客服热线 400 - 820 - 9595

绵密网络 专业服务

中达电通已建立了 48 个分支机构及服务网点，并塑建训练有素的专业团队，提供客户最满意的服务，公司技术人员能在 2 小时内回应您的问题，并在 48 小时内提供所需服务。

上海
电话 :(021)6301-2827
传真 :(021)6301-2307

南昌
电话 :(0791)8625-5010
传真 :(0791)8625-5102

合肥
电话 :(0551)6281-6777
传真 :(0551)6281-6555

南京
电话 :(025)8334-6585
传真 :(025)8334-6554

杭州
电话 :(0571)8882-0610
传真 :(0571)8882-0603

武汉
电话 :(027)8544-8265
传真 :(027)8544-9500

长沙
电话 :(0731)8827-7881
传真 :(0731)8827-7882

南宁
电话 :(0771)5879-599
传真 :(0771)2621-502

厦门
电话 :(0592)5313-601
传真 :(0592)5313-628

广州
电话 :(020)3879-2175
传真 :(020)3879-2178

济南
电话 :(0531)8690-7277
传真 :(0531)8690-7099

郑州
电话 :(0371)6384-2772
传真 :(0371)6384-2656

北京
电话 :(010)8225-3225
传真 :(010)8225-2308

天津
电话 :(022)2301-5082
传真 :(022)2335-5006

太原
电话 :(0351)4039-475
传真 :(0351)4039-047

乌鲁木齐
电话 :(0991)6118-160
传真 :(0991)6118-289

西安
电话 :(029)8669-0780
传真 :(029)86690780-8000

成都
电话 :(028)8434-2075
传真 :(028)8434-2073

重庆
电话 :(023)8806-0306
传真 :(023)8806-0776

哈尔滨
电话 :(0451)5366-0643
传真 :(0451)5366-0248

沈阳
电话 :(024)2334-1612
传真 :(024)2334-1163

长春
电话 :(0431)8892-5060
传真 :(0431)8892-5065

DVP-15MC 系列运动控制器操作手册

DVP-15MC系列
运动控制器操作手册

www.deltaww.com

Smarter. Greener. Together.

台达 DELTA
中达电通股份有限公司

地址：上海市浦东新区民夏路238号
邮编：201209
电话：(021)5863-5678
传真：(021)5863-0003
网址：http://www.deltagreentech.com.cn

DVP-0199410-03
2018/07/18

DVP-15MC 系列运动控制器操作手册

目录

第 11 章 运动指令

注意事项

- ✓ 此操作手册提供功能规格、安装、基本操作和设定介绍。
- ✓ 本机为开放型（OPEN TYPE）机种，因此用户使用本机时，必须将其安装在具防尘、防潮和免在电击/冲击意外的外壳配线箱内。另必须具备保护措施（如：特殊的工具或钥匙才可打开）防止非维护人员操作或意外冲击本体，造成危险和损坏。
- ✓ 请务必仔细阅读本使用手册，并依照本手册指示进行操作，以免造成产品受损，或导致人员受伤。

1

第1章 前言

目录

欢迎您使用 DVP-15MC 系列运动控制器，我们为您提供高端运动控制系统。

本手册介绍以运动控制为核心所构建的 DVP-15MC 系列运动控制器产品，包含产品功能规格、系统架构、安装、接线、控制器执行原理、逻辑指令和运动指令、故障排除、通讯协议及原点回归模式说明等。请确认您对于 DVP-15MC 系列运动控制系统之配置以及操作有充分的了解，以便正确地使用 DVP-15MC 系列运动控制器。

1.1 手册中的图标说明

- **使用前注意**

在操作本产品前，请先详细阅读并注意相关安全信息，确保自身安全及产品安全。

⚠ 危险	该标志表示危险性高，如果不按照说明进行操作，可能会导致死亡、严重的人身伤害或者设备损坏。
⚠ 警告	该标志表示存在危险性，如果不按照说明进行操作，可能会导致死亡、中度的人身伤害或者设备损坏。
⚠ 注意	该标志表示需要注意，如果不按照说明进行操作，可能会出现非预期的结果。

1.2 手册改版记录

DVP-15MC 系列运动控制器操作手册版本修订一览表

版本	变更内容	发行日期
第一版	新版发行	2017/06/21
第二版	新增内容如下： 1. 新增第 8.14.2 节以太网指令 2. 新增第 8.14.3 节 RS485 通讯指令 3. 新增第 8.14.4 节 RS232 通讯指令 4. 新增第 11.6 节 G 代码指令 5. 新增第 11.7 节轴组指令	2018/03/20
第三版	1. 增加 DVP15MC11T-06 机种及其说明 2. 修改第 6.7.1 节 Ethernet 通讯口支持的功能说明 3. 修改/增加第 12.2 节运动指令错误代码	2018/07/18

2

第2章 DVP-15MC 系列运动控制器概述

目录

2.1 产品概述

DVP-15MC 系列运动控制器是台达自主研发的基于 CANopen 现场总线的多轴运动控制器，它遵循 CANopen DS301 基本通讯协议和 DSP402 运动控制协议，同时还支持国际组织所定义的运动控制标准指令库，方便用户快速入门，迅速的进行项目开发。它通过 Motion 接口可以控制多轴，支持位置、速度、扭矩、原点回归等单轴运动指令，支持电子齿轮，电子凸轮、旋切、G 代码等多轴指令。

DVP-15MC 系列运动控制器内建多种通讯口，用户不需要增加模块就可以实现各种通讯功能，同时还支持左侧和右侧两个扩展接口（左侧为高速并行扩展口），左右侧可以扩展 DVP-S 系列左右侧模块。

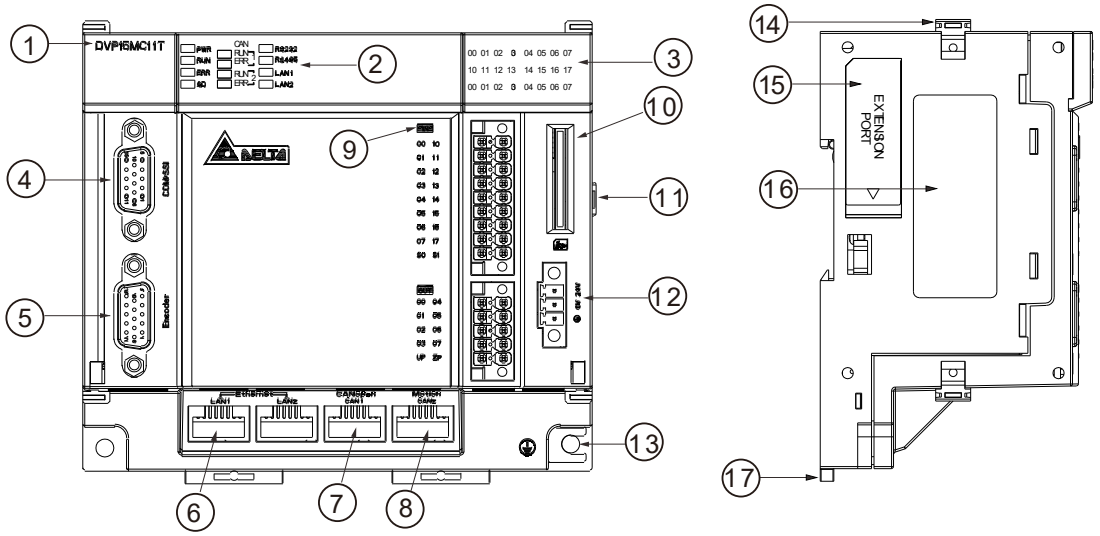
由于采用高可靠性的 CAN 总线为主线的通讯系统，DVP-15MC 系列运动控制器只需为客户提供简单配线，因具有高速可靠的运动控制系统，可广泛应用在包装、印刷、封装、线切割、制药等各种自动化控制领域中。

DVP-15MC 系列运动控制器包含 DVP15MC11T 和 DVP15MC11T-06 这两个机种。这两个机种仅控制的轴数不同，外围通讯口、左右侧扩展及程序容量等规格相同。

2.2 功能简介

- DVP15MC11T 可控制 24 个实轴（轴号范围：1~32）。
- DVP15MC11T-06 可控制 6 个实轴（轴号范围：1~16）。
- DVP15MC11T 内部可构建虚轴及编码器轴（虚轴及编码器轴号范围：1~32，不可与实轴轴号重复）。
- DVP15MC11T-06 内部可构建虚轴及编码器轴(虚轴及编码器轴号范围 :1~16 ,不可与实轴轴号重复)。
- 配备 1GHz 高速浮点运算处理器，支持 64 位浮点数（LREAL），可胜任各种复杂运动控制任务。
- 内建两个增量型编码器接口和一个 SSI 绝对性编码器接口
- 内建 1 个 RS-232、1 个 RS-485、2 个以太网接口。
- 内建 CAN 接口，可以做 CANopen 主站或者从站。
- 强大的现场网络支持（以太网主从站、CANopen 主从站及 Profibus-DP 从站），可组建功能复杂的控制系统。
- 种类众多的 IO 扩展（左侧高速 AIAO、右侧低速 AIAO 及 DIDO，温度模块等）。
- 使用简单、功能完整，方便应用的软件界面。
- 提供标准的总线电缆，终端电阻，分接盒等附件产品，配线简单方便，即插即用。

2.3 外观及各部件介绍



①	机种名称	⑩	SD 卡插槽
②	状态指示灯	⑪	右侧扩展模块接口
③	IO 指示灯	⑫	24V 电源接口
④	COM/SSI 通讯口	⑬	螺钉固定扣
⑤	Encoder（增量型编码器）口	⑭	扩展模块固定扣
⑥	Ethernet 通讯口	⑮	左侧扩展模块接口
⑦	CANopen 通讯口	⑯	铭牌
⑧	Motion 通讯口	⑰	DIN 轨固定扣
⑨	输入输出脚位及标示		

MEMO

3

第3章 规格

目录

3.1 功能规格

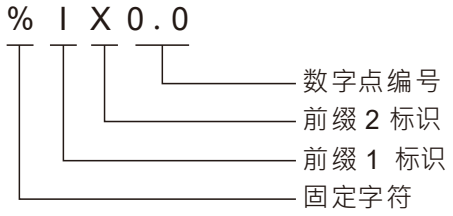
3.1.1 规格

项目				规格			
编程	程序容量	大小		20M			
		数量	POU 定义数	1024			
	变量容量	掉电保持	容量	128K			
		非掉电保持	容量	20M			
	G 代码	单个 G 代码程序	容量	256K			
		G 代码程序	数量	64			
运动控制	控制轴数	单轴控制最大数量		机种名称	实轴数量	虚轴数量	实轴+虚轴数量
				DVP15MC11T	24	32	32
				DVP15MC11T-06	6	16	16
		直线插补控制最多轴数		8			
		圆弧插补控制最多轴数		3			
	凸轮条数	大小	数量	64			
	凸轮关键点	单个凸轮关键点	数量	2048			
本机外部端口		CAN	2	一个支持标准 CANopen 协议，一个做 Motion 用			
		以太网	2	两个独立的以太网口			
		RS-232	1	可以做主站或从站			
		RS-485	1	可以做主站或从站			
		增量型编码器	2	可以构建编码器，Z 信号可以触发中断程序			
		绝对型 SSI 编码器	1	可以构建编码器轴			
		本机输入点	数量	16 点（支持外部中断触发）			
		本机输出点	数量	8 点			
		左侧扩展	1	Slim 系列左侧模块			
		右侧扩展	1	Slim 系列特殊模块			
左右侧扩展	左侧扩展	左侧扩展模块	数量	8 台 Slim 系列左侧模块			
	右侧扩展	特殊模块	台数	8 台 Slim 系列特殊模块			
		数字量	点数	240 点输入和 240 点输出			

3.1.2 装置与数据类型说明

3.1.2.1 装置说明

- **装置名称说明**



- **DVP-15MC 系列运动控制器软件中使用到的相关装置列表如下：**

序号	项目	内容				
1	前缀 1 标识	I	Q	M		
2	前缀 1 名称	输入装置	输出装置	中间装置		
3	前缀 2 标识	X	B	W	D	L
4	前缀 2 数据类型	位装置	字节装置	字装置	双字装置	四字装置
5	装置范例	%IX0.0	%IB0	%IW0	%ID0	%IL0
6		%QX0.0	%QB0	%QW0	%QD0	%QL0
7		%MX0.0	%MB0	%MW0	%MD0	%ML0

- **装置对应关系如下表所示：**

1. 如下表所示，%ML0 包括%MB0~%MB7，%MD0 包括%MB0~%MB3，%MW0 包括%MB0~%MB1。

装置名称	装置对应关系																							
	第 1 个 WORD						第 2 个 WORD						第 3 个 WORD						第 4 个 WORD					
	Bit 0	…	Bit 7	Bit 8	…	Bit 15	Bit 0	…	Bit 7	Bit 8	…	Bit 15	Bit 0	…	Bit 7	Bit 8	…	Bit 15	Bit 0	…	Bit 7	Bit 8	…	Bit 15
%MX	%MX0.0~0.7			%MX1.0~1.7			%MX2.0~2.7			%MX3.0~3.7			%MX4.0~4.7			%MX5.0~5.7			%MX6.0~6.7			%MX7.0~7.7		
%MB	%MB0			%MB1			%MB2			%MB3			%MB4			%MB5			%MB6			%MB7		
%MW	%MW0						%MW1						%MW2						%MW3					
%MD	%MD0												%MD1											
%ML	%ML0																							

2. 如下表所示，%ML1 包括%MB8~%MB15，%MD2 包括%MB8~%MB11，%MW4 包括%MB8~%MB9。%MB8 包括%MX8.0~8.7。

装置名称	装置对应关系																							
	第 5 个 WORD						第 6 个 WORD						第 7 个 WORD						第 8 个 WORD					
	Bit 0	…	Bit 7	Bit 8	…	Bit 15	Bit 0	…	Bit 7	Bit 8	…	Bit 15	Bit 0	…	Bit 7	Bit 8	…	Bit 15	Bit 0	…	Bit 7	Bit 8	…	Bit 15
%MX	%MX8.0~8.7			%MX9.0~9.7			%MX10.0~10.7			%MX11.0~11.7			%MX12.0~12.7			%MX13.0~13.7			%MX14.0~14.7			%MX15.0~15.7		
%MB	%MB8			%MB9			%MB10			%MB11			%MB12			%MB13			%MB14			%MB15		
%MW	%MW4						%MW5						%MW6						%MW7					
%MD	%MD2												%MD3											
%ML	%ML1																							

3.1.2.2 装置有效范围

- **DVP-15MC 系列运动控制器装置有效范围列表如下：**

装置名称	装置表示方法	装置范围
%IX	%IX0.0~%IX0.7	%IX0.0~%IX127.7
%QX	%QX0.0~%QX0.7	%QX0.0~%QX127.7
%MX	%MX0.0	%MX0.0~%MX131071.7
%IB	%IB0	%IB0~%IB127
%QB	%QB0	%QB0~%QB127
%MB	%MB0	%MB0~%MB131071
%IW	%IW0	%IW0~%IW63
%QW	%QW0	%QW0~%QW63
%MW	%MW0	%MW0~%MW65535
%ID	%ID0	%ID0~%ID31
%QD	%QD0	%QD0~%QD31
%MD	%MD0	%MD0~%MD32767
%IL	%IL0	%IL0~%IL15
%QL	%QL0	%QL0~%QL15
%ML	%ML0	%ML0~%ML16383

- **Modbus 装置地址列表**

装置区域	装置类型	装置范围	Modbus 地址	Modbus 地址类型
I（输入）	位装置（bit）	%IX0.0~%IX0.7	0x6000~0x6007	标准 Modbus 地址
		%IX1.0~%IX1.7	0x6008~0x600F	
		……	……	
		%IX127.0~%IX127.7	0x63F8~0x63FF	
	字装置（word）	%IW0~%IW63	0x8000~0x803F	
Q（输出）	位装置（bit）	%QX0.0~%QX0.7	0xA000~0xA007	
		%QX1.0~%QX1.7	0xA008~0xA00F	
		……	……	
		%QX127.0~%QX127.7	0xA3F8~0xA3FF	
	字装置（word）	%QW0~%QW63	0xA000~0xA03F	
M（寄存器）	位装置（bit）	%MX0.0~%MX0.7	0x10000000~0x10000007	台达扩展 Modbus 地址
		%MX1.0~%MX1.7	0x10000008~0x1000000F	
		……	……	
		%MX131071.0~%MX131071.7	0x100FFFF8~0x100FFFFF	
	字装置（word）	%MW0~%MW32767	0x0000~0x7FFF	标准 Modbus 地址
	字装置（word）	%MW32768~%MW65535	0x20008000~0x2000FFFF	台达扩展 Modbus 地址

3.1.2.3 掉电保持装置

%MW0~%MW999 为掉电保持装置。另外，软件中定义的变量可以选择是否为掉电保持。掉电保持的容量为 128K 字节。

3.1.2.4 支持的数据类型及有效范围

DVP-15MC 系列运动控制器使用的软件中变量支持的数据类型及有效范围明细如下表所示：

序号	数据类型	有效范围	初始值
1	BOOL	TRUE or FALSE	FALSE
2	BYTE	16#00 ~ FF	16#00
3	WORD	16#0000 ~ FFFF	16#0000
4	DWORD	16#00000000 ~ FFFFFFFF	16#00000000
5	LWORD	16#0000000000000000 ~ FFFFFFFFFFFFFFFF	16#0000000000000000
6	USINT	0 ~ +255	0
7	UINT	0 ~ +65535	0
8	UDINT	0 ~ +4294967295	0
9	ULINT	0 ~ +18446744073709551615	0
10	SINT	-128 ~ +127	0
11	INT	-32768 ~ +32767	0
12	DINT	-2147483648 ~ +2147483647	0
13	LINT	-9223372036854775808 ~ +9223372036854775807	0
14	REAL	-3.402823e+38 ~ -1.175495e-38， 0， +1.175495e-38 ~ +3.402823e+38	0.0
15	LREAL	-1.79769313486231e+308 ~ -2.22507385850721e-308, 0， +2.22507385850721e−308 ~ +1.79769313486231e+308,	0.0
16	TIME	T#XXXXXXdXXhXXmXXsXX.XXXms，以 ns 为单位。 显示范围：T#0ns~213503d23h34m33s709.551ms	T#0ms
17	DATE	D#年-月-日。显示范围：D#1970-01-01~D#2106-02-07。以秒为单位。	D#1970-01-01
18	TOD	TOD#时：分：秒.毫秒。显示范围： TOD#00:00:00~23:59:59.999。以 ms 为单位。数值为 0 时显示 TOD#00:00:00，数值为 1 时显示 TOD#00:00:00.001，数值为 86399999 时显示 TOD#23:59:59.999，数值为 86400000 时显示 TOD#00:00:00，数值为 4294967295 时显示 TOD#17:2:47.295。	TOD#00:00:00
19	DT	DT#年-月-日-时-分-秒。显示范围： DT#1970-01-01-0:0:0~2106-02-07-6:28:15。以 s 为单位。	DT#1970-01-01-0:0:0

序号	数据类型	有效范围	初始值
20	STRING	0~32000 个字符	''

3.2 电气规格

- **电气规格**

项　目	内　　容
电源电压	24 VDC（-15% ~ +20%）
电源保险丝容量	3 A/30 VDC，可恢复式 （Polyswitch）
隔离电压	500 VDC（Secondary-PE）
消耗电力	8W Max
耐振动/冲击	标准:IEC61131-2,IEC 68-2-6 （TEST Fc）/IEC61131-2 & IEC 68-2-27 （TEST Ea）
抗干扰度	静电： 8KV Air Discharge ,4KV Contact Discharge EFT： Power Line：±2KV， Digital Input: ±1KV, Communication I/O: ±1KV RS：80MHz ~ 1000MHz, 10V/m. Conducted Susceptibility Test: 150kHz ~ 80MHz, 3V/m Surge Test : Power line 0.5KV DM/CM
环境要求	工作： 0°C ~ 55°C （温度），5 ~ 95% （湿度），污染等级 2 储存： -25°C ~ 70°C （温度），5 ~ 95% （湿度）
重量	约 425g

- **输入点电气规格**

项　目	内　　容
输入通道数	16 通道
通道类型	16 通道为高速数字输入型
输入接线端子	接线端子 I0~I7，I10~I17
输入点公共端	接线端子 S0/S1
输入类型	漏型模式（Sink）或者源型模式（Source）
输入延迟时间	2.5μS （ OFF ->ON ），5 μS （ON -> OFF ）
输入电流	24 VDC，5mA
电缆最大长度	有屏蔽：500m；
	无屏蔽：300m

- **输出点电气规格**

项　目	内　　容
输出通道数	8 晶体管输出（N-MOS）
通道分类	8 通道为高速数字输出型

项　目	内　容
输出接线端子	接线端子 Q0~Q7
输出点公共端	接线端子 UP/ZP （用于连接供电电源的正极或负极）
输出点供电电压	24 VDC（-15% ~ +20%）#1
输出延迟时间	2μS （ OFF -> ON）· 3μS （ ON -> OFF ）
最大切换频率	1KHz
最大负载	负载为电阻型：0.5A/1 点（2A/ZP）
	负载为电感型：13W（24VDC）
	负载为灯泡型：2.5W（24VDC）
最大电缆长度	有屏蔽：500m
	无屏蔽：300m

#1: UP、ZP 必须外加辅助电源 24VDC（-15%~20%）.

MEMO

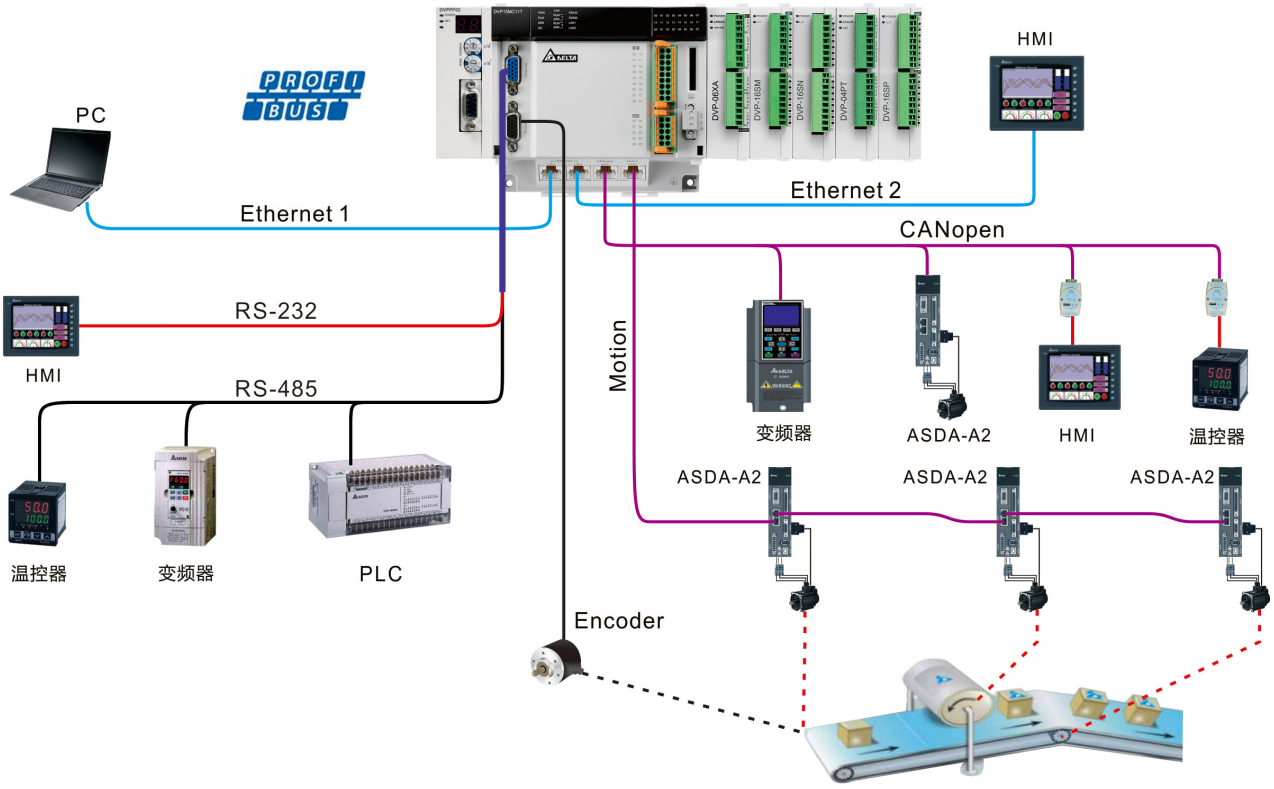
4

第4章 系统架构

目录

4.1 系统构成

使用 DVP-15MC 系列运动控制器可以组建多层工业网络。 如下图所示，使用 DVP-15MC 系列运动控制器，可以构成上层为以太网，中层为 CANopen 总线和 PROFIBUS 总线，下层为 RS-485 总线(支持 Modbus) 的网络。



此系统图说明了 DVP-15MC 系列运动控制器在整个系统中各个端口所连接的外部设备，详细介绍各个通讯口的作用请参考第 6 章节。

4.2 电源

建议用户使用台达电源模块为 DVP-15MC 系列运动控制器供电，电源模块型号如下所示：

序号	模块名称	相数	输入电压	输出电压	功率	输出电流	国标
1	DVPPS02	单相	85~264VAC	24VDC	48W	2A	
2	DVPPS05				120W	5A	

4.3 左侧扩展

4.3.1 可以连接的左侧扩展模块

DVP-15MC 系列运动控制器左侧最多连接 8 台高速扩展模块，可连接的模块如下列表所示。

序号	模块名称	模块类型	模块说明
1	DVP04AD-SL	模拟量模块	模拟量输入
2	DVP04DA-SL	模拟量模块	模拟量输出
3	DVPPF02-SL	网络模块	Profibus 通讯

4.3.2 左侧网络模块地址分配

- **DVP-15MC 系列运动控制器左侧网络模块输入/输出映射区说明**

DVP-15MC 系列运动控制器左侧网络模块做为从站时，在 PLC 主机左侧不同位置的输入/输出映射区如下表所示，在 PLC 主机左侧第一台的位置为 1，第二台的位置为 2，其它位置以此类推。

映射区 / 位置	输出映射区	输入映射区
1	%MW6250~%MW6377	%MW6000~%MW6127
2	%MW6750~%MW6877	%MW6500~%MW6627
3	%MW7250~%MW7377	%MW7000~%MW7127
4	%MW7750~%MW7877	%MW7500~%MW7627
5	%MW8250~%MW8377	%MW8000~%MW8127
6	%MW8750~%MW8877	%MW8500~%MW8627
7	%MW9250~%MW9377	%MW9000~%MW9127
8	%MW9750~%MW9877	%MW9500~%MW9627

左侧扩展模块的映射区域详细划分请参考模块操作手册。需注意模块操作手册中提及的映射地址表示方法的变更。例如 DVPPF02-SL 输出映射区为 D6250~D6349，当该模块连接至 DVP-15MC 系列运动控制器左侧使用时，该区域地址表示方法为%MW6250~%MW6349。

4.3.3 左侧模块操作方法

控制器可以使用 FROM/TO 指令读写左侧模块的寄存器（CR）。如 DVP04AD-SL、DVP04DA-SL 等，可以使用 FROM/TO 指令读写操作。

4.4 右侧扩展

4.4.1 可以连接的右侧扩展模块

DVP-15MC 系列运动控制器右侧可连接 Slim 系列的扩展模块 (数字量模块、模拟量模块、温度模块等)，数字量模块输入点与输出点最多可各接 240 点，模拟量模块最多可接 8 台，可连接的右侧扩展模块如下列表所示。

序号	模块名称	输入长度	输出长度	扩展类型
1	DVP08SM11N	8 bit	-	输入点数扩展
2	DVP16SM11N	16 bit	-	
3	DVP06SN11R	-	6 bit	输出点数扩展
4	DVP08SN11R/T	-	8 bit	
5	DVP16SN11T	-	16 bit	
6	DVP08SP11R/T	4 bit	4 bit	混合输入/输出扩展
7	DVP16SP11R/T	8 bit	8 bit	
8	DVP16SP11TS (PNP)	8 bit	8 bit	
9	DVP32SM11N	32 bit	-	排针式输入
10	DVP32SN11TN	-	32 bit	排针式输出
11	DVP08ST11N	8 bit	-	数位开关
12	DVP04AD-S	4 word	-	模拟量输入
13	DVP06AD-S	6 word	-	
14	DVP04DA-S	-	4 word	模拟量输出
15	DVP02DA-S	-	2 word	
16	DVP06XA-S	4 word	2 word	混合模拟输入/输出
17	DVP04PT-S	4 word	-	传感器 (型号：PT100)
18	DVP06PT-S	6 word	-	
19	DVP04TC-S	4 word	-	传感器 (型号：J、K、R、S、T 热电耦)

4.4.2 右侧扩展模块地址分配

DVP-15MC 系列运动控制器右侧可连接 Slim 系列的扩展模块，连接数字量最多 240 点输入和 240 点输出，同时还可以连接最多 8 台特殊模块，如模拟量模块、温度模块、脉冲模块等。右侧可连接的数字量模块和特殊模块的总和为 14 台。

- **右侧扩展数字量模块编号说明**

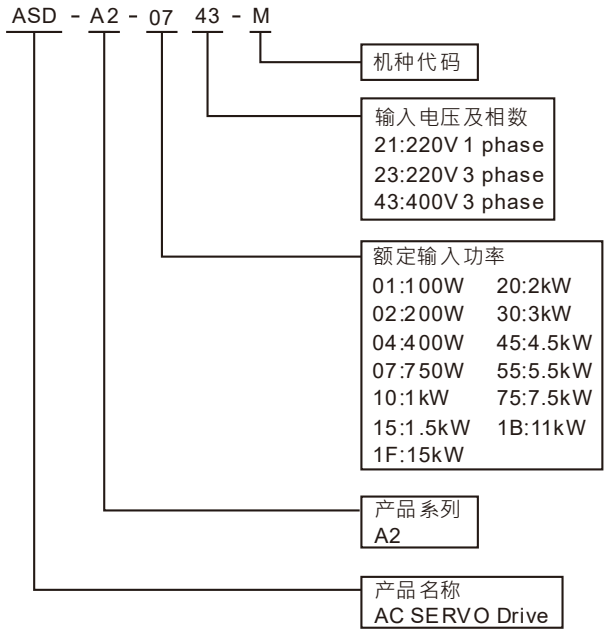
1. 右侧扩展数字量模块编号从 2.0 开始，如第一台数字量扩展模块的输入点从 %IX2.0 开始，输出点从 %QX2.0 开始，未满 8 点以 8 点计算。
2. 数字量输入/输出接点的编号 : (为 8 进制编号)
3. %IX2.0 ~%IX2.7,....., %IX16.0 ~%IX16.7,....., %IX31.0 ~ %IX31.7
4. %QX2.0 ~ %QX2.7,....., %QX16.0 ~ %QX16.7,....., %QX31.0 ~ %QX31.7

- **右侧特殊模块编号说明**
  - 右侧扩展模块如模拟量模块、温度模块、脉冲模块等称之为特殊模块。
  - DVP-15MC 系列运动控制器右侧第一台特殊模块编号为 0，第二台编号为 1，其它以此类推，最多可挂接 8 台特殊模块。右侧特殊模块输入起始地址为%MW10000，右侧特殊模块输出起始地址为%MW10500。
  - DVP-15MC 系列运动控制器可以通过软件的硬件配置界面直接读写右侧模块参数，也可以通过在程序中直接对地址赋值或对已绑定地址的变量赋值的方式读写右侧模块参数。

4.5 Motion 通讯口可以连接的伺服

ASDA-A2 系列伺服驱动器有多种机种，ASDA-A2-XXXX-M 和 ASDA-A2-XXXX-MN 机种支持 CANopen 通讯，仅此型号的伺服驱动器可与 DVP-15MC 系列运动控制器 Motion 通讯口连接组成 CANopen 运动控制网络。DVP-15MC 系列运动控制器与伺服驱动器通过 UC-CMC003-01A (或者 UC-CMC005-01A) 电缆连接，电缆通过 CN6 端口接入伺服驱动器。

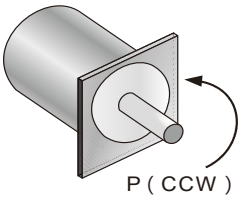
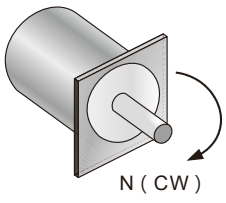
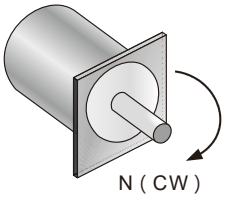
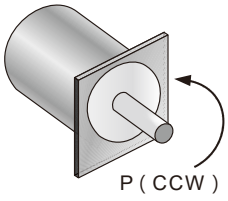
- **伺服驱动器型号说明**



- **DVP15MC11T 控制器与伺服驱动器连接时，伺服驱动器的相关参数设置如下所示：**

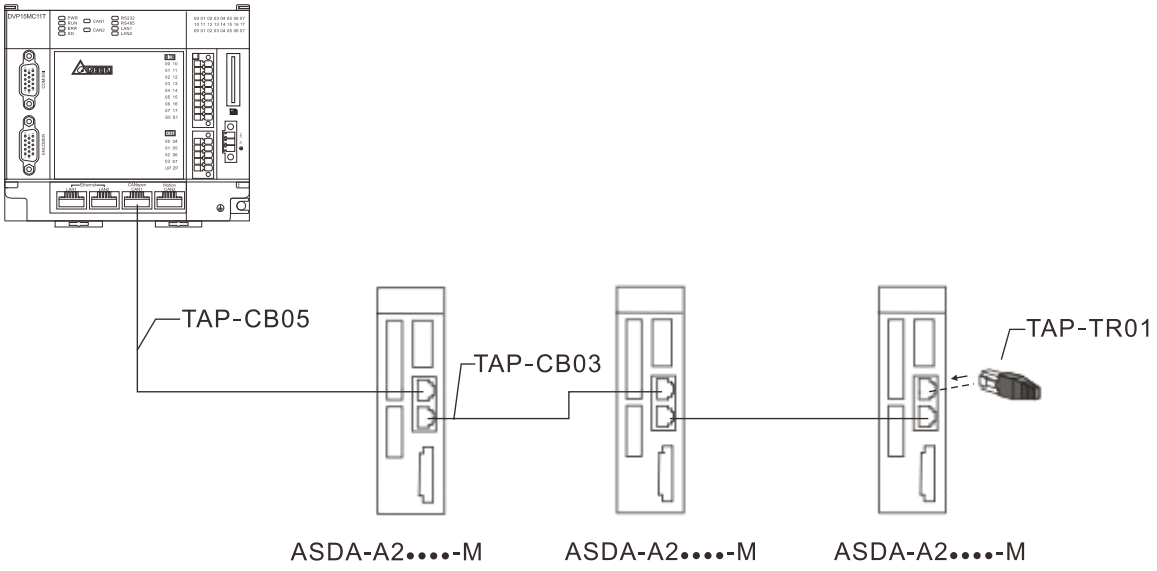
参数名称	参数说明	设置值	设置值说明
P1-01	伺服控制模式设置	X0B*[1]	伺服驱动器设为 CANopen 模式
P3-00	站号设置	设置范围：1~32	此参数的设定值是对应伺服在 CANopen 网络的站号
P3-01	通讯传输速率	0403	此参数的设定值所对应的波特率必须与 DVP15MC11T 的波特率一致。 0403:CANopen 的波特率为 1Mbps 0203: CANopen 的波特率为 500Kbps

*[1]：X 值为 0、1 时，扭矩输出方向控制如下图所示：

	0	1
正方向		
反方向		

- **DVP15MC11T 与 ASDA-A2-XXXX-M 系列伺服驱动器的配线图：**

DVP15MC11T



注：

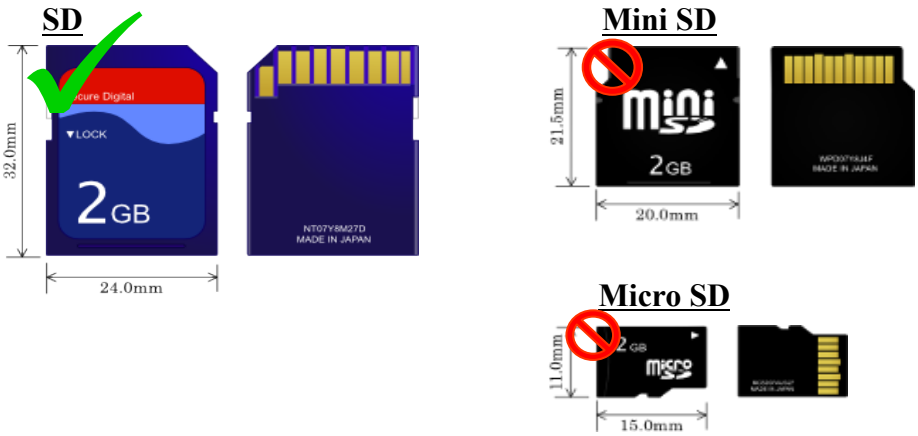
1. ASDA-A2-XXXX-M 系列伺服驱动器与伺服电机及编码器的接线方法请参考伺服手册。
2. 可根据现场状况选用 UC-CMC003-01A、UC-CMC005-01A 或 UC-CMC010-01A 通讯电缆。
3. DVP15MC11T Motion 通讯口内嵌 120 欧姆的终端电阻。DVP15MC11T 的 Motion 通讯口和伺服组成 CANopen 网络时，网络的另一个终端也须接终端电阻 TAP-TR01（可在 DVP15MC11T 控制器包装盒内找到）。

4.6 SD 储存卡

4.6.1 型号与规格

- **型号及外观**

SD 卡依尺寸大小共分为 SD、Mini SD 及 Micro SD 三种，控制器仅支持第一种的标准尺寸。



- **规格**

目前市面上的 SD 卡规格相当繁多，除上述的尺寸区别之外，依支持容量的大小还可以分成 SD、SDHC 及 SDXC 三种类别，而控制器目前则只支持基本的 SD 规格。下列是所有 SD 卡家族的一览表，控制器仅支持 SD 和 SDHC 类型的 SD 卡，选购时请务必谨慎挑选符合规格的商品。

- **SD 卡种类**

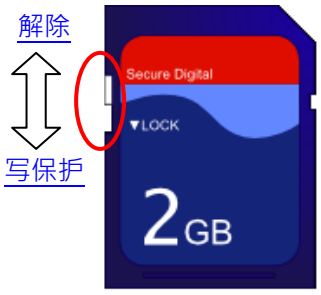
种类	SD	SDHC			SDXC	
容量	32MB~2GB	4GB~32GB			32GB~2TB	
文件系统	FAT16/FAT32	FAT32			exFAT（FAT64）	
尺寸	SD	SDHC	Mini SDHC	Micro SDHC	SDXC	Micro SDXC
SD 速度等级	N/A	CLASS 2 （Min. 2MB/Sec.） CLASS 4 （Min. 4MB/Sec.） CLASS 6 （Min. 6MB/Sec.） CLASS 10 （Min. 10MB/Sec.）			CLASS 2 （Min. 2MB/Sec.） CLASS 4 （Min. 4MB/Sec.） CLASS 6 （Min. 6MB/Sec.） CLASS 10 （Min. 10MB/Sec.）	

* 另外尚有一种 MMC 储存卡在外观上与 SD 卡十分相似，选购时请务必仔细确认。

- **使用储存卡之前**

  - **储存卡的写保护功能**

一般的 SD 卡都会有一个写保护开关，当开关往下拨的时候便代表无法将数据写入 SD 卡中，因此若用户要在控制器上使用 SD 卡且需执行写入功能时，请务必确认 SD 卡的写保护开关已正确解除。


解除
写保护
Secure Digital
LOCK
2GB

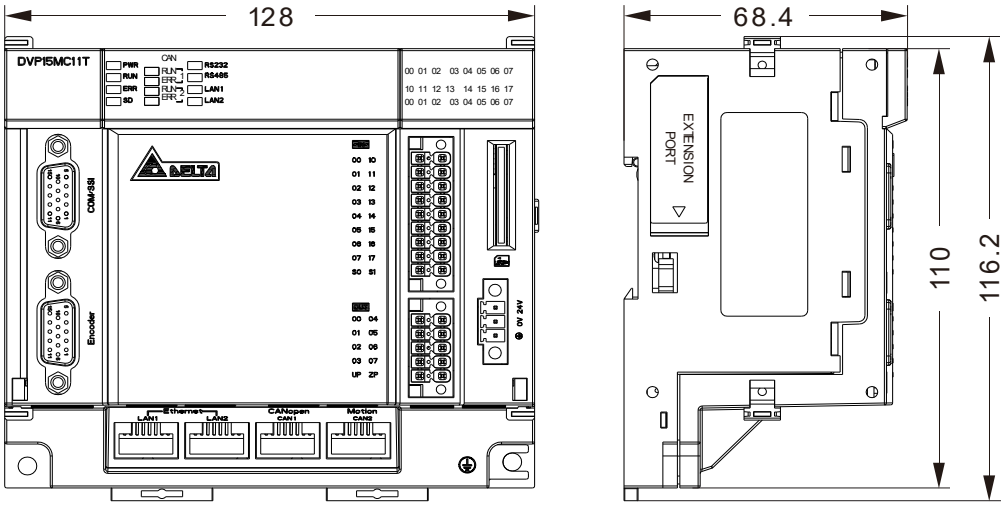
5

第5章 安装

目录

5.1 尺寸

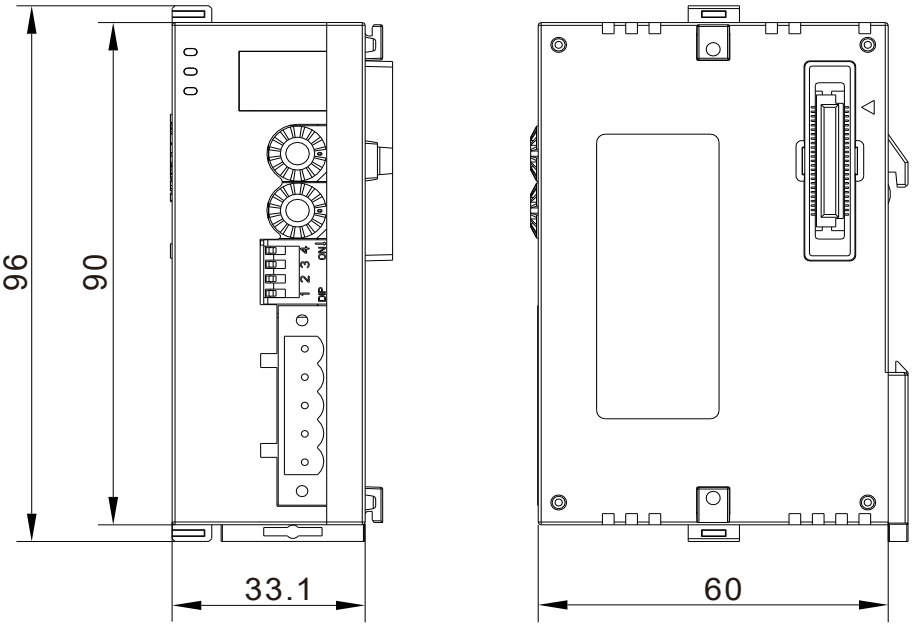
5.1.1 DVP-15MC 系列运动控制器外观尺寸



单位：mm

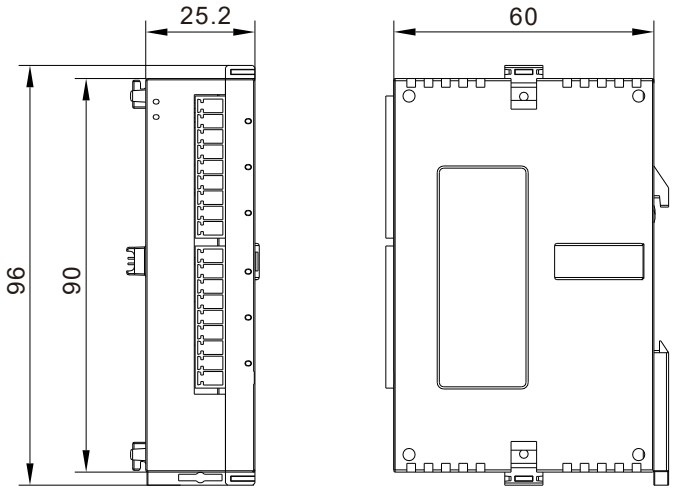
5.1.2 左右侧扩展模块尺寸

- 左侧扩展模块尺寸图，以 DVPCOPM-SL 为例，如下图所示，且所有左侧模块长、宽、高尺寸都与 DVPCOPM-SL 相同。



单位：mm

- 右侧扩展模块尺寸图，以 DVP04AD-S 为例，如下图所示，且所有右侧模块长、宽、高尺寸都与 DVP04AD-S 相同。



单位：mm

5.1.3 与左侧扩展模块的连接

- **安装 DVP-15MC 系列运动控制器与 DVPDNET-SL 模块**
  - 将 DVP-15MC 系列运动控制器左侧上下两端的扩展模块卡口打开，将 DVPDNET-SL 模块沿四角上的导入孔装入，如图 5.1.3.1 步骤 1 所示。
  - 压入 DVP-15MC 系列运动控制器上下两端的卡口，卡紧模块以保证接触良好，如图 5.1.3.1 步骤 2 所示。

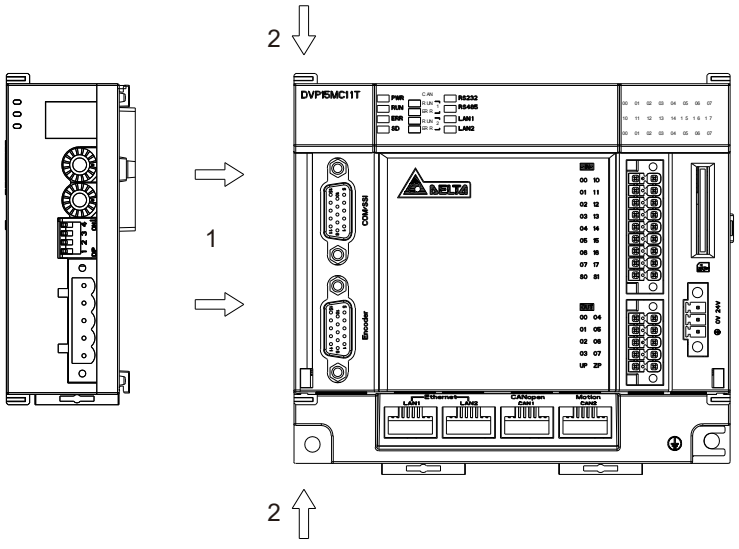


图 5.1.3.1

- **安装 DVP-15MC 系列运动控制器及 DVPDNET-SL 模块于导轨**
  - 请使用 35mm 的标准 DIN 导轨。

- 打开 DVP-15MC 系列运动控制器及 DVPDNET-SL 模块的 DIN 轨固定扣，将 DVP-15MC 系列运动控制器及 DVPDNET-SL 模块嵌入 DIN 导轨上。
- 压入 DVP-15MC 系列运动控制器及 DVPDNET-SL 模块的 DIN 轨固定扣，将 DVP-15MC 系列运动控制器及 DVPDNET-SL 模块固定在 DIN 导轨上，如图 5.1.3.2 所示。

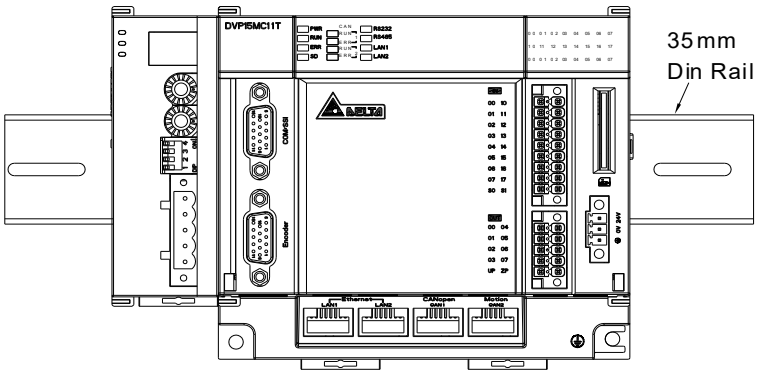


图 5.1.3.2

5.1.4 与右侧扩展模块的连接

- **安装 DVP-15MC 系列运动控制器与 DVP16SP11T 模块**
  - 将 DVP-15MC 系列运动控制器右侧上下两端的扩展模块卡口打开，将 DVP16SP11T 模块沿四角上的导入孔装入，如图 5.1.4.1 步骤 1 所示。
  - 压入 DVP-15MC 系列运动控制器上下两端的卡口，卡紧模块以保证接触良好，如图 5.1.4.1 步骤 2 所示。

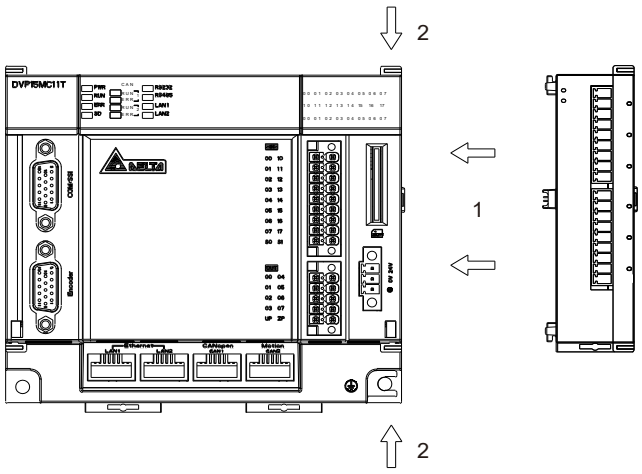


图 5.1.4.1

- **安装 DVP-15MC 系列运动控制器及 DVP16SP11T 模块于导轨**
  - 请使用 35mm 的标准 DIN 导轨。
  - 打开 DVP-15MC 系列运动控制器及 DVP16SP11T 模块的 DIN 轨固定扣，将 DVP-15MC 系列运动控制器主机及 DVP16SP11T 模块嵌入 DIN 导轨上。

■ 压入 DVP-15MC 系列运动控制器及 DVP16SP11T 模块的 DIN 轨固定扣，将 DVP-15MC 系列运动控制器及 DVP16SP11T 模块固定在 DIN 导轨上，如图 5.1.4.2 所示。

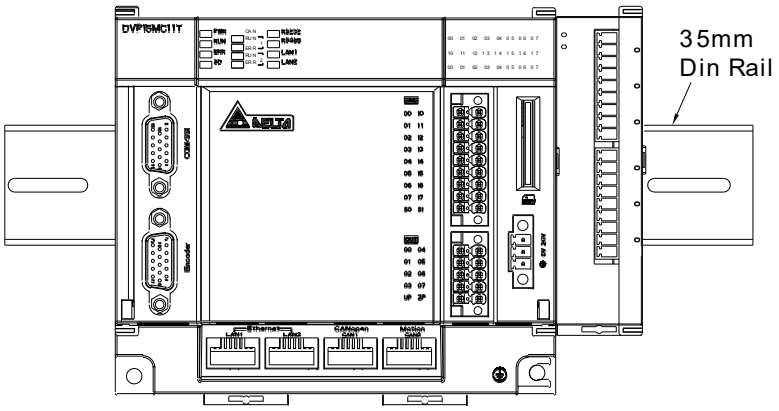
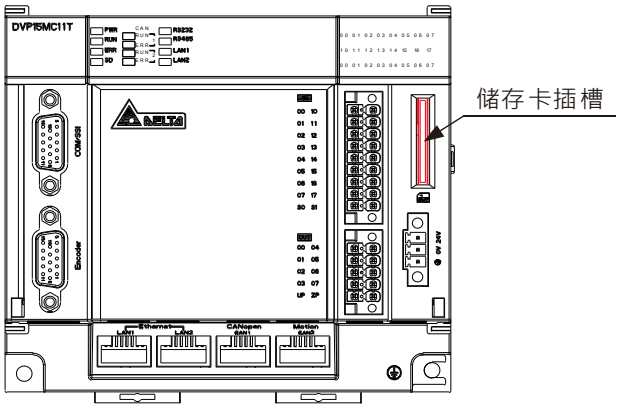


图 5.1.4.2

5.1.5 SD 卡安装与拆除方法

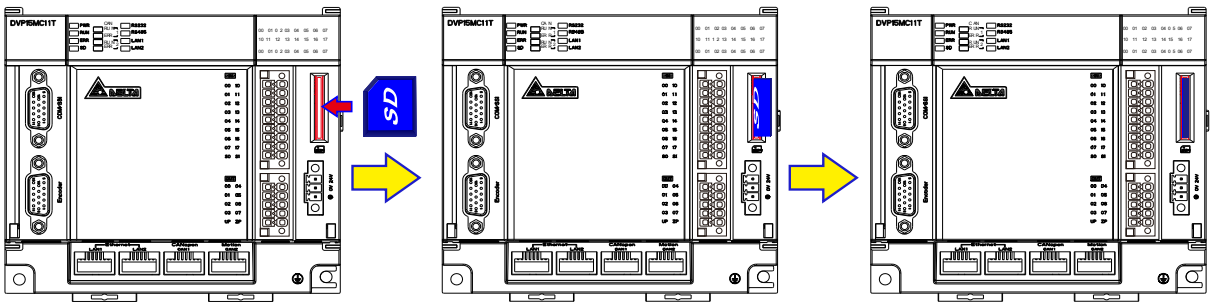
- **主机上的储存卡插槽**

主机的储存卡插槽都被安排在机体正面的右方，如下图所示。



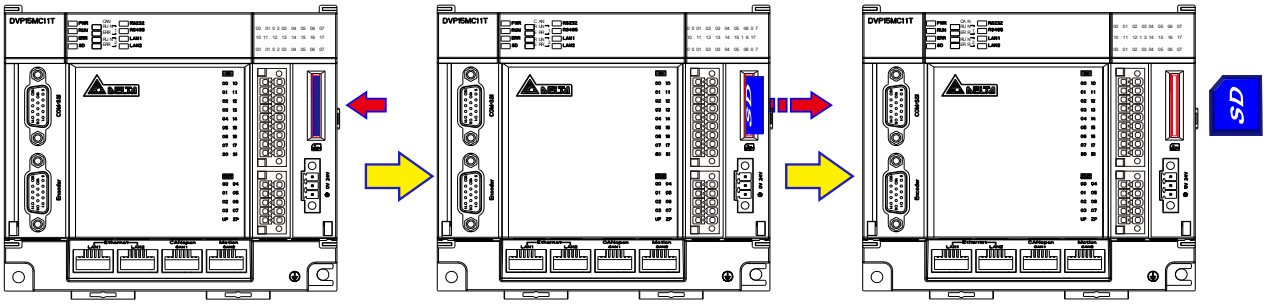
- **储存卡的安装**

直接将储存卡垂直地插入主机的储存卡插槽并将其推至底部，直到听见卡榫固定的声音即可。顺利安装后，SD 卡应该会被牢牢的固定住，若仍是松脱的状态表示并未安装正确。另外，SD 卡本身有防呆设计，若 SD 卡插入的方向错误便会无法将其推至插槽底部，此时请勿强制推入以免造成机体的损坏。插入 SD 卡的正确方向请参考下方图示。



- **储存卡的卸除**

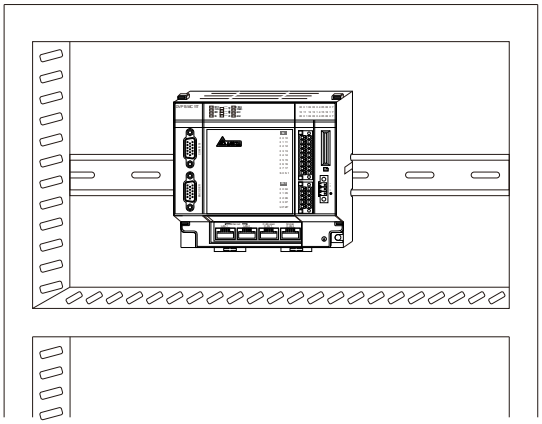
  直接将储存卡推至底部后，储存卡即会松脱弹出，此时便可将其取出。



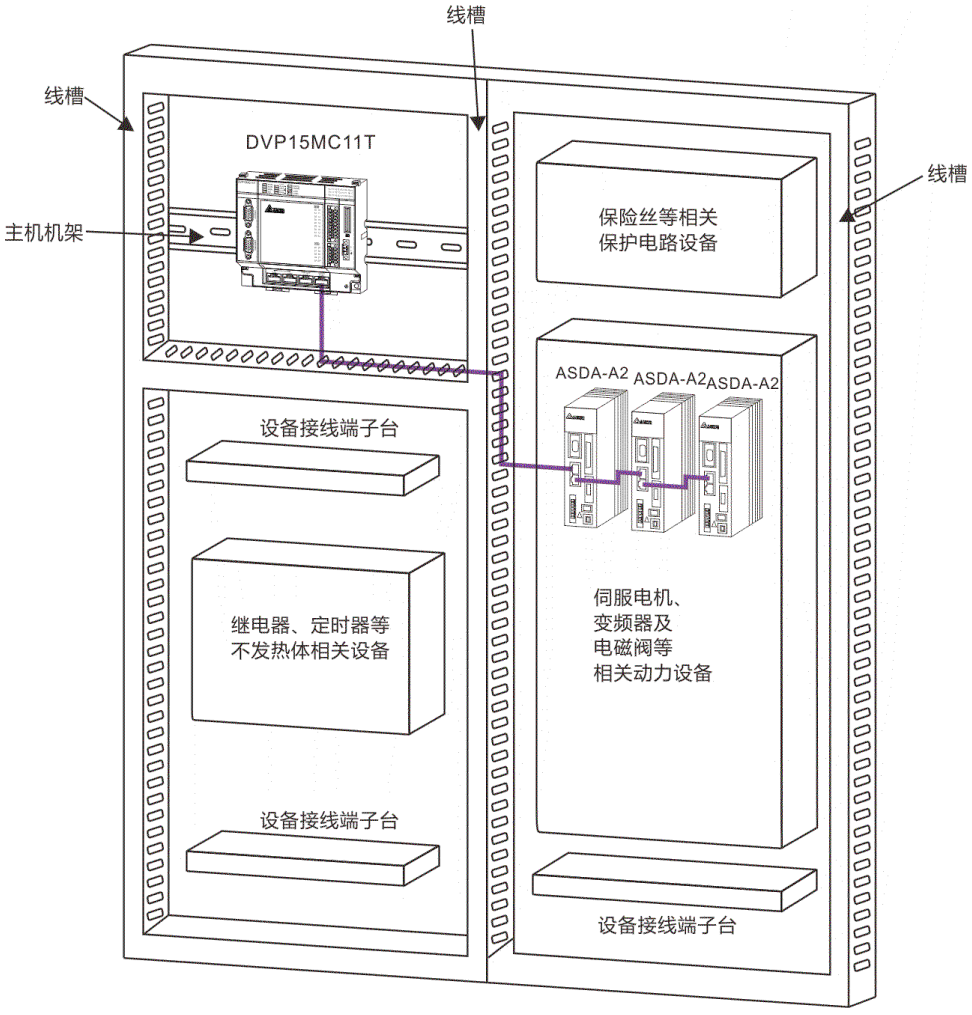
5.2 安装至控制柜

5.2.1 安装至 DIN 导轨

将 DVP-15MC 系列运动控制器模块底部固定扣拉出，将模块底部横槽卡入控制柜中间导轨上，再推上固定扣，则 DVP-15MC 系列运动控制器模块被固定在控制柜内。



5.2.2 控制柜内的安装方法



5.2.3 控制柜内的环境温度要求

- **控制柜内安装 DVP-15MC 系列运动控制器的要求：**

  1. DVP-15MC 系列运动控制器在控制柜里的环境温度为 0°C ~ 55°C （温度）， 5 ~ 95% （湿度）。
  2. 请避免安装在高燃烧或温度易高的设备附近。
  3. 保留足够的通风空间。
  4. 超过环境温度 55°C，必须安装风扇或空调。

⚠ **注意**

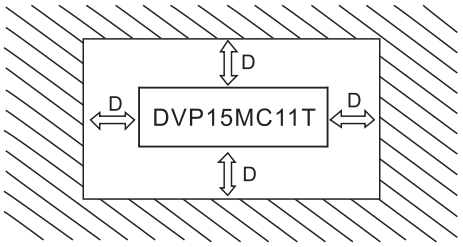
1. 在控制柜安装时，1.0m~2.0m 的高度易于安装和操作。
2. 安装时请远离高压设备和动力设备。
3. 控制柜里电路状态要在断电情况下，才能安装，不可带电安装。

5.2.4 提高抗干扰性的措施

1. 请避免安装在装有高压设备的柜里。
2. 安装时请距离动力线 200mm 以上。
3. 控制柜要有接地线。

5.2.5 控制柜内的尺寸要求

- **盘内安装**

DVP-15MC 系列运动控制器在安装时，请安装于封闭式的控制箱内，其周围应保持一定的空间，以确保其散热功能正常（如右图所示） D > 50mm	

6

第6章 接线、通讯设定及组网

目录

6.1 电源配线

6.1.1 电源输入

DVP-15MC 系列运动控制器电源输入为直流 24V，在使用上应：

⚠ **注意**

- 电源输入电压范围（20.4 VDC ~ 28.8VDC），电源请接于 24V、0V 两端，同时将接地端接地。如果将电源的**正负接反**Ⓧ，容易使 DVP-15MC 系列运动控制器损坏，请用户注意。
- 主机的接地端使用 1.6mm 以上的电线接地。
- 当掉电时间过长或电源电压下降将使 DVP-15MC 系列运动控制器停止运转，输出全部 FALSE 时，停止与伺服驱动器通讯。当电源恢复正常时，DVP-15MC 系列运动控制器将与伺服驱动器重新建立连接。

6.1.2 安全回路配线

由于 DVP-15MC 系列运动控制器控制伺服驱动器，它内部的任一装置的动作都可能影响到外部机械结构的动作，因此任一装置的故障都可能造成整个自动控制系统失控，甚至造成人员伤亡。所以我们建议在电源输入回路中，建议有如下的保护装置。

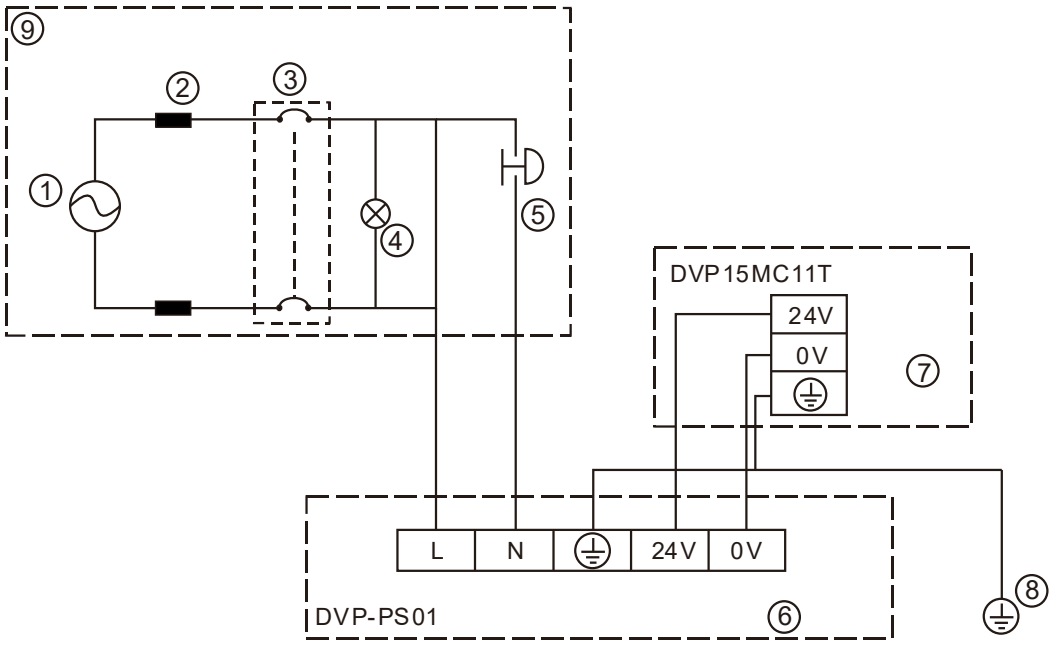


图 6.1.2.1

①	交流电源供应：100 ~ 240VAC；50/60Hz。
②	电源回路保护用保险丝
③	系统回路隔离装置：使用电磁接触器、继电器等开关作为系统电源回路隔离装置，可防止电源断续供电时，造成系统的不稳定。
④	电源指示灯
⑤	紧急停止：为预防突发状况发生，设置的紧急停止按钮，可在状况发生时，切断系统电源。

⑥ 台达电源模块 DVPPS02/24VDC（建议 DVP-15MC 系列运动控制器的电源模块采用 DVPPS02）
⑦ DVP-15MC 系列运动控制器机体
⑧ 接地
⑨ 安全回路

6.2 输入和输出点配线

6.2.1 输入点支持的功能

DVP-15MC 系列运动控制器有 16 个输入点，输入点支持外部中断功能和滤波功能；可以通过输入点捕获编码器位置，详细说明请参考位置捕获指令说明。

- **输入滤波器原理及说明**

输入滤波器是 I0~I7、I10~I17 这 16 个 I 点对短脉冲信号的过滤，从而降低输入干扰信号的影响，通过增大滤波值，可减少输入信号的抖动或外部干扰的影响。

输入滤波时间：t=31us *（0~255） 。因此滤波时间是 31us 的倍数，默认值为 0。输入滤波时间可以通过软件设定。

- **当有滤波设定时,如下图所示：**

如图 6.2.1.1，当滤波时间设定 t（us），输入信号高电平时间或者低电平时间大于 t（us）时，信号有效，输入信号小于 t（us）时，信号被过滤掉；滤波后的输入信号延迟 t（us）之后会被输入。

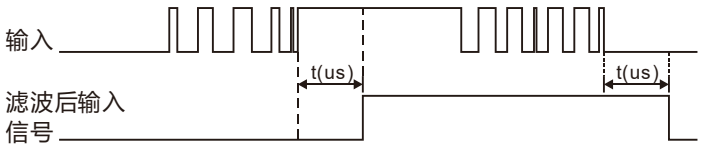


图 6.2.1.1

- **当无滤波设定时，如下图所示：**

如图 6.2.1.2，当不设定滤波时间时，输入信号无变化。

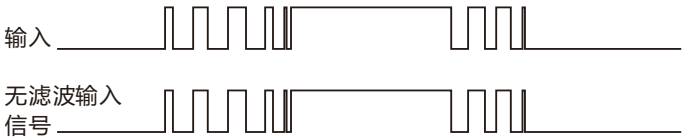


图 6.2.1.2

6.2.2 输入点配线

输入点的输入信号为直流电源输入，共有两种接法：漏型模式（Sink）与源型模式（Source）。详细接法如下：

- **漏型模式（Sink）**

漏型模式特征为电流流入公共端 S0 和 S1，其简化模型如图 6.2.2.1 所示：

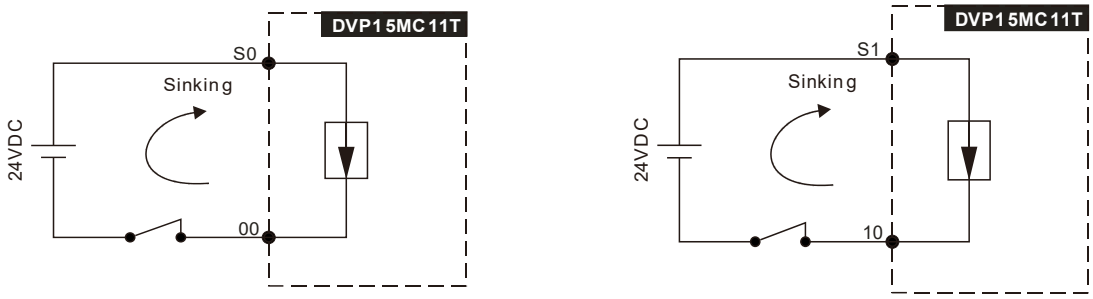


图 6.2.2.1

相关配线回路如下所示：

- DVP-15MC 系列运动控制器输入点 00~07 对应 S0 接点如下图所示：

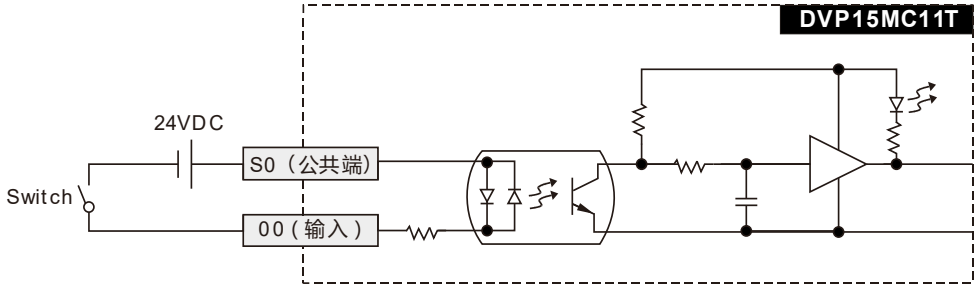


图 6.2.2.2

- DVP-15MC 系列运动控制器输入点 10~17 对应 S1 接点如下图所示：

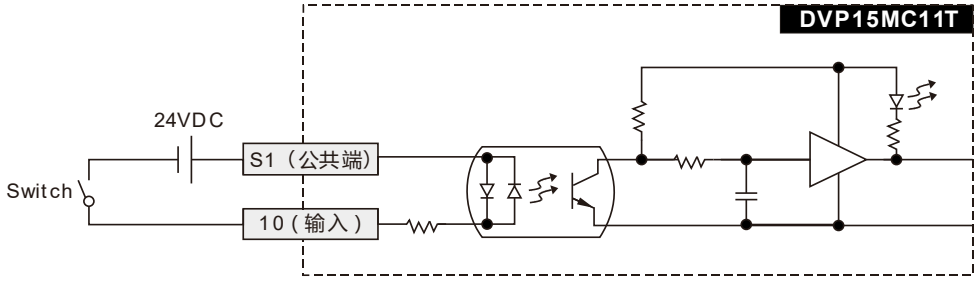


图 6.2.2.3

- **源型模式（Source）**

源型模式特征为电流流出公共端 S0 和 S1，其简化模型如下图所示：

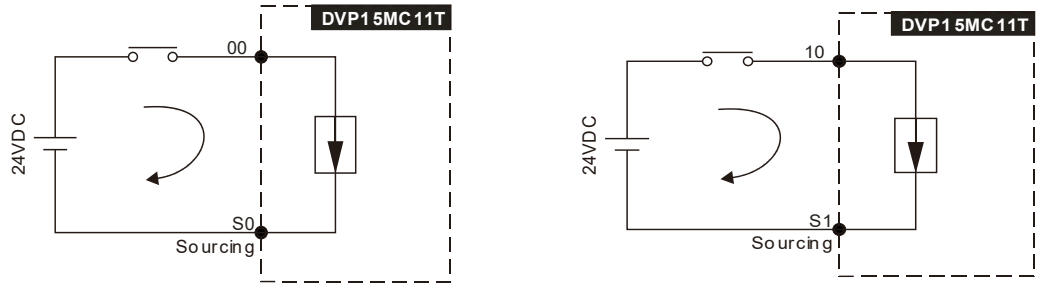


图 6.2.2.4

相关配线回路如下图所示：

- DVP-15MC 系列运动控制器输入点 00~07 对应 S0 接点如下图所示：

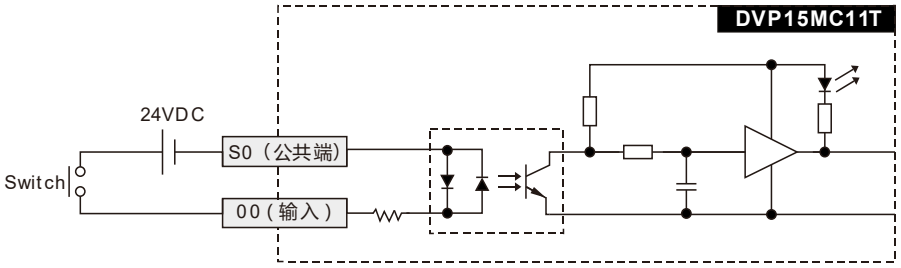


图 6.2.2.5

- DVP-15MC 系列运动控制器输入点 10~17 对应 S1 接点如下图所示：

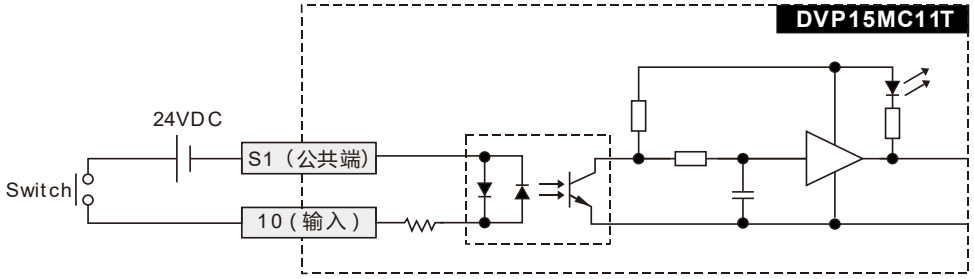


图 6.2.2.6

6.2.3 输出点配线

DVP-15MC 系列运动控制器晶体管输出均已包含反电势保护二极管，对于小功率电感性负载，且 ON/OFF 频率不高的应用已经足够，但在大功率或 ON/OFF 频繁的场合，请依下列方法另接抑制电路以降低干扰及防止过电压或过热而损坏晶体管输出电路。

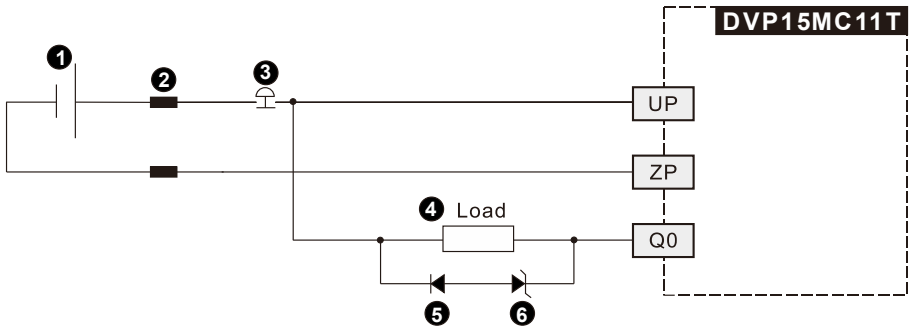


图 6.2.3.1

❶ 24V 直流电源；
❷ 电路回路保护用保险丝；
❸ 急停按钮；
❹ 负载：开关、电感类负载等；
❺ 二极管或等效元件，用于二极管抑制。(负载功率较小时只需使用❺， ❻无需使用。)
❻ 9V 齐纳二极管，5W；(负载为大功率且 ON/OFF 频繁时，❺和❻一起使用。

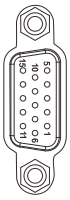
6.3 RS-485 通讯口

6.3.1 RS-485 支持的功能

DVP-15MC 系列运动控制器的 RS-485 通讯口可以做 Modbus 主站或从站，HMI、PLC 或者其它 Modbus 主站设备可以对 DVP-15MC 系列运动控制器内部装置进行数据读写操作。Modbus 主站对 DVP-15MC 系列运动控制器的访问间隔时间需大于 5ms，不能通过 RS-485 通讯口下载程序。RS-485 支持 Modbus 通讯协议，支持 ASCII 模式和 RTU 模式。支持的功能码有 0x01、0x02、0x03、0x05、0x06、0x0F、0x10。站号支持 1~255，不支持广播功能。具体 Modbus 通讯说明及装置的 Modbus 地址见附录 A。

6.3.2 RS-485 通讯口引脚定义

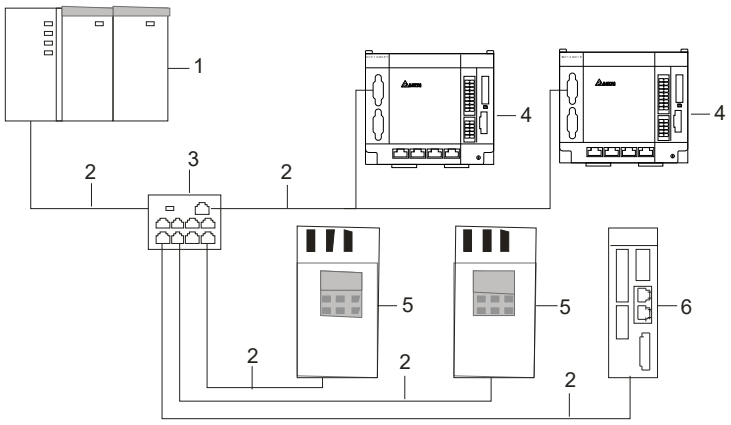
DVP-15MC 系列运动控制器的 COM/SSI 接口为 15 针母头接口，RS-485 通讯口和 SSI 绝对型编码器接口共用一个外部接口，RS-485 通讯口引脚定义如下：

引脚	信号	叙述	
11	D+	RS-485 正极	
12	D-	RS-485 负极	
5	SG	RS-485 信号地	COM/SSI

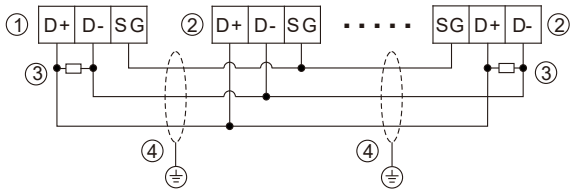
6.3.3 RS-485 硬件连接

- **DVP-15MC 系列运动控制器接入 Modbus 网络示例：**

DVP-15MC 系列运动控制器通过 RS-485 接入 Modbus 网络示意图如下图所示：



设备编号	1	2	3	4	5	6
设备名称	Modbus 主站	通讯电缆	VFD-CM08	DVP-15MC 系列运动控制器	变频器	伺服驱动器



- **RS-485 建议接线方式：**
  - **编号说明**

①	②	③	④
主站	从站	终端电阻	屏蔽线

  - **附注：**
    - 建议在总线两端各接一终端电阻，阻值为 120Ω。
    - 为确保联机质量，线材建议使用具有双层屏蔽线的通讯双绞线（20AWG）。
    - 当两个系统内部地准位存在压降，可通过连接 SG（Signal Ground）让地准位等电位，使通讯更加稳定。

- **RS-485 支持的通讯格式**

RS-485 通讯口支持 ASCII 或 RTU 通讯格式，波特率最高可达 115200bps。

波特率	9600；19200；38400；57600；115200					
模式	ASCII				RTU	
通讯格式	7,E,1	7,E,2	7,N,1	7,N,2	8,E,1	8,E,2
	7,O,1	7,O,2	8,E,1	8,E,2	8,N,1	8,N,2
	8,N,1	8,N,2	8,O,1	8,O,2	8,O,1	8,O,2

6.3.4 RS-485 支持的功能码和异常功能码

- **Modbus 功能码介绍：**
  - **DVP-15MC 系列运动控制器 RS-485 通讯口支持的功能码如下表所示：**

功能码	说明	可否广播	读/写最大值	可操作装置
0x01	读输出位装置寄存器的值。	否	256 个	位装置
0x02	读位装置寄存器的值。	否	256 个	位装置
0x03	读单个或多个字装置寄存器的值。	否	100 个	字装置
0x05	写单个位装置寄存器的值。	是	1 个	位装置
0x06	写单个字装置寄存器的值。	是	1 个	字装置
0x0F	写多个位装置寄存器的值。	是	256 个	位装置
0x10	写多个字装置寄存器的值。	是	100 个	字装置

  - **DVP-15MC 系列运动控制器 RS-485 通讯口支持的异常回应码如下表所示：**

异常回应码	含义
0x01	不支持的功能码
0x02	不支持的 Modbus 地址
0x03	数据长度超出范围

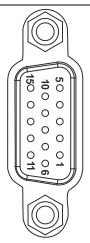
6.4 RS-232 通讯口

6.4.1 RS-232 支持的功能

DVP-15MC 系列运动控制器的 RS-232 通讯口可以做 Modbus 主站或从站，HMI、PLC 或者其它 Modbus 可以对 DVP-15MC 系列运动控制器内部装置进行数据读写操作。不能通过 RS-232 通讯口下载程序。RS-232 支持 Modbus 通讯协议，支持 ASCII 模式和 RTU 模式。支持的功能码有 0x01、0x02、0x03、0x05、0x06、0x0F、0x10。站号支持 1~255，不支持广播功能。具体 Modbus 通讯说明及装置的 Modbus 地址见附录 A。

6.4.2 RS-232 通讯口引脚定义

DVP-15MC 系列运动控制器的 COM/SSI 接口为 15 针接口， RS-232 通讯口引脚定义如下：

引脚	信号	叙述	
3	Tx	RS-232 发送数据	
9	Rx	RS-232 接收数据	
5	GND	RS-232 信号地	COM/SSI

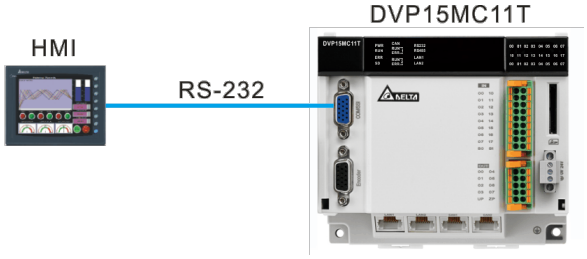
6.4.3 RS-232 硬件连接

DVP15MC11T
COM (RS-232接线)

功能	脚位
	1
	2
TXD	3
	4
GND	5
	6
	7
	8
RXD	9
	10
	11
	12
	13
	14
	15

脚位	功能
1	
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	

- **DVP-15MC 系列运动控制器作为从站时，RS-232 接口连接 HMI。**



- **RS-232 支持的通讯格式如下表所示**

波特率	9600；19200；38400；57600；115200					
模式	ASCII				RTU	
通讯格式	7,E,1	7,E,2	7,N,1	7,N,2	8,E,1	8,E,2
	7,O,1	7,O,2	8,E,1	8,E,2	8,N,1	8,N,2
	8,N,1	8,N,2	8,O,1	8,O,2	8,O,1	8,O,2

6.4.4 RS-232 支持的功能码和异常功能码

- **Modbus 功能码介绍：**
  - **DVP-15MC 系列运动控制器 RS-232 通讯口支持的功能码如下表所示：**

功能码	说明	读/写最大值	可操作装置
0x01	读输出位装置寄存器的值。	256 个	位装置
0x02	读位装置寄存器的值。	256 个	位装置
0x03	读单个或多个字装置寄存器的值。	100 个	字装置
0x05	写单个位装置寄存器的值。	1 个	位装置
0x06	写单个字装置寄存器的值。	1 个	字装置
0x0F	写多个位装置寄存器的值。	256 个	位装置
0x10	写多个字装置寄存器的值。	100 个	字装置

  - **DVP-15MC 系列运动控制器 RS-232 通讯口支持的异常回应码如下表所示：**

异常回应码	含义
0x01	不支持的功能码
0x02	不支持的 Modbus 地址
0x03	数据长度超出范围

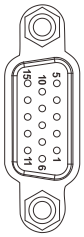
6.5 SSI 绝对型编码器接口

6.5.1 SSI 绝对型编码器功能

DVP-15MC 系列运动控制器的 COM/SSI 接口为 15 针的 D-SUB 接座，可以用于连接 SSI 编码器，此接口还集成 5V（400mA）电源输出，提供给编码器电源。用户可以建立 SSI 编码器，通过编码器接收脉冲数控制从轴运动。

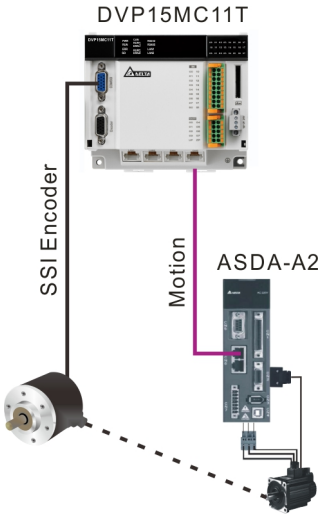
6.5.2 SSI 接口的引脚定义

DVP-15MC 系列运动控制器的 COM/SSI 接口为 15 针 D-SUB 接口，SSI 通讯口引脚定义如下：

引脚	信号	叙述	
1	DATA+	SSI 绝对型编码器数据正极	 COM/SSI
2	DATA-	SSI 绝对型编码器数据负极	
6	CLK+	SSI 绝对型编码器时钟正极	
14	CLK-	SSI 绝对型编码器时钟负极	
8	GND	SSI 绝对型编码器电源接地	
15	5V	SSI 绝对型编码器供电电源	

6.5.3 SSI 绝对型编码器硬件连接

- **SSI 绝对型编码器接线示意图**



- **SSI 绝对型编码器接口接线规格**

  DVP-15MC 系列运动控制器的 SSI Encoder 接口与其接线方式如下：

DVP15MC11T
COM/SSI

SSI 编码器

脚位	功能
1	DATA+
2	DATA-
6	CLK+
14	CLK-
15	5V
8	GND

功能
DATA+
DATA-
CLK+
CLK-
VCC
0V

备注：DVP-15MC 系列运动控制器的 COM/SSI 接口提供的是 5V 电源。

当 VCC = 5V 时，可以将 SSI 编码器电源电压 VCC 直接与 COM/SSI 接口的脚位 15 连接，将 SSI 编码器的 0V 与 COM/SSI 接口的脚位 8 连接；

当 VCC ≠ 5V 时，请根据连接的 SSI 编码器实际电源电压对 SSI 编码器进行单独供电。

- **SSI 绝对型编码器通讯电缆线规格**

  CLK+、CLK - 和 DATA +、DATA – 的信号电缆，请使用带屏蔽的双绞线电缆。

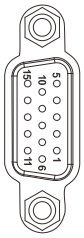
6.6 增量型编码器

6.6.1 增量型编码器功能

DVP-15MC 系列运动控制器 Encoder（增量型编码器）接口为 15 针的 D-SUB 接座，可以用于连接两组独立的增量型编码器。这两组编码器接口均支持差分信号输入，每组编码器的最大工作频率可达到 1MHz（每输入 250K x 4 = 1MHz），同时此接口还集成两组 5V（400mA）电源输出，提供给两组编码器电源。每组编码器均可以建立一个增量型编码器，通过 Encoder（增量型编码器）接口接收脉冲数控制从轴运动。

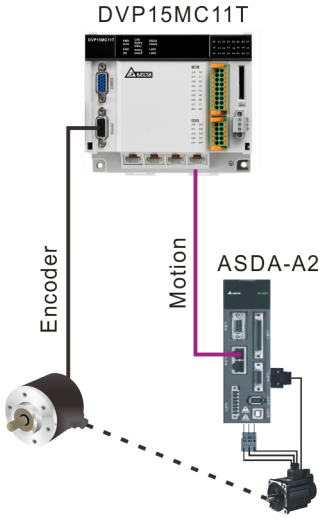
6.6.2 增量型编码器引脚定义

DVP-15MC 系列运动控制器的 Encoder（增量型编码器）接口为 15 针接口，增量型编码器通讯口引脚定义如下：

引脚	信号	叙述	
1	A1+	第一组增量型编码器差分信号	 Encoder
2	A1-		
10	B1+		
11	B1-		
4	Z1+		
5	Z1-		
15	+5V	第一组编码器供电电源	
3	A2+	第二组增量型编码器差分信号	
9	A2-		
6	B2+		
12	B2-		
13	Z2+		
14	Z2-		
7	+5V	第二组编码器供电电源	
8	0V	两组编码器电源共用 0V	
机壳	-	屏蔽层	

6.6.3 增量型编码器硬件连接

- **增量型编码器接线示意图**



- **增量型编码器接口接线规格**

DVP-15MC 系列运动控制器的 Encoder 接口与其接线方式如下：

DVP15MC11T
Encoder

编码器

脚位	功能
1	A1+
2	A1-
10	B1+
11	B1-
4	Z1+
5	Z1-
15	+5V
8	GND

功能
A
$\bar{A}$
B
$\bar{B}$
Z
$\bar{Z}$
Vcc
0V

备注：DVP-15MC 系列运动控制器的 Encoder 接口提供的是 5V 电源。

1. 当 VCC = 5V 时，可以将编码器电源电压 VCC 直接与 Encoder 接口的脚位 15 连接，将编码器的 0V 与 Encoder 接口的脚位 8 连接；
2. 当 VCC ≠ 5V 时，请根据连接的编码器实际电源电压对编码器进行单独供电。

6.7 Ethernet 通讯口

6.7.1 Ethernet 通讯口支持的功能

DVP-15MC 系列运动控制器有两个独立的以太网通讯口 LAN1 和 LAN2，每个以太网通讯口需要独立设置 IP 地址。

两个以太网通讯口均可用于下载配置文件、可执行文件和 CAM 文件。此外，两个以太网通讯口还支持自动跳线功能，当与计算机或交换机连接时，不需要特别作跳线处理。两个以太网通讯口还可以自动检测 10/100 Mbps 传输速率。

HMI、PLC 或者其它 ModbusTCP 主站设备可以通过这两个以太网通讯口对 DVP-15MC 系列运动控制器内部装置进行数据读写操作。具体 Modbus TCP 通讯说明见**附录 B**。

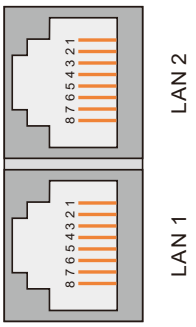
两个以太网通讯口均支持 Modbus TCP 协议，其中 LAN1 端口只能做为从站，LAN2 端口既可做为主站，也可做为从站。LAN2 端口支持 EtherNet/IP 从站功能和 Socket 功能。两个端口的详细规格请参考下表说明：

项目			LAN1 端口	LAN2 端口
通讯协议			MODBUS TCP	MODBUS TCP、Socket、EtherNet/IP
MODBUS TCP	联机数（Server）		16	16
	联机数（Client）		0	
Socket	TCP 联机数		0	8
	UDP 联机数			
EtherNet/IP	设备类别		Adapter	Adapter
	CIP_IO Connection	CIP 联机数	不支持	8
		TCP 联机数		16
		报文传送间隔时间（RPI）		5ms~1000ms
		最大数据量/单笔		200bytes
	CIP_Explicit Message	Class 3（Connected Type）	不支持	8
		UCMM（Non-Connected Type）		16
		支持 CIP 对象		Identity、Message Router、Assembly、Connection Manager、Port、TCP/IP interface、Ethernet link、Vendor specific

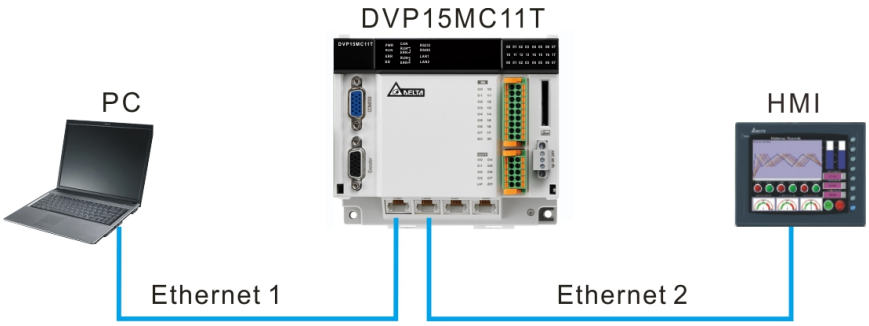
6.7.2 Ethernet 通讯口引脚定义

DVP-15MC 系列运动控制器有两个独立的以太网通讯口 LAN1 和 LAN2，这两个通讯口各自独立，每个通讯口需要独立设置 IP 地址，LAN1 的默认 IP 地址为 192.168.0.1；LAN2 的默认 IP 地址为 192.168.1.1。两个通讯口引脚定义完全相同，定义如下：

引脚	信号	叙述
1	Tx+	传输数据正极
2	Tx-	传输数据负极
3	Rx+	接收数据正极
4	保留	保留
5	保留	保留
6	Rx-	接收数据负极
7	保留	保留
8	保留	保留



6.7.3 Ethernet 通讯口网络连接



6.7.4 Ethernet 通讯口支持的功能码

DVP-15MC 系列运动控制器的 Ethernet LAN1 和 LAN2 通讯口使用 Modbus TCP 协议时支持的功能码及异常回应码如下表所示：

功能码	说明	读/写最大值	可操作装置
0x02	读位装置寄存器的值。	256 个	位装置
0x03	读单个或多个字装置寄存器的值。	100 个	字装置
0x05	写单个位装置寄存器的值。	1 个	位装置
0x06	写单个字装置寄存器的值。	1 个	字装置
0x0F	写多个位装置寄存器的值。	256 个	位装置
0x10	写多个字装置寄存器的值。	100 个	字装置

异常回应码	含义
0x01	不支持的功能码
0x02	不支持的 Modbus 地址
0x03	数据长度超出范围

6.8 Motion 通讯口

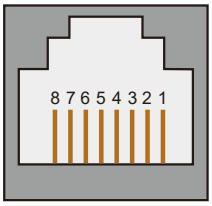
6.8.1 Motion 通讯口支持的功能

Motion 通讯口用作运动控制用。运动指令通过该通讯口控制伺服。可以通过该通讯口发送 SDO 命令。用户不能通过该通讯口进行 PDO 配置。

6.8.2 Motion 通讯口引脚定义

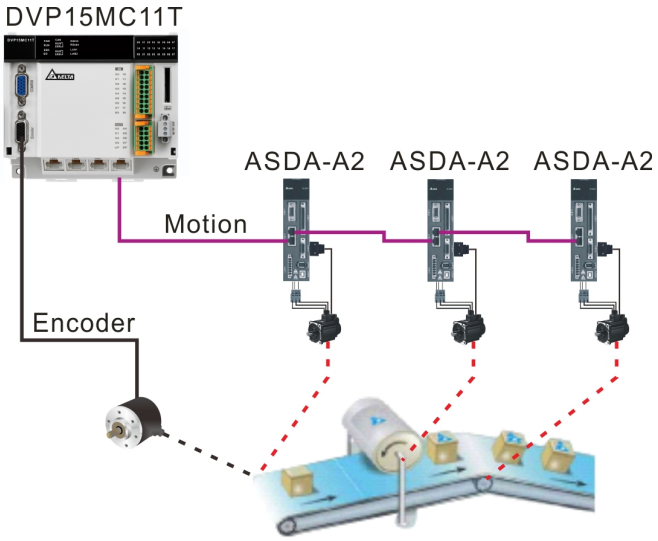
Motion 通讯口用作运动控制用，引脚定义如下图所示。

引脚	信号	叙述
1	CAN_H	Signal+
2	CAN_L	Signal-
3	CAN_GND	0 VDC
4	保留	保留
5	保留	保留
6	保留	保留
7	CAN_GND	0 VDC
8	保留	保留



Motion
CAN2

6.8.3 Motion 网络连接



注：DVP-15MC 系列运动控制器的 Motion 接口内嵌 120 欧姆的终端电阻。

6.8.4 Motion 通讯速率与通讯距离

Motion 总线网络的传输距离由 Motion 总线传输速率决定，下表所示为不同传输速度对应的最大通讯距离。

传输速度（位/秒）	500K	1M
最大通讯距离（米）	100	25

6.9 CANopen 通讯口

6.9.1 CANopen 通讯口支持的功能

CANopen 通讯口可以作为 CANopen 网络的主站使用，也可以作为其它主站的一个从站来使用。

- **当作为主站使用时，有如下功能：**
  - 支持 CANopen 标准协议 DS301v4.02
  - 支持 NMT（Network Management Object：网络管理对象） Master 服务
  - 支持 NMT 错误控制

    NMT 错误控制用于监控从站是否掉线。NMT 错误控制分为 Heartbeat 和 NodeGuarding 两种。本机支持 Heartbeat。
  - 最多可以连接 32 个从站
  - 支持 PDO（Process Data Object：过程数据对象）服务：

    RxPDO 最大支持 200 个，数据量最大支持 1000 个字节

    TxPDO 最大支持 200 个，数据量最大支持 1000 个字节

    每个从站最多可配置 8 个 TxPDO 和 8 个 RxPDO

    PDO 传输类型：支持事件触发，时间触发，同步周期，同步非周期

    PDO 映射：每个 PDO 最大可以映射 32 个参数

    支持的映射数据类型：

存储空间	数据类型
1bit	BOOL
8bit	SINT USINT BYTE
16bit	INT UINT WORD
32bit	DINT UDINT REAL DWORD
64bit	LINT ULINT LREAL LWORD

  - 支持 SDO 服务：

    支持标准 SDO 快速（expedited SDO）传输模式

    支持 Auto SDO 功能，最大可对每一台从站执行 30 笔 Auto SDO

    支持在 PLC 梯形图中使用 SDO 服务读写从站数据
  - 支持同步信号产生器（SYNC producer，range 0-65535ms）

    通过同步报文，可实现多个设备同步动作
  - 支持的 CANopen 通讯速率：20K,50K,125K,250K,500K,1Mbps
- **当作为从站使用时，有如下功能：**
  - 支持 CANopen 标准协议 DS301v4.02
  - 支持 NMT Slave 服务
  - 支持 NMT 错误控制

    支持 Heartbeat Protocol 错误控制，不支持 Node Guarding 错误控制。

- 支持 PDO 服务

  RxPDO 最大支持 8 个，数据量最大支持 64 个字节；

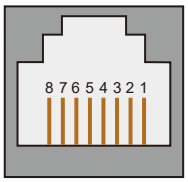
  TxPDO 最大支持 8 个，数据量最大支持 64 个字节。
- PDO 传输类型：支持事件触发，时间触发，同步周期，同步非周期。
- 支持 SDO 服务

  支持标准 SDO 快速（expedited SDO）传输模式。

6.9.2 CANopen 通讯口引脚定义

DVP-15MC 系列运动控制器 CANopen 通讯口用作标准 CANopen 通讯，引脚定义如下图所示。

引脚	信号	叙述
1	CAN_H	Signal+
2	CAN_L	Signal-
3	CAN_GND	0 VDC
4	保留	保留
5	保留	保留
6	保留	保留
7	CAN_GND	0 VDC
8	保留	保留



CANopen

CAN1

6.9.3 CANopen 通讯口的 PDO 映射

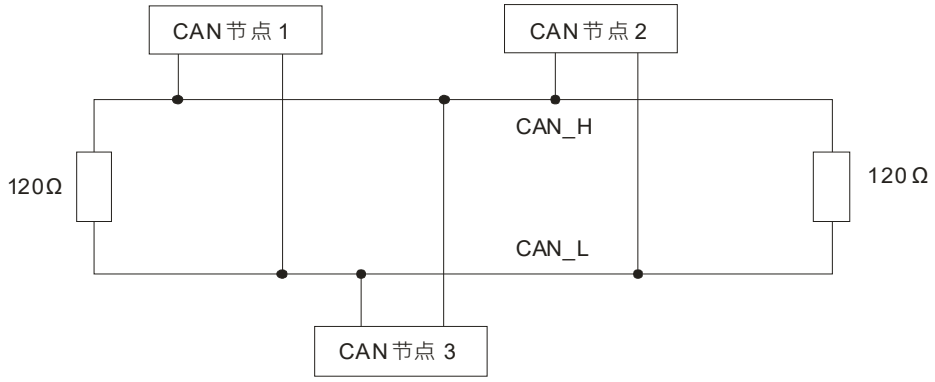
DVP-15MC 系列运动控制器作为 CANopen 主站时，输入映射区为%MW5000~%MW5499，输出映射区为%MW5500~%MW5999。

DVP-15MC 系列运动控制器作为 CANopen 从站时，输入映射区为%MW5000~%MW5031，输出映射区为%MW5500~%MW5531。

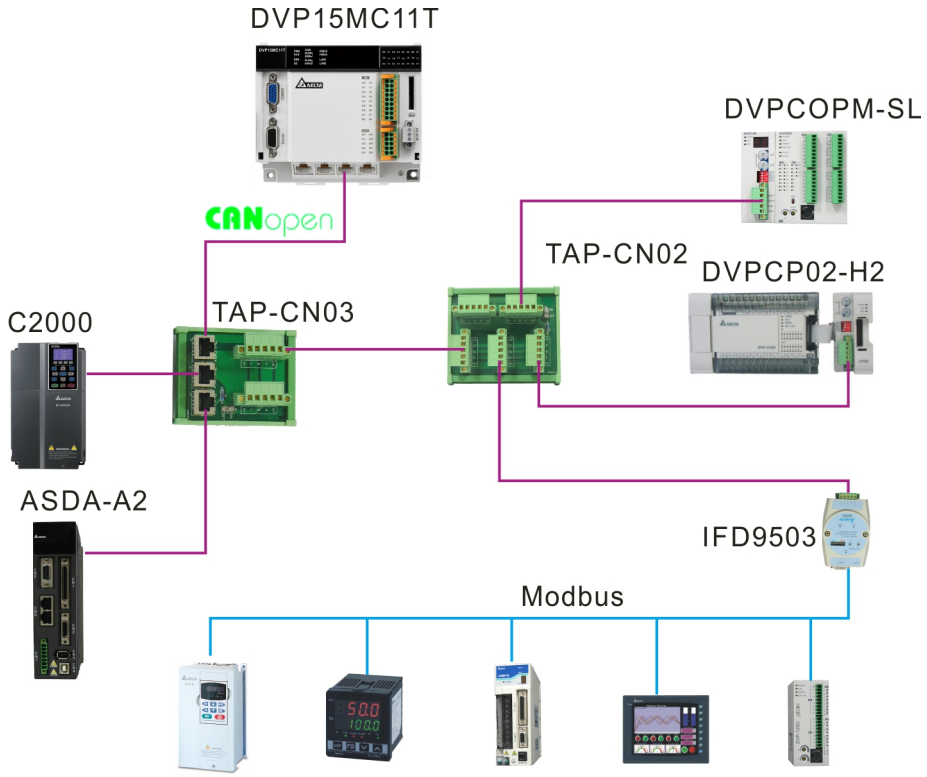
6.9.4 CANopen 通讯口网络连接

- **CANopen 总线终端和网络拓扑结构：**

为了增强 CANopen 通讯的稳定性，CANopen 总线网络的两个终端需接入 120 欧姆的终端电阻。下图所示为基本的 CANopen 网络拓扑结构示意图。



- **CANopen 总线网络拓扑架构**



  - 组建 CANopen 网络时建议使用台达标准电缆：UC-DN01Z-01A 粗缆、UC-DN01Z-02A 细缆和 UC-CMC010-01A 细缆，并且通讯电缆须远离动力电缆。
  - 请在网络两端的 CAN_H 和 CAN_L 之间分别串接电阻值为 120 欧姆的电阻。用户可自行购买台达的标准终端电阻 TAP-TR01.

6.9.5 CANopen 通讯口通讯速率与通讯距离

CANopen 总线网络的传输距离由 CANopen 总线传输速率决定，下表所示为不同传输速率对应的最大通讯距离。

传输速度（位/秒）	20K	50K	125K	250K	500K	1M
最大通讯距离（米）	2500	1000	500	250	100	25

MEMO

7

第7章 控制器执行原理

目录

7.1 任务

- 任务是指为 I/O 刷新和执行用户程序等一系列处理指定执行条件和执行顺序的功能。
- 任务通过名字，优先级以及类型来定义。类型可以定义为周期任务、自由运行任务、事件任务。
- 每个任务都可以指定一系列由该任务触发的 POU。 如果该任务在当前周期被执行，那么这些 POU 将在一个周期时间内被处理。
- 优先级和任务类型
- 综合决定任务的执行顺序。
- 每个任务都可以配置一个看门狗。

7.1.1 任务类型

- **DVP-15MC 系列运动控制器支持三种任务类型**

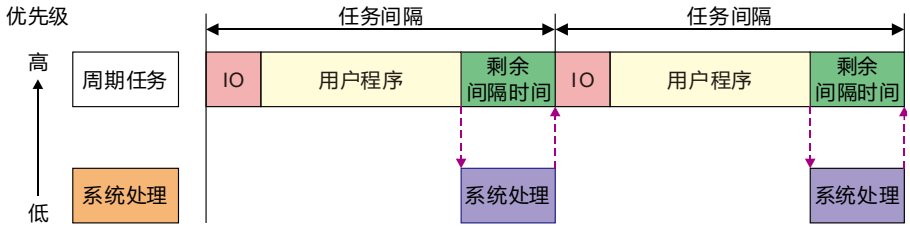
  1. 周期任务； 2. 自由运行任务；3. 事件任务。

- **支持的最大任务数目为 24，分别说明如下：**

  - **周期任务**

    周期任务将按照“间隔”项设定的时间周期执行。

    - **周期任务执行方式如下：**



IO：表示 I/O 刷新。 I/O 包括本机 I/O 点和左右侧扩展模块数据、CANopen 数据。此数据可以指定在设定的任务执行前更新；未指定时，此数据在系统处理中刷新。

用户程序：表示用户程序的执行，按照任务中程序分配顺序执行。

剩余间隔时间：控制器执行系统处理；如果有低优先级的任务，则会先执行低优先级的任务，然后再执行系统处理。

系统处理：控制器执行完所有任务请求后执行系统处理；系统处理包括 Ethernet 通讯处理、RS232、RS485 通讯处理等。

后续章节中出现的上述四个用词，其表达的含义均和上面描述相同。

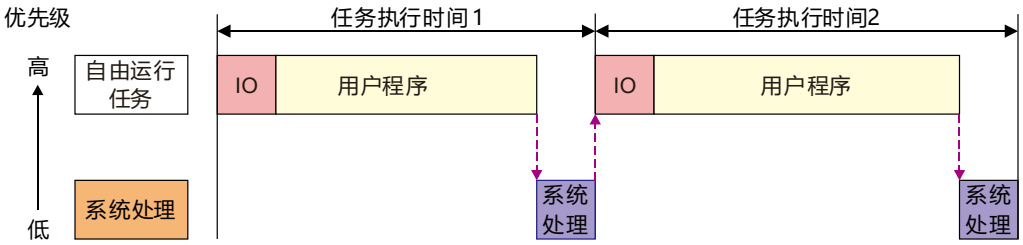
⚠ 注意

如果为某个周期任务定义的周期过短，则该任务会在执行完用户程序后立即重复执行，而不会执行其它低优先级的任务或任何系统处理，这将会影响所有任务的执行。如果为该任务设置了看门狗，则会导致看门狗超时，控制器会进入 Error 状态，并且停止执行用户程序；如果没有为任务设置任务看门狗，控制器会无法执行系统处理，出现通讯逾时等问题。

  - **自由运行任务**

    程序一开始运行任务就被处理，一个运行周期结束后，任务将在下一个周期中被自动重新启动。

➢ **自由运行任务执行方式如下：**



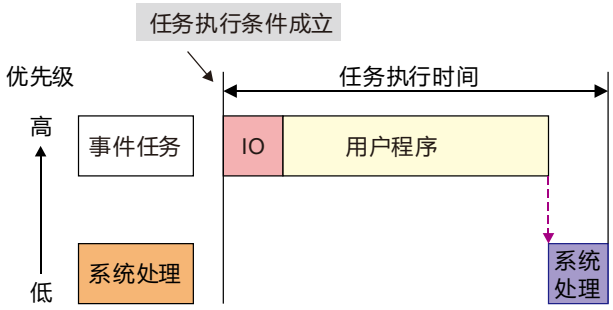
⚠ 注意

自由运行任务没有固定持续时间；所以上图中的任务执行时间 1 和任务执行时间 2 可能不相同。

■ **事件任务**

事件任务是指仅在指定的事件发生时执行一次。事件任务的执行时机取决于事件发生的时机和事件任务的优先级。

➢ **事件任务执行方式如下：**



➢ **可供选择的事件包含以下几种：**

- 运动事件（运动控制任务）；
- 本机输入点（I0~I7，I10~I17）的上升沿或下降沿；
- CANopen 网络的同步信号；
- 增量型编码器 1 或增量型编码器 2 的 Z 脉冲信号的上升沿；

事件任务完成前，该事件任务的执行条件再次成立时，忽略第 2 次的执行条件成立。事件任务完成前是指正在执行的事件任务或正在等待执行的事件任务。

● **运动事件**

控制器 Motion 接口发出 SYNC 同步信号，激活此任务。

⚠ 注意

运动任务默认被配置为优先级 1。可以修改此优先级设置，但必须确保在 Motion 的同步周期内有足够时间执行运行任务。

■ **同步周期设定应满足以下条件：**

➢ 运动任务中定义的程序处理必须有足够的时间执行。

➢ 控制器和伺服驱动器之间必须有足够的时间进行 PDO 和 SDO 数据交换。

如果同步周期的设置不足，这可能会导致受控设备同步丢失和不可预测的操作。同步周期设置的说明请参阅 7.4 同步周期设定方法

- **本机输入点（I0~I7，I10~I17）的上升沿或下降沿**

  当检测到输入点信号上升沿或下降沿时激活此任务。可以通过输入滤波功能设定输入点的响应时间。

- **CANopen 网络的同步信号**

  控制器的 CANopen 接口产生 SYNC 同步信号时，激活此任务。

- **增量型编码器 1**

  控制器的 Encoder 接口检测到第一组编码器 Z 信号上升沿时，激活此任务。

- **增量型编码器 2**

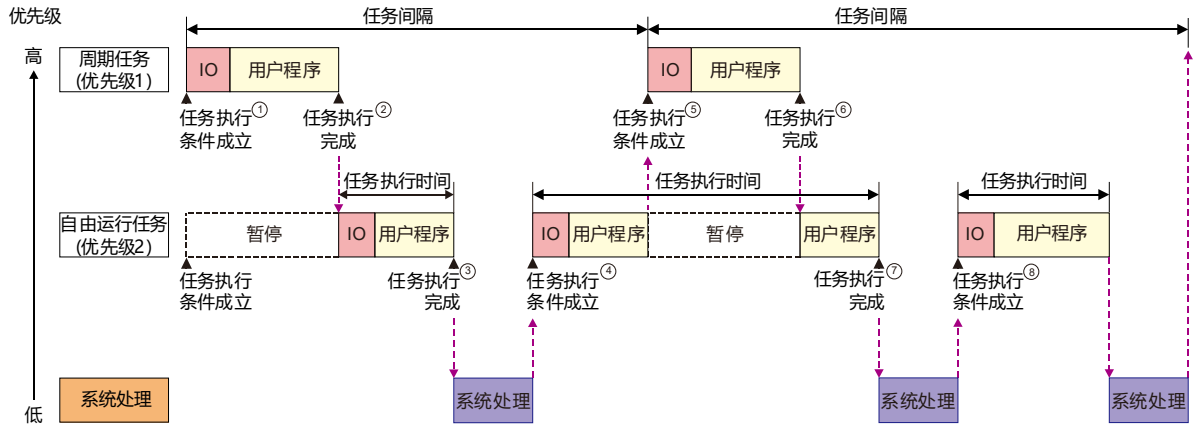
  控制器的 Encoder 接口检测到第二组编码器 Z 信号上升沿时，激活此任务。

7.1.2 任务优先级

控制器无法同时执行多个任务，每个任务须分配一个优先级，各任务按照设定的优先级执行。优先级可以设定为 1 到 24 （1 表示最高优先级， 24 表示最低优先级），每个任务的优先级必须唯一。优先执行优先级较高的任务，高优先级的任务可以中断低优先级的任务。

建议将高实时要求的任务设置为高优先级，将低实时要求的任务设置为低优先级。Motion_CYC 任务默认被配置为优先级 1。

- **下面介绍多个任务同时执行时，任务的执行原理。**
  - **两个任务同时执行时（周期和自由运行）**



① 周期任务和自由运行任务执行条件同时成立，因为周期任务的优先级高，所以先执行周期任务；

② 周期任务执行完成，自由运行任务开始执行；

③ 自由运行任务执行完成，由于没有其它任务请求，控制器执行系统处理；

④ 因为高优先级的周期任务请求还没有到来，所以继续执行自由运行任务；

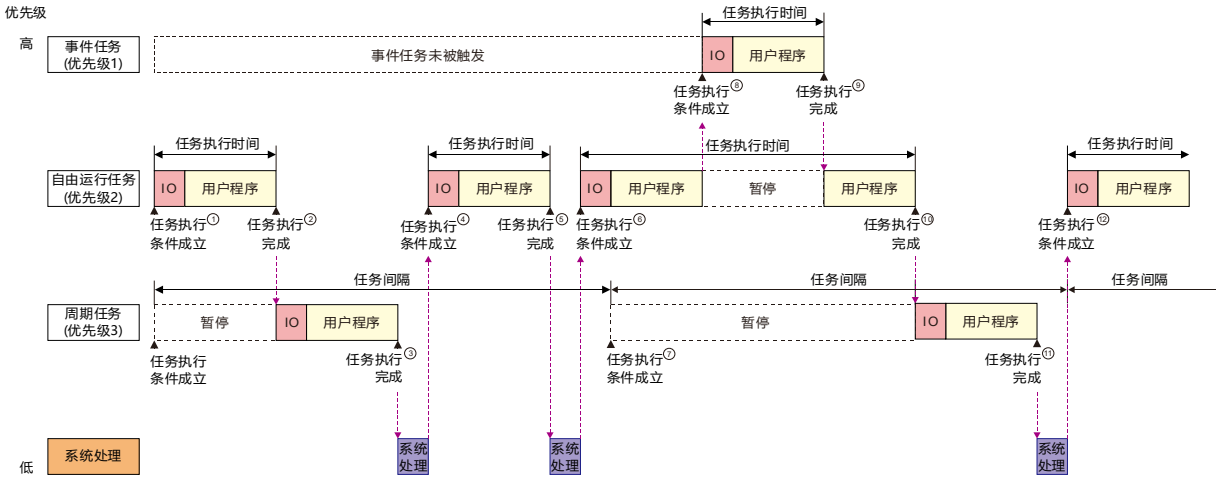
⑤ 自由运行任务执行过程中，高优先级的周期任务请求到来，周期任务打断自由运行任务，控制器执行周期任务；

⑥ 周期任务执行完成，控制器继续执行低优先级的自由运行任务未执行的部分；

⑦ 自由运行任务执行完成，由于没有其它的任务请求，控制器执行系统处理；

⑧ 系统处理完成，因为没有高优先级的周期任务请求，所以继续执行自由运行任务。

■ **三个任务混合执行时（事件、自由运行、周期）**



① 自由运行任务和周期任务执行条件同时成立，因为自由运行任务的优先级高，所以先执行自由运行任务；

② 自由运行任务执行完成，周期任务开始执行；

③ 周期任务执行完成，由于没有其它的任务请求，控制器执行系统处理；

④ 系统处理完成，执行自由运行任务；

⑤ 自由运行任务执行完成，由于没有其它的任务请求，控制器执行系统处理；

⑥ 系统处理完成，执行自由运行任务；

⑦ 周期任务执行条件成立，因为周期任务的优先级低于自由运行任务，所以自由运行任务继续执行，周期任务等待；

⑧ 事件任务执行条件成立，因为事件任务的优先级最高，所以事件任务中断自由运行任务；

⑨ 事件任务执行完毕，控制器继续执行低优先级的自由运行任务未执行的部分；

⑩ 自由运行任务执行完毕，⑦的周期任务请求还未响应，控制器执行周期任务；

⑪ 周期任务执行完毕由于没有其它的任务请求，控制器执行系统处理。

7.1.3 任务看门狗

可以为每个任务配置任务看门狗。当任务的执行时间超过看门狗时间时，控制器会进入 Error 状态，并且停止执行用户程序。

看门狗时间：允许任务执行的最长时间。

7.1.4 每种任务类型可以执行的运动指令

下图展示了不同的任务类型可以执行的运动指令情况，其中“V”表示可以执行，“-”表示不能执行。

种类	指令名称	任务类型			
		周期任务	自由运行	事件任务	
				运动任务	非运动任务
单轴指令	MC_Power	-	-	V	-
	MC_Home	-	-	V	-
	MC_MoveVelocity	-	-	V	-
	MC_Halt	-	-	V	-
	MC_Stop	-	-	V	-
	MC_MoveRelative	-	-	V	-
	MC_MoveAdditive	-	-	V	-
	MC_MoveAbsolute	-	-	V	-
	MC_MoveSuperimposed	-	-	V	-
	MC_Haltsuperimposed	-	-	V	-
	MC_SetPosition	-	-	V	-
	MC_SetOverride	-	-	V	-
	MC_Reset	-	-	V	-
	DMC_SetTorque	-	-	V	-
	MC_ReadAxisError	V	V	V	V
	MC_ReadActualPosition	V	V	V	V
	MC_ReadStatus	V	V	V	V
	MC_ReadMotionState	V	V	V	V
	DMC_ReadParameter_Motion	V	V	V	V
	DMC_WriteParameter_Motion	V	V	V	V
	DMC_TouchProbe	-	-	V	-
	DMC_ChangeMechanismGearRatio	-	-	V	-
	DMC_Jog	-	-	V	-
	DMC_MoveVelocity	-	-	V	-
	DMC_MoveVelocityStopByPos	-	-	V	-
	DMC_MoveVelocityStopByLinePos	-	-	V	-
	DMC_ReadPositionLagStatus	V	V	V	V
	DMC_SwitchSoftLimit	V	V	V	V
	DMC_TorqueControl	V	V	V	V
	DMC_TouchProbeCyclically	-	-	V	-
	DMC_WritePositionLagSetting	V	V	V	V

种类	指令名称	任务类型			
		周期任务	自由运行	事件任务	
				运动任务	非运动任务
多轴指令	MC_GearIn	-	-	V	-
	MC_GearOut	-	-	V	-
	MC_CombineAxes	-	-	V	-
	MC_CamIn	-	-	V	-
	MC_CamOut	-	-	V	-
	DMC_CamAddTappet	V	V	V	V
	DMC_CamDeleteTappet	V	V	V	V
	DMC_CamReadPoint	V	V	V	V
	DMC_CamWritePoint	V	V	V	V
	DMC_CamSet	-	-	V	-
	DMC_CamReadTappetStatus	V	V	V	V
	DMC_CamReadTappetValue	V	V	V	V
	DMC_CamWriteTappetValue	V	V	V	V
应用指令	APF_RotaryCut_Init	-	-	V	-
	APF_RotaryCut_In	-	-	V	-
	APF_RotaryCut_Out	-	-	V	-
G 代码指令	DMC_CartesianCoordinate	-	-	V	-
	DMC_ReadMFunction	V	V	V	V
	DMC_ResetMFunction	V	V	V	V
	DMC_SetG0Para	V	V	V	V
	DMC_SetG1Para	V	V	V	V
	DMC_SetStartPosition	V	V	V	V
轴组指令	DMC_ AddAxisToGroup	-	-	V	-
	DMC_RemoveAxisFromGroup	-	-	V	-
	DMC_UngroupAllAxes	-	-	V	-
	DMC_ GroupEnable	-	-	V	-
	DMC_GroupStop	-	-	V	-
	DMC_GroupInterrupt	-	-	V	-
	DMC_GroupContinue	-	-	V	-
	DMC_MoveDirectAbsolute	-	-	V	-
	DMC_MoveDirectRelative	-	-	V	-
	DMC_MoveLinearAbsolute	-	-	V	-
	DMC_MoveLinearRelative	-	-	V	-
	DMC_MoveCircularAbsolute	-	-	V	-

种类	指令名称	任务类型			
		周期任务	自由运行	事件任务	
				运动任务	非运动任务
	DMC_MoveCircularRelative	-	-	V	-
	DMC_GroupSetOverride	-	-	V	-
	DMC_GroupReadActualPosition	V	V	V	V
通讯指令	DMC_ReadParameter_CANopen	V	V	V	V
	DMC_WriteParameter_CANopen	V	V	V	V
	ETH_Link_Config	V	V	V	V
	ETH_Link_Manage	V	V	V	V
	ETH_Link_Status	V	V	V	V
	ETH_Link_Config_Ext	V	V	V	V
	ETH_ SetServerlinkkeeptime	V	V	V	V
	ETH_Socket_Manage	V	V	V	V
	ETH_Socket_Config	V	V	V	V
	ETH_Socket_Open	V	V	V	V
	ETH_Socket_Send	V	V	V	V
	ETH_Socket_Receive	V	V	V	V
	ETH_Socket_Close	V	V	V	V
	ETH_Socket_Status	V	V	V	V
	RS485_Link_Manage	V	V	V	V
	RS485_Link_Config	V	V	V	V
	RS485_Link_Status	V	V	V	V
	RS485_RS	V	V	V	V
	RS232_Link_Manage	V	V	V	V
	RS232_Link_Config	V	V	V	V
	RS232_Link_Status	V	V	V	V
	RS232_RS	V	V	V	V

7.2 控制器运行或者停止时对变量和装置的影响

当 DVP-15MC 系列运动控制器由 RUN 到 STOP 时,变量和装置保持当前值,当 DVP-15MC 系列运动控制器由 STOP 到 RUN 时,变量及非掉电保持装置的值是否清除有以下两种方式可供选择。

- **变量内容清除**

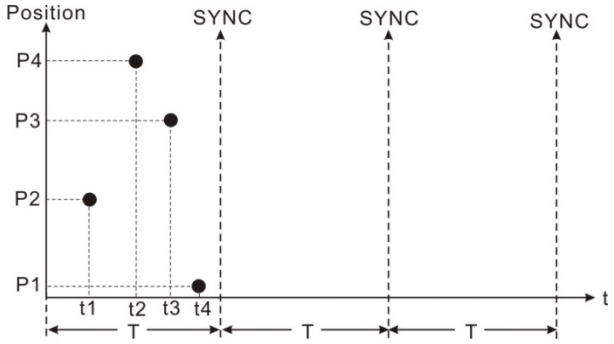
  当 DVP-15MC 系列运动控制器由 STOP 到 RUN 时，清除变量和非掉电保持装置的当前数值，恢复为初始值，如果变量和非掉电保持装置没有初始值，则恢复为默认值 0.

- **变量内容保持**

  当 DVP-15MC 系列运动控制器由 STOP 到 RUN 时，保持变量和装置的当前数值。

7.3 运动程序和运动总线之间的关系

当 DVP-15MC 系列运动控制器接有数台伺服驱动器时，DVP-15MC 系列运动控制器通过同步信号实现同步，同步信号由 DVP-15MC 系列运动控制器以广播的方式发出。伺服驱动器接收到 DVP-15MC 系列运动控制器发出的控制数据，这些控制数据没有马上生效，当同步信号来临时，控制数据同时生效，以此实现多台伺服的同步。如图所示，DVP-15MC 系列运动控制器接有四台伺服驱动器，T 为同步周期。在同步周期内，四台伺服驱动器分别在不同的时刻（t1、t2、t3、t4）接受到控制数据，但是控制数据并没有生效。当伺服驱动器接收到同步信号（SYNC），控制数据同时生效。



7.4 同步周期设定方法

在总线运动控制方式中，同步周期是一个非常重要的参数。因此如果同步周期的设定不当，可能会导致伺服在通讯中出现 AL303/AL302/ AL301 报警，或者一些伺服动作异常现象。

首先我们介绍一下同步周期的构成。每个同步周期开始，会先扫描运动控制程序，扫描结束后将运算所得的控制信息发送给所有轴。因此同步周期可以理解为运动控制程序执行时间 + DVP-15MC 系列运动控制器和所有伺服的通讯时间。

运动控制程序执行时间为运动事件任务中最大程序执行时间（可以双击软件中的任务进行查看），单位为微秒。1000 微秒为 1 毫秒。在实际应用中，建议向上取整。例如最大程序执行时间数值为 2567 微秒=2.5 毫秒，我们可以取 3 毫秒作为程序执行时间。

DVP-15MC 系列运动控制器和伺服通讯的时间可以大概预估为每轴 0.5 毫秒。在时间应用中，建议向上取整，例如 应用中配置 5 个伺服。 通讯时间为 5*0.5 毫秒 = 2.5 毫秒。我们可以取 3 毫秒作为通讯时间。

因此可以得到计算公式： 同步周期时间(毫秒)= 最大程序执行时间数值向上取整(毫秒) + DVP-15MC 系列运动控制器和所有伺服的通讯时间 (毫秒) + 1 (程序变更预留时间) (毫秒)。

为了预防程序变更后，程序的运行时间增大较多，导致之前设定的同步时间不适用，因此在设定的时候尽量做 1 到 2 毫秒的预留时间。

例如：最大程序执行时间为 1634 微秒，应用中共有 5 个伺服。 程序变更预留时间 1 毫秒。

同步周期 = 2 毫秒 (最大程序执行时间 1634 微秒向上取整) + 3 毫秒 (5*0.5 向上取整) + 1 毫秒 (程序预留时间) =6 毫秒。

⚠ 注意

以上方法为大概预估时间，适用于大多数应用。如果对同步时间要求苛刻，应用开发完成后，实际时间可以重新计算，将预留时间拿掉。

8

第8章 逻辑指令

目录

8.1 逻辑指令一览表

指令组	指令	功能	页码
时序输入/输出指令	R_TRIG	上升沿触发	8-11
	F_TRIG	下降沿触发	8-13
	RS	复位	8-15
	SR	置位	8-17
	SEMA	置位	8-19
时序控制指令	JMP	跳转	8-21
数据搬移指令	MOVE	数据搬移	8-22
	MoveBit	位传送	8-24
	TransBit	位传送	8-26
	MoveDigit	数位传送	8-28
	Exchange	数据交换	8-30
	Swap	高低字节交换	8-32
比较指令	LT	小于	8-34
	LE	小于或等于	8-36
	GT	大于	8-38
	GE	大于或等于	8-40
	EQ	等于	8-42
	NE	不等于	8-44
时间相关指令	TON	通电延时定时器	8-46
	TOF	断电延时定时器	8-48
	TP	脉冲型定时器	8-50
	Sys_ReadTime	读取控制器硬件实时时钟时间指令	8-52
	Sys_ReadTotalWorkTime	读取控制器总工作时间指令	8-53
	Sys_ReadPowerOnTime	读取控制器上电时间指令	8-54
计数器指令	CTU	加计数器	8-55
	CTD	减计数器	8-57

指令组	指令	功能	页码
	CTUD	加减计数器	8-59
数学运算指令	ADD	加法指令	8-62
	SUB	减法指令	8-65
	MUL	乘法指令	8-68
	DIV	除法指令	8-71
	MOD	整数取余	8-74
	MODREAL	浮点数取余	8-76
	MODTURNS	计算圈数	8-78
	MODABS	计算相位	8-80
	ABS	取绝对值	8-82
	DegToRad	角度转弧度	8-84
	RadToDeg	弧度转角度	8-86
	SIN	正弦	8-88
	COS	余弦	8-90
	TAN	正切	8-92
	ASIN	反正弦	8-94
	ACOS	反余弦	8-96
	ATAN	反正切	8-98
	LN	自然对数	8-100
	LOG	常用对数	8-102
	SQRT	求平方根	8-104
	EXP	自然指数	8-106
	EXPT	幂指数	8-108
	RAND	随机数	8-110
	TRUNC	浮点数取整数	8-112
	FLOOR	浮点数取整	8-114
	FRACTION	浮点数取小数	8-116
位串指令	AND	逻辑与	8-118
	OR	逻辑或	8-121

指令组	指令	功能	页码
	NOT	取反	8-124
	XOR	异或	8-126
	XORN	同或	8-129
位移指令	SHL	向左移位	8-132
	SHR	向右移位	8-134
	ROL	向左循环移位	8-136
	ROR	向右循环移位	8-138
选择指令	MAX	求最大值	8-140
	MIN	求最小值	8-142
	SEL	二选一	8-144
	MUX	多选一	8-146
	LIMIT	输出限制	8-148
	BAND	死区限制	8-150
	ZONE	输入偏移	8-152
数据类型转换指令	BOOL_TO_***	布尔转换指令群	8-155
	Bit strings_TO_***	位串转换指令群	8-158
	Integers_TO_***	整数转换指令群	8-165
	Real numbers_TO_***	实数转换指令群	8-174
	Times,dates_TO_***	时间，日期转换指令群	8-177
	Strings_TO_***	字符串转换指令群	8-178
通讯指令	DMC_ReadParameter_CANopen	读取参数	8-181
	DMC_WriteParameter_CANopen	写入参数	8-187
	ETH_Link_Config	MODBUS TCP数据交换配置	8-192
	ETH_Link_Manage	开启MODBUS TCP数据交换	8-197
	ETH_Link_Status	MODBUS TCP数据交换状态	8-199
	ETH_Link_Config_Ext	MODBUS TCP数据交换扩展配置	8-207
	ETH_ SetServerlinkkeeptime	设置控制器做为MODBUS TCP从站连接保持时间	8-209
	ETH_Socket_Manage	Socket TCP/UDP管理	8-207

指令组	指令	功能	页码
	ETH_Socket_Config	Socket数据交换配置	8-212
	ETH_Socket_Open	开启Socket	8-215
	ETH_Socket_Send	Socket数据发送	8-217
	ETH_Socket_Receive	Socket 数据接收	8-220
	ETH_Socket_Close	关闭 Socket	8-224
	ETH_Socket_Status	读取 Socket 状态	8-226
	RS485_Link_Manage	RS485通讯管理	8-234
	RS485_Link_Config	RS485通讯参数配置	8-236
	RS485_Link_Status	RS485通讯状态	8-240
	RS485_RS	RS485自由协议	8-246
	RS232_Link_Manage	RS232通讯管理	8-254
	RS232_Link_Config	RS232通讯参数配置	8-256
	RS232_Link_Status	RS232通讯状态	8-260
	RS232_RS	RS232自由协议	8-266
字符串操作指令	CONCAT	连接字符串	8-274
	DELETE	删减字符串	8-276
	INSERT	字符串插入	8-278
	LEFT / RIGHT	左/右截取字符串	8-280
	MID	字符串截取	8-282
	REPLACE	字符替换	8-284
	LEN	计算字符串长度	8-286
	FIND	查找字符串	8-287
IO刷新指令	FROM	模块CR读取数据	8-289
	TO	模块CR写入数据	8-293
	ImmediateInput	输入点立即刷新	8-297
	ImmediateOutput	输出点立即刷新	8-299
PID指令	PID	PID运算	8-301
	GPWM	基本脉冲宽度调变	8-311
取地址	ADR	取地址	8-313

指令组	指令	功能	页码
网络诊断指令	CANmotion_SysDiag	Motion系统诊断指令	8-315
	CANmotion_NodeDiag	Motion轴诊断指令	8-317
	CANopen_SysDiag	CANopen系统诊断指令	8-319
	CANopen_NodeDiag	CANopen从站诊断指令	8-321
	CANopen_State	CANopen主站诊断指令	8-323

8.2 逻辑指令说明

8.2.1 EN 和 ENO 说明

当用户使用的指令带有 EN 和 ENO 时，若 EN 引脚的参数值为 FALSE（0），那么指令定义的功能将不会被执行，指令的输出引脚输出值不会被刷新。相反，若指令定义的 EN 引脚参数值为 TRUE（1），那么指令定义的功能将会被执行，指令输出引脚的值也会被刷新。

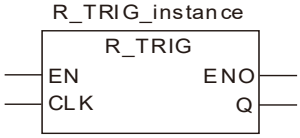
ENO 引脚输出和 EN 引脚的输入保持一致，EN 引脚为 TRUE，ENO 引脚同时变为 TRUE； EN 引脚为 FALSE，ENO 引脚同时变为 FALSE。

当指令为功能块（FB）时，如果功能块（FB）执行后，EN 由 TRUE 变为 FALSE，该功能块（FB）仍继续执行，只是该功能块（FB）输出引脚的值不会被刷新。

8.3 时序输入/输出指令

8.3.1 R_TRIG（上升沿触发指令）

FB/FC	说明	适用机种
FB	本指令用于上升沿触发	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
CLK	输入信号	输入	上升沿触发信号	TRUE或FALSE
Q	输出信号	输出	一个周期的输出	TRUE或FALSE

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLK	●																			
Q	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令功能为CLK引脚处于FALSE到TRUE时，仅一个周期Q输出为TRUE,其余情况是FALSE。

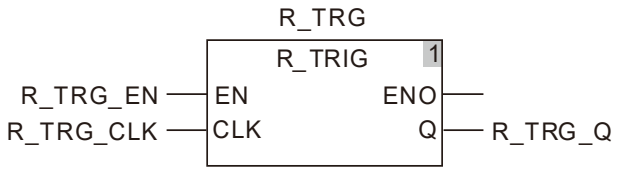
⚠ **注意**

CLK引脚是上升沿信号才能检测到，Q位才会有输出。

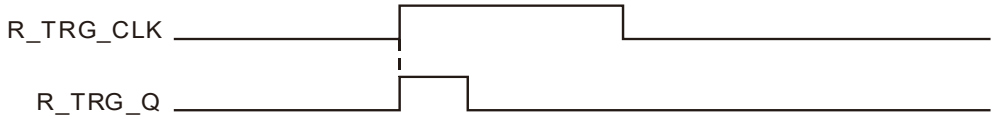
程序范例

■ **变量和程序**

变量名	数据类型	初始值
R_TRG	R_TRIG	
R_TRG_EN	BOOL	FALSE
R_TRG_CLK	BOOL	FALSE
R_TRG_Q	BOOL	

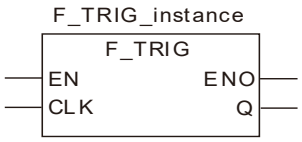


■ **时序图：**



8.3.2 F_TRIG（下降沿触发指令）

FB/FC	说明	适用机种
FB	本指令用于下降沿触发	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
CLK	输入信号	输入	下降沿触发信号	TRUE或FALSE
Q	输出信号	输出	一个周期的输出	TRUE或FALSE

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLK	●																			
Q	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令功能为CLK引脚处于TRUE到FALSE时，仅一个周期Q输出为TRUE,其余情况是FALSE。

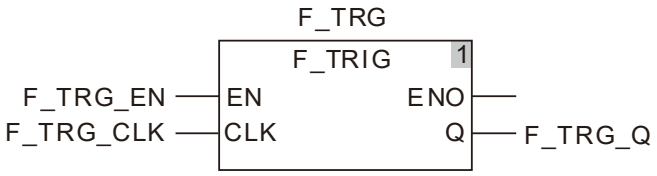
⚠ **注意**

CLK引脚是下降沿信号才能检测到，Q位才会有输出。

程序范例

■ **变量和程序**

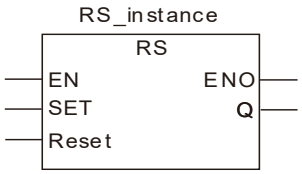
变量名	数据类型	初始值
F_TRG	F_TRIG	
F_TRG_EN	BOOL	FALSE
F_TRG_CLK	BOOL	FALSE
F_TRG_Q	BOOL	



■ **时序图**

8.3.3 RS（复位指令）

FB/FC	说明	适用机种
FB	本指令用于复位优先	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
SET	输入信号	输入	置位信号	TRUE或FALSE
Reset	输入信号	输入	复位信号	TRUE或FALSE
Q	输出信号	输出	输出信号	TRUE或FALSE

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SET	●																			
Reset	●																			
Q	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

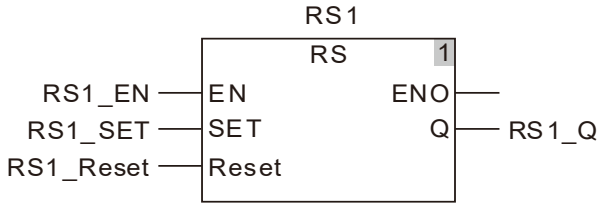
- **功能说明**

本指令功能SET位和Reset位同时为TRUE时，复位优先。

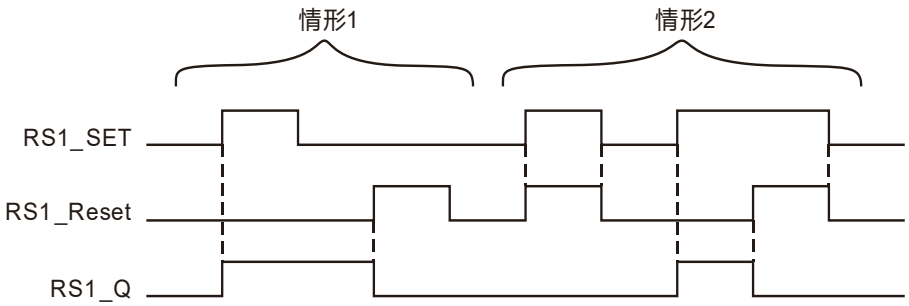
程序范例

■ **变量和程序**

变量名	数据类型	初始值
RS1	RS	
RS1_EN	BOOL	FALSE
RS1_SET	BOOL	FALSE
RS1_Reset	BOOL	FALSE
RS1_Q	BOOL	



■ **时序图**

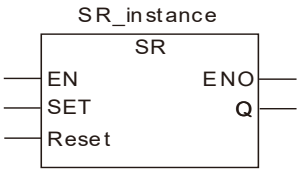


情形1： 当 RS1_SET 为 TRUE 时，输出 RS1_Q 为 TRUE，若 RS1_Reset 为 TRUE，RS1_Q 为 FALSE.

情形2： 当 RS1_Reset 为 TRUE 时，输出 RS1_Q 保持为 FALSE 状态。

8.3.4 SR（置位指令）

FB/FC	说明	适用机种
FB	本指令用于置位优先	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
SET	输入信号	输入	置位信号	TRUE或FALSE
Reset	输入信号	输入	复位信号	TRUE或FALSE
Q	输出信号	输出	输出信号	TRUE或FALSE

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SET	●																			
Reset	●																			
Q	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

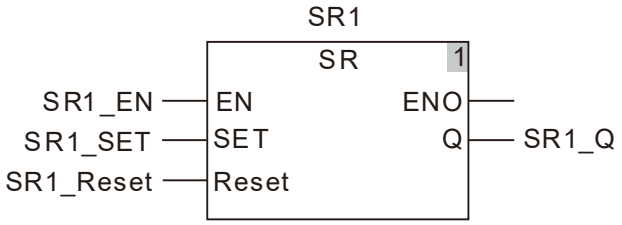
- **功能说明**

本指令功能SET位和Reset位同时为TRUE时，置位优先。

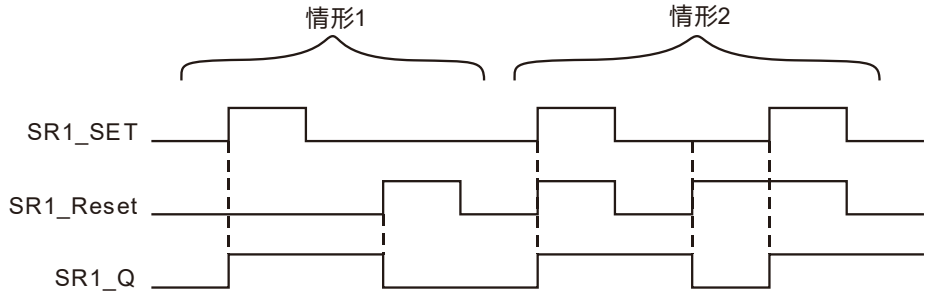
程序范例

■ **变量和程序**

变量名	数据类型	初始值
SR1	SR	
SR1_EN	BOOL	FALSE
SR1_SET	BOOL	FALSE
SR1_Reset	BOOL	FALSE
SR1_Q	BOOL	



■ **时序图**

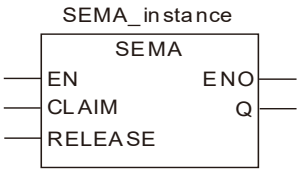


情形1： 当 SR1_SET 为 TRUE 时，SR1_Q 为 TRUE；当 SR1_Reset 为 TRUE 时，SR1_Q 为 FALSE。

情形2： SR1_SET 置位优先 SR1_Reset 复位。

8.3.5 SEMA 指令（置位指令）

FB/FC	说明	适用机种
FB	本指令用于置位优先（第二个周期输出才能有效）	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
CLAIM	输入信号	输入	置位信号	TRUE或FALSE
RELEASE	输入信号	输入	复位信号	TRUE或FALSE
Q	输出信号	输出	输出信号	TRUE或FALSE

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLAIM	●																			
RELE-ASE	●																			
Q	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令功能为CLAIM为TRUE时，则Q为TRUE，RELEASE为TRUE时，则Q为FALSE，CLAIM和RELEASE同时为TRUE，输出Q为TRUE。

⚠ 注意

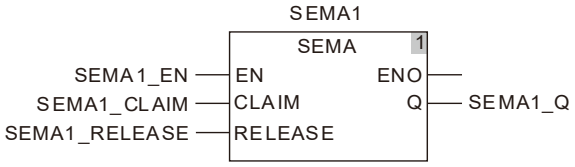
CLAIM为TRUE时，输出Q第二个周期为TRUE。

程序范例

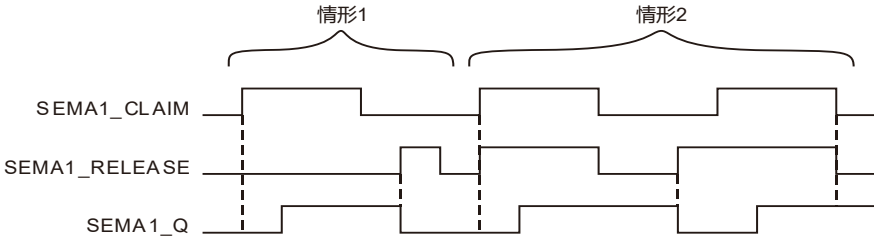
- **变量和程序**

变量名	数据类型	初始值
SEMA1	SEMA	
SEMA1_EN	BOOL	FALSE
SEMA1_CLAIM	BOOL	FALSE

变量名	数据类型	初始值
SEMA1_RELEASE	BOOL	FALSE
SEMA1_Q	BOOL	



■ **时序图**



情形1： 当SEMA1_CLAIM为TRUE时，第二个周期输出SEMA1_Q为TRUE，SEMA1_RELEASE为TRUE时，输出SEMA1_Q立即变为FALSE。

情形2： 当SEMA1_CLAIM为TRUE时，无论SEMA1_RELEASE为何种状态，第二个周期输出SEMA1_Q都为TRUE。

8.4 时序控制指令

8.4.1 JMP（跳转指令）

FB/FC	说明	适用机种
FC	本指令用于跳转至梯形图中由 LABEL 指定任意位置。	DVP15MC11T DVP15MC11T-06

- **功能说明**

本指令用于跳转至梯形图中由LABEL指定的任意位置。

注意

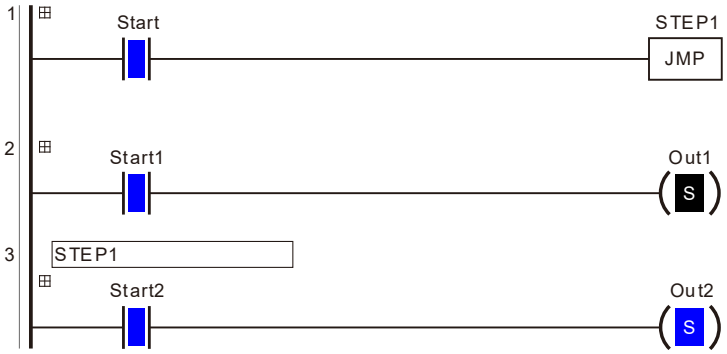
- LABEL为任意字符串。
- 无法将关键字指定为LABEL。
- JMP可以向上跳转。
- 多个JMP指令可以跳转至相同的LABEL。
- JMP指令跳转至梯形图标签位置必须在同一个POU内才有效。

程序范例

- **变量**

变量名	数据类型	当前值
Start	BOOL	TRUE
Start1	BOOL	TRUE
Out1	BOOL	FALSE
Start2	BOOL	TRUE
Out2	BOOL	TRUE

- **程序**

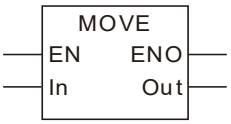


如上示例程序，将 STEP1 作为跳转标签，当 Start 变为 TRUE，JMP 指令跳转至标签 STEP1 位置处执行网络 3 中的程序，Start2 为 TRUE 时，Out2 也为 TRUE，网络 2 不执行。当 Start 为 FALSE 时，网络 1~网络 3 中的程序都执行。

8.5 数据搬移指令

8.5.1 MOVE（数据搬移指令）

FB/FC	说明	适用机种
FC	本指令用于数据搬移	DVP15MC11T DVP15MC11T-06



- 参数

参数名称	含义	输入/输出	描述	参数取值范围
In	输入信号	输入	传送源	由与输入参数所连变量的数据类型决定
Out	输出信号	输出	传送目标	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- 功能说明
  - 将传送源“In”的内容传送至传送目标“Out”内。
  - 本指令支持数组元素中内容传送。

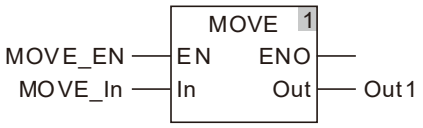
⚠ 注意

Out 的数据类型必须和 In 的数据类型保持一致，否则软件编译报错。

程序范例

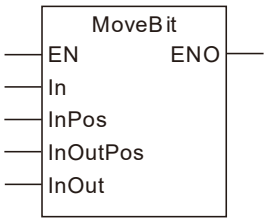
- 变量和程序

变量名	数据类型	当前值
MOVE_EN	BOOL	TRUE
MOVE_In	INT	200
Out1	INT	200


MOVE
1
MOVE_EN
EN
ENO
MOVE_In
In
Out
Out1

8.5.2 MoveBit（位传送指令）

FB/FC	说明	适用机种
FC	本指令用于单个位传送	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	输入信号	输入	传送源	由与输入参数所连变量的数据类型决定
InPos	输入信号	输入	传送源位置	由与输入参数所连变量的数据类型决定
InOutPos	输入信号	输入	传送目标位位置	由与输入参数所连变量的数据类型决定
InOut	输入信号	输入	传送目标	由与输入参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
InPos							●													
InOutPos							●													
InOut		●	●	●	●	●	●	●	●											

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  将传送源“In”的“InPos”位置的一位数据传送至传送目标“InOut”的“InOutPos”位置。

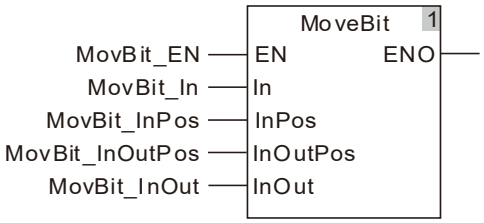
注意

- 本指令没有输出，只有输入。
- InPos的值超过In数据类型有效范围时，不进行传送。
- InOutPos的值超出InOut数据类型有效范围时，也不进行传送。

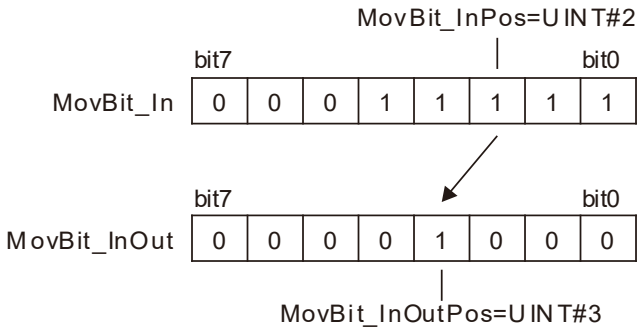
程序范例

■ 变量和程序

变量名	数据类型	当前值
MovBit_EN	BOOL	TRUE
MovBit_In	USINT	31
MovBit_Inpos	UINT	2
MovBit_InOutPos	UINT	3
MovBit_Inout	USINT	8

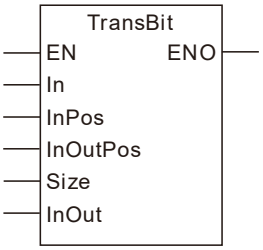


■ 传送示意图



8.5.3 TransBit（位传送指令）

FB/FC	说明	适用机种
FC	本指令用于多个位传送	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	输入信号	输入	传送源	由与输入参数所连变量的数据类型决定
InPos	输入信号	输入	传送源位置	由与输入参数所连变量的数据类型决定
InOutPos	输入信号	输入	传送目标位位置	由与输入参数所连变量的数据类型决定
Size	输入信号	输入	传送位数长度	由与输入参数所连变量的数据类型决定
InOut	输入信号	输入	传送目标	由与输入参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
InPos							●													
InOutPos							●													
Size							●													
InOut		●	●	●	●	●	●	●	●											

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  将传送源“In”的“InPos”位置的“Size”位数据传送至传送目标“InOut”的 “InOutPos”位置。

⚠ **注意**

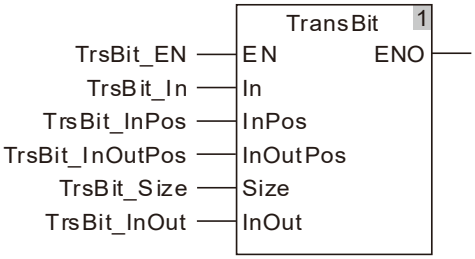
- 本指令没有输出，只有输入。
- Size的值为0时不进行传送。
- InPos的值超过In数据类型有效范围时，不进行传送。

- InOutPos的值超出InOut数据类型有效范围时，也不进行传送。
- Size的值超过有效范围时，不进行传送。

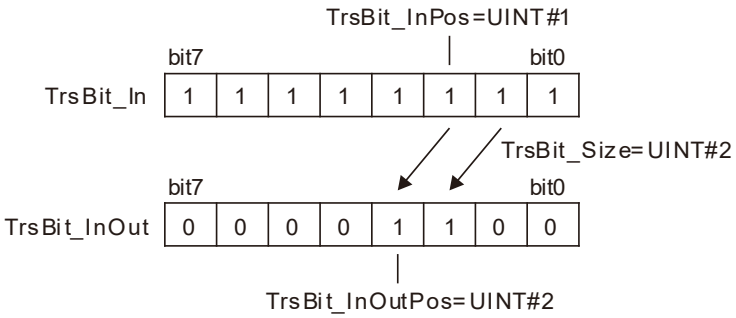
程序范例

- **变量和程序**

变量名	数据类型	当前值
TrsBit_EN	BOOL	TRUE
TrsBit_In	USINT	63
TrsBit_InPos	UINT	1
TrsBit_InOutPos	UINT	2
TrsBit_Size	UINT	2
TrsBit_Inout	USINT	12

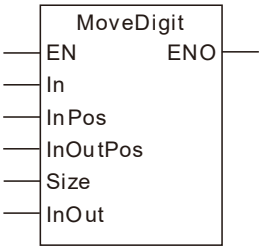


- **传送示意图**



8.5.4 MoveDigit（数位传送指令）

FB/FC	说明	适用机种
FC	本指令用于多个数位传送（一个数位等于 4 个 Bit）。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	输入信号	输入	传送源	由与输入参数所连变量的数据类型决定
InPos	输入信号	输入	传送源位置	由与输入参数所连变量的数据类型决定
InOutPos	输入信号	输入	传送目标数位位置	由与输入参数所连变量的数据类型决定
Size	输入信号	输入	传送数位长度	由与输入参数所连变量的数据类型决定
InOut	输入信号	输入	传送目标	由与输入参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
InPos							●													
InOutPos							●													
Size							●													
InOut		●	●	●	●	●	●	●	●											

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

将传送源“In”的“InPos”数位位置的“Size”数据传送至传送目标“InOut”的“InOutPo”数位位置，指令按位传送。

⚠ 注意

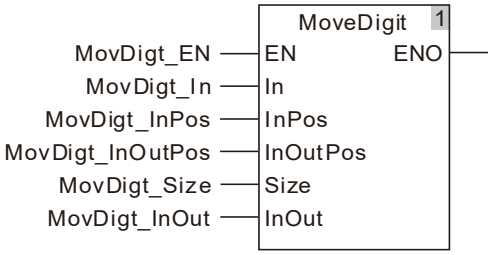
- 本指令没有输出，只有输入。
- Size的值为0时不进行传送。

- InPos的值超过In数据类型有效范围时，不进行传送。
- InOutPos的值超出InOut数据类型有效范围时，不进行传送。
- Size的值超过有效范围时，不进行传送。

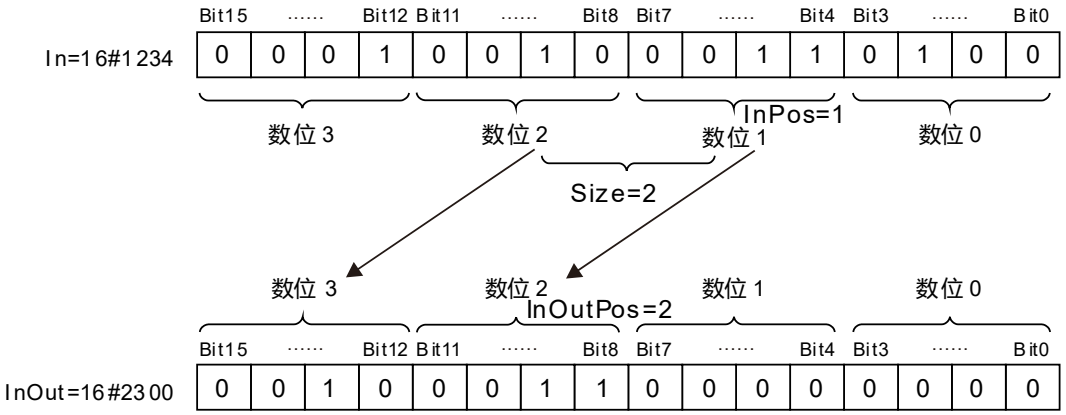
程序范例

- **变量和程序**

变量名	数据类型	当前值
MovDigt_EN	BOOL	TRUE
MovDigt_In	UDINT	16#1234
MovDigt_InPos	UINT	1
MovDigt_InOutPos	UINT	2
MovDigt_Size	UINT	2
MovDigt_Inout	UDINT	16#2300

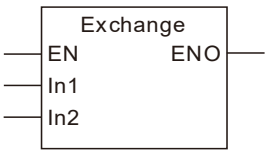


- **传送示意图**



8.5.5 Exchange（数据交换指令）

FB/FC	说明	适用机种
FC	本指令用于数据交换	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	输入信号	输入	交换对象	由与输入参数所连变量的数据类型决定
In2	输入信号	输入	交换对象	由与输入参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
In2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  将In1和In2数据内容交换。

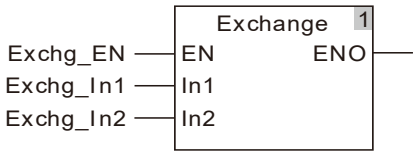
⚠ **注意**

- In1 和 In2 数据类型必须相同。
- 本指令没有输出，只有两个输入。

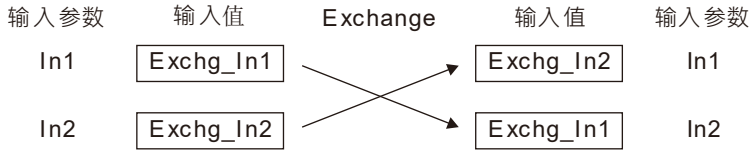
程序范例

- **变量和程序**

变量名	数据类型	当前值
Exchg_EN	BOOL	TRUE
Exchg_In1	INT	30
Exchg_In2	INT	10



■ **交换示意图**

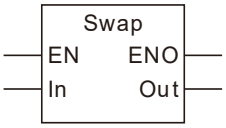


交换In1和In2的值

执行指令，Exchg_In1 和 Exchg_In2 中值一直在交换。

8.5.6 Swap（高低字节交换指令）

FB/FC	说明	适用机种
FC	本指令用于低位与高位字节交换	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	输入信号	输入	交换对象	WORD数据类型范围：0~65535
Out	输入信号	输出	交换结果	WORD数据类型范围：0~65535

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In			●				●													
Out			●				●													

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

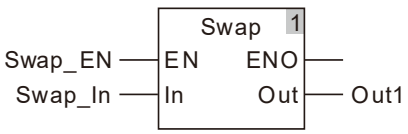
- **功能说明**

  将输入“In”内容低字节与高字节交换，结果放到输出“Out”里。

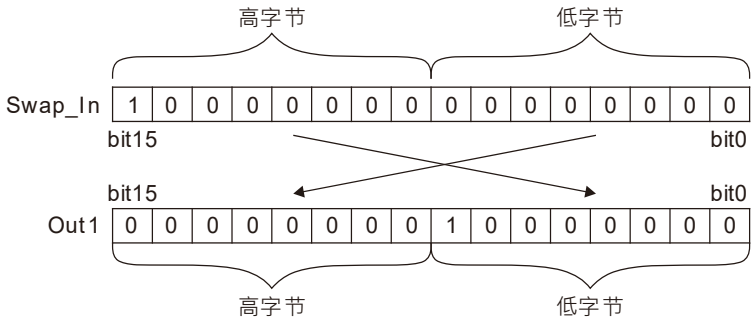
程序范例

- **变量和程序**

变量名	数据类型	当前值
Swap_EN	BOOL	TRUE
Swap_In	UINT	32768
Out1	UINT	128



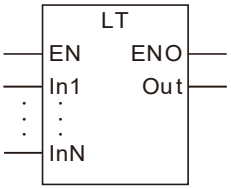
■　**交换示意图**


高字节
低字节
Swap_In
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
bit15
bit0
bit15
bit0
Out1
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
高字节
低字节

8.6 比较指令

8.6.1 LT（小于）

FB/FC	说明	适用机种
FC	本指令用于将两个（或多个）变量或常量作小于比较。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	比较结果	输出	比较结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个（或者多个）变量或常量作小于比较，如果所有输入变量满足*In1<In2<...<InN* ，则*Out*为TRUE，否则*Out*为FALSE。
  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数In1~InN为不同数据类型的变量。

    当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入参数In1~InN均为该数据类型。如In1的数据类型为TIME，则In2~InN的数据类型必须为TIME，否则软件编译时将报错。

注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型必须为BOOL，否则软件编译时将报错。

程序范例

- **变量LT_In1、 LT_In2、 LT_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为BOOL。**

  当LT_In1、 LT_In2、 LT_In3的值分别为-10、50、100， 当变量LT_EN为TRUE时，Out1的值为TRUE（如下表“变量1”所示）；当LT_In1、 LT_In2、 LT_In3的值分别为20、10、100， 当变量LT_EN为TRUE时，Out1的值为FALSE（如下表“变量2”所示）。

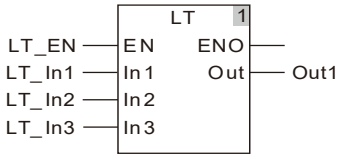
  ➢ **变量 1**

变量名	数据类型	当前值
LT_EN	BOOL	TRUE
LT_In1	INT	-10
LT_In2	UINT	50
LT_In3	DINT	100
Out1	BOOL	TRUE

  ➢ **变量 2**

变量名	数据类型	当前值
LT_EN	BOOL	TRUE
LT_In1	INT	20
LT_In2	UINT	10
LT_In3	DINT	100
Out1	BOOL	FALSE

  ➢ **程序**

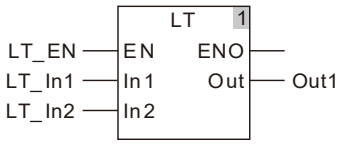


- **变量LT_In1、 LT_In2的数据类型均为TIME，Out1的数据类型为BOOL。**

  当 LT_In1、 LT_In2 的值分别为 T#1ms、T#50ms， 当变量 LT_EN 为 TRUE 时，Out1 的值为 TRUE。

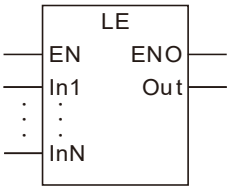
  ➢ **变量和程序**

变量名	数据类型	当前值
LT_EN	BOOL	TRUE
LT_In1	TIME	T#1ms
LT_In2	TIME	T#50ms
Out1	BOOL	TRUE



8.6.2 LE（小于或等于）

FB/FC	说明	适用机种
FC	本指令用于将两个（或多个）变量或常量作小于或等于比较。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	比较结果	输出	比较结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将两个(或者多个)变量或常量作小于或等于比较，如果所有输入变量满足*In1* ≤ *In2*≤ ... ≤ *InN*，则*Out*为TRUE，否则*Out*为FALSE。
  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1~InN*为不同数据类型的变量。

    当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入参数*In1~InN*均为该数据类型。如*In1*的数据类型为TIME，则*In2~InN*的数据类型必须为TIME，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

- 输出变量的数据类型必须为BOOL，否则软件编译时将报错。

程序范例

- **变量LE_In1、 LE_In2、 LE_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为BOOL。**

  当LE_In1、LE_In2、LE_In3的值分别为-10、 50、 50，当变量LE_EN为TRUE时，Out1的值为TRUE（如下表“变量1”所示）；当LE_In1、 LE_In2、 LE_In3的值分别为20、10、100， 当变量LE_EN为TRUE时，Out1的值为FALSE（如下表 “变量2” 所示）。

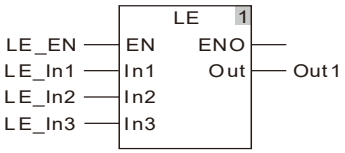
  ➢ **变量 1**

变量名	数据类型	当前值
LE_EN	BOOL	TRUE
LE_In1	INT	-10
LE_In2	UINT	50
LE_In3	DINT	50
Out1	BOOL	TRUE

  ➢ **变量 2**

变量名	数据类型	当前值
LE_EN	BOOL	TRUE
LE_In1	INT	20
LE_In2	UINT	10
LE_In3	DINT	100
Out1	BOOL	FALSE

  ➢ **程序**

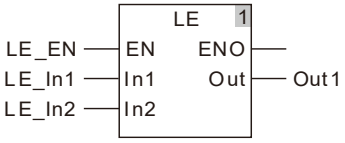


- **变量LE_In1、 LE_In2的数据类型均为TIME，Out1的数据类型为BOOL。**

  当LE_In1、 LE_In2的值分别为T#1ms、T#50ms，当变量LE_EN为TRUE时，Out1的值为TRUE。

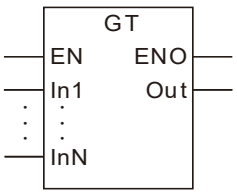
  ➢ **变量和程序**

变量名	数据类型	当前值
LE_EN	BOOL	TRUE
LE_In1	TIME	T#1ms
LE_In2	TIME	T#50ms
Out1	BOOL	TRUE



8.6.3 GT（大于）

FB/FC	说明	适用机种
FC	本指令用于将两个（或多个）变量或常量作大于比较。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	比较结果	输出	比较结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将两个（或者多个）变量或常量作大于比较，如果所有输入变量满足*In1*>*In2*>...>*InN*，则*Out*为TRUE，否则*Out*为FALSE。
  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1~InN*为不同数据类型的变量。

  当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入参数*In1~InN*均为该数据类型。如*In1*的数据类型为TIME，则*In2 ~InN*的数据类型必须为TIME，否则软件编译时将报错。

⚠ **注意**

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型必须为BOOL，否则软件编译时将报错。

程序范例

- **变量GT_In1、 GT_In2、 GT_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为BOOL。**

当GT_In1、 GT_In2、 GT_In3的值分别为100、50、10， 当变量GT_EN为TRUE时，Out1的值为TRUE (如下表“变量1”所示); 当GT_In1、 GT_In2、 GT_In3的值分别为20、10、100，当变量GT_EN为TRUE时，Out1的值为FALSE (如下表 “变量2” 所示)。

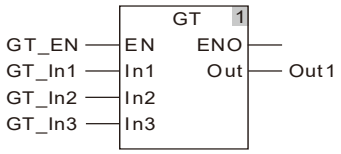
➢ **变量 1**

变量名	数据类型	当前值
GT_EN	BOOL	TRUE
GT_In1	INT	100
GT_In2	UINT	50
GT_In3	DINT	10
Out1	BOOL	TRUE

➢ **变量 2**

变量名	数据类型	当前值
GT_EN	BOOL	TRUE
GT_In1	INT	20
GT_In2	UINT	10
GT_In3	DINT	100
Out1	BOOL	FALSE

➢ **程序**

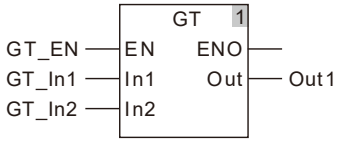


- **变量GT_In1、 GT_In2的数据类型均为TIME，Out1的数据类型为BOOL。**

当GT_In1、 GT_In2的值分别为T#100ms、T#50ms， 当变量GT_EN为TRUE时，Out1的值为TRUE。

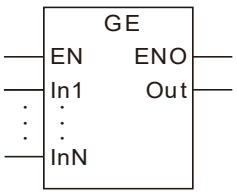
➢ **变量和程序**

变量名	数据类型	当前值
GT_EN	BOOL	TRUE
GT_In1	TIME	T#100ms
GT_In2	TIME	T#50ms
Out1	BOOL	TRUE



8.6.4 GE（大于或等于）

FB/FC	说明	适用机种
FC	本指令用于将两个（或多个）变量或常量作大于或等于比较。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	比较结果	输出	比较结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将两个(或者多个)变量或常量作大于或等于比较，如果所有输入变量满足*In1* ≥ *In2*>...>*InN* 所有*InN*，则*Out*为TRUE，否则*Out*为FALSE。
  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1~InN*为不同数据类型的变量。

  当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入参数*In1~InN*均为该数据类型。如*In1*的数据类型为TIME，则*In2 ~InN*的数据类型必须为TIME，否则软件编译时将报错。

⚠ **注意**

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型必须为BOOL，否则软件编译时将报错。

程序范例

■ **变量GE_In1、 GE_In2、 GE_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为BOOL。**

当GE_In1、 GE_In2、 GE_In3的值分别为100、50、50， 当变量GE_EN为TRUE时，Out1的值为TRUE (如下表“变量1”所示) ; 当GE_In1、 GE_In2、 GE_In3的值分别为10、10、100，当变量GE_EN为TRUE时，Out1的值为FALSE (如下表 “变量2” 所示)。

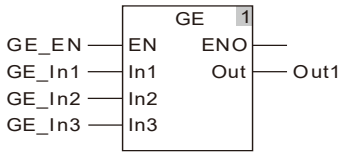
➢ **变量 1**

变量名	数据类型	当前值
GE_EN	BOOL	TRUE
GE_In1	INT	100
GE_In2	UINT	50
GE_In3	DINT	50
Out1	BOOL	TRUE

➢ **变量 2**

变量名	数据类型	当前值
GE_EN	BOOL	TRUE
GE_In1	INT	10
GE_In2	UINT	10
GE_In3	DINT	100
Out1	BOOL	FALSE

➢ **程序**

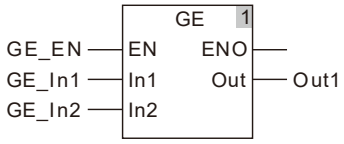


■ **变量GE_In1、 GE_In2的数据类型均为TIME，Out1的数据类型为BOOL。**

当GE_In1、 GE_In2的值分别为T#100ms、T#50ms，当变量GE_EN为TRUE时，Out1的值为TRUE。

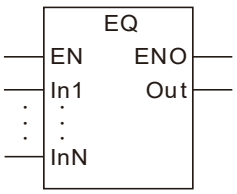
➢ **变量和程序**

变量名	数据类型	当前值
GE_EN	BOOL	TRUE
GE_In1	TIME	T#100ms
GE_In2	TIME	T#50ms
Out1	BOOL	TRUE



8.6.5 EQ（等于）

FB/FC	说明	适用机种
FC	本指令用于将两个（或多个）变量或常量作等于比较。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	比较结果	输出	比较结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将两个（或者多个）变量或常量作等于比较，如果所有输入变量满足*In1 = In2 = ... = InN*，则*Out*为TRUE，否则*Out*为FALSE。
  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1~InN*为不同数据类型的变量。

  当输入变量的数据类型为BOOL、TIME、DATE 、TOD 、STRING中的一种时，要求输入参数*In1~InN*均为该数据类型。如*In1*的数据类型为TIME，则*In2~InN*的数据类型必须为TIME，否则软件编译时将报错。

⚠ **注意**

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型必须为BOOL，否则软件编译时将报错。

程序范例

■ **变量EQ_In1、EQ_In2、EQ_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为BOOL。**

当EQ_In1、EQ_In2、EQ_In3的值分别为50、50、50，当变量EQ_EN为TRUE时，Out1的值为TRUE（如下表“变量1”所示）；当EQ_In1、EQ_In2、EQ_In3的值分别为10、50、100，当变量EQ_EN为TRUE时，Out1的值为FALSE（如下表“变量2”所示）。

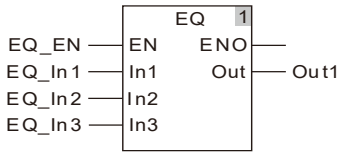
➢ **变量 1**

变量名	数据类型	当前值
EQ_EN	BOOL	TRUE
EQ_In1	INT	50
EQ_In2	UINT	50
EQ_In3	DINT	50
Out1	BOOL	TRUE

➢ **变量 2**

变量名	数据类型	当前值
EQ_EN	BOOL	TRUE
EQ_In1	INT	10
EQ_In2	UINT	50
EQ_In3	DINT	100
Out1	BOOL	FALSE

➢ **程序**

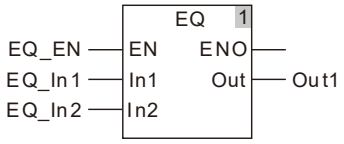


■ **变量EQ_In1、EQ_In2的数据类型均为TIME，Out1的数据类型为BOOL。**

当EQ_In1、EQ_In2的值分别为T#50ms、T#50ms，当变量EQ_EN为TRUE时，Out1的值为TRUE。

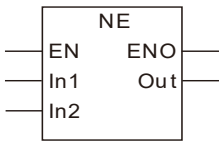
➢ **变量和程序**

变量名	数据类型	当前值
EQ_EN	BOOL	TRUE
EQ_In1	TIME	T#50ms
EQ_In2	TIME	T#50ms
Out1	BOOL	TRUE



8.6.6 NE（不等于）

FB/FC	说明	适用机种
FC	本指令用于将两个变量或常量作不等于比较。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	比较数	输入	比较数	由与输入参数所连变量的数据类型决定
In2	比较数	输入	比较数	由与输入参数所连变量的数据类型决定
Out	比较结果	输出	比较结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 和 In2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个变量或常量作不等于比较，如果输入变量满足*In1*≠*In2*，则*Out*为TRUE，否则*Out*为FALSE。
  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1*和*In2*为不同数据类型的变量。

  当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入参数*In1*和*In2*均为该数据类型。如*In1*的数据类型为TIME，则*In2*的数据类型必须为TIME，否则软件编译时将报错。

⚠ **注意**

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型必须为BOOL，否则软件编译时将报错。

程序范例

- **变量NE_In1、NE_In2的数据类型分别为INT、DINT，Out1的数据类型为BOOL。**

  当NE_In1、NE_In2的值分别为100、50，变量NE_EN为TRUE时，Out1的值为TRUE (如下表“变量1”所示) ; 当NE_In1、NE_In2的值分别为100、100，变量NE_EN为TRUE时，Out1的值为FALSE (如下表“变量2”所示)。

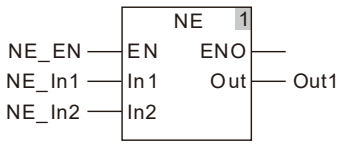
  - **变量 1**

变量名	数据类型	当前值
NE_EN	BOOL	TRUE
NE_In1	INT	100
NE_In2	UINT	50
Out1	BOOL	TRUE

  - **变量 2**

变量名	数据类型	当前值
NE_EN	BOOL	TRUE
NE_In1	INT	100
NE_In2	UINT	100
Out1	BOOL	FALSE

  - **程序**

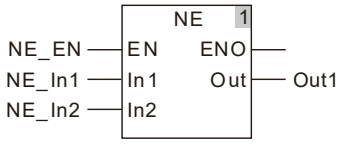


- **变量NE_In1、NE_In2的数据类型均为TIME，Out1的数据类型为BOOL。**

  当NE_In1、NE_In2的值分别为T#10ms、T#50ms，变量NE_EN为TRUE时，Out1的值为TRUE。

  - **变量和程序**

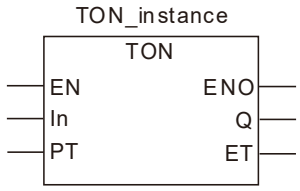
变量名	数据类型	当前值
NE_EN	BOOL	TRUE
NE_In1	TIME	T#10ms
NE_In2	TIME	T#50ms
Out1	BOOL	TRUE



8.7 时间相关指令

8.7.1 TON（通电延时定时器）

FB/FC	说明	适用机种
FB	本指令用于延时通电。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	定时器输入	输入	用于控制定时器计时或复位	TRUE或FALSE
PT	定时时长	输入	设定定时器的定时时间	◆
Q	定时输出	输出	定时器定时到达时，输出为TRUE	TRUE或FALSE
ET	计时时间	输出	从定时器开始计时到当前时刻的时间	◆

◆ T#0ns ~ 213503d23h34m33s709ms551us615ns

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
PT																●				
Q	●																			
ET																●				

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令为定时器功能，用于通电延时。
  - 当输入参数In为TRUE时，定时器开始计时，输出参数ET（计时时间）的值随之增加；当计时时间等于PT（定时时长）时，输出参数Q为TRUE；当输入参数In被设置为FALSE时，计时停止，输出参数Q和ET均复位。

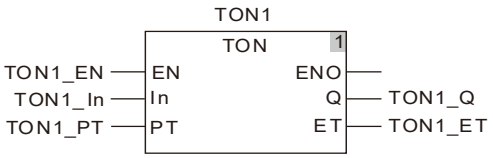
注意

输出参数ET在计时到达后停止计数，在输入参数In由TRUE变为FALSE时复位为0（0ns）。

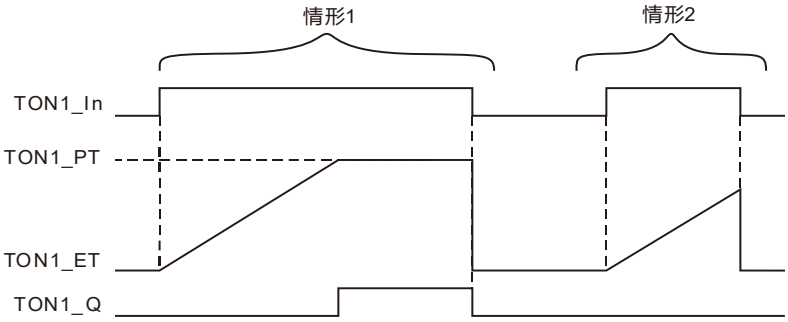
程序范例

■ **变量和程序**

变量名	数据类型	初始值
TON1	TON	
TON1_EN	BOOL	FALSE
TON1_In	BOOL	FALSE
TON1_PT	TIME	
TON1_Q	BOOL	
TON1_ET	TIME	



■ **时序图**

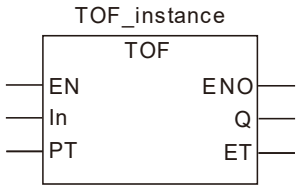


情形1： TON1_PT设定定时时间，当输入TON1_In为TRUE时，定时器开始计时，当前计时时间（TON1_ET）等于设定时间(TON1_PT)，TON1_Q为TRUE，定时器停止计时，TON1_In位复位，TON1_ET位和TON1_Q均复位。

情形2： 当定时器当前计时时间（TON1_ET）小于设定时间（TON1_PT），使In复位，TON1_ET复位，TON1_Q状态保持。

8.7.2 TOF（断电延时定时器）

FB/FC	说明	适用机种
FB	本指令用于延时断电。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	定时器输入	输入	用于控制定时器计时或复位	TRUE或FALSE
PT	定时时长	输入	设定定时器的定时时间	◆
Q	定时器输出	输出	定时器定时到达时，输出为FALSE	TRUE或FALSE
ET	计时时间	输出	从定时器开始计时到当前时刻的时间	◆

◆ T#0ns ~ 213503d23h34m33s709ms551us615ns

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
PT																●				
Q	●																			
ET																●				

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令为定时器功能，用于延时断电。
  - 当输入参数In为TRUE时，定时器输出Q为TRUE；当输入参数In由TRUE变为FALSE时，定时器开始计时，输出参数ET（计时时间）的值随之增加，此时输出Q仍然保持为TRUE；当计时时间等于PT（定时时长）时，输出参数Q为FALSE，并且定时器停止计时；当输入参数In被设置为TRUE时，参数ET复位，同时输出参数Q再次被设置为TRUE。

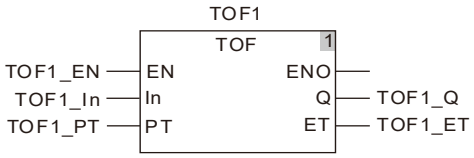
⚠ **注意**

输出参数ET在计时到达后停止计时，在输入参数In由FALSE变为TRUE时复位为0（0ns）。

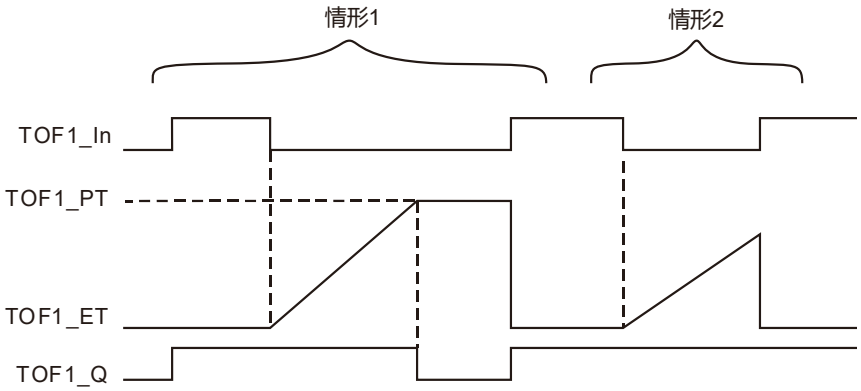
程序范例

■ **变量和程序**

变量名	数据类型	初始值
TOF1	TOF	
TOF1_EN	BOOL	FALSE
TOF1_In	BOOL	FALSE
TOF1_PT	TIME	
TOF1_Q	BOOL	
TOF1_ET	TIME	



■ **时序图**

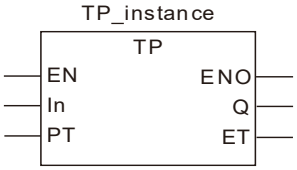


情形1： TOF1_PT为设定断电延迟时间，TOF1_In为TRUE，TOF1_Q为TRUE，TOF1_In为FALSE时，定时器开始计时，当前计时时间 (TOF1_ET) 等于设定时间 (TOF1_PT) 时，TOF1_Q为FALSE，定时器停止计时。

情形2： TOF1_In由TRUE到FALSE时，定时器开始计时，当前计时时间 (TOF1_ET) 小于设定时间 (TOF1_PT) 时，使TOF1_In位TRUE，TOF1_ET位复位，TOF1_Q位状态保持。

8.7.3 TP（脉冲型定时器）

FB/FC	说明	适用机种
FB	本指令用于输入参数 In 为 TRUE 后延时断电。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	定时器输入	输入	用于控制定时器计时或复位	TRUE或FALSE
PT	定时时长	输入	设定定时器的定时时间	◆
Q	定时器输出	输出	定时器定时到达时，输出为FALSE	TRUE或FALSE
ET	计时时间	输出	从定时器开始计时到当前时刻的时间	◆

◆ T#0ns ~ 213503d23h34m33s709ms551us615ns

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
PT																●				
Q	●																			
ET																●				

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

当输入参数In为TRUE时，定时器输出Q为TRUE，同时定时器开始计时，输出参数ET（计时时间）的值随之增加，此时输出Q仍然保持为TRUE；当计时时间等于PT（定时时长）时，输出参数Q为FALSE，并且定时器停止计时；当输入参数In由TRUE变为FALSE时，参数ET复位。

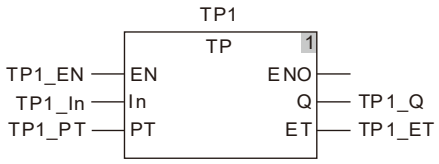
⚠ **注意**

输出参数ET在计时到达后停止计时，在输入参数In由TRUE变为FALSE时复位为0（0ns）。

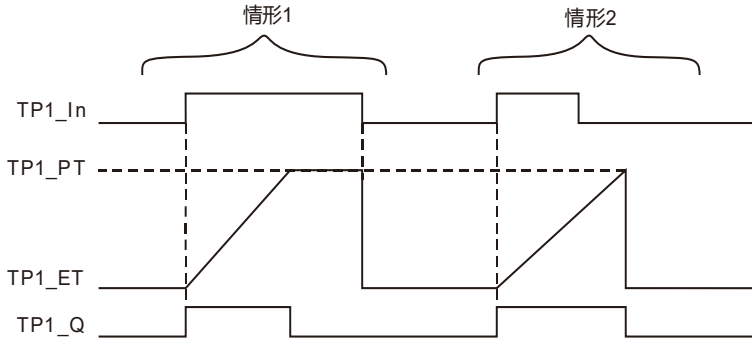
程序范例

■ **变量和程序**

变量名	数据类型	初始值
TP1	TP	
TP1_EN	BOOL	FALSE
TP1_In	BOOL	FALSE
TP1_PT	TIME	
TP1_Q	BOOL	
TP1_ET	TIME	



■ **时序图**

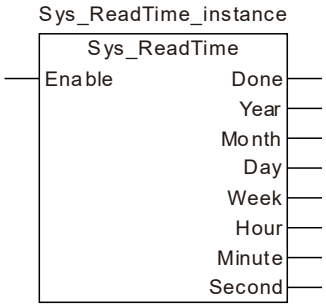


情形1： TP1_PT 设定定时时间，当 TP1_In 为 TRUE，定时器开始计时，TP1_Q 为 TRUE，当前计时时间(TP1_ET)到达设定时间(TP1_PT)，TP1_Q 为 FALSE；当 TP1_In 为 FALSE，TP1_ET 复位。

情形2： TP1_PT 设定定时时间，当 TP1_In 为 TRUE，定时器开始计时，TP1_Q 为 TRUE，当 TP1_In 为 FALSE 时，当前计时时间（TP1_ET）未到达设定时间（TP1_PT），TP1_ET 继续计时，TP1_Q 保持状态不变，当前计时时间（TP1_ET）到达设定时间 （TP1_PT），TP1_ET 和 TP1_Q 均复位。

8.7.4 Sys_ReadTime（读取控制器硬件实时时钟时间指令）

FB/FC	说明	适用机种
FB	此指令用于读取控制器硬件实时时钟时间。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	Enable 为 TRUE 时

- **输出参数**

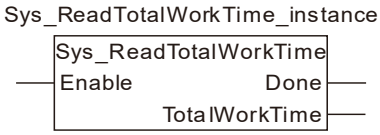
名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Year （年）	年	UINT	1970~2106
Month （月）	月	UINT	1~12
Day （日）	日	UINT	1~ 31
Week （周）	星期	UINT	1~7
Hour （时）	时	UINT	1~24
Minute （分）	分	UINT	1~60
Second （秒）	秒	UINT	0~60

- **功能说明**

此指令用于读取控制器硬件实时时钟的时间。当 Enable 为 TRUE 时，控制器硬件实时时钟的年、月、日、星期、时、分、秒等时间信息读取至指定的变量内。

8.7.5 Sys_ReadTotalWorkTime（读取控制器总工作时间指令）

FB/FC	说明	适用机种
FB	此指令用于读取控制器总工作时间。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	Enable 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
TotalWorkTime （系统时间）	该输出参数显示控制器总工作时间。	TIME	

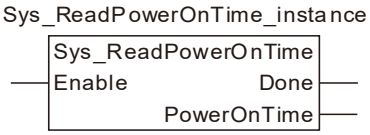
  - ◆ T#0ns ~ 213503d23h34m33s709ms551us615ns

- **功能说明**

此指令用于读取控制器累计工作时间，当 Enable 为 TRUE 时，控制器累计工作时间读取至 TotalWorkTime 指定的变量内。如控制器昨天工作 3 个小时，今天工作 2 个小时，则读取到控制器累计工作时间为 5 个小时。

8.7.6 Sys_ReadPowerOnTime（读取控制器上电时间指令）

FB/FC	说明	适用机种
FB	此指令用于读取控制器上电总时间。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	Enable 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
PowerOnTime （上电时间）	该输出参数显示控制器上电总时间。	TIME	

  - ◆ T#0ns ~ 213503d23h34m33s709ms551us615ns

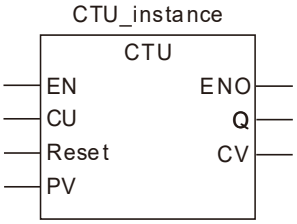
- **功能说明**

此指令用于读取控制器总上电时间，当控制器断电后重新上电，该指令重新开始计时。当 Enable 为 TRUE 时，控制器累计上电时间读取至 PowerOnTime 指定的变量内。

8.8 计数器指令

8.8.1 CTU（加计数器）

FB/FC	说明	适用机种
FB	本指令用于加计数	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
CU	加计数信号	输入	用于控制计数器加计数	TRUE或FALSE
Reset	复位信号	输入	复位计数器当前计数值	TRUE或FALSE
PV	设定值	输入	计数器设定值	0 ~ 4294967295
Q	输出信号	输出	计数器计数到达时，输出为TRUE	TRUE或FALSE
CV	计数值	输出	计数器当前加计数值	0 ~ 4294967295

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	●																			
Reset	●																			
PV								●												
Q	●																			
CV								●												

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令功能为加计数。
  - 当输入参数CU由FALSE变为TRUE时，计数器加计数一次，输出参数CV（当前计数值）的值加1；当CV值等于PV（设定计数值）值时，输出参数Q为TRUE；当输入参数Reset位被设置为TRUE时，CV值清零，输出参数Q复位。

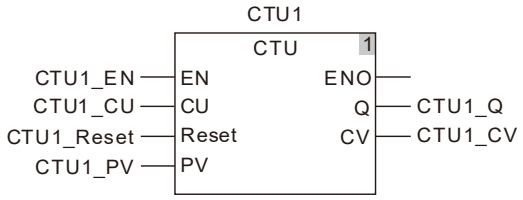
注意

- 当Reset在TRUE的情况下，计数器不会计数。
- 当CV计数值等于PV设定值时，计数器停止计数。

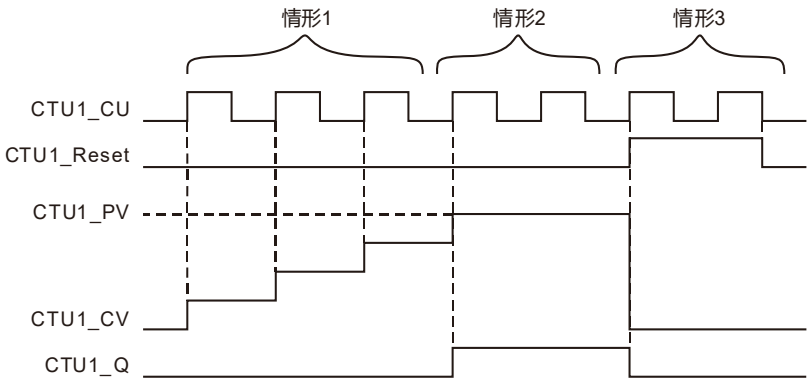
程序范例

- **变量和程序**

变量名	数据类型	初始值
CTU1	CTU	
CTU1_EN	BOOL	FALSE
CTU1_CU	BOOL	FALSE
CTU1_Reset	BOOL	FALSE
CTU1_PV	UDINT	4
CTU1_Q	BOOL	
CTU1_CV	UDINT	



- **时序图**



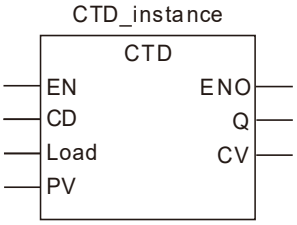
情形1： CTU 正常加计数计数功能，CTU1_CU 每触发一次，CTU1_CV 加 1。

情形2： 当 CTU1_CV 计数值到达 CTU1_PV 值时，同时 CTU1_Q 为 TRUE，CTU 停止计数。

情形3： 当 CTU1_Reset 为 TRUE 时，同时 CTU1_CV 值清零，CTU1_Q 为 FALSE，触发 CTU1_CU 时不会计数。。

8.8.2 CTD（减计数器）

FB/FC	说明	适用机种
FB	本指令用于减计数	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
CD	减计数信号	输入	用于控制计数器减计数	TRUE或FALSE
Load	写入信号	输入	写入计数器减计数值	TRUE或FALSE
PV	设定值	输入	计数器设定值	0 ~ 4294967295
Q	输出信号	输出	计数器减计数到0时，输出为TRUE	TRUE或FALSE
CV	计数值	输出	计数器当前减计数值	0 ~ 4294967295

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	●																			
Load	●																			
PV								●												
Q	●																			
CV								●												

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令功能为减计数。
  - Load先置位再复位，将PV设定值写入到CV中，当输入参数CD位由FALSE变为TRUE时，计数器减计数一次，输出参数CV（当前计数值）的值减1；当CV值等于0时，输出参数Q为TRUE。

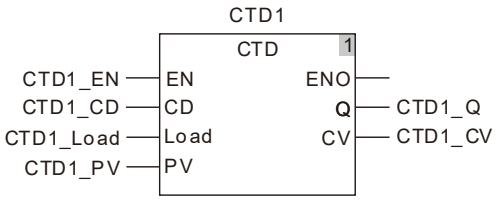
⚠ **注意**

Load在TRUE的情况下，计数器不会减计数。

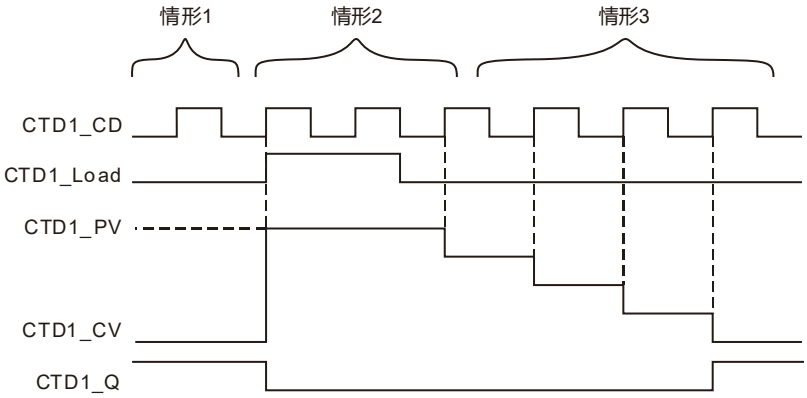
程序范例

■ **变量和程序**

变量名	数据类型	初始值
CTD1	CTD	
CTD1_EN	BOOL	FALSE
CTD1_CD	BOOL	FALSE
CTD1_Load	BOOL	FALSE
CTD1_PV	UDINT	4
CTD1_Q	BOOL	
CTD1_CV	UDINT	



■ **时序图**



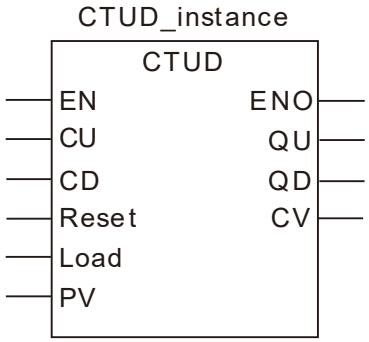
情形1： 当 CTD1_CV 值是 0 的情况下，触发 CTD1_CD 对输出无影响。

情形2： 当 CTD1_Load 为 TRUE，同时 CTD1_CV 值等于 CTD1_PV 设定值，CTD1_Q 由 TRUE 变成 FALSE，触发 CTD1_CD，CTD1_CV 是不会减计数。

情形3： CTD 指令正常减计数功能，当 CTD1_Load 为 FALSE 时，每触发 CTD1_CD 一次，CTD1_CV 值减 1，当 CTD1_CV 减至 0 时，CTD1_Q 为 TRUE。

8.8.3 CTUD（加减计数器）

FB/FC	说明	适用机种
FB	本指令用于加减计数	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
CU	加计数信号	输入	用于控制计数器加计数	TRUE或FALSE
CD	减计数信号	输入	用于控制计数器减计数	TRUE或FALSE
Reset	复位信号	输入	复位计数器当前计数值	TRUE或FALSE
Load	写入信号	输入	写入计数器减计数值	TRUE或FALSE
PV	设定值	输入	计数器设定值	0 ~ 4294967295
QU	输出信号	输出	计数器计数到达时，输出为TRUE	TRUE或FALSE
QD	输出信号	输出	计数器减计数到0时，输出为TRUE	TRUE或FALSE
CV	计数值	输出	计数器当前计数值	0 ~ 4294967295

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	●																			
CD	●																			
Reset	●																			
Load	●																			
PV								●												
QU	●																			
QD	●																			

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CV								●												

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令有加计数器和减计数器的功能，既可以加计数又可以减计数。

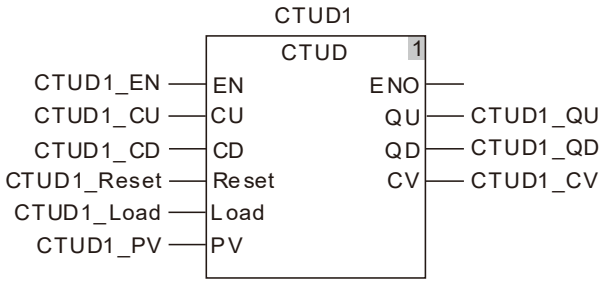
⚠ 注意

- Load在TRUE的情况下，计数器不会减计数。
- Reset在TRUE的情况下，计数器不会加计数。

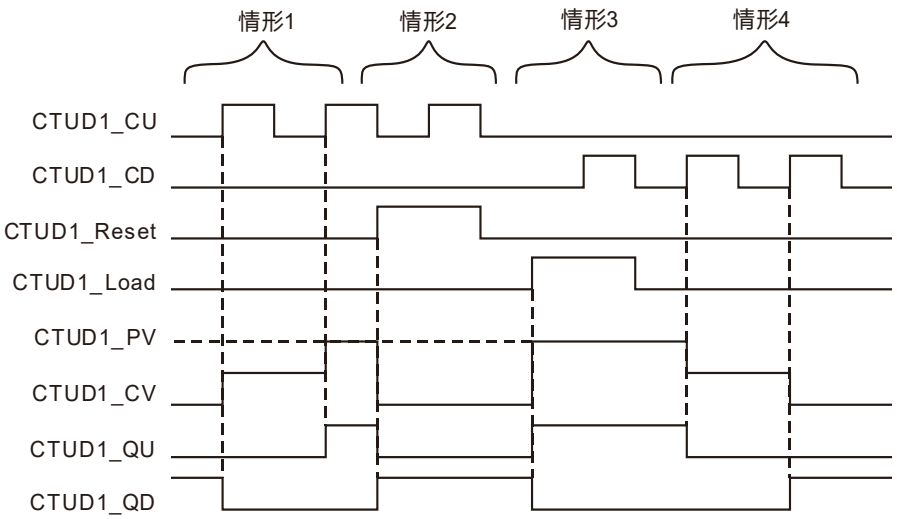
程序范例

- **变量和程序**

变量名	数据类型	初始值
CTUD1	CTUD	
CTUD1_EN	BOOL	FALSE
CTUD1_CU	BOOL	FALSE
CTUD1_CD	BOOL	FALSE
CTUD1_Reset	BOOL	FALSE
CTUD1_Load	BOOL	FALSE
CTUD1_PV	UDINT	4
CTUD1_QU	BOOL	
CTUD1_QD	BOOL	
CTUD1_CV	UDINT	



■ **时序图**



情形1： 指令正常加计数功能，CTUD1_CU每触发一次，CTUD1_CV加1。

情形2： 当CTUD1_Reset为TRUE时，CTUD1_CV值清零，CTUD1_QU变为FALSE，CTUD1_QD变为TRUE。

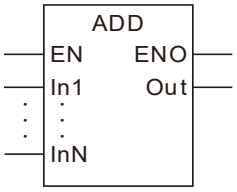
情形3： 当CTUD1_Load为TRUE， CTUD1_CV值等于CTUD1_PV设定值时，CTUD1_QU变为TRUE，CTUD1_QD变为FALSE，此时触发CTUD1_CD，指令不能进行减计数。

情形4： 指令正常减计数功能，当CTUD1_CD为TRUE时，CTUD1_QU为FALSE，CTUD1_CD每触发一次，CTUD1_CV减1，当CTUD1_CV减至0时，CTUD1_QD为TRUE。

8.9 数学运算指令

8.9.1 ADD（加法指令）

FB/FC	说明	适用机种
FC	本指令用于将两个（或者多个）变量或常量相加。	DVP15MC11T DVP15MC11T-06



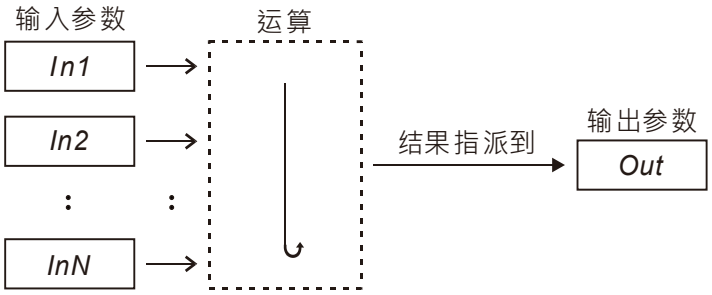
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	被加数	输入	被加数	由与输入参数所连变量的数据类型决定
In2 至 InN	加数	输入	程序编写时，可通过编程软件增加或减少加数，但加数最多为7个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	和	输出	In1 ~ InN的代数和	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	
In2 至 InN		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个（或者多个）变量或常量相加，结果输出至“Out”，即Out = In1 + In2 +…+ InN。
  - 对于位串、整数和实数类型，本指令允许输入参数“In1”~“InN”为不同数据类型的变量。当“In1”~“InN”为不同数据类型的变量时，以能包含“In1”~“InN”所有取值范围的数据类型进行运算。例如“In1”的数据类型为INT，“In2”的数据类型为DINT，则“Out”的数据类型为DINT。



- 对于位串、整数和实数类型，本指令允许输入和输出变量为不同数据类型。当输入和输出变量为不同数据类型时，输出变量数据类型的范围须包含所有输入变量数据类型的范围，否则，软件编译时会报错。例如："In1" 和 "In2" 的数据类型分别为INT和DINT，则 "Out" 的数据类型为DINT，此时若 "Out" 所连变量的数据类型为INT，则软件编译报错；若 "Out" 所连变量的数据类型为LINT，则软件编译正确。
- 对于时间和日期类型，仅支持以下几种组合：
  1. In1为TIME类型，In2为TIME类型，Out为TIME类型。
  2. In1为TOD（TIME_OF_DAY）类型，In2为TIME类型，Out为TOD类型。
  3. In1为DT（DAY_AND_TIME）类型，In2为TIME类型，Out为DT类型。

⚠ **注意**

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- "In1" ~ "InN" 之和可能超出 "Out" 数据类型的有效范围。

  例如："ADD_In1" 与 "ADD_In2" 的数据类型均为INT，值分别为32767和1，如果输出变量数据类型为INT时，输出变量的值为 - 32768（如下表 "变量1" 所示）；如果输出变量的数据类型设置为DINT，则输出变量的值为32768（如下表 "变量2" 所示）。

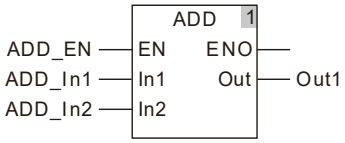
➢ **变量 1**

变量名	数据类型	当前值
ADD_EN	BOOL	TRUE
ADD_In1	INT	32767
ADD_In2	INT	1
Out1	INT	- 32768

➢ **变量 2**

变量名	数据类型	当前值
ADD_EN	BOOL	TRUE
ADD_In1	INT	32767
ADD_In2	INT	1
Out1	DINT	32768

➢ **程序**



程序范例

- 变量ADD_In1、 ADD_In2、Out1的数据类型均为INT，ADD_In1与ADD_In2的值分别为10和50，当变量ADD_EN为TRUE时，Out1的值为60（如下表“变量1”所示）。
- 变量ADD_In1、 ADD_In2、Out1的数据类型均为TIME，ADD_In1与ADD_In2的值分别为TIME #1s和TIME #2s，当变量ADD_EN为TRUE时，Out1的值为TIME #3s（如下表“变量2”所示）。
- 变量ADD_In1、 ADD_In2、Out1的数据类型分别为DT、TIME、DT，ADD_In1与ADD_In2的值分别为DT#2016-9-1-8:00:00和TIME#1H53M34S，当变量ADD_EN为TRUE时，Out1的值为DT#2016-09-01-09:53:34（如下表“变量3”所示）。

➢ **变量 1**

变量名	数据类型	当前值
ADD_EN	BOOL	TRUE
ADD_In1	INT	10
ADD_In2	INT	50
Out1	INT	60

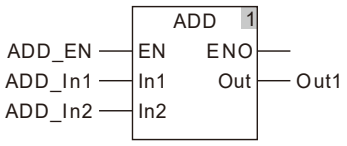
➢ **变量 2**

变量名	数据类型	当前值
ADD_EN	BOOL	TRUE
ADD_In1	TIME	TIME #1s
ADD_In2	TIME	TIME #2s
Out1	TIME	TIME #3s

➢ **变量 3**

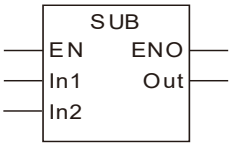
变量名	数据类型	当前值
ADD_EN	BOOL	TRUE
ADD_In1	DT	DT#2016-9-1-8:00:00
ADD_In2	TIME	TIME#1H53M34S
Out1	DT	DT#2016-09-01-09:53:34

➢ **程序**



8.9.2 SUB（减法指令）

FB/FC	说明	适用机种
FC	本指令用于将两个变量或常量相减。	DVP15MC11T DVP15MC11T-06



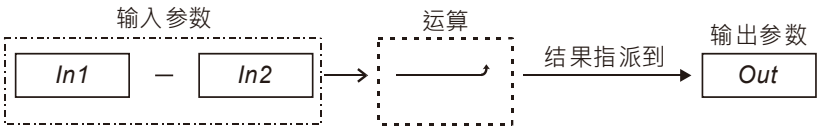
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	被减数	输入	被减数	由与输入参数所连变量的数据类型决定
In2	减数	输入	减数	由与输入参数所连变量的数据类型决定
Out	差	输出	In1与In2的差	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
In2		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个变量或常量相减，结果输出至“Out”，即Out = In1 - In2。
  - 对于位串、整数、实数类型，本指令允许输入参数“In1”与“In2”为不同数据类型的变量。当“In1”与“In2”为不同数据类型的变量时，以能包含“In1”和“In2”所有取值范围的数据类型进行运算，例如“In1”的数据类型为INT，“In2”的数据类型为DINT，则“Out”的数据类型为DINT。



  - 对于位串、整数、实数类型，本指令允许输入和输出变量为不同数据类型。当输入和输出变量为不同数据类型时，输出变量数据类型的范围须包含所有输入变量数据类型的范围，否则，软件编译时会报错。例如：“In1”和“In2”的数据类型分别为INT和DINT，则“Out”的数据类型为DINT，此时若“Out”所连变量的数据类型为INT，则软件编译报错；若“Out”所连变量的数据类型为LINT，则软件编译正确。

- 对于时间和日期类型，仅支持以下几种类型相减：
  1. In1为TIME类型，In2为TIME类型，Out为TIME类型。
  2. In1为TOD类型，In2为TIME类型，Out为TOD类型。
  3. In1为TOD类型，In2为TOD类型，Out为TIME类型。
  4. In1为DATE类型，In2为DATE类型，Out为TIME类型。
  5. In1为DT类型，In2为DT类型，Out为TIME类型。
  6. In1为DT类型，In2为TIME类型，Out为DT类型。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In1”与“In2”之差可能超出“Out”数据类型的有效范围。

  例如：“SUB_In1”与“SUB_In2”的数据类型均为 INT，值分别为 - 32768 和 1，如果输出变量数据类型为 INT 时，输出变量的值为 32767 (如下表“变量 1”所示); 如果输出变量的数据类型设置为 DINT，则输出变量的值为 - 32769 (如下表“变量 2”所示)。

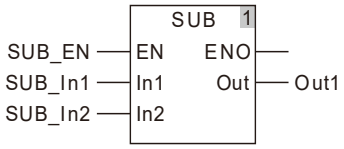
  ➢ **变量 1**

变量名	数据类型	当前值
SUB_EN	BOOL	TRUE
SUB_In1	INT	- 32768
SUB_In2	INT	1
Out1	INT	32767

  ➢ **变量 2**

变量名	数据类型	当前值
SUB_EN	BOOL	TRUE
SUB_In1	INT	- 32768
SUB_In2	INT	1
Out1	DINT	- 32769

  ➢ **程序**



程序范例

- 变量SUB_In1、SUB_In2、Out1的数据类型均为INT，SUB_In1与SUB_In2的值分别为100和40，当变量SUB_EN为TRUE时，Out1的值为60 (如下表“变量1”所示)。
- 变量SUB_In1、SUB_In2、Out1的数据类型均为TIME，SUB_In1与SUB_In2的值分别为TIME#4s和TIME#1s，当变量SUB_EN为TRUE时，Out1的值为TIME#3s (如下表“变量2”所示)。
- 变量SUB_In1、SUB_In2、Out1的数据类型分别为DATE、DATE、TIME，SUB_In1与SUB_In2的值分别为DATE#2016-10-1和DATE#2016-9-1，当变量SUB_EN为TRUE时，Out1的值为TIME#30D (如下表“变量3”所示)。

➢ **变量 1**

变量名	数据类型	当前值
SUB_EN	BOOL	TRUE
SUB_In1	INT	100
SUB_In2	INT	40
Out1	INT	60

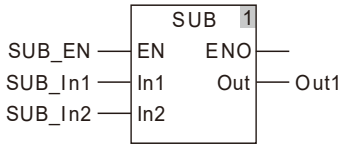
➢ **变量 2**

变量名	数据类型	当前值
SUB_EN	BOOL	TRUE
SUB_In1	TIME	TIME#4s
SUB_In2	TIME	TIME#1s
Out1	TIME	TIME#3s

➢ **变量 3**

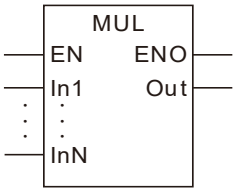
变量名	数据类型	当前值
SUB_EN	BOOL	TRUE
SUB_In1	DATE	DATE#2016-10-1
SUB_In2	DATE	DATE#2016-9-1
Out1	TIME	TIME#30D

程序



8.9.3 MUL（乘法指令）

FB/FC	说明	适用机种
FC	本指令用于将两个（或者多个）变量或常量相乘。	DVP15MC11T DVP15MC11T-06



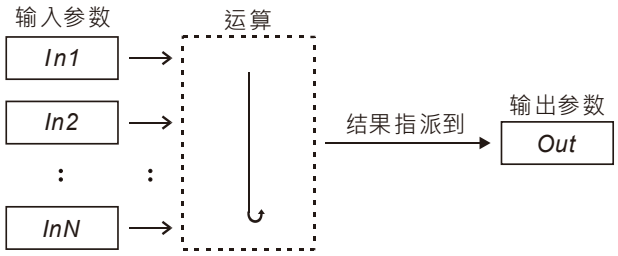
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	被乘数	输入	被乘数	**由与输入参数所连变量的数据类型决定**
In2 至 InN	乘数	输入	程序编写时，可通过编程软件增加或减少乘数，但乘数最多为7个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	积	输出	In1 ~ InN的积	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个（或者多个）变量或常量相乘，结果输出至“Out”，即Out = In1 * In2 * … * InN。
  - 本指令允许输入参数“In1”~“InN”为不同数据类型的变量。当“In1”~“InN”为不同数据类型的变量时，以能包含“In1”~“InN”所有取值范围的数据类型进行运算，例如“In1”的数据类型为INT，“In2”的数据类型为DINT，则“Out”的数据类型为DINT。



■ 本指令允许输入和输出变量为不同数据类型。当输入和输出变量为不同数据类型时，输出变量数据类型的范围须包含所有输入变量数据类型的范围，否则，软件编译时会报错。例如："In1" 和 "In2" 的数据类型分别为 INT 和 DINT，则 "Out" 的数据类型为 DINT，此时若 "Out" 所连变量的数据类型为 INT，则软件编译报错；若 "Out" 所连变量的数据类型为 LINT，则软件编译正确

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

■ "In1" ~ "InN" 之积可能超出 "Out" 数据类型的有效范围。

例如："MUL_In1" 与 "MUL_In2" 的数据类型均为 INT，值分别为 20000 和 2，如果输出变量数据类型为 INT 时，输出变量的值为 - 25536 (如下表 "变量 1" 所示) ; 如果输出变量的数据类型设置为 DINT，则输出变量的值为 40000 (如下表 "变量 2" 所示)。

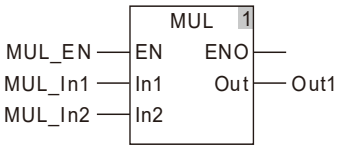
➢ **变量 1**

变量名	数据类型	当前值
MUL_EN	BOOL	TRUE
MUL_In1	INT	20000
MUL_In2	INT	2
Out1	INT	-25536

➢ **变量 2**

变量名	数据类型	当前值
MUL_EN	BOOL	TRUE
MUL_In1	INT	20000
MUL_In2	INT	2
Out1	DINT	40000

➢ **程序**

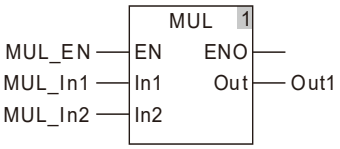


程序范例

■ 变量MUL_In1、MUL_In2、Out1的数据类型均为INT，MUL_In1与MUL_In2的值分别为10和50，当变量MUL_EN为TRUE时，Out1的值为500。

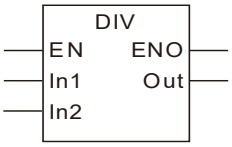
➢ **变量和程序**

变量名	数据类型	当前值
MUL_EN	BOOL	TRUE
MUL_In1	INT	10
MUL_In2	INT	50
Out1	INT	500



8.9.4 DIV（除法指令）

FB/FC	说明	适用机种
FC	本指令用于将两个变量或常量相除。	DVP15MC11T DVP15MC11T-06



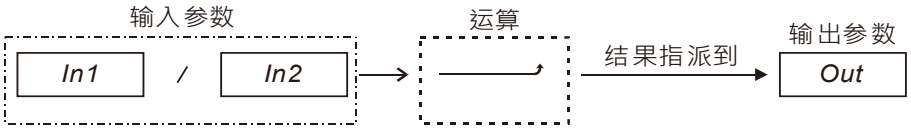
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	被除数	输入	被除数	由与输入参数所连变量的数据类型决定
In2	除数	输入	除数	由与输入参数所连变量的数据类型决定，但0除外
Out	商	输出	In1与In2作除法运算的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
In2		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个变量或常量相除，结果输出至“Out”，即Out = In1 / In2。
  - 本指令允许输入参数“In1”与“In2”为不同数据类型的变量。当“In1”与“In2”为不同数据类型的变量时，以能包含“In1”和“In2”所有取值范围的数据类型进行运算，例如“In1”的数据类型为INT，“In2”的数据类型为DINT，则“Out”的数据类型为DINT。



  - 本指令允许输入和输出变量为不同数据类型。当输入和输出变量为不同数据类型时，输出变量数据类型的范围须包含所有输入变量数据类型的范围，否则，软件编译时会报错。例如：“In1”和“In2”的数据类型分别为INT和DINT，则“Out”的数据类型为DINT，此时若“Out”的数据类型为INT，则软件编译报错；若“Out”的数据类型为LINT，则软件编译正确。

注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In2”的输入值不能为0，即除法运算的除数不能为0；若“In2”的输入值为0时，“Out”的值为0。
- “In1”与“In2”之商可能超出“Out”数据类型的有效范围。

“DIV_In1”与“DIV_In2”的数据类型均为INT，值分别为 - 32768和 - 1，如果输出变量数据类型为INT时，输出变量的值为 - 32768（如下表“变量1”所示）；如果输出变量的数据类型设置为DINT，则输出变量的值为32768（如下表“变量2”所示）。

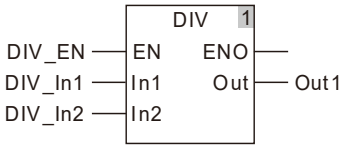
➢ **变量 1**

变量名	数据类型	当前值
DIV_EN	BOOL	TRUE
DIV_In1	INT	- 32768
DIV_In2	INT	- 1
Out1	INT	- 32768

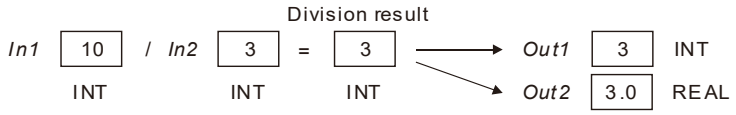
➢ **变量 2**

变量名	数据类型	当前值
DIV_EN	BOOL	TRUE
DIV_In1	INT	- 32768
DIV_In2	INT	- 1
Out1	DINT	32768

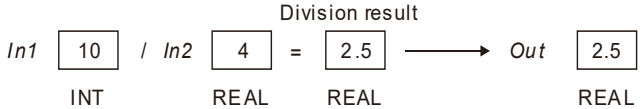
➢ **程序**



- 整数与整数相除，其结果均为整数，即使不能整除。例如“In1”与“In2”的数据类型均为INT，值分别为10和3，当“Out”的数据类型为整数（如INT）时其值为3、3.0，原因在如下图所示：



- 整数与实数或实数与实数相除，“Out”的数据类型均为实数，“Out”的值在不能整除时能获取到商的小数部分，如下图所示：

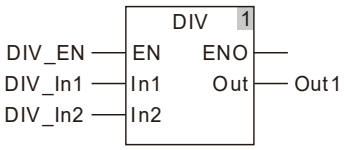


程序范例

- 变量 DIV_In1、DIV_In2、Out1 的数据类型均为 INT，DIV_In1 与 DIV_In2 的值分别为 100 和 20，当变量 DIV_EN 为 TRUE 时，Out1 的值为 5。

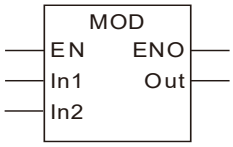
➢ **变量和程序**

变量名	数据类型	当前值
DIV_EN	BOOL	TRUE
DIV_In1	INT	100
DIV_In2	INT	20
Out1	INT	5



8.9.5 MOD（整数取余）

FB/FC	说明	适用机种
FC	本指令用于将两个整型的变量或常量作取余运算。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	被除数	输入	被除数	由与输入参数所连变量的数据类型决定
In2	除数	输入	除数	由与输入参数所连变量的数据类型决定，但0除外
Out	余数	输出	In1除以In2的余数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●							
In2		●	●	●	●	●	●	●	●	●	●	●	●							
Out		●	●	●	●	●	●	●	●	●	●	●	●							

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个整型的变量或常量作取余运算，结果输出至“Out”，即Out = In1 – （ In1 / In2 ） * In2。
  - 本指令允许输入与输入或输入与输出变量为不同数据类型。当输入和输出变量为不同数据类型时，输出变量数据类型的范围须包含所有输入变量数据类型的范围，否则，软件编译时会报错。例如：“In1”“In2”的数据类型分别为INT和DINT，则“Out”的数据类型为DINT，此时若“Out”的数据类型为INT，则软件编译报错；若“Out”的数据类型为LINT，则软件编译正确。

⚠ 注意

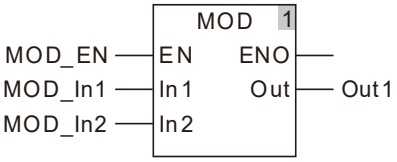
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In2”的输入值不能为0，即除法运算的除数不能为0；若“In2”的输入值为0时，“Out”的值为0。

程序范例

- 变量MOD_In1、MOD_In2、Out1的数据类型均为INT，MOD_In1与MOD_In2的值分别为10和4， 当变量MOD_EN为TRUE时，Out1的值为2。

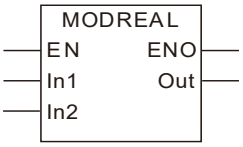
  ➢ **变量和程序**

变量名	数据类型	当前值
MOD_EN	BOOL	TRUE
MOD_In1	INT	10
MOD_In2	INT	4
Out1	INT	2



8.9.6 MODREAL（浮点数取余）

FB/FC	说明	适用机种
FC	本指令用于将两个浮点型的变量或常量作取余运算。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	被除数	输入	被除数	由与输入参数所连变量的数据类型决定
In2	除数	输入	除数	由与输入参数所连变量的数据类型决定，但0除外
Out	余数	输出	In1除以In2的余数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														●	●					
In2														●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个浮点型的变量或常量作取余运算，结果输出至“Out”。
  - 本指令允许输入与输入或输入与输出变量为不同数据类型。

注意

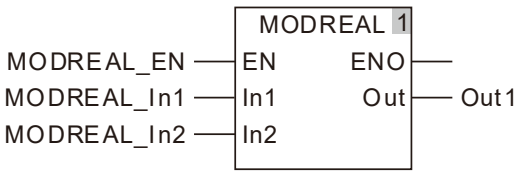
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In2”的输入值不能为0，即除法运算的除数不能为0；若“In2”的输入值为0时，“Out”的值为0。

程序范例

- 变量MODREAL_In1、 MODREAL_In2、Out1的数据类型分别为REAL、REAL、LREAL，MODREAL_In1与MODREAL_In2的值分别为10.5和2.5，当变量MODREAL_EN为TRUE时，Out1的值为0.5。

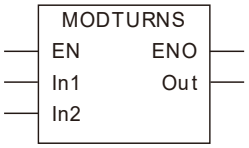
➢ **变量和程序**

变量名	数据类型	当前值
MODREAL_EN	BOOL	TRUE
MODREAL_In1	REAL	10.5
MODREAL_In2	REAL	2.5
Out1	LREAL	0.5



8.9.7 MODTURNS（计算圈数）

FB/FC	说明	适用机种
FC	本指令用于将两个浮点型的变量或常量做模数除法并取带符号的整数部分	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	输入值	输入	输入值	由与输入参数所连变量的数据类型决定
In2	模数范围	输入	模数范围	由与输入参数所连变量的数据类型决定，但0除外
Out	模旋转数	输出	模旋转数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														●	●					
In2														●	●					
Out												●								

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个浮点型的变量或常量做模数除法，取带符号的整数部分，结果输出至“Out”。可由轴的绝对设定位置计算出其模旋转数。
  - 本指令允许输入与输入或输入与输出变量为不同数据类型。

⚠ 注意

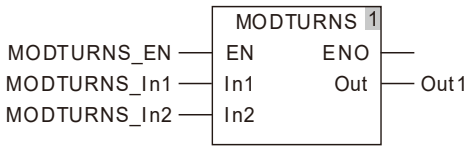
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In2”的输入值不能为0，即除法运算的除数不能为0；若“In2”的输入值为0时，“Out”的值为0。

程序范例

- 变量MODTURNS_In1、MODTURNS_In2的数据类型均为REAL、Out1的数据类型为DINT，MODTURNS_In1与MODTURNS_In2的值分别为800.23和360. 0，当变量MODTURNS_EN为TRUE时，Out1的值为2。

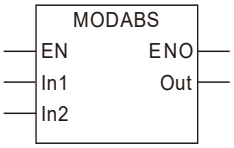
➢ **变量和程序**

变量名	数据类型	当前值
MODTURNS_EN	BOOL	TRUE
MODTURNS_In1	REAL	800.23
MODTURNS_In2	REAL	360.0
Out1	DINT	2



8.9.8 MODABS（计算相位）

FB/FC	说明	适用机种
FC	本指令用于将两个浮点型的变量或常量做模数除法并取无符号的模数值	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	输入值	输入	输入值	由与输入参数所连变量的数据类型决定
In2	模数范围	输入	模数范围	由与输入参数所连变量的数据类型决定，但0除外
Out	模数值	输出	模数值	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														●	●					
In2														●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将两个浮点型的变量或常量做模数除法，取无符号的模数值，结果输出至“Out”。可由轴的绝对位置计算出其模数位置。
  - 本指令允许输入与输入或输入与输出变量为不同数据类型。

8

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In2”的输入值不能为0，即除法运算的除数不能为0；若“In2”的输入值为0时，“Out”的值为0。

程序范例

- 变量MODABS_In1、MODABS_In2的数据类型均为REAL、Out1的数据类型为LREAL，当MODABS_In1与MODABS_In2的值分别为400.23和360.0，当变量MODABS_EN为TRUE时，Out1的值为40.2300109863281（如下表“变量1”所示）；当MODABS_In1与MODABS_In2的值分别为-400.23和360.0，当变量MODABS_EN为TRUE时，Out1的值为319.769989013672（如下表“变量2”所示）。

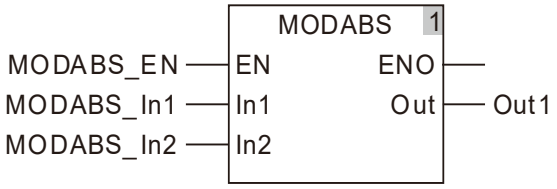
➢ **变量 1**

变量名	数据类型	当前值
MODABS_EN	BOOL	TRUE
MODABS_In1	REAL	400.23
MODABS_In2	REAL	360.0
Out1	LREAL	40.2300109863281

➢ **变量 2**

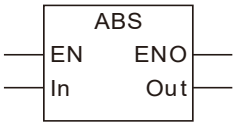
变量名	数据类型	当前值
MODABS_EN	BOOL	TRUE
MODABS_In1	REAL	- 400.23
MODABS_In2	REAL	360.0
Out1	LREAL	319.769989013672

➢ **程序**



8.9.9 ABS（取绝对值）

FB/FC	说明	适用机种
FC	本指令用于取整数或浮点数的绝对值。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	原始值	输入	用于取绝对值的原始值	由与输入参数所连变量的数据类型决定
Out	绝对值	输出	In的绝对值	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于获取输入参数“In”的取绝对值并输出至“Out”，即Out = | In1 |。
  - 本指令允许输入与输出变量为不同数据类型。当输入和输出变量为不同数据类型时，输出变量数据类型的范围须包含所有输入变量数据类型的范围，否则，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

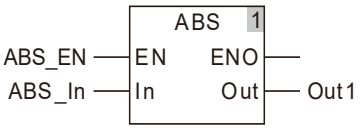
- 变量ABS_In、Out1的数据类型均为INT，ABS_In的值为 - 10，当变量ABS_EN为TRUE时，Out1的值为10（如下表“变量1”所示）；当ABS_In的值为20时，Out1的值为20（如下表“变量2”所示）。
  - **变量 1**

变量名	数据类型	当前值
ABS_EN	BOOL	TRUE
ABS_In	INT	-10
Out1	INT	10

➢ **变量 2**

变量名	数据类型	当前值
ABS_EN	BOOL	TRUE
ABS_In	INT	20
Out1	INT	20

➢ **程序**



8.9.10 DegToRad（角度转弧度）

FB/FC	说明	适用机种
FC	本指令用于将角度转换为弧度。	DVP15MC11T DVP15MC11T-06

- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	角度	输入	用于转换的角度	由与输入参数所连变量的数据类型决定
Out	弧度	输出	转换后的弧度	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将输入参数“In”转换成弧度并输出至“Out”，即Out = （ In / 180 ） * π。
  - “In”的单位为度（°），“Out”的单位为弧度。
  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量DegToRad_In、Out1的数据类型分别为INT和LREAL，当DegToRad_In的值为10，当变量DegToRad_EN为TRUE时，Out1的值为0.174532925199433 （如下表“变量1”所示）；当DegToRad_In的值为 - 10，Out1的值为 - 0.174532925199433（如下表“变量2”所示）。
  - **变量 1**

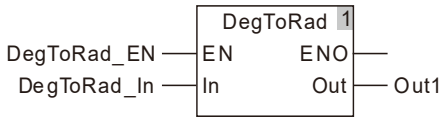
变量名	数据类型	当前值
DegToRad_EN	BOOL	TRUE

变量名	数据类型	当前值
DegToRad_In	INT	10
Out1	LREAL	0.174532925199433

➢ **变量 2**

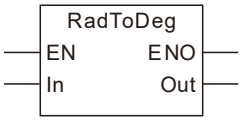
变量名	数据类型	当前值
DegToRad_EN	BOOL	TRUE
DegToRad_In	INT	- 10
Out1	LREAL	- 0.174532925199433

➢ **程序**



8.9.11 RadToDeg（弧度转角度）

FB/FC	说明	适用机种
FC	本指令用于将弧度转换为角度。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	弧度	输入	用于转换的弧度	由与输入参数所连变量的数据类型决定
Out	角度	输出	转换后的角度	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将输入参数“In”转换成角度并输出至“Out”，即Out = （ In / π ） * 180。
  - “In”的单位为弧度，“Out”的单位为度（°）。
  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量RadToDeg_In、Out1的数据类型分别为INT和LREAL，RadToDeg_In的值为10，当变量RadToDeg_EN为TRUE时，Out1的值为572.957795130824（如下表“变量1”所示）；当RadToDeg_In的值为 - 10时，Out1的值为 - 572.957795130824（如下表“变量2”所示）。
  - **变量 1**

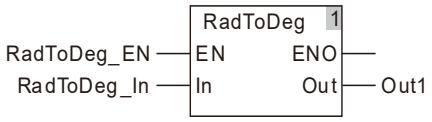
变量名	数据类型	当前值
RadToDeg_EN	BOOL	TRUE

变量名	数据类型	当前值
RadToDeg_In	INT	10
Out1	LREAL	572.957795130824

➢ **变量 2**

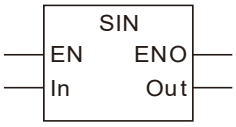
变量名	数据类型	当前值
RadToDeg_EN	BOOL	TRUE
RadToDeg_In	INT	- 10
Out1	LREAL	- 572.957795130824

➢ **程序**



8.9.12 SIN（正弦）

FB/FC	说明	适用机种
FC	本指令用于计算正弦值，结果输出至“Out”。“In”的单位为弧度。	DVP15MC11T DVP15MC11T-06



- **参数**

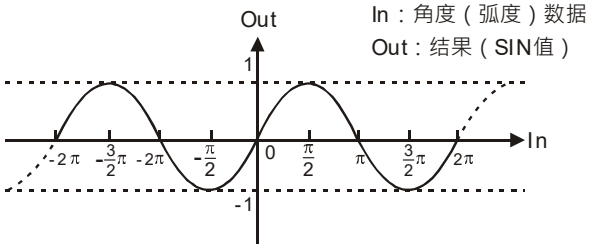
参数名称	含义	输入/输出	描述	参数取值范围
In	弧度	输入	用于求正弦值的弧度	由与输入参数所连变量的数据类型决定
Out	正弦值	输出	输入弧度对应的正弦值	- 1.00000000000000 ~ 1.00000000000000

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于计算输入参数“In”的正弦值，并输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量SIN_In、Out1的数据类型分别为INT和LREAL，SIN_In的值为10，当变量SIN_EN为TRUE时，Out1的值为 - 0.54402111088937（如下表“变量1”所示）；当SIN_In的值为 - 10时，Out1的值为0.54402111088937（如下表“变量2”所示）。

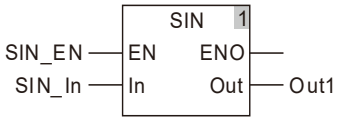
  ➢ **变量 1**

变量名	数据类型	当前值
SIN_EN	BOOL	TRUE
SIN_In	INT	10
Out1	LREAL	- 0.54402111088937

  ➢ **变量 2**

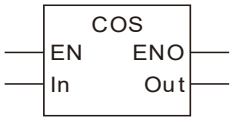
变量名	数据类型	当前值
SIN_EN	BOOL	TRUE
SIN_In	INT	- 10
Out1	LREAL	0.54402111088937

  ➢ **程序**



8.9.13 COS（余弦）

FB/FC	说明	适用机种
FC	本指令用于计算余弦值，结果输出至“Out”。“In”的单位为弧度。	DVP15MC11T DVP15MC11T-06



- **参数**

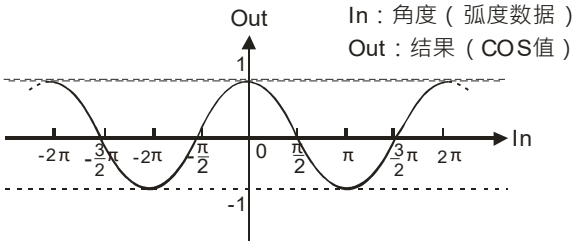
参数名称	含义	输入/输出	描述	参数取值范围
In	弧度	输入	用于求余弦值的弧度	由与输入参数所连变量的数据类型决定
Out	余弦值	输出	输入弧度对应的余弦值	-1.00000000000000~ 1.00000000000000

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于计算输入参数“In”的余弦值，并输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量COS_In、Out1的数据类型分别为INT和LREAL，COS_In的值为10，当变量COS_EN为TRUE时，Out1的值为 - 0.839071529076452 (如下表“变量1”所示); 当COS_In的值为 - 10时，Out1的值为 - 0.839071529076452 (如下表“变量2”所示)。

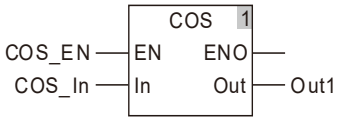
  ➢ **变量和程序**

变量名	数据类型	当前值
COS_EN	BOOL	TRUE
COS_In	INT	10
Out1	LREAL	- 0.839071529076452

  ➢ **变量**

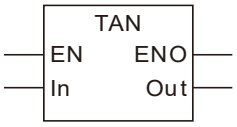
变量名	数据类型	当前值
COS_EN	BOOL	TRUE
COS_In	INT	-10
Out1	LREAL	- 0.839071529076452

  ➢ **程序**



8.9.14 TAN（正切）

FB/FC	说明	适用机种
FC	本指令用于计算正切值，结果输出至“Out”。“In”的单位为弧度。	DVP15MC11T DVP15MC11T-06



- **参数**

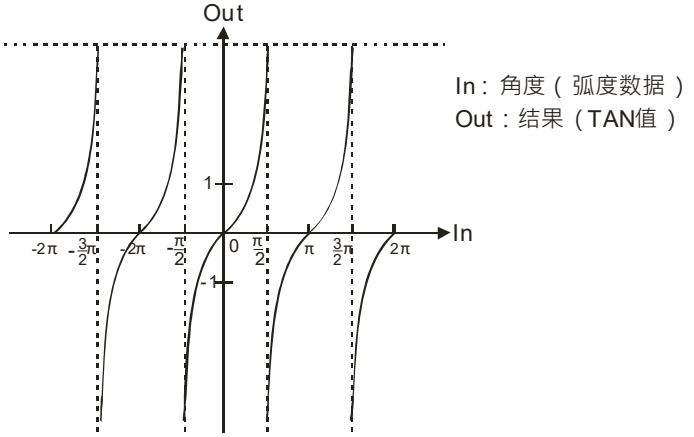
参数名称	含义	输入/输出	描述	参数取值范围
In	弧度	输入	用于求正切值的弧度	由与输入参数所连变量的数据类型决定
Out	正切值	输出	输入弧度对应的正切值	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  ■ 本指令用于计算输入参数“In”的正切值，并输出至“Out”。



  ■ 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量TAN_In、Out1的数据类型分别为INT和LREAL，TAN_In的值为10，当变量TAN_EN为TRUE时，Out1的值为0.648360827459087（如下表“变量1”所示）；当TAN_In的值为 - 10时，Out1的值为-0.648360827459087（如下表“变量2”所示）。

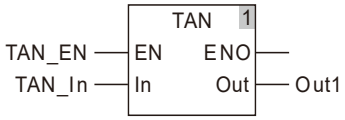
  ➢ **变量**

变量名	数据类型	当前值
TAN_EN	BOOL	TRUE
TAN_In	INT	10
Out1	LREAL	0.648360827459087

  ➢ **变量**

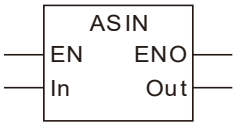
变量名	数据类型	当前值
TAN_EN	BOOL	TRUE
TAN_In	INT	- 10
Out1	LREAL	- 0.648360827459087

  ➢ **程序**



8.9.15 ASIN（反正弦）

FB/FC	说明	适用机种
FC	本指令用于计算反正弦值，结果输出至“Out”。“Out”的单位为弧度。	DVP15MC11T DVP15MC11T-06



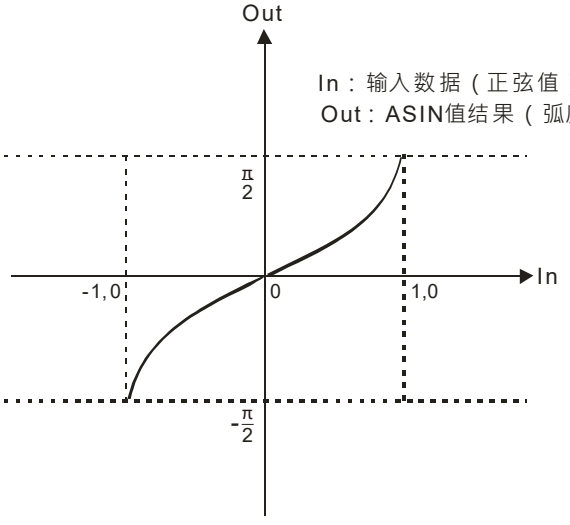
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	正弦值	输入	用于求反正弦的正弦值	由与输入参数所连变量的数据类型决定
Out	弧度	输出	正弦值对应的弧度值	- π/2 ~ π/2

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算输入参数“In”的反正弦值，并输出至“Out”。



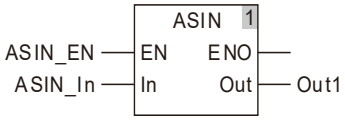
  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In”的输入值在 - 1.0到1.0之间变化时，“Out”的值在 - π/2与π/2之间变化；若“In”的输入值不在 - 1.0到1.0之间时，“Out”的值为非数字，此时指令并不会进入错误状态。

➢ **变量和程序**

变量名	数据类型	当前值
ASIN_EN	BOOL	TRUE
ASIN_In	REAL	2.0
Out1	LREAL	1.#QNAN



程序范例

- 变量ASIN_In、Out1的数据类型分别为REAL和LREAL，ASIN_In的值为1.0，当变量ASIN_EN为TRUE时，Out1的值为1.5707963267949（如下表“变量1”所示）；当ASIN_In的值为 - 1.0时，Out1的值为 - 1.5707963267949（如下表“变量2”所示）。

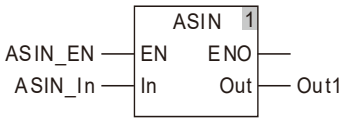
➢ **变量 1**

变量名	数据类型	当前值
ASIN_EN	BOOL	TRUE
ASIN_In	REAL	1.0
Out1	LREAL	1.5707963267949

➢ **变量 2**

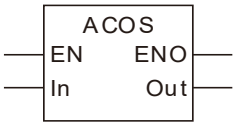
变量名	数据类型	当前值
ASIN_EN	BOOL	TRUE
ASIN_In	REAL	- 1.0
Out1	LREAL	- 1.5707963267949

➢ **程序**



8.9.16 ACOS（反余弦）

FB/FC	说明	适用机种
FC	本指令用于计算反余弦值，结果输出至“Out”。“Out”的单位为弧度。	DVP15MC11T DVP15MC11T-06



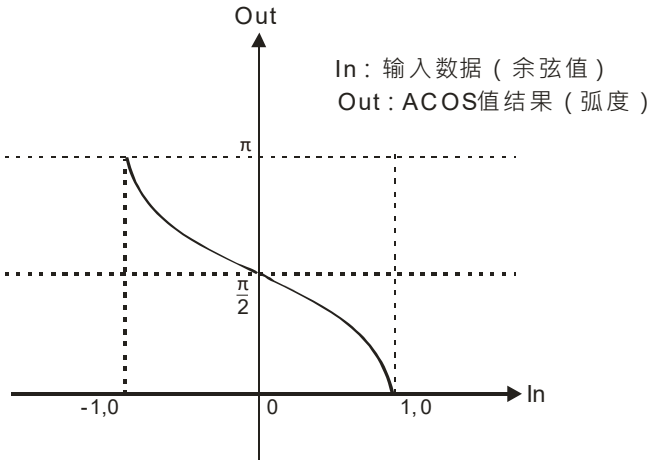
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	余弦值	输入	用于求反余弦的余弦值	由与输入参数所连变量的数据类型决定
Out	弧度	输出	余弦值对应的弧度值	0 ~ π

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算输入参数“In”的反余弦值，并输出至“Out”。



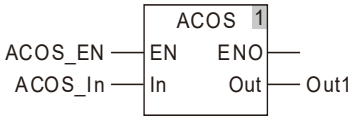
  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In”的输入值在 - 1.0到1.0之间变化时，“Out”的值在0与π之间变化 ;若“In”的输入值不在 - 1.0到1.0之间时，“Out”的值为非数字，此时指令并不会进入错误状态。

➢ **变量和程序**

变量名	数据类型	当前值
ACOS_EN	BOOL	TRUE
ACOS_In	REAL	2.0
Out1	LREAL	1.#QNAN



程序范例

- 变量ACOS_In、Out1的数据类型分别为REAL和LREAL，ACOS_In的值为1.0， 当变量ACOS_EN为TRUE时，Out1的值为0 (如下表“变量1”所示) ; 当ACOS_In的值为 - 1.0时，Ou1的值为3.14159265358979 (如下表“变量2”所示)。

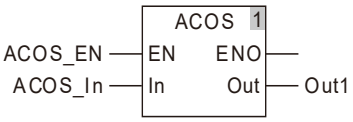
➢ **变量**

变量名	数据类型	当前值
ACOS_EN	BOOL	TRUE
ACOS_In	REAL	1.0
Out1	LREAL	0

➢ **变量**

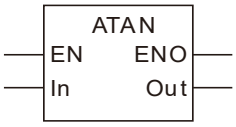
变量名	数据类型	当前值
ACOS_EN	BOOL	TRUE
ACOS_In	REAL	- 1.0
Out1	LREAL	3.14159265358979

➢ **程序**



8.9.17 ATAN（反正切）

FB/FC	说明	适用机种
FC	本指令用于计算反正切值，结果输出至“Out”。“Out”的单位为弧度。	DVP15MC11T DVP15MC11T-06



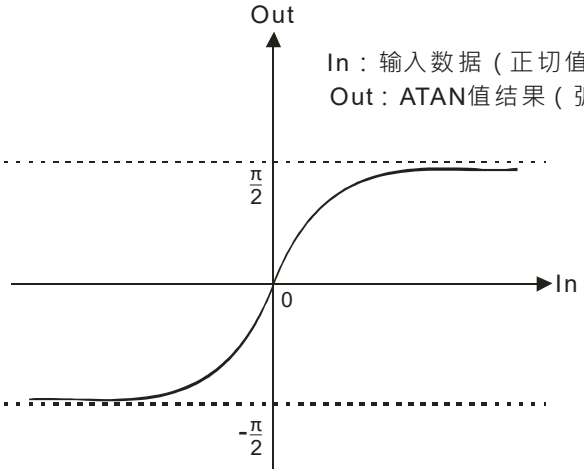
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	正切值	输入	用于求反正切的余弦值	由与输入参数所连变量的数据类型决定
Out	弧度	输出	正切值对应的弧度值	- π/2 ~ π/2

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算输入参数“In”的反正切值，并输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ **注意**

  - 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

- “In”的输入值为 - ∞时，“Out”的输出值为 - π/2；“In”的输入值为 + ∞时，“Out”的输出值为π/2。

程序范例

- 变量ATAN_In、Out1的数据类型分别为REAL和LREAL，ATAN_In的值为1.0，当变量ATAN_EN为TRUE时，Out1的值为0.785398163397448（如下表“变量1”所示）；当ATAN_In的值为 - 1.0时，Out1的值为 - 0.785398163397448（如下表“变量2”所示）。

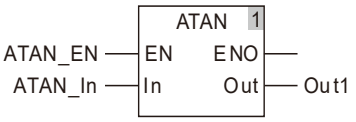
➢ **变量 1**

变量名	数据类型	当前值
ATAN_EN	BOOL	TRUE
ATAN_In	REAL	1.0
Out1	LREAL	0.785398163397448

➢ **变量 2**

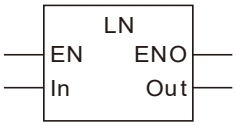
变量名	数据类型	当前值
ATAN_EN	BOOL	TRUE
ATAN_In	REAL	- 1.0
Out1	LREAL	- 0.785398163397448

➢ **程序**



8.9.18 LN（自然对数）

FB/FC	说明	适用机种
FC	本指令用于计算自然对数，结果输出至“Out”。	DVP15MC11T DVP15MC11T-06



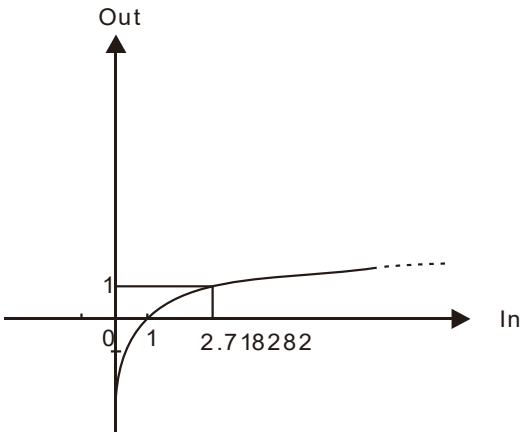
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	运算值	输入	运算值	由与输入参数所连变量的数据类型决定
Out	对数	输出	“In”的自然对数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算“In”的自然对数，即以e(e=2.718282)为底“In”的对数，并将结果输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

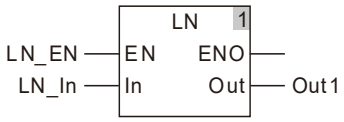
⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

■ “In”输入值为非正数时，“Out”的输出值为非数字，如下图所示：

➢ **变量和程序**

变量名	数据类型	当前值
LN_EN	BOOL	TRUE
LN_In	REAL	- 2.0
Out1	LREAL	1.#QNAN



程序范例

■ 变量LN_In、Out1的数据类型分别为INT和LREAL，LN_In的值为1，当变量LN_EN为TRUE时，Out1的值为0.0（如下表“变量1”所示）；变量LN_In、Out1的数据类型分别为REAL和LREAL，当LN_In的值为2.718282时，Out1的值为1.00000005734143（如下表“变量2”所示）。

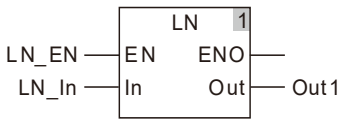
➢ **变量 1**

变量名	数据类型	当前值
LN_EN	BOOL	TRUE
LN_In	INT	1
Out1	LREAL	0.0

➢ **变量 2**

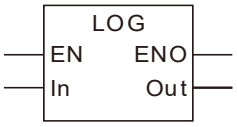
变量名	数据类型	当前值
LN_EN	BOOL	TRUE
LN_In	REAL	2.718282
Out1	LREAL	1.00000005734143

➢ **程序**



8.9.19 LOG（常用对数）

FB/FC	说明	适用机种
FC	本指令用于计算以 10 为底的对数，结果输出至“Out”。	DVP15MC11T DVP15MC11T-06



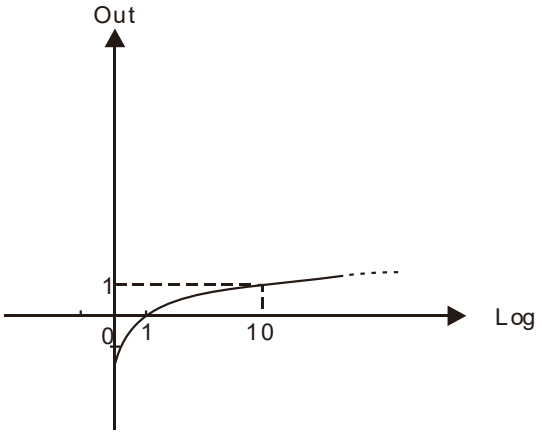
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	运算值	输入	运算值	由与输入参数所连变量的数据类型决定
Out	对数	输出	以10为底“In”的对数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算以10为底“In”的对数，并将结果输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

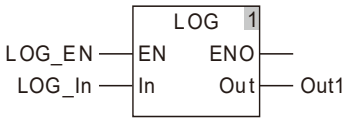
⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

- “In”输入值为非正数时，“Out”的输出值为非数字，如下所示：

  ➢ **变量和程序**

变量名	数据类型	当前值
LOG_EN	BOOL	TRUE
LOG_In	REAL	-2.0
Out1	LREAL	1.#QNAN



程序范例

- 变量LOG_In、Out1的数据类型分别为INT和LREAL，LOG_In的值为1， 当变量LOG_EN为TRUE时，Out1的值为0.0（如下表“变量1”所示）；当LOG_In的值为10时，Out1的值为1.0（如下表“变量2”所示）。

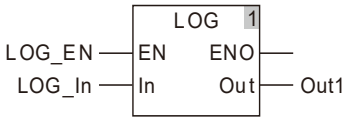
  ➢ **变量 1**

变量名	数据类型	当前值
LOG_EN	BOOL	TRUE
LOG_In	INT	1
Out1	LREAL	0.0

  ➢ **变量 2**

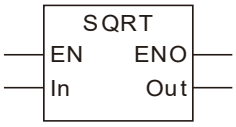
变量名	数据类型	当前值
LOG_EN	BOOL	TRUE
LOG_In	INT	10
Out1	LREAL	1.0

  ➢ **程序**



8.9.20 SQRT（求平方根）

FB/FC	说明	适用机种
FC	本指令用于计算平方根，结果输出至“Out”。	DVP15MC11T DVP15MC11T-06



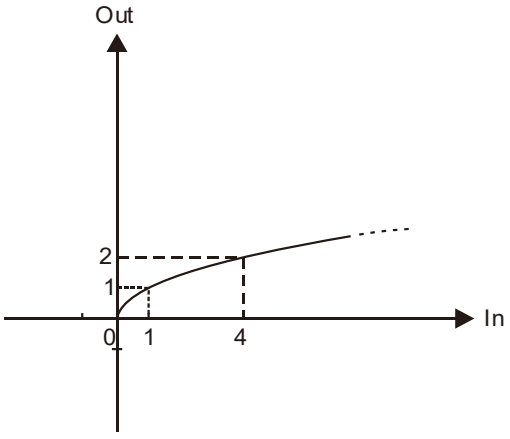
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	运算值	输入	运算值	由与输入参数所连变量的数据类型决定，且为非负数
Out	平方根	输出	平方根	由与输出参数所连变量的数据类型决定，且为非负数

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算 “In”的平方根，并将结果输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

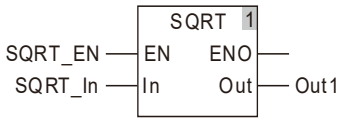
⚠ **注意**

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

■ “In”输入值为负数时，“Out”的输出值为非数字。

➢ **变量和程序**

变量名	数据类型	当前值
SQRT_EN	BOOL	TRUE
SQRT_In	REAL	- 2.0
Out1	LREAL	1.#QNAN



程序范例

■ 变量SQRT_In、Out1的数据类型分别为INT和LREAL，SQRT_In的值为16，当变量SQRT_EN为TRUE时，Out1的值为4.0 (如下表“变量1”所示) ; 当SQRT_In的值为100时，Out1的值为10 (如下表“变量2”所示)。

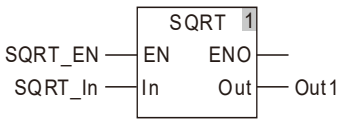
➢ **变量 1**

变量名	数据类型	当前值
SQRT_EN	BOOL	TRUE
SQRT_In	INT	16
Out1	LREAL	4.0

➢ **变量 2**

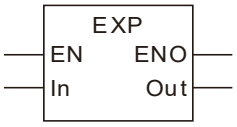
变量名	数据类型	当前值
SQRT_EN	BOOL	TRUE
SQRT_In	INT	100
Out1	LREAL	10.0

➢ **程序**



8.9.21 EXP（自然指数）

FB/FC	说明	适用机种
FC	本指令用于计算以 e 为底，“In”为指数的结果，并将结果输出至“Out”。	DVP15MC11T DVP15MC11T-06



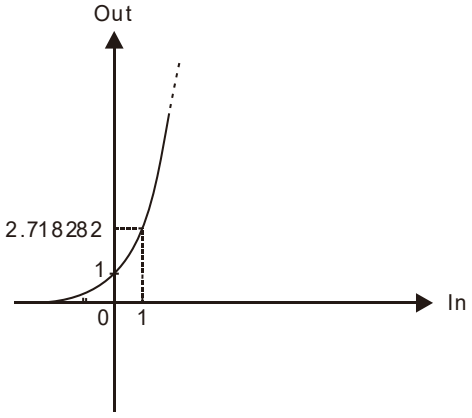
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	指数	输入	指数	由与输入参数所连变量的数据类型决定
Out	计算结果	输出	以e为底，“In”为指数的计算结果	由与输出参数所连变量的数据类型决定，且为非负数

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算以e（e=2.718282）为底，“In”为指数的结果，并将结果输出至“Out”。



  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- “In”输入值为特殊情形时，对应“Out”的输出值如下表所示：

In	Out
0	1.0
正数	正数
负数.	正数
非数字	非数字

程序范例

- 变量EXP_In、Out1的数据类型分别为INT和LREAL，EXP_In的值为0，当变量EXP_EN为TRUE时，Out1的值为1.0 (如下表“变量1”所示) ; 当EXP_In的值为1时，Out1的值为2.71828182845905 (如下表“变量2”所示)。

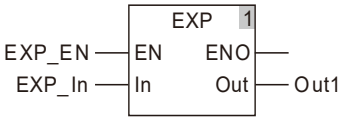
➢ **变量 1**

变量名	数据类型	当前值
EXP_EN	BOOL	TRUE
EXP_In	INT	0
Out1	LREAL	1.0

➢ **变量 2**

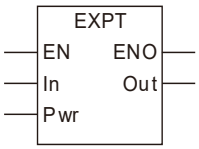
变量名	数据类型	当前值
EXP_EN	BOOL	TRUE
EXP_In	INT	1
Out1	LREAL	2.71828182845905

➢ **程序**



8.9.22 EXPT（幂指数）

FB/FC	说明	适用机种
FC	本指令用于计算以“In”为底，“Pwr”为指数的结果，并将结果输出至“Out”。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	底数	输入	底数	由与输入参数所连变量的数据类型决定
Pwr	指数	输入	指数	由与输入参数所连变量的数据类型决定
Out	计算结果	输出	以“In”为底，“Pwr”为指数的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Pwr		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于计算以“In”为底数，“Pwr”为指数的结果，并将结果输出至“Out”。
  - 本指令允许输入参数选择不同数据类型，但输出参数仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量EXPT_In与EXPT_Pwr的数据类型均为INT，值分别为10、2，Out1的数据类型为LREAL，当变量EXPT_EN为TRUE时，Out1的值为100.0（如下表“变量1”所示）；当EXPT_In与EXPT_Pwr的值分别为 - 10、2时，Out1的值为100.0（如下表“变量2”所示）。

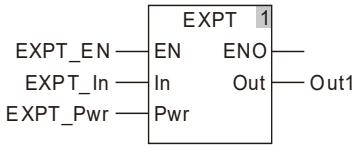
➢ **变量 1**

变量名	数据类型	当前值
EXPT_EN	BOOL	TRUE
EXPT_In	INT	10
EXPT_Pwr	INT	2
Out1	LREAL	100.0

➢ **变量 2**

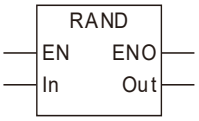
变量名	数据类型	当前值
EXPT_EN	BOOL	TRUE
EXPT_In	INT	- 10
EXPT_Pwr	INT	2
Out1	LREAL	100.0

➢ **程序**



8.9.23 RAND（随机数）

FB/FC	说明	适用机种
FC	本指令用于产生一个随机数	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	保留	输入	保留	由与输入参数所连变量的数据类型决定
Out	随机数值	输出	随机数值	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In								●												
Out												●								

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于产生一个随机数值，结果输出至“Out”。该值产生范围0 ~ 32767。
  - 本指令输入值对产生的随机数结果不产生影响，但是必须要填写。
  - 如果要产生固定范围的随机数，只需对产生的值进行取余运算即可。如需要0~10的随机数只需要“RAND（0） MOD 10”即可。

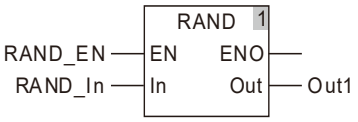
⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

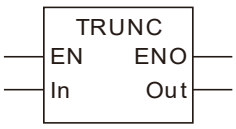
- 产生一个随机数，RAND（0）。
  - **变量和程序**

变量名	数据类型	当前值
RAND_EN	BOOL	TRUE
RAND_In	UDINT	0
Out1	DINT	15243


RAND
1
RAND_EN
EN
ENO
RAND_In
In
Out
Out1

8.9.24 TRUNC（浮点数取整数）

FB/FC	说明	适用机种
FC	该指令用于获取浮点数的整数部分。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	欲转换的浮点数	输入	用于取整数部分的浮点数	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	浮点数的整数部分	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out													●							

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于获取浮点数的整数部分，并将值输出至“*Out*”。
  - 本指令允许输入参数选择不同数据类型，但输出变量仅限于LINT，如果输出参数不为LINT，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量TRUNC _In的数据类型为REAL，值为 - 5.6时，Out1的数据类型为LINT，当变量TRUNC_EN为TRUE时，Out1的值为 - 5（如下表“变量1”所示）；当TRUNC _In的值为10.8时，Out1的值为10（如下表“变量2”所示）。
  - **变量 1**

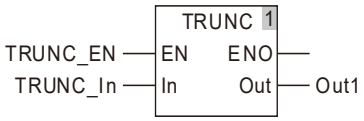
变量名	数据类型	当前值
TRUNC_EN	BOOL	TRUE
TRUNC_In	REAL	- 5.6

变量名	数据类型	当前值
Out1	LINT	- 5

➢ **变量 2**

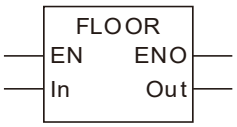
变量名	数据类型	当前值
TRUNC_EN	BOOL	TRUE
TRUNC_In	REAL	10.8
Out1	LINT	10

➢ **程序**



8.9.25 FLOOR（浮点数取整）

FB/FC	说明	适用机种
FC	该指令用于获取浮点数的整数部分。输入浮点数为负数时，输出为浮点数的整数部分减一。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	欲转换的浮点数	输入	用于取整数部分的浮点数	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	浮点数的整数部分	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out													●							

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于获取浮点数的整数部分，并将值输出至“*Out*”。
    输入浮点数为正数时，输出为浮点数的整数部分，如输入为3.5，则输出为3；输入浮点数为负数时，输出为浮点数的整数部分减一，如输入为 - 3.5，则输出为 - 4。
  - 本指令允许输入参数选择不同数据类型，但输出变量仅限于LINT，如果输出参数不为LINT，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量FLOOR_In的数据类型均为REAL，值为5.6时，Out1的数据类型为LINT，当变量FLOOR_EN为TRUE时，Out1的值为5（如下表“变量1”所示）；当FLOOR_In的值为 - 10.2时，Out1的值为 - 11（如下表“变量2”所示）。

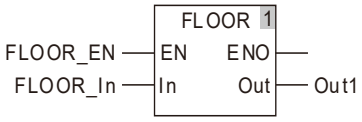
➢ **变量 1**

变量名	数据类型	当前值
FLOOR_EN	BOOL	TRUE
FLOOR_In	REAL	5.6
Out1	LINT	5

➢ **变量 2**

变量名	数据类型	当前值
FLOOR_EN	BOOL	TRUE
FLOOR_In	REAL	- 10.2
Out1	LINT	- 11

➢ **程序**



8.9.26 FRACTION（浮点数取小数）

FB/FC	说明	适用机种
FC	该指令用于获取浮点数的小数部分。	DVP15MC11T DVP15MC11T-06

- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	欲转换的浮点数	输入	用于取小数部分的浮点数	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	浮点数的小数部分	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out															●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于获取浮点数的小数部分，并将值输出至“*Out*”，输出结果的符号与输入一致。
  - 本指令允许输入变量选择不同数据类型，但输出变量仅限于LREAL，如果输出参数不为LREAL，软件编译时会报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量FRACTION _In的数据类型为REAL，值为 - 5.6时，Out1的数据类型为LREAL，当变量FRACTION_EN为TRUE时，Out1的值为 - 0.6（如下表“变量1”所示）；当FRACTION_In的值为10.8时，Out1的值为0.8（如下表“变量2”所示）。
  - **变量 1**

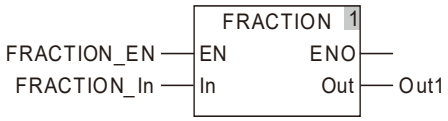
变量名	数据类型	当前值
FRACTION_EN	BOOL	TRUE
FRACTION_In	REAL	-5.6

变量名	数据类型	当前值
Out1	LREAL	-0.6

➢ **变量 2**

变量名	数据类型	当前值
FRACTION_EN	BOOL	TRUE
FRACTION_In	REAL	10.8
Out1	LREAL	0.8

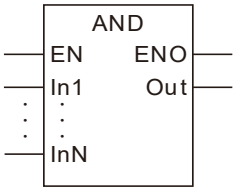
➢ **程序**



8.10 位串指令

8.10.1 AND（逻辑与）

FB/FC	说明	适用机种
FC	本指令用于将两个（或者多个）变量或常量作逻辑与运算。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	运算参数	输入	程序编写时，可通过编程软件增加或减少运算参数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	运算结果	输出	In1 ~ InN作逻辑与运算的结果	由与输出参数所连变量的数据类型决定

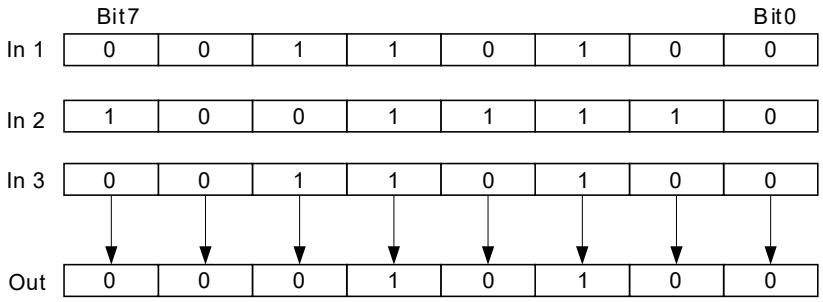
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

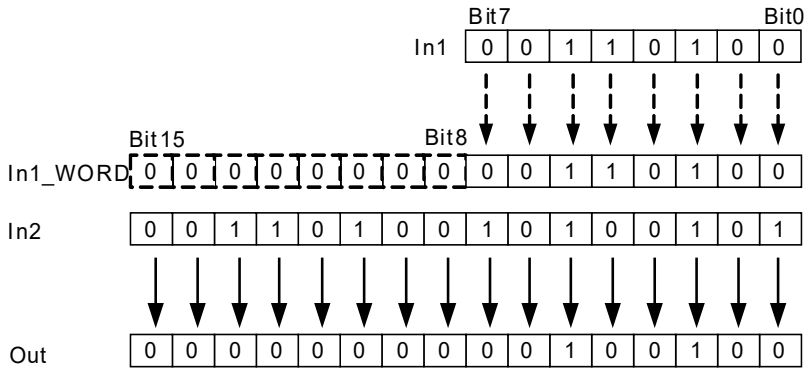
- **功能说明**

  - 本指令用于将两个（或者多个）变量或常量作逻辑与运算（按位与），结果输出至*Out*，即*Out = In1* & *In2* &...& *InN*。

    按位与的运算规则为：所有输入变量的某一对应位均为TRUE时，输出变量的对应位为TRUE，否则为FALSE，如下图所示：



- 当输入变量的数据类型均不为BOOL时，允许*In1~InN*为不同数据类型的变量。当*In1~InN*为不同数据类型的变量时，以能包含*In1~InN*所有取值范围的数据类型进行运算，例如*In1*的数据类型为BYTE，*In2*的数据类型为WORD，则*Out*的数据类型为WORD；运算时将*In1*由BYTE转换成WORD（补齐位全为0，如下图中的Bit8~Bit15）后再与*In2*按位作与运算，如下图所示：



- 若输入变量的数据类型为BOOL时，则要求所有输入和输出变量的数据类型均为BOOL，否则软件编译时将报错。

⚠ 注意

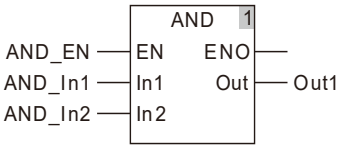
输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量AND_In1、 AND_In2、Out1的数据类型均为BYTE，AND_In1与 AND_In2的值分别为10和50，当变量AND_EN为TRUE时，Out1的值为2。

➢ **变量和程序**

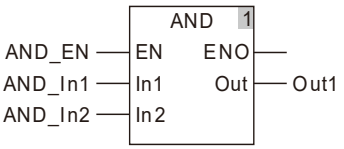
变量名	数据类型	当前值
AND_EN	BOOL	TRUE
AND_In1	BYTE	10
AND_In2	BYTE	50
Out1	BYTE	2



- 变量 AND_In1、 AND_In2、Out1的数据类型分别为BYTE、WORD、WORD，AND_In1与 AND_In2 的值分别为255和256， 当变量AND_EN为TRUE时，Out1的值为0。

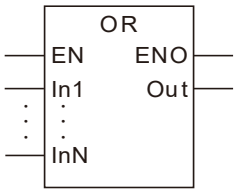
  ➢ **变量和程序**

变量名	数据类型	当前值
AND_EN	BOOL	TRUE
AND_In1	BYTE	255
AND_In2	WORD	256
Out1	WORD	0



8.10.2 OR（逻辑或）

FB/FC	说明	适用机种
FC	本指令用于将两个（或者多个）变量或常量作逻辑或运算。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	运算参数	输入	程序编写时，可通过编程软件增加或减少运算参数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	运算结果	输出	In1 ~ InN作逻辑或运算的结果	由与输出参数所连变量的数据类型决定

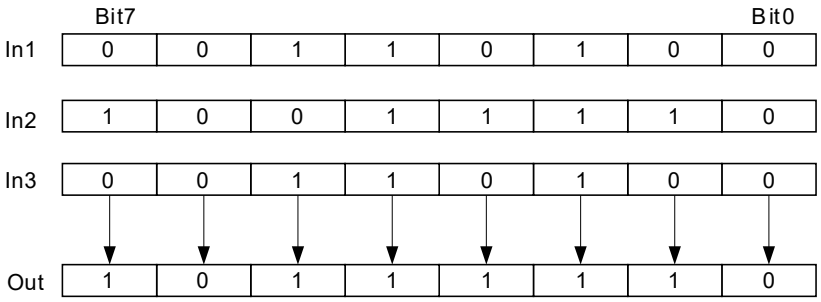
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

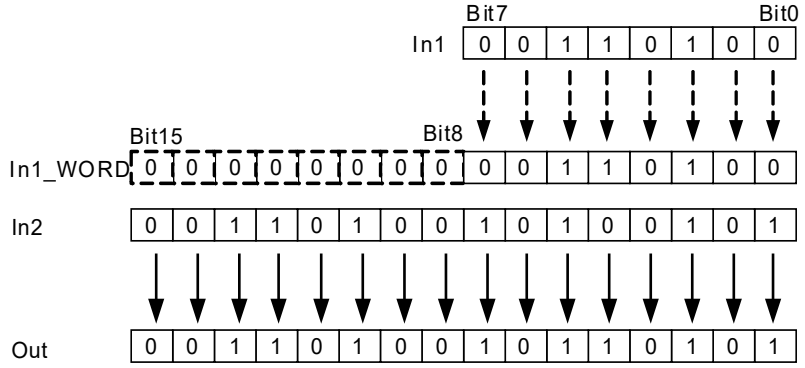
- **功能说明**

  ■ 本指令用于将两个（或者多个）变量或常量作逻辑或运算（按位或），结果输出至*Out*，即*Out= In1* OR I*n2* OR…OR *InN*。

  按位或的运算规则为：所有输入变量的某一对应位均为FALSE时，输出变量的对应位为FALSE，否则为TRUE，如下图所示：



- 当输入变量的数据类型均不为BOOL时，允许*In1~InN*为不同数据类型的变量。当*In1~InN*为不同数据类型的变量时，以能包含*In1~InN*所有取值范围的数据类型进行运算，例如*In1*的数据类型为BYTE，*In2*的数据类型为WORD，则*Out*的数据类型为WORD；运算时将*In1*由BYTE转换成WORD（补齐位全为0，如下图中的Bit8~Bit15）后再与*In2*按位作或运算，如下图所示：



- 若输入变量的数据类型为BOOL时，则要求所有输入和输出变量的数据类型均为BOOL，否则软件编译时将报错。

⚠ 注意

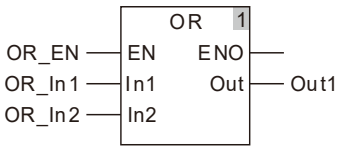
输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量OR_In1、 OR_In2和Out1的数据类型均为BYTE，OR_In1与OR_In2的值分别为10和50，当变量OR_EN为TRUE时，Out1的值为58。

  ➢ **变量和程序**

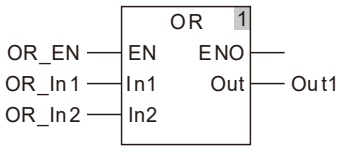
变量名	数据类型	当前值
OR_EN	BOOL	TRUE
OR_In1	BYTE	10
OR_In2	BYTE	50
Out1	BYTE	58



- 变量OR_In1、OR_In2、Out1的数据类型分别为BYTE和WORD，OR_In1与OR_In2的值分别为255和256，当变量OR_EN为TRUE时，Out1的值为511。

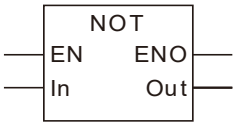
➢ **变量和程序**

变量名	数据类型	当前值
OR_EN	BOOL	TRUE
OR_In1	BYTE	255
OR_In2	WORD	256
Out1	WORD	511



8.10.3 NOT（取反）

FB/FC	说明	适用机种
FC	本指令用于将一个变量或常量作取反运算。	DVP15MC11T DVP15MC11T-06



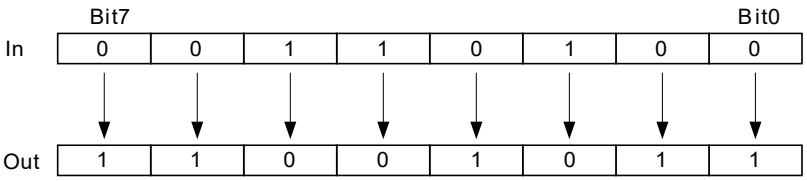
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	运算参数	输入	取反运算的输入参数	由与输入参数所连变量的数据类型决定
Out	运算结果	输出	取反运算的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将一个变量或常量作取反运算（按位取反），结果输出至*Out*。按位取反的运算规则为：若输入变量的某一位为TRUE，则输出变量的对应位为FALSE；若输入变量的某一位为FALSE，则输出变量的对应位为TRUE，如下图所示：



  - *Out*与*In*的数据类型一致。

⚠ 注意

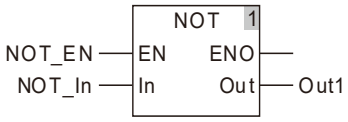
输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量NOT _In、Out1的数据类型均为BYTE，In的值为10，当变量NOT_EN为TRUE时，Out1的值为245。

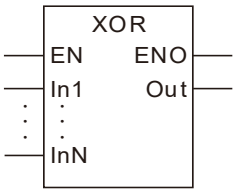
➢ **变量和程序**

变量名	数据类型	当前值
NOT_EN	BOOL	TRUE
NOT_In	BYTE	10
Out1	BYTE	245



8.10.4 XOR（异或）

FB/FC	说明	适用机种
FC	本指令用于将两个（或者多个）变量或常量作异或运算。	DVP15MC11T DVP15MC11T-06



- **参数**

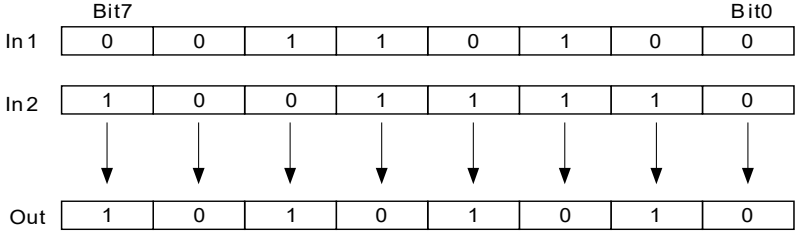
参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	运算参数	输入	程序编写时，可通过编程软件增加或减少运算参数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	运算结果	输出	In1 ~ InN作异或运算的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

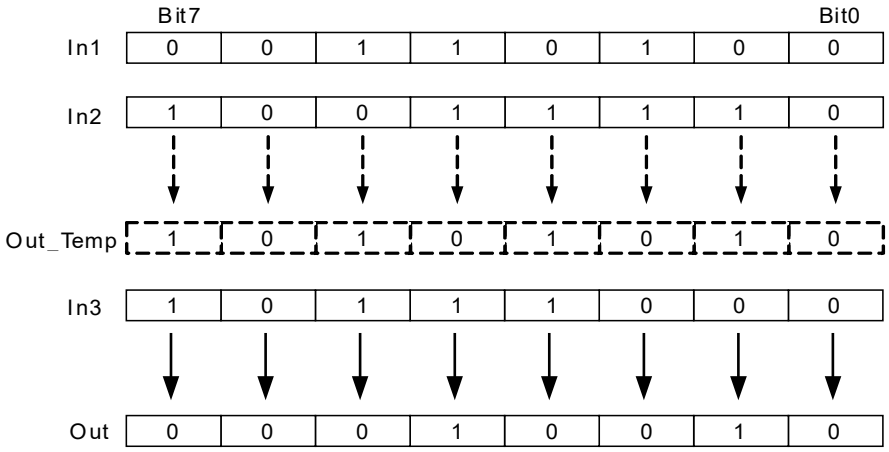
说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

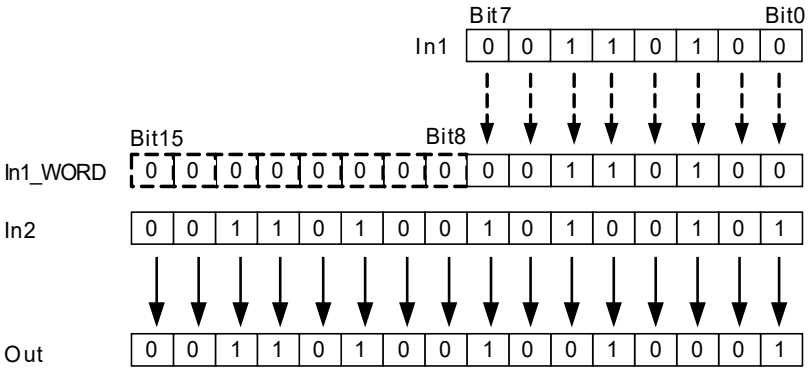
本指令用于将两个（或者多个）变量或常量作异或运算（按位异或），结果输出至*Out*，即*Out= In1* XOR *In2* XOR…XOR *InN*。如下图所示，In1与In2作异或运算的规则：



- 当存在多个输入参数时，运算步骤为：*In1*与*In2*作异或的结果再与*In3*作异或运算，以此类推直至*InN*，如下图所示，In1与In2作异或运算的结果为Out_Temp, Out_Temp与In3作异或运算的结果为Out。



- 当输入变量的数据类型均不为BOOL时，允许*In1~InN*为不同数据类型的变量。当*In1~InN*为不同数据类型的变量时，以能包含*In1~InN*所有取值范围的数据类型进行运算，例如*In1*的数据类型为BYTE，*In2*的数据类型为WORD，则*Out*的数据类型为WORD；运算时将*In1*由BYTE转换成WORD（补齐位全为0，如下图中的Bit8~Bit15）后再与*In2*按位作异或运算，如下图所示：



- 若输入变量的数据类型为BOOL时，则要求所有输入和输出变量的数据类型均为BOOL，否则软件编译时将报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量XOR_In1、XOR_In2、Out1的数据类型均为BYTE，XOR_In1与XOR_In2的值分别为10和50，当变量XOR_EN为TRUE时，Out1的值为56（如下表“变量1”所示）；当变量XOR_In1、XOR_In2和Out1的数据类型分别为BYTE、WORD和WORD，XOR_In1与XOR_In2的值分别为255和256，当变量XOR_EN为TRUE时，Out1的值为511（如下表“变量2”所示）。

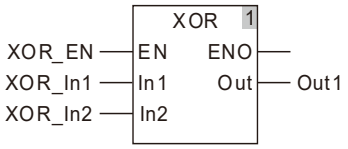
➢ **变量 1**

变量名	数据类型	当前值
XOR_EN	BOOL	TRUE
XOR_In1	BYTE	10
XOR_In2	BYTE	50
Out1	BYTE	56

➢ **变量 2**

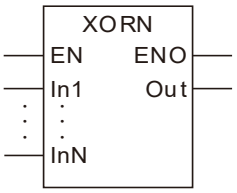
变量名	数据类型	当前值
XOR_EN	BOOL	TRUE
XOR_In1	BYTE	255
XOR_In2	WORD	256
Out1	WORD	511

➢ **程序**



8.10.5 XORN（同或）

FB/FC	说明	适用机种
FC	本指令用于将两个（或者多个）变量或常量作同或运算。	DVP15MC11T DVP15MC11T-06



- **参数**

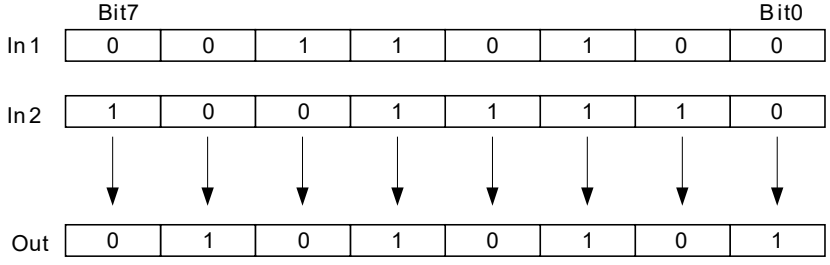
参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	运算参数	输入	程序编写时，可通过编程软件增加或减少运算参数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	运算结果	输出	In1 ~ InN作同或运算的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

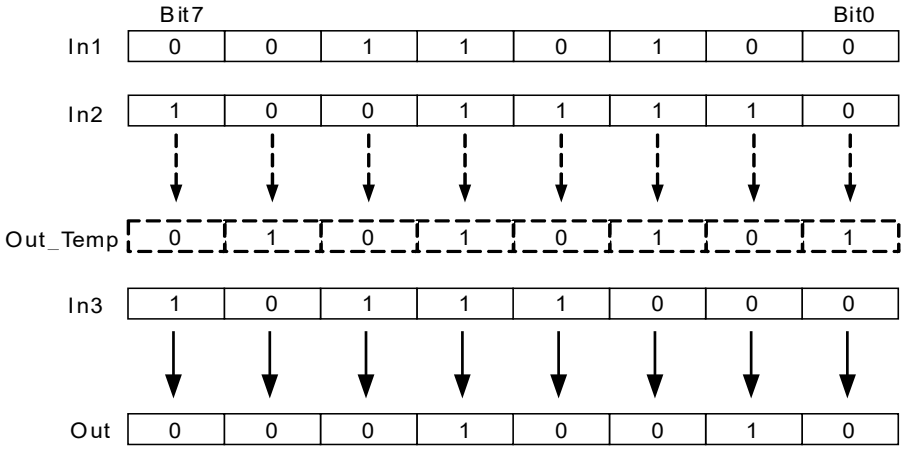
说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

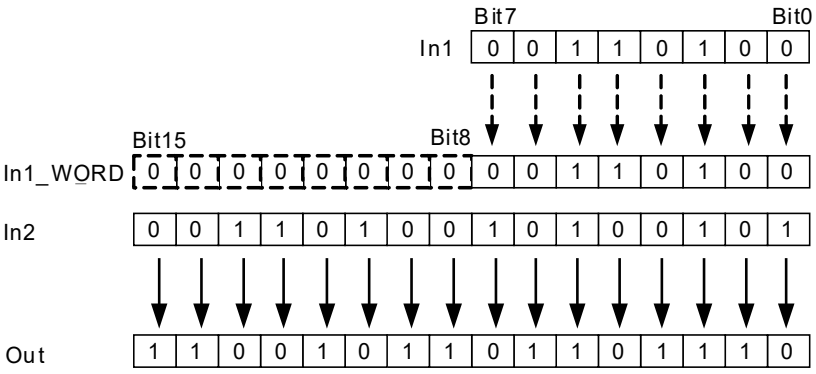
本指令用于将两个（或者多个）变量或常量作同或运算（按位同或），结果输出至*Out*，即*Out*= *In1* XORN I*n2* XORN…XORN *InN*。如下图所示，In1与In2作同或运算的规则：



- 当存在多个输入参数时，运算步骤为：*In1*与*In2*作同或的结果再与*In3*作同或运算，以此类推直至*InN*，如下图所示，In1与In2作同或运算的结果为Out_Temp,Out_Temp与In3作同或运算的结果为Out。



- 当输入变量的数据类型均不为BOOL时，允许*In1~InN*为不同数据类型的变量。当*In1~InN*为不同数据类型的变量时，以能包含*In1~InN*所有取值范围的数据类型进行运算，例如*In1*的数据类型为BYTE，*In2*的数据类型为WORD，则*Out*的数据类型为WORD；运算时将*In1*由BYTE转换成WORD（补齐位全为0，如下图中的Bit8~Bit15）后再与*In2*按位作同或运算，如下图所示：



- 若输入变量的数据类型为BOOL时，则要求所有输入和输出变量的数据类型均为BOOL，否则软件编译时将报错。

⚠ 注意

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8

程序范例

- 变量XORN _In1、XORN _In2、Out1的数据类型均为BYTE，XORN_In1与XORN_In2的值分别为10和50，当变量XORN_EN为TRUE时，Out1的值为199（如下表“变量1”所示）；当变量XORN_In1、XORN_In2和Out1的数据类型分别为BYTE、WORD和WORD，XORN_In1与XORN_In2的值分别为255和256，当变量XORN_EN为TRUE时，Out1的值为65535（如下表“变量2”所示）。

➢ **变量 1**

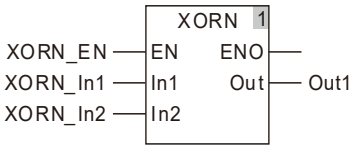
变量名	数据类型	当前值
XORN_EN	BOOL	TRUE

变量名	数据类型	当前值
XORN_In1	BYTE	10
XORN_In2	BYTE	50
Out1	BYTE	199

➢ **变量 2**

变量名	数据类型	当前值
XORN_EN	BOOL	TRUE
XORN_In1	BYTE	255
XORN_In2	WORD	256
Out1	WORD	65535

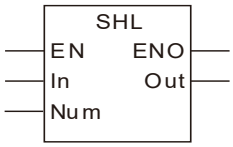
➢ **程序**



8.11 位移指令

8.11.1 SHL（向左移位）

FB/FC	说明	适用机种
FC	本指令用于将变量或常量中的所有位向左移动指定位数，结果输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

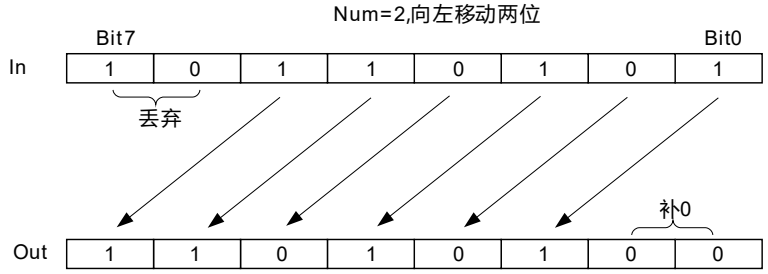
参数名称	含义	输入/输出	描述	参数取值范围
In	移位操作数	输入	用于向左移位操作的原始数据	由与输入参数所连变量的数据类型决定
Num	移位数	输入	向左移位的位数	由与输入参数所连变量的数据类型决定
Out	移位结果	输出	将原始数据的所有位向左移动Num位后的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	与*In*的数据类型一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将*In*中的所有位整体向左移动，移动位数为*Num*，并将移动后的结果输出至*Out*。如下图所示，当*Num*=2时，*In*中的所有位向左移动两位，其中Bit0~Bit1用0补充，Bit6~Bit7被舍弃。



注意

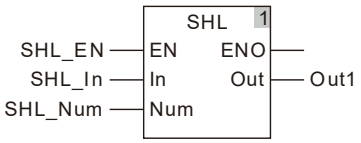
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- Num的值为0时，Out的值和 In的值相等。
- Num的值大于In的位数时，Out输出结果为0。

程序范例

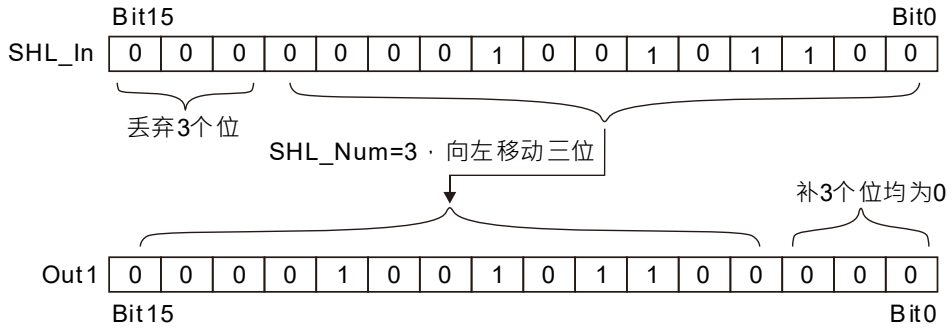
- 变量SHL_In与SHL_Num的数据类型分别为UINT和USINT，值分别为300和3，Out1的数据类型为UINT，当变量SHL_EN为TRUE时，Out1的值为2400。

  ➢ **变量和程序**

变量名	数据类型	当前值
SHL_EN	BOOL	TRUE
SHL_In	UINT	300
SHL_Num	USINT	3
Out1	UINT	2400

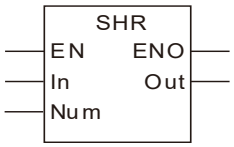


  ➢ **移位范例示意图**



8.11.2 SHR（向右移位）

FB/FC	说明	适用机种
FC	本指令用于将变量或常量中的所有位向右移动指定位数，结果输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	移位操作数	输入	用于向右移位操作的原始数据	由与输入参数所连变量的数据类型决定
Num	移位数	输入	向右移位的位数	由与输入参数所连变量的数据类型决定
Out	移位结果	输出	将原始数据的所有位向右移动Num位后的结果	由与输出参数所连变量的数据类型决定

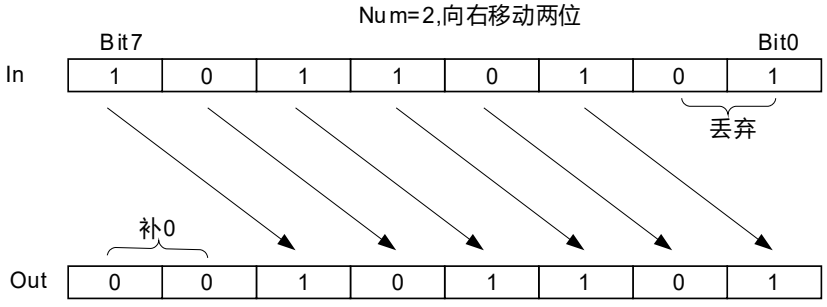
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	与*In*的数据类型一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将*In*中的所有位整体向右移动，移动位数为*Num*，并将移动后的结果输出至*Out*。

    如下图所示，当*Num*=2时，*In*中的所有位向右移动两位，其中Bit0~Bit1被舍弃，Bit6~Bit7补0。



注意

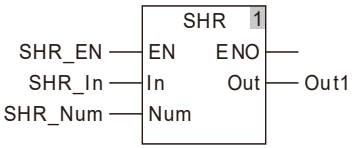
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- Num的值为0时，Out的值和 In的值相等。
- Num的值大于In的位数时，Out输出结果为0。

程序范例

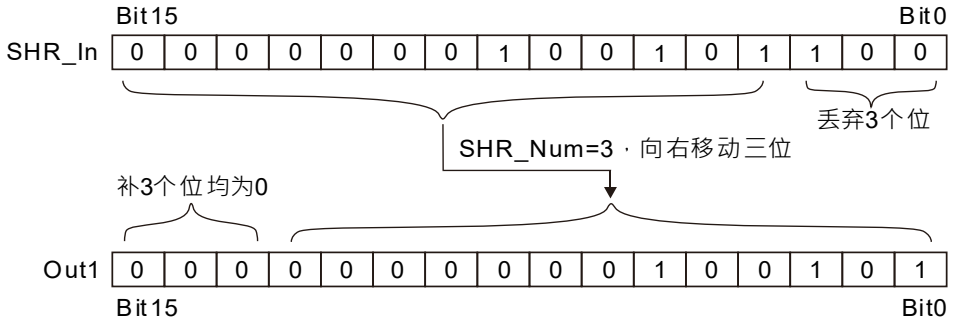
- 变量SHR_In与SHR_Num的数据类型分别为UINT和USINT，值分别为300和3，Out1的数据类型为UINT，当变量SHR_EN为TRUE时，Out1的值为37。

  ➢ **变量和程序**

变量名	数据类型	当前值
SHR_EN	BOOL	TRUE
SHR_In	UINT	300
SHR_Num	USINT	3
Out1	UINT	37

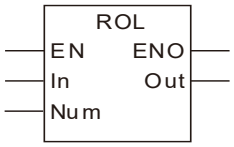


  ➢ **移位范例示意图**



8.11.3 ROL（向左循环移位）

FB/FC	说明	适用机种
FC	本指令用于将变量或常量中的所有位向左循环移动指定位数，结果输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	移位操作数	输入	用于向左循环移位操作的原始数据	由与输入参数所连变量的数据类型决定
Num	移位数	输入	向左循环移位的位数	由与输入参数所连变量的数据类型决定
Out	移位结果	输出	将原始数据的所有位向左循环移动Num位后的结果	由与输出参数所连变量的数据类型决定

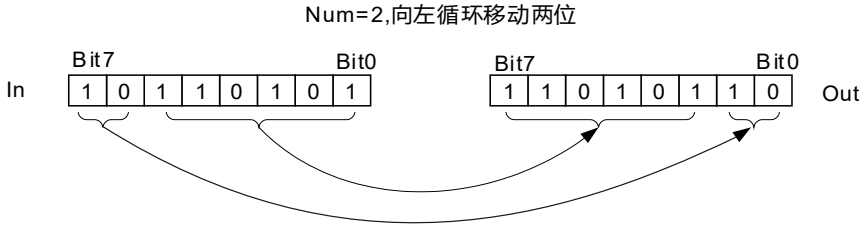
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	与*In*的数据类型一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将*In*中的所有位整体向左循环移动，移动位数为*Num*，并将移动后的结果输出至*Out*。

向左循环移动是指左边移出的位依次填充到右边空出的位中。如下图所示，当*Num*=2时，*In*中的所有位向左循环移动两位，移动方式为：Bit0~Bit5依次移动到Bit2~Bit7，Bit7填充到Bit1中，Bit6填充到Bit0中。



注意

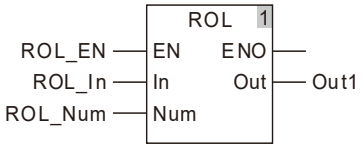
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- Num的值为0时，Out的值和 In的值相等。
- Num的值大于In的位数时，In的值向左移位的位数等于Num MOD In的值。例如：In是BYTE数据类型，Num=USINT#1和Num=USINT#9的Out值相等。

程序范例

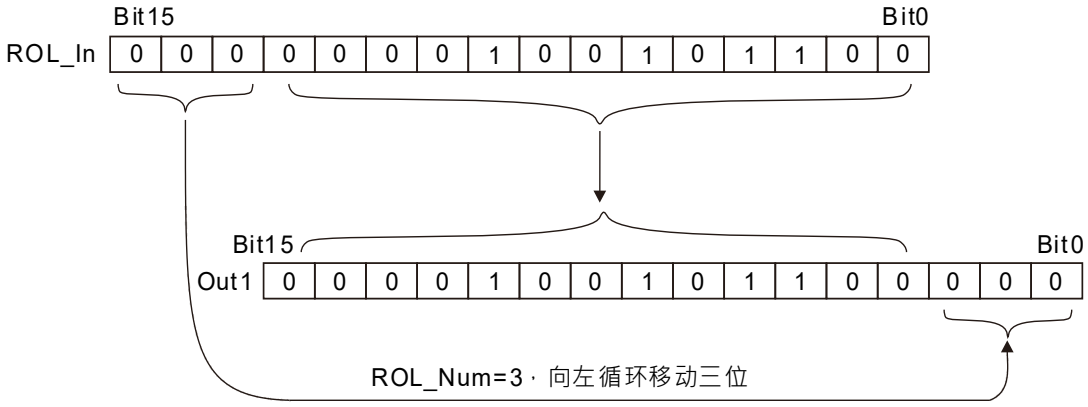
- 变量ROL_In与ROL_Num的数据类型分别为UINT和USINT，值分别为300和3，Out1的数据类型为UINT，当变量ROL_EN为TRUE时，Out1的值为2400。

  ➢ **变量和程序**

变量名	数据类型	当前值
ROL_EN	BOOL	TRUE
ROL_In	UINT	300
ROL_Num	USINT	3
Out1	UINT	2400

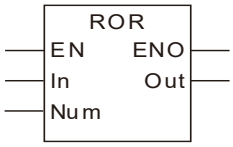


  ➢ **移位范例示意图**



8.11.4 ROR（向右循环移位）

FB/FC	说明	适用机种
FC	本指令用于将变量或常量中的所有位向右循环移动指定位数，结果输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	移位操作数	输入	用于向右循环移位操作的原始数据	由与输入参数所连变量的数据类型决定
Num	移位数	输入	向右循环移位的位数	由与输入参数所连变量的数据类型决定
Out	移位结果	输出	将原始数据的所有位向右循环移动Num位后的结果	由与输出参数所连变量的数据类型决定

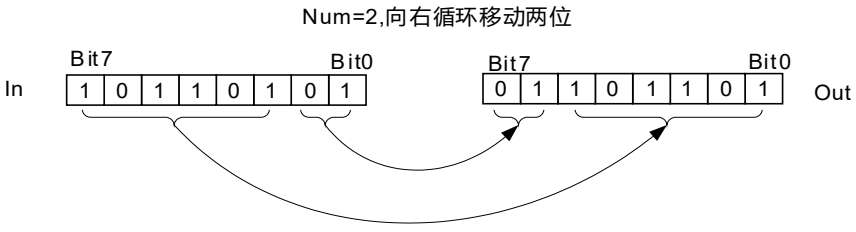
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	与*In*的数据类型一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将*In*中的所有位整体向右循环移动，移动位数为*Num*，并将移动后的结果输出至*Out*。

向右循环移动是指右边移出的位依次填充到左边空出的位中。如下图所示，当*Num*=2时，*In*中的所有位向左循环移动两位，移动方式为：Bit2~Bit7依次移动到Bit0~Bit5中，Bit0填充到Bit6中，Bit1填充到Bit7中。



注意

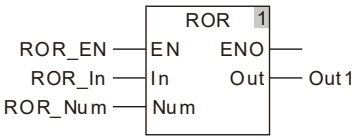
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- Num的值为0时，Out的值和 In的值相等。
- Num的值大于In的位数时，IN的值向右移位的位数等于Num MOD In的值。例如：In是BYTE数据类型，Num=USINT#1和Num=USINT#9的Out值相等。

程序范例

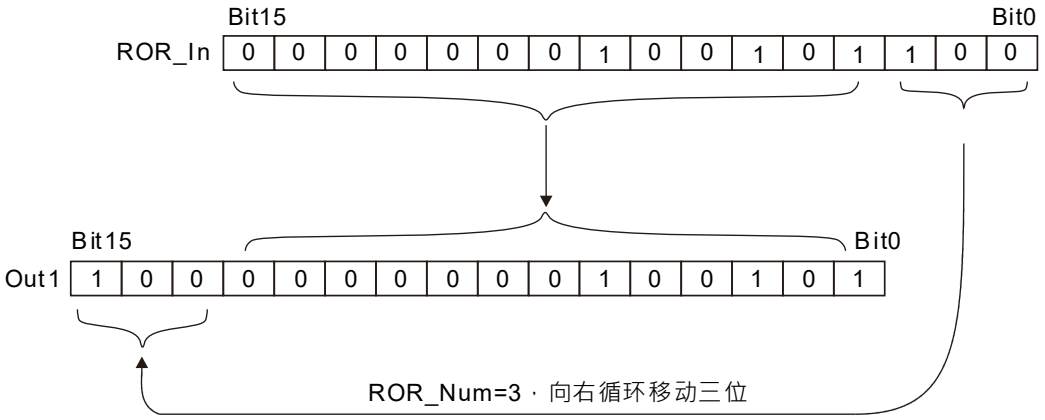
- 变量ROR_In与 ROR_Num的数据类型分别为UINT和USINT，值分别为300和3，Out1的数据类型为UINT，当变量ROR_EN为TRUE时，Out1的值为32805。

  ➢ **变量和程序**

变量名	数据类型	当前值
ROR_EN	BOOL	TRUE
ROR_In	UINT	300
ROR_Num	USINT	3
Out1	UINT	32805



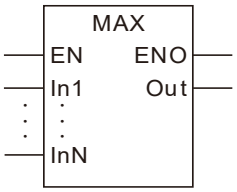
  ➢ **移位范例示意图**



8.12 选择指令

8.12.1 MAX（求最大值）

FB/FC	说明	适用机种
FC	本指令用于求取两个（或者多个）变量或常量的最大值。	DVP15MC11T DVP15MC11T-06



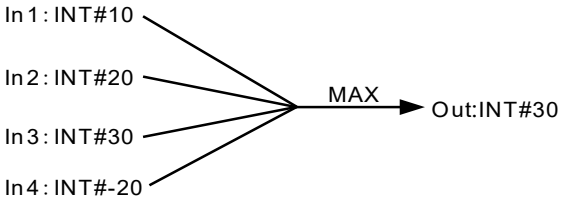
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数，但比较数最多为8个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	最大值	输出	In1 ~ InN中的最大值	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于获取两个（或者多个）变量或常量中的最大值，并将该最大值输出至*Out*。



  - 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1~InN*为不同数据类型的变量。

- 当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入变量和输出变量均为该数据类型。如*In1*的数据类型为TIME，则 *In2~InN* 的数据类型必须为TIME，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型的宽度须包含所有输入参数的宽度，否则软件编译时将报错。

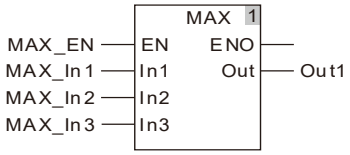
程序范例

- **变量MAX_In1、MAX_In2、MAX_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为DINT。**

当MAX_In1、MAX_In2、MAX_In3的值分别为 - 10、50、100，当变量MAX_EN为TRUE时，Out1的值为100。

➢ **变量和程序**

变量名	数据类型	当前值
MAX_EN	BOOL	TRUE
MAX_In1	INT	- 10
MAX_In2	UINT	50
MAX_In3	DINT	100
Out1	DINT	100

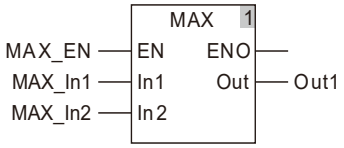


- **变量MAX_In1、MAX_In2的数据类型均为TIME，Out1的数据类型为TIME。**

当MAX_In1、MAX_In2的值分别为T#1ms、T#50ms，当变量MAX_EN为TRUE时，Out1的值为T#50ms：

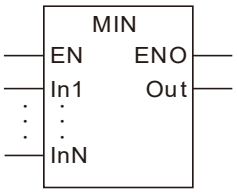
➢ **变量和程序**

变量名	数据类型	当前值
MAX_EN	BOOL	TRUE
MAX_In1	TIME	T#1ms
MAX_In2	TIME	T#50ms
Out1	TIME	T#50ms



8.12.2 MIN（求最小值）

FB/FC	说明	适用机种
FC	本指令用于求取两个（或者多个）变量或常量的最小值。	DVP15MC11T DVP15MC11T-06



- **参数**

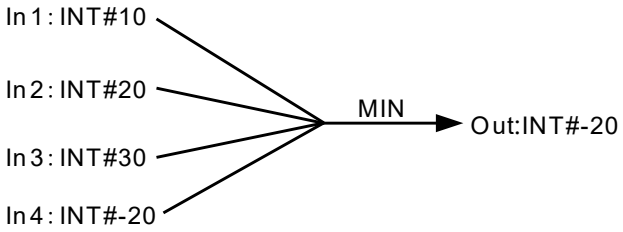
参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	比较数	输入	程序编写时，可通过编程软件增加或减少比较数，但比较数最多为8个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	最大值	输出	In1 ~ InN中的最大值	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  ■ 本指令用于获取两个（或者多个）变量或常量中的最小值，并将该最小值输出至*Out*。



  ■ 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In1~InN*为不同数据类型的变量。

- 当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入变量和输出变量均为该数据类型。如*In1*的数据类型为TIME，则*In2~InN*的数据类型必须为TIME，否则软件编译时将报错。

⚠ 注意

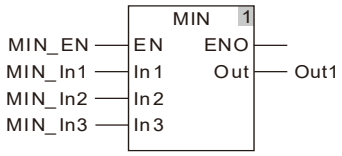
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量的数据类型的宽度须包含所有输入参数的宽度，否则软件编译时将报错。

程序范例

- 变量MIN_In1、MIN_In2、MIN_In3的数据类型分别为INT、UINT、DINT，Out1的数据类型为DINT。当MIN_In1、MIN_In2、MIN_In3的值分别为 - 10、50、100，当变量MIN_EN为TRUE时，Out1的值为 - 10：

  ➢ **变量和程序**

变量名	数据类型	当前值
MIN_EN	BOOL	TRUE
MIN_In1	INT	- 10
MIN_In2	UINT	50
MIN_In3	DINT	100
Out1	DINT	- 10

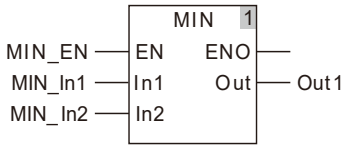


- 变量MIN_In1、MIN_In2的数据类型均为TIME，Out1的数据类型为TIME。

  当MIN_In1、MIN_In2的值分别为T#1ms、T#50ms，当变量MIN_EN为TRUE时，Out1的值为T#1ms：

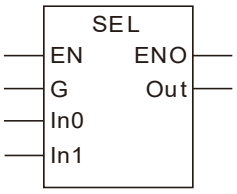
  ➢ **变量和程序**

变量名	数据类型	当前值
MIN_EN	BOOL	TRUE
MIN_In1	TIME	T#1ms
MIN_In2	TIME	T#50ms
Out1	TIME	T#1ms



8.12.3 SEL（二选一）

FB/FC	说明	适用机种
FC	本指令用于选取两个变量或常量的某一个，并将该值输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

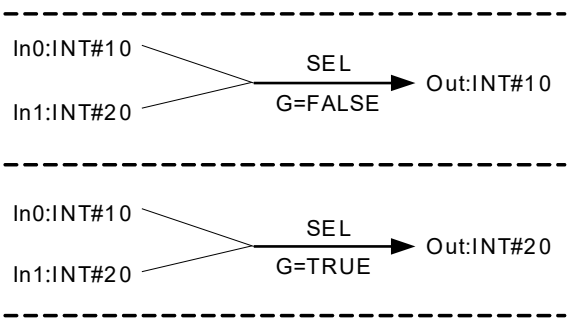
参数名称	含义	输入/输出	描述	参数取值范围
G	选择条件	输入	当G为FALSE时，选择In0 当G为TRUE时，选择In1	由与输入参数所连变量的数据类型决定
In0和In1	待选择数	输入	待选择数	由与输入参数所连变量的数据类型决定
Out	选择结果	输出	选择结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
G	●																			
In0 和 In1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  ■ 本指令用于选取两个变量或常量中的某一个，并将该值输出至*Out*，选择条件为*G*。



- 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数*In0~In1*所连变量为不同数据类型。
- 当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入变量和输出变量均为该数据类型。如*In0*所连变量的数据类型为TIME，则*In1*和*Out*所连变量的数据类型必须为TIME，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量数据类型的宽度须包含*In0*和*In1*所连变量的宽度，否则软件编译时将报错。

程序范例

- 变量SEL_G、SEL_In0、SEL_In1的数据类型分别为BOOL、UINT、DINT，Out1的数据类型为DINT。

当变量SEL_EN为TRUE时，若SEL_G、SEL_In0、SEL_In1的值分别为FALSE、50、100，则Out1的值为50（如下表“变量1”所示）；若SEL_G、SEL_In0、SEL_In1的值分别为TRUE、50、100，则Out1的值为100（如下表“变量2”所示）；

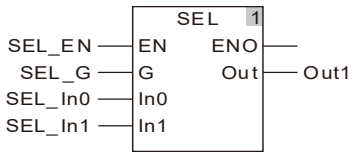
➢ **变量 1**

变量名	数据类型	当前值
SEL_EN	BOOL	TRUE
SEL_G	BOOL	FALSE
SEL_In0	UINT	50
SEL_In1	DINT	100
Out1	DINT	50

➢ **变量 2**

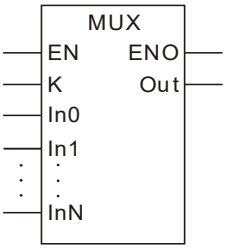
变量名	数据类型	当前值
SEL_EN	BOOL	TRUE
SEL_G	BOOL	TRUE
SEL_In0	UINT	50
SEL_In1	DINT	100
Out1	DINT	100

➢ **程序**



8.12.4 MUX（多选一）

FB/FC	说明	适用机种
FC	本指令用于选取两个（或者多个）变量或常量的某一个，并将该值输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
K	选择条件	输入	选择条件	由与输入参数所连变量的数据类型决定
In0、In1至InN	待选择数	输入	程序编写时，可通过编程软件增加或减少待选择数，但带选择数最多为8个，即N =2 ~ 8	由与输入参数所连变量的数据类型决定
Out	选择结果	输出	选择结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
K						●														
In0、In1至InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于从*In0~InN*中选取一个，并其输出至*Out*，选择条件为*K*。*Out*的取值与*K*的关系如下：

K取值	Out取值
0	In0
1	In1
2	In2

K取值	Out取值
3	In3
4	In4
5	In5
6	In6
7	In7

- 当输入变量的数据类型不为BOOL、TIME、DATE、TOD、STRING时，允许输入参数In0~InN所连变量为不同数据类型。
- 当输入变量的数据类型为BOOL、TIME、DATE、TOD、STRING中的一种时，要求输入变量和输出变量均为该数据类型。如In0所连变量的数据类型为TIME，则In1~InN和Out所连变量的数据类型必须为TIME，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 输出变量数据类型的宽度须包含*In0*至*InN*所连变量的宽度，否则软件编译时将报错。

程序范例

- 变量MUX_K、MUX_In0、MUX_In1的数据类型分别为UINT、UINT、DINT，Out1的数据类型为DINT。

当变量MUX_EN为TRUE时，若MUX_K、MUX_In0、MUX_In1的值分别为0、50、100，则Out1的值为50（如下表“变量1”所示）；若MUX_K、MUX_In0、MUX_In1的值分别为1、50、100，则Out1的值为100（如下表“变量2”所示）。

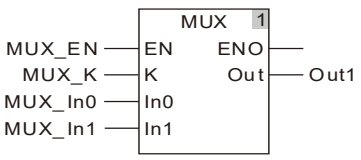
➢ **变量 1**

变量名	数据类型	当前值
MUX_EN	BOOL	TRUE
MUX_K	USINT	0
MUX_In0	UINT	50
MUX_In1	DINT	100
Out1	DINT	50

➢ **变量 2**

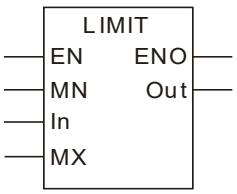
变量名	数据类型	当前值
MUX_EN	BOOL	TRUE
MUX_K	UINT	1
MUX_In0	UINT	50
MUX_In1	DINT	100
Out1	DINT	100

➢ **程序**



8.12.5 LIMIT（输出限制）

FB/FC	说明	适用机种
FC	本指令用于将输出值限制在某一区间内。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
MN	最小值	输入	最小值	由与输入参数所连变量的数据类型决定
In	被限制的数	输入	被限制的数	由与输入参数所连变量的数据类型决定
MX	最大值	输入	最大值	由与输入参数所连变量的数据类型决定
Out	执行限制后的结果	输出	执行限制后的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
MX		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将*Out* 的值限制在MN和MX之间，*Out* 的取值如下表所示：

In取值	Out取值
In < MN	MN
MN ≤ In ≤MX	In
MX < In	MX

  - 本指令允许输入参数MN、In、MX所连变量为不同数据类型，当“MN”、“ In”、“MX”为不同数据类型的变量时，以能包含“MN”、“In”、“MX”所有取值范围的数据类型进行运算。例如“MN”的数据类型为INT，“In”、“MX”的数据类型为DINT，则“Out”的数据类型为DINT。

- 本指令允许输入参数和输出参数所连变量为不同数据类型，但输出变量数据类型的宽度须包含输入参数所连变量的宽度，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量LIMIT_MN、LIMIT_In、LIMIT_MX的数据类型分别为UINT、UINT、DINT，Out1的数据类型为DINT。

当变量LIMIT_EN为TRUE时，若LIMIT_MN、LIMIT_In、LIMIT_MX的值分别为1、50、100，则Out1的值为50（如下表“变量1”所示）；若LIMIT_MN、LIMIT_In、LIMIT_MX的值分别为2、200、100，则Out1的值为100（如下表“变量2”所示）；若LIMIT_MN、LIMIT_In、LIMIT_MX的值分别为50、10、100，则Out1的值为50（如下表“变量3”所示）。

➢ **变量 1**

变量名	数据类型	当前值
LIMIT_EN	BOOL	TRUE
LIMIT_MN	UINT	1
LIMIT_In	UINT	50
LIMIT_MX	DINT	100
Out1	DINT	50

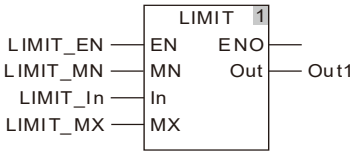
➢ **变量 2**

变量名	数据类型	当前值
LIMIT_EN	BOOL	TRUE
LIMIT_MN	UINT	2
LIMIT_In	UINT	200
LIMIT_MX	DINT	100
Out1	DINT	100

➢ **变量 3**

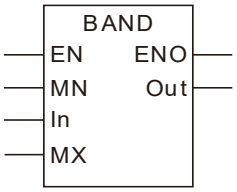
变量名	数据类型	当前值
LIMIT_EN	BOOL	TRUE
LIMIT_MN	UINT	50
LIMIT_In	UINT	10
LIMIT_MX	DINT	100
Out1	DINT	50

➢ **程序**



8.12.6 BAND（死区限制）

FB/FC	说明	适用机种
FC	本指令用于将输入数据作死区限制处理，并将处理后的结果输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
MN	最小值	输入	最小值	由与输入参数所连变量的数据类型决定
In	被限制的数	输入	被限制的数	由与输入参数所连变量的数据类型决定
MX	最大值	输入	最大值	由与输入参数所连变量的数据类型决定
Out	执行限制后的结果	输出	执行限制后的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN														●	●					
In														●	●					
MX														●	●					
Out														●	●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将*In*的值作死区限制处理，并将处理后的结果输出*Out*。死区范围为*MN~MX*，限制规则为如下表所示：

In取值	Out取值
In < MN	In - MN
MN ≤In ≤MX	0
MX < In	In - MX

  - 本指令允许输入参数MN、In、MX所连变量为不同数据类型，当“MN”、“ In”、“MX”为不同数据类型的变量时，以能包含“MN”、“In”、“MX”所有取值范围的数据类型进行运算。例如“MN”的数据类型为REAL，“In”、“MX”的数据类型为LREAL，则“Out”的数据类型为LREAL。

- 本指令允许输入参数和输出参数所连变量为不同数据类型，但输出变量数据类型的宽度须包含输入参数所连变量的宽度，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 当MN的值大于MX的值时，指令正常执行，Out取值为MX的值。

程序范例

- 变量BAND_MN、BAND_In、BAND_MX的数据类型均为REAL，Out1的数据类型为LREAL。

当变量BAND_EN为TRUE时，若BAND_MN、BAND_In、BAND_MX的值分别为1、50、100，则Out1的值为0（如下表“变量1”所示）；若BAND_MN、BAND_In、BAND_MX的值分别为2、250、100，则Out1的值为150（150=250-100）（如下表“变量2”所示）；若BAND_MN、BAND_In、BAND_MX的值分别为50、10、100，则Out1的值为- 40 （ - 40 = 10 – 50 ）（如下表“变量3”所示）；

➢ **变量 1**

变量名	数据类型	当前值
BAND_EN	BOOL	TRUE
BAND_MN	REAL	1
BAND_In	REAL	50
BAND_MX	REAL	100
Out1	LREAL	0

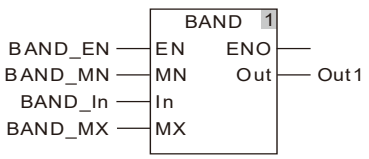
➢ **变量 2**

变量名	数据类型	当前值
BAND_EN	BOOL	TRUE
BAND_MN	REAL	2
BAND_In	REAL	250
BAND_MX	REAL	100
Out1	LREAL	150

➢ **变量 3**

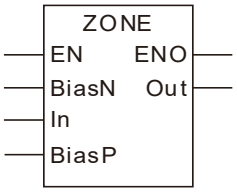
变量名	数据类型	当前值
BAND_EN	BOOL	TRUE
BAND_MN	REAL	50
BAND_In	REAL	10
BAND_MX	REAL	100
Out1	LREAL	-40

➢ **程序**



8.12.7 ZONE（输入偏移）

FB/FC	说明	适用机种
FC	本指令用于将输入数据作指定的偏移处理，并将处理后的结果输出至 *Out*。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
BiasN	负数偏移值	输入	负数偏移值	由与输入参数所连变量的数据类型决定
In	原始数据	输入	原始数据	由与输入参数所连变量的数据类型决定
BiasP	正数偏移值	输入	正数偏移值	由与输入参数所连变量的数据类型决定
Out	执行偏移后的结果	输出	执行偏移后的结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
BiasN														●	●					
In														●	●					
BiasP														●	●					
Out														●	●					

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于将*In*的值按设定的偏移值作偏移处理，并将处理后的结果输出*Out*。*In*为负数时的偏移值为*BiasN*，为正数时，偏移值为*BiasP*，偏移规则为如下表所示：

In取值	Out取值
In<0	In+BiasN
In=0	0
In>0	In+BiasP

- 本指令允许输入参数*BiasN*、*In*、*BiasP*所连变量为不同数据类型，当“*BiasN*”、“*In*”、“*BiasP*”为不同数据类型的变量时，以能包含“*BiasN*”、“*In*”、“*BiasP*”所有取值范围的数据类型进行运算。例如“*BiasN*”的数据类型为REAL，“In”、“*BiasP*”的数据类型为LREAL，则“*Out*”的数据类型为LREAL。
- 本指令允许输入参数和输出参数所连变量为不同数据类型，但输出变量数据类型的宽度须包含输入参数所连变量的宽度，否则软件编译时将报错。

⚠ 注意

- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。
- 当*BiasN*的值大于BiasP的值时，指令正常执行。

程序范例

- 变量ZONE_BiasN、ZONE_In、ZONE_BiasP的数据类型分别为REAL、REAL、LREAL，Out1的数据类型为LREAL。

当变量ZONE_EN为TRUE时，若ZONE_BiasN、ZONE_In、ZONE_BiasP的值分别为1.0、0.0、100.0，则Out1的值为0(如下表“变量1”所示)；若ZONE_BiasN、ZONE_In、ZONE_BiasP的值分别为2.0、50.0、100.0，则Out1的值为150.0 （ 150.0 = 50.0 + 100.0 ）（如下表“变量2”所示)；若ZONE_BiasN、ZONE_In、ZONE_BiasP的值分别为50.0、- 10.0、100.0，则Out1的值为 40.0 （ 40.0 = - 10.0 + 50.0 ）(如下表“变量3”所示)；

➢ **变量 1**

变量名	数据类型	当前值
ZONE_EN	BOOL	TRUE
ZONE_BiasN	REAL	1.0
ZONE_In	REAL	0.0
ZONE_BiasP	LREAL	100.0
Out1	LREAL	0.0

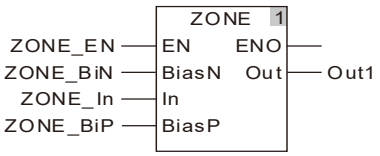
➢ **变量 2**

变量名	数据类型	当前值
ZONE_EN	BOOL	TRUE
ZONE_BiasN	REAL	2.0
ZONE_In	REAL	50.0
ZONE_BiasP	LREAL	100.0
Out1	LREAL	150.0

➢ **变量 3**

变量名	数据类型	当前值
ZONE_EN	BOOL	TRUE
ZONE_BiasN	REAL	50.0
ZONE_In	REAL	- 10.0
ZONE_BiasP	LREAL	100.0
Out1	LREAL	40.0

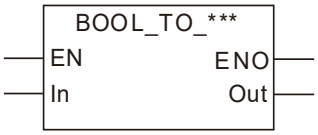
➢ **程序**


ZONE 1
ZONE_EN
EN
ENO
ZONE_BiN
BiasN
Out
Out1
ZONE_In
In
ZONE_BiP
BiasP

8.13 数据类型转换指令

8.13.1 BOOL_TO_***（布尔转换指令群）

FB/FC	说明	适用机种
FC	BOOL_TO_***为一组指令，用于将 BOOL 型的数据转换成基本数据类型的数据。"***" 允许为所有基本数据类型。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	用于转换的数据	输入	用于转换的数据	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	转换结果	由与输出参数所连变量的数据类型决定

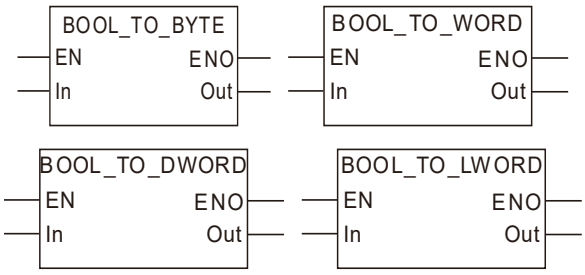
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
Out		与指令名称中的 "***" 一致																		

说明：上表中的 "●" 表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - **BOOL转换为位串**

    ➢ 相关指令如下所示：

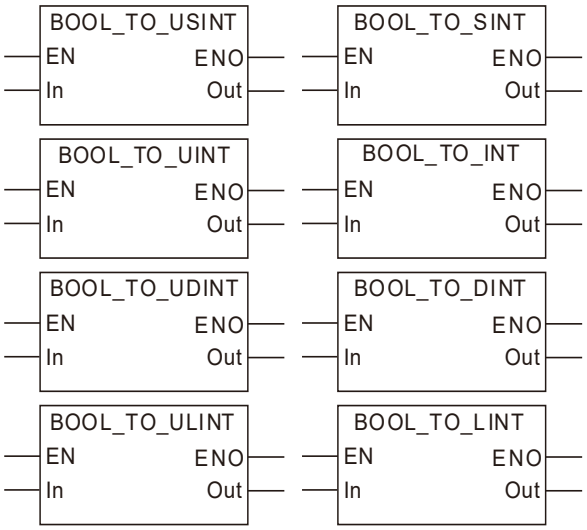


➢ 将BOOL型数据转换为位串型数据的规则如下表：(位串的值的格式以及16进制的表示方法待定)

布尔	位串			
	BYTE	WORD	DWORD	LWORD
FALSE	16#00	16#0000	16#0000_0000	16#0000_0000_0000_0000
TRUE	16#01	16#0001	16#0000_0001	16#0000_0000_0000_0001

■ **BOOL转换为整数**

➢ 相关指令如下所示：

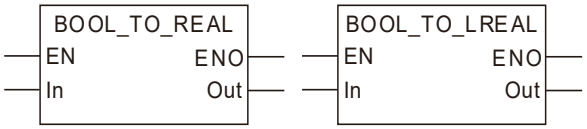


➢ 将BOOL型数据转换为整数型数据的规则如下表：

布尔	整数							
	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT
FALSE	0	0	0	0	0	0	0	0
TRUE	1	1	1	1	1	1	1	1

■ **BOOL转换为实数**

➢ 相关指令如下所示：

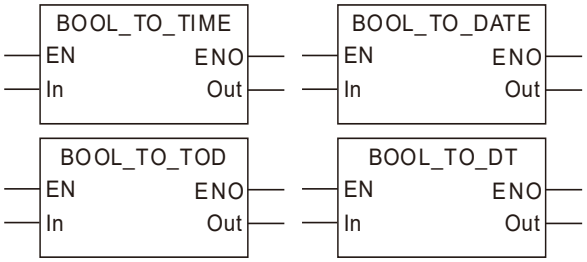


➢ 将BOOL型数据转换为实数型数据的规则如下表：

布尔	实数	
	REAL	LREAL
FALSE	0	0
TRUE	1	1

■ **BOOL转换为时间或日期**

➢ 相关指令如下所示：

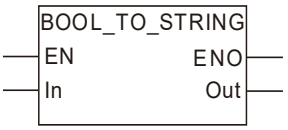


➢ 将BOOL型数据转换为时间或日期型数据的规则如下表：(下表中时间、日期的值格式待定)

布尔	时间、日期			
	TIME	DATE	TOD	DT
FALSE	T#0ms	D#1970-1-1	TOD#0:0:0.000	DT#1970-01-01-00:00:00
TRUE	T#0ms	D#1970-1-1	TOD#0:0:0.001	DT#1970-01-01-00:00:01

■ **BOOL转换为字符串**

➢ 相关指令如下所示：



➢ 将BOOL型数据转换为字符串型数据的规则如下表：(下表中字符串格式待定)

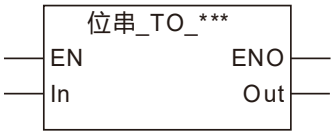
布尔	字符串
	STRING
FALSE	‘FALSE’
TRUE	‘TRUE’

⚠ **注意**

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8.13.2 Bit strings_TO_***（位串转换指令群）

FB/FC	说明	适用机种
FC	位串_TO_***为一组指令，用于将位串型的数据转换成基本数据类型的数据。 “***”允许所有基本数据类型。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	用于转换的数据	输入	用于转换的数据	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	转换结果	由与输出参数所连变量的数据类型决定

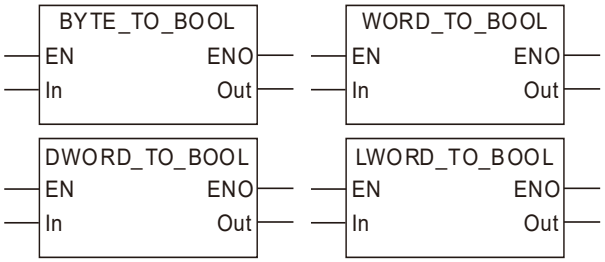
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●															
Out	与指令名称中的“***”一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - **位串转换为BOOL**

    - 相关指令如下所示：



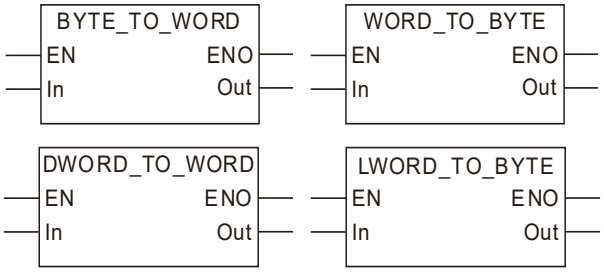
    - 将位串型数据转换为BOOL型数据的规则如下表：

数据类型		取值对应关系	
In	Out	In	Out
BYTE	BOOL	16#00	FALSE
		16#01~16#FF	TRUE
WORD	BOOL	16#0000	FALSE
		16#0001~16#FFFF	TRUE

数据类型		取值对应关系	
In	Out	In	Out
DWORD	BOOL	16#0000_0000	FALSE
		16#0000_0001~16#FFFF_FFFF	TRUE
LWORD	BOOL	16#0000_0000_0000_0000	FALSE
		16#0000_0000_0000_0001~ 16#FFFF_FFFF_FFFF_FFFF	TRUE

■ **位串转换为位串**

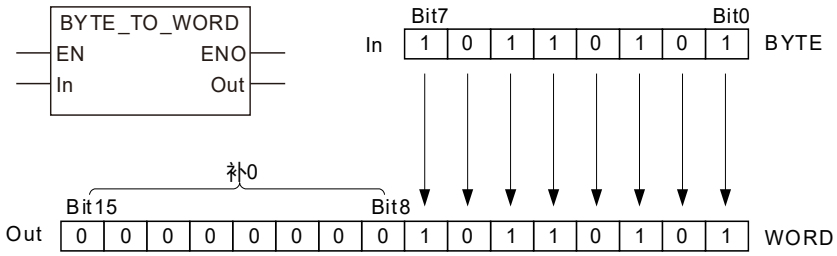
➢ 位串与位串之间可以相互转换，部分指令如下所示：



不同类型的位串数据相互转换时，分为宽度小的数据转换为宽度大的数据和宽度大的数据转换为宽度小的数据两种情形，它们的转换规则以及差别如下所示。

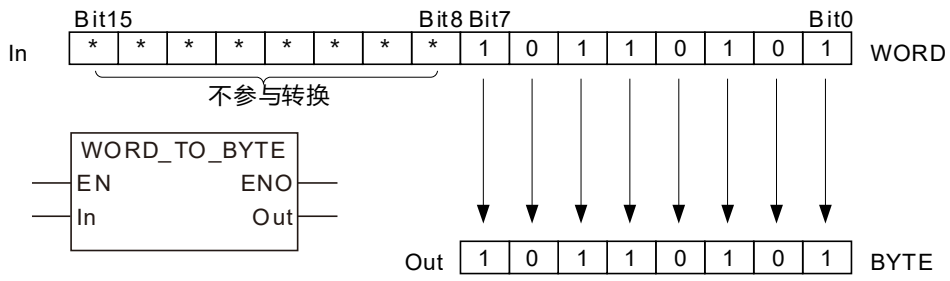
将宽度小的数据转换为宽度大的数据时，将宽度小的数据中的所有位的值复制到宽度大的数据的对应位中，再将宽度大的数据中剩余的位全部设置为0。如下图范例中，将BYTE型数据In转换为WORD型数据Out，转换时，将In的Bit0~Bit7复制到Out的Bit0~Bit7，且Out的Bit8~Bit15均被设置为0。

宽度小的数据转换为宽度大的数据



将宽度大的数据转换为宽度小的数据时，将宽度小的数据中的所有位的值修改为宽度大的数据的对应位的值，宽度大的数据中剩余的位不参与转换，也即是其取值如何对转换无影响。如下图范例中，将WORD型数据In转换为BYTE型数据Out，转换时，将In的Bit0~Bit7复制到Out的Bit0~Bit7，且In的Bit8~Bit15均不参与转换，对转换无影响，不需考虑。

宽度大的数据转换为宽度小的数据

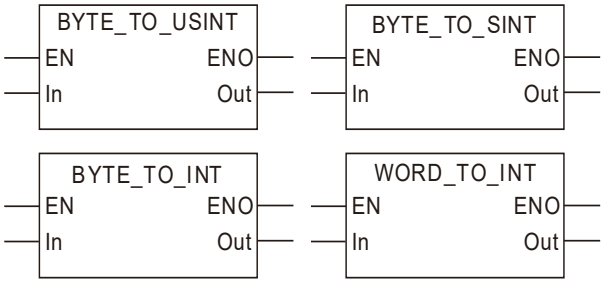


➢ 位串与位串之间相互转换的取值对应关系如下表：

数据类型		取值对应关系	
In	Out	In	Out
BYTE	WORD	16#00~16#FF	16#0000~16#00FF
	DWORD	16#00~16#FF	16#0000_0000~16#0000_00FF
	LWORD	16#00~16#FF	16#0000_0000_0000_0000~ 16#0000_0000_0000_00FF
WORD	BYTE	16#**00~16#**FF	16#00~16#FF
	DWORD	16#0000~16#FFFF	16#0000_0000~16#0000_FFFF
	LWORD	16#0000~16#FFFF	16#0000_0000_0000_0000~ 16#0000_0000_0000_FFFF
DWORD	BYTE	16#****_**00~16#****_**FF	16#00~16#FF
	WORD	16#****_0000~16#****_FFFF	16#0000~16#FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000_0000_0000~ 16#0000_0000_FFFF_FFFF
LWORD	BYTE	16#****_****_****_**00~ 16#****_****_****_**FF	16#00~16#FF
	WORD	16#****_****_****_0000~ 16#****_****_****_FFFF	16#0000~16#FFFF
	DWORD	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	16#0000_0000~16#FFFF_FFFF

■ **位串转换为整数**

➢ 位串与整数之间可以相互转换，部分指令如下所示：



位串型数据转换为整数时的转换规则与位串与位串之间的转换规则一致。

将宽度小的数据转换为宽度大的数据时，将宽度小的数据中的所有位的值复制到宽度大的数据的对应位中，再将宽度大的数据中剩余的位全部设置为0。

将宽度大的数据转换为宽度小的数据时，将宽度小的数据中的所有位的值修改为宽度大的数据的对应位的值，宽度大的数据中剩余的位不参与转换，也即是其取值如何对转换无影响。

若宽度相等时，则将*In*中的所有位的值复制到*Out*中的对应位。

➢ 位串转换为整数时的取值对应关系如下表：

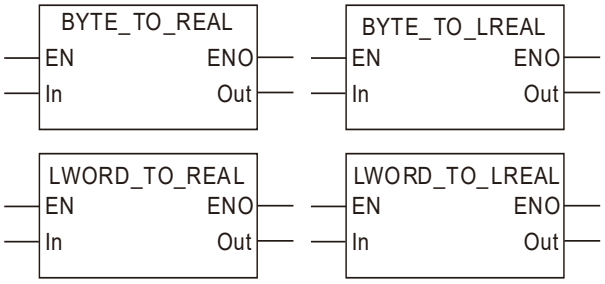
数据类型		取值对应关系	
In	Out	In	Out
BYTE	USINT	16#00~16#FF	0~255
	UINT	16#00~16#FF	0~255
	UDINT	16#00~16#FF	0~255

数据类型		取值对应关系	
In	Out	In	Out
	ULINT	16#00~16#FF	0~255
	SINT	16#00~16#7F	0~127
		16#80~16#FF	-128~-1
	INT	16#00~16#FF	0~255
	DINT	16#00~16#FF	0~255
	LINT	16#00~16#FF	0~255
WORD	USINT	16#**00~16#**FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000~16#FFFF	0~65535
	ULINT	16#0000~16#FFFF	0~65535
	SINT	16#**00~16#**7F	0~127
		16#**80~16#**FF	-128~-1
	INT	16#0000~16#7FFF	0~32767
		16#8000~16#FFFF	-32768~-1
	DINT	16#0000~16#FFFF	0~65535
	LINT	16#0000~16#FFFF	0~65535
DWORD	USINT	16#****_**00~16#****_**FF	0~255
	UINT	16#****_0000~16#****_FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	SINT	16#****_**00~16#****_**7F	0~127
		16#****_**80~16#****_**FF	-128~-1
	INT	16#****_0000~16#****_7FFF	0~32767
		16#****_8000~16#****_FFFF	-32768~-1
	DINT	16#0000_0000~16#7FFF_FFFF	0~2147483647
		16#8000_0000~16#FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
LWORD	USINT	16#****_****_****_**00~ 16#****_****_****_**FF	0~255
	UINT	16#****_****_****_0000~ 16#****_****_****_FFFF	0~65535
	UDINT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#****_****_****_**00~ 16#****_****_****_**7F	0~127
		16#****_****_****_**80~ 16#****_****_****_**FF	-128~-1
	INT	16#****_****_****_0000~ 16#****_****_****_7FFF	0~32767
		16#****_****_****_8000~ 16#****_****_****_FFFF	-32768~-1
	DINT	16#****_****_0000_0000~ 16#****_****_7FFF_FFFF	0~2147483647

数据类型		取值对应关系	
In	Out	In	Out
		16#****_****_8000_0000~ 16#***_*****_FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000_0000_0000~ 16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807
		16#8000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~0

■ **位串转换为实数**

➢ 位串可以转换为实数，部分指令如下所示：

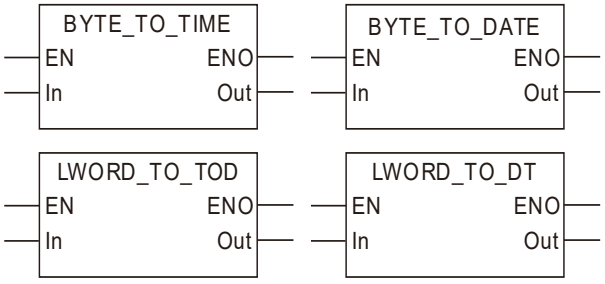


➢ 位串转换为实数的取值对应关系如下表：

数据类型		取值对应关系	
In	Out	In	Out
BYTE	REAL	16#00~16#FF	0~2.55e+2
	LREAL	16#00~16#FF	0~2.55e+2
WORD	REAL	16#0000~16#FFFF	0~6.5535e+4
	LREAL	16#0000~16#FFFF	0~6.5535e+4
DWORD	REAL	16#0000_0000~ 16#FFFF_FFFF	0~4.294967e+9
	LREAL	16#0000_0000~ 16#FFFF_FFFF	0~4.294967295e+9
LWORD	REAL	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	0~1.844674e+19
	LREAL	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	0~1.84467440737095e+19

■ **位串转换为时间或日期**

➢ 位串可以转换为时间或日期，部分指令如下所示：



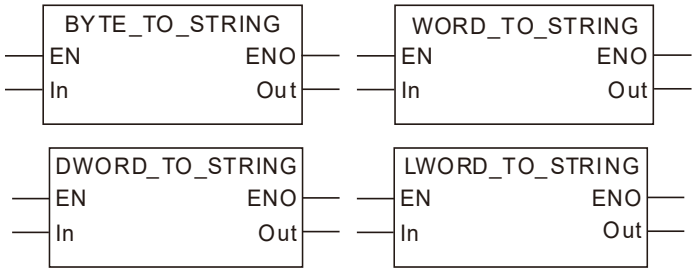
位串转换为时间或日期的规则与位串转换为无符号整数的规则一致。

➢ 位串转换为时间或日期的取值对应关系如下表：

数据类型		取值对应关系	
In	Out	In	Out
BYTE	TIME	16#00~16#FF	T#0ns~T#255ns
	DATE	16#00~16#FF	D#1970-1-1
	TOD	16#00~16#FF	TOD#0:0:0~ TOD#0:0:0.255
	DT	16#00~16#FF	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15
WORD	TIME	16#0000~16#FFFF	T#0ns~T#65us535ns
	DATE	16#0000~16#FFFF	D#1970-1-1
	TOD	16#0000~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	16#0000~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15
DWORD	TIME	16#0000_0000~16#FFFF_FFFF	T#0ns~ T#4s294ms967us295ns
	DATE	16#0000_0000~16#FFFF_FFFF	D#1970-1-1~D#2016-2-7
	TOD	16#0000_0000~16#0526_5BFF	TOD#0:0:0~ TOD#23:59:59.999
		16#0526_5C00~16#0A4C_B7FF	
		…….	
		16#FC57_9C00~16#FFFF_FFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#0000_0000~16#FFFF_FFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15
LWORD	TIME	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	T#213503d23h34m33s709ms 551us615ns
	DATE	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	D#1970-1-1~D#2016-2-7
	TOD	16#****_****_0000_0000~ 16#****_****_0A4C_B7FF	TOD#0:0:0~ TOD#23:59:59.999
		16#****_****_0526_5C00~ 16#****_****_0A4C_B7FF	
		……	
		16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15

■ **位串转换为字符串**

➢ 位串可以转换为字符串，指令如下所示：



➢ 位串转换为字符串时的取值对应关系如下表：

数据类型		取值对应关系	
In	Out	In	Out
BYTE	STRING	16#00~16#FF	‘00’~’FF’
WORD	STRING	16#0000~16#FFFF	‘0000’~’FFFF’
DWORD	STRING	16#0000_0000~16#FFFF_FFFF	‘00000000’~’FFFFFFFF’
LWORD	STRING	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	‘0000000000000000’~ ‘FFFFFFFFFFFFFFFF’

将位串转换为字符串时，输出字符串的长度须满足输入参数的宽度。如使用 BYTE_TO_STRING 指令时，输出字符串的长度须大于 2 个字符，否则编译时，软件将报错！

⚠ **注意**

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8.13.3 Integers_TO_***（整数转换指令群）

FB/FC	说明	适用机种
FC	整数_TO_***为一组指令，用于将整数型的数据转换成基本数据类型的数据。 “***”允许为所有基本数据类型。	DVP15MC11T DVP15MC11T-06

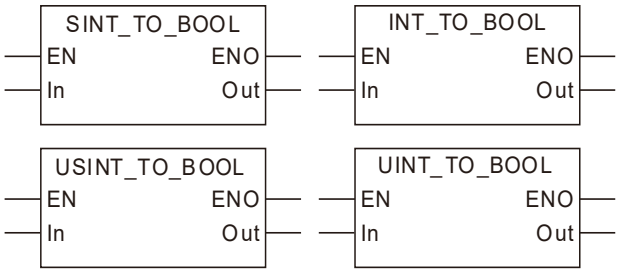
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	用于转换的数据	输入	用于转换的数据	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	转换结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						●	●	●	●	●	●	●	●							
Out	与指令名称中的“***”一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - **整数转换为BOOL**
    - 部分指令如下所示：



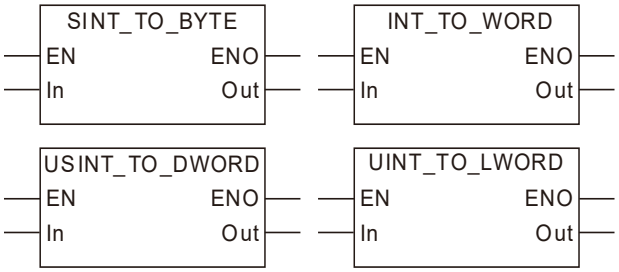
    - 将整数转换为BOOL型数据时，如果整数为0，则转换结果为FALSE，如不为0， 则转换结果为TRUE。详细规则如下表：

数据类型		取值对应关系	
In	Out	In	Out
USINT	BOOL	0	FALSE

数据类型		取值对应关系	
In	Out	In	Out
		1~255	TRUE
UINT	BOOL	0	FALSE
		1~65535	TRUE
UDINT	BOOL	0	FALSE
		1~4294967295	TRUE
ULINT	BOOL	0	FALSE
		1~18446744073709551645	TRUE
SINT	BOOL	0	FALSE
		-128~-1, 1~127	TRUE
INT	BOOL	0	FALSE
		-32768~-1, 1~32767	TRUE
DINT	BOOL	0	FALSE
		-2147483648~-1, 1~2147483647	TRUE
LINT	BOOL	0	FALSE
		-9223372036854775808~-1, 1~9223372036854775807	TRUE

■ **整数转换为位串**

➢ 整数可以转换为位串，部分指令如下所示：



整数转换为位串的方法与位串转换为位串的方法一致，可参阅第8.13.2节内容。

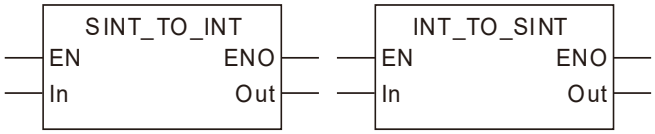
➢ 整数转换为位串时的取值对应关系如下表：

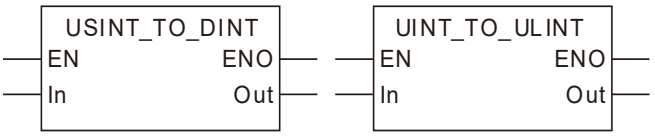
数据类型		取值对应关系	
In	Out	In	Out
USINT	BYTE	16#00~16#FF	16#00~16#FF
	WORD	16#00~16#FF	16#0000~16#00FF
	DWORD	16#00~16#FF	16#0000_0000~16#0000_00FF
	LWORD	16#00~16#FF	16#0000_0000_0000_0000~16#0000_0000_0000_00FF
UINT	BYTE	16#**00~16#**FF	16#00~16#FF
	WORD	16#0000~16#FFFF	16#0000~16#FFFF
	DWORD	16#0000~16#FFFF	16#0000_0000~16#0000_FFFF
	LWORD	16#0000~16#FFFF	16#0000_0000_0000_0000~16#0000_0000_0000_FFFF

数据类型		取值对应关系	
In	Out	In	Out
UDINT	BYTE	16#****_**00~16#****_**FF	16#00~16#FF
	WORD	16#****_0000~16#****_FFFF	16#0000~16#FFFF
	DWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000_0000_0000~ 16#0000_0000_FFFF_FFFF
ULINT	BYTE	16#****_****_****_**00~ 16#****_****_****_**FF	16#00~16#FF
	WORD	16#****_****_****_0000~ 16#****_****_****_FFFF	16#0000~16#FFFF
	DWORD	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF
SINT	BYTE	16#00~16#FF	16#00~16#FF
	WORD	16#00~16#FF	16#0000~16#00FF
	DWORD	16#00~16#FF	16#0000_0000~16#0000_00FF
	LWORD	16#00~16#FF	16#0000_0000_0000_0000~ 16#0000_0000_0000_00FF
INT	BYTE	16#**00~16#**FF	16#00~16#FF
	WORD	16#0000~16#FFFF	16#0000~16#FFFF
	DWORD	16#0000~16#FFFF	16#0000_0000~16#0000_FFFF
	LWORD	16#0000~16#FFFF	16#0000_0000_0000_0000~ 16#0000_0000_0000_FFFF
DINT	BYTE	16#****_**00~16#****_**FF	16#00~16#FF
	WORD	16#****_0000~16#****_FFFF	16#0000~16#FFFF
	DWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000_0000_0000~ 16#0000_0000_FFFF_FFFF
LINT	BYTE	16#****_****_****_**00~ 16#****_****_****_**FF	16#00~16#FF
	WORD	16#****_****_****_0000~ 16#****_****_****_FFFF	16#0000~16#FFFF
	DWORD	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF

■ **整数转换为整数**

➢ 整数与整数之间可以相互转换，部分指令如下所示：





1.整数转换为整数时的转换规则与位串与位串之间的转换规则一致。

2.将宽度小的数据转换为宽度大的数据时，将宽度小的数据中的所有位的值复制到宽度大的数据的对应位中，再将宽度大的数据中剩余的位全部设置为 0。

3.将宽度大的数据转换为宽度小的数据时，将宽度小的数据中的所有位的值修改为宽度大的数据的对应位的值，宽度大的数据中剩余的位不参与转换，也即是其取值如何对转换无影响。

4.若宽度相等时，则将 In 中的所有位的值复制到 Out 中的对应位。

➢ 位串转换为整数时的取值对应关系如下表：

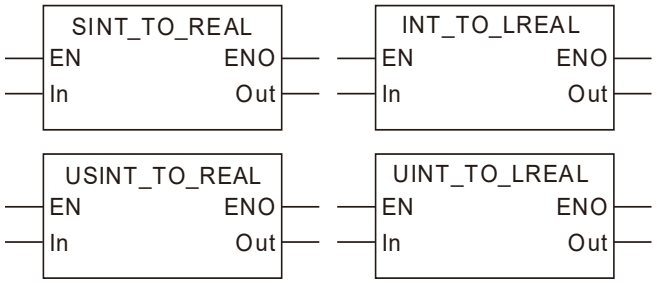
数据类型		取值对应关系	
In	Out	In	Out
USINT	USINT	16#00~16#FF	0~255
	UINT	16#00~16#FF	0~255
	UDINT	16#00~16#FF	0~255
	ULINT	16#00~16#FF	0~255
	SINT	16#00~16#7F	0~127
		16#80~16#FF	- 128~ - 1
	INT	16#00~16#FF	0~255
	DINT	16#00~16#FF	0~255
	LINT	16#00~16#FF	0~255
UINT	USINT	16#**00~16#**FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000~16#FFFF	0~65535
	ULINT	16#0000~16#FFFF	0~65535
	SINT	16#**00~16#**7F	0~127
		16#**80~16#**FF	- 128~ -1
	INT	16#0000~16#7FFF	0~32767
		16#8000~16#FFFF	- 32768~ -1
	DINT	16#0000~16#FFFF	0~65535
	LINT	16#0000~16#FFFF	0~65535
UDINT	USINT	16#****_**00~16#****_**FF	0~255
	UINT	16#****_0000~16#****_FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	SINT	16#****_**00~16#****_**7F	0~127
		16#****_**80~16#****_**FF	-128~-1
	INT	16#****_0000~16#****_7FFF	0~32767
		16#****_8000~16#****_FFFF	-32768~-1
	DINT	16#0000_0000~16#7FFF_FFFF	0~2147483647
		16#8000_0000~16#FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
ULINT	USINT	16#****_****_****_**00~	0~255

数据类型		取值对应关系	
In	Out	In	Out
		16#****_****_****_**FF	
	UINT	16#****_****_****_0000~ 16#****_****_****_FFFF	0~65535
	UDINT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#****_****_****_**00~ 16#****_****_****_**7F	0~127
		16#****_****_****_**80~ 16#****_****_****_**FF	-128~-1
	INT	16#****_****_****_0000~ 16#****_****_****_7FFF	0~32767
		16#****_****_****_8000~ 16#****_****_****_FFFF	-32768~-1
	DINT	16#****_****_0000_0000~ 16#****_****_7FFF_FFFF	0~2147483647
		16#****_****_8000_0000~ 16#****_****_FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000_0000_0000~ 16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807
		16#8000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~0
SINT	USINT	16#00~16#FF	0~255
	UINT	16#00~16#FF	0~255
	UDINT	16#00~16#FF	0~255
	ULINT	16#00~16#FF	0~255
	SINT	16#00~16#7F	0~127
		16#80~16#FF	-128~-1
	INT	16#00~16#FF	0~255
	DINT	16#00~16#FF	0~255
	LINT	16#00~16#FF	0~255
INT	USINT	16#**00~16#**FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000~16#FFFF	0~65535
	ULINT	16#0000~16#FFFF	0~65535
	SINT	16#**00~16#**7F	0~127
		16#**80~16#**FF	-128~-1
	INT	16#0000~16#7FFF	0~32767
		16#8000~16#FFFF	-32768~-1
	DINT	16#0000~16#FFFF	0~65535
	LINT	16#0000~16#FFFF	0~65535
DINT	USINT	16#****_**00~16#****_**FF	0~255
	UINT	16#****_0000~16#****_FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000~16#FFFF_FFFF	0~4294967295

数据类型		取值对应关系	
In	Out	In	Out
	SINT	16#****_**00~16#****_**7F	0~127
		16#****_**80~16#****_**FF	-128~-1
	INT	16#****_0000~16#****_7FFF	0~32767
		16#****_8000~16#****_FFFF	-32768~-1
	DINT	16#0000_0000~16#7FFF_FFFF	0~2147483647
		16#8000_0000~16#FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
LINT	USINT	16#****_****_****_**00~ 16#****_****_****_**FF	0~255
	UINT	16#****_****_****_0000~ 16#****_****_****_FFFF	0~65535
	UDINT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#****_****_****_**00~ 16#****_****_****_**7F	0~127
		16#****_****_****_**80~ 16#****_****_****_**FF	-128~-1
	INT	16#****_****_****_0000~ 16#****_****_****_7FFF	0~32767
		16#****_****_****_8000~ 16#****_****_****_FFFF	-32768~-1
	DINT	16#****_****_0000_0000~ 16#****_****_7FFF_FFFF	0~2147483647
		16#****_****_8000_0000~ 16#****_****_FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000_0000_0000~ 16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807
		16#8000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~0

■ **整数转换为实数**

➢ 整数可以转换为实数，部分指令如下所示：

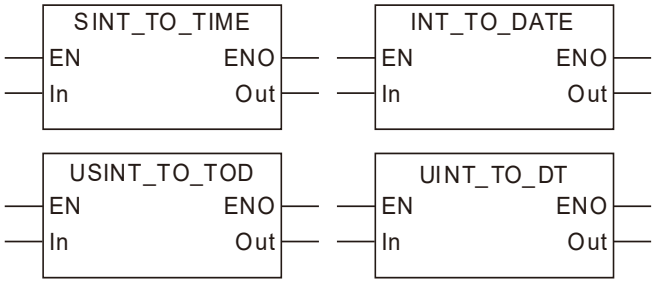


➢ 整数转换为实数的取值对应关系如下表：

数据类型		取值对应关系	
In	Out	In	Out
USINT	REAL	0~255	0~2.55e+2
	LREAL	0~255	0~2.55e+2
UINT	REAL	0~65535	0~6.5535e+4
	LREAL	0~65535	0~6.5535e+4
UDINT	REAL	0~4294967295	0~4.294967e+9
	LREAL	0~4294967295	0~4.294967295e+9
ULINT	REAL	0~18446744073709551615	0~1.844674e+19
	LREAL	0~18446744073709551615	0~1.84467440737095e+19
SINT	REAL	-128~127	-1.28e+2~1.27e+2
	LREAL	-128~127	-1.28e+2~1.27e+2
INT	REAL	-32768~32767	-3.2768e+4~3.2767e+4
	LREAL	-32768~32767	-3.2768e+4~3.2767e+4
DINT	REAL	-2147483648~2147483647	-2.147483e+9~2.147483e+9
	LREAL	-2147483648~2147483647	-2.147483e+9~2.147483e+9
LINT	REAL	-9223372036854775808~ 9223372036854775807	-9.223372e+18~9.223372e+18
	LREAL	-9223372036854775808~ 9223372036854775807	-9.22337203685477e+18~ 9.22337203685477e+18

■ **整数转换为时间或日期**

➢ 整数可以转换为时间或日期，部分指令如下所示：



整数转换为时间或日期的规则与整数转换为无符号整数的规则一致。

➢ 整数转换为时间或日期的取值对应关系如下表：

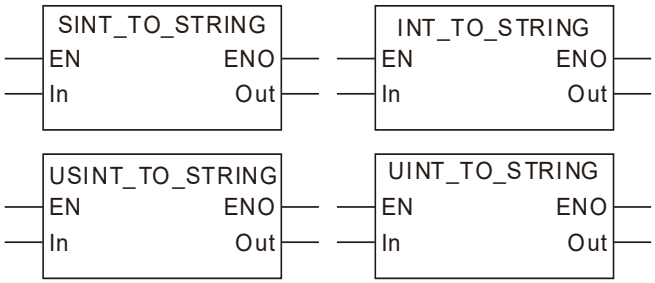
数据类型		取值对应关系	
In	Out	In	Out
USINT	TIME	16#00~16#FF	T#0ms
	DATE	16#00~16#FF	D#1970-1-1
	TOD	16#00~16#FF	TOD#0:0:0~ TOD#0:0:0.255
	DT	16#00~16#FF	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15
UINT	TIME	16#0000~16#FFFF	T#0ms
	DATE	16#0000~16#FFFF	D#1970-1-1
	TOD	16#0000~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	16#0000~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15

数据类型		取值对应关系	
In	Out	In	Out
UDINT	TIME	16#00000000~16#FFFFFFFF	T#0ms~ T#4s294ms
	DATE	16#00000000~16#FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#00000000~16#05265BFF	TOD#0:0:0~ TOD#23:59:59.999
		16#05265C00~16#0A4CB7FF	
		…….	
		16#FC579C00~16#FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#00000000~16#FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15
ULINT	TIME	16#0000000000000000~ 16# FFFFFFFFFFFFFFFF	T#213503d23h34m33s709ms 551us615ns
	DATE	16#********00000000~ 16#********FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#********00000000~ 16#********0A4CB7FF	TOD#0:0:0~ TOD#23:59:59.999
		16#********05265C00~ 16#********0A4CB7FF	
		……	
		16#********00000000~ 16#********FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#********00000000~ 16#********FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15
SINT	TIME	16#00~16#FF	T#0ns~T#255ns
	DATE	16#00~16#FF	D#1970-1-1
	TOD	16#00~16#FF	TOD#0:0:0~ TOD#0:0:0.255
	DT	16#00~16#FF	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15
INT	TIME	16#0000~16#FFFF	T#0ns~T#65us535ns
	DATE	16#0000~16#FFFF	D#1970-1-1
	TOD	16#0000~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	16#0000~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15
DINT	TIME	16#00000000~16#FFFFFFFF	T#0ns~ T#4s294ms967us295ns
	DATE	16#00000000~16#FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#00000000~16#05265BFF	TOD#0:0:0~ TOD#23:59:59.999
		16#05265C00~16#0A4CB7FF	
		…….	
		16#FC579C00~16#FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#00000000~16#FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15
LINT	TIME	16#0000000000000000~ 16# FFFFFFFFFFFFFFFF	T#213503d23h34m33s709ms 551us615ns
	DATE	16#********00000000~	D#1970-1-1~D#2016-2-7

数据类型		取值对应关系	
In	Out	In	Out
		16#********FFFFFFFF	
	TOD	16#********00000000~ 16#********0A4CB7FF	TOD#0:0:0~ TOD#23:59:59.999
		16#********05265C00~ 16#********0A4CB7FF	
		……	
		16#********00000000~ 16#********FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#********00000000~ 16#********FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15

■ **整数转换为字符串**

➢ 整数可以转换为字符串，部分指令如下所示：



➢ 整数转换为字符串时的取值对应关系如下表：

数据类型		取值对应关系	
In	Out	In	Out
USINT	STRING	0~255	‘0’~’255’
UINT	STRING	0~65535	‘0’~’65535’
UDINT	STRING	0~4294967295	‘0’~’4294967295’
ULINT	STRING	0~18446744073709551615	‘0’~’18446744073709551615’
SINT	STRING	-128~127	‘-128’~’127’
INT	STRING	-32768~32767	‘-32768’~’32767’
DINT	STRING	-2147483648~2147483647	‘-2147483648’~’2147483647’
LINT	STRING	-9223372036854775808~ 9223372036854775807	‘-9223372036854775808’~ ‘9223372036854775807’

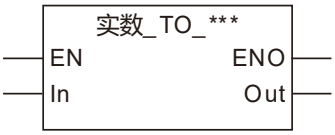
将位串转换为字符串时，输出字符串的长度须满足输入参数的宽度。

⚠ **注意**

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8.13.4 Real numbers_TO_***（实数转换指令群）

FB/FC	说明	适用机种
FC	实数_TO_***为一组指令，用于将实数型的数据转换成基本数据类型的数据。 “***”允许为所有基本数据类型。	DVP15MC11T DVP15MC11T-06



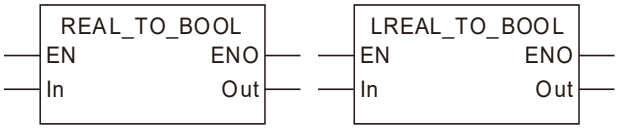
- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	用于转换的数据	输入	用于转换的数据	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	转换结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out	与指令名称中的“***”一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - **实数转换为BOOL**
    - 相关指令如下所示：



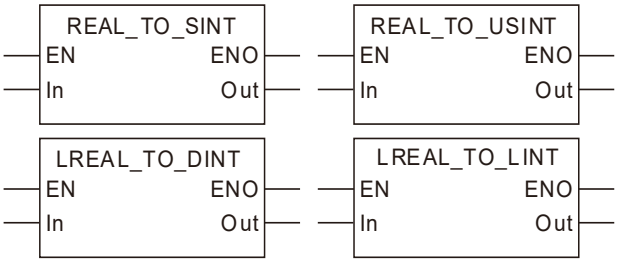
    - 将实数转换为BOOL型数据时，如果实数为0，则转换结果为FALSE，如不为0，则转换结果为TRUE。详细规则如下表：

数据类型		取值对应关系	
In	Out	In	Out
REAL	BOOL	-3.402823E+38~-1.175495E-38	TRUE
		0	FALSE
		1.175495E-38~3.402823E+38	TRUE
LREAL	BOOL	-1.79769313486231E+308~ -2.22507385850721E-308	TRUE
		0	FALSE

数据类型		取值对应关系	
In	Out	In	Out
		2.22507385850721E-308~ 1.79769313486231E+308	TRUE

■ **实数转换为整数**

➢ 实数可以转换为整数，部分指令如下所示：



➢ 实数转换为整数时，满足四舍五入的关系，如下所示：

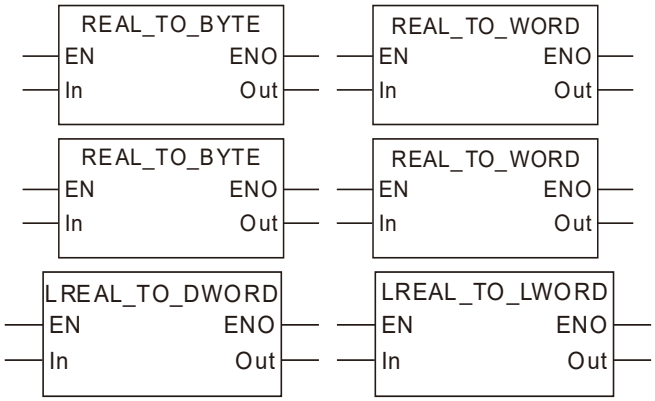
情形1：小数部分的第一位数小于5，则将小数部分舍去，且整数部分不变。

情形2：小数部分的第一位数大于等于5，则将小数部分舍去，且整数部分加1。

输入值		输出结果	
		数据类型	输出值
情形1	1.36	SINT	1
		USINT	1
	-2.4	SINT	-2
		USINT	254
情形2	1.6	SINT	2
		USINT	2
	-2.6	SINT	-3
		USINT	253

■ **实数转换为位串**

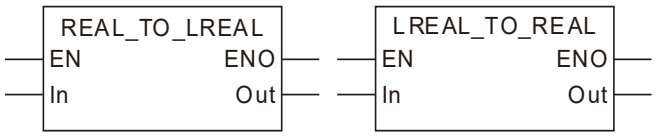
➢ 实数可以转换为位串，部分指令如下所示：



实数转换为位串时的规则与实数转换为无符号整数的规则一致。

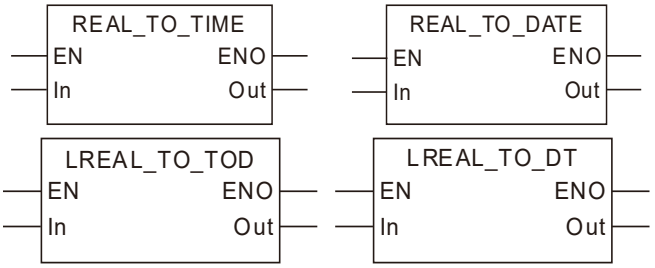
■ **实数转换为实数**

➢ 实数可以转换为实数，部分指令如下所示：



■ **实数转换为时间或日期**

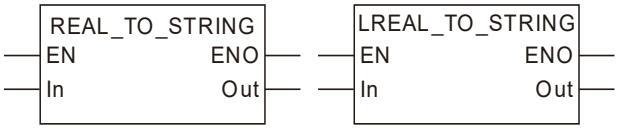
➢ 实数可以转换为时间或日期，部分指令如下所示：



实数转换为时间或日期时，先将实数转换为整数，再将该整数转换为时间或日期，请参阅实数转换为整数和整数转换为时间或日期相关内容。

■ **实数转换为字符串**

➢ 实数可以转换为字符串，指令如下所示：



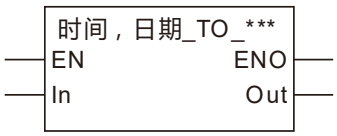
实数转换为字符串的规则与整数转换为字符串的规则一致，请参阅第8.13.3节。

⚠ **注意**

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8.13.5 Times,dates_TO_***（时间，日期转换指令群）

FB/FC	说明	适用机种
FC	时间，日期_TO_***为一组指令，用于将时间或日期型的数据转换成基本数据类型的数据。"***" 允许为所有基本数据类型。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	用于转换的数据	输入	用于转换的数据	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	转换结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																●	●	●	●	
Out	与指令名称中的 "***" 一致																			

说明：上表中的 "●" 表示指令参数允许与该数据类型的变量或常量连接。

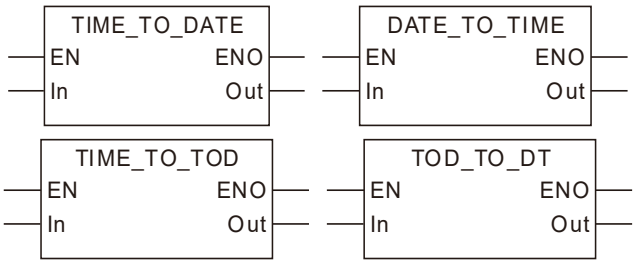
- **功能说明**

  - **时间、日期转换为BOOL、位串、整数、实数、字符串**

    时间、日期转换为BOOL、位串、整数、实数、字符串的规则与无符号整数转换为BOOL、位串、整数、实数、字符串的规则一致，请参阅第8.13.5节。

  - **时间、日期转换为时间、日期**

    ➢ 时间、日期型数据可以相互转换，部分指令如下所示：



时间、日期型数据可以相互转换的方法与无符号整数之间的转换方法一致，但须注意转换时的单位统一（TIME的存储单位为纳秒，其它为毫秒）。

⚠ **注意**

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8.13.6 Strings_TO_***（字符串转换指令群）

FB/FC	说明	适用机种
FC	字符串_TO_***为一组指令，用于将字符串型的数据转换成基本数据类型的数据。“***”允许为所有基本数据类型。	DVP15MC11T DVP15MC11T-06

- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	用于转换的数据	输入	用于转换的数据	由与输入参数所连变量的数据类型决定
Out	转换结果	输出	转换结果	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
Out	与指令名称中的“***”一致																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

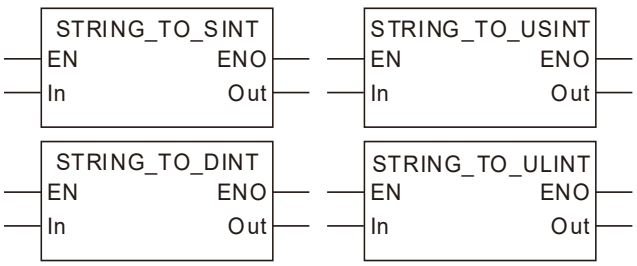
  - **字符串转换为BOOL**

    - 相关指令如下所示：

字符串转换为BOOL型数据的规则为：仅当字符串为 'TRUE' 或 'true' 时，输出的BOOL值为TRUE，其它均为FALSE.

  - **字符串转换为整数**

    - 字符串可以转换为整数，部分指令如下所示：

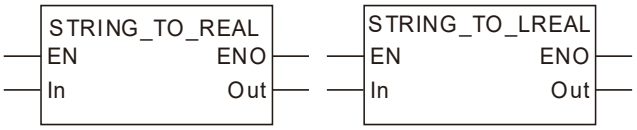


- 将字符串转换为整数时，要求字符串表示的是整数值，如'123'、'-123'、'+123'，而形如 'M123' 的字符串则不允许转换为整数，转换示例如下表：

输入值	输出结果	
	数据类型	输出值
‘123’	SINT	123
‘+123’	SINT	123
‘-123’	SINT	-123
‘M123’	SINT	不允许转换，输出变量保持原值

■ **字符串转换为实数**

- 字符串可以转换为实数，部分指令如下所示：

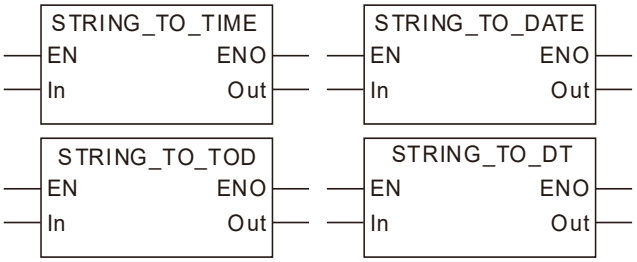


- 将字符串转换为实数时，要求字符串表示的是实数值，如'123'、'-123.123'、'1.23e+5'，转换示例如下表：

输入值	输出结果	
	数据类型	输出值
‘123’	REAL	123
‘-123.123’	REAL	-123.123
‘M123.123’	REAL	不允许转换，输出变量保持原值

■ **字符串转换为时间或日期**

- 字符串可以转换为时间或日期，指令如下所示：

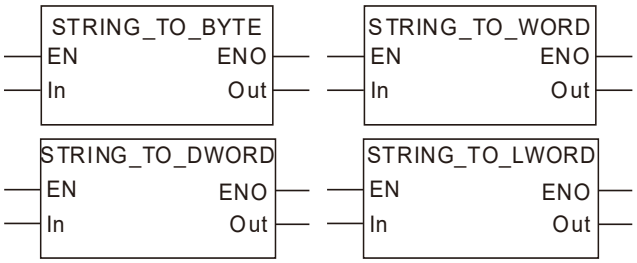


- 将字符串转换为时间或日期时，要求字符串表示的是时间或日期的值，
  如 'T#1ns'、'D#1970-1-1'、'TOD#0:0:0'， 'DT#1970-1-1-0:0:0'，转换示例如下表：

输入值	输出结果	
	数据类型	输出值
'D#1970-1-1'	DATE	D#1970-1-1
'TOD#0:0:0'	TOD	TOD#0:0:0
'DT#1970-1-1-0:0:0'	DT	DT#1970-1-1-0:0:0

■ **字符串转换为位串**

➢ 字符串可以转换为位串，指令如下所示：



字符串转换为位串的方法与字符串转换为整数的方法一致。

⚠ **注意**

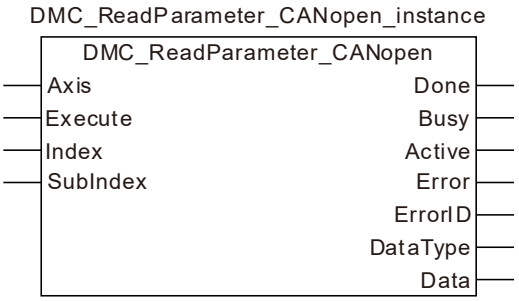
输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

8.14 通讯指令

8.14.1 CANopen 通讯指令

8.14.1.1 DMC_ReadParameter_CANopen（读取参数指令）

FB/FC	说明	适用机种
FB	此指令用于读取从站的参数值。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Axis（从站的站号）	设定指令欲控制的从站	USINT	1~127（不可缺省）	Execute由FALSE变为TRUE
Execute（执行位）	当Execute由 FALSE 变TRUE时，执行该指令。	BOOL	TRUE或FALSE（FALSE）	-
Index（索引）	读取参数的索引。	UINT	0	Execute由FALSE变为TRUE
SubIndex（子索引）	读取参数的子索引。	USINT	0	Execute由FALSE变为TRUE

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active（控制中）	该输出参数为 TRUE 时表示指令正在控制从站	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

名称	功能	数据类型	输出范围
Data Type （数据类型）	读取参数的数据类型。 1：代表读取的参数为 Byte 类型； 2：代表读取的参数为 Word 类型； 4：代表读取的参数为 Double Word 类型。	USINT	-
Data（参数值）	读到的参数值。	UDINT	-

■ **欲读取的从站参数对应索引和子索引如下：**

1. 用户定义参数为欲读取的伺服驱动器参数，长度由用户根据所读取参数的数据类型指定，字节型参数长度为 1，字型参数长度为 2，双字型参数长度为 4。伺服驱动器参数对应索引子索引的计算方法如下：

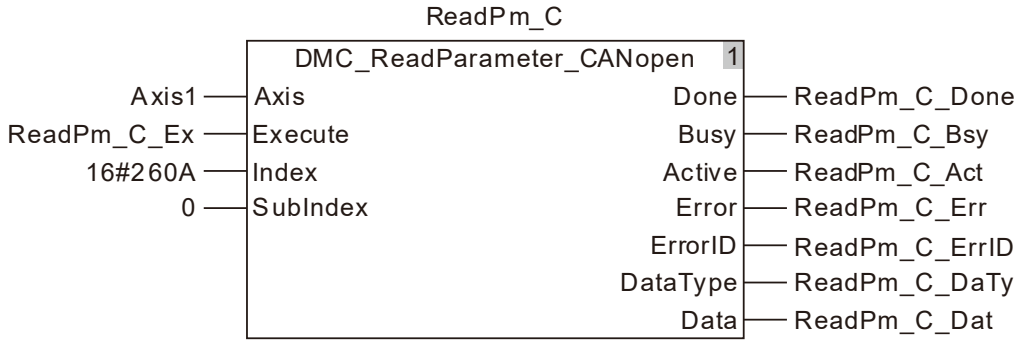
   索引 = 伺服驱动器参数（十六进制）+ 2000（十六进制）

   子索引 = 0.

   举例： 伺服驱动器参数 P6-10 的索引计算方法为 2000 + 060A(P6-10 的十六进制数) = 260A.，子索引为 0.。

➢ **变量和程序**

变量名	数据类型	初始值
ReadPm_C	DMC_ReadParameter_CANopen	
ReadPm_C_Ex	BOOL	FALSE
ReadPm_C_Done	BOOL	
ReadPm_C_Bsy	BOOL	
ReadPm_C_Act	BOOL	
ReadPm_C_Err	BOOL	
ReadPm_C_ErrID	WORD	
ReadPm_C_DaTy	USINT	
ReadPm_C_Dat	UDINT	



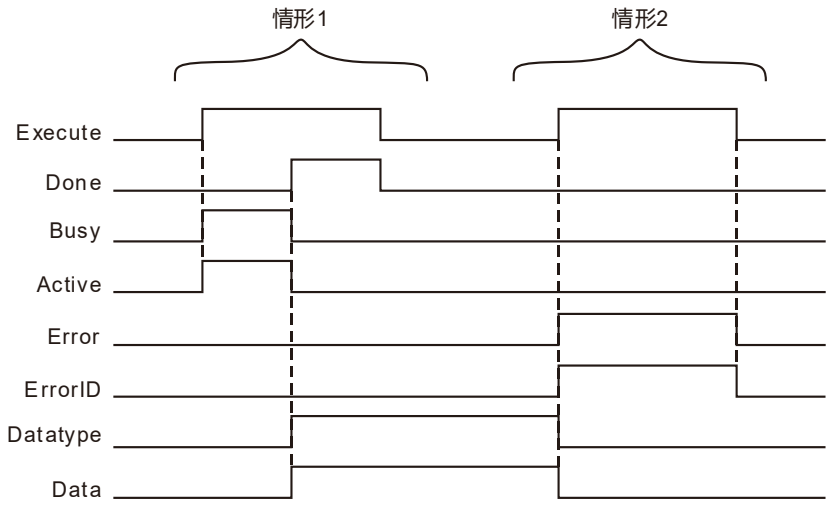
2. 其它从站参数对应索引和子索引，请参考产品的 CANopen 功能相关手册。

● **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当参数内容读取完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Busy 变为 FALSE
Active	◆ 当指令开始控制从站时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Active 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Ex*ecute 由 FALSE 变为 TRUE 后，Busy 和 Active 变为 TRUE，且一个周期后，Done 变为 TRUE，Datatype 和 Data 显示相应的数据。 Done 变为 TRUE 时， 同时 Busy 和 Active 变为 FALSE。当 Execute 由 TRUE 变为 FALSE 后,Done 由 TRUE 变为 FALSE，Datatype 和 Data 保持原有的数据。

情形2： 指令执行之前，输入参数非法（如轴号为 0）。当 Execute 由 FALSE 变为 TRUE 后， Error 由 FALSE 变为 TRUE ，Datatype 和 Data 的数据会清零 ，Error ID 显示相应的错误代码 。当 Execute 由 TRUE 变为 FALSE 后，Error 由 TRUE 变为 FALSE，Error ID 的内容清零。

- **功能说明**

此指令用于读取从站的参数值，用户可以指定欲读取参数的索引（Index）和子索引（SubIndex）。

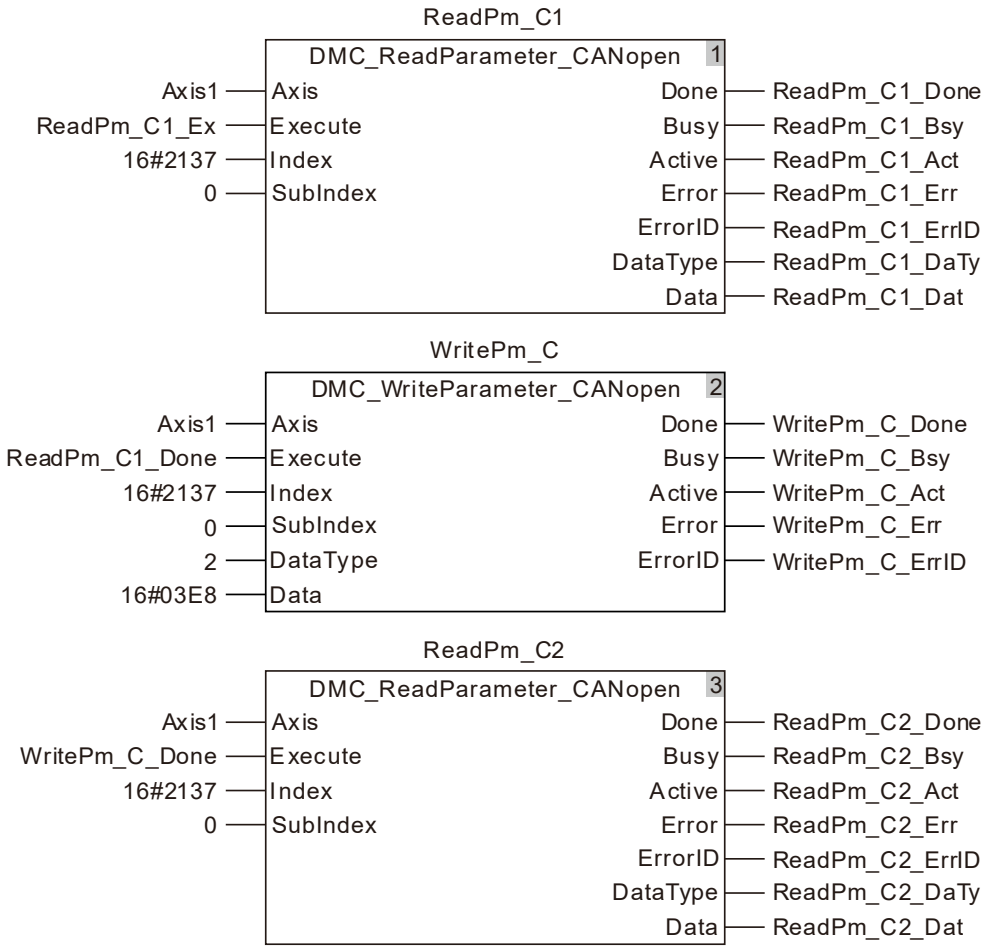
程序范例

DMC_ReadParameter_CANopen 指令执行时的范例如下所示：

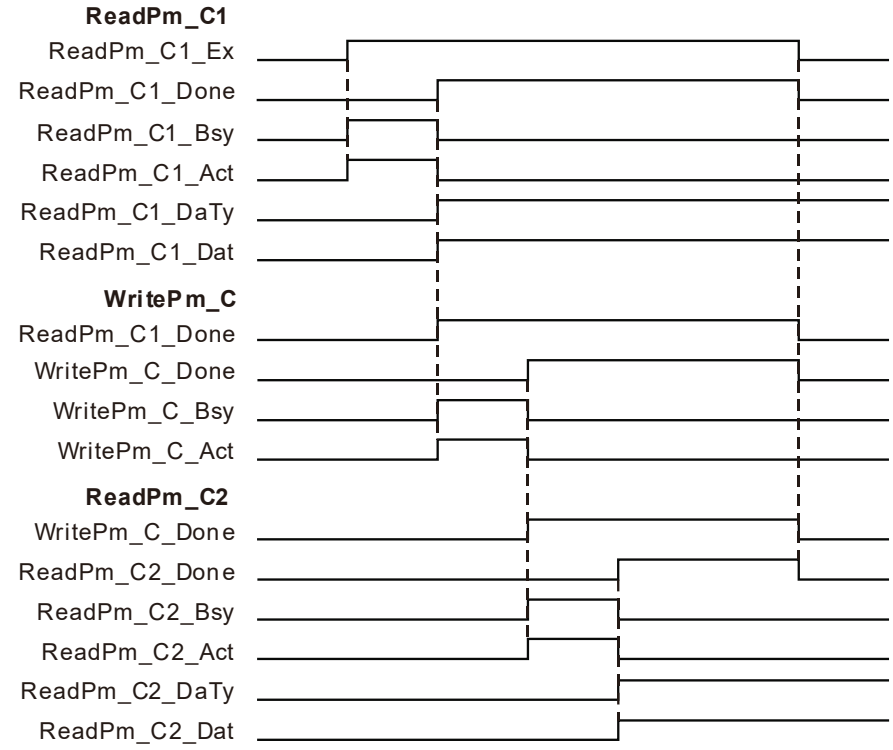
- **变量和程序**

变量名	数据类型	当前值
ReadPm_C1	DMC_ReadParameter_CANopen	
Axis1	USINT	1

变量名	数据类型	当前值
ReadPm_C1_Ex	BOOL	TRUE
ReadPm_C1_Done	BOOL	TRUE
ReadPm_C1_Bsy	BOOL	FALSE
ReadPm_C1_Act	BOOL	FALSE
ReadPm_C1_Err	BOOL	FALSE
ReadPm_C1_ErrID	WORD	FALSE
ReadPm_C1_DaTy	USINT	2
ReadPm_C1_Dat	UDINT	5000
WritePm_C	DMC_WriteParameter_CANopen	
WritePm_C_Done	BOOL	TRUE
WritePm_C_Bsy	BOOL	FALSE
WritePm_C_Act	BOOL	FALSE
WritePm_C_Err	BOOL	FALSE
WritePm_C_ErrID	WORD	FALSE
ReadPm_C2	DMC_ReadParameter_CANopen	
ReadPm_C2_Done	BOOL	TRUE
ReadPm_C2_Bsy	BOOL	FALSE
ReadPm_C2_Act	BOOL	FALSE
ReadPm_C2_Err	BOOL	FALSE
ReadPm_C2_ErrID	WORD	FALSE
ReadPm_C2_DaTy	USINT	2
ReadPm_C2_Dat	UDINT	1000


ReadPm_C1
DMC_ReadParameter_CANopen
1
Axis1
Axis
ReadPm_C1_Ex
Execute
16#2137
Index
0
SubIndex
Done
Busy
Active
Error
ErrorID
DataType
Data
ReadPm_C1_Done
ReadPm_C1_Bsy
ReadPm_C1_Act
ReadPm_C1_Err
ReadPm_C1_ErrID
ReadPm_C1_DaTy
ReadPm_C1_Dat
WritePm_C
DMC_WriteParameter_CANopen
2
Axis1
Axis
ReadPm_C1_Done
Execute
16#2137
Index
0
SubIndex
2
DataType
16#03E8
Data
Done
Busy
Active
Error
ErrorID
WritePm_C_Done
WritePm_C_Bsy
WritePm_C_Act
WritePm_C_Err
WritePm_C_ErrID
ReadPm_C2
DMC_ReadParameter_CANopen
3
Axis1
Axis
WritePm_C_Done
Execute
16#2137
Index
0
SubIndex
Done
Busy
Active
Error
ErrorID
DataType
Data
ReadPm_C2_Done
ReadPm_C2_Bsy
ReadPm_C2_Act
ReadPm_C2_Err
ReadPm_C2_ErrID
ReadPm_C2_DaTy
ReadPm_C2_Dat

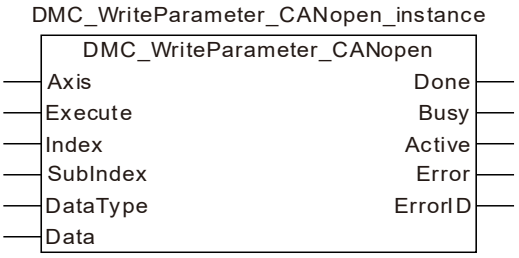
■ **时序图及说明**


ReadPm_C1
ReadPm_C1_Ex
ReadPm_C1_Done
ReadPm_C1_Bsy
ReadPm_C1_Act
ReadPm_C1_DaTy
ReadPm_C1_Dat
WritePm_C
ReadPm_C1_Done
WritePm_C_Done
WritePm_C_Bsy
WritePm_C_Act
ReadPm_C2
WritePm_C_Done
ReadPm_C2_Done
ReadPm_C2_Bsy
ReadPm_C2_Act
ReadPm_C2_DaTy
ReadPm_C2_Dat

- ❖ 当 ReadPm_C1_Ex 由 FALSE 变为 TRUE 时，第一个 DMC_ReadParameter_CANopen 指令开始执行，执行完成时 ReadPm_C1_Done 变为 TRUE，ReadPm_C1_DaTy = 2，ReadPm_C1_Dat=5000，即读取伺服从站参数 P1-55 的内容为 5000（伺服最大速度限制为 5000rpm）。
- ❖ 当 ReadPm_C1_Done 由 FALSE 变为 TRUE 时，DMC_WriteParameter_CANopen 指令开始执行，执行完成时 WritePm_C_Done 变为 TRUE，即写入伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）成功。
- ❖ 当 WritePm_C_Done 由 FALSE 变为 TRUE 时，第二个 DMC_ReadParameter_CANopen 指令开始执行，执行完成时 ReadPm_C2_Done 变为 TRUE，ReadPm_C2_DaTy = 2，ReadPm_C2_Dat=1000，即读取伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）。

8.14.1.2 DMC_WriteParameter_CANopen（写入参数指令）

FB/FC	说明	适用机种
FB	此指令用于设定从站的参数值。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Axis（从站的站号）	设定指令欲控制的从站	USINT	1~127（不可缺省）	Execute由FALSE变为TRUE
Execute（执行位）	当Execute由 FALSE 变TRUE时，执行该指令。	BOOL	TRUE或FALSE（FALSE）	-
Index（索引）	写入参数的索引。	UINT	-	-
SubIndex（子索引）	写入参数的子索引。	USINT	-	-
Data Type（数据类型）	写入参数的数据类型。 **1**：代表写入的参数为Byte类型； **2**：代表写入的参数为Word类型； **4**：代表写入的参数为Double Word类型。	USINT	-	-
Data（参数值）	写入参数的内容值。	UDINT	-	-

注：

1. 数据类型（Data Type）必须为写入参数的数据类型，若该值填写不正确，指令会报错。
2. 关于 CANopen 从站的索引和子索引的计算方法请参考第 9 章 “轴参数介绍”。

- **输出参数**

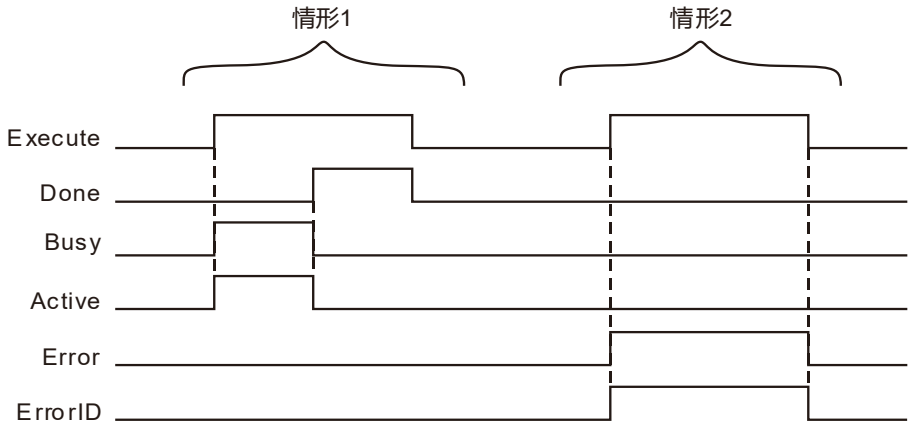
名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制从站	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当参数内容写入完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Busy 变为 FALSE
Active	◆ 当指令开始控制从站时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Active 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后，*Busy* 和 *Active* 变为 TRUE，且一个周期后，Done 变为 TRUE，Done 变为 TRUE 时，同时 Busy 和 Active 变为 FALSE，当 *Execute* 由 TRUE 变为 FALSE 时，Done 由 TRUE 变为 FALSE。

情形2： 指令执行之前，输入参数非法（如轴号为 0）。 当 *Execute* 由 FALSE 变为 TRUE 时，Error 由 FALSE 变为 TRUE，Error ID 显示相应的错误代码。当 *Execute* 由 TRUE 变为 FALSE 时，Error 由 TRUE 变为 FALSE，Error ID 的内容清零。

- **功能说明**

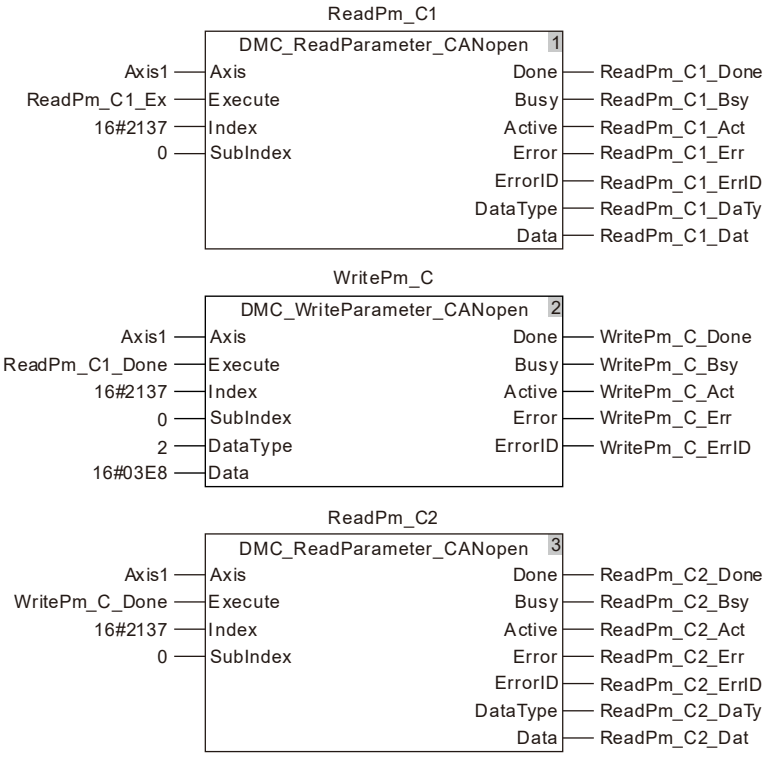
此指令用于设定从站的参数值，用户可以指定欲设定参数的索引（Index）和子索引（Sub-Index）。

程序范例

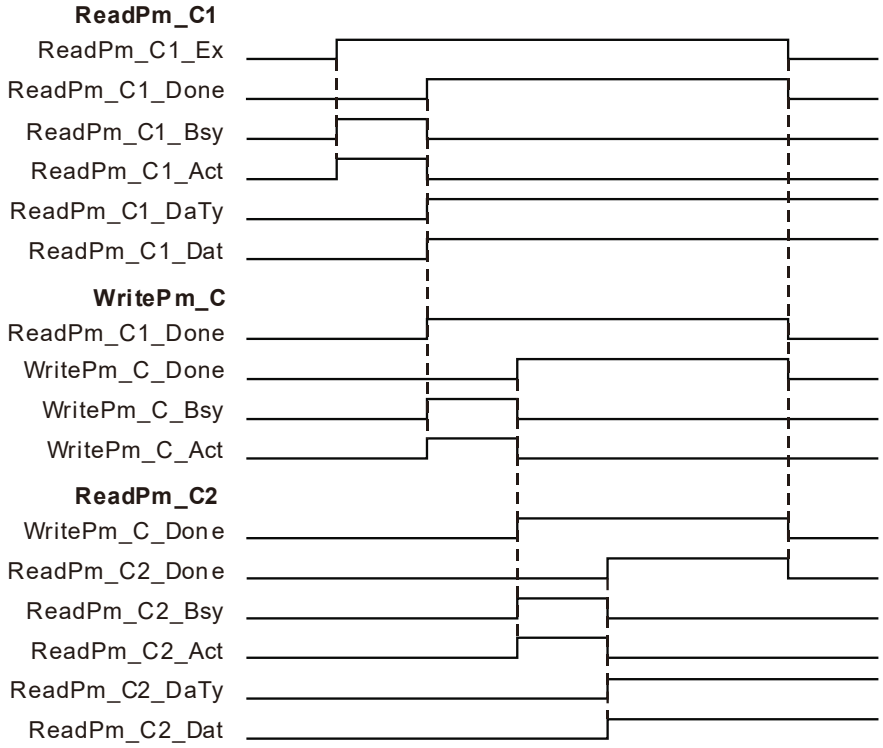
DMC_WriteParameter_CANopen 指令执行时的范例如下所示：

■ **变量和程序**

变量名	数据类型	当前值
ReadPm_C1	DMC_ReadParameter_CANopen	
Axis1	USINT	1
ReadPm_C1_Ex	BOOL	TRUE
ReadPm_C1_Done	BOOL	TRUE
ReadPm_C1_Bsy	BOOL	FALSE
ReadPm_C1_Act	BOOL	FALSE
ReadPm_C1_Err	BOOL	FALSE
ReadPm_C1_ErrID	WORD	FALSE
ReadPm_C1_DaTy	USINT	2
ReadPm_C1_Dat	UDINT	5000
WritePm_C	DMC_WriteParameter_CANopen	
WritePm_C_Done	BOOL	TRUE
WritePm_C_Bsy	BOOL	FALSE
WritePm_C_Act	BOOL	FALSE
WritePm_C_Err	BOOL	FALSE
WritePm_C_ErrID	WORD	FALSE
ReadPm_C2	DMC_ReadParameter_CANopen	
ReadPm_C2_Done	BOOL	TRUE
ReadPm_C2_Bsy	BOOL	FALSE
ReadPm_C2_Act	BOOL	FALSE
ReadPm_C2_Err	BOOL	FALSE
ReadPm_C2_ErrID	WORD	FALSE
ReadPm_C2_DaTy	USINT	2
ReadPm_C2_Dat	UDINT	1000



■ **时序图及说明**



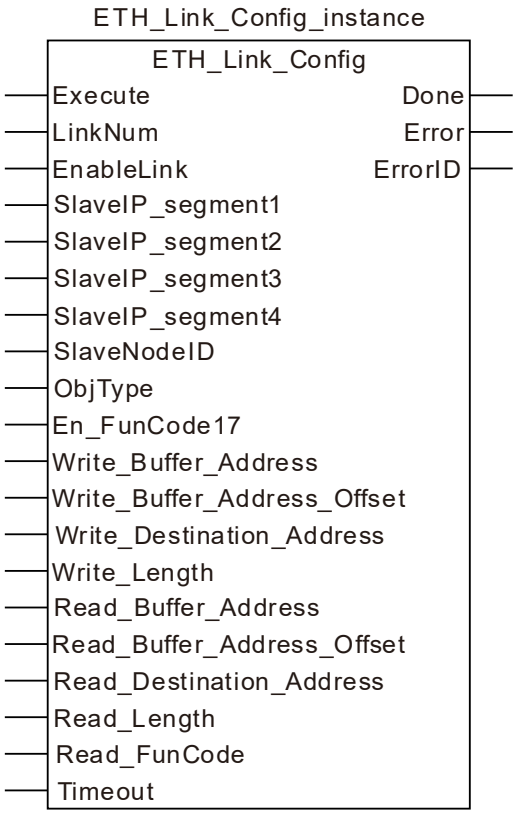
❖ 当 ReadPm_C1_Ex 由 FALSE 变为 TRUE 时，第一个 DMC_ReadParameter_CANopen 指令开始执行，执行完成时 ReadPm_C1_Done 变为 TRUE，ReadPm_C1_DaTy =2，ReadPm_C1_Dat=5000，即读取伺服从站参数 P1-55 的内容为 5000 (伺服最大速度限制为 5000rpm)。

- ❖ 当 ReadPm_C1_Done 由 FALSE 变为 TRUE 时，DMC_WriteParameter_CANopen 指令开始执行，执行完成时 WritePm_C_Done 变为 TRUE，即写入伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）成功。
- ❖ 当 WritePm_C_Done 由 FALSE 变为 TRUE 时，第二个 DMC_ReadParameter_CANopen 指令开始执行，执行完成时 ReadPm_C2_Done 变为 TRUE，ReadPm_C2_DaTy =2，ReadPm_C2_Dat=1000，即读取伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）。

8.14.2 以太网指令

8.14.2.1 ETH_Link_Config（MODBUS TCP 数据交换配置）

FB/FC	说明	适用机种
FB	此指令用于配置 DVP-15MC 系列运动控制器 LAN2 口 MODBUS TCP 数据交换参数。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
LinkNum（编号）	设定 MODBUS TCP 数据交换的编号。	UINT	1~16（0）	Execute 由 FALSE 变为 TRUE 时
EnableLink（连接开启或关闭）	设定该连接是否开启或关闭。	BOOL	TRUE 或 FALSE（FALSE）	Execute 由 FALSE 变为 TRUE 时
SlaveIP_segment1（目标 IP 地址第一段）	设定目标 IP 地址的第一段。	USINT	0~255（0）	Execute 由 FALSE 变为 TRUE 时
SlaveIP_segment2（目标 IP 地址第二段）	设定目标 IP 地址的第二段。	USINT	0~255（0）	Execute 由 FALSE 变为 TRUE 时

名称	功能	数据类型	设定范围 （缺省值）	生效时机
SlaveIP_segment3 （目标 IP 地址第三段）	设定目标 IP 地址的第三段。	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
SlaveIP_segment4 （目标 IP 地址第四段）	设定目标 IP 地址的第四段。	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
SlaveNodeID （从站站号）	设定从站 MODBUS 站号	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
ObjType （欲读写装置类型）	设定读写从站装置的类型 0：Word 类型 1：Bit 类型	USINT	0~1 （0）	Execute 由 FALSE 变为 TRUE 时
En_FunCode17 （是否使用 17 功能码）	设定是否使用 17 功能码	BOOL	TRUE 或 FALSE （FALSE）	Execute 由 FALSE 变为 TRUE 时
Write_Buffer_Address （发送数据存放装置）	设定存放主站发送数据的起始装置	UINT	%MW0~%MW32767 %QW0~%QW63	Execute 由 FALSE 变为 TRUE 时
Write_Buffer_Address_Offset （发送数据存放装置偏移量）	设定主站发送数据起始装置的偏移量：若写从站的 Word 地址时，该引脚填 1 表示偏移 1 个 Word；若写从站的 Bit 地址时，该引脚填 1 表示偏移 1 个 Bit	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
Write_Destination_Address （写入从站 MODBUS 地址）	设定接收主站数据的从站 MODBUS 起始地址	UINT	16#0~16#FFFF （0）	Execute 由 FALSE 变为 TRUE 时
Write_Length （写入长度）	设定写入数据长度	UINT	字装置：0~100 位装置：0~256 字装置或者位装置可以通过 ObjType 的值设定 （0）	Execute 由 FALSE 变为 TRUE 时
Read_Buffer_Address （接收数据存放地址）	设定存放主站接收数据的起始装置	UINT	%MW0~%MW32767 %QW0~%QW63	Execute 由 FALSE 变为 TRUE 时
Read_Buffer_Address_Offset （接收数据存放地址偏移量）	设定主站接收数据起始装置的偏移量：若读从站的 Word 地址时，该引脚	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	填 1 表示偏移 1 个 Word；若读从站的 Bit 地址时，该引脚填 1 表示偏移 1 个 Bit			
Read_Destination_Address （读取从站 MODBUS 地址）	设定主站准备读取从站的 MODBUS 起始地址	UINT	16#0~16#FFFF （0）	Execute 由 FALSE 变为 TRUE 时
Read_Length （读取长度）	设定读取长度	UINT	ObjType 为 0 时：0~100； ObjType 为 1 时：0~256； （0）	Execute 由 FALSE 变为 TRUE 时
Read_FunCode （读 bit 装置功能码）	设定读 Bit 地址时使用的功能码	USINT	1~2 （0）	Execute 由 FALSE 变为 TRUE 时
Timeout （超时时间）	设定主站等待从站响应的时间。从站在设定时间内没有响应主站的请求时，认为从站超时，单位：ms	UINT	0~65535 （0）	Execute 由 FALSE 变为 TRUE 时

说明：

1. 输入参数 SlaveIP_segment1、SlaveIP_segment2、SlaveIP_segment3、SlaveIP_segment4 分别为从站 IP 地址的第一段到第四段，例如从站 IP 为 192.168.1.10，那么参数 SlaveIP_segment1 输入值 192，参数 SlaveIP_segment2 输入值 168，参数 SlaveIP_segment3 输入值 1，参数 SlaveIP_segment4 输入值 10；
2. 输入参数 Write_Buffer_Address、Read_Buffer_Address 表示 MODBUS TCP 主站发送和接收数据存放的装置首地址，两个输入引脚不可缺省，用户可以在引脚处定义变量并绑定地址（如%MW0）。
3. 输入参数 ObjType 代表读写参数的数据类型。当 ObjType 为 0 时表示读、写从站的 Word 装置，此时 Write_Length、Read_Length 的范围为 0~100，Write_Length 和 Read_Length 不能同时为 0；当 ObjType 为 1 时表示读、写从站的 Bit 装置，此时 Write_Length、Read_Length 的范围为 0~256，Write_Length 和 Read_Length 不能同时为 0。
4. 当用户读取从站 Bit 装置时，用户必须设置 Read_FunCode 的值为 01 或 02，用户可以根据读 Bit 装置的类型并参考相应模块手册来选择功能码。

- **输出参数**

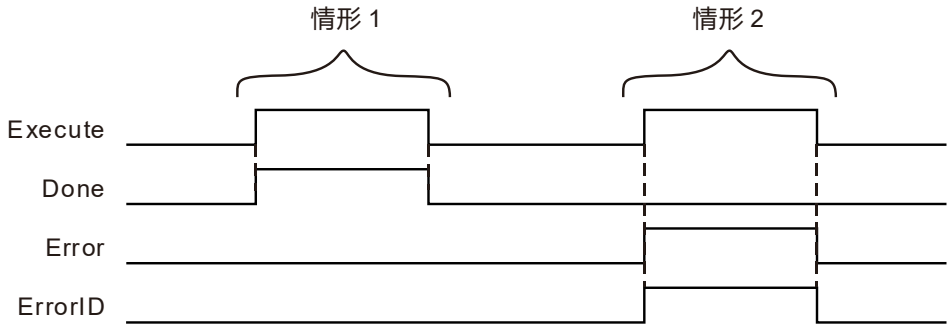
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示参数配置完成。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当参数配置完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 False 变为 TRUE 时， Done 变为 TRUE，参数写入成功；当 Execute 由 TRUE 变为 FALSE 时， Done 同时由 TRUE 变为 FALSE。

情形2： 当 Execute 由 False 变为 TRUE 时，若参数不合法，则 Error 同时变为 TRUE，ErrorID 显示对应的错误码；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

- **功能说明**

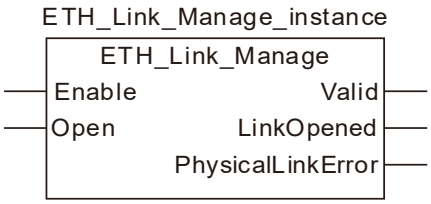
此指令用于 MODBUS TCP 参数配置。V1.01 及以上固件版本支持该功能。

1. 当 DVP-15MC 系列运动控制器使用 MODBUS TCP 主站功能时，仅 LAN2 口支持主站功能，LAN1 口不支持主站功能。
2. ETH_Link_Config 搭配指令 ETH_Link_Manage 使用才能发送 MODBUS TCP 数据。
3. 指令执行过程中修改参数并不会被写入，当用户修改指令参数后，必须重新触发执行位 Execute 后，参数才会被写入。
4. 修改 ETH_Link_Config 指令参数后，新的参数并不会立即生效，只有当重新执行指令 ETH_Link_Manage 时才会生效。
5. 当用户选择读、写从站 WORD 类型装置时，用户可以选择本机的%MW 装置作为读、写数据的存放装置，范围为%MW0~%MW32767，超出该范围或使用其他装置指令报错。
6. 当用户选择读、写从站 BIT 类型装置时，用户可以选择本机的%MW 和%QW 装置作为读、写数据的存放装置，范围为%MW0~%MW32767 和%QW0~%QW63，超出该范围或使用其他装置指令报错。

7. 读、写从站 BIT 类型装置且偏移值为 0 时，主机会从存放数据的起始装置的 bit0 开始读或写 Write_Length、Read_Length 个 bit 数据；当读、写从站 BIT 类型装置且偏移值为 n（n 不等于 0）时，主机会从存放数据装置的 bit0 向后偏移 n 个 bit 开始读或写 Write_Length、Read_Length 个 bit 数据。如 Write_Buffer_Address_Offset 为 0 时，ObjType 为 1，Write_Buffer_Address 绑定的装置为%MW0，Write_Length 为 5，则主机会将 %MX0.0~%MX0.4 的数据发送给从站；如 Write_Buffer_Address_Offset 为 8 时，ObjType 为 1，Write_Buffer_Address 绑定的装置为%MW0，Write_Length 为 5，则主机会将%MX1.0~%MX1.4 的数据发送给从站；
8. 此指令设定的参数值仅在主机运行过程中有效，当 DVP-15MC 系列运动控制器断电之后，重新上电时，断电之前配置的参数全部失效，用户如果需要使用断电前的配置就必须重新执行 ETH_Link_Config 指令。
9. 当 DVP-15MC 系列运动控制器使用 MODBUS TCP 主站功能时，DVP-15MC 系列运动控制器通过以太网口 LAN2 口与其他从站进行通讯和数据交换。

8.14.2.2 ETH_Link_Manage（开启 MODBUS TCP 数据交换）

FB/FC	说明	适用机种
FB	此指令用于开启 MODBUS TCP 数据交换	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当 Enable 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
Open （启动位）	开启 MODBUS TCP 数据交换。	BOOL	TRUE 或 FALSE（FALSE）	Enable 为 TRUE 时

说明：通过 ETH_Link_Config 指令配置 MODBUS TCP 参数后，执行该指令可以开启 MODBUS TCP 数据交换。

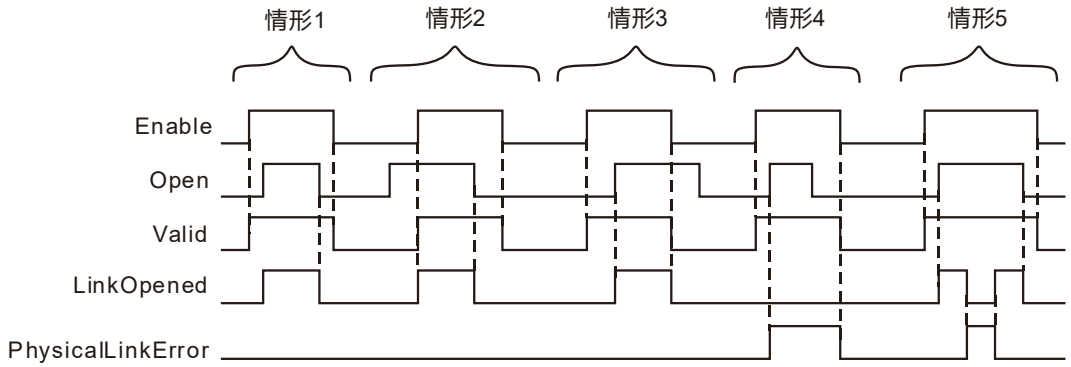
- **输出参数**

名称	功能	数据类型	输出范围
Valid （输出可用）	该输出参数为 TRUE 时表示指令的输出有效。	BOOL	TRUE / FALSE
LinkOpened （连接开启）	该输出参数为 TRUE 时表示开启 MODBUS TCP 数据交换。	BOOL	TRUE / FALSE
PhysicalLinkError （物理连接错误）	该输出参数为 TRUE 时表示主机以太网口物理连接断开。	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
LinkOpened	◆ 当 MODBUS TCP 连接成功时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Open 由 TRUE 变为 FALSE 时 ◆ 当主机以太网口物理连接断开时
PhysicalLinkError	◆ 当 Open 为 TRUE 且主机以太网口物理连接断开时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 LinkOpened 由 False 变为 TRUE 时 ◆ 当主机以太网口物理连接重新连接时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 同时变为 TRUE；当 Open 由 FALSE 变为 TRUE 时，LinkOpened 同时变为 TRUE。当 Open 由 TRUE 变为 FALSE 时，LinkOpened 同时变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Valid 同时变为 FALSE。

情形2： 当 Enable 为 FALSE 时，Open 变为 TRUE，此时指令输出位不会变化；当 Enable 由 FALSE 变为 TRUE 时，Valid 和 LinkOpened 同时变为 TRUE；当 Open 变为 FALSE 时，LinkOpened 同时变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Valid 同时变为 FASLE。

情形3： 当 Enable 变为 TRUE 时，Valid 同时变为 TRUE；当 Open 由 FALSE 变为 TRUE 时，LinkOpened 同时变为 TRUE；当 Enable 由 TRUE 变为 FALSE 时，Valid 和 LinkOpened 同时变为 FALSE，之后再将 Open 由 TRUE 变为 FALSE 时不会影响输出位输出。

情形4： 当 Enable 由 FALSE 变为 TRUE 时，Valid 同时变为 TRUE；若此时 DVP-15MC 系列运动控制器的 LAN2 连接有问题，那么当 Open 变为 TRUE 时，PhysicalLinkError 同时变为 TRUE；当 Open 变为 FALSE 时，PhysicalLinkError 不会变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Valid 和 PhysicalLinkError 同时变为 FALSE。

情形5： 若指令执行过程中 DVP-15MC 系列运动控制器的 LAN2 连接断开时，PhysicalLinkError 位变为 TRUE，同时 LinkOpened 变为 FALSE；当 DVP-15MC 系列运动控制器的 LAN2 连接恢复时，PhysicalLinkError 变为 FALSE，同时 LinkOpended 变为 TRUE；

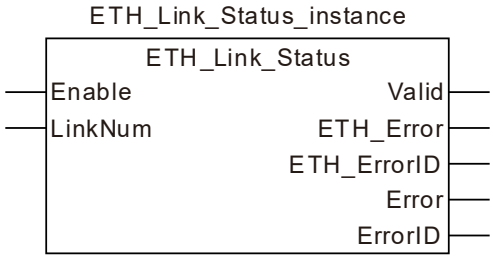
- **功能说明**

此指令用于开启 MODBUS TCP 通讯。V1.01 及以上固件版本支持该功能。

1. ETH_Link_Manage 的 Enable 位变为 TRUE 后，将 Open 设为 TRUE，若主机以太网口物理连接正常，则开始发送 MODBUS TCP 数据，将 Open 由 TRUE 变为 False 时，主机停止发送 MODBUS TCP 数据。
2. 指令正常执行过程中，若将 Enable 设为 FALSE，ETH_Link_Manage 的输出位均变为 FALSE，但主机不会停止发送 MODBUS TCP 数据，只有在 Enable 为 TRUE 时将 Open 设为 FALSE 才能关闭连接。
3. 从站回复错误信息或通讯超时均不会影响该指令执行。

8.14.2.3 ETH_Link_Status（MODBUS TCP 数据交换状态）

FB/FC	说明	适用机种
FB	该指令用于监控编号对应的MODBUS TCP 连接是否出错或从站是否回复错误码。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当 Enable 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
LinkNum（编号）	准备监控的 MODBUS TCP 连接编号。	UINT	1~16（0）	Enable 为 TRUE 时

- **输出参数**

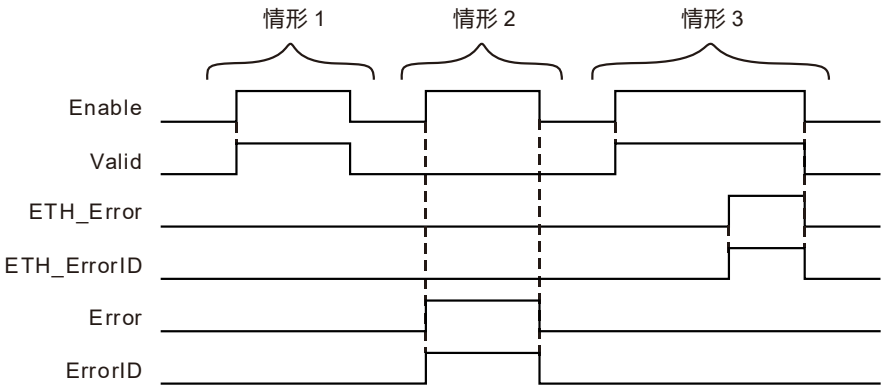
名称	功能	数据类型	输出范围
Valid（输出可用）	该输出参数为 TRUE 时表示指令的输出有效。	BOOL	TRUE / FALSE
ETH_Error（MODBUS TCP 错误）	该输出参数为 TRUE 时表示 MODBUS TCP 数据交换有错误产生。	BOOL	TRUE / FALSE
ETH_ErrorID（MODBUS TCP 错误代码）	MODBUS TCP 数据交换错误信息。 对应的错误代码请参考第 12.2 节。 若 ETH_ErrorID 为 1~10 ,则错误信息请参考相关从站手册说明	WORD	
Error（错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
ETH_Error	◆ 当 MODBUS TCP 数据交换出现错误时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 MODBUS TCP 数据交换恢复正常时
Error	◆ 当指令执行出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当指令参数填写正确时

- **输出参数变化时序图**



情形1： 指令正常执行时，当 Enable 变为 TRUE 时，Valid 同时变为 TRUE；当 Enable 变为 FALSE 时，Valid 同时变为 FALSE。

情形2： 指令执行出错时，当 Enable 变为 TRUE 时，Error 位变为 TRUE，ErrorID 同时变为相应的错误码；当 Enable 变为 FALSE 时，Error 变为 FALSE，同时 ErrorID 变为 0。

情形3： 当 Enable 变为 TRUE 时，Valid 同时变为 TRUE；当执行过程中，MODBUS TCP 发送失败或发生超时时，ETH_Error 变为 TRUE，同时 ETH_ErrorID 变为相应的错误码；当 Enable 变为 FALSE 时，ETH_Error 变为 FALSE，ETH_ErrorID 变为相应错误码。

- **功能说明**

此指令用于监控编号对应的 MODBUS TCP 连接是否出错或从站是否回复错误码。V1.01 及以上固件版本支持该功能。

8.14.2.4 MODBUS TCP 数据交换范例

程序范例一

■ **读、写从站字（WORD）装置范例**

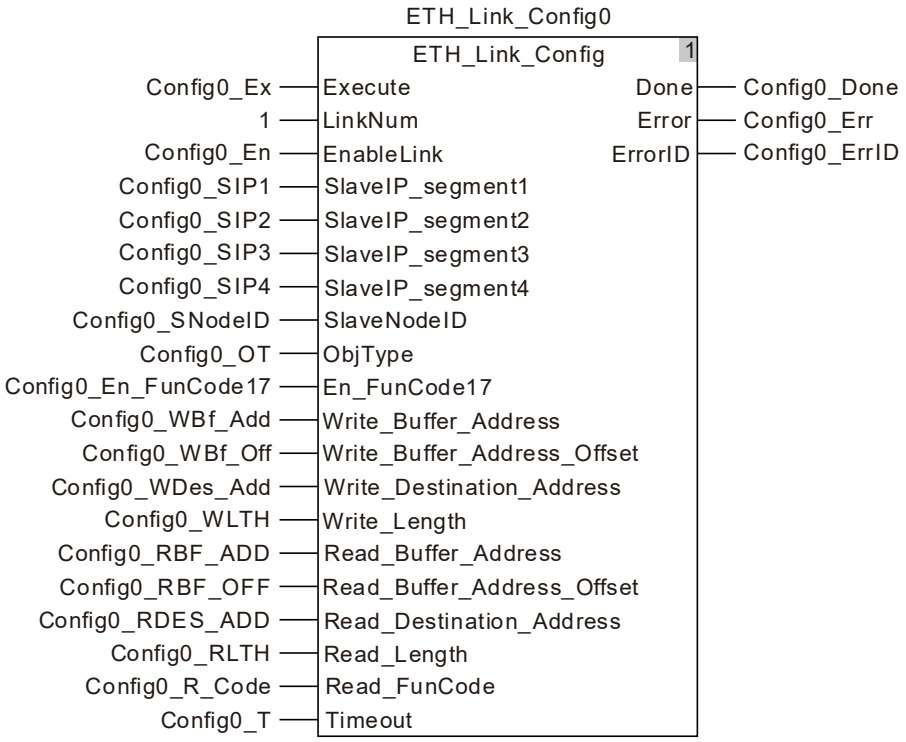
➢ **范例说明**

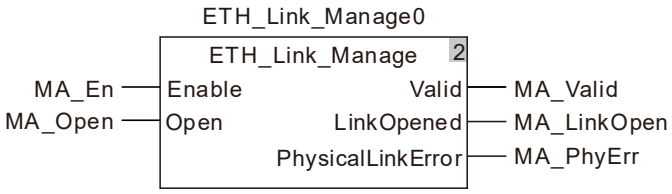
1. DVP-15MC 系列运动控制器作为 MODBUS TCP 主站，DVP12SE 作为 MODBUS TCP 从站，DVP12SE 的 IP 地址为 192.168.1.10。
2. DVP-15MC 系列运动控制器将%MW10~%MW19 的值写到 DVP12SE 的 D0~D9 中，再将 DVP12SE 的 D100~D109 的值读到%MW110~%MW119 中。

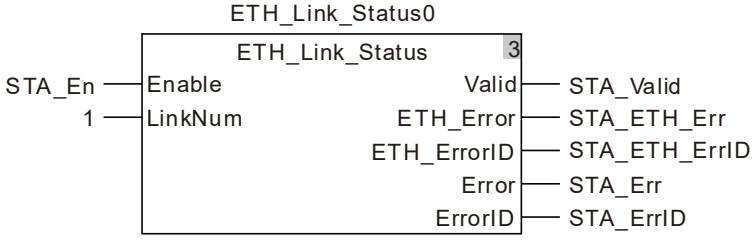
➢ **变量和程序**

变量名	地址	数据类型	初始值
M1		BOOL	
ETH_Link_Config0		ETH_Link_Config	
Config0_Ex		BOOL	
Config0_En		BOOL	TRUE
Config0_SIP1		USINT	192
Config0_SIP2		USINT	168
Config0_SIP3		USINT	1
Config0_SIP4		USINT	10
Config0_SNodeID		USINT	0
Config0_OT		USINT	0
Config0_En_FunCode17		BOOL	FALSE
Config0_WBf_Add	%MW0	UINT	
Config0_WBf_Off		USINT	10
Config0_WDes_Add		UINT	16#1000
Config0_WLTH		UINT	10
Config0_RBF_ADD	%MW100	UINT	
Config0_RBF_OFF		USINT	10
Config0_RDES_ADD		UINT	16#1064
Config0_RLTH		UINT	10
Config0_R_Code		USINT	0
Config0_T		UINT	1000
Config0_Done		BOOL	
Config0_Err		BOOL	
Config0_ErrID		WORD	
ETH_Link_Manage0		ETH_Link_Manage	
MA_En		BOOL	
MA_Open		BOOL	

变量名	地址	数据类型	初始值
MA_Valid		BOOL	
MA_LinkOpen		BOOL	
MA_PhyErr		BOOL	
ETH_Link_Status0		ETH_Link_Status	
STA_En		BOOL	
STA_Valid		BOOL	
STA_ETH_Err		BOOL	
STA_ETH_ErrID		WORD	
STA_Err		BOOL	
STA_ErrID		WORD	







➢ **操作步骤及数据交换说明**

1. 将变量 Config0_WBf_Add 和 Config0_RBF_ADD 分别绑定地址%MW0 和%MW100，Config0_WBf_Off 和 Config0_RBf_OFF 的初始值为 10，编译成功并下载后在线。
2. 先将 M1 设为 TRUE，再将 Config_Ex 设为 TRUE，指令 ETH_Link_Config 执行完成后，将 MA_En 设为 TRUE，然后将 MA_Open 设为 TRUE，指令 ETH_Link_Manage 的输出位 MA_LinkOpen 变为 TRUE 后，DVP-15MC 系列运动控制器开始和 12SE 交换数据。

指令 ETH_Link_Status 可以监控当前通讯的状态。DVP-15MC 系列运动控制器和 DVP12SE 的对应关系如下表所示：

DVP-15MC 系列运动控制器 %MW 装置		DVP12SE 模块 D 寄存器
%MW10	⇨	D0
%MW11		D1
%MW12		D2
…….		……
%MW18		D8
%MW19		D9
%MW110	⇦	D100
%MW111		D101
%MW112		D102
……		……
%MW118		D108
%MW119		D109

程序范例二

■ **读、写从站位（BIT）装置范例**

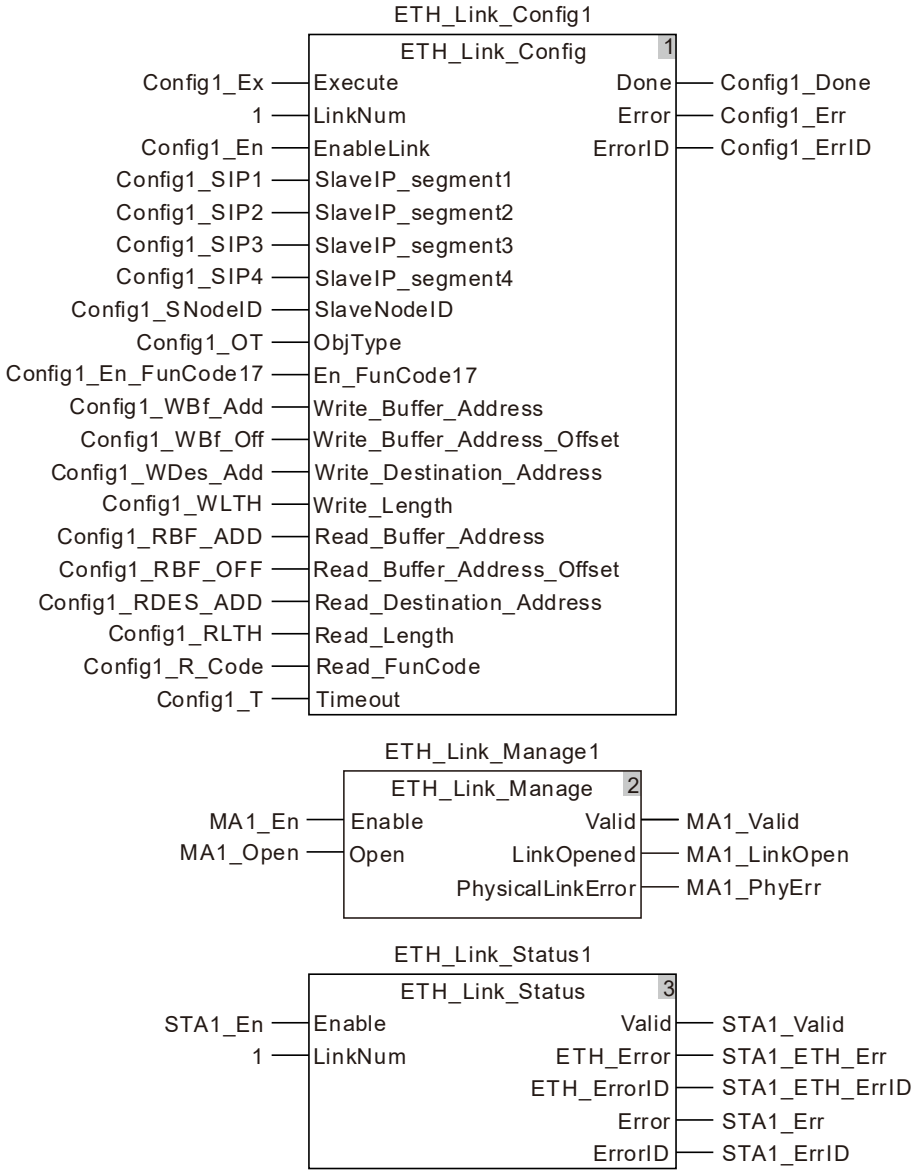
➢ **范例说明**

1. DVP-15MC 系列运动控制器作为 MODBUS TCP 主站，DVP12SE 作为 MODBUS TCP 从站，DVP12SE 的 IP 地址为 192.168.1.10。
2. DVP-15MC 系列运动控制器将%MX0.0~%MX0.7 的值写到 DVP 12SE 的 Y0~Y7 中，再将 DVP12SE 的 Y20~Y27 的状态读到%MX2.0~%MX2.7 中。

➢ **变量和程序**

变量名	地址	数据类型	初始值
M1		BOOL	
ETH_Link_Config1		ETH_Link_Config	
Config1_Ex		BOOL	
Config1_En		BOOL	TRUE
Config1_SIP1		USINT	192

变量名	地址	数据类型	初始值
Config1_SIP2		USINT	168
Config1_SIP3		USINT	1
Config1_SIP4		USINT	10
Config1_SNodeID		USINT	0
Config1_OT		USINT	1
Config1_En_FunCode17		BOOL	FALSE
Config1_WBf_Add	%MW0	UINT	
Config1_WBf_Off		USINT	0
Config1_WDes_Add		UINT	16#0500
Config1_WLTH		UINT	8
Config1_RBF_ADD	%MW1	UINT	
Config1_RBF_OFF		USINT	0
Config1_RDES_ADD		UINT	16#0510
Config1_RLTH		UINT	8
Config1_R_Code		USINT	1
Config1_T		UINT	1000
Config1_Done		BOOL	
Config1_Err		BOOL	
Config1_ErrID		WORD	
ETH_Link_Manage1		ETH_Link_Manage	
MA1_En		BOOL	
MA1_Open		BOOL	
MA1_Valid		BOOL	
MA1_LinkOpen		BOOL	
MA1_PhyErr		BOOL	
ETH_Link_Status1		ETH_Link_Status	
STA1_En		BOOL	
STA1_ Valid		BOOL	
STA1_ETH_Err		BOOL	
STA1_ETH_ErrID		WORD	
STA1_Err		BOOL	
STA10_ErrID		WORD	



➢ **操作步骤及数据交换说明**

1. 将变量 Config1_WBf_Add 和 Config1_RBF_ADD 分别绑定地址%MW0 和%MW1，将 Config1_WLTH 和 Config1_RLTH 的值设为 8，Config1_OT 的值设为 1，Config1_WDes_Add 和 Config1_RDes_Add 分别设为 16#0500 和 16#0510，再将 Config1_R_Code 的值设为 1。
2. 编译成功并下载后在线，先将 M1 设为 TRUE，再 Config_Ex 设为 TRUE，指令 ETH_Link_Config 执行完成后，将 MA_En 设为 TRUE，然后将 MA_Open 设为 TRUE，指令 ETH_Link_Manage 的输出位 MA_LinkOpen 变为 TRUE 后，DVP-15MC 系列运动控制器开始和 12SE 交换数据。

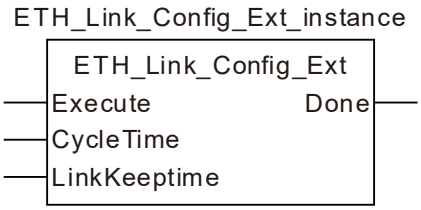
指令 ETH_Link_Status 可以监控当前通讯的状态。DVP-15MC 系列运动控制器和 DVP12SE 的对应关系如下表所示：

DVP-15MC 系列运动控制器装置		DVP12SE 模块位装置 Y
%MX0.0	⇨	Y0
%MX0.1		Y1
%MX0.2		Y2

DVP-15MC 系列运动控制器装置		DVP12SE 模块位装置 Y
…….		……
%MX0.6		Y6
%MX0.7		Y7
%MX2.0	⇦	Y20
%MX2.1		Y21
%MX2.2		Y22
……		……
%MX2.6		Y26
%MX2.7		Y27

8.14.2.5 ETH_Link_Config_Ext（MODBUS TCP 数据交换扩展配置）

FB/FC	说明	适用机种
FB	此指令用于配置 MODBUS TCP 数据交换周期时间与连接保持时间。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
CycleTime（周期时间）	设定主站发送 MODBUS TCP 数据的周期时间（单位：毫秒）	ARRAY [1..16] OF UINT	0~65535 (0)	Execute 由 FALSE 变为 TRUE 时
LinkKeeptime（连接保持时间）	设定主站的连接保持时间（单位：秒）	ARRAY [1..16] OF UINT	0~65535（0）	Execute 由 FALSE 变为 TRUE 时

说明：

1. 参数 CycleTime 和参数 LinkKeeptime 中每一个元素均与指令 ETH_Link_Config 中的编号相对应。如参数 CycleTime 的第一个元素对应编号为 1 的数据发送周期时间。
2. 参数 CycleTime 的单位为毫秒（ms），参数 LinkKeepTime 的单位为秒（s）。
3. 参数 LinkKeeptime 表示连接保持时间（单位为秒），当主站与从站之间超过此时间没有数据交换时，控制器会自动断开与从站的连接。

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当 Execute 为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **功能说明**

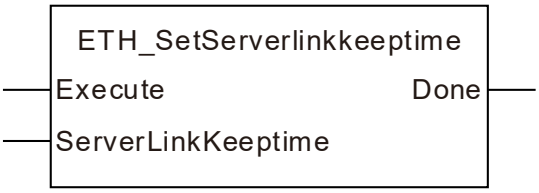
  此指令用于设定 MODBUS TCP 数据交换周期时间及连接保持时间。V1.01 及以上固件版本支持该功能。

  1. 用户使用控制器的 MODBUS TCP 主站功能时，可以使用指令 ETH_Link_Config_Ext 设置 MODBUS TCP 数据发送周期以及连接保持时间。用户在配置 MODBUS TCP 参数时可以不使用该指令，控制器 MODBUS TCP 主站功能仍可使用，此时 MODBUS TCP 数据发送完一笔数据后立即发送下一笔数据且连接保持时间默认为 30s。
  2. 指令执行过程中修改参数并不会被写入，当用户修改指令参数后，必须重新触发执行位 Execute 后，参数才会被写入。
  3. 指令 ETH_Link_Config_Ext 与指令 ETH_Link_Config 没有执行先后顺序的关系。指令 ETH_Link_Config_Ext 在指令 ETH_Link_Manage 之前执行，指令 ETH_Link_Config_Ext 参数才会生效。

8.14.2.6 ETH_ SetServerlinkkeeptime（设置控制器做为 MODBUS TCP 从站连接保持时间）

FB/FC	说明	适用机种
FB	此指令用于设置控制器做为 MODBUS TCP 从站时的连接保持时间。	DVP15MC11T DVP15MC11T-06

ETH_SetServerlinkkeeptime_instance



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，指令开始执行	BOOL	TRUE 或 FALSE（FALSE）	
ServerLinkKeeptime（从站连接保持时间）	控制器做为 MODBUS TCP 从站时连接保持时间（单位：秒）	UINT	0~65535（0）	Execute 由 FALSE 变为 TRUE 时

说明：参数 ServerLinkKeeptime 设置为 0 或为空时，控制器做为 MODBUS TCP 从站的连接保持时间默认为 30s。

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当 Execute 由 FALSE 变为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时

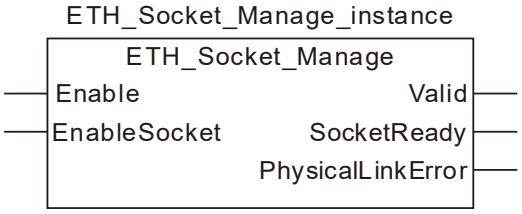
- **功能说明**

此指令用于设置控制器做为 MODBUS TCP 从站时与主站的连接保持时间。

1. 指令设置的连接保持时间是控制器做为 MODBUS TCP 从站使用时才有效，当控制器做 MODBUS TCP 主站时，执行该指令不影响 MODBUS TCP 主站功能。
2. 当控制器从站与主站之间在 ServerLinkKeeptime 设置的时间内没有数据交换时，控制器从站会主动断开与 MODBUS TCP 主站的连接。
3. 执行该指令仅能设置 DVP-15MC 系列运动控制器 LAN2 口做 MODBUS TCP 从站时的连接保持时间。

8.14.2.7 ETH_Socket_Manage（Socket TCP/UDP 管理）

FB/FC	说明	适用机种
FB	此指令用于管理 Socket TCP/UDP。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当 Enable 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE （FALSE）	
EnableSocket （启动位）	开启 Socket。	BOOL	TRUE 或 FALSE （FALSE）	Enable 为 TRUE 时

说明：指令 ETH_Socket_Manage 用于开启 Socket 功能，该指令执行后，其他 Socket 相关指令才能够正常执行·。

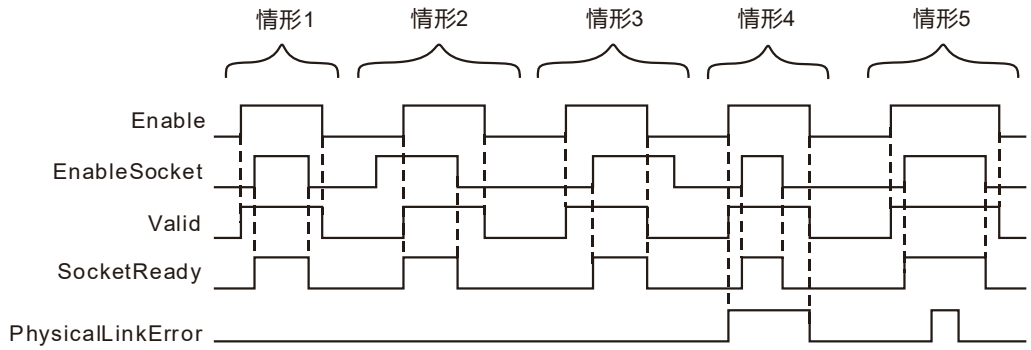
- **输出参数**

名称	功能	数据类型	输出范围
Valid （输出可用）	该输出参数为 TRUE 时表示指令的输出有效。	BOOL	TRUE/FALSE
SokcetReady （Socket 功能开启）	该输出参数为 TRUE 时表示 Socket 开启成功。	BOOL	TRUE/FALSE
PhysicalLinkError （物理连接断开）	该输出参数为 TRUE 时表示主机以太网口物理连接断开。	BOOL	TRUE/FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
SocketReady	◆ 当 Socket 开启成功时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 EnableSocket 由 TRUE 变为 FALSE 时
PhysicalLinkError	◆ 当 Enable 为 TRUE 且主机 LAN2 以太网口物理连接断开时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当主机 LAN2 以太网口物理连接恢复时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时， Valid 同时变为 TRUE；当 EnableSocket 由 FALSE 变为 TRUE 时，SocketReady 同时变为 TRUE。当 EnableSocket 由 TRUE 变为 FALSE 时，SocketReady 同时变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Valid 同时变为 FALSE。

情形2： 当 Enable 为 FALSE 时，EnableSocket 变为 TRUE，此时指令输出位不会变化；当 Enable 由 FALSE 变为 TRUE 时，Valid 和 SocketReady 同时变为 TRUE；当 EnableSocket 变为 FALSE 时，SocketReady 同时变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Valid 同时变为 FASLE。

情形3： 当 Enable 变为 TRUE 时，Valid 同时变为 TRUE；当 EnableSocket 由 FALSE 变为 TRUE 时，SocketReady 同时变为 TRUE；当 Enable 由 TRUE 变为 FALSE 时，Valid 和 SocketReady 同时变为 FALSE，之后再将 EnableSocket 由 TRUE 变为 FALSE 时不会影响输出位输出。

情形4： 当 Enable 由 FALSE 变为 TRUE 时，Valid 同时变为 TRUE；若此时 DVP-15MC 系列运动控制器的 LAN2 口连接有问题，那么当 Enable 变为 TRUE 时，PhysicalLinkError 同时变为 TRUE；当 EnableSocket 变为 TRUE 时，SocketReady 同时变为 TRUE，PhysicalLinkError 状态不变；当 EnableSocket 由 TRUE 变为 FALSE 时，SocketReady 同时变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Valid 和 PhysicalLinkError 同时变为 FALSE。

情形5： 若指令执行过程中主机 LAN2 口连接断开时，PhysicalLinkError 位变为 TRUE，同时 SocketReady 状态不变；当主机 LAN2 口连接恢复时，PhysicalLinkError 变为 FALSE。

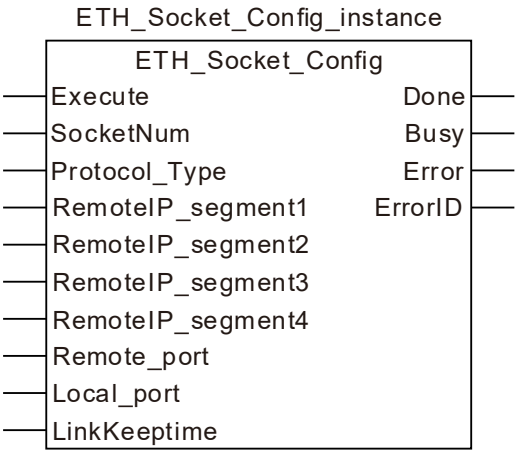
- **功能说明**

此指令用于开启 Socket。V1.02 及以上固件版本支持该功能。

1. 指令正常执行过程中，若将 Enable 设为 FALSE，ETH_Link_Manage 的输出位均变为 FALSE，但主机不会关闭 Socket，只有在 Enable 为 TRUE 时将 EnableSocket 设为 FALSE 才能关闭 Socket。
2. 目标回复错误信息或通讯超时均不会影响该指令执行。

8.14.2.8 ETH_Socket_Config（Socket 数据交换配置）

FB/FC	说明	适用机种
FB	此指令用于配置 Socket 参数。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Execute （执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE （FALSE）	
SocketNum （编号）	设定 Socket 编号。	USINT	1~8 （0）	Execute 由 FALSE 变为 TRUE 时
Protocol_Type （方式选择）	Socket 连接方式： 0：Socket UDP 1：Socket TCP	USINT	0：Socket UDP 1：Socket TCP （0）	Execute 由 FALSE 变为 TRUE 时
RemoteIP_segment1 （远端 IP 地址第一段）	设定远端 IP 地址的第一段。	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
RemoteIP_segment2 （远端 IP 地址第二段）	设定远端 IP 地址的第二段。	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
RemoteIP_segment3 （远端 IP 地址第三段）	设定远端 IP 地址的第三段。	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
RemoteIP_segment4 （远端 IP 地址第四段）	设定远端 IP 地址的第四段。	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
Remote_port （远端端口）	设定远端端口号	UINT	0~65535 （0）	Execute 由 FALSE 变为 TRUE 时
Local_port （本地端口）	设定本地端口号	UINT	0~65535 （0）	Execute 由 FALSE 变为 TRUE 时

名称	功能	数据类型	设定范围（缺省值）	生效时机
LinkKeeptime（设定连接保持时间）	设定连接保持时间（s）	UINT	65535（0）	Execute 由 FALSE 变为 TRUE 时

说明：

1. 输入参数 RemoteIP_segment1、RemoteIP_segment2、RemoteIP_segment3、RemoteIP_segment4 分别为目标 IP 地址的第一段到第四段，例如目标 IP 为 192.168.1.10，那么参数 SlaveIP_segment1 输入值 192，参数 SlaveIP_segment2 输入值 168，参数 SlaveIP_segment3 输入值 1，参数 SlaveIP_segment4 输入值 10；
2. 参数 LinkKeeptime 表示连接保持时间（单位为秒），当建立的连接超过此时间没有资料传输时，主机会自动中断连线。

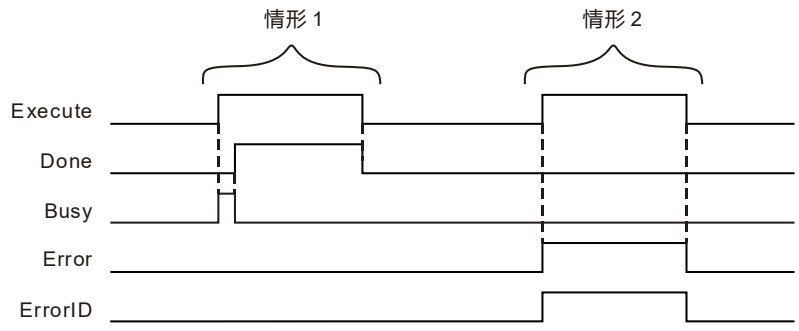
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示参数配置完成。	BOOL	TRUE/FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE/FALSE
Error（错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE/FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当参数配置完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 由 FALSE 变为 TRUE 时
Error	◆ 当指令输入参数不合法时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时， Busy 变为 TRUE，参数写入成功，一个周期后 Done 变为 TRUE，同时 Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时， Done 同时由 TRUE 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，若参数不合法，则 Error 同时变为 TRUE，ErrorID 显示对应的错误码；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

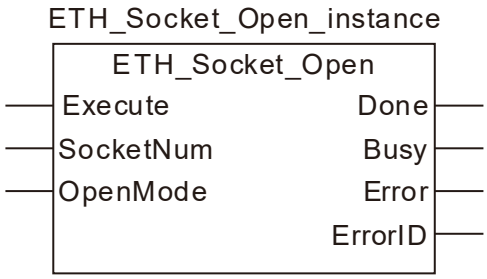
- **功能说明**

此指令用于 Socket 参数配置。V1.02 及以上固件版本支持该功能。

1. 指令执行过程中修改参数并不会被写入，当用户修改指令参数后，必须重新触发执行位 Execute 后，参数才会被写入。
2. 当前连接未断开时，重新触发指令执行时，指令报错。
3. 参数 Local_port 填 0 时，主机会自动分配本地端口给当前连接。当主机选择的是 Socket TCP 服务器模式时，参数 Local_port 不能设置为 0，必须设置一个本地端口号 (非 502 端口) 让主机监听该端口号。
4. 当主机选择 Socket TCP 服务器模式时 ,若参数 RemoteIP_segment1~4 或 Remote_port 未设置为 0，则主机会按照参数 RemoteIP_segment1~4 或 Remote_port 的值来筛选远端主机，当远端主机的 IP 地址或端口号与参数 RemoteIP_segment1~4 或 Remote_port 不一致时，主机会主动断开连接，并停止监听当前连接设置的本地端口。

8.14.2.9 ETH_Socket_Open（开启 Socket）

FB/FC	说明	适用机种
FB	此指令用于开启 Socket TCP/UDP。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Enable 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
SocketNum（Socket 编号）	设定 Socket 编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE 时
OpenMode（开启模式）	开启模式： TRUE：客户端模式； FALSE：服务器模式；	BOOL	TRUE 或 FALSE（FALSE）	Execute 由 FALSE 变为 TRUE 时

说明：

1. 指令 ETH_Socket_Open 用于开启 Socket，执行该指令后控制器尝试与其他节点建立连接或等待其他节点发送连线请求；
2. OpenMode 为 TRUE 时，控制器为客户端模式，指令 ETH_Socket_Open 执行后， 控制器会主动发送连线请求给目标节点；OpenMode 为 FALSE 时，控制器为服务器模式，指令 ETH_Socket_Open 执行后，控制器等待目标节点的连线请求。
3. 若 Socket 模式选择 Socket UDP 模式时，OpenMode 选择 FALSE 或 TRUE 不影响使用。

- **输出参数**

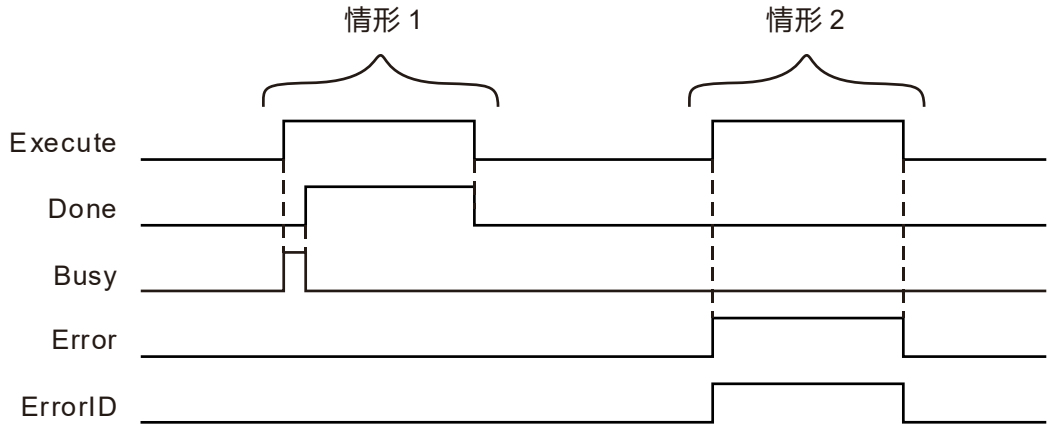
名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完毕。	BOOL	TRUE/FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行。	BOOL	TRUE/FALSE
Error（错误）	该参数为 TRUE 时表示指令执行出错。	BOOL	TRUE/FALSE
ErrorID	指令执行出错时的错误代码。	WORD	

名称	功能	数据类型	输出范围
（错误代码）	对应的错误代码请参考第 12.2 节。		

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 由 FALSE 变为 TRUE 时	◆ 当 Done 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时， Busy 变为 TRUE，参数写入成功，一个周期后 Done 变为 TRUE ,同时 Busy 变为 FALSE ;当 Execute 由 TRUE 变为 FALSE 时，Done 同时由 TRUE 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，若参数不合法，则 Error 同时变为 TRUE，ErrorID 显示对应的错误码；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

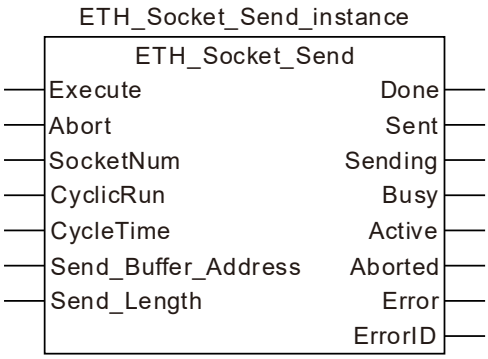
- **功能说明**

此指令用于建立 TCP 连接或启用 UDP 功能。V1.02 及以上固件版本支持该功能。

1. 指令 ETH_Socket_Open 正常执行后 指令 ETH_Socket_Send 和 ETH_Socket_Receive 才能够执行。
2. 指令执行后，用户可以通过指令 ETH_Socket_Status 查看当前连接状态。

8.14.2.10 ETH_Socket_Send（Socket 数据发送）

FB/FC	说明	适用机种
FB	此指令用于发送 Socket 数据。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Execute （执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE （FALSE）	
Abort （打断）	打断指令执行	BOOL	TRUE 或 FALSE （FALSE）	
SocketNum （Socket 编号）	指定 Socket 编号	USINT	1~8 （0）	Execute 由 FALSE 变为 TRUE 时
CyclicRun （周期发送）	设定是否周期发送数据	BOOL	TRUE 或 FALSE （FALSE）	Execute 由 FALSE 变为 TRUE 时
CycleTime （周期时间）	设定周期时间（ms）	UINT	0~65535 （0）	Execute 由 FALSE 变为 TRUE 时
Send_Buffer_Address （发送数据存放地址）	设定存放发送数据的起始装置	USINT	%MB0~%MB65535	Execute 由 FALSE 变为 TRUE 时
Send_Length （发送数据长度）	设定发送数据长度（Byte 数）	UINT	0~200 （0）	Execute 由 FALSE 变为 TRUE 时

说明：

1. 输入参数 Send_Buffer_Address 表示主机发送数据存放的装置首地址，参数 Send_Length 表示主机发送的数据长度，两个输入引脚不可缺省。

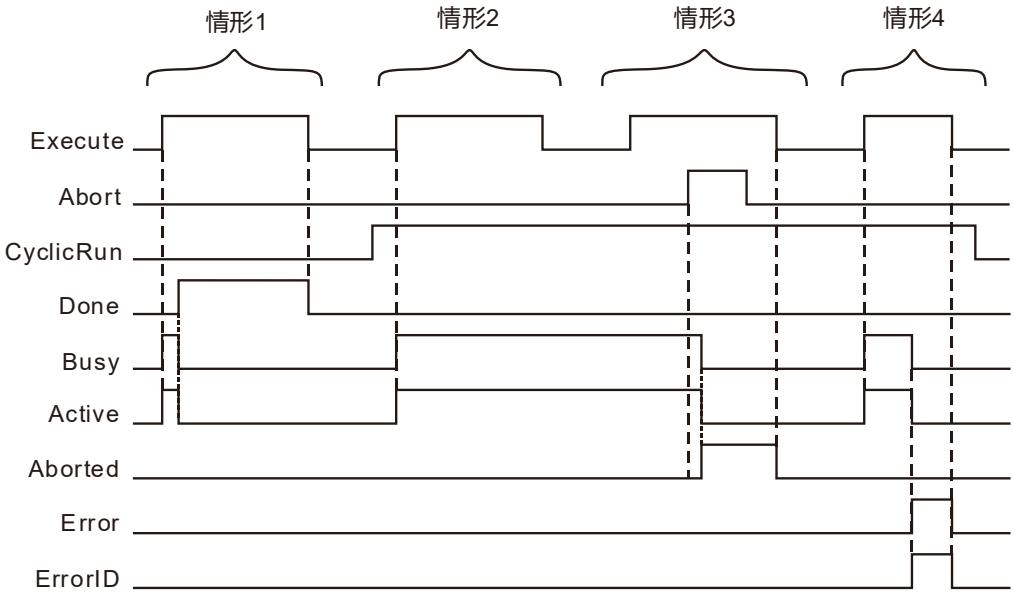
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE/FALSE
Sent （发送完成）	该输出参数为 TRUE 时表示 Socket 数据发送完成。	BOOL	TRUE/FALSE
Sending （正在发送）	该输出参数为 TRUE 时表示 Socket 数据正在发送。	BOOL	TRUE/FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE/FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正控制主机发送数据。	BOOL	TRUE/FALSE
Aborted （中断）	该输出参数为 TRUE 时表示指令执行被打断。	BOOL	TRUE/FALSE
Error （错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE/FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Sent	◆ 当一笔 Socket 数据发送完成时	◆ 当主机开始发送另一笔数据时
Sending	◆ 当一笔 Socket 正在发送时	◆ 当主机发送一笔数据完成时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 由 FALSE 变为 TRUE 时 ◆ 当 Aborted 由 FALSE 变为 TRUE 时 ◆ 当 Error 有 FALSE 变为 TRUE 时
Active	◆ 当指令控制主机发送数据时	◆ 当 Done 由 FALSE 变为 TRUE 时 ◆ 当 Aborted 由 FALSE 变为 TRUE 时 ◆ 当 Error 有 FALSE 变为 TRUE 时
Aborted	◆ 当指令执行被打断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当选择发送一笔的模式时，当 Execute 由 FALSE 变为 TRUE 时，Busy 和 Active 同时变为 TRUE；当一笔 Socket 数据发送完成时，Done 变为 TRUE，同时 Busy 和 Active 同时变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Done 同时变为 FALSE。

情形2： 循环发送模式下，当 Execute 由 FALSE 变为 TRUE 时，Busy 和 Active 同时变为 TRUE，指令开始控制主机发送 Socket 数据；当 Execute 由 TRUE 变为 FALSE 时，Busy 和 Active 保持 TRUE 不变。

情形3： 情形 2 之后再次将 Execute 由 FALSE 变为 TRUE，输出位状态保持不变，将 Abort 由 FALSE 变为 TRUE，一个周期后，Aborted 变为 TRUE，Busy 和 Acitve 同时变为 FALSE；当 Abort 由 TRUE 变为 FALSE 时，输出位状态不变，当 Execute 由 TRUE 变为 FALSE 时，Aborted 同时变为 FALSE。

情形4： 当 Execute 由 FALSE 变为 TRUE 时，Busy 和 Active 同时变为 TRUE；当指令执行有误时，Error 变为 TRUE，ErrorID 同时显示相应错误码，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE，同时 ErrorID 变为 0。

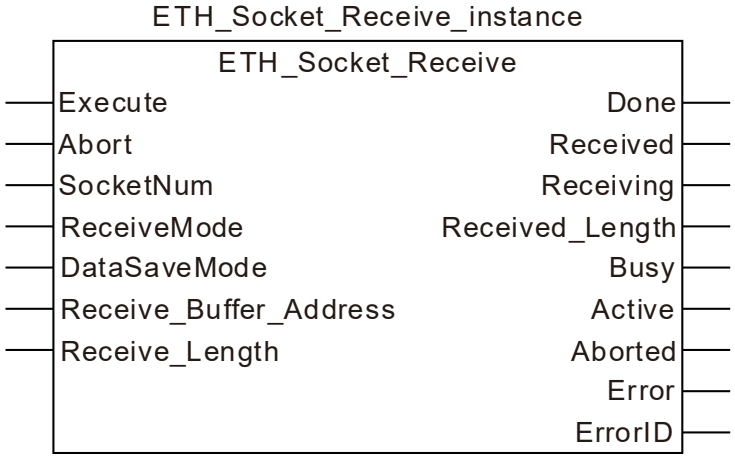
- **功能说明**

此指令用于发送 Socket 数据。V1.02 及以上固件版本支持该功能。

1. 输入参数 CyclicRun 表示是否周期发送数据，参数 CycleTime 表示周期时间。当 CyclicRun 为 TRUE 时，执行指令 ETH_Socket_Send，主机每隔参数 CycleTime 设置的时间发送一笔数据；当 CyclicRun 为 FALSE 时，执行指令 ETH_Socket_Send，主机仅发送一笔数据。
2. 参数 CycleTime 值为 0 且参数 CyclinRun 为 TRUE 时，主机仍会循环发送数据且没有时间间隔限制。
3. 指令执行过程中，改变指令输入参数然后重新触发执行指令，新的输入参数不会生效，必须先使用输入参数 Abort 将指令打断，然后在重新执行指令，新的输入参数才会生效。
4. 参数 Send_Length 不能设为 0，否则执行指令报错。

8.14.2.11 ETH_Socket_Receive（Socket 数据接收）

FB/FC	说明	适用机种
FB	此指令用于接收 Socket 数据。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
Abort（打断）	打断指令执行	BOOL	TRUE 或 FALSE（FALSE）	
SocketNum（Socket 编号）	指定 Socket 编号	USINT	1~8（0）	Execute 由 FALSE 变为 TRUE 时
ReceiveMode（接收模式）	设定数据接收模式： 0：持续接收数据； 1：只接收一笔数据；	USINT	0、1（0）	Execute 由 FALSE 变为 TRUE 时
DataSaveMode（数据保存模式）	设定数据保存模式： 0：拼接模式； 1：覆盖模式；	USINT	0、1（0）	Execute 由 FALSE 变为 TRUE 时
Receive_Buffer_Address（接收数据存放地址）	设定存放接收数据的起始装置	USINT	%MB0~%MB65535	Execute 由 FALSE 变为 TRUE 时
Receive_Length（接收数据长度）	设定接收数据长度（Byte 数）	UINT	0~200（200）	Enable 为 TRUE 时

说明：输入参数 Receive_Buffer_Address 表示主机接收数据存放的装置首地址，参数 Receive_Length 表示主机接收的数据长度，两个输入引脚不可缺省。

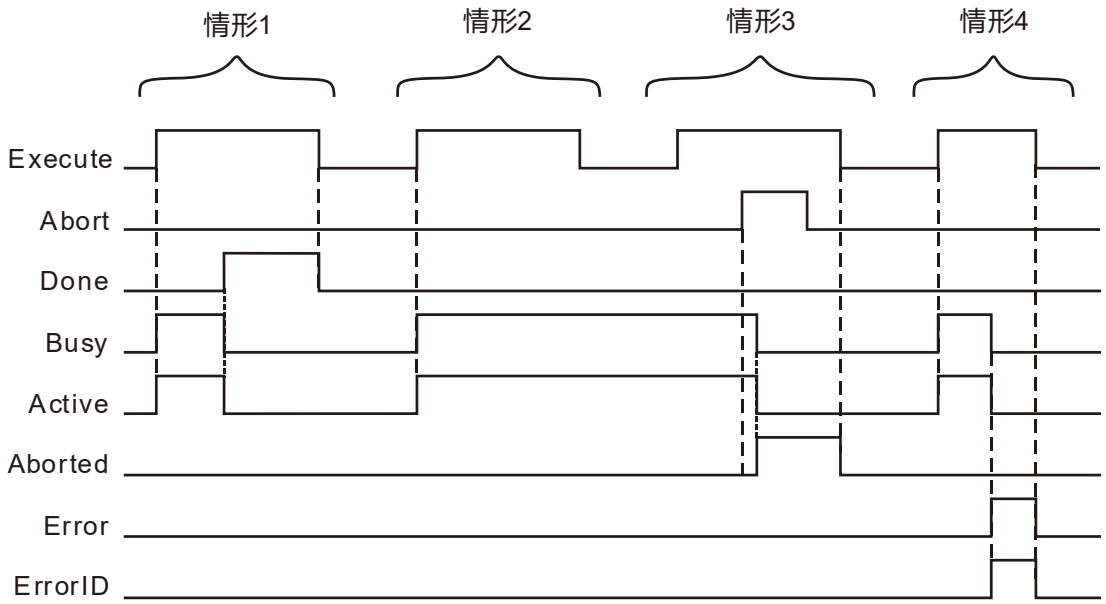
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE/FALSE
Received （接收完成）	该输出参数为 TRUE 时表示 Socket 数据接收完成。	BOOL	TRUE/FALSE
Receiving （正在接收）	该输出参数为 TRUE 时表示 Socket 数据正在接收。	BOOL	TRUE/FALSE
Received_Length （实际接收长度）	该参数表示实际接收的数据长度。	UINT	
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE/FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制主机接收数据。	BOOL	TRUE/FALSE
Aborted （中断）	该输出参数为 TRUE 时表示指令执行被打断。	BOOL	TRUE/FALSE
Error （错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE/FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Received	◆ 当一笔 Socket 数据接收完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令开始接收下一笔数据时
Receiving	◆ 当一笔 Socket 数据正在接收时	◆ 当一笔 Socket 数据接收完成时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 由 FALSE 变为 TRUE 时 ◆ 当指令执行被打断时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Active	◆ 当指令控制主机接收数据时	◆ 当 Done 由 FALSE 变为 TRUE 时 ◆ 当指令执行被打断时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Aborted	◆ 当指令执行被打断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当指令使用接收一笔数据或持续拼接模式时，当 Execute 由 FALSE 变为 TRUE 时，Busy 和 Active 同时变为 TRUE；当一笔数据接收完毕或拼接长度达到或超出设定长度时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Done 同时变为 FALSE。

情形2： 当指令选择持续覆盖模式时，当 Execute 由 FALSE 变为 TRUE 时，Busy 和 Active 同时变为 TRUE；当 Execute 由 TRUE 变为 FALSE 时，Busy 和 Active 保持 TRUE 状态不变，同时指令继续接收数据。

情形3： 在情形 2 之后再次将 Execute 由 FALSE 变为 TRUE，输出位保持不变；将 Abort 由 FALSE 变为 TRUE，一个周期后，Aborted 由 FALSE 变为 TRUE，同时 Busy 和 Active 变为 FALSE；当 Abort 由 TRUE 变为 FALSE 时，输出位状态保持不变；当 Execute 由 TRUE 变为 FALSE 时，Aborted 同时变为 FALSE。

情形4： 当 Execute 由 TRUE 变为 FALSE 时，Busy 和 Active 同时变为 TRUE；当指令报错时，Error 变为 TRUE，ErrorID 显示相应错误码，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE，同时 ErrorID 变为 0。

- **功能说明**

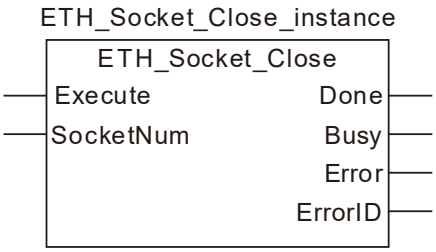
此指令用于接收 Socket 数据。V1.02 及以上固件版本支持该功能。

1. 输入参数 ReceiveMode 为 0 时，指令可以持续接收 Socket 数据，当 DataSaveMode 为 0 时，指令将接收到的数据以拼接的方式存放到 Receive_Buffer_Address 开始的装置中，主机会从上次数据存放的最后一个装置后面开始存放新接收到的数据，当接收数据的总长度大于等于 Receive_Length 时，指令输出参数 Done 变为 TRUE，指令执行完成。
2. 当输入参数 ReceiveMode 为 0、DataSaveMode 为 1 时，指令将接收到的数据以覆盖的方式存到 Receive_Buffer_Address 开始的装置中，主机会将每次接收到的数据从 Receive_Buffer_Address 指定的装置开始存放。

3. 当指令接收到的第一笔数据的长度超过 Receive_Length 设定的长度时，主机会先将 Receive_Length 设定长度的数据写入指定的装置中，超出长度的数据则舍弃，然后指令报警。
4. 指令执行过程中，修改指令的输入参数然后重新触发执行位，新的输入参数并不会生效，必须先使用输入参数 Abort 将指令打断，然后在重新执行指令，新的输入参数才会生效。

8.14.2.12 ETH_Socket_Close（关闭 Socket）

FB/FC	说明	适用机种
FB	此指令用于关闭 Socket。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute （执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE （FALSE）	
SocketNum （Socket 编号）	指定 Socket 编号	USINT	1~8 （0）	Execute 由 FALSE 变为 TRUE 时

说明：通过 ETH_Socket_Close 可以关闭 TCP 连接或 UDP 功能。

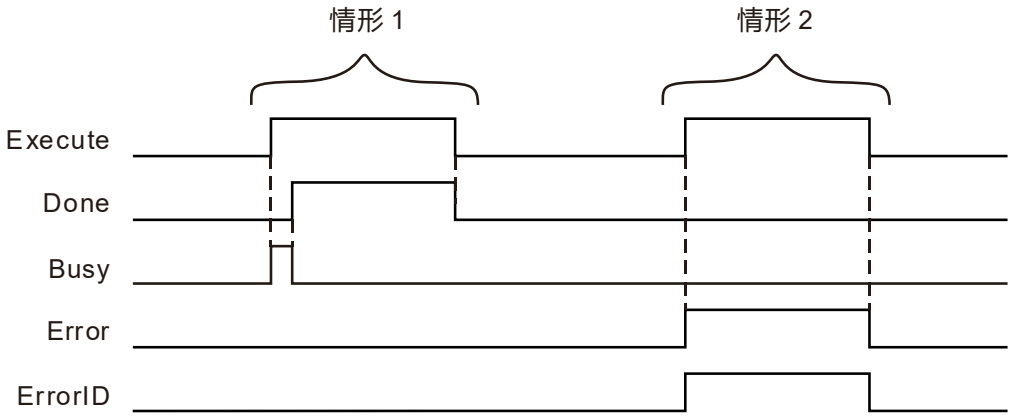
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE/FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE/FALSE
Error （错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE/FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 由 FALSE 变为 TRUE 时	◆ 当 Done 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时， Busy 变为 TRUE，参数写入成功，一个周期后 Done 变为 TRUE ,同时 Busy 变为 FALSE ;当 Execute 由 TRUE 变为 FALSE 时，Done 同时由 TRUE 变为 FALSE。

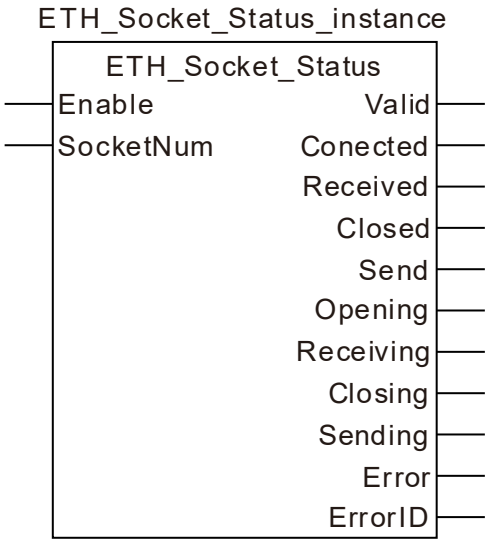
情形2： 当 Execute 由 FALSE 变为 TRUE 时，若参数不合法，则 Error 同时变为 TRUE，ErrorID 显示对应的错误码；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

- **功能说明**

此指令用于断开 TCP 连接或关闭 UDP 功能。V1.02 及以上固件版本支持该功能。

8.14.2.13 ETH_Socket_Status（读取 Socket 状态）

FB/FC	说明	适用机种
FB	此指令用于读取 Socket 当前状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当 Enable 为 TRUE 时，指令执行。	BOOL	TRUE 或 FALSE（FALSE）	
SocketNum（编号）	准备监控的 Socket 编号。	UINT	1~8（0）	当 Enable 为 TRUE 时

说明：该指令用于监控编号对应的 Socket 当前状态以及是否有错误产生。

- **输出参数**

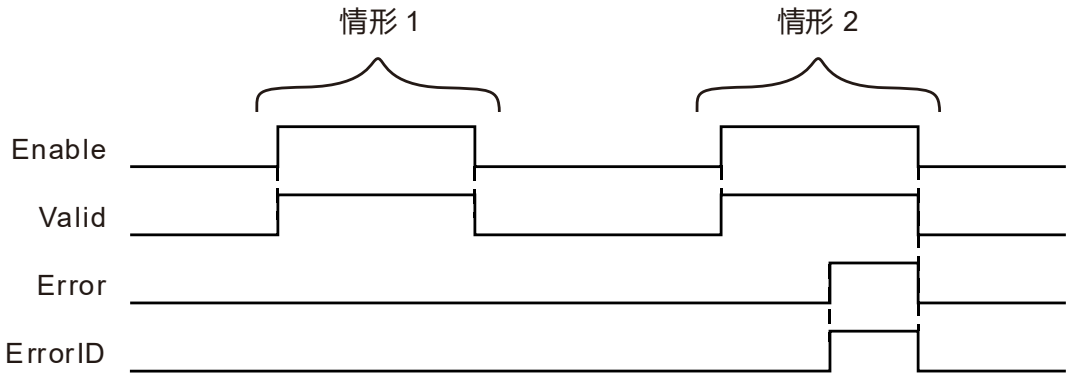
名称	功能	数据类型	输出范围
Valid（执行中）	该输出参数为TRUE时表示Socket发送和接收均完成。	BOOL	TRUE/FALSE
Connected（连接成功）	该输出参数为 TRUE 时表示与目标连接成功。	BOOL	TRUE/FALSE
Received（接收成功）	该输出参数为 TRUE 时表示接收数据成功。	BOOL	TRUE/FALSE
Closed（连接关闭）	该输出参数为 TRUE 时表示连接已关闭。	BOOL	TRUE/FALSE
Send（发送成功）	该输出参数为 TRUE 时表示数据已发送。	BOOL	TRUE/FALSE

名称	功能	数据类型	输出范围
Opening （正在建立连接）	该输出参数为 TRUE 时表示正在与目标建立连接。	BOOL	TRUE/FALSE
Receiving （正在接收）	该输出参数为 TRUE 时表示正在接收数据。	BOOL	TRUE/FALSE
Closing （正在关闭当前连接）	该输出参数为 TRUE 时表示正在关闭当前连接。	BOOL	TRUE/FALSE
Sending （正在发送数据）	该输出参数为 TRUE 时表示发送或接收超时。	BOOL	TRUE/FALSE
Error （错误）	该输出参数为 TRUE 时表示指令执行出错。	BOOL	TRUE/FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 由 FALSE 变为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
Connected	◆ 当 Socket 连接完成时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Closing 由 FALSE 变为 TRUE 时
Received	◆ 当一笔数据接收完成时	◆ 当 Receiving 由 FALSE 变为 TRUE 时 ◆ 当 Enable 由 TRUE 变为 FALSE 时
Closed	◆ 当 Socket 连接关闭时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Opening 由 FALSE 变为 TRUE 时
Send	◆ 当一笔数据发送完成时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Sending 有 TRUE 变为 FALSE 时
Opening	◆ 当开始建立 Socket 连接或等待连接时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Connected 由 FALSE 变为 TRUE 时
Receiving	◆ 当主机接收一笔数据时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Received 由 FALSE 变为 TRUE 时
Closing	◆ 当 Socket 关闭时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Closed 由 FALSE 变为 TRUE 时
Sending	◆ 当一笔数据正在发送时	◆ 当 Enable 由 TRUE 变为 FALSE 时
Error	◆ 当指令执行或 Socket 出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 同时变为 TRUE ;当 Enable 由 TRUE 变为 FALSE 时，Valid 同时变为 FALSE。

情形2： 当 Enable 由 FALSE 变为 TRUE 时，Valid 同时变为 TRUE；当执行指令时或执行过程中出错时，Error 变为 TRUE ,ErrorID 显示错误码 ,同时 Valid 保持 TRUE 不变 ;当 Enable 由 TRUE 变为 FALSE 时，Valid 和 Error 同时变为 FALSE，ErrorID 变为 0。

- **功能说明**

此指令用于监控编号对应的 Socket 当前的状态。V1.02 及以上固件版本支持该功能。

8.14.2.14 以太网自由协议范例

程序范例

■ **ETH_Socket_Config 指令的使用范例如下：**

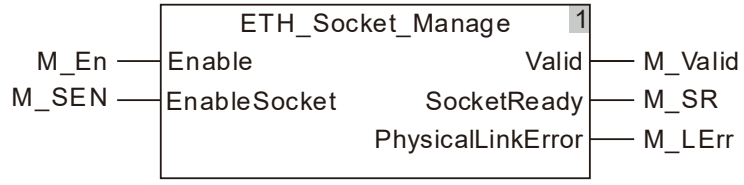
➢ **变量和程序**

变量名	地址	数据类型	初始值
TEST_BEGIN		BOOL	
ETH_Socket_Manage0		ETH_Socket_Manage	
M_En		BOOL	
M_SEN		BOOL	
M_Valid		BOOL	
M_SR		BOOL	
M_LErr		BOOL	
ETH_Socket_Config0		ETH_Socket_Config	
CON_Ex		BOOL	
CON_PType		USINT	0
CON_IP1		USINT	192
CON_IP2		USINT	168
CON_IP3		USINT	1
CON_IP4		USINT	10
CON_RP		UINT	502
CON_LP		UINT	502
CON_KT		UINT	30
CON_Done		BOOL	
CON_Bsy		BOOL	
CON_Err		BOOL	
CON_ErrID		WORD	
ETH_Socket_Open0		ETH_Socket_Open	
O_Ex		BOOL	
O_SMode		BOOL	TRUE
O_Done		BOOL	
O_Bsy		BOOL	
O_Err		BOOL	
O_ErrID		WORD	
ETH_Socket_Send0		ETH_Socket_Send	
S_Ex		BOOL	
S_Abort		BOOL	
S_CR		BOOL	TRUE

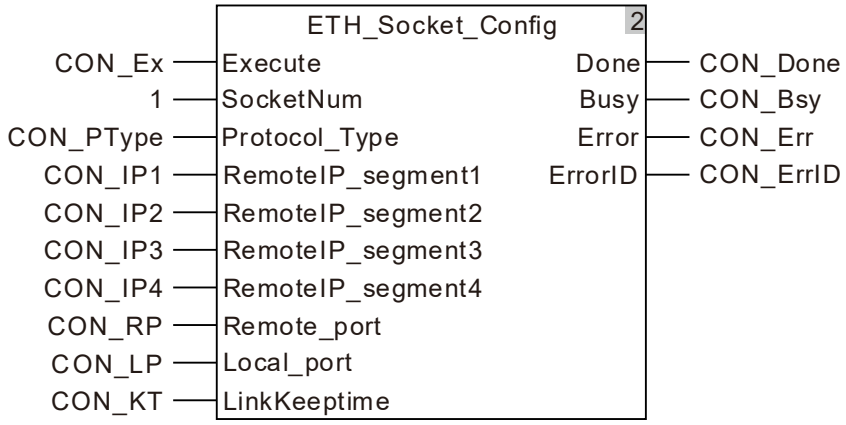
变量名	地址	数据类型	初始值
S_CT		UINT	100
S_BFA	%MB200	USINT	
S_LTH		UINT	17
S_Done		BOOL	
S_Sent		BOOL	
S_Sending		BOOL	
S_Bsy		BOOL	
S_Act		BOOL	
S_Abt		BOOL	
S_Err		BOOL	
S_ErrID		WORD	
ETH_Socket_Receive0		ETH_Socket_Receive	
R_Ex		BOOL	
R_Abort		BOOL	
R_Mode		USINT	0
R_SMode		USINT	1
R_BFA	%MB600	USINT	
R_LTH		UINT	100
R_Done		BOOL	
R_RECD		BOOL	
R_RING		BOOL	
R_RLTH		BOOL	
R_Bsy		BOOL	
R_Act		BOOL	
R_Abt		BOOL	
R_Err		BOOL	
R_ErrID		WORD	
ETH_Socket_Close0		ETH_Socket_Close	
C_Ex		BOOL	
C_Done		BOOL	
C_Bsy		BOOL	
C_Err		BOOL	
C_ErrID		WORD	
ETH_Socket_Status0		ETH_Socket_Status	
STA_En		BOOL	TRUE
STA_Valid		BOOL	
STA_CON		BOOL	
STA_REC		BOOL	

变量名	地址	数据类型	初始值
STA_CLO		BOOL	
STA_Send		BOOL	
STA_OING		BOOL	
STA_RING		BOOL	
STA_CING		BOOL	
STA_SING		BOOL	
STA_Err		BOOL	
STA_ErrID		WORD	

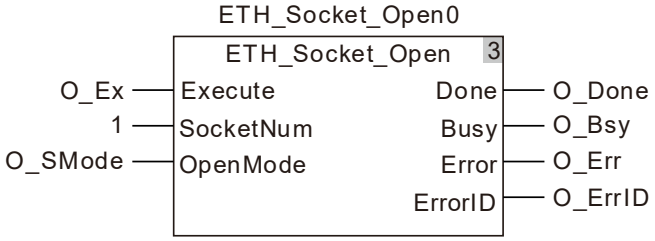
ETH_Socket_Manage0



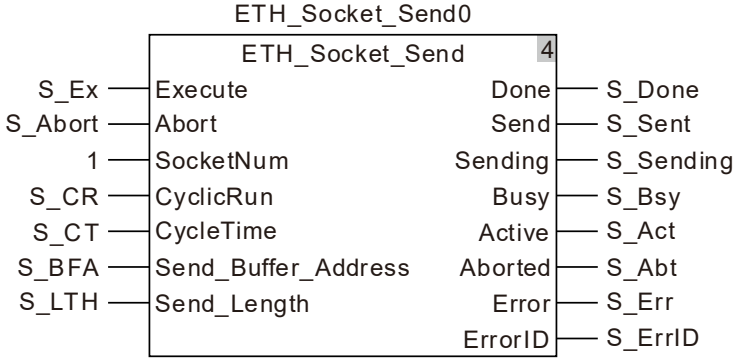
ETH_Socket_Config0

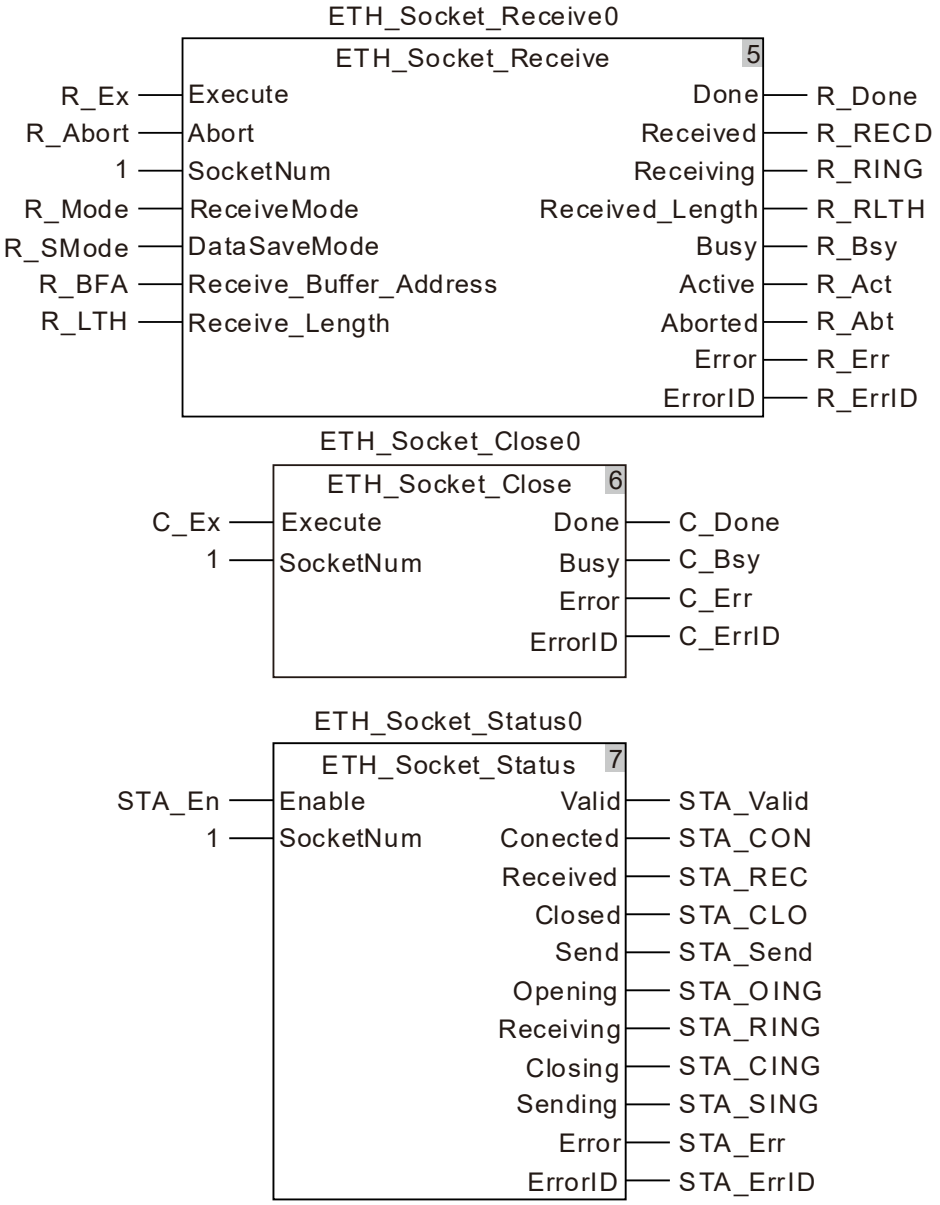


ETH_Socket_Open0



ETH_Socket_Send0





■ 操作步骤及数据交换说明

1. 将变量 S_BFA 和 R_BFA 分别绑定地址%MB200 和%MB600，编译成功并下载后在线。在线后，将标准 MODBUS TCP 数据 “00 00 00 00 00 0b 00 10 10 00 00 02 04 aa bb cc dd” 写入装置%MB200 开始的连续 17 个%MB 装置中。
2. 先执行指令 ETH_Socket_Manage，将 M_En 设为 TRUE，然后在将 M_SEN 设为 TRUE，待变量 M_SR 变为 TRUE 后，再执行指令 ETH_Socket_Config，将变量 CON_Ex 设为 TRUE。
3. 指令 ETH_Socket_Config 指令完成后，执行 ETH_Socket_Open 指令，将 O_Ex 设为 TRUE，主机开始与目标主机建立连接，查看指令 ETH_Socket_Status 的输出位可以看到当前连接状态。
4. 当主机与目标连接成功后，指令 ETH_Socket_Status 的输出位 Connected 变为 TRUE，然后执行 ETH_Socket_Send 指令将变量 S_Ex 设为 TRUE，主机就会以周期发送的方式将装置%MB200 开始的 17 个 Byte 的数据发送给目标主机。

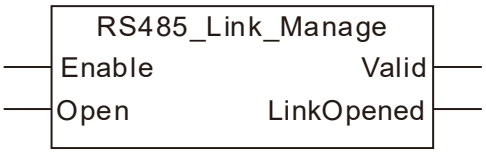
5. 执行指令 ETH_Socket_Receive 可以接收目标主机返回或发送给本主机的数据，本例中，主机会按照持续、覆盖的方式接收和存储数据。
6. 指令 ETH_Socket_Close 用于关闭当前连接，将 C_Ex 设为 TRUE 后，主机会断开连接，并停止发送与接收数据。

8.14.3 RS485 通讯指令

8.14.3.1 RS485_Link_Manage（RS485 通讯管理）

FB/FC	说明	适用机种
FB	此指令用于开启，关闭 RS485 通讯。	DVP15MC11T DVP15MC11T-06

RS485_Link_Manage_instance



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当 Enable 由 FALSE 变为 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
Open （启动位）	开启 RS485 通讯	BOOL	TRUE 或 FALSE （FALSE）	Enable 为 TRUE 时

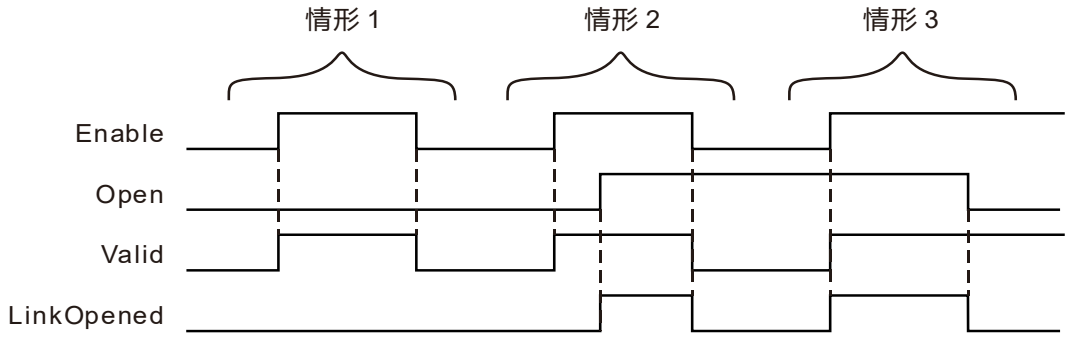
- **输出参数**

名称	功能	数据类型	输出范围
Valid （输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
LinkOpened （Link 生效）	该输出参数为 TRUE 时，说明 RS485 通讯开启	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
LinkOpened	◆ 当 Enable 为 TRUE 时，Open 变为 TRUE	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Open 由 TRUE 变为 FALSE

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE， Open 为 FALSE 时， Valid 变为 TRUE， LinkOpened 为 FALSE。

情形2： 当 Enable 由 FALSE 变为 TRUE，Valid 变为 TRUE，在这种情况下，当 Open 由 FALSE 变为 TRUE 时，LinkOpened 变为 TRUE，当 Enable 由 TRUE 变为 FALSE 时，Valid 和 LinkOpened 变为 FALSE。

情形3： 当 Open 为 TRUE 时， Enable 由 FALSE 变为 TRUE 时，Valid 和 LinkOpened 变为 TRUE，当 Open 变为 FALSE 时，LinkOpened 变为 FALSE。

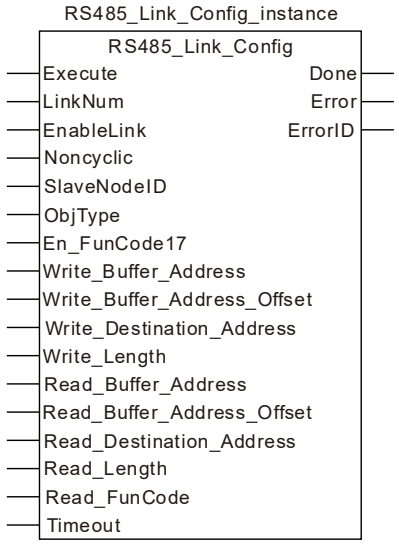
- **功能说明：**

此指令用于控制 RS485 通讯的开关。V1.01 及以上固件版本支持该功能。

1. Enable 用于控制该指令是否生效，Enable 为 FALSE 时，操作该指令其它参数无效。Enable 为 TRUE 时，表示该指令生效，操作其它参数才会有效。
2. 在 Enable 为 TRUE 的情况下，Open 为 TRUE 时 打开 RS485 通讯 Open 为 FALSE 时关闭 RS485 通讯。

8.14.3.2 RS485_Link_Config（RS485 通讯参数配置）

FB/FC	说明	适用机种
FB	此指令用于配置 DVP-15MC 系列运动控制器 RS485 通讯参数。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Execute （执行位）	Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE (FALSE)	
LinkNum （编号）	设定该 Link 的编号	UINT	1~24 （不可缺省）	Execute 由 FALSE 变为 TRUE 时
EnableLink （通讯开启或关闭）	设定该 Link 是否生效	BOOL	TRUE 或 FALSE (FALSE)	Execute 由 FALSE 变为 TRUE 时
Noncyclic （通讯模式）	设定该 Link 是单周期还是多周期工作	BOOL	TRUE 或 FALSE (FALSE)	Execute 由 FALSE 变为 TRUE 时
SlaveNodeID （从站站号）	设定从站 RS485 站号	USINT	0~255 (0)	Execute 由 FALSE 变为 TRUE 时
ObjType （欲读写装置类型）	设定读写从站装置的类型 0：Word 类型 1：Bit 类型	USINT	0~1 （0）	Execute 由 FALSE 变为 TRUE 时
En_FunCode17 （设定是否使用 17 功能码）	设定是否用 17 功能码	BOOL	TRUE 或 FALSE (FALSE)	Execute 由 FALSE 变为 TRUE 时
Write_Buffer_Address	设定存放主站发送数	UINT	%MW0 ~ %MW32767 %QW0 ~ %QW63	Execute 由 FALSE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
（发送数据存放装置）	据的起始装置			变为 TRUE 时
Write_Buffer_Address_Offset （发送数据存放装置偏移量）	设定主站发送数据起始装置的偏移量：若写从站的 Word 地址时，该引脚填 1 表示偏移 1 个 Word ;若写从站的 Bit 地址时，该引脚填 1 表示偏移 1 个 Bit	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
Write_Destination_Address （写入从站 MODBUS 地址）	设定接收主站数据的从站 MODBUS 起始地址	UINT	标准的 Modbus 地址	Execute 由 FALSE 变为 TRUE 时
Write_Length （写入长度）	写入的数据长度	UINIT	字装置 0~100 位装置 0~256	Execute 由 FALSE 变为 TRUE 时
Read_Buffer_Address （接收数据存放地址）	设定存放主站接收数据的起始装置	UINT	%MW0~%MW32767 %QW0~%QW63	Execute 由 FALSE 变为 TRUE 时
Read_Buffer_Address_Offset （接收数据存放地址偏移量）	设定主站接收数据起始装置的偏移量：若读从站的 Word 地址时，该引脚填 1 表示偏移 1 个 Word ;若读从站的 Bit 地址时，该引脚填 1 表示偏移 1 个 Bit	USINT	0~255 (0)	Execute 由 FALSE 变为 TRUE 时
Read_Destination_Address （读取从站 MODBUS 地址）	设定主站准备读取从站的 MODBUS 起始地址	UINT	标准的 Modbus 地址	Execute 由 FALSE 变为 TRUE 时
Read_Length （读取长度）	读取的数据长度	UINT	字装置 0~100 位装置 0~256	Execute 由 FALSE 变为 TRUE 时
Read_FunCode （读 bit 装置功能码）	设定读 Bit 地址时使用的功能码	USINT	1：选择功能码 01 2：选择功能码 02 (2)	Execute 由 FALSE 变为 TRUE 时
Timeout （超时时间）	设定接收超时时间	UINT	大于 0 的整数 （300）	Execute 由 FALSE 变为 TRUE 时

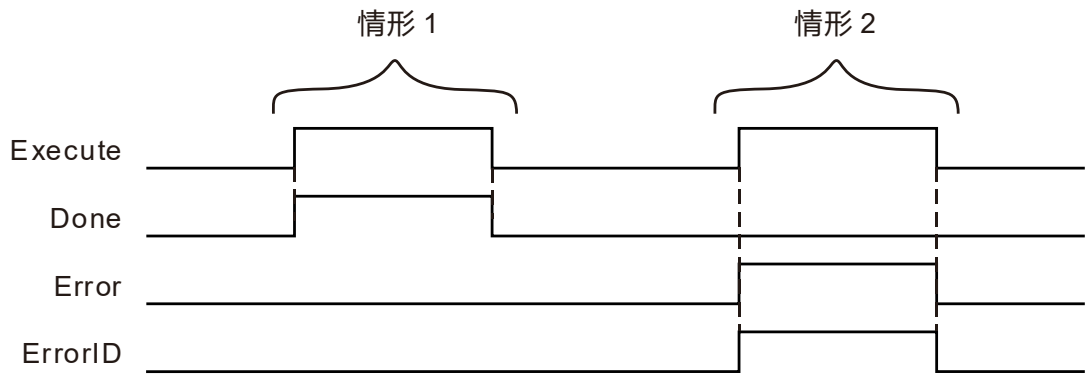
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示该 Link 参数配置成功	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
Error ID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节	WORD	

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当参数配置正确， Execute 由 FALSE 变为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当参数配置错误，Execute 由 FALSE 变为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当参数修改正确后，Execute 重新触发时

- **输出参数变化时序图**



情形1： 在参数配置正确的情况下，当 Execute 由 FALSE 变为 TRUE 时，Done 由 FALSE 变为 TRUE.。

情形2： 当参数配置错误时，当 Execute 由 FALSE 变为 TRUE 时，Done 为 FALSE，Error 由 FALSE 变为 TRUE.。

- **功能说明**

该指令用于对 RS485 通讯 Link 进行参数配置。V1.01 及以上固件版本支持该功能。

- **注意事项:**

1. 输入参数 ObjType 代表读写参数的数据类型。当 ObjType 为 0 时表示读、写从站的 Word 装置，此时 Write_Length、Read_Length 的范围为 0~100，Write_Length 和 Read_Length 不能同时为 0；当 ObjType 为 1 时表示读、写从站的 Bit 装置，此时 Write_Length、Read_Length 的范围为 0~256，Write_Length 和 Read_Length 不能同时为 0。本地地址可以直接填地址，变量（绑定地址），数组（绑定首地址）。
2. 本地地址填写%MB 地址时，只能填写%MW 的低字节，不能填写高字节。
3. 从站地址可以直接填 Modbus 地址、变量。

4. 字操作中的偏移按照字计算，位操作中的偏移按照位计算。

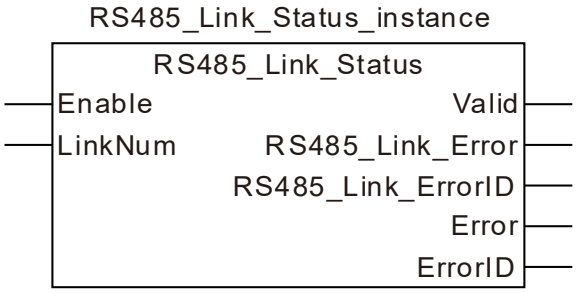
   例如：

   1. 字偏移计算方法：当地址为%MW0，偏移量为 15 时，实际操作地址为%MW15=%MW(0+15) 。
   2. 位偏移计算方法：当地址为%QW0，偏移量为 7 时，则表示%QX0.7。

5. 当用户选择读、写从站 WORD 类型装置时，用户可以选择本机的%MW 装置作为读、写数据的存放装置，范围为%MW0~%MW32767，超出该范围或使用其他装置指令报错.
6. 当用户选择读、写从站 BIT 类型装置时，用户可以选择本机的%MW 和%QW 装置作为读、写数据的存放装置，范围为%MW0~%MW32767 和%QW0~%QW63，超出该范围或使用其他装置指令报错。
7. 此指令设定的参数值仅在主机运行过程中有效，当 DVP-15MC 系列运动控制器断电之后，重新上电时，断电之前配置的参数全部失效，用户如果需要使用断电前的配置就必须重新执行 ETH_Link_Config 指令。
8. 当用户读取从站 Bit 装置时，用户必须设置 Read_FunCode 的值为 01 或 02，用户可以根据读 Bit 装 置的类型并参考相应模块手册来选择功能码。

8.14.3.3 RS485_Link_Status（RS485 通讯状态）

FB/FC	说明	适用机种
FB	该指令用于监控编号对应的 RS485 连接是否出错或从站是否回复错误码。	DVP15MC11T DVP15MC11T-06



- **输入参数：**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当 Enable 由 FALSE 变为 TRUE 时，该指令生效。	BOOL	TRUE 或 FALSE（FALSE）	
LinkNum（编号）	设定该指令显示 Link 的编号	UINT	1~24（不可缺省）	Enable 为 TRUE 时。

- **输出参数：**

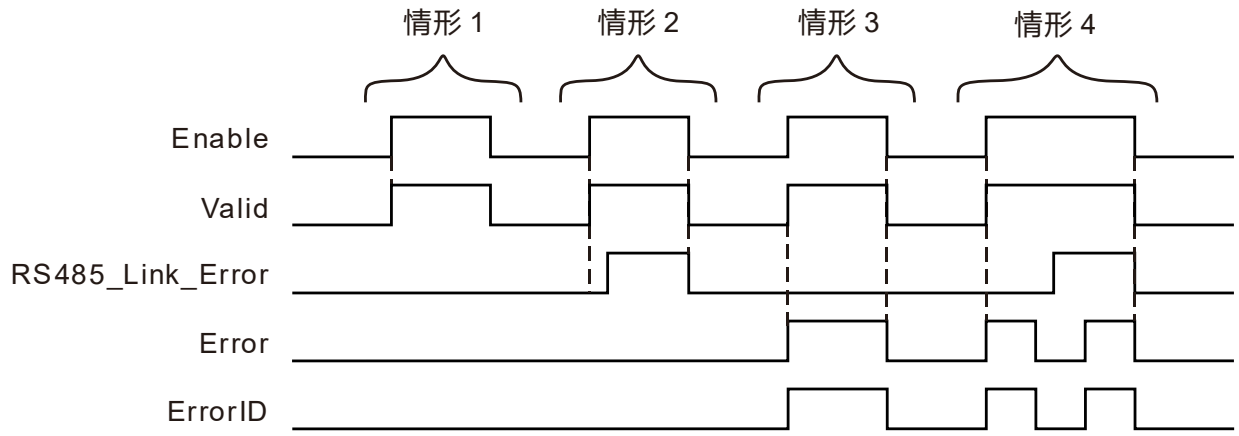
名称	功能	数据类型	输出范围
Valid（输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
RS485_Link_Error（Link 错误）	该指令为 TRUE 表示该指令对应的 Link 通讯有错误发生(在 Error 为 FALSE 的情况下有效)	BOOL	TRUE / FALSE
RS485_Link_ErrorID（Link 错误编号）	输出通讯错误编号。 错误码及其含义如下所示： 1：未识别的功能码 2：从站回复报文中的地址与配置地址不同 3：从站回复报文中的接收数据长度与配置长度不符。 4：接收超时。 5：校验错误。 6：配置的读写长度都为 0。 7：实际接收长度超出最大接收长度。 8：发送超时。 0x80+异常码：异常码指从站回复的异常码。	WORD	
Error（指令错误）	该指令为 TRUE 时表示该指令输入参数有错误	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （指令错误编号）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节	WORD	

- **输出参数刷新时机：**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 由 FALSE 变为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
RS485_Link_Error	◆ 当 Enable 为 TRUE 时，通讯发生错误时。	◆ 当通讯恢复正常（Error 为 FALSE）时 ◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 RS485_Link_Manage 中 Open 由 TRUE 变为 FALSE 时(Error 为 FALSE)
Error	◆ 当 Enable 为 TRUE 时，该指令输入参数有错误时。	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当修改输入参数正确时。

- **输出参数变化时序图：**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE,（通讯与指令输入没有错误）

情形2： 通讯发生错误时， Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE, RS485_Link_Error 在一段时间后变为 TRUE.。

情形3： 当指令输入参数错误时：Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE，Error 变为 TRUE。

情形4： 当通讯发生错误，指令输入同时也有错误时，Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE，Error 变为 TRUE，RS485_Link_Error 仍旧为 FALSE,然后把输入参数修改正确后，Error 变为 FALSE，一段时间后，RS485_Link_Error，变为 TRUE，当输入参数再次错误时，Error 变为 TRUE，此时无论通讯是否正常 RS485_Link_Error 将不会发生改变.在 Enable 为 TRUE，Error 为 TRUE 时，RS485_Link_Error 状态不会刷新，状态无效。

- **功能说明：**

该指令用于显示指令对应 Link 的通讯状态。V1.01 及以上固件版本支持该功能。

- **注意事项：**

  1. 当 Enable 为 TRUE，Error 为 TRUE 时，RS485_Link_Error 状态不会刷新，状态无效。
  2. LinkNum 可以填写数字，变量。

8.14.3.4 RS485 主从通讯范例

程序范例

- 从站为 **DVP32ES2**，站号为 **1**，把主站**%MW100~%MW109** 的 **10** 个字装置的值写入从站 **D0~D9**， 再把从站 **D100~D119** 的 **20** 个字装置的值读到主站的**%MW0~%MW19**，(如下表**"**变量 **1"**所示)。
  - 变量表 **1**

名称	地址	数据类型	初始值
Mng		RS485_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid		BOOL	
Mng_LOpen		BOOL	
Confg		RS485_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	0
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW100	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#1000
Confg _WL		UINT	10
Confg _RBA	%MW0	UINT	
Confg_RBAO		USINT	0
Confg _RDA		UINT	16#44C
Confg _RL		UINT	20
Confg _RFC		USINT	
Confg _Tout		UINT	1000
Confg _Done		BOOL	
Confg _Err		BOOL	
Confg _ErrID		WORD	
LStatus		RS485_Link_Status	
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid		BOOL	
LStatus_RLE		BOOL	

名称	地址	数据类型	初始值
LStatus_RLEID		BOOL	
LStatus_Err		BOOL	
LStatus_ErrID		WORD	

设置完这些参数后，将 Confg _Ex 变量变为 TRUE，然后将 Mng_Open 变为 TRUE 即可。

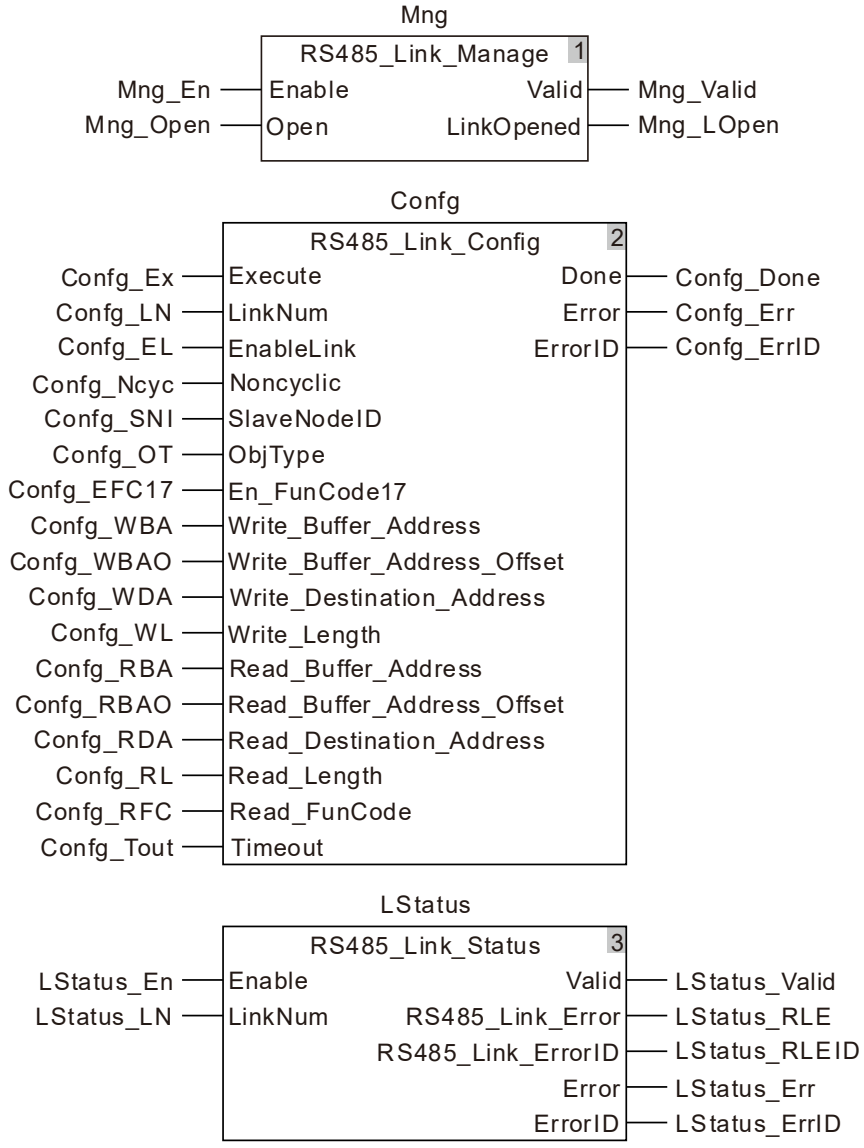
如果在通讯过程中需要更改配置，需要将新配置设置后，将 Confg_Ex 上升沿触发一次，然后，将 Mng_Open 变量变为 FALSE，再变为 TRUE.

- **从站 DVP32ES2,站号为 1，将主站%MX0.0~%MX3.6 的 31 个位装置的值写入从站的 Y0~Y30 装置中，并将从站的 Y0~Y30 装置的值读到本站的%QX0.1~%QX3.7 的 31 个位装置中，(如下表“变量 2”所示)。**
  - **变量表 2**

名称	地址	数据类型	初始值
Mng		RS485_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid			
Mng_LOpen			
Confg		RS485_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	1
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW0	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#0400
Confg _WL		UINT	31
Confg _RBA	%QW0	UINT	
Confg_RBAO		USINT	1
Confg _RDA		UINT	16#0400
Confg _RL		UINT	31
Confg _RFC		USINT	
Confg _Tout		UINT	1000
Confg _Done			
Confg _Err			
Confg _ErrID			
LStatus		RS485_Link_Status	

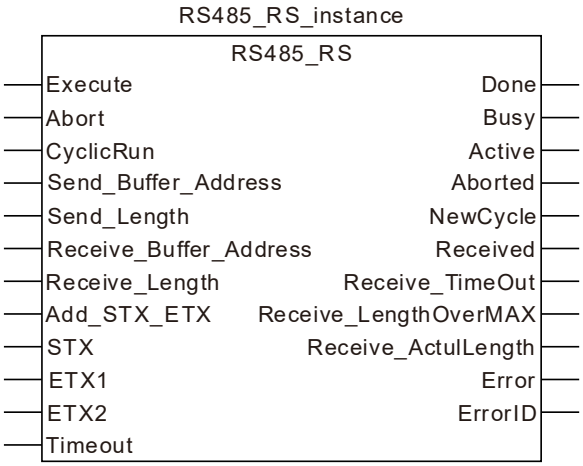
名称	地址	数据类型	初始值
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid			
LStatus_RLE			
LStatus_RLEID			
LStatus_Err			
LStatus_ErrID			

➢ **程序**



8.14.3.5 RS485_RS（RS485 自由协议）

FB/FC	说明	适用机种
FB	此指令用于配置 DVP-15MC 系列运动控制器 RS 485 自由协议通讯参数。	DVP15MC11T DVP15MC11T-06



- **输入参数：**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
Abort（打断）	该输入为 TRUE 时，打断自由协议通讯	BOOL	TRUE 或 FALSE（FALSE）	当 Execute 由 FALSE 变为 TRUE 时
CyclicRun（通讯模式）	CyclicRun 为 TRUE 时，该指令多周期工作	BOOL	TRUE 或 FALSE（FALSE）	当 Execute 由 FALSE 变为 TRUE 时
Send_Buffer_Address（发送数据存放地址）	设定存放本站发送数据的起始装置	USINT	%MB0-%MB32767	当 Execute 由 FALSE 变为 TRUE 时
Send_length（发送数据长度）	设置发送数据长度	UINT	0-200（Byte）(0)	当 Execute 由 FALSE 变为 TRUE 时
Receive_Buffer_Address（接收数据存放地址）	设定存放本站接收数据的起始装置	USINT	%MB0-%MB32767	当 Execute 由 FALSE 变为 TRUE 时
Receive_Length（接收数据长度）	设置接收数据长度	UINT	0-200(Byte)(0)	当 Execute 由 FALSE 变为 TRUE 时
Add_STX_ETX（添加头码尾码）	该输入为 TRUE 时启用头码尾码	BOOL	TRUE 或 FALSE（FALSE）	当 Execute 由 FALSE 变为 TRUE 时
STX（头码）	头码的设定	USINT	0-16#7F(16#3A)	当 Execute 由 FALSE 变为 TRUE 时

名称	功能	数据类型	设定范围 （缺省值）	生效时机
ETX1 （第一个尾码）	第一个尾码的设定	USINT	0-16#7F (16#0D)	当 Execute 由 FALSE 变为 TRUE 时
ETX2 （第二个尾码）	第二个尾码的设定	USINT	0-16#7F (16#0A)	当 Execute 由 FALSE 变为 TRUE 时
Timeout （超时时间）	设置接收超时时间	UINT	0-32767 (0)	当 Execute 由 FALSE 变为 TRUE 时

- **输出参数**

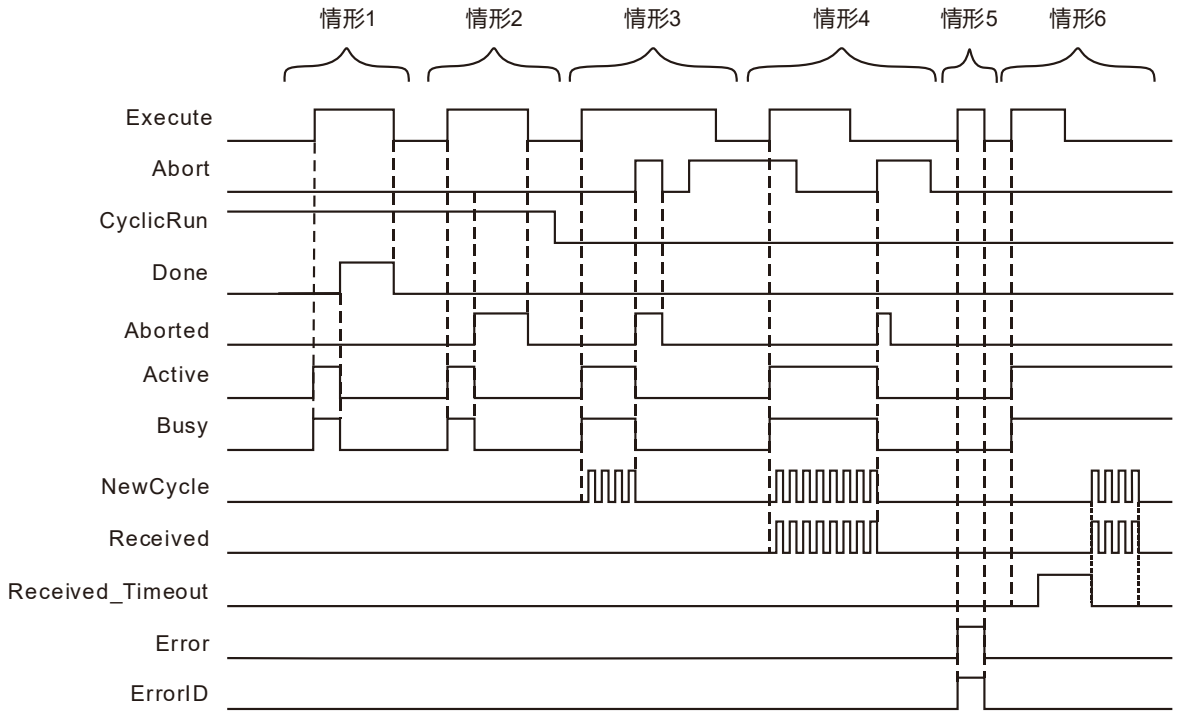
名称	功能	数据类型	输出范围
Done (完成位)	该输出 TRUE 时，表示该指令完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出为 TRUE 时表示正在通讯中	BOOL	TRUE / FALSE
Active （控制中）	该输出为 TRUE 时表示该指令控制端口	BOOL	TRUE / FALSE
Aborted （打断中）	该输出为 TRUE 时表示该指令当前工作被打断	BOOL	TRUE / FALSE
NewCycle （开始下一个周期）	该输出为 TRUE 时，表示该指令完成一次（多周期工作下）	BOOL	TRUE / FALSE
Received （收到回复）	该输出为 TRUE 时表示接收成功	BOOL	TRUE / FALSE
Receive_TimeOut （接收超时）	该指令为 TRUE 表示接收超时	BOOL	TRUE / FALSE
Receive_LengthOverMAX （接收超过最大长度）	该指令为 TRUE 表示接收长度超过最大接收长度	BOOL	TRUE / FALSE
Receive_ActulLength （实际接收长度）	实际接收的数据长度	UINT	0-200
Error （错误）	该指令为 TRUE 表示指令配置有错误	BOOL	TRUE / FALSE
ErrorID （错误编号）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节	WORD	

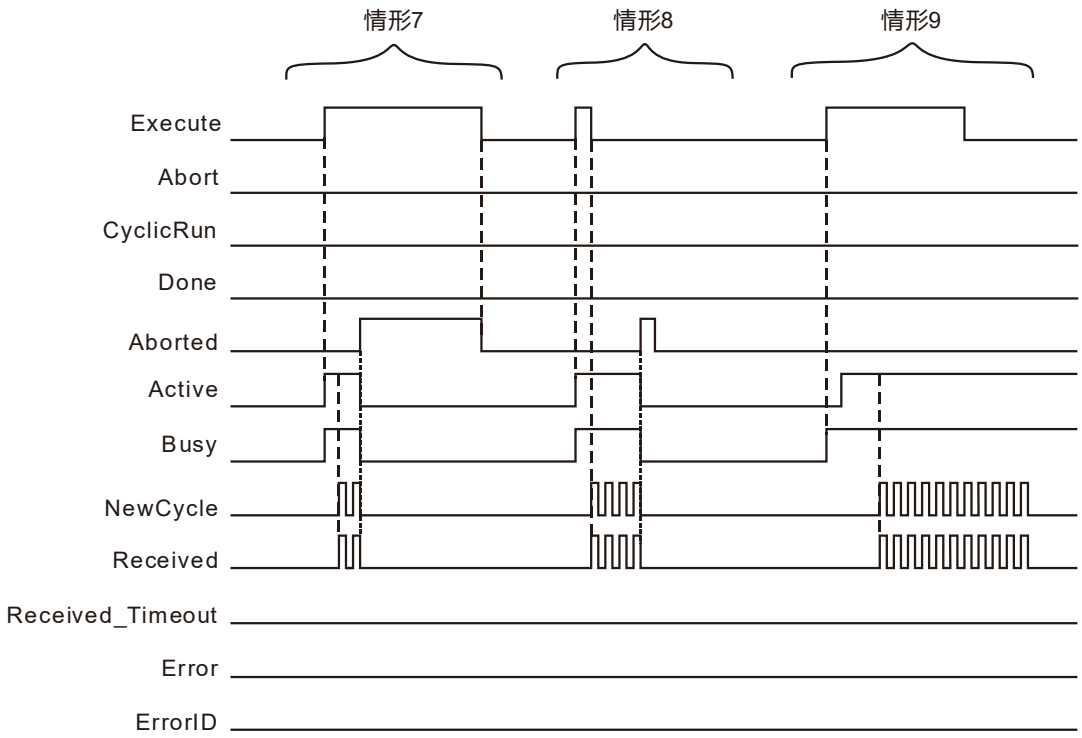
- **输出参数刷新时机：**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE 后（单周期状态下）	◆ 当 Execute 由 TRUE 变为 FALSE 时(单周期状态下）

名称	变为 TRUE 的时机	变为 FALSE 的时机
Busy	◆ 当 Execute 由 FALSE 变为 TRUE 时	◆ 当 Abort 为 TRUE 时（多周期） ◆ 当 Done 变为 TRUE（单周期）时
Active	◆ 当 Execute 由 FALSE 变为 TRUE 后	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当前指令（多周期状态）被另外一个指令打断时 ◆ 当 Done 变为 TRUE（单周期）时
Aborted	◆ 当 ABORT 由 FALSE 变为 TRUE(第一次) 时 ◆ 当前指令（多周期状态）被另外一个指令打断时	◆ 当 ABORT 由 TRUE 变为 FALSE 时 ◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 Execute 为 FALSE，ABORT 由 FALSE 变为 TRUE，Aborted 会变 TRUE 有一个周期后 FALSE 掉时。
NewCycle	◆ 当指令完成一个周期工作时（多周期）	◆ 当指令进入下一个发送接收周期时
Received	◆ 当指令收到回复时	◆ 当收到回复后进入下一个周期时
Receive_TimeOut	◆ 当指令配置了接收，超过接收超时时间没有收到回复时	◆ 当收到回复时
Receive_LengthOverMAX	◆ 当实际接收长度大于最大接收长度时	◆ 当接收数据长度小于最大长度时
Error	◆ 当指令配置错误时	◆ 当指令配置正确时

- **输出参数变化时序图**





情形1： 在单周期工作状态下，当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，当指令动作完成时，Busy 变为 FALSE，Done 变为 TRUE，当 Execute 变为 FALSE 时，Done 变为 FALSE。

情形2： 在单周期工作状态下,当该指令被其他指令打断时，Busy 和 Active 变为 False，Abort 变为 TRUE.Exetuce 变为 False,Aborted 变为 False,如果被打断前，Execute 是 False，Aborted 变为 TRUE 一个周期

情形3： 在多周期只有发送的状态下，当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，一段时间后， NewCycle 变为 TRUE、FALSE 交替闪烁。当 Abort 变为 TRUE 时，Busy 变为 FALSE，Aborted 变为 TRUE，Abort 变为 FALSE，Aborted 依然为 TRUE，当 Abort 再次变为 TRUE 时，Aborted 依然为 TRUE。当 Execute 由 TRUE 变为 FALSE 时，Aborted 变为 FALSE。

情形4： 在多周期发送、 接收都配置的情况下并且 Abort 为 TRUE。当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE， 一段时间后，NewCycle 和 Received 开始 TRUE 和 FALSE 交替闪烁，Execute 由 TRUE 变为 FALSE 后当把 Abort 重新触发一次，Aborted 变为 TRUE 一个周期后变为 FALSE,其余的变为 FALSE.

情形5： 当指令的配置参数有错误时，当 Execute 由 FALSE 变为 TRUE 时，Error 变为 TRUE，当 Execute 变为 FALSE 时，Error 变为 FALSE.

情形6： 存在接收超时的情况时，当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE 一段时间后，Receive_Timeout 变为 TRUE，当通讯恢复时，NewCycle 和 Received 开始 TRUE 和 FALSE 交替闪烁。

情形7： 在多周期模式下（Execute 为 TRUE），当该指令被其他指令打断时，该指令的 Aborted 会变为 TRUE，Execute 变为 FALSE,输出 Aborted 会变为 FALSE,

情形8： 在多周期模式下，（Execute 为 FALSE），当该指令被其他指令打断时，该指令的 Aborted 会变为 TRUE 一个周期，然后 FALSE 掉。

情形9： 该指令打断其他指令时，Busy 变 TRUE，一段时间后，Active 变为 TRUE。其余输出按照之前的情形对应输出。

- **功能说明**

  该指令用于 RS485 自由协议的通讯参数配置和通讯状态监控。V1.01 及以上固件版本支持该功能。

- **注意事项:**

  1. 该指令不会自动加校验码，在 ASCII 模式时，用户发送的数据需要是转换好的 ASCII.
  2. 发送数据的总长度为 200Byte,指令中填写的长度不包括头码尾码。
  3. RS485_RS 指令和 RS_485_Link_Config 指令同时存在时，
     a. RS485_RS 指令的执行位由 FALSE 变为 TRUE 时，会使 PLC 的 RS485Link 功能生效。
     b. RS485_RS 的 Abort 由 FALSE 变为 TRUE 时，会使正在运行中的 RS485Link 重新生效。
     c. 当 RS485_RS 指令的执行位由 FALSE 变为 TRUE 时，RS485_Link_Manage 中 Enable 位 TRUE 时，指令中的 Open 将可以控制 PLCLinkling 和自由协议通讯功能。
     d. 程序中如果有多个 RS485_RS 存在，只有一个生效，最后一次触发的生效。
     e. RS485_RS 发出去的报文，如果从站会进行回复，RS485_RS 必须设置接收地址。

8.14.3.6 RS485 自由协议范例

程序范例

- **使用头码尾码功能。**

1. 用自由协议在 ASCII 模式下发送一笔标准 Modbus 报文（发送地址%MB4000,接收地址%MB5000），并接收从站回复的数据。

   报文内容为：01 10 15 00 00 01 02 00 08，

   首先计算出校验码为 CF，然后将报文转换为 ASCII 码，

   转换的报文内容为： 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46。

2. 当按照变量表 1 配置时，%MB4000~%MB4019 中的数据为 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 。

总线上的报文数据为 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A。

- **变量 1**

名称	地址	数据类型	初始值
RS		RS485_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#30
RS_SL		UINT	20
RS_RBA	%MB5000	USINT	
RS_RL		UINT	20
RS_AddSE		BOOL	1
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Act		BOOL	
RS_Abt		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

■ **不使用头码尾码功能。**

1. 用自由协议在 ASCII 模式下发送一笔标准 Modbus 报文（发送地址%MB4000,接收地址%MB5000），并接收从站回复的数据。

   报文内容为：01 10 15 00 00 01 02 00 08，

   首先计算出校验码为 CF，然后将报文转换为 ASCII 码，

   转换的报文内容为：3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

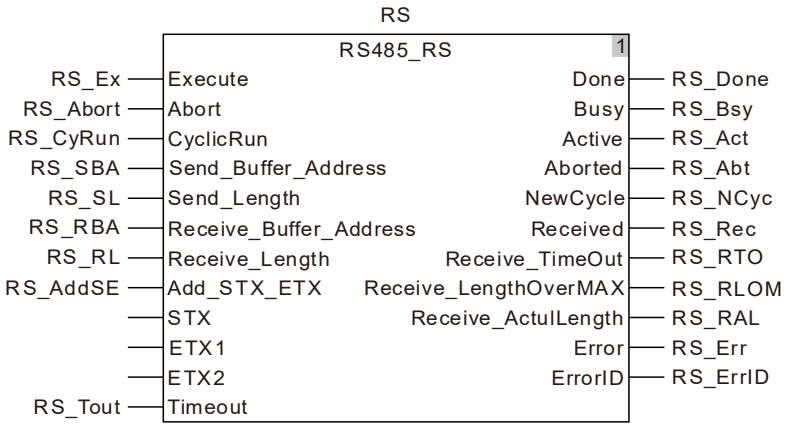
2. 当按照变量表 2 配置时，%MB4000~%MB4022 中的数据为 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

总线上的报文数据为 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

➢ **变量表 2**

名称	地址	数据类型	初始值
RS		RS485_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#3A
RS_SL		UINT	23
RS_RBA	%MB5000	USINT	
RS_RL		UINT	23
RS_AddSE		BOOL	0
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Act		BOOL	
RS_Abt		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

➢ **程序**

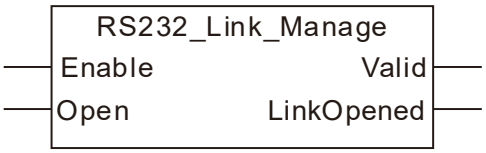

RS
RS485_RS
1
RS_Ex
RS_Abort
RS_CyRun
RS_SBA
RS_SL
RS_RBA
RS_RL
RS_AddSE
RS_Tout
Execute
Abort
CyclicRun
Send_Buffer_Address
Send_Length
Receive_Buffer_Address
Receive_Length
Add_STX_ETX
STX
ETX1
ETX2
Timeout
Done
Busy
Active
Aborted
NewCycle
Received
Receive_TimeOut
Receive_LengthOverMAX
Receive_ActulLength
Error
ErrorID
RS_Done
RS_Bsy
RS_Act
RS_Abt
RS_NCyc
RS_Rec
RS_RTO
RS_RLOM
RS_RAL
RS_Err
RS_ErrID

8.14.4 RS232 通讯指令

8.14.4.1 RS232_Link_Manage(RS232 通讯管理指令)

FB/FC	说明	适用机种
FB	此指令用于开启，关闭 RS232 通讯。	DVP15MC11T DVP15MC11T-06

RS232_Link_Manage_instance



- **输入参数**

名称	功能	数据类型	设定范围 (缺省值)	生效时机
Enable (执行位)	当 Enable 由 FALSE 变为 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE (FALSE)	-
Open (启动位)	开启 RS232 通讯	BOOL	TRUE 或 FALSE (FALSE)	Enable 为 TRUE 时

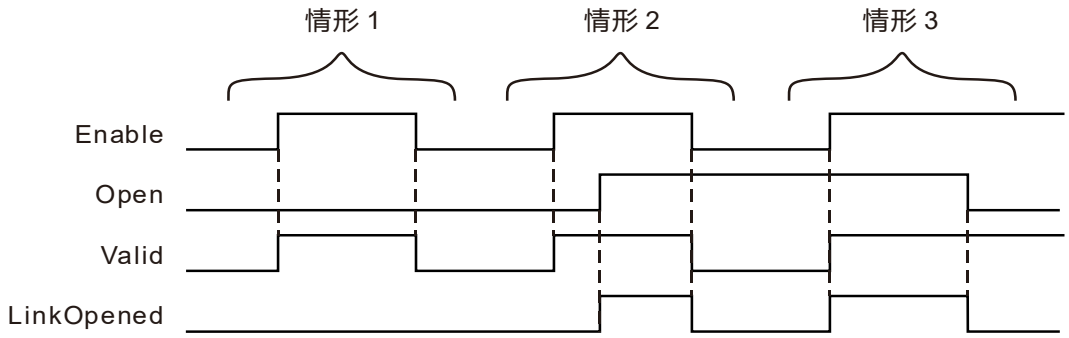
- **输出参数**

名称	功能	数据类型	输出范围
Valid (输出有效)	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
LinkOpened (Link 生效)	该输出参数为 TRUE 时，说明 RS232 通讯开启	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
LinkOpened	◆ 当 Enable 为 TRUE 时，Open 变为 TRUE	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Open 由 TRUE 变为 FALSE

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE， Open 为 FALSE 时， Valid 变为 TRUE， LinkOpened 为 FALSE。

情形2： 当 Enable 由 FALSE 变为 TRUE，Valid 变为 TRUE，在这种情况下，当 Open 由 FALSE 变为 TRUE 时，LinkOpened 变为 TRUE，当 Enable 由 TRUE 变为 FALSE 时，Valid 和 LinkOpened 变为 FALSE。

情形3： 当 Open 为 TRUE 时， Enable 由 FALSE 变为 TRUE 时，Valid 和 LinkOpened 变为 TRUE，当 Open 变为 FALSE 时，LinkOpened 变为 FALSE。

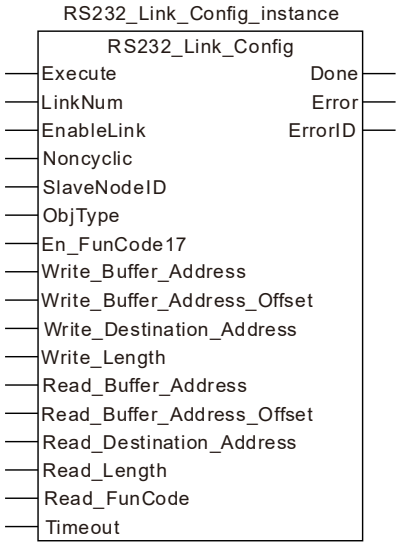
- **功能说明：**

此指令用于控制 RS232 通讯的开关。V1.01 及以上固件版本支持该功能。

1. Enable 用于控制该指令是否生效，Enable 为 FALSE 时，操作该指令其它参数无效。Enable 为 TRUE 时，表示该指令生效，操作其它参数才会有效。
2. 在 Enable 为 TRUE 的情况下，Open 为 TRUE 时 打开 RS232 通讯 Open 为 FALSE 时关闭 RS232 通讯。

8.14.4.2 RS232_Link_Config（RS232 通讯参数配置）

FB/FC	说明	适用机种
FB	此指令用于配置 DVP-15MC 系列运动控制器 RS232 通讯参数。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Execute （执行位）	Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE (FALSE)	
LinkNum （编号）	设定该 Link 的编号	UINT	1~24 （不可缺省）	Execute 由 FALSE 变为 TRUE 时
EnableLink （通讯开启或关闭）	设定该 Link 是否生效	BOOL	TRUE 或 FALSE (FALSE)	Execute 由 FALSE 变为 TRUE 时
Noncyclic （通讯模式）	设定该 Link 是单周期还是多周期工作	BOOL	TRUE 或 FALSE (FALSE)	Execute 由 FALSE 变为 TRUE 时
SlaveNodeID （从站站号）	设定从站 RS232 站号	USINT	0~255 (0)	Execute 由 FALSE 变为 TRUE 时
ObjType （欲读写装置类型）	设定读写从站装置的类型 0：Word 类型 1：Bit 类型	USINT	0~1 （0）	Execute 由 FALSE 变为 TRUE 时
En_FunCode17 （设定是否使用 17 功能码）	设定是否用 17 功能码	BOOL	TRUE 或 FALSE (FALSE)	Execute 由 FALSE 变为 TRUE 时
Write_Buffer_Address （发送数据存放装置）	设定存放主站发送数据的起始装置	UINT	%MW0 ~ %MW32767 %QW0 ~ %QW63	Execute 由 FALSE 变为 TRUE 时

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Write_Buffer_Address_Offset （发送数据存放装置偏移量）	设定主站发送数据起始装置的偏移量：若写从站的 Word 地址时，该引脚填 1 表示偏移 1 个 Word；若写从站的 Bit 地址时，该引脚填 1 表示偏移 1 个 Bit	USINT	0~255 （0）	Execute 由 FALSE 变为 TRUE 时
Write_Destination_Address （写入从站 MODBUS 地址）	设定接收主站数据的从站 MODBUS 起始地址	UINT	标准的 Modbus 地址	Execute 由 FALSE 变为 TRUE 时
Write_Length （写入长度）	写入的数据长度	UINIT	字装置 0~100 位装置 0~256	Execute 由 FALSE 变为 TRUE 时
Read_Buffer_Address （接收数据存放地址）	设定存放主站接收数据的起始装置	UINT	%MW0~%MW32767 %QW0~%QW63	Execute 由 FALSE 变为 TRUE 时
Read_Buffer_Address_Offset （接收数据存放地址偏移量）	设定主站接收数据起始装置的偏移量：若读从站的 Word 地址时，该引脚填 1 表示偏移 1 个 Word；若读从站的 Bit 地址时，该引脚填 1 表示偏移 1 个 Bit	USINT	0~255 (0)	Execute 由 FALSE 变为 TRUE 时
Read_Destination_Address （读取从站 MODBUS 地址）	设定主站准备读取从站的 MODBUS 起始地址	UINT	标准的 Modbus 地址	Execute 由 FALSE 变为 TRUE 时
Read_Length （读取长度）	读取的数据长度	UINT	字装置 0~100 位装置 0~256	Execute 由 FALSE 变为 TRUE 时
Read_FunCode （读 bit 装置功能码）	设定读 Bit 地址时使用的功能码	USINT	1：选择功能码 01 2：选择功能码 02 (2)	Execute 由 FALSE 变为 TRUE 时
Timeout （超时时间）	设定接收超时时间	UINT	大于 0 的整数 （300）	Execute 由 FALSE 变为 TRUE 时

- **输出参数**

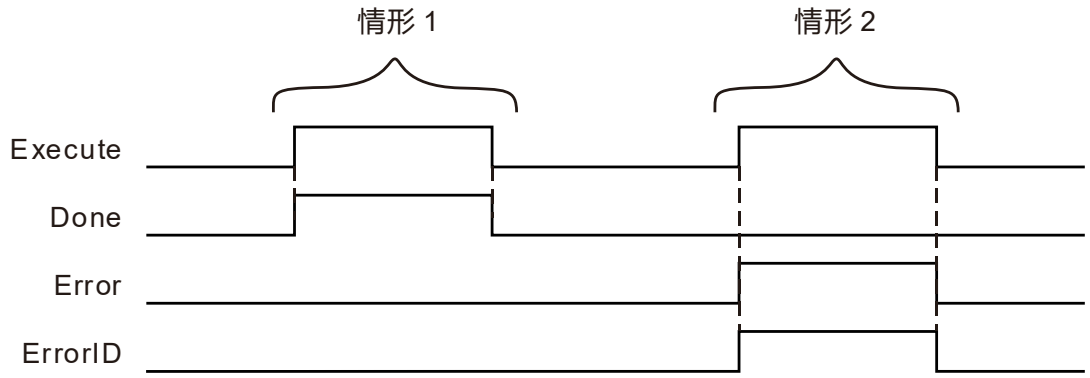
名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示该 Link 参数配置成功	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
Error ID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当参数配置正确， Execute 由 FALSE 变为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当参数配置错误，Execute 由 FALSE 变为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当参数修改正确后，Execute 重新触发时

- **输出参数变化时序图**



情形1： 在参数配置正确的情况下，当 Execute 由 FALSE 变为 TRUE 时，Done 由 FALSE 变为 TRUE.。

情形2： 当参数配置错误时，当 Execute 由 FALSE 变为 TRUE 时，Done 为 FALSE，Error 由 FALSE 变为 TRUE.。

- **功能说明**

该指令用于对 RS232 通讯 Link 进行参数配置。V1.01 及以上固件版本支持该功能。

- **注意事项:**

1. 输入参数 ObjType 代表读写参数的数据类型。当 ObjType 为 0 时表示读、写从站的 Word 装置，此时 Write_Length、Read_Length 的范围为 0~100，Write_Length 和 Read_Length 不能同时为 0；当 ObjType 为 1 时表示读、写从站的 Bit 装置，此时 Write_Length、Read_Length 的范围为 0~256，Write_Length 和 Read_Length 不能同时为 0。本地地址可以直接填地址，变量（绑定地址），数组（绑定首地址）。
2. 本地地址填写%MB 地址时，只能填写%MW 的低字节，不能填写高字节。
3. 从站地址可以直接填 Modbus 地址、变量。
4. 字操作中的偏移按照字计算，位操作中的偏移按照位计算。

例如：

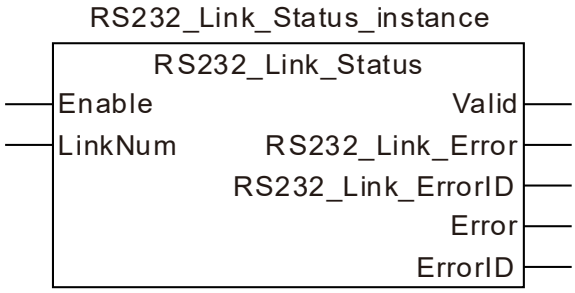
a. 字偏移计算方法：当地址为%MW0，偏移量为 15 时，实际操作地址为%MW15=%MW(0+15) 。

b. 位偏移计算方法：当地址为%QW0，偏移量为 7 时，则表示%QX0.7。

5. 当用户选择读、写从站 WORD 类型装置时，用户可以选择本机的%MW 装置作为读、写数据的存放装置，范围为%MW0~%MW32767，超出该范围或使用其他装置指令报错.
6. 当用户选择读、写从站 BIT 类型装置时，用户可以选择本机的%MW 和%QW 装置作为读、写数据的存放装置，范围为%MW0~%MW32767 和%QW0~%QW63，超出该范围或使用其他装置指令报错。
7. 此指令设定的参数值仅在主机运行过程中有效，当 DVP-15MC 系列运动控制器断电之后，重新上电时，断电之前配置的参数全部失效，用户如果需要使用断电前的配置就必须重新执行 ETH_Link_Config 指令。
8. 当用户读取从站 Bit 装置时，用户必须设置 Read_FunCode 的值为 01 或 02，用户可以根据读 Bit 装 置的类型并参考相应模块手册来选择功能码。

8.14.4.3 RS232_Link_Status（RS232 通讯状态）

FB/FC	说明	适用机种
FB	该指令用于监控编号对应的 RS232 连接是否出错或从站是否回复错误码。	DVP15MC11T DVP15MC11T-06



- **输入参数：**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当 Enable 由 FALSE 变为 TRUE 时，该指令生效。	BOOL	TRUE 或 FALSE（FALSE）	
LinkNum（编号）	设定该指令显示 Link 的编号	UINT	1~24（不可缺省）	Enable 为 TRUE 时。

- **输出参数：**

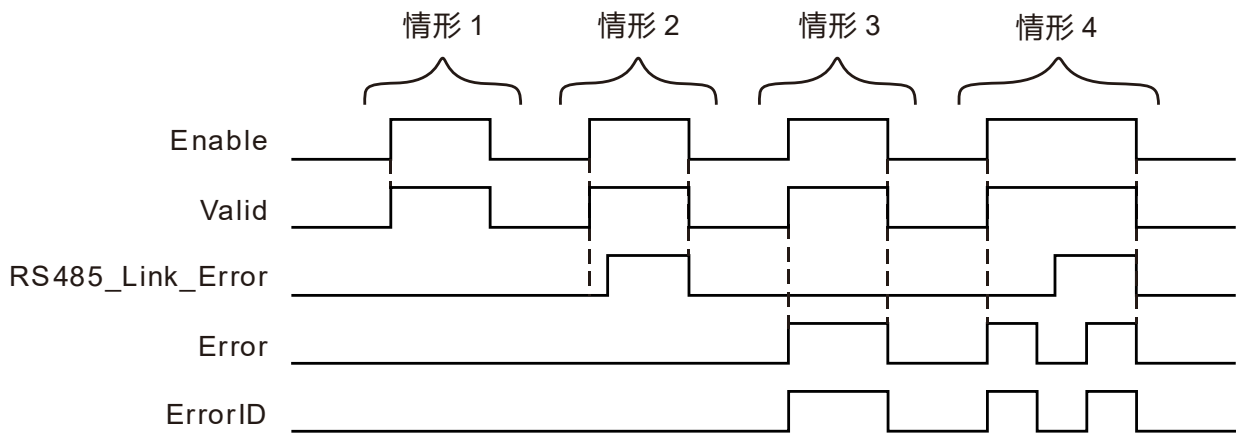
名称	功能	数据类型	输出范围
Valid（输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
RS232_Link_Error（Link 错误）	该指令为 TRUE 表示该指令对应的 Link 通讯有错误发生(在 Error 为 FALSE 的情况下有效)	BOOL	TRUE / FALSE
RS232_Link_ErrorID（Link 错误编号）	输出通讯错误编号。 错误码及其含义如下所示： 1：未识别的功能码 2：从站回复报文中的地址与配置地址不同 3：从站回复报文中的接收数据长度与配置长度不符。 4：接收超时。 5：校验错误。 6：配置的读写长度都为 0。 7：实际接收长度超出最大接收长度。 8：发送超时。 0x80+异常码：异常码指从站回复的异常码。	WORD	
Error（指令错误）	该指令为 TRUE 时表示该指令输入参数有错误	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （指令错误编号）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节	WORD	

- **输出参数刷新时机：**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当 Enable 由 FALSE 变为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时
RS232_Link_Error	◆ 当 Enable 为 TRUE 时，通讯发生错误时。	◆ 当通讯恢复正常（Error 为 FALSE）时 ◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 RS232_Link_Manage 中 Open 由 TRUE 变为 FALSE 时 (Error 为 FALSE)
Error	◆ 当 Enable 为 TRUE 时，该指令输入参数有错误时。	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当修改输入参数正确时。

- **输出参数变化时序图：**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE,（通讯与指令输入没有错误）

情形2： 通讯发生错误时， Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE, RS232_Link_Error 在一段时间后变为 TRUE.。

情形3： 当指令输入参数错误时：Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE，Error 变为 TRUE。

情形4： 当通讯发生错误，指令输入同时也有错误时，Enable 由 FALSE 变为 TRUE 时，Valid 由 FALSE 变为 TRUE，Error 变为 TRUE，RS232_Link_Error 仍旧为 FALSE,然后把输入参数修改正确后，Error 变为 FALSE，一段时间后，RS232_Link_Error，变为 TRUE，当输入参数再次错误时，Error 变为 TRUE，此时无论通讯是否正常 RS232_Link_Error 将不会发生改变.在 Enable 为 TRUE，Error 为 TRUE 时，RS232_Link_Error 状态不会刷新，状态无效。

- **功能说明：**

  该指令用于显示指令对应 Link 的通讯状态。V1.01 及以上固件版本支持该功能。

- **注意事项：**

  1. 当 Enable 为 TRUE，Error 为 TRUE 时，RS232_Link_Error 状态不会刷新，状态无效。
  2. LinkNum 可以填写数字，变量。

8.14.4.4 RS232 主从通讯范例

程序范例

- 从站为 **DVP32ES2**，站号为 **1**，把主站**%MW100~%MW109** 的 **10** 个字装置的值写入从站 **D0~D9**， 再把从站 **D100~D119** 的 **20** 个字装置的值读到主站的**%MW0~%MW19**，(如下表**"**变量 **1"**所示)。
  - 变量表 **1**

名称	地址	数据类型	初始值
Mng		RS232_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid		BOOL	
Mng_LOpen		BOOL	
Confg		RS232_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	0
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW100	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#1000
Confg _WL		UINT	10
Confg _RBA	%MW0	UINT	
Confg_RBAO		USINT	0
Confg _RDA		UINT	16#44C
Confg _RL		UINT	20
Confg _RFC		USINT	
Confg _Tout		UINT	1000
Confg _Done		BOOL	
Confg _Err		BOOL	
Confg _ErrID		WORD	
LStatus		RS232_Link_Status	
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid		BOOL	
LStatus_RLE		BOOL	

名称	地址	数据类型	初始值
LStatus_RLEID		BOOL	
LStatus_Err		BOOL	
LStatus_ErrID		WORD	

设置完这些参数后，将 Confg _Ex 变量变为 TRUE，然后将 Mng_Open 变为 TRUE 即可。

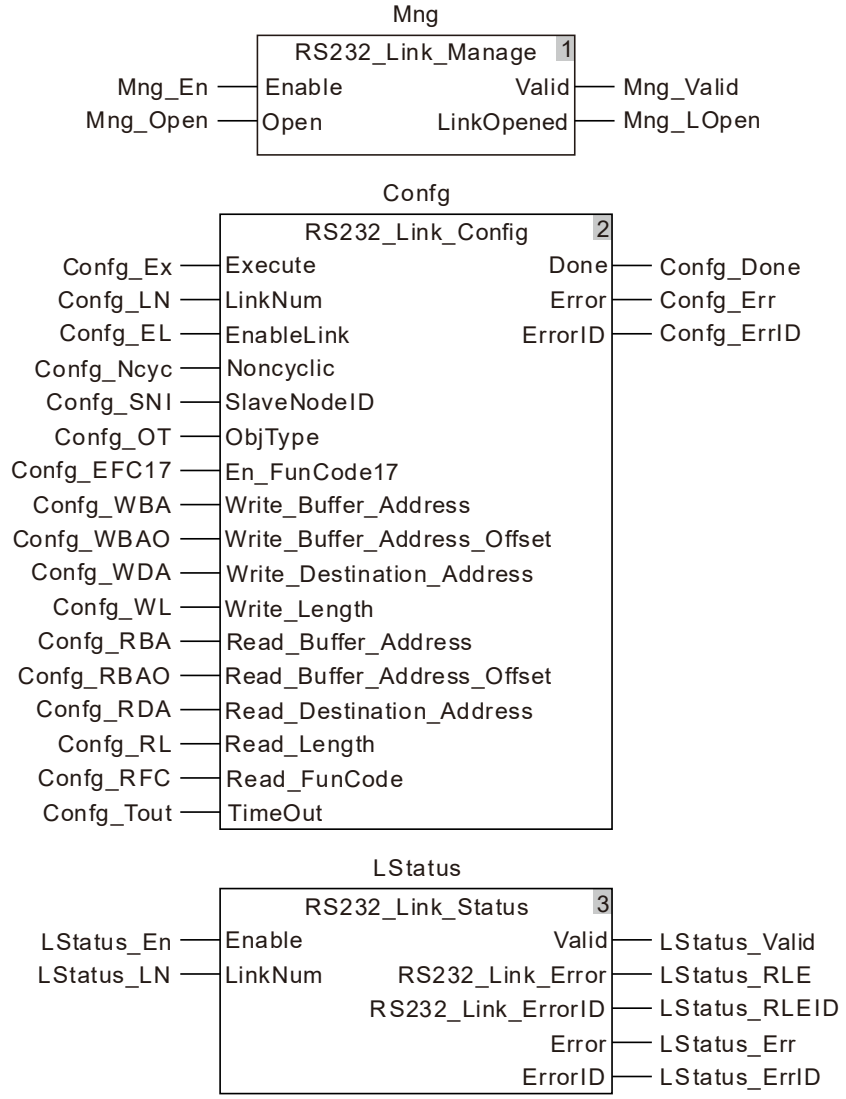
如果在通讯过程中需要更改配置，需要将新配置设置后，将 Confg_Ex 上升沿触发一次，然后，将 Mng_Open 变量变为 FALSE，再变为 TRUE.

- **从站 DVP32ES2,站号为 1，将主站%MX0.0~%MX3.6 的 31 个位装置的值写入从站的 Y0~Y30 装置中，并将从站的 Y0~Y30 装置的值读到本站的%QX0.1~%QX3.7 的 31 个位装置中，(如下表“变量 2”所示)。**
  - **变量表 2**

名称	地址	数据类型	初始值
Mng		RS232_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid			
Mng_LOpen			
Confg		RS232_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	1
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW0	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#0400
Confg _WL		UINT	31
Confg _RBA	%QW0	UINT	
Confg_RBAO		USINT	1
Confg _RDA		UINT	16#0400
Confg _RL		UINT	31
Confg _RFC		USINT	
Confg _Tout		UINT	1000
Confg _Done			
Confg _Err			
Confg _ErrID			
LStatus		RS232_Link_Status	

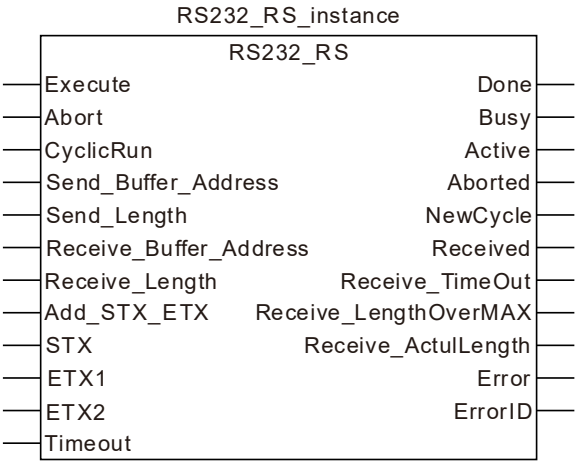
名称	地址	数据类型	初始值
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid			
LStatus_RLE			
LStatus_RLEID			
LStatus_Err			
LStatus_ErrID			

➢ **程序**



8.14.4.5 RS232_RS（RS232 自由协议）

FB/FC	说明	适用机种
FB	此指令用于配置 DVP-15MC 系列运动控制器 RS RS232 自由协议通讯参数。	DVP15MC11T DVP15MC11T-06



- **输入参数：**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，执行该指令	BOOL	TRUE 或 FALSE（FALSE）	
Abort（打断）	该输入为 TRUE 时，打断自由协议通讯	BOOL	TRUE 或 FALSE（FALSE）	当 Execute 由 FALSE 变为 TRUE 时
CyclicRun（通讯模式）	CyclicRun 为 TRUE 时，该指令多周期工作	BOOL	TRUE 或 FALSE（FALSE）	当 Execute 由 FALSE 变为 TRUE 时
Send_Buffer_Address（发送数据存放地址）	设定存放本站发送数据的起始装置	USINT	%MB0-%MB32767	当 Execute 由 FALSE 变为 TRUE 时
Send_length（发送数据长度）	设置发送数据长度	UINT	0-200（Byte）(0)	当 Execute 由 FALSE 变为 TRUE 时
Receive_Buffer_Address（接收数据存放地址）	设定存放本站接收数据的起始装置	USINT	%MB0-%MB32767	当 Execute 由 FALSE 变为 TRUE 时
Receive_Length（接收数据长度）	设置接收数据长度	UINT	0-200(Byte)(0)	当 Execute 由 FALSE 变为 TRUE 时
Add_STX_ETX（添加头码尾码）	该输入为 TRUE 时启用头码尾码	BOOL	TRUE 或 FALSE（FALSE）	当 Execute 由 FALSE 变为 TRUE 时
STX（头码）	头码的设定	USINT	0-16#7F (16#3A)	当 Execute 由 FALSE 变为 TRUE 时
ETX1	第一个尾码的设定	USINT	0-16#7F	当 Execute 由 FALSE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
（第一个尾码）			(16#0D)	变为 TRUE 时
ETX2 （第二个尾码）	第二个尾码的设定	USINT	0-16#7F (16#0A)	当 Execute 由 FALSE 变为 TRUE 时
Timeout （超时时间）	设置接收超时时间	UINT	0-32767 (0)	当 Execute 由 FALSE 变为 TRUE 时

- **输出参数**

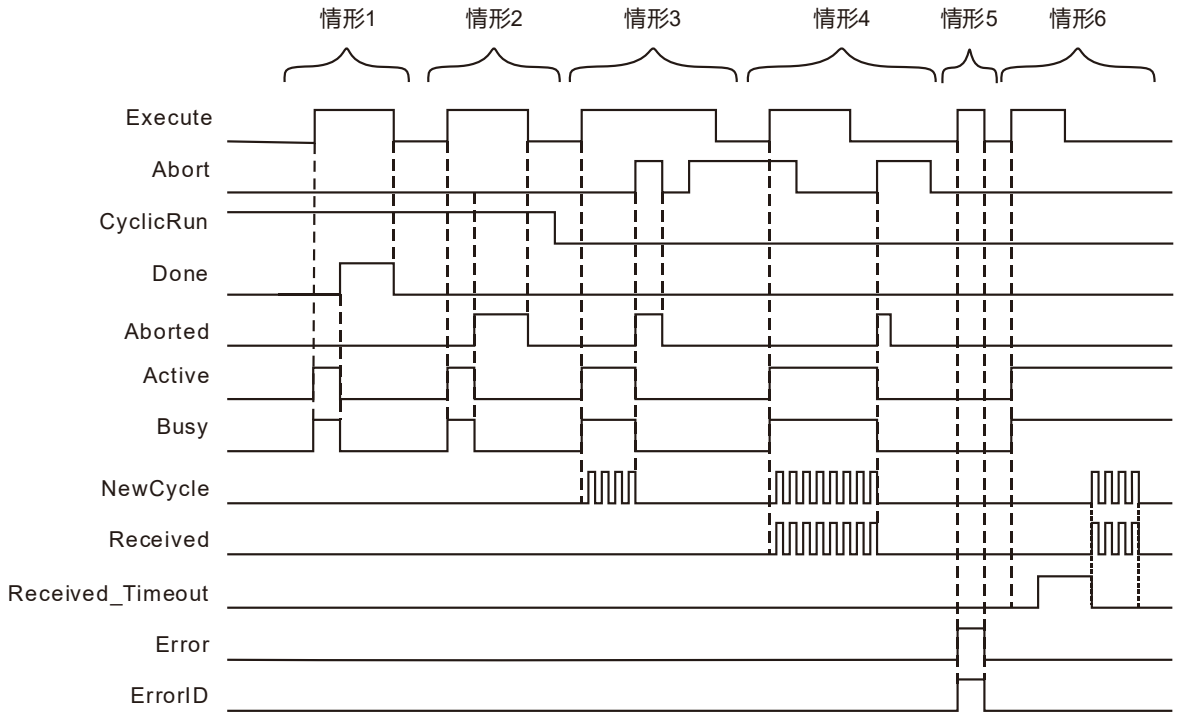
名称	功能	数据类型	输出范围
Done (完成位)	该输出 TRUE 时，表示该指令完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出为 TRUE 时表示正在通讯中	BOOL	TRUE / FALSE
Active （控制中）	该输出为 TRUE 时表示该指令控制端口	BOOL	TRUE / FALSE
Aborted （打断中）	该输出为 TRUE 时表示该指令当前工作被打断	BOOL	TRUE / FALSE
NewCycle （开始下一个周期）	该输出为 TRUE 时，表示该指令完成一次（多周期工作下）	BOOL	TRUE / FALSE
Received （收到回复）	该输出为 TRUE 时表示接收成功	BOOL	TRUE / FALSE
Receive_TimeOut （接收超时）	该指令为 TRUE 表示接收超时	BOOL	TRUE / FALSE
Receive_LengthOverMAX （接收超过最大长度）	该指令为 TRUE 表示接收长度超过最大接收长度	BOOL	TRUE / FALSE
Receive_ActulLength （实际接收长度）	实际接收的数据长度	UINT	0-200
Error （错误）	该指令为 TRUE 表示指令配置有错误	BOOL	TRUE / FALSE
ErrorID （错误编号）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节	WORD	

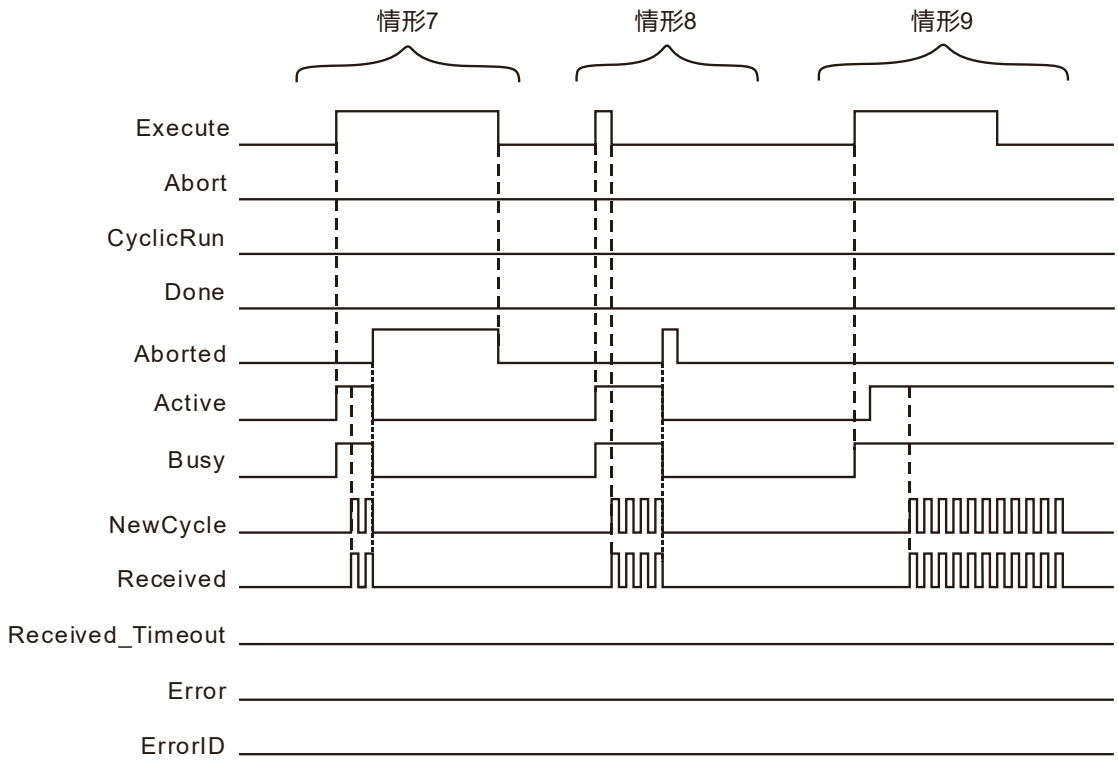
- **输出参数刷新时机：**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE 后（单周期状态下）	◆ 当 Execute 由 TRUE 变为 FALSE 时(单周期状态下）

名称	变为 TRUE 的时机	变为 FALSE 的时机
Busy	◆ 当 Execute 由 FALSE 变为 TRUE 时	◆ 当 Abort 为 TRUE 时（多周期） ◆ 当 Done 变为 TRUE（单周期）时
Active	◆ 当 Execute 由 FALSE 变为 TRUE 后	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当前指令（多周期状态）被另外一个指令打断时 ◆ 当 Done 变为 TRUE（单周期）时
Aborted	◆ 当 ABORT 由 FALSE 变为 TRUE(第一次) 时 ◆ 当前指令（多周期状态）被另外一个指令打断时	◆ 当 ABORT 由 TRUE 变为 FALSE 时 ◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 Execute 为 FALSE，ABORT 由 FALSE 变为 TRUE，Aborted 会变 TRUE 有一个周期后 FALSE 掉时。
NewCycle	◆ 当指令完成一个周期工作时（多周期）	◆ 当指令进入下一个发送接收周期时
Received	◆ 当指令收到回复时	◆ 当收到回复后进入下一个周期时
Receive_TimeOut	◆ 当指令配置了接收，超过接收超时时间没有收到回复时	◆ 当收到回复时
Receive_LengthOverMAX	◆ 当实际接收长度大于最大接收长度时	◆ 当接收数据长度小于最大长度时
Error	◆ 当指令配置错误时	◆ 当指令配置正确时

- **输出参数变化时序图**





情形1： 在单周期工作状态下，当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，当指令动作完成时，Busy 变为 FALSE，Done 变为 TRUE，当 Execute 变为 FALSE 时，Done 变为 FALSE。

情形2： 在单周期工作状态下,当该指令被其他指令打断时，Busy 和 Active 变为 False，Abort 变为 TRUE.Exetuce 变为 False,Aborted 变为 False,如果被打断前，Execute 是 False，Aborted 变为 TRUE 一个周期

情形3： 在多周期只有发送的状态下，当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，一段时间后， NewCycle 变为 TRUE、FALSE 交替闪烁。当 Abort 变为 TRUE 时，Busy 变为 FALSE，Aborted 变为 TRUE，Abort 变为 FALSE，Aborted 依然为 TRUE，当 Abort 再次变为 TRUE 时，Aborted 依然为 TRUE。当 Execute 由 TRUE 变为 FALSE 时，Aborted 变为 FALSE。

情形4： 在多周期发送、 接收都配置的情况下并且 Abort 为 TRUE。当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE， 一段时间后，NewCycle 和 Received 开始 TRUE 和 FALSE 交替闪烁，Execute 由 TRUE 变为 FALSE 后当把 Abort 重新触发一次，Aborted 变为 TRUE 一个周期后变为 FALSE,其余的变为 FALSE.

情形5： 当指令的配置参数有错误时，当 Execute 由 FALSE 变为 TRUE 时，Error 变为 TRUE，当 Execute 变为 FALSE 时，Error 变为 FALSE.

情形6： 存在接收超时的情况时，当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE 一段时间后，Receive_Timeout 变为 TRUE，当通讯恢复时，NewCycle 和 Received 开始 TRUE 和 FALSE 交替闪烁。

情形7： 在多周期模式下（Execute 为 TRUE），当该指令被其他指令打断时，该指令的 Aborted 会变为 TRUE，Execute 变为 FALSE,输出 Aborted 会变为 FALSE,

情形8： 在多周期模式下，（Execute 为 FALSE），当该指令被其他指令打断时，该指令的 Aborted 会变为 TRUE 一个周期，然后 FALSE 掉。

情形9： 该指令打断其他指令时，Busy 变 TRUE，一段时间后，Active 变为 TRUE。其余输出按照之前的情形对应输出。

- **功能说明**

  该指令用于 RS232 自由协议的通讯参数配置和通讯状态监控。V1.01 及以上固件版本支持该功能。

- **注意事项:**

  1. 该指令不会自动加校验码，在 ASCII 模式时，用户发送的数据需要是转换好的 ASCII.
  2. 发送数据的总长度为 200Byte,指令中填写的长度不包括头码尾码。
  3. RS232_RS 指令和 RS_232_Link_Config 指令同时存在时，
     a. RS232_RS 指令的执行位由 FALSE 变为 TRUE 时，会使 PLC 的 RS232Link 功能生效。
     b. RS232_RS 的 Abort 由 FALSE 变为 TRUE 时，会使正在运行中的 RS232Link 重新生效。
     c. 当 RS232_RS 指令的执行位由 FALSE 变为 TRUE 时，RS232_Link_Manage 中 Enable 位 TRUE 时，指令中的 Open 将可以控制 PLCLinkling 和自由协议通讯功能。
     d. 程序中如果有多个 RS232_RS 存在，只有一个生效，最后一次触发的生效。
     e. RS232_RS 发出去的报文，如果从站会进行回复，RS232_RS 必须设置接收地址。

8.14.4.6 RS232 自由协议范例

程序范例

- **使用头码尾码功能。**

1. 用自由协议在 ASCII 模式下发送一笔标准 Modbus 报文（发送地址%MB4000,接收地址%MB5000），并接收从站回复的数据。

   报文内容为：01 10 15 00 00 01 02 00 08，

   首先计算出校验码为 CF，然后将报文转换为 ASCII 码，

   转换的报文内容为： 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46。

2. 当按照变量表 1 配置时，%MB4000~%MB4019 中的数据为 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 。

总线上的报文数据为 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A。

- **变量 1**

名称	地址	数据类型	初始值
RS		RS232_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#30
RS_SL		UINT	20
RS_RBA	%MB5000	USINT	
RS_RL		UINT	20
RS_AddSE		BOOL	1
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Act		BOOL	
RS_Abt		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

■ **不使用头码尾码功能。**

1. 用自由协议在 ASCII 模式下发送一笔标准 Modbus 报文（发送地址%MB4000,接收地址%MB5000），并接收从站回复的数据。

   报文内容为：01 10 15 00 00 01 02 00 08，

   首先计算出校验码为 CF，然后将报文转换为 ASCII 码，

   转换的报文内容为：3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

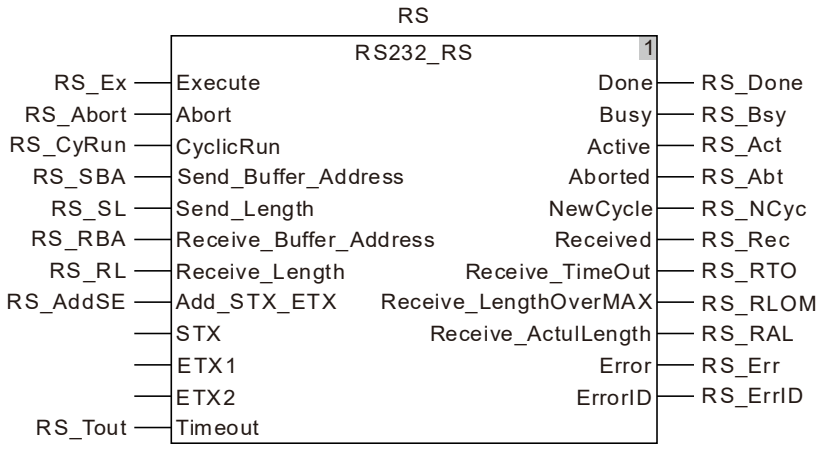
2. 当按照变量表 2 配置时，%MB4000~%MB4022 中的数据为 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

总线上的报文数据为 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

➢ **变量表 2**

名称	地址	数据类型	初始值
RS		RS232_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#3A
RS_SL		UINT	23
RS_RBA	%MB5000	USINT	
RS_RL		UINT	23
RS_AddSE		BOOL	0
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Act		BOOL	
RS_Abt		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

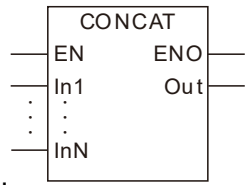
➢ **程序**


RS
RS232_RS
1
RS_Ex
Execute
Done
RS_Done
RS_Abort
Abort
Busy
RS_Bsy
RS_CyRun
CyclicRun
Active
RS_Act
RS_SBA
Send_Buffer_Address
Aborted
RS_Abt
RS_SL
Send_Length
NewCycle
RS_NCyc
RS_RBA
Receive_Buffer_Address
Received
RS_Rec
RS_RL
Receive_Length
Receive_TimeOut
RS_RTO
RS_AddSE
Add_STX_ETX
Receive_LengthOverMAX
RS_RLOM
STX
Receive_ActulLength
RS_RAL
ETX1
Error
RS_Err
ETX2
ErrorID
RS_ErrID
RS_Tout
Timeout

8.15 字符串操作指令

8.15.1 CONCAT（连接字符串）

FB/FC	说明	适用机种
FC	本指令用于将两个(或者多个)字符串类型的变量或常量连接成一个字符串。	DVP15MC11T DVP15MC11T-06



- **参数**

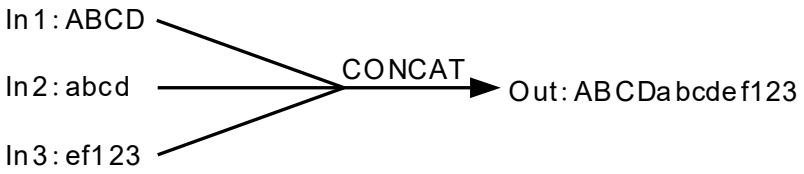
参数名称	含义	输入/输出	描述	参数取值范围
In1 至 InN	用于连接的字符串	输入	程序编写时，可通过编程软件增加或减少运算参数的数量，但其最多为8个，最少为2个，即N = 2 ~ 8	由与输入参数所连变量的数据类型决定
Out	连接后的字符串	输出	连接后的字符串	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 至 InN																				●
Out																				●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将两个 (或者多个) 字符串连接，组成一个新的字符串并输出至*Out*。连接时，*In1*至*InN*的顺序依次连接，如下图所示：



注意

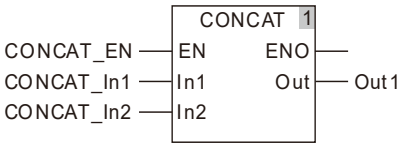
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- 变量CONCAT_In1、CONCAT_In2、Out1的数据类型均为STRING，CONCAT_In1与CONCAT_In2的值分别为'Asasz'和'B1255'，当变量CONCAT_EN为TRUE时，Out1的值为'AsaszB1255'。

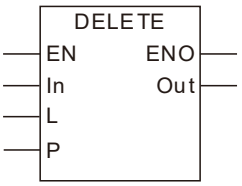
  - **变量和程序**

变量名	数据类型	当前值
CONCAT_EN	BOOL	TRUE
CONCAT_In1	STRING	'Asasz'
CONCAT_In2	STRING	'B1255'
Out1	STRING	'AsaszB1255'



8.15.2 DELETE（删减字符串）

FB/FC	说明	适用机种
FC	本指令用于从字符串类型的变量或常量的指定位置开始删除指定长度的字符串。	DVP15MC11T DVP15MC11T-06



- **参数**

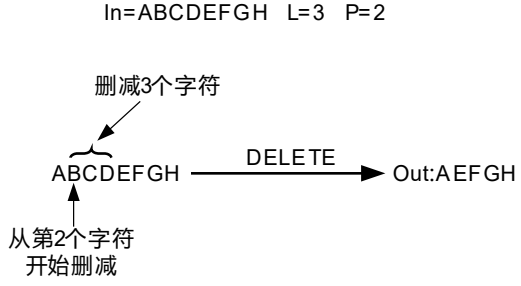
参数名称	含义	输入/输出	描述	参数取值范围
In	用于删减的字符串	输入	用于删减的字符串	由与输入参数所连变量的数据类型决定
L	删减字符的个数	输入	删减字符的个数	0~字符串最大长度
P	删减字符的起始位置	输入	删减字符的起始位置	1~字符串最大长度
Out	删减后的字符串	输出	删减后的字符串	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
L							●													
P							●													
Out																				●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于删除字符串（*In*）中从第*P* 个字符开始的*L*个字符，并将删减后的字符串输出至*Out*。删减方式如下图所示：



注意

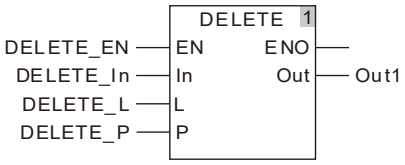
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- DELETE_In为'AaBbCcDd'，DELETE_L= 2，DELETE_P = 3，当DELETE_EN为TRUE时，Out1为'AaCcDd.'

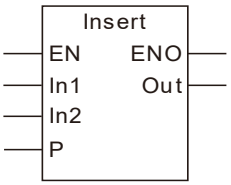
  - **变量和程序**

变量名	数据类型	当前值
DELETE_EN	BOOL	TRUE
DELETE_In	STRING	'AaBbCcDd'
DELETE_L	UINT	2
DELETE_P	UINT	3
Out1	STRING	'AaCcDd'



8.15.3 INSERT（字符串插入）

FB/FC	说明	适用机种
FC	本指令用于向字符串类型的变量或常量中的指定位置插入另一字符串。	DVP15MC11T DVP15MC11T-06



- **参数**

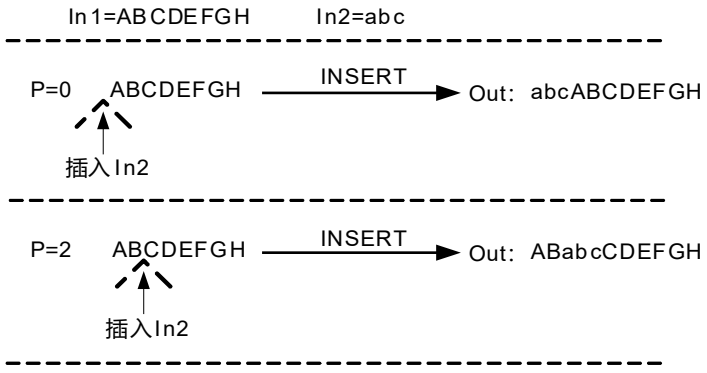
参数名称	含义	输入/输出	描述	参数取值范围
In1	原始字符串	输入	原始字符串	由与输入参数所连变量的数据类型决定
In2	插入的字符串	输入	插入的字符串	由与输入参数所连变量的数据类型决定
P	插入字符串的位置	输入	插入字符串的位置	0~字符串最大长度
Out	插入后的字符串	输出	插入后的字符串	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				●
In2																				●
P							●													
Out																				●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于向字符串（*In1*）中插入另一字符串（*In2*），并将新的字符串输出至*Out*。插入位置为*In1*中**第*P*个与第*P*+1个字符之间**，若P=0，则在In1的最前端插入*In2*。字符串插入方式如下图所示：



注意

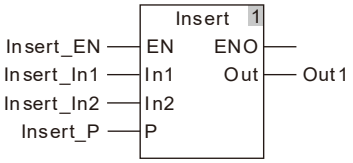
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- Insert_In1为'AaBbCcDd'，Insert_In2为'Ee'，Insert_P=2，当Insert_EN为TRUE时， Out1为'AaEeBbCcDd'。

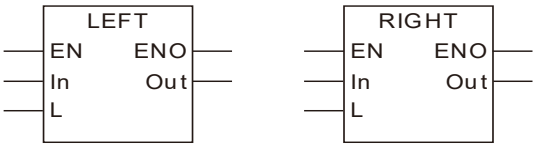
  ➢ **变量和程序**

变量名	数据类型	当前值
Insert_EN	BOOL	FALSE
Insert_In1	STRING	'AaBbCcDd'
Insert_In2	STRING	'Ee'
Insert_P	UINT	2
Out1	STRING	'AaEeBbCcDd'



8.15.4 LEFT / RIGHT（左/右截取字符串）

FB/FC	说明	适用机种
FC	本指令用于从字符串类型的变量或常量中截取指定长度的字符串。	DVP15MC11T DVP15MC11T-06



- **参数**

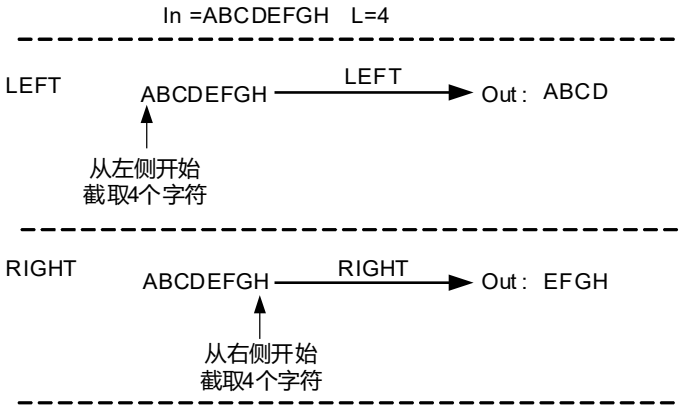
参数名称	含义	输入/输出	描述	参数取值范围
In	原始字符串	输入	用于截取的原始字符串	由与输入参数所连变量的数据类型决定
L	截取字符串的长度	输入	截取字符串的长度	0~字符串最大长度
Out	截取的字符串	输出	截取的字符串	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
L							●													
Out																				●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 指令LEFT和RIGHT均用于从字符串（*In*）中截取指定长度的字符串，并将截取的字符串输出至*Out*。其中LEFT从*In*的左侧开始截取，RIGHT从*In*的右侧开始截取，字符串截取方式如下图所示：



注意

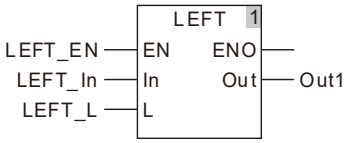
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- LEFT_In为'AaBbCcDd'，LEFT_L=2，当LEFT_EN为TRUE时，Out1为'Aa'（如下表“变量和程序1”所示）；当RIGHT_In为'AaBbCcDd'，RIGHT_L=2，当RIGHT_EN为TRUE时，Out1为'Dd'（如下表“变量和程序2”所示）。

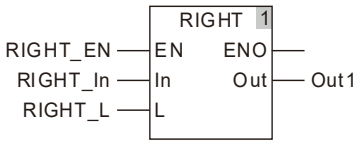
➢ **变量和程序 1**

变量名	数据类型	当前值
LEFT_EN	BOOL	TRUE
LEFT_In	STRING	'AaBbCcDd'
LEFT_L	UINT	2
Out1	STRING	'Aa'



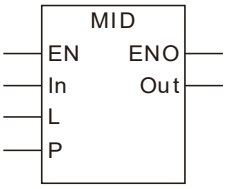
➢ **变量和程序 2**

变量名	数据类型	当前值
RIGHT_EN	BOOL	TRUE
RIGHT_In	STRING	'AaBbCcDd'
RIGHT_L	UINT	2
Out1	STRING	'Dd'



8.15.5 MID（字符串截取）

FB/FC	说明	适用机种
FC	本指令用于从字符串类型的变量或常量的指定位置开始截取指定长度的字符串。	DVP15MC11T DVP15MC11T-06



- **参数**

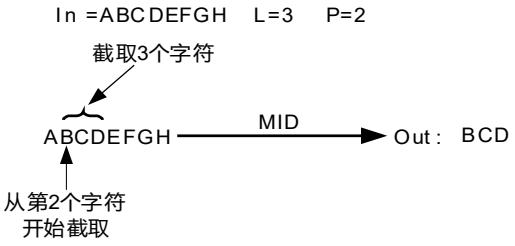
参数名称	含义	输入/输出	描述	参数取值范围
In	用于截取的字符串	输入	用于截取的字符串	由与输入参数所连变量的数据类型决定
L	截取字符的个数	输入	截取字符的个数	0~字符串最大长度
P	截取字符的起始位置	输入	截取字符的起始位置	1~字符串最大长度
Out	截取的字符串	输出	截取的字符串	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
L							●													
P							●													
Out																				●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

  - 本指令用于截取字符串（*In*）中从第*P*个字符开始的*L*个字符，并将截取的字符串输出至*Out*。截取方式如下图所示：



注意

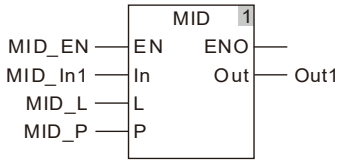
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- MID_In为'AaBbCcDd'，MID_L=2，MID_LP=3，当MID_EN为TRUE时，Out1为'Bb'.

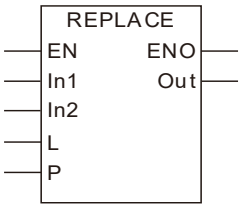
  - **变量和程序**

变量名	数据类型	当前值
MID_EN	BOOL	TRUE
MID_In	STRING	'AaBbCcDd'
MID_L	UINT	2
MID_P	UINT	3
Out1	STRING	'Bb'



8.15.6 REPLACE（字符替换）

FB/FC	说明	适用机种
FC	本指令用于将字符串中指定位置开始指定长度的字符串替换成另一字符串。	DVP15MC11T DVP15MC11T-06



- **参数**

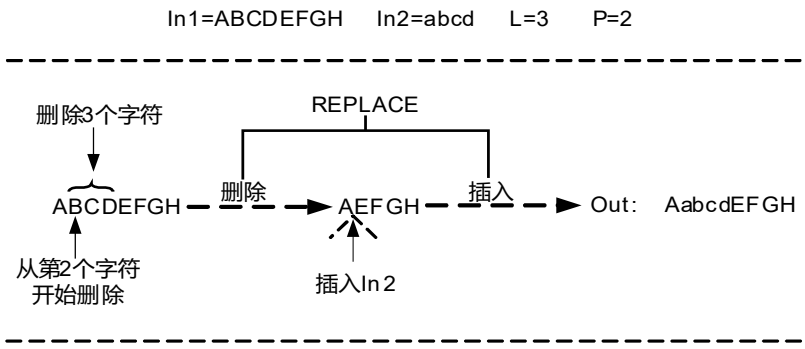
参数名称	含义	输入/输出	描述	参数取值范围
In1	原始字符串	输入	原始字符串	由与输入参数所连变量的数据类型决定
In2	替换字符串	输入	替换字符串	由与输入参数所连变量的数据类型决定
L	替换字符的个数	输入	替换字符的个数	0~字符串最大长度
P	替换字符的起始位置	输入	替换字符的起始位置	1~字符串最大长度
Out	替换后的字符串	输出	替换后的字符串	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				●
In2																				●
L							●													
P							●													
Out																				●

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将字符串（*In1*）中从第*P*个字符开始的*L*个字符替换成另一字符串（*In2*），并替换后的字符串输出至*Out*。字符串替换过程如下图所示：



注意

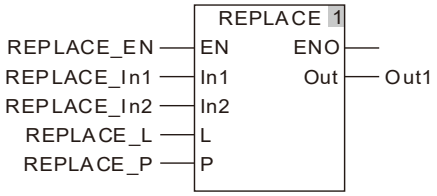
- 输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- REPLACE_In1为'AaBbCcDd'，REPLACE_In2为'DELTA'，REPLACE_L=2，REPLACE_LP=3，当REPLACE_EN为TRUE时，Out1为'AaDELTACcDd'。

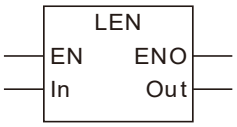
  ➢ **变量和程序**

变量名	数据类型	当前值
REPLACE_EN	BOOL	TRUE
REPLACE_In1	STRING	'AaBbCcDd'
REPLACE_In2	STRING	'DELTA'
REPLACE_L	UINT	2
REPLACE_P	UINT	3
Out1	STRING	'AaDELTACcDd'



8.15.7 LEN（计算字符串长度）

FB/FC	说明	适用机种
FC	本指令用于计算字符串的字符数。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	字符串	输入	字符串	由与输入参数所连变量的数据类型决定
Out	字符数	输出	字符数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
Out							●													

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于计算字符串的字符数，并输出至*Out*。如字符串为ABCDEFGH时，*Out*为8。

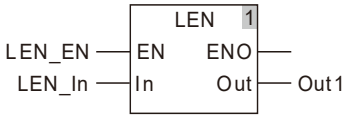
⚠ **注意**

输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

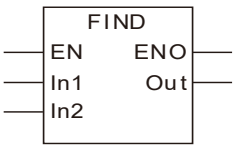
- LEN_In为’AaBbCcDd’，当LEN_EN为TRUE时，Out1为8.
  - **变量和程序**

变量名	数据类型	当前值
LEN_EN	BOOL	TRUE
LEN_In	STRING	AaBbCcDd
Out1	UINT	8



8.15.8 FIND（查找字符串）

FB/FC	说明	适用机种
FC	本指令用于查找一个字符串在另一字符串中的位置。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In1	字符串	输入	字符串	由与输入参数所连变量的数据类型决定
In2	查找的关键字符	输入	查找的关键字符	由与输入参数所连变量的数据类型决定
Out	字符数	输出	字符数	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				●
In2																				●
Out							●													

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**
  - 本指令用于将*In2*中的字符作为关键字符，并查找其在*In1*中的位置，并输出至*Out*。 如*In1*为ABCDEFGH时，*In2*为DE，则*Out*为4。
  - 查找时，从In1的第一个字符开始。
  - 若In1中无法查找到In2，则Out为0。
  - 若In1中存在多个In2，则Out为In2第一次被查找到的位置。

⚠ 注意

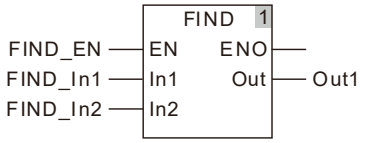
输入变量不允许缺省，若缺省，软件在编译时将报错；输出变量允许缺省。

程序范例

- FIND_In1为’AaBbCcDd’，FIND_In2为’Cc’，当FIND_EN为TRUE时，Out1为5。
  - **变量和程序**

变量名	数据类型	当前值
FIND_EN	BOOL	TRUE

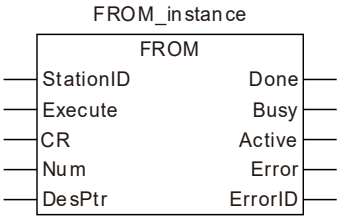
变量名	数据类型	当前值
FIND_In1	STRING	‘AaBbCcDd’
FIND_In2	STRING	‘Cc’
Out1	UINT	5



8.16 IO 刷新指令

8.16.1 FROM（模块 CR 读取数据指令）

FB/FC	说明	适用机种
FB	此指令用于读取左右侧扩展模块 CR 寄存器中的数据。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
StationID （模块位置）	左右侧模块在主机左右侧的位置	USINT	左侧模块：100~107 右侧特殊模块：0~7 主机左侧第一台模块的位置为 100，主机右侧第一台模块的位置为 0。 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当*Execute*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
CR （模块 CR 的编号）	预读取模块寄存器 CR（Controlled Register）的编号。	UINT	0~模块 CR 最大值 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Num （读取 CR 笔数）	读取模块 CR 寄存器数据的笔数。	USINT	1~64 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
DesPtr （读取到的数据）	指令读取到 CR 中数据存放区	INT 或 DINT	读取的 *CR* 寄存器数据范围 （不可缺省）	*Execute* 由 FALSE 变为 TRUE

- **输出参数**

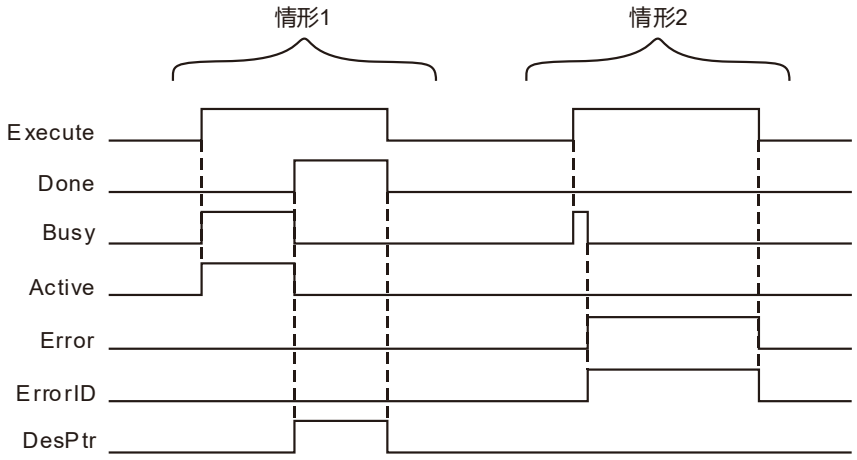
名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在执行	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当参数内容读取完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Busy 变为 FALSE
Active	◆ 当指令开始执行时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Active 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后，*Busy* 和 *Active* 变为 TRUE，且一个周期后，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，DesPtr 显示模块 CR 寄存器内对应的数据。当 *Execute* 由 TRUE 变为 FALSE 后,Done 由 TRUE 变为 FALSE，DesPtr 中值清零。

情形2： 当 *Execute* 由 FALSE 变为 TRUE，指令执行由错误产生时，Error 由 FALSE 变为 TRUE，Error ID 显示相应的错误代码。当 *Execute* 由 TRUE 变为 FALSE 后,Error 由 TRUE 变为 FALSE，Error ID 的内容清零。

- **功能说明**

此指令用于读取左右侧扩展模块寄存器中的数据。

左右侧模块位置由 StationID 指定，右侧模块 StationID 范围是 0~7，0 表示右侧第一个扩展模拟量模块，7 表示右侧第八个扩展模拟量模块；左侧模块 StationID 范围是 100~107,100 表示左侧第一个扩展模块，107 表示左侧第八个扩展模块。若 StationID 的范围超过左右侧模块指定范围时，执行该指令报错。

如果指令需要对多个 CR 读取数据，需要将引脚 DesPtr 定义为数组的第 N 个元素，执行该指令，会将第一个 CR 的数据读取到数组第 N 个元素中，将第二个 CR 的数据读取到数组第 N+1 个元素中，以此类推，将多个 CR 数据读取到数组中。具体使用介绍可参考程序范例二。

⚠ 注意

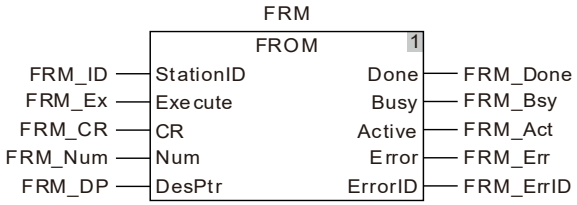
DVP-15MC 系列运动控制器左侧最多可以扩展 8 台模块，右侧最多可以扩展 8 台特殊模块，右侧数字量模块不算该编号。如右侧依次连接 DVP04AD-S、DVP16SP11T、DVP04DA-S 三个模块，则 DVP04AD-S 模块的 StationID 为 0，DVP04DA-S 模块的 StationID 为 1。

程序范例一

FROM 指令执行范例如下所示：

- **变量和程序**

变量名	数据类型	当前值
FRM	FROM	
FRM_ID	USINT	0
FRM _Ex	BOOL	FALSE
FRM _CR	UINT	0
FRM _Num	USINT	1
FRM _DP	INT	
FRM _Done	BOOL	
FRM _Bsy	BOOL	
FRM _Act	BOOL	
FRM _Err	BOOL	
FRM _ErrID	WORD	

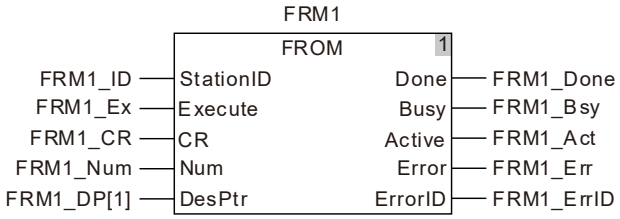


DVP-15MC 系列运动控制器右侧接 DVP-04AD，当 FRM _Ex 由 FALSE 变为 TRUE，FRM _Bsy 和 FRM _Act 同时变为 TRUE，FROM 指令开始执行，当 FRM _Done 变为 TRUE 时，指令执行完成，FRM _DP 显示指令读取 CR0 的值 136，则 DVP-04AD 的版本是 1.36。

程序范例二

■ **变量和程序**

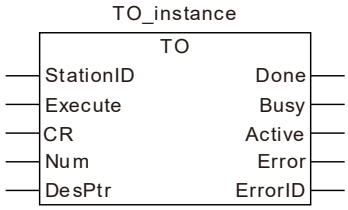
变量名	数据类型	当前值
FRM1	FROM	
FRM1_ID	USINT	0
FRM1_Ex	BOOL	FALSE
FRM1_CR	UINT	2
FRM1_Num	USINT	4
FRM1_DP	Array[1..4] of INT	
FRM1_Done	BOOL	
FRM1_Bsy	BOOL	
FRM1_Act	BOOL	
FRM1_Err	BOOL	
FRM1_ErrID	WORD	



DVP-15MC 系列运动控制器右侧接 DVP-04AD，当 FRM1_Ex 由 FALSE 变为 TRUE，FRM1_Bsy 和 FRM1_Act 同时变为 TRUE，FROM 指令开始执行，当 FRM1_Done 变为 TRUE 时，指令执行完成，读取 DVP-04AD 寄存器 CR2、 CR3、 CR4、 CR5 内的数值存放在数组 FRM1_DP 的四个元素 FRM1_DP[1]、FRM1_DP[2]、FRM1_DP[3]、FRM1_DP[4]中。

8.16.2 TO（模块 CR 写入数据指令）

FB/FC	说明	适用机种
FB	此指令用于给左右侧模块指定的 CR 寄存器写入数据。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
StationID （模块位置）	左右侧模块在主机左右侧的位置	USINT	左侧模块：100~107 右侧特殊模块：0~7 主机左侧第一台模块的位置为 100，主机右侧第一台模块的位置为 0。 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当*Execute*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	
CR （写入 CR 的编号）	预读取模块内部寄存器 *CR*（Controlled Register）起始编号。	UINT	0~模块 *CR* 最大值 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Num （写入 CR 笔数）	写入模块 *CR* 寄存器数据的笔数。	USINT	1~64 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
DesPtr （写入的数据）	指令写入到 *CR* 寄存器中的值	INT 或 DINT	写入的 *CR* 寄存器数据范围 （不可缺省）	*Execute* 由 FALSE 变为 TRUE

- **输出参数**

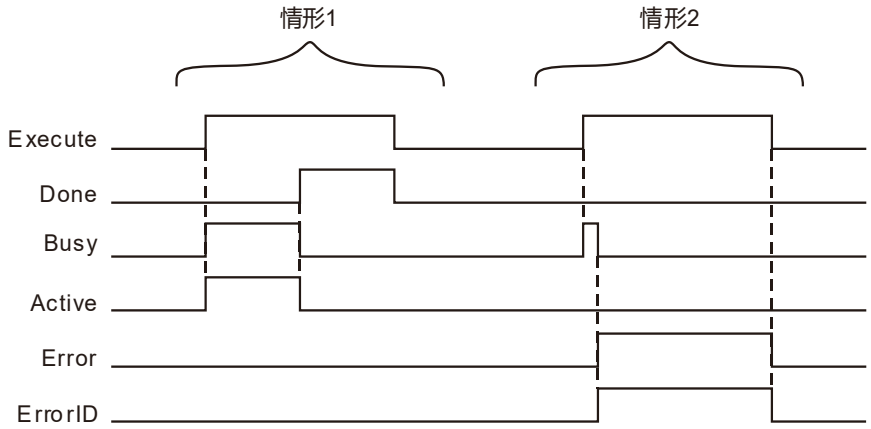
名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在执行	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当参数内容读取完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Busy 变为 FALSE
Active	◆ 当指令开始执行时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Active 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 后，Busy 和 Active 变为 TRUE，且一个周期后，Done 变为 TRUE，同时 Busy 和 Active 由 TRUE 变为 FALSE, 当 Execute 由 TRUE 变为 FALSE 后,Done 由 TRUE 变为 FALSE，

情形2： 当 *Execute* 由 FALSE 变为 TRUE 后，指令有错误产生时，Error 由 FALSE 变为 TRUE，Error ID 显示相应的错误代码 。当 *Execute* 由 TRUE 变为 FALSE 后,Error 由 TRUE 变为 FALSE，Error ID 的内容清零。

- **功能说明**

此指令用于给左右侧模块指定的寄存器写入数据。

左右侧模块位置由 StationID 指定，右侧模块 StationID 范围是 0~7，0 表示右侧第一个扩展模拟量模块，7 表示右侧第八个扩展模拟量模块；左侧模块 StationID 范围是 100~107,100 表示左侧第一个扩展模块，107 表示左侧第八个扩展模块。若 StationID 的范围超过左右侧模块指定范围时，执行该指令报错。

如果指令需要对多个 CR 写入数据，需要将引脚 DesPtr 定义为数组的第 N 个元素，执行该指令，会将数组第 N 个元素的值写入到第一个 CR 中，将数组第 N+1 个元素的值写入到第二个 CR 中，以此类推，将多笔数据写入到多个 CR 中，使用介绍可参考程序范例二。

⚠ 注意

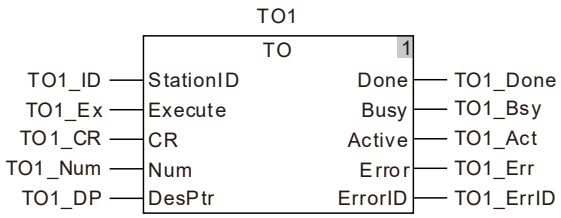
DVP-15MC 系列运动控制器左侧最多可以扩展 8 台模块，右侧最多可以扩展 8 台特殊模块，右侧数字量模块不算该编号。如右侧依次连接 DVP04AD-S、DVP16SP11T、DVP04DA-S 三个模块，则 DVP04AD-S 模块的 StationID 为 0，DVP04DA-S 模块的 StationID 为 1。

程序范例一

TO 指令执行范例如下所示：

■ **变量和程序**

变量名	数据类型	当前值
TO1	TO	
TO1_ID	USINT	0
TO1_Ex	BOOL	FALSE
TO1_CR	UINT	2
TO1_Num	USINT	1
TO1_DP	INT	10
TO1_Done	BOOL	
TO1_Bsy	BOOL	
TO1_Act	BOOL	
TO1_Err	BOOL	
TO1_ErrID	WORD	



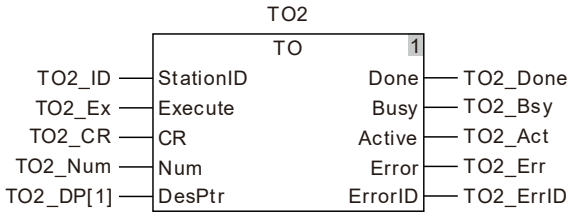
DVP-15MC 系列运动控制器右侧接 DVP-04AD，当 TO1_Ex 由 FALSE 变为 TRUE，TO1_Bsy 和 TO1_Act 同时变为 TRUE，TO 指令开始执行，当 TO1_Done 变为 TRUE 时，指令执行完成，则 DVP-04AD 的 CR2 写入值是 10。

程序范例二

■ **变量和程序**

变量名	数据类型	当前值
TO2	TO	
TO2_ID	USINT	0
TO2_Ex	BOOL	FALSE
TO2_CR	UINT	2
TO2_Num	USINT	4

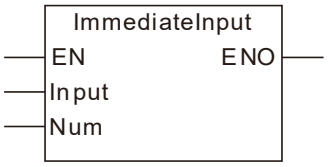
变量名	数据类型	当前值
TO2_DP	Array[1..4] of INT	
TO2_Done	BOOL	
TO2_Bsy	BOOL	
TO2_Act	BOOL	
TO2_Err	BOOL	
TO2_ErrID	WORD	



DVP-15MC 系列运动控制器右侧接 DVP-04AD，当 TO2_Ex 由 FALSE 变为 TRUE，TO2_Bsy 和 TO2_Act 同时变为 TRUE，TO 指令开始执行，当 TO2_Done 变为 TRUE 时，指令执行完成，则 DVP-04AD 的 CR2、CR3、CR4、CR5 写入值是数组 TO2_DP 四个元素 TO2_DP[1]、TO2_DP[2]、TO2_DP[3]、TO2_DP[4]中写入的值。

8.16.3 ImmediateInput（输入点立即刷新）

FB/FC	说明	适用机种
FC	本指令用于输入点立即刷新。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
Input （输入点起始点）	输入点起始点	输入	用于设置输入点起始点	0~15
Num	数量	输入	用于设置输入点立即刷新个数	1~16

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input											●									
Num						●														

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将外部输入点状态刷新至%IX0.0~%IX0.15。如果无本指令，控制器只有每次扫描程序开始时，将外部硬件输入点状态刷新至%IX0.0~%IX0.15一次。

本指令的Input参数0~15对应%IX0.0~%IX0.15，Num参数表示通过Input指定装置的连续个数。如Input的值为0，Num的值为2时，表示将外部输入点的状态刷新至%IX0.0和%IX0.1。

注意

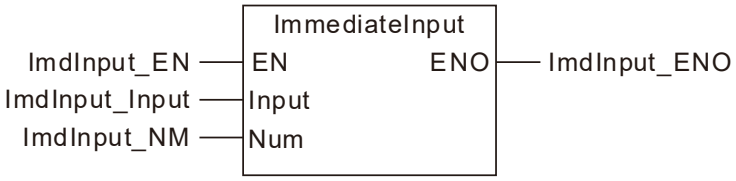
- 本指令只能用于本体输入点立即刷新，不能对扩展模块输入点立即刷新。

程序范例

- **变量和程序**

变量名	数据类型	当前值
ImdInput_EN	BOOL	FALSE
ImdInput_Input	INT	2
ImdInput_NM	USINT	2

变量名	数据类型	当前值
ImdInput_ENO	BOOL	

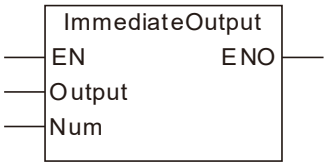


■ **程式说明**

当输入变量 ImdInput_EN 为 TRUE 时，将外部硬件输入点的状态刷新至%IX0.2 和%IX0.3。

8.16.4 ImmediateOutput（输出点立即刷新）

FB/FC	说明	适用机种
FC	本指令用于输出点立即刷新。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
Output （输出起始点）	输出起始点	输入	用于设置输出点为起始点	0~7
Num	数量	输入	用于设置输出点立即刷新个数	1~8

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Output											●									
Num						●														

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

- **功能说明**

本指令用于将内部输出点%QX0.0~%QX0.7当前状态刷新至外部硬件输出点。如果无本指令，控制器在扫描程序结束后，再将内部输出点状态刷新至外部硬件输出点。%QX0.0~%QX0.7的状态由其它指令决定，本指令只是用于将%QX0.0~%QX0.7的状态刷新至外部硬件输出点，本指令不会控制%QX0.0~%QX0.7 TRUE或者FALSE。

本指令的Output参数0~7对应 %QX0.0~%QX0.7，Num参数表示通过Output指定装置的连续个数。如Output的值为0，Num的值为2时，表示将%QX0.0和%QX0.1的状态刷新至外部硬件输出点。

⚠ 注意

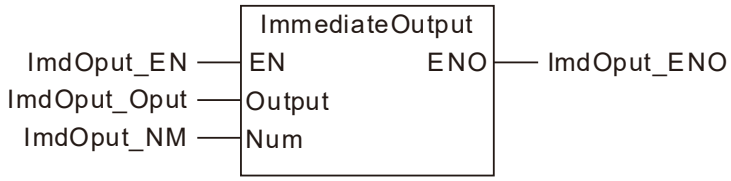
本指令只能用于本体输出点立即刷新，不能对扩展模块输出点立即刷新。

程序范例

- **变量和程序**

变量名	数据类型	当前值
ImdOput_EN	BOOL	FALSE

变量名	数据类型	当前值
ImdOput_Oput	INT	2
ImdOput_NM	USINT	2
ImdOput_ENO	BOOL	



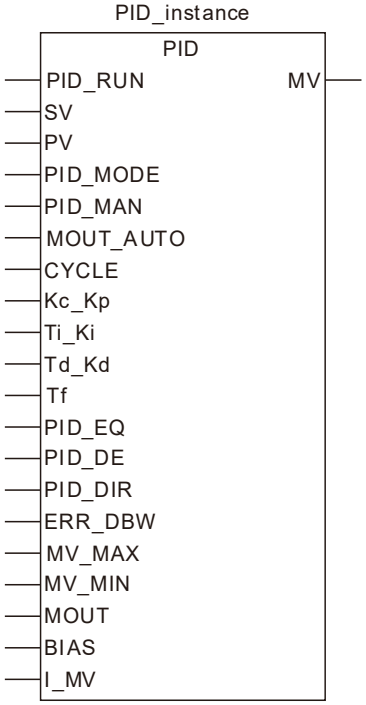
■ **程式说明**

当输入变量 ImdOput_EN 为 TRUE 时，将%QX0.2 和%QX0.3 的状态刷新至外部硬件输出点。

8.17 PID 相关指令

8.17.1 PID（PID 运算）

FB/FC	说明	适用机种
FB	此指令用于 PID 运算。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
PID_RUN	启动PID运算	输入	启动PID运算	TRUE或FALSE
SV	目标值	输入	目标值	−3.402823e+38 ~ −1.175495e-38，0，+1.175495e-38 ~ +3.402823e+38
PV	当前值	输入	当前值	
PID_MODE	PID控制模式	输入	0：自动控制，MV值会由当时输出的MV值开始进行自动运算。 1：自动调整参数功能，调整完毕时将自动进入自动控制模式（PID_MODE改为0），并且填入最适用的Kc_Kp、Ti_Ki、Td_Kd及Tf等参数。	0、1
PID_MAN	PID A/M模式	输入	TRUE：Manual FALSE：Auto	TRUE或FALSE

参数名称	含义	输入/输出	描述	参数取值范围
MOUT_AUTO	保留	输入	-	-
CYCLE	取样时间（T_S）	输入	取样时间（T_S）	1~40,000 （单位: ms）
Kc_Kp	比例项系数	输入	P计算值系数，如果小于0则Kc_Kp将为0。	0， +1.175495e-38 ~ +3.402823e+38
Ti_Ki	积分项系数	输入	I计算值系数，如果小于0则Ti_Ki将为0。	0， +1.175495e-38 ~ +3.402823e+38 （单位:Ti = sec; Ki = 1/sec）
Td_Kd	微分项系数	输入	D计算值系数，如果小于0则Td_Kd将为0。	0， +1.175495e-38 ~ +3.402823e+38 （单位: sec）
Tf	微分作用时间常数(Tf)	输入	微分作用时间常数，如果小于0则Tf将为0。	
PID_EQ	保留	输入	-	-
PID_DE	保留	输入	-	-
PID_DIR	PID正反向	输入	TRUE：正向动作（E=SV-PV） FALSE：反向动作（E=PV-SV）	TRUE或FALSE
ERR_DBW	偏差量（E）不作用范围	输入	偏差量（E）不作用范围	−3.402823e+38 ~ −1.175495e-38， 0， +1.175495e-38 ~ +3.402823e+38
MV_MAX	输出值(MV)饱和上限	输入	输出值（MV）饱和上限	
MV_MIN	输出值(MV)饱和下限	输入	输出值（MV）饱和下限	
MOUT	MV手动值	输入	MV手动值	
BIAS	前馈控制输出值	输入	**前馈控制输出值**	
I_MV	保留	输入	系统用参数，用户请勿使用	
MV	MV输出值	输出	MV值会介于MV_MAX与MV_MIN之间。	−3.402823e+38 ~ −1.175495e-38， 0， +1.175495e-38 ~ +3.402823e+38

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
PID_RUN	●																			

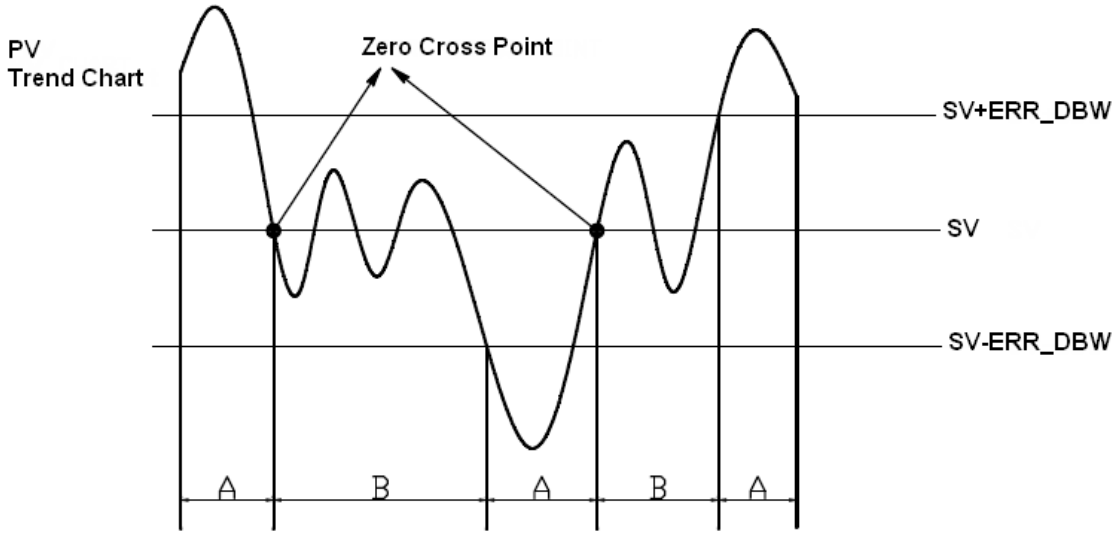
	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SV														●						
PV														●						
PID_MODE												●								
PID_MAN	●																			
MOUT_AUTO	●																			
CYCLE												●								
Kc_Kp														●						
Ti_Ki														●						
Td_Kd														●						
Tf														●						
PID_EQ	●																			
PID_DE	●																			
PID_DIR	●																			
ERR_DBW														●						
MV_MAX														●						
MV_MIN														●						
MOUT														●						
BIAS														●						
I_MV														●						
MV														●						

说明：1. 上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

2. PID 进阶运算控制的专用指令，当指令被主机执行时才会进行 PID 运算与处理。PID 表示“比例、积分和微分”。PID 控制在机械设备、气动设备和电子设备中具有广泛的应用。

- **功能说明**
  - 指令无使用次数之限制，但是I_MV所指定的变量不可被其它程序重复使用。
  - PID指令只能使用在周期性任务中。
  - PID只要被扫描到都会以取样时间（CYCLE）所设定的时间来做PID运算并直接输出MV值，并不会自动计算扫描时间是否有到达取样时间（CYCLE）才输出。

- PID的当前值（PV）在PID执行运算动作前必须是一个稳定值。如果要抓取特殊模块的输入值作PID运算时，请注意这些模块的A/D转换时间。
- 当PV值进入ERR_DBW的范围时，一开始主机仍会依照E值进行PID计算，直到PV穿过SV值时（Zero Cross Point）代表Cross Status成立，此时会将E值视为0代入PID计算，一直到PV值超出ERR_DBW的范围时才会恢复将E值代入PID计算，若PID_DE=True则表示使用PV值来进行微分项的计算，则在Cross Status条件成立后，PLC会将Δ PV视为0进行PID微分项的计算。（Δ PV=当前PV - 前次PV）
例如以下的**PV**趋势图中，A的区段主机会依照正常的PID进行计算，而B的区段主机会将E或Δ **PV**视为0进行PID计算。



➢ **PID 计算公式：**

当 **PID_MODE** 控制模式设定为 0 时，为自动控制模式。

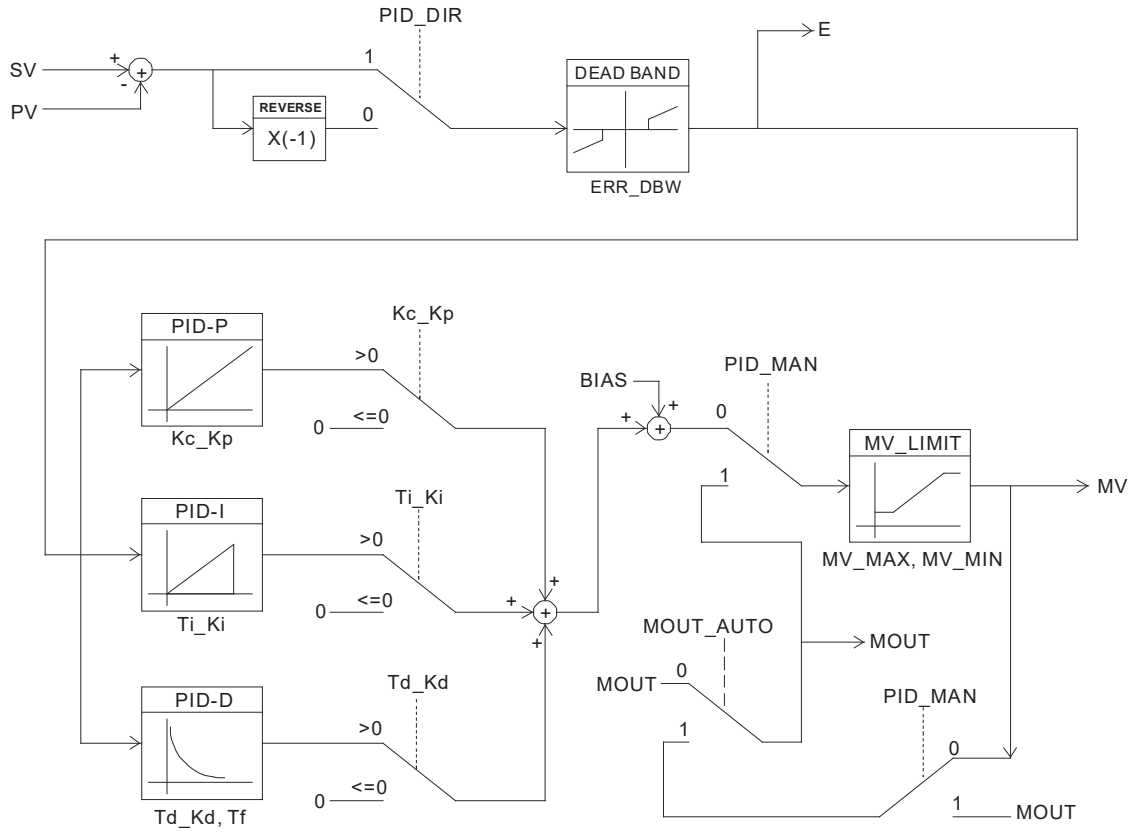
Independent Formula & Derivative of E（**PID_EQ**=False & **PID_DE**=False）

$$MV = K_P E + Ki\int_0^t Edt + K_d * \frac{dE}{dt} + BIAS$$ **其中** $E = SV - PV$ or $E = PV - SV$

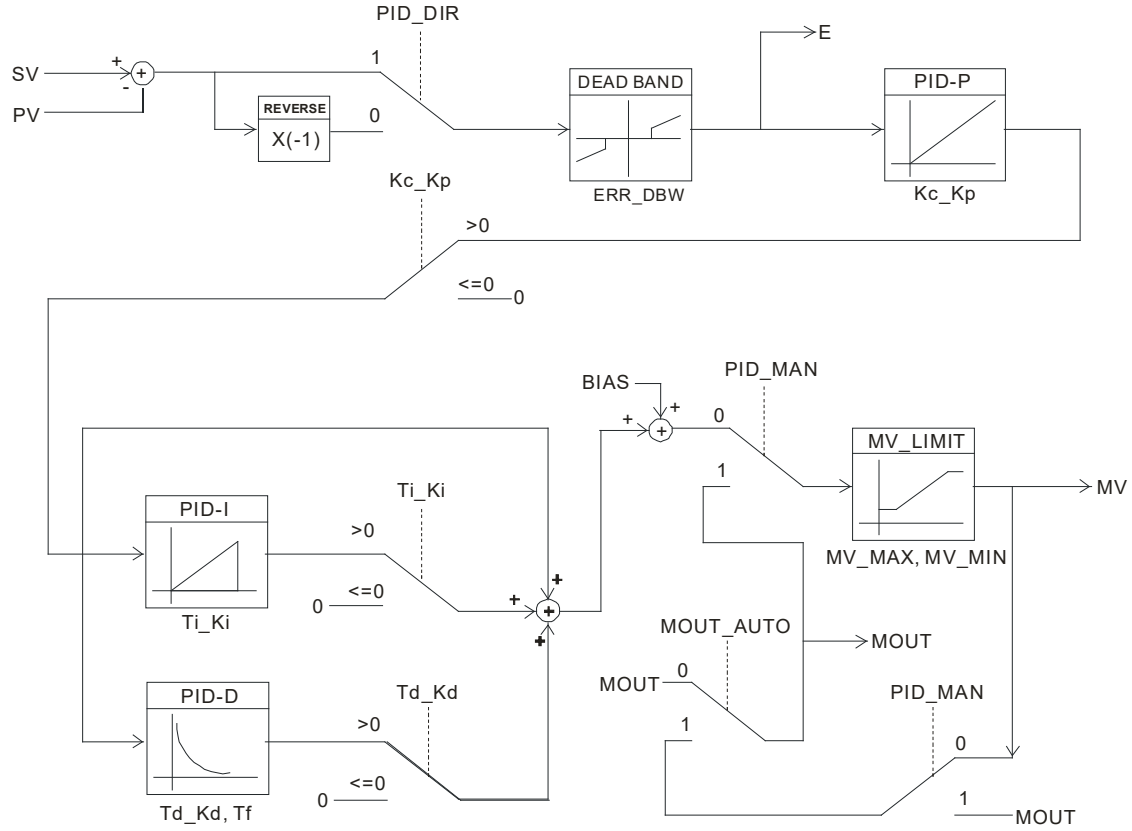
当 **PID_MODE** 控制模式选择为 1，为自动调整模式，当自动调整完成后，**PID_MODE** 会自动变成 0 转换为自动控制模式。

➢ **PID 控制方块图：**

PID Block Diagram (Independent)



PID Block Diagram (Dependent)



➢ **注意事项和建议：**

由于可使用 PID 指令的控制环境很多，因此请适当的选取控制功能，例如：当选择自动调整参数（**PID_MODE**=1）功能时，**MV** 值会在 **MAX** / **MIN** 两个值之间做切换，所以请尽量勿使用于快速反应的马达控制环境中，以免造成在自动调适时系统因快速的剧烈变化，而造成对人员危险或损坏系统的现象发生。

用户于调整 **Kc_Kp**、**Ti_Ki** 及 **Td_Kd** 三个主要参数时（**PID_MODE**=0），请先调整 **Kc_Kp** 值（依经验值设定），而 **Ti_Ki** 及 **Td_Kd** 值先设定为 0，等到调整到大致上可控制时，再依序调整 **Ti_Ki** 值（由小到大）以及 **Td_Kd** 值（由小到大）。其中 **Kc_Kp** 值为 1 则表示 100%，即对偏差值的增益为 1，小于 100%将对偏差值衰减，大于 100%将对偏差值放大。

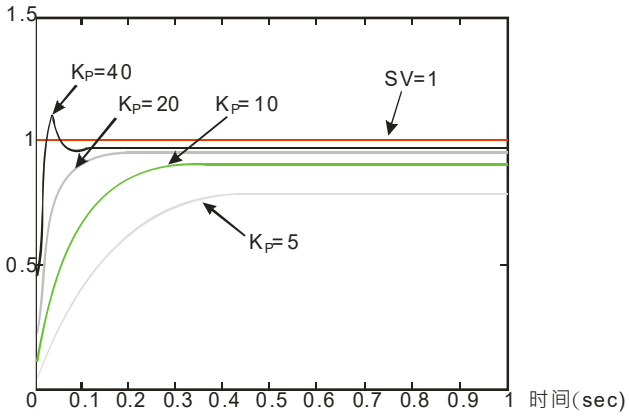
当用户选用（**PID_MODE**=1）时，建议使用具有掉电保持属性的变量来储存参数，以免自动调整过的参数因掉电后而消失。经过自动调整过的参数，并不能保证一定适用于每个控制的环境，因此用户当然可自行修改调整过的参数，不过建议最好只修改 **Ti_Ki** 或 **Td_Kd** 数值就好。

本指令动作须配合许多参数值控制，因此请勿随意设定参数值，以免造成无法控制之现象。

➢ **PID 指令参数调整建议步骤说明**

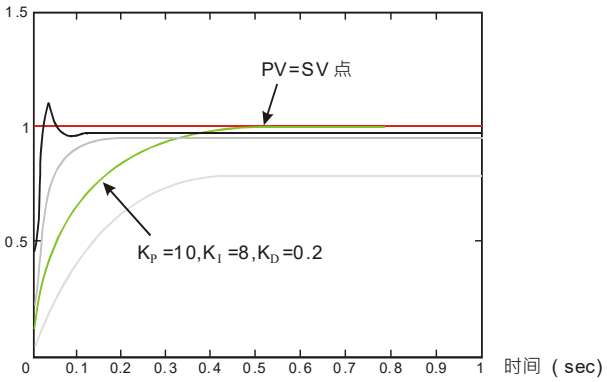
假设控制系统之受控体 G（s）的转移函数为一阶的函数 $G(s)=\frac{b}{s+a}$（一般马达的模型均为此函数），命令值 SV 为 1，取样时间 **CYCLE** 为 10ms。建议调整步骤如下：

步骤 1：首先将 K_i 及 K_d 值设为 0，接着先后分别设定 K_p 为 5、10、20 及 40，并分别记录其 SV 及 PV 状态，其结果如下图所示。



步骤 2：观察上图后得知 K_p 为 40 时，其反应会有过冲现象，因此不选用，而 K_p 为 20 时，其 **PV** 反应曲线接近 **SV** 值且不会有过冲现象，但是由于启动过快，因此输出值 **MV** 瞬间值会很大，所以考虑暂不选用，接着 K_p 为 10 时，其 **PV** 反应曲线接近 **SV** 值并且是比较平滑接近，因此考虑使用此值，最后 K_p 为 5 时，其反应过慢，因此也暂不考虑使用。

步骤 3：选定 K_p 为 10 后，先调整 K_i 值由小到大（如 1、2、4 至 8），以不超过 K_p 值为原则，然后再调整 K_d 由小到大（如 0.01、0.05、0.1 及 0.2），以不超过 K_p 的 10%为原则，最后可得如下图之 **PV** 与 **SV** 的关系图。



附注：本范例仅供参考，因此用户还需依实际控制系统之状况，自行调整其适合之控制参数。

程序范例

- **使用自动调整功能控制温度**
- **控制目的**：利用自动调整功能计算出最佳的 PID 温度控制的参数。
- **控制说明**：由于一般用户对于第一次控制的温度环境特性通常不太了解，因此可先使用自动调整功能（**PID_MODE**=1）做一初步调整，待调整完毕后，本指令将自动修改控制模式为（**PID_MODE**=0）。本实例的控制环境为烤箱。

➢ **变量和程序**

1. 全局变量表

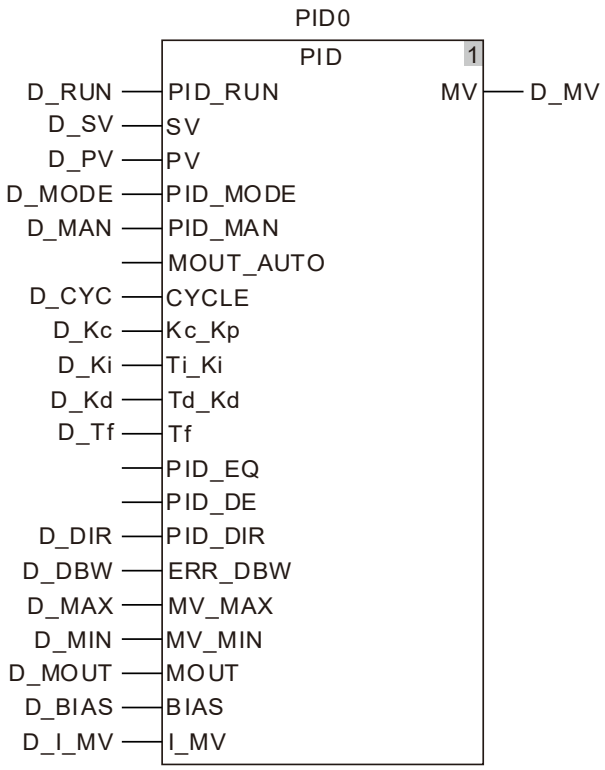
全局变量名	数据类型	当前值
M1	BOOL	FALSE
D_RUN	BOOL	
G_In1	INT	

2. 区域变量表

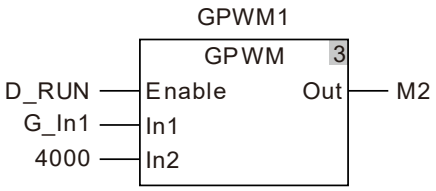
区域变量名	数据类型	当前值
PID0	PID	
D_SV	REAL	
D_PV	REAL	
D_MODE	DINT	
D_MAN	BOOL	
D_CYC	DINT	
D_Kc	REAL	
D_Ki	REAL	
D_Kd	REAL	
D_Tf	REAL	
D_DIR	BOOL	
D_DBW	REAL	
D_MAX	REAL	
D_MIN	REAL	
D_MOUT	REAL	
D_BIAS	REAL	
D_I_MV	REAL	
D_MV	REAL	

3. 程序

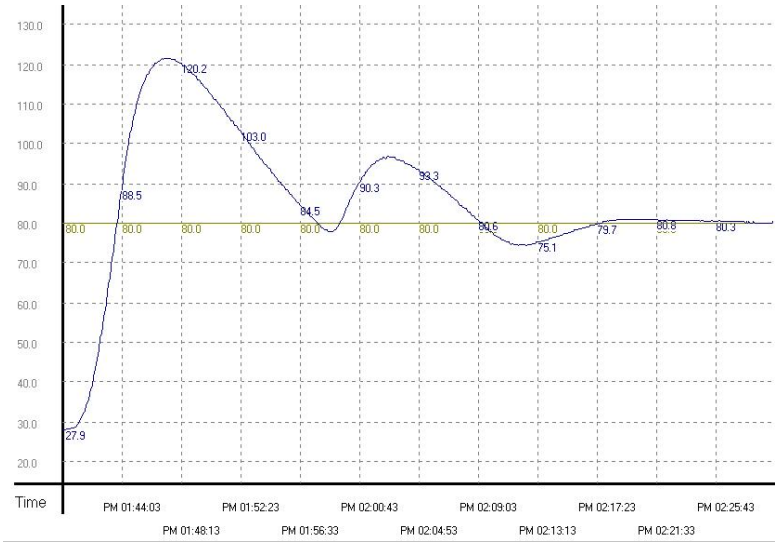
a.POU1（Task 循环类型）:



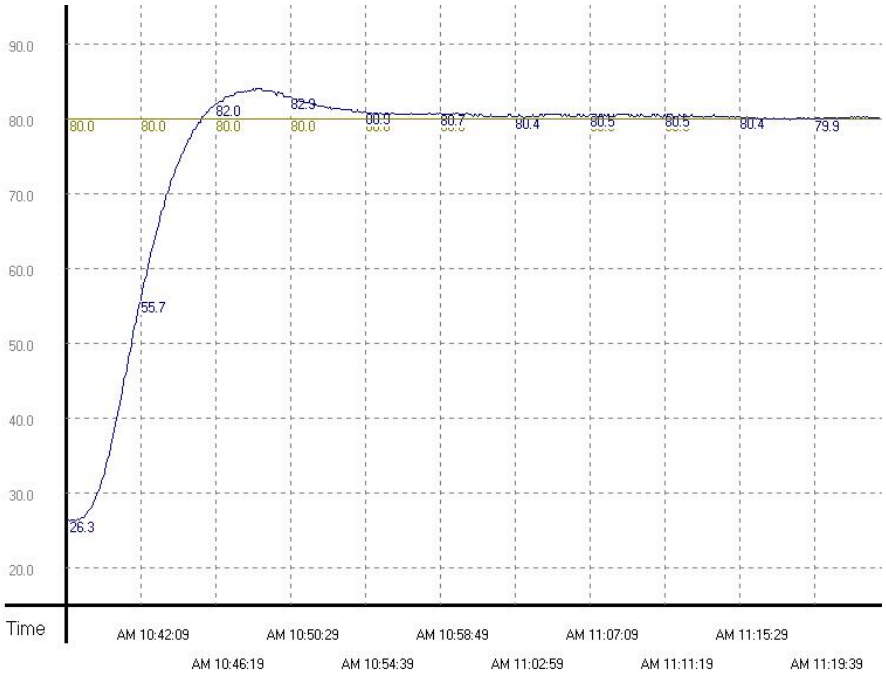
b.POU2（Task 自由运行类型）:



c.自适应功能的实验结果如下所示：

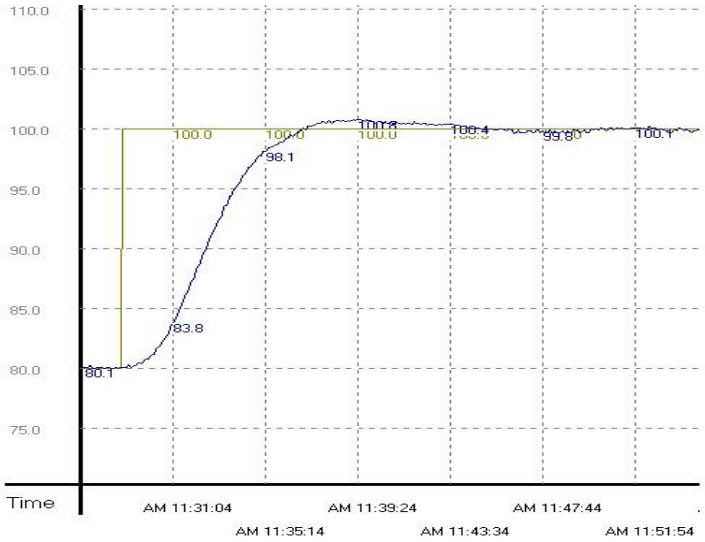


d.使用调整后参数做温度控制专用功能的实验结果如下所示：



e.由上图可看出经过自适应后的温度控制结果还不错 ,而且控制时间大约只使用了 20 分钟。

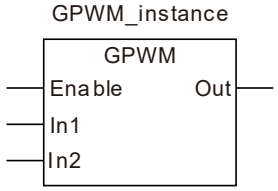
接着验证目标温度由 80 度修改成 100 度，则得到的结果如下图所示：



f. 由上图中可看出由 80 度所调整出来的参数使用到 100 度时，还是可以达到控制温度的目的，而且控制时间也不会太长。

8.17.2 GPWM(基本脉冲宽度调变)

FB/FC	说明	适用机种
FB	本指令用于脉冲输出。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
Enable	执行位	输入	当*Enable*由FALSE变为TRUE时，指令开始执行	TRUE或FALSE
In1	脉冲输出宽度	输入	指令执行时，指定脉冲输出宽度（ms）	0~32767
In2	脉冲输出周期	输入	指令执行时，指定脉冲输出周期（ms）	1~32767
Out	脉冲输出装置	输出	脉冲输出宽度内，输出为TRUE	TRUE或FALSE

	布尔	位串				整数								实数		时间，日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	●																			
In1											●									
In2											●									
Out	●																			

说明：上表中的“●”表示指令参数允许与该数据类型的变量或常量连接。

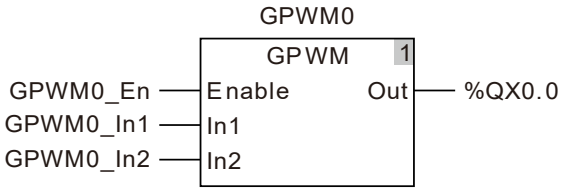
- **功能说明**
  - 本指令为脉冲输出。
  - 请将GPWM置于自由运行类型任务中使用，否则会产生GPWM输出不准确的情况。
  - In1和In2可在GPWM指令执行时更改。
  - 当In1≦0时，脉冲输出装置无输出；当In1≧In2时，脉冲输出装置一直为ON。
  - 本指令输出可以使用变量或任意位装置，装置说明请参考第3.1.2节“装置与数据类型说明”。

程序范例

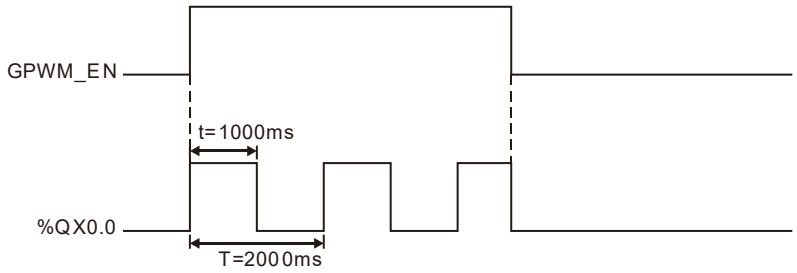
- **变量和程序**

变量名	数据类型	初始值
GPWM0	GPWM	

变量名	数据类型	初始值
GPWM0_En	BOOL	FALSE
GPWM0_In1	INT	1000
GPWM0_In2	INT	2000



■ **时序图**



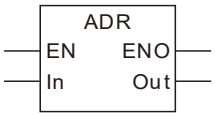
■ **补充说明**：

当 GPWM_EN 为 TRUE 时，指令正常工作；当 GPWM_EN 为 FALSE 时，指令输出立即变为 FALSE。

8.18 取地址

8.18.1 ADR（取地址）

FB/FC	说明	适用机种
FC	本指令用于取变量地址。	DVP15MC11T DVP15MC11T-06



- **参数**

参数名称	含义	输入/输出	描述	参数取值范围
In	输入参数	输入	指令取出地址变量的引脚	由与输入参数所连变量的数据类型决定
Out	运算结果	输出	取出引脚IN变量地址	由与输出参数所连变量的数据类型决定

	布尔	位串				整数								实数		时间，日期				字符串	指针
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	Pointer To xx
In	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out																					●

说明：

1. 上表中的“●”表示指令参数允许与该数据类型的变量连接。
2. **Pointer To** xx 中的 xx 须和 IN 的数据类型完全一致。

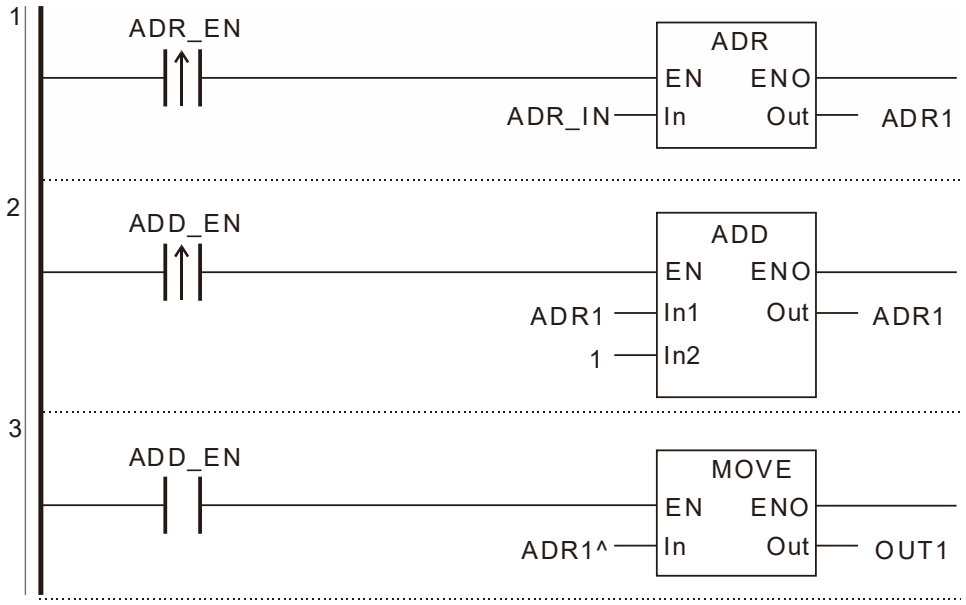
- **功能说明**

本指令用于取变量地址，取出变量地址。使用解指针符号^，取出存放在该地址中的数值。

程序范例

- ADR指令用于取出输入变量ADR_In的地址放入输出变量ADR1中，因为ADR_In变量的地址为%MW0。执行ADR指令后，ADR1指向%MW0。执行ADD指令后， ADR1指向%MW1。ADR1^表示取出ADR1地址内存放的数值（此处ADR1的值为%MW1，ADR1^表示取出%MW1地址内的数值），执行MOVE指令后，%MW1地址内的数值搬移到OUT1变量内。
  - **变量和程序**

变量名	地址	数据类型
ADR_EN		BOOL
ADR_In	%MW0	INT
ADR1		POINTER TO INT
OUT1		INT

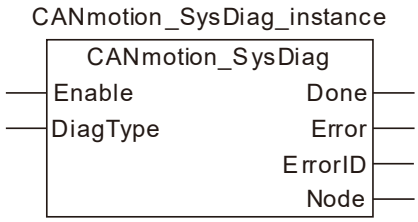


8.19 网络诊断

8.19.1 Motion 诊断

8.19.1.1 CANmotion_SysDiag（Motion 系统诊断指令）

FB/FC	说明	适用机种
FB	此指令用于诊断 Motion 通讯口连接的所有轴状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当*Enable*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	*Enable* 为 TRUE 时
DiagType（诊断类型）	1：轴是否在软件中配置； 2：轴是否和Motion通讯口建立连接； 3：轴是否发出紧急报文。	USINT	1,2,3	*Enable* 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
Node（节点）	根据输入引脚 DiagType 不同值输出所有轴对应状态。	Array[1..32] of BOOL	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行成功时	◆ 当 Enalbe 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **功能说明**

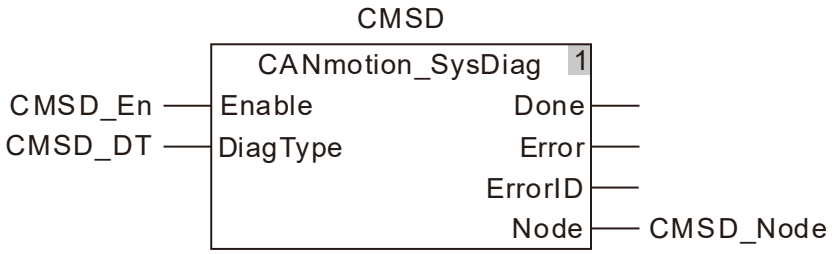
此指令用于诊断 Motion 通讯口连接的从站状态(仅 DVP-15MC 系列运动控制器内嵌 Motion 通讯口可以使用该指令)。输出参数 Node 为 BOOL 型数组，可以输出所有轴的状态。输入参数 DiagType 值不同时，输出参数 Node 的值表示的含义不同。DiagType 的值为 1 时，Node 的值表示轴是否在软件中配置，为 TRUE 表示在软件中配置，为 FALSE 表示没有在软件中配置。DiagType 的值为 2 时，Node 的值表示轴是否和 Motion 通讯口建立连接，为 TRUE 表示建立连接，为 FALSE 表示没有建立连接。DiagType 的值为 3 时，Node 的值表示轴是否发出过紧急报文，为 TRUE 表示发出过，为 FALSE 表示没有发出过。如 DiagType 的值为 2 时，输出参数 Node 对应的变量为 a，则 1 号轴和 Motion 通讯口建立连接时，a[1]的值为 TRUE，轴 1 Motion 通讯线拔掉时，a[1]的值为 FALSE。

程序范例

1. 变量表

区域变量名	数据类型	初始值
CMSD	CANmotion_SysDiag	
CMSD_EN	BOOL	
CMSD_DT	USINT	2
CMSD_Node	ARRAY [1..32] OF BOOL	

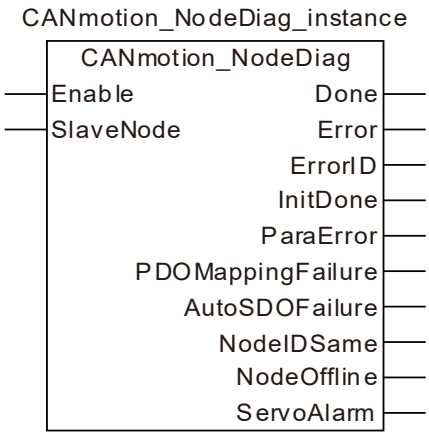
2. 程序



通过 CMSD_Node 数组成员的值可以知道对应的轴是否和控制器的 Motion 通讯口建立连接。如果 CMSD_Node[1]=1,表示 1 号轴和 Motion 通讯口已建立连接;如果 CMSD_Node[1]=0 表示 1 号轴和 Motion 通讯口未建立连接 ,原因可能是 Motion 通讯口和 1 号轴之间的通讯线没有连接或者通讯线插上后又拔掉。如果判断 2 号轴是否和 Motion 通讯口去建立连接 ,可以使用 CMSD_Node[2]的值判断 ,其他轴以此类推。

8.19.1.2 CANmotion_NodeDiag（Motion 轴诊断指令）

FB/FC	说明	适用机种
FB	此指令用于诊断 Motion 通讯口连接的指定轴的状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当*Enable*由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	*Enable* 为 TRUE 时
SlaveNode（诊断轴轴号）	诊断轴轴号	USINT	请参考第 2.2 节**功能简介**（不可缺省）	*Enable* 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
InitDone（初始化完成）	该参数为 TREU 时表示从轴初始化完成	BOOL	TRUE / FALSE
ParaError（关键参数不符）	该参数为 TREU 时表示从轴关键参数不符	BOOL	TRUE / FALSE
PDOMappingFailure（PDO 映射失败）	该参数为 TREU 时表示从轴 PDO 映射失败	BOOL	TRUE / FALSE
AutoSDOFailure（自动 SDO 失败）	该参数为 TREU 时表示从轴自动 SDO 失败	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
NodeIDSame （主从站站号重复）	该参数为 TREU 时表示主从站站号重复	BOOL	TRUE / FALSE
NodeOffLine （从轴掉线）	该参数为 TREU 时表示从轴掉线	BOOL	TRUE / FALSE
ServoAlarm （伺服报警代码）	记录伺服报警代码	UINT	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当指令执行成功时	◆ 当 Enalbe 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

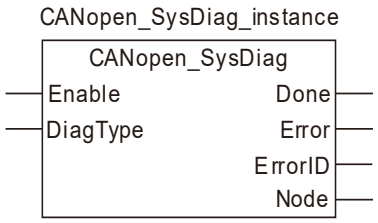
- **功能说明**

此指令用于诊断 Motion 通讯口连接的指定轴的状态(仅 DVP-15MC 系列运动控制器内嵌 Motion 通讯口可以使用该指令)。如果软件中配置的轴不能正常使用，可以使用该指令诊断错误原因。

8.19.2 CANopen 诊断

8.19.2.1 CANopen_SysDiag（CANopen 系统诊断指令）

FB/FC	说明	适用机种
FB	此指令用于诊断 CANopen 通讯口连接的所有从站状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当*Enable*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	*Enable* 为 TRUE 时
DiagType （诊断类型）	1 :从站是否在软件中配置； 2 : 从站是否和CANopen通讯口建立连接； 3 :从站是否发出紧急报文。	USINT	1,2,3 （不可缺省）	*Enable* 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
Node （节点）	根据输入引脚 DiagType 不同值输出所有从站对应状态。	Array[1..32] of BOOL	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行成功时	◆ 当 Enalbe 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **功能说明**

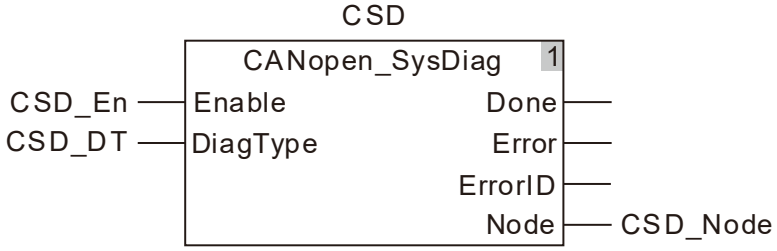
此指令用于诊断 CANopen 通讯口连接的从站状态 (仅 DVP-15MC 系列运动控制器内嵌 CANopen 通讯口可以使用该指令)。输出参数 Node 为 BOOL 型数组，可以输出所有从站的状态。输入参数 DiagType 值不同时，输出参数 Node 的值表示的含义不同。DiagType 的值为 1 时，Node 的值表示从站是否在软件中配置，为 TRUE 表示在软件中配置，为 FALSE 表示没有在软件中配置。DiagType 的值为 2 时，Node 的值表示从站是否和 CANopen 通讯口建立连接，为 TRUE 表示建立连接，为 FALSE 表示没有建立连接。DiagType 的值为 3 时，Node 的值表示从站是否发出过紧急报文，为 TRUE 表示发出过，为 FALSE 表示没有发出过。如 DiagType 的值为 2 时，输出参数 Node 对应的变量为 a，则 1 号从站和 CANopen 通讯口建立连接时，a[1]的值为 TRUE，1 号从站 CANOpen 通讯线拔掉时，a[1]的值为 FALSE。

程序范例

1. 变量表

区域变量名	数据类型	当前值
CSD	CANOpen_SysDiag	
CSD_EN	BOOL	
CSD_DT	USINT	2
CSD_Node	ARRAY [1..32] OF BOOL	

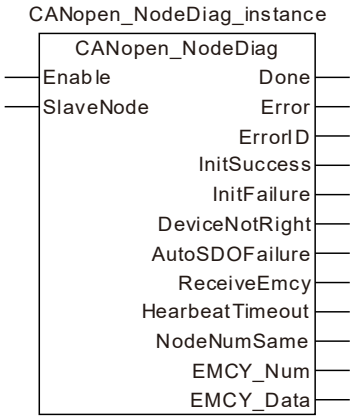
2. 程序



通过 CSD_Node 数组成员的值可以知道对应的从站是否和控制器的 CANopen 通讯口建立连接。如果 CSD_Node[1]=1,表示 1 号轴和 CANopen 通讯口已建立连接;如果 CSD_Node[1]=0 表示 1 号轴和 CANopen 通讯口未建立连接，原因可能是 CANopen 通讯口和 1 号轴之间的通讯线没有连接或者通讯线插上后又拔掉。如果判断 2 号轴是否和 CANopen 通讯口去建立连接，可以使用 CSD_Node[2]的值判断，其他从站以此类推。

8.19.2.2 CANopen_NodeDiag（CANopen 从站诊断指令）

FB/FC	说明	适用机种
FB	此指令用于诊断 CANopen 通讯口连接的指定从站的状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当*Enable*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	*Enable* 为 TRUE 时
SlaveNode（诊断从站站号）		USINT	1~32（不可缺省）	*Enable* 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
InitSuccess（初始化成功）	该参数为 TRUE 时表示从站初始化成功	BOOL	TRUE / FALSE
InitFailure（初始化失败）	该参数为 TRUE 时表示从站初始化失败	BOOL	TRUE / FALSE
DeviceNotRight（从站设备不正确）	该参数为 TRUE 时表示从站设备不正确	BOOL	TRUE / FALSE
AutoSDOFailure（自动 SDO 失败）	该参数为 TRUE 时表示从站自动 SDO 失败	BOOL	TRUE / FALSE
ReceiveEmcy	该参数为 TRUE 时表示从站接收	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
（从站接收 Emergency 报文）	Emergency 报文		
HeartbeatTimeout （Heartbeat 报文超时）	该参数为 TRUE 时表示 Heartbeat 报文超时	BOOL	TRUE / FALSE
NodeNumSame （主从站站号重复）	该参数为 TRUE 时表示主从站站号重复	BOOL	TRUE / FALSE
EMCY _ Num （错误报文数量）	记录控制器接收紧急报文数量。	USINT	0~5
EMCY_Data （错误报文）	记录控制器接收到从站发送的紧急报文。	ARRAY [1..5] OF CANopen_EMCY_Type	-

- **输出参数刷新时机**

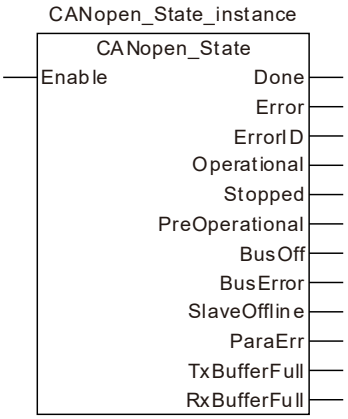
名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当指令执行成功时	◆ 当 Enalbe 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **功能说明**

此指令用于诊断 CANopen 通讯口连接的指定从站的状态（仅 DVP-15MC 系列运动控制器内嵌 CANopen 通讯口可以使用该指令）。如果软件中配置的从站不能正常使用，可以使用该指令诊断错误原因。此指令还会记录从站发送紧急报文次数以及紧急报文信息。

8.19.2.3 CANopen_State（CANopen 主站诊断指令）

FB/FC	说明	适用机种
FB	此指令用于诊断 CANopen 通讯口本身状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable （执行位）	当*Enable*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	*Enable* 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
Operational （运行状态）	该输出参数为 TRUE 时表示主站处于运行状态	BOOL	TRUE / FALSE
Stopped （停止状态）	该输出参数为 TRUE 时表示主站处于停止状态	BOOL	TRUE / FALSE
PreOperational （预运行状态）	该输出参数为 TRUE 时表示主站处于预运行状态	BOOL	TRUE / FALSE
BusOff （总线断开）	该输出参数为 TRUE 时表示总线干扰过大或者网络中有不同波特率的产品。	BOOL	TRUE / FALSE
BusError （总线出错）	该输出参数为 TRUE 时表示总线出错	BOOL	TRUE / FALSE
SlaveOffline （从站掉线）	该输出参数为 TRUE 时表示有从站掉线	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ParaError （主从站配置参数错误）	该输出参数为 TRUE 时表示主从站配置参数错误	BOOL	TRUE / FALSE
TxBufferFull （主站发送缓冲区满）	该输出参数为 TRUE 时表示主站发送缓冲区满	BOOL	TRUE / FALSE
RxBufferFull （主站接收缓冲区满）	该输出参数为 TRUE 时表示主站接收缓冲区满	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当指令执行成功时	◆ 当 Enalbe 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **功能说明**

此指令用于诊断 CANopen 通讯口本身状态（仅 DVP-15MC 系列运动控制器内嵌 CANopen 通讯口可以使用该指令）。诊断 CANopen 通讯口本身状态及是否有从站掉线。

9

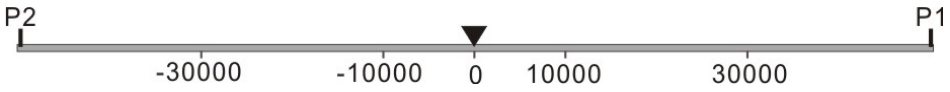
第9章 轴参数介绍

目录

9.1 轴参数详述

序号	参数名称	功能	数据类型	默认值
1	名称	轴的描述名称	STRING	-
“名称”是软件对伺服驱动器的批注文字。它没有实际的意义，只是用于命名伺服驱动器。				
2	站号	轴号，取值范围，请参考第 2.2 节**功能简介**	USINT	-
“站号”是伺服驱动器所对应的 CANopen 站号。				
3	轴类型和单元	轴类型： 用以选择轴的类型（直线轴或者旋转轴）	-	直线轴
		单元：导程的单位。如用户可以在单元后填写毫米或角度等。		unit

直线轴：

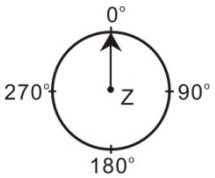


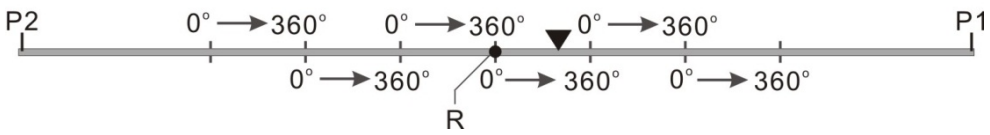
直线轴模型

直线轴模型注解：

P1	正极限
P2	反极限
▼	伺服所在的位置

旋转轴：





旋转轴模型（“模”设为 360）

旋转轴模型注解：

P1	正极限
P2	反极限
▼	伺服所在的位置

序号	参数名称	功能	数据类型	默认值
R	原点位置			
Z	伺服电机的轴心			

直线轴与旋转轴的差异：

直线轴和旋转轴的差异主要在于旋转轴以模为周期。直线轴终端执行机构的位置 500，对应旋转轴的位置为 140（模为 360 时），计算方法为 500 除以模后所得的余数。

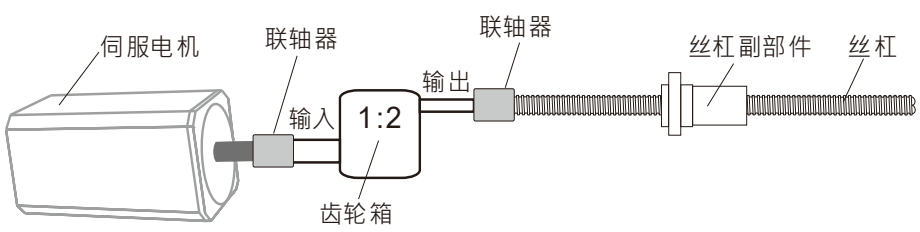
4	模	用以平分终端执行机构实际位置的周期。	LREAL	360
5	软件限制	软件限制使能。 不勾选：软件限制轴的最大/最小位置无效； 勾选：软件限制轴的最大/最小位置有效。	BOOL	0
6	最大位置	软件限制轴的最大位置	LREAL	-
7	最小位置	软件限制轴的最小位置	LREAL	-
8	最大分辨率	伺服脉冲数最大分辨率	UDINT	1280000
9	伺服驱动器 电子齿轮比分子	和电子齿轮比分母一起设定电机转一圈需要的脉冲数	UINT	128
10	伺服驱动器 电子齿轮比分母	和电子齿轮比分子一起设定电机转一圈需要的脉冲数	UINT	1
11	脉冲数/转	伺服电机转一圈需要的脉冲数	UINT	10000

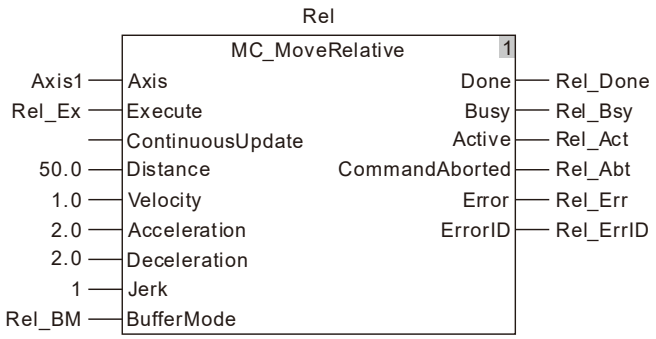
电子齿轮比分子和电子齿轮比分母这两个参数互相配合，用来设置伺服驱动器的电子齿轮比。电子齿轮比用以设定伺服驱动器接收多少个脉冲伺服电机转一圈。

伺服驱动器电机的分辨率为 1280000 脉冲/圈，假设参数 11（脉冲数/转）的值为 N，则 N × (电子齿轮比分子 / 电子齿轮比分母) = 1280000。

12	齿轮箱的输入	与齿轮箱输出一起确定机构齿轮比	UINT	1
13	齿轮箱的输出	与齿轮箱输入一起确定机构齿轮比	UINT	1
14	导程	齿轮箱输出端转一圈对应终端执行机构移动的单元数目	UINT	10000

如下图所示，齿轮箱的输入=1，齿轮箱的输出=2，表示齿轮箱输入机构转 1 圈，齿轮箱输出机构转 2 圈。机构导程表示齿轮箱输出机构转 1 圈对应“丝杠副部件”移动的单元数目。如实际齿轮箱输出转一圈“丝杠副部件”移动的距离为 1 毫米，机构导程设置为 1 时，通过相对位移运动指令走 1 个单元，“丝杠副部件”移动的距离为 1 毫米；机构导程设置为 1000 时，通过相对位移运动指令走 1 个单元，“丝杠副部件”移动的距离为 1/1000 毫米。运动控制指令、G 代码、电子凸轮中位置的单位都为单元。



<table>
<tr><th>序号</th><th>参数名称</th><th>功能</th><th>数据类型</th><th>默认值</th></tr>
<tr><td colspan="5">如上所述，设置机构导程为 1，则下图所示的相对位移指令执行后，丝杠副部件会走 50 毫米，速度为 1 毫米/秒，加速度为 2 毫米/ s^2，加速度的变化率为 1 毫米/ s^3。
变量和程序
<table>
<tr><th>变量名称</th><th>数据类型</th><th>初始值</th></tr>
<tr><td>Rel</td><td>MC_MoveRelative</td><td></td></tr>
<tr><td>Rel_Ex</td><td>BOOL</td><td>FALSE</td></tr>
<tr><td>Rel_BM</td><td>MC_Buffer_Mode</td><td>0</td></tr>
<tr><td>Rel_Done</td><td>BOOL</td><td></td></tr>
<tr><td>Rel_Bsy</td><td>BOOL</td><td></td></tr>
<tr><td>Rel_Act</td><td>BOOL</td><td></td></tr>
<tr><td>Rel_Abt</td><td>BOOL</td><td></td></tr>
<tr><td>Rel_Err</td><td>BOOL</td><td></td></tr>
<tr><td>Rel_ErrID</td><td>WORD</td><td></td></tr>
</table>


</td></tr>
<tr><td>15</td><td>原点回归</td><td>设置伺服驱动器原点回归模式，取值范围 1~35。详细说明参考附录</td><td>UINT</td><td>1</td></tr>
<tr><td rowspan="2">16</td><td>速度 1</td><td>原点回归开始到找到原点开关的速度。单位:转/分钟。数值范围：1~2000 转/分钟。</td><td>UDINT</td><td>20</td></tr>
<tr><td>速度 2</td><td>找到原点开关后到机械原点的速度，单位:转/分钟，数值范围：1~500 转/分钟。</td><td>UDINT</td><td>10</td></tr>
</table>

10

第10章 运动控制功能

目录

DVP-15MC 系列运动控制器是遵循 CANopen DSP402 运动控制协议进行开发的运动控制器，需要使用作为功能块定义的运动控制指令。

MC 功能模块的运动控制指令是以 PLCopen 的运动控制功能块的技术规格作为基础的。

下面将介绍一些在使用运动控制指令时必须了解的内容。

10.1 EN 和 ENO 说明

当用户使用的指令带有 EN 和 ENO 时，若 EN 引脚的参数值为 FALSE （0），那么指令定义的功能将不会被执行，指令的输出引脚输出值不会被刷新。相反，若指令定义的 EN 引脚参数值为 TRUE （1），那么指令定义的功能将会被执行，指令输出引脚的值也会被刷新。

ENO 引脚输出和 EN 引脚的输入保持一致，EN 引脚为 TRUE，ENO 引脚同时变为 TRUE； EN 引脚为 FALSE，ENO 引脚同时变为 FALSE。

当指令为功能块（FB）时，如果功能块（FB）执行后，EN 由 TRUE 变为 FALSE，该功能块（FB）仍继续执行，只是该功能块（FB）输出引脚的值不会被刷新。

10.2 速度、加速度、加速度的变化率之间的关系

DVP-15MC 系列运动控制器使用的是二次曲线加减速方式，这种加减速方式速度波形为 S 型，可以有效减少机械冲击。这样在使用运动控制指令时，至少需要指定速度 *v*、加速度 *Acc* 或减速度 *Dec* 和加速度的变化率 *Jerk*。

速度：表示轴执行过程中最大的速度量，单位是“单元/秒”。

加速度：表示轴执行过程中最大的加速度量，单位是“单元/秒 ^{2}”。

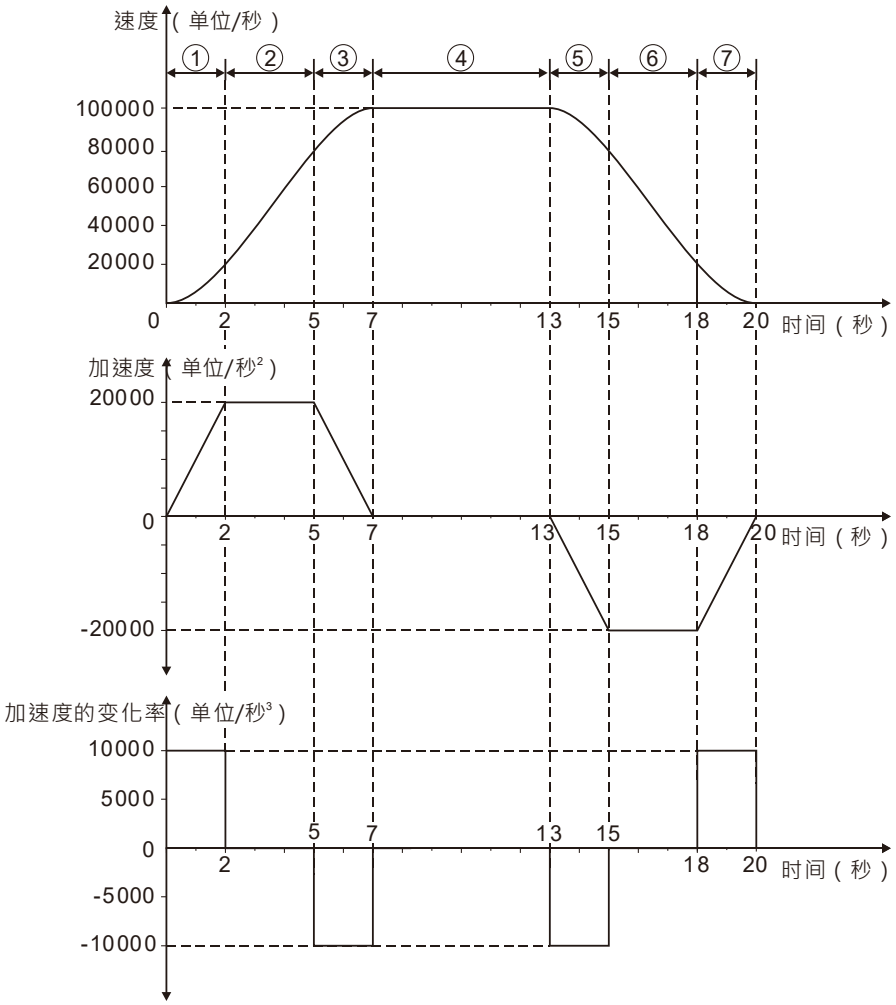
加速度的变化率：表示轴执行过程中最大的加（减）速度的变化比率，也有称为“加加速度”，单位是“单元/秒 ^{3}”。在指令中可以指定“Jerk”的值，在轴加速和减速过程中都使用指令中指定的“Jerk”值。更改“Jerk”的值，可以改善速度的平滑性。

- **速度、加（减）速度和加速度的变化率之间的关系如下：**

$$Acc\ (Dec) = \frac{dv}{dt}$$

$$Jerk = \frac{dAcc}{dt}$$

即加（减）速度是单位时间内速度的变化量；加速度的变化率是单位时间内加速度的变化量。这里使用一个相对位移指令来进一步表示这三者之间的关系，相对位移指令位移距离为 1300000 单元，速度为 100000 单元/秒，加速度为 20000 单元/秒 ^{2}，加速度的变化率为 10000 单元/秒 ^{3}，下图为这三者之间的关系。



- **速度、加速度和加速度的变化率关系图说明如下：**

阶段	时间(秒)	加速度的变化率	加速度（减速度）	速度	运动类型
1	0~2	固定为 10000 单元/秒 ^{3}	加速度增大到 20000 单元/秒 ^{2}	速度增大	加速度增加的加速运动
2	2~5	固定为 0 单元/秒 ^{3}	加速度维持不变为 20000 单元/秒 ^{2}	速度增大	加速度恒定的加速运动
3	5~7	固定为-10000 单元/秒 ^{3}	加速度减小到 0 单元/秒 ^{2}	速度增大到 100000 单元/秒	加速度减小的加速运动
4	7~13	固定为 0 单元/秒 ^{3}	加速度此阶段已经减为 0 单元/秒 ^{2}，维持不变	速度不变，为 100000 单元/秒	匀速运动
5	13~15	固定为-10000 单元/秒 ^{3}	减速度增大到 20000 单元/秒 ^{2}	速度减小	减速度增大的减速运动
6	15~18	固定为 0 单元/秒 ^{3}	减速度已经增大到 20000 单元/秒 ^{2}，维持不变	速度减小	减速度恒定的减速运动
7	18~20	固定为 10000 单元/秒 ^{3}	减速度减小到 0 单元/秒 ^{2}	速度减小到 0	减速度减小的减速运动

10.3 BufferMode 功能介绍

对于同一轴，当有运动指令控制轴在运动过程中，可以启动其它运动指令，前后两个运动指令进行交接时，有 6 种模式可供选择的交接模式，交接模式可以根据后一个运动指令的 BufferMode 引脚参数设置来选择。

BufferMode（交接模式）相关术语的含义如下：

1. 当前指令：当前控制轴的运动指令
2. 交接指令：等待执行的指令
3. 交接速度：当前指令切换到交接指令时的速度
4. 目标速度：指令中的 Velocity 引脚参数
5. 目标位置：位移相关指令中 Position 或 Distance 引脚参数

- **6 种交接模式**

交接模式	动作说明
0：mcAborting （打断）	立即打断当前指令的动作并执行交接指令的动作
1：mcBuffered （等待）	等待当前指令的动作正常执行结束后，并立即执行交接指令的动作
2：mcBlendingLow （以低速交接）	等待当前指令的目标位置到达后，并立即执行交接指令的动作；交接速度为当前指令与交接指令中较低的目标速度。
3：mcBlendingPrevious （以前一指令的速度交接）	等待当前指令的目标位置到达后，并立即执行交接指令的动作；交接速度为当前指令的目标速度。
4：mcBlendingNext （以后一指令的速度交接）	等待当前指令的目标位置到达后，并立即执行交接指令的动作；交接速度为交接指令的目标速度。
5：mcBlendingHigh （以高速交接）	等待当前指令的目标位置到达后，并立即执行交接指令的动作；交接速度为当前指令与交接指令中较高的目标速度。

注意：

1. 同一个轴只支持一级交接，如果进行多级交接，则指令会报 Error。
   例如：当指令 2 和指令 3 BufferMode 参数不为 mcAborting，指令 2（交接指令）交接指令 1（当前指令），指令 1 还未执行完时，再用指令 3 去交接指令 2，指令 3 会报 Error。若指令 3 的 BufferMode 参数为 mcAborting，则会立即打断指令 1 和指令 2，并立即执行指令 3。
2. MC_MoveSuperimposed 指令单独控制轴时，再执行交接指令（除 MC_MoveAdditive）且无论 BufferMode 选择何种交接模式都打断 MC_MoveSuperimposed 指令；当前指令和 MC_MoveSuperimposed 或 MC_HaltSuperimposed 指令共同控制轴，再执行其它的运动指令时，若 BufferMode=mcAborting，则前面所有的指令都被打断，若 BufferMode=mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh，当前指令与交接指令按照设置的 BufferMode 交接模式进行交接，而不会对 MC_MoveSuperimposed 或 MC_HaltSuperimposed 指令有任何影响。

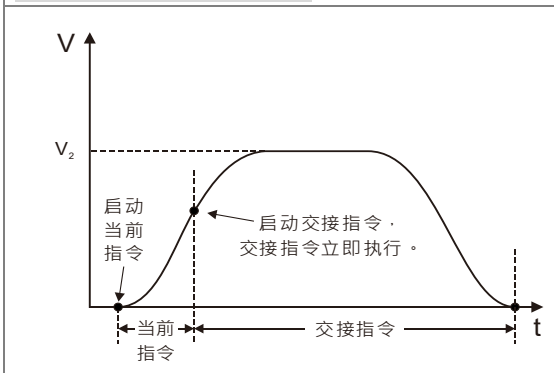
程序范例

用两个相对位移指令来简单说明一下

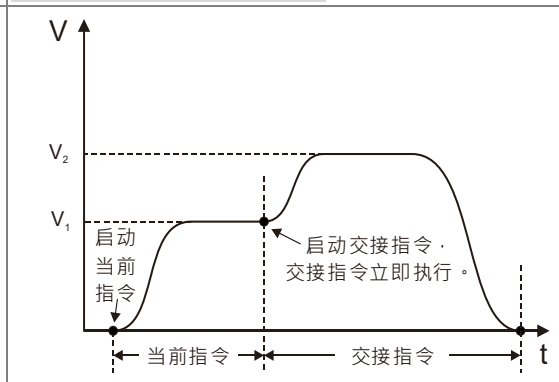
第一个相对位移指令速度最大是 v_1，位移量为 S_1，第二个相对位移指令速度最大是 v_2，位移量是 S_2。改变第二个位移指令的 BufferMode 使得这两个指令有不同的交接过程，如下说明：

- **打断 Buffermode=mcAborting，举例说明四种情况，分别如下：**

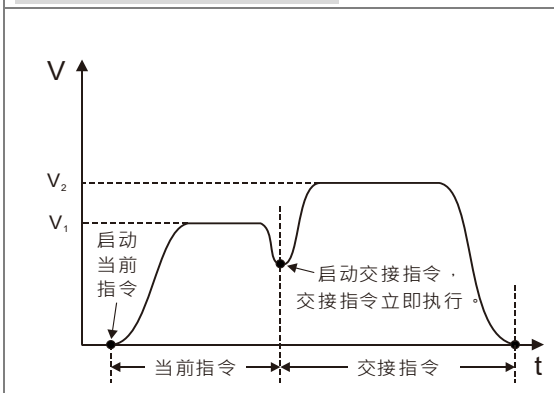
1. 当前指令加速时打断



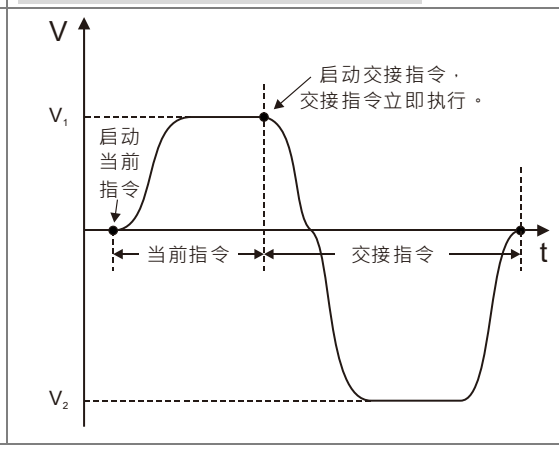
2. 当前指令匀速阶段打断



3. 当前指令减速阶段打断

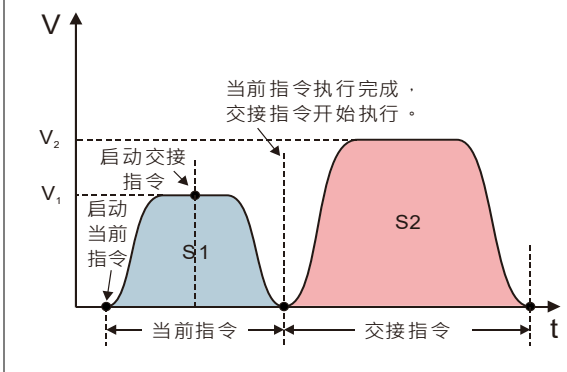


4. 交接指令与当前指令方向相反

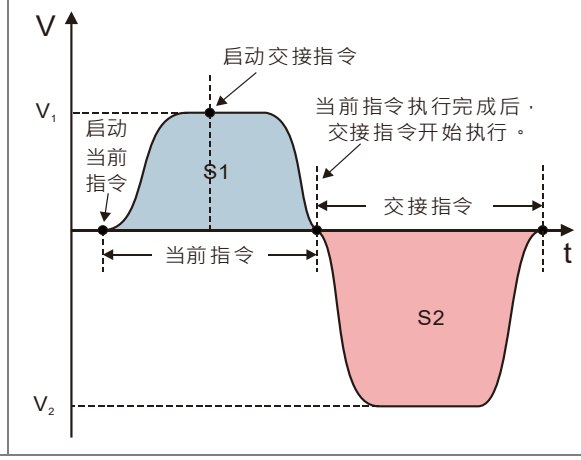


- **等待 Buffermode=mcBuffered，举例说明二种情况，分别如下：**

1. 交接指令与当前指令方向相同

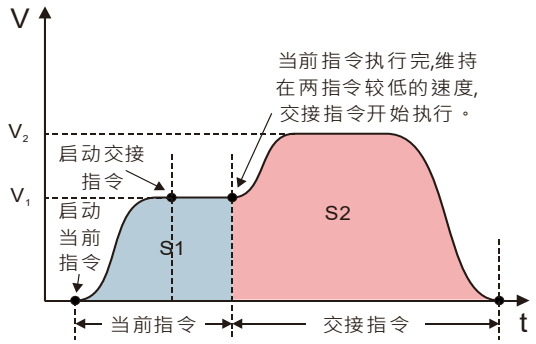


2. 交接指令与当前指令方向相反

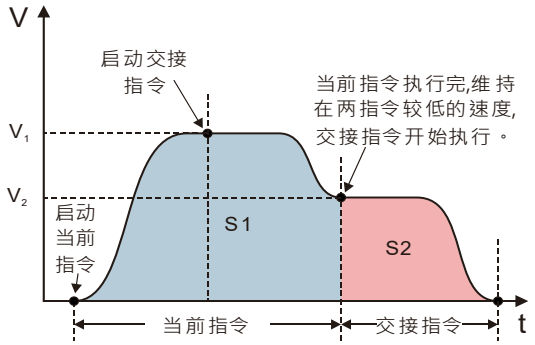


■ **低速交接 Buffermode=mcBlendingLow，举例说明三种情况，分别如下：**

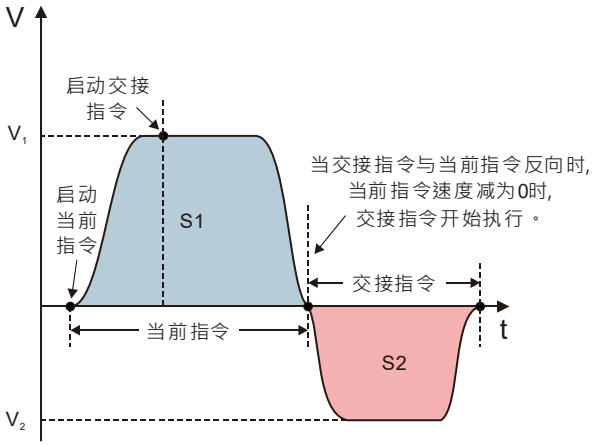
1. 当前指令比交接指令速度小



2. 当前指令比交接指令速度大

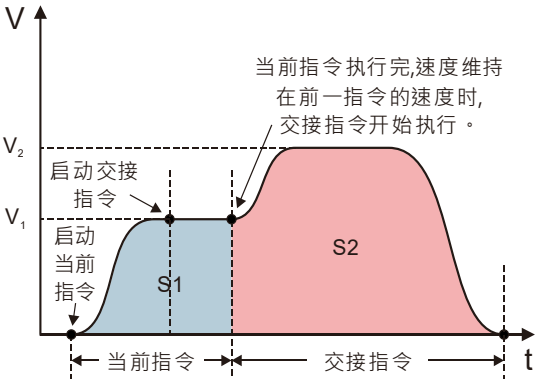


3. 当前指令和交接指令速度方向相反

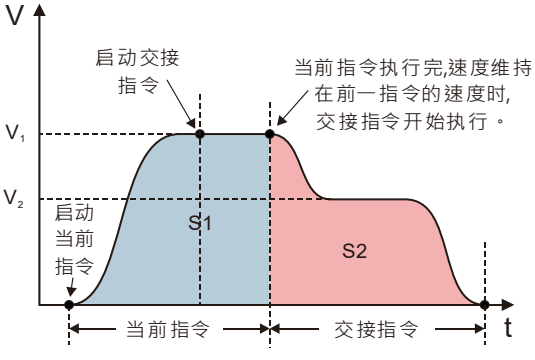


■ **以前一指令速度交接 Buffermode=mcBlendingPrevious，举例说明三种情况，分别如下：**

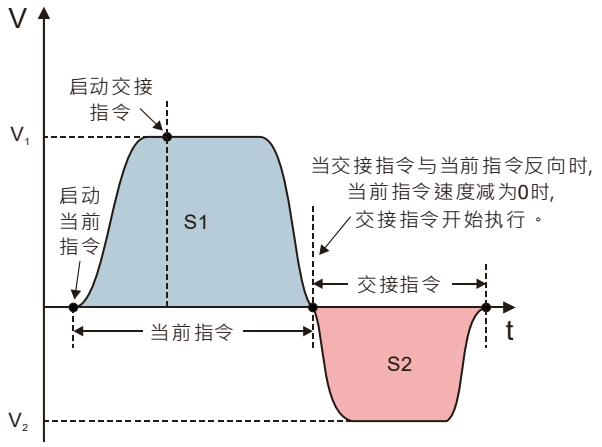
1. 当前指令比交接指令速度小



2. 当前指令比交接指令速度大

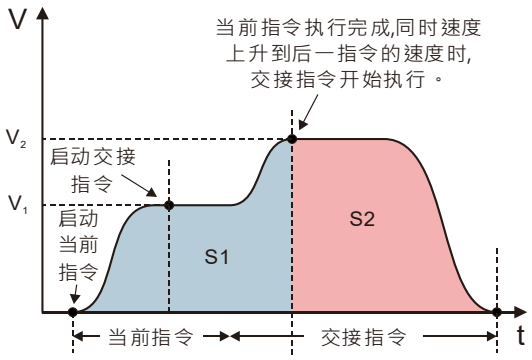


3. 当前指令和交接指令速度方向相反

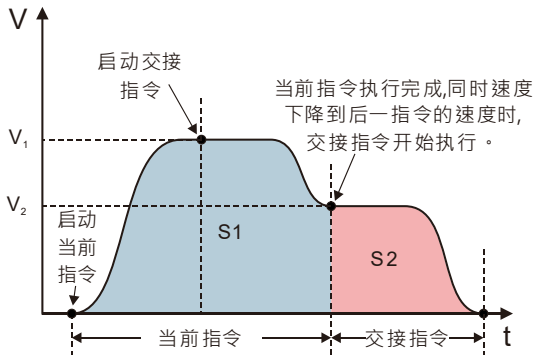


■ **以后一指令速度交接 Buffermode=mcBlendingNext，举例说明三种情况，分别如下：**

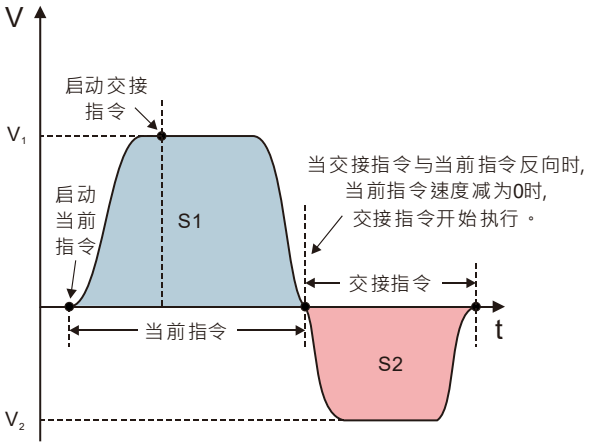
1. 当前指令比交接指令速度小



2. 当前指令比交接指令速度大

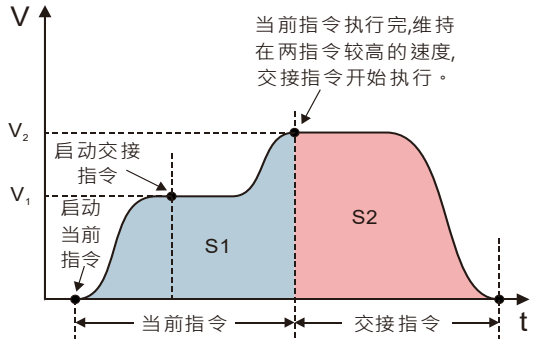


3. 当前指令和交接指令速度方向相反

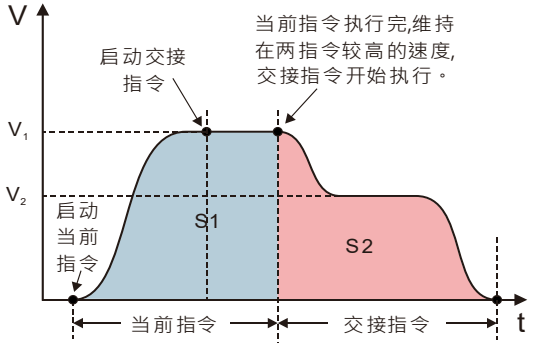


■ **以高速交接 Buffermode=mcBlendingHigh，举例说明三种情况，分别如下：**

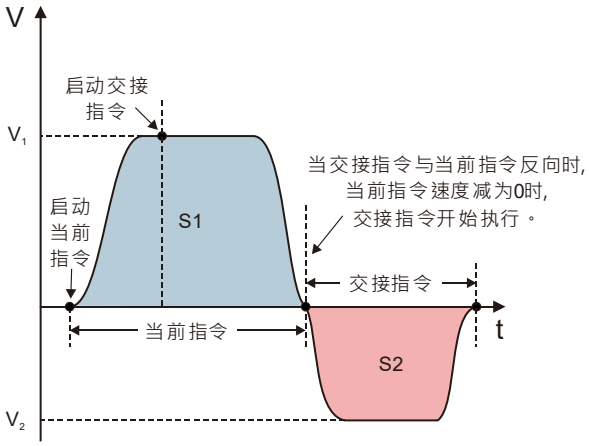
1. 当前指令比交接指令速度小



2. 当前指令比交接指令速度大



3. 当前指令和交接指令速度方向相反



● **各指令支持的交接模式**

当前指令和交接指令交接模式通过设置交接指令的 BufferMode 引脚参数来进行选择，交接指令的 BufferMode 引脚能够设置的值由当前指令所支持的交接模式进行选择，当前指令的 BufferMode 引脚参数是无效的。

例如：MC_MoveRelative 指令的交接模式支持 mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh，MC_MoveVelocity 指令的交接模式支持 mcAborting 和 mcBuffered。

情况1： 若 MC_MoveRelative 指令为当前指令，MC_MoveVelocity 指令为交接指令，则 MC_MoveVelocity 指令的 BufferMode 引脚可以选择的交接模式为 mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh。

情况2： 若 MC_MoveVelocity 指令为当前指令，MC_MoveRelative 指令为交接指令，则 MC_MoveRelative 指令的 BufferMode 引脚可以选择的交接模式为 mcAborting 和 mcBuffered。

根据当前指令，交接指令可以选择的 BufferMode（交接模式）列表如下：

当前指令	交接指令可以选择的 **BufferMode**（交接模式）
MC_MoveAbsolute	【mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh】*1
MC_MoveRelative	【mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh】*1
MC_MoveAdditive	【mcAborting、mcBuffered、mcBlendingLow、mcBlendingPrevious、mcBlendingNext、mcBlendingHigh】*1
MC_MoveSuperimposed	mcAborting
MC_HaltSuperimposed	mcAborting
MC_MoveVelocity	mcAborting、mcBuffered
MC_Home	只有 MC_Stop 指令可以打断 MC_Home 指令
MC_Halt	mcAborting、mcBuffered
MC_ GearIn	mcAborting、mcBuffered
MC_ GearOut	mcAborting、mcBuffered
MC_CombineAxes	mcAborting、mcBuffered
MC_ CamIn	mcAborting、mcBuffered
MC_ CamOut	mcAborting、mcBuffered

*1：交接指令为 MC_GearIn、MC_CamIn、MC_CombineAxes 时，BufferMode 只能选择 mcAborting 和 mcBuffered。

当前指令是否执行完成要看各个指令的完成位进行判断，完成位为 TRUE 表示该指令执行完成，并且开始执行 Bufffer 指令。下表即说明各个指令的完成位是哪一个，便于在选择 BufferMode 模式时进行判断。

指令名	是否是交接指令	是否可以跟随一个交接指令	完成位
MC_Home	否	否	Done
MC_Stop	否	否	Done
MC_Halt	是	是	Done
MC_MoveAbsolute	是	是	Done
MC_MoveRelative	是	是	Done
MC_MoveAdditive	是	是	Done
MC_MoveSuperimposed	否	否	——
MC_HaltSuperimposed	否	否	——
MC_MoveVelocity	是	是	InVelocity
MC_CamIn	是	是	EndOfProfile
MC_CamOut	否	是	Done
MC_GearIn	是	是	InGear
MC_GearOut	否	是	Done

指令名	是否是交接指令	是否可以跟随一个交接指令	完成位
MC_CombineAxes	是	是	InSync

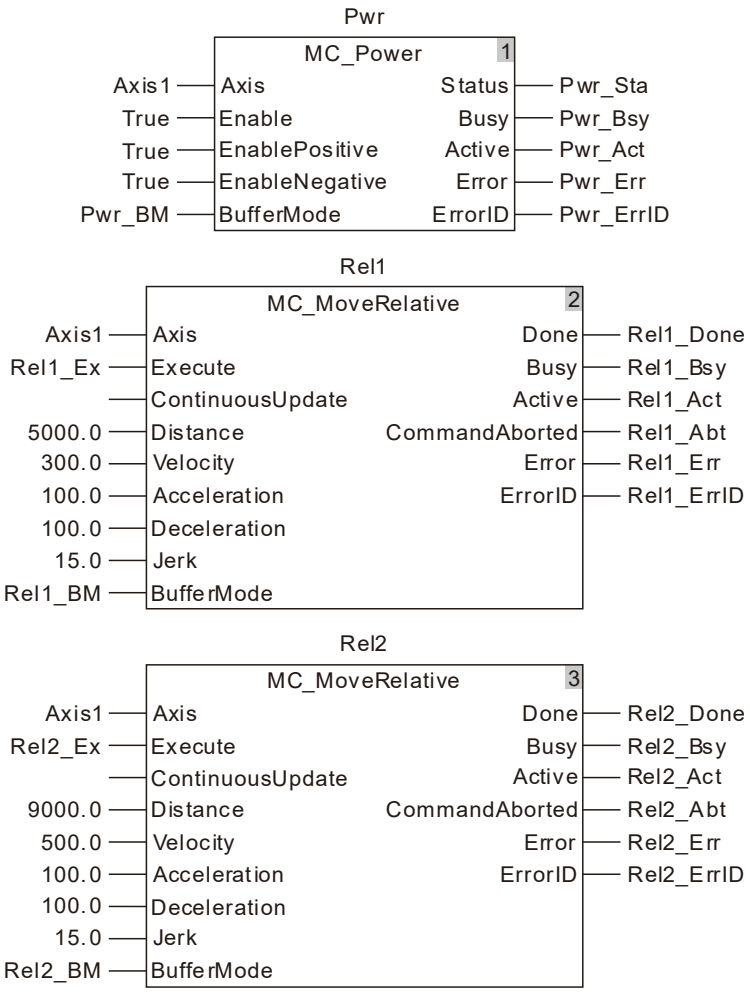
- **BufferMode（交接模式）范例**

程序范例一

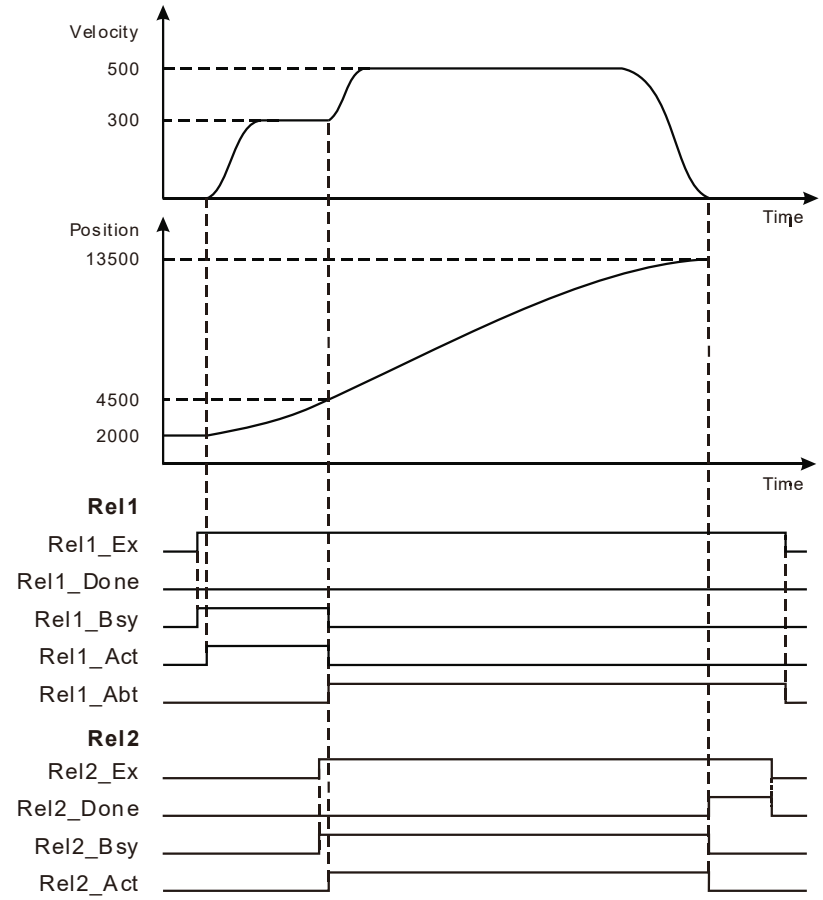
以 2 个 MC_MoveRelative 指令进行交接为例，分别介绍 6 种交接模式的范例：

变量与程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	FALSE
Rel1_BM	MC_Buffer_Mode	0
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	FALSE
Rel2_BM	MC_Buffer_Mode	
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	WORD	


Pwr
MC_Power
1
Axis1
Axis
Status
Pwr_Sta
True
Enable
Busy
Pwr_Bsy
True
EnablePositive
Active
Pwr_Act
True
EnableNegative
Error
Pwr_Err
Pwr_BM
BufferMode
ErrorID
Pwr_ErrID
Rel1
MC_MoveRelative
2
Axis1
Axis
Done
Rel1_Done
Rel1_Ex
Execute
Busy
Rel1_Bsy
ContinuousUpdate
Active
Rel1_Act
5000.0
Distance
CommandAborted
Rel1_Abt
300.0
Velocity
Error
Rel1_Err
100.0
Acceleration
ErrorID
Rel1_ErrID
100.0
Deceleration
15.0
Jerk
Rel1_BM
BufferMode
Rel2
MC_MoveRelative
3
Axis1
Axis
Done
Rel2_Done
Rel2_Ex
Execute
Busy
Rel2_Bsy
ContinuousUpdate
Active
Rel2_Act
9000.0
Distance
CommandAborted
Rel2_Abt
500.0
Velocity
Error
Rel2_Err
100.0
Acceleration
ErrorID
Rel2_ErrID
100.0
Deceleration
15.0
Jerk
Rel2_BM
BufferMode

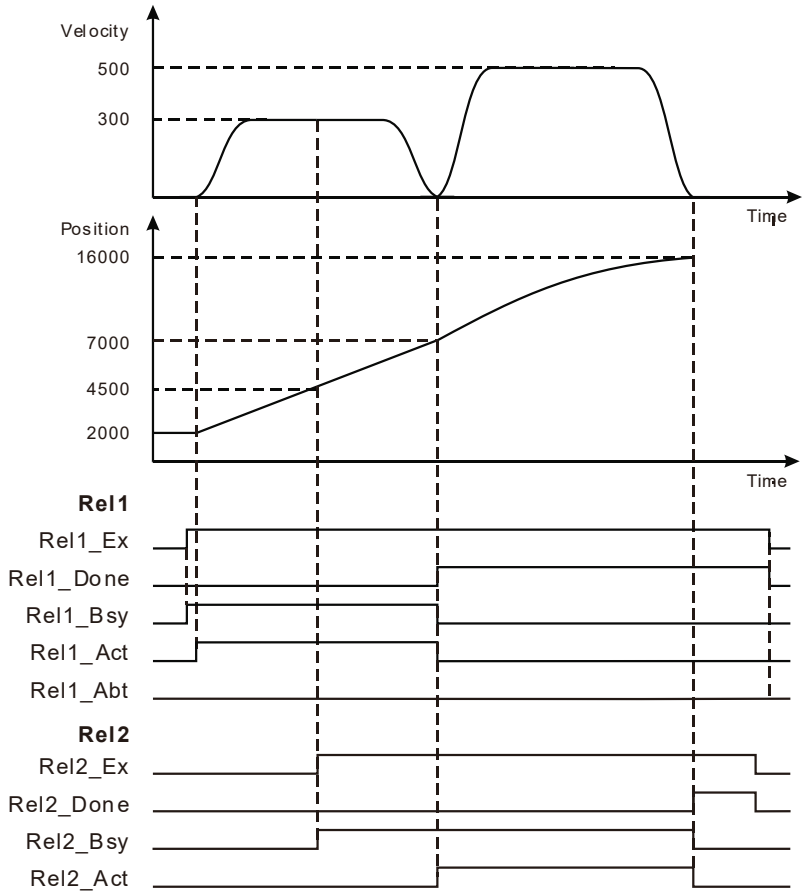
■ **Rel2_BM=mcAborting**



❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时，同时 Rel1_Bsy 变为 TRUE，且一个周期后，Rel1_Act 变为 TRUE，开始执行第一个相对位移指令；当还未到达目标位置时，Rel2_Ex 由 FALSE 变为 TRUE，同时 Rel2_Bsy 变为 TRUE，且一个周期后，Rel1_Abt、Rel2_Act 变为 TRUE，Rel1_Bsy、Rel1_Act 变为 FALSE，同时打断第一个相对位移指令，开始执行第二个相对位移指令；当目标位置到达时，Rel2_Done 变为 TRUE，同时 Rel2_Bsy 和 Rel2_Act 变为 FALSE。

❖ 当 Rel2_Ex 由 TRUE 变为 FALSE 时，Rel2_Done 变为 FALSE。

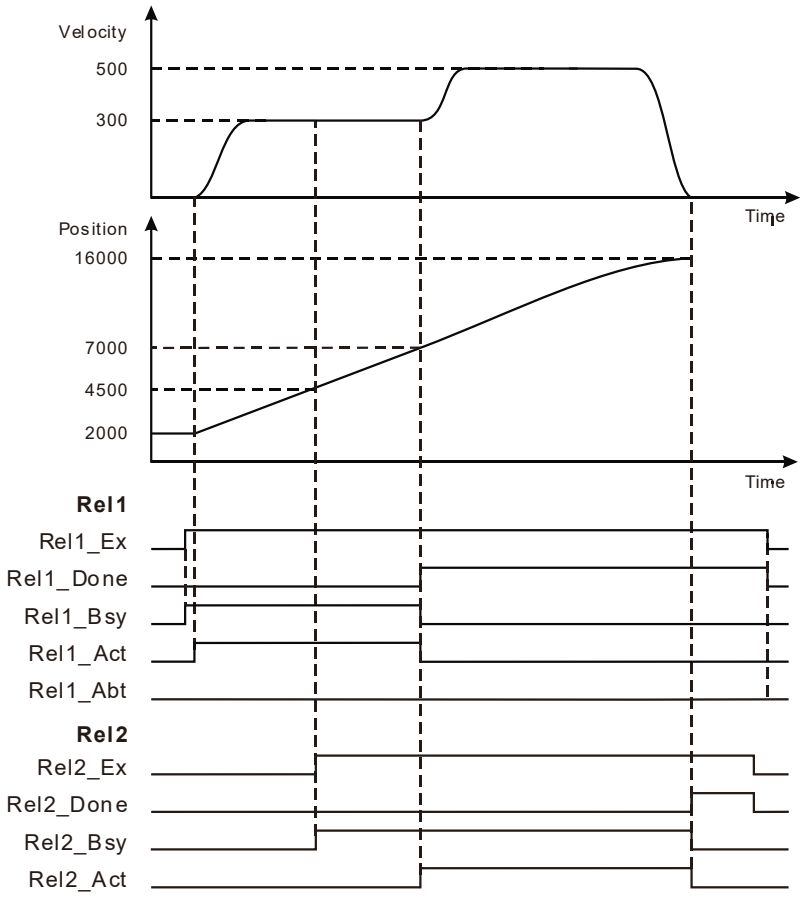
■ **Rel2_BM =mcBuffered**



❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时，同时 Rel1_Bsy 变为 TRUE，且一个周期后，Rel1_Act 变为 TRUE，开始执行第一个相对位移指令；当还未到达目标位置时，Rel2_Ex 由 FALSE 变为 TRUE， 同时 Rel2_Bsy 变为 TRUE，Rel1_Bsy、 Rel1_Act 依然保持为 TRUE，第一个相对位移指令继续执行，当目标位置到达时，Rel1_Done 变为 TRUE，同时 Rel1_Bsy、Rel1_Act 变为 FALSE，Rel2_Act 变为 TRUE，并立即开始执行第二个相对位移指令；当目标位置到达时，Rel2_Done 变为 TRUE，同时 Rel2_Bsy 和 Rel2_Act 变为 FALSE。

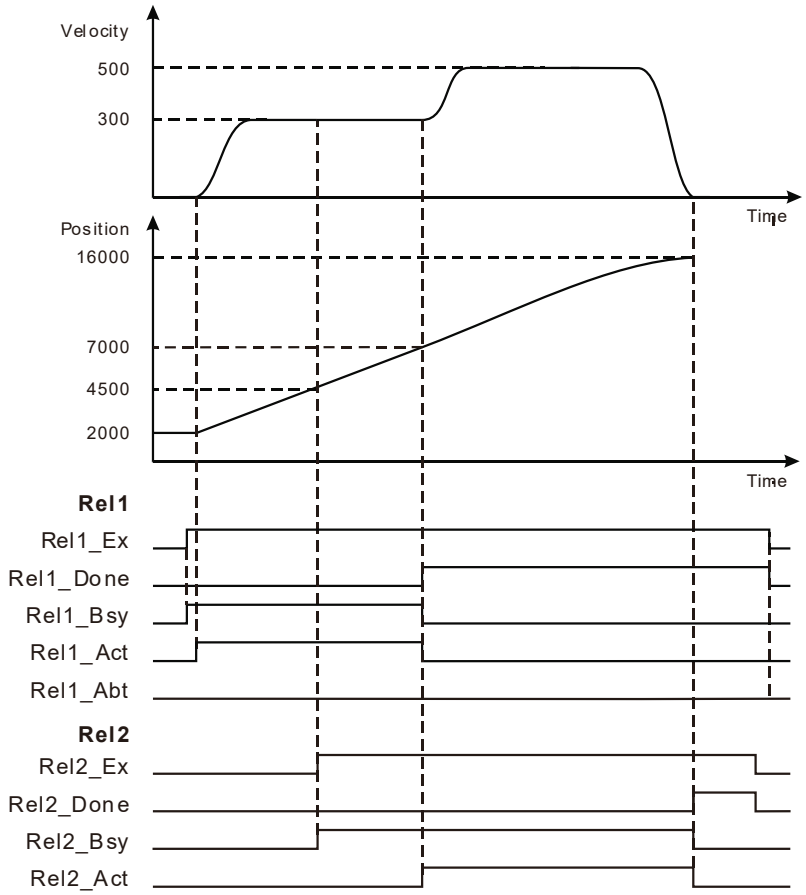
❖ 当 Rel1_Ex 由 TRUE 变为 FALSE 时， 同时 Rel1_Done 变为 FALSE；当 Rel2_Ex 由 TRUE 变为 FALSE 时，Rel2_Done 变为 FALSE。

■ **Rel2_BM =mcBlendingLow**



- ❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时，同时 Rel1_Bsy 变为 TRUE，且一个周期后，Rel1_Act 变为 TRUE，开始执行第一个相对位移指令；当还未到达目标位置时，Rel2_Ex 由 FALSE 变为 TRUE， 同时 Rel2_Bsy 变为 TRUE，Rel1_Bsy、Rel1_Act 依然保持为 TRUE，第一个相对位移指令继续执行，当目标位置到达时，Rel1_Done 变为 TRUE，此时速度为 300 单元/秒,是当前指令与交接指令中较低的目标速度，同时 Rel1_Bsy、Rel1_Act 变为 FALSE，Rel2_Act 变为 TRUE，并立即开始执行第二个相对位移指令；当目标位置到达时，Rel2_Done 变为 TRUE，同时 Rel2_Bsy 和 Rel2_Act 变为 FALSE。
- ❖ 当 Rel1_Ex 由 TRUE 变为 FALSE 时， 同时 Rel1_Done 变为 FALSE；当 Rel2_Ex 由 TRUE 变为 FALSE 时，Rel2_Done 变为 FALSE。

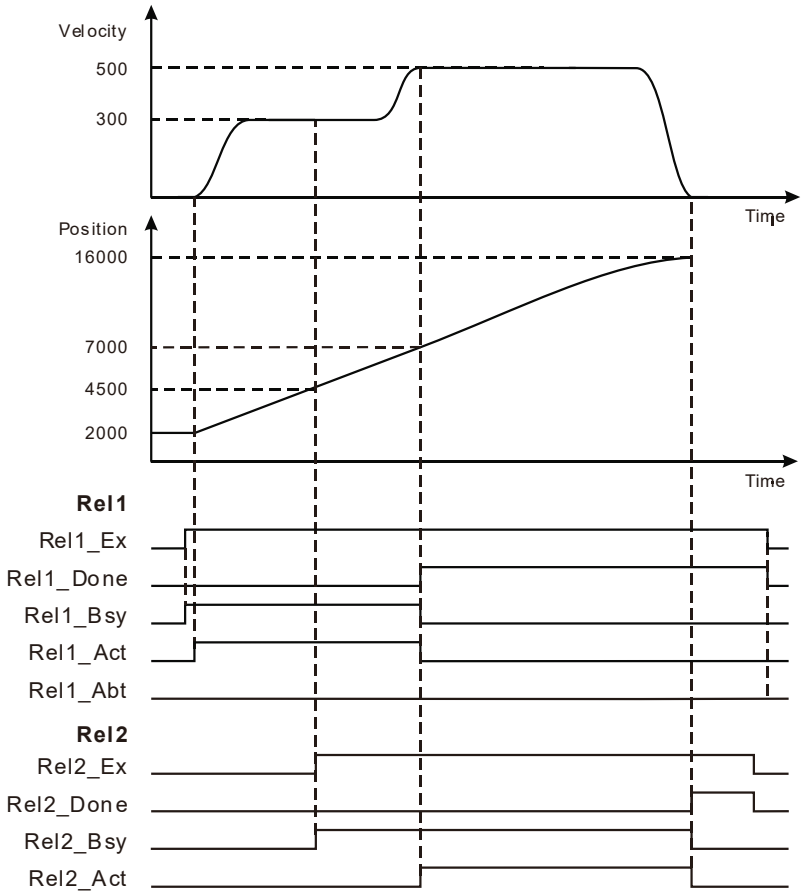
■ **Rel2_BM =mcBlendingPrevious**



❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时，同时 Rel1_Bsy 变为 TRUE，且一个周期后，Rel1_Act 变为 TRUE，开始执行第一个相对位移指令；当还未到达目标位置时，Rel2_Ex 由 FALSE 变为 TRUE， 同时 Rel2_Bsy 变为 TRUE，Rel1_Bsy、 Rel1_Act 依然保持为 TRUE，第一个相对位移指令继续执行，当目标位置到达时，Rel1_Done 变为 TRUE，此时速度为 300 单元/秒，是当前指令的目标速度)，同时 Rel1_Bsy、Rel1_Act 变为 FALSE，Rel2_Act 变为 TRUE，并立即开始执行第二个相对位移指令；当目标位置到达时，Rel2_Done 变为 TRUE，同时 Rel2_Bsy 和 Rel1_Act 变为 FALSE。

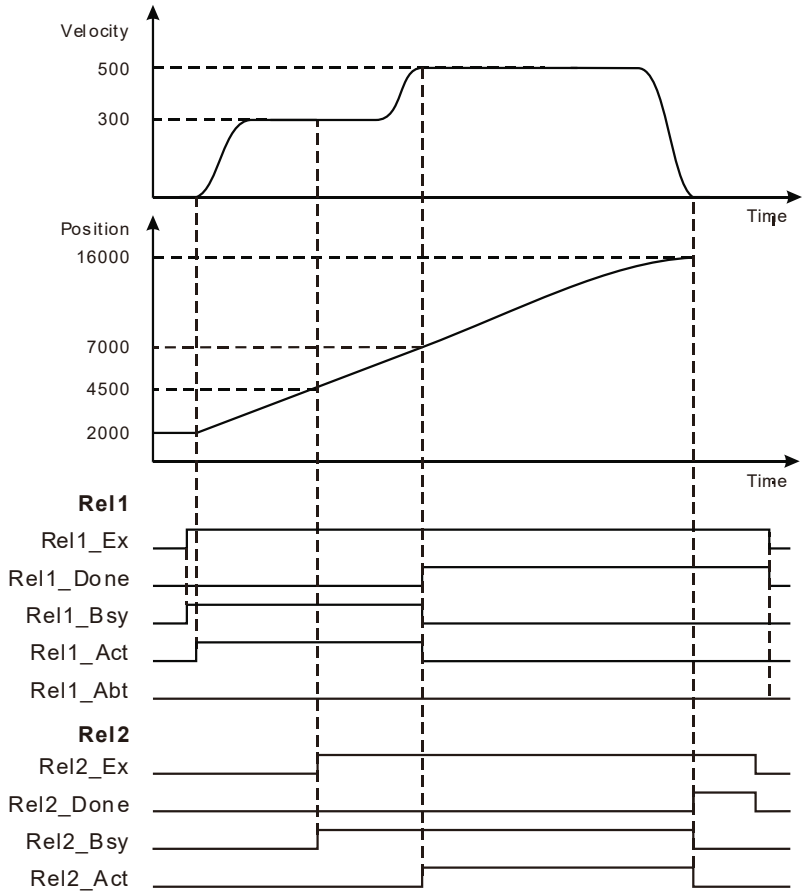
❖ 当 Rel1_Ex 由 TRUE 变为 FALSE 时， 同时 Rel1_Done 变为 FALSE；当 Rel2_Ex 由 TRUE 变为 FALSE 时，Rel2_Done 变为 FALSE。

■ **Rel2_BM =mcBlendingNext**



- ❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时，同时 Rel1_Bsy 变为 TRUE，且一个周期后，Rel1_Act 变为 TRUE，开始执行第一个相对位移指令；当还未到达目标位置时，Rel2_Ex 由 FALSE 变为 TRUE， 同时 Rel2_Bsy 变为 TRUE，Rel1_Bsy、Rel1_Act 依然保持为 TRUE，第一个相对位移指令继续执行，当目标位置到达时，Rel1_Done 变为 TRUE，此时速度为 500 单元/秒 (交接指令的目标速度)，同时 Rel1_Bsy、Rel1_Act 变为 FALSE，Rel2_Act 变为 TRUE，并开始执行第二个相对位移指令；当目标位置到达时，Rel2_Done 变为 TRUE，同时 Rel2_Bsy 和 Rel2_Act 变为 FALSE。
- ❖ 当 Rel1_Ex 由 TRUE 变为 FALSE 时， 同时 Rel1_Done 变为 FALSE；当 Rel2_Ex 由 TRUE 变为 FALSE 时， Rel2_Done 变为 FALSE。

■ **Rel2_BM =mcBlendingHigh**



- ❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时，同时 Rel1_Bsy 变为 TRUE，且一个周期后，Rel1_Act 变为 TRUE，开始执行第一个相对位移指令；当还未到达目标位置时，Rel2_Ex 由 FALSE 变为 TRUE， 同时 Rel2_Bsy 变为 TRUE，Rel1_Bsy、Rel1_Act 依然保持为 TRUE，第一个相对位移指令继续执行，当目标位置到达时，Rel1_Done 变为 TRUE，此时速度为 500 单元/秒 (当前指令与交接指令中较高的目标速度)，同时 Rel1_Bsy、Rel1_Act 变为 FALSE，Rel2_Act 变为 TRUE，并开始执行第二个相对位移指令；当目标位置到达时，Rel2_Done 变为 TRUE，同时 Rel2_Bsy 和 Rel2_Act 变为 FALSE。
- ❖ 当 Rel1_Ex 由 TRUE 变为 FALSE 时， 同时 Rel1_Done 变为 FALSE；当 Rel2_Ex 由 TRUE 变为 FALSE 时， Rel2_Done 变为 FALSE。

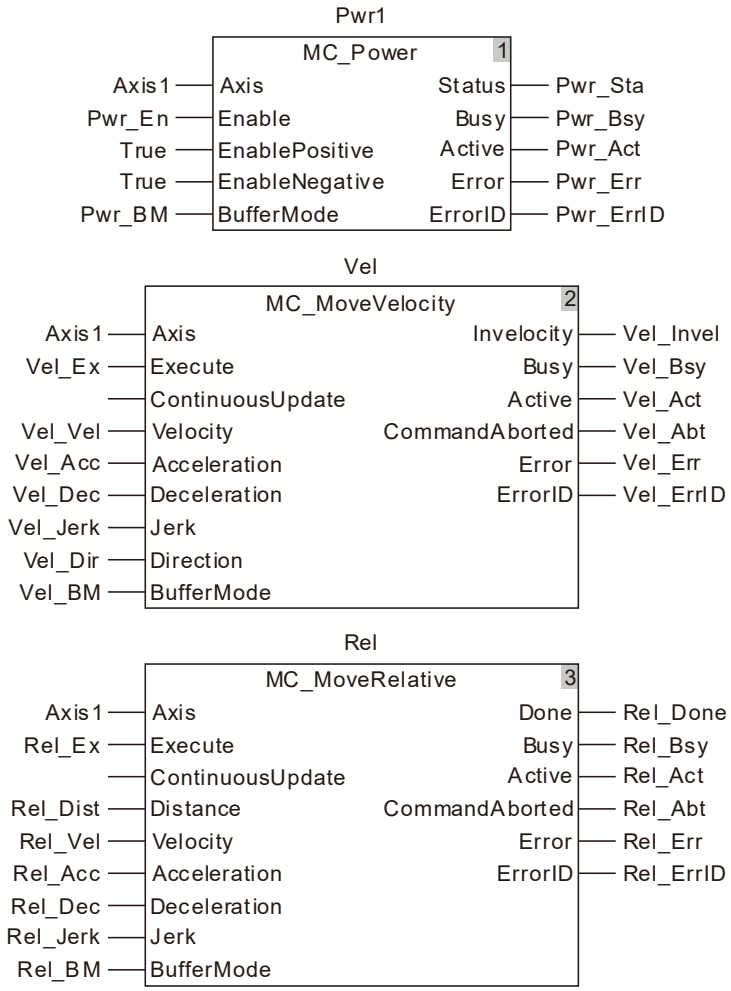
程序范例二

以一个速度指令和一个相对位移指令作为例子讲解不同 BufferMode 状态下，轴的运动状态。

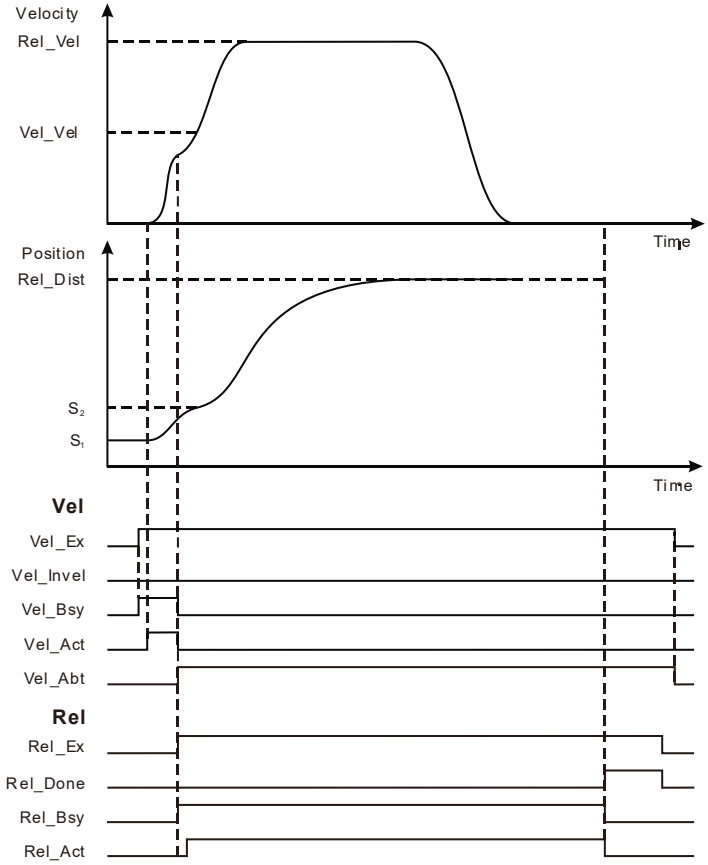
变量与程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE

变量名	数据类型	初始值
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel _Ex	BOOL	FALSE
Vel _Vel	LREAL	10000.0
Vel _Acc	LREAL	10000.0
Vel _Dec	LREAL	10000.0
Vel _Jerk	LREAL	10000.0
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_Dist	LREAL	100000.0
Rel_Vel	LREAL	20000.0
Rel_Acc	LREAL	10000.0
Rel_Dec	LREAL	10000.0
Rel_Jerk	LREAL	10000.0
Rel_BM	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	

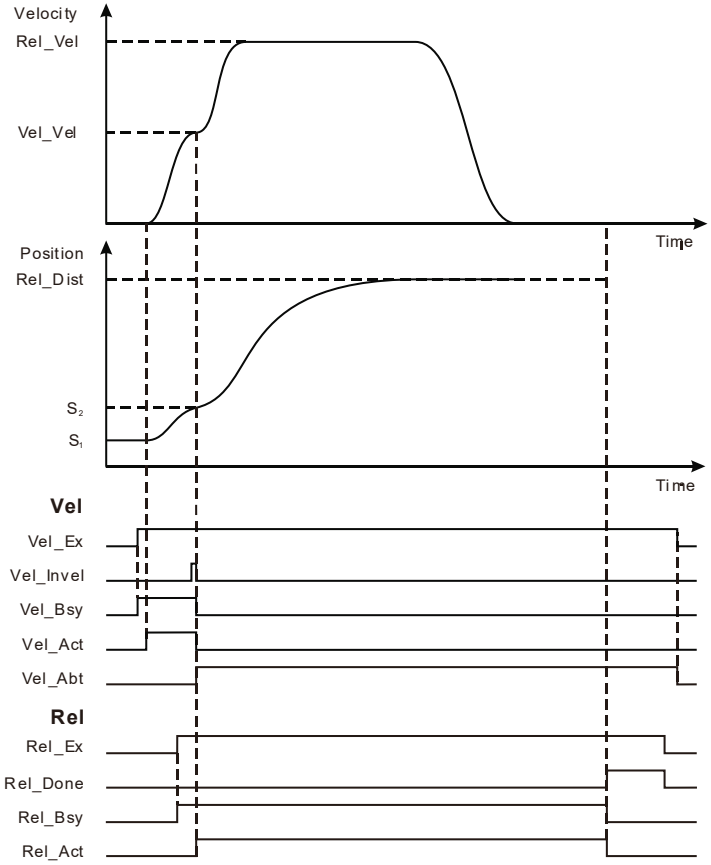

Pwr1
MC_Power
1
Axis1
Axis
Status
Pwr_Sta
Pwr_En
Enable
Busy
Pwr_Bsy
True
EnablePositive
Active
Pwr_Act
True
EnableNegative
Error
Pwr_Err
Pwr_BM
BufferMode
ErrorID
Pwr_ErrID
Vel
MC_MoveVelocity
2
Axis1
Axis
Invelocity
Vel_Invel
Vel_Ex
Execute
Busy
Vel_Bsy
ContinuousUpdate
Active
Vel_Act
Vel_Vel
Velocity
CommandAborted
Vel_Abt
Vel_Acc
Acceleration
Error
Vel_Err
Vel_Dec
Deceleration
ErrorID
Vel_ErrID
Vel_Jerk
Jerk
Vel_Dir
Direction
Vel_BM
BufferMode
Rel
MC_MoveRelative
3
Axis1
Axis
Done
Rel_Done
Rel_Ex
Execute
Busy
Rel_Bsy
ContinuousUpdate
Active
Rel_Act
Rel_Dist
Distance
CommandAborted
Rel_Abt
Rel_Vel
Velocity
Error
Rel_Err
Rel_Acc
Acceleration
ErrorID
Rel_ErrID
Rel_Dec
Deceleration
Rel_Jerk
Jerk
Rel_BM
BufferMode

■ **Rel_BM =mcAborting**



❖ Vel_Ex 由 FALSE 变为 TRUE，Vel_Bsy 为 TRUE，Vel_Act 在一个周期后 TRUE，当速度还未到达时，Rel_Ex 由 FASLE 变为 TRUE，轴立即按照相对位移指令的速度加速度的要求进行运动，Vel_Abt 为 TRUE，Vel_Bsy 和 Vel_Act 同时 FALSE，速度指令被打断，位移指令执行，Rel_Bsy 为 TRUE，Rel_Act 一个周期后 TRUE，当位移完成时，Rel_Done 为 TRUE。

■ **Rel_BM =mcBuffered**



❖ Vel_Ex 由 FASLE 变为 TRUE，Vel_Bsy 为 TRUE，Vel_Act 在一个周期后 TRUE，当速度还未到达时，Rel_Ex 由 FASLE 变为 TRUE，轴等待速度指令执行完成后才会执行相对位移指令，此时，Rel_Bsy 为 TRUE。待速度指令执行完毕，Vel_Invel 为 TRUE 后一个周期，相对位移指令开始执行，Vel_Abt 为 TRUE，表示指令被打断。Rel_Act 为 TRUE，表示相对位移指令开始控制轴运动。等待位移完成后，Rel_Done 为 TRUE。

■ **（Rel _BM =**mcBlendingLow **或** mcBlendingPrevious **或** mcBlendingNext **或** mcBlendingHigh **与** mcBuffered **的效果一致。）**

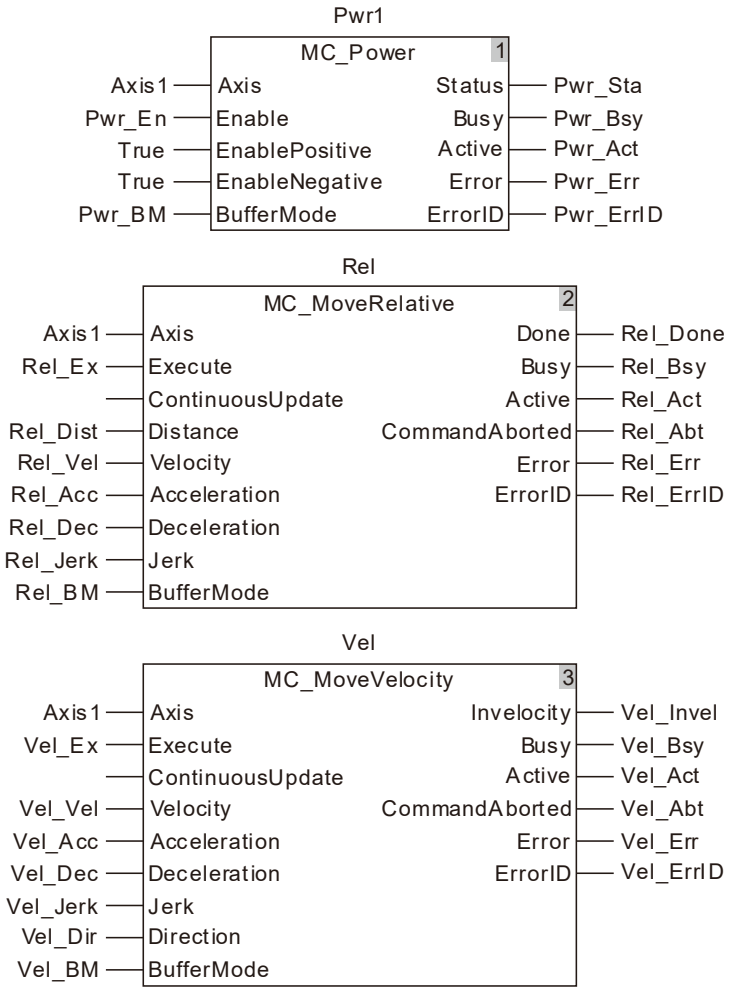
程序范例三

以一个相对位移指令和速度指令为例子讲解不同 BufferMode 状态下，轴的运动状态

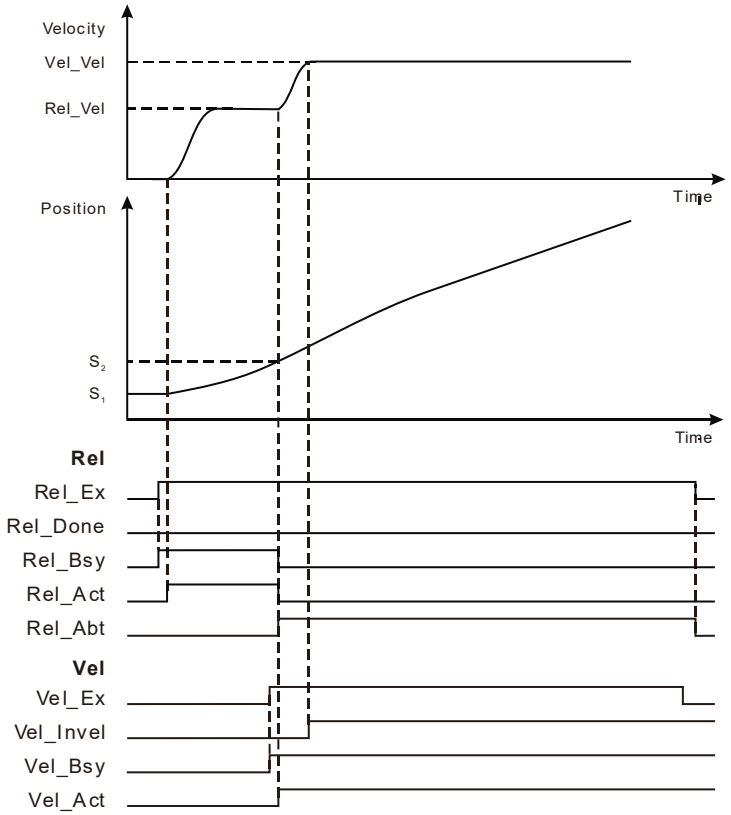
变量与程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	

变量名	数据类型	初始值
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_Dist	LREAL	100000.0
Rel_Vel	LREAL	10000.0
Rel_Acc	LREAL	10000.0
Rel_Dec	LREAL	10000.0
Rel_Jerk	LREAL	10000.0
Rel_BM	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel _Ex	BOOL	FALSE
Vel _Vel	LREAL	20000.0
Vel _Acc	LREAL	10000.0
Vel _Dec	LREAL	10000.0
Vel _Jerk	LREAL	10000.0
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ErrID	WORD	

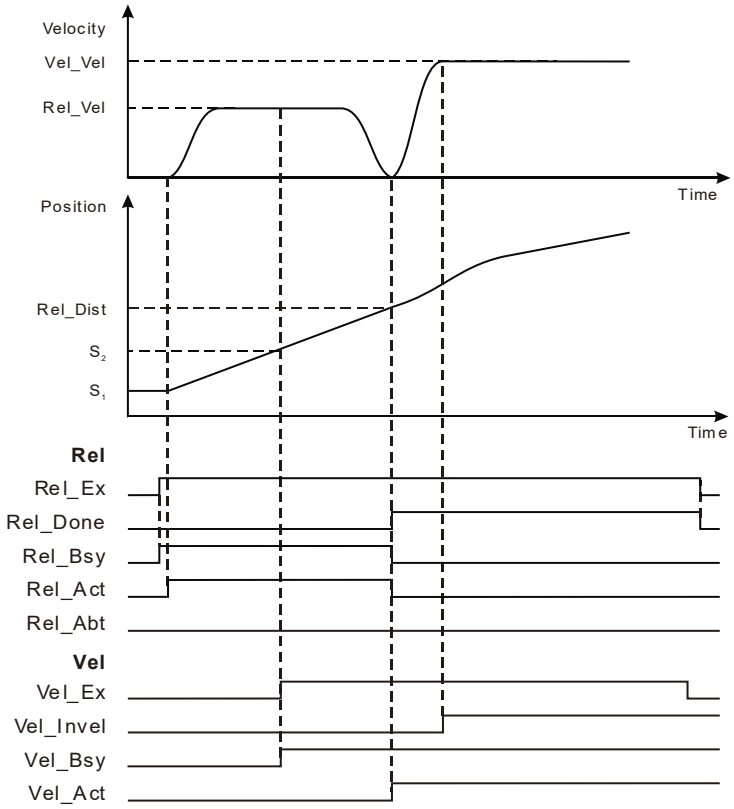

Pwr1
MC_Power
1
Axis1
Axis
Status
Pwr_Sta
Pwr_En
Enable
Busy
Pwr_Bsy
True
EnablePositive
Active
Pwr_Act
True
EnableNegative
Error
Pwr_Err
Pwr_BM
BufferMode
ErrorID
Pwr_ErrID
Rel
MC_MoveRelative
2
Axis1
Axis
Done
Rel_Done
Rel_Ex
Execute
Busy
Rel_Bsy
ContinuousUpdate
Active
Rel_Act
Rel_Dist
Distance
CommandAborted
Rel_Abt
Rel_Vel
Velocity
Error
Rel_Err
Rel_Acc
Acceleration
ErrorID
Rel_ErrID
Rel_Dec
Deceleration
Rel_Jerk
Jerk
Rel_BM
BufferMode
Vel
MC_MoveVelocity
3
Axis1
Axis
Invelocity
Vel_Invel
Vel_Ex
Execute
Busy
Vel_Bsy
ContinuousUpdate
Active
Vel_Act
Vel_Vel
Velocity
CommandAborted
Vel_Abt
Vel_Acc
Acceleration
Error
Vel_Err
Vel_Dec
Deceleration
ErrorID
Vel_ErrID
Vel_Jerk
Jerk
Vel_Dir
Direction
Vel_BM
BufferMode

■ **Vel _BM =mcAborting**



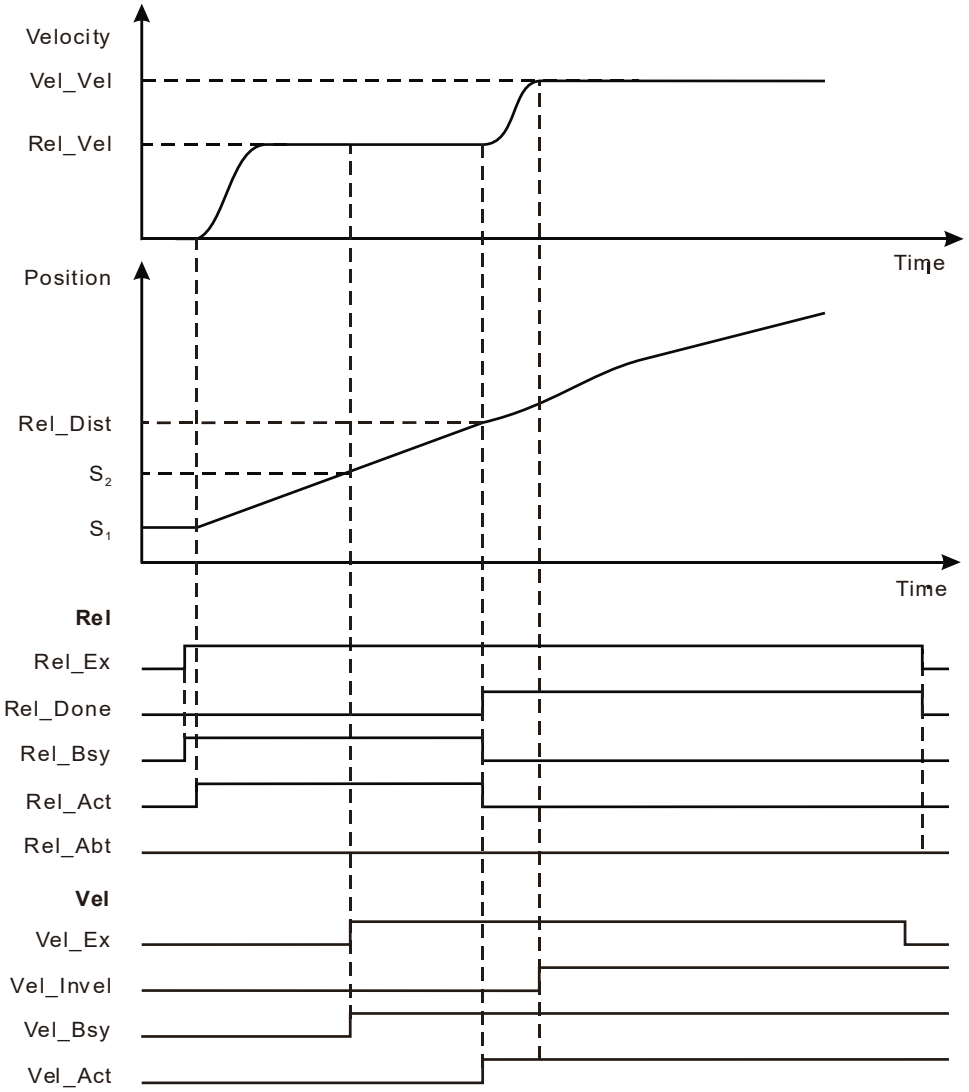
❖ Rel_Ex 由 FALSE 变为 TRUE，Rel_Bsy 为 TRUE，Rel_Act 在一个周期后 TRUE，当位移量还未到达时，Vel_Ex 由 FASLE 变为 TRUE，轴立即按照速度指令的速度加速度的要求进行运动，Rel_Abt 为 TRUE，Rel_Bsy 和 Rel_Act 同时 FALSE，相对位移指令被打断，速度指令执行，Vel_Bsy 为 TRUE，Vel_Act 一个周期后 TRUE，当速度到达时，Vel_Invel 为 TRUE。

■ **Vel _BM =mcBuffered**



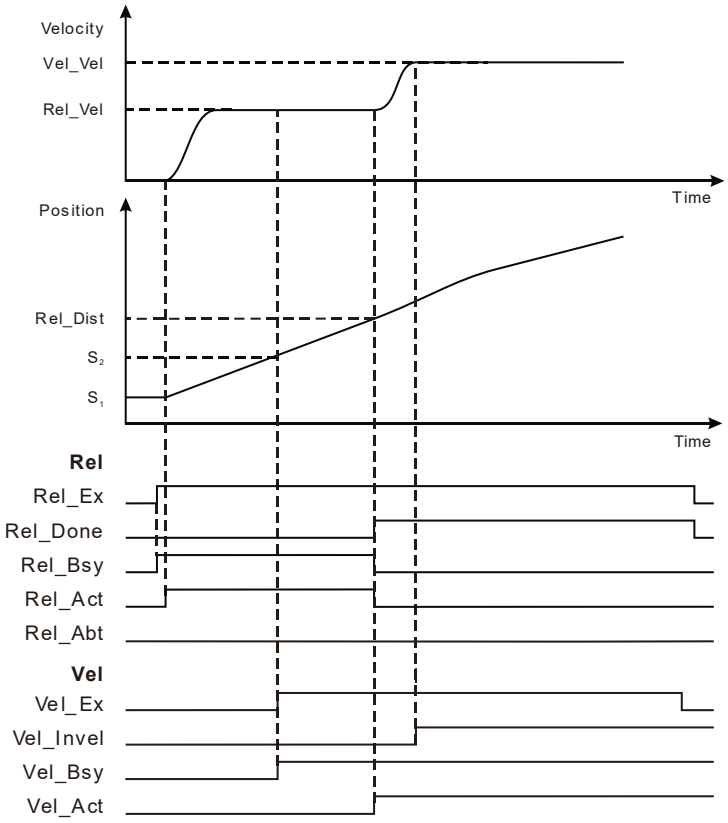
❖ Rel_Ex 由 FALSE 变为 TRUE，Rel_Bsy 为 TRUE，Rel_Act 在一个周期后 TRUE，当位移量还未到达时，Vel_Ex 由 FASLE 变为 TRUE，轴等待相对位移指令执行完成后，速度降为 0，Rel_Done 为 TRUE，Rel_Bsy 为 FALSE，Rel_Act 为 FALSE。然后按照速度指令的速度加速度的要求进行运动， Vel_.Bsy 为 TRUE，Vel_Act 一个周期后 TRUE，当速度到达时，Rel_Invel 为 TRUE。

■ **Vel _BM =mcBlendingLow**



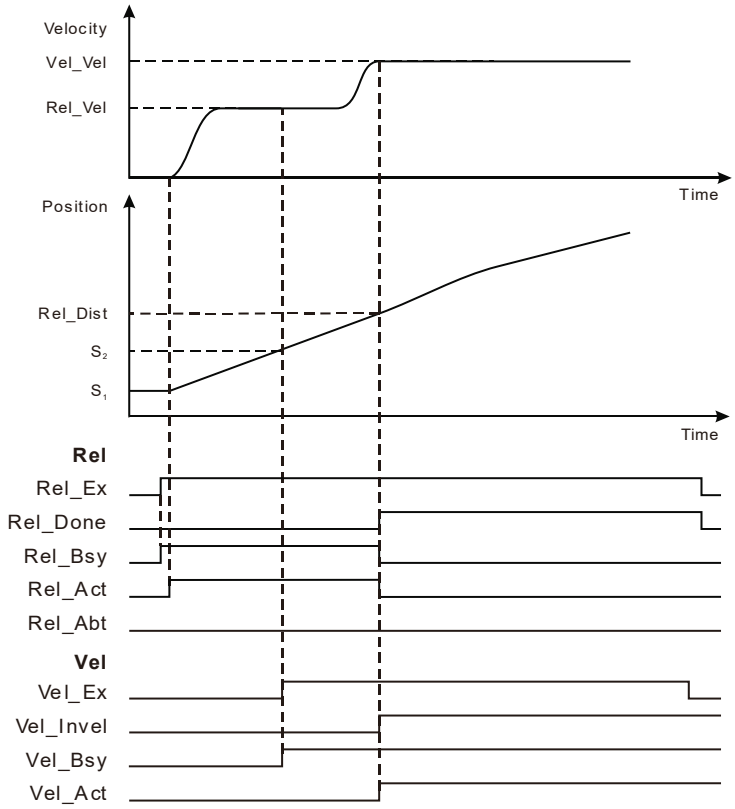
❖ Rel_Ex 由 FALSE 变为 TRUE，Rel_Bsy 为 TRUE，Rel_Act 在一个周期后 TRUE，当位移量还未到达时，Vel_Ex 由 FASLE 变为 TRUE，Vel_Bsy 为 TRUE，轴等待相对位移指令执行完成后，Rel_Done 为 TRUE，Rel_Bsy 为 FALSE，Rel_Act 为 FALSE，同时 Vel_Act 为 TRUE,此时速度为 10000 单元/秒，(当前指令与交接指令中较低的目标速度)，位移指令完成后，开始执行速度指令，。当速度到达设定的目标速度时，Vel_Invel 为 TRUE。

■ **Vel _BM =mcBlendingPrevious**



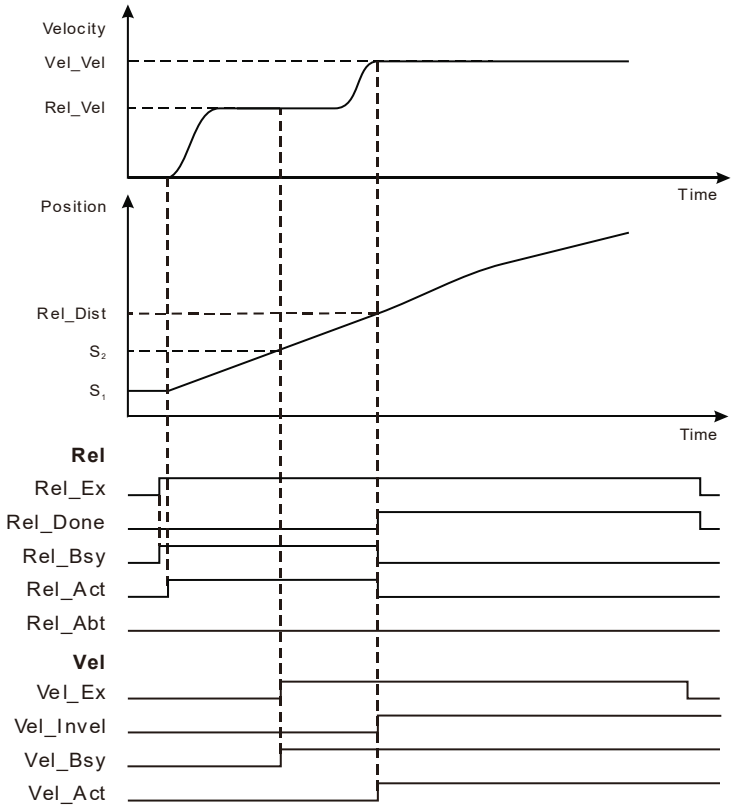
❖ Rel_Ex 由 FALSE 变为 TRUE，Rel_Bsy 为 TRUE，Rel_Act 在一个周期后 TRUE，当位移量还未到达时， Vel_Ex 由 FASLE 变为 TRUE，Vel_Bsy 为 TRUE，，轴等待相对位移指令执行完成后， Rel_Done 为 TRUE，Rel_Bsy 为 FALSE，Rel_Act 为 FALSE，同时 Vel_Act 为 TRUE，此时速度为 10000 单元/秒, (当前指令的目标速度)。当速度到达时，Vel_Invel 为 TRUE。

■ **Vel _BM =mcBlendingNext**



❖ Rel_Ex 由 FALSE 变为 TRUE，Rel_Bsy 为 TRUE，Rel_Act 在一个周期后 TRUE，当位移量还未到达时， Vel_Ex 由 FASLE 变为 TRUE，同时 Vel_Bsy 为 TRUE，轴等待相对位移指令执行完成后， Rel_Done 为 TRUE，Rel_Busy 和 Rel_Active 都为 FALSE。同时 Vel_Act 为 TRUE，此时速度为 20000 单元/秒（交接指令的目标速度）。当速度到达时，Vel_Invel 为 TRUE。

■ **Vel _BM =mcBlendingHigh**

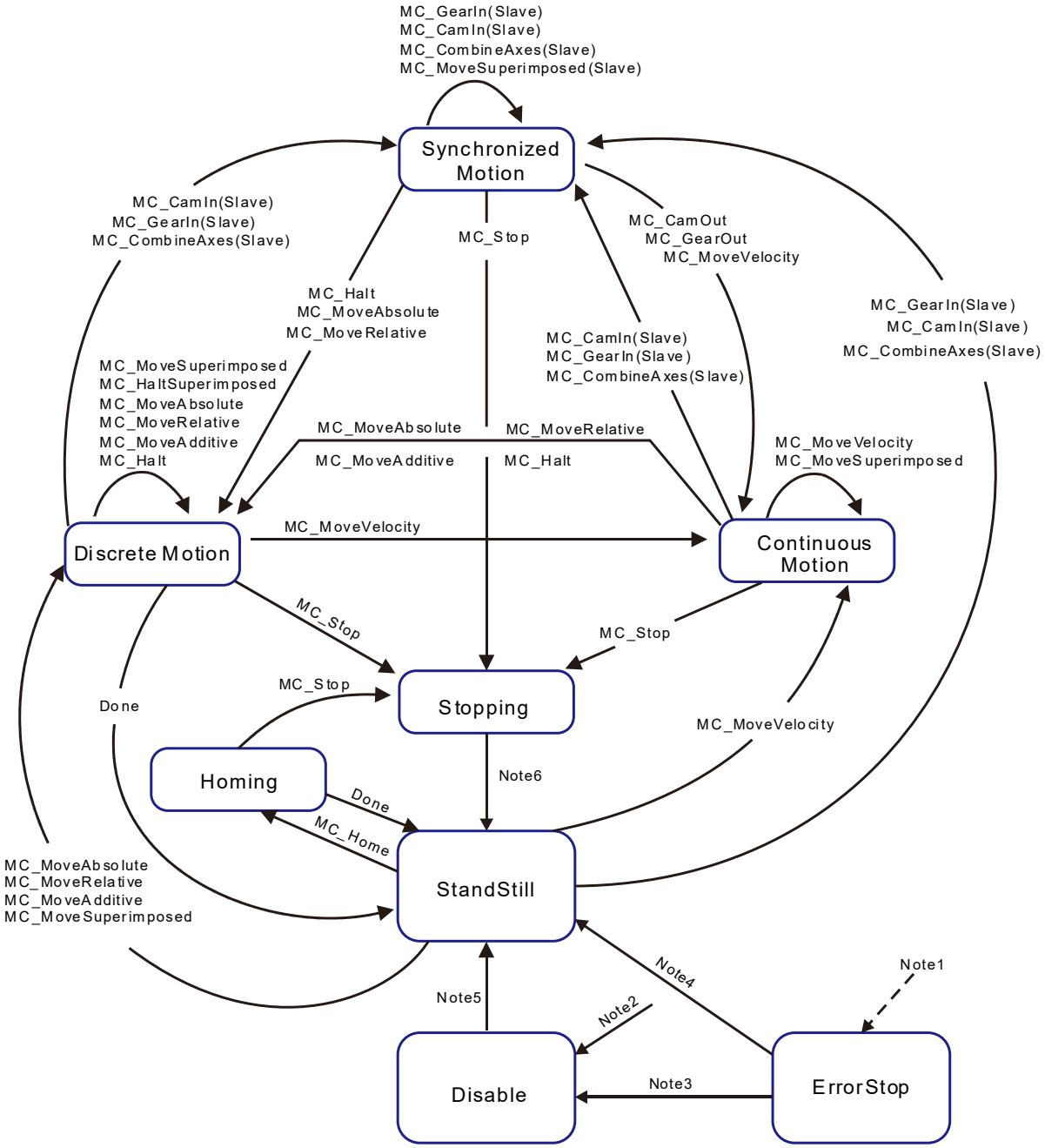


❖ Rel_Ex 由 FALSE 变为 TRUE，Rel_Bsy 为 TRUE，Rel_Act 在一个周期后 TRUE，当位移量还未到达时， Vel_Ex 由 FASLE 变为 TRUE，同时 Vel_Bsy 为 TRUE，轴等待相对位移指令位移执行完成后， Rel_Done 为 TRUE，Rel_Bsy 为 FALSE，Rel_Act 为 FALSE 此时速度为 20000 单元/秒 (当前指令与交接指令中较高的目标速度)。然后按照速度指令的速度加减速度的要求进行运动，当速度到达时，Vel_Invel 为 TRUE。

10.4 状态机

DVP-15MC 系列运动控制器在使用运动控制指令控制每个轴时，每个轴都有一个内部运行状态，轴的状态切换遵循下图所示的状态机。状态机定义了各个状态下可以执行的运动指令及执行运动指令后的状态，用户在使用运动指令时，可以通过状态机判断某一运动指令在当前状态下是否可以使用。

DVP-15MC 系列运动控制器的状态机如下图所示，箭头指示部分为轴的状态。



Note1：在任何状态下，轴发生错误，都会进入 ErrorStop 状态。

Note2：在任何状态下，轴没有发生错误，并且 MC_Power.Enable 为 False(MC_Power 指令 Buffermode 选择 mcAborting 时)。

Note3：MC_Power.Status 为 False 时使用 MC_Reset 指令复位。

Note4：MC_Power.Enable 和 MC_Power.Status 为 True 时使用 MC_Reset 指令复位。

Note5：使用 MC_Power 指令使能并且 MC_Power. Status 为 True。

Note6：MC_Stop.Done 为 True 并且 MC_Stop.Execute 为 False。

序号	轴状态	说明
1	StandStill	准备执行状态
2	Disable	未执行状态
3	ErrorStop	错误状态
4	Stopping	停止状态
5	Homing	原点回归状态
6	Discrete Motion	离散运动状态
7	Continuous Motion	连续运动状态
8	Synchronized Motion	同步运动状态

轴的状态可以根据 MC_ReadStatus（读取轴状态指令）的输出引脚来进行判断，指令具体使用请参考 11.3.17 节。

MEMO

11

第11章　运动指令

目录

11.1 运动指令一览表

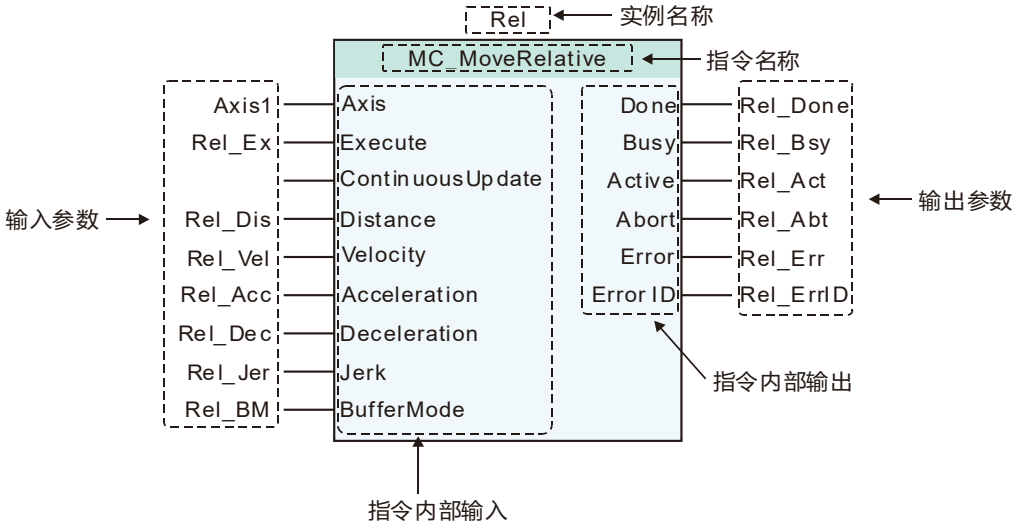
指令组	指令码	功能	页码
单轴指令	MC_Power	使能指令	11-8
	MC_Home	原点回归指令	11-17
	MC_MoveVelocity	速度指令	11-21
	MC_Halt	暂停指令	11-28
	MC_Stop	停止指令	11-33
	MC_MoveRelative	相对位移指令	11-38
	MC_MoveAdditive	附加位移指令	11-46
	MC_MoveAbsolute	绝对位移指令	11-54
	MC_MoveSuperimposed	追加位移指令	11-63
	MC_HaltSuperimposed	暂停追加指令	11-70
	MC_SetPosition	位置设置指令	11-75
	MC_SetOverride	超调值设置指令	11-85
	MC_Reset	复位指令	11-89
	DMC_SetTorque	扭矩设定指令	11-92
	MC_ReadAxisError	读取轴错误指令	11-96
	MC_ReadActualPosition	实际位置读取指令	11-98
	MC_ReadStatus	读取轴状态指令	11-103
	MC_ReadMotionState	读取轴运动状态指令	11-109
	DMC_ReadParameter_Motion	读取参数指令	11-114
	DMC_WriteParameter_Motion	写入参数指令	11-119
	DMC_TouchProbe	位置捕获指令	11-124
	DMC_TouchProbeCyclically	循环位置捕获指令	11-133
	DMC_Jog	点动指令	11-134
	DMC_MoveVelocityStopByPos	速度定点停车指令	11-138
	DMC_MoveVelocityStopByLinePos	速度定点停车指令	11-142
	DMC_ReadPositionLagStatus	位置误差检测指令	11-146
	DMC_WritePositionLagSetting	位置误差设定指令	11-147
	DMC_ChangeMechanismGearRatio	更改轴参数指令	11-149
	DMC_TorqueControl	带速度限制的扭矩控制指令	11-151
	DMC_MoveVelocity	更改速度立即生效指令	11-156
	DMC_SwitchSoftLimit	软件极限开关指令	11-157
多轴指令	MC_GearIn	电子齿轮耦合指令	11-159
	MC_GearOut	电子齿轮脱离指令	11-165

指令组	指令码	功能	页码
	MC_CombineAxes	双主轴联合齿轮指令	11-170
	MC_CamIn	电子凸轮关联指令	11-179
	MC_CamOut	电子凸轮脱离指令	11-199
	DMC_CamReadPoint	读取凸轮点信息指令	11-204
	DMC_CamWritePoint	写入凸轮点信息指令	11-208
	DMC_CamSet	更改凸轮点生效指令	11-210
	DMC_CamReadTappetStatus	读取多个挺杆点状态指令	11-213
	DMC_CamReadTappetValue	读取单个挺杆点信息指令	11-218
	DMC_CamWriteTappetValue	写入挺杆点信息指令	11-222
	DMC_CamAddTappet	增加挺杆点指令	11-226
	DMC_CamDeleteTappet	删除挺杆点指令	11-230
应用指令	APF_RotaryCut_Init	旋切初始化指令	11-238
	APF_RotaryCut_In	旋切耦合指令	11-240
	APF_RotaryCut_Out	旋切脱离指令	11-242
G 代码指令	DMC_CartesianCoordinate	直角坐标机器人指令	11-279
	DMC_ReadMFunction	读取 M 代码状态指令	11-285
	DMC_ResetMFunction	复位 M 代码状态指令	11-288
	DMC_SetG0Para	设置 G0 参数指令	11-291
	DMC_SetG1Para	设置 G1 参数指令	11-294
	DMC_SetStartPosition	设置执行 G 代码轴的起始位置指令	11-297
轴组指令	DMC_ AddAxisToGroup	添加轴到轴组	11-300
	DMC_RemoveAxisFromGroup	从轴组中移除轴	11-302
	DMC_UngroupAllAxes	解除轴组	11-304
	DMC_ GroupEnable	轴组使能	11-306
	DMC_GroupStop	轴组停止运行	11-309
	DMC_GroupInterrupt	轴组暂停	11-315
	DMC_GroupContinue	轴组解除暂停	11-320
	DMC_MoveDirectAbsolute	绝对快速定位	11-322
	DMC_MoveDirectRelative	相对快速定位	11-327
	DMC_MoveLinearAbsolute	绝对直线插补	11-332
	DMC_MoveLinearRelative	相对直线插补	11-346
	DMC_MoveCircularAbsolute	绝对圆弧插补	11-360
	DMC_MoveCircularRelative	相对圆弧插补	11-370
	DMC_GroupSetOverride	设置轴组超调值	11-380
	DMC_GroupReadActualPosition	读取轴组位置指令	11-384

11.2 运动指令基本知识

11.2.1 运动指令构成

以“MC_或 DMC”开头的指令都是运动指令。



11.2.2 运动指令的语言

运动指令支持以下两种编程语言，详细说明请参考软件帮助。

- 梯形图（LD）
- 结构文本（ST）

11.2.3 运动指令的配置

运动指令只能添加在运动事件任务中，如果添加在其它类型的任务中，运动指令不能执行。

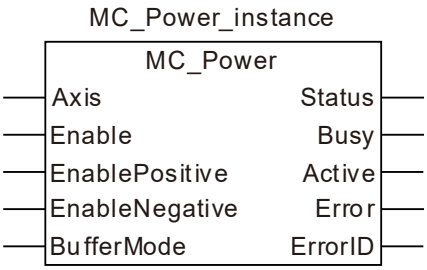
运动指令可以添加在何种类型的任务如下表所示：

任务类型		可否添加
周期		不可以
自由运行		不可以
事件	运动事件	可以
	非运动事件	不可以

11.3 单轴指令

11.3.1 MC_Power（使能指令）

FB/FC	说明	适用机种
FB	此指令用于对相应的伺服轴使能或解除使能。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节功能简介 （不可缺省）	*Enable* 为 TRUE 时
Enable （执行位）	当*Enable*由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	*Enable* 为 TRUE 时
EnablePositive （允许正转）	*Enable*为TRUE的情况下，当*EnablePositive*为TRUE时，轴才允许正转，否则禁止正转。	BOOL	TRUE 或 FALSE （FALSE）	*Enable* 为 TRUE 时
EnableNegative （允许反转）	*Enable*为TRUE的情况下，当*EnableNegative*为TRUE时，轴才允许反转，否则禁止反转。	BOOL	TRUE 或 FALSE （FALSE）	*Enable* 为 TRUE 时
Buffermode （交接模式）	*Enable*变为FALSE时，指定MC_Power的行为模式。	MC_Buffer_Mode	0: mcAborting 1: mcBuffered （0）	*Enable* 为 TRUE 时

说明：伺服轴必须在使能后，运动控制指令才能控制轴作相应的运动，当轴未使能时，运动控制指令均不能执行。

- **输出参数**

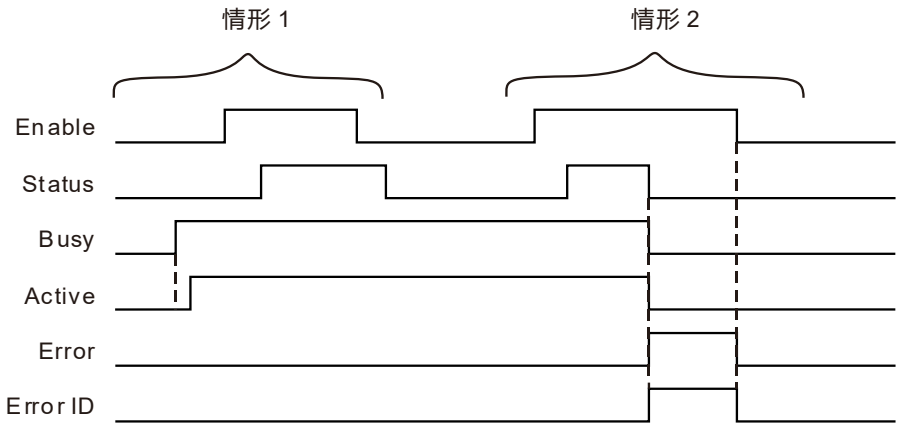
名称	功能	数据类型	输出范围
Status （状态）	该输出参数为 TRUE 时表示轴已经使能。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Status	◆ 当指定轴已经使能时	◆ 当指定轴解除使能时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当该指令执行时	◆ 当 Error 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **输出参数变化时序图**



情形1： 当该指令第一次执行时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当 *Enable* 由 FALSE 变为 TRUE 后且轴使能成功后，*Status* 变为 TRUE；当 *Enable* 由 TRUE 变为 FALSE 后，轴解除使能后，*Status* 由 TRUE 变为 FALSE。

情形2： 当该指令执行过程中，有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Status*、*Busy*、*Active* 变为 FALSE；当错误被清除时，*Error* 变为 FALSE。

- **功能说明**

此指令用于对相应的伺服轴使能或解除使能。

1. 将 *Enable* 设为 TRUE 后，只要轴没有使能，*Status* 就不会变为 TRUE。运行轴时，请务必确认 *Status* 已变为 TRUE 后开始动作。
2. 当 Enable 和 EnablePositive 同时为 TRUE 时，运动指令指定的轴被允许进行正转。
3. 当 Enable 为 TRUE 且 EnablePositive 为 FALSE 时，运动指令指定的轴被禁止进行正转。在这种情况下，如果指令控制轴进行正转，指令会发生错误。如果轴由正转变为反转，正在控制轴的指令会被中断，轴会停止运转，进入 Standstill 状态。
4. 当 Enable 和 EnableNegative 同时为 TRUE 时，运动指令指定的轴可以进行反转。

5. 当 Enable 为 TRUE 且 EnableNegative 为 FALSE 时，运动指令指定的轴被禁止进行反转。在这种情况下，如果指令控制轴进行反转，指令会发生错误。如果轴由反转变为正转，正在控制轴的指令会被中断，轴会停止运动，进入 Standstill 状态。
6. 当轴正在正转时，EnablePositive 由 TRUE 变为 FALSE，轴会以当前控制轴的指令的减速度减速运行，直到轴的速度降为 0。当轴正在反转时，EnableNegative 由 TRUE 变为 FALSE，轴会以当前控制轴的指令的减速度减速运行，直到轴的速度降为 0。
7. 原则上 1 个轴只能使用 1 个 MC_Power，如果控制同一个轴的程序中有两个 MC_Power，则以后执行的 MC_Power 的执行结果为准。
8. 运动指令正在控制轴时，MC_Power 的 Enable 由 TRUE 变为 FALSE，轴能否进入 Disabled 状态取决于 Buffermode 的取值。
9. Buffermode

   当*Enable*由TRUE变为FALSE时，*Buffermode*指定该指令的行为模式。

输出	***Buffermode*** **选择**	功能
Enable	0: mcAborting（打断）	当Enable由TRUE变为FALSE时，轴会立即停止运转，且轴解除使能状态（状态机变为Disabled）。 **请注意：在进行此操作时，可能导致人员伤亡或设备损坏。**
	1: mcBuffered（等待）	当*Enable*由TRUE变为FALSE时，轴不会立即变为Disabled，只有当轴停止运转时，状态机先进入Standstill，一个周期后，状态机再进入Disabled。

程序范例一

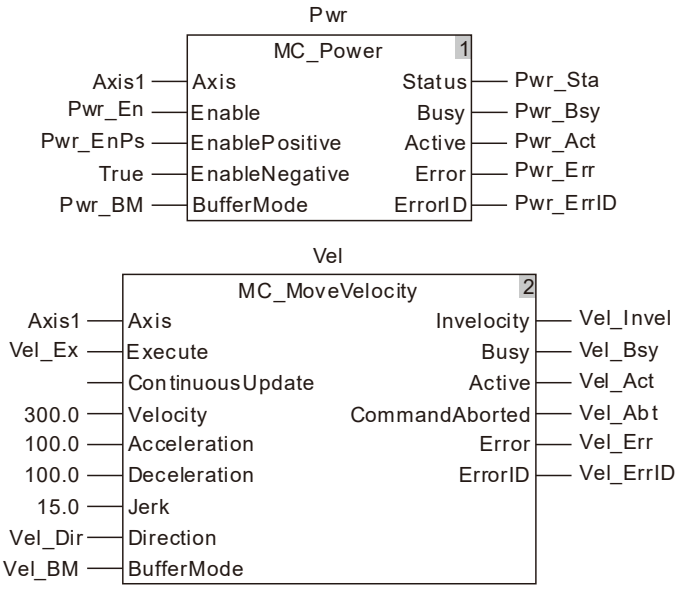
MC_Power 指令执行时的范例如下所示：

当 Pwr_En 为 TRUE 且 Pwr_EnPs 为 FALSE 时，运动指令指定的轴被禁止进行正转。当轴正在正转时，Pwr_EnPs 由 TRUE 变为 FALSE，轴会以当前控制轴的指令的减速度减速运行，直到轴的速度降为 0。

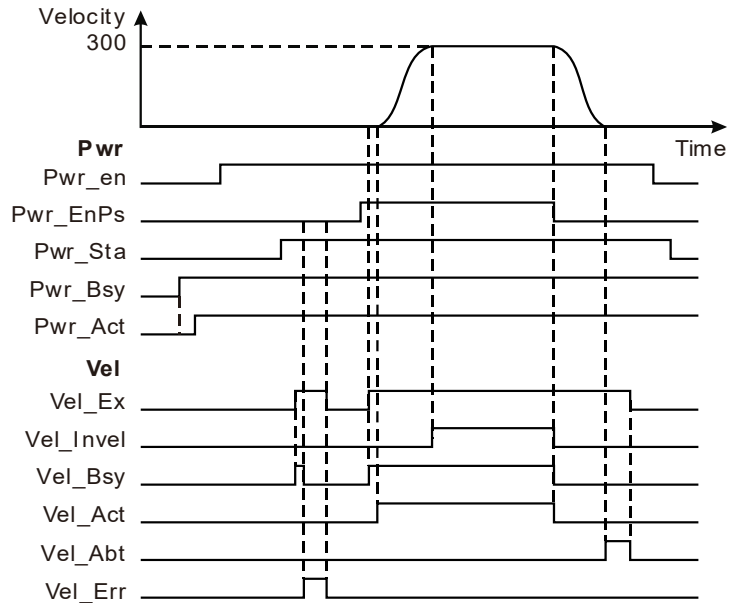
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1

变量名	数据类型	初始值
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	



2. 动曲线和时序图：



- ❖ 当 Vel_Ex 第一次变为 TRUE 时，Vel_Bsy 变为 TRUE，一个周期后 Vel_Err 变为 TRUE。此时伺服电机不能运转是因为 Pwr_EnPs 为 FALSE。
- ❖ 当 Pwr_EnPs 为 TRUE，Vel_Ex 第二次变为 TRUE 时，Vel_Bsy 变为 TRUE，一个周期后 Vel_Act 变为 TRUE，伺服电机开始正转。当伺服电机达到目标速度时，Vel_Invel 变为 TRUE。

- ❖ 当 Pwr_EnPs 变为 FALSE 时，MC_Velocity 指令被中断，伺服电机开始以 MC_Velocity 设定的减速度进行减速，当速度减为 0 时，CommandAborted 变为 TRUE。
- ❖ 当 Vel_Ex 变为 FALSE 时，Vel_Abt 变为 FALSE。
- ❖ 当 Pwr_En 变为 FALSE 时，轴解除使能后，Pwr_Sta 变为 FALSE。

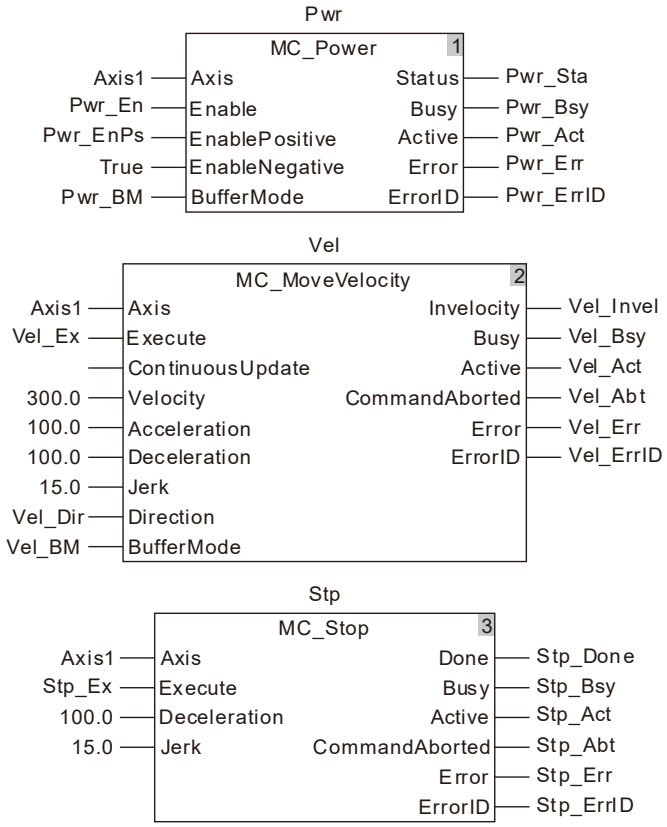
程序范例二

Vel_BM =0 的范例如下所示：

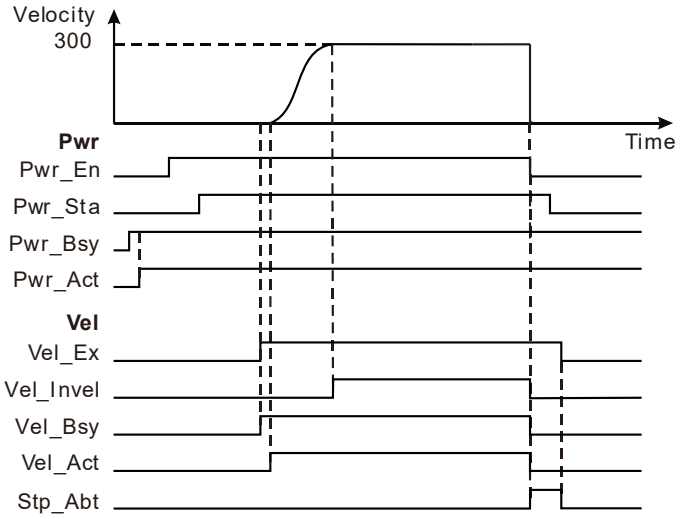
当 Vel_BM 取值为 0 时，Pwr_En 由 TRUE 变为 FALSE，轴会立即进入 Disabled 状态， 轴速度会立即降为 0。

1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	1
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
Stp	MC_Stop	
Stp_Ex	BOOL	FALSE
Stp_Done	BOOL	
Stp_Bsy	BOOL	
Stp_Act	BOOL	
Stp_Abt	BOOL	
Stp_Err	BOOL	
Stp_ErrID	WORD	



2. 运动曲线和时序图：



- ❖ 当 Vel_Ex 变为 TRUE 时， 伺服电机开始正转。当伺服电机速度达到目标速度时，Vel_Invel 变为 TRUE。
- ❖ 当 Pwr_En 变为 FALSE 时， 伺服电机速度会立即降为 0，且轴立即进入 Standstill 状态。同时，Vel_Abt 变为 TRUE，Vel_Bsy、 Vel_Act 变为 FALSE。轴解除使能后，Pwr_Sta 变为 FALSE。
- ❖ 当 Vel_Ex 变为 FALSE 时，Vel_Abt 变为 FALSE。

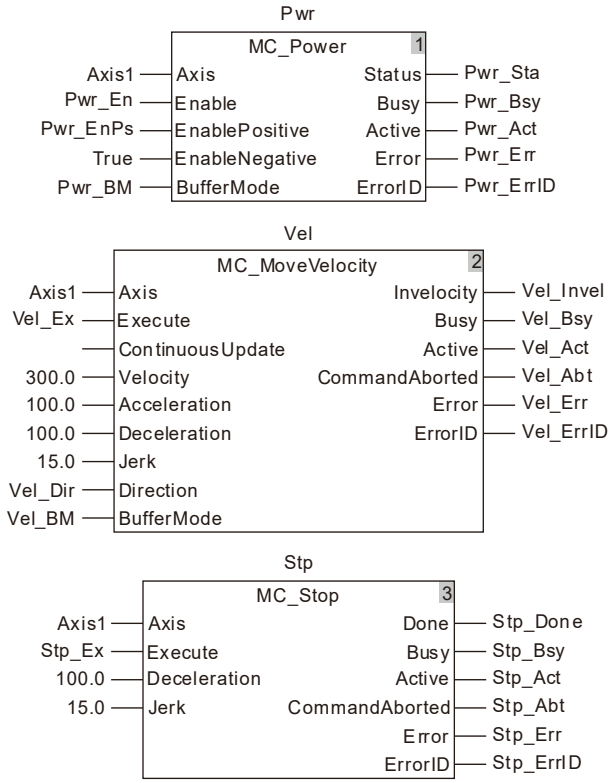
程序范例三

Buffermode=1 的范例如下所示：

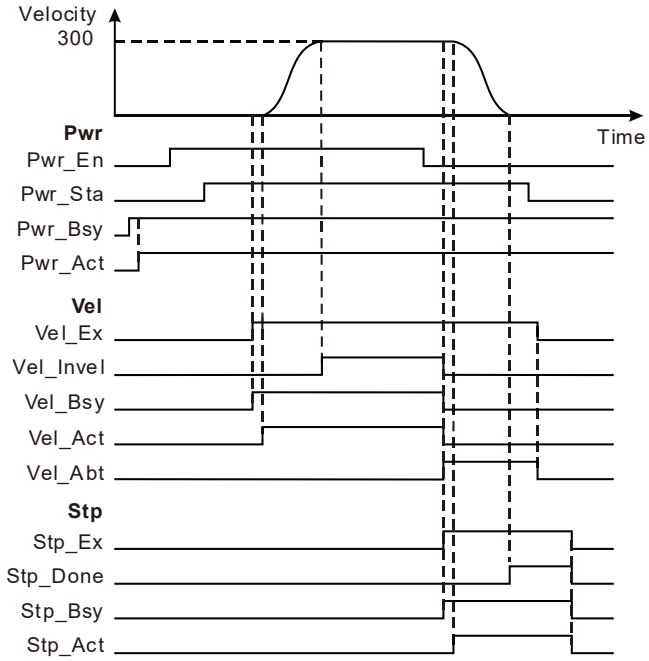
当 *Buffermode* 取值为 1 时，*Enable* 由 TRUE 变为 FALSE，只有当轴停止运转时，MC_Power 指令的 *Status* 才会发生变化。当轴停止运转时，状态机先进入 Standstill，一个周期后，状态机再进入 Disabled。

1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
Stp	MC_Stop	
Stp_Ex	BOOL	FALSE
Stp_Done	BOOL	
Stp_Bsy	BOOL	
Stp_Act	BOOL	
Stp_Abt	BOOL	
Stp_Err	BOOL	
Stp_ErrID	WORD	



2. 运动曲线和时序图

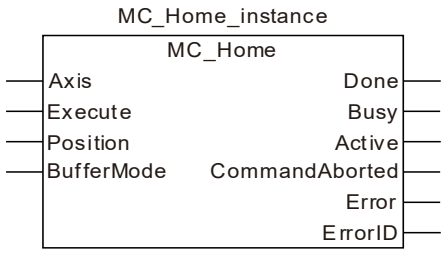


- 当 Vel_Ex 变为 TRUE 时，Vel_Bsy 变为 TRUE，一个周期后，Vel_Act 变为 TRUE，伺服电机开始正转。当伺服电机速度达到目标速度时，Vel_Invel 变为 TRUE。
- 当 Pwr_En 变为 FALSE 时，轴不会立即进入 Standstill 状态。当 Stp_Ex 变为 TRUE 时，Stp_Bsy 变为 TRUE，一个周期后，Stp_Act 变为 TRUE，伺服电机开始减速。当伺服电机速度降为 0 时，Stp_Done 变为 TRUE，同时，轴进入 Standstill 状态，且 Pwr_Sta 变为 FALSE。一个周期后，轴进入 Disabled 状态。

- ❖ 当 Vel_Ex 变为 FALSE 时，Vel_Abt 变为 FALSE。
- ❖ 当 Stp_Ex 变为 FALSE 时，Stp_Done、Stp_Bsy、Stp_Act 变为 FALSE。

11.3.2 MC_Home（原点回归指令）

FB/FC	说明	适用机种
FB	此指令用于控制伺服电机按轴参数设定的模式和速度执行回原点动作。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当*Execute*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	
Position （位置）	伺服原点偏移位置， 单位：单元。	LREAL	负数、正数、0 （0）	*Execute* 由 FALSE 变为 TRUE
BufferMode （交接模式）	保留	-	-	-

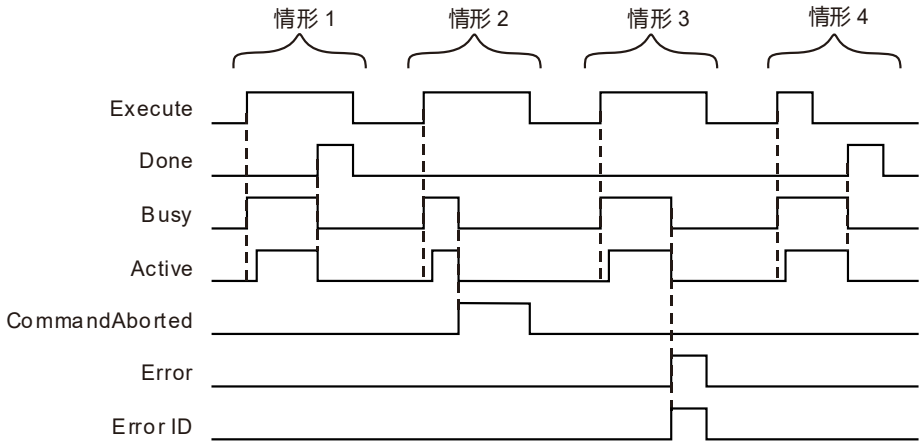
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当原点回归执行完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Command Aborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当定位完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 后，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 Execute 由 FALSE 变为 TRUE 后，当有错误产生时(如轴报警或者掉线时)，Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 和 Active 变为 FALSE ;当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

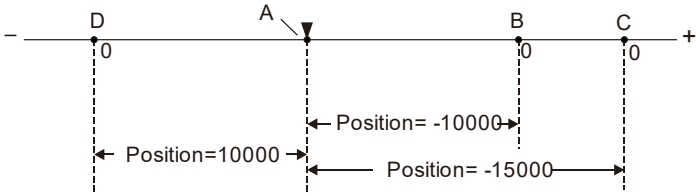
情形4： 在指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明：**

1. 本指令根据所选回原点模式，将原点开关和正向或反向极限开关接到伺服驱动器的外部输入点上以实现原点回归功能。
2. 实轴在软件轴参数部分设置原点回归模式和原点回归的一二段速。原点回归模式的详细说明请见附录 D;虚轴只能设置为原点回归模式 35。
3. 本指令只能在轴处于 StandStill 状态时才可以执行，在其它状态下执行时，本指令报错。
4. 参数 Position 定义了返回原点的位置相对于伺服位置零点的偏移量。

A	机械原点，即光电开关所在的位置
▼	该指令执行成功后伺服的位置

Position 为不同数值时，在此指令的控制下，伺服最终都会停在机械原点 A 处。但是，伺服位置的参考零点却发生了变化，如下图所示。



当 Position=10000 时，伺服位置的参考零点在 D 点，A 点位置为 10000；

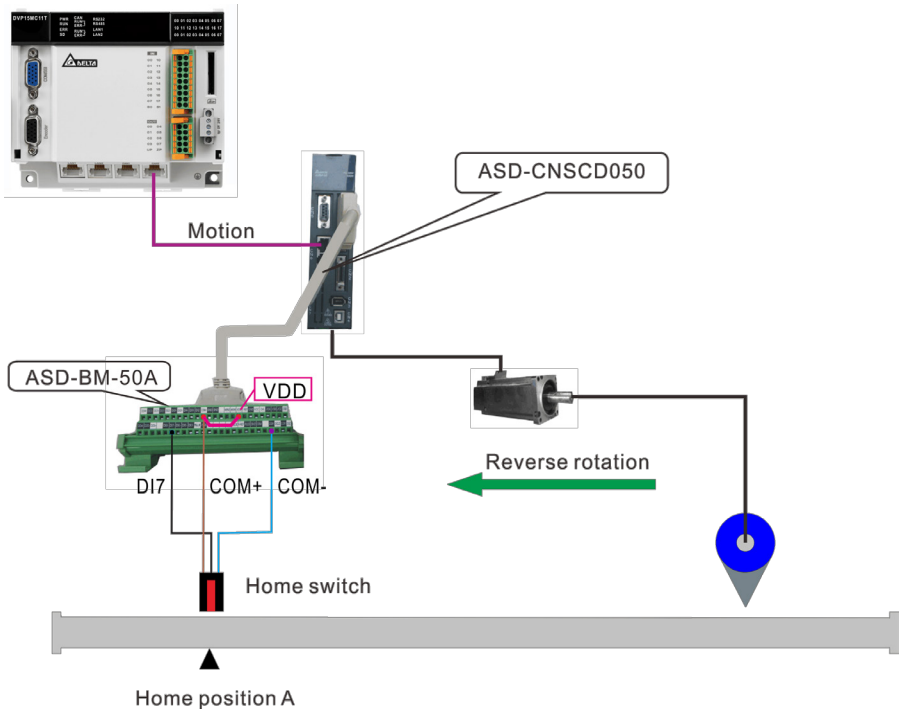
当 Position= -15000 时，伺服位置的参考零点在 C 点，A 点位置为 -15000；

当 Position= -10000 时，伺服位置的参考零点在 B 点，A 点位置为 -10000；

程序范例

通过机构和光电开关位置选择合适的原点回归模式，当 Hom _Ex 由 FALSE 变 TRUE 时,运动控制器控制伺服电机运转,带动机构回到机械原点位置 A.

- **硬件接线**



⚠ 注意

1. 接线时，COM+与 VDD 必须短接
2. 光电开关的棕色（24V+）接 COM+,蓝色（0V）接 COM-,黑色（信号线）接 DI7。
3. 数位输入（DI7）功能设定为原点开关，即 P2-16 设为 124

- **原点回归模式的选择**

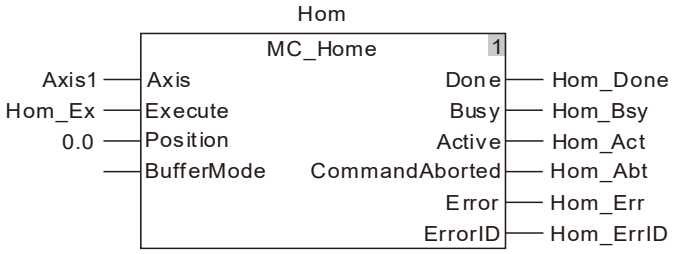
由硬件接线图可以看出:机构是以原点开关位置作为机械原点位置 A;原点开关在找原点前处于低位;机构在找原点的过程中,一开始伺服运转方向是反转的;可以选择原点回归模式 21 来实现原点回归.在相应的轴参数设置中,原点回归设置如下:

原点回归模式	21
第一段速度（到找到原点开关的速度,单位:转/分）	100
第二段速度（找到原点开关后到机械原点的速度，单位:转/分）	10

注：设置的轴参数必须下载后才会生效.

- **变量和程序**

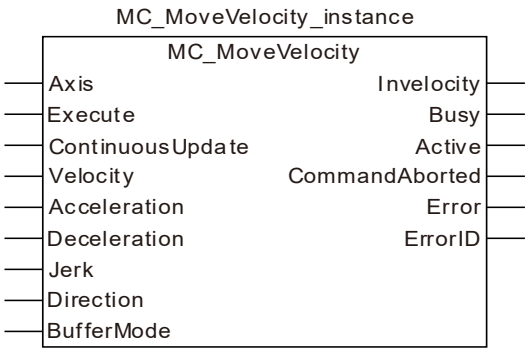
变量名	数据类型	初始值
Hom	MC_Home	
Axis1	USINT	1
Hom_Ex	BOOL	FALSE
Hom_Done	BOOL	
Hom_Bsy	BOOL	
Hom_Act	BOOL	
Hom_Abt	BOOL	
Hom_Err	BOOL	
Hom_ErrID	WORD	



- ❖ 在 Hom_Ex 由 FALSE 变 TRUE 时,运动控制器控制伺服电机运转，机构开始反转，到达原点开关后变为正转，然后在离开原点开关的位置停下，从而带动机构回到机械原点位置 A。
- ❖ 当碰到原点开关后，原点回归完成，Hom_Done 被置位。

11.3.3 MC_MoveVelocity（速度指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴按照设定的加减速运动至设定速度，并匀速运行	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
ContinuousUpdate	保留	-	-	-
Velocity （速度）	设定的目标速度 （单位: 单元/秒）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Acceleration （加速度）	设定的目标加速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Direction （方向）	设定的运转方向 1:正方向 3:反方向 4:延续当前的方向（电机停止时，当前方向为正方向）	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection 4: mcCurrentDirection （1）	*Execute* 由 FALSE 变为 TRUE
BufferMode （交接模式）	设定两个指令之间交接模式 0：打断 1：等待	MC_Buffer_Mode	0：mcAborting 1：mcBuffered 2：mcBlendingLow	*Execute* 由 FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	2：以低速交接 3：以前一指令的速度交接 4：以后一指令的速度交接 5：以高速交接		3：mcBlendingPrevious 4：mcBlendingNext 5：mcBlendingHigh （0）	

说明：

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行。无论该指令是否执行完成， *Execute* 再次由 FALSE 变为 TRUE 时，该指令可以重新执行， 此时能够重新生效的引脚参数包括 *Velocity*、*Acceleration*、*Deceleration*、*Jerk*、*Direction*，其它引脚参数不会生效。当速度指令与其它运动指令产生 *BufferMode* 关系时，改变速度指令的参数，重新触发后速度指令的参数会生效，原来的交接关系仍然保持，交接速度会重新计算。
2. 当该指令执行完成后，即 *Invelocity* 由 FALSE 变为 TRUE 后，即使通过 MC_SetOverride 指令改变目标速度，此时 Invelocity 依然保持为 TRUE。当 MC_MoveVelocity 未完成时，即 *Invelocity* 为 FALSE 时，通过 MC_SetOverride 指令改变目标速度，当到达新的目标速度后，*Invelocity* 才会由 FALSE 变为 TRUE。
3. *Position*、*Velocity*、*Acceleration*、*Jerk* 的关系说明请参考 10.2 节。
4. 关于 *BufferMode* 的详细内容，请参考 10.3 节

- **输出参数**

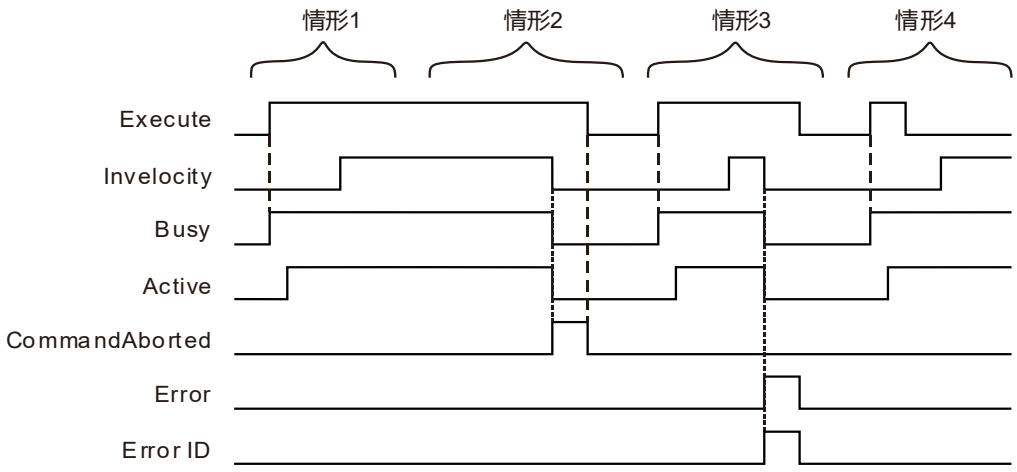
名称	功能	数据类型	输出范围
Invelocity （速度到达）	该输出参数为 TRUE 时表示速度到达	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Invelocity	◆ 当速度到达时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当速度到达后改变输入参数， Execute 再次由 FALSE 变为 TRUE 时，Invelocity 会立即变为 FALSE；当速

名称	变为 TRUE 的时机	变为 FALSE 的时机
		度指令完成后不改变输入参数，Execute 再次由 FALSE 变为 TRUE 时 Invelocity 会立即变为 FALSE，且下个周期 Invelocity 变为 TRUE。
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当速度到达时，*Invelocity* 变为 TRUE，同时 *Busy* 和 *Active* 依然保持为 TRUE 状态。

情形2： 当 *Execut* 为 TRUE 时，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Invelocity*、*Busy* 和 *Active* 变为 FALSE，当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 为 TRUE 时，当有错误产生时（如参数错误），*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Invelocity*、*Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中，当 *Execute* 由 TRUE 变为 FALSE 后，速度到达后，*Invelocity* 变为 TRUE，同时 *Busy* 和 *Active* 依然保持为 TRUE 状态。

- **功能说明**

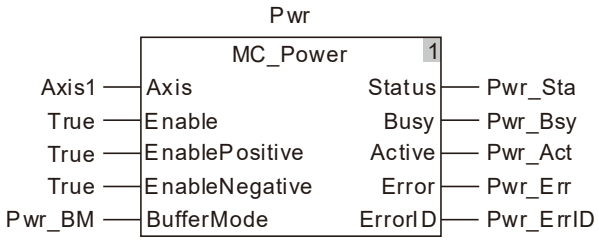
此指令用于控制轴按照设定设定加减速以及加速度变化率，加速或者减速到设定的速度进行匀速运动。匀速运动方向由 *Direction* 引脚控制，1 为正方向，3 为反方向，4 为延续当前方向，若当前轴处于停止状态，则按正方向开始运动。

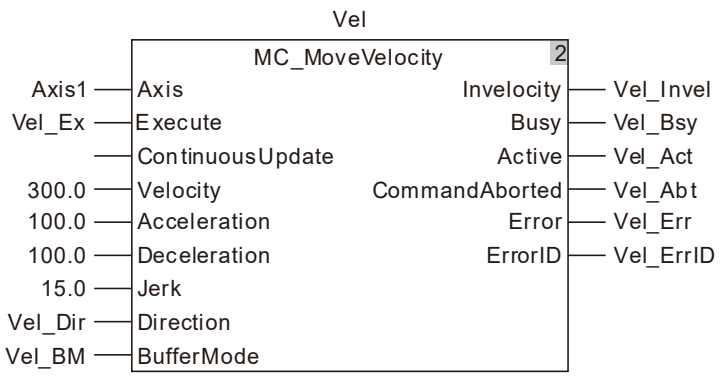
程序范例一

MC_MoveVelocity 指令单独执行时的范例如下所示。

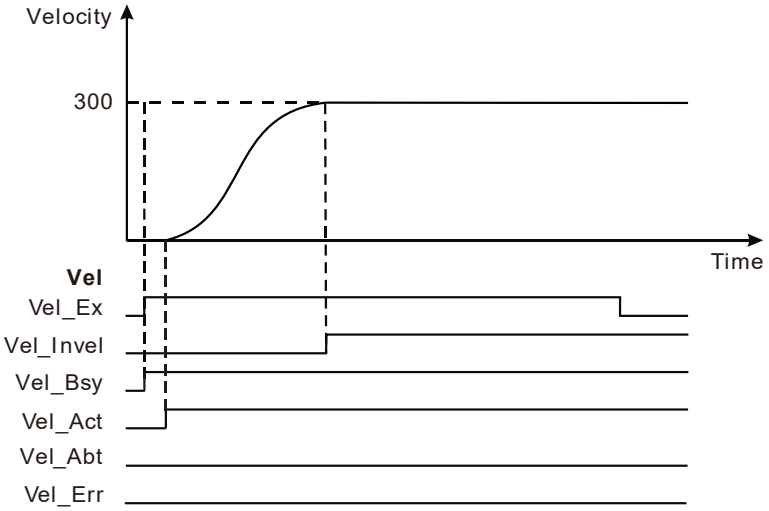
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	1
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	





2. 运动曲线和时序图



- ❖ 当 Vel_Ex 由 FALSE 变为 TRUE 时，同时 Vel_Bsy 变为 TRUE，且一个周期后，Vel_Act 变为 TRUE，开始执行速度指令；当速度到达时，Vel_Invel 变为 TRUE，同时 Vel_Bsy 和 Vel_Act 依然保持为 TRUE。
- ❖ 当 Vel_E 由 TRUE 变为 FALSE 时， Vel_Inve、 Vel_Bsy 和 Vel_Act 依然保持为 TRUE。

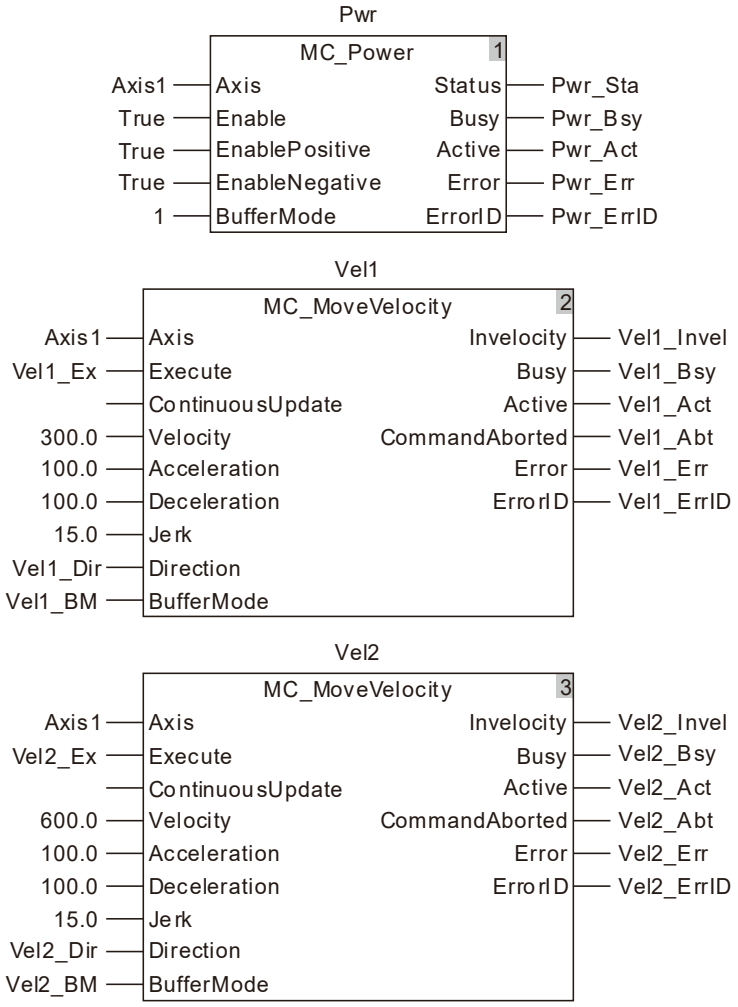
程序范例二

MC_MoveVelocity 指令打断 MC_MoveVelocity 指令的范例如下所示：

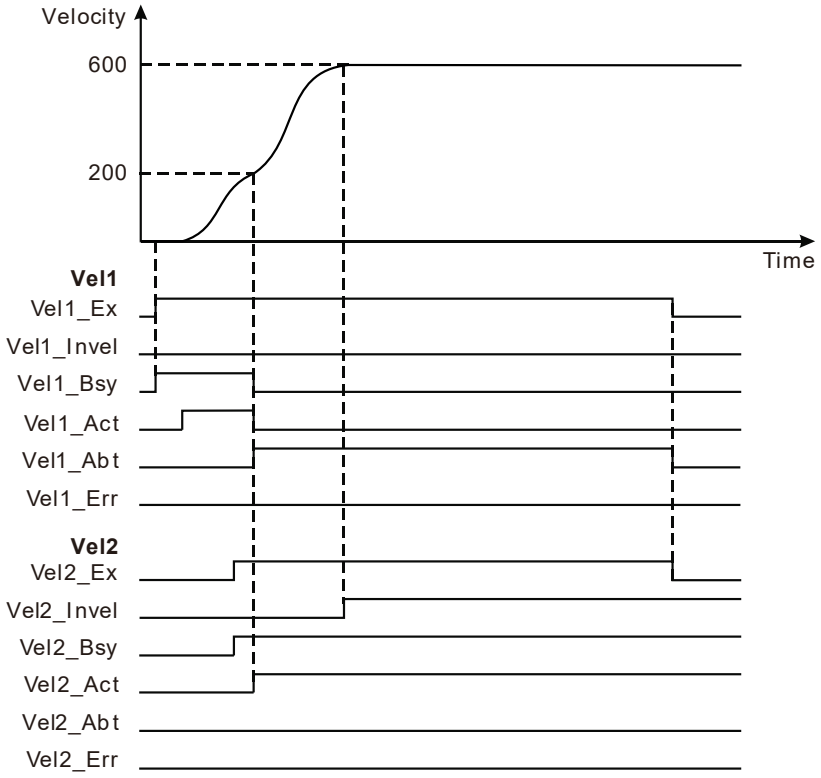
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	1
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel1	MC_MoveVelocity	
Vel1_Ex	BOOL	FALSE
Vel1_Dir	MC_DIRECTION	1
Vel1_BM	MC_Buffer_Mode	0
Vel1_Invel	BOOL	
Vel1_Bsy	BOOL	
Vel1_Act	BOOL	
Vel1_Abt	BOOL	
Vel1_Err	BOOL	
Vel1_ErrID	WORD	
Vel2	MC_MoveVelocity	
Vel2_Ex	BOOL	FALSE

变量名	数据类型	初始值
Vel2_Dir	MC_DIRECTION	1
Vel2_BM	MC_Buffer_Mode	0
Vel2_Invel	BOOL	
Vel2_Bsy	BOOL	
Vel2_Act	BOOL	
Vel2_Abt	BOOL	
Vel2_Err	BOOL	
Vel2_ErrID	WORD	



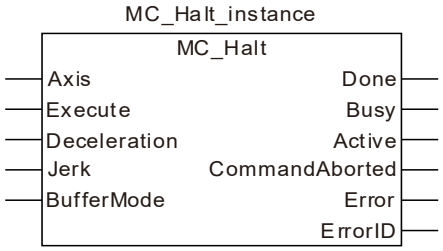
2. 运动曲线和时序图



- 当 Vel1_Ex 由 FALSE 变为 TRUE 时，同时 Vel1_Bsy 变为 TRUE，且一个周期后，Vel1_Act 变为 TRUE，开始执行第一个速度指令；当还未到达目标速度时，Vel2_Ex 由 FALSE 变为 TRUE，同时 Vel2_Bsy 变为 TRUE，且一个周期后，Vel2_Act 变为 TRUE，同时打断第一个速度指令,Vel1_Abt 变为 TRUE，轴开始执行第二个速度指令；当速度到达时，Vel2_Invel 变为 TRUE，同时 Vel2_Bsy 和 Vel2_Act 依然保持为 TRUE。
- 当 Vel1_Ex 由 TRUE 变为 FALSE 时， 同时 Vel1_Abt 变为 FALSE；当 Vel2_Ex 由 TRUE 变为 FALSE 时，Vel2_Invel、Vel2_Bsy 和 Vel2_Act 依然保持为 TRUE。

11.3.4 MC_Halt（暂停指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴按设定的减速度减速，直到停止。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节功能简介 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由FALSE 变为 TRUE
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由FALSE 变为 TRUE
BufferMode （交接模式）	设定两个指令之间交接模式 0: 打断 1: 等待	MC_Buffer_Mode	0: mcAborting 1: mcBuffered （0）	*Execute* 由FALSE 变为 TRUE

说明:

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行。该指令正在执行，*Execute* 由 TRUE 变为 FALSE 时，对该指令执行无影响。
2. 当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行；当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行。
3. Deceleration、Jerk 的关系说明请参考第 10.2 节。
4. 关于 BufferMode 的详细内容，请参考第 10.3 节。

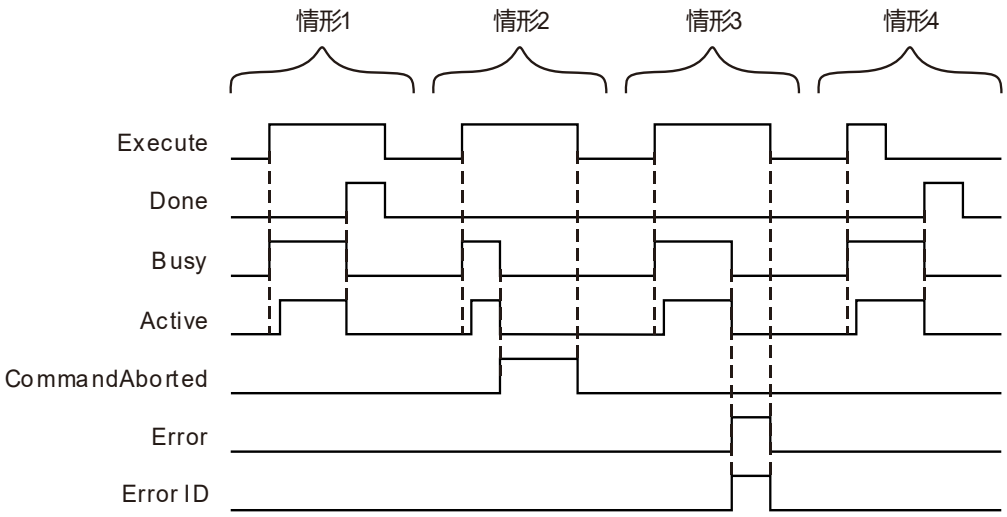
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 减速停止完成，轴速度降为 0 后	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Command Aborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图:**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当减速停止完成，轴速度降为 0 后，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE，该指令被其它指令中断后，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，CommandAborted 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 后 ,当有错误产生时(如轴报警或者掉线时) ,*Error* 变为 TRUE，*ErrorID* 显示对应的错误码 ,同时 *Busy* 和 *Active* 变为 FALSE ;当 *Execute* 由 TRUE 变为 FALSE 后，*Error* 变为 FALSE。

情形4： 在指令执行过程中, *Execute* 由 TRUE 变为 FALSE 后，指令执行完成时, *Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

此指令用于轴按设定的减速度减速，直到停止。

- MC_Halt 指令开始执行，状态机进入 DiscreteMotion，当轴速度降为 0 时，Done 变为 TRUE，同时状态机变为 Standstill。
- 和 MC_Stop 指令相比较，MC_Halt 指令不会锁定轴，控制器可以执行其它的运动指令。在 MC_Halt 指令执行期间，轴在进行减速时，可以执行其它的运动指令来中断 MC_Halt 指令；在 MC_Halt 指令执行完毕，轴已经停下来后，控制器可以执行其它的运动指令来重新启动轴。

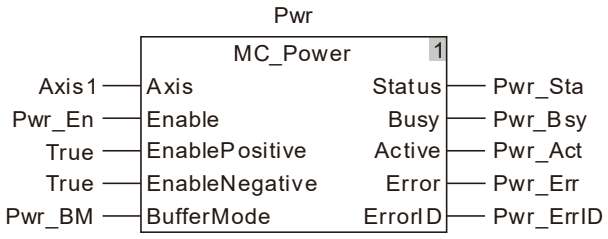
程序范例

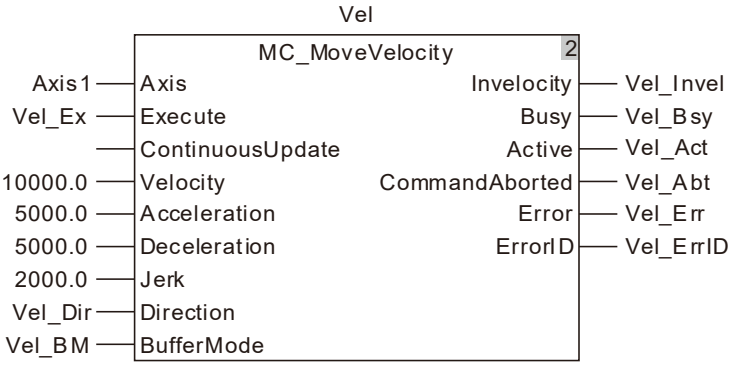
MC_Halt 指令执行时的范例如下所示：

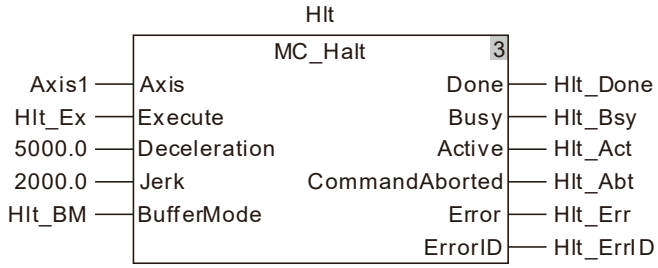
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	

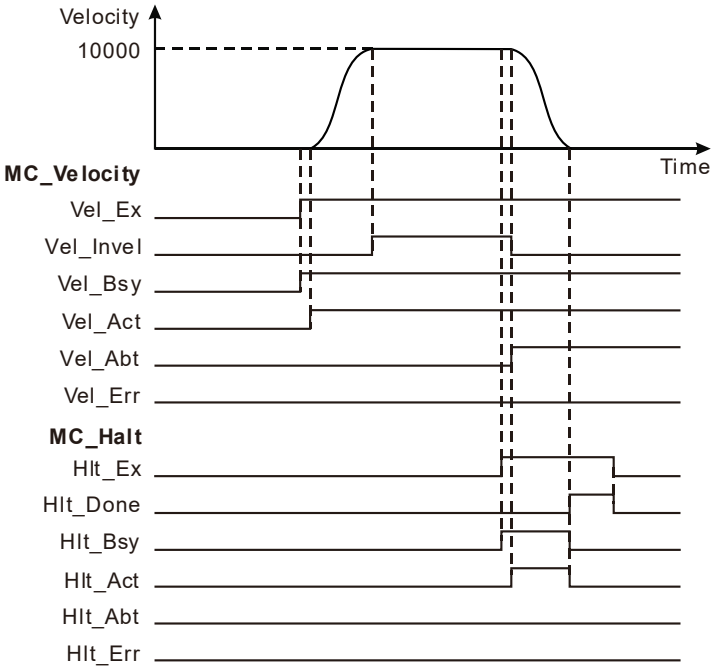
变量名	数据类型	初始值
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
Hlt	MC_Halt	
Hlt_Ex	BOOL	FALSE
Hlt_BM	MC_Buffer_Mode	0
Hlt_Done	BOOL	
Hlt_Bsy	BOOL	
Hlt_Act	BOOL	
Hlt_Abt	BOOL	
Hlt_Err	BOOL	
Hlt_ErrID	WORD	







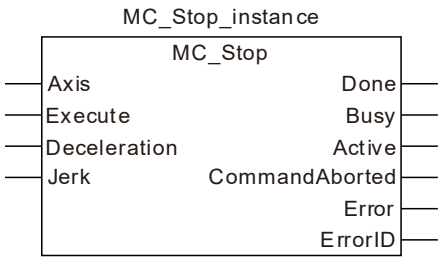
2. 运动曲线和时序图:



- ❖ 当 Vel_Ex 变为 TRUE 时，Vel_Bsy 变为 TRUE，一个周期后 Vel_Act 变为 TRUE，伺服电机开始正转。当伺服电机达到目标速度时，Vel_Invel 变为 TRUE。
- ❖ 当 Hlt_Ex 变为 TRUE 时，Hlt_Bsy 变为 TRUE，一个周期后 Hlt_Act 变为 TRUE，同时 Vel_Invel 变为 FALSE，Vel_Abt 变为 TRUE，伺服电机开始减速。
- ❖ 当轴速度减为 0 时，Hlt_Done 变为 TRUE，同时 Hlt_Bsy、 Hlt_Act 变为 FALSE。
- ❖ 当 Hlt_Ex 变为 FALSE 时，Hlt_Done 变为 FALSE。

11.3.5 MC_Stop（停止指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴按设定的减速度减速，直到停止，轴状态机进入 Stopping。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节功能简介 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE

说明:

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，*Execute* 由 TRUE 变为 FALSE 时，对该指令执行无影响。
2. 当指令正在执行中，*Execute* 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行；当指令执行完成后，*Execute* 再次由 FALSE 变为 TRUE 时，该指令可以重新执行。
3. *Deceleration*、*Jerk* 的关系说明请参考第 10.2 节。

- **输出参数**

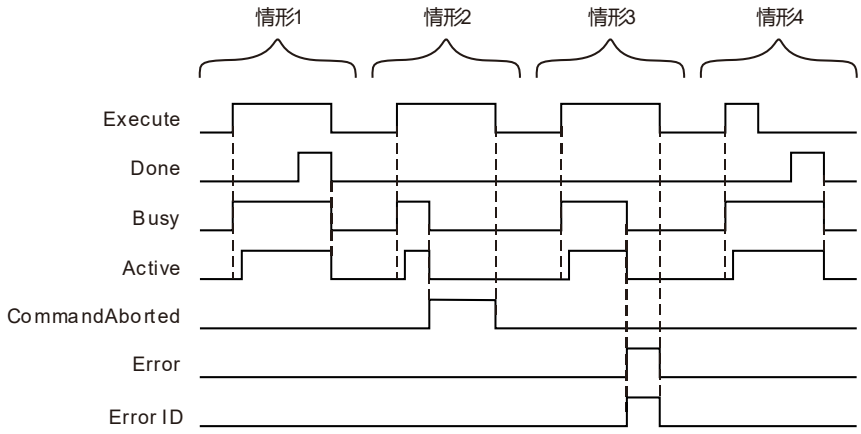
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
CommandAborted（中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 减速停止完成，轴速度降为 0 后	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Done 由 TRUE 变为 FALSE 时，Busy 变为 FALSE
Active	◆ 当指令开始控制轴时	◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Done 由 TRUE 变为 FALSE 时，Active 变为 FALSE
CommandAborted	◆ 当该指令被另外一个 MC_Stop 指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被另一个 MC_Stop 中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图:**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当减速停止完成，轴速度降为 0 后，*Done* 变为 TRUE，*Busy*、*Active* 保持 TRUE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 后，该指令被另一个 MC_Stop 指令中断时，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 后，当有错误产生时(如轴报警或者掉线时)，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，*Error* 变为 FALSE。

情形4： 在指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，指令执行完成时，*Done* 变为 TRUE，*Busy*、*Active* 保持 TRUE，且一个周期后，*Done*、*Busy*、*Active* 同时变为 FALSE。

- **功能说明**

此指令用于轴按设定的减速度减速，直到停止。

- MC_Stop 指令执行完成，轴速度降为 0 后，只要 *Excute* 为 TRUE，轴状态就一直为 Stopping，此时不能执行其它运动指令。
- 控制同一个轴的程序中有两个 MC_Stop 指令时，后执行的 MC_Stop 指令会中断前面执行的 MC_Stop 指令。
- 和 MC_Halt 指令相比较，MC_Stop 指令会锁定轴，控制器在执行 MC_Stop 指令期间不能执行其它的运动指令 (不包括 MC_Stop 指令)。MC_Stop 指令执行完毕，轴已经停下来后，控制器也不可以执行其它的运动指令，只有当 MC_Stop 的 *Excute* 由 TRUE 变为 FALSE 后才能执行其它运动指令。

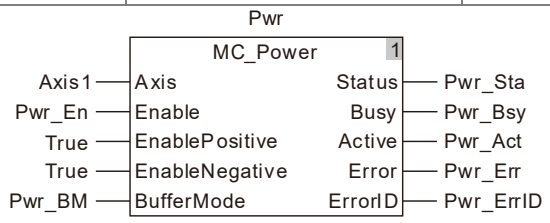
程序范例

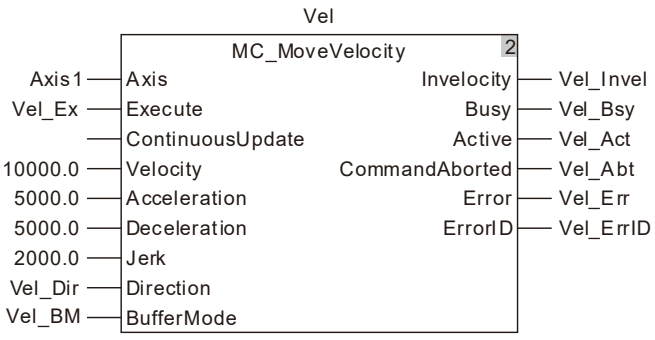
MC_Stop 指令执行时的范例如下所示：

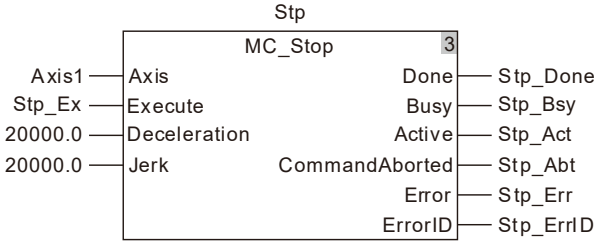
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0

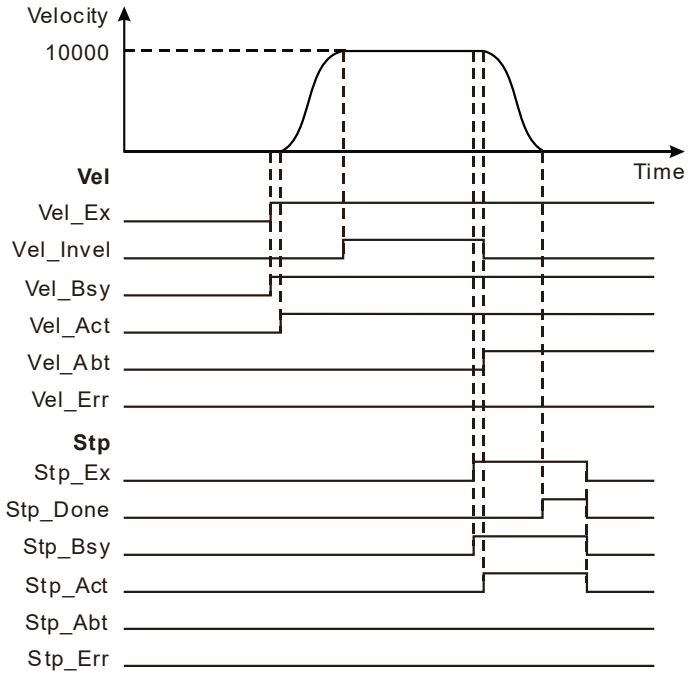
变量名	数据类型	初始值
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
Stp	MC_Stop	
Stp_Ex	BOOL	FALSE
Stp_Done	BOOL	
Stp_Bsy	BOOL	
Stp_Act	BOOL	
Stp_Abt	BOOL	
Stp_Err	BOOL	
Stp_ErrID	WORD	







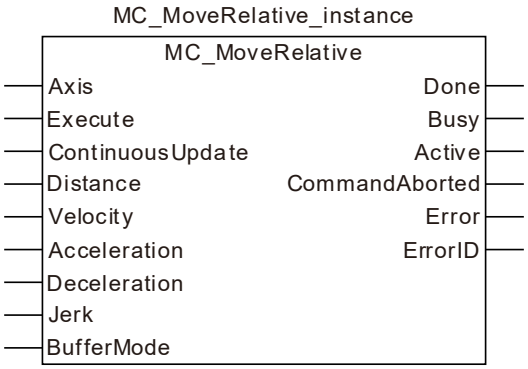
2. 运动曲线和时序图:



- ❖ 当 Vel_Ex 变为 TRUE 时，Vel_Bsy 变为 TRUE，一个周期后 Vel_Act 变为 TRUE，伺服电机开始正转。当伺服电机达到目标速度时，Vel_Invel 变为 TRUE。
- ❖ 当 Stp_Ex 变为 TRUE 时，Stp_Bsy 变为 TRUE，一个周期后 Stp_Act 变为 TRUE，同时 Vel_Invel 变为 FALSE，Vel_Abt 变为 TRUE，伺服电机开始减速。
- ❖ 当轴速度减为 0 时，Stp_Done 变为 TRUE，同时 Stp_Bsy、Stp_Act 保持 TRUE。
- ❖ 当 Stp_Ex 变为 FALSE 时，Stp_Done、Stp_Bsy、Stp_Act 同时变为 FALSE。

11.3.6 MC_MoveRelative（相对位移指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴以当前位置为起点，按设定的速度，加减速，加速度的变化率移动设定的距离	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
ContinuousUpdate	保留	-	-	-
Distance （距离）	设定的移动距离 （单位: 单元）	LREAL	负数、正数、0 （0）	*Execute* 由 FALSE 变为 TRUE
Velocity （速度）	设定的目标速度 （单位: 单元/秒）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Acceleration （加速度）	设定的目标加速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
BufferMode （交接模式）	设定两个指令之间交接模式 0：打断 1：等待	MC_Buffer_Mode	0：mcAborting 1：mcBuffered 2：mcBlendingLow 3：mcBlendingPrevious	*Execute* 由 FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	2：以低速交接 3 :以前一指令的速度交接 4 :以后一指令的速度交接 5：以高速交接		4：mcBlendingNext 5：mcBlendingHigh （0）	

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 当指令正在执行中，*Execute* 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行；当指令执行完成后，*Execute* 再次由 FALSE 变为 TRUE 时，该指令可以重新执行，且按常规方式启动。
3. *Velocity*、*Acceleration*、*Jerk* 的关系说明请参考第 10.2 节。
4. 关于 *BufferMode* 的详细内容，请参考第 10.3 节。

- **输出参数**

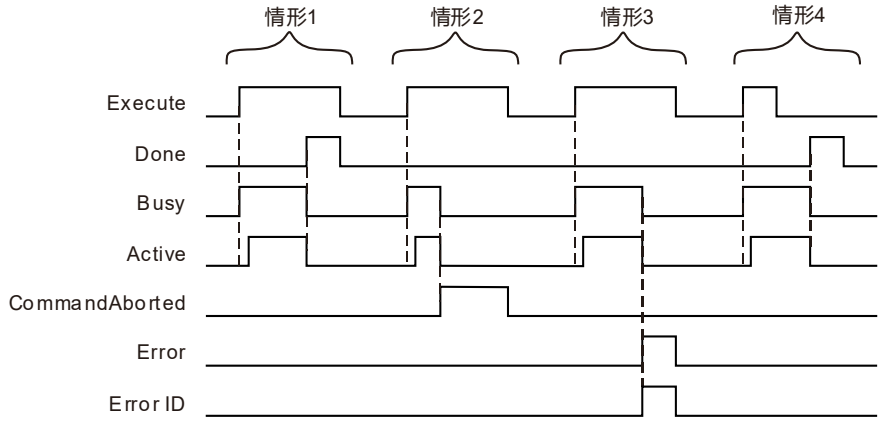
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当定位完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当定位完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE，当有错误产生时(如轴报警或者掉线时)，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Active* 变为 FALSE ;当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

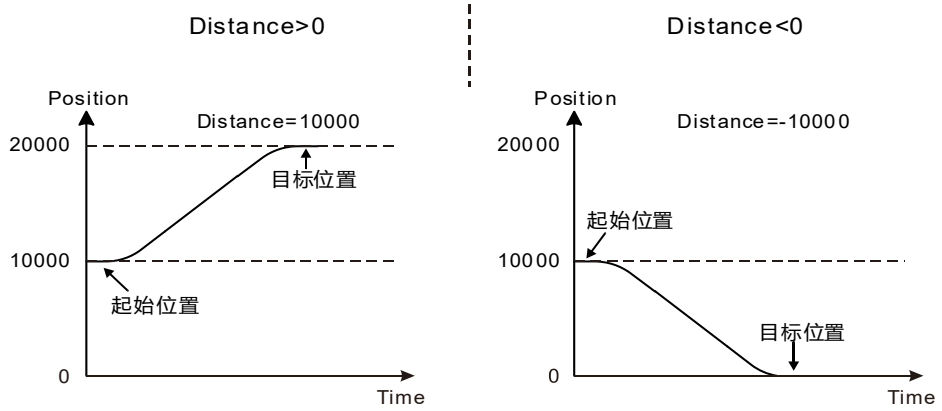
此指令用于控制轴按照设定速度，加减速以及加速度的变化率移动设定的距离，该距离的参考起点为指令开始执行时的轴位置。

■ **Distance**

Distance 与参考起点共同决定轴在该指令控制下移动的目标位置，即目标位置= 参考起点位置 +*Distance*。

当 *Distance* 为 0 时，轴运动的目标位置为当前位置，指令在开始执行的下一周期即执行完成，*Done* 变为 TRUE.

如下图所示，参考起点的位置为 10000，当 Distance>0(10000)时，轴将正向运动，目标位置为 20000（10000+10000），如左下图；当 Distance<0（-10000）时，轴将反转，目标位置为 0（10000-10000），如右下图。

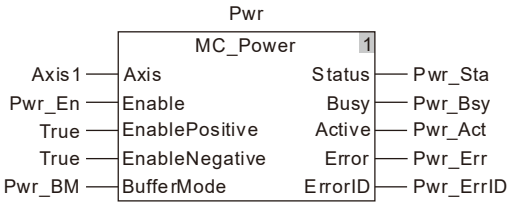


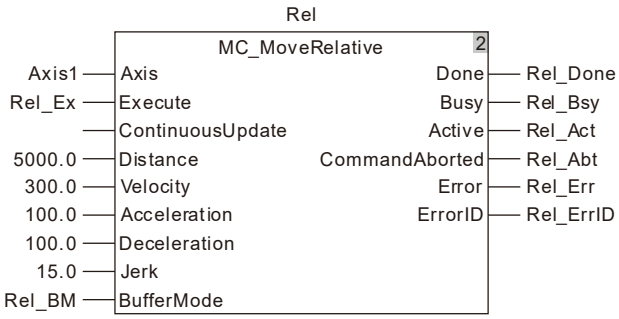
程序范例一

MC_MoveRelative 指令单独执行时的程序范例如下所示：

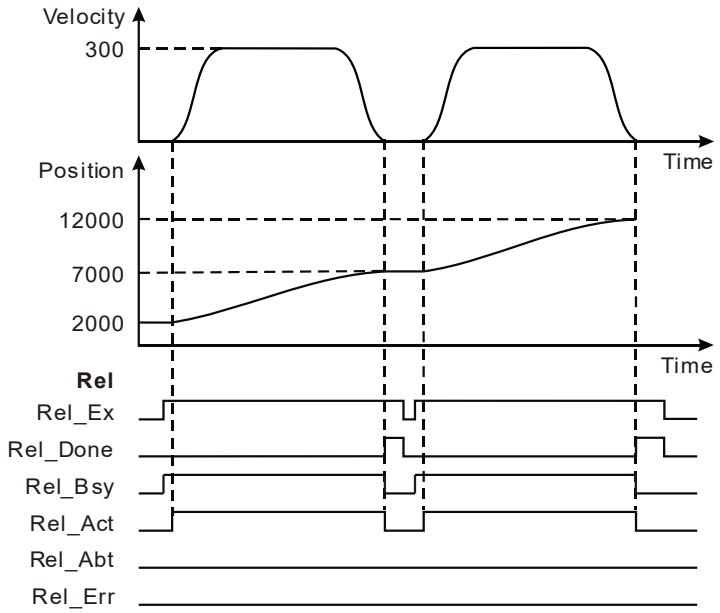
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_BM	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	





2. 运动曲线和时序图



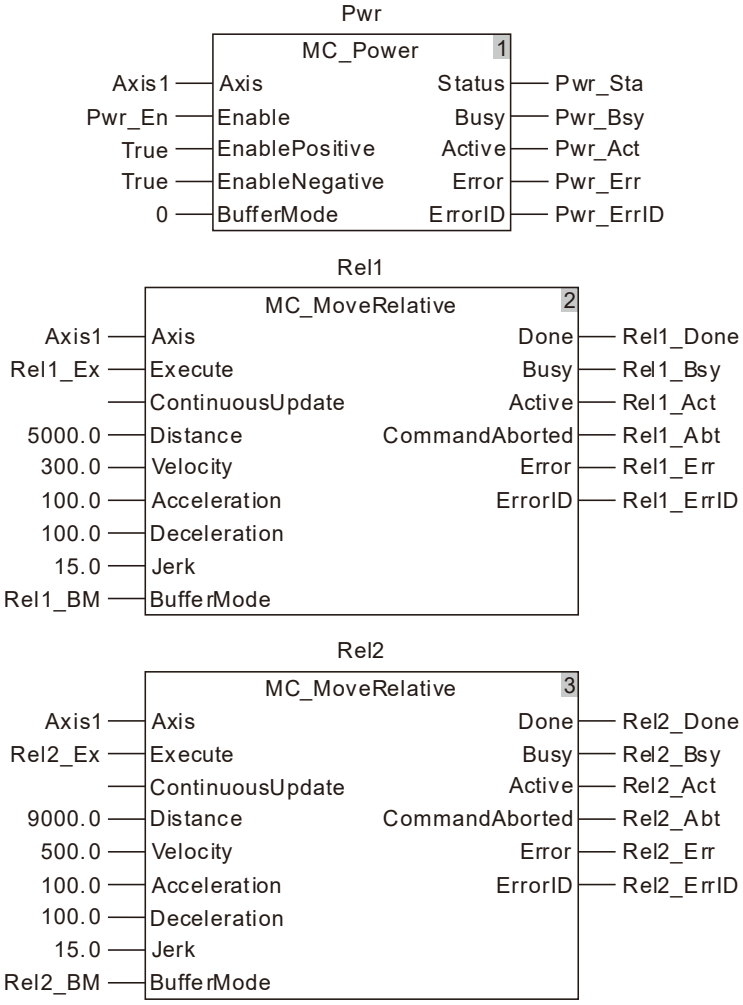
- ❖ 当 Rel_Ex 第一次由 FALSE 变为 TRUE 时， MC_MoveRelative 指令开始第一次执行，此时轴的当前位置为 2000，目标位置为 7000（7000=2000+5000）。
- ❖ 当轴位置到达 7000 时，指令执行完成，输出参数 Done 变为 TRUE。
- ❖ 当 Rel_Ex 第二次由 FALSE 变为 TRUE 时， MC_MoveRelative 指令开始第二次执行，此时轴的当前位置为 7000，目标位置为 12000（12000=7000+5000）。
- ❖ 当轴位置到达 12000 时，指令第二次执行完成，输出参数 Done 第二次变为 TRUE。

程序范例二

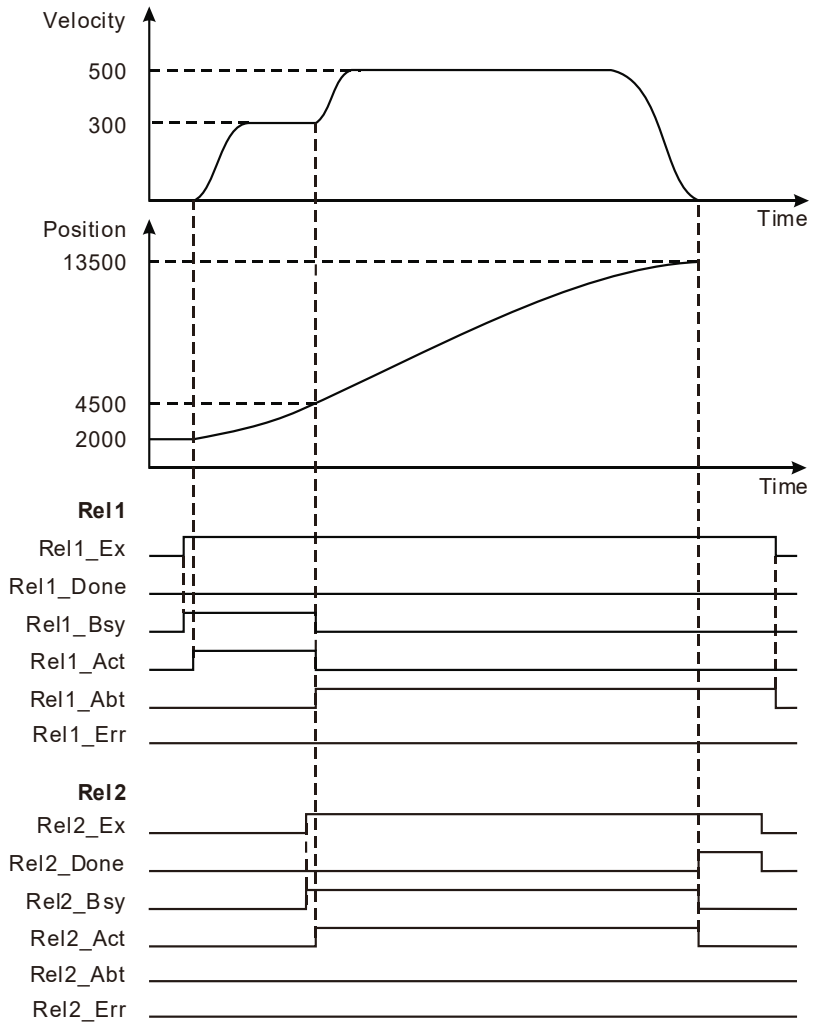
MC_MoveRelative 指令执行过程中被中断的范例如下所示：

1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	FALSE
Rel1_BM	MC_Buffer_Mode	0
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	FALSE
Rel2_BM	MC_Buffer_Mode	0
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	WORD	


Pwr
MC_Power
1
Axis1
Axis
Status
Pwr_Sta
Pwr_En
Enable
Busy
Pwr_Bsy
True
EnablePositive
Active
Pwr_Act
True
EnableNegative
Error
Pwr_Err
0
BufferMode
ErrorID
Pwr_ErrID
Rel1
MC_MoveRelative
2
Axis1
Axis
Done
Rel1_Done
Rel1_Ex
Execute
Busy
Rel1_Bsy
ContinuousUpdate
Active
Rel1_Act
5000.0
Distance
CommandAborted
Rel1_Abt
300.0
Velocity
Error
Rel1_Err
100.0
Acceleration
ErrorID
Rel1_ErrID
100.0
Deceleration
15.0
Jerk
Rel1_BM
BufferMode
Rel2
MC_MoveRelative
3
Axis1
Axis
Done
Rel2_Done
Rel2_Ex
Execute
Busy
Rel2_Bsy
ContinuousUpdate
Active
Rel2_Act
9000.0
Distance
CommandAborted
Rel2_Abt
500.0
Velocity
Error
Rel2_Err
100.0
Acceleration
ErrorID
Rel2_ErrID
100.0
Deceleration
15.0
Jerk
Rel2_BM
BufferMode

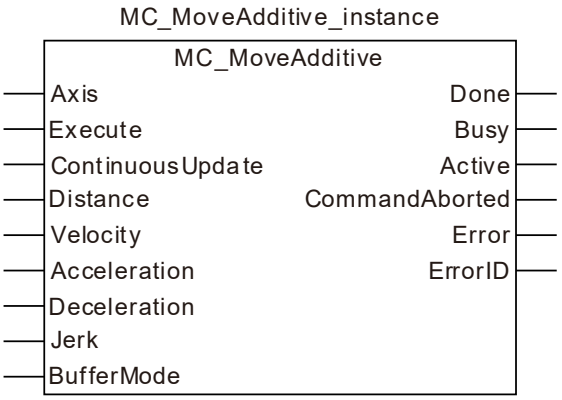
2. 运动曲线和时序图



- ❖ 当 Rel1_Ex 由 FALSE 变为 TRUE 时， 第一个 MC_MoveRelative 指令开始执行，此时轴的当前位置为 2000，目标位置为 7000（7000=2000+5000）。
- ❖ 当轴位置到达 4500 时，Rel2_Ex 由 FALSE 变为 TRUE，第二个 MC_MoveRelative 指令开始执行，且第一个 MC_MoveRelative 指令的执行被中断，其输出参数 Rel1_Abt 变为 TRUE。
- ❖ 当轴位置到达 13500（13500=4500+9000）时，第二个 MC_MoveRelative 执行完成，其输出参数 Rel2_Done 变为 TRUE。

11.3.7 MC_MoveAdditive（附加位移指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴按照设定速度，加减速移动一段附加的距离	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
ContinuousUpdate	保留	-	-	-
Distance （距离）	设定的附加距离 （单位: 单元）	LREAL	负数、正数、0 （0）	*Execute* 由 FALSE 变为 TRUE
Velocity （速度）	设定的目标速度 （单位: 单元/秒）	LREAL	正数或 0 （0）	*Execute* 由 FALSE 变为 TRUE
Acceleration （加速度）	设定的目标加速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
BufferMode （交接模式）	设定两个指令之间交接模式 0：打断 1：等待 2：以低速交接 3：以前一指令的速度交接	MC_Buffer_Mode	0：mcAborting 1：mcBuffered 2：mcBlendingLow 3：mcBlendingPrevious 4：mcBlendingNext	*Execute* 由 FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	4：以后一指令的速度交接 5：以高速交接		5：mcBlendingHigh （0）	

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令执行无影响。
2. 当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行；当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行，且按常规方式启动。
3. Position、Velocity、Acceleration、Jerk 的关系说明请参考第 10.2 节。
4. 关于 BufferMode 的详细内容，请参考第 10.3 节。

- **输出参数**

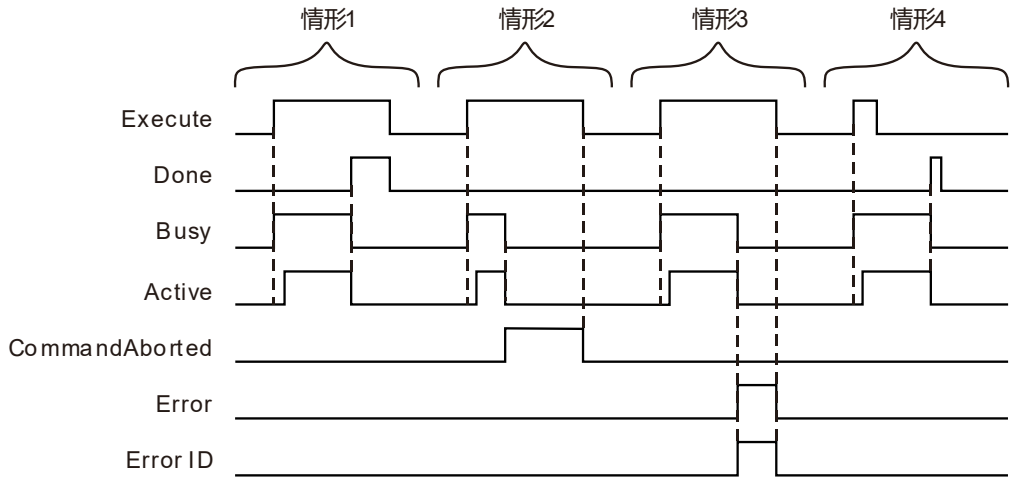
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当附加位移执行完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
		◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE；当定位完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 时，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 时，当有错误产生时（如轴报警或者掉线时）， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 Busy 和 Active 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中， *Execute* 由 TRUE 变为 FALSE 后， 指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

此指令用于控制终端执行机构按设定的速度，加速度移动一段附加的距离。

当前一个指令为位移相关指令时，如未完成，则此指令执行时终端执行机构移动的距离为前一个指令剩余的距离和此指令设定的距离总和，当此指令执行完成时，终端执行机构的最终位置为前一个位移指令和此指令设定的距离总和；当前一个指令为速度指令时，此指令执行时会终止速度指令执行，并按设定的速度，加减速移动设定的距离后停止。

当 MC_MoveSuperimposed 单独执行时执行此指令（MC_MoveAdditive），如果此指令的输入参数 BufferMode 为 0，则此指令会立即打断 MC_MoveSuperimposed，且此指令执行时终端机构移动的距离为此指令设定距离和 MC_MoveSuperimposed 指令剩余距离总和；如果此指令的输入参数 BufferMode 为 1~5，那么此指令会立即报错， MC_MoveSuperimposed 指令继续执行。

当 MC_MoveSuperimposed 与位移指令搭配使用时，再执行 MC_MoveAdditive 指令，如果此指令的输入参数 BufferMode 为 0，则此指令会立即打断 MC_MoveSuperimposed 以及 MC_MoveSuperimposed 搭配的位移指令，此指令执行时，终端机构移动的距离为此指令给定距离、MC_MoveSuperimposed 剩余位移、MC_MoveSuperimposed 搭配的位移指令剩余位移的总和；若此指令的输入参数 BufferMode 为 1~5，则此指令会等 MC_MoveSuperimposed 搭配的位移指令完成后，再执行 MC_MoveAdditive 指令。

- **MC_MoveSuperimposed 执行中启动本指令**

本指令交接模式选择	MC_MoveSuperimposed 是否与其它位移指令搭配	说明
0（中断）	是	◆ MC_MoveSuperimposed 及其搭配的位移指令立即被打断 ◆ MC_MoveAdditive 执行时终端机构移动的距离为 MC_MoveAdditive 给定距离、MC_MoveSuperimposed 剩余位移、MC_MoveSuperimposed 搭配的位移指令剩余位移的总和
	否	◆ MC_MoveSuperimposed 立即被打断 ◆ MC_MoveAdditive 执行时终端机构移动的距离为 MC_MoveSuperimposed 剩余距离和 MC_MoveAdditive 设定距离之和
1~5（等待）	是	◆ MC_MoveSuperimposed 指令不受影响，继续运行 ◆ MC_MoveSuperimposed 指令搭配的位移指令执行完毕后，MC_MoveAdditive 开始执行
	否	◆ MC_MoveSuperimposed 继续执行 ◆ MC_MoveAdditive 立即报错

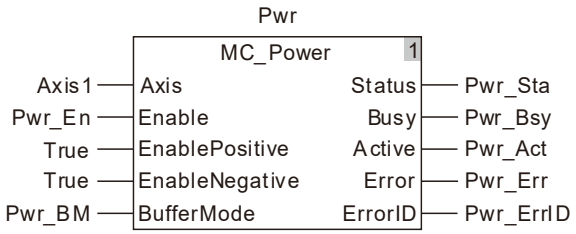
程序范例一

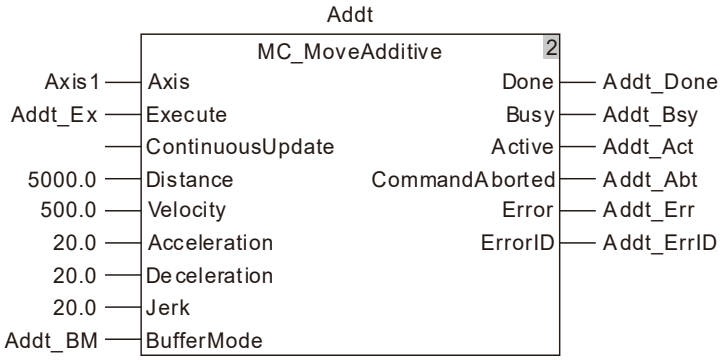
MC_MoveAdditive 指令单独执行时的范例如下所示：

1. 变量和程序

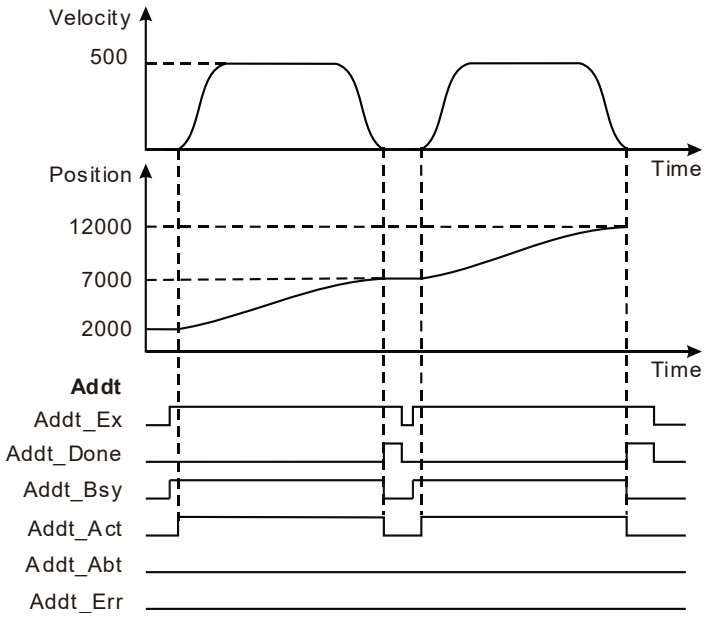
变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	

变量名	数据类型	初始值
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Addt	MC_MoveAdditive	
Addt_Ex	BOOL	FALSE
Addt_BM	MC_Buffer_Mode	0
Addt_Done	BOOL	
Addt_Bsy	BOOL	
Addt_Act	BOOL	
Addt_Abt	BOOL	
Addt_Err	BOOL	
Addt_ErrID	WORD	





2. 运动曲线和时序图



- ❖ 当 Addt_Ex 由 FALSE 变为 TRUE 时，运动控制器控制伺服电机以当前位置为参考点运转，同时 Addt_Bsy 变为 TRUE，Addt_Act 晚一个周期变为 TRUE。当伺服电机完成设定距离后， 完成位 Addt_Done 由 FALSE 变为 TRUE，同时 Addt_Bsy 和 Addt_Act 由 TRUE 变为 FALSE。
- ❖ 当 Addt_Ex 由 TRUE 变为 FALSE 时，完成位 Addt_Done 复位。
- ❖ 伺服电机完成设定距离后，Addt_Ex 再次由 FALSE 变为 TRUE 时，运动控制器控制伺服电机运转，当伺服电机完成设定距离后，完成位 Addt_Done 再次由 FALSE 变为 TRUE。

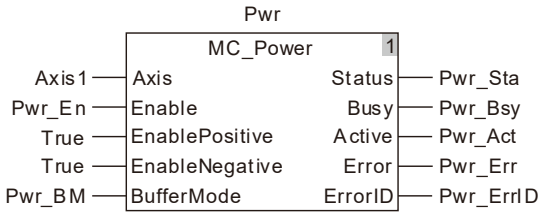
程序范例二

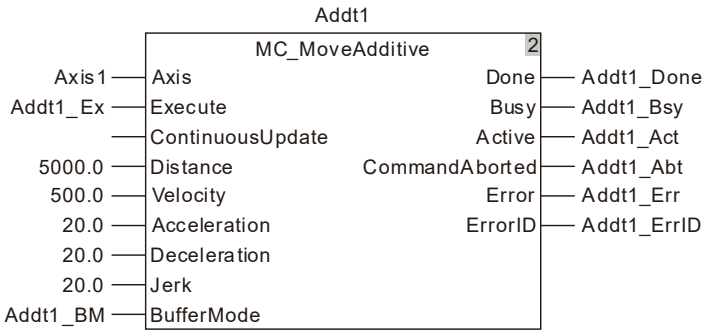
当同一个任务列表里两条附加位移指令搭配使用时，情况如下：

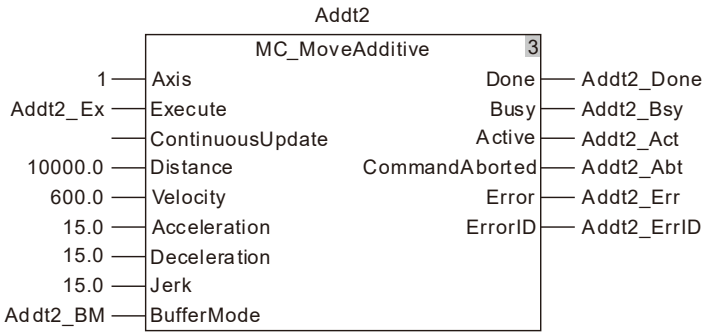
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Addt1	MC_MoveAdditive	
Addt1_Ex	BOOL	FALSE
Addt1_BM	MC_Buffer_Mode	0
Addt1_Done	BOOL	
Addt1_Bsy	BOOL	
Addt1_Act	BOOL	

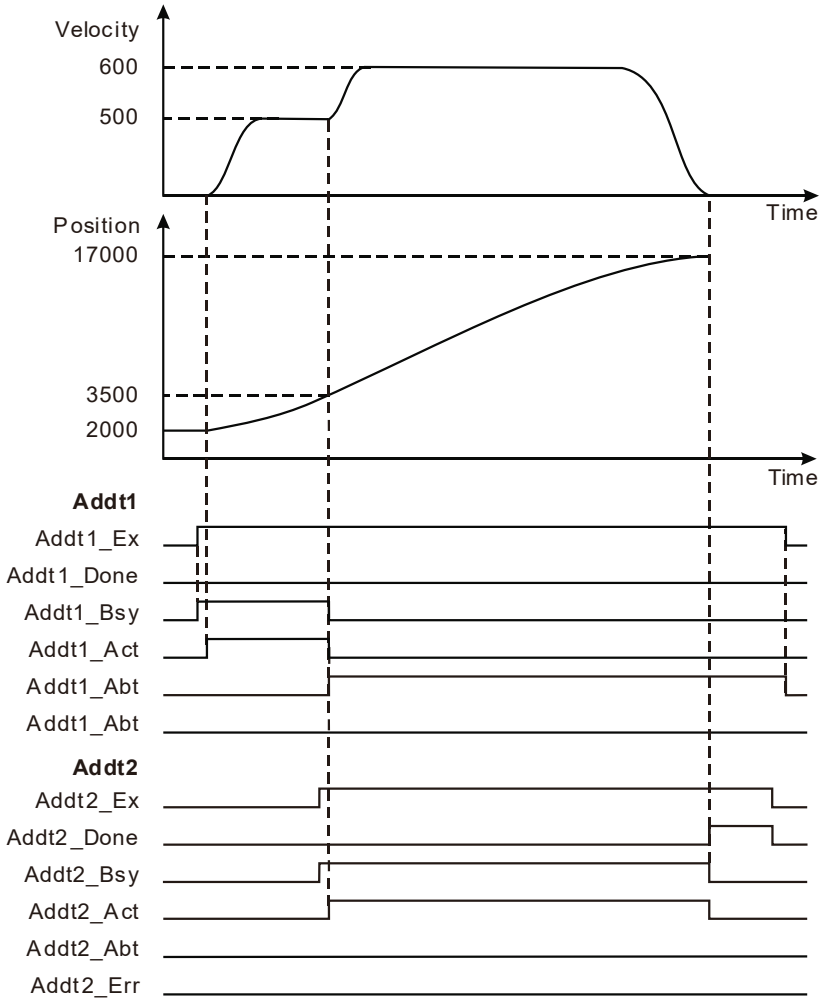
变量名	数据类型	初始值
Addt1_Abt	BOOL	
Addt1_Err	BOOL	
Addt1_ErrID	WORD	
Addt2	MC_MoveAdditive	
Addt2_Ex	BOOL	FALSE
Addt2_BM	MC_Buffer_Mode	0
Addt2_Done	BOOL	
Addt2_Bsy	BOOL	
Addt2_Act	BOOL	
Addt2_Abt	BOOL	
Addt2_Err	BOOL	
Addt2_ErrID	WORD	







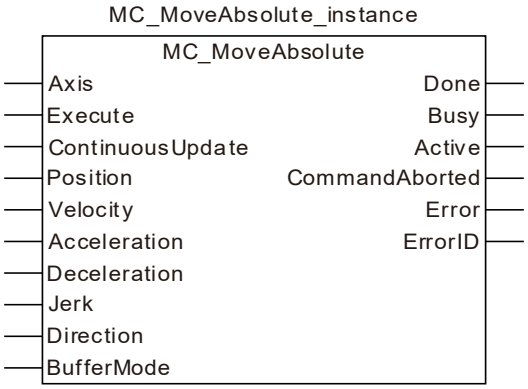
2. 运动曲线和时序图



- ❖ 当 Addt1_Ex 由 FALSE 变为 TRUE 时，运动控制器控制伺服电机以当前位置为参考点运转。当 Addt2_Ex 由 FALSE 变为 TRUE 时，Addt2_Bsy 同时由 FALSE 变为 TRUE，一个周期后第一个附加位置移动指令被中断，中断位 Addt1_Abt 由 FALSE 变为 TRUE。同时，伺服电机以第二个附加位置移动指令的参数进行运动。当伺服电机到达设定距离时 (此时距离为两个指令设定距离总和)，完成位 Addt2_Done 由 FALSE 变为 TRUE。
- ❖ 当 Addt2_Ex 由 TRUE 变为 FALSE 时，完成位 Addt2_Done 复位。

11.3.8 MC_MoveAbsolute（绝对位移指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴按照设定速度，加减速移动到相对于零点的目标位置	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Axis（轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介**（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
ContinuousUpdate	保留	-	-	-
Position（位置）	设定的目标位置 旋转轴：0 ≤ Position<模 直线轴：无限制 （单位: 单元）	LREAL	负数、正数、0（0）	*Execute* 由 FALSE 变为 TRUE
Velocity（速度）	设定的目标速度 （单位: 单元/秒）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Acceleration（加速度）	设定的目标加速度 （单位: 单元/秒^2）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration（减速度）	设定的目标减速度 （单位: 单元/秒^2）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk（加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒^3）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Direction（方向）	设定的运转方向（该参数仅在旋转轴时有效） 1：正向	MC_Direction	1：mcPositiveDirection 2：mcShortestWay 3：mcNegativeDirection 4：mcCurrentDirec	*Execute* 由 FALSE 变为 TRUE 且轴为旋转轴模式

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	2：最短距离 3：反向 4：当前方向		tion （1）	
BufferMode （交接模式）	设定两个指令之间交接模式 0： 打断 1： 等待 2： 以低速交接 3： 以前一指令的速度交接 4： 以后一指令的速度交接 5： 以高速交接	MC_Buffer_Mode	0：mcAborting 1：mcBuffered 2：mcBlendingLow 3：mcBlendingPrevious 4：mcBlendingNext 5：mcBlendingHigh （0）	*Execute* 由 FALSE 变为 TRUE

说明：

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，*Execute* 由 TRUE 变为 FALSE 时，对该指令执行无影响。
2. 当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行；当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行。
3. 当轴为旋转轴时，参数 Position 可以为 0 到模以内但不包括模的值，如果参数 Position 的绝对值大于或等于模，执行此指令会报错；当轴为直线轴时，参数 Position 的大小和模无关，可以设为任意一个常数。
4. 参数 Direction 仅在轴为旋转轴时有效，关于该参数的详细描述请参考该指令功能描述中 Direction 部分。
5. Position、Velocity、Acceleration、Jerk 的关系说明请参考第 10.2 节。
6. 关于 BufferMode 的详细内容，请参考第 10.3 节。

- **输出参数**

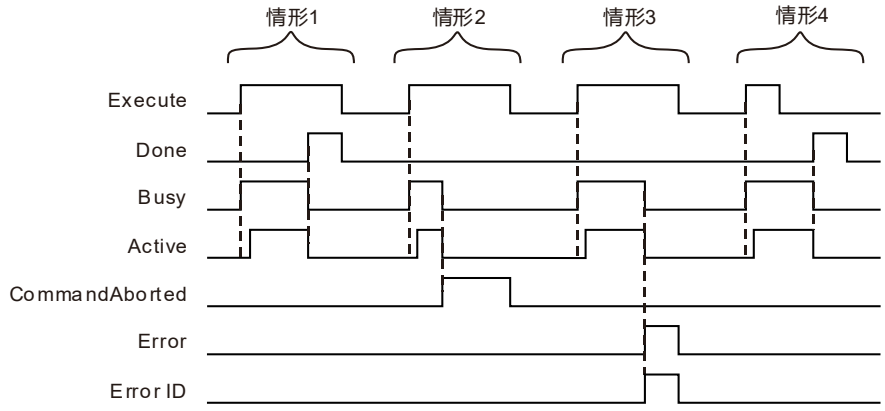
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当定位完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当定位完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE，该指令被其它指令中断时，CommandAborted 变为 TRUE，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，CommandAborted 变为 FALSE。

情形3： 当 Execute 由 FALSE 变为 TRUE，当有错误产生时（如轴报警或者掉线时）， Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

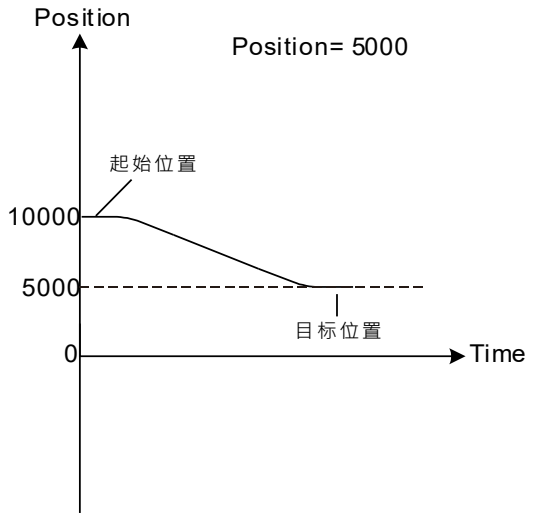
情形4： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

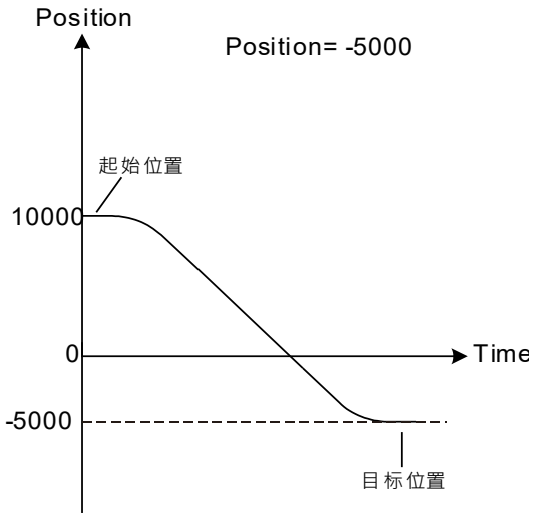
此指令用于控制轴按照设定速度，加减速以及加速度的变化率移动到相对于零点的目标位置。

绝对位移指令执行起点的轴位置为 10000，当 Position>0 （5000）时，轴将反向运动，目标位置为 5000，如左下图所示；当 Position<0（-5000）时，轴将反转，目标位置为 -5000，如右下图所示。

Position>0



Position<0

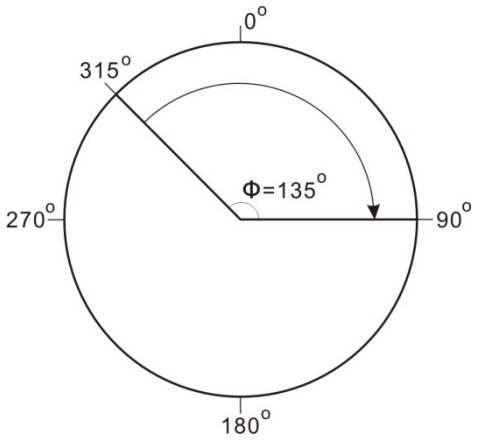


注意：此指令在运行过程中一旦被其它指令终止，剩余未完成的距离将被丢弃，执行新的指令功能。

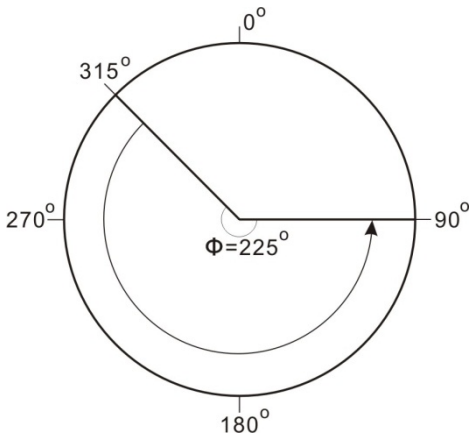
- **Direction**

参数 Direction 仅在轴为旋转轴时生效，其取不同值时，轴的运动方向如下表例子所示（模为 360）：

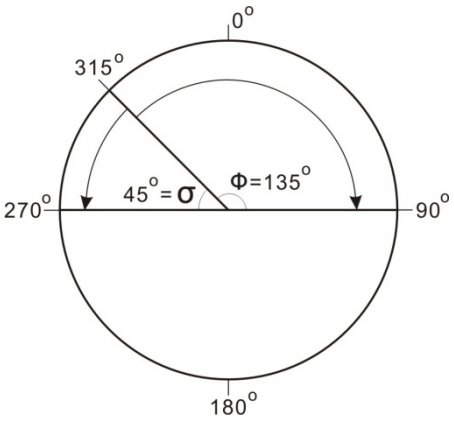
Direction：1（正方向）	Direction：3（反方向）
当前位置：315	当前位置：315
目标位置：90	目标位置：90
移动角度：135°	移动角度：225°



Direction：2（最短）
当前位置：315°
目标位置：90°
移动角度：135°

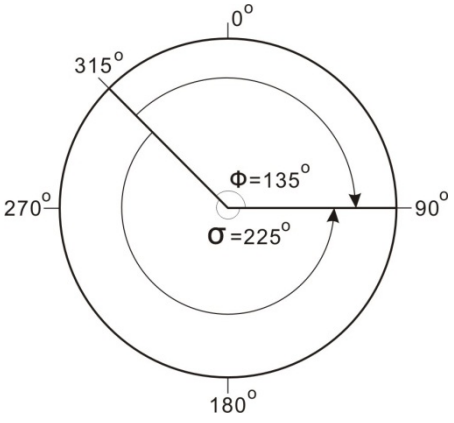


Direction：2（最短）
当前位置：315°
目标位置：270°
移动角度：45°



Direction：4（延续当前的方向）
功能块执行前，旋转轴的状态：正在反转
当前位置：315°
目标位置：90°
移动角度：225°

Direction：4（延续当前的方向）
功能块执行前，轴的状态：静止、正在正转
当前位置：315°
目标位置：90°
移动角度：135°

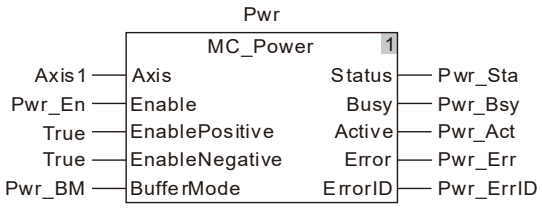


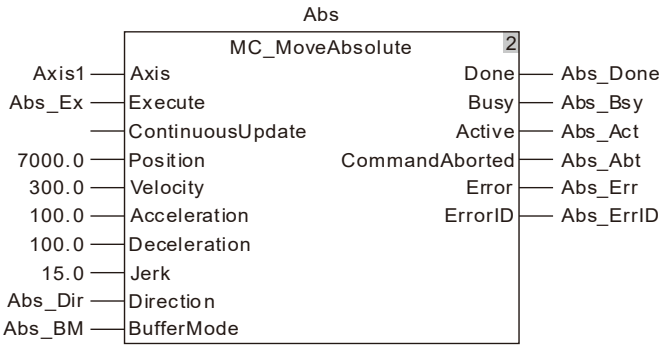
程序范例一

MC_MoveAbsolute 指令单独执行时的范例如下所示：

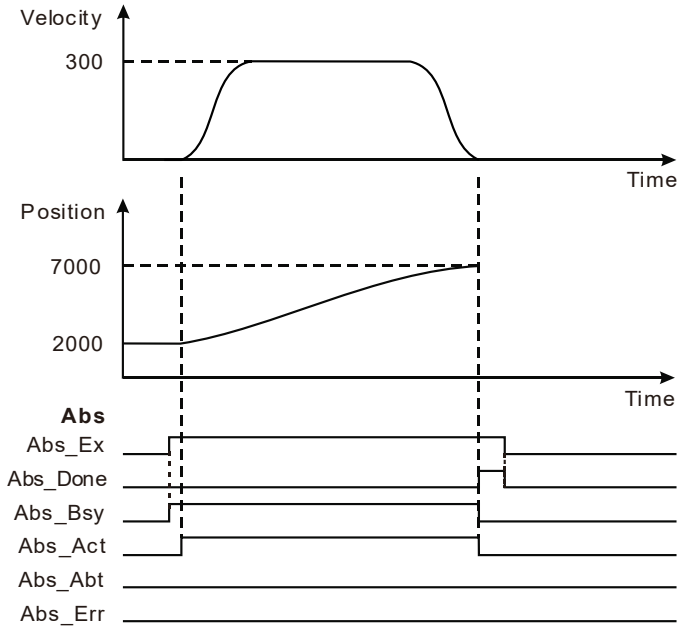
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Abs	MC_Move Absolute	
Abs_Ex	BOOL	FALSE
Abs_Dir	MC_DIRECTION	0
Abs_BM	MC_Buffer_Mode	0
Abs_Done	BOOL	
Abs_Bsy	BOOL	
Abs_Act	BOOL	
Abs_Abt	BOOL	
Abs_Err	BOOL	
Abs_ErrID	WORD	





2. 运动曲线和时序图



- ❖ 当 Abs_Ex 由 FALSE 变为 TRUE 时， MC_MoveAbsolute 指令开始执行，此时轴的当前位置为 2000，目标位置为 7000。
- ❖ 当轴运动到 7000 时，指令执行完成。

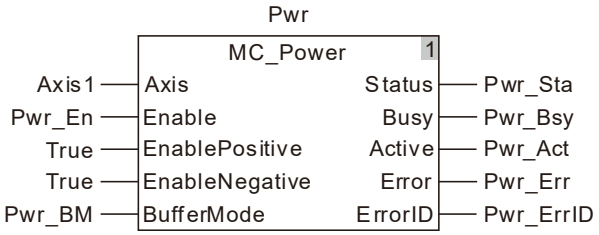
程序范例二

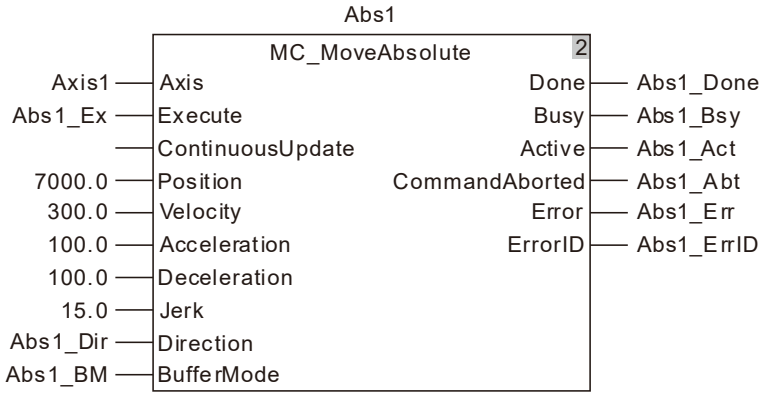
MC_MoveAbsolute 指令打断 MC_MoveAbsolute 指令的范例如下所示：

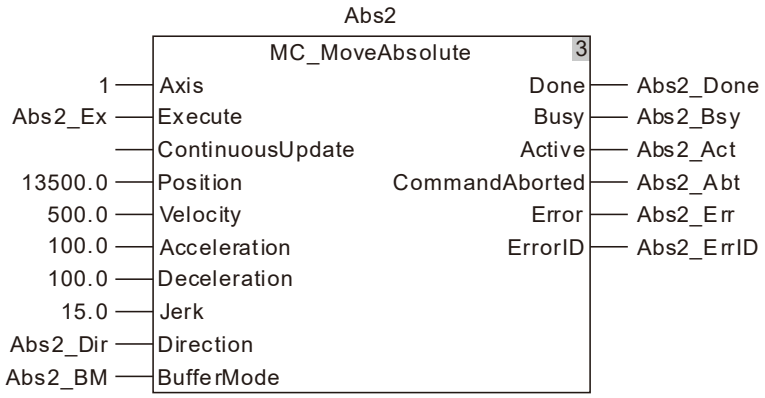
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Abs1	MC_Move Absolute	
Abs1_Ex	BOOL	FALSE
Abs1_Dir	MC_DIRECTION	0
Abs1_BM	MC_Buffer_Mode	0
Abs1_Done	BOOL	
Abs1_Bsy	BOOL	
Abs1_Act	BOOL	
Abs1_Abt	BOOL	
Abs1_Err	BOOL	

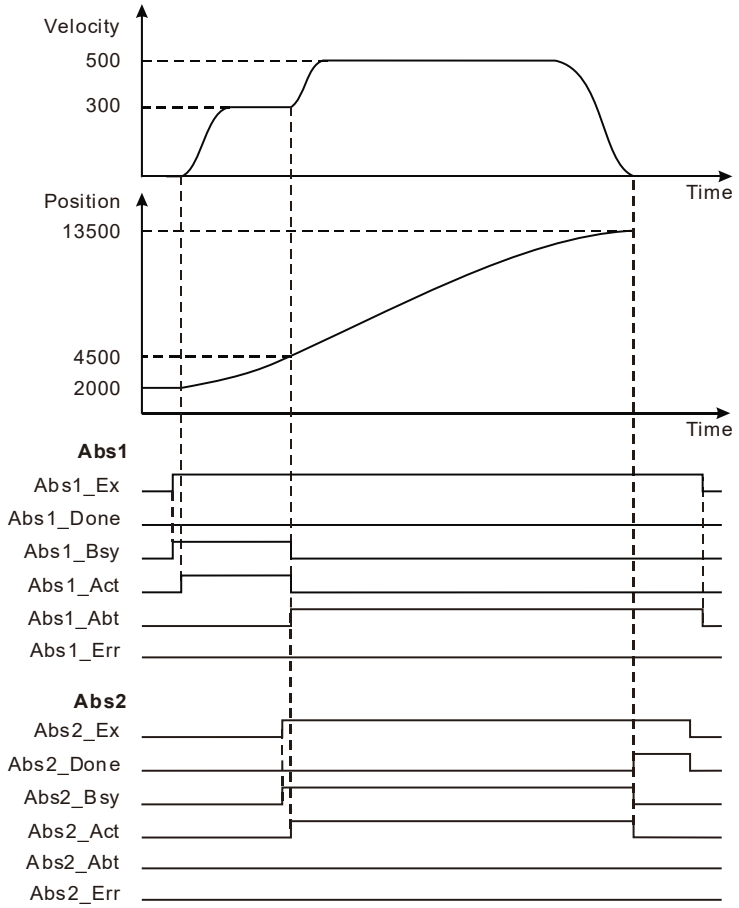
变量名	数据类型	初始值
Abs1_ErrID	WORD	
Abs2	MC_Move Absolute	
Abs2_Ex	BOOL	FALSE
Abs2_Dir	MC_DIRECTION	0
Abs2_BM	MC_Buffer_Mode	0
Abs2_Done	BOOL	
Abs2_Bsy	BOOL	
Abs2_Act	BOOL	
Abs2_Abt	BOOL	
Abs2_Err	BOOL	
Abs2_ErrID	WORD	







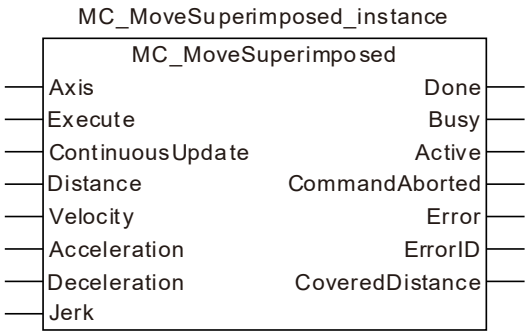
2. 运动曲线和时序图



- ❖ 当 Abs1_Ex 由 FALSE 变为 TRUE 时， 第一个 MC_MoveAbsolute 指令开始执行，此时轴的当前位置为 2000，目标位置为 7000。
- ❖ 当轴位置到达 4500 时，Abs2_Ex 由 FALSE 变为 TRUE，第二个 MC_MoveAbsolute 指令开始执行，且第一个 MC_MoveAbsolute 指令的执行被中断，其输出参数 Abs1_Abt 变为 TRUE。
- ❖ 当轴位置到达 13500 时，第二个 MC_MoveAbsolute 指令执行完成，其输出参数 Abs2_Done 变为 TRUE。

11.3.9 MC_MoveSuperimposed（追加位移指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴在当前运动状态下按设定速度、加减速独立的追加一段设定距离。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Axis（轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介**（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
ContinuousUpdate	保留	-	-	-
Distance（距离）	轴的追加距离（单位: 单元）	LREAL	负数、正数、0（0）	*Execute* 由 FALSE 变为 TRUE
Velocity（速度）	设定的目标速度（单位: 单元/秒）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Acceleration（加速度）	设定的目标加速度（单位: 单元/秒 ^{2}）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration（减速度）	设定的目标减速度（单位: 单元/秒 ^{2}）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk（加速度的变化率）	设定的目标加速度或减速度的变化率（单位: 单元/秒 ^{3}）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE

说明：

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，*Execute* 由 TRUE 变为 FALSE 时，对该指令执行无影响。
2. 当指令正在执行中，Execute 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行；当指令执行完成后，Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行。

3. Velocity、Acceleration、Deceleration、Jerk 的关系说明请参考第 10.2 节。

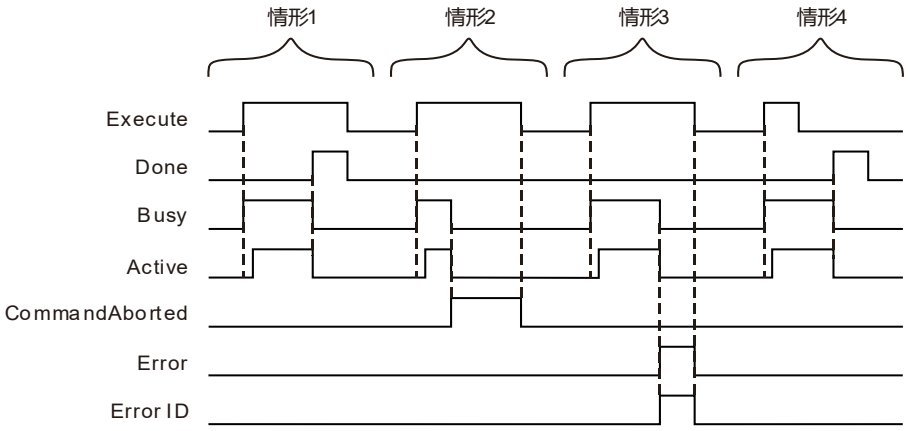
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
CoveredDistance （累计追加距离）	该指令自启动后所累计追加的距离。	LREAL	负数、正数、0

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当追加位移完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当追加位移完成后，*Done* 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 *Execute* 为 TRUE 时，该指令被其它指令中断，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 为 TRUE 时，当有错误产生时（如轴去使能），*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，*Error* 变为 FALSE，ErrorID 值清 0。

情形4： 在指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

此指令用于控制轴在当前运动状态下按设定速度，加减速独立的追加一段设定距离。

1. MC_MoveSuperimposed 指令执行时，不会终止前一个指令（不包括 MC_MoveSuperimposed 和 MC_HaltSuperimposed 指令）的执行，两条指令同时执行，距离、速度、加减速会实时叠加（当某个指令到达设定速度时，其加速度为 0）。当前一个指令执行完毕，将不再叠加其速度、加减速，MC_MoveSuperimposed 指令仍然独立运行。
2. 当轴状态为 Standstill 时，执行 MC_MoveSuperimposed 指令，MC_MoveSuperimposed 指令作用等同于 MC_MoveRelative 指令。
3. MC_MoveSuperimposed 指令和运动指令共同控制轴时，再执行其它的运动指令（不包括 MC_MoveSuperimposed 和 MC_HaltSuperimposed 指令）如果后执行的运动指令的 Buffermode=0，则 MC_MoveSuperimposed 指令和先执行的运动指令都会被中断；如果后执行的运动指令的 Buffermode 为其它值时，则 MC_MoveSuperimposed 指令和先执行的运动指令都不会被中断。
4. MC_MoveSuperimposed 指令和其它运动指令共同控制轴时，再执行另一个 MC_MoveSuperimposed 指令，则前一个 MC_MoveSuperimposed 指令被中断 ，但是其它运动指令不受影响。
5. MC_MoveSuperimposed 指令单独控制轴时，再执行另一个 MC_MoveSuperimposed 指令，则前一个 MC_MoveSuperimposed 指令被中断 。
6. MC_MoveSuperimposed 指令执行中，再执行 MC_HaltSuperimposed 指令，则 MC_MoveSuperimposed 指令被中断 。

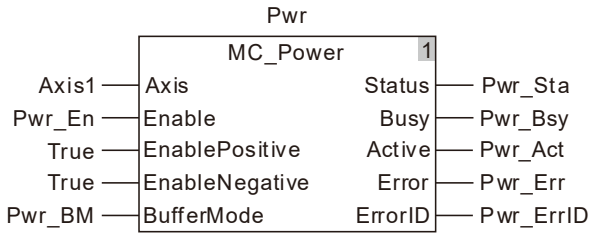
7. 可以对电子齿轮（MC_GearIn）和电子凸轮（MC_ CamIn）的从轴执行 MC_MoveSuperimposed 指令。

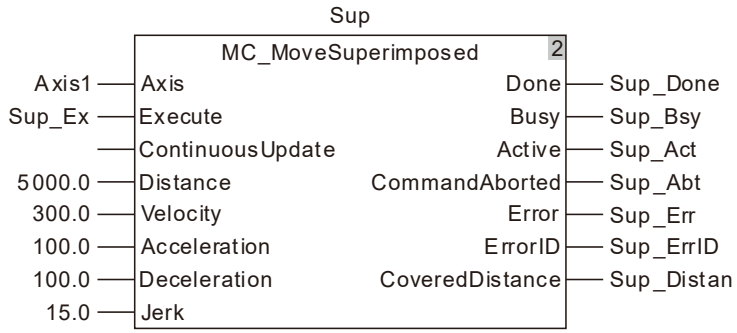
程序范例一

MC_MoveSuperimposed 指令单独执行时的范例如下所示：

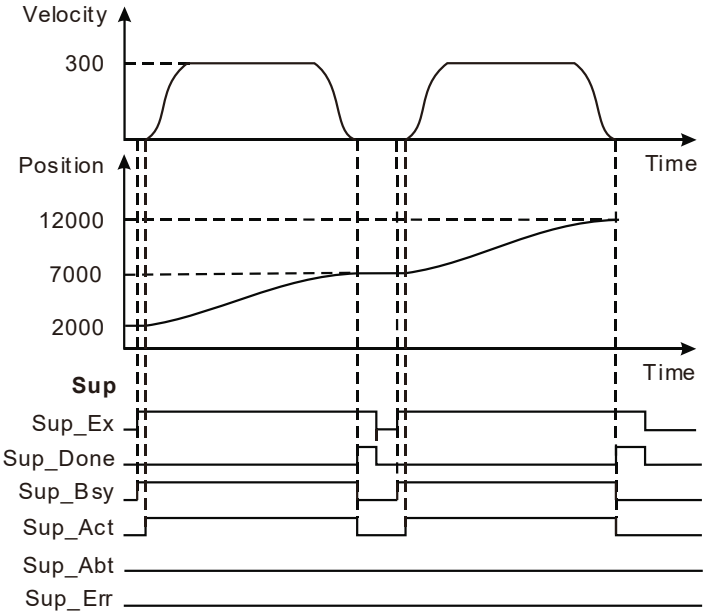
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Sup	MC_MoveSuperimposed	
Sup_Ex	BOOL	FALSE
Sup_Done	BOOL	
Sup_Bsy	BOOL	
Sup_Act	BOOL	
Sup_Abt	BOOL	
Sup_Err	BOOL	
Sup_ErrID	WORD	
Sup_Distan	LREAL	





2. 运动曲线和时序图：



- 当 Sup_Ex 变为 TRUE 时，Sup_Bsy 变为 TRUE，一个周期后，Sup_Act 变为 TRUE，运动控制器控制伺服电机以当前位置为参考点运转。
- 伺服电机完成设定距离后，Sup_Done 变为 TRUE，同时 Sup_Bsy 和 Sup_Act 变为 FALSE。
- 当 Sup_Ex 变为 FALSE 时，Sup_Done 变为 FALSE。
- 伺服电机完成设定距离后，Sup_Ex 再次变为 TRUE 时，运动控制器控制伺服电机运转，当伺服电机完成设定距离后，完成位 Sup_Done 再次变为 TRUE。

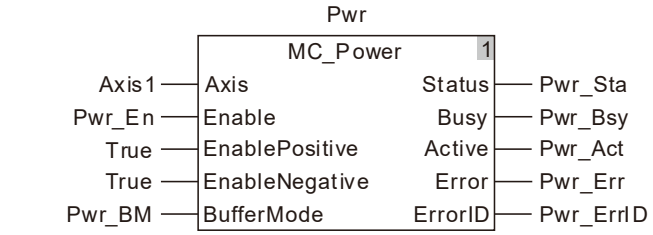
程序范例二

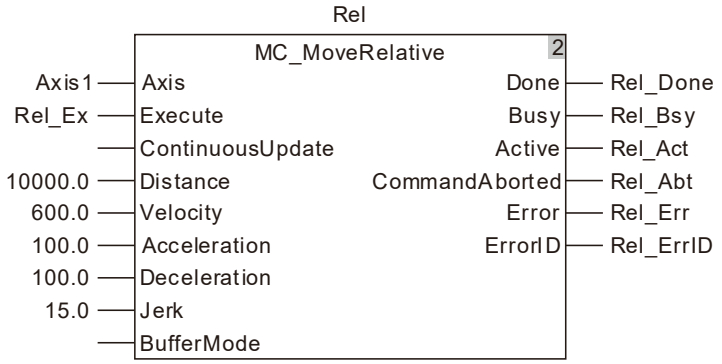
MC_MoveSuperimposed 和 MC_MoveRelative 指令搭配范例如下所示：

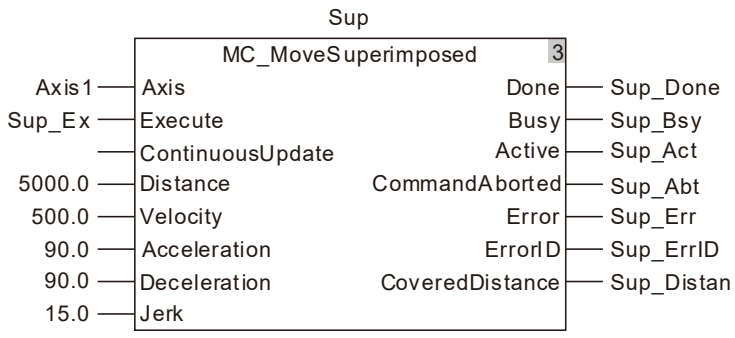
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	

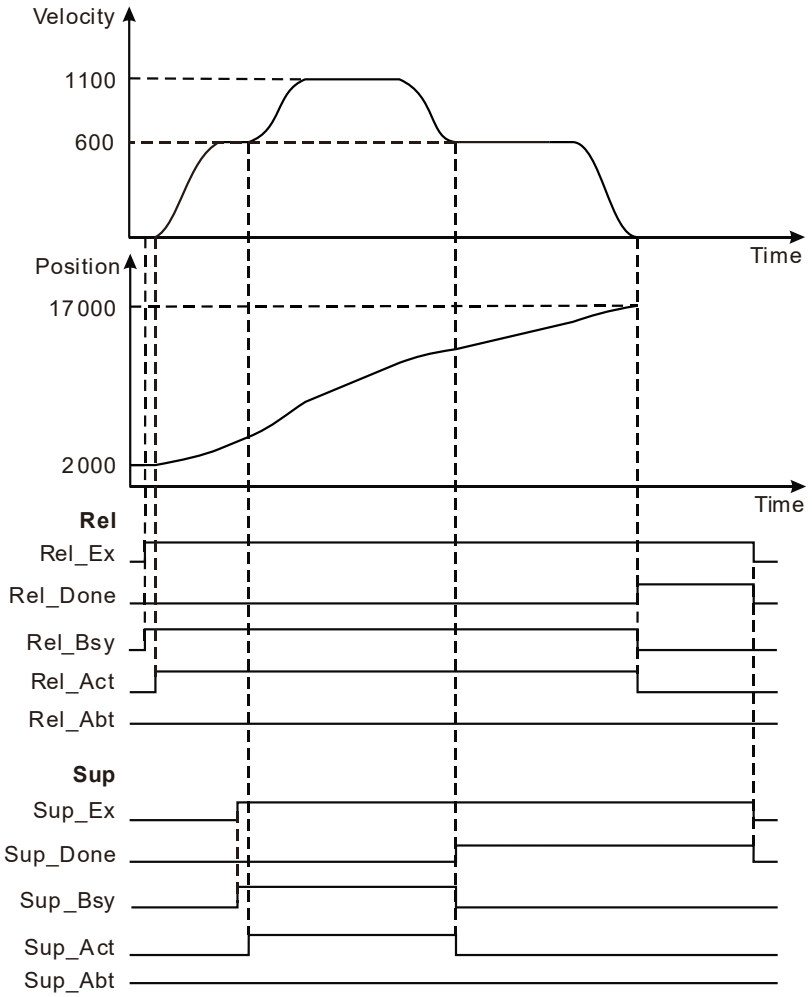
变量名	数据类型	初始值
Rel_Err	BOOL	
Rel_ErrID	WORD	
Sup	MC_MoveSuperimposed	
Sup_Ex	BOOL	FALSE
Sup_Done	BOOL	
Sup_Bsy	BOOL	
Sup_Act	BOOL	
Sup_Abt	BOOL	
Sup_Err	BOOL	
Sup_ErrID	WORD	
Sup_Distan	LREAL	







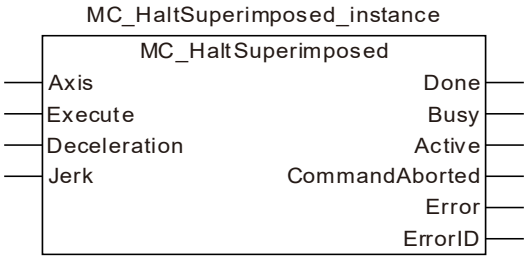
2. 运动曲线和时序图：



- 当 Rel_Ex 变为 TRUE 时，Rel_Bsy 变为 TRUE，一个周期后 Rel_Act 变为 TRUE，运动控制器控制伺服电机以当前位置为参考点运转。
- 当 Sup_Ex 变为 TRUE 时，Sup_Bsy 变为 TRUE，一个周期后,Sup_Act 变为 TRUE，MC_MoveSuperimposed 指令开始执行，伺服电机的速度以及加速度会（此时加速度为 0）叠加。
- 当 MC_MoveSuperimposed 指令追加距离完成时，Sup_Done 变为 TRUE，Sup_Bsy 和 Sup_Act 变为 FALSE。
- 当 MC_MoveRelative 指令的位置完成时，Rel_Done 变为 TRUE，Rel_Bsy 和 Rel_Act 变为 FALSE。轴的最终位置为两个指令设定位置的总和加起始位置。
- 当 Rel_Ex 变为 FALSE 时，Rel_Done 变为 FALSE。当 Sup_Ex 变为 FALSE 时， Sup_Done 变为 FALSE。

11.3.10 MC_HaltSuperimposed（暂停追加指令）

FB/FC	说明	适用机种
FB	此指令用于暂停追加位移指令。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk （减速度的变化率）	设定的目标减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE

说明：

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行。该指令正在执行，*Execute* 由 TRUE 变为 FALSE 时，对该指令执行无影响。
2. *Deceleration*、*Jerk* 的关系说明请参考第 10.2 节。

- **输出参数**

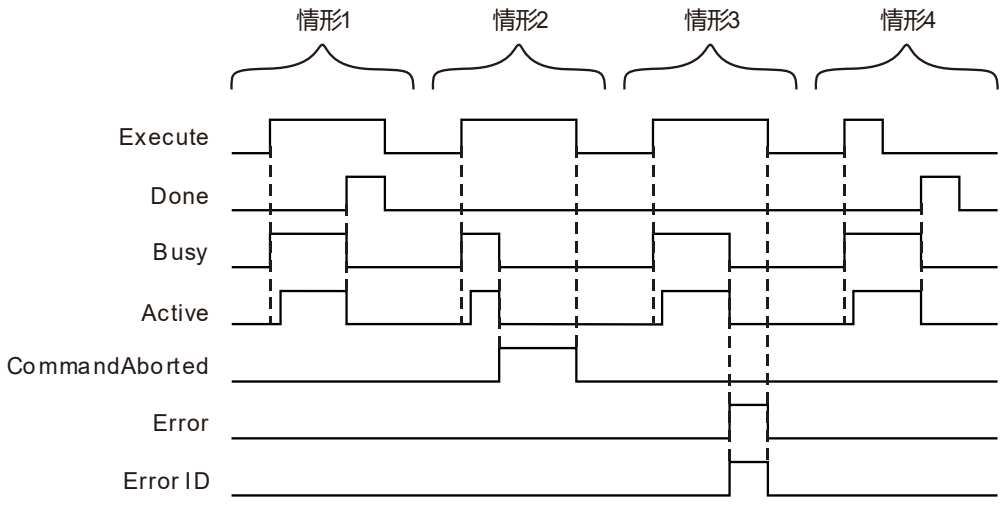
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当暂停追加位移完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当暂停追加位移完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 为 TRUE 时，该指令被其它指令中断后，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 为 TRUE 时，当有错误产生时（如轴去使能），*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 后，*Error* 变为 FALSE。

情形4： 在指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

此指令用于暂停追加位移指令。

- MC_HaltSuperimposed 指令不能单独执行，只能和 MC_MoveSuperimposed 配合使用。
- MC_MoveSuperimposed 和其它运动指令共同控制轴时，再执行 MC_HaltSuperimposed 指令，MC_HaltSuperimposed 指令会中断 MC_MoveSuperimposed 指令，但是其它运动指令的执行不受影响。
- MC_HaltSuperimposed 指令可以中断 MC_HaltSuperimposed 指令。

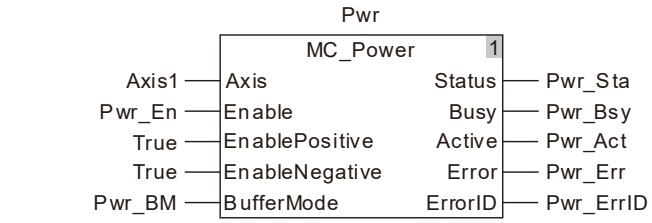
程序范例

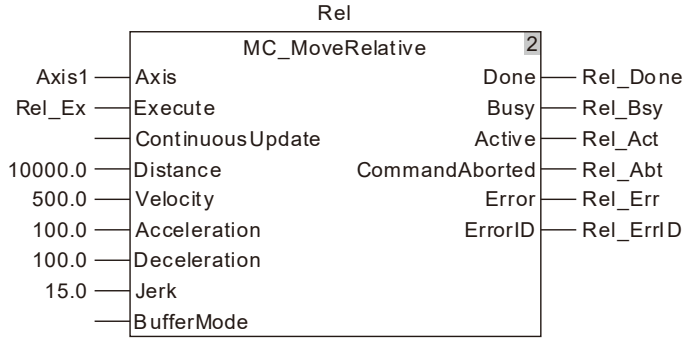
MC_HaltSuperimposed 指令单独执行时的范例如下所示：

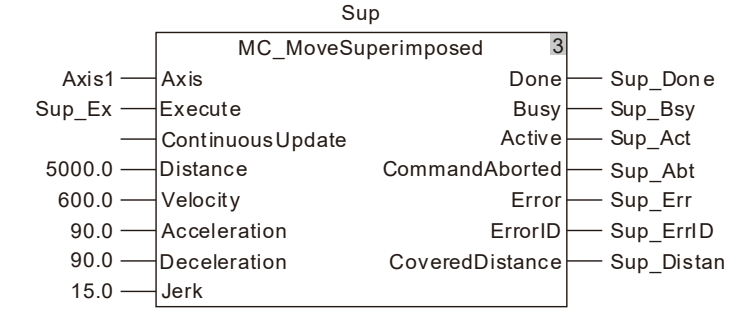
1. 变量和程序

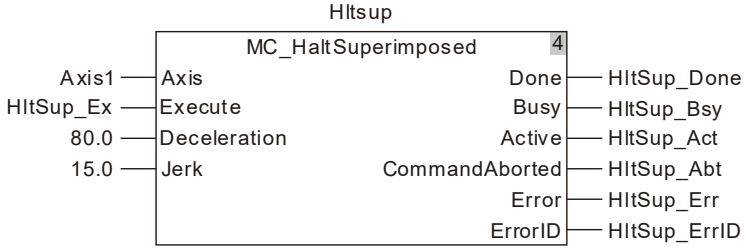
变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Sup	MC_MoveSuperimposed	
Sup_Ex	BOOL	FALSE
Sup_Done	BOOL	

变量名	数据类型	初始值
Sup_Bsy	BOOL	
Sup_Act	BOOL	
Sup_Abt	BOOL	
Sup_Err	BOOL	
Sup_ErrID	WORD	
Sup_Distan	LREAL	
HltSup	MC_HaltSuperimposed	
HltSup_Ex	BOOL	FALSE
HltSup_Done	BOOL	
HltSup_Bsy	BOOL	
HltSup_Act	BOOL	
HltSup_Abt	BOOL	
HltSup_Err	BOOL	
HltSup_ErrID	WORD	

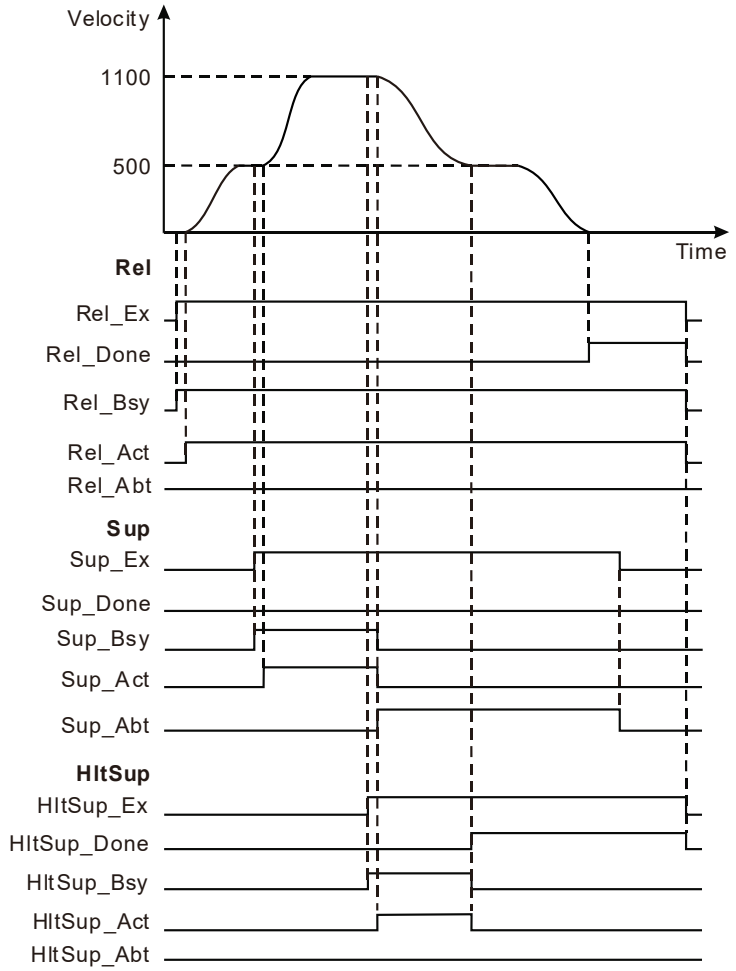








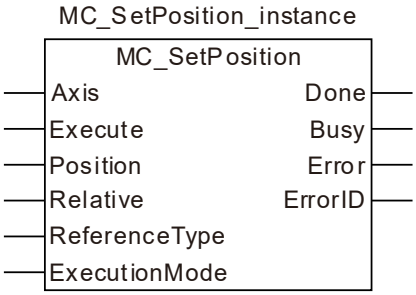
2. 运动曲线和时序图：



- 当 Rel_Ex 变为 TRUE 时，Rel_Bsy 变为 TRUE，一个周期后，变为 TRUE，运动控制器控制伺服电机以当前位置为参考点运转。当 Sup_Ex 变为 TRUE 时，Sup_Bsy 变为 TRUE，一个周期后，Sup_Act 变为 TRUE，MC_MoveSuperimposed 指令开始执行，伺服电机的速度以及加速度会（此时轴加速度为 0）叠加。
- 当 Hltsup_Ex 变为 TRUE 时，Hltsup_Bsy 变为 TRUE，一个周期后，Hltsup_Act 变为 TRUE，MC_HaltSuperimposed 指令开始执行，MC_MoveSuperimposed 指令被中断，Sup_Bsy、Sup_Act 变为 FALSE，同时 Sup_Abt 变为 TRUE。MC_HaltSuperimposed 指令中断 MC_MoveSuperimposed 指令的执行。
- 当 Hltsup_Done 变为 TRUE 时，Hltsup_Bsy 和 Hltsup_Act 变为 FALSE。
- MC_HaltSuperimposed 指令的执行不影响 MC_MoveRelative 指令的执行。

11.3.11 MC_SetPosition（位置设置指令）

FB/FC	说明	适用机种
FB	此指令用于将轴的位置值设定成指定值，且不引起该轴产生实际运动。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
Position （位置）	设定的目标位置 （单位: 单元）	LREAL	负数、正数、0 （0）	*Execute* 由FALSE 变为 TRUE
Relative （相对模式）	设定目标位置与当前位置为相对模式或绝对模式	BOOL	TRUE 或 FALSE （FALSE）	*Execute* 由FALSE 变为 TRUE
ReferenceType （参考位置的类型）	设定参考位置的类型	MC_ ReferenceType	0：mcCommand Position 1：mcActualPosition （0）	*Execute* 由FALSE 变为 TRUE
ExecutionMode	保留	-	-	-

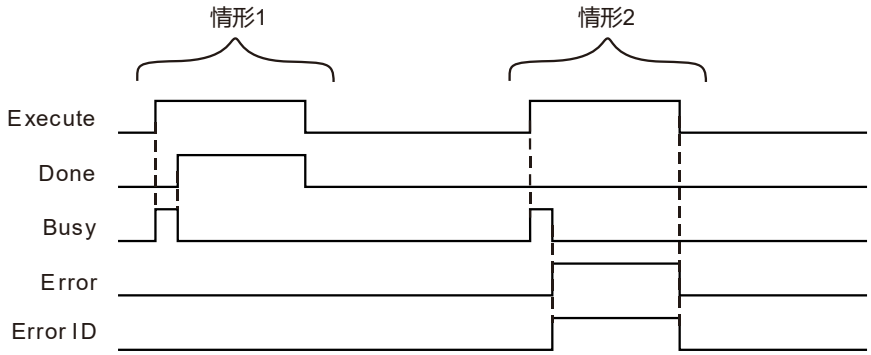
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当定位完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**

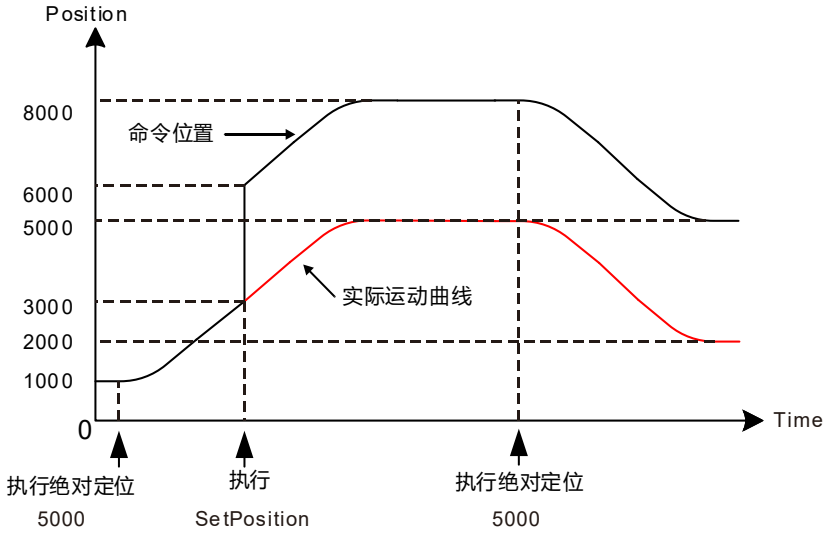


情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，同时 *Busy* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 时，如有错误产生， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

- **功能说明**

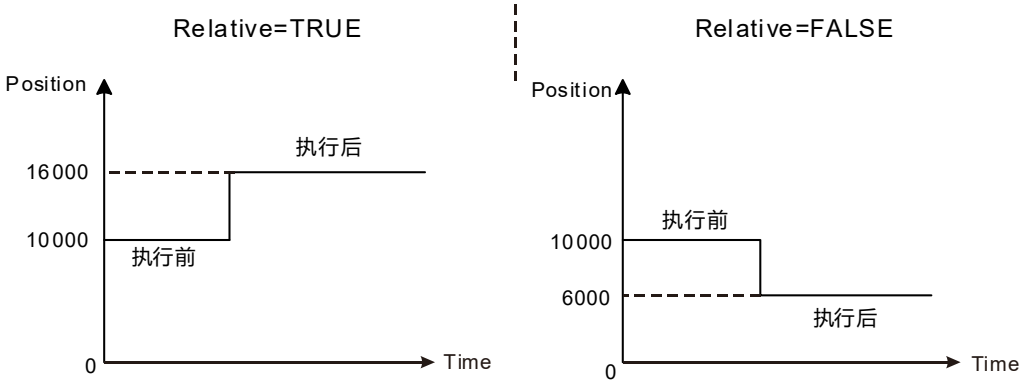
此指令用于将轴的位置值设定成指定值，且不引起该轴产生实际运动。此指令的执行对正在执行的运动不会产生实际影响，但对此指令执行完成后再开始执行的指令的实际执行效果有影响，如下图所示：



■ **Position 与 Relative**

输入参数 *Position*、*Relative* 与指令开始执行时刻的轴位置（本文使用“参考位置”表示）共同决定欲设置的位置值。

输入参数 *Relative* 用于定义输入参数 *Position* 与参考位置之间的关系。当 *Relative*=TRUE 时，*Position* 与参考位置为相对关系，位置设置值=参考位置+*Position*；当 *Relative*=FALSE 时，*Position* 与参考位置为绝对关系，位置设置值=*Position*。如下图所示，指令执行时的参考位置为 10000，输入参数 *Position* 的值为 6000，当输入参数 *Relative* 为不同值时，对应的执行效果分别为左下图和右下图。



■ **ReferenceType**

输入参数 ReferenceType 用于选择参考位置为命令位置或实际位置。ReferenceType=0 时，参考位置为该轴命令位置；ReferenceType=1 时，参考位置为该轴实际位置。

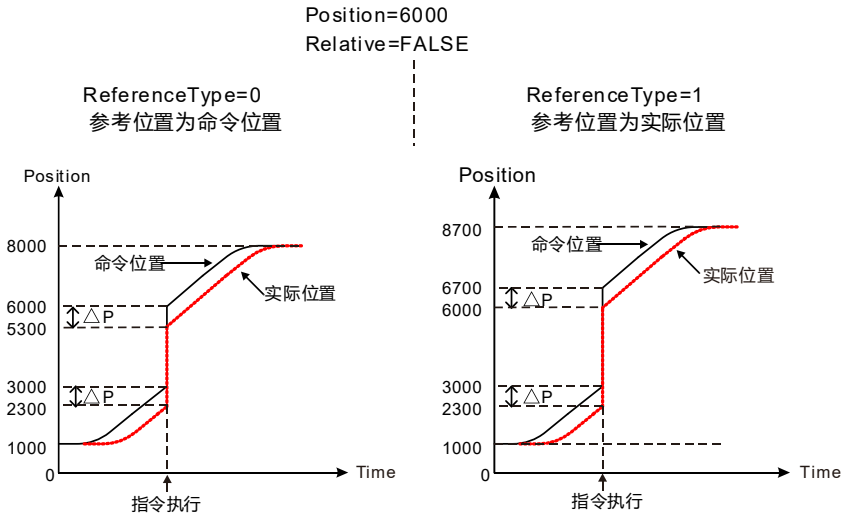
当参考位置选择为命令位置时，指令将根据当前的命令位置和 *Position* 计算出目标命令位置，并将命令位置的值修改为该目标位置的值；同时，该轴的实际位置也将随之变化，变化规律为：实际位置的变化量与命令位置的变化量相等，也即是命令位置与实际位置之间的差值在指令开始执行时刻和指令执行完成时刻保持不变。

参考位置选择为实际位置时的处理方式与参考位置选择为命令位置时的处理方式同理。

若执行 MC_SetPosition 指令时，轴处于静止状态，则参考位置分别选择命令位置和实际位置时的实际执行效果没有差别，原因在于轴静止时，其命令位置与实际位置之间不存在差值（差值为 0）；

若执行 MC_SetPosition 指令时，轴处于运动状态，其命令位置与实际位置之间存在差值（差值不为 0，

由命令响应时间引起)，则参考位置分别选择命令位置和实际位置时的实际执行效果存在差异，如下图范例所示：如下图曲线所示，轴正在作定位运动(目标位置为 5000)时执行 MC_SetPosition 指令(绝对模式，Position=6000)，此时该轴的命令位置和实际位置分别为 3000 和 2300(差值▵P =700)。若参考位置选择命令位置，则指令执行后，该轴的命令位置变为 6000，实际位置变为 5300(5300=6000-▵P)，如左下图所示；若参考位置选择实际位置，则指令执行后，该轴的实际位置变为 6000，命令位置变为 6700(6700=6000+▵P)，如右下图所示。



■ **轴类型与 ReferenceType 的适用关系**

不同类型的轴所适用的 ReferenceType 如下表所示：

轴类型	ReferenceType	
	命令位置	实际位置
实轴	支持	支持
编码器	支持	支持
虚轴	支持	支持

若 MC_SetPosition 指令操作的轴不支持所选择的 ReferenceType，指令执行时将报错。

■ **指令适用情形说明**

对已建立多轴关系的主轴执行 MC_SetPosition 时，由 MC_SetPosition 引起的主轴位置变化不会影响从轴，也即是主轴的位置在 MC_SetPosition 的作用下发生了变化，但从轴不会因为该变化而动作。当作用于从轴时，从轴的位置发生变化，但其与主轴原有的同步关系不受影响。

若在 MC_Stop 指令执行过程中(未执行完成)执行 MC_SetPosition 指令，MC_SetPosition 指令将报错；若在 MC_Stop 指令执行完成后再执行 MC_SetPosition 指令，则可正常执行。

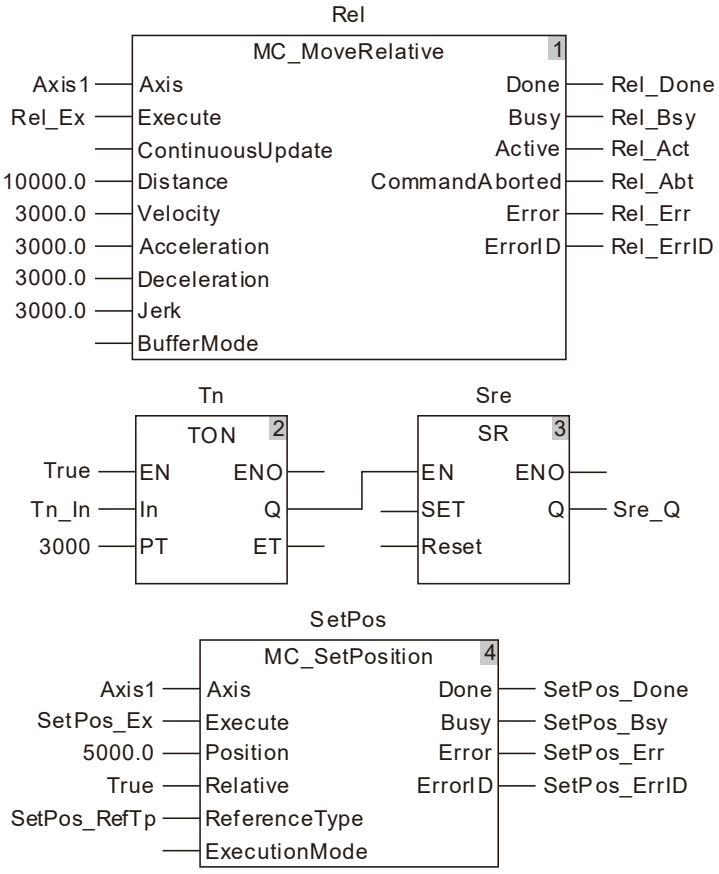
程序范例一

本例描述 MC_SetPosition 指令选择相对模式(*Relative*=TRUE)时，执行该指令对位移指令的影响。

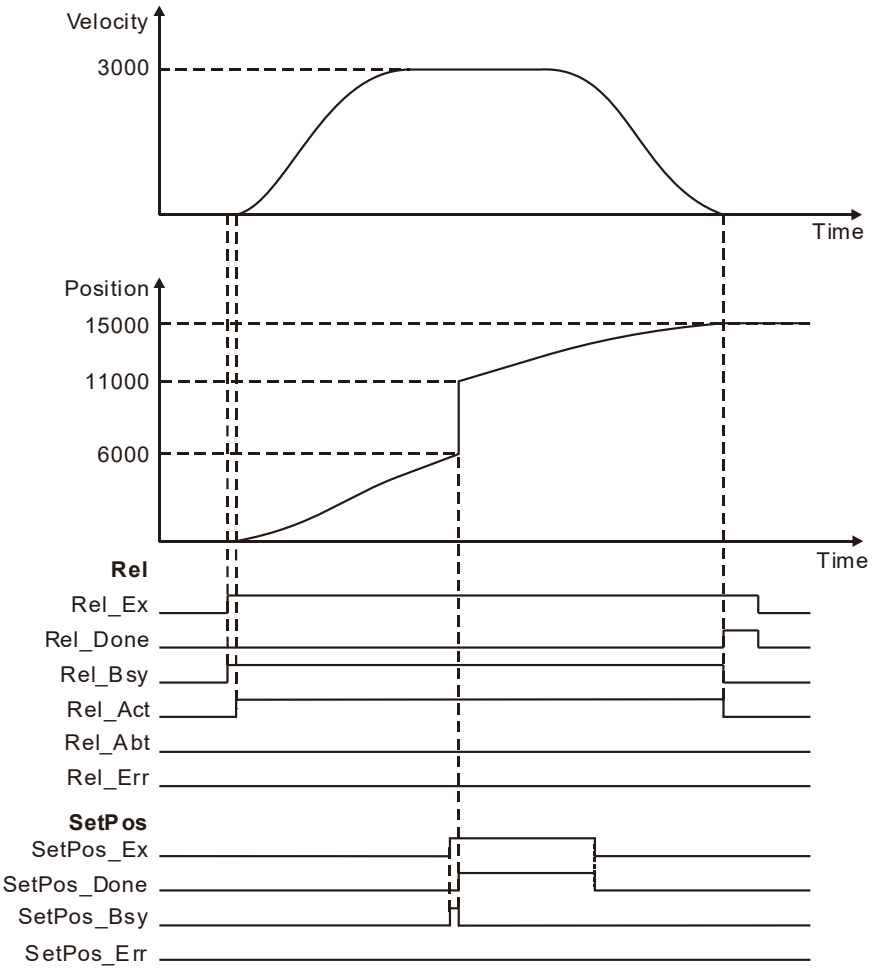
1. 变量和程序

变量名	数据类型	初始值
Rel	MC_MoveRelative	

变量名	数据类型	初始值
Axis1	USINT	1
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Tn	TON	
Tn_In	BOOL	FALSE
SRe	SR	
SRe_Q	BOOL	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNE TYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	



2. 运动曲线和时序图



- ❖ 当 Rel_Ex 由 FALSE 变为 TRUE 时， MC_MoveRelative 指令开始执行，MC_MoveRelative 指令执行 3 秒后 MC_SetPosition 指令执行。
- ❖ MC_SetPosition 指令开始执行时的命令位置为 6000，其执行完成后的命令位置为 11000，其中 11000=6000+5000，MC_MoveRelative 指令执行完成时的位置为 15000。
- ❖ 通过上图中的速度变化曲线可以看出：MC_SetPosition 指令对正在执行的运动不会产生影响。

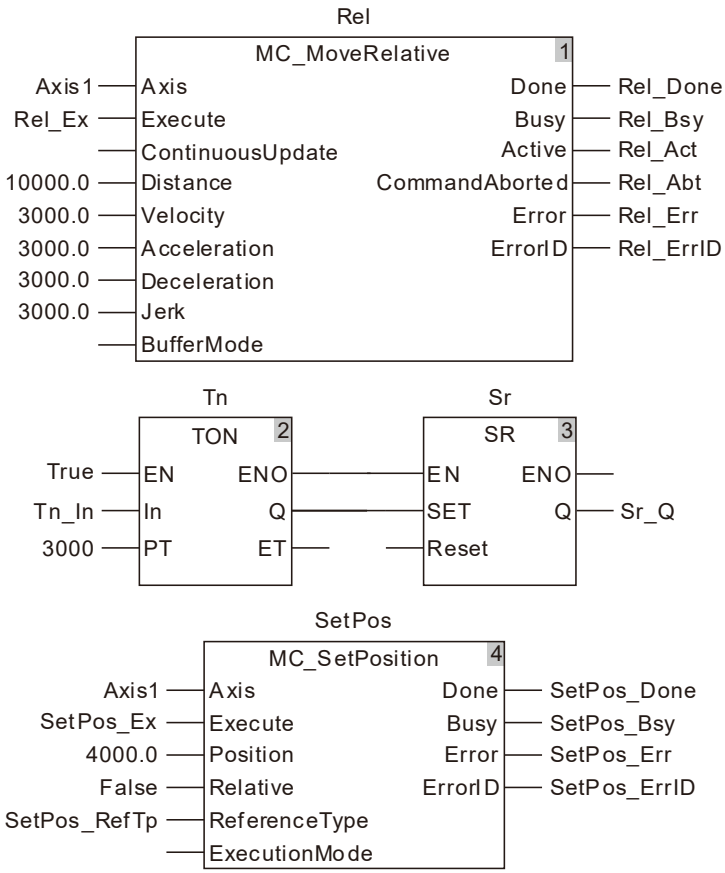
程序范例二

本例描述 MC_SetPosition 指令选择绝对模式（*Relative*=FALSE）时，该指令的执行对轴位置的影响。

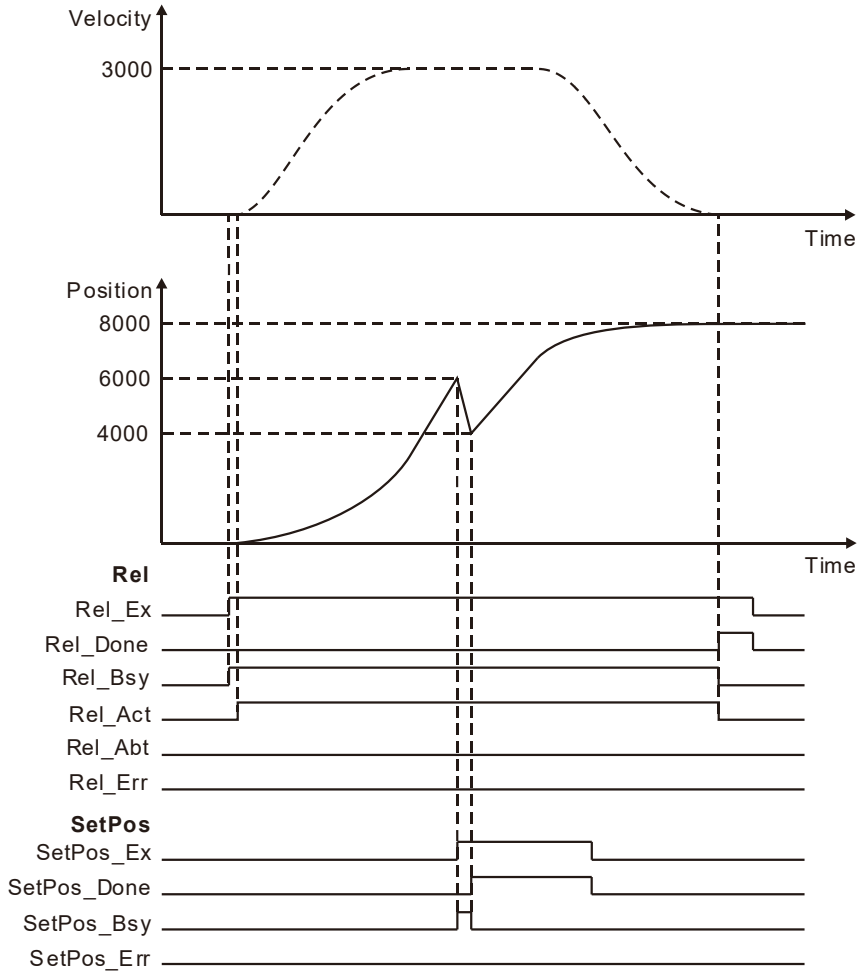
1. 变量和程序

变量名	数据类型	初始值
Rel	MC_MoveRelative	
Axis1	USINT	1
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	

变量名	数据类型	初始值
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Tn	TON	
Tn_In	BOOL	FALSE
SRe	SR	
SRe_Q	BOOL	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNETYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	



2. 运动曲线和时序图



- ❖ 当 Rel_Ex 由 FALSE 变为 TRUE 时， MC_MoveRelative 指令开始执行，MC_MoveRelative 指令执行 3 秒后 MC_SetPosition 指令执行。
- ❖ MC_SetPosition 指令开始执行时的命令位置为 6000，其执行完成后的命令位置为 4000，MC_MoveRelative 指令执行完成时的位置为 8000。
- ❖ 通过上图中的速度变化曲线可以看出：MC_SetPosition 指令对正在执行的运动不会产生影响。

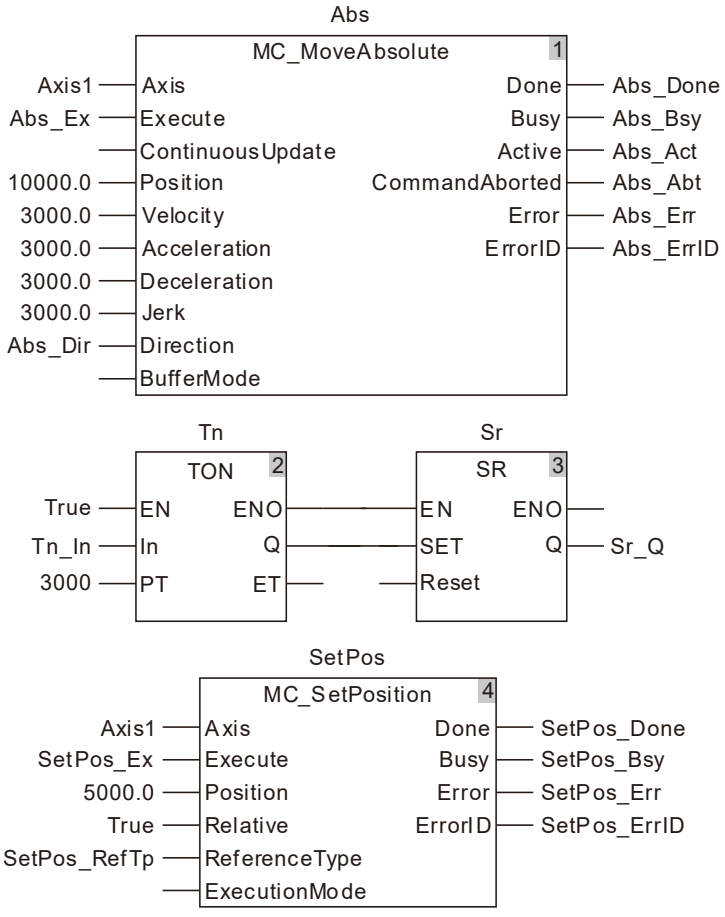
程序范例三

本例描述 MC_SetPosition 指令对正在执行的 MC_MoveAbsolute 指令的执行效果的影响：MC_SetPosition 对正在执行的 MC_MoveAbsolute 指令的实际执行效果无影响。

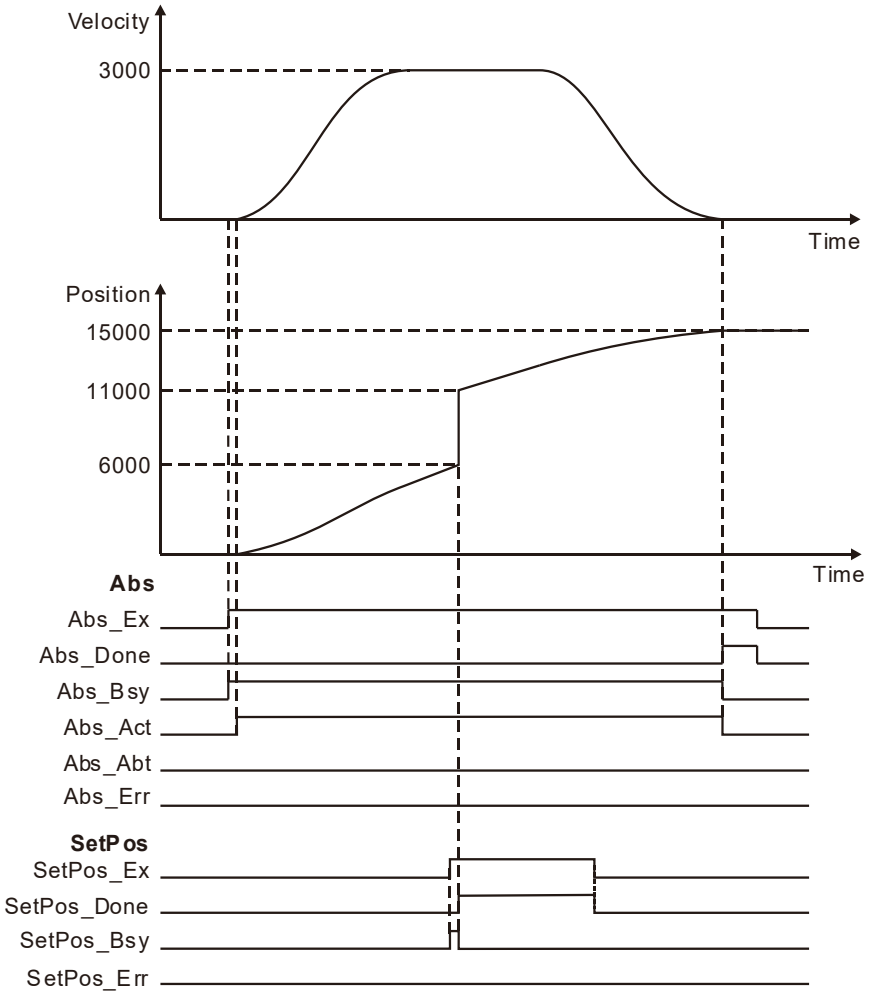
1. 变量和程序

变量名	数据类型	初始值
Abs	MC_MoveAbsolute	
Axis1	USINT	1
Abs_Ex	BOOL	FALSE
Abs_Dir	MC_DIRECTION	1
Abs_Done	BOOL	

变量名	数据类型	初始值
Abs_Bsy	BOOL	
Abs_Act	BOOL	
Abs_Abt	BOOL	
Abs_Err	BOOL	
Abs_ErrID	WORD	
Tn	TON	
Tn_In	BOOL	FALSE
SRe	SR	
SRe_Q	BOOL	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNETYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	



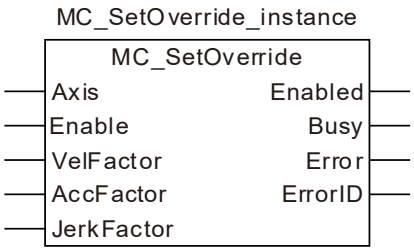
2. 运动曲线和时序图



- 当 Abs_Ex 由 FALSE 变为 TRUE 时， MC_MoveAbsolute 指令开始执行，MC_MoveAbsolute 指令执行 3 秒后 MC_SetPosition 指令执行。
- MC_SetPosition 指令开始执行时的命令位置为 6000，其执行完成后的命令位置为 11000，MC_MoveAbsolute 指令执行完成时的位置为 15000。
- 通过上图中的速度变化曲线可以看出：MC_SetPosition 指令对正在执行的 MC_MoveAbsolute 的实际执行效果不会产生影响。

11.3.12 MC_SetOverride（超调值设置指令）

FB/FC	说明	适用机种
FB	此指令用于变更轴的目标速度	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Enable* 为 TRUE 时
Enable （执行位）	当 *Enable* 由 FALSE 变为 TRUE 时，指令开始执行	BOOL	TRUE 或 FALSE （FALSE）	-
VelFactor （速度超调值）	速度超调值 （单位：%）	LREAL	0~500 （100）	*Enable* 为 TRUE 时
AccFactor	保留	-	-	-
JerkFactor	保留	-	-	-

- **输出参数**

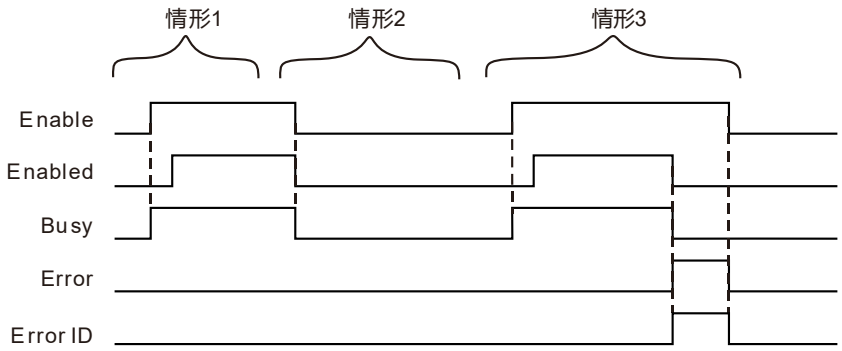
名称	功能	数据类型	输出范围
Enabled （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码， 对应的错误代码请参考第 12.2 节	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Enabled	◆ 当开始执行指令时	◆ 当 Enable 变为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 变为 FALSE 时 ◆ 当 Error 为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Enable* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，指令执行成功后，*Enabled* 变为 TRUE。

情形2： 当 *Enable* 由 TRUE 变为 FALSE 时，*Enabled* 和 *Busy* 同时变为 FALSE。

情形3： 当 *Enable* 由 FALSE 变为 TRUE 时，当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Enalbed* 和 *Busy* 变为 FALSE；当 *Enable* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

- **功能说明**

此指令用于变更轴的目标速度。

1. 要临时变更各指令的目标速度时，使用本指令。因此，对于输入不带目标速度的指令，本指令不起作用。但是，对于本指令无效的指令，即使将 MC_SetOverride 指令设为有效，*Enabled* 也会继续保持 TRUE。
2. 可变更目标速度的指令如下：

MC_MoveAbsolute（绝对位移指令）	MC_MoveRelative（相对位移指令）
MC_MoveAdditive（附加位移指令）	MC_MoveVelocity（速度指令）
MC_MoveSuperimposed(追加位移指令)	

3. 新的目标速度如下：
   变更后的目标速度 = 当前执行中指令的目标速度 × 速度超调值
4. VelFactor 的单位为 %。“100”表示“100%”。VelFactor 的有效范围为 0~500，超出有效范围时执行 MC_SetOverride 指令，指令会报错。
5. 相对变更后的目标速度，轴会根据当前执行指令的 Acceleration（或 Deceleration）加速（或减速）到变更后的目标速度。
6. 变更后的目标速度超过轴参数的最高转速时，轴会发生错误。
7. 将 VelFactor 设置为“0”时，目标速度变为“0”，轴的动作表现为减速，以速度“0”动作。
   希望保持动作状态，但又想暂时停止轴动作时，将 *VelFactor* 设为“0”。此时，轴状态不会发生变化。
8. 当运动指令执行时或者运动指令交接时，可以改变 VelFactor 来设定新的目标速度。
9. Enable 为 TRUE 时，修改 VelFactor，VelFactor 立即生效，不需要重启 MC_SetOverride 指令。

10. Enable 为 TRUE 时，修改 VelFactor，VelFactor 超出有效范围，MC_SetOverride 指令会报错，目标速度返回 100%。
11. Enable 变为 FALSE 时，以 VelFactor=100 为目标进行加速或减速。
12. 在正在执行 MC_SetOverride 指令的轴中，启动其它的 MC_SetOverride 指令，则以后执行的 MC_SetOverride 指令执行结果为准。两个指令的 Enabled 均表现为 TRUE。
13. 在 MC_MoveVelocity 指令执行时，使用 MC_SetOverride 指令，当 MC_MoveVelocity 指令的 InVelocity 变为 TRUE 后，即使变更目标速度，InVelocity 也保持 TRUE 状态。

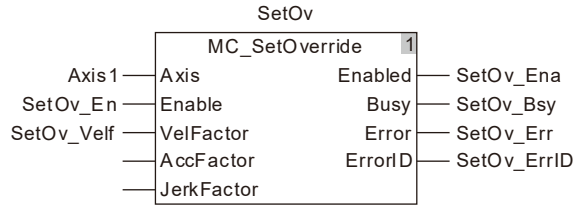
程序范例

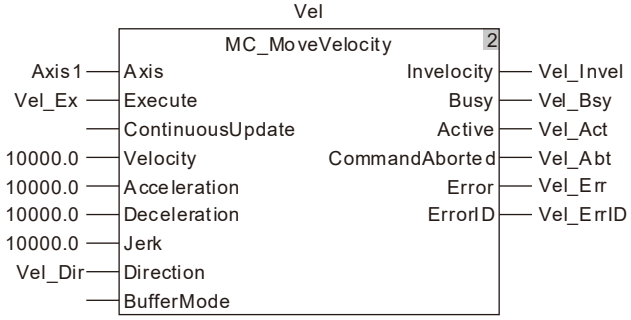
MC_SetOverride 指令执行时的范例如下所示：

范例讲述 MC_SetOverride 指令的执行对 MC_MoveVelocity 指令执行结果的影响。

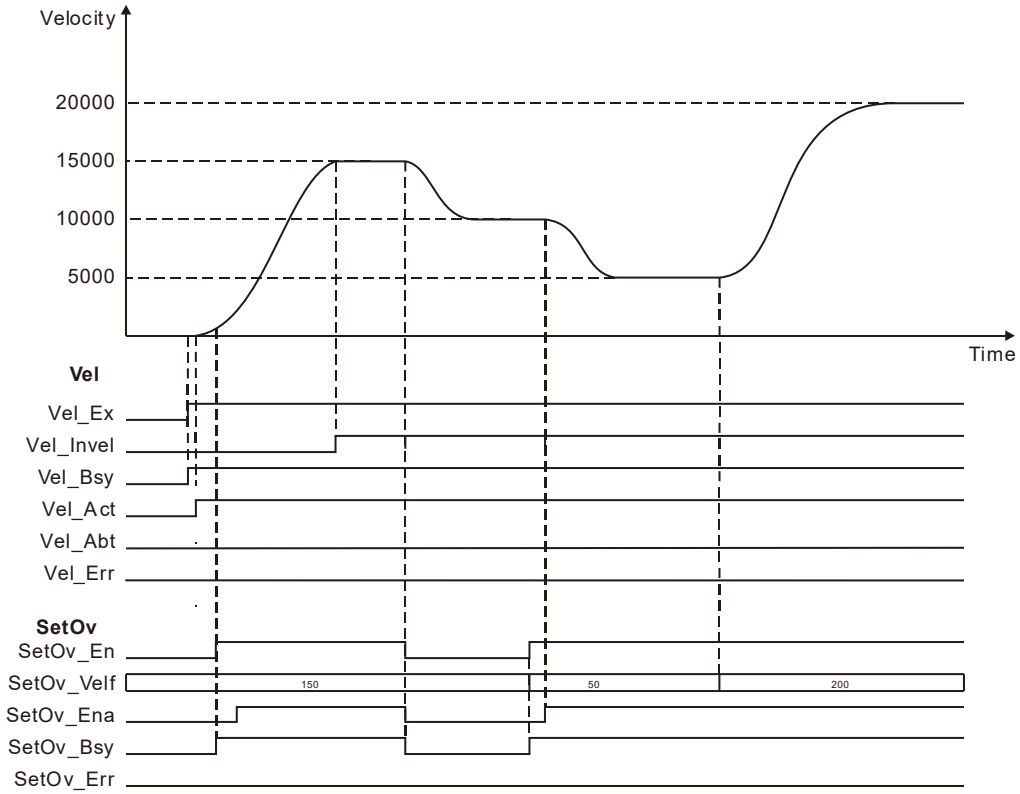
1. **变量和程序**

变量名	数据类型	初始值
SetOv	MC_SetOverride	
Axis1	USINT	1
SetOv_En	BOOL	FALSE
SetOv_Velt	LREAL	0.0
SetOv_Ena	BOOL	
SetOv_Bsy	BOOL	
SetOv_Err	BOOL	
SetOv_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	





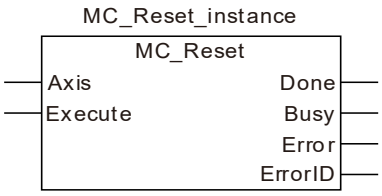
2. **运动曲线和时序图：**



- 当 Vel_Ex 变为 TRUE 时，Vel_Bsy 变为 TRUE，一个周期后，Vel_Act 变为 TRUE，轴开始正转。目标速度还未到达时(Vel_Invel 还未变为 TRUE)，SetOv_En 设置为 TRUE， MC_SetOverride 指令生效，MC_MoveVelocity 指令的目标速度会变成新的目标速度，当 MC_MoveVelocity 指令到达新的目标速度时，Vel_Invel 变为 TRUE。Vel_Invel 变为 TRUE 后，即使变更 VelFactor (本例中为 SetOv_Velf)，Vel_Invel 也保持 TRUE 状态。
- 当 SetOv_En 变为 FALSE 时，相当于 Vel_Invel=100 为目标速度进行减速。
- MC_SetOverride 指令执行过程中修改 SetOv_Velf 的数值，SetOv_Velf 的数值会立即生效，MC_MoveVelocity 指令的目标速度会相应的改变。

11.3.13 MC_Reset（复位指令）

FB/FC	说明	适用机种
FB	此指令用于清除 DVP-15MC 系列运动控制器内部轴的错误状态及轴的报警信息。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Axis（轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节功能简介（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-

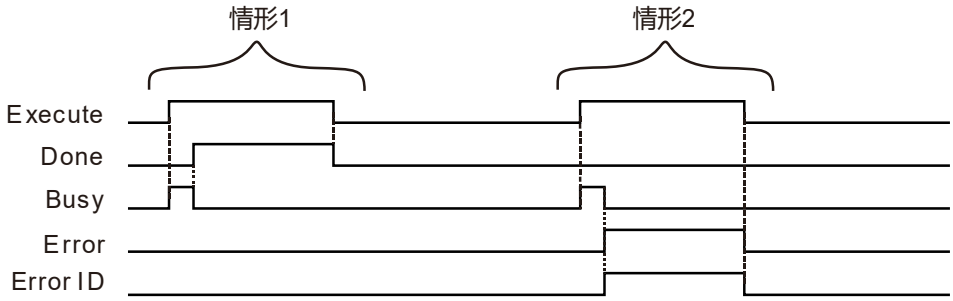
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当该指令执行完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或出现的错误不能清除时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，当指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 位变为 FALSE。当 *Execute* 变为 FALSE 时，*Done* 变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

- **功能说明**

此指令用于清除 DVP-15MC 系列运动控制器内部实轴或者虚轴的错误状态及轴的报警信息。当配置在 DVP-15MC 系列运动控制器内的轴进入错误停止（ErrorStop）状态时（轴的状态可以通过 MC_ReadStatus 指令查看），可以执行此指令清除错误。无论轴是否进入错误停止（ErrorStop）状态，都可以执行该指令。当轴报警、掉线或者状态机切换有问题时，轴进入错误停止（ErrorStop）状态，正在执行的运动指令停止执行。当轴报警时，执行此指令可以清除轴的报警讯息。此指令执行完成后，轴状态由 MC_Power 指令决定，该指令执行完成后，轴状态为 Disable 或者 Stanstill 状态。轴状态说明请参考第 9 章的说明。

轴报警后（回原点过程中遇极限报警除外），报警轴在 DVP-15MC 系列运动控制器内部的状态进入错误停止（ErrorStop）状态。执行此指令后，如果完成位为 TRUE 则轴报警可以消除；如果 Error 位为 TRUE，则轴报警不可以消除，查找导致轴报警的因素是否依然存在。

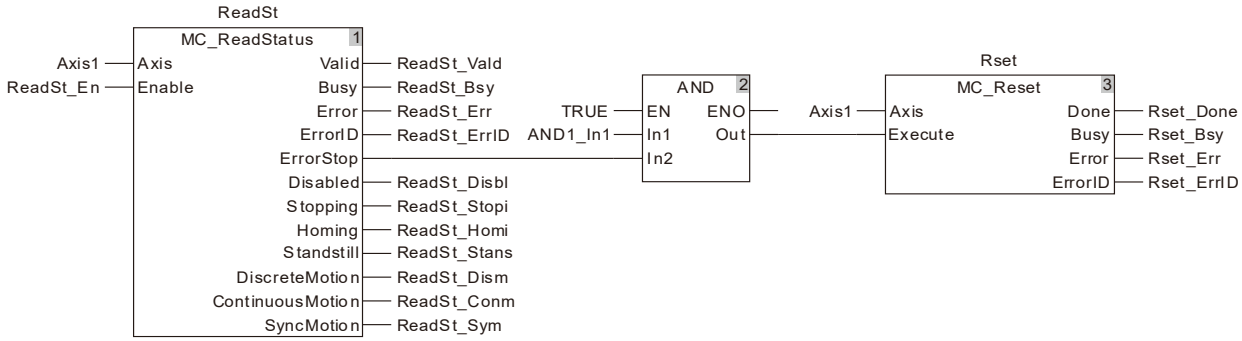
程序范例

ReadSt _En TRUE 时，MC_ReadStatus 指令会检测 1 号轴的状态，当 1 号轴因为掉线或者报警进入 ErrorStop 状态时，MC_ReadStatus 指令的 ErrorStop 位 TRUE，MC_Reset 指令执行。

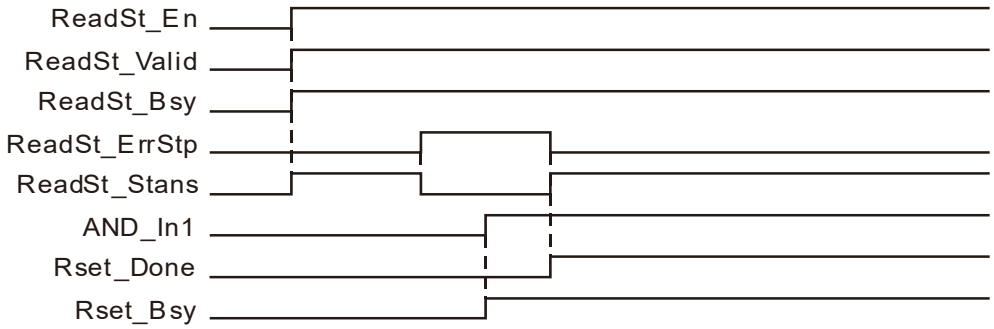
1. 变量和程序

变量名	数据类型	初始值
ReadSt	MC_ReadStatus	
Axis1	USINT	1
ReadSt_En	BOOL	FALSE
ReadSt_Vald	BOOL	
ReadSt_Bsy	BOOL	
ReadSt_Err	BOOL	
ReadSt_ErrID	WORD	
ReadSt_Disbl	BOOL	
ReadSt_Stpin	BOOL	
ReadSt_Homi	BOOL	
ReadSt_Stans	BOOL	

变量名	数据类型	初始值
ReadSt_Dism	BOOL	
ReadSt_Conm	BOOL	
ReadSt_Sym	BOOL	
AND1_In1	BOOL	FALSE
Rset	MC_Reset	
Rset_Done	BOOL	
Rset_Bsy	BOOL	
Rset_Err	BOOL	
Rset_ErrID	WORD	



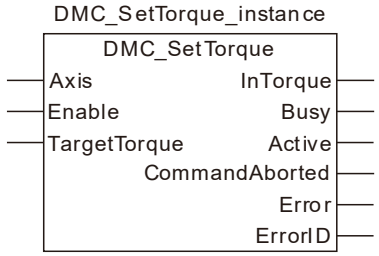
2. 时序图及说明



- ❖ 伺服轴处于使能状态，当 ReadSt_En 由 FALSE 变为 TRUE，ResdSt_Vald 和 ResdSt_Bsy 同时为 TRUE,轴状态处于 Standstill 状态。
- ❖ 当轴进入错误停止（ErrorStop）状态时，使 AND_In1 由 FALSE 变为 TRUE，执行 MC_Reset 指令，Rset_Busy 第一个周期为 TRUE，Rset_Done 第二个周期变为 TRUE，同时轴由 ErrorStop 状态变为 Standstill 状态。

11.3.14 DMC_SetTorque（扭矩设定指令）

FB/FC	说明	适用机种
FB	此指令用于设定伺服轴的扭矩，该指令执行时，伺服轴工作于扭矩模式。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节功能简介 （不可缺省）	*Enable* 由 FALSE 变为 TRUE
Enable （执行位）	当*Enable*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
TargetTorque （扭矩）	此参数用于设置需要的扭矩大小，扭矩大小用伺服轴额定扭矩的千分比来表示，例如设定值为30，则表示设定扭矩为额定扭矩的千分之三十。当执行位为TRUE时，改变此参数值大小，扭矩大小直接跟着变化。	INT	负数、正数、0 （0）	*Enable* 由 FALSE 变为 TRUE

说明：

1. 当扭矩输入值为正数时，伺服作用的方向为正向；当扭矩输入值为负数时，伺服作用的方向变为反向；
2. 当执行位处于高电平时，指令一直有效，改变扭矩设置值大小，扭矩大小直接跟着变化。此功能块不能被其它指令打断（MC_Stop 指令除外）。在执行条件复位时，指令停止执行，可执行其它指令。

- **输出参数**

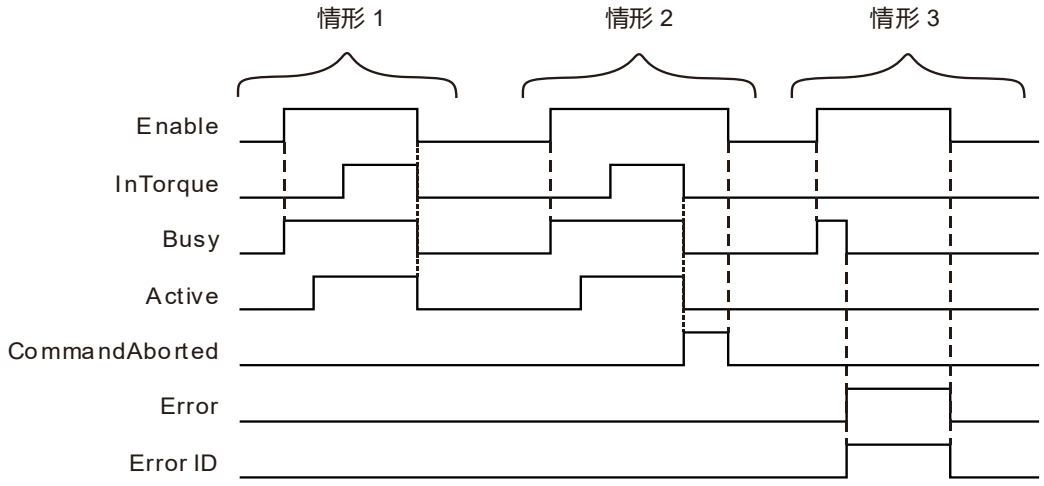
名称	功能	数据类型	输出范围
InTorque （扭矩状态）	该输出参数为 TRUE 时表示扭矩到达.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
InTorque	◆ 当扭矩到达时	◆ 当 *Error* 为 TRUE 时 ◆ 当 *Enable* 由 TRUE 变为 FALSE 时
Busy	◆ 当 *Enable* 为 TRUE 时	◆ 当 *InTorque* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 *InTorque* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，*Enable* 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，*CommandAborted* 被设置为 TRUE，且一个周期后，*CommandAborted* 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Enable* 由 FALSE 变为 TRUE 后，同一个周期 Busy 变为 TRUE，且下一个周期后，*Active* 变为 TRUE，第 3 个周期 *InTorque* 变为 TRUE。当 *Enable* 由 TRUE 变为 FALSE 后，*Busy*、*Active*、*Intorque* 同一个周期变为 FALSE。

情形2： 当 *Enable* 由 FALSE 变为 TRUE 后，该指令被 MC_Stop 指令中断时，*CommandAborted* 变为 TRUE，同时 InTorque、 *Busy* 和 *Active* 变为 FALSE；当 *Enable* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 指令执行之前，输入参数非法（如轴号为 0）。当 *Enable* 由 FALSE 变为 TRUE 后，同一个周期 *Busy* 变为 TRUE，且下一个周期后，*Error* 变为 TRUE，*Busy* 变为 FALSE，*ErrorID* 显示相应的错误代码。当 *Enable* 由 TRUE 变为 FALSE 后，*Error* 由 TRUE 变为 FALSE，*Error ID* 的内容清零。

- **功能说明**

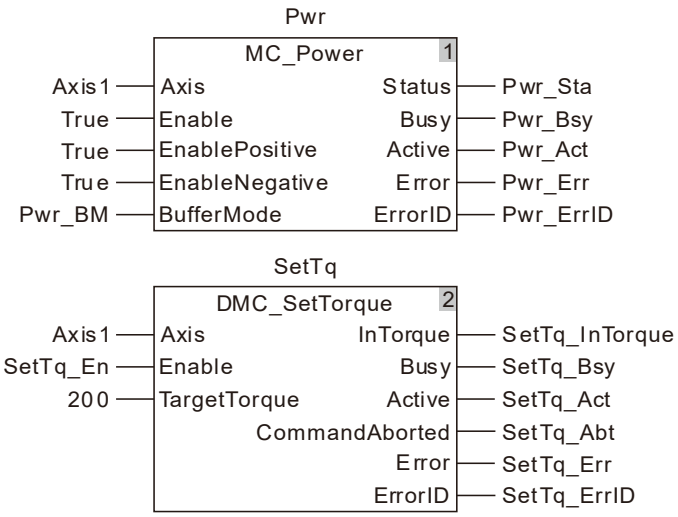
此指令用于设定伺服轴的扭矩，该指令执行时，伺服轴工作于扭矩模式。

程序范例

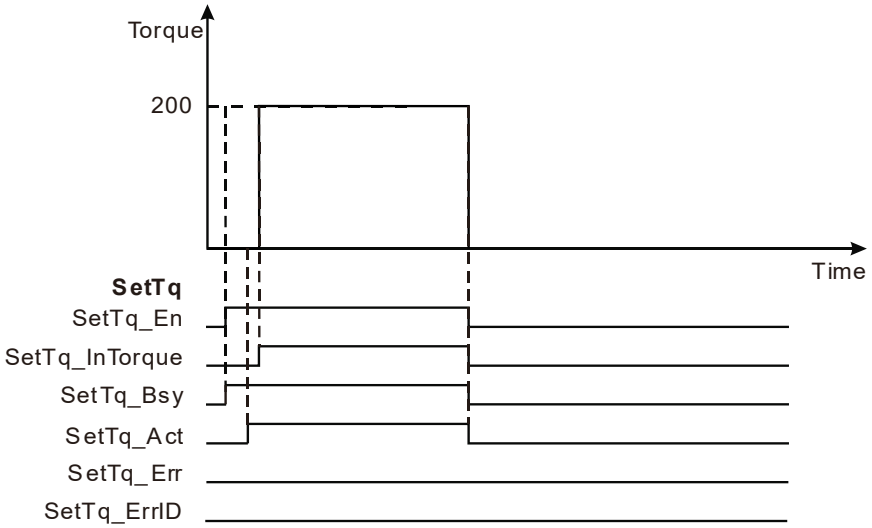
DMC_SETTORQUE 指令执行时的范例如下所示：

1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
SetTq	DMC_SetTorque	
SetTq_En	BOOL	FALSE
SetTq_InTorque	BOOL	
SetTq_Bsy	BOOL	
SetTq_Act	BOOL	
SetTq_Abt	BOOL	
SetTq_Err	BOOL	
SetTq_ErrID	WORD	



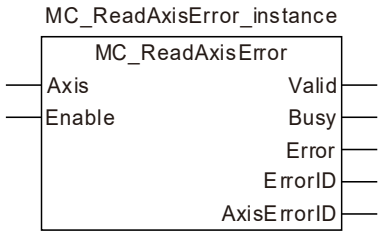
2. 时序图及说明



- ❖ 伺服轴处于使能状态，当 SetTq_En 由 FALSE 变为 TRUE 时，同一周期 SetTq_Bsy 变为 TRUE，且一个周期后，SetTq_Act 变为 TRUE，开始执行扭矩指令；当扭矩到达时，SetTq_InTorque 变为 TRUE， SetTq_Bsy 和 SetTq_Act 依然保持为 TRUE。
- ❖ 当 SetTq_En 由 FALSE 变为 TRUE 时，SetTq_InTorque、SetTq_Bsy 和 SetTq_Act 变为 FALSE。

11.3.15 MC_ReadAxisError（读取轴错误指令）

FB/FC	说明	适用机种
FB	此指令用于读取伺服轴的错误信息。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Axis（轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节**功能简介**（不可缺省）	*Enable* 为 TRUE 时
Enable（执行位）	当 *Enable* 由 FALSE 变为 TRUE 时，指令开始执行	BOOL	TRUE 或 FALSE（FALSE）	-

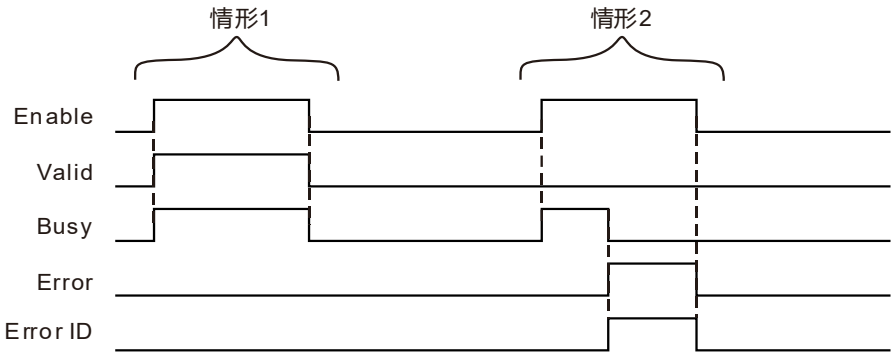
- **输出参数**

名称	功能	数据类型	输出范围
Valid（输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
AxisErrorID（轴错误代码）	输出有效位为TRUE时，AxisErrorID的值为xxx（hex）表示伺服驱动器报警，xxx的值表示伺服驱动器的报警代码。如伺服驱动器报警为AL303时，ErrorID的值为303（hex）。	UINT	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当读到轴的错误时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 和 Busy 同时变为 TRUE。当 Enable 变为 FALSE 时，Valid、Busy 均变为 FALSE。

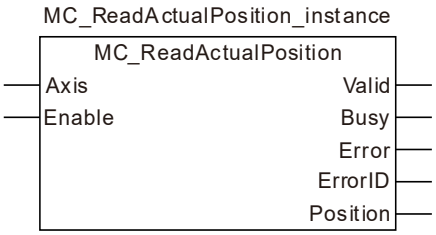
情形2： 当有错误产生时， Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 的值会被清除。

- **功能说明**

此指令用于读取伺服轴的错误信息，例如伺服驱动器面板上显示的警报错误，或伺服轴是否断线等。该指令为高电平触发，当执行位为 TRUE 时，会读取轴错误信息。

11.3.16 MC_ReadActualPosition（实际位置读取指令）

FB/FC	说明	适用机种
FB	此指令用于读取轴的实际位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Enable* 为 TRUE 时
Enable （执行位）	当 *Enable* 为 TRUE 时，指令执行。	BOOL	TRUE 或 FALSE （FALSE）	-

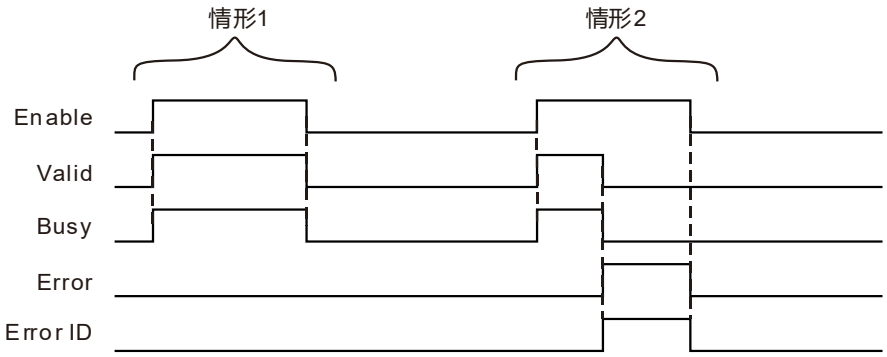
- **输出参数**

名称	功能	数据类型	输出范围
Valid （输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
Position （实际位置）	轴实际位置	LREAL	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 已经读取到实际位置时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 *Valid* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Enable* 由 FALSE 变为 TRUE 时，*Valid* 和 *Busy* 同时变为 TRUE。当 *Enable* 变为 FALSE 时，*Valid* 、*Busy* 均变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE ，*ErrorID* 显示对应的错误码 ，同时 *Busy* 和 *Valid* 变为 FALSE；当 *Enable* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

- **功能说明**

此指令用于读取轴的实际位置（包括实轴、虚轴和编码器轴）。

■ **实际位置**

此指令读取的实际位置的单位为单元，而伺服驱动器反馈给控制器的位置的单位为脉冲，故此实际位置由伺服驱动器的位置反馈脉冲数经转换后得到，转换时需使用轴参数中的电子齿轮比、机构齿轮比以及机构导程，转换关系式如下图所示：

$$ActualPosition = \frac{机构导程}{(脉冲数/转)*机构齿轮比} * 伺服位置反馈脉冲数$$

若轴为线性轴，则指令执行时输出的 Position=ActualPosition;

若轴为旋转轴，则指令执行时输出的 Position=ActualPosition % 模（Position 为 ActualPosition 按轴参数中设置的模做取余运算的结果），所以 Posiiton 的值在 0~模之间变化。

■ **实际位置更新时机**

由于此实际位置来源于伺服驱动器的位置反馈脉冲数，则实际位置的刷新时机与控制器和伺服驱动器之间的通讯周期有关。在一个通讯周期内，伺服反馈其位置脉冲数给控制器的动作仅发生一次，则在一个通讯周期内，读取到的实际位置值保持不变。

介于上述原因，此指令读取轴实际位置的实时性不及位置捕获，若需获取实时性更高的位置，请使用位置捕获功能。

■ **MC_SetPosition 对实际位置的影响**

指令 MC_SetPosition 执行后，MC_ReadActualPosition 指令读取的实际位置须加上是 MC_SetPosition 指令引起的位置偏移量，换算关系式如下图所示：

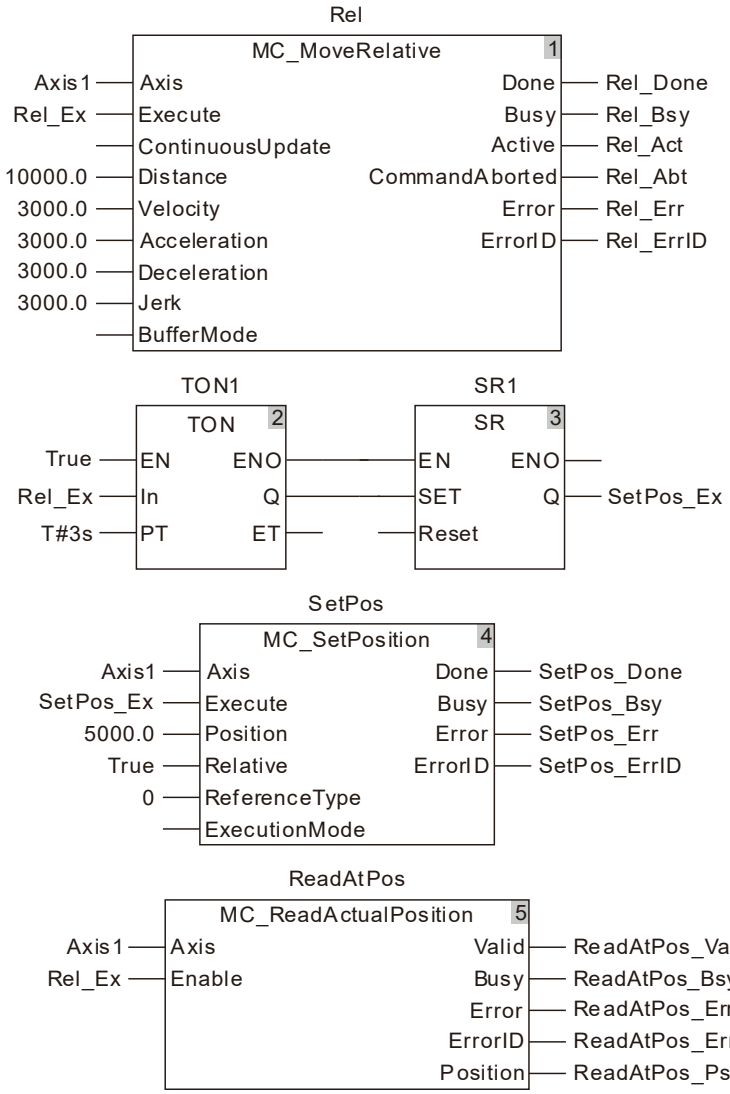
$$ActualPosition = \underset{引起的位置偏移量}{MC\_SetPosition} + \frac{机构导程}{(脉冲数/转)*机构齿轮比} * 伺服位置反馈脉冲数$$

程序范例

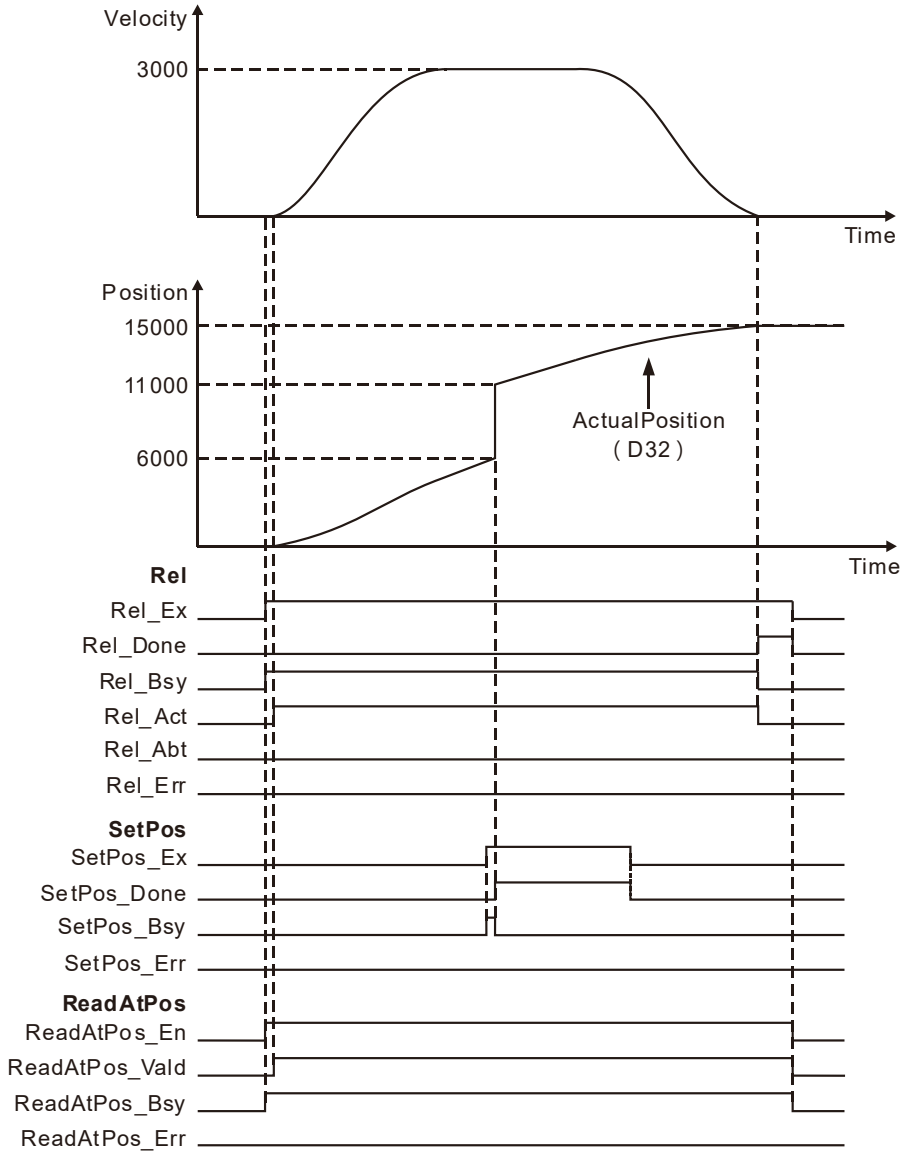
本例描述 MC_SetPosition 指令对 MC_ReadActualPosition 指令的影响，范例程序如下：

1. 变量和程序

变量名	数据类型	初始值
Rel	MC_MoveRelative	
Axis1	USINT	1
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
TON1	TON	
SR1	SR	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNETYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	
ReadAtPos	ReadActualPosition	
ReadAtPos_Vald	BOOL	
ReadAtPos_Bsy	BOOL	
ReadAtPos_Err	BOOL	
ReadAtPos_ErrID	WORD	
ReadAtPos_Pst	LREAL	


Rel
MC_MoveRelative
1
Axis1
Axis
Rel_Ex
Execute
ContinuousUpdate
10000.0
Distance
3000.0
Velocity
3000.0
Acceleration
3000.0
Deceleration
3000.0
Jerk
BufferMode
Done
Rel_Done
Busy
Rel_Bsy
Active
Rel_Act
CommandAborted
Rel_Abt
Error
Rel_Err
ErrorID
Rel_ErrID
TON1
TON
2
True
EN
ENO
Rel_Ex
In
Q
T#3s
PT
ET
SR1
SR
3
EN
ENO
SET
Q
SetPos_Ex
Reset
SetPos
MC_SetPosition
4
Axis1
Axis
SetPos_Ex
Execute
5000.0
Position
True
Relative
0
ReferenceType
ExecutionMode
Done
SetPos_Done
Busy
SetPos_Bsy
Error
SetPos_Err
ErrorID
SetPos_ErrID
ReadAtPos
MC_ReadActualPosition
5
Axis1
Axis
Rel_Ex
Enable
Valid
ReadAtPos_Vald
Busy
ReadAtPos_Bsy
Error
ReadAtPos_Err
ErrorID
ReadAtPos_ErrID
Position
ReadAtPos_Pst

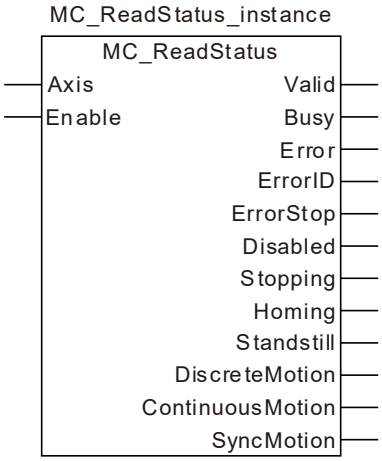
2. 运动曲线和时序图



- 当 Rel_Ex 由 FALSE 变为 TRUE 时，Rel（MC_MoveRelative）、ReadAtPos（MC_ReadActualPosition）同时开始执行，Rel（MC_MoveRelative）指令执行 3 秒后 SetPos（MC_SetPosition）指令执行。
- SetPos（MC_SetPosition）指令开始执行时的 ActualPosition 为 6000，其执行完成后的 ActualPosition 为 11000，其中 11000=6000+5000，Rel（MC_MoveRelative）指令执行完成时的 ActualPosition 为 15000。
- 通过上图中的速度变化曲线可以看出 SetPos（MC_SetPosition）指令对正在执行的运动不会产生影响，但 ActualPosition 曲线却反映出 Rel（MC_MoveRelative）读到的 ActualPosition 值会受到 SetPos（MC_SetPosition）指令的影响。

11.3.17 MC_ReadStatus（读取轴状态指令）

FB/FC	说明	适用机种
FB	此指令用于读取伺服轴在控制器内的轴状态信息	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Enable* 为 TRUE 时
Enable （执行位）	当 *Enable* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-

- **输出参数**

名称	功能	数据类型	输出范围
Vaild （输出可用）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
ErrorStop （异常停止）	请参考第 10.4 节	BOOL	TRUE / FALSE
Disabled （未执行）		BOOL	TRUE / FALSE
Stopping （正常停止）		BOOL	TRUE / FALSE
Homing （原点回归）		BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Standstill （准备执行）		BOOL	TRUE / FALSE
DiscreteMotion （离散运动）		BOOL	TRUE / FALSE
ContinuousMotion （连续运动）		BOOL	TRUE / FALSE
SyncMotion （同步运动）		BOOL	TRUE / FALSE

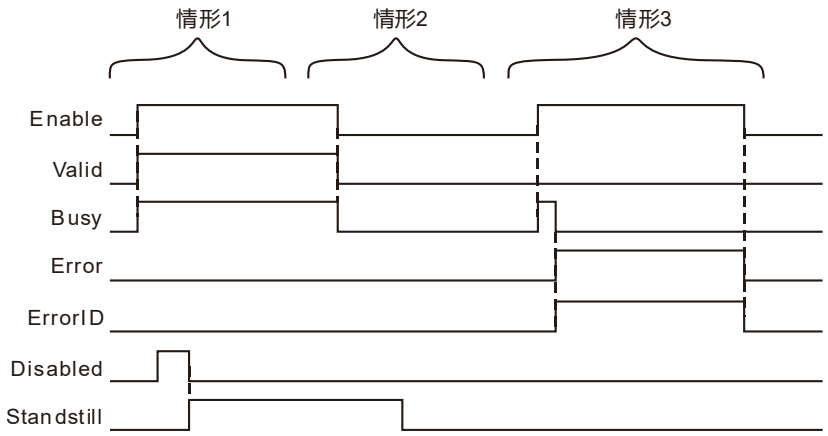
说明：

1. 该指令在 *Enable* 由 FALSE 变为 TRUE 时开始执行，读取轴状态。
2. 该指令在 *Enable* 由 TRUE 变为 FALSE 时，*Vaild*、*Busy* 和 *Error* 变为 FALSE，同时 *ErrorID* 变为 0，*ErrorStop*、*Disabled*、*Stopping*、*Homing*、*Standstill*、*DiscreteMotion*、*ContinuousMotion* 和 *SyncMotion* 的输出结果保持 *Enable* 为 TRUE 时的状态不变。

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Vaild	◆ 当 *Enable* 为 TRUE 时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时 ◆ 当 *Error* 由 FALSE 变为 TRUE 时
Busy	◆ 当 *Enable* 为 TRUE 时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时 ◆ 当 *Error* 由 FALSE 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时
ErrorStop	◆ 当轴状态处于 *ErrorStop* 状态	◆ 当轴状态不处于 *ErrorStop* 状态
Disabled	◆ 当轴状态处于 *Disabled* 状态	◆ 当轴状态不处于 *Disabled* 状态
Stopping	◆ 当轴状态处于 *Stopping* 状态	◆ 当轴状态不处于 *Stopping* 状态
Homing	◆ 当轴状态处于 *Homing* 状态	◆ 当轴状态不处于 *Homing* 状态
Standstill	◆ 当轴状态处于 *Standstill* 状态	◆ 当轴状态不处于 *Standstill* 状态
DiscreteMotion	◆ 当轴状态处于 *DiscreteMotion* 状态	◆ 当轴状态不处于 *DiscreteMotion* 状态
ContinuousMotion	◆ 当轴状态处于 *ContinuousMotion* 状态	◆ 当轴状态不处于 *ContinuousMotion* 状态
SyncMotion	◆ 当轴状态处于 *SyncMotion* 状态	◆ 当轴状态离开 *SyncMotion* 状态

- **输出参数变化时序图**



情形1： 当 *Enable* 由 FALSE 变为 TRUE 时， *Vaild* 和 *Busy* 同时变为 TRUE， *ErrorStop* 、*Disabled* 、*Stopping* 、*Homing* 、*Standstill* 、*DiscreteMotion* 、*ContinuousMotion* 和 *SyncMotion* 根据轴状态变为 TRUE 或 FALSE。

情形2： 当 *Enable* 由 TRUE 变为 FALSE 时，*Vaild* 和 *Busy* 同时变为 FALSE， *ErrorStop* 、*Disabled* 、*Stopping* 、*Homing* 、*Standstill* 、*DiscreteMotion* 、*ContinuousMotion* 和 *SyncMotion* 引脚的输出结果保持 *Enable* 为 TRUE 时的状态不变。

情形3： 当输入参数 Axis 的值超出范围，当 *Enable* 由 FALSE 变为 TRUE 时，*Busy* 同时由 FALSE 变为 TRUE，一个周期后，*Error* 由 FALSE 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 由 TRUE 变为 FALSE ;当 *Enable* 由 TRUE 变为 FALSE 时 ,*Error* 由 TRUE 变为 FALSE ,同时 *ErrorID* 变为 0。

- **功能说明**

此指令用于读取伺服轴在控制器内的轴状态信息，关于轴状态的详细说明请参考第 10.4 节。

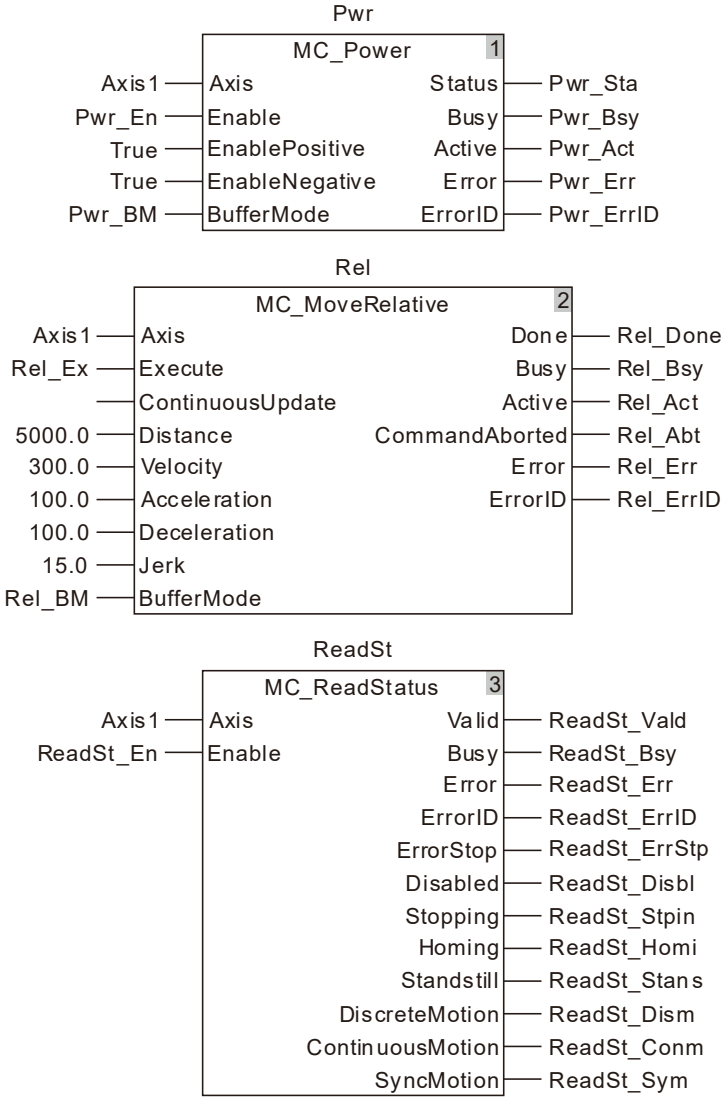
程序范例

MC_ReadStatus 指令执行范例如下所示：

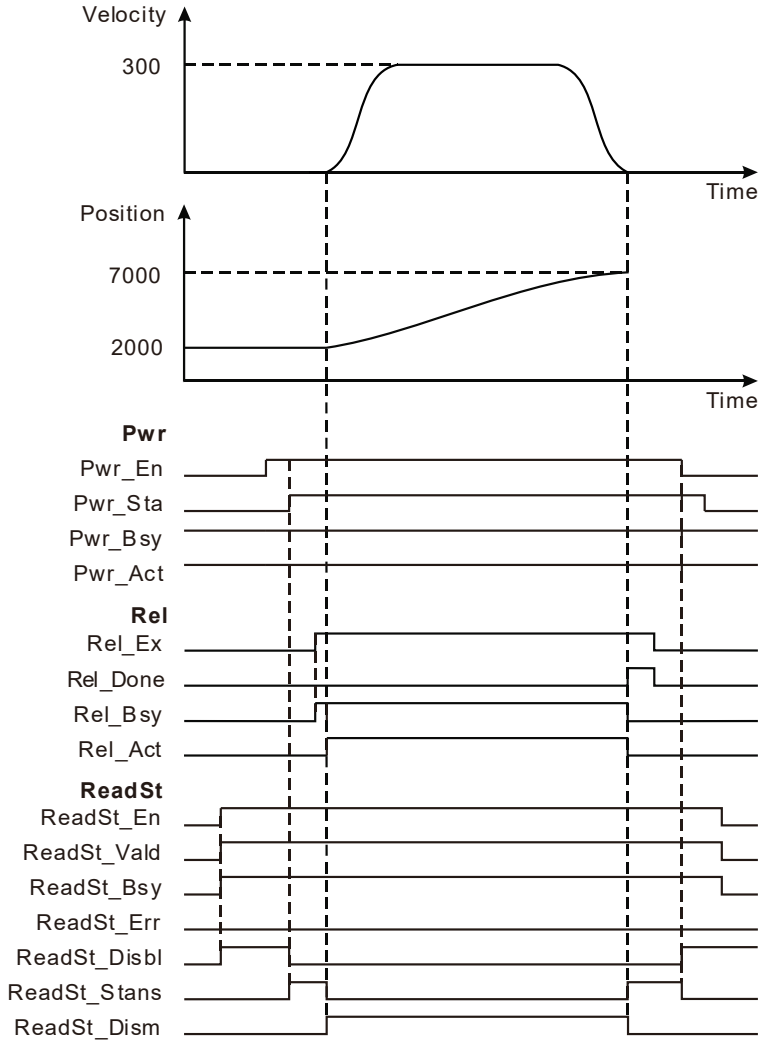
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_BM	MC_Buffer_Mode	0

变量名	数据类型	初始值
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
ReadSt	MC_ReadStatus	
ReadSt_En	BOOL	FALSE
ReadSt_Vald	BOOL	
ReadSt_Bsy	BOOL	
ReadSt_Err	BOOL	
ReadSt_ErrID	WORD	
ReadSt_ErrStp	BOOL	
ReadSt_Disbl	BOOL	
ReadSt_Stpin	BOOL	
ReadSt_Homi	BOOL	
ReadSt_Stans	BOOL	
ReadSt_Dism	BOOL	
ReadSt_Conm	BOOL	
ReadSt_Sym	BOOL	


Pwr
MC_Power
1
Axis1
Axis
Status
Pwr_Sta
Pwr_En
Enable
Busy
Pwr_Bsy
True
EnablePositive
Active
Pwr_Act
True
EnableNegative
Error
Pwr_Err
Pwr_BM
BufferMode
ErrorID
Pwr_ErrID
Rel
MC_MoveRelative
2
Axis1
Axis
Done
Rel_Done
Rel_Ex
Execute
Busy
Rel_Bsy
ContinuousUpdate
Active
Rel_Act
5000.0
Distance
CommandAborted
Rel_Abt
300.0
Velocity
Error
Rel_Err
100.0
Acceleration
ErrorID
Rel_ErrID
100.0
Deceleration
15.0
Jerk
Rel_BM
BufferMode
ReadSt
MC_ReadStatus
3
Axis1
Axis
Valid
ReadSt_Vald
ReadSt_En
Enable
Busy
ReadSt_Bsy
Error
ReadSt_Err
ErrorID
ReadSt_ErrID
ErrorStop
ReadSt_ErrStp
Disabled
ReadSt_Disbl
Stopping
ReadSt_Stpin
Homing
ReadSt_Homi
Standstill
ReadSt_Stans
DiscreteMotion
ReadSt_Dism
ContinuousMotion
ReadSt_Conm
SyncMotion
ReadSt_Sym

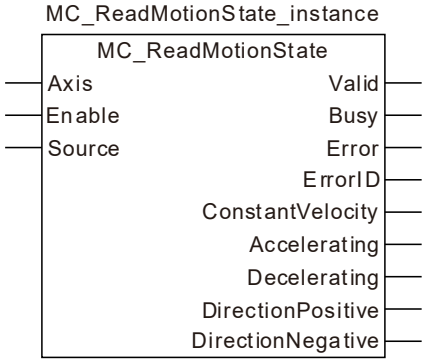
2. 运动曲线和时序图



- ❖ 当 ReadSt_En 由 FALSE 变为 TRUE 时，ReadSt_Vald、 ReadSt_Bsy 和 ReadSt_Disbl 变为 TRUE。
- ❖ 当 Pwr_Sta 由 FALSE 变为 TRUE 时，ReadSt_Stans 变为 TRUE，ReadSt_Disbl 变为 FALSE，轴状态由 *Disabled* 状态变为 *Standstill* 状态。
- ❖ 当 Rel_Act 由 FALSE 变为 TRUE 时，运动控制器控制伺服电机从当前位置转动，同时 ReadSt_Stans 变为 FALSE，ReadSt_Dism 变为 TRUE。当伺服电机转动到目标距离，Rel_Done 和 ReadSt_Stans 变为 TRUE，Rel_Bsy、Rel_Act 和 ReadSt_Dism 变为 FALSE。
- ❖ 当 Rel_Ex 变为 FALSE 时，Rel_Done 也变为 FALSE。
- ❖ 当 Pwr_En 变为 FALSE 时，ReadSt_Disbl 变为 TRUE，ReadSt_Stans 变为 FALSE，若干周期后 Pwr_Sta 也变为 FALSE。
- ❖ 当 ReadSt_En 变为 FALSE 时，ReadSt_Vald 和 ReadSt_Bsy 变为 FALSE，ReadSt_Disbl 保持 TRUE 不变。

11.3.18 MC_ReadMotionState（读取轴运动状态指令）

FB/FC	说明	适用机种
FB	此指令用于读取伺服轴当前的运动状态	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Enable* 由 FALSE 变为 TRUE
Enable （执行位）	当 *Enable* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
Source （源）	保留	-	-	-

说明：该指令在 *Enable* 由 FALSE 变为 TRUE 时开始执行。该指令正在执行时，*Enable* 由 TRUE 变为 FALSE 时，指令停止运行， ConstantVelocity、Accelerating、Decelerating、DirectionPositive、DirectionNegative 的输出结果保持 Enable 为 TRUE 时的状态不变。

- **输出参数**

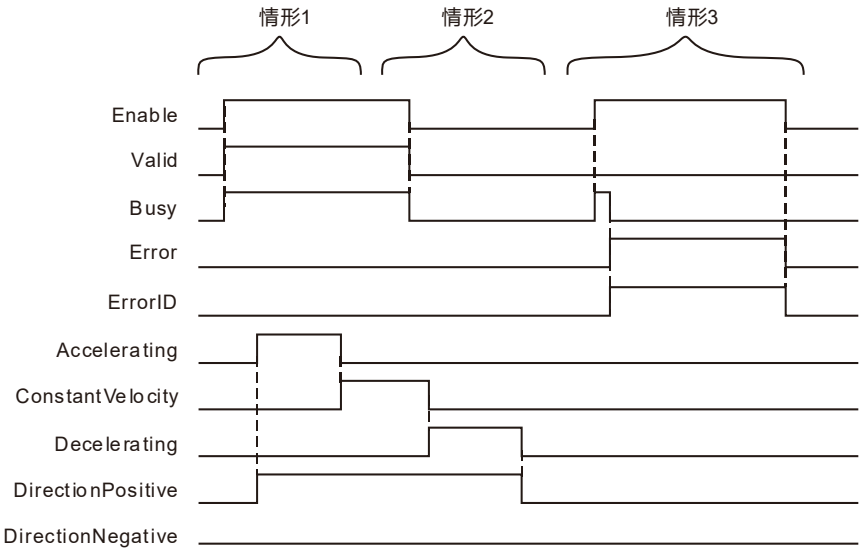
名称	功能	数据类型	输出范围
Valid （输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
ConstantVelocity （匀速状态）	该输出参数为 TRUE 时表示轴正在做匀速运动	BOOL	TRUE / FALSE
Accelerating （加速状态）	该输出参数为 TRUE 时表示轴速度的绝对值增大	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Decelerating（减速状态）	该输出参数为 TRUE 时表示轴速度的绝对值减小	BOOL	TRUE / FALSE
DirectionPositive（轴正转）	该输出参数为 TRUE 时表示轴当前位置增大	BOOL	TRUE / FALSE
DirectionNegative（轴反转）	该输出参数为 TRUE 时表示轴当前位置减小	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当读到轴的实际速度时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时
ErrorID	◆	◆
ConstantVelocity	◆ 当轴速度不变时	◆ 当轴速度发生变化且 Enable 仍然为 TRUE
Accelerating	◆ 当轴速度的绝对值增大时	◆ 当轴速度的绝对值不再变大且 Enable 仍然为 TRUE
Decelerating	◆ 当轴速度的绝对值减小时	◆ 当轴速度的绝对值不再变小且 Enable 仍然为 TRUE
DirectionPositive	◆ 当轴当前位置增大时	◆ 当轴当前位置不再增大且 Enable 仍然为 TRUE
DirectionNegative	◆ 当轴当前位置减小时	◆ 当轴当前位置不再减小且 Enable 仍然为 TRUE

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Vaild 和 Busy 同时变为 TRUE， ConstantVelocity、Accelerating、Decelerating、DirectionPositive、DirectionNegative 引脚的输出结果根据轴状态变为 TRUE 或 FALSE。

情形2： 当 Enable 由 TRUE 变为 FALSE 时， Vaild 和 Busy 同时变为 FALSE，ConstantVelocity、Accelerating、Decelerating、DirectionPositive、DirectionNegative 引脚的输出结果保持 Enable 为 TRUE 时的状态不变。

情形3： 当输入参数 Axis 的值超出范围，当 Enable 由 FALSE 变为 TRUE 时，Busy 同时由 FALSE 变为 TRUE，一个周期后，Error 由 FALSE 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 由 TRUE 变为 FALSE ;当 Enable 由 TRUE 变为 FALSE 时 ,Error 由 TRUE 变为 FALSE ,同时 ErrorID 变为 0。

- **功能说明**

此指令用于读取伺服轴当前的运动状态。伺服轴的运动状态包括：匀速运动、加速运动或减速运动，以及正转或反转。

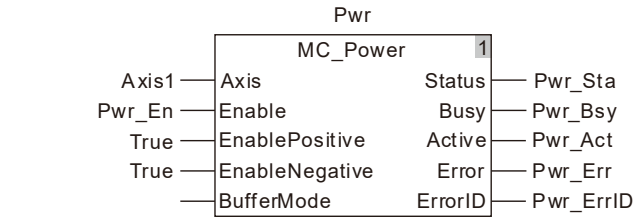
程序范例

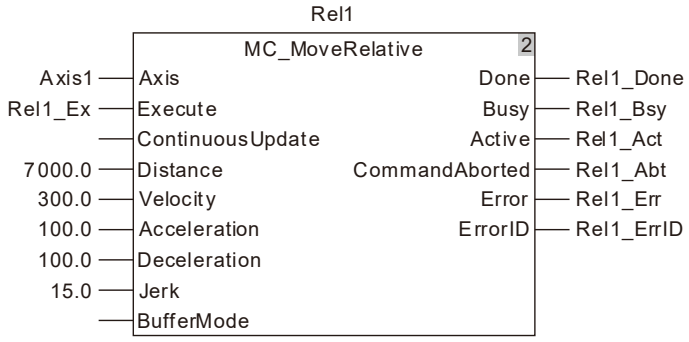
MC_ReadMotionState 指令执行范例如下所示：

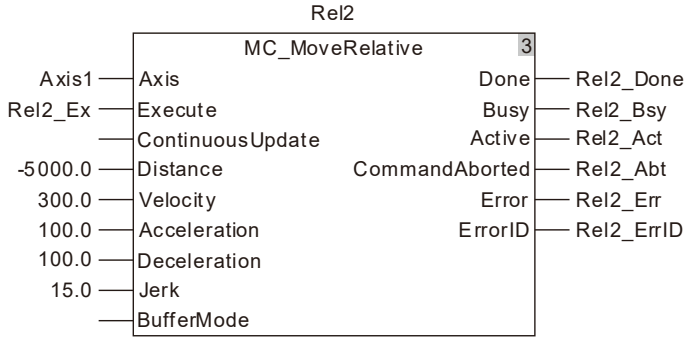
1. 变量和程序

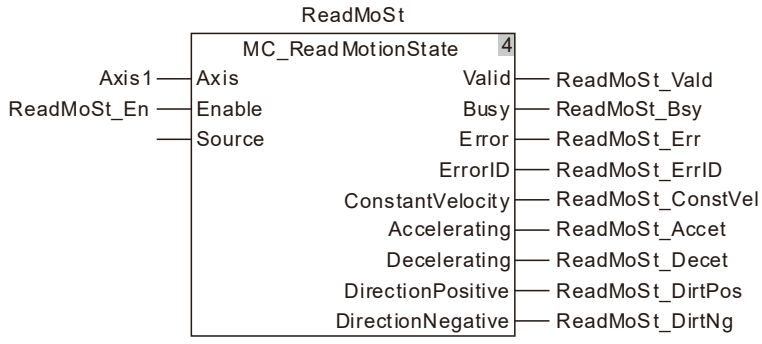
变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	FALSE
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	FALSE
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	WORD	
ReadMoSt	MC_ReadMotionState	

变量名	数据类型	初始值
ReadMoSt_En	BOOL	FALSE
ReadMoSt_Vald	BOOL	
ReadMoSt_Bsy	BOOL	
ReadMoSt_Err	BOOL	
ReadMoSt_ErrID	WORD	
ReadMoSt_ConstVel	BOOL	
ReadMoSt_Accet	BOOL	
ReadMoSt_Decet	BOOL	
ReadMoSt_DirtPos	BOOL	
ReadMoSt_DirtNg	BOOL	

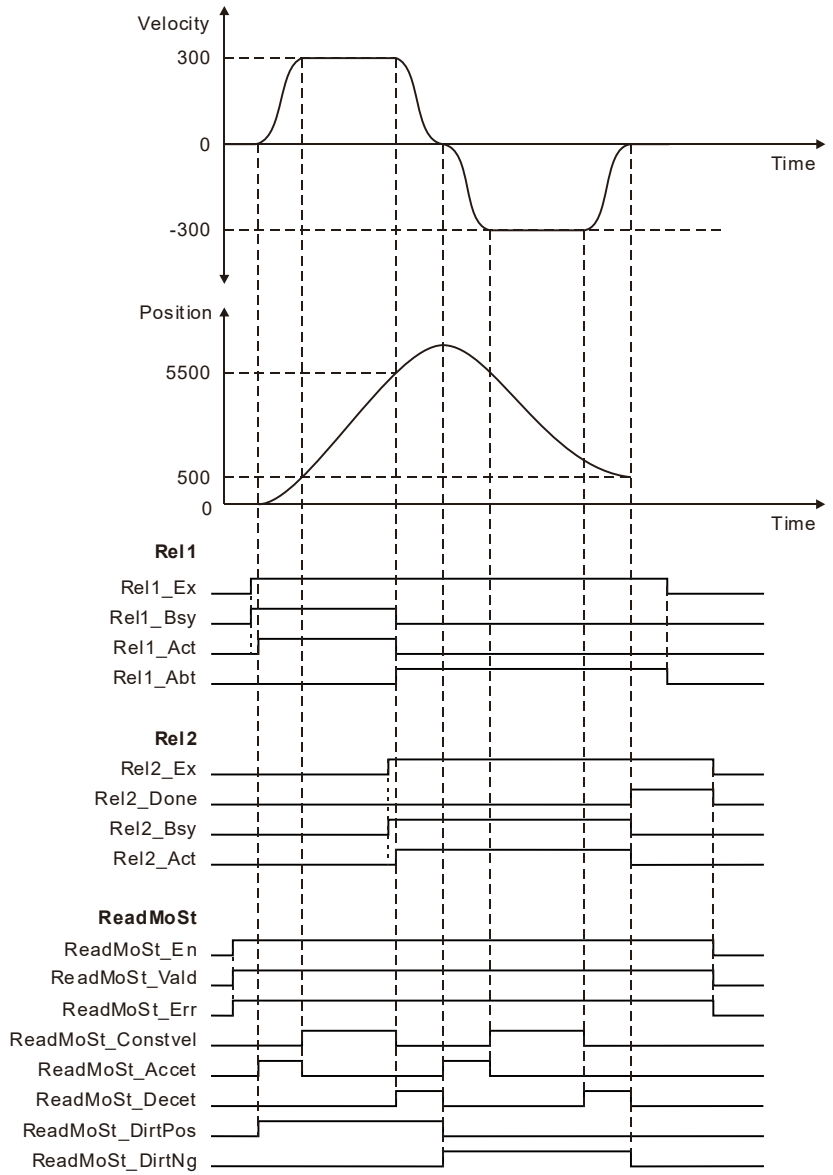








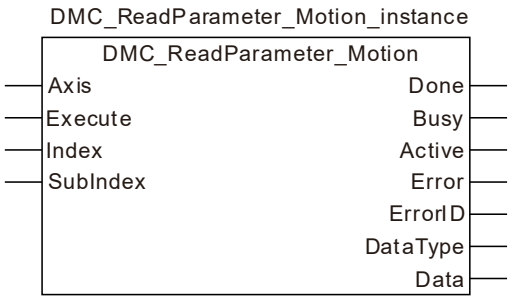
2. 运动曲线和时序图



- ❖ 当 ReadMoSt_En 由 FALSE 变为 TRUE 时，ReadMoSt_Vald、ReadMoSt_Bsy 同时由 FALSE 变为 TRUE。
- ❖ 当 Rel1_Act 由 FALSE 变为 TRUE 时，轴开始正向加速，同时 ReadMoSt_Accet、ReadMoSt_DirtPos 同时变为 TRUE。
- ❖ 当 ReadMoSt_Constvel 由 FALSE 变为 TRUE 时，ReadMoSt_Accet 同时由 TRUE 变为 FALSE，轴进入正向匀加速运动状态。
- ❖ 当 Rel2_Act 由 FALSE 变为 TRUE 时，ReadMoSt_Decet 同时由 FALSE 变为 TRUE，轴开始正向减速。
- ❖ 当 ReadMoSt_Accet 和 ReadMoSt_DirtNg 由 FALSE 变为 TRUE 时，ReadMoSt_Decet 和 ReadMoSt_DirtPos 同时变为 FALSE，轴开始反向加速。
- ❖ 当 Rel2_Done 由 FALSE 变为 TRUE 时，轴停止运转，ReadMoSt_Decet 和 ReadMoSt_DirtNg 同时变为 FALSE。

11.3.19 DMC_ReadParameter_Motion（读取参数指令）

FB/FC	说明	适用机种
FB	此指令用于读取从站的参数值。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （从站的站号）	设定指令欲控制的从站站号	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当*Execute*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
Index （索引）	读取参数的索引。	UINT	0	*Execute* 由 FALSE 变为 TRUE
SubIndex （子索引）	读取参数的子索引。	USINT	0	*Execute* 由 FALSE 变为 TRUE

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制从站	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
Data Type （数据类型）	读取参数的数据类型。 1：代表读取的参数为 Byte 类型； 2：代表读取的参数为 Word 类型； 4：代表读取的参数为 Double Word 类型。	USINT	-

名称	功能	数据类型	输出范围
Data（参数值）	读到的参数值。	UDINT	-

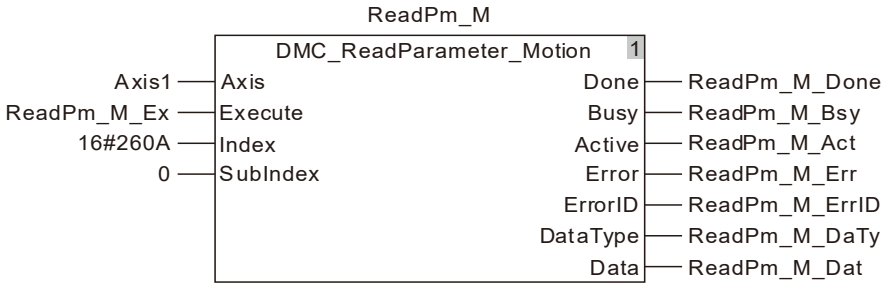
说明：欲读取的从站参数对应索引和子索引如下：

1. 用户定义参数为欲读取的伺服驱动器参数，长度由用户根据所读取参数的数据类型指定，字节型参数长度为 1，字型参数长度为 2，双字型参数长度为 4。伺服驱动器参数对应索引子索引的计算方法如下：

   索引 = 伺服驱动器参数（十六进制）+ 2000（十六进制）

   子索引 = 0.

   举例： 伺服驱动器参数 P6-10 的索引计算方法为 2000 + 060A（P6-10 的十六进制数） = 260A.，子索引为 0.。

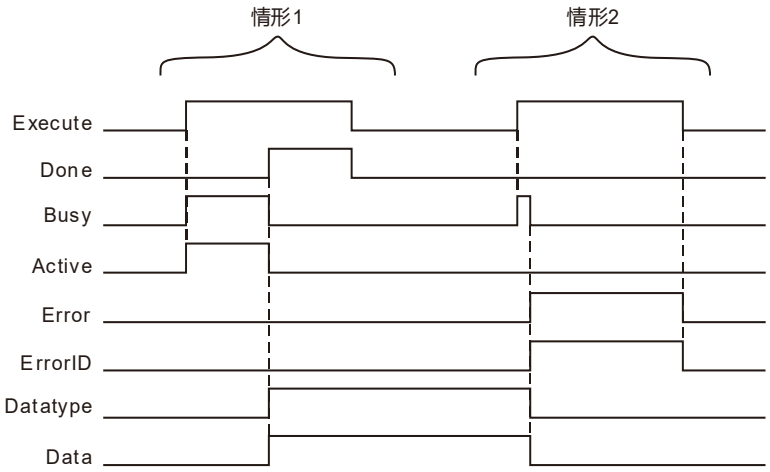


2. 其它从站参数对应索引和子索引，请参考产品的 CANopen 功能相关手册。

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当参数内容读取完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Busy 变为 FALSE
Active	◆ 当指令开始控制从站时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Active 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后，*Busy* 和 *Active* 变为 TRUE，且一个周期后，Done 变为 TRUE，Datatype 和 Data 显示相应的数据。Done 变为 TRUE 后，同一个周期内 Busy 和 Active 变为 FALSE。当 *Execute* 由 TRUE 变为 FALSE 后,Done 由 TRUE 变为 FALSE，Datatype 和 Data 保持原有的数据，如果 Error 位变为 TRUE，Datatype 和 Data 的数据会清零。

情形2： 指令执行之前，输入参数非法(如轴号为 0)，当 *Execute* 由 FALSE 变为 TRUE 后，Error 由 FALSE 变为 TRUE，Error ID 显示相应的错误代码。当 *Execute* 由 TRUE 变为 FALSE 后,Error 由 TRUE 变为 FALSE，Error ID 的内容清零。

- **功能说明**

此指令用于读取从站的参数值，用户可以指定欲读取参数的索引（Index）和子索引（Sub-Index）。

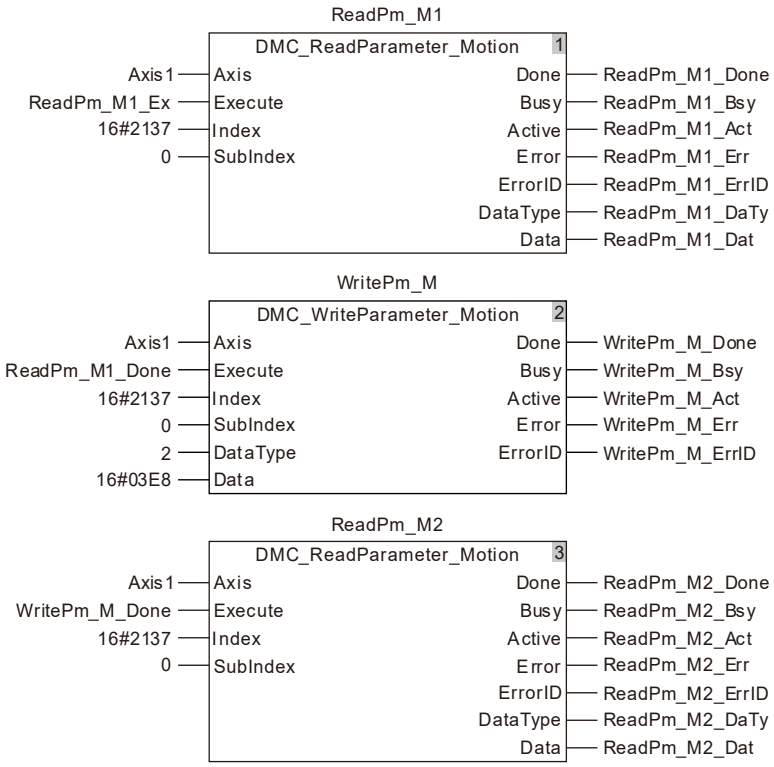
程序范例

DMC_ReadParameter_ Motion 指令执行范例如下所示：

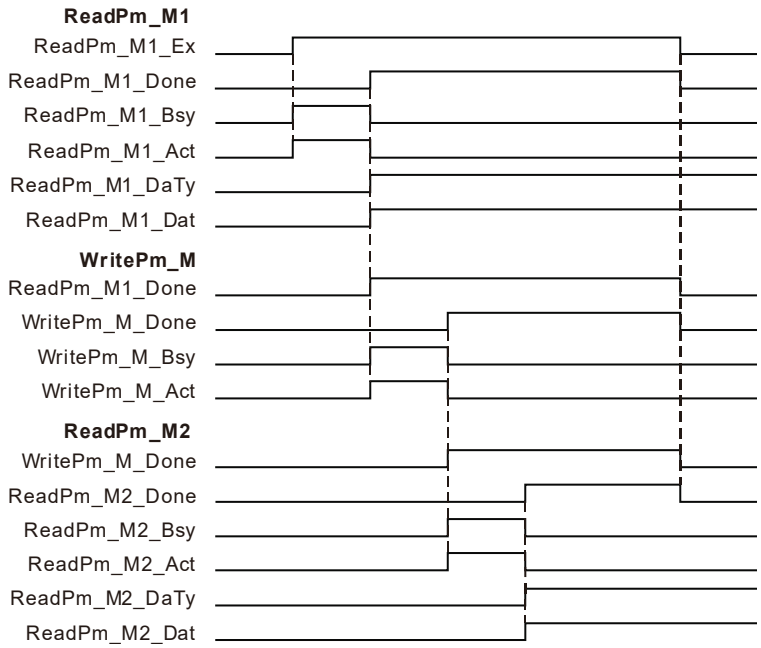
1. 变量和程序

变量名	数据类型	当前值
ReadPm_M1	DMC_ReadParameter_ Motion	
Axis1	USINT	1
ReadPm_M1_Ex	BOOL	TRUE
ReadPm_M1_Done	BOOL	TRUE
ReadPm_M1_Bsy	BOOL	FALSE
ReadPm_M1_Act	BOOL	FALSE
ReadPm_M1_Err	BOOL	FALSE
ReadPm_M1_ErrID	WORD	FALSE
ReadPm_M1_DaTy	USINT	2
ReadPm_M1_Dat	UDINT	5000
WritePm_M	DMC_WriteParameter_Motion	
WritePm_M_Done	BOOL	TRUE
WritePm_M_Bsy	BOOL	FALSE
WritePm_M_Act	BOOL	FALSE
WritePm_M_Err	BOOL	FALSE
WritePm_M_ErrID	WORD	FALSE

变量名	数据类型	当前值
ReadPm_M2	DMC_ReadParameter_ Motion	
ReadPm_M2_Done	BOOL	TRUE
ReadPm_M2_Bsy	BOOL	FALSE
ReadPm_M2_Act	BOOL	FALSE
ReadPm_M2_Err	BOOL	FALSE
ReadPm_M2_ErrID	WORD	FALSE
ReadPm_M2_DaTy	USINT	2
ReadPm_M2_Dat	UDINT	1000



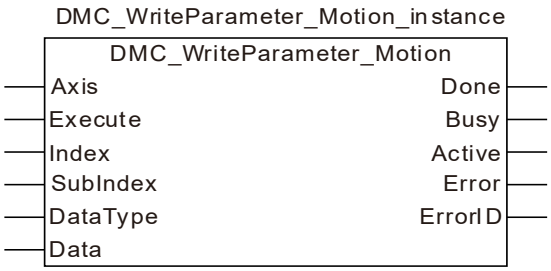
2. 时序图及说明



- 当 ReadPm_M1_Ex 由 FALSE 变为 TRUE 时，第一个 DMC_ReadParameter_Motion 指令开始执行，执行完成时 ReadPm_M1_Done 变为 TRUE，ReadPm_M1_DaTy =2，ReadPm_M1_Dat=5000，即读取伺服从站参数 P1-55 的内容为 5000（伺服最大速度限制为 5000rpm）。
- 当 ReadPm_M1_Done 由 FALSE 变为 TRUE 时，DMC_WriteParameter_Motion 指令开始执行，执行完成时 WritePm_M_Done 变为 TRUE，即写入伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）成功。
- 当 WritePm_M_Done 由 FALSE 变为 TRUE 时，第二个 DMC_ReadParameter_Motion 指令开始执行，执行完成时 ReadPm_M2_Done 变为 TRUE，ReadPm_M2_DaTy =2，ReadPm_M2_Dat=1000，即读取伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）。

11.3.20 DMC_WriteParameter_Motion（写入参数指令）

FB/FC	说明	适用机种
FB	此指令用于设定从站的参数值。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （从站的站号）	设定指令欲控制的从站	USINT	请参考第 2.2 节功能简介 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当*Execute*由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
Index(索引)	写入参数的索引	UINT	-	-
SubIndex （子索引）	写入参数的子索引	USINT	-	-
Data Type （数据类型）	写入参数的数据类型 1：代表写入的参数为 Byte 类型 2 :代表写入的参数为 Word 类型 4：代表写入的参数为 Double Word 类型	USINT	-	-
Data （参数值）	写入参数的内容值	UDINT	-	-

注：

1. 数据类型（Data Type）必须为写入参数的数据类型,若该值填写不正确，指令会报错。
2. 关于伺服轴的索引和子索引的计算方法请参考第 9 章。

- **输出参数**

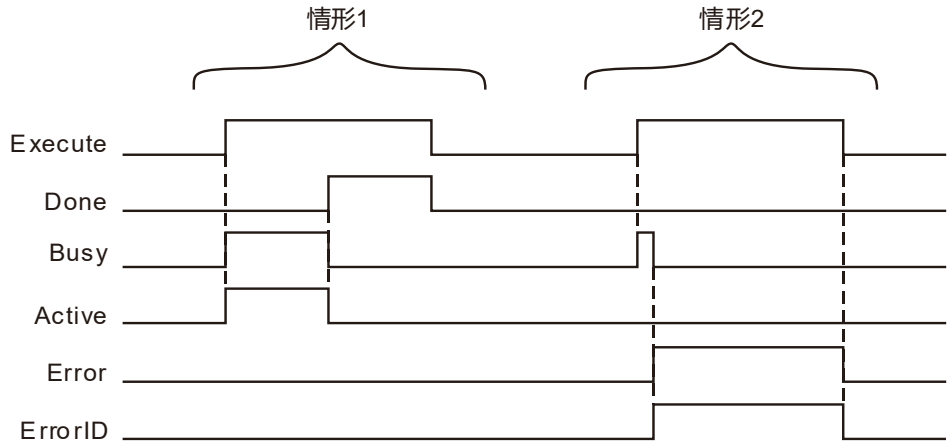
名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制从站	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当参数内容写入完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Busy 变为 FALSE
Active	◆ 当指令开始控制从站时	◆ 当 Error 为 TRUE 时 ◆ 当 Done 由 FALSE 变为 TRUE 时，Active 变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 后， *Busy* 和 *Active* 变为 TRUE， 且一个周期后，Done 变为 TRUE，Done 变为 TRUE 后，同一个周期内 Busy 和 Active 变为 FALSE，当 *Execute* 由 TRUE 变为 FALSE 后,Done 由 TRUE 变为 FALSE。

情形2： 指令执行之前，输入参数非法（如轴号为 0）。当 *Execute* 由 FALSE 变为 TRUE 后，Error 由 FALSE 变为 TRUE，Error ID 显示相应的错误代码。当 *Execute* 由 TRUE 变为 FALSE 后,Error 由 TRUE 变为 FALSE，Error ID 的内容清零。

- **功能说明**

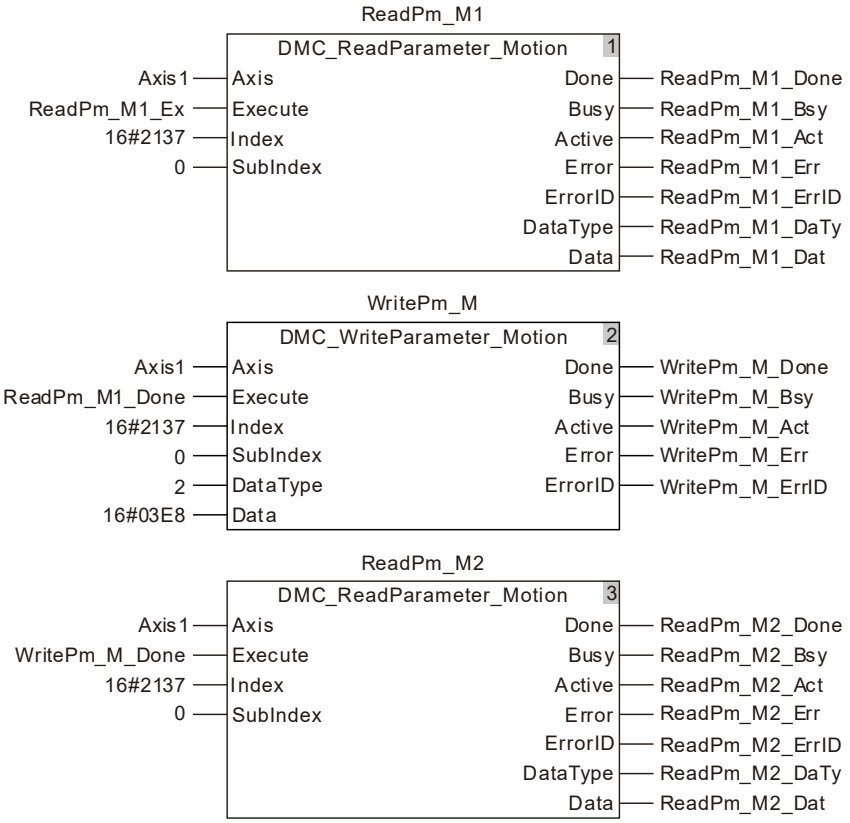
此指令用于设定从站的参数值，用户可以指定欲设定参数的索引（Index）和子索引（Sub-Index）。

程序范例

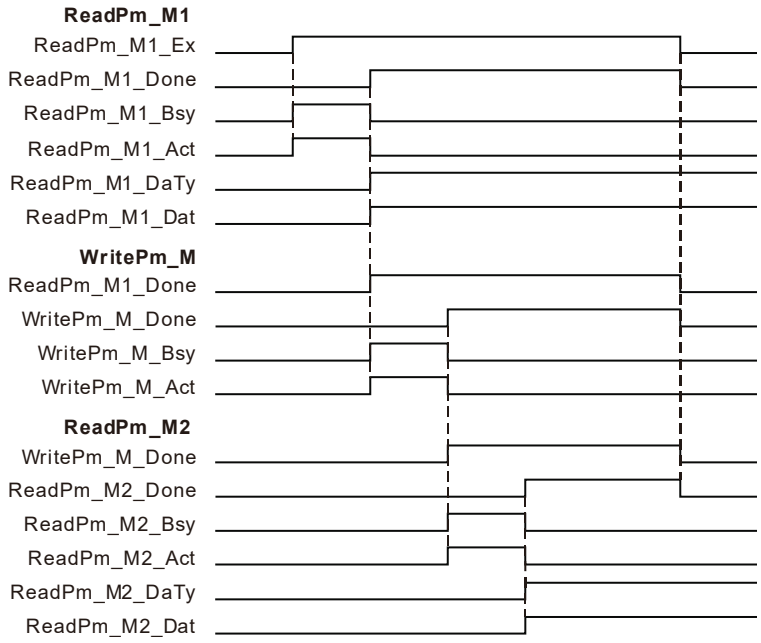
DMC_WriteParameter_ Motion 指令执行范例如下所示：

1. 变量和程序

变量名	数据类型	当前值
ReadPm_M1	DMC_ReadParameter_Motion	
Axis1	USINT	1
ReadPm_M1_Ex	BOOL	TRUE
ReadPm_M1_Done	BOOL	TRUE
ReadPm_M1_Bsy	BOOL	FALSE
ReadPm_M1_Act	BOOL	FALSE
ReadPm_M1_Err	BOOL	FALSE
ReadPm_M1_ErrID	WORD	FALSE
ReadPm_M1_DaTy	USINT	2
ReadPm_M1_Dat	UDINT	5000
WritePm_M	DMC_WriteParameter_Motion	
WritePm_M_Done	BOOL	TRUE
WritePm_M_Bsy	BOOL	FALSE
WritePm_M_Act	BOOL	FALSE
WritePm_M_Err	BOOL	FALSE
WritePm_M_ErrID	WORD	FALSE
ReadPm_M2	DMC_ReadParameter_Motion	
ReadPm_M2_Done	BOOL	TRUE
ReadPm_M2_Bsy	BOOL	FALSE
ReadPm_M2_Act	BOOL	FALSE
ReadPm_M2_Err	BOOL	FALSE
ReadPm_M2_ErrID	WORD	FALSE
ReadPm_M2_DaTy	USINT	2
ReadPm_M2_Dat	UDINT	1000



2. 时序图及说明

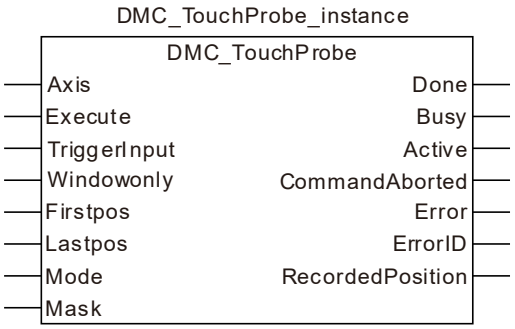


❖ 当 ReadPm_M1_Ex 由 FALSE 变为 TRUE 时，第一个 DMC_ReadParameter_Motion 指令开始执行，执行完成时 ReadPm_M1_Done 变为 TRUE， ReadPm_M1_DaTy=2，ReadPm_M1_Dat=5000，即读取伺服从站参数 P1-55 的内容为 5000（伺服最大速度限制为 5000rpm）。

- ❖ 当 ReadPm_M1_Done 由 FALSE 变为 TRUE 时，DMC_WriteParameter_Motion 指令开始执行，执行完成时 WritePm_M_Done 变为 TRUE，即写入伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）成功。
- ❖ 当 WritePm_M_Done 由 FALSE 变为 TRUE 时，第二个 DMC_ReadParameter_Motion 指令开始执行，执行完成时 ReadPm_M2_Done 变为 TRUE， ReadPm_M2_DaTy =2，ReadPm_M2_Dat=1000，即读取伺服从站参数 P1-55 的内容为 1000（伺服最大速度限制为 1000rpm）。

11.3.21 DMC_TouchProbe（位置捕获指令）

FB/FC	说明	适用机种
FB	此指令用于轴的位置捕获。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲操作的轴	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 为 TRUE 时
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
TriggerInput （触发位）	指定 DVP-15MC 系列运动控制器输入点 I0~I7、I10~I17 中的一个为位置捕获的触发位，引脚输入值 0~15 对应输入点 I0~I7、I10~I17。 该引脚在模式（Mode）等于 0 和 1 时有效，模式（Mode）等于 2、3、4 时无效。	MC_TriggerInput	0:mcTriggerinputI0 ... 7: mcTriggerinputI7 8:mcTriggerinputI10 ... 15: mcTriggerinputI17 （0）	-
Windowonly （窗口功能）	保留	-	-	-
Firstops （起始位置）	保留	-	-	-
Lastops （结束位置）	保留	-	-	-
Mode （模式）	模式0 :触发信号来自DVP-15MC系列运动控制器输入点I0~I7、I10~I17的上升沿，使用输入点由（TriggerInput）指定。通过触发位上升沿捕获位置。捕获的位置为控制器外部编码器接口接收脉冲数通过轴参数换算后的位置。 模式1 :触发信号来自DVP-15MC系列运动控制器输入点I0~I7、I10~I17的下降	INT	-	-

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	沿，使用输入点由触发位(TriggerInput)指定。捕获的位置为控制器外部编码器接口接收脉冲数通过轴参数换算后的位置。 模式2 :触发信号来自伺服驱动器的高速输入点DI7的上升沿，捕获的位置为伺服电机反馈给伺服驱动器的脉冲数通过轴参数换算后的位置。 模式3 :触发信号来自伺服驱动器的高速输入点DI7的上升沿，捕获的位置为伺服驱动器CN1端口接收脉冲数通过轴参数换算后的位置。 模式4 :触发信号来自伺服驱动器的高速输入点DI7的上升沿，捕获的位置为伺服驱动器CN5端口接收脉冲数通过轴参数换算后的位置。			
Mask （遮蔽）	保留	-	-	-

注：

1. 模式0和模式1不能同时使用同一个输入点进行位置捕获。
2. 模式2、3、4不能同时对同一个轴进行捕获。

- **输出参数**

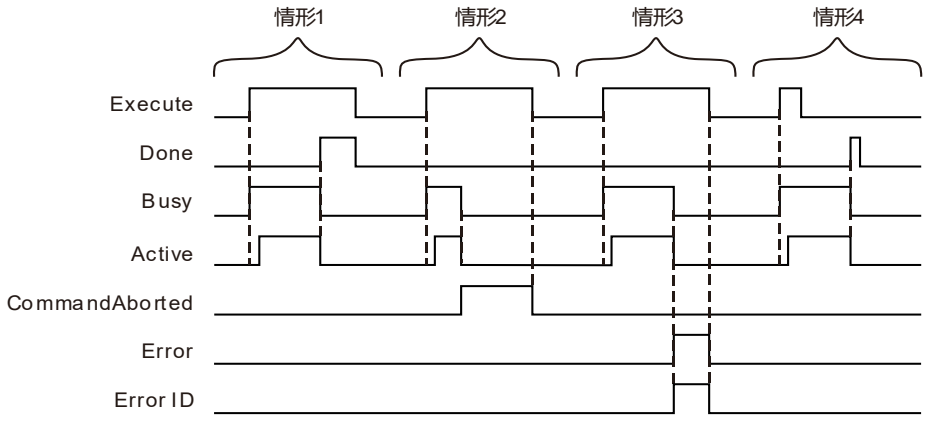
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该指令参数为 TRUE 时表示指令执行中被中断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
RecordedPosition （捕获位置）	位置捕获执行完成后捕获到的位置，详细说明请参考功	LREAL	-

名称	功能	数据类型	输出范围
	能说明部分 1。		

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 指令执行完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Active	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 指令执行的过程中被中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Execute* 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE；当定位完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 时，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 时，当有错误产生时(如轴报警或者掉线时)，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 Busy 和 Active 变为 FALSE ;当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中， *Execute* 由 TRUE 变为 FALSE 后， 指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

1. 位置捕获指令捕获到的位置（RecordedPosition）由其它值根据轴参数换算而来，捕获位置值的换算数据来源如下表所示。

模式	数据来源
模式 0、模式 1	DVP-15MC 系列运动控制器外部编码器接口接收的脉冲数
模式 2	伺服电机反馈给伺服驱动器的脉冲数
模式 3	伺服驱动器 CN1 端口输入端 pulse、/pulse、sign、/sign 接收的脉冲数。
模式 4	伺服驱动器 CN5 端口输入端 A 、/A、B 、/B 接收的脉冲数。

   - 当模式为0,1或2时，数据来源值的范围为-2147483648~2147483647，当数据来源值超过2147483647时，数据来源值变为-2147483648，RecordedPosition值的正负符号不会跟随数据来源的正负符号发生变化，RecordedPosition值继续增加。
   - 当模式为3或4时,数据来源值范围为-2147483648~2147483647 ,当数据来源值超过2147483647时，数据来源值变为-2147483648，RecordedPosition值也会从正数变为负数。

2. 位置捕获指令捕获的位置是根据轴参数换算而来，模式不同时，换算数据来源不同。当轴参数中“伺服齿轮比设置”和“机构齿轮比设置”如下图所示，位置捕获指令模式（Mode）等于3时（请参考下面模式3介绍），CN1端口pulse、/pule、sign、/sign接收的脉冲数为435， 则位置捕获指令捕获到的位置为65.25，计算方法：435×（3×1000）÷（2×10000）=65.25。计算方法中的10000、2、3、1000和下面左图所示的10000、2、3、1000一一对应。其它模式时，该指令捕获到的位置（Positon）计算方法和上述方法相同，只是数据来源不同。

伺服齿轮比设置	机构齿轮比设置
电子齿轮比分子：128	齿轮箱输出：3
电子齿轮比分母：1	齿轮箱输入：2
脉冲数 / 转：10000	机构导程：1000单元 / 转

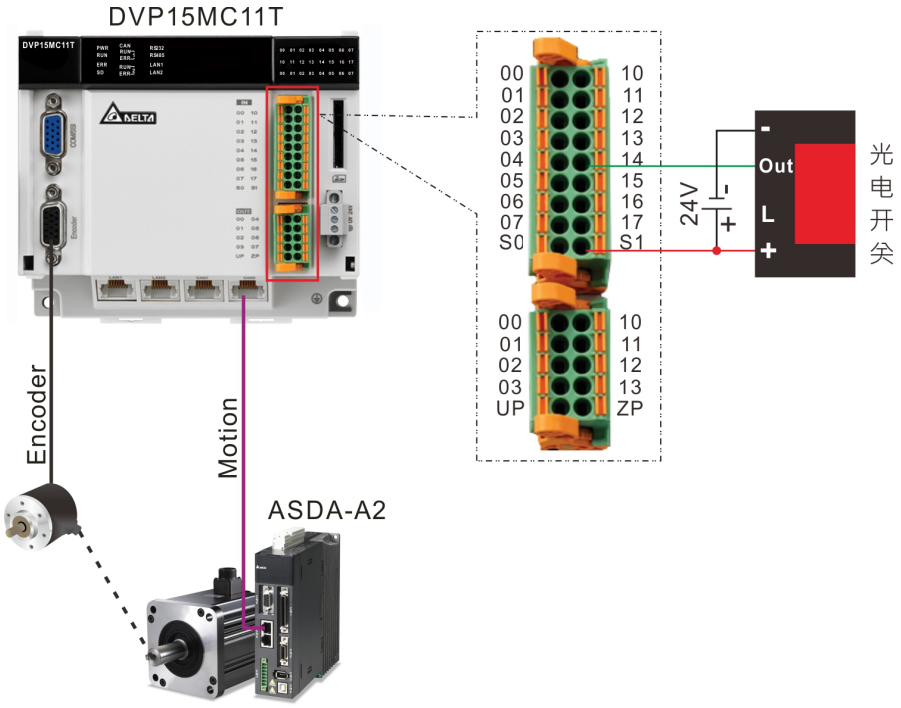
3. 位置捕获指令Mode=0或1时，捕获的位置值也可以按上述方法计算，实际应用时一般通过构建编码器轴进行位置捕获。DVP-15MC系列运动控制器外部编码器接口接收的脉冲数为638时，则位置捕获指令捕获到的位置为95.4，计算方法：638×（3×1000）÷（2×10000）=95.4。计算方法中的1000是编码器中机构导成，2是齿轮箱输入，3是齿轮箱输出，10000是脉冲数/转。I0由FALSE变TRUE 一次，执行一次位置捕获。

- **接线示意图**

  - **模式0、1**

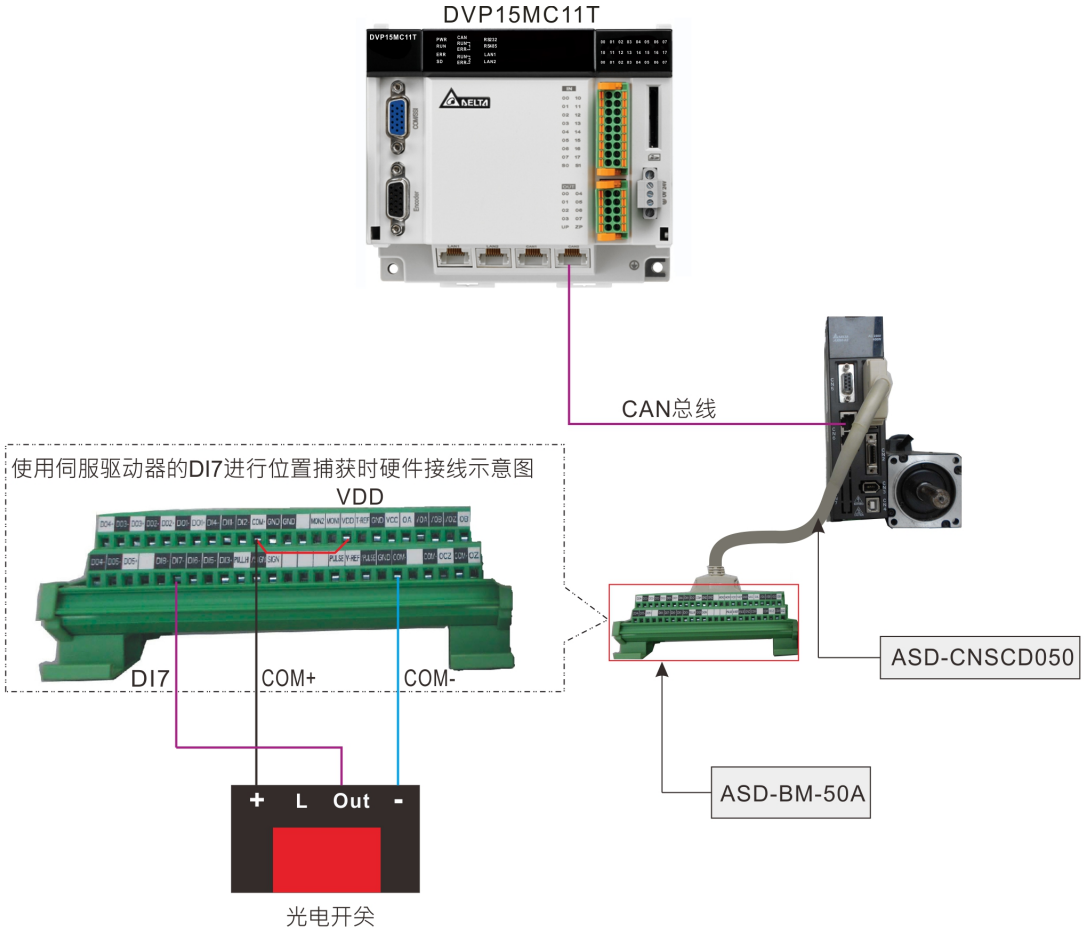
模式 0：外部信号触发 DVP-15MC 系列运动控制器的 I 点，通过触发位上升沿捕获位置，捕获的位置为控制器外部编码器接口接收脉冲数通过轴参数换算后的位置。

模式 1：外部信号触发 DVP-15MC 系列运动控制器的 I 点，通过触发位下降沿捕获位置，捕获的位置为控制器外部编码器接口接收脉冲数通过轴参数换算后的位置。



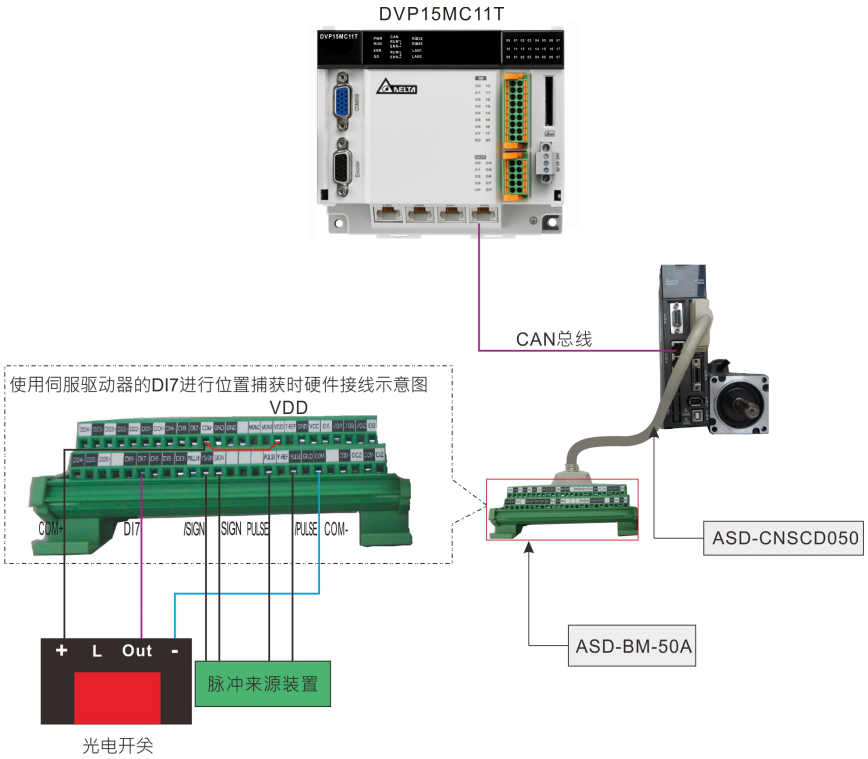
■ **模式2**

外部信号触发驱动器的高速输入点 DI7，捕获的位置为伺服电机反馈给伺服驱动器的脉冲数通过轴参数换算后的位置。



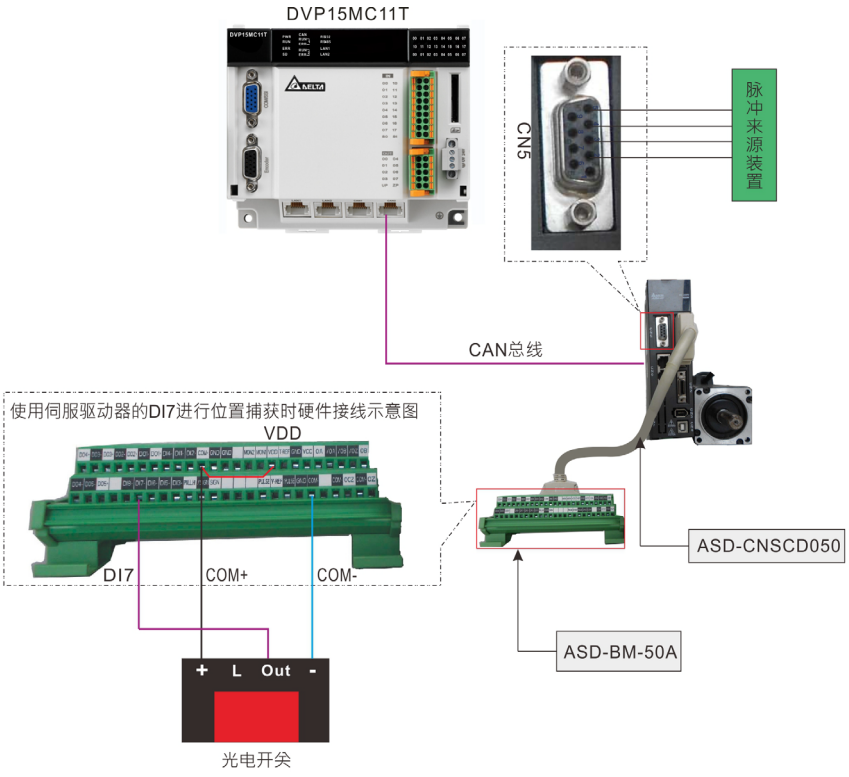
■ **模式3**

外部信号触发驱动器的高速输入点DI7。捕获的位置为伺服驱动器CN1端口接收脉冲数通过轴参数换算后的位置。



■ **模式4**

外部信号触发驱动器的高速输入点DI7。捕获的位置为伺服驱动器CN5端口接收脉冲数通过轴参数换算后的位置。

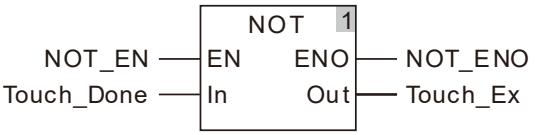


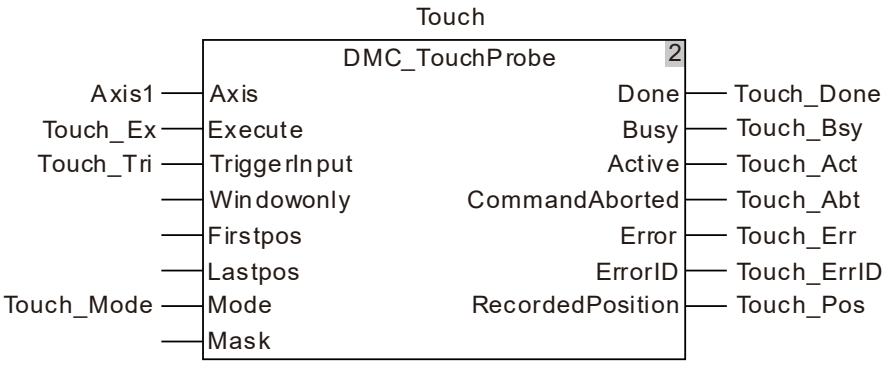
程序范例一

Mode0 情况下使用 I0 上升沿捕获控制器外部编码器轴位置。

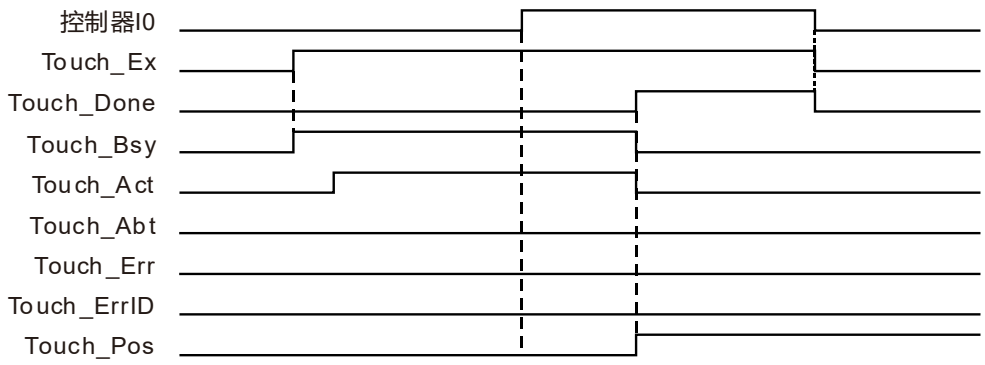
1. 变量和程序

变量名	数据类型	初始值
NOT_EN	BOOL	FALSE
NOT_ENO	BOOL	
Touch	DMC_TouchProbe	
Axis1	USINT	3
Touch_Ex	BOOL	FALSE
Touch_Tri	MC_Triggerinput	0
Touch_Mode	INT	0
Touch_Done	BOOL	
Touch_Bsy	BOOL	
Touch_Act	BOOL	
Touch_Abt	BOOL	
Touch_Err	BOOL	
Touch_ErrID	UINT	
Touch_Pos	LREAL	





2. 时序图



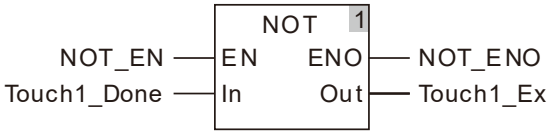
- Touch_Ex 由 FALSE 变为 TRUE，Touch_Bsy 在第一个周期由 FALSE 变为 TRUE，第二个周期 Touch_Act 由 FALSE 变为 TRUE。
- 当外部信号触发控制器 I0 捕获指令执行完成后，Touch_Done 由 FALSE 变为 TRUE，Touch_Pos 输出位置为控制器外部编码器接口接收脉冲数通过轴参数换算后的位置，同时 Touch_Bsy 和 Touch_Act 由 TRUE 变为 FALSE，当 Touch_Ex 由 TRUE 变为 FALSE 时，同时 Touch_Done 由 TRUE 变为 FALSE，且 Touch_Pos 捕获到的位置不会清 0。

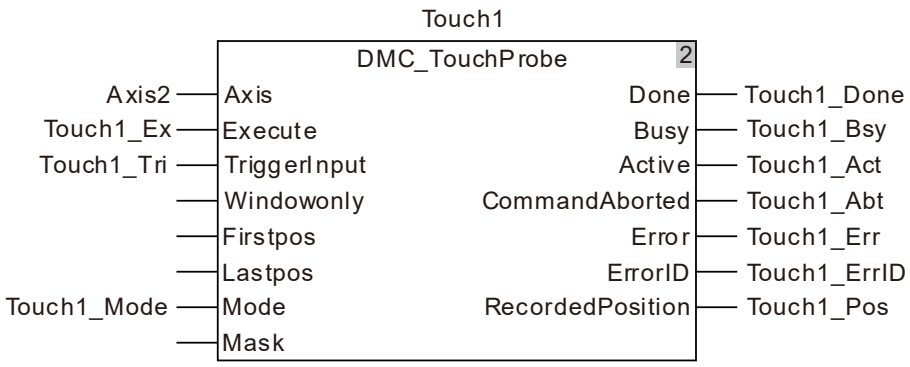
程序范例二

Mode2 情况下外部信号触发伺服 CN1 端子 DI7 端口，捕获伺服电机反馈给伺服驱动器的脉冲数通过轴参数换算后的位置。

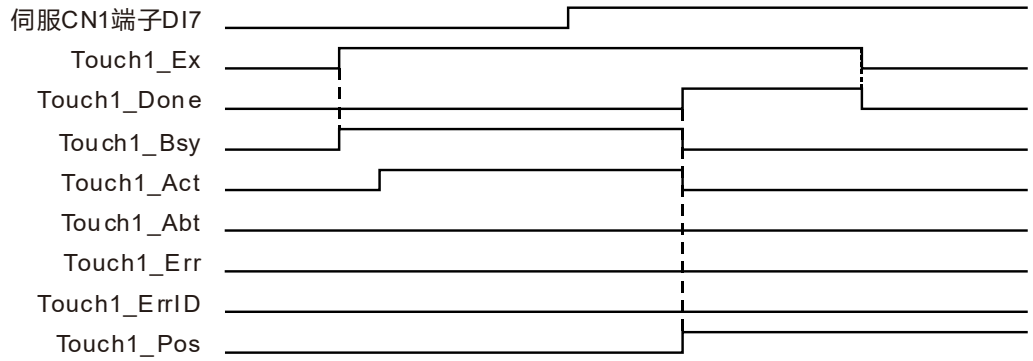
1. 变量和程序

变量名	数据类型	初始值
NOT_EN	BOOL	FALSE
NOT_ENO	BOOL	
Touch1	DMC_TouchProbe	
Axis2	USINT	1
Touch1_Ex	BOOL	FALSE
Touch1_Tri	MC_Triggerinput	
Touch1_Mode	INT	2
Touch1_Done	BOOL	
Touch1_Bsy	BOOL	
Touch1_Act	BOOL	
Touch1_Abt	BOOL	
Touch1_Err	BOOL	
Touch1_ErrID	UINT	
Touch1_Pos	LREAL	





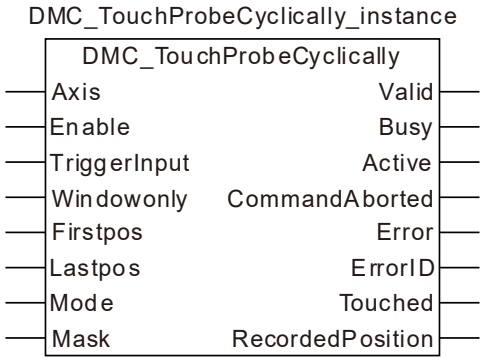
2. 时序图



- ❖ Touch1_Ex 由 FALSE 变为 TRUE，Touch1_Bsy 在第一个周期由 FALSE 变为 TRUE，第二个周期 Touch1_Act 由 FALSE 变为 TRUE。
- ❖ 当外部信号触发伺服 CN1 端子的 DI7 端口，当捕获指令执行完，Touch1_Done 由 FALSE 变为 TRUE，Touch1_Pos 输出伺服电机反馈给伺服驱动器的脉冲数通过轴参数换算后的位置，同时 Touch1_Bsy 和 Touch1_Act 由 TRUE 变为 FALSE，当 Touch1_Ex 由 TRUE 变为 FALSE 时，同时 Touch1_Done 由 TRUE 变为 FALSE，且 Touch1_Pos 捕获到的位置不会清 0。

11.3.22 DMC_TouchProbeCyclically（循环位置捕获指令）

FB/FC	说明	适用机种
FB	此指令用于轴位置循环捕获。	DVP15MC11T DVP15MC11T-06

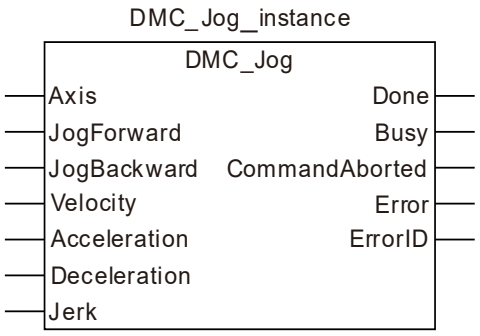


- **功能说明**

此指令用于轴位置循环捕获。该指令功能和 DMC_TouchProbe 指令捕获功能一样，区别在于该指令当 Enable 为 TRUE 时，指令对轴位置可以循环捕获，即检测到外部信号时，可以立即执行位置捕获，不需要重新触发。该指令各个参数的含义和数据类型请参考 DMC_TouchProbe 指令说明。

11.3.23 DMC_Jog（点动指令）

FB/FC	说明	适用机种
FB	此指令用于点动功能。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	JogForward 或 JogBackward 为 TRUE 时
JogForward （正转执行位）	当 JogForward 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
JogBackward （反转执行位）	当 JogBackward 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
Velocity （速度）	设定的目标速度 （单位: 单元/秒）	LREAL	正数 （不可缺省）	JogForward 或 JogBackward 为 TRUE 时
Acceleration （加速度）	设定的目标加速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	JogForward 或 JogBackward 为 TRUE 时
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	JogForward 或 JogBackward 为 TRUE 时
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	JogForward 或 JogBackward 为 TRUE 时

- **输出参数**

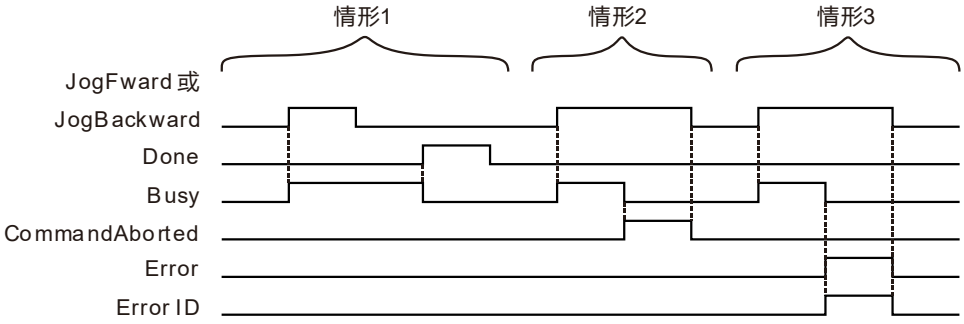
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示点动停止。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当点动停止时 Done 位 TRUE 一个周期	◆ 点动停止一个周期后 Done 变为 FALSE
Busy	◆ 当 JogForward 或 JogBackward 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ JogForward 或 JogBackward 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ JogForward 或 JogBackward 由 TRUE 变为 FALSE 时

- **输出参数变化时序图:**



情形1： 当 *JogForward* 或 *JogBackward* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE。当点动停止，轴速度降为 0 后，*Done* 变为 TRUE，同时 *Busy* 变为 FALSE。

情形2： 当 *JogForward* 或 *JogBackward* 由 FALSE 变为 TRUE，该指令被其它指令中断后，*CommandAborted* 变为 TRUE，同时 *Busy* 变为 FALSE；当 *JogForward* 或 *JogBackward* 由 TRUE 变为 FALSE 后，*CommandAborted* 变为 FALSE。

情形3： 当 *JogForward* 或 *JogBackward* 由 FALSE 变为 TRUE 后，当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 变为 FALSE；当 *JogForward* 或 *JogBackward* 由 TRUE 变为 FALSE 后，*Error* 变为 FALSE，同时 ErrorID 值清零。

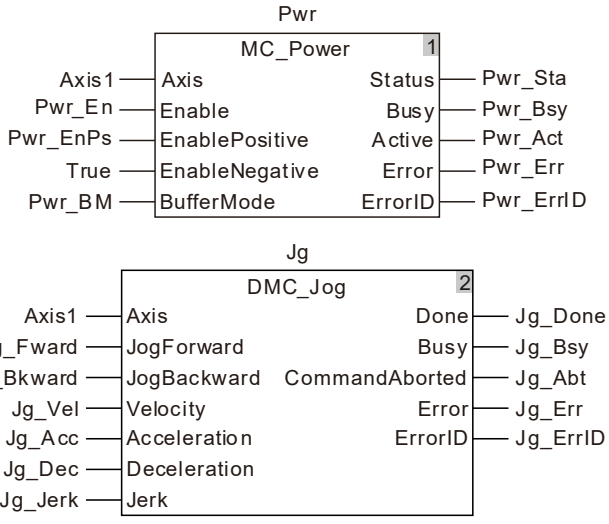
- **功能说明**

此指令用于点动功能。JogForward 引脚控制轴正向运转，JogBackward 引脚控制轴反向运转。当点动停止时，Done 位 TRUE 一个周期后变为 FALSE，同时 Busy 为 FALSE。

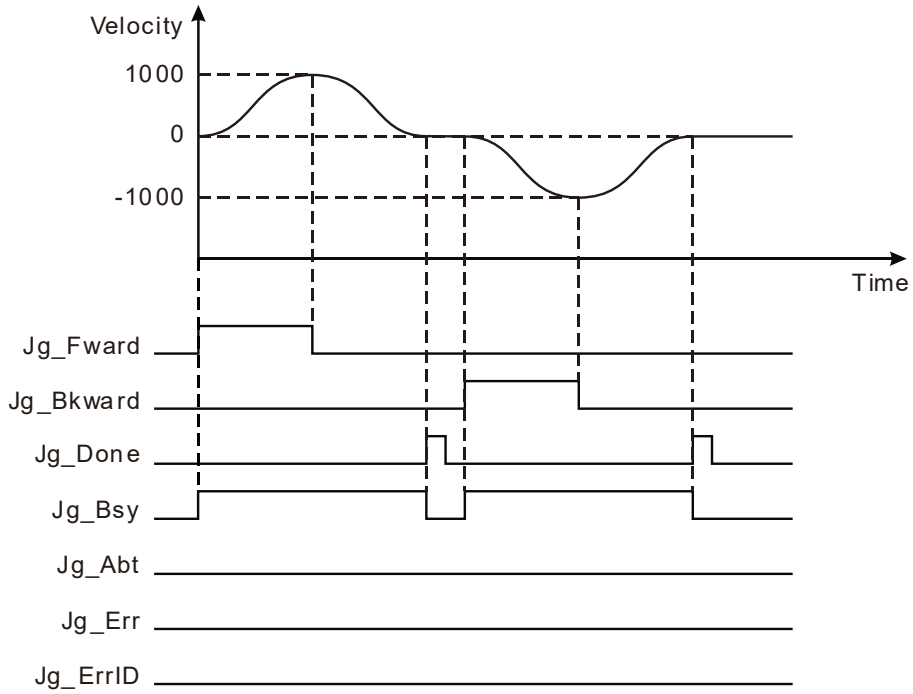
程序范例

1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Jg	DMC_Jog	
Jg_Fward	BOOL	FALSE
Jg_Bkward	BOOL	FALSE
Jg_Vel	LREAL	1000.0
Jg_Acc	LREAL	500.0
Jg_Dec	LREAL	500.0
Jg_Jerk	LREAL	500.0
Jg_Done	BOOL	
Jg_Bsy	BOOL	
Jg_Abt	BOOL	
Jg_Err	BOOL	
Jg_ErrID	WORD	



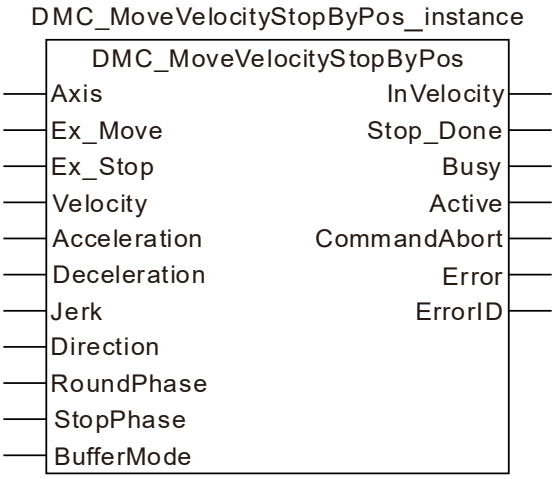
2. 运动曲线和时序图



- ❖ 当 Jg_Fward 由 FALSE 变为 TRUE 时，DMC_Jog 指令开始执行，同时 Jg_Bsy 由 FALSE 变为 TRUE，轴开始正向运转；将 Jg_Fward 由 TRUE 变为 FALSE，轴开始减速，当速度减为 0 时，Jg_Bsy 变为 FALSE，同时 Jg_Done 位 由 FALSE 变为 TRUE 一个周期后变为 FALSE。
- ❖ 当 Jg_Bkward 由 FALSE 变为 TRUE 时，DMC_Jog 指令开始执行，同时 Jg_Bsy 由 FALSE 变为 TRUE，轴开始反向运转；将 Jg_Bkward 由 TRUE 变为 FALSE，轴开始减速，当速度减为 0 时，Jg_Bsy 位变为 FALSE，同时 Jg_Done 位 由 FALSE 变为 TRUE 一个周期后变为 FALSE。

11.3.24 DMC_MoveVelocityStopByPos（指定相位停止指令）

FB/FC	说明	适用机种
FB	此指令用于指定轴运转停止在某一相位。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	Ex_Move 为 TRUE 时
Ex_Move （执行位）	当 Ex_Move 由 FALSE 变 TRUE 时，指令控制轴运动。	BOOL	TRUE 或 FALSE （FALSE）	-
Ex_Stop （停止位）	当 Ex_Stop 由 FALSE 变 TRUE 时，指令控制轴停止。	BOOL	TRUE 或 FALSE （FALSE）	-
Velocity （速度）	设定的目标速度 （单位: 单元/秒）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Acceleration （加速度）	设定的目标加速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Direction （方向）	设定的运转方向 1: 正方向 3: 反方向	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection （1）	Ex_Move 为 TRUE 时
RoundPhase (模)	机构导程对应的模	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时

名称	功能	数据类型	设定范围 （缺省值）	生效时机
StopPhase （停止相位）	指定模中的某一相位	LREAL	0~RoundPhase 设定值 （0）	Ex_Move 为 TRUE 时
BufferMode （交接模式）	设定两个指令之间交接模式 0：打断 1：等待 2：以低速交接 3：以前一指令的速度交接 4：以后一指令的速度交接 5：以高速交接	MC_Buffer_Mode	0：mcAborting 1：mcBuffered 2：mcBlendingLow 3：mcBlendingPrevious 4：mcBlendingNext 5：mcBlendingHigh （0）	Ex_Move 为 TRUE 时

- **输出参数**

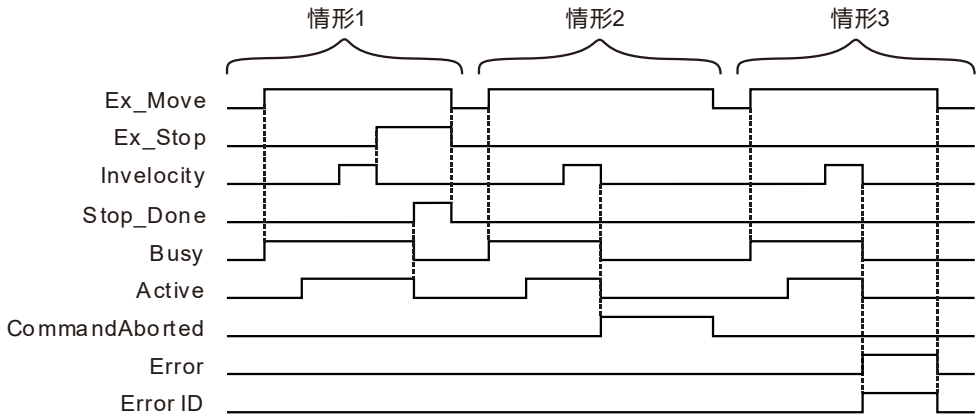
名称	功能	数据类型	输出范围
Invelocity （速度到达）	该输出参数为 TRUE 时表示速度到达	BOOL	TRUE / FALSE
Stop_Done （停止完成）	该输出参数为 TRUE 时表示位置到达	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Invelocity	◆ 当速度到达时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当速度到达后改变输入参数， Ex_Move 再次由 FALSE 变为 TRUE 时，Invelocity 会立即变为 FALSE；当速度完成后不改变输入参数 Ex_Move 再次由 FALSE 变为 TRUE 时 Invelocity 会立即变为 FALSE，且下个周期 Invelocity 变为 TRUE。
Stop_Done	◆ 当位置到达时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ Ex_Stop 为 FALSE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Busy	◆ 当 Ex_Move 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Ex_Move 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Ex_Move 和 Ex_Stop 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Ex_Move 和 Ex_Stop 由 TRUE 变为 FALSE 时

- **输出参数变化时序图:**



情形1： 当 *Ex_Move* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，当指令开始控制轴时，*Active* 变为 TRUE。当 Ex_Stop 由 FALSE 变为 TRUE 时，Invelocity 变为 FALSE，当位置到达后，*Stop_Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。当 Ex_Move 和 Ex_Stop 由 TRUE 变为 FALSE 时，Stop_Done 变为 FALSE。

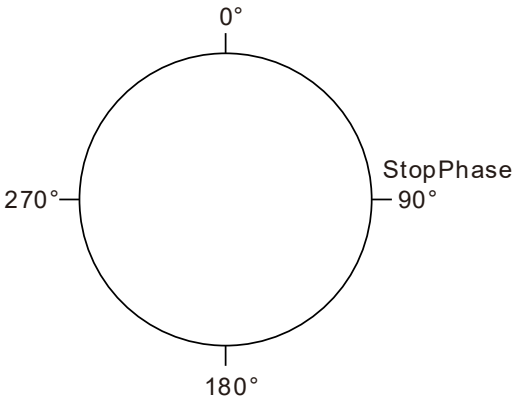
情形2： 当 *Ex_Move* 为 TRUE 时，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Invelocity*、*Busy* 和 *Active* 变为 FALSE，当 Ex_Move 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Ex_Move* 为 TRUE 时，当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Invelocity*、*Busy* 和 *Active* 变为 FALSE；当 *Ex_Move* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

- **功能说明**

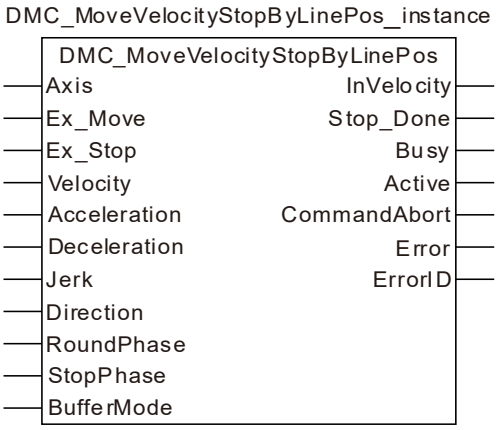
此指令用于控制轴运转后指定停止在某一相位。RoundPhase 为机构导程对应的模，StopPhase 为指定模中的某一相位， StopPhase 要小于 RoundPhase 的值。Ex_Move 执行位从 FALSE 变为 TRUE 时，控制轴运转；Ex_Stop 从 FALSE 变为 TRUE 时，控制轴停止在 StopPhase 指定的相位，轴最终停止的位置为机构导程的整数倍+StopPhase/RoundPhase*机构导程的值。

如下图所示，机构导程为 10000，RoundPhase 为 360，StopPhase 为 90，可以通过该指令控制轴停止在 StopPhase 指定的相位处。终端执行机构可能停止在 12500 单元，22500 单元，32500 单元，42500 单元处。



11.3.25 DMC_MoveVelocityStopByLinePos（指定位置停止指令）

FB/FC	说明	适用机种
FB	此指令用于指定轴运转时停止在某一位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	Ex_Move 为 TRUE 时
Ex_Move （执行位）	当 Ex_Move 由 FALSE 变 TRUE 时，指令控制轴运动。	BOOL	TRUE 或 FALSE （FALSE）	-
Ex_Stop （停止位）	当 Ex_Stop 由 FALSE 变 TRUE 时，指令控制轴停止。	BOOL	TRUE 或 FALSE （FALSE）	-
Velocity （速度）	设定的目标速度 （单位：单元/秒）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Acceleration （加速度）	设定的目标加速度 （单位：单元/秒^2）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Deceleration （减速度）	设定的目标减速度 （单位：单元/秒^2）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位：单元/秒^3）	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
Direction （方向）	设定的运转方向 1：正方向 3：反方向	MC_Direction	1: mcPositiveDirection 3: mcNegativeDirection （1）	Ex_Move 为 TRUE 时
RoundPhase (每段位置)	指定的模	LREAL	正数 （不可缺省）	Ex_Move 为 TRUE 时
StopPhase （停止位置）	指定模中的某一位置	LREAL	0~RoundPhase 设定值 （0）	Ex_Move 为 TRUE 时

名称	功能	数据类型	设定范围 （缺省值）	生效时机
BufferMode （交接模式）	设定两个指令之间交接模式 0：打断 1：等待 2：以低速交接 3：以前一指令的速度交接 4：以后一指令的速度交接 5：以高速交接	MC_Buffer_Mode	0：mcAborting 1：mcBuffered 2：mcBlendingLow 3：mcBlendingPrevious 4：mcBlendingNext 5：mcBlendingHigh （0）	Ex_Move 为 TRUE 时

- **输出参数**

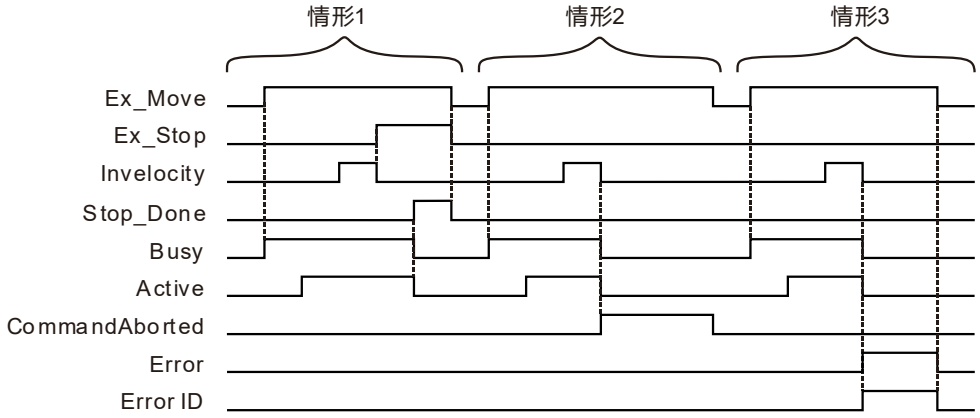
名称	功能	数据类型	输出范围
Invelocity （速度到达）	该输出参数为 TRUE 时表示速度到达	BOOL	TRUE / FALSE
Stop_Done （停止完成）	该输出参数为 TRUE 时表示位置到达	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Invelocity	◆ 当速度到达时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当速度到达后改变输入参数， Ex_Move 再次由 FALSE 变为 TRUE 时，Invelocity 会立即变为 FALSE；当速度完成后不改变输入参数 Ex_Move 再次由 FALSE 变为 TRUE 时 Invelocity 会立即变为 FALSE，且下个周期 Invelocity 变为 TRUE。
Stop_Done	◆ 当位置到达时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ Ex_Stop 为 FALSE 时
Busy	◆ 当 Ex_Move 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Active	◆ 当指令开始控制轴时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Ex_Move 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Ex_Move 和 Ex_Stop 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Ex_Move 和 Ex_Stop 由 TRUE 变为 FALSE 时

- **输出参数变化时序图:**



情形1： 当 *Ex_Move* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，当指令开始控制轴时，*Active* 变为 TRUE。当 Ex_Stop 由 FALSE 变为 TRUE 时，Invelocity 变为 FALSE，当位置到达后，*Stop_Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。当 Ex_Move 和 Ex_Stop 由 TRUE 变为 FALSE 时，Stop_Done 变为 FALSE。

情形2： 当 *Ex_Move* 为 TRUE 时，该指令被其它指令中断时，*CommandAborted* 变为 TRUE，同时 *Invelocity*、*Busy* 和 *Active* 变为 FALSE，当 Ex_Move 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

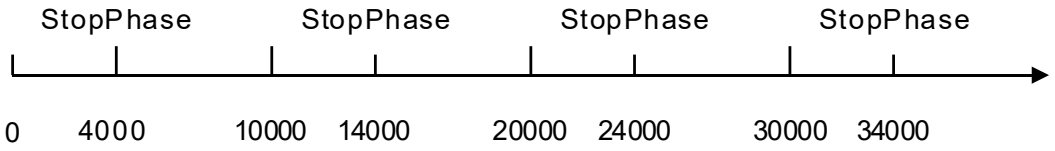
情形3： 当 *Ex_Move* 为 TRUE 时，当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Invelocity*、*Busy* 和 *Active* 变为 FALSE；当 *Ex_Move* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

- **功能说明**

此指令用于控制轴运转后停止在某一位置。RoundPhase 为指定的模，StopPhase 为指定模中的某一位置，StopPhase 要小于 RoundPhase 的值。StopPhase 和 RoundPhase 的单位为单元，和机构导程的单位相同。

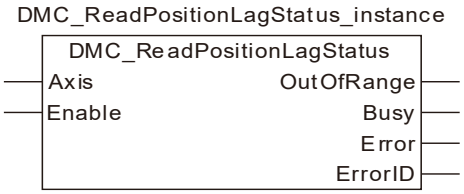
Ex_Move 执行位从 FALSE 变为 TRUE 时，指令控制轴运转；Ex_Stop 停止位从 FALSE 变为 TRUE 时，指令控制轴停止在 StopPhase 设定的位置，轴最终停止的位置为 RoundPhase 的整数倍+StopPhase 的值。

如下图所示，RoundPase 为 10000，StopPhase 为 4000，可以通过该指令控制终端执行机构停止在 StopPhase 指定的位置处，终端执行机构可能停在 4000 单元，14000 单元，24000 单元，或者 34000 单元处。



11.3.26 DMC_ReadPositionLagStatus（位置误差检测指令）

FB/FC	说明	适用机种
FB	此指令用于位置误差检测。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	Enable 为 TRUE 时
Enable （执行位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	Enable 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
OutOfRange （超出范围）	该输出参数为 TRUE 时表示位置误差超出范围。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

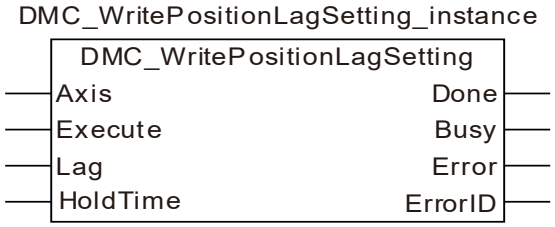
名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行成功时	◆ 当 Enable 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当该指令执行时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **功能说明**

此指令用于位置误差检测。通过指令 DMC_WritePositionLagSetting 设定位置误差值，当命令位置与实际位置差值大于误差设定值时，该指令输出位 OutOfRange 为 TRUE。

11.3.27 DMC_WritePositionLagSetting（位置误差设定指令）

FB/FC	说明	适用机种
FB	此指令用于位置误差设定。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节功能简介 （不可缺省）	Execute 由 FALSE 变为 TRUE
Execute （执行位）	当 Execute 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
Lag （位置误差）	位置误差设定。	LREAL	正数 （不可缺省）	Execute 由 FALSE 变为 TRUE
HoldTime （持续时间）	超过设定位置误差设定值的持续时间。（单位：秒）	LREAL	正数 （不可缺省）	Execute 由 FALSE 变为 TRUE

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

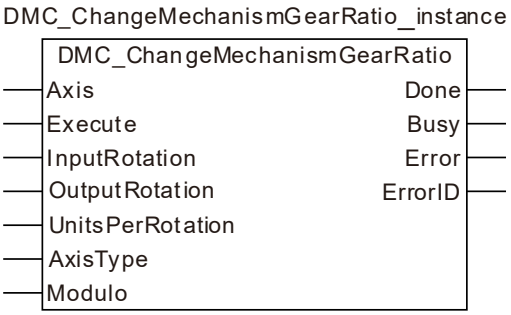
名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行成功时	◆ 当 Enalbe 位为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当该指令执行时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当异常情况解除时

- **功能说明**

  此指令用于位置误差设定。通过指令 DMC_ReadPositionLagStatus 检测位置误差值。

11.3.28 DMC_ChangeMechanismGearRatio（更改轴参数指令）

FB/FC	说明	适用机种
FB	此指令用于更改轴参数。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴	USINT	请参考第 2.2 节功能简介 （不可缺省）	Execute 由 FALSE 变为 TRUE
Execute （执行位）	当 Execute 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
InputRotation （齿轮箱输入）	与齿轮箱输出量一起确定机构齿轮比	LREAL	正整数 （不可缺省）	Execute 由 FALSE 变为 TRUE
OutputRotation （齿轮箱输出）	与齿轮箱输入量一起确定机构齿轮比	LREAL	正整数 （不可缺省）	Execute 由 FALSE 变为 TRUE
UnitsPerRotation （机构导程）	齿轮箱输出端转一圈对应终端执行机构移动的单元数目。 （单位：脉冲/圈）	LREAL	正数 （不可缺省）	Execute 由 FALSE 变为 TRUE
AxisType （轴类型）	轴类型 0：旋转轴 1：直线轴	USINT	0、1 （0）	Execute 由 FALSE 变为 TRUE
Modulo （模）	用以平分终端执行机构位置的周期	LREAL	正数 （不可缺省）	Execute 由 FALSE 变为 TRUE

- **输出参数**

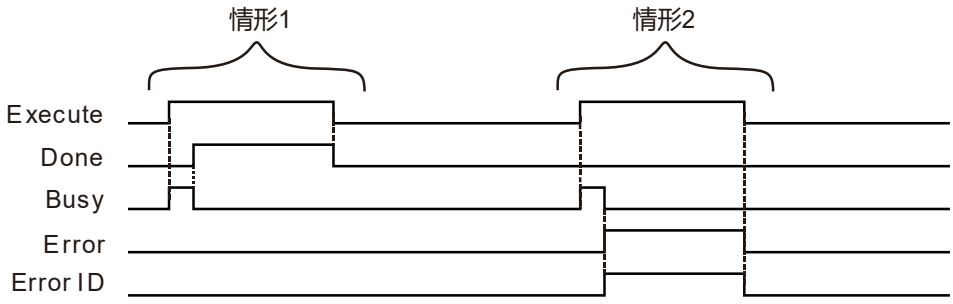
名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ Execute 位为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，当指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 位变为 FALSE。当 *Execute* 变为 FALSE 时，*Done* 变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

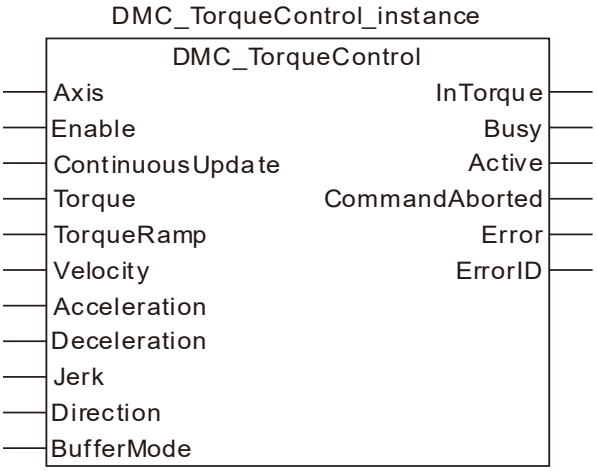
- **功能说明**

此指令用于变更终端执行机构参数。如当终端执行机构参数变更时，可以使用此指令更改轴参数和实际机构参数一致，方便用户使用。

该指令必须在状态机是 Disable 或者 Standstill 状态下执行，在其他状态下执行，指令报错。

11.3.29 DMC_TorqueControl（带速度限制的扭矩控制指令）

FB/FC	说明	适用机种
FB	此指令用于控制轴工作于扭矩模式进行扭矩输出。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Enable* 为 TRUE 时
Enable （执行位）	当*Enable*由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	*Enable* 为 TRUE 时
ContinuousUpdate	保留	-	-	-
Torque （目标扭矩）	此参数用于设置需要的扭矩大小，扭矩大小用伺服轴额定扭矩的千分比来表示，例如设定值为 30，则表示设定扭矩为额定扭矩的千分之三十。当执行位为 TRUE时，扭矩大小立即根据扭矩变化率变化。	INT	负数、正数、0 （0）	*Enable* 为 TRUE 时
TorqueRamp （扭矩变化率）	从当前设定扭矩到更改后设定扭矩的变化率（单位：千分比/秒）	LREAL	负数、正数 （不可缺省）	*Enable* 由 FALSE 变为 TRUE
Velocity （最大限制速度）	扭矩指令控制轴时，限制轴的速度不能超过该设定值 （单位: 单元/秒）	LREAL	正数 （不可缺省）	*Enable* 由 FALSE 变为 TRUE
Acceleration （加速度）	保留	-	-	-

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Deceleration （减速度）	保留	-	-	-
Jerk （加速度的变化率）	保留	-	-	-
Direction （方向）	设定轴的运转方向 1:正方向 3:反方向	MC_Direction	1: mcPositiveDirection、 3: mcNegativeDirection （不可缺省）	*Enable* 由 FALSE 变为 TRUE
BufferMode （交接模式）	保留	-	-	-

- **输出参数**

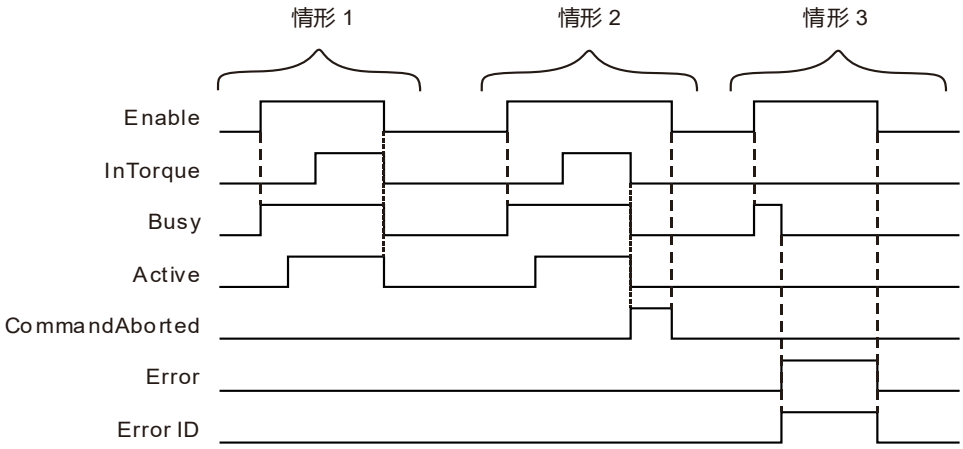
名称	功能	数据类型	输出范围
InTorque （目标扭矩到达）	该输出参数为 TRUE 时表示设定目标扭矩到达.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
InTorque	◆ 当扭矩到达时	◆ 当 *Error* 为 TRUE 时 ◆ 当 *Enable* 由 TRUE 变为 FALSE 时
Busy	◆ 当 *Enable* 为 TRUE 时	◆ 当 *InTorque* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 *InTorque* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，*Enable* 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，*CommandAborted* 被设置为 TRUE，且一个周期后，*CommandAborted* 变为 FALSE

名称	变为 TRUE 的时机	变为 FALSE 的时机
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 后，同一个周期 Busy 变为 TRUE，当指令开始控制轴时，Active 变为 TRUE， 设定目标扭矩达到时，InTorque 变为 TRUE。 当 Enable 由 TRUE 变为 FALSE 后， Busy、Active、Intorque 同一个周期变为 FALSE。

情形2： 当 Enable 由 FALSE 变为 TRUE 后，该指令被 MC_Stop 指令中断时，CommandAborted 变为 TRUE，同时 InTorque、 Busy 和 Active 变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，CommandAborted 变为 FALSE。

情形3： 指令执行之前，输入参数非法（如轴号为 0）。当 Enable 由 FALSE 变为 TRUE 后，同一个周期 Busy 变为 TRUE，且下一个周期后，Error 变为 TRUE，Busy 变为 FALSE，ErrorID 显示相应的错误代码。当 Enable 由 TRUE 变为 FALSE 后，Error 由 TRUE 变为 FALSE，Error ID 的值变为零。

- **功能说明**

此指令用于控制轴工作于扭矩模式进行扭矩输出。轴从当前扭矩到输出目标扭矩的过程中，根据指令设定的扭矩变化率（单位：千分比/秒）变化。在指令执行过程中，改变扭矩值，伺服的扭矩会按照扭矩变化率变化到更改后的扭矩值，然后保持此扭矩运动。指令执行过程中，该指令会控制轴的最大速度不超过设定的最大限制速度。

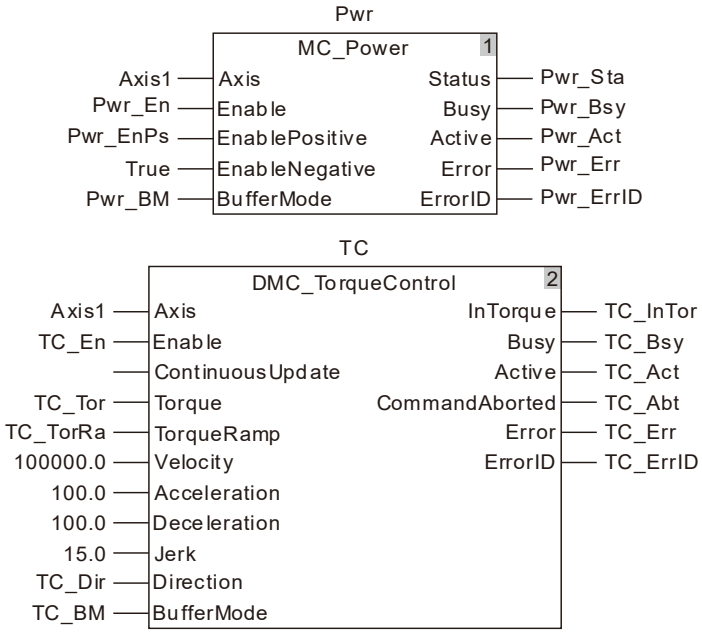
要停止本指令，需将 Enable 位变为 FALSE 或者使用 MC_Stop 指令强制打断此指令。当 Enable 由 TRUE 变为 FALSE 时，轴会立即退出扭矩模式，扭矩会立即变化（扭矩变化率无效）。若需要扭矩按照扭矩变化率缓慢停止，只需将目标扭矩写成 0 即可，待轴实际输出扭矩变成 0 后，再将 Enable 变成 FALSE。

程序范例

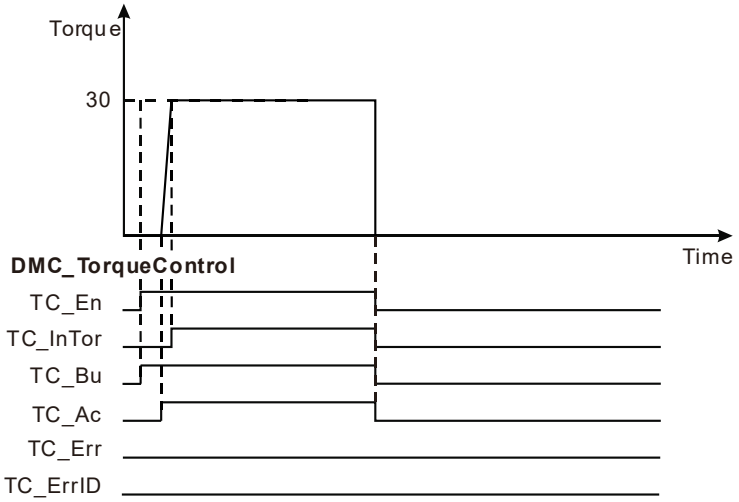
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	

变量名	数据类型	初始值
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
TC	DMC_TorqueControl	
TC_En	BOOL	FALSE
TC_Tor	INT	30
TC_TorRa	LREAL	5.0
TC_Dir	MC_Direction	1
TC_InTor	BOOL	
TC_Bsy	BOOL	
TC_Act	BOOL	
TC_Abt	BOOL	
TC_Err	BOOL	
TC_ErrID	WORD	



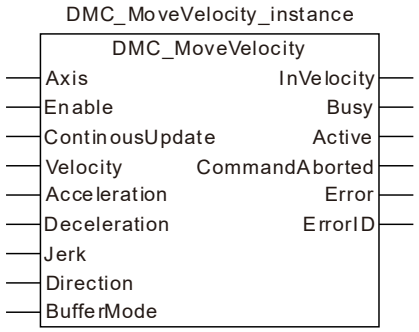
2. 运动曲线和时序图



- 当 TC_En 由 FALSE 变为 TRUE，执行 DMC_TorqueControl 指令，同时 TC_Bsy 由 FALSE 变为 TRUE，当 TC_Act 变为 TRUE 时，指令开始控制轴，扭矩值按照设定的扭矩变化率增加。
- 当 TC_Tor 的值写入 0 后，扭矩值按照设定的扭矩变化率减小到 0。
- 当 TC_En 由 TRUE 变为 FALSE 时，同时 TC_InTor、TC_Bsy、TC_Act 由 FALSE 变为 TRUE，伺服退出扭矩模式。

11.3.30 DMC_MoveVelocity（更改速度立即生效指令）

FB/FC	说明	适用机种
FB	此指令用于速度指令在执行过程中更改速度指令参数使控制轴生效。	DVP15MC11T DVP15MC11T-06

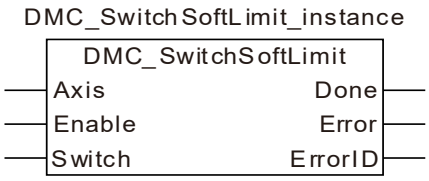


- **功能说明**

此指令用于速度指令在执行过程中更改速度轴速度立即生效。该指令功能和 MC_MoveVelocity 指令功能一样，区别在于该指令在控制轴过程中，当 Enable 为 TRUE 时，更改该指令速度、加速度、减速度、加速度变化率或者方向都会立即生效。该指令各个参数的含义和数据类型请参考 MC_MoveVelocity 指令说明。

11.3.31 DMC_SwitchSoftLimit（软件极限开关指令）

FB/FC	说明	适用机种
FB	此指令用于控制软件极限开关。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Axis （轴的站号）	设定指令欲控制的轴。	USINT	请参考第 2.2 节功能简介 (不可缺省)	*Enable* 为 TRUE 时
Enable （执行位）	当*Enable*由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE (FALSE)	*Enable* 为 TRUE 时
Switch (极限开关)	FALSE ：关闭软件极限开关 TRUE：开启软件极限开关	BOOL-	TRUE 或 FALSE (FALSE)	*Enable* 为 TRUE 时

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 *Error* 为 TRUE 时 ◆ 当 *Enable* 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时

- **功能说明**

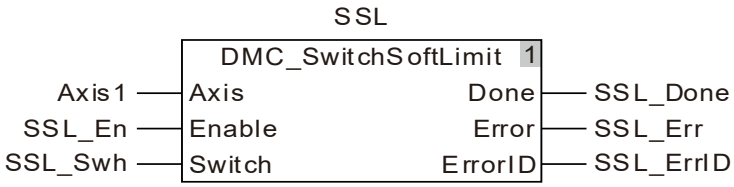
此指令用于设定软件限制开关，当引脚 Switch 为 TRUE 时，执行该指令完成后表示开启软件限制；当引脚 Switch 为 FALSE 时，执行该指令完成后表示关闭软件限制。如果不使用该指令，也可以通过软件中网络配置→Motion →轴配置 →基本设置进行设定，该指令执行时以该指令为准。软件限制的最大位置和最小位置需通过软件来设定。

当开启软件限制之后，若运动指令正在控制运动的轴位置超过软件限制范围时，运动指令会报错。

程序范例

■ **变量和程序**

变量名	数据类型	初始值
SSL	DMC_SwitchSoftLimit	
Axis1	USINT	1
SSL_En	BOOL	FALSE
SSL_Swh	BOOL	FALSE
SSL_Done	BOOL	
SSL_Err	BOOL	
SSL_ErrID	WORD	



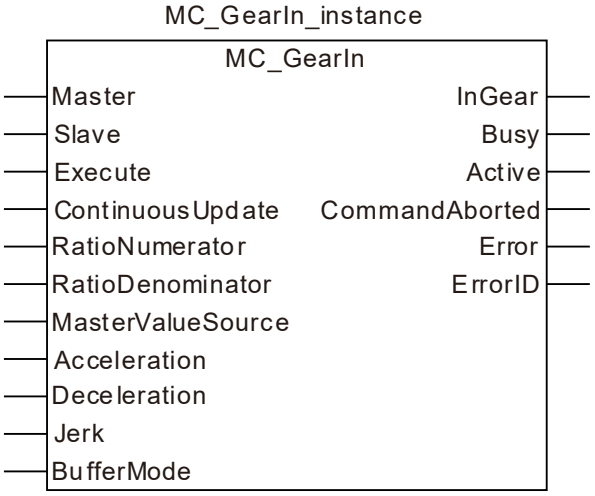
❖ 当 SSL_Swh 为 TRUE 时，使 SSL_En 由 FALSE 变为 TRUE，执行该指令，当指令完成时，SSL_Done 变为 TRUE，则开启 1 号轴软件极限。

❖ 当 SSL_Swh 为 FALSE 时，使 SSL_En 由 FALSE 变为 TRUE，执行该指令，当指令完成时，SSL_Done 变为 TRUE，则关闭 1 号轴软件极限。

11.4 多轴指令

11.4.1 MC_GearIn（电子齿轮耦合指令）

FB/FC	说明	适用机种
FB	此指令用于两个轴之间建立电子齿轮关系	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Master （轴的站号）	设定指令欲控制的主轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Slave （轴的站号）	设定指令欲控制的从轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
ContinuousUpdate	保留	-	-	-
RatioNumerator （电子齿轮分子）	电子齿轮的分子	LREAL	正数、负数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
RatioDenominator （电子齿轮分母）	电子齿轮的分母	LREAL	正数 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
MasterValueSource （位置来源选择）	从轴命令来源选择 0 :从轴跟随主轴的命令位置 1 :从轴跟随主轴的实际位置	MC_Source	0:mcSetValue 1:mcActualValue （0）	*Execute* 由FALSE 变为 TRUE
Acceleration （加速度）	设定的目标加速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Deceleration （减速度）	设定的目标减速度 （单位: 单元/秒 ^{2}）	LREAL	正数 （不可缺省）	*Execute* 由FALSE 变为 TRUE
Jerk （加速度的变化率）	设定的目标加速度或减速度的变化率 （单位: 单元/秒 ^{3}）	LREAL	正数 （不可缺省）	*Execute* 由FALSE 变为 TRUE
BufferMode （交接模式）	设定两个指令之间交接模式 0：打断 1：等待	MC_Buffer_Mode	0 : mcAborting 1 : mcBuffered （0）	*Execute* 由FALSE 变为 TRUE

说明：

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行。 无论该指令是否执行完成， Execute 再次由 FALSE 变为 TRUE 时，该指令可以重新执行，此时能够重新生效的引脚参数包括 Velocity、Acceleration、Deceleration、Jerk，其它引脚参数不会生效。
2. 该指令执行时，该指令的从轴可以执行其它运动指令；其它运动指令打断该指令时，主轴与从轴之间的齿轮关系会解除。可以执行 MC_Halt 或者 MC_Stop 停止从轴。
3. Acceleration、Deceleration、Jerk 的关系说明请参考 10.2 节。
4. 关于 BufferMode 的详细内容，请参考 10.3 节

- **输出参数**

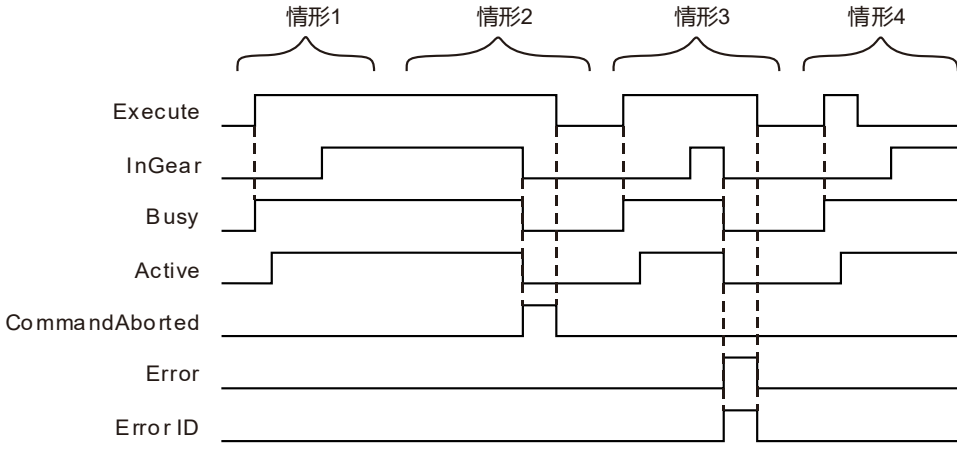
名称	功能	数据类型	输出范围
InGear （达到同步状态）	该输出参数为 TRUE 时表示从轴到达同步状态	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
InGear	◆ 当从轴到达同步状态	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当同步状态到达后改变输入参数， Execute 再次由 FALSE 变为 TRUE 时，InGear 会立即变为 FALSE；当该指令完成后不

名称	变为 TRUE 的时机	变为 FALSE 的时机
		改变输入参数，Execute 再次由 FALSE 变为 TRUE 时 InGear 会立即变为 FALSE，且下个周期 InGear 变为 TRUE。
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Command Aborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当同步状态到达时，InGear 变为 TRUE，同时 *Busy* 和 *Active* 依然保持为 TRUE。

情形2： 当 *Execute* 为 TRUE 时，从轴被其它指令控制时，该指令被其它指令中断，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 时，当有错误产生时(如参数错误)，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 InGear，*Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中，当 *Execute* 由 TRUE 变为 FALSE 后， InGear 变为 TRUE， 同时 *Busy* 和 *Active* 依然保持为 TRUE。

- **功能说明**

1. 此指令用于两个轴之间建立电子齿轮关系。该指令执行后，从轴按照电子齿轮分子、电子齿轮分母、命令来源、加速度、减速度、加速度的变化率、交接模式与主轴进行齿轮动作。主轴可以为实轴、虚轴，或者编码器轴，从轴可以为实轴或者虚轴。
2. 此指令执行时，从轴需要在使能状态下，主轴在使能或去使能状态下都可以。
3. 两个轴没有建立电子齿轮关系时，执行该指令，从轴按照该指令指定的电子齿轮分子、电子齿轮分母、加速度、减速度、加速度的变化率到达目标速度。

$$\text{从轴加（减）速度} = \text{主轴加（减）速度} \times \frac{\text{电子齿轮比分子}}{\text{电子齿轮比分母}}$$

两个轴建立电子齿轮关系后 (该指令的 InGear 为 TRUE 时)，从轴速度与电子齿轮分子、电子齿轮分母和主轴速度的关系如下所示：

$$\text{从轴目标速度} = \text{主轴速度} \times \frac{\text{电子齿轮比分子}}{\text{电子齿轮比分母}}$$

4. 电子齿轮比

$$\text{电子齿轮比} = \frac{\text{电子齿轮比分子}}{\text{电子齿轮比分母}}$$

电子齿轮比为正数时，从轴和主轴的运动方向相同。

电子齿轮比为负数时，从轴和主轴的运动方向相反。

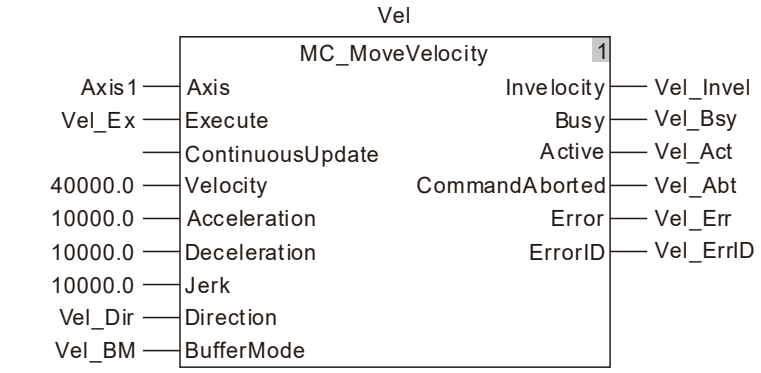
程序范例

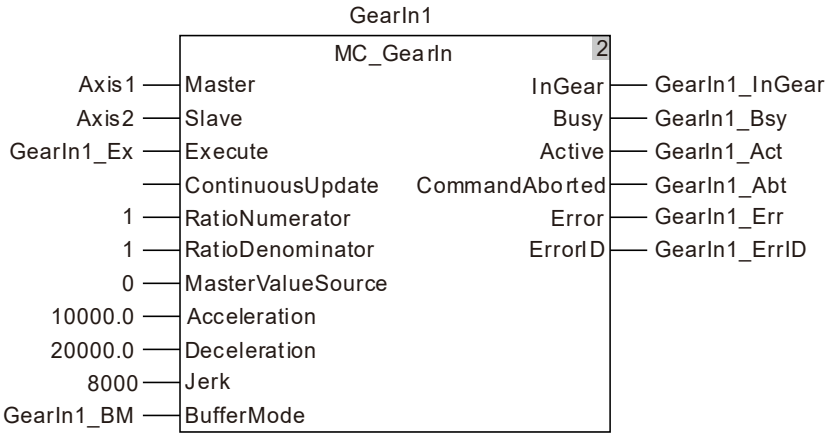
MC_GearIn 指令程序范例如下所示：

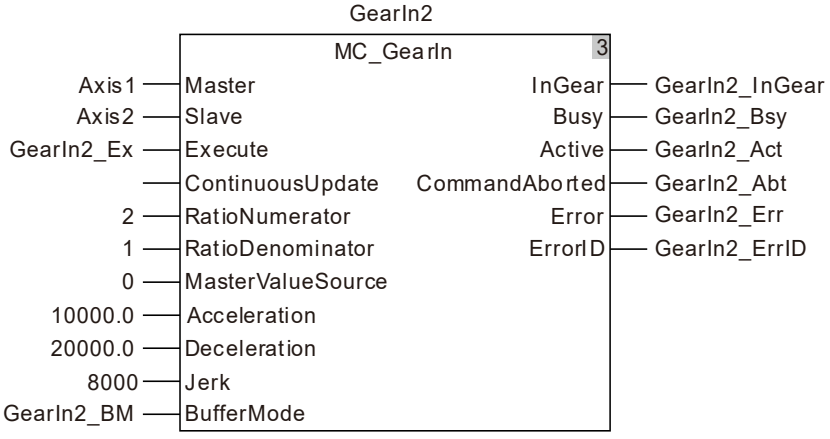
1. 变量和程序

变量名	数据类型	初始值
Vel	MC_MoveVelocity	
Axis1	USINT	1
Axis2	USINT	2
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
GearIn1	MC_GearIn	
GearIn1_Ex	BOOL	FALSE
GearIn1_BM	MC_Buffer_Mode	0
GearIn1_InGear	BOOL	
GearIn1_Bsy	BOOL	
GearIn1_Act	BOOL	
GearIn1_Abt	BOOL	
GearIn1_Err	BOOL	

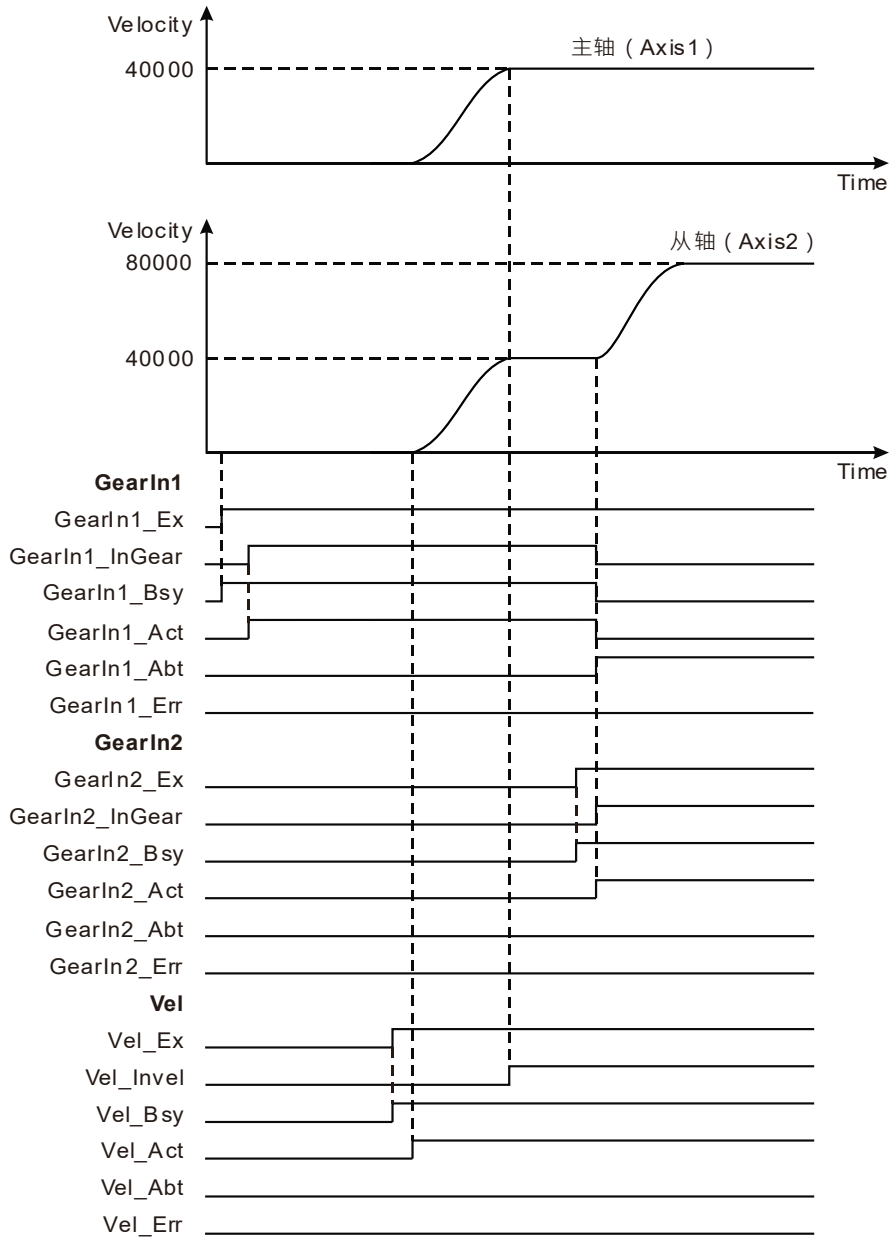
变量名	数据类型	初始值
GearIn1_ErrID	WORD	
GearIn2	MC_GearIn	
GearIn2_Ex	BOOL	FALSE
GearIn2_BM	MC_Buffer_Mode	0
GearIn2_InGear	BOOL	
GearIn2_Bsy	BOOL	
GearIn2_Act	BOOL	
GearIn2_Abt	BOOL	
GearIn2_Err	BOOL	
GearIn2_ErrID	WORD	







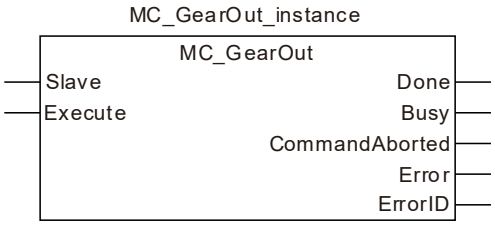
2. 运动曲线和时序图



- GearIn1 电子齿轮比分子和分母都为 1，GearIn1_Ex 由 FALSE 变为 TRUE，同时 GearIn1 _Bsy 变为 TRUE，且一个周期后，GearIn1_InGear 变为 TRUE，主轴和从轴建立齿轮关系。
- 主轴和从轴建立电子齿轮关系后， Vel_Ex 由 FALSE 变为 TRUE，且一个周期后， Vel_Act 变为 TRUE，主轴执行速度指令，从轴跟随主轴运转。
- GearIn2 电子齿轮比分子和分母分别为 2 和 1，GearIn2_Ex 由 FALSE 变为 TRUE，同时 GearIn2_Bsy 变为 TRUE，且一个周期后，GearIn2_Act 和 GearIn1_Abt 变为 TRUE，从轴根据 GearIn2 指令设定的电子齿轮比、命令来源、加速度、加速度的变化率、交接模式到达目标速度。因为 GearIn2 电子齿轮比分子和分母分别为 2 和 1，所以从轴的目标速度为主轴速度的 2 倍。GearIn2_InGear 变为 TRUE 时，从轴的速度为主轴速度的 2 倍。

11.4.2 MC_GearOut（电子齿轮脱离指令）

FB/FC	说明	适用机种
FB	此指令用于解除主从轴间已经建立的齿轮关系。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Slave （轴的站号）	设定指令预脱离的从轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-

说明：

1. 两个轴建立电子齿轮关系（MC_GearIn）后，如果从轴通过MC_GearOut指令脱离电子齿轮关系，从轴将保持脱离时的速度继续运行。
2. 该指令执行完成后，从轴可以执行其它运动指令。
3. 两个轴脱离电子齿轮关系（MC_GearOut）后，如果想停止从轴，可以使用MC_Halt或者MC_Stop指令使从轴停止。

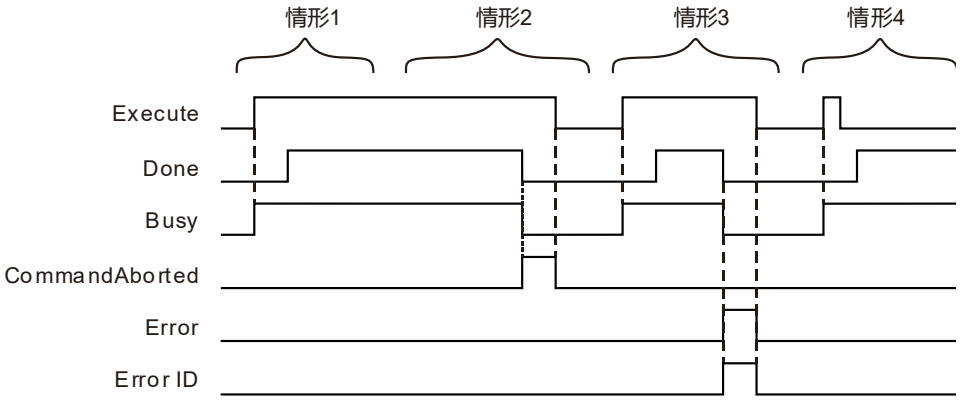
- **输出参数**

名称	功能	数据类型	输出范围
Done （电子齿轮关系解除）	该输出参数为 TRUE 时表示从轴和主轴的电子齿轮关系解除并且该指令在控制轴	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当从轴和主轴的电子齿轮关系解除时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，*Execute* 由 TRUE 变为 FALSE 后，*Busy* 和 *Done* 依然保持为 TRUE。

情形2： 当 *Execute* 为 TRUE 时，若指令被其它指令中断，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Done* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 时，当有错误产生时(如轴去使能)，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Done* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

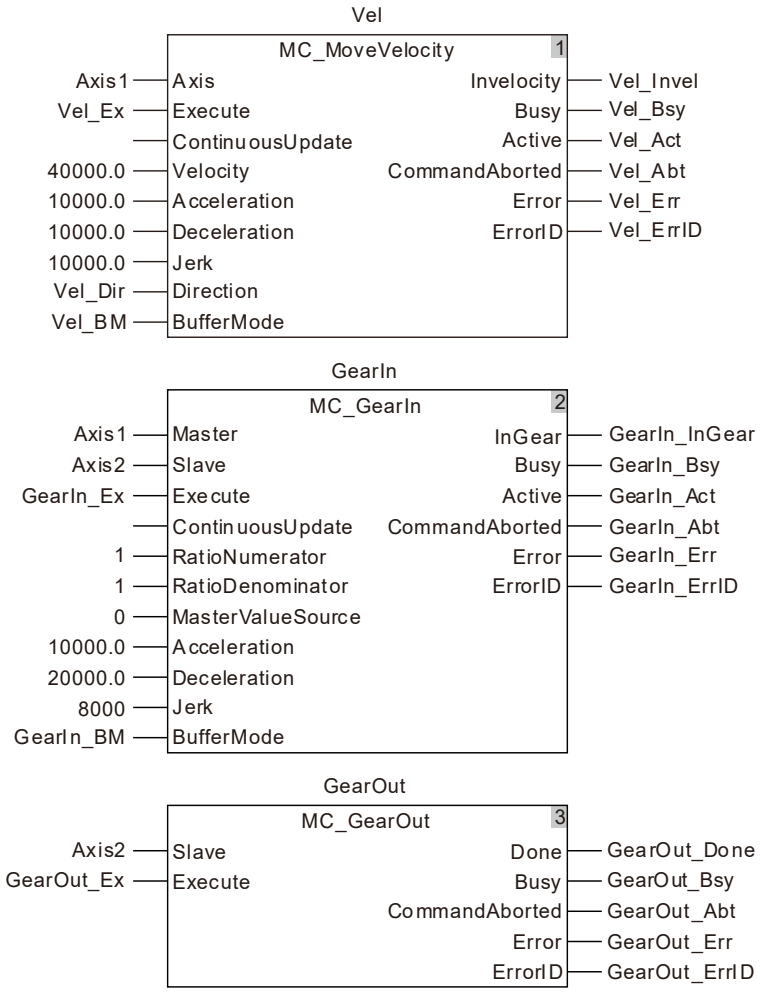
情形4： 在指令执行过程中，不到一个周期时，*Execute* 由 TRUE 变为 FALSE 后，当到达一个周期时，*Done* 变为 TRUE，且 *Busy* 仍然保持为 TRUE。

程序范例

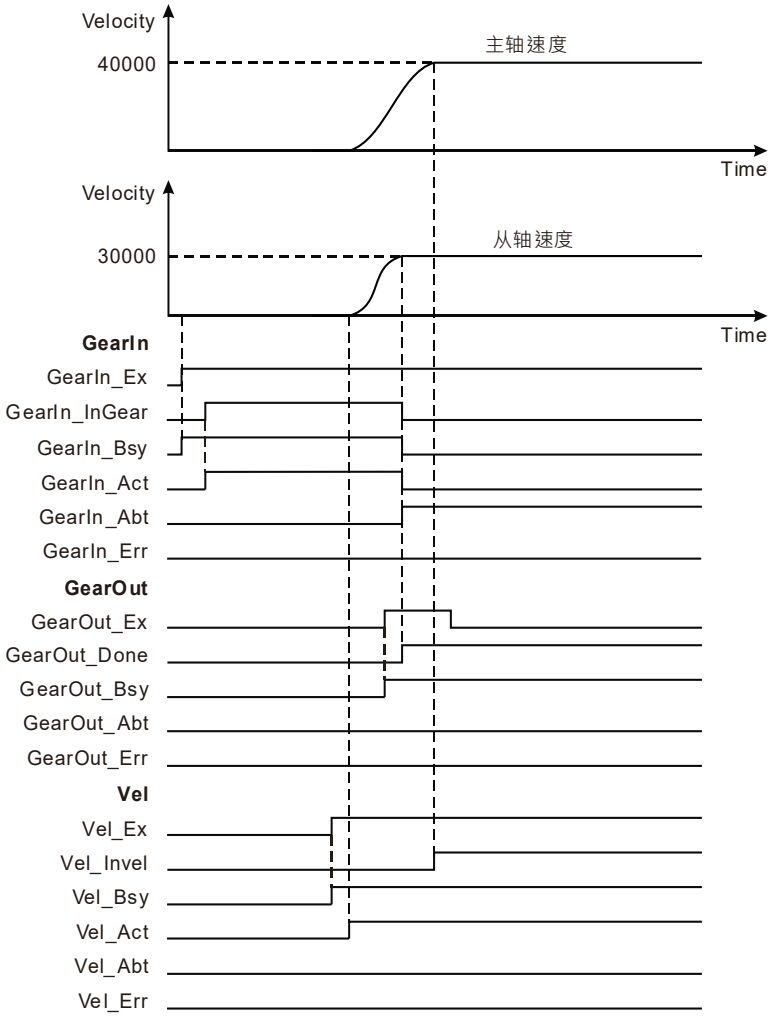
MC_GearOut 指令程序范例如下所示：

1. 变量和程序

变量名	数据类型	初始值
Vel	MC_MoveVelocity	
Axis1	USINT	1
Axis2	USINT	2
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
GearIn	MC_ GearIn	
GearIn_Ex	BOOL	FALSE
GearIn_BM	MC_Buffer_Mode	0
GearIn_InGear	BOOL	
GearIn_Bsy	BOOL	
GearIn_Act	BOOL	
GearIn_Abt	BOOL	
GearIn_Err	BOOL	
GearIn_ErrID	WORD	
GearOut	MC_ GearOut	
GearOut_Ex	BOOL	FALSE
GearOut_Done	BOOL	
GearOut_Bsy	BOOL	
GearOut_Act	BOOL	
GearOut_Abt	BOOL	
GearOut_Err	BOOL	
GearOut_ErrID	WORD	


Vel
MC_MoveVelocity
1
Axis1
Axis
Vel_Ex
Execute
ContinuousUpdate
40000.0
Velocity
10000.0
Acceleration
10000.0
Deceleration
10000.0
Jerk
Vel_Dir
Direction
Vel_BM
BufferMode
InVelocity
Vel_InVel
Busy
Vel_Bsy
Active
Vel_Act
CommandAborted
Vel_Abt
Error
Vel_Err
ErrorID
Vel_ErrID
GearIn
MC_GearIn
2
Axis1
Master
Axis2
Slave
GearIn_Ex
Execute
ContinuousUpdate
1
RatioNumerator
1
RatioDenominator
0
MasterValueSource
10000.0
Acceleration
20000.0
Deceleration
8000
Jerk
GearIn_BM
BufferMode
InGear
GearIn_InGear
Busy
GearIn_Bsy
Active
GearIn_Act
CommandAborted
GearIn_Abt
Error
GearIn_Err
ErrorID
GearIn_ErrID
GearOut
MC_GearOut
3
Axis2
Slave
GearOut_Ex
Execute
Done
GearOut_Done
Busy
GearOut_Bsy
CommandAborted
GearOut_Abt
Error
GearOut_Err
ErrorID
GearOut_ErrID

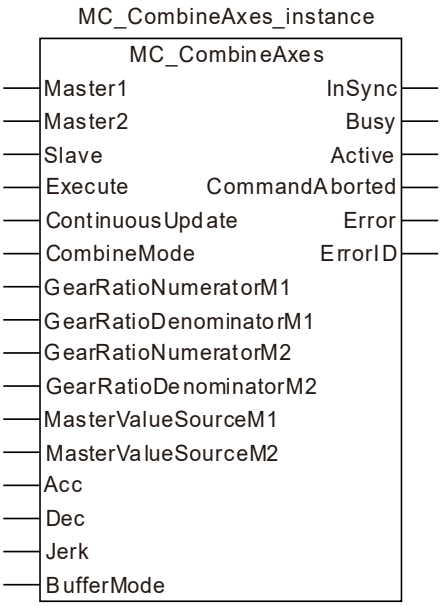
2. 运动曲线和时序图



- ❖ GearIn_Ex 由 FALSE 变为 TRUE，同时 GearIn_Bsy 变为 TRUE，且一个周期后，GearIn_InGear 变为 TRUE，主轴和从轴建立齿轮关系。
- ❖ 主轴和从轴建立电子齿轮关系后， Vel_Ex 由 FALSE 变为 TRUE，且一个周期后， Vel_Act 变为 TRUE，主轴执行速度指令，从轴跟随主轴运转。
- ❖ 主轴执行速度指令时，GearOut_Ex 由 FALSE 变为 TRUE，同时 GearOut_Bsy 变为 TRUE，且一个周期后，GearOut_Done 和 GearIn_Abt 变为 TRUE，从轴以当前的速度继续运转。

11.4.3 MC_CombineAxes（双主轴联合齿轮指令）

FB/FC	说明	适用机种
FB	将两个主轴的位置相加或相减的值作为从轴位置输出	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Master1 （1号主轴）	进行控制时，第一个轴的位置来源。	USINT	请参考第2.2节**功能简介** （不可缺省）	*Execute*由FALSE变为TRUE
Master2 （2号主轴）	进行控制时，第二个轴的位置来源。	USINT	请参考第2.2节**功能简介** （不可缺省）	*Execute*由FALSE变为TRUE
Slave （从轴）	被控制的轴。	USINT	请参考第2.2节**功能简介** （不可缺省）	*Execute*由FALSE变为TRUE
Execute （执行位）	当 *Execute* 由FALSE变为TRUE时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
ContinuousUpdate	保留	-	-	-
CombineMode （合成模式）	选择合成模式。 0：两主轴各自变化位置相加 1：两主轴各自变化位置相减	MC_Combine_Mode	0: mcAddAxes、1: mcSubAxes （0）	*Execute*由FALSE变为TRUE
GearRatioNumeratorM1 （1号主轴齿轮比分子）	设定1号主轴齿轮比分子。	LREAL	正数或负数 （不可缺省）	*Execute*由FALSE变为TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
GearRatioDenominatorM1 （1号主轴齿轮比分母）	设定1号主轴齿轮比分母。	LREAL	正数或负数 （不可缺省）	*Execute*由FALSE变为TRUE
GearRatioNumeratorM2 （2 号主轴齿轮比分子）	设定2号主轴齿轮比分子。	LREAL	正数或负数 （不可缺省）	*Execute*由FALSE变为TRUE
GearRatioDenominatorM2 （2号主轴齿轮比分母）	设定2号主轴齿轮比分母。	LREAL	正数或负数 （不可缺省）	*Execute*由FALSE变为TRUE
MasterValueSourceM1 （1号主轴的同步源）	设定1号主轴的同步源。 0：命令位置 1：实际位置	MC_SOURCE	0:mcSetValue 1:mcActualValue （0）	*Execute*由FALSE变为TRUE
MasterValueSourceM2 （2号主轴的同步源）	设定2号主轴的同步源。 0：命令位置 1：实际位置	MC_SOURCE	0:mcSetValue 1:mcActualValue （0）	*Execute*由FALSE变为TRUE
Acc （加速度）	设定从轴的加速度。	LREAL	正数 （不可缺省）	*Execute*由FALSE变为TRUE
Dec （减速度）	设定从轴的减速度。	LREAL	正数 （不可缺省）	*Execute*由FALSE变为TRUE
Jerk （加速度的变化率）	设定从轴的加速度的变化率。	LREAL	正数 （不可缺省）	*Execute*由FALSE变为TRUE
BufferMode （交接模式）	设定两个指令之间交接模式。 0：打断 1：等待	MC_Buffer_Mode	0 : mcAborting 1 : mcBuffered （0）	*Execute*由FALSE变为TRUE

说明：

1. 该指令在 *Execute* 由 FALSE 变为 TRUE 时开始执行。无论该指令是否执行完成， *Execute* 再次由 FALSE 变为 TRUE 时，该指令无法重新执行， 将维持上一次的设定值。
2. *Position*、*Velocity*、*Acceleration*、*Jerk* 的关系说明请参考第 10.2 节。
3. 关于 *BufferMode* 的详细内容，请参考第 10.3 节

- **输出参数**

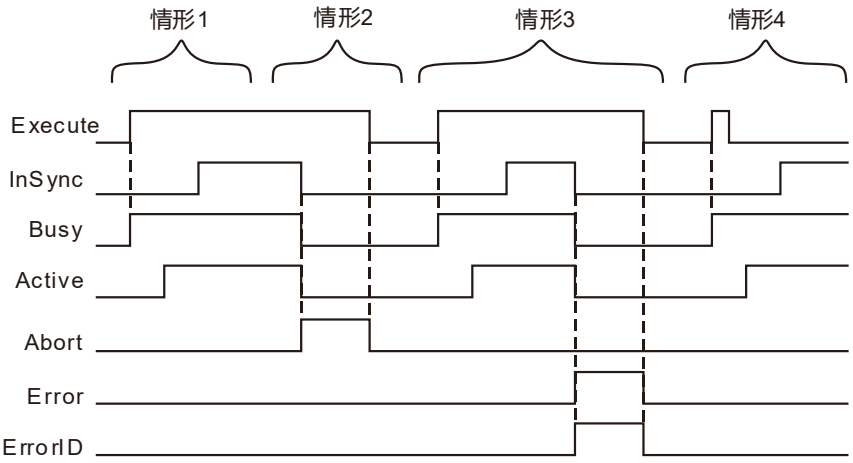
名称	功能	数据类型	输出范围
InSync （已经同步）	该输出参数为 TRUE 时表示从轴已经完成同步动作	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
(控制中)			
CommandAborted (中断)	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error (错误)	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID (错误代码)	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
InSync	◆ 当从轴完成同步动作	◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE。当从轴已经和两个主轴同步，*InSync* 变为 TRUE，同时 *Busy* 和 *Active* 依然保持为 TRUE。

情形2： 当 Execute 为 TRUE 时，Busy 为 TRUE，Active 为 TRUE，当从轴已经和两个主轴同步，InSync 为 TRUE，此时另一个指令打断本指令，CommandAborted 变为 TRUE，同时 Invelocity、Busy 和 Active 变为 FALSE，当 Execute 由 TRUE 变为 FALSE 时，CommandAborted 变为 FALSE。

情形3： 当 Execute 由 FALSE 变为 TRUE 时，当有错误产生时(如轴报警或者掉线时) Error 变为 TRUE，ErrorID 显示对应的错误码，同时 InSync、Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

情形4： 在执行的过程中，当 Execute 由 TRUE 变为 FALSE 后，指令仍然在执行，Busy 和 Active 不会改变状态。当从轴已经和两个主轴同步，InSync 变为 TRUE，同时 Busy 和 Active 依然保持为 TRUE。

- **功能说明**

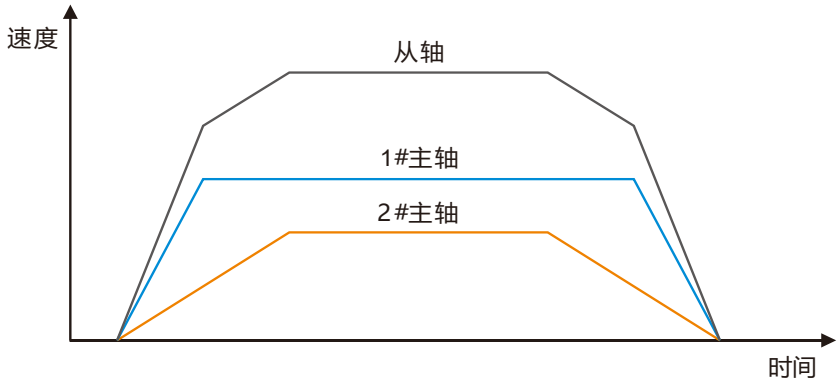
此指令用于将两个主轴的位置相加或相减的值作为从轴位置输出。

- **此指令合成方式分为两种：相加或相减**

一号主轴的位置变化量和二号主轴的位置变化量进行相加或者相减，将计算出的值作为从轴的位置变化量输出。

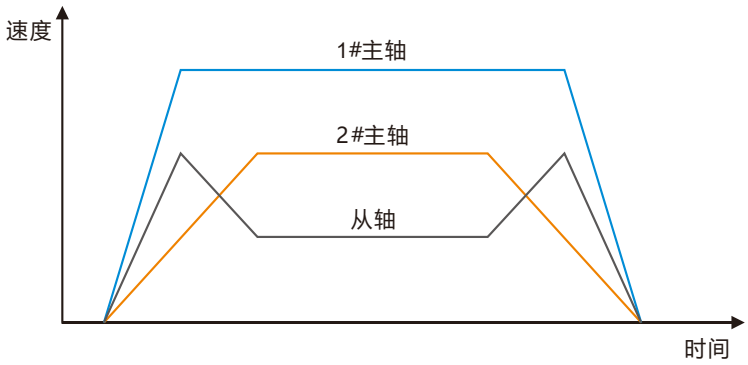
- **当 CombineMode 值为 0 时**

$$从轴位置变化量 = 一号主轴位置变化量 \times \frac{一号主轴齿轮比分子}{一号主轴齿轮比分母} + 二号主轴位置变化量 \times \frac{二号主轴齿轮比分子}{二号主轴齿轮比分母}。$$



- **当 CombineMode 值为 1 时**

$$从轴位置变化量 = 一号主轴位置变化量 \times \frac{一号主轴齿轮比分子}{一号主轴齿轮比分母} - 二号主轴位置变化量 \times \frac{二号主轴齿轮比分子}{二号主轴齿轮比分母}。$$



- 主轴齿轮比分子和分母设定是调节两个主轴位置变化量的因子，使用公式见上。

- 主轴的同步源可以设置为 0：命令位置；1：实际位置，用以确认位置变化量的来源。设置为 0 时是主轴命令位置的变化量进行相加或相减，设置为 1 时是主轴实际位置的变化量进行相加或相减。
- 加速度、减速度和加速度的变化率表示在执行本指令之前，主轴已经在运动状态，此时若执行本指令，从轴会根据加速度、减速度和加速度的变化率进行加速或者减速，以达到和主轴位置变化同步。同步后 InSync 为 TRUE，指令执行完成。
- 要结束本指令的主从轴关系，请使用其它运动指令控制从轴（如 MC_Stop）,BufferMode 输入引脚填 0，以打断本指令，解除主从轴关系。
- 如果要在运动过程中切换主轴齿轮比，请使用另一个 MC_CombineAxes 指令打断正在执行的 MC_CombineAxes 指令即可。

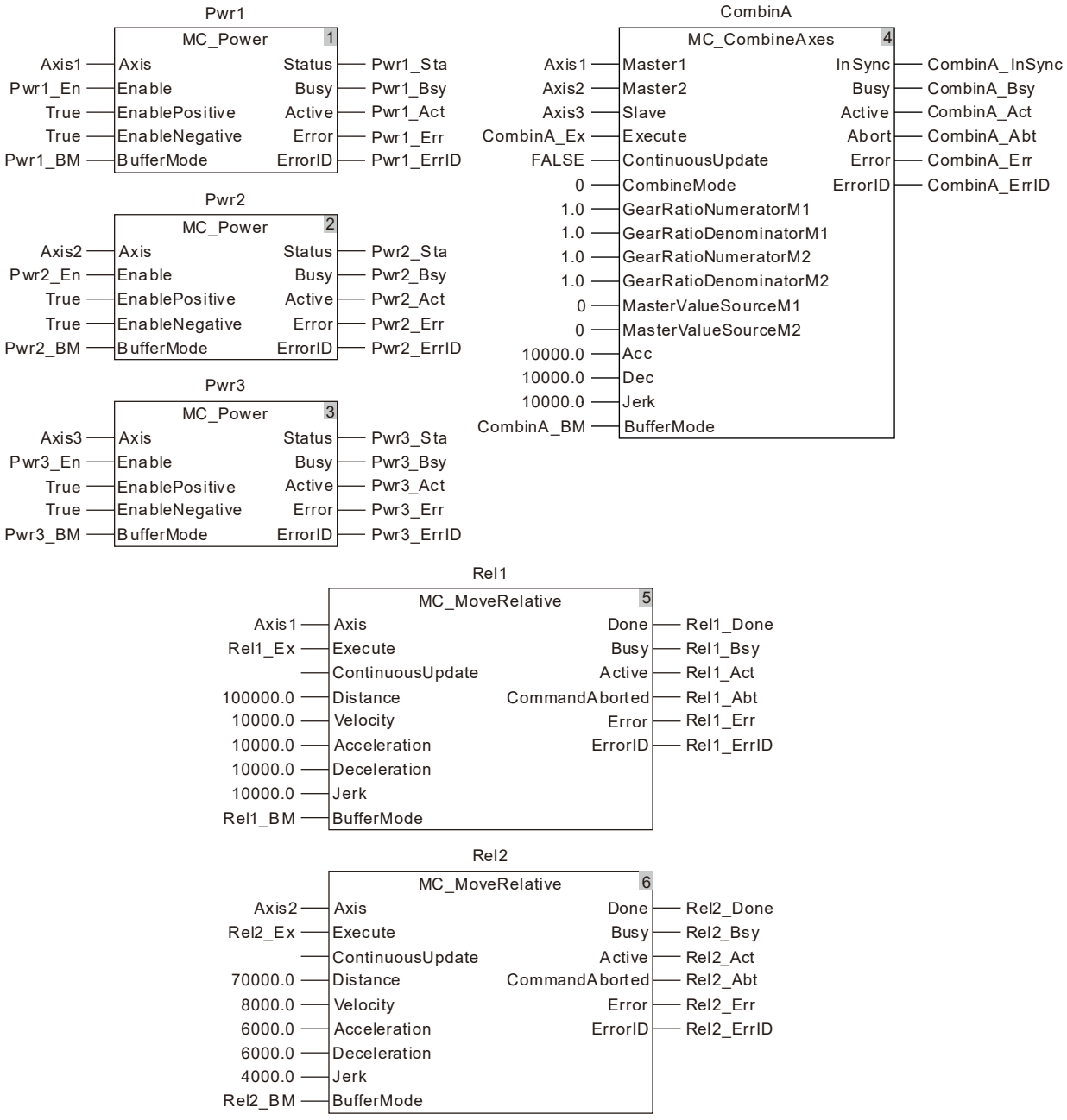
程序范例

MC_CombineAxes 指令执行范例如下：

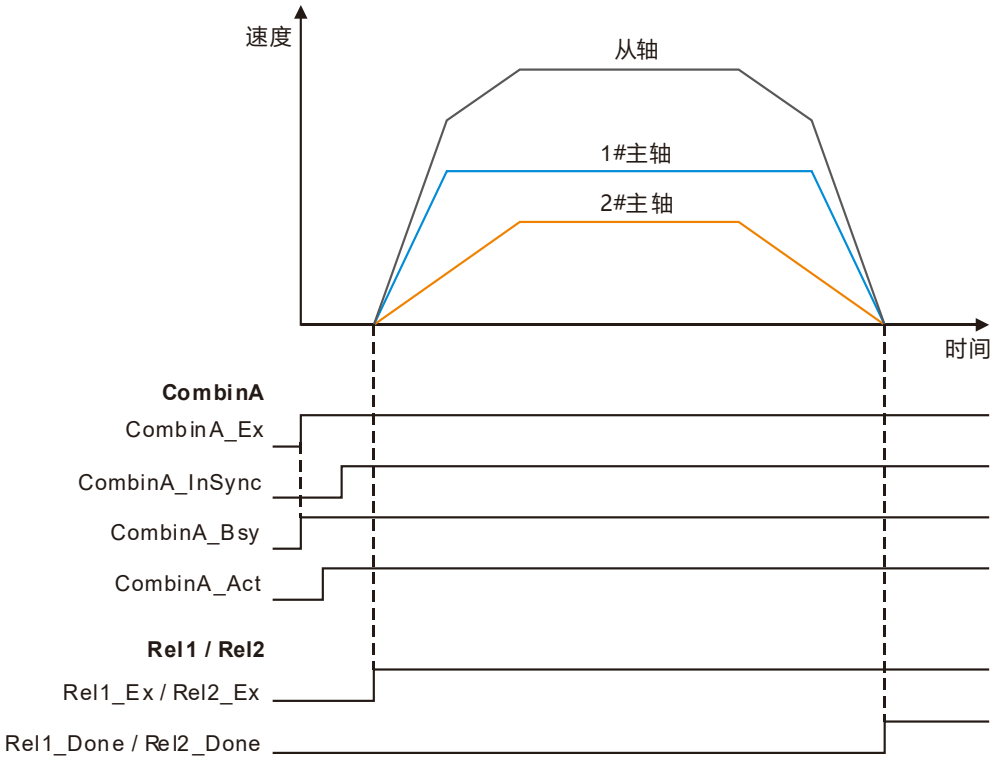
1. 变量和程序

变量名	数据类型	初始值
Pwr1	MC_Power	
Axis1	USINT	1
Pwr1_BM	MC_Buffer_Mode	1
Pwr1_Sta	BOOL	
Pwr1_Bsy	BOOL	
Pwr1_Act	BOOL	
Pwr1_Err	BOOL	
Pwr1_ErrID	WORD	
Pwr2	MC_Power	
Axis2	USINT	1
Pwr2_BM	MC_Buffer_Mode	1
Pwr2_Sta	BOOL	
Pwr2_Bsy	BOOL	
Pwr2_Act	BOOL	
Pwr2_Err	BOOL	
Pwr2_ErrID	WORD	
Pwr3	MC_Power	
Axis3	USINT	1
Pwr3_BM	MC_Buffer_Mode	1
Pwr3_Sta	BOOL	
Pwr3_Bsy	BOOL	
Pwr3_Act	BOOL	
Pwr3_Err	BOOL	
Pwr3_ErrID	WORD	
CombinA	MC_CombineAxes	
CombinA_Ex	BOOL	FALSE
CombinA_BM	MC_Buffer_Mode	1
CombinA_InSync	BOOL	
CombinA_Bsy	BOOL	
CombinA_Act	BOOL	

变量名	数据类型	初始值
CombinA_Abt	BOOL	
CombinA_Err	BOOL	
CombinA_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	FALSE
Rel1_Dir	MC_DIRECTION	1
Rel1_BM	MC_Buffer_Mode	0
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	FALSE
Rel2_Dir	MC_DIRECTION	1
Rel2_BM	MC_Buffer_Mode	0
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	WORD	


Pwr1
MC_Power
1
Axis1
Axis
Status
Pwr1_Sta
Pwr1_En
Enable
Busy
Pwr1_Bsy
True
EnablePositive
Active
Pwr1_Act
True
EnableNegative
Error
Pwr1_Err
Pwr1_BM
BufferMode
ErrorID
Pwr1_ErrID
Pwr2
MC_Power
2
Axis2
Axis
Status
Pwr2_Sta
Pwr2_En
Enable
Busy
Pwr2_Bsy
True
EnablePositive
Active
Pwr2_Act
True
EnableNegative
Error
Pwr2_Err
Pwr2_BM
BufferMode
ErrorID
Pwr2_ErrID
Pwr3
MC_Power
3
Axis3
Axis
Status
Pwr3_Sta
Pwr3_En
Enable
Busy
Pwr3_Bsy
True
EnablePositive
Active
Pwr3_Act
True
EnableNegative
Error
Pwr3_Err
Pwr3_BM
BufferMode
ErrorID
Pwr3_ErrID
CombinA
MC_CombineAxes
4
Axis1
Master1
InSync
CombinA_InSync
Axis2
Master2
Busy
CombinA_Bsy
Axis3
Slave
Active
CombinA_Act
CombinA_Ex
Execute
Abort
CombinA_Abt
FALSE
ContinuousUpdate
Error
CombinA_Err
0
CombineMode
ErrorID
CombinA_ErrID
1.0
GearRatioNumeratorM1
1.0
GearRatioDenominatorM1
1.0
GearRatioNumeratorM2
1.0
GearRatioDenominatorM2
0
MasterValueSourceM1
0
MasterValueSourceM2
10000.0
Acc
10000.0
Dec
10000.0
Jerk
CombinA_BM
BufferMode
Rel1
MC_MoveRelative
5
Axis1
Axis
Done
Rel1_Done
Rel1_Ex
Execute
Busy
Rel1_Bsy
ContinuousUpdate
Active
Rel1_Act
100000.0
Distance
CommandAborted
Rel1_Abt
10000.0
Velocity
Error
Rel1_Err
10000.0
Acceleration
ErrorID
Rel1_ErrID
10000.0
Deceleration
10000.0
Jerk
Rel1_BM
BufferMode
Rel2
MC_MoveRelative
6
Axis2
Axis
Done
Rel2_Done
Rel2_Ex
Execute
Busy
Rel2_Bsy
ContinuousUpdate
Active
Rel2_Act
70000.0
Distance
CommandAborted
Rel2_Abt
8000.0
Velocity
Error
Rel2_Err
6000.0
Acceleration
ErrorID
Rel2_ErrID
6000.0
Deceleration
4000.0
Jerk
Rel2_BM
BufferMode

2. 运动曲线及时序图



❖ 当 CombinA_Ex 由 FALSE 变为 TRUE，MC_CombineAxes 指令开始执行，一段时间后，指令执行成功，CombinA_InSync 变为 TRUE，三个轴按照指令要求达到同步运动状态。此时将两个主轴的 MC_MoveRelative 指令的 Excute 变为 TRUE，两个主轴开始运动，此时从轴也开始根据两个主轴位置的变化量的和进行运动，单位时间内从轴的位置变化量为两主轴位置变化量的和。当主轴指令执行完成后，三个轴依然保持同步状态，若要打断三个轴的同步状态，请使用 MC_Stop 指令打断从轴，即可解除同步状态。

11.4.4 电子凸轮简介

凸轮是一个具有曲线轮廓或凹槽的构件，它把运动特性传递给紧靠其边缘移动的推杆，推杆又带动机架做周期性运动。凸轮机构一般是由凸轮，从动件和机架三部分组成。如图 11.4.4.1 所示，由 A、B、C、D 四个点组成的曲线为凸轮的轮廓曲线，AB′ 为从动件（推杆），推杆与机架相连，δ4 为内休止角，δ2 为外休止角，基圆的半径为 r0，S 为凸轮曲线。

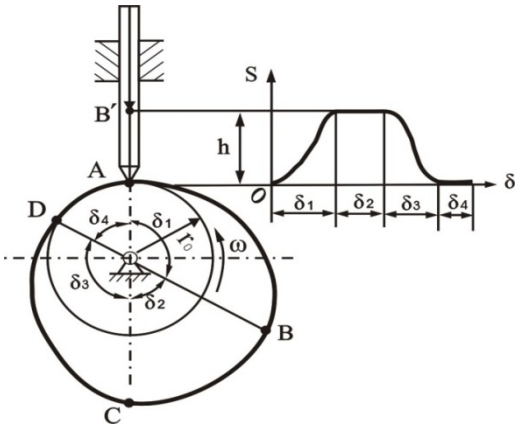


图 11.4.4.1

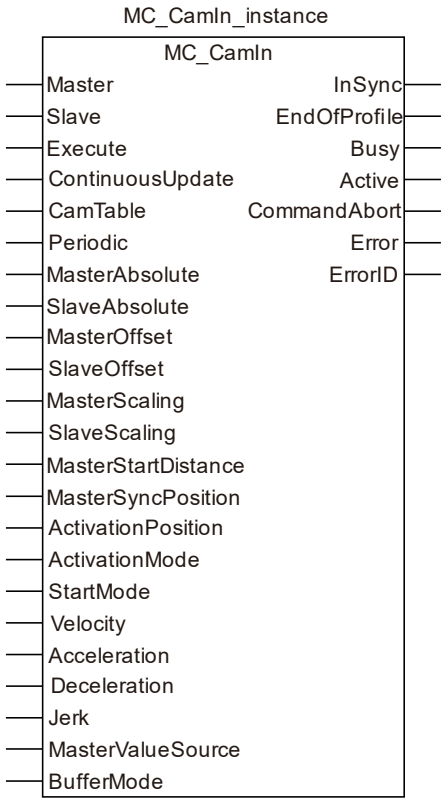
电子凸轮是通过计算机技术来模拟机械凸轮的一种方式，电子凸轮解决了机械凸轮的很多缺点，规划修改简单，无需耗费额外成本，控制效率及精度更高。由于电子凸轮是虚拟的凸轮机构，这就避免了某些凸轮机构易磨损、不适合高速传动等缺点。

用户可以在 DVP-15MC 系列运动控制器对应的编程软件中规划凸轮曲线。

凸轮曲线编辑完成后，需要在运动控制程序中调用它。运动控制程序可以通过 MC_CamIn 指令来调用凸轮曲线。

11.4.5 MC_CamIn（电子凸轮关联指令）

FB/FC	说明	适用机种
FB	此指令用于两个轴按设定参数建立凸轮关系。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Master （主轴的站号）	设定电子凸轮的主轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Slave （从轴的站号）	设定电子凸轮的从轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
ContinuousUpdate （保留）	保留	-	-	-
CamTable （电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64 （不可缺省）	Execute 由 FALSE 变为 TRUE
Periodic （周期运行）	设定电子凸轮为周期运行或仅运行一个周期。	BOOL	TRUE 或 FALSE （FALSE）	Execute 由FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
MasterAbsolute （主轴绝对）	设定主轴的位置模式：为TRUE时，主轴位置为绝对模式；为FALSE时，主轴位置为相对模式。	BOOL	TRUE 或 FALSE （FALSE）	Execute 由FALSE变为 TRUE
SlaveAbsolute （从轴绝对）	设定从轴的位置模式：为TRUE时，从轴位置为绝对模式；为FALSE时，从轴位置为相对模式。	BOOL	TRUE 或 FALSE （FALSE）	Execute 由FALSE变为 TRUE
MasterOffset （主轴位置偏移量）	设定主轴位置偏移量。 （单位: 单元）	LREAL	负数、正数、0 （0）	Execute 由FALSE变为 TRUE
SlaveOffset （从轴位置偏移量）	设定从轴位置偏移量。 （单位: 单元）	LREAL	负数、正数、0 （0）	Execute 由FALSE变为 TRUE
MasterScaling （主轴位置缩放比）	设定主轴位置的缩放比例。	LREAL	正数 （不可缺省）	Execute 由FALSE变为 TRUE
SlaveScaling （从轴位置缩放比）	设定从轴位置的缩放比例。	LREAL	正数或负数 （不可缺省）	Execute 由FALSE变为 TRUE
MasterStartDistance	保留	-	-	-
MasterSyncPosition	保留	-	-	-
ActivationPosition （啮合启动位置）	设定啮合过程开始时的主轴位置，即当主轴经过该位置时，从轴开始执行啮合动作。	LREAL	负数、正数、0 （0）	Execute 由FALSE变为 TRUE
ActivationMode （启动位置的模式）	设定啮合启动位置的模式。	MC_ACTIVATION_MODE	0: mcRelative （相对轴位置） 1: mcAbsolute （绝对轴位置） 2: mcPhaseAxis （绝对轴相位） 3: mcPhaseCAM （绝对凸轮相位） （0）	Execute 由FALSE变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
StartMode （啮合模式）	设定从轴执行啮合动作的方式。	MC_START_MODE	0: mcRampInShortest（最短距离） 1: mcRampInPositive（正向） -1:mc RampInNegative（反向） （0）	Execute 由FALSE 变为 TRUE
Velocity （速度）	设定从轴执行啮合动作过程中允许的最大叠加速度 （单位: 单元/秒）	LREAL	正数 （不可缺省）	Execute 由FALSE 变为 TRUE
Acceleration （加速度）	设定从轴执行啮合动作过程中允许的最大叠加加速度 （单位: 单元/秒^2）	LREAL	正数 （不可缺省）	Execute 由FALSE 变为 TRUE
Deceleration （减速度）	设定从轴执行啮合动作过程中允许的最大叠加减速度 （单位: 单元/秒^2）	LREAL	正数 （不可缺省）	Execute 由FALSE 变为 TRUE
Jerk （加速度的变化率）	保留	-	-	-
MasterValueSource （主轴位置来源）	设定电子凸轮计算过程中主轴位置的类型。	MC_SOURCE	0：mcSetValue 1：mcActualValue （0）	Execute 由FALSE 变为 TRUE
BufferMode （交接模式）	设定两个指令之间的交接模式	MC_Buffer_Mode	0：mcAborting（打断） 1：mcBuffered（等待） （0）	Execute 由FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行。该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 当指令正在执行中，*Execute* 再次由 FALSE 变为 TRUE 时，对指令的执行不会产生影响，指令将继续按照之前的方式执行。
3. 关于 *BufferMode* 的详细内容，请参考第 10.3 节。

- **输出参数**

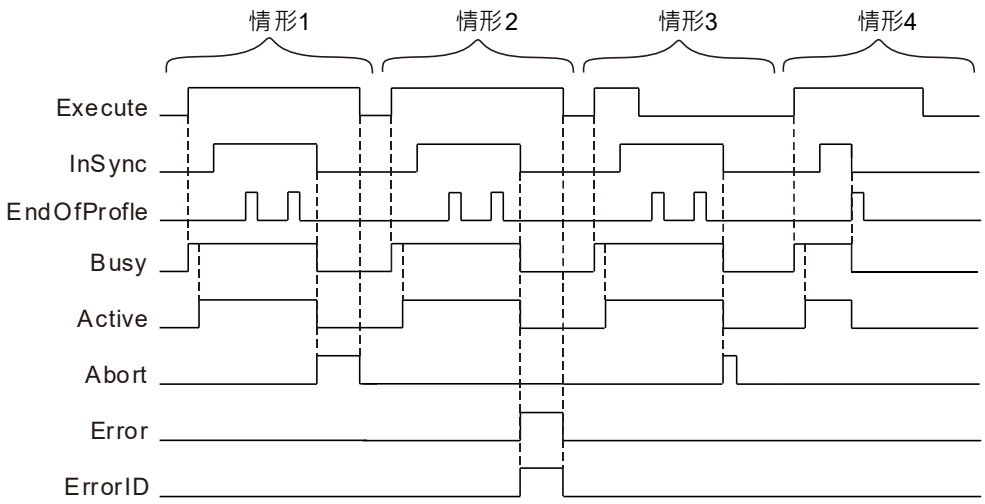
名称	功能	数据类型	输出范围
InSync （同步中）	该输出参数为 TRUE 时表示主从轴按凸轮同步运行中	BOOL	TRUE / FALSE
EndOfProfile （凸轮终点执行标志）	该输出参数为 TRUE 时表示凸轮终点被执行	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
InSync	◆ 当从轴与主轴实现凸轮同步时.	◆ 当从轴与主轴的凸轮关系解除时 ◆ 当凸轮为非周期执行（*Periodic*=FALSE），且 *EndOfProfile* 变为 TRUE 时 ◆ 当 *Error* 变为 TRUE 时 ◆ 当 *CommandAborted* 变为 TRUE 时
EndOfProfile	◆ 当执行到凸轮表中的终点时	◆ *EndOfProfile* 变为 TRUE 的后一周期变为 FALSE
Busy	◆ 当 *Execute* 变为 TRUE 时	◆ 当凸轮为非周期执行（*Periodic*=FALSE），且 *EndOfProfile* 变为 TRUE 时 ◆ 当 *Error* 变为 TRUE 时 ◆ 当 *CommandAborted* 变为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当凸轮为非周期执行（*Periodic*=FALSE），且 *EndOfProfile* 变为 TRUE 时 ◆ 当 *Error* 变为 TRUE 时 ◆ 当 *CommandAborted* 变为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 *Execute* 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，*CommandAborted* 被设置为 TRUE，且一个周期后，*CommandAborted* 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Execute* 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE；当从轴与主轴实现同步时，*InSync* 由 FALSE 变为 TRUE；当执行到凸轮周期的终点时，*EndOfProfile* 由 FALSE 变为 TRUE，且在一个周期后变为 FALSE；当从轴与主轴的凸轮关系解除时(如执行 MC_CamOut 指令)，*CommandAborted* 由 FALSE 变为 TRUE，*InSync*、*Busy*、*Active* 均由 TRUE 变为 FALSE，此后 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 由 TRUE 变为 FALSE。

情形2： 当指令执行过程中发生错误时，*Error* 由 FALSE 变为 TRUE，*ErrorID* 为对应错误码，*InSync*、*Busy*、*Active* 均有 TRUE 变为 FALSE，此后 *Execute* 由 TRUE 变为 FALSE 时，*Error* 由 TRUE 变为 FALSE，*ErrorID* 的值变为 0。

情形3： 在指令执行过程中，*Execute* 由 TRUE 变为 FALSE 后，指令仍正常执行，*InSync*、*EndOfProfile*、*Busy*、*Active* 的变化时序与 *Execute* 为 TRUE 时一致；在此之后，若从轴与主轴的凸轮关系解除，*InSync*、*Busy*、*Active* 均由 TRUE 变为 FALSE，同时 CommandAborted 由 FALSE 变为 TRUE，且在一个周期后变为 FALSE。

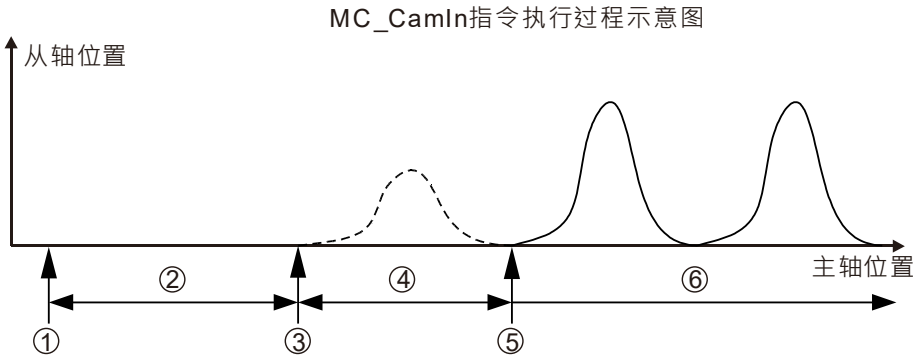
情形4： 若选择非周期执行凸轮（*Periodic*=FASLE），当执行到凸轮周期的终点时，*EndOfProfile* 由 FALSE 变为 TRUE，同时 *InSync*、*Busy*、*Active* 均由 TRUE 变为 FALSE，且一个周期后，*EndOfProfile* 由 TRUE 变为 FALSE。

- **功能说明**

MC_CamIn 指令用于控制从轴按照预先规划的凸轮关系与主轴实现凸轮同步运动；*MC_CamOut* 指令则用于解除凸轮关系。

- **MC_CamIn 指令概述**
  - ***MC_CamIn* 指令的执行流程**

MC_CamIn 指令的执行流程如下图所示：



阶段①：触发 MC_CamIn 指令执行

阶段②：等待啮合开始

阶段③：主轴到达啮合开始位置，从轴开始执行啮合动作

阶段④：啮合过程中

阶段⑤：啮合完成，主从轴实现同步

阶段⑥：主从轴同步运动中

阶段 ①：触发 *MC_CamIn* 指令执行

MC_CamIn 指令在此刻开始执行，从轴将立刻进入等待啮合开始状态。

注意：若 *ActivationPosition* 为 0 且 *ActivationMode* 为 0（相对轴位置）时，从轴将以当前速度朝同步速度运动，除此之外，从轴将立即停止运动！*MC_CamIn* 指令的所用输入参数将在此刻读入指令并锁定，供指令在执行过程中使用。

阶段 ②：等待啮合开始

从轴在静止状态下，等待开始执行啮合动作时机的到来，即等待主轴经过参数 *ActivationPosition* 所指定的位置。从轴等待的时间在不同情形下会有不同，如果 *MC_CamIn* 指令开始执行时，主轴即处在参数 *ActivationPosition* 所指定的位置，则从轴立即开始执行啮合动作；如果主轴永远没有机会到达参数 *ActivationPosition* 所指定的位置，则从轴将永远无法开始执行啮合，永远无法实现凸轮同步。参数 *ActivationPosition*、*ActivationMode* 作用于此阶段。

阶段 ③：主轴到达啮合开始位置，从轴开始执行啮合动作

当主轴经过参数 *ActivationPosition* 所指定的位置时，从轴开始执行啮合动作。参数 *MasterAbsolute*、*SlaveAbsolute*、*MasterOffset*、*SlaveOffset*、*MasterScaling*、*SlaveScaling* 将在此刻开始作用，用于确定主从轴的轴位置与其凸轮相位之间的对应关系。

阶段 ④：啮合过程中

从轴按参数 *StartMode* 所指定的方式执行啮合动作。除参数 *StartMode* 外，参数 *Velocity*、*Acceleration*、*Deceleration* 也作用于此阶段，它们将决定啮合过程中，从轴速度、加/减速度这几项运动特性。

阶段 ⑤：啮合完成，主从轴实现同步

在从轴开始执行啮合动作后，如果主从轴对应的凸轮相位满足规划的凸轮关系，则啮合完成，从轴与主轴实现凸轮同步。

说明：上图仅表示啮合开始时的主轴位置大于 *MC_CamIn* 指令开始执行时刻的主轴位置的情形，等于以及小于的情形可同理推导。

■ **ActivationPosition**

参数 *ActivationPosition* 为凸轮啮合的起始位置（该位置为主轴的“位置”），即在 *MC_CamIn* 指令正确触发执行且主轴到达 *ActivationPosition* 时，从轴开始执行啮合动作。

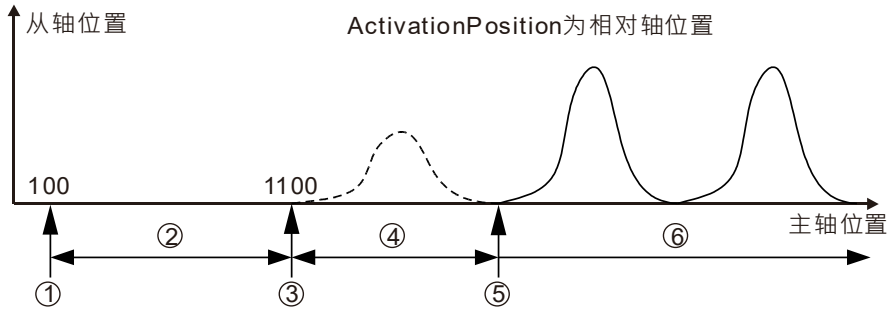
ActivationPosition 可以为：主轴位置、主轴相位、主轴凸轮相位，通过参数 *ActivationMode* 进行选择

➢ ***ActivationPosition* 为相对轴位置**

当参数 *ActivationMode*=0 时，*ActivationPosition* 为轴位置，且与 *MC_CamIn* 指令开始执行时刻的主轴位置为相对关系，即实际啮合开始时的主轴位置为 *MC_CamIn* 指令开始执行时刻的主轴位置加上 *ActivationPosition*。

例如：*MC_CamIn* 指令开始执行时刻的主轴位置为 100，*ActivationPosition* 为 1000，则实际啮合开始时的主轴位置为 1100（1100=100+1000）。

MC_CamIn指令执行过程示意图



阶段①：触发 MC_CamIn 指令执行，此时主轴绝对位置为 100

阶段②：等待啮合开始

阶段③：主轴到达啮合开始位置（1100），从轴开始执行啮合动作

阶段④：啮合过程中

阶段⑤：啮合完成，主从轴实现同步

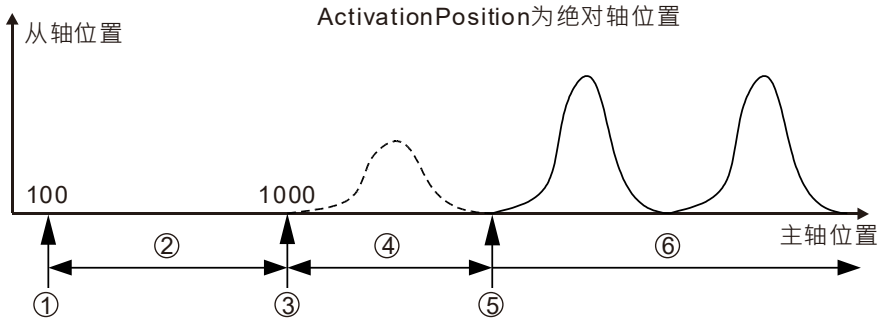
阶段⑥：主从轴同步运动中

➢ ***ActivationPosition* 为绝对轴位置**

当参数 *ActivationMode* =1 时，*ActivationPosition* 为轴位置，且与 *MC_CamIn* 指令开始执行时刻的主轴位置为绝对关系，即实际啮合开始时的主轴位置为 *ActivationPosition*。

例如：*MC_CamIn* 指令开始执行时刻的主轴位置为 100，*ActivationPosition* 为 1000，则实际啮合开始时的主轴位置为 1000（1000=*ActivationPosition*）。

MC_CamIn指令执行过程示意图



阶段①：触发 MC_CamIn 指令执行，此时主轴绝对位置为 100

阶段②：等待啮合开始

阶段③：主轴到达啮合开始位置（1000），从轴开始执行啮合动作

阶段④：啮合过程中

阶段⑤：啮合完成，主从轴实现同步

阶段⑥：主从轴同步运动中

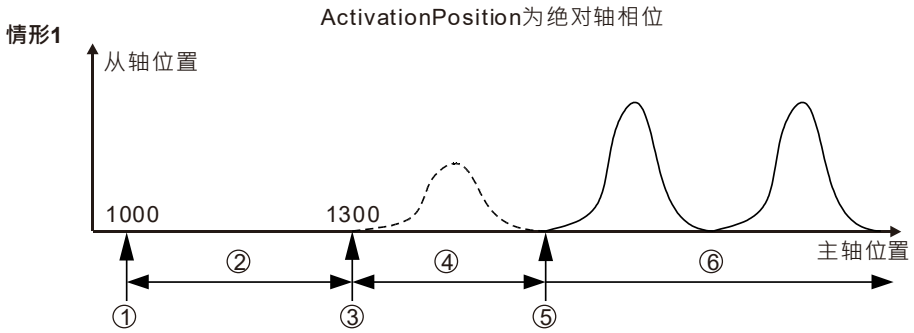
- ***ActivationPosition* 为绝对轴相位**

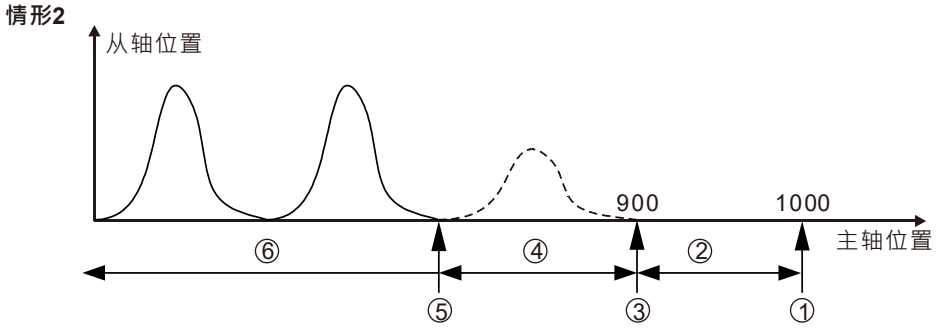
当参数 *ActivationMode* =2 时，*ActivationPosition* 为绝对轴相位（绝对轴相位为轴绝对位置按模做取余运算后的结果）。当主轴绝对轴相位为 *ActivationPosition* 时，从轴开始执行啮合动作。

绝对轴相位具有周期循环特性，在主轴运转过程中，其绝对轴相位可能多次出现等于 *ActivationPosition* 的情形，但仅在 MC_CamIn 指令开始执行后，主轴绝对轴相位第一次等于 *ActivationPosition* 时，从轴开始执行啮合动作。

例如：主轴的模为 400， *ActivationPosition*=100， *MC_CamIn* 指令开始执行时刻的主轴位置为 1000，由于 *MC_CamIn* 指令开始执行时刻的主轴绝对轴相位为 200（200=1000%400），从轴不会执行啮合动作。此后，当主轴位置为 1300（绝对轴相位为 100=1300%400）或 900（绝对轴相位为 100=900%400）时，从轴开始执行啮合动作（%表示取余运算）。

MC_CamIn指令执行过程示意图





阶段①：触发 MC_CamIn 指令执行，此时主轴绝对位置为 1000（绝对轴相位为 200）

阶段②：等待啮合开始

阶段③：主轴到达啮合开始位置（情形 1 为 1300，情形 2 为 900），从轴开始执行啮合动作

阶段④：啮合过程中

阶段⑤：啮合完成，主从轴实现同步

阶段⑥：主从轴同步运动中

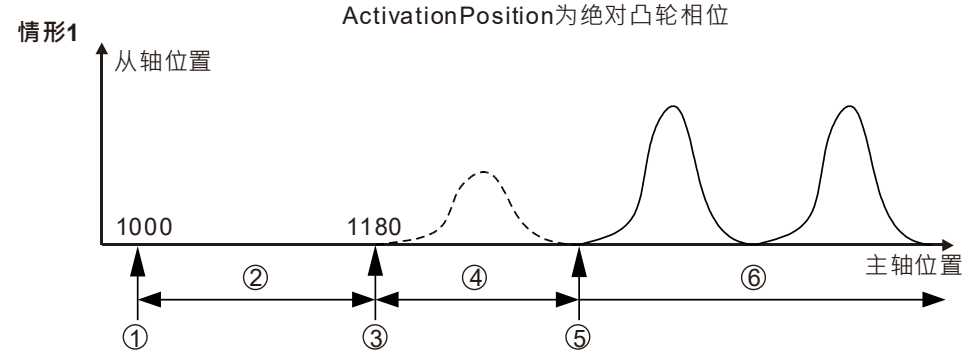
注意：当 *ActivationPosition* 为绝对轴相位时，参数 *ActivationPosition* 的有效范围为：0~模（不包括模）。如果参数 *ActivationPosition* 的值不在有效范围内，*MC_CamIn* 指令执行时，将报错且执行失败！

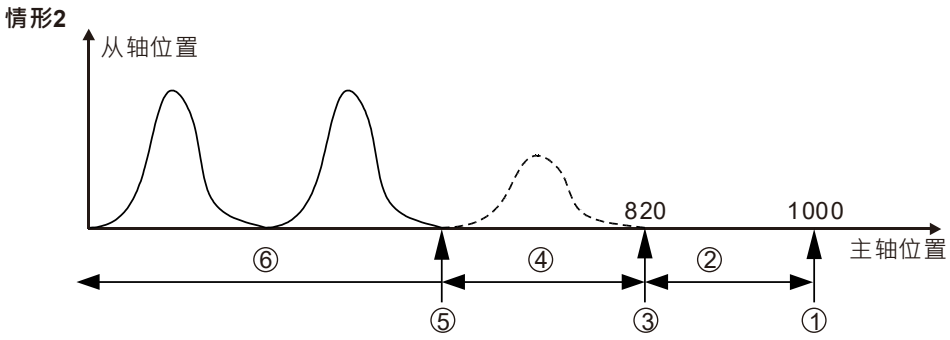
- ***ActivationPosition* 为绝对凸轮相位**

当参数 *ActivationMode*=3 时，*ActivationPosition* 为绝对凸轮相位（绝对凸轮相位为轴绝对位置按其凸轮周期做取余运算后的结果）。当主轴凸轮相位为 *ActivationPosition* 时，从轴开始执行啮合动作。

凸轮相位具有周期循环特性，在主轴运转过程中，其凸轮相位可能多次出现等于 *ActivationPosition* 的情形，但仅在 MC_CamIn 指令开始执行后，主轴凸轮相位第一次等于 *ActivationPosition* 时，从轴开始执行啮合动作。例如：凸轮表中主轴的最大范围值为 360，*ActivationPosition*=100，*MC_CamIn* 指令开始执行时刻的主轴位置为 1000，由于 *MC_CamIn* 指令开始执行时刻的主轴绝对凸轮相位为 280（280=1000%360），从轴不会执行啮合动作。此后，当主轴位置为 1180（绝对凸轮相位为 100=1180%360）或 820（绝对凸轮相位为 100=820%360）时，从轴开始执行啮合动作。

MC_CamIn指令执行过程示意图





阶段①：触发 MC_CamIn 指令执行，此时主轴绝对位置为 1000（绝对凸轮相位为 280）

阶段②：等待啮合开始

阶段③：主轴到达啮合开始位置（情形 1 主轴位置为 1180，情形 2 主轴位置为 820），从轴开始执行啮合动作

阶段④：啮合过程中

阶段⑤：啮合完成，主从轴实现同步

阶段⑥：主从轴同步运动中

注意：当 *ActivationPosition* 为绝对凸轮相位时，参数 *ActivationPosition* 的有效范围为：0~凸轮周期(不包括周期值)。如果参数 *ActivationPosition* 的值不在有效范围内，*MC_CamIn* 指令执行时，将报错且执行失败！

■ **主从轴位置关系**

通过软件预先规划的凸轮关系是主从轴之间的位置关系，此处的"位置"为主从轴的凸轮相位，而非实际轴位置。若将预先规划的凸轮关系视为函数 CAM，则函数 CAM 的输入为主轴凸轮相位，输出为从轴凸轮相位，如下所示：

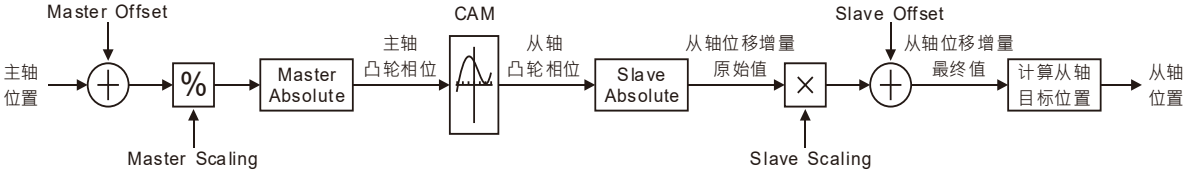
y = CAM (x)

x：主轴凸轮相位

y：从轴凸轮相位

凸轮相位来源于轴位置，它们之间存在换算关系。轴位置与凸轮相位之间的换算关系与参数 *MasterAbsolute*、*SlaveAbsolute*、*MasterOffset*、*SlaveOffset*、*MasterScaling*、*SlaveScaling* 有关，请参阅相关详细内容。

从轴在 *MC_CamIn* 指令的作用下跟随主轴做凸轮同步运动。凸轮同步运动中，主从轴的轴位置对应关系建立在预先规划的凸轮关系（凸轮关系曲线或凸轮表）基础之上，由主轴轴位置计算从轴轴位置的过程如下图所示：



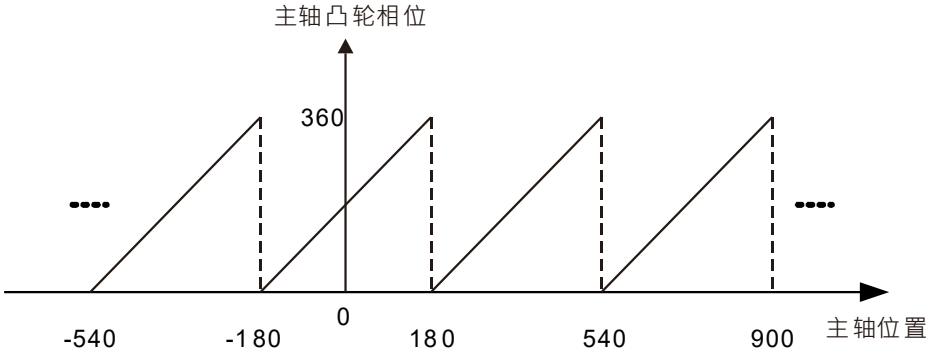
■ **MasterAbsolute 与 SlaveAbsolute**

参数 *MasterAbsolute* 用于指定主轴的轴位置与其凸轮相位之间的对应关系：当参数值为 TRUE 时，为绝对关系；当参数值为 FALSE 时，为相对关系。参数 *SlaveAbsolute* 与 *MasterAbsolute* 同理。

参数 *MasterAbsolute* 和 *SlaveAbsolute* 作用于啮合开始时刻，也就是轴位置与凸轮相位之间的对应关系在啮合开始时刻建立（注意：对应关系是在啮合动作开始时刻建立，而不是 *MC_CamIn* 指令开始执行时刻）。在此之后，凸轮相位的计算，将依照该对应关系。

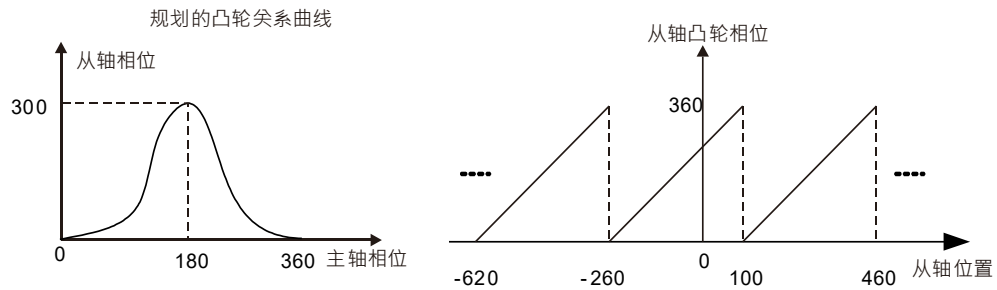
➢ **相对模式**

当参数 *MasterAbsolute* 为 FALSE 时，主轴的轴位置与其凸轮相位之间为相对关系，即啮合开始时刻的主轴的轴位置对应其凸轮相位为 0，此后，主轴凸轮相位的计算，将依照该对应关系。例如：主轴为相对模式，凸轮关系中主轴最大范围值为 360，啮合开始时刻主轴的轴位置为 180，则主轴的轴位置 180 对应其凸轮相位为 0，轴位置 200 对应其凸轮相位为 20（20=（200-180）%360），并可以此类推。这种情形下，主轴的轴位置（主轴位置）与与其凸轮相位之间的关系如下图所示：

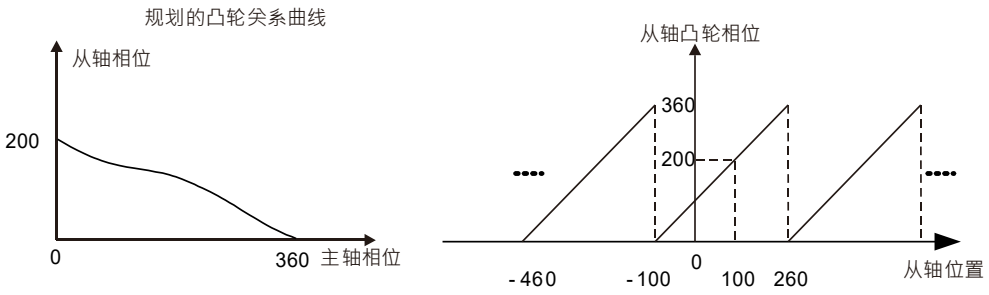


当参数 *SlaveAbsolute* 为 FALSE 时，从轴的轴位置与其凸轮相位之间为相对关系，即啮合开始时刻从轴的凸轮相位与此刻主轴的凸轮相位满足规划的凸轮关系。从轴为相对模式时，确定从轴凸轮相位的方法与主轴不同，在确定从轴凸轮相位时需满足条件：啮合开始时刻从轴的凸轮相位与此刻主轴的凸轮相位满足规划的凸轮关系。例如：从轴为相对模式，凸轮关系中从轴最大范围值为 360，啮合开始时刻从轴的轴位置为 100，若此刻主轴凸轮相位为 0（按凸轮关系要求从轴凸轮相位为 0），则从轴轴位置 100 对应其凸轮相位为 0，如下图情形 1 所示；若按凸轮关系要求从轴凸轮相位为 200，则从轴轴位置 100 对应其凸轮相位为 200，如下图情形 2 所示。

情形1

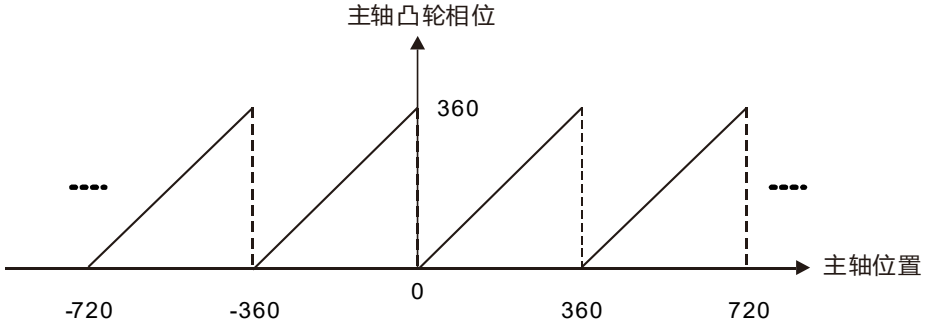


情形2



➢ **绝对模式**

当参数 *MasterAbsolute* 为 TRUE 时，主轴的轴位置与其凸轮相位之间为绝对关系，任意时刻，主轴的凸轮相位等于该时刻主轴的轴位置与凸轮关系中的主轴最大范围值做取余运算的结果。例如：主轴为绝对模式，凸轮关系中的主轴最大范围值为 360，则主轴的轴位置为 100 时，其凸轮相位为 100（100=100%360）；主轴的轴位置为 500 时，其凸轮相位为 140（140=500%360），并可以此类推，主轴轴位置与与其凸轮相位之间的关系如下图所示



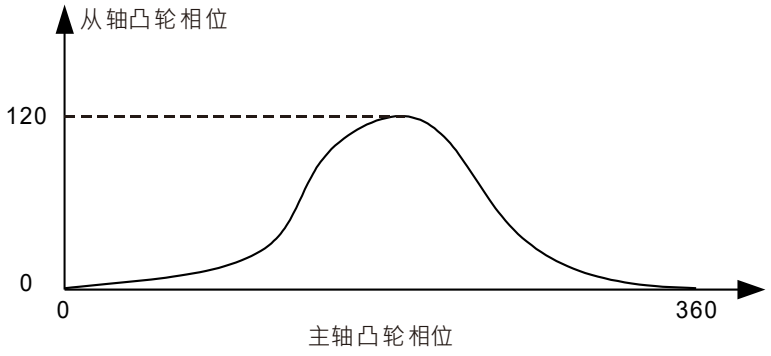
当参数 *SlaveAbsolute* 为 TRUE 时，从轴的轴位置与其凸轮相位之间为绝对关系，任意时刻，从轴的凸轮相位等于该时刻从轴的轴位置与凸轮关系中的从轴最大范围值做取余运算的结果。从轴为绝对模式时，其轴位置与其凸轮相位之间的对应关系与主轴一致。

■ **位置偏移和位置比例缩放（Offset 和 Scaling）**

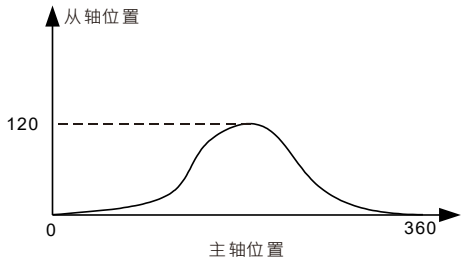
主从轴的凸轮关系为预先规划，但在执行凸轮时，可通过参数"*Offset*"和"*Scaling*"在预先规划的凸轮关系基础上进行位置偏移或位置缩放，例如：加工的同一产品有几种不同尺寸，则只需规划一种凸轮关系，然后通过改变参数"*Offset*"和"*Scaling*"以适应不同尺寸产品之间的加工切换。

参数 *MasterOffset* 在主轴为相对或绝对模式均有效；参数 *SlaveOffset* 仅在**从轴为绝对模式**（*SlaveAbsolute*=TRUE）时有效，从轴为相对模式（*SlaveAbsolute*=FALSE）时无效。

主从轴的位置偏移和比例缩放共同决定实际执行的凸轮关系，其作用效果通过以下范例进行描述。预先规划的凸轮关系如下图所示：

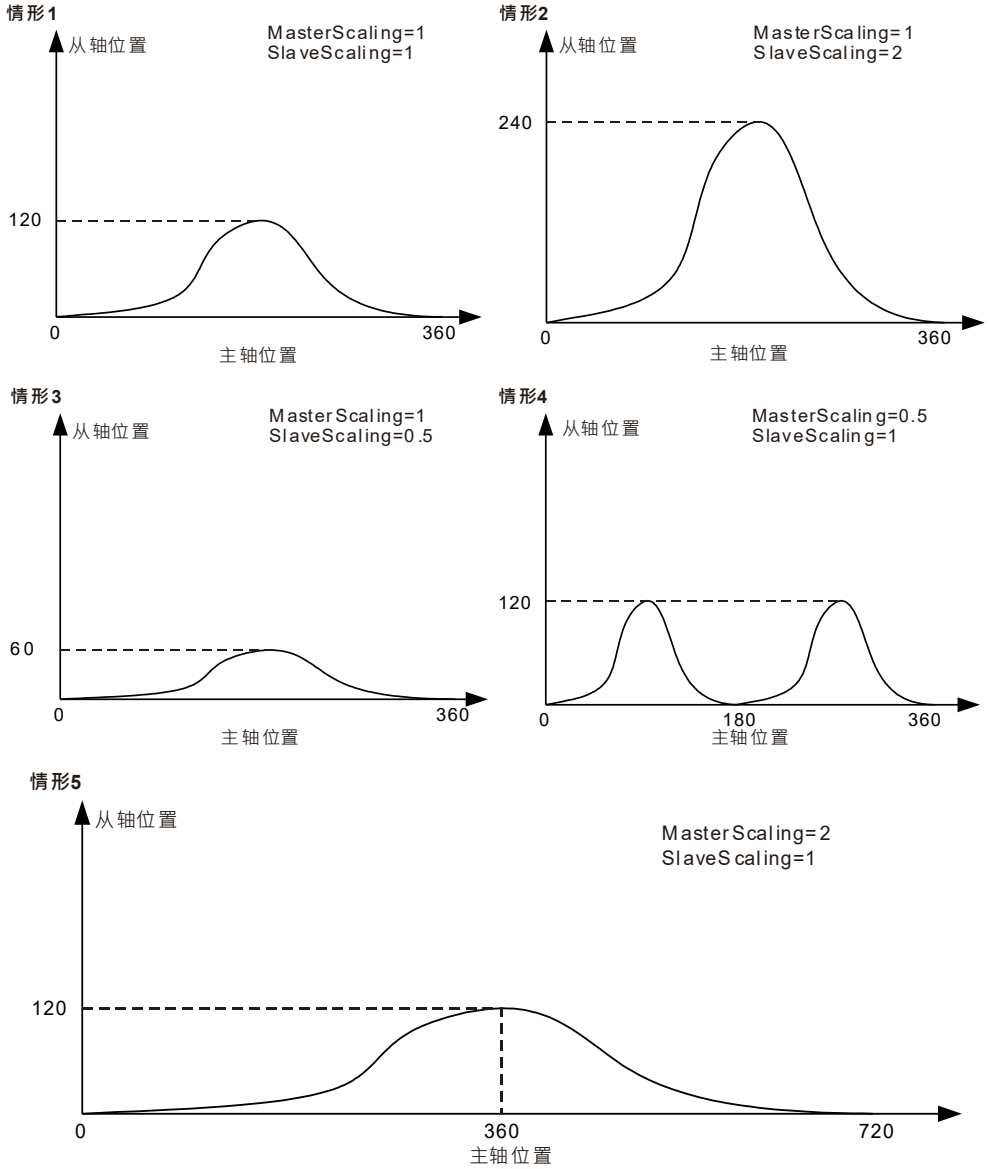


若主从轴均为绝对模式，且执行啮合动作时，主从轴的轴位置均为 0，在不使用偏移和缩放时（默认值），凸轮执行过程中的主从轴实际位置对应关系如下图所示：



当位置偏移量或缩放比例不为默认值时，其对凸轮执行过程中主从轴实际位置对应关系的影响如下：

➢ **主从轴偏移量为 0，主从轴缩放比例对实际执行的凸轮关系的影响**



情形1：当主从轴缩放比例为 1，偏移量为 0 时，实际凸轮关系与预先规划的一致。

情形2：当主轴缩放比例为 1，从轴缩放比例为 2，主从偏移量为 0 时，与主轴位置对应的从轴位置变为预先规划的两倍。

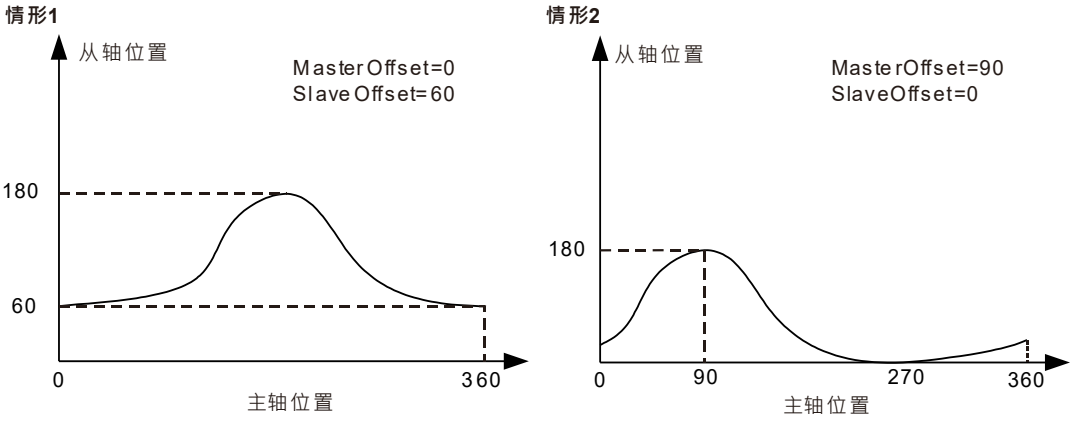
情形3：当主轴缩放比例为 1，从轴缩放比例为 0.5，主从轴偏移量为 0 时，与主轴位置对应的从轴位置变为预先规划的 1/2。

情形4： 当主轴缩放比例为 0.5，从轴缩放比例为 1，主从轴凸轮偏移量为 0 时，与从轴位置对应的主轴位置变为预先规划的 1/2。如果从凸轮相位的角度看，则是主轴的凸轮相位为预先规划的 1/2，即主轴凸轮周期从 360 变为 180（180=360*0.5），从轴凸轮相位不变。

情形5： 当主轴缩放比例为 2，从轴缩放比例为 1，主从轴偏移量为 0 时，与从轴位置对应的主轴位置变为预先规划的两倍。如果从凸轮相位的角度看，则是主轴凸轮相位为原来的两倍，即主轴凸轮周期从 360 变为 720（720=360*2），从轴凸轮相位不变。

➢ **主从轴缩放比例为 1，主从轴偏移实际执行的凸轮关系的影响**

主轴偏移相当于将执行凸轮时的实际轴位置关系曲线作横向移动；从轴偏移相当于将执行凸轮时的轴位置关系曲线作纵向移动。



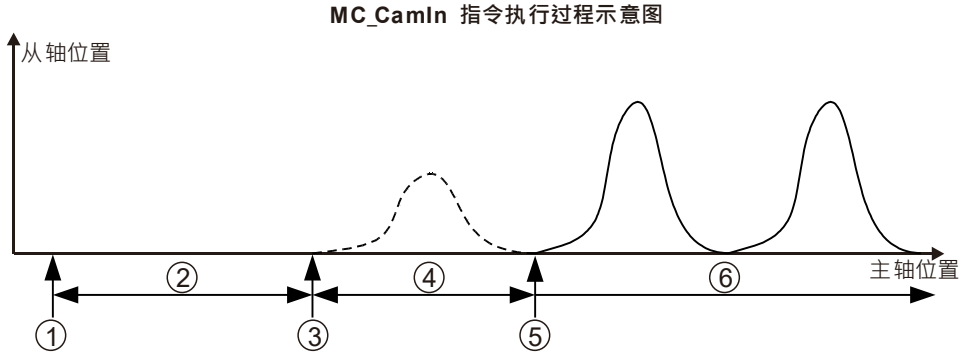
情形1： 当主从轴缩放比例为 1，主轴偏移量为 0，从轴偏移量为 60 时，与主轴位置对应的从轴位置均在预先规划的基础上加上 60。例如：规划的凸轮关系中，主轴轴位置 180 对应的从轴轴位置为 180，实际执行时，对应的从轴轴位置为 240（240=180+60）。

情形2： 当主从轴缩放比例为 1，主轴偏移量为 90，从轴偏移量为 0 时，与从轴轴位置对应的主轴轴位置在预先规划的基础上偏移 90（加上偏移量）。例如：规划的凸轮关系中，主轴轴位置 180 对应的从轴轴位置为 180，实际执行时，主轴轴位置 90 对应的从轴轴位置为 180，即预先规划的凸轮关系中主轴轴位置 180（180=90+90）所对应的从轴轴位置。

■ **StartMode**

啮合过程中，从轴的动作方式可通过参数 *StartMode* 进行指定，即 *StartMode* 作用于 *MC_CamIn* 指令的第四阶段，如下图所示：

MC_CamIn 指令执行过程示意图



阶段①：触发 MC_CamIn 指令执行

阶段②：等待啮合开始

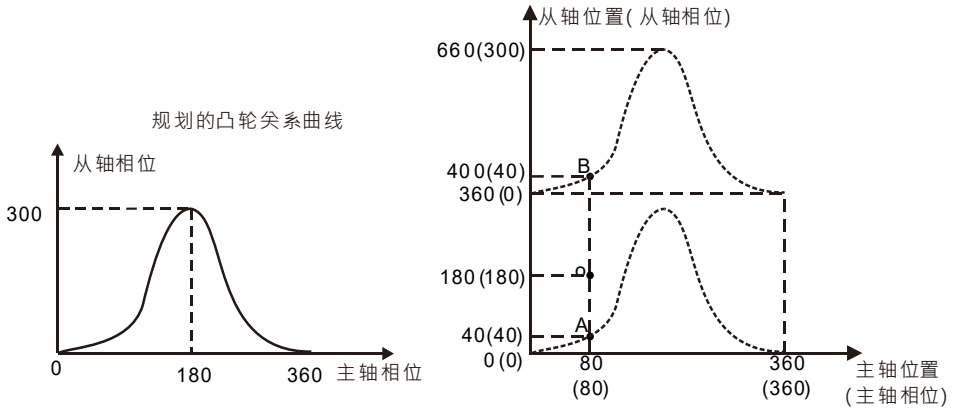
阶段③：主轴到达啮合开始位置，从轴开始执行啮合动作

阶段④：啮合过程中

阶段⑤：啮合完成，主从轴实现同步

阶段⑥：主从轴同步运动中

凸轮同步要求主从轴的凸轮相位满足定义的凸轮关系，啮合过程则为从轴朝同步相位运动的过程，该同步相位与主轴凸轮相位满足定义的凸轮关系。由于轴的凸轮相位具有周期循环特性，即每个凸轮相位均有多个轴位置与之对应，啮合时，期望的同步位置存在多种选择，也就使得啮合方式有多种选择。例如：开始执行啮合时，主从轴的凸轮相位分别为 80 和 180（如右下图中的 O 点），但定义的凸轮关系要求从轴的凸轮相位为 40，则此刻从轴期望的同步位置为 40 或 400（如右下图中的点 A 和点 B），可通过参数 *StartMode* 选择啮合过程是从 O 到 A 还是 O 到 B。



StartMode 有三种模式供选择，分别为：最短距离渐变（*StartMode*=0）、正向渐变（*StartMode*=1）和反向渐变（*StartMode*= -1），用户可根据实际需求选择不同的啮合模式。

- ***StartMode=0*（最短距离渐变）**

  若参数 *StartMode*=0，执行啮合动作时，从轴朝与同步位置距离最短的方向运动，此时从轴的运动受参数 *Velocity*、*Acceleration*、*Deceleration* 影响。

- ***StartMode=1*（正向渐变）**

  若参数 *StartMode*=1， 执行啮合动作时，从轴正向朝同步位置运动， 此时从轴的运动受参数 *Velocity*、*Acceleration*、*Deceleration* 影响。

- ***StartMode= -1*（反向渐变）**

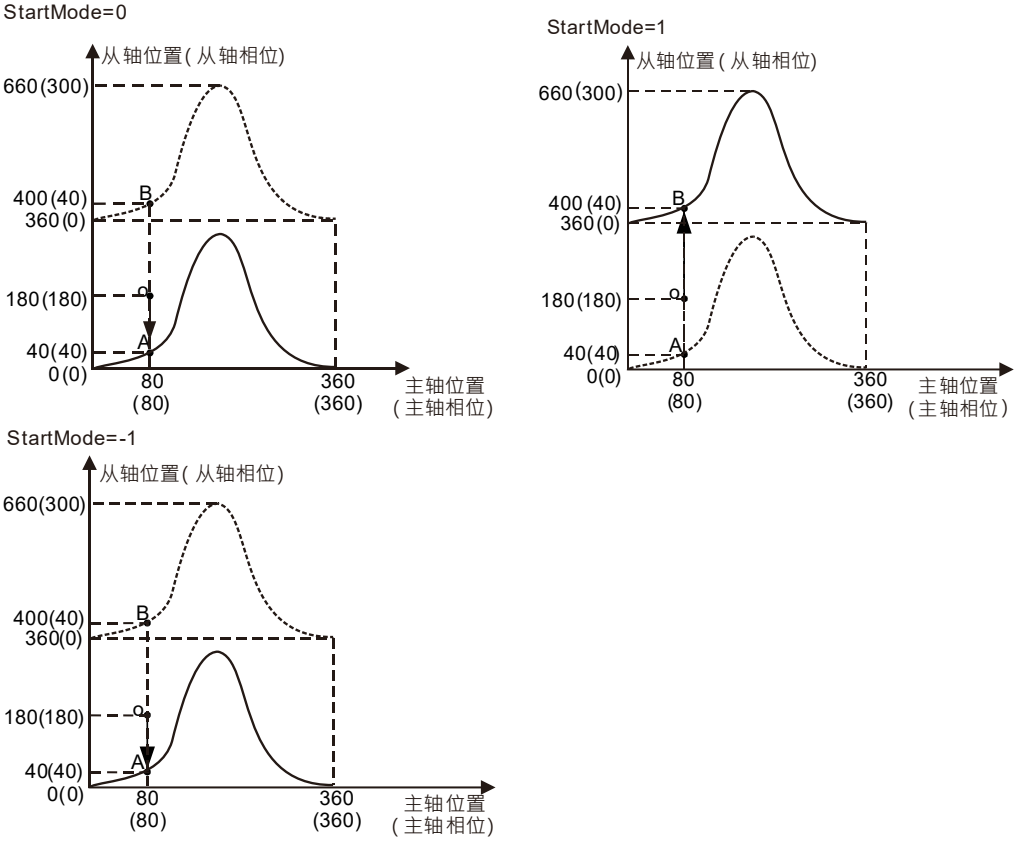
  若参数 *StartMode*= -1，执行啮合动作时，从轴反向朝同步位置运动，此时从轴的运动受参数 *Velocity*、*Acceleration*、*Deceleration* 影响。

例如：开始执行啮合动作时，主从轴的凸轮相位分别为 80 和 180（如下图的 O 点），根据定义的凸轮关系，要求主轴凸轮相位为 80 时，从轴凸轮相位为 40（如下图中的 A 点或 B 点），则 *StartMode* 选择不同模式时，啮合过程中，从轴的动作方式如下图所示。

StartMode=0：从轴立即从 O 点渐变到 A 点，在 A 点实现同步，因为从 O 点到 A 点的距离小于 O 点到 B 的距离；

StartMode=1：从轴正向立即从 O 点渐变到 B 点；

StartMode= -1：从轴反向立即从 O 点渐变到 A 点；

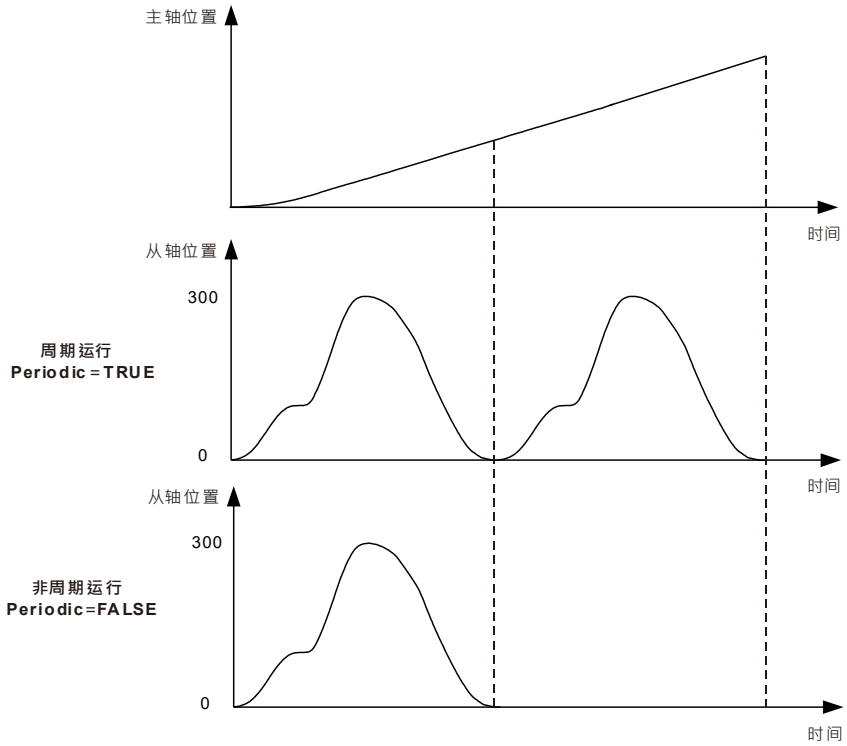


■ **周期/非周期执行凸轮（Periodic）**

在电子凸轮实际应用中，有的可能需要按周期循环运行，而有的则可能仅需要运行一个周期，参数 *Periodic* 即用于对这两种情形进行选择。

当参数 *Periodic*=TRUE 时，从轴跟随主轴按周期循环执行凸轮，直到凸轮关系解除；

当参数 *Periodic*=FALSE 时，从轴与主轴凸轮同步后，执行到凸轮周期结束点时，从轴与主轴的凸轮关系解除，且从轴立即停止运动。



■ **其它指令对凸轮运行的影响**

➢ ***MC_CamOut* 对凸轮运行的影响**

需要终止已经运行的凸轮，可使用 *MC_CamOut* 指令实现。

➢ ***MC_SetPosition* 对凸轮运行的影响**

由于 *MC_SetPosition* 指令的执行对正在执行的运动指令不会产生影响，故在凸轮运行中，对主从轴执行 *MC_SetPosition* 指令，不会对凸轮产生影响；在 *MC_SetPosition* 指令执行之后触发运行的凸轮会受 *MC_SetPosition* 指令对轴位置改变的影响

➢ ***MC_Stop*、*MC_Halt* 对凸轮运行的影响**

指令 *MC_Stop*、*MC_Halt* 作用于从轴时，*MC_CamIn* 指令被中断，凸轮关系解除，从轴减速停止。

➢ ***MC_Home* 对凸轮运行的影响**

MC_Home 指令不可作用于从轴，但可作用于主轴。*MC_Home* 指令作用于主轴时，可能使得主轴位置在极短的时间内发生较大的变化，从而引起从轴抖动，故建议在同步关系解除后执行 *MC_Home* 指令。

■ **其它注意事项**

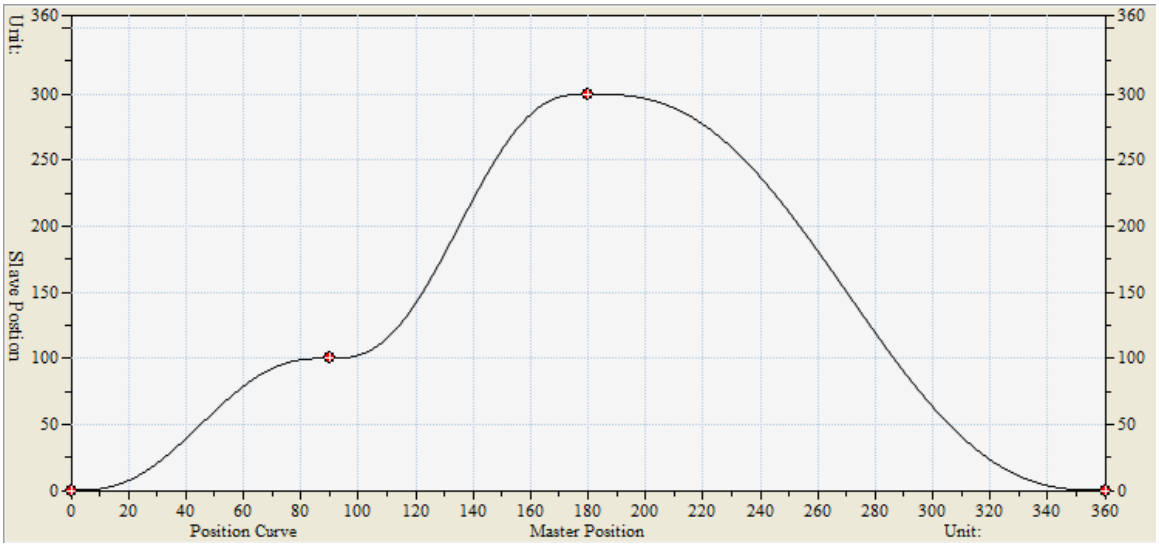
不同类型的轴在凸轮关系中为主轴或为从轴的规则如下表：

轴类型	作为凸轮主轴	作为凸轮从轴
伺服实轴	允许	允许
编码器	允许	不允许
虚轴	允许	允许

程序范例

本例描述 ***MC_CamIn*** 指令的执行效果。

1. 规划的凸轮关系曲线如下图所示：



2. 凸轮关系关键点如下表：

编号	主轴凸轮相位	从轴凸轮位置	速度	加速度
1	0	0	0	0
2	90	100	0	0
3	180	300	0	0
4	360	0	0	0

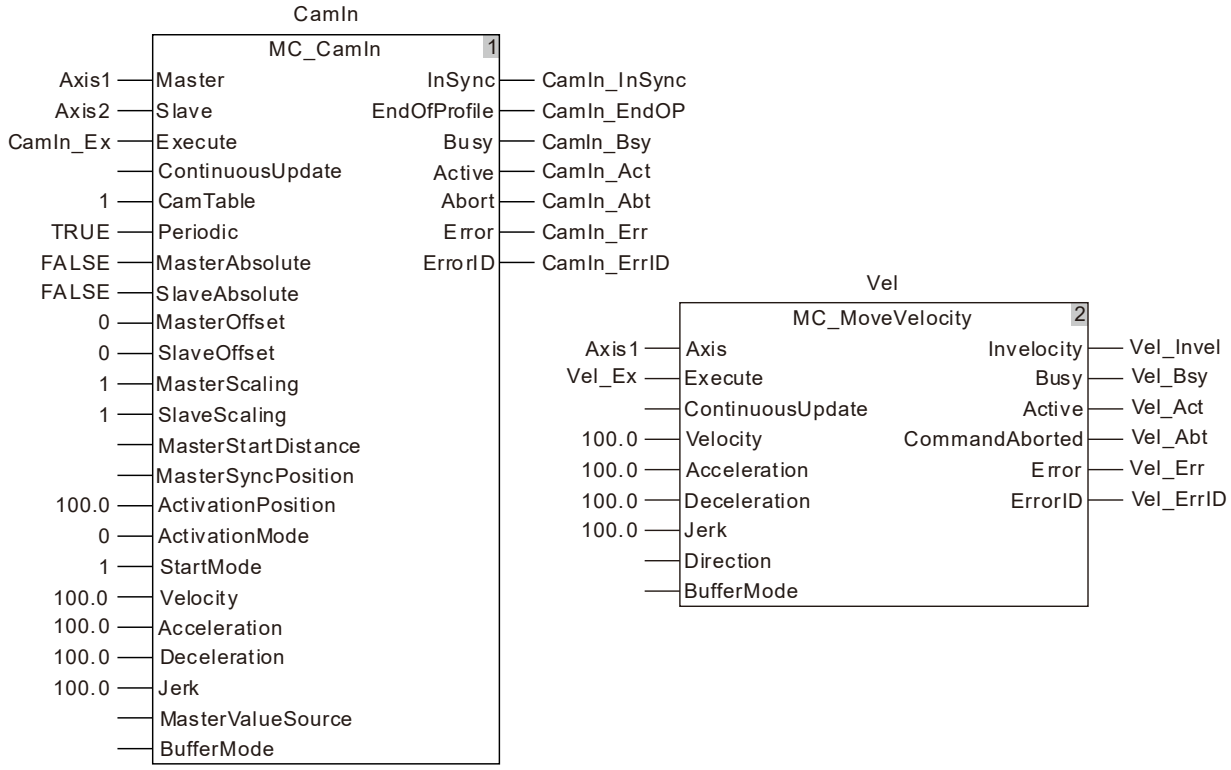
3. 程序说明：

主从轴凸轮周期	360
主从轴凸轮相位缩放比例	1
主从轴相位偏移量	0
主轴凸轮相位模式	相对
从轴凸轮相位模式	相对
周期执行/非周期执行	周期执行
啮合开始位置	相对轴位置：100
啮合模式	最短距离

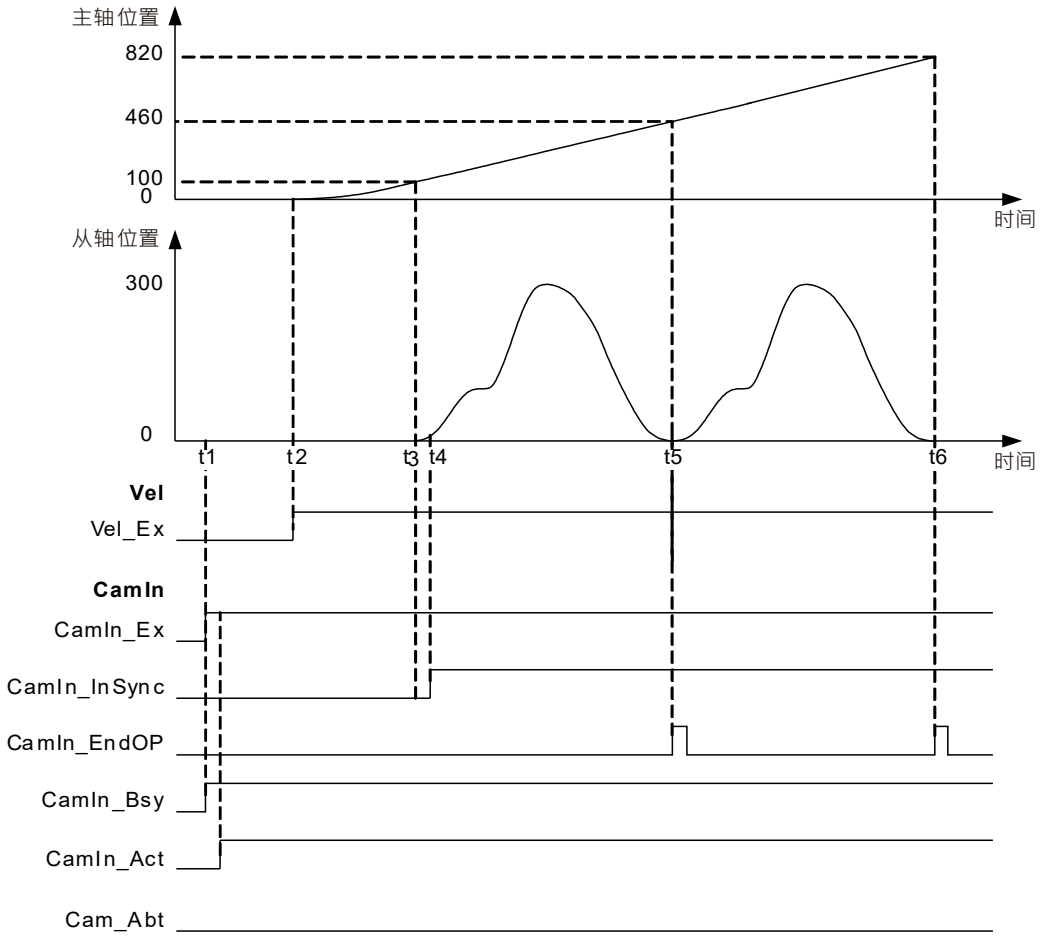
4. 变量和程序

变量名	数据类型	初始值
CamIn	MC_CamIn	
CamIn_Ex	BOOL	
CamIn_InSync	BOOL	
CamIn_EndOP	BOOL	
CamIn_Bsy	BOOL	
CamIn_Act	BOOL	
CamIn_Abt	BOOL	

变量名	数据类型	初始值
CamIn_Err	BOOL	
CamIn_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel _Ex	BOOL	
Vel _InVel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ ErrID	WORD	



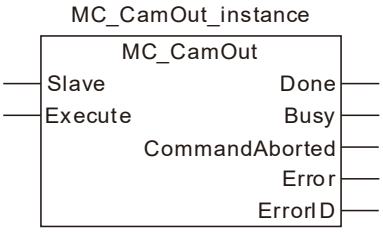
5. **运动曲线和时序图如下：**



- ❖ 当变量 CamIn_Ex 由 FALSE 变为 TRUE 时， MC_CamIn 指令开始执行，如上图 t1 时刻，此刻主从轴的轴位置均为 0。由于参数 ActivationPosition 为 100，ActivationMode 为 0，则需当主轴位置为 100（t1 时刻的主轴位置+ActivationPosition）时，从轴才开始执行啮合动作。
- ❖ 当变量 Vel_Ex 由 FALSE 变为 TRUE 时， MC_MoveVelocity 指令开始执行，如上图 t2 时刻，此刻主轴的轴位置为 0，从轴继续处于等待啮合动作开始的状态。此后主轴在 MC_MoveVelocity 指令的作用下从 0 开始正向运动。
- ❖ 当主轴轴位置经过 100 时，到达啮合动作开始位置，如上图中的 t3 时刻。如上图所示，从轴在 t3 时刻按 StartMode 开始执行啮合动作，并在 t4 时刻实现同，步输出参数 InSync（变量 CamIn1_InSync）由 FALSE 变为 TRUE。
- ❖ 每当主从同步运动到凸轮周期的结束点(如上图中的 t5 和 t6 时刻所示)时 ,输出参数 EndOfProfile（变量 CamIn1_EndPro）均变为 TRUE，并在一个程序周期后变为 FALSE。

11.4.6 MC_CamOut（电子凸轮脱离指令）

FB/FC	说明	适用机种
FB	此指令用于解除已建立的电子凸轮关系。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Slave （从轴的站号）	指定欲解除凸轮关系的从轴。	USINT	请参考第 2.2 节功能简介 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-

- **输出参数**

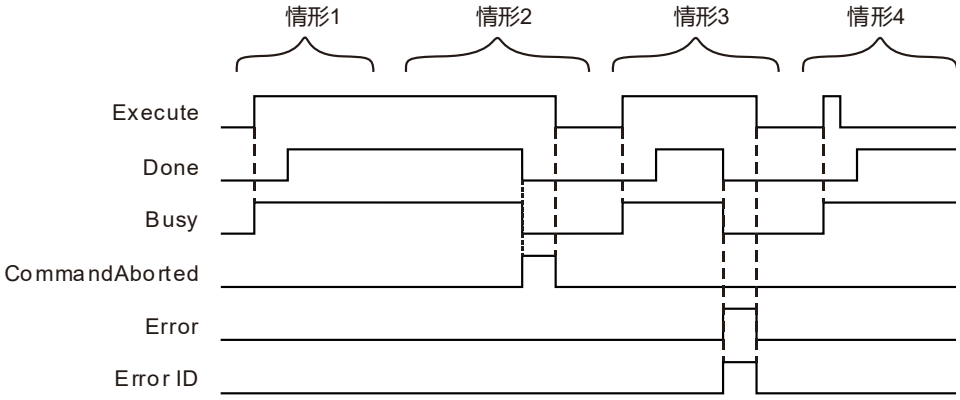
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当从轴和主轴的电子齿轮关系解除时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted

名称	变为 TRUE 的时机	变为 FALSE 的时机
		被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，*Execute* 由 TRUE 变为 FALSE 后，*Busy* 和 *Done* 依然保持为 TRUE。

情形2： 当 *Execute* 为 TRUE 时，若指令被其它指令中断，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Done* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 时，当有错误产生时(如轴去使能)，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Done* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中，不到一个周期时，*Execute* 由 TRUE 变为 FALSE 后，当到达一个周期时，*Done* 变为 TRUE，且 *Busy* 仍然保持为 TRUE。

- **功能说明**

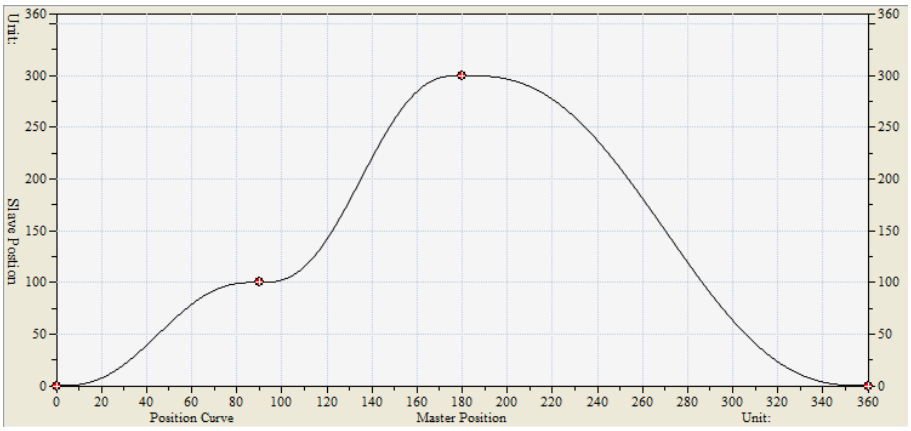
MC_CamOut 指令用于解除已建立的电子凸轮关系。指令作用于凸轮从轴，且从轴将保持脱离时的速度继续运行。

若需控制凸轮从轴停止运动，可对从轴使用 MC_Halt 或者 MC_Stop 指令。MC_Halt 或者 MC_Stop 指令执行完成后，从轴停止运动且凸轮关系解除。

程序范例

本例描述 *MC_CamOut* 指令的执行效果。

1. 规划的凸轮关系曲线如下图所示：



2. 凸轮关系关键点如下表：

编号	主轴凸轮相位	从轴凸轮位置	速度	加速度
1	0	0	0	0
2	90	100	0	0
3	180	300	0	0
4	360	0	0	0

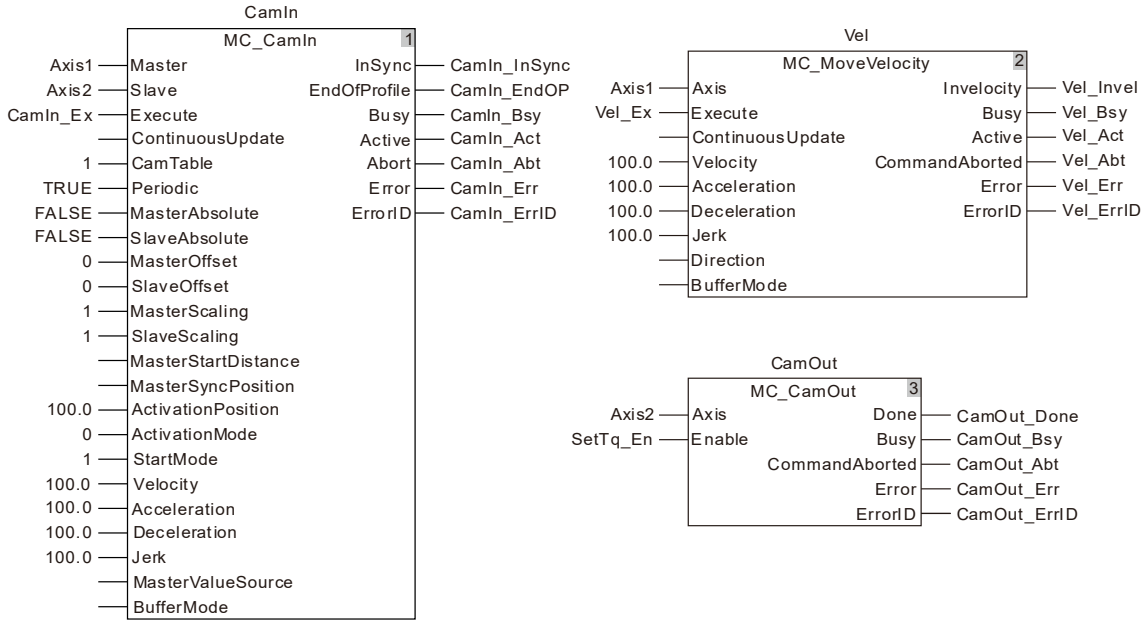
3. 程序说明：

主从轴凸轮周期	360
主从轴凸轮相位缩放比例	1
主从轴相位偏移量	0
主轴凸轮相位模式	相对
从轴凸轮相位模式	相对
周期执行/非周期执行	周期执行
啮合开始位置	相对轴位置：100
啮合模式	最短距离

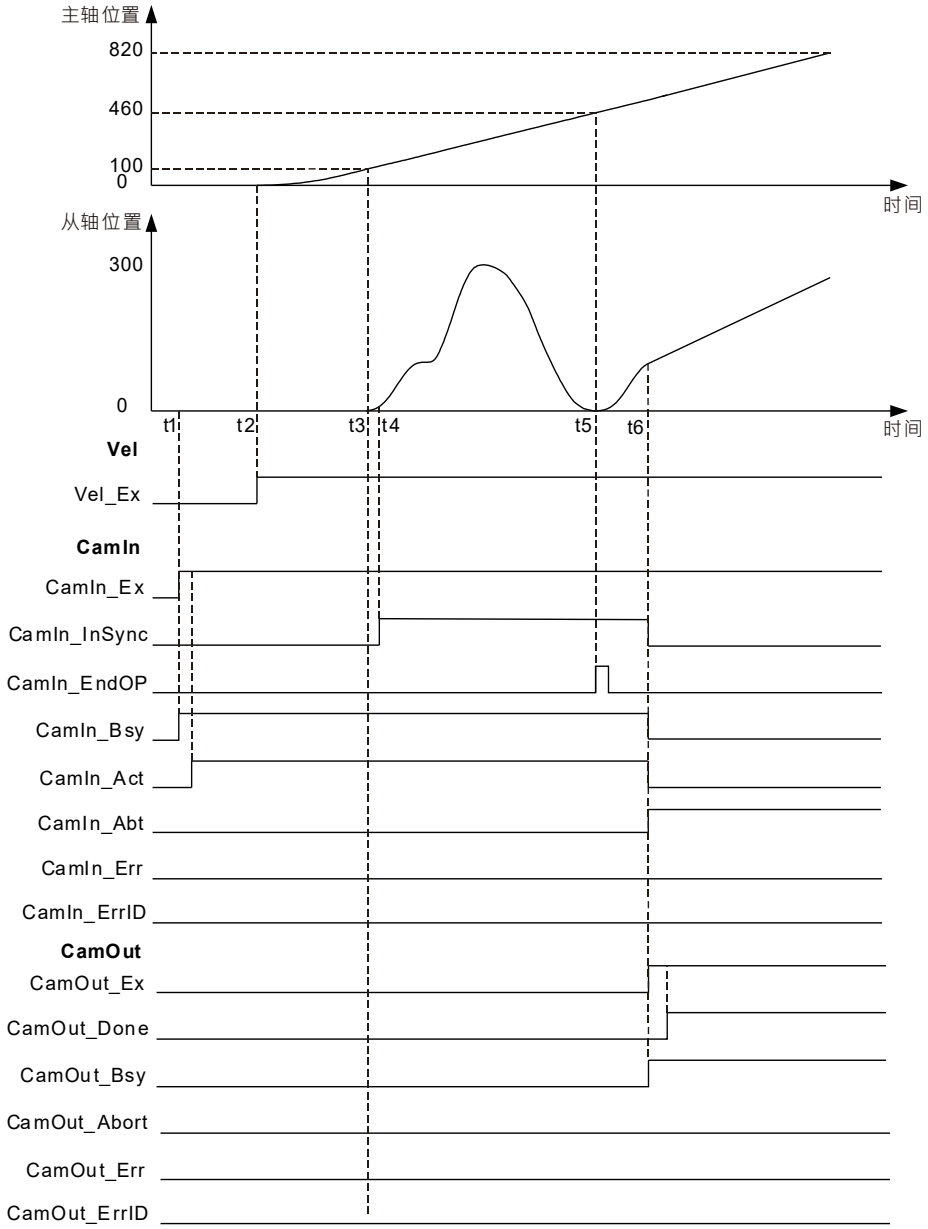
4. 变量和程序

变量名	数据类型	初始值
CamIn	MC_CamIn	
CamIn_Ex	BOOL	
CamIn_InSync	BOOL	
CamIn_EndOP	BOOL	
CamIn_Bsy	BOOL	
CamIn_Act	BOOL	
CamIn_Abt	BOOL	
CamIn_Err	BOOL	
CamIn_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	
Vel_InVel	BOOL	
Vel_Bsy	BOOL	

变量名	数据类型	初始值
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
CamOut	MC_CamOut	
CamOut_Ex	BOOL	
CamOut_Done	BOOL	
CamOut_Bsy	BOOL	
CamOut_Abt	BOOL	
CamOut_Err	BOOL	
CamOut_ErrID	WORD	



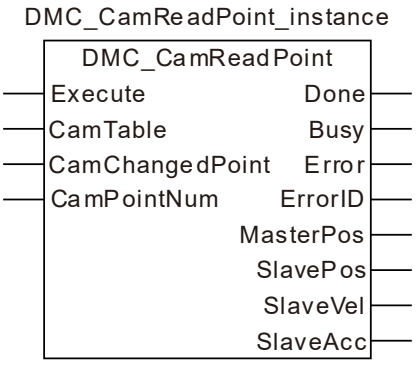
5. **运动曲线和时序图如下：**



- 当变量 CamIn_Ex 由 FALSE 变为 TRUE 时， MC_CamIn 指令开始执行，如上图 t1 时刻，此刻主从轴的轴位置均为 0。由于参数 ActivationPosition 为 100，ActivationMode 为 0，则需当主轴位置为 100（t1 时刻的主轴位置+ActivationPosition）时，从轴才开始执行啮合动作。
- 当变量 Vel_Ex 由 FALSE 变为 TRUE 时， MC_MoveVelocity 指令开始执行，如上图 t2 时刻，此刻主轴的轴位置为 0，从轴继续处于等待啮合动作开始的状态。此后主轴在 MC_MoveVelocity 指令的作用下从 0 开始正向运动。
- 当主轴轴位置经过 100 时，到达啮合动作开始位置，如上图中的 t3 时刻。如上图所示，从轴在 t3 时刻按 StartMode 开始执行啮合动作，并在 t4 时刻实现同，步输出参数 InSync（变量 CamIn1_InSync）由 FALSE 变为 TRUE。
- 在从轴跟随主轴做凸轮同步运动过程中，执行 MC_CamOut 指令解除凸轮关系，如上图中的 t6 时刻。MC_CamOut 指令执行后，从轴以脱离时刻的速度继续运行。

11.4.7 DMC_CamReadPoint（读取凸轮点信息指令）

FB/FC	说明	适用机种
FB	此指令用于读取凸轮点信息。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
CamTable（电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64（不可缺省）	Execute 由 FALSE 变为 TRUE
CamChangedPoint（凸轮点更改前或后信息选择）	当为 FALSE 时，指令读取改凸轮点更改前信息； 当为 TRUE 时，指令读取改凸轮点更改后信息。	BOOL	TRUE / FALSE	Execute 由 FALSE 变为 TRUE
CamPointNum（凸轮点编号）	选择读取的凸轮点	UINT	1~2048（不可缺省）	Execute 由 FALSE 变为 TRUE

- **输出参数**

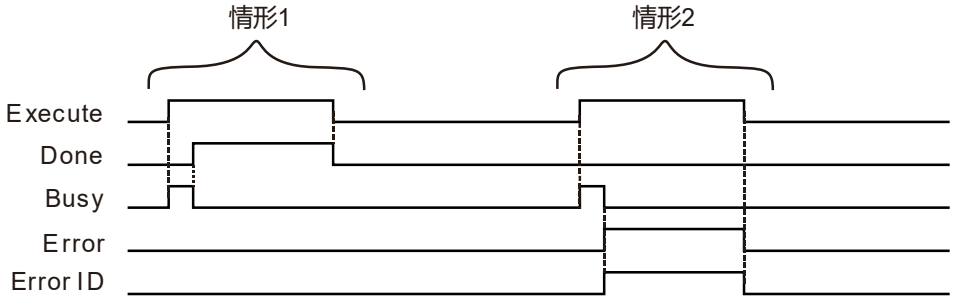
名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
MasterPos	读取指令设置电子凸轮点的主轴位置。	LREAL	0、正数

名称	功能	数据类型	输出范围
(主轴位置)			
SlavePos (从轴位置)	读取指令设置电子凸轮点的从轴位置。	LREAL	0、正数
SlaveVel (从轴速度)	读取指令设置电子凸轮点的从轴速度。	LREAL	0、正数
SlaveAcc (从轴加速度)	读取指令设置电子凸轮点的从轴加速度。	LREAL	0、正数

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ Execute 位为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，当指令执行完成时，Done 变为 TRUE，同时 Busy 位变为 FALSE。当 Execute 变为 FALSE 时，Done 变为 FALSE。

情形2： 当有错误产生时，Error 变为 TRUE，ErrorID 显示对应的错误码，当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 的值会被清除。

- **功能说明**

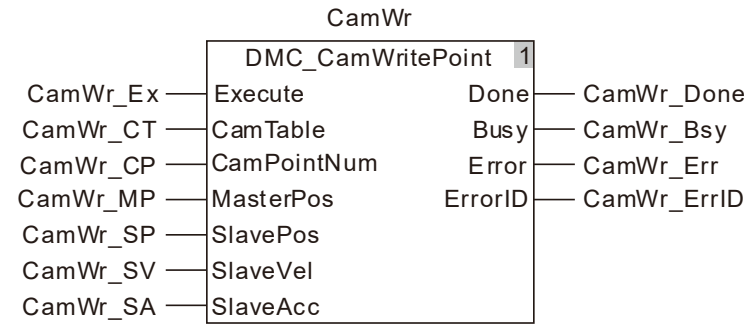
此指令用于读取电子凸轮表中凸轮点的信息。当 CamChangedPoint 为 FALSE 时，读取到的是 DMC_CamSet 指令更改前的凸轮点信息，当 CamChangedPoint 为 TRUE 时，读取到的是 DMC_CamSet 指令更改后的凸轮点信息。

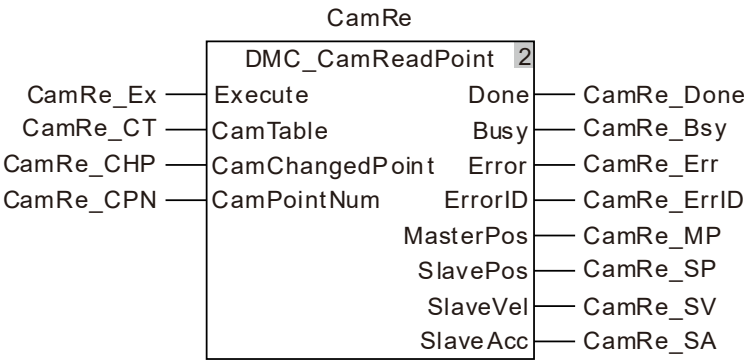
程序范例

1. 变量和程序

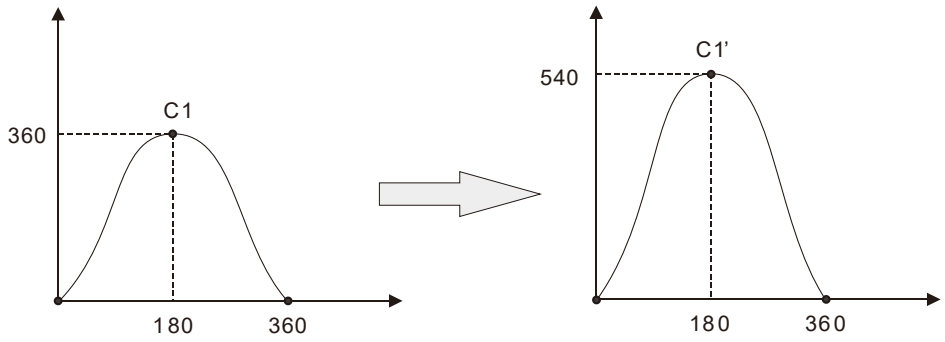
变量名	数据类型	初始值
CamWr	DMC_CamWritePoint	

变量名	数据类型	初始值
CamWr_Ex	BOOL	FALSE
CamWr_CT	USINT	1
CamWr_CP	UINT	2
CamWr_MP	LREAL	180.0
CamWr_SP	LREAL	540.0
CamWr_SV	LREAL	0.0
CamWr_SA	LREAL	0.0
CamWr_Done	BOOL	
CamWr_Bsy	BOOL	
CamWr_Err	BOOL	
CamWr_ErrID	WORD	
CamRe	DMC_CamReadPoint	
CamRe_Ex	BOOL	FALSE
CamRe_CT	USINT	1
CamRe_CHP	BOOL	TRUE
CamRe_CPN	UINT	2
CamRe_Done	BOOL	
CamRe_Bsy	BOOL	
CamRe_Err	BOOL	
CamRe_ErrID	WORD	
CamRe_MP	LREAL	
CamRe_SP	LREAL	
CamRe_SV	LREAL	
CamRe_SA	LREAL	





2. 凸轮曲线



如上图所示，凸轮曲线 C1 更改为曲线 C1′。

- ❖ 凸轮曲线中有三个凸轮点，CamWr_Ex 为 TRUE 则执行 DMC_CamWritePoint 指令，当 CamWr_Done 为 TRUE 时表示该指令写入凸轮点信息完成，程序中是写第 2 个凸轮点信息。
- ❖ CamRe_Ex 为 TRUE 则执行 DMC_CamReadPoint 指令，当引脚 CamChangedPoint (变量 CamRe_CHP) 为 FALSE 时，该指令读取到的凸轮点信息是写入点前的凸轮点信息，如下表：

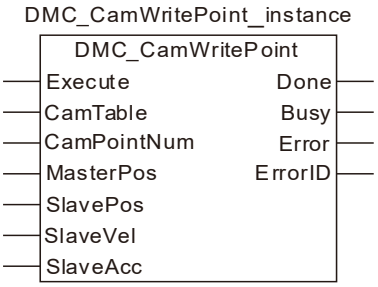
编号	主轴	从轴	速度	加速度
2	180.0	360.0	0.0	0.0

- ❖ 当引脚 CamChangedPoint (变量 CamRe_CHP) 为 TRUE 时，该指令读取到的凸轮点信息是写入点后的凸轮点信息，如下表：

编号	主轴	从轴	速度	加速度
2	180.0	540.0	0.0	0.0

11.4.8 DMC_CamWritePoint（写入凸轮点信息指令）

FB/FC	说明	适用机种
FB	此指令用于写入凸轮点信息。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 Execute 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
CamTable（电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64（不可缺省）	Execute 由 FALSE 变为 TRUE
CamPointNum（凸轮点编号）	写入凸轮点的编号	UINT	1~2048（不可缺省）	Execute 由 FALSE 变为 TRUE
MasterPos（主轴位置）	写入电子凸轮点的主轴位置。	LREAL	0、正数	-
SlavePos（从轴位置）	写入电子凸轮点的从轴位置。	LREAL	0、正数	-
SlaveVel（从轴速度）	写入电子凸轮点的从轴速度。	LREAL	0、正数	-
SlaveAcc（从轴加速度）	写入电子凸轮点的从轴加速度。	LREAL	0、正数	-

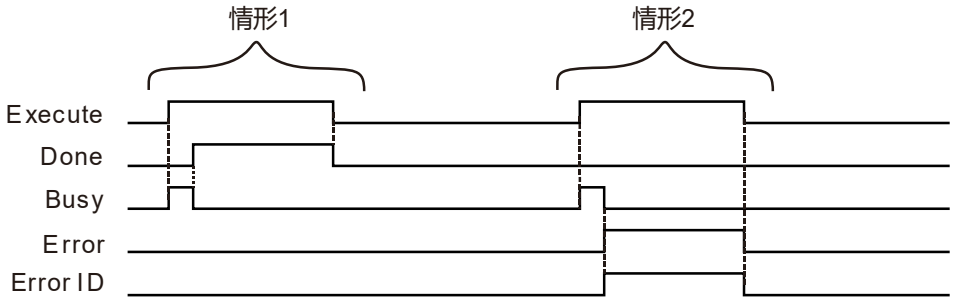
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ Execute 位为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 CommandAborted 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，当指令执行完成时，Done 变为 TRUE，同时 Busy 位变为 FALSE。当 Execute 变为 FALSE 时，Done 变为 FALSE。

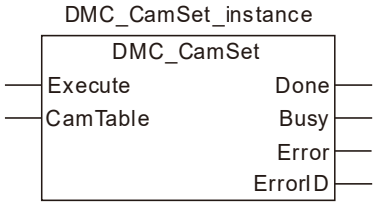
情形2： 当有错误产生时，Error 变为 TRUE，ErrorID 显示对应的错误码，当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 的值会被清除。

- **功能说明**

此指令用于写电子凸轮表中凸轮点的信息。写电子凸轮点成功后，更改后的曲线并不能立即生效，需要执行 DMC_CamSet 指令更改后凸轮曲线才生效。使用范例请参考 DMC_CamSet 指令中的范例说明。

11.4.9 DMC_CamSet（更改凸轮点生效指令）

FB/FC	说明	适用机种
FB	此指令用于将更改的凸轮点生效。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
CamTable （电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64 （不可缺省）	Execute 由 FALSE 变为 TRUE

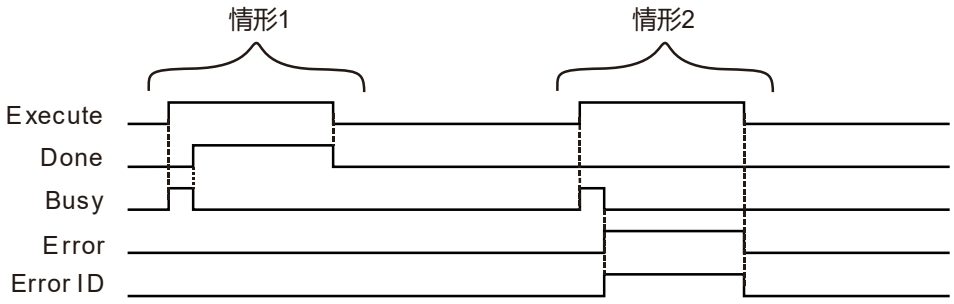
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ Execute 位为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，当指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 位变为 FALSE。当 *Execute* 变为 FALSE 时，*Done* 变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

- **功能说明**

此指令用于将更改的凸轮点生效。首先使用 DMC_CamWritePoint 指令将电子凸轮表中对应的凸轮点信息写入，然后执行 DMC_CamSet 指令，更改后凸轮点信息才生效。

先执行 DMC_CamIn 指令，再执行 DMC_CamSet 指令，更改后的凸轮曲线下个周期生效。

先执行 DMC_CamSet 指令，再执行 DMC_CamIn 指令，更改后的凸轮曲线立即生效。

⚠ **注意**

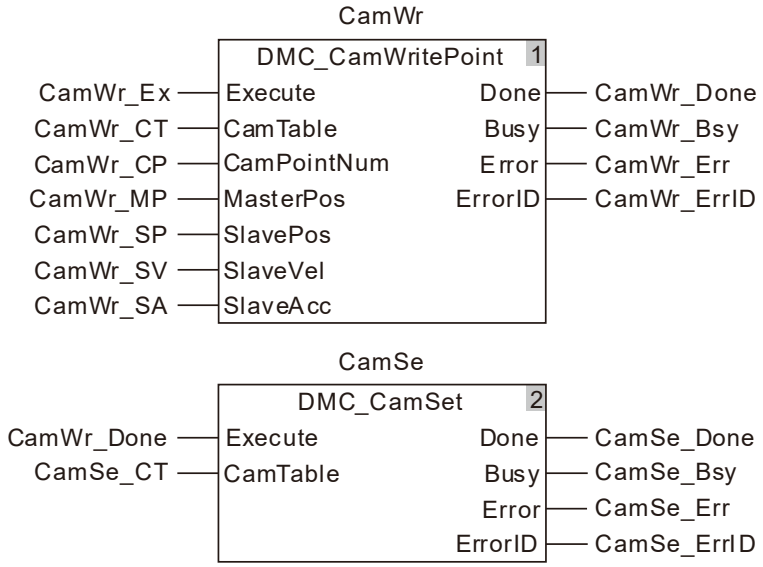
若要使用DMC_CamSet指令更改凸轮曲线，请务必确保该凸轮曲线没有被两个或者两个以上MC_CamIn指令调用，如果被多个 MC_CamIn 指令调用，使用 DMC_CamSet 指令更改凸轮曲线后，不能确保凸轮曲线更改时刻。

程序范例

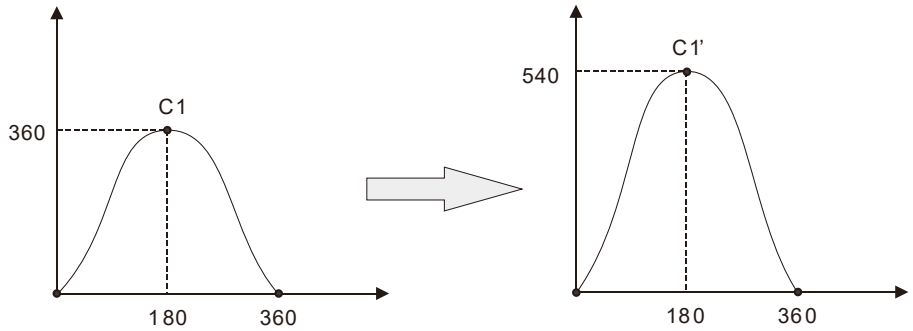
1. 变量和程序

变量名	数据类型	初始值
CamWr	DMC_CamWritePoint	
CamWr_Ex	BOOL	FALSE
CamWr_CT	USINT	1
CamWr_CP	UINT	2
CamWr_MP	LREAL	180.0
CamWr_SP	LREAL	540.0
CamWr_SV	LREAL	0.0
CamWr_SA	LREAL	0.0
CamWr_Done	BOOL	
CamWr_Bsy	BOOL	
CamWr_Err	BOOL	
CamWr_ErrID	WORD	
CamSe	DMC_CamSet	
CamSe_CT	USINT	1

变量名	数据类型	初始值
CamSe_Done	BOOL	
CamSe_Bsy	BOOL	
CamSe_Err	BOOL	
CamSe_ErrID	WORD	



2. 凸轮曲线



如上图所示，凸轮曲线中有三个凸轮点，将曲线 C1 更改为曲线 C1′;

❖ 更改前凸轮点信息如下表：

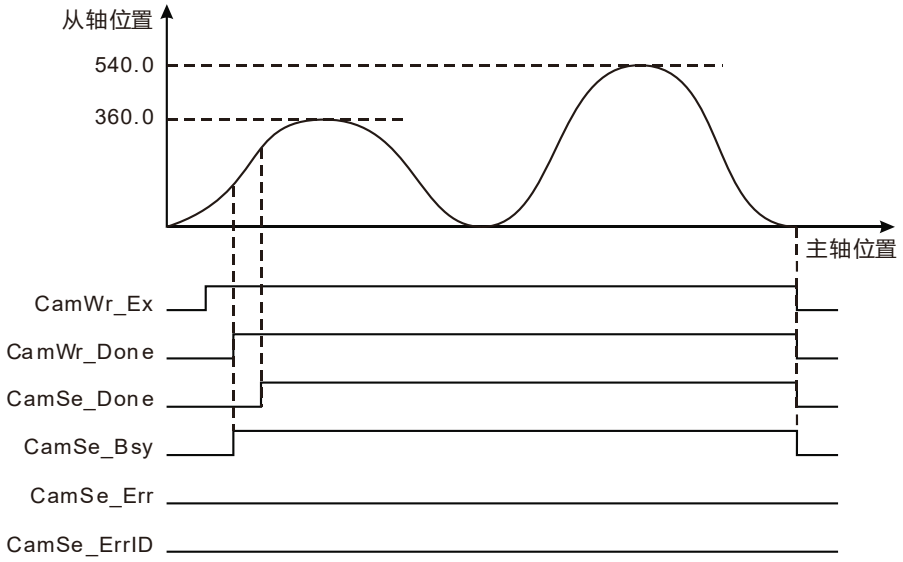
编号	主轴	从轴	速度	加速度
1	0.0	0.0	0.0	0.0
2	180.0	360.0	0.0	0.0
3	360.0	0.0	0.0	0.0

由曲线可以看出修改了第 2 个凸轮点，如上程序，执行 DMC_CamWritePoint 指令，再执行 DMC_CamSet 指令，可以更改凸轮曲线。

❖ 更改后凸轮点信息如下表：

编号	主轴	从轴	速度	加速度
1	0.0	0.0	0.0	0.0
2	180.0	540.0	0.0	0.0
3	360.0	0.0	0.0	0.0

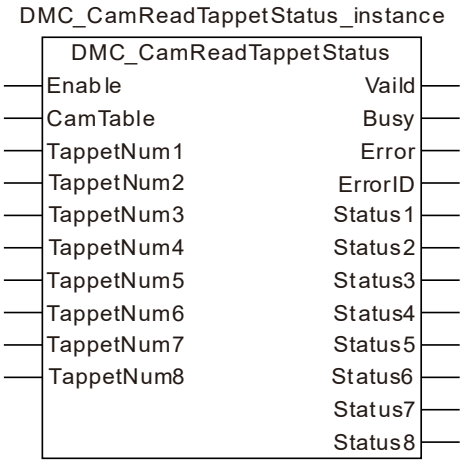
3. 时序图:



- ❖ CamWr_Ex 由 FALSE 变为 TRUE 则执行 DMC_CamWritePoint 指令，当 CamWr_Done 为 TRUE 时，更改的凸轮点写完成；
- ❖ CamWr_Done 由 FALSE 变为 TRUE，则执行 DMC_CamSet 指令，当 CamSe_Done 为 TRUE 时，更改凸轮点完成，等到当前凸轮周期结束后，下个周期更改后的凸轮曲线生效。

11.4.10 DMC_CamReadTappetStatus（读取多个挺杆点状态指令）

FB/FC	说明	适用机种
FB	此指令用于读取多个挺杆点状态。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当 *Enable* 由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	*Enable* 为 TRUE 时
CamTable（电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64（不可缺省）	*Enable* 为 TRUE 时
TappetNum1（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum2（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum3（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum4（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum5（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum6（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum7（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时
TappetNum8（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时

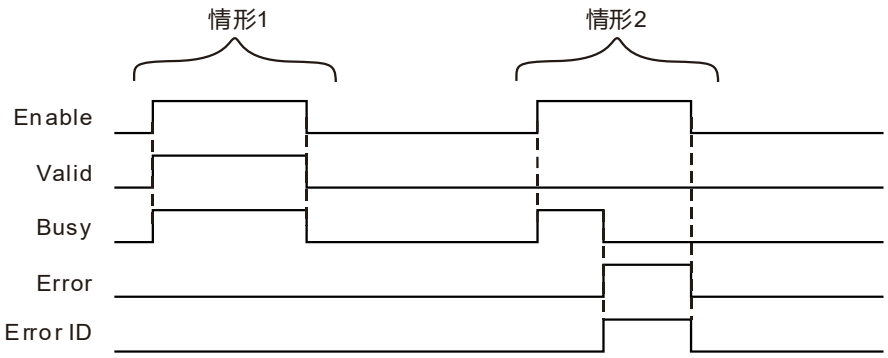
- **输出参数**

名称	功能	数据类型	输出范围
Valid （输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
Status1 （挺杆点 1 状态）	TappetNum1 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status2 （挺杆点 2 状态）	TappetNum2 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status3 （挺杆点 3 状态）	TappetNum3 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status4 （挺杆点 4 状态）	TappetNum4 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status5 （挺杆点 5 状态）	TappetNum5 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status6 （挺杆点 6 状态）	TappetNum6 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status7 （挺杆点 7 状态）	TappetNum7 指定挺杆点编号的状态	BOOL	TRUE / FALSE
Status8 （挺杆点 8 状态）	TappetNum8 指定挺杆点编号的状态	BOOL	TRUE / FALSE

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Valid	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ Enable 位为 FALSE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enalbe 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 和 Busy 同时变为 TRUE。当 Enable 变为 FALSE 时，Valid、Busy 均变为 FALSE。

情形2： 当有错误产生时， Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 的值会被清除。

- **功能说明**

此指令用于读取八个挺杆点状态。每个挺杆点状态为主轴正向或者反向经过挺杆点时的状态，每个挺杆点的状态由每个挺杆点的设置决定。每个挺杆点的状态在主轴正向经过凸轮终点或者反向经过凸轮起点时变为 FALSE。

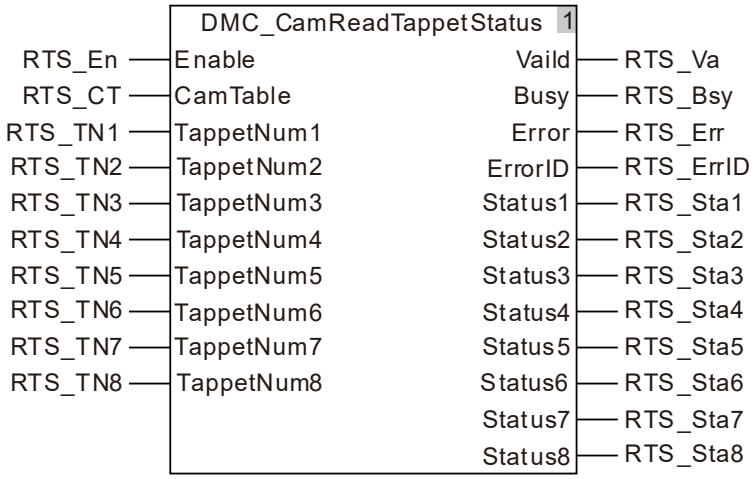
程序范例

1. 变量和程序

变量名	数据类型	初始值
RTS	DMC_CamReadTappetStatus	
RTS_En	BOOL	FALSE
RTS_CT	USINT	1
RTS_TN1	UINT	1
RTS_TN2	UINT	2
RTS_TN3	UINT	3
RTS_TN4	UINT	4
RTS_TN5	UINT	5
RTS_TN6	UINT	6
RTS_TN7	UINT	7
RTS_TN8	UINT	8
RTS_Va	BOOL	
RTS_Bsy	BOOL	
RTS_Err	BOOL	
RTS_ErrID	WORD	
RTS_Sta1	BOOL	
RTS_Sta2	BOOL	
RTS_Sta3	BOOL	
RTS_Sta4	BOOL	
RTS_Sta5	BOOL	

变量名	数据类型	初始值
RTS_Sta6	BOOL	
RTS_Sta7	BOOL	
RTS_Sta8	BOOL	

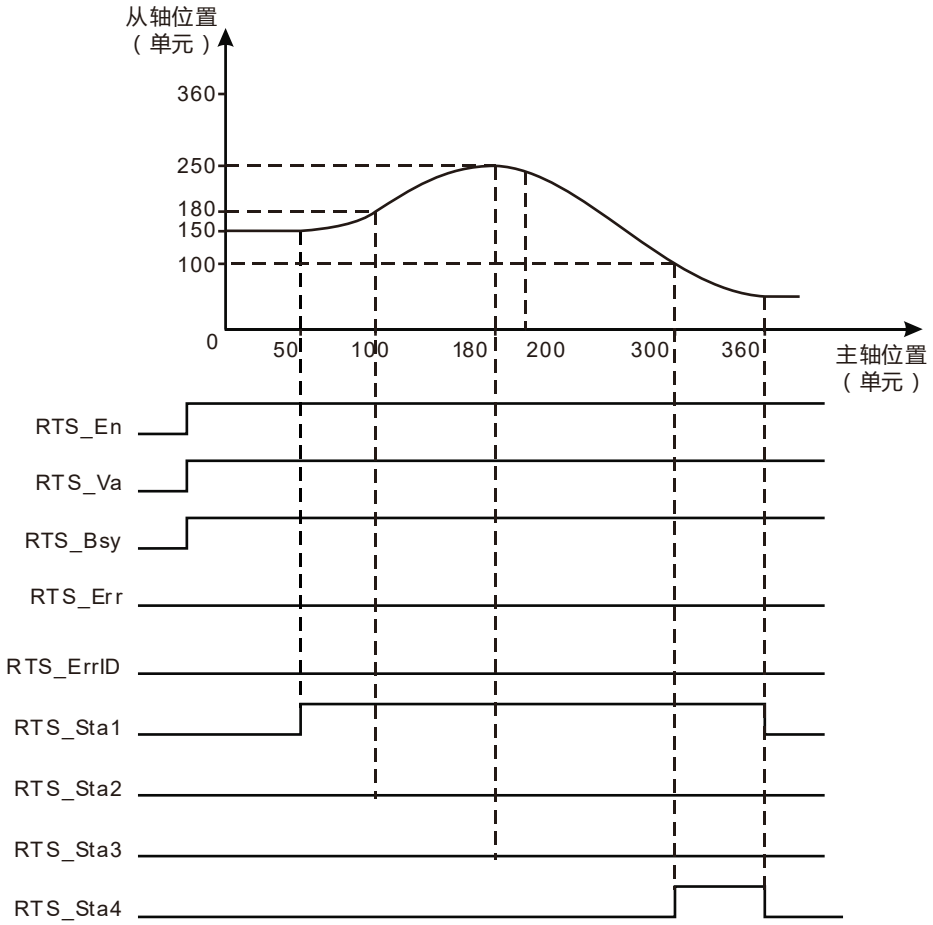
RTS



2. 挺杆点表

3. 索引	4. 主轴	5. 从轴	6. 正向经过	7. 反向经过
8. 1	9. 50.0	10. 150.0	11. 置位	12. 复位
13. 2	14. 100.0	15. 180.0	16. 关闭	17. 复位
18. 3	19. 180.0	20. 250.0	21. 复位	22. 复位
23. 4	24. 300.0	25. 100.0	26. 取反	27. 复位

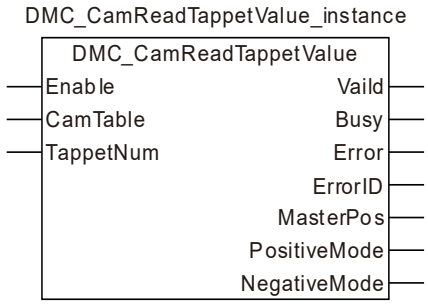
3. 位置曲线和时序图



- ❖ 在已建立的凸轮曲线中加入如上表所示的挺杆点，挺杆点 1 正向经过选择置位，挺杆点 2 正向经过选择关闭，挺杆点 3 正向经过选择复位，挺杆点 4 正向经过选择取反。
- ❖ 轴在正转的情况下，当 RTS_En 由 FALSE 变为 TRUE 时，执行 DMC_CamReadTappetStaus 指令，轴经过凸轮曲线中第 1 个挺杆点时，RTS_Sta1 由 FALSE 变为 TRUE（第一个挺杆点选择的是置位），经过第 2 个挺杆点时 RTS_Sta2 为 FALSE（第二个挺杆点选择的是状态不变），经过第 3 个挺杆点时 RTS_Sta3 为 FALSE（第三个挺杆点选择的是复位），经过第 4 个挺杆点时 RTS_Sta4 由 FALSE 变为 TRUE（第四个挺杆点选择的是反向）。
- ❖ 当主轴正向经过凸轮主轴终点时，RTS_Sta1 和 RTS_Sta4 变为 FALSE。

11.4.11 DMC_CamReadTappetValue（读取单个挺杆点信息指令）

FB/FC	说明	适用机种
FB	此指令用于读取单个挺杆点信息。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Enable（执行位）	当 *Enable* 由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	*Enable* 为 TRUE 时
CamTable（电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64（不可缺省）	*Enable* 为 TRUE 时
TappetNum（挺杆点编号）	读取挺杆点编号	UINT	1~128（不可缺省）	*Enable* 为 TRUE 时

- **输出参数**

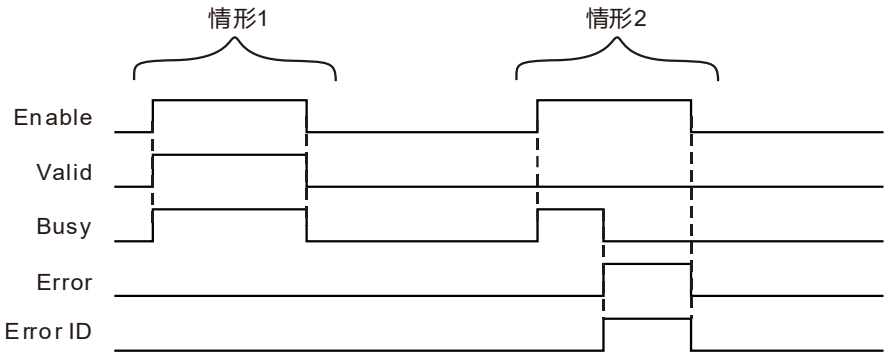
名称	功能	数据类型	输出范围
Valid（输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
MasterPos（主轴位置）	显示主轴位置	LREAL	-
PositiveMode（正向经过模式）	轴正转经过时挺杆点选择的模式。	PositiveMode_Type	0：PositiveDisable 1：PositiveOn 2：PositiveOff 3：PositiveInvert
NegativeMode（反向经过模式）	轴反转经过时挺杆点选择的模式。	NegativeMode_Type	0：NegativeDisable 1：NegativeOn

名称	功能	数据类型	输出范围
			2：NegativeOff 3：NegativeInvert

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ Enable 位为 FALSE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enalbe 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Valid 和 Busy 同时变为 TRUE。当 Enable 变为 FALSE 时，Valid、Busy 均变为 FALSE。

情形2： 当有错误产生时， Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 的值会被清除。

- **功能说明**

挺杆点信息包括该挺杆点的主轴位置、正向经过模式和反向经过模式。当轴正转时挺杆点可以选择的模式有 PositiveDisable、PositiveOn、PositiveOff 或 PositiveInvert；当轴反转时挺杆点可以选择的模式有 NegativeDisable、NegativeOn、NegativeOff 或 NegativeInvert。各个模式代表的含义如下表所示：

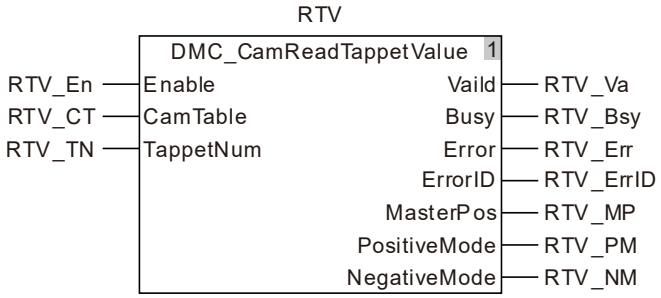
模式	功能	含义
PositiveDisable	关闭	主轴正向经过该点时，读取到该挺杆点时的状态无变化
PositiveOn	置位	主轴正向经过该点时，读取到该挺杆点时的状态为置位状态。
PositiveOff	复位	主轴正向经过该点时，读取到该挺杆点时的状态为复位状态。
PositiveInvert	取反	主轴正向经过该点时，正向经过该点前的状态为置位，则读取到该挺杆点时的状态为复位状态；正向经过该点前的状态为复位，则读取到该挺杆点时的状态为置位状态。
NegativeDisable	关闭	主轴反向经过该点时，读取到该挺杆点时的状态无变化
NegativeOn	置位	主轴反向经过该点时，读取到该挺杆点时的状态为置位状态。

模式	功能	含义
NegativeOff	复位	主轴反向经过该点时，读取到该挺杆点时的状态为复位状态。
NegativeIvert	取反	主轴反向经过该点时，反向经过该点前的状态为置位，则读取到该挺杆点时的状态为复位状态；反向经过该点前的状态为复位，则读取到该挺杆点时的状态为置位状态。

程序范例

1. 变量和程序

变量名	数据类型	初始值
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	2
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	



2. 挺杆点表

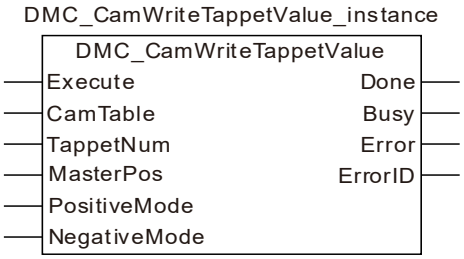
索引	主轴	从轴	正向经过	反向经过
1	108.0	235.0	PositiveOn（置位）	NegativeIvert（取反）
2	200.0	250.0	PositiveOff（复位）	PositiveOff（复位）
3	300.0	192.0	PositiveInvert（取反）	PositiveOn（置位）

如上所示挺杆点表，当 RTV_En 由 FALSE 变为 TRUE 时，执行 DMC_CamReadTappetValue 指令，当 RTV_Va 变为 TRUE 时，指令完成，读取到的挺杆点信息如下表。

索引	主轴	正向经过	反向经过
2	200.0	PositiveOff（复位）	NegativeOff（复位）

11.4.12 DMC_CamWriteTappetValue（写入挺杆点信息指令）

FB/FC	说明	适用机种
FB	此指令用于写入挺杆点信息。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
CamTable（电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64（不可缺省）	Execute 由 FALSE 变为 TRUE
TappetNum（挺杆点编号）	设定挺杆点编号	UINT	1~128（不可缺省）	Execute 由 FALSE 变为 TRUE
MasterPos（主轴位置）	设定挺杆点主轴位置。	LREAL	0、正数	Execute 由 FALSE 变为 TRUE
PositiveMode（正向经过模式）	轴正转时设定挺杆点模式。 0：关闭 1：置位 2：复位 3：取反	PositiveMode_Type	0：PositiveDisable 1：PositiveOn 2：PositiveOff 3：PositiveInvert	Execute 由 FALSE 变为 TRUE
NegativeMode（反向经过模式）	轴反转时设定挺杆点模式。 0：关闭 1：置位 2：复位 3：取反	NegativeMode_Type	0：NegativeDisable 1：NegativeOn 2：NegativeOff 3：NegativeInvert	Execute 由 FALSE 变为 TRUE

- **输出参数**

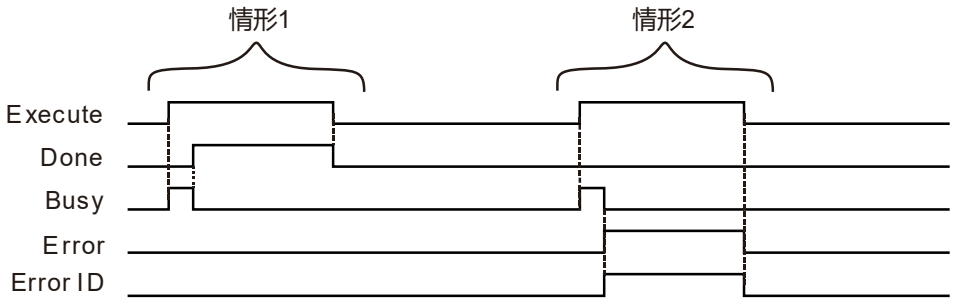
名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ *Execute* 位为 FALSE 时
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enalbe 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Eexcute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE；当指令执行完成 *Done* 变为 TRUE，同时 *Busy* 变为 FALSE；当 *Execute* 变为 FALSE 时，*Done* 变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

- **功能说明**

此指令用于写挺杆点信息。挺杆点信息包括该挺杆点的主轴位置、正向经过模式和反向经过模式。挺杆点一般在软件建立的凸轮中设置，如果用户需要动态更改凸轮点的设置，可以使用此指令更改。设置挺杆点后，可以使用“读取单个挺杆点状态”或者“读取多个挺杆点状态” 指令读取主轴经过该挺杆点时的状态。

当轴正转时挺杆点可以选择的模式有 PositiveDisable、PositiveOn、PositiveOff 或 PositiveIvert；当轴反转时挺杆点可以选择的模式有 NegativeDisable、NegativeOn、NegativeOff 或 NegativeIvert。各个模式代表的含义如下表所示：

模式	功能	含义
PositiveDisable	关闭	主轴正向经过该点时，读取到该挺杆点时的状态无变化
PositiveOn	置位	主轴正向经过该点时，读取到该挺杆点时的状态为置位状态。
PositiveOff	复位	主轴正向经过该点时，读取到该挺杆点时的状态为复位状态。
PositiveInvert	取反	主轴正向经过该点时，正向经过该点前的状态为置位，则读取到该挺杆点时

模式	功能	含义
		的状态为复位状态；正向经过该点前的状态为复位，则读取到该挺杆点时的状态为置位状态。
NegativeDisable	关闭	主轴反向经过该点时，读取到该挺杆点时的状态无变化
NegativeOn	置位	主轴反向经过该点时，读取到该挺杆点时的状态为置位状态。
NegativeOff	复位	主轴反向经过该点时，读取到该挺杆点时的状态为复位状态。
NegativeIvert	取反	主轴反向经过该点时，反向经过该点前的状态为置位，则读取到该挺杆点时的状态为复位状态；反向经过该点前的状态为复位，则读取到该挺杆点时的状态为置位状态。

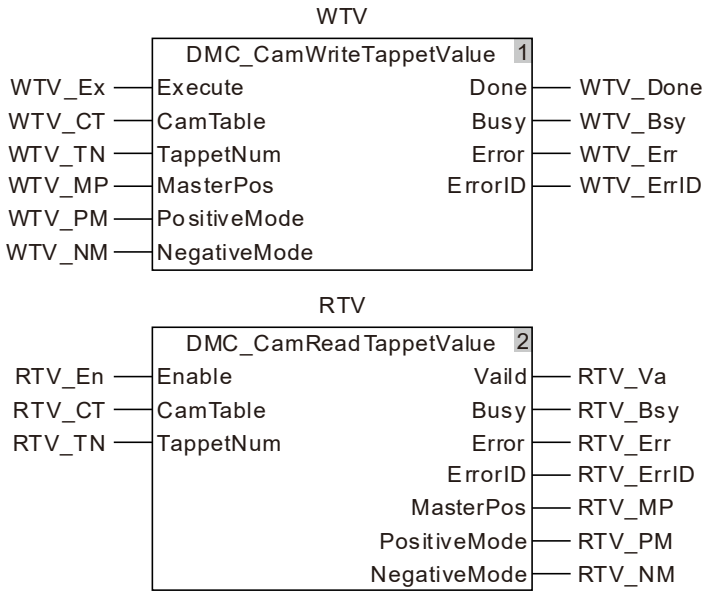
⚠ 注意

若要使用 DMC_CamWriteTappetValue 指令写凸轮曲线上的挺杆点，请务必确保该凸轮曲线没有被两个或两个以上 MC_CamIn 指令调用，如果被多个 MC_CamIn 指令调用，使用 DMC_CamWriteTappetValue 指令更改某个挺杆点，并且程序中有使用将要更改的挺杆点，则被用到的该挺杆点都被更改。

程序范例

1. 变量和程序

变量名	数据类型	初始值
WTV	DMC_CamWriteTappetValue	
WTV_Ex	BOOL	FALSE
WTV_CT	USINT	1
WTV_TN	UINT	2
WTV_MP	LREAL	200.0
WTV_PM	LREAL	PositiveOff
WTV_NM	LREAL	NegativeOff
WTV_Done	BOOL	
WTV_Bsy	BOOL	
WTV_Err	BOOL	
WTV_ErrID	WORD	
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	2
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	



2. 挺杆点表

索引	主轴	从轴	正向经过	反向经过
1	108.0	235.0	置位	取反
2	200.0	250.0	复位	复位
3	300.0	192.0	取反	置位

❖ 当 WTV_Ex 由 FALSE 变为 TRUE 时，执行 DMC_CamWriteTappetValue 指令，当 WTV_Done 为 TRUE 时，该指令执行完成，写入第二个挺杆点信息如下表：

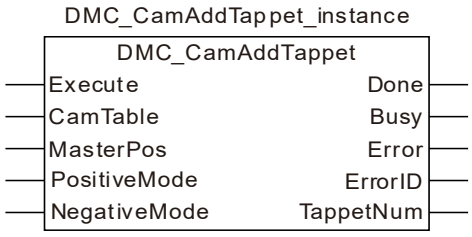
索引	主轴	正向经过	反向经过
2	250.0	PositiveOff（复位）	NegativeOff（复位）

❖ 当 RTV_En 由 FALSE 变为 TRUE 时，执行 DMC_CamReadTappetValue 指令，当 RTV_Done 为 TRUE 时，该指令执行完成，读取到的挺杆点信息如下表。

索引	主轴	正向经过	反向经过
2	250.0	PositiveOff（复位）	NegativeOff（复位）

11.4.13 DMC_CamAddTappet（增加挺杆点指令）

FB/FC	说明	适用机种
FB	此指令用于增加挺杆点。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
CamTable（电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64（不可缺省）	Execute 由 FALSE 变为 TRUE
MasterPos（主轴位置）	设定挺杆点主轴位置。	LREAL	0、正数	Execute 由 FALSE 变为 TRUE
PositiveMode（正向经过）	轴正转时设定挺杆点模式。 0：关闭 1：置位 2：复位 3：取反	PositiveMode_Type	0：PositiveDisable 1：PositiveOn 2：PositiveOff 3：PositiveInvert	Execute 由 FALSE 变为 TRUE
NegativeMode（反向经过）	轴反转时设定挺杆点模式。 0：关闭 1：置位 2：复位 3：取反	NegativeMode_Type	0：NegativeDisable 1：NegativeOn 2：NegativeOff 3：NegativeInvert	Execute 由 FALSE 变为 TRUE

- **输出参数**

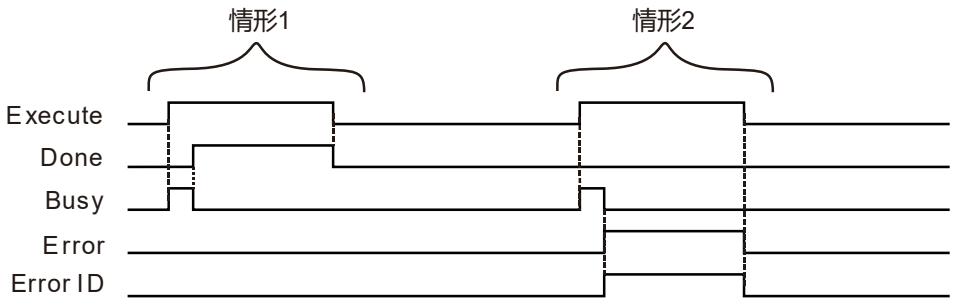
名称	功能	数据类型	输出范围
Done（完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
TappetNum （挺杆点编号）	增加的挺杆点编号	UINT	1~128

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ *Execute* 位为 FALSE 时
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enalbe 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Eexcute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE；当指令执行完成 *Done* 变为 TRUE 时，同时 *Busy* 变为 FALSE；当 *Execute* 变为 FALSE 时，*Done* 变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

- **功能说明**

此指令用于增加挺杆点信息。挺杆点信息包括该挺杆点的编号、主轴位置、正向经过模式和反向经过模式。挺杆点一般在软件建立的凸轮中设置。增加的挺杆点编号按照已有挺杆点编号依次增加，如原来挺杆点最大编号为 3，执行该指令后，挺杆点最大编号变为 4。

当轴正转时挺杆点可以选择的模式有 PositiveDisable、PositiveOn、PositiveOff 或 PositiveIvert；当轴反转时挺杆点可以选择的模式有 NegativeDisable、NegativeOn、NegativeOff 或 NegativeIvert。各个模式代表的含义如下表所示：

模式	功能	含义
PositiveDisable	关闭	主轴正向经过该点时，读取到该挺杆点时的状态无变化
PositiveOn	置位	主轴正向经过该点时，读取到该挺杆点时的状态为置位状态。
PositiveOff	复位	主轴正向经过该点时，读取到该挺杆点时的状态为复位状态。
PositiveInvert	取反	主轴正向经过该点时，正向经过该点前的状态为置位，则读取到该挺杆点时的状态为复位状态；正向经过该点前的状态为复位，则读取到该挺杆点时的状态为置位状态。

模式	功能	含义
NegativeDisable	关闭	主轴反向经过该点时，读取到该挺杆点时的状态无变化
NegativeOn	置位	主轴反向经过该点时，读取到该挺杆点时的状态为置位状态。
NegativeOff	复位	主轴反向经过该点时，读取到该挺杆点时的状态为复位状态。
NegativeIvert	取反	主轴反向经过该点时，反向经过该点前的状态为置位，则读取到该挺杆点时的状态为复位状态；反向经过该点前的状态为复位，则读取到该挺杆点时的状态为置位状态。

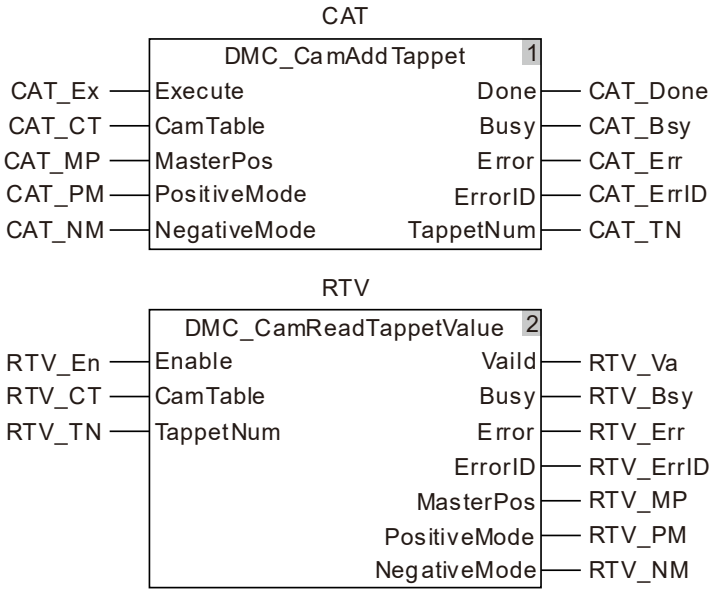
注意

若要使用 DMC_CamAddTappet 指令增加凸轮曲线上的挺杆点，请务必确保该凸轮曲线没有被两个或两个以上 MC_CamIn 指令调用，如果被多个 MC_CamIn 指令调用，使用 DMC_CamAddTappet 指令增加挺杆点，并且程序中有使用将要增加的挺杆点，则增加的该挺杆点都可以被使用。

程序范例

1. 变量和程序

变量名	数据类型	初始值
CAT	DMC_CamAddTappet	
CAT_Ex	BOOL	FALSE
CAT_CT	USINT	1
CAT_MP	LREAL	200.0
CAT_PM	LREAL	PositiveOff
CAT_NM	LREAL	NegativeOff
CAT_Done	BOOL	
CAT_Bsy	BOOL	
CAT_Err	BOOL	
CAT_ErrID	WORD	
CAT_TN	UINT	
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	4
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	



2. 挺杆点表

编号	主轴	从轴	正向经过	反向经过
1	108.0	235.0	置位	取反
2	200.0	250.0	复位	复位
3	300.0	192.0	取反	置位

在建立的凸轮曲线中已增加如上表所示的三个挺杆点，现使用 DMC_CamAddTappet 指令增加第 4 个挺杆点。

❖ 当 CAT_Ex 由 FALSE 变为 TRUE 时，执行 DMC_CamAddTappet 指令，当 CAT_Done 为 TRUE 时，该指令执行完成，增加第 4 个挺杆点信息如下表：

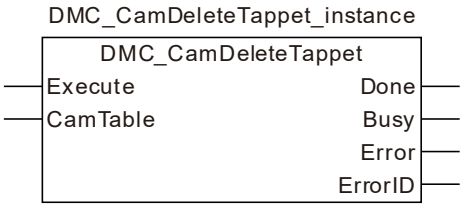
编号	主轴	正向经过	反向经过
4	250.0	PositiveOff（复位）	NegativeOff（复位）

❖ 当 RTV_En 由 FALSE 变为 TRUE 时，执行 DMC_CamReadTappetValue 指令，当 RTV_Done 为 TRUE 时，该指令执行完成，读取到的挺杆点信息如下表。

编号	主轴	正向经过	反向经过
4	250.0	PositiveOff（复位）	NegativeOff（复位）

11.4.14 DMC_CamDeleteTappet（删除挺杆点指令）

FB/FC	说明	适用机种
FB	此指令用于删除挺杆点。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
CamTable （电子凸轮表编号）	用于设定主从轴建立凸轮关系所依据的凸轮表。	USINT	1~64 （不可缺省）	Execute 由 FALSE 变为 TRUE

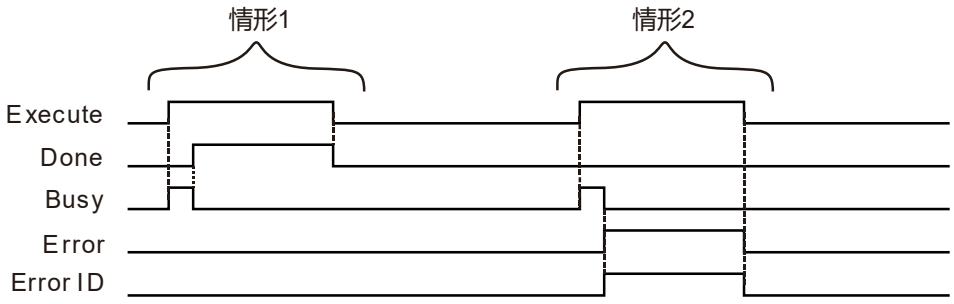
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成位）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当指令执行完成时	◆ 当 Error 为 TRUE 时 ◆ *Execute* 为 FALSE 时
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enalbe 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Eexcute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE；当指令执行完成 *Done* 变为 TRUE 时，同时 *Busy* 变为 FALSE；当 *Execute* 变为 FALSE 时，*Done* 变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，当 *Eexcute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

- **功能说明**

此指令用于删除挺杆点，删除的挺杆点是编号最大的挺杆点。每执行一次该指令，则删除一个编号最大的挺杆点。

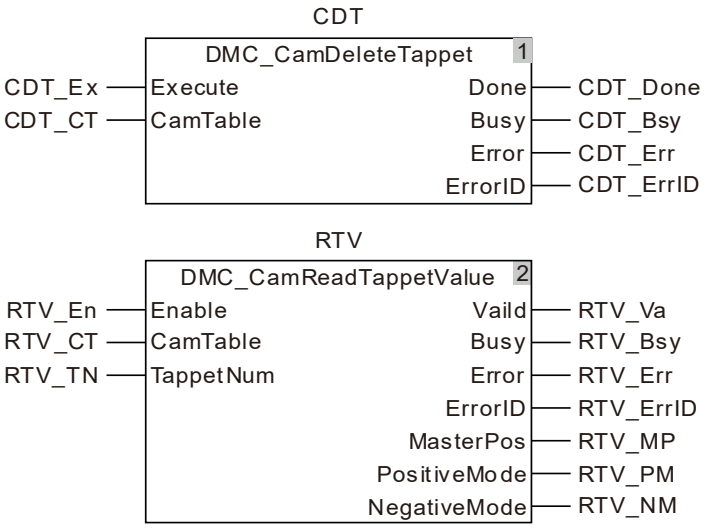
⚠ 注意

若要使用 DMC_CamDeleteTappet 指令删除凸轮曲线上的挺杆点，请务必确保该凸轮曲线没有被两个或两个以上 MC_CamIn 指令调用，如果被多个 MC_CamIn 指令调用，使用 DMC_CamDeleteTappet 指令删除某个挺杆点，并且程序中有使用将要删除的挺杆点，则被用到的该挺杆点都被删除。

程序范例

1. 变量和程序

变量名	数据类型	初始值
CDT	DMC_CamDeletTappet	
CDT_Ex	BOOL	FALSE
CDT_CT	USINT	1
CDT_Done	BOOL	
CDT_Bsy	BOOL	
CDT_Err	BOOL	
CDT_ErrID	WORD	
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	3
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	



2. 挺杆点表

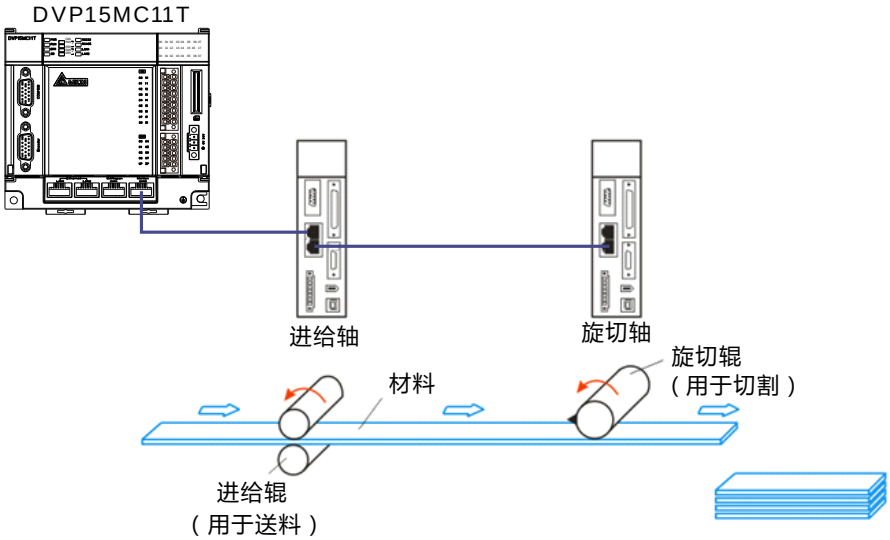
索引	主轴	从轴	正向经过	反向经过
1	108.0	235.0	置位	取反
2	200.0	250.0	复位	复位
3	300.0	192.0	取反	置位

- ❖ 当 CDT_Ex 由 FALSE 变为 TRUE 时 执行 DMC_CamDeletTappet 指令 当 CDT_Done 为 TRUE 时，该指令执行完成，删除最后一个挺杆点。
- ❖ 当 RTV_En 由 FALSE 变为 TRUE 时，执行 DMC_CamReadTappetValue 指令， 因为第三个挺杆点不存在，所以指令会报错 16#5505（该挺杆点不存在）。

11.5 应用指令

11.5.1 旋切功能工艺介绍

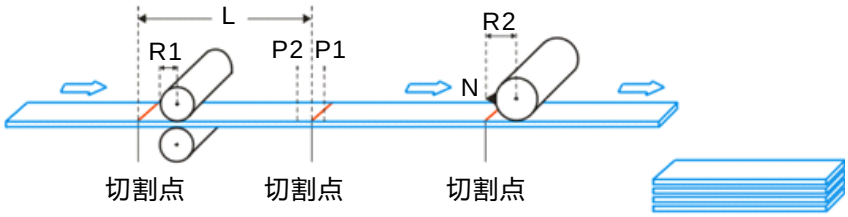
旋切是指对材料在传输过程中垂直方向对其进行切割的工艺。随着旋切轴的旋转，刀头将对切割面进行周期性切割。



注：进给轴用于控制进给辊，旋切轴用于控制旋切辊，而刀头安装于旋切辊。

旋切功能一般用于薄料或中等厚度的材料切割。可将其应用于包装机、旋切机、印花机、冲孔机等。

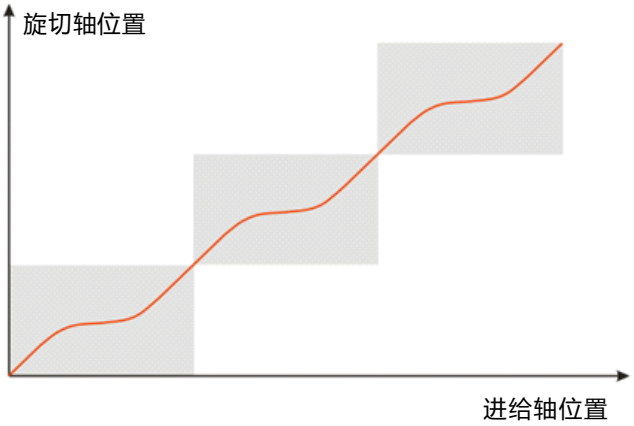
11.5.2 旋切功能工艺参数



图中参数	实际含义	指令中的名称
L	材料的切割长度。	APF_RotaryCut_Init.CutLength
R1	进给轴半径，即进给辊半径长度。	APF_RotaryCut_Init.FeedRadius
R2	旋切轴半径，即旋切辊圆心到刀尖的距离。	APF_RotaryCut_Init.RotaryRadius
N	旋切轴的刀头数，图中的刀头数为 1。	APF_RotaryCut_Init.KnifeNum
P1	同步区开始位置。	APF_RotaryCut_Init.SyncStartPos
P2	同步区结束位置。	APF_RotaryCut_Init.SyncStopPos

11.5.3 旋切功能控制特性

旋切功能是一种特殊的电子凸轮功能。连续切割时，其旋切曲线示意图如下：



- **功能特性**

1. 用户可根据工艺要求自由设置切割长度，切割长度可以小于或大于切刀周长。
2. 在同步区内，进给轴与旋切轴按一定的速度比例运转（速度通常相等），并且材料的切割发生在同步区内。
3. 旋切功能支持多刀头的旋切辊。
4. 旋切功能启动后，旋切轴跟随进给轴的相位动作，因此进给轴可以作匀速、加速、减速运动、不规则运动。
5. 旋切功能结束后，旋切刀停止在系统零点，即进入点。

11.5.4 旋切功能凸轮介绍

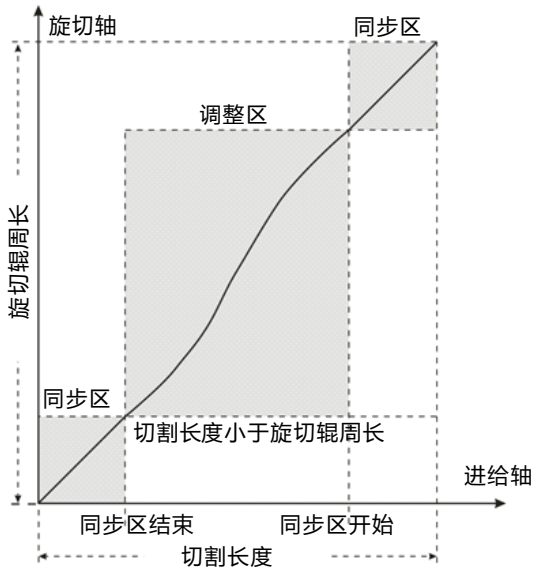
旋切曲线分为同步区与调整区。

同步区：此时进给轴与旋切轴按固定的速度比例运转（刀头的线速度与切割面的线速度通常相等），并且材料的切割发生在同步区内。

调整区：由于切割长度不同，需要做相应的位移调整。根据切割长度调整区可以分为下面三种情况。

1. 短料切割

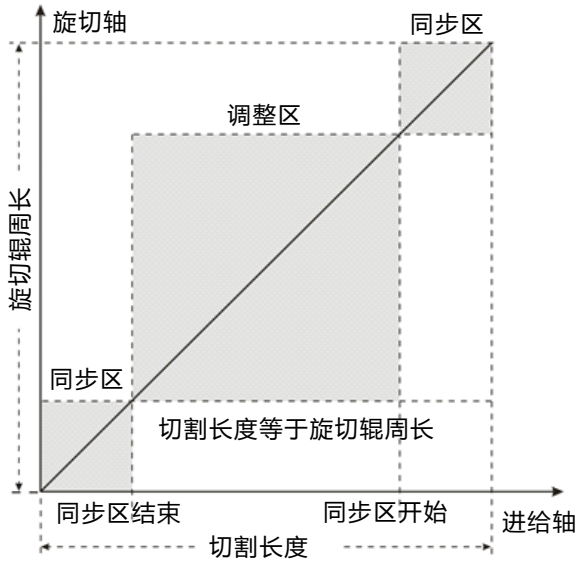
切割长度小于旋切辊周长时，任一周期的旋切曲线如下：



短料切割时，旋切轴必须在调整区内先加速，然后再减速到同步速度。

2. 等长切割

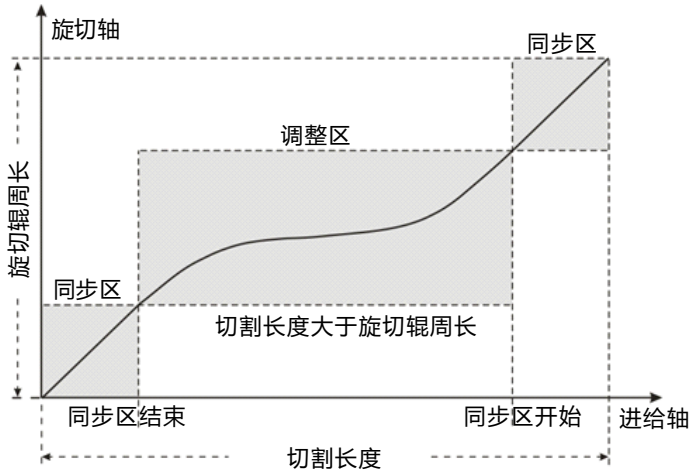
旋切长度等于刀辊周长时，任一周期的旋切曲线如下：



此情况下，同步区与非同步区进给轴和旋切轴一直保持速度同步，旋切轴不需要调整。

3. 长料切割

旋切长度大于刀辊周长时，任一周期的旋切曲线如下：

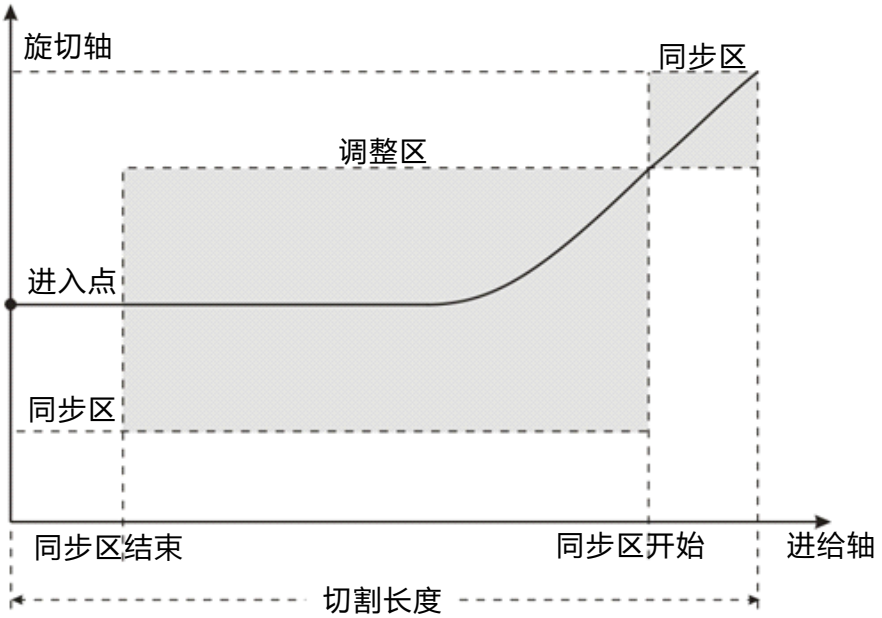


此情况下，旋切轴应该在调整区内先减速，然后再加速到同步速度。如果旋切长度远大于刀辊周长，则刀辊有可能减速到零，停留一段时间，然后再加速到同步速度。切割长度越长，停留的时间越长。

旋切功能在启动和脱离时，使用不同的旋切曲线：

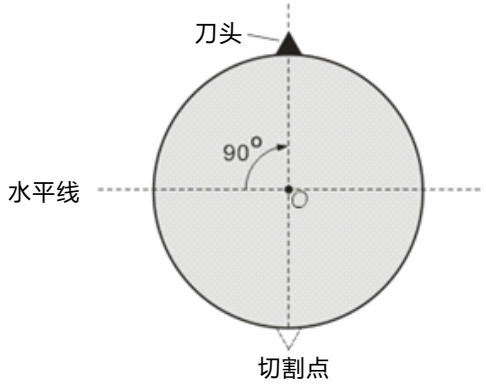
4. 进入曲线

旋切功能在启动时的旋切曲线

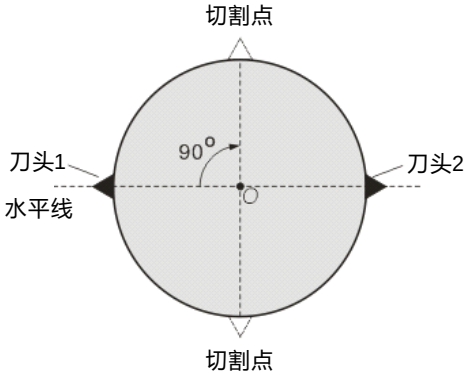


此曲线为旋切功能进入曲线。当旋切功能启动时，旋切轴通过此曲线跟随进给轴。

进入点是针对旋切轴而言的。单刀头时，若进入点在旋切辊的正上方，那么切割点在旋切辊的正下方，如下图所示。在旋切功能启动之前，须将刀头旋转至旋切辊的正上方。否则，很可能在调整区中进行切割。

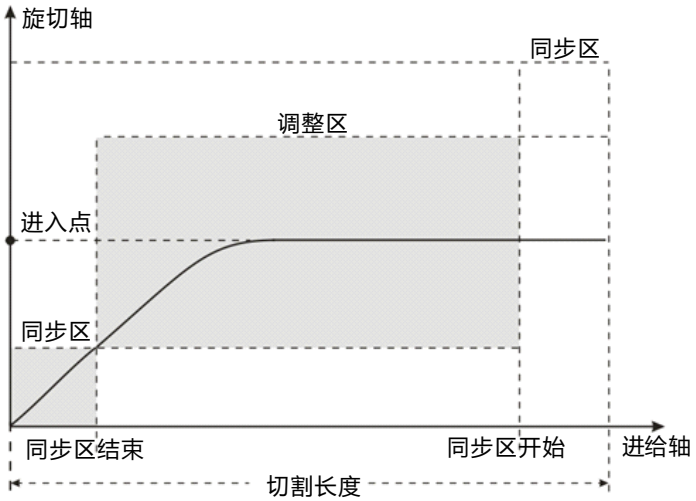


在旋切辊上安装了多个刀头时，刀头间距须相等，而切割点在刀头间距的最中央。下图为双刀头示意图。



5. 结束曲线

旋切功能在脱离时的旋切曲线



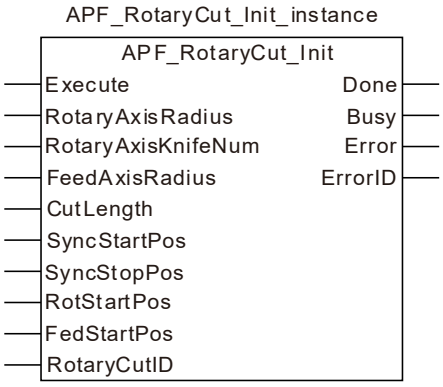
旋切脱离指令启动后，系统会借助这条曲线将旋切轴从旋切状态脱离。最终刀头将停止在结束点，如上图所示。

结束点也是针对旋切轴而言的。单刀头时，结束点与进入点是同一点，结束点也位于旋切辊的正上方。

11.5.5 旋切指令介绍

11.5.5.1 APF_RotaryCut_Init（旋切初始化指令）

FB/FC	说明	适用机种
FB	此指令用于初始化旋切轴半径、进给轴半径、切割长度、同步区等参数。	DVP15MC11T DVP15MC11T-06

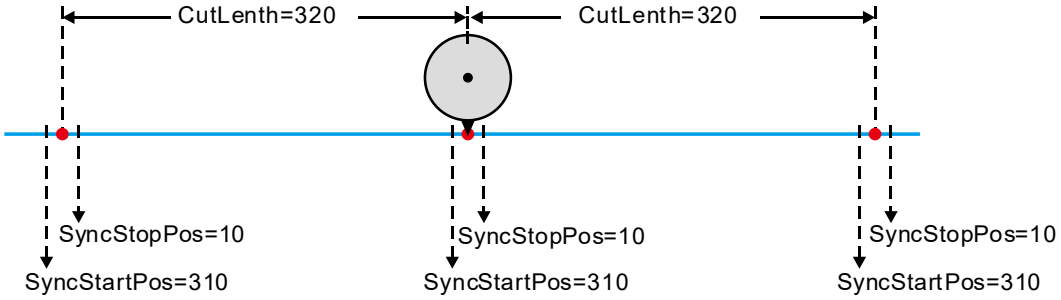


- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当*Execute*由FALSE变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	-
RotaryAxisRadius（旋切轴半径）	旋切轴半径，即旋切辊圆心到刀尖的距离。	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
RotaryAxisKnifeNum（旋切轴刀头数）	旋切轴的刀头数，即安装于旋切辊的刀头数量。	USINT	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
FeedAxisRadius（进给轴半径）	进给轴半径，即进给辊半径长度。	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
CutLength（切割长度）	材料的切割长度。	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
SyncStartPos（同步区开始位置）	同步区开始位置，即同步区开始时对应的进给轴位置。	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
SyncStopPos（同步区结束位置）	同步区结束位置，即同步区结束时对应的进给轴位置。	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
RotStartPos	保留	-	-	-
FedStartPos	保留	-	-	-
RotaryCutID（旋切编号）	旋切指令组合编号，同组的旋切指令使用相同编号。设置范围：1~8。	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE

说明：

1. 同步区起始位置（SyncStartPos）总是大于同步区结束位置（SyncStopPos）。如下图所示，切长为 320，同步区起始位置为 310，同步区结束位置为 10。



2. 同步区有限制，它不能大于切割长度的一半。如上图所示，同步区为 20，切割长度的一半为 160。
3. 此功能中的长度参数有：旋切轴半径、进给轴半径、切割长度、同步区开始位置、同步区结束位置。这些参数的单位必须一致。换言之，若其中某个参数的单位为厘米，那么其它参数的单位必须都为厘米。

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当旋切初始化执行完成时.	◆ 该指令执行完成后，当 *Execute* 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 *Execute* 由 TRUE 变为 FALSE 时，在指令执行完成时，*Done* 变为 TRUE，且一个周期后，*Done* 变为 FALSE
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 *Done* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Execute* 由 TRUE 变为 FALSE 时

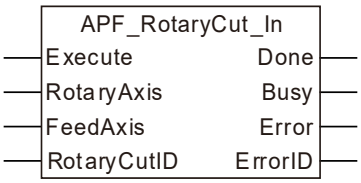
- **功能说明**

若旋切关系未建立，此指令用于初始化旋切轴半径、进给轴半径、切割长度、同步区等参数，指令执行成功后，相关参数被载入，以便在建立旋切关系时调用；若旋切关系已经建立，此指令用于修改旋切参数，当指令执行完毕后，新参数将在下个周期生效。

11.5.5.2 APF_RotaryCut_In（旋切耦合指令）

FB/FC	说明	适用机种
FB	此指令用于建立旋切关系，可根据应用需求指定旋切轴与进给轴的站号。	DVP15MC11T DVP15MC11T-06

APF_RotaryCut_In_instance



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
Execute （执行位）	当*Execute*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	-
RotaryAxis （旋切轴号）	旋切轴的站号	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
FeedAxis （进给轴号）	进给轴的站号。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
RotaryCutID （旋切编号）	旋切指令组合编号，同组的旋切指令使用相同编号。 设置范围：1~8。	USINT	1~8 （不可缺省）	*Execute* 由 FALSE 变为 TRUE

- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当原点回归执行完成时.	◆ 该指令执行完成后，当 *Execute* 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 *Execute* 由 TRUE 变为 FALSE 时，在指令执行完成时，*Done* 变为 TRUE，且一个周期后，*Done* 变为 FALSE
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Execute* 由 TRUE 变为 FALSE 时

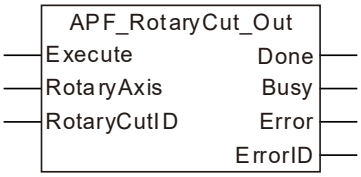
- **功能说明**

此指令用于建立旋切关系，可根据应用需求指定旋切轴与进给轴的站号。当指令执行成功后，旋切轴根据旋切曲线跟随进给轴运动。

11.5.5.3 APF_RotaryCut_Out（旋切脱离指令）

FB/FC	说明	适用机种
FB	此指令用于解除旋切轴与进给轴之间已经建立的旋切关系。	DVP15MC11T DVP15MC11T-06

APF_RotaryCut_Out_instance



- **输入参数**

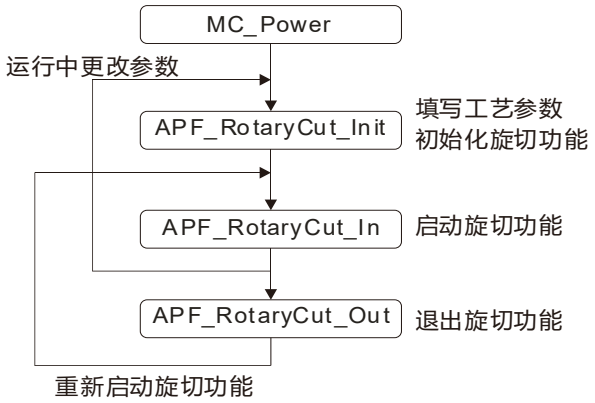
名称	功能	数据类型	设定范围 （缺省值）	生效时机
Execute （执行位）	当*Execute*由 FALSE 变 TRUE时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	-
RotaryAxis （旋切轴号）	旋切轴轴号	USINT	请参考第 2.2 节**功能简介** （不可缺省）	*Execute* 由 FALSE 变为 TRUE
RotaryCutID （旋切编号）	旋切指令组合编号，一个组的旋切参数使用统一编号。 设置范围：1~8	USINT	1~8 （不可缺省）	*Execute* 由 FALSE 变为 TRUE

- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

注：

1. 旋切功能控制顺序如下：



2. 旋切功能执行时，旋切轴只能执行 APF_RotaryCut_Out 与 MC_Stop 指令，其它指令无效。

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当旋切轴和给进轴已经耦合完成时.	◆ 该指令执行完成后，当 *Execute* 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 *Execute* 由 TRUE 变为 FALSE 时，在指令执行完成时，*Done* 变为 TRUE，且一个周期后，*Done* 变为 FALSE
Busy	◆ 当 *Execute* 为 TRUE 时	◆ 当 *Done* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Execute* 由 TRUE 变为 FALSE 时

- **功能说明**

此指令用于解除旋切轴与进给轴之间已经建立的旋切关系。旋切关系解除后，旋切轴的刀头会停在进入点，且旋切轴不再跟随进给轴。此指令不会影响进给轴的运动。

11.5.6 旋切指令应用范例

本节主要介绍旋切参数的设置、旋切关系的建立、旋切关系的脱离，范例程序如下。

范例关键参数：

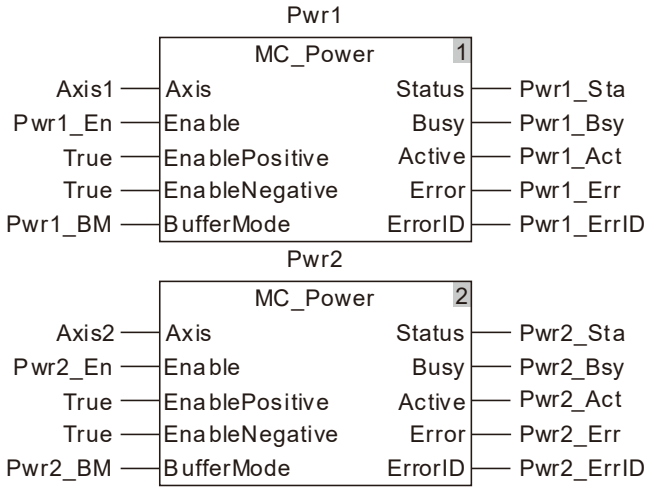
参数名称	当前数值
旋切轴的站号	2
进给轴的站号	1
旋切轴半径	10（单位：单元）
旋切轴刀头数	1
进给轴半径	20（单位：单元）
切割长度	30（单位：单元）
同步区开始位置	19（单位：单元）
同步区结束位置	1（单位：单元）

程序范例

1. 当 Pwr1_En 为 TRUE 时，站号为 1 的伺服会使能。当 Pwr2_En 为 TURE 时，站号为 2 的伺服会使能。

变量和程序

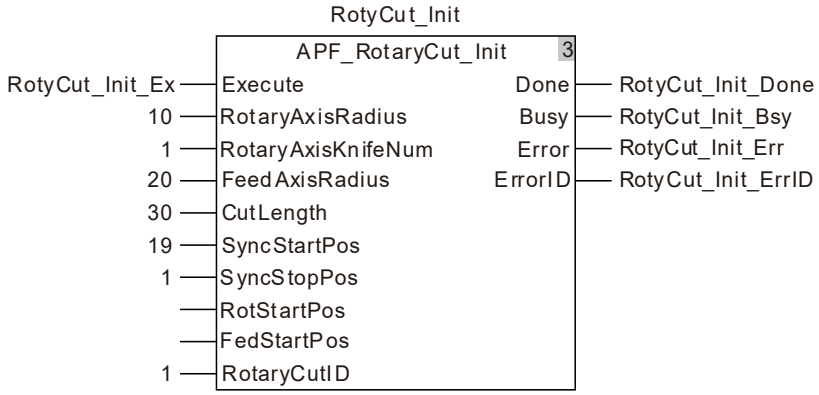
变量名	数据类型	初始值
Pwr1	MC_Power	
Axis1	USINT	1
Pwr1_En	BOOL	TRUE
Pwr1_BM	MC_Buffer_Mode	0
Pwr1_Sta	BOOL	TRUE
Pwr1_Bsy	BOOL	
Pwr1_Act	BOOL	
Pwr1_Err	BOOL	
Pwr1_ErrID	WORD	
Pwr2	MC_Power	
Axis2	USINT	1
Pwr2_En	BOOL	TRUE
Pwr2_BM	MC_Buffer_Mode	0
Pwr2_Sta	BOOL	TRUE
Pwr2_Bsy	BOOL	
Pwr2_Act	BOOL	
Pwr2_Err	BOOL	
Pwr2_ErrID	WORD	



2. 设置旋切工艺参数。旋切轴半径为 10，旋切轴刀头数为 1，进给轴半径为 20，进给轴的切割长度为 30，同步区的开始位置为 19，同步区的结束位置为 1，旋切组合编号为 1，当 RotyCut_Init_Ex 为 TRUE 时，旋切工艺参数将被初始化。

变量和程序

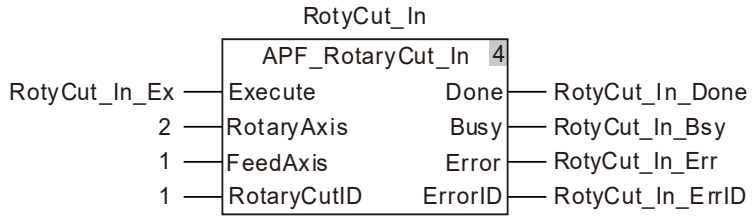
变量名	数据类型	初始值
RotyCut_Init	APF_RotaryCut_Init	
RotyCut_Init_Ex	BOOL	TRUE
RotyCut_Init _Done	BOOL	TRUE
RotyCut_Init _Bsy	BOOL	
RotyCut_Init _Err	BOOL	
RotyCut_Init _ErrID	WORD	



3. 当 RotyCut_In_Ex 为 TRUE 时，开始建立旋切关系。RotyCut_In _Done 为 TRUE，代表旋切关系建立成功。1 号伺服为进给轴，2 号伺服为旋切轴。

变量和程序

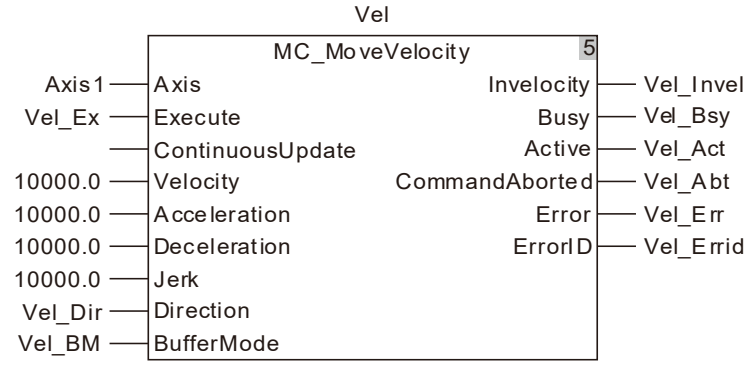
变量名	数据类型	初始值
RotyCut_In	APF_RotaryCut_In	
RotyCut_In_Ex	BOOL	TRUE
RotyCut_In _Done	BOOL	TRUE
RotyCut_In _Bsy	BOOL	
RotyCut_In _Err	BOOL	
RotyCut_In _ErrID	WORD	



4. 当 Vel _Ex 为 TRUE，进给轴开始执行速度指令。此时，旋切轴根据进给轴的相位来执行旋切动作。

变量和程序

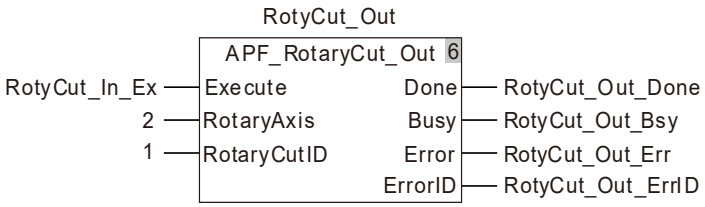
变量名	数据类型	初始值
Vel	MC_MoveVelocity	
Axis1	USINT	1
Vel _Ex	BOOL	TRUE
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	0
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ErrID	WORD	



5. 当 RotyCut_Out_Ex 为 TRUE 时，旋切轴开始脱离进给轴。RotyCut_Out_Done 为 TRUE，代表旋切轴脱离成功。旋切轴脱离进给轴后，将会回到进入点，进给轴的运动不再影响从轴。

变量和程序

变量名	数据类型	初始值
RotyCut_Out	APF_RotaryCut_Out	
RotyCut_Out_Ex	BOOL	TRUE
RotyCut_Out_Done	BOOL	TRUE
RotyCut_Out_Bsy	BOOL	
RotyCut_Out_Err	BOOL	
RotyCut_Out_ErrID	WORD	


RotyCut_Out
APF_RotaryCut_Out 6
RotyCut_In_Ex
Execute
Done
RotyCut_Out_Done
2
RotaryAxis
Busy
RotyCut_Out_Bsy
1
RotaryCutID
Error
RotyCut_Out_Err
ErrorID
RotyCut_Out_ErrID

11.6 G 代码指令

11.6.1 CNC 简介

DVP-15MC 系列运动控制器作为一个多轴运动控制器，支持标准的 CNC 功能，实现简易的数控机床，同时也可应用于其它一些通过 G 代码进行定位及路径规划的场合。

CANopen Builder 软件提供 CNC 的 G 代码编辑功能，用户可以在 CNC 编辑器中编写 G 代码，也可以将由其它模具设计软件转换成安全 G 代码导入此编辑器。当 G 代码输入上方代码列表后，软件下方的预览窗口将会输出 G 代码即将产生的三维图形。下载程序后，会将所有的 G 代码下载至控制器运行

G 代码编辑完成后，需要在运动控制程序中调用。使用 DMC_CartesianCoordinate 指令即可控制伺服轴进行位置插补。

11.6.2 G 代码输入格式

DVP-15MC 系列运动控制器支持的 G 代码及输入格式如下表所示：

支持代码	功能描述	支持轴数	格式
G0	快速定位	8 轴	格式 1： G0 X_ Y_ Z_ A_ B_ C_ P_ Q_
G1	直线插补	8 轴	格式 1： G1 X_ Y_ Z_ A_ B_ C_ P_ Q_ E_ F_
G2	顺时针圆弧插补 、螺旋插补	8 轴	格式 1 : G2 X_ Y_ Z_ A_ B_ C_ P_ Q_ I_ J_ (I_ K_ / J_ K_)T_ E_ F_ 格式 2 : G2 X_ Y_ Z_ A_ B_ C_ P_ Q_ R_ T_ E_ F_
G3	逆时针圆弧插补 、螺旋插补	8 轴	格式 1 : G3 X_ Y_ Z_ A_ B_ C_ P_ Q_ I_ J_ (I_ K_ / J_ K_)T_ E_ F_ 格式 2 : G3 X_ Y_ Z_ A_ B_ C_ P_ Q_ R_ T_ E_ F_
G4	时间延时	--	格式 1： G4 K_
G17	XY 平面选择	--	格式 1： G17
G18	ZX 平面选择	--	格式 1： G18
G19	YZ 平面选择	--	格式 1： G19
G90	绝对模式	--	格式 1： G90
G91	相对模式	--	格式 1： G91
G50	精确停止	--	格式 1： G50
G51	圆弧交接	--	格式 1： G51 D_
G52	圆滑交接	--	格式 1： G52

支持代码	功能描述	支持轴数	格式
M0~M99	M 代码	--	格式 1： M_ D_

注：格式中的下划线处为欲设定的参数数值。在 CANopen Builder 软件 CNC 程序中输入 G 代码时，G 代码前面要输入 N_， N_表示 G 代码在 NC 程序中所在行的编号，每行只能输入一条 G 代码。G 代码在 CANopen Builder 软件输入格式如下：N0 G0 X100 Y100

11.6.3 G 代码格式说明

- **G 代码的单位**

G 代码中 X_、Y_、Z_、A_、B_、C_、P_、Q_轴的位置单位与轴参数一致。请将轴的物理单位统一设定。例如将单位设置为毫米，则 G0 X100.5 Y300 Z30.6 表示 X、Y、Z 轴分别运行至 100.5 毫米、300 毫米、30.6 毫米处。

- **G 代码参数缺省**

1. G0 指令可缺省 X_、Y_、Z_、A_、B_、C_、P_、Q_中的一项或多项。
2. G1 指令可缺省 X_、Y_、Z_、A_、B_、C_、P_、Q_、E_、E_、F_中的一项或多项。
3. G2、G3 指令可缺省 X_、Y_、Z_、A_、B_、C_、P_、Q_、E_、E_、F_中的一项或多项，但 I_、J_、K_、R_不能缺省。
4. G4、G51 指令后的参数不能缺省。
5. M 代码可缺省 D_。
6. CANopen Builder 中的 CNC 编辑区内同一行只能书写一条 G 代码。

- **G 代码特殊功能**

  - **G 代码中可通过%ML 装置代表关键数值。**

如 X_、Y_、Z_、A_、B_、C_、P_、Q_、E_、F_、 I_、J_、K_、R_、T_、E_、F_ 皆可使用 %ML 寄存器。其中 T 为 ULINT，其余类型为 LREAL 型。书写时，需将“%ML_”的“%”去掉，改写为“ML_”再在前后加上“$”即可，具体见下面的范例。

范例：N00 G0 X$ML0$ Y$ML1$ Z$ML2$ （%ML0=100.0，%ML1=200.0，%ML2=300.0）

说明：执行此 G 代码后 X 轴运行至 100 单位；Y 轴运行至 200 单位；Z 轴运行至 300 单位。

  - **G 代码的交接。**

通过 G50/G51/G52 可以更改 G 代码的交接模式，G0/G1/G2/G3 可以使用的交接模式如下表：

	G50 （准确停止）	G51 （圆弧交接）	G52 （圆滑交接）
G0	可以使用	交接模式没有作用，运行效果和 G50 一致	交接模式没有作用，运行效果和 G50 一致
G1	可以使用	可以使用	可以使用
G2	可以使用	可以使用	直线/圆弧和圆弧相切或接近相切时可以使用
G3	可以使用	可以使用	直线/圆弧和圆弧相切或接近相切时可以使用

- **各项默认情况**

1. 相对、绝对默认：默认模式为绝对模式，可通过 G90/G91 设定。
2. 平面默认：默认平面为 XY 平面。可通过 G17/G18/G19 切换平面。
3. 交接模式默认：默认精确停止。可通过 G50/G51/G52 切换交接模式。
4. G0 相关默认值：速度、加速度，减速度和加速度的变化率为轴组参数中设置的各轴的速度、加速度、减速度和加速度的变化率。可以通过 DMC_SetG0Para 指令来设置这些参数。
5. G1/G2/G3 相关默认值：速度、加速度，减速度和加速度的变化率为轴组参数中设置的终端机构的速度，加速度，减速度和加速度的变化率，可以通过 DMC_SetG1Para 来设定默认参数。可以通过 E、F 参数更改。如需要加速段和减速段有不同的加速度，可在 G 代码的输入中使用 E+和 E- 来设置。

   范例：G1 X10000 Y32105.6 E+20000 E-90000

   说明：指令执行时刀具的加速阶段以 20000 单元/秒的加速度来执行，减速时以 90000 单元/秒的减速度来执行。

11.6.4 G 代码功能详细介绍

11.6.4.1 G90（绝对模式）

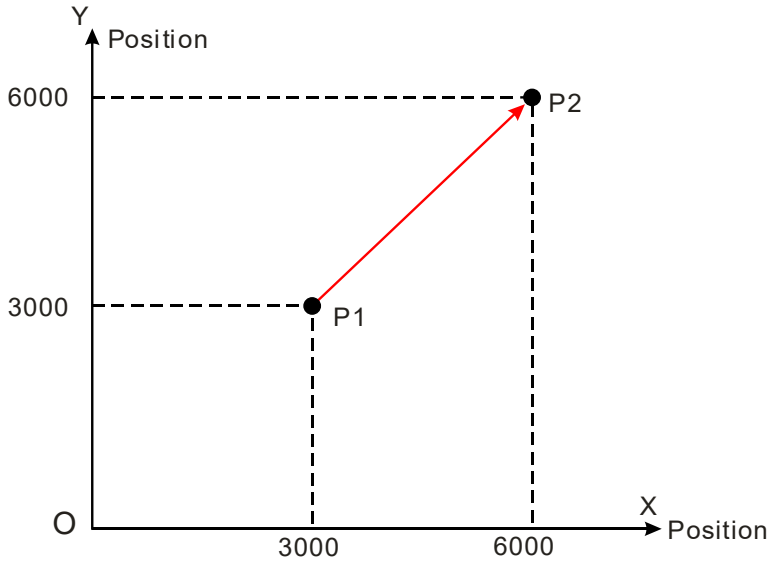
- 功能：G90 执行后，其后 G 代码中各轴的终点位置值均以 0 单元为基准，中途可以用 G91 指令切换为相对模式。NC 程序默认为绝对模式。
- 格式：N_G90
- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号。
- 范例：

  X、Y 轴的初始位置都为 3000 单元，轴参数都为默认值。将要执行的 G 代码如下：

```
N00 G90
N01 G0 X6000 Y6000
```

  G 代码执行后，整个过程的 Y/X 曲线如下：



11.6.4.2 G91（相对模式）

- 功能：G91 执行后，其后 G 代码中各轴的终点位置均以从当前位置开始的增量值计算，中途可以用 G90 指令切换为绝对模式。
- 格式：N_G91
- 参数说明：

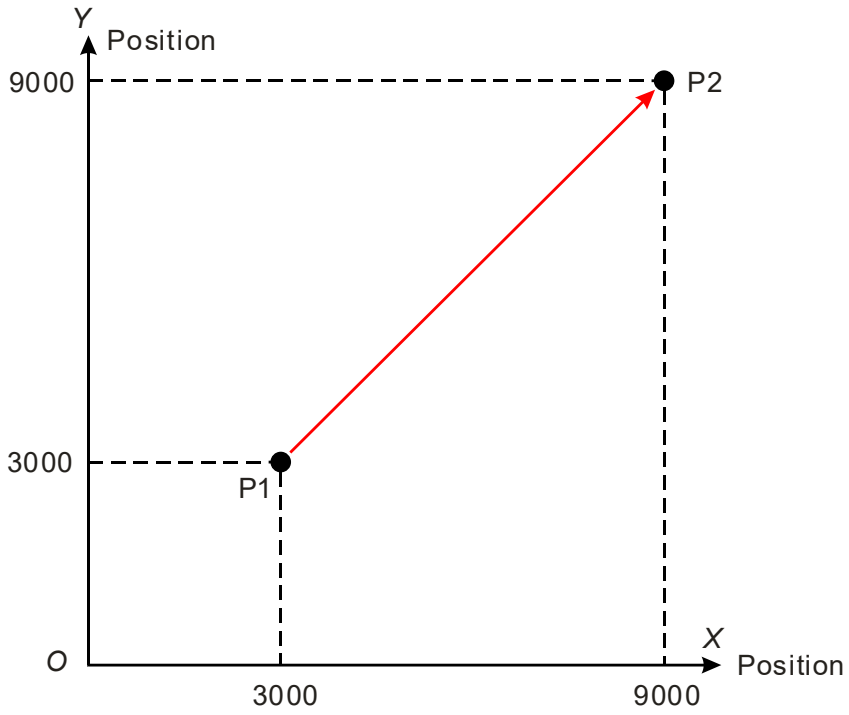
  N_：G 代码在 NC 程序中所在行的编号。
- 范例：

  X、Y 轴的初始位置都为 3000 单元，轴参数都为默认值。将要执行的 G 代码如下：

  N00 G91

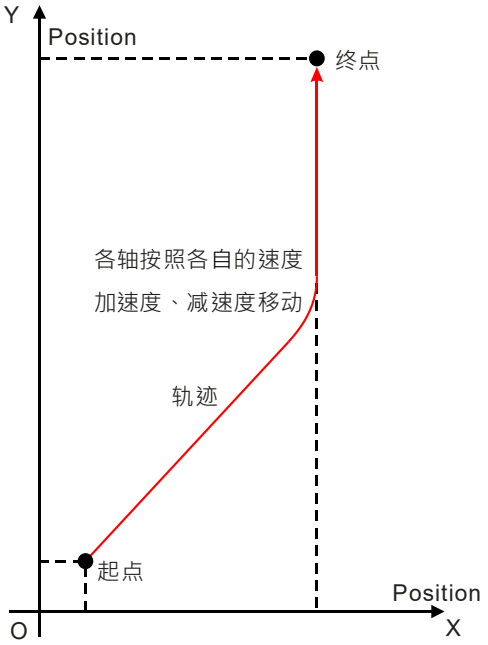
  N01 G0 X6000 Y6000

  G 代码执行后，整个过程的 Y/X 曲线如下：



11.6.4.3 G0（快速定位）

- 功能：各轴以指定速度从当前位置运动到终点位置。最多可控制八个轴，运动过程中各轴是相互独立的，运动轨迹如下所示。



- 格式：N_G0 X_Y_Z_A_B_C_P_Q_
- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号。

  X_：指定 X 轴终点位置，单位：单元，数据类型：LREAL。

  Y_：指定 Y 轴终点位置，单位：单元，数据类型：LREAL。

  Z_：指定 Z 轴终点位置，单位：单元，数据类型：LREAL。

  A_：指定 A 轴终点位置，单位：单元，数据类型：LREAL。

  B_：指定 B 轴终点位置，单位：单元，数据类型：LREAL。

  C_：指定 C 轴终点位置，单位：单元，数据类型：LREAL。

  P_：指定 P 轴终点位置，单位：单元，数据类型：LREAL。

  Q_：指定 Q 轴终点位置，单位：单元，数据类型：LREAL。

- 指令说明：

  1. G0 可以控制其中一个轴或多个轴，其他轴可以被缺省。
  2. 各轴运动时的速度、加速度、减速度和加速度的变化率分别由轴组参数的各轴相关参数决定。可以通过 DMC_SetG0Para 指令来设置这些参数。
  3. 绝对模式（由 G90 决定）：将 G0 的终点位置以 0 单元为基准。
  4. 相对模式（由 G91 决定）：将 G0 的终点位置作为从当前位置开始的增量值处理。

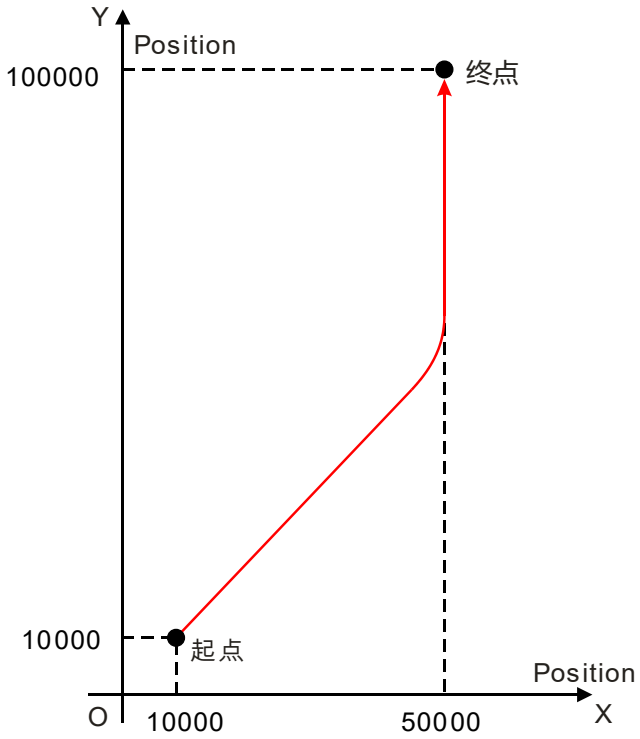
  ■ 绝对模式范例：

  X、Y 轴的初始位置都为 10000 单元，轴参数都为默认值。将要执行的 G 代码如下：

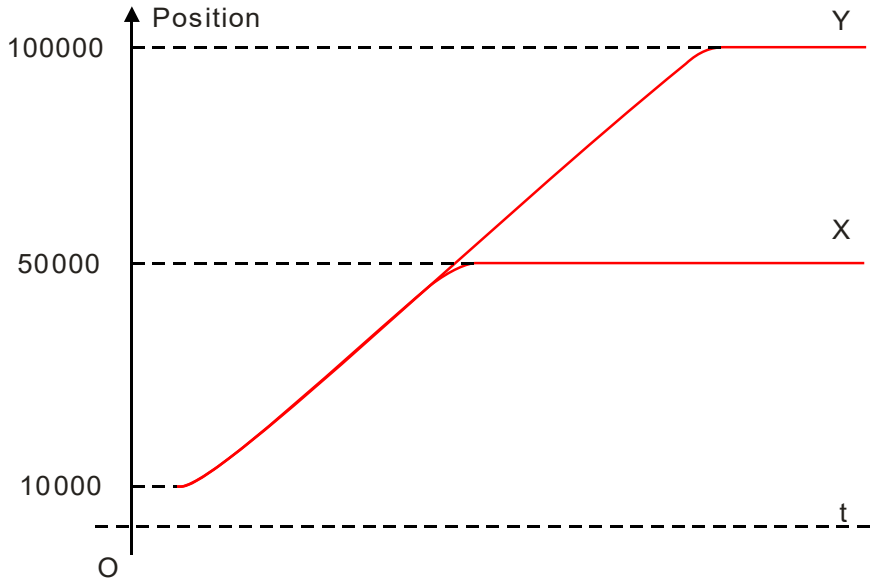
  N00 G90

N01 G0 X50000 Y100000

➢ G 代码执行后，整个过程的 Y/X 曲线如下：



➢ G 代码执行后，整个过程的位置/时间曲线如下：

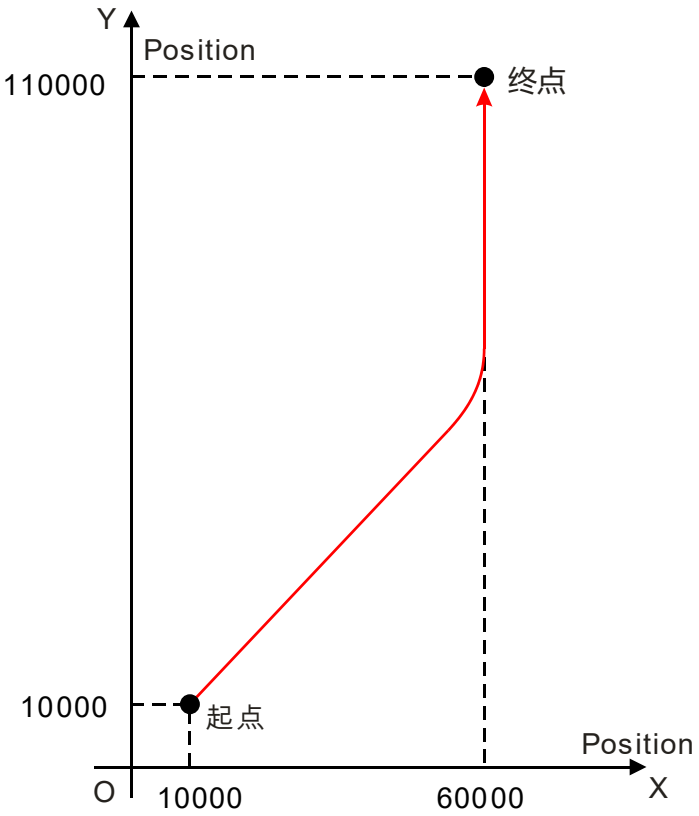


■ 相对模式范例：

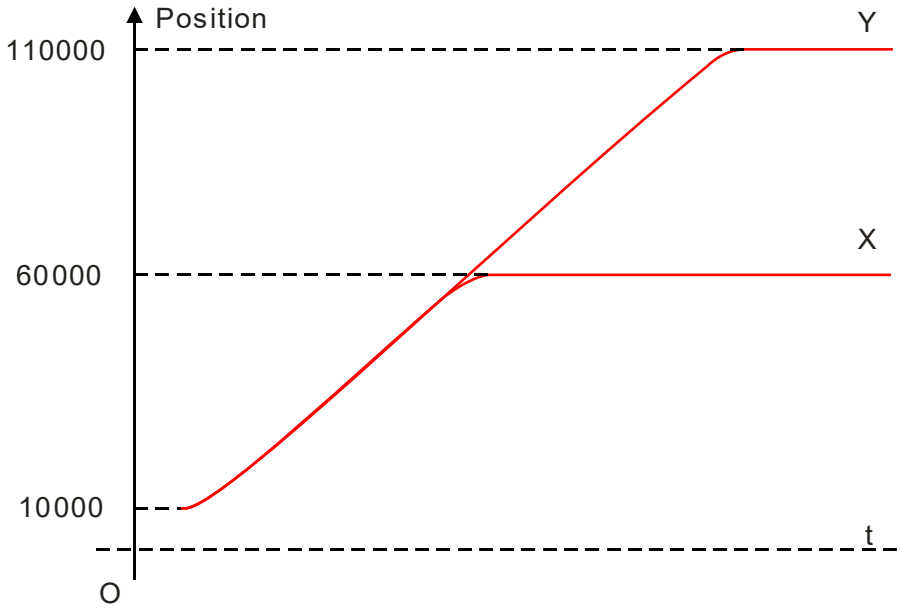
X、Y 轴的初始位置都为 10000 单元，轴参数都为默认值。将要执行的 G 代码如下：

N00 G91
N01 G0 X50000 Y100000

➢ G 代码执行后，整个过程的 Y/X 曲线如下：

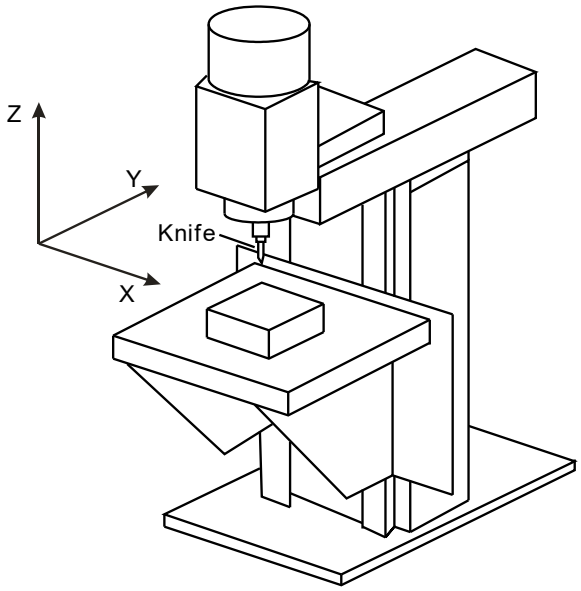


➢ G 代码执行后，整个过程的位置/时间曲线如下：



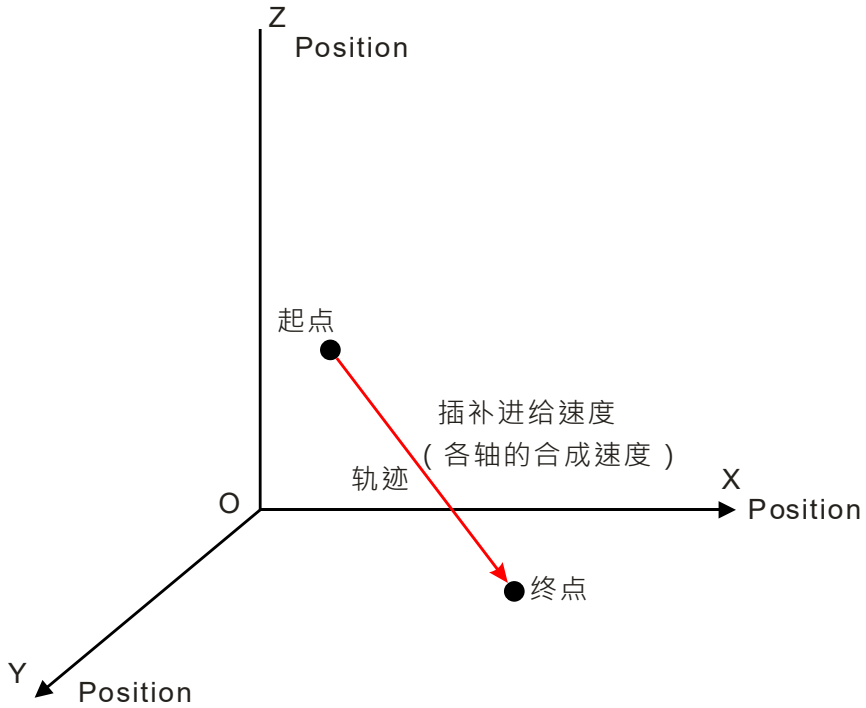
11.6.4.4 G1（直线插补）

- 功能：刀具以指定速度从某一点出发，直线移动到目标位置。此指令最多可控制八个轴，且各轴同起同停。如下图所示，三个轴一起控制刀具的位置。



立式铣床

运动轨迹如下图所示：



- 格式：N_G1 X_Y_Z_A_B_C_P_Q_ E_ E_F_
- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号。

  X_：指定 X 轴终点位置，单位：单元，数据类型：LREAL。

Y_：指定 Y 轴终点位置，单位：单元，数据类型：LREAL。

Z_：指定 Z 轴终点位置，单位：单元，数据类型：LREAL。

A_：指定 A 轴终点位置，单位：单元，数据类型：LREAL。

B_：指定 B 轴终点位置，单位：单元，数据类型：LREAL。

C_：指定 C 轴终点位置，单位：单元，数据类型：LREAL。

P_：指定 P 轴终点位置，单位：单元，数据类型：LREAL。

Q_：指定 Q 轴终点位置，单位：单元，数据类型：LREAL。

E_：指定刀头的加速度和减速度，正数表示加速度，负数表示减速度，单位：单元/秒 2，数据类型：LREAL 。

F_：指定刀头的进给速度，单位：单元/秒，数据类型：LREAL。

刀头匀速时，G 代码中所有轴的合成速度与 F 值相等。计算方法如下：

若存在两个轴时，$F=\sqrt{V_1^2+V_2^2}$ 。

若存在三个轴时，$F=\sqrt{V_1^2+V_2^2+V_3^2}$ 。

若存在更多轴时，按照上述方法类推。

- 指令说明：

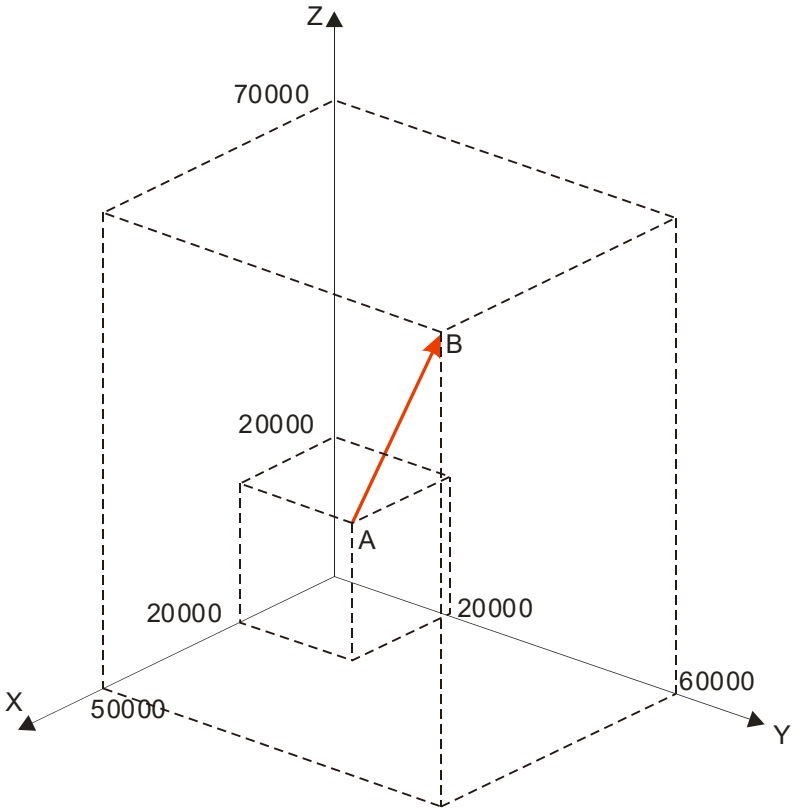
1. G1 可以控制其中一个轴或多个轴，其他轴可以被缺省。
2. E、F 都可被缺省。若在 CNC 编辑区内若仅有一行代码，缺省后刀头的速度、加速度、减速度和加速度的变化率由轴组的轴参数决定，可以通过 DMC_SetG1Para 指令来设置这些参数；若 CNC 编辑区有多行代码且 G1 代码缺省 E、F，则刀头的速度、加速度、减速度以 G1 行之前代码中生效的 E 和 F 为准，若之前 G 代码没有指定 E、F，则以轴组的轴参数决定。
3. 绝对模式 (由 G90 决定): 将 G1 的终点位置以 0 单元为基准。
4. 相对模式 (由 G91 决定): 将 G1 的终点位置作为从当前位置开始的增量值处理。

- 绝对模式范例：

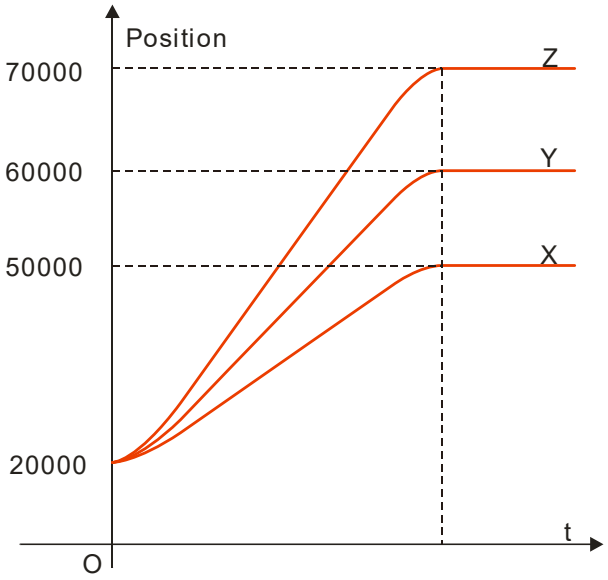
X、Y、Z 轴的初始位置都为 20000 单元，轴参数都为默认值。将要执行的 G 代码如下：

```
N00 G90
N01 G1 X50000 Y60000 Z70000
```

➢ G 代码执行后，整个过程的 Y/X 曲线如下：



➢ G 代码执行后，整个过程的位置/时间曲线如下：



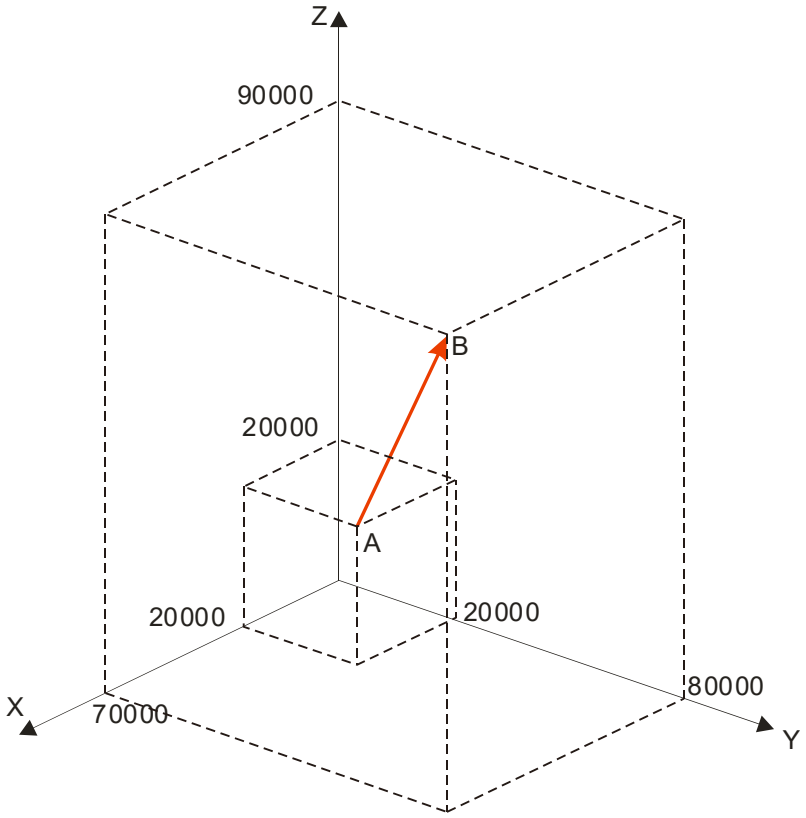
■ 相对模式范例：

X、Y、Z 轴的初始位置都为 20000 单元，轴参数都为默认值。将要执行的 G 代码如下：

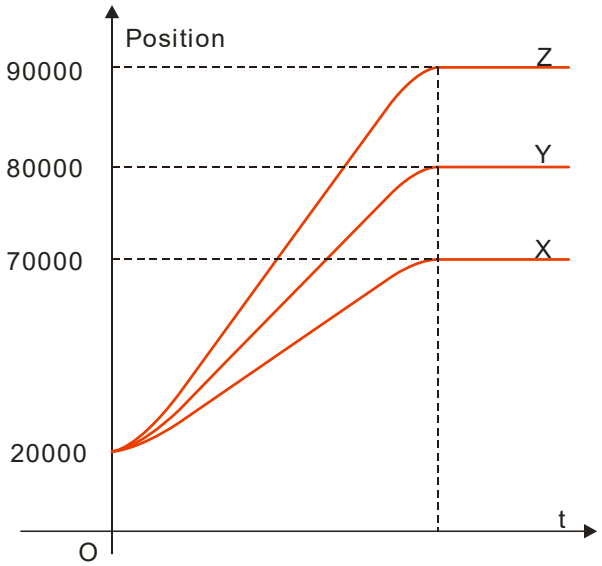
N00 G91

N01 G1 X50000 Y60000 Z70000

➢ G 代码执行后，整个过程的 Y/X 曲线如下：



➢ G 代码执行后，整个过程的位置/时间曲线如下：

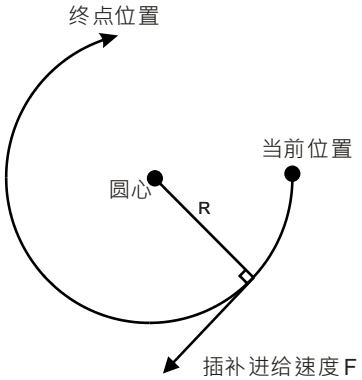


11.6.4.5 G2（顺时针圆弧/螺旋插补）

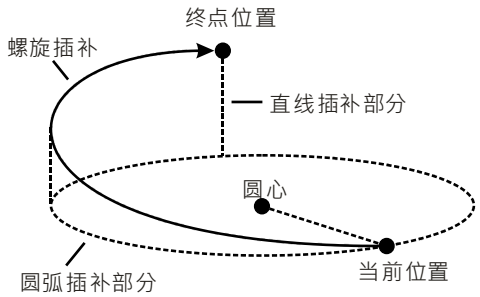
- 功能

  圆弧插补：刀具在指定平面及由圆心或半径确定的圆弧上，以参数 F 设定的进给速度，对加工物件进行顺时针方向圆弧切削。

  螺旋插补：刀具在指定的平面及圆弧上顺时针方向移动（圆弧插补）时，同时在指定平面的垂直方向上进行直线移动（直线插补），进给速度有参数 F 指定。



圆弧插补



螺旋插补

- 格式：

  格式 1：N_G2 X_Y_Z_A_B_C_P_Q_I_J_（I_K_/J_K_）T_ E_E_F_

  格式 2：N_G2 X_Y_Z_A_B_C_P_Q_R_T_ E_E_F_

- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号

  X_Y_Z_：指定圆弧终点对应 X 轴、Y 轴和 Z 轴的终点位置，单位：单元，数据类型：LREAL。

  A_B_C_P_Q_：指定各附加轴的终点位置，单位：单元，数据类型：LREAL。

  I_J_：指定 XY 平面的圆心坐标位置，单位：单元，数据类型：LREAL。

  I_K_：指定 XZ 平面的圆心坐标位置，单位：单元，数据类型：LREAL。

  J_K_：指定 YZ 平面的圆心坐标位置，单位：单元，数据类型：LREAL。

  T_：指定整圆圈数，单位：圈，数据类型：ULINT。

  E_：指定刀头的加速度和减速度，正数表示加速度，负数表示减速度，单位：单元/秒 2，数据类型：LREAL。

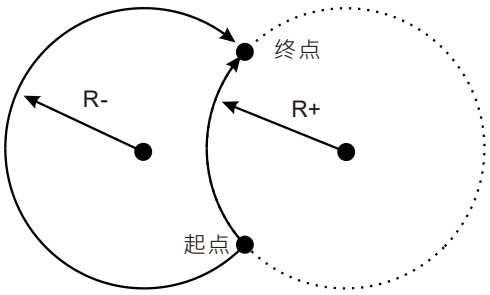
  F_：指定刀头的进给速度，单位：单元/秒，数据类型：LREAL。

- 指令说明：

  1. X、Y、Z 中的两个轴在指定平面（平面由 G17/G18/G19 指令指定平面）上做圆弧插补运动，第 3 个轴垂直指定平面做直线插补运动。
  2. 附加轴 A、B、C、P、Q 做直线插补运动，与圆弧动作同启同停。
  3. E_、F_都可被缺省。在 CNC 编辑区内若仅有一行代码，缺省后刀头的速度、加速度、减速度由轴组参数决定, 可以通过 DMC_SetG1Para 指令来设置这些参数；若 CNC 编辑区有多行代码且 G2 代码缺省 E、F，则刀头的速度、加速度、减速度以 G2 行之前代码中生效的 E 和 F 为准，若之前 G 代码没有指定 E、F，则以轴组参数中的速度、最大加速度、最大减速度为准。

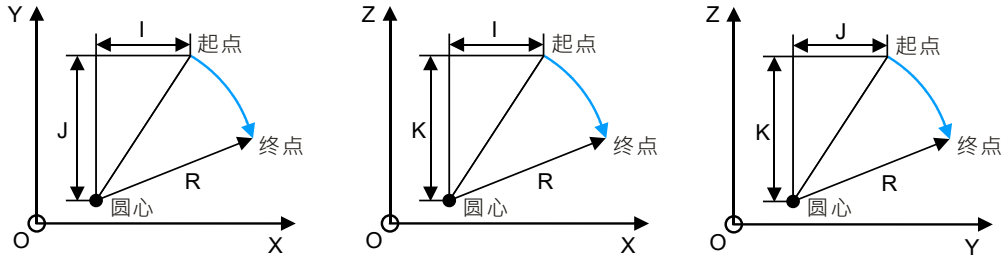
4. 在 G90 绝对模式时，圆弧终点是以各自方向上的 0 单元为参考的绝对坐标值。在 G91 相对模式时，圆弧终点是相对圆弧起点的增量值。
5. 不管是绝对还是相对模式，圆心坐标 I_J_（I_K_/J_K_） 始终是以起点为参考的相对坐标。
6. T 为整圆圈数，为零时轨迹为弧的长度；为常数时即为相应的设置圈数再加上弧的长度。
7. 格式 2 与格式 1 的不同在于，它通过起点、终点和半径来确定一段圆弧。 R 参数后面输入正数时（R+）代表选择劣弧（小于 180 度），输入负数时（R-）代表选择优弧（大于 180 度）。

   下图所示实线部分为 G2 指令选择 R+和 R-时的运行轨迹，圆弧上的箭头表示运转方向。



顺时针运转

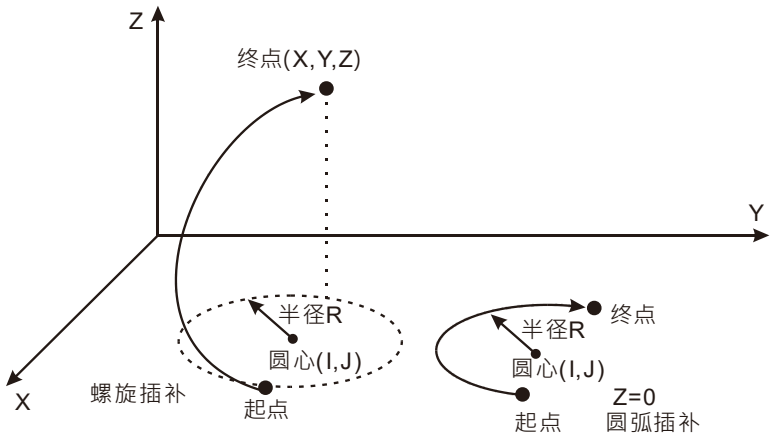
- 下图讲解在不同平面的坐标关系：

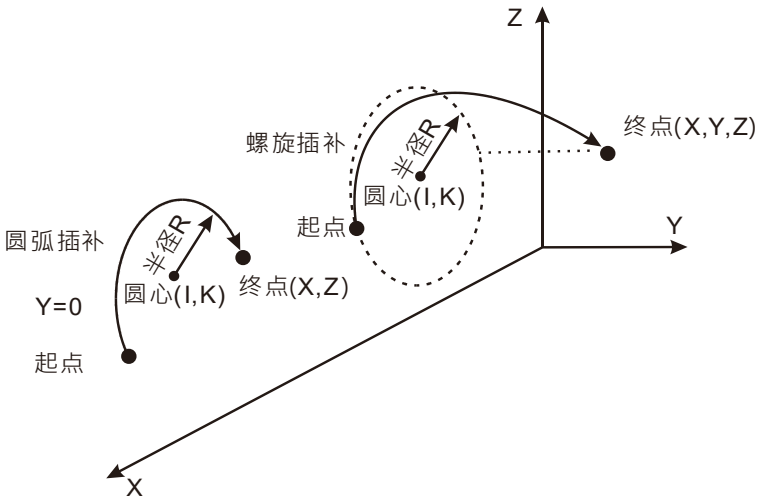
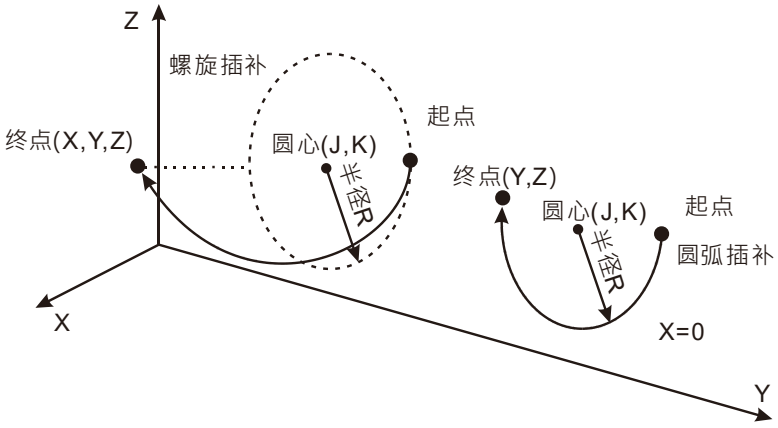


a) XY 平面圆弧　　b) XZ 平面圆弧　　c) YZ 平面圆弧

注意：坐标平面与 I、J、K 的关系，一条圆弧指令中最多出现 I、J、K 中的两个，至于出现哪两个则由相应的平面决定，如 XY 平面只能出现 I、J。

坐标平面可由 G17、G18 和 G19 来设置，G2 在不同坐标平面的圆弧和螺旋运转轨迹如下：

G 代码	功能说明	轨迹示意图
G17	XY 平面： 当起点和终点对应 Z 轴坐标没有变化量时运动轨迹为圆弧插补，否则为螺旋插补。	

G 代码	功能说明	轨迹示意图
G18	XZ 平面： 当起点和终点对应 Y 轴坐标没有变化量时运动轨迹为圆弧插补，否则为螺旋插补。	
G19	YZ 平面： 当起点和终点对应 X 轴坐标没有变化量时运动轨迹为圆弧插补，否则为螺旋插补。	

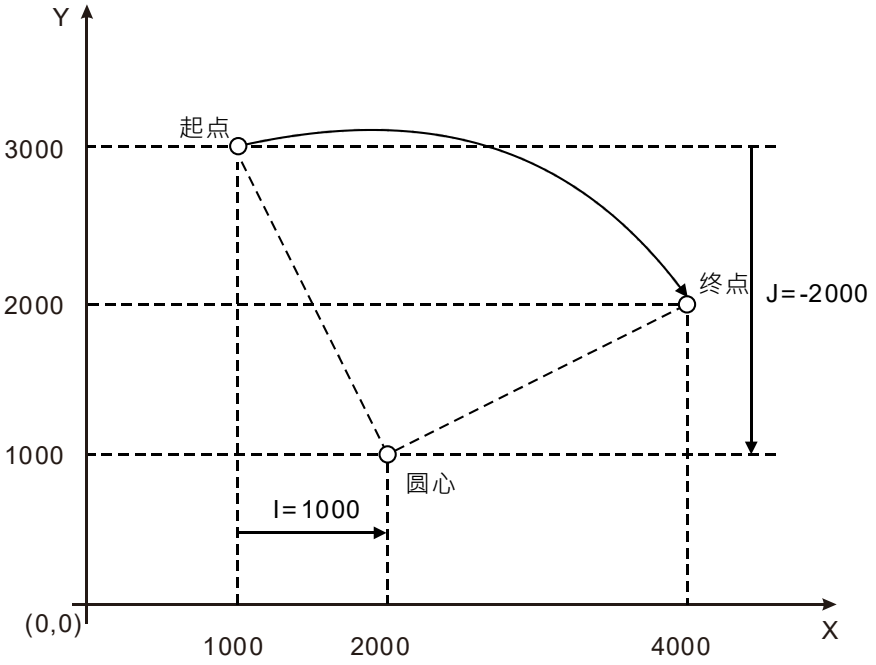
程序范例一

- **绝对模式下指定圆心圆弧插补**

  ■ 当前位置为（1000，3000），轴参数都为默认值，将要执行的 G 代码如下：

  N00 G90
  N01 G17
  N02 G2 X4000 Y2000 I1000 J-2000 E5000 F5000

- G 代码执行后，整个过程的 Y/X 曲线如下：



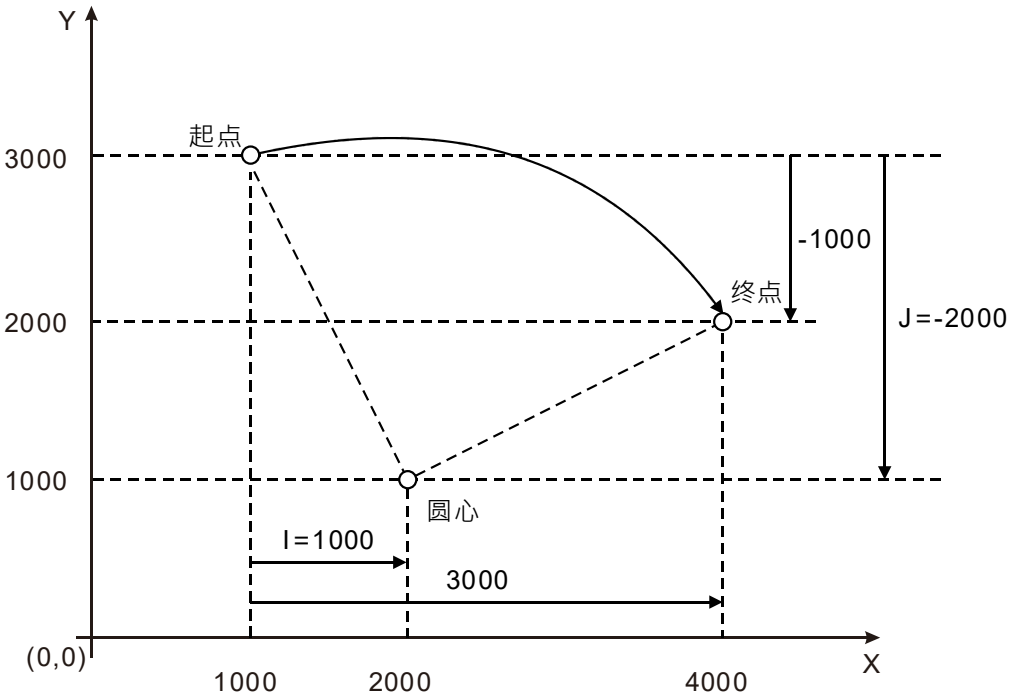
程序范例二

- **相对模式下指定圆心圆弧插补**

  - 当前位置为（1000，3000），轴参数都为默认值。将要执行的 G 代码如下：

    N00 G91
    N01 G17
    N02 G2 X3000 Y-1000 I1000 J-2000

  - G 代码执行后，整个过程的 Y/X 曲线如下：



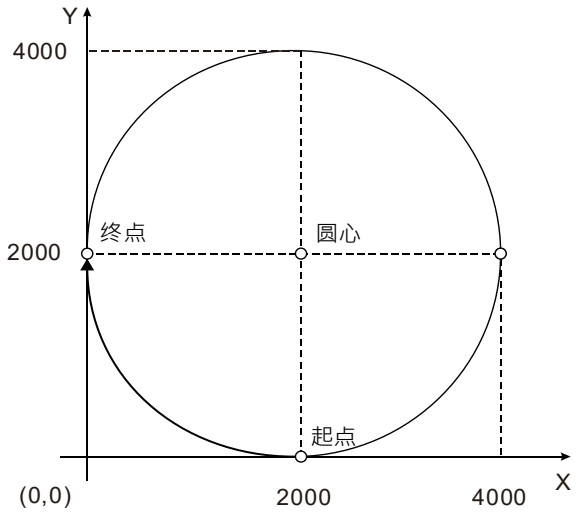
程序范例三

- **相对模式下指定圆心且带 T 的圆弧插补：**

  - 当前位置为（2000，0），轴参数都为默认值。将要执行的 G 代码如下：

    N00 G91
    N01 G17
    N02 G2 X-2000 Y2000 I0 J2000 T3

  - G 代码执行后，圆弧的轨迹为 3 圈+1/4 圆弧（粗体圆弧），整个过程的 Y/X 曲线如下



程序范例四

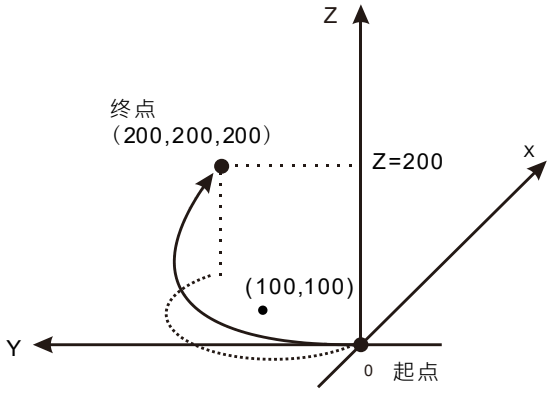
- **XY 平面指定圆心的螺旋插补**

  当前位置为（0，0），轴参数都为默认值。将要执行的 G 代码如下：

  N00 G17
  N01 G91
  N02 G2 X200 Y200 Z200 I100 J100 E+10000 E-20000 F1000

- **指令说明：**

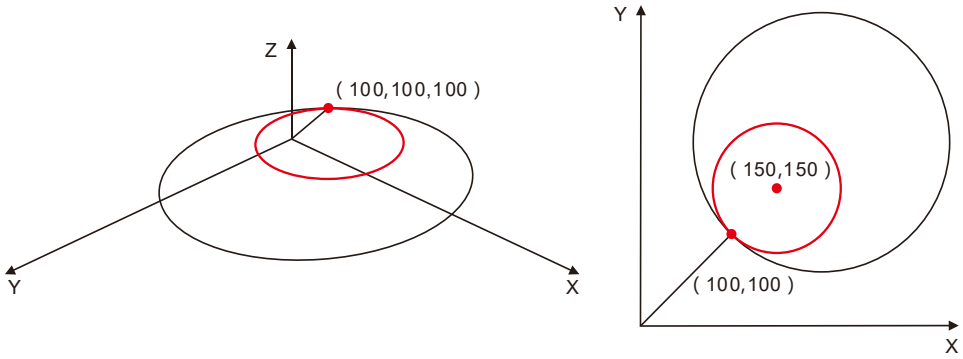
执行 G2 时，轴以零为起点，以轴坐标参数为终点，按照顺时针画圆，运行轨迹为螺旋曲线。在 XY 面投影为以（100，100）为圆心的半圆弧。



程序范例五

- **缺省格式介绍，将要执行的 G 代码如下：**

```
N00 G0 X0 Y0 Z0
N01 G1 X100 Y100 Z100
N02 G2 I100 J100
N03 G91
N04 G2 I50 J50
```



- **指令说明：**

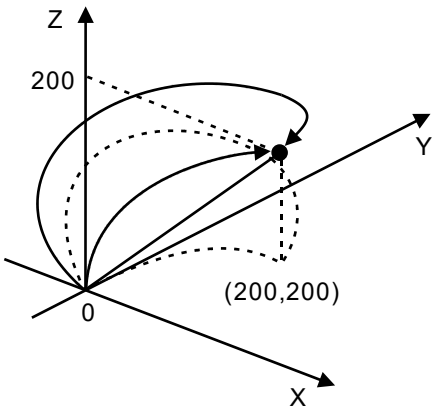
1. N01 行执行完毕轴位置为（100，100，100）；
2. N02 行指令只有 I、J 参数，其余缺省的参数的值皆以其上一条生效的为准，即相当于指令为 N02 X100 Y100 Z100 I100 J100，所以其起点和终点坐标均为（100，100，100），故其运行轨迹为一整圆；
3. N03 行指令为 G91 即其后面的坐标为相对模式，但是 N04 由于省略了 X、Y、Z，这条的终点位置还是绝对位置(100,100,100)，所以 N04 的运行轨迹为起点终点都为(100,100,100)，圆心坐标为(150,150)的一整圆。

程序范例六

- XY 平面指定半径的螺旋插补范例（当前位置为 0），将要执行的 G 代码如下：

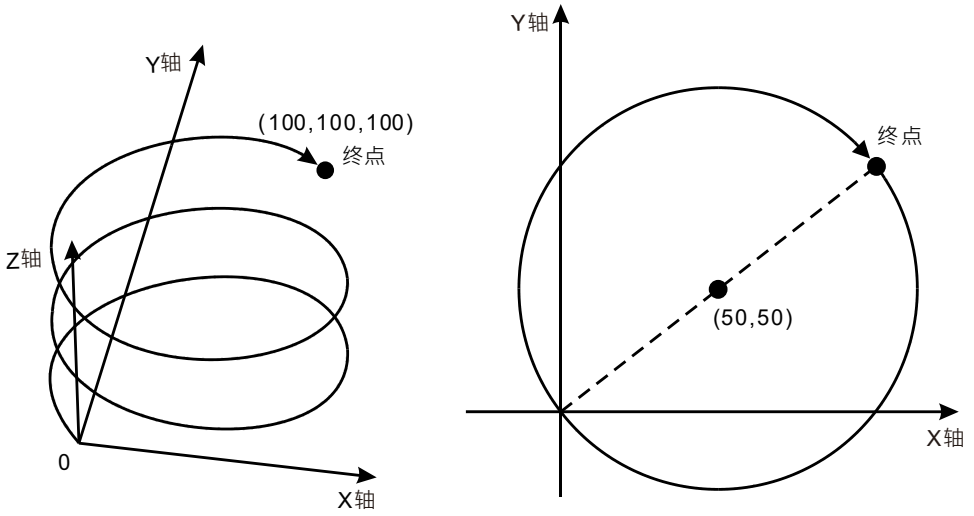
```
N00 G2 X200 Y200 Z200 R-200
N01 G0 X0 Y0 Z0
N02 G2 X200 Y200 Z200 R200
```

- 指令说明：执行第一个 G2 代码时运动轨迹为优弧，执行第二个 G2 代码是运动轨迹为劣弧。



程序范例七

- XY 平面指定圆心带 T 的螺旋插补范例（当前位置为 0），将要执行的 G 代码如下：
  N00 G2 X100 Y100 Z100 I50 J50 T2
- 指令说明：运行轨迹为螺旋曲线，在 XY 面投影为整圆，圆心为（50，50）。

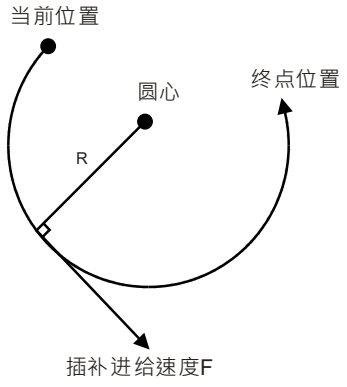


11.6.4.6 G3（逆时针圆弧/螺旋插补）

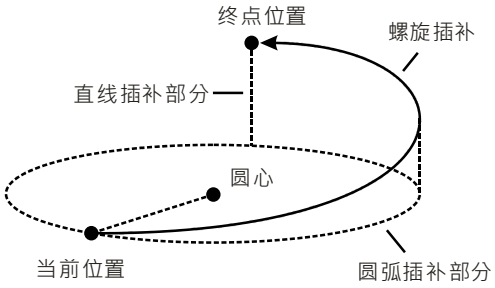
- 功能说明：

  圆弧插补：刀具在指定平面及由圆心或半径确定的圆弧上，以参数 F 设定的进给速度，对加工物件进行逆时针方向圆弧切削。

  螺旋插补：刀具在指定的平面及圆弧上逆时针方向移动（圆弧插补）时，同时在指定平面的垂直方向上进行直线移动（直线插补），进给速度由参数 F 指定。



圆弧插补



螺旋插补

- 格式：

  格式 1：N_G3 X_Y_Z_A_B_C_P_Q_I_J_（I_K_/J_K_）T_ E_E_F_

  格式 2：N_G3 X_Y_Z_A_B_C_P_Q_R_T_ E_E_F_

- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号

  X_Y_Z_：指定圆弧终点对应 X 轴、Y 轴和 Z 轴的终点位置，单位：单元，数据类型：LREAL。

  A_B_C_P_Q_：指定附加轴的终点位置，单位：单元，数据类型：LREAL。

  I_J_：指定圆心对应 X 轴和 Y 轴的坐标位置，单位：单元，数据类型：LREAL。

  I_K_：指定圆心对应 X 轴和 Z 轴的坐标位置，单位：单元，数据类型：LREAL。

  J_K_：指定圆心对应 Y 轴和 Z 轴的坐标位置，单位：单元，数据类型：LREAL。

  T_：指定整圆圈数，单位：圈，数据类型：ULINT。

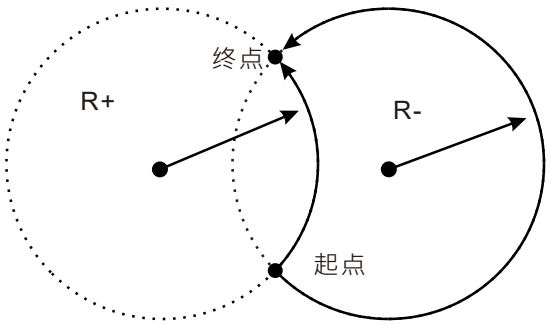
  E_：指定刀头的加速度和减速度，正数表示加速度，负数表示减速度，单位：单元/秒 2，数据类型：LREAL。

  F_：指定刀头的进给速度，单元/秒，数据类型：LREAL。

- 指令说明：

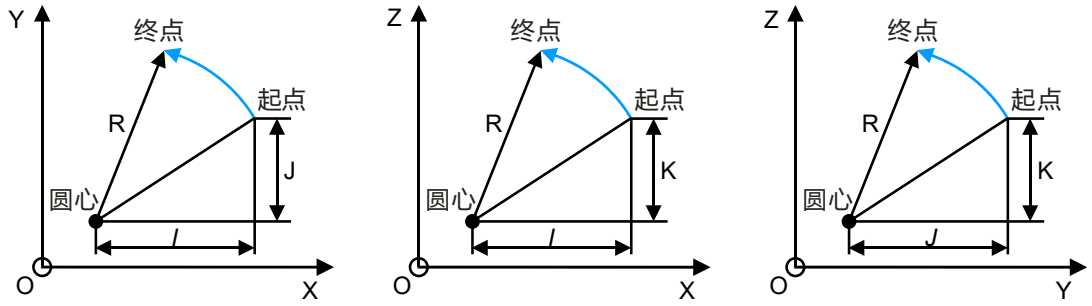
  1. X、Y、Z 中的两个轴在指定平面（由 G17/G18/G19 指定平面）上做圆弧插补运动，第 3 个轴垂直指定平面做直线插补运动。
  2. 附加轴 A、B、C、P、Q 做直线插补运动，与圆弧动作同启同停。
  3. E_、F_都可被缺省。在 CNC 编辑区内若仅有一行代码，缺省后刀头的速度、加速度、减速度由轴组参数决定；若 CNC 编辑区有多行代码且 G3 代码缺省 E、F，则刀头的速度、加速度、减速度以 G3 行之前代码中生效的 E 和 F 为准，若之前 G 代码没有指定 E、 F， 则以轴组参数中的最大速度、最大加速度、最大减速度为准。

4. 在 G90 绝对模式时，圆弧终点是以各自方向上的 0 单元为参考的绝对坐标值。在 G91 相对模式时，圆弧终点是相对圆弧起点的增量值。
5. 不管是绝对模式和还是相对模式，圆心坐标 I_J_（I_K_/J_K_） 始终是以起点为参考的相对坐标。
6. T 为整圆圈数，为零时轨迹为弧的长度；为常数时即为相应的设置圈数再加上弧的长度。
7. 格式 2 与格式 1 的不同在于：它通过起点、终点和半径来确定一段圆弧。R 参数后面输入正数时(R+) 代表选择劣弧（小于 180 度），输入负数时（R-）代表选择优弧（大于 180 度）。
8. 下图所示实线部分为 G3 指令选择 R+和 R-时的运行轨迹，圆弧上的箭头表示运转方向。



逆时针运转

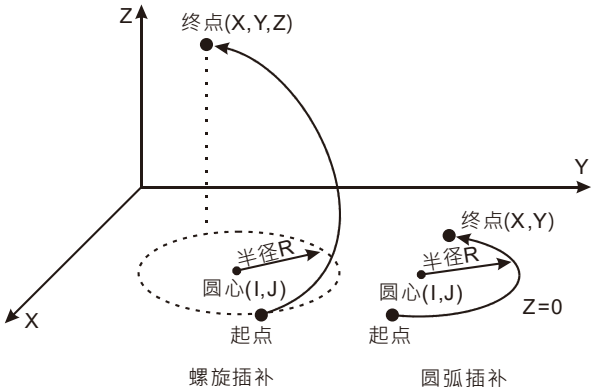
- 下图讲解在不同平面的坐标关系：

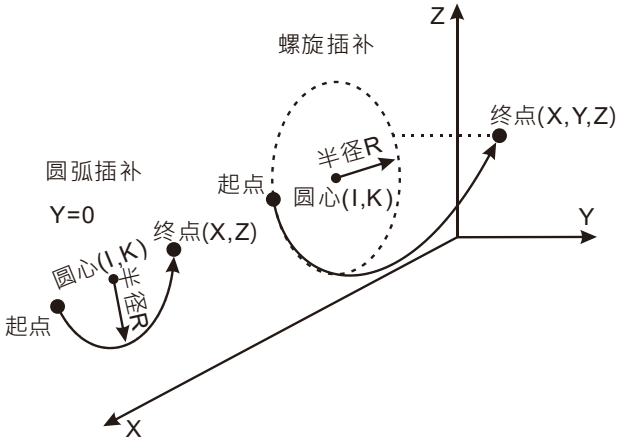
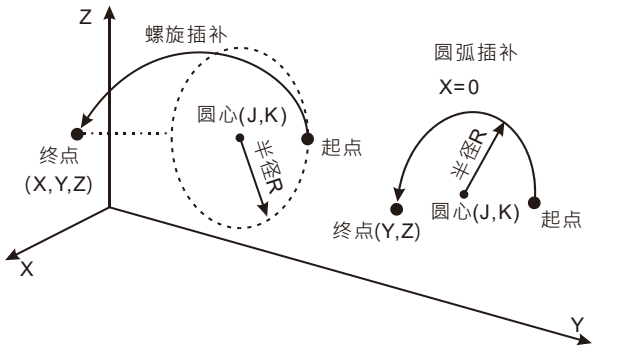


a) XY 平面圆弧　　b) XZ 平面圆弧　　c) YZ 平面圆弧

注意：坐标平面与 I、J、K 的关系，一条圆弧指令中最多出现 I、J、K 中的两个，至于出现哪两个则由相应的平面决定，如 XY 平面只能出现 I、J。

坐标平面可由 G17、G18 和 G19 来设置，G3 在不同坐标平面的圆弧和螺旋运转轨迹如下：

G 代码	功能说明	轨迹示意图
G17	XY 平面： 当起点和终点对应 Z 轴坐标有变化量时，运动轨迹为螺旋插补，否则为 XY 面内的圆弧插补。	

G 代码	功能说明	轨迹示意图
G18	XZ 平面： 当起点和终点对应 Y 轴坐标有变化量时，运动轨迹为螺旋插补，否则为XZ面内的圆弧插补。	
G19	YZ 平面： 当起点和终点对应 X 轴坐标有变化量时，运动轨迹为螺旋插补，否则为YZ面内的圆弧插补。	

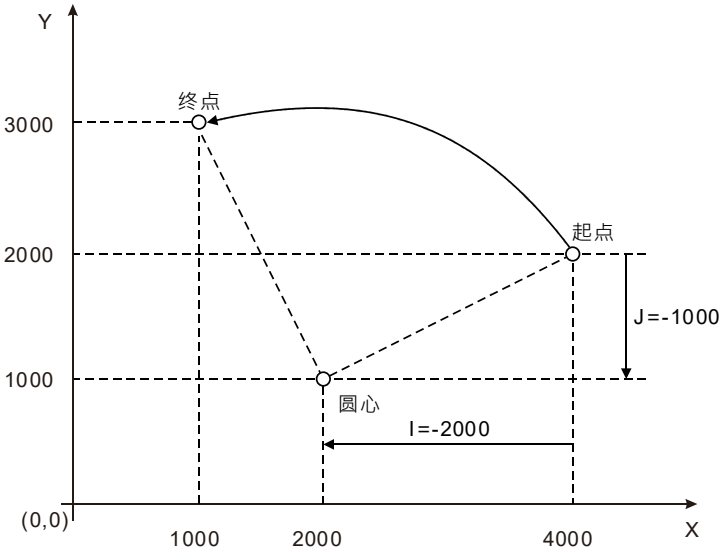
程序范例一

- 绝对模式下指定圆心圆弧插补：

  当前位置为（4000，2000），轴参数都为默认值。将要执行的 G 代码如下：

  N00 G90
  N01 G17
  N02 G3 X1000 Y3000 I-2000 J-1000

  G 代码执行后，整个过程的 Y/X 曲线如下：



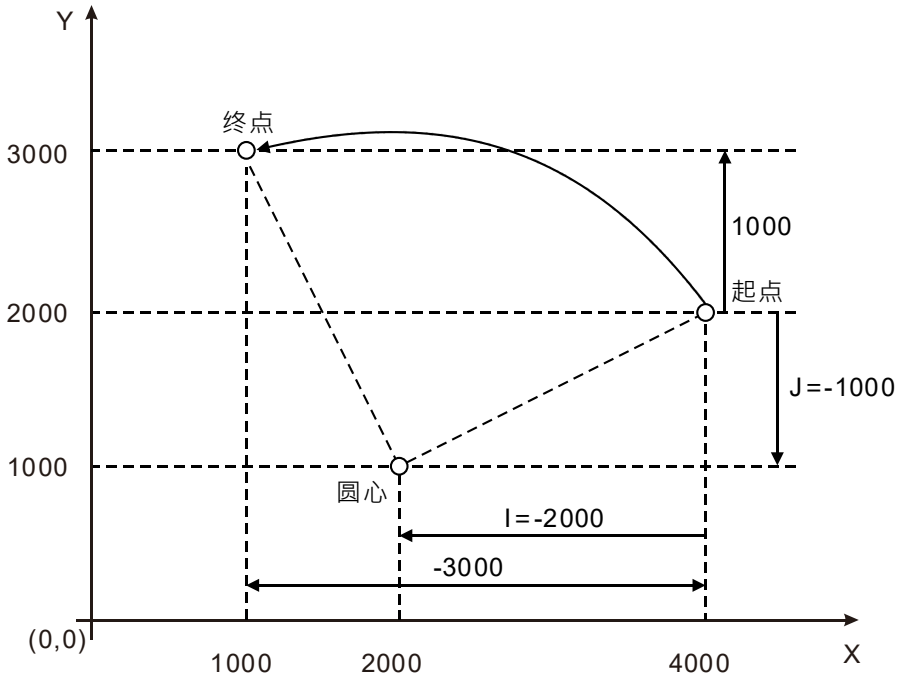
程序范例二

- 相对模式下指定圆心圆弧插补

  当前位置为（4000，2000），轴参数都为默认值。将要执行的 G 代码如下：

  N00 G91
  N01 G17
  N02 G3 X-3000 Y1000 I-2000 J-1000

  G 代码执行后，整个过程的 Y/X 曲线如下：



程序范例三

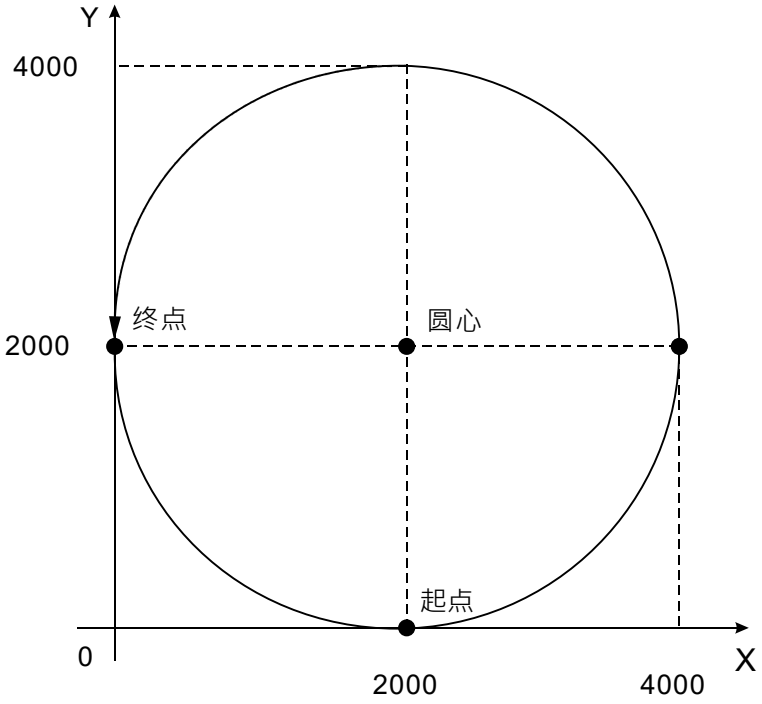
- 相对模式下指定圆心且带 T 的圆弧插补：

  当前位置为（2000，0），轴参数都为默认值。将要执行的 G 代码如下：

  N00 G91
  N01 G17
  N02 G3 X-2000 Y2000 I0 J2000 T3

- 指令说明：

  G 代码执行后，运行轨迹为 XY 面的圆弧，弧长为（3+3/4）倍圆的周长。



程序范例四

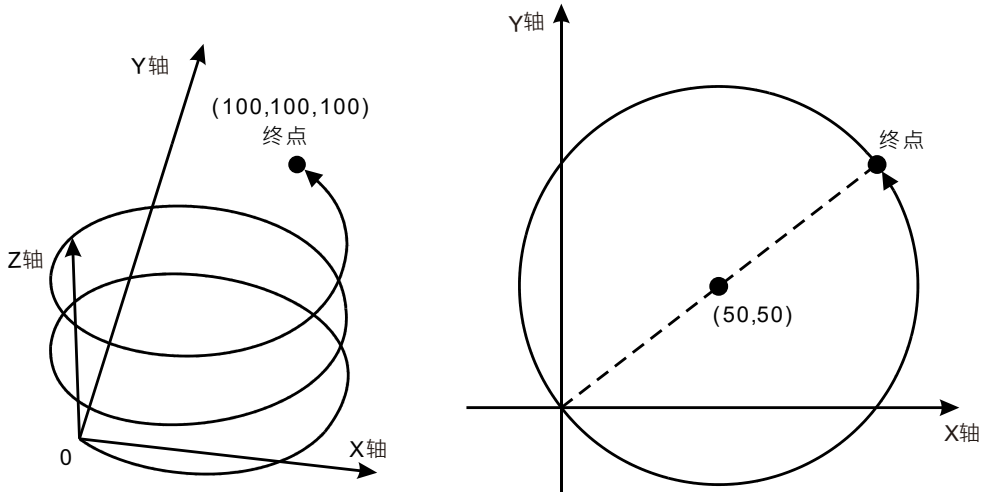
- 指定圆心螺旋插补：当前位置为（0，0），轴参数都为默认值。将要执行的 G 代码如下：

```
N00 G17
N01 G3 X100 Y100 Z100 I50 J50 T2
```

- 指令说明：

  由于 Z 轴的变化量为 100.故运行轨迹为螺旋曲线，XY 面的投影为整圆；

  若 Z 无变化量时，运行轨迹为 XY 面的圆弧，圆心为（50，50），弧长为 2.5 倍圆的周长。



11.6.4.7 G17/G18/G19（指定圆弧插补平面）

- 功能：

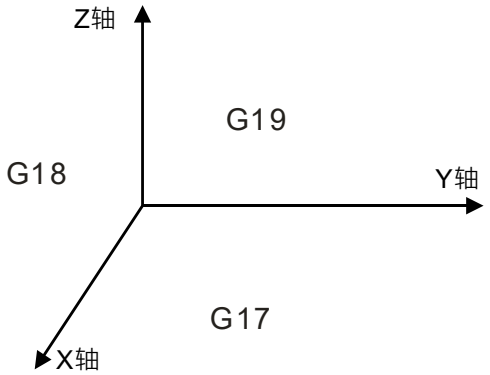
  此三个指令用于决定圆弧\螺旋插补平面选择，对于直线插补无影响。

  程序执行时，此三个工作平面可相互切换。程序中若无设定任何平面选择，系统初始状态为 XY 平面（G17）状态。

- 格式:

  N_G17
  N_G18
  N_G19

- 参数说明: N_：G 代码在 NC 程序中所在行的编号。
- 平面示意图如下图所示：



11.6.4.8 G4（延时指令）

- 功能：延时指令。
- 格式：N_G4 K_
- 参数说明：

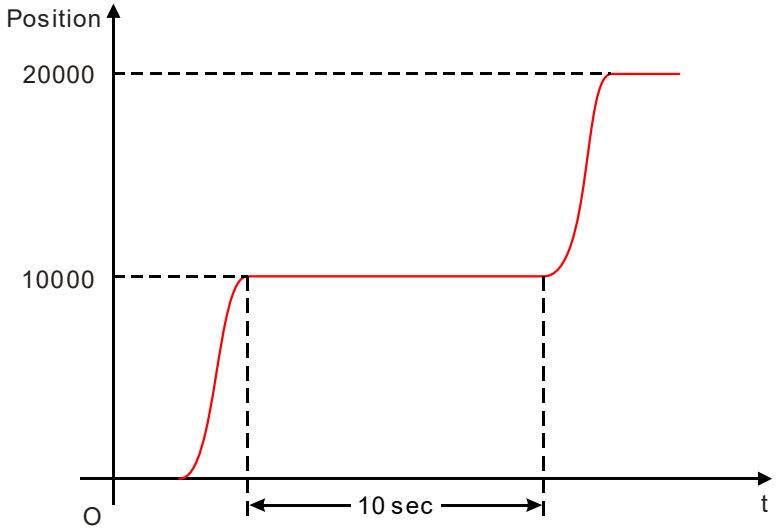
  N_：G 代码在 NC 程序中所在行的编号

  K_：指定延时时间，单位：秒。取值范围： 0.001 秒~100000 秒。
- 指令说明：

  当机床完成某一阶段的加工后，需要刀具暂时停止移动。此时，利用 G4 使刀具停止一段时间。

程序范例

```
N00 G1 X10000
N01 G4 K10
N02 G1 X20000
```



当编号为 N0 的指令执行完毕后，程序会延时 10 秒钟，之后再继续执行编号为 N2 的指令。

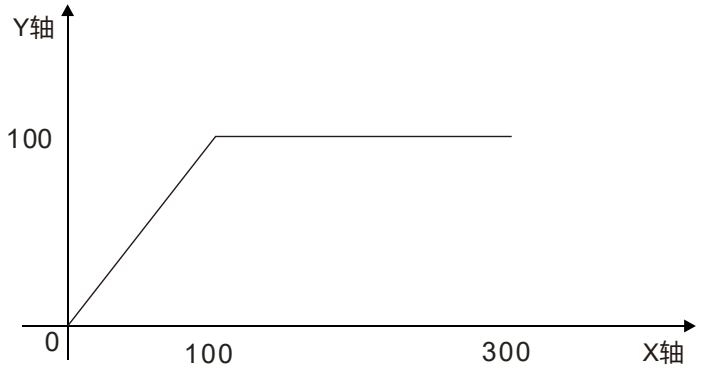
11.6.4.9 G50（准确停止）

- 功能：改变交接模式为准确停止，之后的交接模式都是准确停止，执行过程中可以使用 G51/G52 进行交接模式切换。每条 G 代码之间，终端机构的速度都会减为 0。
- 格式：N_ G50
- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号。

程序范例

```
N00 G50
N01 G1 X100 Y100
N02 G1 X300 Y100
```



11.6.4.10 G51（圆弧交接）

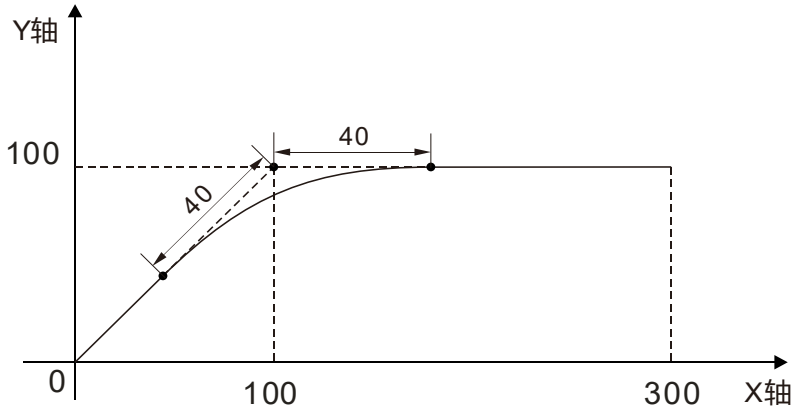
- 功能：改变交接模式为圆弧交接，之后的交接模式都是圆弧交接，执行过程中可以使用 G50/G52 进行交接模式切换。每条 G 代码之间交接，终端机构的不减速，交接曲线为圆弧。在该模式下，整体运行速度以第一条 G 代码的速度为准，无法在运行中通过 F 分量变更速度。如果需要变更速度，可以通过 DMC_CartesianCoordinate 的 VelOverride 引脚变更速度超调量来控制终端机构的运行速度。
- 格式：N_ G51 D_
- 参数说明：

  N_：G 代码在 NC 程序中所在行的编号。

  D_：圆弧的半径。

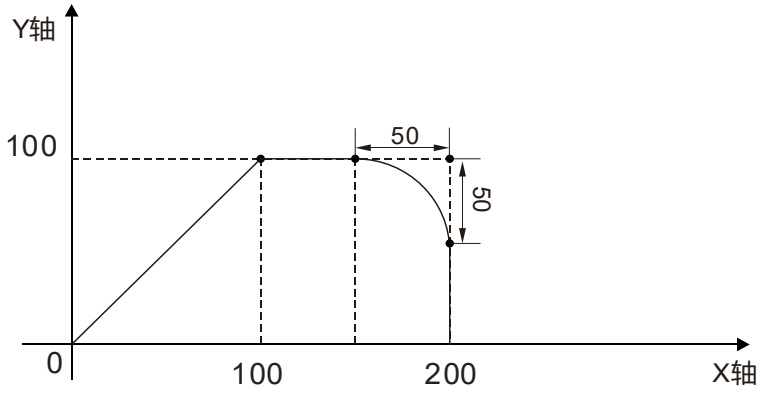
程序范例一

```
N00 G51 D40
N01 G1 X100 Y100
N02 G1 X300 Y100
```



程序范例二

```
N00 G1 X100 Y100
N01 G51 D50
N02 G1 X200 Y100
N03 G1 X200 Y0
```



11.6.4.11 G52（圆滑交接）

- 功能：改变交接模式为圆滑交接，之后的交接模式都是圆滑交接，执行过程中可以使用 G50/G51 进行交接模式切换。每条 G 代码之间交接，终端机构的不减速，适用于小线段连续插补。在该模式下，整体运行速度以第一条 G 代码的速度为准，无法在运行中通过 F 分量变更速度。如果需要变更速度，可以通过 DMC_CartesianCoordinate 的 VelOverride 引脚变更速度超调量来控制终端机构的运行速度。
- 格式：N_ G52
- 参数说明：

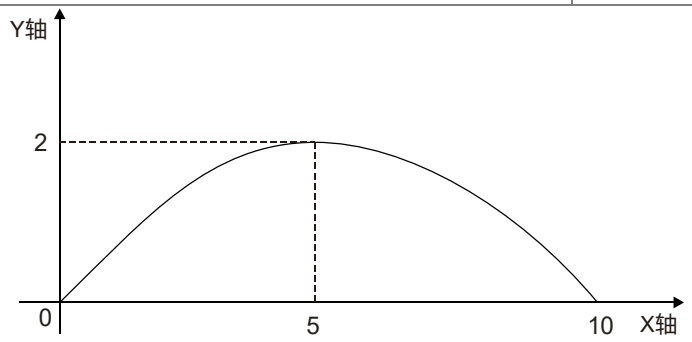
  N_：G 代码在 NC 程序中所在行的编号。

程序范例

画出一个 sin 曲线的一部分

N00 G52	N01 G1 X0 Y0 E5 E-5 F5
N02 G1 X0.1 Y0.06282151816	N03 G1 X0.2 Y0.1255810391
N04 G1 X0.3 Y0.1882166266	N05 G1 X0.4 Y0.2506664671
N06 G1 X0.5 Y0.3128689301	N07 G1 X0.6 Y0.3747626292
N08 G1 X0.7 Y0.4362864828	N09 G1 X0.8 Y0.4973797743
N10 G1 X0.9 Y0.5579822121	N11 G1 X1 Y0.6180339887
N12 G1 X1.1 Y0.6774758405	N13 G1 X1.2 Y0.7362491054
N14 G1 X1.3 Y0.7942957813	N15 G1 X1.4 Y0.8515585831
N16 G1 X1.5 Y0.9079809995	N17 G1 X1.6 Y0.9635073482
N18 G1 X1.7 Y1.018082832	N19 G1 X1.8 Y1.07165359
N20 G1 X1.9 Y1.124166756	N21 G1 X2 Y1.175570505
N22 G1 X2.1 Y1.225814107	N23 G1 X2.2 Y1.274847979
N24 G1 X2.3 Y1.322623731	N25 G1 X2.4 Y1.369094212
N26 G1 X2.5 Y1.414213562	N27 G1 X2.6 Y1.457937255
N28 G1 X2.7 Y1.500222139	N29 G1 X2.8 Y1.541026486
N30 G1 X2.9 Y1.580310025	N31 G1 X3 Y1.618033989
N32 G1 X3.1 Y1.654161149	N33 G1 X3.2 Y1.688655851
N34 G1 X3.3 Y1.721484054	N35 G1 X3.4 Y1.75261336
N36 G1 X3.5 Y1.782013048	N37 G1 X3.6 Y1.809654105
N38 G1 X3.7 Y1.835509251	N39 G1 X3.8 Y1.859552972
N40 G1 X3.9 Y1.881761538	N41 G1 X4 Y1.902113033
N42 G1 X4.1 Y1.920587371	N43 G1 X4.2 Y1.937166322
N44 G1 X4.3 Y1.951833524	N45 G1 X4.4 Y1.964574501
N46 G1 X4.5 Y1.975376681	N47 G1 X4.6 Y1.984229403
N48 G1 X4.7 Y1.991123929	N49 G1 X4.8 Y1.996053457
N50 G1 X4.9 Y1.999013121	N51 G1 X5 Y2
N52 G1 X5.1 Y1.999013121	N53 G1 X5.2 Y1.996053457
N54 G1 X5.3 Y1.991123929	N55 G1 X5.4 Y1.984229403
N56 G1 X5.5 Y1.975376681	N57 G1 X5.6 Y1.964574501
N58 G1 X5.7 Y1.951833524	N59 G1 X5.8 Y1.937166322
N60 G1 X5.9 Y1.920587371	N61 G1 X6 Y1.902113033

N62 G1 X6.1 Y1.881761538	N63 G1 X6.2 Y1.859552972
N64 G1 X6.3 Y1.835509251	N65 G1 X6.4 Y1.809654105
N66 G1 X6.5 Y1.782013048	N67 G1 X6.6 Y1.75261336
N68 G1 X6.7 Y1.721484054	N69 G1 X6.8 Y1.688655851
N70 G1 X6.9 Y1.654161149	N71 G1 X7 Y1.618033989
N72 G1 X7.1 Y1.580310025	N73 G1 X7.2 Y1.541026486
N74 G1 X7.3 Y1.500222139	N75 G1 X7.4 Y1.457937255
N76 G1 X7.5 Y1.414213562	N77 G1 X7.6 Y1.369094212
N78 G1 X7.7 Y1.322623731	N79 G1 X7.8 Y1.274847979
N80 G1 X7.9 Y1.225814107	N81 G1 X8 Y1.175570505
N82 G1 X8.1 Y1.124166756	N83 G1 X8.2 Y1.07165359
N84 G1 X8.3 Y1.018082832	N85 G1 X8.4 Y0.9635073482
N86 G1 X8.5 Y0.9079809995	N87 G1 X8.6 Y0.8515585831
N88 G1 X8.7 Y0.7942957813	N89 G1 X8.8 Y0.7362491054
N90 G1 X8.9 Y0.6774758405	N91 G1 X9 Y0.6180339887
N92 G1 X9.1 Y0.5579822121	N93 G1 X9.2 Y0.4973797743
N94 G1 X9.3 Y0.4362864828	N95 G1 X9.4 Y0.3747626292
N96 G1 X9.5 Y0.3128689301	N97 G1 X9.6 Y0.2506664671
N98 G1 X9.7 Y0.1882166266	N99 G1 X9.8 Y0.1255810391
N100 G1 X9.9 Y0.06282151816	N101 G1 X10 Y0



11.6.4.12 M 代码

- 功能：与一般程序进行交互使用。
- 格式：N_ M_ D_
- 参数说明：

  N_：M 代码在 NC 程序中所在行的编号。

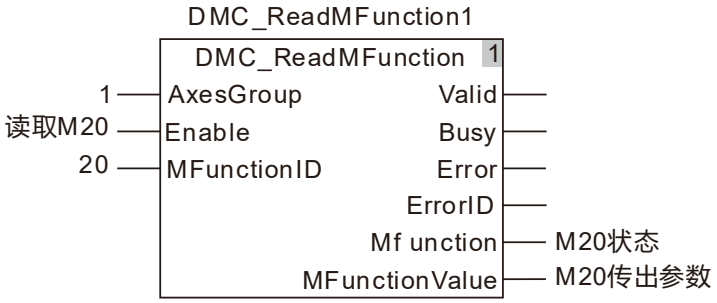
  M_：M 代码的编号，范围为 0~99。

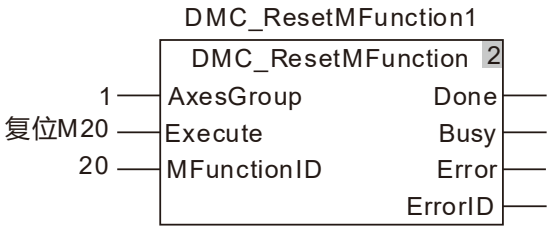
  D_：传出的参数，数据类型 LREAL。
- 指令说明：

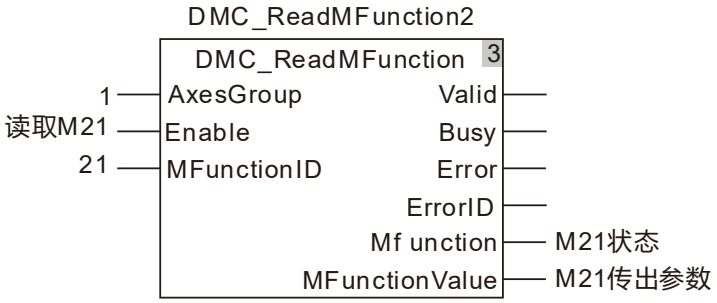
  1. D 分量可以被省略，执行 M 代码时就不会传出参数。
  2. M 代码有两种使用方式：一种是和 G 代码不写在同一行，一种是和 G 代码写在同一行。
  3. 当 M 代码和 G 代码不在同一行，例如：N0 M10 D10.02；N1 G1 X100 Y100。当执行到 N0 这一行时，G 代码执行会停下，与此同时，使用 DMC_ReadMFunction 指令读取 M 代码状态，输出位 MFuntion 置为 TRUE，并且 MFunctionValue 的值为 10.02。当使用 DMC_ResetMFunction 指令复位 M 代码后，G 代码才会继续向下执行 N1。
  4. 当 M 代码和 G 代码在同一行，并且只能放在 G 代码之后。例如：N0 G1 X100 M10 D10.02。当执行到这一行时，会执行 G1，同时使用 DMC_ReadMFunction 指令读取 M 代码状态，输出位 MFuntion 置为 TRUE，并且 MFunctionValue 的值为 10.02。G1 执行完成后则会继续向下执行，不需要使用指令复位 M 代码。

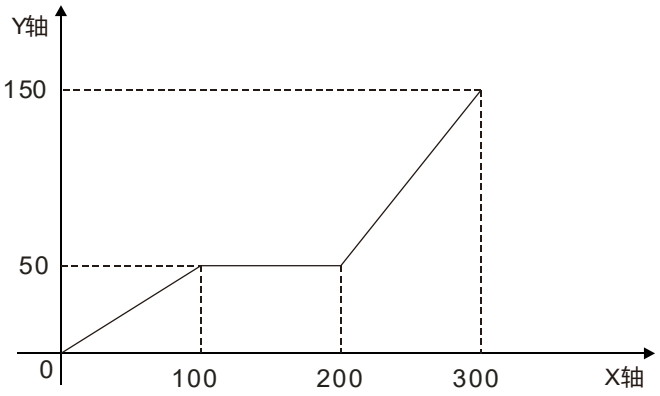
程序范例

```
N00 G1 X100 Y50
N01 M20
N02 G1 X200 Y50 M21 D3.14
N03 G1 X300 Y150
```







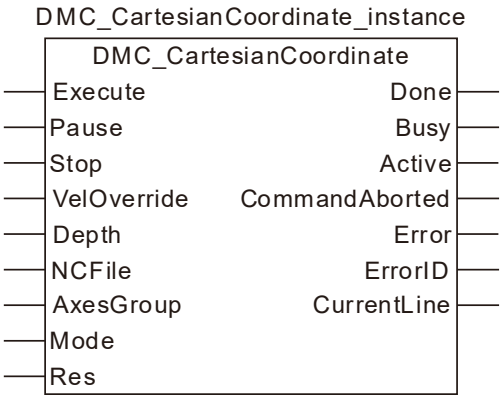


当两个变量“读取 M20”和“读取 M21”为 TRUE 时，开始执行 G 代码，终端机构运行到(100，50)的位置停下，此时变量“M20 状态”为 TRUE，等待其它动作执行完成后，再将变量“复位 M20”为 TRUE，复位 M 代码。此时 G 代码继续运行，终端机构开始运行到(200，50)，同时变量“M21 状态”为 TRUE，变量“M21 传出参数”值为 3.14，终端机构此时不会停止，到达(200，50)后，继续运行到(300，150)，执行完成。

11.6.5 G 代码指令介绍

11.6.5.1 DMC_CartesianCoordinate（直角坐标机器人指令）

FB/FC	说明	适用机种
FB	此指令用于控制直角坐标机器人按照 G 代码进行插补运动	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
Pause（暂停）	当 Pause 由 FALSE 变为 TRUE 时，暂停直角坐标机器人执行 G 代码。	BOOL	TRUE 或 FALSE（FALSE）	
Stop（停止）	当 Stop 由 FALSE 变为 TRUE 时，终止直角坐标机器人执行 G 代码	BOOL	TRUE 或 FALSE（FALSE）	
VelOverride（速度超调值）	速度超调值（%）	LREAL	0~500（0）	Execute 由 FALSE 变为 TRUE
Depth（深度）	填写1，内部保留	UINT	1	Execute 由 FALSE 变为 TRUE
NCFile（NC 文件）	NC文件编号	UINT	1~64（不可缺省）	Execute 由 FALSE 变为 TRUE
AxesGroup（轴组）	轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE
Mode（模式）	填写0，内部保留	INT	0	*Execute* 由 FALSE 变为 TRUE
Res	保留			

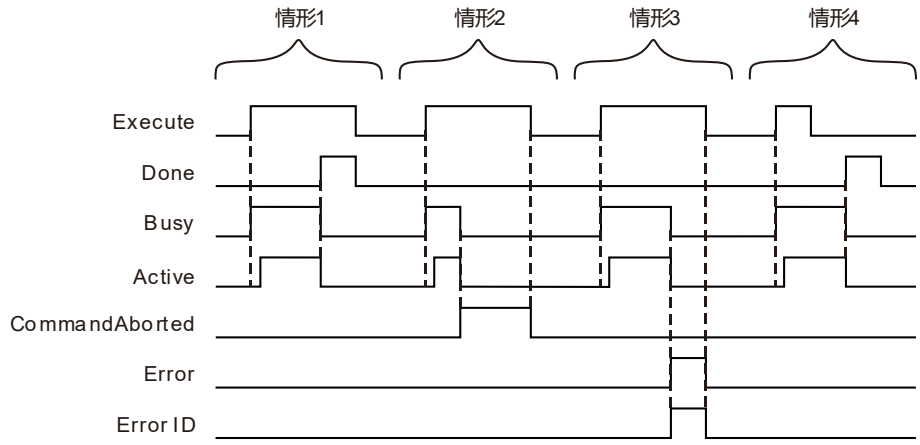
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	
CurrentLine （当前执行 G 代码的行数）	当前执行 G 代码的行数	UDINT	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当 G 代码执行完成时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 时，在指令执行完成时，Done 变为 TRUE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAborted 为 TRUE 时
CommandAborted	◆ 当指令的执行被其它运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当指令执行过程中，Execute 由 TRUE 变为 FALSE 后，该指令被其它指令中断时，CommandAborted 被设置为 TRUE，且一个周期后，CommandAborted 位变为 FALSE
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Active* 变为 TRUE；当定位完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 时，该指令被 MC_Stop 或 MC_Halt 指令中断时，*CommandAborted* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*CommandAborted* 变为 FALSE。

情形3： 当 *Execute* 由 FALSE 变为 TRUE 时，当有错误产生时， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 Busy 和 Active 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

情形4： 在指令执行过程中， *Execute* 由 TRUE 变为 FALSE 后， 指令执行完成时，*Done* 变为 TRUE，同时 *Busy* 和 *Active* 变为 FALSE，且一个周期后，*Done* 变为 FALSE。

- **功能说明**

此指令用于控制直角坐标机器人按照 G 代码进行插补运动，适用于直角雕刻机、缝纫机等数学模型为直角机器人的机构。V1.01 及以上固件版本支持该功能。

1. Pause 引脚用于暂停 G 代码的执行，置 TRUE 后，终端机构将按照设置的减速度减速到 0。当暂停结束，置 Pause 引脚为 FALSE，终端机构将按照设置的加速度开始加速到设置的速度，继续 G 代码插补运动。
2. Stop 引脚用于停止 G 代码的执行，置 TRUE 后，终端机构将立即停下，同时指令的 Done 位变为 TRUE，表示解析完成。
3. VelOverride 引脚用于变更终端速度，输入范围为 0~500，单位为%，“100”表示“100%”。变更后的终端速度 = 变更前的终端速度 × 速度超调值。相对变更后的目标速度，轴会根据当前执行的 G 代码的加速度和减速度，加速或减速到变更后的目标速度。
4. NCFile 引脚用于指定执行 NC 文件编号，该编号即为编程软件中建立的 CNC 文件的 ID 号。
5. AxesGroup 引脚用于指定运行 G 代码的轴组编号。
6. 使用本指令前轴组内各轴必须处于 StandStill 状态，否则指令执行报错。
7. 在该指令控制轴运动前，请先将各个单轴通过单轴指令调整到 G 代码开始执行的位置，其次通过 DMC_AddAxisToGroup 指令把各个单轴加入到轴组中，然后通过 DMC_SetG0Para 和 DMC_SetG1Para 指令设置 G0 和 G1/G2/G3 相关运行参数，最后执行本指令来控制各轴按照 G 代码轨迹插补运动。

程序范例

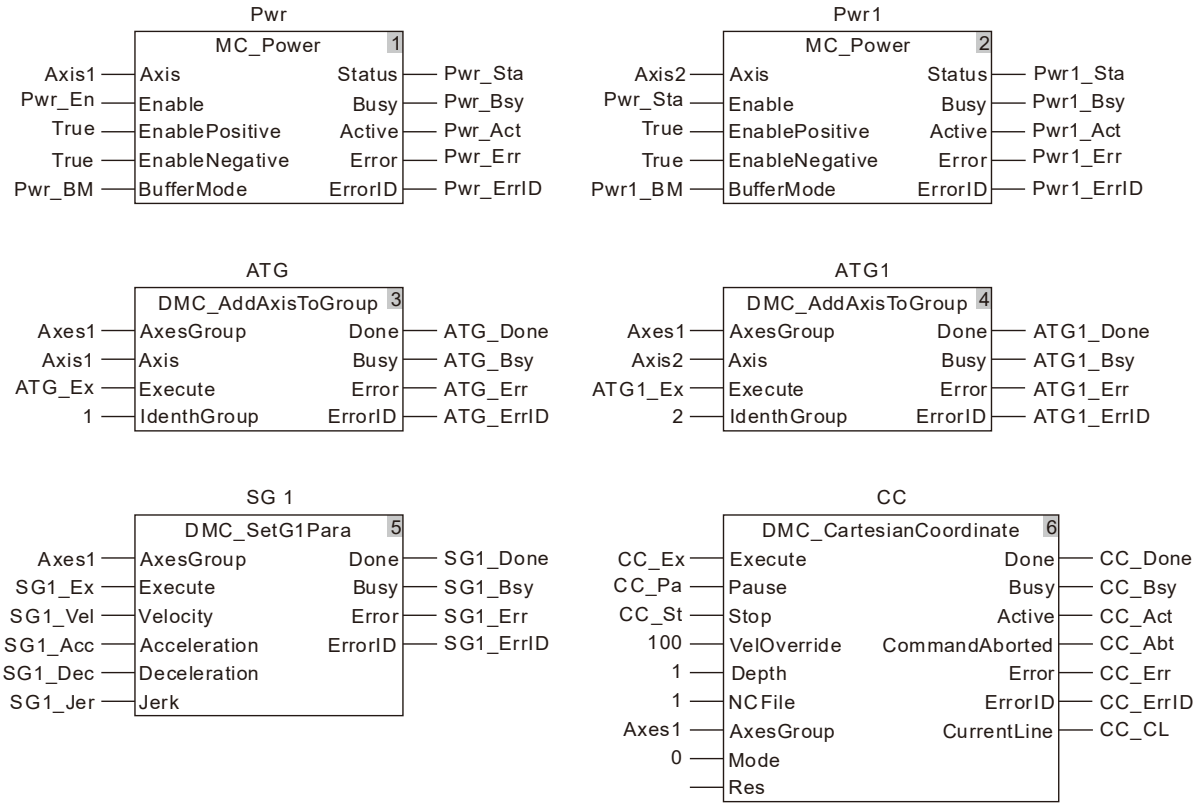
1. 变量和程序

变量名	数据类型	初始值
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Pwr1	MC_Power	
Axis2	USINT	2
Pwr1_BM	MC_Buffer_Mode	0
Pwr1_Sta	BOOL	
Pwr1_Bsy	BOOL	
Pwr1_Act	BOOL	
Pwr1_Err	BOOL	
Pwr1_ErrID	WORD	
ATG	DMC_AddAxisToGroup	
Axes1	USINT	1
ATG_Ex	BOOL	FALSE
ATG_Done	BOOL	
ATG_Bsy	BOOL	
ATG_Err	BOOL	
ATG_ErrID	WORD	
ATG1	DMC_AddAxisToGroup	
ATG1_Ex	BOOL	FALSE
ATG1_Done	BOOL	
ATG1_Bsy	BOOL	
ATG1_Err	BOOL	
ATG1_ErrID	WORD	
SG1	DMC_SetG1Para	
SG1_AG	USINT	1
SG1_Ex	BOOL	FALSE
SG1_Vel	LREAL	10000
SG1_Acc	LREAL	5000
SG1_Dec	LREAL	5000
SG1_Jer	LREAL	5000
SG1_Done	BOOL	
SG1_Bsy	BOOL	
SG1_Err	BOOL	

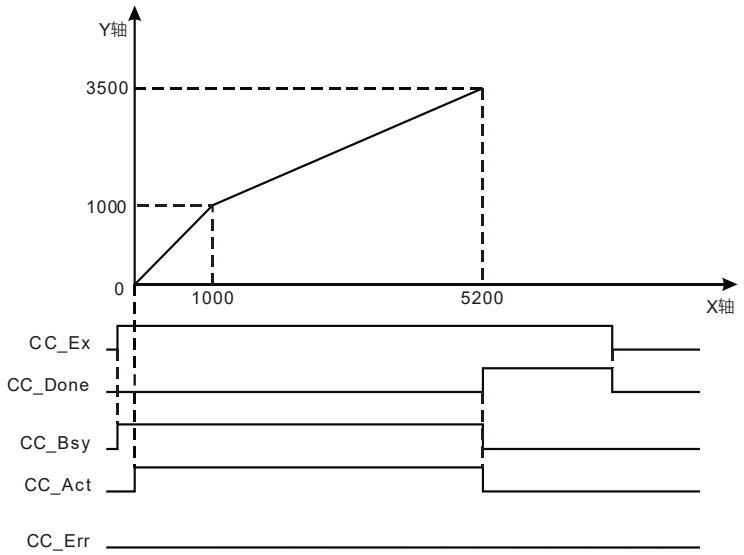
变量名	数据类型	初始值
SG1_ErrID	WORD	
CC	DMC_CartesianCoordinate	
CC_Ex	BOOL	FALSE
CC_Pa	BOOL	FALSE
CC_St	BOOL	FALSE
CC_Done	BOOL	
CC_Bsy	BOOL	
CC_Act	BOOL	
CC_Abt	BOOL	
CC_Err	BOOL	
CC_ErrID	WORD	
CC_CL	UDINT	

G 代码：

N00 G1 X1000 Y1000
N01 G1 X5200 Y3500



2. 运动曲线



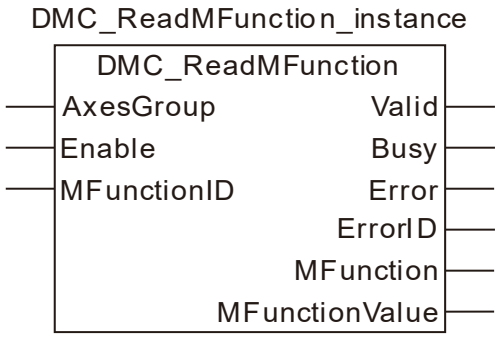
❖ 先将 Pwr_En 置 TRUE，执行 MC_Power 指令使能两个轴。使能完成后，再将 ATG_Ex 和 ATG1_Ex 置 TRUE，执行 DMC_AddAxisToGroup 指令，将 Axis1 和 Axis2 加入到 Axes1 轴组中。加入完成后，将 SG1_Ex 置 TRUE，执行 DMC_SetG1Para 指令，设置 G1/G2/G3 的默认速度。最后，将 CC_Ex 置 TRUE，执行 DMC_ CartesianCoordinate 指令控制 1 号轴和 2 号轴按照 G 代码规划的轨迹进行插补运动。

❖ 当 CC_Ex 由 FALSE 变为 TRUE 时，执行 DMC_CartesianCoordinate 指令，同一个周期 CC_Bsy 由 FALSE 变为 TRUE，第二个周期 CC_Act 由 FALSE 变为 TRUE，机器人按照 G 代码参数运动，当 G 代码执行完成后，CC_Done 位由 FALSE 变为 TRUE,同时 CC_Bsy 和 CC_Act 由 TRUE 变为 FALSE。

❖ 当 CC_Ex 由 TRUE 变为 FALSE 时，CC_Done 由 TRUE 变为 FALSE。

11.6.5.2 DMC_ReadMFunction（读取 M 代码状态指令）

FB/FC	说明	适用机种
FB	此指令用于读取 M 代码状态及其传出参数值。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组）	轴组编号	USINT	1~8（不可缺省）	*Enable* 为 TRUE 时
Enable（执行位）	当 *Enable* 为 TRUE 时，指令执行。	BOOL	TRUE 或 FALSE（FALSE）	-
MFunctionID（M 代码编号）	M 代码编号。	USINT	0~99 (不可缺省)	*Enable 为 TRUE 时*

- **输出参数**

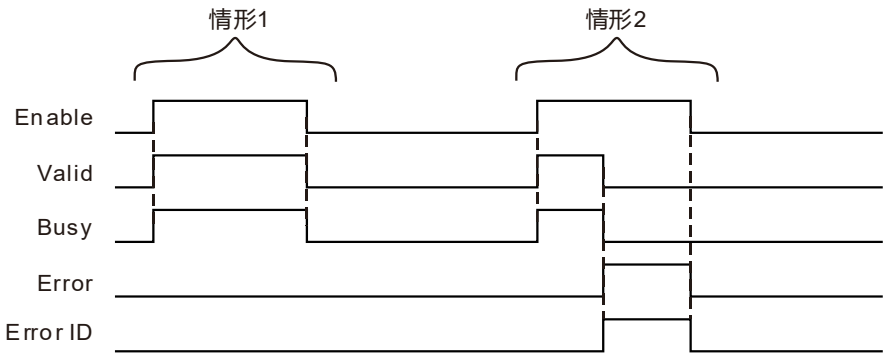
名称	功能	数据类型	输出范围
Valid（输出有效）	该输出参数为 TRUE 时表示指令的输出有效	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-
MFunction（M 代码状态）	该输出参数为 TRUE 时表示 G 代码执行到该 M 代码处	BOOL	TRUE / FALSE
MFunctionValue（M 代码参数值）	输出参数 MFunction 为 TRUE 时，该引脚输出 M 代码参数值	LREAL	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当指令读取到 M 代码状态	◆ 当 *Enable* 由 TRUE 变为 FALSE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
	时	
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 *Valid* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 *Enable* 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Enable* 由 FALSE 变为 TRUE 时，*Valid* 和 *Busy* 同时变为 TRUE。当 *Enable* 变为 FALSE 时，*Valid*、*Busy* 均变为 FALSE。

情形2： 当有错误产生时，*Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 和 *Valid* 变为 FALSE；当 *Enable* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE，*ErrorID* 的值会被清除。

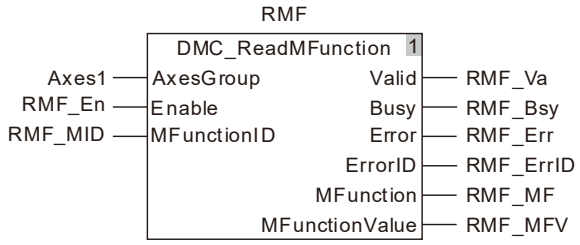
- **功能说明**

此指令用于读取 M 代码状态及其传出参数值，当 G 代码执行到指令设定的 M 代码处时，引脚 MFunction 变为 TRUE，同时 MFunctionValue 输出 M 代码其后的参数值。V1.01 及以上固件版本支持该功能。

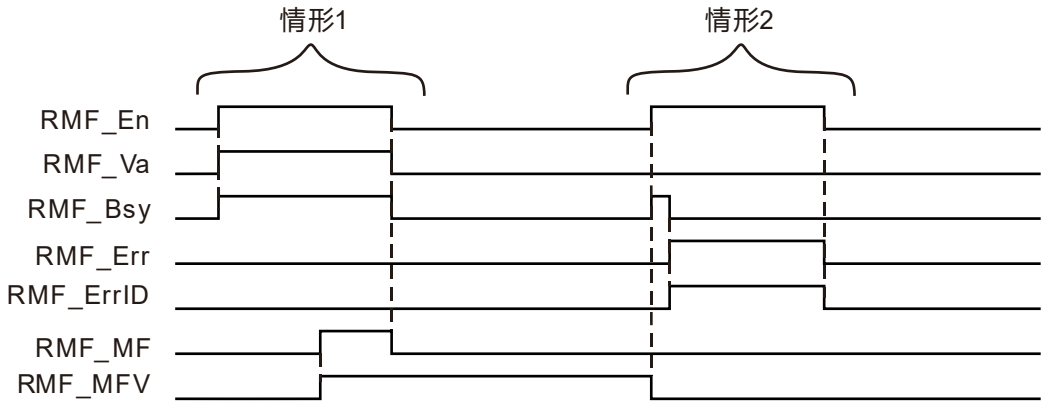
程序范例

1. 变量和程序

变量名	数据类型	初始值
RMF	DMC_ReadMFunction	
Axes1	USINT	1
RMF_En	BOOL	FALSE
RMF_MID	USINT	0
RMF_Va	BOOL	
RMF_Bsy	BOOL	
RMF_Err	BOOL	
RMF_ErrID	WORD	
RMF_MF	BOOL	
RMF_MFV	LREAL	



2. 指令时序图

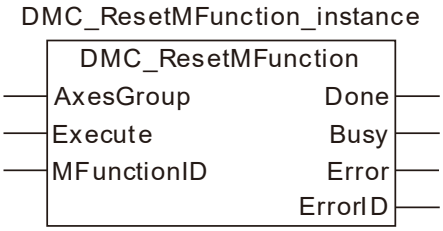


❖ 当 RMF_En 由 FALSE 变为 TRUE 时，执行 DMC_ReadMFunction 指令，第一个周期 RMF_Va 和 RMF_Bsy 由 FALSE 变为 TRUE，当 G 代码执行到指令设定的 M 代码处时，RMF_MF 由 FALSE 变为 TRUE，同时 RMF_MFV 输出 M 代码参数值；当 RMF_En 由 TRUE 变为 FALSE 时，同一个周期 RMF_Va、RMF_Bsy、RMF_MF 由 TRUE 变为 FALSE，RMF_MFV 的 M 代码参数值不会清除。

❖ 当 RMF_En 由 FALSE 变为 TRUE 时，指令参数填写错误，第一个周期 RMF_Bsy 由 FALSE 变为 TRUE，同时清除指令上次执行的 M 代码参数值，第二个周期 RMF_Err 由 FALSE 变为 TRUE，同时 RMF_Bsy 由 TRUE 变为 FALSE,RMF_ErrID 输出对应错误码；当 RMF_En 由 TRUE 变为 FALSE 时，RMF_Err 由 TRUE 变为 FALSE,同时 RMF_ErrID 错误码清除。

11.6.5.3 DMC_ResetMFunction（复位 M 代码状态指令）

FB/FC	说明	适用机种
FB	此指令用于复位 M 代码。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup （轴组）	轴组编号	USINT	1~8 （不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute （执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE （FALSE）	-
MFunctionID （M 代码编号）	M 代码编号。	USINT	0~99 (不可缺省)	*Execute* 由 FALSE 变为 TRUE

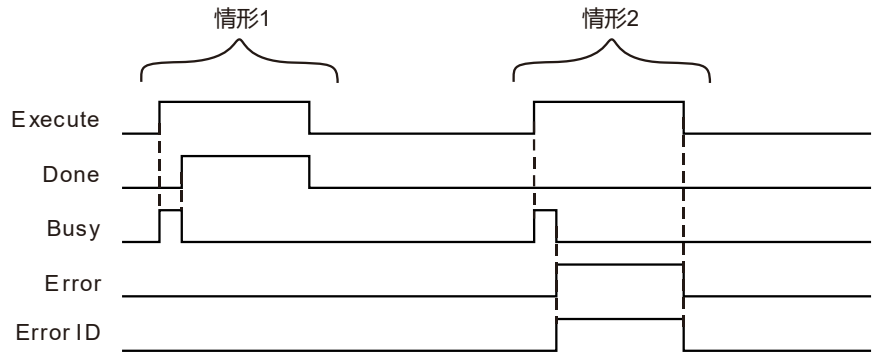
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当复位 M 代码完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 *Done* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，同时 *Busy* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 时，如有错误产生， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

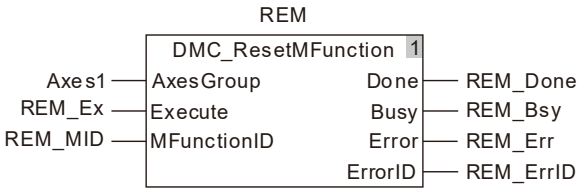
- **功能说明**

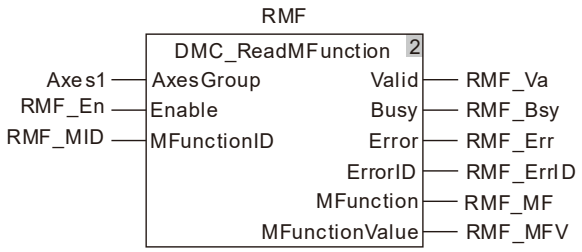
此指令用于复位 M 代码状态。当 G 代码执行到指令设定的 M 代码处时，M 代码输出状态是 TRUE，执行 DMC_ResetMFunction 指令，将 M 代码输出状态复位为 FALSE，继续 G 代码执行。V1.01 及以上固件版本支持该功能。

程序范例

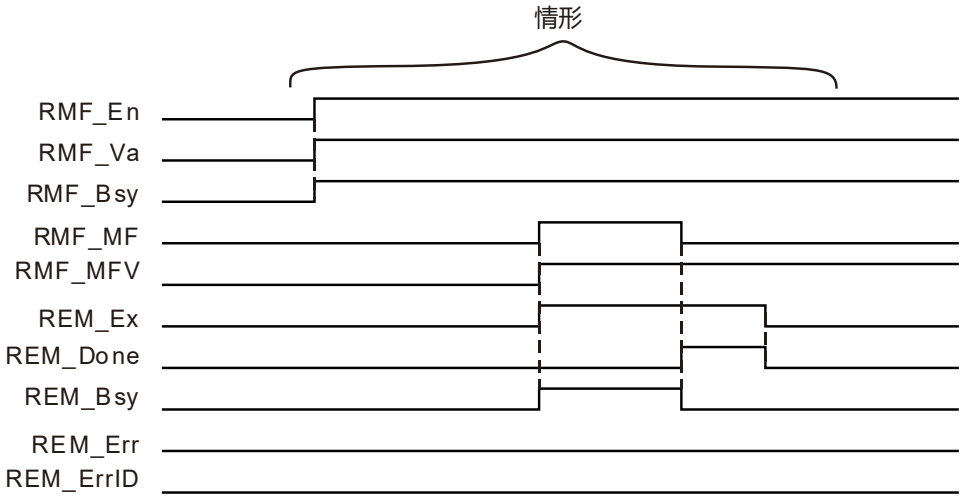
1. 变量和程序

变量名	数据类型	初始值
RMF	DMC_ReadMFunction	
Axes1	USINT	1
RMF_En	BOOL	FALSE
RMF_MID	USINT	0
RMF_Va	BOOL	
RMF_Bsy	BOOL	
RMF_Err	BOOL	
RMF_ErrID	WORD	
RMF_MF	BOOL	
RMF_MFV	LREAL	
REM	DMC_ResetMFunction	
REM_Ex	BOOL	FALSE
REM_MID	USINT	0
REM_Done	BOOL	
REM_Bsy	BOOL	
REM_Err	BOOL	
REM_ErrID	WORD	





2. 指令时序图

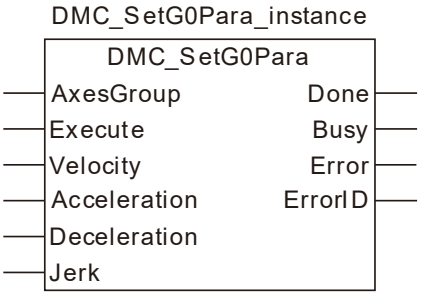


❖ 当 RMF_En 由 FALSE 变为 TRUE 时，执行 DMC_ReadMFunction 指令，同一个周期 RMF_Va 和 RMF_Bsy 由 FALSE 变为 TRUE，当 G 代码执行到指令设定的 M 代码处时，RMF_MF 由 FALSE 变为 TRUE，同时 RMF_MFV 输出 M 代码参数值；

❖ 当 REM_Ex 由 FALSE 变为 TRUE 时，执行 DMC_ResetMFunction 指令，同一个周期 REM_Bsy 由 FALSE 变为 TRUE，第二个周期 REM_Done 由 FALSE 变为 TRUE，同时 指令 DMC_ReadMFunction 引脚 RMF_MF 由 TRUE 变为 FALSE，引脚 REM_Bsy 由 TRUE 变为 FALSE。当 REM_Ex 由 TRUE 变为 FALSE 时，同一个周期 REM_Done 由 TRUE 变为 FALSE.

11.6.5.4 DMC_SetG0Para（设置 G0 参数指令）

FB/FC	说明	适用机种
FB	此指令用于设置 G0 速度、加速度、减速度、加速度和减速度变化率。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组）	轴组编号	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
Velocity（速度）	设定的目标速度（单位: 单元/秒）	ARRAY [1..8] OF LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Acceleration（加速度）	设定的目标加速度（单位: 单元/秒 ^{2}）	ARRAY [1..8] OF LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration（减速度）	设定的目标减速度（单位: 单元/秒 ^{2}）	ARRAY [1..8] OF LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk（加速度的变化率）	设定的目标加速度或减速度的变化率（单位: 单元/秒 ^{3}）	ARRAY [1..8] OF LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE

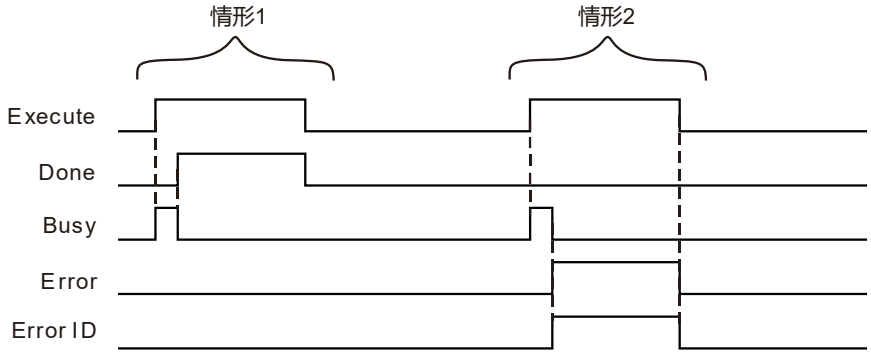
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当设定完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 *Done* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，同时 *Busy* 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，如有错误产生， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

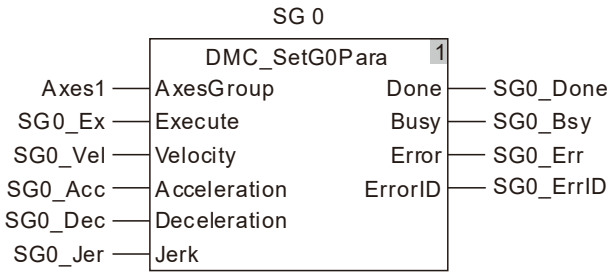
- **功能说明**

此指令用于设定 G0 速度、加速度、减速度、加速度和减速度变化率。当 G 代码执行 G0 时，速度、加速度、减速度、加速度和减速度变化率会按照 DMC_SetG0Para 指令设定的参数执行。V1.01 及以上固件版本支持该功能。

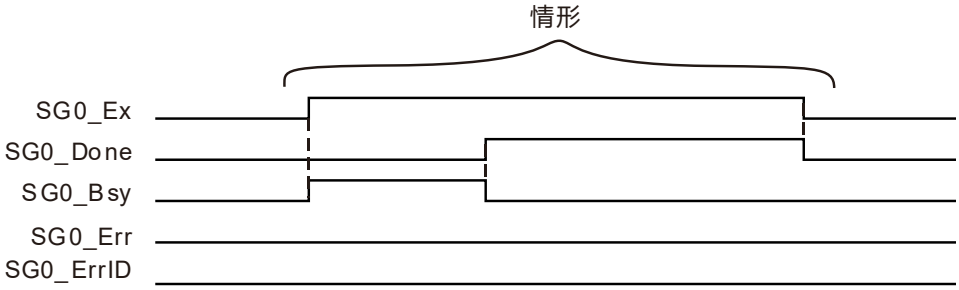
程序范例

1. 变量和程序

变量名	数据类型	初始值
SG0	DMC_SetG0Para	
Axes1	USINT	1
SG0_Ex	BOOL	FALSE
SG0_Vel	ARRAY[1..8]OF LREAL	
SG0_Acc	ARRAY[1..8]OF LREAL	
SG0_Dec	ARRAY[1..8]OF LREAL	
SG0_Jer	ARRAY[1..8]OF LREAL	
SG0_Done	BOOL	
SG0_Bsy	BOOL	
SG0_Err	BOOL	
SG0_ErrID	WORD	



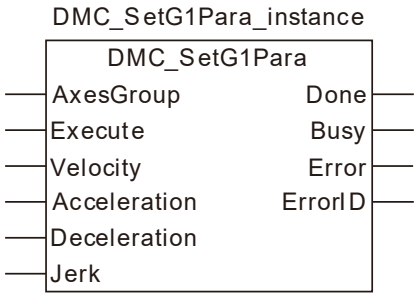
2. 指令时序图



- ❖ 当 SG0_Ex 由 FALSE 变为 TRUE 时，执行 DMC_SetG0Para 指令，同一个周期 SG0_Bsy 由 FALSE 变为 TRUE，第二个周期 SG0_Done 由 FALSE 变为 TRUE，同时 SG0_Bsy 由 TRUE 变为 FALSE，则 G 代码执行 G0 时的速度、加速度、减速度、加速度和减速度变化率令是该指令设定的参数。
- ❖ 当 SG0_Ex 由 TRUE 变为 FALSE 时，SG0_Done 同时由 TRUE 变为 FALSE。

11.6.5.5 DMC_SetG1Para（设置 G1 参数指令）

FB/FC	说明	适用机种
FB	此指令用于设置 G1/G2/G3 的默认速度、加速度、减速度、加速度和减速度变化率。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组）	轴组编号	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
Velocity（速度）	设定的目标速度（单位: 单元/秒）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Acceleration（加速度）	设定的目标加速度（单位: 单元/秒 ^{2}）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Deceleration（减速度）	设定的目标减速度（单位: 单元/秒 ^{2}）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Jerk（加速度的变化率）	设定的目标加速度或减速度的变化率（单位: 单元/秒 ^{3}）	LREAL	正数（不可缺省）	*Execute* 由 FALSE 变为 TRUE

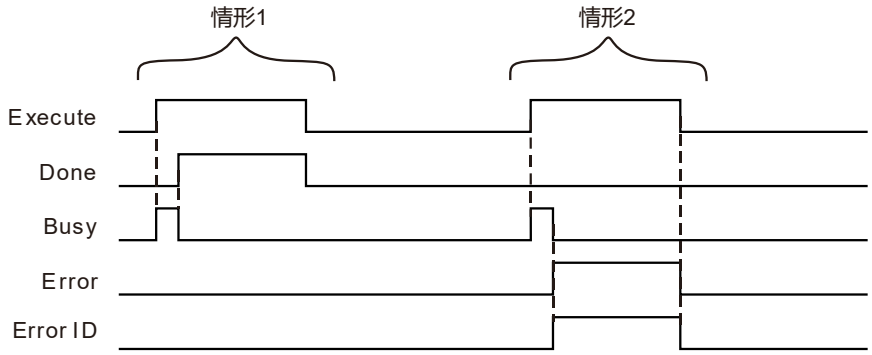
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当设定完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 *Done* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，同时 *Busy* 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，如有错误产生， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

- **功能说明**

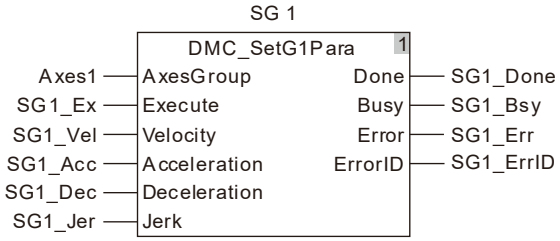
此指令用于设定 G1/G2/G3 的默认速度、加速度、减速度、加速度和减速度变化率。当 G 代码执行 G1/G2/G3 时，如果没有指定 E 和 F 的值，那么 G1/G2/G3 按照此指令设定的速度、加速度、减速度执行，否则按照 G 代码中填写的 E 和 F 进行加减速和运行。V1.01 及以上固件版本支持该功能。

程序范例

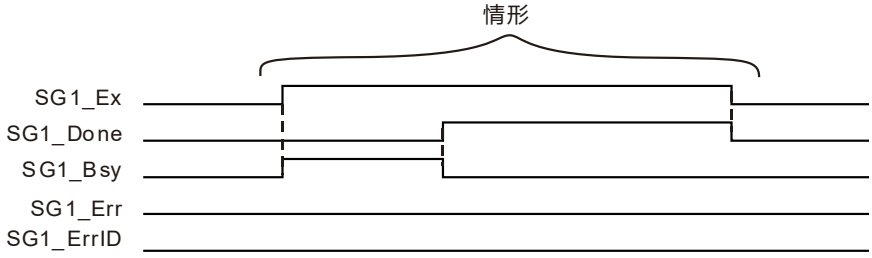
1. 变量和程序

变量名	数据类型	初始值
SG1	DMC_SetG1Para	
Axes1	USINT	1
SG1_Ex	BOOL	FALSE
SG1_Vel	LREAL	5000
SG1_Acc	LREAL	1000
SG1_Dec	LREAL	1000
SG1_Jer	LREAL	1000
SG1_Done	BOOL	
SG1_Bsy	BOOL	
SG1_Err	BOOL	

变量名	数据类型	初始值
SG1_ErrID	WORD	



2. 指令时序图

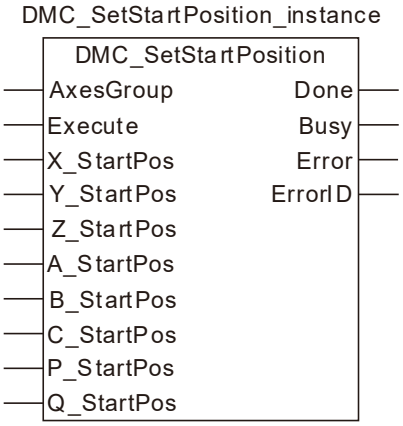


❖ 当 SG1_Ex 由 FALSE 变为 TRUE 时，执行 DMC_SetG1Para 指令，同一个周期 SG1_Bsy 由 FALSE 变为 TRUE，第二个周期 SG1_Done 由 FALSE 变为 TRUE，同时 SG1_Bsy 由 TRUE 变为 FALSE，则 G 代码执行 G1 时的速度、加速度、减速度、加速度和减速度变化率令是该指令设定的参数。

❖ 当 SG1_Ex 由 TRUE 变为 FALSE 时，SG1_Done 同时由 TRUE 变为 FALSE。

11.6.5.6 DMC_SetStartPosition（设置执行 G 代码轴的起始位置指令）

FB/FC	说明	适用机种
FB	此指令用于设置执行 G 代码各分量轴的起始位置	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组）	轴组编号	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 *Execute* 由 FALSE 变为 TRUE 时，指令开始执行。	BOOL	TRUE 或 FALSE（FALSE）	-
X_StartPos（X 轴起始位置）	设定X轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
Y_StartPos（Y 轴起始位置）	设定Y轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
Z_StartPos（Z 轴起始位置）	设定Z轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
A_StartPos（A 轴起始位置）	设定A轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
B_StartPos（B 轴起始位置）	设定B轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
C_StartPos（C 轴起始位置）	设定C轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
P_StartPos（P 轴起始位置）	设定P轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE
Q_StartPos（Q 轴起始位置）	设定Q轴起始位置	LREAL	正数、0、负数（0）	*Execute* 由 FALSE 变为 TRUE

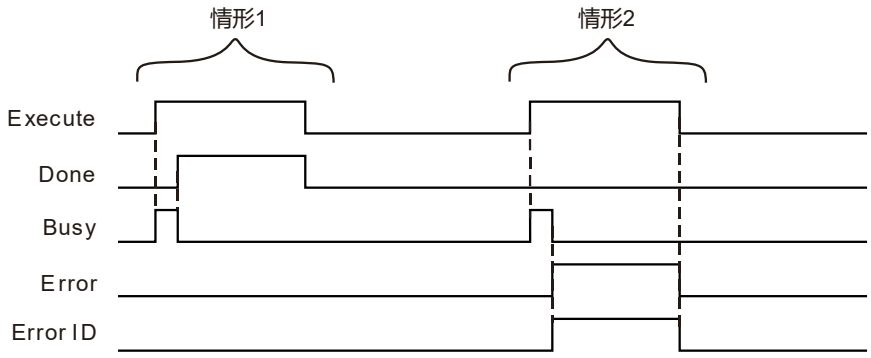
- **输出参数**

名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成.	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	-

- **输出参数刷新时机**

名称	变为 **TRUE** 的时机	变为 **FALSE** 的时机
Done	◆ 当设定完成时.	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 *Done* 为 TRUE 时 ◆ 当 *Error* 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 *Execute* 由 FALSE 变为 TRUE 时，*Busy* 变为 TRUE，且一个周期后，*Done* 变为 TRUE，同时 *Busy* 变为 FALSE。

情形2： 当 *Execute* 由 FALSE 变为 TRUE 时，如有错误产生， *Error* 变为 TRUE，*ErrorID* 显示对应的错误码，同时 *Busy* 变为 FALSE；当 *Execute* 由 TRUE 变为 FALSE 时，*Error* 变为 FALSE。

- **功能说明**

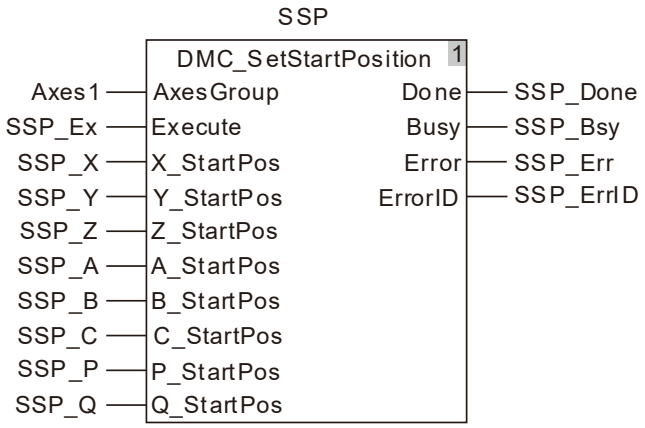
此指令用于设定 G 代码中 8 个轴的起始位置。当执行该指令后，G 代码中 X 轴、Y 轴、Z 轴、A 轴、B 轴、C 轴、P 轴、Q 轴起始位置按照指令设定的位置开始运动，比如设定 X 轴起始位置是 10000，G 代码为 G0 X1000，则执行 G 代码，X 轴从位置 10000 开始，运动到 1000 的位置处。V1.01 及以上固件版本支持该功能。

如果使用了 DMC_CartesianCoordinate 指令来执行 G 代码，则不需要使用本指令来设置起点，DMC_CartesianCoordinate 指令会自动进行起点的设定。

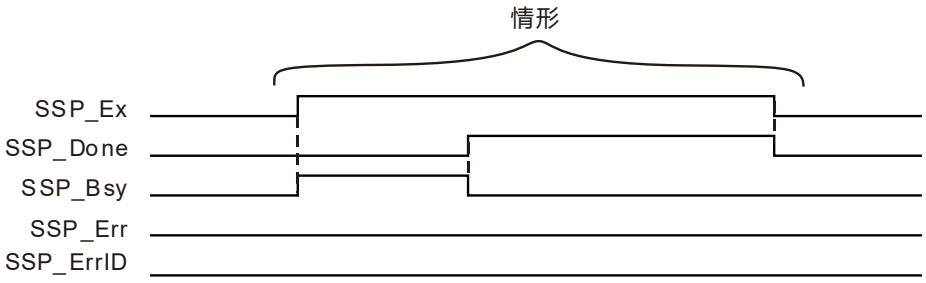
程序范例

1. 变量和程序

变量名	数据类型	初始值
SSP	DMC_SetStartPosition	
Axes1	USINT	1
SSP_Ex	BOOL	FALSE
SSP_X	LREAL	1000
SSP_Y	LREAL	1000
SSP_Z	LREAL	1000
SSP_A	LREAL	1000
SSP_B	LREAL	1000
SSP_C	LREAL	1000
SSP_P	LREAL	1000
SSP_Q	LREAL	1000
SSP_Done	BOOL	
SSP_Bsy	BOOL	
SSP_Err	BOOL	
SSP_ErrID	WORD	



2. 指令时序图



❖ 当 SSP_Ex 由 FALSE 变为 TRUE 时，执行 DMC_SetStartPosition 指令，同一个周期 SSP_Bsy 由 FALSE 变为 TRUE，第二个周期 SSP_Done 由 FALSE 变为 TRUE，同时 SSP_Bsy 由 TRUE 变为 FLASE，则 G 代码中轴执行的起始位置是指令设定的起始位置。

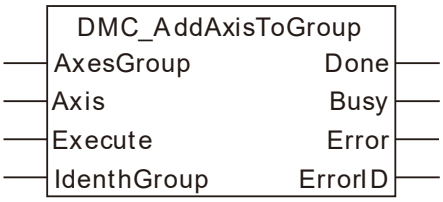
❖ 当 SSP_Ex 由 TRUE 变为 FALSE 时，SSP_Done 同时由 TRUE 变为 FALSE。

11.7 轴组指令

11.7.1 DMC_AddAxisToGroup（添加轴到轴组）

FB/FC	说明	适用机种
FB	此指令用于将轴添加到轴组。	DVP15MC11T DVP15MC11T-06

DMC_AddAxisToGroup_instance



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
AxesGroup （轴组编号）	设定欲操作的轴组编号	USINT	1~8 （不可缺省）	Execute 由 FALSE 变为 TRUE
Axis （轴的站号）	设定欲添加进轴组的轴。	USINT	请参考第 2.2 节**功能简介** （不可缺省）	Execute 由 FALSE 变为 TRUE
Execute （执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	
IdentInGroup	设定该轴在轴组中的轴号。	USINT	1~8 （不可缺省）	Execute 由 FALSE 变为 TRUE

- **输出参数**

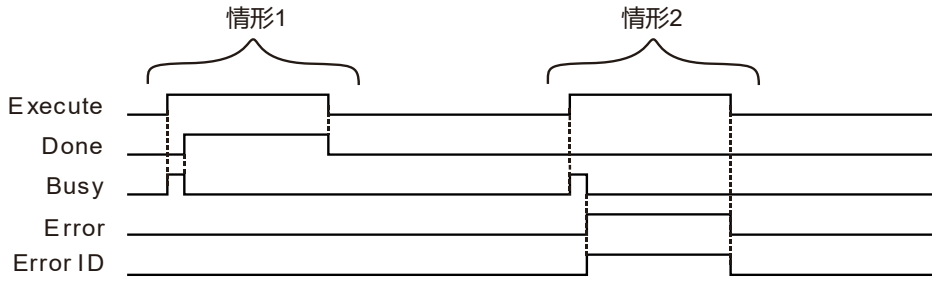
名称	功能	数据类型	输出范围
Done （完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 将轴添加到轴组完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 DONE 为 TRUE 时 ◆ 当 Error 为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且一个周期后，Done 变为 TRUE，Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Done 位同时变为 FALSE。

情形2： 当指令执行出错时，Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 变为 FALSE；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为零。

- **功能说明**

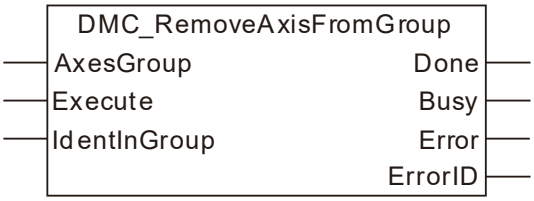
此指令用于将轴添加到轴组并设定该轴在轴组中的轴号。V1.01 及以上固件版本支持该功能。

1. 当指令的 Done 位变为 TRUE 时，表示该轴成功添加到轴组中；将指令的 Execute 设为 FALSE 并不能将轴从轴组中删除，如果需要将该轴从轴组中删除，则需要使用 DMC_RemoveAxisFromGroup 指令，关于 DMC_RemoveAxisFromGroup 指令的说明请参考第 11.7.2 节。
2. IdentInGroup 表示该轴在轴组中的轴号，范围为 1~8。1 代表 X 轴，2 代表 Y 轴，3 代表 Z 轴，4 代表 A 轴，5 代表 B 轴，6 代表 C 轴，7 代表 P 轴，8 代表 Q 轴。
3. 只有在轴组使能之前，才能够执行 DMC_AddAxisToGroup 指令，轴组使能之后再执行该指令，该指令会报错。

11.7.2 DMC_RemoveAxisFromGroup（从轴组中移除轴）

FB/FC	说明	适用机种
FB	此指令用于将轴从轴组中移除。	DVP15MC11T DVP15MC11T-06

DMC_RemoveAxisFromGroup_instance



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲操作的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE 时
Execute（使能位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
IdentInGroup（轴号）	设定欲从轴组中删除的轴	INT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE 时

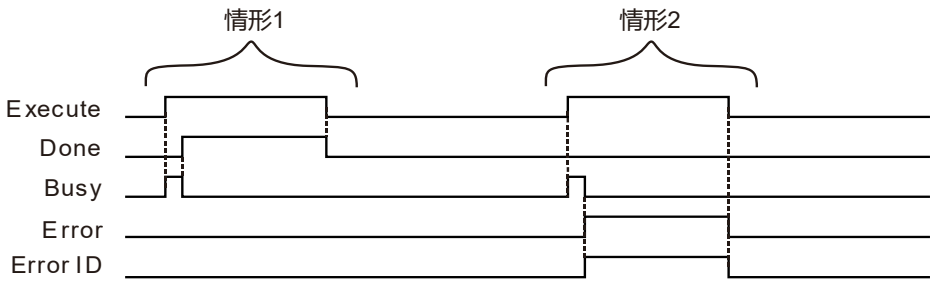
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出 错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当轴从轴组中删除时	◆ 当 Execute 变为 FALSE 时
Busy	◆ 当该指令执行时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 DONE 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且一个周期后，Done 变为 TRUE，Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Done 变为 FALSE。

情形2： 当指令输入参数有误或轴组未去使能时，Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，一个周期后，Error 变为 TRUE，ErrorID 显示错误码，同时 Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 清零。

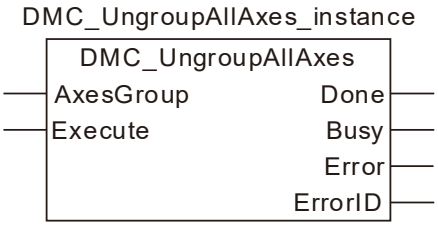
- **功能说明**

此指令用于将轴从轴组中删除，输入参数 IdentInGroup 的有效范围为 1~8，超出该范围指令报错。V1.01 及以上固件版本支持该功能。

当轴组处于使能状态时，使用该指令后，该指令会立即报错。只有当轴组处于未使能的状态时，才能将使用此指令。

11.7.3 DMC_UngroupAllAxes（解除轴组）

FB/FC	说明	适用机种
FB	此指令用于将某个轴组的轴全部移除。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲操作的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE 时
Execute（使能位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	

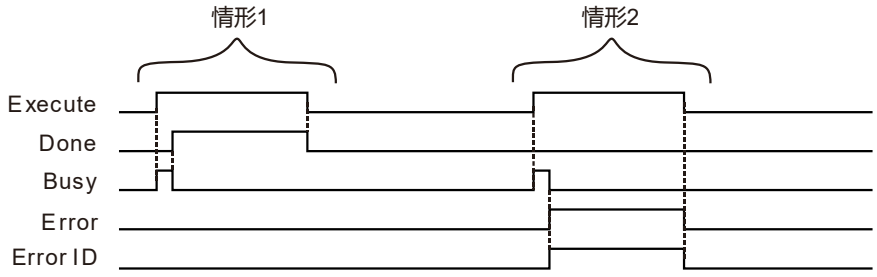
- **输出参数**

名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当轴组解除时	◆ 当 Execute 变为 FALSE 时
Busy	◆ 当该指令执行时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 DONE 为 TRUE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且一个周期后，Done 变为 TRUE，Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Done 变为 FALSE。

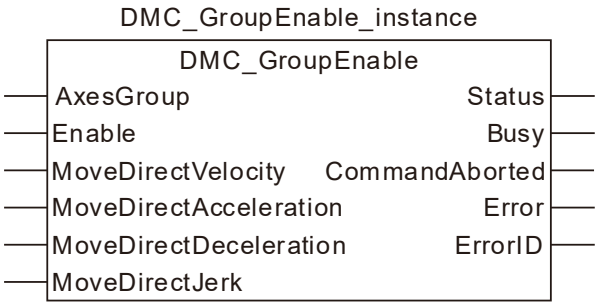
情形2： 当指令输入参数有误或轴组未去使能时，Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，一个周期后，Error 变为 TRUE，ErrorID 显示错误码，同时 Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 清零。

- **功能说明**

此指令用于将轴组中的所有轴删除，当轴组处于使能状态时，使用该指令后，该指令会立即报错。V1.01 及以上固件版本支持该功能。

11.7.4 DMC_GroupEnable（轴组使能）

FB/FC	说明	适用机种
FB	此指令用于将轴组使能。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲使能的轴组编号	USINT	1~8（不可缺省）	Enable 为 TRUE 时
Enable（使能位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
MoveDirectVelocity（速度）	设定轴组快速定位时各轴运行的速度。	ARRAY [1..8] OF LREAL	正数（不可缺省）	Enable 为 TRUE 时
MoveDirectVelocity（速度）	设定轴组快速定位时各轴运行的速度。	ARRAY [1..8] OF LREAL	正数（不可缺省）	Enable 为 TRUE 时
MoveDirectAcceleration	设定轴组快速定位时各轴运行的加速度。	ARRAY [1..8] OF LREAL	正数（不可缺省）	Enable 为 TRUE 时
MoveDirectDeceleration	设定轴组快速定位时各轴运行的减速度。	ARRAY [1..8] OF LREAL	正数（不可缺省）	Enable 为 TRUE 时
MoveDirectJerk	设定轴组快速定位时各轴运行的加速度变化率。	ARRAY [1..8] OF LREAL	正数（不可缺省）	Enable 为 TRUE 时

说明：轴组必须在使能后才能控制轴组作相应动作，当轴组未使能时，快速定位、直线插补和圆弧插补指令不能执行。

- **输出参数**

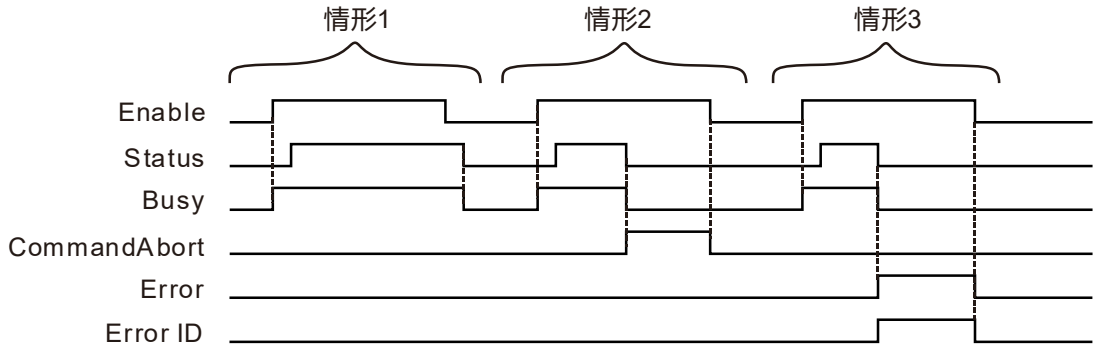
名称	功能	数据类型	输出范围
Status（完成）	该输出参数为 TRUE 时表示轴组已经使能。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
CommandAborted（中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Status	◆ 当指定轴组已经使能时	◆ 当指定轴组解除使能时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当该指令执行时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 为 TRUE 时
CommandAborted	◆ 当指令的执行被中断时	◆ 当 Enable 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 False 变为 TRUE 时，Busy 变为 TRUE，当轴组使能成功后，Status 变为 TRUE；当 Enable 由 TRUE 变为 FALSE 后，轴组解除使能后，Busy 和 Status 由 TRUE 变为 FALSE。

情形2： 当该指令执行过程中被打断时，CommandAborted 变为 TRUE，同时 Status、Busy 变为 FALSE，轴组解除使能；当 Enable 变为 FALSE 时，CommandAborted 同时变为 FALSE。

情形3： 当该指令执行过程中，有错误产生时，Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Status、Busy 变为 FALSE；当 Enable 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

- **功能说明**

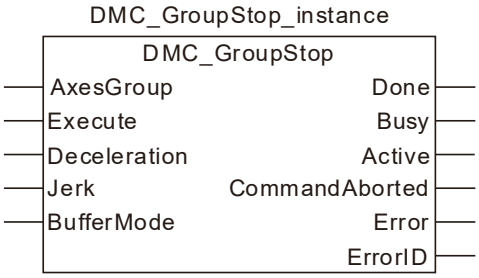
此指令用于对相应轴组进行使能或解除使能。V1.01 及以上固件版本支持该功能。

1. 执行指令 DMC_GroupEnable 时，轴组内所有轴的状态机必须为准备执行状态（StandStill），DMC_GroupEnable 才能正常执行。各个轴须使用 MC_Power 指令将轴使能后进入准备执行状态（StandStill）。

2. 各个轴的状态在准备执行状态（StandStill）时，将 Enable 设为 TRUE 后，Status 也会变为 TRUE。运行轴组时，请务必确认 Status 变为 TRUE 后再开始动作。Status 变为 TRUE 后，轴状态变为离散运动状态（Discrete Motion），需要轴状态在准备执行状态（StandStill）才可以执行的其他运动指令，需要将 Enable 设为 FALSE 后才可以执行。
3. 当 Enable 由 TRUE 变为 FALSE 后，轴组会去使能，轴状态变为准备执行状态（StandStill）。轴组（如 DMC_MoveDirectAbsolute 指令）正在执行时，Enable 由 TRUE 变为 FALSE，轴组指令会报错，轴组中的所有轴会停止，轴状态变为准备执行状态（StandStill）。
4. 轴组指令执行过程中，若执行单轴指令，当单轴指令的 BufferMode 为 0（mcAbroting）时，单轴指令会立即打断轴组指令运行，同时轴组解除使能状态；当单轴指令的 BufferMode 为非 0（1~5）时，则单轴指令不会执行。
5. 执行指令 DMC_GroupEnable 时，轴组内所有轴的状态机必须为 StandStill，DMC_GroupEnable 才能正常执行。

11.7.5 DMC_GroupStop（轴组停止运行）

FB/FC	说明	适用机种
FB	此指令用于停止当前轴组运动。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲使能的轴组编号	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
Deceleration	保留	-	-	-
Jerk	保留	-	-	-
BufferMode（交接模式）	设定两个指令之间交接模式 0：打断	MC_Buffer_Mode	0：mcAborting	Execute 由 FALSE 变为 TRUE

说明：

1. 参数 Deceleration、Jerk 保留，填写数值无效。
2. 参数 BufferMode 仅支持模式 0（mcAborting），不支持其他模式。选择其他模式时，执行该指令，该指令会报错。

- **输出参数**

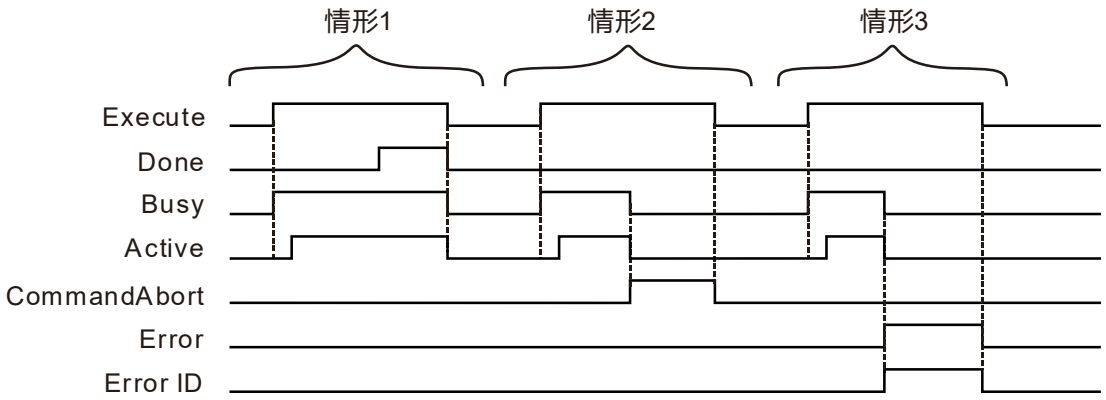
名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active（控制中）	该输出参数为 TRUE 时表示指令正在控制轴组。	BOOL	TRUE / FALSE
CommandAborted（中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE

名称	功能	数据类型	输出范围
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当轴组停止完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 Error 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 CommandAborted 由 FALSE 变为 TRUE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时
Active	◆ 当指令控制轴组时	◆ 当 Execute 由 TRUE 变为 FALSE 时 ◆ 当 CommandAborted 由 FALSE 变为 TRUE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时
CommandAborted	◆ 当指令的执行被中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 False 变为 TRUE 时，Busy 变为 TRUE，一个周期后，Active 变为 TRUE，当停止成功后，Done 变为 TRUE，同时 Busy 与 Active 保持 TRUE 不变；当 Execute 由 TRUE 变为 FALSE 时， Done、Busy 和 Active 位同时变为 FALSE。

情形2： 当该指令执行过程中被打断时，CommandAborted 变为 TRUE，同时 Busy、Active 变为 FALSE；当 Execute 变为 FALSE 时，CommandAborted 同时变为 FALSE。

情形3： 当该指令执行过程中，有错误产生时，Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy、Active 变为 FALSE；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

- **功能说明**

此指令用于停止轴组动作。

1. 指令输入参数 Deceleration、Jerk 功能保留，用户输入数值无效，执行 DMC_GroupStop 指令时，轴组中的各个轴会立即停止。
2. 只有当轴组指令 DMC_MoveDirectAbsolute、DMC_MoveDirectRelative、DMC_MoveLinearAbsolute、DMC_MoveLinearRelative、DMC_MoveCircularAbsolute 与 DMC_MoveCircularRelative 正在执行时，才可以执行 DMC_GroupStop 指令，该指令执行完成后，轴组中各个轴全部停止，轴状态仍为离散运动状态（Discrete Motion）。需要轴状态在准备执行状态（StandStill）才可以执行的其他运动指令，需要将 DMC_ GroupEnable 指令 Enable 设为 FALSE 后才可以执行。
3. DMC_GroupStop 指令 Execute 位由 FALSE 变为 TRUE 时，该指令执行，执行完成后，轴组指令 DMC_MoveDirectAbsolute、DMC_MoveDirectRelative、DMC_MoveLinearAbsolute、DMC_MoveLinearRelative、DMC_MoveCircularAbsolute 与 DMC_MoveCircularRelative 均无法执行，只有当 DMC_GroupStop 指令的执行位 Execute 变为 FALSE 后，上述指令才能执行。

程序范例一

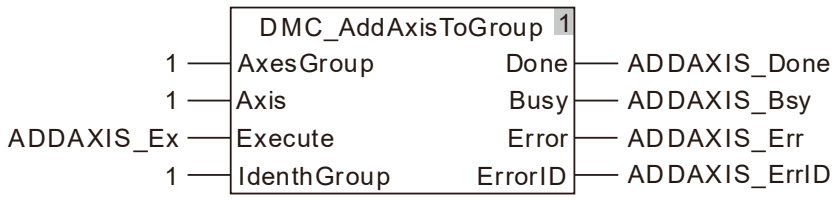
DMC_GroupStop 指令使用范例如下：

1. 变量和程序

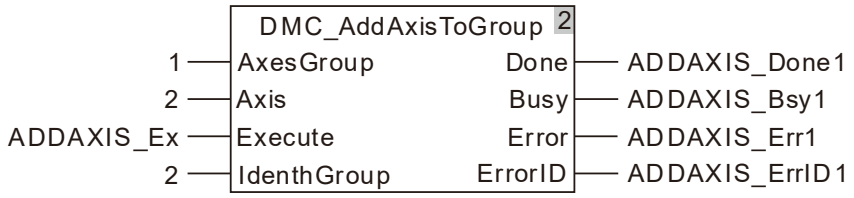
变量名	数据类型	初始值
M1	BOOL	TRUE
DMC_AddAxisToGroup0	DMC_AddAxisToGroup	
ADDAXIS_Ex	BOOL	
ADDAXIS_Done	BOOL	
ADDAXIS_Bsy	BOOL	
ADDAXIS_Err	BOOL	
ADDAXIS_ErrID	WORD	
DMC_AddAxisToGroup1	DMC_AddAxisToGroup	
ADDAXIS_Done1	BOOL	
ADDAXIS_Bsy1	BOOL	
ADDAXIS_Err1	BOOL	
ADDAXIS_ErrID1	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Acc	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Dec	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Jerk	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DMC_MoveLinearAbsolute0	DMC_MoveLinearAbsolute	

变量名	数据类型	初始值
LA_Ex	BOOL	
LA_Pos	ARRAY [1..8] OF LREAL	[200000,200000]
LA_Done	BOOL	
LA_Bsy	BOOL	
LA_Act	BOOL	
LA_Abt	BOOL	
LA_Err	BOOL	
LA_ErrID	WORD	
DMC_GroupStop0	DMC_GroupStop	
GS_Ex	BOOL	
GS_Done	BOOL	
GE_Bsy	BOOL	
GE_Act	BOOL	
GE_Abt	BOOL	
GS_Err	BOOL	
GS_ErrID	WORD	

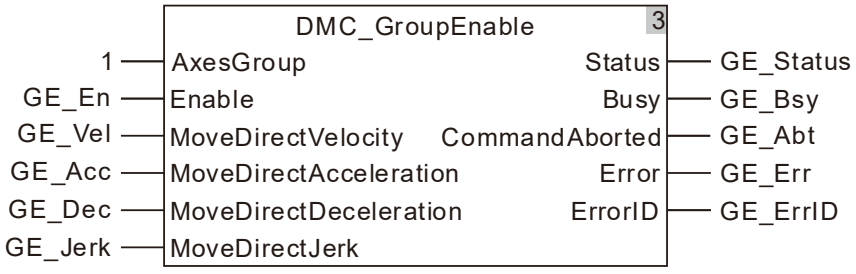
DMC_AddAxisToGroup0

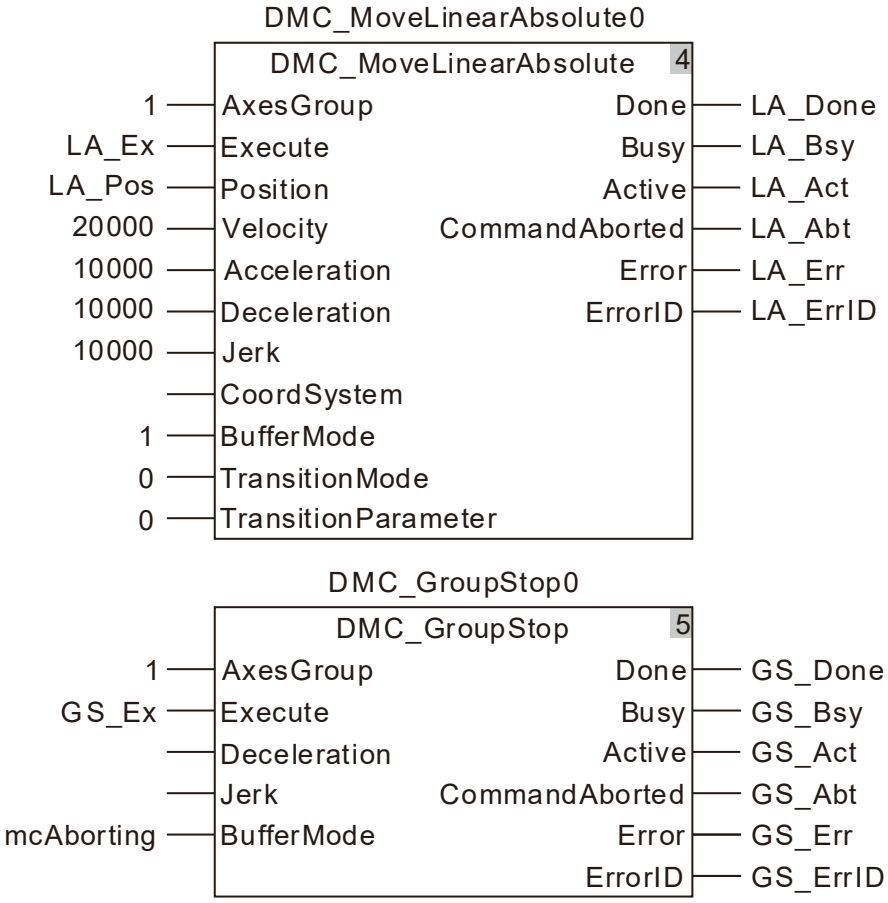


DMC_AddAxisToGroup1

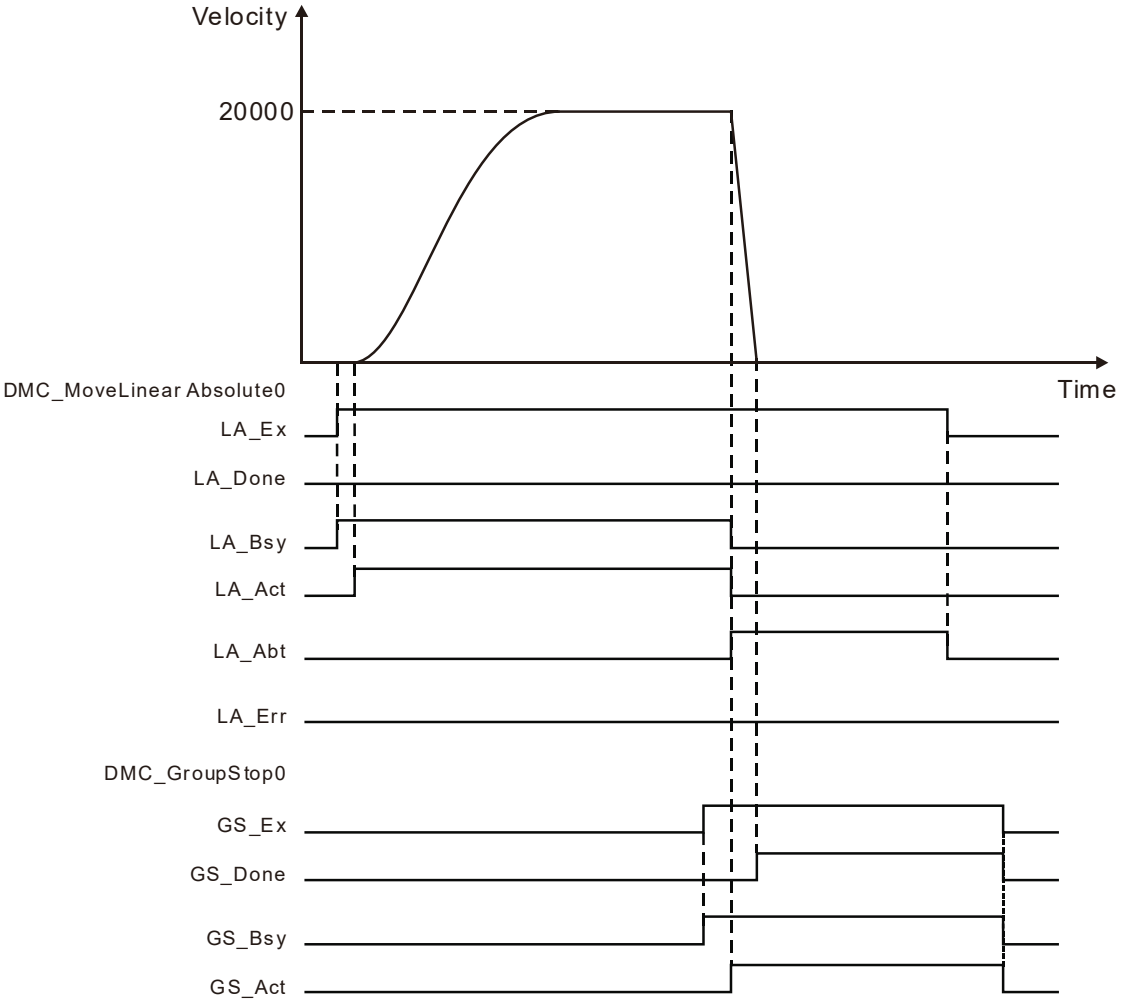


DMC_GroupEnable0




DMC_MoveLinearAbsolute0
DMC_MoveLinearAbsolute
4
1
AxesGroup
LA_Ex
Execute
LA_Pos
Position
20000
Velocity
10000
Acceleration
10000
Deceleration
10000
Jerk
CoordSystem
1
BufferMode
0
TransitionMode
0
TransitionParameter
Done
LA_Done
Busy
LA_Bsy
Active
LA_Act
CommandAborted
LA_Abt
Error
LA_Err
ErrorID
LA_ErrID
DMC_GroupStop0
DMC_GroupStop
5
1
AxesGroup
GS_Ex
Execute
Deceleration
Jerk
mcAborting
BufferMode
Done
GS_Done
Busy
GS_Bsy
Active
GS_Act
CommandAborted
GS_Abt
Error
GS_Err
ErrorID
GS_ErrID

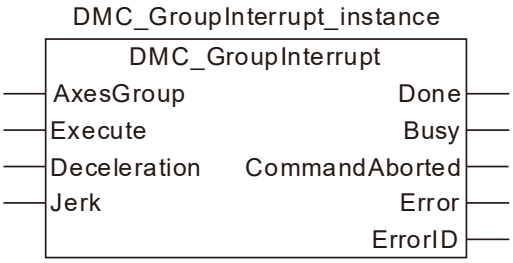
2. 运动曲线和时序图



- ❖ 按照 DMC_AddAxisToGroup、DMC_GroupEnable 的顺序执行指令，将轴组使能，当轴组使能完成后，执行指令 DMC_MoveLinearAbsolute。
- ❖ 指令 DMC_MoveLinearAbsolute 正常执行过程中，执行 DMC_GroupStop 指令，此时轴组速度会很快变为 0，而 DMC_MoveLinearAbsolute 指令被打断。当轴组停止运动后，指令 DMC_GroupStop 的完成位变为 TRUE。

11.7.6 DMC_GroupInterrupt（轴组暂停）

FB/FC	说明	适用机种
FB	此指令用于暂停当前轴组运动。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲使能的轴组编号	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
Deceleration	保留	-	-	-
Jerk	保留	-	-	-

说明：参数 Deceleration、Jerk 保留，填写数值无效。

- **输出参数**

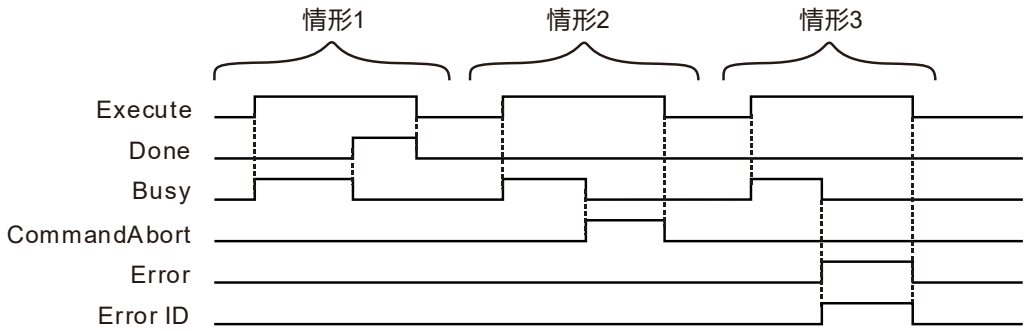
名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
CommandAborted（中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当轴组暂停完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 由 FALSE 变为 TRUE 时 ◆ 当 CommandAborted 由 FALSE 变为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
		◆ 当 Error 由 FALSE 变为 TRUE 时
CommandAborted	◆ 当指令的执行被中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 False 变为 TRUE 时，Busy 变为 TRUE，当暂停成功后，Done 变为 TRUE，同时 Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时， Done 位同时变为 FALSE。

情形2： 当该指令执行过程中被打断时，CommandAborted 变为 TRUE，同时 Busy 变为 FALSE；当 Execute 变为 FALSE 时，CommandAborted 同时变为 FALSE。

情形3： 当该指令执行过程中，有错误产生时，Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 变为 FALSE；当 Execute 变为 FALSE 时，Error 变为 FALSE，ErrorID 变为 0。

- **功能说明**

此指令用于暂停当前轴组动作。

1. 指令输入参数 Deceleration、Jerk 功能保留，用户输入数值无效，执行 DMC_GroupInterrupt 指令时，主机会使用正在控制轴组运行的轴组指令的减速度来减速，直到轴组停止运动。
2. 只有当轴组指令 DMC_MoveDirectAbsolute、DMC_MoveDirectRelative、DMC_MoveLinearAbsolute、DMC_MoveLinearRelative、DMC_MoveCircularAbsolute 与 DMC_MoveCircularRelative 正常执行时，才可以执行 DMC_GroupInterrupt 指令。
3. 执行指令 DMC_GroupInterrupt 后，用户可以使用指令 DMC_GroupContinue 恢复轴组暂停前的动作。
4. 多个 DMC_GroupInterrupt 同时执行时，它们的执行效果一致。

程序范例一

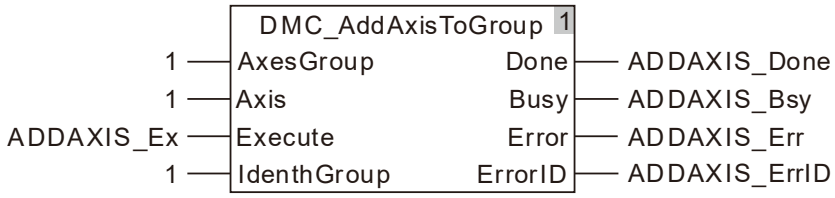
DMC_GroupInterrupt 与 DMC_GroupContinue 指令搭配使用范例如下：

1. 变量和程序

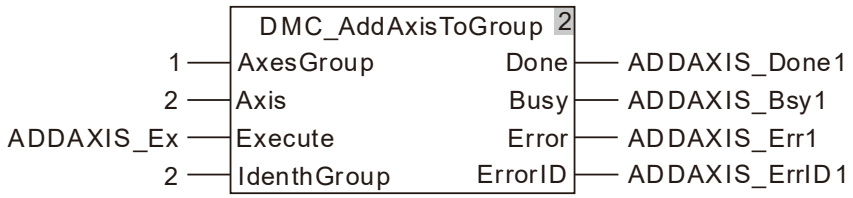
变量名	数据类型	初始值
M1	BOOL	TRUE
DMC_AddAxisToGroup0	DMC_AddAxisToGroup	
ADDAXIS_Ex	BOOL	
ADDAXIS_Done	BOOL	

变量名	数据类型	初始值
ADDAXIS_Bsy	BOOL	
ADDAXIS_Err	BOOL	
ADDAXIS_ErrID	WORD	
DMC_AddAxisToGroup1	DMC_AddAxisToGroup	
ADDAXIS_Done1	BOOL	
ADDAXIS_Bsy1	BOOL	
ADDAXIS_Err1	BOOL	
ADDAXIS_ErrID1	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Acc	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Dec	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Jerk	ARRAY [1..8] OF LREAL	[10000,10000]
GE_Status	BOOL	
GE_Buy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DMC_MoveLinearAbsolute0	DMC_MoveLinearAbsolute	
LINABS_Ex	BOOL	
LINABS_Pos	ARRAY [1..8] OF LREAL	[200000,200000]
LINABS_Done	BOOL	
LINABS_Bsy	BOOL	
LINABS_Act	BOOL	
LINABS_Abt	BOOL	
LINABS_Err	BOOL	
LINABS_ErrID	WORD	
DMC_GroupInterrupt0	DMC_GroupInterrupt	
GI_Ex	BOOL	
GI_Done	BOOL	
GI_Bsy	BOOL	
GI_Abt	BOOL	
GI_Err	BOOL	
GI_ErrID	WORD	
DMC_GroupContinue0	DMC_GroupContinue	
GC_Ex	BOOL	
GC_Done	BOOL	
GC_Bsy	BOOL	
GC_Abt	BOOL	
GC_Err	BOOL	
GC_ErrID	WORD	

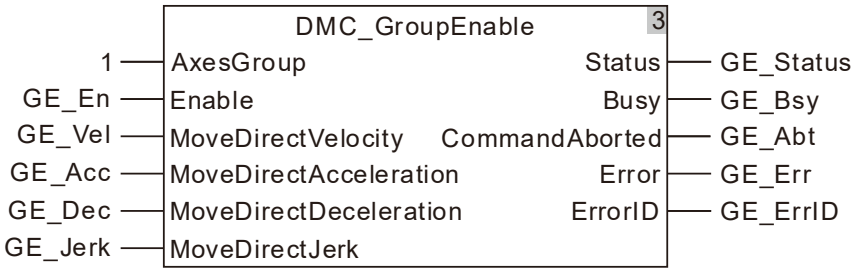
DMC_AddAxisToGroup0



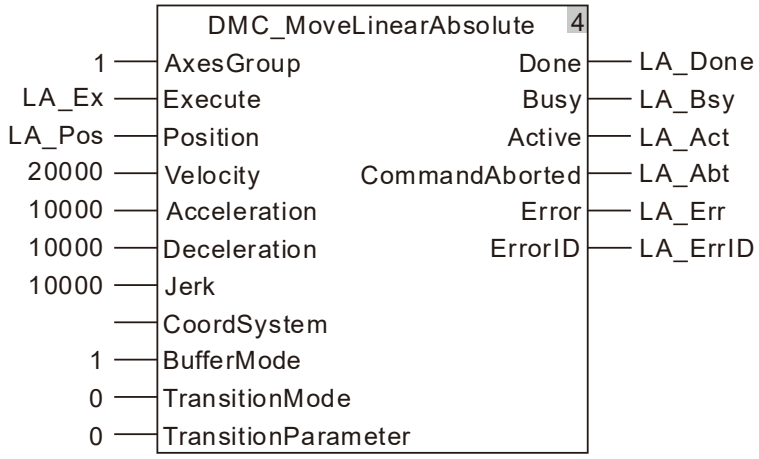
DMC_AddAxisToGroup1



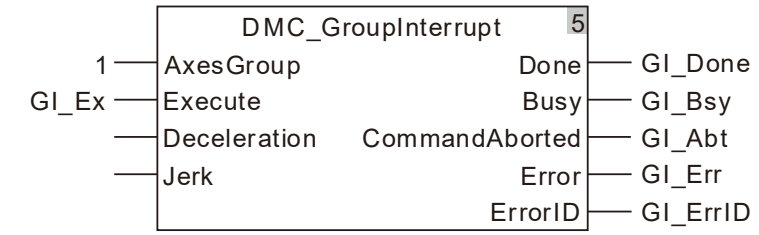
DMC_GroupEnable0

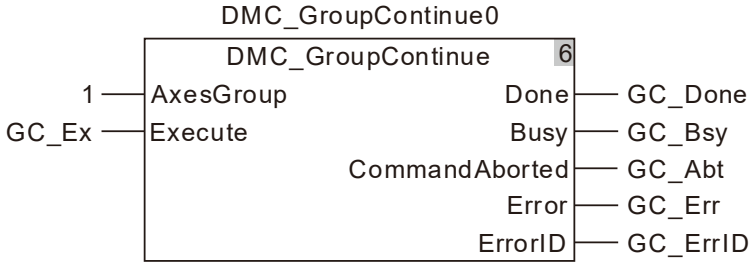


DMC_MoveLinearAbsolute0

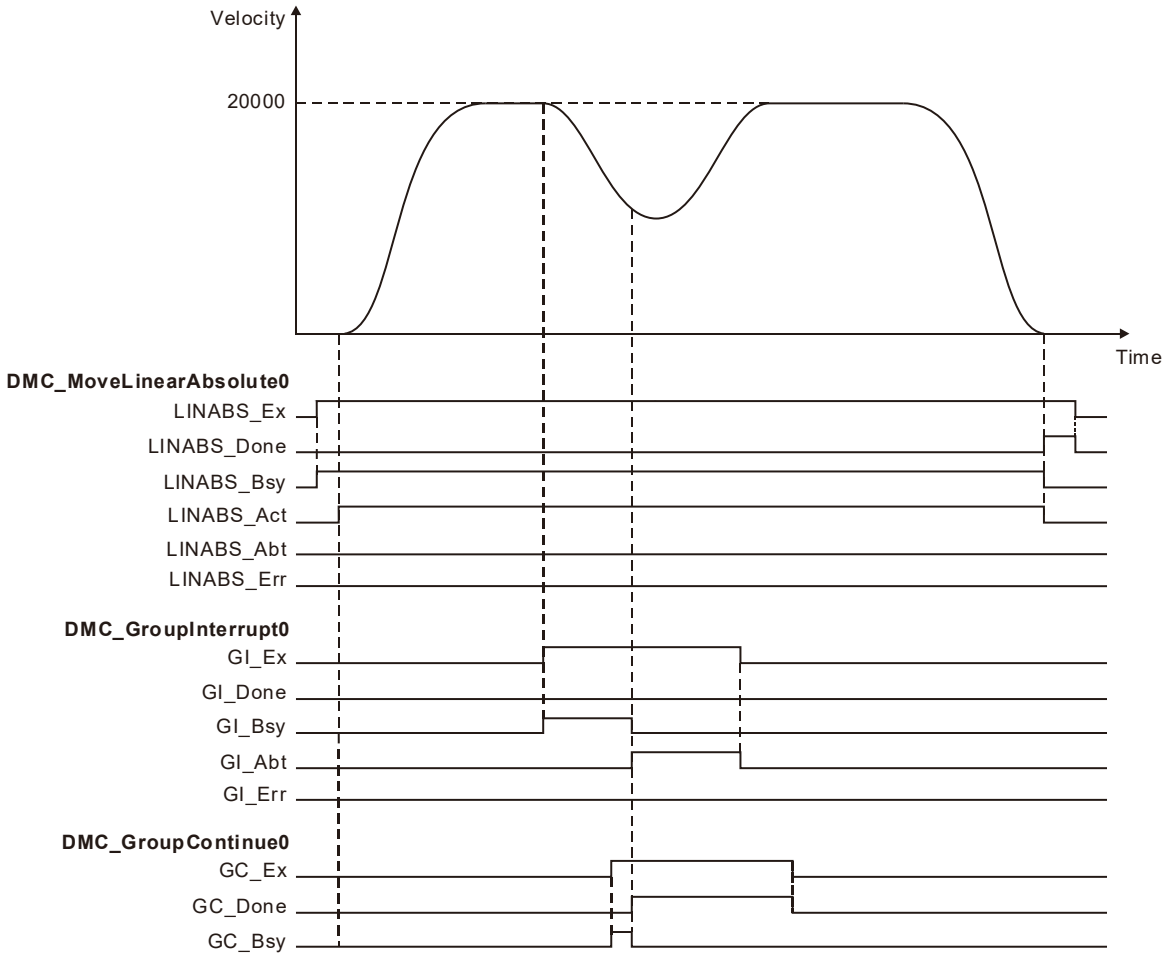


DMC_GroupInterrupt0





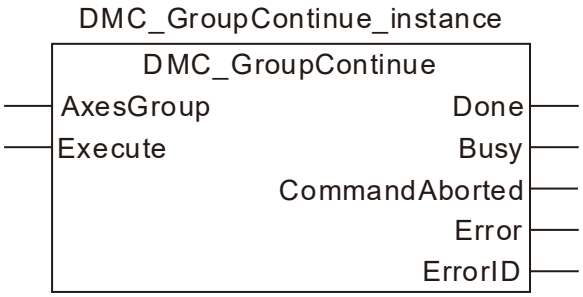
2. 程序运行过程中，轴组合成速度曲线和时序图如下：



- 按照 DMC_AddAxisToGroup、DMC_GroupEnable 的顺序执行指令，将轴组使能，当轴组使能完成后，执行指令 DMC_MoveLinearAbsolute。
- 指令 DMC_MoveLinearAbsolute 执行过程中执行 DMC_GroupInterrupt 指令，DMC_GroupInterrupt 开始执行后，轴组按照指令 DMC_MoveLinearAbsolute 的减速度进行减速运动，但是 DMC_MoveLinearAbsolute 并不会被打断。
- 轴组减速过程中，执行 DMC_GroupContinue 指令，DMC_GroupContinue 会立即打断指令 DMC_GroupInterrupt 的运行，同时指令 DMC_MovelinearAbsolute 继续运行。

11.7.7 DMC_GroupContinue（轴组解除暂停）

FB/FC	说明	适用机种
FB	此指令用于解除轴组暂停。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲使能的轴组编号	USINT	1~8（不可缺省）	*Execute* 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	

- **输出参数**

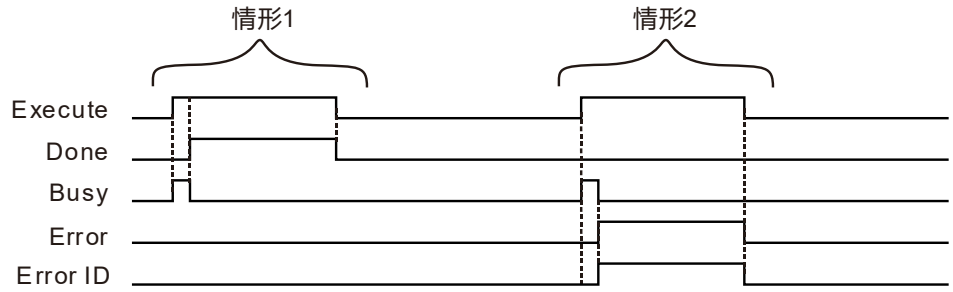
名称	功能	数据类型	输出范围
Done（完成）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
CommandAborted（中断）	该输出参数为 TRUE 时表示指令的执行被中断	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当轴组解除暂停完成时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 由 FALSE 变为 TRUE 时 ◆ 当 CommandAborted 由 FALSE 变为 TRUE 时 ◆ 当 Error 由 FALSE 变为 TRUE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
CommandAborted	◆ 当指令的执行被中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，一个周期后，Done 变为 TRUE，同时 Busy 变为 FALSE，解除暂停成功；当 Execute 由 TRUE 变为 FALSE 时， Done 位同时变为 FALSE。

情形2： 当指令执行时发生错误时，当 Execute 由 FALSE 变为 TRUE 时，Busy 同时变为 TRUE；一个周期后，Error 变为 TRUE，同时 ErrorID 显示相应的错误码，同时 Busy 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 同时变为 FALSE，同时 ErrorID 变为 0.

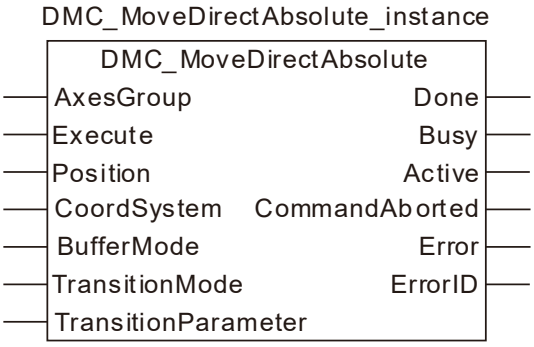
- **功能说明**

此指令用于解除轴组暂停状态。

1. 只有当轴组指令 DMC_MoveDirectAbsolute、DMC_MoveDirectRelative、DMC_MoveLinearAbsolute、DMC_MoveLinearRelative 、 DMC_MoveCircularAbsolute 与 DMC_MoveCircularRelative 被 DMC_GroupInterrupt 暂停时，才可以执行 DMC_GroupContinue 指令。
2. 执行指令 DMC_GroupContinue 后，轴组暂停状态立即解除并继续执行轴组暂停前的动作。

11.7.8 DMC_MoveDirectAbsolute（绝对快速定位）

FB/FC	说明	适用机种
FB	此指令用于控制轴组内各轴以指定速度从当前位置运动到终点位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲控制的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
Position（终点位置）	设定各轴的终点位置	ARRAY [1..8] OF LREAL	负数、正数、0（0）	Execute 由 FALSE 变为 TRUE
CoordSystem	系统保留			
BufferMode（交接模式）	设定两个轴组指令之间交接模式 1：等待	MC_Buffer_Mode	1: mcBuffered	Execute 由 FALSE 变为 TRUE
TransitionMode（过渡模式）	设定两个轴组指令之间的过渡模式 0：无过渡曲线插入	MC_Transition_Mode	0: mcTMNone	Execute 由 FALSE 变为 TRUE
TransitionParameter（过渡参数）	设定特定过渡模式所需的过渡参数	LREAL	正数、0（0）	Execute 由 FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 输入参数 Position[1] ~ Position[8]的值表示快速定位的终点位置。
3. 该指令输入参数 BufferMode 仅支持 mcBuffered 模式，如果选择其他模式，那么执行该指令时指令报错。关于 BufferMode 的详细内容，请参考第 10.3 节。

4. 该指令输入参数 TransitionMode 仅支持 mcTMNone 模式，如果选择其他模式，那么执行该指令时指令报错。
5. 该指令输入参数 TransitionParameter 的值无效。

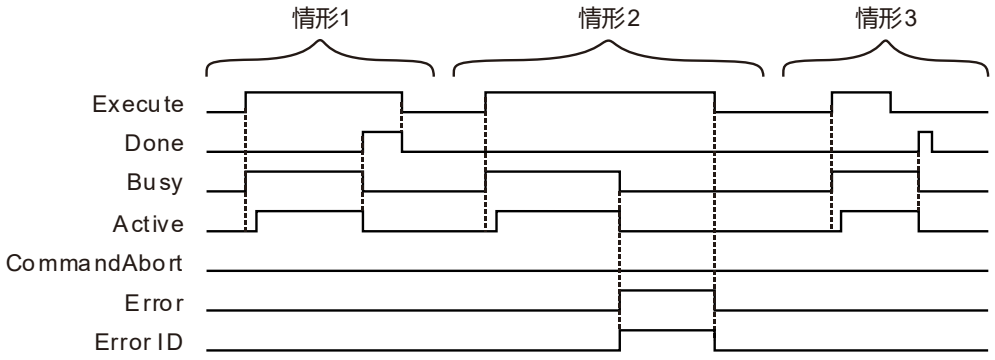
- **输出参数**

名称	功能	数据类型	输出范围
Done （控制中）	该输出参数为 TRUE 时表示正在指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被打断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出 错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当到达终点位置时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TURE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
CommandAbort	◆ 当指令的执行被其他运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且两个周期后，Active 变为 TRUE；当轴组到达终点位置时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时 ,当有错误产生时(如轴组状态机异常时) ,Error 变为 TRUE，ErrorID 显示对应的错误码 ,同时 Busy 和 Active 变为 FALSE ;当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

情形3： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，且一个周期后，Done 变为 FALSE。

- **功能说明**

此指令用于轴组进行快速定位，可以控制轴组中的一个轴或多个轴。V1.01 及以上固件版本支持该功能。

运动过程中各轴是相对独立的，各轴运动时的速度、加速度、减速度和加速度的变化率由指令 DMC_GroupEnable 的输入参数 MoveDirectVelocity、MoveDirectAcceleration、MoveDirectDeceleration、MoveDirectJerk 的值决定。

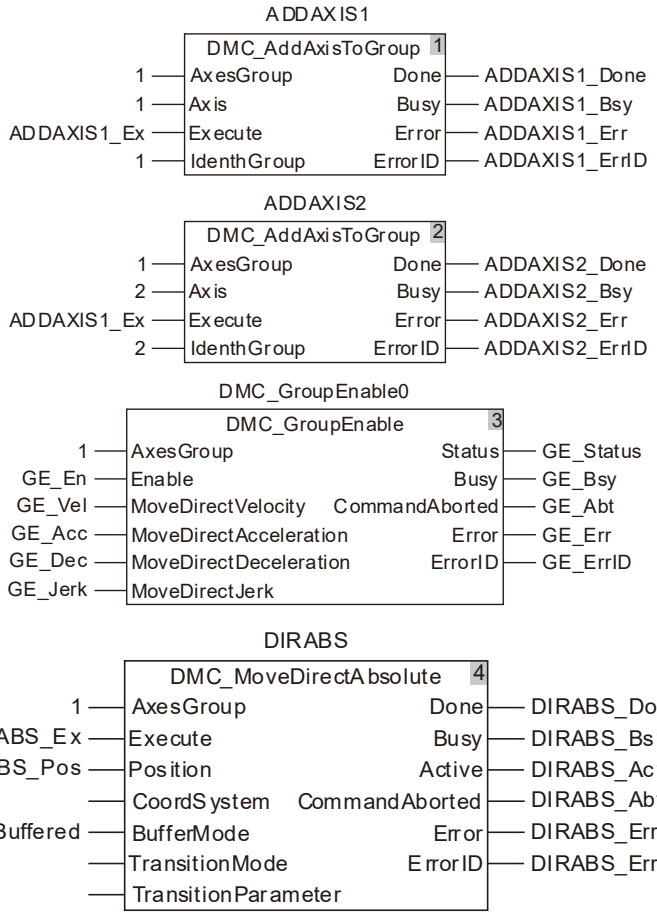
程序范例一

DMC_MoveDirectAbsolute 指令单独执行的范例如下：

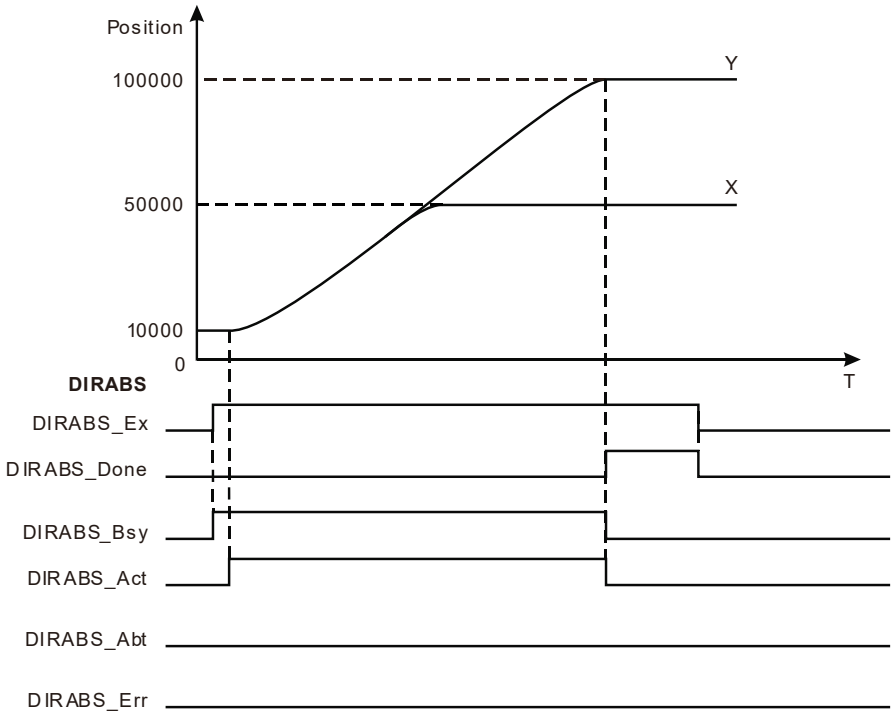
1. 变量和程序

变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	

变量名	数据类型	初始值
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DIRABS	DMC_MoveDirectAbsolute	
DIRABS_Ex	BOOL	
DIRABS_Pos	ARRAY [1..8] OF LREAL	
DIRABS_Done	BOOL	
DIRABS_Bsy	BOOL	
DIRABS_Act	BOOL	
DIRABS_Abt	BOOL	
DIRABS_Err	BOOL	
DIRABS_ErrID	WORD	



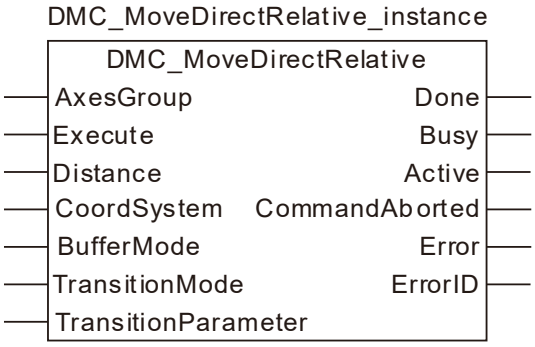
2. 运动过程中，X 轴和 Y 轴的运动曲线和时序图如下：



- X 轴和 Y 轴初始位置均为 10000，设置 DMC_MoveDirectAbsolute 指令的输入参数 DIRABS_Pos[1]、DIRABS_Pos[2]分别为 50000、100000。
- 当 DIRABS_Ex 变为 TRUE 时，DIRABS_Bsy 变为 TRUE，两个周期后，DIRABS_Act 变为 TRUE，轴组开始运行。
- X 轴和 Y 轴的按照 DMC_GroupEnable 指令设定的速度、加速度和减速度运行。当 X 轴运行到 50000 位置时，X 轴停止运行，此时 Y 轴继续运行，当 Y 轴到达 100000 位置时，指令完成。

11.7.9 DMC_MoveDirectRelative（相对快速定位）

FB/FC	说明	适用机种
FB	此指令用于控制轴组内各轴以指定速度从当前位置运动运动相应距离。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲控制的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
Distance（运动距离）	设定各轴的运动距离	ARRAY [1..8] OF LREAL	负数、正数、0（0）	Execute 由 FALSE 变为 TRUE
CoordSystem	系统保留			
BufferMode（交接模式）	设定两个轴组指令之间交接模式 1：等待	MC_Buffer_Mode	1: mcBuffered	Execute 由 FALSE 变为 TRUE
TransitionMode（过渡模式）	设定两个轴组指令之间的过渡模式 0：无过渡曲线插入	MC_Transition_Mode	0: mcTMNone	Execute 由 FALSE 变为 TRUE
TransitionParameter（过渡参数）	设定特定过渡模式所需的过渡参数	LREAL	正数、0（0）	Execute 由 FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 输入参数 Distance[1] ~ Distance [8]的值表示各轴从起点到终点的运动距离。
3. 该指令输入参数 BufferMode 仅支持 mcBuffered 模式，如果选择其他模式，那么执行该指令时指令报错。关于 BufferMode 的详细内容，请参考第 10.3 节。

4. 该指令输入参数 TransitionMode 仅支持 mcTMNone 模式，如果选择其他模式，那么执行该指令时指令报错。
5. 该指令输入参数 TransitionParameter 的值无效。

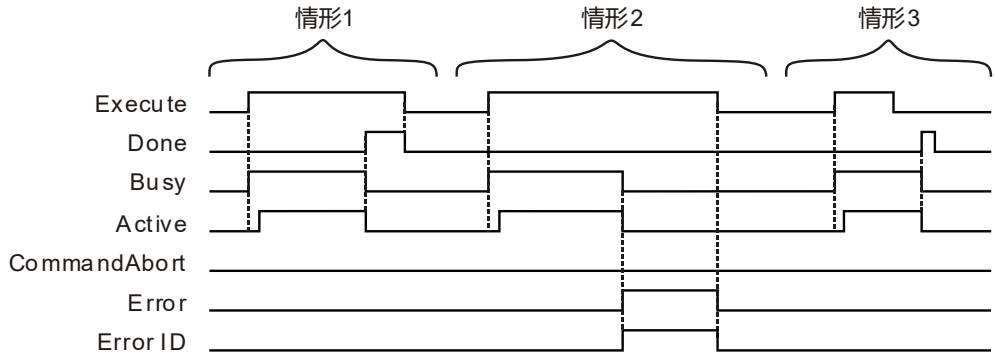
- **输出参数**

名称	功能	数据类型	输出范围
Done （控制中）	该输出参数为 TRUE 时表示指令执行完毕。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被打断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出 错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当到达终点位置时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TURE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
CommandAbort	◆ 当指令的执行被其他运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且两个周期后，Active 变为 TRUE；当轴组到达终点位置时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，当有错误产生时(如轴组状态机异常时) Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

情形3： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，且一个周期后，Done 变为 FALSE。

- **功能说明**

此指令用于轴组进行快速定位，可以控制轴组中的一个轴或多个轴。V1.01 及以上固件版本支持该功能。

运动过程中各轴是相对独立的，各轴运动时的速度、加速度、减速度和加速度的变化率由指令 DMC_GroupEnable 的输入参数 MoveDirectVelocity、MoveDirectAcceleration、MoveDirectDeceleration、MoveDirectJerk 的值决定。

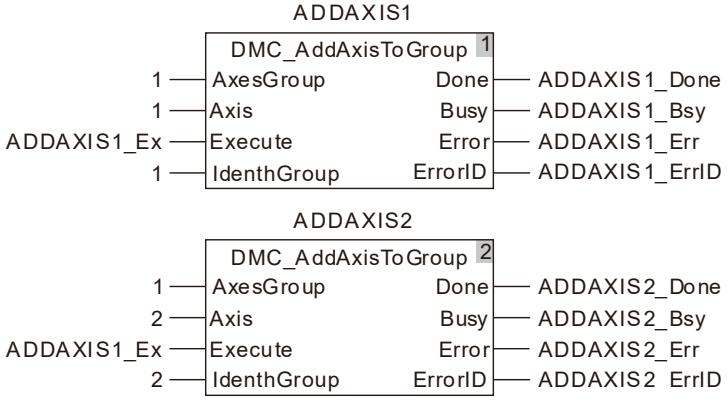
程序范例一

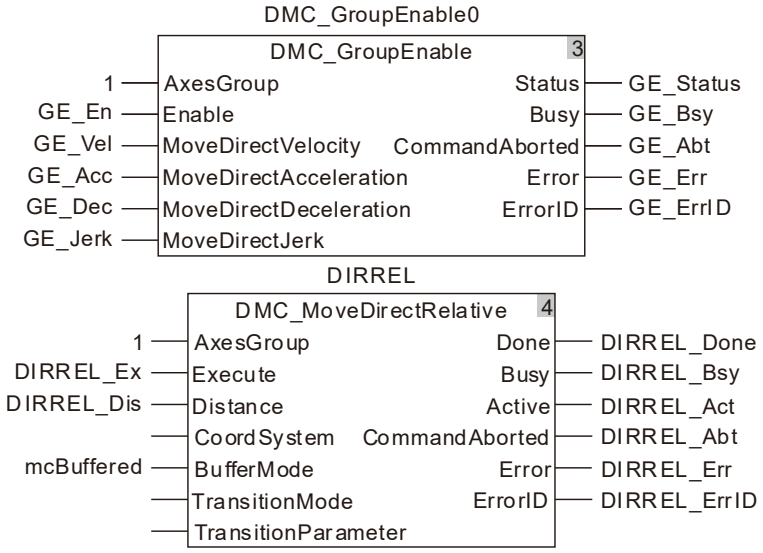
DMC_MoveDirectRelative 指令单独执行的范例如下：

1. 变量和程序

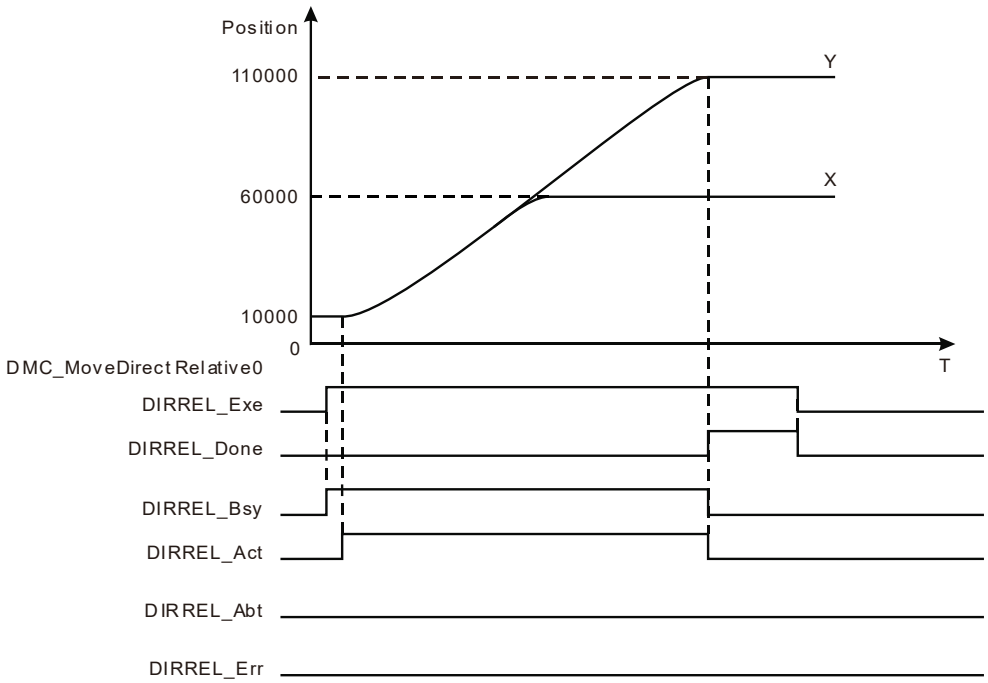
变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	

变量名	数据类型	初始值
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DIRREL	DMC_MoveDirectRelative	
DIRREL_Ex	BOOL	
DIRREL_Dis	ARRAY [1..8] OF LREAL	
DIRREL_Done	BOOL	
DIRREL_Bsy	BOOL	
DIRREL_Act	BOOL	
DIRREL_Abt	BOOL	
DIRREL_Err	BOOL	
DIRREL_ErrID	WORD	





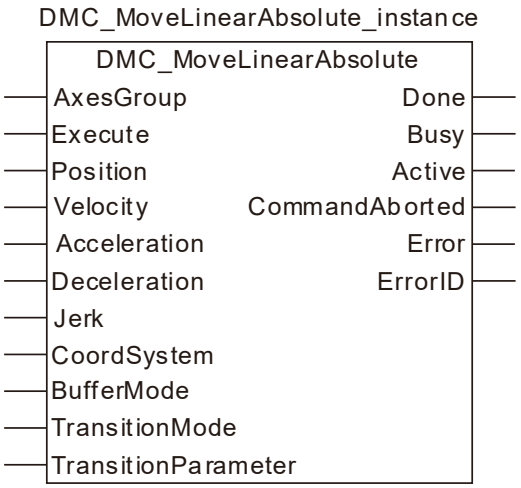
2. 运动过程中，X 轴和 Y 轴的运动曲线和时序图如下：



- X 轴和 Y 轴初始位置均为 10000，设置 DMC_MoveDirectRelative 指令的输入参数 DIRREL_Dis[1]、DIRREL_Dis[2]分别为 50000、100000。
- 当 DIRREL_Ex 变为 TRUE 时，DIRREL_Bsy 变为 TRUE，两个周期后，DIRREL_Act 变为 TRUE，轴组开始运行。
- X 轴和 Y 轴的按照 DMC_GroupEnable 指令设定的速度、加速度和减速度运行。当 X 轴运行到 60000 位置时，停止运行，此时 Y 轴继续运行，当 Y 轴到达 110000 位置时，指令完成。

11.7.10 DMC_MoveLinearAbsolute（绝对直线插补）

FB/FC	说明	适用机种
FB	此指令用于控制轴执行直线插补功能，其中轴的终点位置为绝对位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲控制的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
Position（终点位置）	设定直线插补终点位置	ARRAY [1..8] OF LREAL	负数、正数、0（0）	Execute 由 FALSE 变为 TRUE
Velocity（合成速度）	设定的目标合成速度（单位：单元/秒）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
Acceleration（合成加速度）	设定的目标合成加速度（单位：单元/秒²）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
Deceleration（合成减速度）	设定的目标合成减速度（单位：单元/秒²）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
Jerk（合成加速度变化率）	设定的目标合成加速度变化率（单位：单元/秒³）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
CoordSystem	系统保留			
BufferMode（交接模式）	设定两个轴组指令之间交接模式 1：等待	MC_Buffer_Mode	1: mcBuffered 3: mcBlending-Previous	Execute 由 FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	3：以前一指令的速度交接			
TransitionMode （过渡模式）	设定两个轴组指令之间的过渡模式 0：无过渡曲线插入 2：以给定的恒定速度过渡 3：以给定的角距过渡	MC_Transition_Mode	0: mcTMNone 2: mcTMConstantVelocity 3: mcTMCornerDistance	Execute 由 FALSE 变为 TRUE
TransitionParameter （过渡参数）	设定特定过渡模式所需的过渡参数	LREAL	正数、0 （0）	Execute 由 FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 输入参数 Position[1] ~ Position[8]的值表示直线插补的终点位置。
3. 轴组指令 Velocity、Acceleration、Deceleration、Jerk 的关系说明请参考第 10.2 节。
4. 关于 BufferMode 的详细内容，请参考第 10.3 节。
5. 输入参数 TransitionParameter 仅在 TransitionMode 为 mcTMCornerDistance 时有效。

- **输出参数**

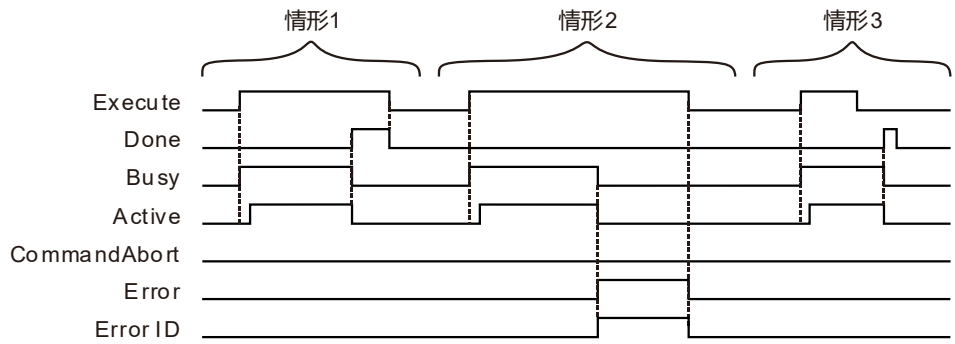
名称	功能	数据类型	输出范围
Done （控制中）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被打断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出 错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当到达终点位置时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TURE，且一个周期后，Done 变为 FALSE

名称	变为 TRUE 的时机	变为 FALSE 的时机
Busy	◆ 当Execute为TRUE时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
CommandAbort	◆ 当指令的执行被其他运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且两个周期后，Active 变为 TRUE；当轴组到达终点位置时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时 ,当有错误产生时(如轴组状态机异常时) ,Error 变为 TRUE，ErrorID 显示对应的错误码 ,同时 Busy 和 Active 变为 FALSE ;当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

情形3： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，且一个周期后，Done 变为 FALSE。

- **功能说明**

此指令用于轴组进行直线插补，可以控制轴组中的一个轴或多个轴。V1.01 及以上固件版本支持该功能。

1. DMC_MoveLinearAbsolute 指令的参数 Velocity 为终端机构的目标速度，终端机构速度与各轴速度的关系如下：终端机构速度的平方 = 各轴速度的平方和。该指令参数 Acceleration、Deceleration 为终端机构的目标加速度和目标减速度，终端机构的加速度和减速度与各轴加、减速度的关系为：终端机构的加（减）速度 = 各轴加（减）速度的平方和。
2. DMC_MoveLinearAbsolute 指令参数 Jerk 保留。

3. 参数 BufferMode、TransitionMode 与 TransitionParameter 之间的关系说明如下表所示。BufferMode 选择 mcBuffered 模式时，TransitionMode 仅支持 mcTMNone 模式；BufferMode 选择 mcBlendingPrevious 模式时，TransitionMode 支持 mcTMConstantVelocity 与 mcTMCornerDistance 两种模式。

BufferMode 值	TransitionMode 值	说明
mcBuffered(1)	mcTMNone(0)	等待前一个插补指令的动作正常执行结束后，立即执行当前指令
mcBlending-Previous(3)	mcTMConstant-Velocity(2)	圆滑过渡：等待前一个插补指令执行完成，并立即执行当前指令，交接速度为前一个指令的合成速度，交接完成后终端机构先进行圆滑过渡，然后在做直线运动，如范例二所示。
	mcTMCorner-Distance(3)	角距过渡：等待前一个插补指令执行完成，并立即执行当前指令。前一个插补指令执行过程中，当终端机构与前一个插补指令终点距离等于参数 TransitionParameter 设置的值时，前一个插补指令立即完成，当前指令开始执行。当前指令执行时，终端机构先走过一段圆弧再作直线插补，圆弧终点到前一个插补指令终点的距离仍等于参数 TransitionParameter 的值，如范例三所示。

程序范例一

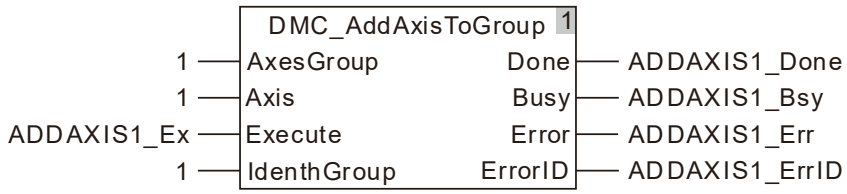
DMC_MoveLinearAbsolute 指令单独执行的范例如下：

1. 变量和程序

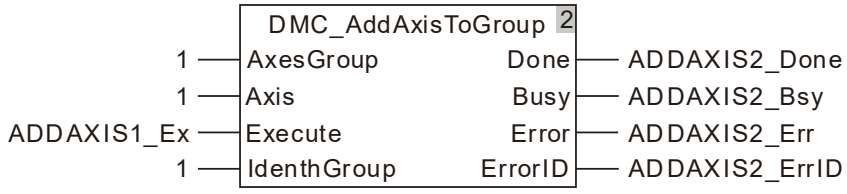
变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS 2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
ADDAXIS3	DMC_AddAxisToGroup	
ADDAXIS3_Done	BOOL	
ADDAXIS3_Bsy	BOOL	
ADDAXIS3_Err	BOOL	

变量名	数据类型	初始值
ADDAXIS3_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINABS	DMC_MoveLinearAbsolute	
LINABS_Ex	BOOL	
LINABS_Pos	ARRAY [1..8] OF LREAL	
LINABS_Done	BOOL	
LINABS_Bsy	BOOL	
LINABS_Act	BOOL	
LINABS_Abt	BOOL	
LINABS_Err	BOOL	
LINABS_ErrID	WORD	

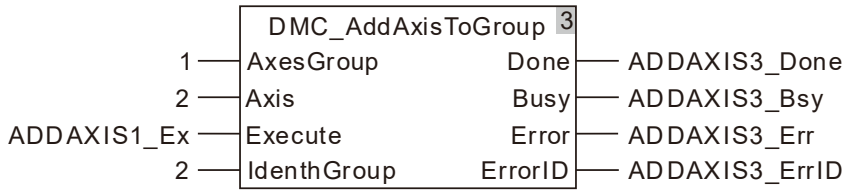
ADDAXIS1



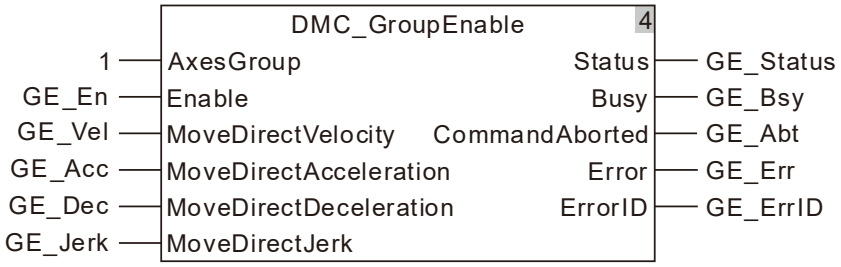
ADDAXIS2



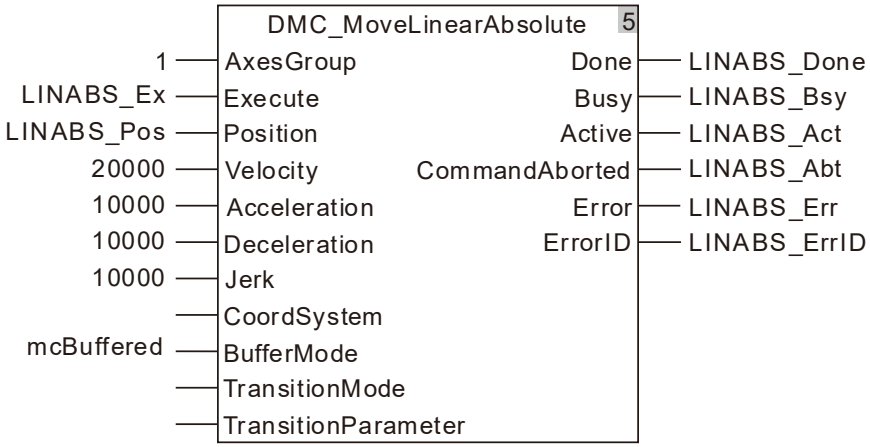
ADDAXIS3



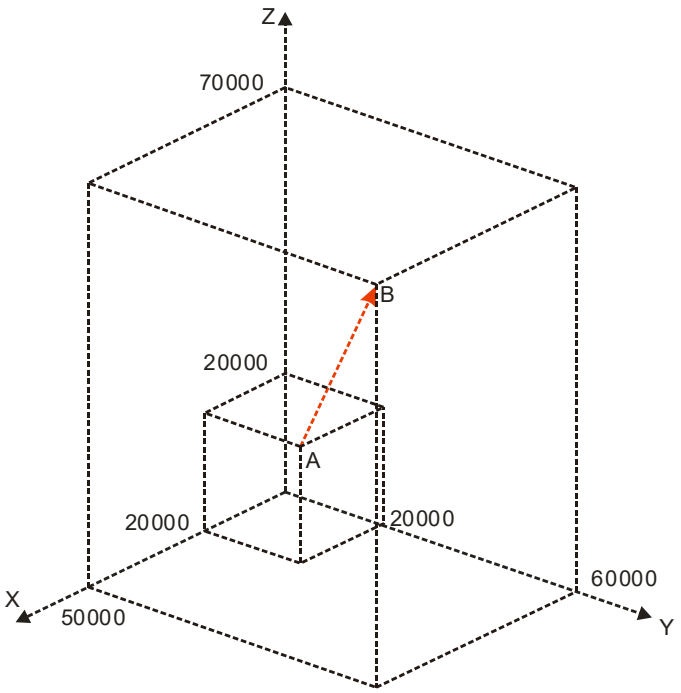
DMC_GroupEnable0



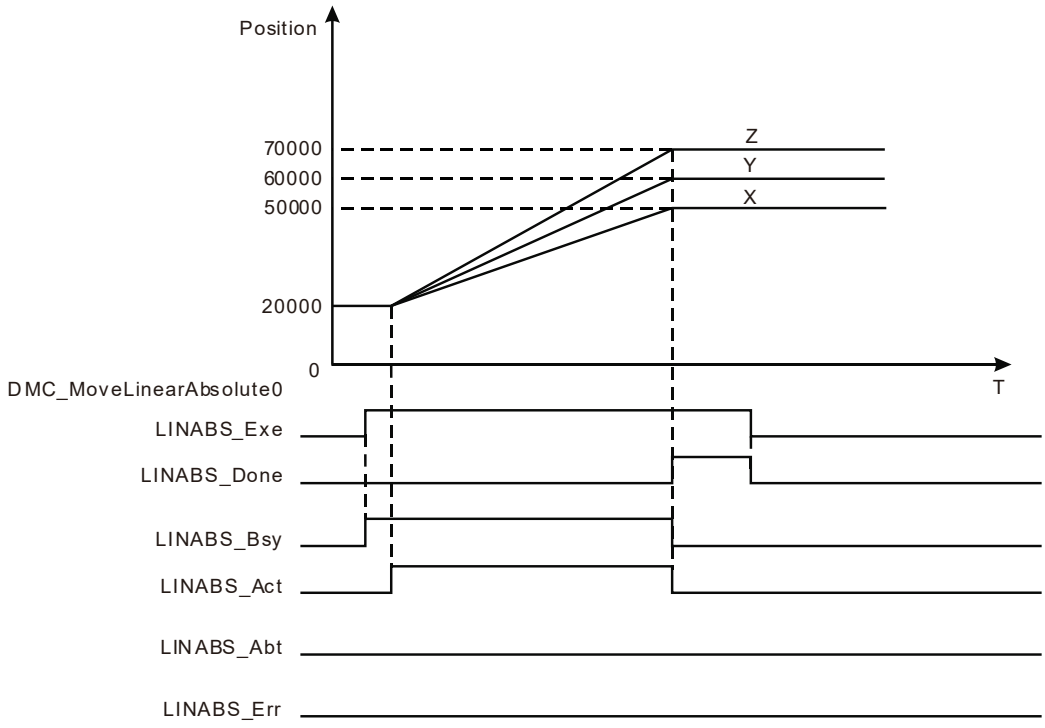
LINABS



2. 指令执行后，整个运动过程如下图所示：



3. 运动过程中，X 轴、Y 轴和 Z 轴的运动曲线和时序图如下：



- X 轴、Y 轴和 Z 轴的初始位置均为 20000，设置 DMC_MoveLinearAbsolute 指令的输入参数 LINABS_Pos[1]、LINABS_Pos[2]和 LINABS_Pos[3]分别为 50000、60000 和 70000。
- 当 LINABS_Ex 变为 TRUE 时，LINABS_Bsy 变为 TRUE，两个周期后，LINABS_Act 变为 TRUE，轴组开始运行。

❖ DMC_MoveLinearAbsolut 指令完成时，X 轴、Y 轴和 Z 轴同时到达各自目标位置，LINABS_Done 变为 TRUE，LINABS_Bsy 和 LINABS_Act 变为 FALSE。

程序范例二

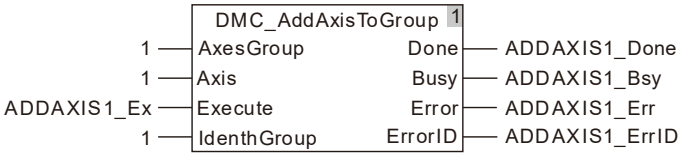
两个 DMC_MoveLinearAbsolute 指令以 mcTMConstantVelocity 方式交接的范例如下：

1. 变量和程序

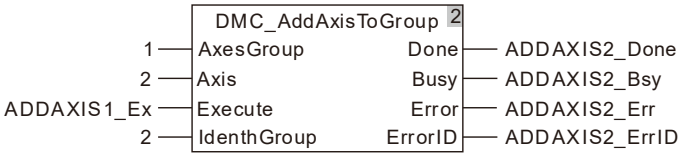
变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINABS1	DMC_MoveLinearAbsolute	
LINABS1_Ex	BOOL	
LINABS1_Pos	ARRAY [1..8] OF LREAL	
LINABS1_Done	BOOL	
LINABS1_Bsy	BOOL	
LINABS1_Act	BOOL	
LINABS1_Abt	BOOL	
LINABS1_Err	BOOL	
LINABS1_ErrID	WORD	
LINABS2	DMC_MoveLinearAbsolute	

变量名	数据类型	初始值
LINABS2_Pos	ARRAY [1..8] OF LREAL	
LINABS2_Done	BOOL	
LINABS2_Bsy	BOOL	
LINABS2_Act	BOOL	
LINABS2_Abt	BOOL	
LINABS2_Err	BOOL	
LINABS2_ErrID	WORD	

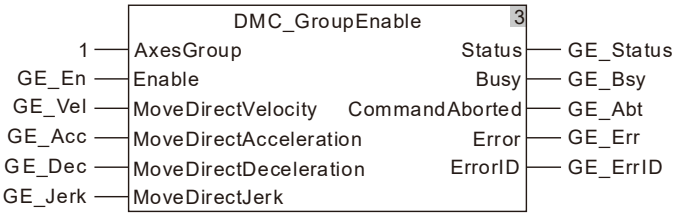
ADDAXIS1



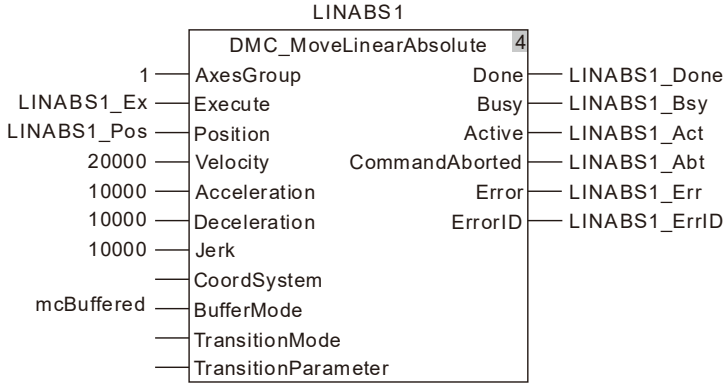
ADDAXIS2



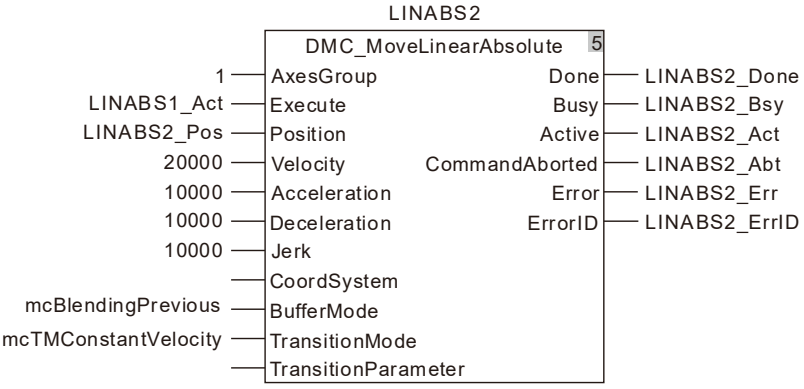
DMC_GroupEnable0



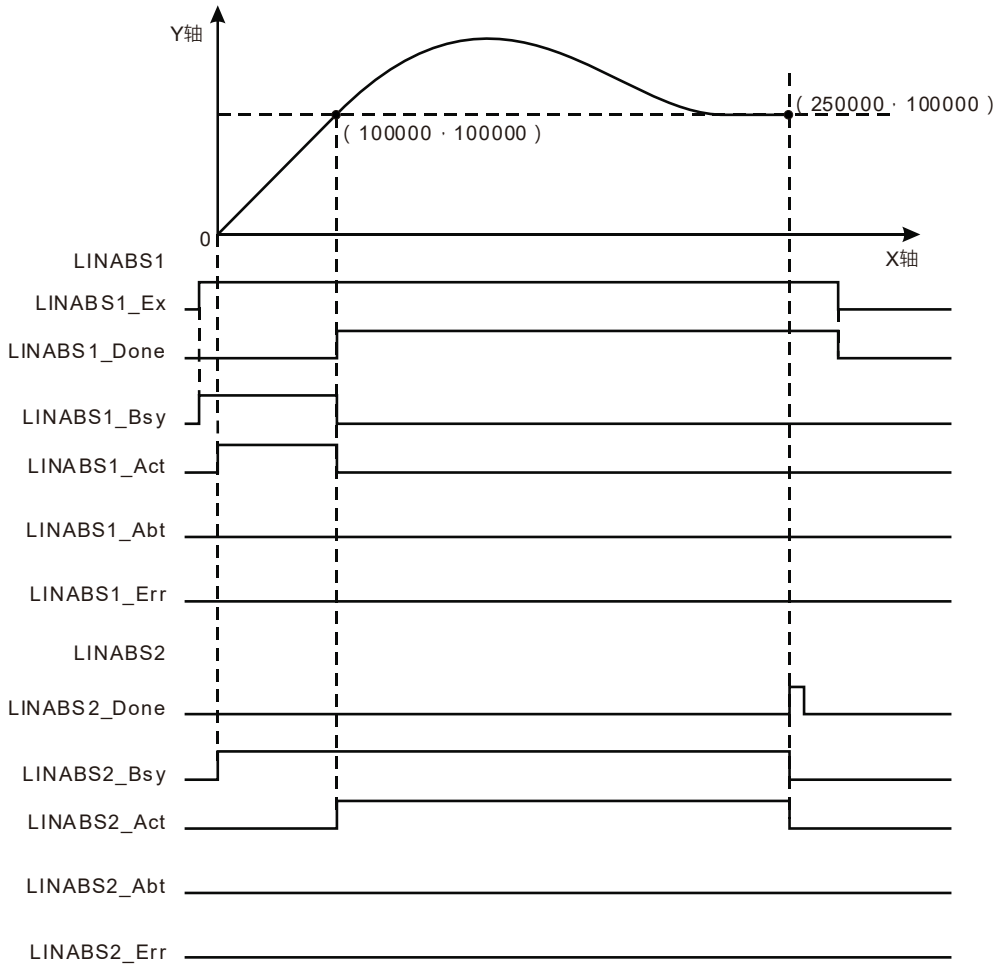
LINABS1



LINABS2



2. 终端机构的运行曲线和时序图如下：



- 设置 LINABS2 的 BufferMode 为 mcBlendingPrevious，TransitionMode 为 mcTMConstantVelocity 模式。
- 设置 LINABS1 的输入参数 LINABS1_Pos [1]、LINABS1_Pos [2]的值为 100000，设置 LINABS2 的输入参数 LINABS2_ Pos [1]、LINABS2_ Pos [2]的值分别为 250000、100000。
- 先执行 ADDAXIS1 与 ADDAXIS2，然后执行 DMC_GroupEnable0，当轴组使能成功后，执行指令 LINABS1，然后立即执行 LINABS2。
- 当终端机构运行到坐标（100000，100000）时，LINABS1_Done 变为 TRUE，同时 LINABS2_Act 变为 TRUE，LINABS2 开始执行，此时终端机构速度为前一个指令的目标速度 20000。
- 执行 LINABS2 后，终端机构先进行圆滑过渡，然后在做直线运动。当终端机构到达坐标（250000，100000）时，指令完成。

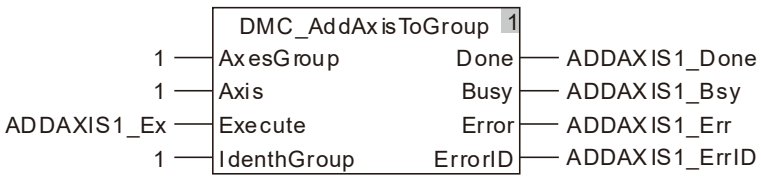
程序范例三

两个 DMC_MoveLinearAbsolute 指令以 mcTMCornerDistance 方式交接的范例如下：

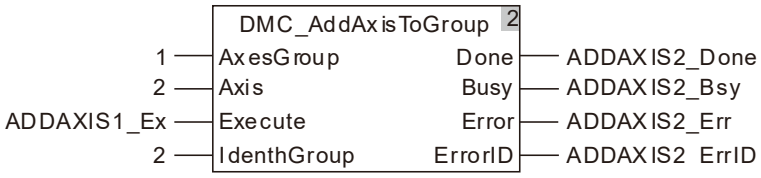
1. 变量和程序

变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINABS1	DMC_MoveLinearAbsolute	
LINABS1_Ex	BOOL	
LINABS1_Pos	ARRAY [1..8] OF LREAL	
LINABS1_Done	BOOL	
LINABS1_Bsy	BOOL	
LINABS1_Act	BOOL	
LINABS1_Abt	BOOL	
LINABS1_Err	BOOL	
LINABS1_ErrID	WORD	
LINABS2	DMC_MoveLinearAbsolute	
LINABS2_Pos	ARRAY [1..8] OF LREAL	
LINABS2_Done	BOOL	
LINABS2_Bsy	BOOL	
LINABS2_Act	BOOL	
LINABS2_Abt	BOOL	
LINABS2_Err	BOOL	
LINABS2_ErrID	WORD	

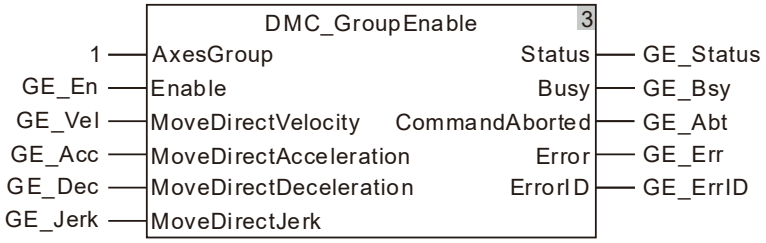
ADDAXIS1



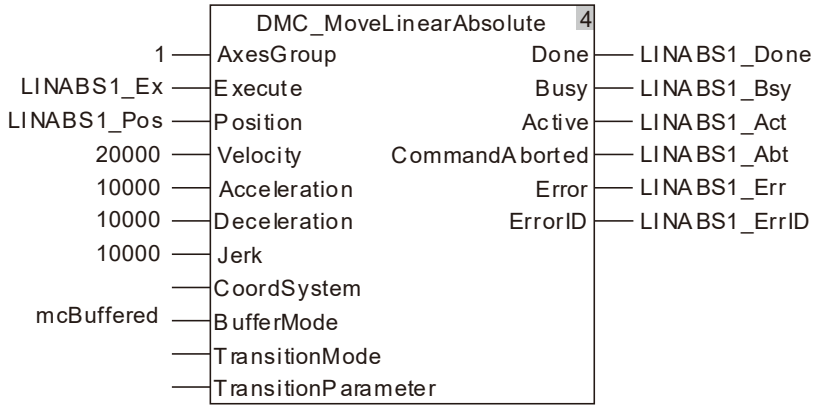
ADDAXIS2



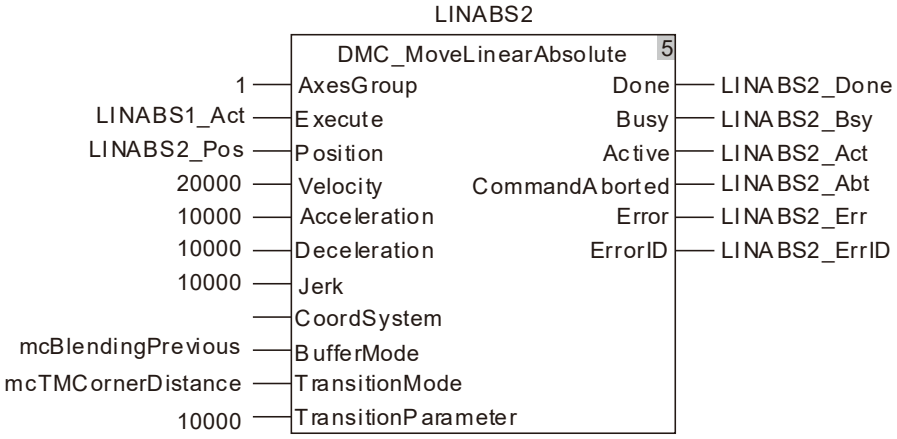
DMC_GroupEnable0



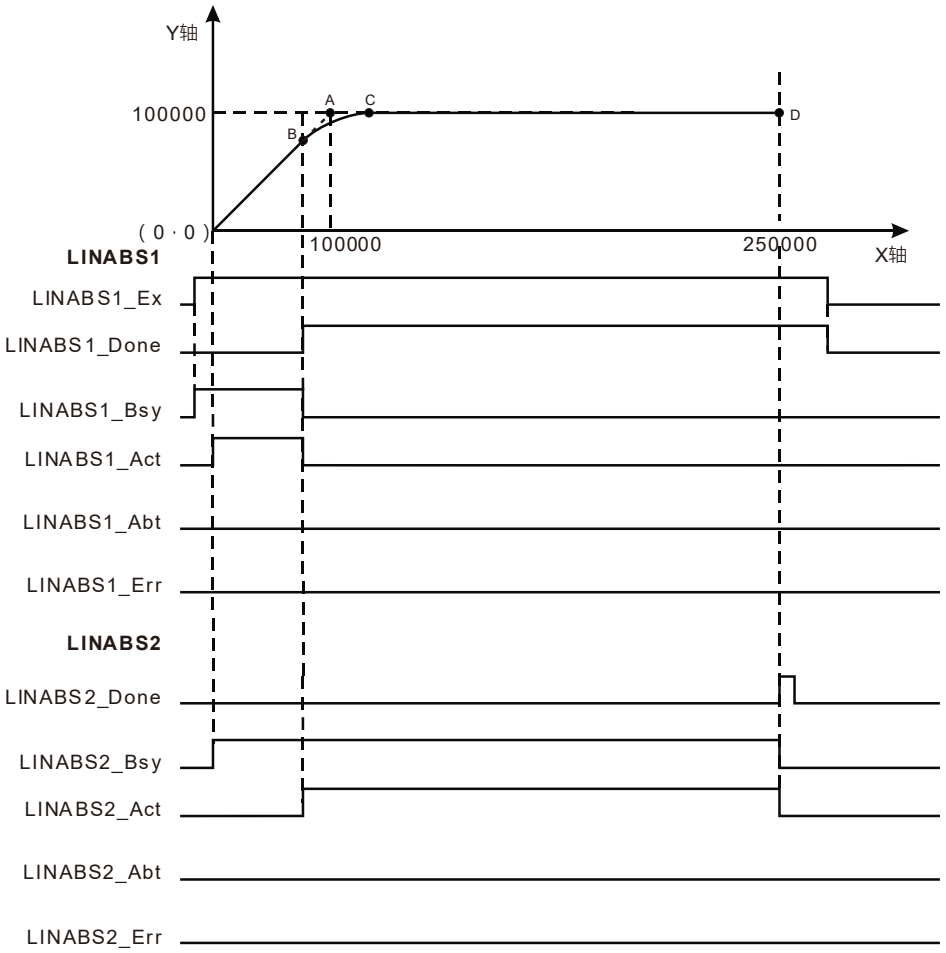
LINABS1



LINABS2



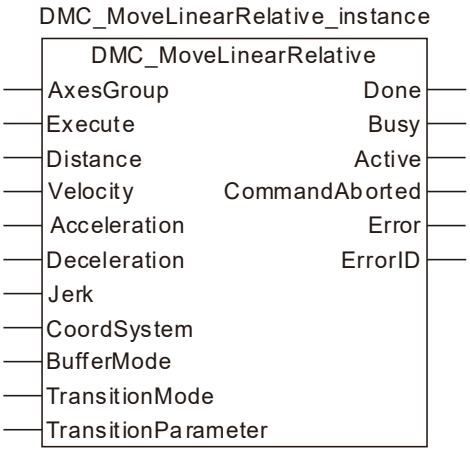
2. 终端机构的运行曲线和时序图如下：



- 设置 LINABS2 的 BufferMode 为 mcBlendingPrevious，TransitionMode 为 mcTMCornerDistance 模式，TransitionParameter 的值为 10000。
- 设置 LINABS1 的输入参数 LINABS1_Pos[1]、LINABS1_Pos[2]的值为 100000，设置 LINABS2 的输入参数 LINABS2_ Pos [1]、LINABS2_ Pos [2]的值分别为 250000、100000。
- 先执行 ADDAXIS1 与 ADDAXIS2，然后执行 DMC_GroupEnable0，当轴组使能成功后，执行指令 LINABS1，然后立即执行 LINABS2。
- 当终端机构运行到坐标点 B 时，LINABS1_Done 变为 TRUE，同时 LINABS2_Act 变为 TRUE，LINABS2 开始执行。
- 执行 LINABS2 后，终端机构先走过一段圆弧，当运行到坐标点 C 后，终点机构继续作直线插补。
- 点 B 到点 A 的距离等于点 C 到点 A 的距离，且该距离等于 LINABS2 的参数 TransitionParameter 的值 10000。

11.7.11 DMC_MoveLinearRelative（相对直线插补）

FB/FC	说明	适用机种
FB	此指令用于控制轴执行直线插补功能，其中轴的终点位置为相对位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲控制的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
Distance（各轴移动距离）	设定各轴移动距离	ARRAY [1..8] OF LREAL	负数、正数、0（0）	Execute 由 FALSE 变为 TRUE
Velocity（合成速度）	设定的目标合成速度（单位：单元/秒）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
Acceleration（合成加速度）	设定的目标合成加速度（单位：单元/秒²）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
Deceleration（合成减速度）	设定的目标合成减速度（单位：单元/秒²）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
Jerk（合成加速度变化率）	设定的目标合成加速度变化率（单位：单元/秒³）	LREAL	正数（0）	Execute 由 FALSE 变为 TRUE
CoordSystem	系统保留			
BufferMode（交接模式）	设定两个轴组指令之间交接模式	MC_Buffer_Mode	1: mcBuffered 3: mcBlending-Previous	Execute 由 FALSE 变为 TRUE

名称	功能	数据类型	设定范围 （缺省值）	生效时机
	1：等待 3 : 以前一指令的速度交接			
TransitionMode （过渡模式）	设定两个轴组指令之间的过渡模式 0：无过渡曲线插入 2 :以给定的恒定速度过渡 3：以给定的角距过渡	MC_Transition_Mode	0: mcTMNone 2：mcTMConstantVelocity 3: mcTMCornerDistance	Execute 由 FALSE 变为 TRUE
TransitionParameter （过渡参数）	设定特定过渡模式所需的过渡参数	LREAL	正数、0 （0）	Execute 由 FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 输入参数 Distance[1] ~ Distance[8]的值表示直线插补的终点位置。
3. 轴组指令 Velocity、Acceleration、Deceleration、Jerk 的关系说明请参考第 10.2 节。
4. 关于 BufferMode 的详细内容，请参考第 10.3 节。
5. 输入参数 TransitionParameter 仅在 TransitionMode 为 mcTMCornerDistance 时有效。

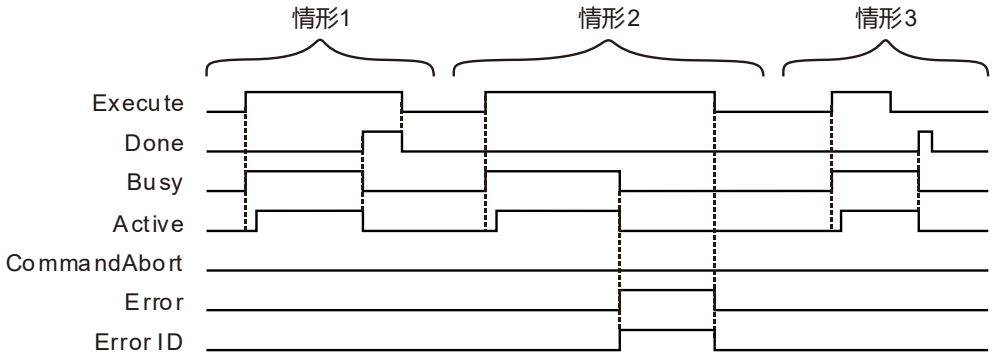
- **输出参数**

名称	功能	数据类型	输出范围
Done （控制中）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被打断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出 错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当到达终点位置时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TURE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
CommandAbort	◆ 当指令的执行被其他运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且两个周期后，Active 变为 TRUE；当轴组到达终点位置时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时 ,当有错误产生时(如轴组状态机异常时) ,Error 变为 TRUE，ErrorID 显示对应的错误码 ,同时 Busy 和 Active 变为 FALSE ;当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

情形3： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，且一个周期后，Done 变为 FALSE。

- **功能说明**

此指令用于轴组进行直线插补，可以控制轴组中的一个轴或多个轴。V1.01 及以上固件版本支持该功能。

1. DMC_MoveLinearRelative 指令的参数 Velocity 为终端机构的目标速度，终端机构速度与各轴速度的关系如下：终端机构速度的平方 = 各轴速度的平方和。该指令参数 Acceleration、Deceleration 为终端机构的目标加速度和目标减速度，终端机构的加速度和减速度与各轴加、减速度的关系为：终端机构的加（减）速度 = 各轴加（减）速度的平方和。
2. DMC_MoveLinearRelative 指令参数 Jerk 保留。
3. 参数 BufferMode、TransitionMode 与 TransitionParameter 之间的关系说明如下表所示。BufferMode 选择 mcBuffered 模式时，TransitionMode 仅支持 mcTMNone 模式；BufferMode 选择 mcBlendingPrevious 模式时，TransitionMode 支持 mcTMConstantVelocity 与 mcTMCornerDistance 两种模式。

BufferMode 值	TransitionMode 值	说明
mcBuffered(1)	mcTMNone(0)	等待前一个插补指令的动作正常执行结束后，立即执行当前指令
mcBlending-Previous(3)	mcTMConstant-Velocity(2)	圆滑过渡：等待前一个插补指令执行完成，并立即执行当前指令，交接速度为前一个指令的合成速度，交接完成后终端机构先进行圆滑过渡，然后在做直线运动，如范例二所示。
	mcTMCorner-Distance(3)	角距过渡：等待前一个插补指令执行完成，并立即执行当前指令。前一个插补指令执行过程中，当终端机构与前一个插补指令终点距离等于参数 TransitionParameter 设置的值时，前一个插补指令立即完成，当前指令开始执行。当前指令执行时，终端机构先走过一段圆弧再作直线插补，圆弧终点到前一个插补指令终点的距离仍等于参数 TransitionParameter 的值，如范例三所示。

程序范例一

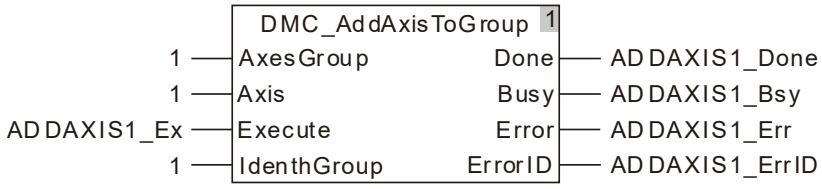
DMC_MoveLinearRelative 指令单独执行的范例如下：

1. **变量和程序**

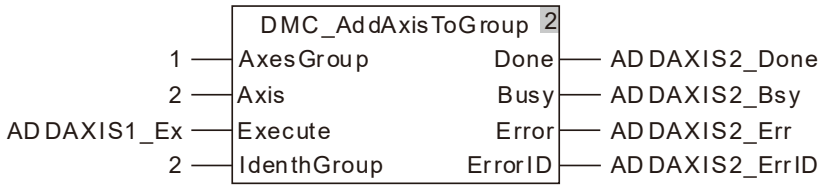
变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	

变量名	数据类型	初始值
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
ADDAXIS3	DMC_AddAxisToGroup	
ADDAXIS3_Done	BOOL	
ADDAXIS3_Bsy	BOOL	
ADDAXIS3_Err	BOOL	
ADDAXIS3_ErrID	WORD	
DMC_GroupEnable	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINREL	DMC_MoveLinearRelative	
LINREL_Ex	BOOL	
LINREL_Dis	ARRAY [1..8] OF LREAL	
LINREL_Done	BOOL	
LINREL_Bsy	BOOL	
LINREL_Act	BOOL	
LINREL_Abt	BOOL	
LINREL_Err	BOOL	
LINREL_ErrID	WORD	

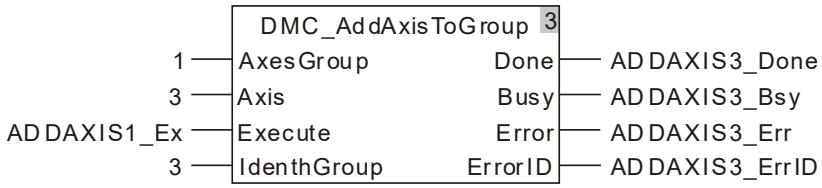
ADDAXIS1



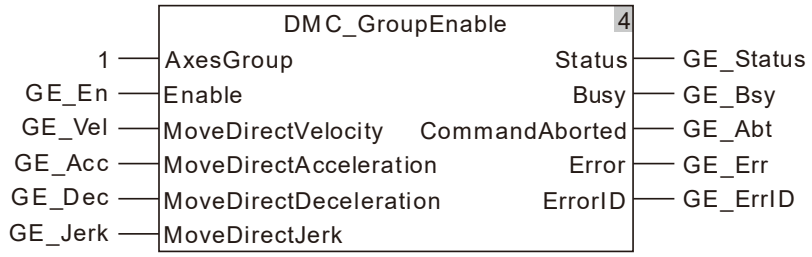
ADDAXIS2



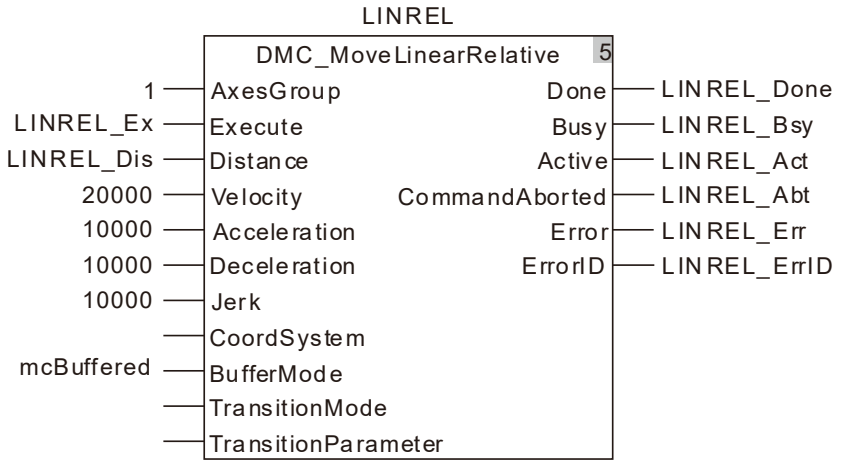
ADDAXIS3



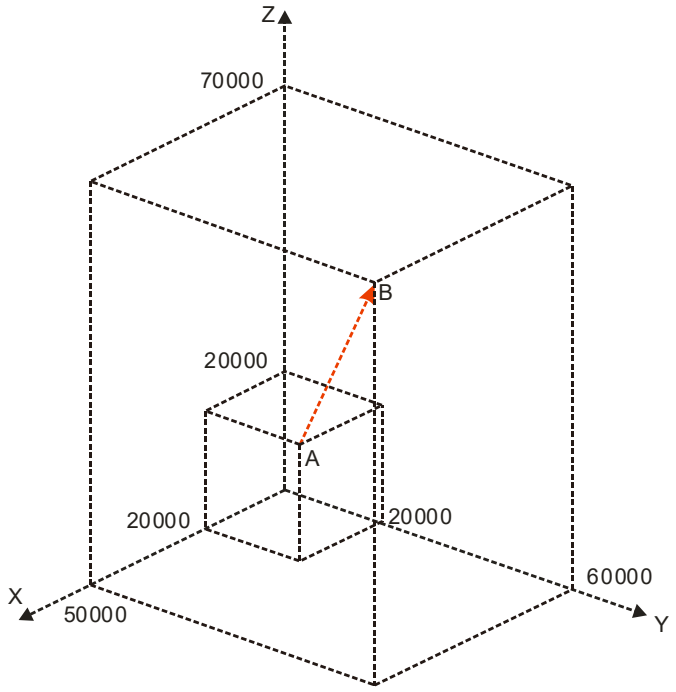
DMC_GroupEnable0



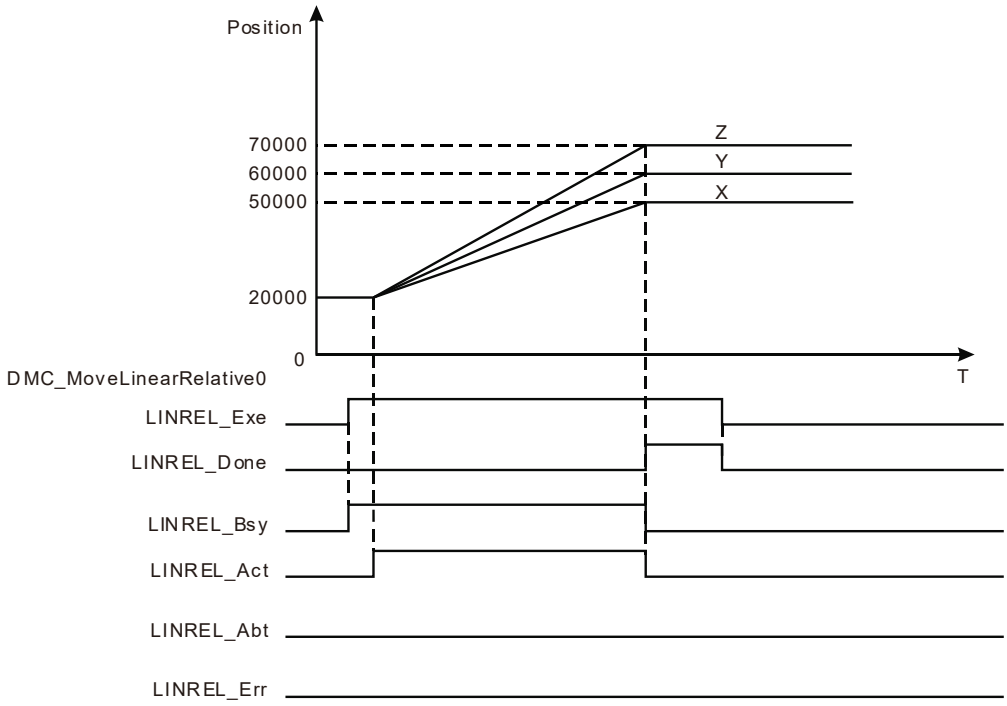
LINREL



2. **指令执行后，整个运动过程如下图所示：**



3. **运动过程中，X 轴、Y 轴和 Z 轴的运动曲线和时序图如下：**



- X 轴、Y 轴和 Z 轴的初始位置均为 20000，设置 DMC_MoveLinearRelative 指令的输入参数 LINREL_Dis[1]、LINREL_Dis[2]和 LINREL_Dis[3]分别为 30000、40000 和 50000。
- 当 LINREL_Ex 变为 TRUE 时，LINREL_Bsy 变为 TRUE，两个周期后，LINREL_Act 变为 TRUE，轴组开始运行。

❖ DMC_MoveLinearRelative 指令完成时，X 轴、Y 轴和 Z 轴同时到达各自目标位置，LINREL_Done 变为 TRUE，LINREL_Bsy 和 LINREL_Act 变为 FALSE。

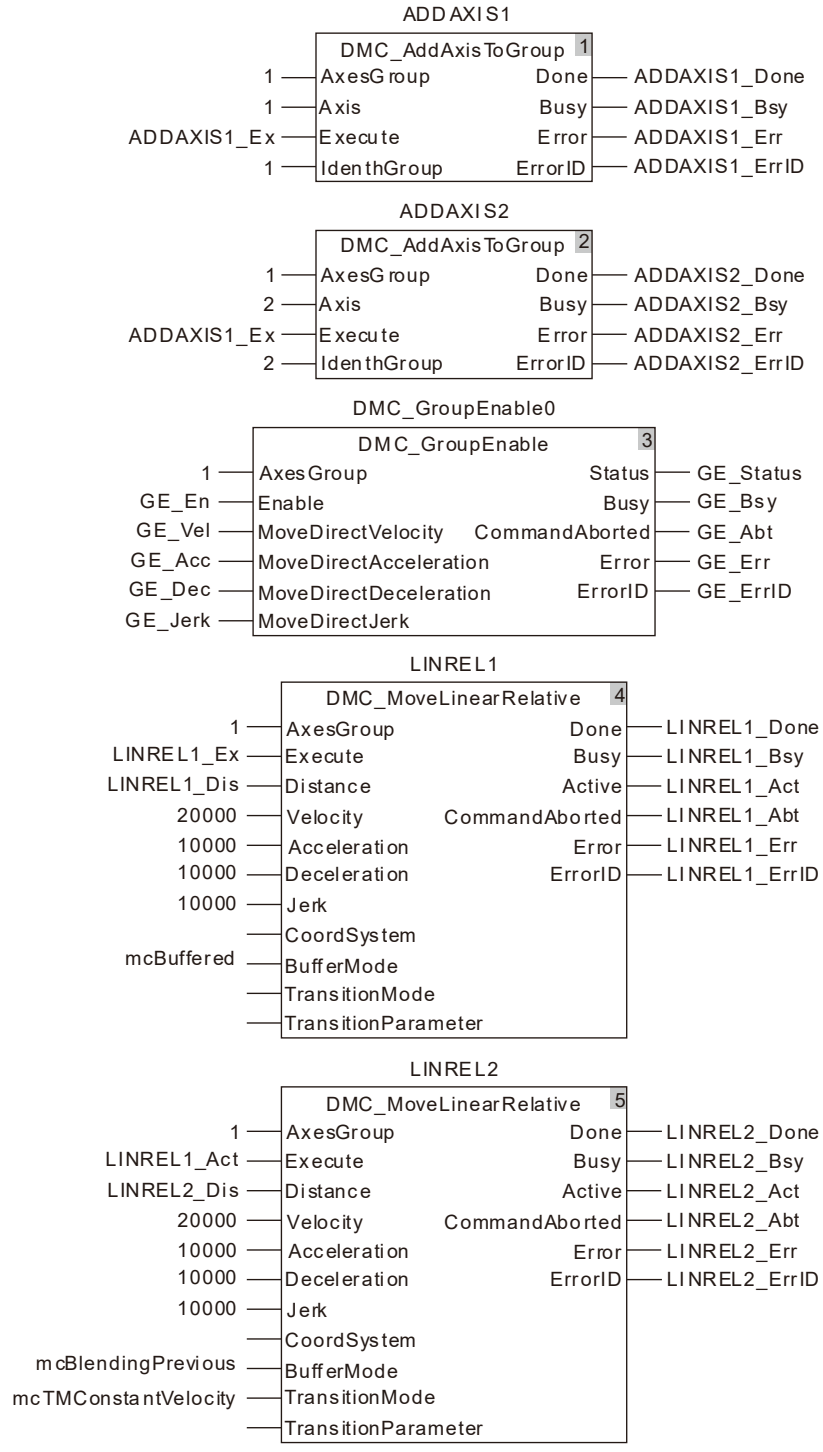
程序范例二

两个 DMC_MoveLinearRelative 指令以 mcTMConstantVelocity 方式交接的范例如下：

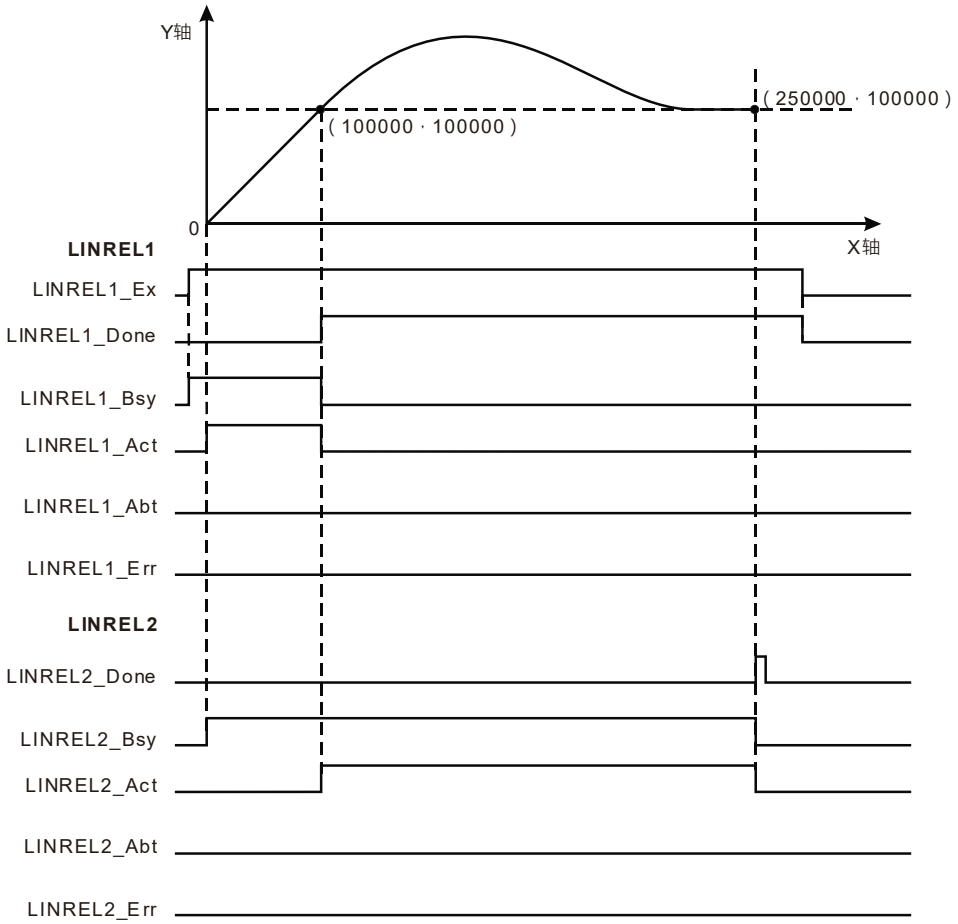
1. 变量和程序

变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINREL1	DMC_MoveLinearRelative	
LINREL1_Ex	BOOL	
LINREL1_Dis	ARRAY [1..8] OF LREAL	
LINREL1_Done	BOOL	
LINREL1_Bsy	BOOL	
LINREL1_Act	BOOL	
LINREL1_Abt	BOOL	
LINREL1_Err	BOOL	
LINREL1_ErrID	WORD	
LINREL2	DMC_MoveLinearRelative	

变量名	数据类型	初始值
LINREL2_Dis	ARRAY [1..8] OF LREAL	
LINREL2_Done	BOOL	
LINREL2_Bsy	BOOL	
LINREL2_Act	BOOL	
LINREL2_Abt	BOOL	
LINREL2_Err	BOOL	
LINREL2_ErrID	WORD	


ADDAXIS1
DMC_AddAxisToGroup 1
1 AxesGroup
1 Axis
ADDAXIS1_Ex Execute
1 IdenthGroup
Done ADDAXIS1_Done
Busy ADDAXIS1_Bsy
Error ADDAXIS1_Err
ErrorID ADDAXIS1_ErrID
ADDAXIS2
DMC_AddAxisToGroup 2
1 AxesGroup
2 Axis
ADDAXIS1_Ex Execute
2 IdenthGroup
Done ADDAXIS2_Done
Busy ADDAXIS2_Bsy
Error ADDAXIS2_Err
ErrorID ADDAXIS2_ErrID
DMC_GroupEnable0
DMC_GroupEnable 3
1 AxesGroup
GE_En Enable
GE_Vel MoveDirectVelocity
GE_Acc MoveDirectAcceleration
GE_Dec MoveDirectDeceleration
GE_Jerk MoveDirectJerk
Status GE_Status
Busy GE_Bsy
CommandAborted GE_Abt
Error GE_Err
ErrorID GE_ErrID
LINREL1
DMC_MoveLinearRelative 4
1 AxesGroup
LINREL1_Ex Execute
LINREL1_Dis Distance
20000 Velocity
10000 Acceleration
10000 Deceleration
10000 Jerk
CoordSystem
mcBuffered BufferMode
TransitionMode
TransitionParameter
Done LINREL1_Done
Busy LINREL1_Bsy
Active LINREL1_Act
CommandAborted LINREL1_Abt
Error LINREL1_Err
ErrorID LINREL1_ErrID
LINREL2
DMC_MoveLinearRelative 5
1 AxesGroup
LINREL1_Act Execute
LINREL2_Dis Distance
20000 Velocity
10000 Acceleration
10000 Deceleration
10000 Jerk
CoordSystem
mcBlendingPrevious BufferMode
mcTMConstantVelocity TransitionMode
TransitionParameter
Done LINREL2_Done
Busy LINREL2_Bsy
Active LINREL2_Act
CommandAborted LINREL2_Abt
Error LINREL2_Err
ErrorID LINREL2_ErrID

2. 终端机构的运行曲线和时序图如下：



- ❖ 设置 LINREL2 的 BufferMode 为 mcBlendingPrevious，TransitionMode 为 mcTMConstantVelocity 模式。
- ❖ 设置 LINREL1 的输入参数 LINREL1_Dis [1]、LINREL1_Dis [2]的值为 100000，设置 LINREL2 的输入参数 LINREL2_Dis [1]、LINREL2_Dis [2]的值分别为 150000、0。
- ❖ 先执行 ADDAXIS1 与 ADDAXIS2，然后执行 DMC_GroupEnable0，当轴组使能成功后，执行指令 LINREL1，然后立即执行 LINREL2。
- ❖ 当终端机构运行到坐标（100000，100000）时，LINREL1_Done 变为 TRUE，同时 LINREL2_Act 变为 TRUE，LINREL2 开始执行，此时终端机构速度为前一个指令的目标速度 20000。
- ❖ 执行 LINREL2 后，终端机构先进行圆滑过渡，然后在做直线运动。当终端机构到达坐标（250000，100000）时，指令完成。

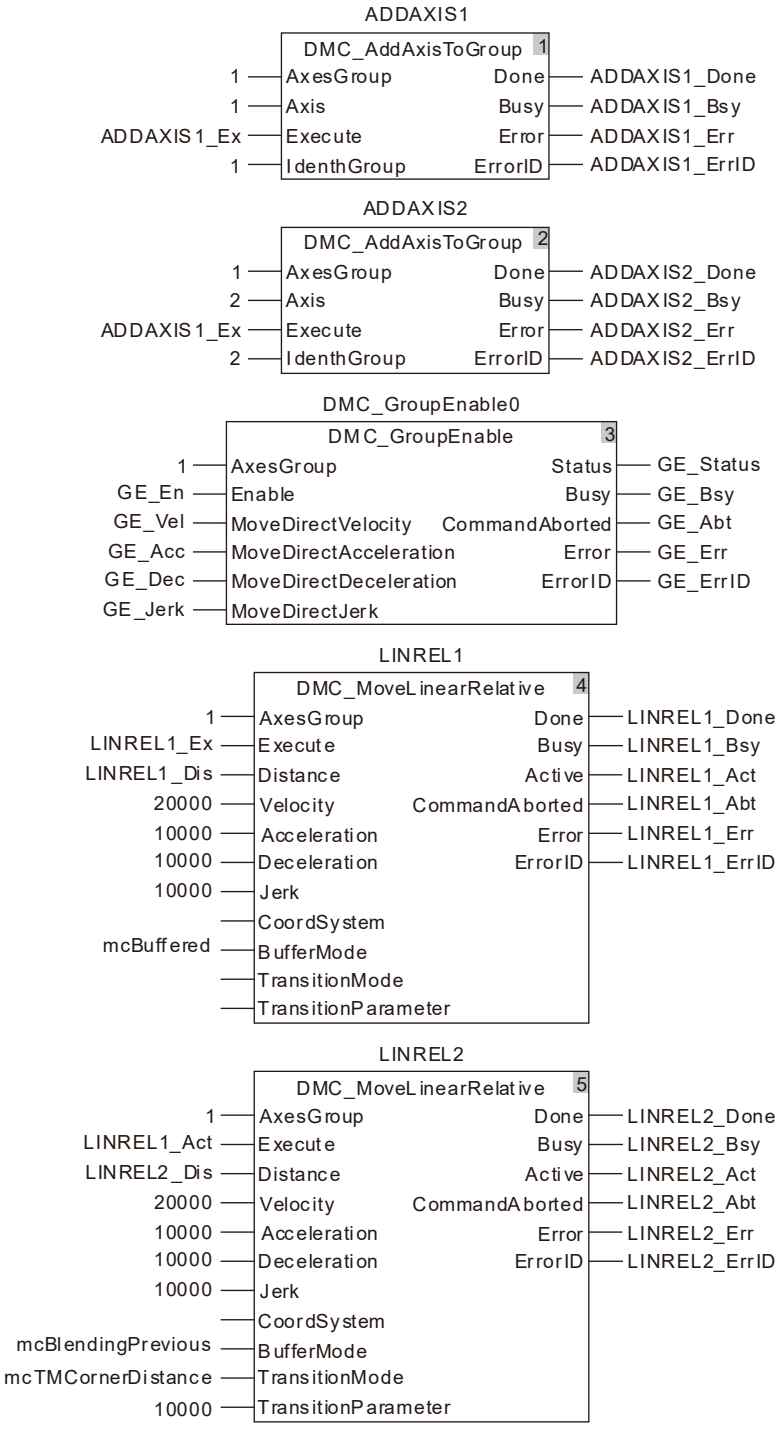
程序范例三

两个 DMC_MoveLinearRelative 指令以 mcTMCornerDistance 方式交接的范例如下：

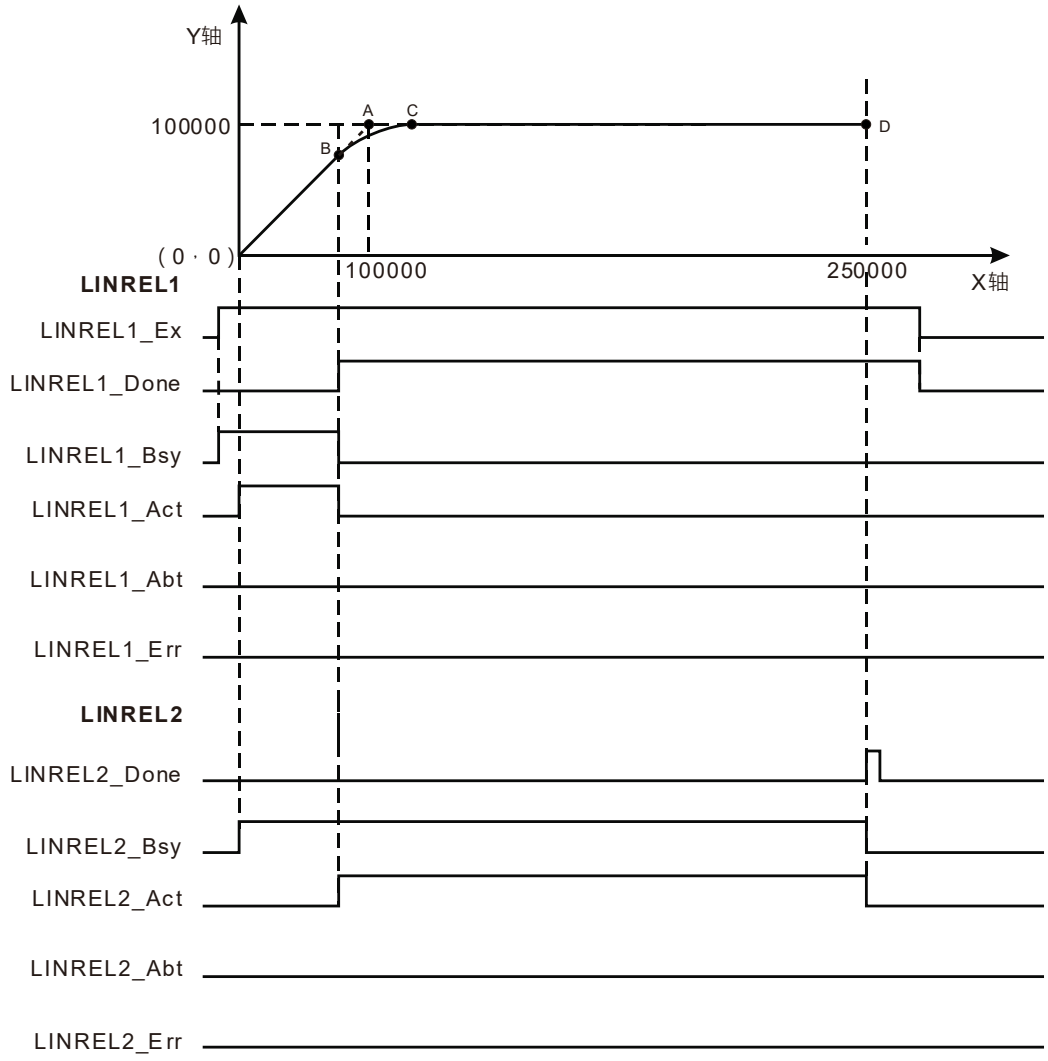
1. 变量和程序

变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	

变量名	数据类型	初始值
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINREL1	DMC_MoveLinearRelative	
LINREL1_Ex	BOOL	
LINREL1_Dis	ARRAY [1..8] OF LREAL	
LINREL1_Done	BOOL	
LINREL1_Bsy	BOOL	
LINREL1_Act	BOOL	
LINREL1_Abt	BOOL	
LINREL1_Err	BOOL	
LINREL1_ErrID	WORD	
LINREL2	DMC_MoveLinearRelative	
LINREL2_Dis	ARRAY [1..8] OF LREAL	
LINREL2_Done	BOOL	
LINREL2_Bsy	BOOL	
LINREL2_Act	BOOL	
LINREL2_Abt	BOOL	
LINREL2_Err	BOOL	
LINREL2_ErrID	WORD	


ADDAXIS1
DMC_AddAxisToGroup 1
1 AxesGroup
1 Axis
ADDAXIS1_Ex Execute
1 IdenthGroup
Done ADDAXIS1_Done
Busy ADDAXIS1_Bsy
Error ADDAXIS1_Err
ErrorID ADDAXIS1_ErrID
ADDAXIS2
DMC_AddAxisToGroup 2
1 AxesGroup
2 Axis
ADDAXIS1_Ex Execute
2 IdenthGroup
Done ADDAXIS2_Done
Busy ADDAXIS2_Bsy
Error ADDAXIS2_Err
ErrorID ADDAXIS2_ErrID
DMC_GroupEnable0
DMC_GroupEnable 3
1 AxesGroup
GE_En Enable
GE_Vel MoveDirectVelocity
GE_Acc MoveDirectAcceleration
GE_Dec MoveDirectDeceleration
GE_Jerk MoveDirectJerk
Status GE_Status
Busy GE_Bsy
CommandAborted GE_Abt
Error GE_Err
ErrorID GE_ErrID
LINREL1
DMC_MoveLinearRelative 4
1 AxesGroup
LINREL1_Ex Execute
LINREL1_Dis Distance
20000 Velocity
10000 Acceleration
10000 Deceleration
10000 Jerk
CoordSystem
mcBuffered BufferMode
TransitionMode
TransitionParameter
Done LINREL1_Done
Busy LINREL1_Bsy
Active LINREL1_Act
CommandAborted LINREL1_Abt
Error LINREL1_Err
ErrorID LINREL1_ErrID
LINREL2
DMC_MoveLinearRelative 5
1 AxesGroup
LINREL1_Act Execute
LINREL2_Dis Distance
20000 Velocity
10000 Acceleration
10000 Deceleration
10000 Jerk
CoordSystem
mcBlendingPrevious BufferMode
mcTMCornerDistance TransitionMode
10000 TransitionParameter
Done LINREL2_Done
Busy LINREL2_Bsy
Active LINREL2_Act
CommandAborted LINREL2_Abt
Error LINREL2_Err
ErrorID LINREL2_ErrID

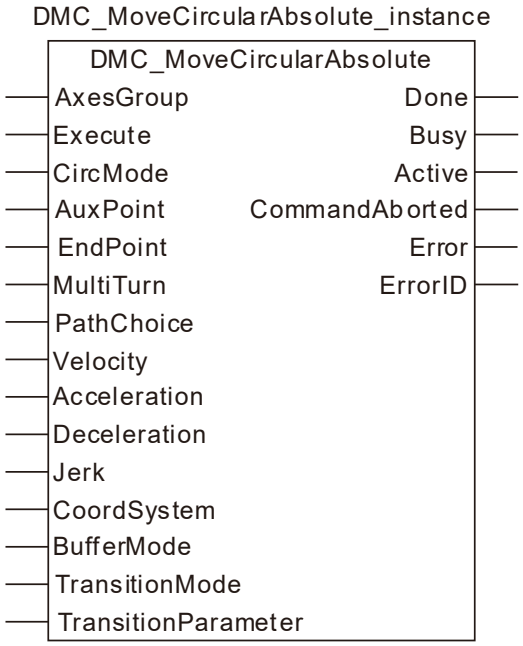
2. 终端机构的运行曲线和时序图如下：



- ❖ 设置 LINREL2 的 BufferMode 为 mcBlendingPrevious，TransitionMode 为 mcTMCornerDistance 模式，TransitionParameter 的值为 10000。
- ❖ 设置 LINREL1 的输入参数 LINREL1_Dis [1]、LINREL1_Dis [2]的值为 100000，设置 LINREL2 的输入参数 LINREL2_Dis [1]、LINREL2_Dis [2]的值分别为 150000、0。
- ❖ 先执行 ADDAXIS1 与 ADDAXIS2，然后执行 DMC_GroupEnable0，当轴组使能成功后，执行指令 LINREL1，然后立即执行 LINREL2。
- ❖ 当终端机构运行到坐标点 B 时，LINREL1_Done 变为 TRUE，同时 LINREL2_Act 变为 TRUE，LINREL2 开始执行。
- ❖ 执行 LINREL2 后，终端机构先走过一段圆弧，当运行到坐标点 C 后，终点机构继续作直线插补。
- ❖ 点 B 到点 A 的距离等于点 C 到点 A 的距离，且该距离等于 LINREL2 的参数 TransitionParameter 的值 10000。

11.7.12 DMC_MoveCircularAbsolute（绝对圆弧插补）

FB/FC	说明	适用机种
FB	此指令用于控制轴执行圆弧插补功能，其中轴的终点位置为绝对位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲控制的轴组编号	USINT	1~8（不可缺省）	Execute 由 FALSE 变为 TRUE
Execute（执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
CircMode（圆弧插补方式）	设定圆弧插补的方式 0：XY 平面圆心法画圆 1：ZX 平面圆心法画圆 2：YZ 平面圆心法画圆 3：XY 平面半径法画圆 4：ZX 平面半径法画圆 5：YZ 平面半径法画圆	INT	0~5（0）	Execute 由 FALSE 变为 TRUE
AuxPoint（圆心坐标或半径）	圆心法画圆时，该参数表示圆心坐标； 半径法画圆时，Auxpoint[1]表示半径，AuxPoint[2]无作用。	ARRAY [1..2] OF LREAL	负数、正数、0（0）	Execute 由 FALSE 变为 TRUE -

名称	功能	数据类型	设定范围 （缺省值）	生效时机
EndPoint （圆弧终点坐标）	圆弧终点各轴的位置	ARRAY [1..8] OF LREAL	负数、正数、0 （0）	Execute 由 FALSE 变为 TRUE -
MultiTurn （圈数）	设定螺旋插补的圈数	UINT	正数、0 （0）	Execute 由 FALSE 变为 TRUE
PathChoice （圆弧方向）	圆弧插补的方向 0：顺时针方向 1：逆时针方向	INT	0、1 （0）	Execute 由 FALSE 变为 TRUE
Velocity （合成速度）	设定的目标合成速度 （单位：单元/秒）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
Acceleration （合成加速度）	设定的目标合成加速度 （单位：单元/秒²）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
Deceleration （合成减速度）	设定的目标合成减速度 （单位：单元/秒²）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
Jerk （合成加速度变化率）	设定的目标合成加速度变化率 （单位：单元/秒³）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
CoordSystem	系统保留			
BufferMode （交接模式）	设定两个轴组指令之间交接模式 1：等待 3：以前一指令的速度交接	MC_Buffer_Mode	1: mcBuffered 3: mcBlendingPrevious	Execute 由 FALSE 变为 TRUE
TransitionMode （过渡模式）	设定两个轴组指令之间的过渡模式 0：无过渡曲线插入 3：以给定的角距过渡	MC_Transition_Mode	0: mcTMNone 3: mcTMCornerDistance	Execute 由 FALSE 变为 TRUE
TransitionParameter （过渡参数）	设定特定过渡模式所需的过渡参数	LREAL	正数、0 （0）	Execute 由 FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 当使用圆心法画圆弧时，输入参数 AuxPoint[1]、AuxPoint[2]的值为圆心与圆弧起点的距离。使用半径法画圆弧时，输入参数 AuxPoint[1]表示半径的值，AuxPoint[2]的值则无意义。
3. 输入参数 EndPoint[1]~EndPoint[8]的值表示各轴终点坐标。
4. Velocity、Acceleration、Deceleration、Jerk 的关系说明请参考第 10.2 节。

5. 关于 BufferMode 的详细内容，请参考第 10.3 节。

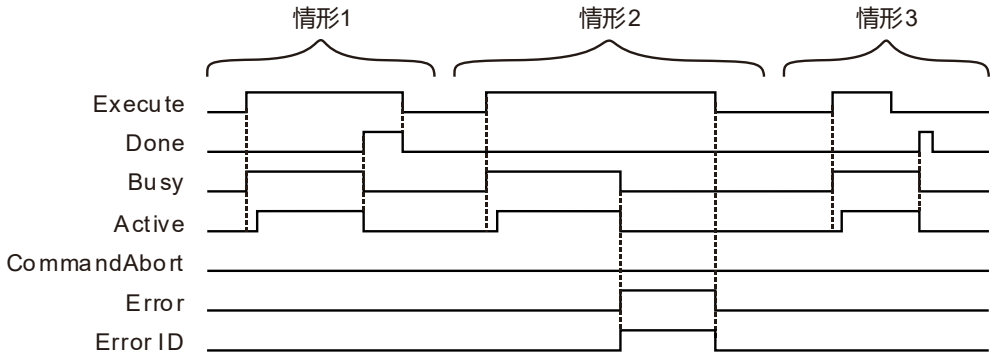
- **输出参数**

名称	功能	数据类型	输出范围
Done （控制中）	该输出参数为 TRUE 时表示指令执行完毕。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被打断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当到达终点位置时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TURE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
CommandAbort	◆ 当指令的执行被其他运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且三个周期后，Active 变为 TRUE；当轴组到达终点位置时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，当有错误产生时(如轴组状态机异常时) Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

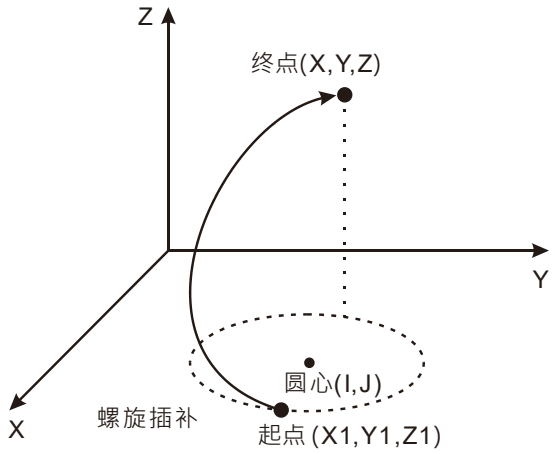
情形3： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，且一个周期后，Done 变为 FALSE。

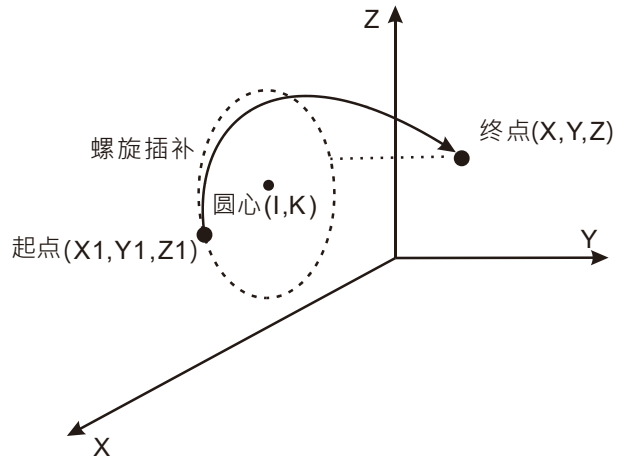
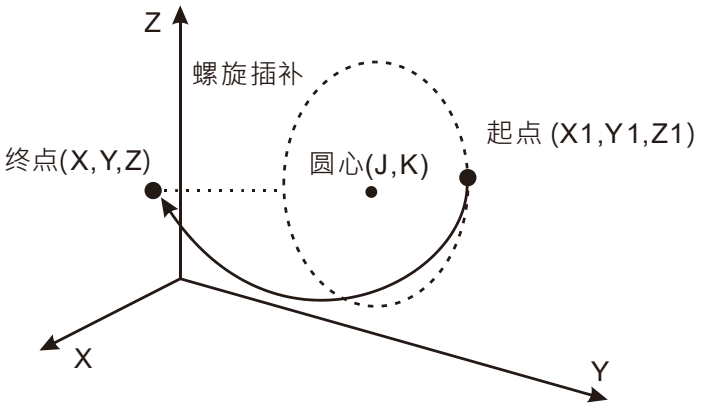
- **功能说明**

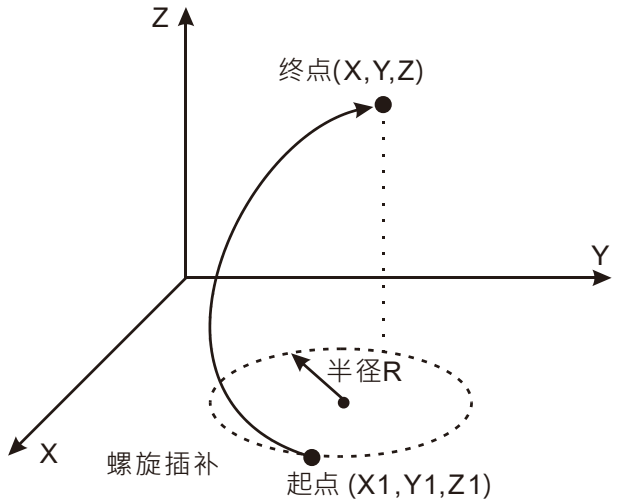
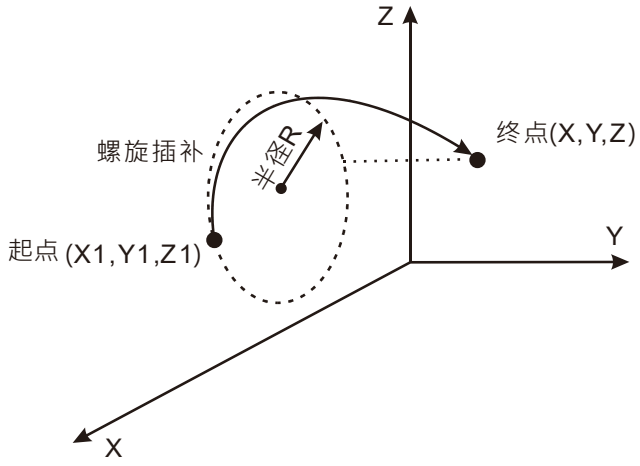
此指令用于轴组进行圆弧插补。V1.01 及以上固件版本支持该功能。

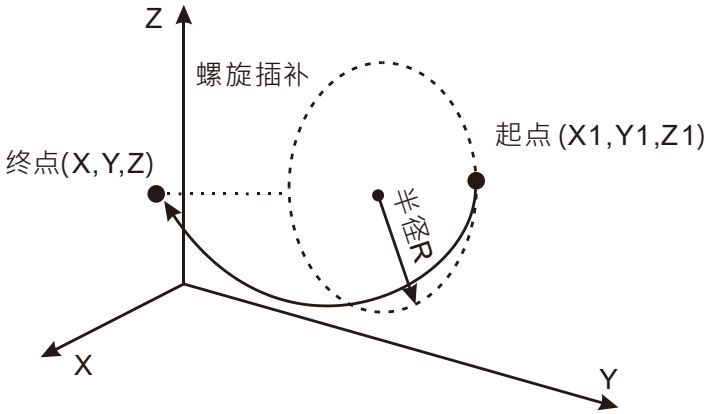
■ CirMode (圆弧插补模式)

CirMode 共有六种模式，具体如下表所示：

CirMode 值	说明
0	XY 平面圆心法画圆。使用该方法时，AuxPoint[1]代表圆心位置相对于起点位置的 X 轴偏移量，AuxPoint[2]代表圆心位置相对于起点位置的 Y 轴偏移量。  上图中，圆心坐标 I=X1+AuxPoint[1]，J=Y1+AuxPoint[2]。 终点坐标 X=EndPoint[1]，Y=EndPoint[2]，Z=EndPoint[3]。
1	ZX 平面圆心法画圆。使用该方法时，AuxPoint[1]代表圆心位置相对于起点位置的 X 轴偏

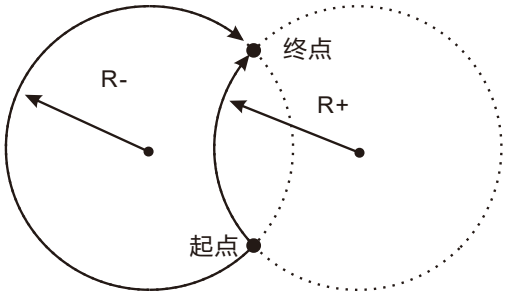
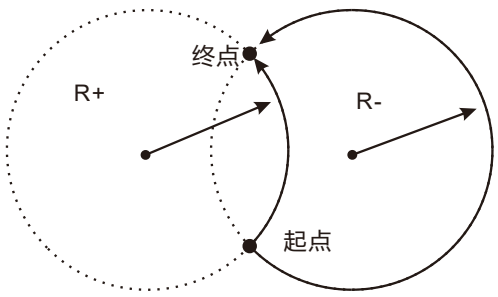
CirMode 值	说明
	移量，AuxPoint[2]代表圆心位置相对于起点位置的 Z 轴偏移量。  上图中，圆心坐标 I=X1+AuxPoint[1]，K=Z1+AuxPoint[2]。 终点坐标 X=EndPoint[1]，Y=EndPoint[2]，Z=EndPoint[3]。
2	YZ 平面圆心法画圆。使用该方法时，AuxPoint[1]代表圆心位置相对于起点位置的 Y 轴偏移量，AuxPoint[2]代表圆心位置相对于起点位置的 Z 轴偏移量。  上图中，圆心坐标 J=Y1+AuxPoint[1]，K=Z1+AuxPoint[2]。 终点坐标 X=EndPoint[1]，Y=EndPoint[2]，Z=EndPoint[3]。
3	XY 平面半径法画圆。使用该方法时，AuxPoint[1]的值表示 XY 平面上圆的半径，此时 AuxPoint[2]的值无意义。 AuxPoint[1]大于 0，圆弧为劣弧；AuxPoint[1]小于 0，圆弧为优弧。

CirMode 值	说明
	 上图中，半径 R=AuxPoint[1]。 终点坐标 X=EndPoint[1]，Y=EndPoint[2]，Z=EndPoint[3]。
4	ZX 平面半径法画圆。使用该方法时，AuxPoint[1]的值表示 ZX 平面上圆的半径，此时 AuxPoint[2]的值无意义。  上图中，半径 R=AuxPoint[1]。 终点坐标 X=EndPoint[1]，Y=EndPoint[2]，Z=EndPoint[3]。
5	XY 平面半径法画圆。使用该方法时，AuxPoint[1]的值表示 XY 平面上圆的半径，此时 AuxPoint[2]的值无意义。

CirMode 值	说明
	 上图中，半径 R=AuxPoint[1]。 终点坐标 X=EndPoint[1]，Y=EndPoint[2]，Z=EndPoint[3]。

■ PathChoice

该参数决定圆弧插补的方向，具体说明如下：

参数值	说明
0	轴组在指定的平面及圆弧上进行顺时针圆弧插补。以半径法为例，圆弧运行轨迹如下图所示：  顺时针运转
1	轴组在指定的平面及圆弧上进行逆时针圆弧插补。以半径法为例，圆弧运行轨迹如下图所示：  逆时针运转

■ BufferMode

指定当前指令和前一个插补指令的交接方式，具体如下表：

BufferMode 值	说明
mcBuffered（1）	等待前一个插补指令的动作正常执行结束后，并立即执行当前指令。
mcBlendingPrevious（3）	当前指令与前一个插补指令以 TransitionMode 指定的模式进行交接。只有当前一个指令为直线插补指令时，DMC_MoveCircularRelative 的 BufferMode 才能选择这种模式。

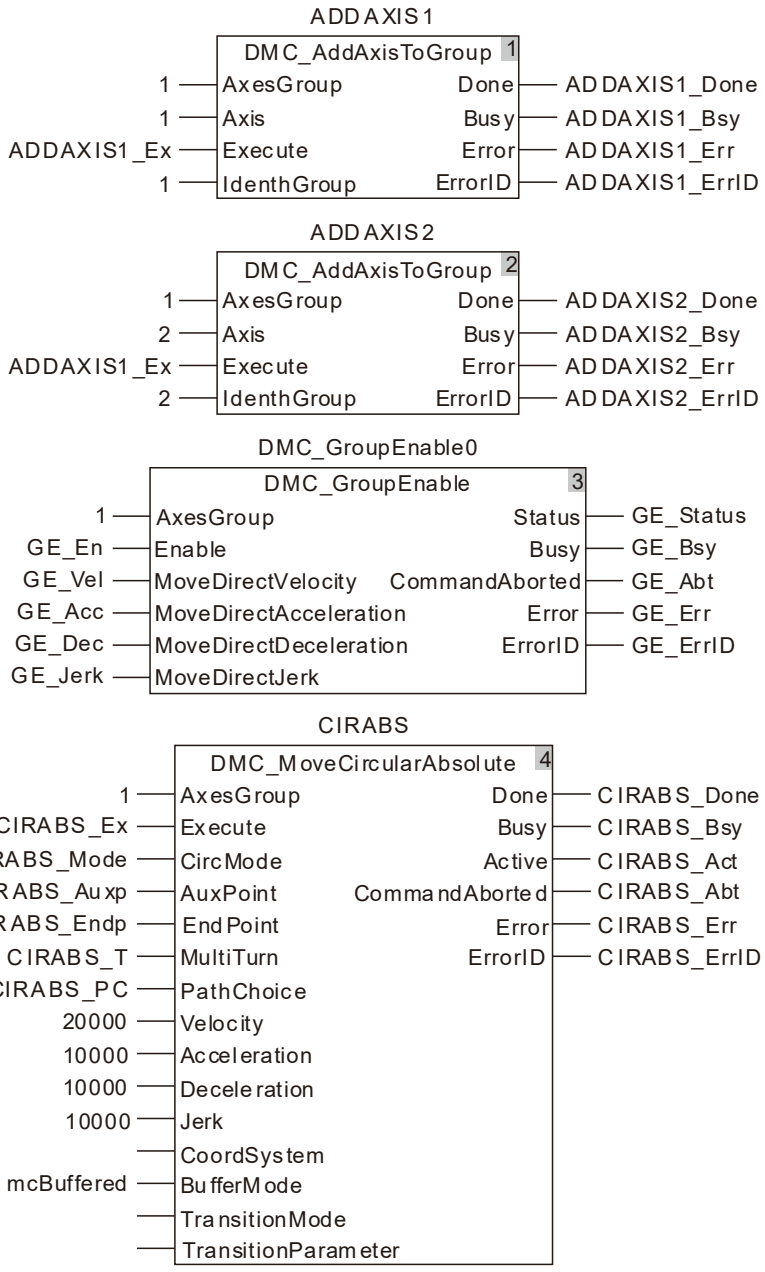
程序范例一

DMC_MoveCircularAbsolute 指令单独执行的范例如下：

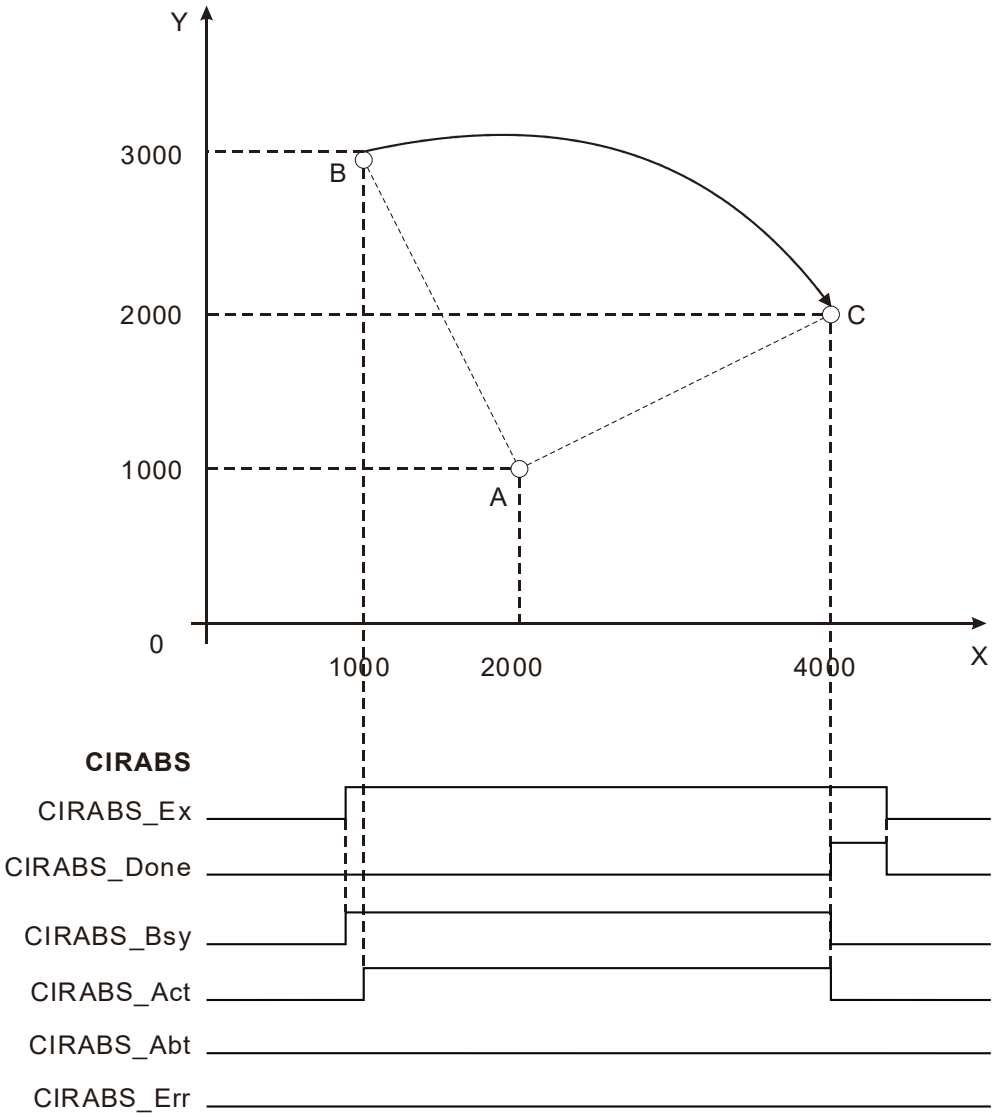
1. 变量和程序

变量名	数据类型	初始值
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ ErrID	WORD	
CIRABS	DMC_MoveCircularAbsolute	
CIRABS_Ex	BOOL	
CIRABS_Mode	INT	0
CIRABS_Auxp	ARRAY [1..2] OF LREAL	

变量名	数据类型	初始值
CIRABS_Endp	ARRAY [1..8] OF LREAL	
CIRABS_T	UINT	0
CIRABS_PC	INT	0
CIRABS_Done	BOOL	
CIRABS_Bsy	BOOL	
CIRABS_Act	BOOL	
CIRABS_Abt	BOOL	
CIRABS_Err	BOOL	
CIRABS_ErrID	WORD	



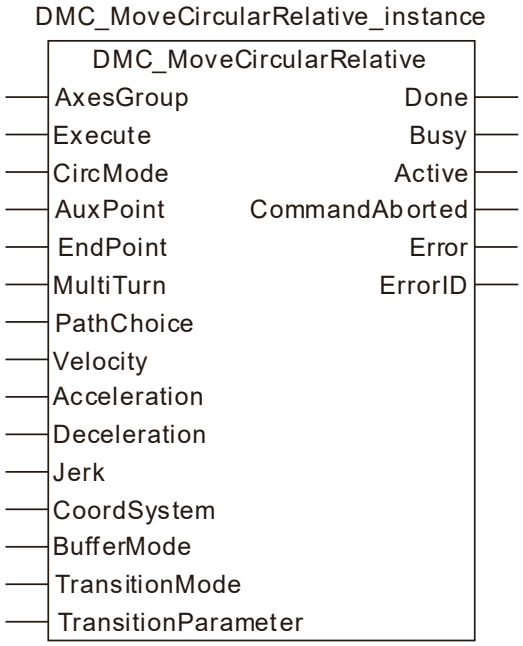
2. 运动过程中，X 轴和 Y 轴的运动曲线和时序图如下：



- 设定 CIRABS_Mode 的值为 0，CIRABS_Auxp[1]的值为 1000，CIRABS_Auxp[2]的值为-2000，CIRABS_Endp[1]的值为 4000，CIRABS_Endp[2]的值为 2000，CIRABS_Mode 的值为 0。
- 上图中，B 点为圆弧的起点（1000，3000），A 点为圆弧所在圆的圆心，坐标为（1000+CIRABS_Auxp[1]=2000，3000+CIRABS_Auxp[2]=1000），C 点为圆弧的终点（CIRABS_Endp[1]=4000，CIRABS_Endp[2]=2000）。
- DMC_MoveCircularAbsolute 指令执行后，从 B 点开始以 A 为圆心顺时针作圆弧插补，当达到终点 C 点时，指令完成。

11.7.13 DMC_MoveCircularRelative（相对圆弧插补）

FB/FC	说明	适用机种
FB	此指令用于控制轴执行圆弧插补功能，其中轴的终点位置为相对位置。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围 （缺省值）	生效时机
AxesGroup （轴组编号）	设定欲控制的轴组编号	USINT	1~8 （不可缺省）	Execute 由 FALSE 变为 TRUE
Execute （执行位）	当 Execute 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE （FALSE）	
CircMode （圆弧插补方式）	设定圆弧插补的方式 0：XY 平面圆心法画圆 1：ZX 平面圆心法画圆 2：YZ 平面圆心法画圆 3：XY 平面半径法画圆 4：ZX 平面半径法画圆 5：YZ 平面半径法画圆	INT	0~5 （0）	Execute 由 FALSE 变为 TRUE
AuxPoint （圆心坐标或半径）	圆心法画圆时，该参数表示圆心坐标； 半径法画圆时， Auxpoint[1]表示半径， AuxPoint[2]无作用。	ARRAY [1..2] OF LREAL	负数、正数、0 （0）	Execute 由 FALSE 变为 TRUE -

名称	功能	数据类型	设定范围 （缺省值）	生效时机
EndPoint （各轴移动距离）	圆弧终点各轴坐标到起点的距离	ARRAY [1..8] OF LREAL	负数、正数、0 （0）	Execute 由 FALSE 变为 TRUE -
MultiTurn （圈数）	设置螺旋插补的圈数	UINT	正数、0 （0）	Execute 由 FALSE 变为 TRUE
PathChoice （圆弧方向）	圆弧插补的方向 0：顺时针方向 1：逆时针方向	INT	0、1 （0）	Execute 由 FALSE 变为 TRUE
Velocity （合成速度）	设定的目标合成速度 （单位：单元/秒）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
Acceleration （合成加速度）	设定的目标合成加速度 （单位：单元/秒²）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
Deceleration （合成减速度）	设定的目标合成减速度 （单位：单元/秒²）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
Jerk （合成加速度变化率）	设定的目标合成加速度变化率 （单位：单元/秒³）	LREAL	正数 （0）	Execute 由 FALSE 变为 TRUE
CoordSystem	系统保留			
BufferMode （交接模式）	设定两个轴组指令之间交接模式 1：等待 3：以前一指令的速度交接	MC_Buffer_Mode	1: mcBuffered 3: mcBlendingPrevious	Execute 由 FALSE 变为 TRUE
TransitionMode （过渡模式）	设定两个轴组指令之间的过渡模式 0：无过渡曲线插入 3：以给定的角距过渡	MC_Transition_Mode	0: mcTMNone 3: mcTMCornerDistance	Execute 由 FALSE 变为 TRUE
TransitionParameter （过渡参数）	设定特定过渡模式所需的过渡参数	LREAL	正数、0 （0）	Execute 由 FALSE 变为 TRUE

说明：

1. 该指令在 Execute 由 FALSE 变为 TRUE 时开始执行，该指令正在执行，Execute 由 TRUE 变为 FALSE 时，对该指令的执行无影响。
2. 当使用圆心法画圆弧时，输入参数 AuxPoint[1]、AuxPoint[2]的值为圆心与圆弧起点的距离。使用半径法画圆弧时，输入参数 AuxPoint[1]表示半径的值，AuxPoint[2]的值则无意义。
3. 输入参数 EndPoint[1]~EndPoint[8]的值表示各轴终点到对应起点的距离。
4. Velocity、Acceleration、Deceleration、Jerk 的关系说明请参考第 10.2 节。

5. 关于 BufferMode 的详细内容，请参考第 10.3 节。

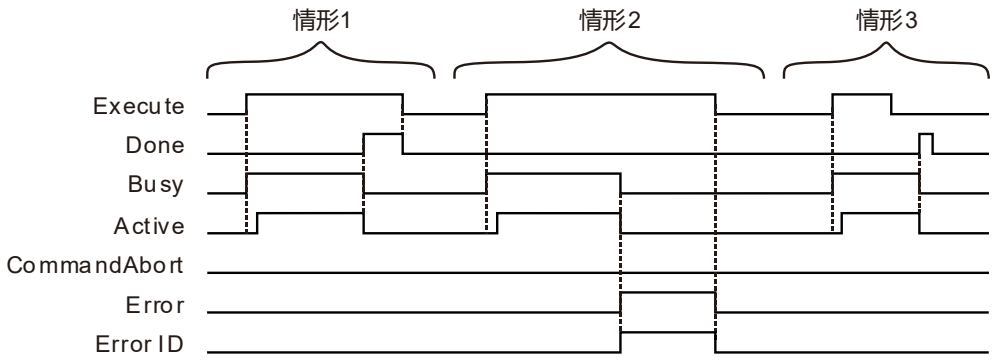
- **输出参数**

名称	功能	数据类型	输出范围
Done （控制中）	该输出参数为 TRUE 时表示指令执行完成。	BOOL	TRUE / FALSE
Busy （执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Active （控制中）	该输出参数为 TRUE 时表示指令正在控制轴组。	BOOL	TRUE / FALSE
CommandAborted （中断）	该输出参数为 TRUE 时表示指令的执行被打断。	BOOL	TRUE / FALSE
Error （错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID （错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Done	◆ 当到达终点位置时	◆ 该指令执行完成后，当 Execute 由 TRUE 变为 FALSE 时 ◆ 该指令执行过程中 Execute 由 TRUE 变为 FALSE 后，在指令执行完成时，Done 变为 TURE，且一个周期后，Done 变为 FALSE
Busy	◆ 当 Execute 为 TRUE 时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
Active	◆ 当指令开始控制轴时	◆ 当 Done 为 TRUE 时 ◆ 当 Error 为 TRUE 时 ◆ 当 CommandAbort 为 TRUE 时
CommandAbort	◆ 当指令的执行被其他运动指令中断时	◆ 当 Execute 由 TRUE 变为 FALSE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Execute 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Execute 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且三个周期后，Active 变为 TRUE；当轴组到达终点位置时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE。

情形2： 当 Execute 由 FALSE 变为 TRUE 时，当有错误产生时(如轴组状态机异常时) Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 和 Active 变为 FALSE；当 Execute 由 TRUE 变为 FALSE 时，Error 变为 FALSE。

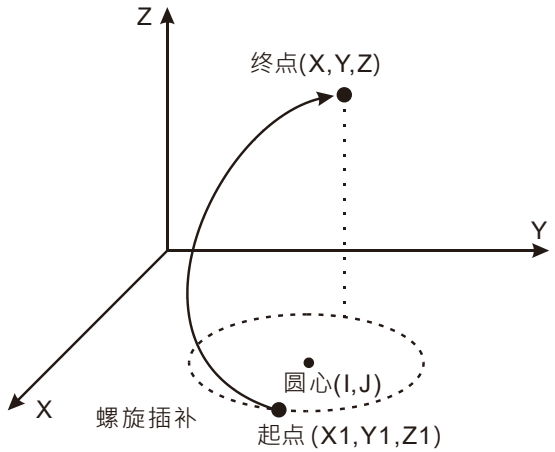
情形3： 在指令执行过程中，Execute 由 TRUE 变为 FALSE 后，指令执行完成时，Done 变为 TRUE，同时 Busy 和 Active 变为 FALSE，且一个周期后，Done 变为 FALSE。

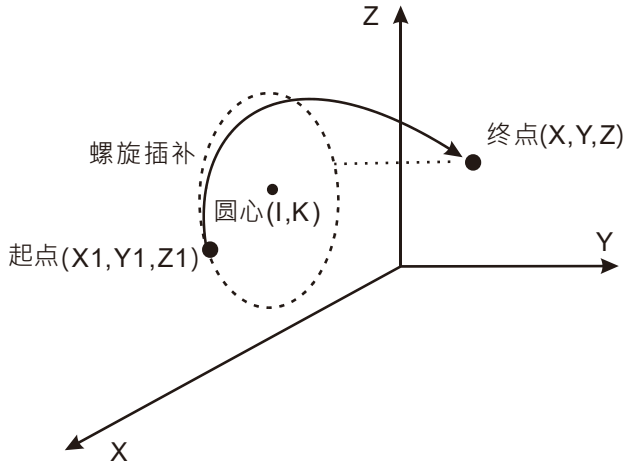
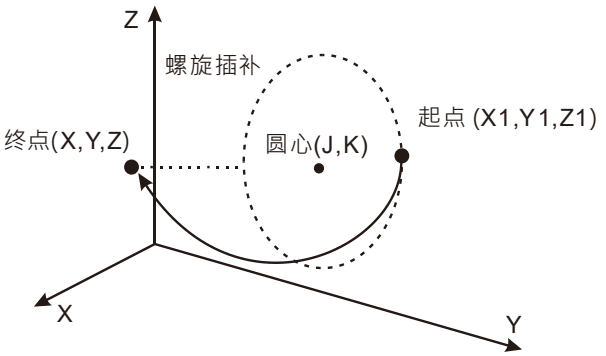
- **功能说明**

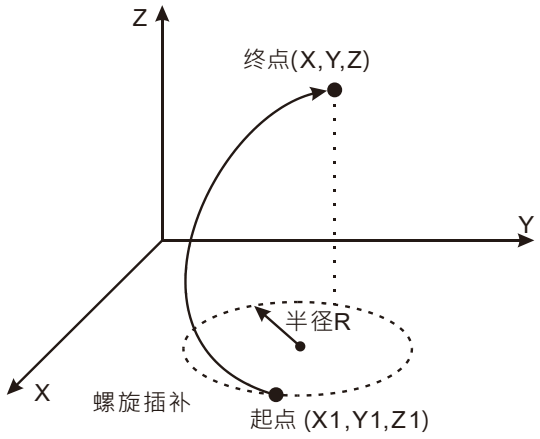
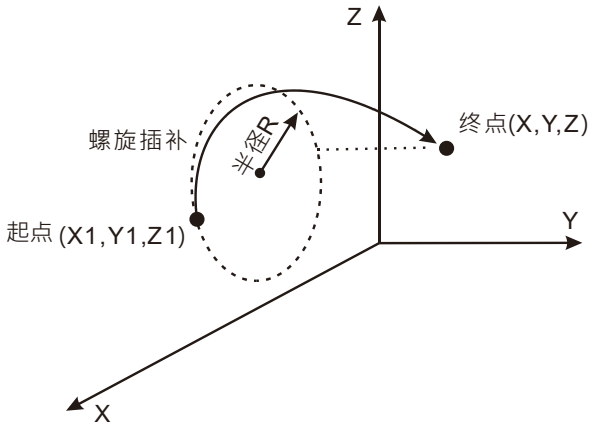
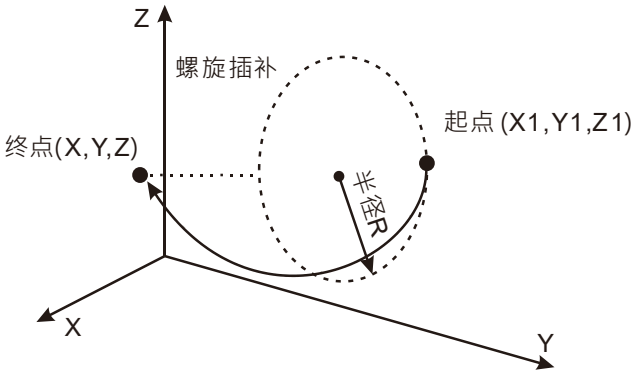
此指令用于轴组进行圆弧插补。V1.01 及以上固件版本支持该功能。

■ CirMode（圆弧插补模式）

CirMode 共有六种模式，具体如下表所示：

CirMode 值	说明
0	XY 平面圆心法画圆。使用该方法时，AuxPoint[1]代表圆心位置相对于起点位置的 X 轴偏移量，AuxPoint[2]代表圆心位置相对于起点位置的 Y 轴偏移量。  上图中，圆心坐标 I=X1+AuxPoint[1]，J=Y1+AuxPoint[2]。 终点坐标 X=X1+EndPoint[1]，Y=Y1+EndPoint[2]，Z=Z1+EndPoint[3]。
1	ZX 平面圆心法画圆。使用该方法时，AuxPoint[1]代表圆心位置相对于起点位置的 X 轴

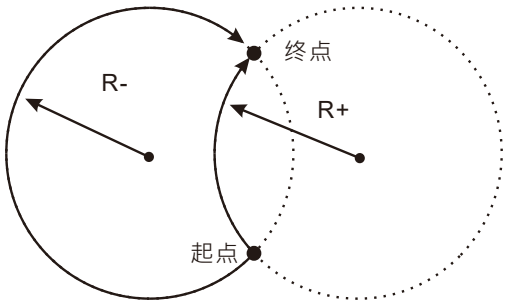
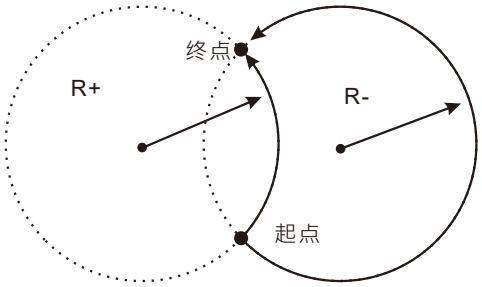
CirMode 值	说明
	偏移量，AuxPoint[2]代表圆心位置相对于起点位置的 Z 轴偏移量。  上图中，圆心坐标 I=X1+AuxPoint[1]，K=Z1+AuxPoint[2]。 终点坐标 X=X1+EndPoint[1]，Y=Y1+EndPoint[2]，Z=Z1+EndPoint[3]。
2	YZ 平面圆心法画圆。使用该方法时，AuxPoint[1]代表圆心位置相对于起点位置的 Y 轴偏移量，AuxPoint[2]代表圆心位置相对于起点位置的 Z 轴偏移量。  上图中，圆心坐标 J=Y1+AuxPoint[1]，K=Z1+AuxPoint[2]。 终点坐标 X=X1+EndPoint[1]，Y=Y1+EndPoint[2]，Z=Z1+EndPoint[3]。
3	XY 平面半径法画圆。使用该方法时，AuxPoint[1]的值表示 XY 平面上圆的半径，此时 AuxPoint[2]的值无意义。 AuxPoint[1]大于 0，圆弧为劣弧；AuxPoint[1]小于 0，圆弧为优弧。

CirMode 值	说明
	 上图中，半径 R=AuxPoint[1]。 终点坐标 X=X1+EndPoint[1]，Y=Y1+EndPoint[2]，Z=Z1+EndPoint[3]。
4	ZX 平面半径法画圆。使用该方法时，AuxPoint[1]的值表示 ZX 平面上圆的半径，此时 AuxPoint[2]的值无意义。  上图中，半径 R=AuxPoint[1]。 终点坐标 X=X1+EndPoint[1]，Y=Y1+EndPoint[2]，Z=Z1+EndPoint[3]。
5	XY 平面半径法画圆。使用该方法时，AuxPoint[1]的值表示 XY 平面上圆的半径，此时 AuxPoint[2]的值无意义。 

CirMode 值	说明
	上图中，半径 R=AuxPoint[1]。 终点坐标 X=X1+EndPoint[1]，Y=Y1+EndPoint[2]，Z=Z1+EndPoint[3]。

- PathChoice

  该参数决定圆弧插补的方向，具体说明如下：

参数值	说明
0	轴组在指定的平面及圆弧上进行顺时针圆弧插补。以半径法为例，圆弧运行轨迹如下图所示：  顺时针运转
1	轴组在指定的平面及圆弧上进行逆时针圆弧插补。以半径法为例，圆弧运行轨迹如下图所示：  逆时针运转

- BufferMode

  指定当前指令和前一个插补指令的交接方式，具体如下表：

BufferMode 值	说明
mcBuffered（1）	等待前一个插补指令的动作正常执行结束后，并立即执行当前指令。
mcBlendingPrevious（3）	当前指令与前一个插补指令以 TransitionMode 指定的模式进行交接。只有当前一个指令为直线插补指令时，DMC_MoveCircularRelative 的 BufferMode 才能选择这种模式。

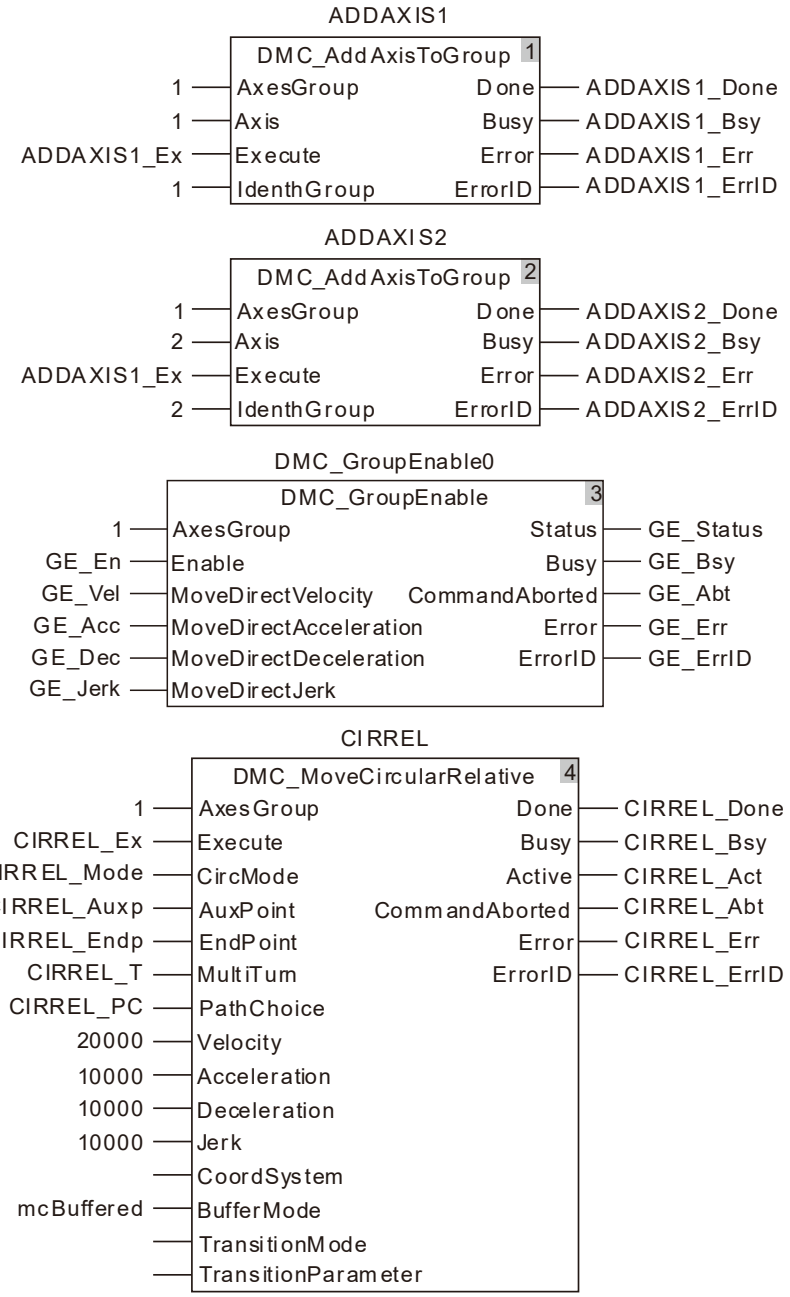
程序范例

DMC_MoveCircularRelative 指令单独执行的范例如下：

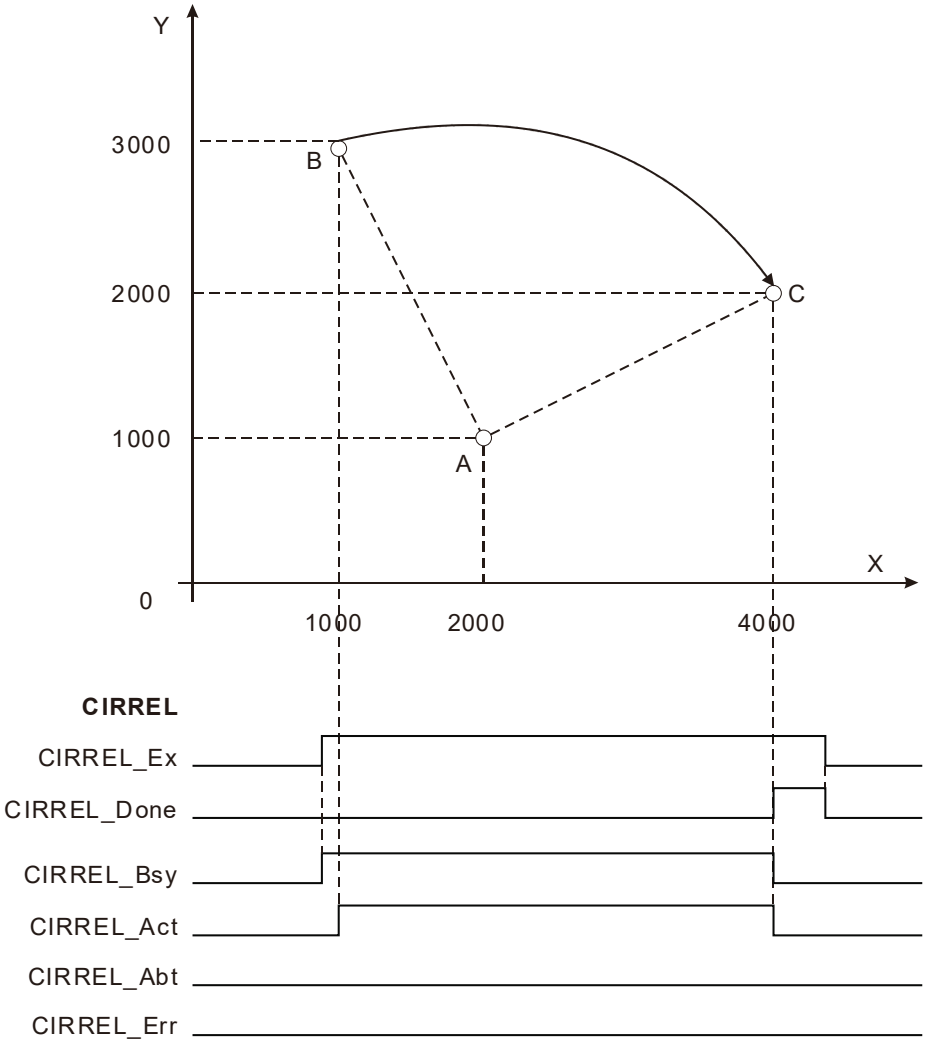
1. 变量和程序

变量名	数据类型	初始值
M1	BOOL	
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] OF LREAL	
GE_Acc	ARRAY [1..8] OF LREAL	
GE_Dec	ARRAY [1..8] OF LREAL	
GE_Jerk	ARRAY [1..8] OF LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
CIRREL	DMC_MoveCircularRelative	
CIRREL_Ex	BOOL	
CIRREL_Mode	INT	0
CIRREL_Auxp	ARRAY [1..2] OF LREAL	
CIRREL_Endp	ARRAY [1..8] OF LREAL	
CIRREL_T	UINT	0
CIRREL_PC	INT	0
CIRREL_Done	BOOL	
CIRREL_Bsy	BOOL	
CIRREL_Act	BOOL	
CIRREL_Abt	BOOL	

变量名	数据类型	初始值
CIRREL_Err	BOOL	
CIRREL_ErrID	WORD	



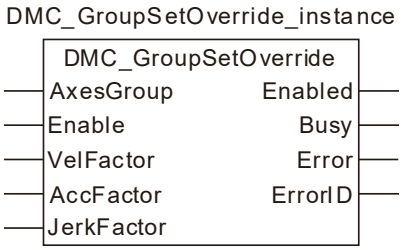
2. 运动过程中，X 轴和 Y 轴的运动曲线和时序图如下：



- 设定 CIRREL_Mode 的值为 0，CIRREL_Auxp[1]的值为 1000，CIRREL_Auxp[2]的值为-2000，CIRREL_Endp[1]的值为 3000，CIRREL_Endp[2]的值为-1000，CIRREL_Mode 的值为 0。
- 上图中，B 点为圆弧的起点（1000，3000），A 点为圆弧所在圆的圆心，坐标为（1000+CIRREL_Auxp[1]=2000，3000+CIRREL_Auxp[2]=1000），C 点为圆弧的终点（1000+CIRREL_Endp[1]=4000，3000+CIRREL_Endp[2]=2000）。
- DMC_MoveCircularRelative 指令执行后，从 B 点开始以 A 为圆心顺时针作圆弧插补，当达到终点 C 点时，指令完成。

11.7.14 DMC_GroupSetOverride（设置轴组超调值）

FB/FC	说明	适用机种
FB	此指令用于设置轴组的速度超调值。	DVP15MC11T DVP15MC11T-06



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲操作的轴组编号	USINT	1~8（不可缺省）	Enable 为 TRUE 时
Enable（执行位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
VelFactor	超调值，单位为%，如“100”表示 100%。	LREAL	0~500（不可缺省）	Enable 为 TRUE 时
AccFactor	系统保留	-	-	-
JerkFactor	系统保留	-	-	-

- **输出参数**

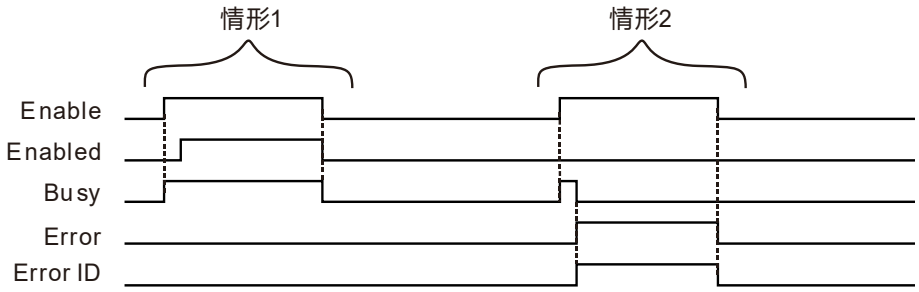
名称	功能	数据类型	输出范围
Enabled（控制中）	该输出参数为 TRUE 时表示正在控制轴组。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Enabled	◆ 当开始执行指令时	◆ 当 Enable 变为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当 Enable 为 TRUE 时	◆ 当 Enable 变为 FALSE 时

名称	变为 TRUE 的时机	变为 FALSE 的时机
		◆ 当 Error 变为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且一个周期后，Enabled 变为 TRUE；当 Enable 由 TRUE 变为 FALSE 时，Enabled 位和 Busy 位同时变为 FALSE。

情形2： 当指令执行出错时，Error 变为 TRUE，ErrorID 显示对应的错误码，同时 Busy 变为 FALSE；当 Enable 变为 FALSE 时，Error 变为 FALSE，ErrorID 变 为零。

- **功能说明**

此指令用于设置轴组的速度超调值。V1.01 及以上固件版本支持该功能。

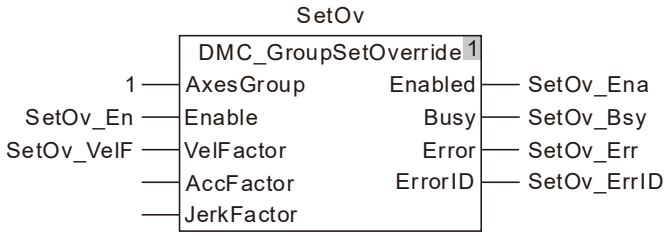
1. 可变更目标速度的指令有：DMC_MoveDirectAbsolute、DMC_MoveDirectRelative、DMC_MoveLinearAbsolute、DMC_MoveLinearRelative、DMC_MoveCircularAbsolute、DMC_MoveCircularRelative。
2. VelFactor 的单位为%，“100”表示“100%”。VelFactor 的有效范围为 0~500，超出有效范围时执行此指令，此指令会报错。
3. 执行此指令后，如果此时正在执行的指令为 DMC_MoveDirectAbsolute 或 DMC_MoveDirectRelative，则轴组中各轴新的目标速度 = 当前各轴的目标速度 * 速度超调值；如果此时正在执行的指令为 DMC_MoveLinearAbsolute、DMC_MoveLinearRelative、DMC_MoveCircularAbsolute 或 DMC_MoveCircularRelative，则新的轴组内各轴的合成目标速度 = 当前各轴的合成目标速度 * 速度超调值。
4. 执行此指令后，轴组会根据当前执行指令的加速度（或减速度）加速（或减速）到变更后的目标速度。
5. 变更后的目标速度超过轴参数的最高转速时，轴会发生错误。
6. 将 VelFactor 设置为 0 时，目标速度变为“0”，轴组以速度 0 动作。
7. Enable 为 TRUE 时，修改 VelFactor 会立即生效，不需要重启此指令；当 Enable 为 TRUE 时，修改 VelFactor 的值超出有效范围，此指令会立即报错，目标速度会返回 100%。
8. 当 Enable 变为 FALSE 时，以 VelFactor=100 为目标进行加速或减速。

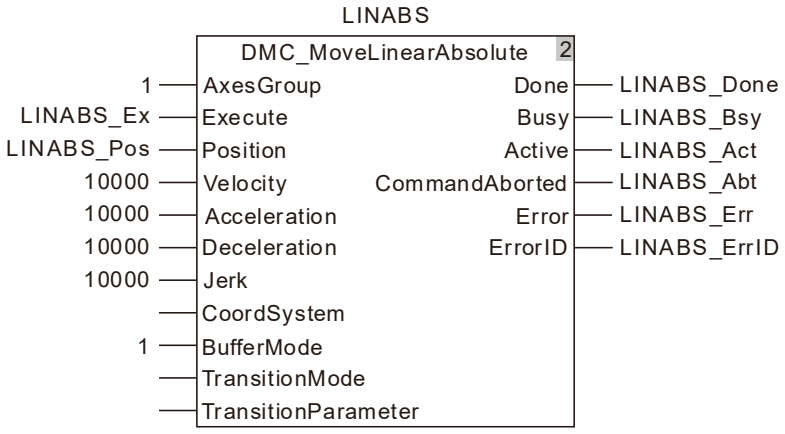
程序范例

DMC_GroupSetOverride 指令执行时的范例如下所示：

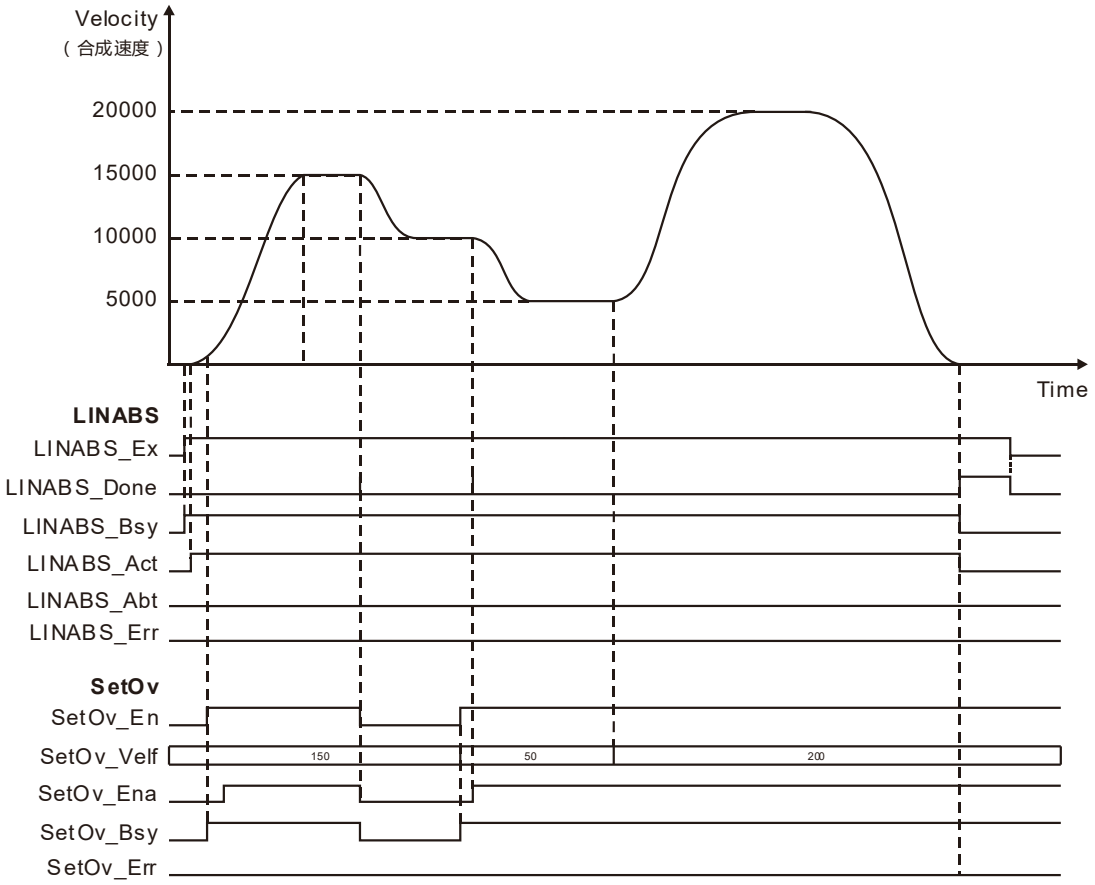
1. 变量和程序

变量名	数据类型	初始值
M1	BOOL	
SetOv	DMC_GroupSetOverride	
SetOv_En	BOOL	
SetOv_VelF	LREAL	
SetOv_Ena	BOOL	
SetOv_Bsy	BOOL	
SetOv_Err	BOOL	
SetOv_ErrID	WORD	
LINABS	DMC_MoveLinearAbsolute	
LINABS_Ex	BOOL	
LINABS_Pos	ARRAY [1..8] OF LREAL	
LINABS_Done	BOOL	
LINABS_Bsy	BOOL	
LINABS_Act	BOOL	
LINABS_Abt	BOOL	
LINABS_Err	BOOL	
LINABS_Eid	WORD	





2. 运动曲线和时序图

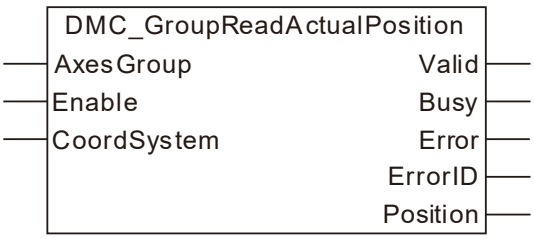


- ❖ 当 LINABS_Ex 变为 TRUE 是，LINABS_Bsy 变为 TRUE，两个周期后，LINABS_Act 变为 TRUE，轴组开始运动。轴组还未到达目标速度时，SetOv_En 设置为 TRUE，DMC_GroupSetOverride 指令生效，轴组的目标速度会变成新的目标速度。
- ❖ 当 SetOv_En 变为 FALSE 时，轴组的目标速度变为 10000。
- ❖ DMC_GroupSetOverride 指令执行过程中修改 SetOv_VelF 的数值，SetOv_VelF 的数值会立即生效，DMC_MoveLinearAbsolute 指令的目标速度会相应的改变。

11.7.15 DMC_GroupReadActualPosition（读取轴组位置指令）

FB/FC	说明	适用机种
FB	此指令用于读取轴组中各轴的位置。	DVP15MC11T DVP15MC11T-06

DMC_GroupReadActualPosition_instance



- **输入参数**

名称	功能	数据类型	设定范围（缺省值）	生效时机
AxesGroup（轴组编号）	设定欲操作的轴组编号	USINT	1~8（不可缺省）	Enable 为 TRUE 时
Enable（使能位）	当 Enable 由 FALSE 变 TRUE 时，执行该指令。	BOOL	TRUE 或 FALSE（FALSE）	
CoorSystem	系统保留	-	-	-

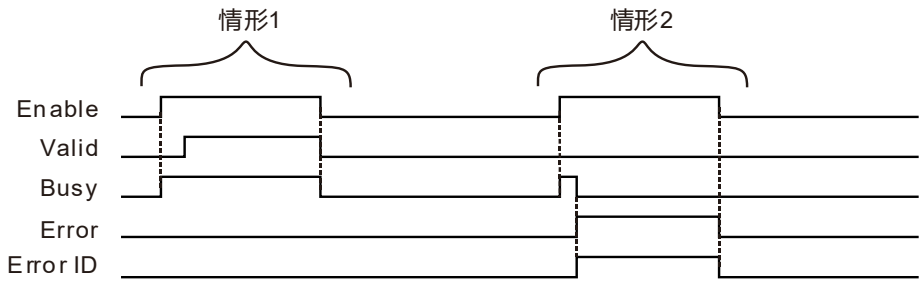
- **输出参数**

名称	功能	数据类型	输出范围
Valid（输出有效）	该输出参数为 TRUE 时表示指令的输出有效。	BOOL	TRUE / FALSE
Busy（执行中）	该输出参数为 TRUE 时表示指令正在执行中。	BOOL	TRUE / FALSE
Error（错误）	该输出参数为 TRUE 时表示指令的执行出错。	BOOL	TRUE / FALSE
ErrorID（错误代码）	指令执行出错时的错误代码。 对应的错误代码请参考第 12.2 节。	WORD	
Position（位置）	轴组中各轴的位置	ARRAY [1..8] OF LREAL	

- **输出参数刷新时机**

名称	变为 TRUE 的时机	变为 FALSE 的时机
Valid	◆ 当指令读到轴组中各轴的位置时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 变为 TRUE 时
Busy	◆ 当该指令执行时	◆ 当 Enable 由 TRUE 变为 FALSE 时 ◆ 当 Error 为 TRUE 时
Error	◆ 当指令输入参数不合法或指令执行过程中出错时	◆ 当 Enable 由 TRUE 变为 FALSE 时

- **输出参数变化时序图**



情形1： 当 Enable 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，且一个周期后，Valid 变为 TRUE；当 Enable 由 TRUE 变为 FALSE 时，Valid 和 Busy 同时变为 FALSE。

情形2： 当指令输入参数有误时，Enable 由 FALSE 变为 TRUE 时，Busy 变为 TRUE，一个周期后，Error 变为 TRUE，ErrorID 显示错误码，同时 Busy 变为 FALSE；当 Enable 由 TRUE 变为 FALSE 时，Error 变为 FALSE，ErrorID 清零。

- **功能说明**

此指令用于读取轴组中各轴的当前位置，指令输出参数 Position 是一个数组，数组中每个成员对应一个轴的位置，如 Position[1]对应 X 轴的位置，Position[2]对应 Y 轴的位置，以此类推。V1.01 及以上固件版本支持该功能。

MEMO

12

第12章　故障诊断

目录

12.1 LED 灯指示说明

- **PWR LED 灯显示说明**

PWR LED 用于显示 DVP-15MC 系列运动控制器电源供电状态。

LED 灯状态	显示说明	处理方法
绿灯亮	电源正常。	无需处理。
灯灭或闪烁	电源异常。	检查 DVP-15MC 系列运动控制器供电电源是否正常。

- **RUN LED 灯显示说明**

RUN LED 用于显示 DVP-15MC 系列运动控制器的程序运行状态。

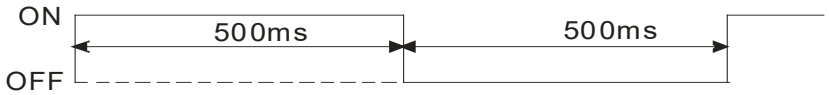
LED 灯状态	显示说明	处理方法
绿灯亮	DVP-15MC 系列运动控制器处于运行状态。	无需处理。
灯灭	DVP-15MC 系列运动控制器处于停止状态。	根据需要将 PLC 切换为运行状态。

- **ERR LED 灯显示说明**

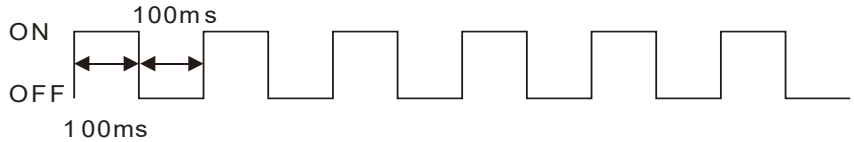
ERR LED 用于显示 DVP-15MC 系列运动控制器的错误状态。

LED 灯状态	显示说明	处理方法
灯灭	DVP-15MC 系列运动控制器处于正常状态。	无需处理。
红灯闪烁	程序错误或配置错误	根据错误诊断功能获取具体错误信息。
红灯常亮	硬件错误	请联系技术人员

■ **ERR LED 红灯闪烁（1HZ）**



■ **ERR LED 红灯快速闪烁（10HZ）**



- **SD LED 灯显示说明**

SD LED 用于显示 DVP15MC11T SD 卡的状态

LED 状态	显示说明	处理方法
灯灭	1. DVP15MC11T 中无 SD 卡插入 2. 读写文件出错	根据需求决定是否插入 SD 卡。

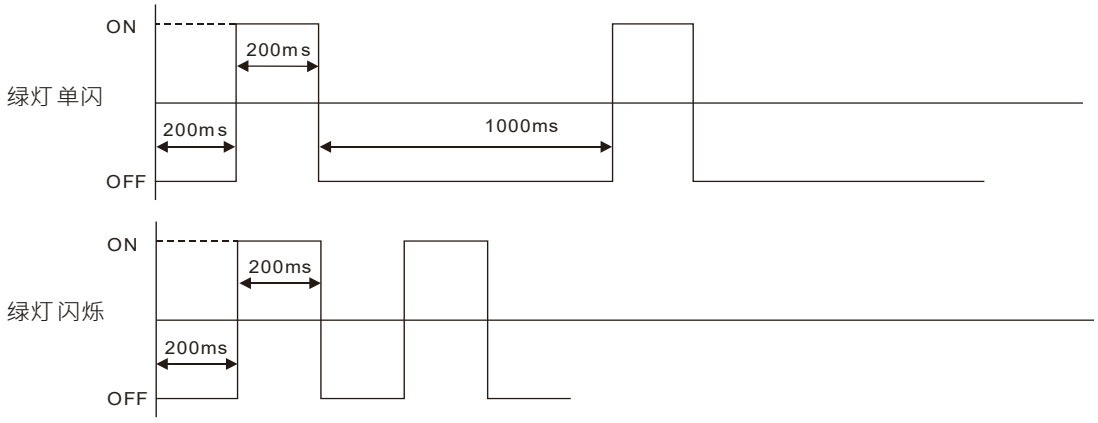
LED 状态	显示说明	处理方法
绿灯快速闪烁	DVP15MC11T 中 SD 卡正在数据交换	无需处理
绿灯常亮	DVP15MC11T 中 SD 卡无数据交换	-

- **CAN1（CANopen） LED 灯显示说明**

  - **RUN LED 灯显示说明**

LED 灯状态	显示说明	处理方法
绿灯单闪	CAN1（CANopen）通讯口处于停止状态	上位机正在下载网络配置，等待下载完成。
绿灯闪烁	CAN1（CANopen） 通讯口处于预运行状态	1. 检查 CANopen 网络中总线线缆接线正确。 2. 检查 CANopen 总线线缆为台达标准 CANopen 线缆。 3. 检查 CANopen 总线两端有接终端电阻。 4. 检查主站和其它从站的波特率相同。 5. 检查网络配置的所有从站实际连接至网络中。 6. 检查是否有从站不能和主站建立连接。 7. 检查是否有从站掉线。
绿灯常亮	CAN1（CANopen） 通讯口处于运行状态	无需处理

  - **RUN LED 绿灯单闪和绿灯闪烁的区别：**

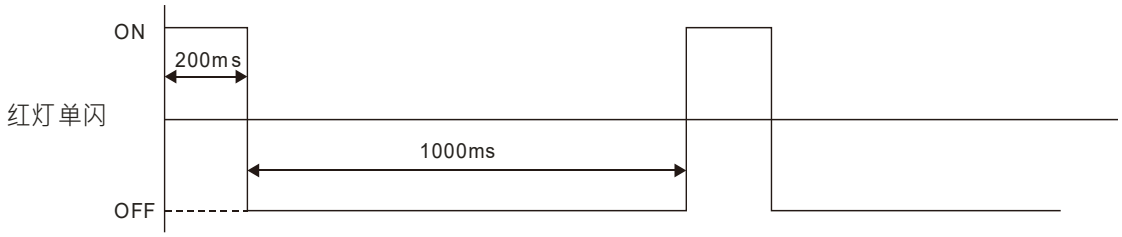


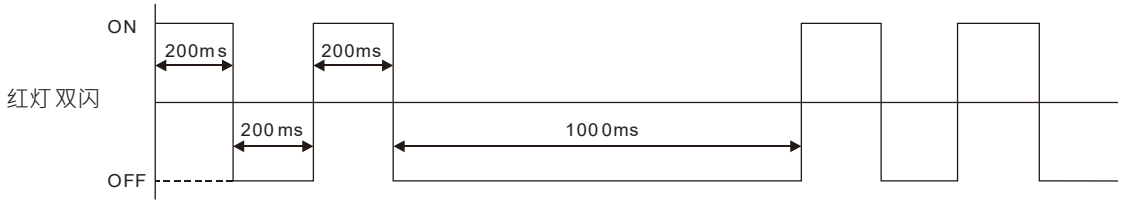
  - **ERR LED 灯显示说明**

LED 灯状态	显示说明	处理方法
灯灭	正常	无需动作
红灯双闪	有从站掉线	1. 检查 CANopen 总线线缆为台达标准 CANopen 线缆。 2. 检查 CANopen 总线两端有接终端电阻。 3. 检查网络配置的从站实际连接至网络中。 4. 检查 CANopen 总线线缆周围是否干扰过大。

LED 灯状态	显示说明	处理方法
红灯单闪	总线错误超出警戒水平	1. 检查 CANopen 网络中总线线缆接线正确。 2. 检查 CANopen 总线线缆为台达标准 CANopen 线缆。 3. 检查 CANopen 总线两端有接终端电阻。 4. 检查 CANopen 通讯口和其它从站的波特率相同。 5. 检查 CANopen 总线线缆周围是否干扰过大。
红灯常亮	总线脱离（Bus-off）	

■ **ERR LED 红灯单闪和红灯双闪的区别：**



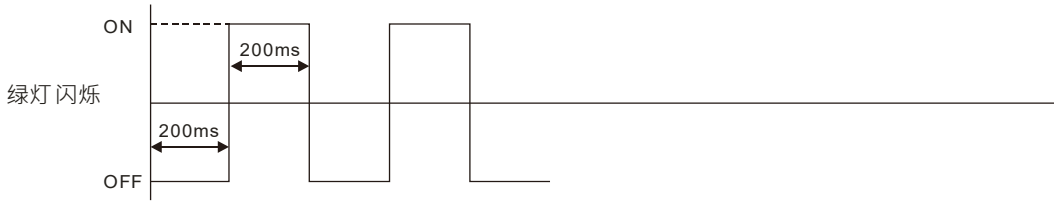


● **CAN2（Motion） LED 灯显示说明**

■ **RUN LED 显示说明**

LED 灯状态	显示说明	处理方法
灯灭	CAN2（Motion）在软件中没有配置轴	将 Motion 通讯口连接的从站在软件中配置到主站内
绿灯闪烁	CAN2（Motion）在软件中配置的轴未全部在线	1. 检查 Motion 网络中总线线缆接线正确。 2. 检查 Motion 总线线缆为台达标准 CANopen 线缆。 3. 检查 Motion 总线两端有接终端电阻。 4. 检查 Motion 通讯口和其它从站的波特率相同。 5. 检查网络配置的所有从站实际连接至网络中。 6. 检查是否有从站不能和主站建立连接。 7. 检查是否有从站掉线。
绿灯常亮	CAN2（Motion）在软件中配置的轴全部在线	无需处理

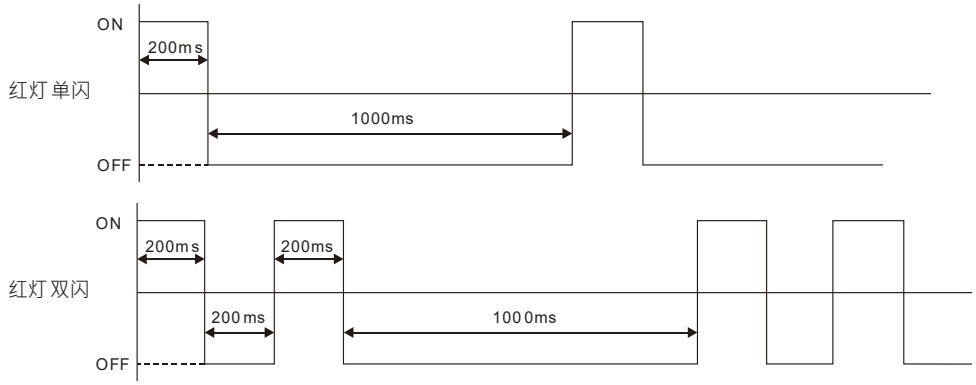
■ **RUN LED 绿灯闪烁**



■ **ERR LED 显示说明**

LED 灯状态	显示说明	处理方法
灯灭	正常	无需动作
红灯双闪	有从站掉线	1. 检查 Motion 总线线缆为台达标准 CANopen 线缆。 2. 检查 Motion 总线两端有接终端电阻。 3. 检查网络配置的从站实际连接至网络中。 4. 检查 Motion 总线线缆周围是否干扰过大。
红灯单闪	总线错误超出警戒水平	1. 检查 Motion 网络中总线线缆接线正确。 2. 检查 Motion 总线线缆为台达标准 CANopen 线缆。 3. 检查 Motion 总线两端有接终端电阻。 4. 检查 Motion 通讯口和其它从站的波特率相同。 5. 检查 Motion 总线线缆周围是否干扰过大。
红灯常亮	总线脱离（Bus-off）	

■ **ERR LED 红灯单闪和红灯双闪的区别：**



● **LAN1 LED 显示说明**

LAN1 用来显示 DVP-15MC 系列运动控制器左边第一个以太网通讯口的网络状态

LED 灯	灯状态	显示说明
黄灯	闪烁	DVP-15MC 系列运动控制器第一个以太网通讯口有发送或接收数据
	灯灭	DVP-15MC 系列运动控制器第一个以太网口没有接入以太网络

- **LAN2 LED 显示说明**

LAN2 用来显示 DVP-15MC 系列运动控制器左边第二个以太网通讯口的网络状态

LED 灯	灯状态	显示说明
黄灯	常亮	DVP-15MC 系列运动控制器第二个以太网通讯口有发送或接收数据
	灯灭	DVP-15MC 系列运动控制器第二个以太网口没有接入以太网络

- **RS-232 LED 灯显示说明**

RS-232 LED 为 DVP-15MC 系列运动控制器的 RS-232 通讯指示灯，用于显示 DVP-15MC 系列运动控制器 RS-232 通讯口的通讯状态

LED 灯状态	显示说明
黄灯闪烁	RS-232 有回复数据
灯灭	RS-232 无回复数据

- **RS485 LED 灯显示说明**

RS485 LED 为 DVP-15MC 系列运动控制器的 RS-485 通讯指示灯，用于显示 DVP-15MC 系列运动控制器 RS-485 通讯口的通讯状态

LED 灯状态	显示说明
黄灯闪烁	RS-485 有回复数据
灯灭	RS-485 无回复数据

- **输入点 LED 灯显示说明**

DVP-15MC 系列运动控制器共有 16 个输入点 LED 灯，用于显示 DVP-15MC 系列运动控制器数字量输入点处于闭合或者断开状态。

LED 灯状态	显示说明
红灯亮	输入点闭合
灯灭	输入点断开

- **输出点 LED 灯显示说明**

DVP-15MC 系列运动控制器共有 8 个输出点 LED 灯，用于显示 DVP-15MC 系列运动控制器数字量输出点处于闭合或者断开状态。

LED 灯状态	显示说明
红灯亮	输出点闭合
灯灭	输出点断开

12.2 运动指令 Error ID 含义说明

当运动指令出错时，可以通过下表 ErrorID 的值判断错误原因并排除错误。

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x1001	4097	轴号输入超出范围	修改指令中输入变量 Axis 的值在允许范围内
0x1002	4098	加速度输入超出范围	修改指令中输入变量 Acceleration 的值在正数范围内
0x1003	4099	减速度输入超出范围	修改指令中输入变量 Deceleration 的值在正数范围内
0x1004	4100	加速度的变化率输入超出范围	修改指令中输入变量 Jerk 的值在正数范围内
0x1005	4101	速度输入超出范围	修改指令中输入变量 Velocity 的值为非 0 的数值
0x1006	4102	位置输入超出范围	修改 MC_MoveAbsolute 指令中输入变量 Position 的值为不超过轴参数中模数的数值
0x1007	4103	方向输入超出范围	修改指令中输入变量 Direction 的值为该指令可设置的值
0x1008	4104	交接模式输入超出范围	修改指令中输入变量 BufferMode 的值为该指令可设置的值
0x1009	4105	参考位置的类型输入错误	修改指令中输入变量 ReferenceType 的值为该指令可设置的值
0x100A	4106	位置设置指令执行时机错误	请修改 MC_SetPosition 执行时机。请勿在 MC_Home 或者 MC_Stop 执行时执行 MC_SetPosition 指令。
0x100B	4107	电子凸轮表编号输入错误	修改指令中输入变量 CamTable 的值为软件中设置的 CamId 值
0x100C	4108	主轴轴号输入错误	修改指令中输入变量 Axis 的值在允许范围内
0x100D	4109	啮合模式输入错误	修改指令中输入变量 StartMode 的值为该指令可设置的值
0x100E	4110	主轴位置缩放比输入错误	修改指令中输入变量 MasterScaling 的值为正数
0x100F	4111	从轴位置缩放比输入错误	修改指令中输入变量 SlaveScaling 的值为非 0 的数值
0x1010	4112	主轴位置来源选择错误	修改指令中输入变量 MasterValueSource 的值为该指令可设置的值
0x1011	4113	主从轴轴号冲突	修改指令中输入变量 Master 和 Slave 的值为不同的数值
0x1012	4114	电子齿轮分子输入错误	修改指令中输入变量 Numerator 的值为非 0 的数值
0x1013	4115	电子齿轮分母输入错误	修改指令中输入变量 Denominator 的值为非 0 的数值
0x1014	4116	速度超调值输入错误	修改指令中输入变量 VelFactor 的值为该指令可设置的数值

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x1015	4117	CANopen 网络（或 Motion 网络）SDO 超时	1. 检查指令中指定的从站是否存在 2. 检查 CANopen 通讯口或 Motion 通讯口和所访问的从站连接是否正常 3. 检查 CANopen 通讯口或 Motion 通讯口和所访问的从站波特率是否一致
0x1016	4118	SDO 指令输入参数错误	检查 SDO 指令的输入参数是否合理，如访问 Index 和 SubIndex 是否存在、DataType 值是否合法。
0x1017	4119	CANopen 网络（或 Motion 网络）SDO 其它错误	检查从站是否处于正常工作状态。
0x1018	4120	位置捕获指令捕获触发位输入错误	修改指令中输入变量 TriggerInput 的值，TriggerInput 的值只能设置为 0~15，分别代表 I0~I7，I10~I15
0x1019	4121	位置捕获指令获触发位指定的输入点已经在其它位置捕获指令中使用	修改指令中输入变量 TriggerInput 的值为尚未使用的数值
0x101A	4122	捕获功能窗口功能异常	修改窗口设置上下限范围为正常范围
0x101B	4123	两个位置捕获指令同时对同一个轴进行位置捕获	避免两个位置捕获指令同时对同一个轴进行位置捕获
0x101C	4124	位置捕获指令模式设定错误	修改指令中输入变量 Mode 的值为指令中可设置的值
0x101D	4125	位置捕获指令指定的轴非编码器轴	修改指令中输入变量 Axis 的值为已配置编码器轴轴号
0x101E	4126	凸轮啮合功能激活位置范围错误	修改指令中输入变量 ActivationPosition 的值为该指令可设置的值
0x101F	4127	编码器轴无法执行此操作	修改指令轴号为实轴或虚轴轴号
0x1020	4128	使用轴未在软件 Motion 网络中配置	修改指令中输入变量 Axis 的值为 Motion 网络中已配置轴的轴号
0x1021	4129	旋切轴半径错误	修改指令中输入变量 RotaryAxisRadius 大于 0
0x1022	4130	进给轴半径错误	修改指令中输入变量 FeedAxisRadius 的值大于 0
0x1023	4131	旋切长度错误	修改指令中输入变量 CutLenth 的值大于 0
0x1024	4132	同步区开始位置错误	修改指令中输入变量 SyncStartPos 的值，范围在 0 和旋切长度之间
0x1025	4133	同步区结束位置错误	修改指令中输入变量 SyncStopPos 的值，范围在 0 和旋切长度之间
0x1026	4134	同步区同步位置设置错误	修改指令中输入变量 SyncStopPos 的值小于 SyncStartPos 的值

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x1027	4135	旋切编号错误	修改指令中输入变量 RotCutID 的值为 1-8 之间
0x1028	4136	旋切刀数错误	修改指令中输入变量 RotaryAxisKnifeNum 的值为 1~16 之间
0x1029	4137	旋切内部状态非法	修改旋切初始化参数
0x103A	4154	旋切指令初始化失败	APF_RotaryCut_Init 指令未执行，请先执行 APF_RotaryCut_Init 指令，再执行 APF_RotaryCut_In 指令
0x103B	4155	轴断线，捕获功能不能执行	轴正常连接后再执行捕获指令
0x103C	4156	电子凸轮指令主轴位置偏移量超过主轴凸轮周期范围	修改指令中输入变量 MasterOffset 的值为主轴凸轮周期范围的负值到正值之间(主轴凸轮周期范围=主轴凸轮周期最大值-主轴凸轮周期最小值)
0x103D	4157	电子凸轮指令从轴位置偏移量超过从轴凸轮周期范围	修改指令中输入变量 SlaveOffset 的值为从轴凸轮周期范围的负值到正值之间(从轴凸轮周期范围=从轴凸轮周期最大值-从轴凸轮周期最小值)
0x103E	4158	指令深度范围超出	修改指令 Depth 引脚值大小，避免过大超出范围
0x103F	4159	指令速度超调范围非法	修改指令 VelOverride 引脚值大小，避免过大超出范围
0x1040	4160	文件编码非法	修改指令 NCFile 引脚值为正常文件编码值
0x1041	4161	轴状态未在 standstill 状态时，执行 DMC_SetTorque 指令	确保轴状态在 standstill 状态时，执行 DMC_SetTorque 指令
0x1042	4162	MC_Reset 未执行成功	1. 检查 MC_Reset 指令指定的轴是否存在 2. 消除伺服报警后再执行 MC_Reset 指令
0x1043	4163	执行该指令会导致轴位置超出软件设定范围	修改指令执行最终位置使其不超出软件极限范围
0x1044	4164	MC_CamIn 指定的凸轮曲线未在软件中建立	检查MC_CamIn指令中CamTable的值能对应到软件中建立的凸轮曲线
0x1045	4165	轴组编号输入错误	检查指令的 GroupID 输入参数值是否在范围内，范围是 1~8
0x1046	4166	模式输入错误	指令的 Mode 输入参数只能为 0
0x1047	4167	From/To 指令模块编号输入错误	检查指令 Station 输入参数是否正确，编号是否正确。
0x1048	4168	From/To指令读写CR的笔数输入错误	检查指令 Num 输入参数是否正确，输入范围 1~64
0x1049	4169	指令的输入引脚没有填写变量	指令的输入引脚填写变量。
0x104A	4170	From/To 指令未收到回应	检查模块间连接是否正常，扩展模块是否工作正常。
0x104B	4171	CNC 文件为空	检查指令的 NCFile 的值，对应的 CNC 文件是否为空。

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x104C	4172	路径解析错误	确认 G 代码或者轴组指令填写路径是否正确。
0x104D	4173	位置捕获指令在窗口范围内没有接收到捕获信号，捕获失败	检查指令窗口范围设置是否合理
0x104E	4174	G 代码识别错误	确认 G 代码文件编写是否正确
0x104F	4175	G 代码预读格式错误	确认 G 代码文件编写是否正确
0x1050	4176	G 代码预读错误	确认 G 代码文件编写是否正确
0x1051	4177	G 代码格式错误	确认 G 代码文件编写是否正确
0x1052	4178	MC_Home 指令指定的轴驱动器设定 Positon 失败，驱动器可能不支持此参数	确认驱动器是否支持此参数
0x1053	4179	MC_Home 指令不支持编码器模式为数据源的编码器轴和 SSI 绝对型编码器轴	修改轴类型为其它类型的轴
0x1054	4180	G 代码中 G26 嵌套层数过多	确认 G26 嵌套层数是否超过 16 层
0x2002	8194	运动方向限制，不能执行	修改 MC_Power 指令 EnablePositive 和 EnableNegative 为 TRUE，取消轴运动方向限制
0x2004	8196	未执行 MC_MoveSuperimposed 指令，不能执行 MC_HaltSuperimposed	执行 MC_HaltSuperimposed 指令时需正在执行 MC_MoveSuperimposed 指令，修改 MC_HaltSuperimposed 指令执行时机
0x2100	8448	状态机限制不能执行此功能	修改指令执行时机，运动指令的执行请参考 10.4 节中状态机部分
0x2101	8449	BufferMode 缓存满	运动指令 BufferMode 只支持一级交接，修改当前指令执行时机。避免存在有的指令在等待执行（BufferMode 不为 0） 时，又有其它指令也开始等待执行（BufferMode 不为 0）的情况
0x2102	8450	指令不能执行 Buffer 功能	该指令无法进行 BufferMode 操作
0x3001	12289	轴类型设置错误	修改轴配置界面中轴类型
0x3002	12290	伺服报警	清除伺服报警再进行控制
0x3003	12291	伺服通讯超时	检查控制器和伺服之间连线是否正常
0x3004	12292	命令位置超过软件中设置的极限位置	检查软件中软件中设置的极限位置是否合适或者关闭软件中设置极限位置
0x3005	12293	控制器出现过从运行变为停止的过程（执行运动指令过程中）	通过 MC_Reset 指令清除错误再执行其它运动指令。
0x600D	24589	连接错误	检查通信线

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x6200	25088	TCP 远端 IP 错误	检查 TCP 远端 IP 地址格式是否正确
0x6201	25089	TCP 远端端口错误	检查 TCP 远端端口设置是否在范围内
0x6203	25091	TCP 发送装置地址错误	检查指令发送装置是否在范围内
0x6206	25094	TCP 接收装置地址错误	检查指令接收装置是否在范围内
0x6208	25096	TCP 主站实际接收数据超出设定长度	修改设定的数据接收长度
0x6209	25097	UDP 远端 IP 地址错误	检查 UDP 远端 IP 地址格式是否正确
0x620A	25098	UDP 通讯端口错误	修改 UDP 通讯端口
0x620C	25100	UDP 发送装置地址错误	检查指令发送装置是否在范围内
0x620F	25103	UDP 接收装置地址错误	检查指令接收装置是否在范围内
0x6210	25104	UDP 实际接收数据超出设定长度	修改设定的数据接收长度
0x6212	25106	以太网连接超时	修改指令超时时间或确认远端设备是否保持连接
0x6213	25107	TCP 从站接收数据超出设定长度	修改设定的数据发送长度
0x6214	25108	连接异常关闭	确认远端设备工作正常
0x6215	25109	连接失败或连接未打开	检查指令操作顺序是否正确
0x6220	25120	超时	修改指令超时时间或确认远端设备是否保持连接
0x6300	25344	连接数量超限	检查连接数量是否在允许范围内
0x8000	32768	本指令只能当本机为 CANopen 主站时才能进行诊断	更改本机为 CANopen 主站后再使用指令
0x8800	34816	任务的优先级编号大于 31	修改任务的优先级编号小于 31
0x8801	34817	该任务的看门狗功能并未开启	打开看门狗功能后再执行指令
0x8808	34824	诊断的从站未配置	配置从站后再诊断
0x8810	34832	诊断类型错误	更改诊断类型
0x8818	34840	诊断的轴未配置	配置轴后再诊断
0x8820	34848	诊断类型错误	更改诊断类型
0x9000	36864	以太网 LINK 编号超出范围（1~16）	修改以太网 LINK 编号为 1~16
0x9001	36865	以太网 LINK 功能所配置的写入长度超过允许的最大值	修改以太网 LINK 功能配置的写入长度在允许的范围之内
0x9002	36866	以太网 LINK 功能所配置的读取长度超过允许的最大值	修改以太网 LINK 功能配置的读取长度在允许的范围之内
0x9003	36867	以太网物理连接错误	检查网络硬件是否连接正常，如网线是否正确连接

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x9004	36868	Socket 编号超出范围	修改 Socket 编号为 1~4
0x9005	36869	Socket 功能所配置的发送长度超过允许的最大值	修改 Socket 功能所配置的发送长度为 0~200
0x9006	36870	Socket 功能所配置的接收长度超过允许的最大值	修改 Socket 功能所配置的接收长度为 0~200
0x9007	36871	以太网 LINK 功能配置的通讯超时时间不正确	修改超时时间，使其大于 0
0x9008	36872	Socket 功能所配置的发送接收长度均为 0	修改 Socket 功能配置的发送、接收长度任一不为 0
0x9010	36880	RS485 PLC LINK 的编号超出范围（1-24）	修改 PLC LINK 编号在 1-24 内
0x9011	36881	RS485 PLC LINK 功能所配置的写入长度超过允许的最大值	修改 RS485 PLC LINK 功能配置的写入长度在允许的范围之内
0x9012	36882	RS485 PLC LINK 功能所配置的读取长度超过允许的最大值	修改 RS485 PLC LINK 功能配置的读取长度在允许的范围之内
0x9013	36883	RS485 自由通讯协议功能所配置的发送长度超过允许的最大值	修改 RS485 自由协议功能配置的发送长度在允许的范围之内
0x9014	36884	RS485 自由通讯协议功能所配置的接收长度超过允许的最大值	修改 RS485 自由协议功能配置的接受长度在允许的范围之内
0x9015	36885	RS232 PLC LINK 的编号超出范围（1~24）	修改 PLC LINK 编号在 1-24 内
0x9016	36886	RS232 PLC LINK 功能所配置的写入长度超过允许的最大值	修改 RS232PLC LINK 功能配置的写入长度在允许的范围之内
0x9017	36887	RS232 PLC LINK 功能所配置的读取长度超过允许的最大值	修改 RS232 PLC LINK 功能配置的读取长度在允许的范围之内
0x9018	36888	RS232 自由通讯协议功能所配置的发送长度超过允许的最大值	修改 RS232 自由协议功能配置的发送长度在允许的范围之内
0x9019	36889	RS232 自由通讯协议功能所配置的接收长度超过允许的最大值	修改 RS232 自由协议功能配置的接受长度在允许的范围之内

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x901A	36890	RS485/RS232 PLC LINK 功能配置的通讯超时时间不正确	修改超时时间，使其大于 0
0x901B	36891	以太网/RS485/RS232 LINK 功能配置的读写长度均为 0	设置配置的读、写长度任一不为 0
0x901C	36892	RS485/RS232 自由通讯协议功能配置的发送和接收长度均为 0	设置发送、接收长度任一不为 0
0x9020	36896	为 WORD 装置写操作所指定的本地 Buffer 没有足够的空间满足指定的数据长度（%MW0~%MW32767）	重新设置本地 Buffer 的起始地址使其有足够空间满足指定的数据长度
0x9021	36897	为 WORD 装置写操作所指定的本地 Buffer 的起始地址不在允许的字装置区域内（%MW0~%MW32767）	重新设置本地 Buffer 的起始地址在允许的字装置区域内
0x9022	36898	为 WORD 装置写操作所指定的本地 Buffer 的起始地址在允许的字装置区域内 ,但不满足字装置地址的对齐关系（%MW0~%MW32767）	重新设置本地 Buffer 的起始地址或者修改偏移长度
0x9023	36899	为 WORD 装置读操作所指定的本地 Buffer 没有足够的空间满足指定的数据长度（%MW0~%MW32767）	重新设置本地 Buffer 的起始地址使其有足够空间满足指定的数据长度
0x9024	36900	为 WORD 装置读操作所指定的本地 Buffer 的起始地址超出了允许的字装置区域（%MW0~%MW32767）	重新设置本地 Buffer 的起始地址在允许的字装置区域内
0x9025	36901	为 WORD 装置读操作所指定的本地 Buffer 的起始地址在允许的字装置区域内 ,但不满足字装置地址的对齐关系（%MW0~%MW32767）	重新设置本地 Buffer 的起始地址或者修改偏移长度
0x9026	36902	为位装置写操作所指定的本地 Buffer 没有足够的空间满足指定的数据长度	重新设置本地 Buffer 的起始地址使其有足够空间满足指定的数据长度

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
		（%QX0.0~%QX127.7,%MX0.0~%MX65535.7）	
0x9027	36903	为位装置写操作所指定的本地 Buffer 的起始地址超出了允许的区域（%QX0.0~%QX127.7,%MX0.0~%MX65535.7）	重新设置本地 Buffer 的起始地址在允许的字装置区域内
0x9028	36904	为位装置读操作所指定的本地 Buffer 没有足够的空间满足指定的数据长度（%QX0.0~%QX127.7,%MX0.0~%MX65535.7）	重新设置本地 Buffer 的起始地址使其有足够空间满足指定的数据长度
0x9029	36905	为位装置读操作所指定的本地 Buffer 的起始地址超出了允许的区域（%QX0.0~%QX127.7,%MX0.0~%MX65535.7）	重新设置本地 Buffer 的起始地址在允许的字装置区域内
0x9030	36912	操作对象的类型指定错误	修改指令输入参数的值
0x9031	36913	为读操作所指定的功能码超出范围	重新指定允许的功能码
0x9040	36928	为发送操作所指定的本地 Buffer 没有足够的空间满足指定的数据长度（%MW0~%MW32767）	修改本地 Buffer 的起始地址或修改指定的数据长度
0x9042	36930	为发送操作所指定的本地 Buffer 的起始地址超出了允许的字装置区域（%MW0~%MW32767）	修改本地 Buffer 的起始地址
0x9043	36931	为接收操作所指定的本地 Buffer 没有足够的空间满足指定的数据长度（%MW0~%MW32767）	修改本地 Buffer 的起始地址或修改指定的数据长度
0x9045	36933	为接收操作所指定的本地 Buffer 的起始地址超出了允许的字装置区域（%MW0~%MW32767）	修改本地 Buffer 的起始地址

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0x9100	37120	Socket 指令参数超出范围	修改 Socket 指令参数到允许范围内
0x9101	37121	以太网物理连接断开	检查网线是否插好
0x9102	37122	TCP 远端 IP 地址错误	修改远端 IP 地址设置
0x9103	37123	TCP 端口错误	修改远端端口设置
0x9105	37125	TCP 发送装置地址错误	修改 TCP 发送装置的地址
0x9106	37126	TCP/UDP 接收已触发	上次接收未完成，不能再次触发
0x9107	37127	TCP 接收装置地址错误	修改 TCP 接收装置的地址
0x9108	37128	TCP 服务器模式下，接收数据长度超出设定长度	修改接收长度大于等于接收到的第一笔数据的长度（Byte 数）
0x9109	37129	UDP 接收数据长度超出设定长度	修改接收长度大于等于接收到的第一笔数据的长度（Byte 数）
0x910A	37130	UDP 远端 IP 地址错误	修改远端 IP 地址设置
0x910B	37131	UDP 端口错误	修改本地和远端端口不能同时为 0
0x910C	37132	UDP 发送装置地址错误	修改发送数据读取的装置地址
0x910D	37133	UDP 接收装置地址错误	修改接收数据存放的装置地址
0x910E	37134	TCP 连接超时	检查 Socket 配置是否正确或远端设备是否正常
0x910F	37135	TCP 客户端模式，接收数据长度超出设定长度	修改接收长度大于等于接收到的第一笔数据的长度（Byte 数）
0x9110	37136	TCP 连接被远端设备拒绝	检查远端设备是否正常或重新尝试连接
0x9111	37137	TCP/UDP 连接未开启	检查连接是否是开启状态
0x9112	37138	TCP/UDP 连接已触发	正在建立连接，不能重复触发建立连接
0x9113	37139	TCP/UDP 数据发送已触发	上次发送未完成，不能再次触发
0x9114	37140	TCP/UDP 连接已建立	连接已建立，不能重复触发建立连接
0x9115	37141	TCP/UDP 连接正在关闭	正在关闭，不能重复触发关闭
0x9116	37142	TCP/UDP 连接未关闭	只能在连接关闭的情况下才能配置 Socket 参数
0x9117	37143	TCP/UDP 发送长度参数超出限制	修改发送长度在允许范围内
0x9118	37144	TCP/UDP 接收长度参数超出限制	修改接收长度在允许范围内
0xA000	40960	凸轮点编号错误	修改凸轮表编号
0xA001	40961	挺杆点编号错误	修改挺杆点编号
0xA008	40968	M 代码编号错误	修改 M 代码编号
0xA010	40976	轴组状态机错误	更改指令执行时机
0xA011	40977	该轴组标识符超出范围	修改标识符数值
0xA012	40978	TransitionMode 输入错误	修改输入值
0xA013	40979	TransitionParameter 填写错误	修改输入值

ErrorID		ErrorID 值含义	处理方法
十六进制	十进制		
0xA014	40980	BufferMode 和 TransitionMode 不匹配	修改输入值
0xA015	40981	CircMode 填写错误	修改输入值
0xA016	40982	PathChoice 填写错误	修改输入值
0xA800	43008	输入输出齿轮比不能是 0 或者负数	修改输入输出齿轮比
0xA801	43009	机构导程不能是负数或 0	修改机构导程
0xA802	43010	轴类型错误	修改轴类型
0xA803	43011	模的值不能为非正数	修改模的值
0xA808	43016	指令还未执行完成时再次执行 Ex_Move	修改指令执行时机
0xA809	43017	执行 Ex_Stop 之前需执行 Ex_Move	修改指令执行时机
0xA80A	43018	每圈的相位需大于 0	修改 RoundPhase 的值为大于 0 的数值
0xA80B	43019	停下的相位需大于 0，小于每圈的相位	修改 StopPhase 的值为大于 0 的数值
0xA810	43024	TorqueRamp（扭矩变化率）的值为 0 或者负数	修改指令中 TorqueRamp（扭矩变化率）的值为
0xA818	43032	Lag（位置误差）的值为负数	修改 Lag（位置误差）的值为非负数
0xA819	43033	HoldTime（持续时间）的值为负数	修改 HoldTime（持续时间）的值为非负数
0xA820	43040	挺杆点个数已经达到最大值	确保挺杆点的个数不要超过规定值
0xA821	43041	写入挺杆点主轴位置（MasterPos）超过主轴范围	修改挺杆点主轴位置在软件中设定的主轴范围内
0xA830	43056	该轴组中该标识已经被使用	修改标识符
0xA831	43057	该轴已经被加入到其他的轴组的轴里	修改轴
0xA838	43064	轴组的运行状态被打断	修改指令执行时机
0xA839	43065	执行指令时，Pause 和 Stop 必须为 FALSE	修改指令执行时机
0xA83A	43066	轴组为空	修改指令执行时机
0xA83B	43067	轴组内各轴的轴状态非 StandStill	修改指令执行时机

12.3 通过系统错误码诊断系统故障

当 DVP-15MC 系列运动控制器 ERR 灯闪烁或者常亮时，可以通过系统错误变量显示的错误码的值判断错误原因并排除错误。

系统错误码		含义	处理方法
十六进制	十进制		
0x1000	4096	内部 RAM 检测失败	重新上电，若错误依然存在，请联系技术人员
0x1001	4097	内部 Flash 检测失败	
0x1002	4098	扩展接口检测失败	
0x1003	4099	内部电压检测异常（LV）	调整电源接口输入电压稳定到 24V
0x1004	4100	闪存初始化失败	重新上电，若错误依然存在，请联系技术人员
0x1005	4101	闪存 ID 检测失败	
0x1007	4103	闪存访问失败（以太网区）	重新下载程序、恢复出厂值，如仍不能恢复，请联系技术人员
0x1008	4104	闪存访问失败（扩展区）	
0x1009	4105	闪存访问失败（程序区）	
0x100A	4106	闪存访问失败（CAN Motion 区）	
0x100B	4107	闪存访问失败 （Task 区）	
0x100C	4108	闪存访问失败（CANopen ）	
0x100D	4109	闪存访问失败（硬件配置）	
0x100E	4110	闪存访问失败（CAM 区）	
0x100F	4111	闪存访问失败（闪存管理表读）	
0x1010	4112	闪存访问失败（闪存管理表修改表 1）	
0x1011	4113	闪存访问失败（闪存管理表修改表 2）	
0x1012	4114	闪存读失败	
0x1013	4115	闪存写失败	
0x1014	4116	闪存擦除失败	
0x1015	4117	CNC 文件编码超出范围	检查软件中 CNC 文件 ID 是否超出 64，再次重新下载程序，如仍不能恢复，请更新软件后再次下载
0x1016	4118	CNC 文件大小超出范围	CNC 文件过大，减小 CNC 文件大小后重新下载
0x1017	4119	增量型编码器 1 短时间位置变化过大	检查编码器输入是否过快，或者增大编码器分辨率
0x1018	4120	增量型编码器 2 短时间位置变化过大	检查编码器输入是否过快，或者增大编码器分辨率

系统错误码		含义	处理方法
十六进制	十进制		
0x1019	4121	系统堆栈耗尽	需更改程序，程序段中间变量过多
0x101A	4122	掉电保持文件过大	掉电保持变量过多，减少后重新下载程序
0x101B	4123	掉电保持文件访问失败	恢复出厂值后重新下载程序
0x1030	4144	硬件匹配错误	请联系技术人员
0x1401	5121	以太网 LAN1 初始化失败	重新上电，若无法解决，请联系技术人员
0x1402	5122	以太网 LAN1 缓存溢出	
0x1403	5123	以太网 LAN1 发送失败	
0x1404	5124	以太网发送缓存内存分配失败	
0x1601	5633	以太网 LAN2 初始化失败	重新上电，若无法解决，请联系技术人员
0x1602	5634	以太网 LAN2 缓存溢出	
0x3000	12288	指令输入输出超出限定 32	重新编排自定义 POU 的输入输出变量个数，保证输入或输出变量个数不超过 32
0x3001	12289	POU 内容超出限定大小 65535	更改 POU 的变量大小，减少变量对内存的占用
0x3002	12290	POU 个数超过限定数量 1000	减少任务调用的 POU 数量，重新下载程序
0x3003	12291	POU 类型非法	重新编译下载，重新上电，若依旧出现请更新软件
0x3004	12292	程序参数类型非法	
0x3005	12293	程序参数位偏移非法	
0x3006	12294	程序参数数据类型非法	
0x3007	12295	程序跳转范围非法	
0x3008	12296	程序内存分配对齐格式错误	
0x3009	12297	虚轴编码器内存对齐格式错误	
0x300A	12298	位访问超出(只能访问(Bit0~Bit7)	重新编译下载，重新上电，若依旧出现请更新软件
0x300B	12299	程序运行检测数据类型非法	重新编译下载，重新上电，若依旧出现请更新软件
0x300C	12300	String 数据类型长度过大	String 类型字符串过长，需修改程序，重新编译下载，若依旧出现请更新软件
0x300D	12301	变量寻址方式非法	重新编译下载，重新上电，若依旧出现请更新软件
0x3020	12320	下载数据校验非法 - CAN Motion 配置	重新编译下载，重新上电，若依旧出现请更新软件
0x3021	12321	下载数据校验非法 - 扩展配置	

系统错误码		含义	处理方法
十六进制	十进制		
0x3022	12322	下载数据校验非法 - 程序	
0x3023	12323	下载数据校验非法 - 任务	
0x3024	12324	下载数据校验非法-CANopen	
0x3025	12325	下载数据校验非法 - 硬件配置	
0x3026	12326	程序看门狗超时	程序运行超时，检查程序是否正确，是否含有无法跳出的循环
0x3027	12327	轴状态机调用失败	重新下载程序、恢复出厂值，如果错误仍然存在，请联系技术人员
0x3028	12328	CNC 列表解析错误	检查 CNC 文件是否正确，重新下载程序
0x3029	12329	CNC 文件解析错误	检查 CNC 文件是否正确，重新下载程序
0x3050	12368	优先级为 0 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x3051	12369	优先级为 1 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3052	12370	优先级为 2 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3053	12371	优先级为 3 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3054	12372	优先级为 4 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3055	12373	优先级为 5 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3056	12374	优先级为 6 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序

系统错误码		含义	处理方法
十六进制	十进制		
0x3057	12375	优先级为 7 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3058	12376	优先级为 8 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3059	12377	优先级为 9 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x305A	12378	优先级为 10 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x305B	12379	优先级为 11 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x305C	12380	优先级为 12 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x305D	12381	优先级为 13 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x305E	12382	优先级为 14 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x305F	12383	优先级为 15 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3060	12384	优先级为 16 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3061	12385	优先级为 17 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序

系统错误码		含义	处理方法
十六进制	十进制		
0x3062	12386	优先级为 18 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3063	12387	优先级为 19 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3064	12388	优先级为 20 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3065	12389	优先级为 21 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3066	12390	优先级为 22 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3067	12391	优先级为 23 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3068	12392	优先级为 24 的任务实际执行时间超过看门狗设定时间	1. 增大该任务的看门狗设定时间 2. 检查该任务调用的程序是否有死循环 3. 更改程序或者设定后重新下载程序
0x3069	12393	优先级为 25 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x306A	12394	优先级为 26 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x306B	12395	优先级为 27 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x306C	12396	优先级为 28 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x306D	12397	优先级为 29 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x306E	12398	优先级为 30 的任务实际执行时间超过设定的超时时间	请联系技术人员

系统错误码		含义	处理方法
十六进制	十进制		
0x306F	12399	优先级为 31 的任务实际执行时间超过设定的超时时间	请联系技术人员
0x5000	20480	扩展通讯校验失败	重新上电，若无法解决，请联系技术人员
0x5001	20481	扩展通讯超时	
0x5100	20736	扩展配置不符	重新上电，若无法解决，请联系技术人员
0x5200	20992	CANopen 接收缓存满	调整 CANopen 配置，检查任务设置
0x5201	20993	CANopen 发送缓存满	
0x5300	21248	CAN Motion 接收缓存满	调整 CAN Motion 配置，检查任务设置
0x5301	21249	CAN Motion 发送缓存满	

A

附录A Modbus 通讯说明

目录

A.1 ASCII 模式报文结构

- **通讯数据结构**

字段名	组成	解释说明
起始字符	STX	起始字符为“：”，冒号的 ASCII 码为 0x3A
通讯站号	ADR 1	通讯站号由两个 ASCII 码组成
	ADR 0	
功能码	CMD 1	功能码由两个 ASCII 码组成
	CMD 0	
数据	DATA （0）	数据内容由 2n 个 ASCII 码组成，n≤205
	DATA （1）	
	…….	
	DATA （n-1）	
LRC 校验码	LRC CHK 1	LRC 校验码由 2 个 ASCII 码组成
	LRC CHK 0	
结束字符	END1	结束字符由 2 个 ASCII 码组成 END1 = CR （0x0D）， END0 = LF （0x0A）
	END0	

16 进制位与 ASCII 码对应关系如下表所示：

16 进制位元	″0″	″1″	″2″	″3″	″4″	″5″	″6″	″7″
ASCII 码	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
16 进制位元	″8″	″9″	″A″	″B″	″C″	″D″	″E″	″F″
ASCII 码	0x38	0x39	0x41	0x42	0x43	0x44	0x45	0x46

- **ADR（通讯站号）**

有效的通讯站号范围为 0～254。当通讯站号为 0 时表示对所有 Modbus 从站广播，收到广播消息的从站不会对广播消息做回应。当通讯站号不为 0 时，从站会回应正常信息给主站设备。例如，通讯站号为 16 的 ASCII 码表示方法如下所示（十进制数 16 的十六进制为 10）:（ADR 1，ADR 0）=’10’，‘1’=31H, ‘0’ = 30H。

- **功能码及数据**

数据的格式取决于功能码。例如，读取 DVP-15MC 系列运动控制器以 0x0000（十六进制）为起始地址的连续 2 个地址的数据，DVP-15MC 系列运动控制器的通讯站号为 1， 0x0000 为 DVP-15MC 系列运动控制器 PLC 模块内部%MW0 的 modbus 地址。数据含义解释如下：

PC→DVP-15MC 系列运动控制器：3A 30 31 30 33 30 30 30 30 30 30 30 32 46 41 0D 0A

DVP-15MC 系列运动控制器→PC：3A 30 31 30 33 30 34 30 30 30 31 30 30 30 32 46 35 0D 0A

■ **请求信息：**

字段名	字段字符	字段字符对应的 ASCII 码
起始字符	" : "	3A
通讯站号：01	"0"	30
	"1"	31
功能码：03	"0"	30
	"3"	33
起始数据地址：0x0000	"0"	30
	"0"	30
	"0"	30
	"0"	30
数据个数 (以字为单位) : 2	"0"	30
	"0"	30
	"0"	30
	"2"	32
LRC 校验码：0xFA	"F"	46
	"A"	41
结束字符 1	CR	0D
结束字符 0	LF	0A

■ **回应信息：**

字段名	字段字符	字段字符对应的 ASCII 码
起始字符	" : "	3A
通讯站号：01	"0"	30
	"1"	31
功能码：03	"0"	30
	"3"	33
读取数据个数 (以字节为单位)	"0"	30
	"4"	34
读取 0x1000 地址的内容值	"0"	30
	"0"	30
	"0"	30
	"1"	31
读取 0x1001 地址的内容值	"0"	30
	"0"	30

字段名	字段字符	字段字符对应的 ASCII 码
	“0”	30
	“2”	32
LRC 校验码: 0xF5	“F”	46
	“5”	35
结束字符 1	CR	0D
结束字符 0	LF	0A

- **LRC 校验（校验和）**

LRC 校验码为从通讯站号至最后一个数据内容的 16 进制数叠加后的值各位取反后再加 1 的值。如下例所示，LRC 校验码的值为 0xFA。LRC 校验码的计算方法如下 :0x01+ 0x03 + 0x00 + 0x00 + 0x00 + 0x02 = 0x06，0x06 各位取反后再加 1 的结果为 0xFA。

字段名	字段字符	字段字符对应的 ASCII 码
起始字符	“：”	3A
通讯站号：01	“0”	30
	“1”	31
功能码：03	“0”	30
	“3”	33
起始数据地址：0x0000	“0”	30
	“0”	30
	“0”	30
	“0”	30
数据个数（以字为单位）: 2	“0”	30
	“0”	30
	“0”	30
	“2”	32
LRC 校验码：0xFA	“F”	46
	“A”	41
结束字符 1：CR	CR	0D
结束字符 0：LF	LF	0A

A.2 RTU 模式报文结构

- **RTU 模式**

■ 通讯数据结构

开始	保持无输入数据≥10 ms
通讯站号	从站地址: : 8 位二进制数地址
功能码	功能码: : 8 位二进制数地址
数据 （n-1）	数据内容 n × 8 位二进制数, n<=202
…….	
数据 0	
CRC 校验和低字节	CRC 校验和 CRC 校验和由两个 8 位二进制数组成
CRC 校验和高字节	
结束	保持无输入数据≥10 ms

■ 通讯站号

有效的通讯站号范围为 0～254。当通讯站号为 0 时表示对所有从站进行广播，收到广播消息的从站不会对广播消息做回应。当从站站号不为 0 时，从站会回应正常消息给主站设备。例如，当对通讯站号为 16 的从站进行通讯时，从站站号设为 0x10，十进制数 16 的十六进制为 10。

■ 功能码及数据

数据的格式取决于功能码。

例如 :读取 DVP-15MC 系列运动控制器以 0x0000 为起始地址的连续 2 个地址的数据 ,DVP-15MC 系列运动控制器的通讯站号为 1，0x0000 为 DVP-15MC 系列运动控制器 PLC 模块内部%MW0 的 modbus 地址。

通讯线上的数据及数据含义解释如下：

PC→DVP-15MC 系列运动控制器：01 03 00 00 00 02 C4 0B

DVP-15MC 系列运动控制器→PC：01 03 04 00 01 02 00 2A 32

➢ **请求信息：**

字段名	字段
开始	保持无输入数据≥10 ms
通讯站号	01
功能码	03
modbus 地址高字节	00
modbus 地址低字节	00
读取数据个数高字节	00
读取数据个数低字节	02
CRC 校验和低字节	C4
CRC 校验和高字节	0B

字段名	字段
结束	保持无输入数据≥10 ms

➢ **回应信息：**

字段名	字段
开始	保持无输入数据≥10 ms
通讯站号	01
功能码	03
读取数据个数（以字节为单位）	04
读取数据内容高字节	00
读取数据内容低字节	01
读取数据内容高字节	00
读取数据内容低字节	02
CRC 校验和低字节	2A
CRC 校验和高字节	32
结束	保持无输入数据≥10 ms

■ **CRC 校验 （校验和）**

CRC 校验从“通讯站号”开始，至“最后一个数据内容”结束。CRC 校验计算方法如下：

步骤 1：载入一个内容值为 FFFF（十六进制）的 16 位寄存器（称为 CRC 寄存器）。

步骤 2：指令信息中的第一个字节的 8 位数据与 CRC 寄存器低字节的 8 位数据进行异或运算，运算结果存储于 CRC 寄存器内。

步骤 3: CRC 寄存器的内容值右移 1 位并将其最高位填入 0。

步骤 4：检查 CRC 寄存器最低位的值，如果为 0 则重复步骤 3；如果为 1，CRC 寄存器的内容与 A001（十六进制）进行异或运算，运算结果存储于 CRC 寄存器内。

步骤 5：重复步骤 3 及步骤 4，直到 CRC 寄存器的内容被右移了 8 位。此时，指令信息的第一个字节已完成处理。

步骤 6：对指令信息的下一个字节重复步骤 2 至步骤 5 的操作，直到指令信息的所有字节都被处理完成。CRC 寄存器最后的内容就是 CRC 校验值。在指令信息中传送 CRC 校验值时，计算出的 CRC 校验值高低字节须互换，即 CRC 校验值低字节先被传送。

下面为用 C 语言求 CRC 校验值的计算范例

```
unsigned char* data     ← // 指令信息内容指针
unsigned char length    ← // 指令信息的长度
unsigned int crc_chk（unsigned char* data, unsigned char length）
{
int j;
unsigned int reg_crc=0xffff;
while（length-- ）
{
reg_crc ^= *data++;
```

```
for （j=0;j<8;j++）
{
  If （reg_crc & 0x01） reg_crc=（reg_crc>>1） ^ 0xa001; /* LSB（b0）=1 */
else reg_crc=reg_crc >>1;
}
}
return reg_crc; // the value that sent back to the CRC register finally
}
```

A.3 支持的 Modbus 功能码

- **DVP-15MC 系列运动控制器 COM2 通讯口为运动控制模块所有时支持的功能码如下表所示：**

功能码	说明	可操作装置
0x01	读输出位装置寄存器（一次最多可以读 256 个 位（bit）的数据）的值。	%QX
0x02	读位装置寄存器（一次最多可以读 256 个 位（bit）的数据）的值。	%IX,%QX
0x03	读单个或多个字装置寄存器（一次最多可读 100 个字（Word）的数据）的值。	%MW,%QW,%IW
0x05	写单个位装置位寄存器的值。	%QX
0x06	写单个字装置寄存器的值。	%MW,%QW
0x0F	写多个位装置寄存器（一次最多可以写 256 个位（bit）的数据）的值。	%QX
0x10	写多个字装置寄存器的值(一次最多可写 100 个字(Word)的数据)。	%MW,%QW

A.4 Modbus 异常回应码

- **DVP-15MC 系列运动控制器支持的异常回应码如下表所示：**

异常回应码	含义
0x01	非法命令码：PLC 接收的命令信息中的命令码无效
0x02	非法的装置地址：接收的命令信息中的地址无效
0x03	非法装置值：PLC 接收的命令信息中的数据内容无效
0x07	◆ 校验和错误 ✓ 检查校验和是否正确 ◆ 非法的命令信息 ✓ 命令信息太短 ✓ 命令信息长度超出范围

A.5 Modbus 功能码介绍

- **功能码：03, 读取单个或多个字装置寄存器的值**

■ 请求信息数据的数据结构：

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	读取字装置首地址	高字节
Byte3		低字节
Byte4	读取字装置地址个数（以 Word 为单位）	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ 回应信息数据结构：

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	读取字装置地址个数（以 Byte 为单位）	单字节
Byte3	字装置地址内容	高字节
Byte4		低字节
…	字装置地址内容	高字节
…		低字节
Byte n	字装置地址内容	高字节
Byte n+1		低字节
Byte n+2	CRC 校验和低字节	低字节
Byte n+3	CRC 校验和高字节	高字节

■ 异常回应信息数据结构：

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	0x80+功能码	单字节
Byte2	异常回应码	单字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

■ **范例**

通过 03 功能码读取 DVP-15MC 系列运动控制器 0x0000，0x0001 地址的内容值，0x0000、0x0001 为 DVP-15MC 系列运动控制器内部%MW0、%MW1 的 modbus 地址，假设%MW0 的值为 0x0001、%MW1 的值为 0x0002。

请求信息：01 03 00 00 00 02 C4 0B

回应信息：01 03 04　00 01 00 02 2A 32

- **功能码：06, 写单个字装置寄存器的值**

■ **请求信息数据的数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	写入 DVP-15MC 系列运动控制器装置地址	高字节
Byte3		低字节
Byte4	写入字装置数值	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	写入字装置地址	高字节
Byte3		低字节
Byte4	写入字装置数值	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	0x80+功能码	单字节
Byte2	异常回应码	单字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

■ **范例：**

通过 06 功能码将数值 0x0100 写入 DVP-15MC 系列运动控制器 0x0000 地址内。

请求信息：01 06 00 00 01 00 88 5A

回应信息：01 06 00 00 01 00 88 5A

- **功能码：0x10, 写多个字装置寄存器的值**

■ **请求信息数据的数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	写入数值的字装置首地址	高字节
Byte3		低字节
Byte4	写入字装置地址个数（以 Word 为单位）	高字节
Byte5		低字节
Byte6	写入字装置地址个数（以 Byte 为单位）	单字节
Byte7	写入字装置地址的数值	高字节
Byte8		低字节
…	写入字装置地址的数值	高字节
…		低字节
Byte n	写入字装置地址的数值	高字节
Byte n+1		低字节
Byte n+2	CRC 校验和低字节	低字节
Byte n+3	CRC 校验和高字节	高字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	写入字装置首地址	高字节
Byte3		低字节
Byte4	写入字装置地址个数（以 Word 为单位）	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节

数据顺序	名称	字节说明
Byte1	0x80+功能码	单字节
Byte2	异常回应码	单字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

■ **范例：**

通过 0x10 功能码将 0x0100、0x0200 写入 DVP-15MC 系列运动控制器 0x0000、0x0001 地址内，0x0000，0x0001 为 DVP-15MC 系列运动控制器内部%MW0、%MW1 的 modbus 地址。

请求信息：01 10 00 00 00 02 04 01 00 02 00 F3 33

回应信息：01 10 00 00 00 02 41 C8

- **功能码：0x01，读输出位装置寄存器的值。**

■ **请求信息数据的数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	读取位装置起始地址	高字节
Byte3		低字节
Byte4	读取位装置的个数	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	读取位装置的 Byte 数目。	单字节
Byte3	读取到位装置的状态值	单字节
…	读取到位装置的状态值	单字节
Byte n	读取到位装置的状态值	单字节
Byte n+1	CRC 校验和低字节	低字节
Byte n+2	CRC 校验和高字节	高字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	0x80+功能码	单字节

数据顺序	名称	字节说明
Byte2	异常回应码	单字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

备注：回应信息中 Byte2 的数值由请求信息中 Byte4 和 Byte5 的值来决定。假设请求信息中读取位装置的个数为 A，A 除以 8 的商为 B，若整除则回应信息中读取位装置的 Byte 数目为 B；否则 Byte 数目为 B + 1，详细请看下面的范例。

■ **范例：**

通过 01 功能码读取 DVP-15MC 系列运动控制器 %QX2.0~%QX3.4 的状态值，%QX2.0 的地址为 0xA010，假设%QX2.0~%QX2.7=1000 0001, %QX3.0~%QX3.4=1 0001

请求信息：01 01 A0 10 00 0D DE 0A

回应信息：01 01 02 81 11 19 A0

- **功能码：0x02, 读位装置寄存器的值。**

■ **请求信息数据的数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	读取位装置起始地址	高字节
Byte3		低字节
Byte4	读取位装置的个数	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	读取位装置的 Byte 数目。	单字节
Byte3	读取到的位装置的状态值	单字节
…	读取到的位装置的状态值	单字节
Byte n	读取到的位装置的状态值	单字节
Byte n+1	CRC 校验和低字节	低字节
Byte n+2	CRC 校验和高字节	高字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	0x80+功能码	单字节
Byte2	异常回应码	单字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

备注：回应信息中 Byte2 的数值由请求信息中 Byte4 和 Byte5 的值来决定。假设请求信息中读取位装置的个数为 A，A 除以 8 的商为 B，若整除则回应信息中读取位装置的 Byte 数目为 B；否则 Byte 数目为 B + 1，详细请看下面的范例。

■ **范例：**

通过 02 功能码读取 DVP-15MC 系列运动控制器 %QX2.0~%QX3.4 的状态值，%QX2.0 的地址为 0xA010，假设%QX2.0~%QX2.7=1000 0001, %QX3.0~%QX3.4=1 0001。

请求信息：01 02 A0 10 00 0D 9A 0A

回应信息：01 02 02 81 11 19 E4

- **功能码：0x05, 设置单个位装置寄存器的值**

■ **请求信息数据数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	位装置的 Modbus 地址	高字节
Byte3		低字节
Byte4	位装置的写入值	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	位装置的 Modbus 地址	高字节
Byte3		低字节
Byte4	位装置的写入值	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节

数据顺序	名称	字节说明
Byte7	CRC 校验和高字节	高字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	0x80+功能码	单字节
Byte2	异常回应码	单字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

备注：写入值为 0x0000 表示将 FALSE 写入位装置，0xFF00 表示将 TRUE 写入位装置。

■ **范例：**

通过 05 功能码设置 DVP-15MC 系列运动控制器 %QX0.0 的值为 TRUE，%QX0.0 的地址为 0xA000。

请求信息：01 05 A0 00 FF 00 AE 3A

回应信息：01 05 A0 00 FF 00 AE 3A

- **功能码：0x0F, 写多个位装置寄存器的值。**

■ **请求信息数据数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节
Byte2	写入位装置首地址	高字节
Byte3		低字节
Byte4	写入位装置的个数	高字节
Byte5		低字节
Byte6	写入位装置的 Bytes 数目	单字节
Byte7	写入位装置的值	单字节
…	写入位装置的值	单字节
Byte n	写入位装置的值	单字节
Byte n+1	CRC 校验和低字节	低字节
Byte n+2	CRC 校验和高字节	高字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	功能码	单字节

数据顺序	名称	字节说明
Byte2	写入位装置首地址	高字节
Byte3		低字节
Byte4	写入位装置的个数	高字节
Byte5		低字节
Byte6	CRC 校验和低字节	低字节
Byte7	CRC 校验和高字节	高字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	Modbus 站号	单字节
Byte1	0x80+功能码	单字节
Byte2	异常回应码	高字节
Byte3	CRC 校验和低字节	低字节
Byte4	CRC 校验和高字节	高字节

备注：请求信息中有多少个 Byte 的数据由请求信息中欲写入位数值的个数决定。

■ **范例：**

通过 0F 功能码设置 DVP-15MC 系列运动控制器 %QX0.0~%QX0.7=1000 0001，%QX0.0 的地址为 0xA000。

请求信息：01 0F A0 00 00 08 01 81 26 55

回应信息：01 0F A0 00 00 08 76 0D

A.6 装置对应 Modbus 地址列表

- **DVP-15MC 系列运动控制器装置编号及对应的装置地址：**

装置名称	装置编号	装置说明	装置地址（hex）	装置属性
I	%IX0.0~%IX127.7	位装置寄存器	6000 ~ 63FF	只读
Q	%QX0.0~%QX127.7		A000 ~ A3FF	读/写
I	%IW0~%IW63	字装置寄存器	8000 ~ 803F	只读
Q	%QW0~%QW63		A000 ~ A03F	读/写
M	%MW0~%MW32767		0000 ~ 7FFF	读/写

MEMO

B

附录B ModbusTCP 通讯说明

目录

B.1 ModbusTCP 报文结构

- **Modbus TCP 报文结构：**

数据顺序	名称		说明
Byte0	事务标识符	高字节	事务标识符为 0
Byte1		低字节	
Byte2	协议标识符	高字节	协议标识符为 0
Byte3		低字节	
Byte4	Modbus 数据长度	高字节	Modbus 站号后的字节数目和（包括 Modbus 站号）。
Byte5		低字节	
Byte6	Modbus 站号	单字节	0 ~ 0xFF
Byte7	功能码	单字节	
Byte8	装置地址	高字节	0~0xFFFF
Byte9		低字节	
Byte10	Modbus 数据	高字节	Modbus 数据字节数由功能码决定.

B.2 ModbusTCP 中支持的 Modbus 功能码

- **DVP-15MC 系列运动控制器支持的 Modbus 功能码：**

功能码	说明	适用元件种类
0x02	读位装置寄存器（一次最多可以读 256 个 位（bit）的数据）的值。	%IX、%QX
0x03	读单个或多个字装置寄存器（一次最多可读 100 个字（Word）的数据）的值。	%IW、%QW、%MW
0x05	写单个位装置位寄存器的值。	% QX
0x06	写单个字装置寄存器的值。	%QW、%MW
0x0F	写多个位装置寄存器（一次最多可以写 256 个位（bit）的数据）的值。	% QX
0x10	写多个字装置寄存器的值(一次最多可写 100 个字(Word)的数据)。	%QW、%MW

B.3 ModbusTCP 异常回应码

- **DVP-15MC 系列运动控制器支持的 Modbus 异常回应码如下表所示：**

异常回应码	含义
0x01	不支持的功能码
0x02	不支持的 Modbus 地址
0x03	数据长度超出范围

B.4 ModbusTCP 中 Modbus 功能码介绍

- **功能码：03，读单个或多个字装置寄存器的值。**

■ 请求信息数据数据结构：

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	低字节
Byte7	功能码	单字节
Byte8	读取字装置首地址	高字节
Byte9		低字节
Byte10	读取字装置地址个数（以 Word 为单位）	高字节
Byte11		低字节

■ 回应信息数据结构：

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	读取字装置地址的数目（以 Byte 为单位）	单字节
Byte9	字装置地址内容	高字节
Byte10		低字节
…	字装置地址内容	高字节
Byte n		低字节

■ 异常回应信息数据结构：

数据顺序	名称	字节说明
Byte0	事务标识符	高字节

数据顺序	名称	字节说明
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	0x80 + 功能码	单字节
Byte8	异常回应码	单字节

■ **范例：**

通过 03 功能码读取 DVP-15MC 系列运动控制器 0x0000，0x0001 地址的内容值，0x0000、0x0001 为 DVP-15MC 系列运动控制器内部%MW0、%MW1 的 Modbus 地址，假设%MW0 的值为 0x0100，%MW1 的值为 0x0200。

请求信息：00 00 00 00 00 06 01 03 00 00 00 02

回应信息：00 00 00 00 00 07 01 03 04 01 00 02 00

- **功能码：06，写单个字装置寄存器的值。**

■ **请求信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	写入数值的字装置地址	高字节
Byte9		低字节
Byte10	写入字装置地址的数值	高字节
Byte11		低字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节

数据顺序	名称	字节说明
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	写入数值的字装置地址	高字节
Byte9		低字节
Byte10	写入字装置地址的数值	高字节
Byte11		低字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	0x80+功能码	单字节
Byte8	异常回应码	单字节

■ **范例：**

通过 06 功能码将数值 0x0100 写入 DVP-15MC 系列运动控制器 0x0000 地址内。

请求信息：00 00 00 00 00 06 01 06 00 00 01 00

回应信息：00 00 00 00 00 06 01 06 00 00 01 00

- **功能码：0x10，写多个字装置寄存器的值**

■ **请求信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节

数据顺序	名称	字节说明
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	写入数值的字装置首地址	高字节
Byte9		低字节
Byte10	写入数值的字装置地址个数（以 Word 为单位）	高字节
Byte11		低字节
Byte12	写入数值的字装置地址个数（以 Byte 为单位）	单字节
Byte13	写入字装置地址的数值	高字节
Byte14		低字节
…	写入字装置地址的数值	高字节
Byte n		低字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	写入数值的字装置首地址	高字节
Byte9		低字节
Byte10	写入数值的字装置地址个数（以 Word 为单位）	高字节
Byte11		低字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节

数据顺序	名称	字节说明
Byte6	Modbus 站号	单字节
Byte7	0x80+功能码	单字节
Byte8	异常回应码	单字节

备注 :回应信息中有多少个 Byte 由请求信息中读取 DVP-15MC 系列运动控制器装置地址个数决定，所以回应信息中 Byte n 中 n 的数值可以读取 DVP-15MC 系列运动控制器装置地址个数算出，详细可看下面的范例。

■ **范例：**

通过 06 功能码将 0x0100，0x0200 写入 DVP-15MC 系列运动控制器 0x0000，0x0001 地址内，0x0000， 0x0001 为 DVP-15MC 系列运动控制器内部%MW0，%MW1 的 modbus 地址。

请求信息：00 00 00 00 00 0B 01 10 00 00 00 02 04 01 00 02 00

回应信息：00 00 00 00 00 06 01 10 00 00 00 02

- **功能码：0x02，读位装置寄存器的值。**

■ **请求信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	读取状态的位装置首地址	高字节
Byte9		低字节
Byte10	读取位装置的个数	高字节
Byte11		低字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节

数据顺序	名称	字节说明
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	读取位装置的 Byte 数目。	单字节
Byte9	读取到的位装置的状态值	单字节
…	读取到的位装置的状态值	单字节
Byte n	读取到的位装置的状态值	单字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	0x80+功能码	单字节
Byte8	异常回应码	单字节

■ **范例：**

通过 02 功能码读取 DVP-15MC 系列运动控制器 %QX2.0~%QX3.4 的状态值，%QX2.0 的地址为 0xA010，假设%QX2.0~%QX2.7=1000 0001， %QX3.0~%QX3.4=10001。

请求信息：00 00 00 00 00 06 01 02 A0 10 00 0D

回应信息：00 00 00 00 00 06 01 02 02 81 11

- **功能码：0x05, 写单个位装置寄存器的值。**

■ **请求信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节

数据顺序	名称	字节说明
Byte8	位装置的 Modbus 地址	高字节
Byte9		低字节
Byte10	位装置的写入值	高字节
Byte11		低字节

■ **回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	位装置的 Modbus 地址	高字节
Byte9		低字节
Byte10	位装置的写入值	高字节
Byte11		低字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	0x80+功能码	单字节
Byte8	异常回应码	单字节

备注：写入值为 0x0000 表示将 0 写入位装置，0xFF00 表示将 1 写入位装置。

■ **范例：**

通过 05 功能码设置 DVP-15MC 系列运动控制器 %QX0.0 的值为 1，%QX0.0 的地址为 0xA000。

请求信息：00 00 00 00 00 06 01 05 A0 00 FF 00

回应信息：00 00 00 00 00 06 01 05 A0 00 FF 00

- **功能码：0x0F, 写多个位装置寄存器的值。**

■ 请求信息数据结构：

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	功能码	单字节
Byte8	写入数值的位装置首地址	高字节
Byte9		低字节
Byte10	写入数值的位装置的个数	高字节
Byte11		低字节
Byte12	写入数值的位装置的 Bytes 数目	单字节
Byte13	写入位装置的值	单字节
Byte n	写入位装置的值	单字节

■ 回应信息数据结构：

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	单字节
Byte5	Modbus 站号	高字节
Byte6		低字节
Byte7	功能码	单字节
Byte8	读取状态的位装置首地址	高字节
Byte9		低字节
Byte10	写入数值的位装置的个数	高字节
Byte 11		低字节

■ **异常回应信息数据结构：**

数据顺序	名称	字节说明
Byte0	事务标识符	高字节
Byte1		低字节
Byte2	协议标识符	高字节
Byte3		低字节
Byte4	Modbus 数据长度	高字节
Byte5		低字节
Byte6	Modbus 站号	单字节
Byte7	0x80+功能码	单字节
Byte8	异常回应码	单字节

■ **范例：**

通过 0F 功能码设置 DVP-15MC 系列运动控制器 %QX0.0~%QX0.7=1000 0001，%QX0.0 的地址为 0xA000。

请求信息：00 00 00 00 00 0A 01 0F A0 00 00 08 01 81

回应信息：00 00 00 00 00 06 01 0F A0 00 00 08

B.5 装置对应 Modbus 地址列表

装置名称	装置编号	装置说明	装置地址（hex）	装置属性
I	%IX0.0~%IX127.7	位装置寄存器	6000 ~ 63FF	只读
Q	%QX0.0~%QX127.7		A000 ~ A3FF	读/写
I	%IW0~%IW63	字装置寄存器	8000 ~ 803F	只读
Q	%QW0~%QW63		A000 ~ A03F	读/写
M	%MW0~%MW32767		0000 ~ 7FFF	读/写

MEMO

C

附录C CANopen 通讯说明

目录

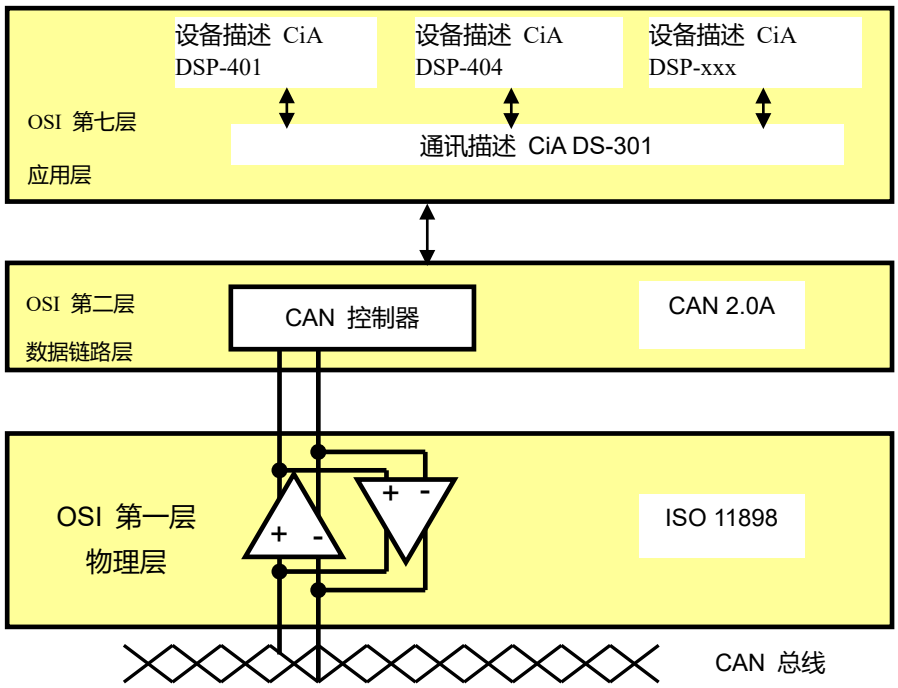
- **关于 CANopen 协议**

CAN（Controller Area Network）现场总线仅仅定义了物理层、数据链路层（见 ISO11898 标准），没有规定应用层；实际设计中，物理层、数据链路层完全由硬件实现。所以 CAN 现场总线本身并不完整，需要一个高层协议来定义 CAN 报文中的 11/29 位标识符、8 字节数据的使用。

CANopen 协议是一种基于 CAN 的高层协议，它是由 CiA（CAN-in-Automation）定义并维护的协议之一,它是在 CAL（CAN Application Layer）协议基础上开发的,使用了 CAL 通信和服务协议子集。

CANopen 协议涵盖了应用层和通讯描述（CiA DS301），另外还包括可编程设备的构架（CiA 302），电缆和连接器的介绍（CiA 303-1）以及单位和称谓表示法（CiA 303-2）。

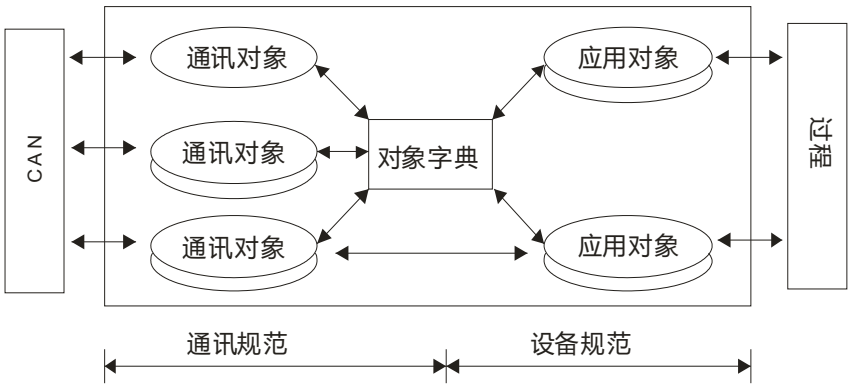
在 OSI 模型中，CAN 标准、CANopen 协议之间的关系如下图所示：



- **对象字典**

CANopen 使用基于对象的方法来定义标准设备，每个设备都表现为一组对象的集合，能够被网络所访问。CANopen 设备模型如下图所示，从下图可以看出对象字典是通讯程序和上层应用程序之间的接口。

CANopen 的核心概念是设备对象字典（Object Dictionary，OD），它是一个有序的对象组，每个对象采用一个 16 位的索引值来寻址，为了允许访问数据结构中的单个元素，同时定义了一个 8 位的子索引。CANopen 网络中每个节点都有一个对象字典。对象字典包含了描述这个设备和它的网络行为的所有参数。一个节点的对象字典是在电子数据文档（Electronic Data Sheet，EDS）中描述。



- **CANopen 通讯对象**

CANopen 通讯协议包括 PDO、SDO、NMT 及其它预定义 CANopen 通讯对象，

PDO 详细说明请参阅本手册附录 C.3 节【PDO 介绍】。

SDO 详细说明请参阅本手册附录 C.4 节【SDO 介绍】。

NMT 详细说明请参阅本手册附录 C.2 节【网络管理（NMT）】。

- **其它预定义 CANopen 通讯对象（SYNC,EMCY）**

■ **同步对象（Sync Object）**

同步对象由网络中主站节点以广播的形式周期发送到 CAN 总线的报文。这个对象用来实现基本的网络时钟信号，每个设备可以根据自己的配置，决定是否使用该事件和其它网络设备进行同步通讯。如在控制驱动装置时，各个装置收到主站发送的动作命令后并不立即动作，而是等收到同步报文后一起动作，如此可以实现多个装置同步动作。

SYNC 报文格式如下图所示：

COB-ID
80（hex）

■ **紧急事件对象（Emergency Object）**

紧急事件对象是由 CANopen 设备用来标识内部紧急错误的，当设备出现紧急错误时，设备发出紧急事件报文（报文中包含紧急错误码），设备进入错误状态。当错误消除后，设备发出紧急事件报文报错消除，紧急错误代码为 0，设备进入正常状态。

Emergency 报文格式如下图所示：

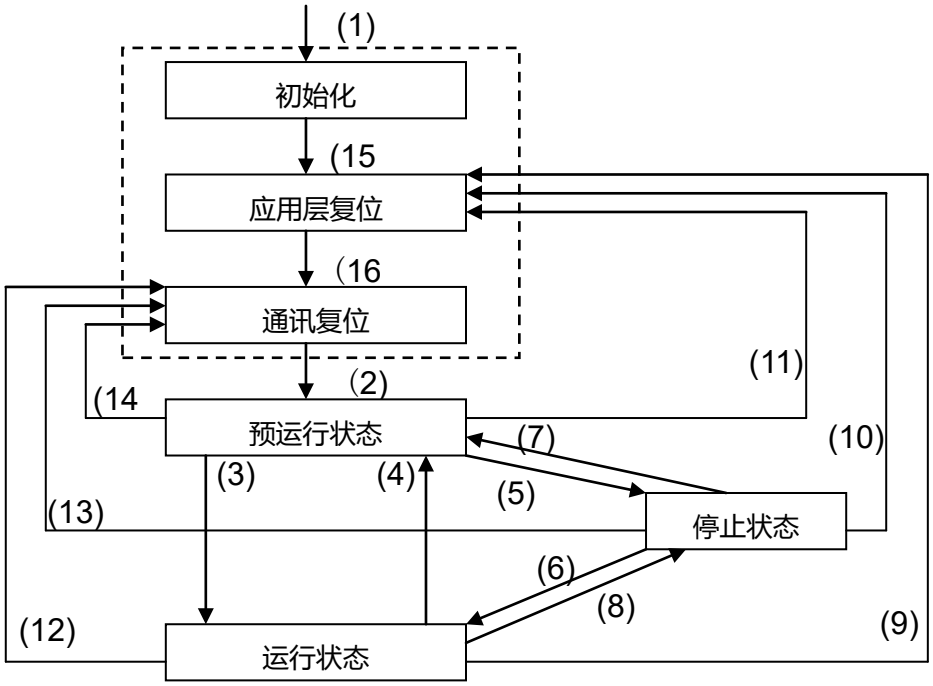
COB-ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
80（hex）+Node-ID	紧急错误码		错误暂存器	厂商自定义错误码				
	LSB	MSB						

备注：错误暂存器内的值映像到对象字典（Object Dictionary）中的索引地址为 1001（hex）。若该值等于 0，则表示无错误发生；若该值等于 1，则表示发生了一般性错误；若该值等于 H'80，则表示发生了设备内部错误。

C.1 节点状态说明

- **Module control services（节点状态控制服务）**

节点状态控制是指 CANopen 网络中主站节点通过发送命令控制从站的状态，从站收到主站的命令后执行，不需要回复。所有的 CANopen 节点都有一个内部的 NMT 状态，从站节点共有 4 种状态：初始化状态、预运行状态、运行状态、停止状态。设备的状态图如下图所示：



（1）上电后，自动进入初始化状态　（2）初始化完成后，自动进入预运行状态

（3）（6）启动远程节点　（4）（7）进入预运行状态

（5）（8）停止远程节点　（9）（10）（11）应用层复位

（12）（13）（14）通讯复位　（15）自动进入应用层复位状态

（16）自动进入通讯复位状态

通讯对象与状态的关系如下表所示，通讯对象服务只有在适当的状态下才可以执，如 SDO 只能在运行和预运行状态下执行。

	初始化	预运行	运行	停止
PDO（过程数据）			X	
SDO（服务数据）		X	X	
SYNC（同步对象）		X	X	
Time Stamp （时间戳）		X	X	
EMCY（紧急事件）		X	X	
Boot-up（启动引导）	X			
NMT（网络管理）		X	X	X

节点状态控制报文格式如下表所示：

COB-ID	Byte 0	Byte 1
0	命令说明符（CS）	从站站号（0 表示广播）

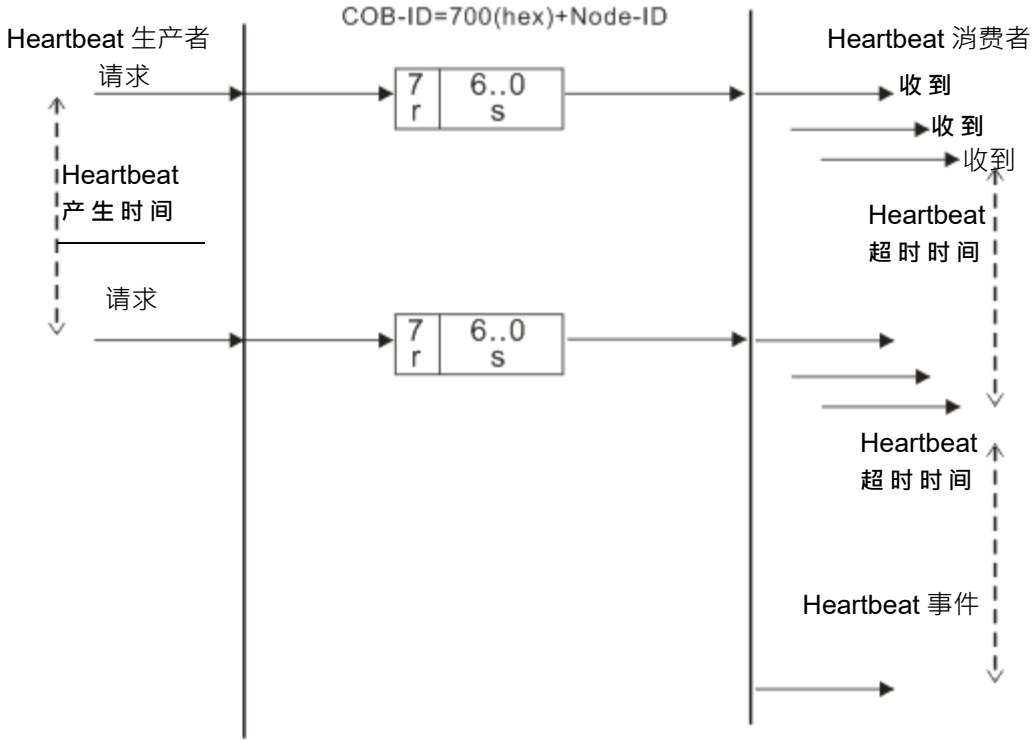
各命令说明符的功能见下表：

命令说明符（hex）	功能
01	启动远程节点
02	停止远程节点
80	进入预运行状态
81	应用层复位
82	通讯复位

- **Error Control services（错误控制服务）**

错误控制服务用于监测 CANopen 网络中是否有节点掉线。错误控制服务分为两种 :heartbeat 和 node guarding，本机只支持 heartbeat。如从站启动 heartbeat 服务后，主站才可以监视从站是否掉线。

Heartbeat 原理如下图所示：Heartbeat 生产者按照设定的 Heartbeat 产生时间定时发送 Heartbeat 报文，一个或者多个 Heartbeat 消费者监视 Heartbeat 生产者发送的报文，当消费者在设定的超时时间内没有收到生产者发送的报文时，产生 Heartbeat 事件表明 CANopen 通讯异常。



- **Boot-up services（启动引导服务）**

从站在初始化完成进入预运行状态后，会发送一笔 Boot-up 报文，表示初始化完成。

C.2 网络管理（NMT）

CANopen 的网络管理遵循“主/从”模式。一个 CANopen 网络里只能存在一个 NMT 主站，其它节点均被当成从站。NMT 可实现 3 种服务：Module control services（节点状态控制服务）、Error Control services（错误控制服务）和 Boot-up services（启动引导服务）。详细说明请参阅本手册第 A.15 节【节点状态说明】。

C.3 PDO 介绍

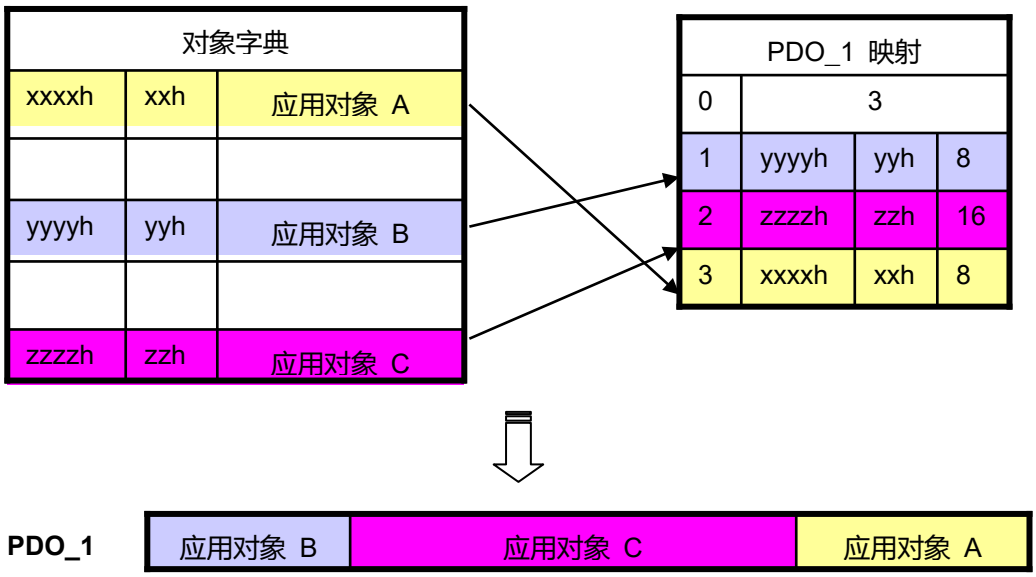
- **PDO （Process Data Object:过程数据对象）**
  - PDO 提供设备应用对象的直接访问通道，用来传输实时数据，具有较高的优先权。PDO CAN 报文数据域中每个字节都用作数据传输，报文利用率高。
  - PDO 通过“生产者/消费者”模式来描述，数据从一个生产者传到一个或者多个消费者，数据传送限制在 1~8 个字节。生产者传输数据后，不需要消费者确认，网络上的每个节点都会检测发送节点发出的数据信息，然后节点会决定接收到的信息是否需要处理。
  - 每个 PDO 有两种 PDO 服务：TxPDO 和 RxPDO。生产者发出的 PDO 称为该设备的发送 PDO（TxPDO），消费者设备接收的 PDO 称为该设备的接收 PDO （RxPDO）。
  - 每个 PDO 在对象字典中用 2 个对象描述：PDO 通讯参数和 PDO 映射参数
    PDO 通讯参数：包含哪个 COB-ID 将被 PDO 使用，传输类型，禁止时间和定时器周期。
    PDO 映射参数：包含一个对象字典中对象的列表，这些对象映射到 PDO 里，包括它们的数据长度（in bits）。生产者和消费者必须知道这个映射，以解释 PDO 内容。
  - PDO 传输模式：同步和异步
    同步：同步周期和同步非周期
    异步：数据变化时传送或由事件触发传送

    ➢ **PDO 支持的传输模式如下表所示：**

类型			PDO 传输		
	周期	非周期	同步	异步	RTR
0		X	X		
1 – 240	X		X		
254				X	
255				X	

模式 0：只有当 PDO 数据已改变且同步信号（SYNC）到来时，才传送 PDO 信息。
模式 1~240：每隔 1~240 个同步信号传送一笔 PDO 信息。
模式 254：传送触发事件是制造厂所定义的，本机定义同模式 255。
模式 255：数据变化时传送或由事件触发传送。

➢ **PDO 中的所有传送数据必须由对象字典中映像进来。以下是一个 PDO 映像实例：**



➢ **RxPDO 和 TxPDO 报文格式如下：**

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
对象标识符	数据							

C.4 SDO 介绍

- **SDO（Service Data Object:服务数据对象）**
  - SDO 是用来建立两个 CANopen 设备之间的客户/服务器关系的，客户设备可以对服务器设备的对象字典进行读/写访问操作。SDO 的访问模式为“客户端/服务器”模式，被访问的节点为 SDO 服务器。每个 CANopen 设备至少有一个服务数据对象，用来提供该设备对象字典的访问通道。SDO 可以对对象字典内的所有对象进行读/写访问操作。
  - SDO 报文中包含索引和子索引信息，如此方便对象在对象字典中定位，而且对象字典中的复合数据结构易于通过 SDO 访问。SDO 的触发方式为命令响应型，即 SDO 客户发出读/写请求后，SDO 服务器须给予回应；客户端和服务器均可以主动终止 SDO 的传输；请求报文和响应报文通过不同的 COB-ID 进行区分。
  - SDO 可以传送任意长度的数据。如果传送的数据超过 4 个字节，则必须实行分段传送。最后一段数据包含一个结束标志。
  - SDO 请求报文和响应报文的结构如下：

➢ **请求报文格式：**

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
600（hex）+Node-ID	请求码	对象索引		对象子索引	请求数据			
		LSB	MSB		bit7-0	bit15-8	bit23-16	bit31-24

➢ **请求报文中请求码的含义如下表所示：**

请求码（hex）	含义说明
23	写一个 4 字节数据
2B	写一个 2 字节数据
2F	写一个 1 字节数据
40	读数据
80	停止当前 SDO 命令

➢ **响应报文格式：**

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
580（hex）+Node-ID	响应码	对象索引		对象子索引	响应数据			
		LSB	MSB		bit7-0	bit15-8	bit23-16	bit31-24

➢ **响应报文中响应码的含义如下表所示：**

响应码（hex）	含义说明
43	读 4 字节数据
4B	读 2 字节数据
4F	读 1 字节数据
60	写 1/2/4 字节数据
80	终止 SDO 命令

D

附录D 原点回归模式说明

目录

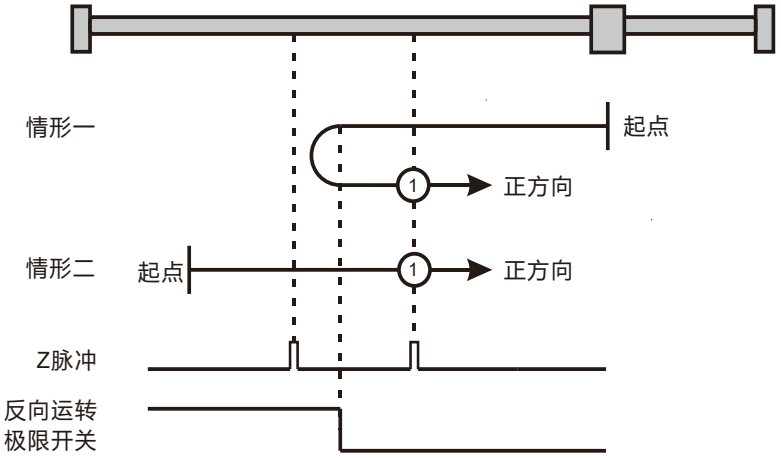
D.1 原点回归各模式详细说明

DVP-15MC 系列运动控制器有多种原点回归模式可供用户选择，用户可根据现场条件和工艺要求选择合适的原点回归模式。

➔ 原点回归模式 1　取决于反向运转极限开关和 Z 脉冲的原点回归

情形一：当反向运转极限开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到反向运转极限开关处于高位时，运动方向改变且以第二段速开始运动；在反向运转极限开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当反向运转极限开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，在反向运转极限开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

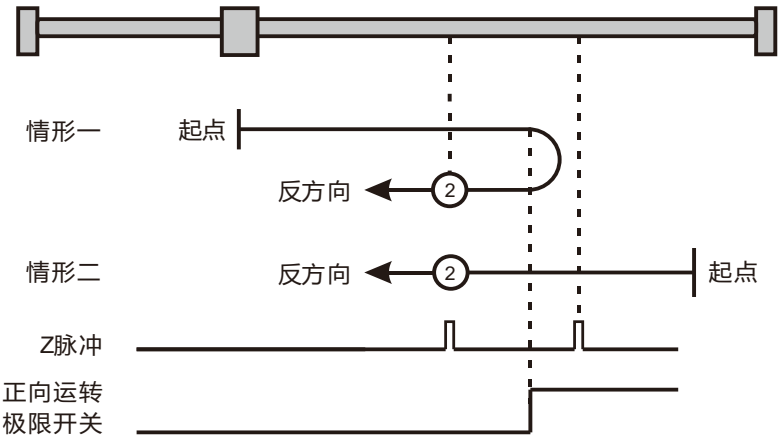


取决于反向运转极限开关和 Z 脉冲的原点回归，上图中的①表示原点回归模式 1。

➔ 原点回归模式 2　取决于正向运转极限开关和 Z 脉冲的原点回归

情形一：当正向运转极限开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到正向运转极限开关处于高位时，运动方向改变且以第二段速开始运动，在正向运转极限开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当正向运转极限开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，在正向运转极限开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。



取决于正向运转极限开关和 Z 脉冲的原点回归，上图中的②表示原点回归模式 2。

模式 3 ~ 模式 6 取决于原点开关和 Z 脉冲的原点回归

➔ 原点回归模式 3

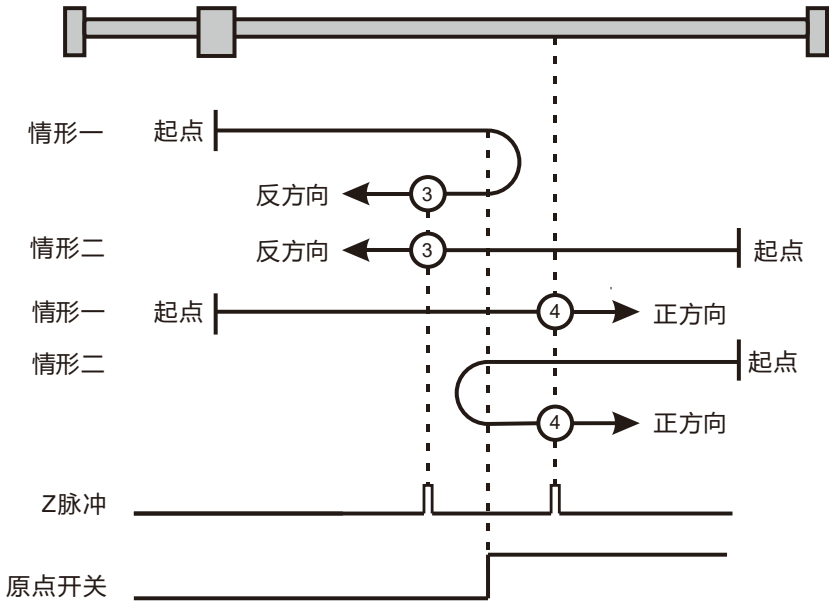
情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

➔ 原点回归模式 4

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。



取决于原点开关和 Z 脉冲的原点回归，上图中的③、④表示原点回归模式 3、模式 4。

➔ 原点回归模式 5

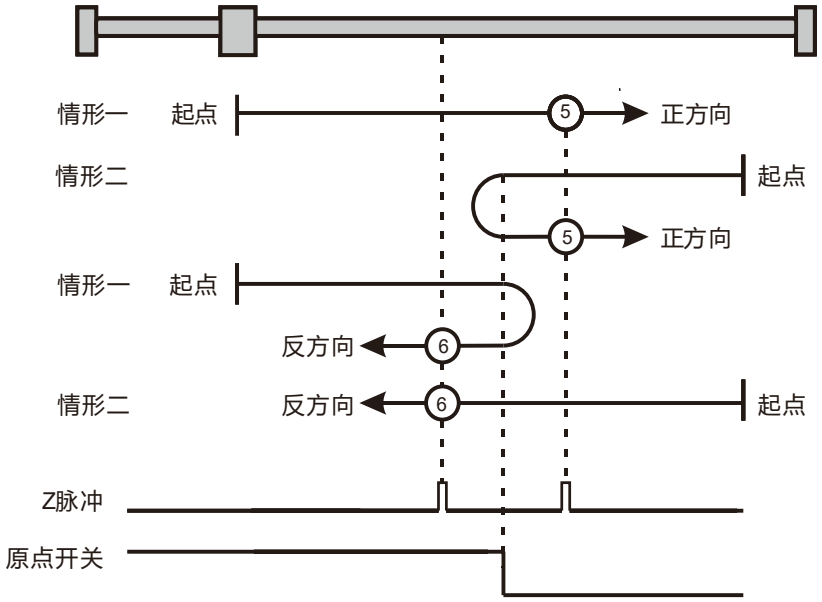
情形一：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

➔ 原点回归模式 6

情形一：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。



取决于原点开关和 Z 脉冲的原点回归，上图中的⑤、⑥表示原点回归模式 5、模式 6。

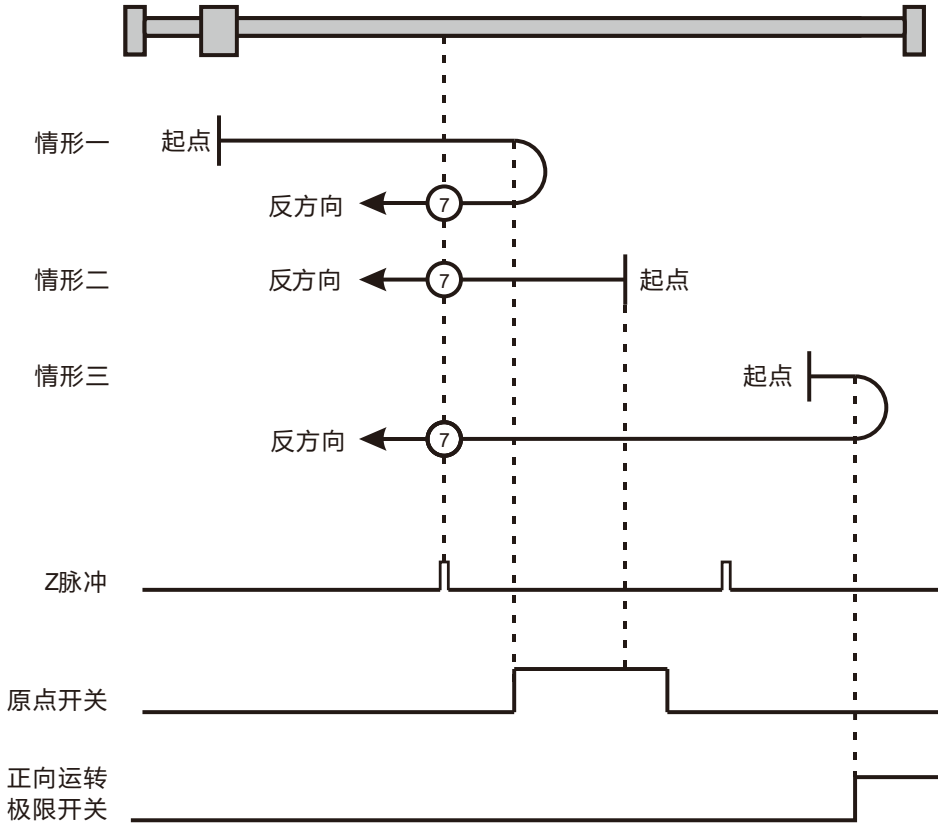
模式 7 ~ 模式 10　取决于原点开关、正向运转极限和 Z 脉冲的原点回归

➔ 原点回归模式 7

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。



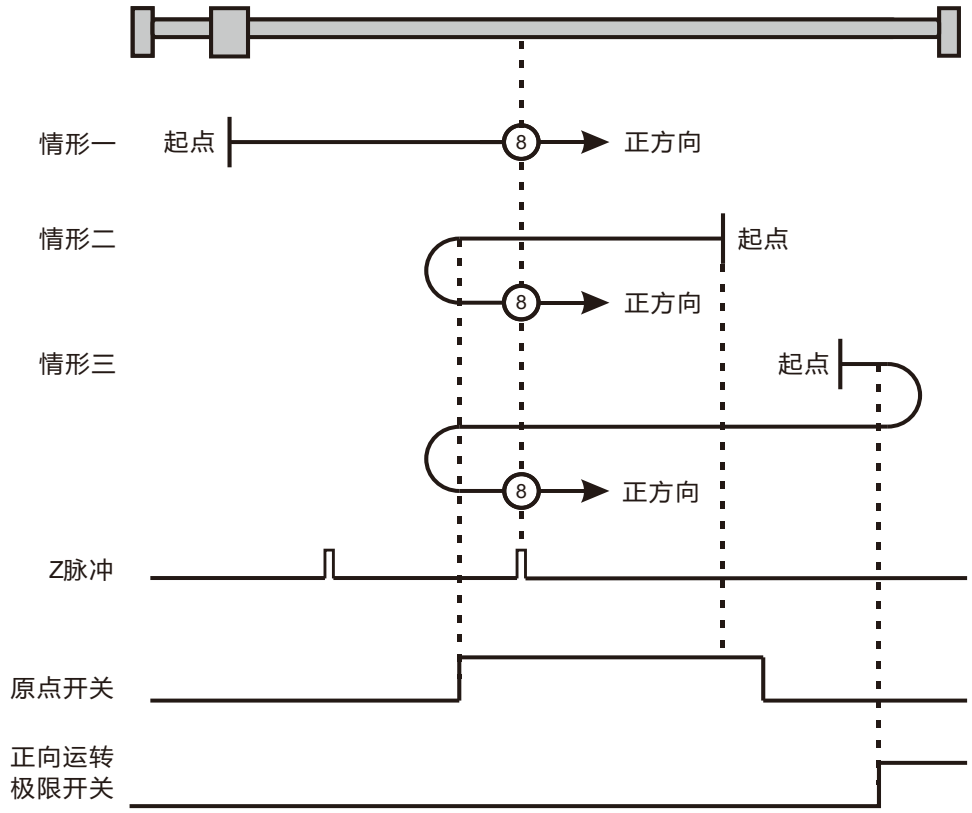
取决于原点开关、正向运转极限开关和 Z 脉冲的原点回归，上图中的⑦表示原点回归模式 7。

➔ 原点回归模式 8

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，仍以第一段速运动，在原点开关状态处于低位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。



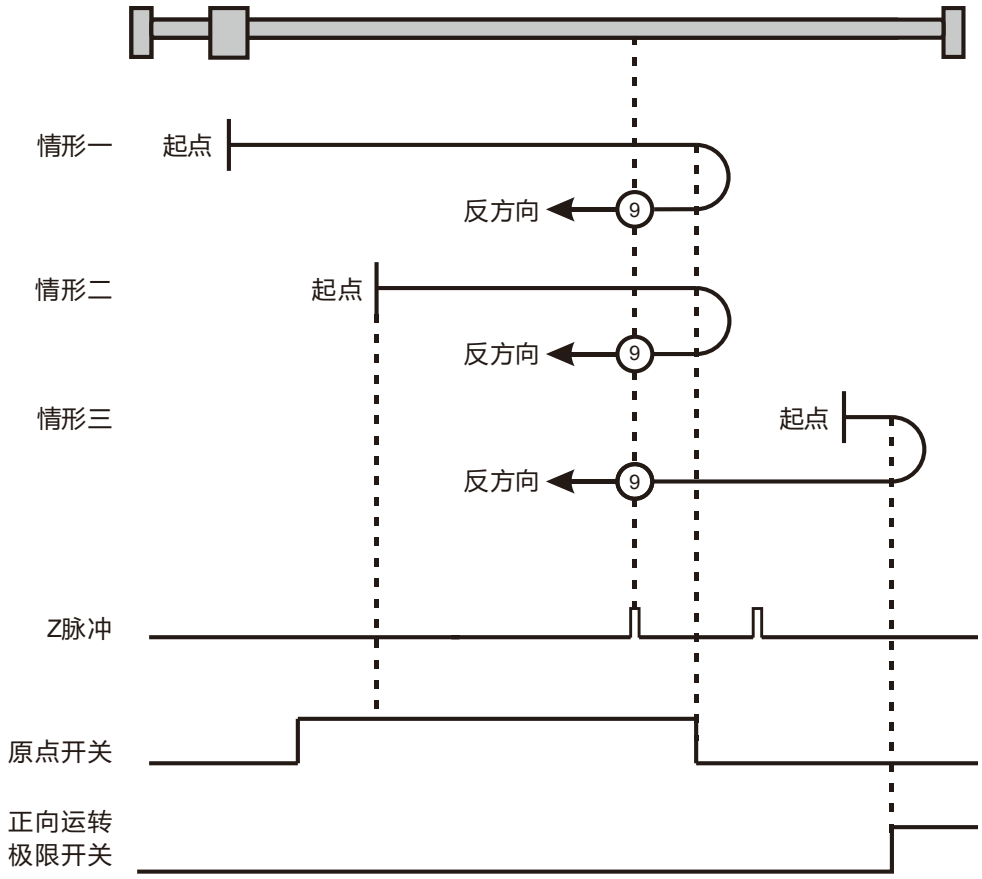
取决于原点开关、正向运转极限开关和 Z 脉冲的原点回归，上图中的⑧表示原点回归模式 8。

➔ 原点回归模式 9

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速正向运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。



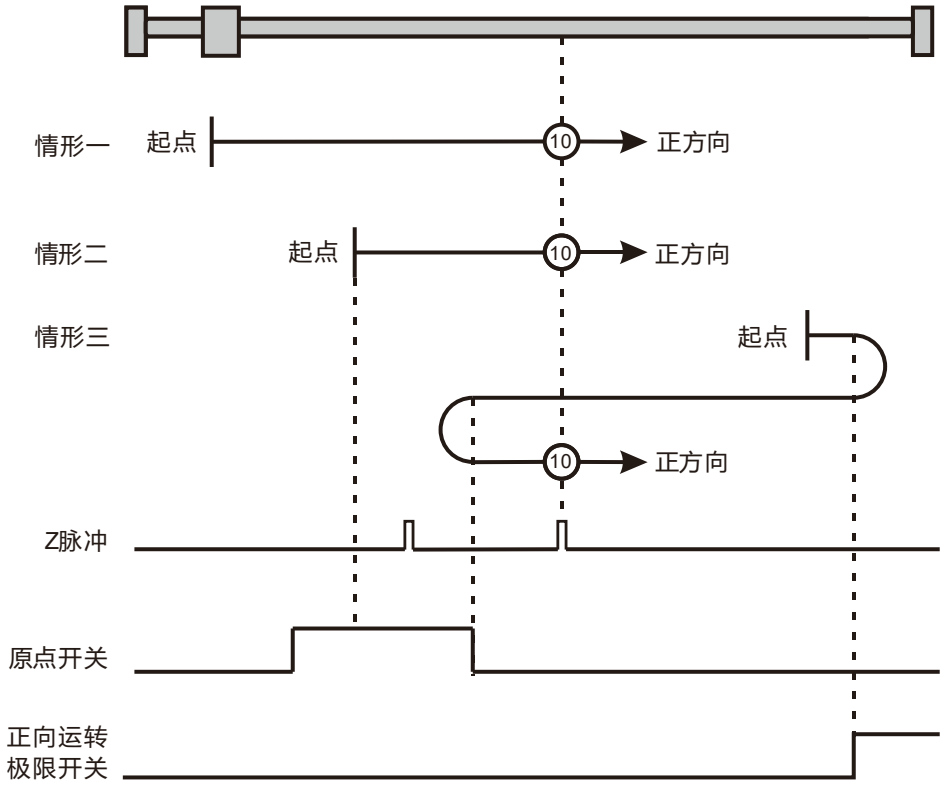
取决于原点开关、正向运转极限开关和 Z 脉冲的原点回归，上图中的⑨表示原点回归模式 9。

➔ 原点回归模式 10

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速正向运动，当遇到原点开关处于低位时，遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，运动方向再次改变且以第二段速开始运动，当原点开关处于低位时，遇到第一个 Z 脉冲的位置就是原点位置。



取决于原点开关、正向运转极限开关和 Z 脉冲的原点回归，上图中的⑩表示原点回归模式 10。

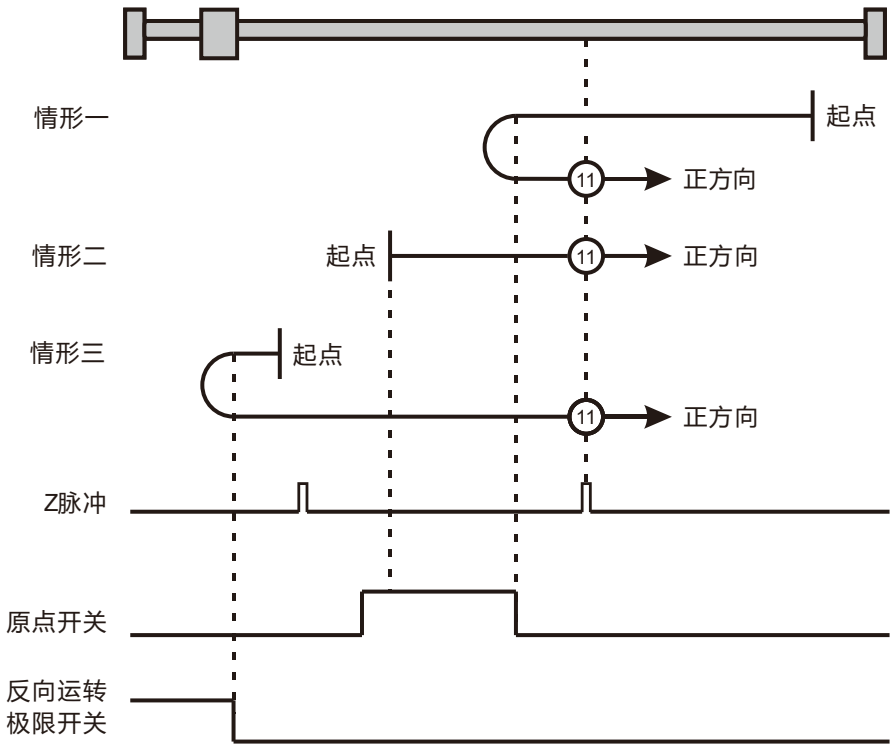
在模式 11 ~ 模式 14 取决于原点开关、反向运转极限和 Z 脉冲的原点回归

➔ 原点回归模式 11

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，在原点开关状态处于低位时遇到第一个 Z 脉冲的位置就是原点位置。



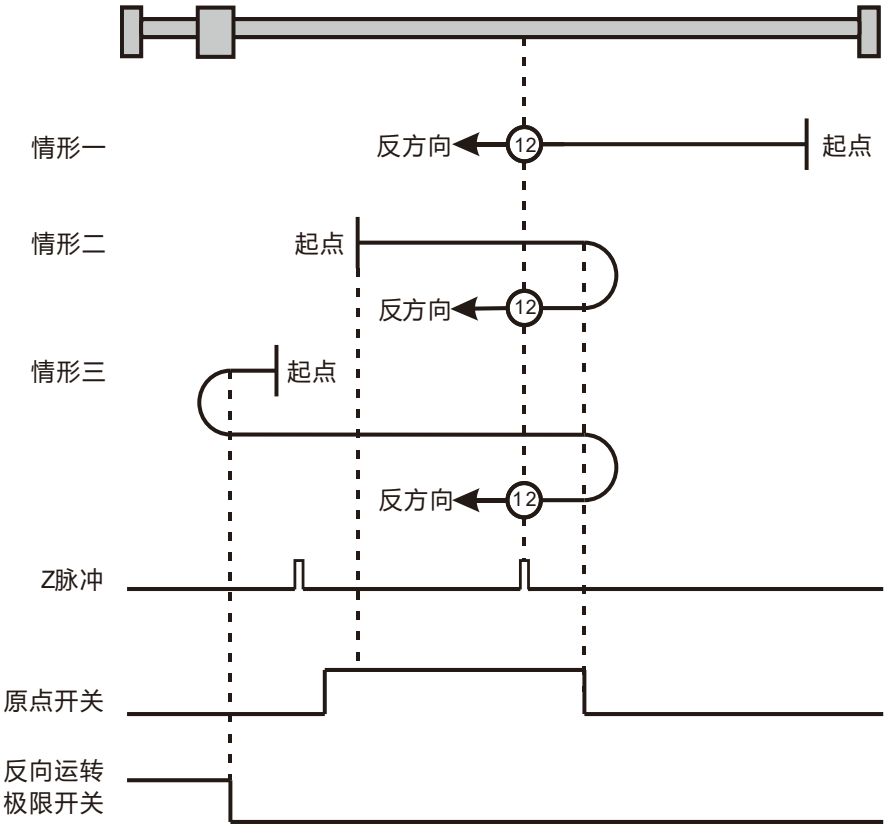
取决于原点开关、反向运转极限开关和 Z 脉冲的原点回归，上图中的⑪表示原点回归模式 11。

➔ 原点回归模式 12

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，仍以第一段速运动，在原点开关状态处于低位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。



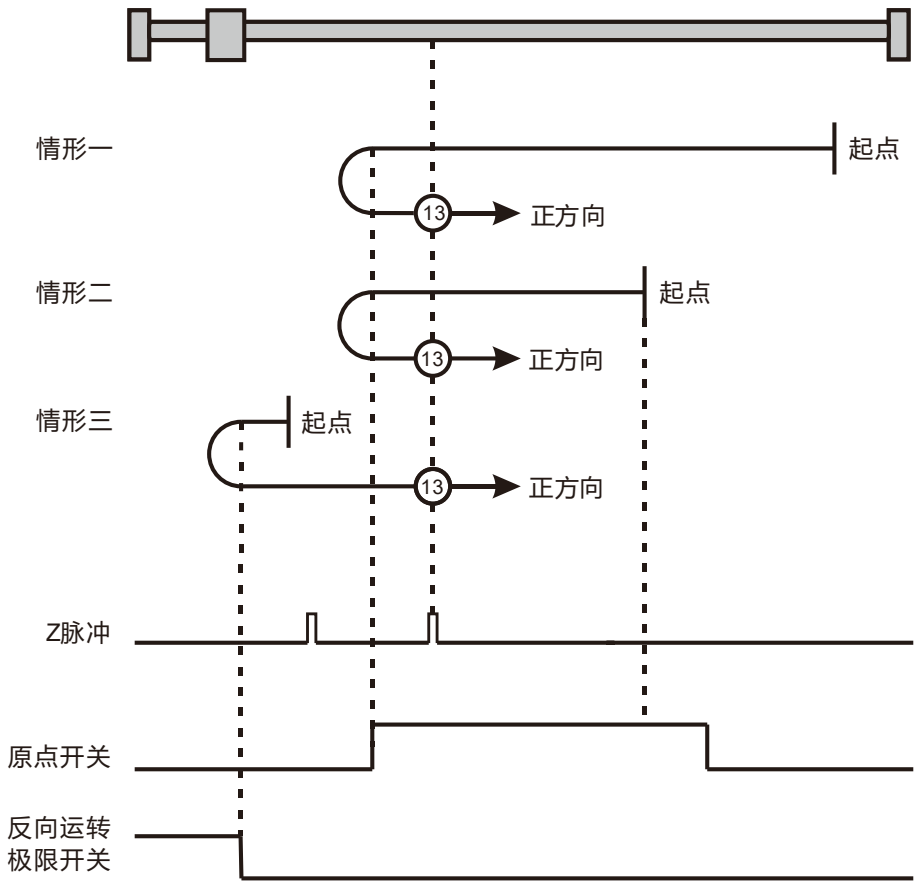
取决于原点开关、反向运转极限开关和 Z 脉冲的原点回归，上图中的⑫表示原点回归模式 12。

➔ 原点回归模式 13

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速反向运动，当遇到原点开关处于低位时，运动方向改变且以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，遇到第一个 Z 脉冲的位置就是原点位置。



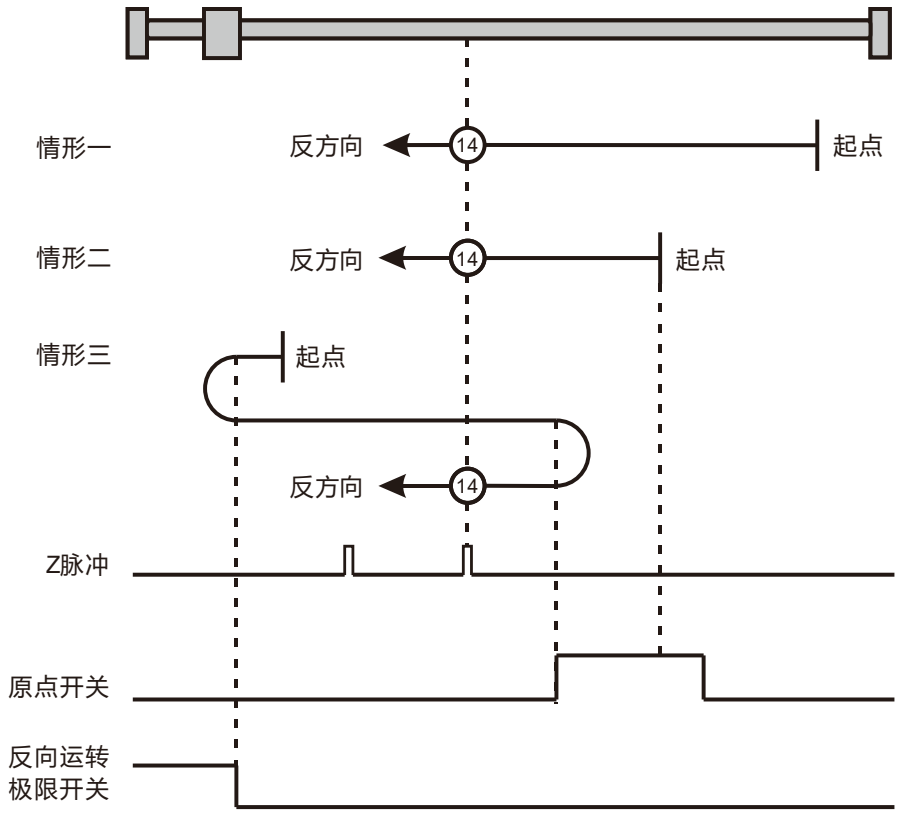
取决于原点开关、反向运转极限开关和 Z 脉冲的原点回归，上图中的⑬表示原点回归模式 13。

➔ 原点回归模式 14

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时，遇到第一个 Z 脉冲的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速反向运动，当遇到原点开关处于低位时，遇到第一个 Z 脉冲的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，运动方向再次改变且以第二段速开始运动，当遇到原点开关处于低位时，遇到第一个 Z 脉冲的位置就是原点位置。



取决于原点开关、反向运转极限开关和 Z 脉冲的原点回归，上图中的⑭表示原点回归模式 14。

模式 15 和模式 16 保留

模式 15 和模式 16 被保留，作为以后发展的原点回归模式。

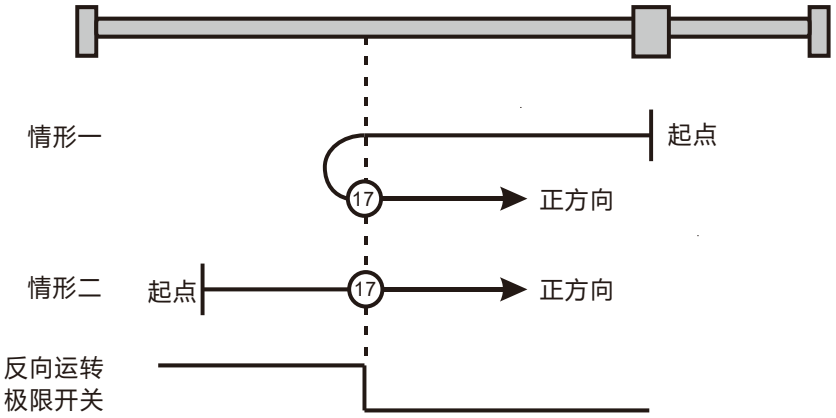
模式 17 ~ 模式 30 不需要 Z 脉冲的原点回归

模式 17 ~ 模式 30 分别和前面所讲的模式 1 ~ 模式 14 相似，只是它们的原点回归位置的定位不再需要 Z 脉冲，而是仅仅根据相关原点开关和极限开关的状态改变来实现。模式 17 与模式 1 相似，模式 18 与模式 2 相似，模式 19 和模式 20 同前面的模式 3 相似，模式 21 和模式 22 同前面的模式 5 相似，模式 23 和模式 24 同前面的模式 7 相似，模式 25 和模式 26 同前面的模式 9 相似。模式 27 和模式 28 同前面的模式 11 相似，模式 29 和模式 30 同前面的模式 13 相似。

➔ 原点回归模式 17 取决于反向运转极限开关的原点回归

情形一：当反向运转极限开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到反向运转极限开关处于高位时，运动方向改变且以第二段速开始运动；在反向运转极限开关状态处于低位时的位置就是原点位置。

情形二：当反向运转极限开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，在反向运转极限开关状态处于低位时的位置就是原点位置。

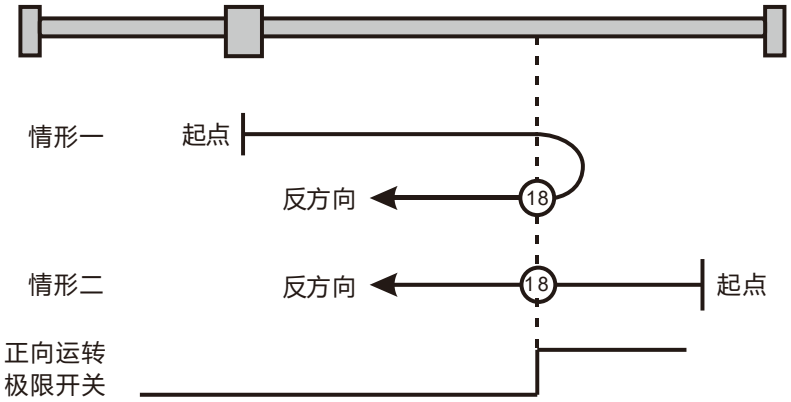


取决于反向运转极限开关的原点回归，上图中的⑰表示原点回归模式 17。

➔ 原点回归模式 18　取决于正向运转极限开关的原点回归

情形一：当正向运转极限开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到正向运转极限开关处于高位时，运动方向改变且以第二段速开始运动，在正向运转极限开关状态处于低位时的位置就是原点位置。

情形二：当正向运转极限开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，在正向运转极限开关状态处于低位时的位置就是原点位置。

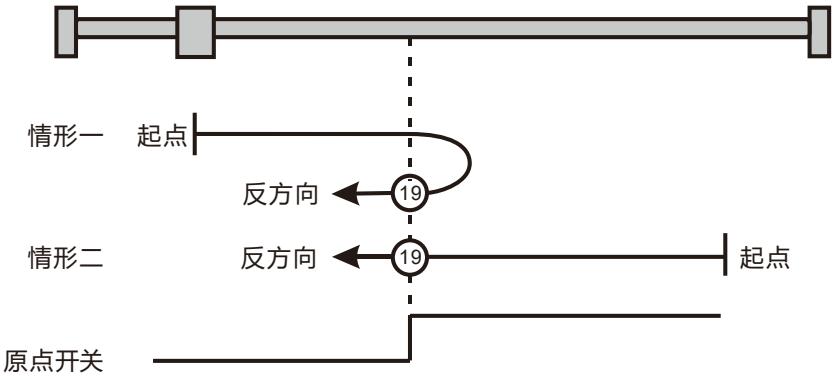


取决于正向运转极限开关的原点回归，上图中的⑱表示原点回归模式 18。

➔ 原点回归模式 19

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，当遇到原点开关处于低位时的位置就是原点位置。

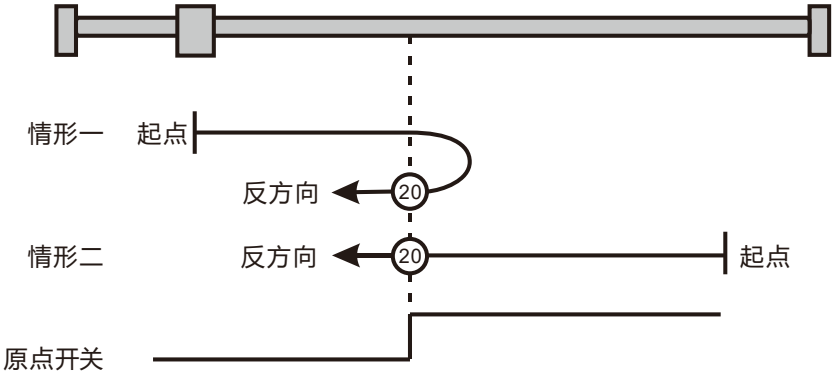


取决于原点开关的原点回归，上图中的⑲表示原点回归模式 19。

➔ 原点回归模式 20

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，当遇到原点开关处于低位时的位置就是原点位置。

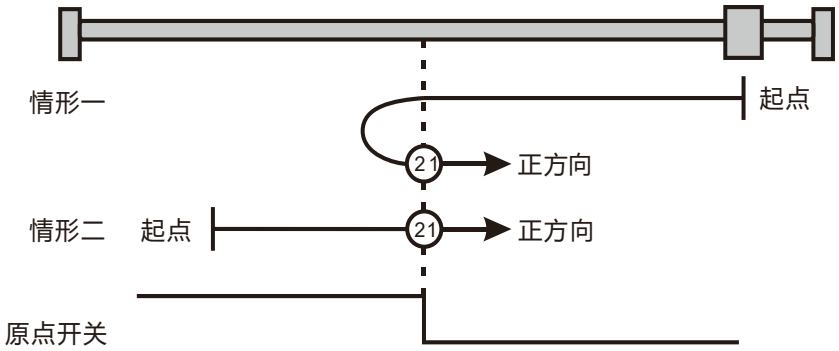


取决于原点开关的原点回归，上图中的⑳表示原点回归模式 20。

➔ 原点回归模式 21

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，当遇到原点开关处于低位时的位置就是原点位置。

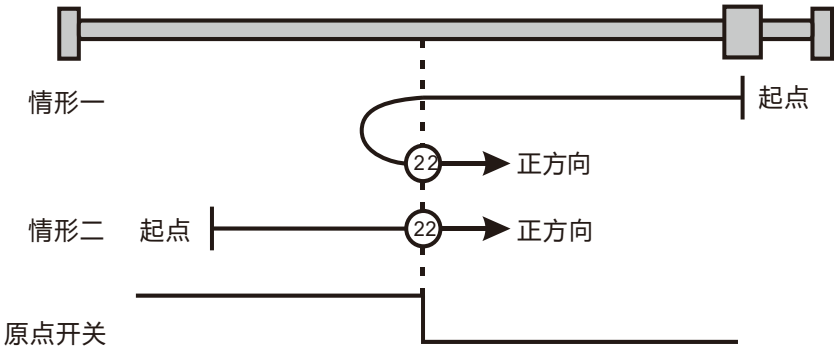


取决于原点开关的原点回归，上图中的㉑表示原点回归模式 21。

➔ 原点回归模式 22

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，当遇到原点开关处于低位时的位置就是原点位置。



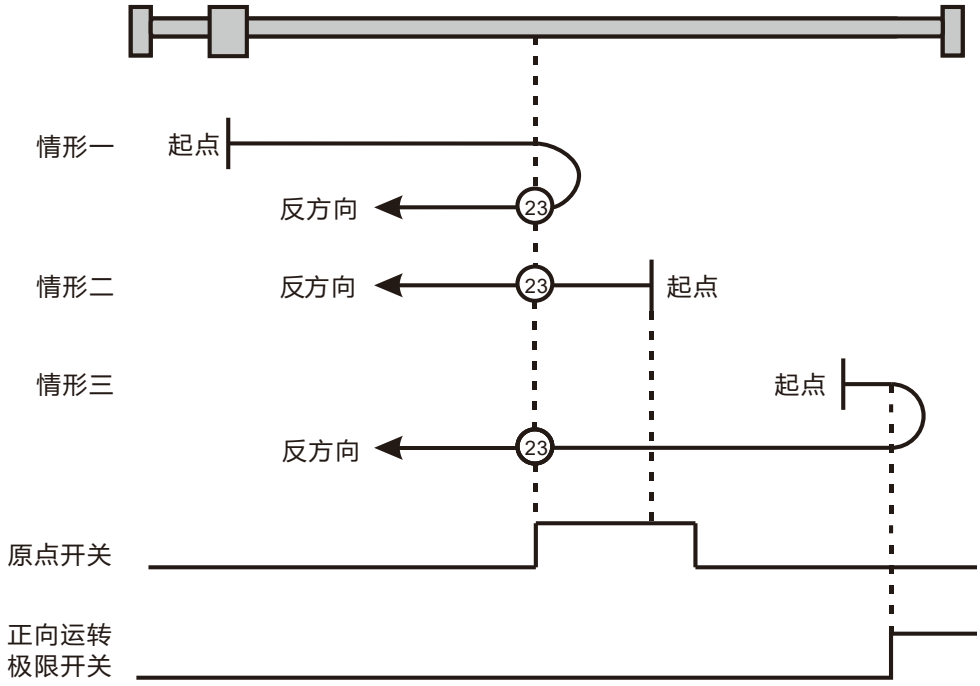
取决于原点开关的原点回归，上图中的㉒表示原点回归模式 22。

➔ 原点回归模式 23

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，在原点开关状态处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。



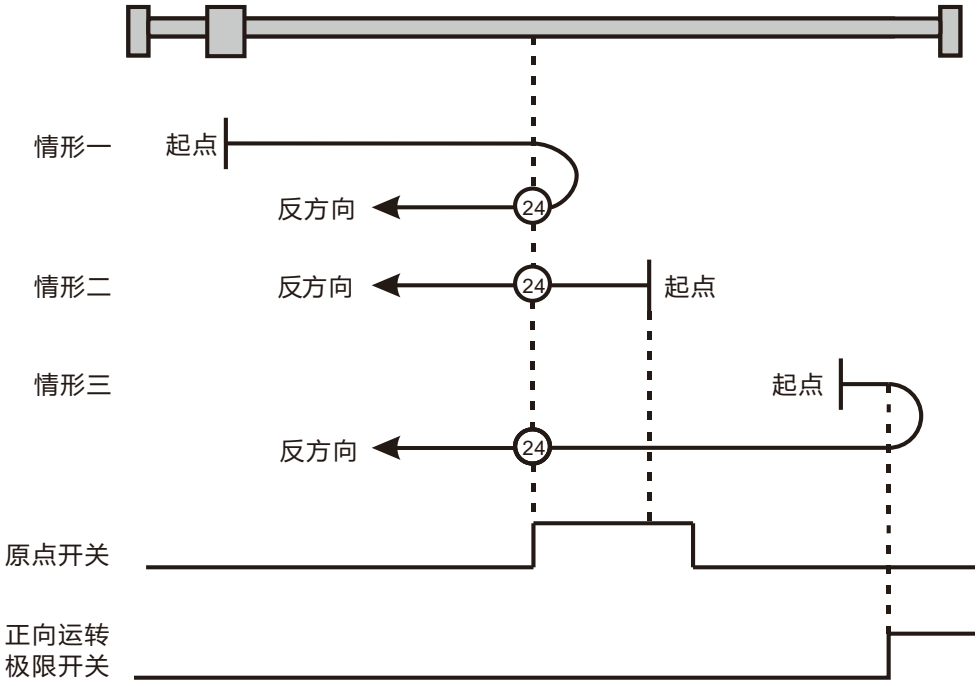
取决于原点开关、正向运转极限开关的原点回归，上图中的㉓表示原点回归模式 23。

➔ 原点回归模式 24

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始反向运动，在原点开关状态处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。



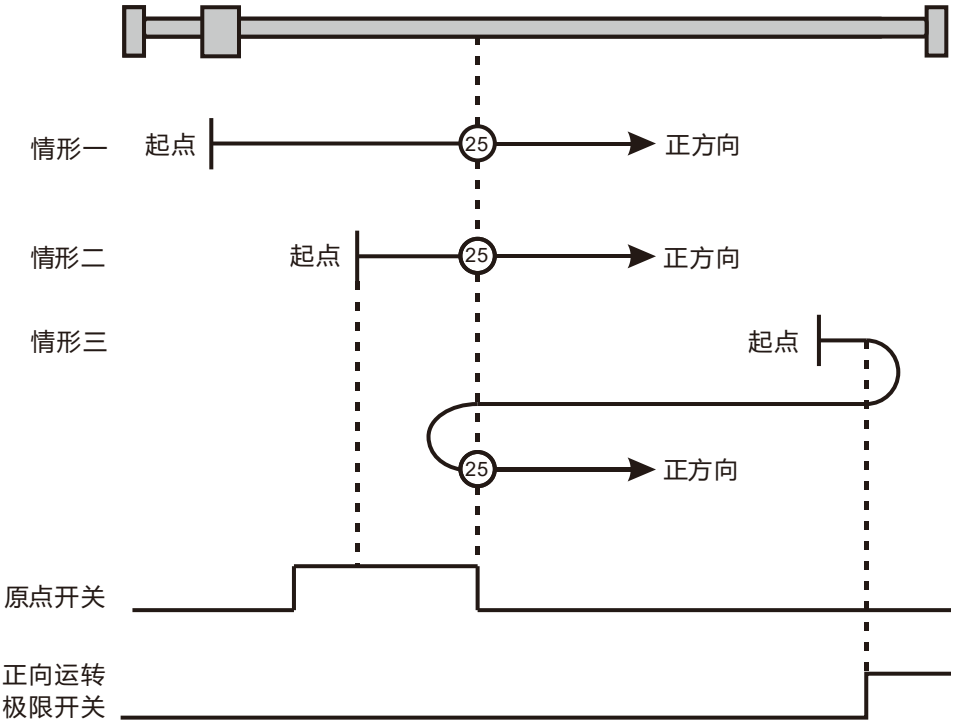
取决于原点开关、正向运转极限开关的原点回归，上图中的㉔表示原点回归模式 24。

➔ 原点回归模式 25

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速正向运动，当遇到原点开关处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，运动方向再次改变且以第二段速开始运动，当原点开关处于低位时的位置就是原点位置。



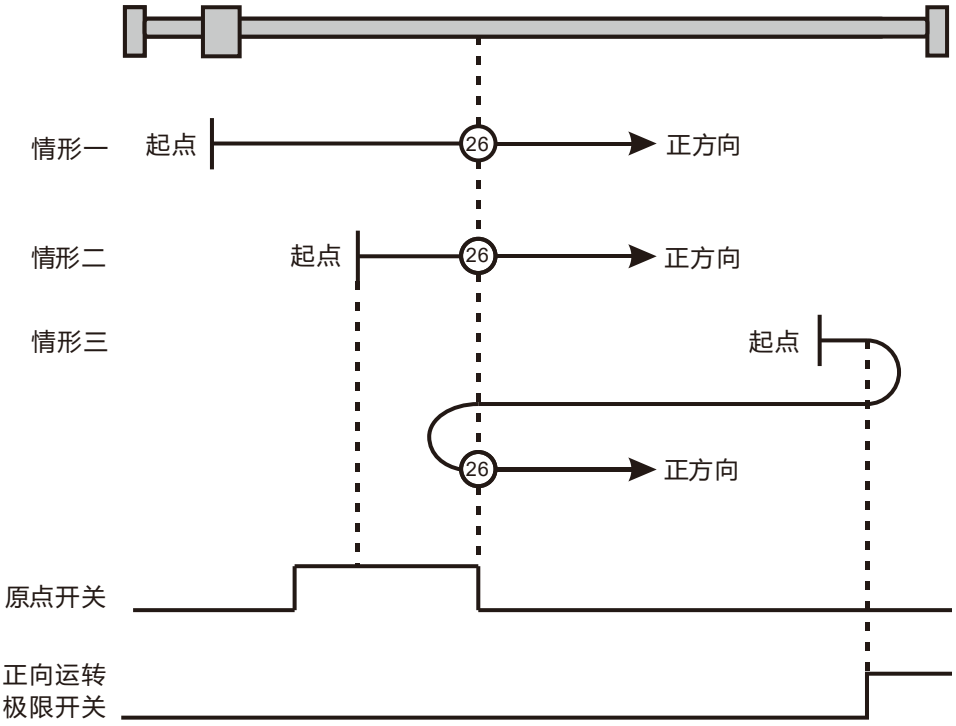
取决于原点开关、正向运转极限开关的原点回归，上图中的㉕表示原点回归模式 25。

➔ 原点回归模式 26

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速正向运动，当遇到原点开关处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速正向运动，当原点开关处于低位且遇到正向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，运动方向再次改变且以第二段速开始运动，当原点开关处于低位时的位置就是原点位置。



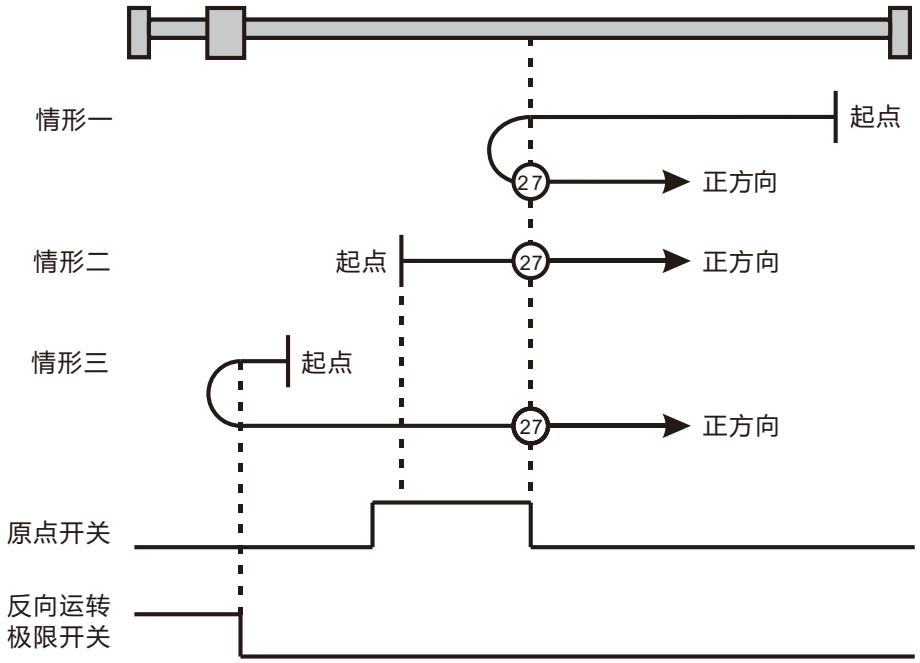
取决于原点开关、正向运转极限开关的原点回归，上图中的(26)表示原点回归模式 26。

➔ 原点回归模式 27

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，在原点开关状态处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。



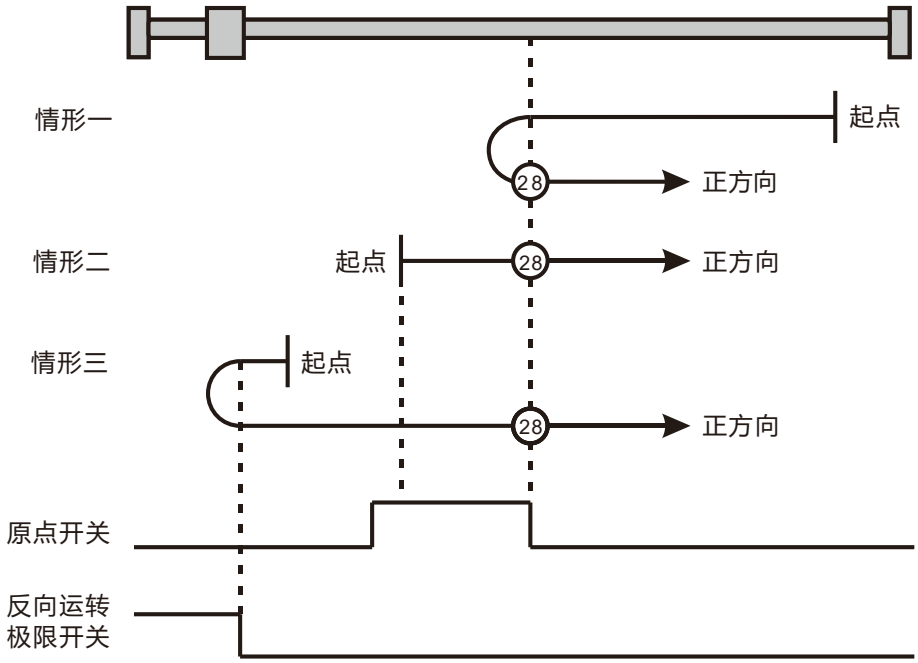
取决于原点开关、反向运转极限开关的原点回归，上图中的㉗表示原点回归模式 27。

➔ 原点回归模式 28

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，运动方向改变且以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴直接以第二段速开始正向运动，在原点开关状态处于低位时的位置就是原点位置。。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，以第二段速开始运动，在原点开关状态处于低位时的位置就是原点位置。



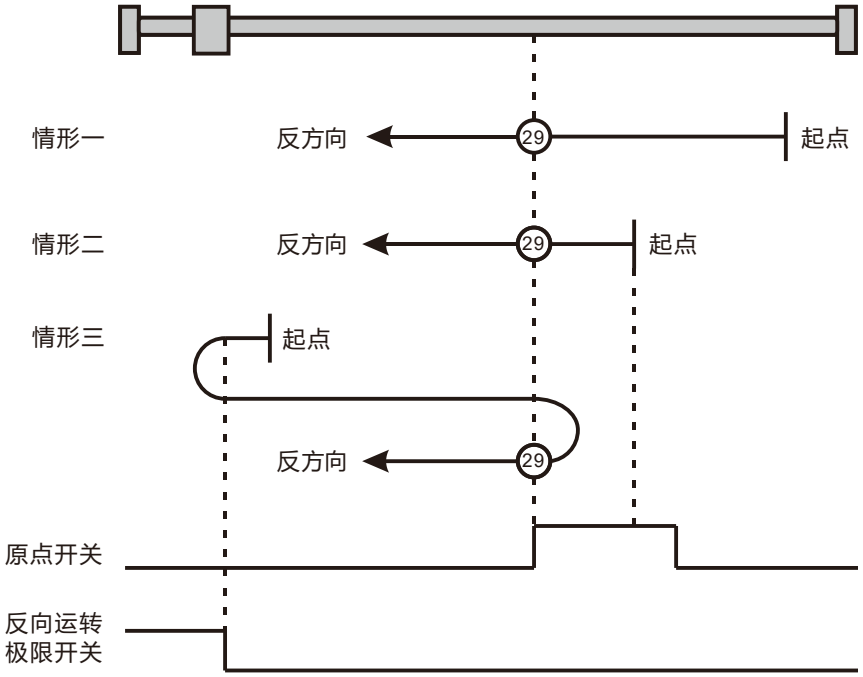
取决于原点开关、反向运转极限开关的原点回归，上图中的㉘表示原点回归模式 28。

➔ 原点回归模式 29

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速反向运动，当遇到原点开关处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，运动方向再次改变且以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。



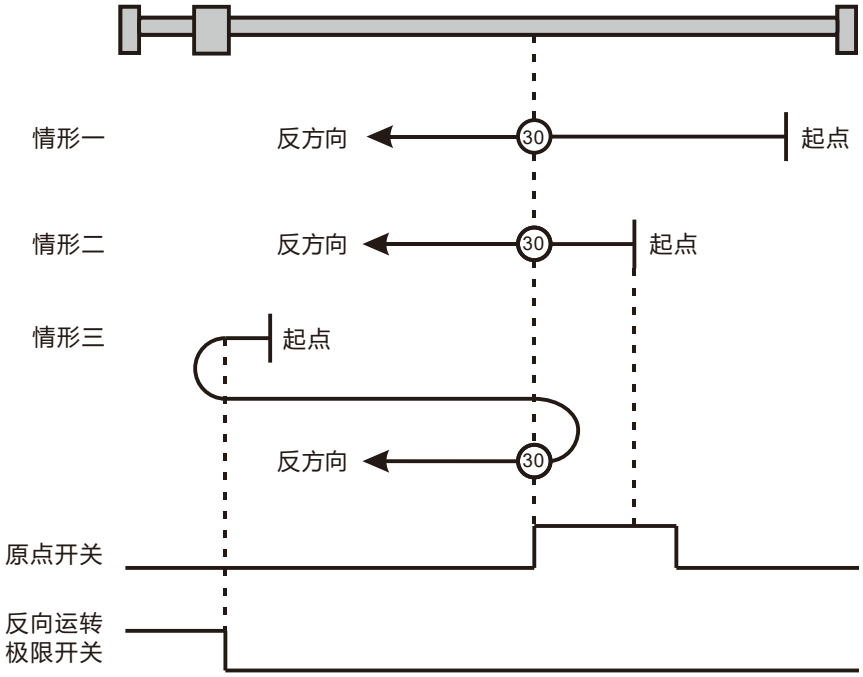
取决于原点开关、反向运转极限开关的原点回归，上图中的㉙表示原点回归模式 29。

➔ 原点回归模式 30

情形一：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当遇到原点开关处于高位时，以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。

情形二：当原点开关状态处于高位时执行 MC_Home 指令，轴开始以第二段速反向运动，当遇到原点开关处于低位时的位置就是原点位置。

情形三：当原点开关状态处于低位时执行 MC_Home 指令，轴开始以第一段速反向运动，当原点开关处于低位且遇到反向运转极限开关处于高位时，运动方向改变且以第一段速开始运动，当遇到原点开关处于高位时，运动方向再次改变且以第二段速开始运动，当遇到原点开关处于低位时的位置就是原点位置。



取决于原点开关、反向运转极限开关的原点回归，上图中的㉚表示原点回归模式 30。

模式 31 和模式 32 保留

模式 31 和模式 32 被保留，作为以后发展的原点回归模式。

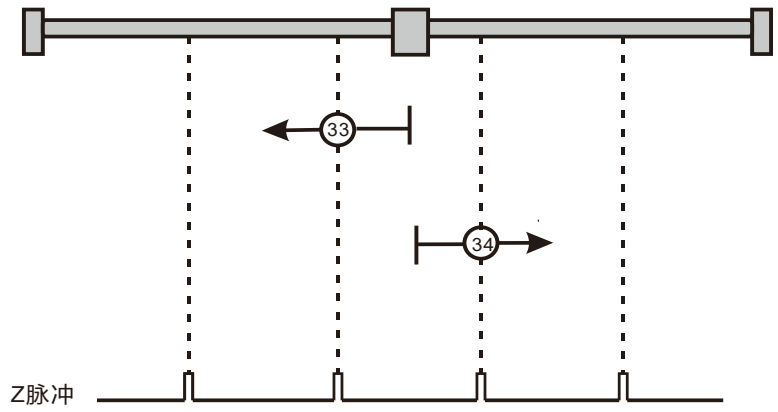
模式 33 ~ 模式 34 取决于 Z 脉冲的原点回归

➔ 原点回归模式 33

在模式 33 下，执行 MC_Home 指令，轴开始以第二段速反向运动，当遇到第一个 Z 脉冲的位置就是原点位置。

➔ 原点回归模式 34

在模式 34 下，执行 MC_Home 指令，轴开始以第二段速正向运动，当遇到第一个 Z 脉冲的位置就是原点位置。



取决于 Z 脉冲的原点回归，上图中的㉝、㉞表示原点回归模式 33、模式 34。

➔ 原点回归模式 35　取决于当前位置的原点回归

在模式 35 下，执行 MC_Home 指令，轴不运动，轴的当前位置被认为是原点回归的位置。

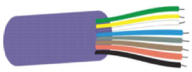
MEMO

附录E 配件说明

目录

E.1 CANopen 通讯相关配件

- 电缆

图示	型号	长度	线径（AWG）
	UC-DN01Z-01A	305M	2#15，2#18 SHLD PVC（粗）
	UC-DN01Z-02A	305M	2#22，2#24 SHLD PVC（细）
	UC-CMC003-01A	0.3M	4#26，1#24　PVC（细）
	UC-CMC005-01A	0.5M	4#26，1#24　PVC（细）
	UC-CMC010-01A	1.0M	4#26，1#24　PVC（细）
	UC-CMC015-01A	1.5M	4#26，1#24　PVC（细）
	UC-CMC020-01A	2.0M	4#26，1#24　PVC（细）
	UC-CMC030-01A	3.0M	4#26，1#24　PVC（细）
	UC-CMC050-01A	5.0M	4#26，1#24　PVC（细）
	UC-CMC100-01A	10.0M	4#26，1#24　PVC（细）
	UC-CMC200-01A	20.0M	4#26，1#24　PVC（细）

注：

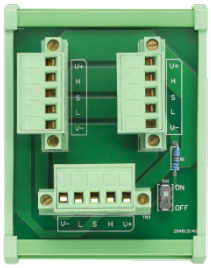
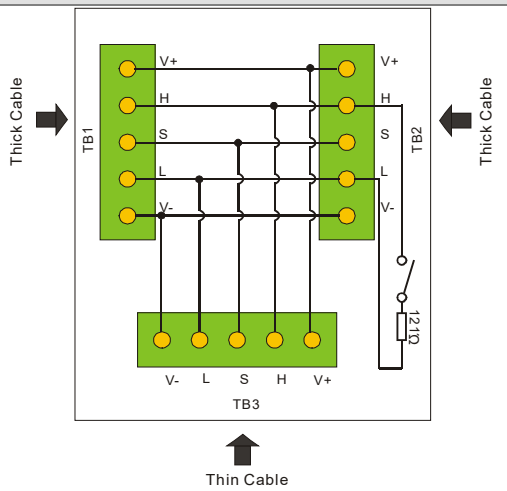
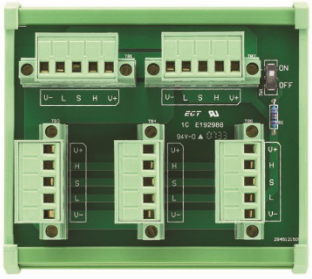
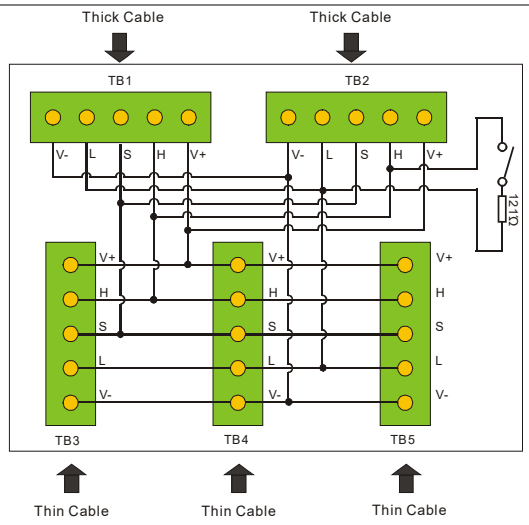
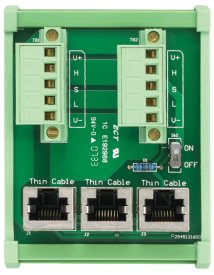
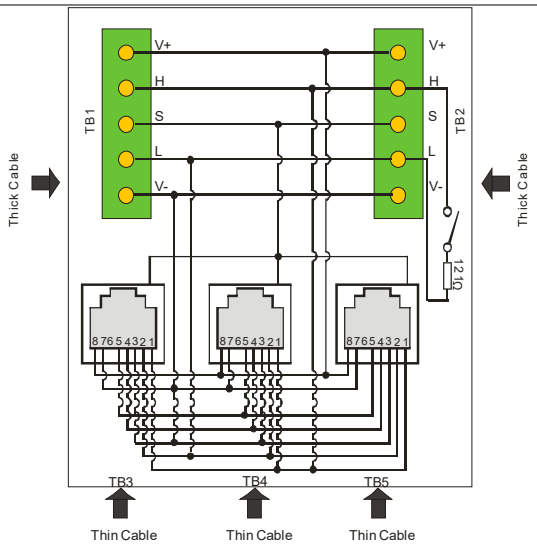
1. 此电缆的最大订购长度为 305M（一卷），最小订购长度为 1M，以公尺为单位。
2. UC-DN01Z-01A 和 UC-DN01Z-02A 可以做为主干线电缆，也可以做为支线电缆。须注意各支持的最长通讯距离不同，此二电缆在不同 CANopen 传输速率下所支持的最长通讯距离如下所示：

CANopen 传输速率（bit/s）	125K	250K	500K	1M
UC-DN01Z-01A 最长通讯距离（m）	500	250	100	40
UC-DN01Z-02A 最长通讯距离（m）	100	100	100	40

3. CANopen 规范了传输速率的最长通讯距离，各通讯速率与最长通讯距离关系如下表所示。

传输速度（bit/s）	10K	20K	50K	125K	250K	500K	800K	1M
最长通讯距离（m）	5000	2500	1000	500	250	100	50	40

- **分接盒**

型号		电路图
TAP-CN01		
TAP-CN02		
TAP-CN03		
连接器	脱落式端子（5.08mm）	
终端电阻	120Ω	

- **终端电阻**

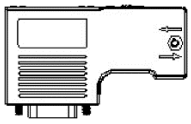
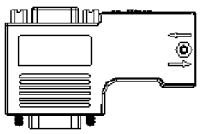
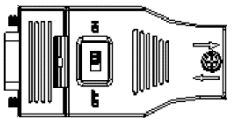
CANopen 建议于 CANopen 通讯电缆的两端分别安装终端电阻，终端电阻作用为通讯信号的阻抗匹配，可降低信号反射干扰正常信号传送的现象，终端电阻阻值为 120Ω（1/4 瓦）。

- 电缆起点的终端电阻：可使用分接盒上的终端电阻，亦即将终端电阻开关设定为 ON。
- 电缆终点的终端电阻：需接一终端电阻 TAP-TR01。
- 终端电阻型号：TAP-TR01，阻值：120Ω（1/4 瓦），如下图所示。

TAP-TR01

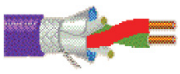

E.2 PROFIBUS DP 通讯相关配件

- **连接器**

	①	②	③
型号	 UN-03PF-01A	 UN-03PF-02A	 UN-03PF-03A
连接头	DB9 公座	DB9 公座	DB9 公座
程序规划接头	--	DB9 母座	--
3. **终端电阻*1**	120Ω	120Ω	120Ω

4. *1.:当连接器位于 PROFIBUS 网络的两端时，请将连接器开关设定为「ON」;当连接器不是位于 PROFIBUS 网络的两端时，请将连接器开关设为「OFF」。

- **电缆**

	型号	长度	线径
	UC-PF01Z-01A	305M	1PR #22 AWG FRFPE FRPE

注：此电缆的最大订购长度为 305M（一卷），最小订购长度为 1M，以公尺为单位。

E.3 DeviceNet 通讯相关配件

- **电缆**

图示	型号	长度	线径（AWG）
	UC-DN01Z-01A	305M	2#15，2#18 SHLD PVC（粗）
	UC-DN01Z-02A	305M	2#22，2#24 SHLD PVC（细）

注：

1. 此电缆的最大订购长度为 305M（一卷），最小订购长度为 1M，以公尺为单位。

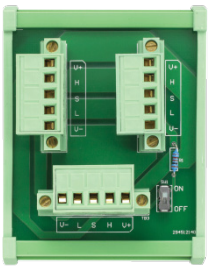
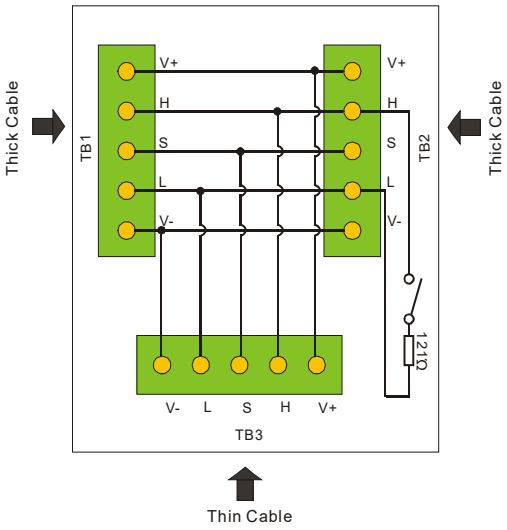
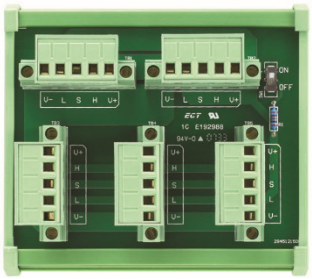
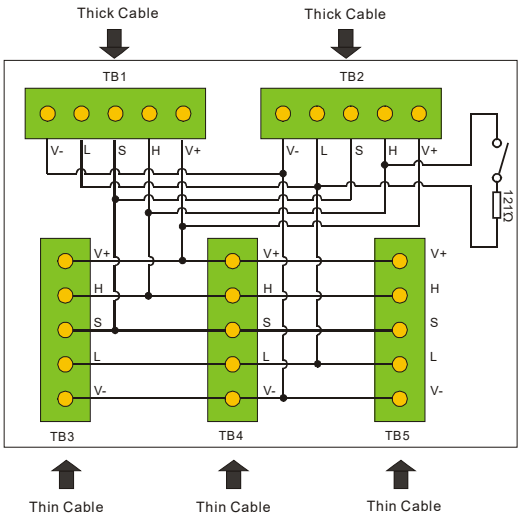
2. UC-DN01Z-01A 和 UC-DN01Z-02A 可以做为主干线电缆，也可以做为支线电缆。须注意各支持的最长通讯距离不同，此二电缆在不同 DeviceNet 传输速率下所支持的最长通讯距离如下所示：

DeviceNet 传输速率（bit/s）	125K	250K	500K
UC-DN01Z-01A 最长通讯距离（m）	500	250	100
UC-DN01Z-02A 最长通讯距离（m）	100	100	100

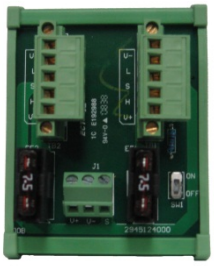
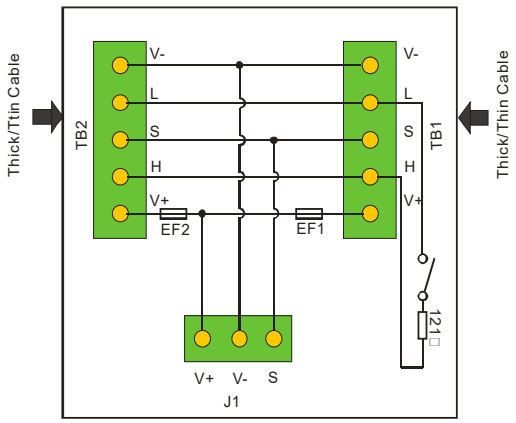
3. DeviceNet 规范了传输速率的最长通讯距离，各通讯速率与最长通讯距离关系如下表所示。

传输速度（bit/s）	10K	20K	50K	125K	250K	500K
最长通讯距离（m）	5000	2500	1000	500	250	100

- **分接盒**

型号		**电路图**
TAP-CN01		
TAP-CN02		

E

型号		电路图
TAP-CP01 （电源分接盒）		
连接器	脱落式端子（5.08mm）	
终端电阻	120Ω	

- **终端电阻**

  DeviceNet 要求在 DeviceNet 通讯电缆的两端分别安装终端电阻，终端电阻的阻值为 120Ω（1/4 瓦）。

  1. 电缆起点的终端电阻：可使用分接盒上的终端电阻，亦即将终端电阻开关设定为 ON。
  2. 电缆终点的终端电阻：需接一终端电阻，阻值为 120Ω（1/4 瓦）。