



AX系列可编程控制器编程手册



深圳市英威腾电气股份有限公司
SHENZHEN INVT ELECTRIC CO., LTD.

前 言

非常感谢您使用 AX 系列可编程控制器。

本手册记载了使用 AX 系列可编程控制器所必需的信息。使用前请详细阅读本手册，充分理解其功能和性能，完成系统构建，发挥其优越性能。

阅读对象

本手册适用于具有电工专业知识的人员（合格的电气工程师或具有同等知识的人员）阅读。

适用产品

AX70 可编程控制器

AX71 可编程控制器

在线支持

除本手册外，还可以通过登录英威腾官方网站获取产品资料和技术支持。

网址：<http://www.invt.com.cn>

如果最终使用为军事单位，或将本产品用于兵器制造等用途时，本产品将列入《中华人民共和国对外贸易法》规定的出口产品管制对象，在出口时，需要进行严格审查，并办理所需的出口手续。

本公司保留对产品不断改进的权利，恕不另行通知。

目 录

前 言	i
阅读对象	i
适用产品	i
在线支持	i
目 录	ii
第 1 章 基础指令	1
1.1 Standard.lib 库指令说明	1
1.1.1 字符串功能	1
1.1.2 双稳态功能模块	4
1.1.3 触发器	4
1.1.4 计数器	5
1.1.5 定时器	7
1.2 Util.lib 库指令说明	11
1.2.1 BCD 转换	11
1.2.2 位/字节函数	11
1.2.3 数学辅助功能块	12
1.2.4 调节器	15
1.2.5 信号发生器	18
1.2.6 操作功能块	20
1.2.7 模拟值处理	22
第 2 章 运动控制指令	24
2.1 SM3_Basic Library 库使用说明	24
2.1.1 MC_Power	24
2.1.2 MC_Halt	25
2.1.3 MC_Home	26
2.1.4 MC_MoveAbsolute	27
2.1.5 MC_AccelerationProfile	29
2.1.6 MC_MoveAdditive	31
2.1.7 MC_MoveRelative	33
2.1.8 MC_MoveSuperImposed	35
2.1.9 MC_MoveVelocity	37
2.1.10 MC_PositionProfile	39
2.1.11 MC_ReadActualPosition	40
2.1.12 MC_ReadBoolParameter	42
2.1.13 MC_ReadAxisError	43
2.1.14 MC_ReadStatus	44
2.1.15 MC_ReadParameter	45
2.1.16 MC_Reset	47
2.1.17 MC_Stop	48
2.1.18 MC_VelocityProfile	49
2.1.19 MC_WriteBoolParameter	51

2.1.20 MC_WriteParameter	52
2.1.21 MC_AbortTrigger.....	53
2.1.22 MC_ReadActualTorque	55
2.1.23 MC_ReadActualVelocity	56
2.1.24 MC_SetPosition	57
2.1.25 MC_TouchProbe	58
2.1.26 MC_MoveContinuousAbsolute.....	60
2.1.27 MC_MoveContinuousRelative.....	62
2.1.28 MC_Jog.....	64
2.1.29 MC_Inch.....	65
2.1.30 SMC3_PersistPosition	67
2.1.31 SMC3_PersistPositionSingleturn	69
2.1.32 SMC3_PersistPositionLogical	71
2.1.33 SMC_Homing.....	72
2.1.34 MC_Camin	77
2.1.35 MC_Camout.....	84
2.1.36 MC_CamTableSelect	86
2.1.37 MC_GearIn	87
2.1.38 MC_GearOut.....	89
2.1.39 MC_GearInPos	90
2.1.40 MC_Phasing	94
第 3 章 INVT 指令	97
3.1 通信指令.....	97
3.1.1 ModbusRTU 主站.....	97
3.1.2 ModbusRTU 从站.....	99
3.1.3 ModbusTCP 主站	100
3.1.4 ModbusTCP 从站	102
3.2 脉冲指令.....	103
3.2.1 MC_InitSys_Invt.....	103
3.2.2 MC_Axis.....	104
3.2.3 MC_Power_Invt	105
3.2.4 MC_Stop_Invt	106
3.2.5 MC_Reset_Invt	107
3.2.6 MC_JOG_Invt	108
3.2.7 MC_JOGP_Invt.....	108
3.2.8 MC_MoveAbsolute_Invt.....	110
3.2.9 MC_MoveRelative_Invt.....	111
3.2.10 MC_MoveVelocity_Invt.....	112
3.2.11 MC_ReadActualPosition_Invt.....	113
3.2.12 MC_ActiveSpeed_Invt.....	113
3.2.13 MC_HOME_Invt.....	114
3.2.14 AMC_Reset_Invt.....	115
3.2.15 AMC_Stop_Invt.....	116
3.2.16 AMC_ActiveSpeed_Invt	117
3.2.17 AMC_MoveAbsolute_Invt.....	117

3.2.18 AMC_MoveRelative_Invt.....	119
3.2.19 AMC_ArcDel_Invt.....	120
3.2.20 MC_SendData_Invt.....	121
3.2.21 MC_GetVersion_Invt.....	122
3.2.22 MC_SetPosition_Invt	123
3.2.23 MC_ReadParameter_Invt	123
3.2.24 MC_ReadStatus_Invt.....	124
3.3 增强指令	126
3.3.1 FB_ReinitDrv.....	126
3.3.2 FB_AxisEasyControl	127
3.3.3 FB_Relative_INVt	128

第1章 基础指令

1.1 Standard.lib 库指令说明

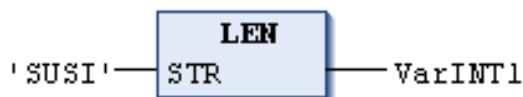
默认条件下, INVTMATIC STUDIO 提供库 standard.lib, 包括由 IEC61131-3 作为一个 IEC 开发系统提供的标准的所有功能和功能块。尽管开发系统明确的辨认出操作符, 标准模块一定会以库的形式导入到一个项目中。

1.1.1 字符串功能

1.1.1.1 LEN

该函数用于获取字符串的长度。输入变量 STR 为 STRING 类型, 返回值为 INT 类型。

FBD 中示例:



ST 中示例:

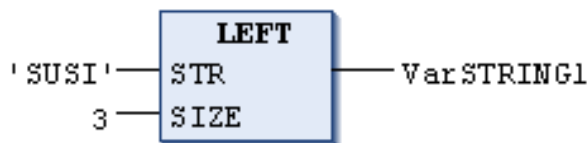
```
VarINT1 := LEN ('SUSI');
```

1.1.1.2 LEFT

该函数功能是获取源字符串中从左边开始的部分字符。输入变量 STR 为 STRING 类型, SIZE 为 INT 类型, 返回值为 STRING 类型。

LEFT (STR, SIZE) 意思是: 获取从字符串 STR 左边开始, 长度为 SIZE 的部分字符。

FBD 中示例:



ST 中示例:

```
VarSTRING1 := LEFT ('SUSI',3);
```

1.1.1.3 RIGHT

该函数功能是获取源字符串中从右边开始的部分字符。输入变量 STR 为 STRING 类型, SIZE 为 INT 类型, 返回值为 STRING 类型。

RIGHT (STR, SIZE) 意思是: 获取从字符串 STR 右边开始, 长度为 SIZE 的部分字符。

FBD 中示例:



ST 中示例：

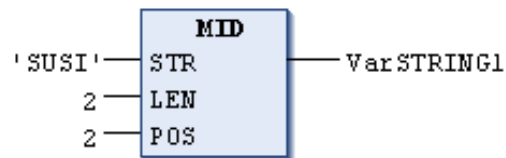
```
VarSTRING1 := RIGHT ('SUSI',3);
```

1.1.1.4 MID

该函数用于获取源字符串中的一部分字符。输入变量 STR 为 STRING 类型，LEN 与 POS 为 INT 类型，返回值为 STRING 类型。

MID (STR, LEN, POS) 的意思是：获取字符串 STR 中从第 POS 个字符位置开始，长度为 LEN 的部分字符。

FBD 中示例：



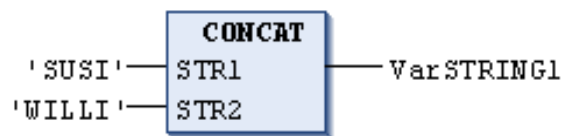
ST 中示例：

```
VarSTRING1 := MID ('SUSI',2,2);
```

1.1.1.5 CONCAT

该函数功能是将两个字符串进行串联组合。输入变量 STR1 与 STR2 以及返回值均为 STRING 类型。

FBD 中示例：



ST 中示例：

```
VarSTRING1 := CONCAT ('SUSI','WILLI');
```

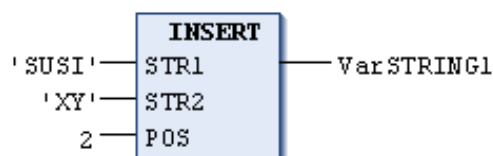
1.1.1.6 INSERT

该函数用于在一个字符串中所指定的位置插入另一个字符串。

输入变量 STR1 与 STR2 都为 STRING 类型，POS 为 INT 类型，返回值为 STRING 类型。

INSERT(STR1, STR2, POS)的意思是：在字符串 STR1 指定的 POS 位置之后插入字符串 STR2。

FBD 中示例：



ST 中示例：

```
VarSTRING1 := INSERT ('SUSI','XY',2);
```

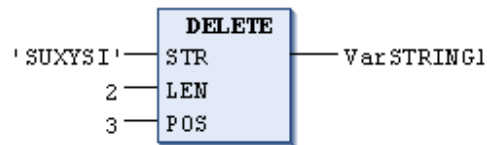
1.1.1.7 DELETE

该函数用于从一个字符串中的指定位置开始删除给定的部分字符。

输入变量 STR 为 STRING 类型，LEN 与 POS 为 INT 类型，返回值为 STRING 类型。

DELETE (STR, L, POS) 意思是：在 STR 中删除 L 字符，在 POS 字符位置开始。

FBD 中示例：



ST 中示例：

```
Var1 := DELETE ('SUXYSI',2,3);
```

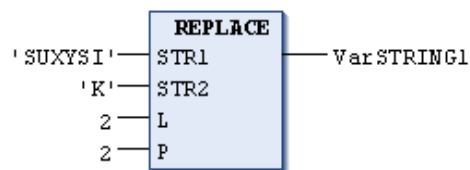
1.1.1.8 REPLACE

该函数用于将一个字符串中的部分字符在指定的位置替换为给定的另一个字符串。

输入变量 STR1 与 STR2 都为 STRING 类型，L 与 P 都为 INT 类型，返回值为 STRING 类型。

REPLACE(STR1, STR2, L, P)的意思是：用字符串 STR2 替换字符串 STR1 中指定的第 P 个字符位置开始，给定长度为 L 的部分字符。

FBD 中示例：



ST 中示例：

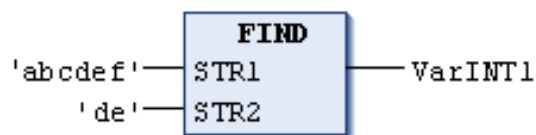
```
VarSTRING1 := REPLACE ('SUXYSI','K',2,2);
```

1.1.1.9 FIND

该函数用于在字符串中查找部分字符。输入变量 STR1 与 STR2 均为 STRING 类型，返回值为 INT 类型。

FIND(STR1, STR2)意思是：寻找到 STR2 首次出现在 STR1 的第一个字符的位置。如果在 STR1 中没有找到 STR2，那么 OUT: =0。

FBD 中示例：



ST 中示例：

```
arINT1 := FIND ('abcdef','de');
```

1.1.2 双稳态功能模块

1.1.2.1 SR

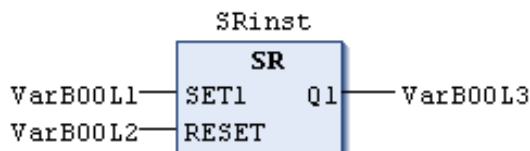
该功能块是置位优先的双稳态功能块。 $Q1 = SR (SET1, RESET)$ 的意思是： $Q1 = (NOT\ RESET\ AND\ Q1)\ OR\ SET1$ 。

输入变量 SET1 与 RESET 以及输出变量 Q1 均为 BOOL 类型。

声明示例：

```
SRInst : SR ;
```

FBD 中示例：



ST 中示例：

```
SRInst(SET1:= VarBOOL1 , RESET:=VarBOOL2 );
VarBOOL3 := SRInst.Q1 ;
```

1.1.2.2 RS

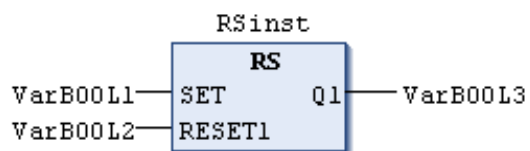
该功能块是复位优先的双稳态功能块。 $Q1 = RS (SET, RESET1)$ 的意思是： $Q1 = NOT\ RESET1\ AND\ (Q1\ OR\ SET)$ 。

输入变量 SET1 与 RESET 以及输出变量 Q1 均为 BOOL 类型。

声明示例：

```
RSInst : RS;
```

FBD 中示例：



ST 中示例：

```
RSInst(SET:= VarBOOL1 , RESET1:=VarBOOL2 );
VarBOOL3 := RSInst.Q1 ;
```

1.1.3 触发器

1.1.3.1 R_TRIG

检测上升沿功能块。

输入：CLK : BOOL; 布尔输入信号用于检测上升沿

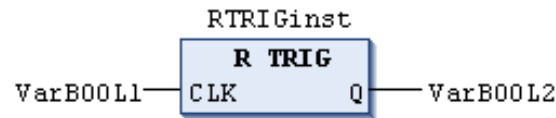
输出：Q : BOOL; 如果 CLK 检测到上升沿输出为 TRUE

当输入变量 CLK 的值为 FALSE，输出变量 Q 和中间辅助变量 M 的值一直保持为 FALSE。当 CLK 的值转为 TRUE，Q 的值将先转为 TRUE，然后 M 的值被置为 TRUE。这意味着当功能块每次被调用时，当 CLK 在上升沿之后转为下降沿时，Q 将返回 FALSE。

声明示例：

```
RTRIGInst : R_TRIG ;
```

FBD 中示例：



ST 中示例：

```
RTRIGInst(CLK:= VarBOOL1);
```

```
VarBOOL2 := RTRIGInst.Q;
```

1.1.3.2 F_TRIG

检测下降沿功能块。

输入：CLK : BOOL; 布尔输入信号用于检测下降沿

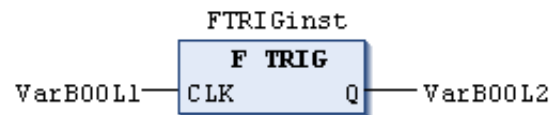
输出：Q : BOOL; 如果 CLK 检测到下降沿输出为 TRUE

只要输入变量 CLK 为 TRUE, 输出变量 Q 和中间变量 M 将一直保持为 FALSE。一旦 CLK 转为 FALSE, Q 将先转为 TRUE, 然后 M 将置为 TRUE。这意味着功能块每次被调用时, 当 CLK 在下降沿之后转为上升沿时, Q 将返回 FALSE。

声明示例：

```
FTRIGInst : F_TRIG ;
```

FBD 中示例：



ST 中示例：

```
FTRIGInst(CLK:= VarBOOL1);
```

```
VarBOOL2 := FTRIGInst.Q;
```

1.1.4 计数器

1.1.4.1 CTU

该功能块为向上计数功能块。

输入：

CU : BOOL; 上升沿触发 CV 的递增计数

RESET : BOOL; 为 TRUE 时, CV 被复位为 0

PV : WORD; CV 计数的上限

输出：

Q : BOOL; 当 CV 达到计数上限 PV 时, 其值为 TRUE

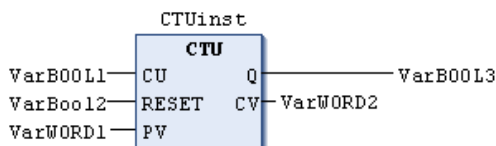
CV : WORD; 不断加 1 的值, 直至 CV

如果 RESET 的值为 TRUE, 计数值 CV 将初始化为 0。如果 CU 从 FALSE 到 TRUE 有一个上升沿, CV 的值加 1。当 CV 大于等于计数上限值 PV 时, Q 的值为 TRUE。

声明示例:

CTUInst:CTU ;

FBD 中示例:



ST 中示例:

CTUInst(CU:= VarBOOL1, RESET:=VarBOOL2 , PV:= VarWORD1);

VarBOOL3 := CTUInst.Q ;

VarWORD2 := CTUInst.CV;

1.1.4.2 CTD

该功能块为向下计数功能块。

输入:

CD : BOOL; 上升沿触发 CV 的递减计数

LOAD : BOOL; 为 TRUE 时, CV 被置为上限值 PV

PV : WORD; CV 开始递减时的初始值

输出:

Q : BOOL; 当 CV 达到 0 时, 其值为 TRUE

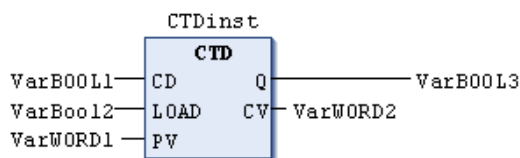
CV : WORD; 不断减 1 的值, 从 PV 开始直至 0

当 LOAD 的值为 TRUE 时, 计数值 CV 将初始化为计数上限值 PV。如果 CD 从 FALSE 到 TRUE 有一个上升沿, 且 CV 大于 0, 则 CV 的值减 1 (即 CV 的值不能小于 0)。当 CV 等于 0 时, Q 的值为 TRUE。

声明示例:

CTDInst:CTD ;

FBD 中示例:



ST 中示例:

CTDInst(CD:= VarBOOL1, LOAD:=VarBOOL2 , PV:= VarWORD1);

VarBOOL3 := CTDInst.Q ;

VarWORD2 := CTDInst.CV;

1.1.4.3 CTUD

该功能块为向上/向下计数功能块。

输入:

CU: BOOL; 上升沿触发 CV 的递增计数

CD: BOOL; 上升沿触发 CV 的递减计数

RESET: BOOL; 为 TRUE 时, CV 被复位为 0

LOAD: BOOL; 为 TRUE 时, CV 被置为上限值 PV

PV: WORD; CV 开始递增时的上限值, 或 CV 递减时的初始值

输出:

QU: BOOL; 当 CV 达到计数上限 PV 时, 其值为 TRUE

QD: BOOL; 当 CV 达到 0 时, 其值为 TRUE

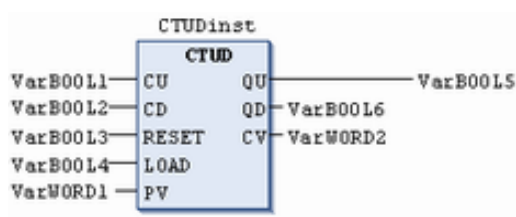
CV: WORD; 不断减 1 的值, 从 PV 开始直至 0

如果 CU 从 FALSE 到 TRUE 有一个上升沿, CV 的值加 1。如果 CD 从 FALSE 到 TRUE 有一个上升沿, 且 CV 大于 0, 则 CV 的值减 1。当 CV 大于等于 PV 时, QU 的值为 TRUE。当 CV 等于 0 时, QD 的值为 TRUE。

声明示例:

CTUDInst:CTUD;

FBD 中示例:



ST 中示例:

```
CTUDInst(CU := VarBOOL1, CD:= VarBOOL2, RESET := VarBOOL3, LOAD:=VarBOOL4 , PV:= VarWORD1);
```

```
VarBOOL5 := CTUDInst.QU;
```

```
VarBOOL6 := CTUDInst.QD;
```

```
VarWORD2 := CTUDInst.CV;
```

1.1.5 定时器

1.1.5.1 TP

该功能块是一个触发定时器功能块。

输入:

IN:BOOL; 上升沿触发 ET 端计时

PT:TIME; ET 计时时间的上限值

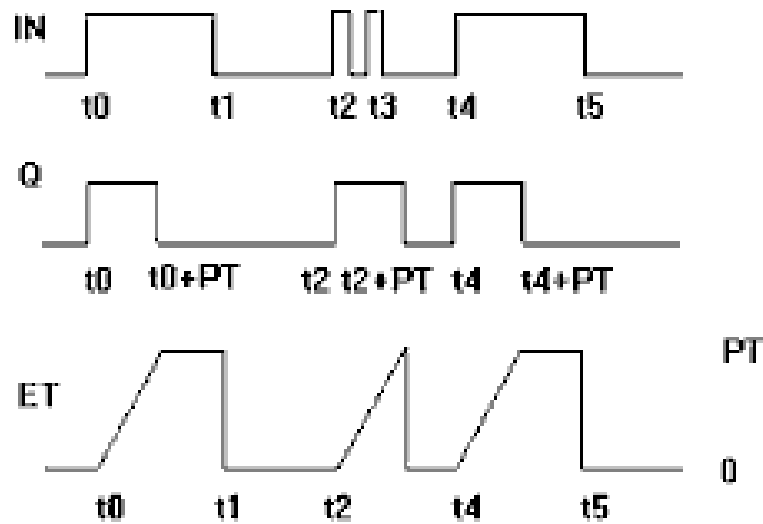
输出:

Q:BOOL; 当 ET 在计时的时候, 其值为 TRUE

ET:TIME; 时间的当前状态

如果 IN 的值为 FALSE，则 Q 的值为 FALSE，且 ET 的值为 0。当 IN 的值为 TRUE 时，ET 中的时间将以毫秒为单位开始计数，直至 ET 等于 PT。ET 等于 PT 之后将一直保持这个常量。如果 IN 为 TRUE 且 ET 小于等于 PT 时，则 Q 为 TRUE，否则 Q 为 FALSE。

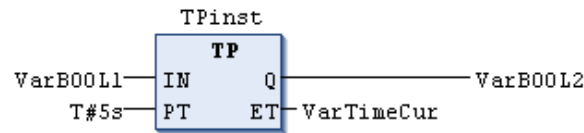
在 PT 所定义的时间周期中，Q 为 TRUE。TP 的时间顺序图如下：



声明示例：

TPInst : TP ;

FBD 中示例：



ST 中示例：

```
TPInst(IN := VarBOOL1, PT:= T#5s);  
  
VarBOOL2 :=TPInst.Q;
```

1.1.5.2 TON

该功能块是一个延时接通定时器。

输入：

IN:BOOL; 上升沿触发 ET 端计时

PT:TIME; ET 计时时间的上限值（即延时时间）

输出：

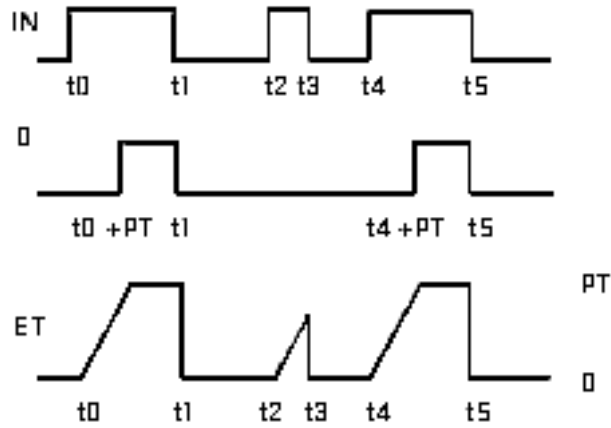
Q:BOOL; 当 ET 计时达到上限值 PT，输出一个上升沿

ET:TIME; 时间的当前状态

TP(IN, PT, Q, ET)的意思是：IN 与 PT 分别为 BOOL 类型与 TIME 类型的输入变量。Q 与 ET 分别为 BOOL 类型与 TIME 类型的输出变量。如果 IN 的值为 FALSE，则 Q 的值为 FALSE，且 ET 的值为 0。

当 IN 的值为 TRUE 时,ET 中的时间将以毫秒为单位开始计数,直至 ET 等于 PT。ET 等于 PT 之后将一直保持这个常量。
如果 IN 为 TRUE 且 ET 等于 PT,则 Q 为 TRUE。否则 Q 为 FALSE。因此,当延时(PT 所定义的时间)完成后,Q 出现一个上升沿。

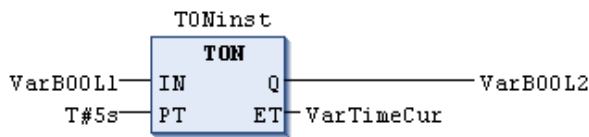
TON 的时间顺序图如下:



声明示例:

```
TONInst : TON ;
```

FBD 中示例:



ST 中示例:

```
TONInst(IN := VarBOOL1, PT:= T#5s);
```

1.1.5.3 TOF

该功能块是一个延时关断定时器。

输入:

IN:BOOL; 下降沿触发 ET 端计时

PT:TIME; ET 计时时间的上限值 (即延时时间)

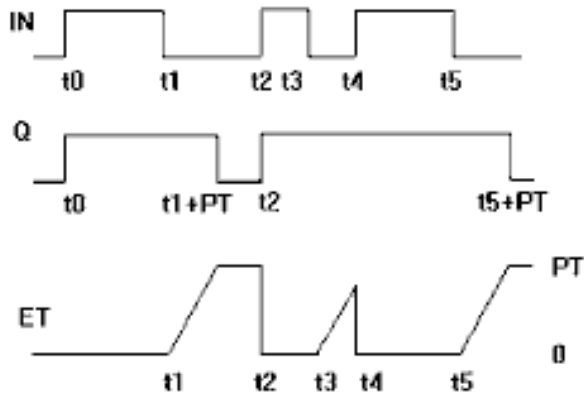
输出:

Q:BOOL; 当 ET 计时达到上限值 PT, 输出一个下降沿

ET:TIME; 时间的当前状态

TOF(IN, PT, Q, ET) 的意思是: 如果 IN 为 TRUE, 则 Q 为 TRUE。当 IN 的值为 FALSE 时, ET 中的时间将以毫秒为单位开始计数,直至 ET 等于 PT。ET 等于 PT 之后将一直保持这个常量。如果 IN 为 FALSE 且 ET 等于 PT, 则 Q 为 FALSE, 否则 Q 为 TRUE。因此当延时完成后, Q 出现一个下降沿。

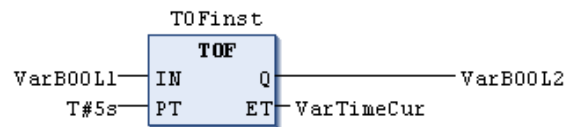
TOF 的时间顺序图如下:



声明示例:

TOFInst :TOF ;

FBD 中示例:



ST 中示例:

```
TOFInst(IN := VarBOOL1, PT:= T#5s);  
VarBOOL2 :=TOFInst.Q;
```

1.1.5.4 RTC

该功能块是一个实时钟定时器功能块。

输入:

EN:BOOL; 上升沿触发 CDT 端的计时
PDT:DATE_AND_TIME; 计时开始的日期和时间

输出:

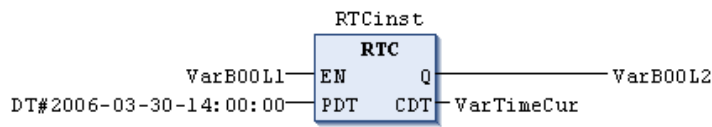
Q:BOOL; 当 CDT 端开始计时，输出为 TRUE
CDT: DATE_AND_TIME; 计时的当前日期和时间

VarBOOL2:=RTC(EN, PDT, Q, CDT)表示: 当 EN 为 FALSE 时, 输出变量 Q 为 FALSE, CDT 为 DT#1970-01-01-00:00:00。一旦 EN 变为 TRUE（上升沿到来），只要 EN 一直保持为 TRUE，CDT 端以 PDT 为初始值以秒为精度递增计时。当 EN 复位为 FALSE，CDT 复位为初始值 DT#1970-01-01-00:00:00。

声明示例:

RTCInst:RTC ;

FBD 中示例:



ST 中示例:

```
RTCInst (EN:=VarBOOL1, PDT:=DT#2006-03-30-14:00:00, Q=>VarBOOL2, CDT=>VarTimeCur);
```

1.2 Util.lib 库指令说明

Util.lib 库包含用于 BCD 码转换、位/字节函数、数学辅助函数和模拟值处理的多种功能块。

1.2.1 BCD 转换

1.2.1.1 BCD_TO_INT

该函数用于将 BCD 格式下的一个字节转换为一个整数值。

输入变量为 BYTE 类型，输出变量为 INT 类型。

当所要转换的字节不是 BCD 格式时，输出为-1。

ST 中示例:

```
i:=BCD_TO_INT(73); (* Result is 49 *)
k:=BCD_TO_INT(151); (* Result is 97 *)
l:=BCD_TO_INT(15); (* Output -1, because it is not in BCD format *)
```

1.2.1.2 INT_TO_BCD

该函数用于将一个整数值转换为 BCD 格式下的一个字节。

输入变量为 INT 类型，输出变量为 BYTE 类型。

当所要转换的整数值不能转换为 BCD 格式时，输出为 255。

ST 中示例:

```
i:=INT_TO_BCD(49); (* Result is 73 *)
k:=BCD_TO_INT(97); (* Result is 151 *)
l:=BCD_TO_INT(100); (* Error! Output: 255 *)
```

1.2.2 位/字节函数

1.2.2.1 EXTRACT

该函数的输入变量 X 为 DWORD 类型，N 为 BYTE 类型。输出变量为 BOOL 类型，输出为输入变量 X 的第 N 位的数值，N 从第 0 位开始计算。

ST 中示例:

```
FLAG:=EXTRACT(X:=81, N:=4);
(* Result : TRUE, because 81 is binary 1010001, so the 4th bit is 1 *)
FLAG:=EXTRACT(X:=33, N:=0);
(* Result : TRUE, because 33 is binary 100001, so the bit '0' is 1 *)
```

1.2.2.2 PACK

该函数用于将 8 个 BOOL 类型输入变量 B0, B1, ..., B7 组合为一个 BYTE 类型数据。

功能块 UNPACK 与该功能有着密切的关系。

1.2.2.3 PUTBIT

该函数的输入变量 X, N 与 B 分别为 DWORD 类型, BYTE 类型与 BOOL 类型。

PUTBIT 用于将 X 的第 N 位设置为值 B, N 从第 0 位开始计算。

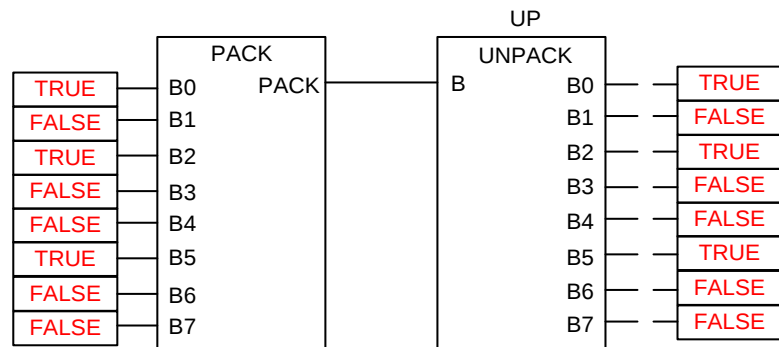
ST 中示例:

```
var1:=38; (* binary 100110 *)
var2:=PUTBIT(A,4,TRUE); (* Result: 54 = 2#110110 *)
var3:=PUTBIT(A,1,FALSE); (* Result: 36 = 2#100100 *)
```

1.2.2.4 UNPACK

UNPACK 用于将 BYTE 类型的输入变量 B 拆分为 8 个 BOOL 类型的输出变量 B0, B1, ...B7。它与 PACK 的功能相反。

FBD 中示例:



1.2.3 数学辅助功能块

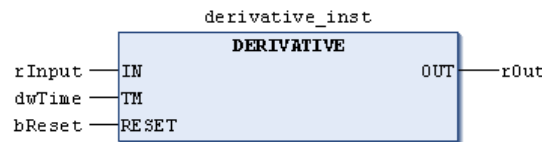
1.2.3.1 DERIVATIVE

该功能块用于确定局部近似导数。

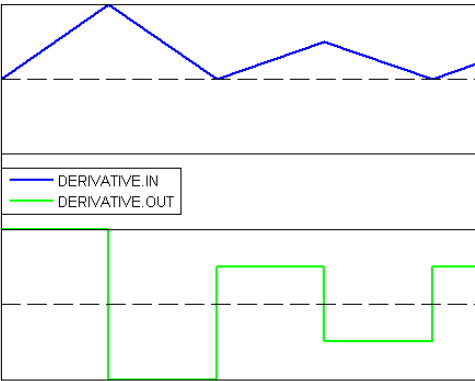
输入变量 IN 为 REAL 类型, TM 为 DWORD 类型, 以毫秒为单位表示时间; RESET 为 BOOL 类型, 其值为 TRUE 时, 功能块复位。输出变量 OUT 为 REAL 类型。

为了取得最精确的结果, DERIVATIVE 近似地使用最后 4 个数值, 以便输入参数中所产生的不精确性尽可能降低。

FBD 中示例:



DERIVATIVE 输入输出：



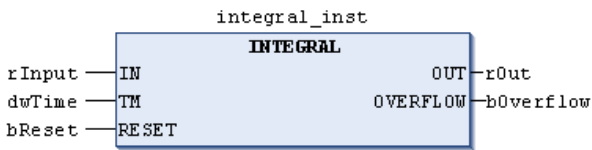
1.2.3.2 INTEGRAL

该功能块用于近似积分。

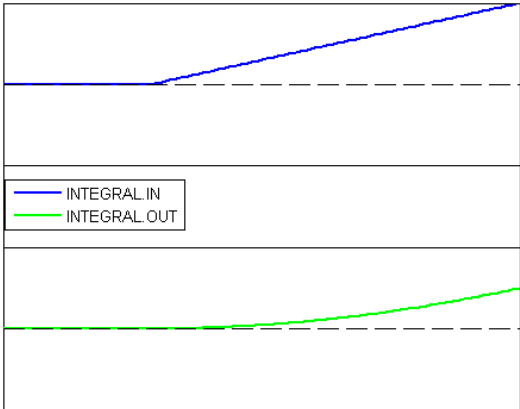
与 DERIVATIVE 类似，输入变量 IN 为 REAL 类型，TM 为 DWORD 类型，以毫秒为单位表示时间；RESET 为 BOOL 类型，当该值为 TRUE 时，功能块复位。输出变量 OUT 为 REAL 类型。

积分为两个阶跃函数的近似值，数据的平均值为近似积分。

FBD 中示例：



INTEGRAL 输入输出：

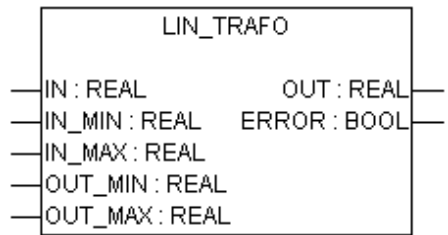


1.2.3.3 LIN_TRAFO

该功能块将由上限值和下限值确定的范围内的实数，转换为由另外的上限值和下限值确定的范围内的实数。

以下表达式基于这种变换：

$$(IN - IN_MIN) : (IN_MAX - IN) = (OUT - OUT_MIN) : (OUT_MAX - OUT)$$



输入变量:

变量	数据类型	描述
IN	REAL	输入变量
IN_MIN	REAL	变量范围低值
IN_MAX	REAL	变量范围高值
OUT_MIN	REAL	输出范围的低值
OUT_MAX	REAL	输出范围的高值

输出变量:

变量	数据类型	描述
OUT	REAL	输出值
ERROR	BOOL	错误发生: TRUE, 如果 IN_MIN = IN_MAX, 或者 IN 超出指定的输入范围

应用示例:

一个温度是由电压值提供的(输入 IN)。然而这些温度值将会被转化为温度值 (输出 OUT)。输入(电压) 值的范围是通过 IN_MIN=0 和 IN_MAX=10 进行限制。输出 (温度值) 值的范围是通过 OUT_MIN=-20 和 OUT_MAX=40 进行限制的。因此对应一个 5 伏电压值将会产生一个 10 度的温度输出。

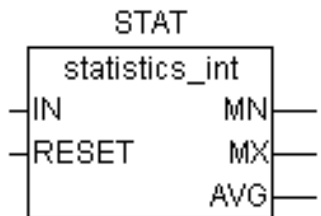
1.2.3.4 STATISTICS_INT

该功能块用于计算一些标准统计数值。

输入变量 IN 为 INT 类型。当 BOOL 类型的输入变量 RESET 为 TRUE 时, 所有数值重新初始化。

输出变量 MN 为 IN 的最小值, MX 为 IN 的最大值, AVG 为平均值。三个输出变量均为 INT 类型。

FBD 中示例:



1.2.3.5 STATISTICS_REAL

该功能块和 STATISTICS_INT 相似, 只是输入变量 IN, 输出变量 MN、MX、AVG 均为 REAL 类型。

1.2.3.6 VARIANCE

VARIANCE 用于计算输入数值的方差。

输入变量 IN 为 REAL 类型，RESET 为 BOOL 类型，输出变量 OUT 为 REAL 类型。

这个功能块用于计算输入数值的方差。当 RESET=TRUE 时，VARIANCE 将被复位。

通过计算方差平方根，可以很容易地得到标准偏差。

1.2.4 调节器

1.2.4.1 PD

该功能块比例-微分调节功能块。

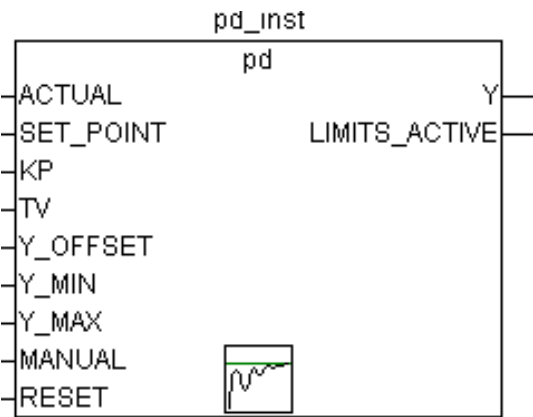
输入变量：

变量	数据类型	描述
ACTUAL	REAL	控制变量的当前值
SET_POINT	REAL	描述值，命令值
KP	REAL	比例系数，P-部分的比例增益
TV	REAL	微分时间，以秒定义的 D-部分时间，例如 "0.5" 表示 500 秒
Y_MANUAL	REAL	如果 MANUAL = TRUE 定义输出值 Y
Y_OFFSET	REAL	操作值 Y 的偏移值
Y_MIN, Y_MAX	REAL	操作值 Y 的低限制值以及高限制值。如果 Y 到达限制值，输出 LIMITS_ACTIVE 将会被设置为 TRUE 并且 Y 将会保持在制定的范围内。这个功能块只在 Y_MIN<Y_MAX 时工作。
MANUAL	BOOL	如果为 TRUE，手动操作将会被激活，那么输出值将会通过 Y_MANUAL 进行定义。
RESET	BOOL	TRUE 复位控制器；在重新初始化时 Y = Y_OFFSET。

输出变量：

变量	数据类型	描述
Y	REAL	操作值，由功能块定义(参阅下文)
LIMITS_ACTIVE	BOOL	TRUE 表明 Y 到达给定的限位值 (Y_MIN, Y_MAX)

FBD 中示例：



Y_OFFSET, Y_MIN 以及 Y_MAX 用于在指定的范围内进行操作数的转换。

MANUAL 可以用于启动或者关闭手动操作。RESET 用于复位控制器。

正常操作时(MANUAL = RESET = LIMITS_ACTIVE = FALSE)，控制器计算偏差值 SET_POINT- ACTUAL，产生时间的相

关微分 de/dt 并将其作为内部变量存储。

输出值 Y ，通过以下方式获得：

$$Y = KP \cdot \left(\Delta + TV \frac{\delta \Delta}{\delta t} \right) + Y_OFFSET$$

其中 $\Delta = SET_POINT - ACTUAL$

所以除了 P -部分以及控制器当前的偏差 (D -部分) 都将会影响计算输出。

另外 Y 限制在 Y_MIN 和 Y_MAX 定义的范围内。如果 Y 处理时到达这个限制值, $LIMITS_ACTIVE$ 将会变为 $TRUE$ 。如果没有计算的限制值, Y_MIN 和 Y_MAX 必须被设置为 0 。

一旦 $MANUAL=TRUE$, Y_MANUAL 将会被写入 Y 。

一个 P 调节可以通过设置 $TV=0$ 实现。

1.2.4.2 PID

该功能块比例-积分-微分调节功能块。

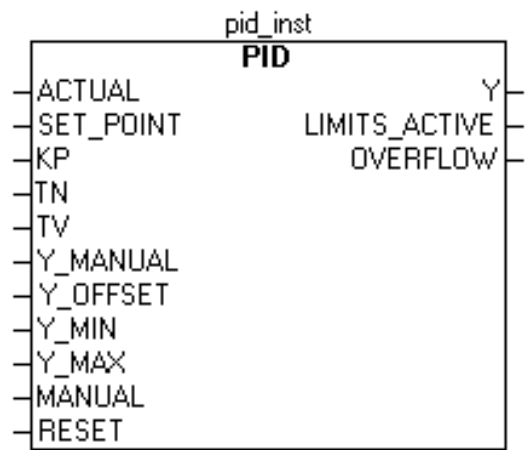
输入变量：

变量	数据类型	描述
ACTUAL	REAL;	控制变量当前的值
SET_POINT	REAL;	期望值, 命令变量
KP	REAL;	比例系数, 在 P -部分的单位增益这个值不能为 0 , 否则功能块将不会执行任何计算。
TN	REAL;	重置时间, i 部分的单位增益定在几秒钟内, 例如“ 0.5 ”为 500 毫秒值必须为 >0 , 否则功能块将不会执行任何计算。 TN 越小, 就会得到更多积分部分, 包括操作变量的值。 TN 增加越多, 这种情况越少。
TV	REAL;	微分作用的时候, D -部分的单位增益定在几秒钟内, 例如“ 0.5 ”为 500 毫秒
Y_MANUAL	REAL;	如果 $MANUAL = TRUE$ 则定义输出值 Y
Y_OFFSET	REAL;	偏移操作变量 Y
Y_MIN, Y_MAX	REAL;	resp 越低, 操作变量 Y 的上限更高。如果 Y 超过这些限制, 输出 $LIMITS_ACTIVE$ 将会设置成真并且 Y 将会保持在规定范围内。如果 $Y_MIN < Y_MAX$, 这些控制才会起作用。
MANUAL	BOOL	如果 $TRUE$, 手动操作将会被激活, 操作变量将会通过 Y_MANUAL 被定义。
RESET	BOOL	在初始化过程 $Y = Y_OFFSET$ 过程中, $TRUE$ 重置控制器。

输出变量：

变量	数据类型	描述
Y	REAL;	操作变量值, 功能块计算 (参见下文)
LIMITS_ACTIVE	BOOL	$TRUE$ 意味着 Y 已经超出了给出的限制 (Y_MIN, Y_MAX)。
OVERFLOW	BOOL	$TRUE$ 意味着溢出 (参见下文)

FBD 中示例:



`Y_OFFSET`, `Y_MIN` 以及 `Y_MAX` 用于在指定的范围内进行操作数的转换。

`MANUAL` 可以用于启动或者关闭手动操作。`RESET` 用于复位控制器。

正常操作时(`MANUAL = RESET = LIMITS_ACTIVE = FALSE`)，控制器计算偏差值 `SET_POINT- ACTUAL`，产生时间的相关微分 de/ dt 并将其作为内部变量存储。

输出值 `Y`，通过以下方式获得：

$$Y = KP \cdot \left(\Delta + \frac{1}{TN} \int edt + TV \frac{\delta \Delta}{\delta t} \right) + Y_OFFSET$$

其中 $\Delta = SET_POINT - ACTUAL$

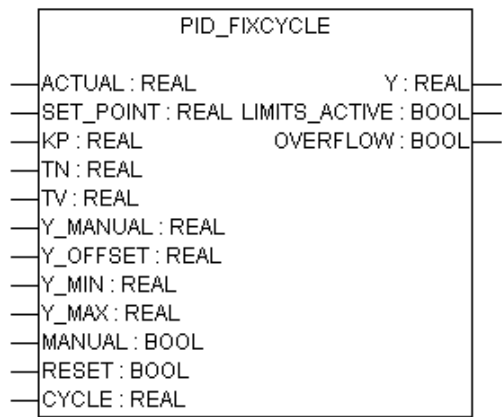
所以除了 `P`-部分以及控制器当前的偏差 (`D`-部分) 都将会影响计算输出。

`PID` 控制器可以通过设置 `TV=0` 轻松地转变成 `PI` 控制器。

因为另外的积分部分，如果错误 `e` 的积分部分变得更大，控制器错误的参数设置可能造成溢出。因此为了安全起见，输出调用 `OVERFLOW` 是可以的，在这种情况下值是 `TRUE`。仅由于不正确的参数设置控制系统不稳定时这种情况才会发生。在相同的时间，控制器将被暂停，将只有通过重新初始化才能再次激活。

1.2.4.3 PID_FIXCYCLE

FBD 中示例:



该功能模块功能和 `PID` 控制器一致，区别是其周期时间由 `CYCLE`（秒）设置，而不是通过一个内部函数自动测量。

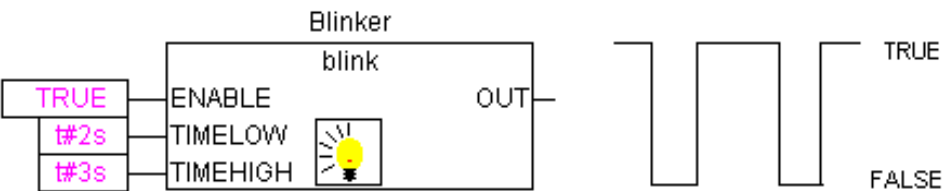
1.2.5 信号发生器

1.2.5.1 BLINK

该功能块用于产生一个脉冲信号。输入变量 **ENABLE** 为 **BOOL** 类型，**TIMELOW** 和 **TIMEHIGH** 为 **TIME** 类型。输出变量 **OUT** 为 **BOOL** 类型。

如果 **ENABLE** 的数值为 **TRUE**，**BLINK** 启动。在 **TIMEHIGH** 中所设定的时间周期内 **OUT** 为 **TRUE**，在 **TIMELOW** 中所设定的时间周期内 **OUT** 为 **FALSE**。

CFC 中示例：



1.2.5.2 FREQ_MEASURE

这个功能块测量布尔输入信号的（平均）频率值(Hz)。可以指定测量周期，一个周期是两个信号上升沿直接的间隔。

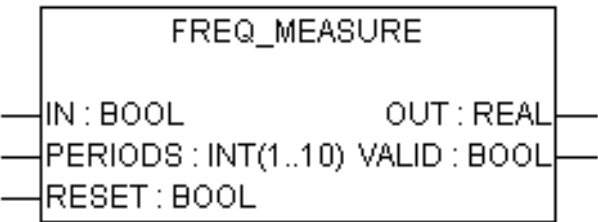
输入变量：

变量	数据类型	描述
IN	BOOL	输入信号
PERIODS	INT	周期序号，两个上升沿之间的时间间隔，通过这个计算输入信号的平均频率。可能的值：1 到 10
RESET	BOOL	复位所有参数到 0

输出变量：

变量	数据类型	描述
OUT	REAL	结果频率[Hz]
VALID	BOOL	FALSE 直到第一次测量结束，或者如果周期 > 3*OUT (表示输入由错误)

FBD 中示例：



1.2.5.3 GEN

该功能块用于生成标准振荡周期。

输入变量 **MODE** 可以预定义 **GEN_MODE** 类型；**BASE** 为 **BOOL** 类型；**PERIOD** 为 **TIME** 类型；**CYCLES** 与 **AMPLITUDE** 为 **INT** 类型；**RESET** 为 **BOOL** 类型。

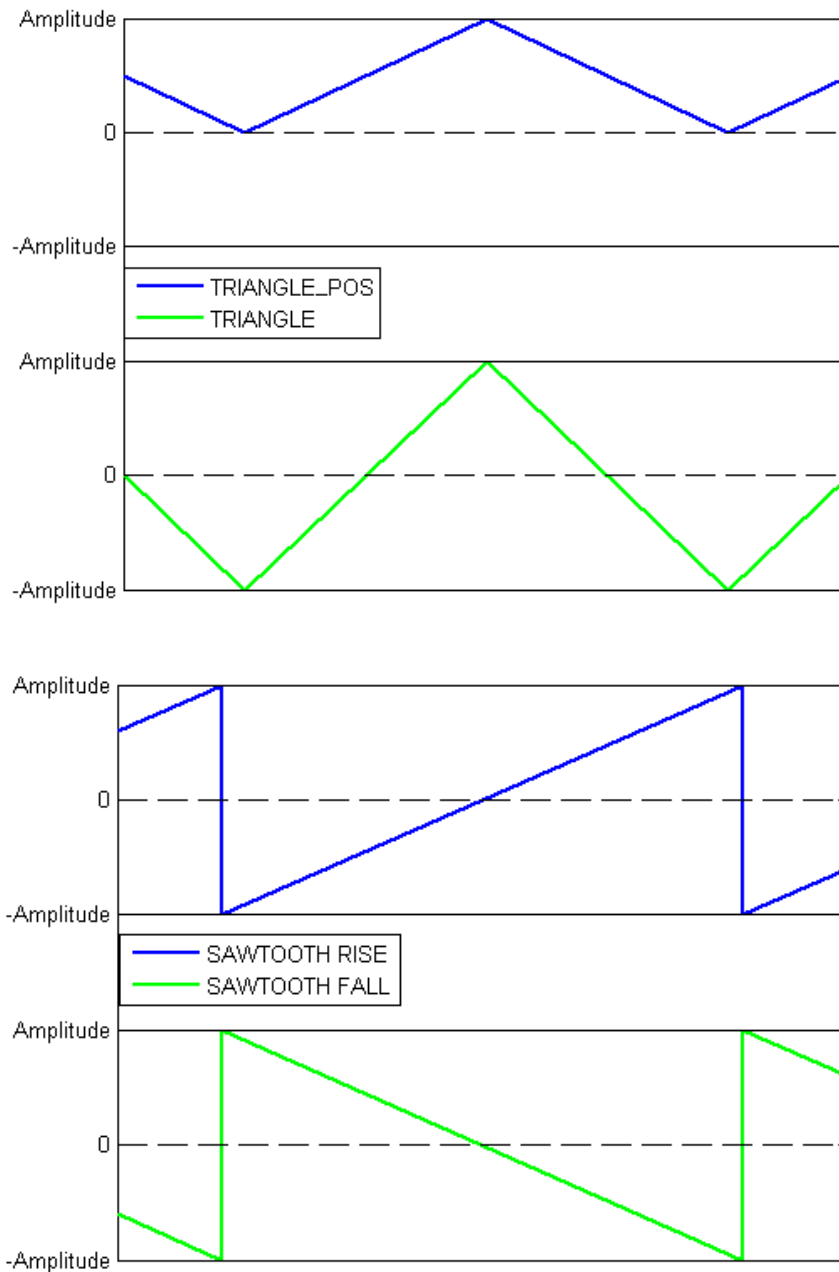
MODE 用于定义所产生的振荡周期模式。其中，枚举值 **TRIANGLE** 与 **TRIANGLE_POS** 为三角波，**SAWTOOTH_RISE** 为递增锯齿波，**SAWTOOTH_FALL** 为递减锯齿波，**RECTANGLE** 为方波，**SINUS** 与 **COSINUS** 分别为正弦波与余弦波。

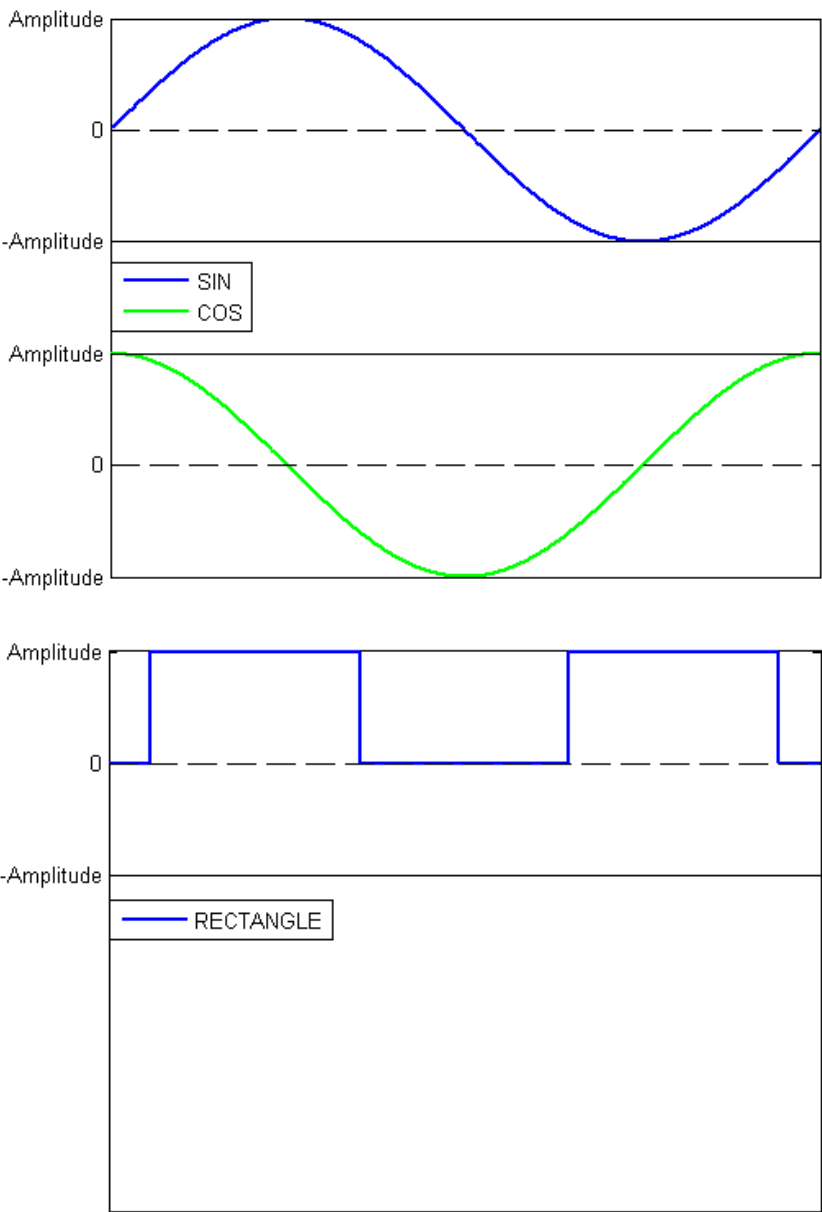
BASE 用于定义是否使用所设置的时间定义循环周期（**BASE=TRUE**）或者使用表示调用功能块次数的特定周期数值定义循环周期（**BASE=FALSE**）。

PERIOD 或 **CYCLES** 用于定义相应的循环周期。

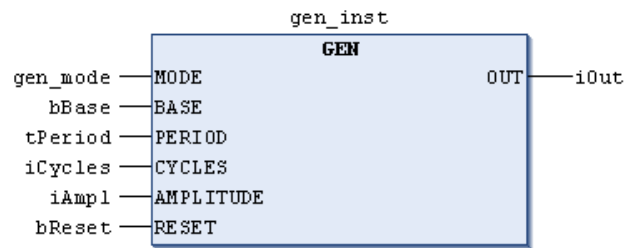
AMPLITUDE 用于定义所产生的振幅。

当 **RESET=TRUE**，信号发生器复位为 0。





CFC 中示例：



1.2.6 操作功能块

1.2.6.1 CHARCURVE

该功能块用于以线性的形式逐渐地表示数值。

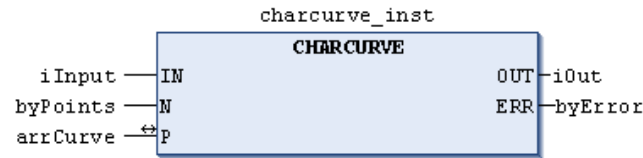
输入 IN 为 INT 类型，用于设置进行处理数值；N 为 BYTE 类型，用于设置点的数量。P 为基于两个整数值（X 与 Y）的预定义 POINT 类型，数组 P[0..10]用于产生特征线。

输出变量 OUT 为 INT 类型，用于输出已处理的数据；ERR 为 BYTE 类型，用于显示错误。

数组中的点 P[0]..P[N-1]必须依照它们的 X 值的大小来存储,否则 ERR 返回值 1。如果输入 IN 的数值不在 P[0].X 与 P[N-1].X 之间，ERR=2，并且 OUT 为相应的限值 P[0].Y 或 P[N-1].Y。

如果 N 的数值在允许值 2~11 之外，那么 ERR=4。

FBD 中示例：



ST 中示例：

首先定义数组 P

VAR

...

CHARACTERISTIC_LINE:CHARCURVE;

KL:ARRAY[0..10] OF POINT:=[(X:=0,Y:=0),(X:=250,Y:=50),
(X:=500,Y:=150),(X:=750,Y:=400),7((X:=1000,Y:=1000))];

COUNTER:INT;

...

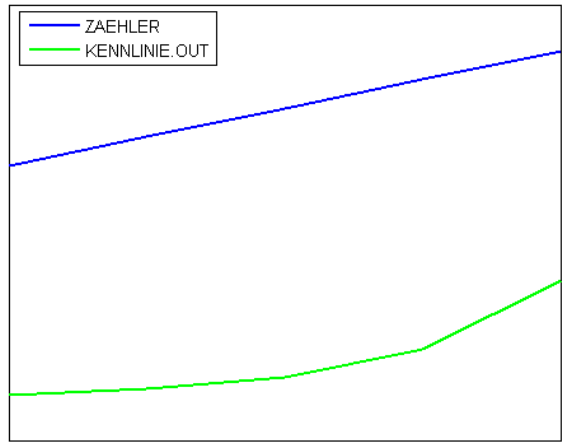
END_VAR

然后定义 CHARCURVE 一个增加值例如：

COUNTER:=COUNTER+10;

CHARACTERISTIC_LINE(IN:=COUNTER,N:=5,P:=KL);

跟踪显示效果：



1.2.6.2 RAMP_INT

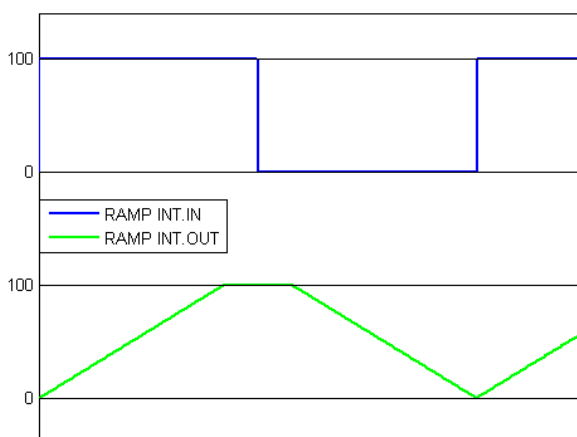
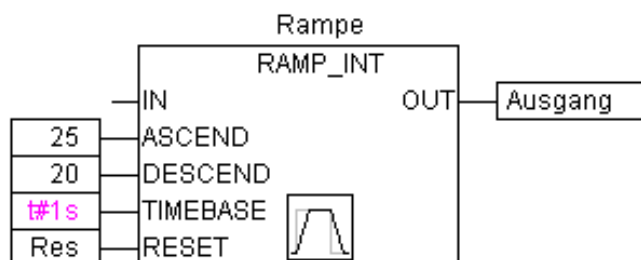
该功能块用于限制输入数值的上升率或下降率。

输入变量 IN, ASCEND 与 DESCEND 为 INT 类型: IN 为输入值; ASCEND 与 DESCEND 分别为给定时间内的最大增量值与减量值。TIMEBASE 为 TIME 类型, 用于设置给定时间。当 RESET 的数值为 TRUE 时, RAMP_INT 将重新初始化。

输出变量 OUT 为 INT 类型, 包含已被限制上升率和下降率的数值。

当 TIMEBASE 的数值为 t#0S 时, ASCEND 和 DESCEND 与时间间隔无关, 但保持相同。

CFC 中示例:



1.2.6.3 RAMP_REAL

RAMP_REAL 与 RAMP_INT 功能相似, 两者之间的区别是 RAMP_REAL 的输入 IN, ASCEND, DESCEND 和输出 OUT 是 REAL 类型。

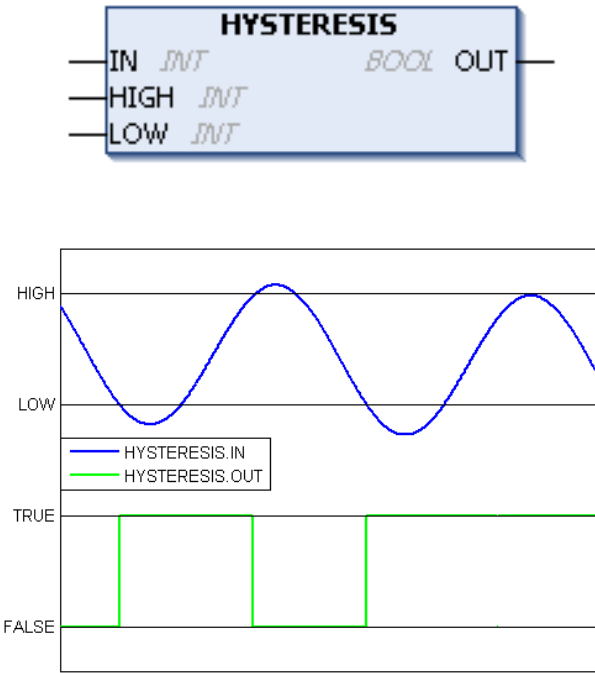
1.2.7 模拟值处理

1.2.7.1 HYSTERESIS

这个功能块的输入包含三个 INT 变量 IN, HIGH 和 LOW。输出 OUT 的类型为 BOOL。

如果 IN 低于下限值 LOW, OUT 的数值为 TRUE。如果 IN 高于上限值 HIGH, OUT 的值为 FALSE。

FBD 中示例:



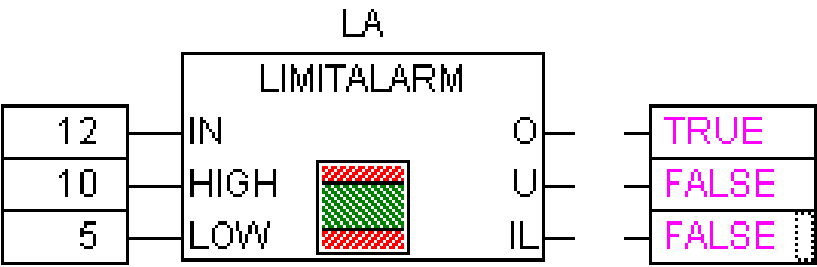
1.2.7.2 LIMITALARM

该功能块用于检查输入数值是否在所设置的范围内，超越了哪个极限。

输入变量 **IN**、**HIGH** 和 **LOW** 均为 **INT** 型。输出变量 **O**、**U** 与 **IL** 均为 **BOOL** 型。

如果 **IN** 到达输入上界 **HIGH**，**O** 将会被置为 **TRUE**，当 **IN** 位 **LOW** 之下时，**U** 将会被置为 **TRUE**。如果 **IN** 位于 **LOW** 和 **HIGH** 之间 **IL** 将会被置为 **TRUE**。

FBD 中示例：



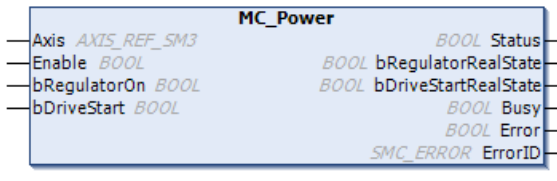
第2章 运动控制指令

2.1 SM3_Basic Library 库使用说明

2.1.1 MC_Power

MC_Power：将伺服驱动器切换为使能状态

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Power	轴使能		<pre>MC_Power(Axis:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
bRegulatorOn	执行条件	BOOL	TRUE, FALSE	FALSE	
bDriveStart	执行条件	BOOL	TRUE, FALSE	FALSE	

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Status	使能	BOOL	TRUE, FALSE	FALSE	进入使能状态时变为 TRUE
bRegulatorRealState		BOOL	TRUE, FALSE	FALSE	bRegulatorOn 置为 TRUE 后该位置为 TRUE
bDriveStartRealState		BOOL	TRUE, FALSE	FALSE	bDriveStart 置为 TRUE 后该位置为 TRUE
Busy	执行中	BOOL	TRUE, FALSE	FALSE	接受指令后变为 TRUE
Error	错误代码	BOOL	TRUE, FALSE	FALSE	发生异常时变为 TRUE
ErrorID	错误代码	SMC_ERROR			发生异常时，输出错误代码

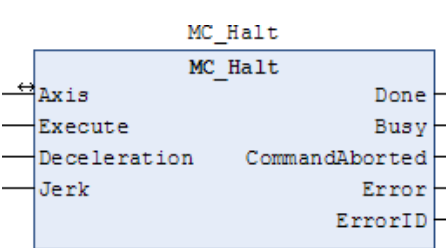
(3) 功能说明

将 **Enable**（有效）设为 **TRUE** 时，**Axis**（轴）指定的轴进入可运行状态。将轴设为可运行状态，可实现轴控制。将 **Enable**（有效）设为 **FALSE**，可解除 **Axis**（轴）指定轴的可运行状态。解除可运行状态后，轴不接收动作指令，无法实现轴控制。并且，解除可运行状态的轴相应的动作指令表现为异常。但是，即使是解除状态也可以执行 **MC_Power**（可运行）指令、**MC_Reset**（轴错误复位）指令。

2.1.2 MC_Halt

MC_Halt：停止指定轴的运动

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Halt	轴正常停止命令		<pre> MC_Halt(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Deceleration	减速度	LREAL	“正数” + “0”	0	功能块的减速度(μ/S2)
Jerk	执行条件	LREAL	“正数” + “0”	0	指定跃度 [指令单位/S3]

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

本功能块启动可以停止一个轴的运动，而在这操作后，其它一个运动轴指令再启动运行时可以终止功能块的运行。

本功能块在轴为运行状态下才能运行，其他状态下启动指令无效。

本功能块在输入变量执行条件的上升沿启动。

本功能块执行过程中轴的状态为离散运动状态，功能运行完成后轴状态变为停机静止状态。

2.1.3 MC_Home

MC_Home：确定轴的零点位置

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Home	轴回零指令		<pre> MC_Home(Axis:= , Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Position	轴到达位置	LREAL	数据范围	0	代表轴的回零位置

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

本功能块为回零操作，启动指令里输入变量执行条件的上升沿触发，输入变量轴到达的位置为零点位置。本功能块只有在停机静止状态才能运行，且在执行指令前需要设置伺服回零模式，指令运行时运动过程中轴的状态为回零状态。

回零模式设置方法主要有两种：

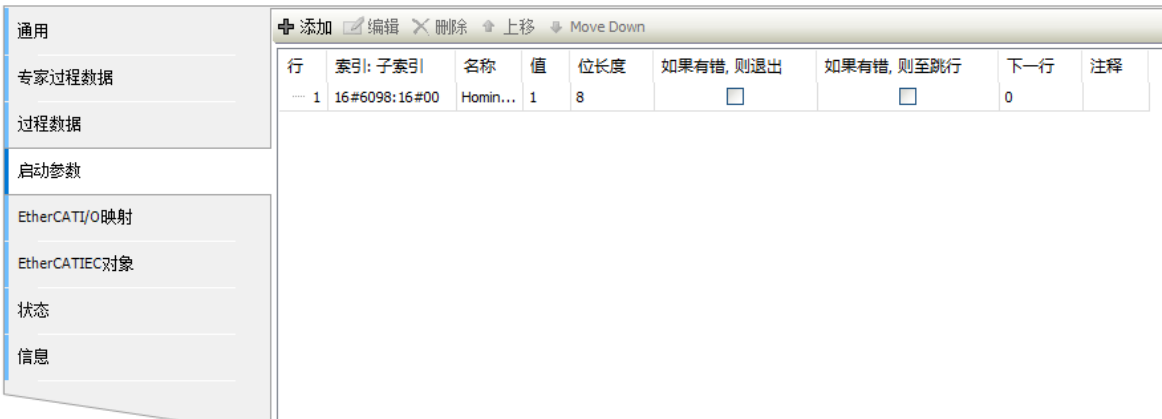
①通过手动设置伺服功能码设置回零模式。

在英威腾伺服 DA200 设置 P5.10 来设置回零模式。

②通过 AX 系列从站的启动参数也可以设置相应的回零模式，采用通讯的方式必须设置如下索引和子索引的数据。

项目	索引	子索引	描述
回零方式	0x6098		根据伺服手册可以选择设定的具体参数
找原点速度	0x6099	0x01	一般定义的速度相对比较高，减少归零时间
找零点速度	0x6099	0x02	一般定义的速度相对较低
原点回归加减速	0x609A		在原点回归时的加减速变化
原点回归超时时间	0x2005	0x24	回归时间超过设定时间，系统报 Err.601 错误

AX 系列相应的设置界面参考如下：



2.1.4 MC_MoveAbsolute

MC_MoveAbsolute：指定绝对坐标的目标位置，进行定位

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_ MoveAbsolute	轴绝对位置 控制指令		<pre>MC_MoveAbsolute(Axis:= , Execute:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

（2） 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Position	轴到达位置	LREAL	数据范围	0	代表轴的绝对位置数据
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
Direction	指令极性	MC_ DIRECTION	Negative, Shortest, Positive, current, fastest	shortest	Negative: 反向移动; Shortest: 根据最短路径选择方向; Positive: 正向移动; Current: 按当前方向移动; Fastest: 自动选择最快的方向移动; (此功能轴在旋转模式下有效)

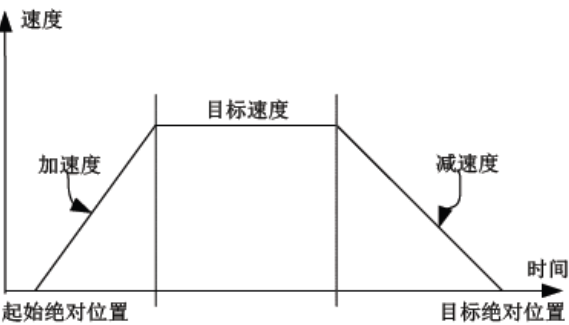
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

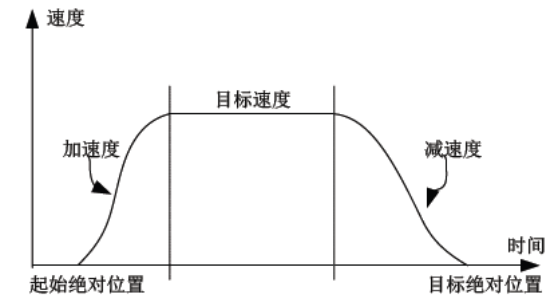
(3) 功能说明

本功能块为轴绝对定位指令，position 为轴的绝对位置。本功能块运行状态为 Standstill 中，当启动指令 Execute 上升沿触发后，运行状态切换为 Discrete Motion，并运动到指定 position 位置处。当 Jerk 为 0 时，轴做梯形加减速运动；当 Velocity、Acceleration、Deceleration 和 Jerk 有数据时，则做 S 型曲线加减速运动。

◇ 梯形加减速动作

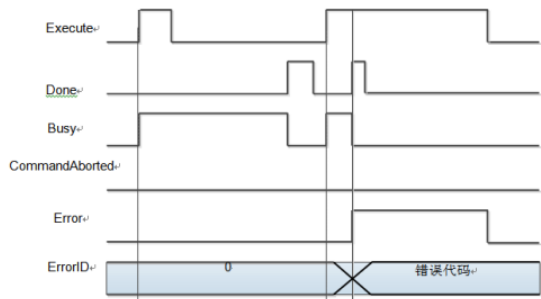


◇ S 曲线加减速动作



(4) 时序图

- ◇ 轴必须处于 Standstill 状态；
- ◇ 功能块的 Execute 上升沿触发；
- ◇ 功能块 Done 为 TRUE 时表示执行完成，否则 Busy 为 TRUE。



2.1.5 MC_AccelerationProfile

MC_AccelerationProfile：时间段和加减速度的轮廓运动模型

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_AccelerationProfile	加速度轮廓指令		<pre>MC_AccelerationProfile(Axis:= , TimeAcceleration:= , Execute:= , ArraySize:= , AccelerationScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
TimeAcceleration	轴加速时间和加速度描述	MC_TA_REF			轴加速时间和加速度数据描述，加速数据由多组数据组成

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
ArraySize	动态数组	INT	数据范围	0	运行轮廓中使用的数组个数
AccelerationScale	综合因子	LREAL	“正数”“+”“0”	1	MC_TA_REF 中加速度或减速度的比例因子
Offset	偏移	LREAL		0	加减速度的总体偏移值

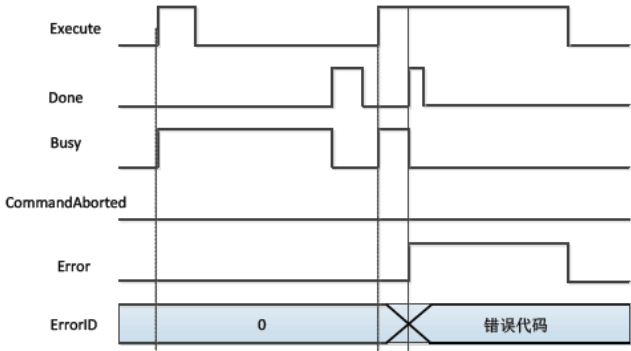
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

（3） 功能说明

本功能块为时间段和加减速度的轮廓运动模型，运行模式为 Discrete Motion，按用户在 TimeAcceleration 变量中设定的数据运行。本功能块运行状态为 Standstill 中，指令运行时的状态为 Discrete Motion，其他状态无法运行。启动指令为 Execute 的上升沿启动，本指令在 Discrete Motion 运行速度都是在上一轮的叠加，容易引起系统故障。

（4） 时序图



2.1.6 MC_MoveAdditive

MC_MoveAdditive: 轴在原来指令位置上再叠加 Distance 的距离，进行定位。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_MoveAdditive	叠加绝对运动指令		<pre> MC_MoveAdditive(Axis:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Distance	轴到达位置	LREAL	数据范围	0	轴的叠加位置数据
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

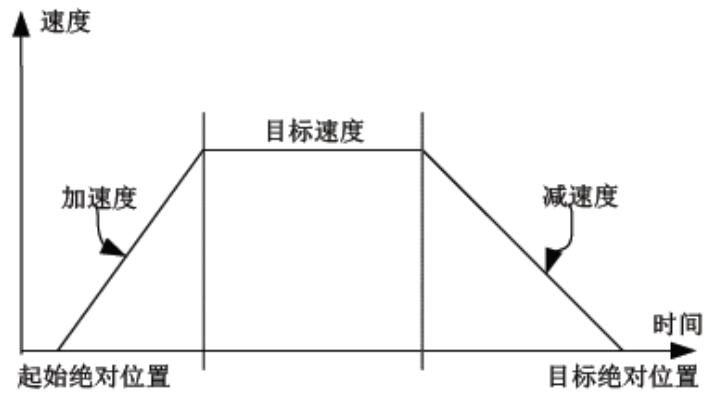
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

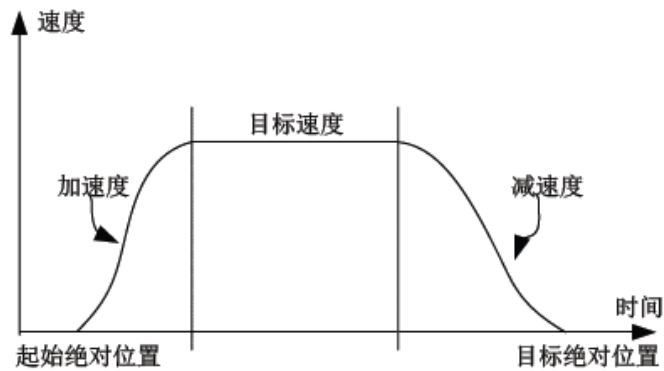
（3）功能说明

启动指令为 **Execute**，上升沿触发功能块，**Distance** 数据为轴的叠加数据。本功能块运行状态如果为 **Discrete Motion**，使用情况会把其他指令的 **CommandAbort** 置位；在 **standstill** 状态，本指令能独立运行，实现相对定位需求；**Acceleration** 或 **Deceleration** 为零，指令运行都为异常状态，但轴的状态为 **Discrete Motion**；当 **Jerk** 为 0 时，轴做梯形加减速运动；当 **Velocity**、**Acceleration**、**Deceleration** 和 **Jerk** 有数据时，则做 S 型曲线加减速运动。

✧ 梯形加减速动作

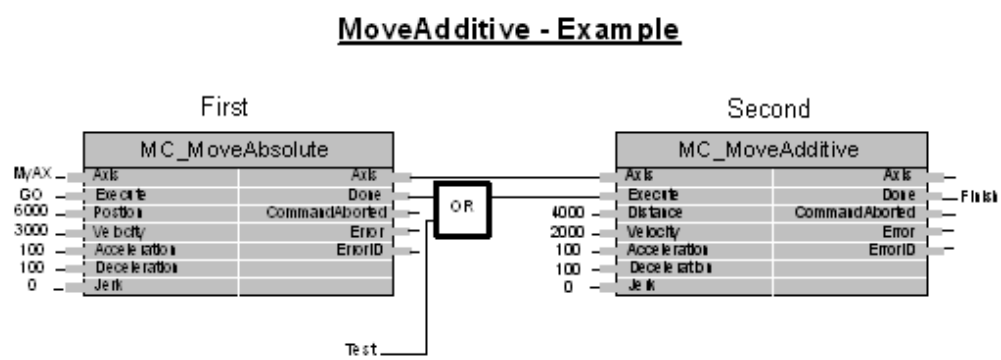


✧ S 曲线加减速动作

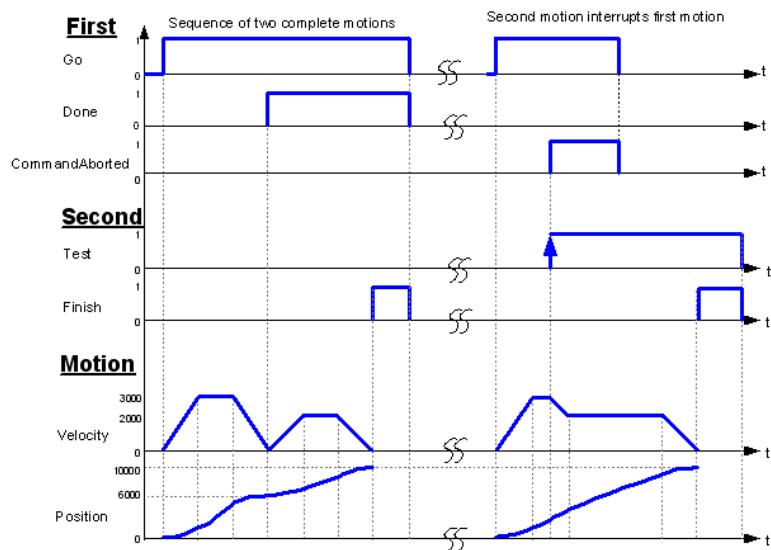


（4）时序图

✧ 举例说明



◇ 时序操作说明



2.1.7 MC_MoveRelative

MC_Move Relative: 指定自指令当前位置起的移动距离，进行定位

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_MoveRelative	轴相对定位指令		<pre>MC_MoveRelative(Axis:=, Execute:=, Distance:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Distance	运动相对位置	LREAL	数据范围	0	此数据为运动的相对位置
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值

输入变量	名称	数据类型	有效范围	初始值	描述
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

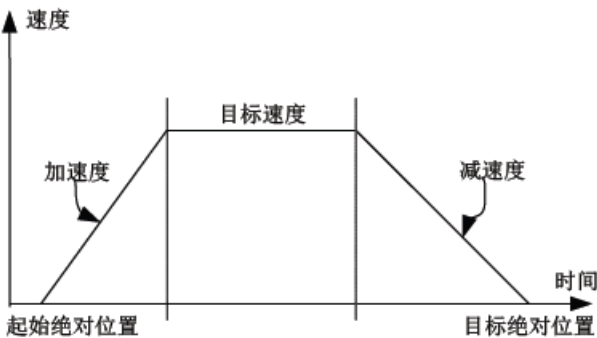
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

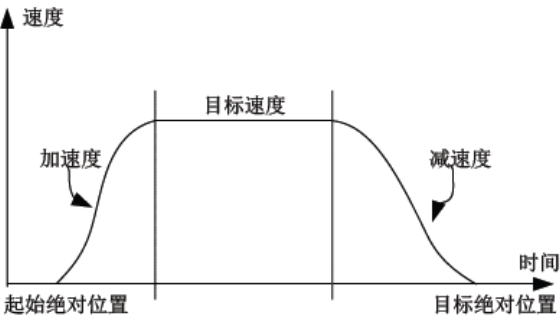
（3）功能说明

本功能块运行状态为 Standstill 中，指令运行时的状态为 Discrete Motion，在指令执行中关注本轴的运行状态，避免打断本轴的其他指令或被其他指令打断本轴的执行。启动指令为 Execute 的上升沿启动，本指令在 Discrete Motion 可以重复上升沿有效，每次都可以刷新最新的 Position 位置。Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion；

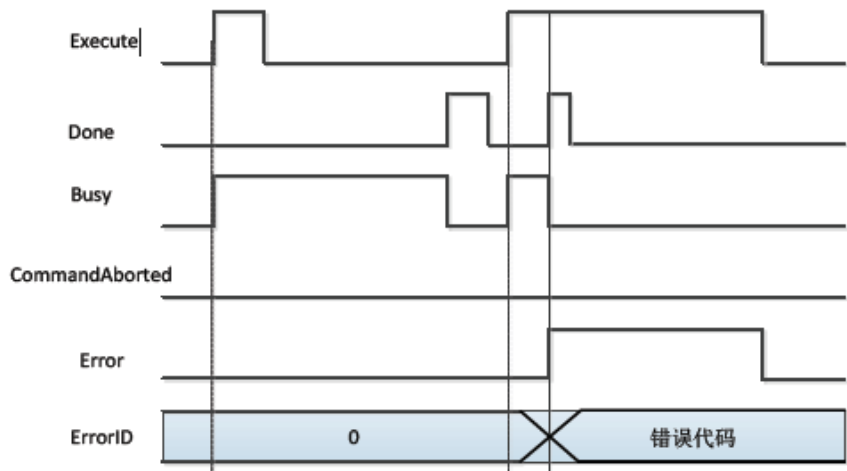
◇ 梯形加减速动作



◇ S 曲线加减速动作



（4） 时序图



Execute 上升沿触发功能块，Busy 置位表示功能块执行中，执行完成后，置位 Done。

2.1.8 MC_MoveSuperImposed

MC_MoveSuperImposed: 轴在原来指令速度和位置的基础上叠加速度和位置的数据在运行的指令上，对整个原来的指令执行时间模型上没有变化

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_MoveSuperImposed	叠加 相对 运动 指令		<pre>MC_MoveSuperImposed(Axis:= , Execute:= , Distance:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

（2） 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Distance	运动相对位置	LREAL	数据范围	0	此数据为相对位置
VelocityDiff	叠加速度	LREAL	数据范围	0	轴运行叠加速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值

输入变量	名称	数据类型	有效范围	初始值	描述
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

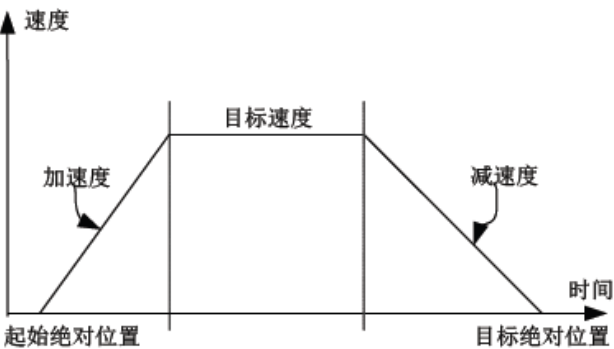
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

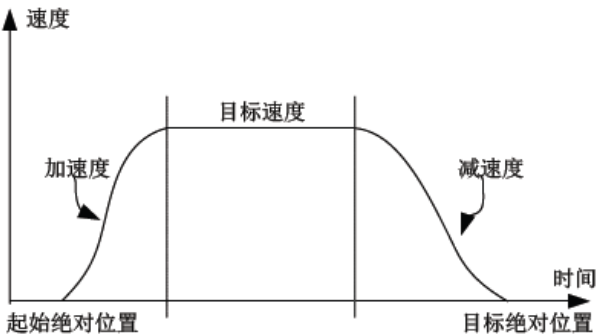
（3）功能说明

本功能块为叠加位置和速度指令，Execute 上升沿触发，VelocityDiff 和 Distance 分别为叠加在其他指令上的速度和位置；在运动模式下 MC_MoveSuperImposed 可以叠加在任何其他指令；通过此指令能解决我们实际运行中由皮带和齿轮间隙误差补偿，能保证运动的一致性；指令运行时需要设置参数叠加位置。

◇ 梯形加减速动作

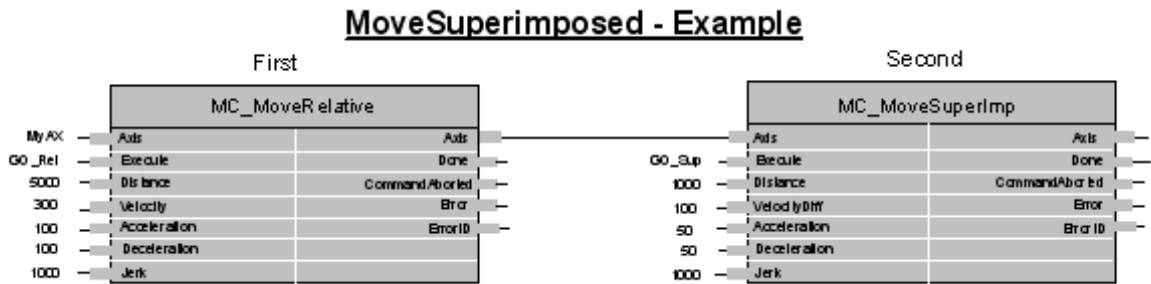


◇ S 曲线加减速动作

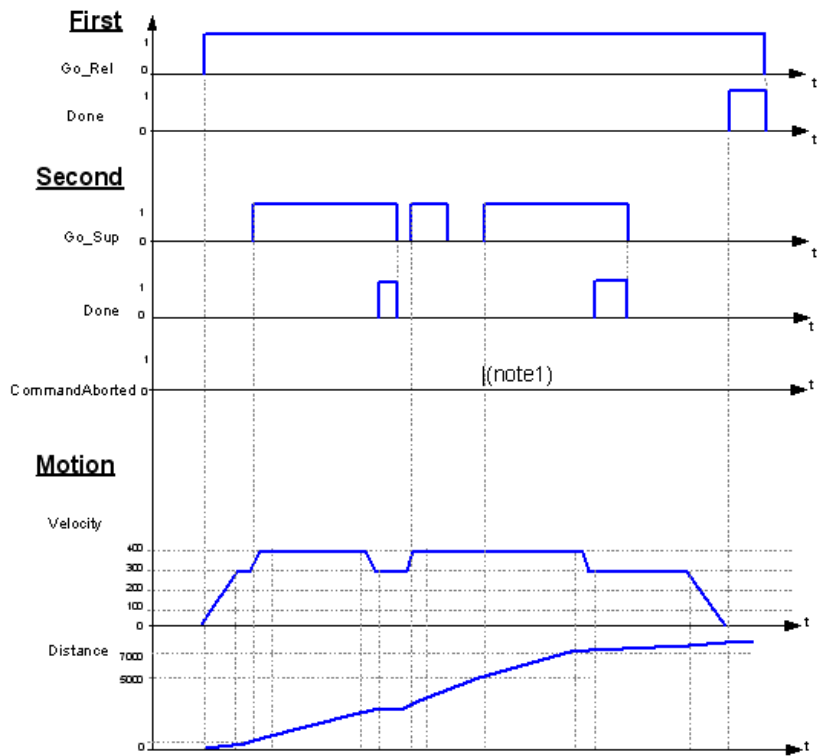


（4） 时序图

◇ 举例说明



◇ 时序操作说明



2.1.9 MC_MoveVelocity

MC_MoveVelocity: 使用伺服驱动器的位置控制模式，进行模拟速度控制

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_MoveVelocity	速度控制指令	MC_MoveVelocity - Axis <i>AXIS_REF_SM3</i> - Execute <i>BOOL</i> - Velocity <i>LREAL</i> - Acceleration <i>LREAL</i> - Deceleration <i>LREAL</i> - Jerk <i>LREAL</i> - Direction <i>MC_Direction</i>	<pre>MC_MoveVelocity(Axis:=, Execute:=, Velocity:=, Acceleration:=, Deceleration:=, Jerk:=, Direction:=, InVelocity=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴,即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理
Velocity	运行速度	LREAL	数据范围	0	指定目标速度运行
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
Direction	运行方向	MC_Direction	positive, negative, current	current	为运行方向的指令操作

✧ 输出变量

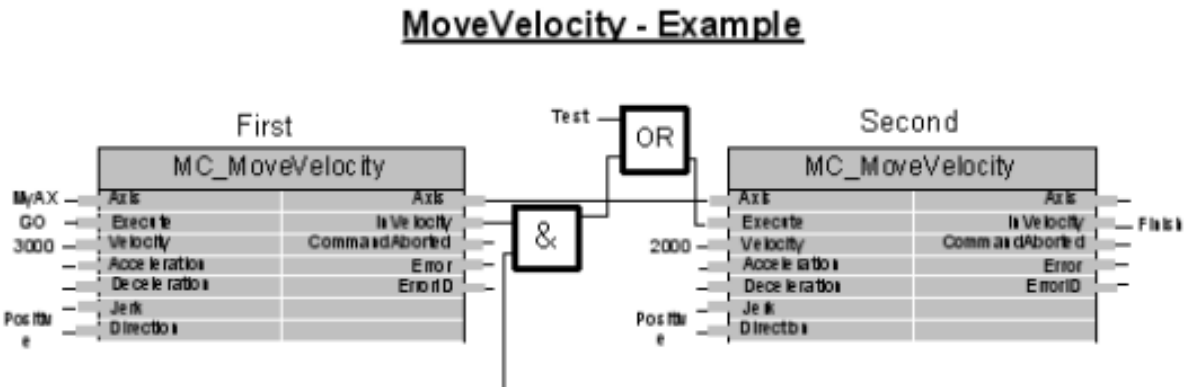
输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

(3) 功能说明

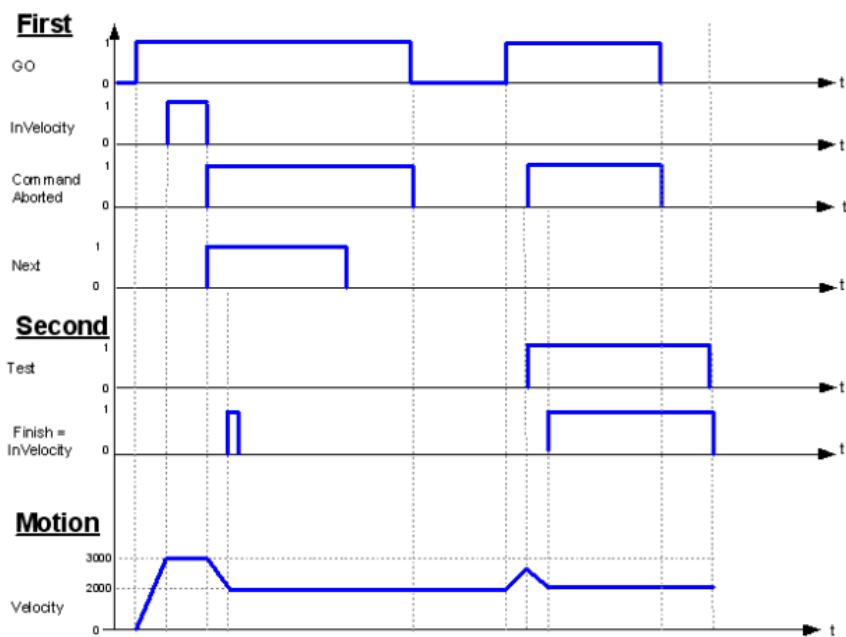
Eexecute 上升沿触发该功能块, 使驱动器按照 Velocity 值进行速度控制, 功能块的 InVelocity 表示指令的运行速度达到设定值。

(4) 时序图

✧ 举例说明



◇ 时序操作说明



2.1.10 MC_PositionProfile

MC_PositionProfile: 时间段和位置的轮廓运动模型

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_PositionProfile	位置 轮廓 指令		<pre>MC_PositionProfile(Axis:=, TimePosition:=, Execute:=, ArraySize:=, PositionScale:=, Offset:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
TimePosition	轴位置运行时间和位置描述	MC_TP_REF			轴位置运行时间和位置数据描述，数据由多组数据组成

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理

输入变量	名称	数据类型	有效范围	初始值	描述
ArraySize	动态数	INT	数据范围	0	运行轮廓中使用的数组个数
PositionScale	综合因子	LREAL	“正数” + “0”	0	MC_TP_REF 中位置的比例因子
Offset	偏移	LREAL		0	位置的总体偏移值

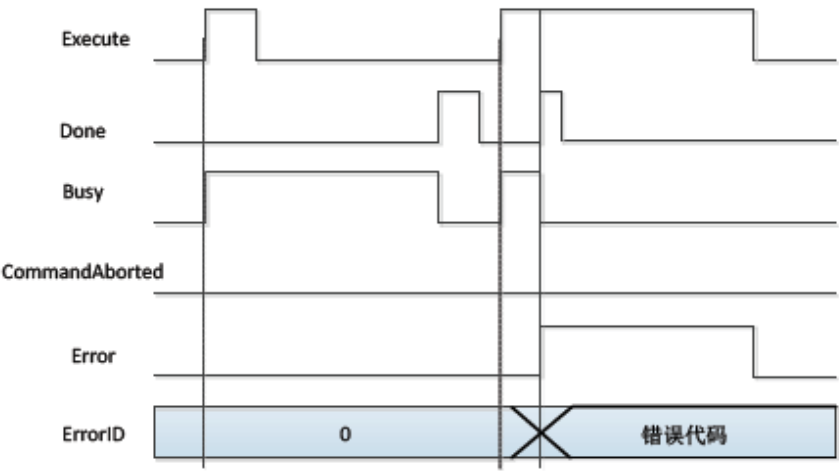
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

（3）功能说明

本功能块为时间段和位置的轮廓运动模型，按用户在 TimePosition 变量中设定的数据运行。本功能块运行在状态 Stanstill 中，Execute 上升沿触发功能块运行，功能块执行过程中状态为 Discrete Motion。

（4）时序图



2.1.11 MC_ReadActualPosition

MC_ReadActualPosition：读取驱动器运行的实际位置，并保存在自定义变量中。

（1）指令格式

指令	名称	图形表现	ST 表现
MC_ReadActualPosition	实际位置读取指令		<pre>MC_ReadActualPosition(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	输入的一个上升沿将启动功能块的处理

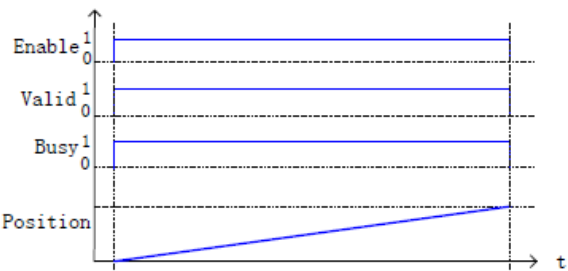
✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	位置数据可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码
Position	获取到的轴位置	LREAL	轴位置	0	指令读出来的轴位置数据

(3) 功能说明

Eexecute 上升沿触发该功能块，通过本指令可读取轴的位置值；Valid 为 TRUE 时表示读取的位置值是有效值；该功能块可重复调用，互不影响。

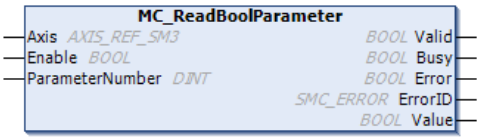
(4) 时序图



2.1.12 MC_ReadBoolParameter

MC_ReadBoolParameter: 指令读取驱动轴的位参数，并保存在自定义变量中。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadBoolParameter	读取轴的位参数		<pre> MC_ReadBoolParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=>, Busy=>, Error=>, ErrorID=>, Value=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理
ParameterNumber	轴参数的序号	DINT		0	访问轴参数的索引和子索引和序号

注:

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DWORD(usiDataLength), 24) (对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (对象字典中的索引-16BIT)

+ usisubIndex(对象字典中的子索引-8BIT))

usiDataLength: 按字节数填写; 1 字节为 16#01; 2 字节为 16#02; 4 字节为 16#04 等。

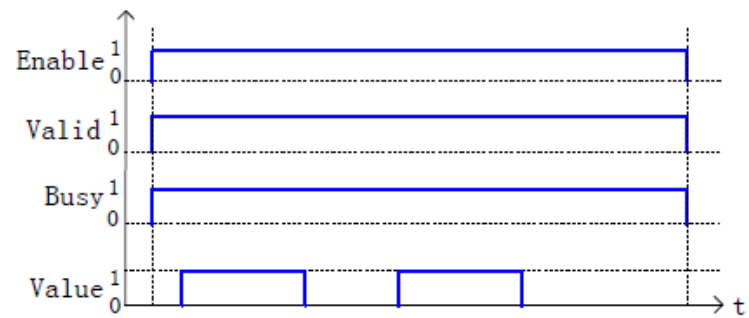
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	位置数据可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的位置, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
Value	获取到的轴位状态	BOOL	TRUE, FALSE	FALSE	读出轴为错误码

(3) 功能说明

通过 MC_ReadBoolParam 读取驱动器中的位数据状态, 指令为 Enable 电平使能有效。指令可以重复多次使用, 互不影响; 功能块的 Valid: =TRUE 表示读出值为有效的位状态数据; 功能块的 Busy: =TRUE 表示当前功能块正在执行中。

(4) 时序图



2.1.13 MC_ReadAxisError

MC_ReadAxisError: 读取轴的错误信息，并保存在自定义变量中。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadAxisError	读取轴的错误信息		<pre>MC_ReadAxisError(Axis:= , Enable:= , Valid=>, Busy=>, Error=>, ErrorID=>, AxisError=>, AxisErrorID=>, SWEndSwitchActive=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	错误数据可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	TRUE, FALSE	FALSE	发生异常时，输出错误代码
AxisError	轴错误标示	BOOL	TRUE, FALSE	FALSE	读出轴为错误，对应的标示置位
AxisErrorID	轴错误码	DWORD		0	读出轴的错误码
SWEndSwitchActive	软限位开关有效	BOOL	TRUE, FALSE	FALSE	在指令读取中，检查软限位开关状态

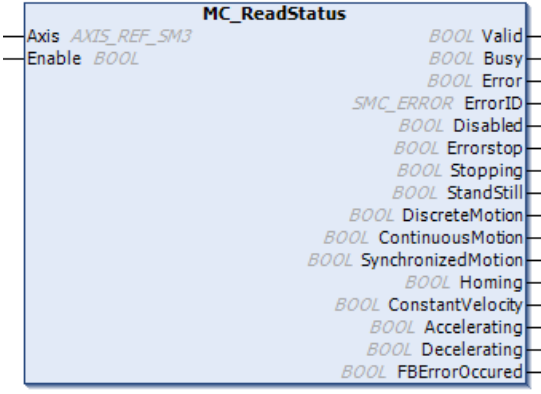
（3）功能说明

通过该功能块读取轴的错误信息，Enable 为 TRUE 有效；功能块的 Valid 表示读出的 AxisError 和 AxisErrorID 为有效的数据值；功能块的 Busy 表示当前功能块正在执行中；指令可重复调用互不影响。

2.1.14 MC_ReadStatus

MC_ReadStatus：指令读取轴的状态数据，并保存在自定义变量中。

（1）指令格式

指令	名称	图形表现	ST 表现
MC_ReadStatus	读取轴的状态		<pre>MC_ReadStatus(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Disabled=> , Errorstop=> , Stopping=> , StandStill=> , DiscreteMotion=> , ContinuousMotion=> , SynchronizedMotion=> , Homing=> , ConstantVelocity=> , Accelerating=> , Decelerating=> , FBErrorOccured=>);</pre>

（2）相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	错误数据可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	TRUE, FALSE	FALSE	发生异常时，输出错误代码
Disabled	轴未使能	BOOL	TRUE, FALSE	FALSE	轴在没使能状态为 TRUE
Errorstop	轴错误状态	BOOL	TRUE, FALSE	FALSE	轴在错误运行状态为 TRUE
Stoping	轴停止过程状态	BOOL	TRUE, FALSE	FALSE	轴在停止过程中为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
StandStill	轴标准状态	BOOL	TRUE, FALSE	FALSE	轴在标准(能运行)状态中为 TRUE
DiscreteMotion	轴离散运动状态	BOOL	TRUE, FALSE	FALSE	轴在离散运动状态中为 TRUE
ContinuousMotion	轴连续运动状态	BOOL	TRUE, FALSE	FALSE	轴在连续运动状态中为 TRUE
SynchronizedMotion	轴同步运行状态	BOOL	TRUE, FALSE	FALSE	轴在同步运动状态中为 TRUE
Homing	轴回原点状态	BOOL	TRUE, FALSE	FALSE	轴在回原点状态中为 TRUE
ConstantVelocity	轴运行速度到达	BOOL	TRUE, FALSE	FALSE	轴达到运行速度中为 TRUE
Accelerating	轴加速过程状态	BOOL	TRUE, FALSE	FALSE	轴加速过程状态为 TRUE
Dccelerating	轴减速过程状态	BOOL	TRUE, FALSE	FALSE	轴减速过程状态为 TRUE
FBErrorOccured	轴功能块错误出现标志	BOOL	TRUE, FALSE	FALSE	轴功能块错误标志为 TRUE

(3) 功能说明

通过 MC_ReadStatus 对应轴的各种状态, 指令为 Enable 为 TRUE 使能效应。指令可以重复多次使用, 互不影响; 功能块的 Valid 表示读出的后面各状态标志的各种数据; 功能块的 Busy 表示当前功能块正在执行中。

2.1.15 MC_ReadParameter

MC_ReadParameter: 指令读取驱动轴的参数, 并保存在自定义变量中。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadParameter	读取轴的参数		<pre> MC_ReadParameter(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>); </pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理
ParameterNumber	轴参数的序号	DINT		0	访问轴参数的索引和子索引和序号

注：

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24)(对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (对象字典中的索引-16BIT)

+ usisubIndex(对象字典中的子索引-8BIT))

usiDataLength: 按字节数填写；1 字节为 16#01；2 字节为 16#02；4 字节为 16#04 等。

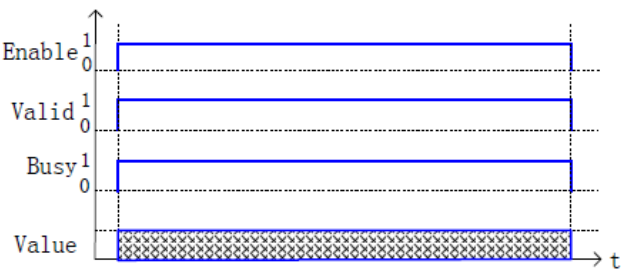
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	位置数据可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的位置，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码
Value	获取到的轴位状态	BOOL	TRUE, FALSE	FALSE	读出轴为错误码

（3）功能说明

通过 MC_ReadBoolParameter 读取驱动器中的位数据状态，Enable 高电平有效。指令可以重复多次使用，互不影响；功能块的 Valid 表示读出的 Valid 为有效的位状态数据；功能块的 Busy 表示当前功能块正在执行中。

（4）时序图



2.1.16 MC_Reset

MC_Reset: 指令解除轴的异常。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Reset	轴错误状态复位指令		<pre>MC_Reset(Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿将启动功能块的处理

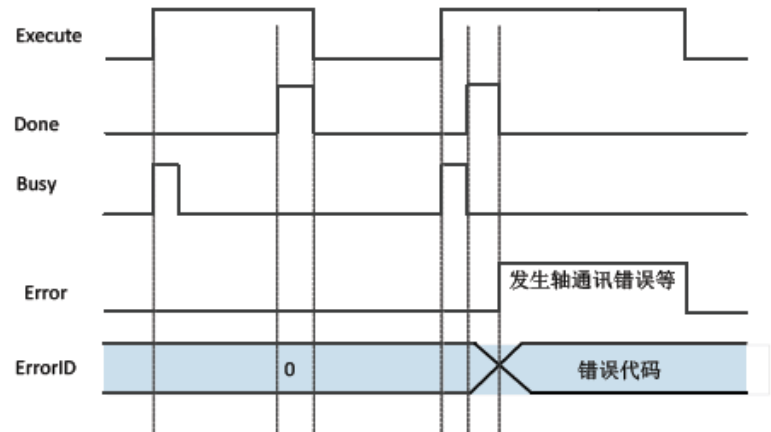
✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

本功能块在轴通讯正常的情况下，把轴状态处于 errorstop 变为 Standstill，把轴的异常状态变为正常可运行的状态。如果遇到轴 errorstop 无法复位,Axis.bCommunication 为 FLASE 状态，必须要重新建立主站和从站轴的通讯。

(4) 时序图



2.1.17 MC_Stop

MC_Stop: 指令使轴减速停止。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Stop	轴停止命令		<pre> MC_Stop(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿将启动功能块的处理
Deceleration	减速度	LREAL	“正数”+“0”	0	功能块的减速度(u/S2)
Jerk	跃度	LREAL	“正数”+“0”	0	指定跃度 [指令单位/S3]

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

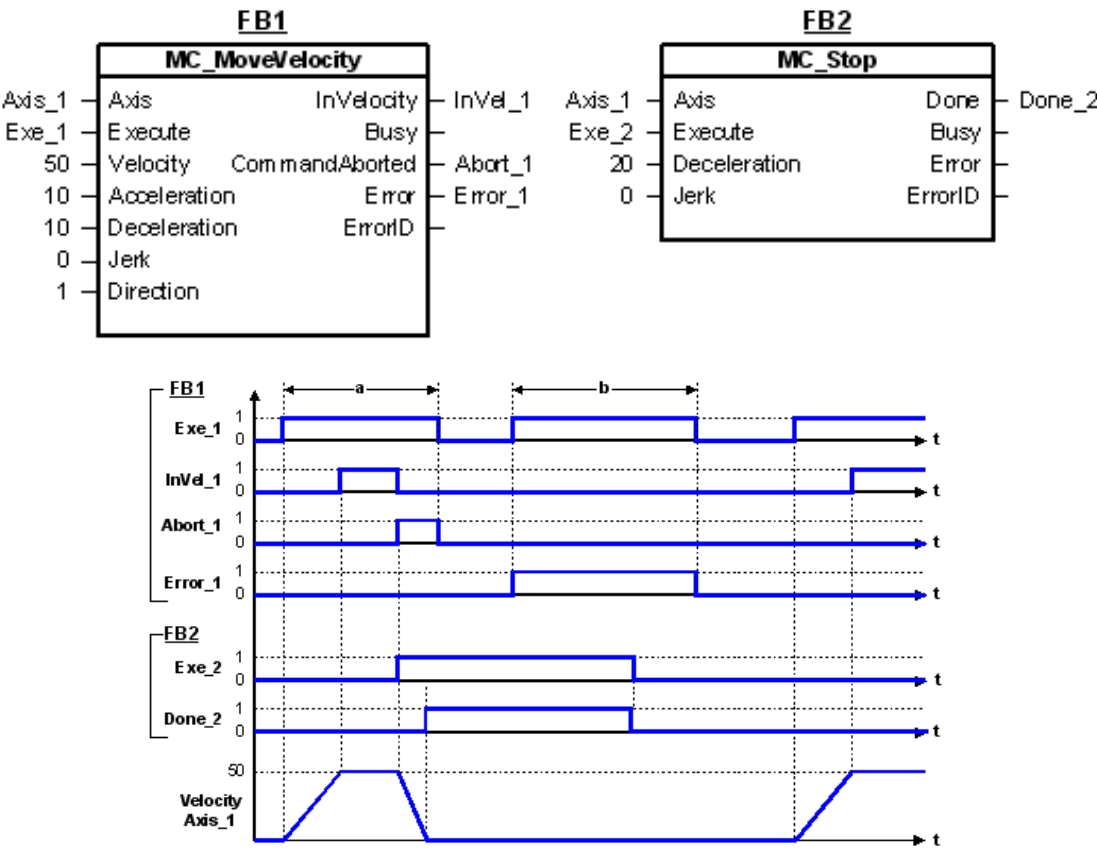
本功能块为在正常运行的情况下停止一个轴的运动，当轴处于 stopping 状态对本轴的任何指令都是无效的。

当轴状态处于 stopping 时，Execute 为 Flase 状态，Done 输出状态为 True，轴状态变为 Standstill；本功能块运行状态为运行状态(Motion) 才能运行，其他状态无法运行。本功能块在上升沿启动。在 MC_Stop 有效执行的过程 Busy 有效时，再一次启动 MC_Stop 指令系统处于 Errorstop 状态。

(4) 时序图

✧ 时序操作例程如下

例程：在执行 MC_MoveVelocity 指令和 MC_Stop 指令在不同的时序操作中对应的标志位的变化；



2.1.18 MC_VelocityProfile

MC_VelocityProfile：本功能块为时间段和速度的轮廓运动模型。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_VelocityProfile	速度轮廓指令		<pre>MC_VelocityProfile(Axis:=, TimeVelocity:=, Execute:=, ArraySize:=, VelocityScale:=, Offset:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
TimeVelocity	轴速度运行时间和速度描述	MC_TV_REF			轴速度运行时间和速度数据描述，由多组数据组成

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿将启动功能块的处理
ArraySize	动态数组	INT		0	运行轮廓中使用的数组个数
VelocityScale	速度因子	LREAL	“正数”“+”“0”	1	速度的比例因子
Offset	偏移	LREAL		0	速度值的总体偏移值

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	轴指令执行完成，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

本功能块为时间段和速度的轮廓运动模型，轴的运行状态为 Continuous Motion，按用户在 TimeVelocity 变量中设定的数据运行。本功能块轴的运行状态为 Standstill 中，指令运行时轴的状态为 Discrete Motion，其他状态无法运行。启动指令为 Execute 的上升沿启动，本指令在 Discrete Motion 重复运行。TimeVelocity 为 MC_TV_REF 数据类型。

MC_TV_REF 具体描述如下：

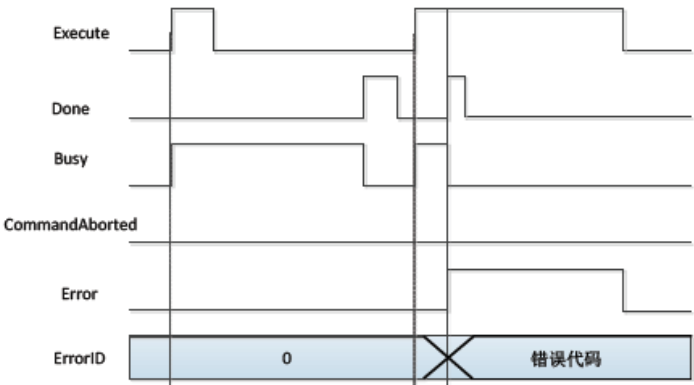
成员	类型	初始值	描述
Number_of_pairs	INT	0	轮廓路径的段数
IsAbsolute	BOOL	TRUE	绝对运动(TRUE) 和相对运动选择
MC_TV_Array	ARRAY[1..N] OF SMC_TV		时间和速度的数组

SMC_TV 具体描述如下：

成员	类型	初始值	描述
delta_time	TIME	TIME#0ms	速度值段的时间
Velocity	LREAL	0	当前记录的速度值

注：整个速度的过程为 S 曲线加减速的方式，每段轮廓的速度都为叠加的计算方式；在指令重复运行时，速度也为叠加方式，在指令使用时避免速度超限的出现；重复运行一定把本轴的状态重回到 Standstill 状态。

(4) 时序图



2.1.19 MC_WriteBoolParameter

MC_WriteBoolParameter: 指令设置驱动轴的位参数。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_WriteBoolParameter	设置轴的位参数		<pre>MC_WriteBoolParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理
ParameterNumber	轴参数的序号	DINT		0	访问轴参数的索引和子索引和序号
Value	设定值	BOOL	TRUE, FALSE	FALSE	设置位参数值

注:

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DWORD(usiDataLength), 24)) (对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (对象字典中的索引-16BIT)

+ usisubIndex(对象字典中的子索引-8BIT))

usiDataLength: 按字节数填写; 1 字节为 16#01; 2 字节为 16#02; 4 字节为 16#04 等。

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE, FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

(3) 功能说明

通过 MC_WriteBoolParameter 设置轴的位参数, 指令为 Execute 上升沿触发。指令可以重复多次使用, 互不影响。

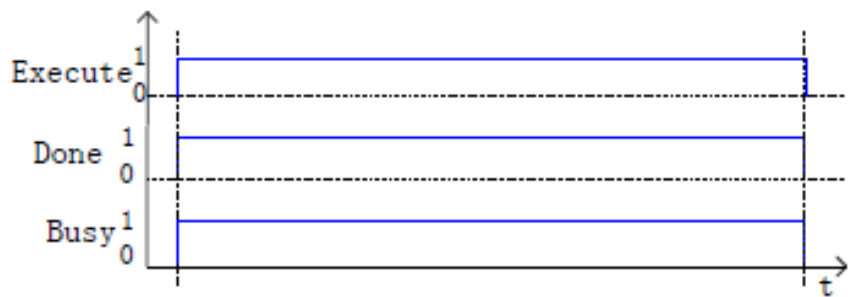
（4） 时序图

功能块的 **Execute** 必须为上升沿触发条件；

功能块的 **Done** 表示设置操作成功；

功能块的 **Busy** 表示当前功能块正在执行中；

✧ 时序操作说明：



2.1.20 MC_WriteParameter

MC_WriteParameter：指令设置驱动轴的参数。

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_WriteParameter	设置轴的参数		<pre>MC_WriteParameter(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

（2） 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理
ParameterNumber	轴参数的序号	DINT		0	访问轴参数的索引和子索引和序号
Value	设定值	BOOL	TRUE, FALSE	FALSE	设置位参数值

注：

ParameterNumber (DINT) = -DWORD_TO_DINT(SHL(USINT_TO_DOWRD(usiDataLength), 24))(对象字典中的数据长度)

+ SHL(UINT_TO_DWORD(uiIndex), 8) (对象字典中的索引-16BIT)

+ usisubIndex(对象字典中的子索引-8BIT))

usiDataLength: 按字节数填写；1 字节为 16#01；2 字节为 16#02；4 字节为 16#04 等。

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE, FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

（3）功能说明

通过 MC_WriteParameter 设置轴的位参数，指令为 Execute 上升沿触发。指令可以重复多次使用，互不影响。

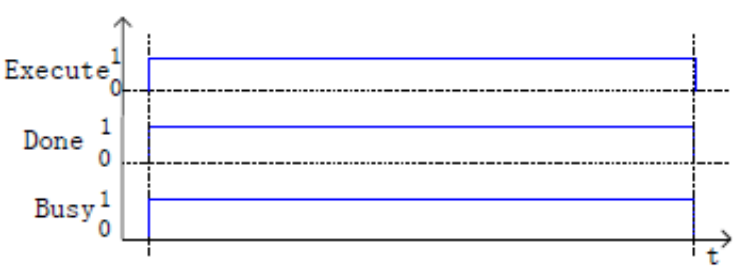
（4）时序图

功能块的 Execute 必须为上升沿触发条件；

功能块的 Done 表示设置操作成功；

功能块的 Busy 表示当前功能块正在执行中；

◇ 时序操作说明：



2.1.21 MC_AbortTrigger

MC_AbortTrigger: 功能块终止输入锁存相关事件的关联特性，和 MC_Touchprobe 配套使用。

（1）指令格式

指令	名称	图形表现	ST 表现
MC_AbortTrigger	功能块终止事件关联		<pre>MC_AbortTrigger(Axis:= , TriggerInput:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例
TruggerInput	触发信号	TRIIGGER_REF			触发信号、触发属性等描述

◇ TRIIGGER_REF 说明

输入输出变量	名称	数据类型	初始值	描述
TRIIGGER_REF	iTriggerNumber	INT	--1	在驱动器模式下, 锁定功能中的哪一个。 0: 探针 1 上升沿锁存 1: 探针 1 下降沿锁存 2: 探针 2 上升沿锁存 3: 探针 2 下降沿锁存
	bFastLatching	BOOL	TRUE	指定锁存触发的模式: TRUE: 驱动器模式 FALSE: 控制器模式
	bInput	BOOL		当 bFastLatching=FLASE 时, 由控制器 Input 信号触发
	bActive	BOOL		触发的有效信号

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	设置操作成功	BOOL	TRUE, FALSE	FALSE	设置操作成功置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

(3) 功能说明

通过 MC_AbortTrigger 功能块是把触发信号或属性和相关的触发指令终止其关联操作。功能块的 Execute 必须为上升沿触发条件; 功能块的 Done 表示设置操作成功; 功能块的 Busy 表示当前功能块正在执行中。

2.1.22 MC_ReadActualTorque

MC_ReadActualTorque: 指令读取驱动器运行的当前力矩值, 读取的当前力矩值保存在自己定义的变量单元中。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadActualTorque	当前力矩值读取指令		<pre> MC_ReadActualTorque(Axis:=, Enable:=, Valid=>, Busy=>, Error=>, ErrorID=>, Torque=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	置为 TRUE 将启动功能块的处理

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	当前力矩值可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的力矩值, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR	-	0	发生异常时, 输出错误代码
Torque	获取的当前力矩值	LREAL	力矩值	0	指令读出来的当前力矩数据

(3) 功能说明

通过 MC_ReadActualTorque 读取驱动器中的当前力矩值指令, 指令为 Enable 电平使能效应。指令可以重复多次使用, 互不影响。

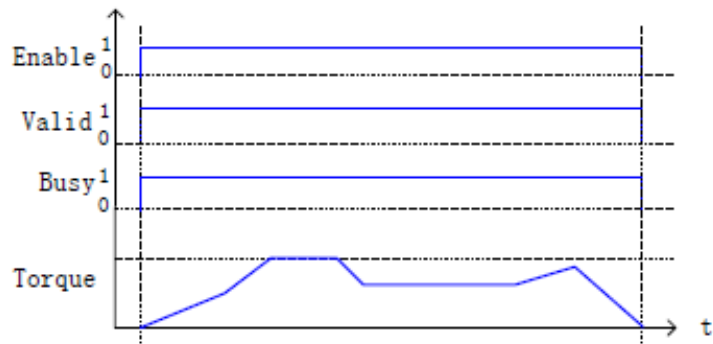
(4) 时序图

功能块的 Enable 必须为 TRUE 的条件;

功能块的 Valid 表示读出的 Torque 为有效的数据值;

功能块的 Busy 表示当前功能块正在执行中。

◇ 时序操作说明



2.1.23 MC_ReadActualVelocity

MC_ReadActualVelocity: 指令读取驱动器运行的当前速度值，读取的当前力矩值保存在自己定义的变量单元中。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadActualVelocity	当前速度读取指令		<pre>MC_ReadActualVelocity(Axis:= , Enable:= , Valid=>, Busy=>, Error=>, ErrorID=>, Velocity=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE,FALSE	FALSE	置为 TRUE 将启动功能块的处理

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	当前力矩值可获取标志	BOOL	TRUE, FALSE	FALSE	能正确的获取驱动器的力矩值，置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	-	0	发生异常时，输出错误代码
Velocity	获取的当前速度值	LREAL	速度值	0	指令读出来的当前速度数据

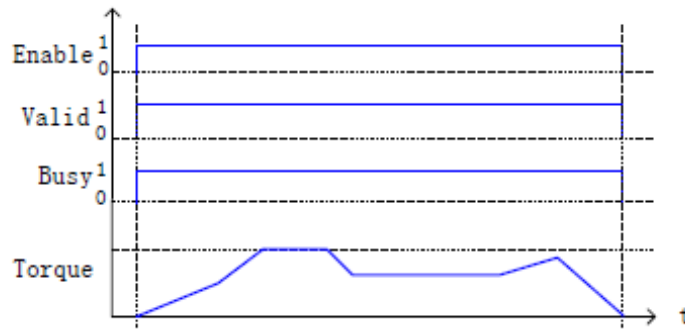
（3） 功能说明

通过 MC_ReadActualVelocity 读取驱动器中的当前速度值指令，Enable 高电平有效。指令可以重复多次使用，互不影响。

（4） 时序图

功能块的 Enable 必须为 TRUE 的条件；
功能块的 Valid 表示读出的 Torque 为有效的数据值；
功能块的 Busy 表示当前功能块正在执行中。

✧ 时序操作说明



2.1.24 MC_SetPosition

MC_SetPosition: 将指令中的位置数据设为当前轴的位置数据，对设置位置数据操作不会产生任何位移移动，用于产生坐标系的位移。

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_SetPosition	当前位置变更		<pre>MC_SetPosition(Axis:= , Execute:= , Position:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

（2） 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿触发将启动功能块的处理
Position	轴位置数据	LREAL	-	0	位置数据
Mode	设定值	BOOL	TRUE, FALSE	FALSE	位置模式；TRUE：相对位置 (RELATIVE)；

输入变量	名称	数据类型	有效范围	初始值	描述
					FALSE：绝对位置 (ABSOLUTE);

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE
ErrorID	错误代码	SMC_ERROR	-	0	发生异常时，输出错误代码
Velocity	获取的当前速度值	LREAL	速度值	0	指令读出来的当前速度数据

（3）功能说明

通过 MC_SetPosition 设置轴的位置参数，不产生任何位移，但形成了坐标偏移；指令为 Execute 上升沿触发；指令可重复多次使用，互不影响。

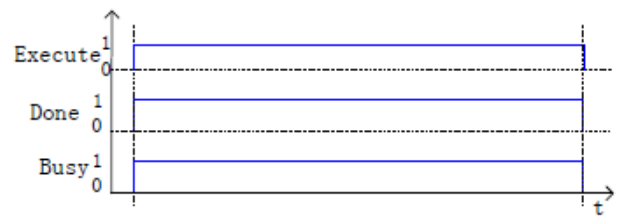
（4）时序图

功能块的 Execute 必须为上升沿触发条件；

功能块的 Done 表示设置操作成功；

功能块的 Busy 表示当前功能块正在执行中。

◇ 时序操作说明



2.1.25 MC_TouchProbe

MC_TouchProbe：指令通过外部信号触发，保存当前轴的位置数据。

（1）指令格式

指令	名称	图形表现	ST 表现
MC_TouchProbe	启用外部锁定		<pre>MC_TouchProbe(Axis:= , TriggerInput:= , Execute:= , WindowOnly:= , FirstPosition:= , LastPosition:= , Done=> , Busy=> , Error=> , ErrorID=> , RecordedPosition=> , CommandAborted=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例
TruggerInput	触发信号	TRIIGGER_REF			触发信号或触发属性等关联属性

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿触发将启动功能块的处理
WindowOnly	触发窗口	BOOL	TRUE, FALSE	FALSE	
FirstPosition	触发开始位置	LREAL		0	指定接收触发的开始位置
LastPosition	触发结束位置	LREAL		0	指定接收触发的结束位置

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
Velocity	获取的当前速度值	LREAL	速度值	0	指令读出来的当前速度数据
RecordedPosition	触发记录位置	LREAL			触发发生时当前的位置
CommandAbort	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE

(3) 功能说明

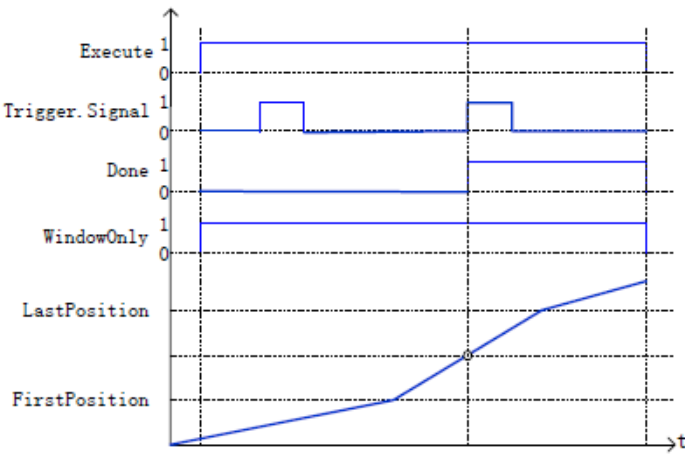
通过 MC_TouchProbe 功能块 TruggerInput 的信号触发时记录运行轴的当前位置。Execute 上升沿执行执行驱动器锁存时: 驱动器采集到锁存信号会在记录位置。

(4) 时序图

功能块的 Execute 必须为上升沿触发条件;

功能块的 Done 表示设置操作成功;

◇ 时序操作说明



2.1.26 MC_MoveContinuousAbsolute

MC_MoveContinuousAbsolute: 轴按绝对位置连续运行(单位按轴设置), 绝对位置由 Distance 指定, 最后的运行速度 EndVelocity 运行。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_MoveContinuousAbsolute	轴绝对位置连续控制指令		<pre>SMC_MoveContinuousAbsolute(Axis:= , Execute:= , Position:= , Velocity:= , EndVelocity:= , EndVelocityDirection:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , InEndVelocity=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿触发将启动功能块的处理
Distance	运动相对位置	LREAL	数据范围	0	此数据为运动的相对位置
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度

输入变量	名称	数据类型	有效范围	初始值	描述
EndVelocity	运行结束速度	LREAL	数据范围	0	指令执行完成后的运行速度
EndVelocity-Direction	结束速度的方向	MC_Direction	positive, negative, current;	Current	可以使用: positive, negative, current; 不可使用: shortest, fastest
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Direction	运行方向	shortest	数据范围	shortest	对于线性/ 直线轴: positive, negative; 对于旋转/ 圆周轴: positive, negative, current, shortest, fastest

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InEndVelocity	指令位置到达	BOOL	TRUE, FALSE	FALSE	轴指令执行位置到达, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
CommandAbort	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE

(3) 功能说明

本功能块为轴绝对定位指令, **Distance** 数据为轴的绝对位置。本功能块运行状态为 **Standstill** 中, 指令运行时的状态为 **Discrete Motion**, 一个完整的运行过程一定要控制轴的不同运动状态。启动指令为 **Execute** 的上升沿启动, 本指令在 **Discrete Motion** 可以重复上升沿有效, 每次都可以刷新最新的 **Position** 位置。**Acceleration** 或 **Deceleration** 为零, 指令运行都为异常状态, 但轴状态为 **Discrete Motion**。

(4) 时序图

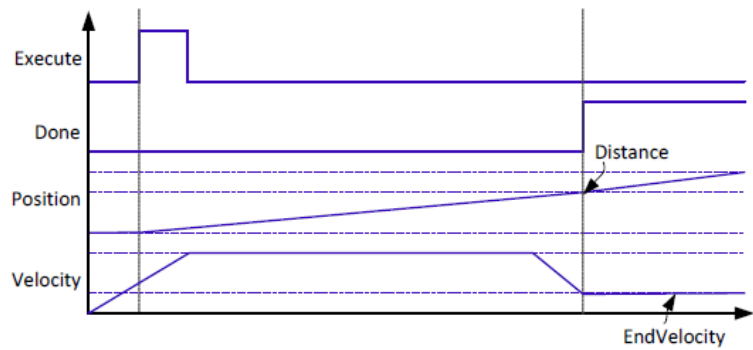
轴必须处于 **Standstill** 状态指令才能运行;

功能块的 **Execute** 必须为上升沿触发条件;

功能块的 **Done** 表示设置操作成功;

功能块的 **Busy** 表示当前功能块正在执行中;

◇ 时序操作说明



2.1.27 MC_MoveContinuousRelative

MC_MoveContinuousRelative: 轴按相对位置连续运行（单位按轴设置），绝对位置由 Distance 指定，最后的运行速度 EndVelocity 运行。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_MoveContinuousRelative	轴相对位置连续控制指令		<pre>SMC_MoveContinuousRelative(Axis:= , Execute:= , Distance:= , Velocity:= , EndVelocity:= , EndVelocityDirection:= , Acceleration:= , Deceleration:= , Jerk:= , InEndVelocity=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行条件	BOOL	TRUE, FALSE	FALSE	上升沿触发将启动功能块的处理
Distance	运动相对位置	LREAL	数据范围	0	此数据为运动的相对位置
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
EndVelocity	运行结束速度	LREAL	数据范围	0	指令执行完成后的运行速度
EndVelocity-Direction	结束速度的方向	MC_Direction	Positive, Negative, current	Current	可以使用: positive, negative, current, 不可使用: shortest, fastest
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值

输入变量	名称	数据类型	有效范围	初始值	描述
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值
Direction	运行方向	shortest	数据范围	shortest	对于线性/ 直线轴: positive, negative, 对于旋转/ 圆周轴: positive, negative, current, shortest, fastest

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InEndVelocity	指令位置到达	BOOL	TRUE, FALSE	FALSE	轴指令执行位置到达, 置为 TRUE
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
CommandAbort	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE

(3) 功能说明

本功能块运行状态为 Standstill 中, 指令运行时的状态为 Discrete Motion, 在指令执行中关注本轴的运行状态, 避免打断本轴的其他指令或被其他指令打断本轴的执行。启动指令为 Execute 的上升沿启动, 本指令在 Discrete Motion 可以重复上升沿有效, 每次都可以刷新最新的 Position 位置。Acceleration 或 Deceleration 为零, 指令运行都为异常状态, 但轴的状态为 Discrete Motion。

(4) 时序图

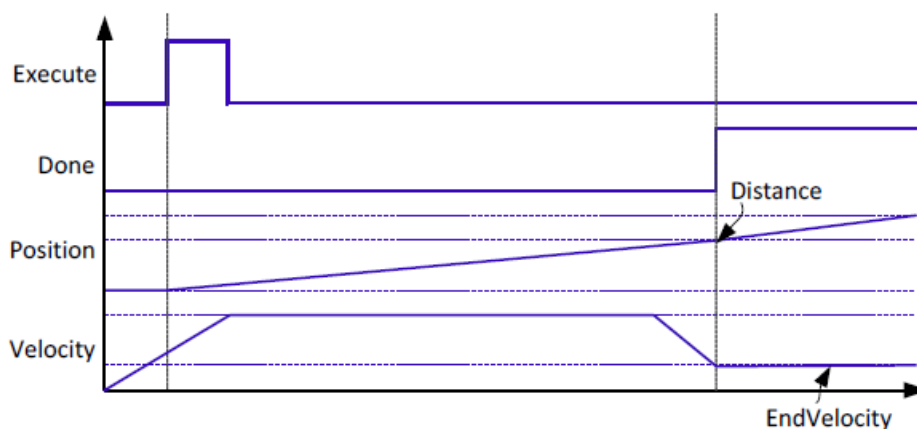
轴必须处于 Standstill 状态指令才能运行;

功能块的 Execute 必须为上升沿触发条件;

功能块的 Done 表示设置操作成功;

功能块的 Busy 表示当前功能块正在执行中;

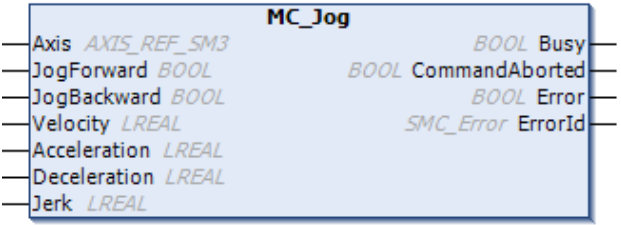
◇ 时序操作说明



2.1.28 MC_Jog

MC_Jog: 根据指定目标速度执行微动移动

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Jog'	轴点动指令		<pre> MC_Jog(Axis:= , JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
JogForward	正向有效	BOOL	TRUE, FALSE	FALSE	设置为 TRUE 则开始正向移动；设置为 FALSE 则停止正向移动
JogBackward	负向有效	BOOL	TRUE, FALSE	FALSE	设置为 TRUE 则开始反向移动；设置为 FALSE 则停止反向移动
Velocity	目标速度	LREAL	正数或“0”	0	指定目标速度。单位：[指令单位/s]
Acceleration	加速度	LREAL	正数或“0”	0	指定加速度。单位：[指令单位/s]
Deceleration	减速度	LREAL	正数或“0”	0	指定减速度。单位：[指令单位/s]
Jerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中，置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断，置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时，置为 TRUE

输出变量	名称	数据类型	有效范围	初始值	描述
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

（3） 功能说明

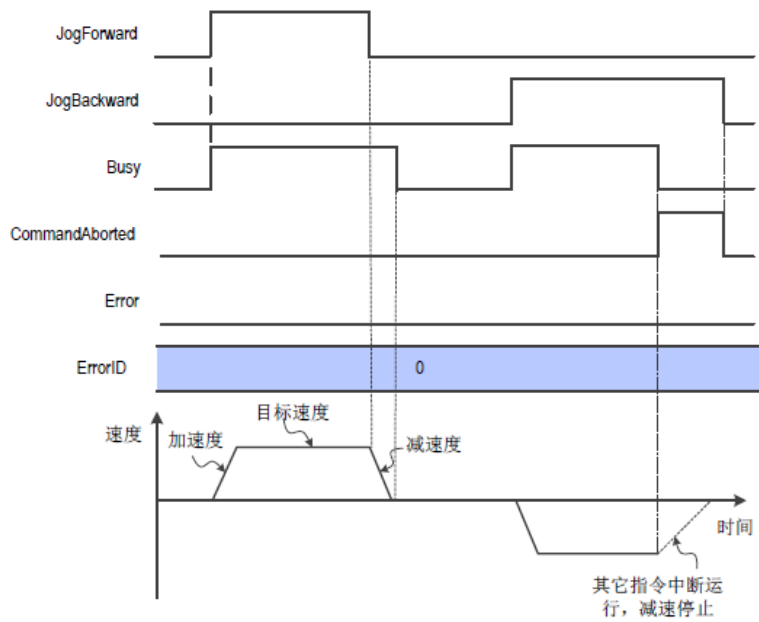
根据指定 Velocity（目标速度）执行点动运行。需要正向运行时，将 JogForward（正向运行有效）置为 TRUE；需要反向运行时，将 JogBackward（负向运行有效）置为 TRUE。同时将 JogForward（正向运行有效）和 JogBackward（负向运行有效）置为 TRUE，将不会有运动发生。如果 MC_Jog 指令的指令速度设置值超过轴参数中的点动最高速度，则以点动最高速度执行。

（4） 时序图

在启动 JogForward（正向运行有效）或 JogBackward（负向运行有效）的同时，Busy（执行中）变为 TRUE；在 JogForward（正向运行有效）或 JogBackward（负向运行有效）的下降沿开始减速并停止轴的同时，Busy（执行中）变为 FALSE。

利用其它指令中止本指令时，CommandAborted（执行中断）变为 TRUE，Busy（执行中）变为 FALSE。

◇ 时序操作说明



2.1.29 MC_Inch

MC_Inch：轴单步运动控制，通过程序能实现一步接一步的单步控制。

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_Inch	轴相对定位指令	The block diagram for the SMC_Inch instruction shows the following inputs and outputs: - Inputs: Axis (AXIS_REF_SM3), InchForward (BOOL), InchBackward (BOOL), Distance (LREAL), Velocity (LREAL), Acceleration (LREAL), Deceleration (LREAL), Jerk (LREAL). - Outputs: Busy (BOOL), CommandAborted (BOOL), Error (BOOL), ErrorId (SMC_ERROR).	<pre>SMC_Inch(Axis:= , InchForward:= , InchBackward:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
InchForward	正向执行	BOOL	TRUE, FALSE	FALSE	如果 InchForward 为 TRUE, 轴将按照给定的速度以正向直到到达距离。输入必须被指定为 FALSE 然后再为 TRUE 再次启动运动。 如果 InchForward 在到达位置之前被设置为 FALSE, 那么轴将立即减速到 0 并且 Busy 将会被设置为 FALSE。如果输入 InchBackward 在仿真情况下被设置为 TRUE, 那么将不会有运动产生。
InchBackward	反向执行	BOOL	TRUE, FALSE	FALSE	如果 InchBackward 为 TRUE, 轴将会按照给定的速度值进行运动以反向运动到设定位置。然后输入必须被设置为 FALSE, 然后再设置为 TRUE 启动另一个运动。 如果输入信号 InchForward 同时被设置为 TRUE, 那么将不会有轴运动。
Distance	移动的距离	LREAL	数据范围	0	此数据为运动的距离
Velocity	运行速度	LREAL	数据范围	0	轴运行到目标位置的最大速度
Acceleration	加速度	LREAL	数据范围	0	速度变大时加速度值
Deceleration	减速度	LREAL	数据范围	0	速度变小时减速度值

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	当前指令被中断, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

（3）功能说明

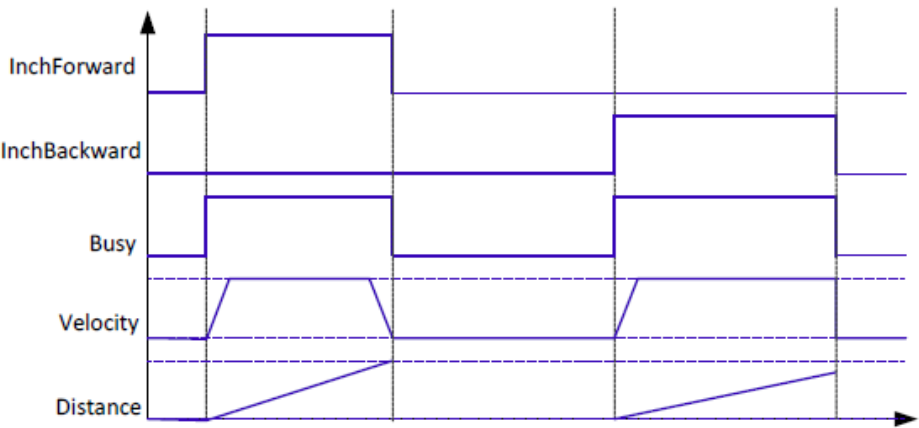
本功能块运行状态为 Standstill 中，指令运行时的状态为 Discrete Motion，在指令执行中关注本轴的运行状态，避免打断本轴的其他指令或被其他指令打断本轴的执行。Acceleration 或 Deceleration 为零，指令运行都为异常状态，但轴的状态为 Discrete Motion；

（4）时序图

功能块的 InchForward/ InchBackward 必须有 TRUE/FALSE 的条件；

功能块的 Busy 表示当前功能块正在执行中；

✧ 时序操作说明



2.1.30 SMC3_PersistPosition

该指令用来保持纪录实轴绝对值编码器的位置（断电重启控制器后，恢复断电前位置记录值）。如果伺服电机使用的是绝对值编码器，使用该功能块配合使用。

（1）指令格式

指令	名称	图形表现	ST 表现
SMC3_PersisitPosition	轴位置保持		<pre>SMC3_PersistPosition(Axis:= , PersistentData:= , bEnable:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);</pre>

（2）相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_PersistPosition_Data			存储位置信息的断电保持型 数据结构

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行	BOOL	TRUE, FALSE	FALSE	True 功能块执行, false 不执行功能块

◇ 输出变量

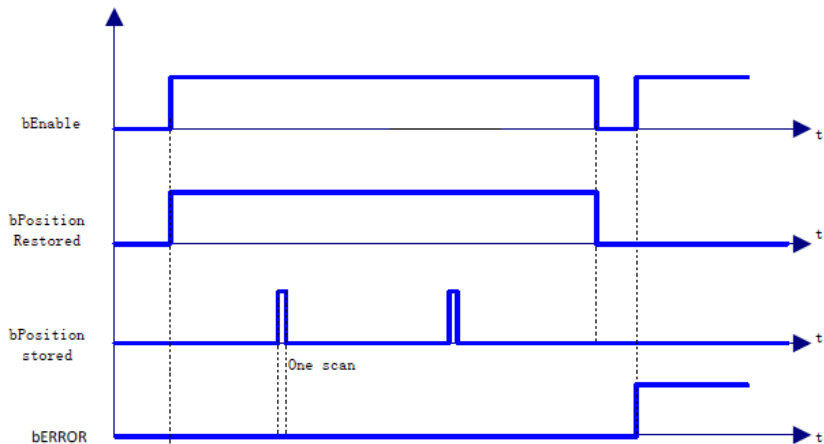
输出变量	名称	数据类型	有效范围	初始值	描述
bPositionRestored	位置恢复	BOOL			TRUE, 轴重启后位置恢复完成
bPositionStored	位置保存	BOOL			TRUE, 调用功能块后保存位置完成
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
eRestoringDiag	恢复诊断	SMC3_Persist-PositionDiag			位置恢复中的诊断信息 SMC3_PPD_RESTORING_OK: 位置成功恢复 SMC3_PPD_AXIS_PROP_CHANGED: 轴参数有更改, 无法恢复位置 SMC3_PPD_DATA_STORED_DURING_WRITING: 功能块从轴参数数据结构复制数据, 而不是从 PersistentData 数据中复制。可能原因: 非同步性持续变量、控制器崩溃死机

(3) 功能说明

PLC 重启 bEnable 信号为 TRUE, 则 bPositionRestored 输出为 TRUE。

（4）时序图

功能块的 **Busy** 表示当前功能块正在执行中。



2.1.31 SMC3_PersistPositionSingleturn

该指令用来保持纪录实轴绝对值编码器（单圈绝对值）的位置（断电重启控制器后，恢复断电前位置记录值）。如果伺服电机使用的是单圈绝对值编码器，使用该功能块配合使用。

（1）指令格式

指令	名称	图形表现	ST 表现
SMC3_PersistPositionSingleturn	轴位置保持		<pre>SMC3_PersistPositionSingleturn (Axis:= , PersistentData:= , bEnable:= , usiNumberOfAbsoluteBit S:= , bPositionRestored=> , bPositionStored=> , bBusy=> , bError=> , eErrorID=> , eRestoringDiag=>);</pre>

（2）相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_PersistPosition_Data			存储位置信息的断电保持型数据结构

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行	BOOL	TRUE, FALSE	FALSE	True 功能块执行, false 不执行功能块

◇ 输出变量

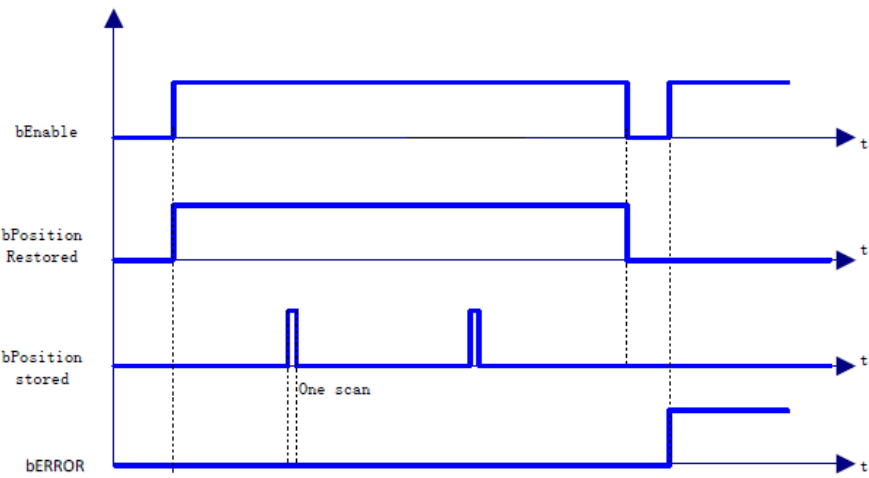
输出变量	名称	数据类型	有效范围	初始值	描述
bPositionRestored	位置恢复	BOOL			TRUE, 轴重启后位置恢复完成
bPositionStored	位置保存	BOOL			TRUE, 调用功能块后保存位置完成
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
eRestoringDiag	恢复诊断	SMC3_Persistent-PositionDiag			位置恢复中的诊断信息 SMC3_PPD_RESTORING_OK: 位置成功恢复 SMC3_PPD_AXIS_PROP_CHANGED: 轴参数有更改, 无法恢复位置 SMC3_PPD_DATA_STORED_DURING_WRITING: 功能块从轴参数数据结构复制数据, 而不是从 PersistentData 数据中复制。 可能原因: 非同步性持续变量、控制器崩溃死机

(3) 功能说明

PLC 重启 bEnable 信号为 TRUE, 则 bPositionRestroed 输出为 TRUE。

(4) 时序图

功能块的 Busy 表示当前功能块正在执行中;



2.1.32 SMC3_PersistPositionLogical

该指令用来保持纪录实轴绝对值编码器（单圈绝对值）的位置（断电重启控制器后，恢复断电前位置记录值）。如果伺服电机使用的是单圈绝对值编码器，使用该功能块配合使用。

(1) 指令格式

指令	名称	图形表现	ST 表现
SMC3_PersistPositionLogical	轴位置保持		<pre>SMC3_PersistPositionLogical(Axis:=, PersistentData:=, bEnable:=, bPositionRestored=>, bPositionStored=>, bBusy=>, bError=>, eErrorID=>, eRestoringDiag=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例
PersistentData	保持数据	SMC3_PersistPosition_Data			存储位置信息的断电保持型数据结构

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行	BOOL	TRUE, FALSE	FALSE	True 功能块执行, false 不执行功能块

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bPositionRestored	位置恢复	BOOL			TRUE, 轴重启后位置恢复完成
bPositionStored	位置保存	BOOL			TRUE, 调用功能块后保存位置完成
Busy	指令正在执行	BOOL	TRUE, FALSE	FALSE	当前指令正在执行中, 置为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
eRestoringDiag	恢复诊断	SMC3_Persist-PositionDiag			位置恢复中的诊断信息 SMC3_PPD_RESTORING_OK: 位置成功恢复 SMC3_PPD_AXIS_PROP_CHANGED: 轴参数有更改, 无法恢

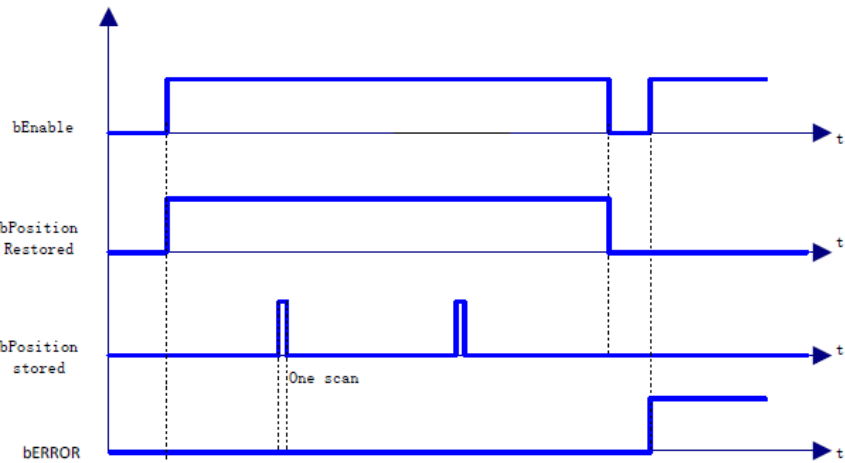
输出变量	名称	数据类型	有效范围	初始值	描述
					复位置 SMC3_PPD_DATA_STORED_DURING_WRITING: 功能块从轴 参数数据结构复制数据，而不是从 PersistentData 数据中复制。 可能原因：非同步性持续变量、 控制器崩溃死机

（3）功能说明

PLC 重启 bEnable 信号为 TRUE，则 bPositionRestroed 输出为 TRUE。

（4）时序图

功能块的 Busy 表示当前功能块正在执行中。



2.1.33 SMC_Homing

轴回零指令，与MC_Home有区别，MC_Home为伺服驱动器控制的回零方式，SMC_Homing指令为PLC控制器控制的回零方式。

（1）指令格式

指令	名称	图形表现	ST 表现
SMC_Homing	轴回零指令		<pre>SMC_Homing(Axis:=, bExecute:=, fHomePosition:=, fVelocitySlow:=, fVelocityFast:=, fAcceleration:=, fDeceleration:=, fJerk:=, nDirection:=, bReferenceSwitch:=, fSignalDelay:=, nHomingMode:=, bReturnToZero:=, bIndexOccured:=,</pre>

指令	名称	图形表现	ST 表现
			findexPosition:=, bIgnoreHWLimit:=, bDone=>, bBusy=>, bCommandAborted=>, bError=>, nErrorID=>, bStartLatchingIndex=>);

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF	-	-	映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	执行	BOOL	TRUE, FALSE	FALSE	True 功能块执行, false 不执行功能块
fHomePosition	原点设定位置	LREAL		0	回零后原点设定位置, 单位为用户标定后的单位
fVelocitySlow	慢速	LREAL		0	离开参考开关后慢速设定速度
fVelocityFast	快速	LREAL		0	接近参考开关置位时, 快速设定速度
fAcceleration	加速度	LREAL		0	加速度设定值
fDeceleration	减速度	LREAL		0	减速度设定值
fJerk	加加速度	LREAL		0	
nDirection	回零方向	MC_DIRECTION		negative	回零开始方向, 参考 MC_DIRECTION
bReferenceSwitch	参考开关	BOOL	TRUE, FALSE	FALSE	连接参考开关, TRUE: 参开开关触发, FALSE: 参考开关闭合
fSignalDelay	延迟	LREAL		0	参考开关的传输时间, 用来补偿死区时间。 单位为秒。
nHomingMode	回零模式	SMC_HOMING_MODE			参考 SMC_HOMING_MODE
bReturnTozero	返回零位	BOOL	TRUE, FALSE	FALSE	TRUE: 回零完成后轴运行到位置零 (注意: 如果 fHomePosition=10, 则回零完成后轴位置变为 10, bReturnTozero 为

输入变量	名称	数据类型	有效范围	初始值	描述
					ture 则回零完成后轴反向走 10 个单位到 0 位)
bIndexOccured		BOOL	TRUE, FALSE	FALSE	True, 标志脉冲记录, 回零模式为 FAST_BSLOW_I_S_STOP, FAST_SLOW_I_S_STOP 时生效
fIndexPosition		LREAL		0	标志脉冲时记录的位置
bIgnoreHWLimit	忽略硬限位	BOOL	TRUE, FALSE	FALSE	TRUE, 设定硬件限位开关使能 为 false, 如果相同的物理开关用于硬件限位开关和参考开关, 那么硬件控制将被设置为假

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone		BOOL	TRUE, FALSE	FALSE	TRUE, 回零完成
Busy		BOOL	TRUE, FALSE	FALSE	TRUE, 功能块正在执行
bCommandAborted		BOOL	TRUE, FALSE	FALSE	True, 功能块被其他动作指令中断
Error		BOOL	TRUE, FALSE	FALSE	异常发生时, 置为 TRUE
ErrorID		SMC_ERROR		0	发生异常时, 输出错误代码
bStartLatchingIndex		BOOL	TRUE, FALSE	FALSE	由“bIndexOccured”和“fIndexPosition”共同作用产生

具体的回零模式描述, 见下表:

模式	类型	初始值	描述
FAST_BSLOW_S_STOP	SMC_HOMING_MODE	0	轴按照设定方向以快速走向原点开关, 碰到原点开关后反向以慢速离开原点开关, 离开后先执行 MC_setPosition 将当前位置设为 fHomePosition 设定值, 然后执行 MC_stop
FAST_BSLOW_STOP_S	SMC_HOMING_MOD	1	按照设定方向以快速走向原点开关, 碰到原点开关后反向以慢速离开原点开关, 离开后先执行 MC_stop 将轴停止, 然后执行 MC_setPosition, 将当前位置设为 fHomePosition 设定值
FAST_BSLOW_I_S_STOP	SMC_HOMING_	2	按照设定方向以快速走向原点开关, 碰

模式	类型	初始值	描述
	MOD		到原点开关后反向以慢速离开原点开关, bIndexOccured 信号到时先执行 MC_setPosition 再执行 MC_stop
FAST_SLOW_S_STOP	SMC_HOMING_MOD	4	按照设定方向以快速走向原点开关, 碰到原点开关后以慢速离开原点开关, 离开后先执行 MC_setPosition 将当前位置设为 fHomePosition 设定值, 然后执行 MC_stop
FAST_SLOW_STOP_S	FAST_SLOW_STOP_S SMC_HOMING_MOD	5	按照设定方向以快速走向原点开关, 碰到原点开关后以慢速离开原点开关, 离开后先执行 MC_stop, 然后执行 MC_setPosition 将当前位置设为 fHomePosition 设定值。
FAST_SLOW_I_S_STOP	SMC_HOMING_MOD	6	按照设定方向以快速走向原点开关, 碰到原点开关后原方向以慢速离开原点开关, bIndexOccured 信号到时先执行 MC_setPosition 再执行 MC_stop

(3) 功能说明

SMC_HOMING 通过 bExecute 的上升沿启动之后, 轴将会按照速度 fVelocityFast 并以 nDirection 定义的方向开始运动, 直到 bReferenceSwitch = FALSE。然后轴将会缓慢停止并按照相反的方向以速度 fVelocitySlow 离开参考开关。bReferenceSwitch = TRUE 后回零完成。

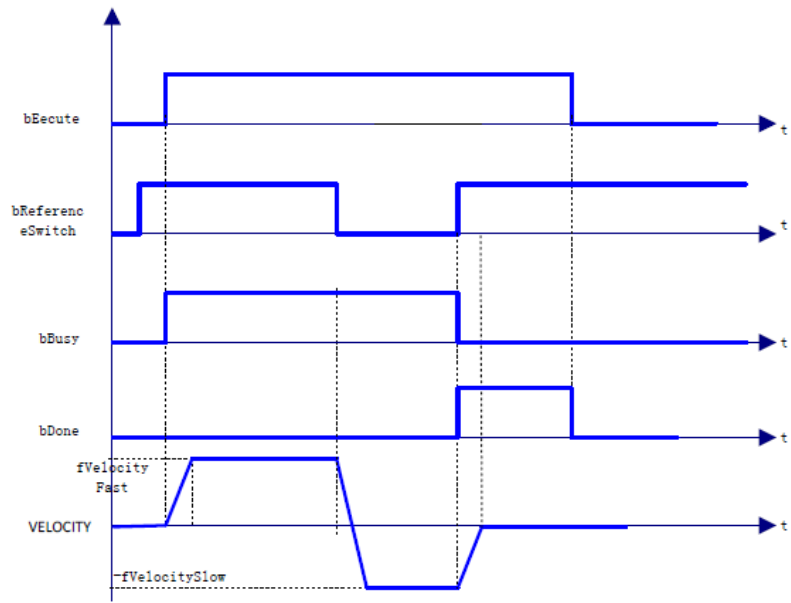
使能回零指令后 bReferenceSwitch 的状态为 ON->OFF->ON, 在 OFF->ON 的上升沿回零完成, 设置参考位置。参考位置 = fHomePosition + ((fSignalDelay * 1000 + 1 个 DC 时钟周期) / 1000) * fVelocitySlow 实际就是补偿了设置的 bReferenceSwitch 采样延迟和一个通讯周期位移延迟。

如果 bReturnToZero = TRUE, bReferenceSwitch 的状态在 OFF->ON 的上升沿将参考位置设置为 fHomePosition + ((fSignalDelay * 1000 + 1 个 DC 时钟周期) / 1000) * fVelocitySlow, 然后按照速度 fVelocityFast 运行到 0 位置。

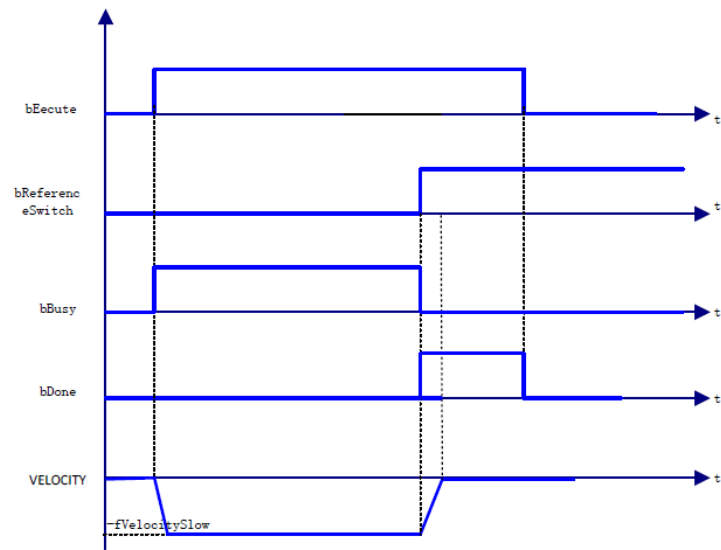
注意: Done 完成信号后, 轴位置设定为: fHomePosition。设定的时机跟 nHomingMode 有关 (详情参考 SMC_HOMING_MODE)。

(4) 时序图

① 指令执行时 bReferenceSwitch TRUE 时；



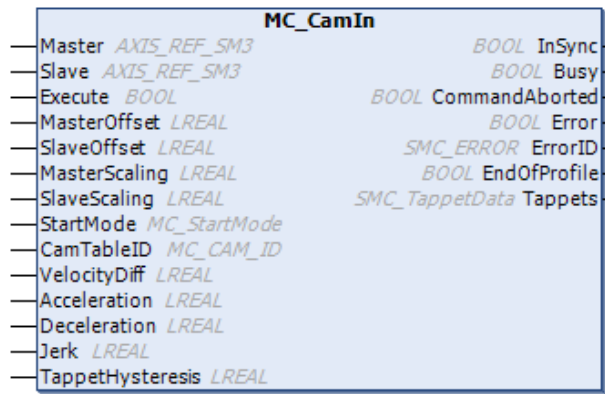
② 指令执行时 bReferenceSwitch FALSE 时



2.1.34 MC_Camin

MC_Camin: 使用指定的凸轮表开始执行电子凸轮动作, 可根据应用需求指定主从轴偏置值、缩放比和工作模式。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Camin	凸轮动作 开始		<pre> MC_Camin(Master:= , Slave:= , Execute:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , StartMode:= , CamTableID:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , TappetHysteresis:= , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , EndOfProfile=> , Tappets=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF	-	-	映射到轴, 即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF			映射到轴, 即 AXIS_REF_SM3 的一个实例

注: 主/从轴必须指定不同轴, 否则可能报错。

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行凸轮功 进入能块	BOOL	TRUE, FALSE	FALSE	上升沿, 执行电子凸轮
MasterOffset	主轴偏置	LREAL	负数, 正数, 0	0	以指定的偏移值移动主轴的相位
SlaveOffset	从轴偏执	LREAL	负数, 正数, 0	0	以指定的偏移值移动从轴的相位
MasterScaling	主轴预编译 比例	LREAL	>0.0	1	以指定的比例放大/ 缩小主轴的相位
SlaveScaling	从轴预编译 比例	LREAL	>0.0	1	以指定的比例放大/ 缩小从轴轴的相位
StartMode	从轴相对凸 轮输出模式	MC_ StartMode		absolute	0: absolute 绝对位置 1: relative 相对位置

输入变量	名称	数据类型	有效范围	初始值	描述
					2: ramp_in (斜坡切入) 3: ramp_in_pos (正向斜坡切入) 4: ramp_in_neg 反向斜坡切入
CamTableID	表格编号	MC_CAM_ID			定义 cam 表格的使用, 与 MC_CamTableSelect 的输出点 CamTableID 配合使用
VelocityDiff		LREAL			与 ramp_in 不同的最大速度
Acceleration		LREAL			ramp_in 时加速度
Deceleration		LREAL			ramp_in 时减速度
Jerk		LREAL			ramp_in 的加速度
TappetHysteresis		LREAL			挺杆的阻尼系数

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InSync	凸轮生效	BOOL	TRUE, FALSE	FALSE	主轴和从轴建立凸轮关系后, InSync 被置位, 当指令的执行条件 OFF 时, InSync 被复位。
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	Execute 输入上升沿时, 置位 TRUE, TRUE 时表示凸轮关系耦合中, 需用 Cam_out 指令复位, 指令执行条件复位不能复位该状态。
CommandAborted	指令被中断	BOOL	TRUE, FALSE	FALSE	从轴被其他控制指令中断输出为 TRUE
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误, Error 位被置位; 当指令的执行条件 OFF 时, Error 位被复位
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
EndOfProfile	曲线完成	BOOL		FALSE	如果 MC_CamTableSelect 指令执行时 Periodic 参数为 0(非周期), 当凸轮曲线执行完一次后, EndOfProfile 位被置位, 当指令的执行条件 OFF 时, EndOfProfile 位被复位。
Tappets		SMC_TappetData			关联一个凸轮挺杆, 可被 MC_GetTappetValue 指令读出

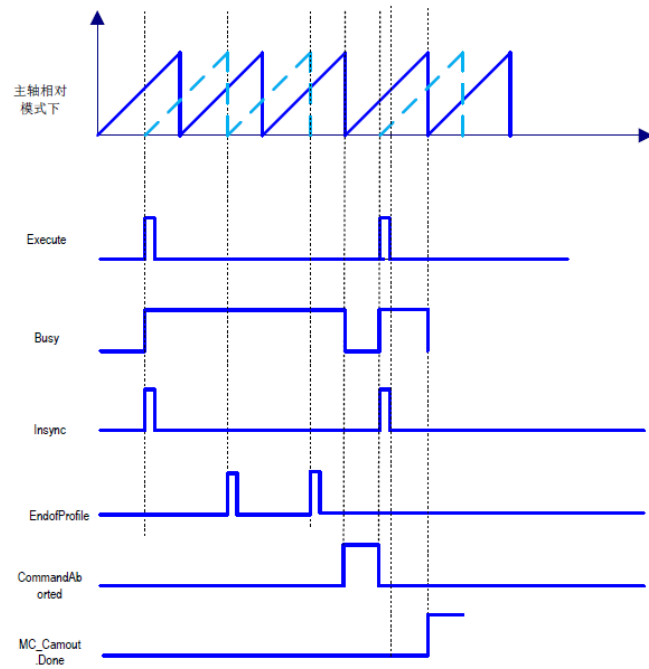
（3） 功能说明

在选择正确的凸轮表以及轴无报错的条件下，Execute 上升沿触发该功能块；在一个凸轮运动系统中，要调用一条凸轮曲线，先调用 MC_CamTableSelect 指令选择相应的凸轮表，再执行 MC_CamIn；如要更换凸轮曲线，则再调用 MC_CamTableSelect 指令重新选择凸轮表。需使用 Camout 指令断开主从轴的凸轮耦合关系。该指令执行时，该指令的从轴再执行其它运动指令时，从轴和主轴之间的凸轮关系会解除，并且 Command-Aborted 输出为 TRUE。

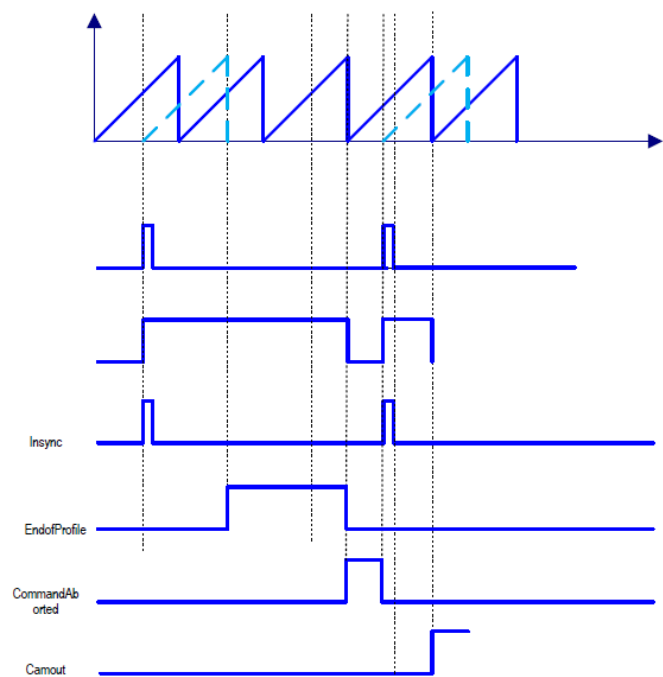
（4） 时序图

✧ 周期模式（MC_CamTableSelect.Periodic 设为 TRUE）如下图：

注意：MC_Camout 指令只切断主从轴的凸轮耦合关系，如果断开时候从轴速度不为 0，则从轴不会自动减速为 0，需配合使用 MC_STOP 指令。



✧ 非周期模式（MC_CamTableSelect.Periodic 设为 FALSE）如下图：

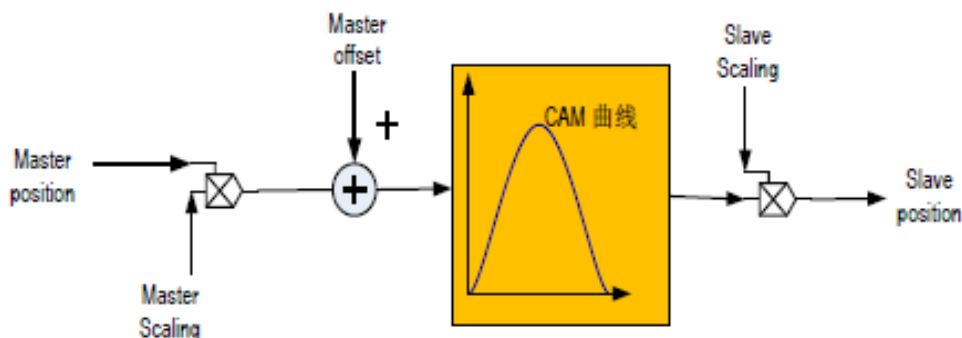


(5) 功能块详解

✧ 指令启动条件

在主轴停止中、位置控制中、速度控制中、同步控制中任意状态下都可以启动本指令。

凸轮曲线中齿合点的计算方法如下：



由上图得出计算方法如下：

$$\text{Position_Slave} = \text{SlaveScaling} * \text{CAM}(\text{MasterScaling} * \text{MasterPosition} + \text{MasterOffset}) + \text{SlaveOffset}$$

该公式中的主轴位置、从轴位置并不是代表实际物理轴的位置，而是凸轮函数曲线相关的主从轴位置。

主从轴位置跟主从实轴位置之间的关系有详细描述。

注意：主轴位置、从轴位置均指的是凸轮函数曲线所需的主从轴位置，而不是主轴、从轴物理实轴的位置。

✧ 周期模式与 EndOfProfile 的关系：

周期模式非周期模式决定了主轴到终止位置后电子凸轮是否要再次进行。

非周期模式：MC_CamTableSelect 指令 Periodic 选择 False：

非周期模式时，凸轮完成 EndofProfile 信号输出为 True，执行输入 FALSE 则 EndofProfile 输出 FALSE。此时凸轮只执行一个主轴周期。

注意：主轴周期指的是电子凸轮主轴位置从起始位置到终止位置的范围。

✧ 周期模式与 EndOfProfile 的关系：

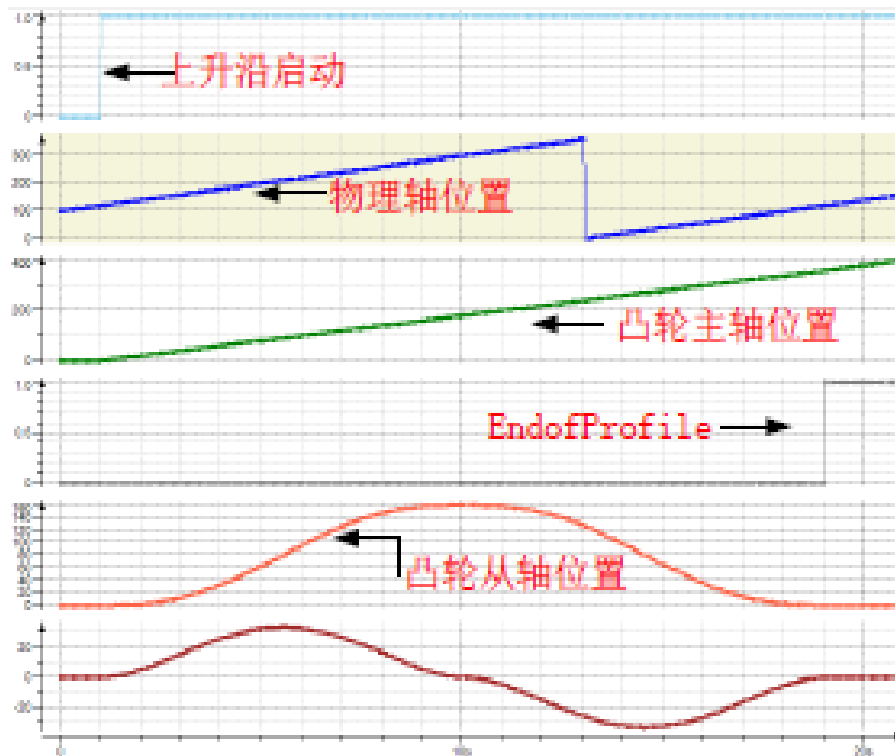
周期模式非周期模式决定了主轴到终止位置后电子凸轮是否要再次进行。

非周期模式：MC_CamTableSelect 指令 Periodic 选择 False：

非周期模式时，凸轮完成 EndofProfile 信号输出为 True，执行输入 FALSE 则 EndofProfile 输出 FALSE。此时凸轮只执行一个主轴周期。

注意：主轴周期指的是电子凸轮主轴位置从起始位置到终止位置的范围。

周期模式：MC_CamTableSelect 指令 Periodic 选择 TRUE

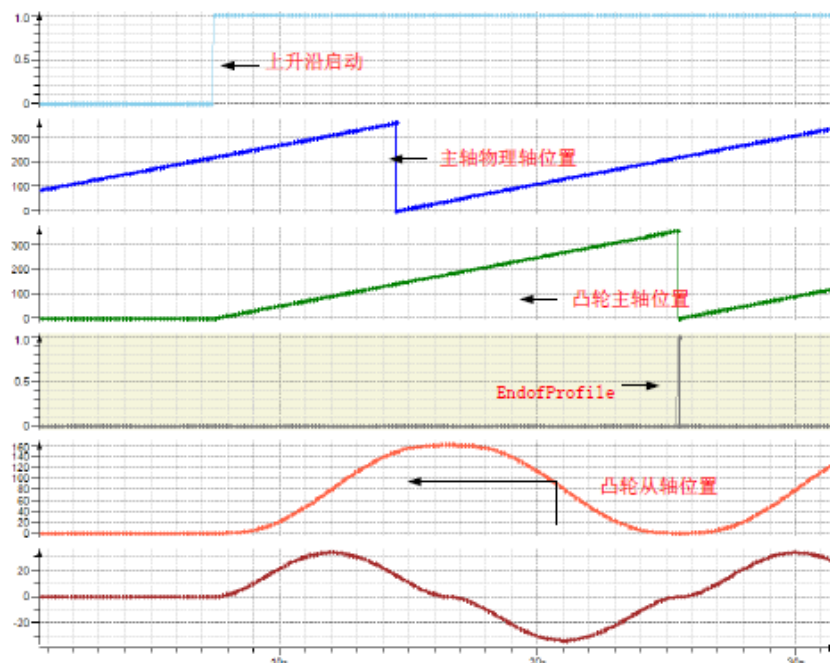


此时凸轮一个主轴周期完成后会接着执行下一个周期，且 EndofProfile 信号 TRUE 输出只维持一个任务周期。

注意：当凸轮主轴位置大于等于凸轮终止位置时 EndofProfile 信号输出为 TRUE，并且凸轮主轴位置更新为凸轮起始位置 + 当前位置 - 终止位置。例如：电子凸轮主轴起始位置为 0、终止位置为 360、主从轴缩放设置为 1、主从轴偏移值设为 0、任务周期 2ms，主轴速度为 100。当某个任务周期凸轮主轴位置为 359.99，那么下个周期 EndofProfile 输出为 True 且主轴位置变为 $359.99 + 100 \times 0.002 - 360 = 0.19$ 。

周期模式下设计的凸轮曲线起始位置跟终止位置最好能保持平滑过渡，否则会产生跳变

比如说起始速度为 0，终止速度不为 0，会导致主轴在周期完成和新周期开始处产生跳变。



StartMode 与 MC_CamTableSlect 中主从轴绝对相对模式的关系：

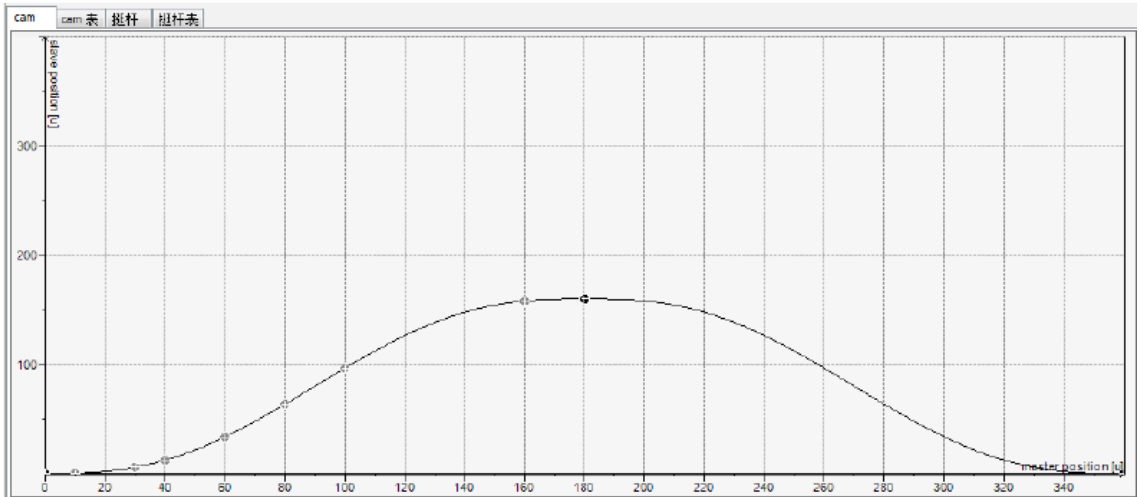
- 绝对模式：**在新的电子凸轮循环开始时，电子凸轮的计算与当前从轴的位置无关。如果从轴相对于主轴起始位置不同于从轴相对于主轴的终止位置，其将造成一个跳变。
- 相对模式：**新的电子凸轮会根据当前从轴的位置进行改变；也就是说，从轴在上一个电子凸轮循环结束的位置，会被现在的电子凸轮运动作为“从轴偏移”进行位置相加的计算。但是，如果在电子凸轮定义中，与主轴起始位置对应的从轴位置不是 0，其将造成一个跳变。
- 斜坡输入：**通过增加一个补偿运动（根据极限值 VelocityDiff、加速度、减速度得出的运动）来防止电子凸轮在开始时的潜在跳变。因此，只要从轴是旋转的方式，正向斜坡输入选项则只进行正向补偿，而反向斜坡输入则只进行反向补偿。对于线性运动的从轴，补偿方向可以自动实现，也就是说，正向斜坡输入和反向斜坡输入可以被用斜坡输入的方式进行解释。）
- 关系表如下表所示：

MC_CamTableSelect.MasterAbsolute	主轴模式
absolute	绝对模式
relative	相对模式

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	从轴模式
absolute	TRUE	绝对模式
absolute	FALSE	相对模式
relative	TRUE	相对模式
relative	FALSE	相对模式
ramp_in	TRUE	斜坡切入绝对模式
ramp_in	FALSE	斜坡切入相对模式
ramp_in_pos	TRUE	正向斜坡切入绝对模式
ramp_in_pos	FALSE	正向斜坡切入相对模式
ramp_in_neg	TRUE	反向斜坡切入绝对模式
ramp_in_neg	FALSE	反向斜坡切入相对模式

详细关系描述如下：

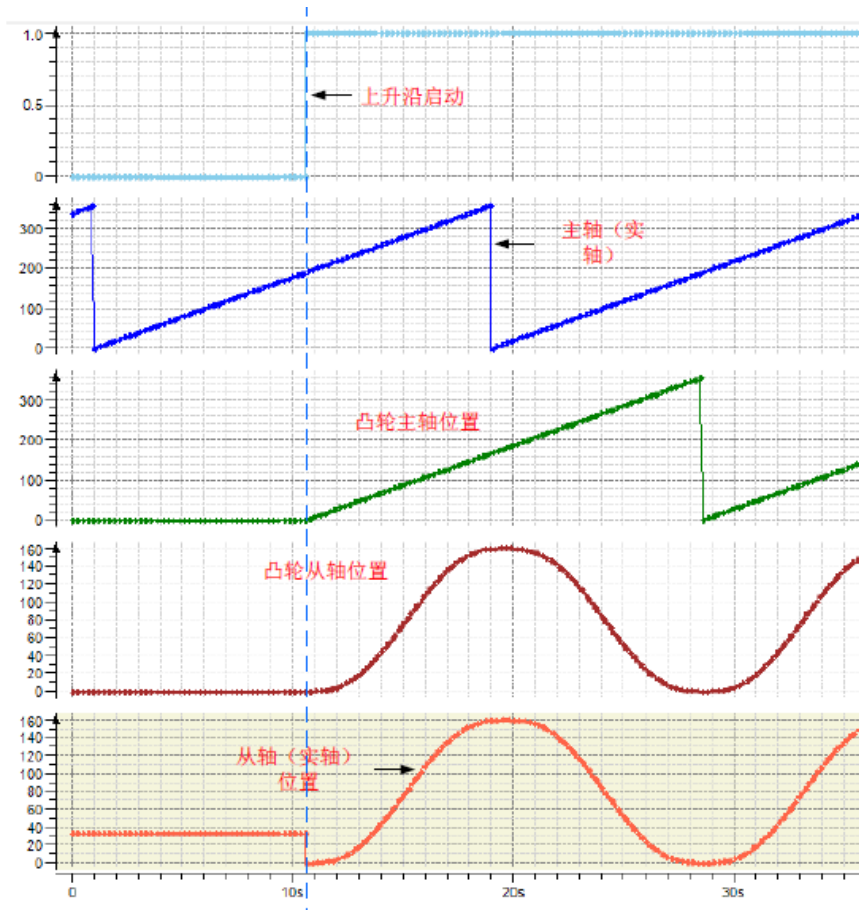
凸轮主轴范围（0-360）、凸轮从轴范围为（0-180）、周期模式、主从轴偏移值 0、主从轴缩放比 1。设计的凸轮表如下图所示：



StartMode 为 0（绝对模式）

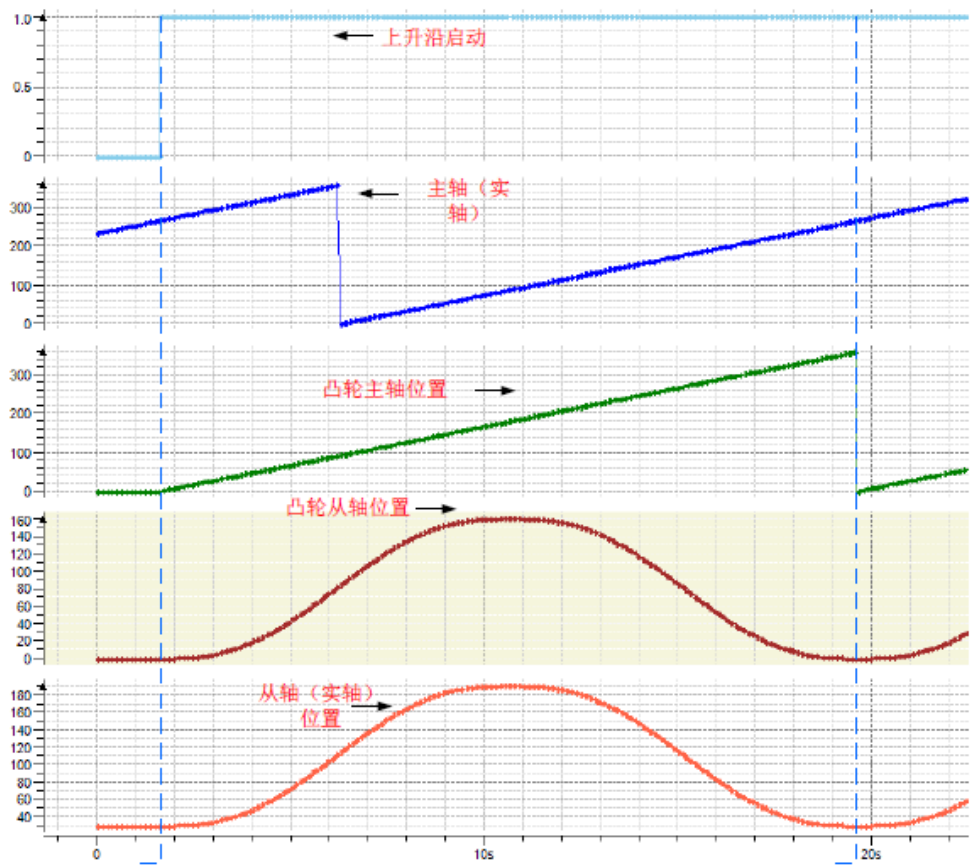
当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE, SlaveAbsolute 设置为 TRUE 时, 此时主轴工作在相对模式, 从轴工作在绝对模式。当 Excute 上升沿, 凸轮启动时, 凸轮主轴从凸轮表“起始位置”(0) 开始, 凸轮从轴按照上述的“凸轮表齿合公式”计算输出, 从轴实轴指令位置等于齿合计算输出值。比如说凸轮从轴起始位置为 0, 凸轮启动时从轴实轴位置为 20, 则启动开始从轴实轴的位置指令为 0 产生跳变。

注意: 该情况下从轴(实轴)开始位置不在凸轮从轴开始位置则会产生跳变。



当 MC_CamTableSlect 指令 MasterAbsolute 设置为 FALSE, SlaveAbsolute 设置为 FALSE 时, 此时主轴工作在相对模式, 从轴工作在相对模式。当 Excute 上升沿, 凸轮启动时, 凸轮主轴从凸轮表“起始位置”(0) 开始, 凸轮从轴按照上述的“凸轮表齿合公式”计算输出, 从轴实轴指令位置等于齿合计算输出值(凸轮从轴位置)+启动时从轴实轴位置。

比如说凸轮启动时从轴实轴位置为 20, 凸轮表从轴起始位置为 0, 则启动凸轮时从轴实轴位置指令为 20, 后续为 $20 + \text{凸轮表计算值}$, 最高峰值为 $20 + \text{凸轮表计算最大值}$ (此处为 180) = 200。



2.1.35 MC_Camout

MC_Camout: 断开从轴的凸轮耦合关系。注意：执行该指令后从轴会按照分离前的速度继续运行，所以需要跟 MC_Stop 等指令配合使用。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Camout	断开凸轮耦合		<pre>MC_CamOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Slave	从轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行凸轮功退 出能块	BOOL	TRUE, FALSE	FALSE	上升沿，执行电子凸轮

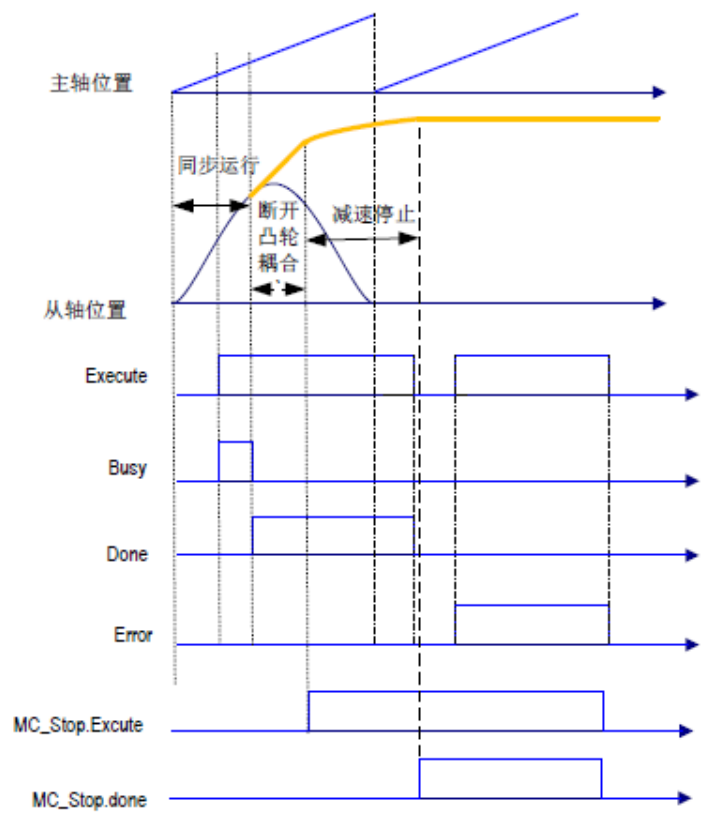
◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE, FALSE	FALSE	完成与主轴的凸轮耦合断开
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	指令执行中。
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误，Error 位被置位；
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

执行本指令解除从轴的凸轮耦合关系。Excute 上升沿时执行从轴的凸轮耦合关系断开；凸轮关系断开后，从轴并不一定是停止的。如果从轴在执行该指令前速度不为 0，则指令 DONE 信号完成后凸轮耦合关系断开，但是从轴仍然按照凸轮关系断开前速度运行。假如从轴没有凸轮耦合关系时执行该指令，输出 Error。

(4) 时序图



2.1.36 MC_CamTableSelect

MC_CamTableSelect: 指定凸轮表，与 MC_CamIn 指令配合使用。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_CamTableSelect	凸轮表指定		<pre>MC_CamTableSelect(Master:=, Slave:=, CamTable:=, Execute:=, Periodic:=, MasterAbsolute:=, SlaveAbsolute:=, Done=>, Busy=>, Error=>, ErrorID=>, CamTableID=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例
CamTable	选择表	MC_CAM_REF			映射到 CAM 表格描述，即 MC_CAM_REF 的一个实例

使用注意事项：主轴和从轴不能指定为同一轴，否则会有报错输出，CamTable 所对应的凸轮表编辑需要正确无误，否则也会导致指令报错。主轴、从轴可以为实轴也可以是虚轴。

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE, FALSE	FALSE	上升沿，执行电子凸轮
Periodic	重复模式	BOOL	TRUE, FALSE	FALSE	指定是反复执行指定的凸轮表还是只执行一次； TRUE: 重复 False: 不重复
MasterAbsolute	主轴绝对模式	BOOL	TRUE, FALSE	FALSE	指定主轴跟踪距离坐标系是按绝对位置还是相对位置； 1: 绝对位置 0: 相对位置
SlaveAbsolute	从轴绝对模式	BOOL	TRUE, FALSE	FALSE	与 MC_CamIn 指令中的 StartMode 综合指定从轴当前指令位置为凸轮表输出的绝对（当前主轴位置对应的凸轮表输出值）还是相对（凸轮表输出值叠加指令开

输入变量	名称	数据类型	有效范围	初始值	描述
					始时从轴位置) 1: 绝对位置 0: 相对位置。

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE, FALSE	FALSE	完成与主轴的凸轮耦合断开
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	指令执行中。
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误, Error 位被置位;
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码
CamTableID	生效 CAMID	MC_CAM_ID			选择生效的 Cam_ID, 与 MC_CamIn 指令中的 CamTableID 配合使用

(3) 功能说明

本指令指定电子凸轮运行所需凸轮表, 所以在使用本指令之前先要将凸轮表编辑好 (凸轮编辑器编辑或者在线编辑好)。Excute 上升沿, 执行指定凸轮表, 亦可凸轮表更新后刷新指定凸轮表; Done 信号输出为 TRUE 时, 则输出变量“CamTableID”生成并且生效; 指令执行中, Busy 信号输出 TRUE, Done 信号 TRUE、Busy 信号为 FALSE; MasterAbsolute、SlaveAbsolute、Periodic 参数具体作用参考 MC_CamIn 指令详情。

2.1.37 MC_GearIn

MC_GearIn: 设定从轴与主轴间的齿轮比, 进行电子齿轮动作。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_GearIn	电子齿轮功能块		<pre> MC_GearIn(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , Acceleration:= , Deceleration:= , Jerk:= , InGear=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF			映射到轴, 即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF			映射到轴, 即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE, FALSE	FALSE	上升沿，触发功能块
RatioNumerator	齿轮比分子	DINT	正数、负数	1	齿轮比分子
RatioDenominator	齿轮比分母	UDINT	正数	1	齿轮比分母
Acceleration	加速度	LREAL	正数或 0		指定加速度
Deceleration	减速度	LREAL	正数或 0		指定减速度
Jerk	跃度	LREAL	正数或 0		加加速度

◇ 输出变量

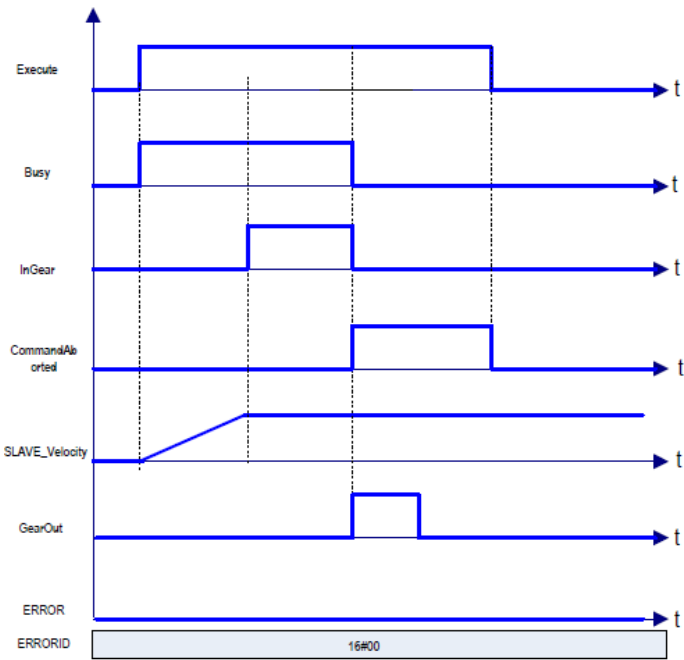
输出变量	名称	数据类型	有效范围	初始值	描述
InGear	齿轮比到达	BOOL	TRUE, FALSE	FALSE	True，从轴达到目标速度
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	指令执行中。
CommandAborted	中断	BOOL	TRUE, FALSE	FALSE	True，被其他控制指令中断
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误，Error 位被置位；
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

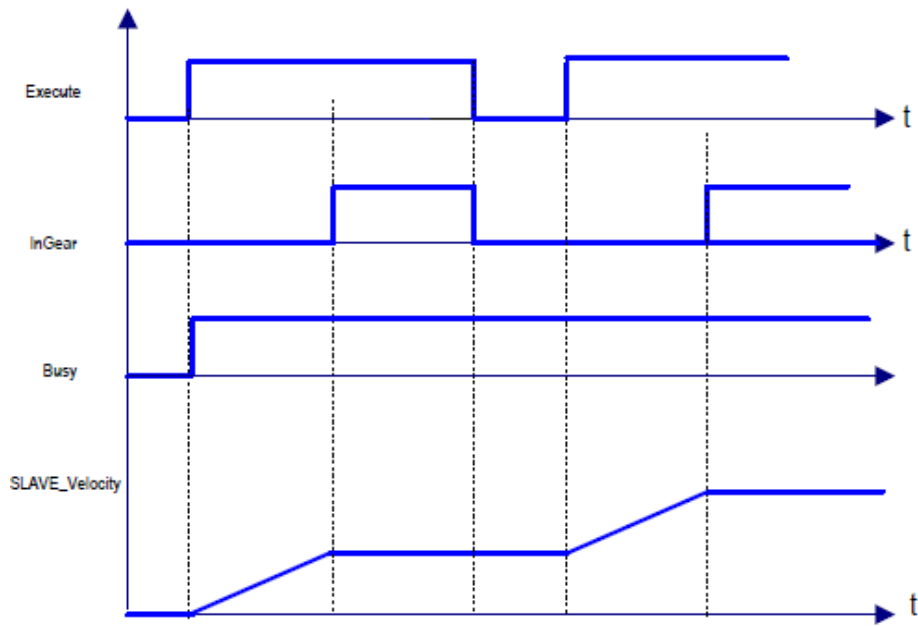
Execute 上升沿，开始电子齿轮动作。执行电子齿轮后要解除耦合必须通过 GearOut 指令。该指令为速度电子齿轮功能，加速过程中造成的同步距离丢失不会自动补偿。指令执行中 Busy 信号为 TRUE 时，如果从轴目标速度没有达到此时 Execute 新的上升沿不会影响它。指令执行中 Busy 信号为 TRUE 时，如果从轴目标速度达到此时 Execute 新的上升沿不会影响它。到达目标速度，InGear 为 TRUE，此后从轴移动量= 主轴移动量* RatioNumerator/ RatioDenominator。如果主轴速度实时变化的情况下请注意慎重使用该指令。

注意：执行指令过程中请不要使用 MC_SetPosition 指令以免引起电机急速运转产生事故。

(4) 时序图



变更齿轮比参数后重启指令时序图如下：



2.1.38 MC_GearOut

MC_GearOut: 终止执行中的 MC_GearIn,MC_GearInPos 指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_GearOut	电子齿轮耦合断开		MC_GearOut(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Slave	从轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE, FALSE	FALSE	上升沿，触发功能块

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE, FALSE	FALSE	True，从轴与主轴电子齿轮耦合断开
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	指令执行中

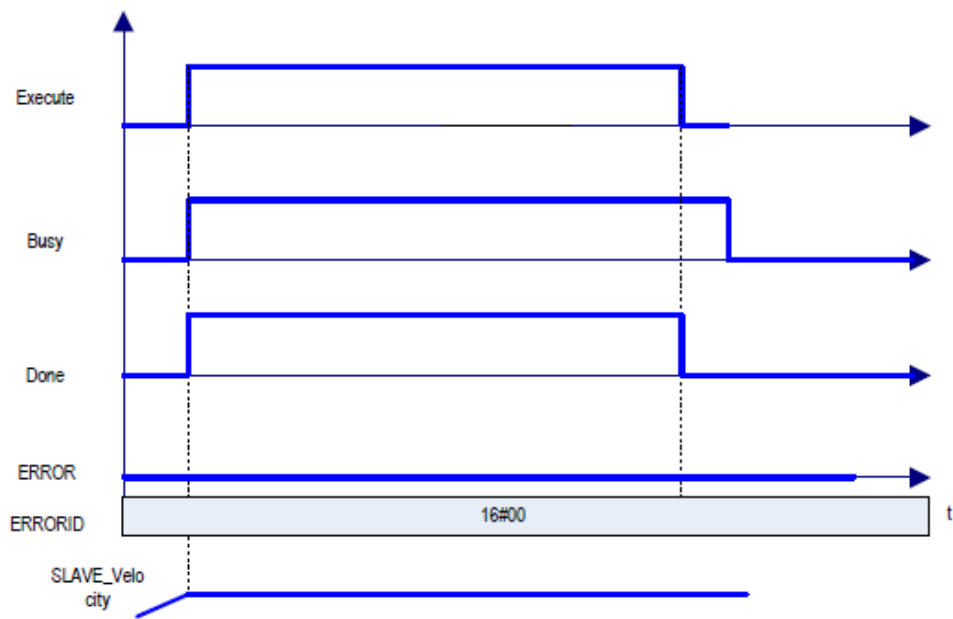
输出变量	名称	数据类型	有效范围	初始值	描述
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误, Error 位被置位
ErrorID	错误代码	SMC_ERROR		0	发生异常时, 输出错误代码

（3）功能说明

Execute 上升沿, 执行切出电子齿轮动作。Excute 为 TRUE, ERROR 为 FALSE 则 Busy 输出为 TRUE, Done 输出为 TRUE。

切出电子齿轮动完成后此时从轴的速度为切出前的速度, 所以需配合以 MC_Stop 指令停止从轴。Execute 下降沿则, Done 为 FALSE。

（4）时序图



2.1.39 MC_GearInPos

MC_GearInPos: 设定主轴与从轴之间的电子齿轮比, 进行电子齿轮动作。指定开始同步的主轴位置、从轴位置、主轴开始同步距离, 并以此来完成切入电子齿轮动作。

（1）指令格式

指令	名称	图形表现	ST 表现
MC_GearInPos	指定位置切入电子齿轮耦合		<pre>MC_GearInPos(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , MasterSyncPosition:= , SlaveSyncPosition:= , MasterStartDistance:= , AvoidReversal:= , StartSync=> , InSync=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例
Slave	从轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE, FALSE	FALSE	上升沿，触发功能块
RatioNumerator	齿轮比分子	DINT	TRUE, FALSE		主从轴速度比的分子
RatioDenominator	齿轮比分母	DINT			主从轴速度比的分母
MasterSyncPosition	主轴同步位置	LREAL			主从轴齿轮比耦合时主轴位置
SlaveSyncPosition	从轴同步位置	LREAL			主从轴齿轮比耦合时从轴位置
MasterStartDistance	执行同步主轴位置	LREAL			从轴按照该位置值和 -MasterSyncPosition 以及 SlaveSyncPosition 值计算一条平滑曲线使从轴在 SlaveSyncPosition 时跟主轴齿轮同步，曲线主轴范围为 [MasterStartDistance, MasterSyncPosition]
AvoidReversal	禁止反转	BOOL	TRUE, FALSE	FALSE	设置为 FALSE，如果从轴物理位置超前的情况下进行反转。设置为 TRUE 如果从轴物理上不能实现反转或者导致危险。只在模态轴下适用。如果反转不能被避免，那么轴将错误停止。

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
StartSync	开始耦合处理	BOOL	TRUE, FALSE	FALSE	True, 开始电子齿轮耦合处理
InSync	耦合中	BOOL	TRUE, FALSE	FALSE	True, 电子齿轮耦合处理完成，主从轴齿轮比耦合中
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	指令执行中
CommandAborted	指令中断	BOOL	TRUE, FALSE	FALSE	被其它控制指令中断
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误，Error 位被置位
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

（3） 功能说明

Execute 上升沿信号，开始执行指令。

开始动作后，Slave（从轴）以 Master（主轴）速度乘以齿轮比得到的速度为目标速度，进行加减速动作。

该功能块同步开始到同步结束的过程本质为同步区间内从轴跟随主轴的一个电子凸轮、此时根据主轴范围（MasterSyncPosition- MasterStartDistance, MasterSyncPosition），从轴范围（当前位置, SlaveSyncPosition），指令会根据设置齿轮比以及上述三个参数自动设计一条凸轮曲线，执行同步时从轴跟随主轴完成凸轮动作。

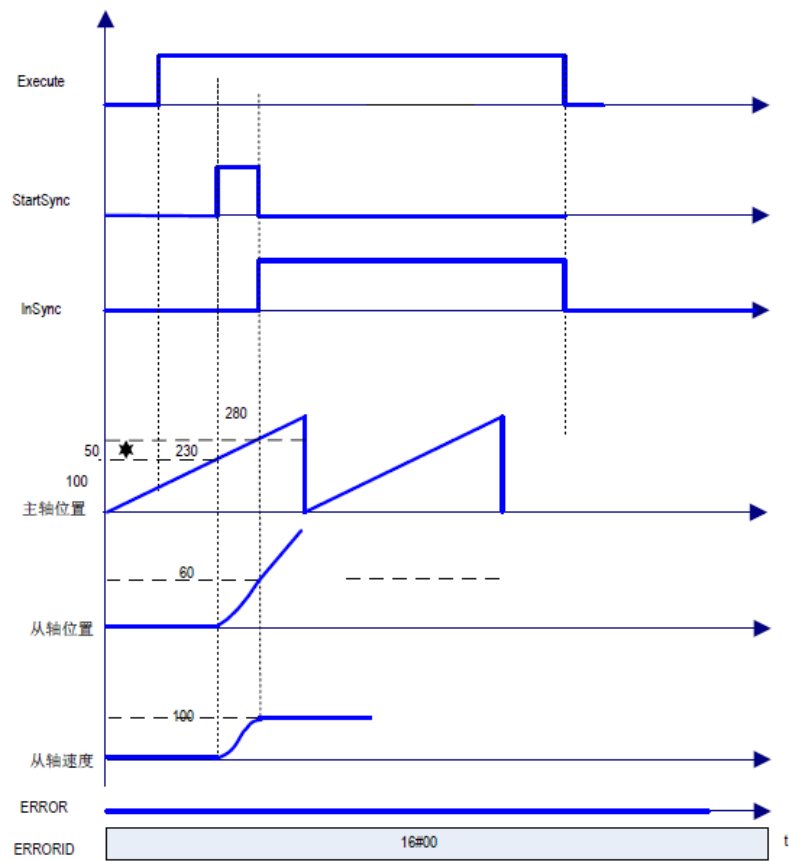
注意：如果主从轴工作在线性模式需保证需保证上述几个参数设置合理否则齿轮动作无法正确进行，故而建议使用该指令时主从轴工作在周期模式。

例如：主从轴线性工作模式都向正向运动，如果执行指令时主轴位置>MasterSyncPosition-MasterStartDistance，或者从轴位置> SlaveSyncPosition 则无法切入电子齿轮动作。

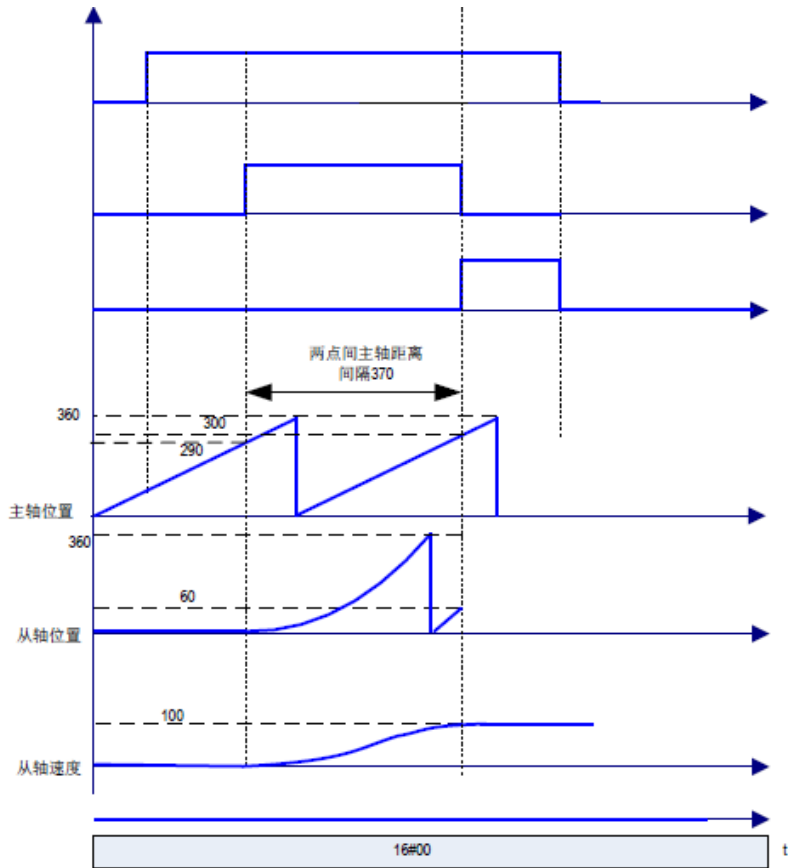
下面给出了几种不同参数下的样例时序图：

当主轴工作在周期模式（360 循环）、从轴工作在周期模式（360 循环）：

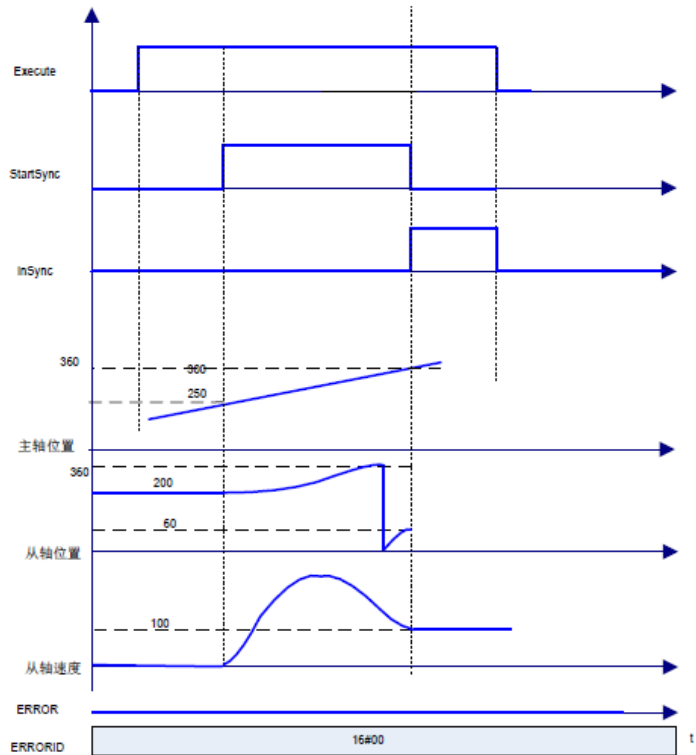
a) MasterSyncPosition=280、MasterStartDistance=50、SlaveSyncPosition=60，主轴速度为 50、AvoidReversal=FALSE



b) MasterSyncPosition=300、MasterStartDistance=370、SlaveSyncPosition=60，主轴速度为 50、AvoidReversal=FALSE



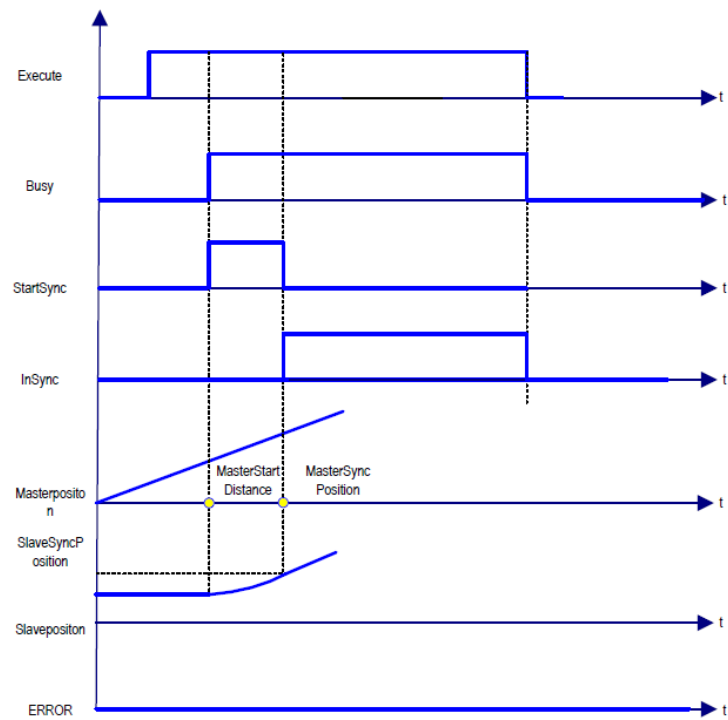
c) MasterSyncPosition=300、MasterStartDistance=50、SlaveSyncPosition=60，主轴速度为 50、AvoidReversal=FALSE、从轴起始位置大于 60



同步完成 (InSync 为 TRUE) 的同时达到目标速度，此后从轴移动量= 主轴移动量* RatioNumerator/RatioDenominator。

对于 AvoidReversal：如果从轴是模态轴并且主轴速度（齿轮比的倍数关系）不是相对于从轴的速度关系，那么 MC_GearInPos 会尝试着避免从轴的反转。它试图通过增加 5 个从轴周期“拉伸”从轴的运动。如果这个“拉伸”无效，那么会有错误出现并且从轴错误停止。如果从轴速度关联主轴速度（是齿轮比的倍数），那么会有错误出现，并且轴错误停止。如果从轴是线性轴模式轴，那么一个错误会产生在 Execute 输入上升沿处理时。

（4） 时序图



2.1.40 MC_Phasing

MC_Phasing：指定主从轴之间的相位偏差。

（1） 指令格式

指令	名称	图形表现	ST 表现
MC_Phasing	电子齿轮耦合断开		<pre>MC_Phasing(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

（2） 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Master	主轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例

输入输出变量	名称	数据类型	有效范围	初始值	描述
Slave	从轴	AXIS_REF			映射到轴，即 AXIS_REF_SM3 的一个实例

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Execute	执行	BOOL	TRUE, FALSE	FALSE	上升沿，触发功能块
PhaseShift	主从轴相位偏差值	LREAL		0	主从轴相位偏差值，正数代表从轴滞后
Velocity	速度	LREAL		0	执行相位偏移时最大速度值
Acceleration	加速度	LREAL		0	执行相位偏移时最大加速度值
Deceleration	减速度	LREAL		0	执行相位偏移时最大减速度值
Jerk	速度二次导数	LREAL		0	执行相位偏移时最大 Jerk 值

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成	BOOL	TRUE, FALSE	FALSE	True，从轴与主轴电子齿轮耦合断开
Busy	同步运行中	BOOL	TRUE, FALSE	FALSE	指令执行中。
CommandAborted	指令中断	BOOL	TRUE, FALSE	FALSE	被其它控制指令中断
Error	错误	BOOL	TRUE, FALSE	FALSE	如果检测到有错误，Error 位被置位；
ErrorID	错误代码	SMC_ERROR		0	发生异常时，输出错误代码

(3) 功能说明

Execute 上升沿执行相位偏移，从轴自动计算一条平滑曲线，完成从轴对主轴的相位偏移，主从轴相位差为输入信号的 PhaseShift 值，正值为从轴滞后于主轴。

完成偏移后 Done 信号输出为 True。

根据设定的 PhaseShift、Velocity、Acceleration、Deceleration 对主从轴相位差进行补偿。

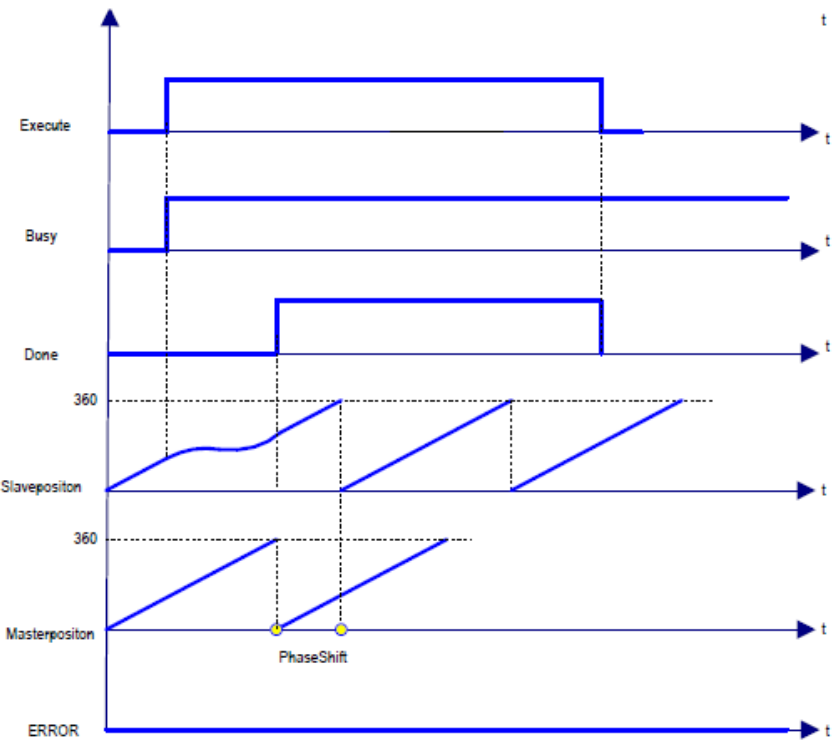
主从轴相位差达到 PhaseShift 时，Done 信号输出。

执行指令时主轴指令位置与反馈位置不变，从轴进行调整，完成后，从轴跟主轴之间相位差为 PhaseShift。

该指令最终结果为轴给定值之间的相位偏移，所以实轴的实际反馈值与最终的偏移可能不一致。

(4) 时序图

以主从轴都按 360 周期运动，Execute 信号上升沿执行调整，调整完成后从轴与主轴之间相位偏差为 PhaseShift 设定的值。



第3章 INVT 指令

3.1 通信指令

3.1.1 ModbusRTU 主站

ModbusRTU 主站指令库变量定义及使用

● 变量定义

变量从属模块	变量名称	类型		功能	注释
ModbusRTU_ Master_Init_COM1	Execute1	INPUT	BOOL	串口初始化功能激活	0: 非激活 1: 激活
	Baud1		DINT	波特率	例: 115200
	Databits1		INT	数据位	例: 8 位(无 7 位 ASCII)
	Stopbits1		INT	停止位	例: 1 停止位、2 停止位
	Parity1		INT	读写标志	0: 无校验 1: 偶校验 2: 奇校验
	Slave1		UINT	从站 ID	1-128
	Timeout1		DINT	超时时间	例: 1000
	bDone1	OUTPUT	BOOL	完成标志	0: 指令正在执行 1: 执行指令完成
	Error1		BOOL	错误标志	0: 无错误 1: 有错误
	ErrorID1		INT	错误码	见 ModbusRTU 错误码表
ModbusRTU_ Master_Fun_COM1	xExecute1	INPUT	BOOL	读写功能激活	0: 非激活 1: 激活
	Fun_Code1		INT	功能码	0x01、0x03、0x05、 0x06、0x0f、0x10
	Addr1		UINT	地址	0x0000~0xFFFF
	DataCount1		UINT	数目	读: 1~250 写: 1~240
	DataPtr1		POINTER TO INT	数据指针	指向读写数据存放的地址
	Error1	OUTPUT	BOOL	错误标志	0: 无错误 1: 有错误
	ErrorID1		INT	错误码	见 ModbusRTU 错误码表

串口 2 做 ModbusRTU_Master 主站时变量数量相同，变量名称后的数字由“1”改变成“2”，例如

“ModbusRTU_Master_Init_COM2”

● 使用教程

1) ModbusRTU_Master 主站连接从站设置

从属模块	设置项	功能	示例
ModbusRTU_Master_Init_COM1	Execute1	从站使能变量	Enable:=TRUE
	Baud1	波特率	Baud1:=19200
	Databits1	数据位	Port:=8
	Stopbits	停止位	Unit:=1
	Parity1	校验位	Parity1:=2
	Slave1	从站 ID	Slave1:=12
	Timeout1	超时时间	Delay Time:=1000

若定义将要连接的 ModbusRTU 从站时，参考上表 COM1 参数进行统一配置。参考示例（结构化文本 ST）如下：

```
1 PROGRAM ModbusRTU_Master
2 VAR
3   Execute1_1:BOOL:= FALSE; //RTU初始化使能
4   Baud1_1:DINT :=19200; //波特率
5   Databits1_1:INT :=8; //数据位
6   Stopbits1_1:INT :=1; //停止位
7   Parity1_1:INT :=2; //校验位 0:None Parity 1:Odd Parity 2:Even Parity
8   Slave1_1:BYTE := 2; //slave ID
9   Timeout1_1:DINT :=1000; //超时时间
10
11  xExecute1_1:BOOL := FALSE; //功能码使能
12  Fun_Code1_1:INT := 3; //功能码, 提示: 0x03读保持寄存器, 0x06写单个寄存器, 0x10写多个寄存器
13  Addr1_1:UINT := 4; //地址
14  DataCount1_1 : UINT := 10; //数目
15  DataPtr1_1:ARRAY [0..9]OF INT; //数据指针
16
17  ModbusRTU_Master_Init_COM1_1: ModbusRTU_Master_Init_COM1;
18  ModbusRTU_Master_Fun_COM1_1: ModbusRTU_Master_Fun_COM1;
19 END VAR
20
21 ModbusRTU_Master_Init_COM1_1{
22   Execute1:= Execute1_1,
23   Baud1:= Baud1_1,
24   Databits1:= Databits1_1,
25   Stopbits1:= Stopbits1_1,
26   Parity1:= Parity1_1,
27   Slave1:= Slave1_1,
28   Timeout1:= Timeout1_1,
29   bDone1=>,
30   Error1=>,
31   ErrorID1=> };
32
33 ModbusRTU_Master_Fun_COM1_1{
34   xExecute1:= xExecute1_1,
35   Fun_Code1:= Fun_Code1_1,
36   Addr1:= Addr1_1,
37   DataCount1:= DataCount1_1,
38   DataPtr1:= ADR(DataPtr1_1),
39   Error1=>,
40   ErrorID1=> };
```

图 3.1 ModbusRTU 主站连接从站参数配置示例

2) 完成 ModbuRTU_Master 主站连接从站相关参数配置后，对通信功能进行设置，设置参数如下表，设置参考示例如下图所示。

设置项	功能	示例
xExecute1	RTU 通信功能使能码	RW:=TRUE
Fun_Code1	功能码	Fun_Code1:=0x03
Addr1	读写寄存器开始地址	Addr:=2001
DataCount1	读写寄存器数量	Conut:=12
DataPtr1	读写数据存放区域地址指针	ADR (DATE_RTU1)

```
1  ModbusRTU_Master_Init_COM1_1(
2      Executel:= Executel_1,
3      Baud1:= Baud1_1,
4      Databits1:= Databits1_1,
5      Stopbits1:= Stopbits1_1,
6      Parity1:= Parity1_1,
7      Slavel:= Slavel_1,
8      Timeout1:= Timeout1_1,
9      bDone1=> ,
10     Error1=> ,
11     ErrorID1=> );
12
13  ModbusRTU_Master_Fun_COM1_1(
14     xExecutel:= xExecutel_1,
15     Fun_Code1:= Fun_Code1_1,
16     Addr1:= Addr1_1,
17     DataCount1:= DataCount1_1,
18     DataPtr1:= ADR(DataPtr1_1),
19     Error1=> ,
20     ErrorID1=> );
```

图 3.2 ModbusRTU 主站和从站通信参数设置示例

3.1.2 ModbusRTU 从站

ModbusRTU 从站库变量定义及使用

- 变量定义

变量从属模块	变量名称	类型		功能	注释
ModbusRTU_Slave1	Execute1	INPUT	BOOL	串口初始化功能激活	0: 非激活 1: 激活
	Baud1		DINT	波特率	例: 115200
	Databits1		INT	数据位	例: 8 位、7 位
	Stopbits1		INT	停止位	例: 1 停止位、2 停止位
	Parity1		INT	读写标志	0: 无校验 1: 偶校验 2: 奇校验
	Slave_Addr1		UINT	从站号	1~128
	Enable1		BOOL	读写功能激活	0: 非激活 1: 激活
	Done1	OUTPUT	BOOL	完成标志	0: 未完成 1: 完成
	ErrorID1		BYTE	错误码	见 ModbusRTU 错误码表

- 使用教程

1) 配置串口参数，建立 ModbusRTU 主站与从站连接

从属模块	设置项	功能	示例
ModbusRTU_Slave1	Execute1	从站使能变量	Enable:=TRUE
	Baud1	波特率	Baud1:=19200
	Databits1	数据位	Port:=8
	Stopbits	停止位	Unit:=1
	Parity1	校验位	Parity1:=2
	Timeout1	超时时间	Delay Time:=1000
	Slave_Addr1	从站号	Slave1:=12

按照 ModbusRTU 主站的串口配置参数，参考上表中的参数对从站进行设置。（此处的 Slave_Addr1 应和主站的 Slave1 对应）

2) ModbusRTU 主站和 ModbusRTU 从站进行读写数据通信

使能 Execute1 使 ModbusRTU 从站处于激活状态，若主站功能码为 0x03 读保持寄存器，若主站功能码为 0x10 写多个寄存器，可在变量区定义相应的存储区域，其大小不应小于 ModbusTCP 主站将要写入数据的大小，相应的主站功能码为 0x0F（写多个线圈）或者其他功能码时，操作与上述过程相同。

3.1.3 ModbusTCP 主站

ModbusTCP 主站指令库变量定义及使用

- 变量定义

变量名称	类型	功能	注释
Enable	INPUT	BOOL	ModbusTCP 功能激活 0: 非激活 1: 激活
IP		STRING	从站 IP 地址 例如: '192.168.1.13'
Port		DINT	从站端口号 例: 502
Unit		INT	从站单元号 非负整数
DelayTime		INT	回复延时时间 非负整数
Fun_Enable		BOOL	功能码使能 0: 非激活 1: 激活
fun_code		BYTE	功能码 0x03: 读多个寄存器模式 0x10: 写多个寄存器模式
Addr		UINT	读写寄存器地址 例: 2000、2001
Count		INT	读写寄存器数量 一次性读写寄存器数量最多 120 个
CoilSingleData		INT	写单个线圈 值为 0 或 1
BitPtr		POINTER TO BOOL	读写位数据指针 保存需要读写的位数据
DataPtr		POINTER TO INT	读写指针 存放读取到数据的位置信息或存放要写到寄存器的数据
Done	OUTPUT	BOOL	完成标志 0: 指令正在执行 1: 执行指令完成
Error		BOOL	错误标志 0: 无错误 1: 有错误

变量名称	类型	功能	注释
ErrorID	INT	错误码	见 ModbusTCP 错误码表

● 使用教程

1) ModbusTCP_Master 主站连接从站设置

在工程监控状态下设置将要连接的 ModbusTCP 从站的参数如下表：

设置项	功能	示例
Enable	从站使能变量	Enable:=TRUE
IP 地址	主站连接 ModbusTCP 从站的 IP 地址	IP := '192.168.1.13'
Port	主站连接 ModbusTCP 从站的端口号	Port:='502'
Unit	主站连接 ModbusTCP 从站的单元号	Unit:=3
Delay Time	功能启动超时时间	Delay Time:=1000

主站访问单个从站时，需将上述变量分别赋值。参考示例（功能块图 FDB 创建主程序）如下：

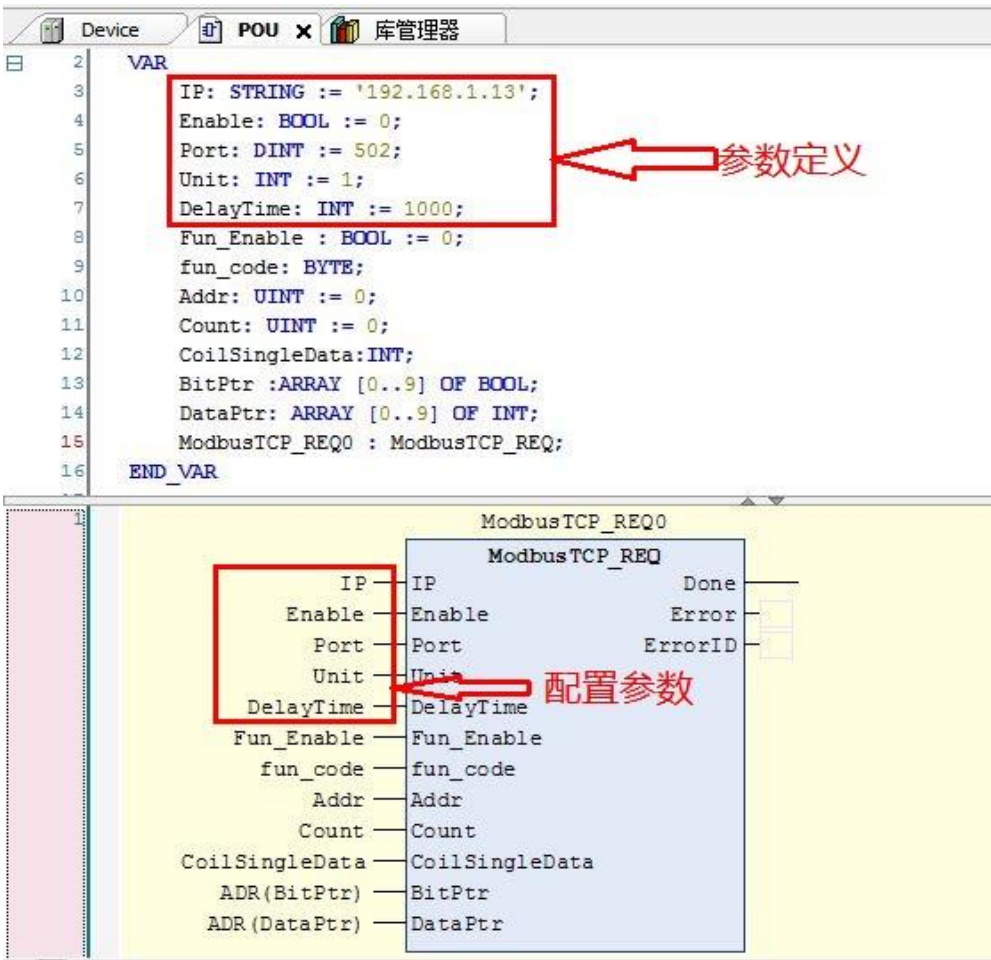


图 3.3 ModbusTCP 主站连接从站参数配置示例

上图中的功能块表示一个独立的 ModbusTCP 主站和从站连接，若添加新的 ModbusTCP 主站从站连接，可先创建一个新的功能块，按上图参数配置示例配置新的参数即可完成新的 ModbusTCP 主站从站连接。

2) 完成 ModbusTCP 主站连接从站相关参数配置后，对通信参数进行设置，设置参数如下表，设置参考示例如下图。

设置项	功能	示例
Fun_Enable	功能码使能开关	Fun_Enable:=TRUE
fun_code	读写多个寄存器线圈功能	Fun_code:=3
Addr	读写寄存器开始地址	Addr:=2001
Count	读写寄存器数量	Conut:=12
DataPtr	读写数据存放区域地址指针	ADR (DATE_TCP)

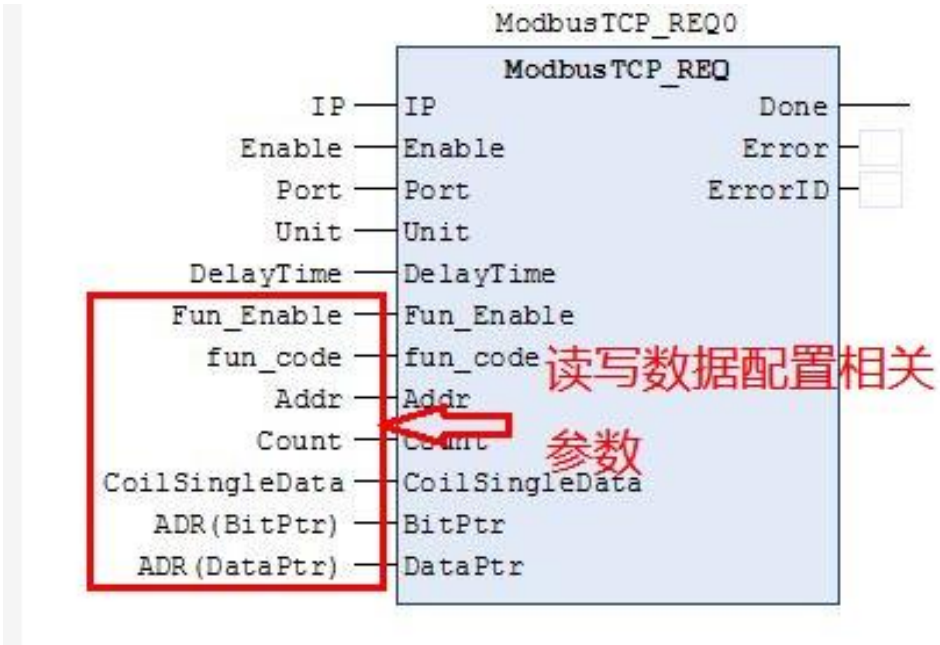


图 3.4 ModbusTCP_Master 主站和从站通信参数设置示例

上图中的每一个运算块表示一个 ModbusTCP 请求。图中定义了一个 ModbusTCP_Master 主站和从站的连接，第一个和第三个运算模块分别表示读不同从站的保持寄存器（0X03）操作，第二个和第四个运算模块表示写入不同从站寄存器内一定数目数据。

若想继续添加上述同一 ModbusTCP_Master 主站从站连接的不同通信要求，可先创建相同 ModbusTCP_Master 主站从站连接的运算块，再按图中示例更改通信参数即可实现新的通信请求。

3.1.4 ModbusTCP 从站

ModbusTCP 从站指令库变量定义及使用

● 变量定义

变量名称	类型		功能	注释
Enable	INPUT	BOOL	ModbusTCP_Slave 功能激活	0：非激活 1：激活
Port		DINT	从站端口号	采用默认值 502
Unit		INT	从站单元号	从站单元号（1-247）
Done	OUTPUT	BOOL	完成标志	0：指令正在执行 1：指令执行完成
IP		STRING	从站的 IP 地址	本机的 IP 地址（此处不可更改）
Error		BOOL	错误标志	0：无错误 1：有错误
ErrorID		INT	错误码	见 ModbusTCP 错误码表

● 使用说明

1) ModbusTCP 主站从 ModbusTCP_Slave 从站读取数据

使能 Enable 使 ModbusTCP_Slave 从站处于激活状态,若主站功能码为 0x03 读保持寄存器,首先设置 InputSize 的大小,并创建 InputSize 大小的数组用于存放主站要读取的数据,其次将数组的地址赋值给 Inputs 指针。相应的主站功能码为 0x01 (读线圈)时,操作与上述过程相同。

2) ModbusTCP 主站将数据写入 ModbusTCP_Slave 从站数据

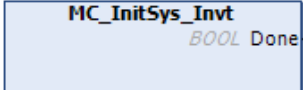
使能 Enable 使 ModbusTCP_Slave 从站处于激活状态,若主站功能码为 0x10 写多个寄存器,可在变量区定义相应的存储区域,其大小不应小于 ModbusTCP 主站将要写入数据的大小,相应的主站功能码为 0x0F (写多个线圈)或者其他功能码时,操作与上述过程相同。

3.2 脉冲指令

3.2.1 MC_InitSys_Invt

MC_InitSys_Invt: 运控模块轴初始化指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_InitSys_Invt	轴初始化指令		MC_InitSys_Invt_0(Done=>);

(2) 相关变量

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Mode	执行条件	BOOL	TRUE, FALSE	TRUE	TRUE-脉冲型, FALSE-总线型(修改后必须重启控制器,否则运行会发生未知状况)
Period	执行条件	UINT	—	1	运行周期,脉冲必须为 1,总线型必须大于 0(修改后必须重启控制器,否则运行会发生未知状况)

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 初始化完成,FALSE 初始化未完成
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	—	0	默认 0, >100 警告 ID, <100 报错 ID

(3) 功能说明

将运控模块进行初始化,当 Done 输出 TRUE 时,表示初始化完成,反之则未完成初始化。

3.2.2 MC_Axis

MC_Axis: 轴定义指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Axis	轴定义指令		<pre> MC_Axis(AxisID:= , MaxVelocity:= , MaxAcceleration:= , MaxHomeSpeed:= , MaxVim2Speed:= , MaxVim1Speed:= , MaxJogSpeed:= , MaxDeceleration:= , LimitEnable:= , MaxPLimit:= , MaxNLimit:= , PulseData:= , DistanceData:= , MaxJerkTime:= , Mode:= , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
AxisID	轴 ID	BYTE	0~3	0	当前轴对应 ID(0-3)
MaxVelocity	轴运行最大速度	LREAL	1~5000	2000	当前轴最大运行速度, mm/min
MaxAcceleration	轴运行最大加速度	LREAL	正数或“0”	1800000	当前轴最大允许加速度, mm/min ²
MaxHomeSpeed	轴运行最大回零速度	LREAL	1~5000	1000	当前轴最大回零速度, mm/min
MaxVim2Speed	第二阶段最大回零速度	LREAL	1~50	20	当前轴第二阶段最大回零速度, mm/min
MaxVim1Speed	第一阶段最大回零速度	LREAL	50~600	200	当前轴第一阶段最大回零速度, mm/min
MaxJogSpeed	轴运行最大手动速度	LREAL	1~5000	1000	当前轴最大手动速度, mm/min
MaxDeceleration	轴运行最大减速度	LREAL	正数或“0”	1800000	当前轴最大允许减速度, mm/min ²
LimitEnable	软限位使能标志	BOOL	TRUE, FALSE	TRUE	软限位使能标志 TRUE-软限位开启; FALSE-软限位关闭(单轴速度模式不考虑软限位)
MaxPLimit	最大正限位位置	LREAL	正数或“0”	999999.999	当前轴最大正限位位置, mm
MaxNLimit	最大负限位位置	LREAL	负数或“0”	-999999.999	当前轴最大负限位位置, mm

输入变量	名称	数据类型	有效范围	初始值	描述
PulseData	当前轴旋转一圈所需脉冲数	UDINT	正数或“0”	10000	当前轴运行一转需要脉冲数，pulse
DistanceData	当前轴运行一转运行距离	LREAL	正数或“0”	10	当前轴运行一转运行距离，mm
MaxJerkTime	当前轴最大允许加速度变化时间	UINT	10~400	100	当前轴最大允许加速度变化时间
Mode	当前轴脉冲控制方式	USINT	0~2	2	当前轴脉冲控制方式：0--脉冲+符号方式；1：正/反转脉冲列方式；2：正交编码脉冲方式；

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	—	0	默认 0, >100 警告 ID, <100 报错 ID

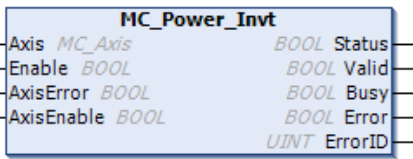
(3) 功能说明

声明轴 ID 及其控制参数。

3.2.3 MC_Power_Invt

MC_Power_Invt: 轴使能指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Power_Invt	轴使能指令		<pre> MC_Power_Invt(Axis:= , Enable:= , AxisError:= , AxisEnable:= , Status=> , Valid=> , Busy=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位,高电平触发
AxisError	轴报警标志	BOOL	TRUE, FALSE	FALSE	设置为 TRUE 轴报警, FALSE 轴

输入变量	名称	数据类型	有效范围	初始值	描述
					未报警
AxisEnable	轴使能标志	BOOL	TRUE, FALSE	FALSE	设置为 TRUE 轴使能完成, FALSE 轴未使能

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Status	可运行	BOOL	TRUE, FALSE	FALSE	TRUE 轴已经准备好; FALSE 轴未准备好
Valid	轴使能标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴使能有效; FALSE 轴使能无效
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于运控模块轴使能, 在控制过程中会只有轴未发生报警而且伺服使能完成后运动模块才会使用该轴进行相关的运控功能开发; 通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.4 MC_Stop_Invt

MC_Stop_Invt: 轴停止指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Stop_Invt	轴停止指令		<pre> MC_Stop_Invt(Axis:= , Enable:= , Deceleration:= , JerkTime:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Deceleration	减速度	LREAL	正数或“0”	1800000	减速度, mm/min ² (当减速度设置不合理时, 将会使用轴控功能的 Deceleration)
JerkTime	加速度变化时	UINT	0~400	0	加速度变化时间(周期数)

输入变量	名称	数据类型	有效范围	初始值	描述
	间				
Mode	减速停止方式	BOOL	TRUE, FALSE	TRUE	TRUE--减速停; FALSE--立即停

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 初始化完成,FALSE 初始化未完成
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于停止功能块, 在控制过程支持单轴停; 支持 T 型加减速和 S 型加减速速度规划, JerkTime>0 对应 S 型加减速规划, JerkTime=0 对应 T 型加减速规划; 支持减速停和立即停两种控制模式。

3.2.5 MC_Reset_Invt

MC_Reset_Invt: 轴复位指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_Reset_Invt	轴复位指令		<pre>MC_Reset_Invt(Axis:= , Enable:= , Done=> , Busy=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 初始化完成,FALSE 初始化未完成
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于单轴控制，用于轴复位控制模块，当任何情况下发生 STOP 功能后，必须通过此模块进去 STOP 状态的消除功能。

3.2.6 MC_JOG_Invt

MC_JOG_Invt: JOG 运动指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_JOG_Invt	JOG 运动指令		<pre> MC_JOG_Invt(Axis:= , JogDirection:= , RunStart:= , Velocity:= , Acceleration:= , Deceleration:= , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
JogDirection	运动方向	BOOL	TRUE, FALSE	FALSE	TRUE->正向运动, FALSE->负向运行
RunStart	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Velocity	速度	LREAL	正数或“0”	500	JOG 运动的速度, 单位 mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, 单位 mm/min ²

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	TRUE 被其它命令打断;FALSE 未被打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

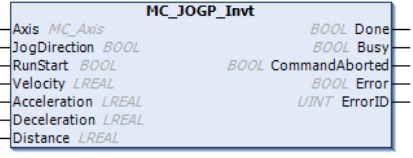
(3) 功能说明

用于单轴进行 JOG 运动，可通过该模块进行运动轴的速度、加速度和减速度进行设置；可通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.7 MC_JOGP_Invt

MC_JOGP_Invt: 轴点动指令

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_JOGP_Invt	点动运动指令		<pre> MC_JOGP_Invt(Axis:=, JogDirection:=, RunStart:=, Velocity:=, Acceleration:=, Deceleration:=, Distance:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
JogDirection	运动方向	BOOL	TRUE, FALSE	FALSE	TRUE->正向运动, FALSE->负向运行
RunStart	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Velocity	速度	LREAL	正数或“0”	500	点位运动的速度, mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/min ²
Distance	移动距离	LREAL	数据范围	0.5	移动距离, mm

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 初始化完成, FALSE 初始化未完成
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	TRUE 被其它命令打断; FALSE 未被打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

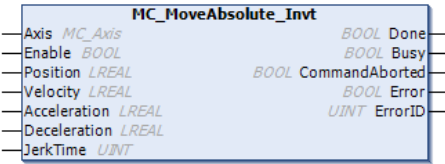
(3) 功能说明

用于单轴进行点位运动, 可通过该模块进行运动轴的速度、加速度和减速度进行设置; 通过点动功能可以将轴定位到 0.001mm; 可通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.8 MC_MoveAbsolute_Invt

MC_MoveAbsolute_Invt: 单轴绝对位置控制指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_MoveAbsolute_Invt	单轴绝对位置指令		<pre> MC_MoveAbsolute_Invt(Axis:= , Enable:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , JerkTime:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Position	距离	LREAL	数据范围	0	距离
Velocity	速度	LREAL	正数	1000	速度,mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/s ²
JerkTime	加速度变化时间	UINT	0~400	0	加速度变化时间(周期数)

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 初始化完成,FALSE 初始化未完成
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	TRUE 被其它命令打断;FALSE 未被打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于单轴绝对位置控制, 支持 T 型加减速和 S 型加减速速度规划, JerkTime>0 对应 S 型加减速规划, JerkTime=0 对应 T 型加减速规划; 通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.9 MC_MoveRelative_Invt

MC_MoveRelative_Invt: 单轴相对位置控制指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ MoveRelative _Invt	单轴相对位置指令		<pre> MC_MoveRelative_Invt(Axis:= , Enable:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , JerkTime:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Position	距离	LREAL	数据范围	0	距离
Velocity	速度	LREAL	正数	1000	速度, mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/s ²
JerkTime	加速度变化时间	UINT	0~400	0	加速度变化时间(周期数)

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 初始化完成, FALSE 初始化未完成
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	TRUE 被其它命令打断; FALSE 未被打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

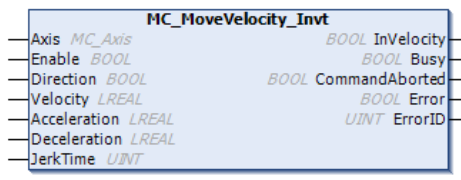
(3) 功能说明

用于单轴相对位置控制, 支持 T 型加减速和 S 型加减速速度规划, JerkTime>0 对应 S 型加减速规划, JerkTime=0 对应 T 型加减速规划; 通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.10 MC_MoveVelocity_Invt

MC_MoveVelocity_Invt: 单轴速度控制指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_MoveVelocity_Invt	单轴速度控制指令		<pre> MC_MoveVelocity_Invt(Axis:=, Enable:=, Direction:=, Velocity:=, Acceleration:=, Deceleration:=, JerkTime:=, InVelocity=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Direction	运动方向	BOOL	TRUE, FALSE	TRUE	方向(TRUE-正向, FALSE-负向)
Velocity	速度	LREAL	正数或“0”	500	速度, mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/s ²
JerkTime	加速度变化时间	UINT	0~400	0	加速度变化时间(周期数)

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
InVelocity	速度值第一次是否到达标志	BOOL	TRUE, FALSE	FALSE	TRUE: 速度值第一次达到
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	TRUE 被其它命令打断;FALSE 未被打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于单轴速度控制, 在控制过程不需要回零完成标志, 支持 T 型加减速和 S 型加减速速度规划, JerkTime>0 对应 S 型加减速规划, JerkTime=0 对应 T 型加减速规划; 通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.11 MC_ReadActualPosition_Invt

MC_ReadActualPosition_Invt: 轴实际位置读取指令

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadActualPosition_Invt	轴实际位置 读取指令		<pre>MC_ReadActualPosition_Invt(Axis:= , Enable:= , Busy=> , Value=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
Value	位置值	LREAL	数据范围	0	位置参数返回值
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

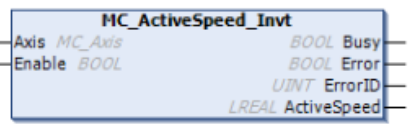
(3) 功能说明

用于轴实际位置的读取, 由 Axis 决定获取那个轴, Value 返回轴实际的坐标位置。

3.2.12 MC_ActiveSpeed_Invt

MC_ActiveSpeed_Invt: 轴实际速度读取指令

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ActiveSpeed_Invt	轴实际速度 读取指令		<pre>MC_ActiveSpeed_Invt(Axis:= , Enable:= , Busy=> , Error=> , ErrorID=> , ActiveSpeed=>);</pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 高电平触发

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中;FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错
ActiveSpeed	速度值	LREAL	数据范围	0	速度参数返回值

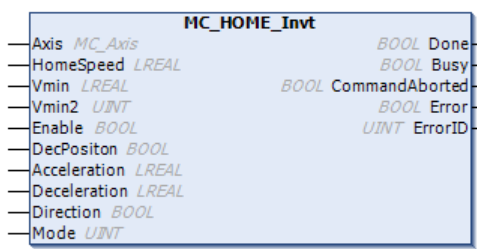
(3) 功能说明

用于轴实际速度的读取, 由 Axis 决定获取那个轴, ActiveSpeed 返回轴实际的运行速度。

3.2.13 MC_HOME_Invt

MC_HOME_Invt: 轴回零指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_HOME_Invt	轴回零指令		<pre> MC_HOME_Invt(Axis:=, HomeSpeed:=, Vmin:=, Vmin2:=, Enable:=, DecPositon:=, Acceleration:=, Deceleration:=, Direction:=, Mode:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
HomeSpeed	回零速度	LREAL	正数	800	回零速度
Vmin	速度 1	LREAL	50~600	200	在减速挡块运行速度
Vmin2	速度 2	LREAL	1~50	30	等待电机 Z 信号运行速度
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
DecPositon	减速挡块信号	BOOL	TRUE, FALSE	FALSE	减速挡块信号

输入变量	名称	数据类型	有效范围	初始值	描述
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/min ²
Direction	回零方向	BOOL	TRUE, FALSE	TRUE	回零方向(TRUE-正方向; FALSE-负方向)
Mode	回零模式	UINT	-	1	回零模式(暂时不起作用, 用于后续扩充)

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 找到机械原点, FALSE 未找到
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错
ActiveSpeed	速度值	LREAL	数据范围	0	速度参数返回值

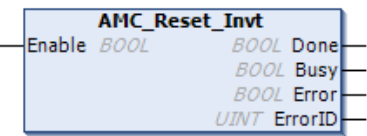
(3) 功能说明

用于轴寻找机械零点控制功能, 回零模式可根据机械结构进行定制功能, 目前暂时只支持一种回零模式; 支持四个轴同时回零; 可通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.14 AMC_Reset_Invt

AMC_Reset_Invt: 多轴复位指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
AMC_Reset_Invt	多轴复位指令		<pre> AMC_Reset_Invt(Enable:= , Done=> , Busy=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 找到机械原点, FALSE 未找到
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

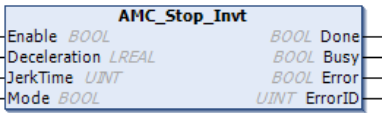
(3) 功能说明

用于多轴控制，用于轴复位控制模式，当任何情况下发生 STOP 功能后，必须通过此模块解除 STOP 状态功能。

3.2.15 AMC_Stop_Invt

AMC_Stop_Invt: 多轴停止指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
AMC_Stop_Invt	多轴停止指令		<pre> AMC_Stop_Invt(Enable:= , Deceleration:= , JerkTime:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位，上升沿触发
Deceleration	减速度	LREAL	正数或“0”	1800000	减速度，mm/min ² (当减速度设置不合理时，将会使用轴控功能的 Deceleration)
JerkTime	加速度变化时间	UINT	0~400	0	加速度变化时间（周期数）
Mode	停止模式	BOOL	TRUE, FALSE	TRUE	TRUE--减速停；FALSE--立即停

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	TRUE 轴已停止；FALSE 轴减速停止中
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中；FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错，FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0，>100 警告，<100 报错

(3) 功能说明

用于停止功能块控制，在控制过程支持指定轴停；支持 T 型加减速和 S 型加减速速度规划，JerkTime>0 对应 S 型加减速规划，JerkTime=0 对应 T 型加减速规划；支持减速停和立即停两种控制模式。

3.2.16 AMC_ActiveSpeed_Invt

AMC_ActiveSpeed_Invt: 多轴合成速度读取指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
AMC_ActiveSpeed_Invt	多轴合成速度读取指令		<pre>AMC_ActiveSpeed_Invt(Enable:= , Busy=> , Error=> , ErrorID=> , ActiveSpeed:=);</pre>

(2) 相关变量

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 高电平触发

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错
ActiveSpeed	速度值	LREAL	数据范围	0	轴对应的实际速度

(3) 功能说明

读取指定轴的实际速度, 由 ActiveSpeed 返回轴的实际速度值。

3.2.17 AMC_MoveAbsolute_Invt

AMC_MoveAbsolute_Invt: 多轴绝对位置移动指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
AMC_MoveAbsolute_Invt	多轴绝对位置移动指令		<pre>AMC_MoveAbsolute_Invt(Axis1_Use:= , Axis2_Use:= , Axis3_Use:= , Axis4_Use:= , Enable:= , Position1:= , Position2:= , Position3:= , Position4:= , Velocity:= , Acceleration:= , Deceleration:= , JerkTime:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);</pre>

(2) 相关变量

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Axis1_Use	轴 1 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 1 使用标志, TRUE-使用, FALSE-未使用
Axis2_Use	轴 2 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 2 使用标志, TRUE-使用, FALSE-未使用
Axis3_Use	轴 3 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 3 使用标志, TRUE-使用, FALSE-未使用
Axis4_Use	轴 4 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 4 使用标志, TRUE-使用, FALSE-未使用
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Position1	轴 1 距离	LREAL	数据范围	0	轴 1 距离
Position2	轴 2 距离	LREAL	数据范围	0	轴 2 距离
Position3	轴 3 距离	LREAL	数据范围	0	轴 3 距离
Position4	轴 4 距离	LREAL	数据范围	0	轴 4 距离
Velocity	速度	LREAL	正数或“0”	500	速度, mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/min ²
JerkTime	加速变化时间	UINT	0~400	0	加速度变化时间(周期数)

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	完成标志, TRUE-运行到指定位置; FALSE-未运行到指定位置
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	TRUE 被其它命令打断; FALSE 未被打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

多轴绝对位置模式支持四轴同时运动, 绝对位置由 Position 指定, 设置相应的运动方向, 速度, 加速度等, 使 Enable 由 FALSE→TRUE 启动功能块, 指定指令轴由当前位置运动到绝对位置处。

多轴绝对位置模式与单轴绝对位置模式原理相同, 多轴绝对位置模式可以控制多个轴同时进行绝对位置移动。

多轴绝对位置模式加减速支持 T 型加减速与 S 型加减速。

3.2.18 AMC_MoveRelative_Invt

AMC_MoveRelative_Invt: 多轴相对位置移动指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
AMC_MoveRelative_Invt	多轴相对位置移动指令		<pre> AMC_MoveRelative_Invt(Axis1_Use:= , Axis2_Use:= , Axis3_Use:= , Axis4_Use:= , Enable:= , Position1:= , Position2:= , Position3:= , Position4:= , Velocity:= , Acceleration:= , Deceleration:= , JerkTime:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Axis1_Use	轴 1 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 1 使用标志, TRUE-使用, FALSE-未使用
Axis2_Use	轴 2 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 2 使用标志, TRUE-使用, FALSE-未使用
Axis3_Use	轴 3 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 3 使用标志, TRUE-使用, FALSE-未使用
Axis4_Use	轴 4 使用标志	BOOL	TRUE, FALSE	FALSE	实际轴 4 使用标志, TRUE-使用, FALSE-未使用
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Position1	轴 1 距离	LREAL	数据范围	0	轴 1 距离
Position2	轴 2 距离	LREAL	数据范围	0	轴 2 距离
Position3	轴 3 距离	LREAL	数据范围	0	轴 3 距离
Position4	轴 4 距离	LREAL	数据范围	0	轴 4 距离
Velocity	速度	LREAL	正数或“0”	500	速度, mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/min ²
JerkTime	加速变化时间	UINT	0~400	0	加速度变化时间(周期数)

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	完成标志, TRUE-运行到指定位置; FALSE-未运行到指定位置
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功

输出变量	名称	数据类型	有效范围	初始值	描述
					能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	被其它命令打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

多轴相对位置模式支持四轴同时运动, 相对位置由 Position 指定, 设置相应的运动方向, 速度, 加速度等, 使 Enable 由 FALSE→TRUE 启动功能块, 指定指令轴由当前位置运动到相对位置处。

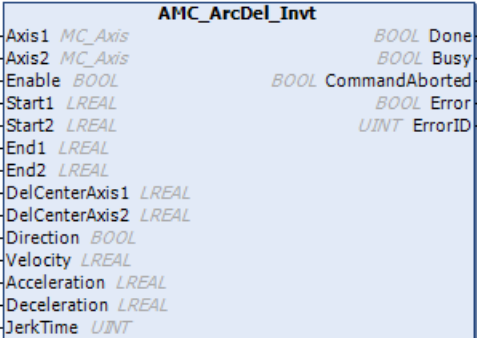
多轴相对位置模式与单轴相对位置模式原理相同, 多轴相对位置模式可以控制多个轴同时进行相对位置移动。

多轴相对位置模式加减速支持 T 型加减速与 S 型加减速。

3.2.19 AMC_ArcDel_Invt

AMC_ArcDel_Invt: 平面圆弧插补指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
AMC_ArcDel_Invt	平面圆弧插补指令		<pre> AMC_ArcDel_Invt(Axis1:=, Axis2:=, Enable:=, Start1:=, Start2:=, End1:=, End2:=, DelCenterAxis1:=, DelCenterAxis2:=, Direction:=, Velocity:=, Acceleration:=, Deceleration:=, JerkTime:=, Done=>, Busy=>, CommandAborted=>, Error=>, ErrorID=>); </pre>

(2) 相关变量

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Axis1	轴 1	MC_AXIS	-	-	指令轴 1 号码[0..3]及轴控制设置
Axis2	轴 2	MC_AXIS	-	-	指令轴 2 号码[0..3]及轴控制设置
Enable	执行条件	BOOL	TRUE, FALSE		功能块的使能位, 上升沿触发
Start1	轴 1 起点位置	LREAL	数据范围	0	轴 1 起点位置
Start2	轴 2 起点位置	LREAL	数据范围	0	轴 2 起点位置
End1	轴 1 终点位置	LREAL	数据范围	0	轴 1 终点位置
End2	轴 2 终点位置	LREAL	数据范围	0	轴 2 终点位置
DelCenterAxis1	轴 1 起点到圆心的距离	LREAL	数据范围	0	轴 1 起点到圆心的距离

输入变量	名称	数据类型	有效范围	初始值	描述
DelCenterAxis2	轴 2 起点到圆心的距离	LREAL	数据范围	0	轴 2 起点到圆心的距离
Direction	运行方向	BOOL	TRUE, FALSE	FALSE	TRUE--顺时针 FALSE--逆时针
Velocity	速度	LREAL	正数或“0”	500	速度, mm/min
Acceleration	加速度	LREAL	正数或“0”	1800000	加速度, mm/min ²
JerkTime	加速变化时间	UINT	0~400	0	加速度变化时间(周期数)

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	指令执行完成	BOOL	TRUE, FALSE	FALSE	完成标志, TRUE-运行到指定位置; FALSE-未运行到指定位置
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
CommandAborted	命令是否被打断标志	BOOL	TRUE, FALSE	FALSE	被其它命令打断
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于平面内圆弧插补; 支持 T 型加减速和 S 型加减速速度规划, JerkTime>0 对应 S 型加减速规划, JerkTime=0 对应 T 型加减速规划; 支持顺时针和逆时针运行; 使用此模块需要制定起点位置和终点位置以及起点位置到圆心的距离; 只允许 XY/XZ/ZY 三个平面进行圆弧插补, 涉及到附加轴圆弧插补将报错处理; 通过 OUTPUT 返回功能块运行状态、是否发生错误和错误信息。

3.2.20 MC_SendData_Invt

MC_SendData: 将数据发送到底层指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_SendData_Invt	发数模块指令		<pre>MC_SendData_Invt_0(Mode:= , Axis1Pulse=> , Axis2Pulse=> , Axis3Pulse=> , Axis4Pulse=> , Axis1Direction=> , Axis2Direction=> , Axis3Direction=> , Axis4Direction=>);</pre>

(2) 相关变量

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Mode	执行条件	BOOL	TRUE, FALSE	TRUE	TRUE-脉冲型, FALSE-总线型

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Axis1Pulse	轴 1 脉冲反馈	UINT	-	-	轴 1 脉冲反馈
Axis2Pulse	轴 2 脉冲反馈	UINT	-	-	轴 2 脉冲反馈
Axis3Pulse	轴 3 脉冲反馈	UINT	-	-	轴 3 脉冲反馈
Axis4Pulse	轴 4 脉冲反馈	UINT	-	-	轴 4 脉冲反馈
Axis1Direction	轴 1 方向	BOOL	TRUE, FALSE	-	轴 1 方向: TRUE-负方向; FALSE-正方向
Axis2Direction	轴 2 方向	BOOL	TRUE, FALSE	-	轴 2 方向: TRUE-负方向; FALSE-正方向
Axis3Direction	轴 3 方向	BOOL	TRUE, FALSE	-	轴 3 方向: TRUE-负方向; FALSE-正方向
Axis4Direction	轴 4 方向	BOOL	TRUE, FALSE	-	轴 4 方向: TRUE-负方向; FALSE-正方向

(3) 功能说明

用于轴控制数据转化为脉冲数据，并将脉冲数据发送到 FPGA 处理过程。

3.2.21 MC_GetVersion_Invt

MC_GetVersion_Invt: 获取当前运动控制库版本号指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_GetVersion_Invt	获取运动控制库版本号指令		MC_GetVersion_Invt(Version=>);

(2) 相关变量

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Version	运动控制库版本号	STRING	-	-	当前运动控制库版本号

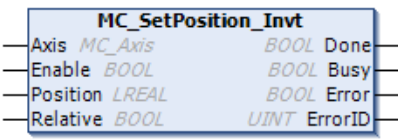
(3) 功能说明

用于获取当前运动控制库版本号，由 Version 返回版本号值。

3.2.22 MC_SetPosition_Invt

MC_SetPosition_Invt: 轴位置设定指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_SetPosition_Invt	轴位置设定指令		<pre> MC_SetPosition_Invt(Axis:= , Enable:= , Position:= , Relative:= , Done=> , Busy=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
Position	位置	LREAL	正数或“0”	0	位置
Relative	逻辑地址相对位置标志	BOOL	TRUE, FALSE	FALSE	FALSE: 逻辑地址的绝对位置; TRUE: 逻辑地址的相对位置

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Done	完成标志	BOOL	TRUE, FALSE	FALSE	完成标志
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	默认 0, >100 警告, <100 报错

(3) 功能说明

用于轴实际位置的设置功能, 由 Axis、Position 和 Relative 决定。

3.2.23 MC_ReadParameter_Invt

MC_ReadParameter_Invt: 读取轴参数指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadParameter_Invt	读取轴参数指令		<pre> MC_ReadParameter_Invt(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Value=> , Error=> , ErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 上升沿触发
ParameterNumber	参数号	DINT	-	100	参数号

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	执行有效标志	BOOL	TRUE, FALSE	FALSE	执行有效标志
Value	参数返回值	LREAL	-	0	参数返回值
Error	轴出错标志	BOOL	TRUE, FALSE	FALSE	轴出错标志: TRUE 轴出错, FALSE 轴未出错
ErrorID	错误码	UINT	-	0	错误码: 默认 0; >100 警告; <100 报错

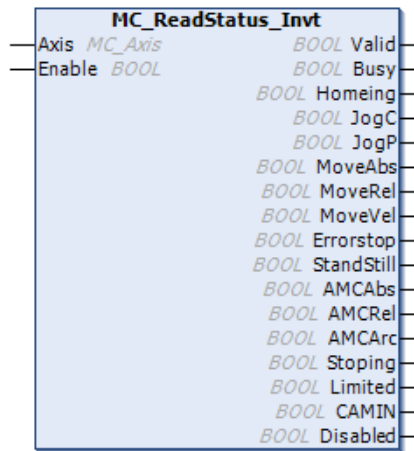
(3 功能说明

读取指定轴的参数。

3.2.24 MC_ReadStatus_Invt

MC_ReadStatus_Invt: 读取轴动作指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
MC_ReadStatus_Invt	读取轴动作指令		<pre> MC_ReadStatus_Invt(Axis:= , Enable:= , Valid=> , Busy=> , Homeing=> , JogC=> , JogP=> , MoveAbs=> , MoveRel=> , MoveVel=> , Errorstop=> , StandStill=> , AMCAbs=> , AMCRel=> , AMCArc=> , Stopping=> , Limited=> , Disabled=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	MC_Axis	-	-	指令轴号码[0..3]及轴控制设置

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
Enable	执行条件	BOOL	TRUE, FALSE	FALSE	功能块的使能位, 高电平触发

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
Valid	执行有效标志	BOOL	TRUE, FALSE	FALSE	执行有效标志
Busy	执行中	BOOL	TRUE, FALSE	FALSE	TRUE 功能块使能中; FALSE 功能块未使能
Homeing	单轴回零状态标志	BOOL	TRUE, FALSE	FALSE	TRUE 单轴回零运行中; FALSE 未发生单轴回零操作
JogC	单轴手动运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 单轴手动连续运行中; FALSE 未发生单轴手动连续操作
JogP	单轴点动运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 单轴手动点动运行中; FALSE 未发生单轴手动点动操作
MoveAbs	单轴绝对位置运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 单轴绝对位置运行中; FALSE 未发生单轴绝对位置操作
MoveRel	单轴相对位置运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 单轴相对位置运行中; FALSE 未发生单轴相对位置操作
MoveVel	单轴速度运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 单轴速度运行中; FALSE 未发生单轴速度操作
Errorstop	报警标志	BOOL	TRUE, FALSE	FALSE	TRUE 报警停止; FALSE 未报警
StandStill	保持静止标志	BOOL	TRUE, FALSE	FALSE	TRUE 保持静止; FALSE 运行中
AMCAbs	多轴绝对位置运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 多轴绝对位置运行中; FALSE 未发生多轴绝对位置操作
AMCRel	多轴相对位置运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 多轴相对位置运行中; FALSE 未发生多轴相对位置操作
AMCArc	多轴圆弧插补运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 多轴圆弧插补运行中; FALSE 未发生多轴圆弧插补操作
Stoping	减速停止运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 减速停止运行中; FALSE 未发生减速停止操作
Limited	限位标志	BOOL	TRUE, FALSE	FALSE	TRUE 限位中; FALSE 未发生限位(限位发生后复位为 FALSE)
CAMIN	电子凸轮运行标志	BOOL	TRUE, FALSE	FALSE	TRUE 电子凸轮运行中; FALSE 未发生电子凸轮操作
Disabled	轴是否可用标志	BOOL	TRUE, FALSE	FALSE	TRUE 轴不可用; FALSE 轴可用

(3) 功能说明

读取指定轴的动作。

3.3 增强指令

3.3.1 FB_ReinitDrv

FB_ReinitDrv: 伺服断线自动重连指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
FB_ReinitDrv	伺服断线自动重连指令		<pre> FB_ReinitDrv(apAxis:= , pEtherCATMasterSoftMotion:= , bEnable:= , bForceReinitETC:= , bForceReinitDrv:= , bReinitSuccess=> , iAxisCnt=> , bBusy=> , bError=> , iErrorID=>); </pre>

(2) 相关变量

✧ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
apAxis	轴指针数组	ARRAY[1..16] OF POINTER TO AXIS_REF_SM3	-	NULL	16 轴指针数组, 可支持最多 16 个轴指针赋值, 未赋值指针默认指向地址 0
pEtherCATMasterSoftMotion	EtherCAT 主站指针	POINTER TO IODrvEtherCAT	-	NULL	EtherCAT 主站指针

✧ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	使能	BOOL	TRUE, FALSE	FALSE	True: 使能功能块执行, False: 功能块失能
bForceReinitEtherCAT	强制 EtherCAT 主站复位	BOOL	TRUE, FALSE	FALSE	上升沿强制 EtherCAT 通讯复位, 用于当功能块未检测到轴故障时手动强制通讯复位

✧ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bReinitSuccess	复位成功	BOOL	TRUE, FALSE	FALSE	True: 轴复位通讯成功, 当前为可操作状态, False: 轴复位通讯失败
iAxisCnt	轴数	INT	1~16	0	发现已连接的轴总数
bBusy	正在执行	BOOL	TRUE, FALSE	FALSE	True: 功能块正在执行操作,

输出变量	名称	数据类型	有效范围	初始值	描述
					False: 功能块未工作
bError	错误	BOOL	TRUE, FALSE	FALSE	功能块执行过程中发生错误
iErrorID	错误代码	INT	-	0	错误码

(3) 功能说明

此功能块为伺服断线自连模块，用于实现对伺服通讯状态的实时检测以及在通讯断开后自动重新连接的功能，通讯成功后返回相关信息并清除伺服报警。本功能块最多可支持 16 个轴输入。

3.3.2 FB_AxisEasyControl

FB_AxisEasyControl: 单轴易用运动功能块指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
FB_AxisEasyControl	单轴易用运动功能块指令		<pre> FB_AxisEasyControl(axisEasyControl:=, bEnable:=, bPower:=, eHomeType:=, bHomeSignal:=, eRunMode:=, bRun:=, bReset:=, bReinit:=, IrPositionOrDistance:=, IrVelocity:=, IrAcceleration:=, IrDeceleration:=, eDirection:=, IrJerk:=, bContinueJog:=, bJogFwd:=, bJogBwd:=, bStop:=, tTimeOut:=, bDone=>, bBusy=>, bError=>, eErrorID=>); </pre>

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
axisEasyControl	轴	AXIS_REF_SM3	-	-	映射到轴

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bEnable	使能指令	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 指令开始执行
bPower	上电	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 上电
eHomeType	回零方式	MoveCtrl_HomeType	数据范围	0	设置回零方式
bHomeSignal	回零信号	BOOL	TRUE,FALSE	FALSE	外部回零信号
eRunMode	运行模式	MoveCtrl_Mode	数据范围	0	设置运动模式
bRun	运行	BOOL	TRUE,FALSE	FALSE	上升沿触发运行模式

输入变量	名称	数据类型	有效范围	初始值	描述
bReset	复位轴	BOOL	TRUE,FALSE	FALSE	复位轴
bReinit	重新初始化轴	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 轴重新初始化
lrPositionOrDistance	位置或距离	LREAL	数据范围	0	设置需要运行的距离或位置
lrVelocity	速度	LREAL	数据范围	0	运行速度值
lrAcceleration	加速度	LREAL	数据范围	0	加速度值
lrDeceleration	减速度	LREAL	数据范围	0	减速度值
eDirection	回零方向	MC_DIRECTION	数据范围		回零初始方向
lrJerk	跃度	LREAL	数据范围	0	曲线加减速的斜率变化值
bContinueJog	连续点动	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 点动连续
bJogFwd	正方向点动	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 正方向点动
bJogBwd	负方向点动	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 负方向点动
bStop	轴停止	BOOL	TRUE,FALSE	FALSE	设置为 TRUE 轴停止
tTimeOut	超时时间	TIME	数据范围	30S	运动超时间, 超过设置时间停止运动

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone	指令执行完成	BOOL	TRUE,FALSE	FALSE	功能块指令执行完成为 TRUE
bBusy	指令正在执行	BOOL	TRUE,FALSE	FALSE	功能块正在执行置为 TRUE
bError	错误	BOOL	TRUE,FALSE	FALSE	设置为 TRUE, 有错
eErrorID	错误代码	SMC_ERROR	-	0	错误码

(3) 功能说明

为了使用户快速调用和使用单轴运动控制功能, 功能块封装了 Codesys 轴模块的上电功能, 回零功能, 点动功能, 相对运动, 绝对运动, 速度运行, 停止, 复位等功能。用户如果需要使用单轴运动控制功能只需要调用此功能块则可以完成上面列举所有功能, 提高用户应用程序的开发效率。

3.3.3 FB_Relative_INVT

FB_Relative_INVT: 相对运动暂停指令。

(1) 指令格式

指令	名称	图形表现	ST 表现
FB_Relative_INVT	相对运动暂停指令		<pre> FB_Relative_INVT(Axis:= , bExecute:= , bStopExe:= , lrDistance:= , lrVelocity:= , lrAcceleration:= , lrDeceleration:= , </pre>

			IrJerk:= , bDone=> , bBusy=> , bCommandAborted=> , bError=> , eErrorID=>);
--	--	--	--

(2) 相关变量

◇ 输入输出变量

输入输出变量	名称	数据类型	有效范围	初始值	描述
Axis	轴	AXIS_REF_SM3	-	-	运动轴

◇ 输入变量

输入变量	名称	数据类型	有效范围	初始值	描述
bExecute	使能	BOOL	TRUE,FALSE	FALSE	上升沿触发
bStopExe	外部暂停	BOOL	TRUE,FALSE	FALSE	置 true 后停止运行，置为 false 后接 stop 前状态继续运行
IrDistance	相对运动目标位置	LREAL	数值范围	0	代表轴的相对位置数据
IrVelocity	速度	LREAL	正数或 0	0	轴运行到目标位置的最大速度
IrAcceleration	加速度	LREAL	正数	0	速度变大时加速度值
IrDeceleration	减速度	LREAL	正数	0	速度变小时减速度值
IrJerk	跃度	LREAL	正数或 0	0	曲线加减速的斜率变化值

◇ 输出变量

输出变量	名称	数据类型	有效范围	初始值	描述
bDone	指令执行完成	BOOL	TRUE,FALSE	FALSE	功能块指令执行完成为 TRUE
bBusy	指令正在执行	BOOL	TRUE,FALSE	FALSE	功能块正在执行置为 TRUE
bCommandAborted	指令被中断	BOOL	TRUE,FALSE	FALSE	当前指令被中断，置为 TRUE
bError	错误	BOOL	TRUE,FALSE	FALSE	异常发生时，置为 TRUE
eErrorID	错误代码	SMC_ERROR	数值范围	0	非 0 值表示有异常

(3) 功能说明

此功能块实现了一个 Codesys 上的相对运动功能且可以暂停，当轴在运行过程中置位 bStopExe 标志，暂停运行，当复位标志继续运行到 IrDistance 剩下的距离。



服务热线：400-700-9997 网址：www.invt.com.cn

产品属深圳市英威腾电气股份有限公司所有 委托下面两家公司生产：（产地代码请见铭牌序列号第2、3位）

深圳市英威腾电气股份有限公司（产地代码：01）
地址：深圳市光明区马田街道松白路英威腾光明科技大厦

苏州英威腾电力电子有限公司（产地代码：06）
地址：苏州高新区科技城昆仑山路1号

工业自动化：■ HMI	■ PLC	■ 变频器	■ 伺服系统
■ 电梯智能控制系统	■ 轨道交通牵引系统		
能源电力：■ UPS	■ 数据中心基础设施	■ 光伏逆变器	■ SVG
■ 新能源汽车动力总成系统	■ 新能源汽车充电系统	■ 新能源汽车电机	



66001-00650