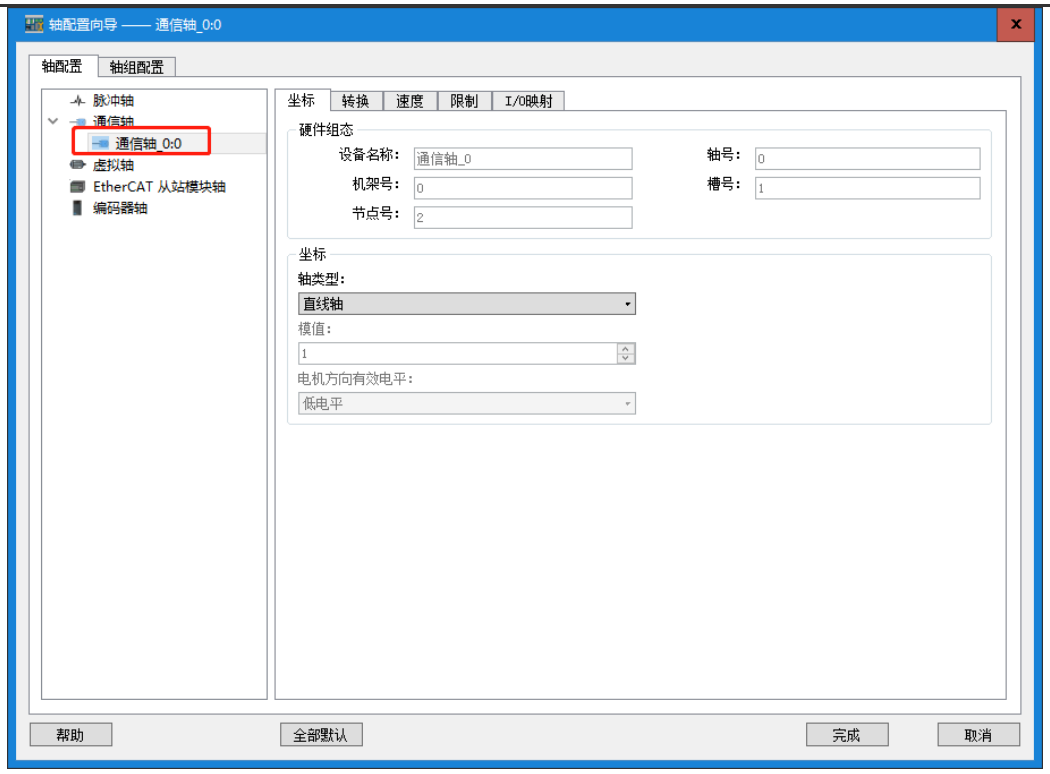


3）完成以上配置后在“工具”选项中选择“轴向导配置”，在轴向导配置界面双击“通信轴”即可添加通信轴，“轴名称”可修改，点击“确定”完成添加。



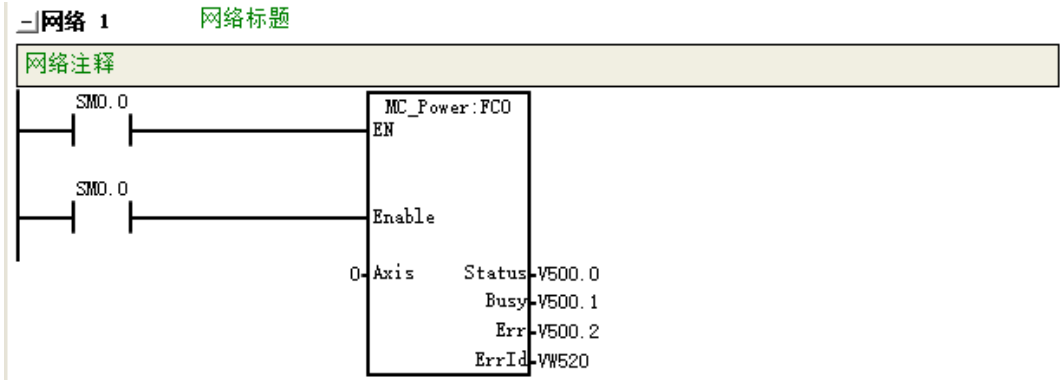
注意：一个从站只可对应添加一个通信轴。

4）添加的通信轴界面如下所示：

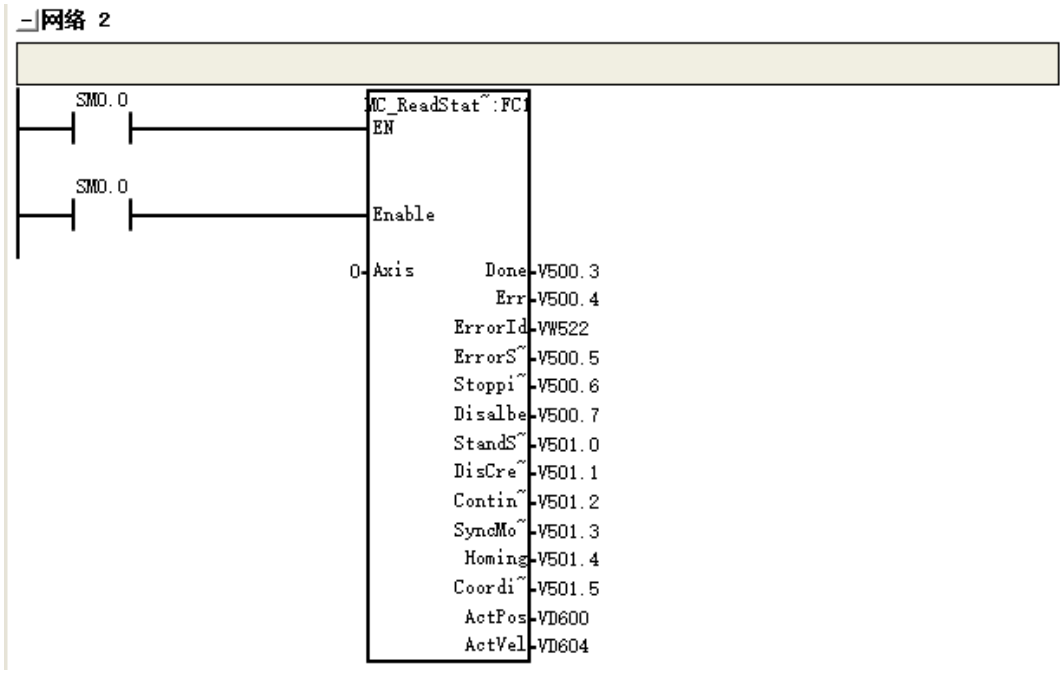


步骤三：在主程序中调用 SMC_Home 指令（特殊原点回归指令）进行编程。

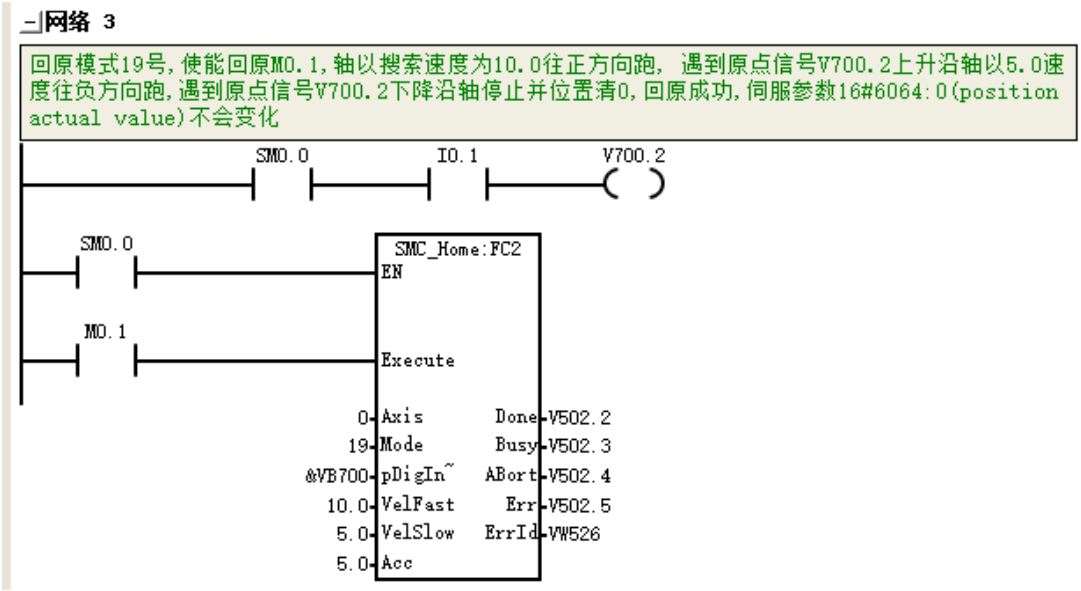
网络 1：使轴进行使能操作，轴 ID 号 Axis 设置为 0。



网络 2：读取轴状态。



网络 3：调用 SMC_Home 特殊原点回归指令，选用回原模式 19，轴的搜索速度为 10.0，加速度为 5。



4.3 SM253 运动控制库

PLC 通过调用 SM253 运动控制库可实现多轴独立控制功能。

4.3.1 sm253_motion_ctrl_lib 库指令一览表

库指令一览表

函数名	指令名称、功能	扩展模块 EM253、SM253	CPU **CTH200 系列：** H224X \ H226XL \ H226XM H226IM \ H226IL \ H226IH **CTMC 系列：** M228IL \ M228ML\ M228SL **CTSC 系列：** 224+（V5 平台） 224I（V5 平台）） 224XP（V5 平台） 226I（V5 平台） 226M（V5 平台） 226L（V5 平台） 226H（V5 平台）
MC253_EXT_RESET_EN	外部复位坐标使能指令	不支持	支持
MC253_INIT_DIR	配置电机方向指令	支持	支持
MC253_READ_POS	读位置指令	支持	支持
MC253_PTP_R	单轴相对运动指令	支持	支持
MC253_CIRCLE_R	两轴圆弧插补运动指令	不支持	支持
MC253_SPEED_CTL	速度控制指令	支持	支持
MC253_SET_POS_ZERO	软件回零指令	支持	支持
MC253_SET_POS_PV	设置目标位置指令	支持	支持
MC253_LINE_R	两轴直线插补运动指令	不支持	支持
MC253_EXT_RESET_EN_EXT	外部复位坐标使能指令II	支持	支持
MC253_SET_MAX_ACCELE	设置最大加速度指令	支持	支持

MC253_SET_CI_MODE	设置连续插补功能指令	不支持	支持
MC253_PWM	脉冲宽度调制指令	支持	支持
MC253_INIT	运控模块初始化指令	支持且必须在程序初始调用	支持
MC253_DO_CTRL	控制模块输出指令	支持	支持
MC253_READ_DI	读取模块输入状态指令	支持	支持
MC253_HSC_INIT	设置模块高速计数器指令	支持	支持
MC253_READ_HSC	读取模块高速计数器状态指令	支持	支持
MC253_PTP_A	单轴绝对运动指令	支持	支持
MC253_HOMING	回原指令	支持	支持

4.3.2 sm253_motion_ctrl_lib 库指令详解

对 Micro/Win 编程而言，运控模块的指令和 CPU 的指令格式是一样的，而区别在于所使用的指令库不同，CPU 指令前缀为 MC，而运控模块 SM253 的指令前缀为 MC253。CPU 指令的轴号为 0~3，而运控模块指令从 0 开始编号，可以往下延续，系统自动分配到每个运控模块上，如：运控模块指令轴号 0~1，分别对应第一个运控模块的 0 轴和 1 轴；运控模块指令轴号 2~3，分别对应第二个运控模块的 0 轴和 1 轴；类推，若超出运控模块范围，则认为参数错误，如：只有 1 个运控模块（只有 2 轴），而运控模块指令指定轴 2，则轴 2 找不到对应，从而无法使用。

同样地，模块高速计数器与 CPU 高速计数器不是统一编号的，使用模块的高速计数器必须调用模块指令，模块高速计数器从 0 开始编号，可以往下延续，系统自动分配到每个运控模块上，直至模块不足分配。如：模块指令高速计数器标号 0~1，分别对应第一个模块的 HSC0 和 HSC1；模块指令高速计数器 2~3，分别对应第二个模块的 HSC0 和 HSC1。

※特别注意

1、初始化和库存储区

使用 SM253 运动控制模块，要求在系统程序初始调用指令 MC253_INIT 以初始化系统控制变量（需且仅需一次，即只能用 SM0.1 调用）。注意，SM253 运控模块使用了 V 内存空间，用户可通过“库存储区分配”自行指定库存储区地址，在用户程序中用户不得使用库所占用的地址空间。（若 CPU 后只挂了 1 个 SM253 模块，则库占用了 218 个字节；若挂了 2 个，则库占用了 236 个字节；若挂了 3 个，则库占用了 254 个字节；若挂了 4 个，则库占用了 272 个字节；若挂了 5 个，则库占用了 290 个字节。）

2、模拟量滤波

当使用 SM253 运动控制模块时，必须在上位机软件系统块->输入滤波器->模拟量中将 SM253 运动控制模块对应的通道清除滤波功能，并将系统块下载到 PLC；否则将造成 SM253 运控模块的库指令不可使用。（查看 PLC 信息，其中 4 入 4 出的模拟量模块对应的通道即为 SM253 运控模块对应的通道。）

3、通信状态位

SM253 运控模块除了 MC253_INIT 之外的其它所有指令新增了通信状态位，若此状态位报警则表明通信有故障，通信信息未必可靠。此时要查看总线连接是否正常或者运控通道的模拟量滤波是否清除等。

4、脉冲输出指令的执行顺序

同轴脉冲输出指令（MC253_PTP_R、MC253_SPEED_CTL、MC253_PWM）最多只有 3 份数据缓冲区，即同一时刻只处理 3 条该类指令。当同一轴（比如均为 0 轴）上有多条该类指令时，若缓冲区未满（已使能的输出指令不足 3 条），则新使能的指令可得到及时处理，那么指令的执行顺序与指令使能的时间

顺序一致，而与指令位置顺序没有必然关系；若缓冲区已满（已使能的输出指令不小于 3 条），则新使能的指令不能得到及时处理，而只能待之前使能的指令结束并释放缓冲区后由程序扫描按位置先后顺序获得处理，即是执行顺序与其位置顺序有关，而不一定是按指令使能的时间顺序。可简单理解为“未满 3 按时间，满 3 按位置”（如例 1 所示）。

故，要实现一定的运动轨迹，若这些运动指令的位置顺序与期望实现的轨迹一致，则可一次性将这些指令全部使能。若这些运动指令的位置顺序与期望实现的轨迹不一致，则必须在使能的指令未满 3 条的情形下去使能期望轨迹所对应的指令（如例 2 所示）。

例 1）一次性使能所有指令

缓冲区未填满 3 份时使能的指令（指令 A/B/C）按使能时间顺序执行。如下表：将五条指令一次性全部使能，使能的顺序依次为：指令 A-->指令 C-->指令 B-->指令 E->指令 D，如表列 1 所示；则先使能的三条指令按照使能先后依次执行，即执行顺序为：指令 A-->指令 C-->指令 B，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 1—RUN 指令 A	1
3	0—AXIS 1—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
5	0—AXIS 1—RUN 指令 D	--
4	0—AXIS 1—RUN 指令 E	--

缓冲区已填满 3 份时剩余已使能的指令（指令 D/E）按其位置顺序依次执行。当指令 A 执行结束后释放缓冲区，由于程序自上而下扫描，指令 D 先获得该缓冲区；待指令 C 执行结束并释放缓冲区后，指令 E 获得该缓冲区。

指令使能的顺序依次为：指令 A-->指令 C-->指令 B-->指令 E->指令 D，如表列 1 所示；指令执行的顺序为：指令 A-->指令 C-->指令 B-->指令 D->指令 E，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 0—RUN 指令 A	1
3	0—AXIS 0—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
5	0—AXIS 1—RUN 指令 D	4
4	0—AXIS 1—RUN 指令 E	5

例 2）先使能三条指令，等有指令执行完成释放缓冲区后，再接着使能剩余的指令。

缓冲区未填满 3 份时使能的指令（指令 A/B/C）按使能时间顺序执行，如下表：先使能 3 条指令，使能顺序依次为：指令 A-->指令 C-->指令 B，如表列 1 所示，则此三条指令按照使能先后依次执行，即执行顺序为：指令 A-->指令 C-->指令 B，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 1—RUN 指令 A	1
3	0—AXIS 1—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
--	0—AXIS 0—RUN 指令 D	--
--	0—AXIS 0—RUN 指令 E	--

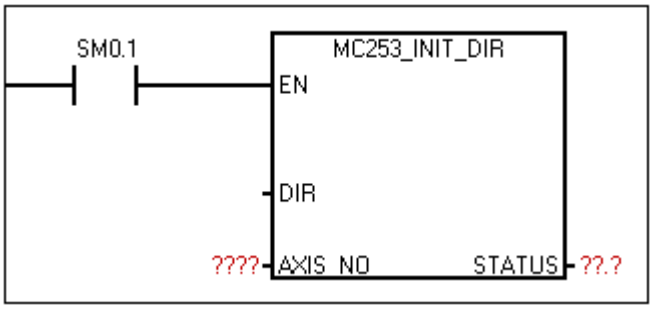
待指令 A 执行完成并释放缓冲区后，先使能指令 E，则指令 E 获得该缓冲区；待指令 C 执行完成并释放缓冲区后，再使能指令 D，则指令 D 获得该缓冲区。

指令使能的顺序依次为：指令 A-->指令 C-->指令 B-->指令 E->指令 D，如表列 1 所示；指令执行的顺序为：指令 A-->指令 C-->指令 B-->指令 E->指令 D，如表列 3 所示：

指令使能顺序	指令位置顺序	指令执行顺序
1	0—AXIS 0—RUN 指令 A	1
3	0—AXIS 0—RUN 指令 B	3
2	0—AXIS 1—RUN 指令 C	2
5	0—AXIS 1—RUN 指令 D	5
4	0—AXIS 1—RUN 指令 E	4

MC253_INIT_DIR（配置电机方向指令）

① 函数名：MC253_INIT_DIR



② 功能：配置电机的方向

提示

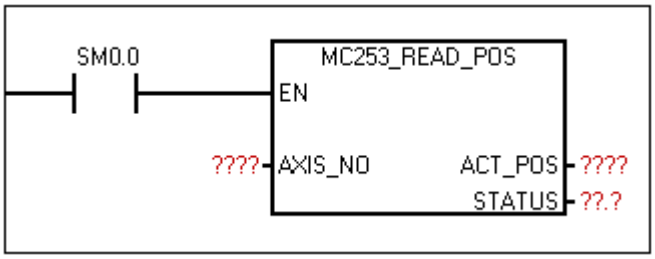
执行此指令只在 CPU 上电第一个扫描周期执行一次。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
DIR	IN	配置方向信号为正向时的有效电平。 DIR为1时，设置对应方向轴输出“1”时为电机正转。 DIR为0时，设置对应方向轴输出“0”时为电机反转。	Bool	0~1	默认值：1，即默认方向轴输出为“1”，为电机正转。
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

MC253_READ_POS（读位置指令）

① 函数名：MC253_READ_POS



② 功能：读取每轴的绝对坐标值。一旦设定原点坐标后，那么该值会根据输出的脉冲和方向的关系进行代数计算：正转输出一个脉冲：＋1，反转输出一个脉冲：－1。最后得到的是一个以设定点为原点的绝对坐标。

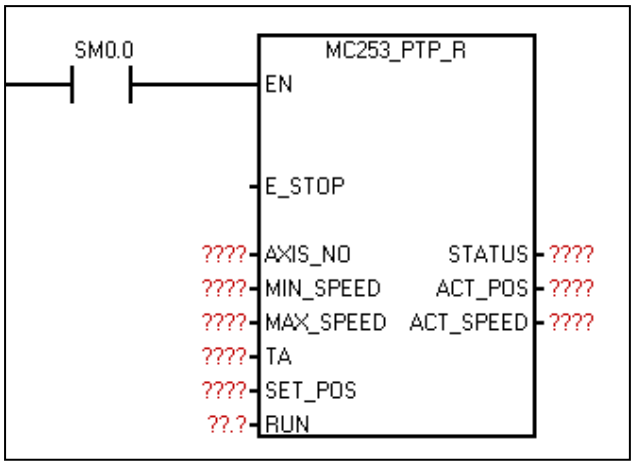
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	
ACT_POS	OUT	当前轴的绝对坐标（1个脉冲代表1个单位坐标）	Dint	-2147483648 ~ +2147483647	此指令无错误状态输出，轴号必须设置正确。
STATUS	OUT	通信状态标志位	Bool	0~1	

		1：通信超时			

MC253_PTP_R（单轴相对运动指令）

① 函数名：MC253_PTP_R



② 功能：用作单轴点对点控制（单轴定长驱动）。调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

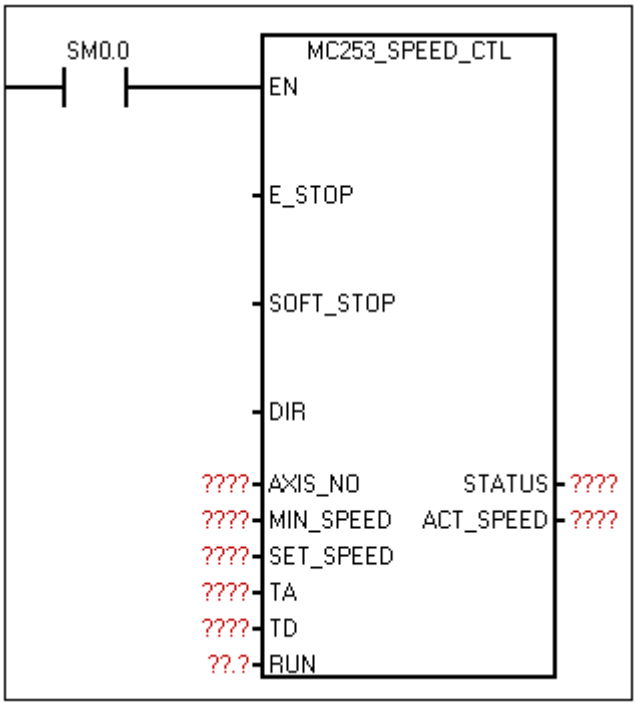
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	Bool	0/1	1、只有 Run ==1 与 E_Stop ==0 时才能运行。 2、当E_STOP 为1时，RUN 内部复位。
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数决定）	Byte	0~255	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：HZ	Dword	500~200000	• 最小速度的设定要小于最大速度。 • 此参数在运行过程中可以修改。
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：HZ	Dword	500~200000	
TA	IN	加速/减速时间。 单位：ms	Dword	0~10000	该参数在运行过程中可以修改（加速时间设置见错误!未找到引用源。**附注2**）
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿X轴的正方向，负脉冲数表示沿着 X 轴的负方向。	Dint	-2147483648 ~ +2147483647	该参数在运行过程中可以修改。 当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。 当新设定值小于已输出脉冲数，那么会马上停止脉

					冲输出。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	Bool	0/1	1、只有 RUN ==1与 E_STOP ==0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1时，RUN 内部复位。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4：模块急停标志（**见附注 3**） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。 Bit5~Bit6：预留 Bit7：指令通信状态标志 1—通信超时 0—无超时	Byte	0~255	Bit0： 1、只对轴参数和TA配置错误进行判断； 2、MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。
ACT_POS	OUT	当前的相对坐标或本指令已输出的脉冲数。	Dint	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度。	Dword	500~200000	

MC253_SPEED_CTL（速度控制指令）

① 函数名：MC253_SPEED_CTL



② 功能：控制单轴输出脉冲的频率，可任意时候改变输出脉冲的频率（速度）。当接收到软停止命令时，会自动减速停止。当收到紧急停止命令时，会马上停止脉冲输出，不经过减速。

③ 参数

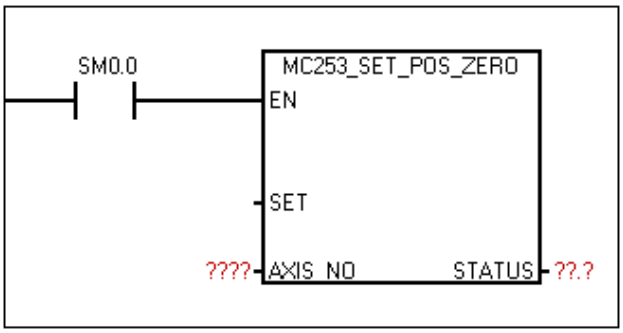
参数名	输入输出属性	参数描述	类型	数值范围	备注
RUN	IN	运行使能位。 1：有效， 0：无效。	Bool	0/1	1、只有 RUN ==1 与E_Stop == 0 与SOFT_STOP==0时才能运行。 2、当运行完成后，RUN内部复位。 3、当E_STOP 为1时，RUN内部复位。
E_STOP	IN	紧急停止位。1：有效，0：无效。 当收到有效紧急停止命令后，输出脉冲会马上停止，不经过减速。	Bool	0/1	1、只有 RUN ==1 与E_Stop ==0 与Soft_Stop==0时才能运行。 2、当E_STOP 为 1时，RUN内部复位。
SOFT_STOP	IN	软停止位。1：有效，0：无效。 当收到有效软停止命令时，输出脉冲会减速停止。	Bool	0/1	只有RUN ==1 与E_Stop==0 与SOFT_STOP==0时才能运行。
DIR	IN	脉冲的方向位	Bool	0/1	该参数在运行过程中能修改。
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	该参数在运行过程中不能修改。

MIN_SPEED	IN	最小速度，即启动时或停止时的速度。 单位：HZ	Dword	0~200000	1、设定速度为0时，没有脉冲输出；设定最小速度非0且小于92时，模块默认为92；若设定速度小于最小速度，模块默认将最小速度的值修改为设定速度的值。 2、该参数在运行过程中可以修改。
SET_SPEED	IN	设定速度，在收到停止命令前，输出脉冲会加速或减速到此速度。	Dword	0~200000	
TA	IN	加速时间，从最小速度到设定速度的加速时间。 单位：毫秒	Dword	0~10000 (见附注1)	注： 该参数在运行过程中可以修改。（加速时间设置见**附注**）。
TD	IN	减速时间，从设定速度到最小速度的减速时间。 单位：毫秒	Dword	0~10000 (见附注1)	
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4：模块急停标志（见附注3） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。 Bit5~Bit6：预留 Bit7：指令通信状态标志 1—通信超时	Byte	0~255	Bit0： 1、只对轴参数和TA/TD配置错误进行判断； 2、MIN_SPEED/SET_SPEED等参数不作报错，会自动设置成一个最接近的合理值。 3、253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。

		0—无超时			
ACT _SPEED	OUT	当前速度（频率）输出	Dword	500~200000	该值可能跟实际值会有一点偏差，最大不超过5K，跟加速时间和设定的速度有关。

MC253_SET_POS_ZERO（软件回零指令）

① 函数名：MC253_SET_POS_ZERO



② 功能：复位绝对坐标。

提示

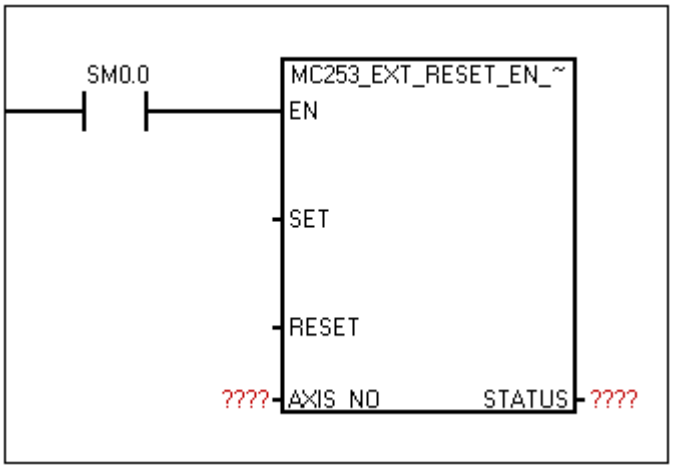
当机器运动到某一位置时，调用该指令，相当于把该轴的原点设定在该位置。那么以后每次调用“读绝对坐标”命令，就能得到相对于该点的坐标值。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围
SET	IN	清零功能使能位。 在SET上升沿把绝对坐标清0，每次调用时，SET应先置0，然后再置1。	Bool	0~1
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1

MC253_EXT_RESET_EN_EXT（外部复位坐标使能指令 II）

① 函数名：MC253_EXT_RESET_EN_EXT



② 功能：当调用该指令，设置是否使能外部 IO 复位绝对坐标值。

提示

轴号与外部复位信号的对应关系：

轴 0 ——I0.2（MC253_HSC0）

轴 1 ——I0.6（MC253_HSC1）

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围
SET	IN	SET上升沿，设置外部复位使能，每次调用时，SET应先复位，然后再置1。	Bool	0~1
RESET	IN	RESET上升沿，禁止外部复位使能，每次调用时，RESET 应先复位，然后再置1。	Bool	0~1
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255
STATUS	OUT	状态位： Bit0：复 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 位状态标志位 1—复位完成 0—复位未完成 Bit1~Bit6：预留 Bit7：通信状态标志位 1—通信超时 0—无超时	Byte	0~255

④ 使用说明

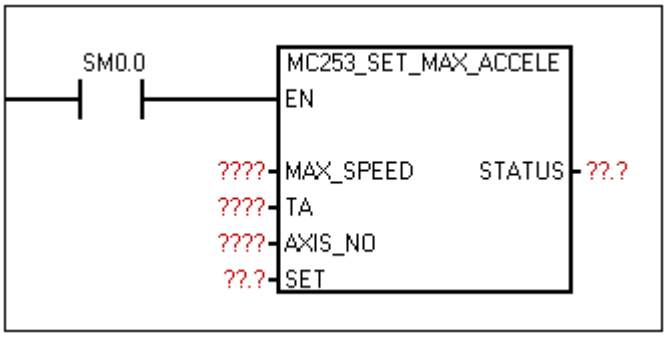
设 0 轴调用此指令。在 SET 上升沿使能外部复位功能之后，若 I0.2 检测到“有效复位信号”，则系统复位 0 轴绝对坐标，同时 STATUS 置位指示复位完成。在 RESET 上升沿禁止外部复位功能之后，即使 I0.2 检测到“有效复位信号”，系统亦不复位 0 轴绝对坐标，同时 STATUS 清零指示非复位状态。

提示

所谓“有效复位信号”，每轴的复位信号都有一个外部 IO 与其对应，同时由相应的寄存器设置其有效电平。如，0 轴对应 I0.2，由 MC253_HSC0 指令 MC253_HSC_INIT 参数 CONTROL 的第 0 位可以设置 0 轴的有效复位电平，当设置为 0 时，则 0 轴的有效复位信号为 I0.2 的高电平状态；当设置为 1 时， 则 0 轴的有效复位信号为 I0.2 的低电平状态；此设置当且仅当相应高速计数器（0 轴对应为 MC253_HSC0）得到使能时才有效，否则（即没有使能高速计数器）系统默认高电平为有效复位信号，如 0 轴，则 I0.2 高电平为有效复位信号。其它轴同理，各轴相关控制对应关系见此节②。

MC253_SET_MAX_ACCELE（设置最大加速度指令）

① 函数名：MC253_SET_MAX_ACCELE



② 功能：设置最大加速度（= MAX_SPEED/TA）（TA≠0）（若没有调用此指令，则认为没有设置最大加速度）

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
MAX_SPEED	IN	长轴最大速度，即运行中的最大速度。 单位：HZ	Dword	0~200000	运行过程中可以修改。
TA	IN	加速/减速时间。 单位：ms	Dword	0~10000 （见附注1）	运行过程中可以修改；若TA=0，则认为没有设置最大加速度。
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	
SET	IN	在以上参数确定后，给SET一个上升沿以使设置生效。	Bool	0~1	
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

④ 使用说明

若设置 X 轴参数 TA=0，或者 X 轴没有调用此指令，则认为 X 轴没有设置最大加速度；否则认为 X 轴设置有最大加速度，其值 MAX_ACCELE=MAX_SPEED/TA。此指令的意义在于：

- 可以设置一个合适的加速度限制某轴上各指令的加速度

如 PTP 指令，设置 AXIS_NO=0，MIN_SPEED=1000，MAX_SPEED=11000，TA=500，则理论上此 PTP 运动的加速度为 20HZ/ms（=（MAX_SPEED-MIN_SPEED）/TA）；若 0 轴调用了 MC253_SET_MAX_ACCELE 指令设置最大加速度为 15HZ/ms，则 PTP 实际可行的加速度为 15HZ/ms。（MC253_LINE_R 和 MC253_CIRCLE_R 亦然）

- 某轴上某指令要获取最大的加速度运行

如 PTP 指令要以最大加速度运行，可以先在同轴上调用 MC253_SET_MAX_ACCELE 指令设置最大加速度（即 MC253_SET_MAX_ACCELE 指令的参数 TA 不能为 0，否则无法得到最大加速度），同时将 PTP 指令的参数 TA 设为 0。若没有设置最大加速度而 PTP 指令的 TA=0，则在 PTP 指令上警报参数故障。（MC253_LINE_R 和 MC253_CIRCLE_R 亦然）

提示

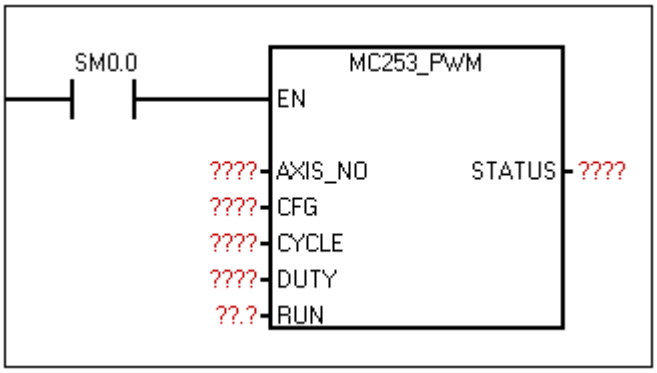
- 对于双轴指令 MC253_LINE_R 和 MC253_CIRCLE_R，若两轴均设置了最大加速度，则取两者中的较小值作为此双轴系统的最大加速度；若只有一轴设置了最大加速度，则取其为此

双轴系统的最大加速度；若双轴均没有设置最大加速度，则此双轴系统对加速度没有限制。

- 在确定该指令的参数 MAX_SPEED、TA、AXIS_NO 后，要给 SET 一个上升沿，这些参数才更新生效。

MC253_PWM（脉冲宽度调制指令）

① 函数名：MC253_PWM



② 功能：通过设置周期和占空比参数，可以输出不同周期和占空比的脉冲。

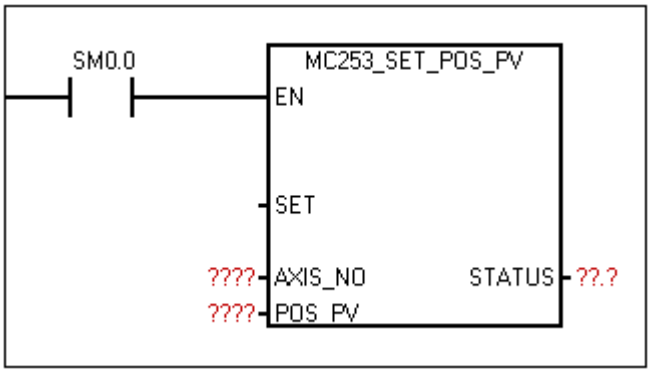
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
AXIS_NO	IN	设置轴号（每个SM253模块有2轴，轴号范围由运控模块数目决定）	Byte	0~255	运行过程中不能修改。
CFG	IN	基准时间单元 0：1us，1：0.5ms	Byte	0~1	运行过程中不能修改。
CYCLE	IN	脉冲的周期	Word	2~65535	运行过程中不能修改。
DUTY	IN	脉冲的占空比	Word	0~65535	运行过程中不能修改。
RUN	IN	运行使能	Bool	0~1	
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志	Byte	0~255	Bit0： ● 只对轴参数配置错误进行判断； ● CYCLE/DUTY等参数不作报错，会自动设置成一个最接近的合理值。 ● 253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。

		1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4：模块急停标志（见附注3） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。 Bit5~Bit6：预留 Bit7：指令通信状态标志 1—通信超时 0—无超时			

MC253_SET_POS_PV（设置目标位置指令）

① 函数名：MC253_SET_POS_PV



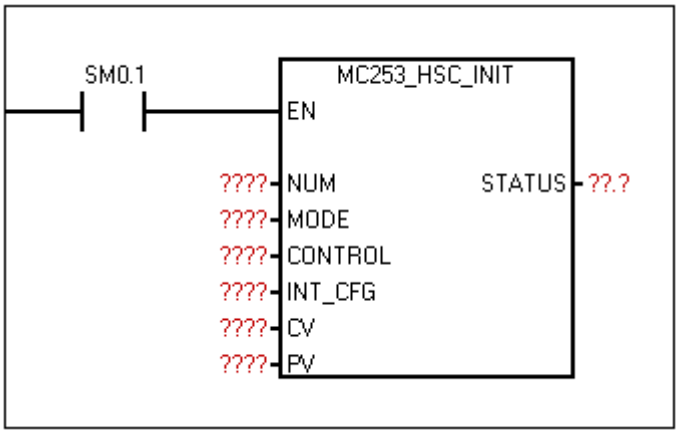
② 功能：此指令用于将机器所处的绝对位置写入到模块。比如，机器运行到某一位置时断电，可将其此时所处的位置保存下来，等下次上电时，将此位置写回到模块，则模块绝对位置计数起点与机器实际起点位置一致，而机器不需回到原点；若此位置刚好为原点，则此指令与 MC253_SET_POS_ZERO 相同效果。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围
AXIS_NO	IN	设置轴号（每个 SM253 模块有 2 轴，轴号范围由运控模块数目决定）	Byte	0~255
SET	IN	SET 上升沿，指令使能，每次调用时，SET 应先复位，然后再置 1。	Bool	0~1
POS_PV	IN	设定的目标位置，分正负。输出正脉冲表示沿 X 轴的正方向，负脉冲数表示沿着 X 轴的负方向。	Dint	-2147483648 ~ +2147483647
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1

MC253_HSC_INIT（设置模块高速计数器指令）

①函数名：MC253_HSC_INIT



②功能：配置高速计数器。

提示

执行此指令只在 CPU 上电第一个扫描周期执行一次。

③参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
NUM	IN	高速计数器标号(每个 SM253 模块有 2 个高速计数器，标号范围由运控模块数目决定)	Byte	0~255	
MODE	IN	计数模式	Byte	0~12	并不支持所有，详见附表。
CONTROL	IN	控制字： 7 6 5 4 3 2 1 0 Bit0：复位有效电平控制位 1—低电平 0—高电平 Bit1：预留 Bit2：正交计数器速率选择 1—1x 速率 0—4x 速率 Bit3：方向控制位 1—增计数 0—减计数 Bit4：更新方向 1—更新方向 0—不更新 Bit5：更新预设值 1—写新的预设值 0—不更新 Bit6：更新当前值 1—写新的当前值 0—不更新 Bit7：有效位 1—有效 0—无效	Byte	0~255	
INT_CFG	IN	中断配置（预留）	Byte	预留	预留，暂不支持中断

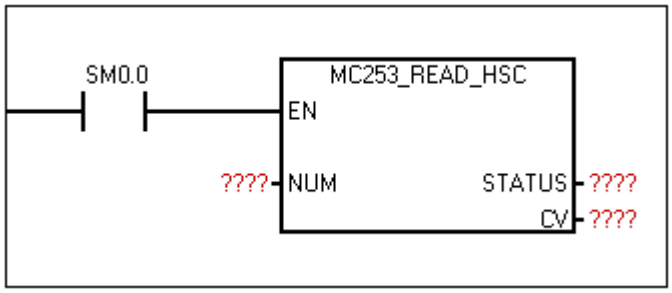
					功能。
CV	IN	新的当前值	Dword	-2147483648 ~ +2147483647	
PV	IN	新的预设值	Dword	-2147483648 ~ +2147483647	
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

附表

模式	描述	输入点		
	MC253_HSC0	I0.0	I0.1	I0.2
	MC253_HSC1	I0.4	I0.5	I0.6
0	带内部方向控制的单相计数器	时钟		
1		时钟		复位
3	带外部方向控制的单相计数器	时钟	方向	
4		时钟	方向	复位
9	A/B相正交计数器	时钟 A	时钟 B	
10		时钟 A	时钟 B	复位

MC253_READ_HSC（读取模块高速计数器状态指令）

①函数名：MC253_READ_HSC



②功能：读取模块高速计数器的计数状态和计数当前值。

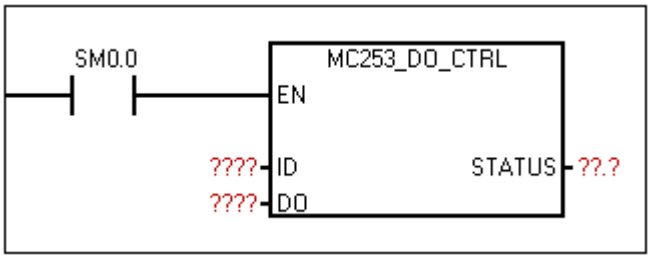
③参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
NUM	IN	高速计数器标号（每个 SM253 模块有 2 个高速计数器，标号范围由运控模块数目决定）	Byte	0~255	
STATUS	OUT	状态字： 7 6 5 4 3 2 1 0 Bit0：通信状态位 1—通信超时 Bit1~Bit4：预留 Bit5：当前计数方向位 1—增计数 Bit6：当前值等于预设值位 1—等于	Byte	0~255	

		Bit7：当前值大于预设值位 1—大于			
CV	OUT	新的当前值	Dword	-2147483648 ~ +2147483647	

MC253_DO_CTRL（控制模块输出指令）

①函数名：MC253_DO_CTRL



②功能：控制运控模块的输出值。

③参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
ID	IN	运控模块 ID 号	Byte	0~255	只适用于运控模块
DO	IN	模块输出端口值	Byte	0~255	DO 对应 8 路 Q 点输出，字节低位对应 Q 点低位，即 DO 的 bit0 对应 Q0.0，类推。
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

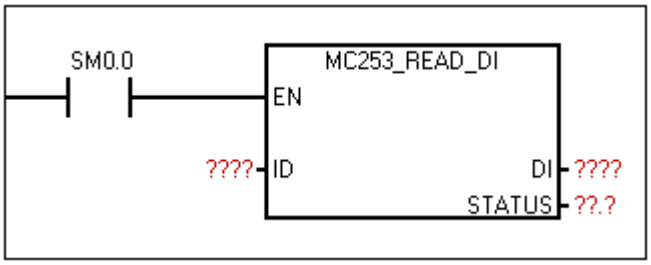
提示

1）运控模块的编号只是内部统一，即若 PLC 带 3 个模块，其中模块 0 为运控模块，模块 1 为非运控模块，模块 2 为运控模块，则模块 0 为 ID0，模块 1 不是运控模块不可控，模块 2 为 ID1。

2）若 CPU 后面挂单个运控模块，则由于传输延时，输出指令从使能到执行延时约为 780us；若 CPU 后面挂最多 5 个运控模块，则输出指令从使能到执行延时约为 930us。

MC253_READ_DI（读取模块输入状态指令）

①函数名：MC253_READ_DI



②功能：读取控制运控模块的输入值。

③参数

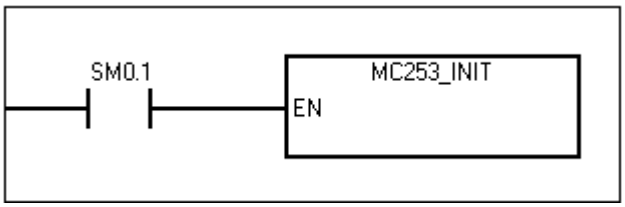
参数名	输入输出属性	参数描述	类型	数值范围	备注
ID	IN	运控模块 ID 号	Byte	0~255	只适用于运控模块
DI	OUT	模块输入端口值	Byte	0~255	DI：对应 8 路 I 点输入，字节低位对应 I 点低位，即 DI 的 bit0 对应 I0.0，类推。
STATUS	OUT	通信状态标志位 1：通信超时	Bool	0~1	

提示

- 运控模块的编号只是内部统一的，即若 PLC 带 3 个模块，其中模块 0 为运控模块，模块 1 为非运控模块，模块 2 为运控模块，则模块 0 为 ID0，模块 1 不是运控模块不可控，模块 2 为 ID1。
- 当某轴调用运动控制指令，若指令符合运行条件，则运行过程中该轴端口按指令输出脉冲或方向（对应端口不受指令 MC253_DO_CTRL 控制），运行结束后端口可恢复普通 IO 功能（即可接受指令 MC253_DO_CTRL 控制）；用户使用指令 MC253_DO_CTRL 当慎重，以免影响运动控制的使用需要。

MC253_INIT（运控模块初始化指令）

①函数名：MC253_INIT

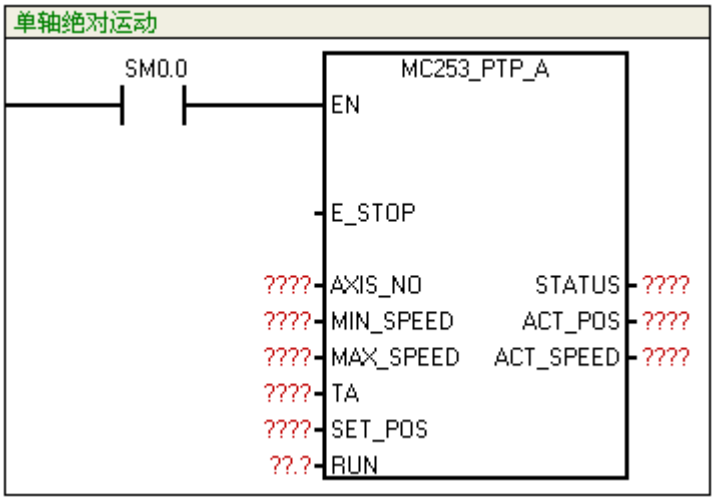


②功能：初始化运控模块的系统控制变量（参考本章节特别注意 1）。

③参数：无

MC253_PTP_A（单轴绝对运动指令）

① 函数名：MC253_PTP_A



② 功能：用作单轴点对点控制（非定长，而是定点）。调用一次可在原脉冲数基础上输出脉冲至指定脉冲数，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当

脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

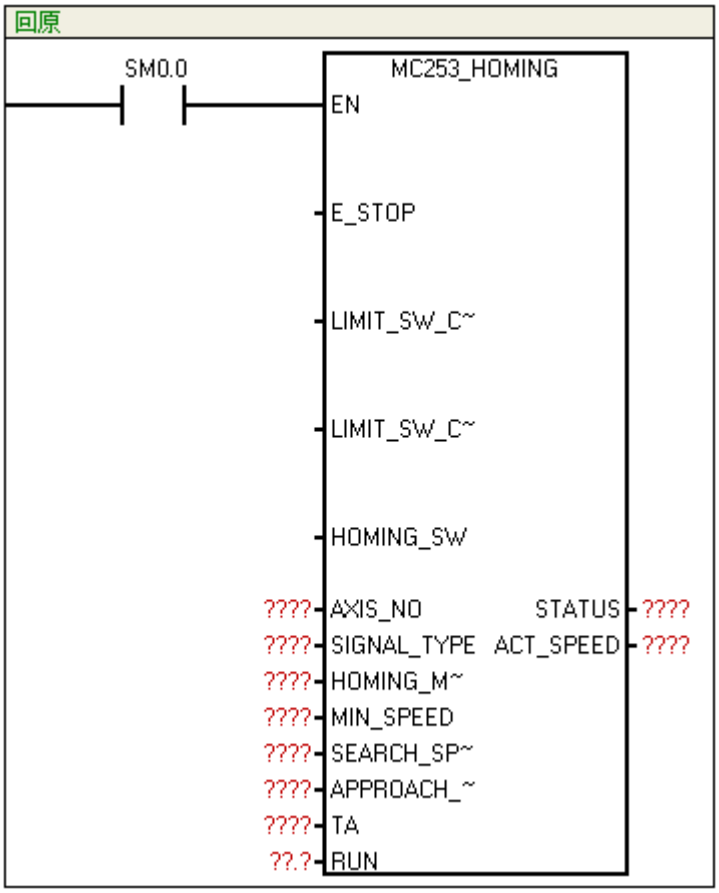
③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效 0：无效	BOOL	0/1	1、只有 Run =1 与 E_Stop=0 时才能运行。 2、当E_STOP为1时，RUN 内部复位。
AXIS_NO	IN	设置轴号	BYTE	0~255	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：HZ。	DWORD	500~200000	1、最小速度的设定要小于最大速度。 2、该参数在运行过程中可以修改。
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：HZ	DWORD	500~200000	
TA	IN	加速 /减速时间，单位 ms	DWORD	0~10000 （见附注1）	该参数在运行过程中可以修改（加速时间设置见附注2）
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿 X 轴的正方向，负脉冲数表示沿着 X 轴的负方向(此为绝对坐标)	DINT	-2147483648 ~ +2147483647	该参数在运行过程中可以修改，当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。当新设定值小于已输出脉冲数，则会马上停止脉冲输出。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	BOOL	0/1	3）只有RUN=1 与 E_STOP=0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当E_STOP 为1时，RUN 内部复位。

STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕 Bit2：完成标志 1—完成，指令执行完毕 0—未完成，指令未执行或指令正在执行中但未完成 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用 0—忙标志无效，指令正在执行或此执行已完成 Bit4：模块急停标志（**见附注3**） 1—模块急停标志有效，该轴被外部条件禁止运行。 0—模块急停标志无效。 Bit5~Bit6：预留 Bit7：指令通信状态标志 1—通信超时 0—无超时	BYTE	0~255	Bit0 ： 1、只对轴参数进行判断； 2、MIN_SPEED/ MAX_SPEED/TA 等参数不作报错，会自动设置成一个最接近的合理值。 3、253模块未被PLC识别或者是没有连接该模块时会提示参数配置错误。
ACT_POS	OUT	当前的绝对坐标	DINT	-2147483648 ~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度	DWORD	500~200 000	该值可能跟实际值会有一点偏差，最大不超过 5K，跟加速时间和设定的速度有关。

MC253_HOMING 回原指令

① 函数名：MC253_HOMING



② 功能：通过设置回原模式等参数，可寻找设备原点。

轴号与外部复位 IO 信号（如回原 Z 相信号）的对应关系：

轴 0 → I0.2（MC253_HSC0）

轴 1 → I0.6（MC253_HSC1）

若回原模式以原点开关为参考时（模式 3 或模式 4），必须将原点开关信号（指令的 HOMING_SW 参数）接至上述对应点，否则无法找到原点。

③ 参数

参数名	输入输出属性	参数描述	类型	数值范围	备注
E_STOP	IN	紧急停止位。 1：有效，0：无效	BOOL	0~1	1、只有 RUN =1 与 E_STOP =0 时才能运行。 2、当 E_STOP 为1 时，RUN内部复位。
LIMIT_SW_CCW	IN	CCW 限位开关	BOOL	0~1	
LIMIT_SW-CW	IN	CW 限位开关	BOOL	0~1	
HOMING_SW	IN	原点开关	BOOL	0~1	
AXIS_NO	IN	轴号	BYTE	0~255	该参数在运行过程中不能修改。
SIGNAL_TYPE	IN	7 6 5 4 3 2 1 0 信号类型 Bit0：逆时针限位开关信号类型 0—高电平 1—低电平 Bit1：顺时针限位开关信号类型	BYTE	0~255	

		0—高电平 1—低电平 Bit2：原点开关信号类型 0—高电平 1—低电平			
HOMING_MODE	IN	回原模式	BYTE	1~14	参见附注 4
MIN_SPEED	IN	最小速度。 单位：HZ	DWORD	0~200000	1、最小速度的设定要小于最大速度； 2、该参数在运行过程中可以修改。 3、搜索速度不应太大，接近速度应尽量小。
SEARCH_SPEED	IN	原点搜索速度。 单位：HZ	DWORD	0~200000	
APPROACH_SPEED	IN	原点接近速度。 单位：HZ	DWORD	0~200000	
TA	IN	加减速时间。单位：ms	DWORD	0~10000（见附注 1）	该参数在运行过程中可以修改（加速时间设置见附注 2）
RUN	IN	运行使能位 1：有效	BOOL	0~1	1）只有RUN =1与E_STOP =0时才能运行。 2）当运行完成后，RUN 内部复位。 3、当 E_STOP为1时，RUN 内部复位。

④ 说明

程序对各开关的检测以扫描方式实现，故当开关量变化时处理不及时，可能有些延迟。若回原速度（包括搜索速度和接近速度）太大时，这个处理延迟被放大，导致回原不准。

附注

1、当 TA≠0，加速度=（MAX_SPEED-MIN_SPEED）/TA（若设置有最大加速度，则受限于最大加速度）；

若 TA=0，则采用指令 MC253_SET_MAX_ACCELE 设置的最大加速度，若没有设置最大加速度，则报参数故障。TD 亦然。对于双轴指令，若两轴均设置了最大加速度，则采用其中的较小值作为系统加速度。

2、理论上，指令加速度=【(MAX_SPEED - MIN_SPEED) / TA】，如果由此所求的加速度过于小（小于 1），则指令内部默认加速度为 1。用户可按预期加速度，合理设置 TA/TD。

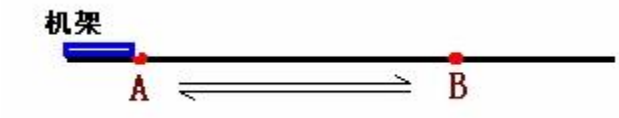
3、在模块输入端口引入了“模块急停信号”：当模块检测到此信号时，禁止脉冲输出，并在脉冲输出指令（如 MC253_PTP_R、MC253_SPEED_CTL 和 MC253_PWM 指令）状态位处报警。轴号与急停信号的对应关系：

轴 0 → I0.3
轴 1 → I0.7

4.3.3 调试示例

混合运动控制指令使用

控制步进电机从 A 点到 B 点往返运动，步进电机细分 1000，丝杆导程 5mm，A 到 B 的位移 L 为 2000mm。



【系统说明】

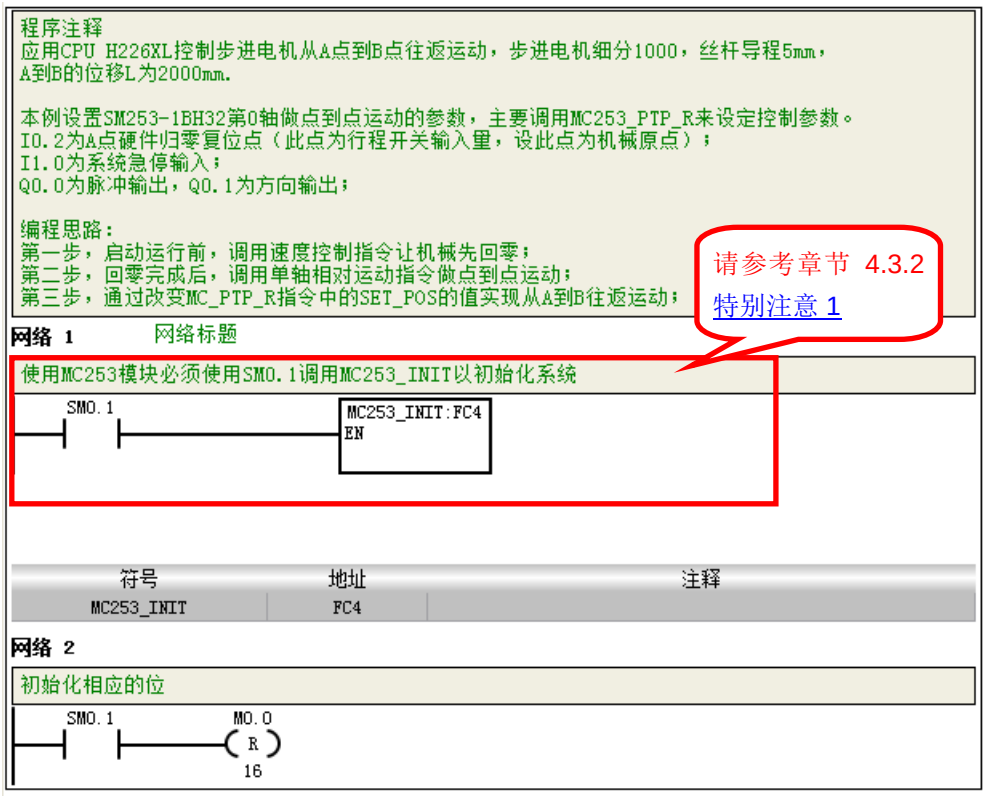
本例中，在 CPU H226XL 后挂 1 个 SM253 模块，设置 SM253 模块第 0 轴做点到点运动的参数。主要调用 MC253_PTP_R 来设定控制参数。

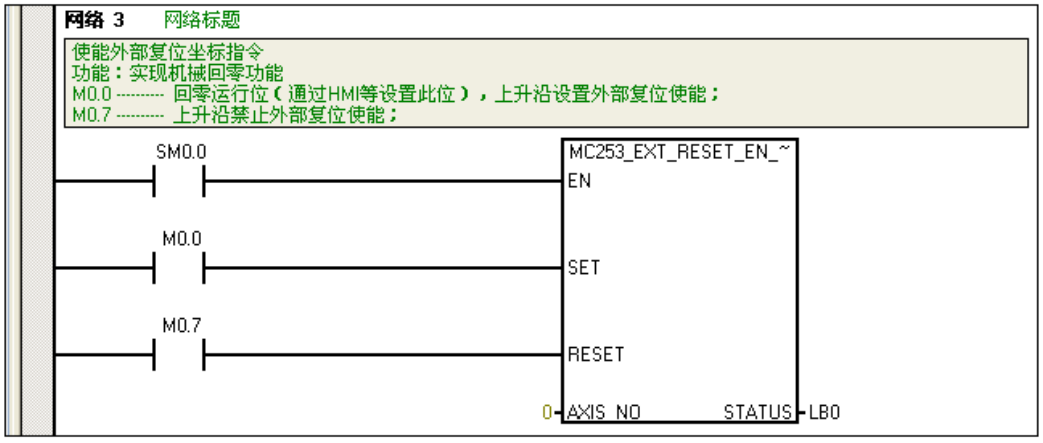
I0.2 为 A 点硬件归零复位点（此点为行程开关量输入，设此点为机械原点）；

I1.0 为系统急停输入；

Q0.0 为脉冲输出，Q0.1 方向输出。

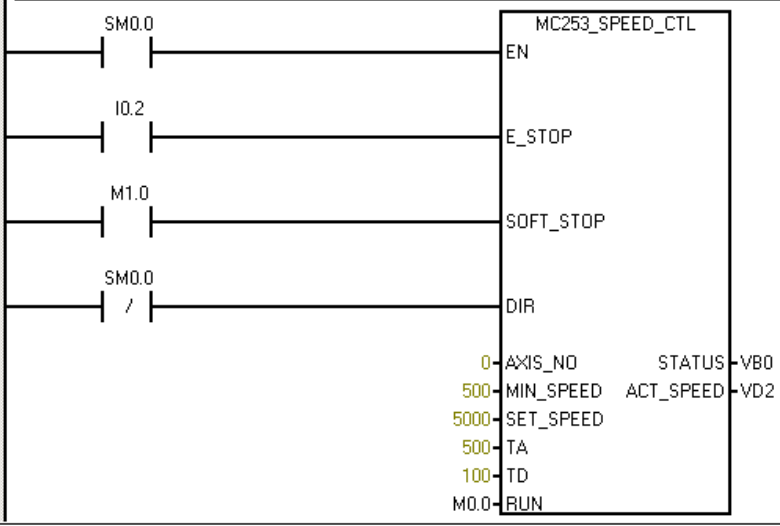
【程序块】





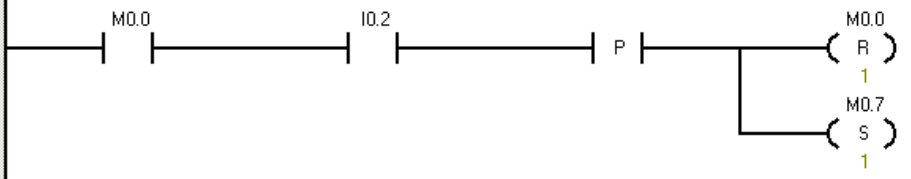
网络 4

速度控制指令
功能：先控制0轴反向旋转，回到机械原点（即A点为运动的起始点）
M0.0 ---------- 运行使能位;
I0.2 ---------- 行程限位开关，作指令的急停开关，即回到零点时关闭速度控制指令;
M1.0 ---------- 软停止位;
脉冲方向为反转，实现回零；
轴号为0；
启动/停止速度为500HZ；
设定速度为5000HZ（加速时间到后的正常速度）；
加速时间500ms；
减速时间100ms；
输出状态字放在VB0中；
当前速度（频率）放在VD2中



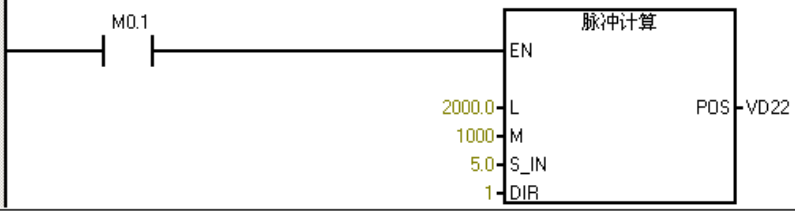
网络 5

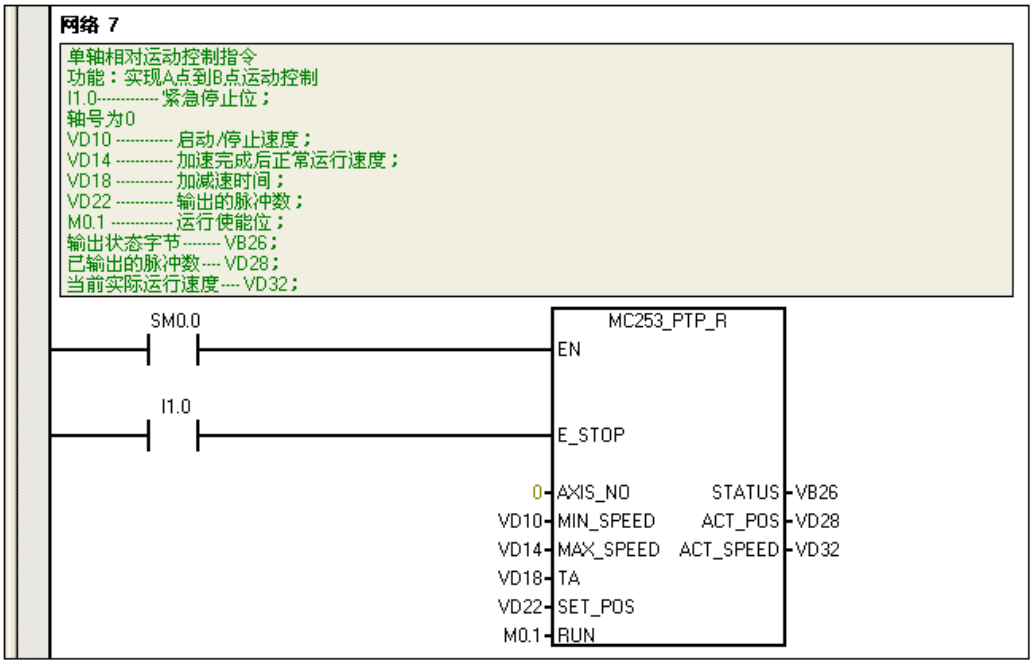
归零完成后复位相应的位；

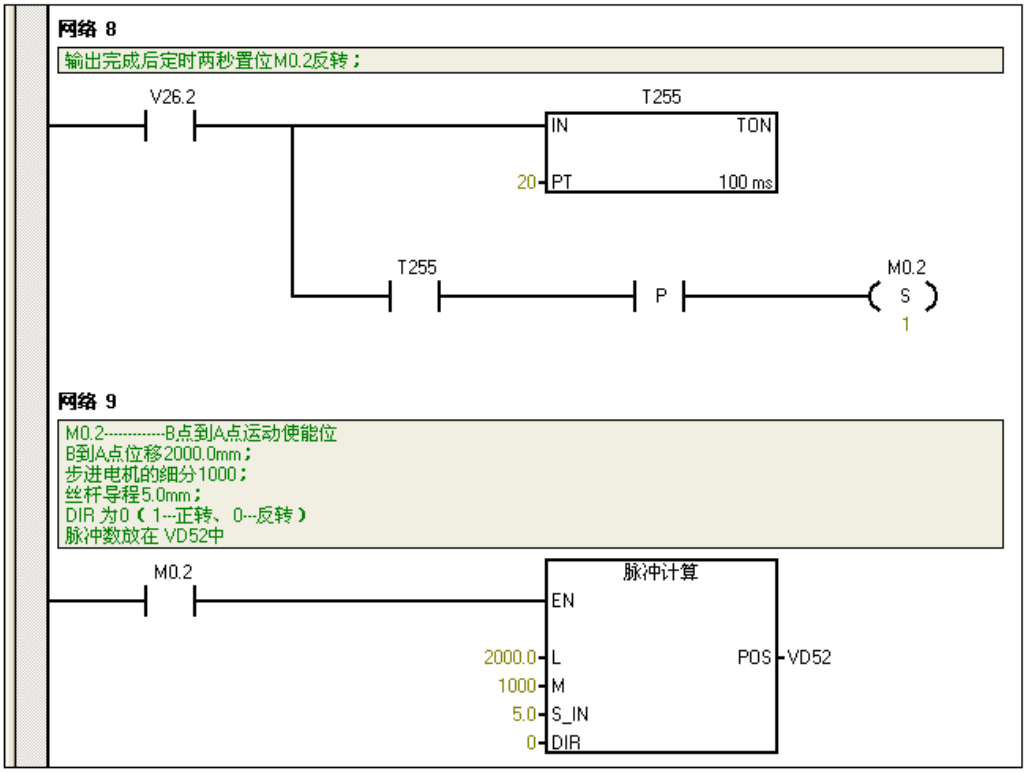


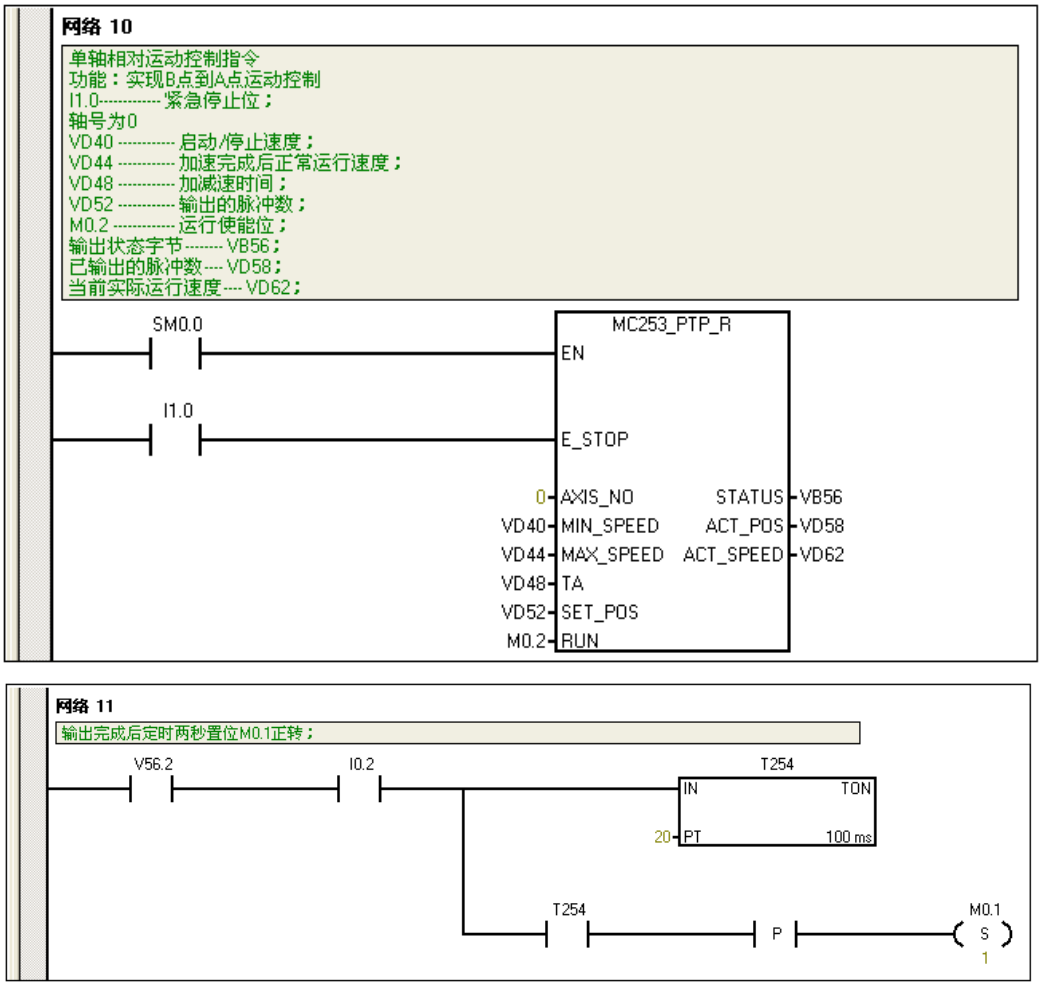
网络 6

M0.1---------- A点到B点运动使能位
A到B点位移2000.0mm；
步进电机的细分1000；
丝杆导程5.0mm；
DIR 为1（1---正转、0---反转）
脉冲数放在 VD22中




网络 7
单轴相对运动控制指令
功能：实现A点到B点运动控制
I1.0---------- 紧急停止位；
轴号为0
VD10 ---------- 启动/停止速度；
VD14 ---------- 加速完成后正常运行速度；
VD18 ---------- 加减速时间；
VD22 ---------- 输出的脉冲数；
M0.1 ---------- 运行使能位；
输出状态字节-------- VB26；
已输出的脉冲数---- VD28；
当前实际运行速度---- VD32；
SM0.0
I1.0
MC253_PTP_R
EN
E_STOP
0 AXIS_NO
VD10 MIN_SPEED
VD14 MAX_SPEED
VD18 TA
VD22 SET_POS
M0.1 RUN
STATUS VB26
ACT_POS VD28
ACT_SPEED VD32


网络 8
输出完成后定时两秒置位M0.2反转；
V26.2
T255
IN TON
20 PT 100 ms
T255
P
M0.2
S
1
网络 9
M0.2----------B点到A点运动使能位
B到A点位移2000.0mm；
步进电机的细分1000；
丝杆导程5.0mm；
DIR 为0（1---正转、0---反转）
脉冲数放在 VD52中
M0.2
脉冲计算
EN
2000.0 L
1000 M
5.0 S_IN
0 DIR
POS VD52



4.4 CTH200 系列热电偶型温度 PID 模块控制库

4.4.1 功能介绍

“PID_setting”功能库是专门为热电偶型的 PID 扩展模块（如：SM231-7TD、SM231-7TF）提供参数设定的。热电偶 PID 模块内部集成 PID 算法，用户无需复杂编程，只需调用“PID_setting”库给模块设置一些简单的参数就可以使用，温度控制准确。

提示

- 如需该库，请前往 http://www.co-trust.com 网站免费下载。
- 此库适用 CTH2 231-7TD32、CTH2 231-7TF32 模块；
- SM231-7TD32 和 SM231-7TF32 使用时会占用一部分 V 存储区，请编程人员编程时不要使用这些 V 存储区。
- 西门子 CPU222 因数据空间限制不能使用此库。

4.4.2 指令详解

① 指令名称：SOCK_Close

② 功能：关闭连接

③ 参数说明

参数地址	说明	类型	数值范围	备注
Run	运行	位	0 或 1	
Slot	槽号从 0 开始	字，常数或变量	0~6	
Channel	通道号	字，常数或变量	0~7	
SP	设定值	字，常数或变量	-2000~32767	单位：0.1℃
CTRLByte	控制字节，控制 PID 运行	常数或变量		常用控制字节： 1.16#03(只有加热输出，带自适应功能) 2.16#07(加热冷却输出，带自适应功能)
Cycle	脉冲输出周期	字，常数或变量	1~255	单位：秒
Kp	比例系数	字，整数，常数或变量		
Ti	积分时间	字，整数，常数或变量	1~3600	单位：秒
Td	微分时间	字，整数，常数或变量	0~3600	单位：秒
Heating	加热输出	位		
Cooling	冷却输出	位		
PV	测量值（反馈值）	字，变量	-2000~32767	单位：0.1℃
PID_out	PID 模拟输出	字，整数，变量	定义为仅加热输出时，输出范围：0~32000，带冷却输出时：-32000~32000	

※ 控制位地址

控制字的位地址意义如下：

控制字位	设置	备注
0	0	PID 停止
	1	PID 运行
1	0	积分一直起作用，比例系数 Kp 不自动调整
	1	积分分离及比例系数自动调整
2	0	PID 单极输出
	1	PID 双极输出
3	0	保留
	1	保留
4	0	积分起作用
	1	积分不起作用
5	0	微分起作用
	1	微分不起作用
6		保留
7		保留

※ PID 地址与参数配置

• PID 地址计算公式

地址名称	计算公式	备注
PID 参数地址	A=(2048+S*256)+16*C	S 为模块所在的槽号（范围：0～6） C 为通道号
PID 正向脉冲输出地址	X=(2048+S*256)+12	
PID 负向脉冲输出地址	Y=(2048+S*256)+13	

• PID 参数输出部分（模块到 CPU）

内容	地址	数值设置范围	实际对应数值
实际温度	VW A	0～13000	0～1300 度
状态字	VW A+2		
PID 模拟量输出	VW A+4	-32000～32000	

• PID 参数输入部分(CPU 到模块)

内容	地址	数值设置范围	实际对应数值
设定温度	VW A+128	0～13000	0～1300 度
控制字节		VB A+130 位等于 0 时	VB A+130 位等于 1 时
	V(A+130).0	PID 不运行，没输出	PID 运行
	V(A+130).1	积分一直起作用，比例系数 Kp 不自动调整	积分分离及比例系数自动调整
	V(A+130).2	PID 单极输出，0～32000	PID 双极输出，-32000-32000，具有加热和冷却功能
	V(A+130).3	未使用	
	V(A+130).4	积分起作用	积分不起作用
	V(A+130).5	微分起作用	微分不起作用
	V(A+130).6	实际温度值滤波，抗干扰能力更强	实际温度值不滤波
PID 脉冲输出周期设定	VW A+132	1～255	1～255 秒
Kp(比例系数)	VW A+134	0～9999	0～999.9
Ti(积分时间)	VW A+136	0～3600	0～3600 秒
Td(微分时间)	VW A+138	0～3600	0～3600 秒

• 加热脉冲输出地址

0 通道脉冲输出	V X.0
1 通道脉冲输出	V X.1
2 通道脉冲输出	V X.2
3 通道脉冲输出	V X.3
4 通道脉冲输出	V X.4
5 通道脉冲输出	V X.5
6 通道脉冲输出	V X.6
7 通道脉冲输出	V X.7

• 冷却脉冲输出地址

0 通道脉冲输出	V Y.0
1 通道脉冲输出	V Y.1

2 通道脉冲输出	V Y.2
3 通道脉冲输出	V Y.3
4 通道脉冲输出	V Y.4
5 通道脉冲输出	V Y.5
6 通道脉冲输出	V Y.6
7 通道脉冲输出	V Y.7

4.4.3 应用举例

系统说明

本例程设置 1 个 SM231-7TD 扩展模块（插槽号为 0）第 1 个 PID 回路（通道号为 0）的参数。调用 PIDSetting 来设定该回路的参数，无需计算 PID 参数地址，只需输入回路所在的插槽号和通道号，再使能 Run 来运行该回路。

Q0.0 为正向脉冲输出；Q0.1 为负向脉冲输出；

VW0 为实际温度；VW2 为 PID 模拟量输出；

修改 PID 设定参数使用其他地址；

设定温度：VW120；

控制字：VB122；

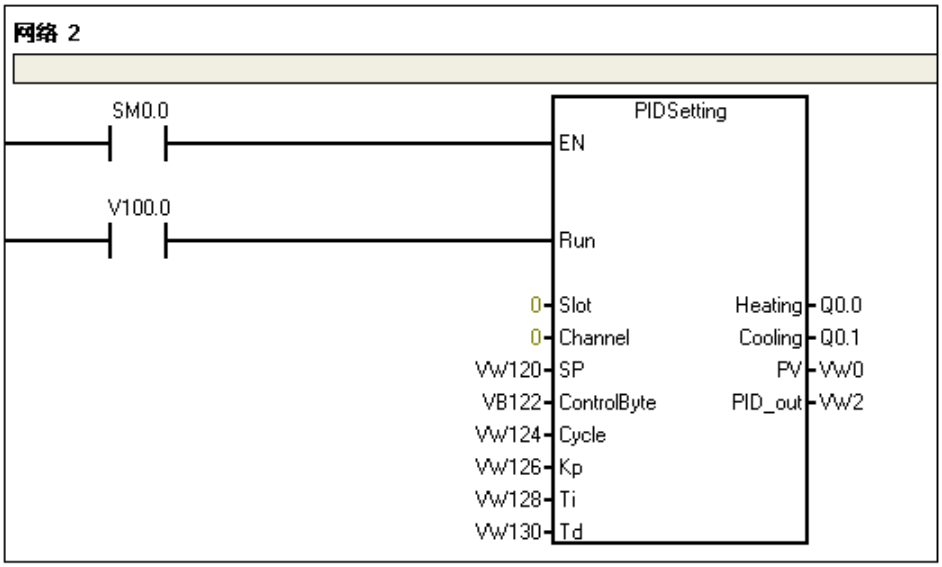
脉冲输出周期：VW124；

比例系数：VW126；

积分时间：VW128；

微分时间：VW130；

应用程序



提示

为了保证 PID 模块能正常使用，编写其它程序时请务必不要使用如下的 PID 模块占用的 V 存储区。

模块在第 0 号插槽所占用的地址为：VW2048 到 VW2298

模块在第 1 号插槽所占用的地址为：VW2304 到 VW2554

模块在第 2 号插槽所占用的地址为：VW2560 到 VW2810
模块在第 3 号插槽所占用的地址为：VW2816 到 VW3066
模块在第 4 号插槽所占用的地址为：VW3072 到 VW3322
模块在第 5 号插槽所占用的地址为：VW3328 到 VW3578
模块在第 6 号插槽所占用的地址为：VW3584 到 VW3834

称重指令库

5

5.1 EM231_7WA_LIB（V2.50）

称重模块属于 CTH200 自动化系统中的一个模拟量扩展模块，用于检测灵敏度为 1 到 4mV/V 的称重测量传感器，实现模拟信号到数字信号转换，为得到实际的重量，需在 PLC 中使用合信提供的称重库（可访问合信官网下载，网址：http://www.co-trust.com）。

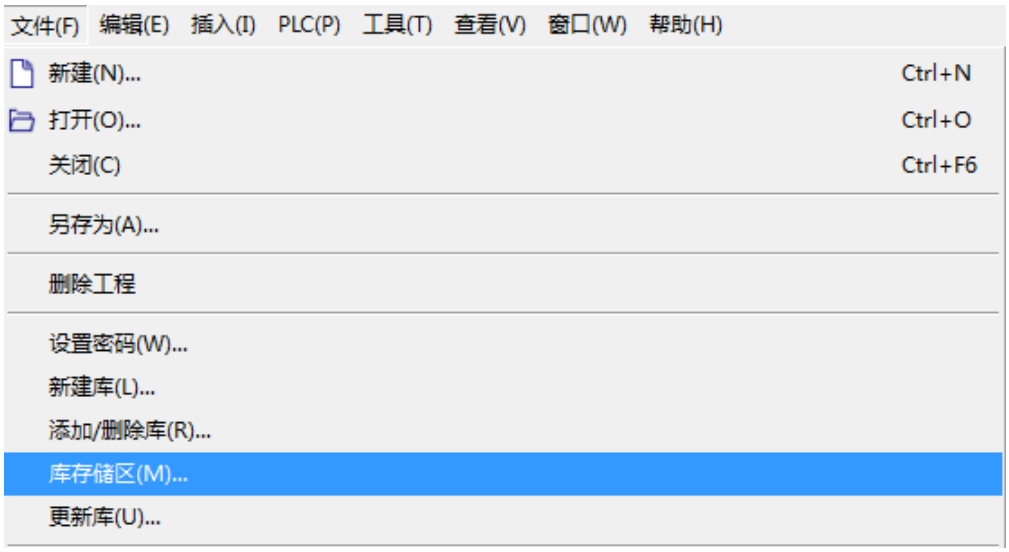
目前合信称重库含 ct_em231_7wa_d_lib(v1.1)和 ct_em231_7wa_lib(v2.4)，两者参数定义一样，只数据位宽不同。称重库包括配置库、初始化库、标准库和扩展库四部分，用户可根据具体设置测得实际重量的模拟信号，并将其转换为数字信号输出，从而获得实际重量。

本章分别介绍称重库（以 ct_em231_7wa_lib(V2.4)为例）的相关库参数说明。

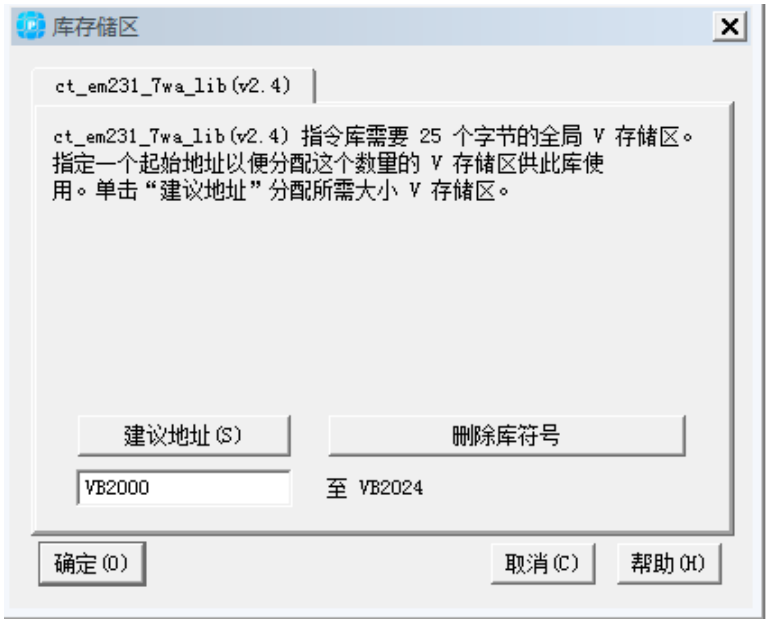
5.1.1 配置库存储区

在 MagicWorks PLC 中进行配置

在菜单栏中选择“文件” →“库存储区”，如下图所示：



执行以上操作后弹出如下窗口，指定一个不被占用的起始地址以便分配指定数量的 V 存储区供库运行使用，最后点击“确认”，即配置库存储区成功。



注意：分配的存储区不能被程序其它地方使用，否则会导致称重库不能正常运行。

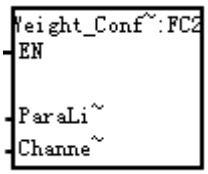
5.1.2 指令详解

Weight_Config（称重配置指令）

①指令名称：Weight_Config；

②功能：设置通道数目和各通道参数表的起始地址；

③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
	EN	IN	BYTE	--	--	使能端
	ParaListBase	IN	DWORD	--	--	设置参数表起始地址指针，从该地址开始存放参数表，参数具体定义请参见“**参数表说明**” 注：输入操作数前必须有一个“**&**”符号
	ChannelNum	IN	BYTE	0~255	--	设置连接的需要的通道总数目，如 6 通道模块，则配置为 6，若使用两个六路称重模块，则配置为 12

注意：必须调用一次，且只在第一个循环周期执行一次。如果有 N 个通道，则所有通道参数表总共需要占用 EM231_ChannelParaNum * N 个字节。用户必须自己确保通道参数表占用的 V 内存没有超出允许范围，并且没有覆盖其它程序占用的空间。

参数表说明

调用称重库时，每个通道占用 72 个字节内存，该存储区保存的参数定义如下表（以通道 0 起始地址 VB0 为例）：

参数名	地址	说明	备注
Mode	VB0	模式	基本/扩展模式
Sensitivity	VB1	传感器灵敏度	
LimitFreq	VB2	低通滤波频率	
FilterDepth	VB3	滤波深度	0-255，0 或 1 时表示不进行均值滤波
WeightRange	VW4	最大秤量范围	
FirCalWeight	VW6	一次校准重量	
SecCalWeight	VW8	二次校准重量	
TareInput	VW10	皮重输入	
MinWeight	VW12	最小秤量范围	一般为 20d，d 为数字阶跃
Step	VB14	数字阶跃	范围：1、2、5、10、20
StandstillTime	VW15	停顿时间	单位：ms
StandstillRange	VW17	停顿范围	
ZeroValue	VW19	零点采样值	
FirCalValue	VW21	一次校准采样值	
SecCalValue	VW23	二次校准采样值	
GWProcessValue	VW25	毛重过程值	
NWProcessValue	VW27	净重过程值	
TWProcessValue	VW29	皮重过程值	
AnalogValueInit	VW31	滤波前采样值	
AnalogValue	VW33	滤波后采样值	
GrossWeight	VW35	毛重	
NetWeight	VW37	净重	
TareWeight	VW39	皮重	
Status_I	VB41	"Weight_Init"状态字节	

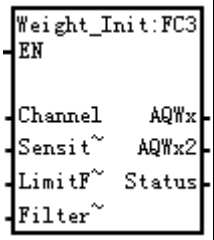
Status_D	VW42	"Weight_Default"状态字	
Status_E	VB44	"Weight_Extend"状态字节	
InternalVariable1	VB45	内部变量 1	
AQWx	VW46	L 区备份	
AQWx2	VW48	L 区备份	
LB21_D	VB50	Weight_Default L 区备份	
LB57_D	VB51	Weight_Default L 区备份	
LB58_D	VB52	Weight_Default L 区备份	
LB59_D	VB53	Weight_Default L 区备份	
LB59_E	VB54	Weight_Extend L 区备份	
LW36_E	VW55	Weight_Extend L 区备份	
LD40_E	VD57	Weight_Extend L 区备份	
Reserved	VB61	保留	
ZeroTraceTime	VD62	零点跟踪计时	
StandBeginTime	VD66	停顿状态判断的起始时间	
ZeroTraceValue	VW70	零点跟踪调整值	

Weight_Init（称重初始化指令）

①指令名称：Weight_Init

②功能：配置称重模块所接传感器灵敏度、低通滤波截止频率、平均滤波深度；

③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
	Channel	IN	BYTE	0~255		通道号
	Sensitivity	IN/ OUT	BYTE	1：1mV/V 2：2mV/V 4：4mV/V 不允许有其它定义	2	传感器灵敏度
	LimitFreq	IN/ OUT	BYTE	3：fg = 5Hz 4：fg = 2Hz 5：fg = 1Hz 6：fg = 0.5Hz 7：fg = 0.2Hz 8：fg = 0.1Hz 9：fg = 0.05Hz 不允许有其它定义	4	低通滤波截止频率
	FilterDepth	IN/ OUT	BYTE	0~255，0 或 1 时表示不进行平均滤波	15	平均滤波深度
	AQWx	OUT	WORD	格式："0x53" + "Sensitivity"		对应称重通道第一个模拟量输出值
	AQWx2	OUT	WORD	格式："LimitFreq" + "FilterDepth"		对应称重通道第二个模拟量输出值
	Status	OUT	BYTE	Bit0：特征值设置错误 Bit1：低通滤波频率设置错误 Bit2：非法通道号 Bit3：加载出厂设置完成		状态字节

注意：该指令通过 SM0.0 调用。

关于传感器灵敏度、低通滤波截止频率和平均滤波深度的参数说明，请参见章节 4.1.4 称重库说明。

Weight_Default（称重标准指令）

①指令名称：Weight_Default

②功能：实现校秤（调零，校准）和测量（去皮）等功能。

③参数说明

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
	Mode	IN	BOOL	--		模式选择 0：基本模式，1：扩展模式
	LoadFacSetting	IN	BOOL	--		装载出厂设置，上升沿有效
	ZeroSet	IN	BOOL	--		设置零点，上升沿有效
	FirCalibrate	IN	BOOL	--		一次校准，上升沿有效
	SeekTare	IN	BOOL	--		求皮重，上升沿有效
	DeleteTare	IN	BOOL	--		删除皮重，上升沿有效
	AnalogValueInit	IN	WORD			滤波前模拟值，对应该称重通道的第一个模拟量输入
	AnalogValue	IN	WORD			滤波后模拟值，对应该称重通道的第二个模拟量输入
	Channel	IN	BYTE			通道号 0-6
	WeightRange	IN/OUT	INT	0~32767	2000	最大称量范围
	FirCalWeight	IN/OUT	INT	大于量程的 5%	2000	一次校准所使用的砝码重量
	ZeroValue	IN/OUT	WORD		5461	零点采样值
	FirCalValue	IN/OUT	WORD		60074	一次校准采样值
	GNWeight	OUT	INT			毛重/净重
	TareWeight	OUT	INT			皮重
	Status	OUT	WORD			状态

注意：关于模式选择、一次校准、皮重等参数说明，请参见章节 4.1.4 称重库说明。

Status 注释：

位	状态名称	适用范围	描述
Bit0	掉电报警	基本/扩展模式	0：模块电源正常，1：模块无电源 Bit0 为 1 时，采样值为 0xFFFF
Bit1	断线报警	基本/扩展模式	0：传感器连接正常，1：传感器断线 Bit1 为 1 时，采样值为 0xFFFE
Bit2	超量程报警	基本模式	1：毛重值≥额定重量，采样值为 0xFFFD
	Max+9e	扩展模式	1：毛重值≥额定重量+9e，此处 e 为数字阶跃
Bit3	已定皮重	基本/扩展模式	1：皮重存储器被占用（皮重过程值≠0）
Bit4	已预设皮重	扩展模式	1：预设皮重
Bit5	1/4d	扩展模式	1：毛重小于±0.25d，此处 d 为数字阶跃
Bit6	停顿	扩展模式	1：停顿状态确定
Bit7	已找到零点	基本/扩展模式	1：已找到零点
Bit8	已校准	基本模式	1：一次校准完成
		扩展模式	1：一次校准完成(二次校准重量为 0)或者一次校准和二次校准完成
Bit9	低重量	扩展模式	1：当前重量小于最小秤量范围

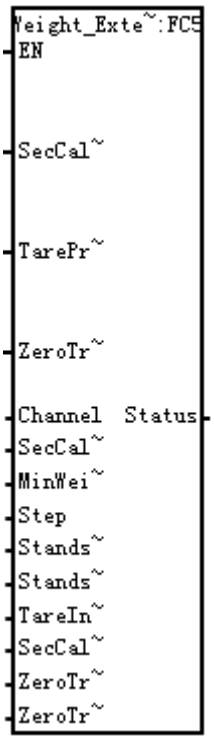
Bit10	命令只能在停顿状态执行	扩展模式	1：扩展模式时，找零点及求皮重必须处于停顿状态
Bit11	命令只允许在已调零状态	基本/扩展模式	1：一次校准或者二次校准前必须已调零
Bit12	命令只允许在已校准状态	基本/扩展模式	1：求皮重或预设皮重时必须处于已校准状态
Bit13	校准砝码重量太小	基本/扩展模式	1：一次校准砝码与零点；二次校准砝码与一次校准砝码之间重量差值不能小于 5%FS
Bit14	非法皮重值	基本/扩展模式	1：皮重值不能小于 0 或者大于最大秤量范围
Bit15	非法通道号	基本/扩展模式	--

Weight_Extend（称重扩展指令）

① 指令名称：Weight_Extend

② 功能：实现二次校准，设置最小重量，设置数字阶跃，停顿状态检测、预设皮重和零点跟踪等功能。

③ 参数说明

库函数	参数名称	输入输出属性	类型	数值范围	出厂值	说明
	SecCalibrate	IN	BOOL	--	--	二次校准，上升沿有效
	TarePreset	IN	BOOL	--	--	预设皮重
	Channel	IN	BYTE	0-6	--	通道号
	SecCalWeight	IN/OUT	INT	0	0	二次校准时的砝码重量(默认为 0，表示不进行二次校准)
	MinWeight	IN/OUT	INT		20	最小重量,默认为 20d,此重量值只能用于具有最小重量以上的规定数字阶跃的校准记录。最小重量取决于所用的传感器类型与型号。
	Step	IN/OUT	BYTE	1/2/5/10/20,不允许其他定义	1	数字阶跃（默认为 1）
	StandstillTime	IN/OUT	INT		1000	停顿时间(默认为 1000)单位：ms
	StandstillRange	IN/OUT	INT		10	停顿范围（默认为 10）
	TareInput	IN/OUT	INT		0	皮重输入，即预设皮重值(默认为 0)
	SecCalValue	IN/OUT	WORD		0	二次校准模拟值(默认为 0)
	ZeroTraceEn	IN	BOOL		--	使能零点跟踪
	ZeroTraceRange	IN/OUT	INT	1~3	--	零点跟踪范围(默认为 1)
	ZeroTraceTime	IN/OUT	INT	1000	--	零点跟踪时间，单位 ms，默认为 1000ms
	Status	OUT	BYTE	--	--	状态字节

注意：关于停顿时间、数字阶跃、零点跟踪、二次校准等参数说明，请参见章节 4.1.4 称重库说明。

Status 注释

位	状态名称	解释
Bit0	二次校准时的砝码重量太小	二次校准时的砝码重量与一次校准时的砝码重量之差小于量程的 5%
Bit1	数字阶跃不正确	数字阶跃只能为 1、2、5、10、20
Bit2	停顿时间不正确	停顿时间必须大于 0
Bit3	停顿范围不正确	停顿范围必须大于 0
Bit4	预设皮重值超范围	预设皮重值不能为负或超过额定称量范围
Bit5	预留	--
Bit6	停顿	停顿状态确定

Bit7	非法通道号	--

5.1.3 称重库说明

本节说明称重库各个指令的重要参数以及使用时的注意事项。

1. 称重配置库 Weight_Config

该库使用 SM0.1 调用，其中，ParaListBase 参数设定用户指定参数表起始地址指针和所用内存范围，每个通道占用 72 个字节的内存；ChanneINum 为用户连接的 SM231 通道数目，该参数只能大于或等于实际使用的通道数目。

2. 称重初始化库 Weight_Init

该库使用 SM0.0 调用，用户可以根据应用需求修改传感器灵敏度 Sensitivity、低通滤波截止频率 LimitFreq、平均滤波深度 Filterdepth，这三个参数在装载出厂设置时会给相应的值。（传感器灵敏度必须在校准前修改，低通滤波截至频率和平均滤波深度任意时刻修改都能生效。）

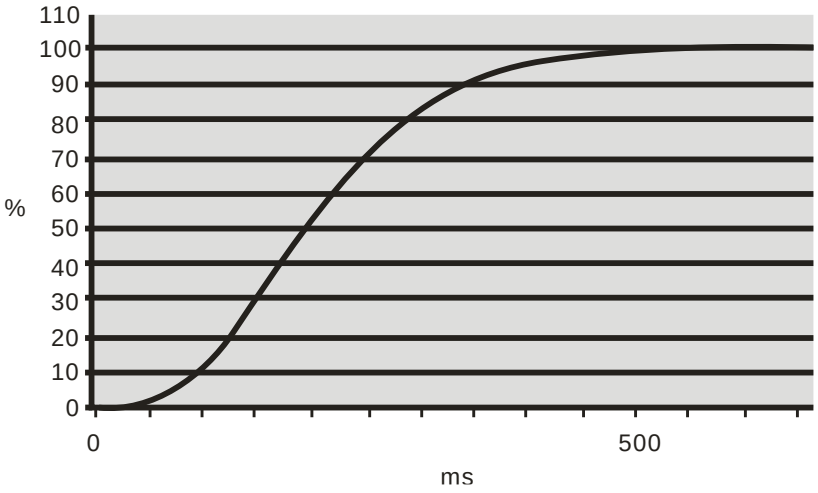
- **传感器灵敏度**

根据所连接称重传感器的特征值，可选择 1mV/V，2mV/V 或 4mV/V。

例如：如果所连称重传感器的特征值为 2.85mV/V，那么下一个较高特征值通常须设为 4mV/V。

- **低通滤波器截止频率**

模块中配备了一个临界条件下导电的低通滤波器，其目的是抑制干扰。下图显示了在 fg=2Hz 时数字低通滤波器的阶跃响应：



为了抑制干扰，必须定义合适的截止频率。称重模块对测量值变化的反应“速度”是通过定义极限频率而设定的。

例如：截止频率为 5Hz 时，称重模块对重量变化的反应会非常快，而截止频率为 0.5Hz 时，相应反应则会迟缓许多。

- **平均滤波深度**

滤波器可平滑稳定称量值，防止干扰。重量值根据 n 个重量平均值测定；传感器采样频率为 50Hz，因此每过 20 毫秒计算一次重量。例如，如果滤波深度 n=10，则表示有 10 个测量值被用来计算平均值。每过 20 毫秒便轮换一次计算值。

如果干扰频率对应于关系“1/（滤波器深度 * 20 ms）的整数倍，那么平均值滤波器也能实现周期性干扰的非常好的阻尼。

3. 称重标准库 Weight_Default

该库使用 SM0.0 调用，使用扩展指令进行第二次校准时，Mode 置 1。第一次使用 231-7WA 模块时需装载出厂设置，相应参数会自动赋值，不需要手动赋值。具体哪些参数需装载出厂设置请参见前文的库指令说明。

- **模式说明**

称重模块提供两种模式的库使用方法。通过配置"Weight_Default'中的参数"Mode"可以选择使用基本模式

或者扩展模式的称重库指令。

1) 基本模式（Mode=0）

（1）只需调用"Weight_Config"、"Weight_Init"和"Weight_Default"3 个库；

（2）支持的功能如下表所示

设置传感器灵敏度、低通截止频率、滤波深度
装载出厂参数
设置最大称量范围
设定零点
一次校准
求皮重
删除皮重
读取滤波前后采样值
掉电报警、断线报警、超量程报警、已调零、已校准、已设皮重等状态指示

2) 扩展模式（Mode=1）

（1）调用"Weight_Config"、"Weight_Init"、"Weight_Default"和"Weight_Extend"4 个库；

（2）支持的功能有：

基本模式下的所有功能
设置最小秤量范围
二次校准
预设皮重
数字阶跃可定义为 1/2/5/10/20
停顿状态确定（设置零点和求皮重时必须处于停顿状态）
Max+9e(GB/T 7724-2008)、已预设皮重、1/4d（GB/T 23111-2008）、停顿、低重量等
零点跟踪

- **皮重**

皮重指商品外包装材料的重量（即运输包装的重量）。

比如，汽车上磅前 GNweight 参数值为汽车重量，用上升沿指令导通 SeekTare 后，GNweight 变为 0，Tareweight 为之前的 GNweight 值。

汽车装满货物后再次上磅时，GNweight 参数值为货物净重。需要指出的是，无皮重时，GNWeight 为毛重；有皮重时，GNWeight 为净重。

- **一次校准**

模块接收的是传感器的电信号，未校准时重量和传感器的电信号无准确的对应关系。

比如在称重台上放置一个 1000g 的砝码，程序中 GNweight 具体的显示值由第一次校准时设定，如果第一次校准时放置 1000g 砝码并将 FirCalweight 设为 2000（200g），则 GNweight 随后会将每 1000g 物体转换为 200g。因此第一次校准非常重要。

4. 称重扩展库 Weight_Extend

该库在扩展模式（Mode=1）下才能使用，需使用 SM0.0 来调用，可用来进行二次校准、预设皮重、设定数字阶跃以及启用零点跟踪功能。

停顿时间

停顿监视用于识别称重台何时能处于一种稳定的平衡状态。如果重量值在一段规定时间（停顿时间）内的变化小于规定的偏差范围（停顿值），则确定称重台停顿。停顿监视用在称重台的静态操作中（主要用于命令：零点采样，求皮重）。

数字阶跃

数字阶跃可以相应地定义为 1，2，5，10 或 20。

零点跟踪

用于消除秤长时间使用后出现的零点漂移，导通则有效，不导通无效。

启用零点跟踪功能时，当毛重/净重的绝对值小于阈并持续一段时间，将当前滤波后模拟量写入零点时模拟量，并将原零点时模拟量保存，毛重/净重输出为 0。

取消零点跟踪功能后，将原零点时模拟值写回。零点跟踪的最大范围为 10d(d 是数字阶跃)，超过最大范围后零点跟踪无效。

预设皮重

TareInput 参数为输入皮重参数，写入一个皮重值后用上升沿导通 TarePreset，Tareweight 参数值则为 TareInput 值，上升沿指令导通 DeleteTare，皮重值变为零。例如：

称重台上不放任何重物，TareInput 值设为 500，导通预设皮重指令 TareInput 后，TareWeight 为 500，上升沿导通删除皮重 DeleteTare 指令后，TareWeight 值变为 0，GNweight 值变为-500。

Weight_Extend 指令中的预设皮重与 Weight_Default 指令中求皮重的不同点与相同点：

不同点：

TareInput 可以任意设置，与是否放置实物无关，设置值即为皮重显示值；

Weight_Default 指令中必须在放置实物后，GNweight 中有值，才能求皮重，皮重值即为 GNweight 值。

相同点：

均通过上升沿导通 DeleteTare 指令来删除皮重，从而回到去皮重之前的状态；

求皮重后，GNweight 值为求皮重前的 GNweight 值减皮重值。

二次校准

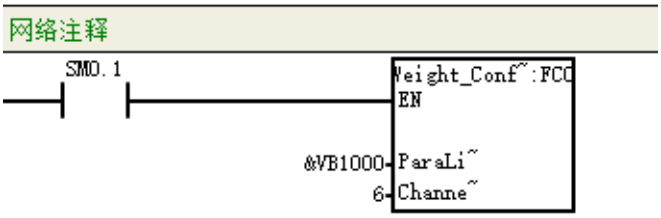
Secselweight 参数为第二次放置砝码值（1000g 砝码设为 10000），注意：二次校准时的砝码重量与第一次校准时的砝码重量之差大于量程的 5%。

一次校准后，待 Weight_Extend 指令中状态字为 64（停顿状态确定）后，在称重台上放置二次校准的砝码。

5.1.4 两款模块称重库使用差异

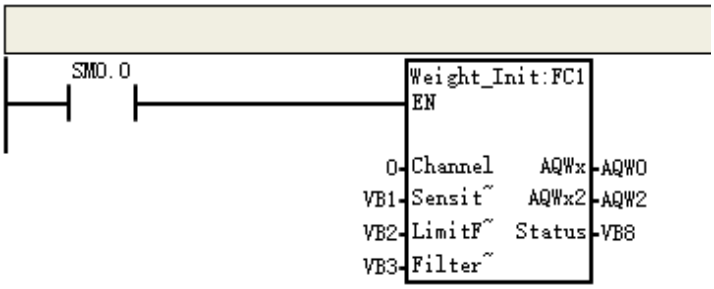
由于 SM231-7WA 和 SM231-7WF 性能和功能上存在差异，故两者在称重库的使用上也有所区别，本节以 ct_em231_7wa_d_lib(V1.2)为例，介绍两款称重模块的称重库使用差异。

1、Weight_Config（称重配置库）



此部分与一路称重模块使用无差别。

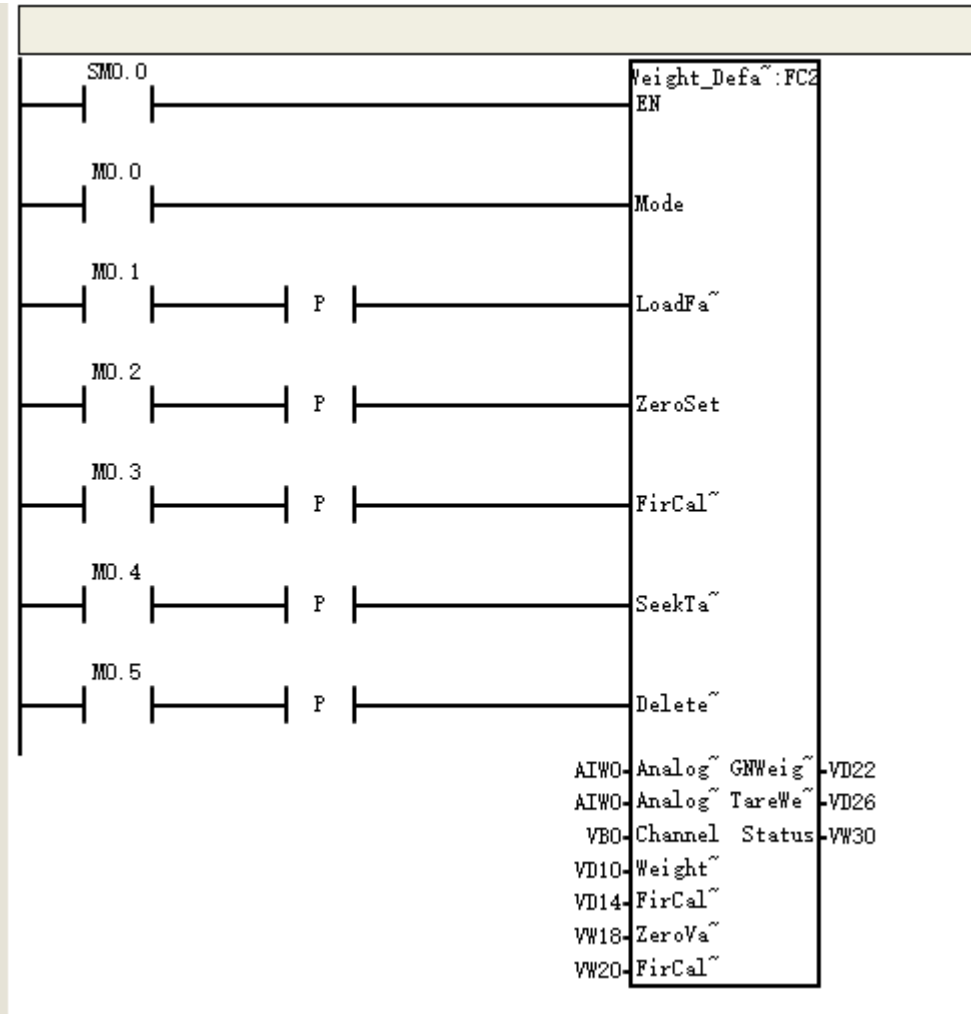
2、Weight_Init（称重初始化库）



SM231-7WF 的六路配置参数只能是一样，一个模块只需调用一次 Weight_Init，并将模块的 AQW 填至对应的“AQWx”和“AQWx2”位置。

FilterDepth：平均值滤波器的深度，范围：0~150（SM231-7WA 使用范围 0~255），超过 150 则默认为 150。

3、Weight_Default（称重标准库）



SM231-7WF 没有滤波前的数据，只有一个滤波后的值，因此，Weight_Default 库在使用时将 AnalogValueInit、AnalogValue 均填上滤波后的值即可，例如第一个通道的数据为 AIW0，则将该值赋给 AnalogValueInit、AnalogValue 即可，Channel 为通道号（范围 0~6），每个测量通道需不一样。

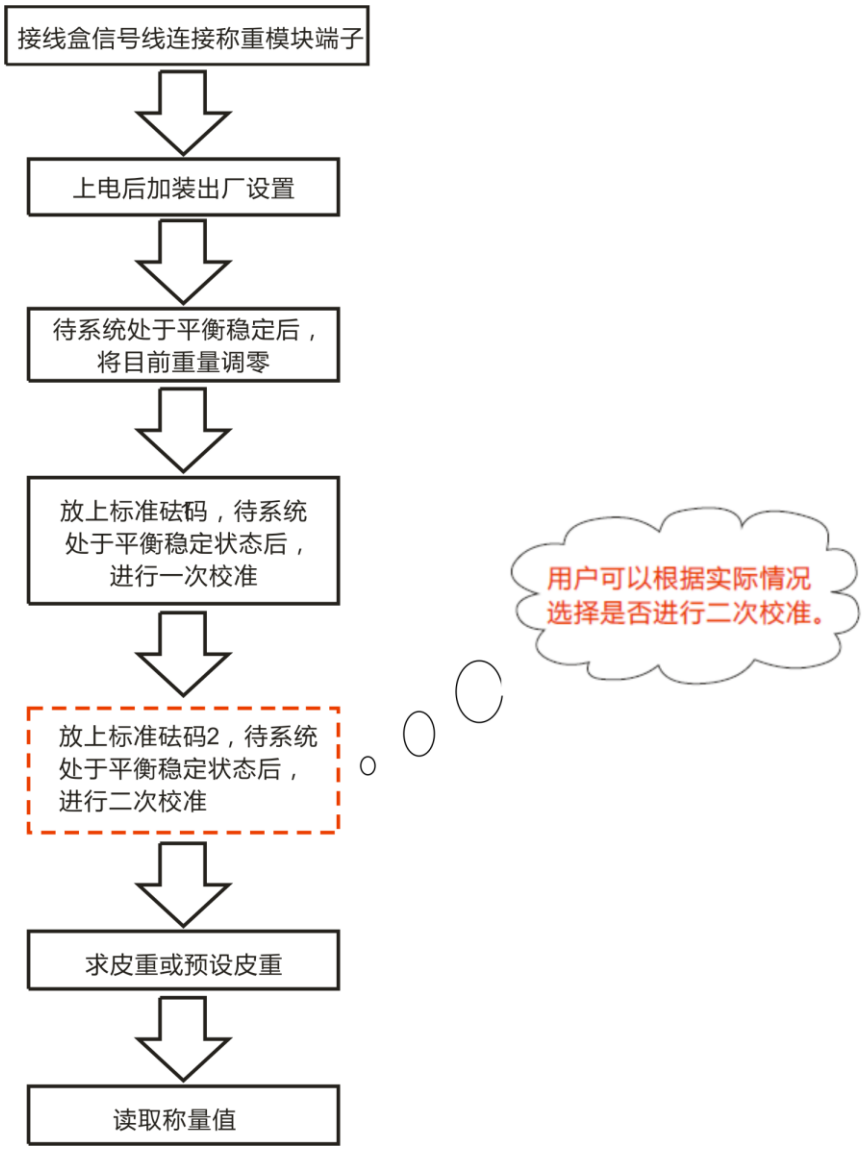
其余使用与两款模块没有区别，六个通道则需调用 6 次“Weight_Default”库，通道号配置不一样。

4、Weight_Extend（称重扩展库）

两款模块没有使用区别。

5.1.5 调零和校准

调零和校准是为了让称重模块与荷重元的重量值相符合，调校步骤如下图所示。



提示

如果 SecCalWeight（二次校准重量）为 0，则默认只进行一次校准。如需进行二次校准，请为 SecCalWeight（二次校准重量）设置一个合法重量值，即 SecCalWeight（二次校准重量）与 FirCalWeight（一次校准重量）之差大于量程的 5%。

将称重模块连接至 CTH200 CPU，在传感器平稳之后通过称重库指令对传感器进行调零和校准。

示例步骤如下：

1）**接线**

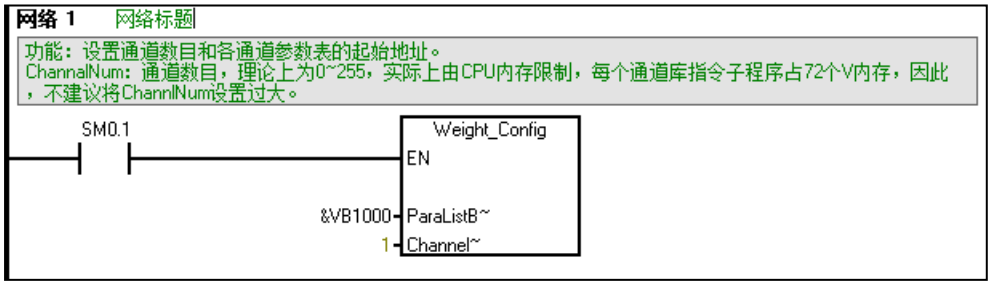
将称重模块连接至主机，传感器连接至称重模块，连接完毕后请按需求给各部件供应电源。

2）**称重库参数配置**

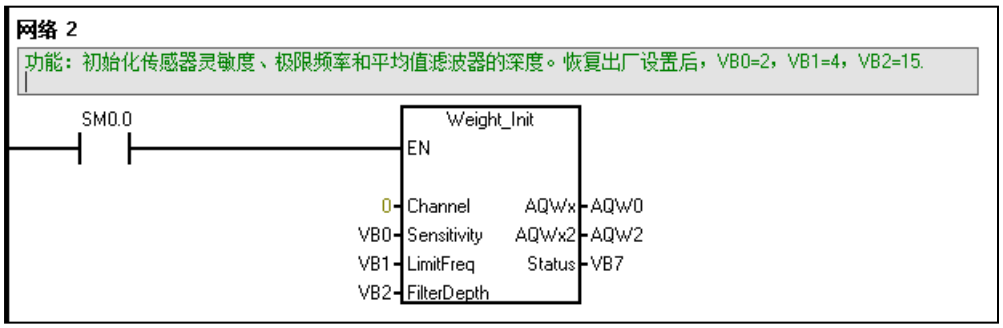
在称重库 EM231_7WA_LIB 中，依照使用者实际测量设置及荷重元规格配置各项参数，本例对传感器进行调零、一次校准和二次校准，称重库参数设定如下：

Weight_Config：

ChannelNum（通道总数）设为 1（此例只挂接了一个 1 路称重模块），该指令在第一个循环周期必须调用一次，用户必须确保设定的通道参数表 V 内存未超过允许范围。

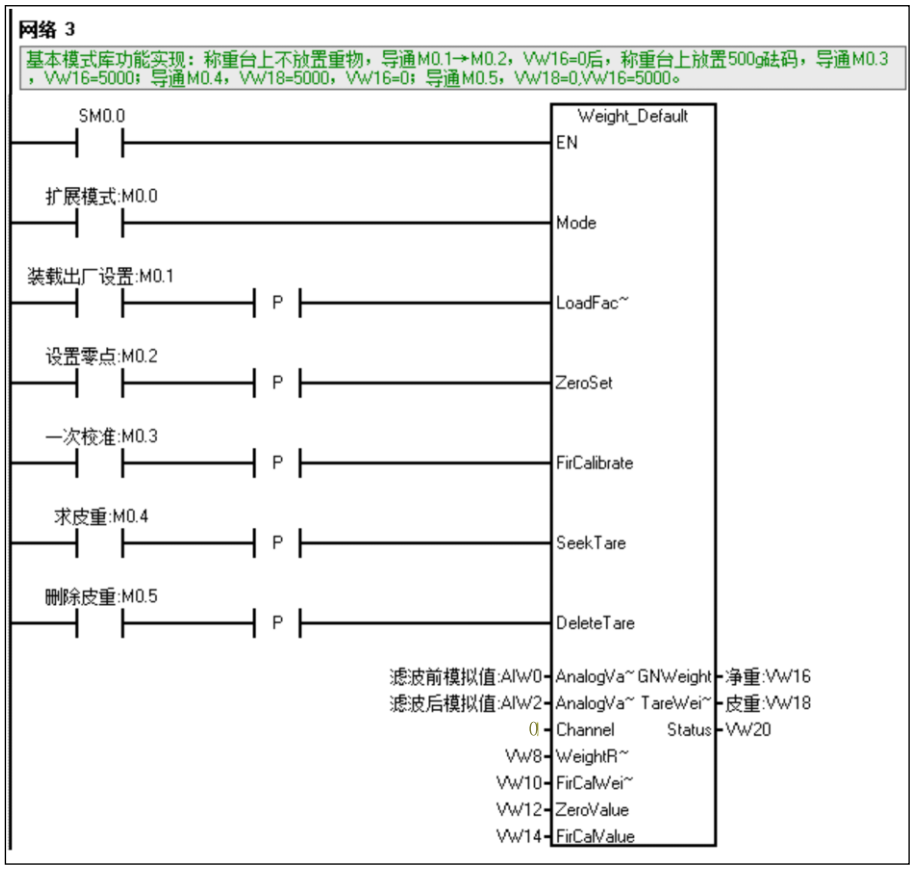


Weight_Init：Sensitivity（传感器灵敏度）设为 2mv/v（可选择 1mv/v、2mv/v、4mv/v），Channel（通道号）设为 0（此例只挂接了一个称重模块 SM231-7WA，最大通道号不能超过组态的通道总数），其他参数保持出厂值不变，具体配置如下图。



Weight_Default：

Channel（通道号）设为 0，WeightRange（量程）设为 30000，FirCalWeight（一次校准重量）设为 5000，具体配置如下图。



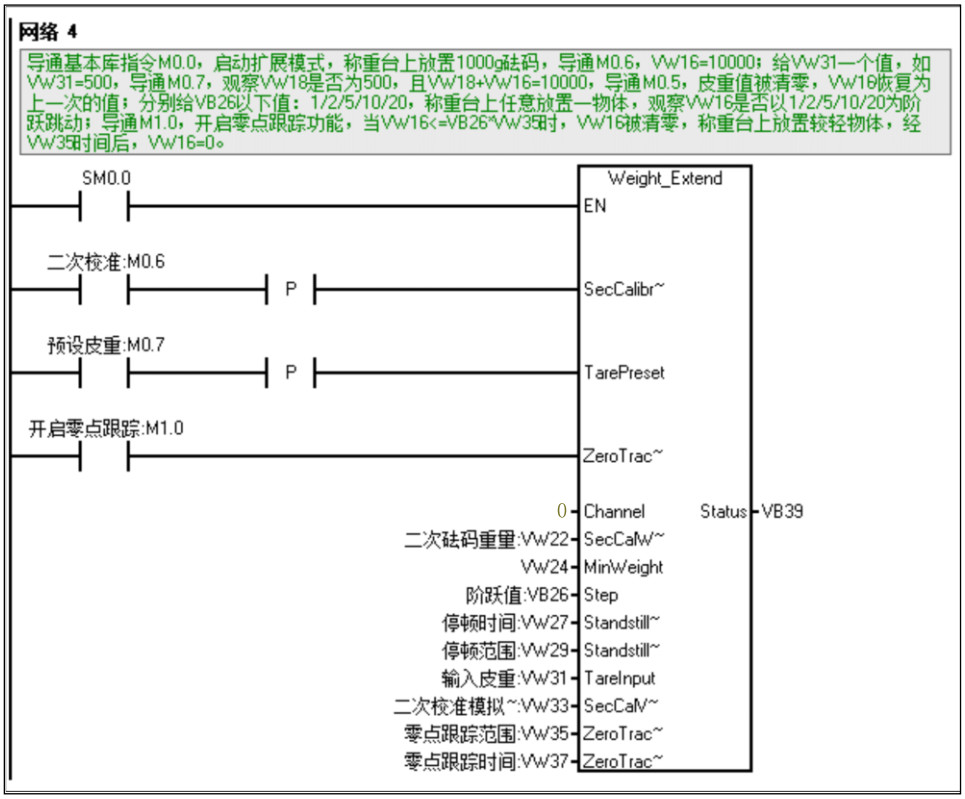
提示

1) FirCalWeight 的设置范围：5%FS~100%FS；

2) 称重分度值为 0.1g，本例的 WeightRange（称重量程）设置值为 30000，即 3KG，FirCalWeight（一次校准重量）设置值为 5000，即 0.5KG，皮重值设为 0。

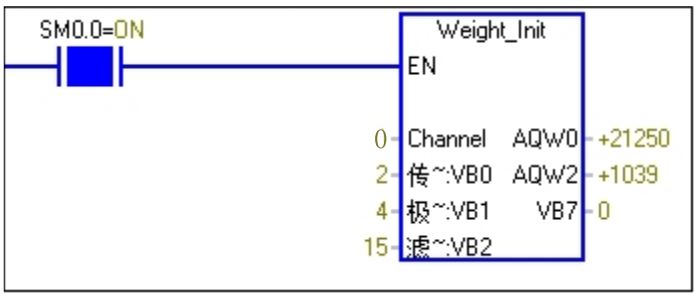
Weight_Extend：Channel（通道号）设为 0，SecCalweight（二次校准重量）设为 10000。

SecCalWeight 的设置范围：5%FS~100%FS，本例中 SecCalWeight 设置为 1KG。具体配置如下图。



3）**使能控制程序**

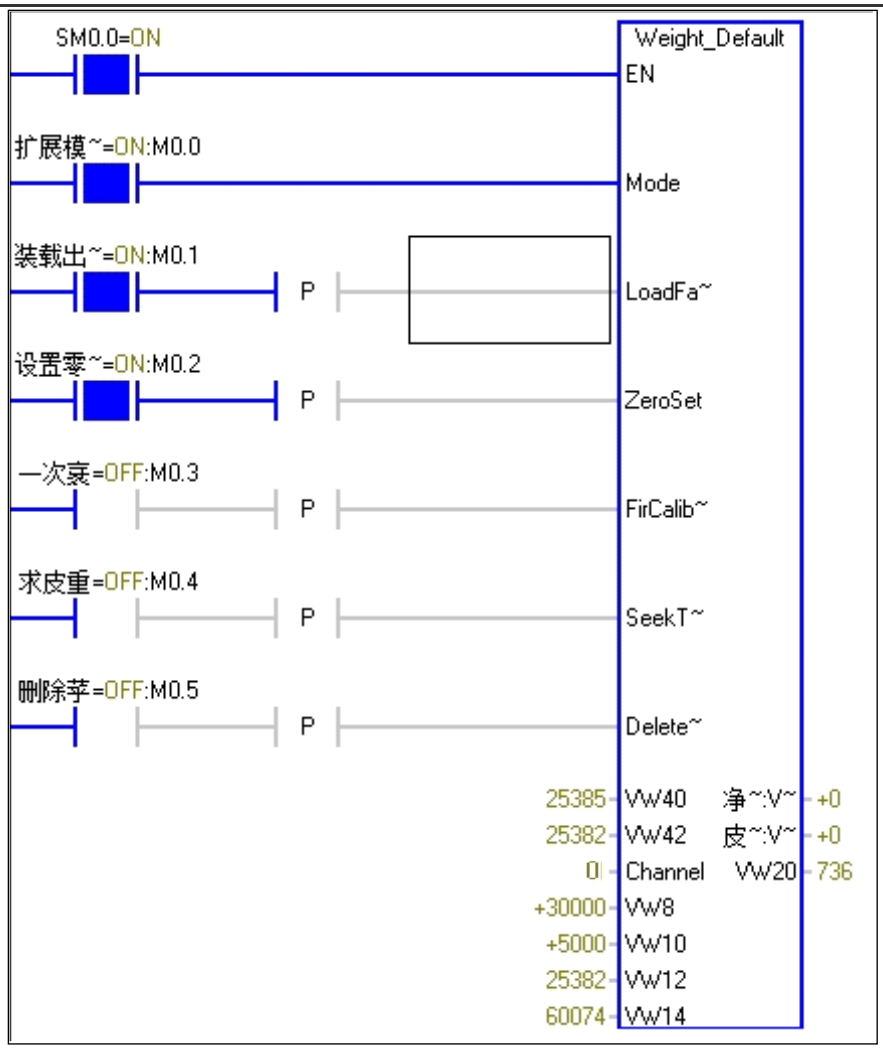
配置设置完成后，请首先使能 Weight_Config 和 Weight_Int 库，然后进行接下来的操作。



4）**调零**

称量物体重量时首先要调零，使用库指令装载出厂设置后也需要调零。未调零前，装载出厂设置后 GNweight 有值。

给参数 ZeroSet（调零参数）M0.2 写入 ON（上升沿触发），则可以对当前称重平台进行调零。如下图程序块所示，执行调零操作后 VW108（GNWeight）归零，即表示调零完成。

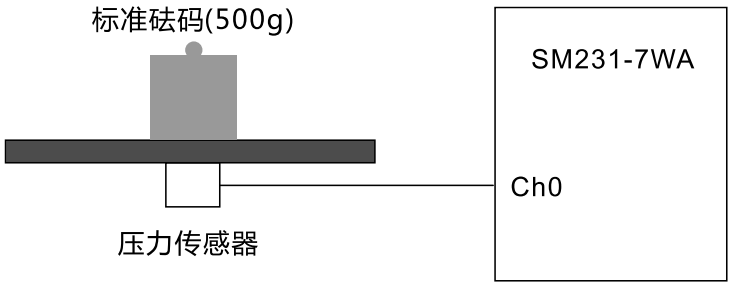


提示

调零时，托盘上不放任何重物，待稳定（即 status 位处于 Bit6 停顿状态）后才可以通过调零操作得到零点模拟值。

5）**一次校准**

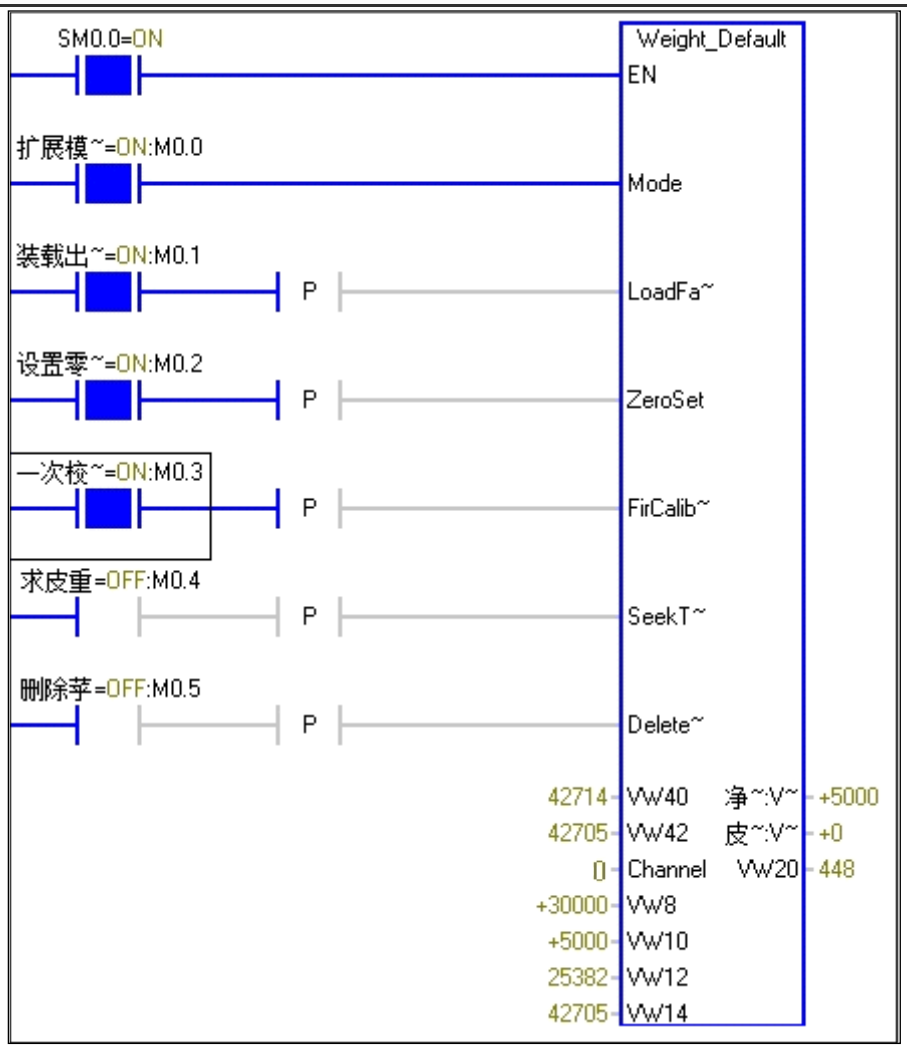
在荷重元上放置一个 500g 的标准砝码。



提示

请参考当前使用的传感器能承受的最大重量放置荷重元，请勿超过量程，否则称重工作将无法正常完成。

给参数 FirCalibrate（一次校准）M0.3 写入 ON（上升沿触发），则完成了一次校准，如下程序所示：



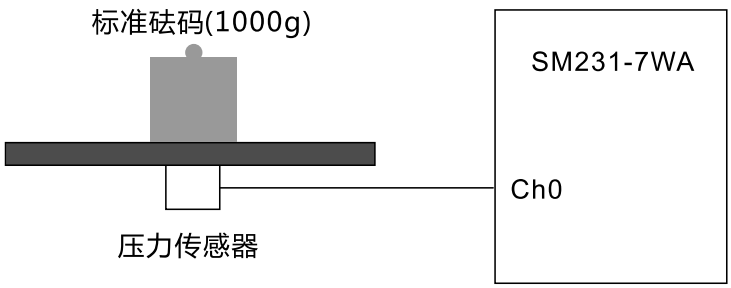
首先放置一个 500g 砝码到称重台上，设置一个最大称量范围 WeightRange，Fircalweight 设为 5000；然后用沿指令导通 FirCalibrate，可以发现 GNweight 值变成 5000。

提示

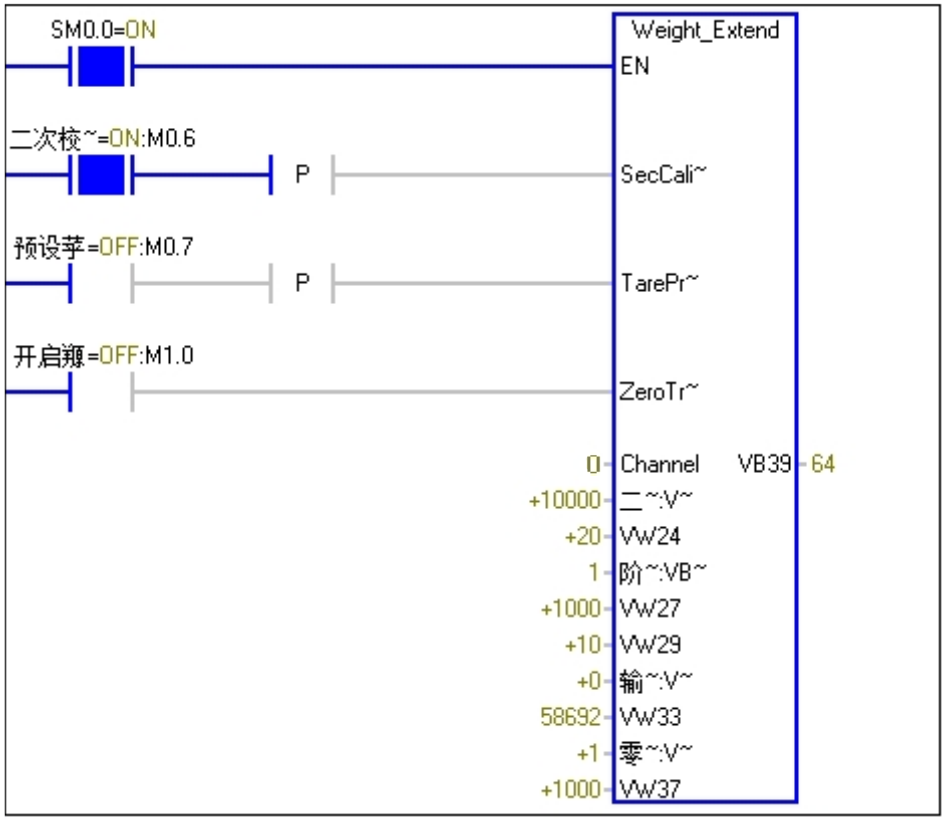
一次校准时，托盘上放有校准用的砝码，待稳定（即 status 位处于 Bit6 停顿状态）后才可以通过一次校准得到校准模拟值。

6）二次校准

请将标准库 Weight_Default 中的 Mode 写入 ON，则可进行二次校准。在荷重元上放置一个 1000g 的标准砝码。



给参数 SecCalibrate（二次校准）M0.6 写入 ON（上升沿触发），完成二次校准。



注意：二次校准砝码重量与第一次校准砝码重量之差应大于量程的 5%。一次校准后，待 Weight_Extend 指令中状态字为 64（停顿状态确定）后，才能在称重台上放置二次校准砝码。此时，Weight_Default 指令中 GNweight 参数值为 10000。

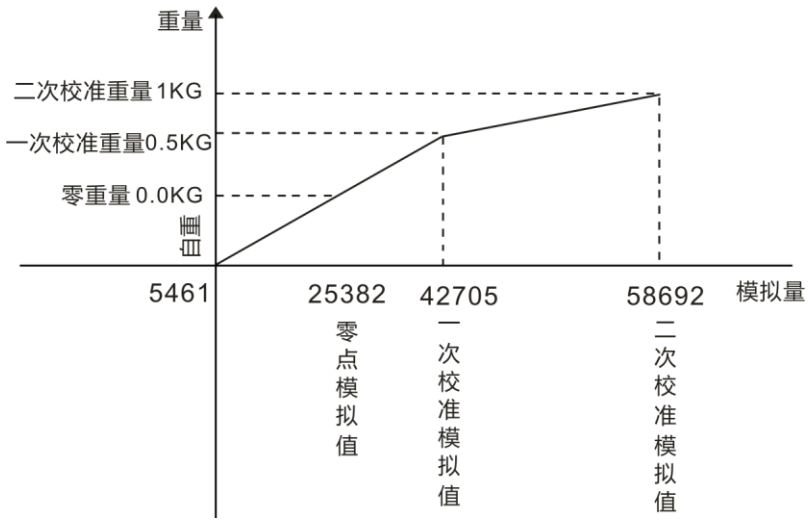
提示

二次校准时，托盘上放有校准用的砝码，待稳定（即 status 位处于 Bit6 停顿状态）后才可以通过二次校准得到校准模拟值。

调校完成后，通过零点模拟值和校准模拟值可定义传感器特征曲线，现在就能在整个量程范围上执行重量值的计算。

零点 = 0.0 KG（始终）	给出的零点模拟值是 25382
一次校准重量 = 0.5KG	给出的一次校准模拟值是 42705
二次校准重量 = 1KG	给出的二次校准模拟值是 58692

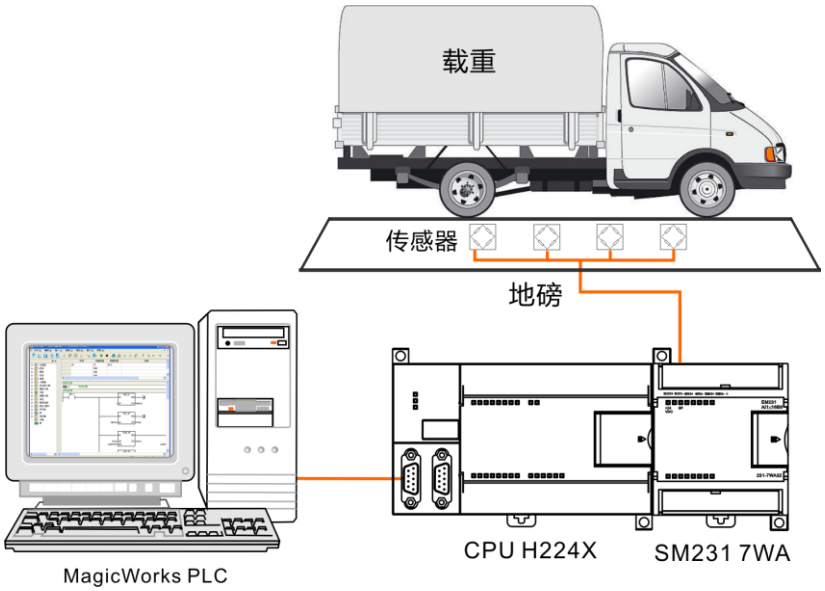
以下示意图显示了本例的模拟值和重量之间的关系。



5.1.6 应用举例

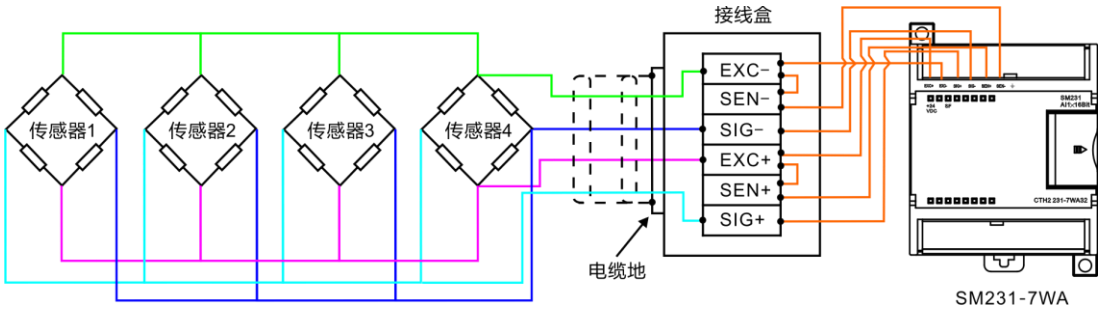
本章物品称重为例介绍称重模块的使用(以 SM231-7WA 为例)。

此例为地磅的应用，主机选用 CTH200 CPU，称重模块 SM231-7WA 用于检测汽车的载重，测量值经主机运算后由称重模块输出至显示系统，用以对汽车运载的货物进行称重计量，连接示意图如下所示：



硬件连接

使用四线式接线方法，传感器通过接线盒连接至称重模块，接线方式如下图：



称重模块 SM231-7WA 接通 24V 电源电压，并经过一个短暂的初始化阶段之后，称重模块将进入到操作状态。如果设备运行正常，那么 LED 指示灯必须处于以下状态：

SF（系统故障）	熄灭状态
+24VDC（电源）	亮起状态

确认模块已经组装完毕，所有连接都已经接好并建立通信，随后可开始执行调校操作。

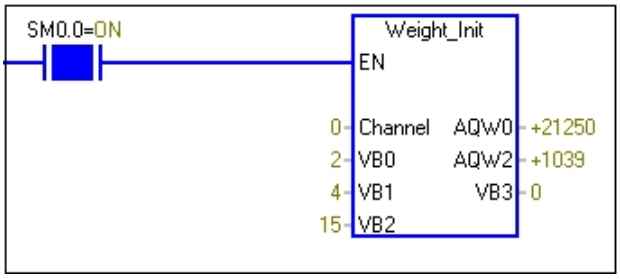
在 MagicWorks PLC 中调校参数

1）安装称重库 EM231_7WA_LIB(V2.4)

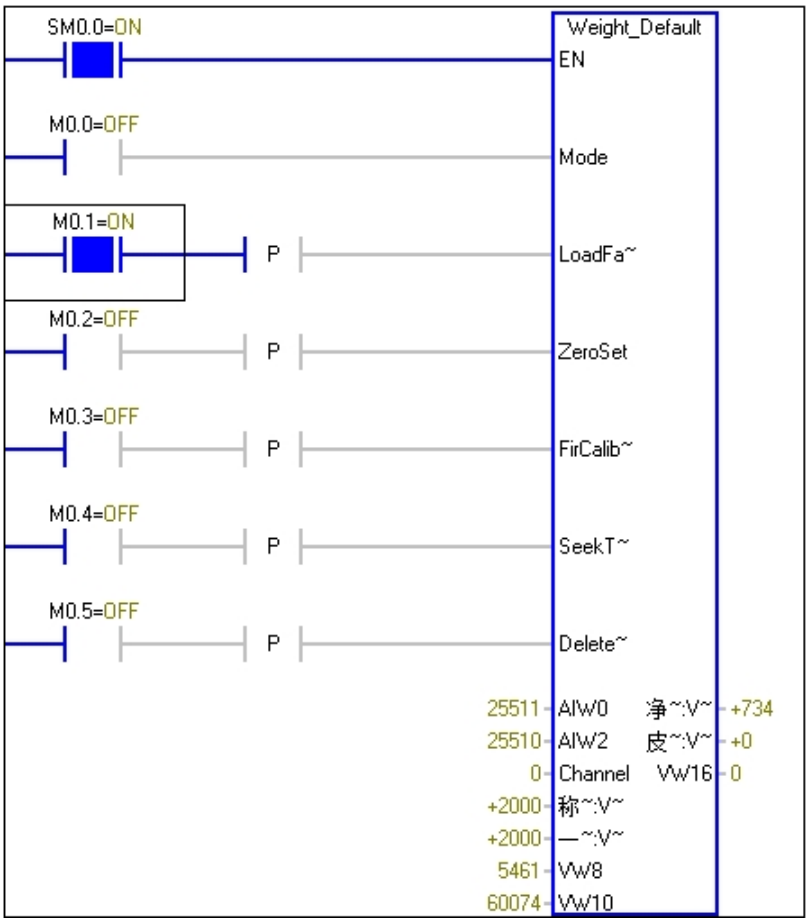
2）在程序中调用称重库

参照章节 4.1.3 库指令功能说明 中的指令说明在程序中调用称重库 EM231_7WA_LIB，通过称重库配置传感器参数并进行调零和校准操作。具体程序运行步骤如下：

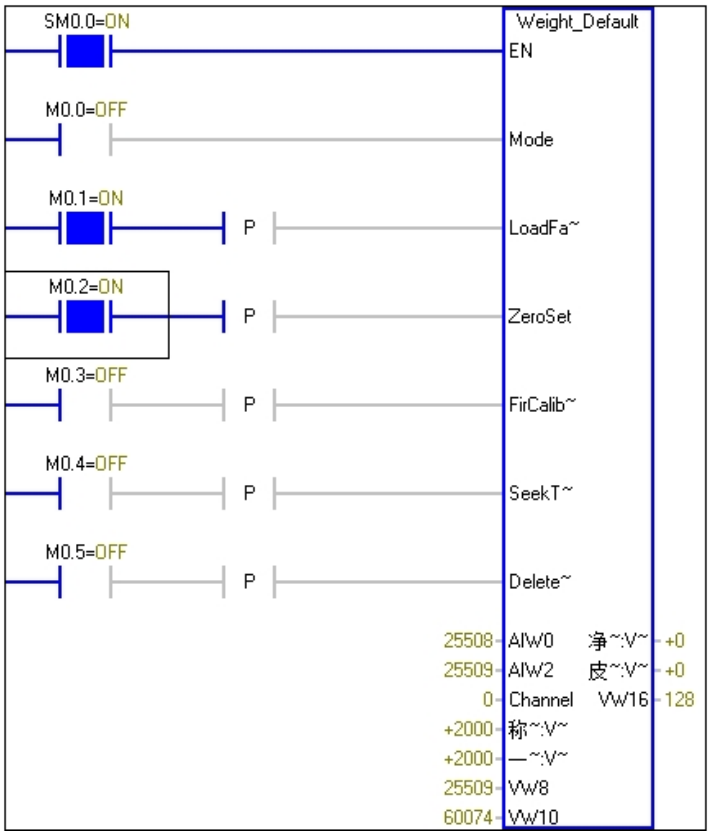
- 启动程序运行


SM0.0=ON
Weight_Init
EN
0 Channel AQW0 +21250
2 VB0 AQW2 +1039
4 VB1 VB3 0
15 VB2

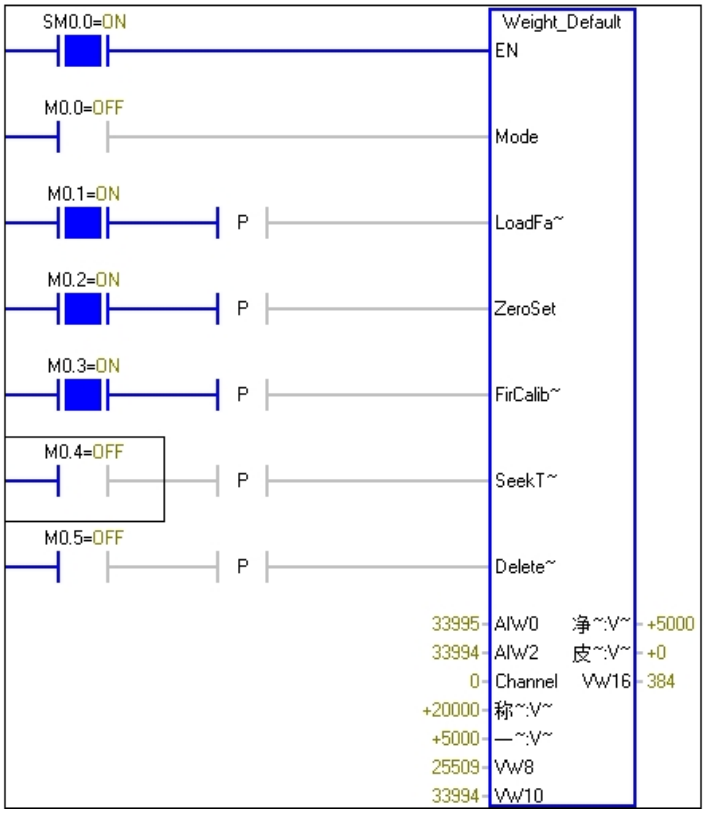
- 恢复出厂设置


SM0.0=ON
Weight_Default
EN
M0.0=OFF
Mode
M0.1=ON
P
LoadFa~
M0.2=OFF
P
ZeroSet
M0.3=OFF
P
FirCalib~
M0.4=OFF
P
SeekT~
M0.5=OFF
P
Delete~
25511 AIW0 净~:V~ +734
25510 AIW2 皮~:V~ +0
0 Channel VW16 0
+2000 称~:V~
+2000 —~:V~
5461 VW8
60074 VW10

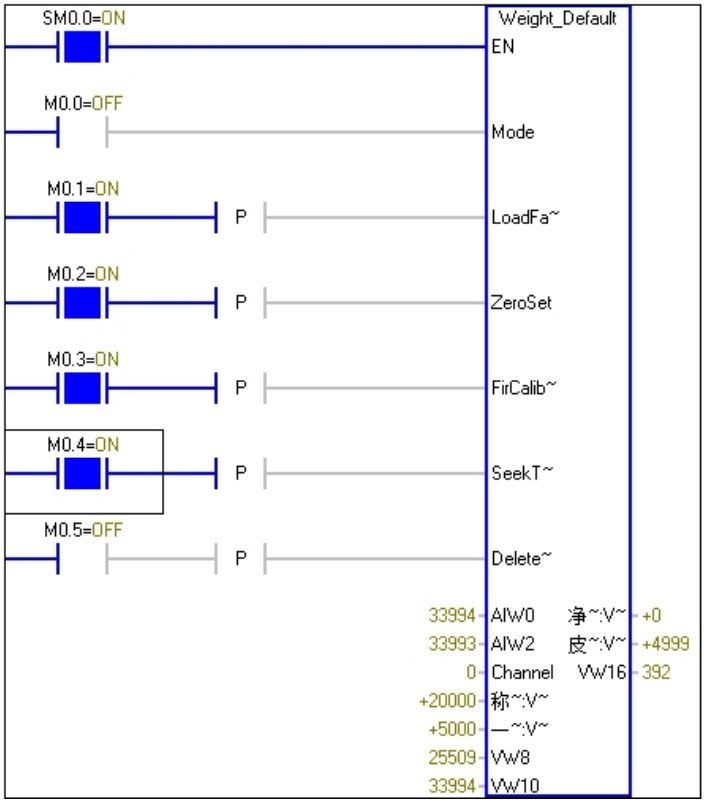
- 设置零点



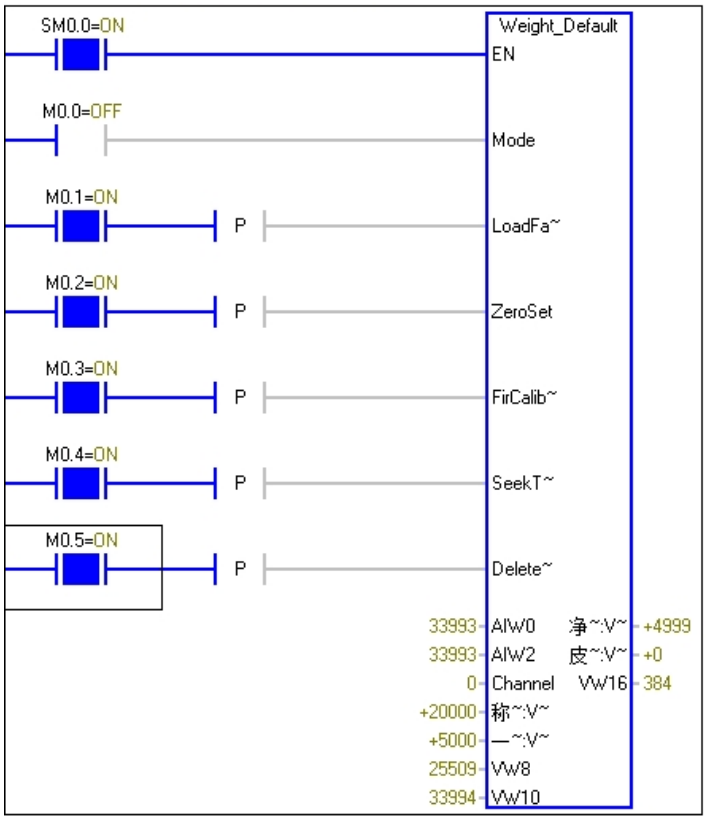
- 一次校准



- 求皮重

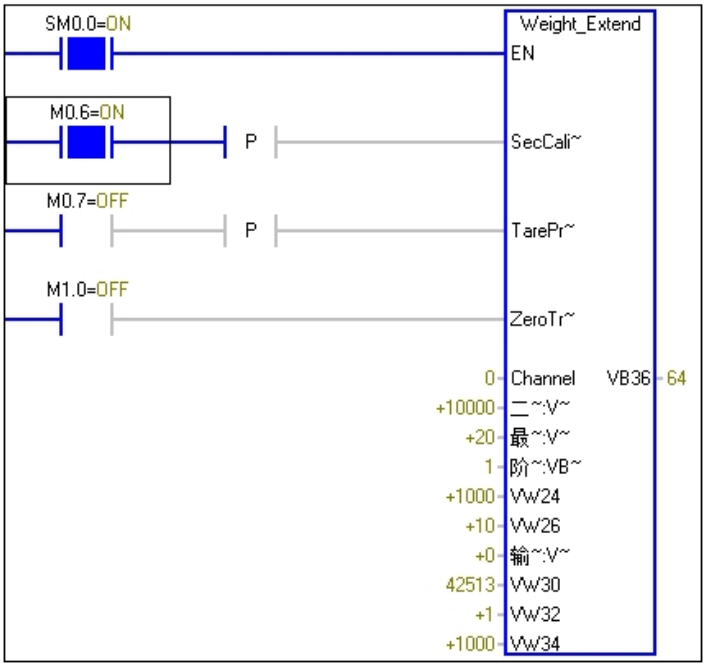


- 删除皮重



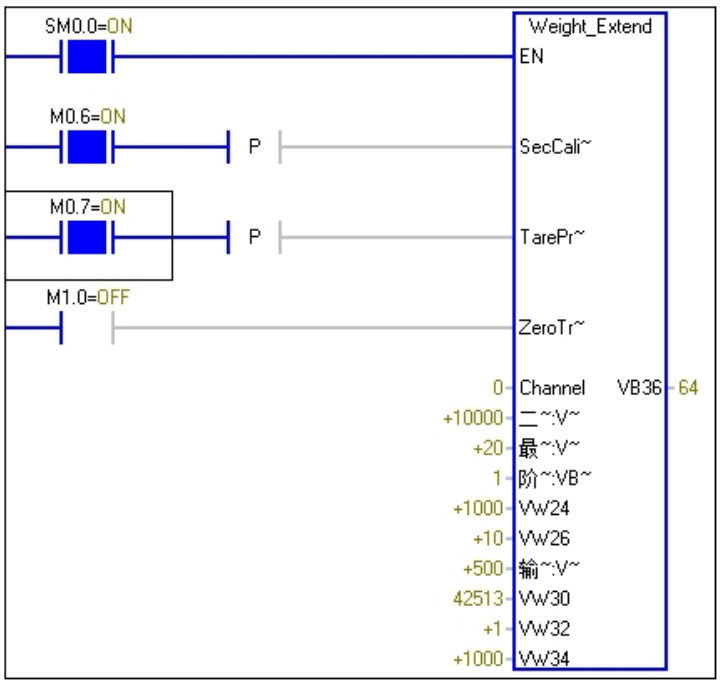
将 Weight_Default 指令中 M0.0 置 1，即可启用扩展模式：

- 二次校准



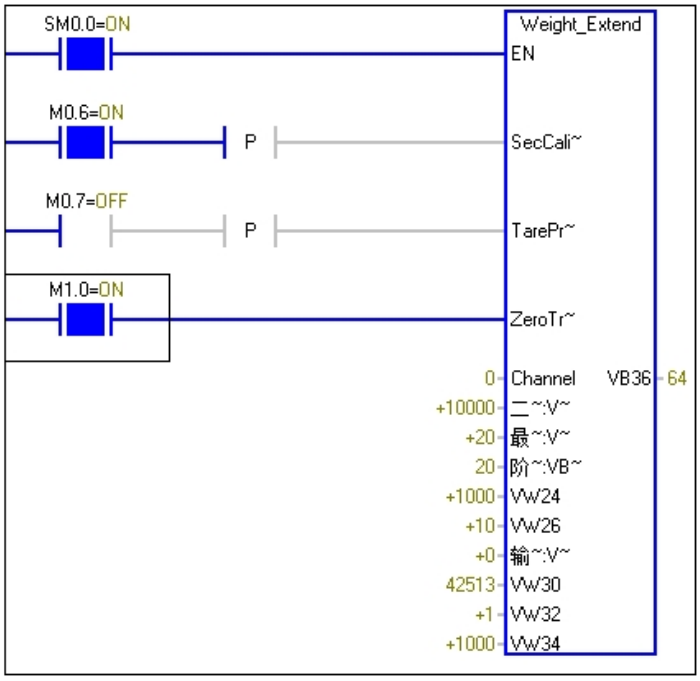
校准完后，Weight_Default 指令中 GNweight 为二次校准砝码值。

- 预设皮重

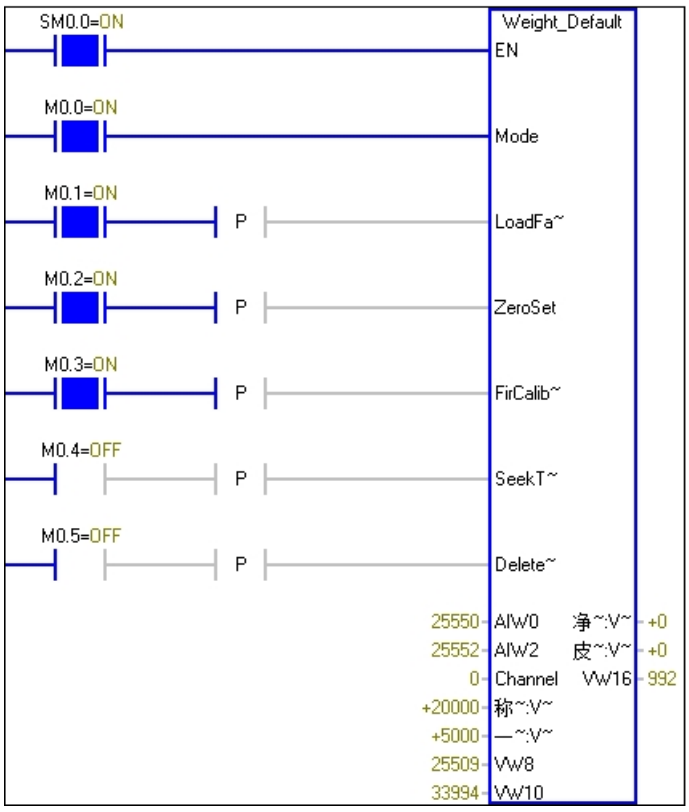


- 零点跟踪功能

阶跃值设为 20，零点跟踪范围设为 1，导通 M1.0 启用零点跟踪功能。



当 GNweighe 小于等于 Step*ZeroTraceRange 时，当前重量值恢复为零。



提示

调零和校准操作须在传感器平稳（即 status 位处于 Bit6 停顿状态）之后进行。

在以上程序中，完成调零和校准后即可执行如下称重任务：

输入皮重

汽车上磅后，GNWeight 参数值即为汽车重量，然后给参数 SeekTare（求皮重）写入 ON（上升沿触发），则将测得的汽车重量设为皮重 TareWeight，此时 GNWeight 参数值清零。

备注：无皮重时，GNWeight 为毛重；有皮重时，GNWeight 为净重。

求皮重

求皮重实质就是当运输包装在称重台上时再设置一次零点，然后将皮重值给 Tareweight，下一次称量不

同物体时需要删除皮重，DeleteTare 上升沿导通，GNweight 值为称重台上物体的重量，Tareweight 参数重新清零。

计算净重

将货物装满汽车后，GNWeight 参数值即为货物重量。

计算毛重

将货物重量 GNWeight 与皮重 TareWeight 相加即为当前汽车和货物的毛重。

扩展数据指令库

6

6.1 PLC 扩展 100K 数据块库“Ext_Mem”

扩展数据空间是 CTSC-200 和 CTH200 CPU 在已有的存储区域之外扩展的 100K 用户可访问的数据存储空间，这块存储空间的数据保持特性同 V 内存数据空间，在 CPU 断电情况下通过超级电容来保持数据，数据最长保持时间可达 100 个小时，数据存储量为 100K 字节。我司专门为这块数据的访问提供了库指令，用户可以将 COTRUST 提供的库添加到 MicroWin 中，通过库中提供的读写指令访问这块数据空间，实现扩展数据空间和其他数据空间的数据交换。

提示

如需使用该库，请前往合信官网下载：www.co-trust.com

6.1.1 库支持的 PLC 型号

CTH200 系列	H224X（老平台高性能型、V5 高性能型升级版） H226XL（老平台高性能型、V5 高性能型升级版） H228XL（老平台高性能型） H226XM（V5 高性能型升级版） H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版）
CTSC 系列	224E（老平台） 224+（老平台） 224I（老平台） 224XP（老平台） 226I（老平台） 226M（老平台） 226L（老平台） 226M-CAN（老平台） 226H（老平台） 224+（V5 平台） 224I（V5 平台） 224XP（V5 平台） 226I（V5 平台） 226M（V5 平台） 226L（V5 平台） 226H（V5 平台）
CTH300-H 系列	H36-01 H32-01 H35-00 H31-00 H56-10 H52-10

6.1.2 指令详解

ReadExtVMem（读取扩展空间数据）

① 指令名称：ReadExtVMem；

② 功能：从扩展数据空间读取数据

③ 参数说明

参数地址	说明	类型	数值范围	备注
Ptr	表示要读到的内存地址指针	DWORD		如：&VB0，&IB0
Offset	表示读取扩展内存中的起始偏移地址	DWORD	0-102399	
Len	表示读取内存长度(字节数)	DWORD		
Err	返回值表示读取是否成功	BYTE		0 表示读取成功，其它表示失败。

WriteExtVMem（扩展空间写入数据）

① 指令名称：WriteExtVMem；

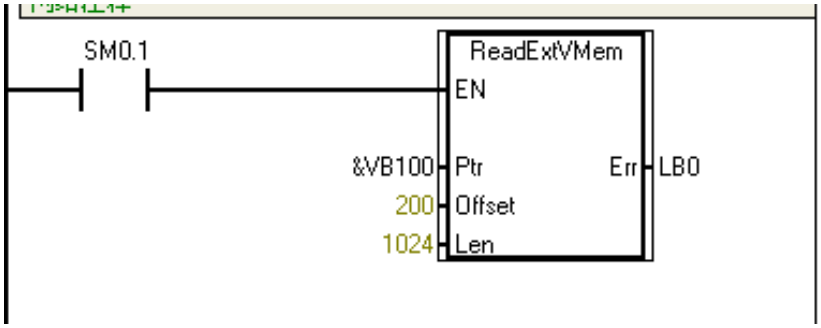
② 功能：向扩展数据空间写入数据

③ 参数说明

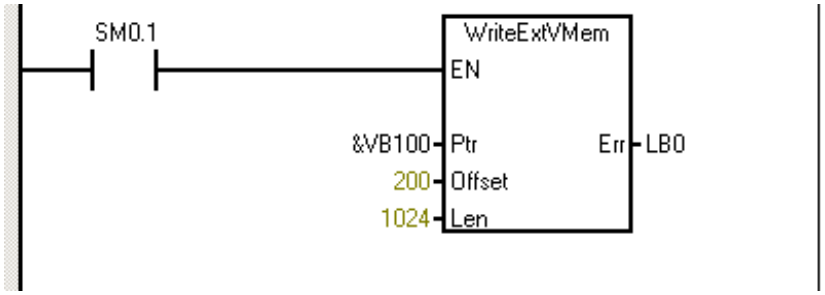
参数地址	说明	类型	数值范围	备注
Ptr	表示要写到扩展数据空间的源数据内存地址指针	DWORD		如：&VB0，&IB0
Offset	表示写到扩展内存中的起始偏移地址	DWORD	0-102399	
Len	表示写入内存长度(字节数)	DWORD		
Err	返回值表示写入是否成功	BYTE		0 表示写入成功，其它表示失败。

6.1.3 应用示例

1、把扩展内存中从偏移量 200 开始的 1024 个字节读到 VB100 开始的内存中



2、把 VB100 开始的 1024 个字节写到扩展内存中从偏移量 200 开始的内存中



6.2 CPU 扩展程序空间

动态库功能块是CTH200 系列CPU 为扩大用户编程程序空间和增加程序保密性给用户提供的一个特殊功能。CTH200 系列 CPU 动态库是提前下载到 PLC 中，应用程序下载时再编译到程序中的功能独立的程序块。CTH200 系列 CPU 可加载两个大小各个 24K 的动态库(“ct_lib1”和“ct_lib2”)。

6.2.1 库支持的 PLC 型号

【动态库的使用范围】

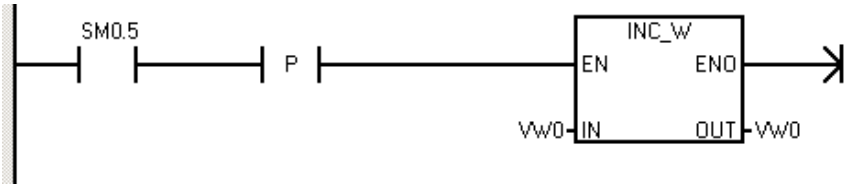
CTH200 系列 CPU 最多可加载两个动态库(“ct_lib1”和“ct_lib2”)，不同类型的 CPU 所支持的动态库及动态库的大小如下表所示：

CPU型号	ct_lib1	ct_lib2
H226L/H226M /H224	允许4K	不支持
H224X	允许24K	不支持
H226XL/H226XM/H228XL/M228IL/M228ML/M228SL	允许24K	
H226IM/H226IL/H226IH	允许64K	

6.2.2 指令详解

【创建动态库】

在工程中，创建所有要作为动态库的程序块，将主程序块的名称命名为 ct_lib1 或 ct_lib2，下载到 PLC 中，在 PLC 中就生成了库函数包括工程中的所有子程序块的动态连接库。



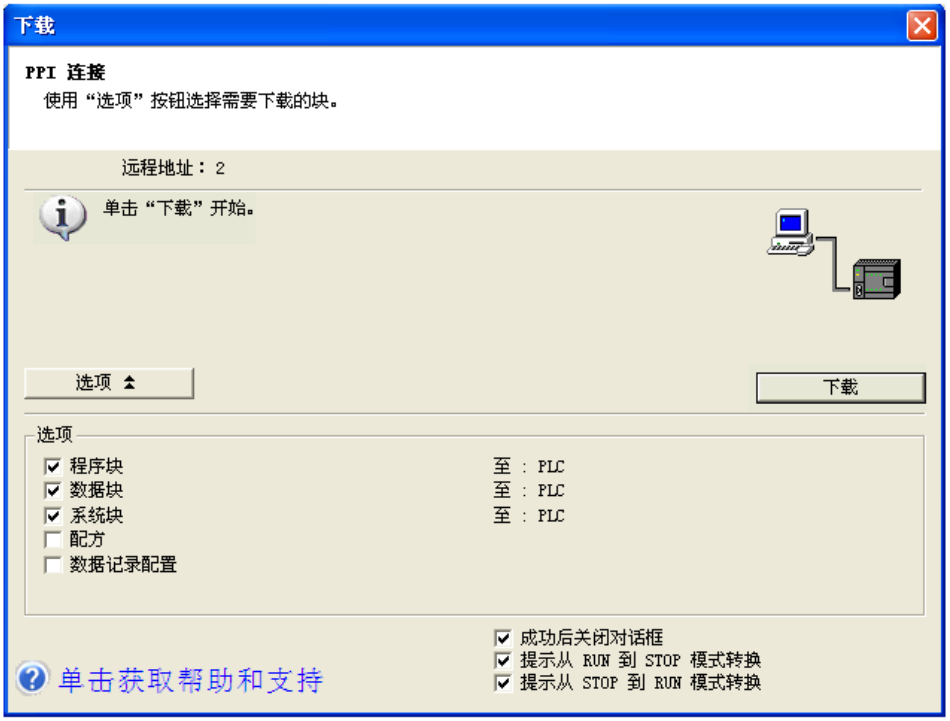
【下载动态库】

将编辑好的动态库程序下到 PLC 中（如下图所示），在 PLC 中就生成了库函数包括工程中的所有子程序块的动态连接库。

每次成功下载了新的动态库之后，PLC 中原来的库和程序块完全被清除。PLC 中生成了名为ct_lib1 的动态库。

提示

在下载动态库时确保只下载程序块。

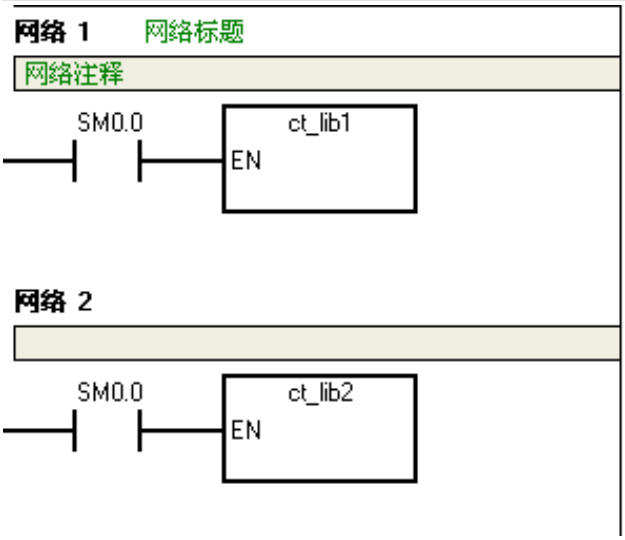


同样的流程适用于下载另一个名为 ct_lib2 的动态库到 PLC 中。

【使用动态库】

在工程中，先创建已经下载到 PLC 中的动态库同名的空的子程序块，如 ct_lib1 或 ct_lib2，在工程中再调用这些空的子程序块，程序下载到 PLC 后，此工程中的空程序块在 CPU 内部会被替换成原下载的与之同名的库函数，运行 CPU 时执行的就是相应动态库里的程序。

提示
使用时最好先加载动态库，然后再加载使用动态库的程序。



【清除动态库】

下载新的动态库时，PLC 中原有的动态库被清除。下载一个主程序名称为 ct_lib1 或 ct_lib2 的空程序块工程到 PLC 中，PLC 中的相应的动态库就会被完全清除。

6.3 永久保存 V 内存功能库“ct_savevmem”

ct_savevmem 作为一个库函数提供给用户使用。其功能是将用户需要保存的一段 V 内存的数据保存到永久性内存中，使得这些数据在很长一段时间内不丢失。

提示

- 适用于参数保存，不可过于频繁保存。
- 请勿在系统块中为需要使用的永久保存 V 内存地址设置断电保持。
- 确保在写完成前 EN 一直处于接通状态，即最好使用 sm0.0 或 act_en 调用。

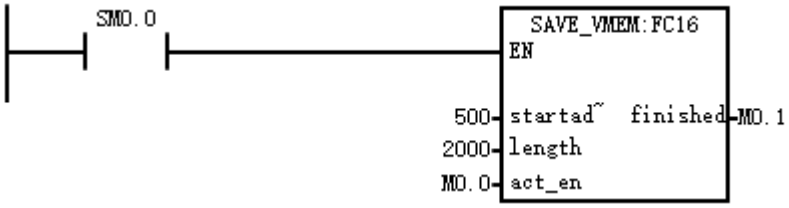
6.3.1 库支持的 PLC 型号

CPU 型号		数据存储空间
CTH200 系列	H224X（老平台高性能型）	基本 8KB，可扩展至 108KB
	H226XL（老平台高性能型） H228XL（老平台高性能型）	基本 10KB，可扩展至 110KB
CTSC 系列	224E（老平台） 224+（老平台） 224I（老平台） 224XP（老平台）	基本 8KB，可扩展至 108KB
	226I（老平台） 226M（老平台） 226L（老平台） 226M-CAN（老平台） 226H（老平台）	基本 10KB，可扩展至 110KB
CTH300-H 系列	H36-01 H32-01 H35-00 H31-00	

6.3.2 指令详解

SAVE_VMEM 指令

① 指令名称：SAVE_VMEM



② 功能：永久保存一段连续的 V 内存数据。程序中同一时间只能有一条库指令导通，例如：程序中有多条 SAVE_VMEM 库指令时，不能使用 SM0.0 来调用此 FC，而应该用上一条的完成位来作为下一条的 EN 和 act_en。

③ 参数说明

参数地址	说明	类型	备注
str_addr	V 内存的起始地址	DWORD	可以是常量或变量。 例如起始地址 VW500 的 str_addr 是 500。

length	长度（以字为单位）	DWORD	要永久保存连续V内存的长度。 例 如 VB500-VB4499 的 length 是 2000 ， 或 VW500-VW4498 的length是2000。
act_en	写允许位	BOOL	此位为 1 时开始写永久内存，写结束后此位自动复位，写入期间此位要保持为 1。
finished	写结束标志位	BOOL	写入开始时此位自动复位，写结束时自动置位为 1。
注：写的总长度为字的整数倍。			

6.4 flash_access_lib（v1.0）库

6.4.1 功能介绍

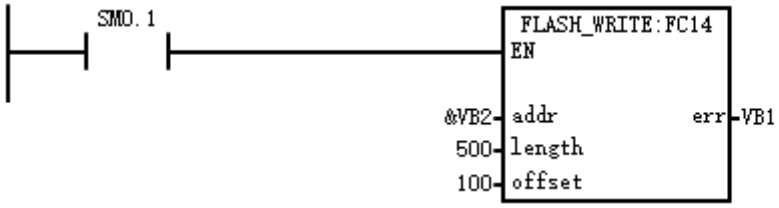
ct_flash_access_lib 提供两个接口，供用户将数据保存到 FLASH 中，或从 FLASH 中读取数据，最多 256KB。

该库适用于 H36-01、H32-01、H35-00、H31-00、H56-10、H52-10。

6.4.2 指令详解

FLASH_WRITE

① 指令名称：FLASH_WRITE；



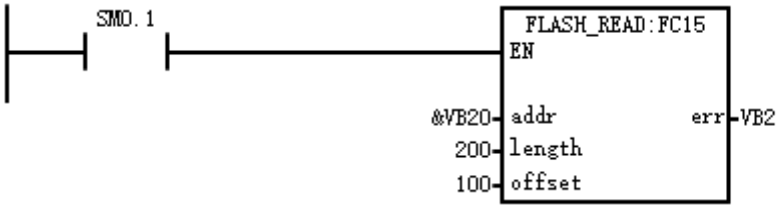
② 功能：将数据保存到 FLASH 中

③ 参数说明

名称	变量类型	数据类型	说明
addr	IN	DWORD	写入 FLASH 的数据起始地址指针
length	IN	DWORD	写入 FLASH 的数据长度
offset	IN	DWORD	写入的数据在 FLASH 块内的偏移量
err	OUT	BYTE	写入错误码 0 无错误 1 超出了用户数据存储块的范围 2 部分数据地址非法 3 从 flash 读取数据失败

FLASH_READ

① 指令名称：FLASH_READ



② 功能：从 FLASH 中读取数据

③ 参数说明

名称	变量类型	数据类型	说明
addr	IN	DWORD	读取的数据起始地址指针
length	IN	DWORD	读取的数据长度
offset	IN	DWORD	读取的数据在 FLASH 块内的偏移量
err	OUT	BYTE	读取错误码 0 无错误 1 超出了用户数据存储块的范围 2 部分数据地址非法 3 从 flash 读取数据失败

高速计数器指令库

7

7.1 HSC_300_LIB 库

7.1 HSC_300_LIB 库

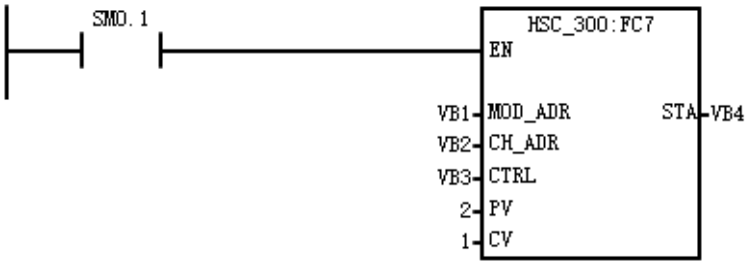
7.1.1 库支持的 PLC 型号

适用于 CTH300-H 系列 H31-00、H32-01、H35-00、H36-01、H56-10、H52-10。

7.1.2 指令详解

HSC_300（设置计数器指令）

函数名：HSC_300



功能：配置计数器参数。

参数说明

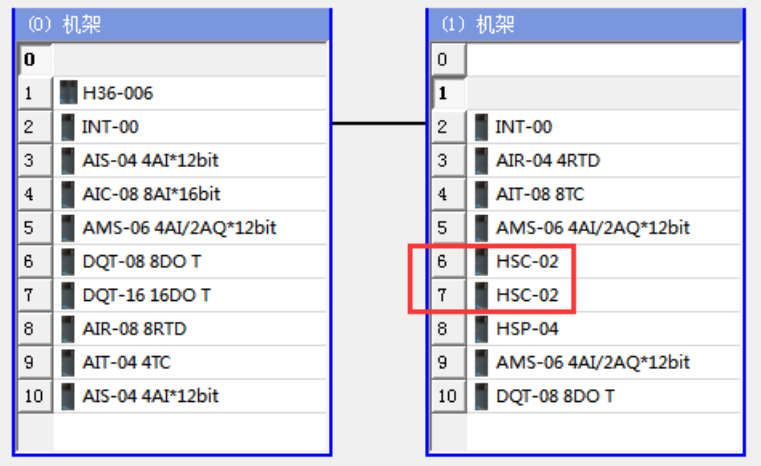
参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道 0：HSC0；1：HSC1	BYTE	0~1	
CTRL	IN	控制字	BYTE		见下表
PV	IN	预设值	DINT		
CV	IN	当前值	DINT		
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

提示

MOD_ADR 模块地址说明了机架号和槽号的取值范围，该参数在多条 HSC 库指令中通用，因此相关设置都是一样。

示例

下图两个 HSC 挂在第二个中继的第 5 和第 6 个模块位置，而 MOD_ADR 是要看硬件组态中的模块所在的机架号以及对应的槽号（16#16 和 16#17），而不是 16#15 和 16#16。



控制字（R/W）

7	6	5	4	3	2	1	0
hsc_en	hsc_cv_update	hsc_pv_update	hsc_dir_update	hsc_dir	quad_rate		reset_level

reset_level：复位电平，1--高电平复位，0--低电平复位

quad_rate[1:0]：正交计数选择，00--4x 倍数，01--2x 倍数，10--1x 倍数

hsc_dir：计数方向，0--减计数，1--增计数

hsc_dir_update：计数方向更新，0--不更新，1--更新

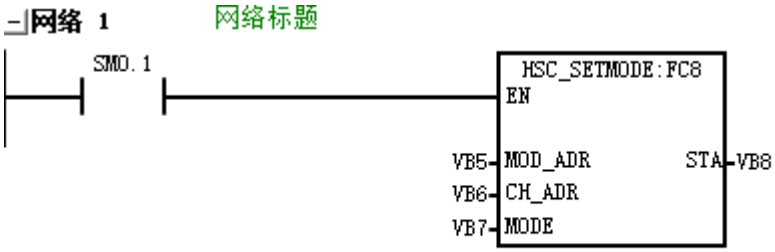
hsc_pv_update：预设值更新，0--不更新，1--更新

hsc_cv_update：当前值更新，0--不更新，1--更新

hsc_en：计数使能，0--不使能，1--使能

HSC_SETMODE（设置模式指令）

函数名：HSC_SETMODE



功能：设置计数器模式

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道	BYTE	0~1	
MODE	IN	计数模式	BYTE		Bit0~Bit3：HSC 计数模式（见下表） Bit4：Z 信号锁存功能，0：锁存，1：不锁存 Bit5：Z 信号清零功能，0：清零，1：不清零 Bit6：预留 Bit7：锁存值清零 0：无效，1：有效
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

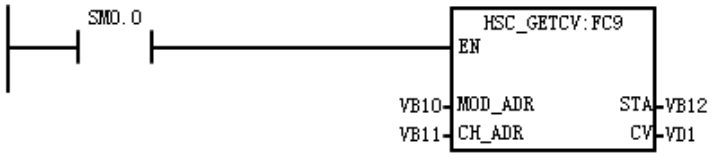
计数器模式说明

模式	描述	输入			软件控制
	HSC0	A0	B0	Z0	INT
	HSC1	A1	B1	Z1	
0	具有内部方向控制的单相计数器	时钟			
1		时钟		重设	
2		时钟		重设	启动（外部同步）
3	具有外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	重设	
5		时钟	方向	重设	启动（外部同步）
6	具有 2 个时钟输入的双相计数器	向上时钟	向下时钟		
7		向上时钟	向下时钟	重设	
8		向上时钟	向下时钟	重设	启动（外部同步）
9	A/B 相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	重设	
11		时钟 A	时钟 B	重设	启动（外部同步）

注意：如果选择计数器模式 2/5/8/11，需通过 INT 控制启动信号才能开始计数。

HSC_GETCV（获取当前计数值指令）

函数名：HSC_GETCV



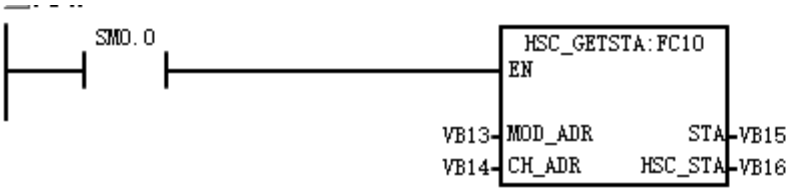
功能：获取当前计数值。

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道	BYTE	0~1	
CV	OUT	当前计数值	DINT		当前计数值
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

HSC_GETSTA（获取当前计数状态指令）

函数名：HSC_GETSTA



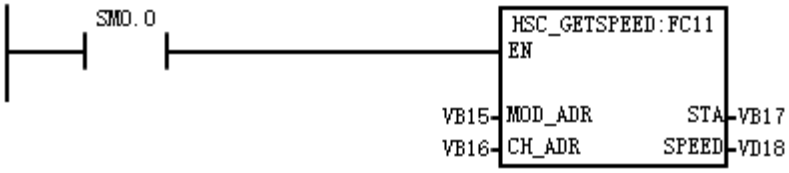
功能：获取当前计数状态。

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道	BYTE	0~1	
HSC_STA	OUT	计数器状态	BYTE		Bit0~Bit3：当前模式 Bit4：预留 Bit5：HSC0当前计数方向位：1=增计数 Bit6=1：当前值等于预设值位 Bit7=1：当前值大于预设值位
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

HSC_GETSPEED（获取当前速度指令）

函数名：HSC_GETSPEED



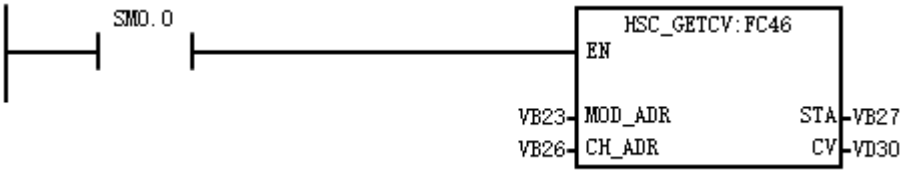
功能：获取当前速度

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道	BYTE	0~1	
SPEED	OUT	当前速度	DWORD		Hz
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

HSC_GETLOCK（获取当前锁存值指令）

函数名：HSC_GETLOCK



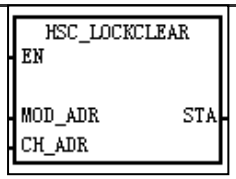
功能：获取当前锁存值。

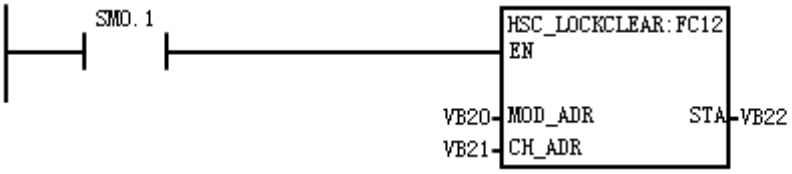
参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道	BYTE	0~1	
LOCKDATA	OUT	当前锁存值	DINT		当前锁存值
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

HSC_LOCKCLEAR（清除锁存值指令）

函数名：HSC_LOCKCLEAR





功能：清除锁存值。

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址 BIT0~BIT3：槽号 BIT4~BIT7：机架号	BYTE		Rackx（0~3）的 y（3~10）槽号上的模块 例如：16#16 为第 1 个机架的 6 号模块。
CH_ADR	IN	通道	BYTE	0~1	
STA	OUT	返回状态	BYTE	0~255	0：OK 1：参数错 2：访问模块出错

高速脉冲输出指令库

8

8.1 hsp_libv1.4

8.1 hsp_libv1.4

8.1.1 库支持的 PLC 型号

HSP_libv1.4 适用于 CTH300 系列 H36-01、H32-01、H35-00、H31-00、H56-10、H52-10。
此外，HSP-04 脉冲输出模块配合 MagicWorks PLC 编程软件中的脉冲输出指令库 hsp_libv1.4 进行使用。

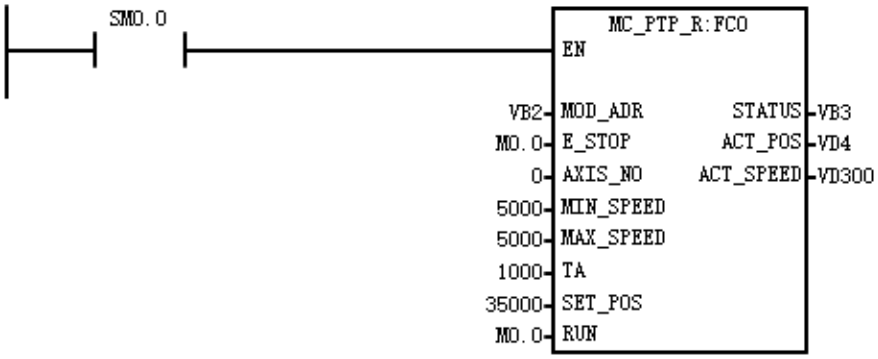
脉冲输出控制指令

指令名称	指令功能
MC_INIT_DIR	配置电机的方向指令
MC_PTP_R	单轴相对运动指令
MC_PTP_A	单轴绝对位置指令
MC_SPEED_CTRL	速度控制指令
MC_SET_POS	设置当前绝对位置
MC_READPOS	读取当前绝对坐标
MC_SET_MODE	设置输出模式指令

8.1.2 指令详解

MC_PTP_R（单轴相对运动指令）

函数名：MC_PTP_R



功能：用作单轴点对点控制（单轴定长驱动）。
调用一次可输出固定脉冲，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡死。

参数说明

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
MOD_ADR	IN	模块地址	Byte	高4位0~3 低4位 3~10	Bit4~Bit7：机架号 Bit3~Bit0：槽号
E_STOP	IN	紧急停止位。 1：有效 0：无效	Bool	0/1	1、只有Run为1与E_Stop为0时运行。 2、当E_STOP为1时，RUN内部复位。
AXIS_NO	IN	设置轴号	Byte	0~3	该参数在运行过程中不能修改。

MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：HZ	Dword	100~4000000	• 最小速度的设定要小于最大速度。 • 此参数在运行过程中可以修改。 • 单轴最大速度设为500K，差分最大速度4M。
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：HZ	Dword	100~4000000	
TA	IN	加速/减速时间。 单位：ms	Dword	0~10000	该参数在运行过程中可以修改
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿X轴的正方向，负脉冲数表示沿着 X 轴的负方向。	Dint	-2147483648 ~ +2147483647	该参数在运行过程中可以修改。 当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准。 当新设定值小于已输出脉冲数，那么会马上停止脉冲输出。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	Bool	0/1	1、只有 RUN 为1与E_STOP为0时运行。 2、当运行完成后，RUN 内部复位。 3、当 E_STOP 为 1时，RUN 内部复位。 4、当指令处于运行状态时，设RUN为0，则实现软停功能。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志	Byte	0~255	Bit0： 1、只对轴参数和TA,MOD_ADR配置错误进行判断； 2、MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。

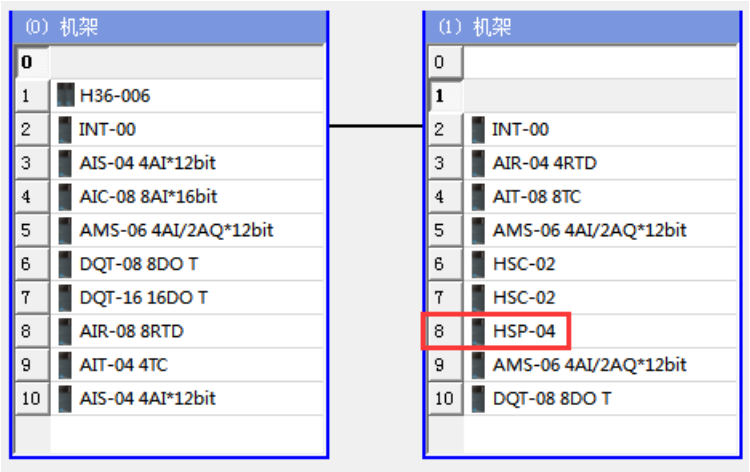
		1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4：模块状态。 1：模块访问出错。 0：模块访问没有出错。 Bit6：被其它指令中止			
ACT_POS	OUT	当前的相对坐标或本指令已输出的脉冲数。	Dint	-2147483648~ +2147483647	
ACT_SPEED	OUT	当前实际运行速度。	Dword	100~4000000	

提示

MOD_ADR 模块地址说明了机架号和槽号的取值范围，该参数在多条 HSP 库指令中通用，因此相关设置都是一样。

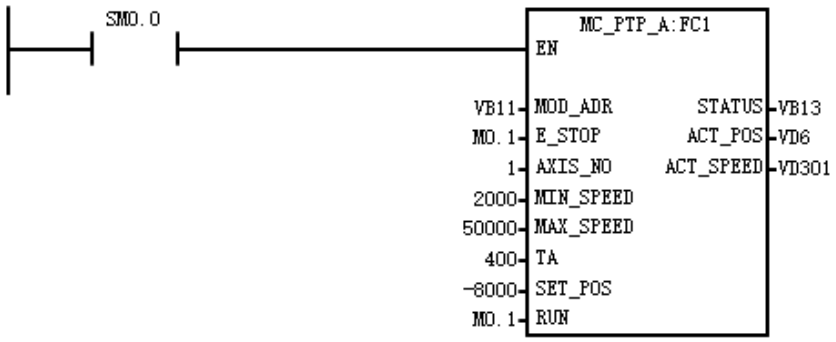
示例

下图 HSP 挂在第二个中继的第 6 个模块位置，而 MOD_ADR 是要看硬件组态中的模块所在的机架号以及对应的槽号（16#18），而不是模块所在位置（16#16）。



MC_PTP_A（单轴绝对运动指令）

函数名：MC_PTP_A



功能：用作单轴点对点控制（非定长，而是定点）。调用一次可在原脉冲数基础上输出脉冲至指定脉冲数，通过最大、最小速度和加减速时间的设定，输出的脉冲在启动时会逐渐的加速到最大的速度，当脉冲数快要跑完时，脉冲的频率会自动减下来，以防止在启动或停止时的机器的惯性太大而引起振动或卡

死。

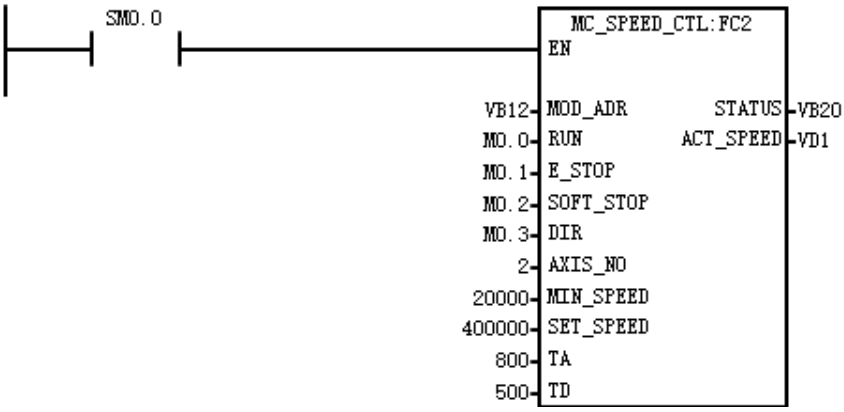
参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址	Byte	高4位0~3 低4位3~10	Bit4~Bit7:机架号 Bit3~Bit0:槽号
E_STOP	IN	紧急停止位。 1：有效 0：无效	Bool	0/1	1、只有RUN为1与E_STOP为0时运行。 2、当E_STOP 为1时，RUN内部复位。
AXIS_NO	IN	设置轴号	Byte	0~3	该参数在运行过程中不能修改。
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：HZ	Dword	100~4000000	1、最小速度的设定要小于最大速度。 2、此参数在运行过程中可以修改。 3、单轴最大速度设为500K，差分最大速度4M。
MAX_SPEED	IN	最大速度，即运行中的最大速度。单位：HZ	Dword	100~4000000	
TA	IN	加速/减速时间。 单位：ms	Dword	0~10000	该参数在运行过程中可以修改
SET_POS	IN	输出的脉冲数，分正负。正脉冲数表示沿X轴的正方向，负脉冲数表示沿着 X 轴的负方向。	Dint	-2147483648 ~ +2147483647	该参数在运行过程中可以修改。 当新设定值大于已输出的脉冲数，那么最后输出的脉冲会以新设定值为准； 当新设定值小于已输出脉冲数，那么会马上停止脉冲输出。
RUN	IN/OUT	运行使能位。 1：有效 0：无效	Bool	0/1	1、只有 RUN ==1与E_STOP ==0 时才能运行。 2、当运行完成后，RUN 内部复位。 3、当E_STOP为1时，RUN内部复位。 4、当指令处于运行状态时，设RUN为0，则实现软停功能。
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。	Byte	0~255	Bit0： 1、只对轴参数和TA,MOD_ADR配置错误进行判断； 2、MIN_SPEED/MAX_SPEED等参数不作报错，会自动设置成一个最接近的合理值。

		0—不运行，因公共资源被其他指令占用，所以指令未运行；或者指令已经运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志 1—忙标志有效，该轴正在被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit4：模块状态。 1—模块访问出错。 0—模块访问没有出错。 Bit6:被其它指令中止			
ACT_POS	OUT	当前的绝对坐标	Dint	-2147483648~+2147483647	
ACT_SPEED	OUT	当前实际运行速度。	Dword	100~4000000	

MC_SPEED_CTL（速度控制指令）

函数名：MC_SPEED_CTL



功能：控制单轴输出脉冲的频率，可任意时候改变输出脉冲的频率（速度）。当接收到软停止命令时，会自动减速停止。当收到紧急停止命令时，会马上停止脉冲输出，不经过减速。

参数说明

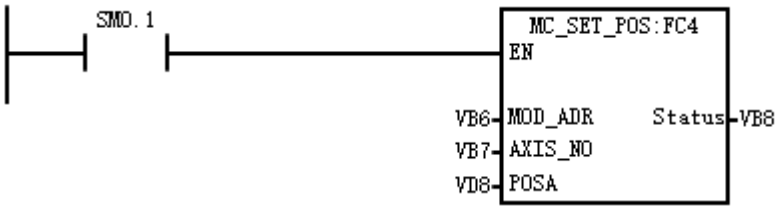
参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址	Byte	高4位 0~3 低4位 3~10	Bit4~Bit7：机架号 Bit3~Bit0：槽号
RUN	IN	运行使能位。 1：有效， 0：无效。	Bool	0/1	1、只有 RUN为1 与E_Stop为0与SOFT_STOP为0时才能运行。 2、当运行完成后，RUN

					内部复位。
E_STOP	IN	紧急停止位。1:有效，0:无效。 当收到有效紧急停止命令后，输出脉冲会马上停止，不经过减速。	Bool	0/1	只有RUN为1 与E_Stop为0与Soft_Stop为0时才能运行。
SOFT_STOP	IN	软停止位。1：有效，0：无效。 当收到有效软停止命令时，输出脉冲会减速停止。	Bool	0/1	只有RUN为1与E_Stop为0 与SOFT_STOP为0时才能运行。
DIR	IN	脉冲的方向位	Bool	0/1	该参数在运行过程中不能修改。
AXIS_NO	IN	设置轴号	Byte	0~3	
MIN_SPEED	IN	最小速度，即启动时或停止时的速度。单位：HZ	Dword	0~4000000	7）设定速度为0时，没有脉冲输出；设定最小速度非0且小于100时，模块默认为100；若设定速度小于最小速度，模块默认将最小速度值修改为设定速度的值。 2、SET_SPEED在运行过程中可修改。 MIN_SPEED在运行过程中可修改。
SET_SPEED	IN	设定速度，在收到停止命令前，输出脉冲会加速或减速到此速度。	Dword	0~4000000	
TA	IN	加速时间，从最小速度到设定速度的加速时间。 单位：毫秒	Dword	0~10000	该参数在运行过程中可以修改。
TD	IN	减速时间，从设定速度到最小速度的减速时间。 单位：毫秒	Dword	0~10000	
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志 1—参数配置错误 0—参数配置正常 Bit1：运行标志 1—正在运行，该指令正在输出脉冲，且指令未执行完。 0—不运行，因公共资源被其他指令占用，所以指令还未运行；或者指令已运行完毕。 Bit2：完成标志 1—完成，指令执行完毕。 0—未完成，指令未执行或指令正在执行中但未完成。 Bit3：忙标志 1—忙标志有效，该轴正在	BYTE	0~255	Bit0： • 只对轴参数和TA/TD,MOD_ADR配置错误进行判断； 2、MIN_SPEED/ SET_SPEED等参数不作报错，会自动设置成一个最接近的合理值。

		被其它指令占用。 0—忙标志无效，指令正在执行或此执行已完成。 Bit5~Bit6：预留 Bit7：指令通信状态标志 1—通信超时；0—无超时			
ACT_SPEED	OUT	当前速度（频率）输出。	Dword	100~400000	该值可能跟实际值会有一点偏差。

MC_SET_POS（设置当前绝对坐标）

函数名：MC_SET_POS



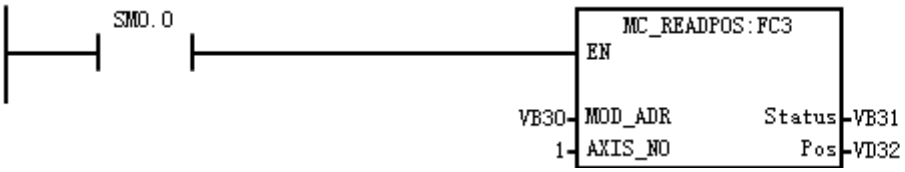
功能：设置当前绝对坐标，本指令只能在轴停止时使用。

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址	BYTE	高4位0~3 低4位3~10	Bit4~Bit7：机架号 Bit3~Bit0：槽号
AXIS_NO	IN	设置轴号	BYTE	0~3	
POSA	IN	设定的坐标		-2147483648~ +2147483647	-
Status	OUT	状态	BYTE	0~255	0：OK 1：参数出错 2：访问模块出错

MC_READPOS（获取当前绝对位置指令）

函数名：MC_READPOS



功能：读取当前位置的绝对坐标值指令。

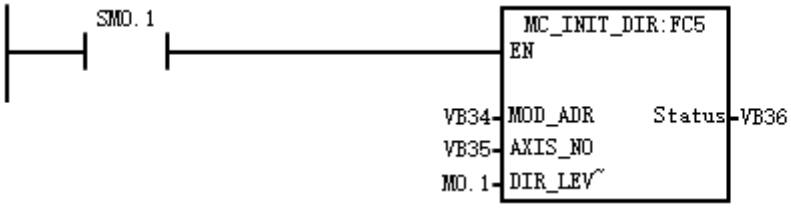
参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址	Byte	高4位0~3 低4位3~10	Bit4~Bit7：机架号 Bit3~Bit0：槽号
AXIS_NO	IN	设置轴号	Byte	0~3	该参数在运行过程中不能修改。

STATUS	OUT	状态	Byte	0~255	0：OK 1：参数出错 2：访问模块出错
POS	OUT	当前的绝对坐标	Dint	-2147483648 ~+2147483647	

MC_INIT_DIR（设置电机方向指令）

函数名：MC_INIT_DIR



功能：配置电机的方向。

提示

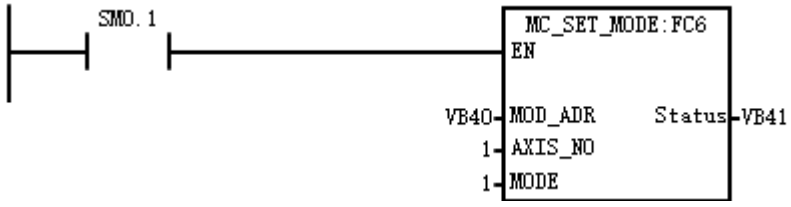
执行此指令只在 CPU 上电第一个扫描周期执行一次。

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址	Byte	高4位0~3 低4位 3~10	Bit4~Bit7：机架号 Bit3~Bit0：槽号
AXIS_NO	IN	设置轴号	Byte	0~3	该参数在运行过程中不能修改。
DIR	IN	配置方向信号为正向时的有效电平 DIR为1时，设置对应方向轴输出“1”时为电机正转 DIR为0时，设置对应方向轴输出“0”时为电机反转	Bool	0~1	默认值：0，即默认方向轴输出为“1”，为电机正转。
STATUS	OUT	状态	Byte	0~255	0：OK 1：参数出错 2：访问模块出错

MC_SET_MODE（设置轴输出模式）

函数名：MC_SET_MODE



功能：设置轴输出模式

参数说明

参数名	输入输出属性	参数描述	类型	数值范围	备注
MOD_ADR	IN	模块地址	Byte	高4位0~3 低4位3~10	Bit4~Bit7：机架号 Bit3~Bit0：槽号
AXIS_NO	IN	设置轴号	Byte	0~3	
MODE	IN	设定的模式	Byte	0，1，2	0：方向 + 脉冲 1：正负脉冲 2： AB相模式
Status	OUT	状态	Byte	0~255	0：OK 1：参数出错 2：访问模块出错

脉冲宽度调制指令库

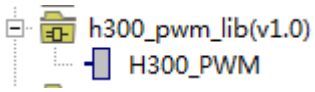
9

8.1 h300_pwm_lib 库

9.1 h300_pwm_lib 库

9.1.1 库支持的 PLC 型号

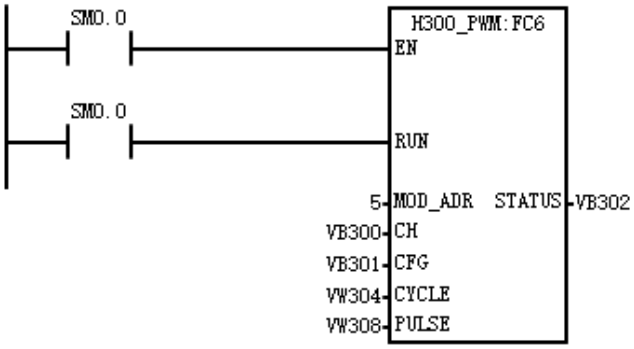
适用于豪精系列，其余型号 PLC 均不支持。



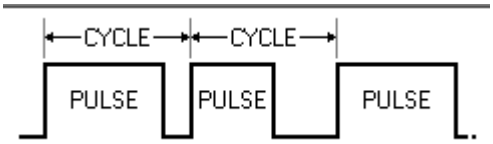
9.1.2 指令详解

H300_PWM（冲宽度调制指令）

① 函数名：H300_PWM



② 功能：通过设置周期和占空比参数，可以输出不同周期和占空比的脉冲。



③ 参数

参数名	输入输出属性	参数描述	数据类型	数值范围	备注
MOD_ADR	IN	模块地址 高4位：0~3 低4位：3~10 Bit4~Bit7：机架号 Bit3~Bit0：槽号	Byte	0~255	运行过程中不能修改。
CH	IN	设置轴号（每个模块有4轴，轴号范围由运控模块数目决定）	Byte	0~（N-1）（N为每个模块所带轴数）	运行过程中不能修改。
CFG	IN	基准时间单元 0：1us，1：1ms	Byte	0~1	运行过程中不可修改。
CYCLE	IN	脉冲的周期	Word	2~65535	运行过程中修改。
DUTY	IN	脉冲的占空比	Word	0~65535	运行过程中可修改。
RUN	IN	运行使能	Bool	0~1	
STATUS	OUT	输出状态字节： 7 6 5 4 3 2 1 0 Bit0：参数配置错误标志	Byte	0~255	Bit0： ● 只对轴参数配置错误进行判断；

		1：参数配置错误 0：参数配置正常 Bit1：运行标志 1：正在运行，该指令正在输出脉冲，且指令未执行完。 0：不运行，因公共资源被其他指令占用，所以指令还没得以运行；或者指令已经运行完毕。 bit4：模块状态。 1：模块访问出错。 0：模块访问没有出错。 Bit6：被其它指令中止			● CYCLE/DUTY 等参数不作报错，会自动设置成一个最接近的合理值。

PWM 模块在硬件组态中使用 DQT-16 模块。

PID 库

10

10.1 CPU 嵌入式 PID 库

10.1 CPU 嵌入式 PID 库

PID 功能块集成在 CPU 内部，不占用用户数据空间，作为一个库函数提供给用户使用。PID 主要针对温度控制的智能 PID 功能，带有自整定、自适应功能，用户无需复杂编程，只需调用和设置一些简单的参数就可以使用，温度控制准确。

提示

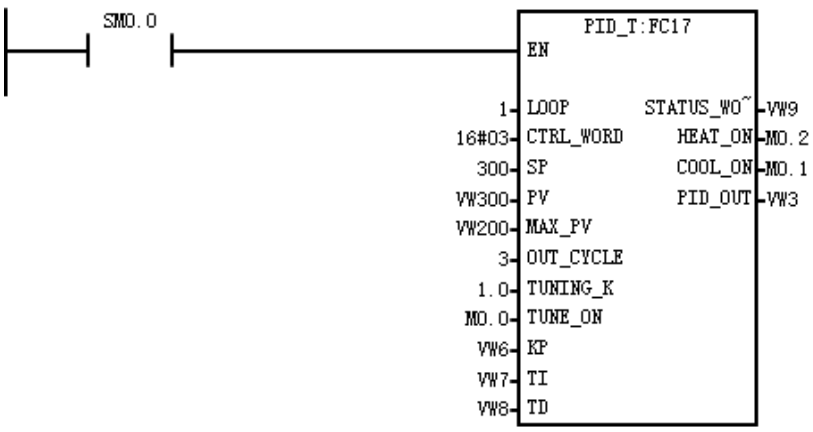
如需该库请前往合信官网免费下载，网址：http://www.co-trust.com

10.1.1 库支持的 PLC 型号

CTH200 系列	H224X（V5 高性能型升级版） H226XM（V5 高性能型升级版） H226XL（V5 高性能型升级版） H226IM（I6 高性能型升级版） H226IL（I6 高性能型升级版） H226IH（I6 高性能型升级版）
CTMC 系列	M228IL M228ML M228SL
CTSC 系列	224+（V5 平台） 224I（V5 平台） 224XP（V5 平台） 226I（V5 平台） 226M（V5 平台） 226L（V5 平台） 226H（V5 平台）
CTH300-H 系列	H36-01 H32-01 H35-00 H31-00 H56-10 H52-10

10.1.2 指令详解

PID_T



参数说明

参数名称	说明	类型	数值范围	备注
LOOP	属于第几个回路PID，不能重复，从0开始。	字，常数或变量	0-63	表示该控制回路ID。
CTRL_WORD	控制字（控制PID运行）	字，常数或变量		常用控制字： 1）16#03(只有加热输出，带自适应功能) 2）16#07(加热冷却输出，带自适应功能)
SP	设定值	字，常数或变量	-32768-32767	单位：0.1℃
PV	测量值（反馈值）	字，变量	-32768-32767	单位：0.1℃
MAX_PV	测量值的最大值	字，常数或变量	-32768-32767	单位：0.1℃
OUT_CYCLE	脉冲输出周期	字，常数或变量	1-255	单位：秒
TUNING_K	自整定系数	双字，浮点数	0.5-2.0	0.5:要求系统控制超调量小。 1.0:正常响应。 2.0:要求系统控制响应快，超调量大。
TUNING_ON	启动自整定	位，变量		自整定结束后自动复位。
Kp	比例系数	字变量		如果字变量赋值为常数，则不能执行自整定功能。
Ti	积分时间	字变量	1-3600	单位：秒 如果字变量赋值为常数，则不能执行自整定功能。
Td	微分时间	字变量	0-3600	单位：秒 如果字变量赋值为常数，则不能执行自整定功能
STATUS_WORD	状态字	字，变量		运行状态及报警状态。
HEAT_ON	加热输出	位		
COOL_ON	冷却输出	位		
PID_OUT	PID模拟输出	字，整数，变量	定义为仅加热输出时，输出范围为：0-32000 带冷却输出时：-32000-32000	

【控制字、状态字位地址】

控制字的位地址意义如下：

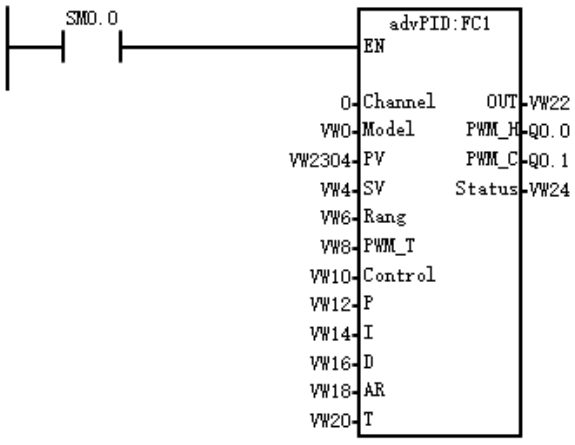
控制字位	设置	备注
0	0	PID 停止
	1	PID 运行
1	0	积分一直起作用，比例系数 Kp 不自动调整
	1	积分分离及比例系数自动调整
2	0	PID 单极输出

	1	PID 双极输出
3	0	保留
	1	保留
4	0	积分起作用
	1	积分不起作用
5	0	微分起作用
	1	微分不起作用
6		保留
7		保留

状态字位地址的意义：

状态字位	值	备注
0	0	无断线故障
	1	断线故障
1	0	自整定未进行
	1	正在自整定
2	0	无自整定故障
	1	自整定故障
3	0	不加热
	1	正在加热
4	0	不冷却
	1	正在冷却
5	0	PID 停止状态
	1	PID 运行状态
6		保留
7		保留

advPID



参数	输入输出属性	数据类型	描述	默认值
Channel	IN	WORD	表示 PID 通道编号，每个通道的计算相互独立，在控制多个对象时，调用多个 PID 块，应使用不同的通道编号来控制。	0

<table>
<tr><td></td><td></td><td></td><td>取值范围：0~7</td><td></td></tr>
<tr><td>Model</td><td>IN</td><td>WORD</td><td>表示 PID 算法模式选择，可根据被控对象选择相应的模式，PID 块会给出相应的 PID 参数。
0：默认模式，适用于绝大多数控制对象。
1：开启 PID 在线自校正的模式，在该模式下，PID 算法会根据调节的效果对 P，I 参数进行实时校正，控制对象和设定值（SV）的条件会不断变化，该控制系统可自动计算 PID 并进行控制。虽然不要求高控制性能，但由于控制对象的条件频繁变化，当自动调整无法运行时该控制系统将非常有效。
注意：在此模式下不要手动调节 PID 参数，否则可能引起无法预知的影响。
10：控制对象为 TEC（半导体制冷器）时专用的算法模式。
取值范围：0~32000</td><td>0</td></tr>
<tr><td>PV</td><td>IN</td><td>WORD</td><td>PV（Process Variable）表示被控对象的值。比如被控对象的温度，压力，流量，等。最后一位为小数，例 4000 表示 400.0。
注意：值范围不能超过 Rang 表示的范围
取值范围：-20000~20000</td><td>0</td></tr>
<tr><td>SV</td><td>IN</td><td>WORD</td><td>表示对被控对象的期望值，设定值。最后一位为小数，例 4000 表示 400.0。
注意：值范围不能超过 Rang 表示的范围
取值范围：-20000~20000</td><td>0</td></tr>
<tr><td>Rang</td><td>IN</td><td>WORD</td><td>表示被控对象的动作范围，比如温度可以取传感器的量程换算，如-200-400℃换算为 6000，电压电流也可以先比例换算为合适的范围，如 0-10V 换算为 10000。
取值范围：1~20000</td><td>4000</td></tr>
<tr><td>PWM_T</td><td>IN</td><td>WORD</td><td>设定 PWM 输出的周期，单位：S。
注意：为了减少外围输出元件寿命损耗，建议对于 SSR（固态继电器）输出取值一般不小于 2，对于机械继电器取值一般不小于 30。
取值范围：0~32000</td><td>0</td></tr>
<tr><td>Control</td><td>IN_OUT</td><td>WORD</td><td>PID 算法的控制设定，以字的 16 个位进行功能选择
注：如果没有选择控制极性，默认为单极反向控制。
DEC(十进制)操作加上表格中相应数字即相应位置 1
<table>
<tr><td>Bit 0
DEC:+1</td><td>1：启动 PID 计算
0：关闭 PID，PID 将停止运行，输出为 0</td></tr>
<tr><td>Bit 1
DEC:+2</td><td>1：启动 PID 自整定算法，整定结束后会自动复位
0：关闭 PID 自整定算法</td></tr>
<tr><td>Bit 2
DEC:+4</td><td>1：输出为单极反向控制，PWM 仅热端输出，即 SV 大于 PV 时，输出增加，用于加热控制，输出范围为 0 - 32000
0：未选择</td></tr>
<tr><td>Bit 3
DEC:+8</td><td>1：输出为单极正向控制，PWM 仅热端输出，即 SV 大于 PV 时，输出减少，用于制冷控制，输出范围为 0 - 32000
0：未选择</td></tr>
<tr><td>Bit 4
DEC:+16</td><td>1：输出为对偶控制，PWM 热端反向，PWM 冷端正向，即 SV 大于 PV 时，输出增加，PWM 端极性相反，输出范围为</td></tr>
</table></td><td>0</td></tr>
</table>

			<table><tr><td></td><td>-32000~32000 0：未选择</td></tr><tr><td>Bit 5 DEC:+32</td><td>1：输出为对偶控制，PWM 热端反向，PWM 冷端反向，即 SV 大于 PV 时，输出增加，PWM 端极性相同，输出范围为 -32000~32000 0：未选择（暂未实现）</td></tr><tr><td>Bit 6 DEC:+64</td><td>1：输出为对偶控制，PWM 热端正向，PWM 冷端正向，即 SV 大于 PV 时，输出增加，PWM 端极性相同，输出范围为 -32000~32000 0：未选择 (暂未实现)</td></tr><tr><td>Bit 7 DEC: +128</td><td>1：输出为对偶控制，PWM 热端正向，PWM 冷端反向，即 SV 大于 PV 时，输出增加，PWM 端极性相反，输出范围为 -32000~32000 0：未选择 (暂未实现)</td></tr><tr><td>Bit 15 DEC: +32768</td><td>1：PID 参数复位 0：无</td></tr><tr><td>Bit 0 DEC:+1</td><td>1：启动 PID 计算 0：关闭 PID，PID 将停止运行，输出为 0</td></tr></table>	
P	IN_OUT	WORD	比例带，设定 PID 参数的比例范围，有效范围为 0~999%，最后一位为小数 ---取值范围：0~9999	0
I	IN_OUT	WORD	积分时间，有效范围为 0 - 3200 秒，最后一位为小数 注：当值为 0 时，取消积分作用 ---取值范围：0~32000	0
D	IN_OUT	WORD	微分时间，有效范围为 0~3200 秒，最后一位为小数 注：当值为 0 时，取消微分作用 ---取值范围：0~32000	0
AR	IN_OUT	WORD	抗积分饱和，设定进行积分工作的范围，有效范围为 0 - 100%。 最后一位为小数 ---取值范围：0~1000	1000
T		WORD	PID 计算周期时间，有效范围为 1~100 秒，最后一位为小数。 取值范围：1~1000	1
OUT	OUT	WORD	PID 的输出 范围：单极性 0~32000，双极性 -32000~32000	0
PWM_H	OUT	BOOL	PWM 热端输出，1 为 ON，0 为 OFF	0
PWM_C	OUT	BOOL	PWM 冷端输出，1 为 ON，0 为 OFF	
Status	OUT	WORD	当前 PID 的运行状态字，以相应位来表示状态 <table><tr><td>状态位</td><td>描述</td></tr><tr><td>Bit 0</td><td>1：PID 正在运行 0：PID 停止运行</td></tr><tr><td>Bit 1</td><td>1：PID 自整定正在运行 0：PID 自整定未运行</td></tr><tr><td>Bit 2</td><td>1：PID 自适应正在运行 0：PID 自适应未运行</td></tr><tr><td>Bit 3</td><td>1：PWM 热端输出 ON 0：PWM 热端输出 OFF</td></tr><tr><td>Bit 4</td><td>1：PWM 冷端输出 ON</td></tr></table>	0

				0：PWM 冷端输出 OFF	
			Bit 14	1：PID 计算故障 0：正常状态	
			Bit 15	1：PID 自整定失败 0：正常状态	

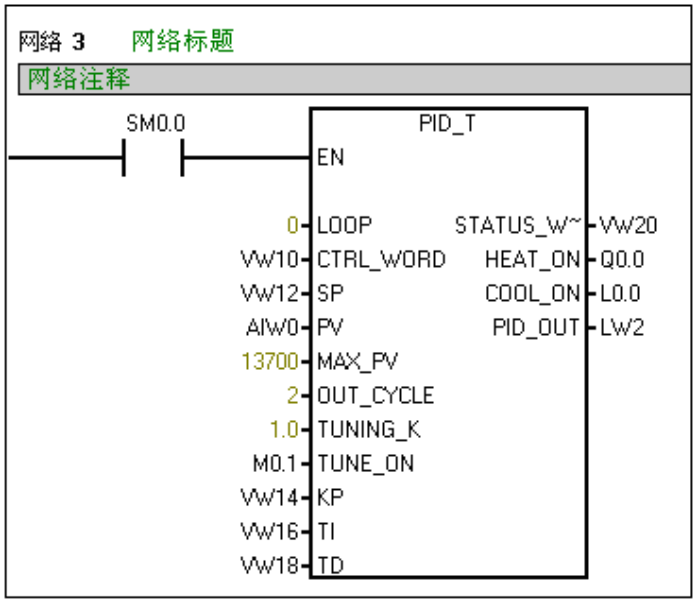
10.1.3 应用举例

系统需求

在这个例子里，我们以一个温区为例，系统及 I/O 配置表如下：

系统配置	CPU H226XL + 231-7PD32	温度采集采用四路热电偶模块	
控制要求	1、只有加热输出，没有冷却输出 2、要求自整定参数 3、K型热电偶		
I/O点分布			
Q0.0	加热输出		
AIW0	温度输入	K型热电偶	
M0.0	PID运行/停止位		
M1.0	自整定启动位		

应用程序



程序说明

PID_T 的参数说明如下：

参数	地址或数值	说明	备注
LOOP	0	此为第一个回路，所以是0	
CTRL_WORD	VW10		
SP	VW12		
PV	AIW0		
MAX_PV	13700	因为K型热电偶的最大输入为13700	
OUT_CYCLE	2	2秒，这是脉冲输出周期	
TUNING_K	1.0		
TUNING_ON	M0.1	置1时开始整定，整定结束后复位	

Kp	VW14	比例系数，整定结束后，整定值自动写到此变量，用户还可以自行调整	
Ti	VW16	积分时间，整定结束后，整定值自动写到此变量，用户还可以自行调整	
Td	VW18	微分时间，整定结束后，整定值自动写到此变量，用户还可以自行调整	
STATUS_WORD	VW20	状态字	
HEAT_ON	Q0.0	加热输出点	
COOL_ON	L0.0	因为没有用到，所以用了一个局部变量	
PID_OUT	LW2	因为没有用到，所以用了一个局部变量	

服务热线

400-700-4858

深圳市合信自动化技术有限公司

深圳市南山区深圳国际创新谷 6 栋 A 座 9 层

0755-86226822

sales@co-trust.com

www.co-trust.com

官方微信公众号