专业·创新·责任·服务

ICS Studio

指令手册

www.i-con.cn

英孚康（浙江）工业技术有限公司
Info Convergence Technology Co,Ltd

目录

一、组操作指令

数组左移 ArySHL 指令

ArySHL 指令将对数组中的多个元素向左（高位方向）移动。

1）指令格式

指令	名称	梯形图表现	ST 表现
ArySHL	数组左移		ArySHL(Array[0],Size1,Num1,Out);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
InOut	数组	整型(详见支持的数据类型)	变量	移位数组

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Size	长度	UDINT	立即数 变量	初始化对象
Num	偏移	UDINT	立即数 变量	移位数量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Out	返回值	BOOL	变量	返回值

支持的数据类型

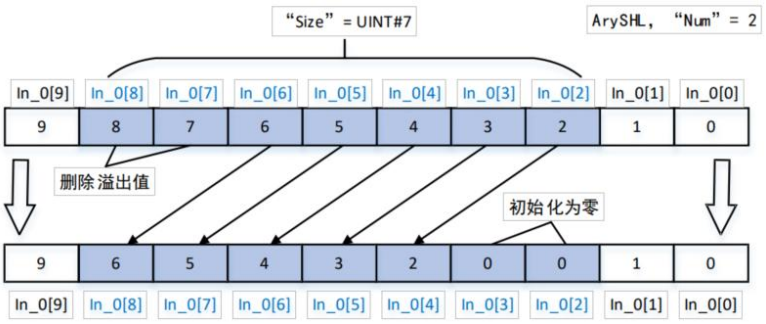
	布尔	整数								实数		字符串		其他
	BOOL	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	STRING	WSTRING	结构体
In Out	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

3）功能说明

1、针对移位对象数组 InOut[ ]的高位“Size”个的元素，进行“Num”位的左移位。

2、移位后清除移位对象数组的溢出元素。

3、移位后将对象数组 InOut[ ]的空缺位元素初始化，即把相应数据类型默认的初始值赋值给空缺位元素。



4、各数据类型默认的初始值如下表所示。

数据类型	初始值
BOOL	0
USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT,	0
REAL, LREAL	0.0
STRING,	‘’
WSTRING	""

4）注意事项

1、 “Num”或“Size”的值为 0 时，不进行移位动作 ，函数返回值的值为 TRUE，InOut[ ]不变。

2、 InOut[]数组下标超出范围编译报错:无效的数组下标。

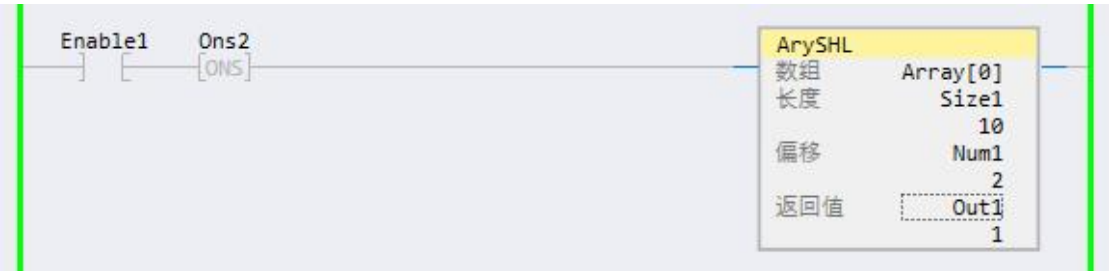
3、 Size 取值不正确（ArySHL 正确范围：1~数组长度-数组起始下标），编译不报错，数组不进行移位，返回值为 FALSE;

4、 Size 取值正确，Num 取值不正确（为负数或“Num” 的值大于“Size”），编译不报错，Size 范围内数组元素赋初始值，函数返回为 TRUE。；

5）错误说明

6）示例

梯形图：



ST 示例：

```
My_OSR.InputBit:=Enable1;
OSRI(My_OSR);
IF My_OSR.OutputBit THEN
    ArySHL(Array[0],Size1,Num1,Out1);
END_IF;
```

数组右移 ArySHR 指令

ArySHR 指令将对数组中的多个元素向右(低位方向)移动。

1）指令格式

指令	名称	梯形图表现	ST 表现
ArySHR	数组右移		ArySHR(Array[0],Size1, Num1,Out);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
InOut	数组	整型(详见支持的数据类型)	变量	移位数组

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Size	长度	UDINT	立即数 变量	初始化对象
Num	偏移	UDINT	立即数 变量	移位数量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

Out	BOOL	BOOL	变量	返回值

支持的数据类型

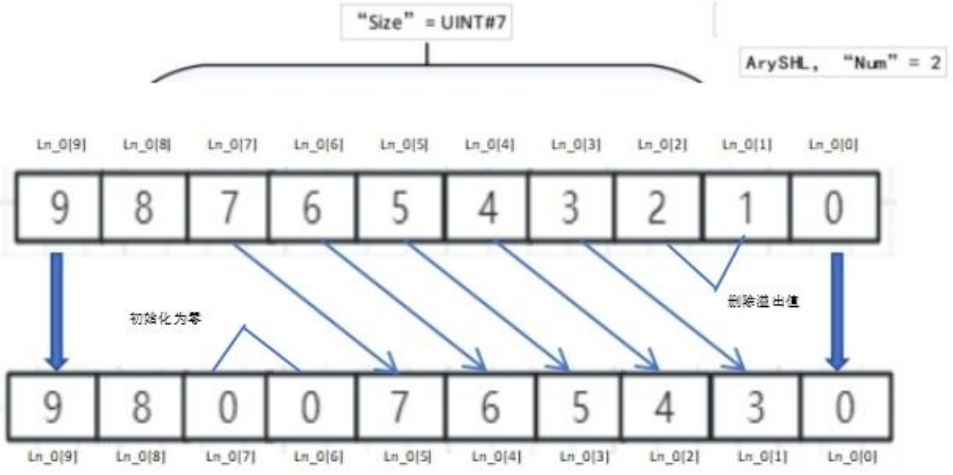
	布尔	整数								实数		字符串		其他
	BOOL	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	STRING	WSTRING	结构体
In Out	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

3）功能说明

1、针对移位对象数组 InOut[ ]的低位“Size”个的元素，进行“Num”位的右移位。

2、移位后清除移位对象数组的溢出元素。

3、移位后将对象数组 InOut[ ]的空缺位元素初始化，即把相应数据类型默认的初始值赋值给空缺位元素。



4、 各数据类型默认的初始值如下表所示。

数据类型	初始值
BOOL	0
USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT,	0
REAL, LREAL	0.0
STRING,	‘’
WSTRING	""

4）注意事项

1、 “Num”或“Size”的值为 0 时，不进行移位动作 ，函数返回值的值为 TRUE，InOut[ ]不变。

2、 InOut[]数组下标超出范围编译报错:无效的数组下标。

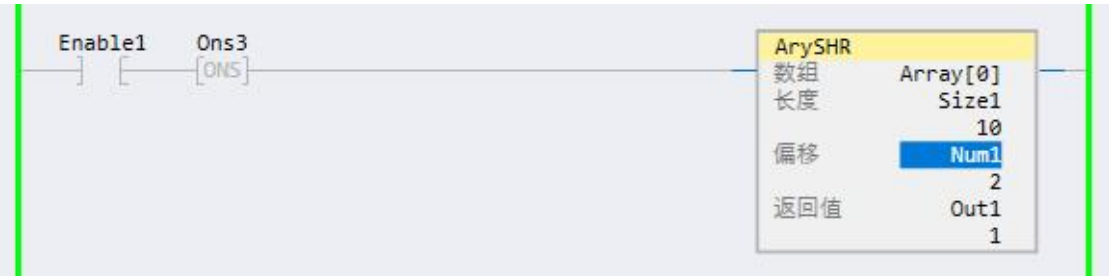
3、 Size 取值不正确（ArySHR 正确范围：1~数组起始下标+1），编译不报错，数组不进行移位，返回值为 FALSE;

4、 Size 取值正确，Num 取值不正确（为负数或“Num” 的值大于“Size”），编译不报错，Size 范围内数组元素赋初始值，函数返回为 TRUE。；

5）错误说明

6）示例

梯形图:



ST 示例：

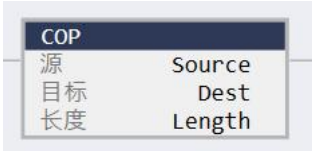
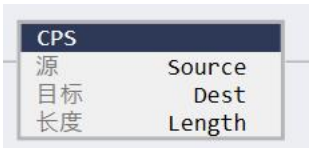
```
My_OSR.InputBit:=Enable1;
OSRI(My_OSR);
IF My_OSR.OutputBit THEN
    ArySHR(Array[0],Size1,Num1,Out1);
END_IF;
```

复制文件 COP/CPS 指令

COP 指令将指定数据源值按长度复制到目标。

CPS 指令将指定数据源值按长度同步复制到目标。

1）指令格式

指令	名称	梯形图表现	ST 表现
COP	复制文件		COP(Source,Dest,Length);
CPS	同步复制文件		CPS(Source,Dest,Length);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	SINT INT DINT LINT REAL 字符串类型 结构	变量	要复制的初始元素

Length	长度	SINT INT DINT	立即数 变量	要复制的目标元素数目

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT REAL 字符串类型 结构 REAL STRIGN 结构	变量	将被 Source 覆盖的初始元素

3）功能说明

COP 和 CPS 指令用于将 Source 中的值复制到 Dest 的值中。Source 保持不变。

条件	梯形图操作	结构化文本操作
预扫描	不适用	请参见“梯形图”表中的“预扫描”行
梯级输入条件为假	将梯级输出条件设置为梯级输入	请参见“梯形图”表中的“梯级输

	条件。	入条件为假”行。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 该指令会复制数据。	请参见“梯形图”表中的“梯级输入条件为真”行。
后扫描	不适用	不适用

4）注意事项

在 COP 和 CPS 指令的执行过程中，其他控制器操作可能试图中断复制操作并更改源数据或目标数据。

如果源或目标为	并且需要	则选择	说明
生产者变量 消费者变量 I/O 数据 另一个任务可以覆盖的数据	防止在复制操作过程中数据发生更改	CPS	延迟试图中断 CPS 指令的任务，直到指令完成.
生产者变量 消费者变量 I/O 数据 另一个任务可以覆盖的数据	允许在复制操作过程中数据发生更改。	COP	
以上都不是		COP	

复制的字节数为

字节数=Length*(Destination 数据类型的字节数)；

注：如果字节数大于 Source 的长度，将为剩余元素复制无法预料的数据。

您必须测试并确认指令不会更改您不希望更改的数据。

COP 和 CPS 指令对连续的内存进行操作。它们进行直接的字节到字节的内存复制。在某些情况下，它们将跨过数组写入标记的其他成员如果长度过大并且标记是用户自定义的数据类型，则会发生这种情况。

如果变量是	则
用户自定义的数据类型	如果长度过大，该指令将跨过数组末尾写入标记的其他成员。它停止于标记的末尾。未发生严重错误。
非用户自定义的数据类型	如果长度过大，该指令将停止于数组末尾。未发生严重错误。

如果长度值超过 Destination 数组中的总元素数，则表示长度过大。

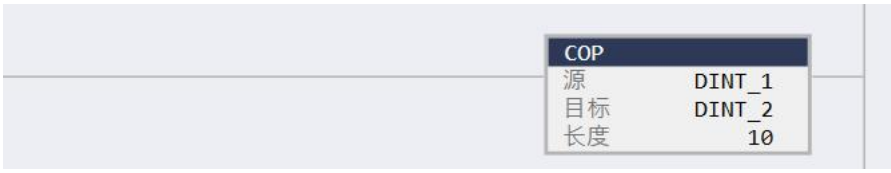
5）错误说明

6）示例

示例 1:

DINT_1 和 DINT_2 的数据类型相同。启用时，COP 指令将 DINT_1 中的前 10 个元素复制到 DINT 2 中的前 10 个元素中。

梯形图:



ST 示例：

```
COP(DINT_1,DINT_2,10);
```

填充 FLL 指令

FLL 指令将填充一块存储区域。

1）指令格式

指令	名称	梯形图表现	ST 表现
FLL	填充填充		FLL(Source,Dest,Length);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL STRUCT	立即数 或变量 Tag	要复制的元素

		ENUM 成员 STRUCT 成员 STRING		
Length	长度	SINT INT DINT LINT USINT UINT UDINT ULINT	立即数 或变量 Tag	传送数组元素数量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	BOOL SINT INT DINT LINT USINT UINT UDINT ULINT	变量 Tag	目标数组

		REAL LREAL STRUCT STRING 以及对应类型的数组		

3）功能说明

FLL 指令将提供的源值填入内存块。Source 保持不变。

如果目标数组为 SINT、INT、DINT 或 REAL 型，而源值为不同类型，则源值将在存储前转换为目标类型。较小的整型类型将通过符号扩展方式转换为较大的整型类型。

如果目标数组是一个结构，源值将直接写入而不进行转换。

执行行为：

条件/状态	执行的操作
预扫描(Prescan)	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令将填充内存
后扫描(Postscan)	不适用

4）注意事项

1、数组越界检查在 FLL 指令有效；

2、Length<1 时会导致编译报错；

3、传送源与目标数组数据类型不一致，且其中之一为 BOOL、结构体、STRING 类型，

会导致编译报错；

5）错误说明

6）示例

梯形图：

ST 示例：

```
My_OSR.InputBit:=Enable1;
OSRI(My_OSR);
IF My_OSR.OutputBit THEN
    FLL(Source1,Dest_5,1);
END_IF;
```

最大值捕捉 MAXC 指令-ST

MAXC 指令用于保留某段时间内输入的最大值，并支持用户根据需要重新确定最大值。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAXC	最大值捕捉	此指令不可用于梯形图中	MAXC(MAXC_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
MAXC_tag	MAXIMUM_CAPTURE	结构	MAXC 结构

MAXIMUM_CAPTURE 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	使能输入。如果清零，该指令将不会执行， 并且不会更新输出。 默认值为置位。
In	模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0

Reset	复位	BOOL	变量 立即数	复位控制算法请求。只要 Reset 置位，指令就会设置 Out = ResetValue。 默认清零。
ResetValue	指令的复位值	REAL	变量 立即数	指令的复位值。只要 Reset 置位，指令就会设置 Out = ResetValue。 有效值 = 任意浮点值 默认值 = 0.0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	启用输出。
Out	计算所得的算法输出	REAL	变量	计算所得的算法输出。

3）功能说明

MAXC 指令执行此算法：

条件	操作
Reset 置位	LastMaximum = 复位值 Out = LastMaximum
Reset 清零	如果 In > LastMaximum，则更新 LastMaximum。 Out = LastMaximum。

执行行为

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

ST 示例：

```
MAXCTag.In := input_value;

MAXCTag.Reset := reset_input;

MAXCTag.ResetValue := reset_value;

MAXC(MAXCTag);

result := MAXCTag.Out;
```

最小值捕捉 MINC 指令-ST

MINC 指令用于保留某段时间内输入的最小值，并支持用户根据需要重新确定最小值。

1）指令格式

指令	名称	梯形图表现	ST 表现
MINC	最小值捕捉	此指令不可用于梯形图中	MINC(MINC_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
MINC_tag	MINIMUM_CAPTURE	结构	MINC 结构

MINIMUM_CAPTURE 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	使能输入。如果清零，该指令将不会执行，并且不会更新输出。 默认值为置位。
In	模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
Reset	复位	BOOL	变量 立即数	复位控制算法请求。只要 Reset 置位，指令就会设置 Out = ResetValue。

				默认清零。
ResetValue	指令的复位值	REAL	变量 立即数	指令的复位值。只要 Reset 置位，指令就会设置 Out = ResetValue。 有效值 = 任意浮点值 默认值 = 0.0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	启用输出。
Out	计算所得的算法输出	REAL	变量	计算所得的算法输出。

3）功能说明

MINC 指令执行此算法：

条件	操作
Reset 置位	LastMinimum = 复位值 Out = LastMinimum
Reset 清零	如果 In > LastMinimum，则更新 LastMinimum。 Out = LastMinimum。

执行行为

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

ST 示例：

```
MINCTag.In := input_value;
MINCTag.Reset := reset_input;
MINCTag.ResetValue := reset_value;
MINC(MINCTag);
result := MINCTag.Out;
```

返回元素数量 SIZE 指令

SIZE 指令返回数组元素的数量。

1）指令格式

指令	名称	梯形图表现	ST 表现
SIZE	返回元素数量		SIZE(Source,0,Size);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	SINT INT DINT REAL 结构 字符串类型	数组变量	指令执行运算要使用的数组的第一个元素 在验证过程中将不接受非数组变量
Dimtovary	维度	DINT	立即数 （0、1、2）	要使用的维度： 第一维: 0 第二维: 1 第三维: 2

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

Size	大小	SINT INT DINT REAL	变量	用于存储数组指定维度中元素数目的变量

3）功能说明

SIZE 指令获取 Source 数组的指定维度中的元素数目(大小)，并将结果放到 Size 操作数中。

该指令获取数组的某一维度的大小。

条件	梯形图操作	结构化文本操作
预扫描	不适用	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。	参考梯形图操作
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 指令执行。	参考梯形图操作
后扫描	不适用	不适用

该指令可对以下类型的数据执行运算：

数组

结构中的数组

作为较大数组一部分的数组

字符串型变量

4）注意事项

应注意以下几点：

1.输出变量操作数被改写。

2.结构操作数的组成部分被改写。

3.除非另外指定，否则结构操作数由多条指令共用。

5）错误说明

6）示例

示例 1：获取 Source 的维度 0(第一维)的元素数目。将大小存储到 Size。在此示例中，Source 的维度 0 有 10 个元素。

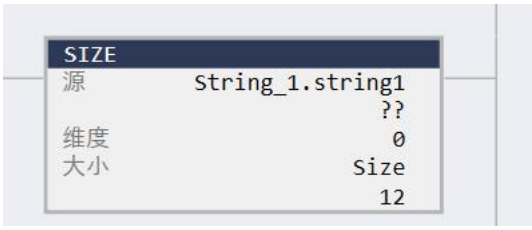
梯形图：

ST：

```
SIZE(Source,0,Size);
```

示例 2：String_1 是字符串结构组成的数组。SIZE 指令获取字符串结构的 DATA 成员中元素的数目，并将大小存储到 Size。在此示例中，DATA 成员有 12 个元素。(该字符串结构的用户指定长度为 12。

梯形图：



ST 示例:

```
SIZE(String_1.string1,0,Size);
```

二、字符串函数指令

字符串连接 CONCAT 指令

CONCAT 指令将源字符串 A 和 B 进行连接

1）指令格式

指令	名称	梯形图表现	ST 表现
CONCAT	字符串连接		CONCAT(Source1,Source2,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce1	源数据	STRING WSTRING UDS	变量	包含起始字符的变量
Scurce2	源数据	STRING WSTRING UDS	变量	包含结束字符的变量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

Dest	目标	STRING WSTRING UDS	变量	要存储大写字符的变量

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为 false	不适用
梯级输入条件为 true	指令执行
后期扫描	不适用

4）注意事项

CNCAT 指令会将 Source 1 中的字符与 Source 2 中的字符相结合,并将结果放入 Dest 中。

Source 1 中的字符在前，Source 2 中的字符在后。

除非 Source 1 和 Dest 是同一变量，否则 Source 2 保持不变

5）错误说明

6）示列

梯形图:

ST 示列：

Scurce1 为 AA，Scurce2 为 88，CONCAT 启动时，会把 Scurce1 里的值与 Scurce2 里的值连接在一起，并把得到数值“AA88”储存到 Dest

CONCAT(Scurce1,Scurce2,Dest);

字符串删除 DELETE 指令

DELETE 指令将源字符串中的部分字符并删除。

1）指令格式

指令	名称	梯形图表现	ST 表现
DELETE	字符串删除		DELETE(Source,Quantity,Start,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	字符串类型	变量	变量，从其中包含的字符串中删除字符
Quantity	字符数	SINT INT DINT LINT	变量 立即数	要删除的字符数
Start	起始字符位	SINT INT	变量 立即数	要删除的第一个字符的位置

		DINT LINT		

输出变量(Output)

参数(英文)	参数(中文)	STRING WSTRING UDS	格式	说明
Dest	目标	STRING WSTRING UDS	变量	用于存储结果的变量

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为 false	不适用
梯级输入条件为 true	指令执行
后期扫描	不适用

4）注意事项

DELETE 指令从 Source 中删除（移除）一个或多个字符，并将其余字符放在 Dest 中。

❷ Start 位置和 Quantity 可确定要删除的字符。

除非 Source 和 Destination 是同一变量，否则 Source 保持不变

5）错误说明

6）示列

梯形图示列 ：

ST 示列：

Scurce 为“ABCDEF”，Quantity 为 2，Start 为 2，DELETE 启动时，会把 Scurce 里的值中的“BC”，删除掉把剩余的数值“ADEF”,储存到 Dest。

DELETE(Scurce,Quantity,Start,Dest);

浮点型转换到字符串 DTOS 指令

RTOS 指令将浮点型数字转换为字符串表示。

1）指令格式

指令	名称	梯形图表现	ST 表现
RTOS	浮点型转换到字符串		DTOS(Source,Decimal,DEST);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	REAL	变量	包含 REAL 型值的变量
Style	类型	SINT INT DINT USINT UINT	枚举变量	0:Decimal 1:Hex

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Destination	目标	STRING	变量	要存储 ASCII 值的

		WSTRING UDS		变量

3）功能说明

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

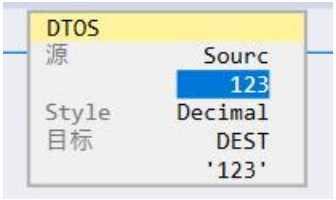
4）注意事项

RTOS 指令将 Source 转换为 ASCII 字符串并将结果放在 Dest 中。

5）错误说明

6）示列

梯形图：



Scurce 为“123”,REPLACE 启动时，输出“123”储存到 DEST。

ST 示列：

DTOS(Scurce,Decimal,Destination);

字符串查找 FIND 指令

FIND 指令将在源字符串中查找且定位到特定字符串的起始位置。

1）指令格式

指令	名称	梯形图表现	ST 表现
FIND	字符串查找		FIND(Source,Search,Start,Result);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	SINT INT DINT LINT	变量	要在其中进行搜索的字符串
Search	源数据	STRING WSTRING UDS	变量	要查找的字符串
Start	数据搜索位置	SINT INT DINT	变量 立即数	Source 中要开始进行搜索的位置

		LINT		

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Rseult	数据搜索到位置	SINT INT DINT LINT	变量	Source 中找到搜索字符串的位置

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为 false	不适用
梯级输入条件为 true	指令执行
后期扫描	不适用

4）注意事项

FIND 指令在 Source 字符串中搜索 Search 字符串。如果指令找到 Search 字符串，Result 会显示 Search 字符串在 Source 字符串中的起始位置。否则 Result 为零

5）错误说明

6）示列

梯形图示列 ：

ST 示列：

Scurce 为“ABCDEF”，Search 为 D，Start 为 2，FIND 启动时，会从 Scurce 里的值第二位开始检索并找到 D 的起始位，并把 D 起始“4”,储存到 Dest

FIND(Source,Search,Start,Result);

字符串插入 INSERT 指令

INSERT 指令将源字符串插入到目标字符串的指定位置。

1）指令格式

指令	名称	梯形图表现	ST 表现
INSERT	字符串插入		INSERT(Source1,Source2,Start,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce1	源数据	STRING WSTRING UDS	变量	要添加字符的字符串
Scurce2	源数据	STRING WSTRING UDS	变量	包含要添加的字符的字符串
Start	数据位置	SINT INT DINT LINT	变量 立即数	Source1 中要添加字符的位置

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	字符串类型	变量	用于储存结果的字符串

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为 false	不适用
梯级输入条件为 true	指令执行
后期扫描	不适用

4）注意事项

INSERT 指令会将 Source 2 中的字符添加到 Source 1 中的指定位置,并将结果放入 Dest 中。

Start 定义 Source 1 中添加 Source 2 的位置。除非 Source 1 和 Dest 是同一变量，否则 Source 1 保持不变

5）错误说明

6）示列

梯形图:

ST 示列：

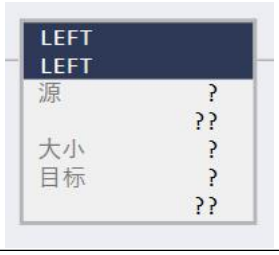
Scurce 为“ABEF”，Search 为 CD，Start 为 3，INSERT 启动时，会从 Scurce 里的值第三位开始插入“CD”，并把得到的结果“ABCDEF”,储存到 Dest

INSERT(Source1,Source2,Start,Dest);

从左边取字符串 LEFT 指令

LEFT 指令将源字符串从左边取字符串。

1）指令格式

指令	名称	梯形图表现	ST 表现
LEFT	从左边取字符串		LEFT(Source,Size,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	STRING WSTRING UDS	立即数 变量	要提取的源字符串
Size	大小	SINT INT DINT LINT	立即数 变量	要提取的字符数量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	提取后的部分字符串

3）功能说明

1、使用指令“LEFT”提取以 Source 输入参数中字符串的第一个字符开头的部分字符串。在 Size 参数中指定要提取的字符数。提取的字符以 (W)STRING 格式通过 Dest 输出参数输出。

2、如果要提取的字符数大于字符串的当前长度，则 Dest 输出参数会将输入字符串作为结果返回。如果 Size 参数包含值“0”或者输入值为空字符串，则将返回空字符串。如果 Size 参数中的值为负数，则将输出空字符串。

3、Source 和 Dest 需为同一种数据类型，即都为 STRING 或 WSTRING，否则需要编译报错：无效数据类型。实际参数必须匹配形式参数数据类型。

4、如果生成的字符串的长度大于 Dest 输出参数中指定的变量长度，则将生成的字符串限制到可用长度。

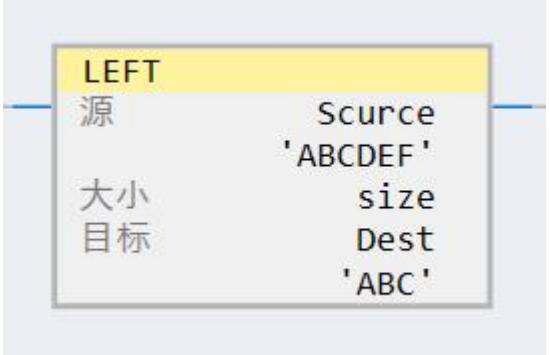
4）注意事项

Size 大小应在 0 和 Source 长度之间；

5）错误说明

6）示例

梯形图：



Scurce 为“ABCDEF”，Size 为 3，DELETE 启动时，会把 Scurce 里的值中的“ABC”提取出来储存到 Dest 里。

ST 示例：

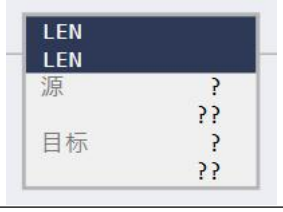
```
LEFT(Scurce,Size,Dest);
```

获取字符串长度 LEN 指令

LEN 指令将获取源字符串的长度。

1）指令格式

指令	名称	梯形图表现	ST 表现

LEN	获取字符串长度		LEN(Source,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	STRING WSTRING UDS	立即数 变量	要提取的源字符串

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT	变量	字符串长度

3）功能说明

1. 使用指令“LEN”，可查询 Source 输入参数中指定的字符串当前长度。并将其作为数值输出到输出参数 Dest 中。空字符串的长度为零。
2. 字符串中的空格也占用一个字符长度。

4）注意事项

当 Source 的字符串长度超出 Dest 的范围时，Dest 会重新从最小值循环计数（无符号整形从 0 开始，有符号整形从负极限开始）；

例如：Source 的字符串长度 128，而 Dest 使用 SINT（-128~127），则最终的输　出结果为-1。

5）错误说明

此指令无特定错误

6）示例

梯形图：

Scurce 为“ABCDEF”，STOR 启动时，会把 Scurce 里字符串的长度“6”储存到 Dest

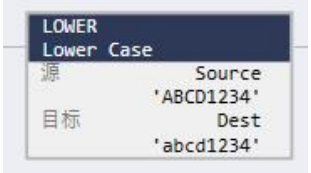
ST 示例：

```
LEN(Scurce,Dest);
```

字符串转换为小写 LOWER 指令

LOWER 指令将字符串中的字母字符转换为小写字符。

1）指令格式

指令	名称	梯形图表现	ST 表现
LOWER	字符串转换为小写		LOWER(Source,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	STRING WSTRING UDS	变量	包含要转换为小写写形式的字符的变量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	要存储小写字符的变量

3）功能说明

条件/状态	执行的操作

预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

4）注意事项

LOWER 指令将 Source 中的所有字母转换为小写形式，并将结果放在 Dest 中。

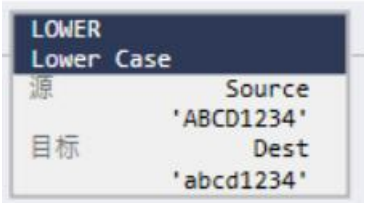
ASCII 字符区分大小写。大写 A ($41) 不等于小写 a ($61)。

直接输入 ASCII 字符，应在比较前将字符转换为全大写或全小写

5）错误说明

6）示列

梯形图：



ST 示列：

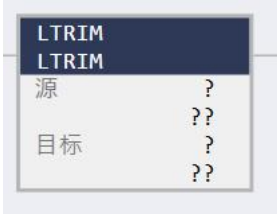
Scurce 为 AA88，LOWER 启动时，会把 Scurce 里的值转化为小写，并把得到数值“aa88”储存到 Dest

LOWER(Scurce,Dest);

字符串左侧调整 LTRIM 指令

LTRIM 指令将源字符串左侧的空格删除。

1）指令格式

指令	名称	梯形图表现	ST 表现
LTRIM	字符串左侧调整		LTRIM(Source,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	STRING WSTRING UDS	变量	要调整的源字符串

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	调整后的字符串

3）功能说明

1. 使用指令“LTRIM”删除 Source 输入参数字符串开头的空格，若字符串的开头

没有空格，则输出的结果与输入一致。

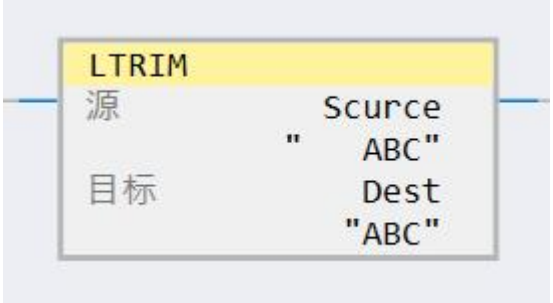
2. Source 和 Dest 需为同一种数据类型，即都为 STRING 或 WSTRING，否则需要编译报错：无效数据类型。实际参数必须匹配形式参数数据类型。
3. 如果生成的字符串的长度大于 Dest 输出参数中指定的变量长度，则将生成的字符串限制到可用长度。

4）注意事项

5）错误说明

6）示例

梯形图：



Scurce 为“ ABC”，STOR 启动时，会把 Scurce 里字符串左侧的空格删除，并把剩下的字符储存到 Dest

ST 示例：

```
LTRIM(Scurce,Dest);
```

字符串片段复制 MID 指令

MID 指令从源字符串中复制部分字符串，可自定义数量和起始位置

1）指令格式

指令	名称	梯形图表现	ST 表现
MID	字符串片段 复制		MID(Source,Quantity,Start,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	STRING WSTRING UDS	变量	要从中复制字符的 字符串
Quantity	源数据	SINT INT DINT LINT	变量 立即数	要复制的字符数
Start	源数据位置	SINT INT DINT	变量 立即数	要复制的第一个字 符的位置

		LINT		

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	要将字符复制到的字符串

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为 false	不适用
梯级输入条件为 true	指令执行
后期扫描	不适用

4）注意事项

MID 指令从 Source 中复制一组字符并将结果放在 Destination 中。Start 位置和

Quantity 可确定要复制的字符。

❷ 除非 Source 和 Dest 是同一变量，否则 Source 保持不变

5）错误说明

此指令无特定错误

6）示列

梯形图示列 ：

ST 示列：

Scurce1 为“ABCDEF”，Quantity 为“2”，Start 为 1，,MID 启动时，会从 Scurce 里的值第 1 位开始复制，复制长度为 2，并把得到结果“AB”,储存到 Dest

MID(Source,Quantity,Start,Dest);

字符串替换 REPLACE 指令

REPLACE 用目标字符串替换掉原字符串中的一部分，将替换后生成的新字符串作为返回值

1）指令格式

指令	名称	梯形图表现	ST 表现
REPLACE	字符串替换		REPLACE(Source1, Source2, Start1, Start2, DEST);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce1	源数据	STRING WSTRING UDS	变量	要替换字符的字符串
Scurce2	源数据	STRING WSTRING UDS	变量	包含要添加的字符的字符串

Start1	数据长度	SINT INT DINT LINT	变量 立即数	Source1 中要添加字符的长度
Start2	数据位置	SINT INT DINT LINT	变量 立即数	Source1 中要添加字符的位置

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	用于储存结果的字符串

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为 false	不适用
梯级输入条件为 true	指令执行
后期扫描	不适用

4）注意事项

REPLACE 指令会将 Source 2 中的字符替换添加到 Source 1 中的指定位置，并将结果放入 Dest 中。

Start1 定义 Source 1 中替换 Source 2 的字符串长度，Start2 定义 Source1 中替换 Source2 的字符串位置。

5）错误说明

6）示列

梯形图：

ST 示列：

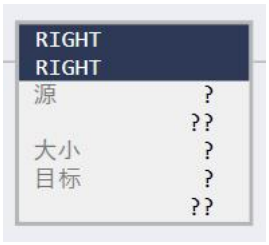
Scurce1 为“ABCDEF”，Search2 为“abc”，Start1 为 3，Start2 为 2,REPLACE 启动时，会从 Scurce1 里的值第二位开始把“BCD”替换为“abc”，并把得到的结果“AabcEF”,储存到 DEST

REPLACE(Source1,Source2,Start1,Start2,DEST);

从右边取字符串 RIGHT 指令

RIGHT 指令将源字符串从右边取字符串。

1）指令格式

指令	名称	梯形图表现	ST 表现
RIGHT	从右边取字符串		RIGHT(Source,Size,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	STRING WSTRING UDS	立即数 变量	要提取的源字符串
Size	大小	SINT INT DINT LINT	立即数 变量	要提取的字符数量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING	变量	提取后的部分字符串

		UDS		

3）功能说明

1. 使用指令“RIGHT”提取以 Source 输入参数中字符串的最后一个 Size 字符。在 Size 参数中指定要提取的字符数。提取的字符以 (W)STRING 格式通过 Dest 输出参数输出。
2. 如果要提取的字符数大于字符串的当前长度，则 Dest 输出参数会将输入字符串作为结果返回。如果 Size 参数包含值“0”或者输入值为空字符串，则将返回空字符串。如果 Size 参数中的值为负数，则将输出空字符串。
3. Source 和 Dest 需为同一种数据类型，即都为 STRING 或 WSTRING，否则需要编译报错：无效数据类型。实际参数必须匹配形式参数数据类型。
4. 如果生成的字符串的长度大于 Dest 输出参数中指定的变量长度，则将生成的字符串限制到可用长度。

4）注意事项

Size 大小应在 0 和 Source 长度之间;

5）错误说明

6）示例

梯形图:

ST 示例：

Scurce 为“ABCDEF”，Size 为“3”STOR 启动时，会把 Scurce 从右取三个字符储存到 Dest 里

RIGHT(Scurce,size,Dest);

浮点型转换到字符串 RTOS 指令

RTOS 指令将浮点型数字转换为字符串表示。

1）指令格式

指令	名称	梯形图表现	ST 表现
RTOS	浮点型转换到字符串		RTOS(Source,DEST);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	REAL	变量	包含 REAL 型值的变量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Destination	目标	STRING WSTRING UDS	变量	要存储 ASCII 值的变量

3）功能说明

条件/状态	执行的操作

预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

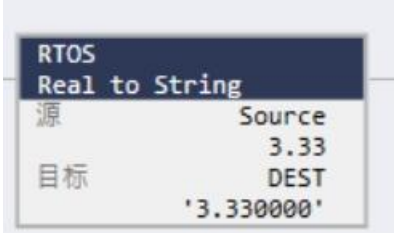
4）注意事项

RTOS 指令将 Source 转换为 ASCII 字符串并将结果放在 Dest 中。

5）错误说明

6）示列

梯形图：



Scurce 为“3.33”,REPLACE 启动时，输出“3.330000”储存到 DEST。

ST 示列：

RTOS(Scurce,Destination);

字符串右侧调整 RTRIM 指令

RTRIM 指令将源字符串右侧的空格删除。

1）指令格式

指令	名称	梯形图表现	ST 表现
RTRIM	字符串右侧调整		RTRIM(Source,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	STRING WSTRING UDS	立即数 变量	要调整的源字符串

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	调整后的字符串

3）功能说明

1. 使用指令“RTRIM”删除 Source 输入参数字符串末尾的空格，若字符串的

末尾没有空格，则输出的结果与输入一致。

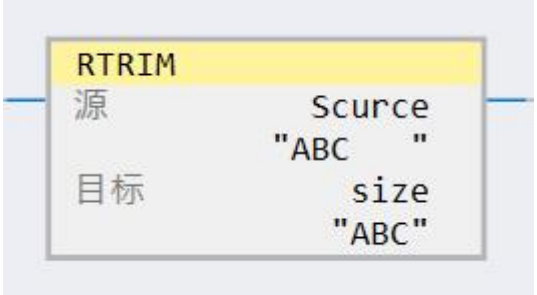
2. Source 和 Dest 需为同一种数据类型，即都为 STRING 或 WSTRING，否则需要编译报错：无效数据类型。实际参数必须匹配形式参数数据类型。
3. 如果生成的字符串的长度大于 Dest 输出参数中指定的变量长度，则将生成的字符串限制到可用长度。

4）注意事项

5）错误说明

6）示例

梯形图：



Scurce 为“ABC ”，STOR 启动时，会把 Scurce 里字符串右侧的空格删除，并把剩下的字符储存到 Dest

ST 示例：

```
RTRIM(Scurce,size);
```

字符串转换到整型 STOD 指令

STOD 指令将字符串中的数字字符转换为整数或 REAL 值。

1）指令格式

指令	名称	梯形图表现	ST 表现
STOD	字符串转换到 整型		STOD(Scurec,Decimal,Destination)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	STRING WSTRING UDS	变量	包含 ASCII 形式的 值的变量
Decimal	元素类型	Decimal HEX	类型	0:Decimal 1:Hex

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Destination	目标	SINT INT DINT	变量 立即数	要存储整型值的变 量

		LINT REAL		

3）功能说明

条件	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。 Destination 清零。 该指令转换 Source。
后扫描	不适用

4）注意事项

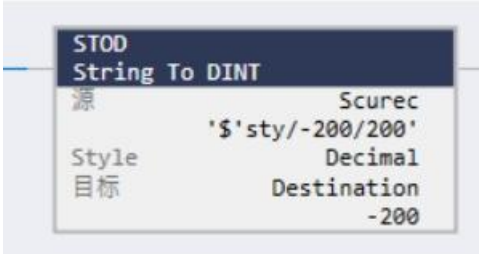
STOD 指令将 Source 转换为整数或 REAL 值并将结果放入 Destination 中。 该指令可转换正数和负数。

如果 Source 字符串包含非数字字符，STOD 将转换第一组连续数字： 该指令将跳过任何起始控制字符或非数字字符（数字前的负号除外）。如果字符串包含多组数字，以分隔符（例如 /）分隔，则指令只转换第一组数字。

5）错误说明

6）示列

梯形图示列 ：



Scurce 为“$’sty/-200/200”，Style 选择十进制（Decimal），REPLACE 启动时，输出“-200”储存到 Destination。

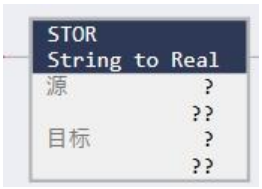
ST 示列：

STOD(Scurce,Decimal,Destination);

字符串转换到浮点型 STOR 指令

STOR 指令将字符串中的数字字符表示转换为浮点型。

1）指令格式

指令	名称	梯形图表现	ST 表现
STOR	字符串转换到浮点型		STOR(Source,Dest)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	STRING WSTRING UDS	变量	包含 ASCII 值的变量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	REAL	变量	要存储 REAL 值的变量

3）功能说明

条件	梯形图操作

预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

4）注意事项

STOR 指令将 Source 转换为 REAL 值，并将结果放入 Dest 中。

❷ 该指令可转换正数和负数。

❷ 如果 Source 字符串包含非数字字符，STOR 将转换第一组连续数 字，包括小数点 [.]。

该指令将跳过任何起始控制字符或非数字字符（数字前的负号除外）。

如果字符串包含多组数字，以分隔符（例如 /）分隔，则指令只转换第一组数字

5）错误说明

6）示列

梯形图：

ST 示列：

Scurce 为 8888，STOR 启动时，会把 Scurce 里的值转化为浮点型，并把得到数 值“8888.0”储存到 Dest

STOR(Scurce,Dest);

字符串转换为宽字符串 STOW 指令

STOW 指令将源字符串转换为宽字符串。

1）指令格式

指令	名称	梯形图表现	ST 表现
STOW	字符串转换为宽字符串		STOW(Source,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	STRING UDS	立即数 变量	要转换的源字符串

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	WSTRING UDS	变量	转换后的宽字符串

3）功能说明

1. 使用指令“STOW”，可将 Source 输入参数的 String 转换为 Wstring，并将 Wstring 的值输出到输出参数 Dest 中。

2. 每一个 String 的成员都可以用两位 16 进制数$xx 来表示，转换为 Wstring 后，会自动补全前两位，为四位 16 进制数$00xx。
3. 若 String 的值为空或者空格，则转换后的 Wstirng 的值也为空或空格。
4. 如果生成的字符串的长度大于 Dest 输出参数中指定的变量长度，则将生成的字符串限制到可用长度。

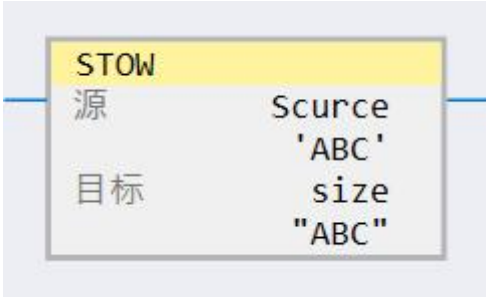
4）注意事项

为防止数据截断，应使生成的 Dest 长度大于 Source 中的字符串长度；

5）错误说明

6）示例

梯形图：



Scurce 为“ABC”，STOR 启动时，会把 Scurce 里 string 类型的字符串转成 Wstring 类型的字符串，储存到 Dest

ST 示例：

```
STOW(Scurce,size);
```

字符串转换为大写 UPPER 指令

UPPER 指令将字符串中的字母字符转换为大写字符

1）指令格式

指令	名称	梯形图表现	ST 表现
UPPER	字符串转换为大写字母		UPPER(Source,Dest)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Scurce	源数据	STRING WSTRING UDS	变量	包含要转换为大写形式的字符的变量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING WSTRING UDS	变量	要存储大写字符的变量

3）功能说明

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	指令执行。
后扫描	不适用

4）注意事项

UPPER 指令将 Source 中的所有字母转换为大写形式，并将结果放在 Dest 中。

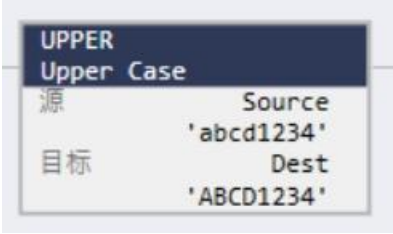
ASCII 字符区分大小写。大写 A ($41) 不等于小写 a ($61)。

直接输入 ASCII 字符，应在比较前将字符转换为全大写或全小写。

5）错误说明

6）示列

梯形图：



ST 示列：

Scurce 为 aa88， UPPER 启动时，会把 Scurce 里的值转化为大写，并把得到数值“AA88”储存到 Dest

UPPER(Scurce,Dest);

宽字符串转换为字符串 WTOS 指令

WTOS 指令将源宽字符串转换为字符串。

1）指令格式

指令	名称	梯形图表现	ST 表现
WTOS	宽字符串转换为字符串		WTOS(Source,Dest)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	WSTRING UDS	立即数 变量	要转换的源字符串

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	STRING UDS	变量	转换后的字符串

3）功能说明

1. 使用指令“DTOS”，可将 Source 输入参数的变量的值转换为 String，其转换的值由输入参数 Style 决定,内部计算时先将输入参数转化为对应的 Style，转化后的值为 String 类型的输出参数，并将 String 的值输出到 Dest。

2. Style 使用枚举值为 Decimal，则与原先的 DTOS 指令无异，输入参数默认为十进制，十进制的输入参数转换成 String 类型的变量。如输入参数的值为 10#1234,Style 的值为 Decimal，转换后的值为’1234’。
3. Style 使用枚举值为 Hex，十六进制的输入参数转换成 String 类型的变量。如输入参数的值为 16#123AF012,Style 的值为 Hex，转换后的值为’123AF012’。
4. 如果生成的字符串的长度大于 Dest 输出参数中指定的变量长度，则将生成的字符串限制到可用长度。
5. 若输入参数存在负数，则会将负的符号转换为字符串的’-’，如 Source 的值为 10#-100,则 Dest 的值为’-100’

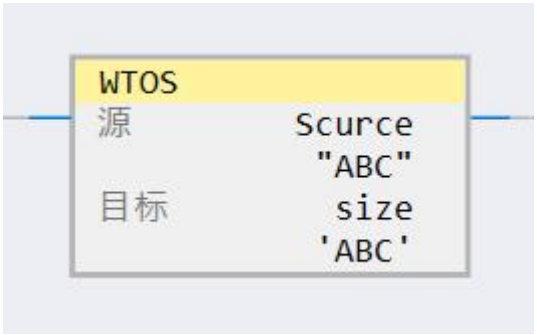
4）注意事项

为防止数据截断，应使生成的 Dest 长度大于 Source 中的源字符串长度；

5）错误说明

6）示例

梯形图：



Scurce 为“ABC”，STOR 启动时，会把 Scurce 里 Wstring 类型的字符串转成 String 类型的字符串，储存到 Dest

ST 示例：

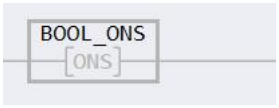
WTOS (Scurce,size);

三、位操作指令

单次触发 ONS 指令

ONS 指令根据存储位的状态启用或禁用梯级的其余部分。

1）指令格式

指令	名称	梯形图表现	ST 表现
ONS	单次触发		此指令不支持结构化文本。

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Storage_Bit	存储位	BOOL	变量	内部存储位存储上次执行指令时的梯级输入条件

3）功能说明

情况	行为
预扫描	设置存储位以防止第一次扫描期间出现无效触发梯级输出条件设置为 false。
梯级输入条件为 false	清除存储位梯级输出条件设置为 false。
梯级输入条件为 true	数据位=0 时设置存储位梯级输出条件设置为 true。 数据位=1 时存储位保持为设置状态梯级输出条件设置为 false。
后期扫描	清除存储位梯级输出条件设置为 false。

4）注意事项

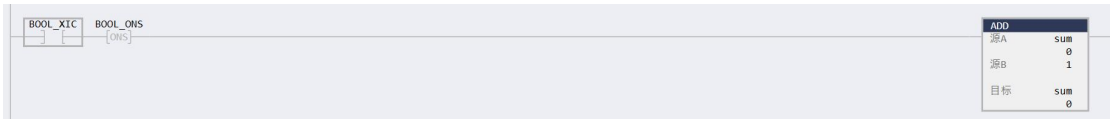
启用时，如果清除了存储位，ONS 指令将启用梯级的其余部分。当禁用时或者当设置　　了存储位时，ONS 指令将禁用梯级的其余部分。

5）错误说明

6）示例

梯形图：

通常需要在 ONS 指令前加上输入指令，因为您会在启用和禁用 ONS 指令时对它进行扫描，以使它正常工作。一旦启用了 ONS 指令，必须清除梯级输入条件或存储位，ONS 指令才能再次启用。在清除了 BOOL_XIC 或设置了 BOOL_ONS 的任何扫描上，此梯级没有任何作用。在设置了 BOOL_XIC 并清除了 BOOL_ONS 的任何扫描上 ONS 指令设置 BOOL_ONS ,并且 ADD 指令将 sum 加 1。只要 BOOL_XIC 保持设置状态，sum 就保持相同的值不变。BOOL_XIC 必须再次由清除状态变为设置状态，sum 才能再次递增。



ST 示例：

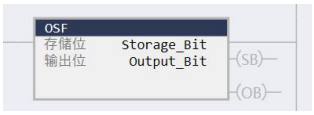
结构化文本没有 ONS 指令，但您可以使用 IF...THEN 结构实现相同的结果。

```
IF BOOL_XIC AND NOT BOOL_ONS THEN
    sum:=sum+1;
END_IF;
BOOL_ONS:=BOOL_XIC;
```

下降沿单次触发 OSF 指令

OSF 指令存储位的状态设置或清除输出位。

1）指令格式

指令	名称	梯形图表现	ST 表现
OSF	下降沿单次触发		OSRI(OSRI_01);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Storage_Bit	存储位	BOOL	变量	内部存储位存储上次执行指令时的梯级输入条件

输出变量(Output)

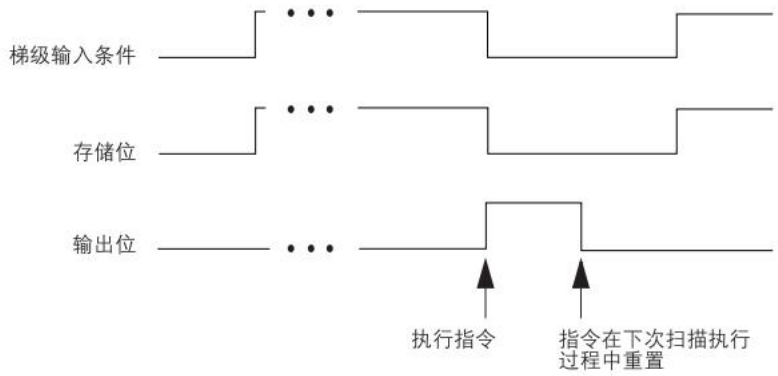
参数(英文)	参数(中文)	数据类型	格式	说明
Output_Bit	输出位	BOOL	变量	要设置的位

3）功能说明

情况	行为
预扫描	清除存储位以防止在第一次扫描期间出现无效触发。 清除输出位。 梯级输出条件设置为 false。

梯级输入条件为 true	设置存储位、清除输出位、梯级输出条件设置为 true。
梯级输入条件为 false	存储位=0 时存储位保持清除状态清除输出位梯级输出条件设置为 false。 数据位=1 时存储位清除设置输出位梯级输出条件设置为 false。
后期扫描	请参见上面梯级输入条件为 false 的情况。

时序图：



4）注意事项

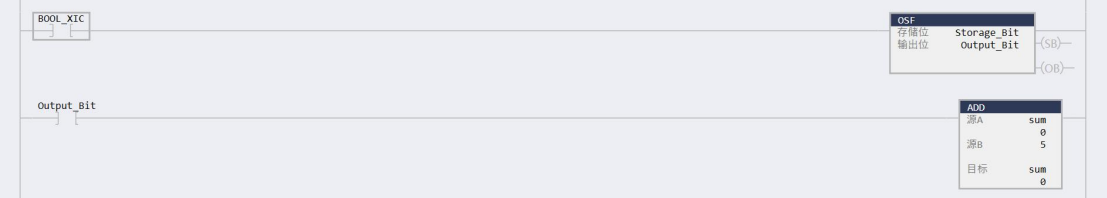
禁用时，如果设置了存储位，OSF 指令将设置输出位。当已禁用但存储位已清除时，或者当已启用时，OSF 指令将清除输出位。

5）错误说明

6）示例

梯形图：

每当 BOOL_XIC 从设置状态变为清除状态时，OSF 指令都设置 Output_Bit 并且 ADD 指令将 sum 加 5。只要 BOOL_XIC 保持清除状态，sum 就保持相同的值不变。BOOL_XIC 必须再次从设置状态变为清除状态，sum 才能再次递增。可以在多个梯级上使用 Output_Bit 以触发其他操作。



ST 示例：

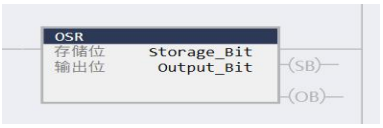
该指令在结构化文本和功能块中可以作为 OSFI 使用。

```
OSFI_01.InputBit:=BOOL_XIC;

OSRI(OSFI_01);

State:=OSFI_01.OutputBit;
```

上升沿单次触发 OSR 指令

OSR 指令根据存储位的状态设置或清除输出位，

1）指令格式

指令	名称	梯形图表现	ST 表现
OSR	上升沿单次触发		OSRI(OSRI_01);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Storage_Bit	存储位	BOOL	变量	内部存储位存储上次执行指令时的梯级输入条件

输出变量(Output)

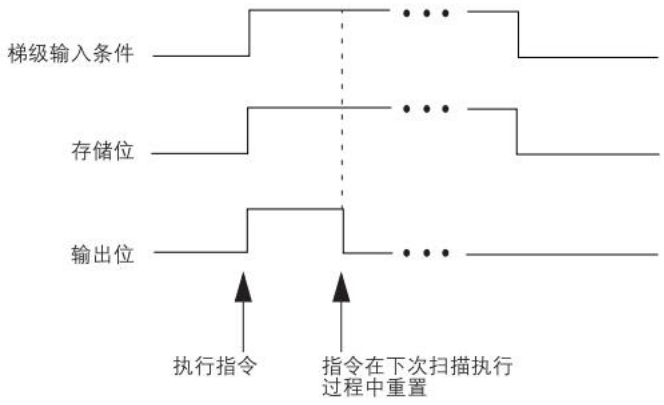
参数(英文)	参数(中文)	数据类型	格式	说明
Output_Bit	输出位	BOOL	变量	要设置的位

3）功能说明

情况	行为
预扫描	设置存储位以防止第一次扫描期间出现无效触发。 清除输出位。 梯级输出条件设置为 false。
梯级输入条件为 false	清除存储。位输出位未修改。梯级输出条件设置为 false。

梯级输入条件为 true	存储位=0 时设置存储位。设置输出位。梯级输出条件设置为 true。 数据位=1 时存储位保持为设置状态。清除输出位。梯级输出条件设置为 true。
后期扫描	清除存储位。输出位未修改。梯级输出条件设置为 false。

时序图：



4）注意事项

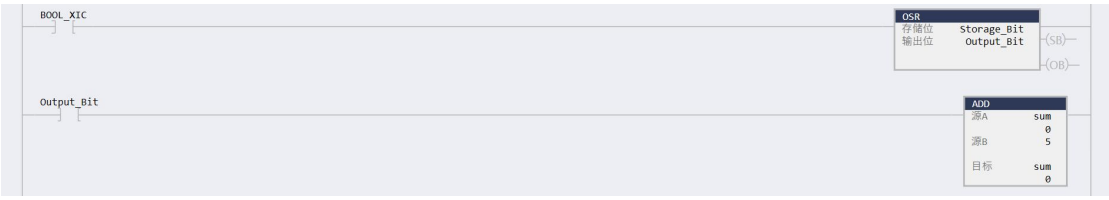
启用时，如果清除了存储位，OSR 指令将设置输出位。当已启用但存储位已设置时或者当已禁用时，OSR 指令清除输出位。

5）错误说明

6）示例

梯形图：

每当 BOOL_XIC 从清除状态变为设置状态时，OSR 指令都设置 Output_Bit 并且 ADD 指令将 sum 加 5。只要 BOOL_XIC 保持设置状态，sum 就保持相同的值不变。BOOL_XIC 必须再次由清除状态变为设置状态，sum 才能再次递增。可以在多个梯级上使用 Output_Bit 以触发其他操作。



ST 示例：

该指令在结构化文本和功能块中可以作为 OSRI 使用。

OSRI_01.InputBit:=BOOL_XIC;

OSRI(OSRI_01);

State:=OSRI_01.OutputBit;

线圈输出 OTE 指令

OTE 指令用于设置或清除数据位。

1）指令格式

指令	名称	梯形图表现	ST 表现
OTE	线圈输出		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Storage_RES	数据位	BOOL	变量	要测试的位

3）功能说明

情况	行为
预扫描	清除数据位。梯级输出条件设置为 false。
梯级输入条件为 false	清除数据位。梯级输出条件设置为 false。
梯级输入条件为 true	设置数据位梯级输出条件设置为 true。
后期扫描	清除数据位。梯级输出条件设置为 false。

4）注意事项

当 OTE 指令启用时，控制器设置数据位。当 OTE 指令禁用时，控制器清除数据位。

5）错误说明

6）示例

梯形图：

当设置了 BOOL_XIC 时，OTE 指令设置(打开)BOOL_OTE。当清除了 BOOL_XIC 时，OTE 指令清除(关闭)BOOL_OTE。

ST 示例：

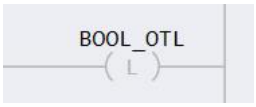
结构化文本没有 OTE 指令，但您可以使用非保持赋值语句实现相同的结果。

BOOL_OTE:=BOOL_XIC;

线圈置位 OTL 指令

OTL 指令用于设置(锁存)数据位。

1）指令格式

指令	名称	梯形图表现	ST 表现
OTL	线圈置位		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Storage_OTL	数据位	BOOL	变量	要设置的位

3）功能说明

情况	行为
预扫描	数据位未修改、梯级输出条件设置为 false。
梯级输入条件为 false	数据位未修改、梯级输出条件设置为 false。
梯级输入条件为 true	设置数据位、梯级输出条件设置为 true。
后期扫描	数据位未修改、梯级输出条件设置为 false。

4）注意事项

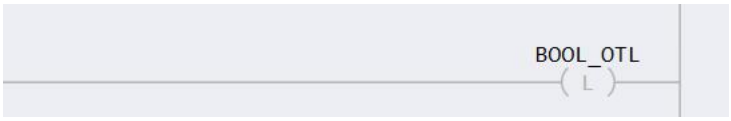
OTL 指令设置数据位。该数据位保持设置状态，直到被清除启用时，清除操作通常由 OTU 指令来执行。禁用时，OTL 指令不更改数据位的状态。

5）错误说明

6）示例

梯形图：

启用时，OTL 指令设置 BOOL_OTL。该数据位保持设置状态，直到被清除。清除操作通常由 OTU 指令来执行。



ST 示例：

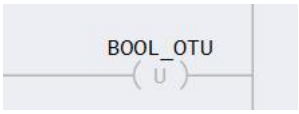
结构化文本没有 OTL 指令，但您可以使用 IF...THEN 结构和赋值语句实现相同的结果。

```
IF BOOL_expression THEN
    BOOL_OTL:=1;
END_IF;
```

线圈复位 OTU 指令

OTU 指令用于清除(解锁存)数据位。

1）指令格式

指令	名称	梯形图表现	ST 表现
OTU	线圈复位		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Storage_OTU	数据位	BOOL	变量	要清除的位

3）功能说明

情况	行为
预扫描	数据位未修改，梯级输出条件设置为 false。
梯级输入条件为 false	数据位未修改、梯级输出条件设置为 false。
梯级输入条件为 true	清除数据位、梯级输出条件设置为 true。
后期扫描	数据位未修改、梯级输出条件设置为 false。

4）注意事项

启用时，OTU 指令清除数据位。禁用时，OTU 指令不更改数据位的状态。

5）错误说明

6）示例

梯形图：

启用时， OTU 指令清除 BOOL_OTU。

ST 示例：

结构化文本没有 OTU 指令，但您可以使用 IF...THEN 结构和赋值语句实现相同的结果。

```
IF BOOL_expression THEN
    BOOL_OTU:=0;
END_IF;
```

常开触点 XIC 指令

XIC 指令用于检查数据位是否被置位。

1）指令格式

指令	名称	梯形图表现	ST 表现
XIC	常开触点		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
BOOL_XIC	数据位	BOOL	变量	要测试的位

3）功能说明

XIC 指令检查数据位是否置位。

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	数据位=0 时梯级输出条件设置为 false 数据位=1 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

4）注意事项

5）错误说明

6）示例

梯形图：

如果设置了 BOOL_XIC ，这将启用下一条指令 (梯级输出条件为 true)。

ST 示例：

结构化文本没有 XIC 指令，但您可以使用 IF...THEN 结构实现相同的结果。

```
IF BOOL_XIC THEN
    <statement>;
END_IF;
```

下降沿指令 XIF 指令

XIF 指令检查操作数的信号下降沿

1）指令格式

指令	名称	梯形图表现	ST 表现
XIF	下降沿指令		此指令不可用于结构化文本中

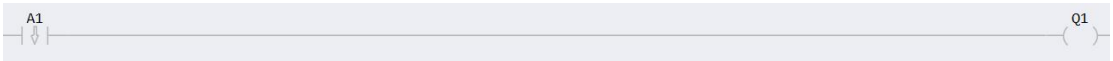
2）相关变量

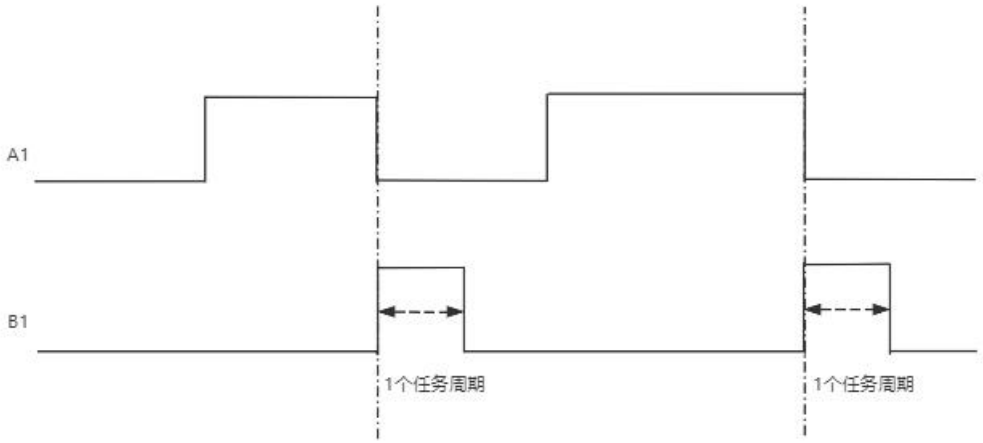
输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Input Data bit	输入数据位	BOOL	立即数 变量	检查到数据位跳变（1->0）时，该触点导通且只能持续一次扫描周期的时间。不可以使用 XIF 结束梯级行。

3）功能说明

时序图：





4）注意事项

当数据位的变量未声明或者数据类型错误，则梯形图梯级行报错。

5）错误说明

6）示例

梯形图示例：

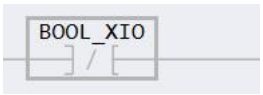
ST 示例：

结构化文本没有 XIF 指令

常闭触点 XIO 指令

XIO 指令用于检查数据位是否已设置。

1）指令格式

指令	名称	梯形图表现	ST 表现
XIO	常闭触点		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
BOOL_XIO	数据位	BOOL	变量	要测试的位

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	数据位=0 时梯级输出条件设置为 true 数据位=1 时梯级输出条件设置为 false
后期扫描	梯级输出条件设置为 false

4）注意事项

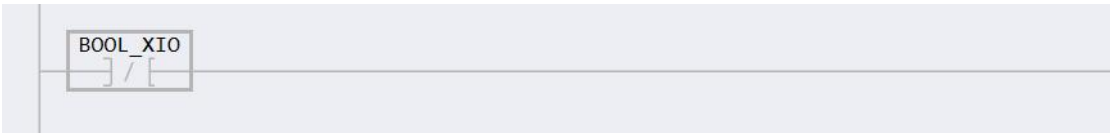
XI0 指令检查数据位是否已清除:

5）错误说明

6）示例

梯形图：

如果清除了 BOOL_XIO ，这将启用下一条指令（梯级输出条件为 true)。



ST 示例：

结构化文本没有 XIO 指令，但您可以使用 IF...THEN 结构实现相同的结果。

```
IF BOOL_XIO THEN
   <statement>;
END_IF;
```

上升沿指令 XIR 指令

XIR 指令检查操作数的信号上升沿

1）指令格式

指令	名称	梯形图表现	ST 表现
XIR	上升沿指令		此指令不可用于结构化文本中

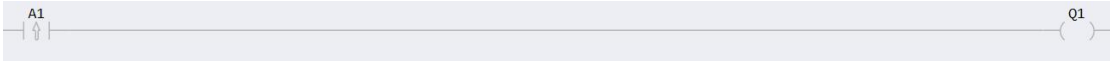
2）相关变量

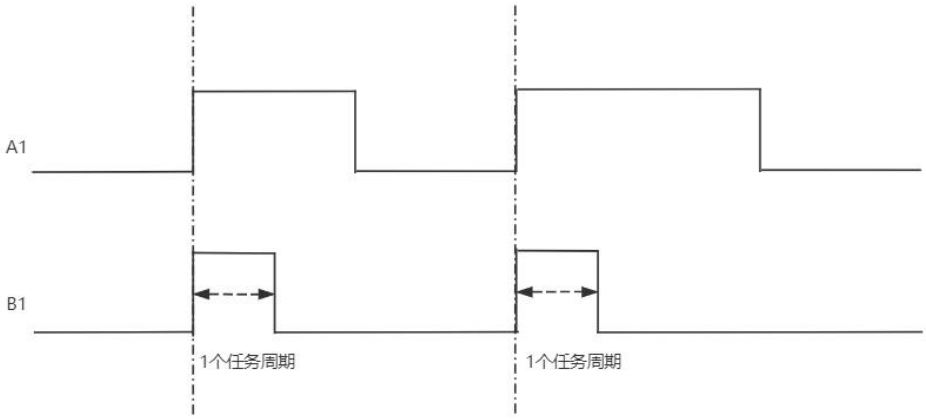
输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Input Data bit	输入数据位	BOOL	立即数 变量	检查到数据位跳变（0->1）时，该触点导通且只能持续一次扫描周期的时间。不可以使用 XIR 结束梯级行。

3）功能说明

时序图：





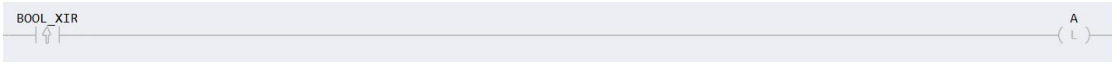
4）注意事项

当数据位的变量未声明或者数据类型错误，则梯形图梯级行报错。

5）错误说明

6）示例

梯形图：



ST 示例：

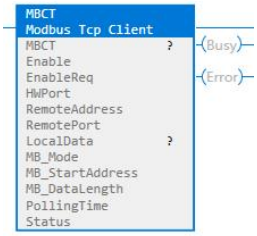
结构化文本没有 XIR

四、通讯服务

Modbus 客户端 MBCT 指令

触发 MODBUS/TCP 客户端命令请求并获取结果，

1）指令格式

指令	名称	梯形图表现	ST 表现
MBCT	Modbus 客户端指令		MBCT_1(LocalData:=, Enable:=, EnableReq:=, HWPort:=, LocalPort:=, RemoteAddress:=, RemotePort:=, UnitID:=, MB_Mode:=, MB_StartAddress:=, MB_DataLength:=, PollingTime:=, Timeout:=, Status=>, Busy=>,

			Error=>, Connected=>, RequestCount=>, SuccessCount=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
LocalData	本地数据	SINT INT DINT LINT REAL LREAL 以及对应数组， BOOL 数组	对应类型范围	0	变量	Modbus 读/写存储的内容,通常为 BOOL 数组或者 INT 数组，也可为其它原子类型(BOOL,SINT,INT,DINT,LINT,REAL,LREAL)以及对应原子类型数组,不支持 BOOL 但支持 BOOL 数组

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
bEnable	连接使能	BOOL	0-1	0	变量或	是否建立与服务

					立即数	端的连接
EnableReq	读写使能	BOOL	0-1	0	变量或 立即数	是否开启对服务 端的 Modbus 查 询
HWPort	网口号	INT	1~3	1	变量或 立即数	PLC 硬件网口 号，取值范围 1/2/3，默认为 1
LocalPort	本地端口	DINT	0-65535	0	变量或 立即数	本地端口,0 代表 ICC 自动分配,>0 代表指定端口,默 认为 0
RemoteAddress	目标 IP	STRING	合法 IP	''	变量或 立即数	目标 IP
RemotePort	目标端口	DINT	502~566	502	变量或 立即数	目标端口，默认 值 502
UnitID	单元 ID	SINT	1~64	1	变量或 立即数	ModbusTCP 网 关下面串口 Modbus 设备地 址，默认值为 1
MB_Mode	访问模式	SINT	0~3	0	变量或 立即数	访问模式读还是 写(0=读 1=写 2=读大地址 3=

						写大地址 其它值预留)
MB_StartAddress	MB 起始地址	DINT	0~65535	1	变量或立即数	Modbus 起始地址
MB_DataLength	MB 数据长度	INT	1~2000(BOOL) 1~125(INT)	1	变量或立即数	Modbus 数据长度
PollingTime	轮询周期	DINT	10~3000	1000	变量或立即数	轮询数据更新周期,ms 毫秒，默认值 1000
Timeout	超时时间	DINT	3000~10000	3000	变量或立即数	超时，ms，默认值 3000

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Status	状态	DINT	0x0000-0x700B	0	变量或立即数	状态与故障码
Connected	已连接	BOOL	0-1	0	变量或立即数	与服务端连接上,原 CD(Connected)
Busy	数据交换中	BOOL	0-1	0	变量	与服务端正交换

					或 立即 数	MB 数据
Error	错误	BOOL	0-1	0	变量 或 立即 数	指令出错
RequestCount	请求次数	DINT	DINT 范围	0	变量 或 立即 数	统计请求次数
SuccessCount	成功次数	DINT	DINT 范围	0	变量 或 立即 数	统计成功次数

3）功能说明

1、访问符合 Modbus TCP 标准协议的 server 服务端。可指定网口(默认 1 号网口)

2、屏蔽了 Modbus 难以记忆的功能码，采用读/写模式(MB_Mode)与起始地址(MB_StartAddress)共同内部决定功能码。

3、模式与起始地址以及功能码对应表如下：

MB_Mode	MB_StartAddress	MB_DataLength	Modbus 功能码	功能说明/备注

0	1 到 9999	1 到 2000	01	在远程地址 0 到 9998(小地址) 处，读取 1 到 2000 个输出位
0	10001 到 19999	1 到 2000	02	在远程地址 0 到 9998(小地址) 处，读取 1 到 2000 个输入位
0	40001 到 49999	1 到 125	03	在远程地址 0 到 9998 (小地址) 处，读取 1 到 125 个保持性寄存器
0	30001 到 39999	1 到 125	04	在远程地址 0 到 9998 (小地址) 处，读取 1 到 125 个输入寄存器
1	1 到 9999	1	05	在远程地址 0 到 9998 (小地址) 处，写入 1 个输出位
1	40001 到 49999	1	06	在远程地址 0 到 9998 (小地址) 处，写入 1 个保持性毒存器
1	1 到 9999	2 到 1968	15	在远程地址 0 到 9998 (小地址) 处，写入 2 到 1968 个输出位
1	40001 到 49999	2 到 123	16	在远程地址 0 到 9998 (小地址) 处，写入 2 到 123 个保持性寄存器
2	1 到 65535	1 到 2000	01	在远程地址 0 到 65534(大地址) 处，读取 1 到 2000 个输出位
2	100001 到 165535	1 到 2000	02	在远程地址 0 到 65534(大地址) 处，读取 1 到 2000 个输入位
2	400001 到 465535	1 到 125	03	在远程地址 0 到 65534(大地址) 处，读取 1 到 125 个保持性寄存器

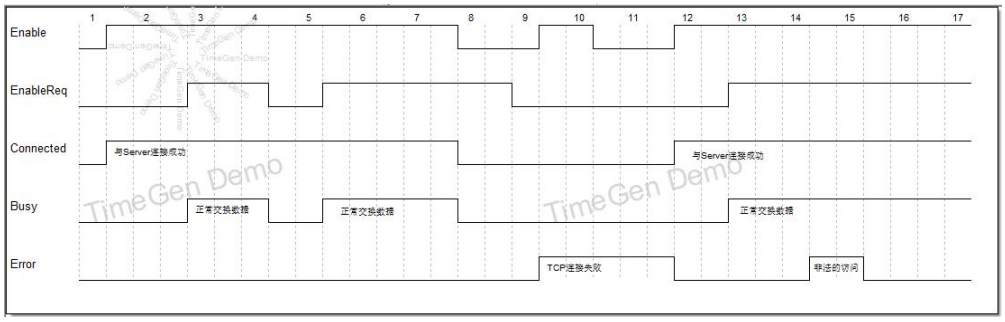
2	300001 到 365535	1 到 125	04	在远程地址 0 到 65534(大地址)处，读取 1 到 125 个输入寄存器
3	1 到 65535	1	05	在远程地址 0 到 65534(大地址)处，写入 1 个输出位
3	400001 到 465535	1	06	在远程地址 0 到 65534(大地址)处，写入 1 个保持性毒存器
3	1 到 65535	2 到 1968	15	在远程地址 0 到 65534(大地址)处，写入 2 到 1968 个输出位
3	400001 到 465535	2 到 123	16	在远程地址 0 到 65534(大地址)处，写入 2 到 123 个保持性寄存器

4、可自定义数据轮询刷新周期(Polling Time)，最快可达 10ms，本地数据 LocalData 接口可定义为任意数据类型，免去数据转换的麻烦。

5、最大可开 64 个 Client 实例，水龙头式服务开启方式。

时序图

下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：



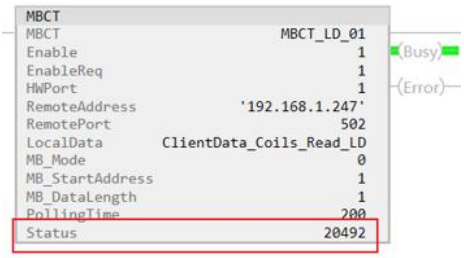
4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：



Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据

0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败

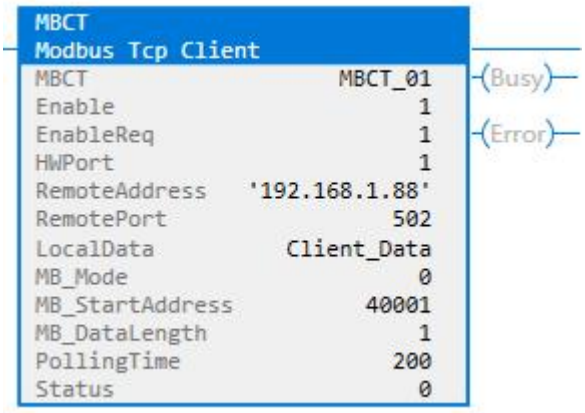
参数错误	
0x6000	**参数无效**
0x6001	**MODBUS 数据区定义无效**
0x6002	**MODBUS 参数 MB_RWMode 值无效**
0x6003	**MODBUS 参数 MB_StartAddress 无效**
0x6004	**MODBUS 参数 MB_DataLength 无效**
0x6005	**MODBUS 参数本地地址无效**
0x6006	**MODBUS 参数 PollingTime 不合法**
0x6007	**MODBUS 本端口正在处理其他 modbus 请求**
0x6008	**MODBUS 后台 tcpserver 启动失败**
0x6009	**SOCKET 句柄无效**
0x600A	**SOCKET 指令目标 IP 地址不合法**
用户操作错误	
0x7000	**用户操作错误**
0x7001	**MODBUS 用户在线修改参数**
0x7002	**MODBUS 后台实例不存在**
0x7003	**MODBUS 后台实例超过上限**
0x7004	**MODBUS 用户指定网络地址参数不合法**
0x7005	**MODBUS 用户指定数据模式不正确**
0x7006	**MODBUS 用户提供的参数在本地数据区越界了**
0x7007	**MODBUS 用户在线修改后收到了在线修改前的包**

0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：

下图演示了在本地 1 号网口上使用 MBCT 客户端指令，访问远程 IP 地址为’192.168.1.88’且远程端口为 502 的 Modbus TCP Server 服务器，访问模式为读取(MB_Mode=0)，访问服务器上起始地址为 40001(MB_StartAddress=40001)且寄存器个数为 1(MB_DataLength=1)，本地数据区内容存储在变量名为 Client_Data(本例变量类型为 INT[XX]数组)的变量上。



名称	数据类型
MBCT_01	MBCT

名称	数据类型
Client_Data	INT[5]

ST 示例：

ST 实现前面梯形图实例同样功能。

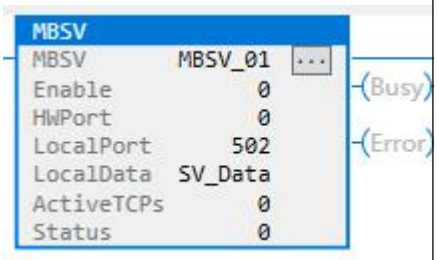
```
MBCT_1(LocalData:=any_in,              //Modbus 读/写存储的内容
       Enable:=bool_in,                //是否建立与服务端的连接
       EnableReq:=bool_in,             //是否开启对服务
       HWPort:=int_in,                 //物理网口
       LocalPort:=dint_in,             //本地端口
       RemoteAddress:=string_in,//目标 IP
       RemotePort:=dint_in,            //目标端口
       UnitID:=sint_in,                //odbusTCP 网关下面串口 Modbus 设
备地址
       MB_Mode:=sint_in,               //访问模式
       MB_StartAddress:=dint_in, //Modbus 起始地址
       MB_DataLength:=int_in,          //Modbus 数据长度
       PollingTime:=dint_in,           //轮询数据更新周期,ms 毫秒
       Timeout:=dint_in,               //指令执行超时设定参数(毫秒)
       Status=>dint_out,               //状态与故障码
       Busy=>bool_out,                 //与服务端正交换 MB 数据
       Error=>bool_out,                //指令出错
```

```
Connected=>bool_out,      //与服务端连接上
RequestCount=>dint_out,   //统计请求次数
SuccessCount=>dint_out);  //统计成功次数
```

Modbus 服务器 MBSV 指令

启用 MODBUS/TCP 服务器端的服务。

1）指令格式

指令	名称	梯形图表现	ST 表现
MBSV	ModbusServer 服务端指令		MBSV_1(LocalData:=, Enable:=, HWPort:=, LocalPort:=, Timeout:=, IR_StartAddress:=, HR_StartAddress:=, DI_StartAddress:=, DO_StartAddress:=, ActiveTCPs=>, Status=>, Busy=>, Error=>, Connected=>, RequestCount=>, SuccessCount=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
MBSV	指令变量	MBSV	X	X	变量	指令的背景数据变量
LocalData	本地数据	MBSV_DATA	-32768~32767	0	变量	提供给 ModbusTCP 客服端访问的数据区变量结构体，包含了 4 个独立的数组 Coils[], Discinputs[], InpRegister[], HoldRegister[]

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
bEnable	服务使能	BOOL	0-1	0	变量或者立即数	是否开启 MBSV 服务

HWPort	网口号	INT	0-3	0	变量或者立即数	PLC 硬件网口号，取值范围 0/1/2/3, 0 代表使用所有网口
LocalPort	本地端口号	DINT	502~566	502	变量或者立即数	本地端口，默认值 502
Timeout	超时时间	DINT	60000~180000	60000	变量或者立即数	超时，ms，默认值 60000，隐藏接口
IR_StartAddress	IR 偏移地址	DINT	0-65535	0	变量或者立即数	InputRegister 区起始偏移地址
HR_StartAddress	HR 偏移地址	DINT	0-65535	0	变量或者立即数	HoldRegister 区起始偏移地址
DI_StartAddress	DI 偏移地址	DINT	0-65535	0	变量或者立即数	DiscInputs 区起始偏移地址
DO_StartAddress	DO 偏移地址	DINT	0-65535	0	变量或者立即数	Coils 区起始偏移地址

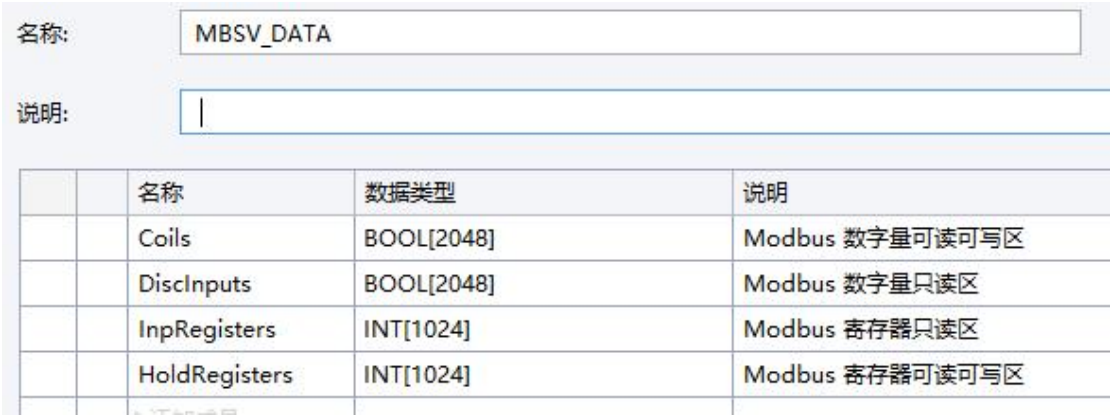
输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
ActiveTCPs	连接数量	DINT	0-64	0	变量	有多少 TCP 连接被建立
Status	状态	DINT	0x0000-0x700B	0	变量	状态与故障码,原 StatusID

Busy	**指令忙**	**BOOL**	**0-1**	**0**	**变量**	**Server 准备好,原 WT(Waiting)**
Connected	**已连接**	**BOOL**	**0-1**	**0**	**变量**	**至少有一个客户端连接上，原 AC(Accepted),不显示在 bitLeg**
Error	**错误**	**BOOL**	**0-1**	**0**	**变量**	**显示指令出错，错误原因详细信息在 StatusID 里面可查看**
RequestCount	**请求次数**	**DINT**	**DINT 正范围**	**0**	**变量**	**统计请求次数**
SuccessCount	**成功次数**	**DINT**	**DINT 正范围**	**0**	**变量**	**统计成功次数**

其它数据类型：

MBSV_DATA 类型成员说明（Modbus Server 的本地服务数据区数据类型）：



3）功能说明

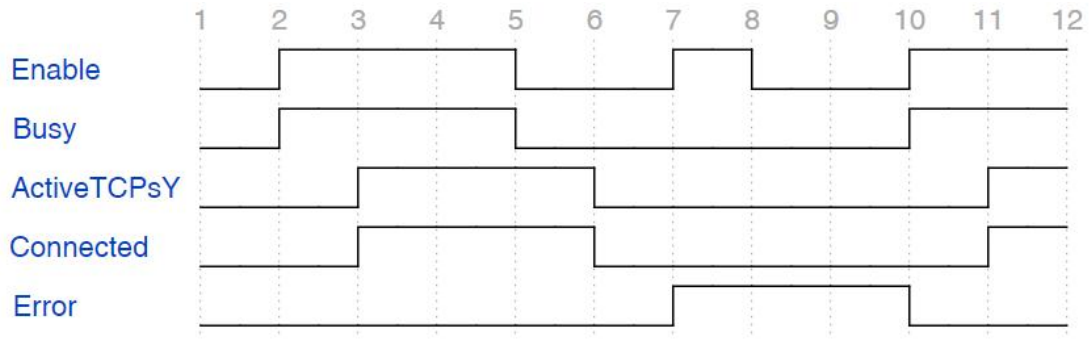
接受来自符合 Modbus TCP 标准协议的客户端 Client 的访问。可指定任意网口(默认所有网口)，任意端口(默认 502)

单个 Server 指令能提供 2048 个可读写数字量(Coils)，2048 个可读数字量(DiscInputs), 1024 个可读写寄存器(保持寄存器 HoldRegisters)，1024 个可读寄存器(输入寄存器 InputRegisters)的访问区

最大可开 64 个 Server 实例(大部分应用场合通常 1 个实例够用，若 PLC 只有一个网口，可通过开启不同的端口来开启多个 Server 服务，比如开启 502,503,504 等端口，虽然默认是 502 端口)

水龙头式服务开启方式

时序图



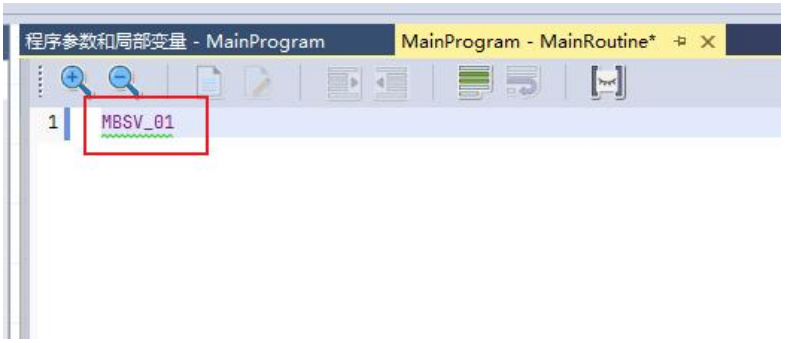
4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

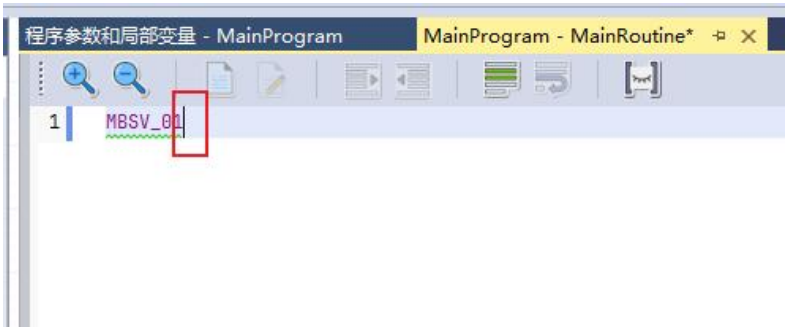
ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

5、 变量表区建立 MBSV 指令对应的指令实例变量：

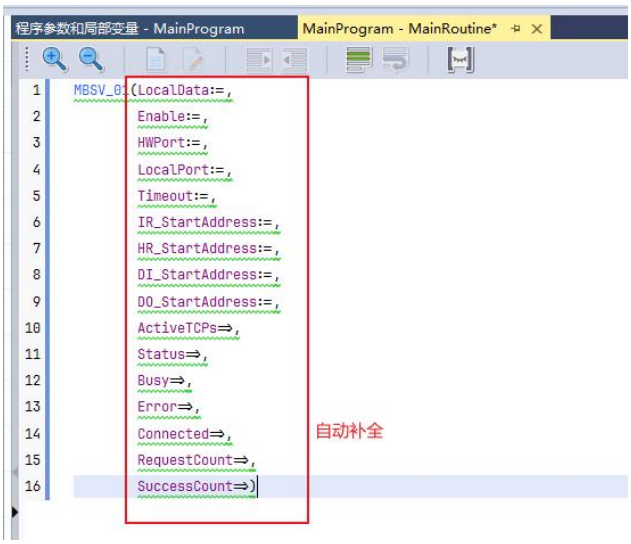
6、 ST 编辑区输入对应 MBSV 实例变量名：



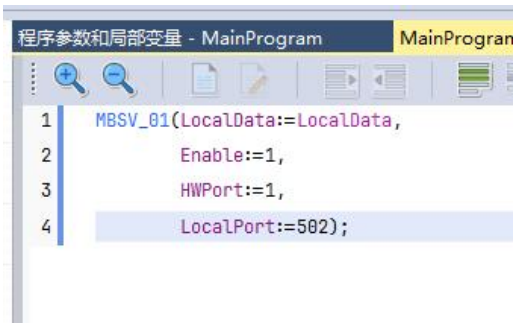
7、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令报错）：



按下 tab 键系统会自动补全引脚内容：

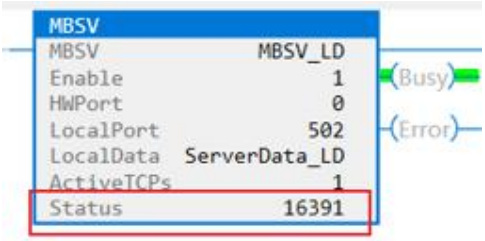


用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：



5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：



Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误

正常状态信息	
0x4000	**正常状态**
0x4001	**连接已建立**
0x4002	**连接已终止**
0x4003	**未激活任何作业且未建立任何连接**
0x4004	**已触发连接建立操作**
0x4005	**正在建立连接**
0x4006	**正在终止连接**
0x4007	**连接已建立但未激活任何作业**
0x4008	**正在发送数据**
0x4009	**正在接收数据**
0x400A	**超时**
0x400B	**连接需要重置(内部自动或者用户手动)**
0x400C	**端口无 ip**
协议错误	
0x5000	**协议错误**
0x5001	**MODBUS 服务端没有在指定时间内回复**
0x5002	**MODBUS 收到的 modbus 帧错误**
0x5003	**MODBUS 不支持的功能码**
0x5004	**MODBUS 帧头中填入的数据长度非法**

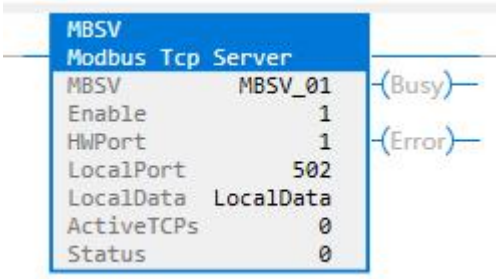
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败

0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图 1：

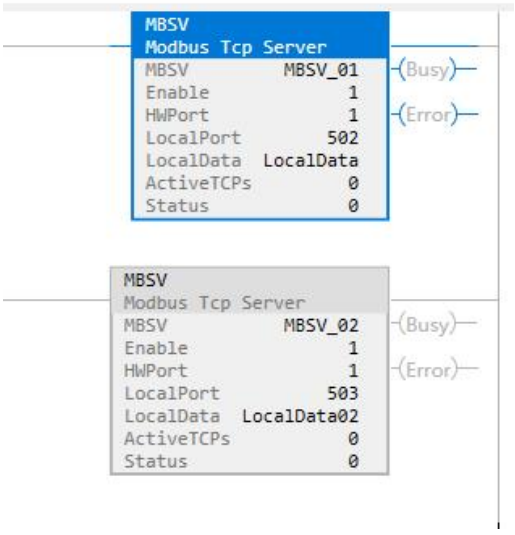
下图演示了在本地 1 号网口上，使用本地端口 502，开启 Modbus Server 服务，本地数据区内容存储在变量名为 LocalData(变量类型为 MBSV_DATA)的变量上。



名称	数据类型
MBSV_01	MBSV
LocalData	MBSV_DATA

梯形图 2：

下图演示了本地 1 号网口上，使用本地端口 502 和 503，开启 2 个独立 Modbus Server 服务，本地数据区内容存储在变量名为 LocalData 和 LocalData02(变量类型为 MBSV_DATA)的变量上。



名称	数据类型
MBSV_01	MBSV
LocalData	MBSV_DATA
MBSV_02	MBSV
LocalData02	MBSV_DATA

ST 示例：

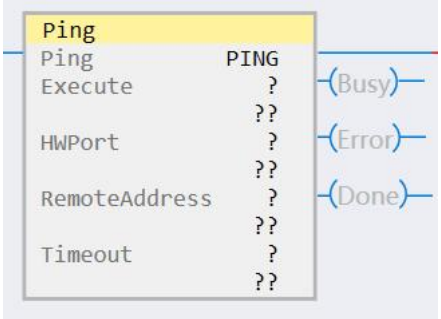
下图演示了在本地 1 号网口上，使用本地端口 502，开启 Modbus Server 服务，本地数据区内容存储在变量为 LocalData(变量类型为 MBSV_DATA)的变量上。

```
MBSV_1(LocalData:=mbsv_data_inout,//提供给 ModbusTCP 客服端访问的数据区变量
        Enable:=bool_in,              //是否开启 MBSV 服务
        HWPort:=int_in,               //物理网口
        LocalPort:=dint_in,           //本地端口
        Timeout:=dint_in,             //指令执行超时设定参数(毫秒)
        IR_StartAddress:=dint_in,     //InputRegister 区起始偏移地址
        HR_StartAddress:=dint_in,     //HoldRegister 区起始偏移地址
        DI_StartAddress:=dint_in,     //DiscInputs 区起始偏移地址
        DO_StartAddress:=dint_in,     //Coils 区起始偏移地址
        ActiveTCPs=>dint_out,         //TCP 连接被建立数
        Status=>dint_out,             //状态与故障码
        Busy=>bool_out,                   //Server 准备好
        Error=>bool_out,              //至少有一个客户端连接上
        Connected=>bool_out,          //指令出错
        RequestCount=>dint_out,           //统计请求次数
        SuccessCount=>dint_out);      //统计成功次数
```

Ping 指令

调用系统 PING 指令，并返回结果。

1）指令格式

指令	名称	梯形图表现	ST 表现
Ping	PING		Ping_1(Execute:=, HWPort:=, RemoteAddress:=, Timeout:=, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Ping	本地数据	Ping	X	X	变量	

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Done	完成	BOOL	0-1	0	变量或 立即数	执行成功状态位
Busy	进行中	BOOL	0-1	0	变量或	正在运行状态位

					立即数	
Error	错误	BOOL	0-1	0	变量或 立即数	执行错误状态
Status	错误码	DINT	0x0000-0xFFFF	0	变量	错误/状态代码字

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0-1	0	变量 或 立即 数	触发执行控制位，上升沿触发
HWPort	通信口	INT	1、2、3	1	变量 或 立即 数	1/2/3 代表物理网口
RemoteAddress	远程地址	STRING	对应类型范围	‘ ’	变量 或 立即 数	目标 IP 地址
Timeout	超时	DINT	0-60000	5000	变量 或	指令执行超时设定参数(毫秒)

					立即 数	

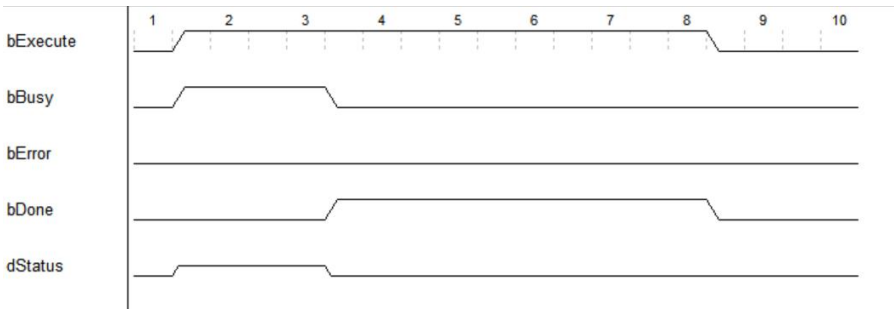
3）功能说明

1. 功能通过 Execute 上升沿触发。
2. RemoteAddress：存储有指定通信对象的 IP 地址。
3. Error：返回故障标志位。如果没有收到 4 个回复，则认为失败。
4. Busy：等待回复，输出 true；超时返回或者收到回复，输出 false。
5. Execute：信号复位，触发 Error/Status/Busy 复位。
6. HWPort：1，2，3 分别指定 PLC 的 3 个网口
7. Done：指令返回，执行完成。如果 Timeout 内收到 4 个报文，则成功

时序图

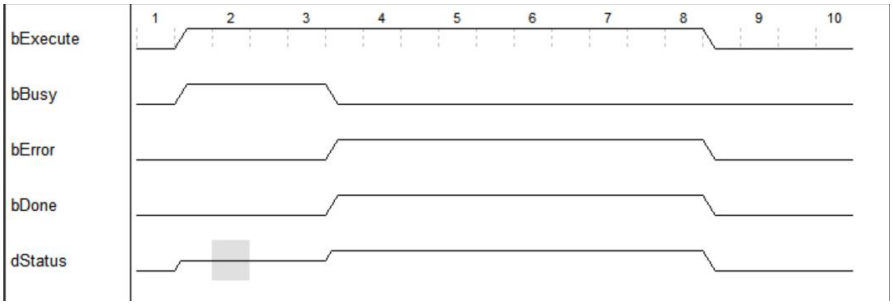
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
PING(Execute:=Execute_1,
     HWPort:=0,
     RemoteAddress:='192.168.1.110',
     Timeout:=5000,
     Done⇒Done_1,
     Busy⇒Busy_1,
     Error⇒Error_1,
     Status⇒Status_1);
```

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止

0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”

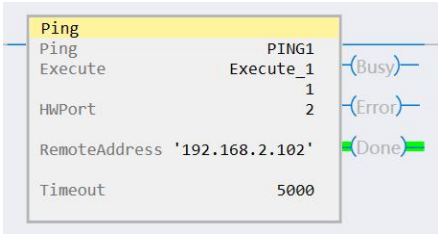
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误

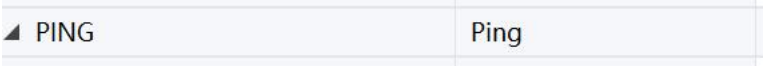
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：

大部分情况下，Ping 仅需配置需要的远程地址，其余参数直接新建变量按默认值即可使用。该示例中 RemoteAddress(远程地址)为’192.168.2.102’；HWPort 设置通讯口为 2；Timeout 设置超时时间为 5000ms。





◢ PING.RemoteAddress	STRING
▷ PING.RemoteAddress.LEN	DINT

ST 示例：

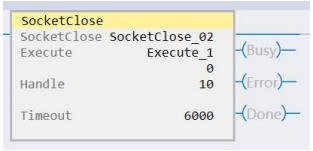
ST 实现前面梯形图实例同样功能。

```
Ping_1(Execute:=bool_in,          //触发执行控制位，上升沿触发
       HWPort:=int_in,            //物理网口
       RemoteAddress:=string_in,//目标 IP 地址
       Timeout:=dint_in,          //指令执行超时设定参数(毫秒)
       Done=>bool_out,              //执行成功状态位
       Busy=>bool_out,            //正在运行状态位
       Error=>bool_out,           //执行错误状态
       Status=>dint_out);         //错误/状态代码字
```

关闭 Socket SocketClose 指令

关闭指定的 SOCKET 连接通道。

1）指令格式

指令	名称	梯形图表现	ST 表现
SocketClose	Socket 关闭		SocketClose_1(Execute:=, Handle:=, Timeout:=, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
SocketClose	指令变量	SocketClose	X	X	变量	指令的背景数据变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0/1	0	变量或者	触发执行控制位，上升沿触发

					立即数	
Handle	句柄	DINT	0-65536	0	变量或者 立即数	输入句柄
Timeout	超时	DINT	0-60000	5000	变量或者 立即数	指令执行超时设定参数(毫秒)

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Done	进行中	BOOL	0/1	0	变量或者 立即数	执行成功状态位
Busy	错误	BOOL	0/1	0	变量或者 立即数	正在运行状态位
Error	完成	BOOL	0/1	0	变量或者 立即数	执行错误状态

Status	错误码	DINT	0x0000-0xFFFF	0	变量或者立即数	错误/状态代码字

3）功能说明

1、 Execute：上升沿触发关闭 TCP 连接。根据 Handle 参数，关闭对应的 Socket 链接。

2、 Handle：监听/接受/接收的 Handle；。

3、 Busy：指令正在执行状态。

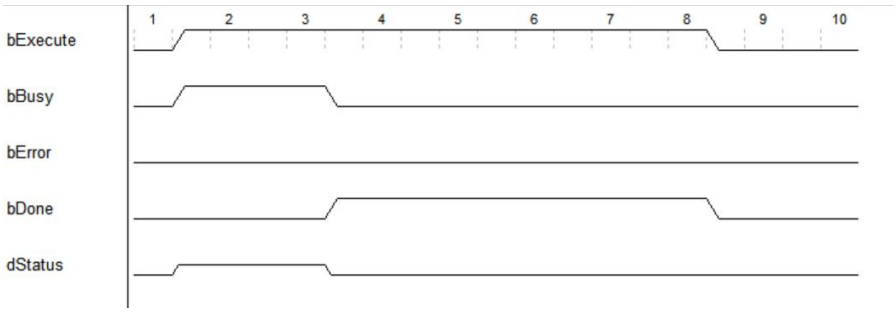
4、 Error：故障信号输出。

5、 Done：指令成功执行。

时序图

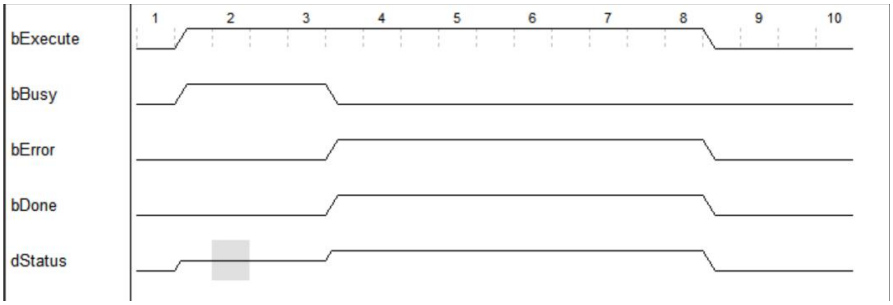
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

1、 变量表区建立 SocketClose 指令对应的指令实例变量：

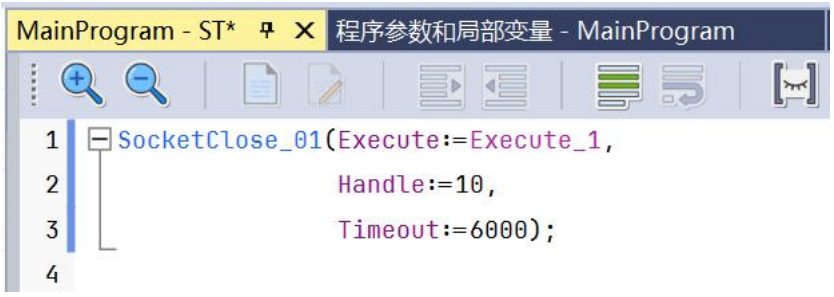
名称	数据类型
SocketClose_01	SocketClose

2、 ST 编辑区输入对应 SocketClose 实例变量名：

3、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令报错）：

按下 tab 键系统会自动补全引脚内容：

用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：



5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
SocketClose_01(Execute:=bool_in,
               Handle:=dint_in,
               Timeout:=dint_in,
               Done⇒bool_out,
               Busy⇒bool_out,
               Error⇒bool_out,
               Status⇒dint_out);
```

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	

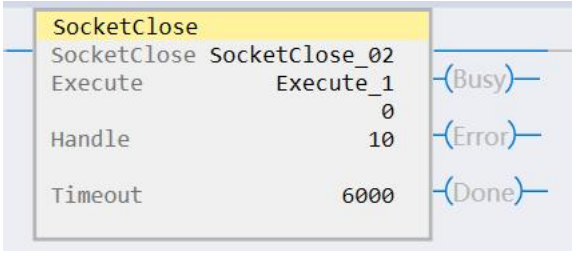
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误

0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法

用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：



ST 示例：

SocketClose_1(Execute:=bool_in,　//触发执行控制位，上升沿触发

Handle:=dint_in，//输入句柄

Timeout:=dint_in,//指令执行超时设定参数(毫秒）

Done=>bool_out，//执行成功状态位

Busy=>bool_out，//正在运行状态位

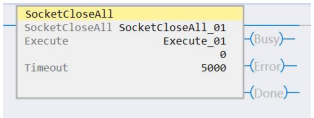
Error=>bool_out，//执行错误状态

Status=>dint_out);//错误/状态代码字

关闭所有 Socket SocketCloseALL 指令

关闭当前建立的所有 SOCKET 连接。

1）指令格式

指令	名称	梯形图表现	ST 表现
SocketCloseAll	Socket 全部关闭		SocketCloseAll_1(Execute:=, Timeout:=, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
SocketCloseAll	指令变量	SocketCloseAll	X	X	变量	指令的背景数据变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0/1	0	变量或者	触发执行控制位，上升

					立即数	沿触发
Timeout	超时	DINT	0-60000	5000	变量或者 立即数	指令执行超时设定参数(毫秒)

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Done	进行中	BOOL	0/1	0	变量或者 立即数	执行成功状态位
Busy	错误	BOOL	0/1	0	变量或者 立即数	正在运行状态位
Error	完成	BOOL	0/1	0	变量或者 立即数	执行错误状态
Status	错误码	DINT	0x0000-0xFFFF	0	变量或者 立即数	错误/状态代码字

3）功能说明

1、 Execute：上升沿触发关闭系统中的所有已建立的 Socket 连接。

2、 SocketComm： 用于存储相关信息状态。

3、 Busy：指令正在执行状态。

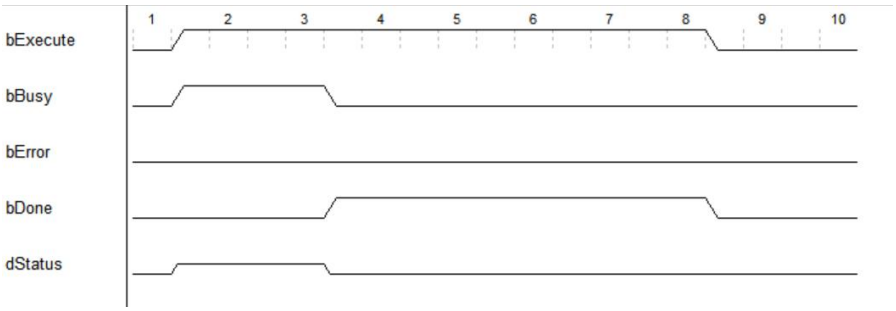
4、 Error:故障信号输出。

5、 Done：指令成功执行。

时序图

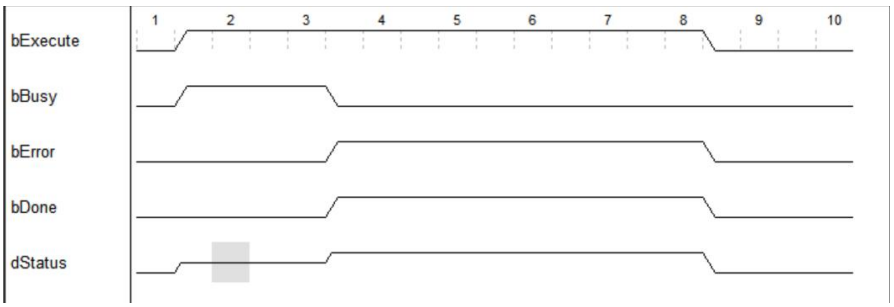
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

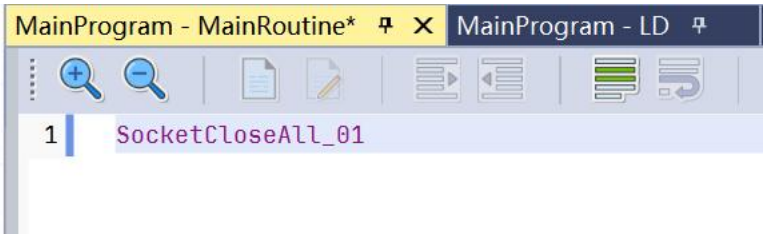
LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

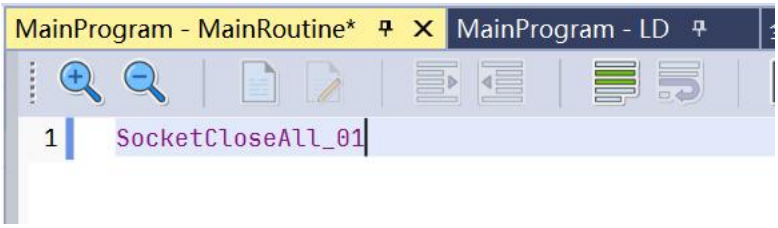
1、 变量表区建立 SocketCloseAll 指令对应的指令实例变量：

名称	数据类型
▷ SocketCloseAll_01	SocketCloseAll

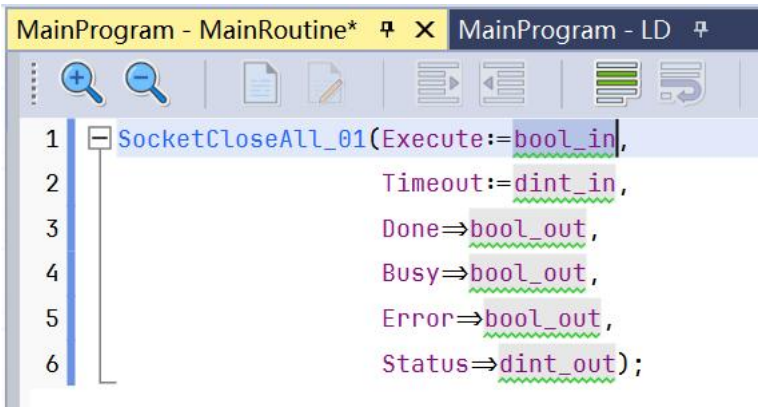
2、 ST 编辑区输入对应 SocketCloseAll 实例变量名：



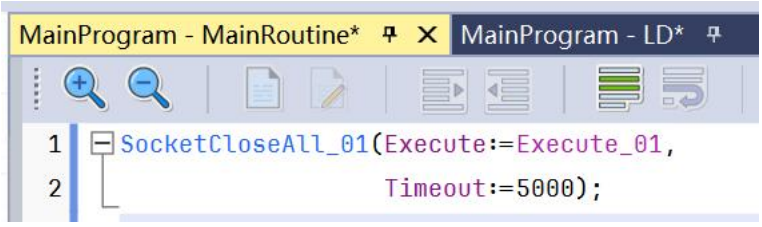
3、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令报错）：



按下 tab 键系统会自动补全引脚内容：



用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：



5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
SocketCloseAll_01(Execute:=bool_in,
                  Timeout:=dint_in,
                  Done⇒bool_out,
                  Busy⇒bool_out,
                  Error⇒bool_out,
                  Status⇒dint_out);
```

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)

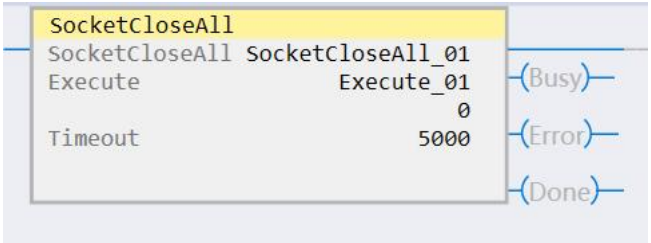
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效

0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接

0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：



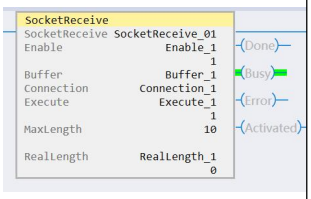
ST 示例

```
SocketCloseAll_1(Execute:=bool_in, //触发执行控制位，上升沿触发
                 Timeout:=dint_in, //指令执行超时设定参数(毫秒)
                 Done=>bool_out,  //执行成功状态位
                 Busy=>bool_out,  //正在运行状态位
                 Error=>bool_out,  //执行错误状态
                 Status=>dint_out); //错误/状态代码字
```

Socket 建立通信并接收 SocketReceive 指令

建立 Socket 通信，并接收数据报文。

1）指令格式

指令	名称	梯形图表现	ST 表现
SocketReceive	Socket 建立通信并接收指令		SocketReceive_1(Buffer:=, Connection:=, Enable:=, Execute:=, MaxLength:=, RealLength=>, Activated=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
SocketReceive	指令变量	SocketReceive	X	X	变量	指令的背景数据变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Connection	连接	SocketConnection	X	X	变量	连接参数，包含 5 个成员 HWPort、LocalPort、RemoteAddress、RemotePort、Type
Enable	使能	BOOL	0/1	0	变量	指令使能，1=指令和通信连接打开，0=指令关闭通信资源，并关闭接收允许
Execute	执行	BOOL	0/1	0	变量	1=打开接收 0=关闭接收
MaxLength	最大长度	int	0-1024	1	变量或者立即数	接收最大长度设定

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Buffer	缓冲区	指针	X	X	变量或者立即数	接收缓冲区

Activated	建立连接	BOOL	0/1	0	变量或者立即数	1=通信建立状态
Done	完成	BOOL	0/1	0	变量或者立即数	1=接收动作完成,此状态将仅显示一个周期
Busy	指令忙	BOOL	0/1	0	变量或者立即数	1=接收中忙碌
Error	错误	BOOL	0/1	0	变量或者立即数	1=错误状态
Status	状态	int	0x0000-0xFFFF	0	变量或者立即数	状态&错误代码
RealLength	读取长度	int	0-1024	0	变量	实际接收到的长度

3）功能说明

1、Enable：为 1 的时候，建立通信。待通信建立后，Execute 的命令生效。Enable 为 0 的时候，关闭通信，Execute 的命令失效。

2、Activated：为 1 代表通信实际打开状态，为 0 代表通信实际关闭状态。通信正常

状态下，且 Execute=1 的时候，指令执行数据接收，当 Done 置位后，指令继续执行下一次的数据接收，反复循环。Execute=0 的时候，指令不执行数据接收。

3、 Execute：通信正常状态下，且 Execute=1 时，指令执行数据接收，当 Done 置位后，指令继续执行下一次的数据接收，反复循环。Execute=0 的时候，指令不执行数据接收。

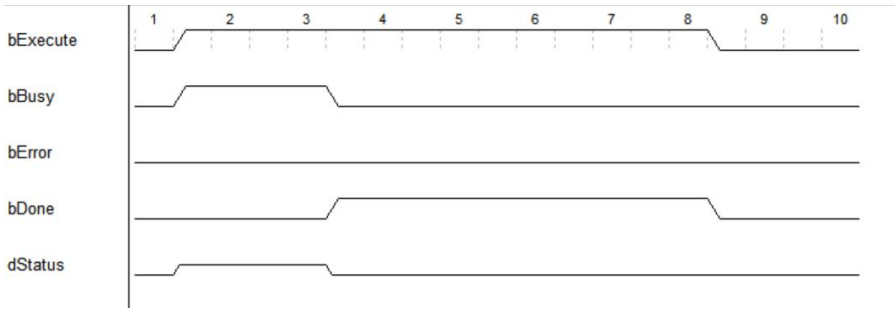
4、 当指令执行内出现错误时：如 Create，Listen, Accept, Connect,Send,Receive 等内嵌操作出现错误，甚至不执行 Execute 指令内部也能检测到对方关闭通信资源，或拔了网线的情况； 指令内部视情况执行 SocketClose 并回归初始状态。此时当 Enable=0 的时候，保持此初始状态； 当 Enable=1 的时候，指令重新建立通信。

5、 用户使用行为：通信正常 Activated=1 时，用户设置 Execute=1 执行循环接收，当检测到 Done 信号时，用户通过 Buffer 和 RealLength 获得字符串的接收结果。当 Socket 接受区有数据的时候，Done 完成将获得一个 Length>0 的字符串。当 Socket 接受区无数据的时候，指令内部持续接收并经过一个固定的超时后，指令仍旧没有接收到数据，Done 完成将获得一个 Length=0 的空字符串。用户设置 Execute=0 关闭接收。

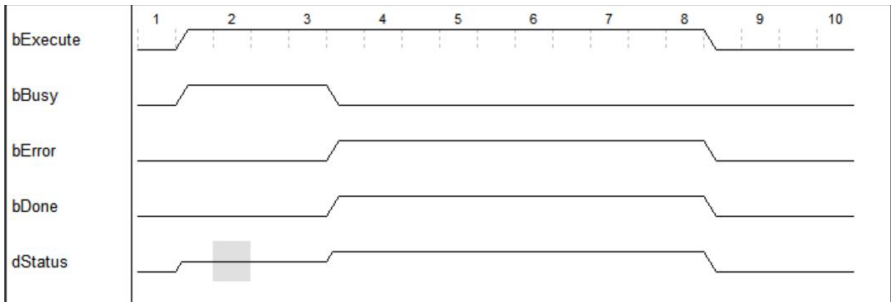
时序图

下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。时序图如下：



4）注意事项

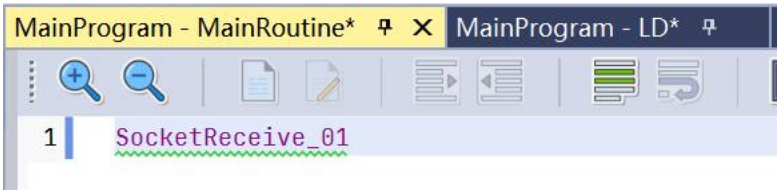
LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

1、 变量表区建立 SocketReceive 指令对应的指令实例变量：

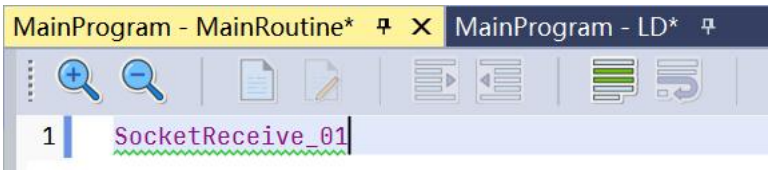
名称	数据类型
▷ SocketReceive_01	SocketReceive

2、 ST 编辑区输入对应 SocketReceive 实例变量名：

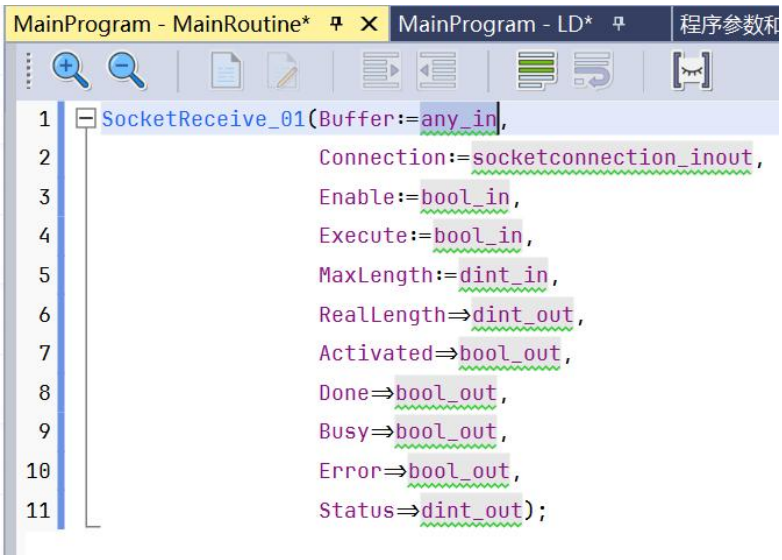


3、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令

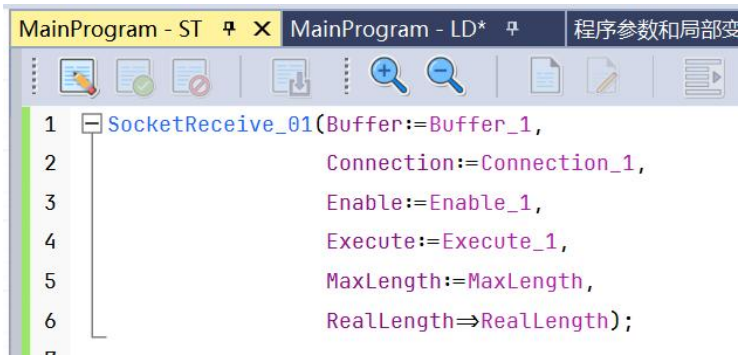
报错）：



按下 tab 键系统会自动补全引脚内容：



用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：



5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
SocketReceive_01(Buffer:=any_in,
                 Connection:=socketconnection_inout,
                 Enable:=bool_in,
                 Execute:=bool_in,
                 MaxLength:=dint_in,
                 RealLength⇒dint_out,
                 Activated⇒bool_out,
                 Done⇒bool_out,
                 Busy⇒bool_out,
                 Error⇒bool_out,
                 Status⇒dint_out);
```

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时

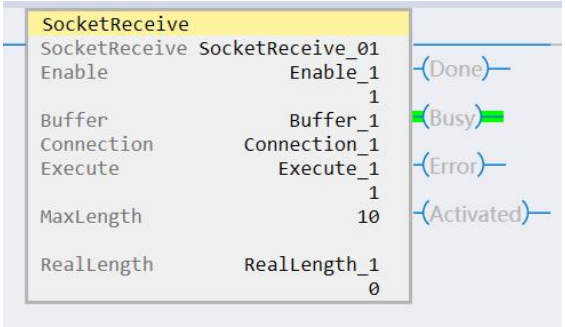
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	

0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法

0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：



ST 示例：

```
SocketReceive_1(Buffer:=any_in,                    //接收缓冲区
                Connection:=socketconn_inout, //连接参数
                Enable:=bool_in,                   //触发执行控制位，上升沿触发
                Execute:=bool_in,                  //接收执行
                MaxLength:=dint_in,                //接收最大长度设定
                RealLength=>dint_out,              //实际接收到的长度
                Activated=>bool_out,               //通信建立状态
                Done=>bool_out,                    //接收动作完成
                Busy=>bool_out,                    //接收中忙碌
```

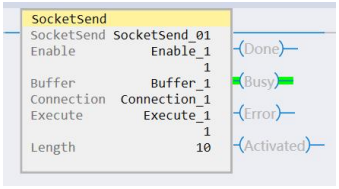
Error=>bool_out,　　//错误状态

Status=>dint_out);　　//状态&错误代码

Socket 建立通信并发送 SocketSend 指令

建立 Socket 通信，并发送数据报文。

1）指令格式

指令	名称	梯形图表现	ST 表现
SocketSend	Socket 建立通信并发送指令		SocketSend_1(Buffer:=, Connection:=, Enable:=, Execute:=, Length:=, Activated=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
SocketSend	指令变量	SocketSend	X	X	变量	指令的背景数据变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Connection	连接	SocketConnection	X	X	变量或者立即数	连接参数，包含 5 个成员 HWPort、LocalPort、RemoteAddress、RemotePort、Type
Enable	使能	BOOL	0/1	0	变量	指令使能，1=指令和通信连接打开，0=指令关闭通信资源，并关闭发送允许
Execute	执行	BOOL	0/1	0	变量	发送数据，上升沿执行
Buffer	缓冲区	指针	X	X	变量	发送缓冲区
Length	发送长度	int	1-32767	1	变量或者立即数	发送数据长度

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明

Activated	建立连接	BOOL	0/1	0	变量或者立即数	1=通信建立状态
Done	完成	BOOL	0/1	0	变量或者立即数	1=发送动作完成
Busy	指令忙	BOOL	0/1	0	变量或者立即数	1=发送中忙碌
Error	错误	BOOL	0/1	0	变量或者立即数	1=错误状态
Status	状态	int	0x0000-0xFFFF	0	变量或者立即数	状态&错误代码

3）功能说明

1、Enable：为 1 的时候，建立通信。待通信建立后，Execute 的命令生效。Enable 为 0 的时候，关闭通信，Execute 的命令失效。

2、Activated：为 1 代表通信实际打开状态，为 0 代表通信实际关闭状态。通信正常状态下，且 Execute=1 的时候，指令执行数据接收，当 Done 置位后，指令继续执行下一次的数据接收，反复循环。Execute=0 的时候，指令不执行数据接收。

3、 Execute：通信正常状态下，Execute 上升沿，指令执行数据发送。

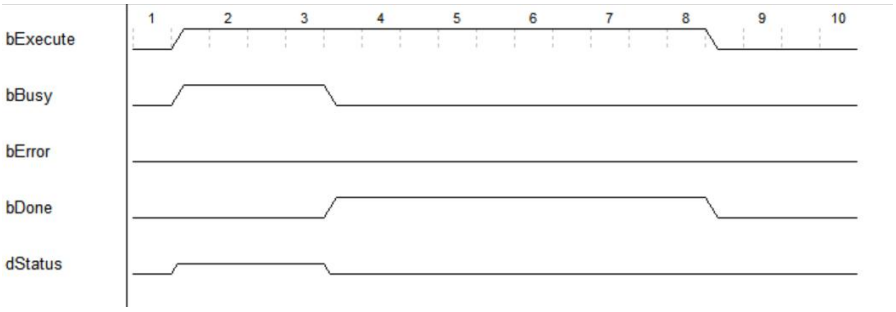
4、 当指令执行内出现错误时：如 Create，Listen, Accept, Connect,Send,Receive 等内嵌操作出现错误，甚至不执行 Execute 指令内部也能检测到对方关闭通信资源，或拔了网线的情况； 指令内部视情况执行 SocketClose 并回归初始状态。此时当 Enable=0 的时候，保持此初始状态； 当 Enable=1 的时候，指令重新建立通信。

5、 用户使用行为：通信正常 Activated=1 时，设置 Execute=1，指令执行数据发送；用户收到 Done 后，将 Execute 设置 0 复位发送操作。 下次需要再次发送时，用户设置 Execute=1。

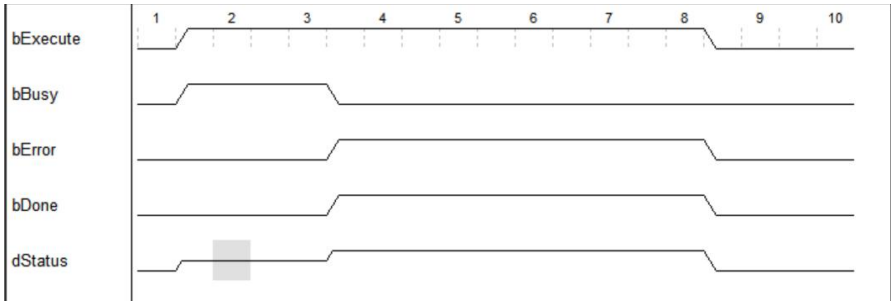
时序图

下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。时序图如下：



4）注意事项

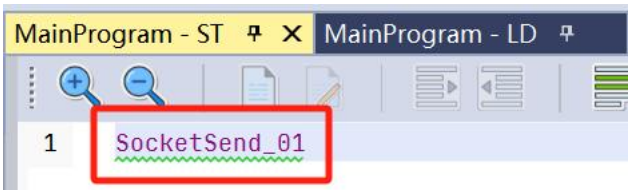
LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

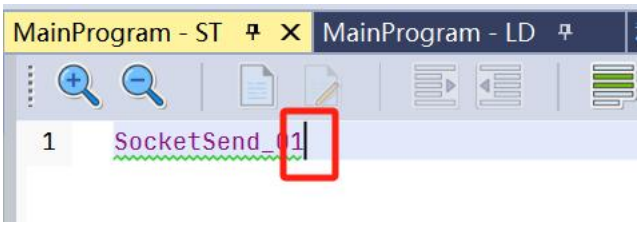
1、 变量表区建立 SocketSend 指令对应的指令实例变量：

名称	数据类型
SocketSend_01	SocketSend

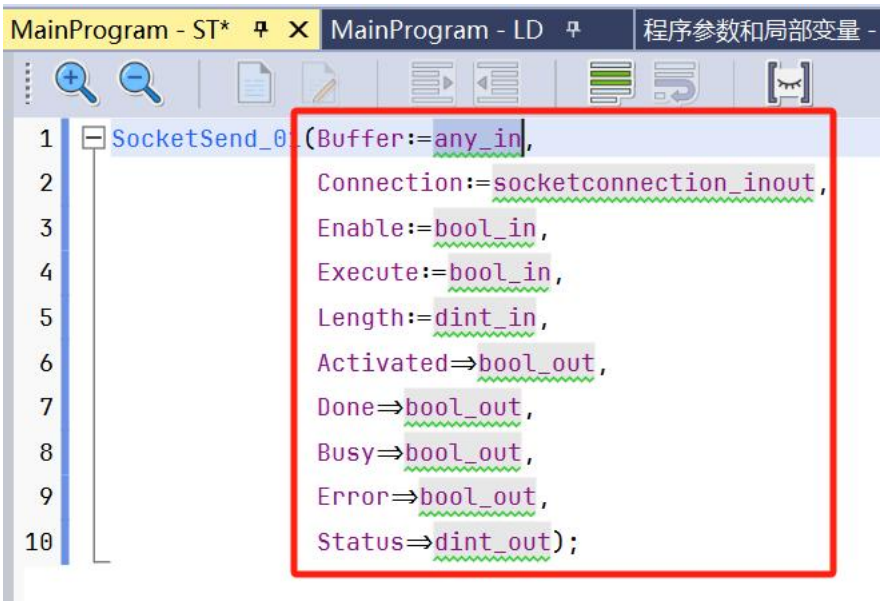
2、 ST 编辑区输入对应 SocketSend 实例变量名：



3、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令报错）：



按下 tab 键系统会自动补全引脚内容：



用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：

5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
SocketSend_01(Buffer:=Buffer_1,
              Connection:=Connection_1,
              Enable:=Enable_1,
              Execute:=Execute_1,
              Length:=Length_1,
              Activated⇒Activated_1,
              Done⇒Done_1,
              Busy⇒Busy_1,
              Error⇒Error_1,
              Status⇒Status_1)
```

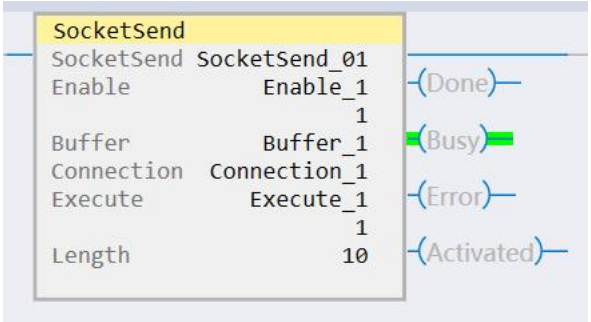
Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误

0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法

0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：



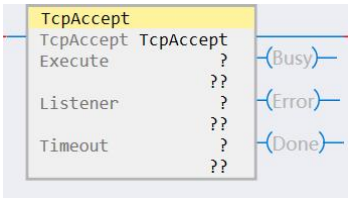
ST 示例：

```
SocketSend_1(Buffer:=any_in,                    //发送缓冲区
             Connection:=socketconn_inout,      //连接参数
             Enable:=bool_in,                   //指令使能
             Execute:=bool_in,                  //发送数据，上升沿执行
             Length:=dint_in,                   //发送数据长度
             Activated=>bool_out,               //通信建立状态
             Done=>bool_out,                    //发送动作完成
             Busy=>bool_out,                    //发送中忙碌
             Error=>bool_out,                   //错误状态
             Status=>dint_out);                 //状态&错误代码
```

TCP 服务器接收连接 TcpAccept 指令

TCP 服务器接受客户端的连接请求。

1）指令格式

指令	名称	梯形图表现	ST 表现
TcpAccep	TCP 服务器 接收指令		TcpAccept_1(Execute:=, Listener:=, Timeout:=, Handle=>, RemoteAddress=>, RemotePort=>, HWPort=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0-1	0	变量	触发执行控制位，上升沿触发
Listener	监听者	DINT	0-65536	0	变量或	输入句柄

					立即数	
Timeout	超时	DINT	0-60000	5000	变量或 立即数	指令执行超时设定参数 (毫秒)

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Handle	句柄	DINT	0-65536	0	变量	输出句柄
RemoteAddress	目标 IP	STRING	合法 IP	''	变量	目标 IP 地址
RemotePort	目标端口	DINT	0-65536	0	变量	目标端口
HWPort	通信口	INT	1-3	1	变量	1/2/3 代表物理 网口
Done	完成	BOOL	0-1	0	变量	执行成功状态位
Busy	数据交换中	BOOL	0-1	0	变量	正在运行状态位
Error	错误	BOOL	0-1	0	变量	执行错误状态
Status	状态	DINT	0x0000-0xFFFF	0	变量	错误/状态代码 字

3）功能说明

1. Execute：上升沿触发接收数据。指令根据 ListenPoint 提供的参数，接受收到的客户端连接请求，建立 Socket 链接。
2. Handle：监听指令生成的 Handle 传递到此参数中。
3. Timeout：指令返回超时时间，单位毫秒 。
4. Busy：指令正在执行状态。

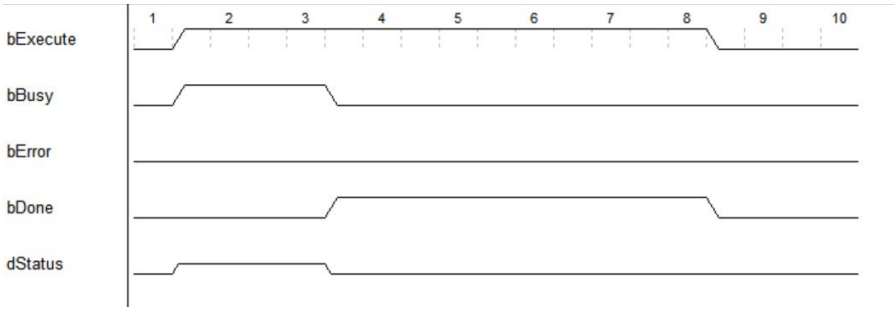
5. Error:故障信号输出。

6. Done：指令成功执行。

时序图

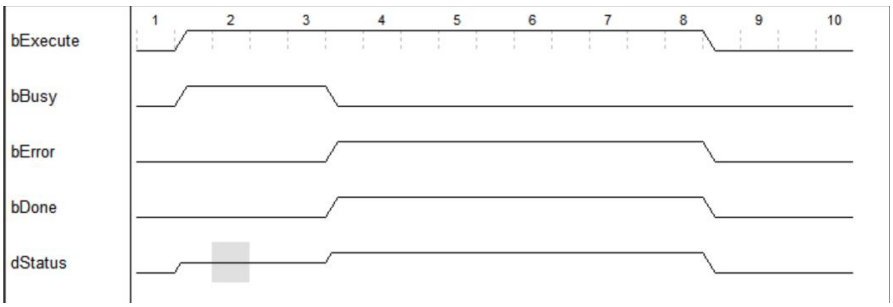
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1.正确执行返回的时序如下：



2.Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位，对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
//正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
//协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法

0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
//参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效

0x600A	SOCKET 指令目标 IP 地址不合法
//用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

ST 示例

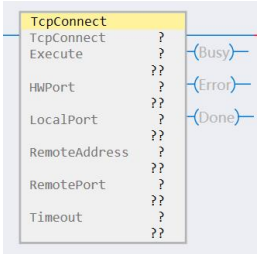
```
TcpAccept_1(Execute:=bool_in,          //触发执行控制位，上升沿触发
            Listener:=dint_in,         //输入句柄
            Timeout:=dint_in,          //指令执行超时设定参数(毫秒)
            Handle=>dint_out,          //输出句柄
            RemoteAddress=>string_out,  //目标 IP 地址
```

```
RemotePort=>dint_out,      //目标端口
HWPort=>int_out,           //物理网口
Done=>bool_out,                 //执行成功状态位
Busy=>bool_out,                 //正在运行状态位
Error=>bool_out,           //执行错误状态
Status=>dint_out);         //错误/状态代码字
```

TCP 连接指令 TcpConnect 指令

获取系统资源，并建立与服务器的 TCP 连接。

1）指令格式

指令	名称	梯形图表现	ST 表现
TcpConnect	TCP 连接指令		TcpConnect_1(Execute:=, HWPort:=, LocalPort:=, RemoteAddress:=, RemotePort:=, Timeout:=, Handle=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0-1	0	变量	触发执行控制位，上升沿触发
HWPort	通信口	INT	1-3	1	变量或	1/2/3 代表物理网口

					立即数	
LocalPort	本地端口	DINT	0-65536	0	变量或立即数	本地端口
RemoteAddress	远程地址	STRING	合法 IP	''	变量或立即数	目标 IP 地址
RemotePort	远程端口	DINT	0-65536	0	变量或立即数	目标端口
Timeout	超时	DINT	0-60000	45000	变量或立即数	指令执行超时设定参数(毫秒)

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Handle	句柄	DINT	0-65536	0	变量	输出句柄
Done	完成	BOOL	0-1	0	变量	执行成功状态位
Busy	数据交换中	BOOL	0-1	0	变量	正在运行状态位
Error	错误	BOOL	0-1	0	变量	执行错误状态
Status	状态	DINT	0x0000-0xFFFF	0x0000	变量	错误/状态代码字

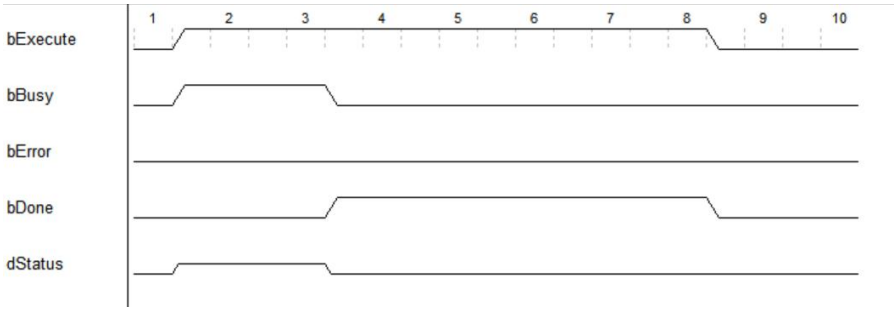
3）功能说明

1. Execute ：上升沿触发，和目标服务端建立 tcp 连接。服务端地址通过 TcpConnect 中的 RemoteAddress 和 RemotePort 描述，LocalPort 描述自身客户端端口，如果是 0，表示系统自动分配，大于 0 小于 65536 即指定特定端口。
2. RemoteAddress 指定通信服务器对象的 IP 地址，标准 xxx.xxx.xxx.xxx 格式。格式错误会输出错误代码。不可设置为 000.000.000.000.
3. RemotePort：指定通信服务端对象的端口号。如果出现非正数，或者大于 65535 的数值，则输出错误代码。
4. 如果使用同样的 RemotePort 和 RemoteAddress 以及 LocalPort 重复创建 Connect 连接，系统会返回错误，提示 socket 重复创建。
5. LocalPort：设置本地端口号。0 表示系统自动分配。1-65535 表示指定特定端口号，大于 65535 或小于 0 输出错误代码。
6. TimeOut：建立连接超时时间，单位毫秒。>0 :如果在设定时间内系统 socket 未能返回结果（包括明确的故障信息，或者成功建立连接），则指令返回超时错误。
7. HWPort：网口编号：1-3. 1： 1 号网口，2:2 号网口，3:3 号网口。>3 或者 <1 非法数据，输出错误报警信息。
8. Error：提示当前执行的功能是否出现故障。故障的复位来自于触发功能的指令（Execute）的复位。
9. Status；当出现 Error 信号时，对应的故障代码输出。输出内容和 Error 同步，当 Error 复位时，Status=0 。
10. 由于系统端口无法复用，所以当客户端使用同一个本地端口发出第二个连接指令时，系统会返回错误，提示端口已经被占用。

时序图

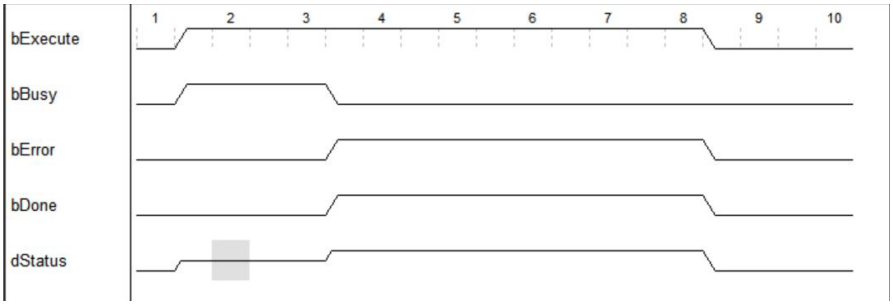
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1.正确执行返回的时序如下：



2.Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位，对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误

//正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
//协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误

0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
//参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
//用户操作错误	

0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

ST 示例

```
TcpConnect_1(Execute:=bool_in,              //触发执行控制位，上升沿触发
             HWPort:=int_in,                //物理网口
             LocalPort:=dint_in,            //本地端口
             RemoteAddress:=string_in,//目标 IP 地址
             RemotePort:=dint_in,           //目标端口
             Timeout:=dint_in,              //指令执行超时设定参数(毫秒)
             Handle=>dint_out,              //输出句柄
```

Done=>bool_out,　　//执行成功状态位

Busy=>bool_out,　　//正在运行状态位

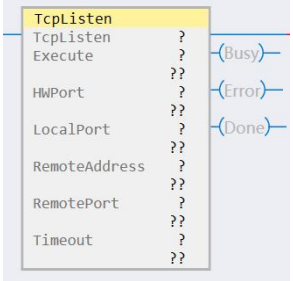
Error=>bool_out,　　//执行错误状态

Status=>dint_out);　　//错误/状态代码字

TCP 服务器监听 TcpListen 指令

启用 TCP 服务器，并监听特定端口等待连接请求。

1）指令格式

指令	名称	梯形图表现	ST 表现
TcpListen	TCP 服务器监听指令		TcpListen_1(Execute:=, HWPort:=, LocalPort:=, RemoteAddress:=, RemotePort:=, Timeout:=, Handle=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明

Execute	执行	BOOL	0-1	0	变量	触发执行控制位，上升沿触发
HWPort	通信口	INT	1-3	1	变量或立即数	1/2/3 代表物理网口
LocalPort	本地端口	DINT	0-65536	0	变量或立即数	本地端口
RemoteAddress	远程地址	STRING	合法 IP	''	变量或立即数	目标 IP 地址
RemotePort	远程端口	DINT	0-65536	0	变量或立即数	目标端口
Timeout	超时	DINT	0-60000	5000	变量或立即数	指令执行超时设定参数(毫秒)

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Handle	句柄	DINT	0-65536	0	变量	输出句柄
Done	完成	BOOL	0-1	0	变量	执行成功状态位
Busy	数据交换中	BOOL	0-1	0	变量	正在运行状态位
Error	错误	BOOL	0-1	0	变量	执行错误状态
Status	状态	DINT	0x0000-0xFFFF	0	变量	错误/状态代码字

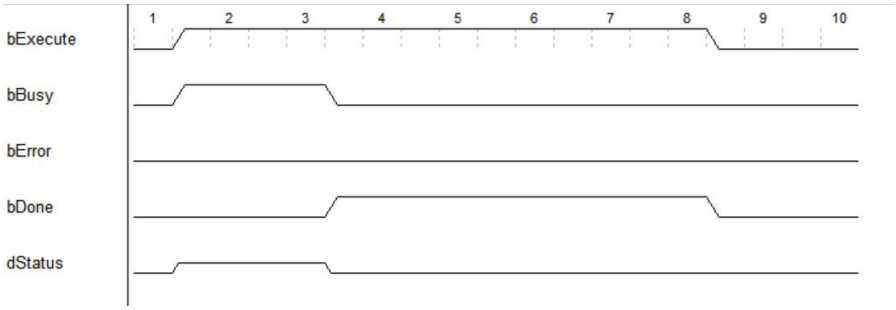
3）功能说明

1. Execute：上升沿触发接收数据。指令实际功能为获取 socket 接收缓存区中的数据。
2. Busy：指令正在执行状态。
3. Error:故障信号输出。
4. Done：指令成功执行。

时序图

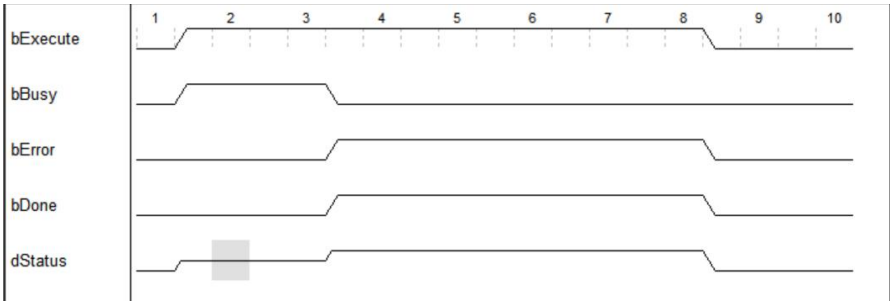
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1.正确执行返回的时序如下：



2.Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位，对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
//正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
//协议错误	

0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
//参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效

0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
//用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

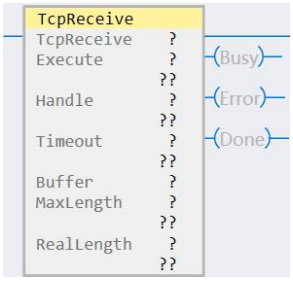
ST 示例

```
TcpListen_1(Execute:=bool_in,              //触发执行控制位，上升沿触发
            HWPort:=int_in,                //代表物理网口
            LocalPort:=dint_in,            //本地端口
            RemoteAddress:=string_in,      //目标 IP 地址
            RemotePort:=dint_in,           //目标端口
            Timeout:=dint_in,              //指令执行超时设定参数(毫秒)
            Handle=>dint_out,              //输出句柄
            Done=>bool_out,                    //执行成功状态位
            Busy=>bool_out,                    //正在运行状态位
            Error=>bool_out,               //执行错误状态
            Status=>dint_out);             //错误/状态代码字
```

TCP 接收数据指令 TcpReceive 指令

在 TCP 连接通道上，接收数据报文。

1）指令格式

指令	名称	梯形图表现	ST 表现
TcpReceive	TCP 接收指令		TcpReceive_1(Buffer:=, Execute:=, Handle:=, Timeout:=, MaxLength:=, RealLength=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明

Execute	执行	BOOL	0-1	0	变量	触发执行控制位，上升沿触发
Handle	句柄	DINT	0-65536	0	变量或立即数	输入句柄
Timeout	超时	DINT	0-60000	5000	变量或立即数	指令执行超时设定参数(毫秒)
MaxLength	最大长度	DINT	0-1024	0	变量或立即数	报文数据长度设定

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Buffer	缓冲区	指针	-	-	变量	数据接受区
RealLength	实际长度	DINT	0-1024	0	变量	报文数据长度实际
Done	完成	BOOL	0-1	0	变量	执行成功状态位
Busy	数据交换中	BOOL	0-1	0	变量	正在运行状态位
Error	错误	BOOL	0-1	0	变量	执行错误状态
Status	状态	DINT	0x0000-0xFFFF	0	变量	错误/状态代码字

3）功能说明

1. Execute：上升沿触发发送数据。
2. Buffer：接收数据字符串。
3. MaxLength：接收数据最大长度，单位 byte。如果小于 1 或者大于 1024，输出错误代码。

4. RealLength:接收数据的实际长度。

5. Timeout：指令返回超时时间，单位毫秒 。

6. Busy：指令正在执行状态。

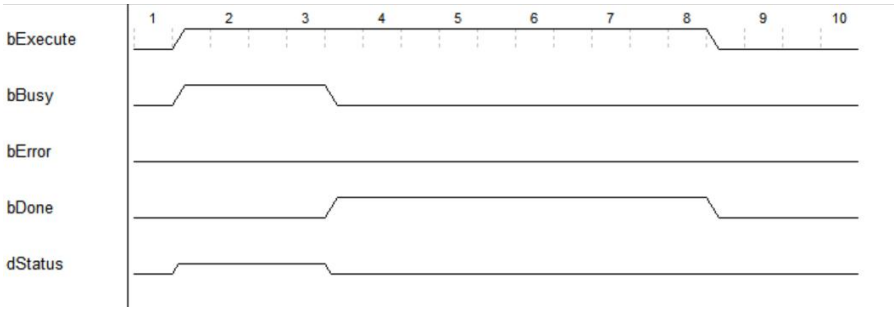
7. Error:故障信号输出。

8. Done：指令成功执行。

时序图

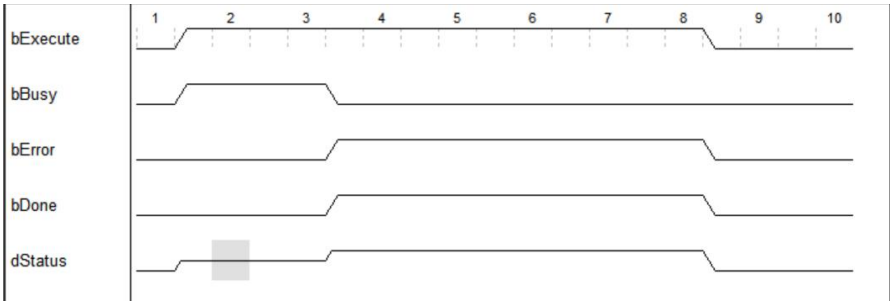
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1.正确执行返回的时序如下：



2.Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位，对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
//正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
//协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复

0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
//参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法

0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
//用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

ST 示例

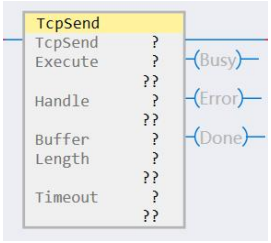
```
TcpReceive_1(Buffer:=any_in,        //数据接受区
             Execute:=bool_in,      //触发执行控制位，上升沿触发
```

```
Handle:=dint_in,          //输入句柄
Timeout:=dint_in,         //指令执行超时设定参数(毫秒)
MaxLength:=dint_in,       //报文数据长度设定
RealLength=>dint_out,     //报文数据长度实际
Done=>bool_out,           //执行成功状态位
Busy=>bool_out,           //正在运行状态位
Error=>bool_out,          //执行错误状态
Status=>dint_out);        //错误/状态代码字
```

TCP 发送指令 TcpSend 指令

在 TCP 连接通道上，发送数据报文。

1）指令格式

指令	名称	梯形图表现	ST 表现
TcpSend	TCP 发送指令		TcpSend_1(Buffer:=, Execute:=, Handle:=, Timeout:=, Length:=, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0-1	0	变量	触发执行控制位，上升沿触发
Handle	句柄	DINT	0-65536	0	变量或	输入句柄

					立即数	
Timeout	超时	DINT	0-60000	5000	变量或 立即数	指令执行超时设定参数 (毫秒)
Length	长度	DINT	0-1024	0	变量或 立即数	报文数据长度
Buffer	缓冲区	指针	-	-	变量	数据发送区

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Done	完成	BOOL	0-1	0	变量	执行成功状态位
Busy	数据交换中	BOOL	0-1	0	变量	正在运行状态位
Error	错误	BOOL	0-1	0	变量	执行错误状态
Status	状态	DINT	0x0000-0xFFFF	0	变量	错误/状态代码字

3）功能说明

1. Execute：上升沿触发接收数据。指令为异步非阻塞执行方式。指令实际功能为获取 socket 发送缓冲区中的数据，如果发送缓存区没有任何数据，则返回空字符串，长度为 0。
2. Handle：建立连接后的 Socket 句柄。
3. Buffer：发送的实际数据字符串。
4. Length：发送的数据长度，单位 byte。如果小于 1 或者大于 1024，输出错误代码。
5. Timeout：指令返回超时时间，单位毫秒 。指令在指定时间内 socket 发送缓冲

区数据信息，或者返回特定错误信息。

6. Busy：指令正在执行状态。

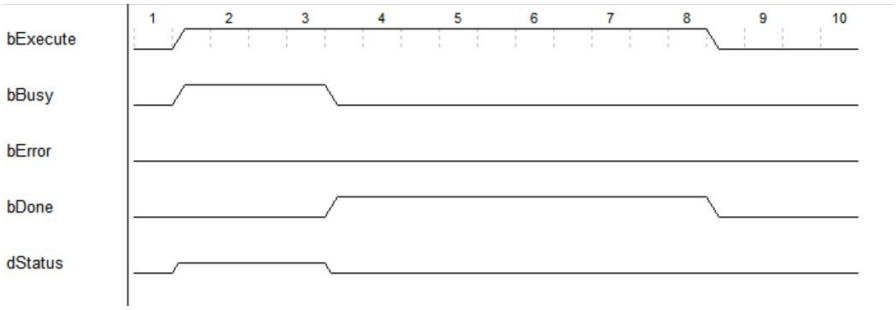
7. Error:故障信号输出。

8. Done：指令成功执行。

时序图

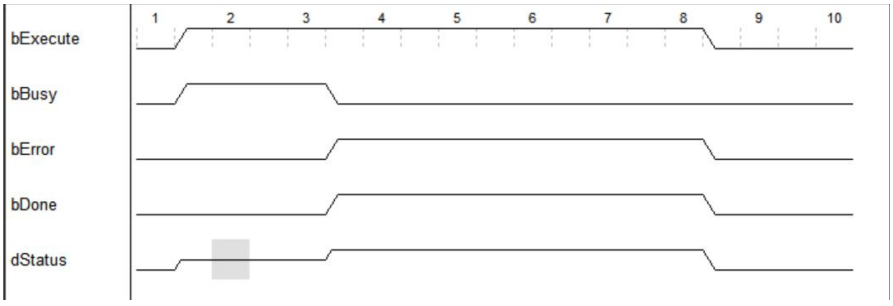
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1.正确执行返回的时序如下：



2.Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位，对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
//正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
//协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误

0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
//参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求

0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
//用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

ST 示例

```
TcpSend_1(Buffer:=any_in，//数据发送区
          Execute:=bool_in, //触发执行控制位，上升沿触发
          Handle:=dint_in，//输入句柄
          Timeout:=dint_in,//指令执行超时设定参数(毫秒）
```

Length:=dint_in，//报文数据长度

Done=>bool_out，//执行成功状态位

Busy=>bool_out，//正在运行状态位

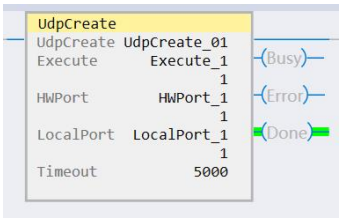
Error=>bool_out，//执行错误状态

Status=>dint_out);//错误/状态代码字

UDP 创建指令 UdpCreate 指令

获取系统资源，并创建 UDP 通讯。

1）指令格式

指令	名称	梯形图表现	ST 表现
UdpCreate	UDP 创建指令		UdpCreate_1(Execute:=, HWPort:=, LocalPort:=, Timeout:=, Handle=>, Done=>, Busy=>, Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
UdpCreate	指令变量	UdpCreate	X	X	变量	指令的背景数据变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0/1	0	变量或者立即数	触发执行控制位，上升沿触发
HWPort	网口号	INT	0/1/2/3	0	变量或者立即数	1/2/3 代表物理网口
LocalPort	本地端口号	DINT	0-65536	0	变量或者立即数	本地端口
Timeout	超时	DINT	0-60000	5000	变量或者立即数	指令执行超时设定参数(毫秒)

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Handle	句柄	DINT	0-65536	0	变量或者立即数	输出句柄
Done	进行中	BOOL	0/1	0	变量或者立即数	执行成功状态位
Busy	完成	BOOL	0/1	0	变量或者立即数	正在运行状态位

Error	错误	BOOL	0/1	0	变量或者立即数	执行错误状态
Status	错误码	DINT	0x0000-0xFFFF	0	变量或者立即数	错误/状态代码字

3）功能说明

1、Execute：上升沿触发创建 UDP 终端。

2、Connection.LocalPort：创建 UDP 终端的本地端口号，不能为 0；

3、SocketComm.Handle：Socket Handle 元素。

4、Busy：指令正在执行状态。

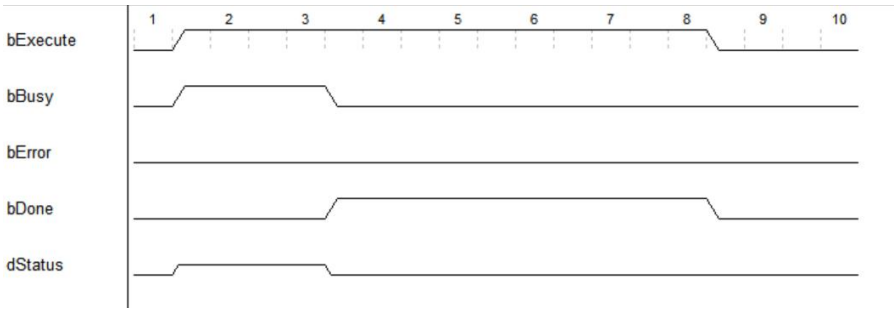
5、Error：故障信号输出。

6、Done：指令成功执行。

时序图

下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

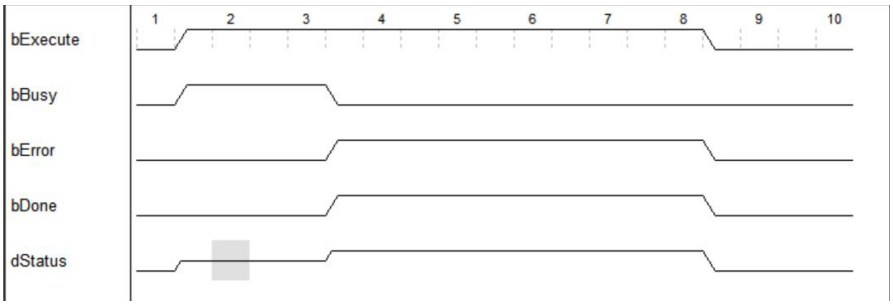
1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号

才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

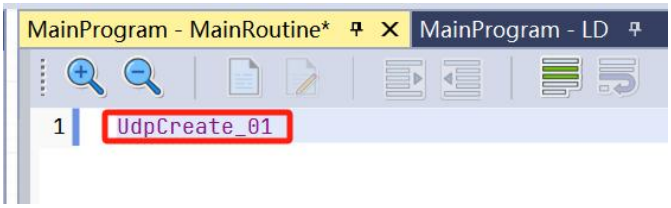
LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

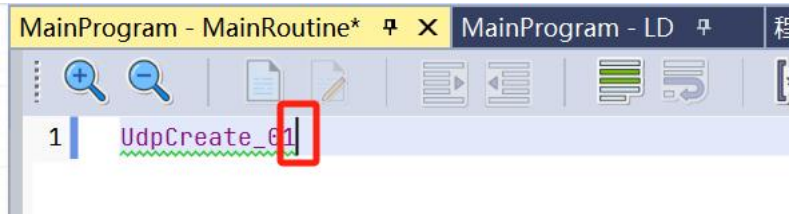
1、 变量表区建立 UdpCreate 指令对应的指令实例变量：

名称	数据类型
▷ UdpCreate_01	UdpCreate

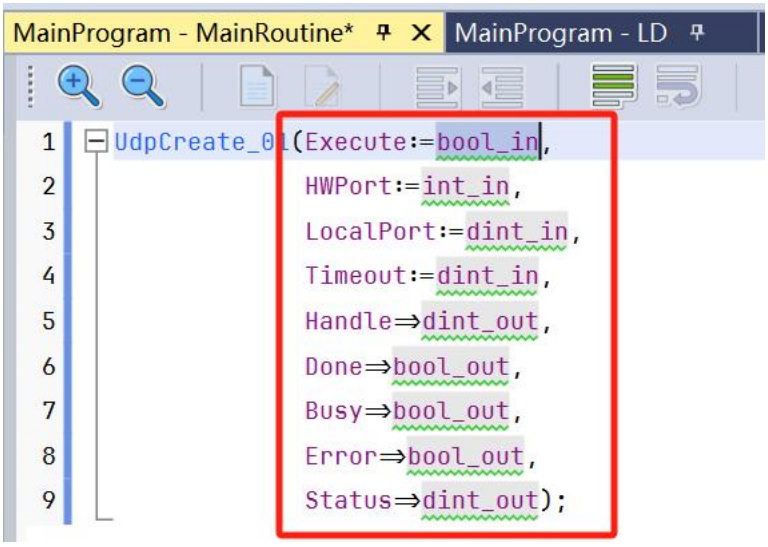
2、 ST 编辑区输入对应 UdpCreate 实例变量名：



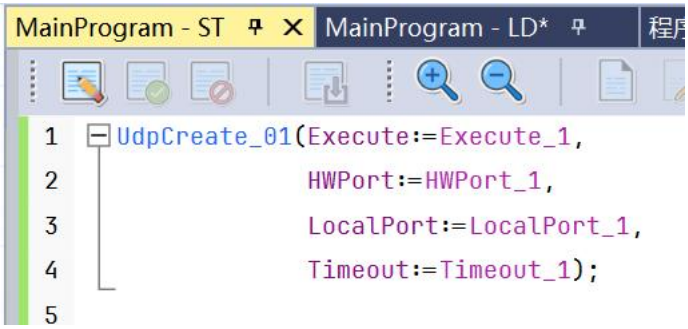
3、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令报错）：



按下 tab 键系统会自动补全引脚内容：



用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：



5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
SocketClose_01(Execute:=bool_in,
               Handle:=dint_in,
               Timeout:=dint_in,
               Done⇒bool_out,
               Busy⇒bool_out,
               Error⇒bool_out,
               Status⇒dint_out);
```

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误

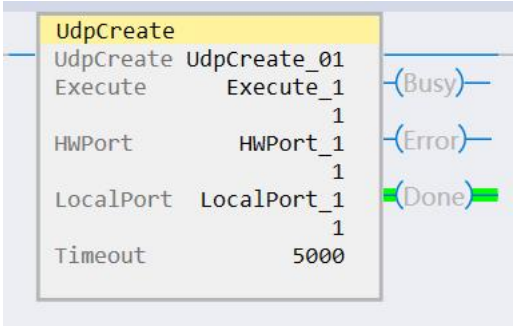
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址

0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效

0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：



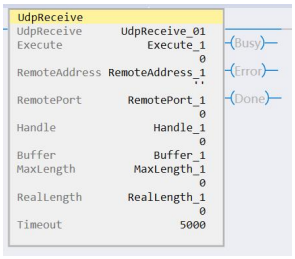
ST 示例：

```
UdpCreate_1(Execute:=bool_in, //触发执行控制位，上升沿触发
            HWPort:=int_in,     //物理网口
            LocalPort:=dint_in, //本地端口
            Timeout:=dint_in,   //指令执行超时设定参数(毫秒)
            Handle=>dint_out,   //输出句柄
            Done=>bool_out,         //执行成功状态位
            Busy=>bool_out,         //正在运行状态位
            Error=>bool_out,    //执行错误状态
            Status=>dint_out);  //错误/状态代码字
```

UPD 接收指令 UdpReceive 指令

在指定端口上接收 UDP 报文。

1）指令格式

指令	名称	梯形图表现	ST 表现
UdpReceive	UPD 接收指令		UdpReceive_1(Buffer:=, Execute:=, Handle:=, Timeout:=, MaxLength:=, RemoteAddress=>, RemotePort=>, RealLength=>, Done=>, Busy=>,

			Error=>, Status=>);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
UdpReceive	指令变量	UdpReceive	X	X	变量	指令的背景数据变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	X	X	变量	触发执行控制位，上升沿触发
Handle	句柄	DINT	0-65536	0	变量或者立即数	输入句柄
Timeout	超时	DINT	0-60000	5000	变量或者立即数	指令执行超时设定参数(毫秒)
MaxLength	最大长度	DINT	0-1024	0	变量或者立即数	报文数据长度设定

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明

RemoteAddress	远程地址	STRING	对应类型范围	‘ ’	变量或者立即数	目标 IP 地址
RemotePort	远程端口	DINT	0-65536	0	变量或者立即数	目标端口
Buffer	缓冲区	指针	X	X	变量或者立即数	数据接受区
RealLength	实际长度	DINT	0-1024	0	变量	报文数据长度实际
Done	完成	BOOL	0/1	0	变量或者立即数	执行成功状态位
Busy	进行中	BOOL	0/1	0	变量或者立即数	正在运行状态位
Error	错误	BOOL	0/1	0	变量	执行错误状态
Status	错误码	DINT	0x0000-0xFFFF	0	变量或者立即数	错误/状态代码字

3）功能说明

1、 Execute：上升沿触发接收数据。接收数据属于异步非阻塞操作，不会等待数据到达，如果 socket 接收缓存区没有数据，则返回空字符串，长度为 0 。

2、 Connection：用来存储对方的 IP 地址和端口；

3、 Handle：Udp 通信用来接收数据的 socket ID。

4、 RecvData：接收数据保存地址。

5、 MaxLength：期望接收数据最大字节数。如果小于 1 或者大于 1024，输出错误代码。

6、 RecvLength ：接受到的实际数据长度。没有接收到数据置 0 。

7、 Timeout：指令返回超时时间，单位毫秒。

8、 Busy：指令正在执行状态。

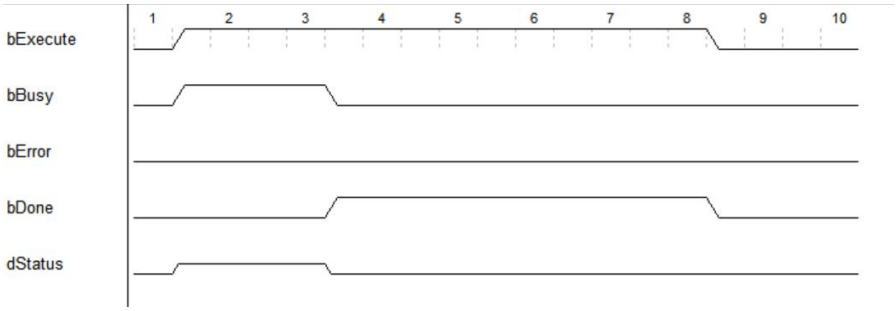
9、 Error：故障信号输出。

10、Done：指令成功执行。

时序图

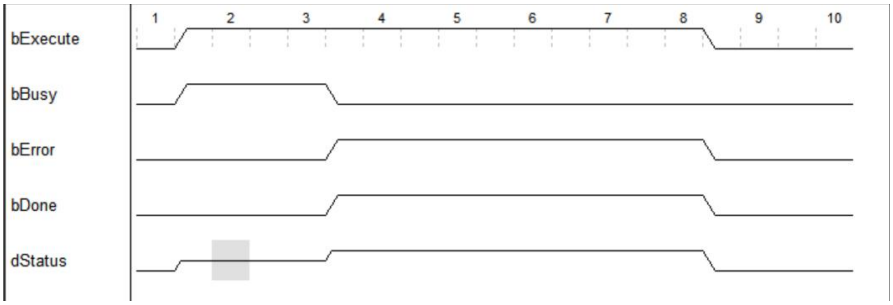
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1、正确执行返回的时序如下：



2、Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位,对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

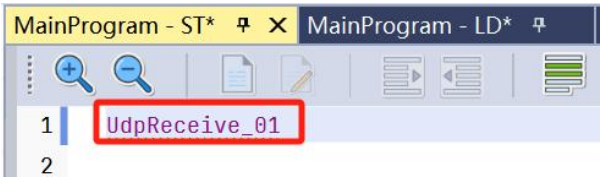
LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤示例如下：

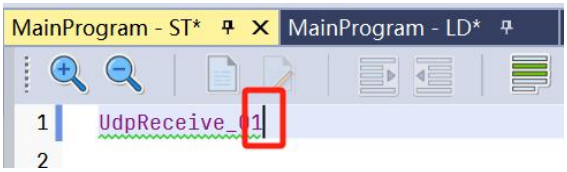
1、 变量表区建立 UdpReceive 指令对应的指令实例变量：

名称	数据类型
▷ UdpReceive_01	UdpReceive

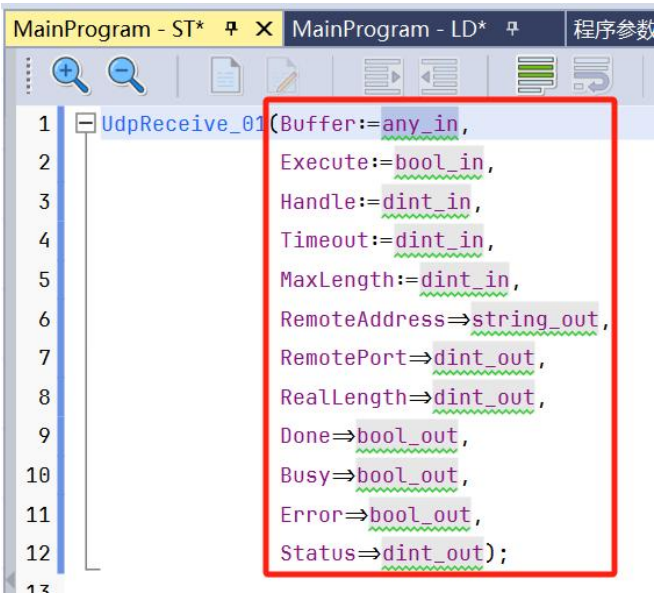
2、 ST 编辑区输入对应 UdpReceive 实例变量名：



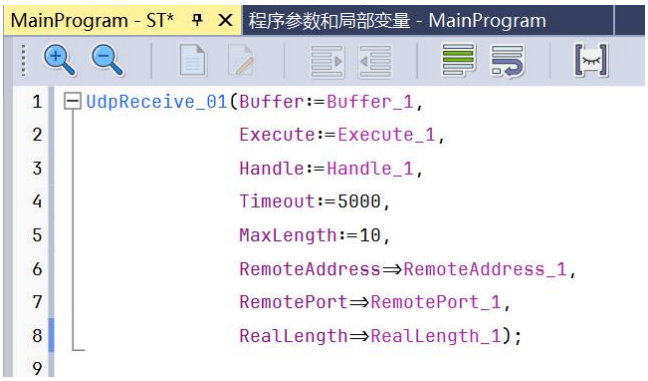
3、 光标放在变量名末尾，按 TAB 键，编辑器会自动补全所有指令引脚以及名称，用户根据需要进行删除部分不必要引脚（注意 InOut 类型指令引脚必须保留否则指令报错）：



按下 tab 键系统会自动补全引脚内容：



用户可以根据需要删除不必要的引脚，简化使用，比如如下形式：



5）错误说明

对于指令接口引脚 Status，包含了所有错误状态内容，详细如下：

```
UdpReceive_01(Buffer:=Buffer_1,
              Execute:=Execute_1,
              Handle:=Handle_1,
              Timeout:=5000,
              MaxLength:=10,
              RemoteAddress⇒RemoteAddress_1,
              RemotePort⇒RemotePort_1,
              RealLength⇒RealLength_1,
              Done⇒Done_1,
              Busy⇒Busy_1,
              Error⇒Error_1,
              Status⇒Status_1);
```

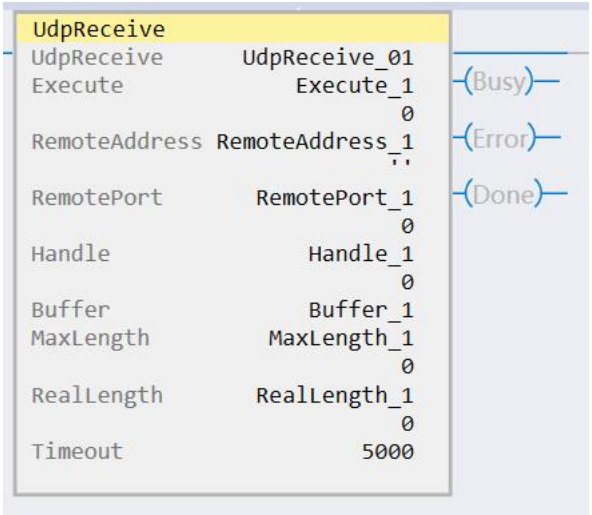
Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
协议错误	
0x5000	协议错误
0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误

0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效
0x6006	MODBUS 参数 PollingTime 不合法

0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

梯形图：



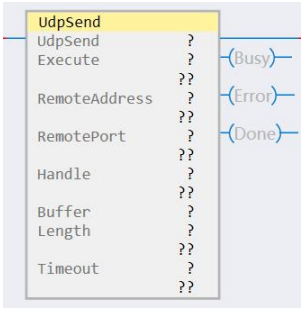
ST 示例：

```
UdpReceive_1(Buffer:=any_in,              //数据接收区
             Execute:=bool_in,            //触发执行控制位，上升沿触发
             Handle:=dint_in,             //输入句柄
             Timeout:=dint_in,            //指令执行超时设定参数(毫秒)
             MaxLength:=dint_in,          //报文数据长度设定
             RemoteAddress=>string_out,   //目标 IP 地址
             RemotePort=>dint_out,        //目标端口
             RealLength=>dint_out,        //报文数据长度实际
             Done=>bool_out,              //执行成功状态位
             Busy=>bool_out,              //正在运行状态位
             Error=>bool_out,             //执行错误状态
             Status=>dint_out);           //错误/状态代码字
```

UDP 发送指令 UdpSend 指令

在指定端口上发送 UDP 报文。

1）指令格式

指令	名称	梯形图表现	ST 表现
UdpSend	UDP 发送指令		UdpSend_1(Buffer:=, Execute:=, Handle:=, RemoteAddress:=, RemotePort:=, Timeout:=, Length:=, Done=>, Busy=>, Error=>, Status=>)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0-1	0	变量	触发执行控制位，上升沿触发
Handle	句柄	DINT	0-65536	0	变量或	输入句柄

					立即数	
RemoteAddress	远程地址	STRING	合法 IP	''	变量或 立即数	目标 IP 地址
RemotePort	远程端口	DINT	0-65536	0	变量或 立即数	目标端口
Timeout	超时	DINT	0-60000	5000	变量或 立即数	指令执行超时设定参数(毫秒)
Length	长度	DINT	0-1024	0	变量或 立即数	报文数据长度
Buffer	缓冲区	指针	-	-	变量	数据发送区

输出变量(Output)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Done	完成	BOOL	0-1	0	变量	执行成功状态位
Busy	数据交换中	BOOL	0-1	0	变量	正在运行状态位
Error	错误	BOOL	0-1	0	变量	执行错误状态
Status	状态	DINT	0x0000-0xFFFF	0	变量	错误/状态代码字

3）功能说明

1. Execute：上升沿触发发送数据。
2. Handle: 传递来自 UDP 创建的连接句柄，从此创建的 Socket 上发出数据。
3. Buffer：发送实际数据字符串。
4. Length：发送的数据长度，单位 byte。如果小于 0 或者大于 1024，输出错误代

码。

5. Timeout：指令返回超时时间，单位毫秒 。

6. Busy：指令正在执行状态。

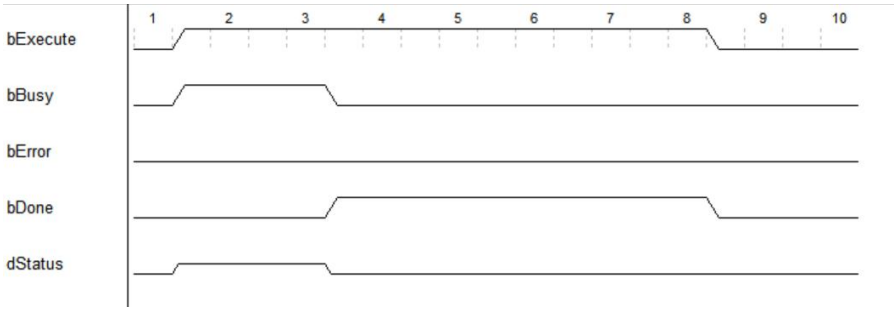
7. Error:故障信号输出。

8. Done：指令成功执行。

时序图

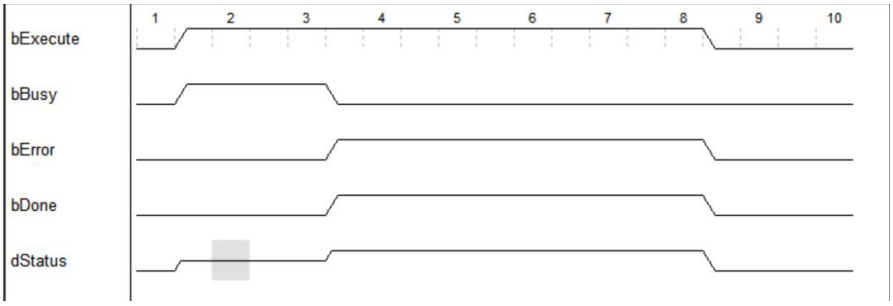
下图演示了正常的通讯触发以及非正常触发的相关状态位波形图：

1.正确执行返回的时序如下：



2.Error 信号复位 ：如果出现故障信号，需要对应的触发信号复位，Error 信号才会在一个扫描周期后自动复位，对应的 Status 信息也会在一个扫描周期后自动复位。

时序图如下：



4）注意事项

LD 梯形图上使用与调用方式与其它指令完全一致。

ST 结构文本上使用了新的调用和输入方式，具体操作步骤请参考新调用方式相关文档

5）错误说明

Status 状态/故障码	说明
0x0000	指令已执行，且无任何错误
//正常状态信息	
0x4000	正常状态
0x4001	连接已建立
0x4002	连接已终止
0x4003	未激活任何作业且未建立任何连接
0x4004	已触发连接建立操作
0x4005	正在建立连接
0x4006	正在终止连接
0x4007	连接已建立但未激活任何作业
0x4008	正在发送数据
0x4009	正在接收数据
0x400A	超时
0x400B	连接需要重置(内部自动或者用户手动)
0x400C	端口无 ip
//协议错误	
0x5000	协议错误

0x5001	MODBUS 服务端没有在指定时间内回复
0x5002	MODBUS 收到的 modbus 帧错误
0x5003	MODBUS 不支持的功能码
0x5004	MODBUS 帧头中填入的数据长度非法
0x5005	MODBUS 客户端访问了服务端的非法地址
0x5006	MODBUS 报文中数据的值错误
0x5007	MODBUS 不支持的诊断码 or 子功能码与发送的不一致
0x5008	MODBUS 接受到的功能码与最初发送的代码不一致
0x5009	MODBUS 接收到的 Modbus TCP 帧协议 ID 不为 “0”
0x500A	MODBUS 客户端请求与服务端返回数据长度不一致
0x500B	MODBUS 客户端收到报文 unit id 异常
0x500C	MODBUS 收到包含异常码的回复
0x500D	SOCKET 指令硬件端口绑定失败
0x500E	SOCKET 指令本地端口绑定失败
0x500F	SOCKET 指令 UDP 绑定对端失败
//参数错误	
0x6000	参数无效
0x6001	MODBUS 数据区定义无效
0x6002	MODBUS 参数 MB_RWMode 值无效
0x6003	MODBUS 参数 MB_StartAddress 无效
0x6004	MODBUS 参数 MB_DataLength 无效
0x6005	MODBUS 参数本地地址无效

0x6006	MODBUS 参数 PollingTime 不合法
0x6007	MODBUS 本端口正在处理其他 modbus 请求
0x6008	MODBUS 后台 tcpserver 启动失败
0x6009	SOCKET 句柄无效
0x600A	SOCKET 指令目标 IP 地址不合法
//用户操作错误	
0x7000	用户操作错误
0x7001	MODBUS 用户在线修改参数
0x7002	MODBUS 后台实例不存在
0x7003	MODBUS 后台实例超过上限
0x7004	MODBUS 用户指定网络地址参数不合法
0x7005	MODBUS 用户指定数据模式不正确
0x7006	MODBUS 用户提供的参数在本地数据区越界了
0x7007	MODBUS 用户在线修改后收到了在线修改前的包
0x7008	MODBUS 用户提供的 start address 的偏移值非法
0x7009	SOCKET 句柄未连接
0x700A	SOCKET 句柄未处于 listen 状态
0x700B	SOCKET 句柄对应的 fd 无效

6）示例

ST 示例

```
UdpSend_1(Buffer:=any_in,          //数据发送区
```

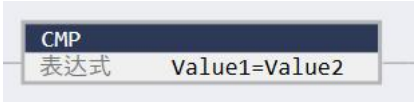
```
Execute:=bool_in,          //触发执行控制位，上升沿触发
Handle:=dint_in,           //输入句柄
RemoteAddress:=string_in,  //目标 IP 地址
RemotePort:=dint_in,       //目标端口
Timeout:=dint_in,          //指令执行超时设定参数(毫秒)
Length:=dint_in,           //报文数据长度
Done=>bool_out,            //执行成功状态位
Busy=>bool_out,            //正在运行状态位
Error=>bool_out,           //执行错误状态
Status=>dint_out);         //错误/状态代码字
```

五、比较指令

比较指令 CMP 指令

CMP 指令对表达式进行运算后进行比较，输出值为布尔值。

1）指令格式

指令	名称	梯形图表现	ST 表现
CMP	比较		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Expression	表达式	INT DINT LINT SINT REAL LREAL STRING	变量 立即数	表达式由变量和/或由运算符分开的立即值组成

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	表达式为 false 时梯级输出条件设置为 false 表达式为 true 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

如果输入不带比较运算符的表达式，如 value1+value_2 或 value_1 则该指令将如下所述计算表达式:

如果表达式为:	则梯级输出条件设置为:
非零	true
零	false

CMP 表达式

下面几节内容提供了有关有效的运算符、格式、运算顺序的信息，这些说明对于 CMP 指令适用。

有效运算符

运算符:	说明:	最优:
＋	加	DINT、REAL
－	减/求负	DINT、REAL
*	乘	DINT、REAL
/	除	DINT、REAL
＝	等于	DINT、REAL
<	小于	DINT、REAL

⩽	小于等于	DINT、REAL
>	大于	DINT、REAL
⩾	大于等于	DINT、REAL
<>	不等于	DINT、REAL
**	指数	DINT、REAL
ABS	绝对值	DINT、REAL
AND	按位与	DINT
COS	余弦	REAL
DEG	弧度转角度	DINT、REAL
LN	自然对数	REAL
LOG	以 10 为底的对数	REAL
MOD	模数除法	DINT、REAL
NOT	按位求补	DINT
OR	按位或	DINT
RAD	角度转弧度	DINT、REAL
SIN	正弦	DINT、REAL
SQRT	平方根	DINT、REAL
TAN	正切	REAL
XOR	按位异或	DINT

格式化表达式

对于表达式中使用的每个运算符，必须提供一个或两个操作数(变量或立即值)。请根据下表设置表达式中运算符和操作数的格式:

S 运算符涉及的操作数的个数：	使用格式：	示例：
1	运算符(操作数)	ABS(tag_a)
2	操作数_a 运算符 操作数_b	tag b+5 tag_c AND tag_d (tag_e **22) MOD(tag_f/tag_g)

确定运算次序

指令按预定的次序执行写入表达式中的运算(不一定是写入次序)可以使用括号将运算条目分组，强制指令在执行其他运算之前先执行括号中的运算，从而更改运算次序。

次序相等的运算从左向右执行。

次序：	运算：
1	（）
2	ABS、LN、LOG、RAD、SIN、SQRT、TAN
3	**
4	-（求负）、NOT
5	*、/、MOD
6	<、≤、>、≥、=
7	-（减）、+
8	AND
9	XOR
10	OR

在表达式中使用字符串

可以使用梯形图或结构化文本表达式比较字符串数据类型。要在表达式中使用字符串，请遵循以下准则:

表达式可以比较两个字符串标记

不能在表达式中直接输入 ASCII 字符

仅允许使用以下运算符

=	等于
<	小于
≤	小于等于
>	大于
≥	大于等于
<>	不等于

如果两个字符串的字符相匹配，则这两个字符串相等 ASCII 字符区分大小写。

字符的十六进制值决定一个字符串是否小于或大于另一个字符串。

如果两个字符串按电话目录的方式排序，则字符串的顺序决定字符串的大小。

4）注意事项

使用运算符、变量和立即数定义 CMP 表达式。使用括号（）定义更复杂的表达式的各部分。

CMP 指令执行起来比其他比较指令要稍微慢一些，使用的内存也比其他比较指令多。CMP 指令的优点在于它允许您在一个指令中输入复杂的表达式。

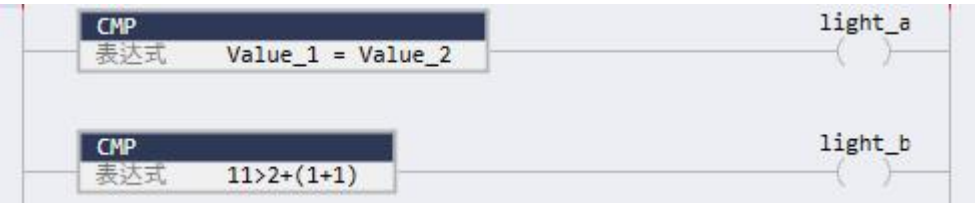
只有当表达式中包含影响算术状态标志的运算符（如： +、-、*、/）时 CMP 指令才会影响算术状态标志。

5）错误说明

6）示例

梯形图：

如果 CMP 指令发现表达式为 true,则梯级输出条件设置为 true。



ST 示例：

结构化文本没有 CMP 指令，但您可以使用 IF...THEN 结构实现相同的结果。

```
IF BOOL_Expression THEN
  <statement>;
END_IF;
```

等于指令 EQ 指令

EQ 指令判断源 A 是否等于源 B。

1）指令格式

指令	名称	梯形图表现	ST 表现
EQ	等于		Source_A=Source_B

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源 A	INT DINT LINT SINT REAL LREAL STRING WSTRING	变量 立即数	针对 Source B 进行测试的值。
SourceB	源 B	INT DINT	变量 立即数	针对 Source A 进行测试的值。

		LINT SINT REAL LREAL STRING WSTRING		

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	SourceA ≠ SourceB 时梯级输出条件设置为 false SourceA = SourceB 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

4）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

字符串数据类型为:

默认 STRING 数据类型

您创建的任何新的字符串数据类型要测试字符串的字符，请为 Source A 和 Source B 都输入一个字符串标记。

使用 EQ 指令比较两个数或两个 ASCII 字符串。比较字符串时

如果字符串的字符都一致，则字符串相等。

ASCII 字符是区分大小写的。

5）错误说明

6）示例

梯形图：

如果 Value1 等于 Value2 则 BOOL_OTE 为 true。如果 Value1 不等于 Value2 则 BOOL_OTE 为 false。

ST 示例：

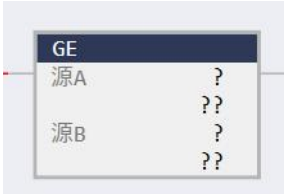
在表达式中将等号“=”用作运算符。

```
IF Value1=Value2 THEN
  <statement>;
END_IF;
```

大于等于指令 GE 指令

GE 指令判断源 A 是否大于或等于源 B。

1）指令格式

指令	名称	梯形图表现	ST 表现
GE	大于等于		在表达式中使用相邻的大于号和等于号“>=”作为运算符。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源 A	INT DINT LINT SINT REAL LREAL STRING WSTRING	变量 立即数	针对 Source B 进行测试的值。
SourceB	源 B	INT DINT	变量 立即数	针对 Source A 进行测试的值。

		LINT SINT REAL LREAL STRING WSTRING		

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	SourceA 不≥SourceB 时梯级输出条件设置为 false SourceA≥SourceB 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

4）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

字符串数据类型为：

默认 STRING 数据类型

您创建的任何新的字符串数据类型

要测试字符串的字符，请为 Source A 和 Source B 都输入一个字符串标记。

比较字符串时

字符的十六进制值确定两个字符串之间的大小关系。

当两个字符串按在电话号码簿中的方式排序时，字符串的顺序决定了哪一个是较大的。

5）错误说明

6）示例

梯形图：

如果 Value1 大于或等于 Value2 则 BOOL_OTE 为 true。如果 Value1 小于 Value2 则 BOOL_OTE 为 false。

ST 示例：

在表达式中同时使用小于号和大于号“>=”作为运算符。

```
IF Value1 >= Value2 THEN
  <statement>;
END_IF;
BOOL_OTE:=(Value1 >= Value2);
```

大于指令 GT 指令

GT 指令判断源 A 是否大于源 B。

1）指令格式

指令	名称	梯形图表现	ST 表现
GT	大于		在表达式中使用大于号“>”作为运算符。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源 A	INT DINT LINT SINT REAL LREAL STRING WSTRING	变量 立即数	针对 Source B 进行测试的值。
SourceB	源 B	INT DINT	变量 立即数	针对 Source A 进行测试的值。

		LINT SINT REAL LREAL STRING WSTRING		

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	SourceA 不>SourceB 时梯级输出条件设置为 false SourceA>SourceB 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

4）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

字符串数据类型为：

默认 STRING 数据类型

您创建的任何新的字符串数据类型

要测试字符串的字符，请为 Source A 和 Source B 都输入一个字符串标记。

比较字符串时

字符的十六进制值确定两个字符串之间的大小关系。

当两个字符串按在电话号码簿中的方式排序时，字符串的顺序决定了哪一个是较大的。

5）错误说明

6）示例

梯形图：

如果 Value1 大于 Value2 则 BOOL_OTE 为 true。如果 Value1 小于或等于 Value2 则 BOOL_OTE 为 false。

ST 示例：

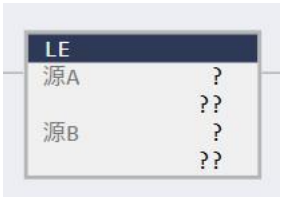
在表达式中同时使用小于号和大于号“>”作为运算符。

```
IF Value1 > Value2 THEN
  <statement>;
END_IF;
BOOL_OTE:=(Value1 > Value2);
```

小于等于指令 LE 指令

LE 指令判断源 A 是否小于或等于源 B。

1）指令格式

指令	名称	梯形图表现	ST 表现
LE	小于等于		在表达式中使用相邻的小于号和 等于号“<=”作为运算符。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源 A	INT DINT LINT SINT REAL LREAL STRING WSTRING	变量 立即数	针对 Source B 进行测试的值。
SourceB	源 B	INT DINT	变量 立即数	针对 Source A 进行测试的值。

		LINT SINT REAL LREAL STRING WSTRING		

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	SourceA 不≤SourceB 时梯级输出条件设置为 false SourceA≤SourceB 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

4）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

字符串数据类型为：

默认 STRING 数据类型

您创建的任何新的字符串数据类型

要测试字符串的字符，请为 Source A 和 Source B 都输入一个字符串标记。

比较字符串时

字符的十六进制值确定两个字符串之间的大小关系。

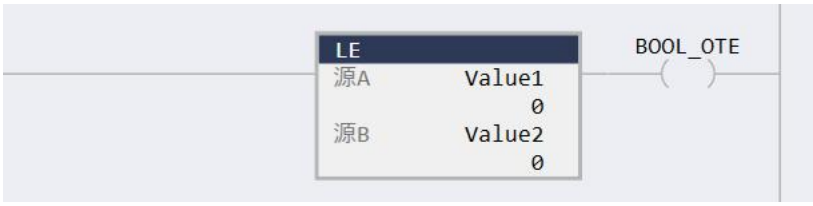
当两个字符串按在电话号码簿中的方式排序时，字符串的顺序决定了哪一个是较大的。

5）错误说明

6）示例

梯形图：

如果 Value1 小于或等于 Value2 则 BOOL_OTE 为 true。如果 Value1 大于 Value2 则 BOOL_OTE 为 false。



ST 示例：

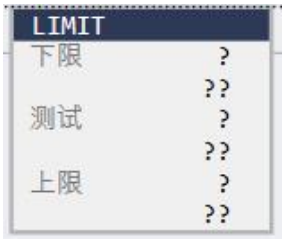
在表达式中同时使用小于号和大于号“≤”作为运算符。

```
IF Value1 ≤ Value2 THEN
  <statement>;
END_IF;
BOOL_OTE:=(Value1≤ Value2);
```

范围判断指令 LIMIT 指令

LIMIT 指令判断测试值是否在限制范围内。

1）指令格式

指令	名称	梯形图表现	ST 表现
LIMIT	范围判断		结构化文本没有 LIMIT 指令，但可通过结构化文本获得相同结果。 IF LowLimit ≤ Test AND Test ≤ HighLimit AND THEN ; <statement>; END_IF;。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Low limit	下限	INT DINT LINT SINT REAL LREAL	立即数 标签	下限值

Test	**测试**	**INT** **DINT** **LINT** **SINT** **REAL** **LREAL**	**立即数** **标签**	**要检验的值。**
High limit	**上限**	**INT** **DINT** **LINT** **SINT** **REAL** **LREAL**	**立即数** **标签**	**上限值**

3）功能说明

情况	**行为**
预扫描	**梯级输出条件设置为 false**
梯级输入条件为 false	**梯级输出条件设置为 false**
梯级输入条件为 true	**评估限制：** **比较结果为真=梯形输出条件设置为真** **比较结果为假=梯形输出条件设置为假**
后期扫描	**梯级输出条件设置为 false**

4）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

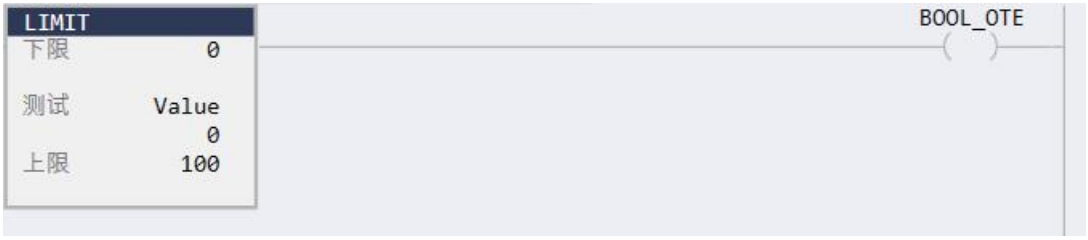
5）错误说明

6）示例

梯形图：

下限≤上限：

如果 0≤Value≤100，则 BOOL_OTE 为 true。如果 Value<0 或 Value>100，则 BOOL_OTE 为 false。



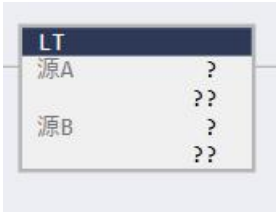
ST 示例：

```
IF    Values ≤100 AND Value≥0 THEN
      BOOL_OTE:=1;
ELSE
      B0OL_OTE:=0;
END_IF;
```

小于指令 LT 指令

LT 指令判断源 A 是否小于源 B。

7）指令格式

指令	名称	梯形图表现	ST 表现
LT	小于		在表达式中使用大于号“<”作为运算符。

8）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源 A	INT DINT LINT SINT REAL LREAL STRING WSTRING	变量 立即数	针对 Source B 进行测试的值。
SourceB	源 B	INT DINT	变量 立即数	针对 Source A 进行测试的值。

		LINT SINT REAL LREAL STRING WSTRING		

9）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	SourceA<SourceB 时梯级输出条件设置为 true SourceA 不<SourceB 时梯级输出条件设置为 false
后期扫描	梯级输出条件设置为 false

10）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

字符串数据类型为：

默认 STRING 数据类型

您创建的任何新的字符串数据类型

要测试字符串的字符，请为 Source A 和 Source B 都输入一个字符串标记。

比较字符串时

字符的十六进制值确定两个字符串之间的大小关系。

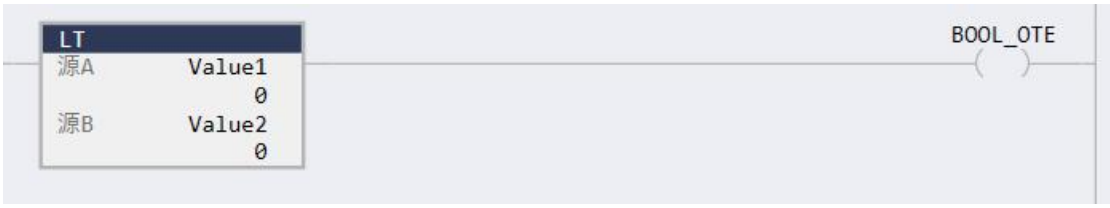
当两个字符串按在电话号码簿中的方式排序时，字符串的顺序决定了哪一个是较大的。

11）错误说明

12）示例

梯形图：

如果 Value1 小于 Value2 则 BOOL_OTE 为 true。如果 Value1 大于或等于 Value2 则 BOOL_OTE 为 false。



ST 示例：

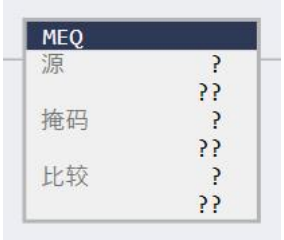
在表达式中同时使用小于号和大于号“<”作为运算符。

```
IF Value1 < Value2 THEN
  <statement>;
END_IF;
BOOL_OTE:=(Value1 < Value2);
```

掩码等于指令 MEQ 指令

MEQ 指令通过掩码处理后判断源值是否等于比较值。

1）指令格式

指令	名称	梯形图表现	ST 表现
MEQ	掩码等于		结构化文本没有 MEQ 指令。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	INT DINT LINT SINT UINT UDINT ULINT USINT	变量 立即数	要对照 Compare 进行测试的值。
Mask	掩码	INT	变量	定义要屏蔽或传递哪些

		DINT LINT SINT UINT UDINT ULINT USINT	立即数	位。
Compare	比较	INT DINT LINT SINT UINT UDINT ULINT USINT		要对照 Source 进行测试的值。

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	掩码源值≠掩码比较值时梯级输出条件设置为 false 掩码源值=掩码比较值时梯级输出条件设置为 true

后期扫描	梯级输出条件设置为 false

4）注意事项

掩码中的“1”表示传递该数据位。掩码中的“0”表示屏蔽该数据位。通常，Source、Mask 和 Compare 值是同一种数据类型

如果您混用整数数据类型，该指令将用 0 填充较小的整数数据类型的高位，以使它们的大小与最大的数据类型相同。

输入立即数掩码值

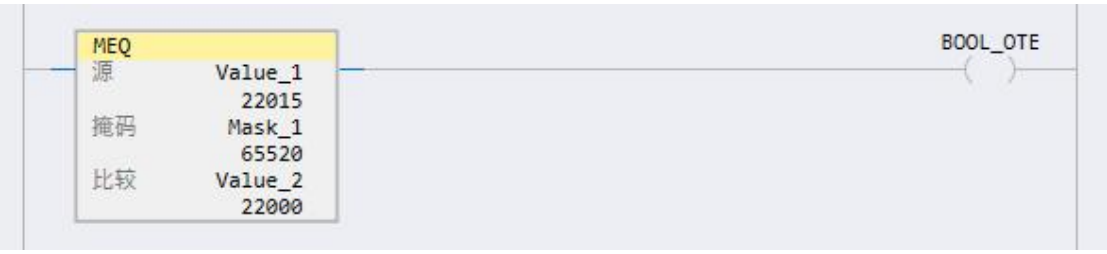
前缀：	说明：
16#	十六进制 例如：16#0F0F
8#	八进制： 例如：8#16
2#	二进制： 例如：2#00110011

5）错误说明

6）示例

梯形图：

如果应用掩码后的 Value_1 等于应用掩码的 Value_2 则 BOOL_OTE 为 true。如果应用掩码后的 Value_1 不等于应用掩码的 Value_2 则 BOOL_OTE 为 false。



ST 示例：

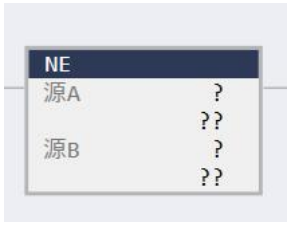
结构化文本没有 MEQ 指令，但您可以使用结构化文本实现相同的结果。

```
IF (Source AND Mask)=(Compare AND Mask) THEN
    <statement>;
END_IF;
```

不等于指令 NE 指令

NE 指令判断源 A 是否不等于源 B。

1）指令格式

指令	名称	梯形图表现	ST 表现
NE	不等于		在表达式中同时使用小于号和大于号”<>“作为运算符。该表达式计算 sourceA 是否不等于 sourceB。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源 A	INT DINT LINT SINT REAL LREAL STRING WSTRING	变量 立即数	针对 Source B 进行测试的值。
SourceB	源 B	INT	变量	针对 Source A 进

		DINT LINT SINT REAL LREAL STRING WSTRING	立即数	行测试的值。

3）功能说明

情况	行为
预扫描	梯级输出条件设置为 false
梯级输入条件为 false	梯级输出条件设置为 false
梯级输入条件为 true	SourceA≠SourceB 时梯级输出条件设置为 false SourceA=SourceB 时梯级输出条件设置为 true
后期扫描	梯级输出条件设置为 false

4）注意事项

如果您输入 SINT 或 INT 标记，该值将通过符号扩展转换为 DINT 值。

字符串数据类型为：

默认 STRING 数据类型

您创建的任何新的字符串数据类型

要测试字符串的字符，请为 Source A 和 Source B 都输入一个字符串标记。

比较字符串时

如果两个字符串的字符不匹配，则这两个字符串不相等。

ASCII 字符串区分大小写。

5）错误说明

6）示例

梯形图：

如果 Value1 等于 Value2 则 BOOL_OTE 为 true。如果 Value1 不等于 Value2 则 BOOL_OTE 为 false。

ST 示例：

在表达式中同时使用小于号和大于号“<>”作为运算符。

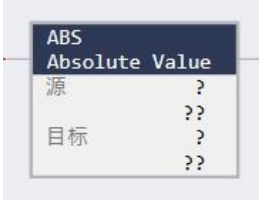
```
IF Value1 <> Value2 THEN
  <statement>;
END_IF;
BOOL_OTE:=(Value1 <> Value2);
```

六、运算指令

绝对值运算 ABS 指令

ABS 指令计算源值的绝对值。

1）指令格式

指令	名称	梯形图表现	ST 表现
ABS	绝对值运算		Dest:=ABS(Source)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT USINT UINT UDINT ULINT	变量 立即数	待求绝对值的值

		REAL LREAL		

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，ABS 指令和运算符用于求 Source 的绝对值。指令将结果存储在 Dest 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。

梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的绝对值。
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图 ：

A 的值为-7，ABS 启用时，会把 A 的绝对值 7，存储到 S 里

ST 示列 ：

S:=ABS(A);

加法运算 ADD 指令

ADD 指令计算源 A 加上源 B 的值。

1）指令格式

指令	名称	梯形图表现	ST 表现
ADD	加法运算		此指令不可用于 ST 结构

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 变量	要与 Source B 相加的值

SourceB	源数据 B	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	立即数 变量	要与 Source A 相加的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest		SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，ADD 指令和运算符“+”用于将 Source A 与 Source B 相加。

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件，Dest = Source A + Source B
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图:

A1 的值为 1，B1 的值为 2，ADD 启用时，会把 A1 的值与 B1 的值相加后得到值 3，存储到 S 里

ST 示列:

结构化文本没有 ADD 指令，但您可以使用“+”实现相同的结果

S:=A1+B1;

计算表达式值 CPT 指令

CPT 指令对表达式进行运算，输出值为数值型。

1）指令格式

指令	名称	梯形图表现	ST 表现
CPT	计算表达式值		此指令不可用于 ST 结构

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Expression	表达式	SINT INT DINT REAL	变量 立即数	需要执行的表达式，由变量或立即数组成，并以运算符分隔

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		REAL		

3）功能说明

使能后，CPT 指令会求表达式的值并将结果放入 Dest 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 指令会求表达式的值并将结果放入 Dest 中。
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图：

A 值为 7，B 值为 1。启用时，CPT 指令将执行运算表达式，并将结果 8 存储到 S 中

ST 示列 ：

结构化文本没有 CPT 指令，但您可以在结构化文本中编入与 CPT 表达式中相同的内容，即可实现相同的结果。

S:=A+B;

除法运算 DIV 指令

DIV 指令计算源 A 除以源 B 的值。

1）指令格式

指令	名称	梯形图表现	结构化文本表现
DIV	除法运算		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	被除数的值

SourceB	源数据 B	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	除数

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，DIV 指令和运算符“/”用于将 Source A 减 Source B

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest = Source A / Source B
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图：

A1 的值为 2，B1 的值为 1，DIV 启用时，会把 A1 的值与 B1 的值相除后得到值 2，存储到 S 里

ST 示列:

结构化文本没有 DIV 指令，但您可以使用“/”实现相同的结果.

S:=A1/B1;

计算 X 的 Y 次幂 EXPT 指令

EXPT 指令计算 X 的 Y 次幂。

1）指令格式

指令	名称	梯形图表现	结构化文本表现
EXPT	计算 X 的 Y 次幂		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	底数

SourceB	源数据 B	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	指数

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

EXPY 指令计算 Source A (X) 的 Source B (Y) 次幂，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假。	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真。	将梯级输出条件设置为梯级输入条件。 Dest = Source X 的值的 Source Y 次幂。
后扫描	不适用。

4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 4，B 的值为 2，EXPT 启用时，会把 A 的值 2 次方后得到的数值 16，存储到 S 里

ST 示列:

结构化文本没有 DIV 指令，但您可以使用“/”实现相同的结果.

S:=AB;**

计算自然对数 LN 指令

LN 指令计算源值以 e 为底的自然对数。

1）指令格式

指令	名称	梯形图表现	ST 表现
LN	计算自然对数		Dest:=LN(Source)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	计算该值的自然对数

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

LN 指令计算 Source 的自然对数，并将结果存储在 Destination 中。Source 必须大于零，否则将出现轻微故障。

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。

	Dest = Source 的自然对数值。
后扫描	不适用

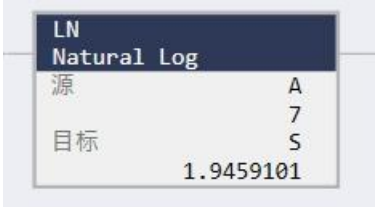
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 7，LN 启用时，LN 指令对 A 的值求自然对数，并把求到数值“1.9459101”存储到 S 里



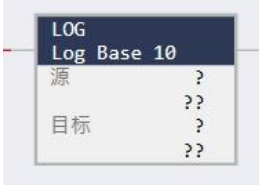
ST 示列：

```
S:= LN(A);
```

计算以 10 为低的对数 LOG 指令

LOG 指令计算源值以 10 为底的对数。

1）指令格式

指令	名称	梯形图表现	ST 表现
LOG	计算以 10 为低的对数		Dest:=LOG(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	指令查找其对数的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

LOG 指令计算 Source 的以 10 为底的对数，并将结果存储在 Destination 中。

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。

	Dest = Source 的以 10 为底的对数值。
后扫描	不适用。

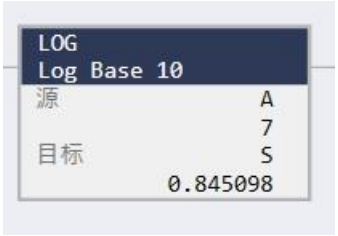
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 7，ENG 启用时，LOG 指令将 A 的值求对数，并把得到的数值“0.845098”，存储到 S 里



ST 示列：

S:=LOG(A)

余数运算 MOD 指令

MOD 指令计算源 A 除以源 B 的余数。

1）指令格式

指令	名称	梯形图表现	ST 表现
MOD	余数运算		Dest:=SourceA MOD SourceB

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	被除数的值

SourceB	源数据 B	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	除数的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后,MOD 指令和运算符用于将 Source A 除以 Source B，并将余数放入 Dest 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置（设置为余数）。
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图:

A 的值为 1，B 的值为 2，MOD 启用时，会把 A 的值与 B 的值相除后得到余数值 1，存储到 S 里

ST 示列 ：

S:=A MOD B;

乘法运算 MUL 指令

MUL 指令计算源 A 乘源 B 的值。

1）指令格式

指令	名称	梯形图表现	结构化文本表现
MUL	乘法运算		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	被乘数

SourceB	源数据 B	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	乘数

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，MUL 指令和运算符“*”用于将 Source A 和 Source B 相乘

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest = Source A x Source B
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图：

A1 的值为 2，B1 的值为 1，SUB 启用时，会把 A1 的值与 B1 的值相乘后得到值 2，存储到 S 里

ST 示列:

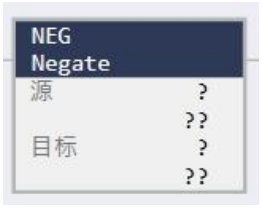
结构化文本没有 ADD 指令，但您可以使用“*”实现相同的结果

S:=A1*B1;

数值取反运算 NEG 指令

NEG 指令更改源值的符号。

1）指令格式

指令	名称	梯形图表现	ST 表现
NEG	数值取反运算		此指令不可用于 ST 结构

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	要取反的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，NEG 指令和运算符会用零减去 Source 值。

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。

	Dest = 0 - Source。
后扫描	不适用

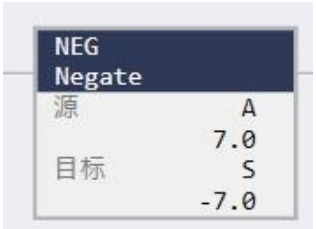
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 7，ENG 启用时，会把 A 取反的数值“-7”，存储到 S 里



ST 示列：

结构化文本没有 NEG 指令，但您可以使用“-”实现相同的结果

S:=-A

随机数 Rand 指令

Rand 指令生成 0 到 1 之间的随机数。

1）指令格式

指令	名称	梯形图表现	ST 表现
Rand	随机数		Rand(Dest);

2）相关变量

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	REAL LREAL	变量	生成最大值和最小值之间的随机数

3）功能说明

1、 触发方式：高电平触发；

2、 执行行为：

条件/状态	执行的操作
预扫描(Prescan)	不适用
梯级输入条件为假	不执行
梯级输入条件为真	Rand 指令一直执行
后扫描(Postscan)	不适用

3、 Rand 指令每次执行后，随机数自动变化；

4、ST 中右击该指令可打开参数列表；

4）注意事项

Dest 设置为变量时，必须为浮点数数据类型；

5）错误说明

6）示例

梯形图：

ST 示例：

```
My_OSR.InputBit:=Enable1;
  OSRI(My_OSR);
IF My_OSR.OutputBit THEN
  Rand(Dest_2);
END_IF;
```

最大值和最小值间的随机数 RandBetween 指令

RandBetween 指令生成最大值和最小值之间的随机数。

1）指令格式

指令	名称	梯形图表现	ST 表现
RandBetween	之间的随机数		RandBetween(Min,Max,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Size	长度	整型(详见支持的数据类型)	立即数 变量	生成随机数的最小值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	整型(详见支持的数据类型)	变量	生成最大值和最小值之间 的随机数

支持的数据类型

	布尔	整数	实数	字符串	其他

	BOOL	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	STRING	WSTRING	立即数
Min	—	✔	✔	✔	✔	✔	✔	✔	✔	—	—	—	—	✔
Max	—	✔	✔	✔	✔	✔	✔	✔	✔	—	—	—	—	✔
Dest	—	✔	✔	✔	✔	✔	✔	✔	✔	—	—	—	—	—

3）功能说明

1、 触发方式：高电平触发；

2、 执行行为：

条件/状态	执行的操作
预扫描(Prescan)	不适用
梯级输入条件为假	不执行
梯级输入条件为真	RandBetween 指令一直执行
后扫描(Postscan)	不适用

3、 RandBetween 指令每次执行后，随机数自动变化；

4、 Max>Min 时，生成 Max 和 Min 之间的随机数；

5、 Max=Min 时，生成随机数等于 Max(Min)；

6、 Max<Min 时，RandBetween 指令不执行，Dest 保持不变；

7、 ST 中右击该指令可打开参数列表；

4）注意事项

1、Max、Min、Dest 设置为变量时，必须为整型数据类型；

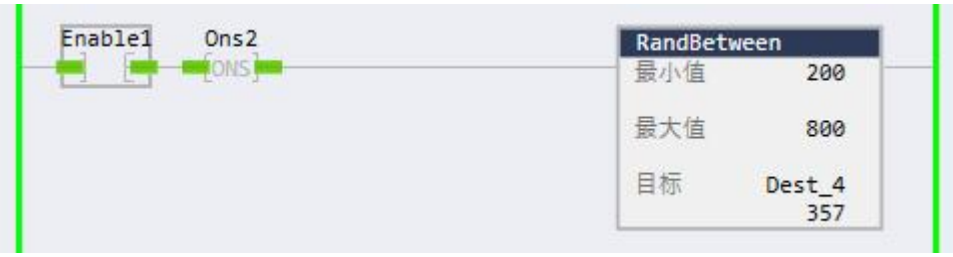
2、Max、Min 支持立即数的整型形式(如：1、2)；

5）错误说明

Max、Min 设置为立即数的浮点数形式(如：1.0、1.$、1.#NAN、1.0e+1)，则编译报错；

6）示例

梯形图：



ST 示例：

```
My_OSR.InputBit:=Enable1;
IF My_OSR.OutputBit THEN
    RandBetween(100,10000,Dest_3);
END_IF;
```

计算平方根值 SQRT 指令

SQRT 指令计算源值的平方根。

1）指令格式

指令	名称	梯形图表现	ST 表现
SQRT	计算平方根值		Dest :=SQRT(Source)

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT USINT UINT UDINT ULINT	变量 立即数	计算该值的平方根

		REAL LREAL		

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

SQR 指令和运算符用于计算 Source 的平方根，并将结果放入 Dest 中。

条件/状态	执行的操作
预扫描	不适用
将梯级输出条件设置为梯级输入条件。 Dest = Source 的平方根。	将梯级输出条件设置为梯级输入条件。
不适用	
后扫描	

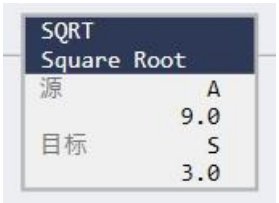
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 9，SQRT 启用时，会把 A 平方根的值“3”，存储到 S 里



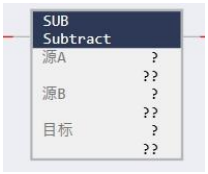
ST 示列：

S:=SQRT(A)

减法运算 SUB 指令

SUB 指令计算源 A 减去源 B 的值。

1）指令格式

指令	名称	梯形图表现	ST 表现
SUB	减法运算		此指令不可用于 ST 结构

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明

SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	用来减去源数据 B 的值
SourceB	源数据 B	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量 立即数	源数据 A 要减去的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

SUB 指令和运算符“-”用于将 Source A 减 Source B

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest = Source A - Source B
后扫描	不适用

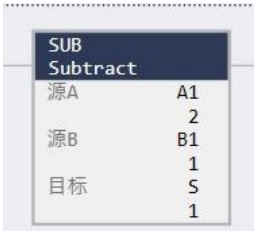
4）注意事项

5）错误说明

6）示列

梯形图示列 ：

A1 的值为 2，B1 的值为 1，SUB 启用时，会把 A1 的值与 B1 的值相减后得到值 1，存储到 S 里



ST 示列 ：

结构化文本没有 SUB 指令，但您可以使用“-”实现相同的结果

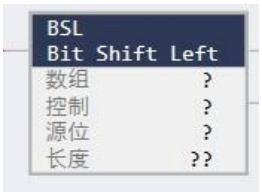
```
S:=A1-B1;
```

七、数据结构操作指令

位左移 BSL 指令

BSL 指令用于将数组中的指定位左移一位

1）指令格式

指令	名称	梯形图表现	ST 表现
BSL	位左移		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Array	数组	DINT	变量	要修改的数组 指定元素组的第一个元素 不要在下标中使用 CONTROL.POS
Control	控制	CONTROL	变量	控制运算的结构

Source_bit	源位	Bool	变量	要移动的位
Length	长度	DINT	立即数	数组中要移动的位数

CONTROL 结构

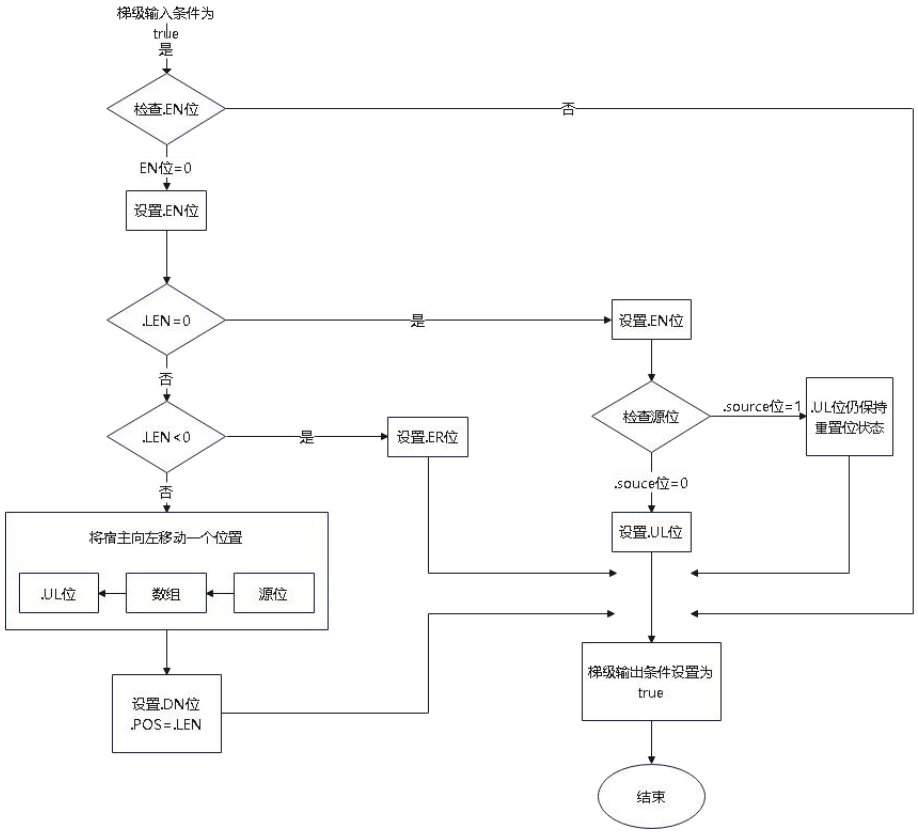
助记码:	说明:
.EN （启用）	启用位指示 BSL 指令已启用。
.DN （完成）	设置完成位以指示位已经向左移动了一个位置。
.UL （卸载位）	卸载位是指令的输出。.UL 位存储已移出位范围的位的状态。
.ER （错误）	当.LEN<0 时将设置错误位。
.LEN （移动数目）	长度指定要移动的数组位的数目。

3）功能说明

条件/状态	执行的操作
预扫描	.EN 位设置为假。 .DN 位设置为假。 .ER 位设置为假。 .POS 值清零
梯级输入条件为假	.EN 位设置为假。

	.DN 位设置为假。 **.ER 位设置为假。** **.POS 值清零。**
梯级输入条件为真	**请参见“BSL 流程图（真）”**
后扫描	**不适用**

流程图 ：



4）注意事项

您必须测试并确认指令不会更改您不希望更改的数据。BSL 指令对连续内存进行操作。在某些情况下，该指令跨过数组将位移到标记的其他成员。如果长度过大并且标记是用户自

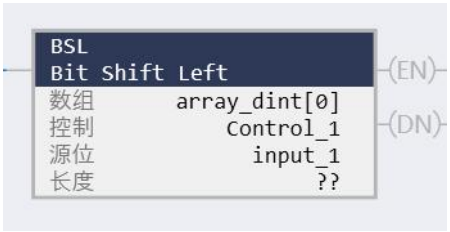
定义的数据类型，则会发生这种情况

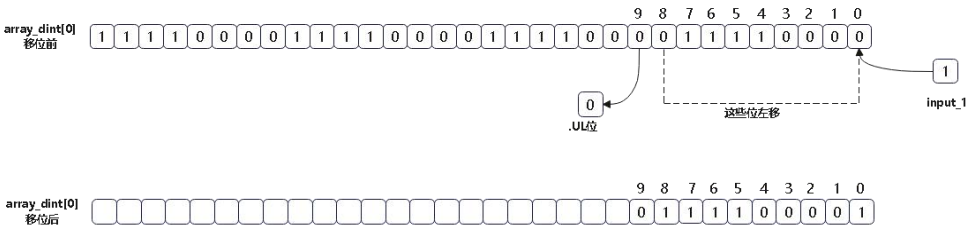
5）错误说明

6）示列

梯形图示列 ：

启用时，BSL 指令起始于 array_dint[0] 中的 0 位。指令将 array_dint[0].9 卸载到.UL 位，同时移动其余的位，并将 input_1 加载到 array_dint[0].0。其余位（10-31)中的值无效。

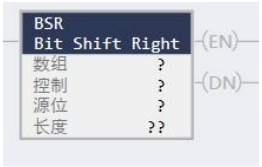




位右移 BSR 指令

BSR 指令用于将数组中的指定位右移一位。使能后，指令将数组的位 0 的值卸载到 .UL 位，将其余的位右移一位，并从位地址加载该位

1）指令格式

指令	名称	梯形图表现	ST 表现
BSR	位右移		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Array	数组	DINT	变量	要修改的数组 指定元素组的第一个元素 不要在下标中使用 CONTROL.POS
Control	控制	CONTROL	变量	控制运算的结构
Source_bit	源位	Bool	变量	要移动的位

Length	长度	DINT	立即数	数组中要移动的位数

CONTROL 结构

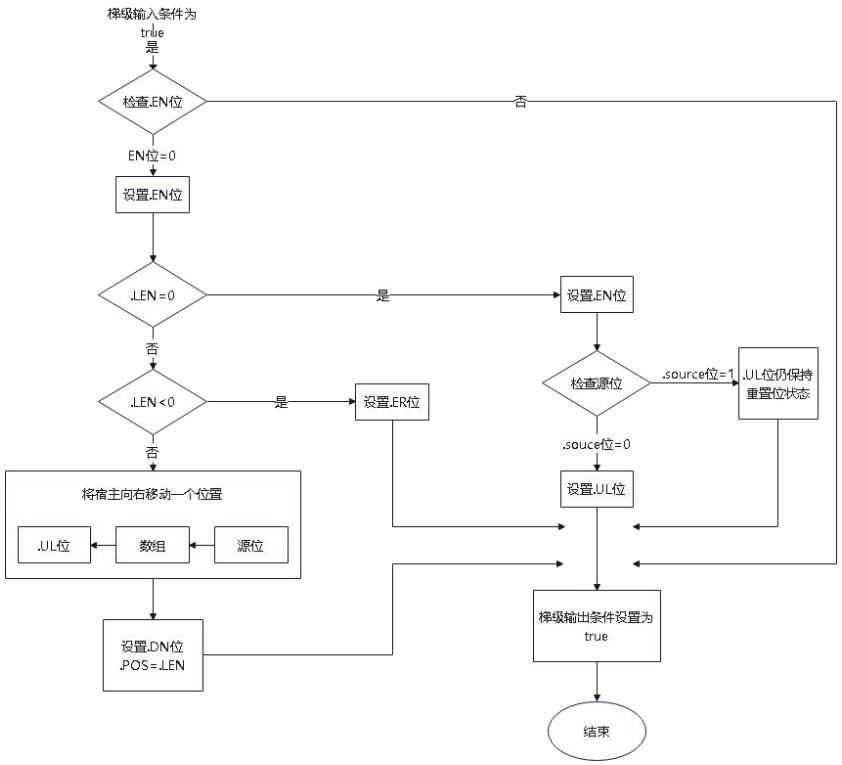
助记码:	说明:
.EN （启用）	启用位指示 BSR 指令已启用。
.DN （完成）	设置完成位以指示位已经向右移动了一个位置。
.UL （卸载位）	卸载位是指令的输出。.UL 位存储已移出位范围的位的状态。
.ER （错误）	当.LEN<0 时将设置错误位。
.LEN （移动数目）	长度指定要移动的数组位的数目。

3）功能说明

条件/状态	执行的操作
预扫描	.EN 位设置为假。 .DN 位设置为假。 .ER 位设置为假。 .POS 值清零。
梯级输入条件为假	.EN 位设置为假。 .DN 位设置为假。

	.ER 位设置为假。 **.POS 值清零。**
梯级输入条件为真	**请参见下文 BSR 流程图（真）**
后扫描	**不适用**

流程图 ：



4）注意事项

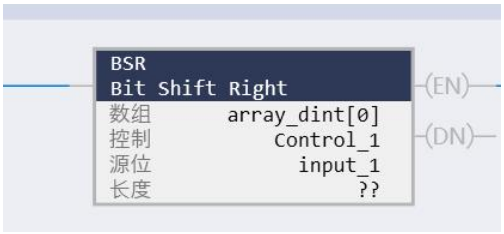
您必须测试并确认指令不会更改您不希望更改的数据。BSR 指令对连续内存进行操作。在某些情况下，该指令跨过数组将位移到标记的其他成员。如果长度过大并且标记是用户自定义的数据类型，则会发生这种情况

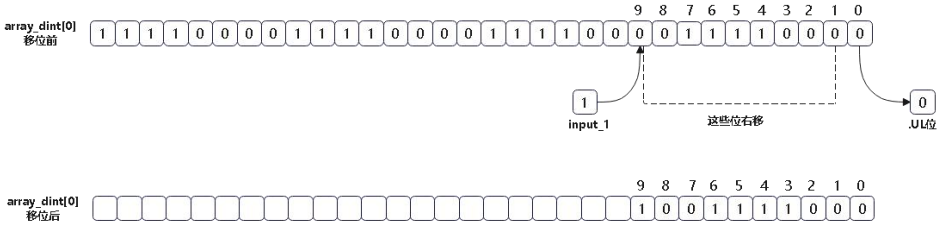
5）错误说明

6）示列

梯形图：

启用时，BSR 指令从 array_dint[O]的位 9 开始执行。指令将 array_dint[o].0 卸载到.UL 位，右移其余的位，并将 input_1 加载到 array_dint[0].9。其余位 (10-31) 中的值无效。





八、逻辑运算指令

按位与 AND 指令

AND 指令将源 A 与源 B 的值按位与运算。

1）指令格式

指令	名称	梯形图表现	ST 表现
AND	按位与		此指令不可用于结构化文本中（但在表达式中使用 AND 运算可以实现相同效果）

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT	变量 立即数	要与源数据 B 执行按位与的值
SourceB	源数据 B	SINT	变量	要与源数据 A 执行按位与的

		INT DINT LINT USINT UINT UDINT ULINT	立即数	值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT	变量	存储计算结果的变量

3）功能说明

使能后，AND 指令利用 SourceA 和 SourceB 中的位执行按位与运算，并将结果放在 Destination 中。

如果 SourceA 的位值为：	如果 SourceB 的位值为：	Dest 中的位值为：
0	0	0
0	1	0
1	0	0
1	1	1

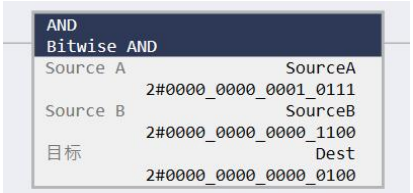
条件	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

4）注意事项

5）错误说明

6）示例

梯形图：



ST 示例：

Dest:=SourceA AND SourceB

位域复制 BTD 指令

BTD 指令将从 Source 复制指定位，将这些位进行相应移位，再将这些位写入 Destination 中

1）指令格式

指令	名称	梯形图表现	ST 表现
BTD	位域复制		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	SINT INT DINT SINT 或 INT 变量通过符号扩展转换为 DINT 值。	立即数或变量	要移动的位的变量
Source Bit	源位	DINT	立即数 (0-31 DINT) (0-15INT) (0-7 SINT)	位编号（最低位编号），将从该位置开始移

				动 必须在 Source 数 据类型的有 效 范围内
Dest	目标	SINT INT DINT	变量	位要移动到 的位置的标 记
Dest Bit	目标位	DINT	立即数 (0-31 DINT) (0-15INT) (0-7 SINT)	位编号（最 低位编号）， 将从该 位置开始复 制 Source 中的位 必须在 Destinatio n 数据类型 的有 效范围内
Length	长度	DINT	立即数	要移动的位

				数

3）功能说明

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假。	不适用
梯级输入条件为真。	该指令读取 Source 位，并在移位后复制到 Destination 中。
后扫描	不适用

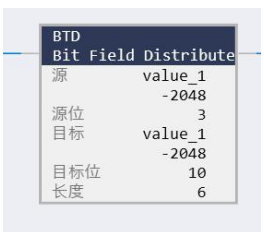
4）注意事项

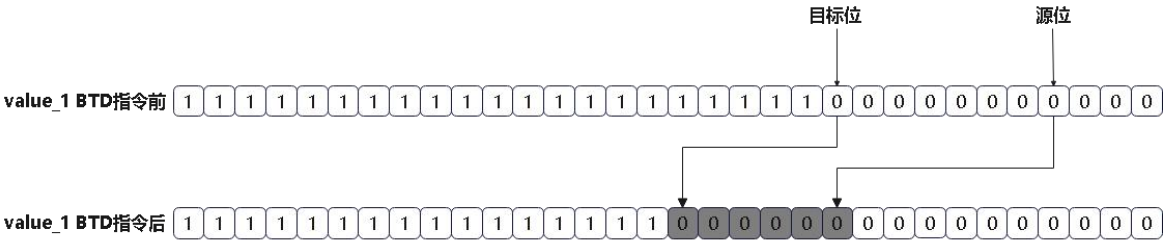
5）错误说明

6）示列

梯形图 ：

启用时，BTD 指令在 value_1 内部移动位。





带阴影的框显示的是value_1中已更改的值

位域复制 BTDT 指令-ST

BTDT 指令首先将 Target 复制到 Destination。随后从 Source 中复制指定位，将这些位进行相应移位，再将这些位写入 Destination 中。Target 和 Source 值则保持不变。

1）指令格式

指令	名称	梯形图表现	ST 表现
BTDT	位域复制	此指令不可用于梯形图中	BTDT(BTDT_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
BTDT_tag	FBD_BIT_FIELD_DISTRIBUTE	结构	BTDT 结构

FBD_BIT_FIELD_DISTRIBUTE 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
Source	源输入值	DINT	变量 立即数	输入值，其中包含要移动到 Destination 中的位。

				有效值 = 任意整数
SourceBit	移动起始位的最低位编号	DINT	变量 立即数	Source 中的位编号（移动起始位的最低位编号）。 有效值 = 0-31
Length	要移动的位数	DINT	变量 立即数	要移动的位数。 有效值 = 1-32
DesBit	目标起始位的最低位编号	DINT	变量 立即数	Dest 中的位编号（目标起始位的最低位编号）。 有效值 = 0-31
Target	目标	DINT	变量 立即数	在移动 Source 中的位之前要移动到 Dest 中的输入值。 有效值 = 任意整数

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	启用输出。
Dest	结果输出	DINT	变量	位移动操作的结果。

3）功能说明

当条件为真时，BTDT 指令首先将 Target 复制到 Destination 中，然后将 Source 中的一组位复制到 Destination 中。这组位由 Source bit（位组的最低位编号）和

Length（要复制的位数）确定。Destination bit 用于确定 Destination 中的最低起始位号。Source 和 Target 保持不变。

如果位域的长度超出 Destination 的范围，则该指令不会保存多余的位。任何多余位都不会回绕到下一个字。

执行行为

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

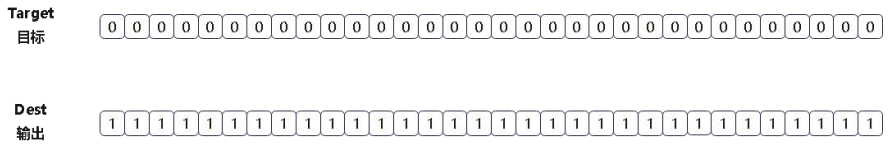
4）注意事项

5）错误说明

6）示例

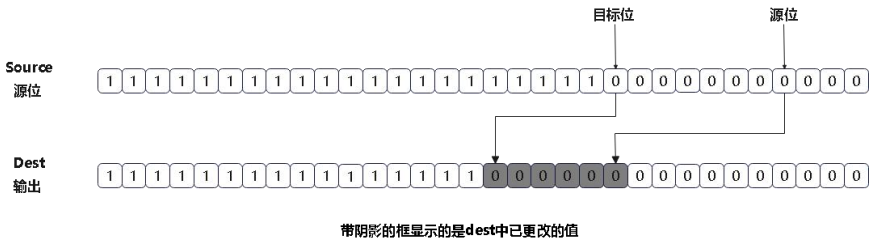
步骤一

控制器将 Target 复制到 Dest。



步骤二

SourceBit 和 Length 指定将 Source 中的哪些位复制到 Dest 中。目标起始位由 DestBit 确定，Source 和 Target 保持不变。



ST 示例：

```
BTDT_01.Source := sourceSTX;

BTDT_01.SourceBit := source_bitSTX;

BTDT_01.Length := LengthSTX;

BTDT_01.DestBit := dest_bitSTX;

BTDT_01.Target := TargetSTX;

BTDT(BTDT_01);

distributed_value := BTDT_01.Dest;
```

数值清零 CLR 指令

CLR 指令可将 Dest 的所有位清零

1）指令格式

指令	名称	梯形图表现	ST 表现
CLR	数值清零		CLR(Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT	变量	要清零的变量

		ULINT REAL LREAL		

3）功能说明

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 将 Dest 清零。
后扫描	不适用

4）注意事项

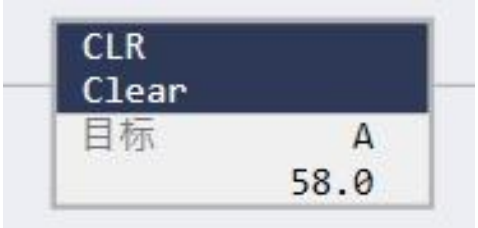
5）错误说明

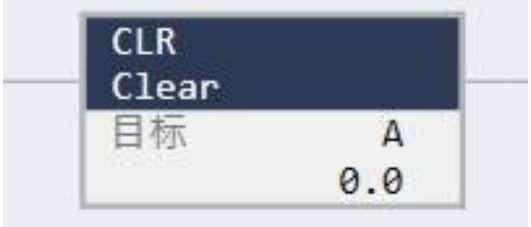
6）示列

梯形图：

示例 1：

CLR，指令启动时，会把“A”里面的数据清零





ST 示列：

CLR(A);

D 触发器 DFF 指令-ST

当时钟从清零变为设置时，DFF 指令设置 Q=D。设置 Clear 时，Q 被清除。DFF 指令将 QNot 设置为 与 Q 相反的状态。

1）指令格式

指令	名称	梯形图表现	ST 表现
DFF	D 触发器	此指令不可用于梯形图中	DFF(DFF_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
DFF_tag	FLIP_FLOP_D	结构	DFF 结构

DOMINANT_SET 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
D	指令的输入	BOOL	变量	指令的输入。

			立即数	默认清零。
Clear	指令的复位输入	BOOL	变量 立即数	指令的复位输入。 如果设置,指令清除 Q 并设置 QNot
Clock	时钟输入到指令	BOOL	变量 立即数	时钟输入到指令。 默认已清除。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Q	指令输出	BOOL	变量	指示的输出
QNot	指令输出取反	BOOL	变量	指令输出的取反。

3）功能说明

当清除信号设置时，指令清除 Q 并设置 QNot。否则，如果时钟信号设置且上一周期的时钟信号（Clockn-1）被清除，指令将设置 Q=D，并设置 QNot = 非(D)。

每扫描一次，指令将上一周期的时钟状态（Clockn-1）设置为当前时钟状态。

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

当时钟从清除状态变为设置状态时，DFF 指令将 Q 设置为等于 D。当清除信号被设置时，Q 被清除。DFF 指令将 QNot 设置为与 Q 相反的状态。

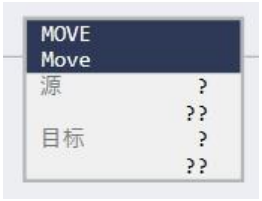
ST 示例：

```
DFF_03.D := d_input;
DFF_03.Clear := clear_input;
DFF_03.Clock := clock_input;
DFF(DFF_03);
q_output := DFF_03.Q;
qNot_output := DFF_03.QNot;
```

移动赋值 MOVE 指令

MOVE 指令可将 Source 的副本移动到 Dest 中。Source 保持不变

1）指令格式

指令	名称	梯形图表现	ST 表现
MOVE	移动赋值		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT REAL SINT 或 INT 变量通过符号扩展转换为 DINT 值	变量 立即数	要移动（复制）的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT	变量	存储计算结果的变

		INT DINT REAL		量

3）功能说明

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 该指令可将 Source 复制到 Dest 中。 字符串型操作数： 如果 Source.LEN > SIZE(Dest.DATA) 字符串将截断为相应长度 S:V 置位。
后扫描	不适用

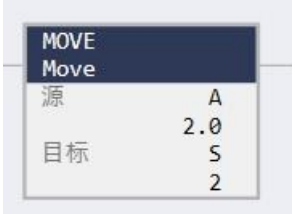
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 2 ，MOVE 启用时，MOVE 指令将 A 的值 2，移动到 S 里存储。



ST 示列：

结构化文本没有 MOVE 指令，但您可以使用“=”实现相同的结果

S:=A;

屏蔽移动 MVM 指令

MVM 指令将源 Source 复制到 Destination，并允许屏蔽数据的某些部分。

MVM 指令使用屏蔽码来传递或屏蔽 Source 数据位。屏蔽码中的“1”表示传递数据位；屏蔽码中的“0”表示屏蔽数据位

1）指令格式

指令	名称	梯形图表现	ST 表现
MVM	屏蔽移动		此指令不可用于 ST 结构

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	SINT INT DINT SINT 或 INT 变量通过符号扩展转换为 DINT 值。	立即数或变量	要移动的值

Mask	掩码	SINT INT DINT SINT 或 INT 变量 通过符号扩展转换 为 DINT 值。	立即数或变量	要阻止传递的位

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储结果的变量

3）功能说明

操作数	数据类型
Source	SINT INT DINT
Mask	SINT

	INT DINT
Dest	SINT INT DINT
操作数	数据类型

4）注意事项

5）错误说明

6）示列

梯形图：

屏蔽移动 MVMT 指令-ST

MVMT 指令将掩码部分的源值复制到目标值，并允许屏蔽部分数据。

1）指令格式

指令	名称	梯形图表现	ST 表现
MVMT	屏蔽移动	此指令不可用于梯形图中。	MVMT(MVMT_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
MVMT_tag	FBD_MASKED_MOVE	结构	MVMT 结构

FBD_MASKED_MOVE 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
Source	源输入值	DINT	变量 立即数	要经 Mask 屏蔽码处理后移动到 Destination 中的输入值。 有效值 = 任意整数

Mask	掩码	DINT	变量 立即数	要从 Source 移动到 Dest 中的位的屏蔽码。如果将所有位置 1，可将相应位从 Source 移动到 Dest 中。如果将所有位清零，将阻止相应位从 Source 移动到 Dest 中。 有效值 = 任意整数
Target	目标	DINT	变量 立即数	在移动 Source 中的位之前要移动到 Dest 中的输入值。 有效值 = 任意整数

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	启用输出。
Dest	结果输出	DINT	变量	屏蔽移动操作的结果。

3）功能说明

使能后，MVMT 指令使用屏蔽码来传递或屏蔽 Source 数据位。屏蔽码中的“1”表示将传递相应的数据位。屏蔽码中的“0”表示会阻止相应的数据位。

如果混用多种整型数据类型，该指令会将较小整型数据类型的高位填零，从而确保其与最大数据类型大小相同。

4）注意事项

输入掩码时，编程软件默认为十进制值。如果要使用其他格式输入掩码，请在值的前面加上正确的前缀。

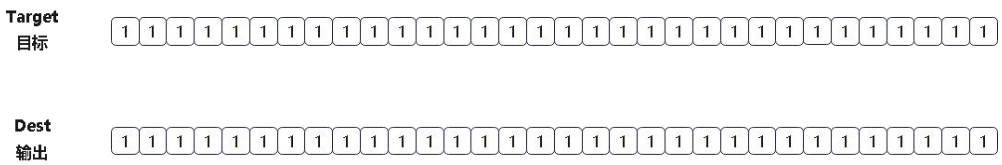
前缀：	说明
16#	16 进制 例如：16#0A0A
8#	8 进制 例如：8#23
2#	2 进制 例如：2#00110011

5）错误说明

6）示例

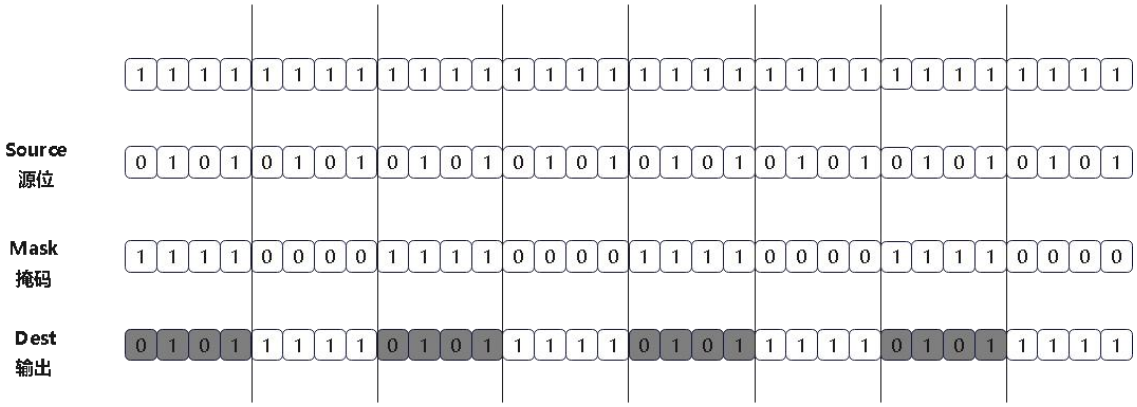
步骤 1

控制器将 Target 复制到 Dest。



步骤 2

该指令对 Source 进行屏蔽处理，并将其与 Dest 进行比较。在 Dest 中进行任何必要的更改后，即构成 value_masked 的输入参数。Source 和 Target 保持不变。屏蔽码中的 0 会阻止指令比较相应位。



ST 示例：

```
MVMT_01.Source := value_1;

MVMT_01.Mask := mask_1;

MVMT_01.Target := target;

MVMT(MVMT_01);

value_masked := MVMT_01.Dest;
```

按位非 NOT 指令

NOT 指令将源的值按位非运算给目标值。

1）指令格式

指令	名称	梯形图表现	ST 表现
NOT	按位非		此指令不可用于结构化文本中（但在表达式中使用 NOT 运算可以实现相同效果）

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT USINT UINT UDINT ULINT	变量 立即数	要执行按位取反的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT	变量	存储计算结果的变量

3）功能说明

使能后，NOT 指令利用 Source 中的位执行按位非运算，并将结果放在 Destination 中。

如果 Source 的位值为：	Dest 中的位值为：
0	1
1	0

条件	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说

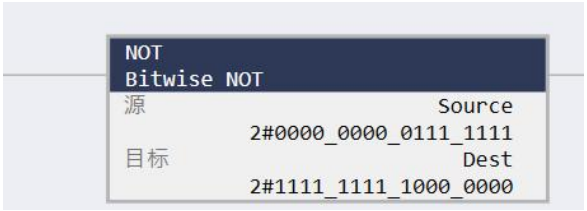
	明”部分所述进行设置。
后扫描	不适用

4）注意事项

5）错误说明

6）示例

梯形图：



ST 示例：

```
Dest:=Soure;
```

按位或 OR 指令

OR 指令将源 A 与源 B 的值按位或运算。

1）指令格式

指令	名称	梯形图表现	ST 表现
OR	按位或		此指令不可用于结构化文本中（但在表达式中使用 OR 运算可以实现相同效果）

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT	变量 立即数	要与源数据 B 执行按位或的值
SourceB	源数据 B	SINT INT DINT	变量 立即数	要与源数据 A 执行按位或的值

		LINT USINT UINT UDINT ULINT		

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT	变量	存储计算结果的变量

3）功能说明

使能后，OR 指令利用 SourceA 和 SourceB 中的位执行按位或运算，并将结果放在 Destinatio 中。

如果 SourceA 的位值为：	如果 SourceB 的位值为：	Dest 中的位值为：

0	0	0
0	1	1
1	0	1
1	1	1

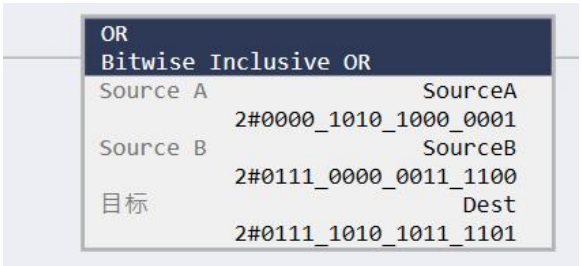
条件	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

4）注意事项

5）错误说明

6）示例

梯形图：



ST 示例：

```
Dest:=SourceA OR SourceB;
```

优先置位 SETD 指令-ST

SETD 指令使用 set 和 Reset 输入来控制锁存输出。Set 输入优先于 Reset 输入。

1）指令格式

指令	名称	梯形图表现	ST 表现
SETD	优先置位	此指令不可用于梯形图中	SETD(SETD_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
SETD_tag	DOMINANT_SET	结构	SETD 结构

DOMINANT_SET 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
Set	源输入值	BOOL	变量 立即数	指令的置位输入。 默认清零。
Clear	移动起始位的	BOOL	变量	指令的复位输入。

	最低位编号		立即数	默认清零。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	BOOL	变量	指令是否处于启用状态。
OutNot	指令输出取反	BOOL	变量	指令输出的取反。

3）功能说明

优先置位指令使用 Set 和 Reset 输入参数控制锁存输出参数 Out 和 OutNot。Set 输入的优先级高于 Reset 输入。

Set 输入参数设置为真时，Out 将锁存为真状态。只有当 Set 输入为假时，将 Reset 参数设置为真才会使 Out 变为假。OutNot 将设置为 Out 的非状态。

执行行为

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

Set 为真时，Out 设置为真。Set 为假且 Reset 为真时，Out 清零。Set 输入的优先级高于 Reset 输入。SETD 指令将 OutNot 设置为 Out 的非状态。

ST 示例：

```
SETD_01.Set := set_input;
SETD_01.Reset := reset_input;
SETD(SETD_01);
out_output := SETD_01.Out;
outNot_output := SETD_01.OutNot;
```

交换字节 SWPB 指令

SWPB 指令可重新排列 Source 的字节顺序。然后将结果放入 Destination 中

1）指令格式

指令	名称	梯形图表现	ST 表现
SWPB	交换字节		SWPB(Source,Order Mode,Dest);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	INT DINT	变量	包含要重新排列的字节的变量

Order Mode	排列方式		输入 REVERSE(或输入 0)，排列方式为 ABCD→DCBA； 输入 WORD (或输入 1)，排列方式为 ABCD→CDAB； 输入 HIGH/LOW(或输入 2)，排列方式为 ABCD→BADC；	选择更改字节的排列方式

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	INT DINT	变量（如果 Source 是 INT， 则 Dest 必须是：INT 或 DINT；如果 Source 是 DINT，则 Dest 必须是：DINT；）	按新顺序存储字节的变量

3）功能说明

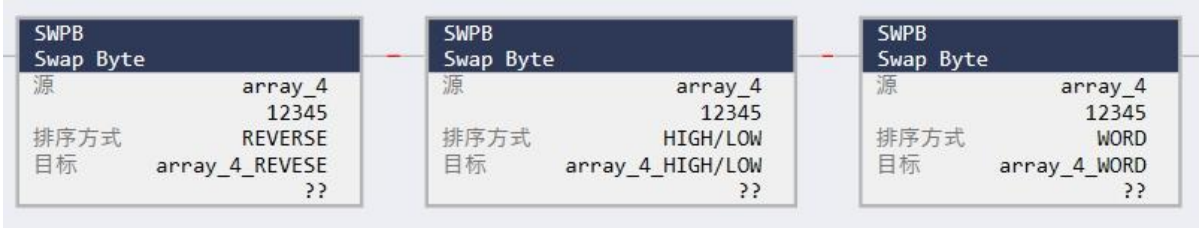
条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	该指令可将指定字节重新排列。
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图：



ST 示列：

```
SWPB(DINT_1,REVERSE,DINT_1_reverse);

SWPB(DINT_1,WORD,DINT_1_swap_word);

SWPB(DINT_1,HIGHLOW,DINT_1_swap_high_low);
```

按位异或 XOR 指令

XOR 指令将源 A 与源 B 的值按位异或运算。

1）指令格式

指令	名称	梯形图表现	ST 表现
XOR	按位异或		此指令不可用于结构化文本中（但在表达式中使用 XOR 运算可以实现相同效果）

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
SourceA	源数据 A	SINT INT DINT LINT USINT UINT UDINT ULINT	变量 立即数	要与源数据 B 执行按位异或的值
SourceB	源数据 B	SINT	变量	要与源数据 A 执行按位异或

		INT DINT LINT USINT UINT UDINT ULINT	立即数	的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT USINT UINT UDINT ULINT	变量	存储计算结果的变量

3）功能说明

使能后，XOR 指令利用 Source A 和 Source B 中的位执行按位异或运算，并 将结果放在 Destination 中。

如果 SourceA 的位值为：	如果 SourceB 的位值为：	Dest 中的位值为：
0	0	0
0	1	1
1	0	1
1	1	0

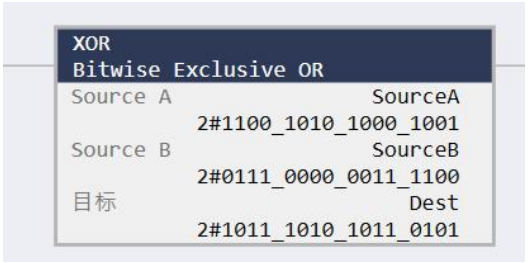
条件	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件
梯级输入条件为真	将梯级输出条件设置为梯级输入条件 Dest 按“说明”部分所述进行设置。
后扫描	不适用

4）注意事项

5）错误说明

6）示例

梯形图：



ST 示例：

```
Dest:=SourceA XOR SourceB
```

九、运动功能指令

运动轴注册 MAR 指令

MAR 指令对指定轴的指定运动模块注册输入事件检查。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAR	运动轴注册		MAR(Axis, //轴 MotionControl, //指令变量 Positive_Edge, //触发条件 Disabled, //窗口登录 Minimum_Position,//最小位置 Maximum_Position,//最大位置 1); //输入编号

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Trigger Condition	触发条件	BOOL	立即数	定义了注册输的转换，该转换决定了注册事件。可以选择以下两种触发条件之一： 0 = 在正边沿触发： 当注册输入信号从低电平变为高电平时，触发记录事件。 1 = 在负边沿触发： 当注册输入信号从高电平变为低电平时，触发记录事件。
Windowed Registration	窗口记录	BOOL	立即数	如果记录事件需要在指定的最小和最大位置限制内有效，即计算出的记录位置必须落在这些限制内才被视为有效的记录事件，则启用（1）。选择以下选项之一： 0 = 禁用 1 = 启用
Minimum Position	最小位置	REAL	立即数 变量	当启用窗口记录时使用。记录位置必须大于最小位置限制，记录事件才会被接受

Maximum Position	最大位置	REAL	立即数 变量	当启用窗口记录时使用。记录位置必须小于最大位置限制，记录事件才会被接受。
Input Number	输入信号通道	UINT32	立即数	指定要选择的记录输入。 1 = 记录 1 位置 2 = 记录 2 位置

MOTION_INSTRUCTION 结构

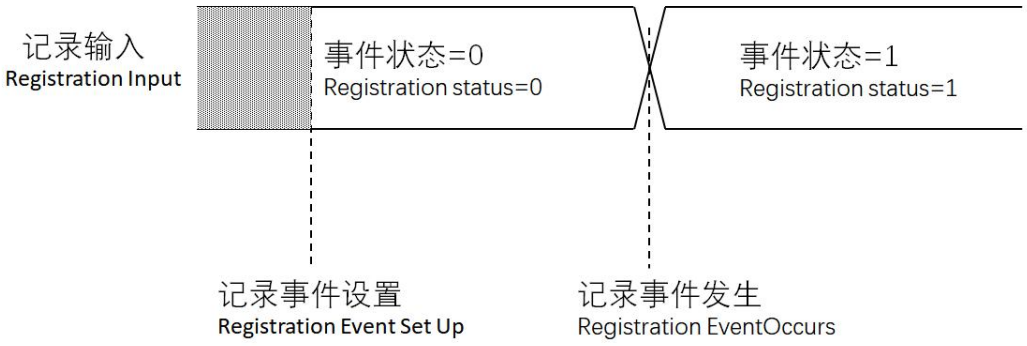
助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴记录事件插件已经成功准备后，完成位被置 1。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。
.IP（正在进行）位 26	在正的梯级过渡时被置 1，在记录事件发生后，或者被其他运动记录准备所取代或者被运动禁止记录命令所终止时被置 0
.PC（过程完成）位 27	当记录事件发生时被置 1。

3）功能说明

MAR 指令设置一个记录事件，用于在指定的专用高速记录输入的选定边沿上存储指定物理轴的实际位置。

当执行 MAR 指令时，RegEventStatus 位被置为 0（FALSE），并监测指定轴的选定记录输入，直到发生选定类型的记录输入转换（即记录事件）。当记录事件发生时，该轴的

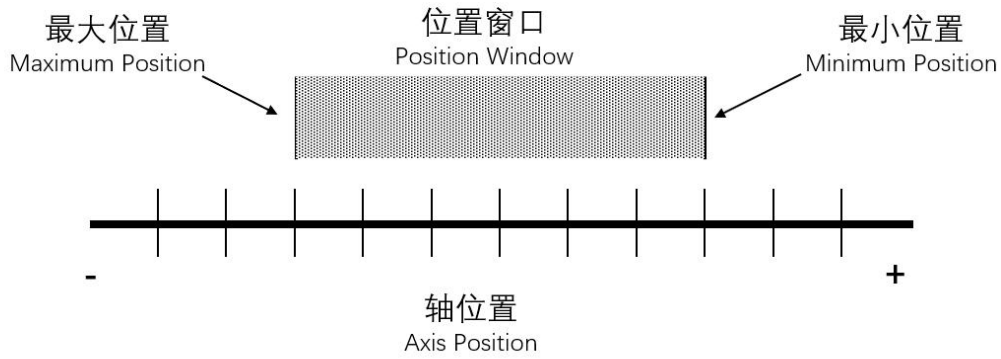
RegEventStatus 位被置为 1（TRUE），并且轴的实际位置被存储在对应于注册输入的记录位置变量中（例如，记录 1 位置（Registration1Position）或记录 2 位置（Registration2Position））。



对于给定的轴，可以同时激活多个记录事件，但每个记录输入只能激活一个事件。每个事件都是独立监控的，并且可以使用相应的 RegEventStatus 位进行检查。

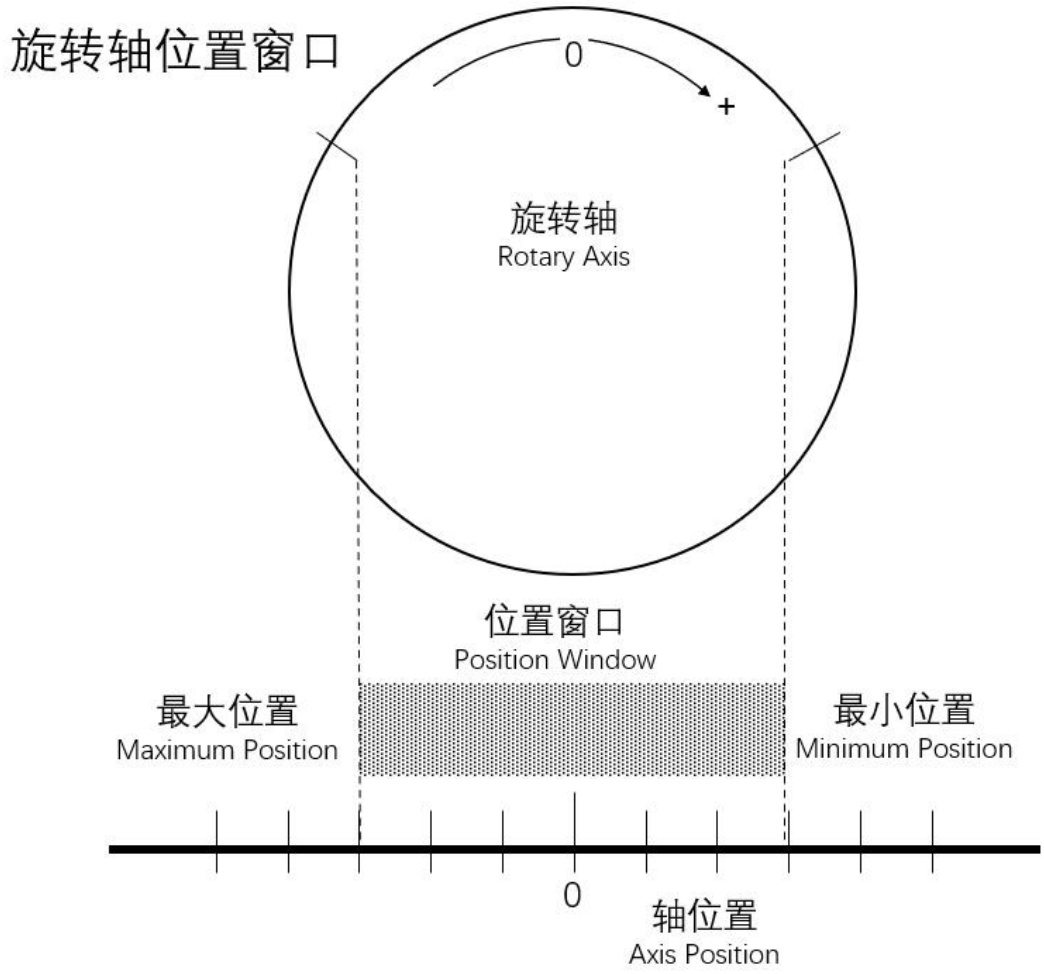
窗口记录

当选中“窗口记录”复选框时，只有当轴处于由最小和最大位置定义的窗口内时，选定的触发状态才会导致记录事件的发生。



输入定义位置窗口的期望绝对位置的值或变量，该位置窗口内选定的注册输入的触发状态是有效的。窗口记录有助于提供一种机制，以忽略记录传感器的虚假或随机转换，从而提高高速记录输入的抗噪能力。

对于线性轴，这些值可以是正数、负数或两者的组合。但是，为了使记录事件发生，最小位置值必须小于最大位置值。对于旋转轴，这两个值都必须小于运动控制器的机器设置菜单中设置的解绕值。对于跨越轴的解绕点的记录窗口，最小位置值可以大于最大位置值，如下所示：



执行行为：

情况	行为
Prescan	.EN、.DN、.ER 和 .IP 位被置为 False。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。
执行条件为 TRUE	.EN 置为 True，指令执行

Postscan	不适用

MAR 对状态位的更改：

位名称	状态	含义
RegEvent1ArmedStatus RegEvent2ArmedStatus	TRUE	轴正在寻找记录事件。
RegEvent1Status RegEvent2Status	FALSE	之前的记录事件已被清除

4）注意事项

1. 在大型 I/O 连接中，强制值可能会降低控制器处理重复运动注册的速度。

2. 要成功执行 MAR 指令，目标轴必须被配置为伺服轴（Servo axis）或仅反馈轴（Feedback Only axis）。否则，指令将出错。

3. 该指令的执行可能需要多个扫描周期来完成，因为它需要进行多次粗略更新才能完成请求。完成位（.DN）不会立即置 1，只有在请求完成后才会被置 1。

5）错误说明

扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAR 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码(十进制)	扩展错误代码（十进制）	说明

SERVO_MESSAGE_FAILURE（12）	无资源（2）	没有足够的内存资源来完成请求。
SERVO_MESSAGE_FAILURE（12）	无效数据（3）	提供的记录输入超出了范围。
SERVO_MESSAGE_FAILURE（12）	设备处于错误状态（16）	重新定义位置、归零和记录 2 是相互排斥的。

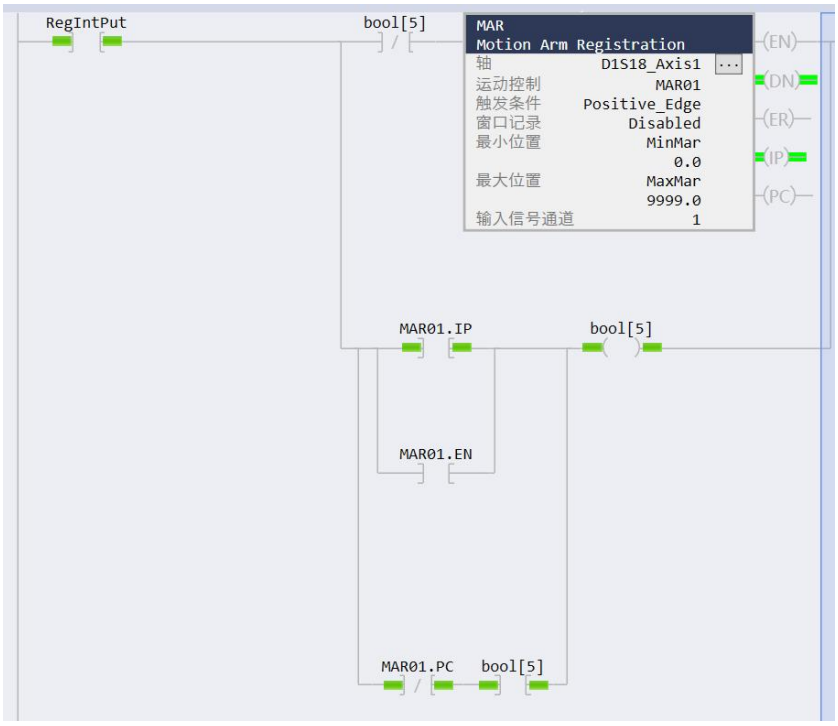
参数超出范围错误的扩展错误代码工作方式有点不同。不是采用标准的枚举，扩展错误代码所显示的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MAR 指令，扩展错误代码 4 指的是最小位置值。然后必须将您的值与指令可以接受的值范围进行对比。

6）示例

梯形图：

如果您的应用需要快速连续地探测记录传感器，我们建议您使用以下逻辑：

当 MAR 输入条件为真时，控制器准备 Axis 上的伺服模块记录事件检查。



ST 示例：

```
MAR(
    Axis,                   //轴
    MotionControl,          //指令变量
    Positive_Edge,          //触发条件
    Disabled,               //窗口登录
    Minimum_Position,       //最小位置
    Maximum_Position,//最大位置
    1);                     //输入编号
```

运动轴监控 MAW 指令

MAW 指令指定轴的监控位置事件检查。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAW	运动轴监控		MAW(Axis, //轴 MotionControl, //指令名称 Forward, //触发条件 Position); //位置

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
MotionControl	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
TriggerCondition	触发条件	BOOL	立即数	选择监视事件触发条件： 0 = 正向 ：伺服模块查找实际位置从小于监视位置变为大于监视位置的变化。

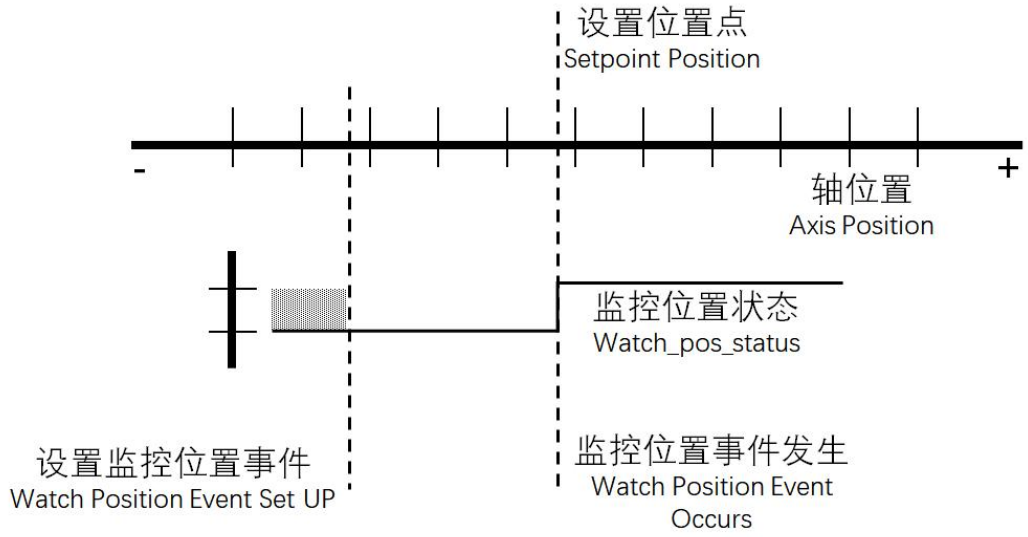
				1 = 反向 ：伺服模块查找实际位置从大于监视位置变为小于监视位置的变化。
Position	位置	REAL	立即数 变量	位置监视的新值。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴监视事件插件已经成功准备后，完成位被置 1。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。
.IP（正在进行）位 26	在正的梯级过渡时被置 1，在监视事件发生后，或者被其他关闭运动监视所取代或者被运动禁止监视命令所终止时被置 0.
.PC（过程完成）位 27	当监视事件发生时被置 1。

3）功能说明

MAW 指令设置一个监视位置事件，当指定的物理轴达到指定的设定点位置时触发该事件。如下所示：



设置点位置：

监视位置事件在轴移动时用于将操作同步到指定的轴位置，例如在特定轴位置激活电磁阀以将纸箱从传送带上推下。选择或输入所需的物理轴、所需的触发条件，并输入所需的监视位置的值或变量。

如果目标轴没有出现在可用轴的列表中，说明该轴尚未被配置为可操作。可以使用变量编辑器（Tag Editor）来创建和配置新的轴。

当执行监视位置指令时，WatchEventStatus 位被置为 0（FALSE），并且物理轴的实际位置（以伺服循环更新速率）会被监测，直到它达到指定的监视位置。监视位置事件发生后，该轴的 WatchEventStatus 位被置为 1（TRUE）。

在给定时间内可以激活多个监视位置事件，但任何给定的物理轴上一次只能激活一个。每个事件都是独立监控的，并且可以使用相应的 WatchEventStatus 位进行检查。

执行行为：

情况	行为
Prescan	.EN、.DN、.ER 和 .IP 位被置为 False。
执行条件为	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。否则，.EN 位不会受到影

FALSE	响。.DN、.ER、和.PC 位不受影响
执行条件为 TRUE	.EN 置为 True，指令执行
Postscan	不适用

MAW 对状态位的更改：

位名称	状态	含义
WatchEventArmedStatus	TRUE	轴正在寻找监视位置事件。
WatchEventStatus	FALSE	之前的监视事件已被清除。

4）注意事项

在大型 I/O 连接中，强制值可能会降低控制器处理重复运动注册的速度。

要成功执行 MAR 指令，目标轴必须被配置为伺服轴（Servo axis）或仅反馈轴（Feedback Only axis）。否则，指令将出错

该指令的执行可能需要多个扫描周期来完成，因为它需要进行多次粗略更新才能完成请求。完成位（.DN）不会立即置 1，只有在请求完成后才会被置 1。

5）错误说明

扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAW 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码 (十进制)	扩展错误代码 （十进制）	说明

SERVO_MESSAGE_FAILURE (12)	无资源（2）	没有足够的内存资源来完成请求。

6）示例

梯形图：

当 MAW 输入条件为真时，控制器准备 Axis 上的位置监视事件检查。

Position	REAL	
▸ MotionControl	MOTION_INSTRUCTION	
▸ Axis	AXIS_CIP_DRIVE	

ST 示例：

当 MAW 输入条件为真时，控制器准备 Axis 上的位置监视事件检查。

```
MAW(
    Axis,           //轴
    MotionControl,  //指令名称
    Forward,        //触发条件
    Position);      //位置
```

撤销运动轴注册 MDR 指令

MDR 指令撤销指定轴的指定运动模块注册输入事件检查。

1）指令格式

指令	名称	梯形图表现	ST 表现
MDR	撤销运动 轴注册		MDR(Axis, //轴 MotionControl, //指令变量 1); //输入编号

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Input Number	输入信号通道	UINT32	立即数	指定要选择的记录输入。 1 = 记录 1 位置 2 = 记录 2 位置

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴监视事件检查已经成功禁止后，完成位被置 1。。
.ER （错误）位 28	这个位被置 1 表示指令检测到错误，例如您指定了一个未配置的轴。

3）功能说明

MDR 指令用于取消由先前的运动记录指令（MAR）建立的记录事件检查。只有与指定记录输入相关的检查会被禁用。如果目标轴没有出现在可用轴的列表中，说明该轴尚未被配置为可操作。可以使用变量编辑器（Tag Editor）来创建和配置新的轴。

执行行为：

情况	行为
Prescan	.EN、.DN、.ER 和 .IP 位被置为 False。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。
执行条件为 TRUE	.EN 置为 True，指令执行
Postscan	不适用

MDR 对状态位的更改：

位名称	状态	含义
RegEventArmedStatus	FALSE	轴没有在寻找记录事件

RegEventStatus	FALSE	之前的记录事件已清除。

4）注意事项

要成功执行 MDR 指令，目标轴必须被配置为伺服轴（Servo axis）或仅反馈轴（Feedback Only axis）。否则，指令将出错。

该指令的执行可能需要多个扫描周期来完成，因为它需要进行多次粗略更新才能完成请求。完成位（.DN）不会立即置 1，只有在请求完成后才会被置 1。

5）错误说明

扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MDR 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码 (十进制)	扩展错误代码 （十进制）	说明
SERVO_MESSAGE_FAILURE (12)	无资源（3）	提供的记录输入超出了范围。

参数超出范围(13)错误的扩展错误代码工作方式有点不同。不是采用标准的枚举，扩展错误代码所显示的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MDR 指令，扩展错误代码 2 指的是输入编号操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

6）示例

梯形图：

当 MDR 输入条件为真时，控制器禁止 Axis 上的记录事件检查

▶ MotionControl	MOTION_INSTRUCTION	
▶ Axis	AXIS_CIP_DRIVE	

ST 示例：

当 MDR 输入条件为真时，控制器禁止 Axis 上的记录事件检查

```
MDR(
    Axis,              //轴
    MotionControl,  //指令变量
    1);                //输入编号
```

撤销运动轴监控 MDW 指令

MDW 指令撤销指定轴的监控位置事件检查。

1）指令格式

指令	名称	梯形图表现	ST 表现
MDW	撤销运动轴监控		MDW(Axis, //轴 MotionControl); //指令变量

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
MotionControl	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴监视事件检查已经成功禁止后，完成位被置 1。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。

3）功能说明

MDW 指令用于取消由先前的开启运动轴监视（MAW）指令设置的监视位置事件检查。解除监视位置事件检查需要指定目标轴，该指令会置 0 轴数据结构中的监视事件状态位和监视已准备状态位。执行此指令还会置 0 与控制运动轴监视（MAW）指令相关的“IP”位。

执行行为：

情况	行为
Prescan	.EN、.DN、.ER 和 .IP 位被置为 False。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。
执行条件为 TRUE	.EN 置为 True，指令执行
Postscan	不适用

MDW 对状态位的更改：

位名称	状态	含义
WatchEventArmedStatus	FALSE	轴没有在寻找监视位置事件。
WatchEventStatus	FALSE	之前的监视事件已清除。

4）注意事项

如果目标轴没有出现在可用轴的列表中，说明该轴尚未被配置为可操作。可以使用变量编辑器（Tag Editor）来创建和配置新的轴。

要成功执行 MDW 指令，目标轴必须被配置为伺服轴（Servo axis）或仅反馈轴（Feedback Only axis）。否则，指令将出错

该指令的执行可能需要多个扫描周期来完成，因为它需要进行多次粗略更新才能完成请求。完成位（.DN）不会立即置 1，只有在请求完成后才会被置 1。

5）错误说明

扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MDW 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

相关联的错误代码 (十进制)	扩展错误代码 （十进制）	说明
SERVO_MESSAGE_FAILURE (12)	无资源（3）	提供的记录输入超出了范围。

参数超出范围(13)错误的扩展错误代码工作方式有点不同。不是采用标准的枚举，扩展错误代码所显示的编号指的是操作数的编号，在面板中从上到下列出，第一个操作数被计为零。因此对于 MDW 指令，扩展错误代码 2 指的是输入编号操作数的值。然后必须将您的值与指令可以接受的值范围进行对比。

6）实例

梯形图：

当 MDW 输入条件为真时，控制器禁止 Axis 上的位置监视事件检查:

▶ MotionControl	MOTION_INSTRUCTION	
▶ Axis	AXIS_CIP_DRIVE	

ST 示例：

当 MDW 输入条件为真时，控制器禁止 Axis 上的位置监视事件检查:

```
MDW(
    Axis,              //轴
    MotionControl);    //指令变量
```

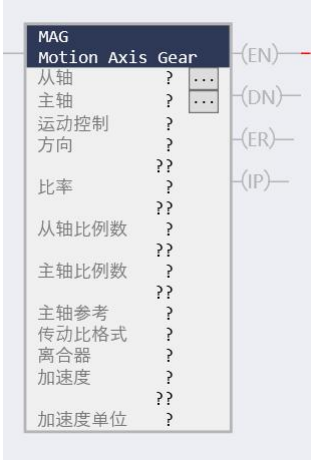
十、运动位移指令

轴电子齿轮 MAG 指令

MAG 指令让从轴以指定方向和指定比率跟随主轴运动。

1）指令格式

指令	名称	梯形图表现	ST 表现

MAG	轴电子齿轮		MAG(SlaveAxis, //主轴
			MasterAxis, //从轴
			miMAG3, //运动控制
			0, //方向
			MAGRatio1, //比率
			1, //从轴比例数
			1, //主轴比例数
			Command, //主轴参考
			real, //传动比格式
			Disabled, //离合器
			0, //加速度
			Unitspersec2); //加速度单位

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Slave axis	从轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。
Master axis	主轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	从轴跟随的轴。
Motion	指令变量	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

Control				
Direction	方向	DINT	立即数 变量	从轴跟踪主轴的相对方向。 选择下列值之一： 选择下列值之一： 0 = 从轴在主轴的方向上移动 1 = 从轴在与其当前方向相反
Ratio	比率	REAL	立即数 变量	有符号实数值，按照从用户单位/主用户单位的形式建立传动比。
Slave counts	从轴比例数	DINT	立即数 变量	整数值，表示用于指定分数传动比的从计数。
Master counts	主轴比例数	DINT	立即数 变量	整数值，表示用于指定分数传动比的主计数。
Master reference	主轴参考	BOOL	立即数	将主位置参考设置为命令位置或实际位置。 0 = 实际 – 根据主轴的编码器或其他反馈装置测定的主轴当前位置产生从轴运动。 1 = 命令 – 从主轴所需位置或命令位置产生从轴运动。
Ratio	传动比格式	BOOL	立即数	所需比率规范格式。二选一：

format				0 = 实数传动比 1 = 从编码器计数与主编码器计数之比的整数部分
Clutch	离合器	BOOL	立即数	如果启用了离合，则运动控制使从轴以指令的定义加速度值缓变到传动速度。如果没有启用，则从轴立即锁定到主轴。 如果主轴当前正在移动，这种 情况会导致出现从轴突然不 受控制地加速的事件，该事件可能导致轴出现故障。 二选一： 0 = 启用 1 = 禁用
Accel rate	加速度	REAL	立即数 变量	从轴的加速度，单位为百分比或加速度单位。它在启用离合功能时应用。
Accel units	加速度单位	DINT	立即数	显示加速度值所用的单位。 二选一： 0 = 单位/秒 ^{2} 1 = 最大加速度的百分比

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴传动成功启动时置位。
.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如指定了未配置的轴。
.IP （正在处理）	此位在正梯级转换时置位，在由另一个运动传动轴命令取代，或由运动停止命令、合并、 关闭或伺服故障终止时清除。

3）功能说明

运动轴传动（MAG）指令实现两个轴之间按指定比率进行的电子传动。电子传动使任何物理轴都可以按照精确比率与另一个物理轴的实际或 命令位置保持同步。它提供两个轴之间的直接边到边锁定。

选择或输入所需主轴、从轴和方向，并输入所需比率的值或标记变量。如果某个轴在从轴弹出菜单中变暗（灰色）或不显示，则该物理轴没有定义用于伺服操作。

如果目标轴没有显示在可用轴列表中，则表示该轴尚未配置用于伺服 操作。使用标记编辑器创建和配置一个新轴。

在从轴的摇动或移动过程的任何后续执行过程中，电子传动保持为活动状态。这样，电子传动中即可叠加摇动或移动轨迹，从而创建复杂的运动和同步。

参考实际位置：

如果输入或选择 Actual Position （实际位置）作为主参考源，则从主轴的实际位置产生从轴运动。实际位置是轴编码器测定的物理轴当前位置。如果主轴的轴类型配置为

Feedback Only （仅反馈）时，这是唯一有效的选择。

参考命令位置：

如果输入或选择 Command Position（命令位置）作为主参考源，则从主轴的命令位置产生从轴运动。命令位置（仅主轴的轴类型配置为 Servo（伺服）时有效）是当前所需或命令的主轴位置。

由于命令位置不包括任何相关联的跟踪误差、外部位置干扰或量化噪声，因此它是更为准确和稳定的传动参考。传动到主轴的命令位置时，要在从轴上产生任何运动，都必须命令主轴移动。

同向传动：

如果选择或输入 Same（同向）作为方向时，如果主轴在其正方向上移动，则从轴也以指定传动比在其正方向上移动，反之亦然。

反向传动：

如果选择或输入 Opposite（反向）作为方向时，如果主轴在其正方向上移动，则从轴以指定传动比在其负 方向上移动，反之亦然。

更改传动比：

如果选择或输入 Unchanged（不变）作为方向，则在保持当前传动方向（同向或反向）期间可以更改传动比。这在当前方向未知或不重要时十分有用。

反转传动方向：

如果选择或输入 Reverse（反转）作为方向，则电子传动的当前方向由同向更改为反向，或由反向更改为同向。这对于传动比必须在绞盘的每一端反转的卷动应用十分有用。

实数传动比：

如果选择或输入 Real（实数）作为比率格式，则将传动比指定为值 在 0.00001 至 9.99999 （含 0.00001 和 9.99999）之间的实数或标记变量，表示所需的从轴位置单位与主轴位置单位之比。由于使用此方式表示的传动比是以轴的配置位置单位进行定义的，因此易于理解。

分数传动比：

如果选择或输入 Fraction（分数）作为比率格式，则将传动比指定为一对整数或标记变量，表示从轴反馈计数的数目与主轴反馈计数的数目之比。从计数最多可以使用五位数 (99999)，主计数最多可以使用九位数 (999999999)。

如果传动比无法正确表示为小数点右侧最多五位的实数，则使用 Fraction （分数）作为比率格式。

如果将传动比指定为分数,则可以在无累计定位误差或舍入的情况下实现直接无理数传动比（如 1/3）。由于主计数和从计数值为整数，并且不使用轴转换常数，因此从轴和主轴之间的实际传动比关系与指定比率完全一致。

例如，无理数传动比 1/3 可以等效地指定为从计数 1 比主计数 3，从计数 10 比主计数 30，从计数 3 比主计数 9，等等。

离合：

如果选中 Clutch（离合）复选框，则从轴加速或减速到特定速度，达到该速度时，如果从轴当前由所选主轴传动，则使用梯形速度轨迹 （线性加速或减速）以指定传动比和方向，从轴即会移动。一旦从轴达到传动速度，即根据其他选择自动激活电子传动。按照当前配置的最大加速度值的百分比或直接以配置的加速度用户单位为单位输入所需的加速度。

使用离合功能可避免不受控制的加速或减速，这种情况在主轴移动时启用电子传动就会发生。离合功能还可以用于动态合并传动比变化，甚至是方向的变化。运动控制器自动按照新比率和/或方向将从轴缓变到主轴设置的速度。

离合缓变生成器的操作对从轴上可能进行的摇动或移动过程没有任何影响。

更改主轴：

任何时候，甚至在当前启用了传动时，都可以更改电子传动的主轴。但是，由于可能同时在多个轴上启用电子传动，因此在伺服主轴和从轴反转时，这些轴会变为交错耦合，可能产生意外的运动。

例如，如果要由轴 1（定义为伺服轴）传动轴 0，然后希望进行更改，由轴 0 传动轴 1，则必须首先禁用轴 0 的传动。这是因为在轴 0 为主轴的情况下将轴 1 指定为从轴，不会自动将轴 1 用作主轴，同时禁用轴 0 作为从轴。

传动时移动：

增量 MAM 指令可以在启用电子传动时用于从轴（或轴类型配置为伺服时用于主轴）。这对于完成加速/减速控制尤其有用。增量移动距离可以用于消除主轴和从轴之间的任何相位误差，也可以用于创建正确的非零相位关系。增量 MAM 指令还可以与电子传动结合使

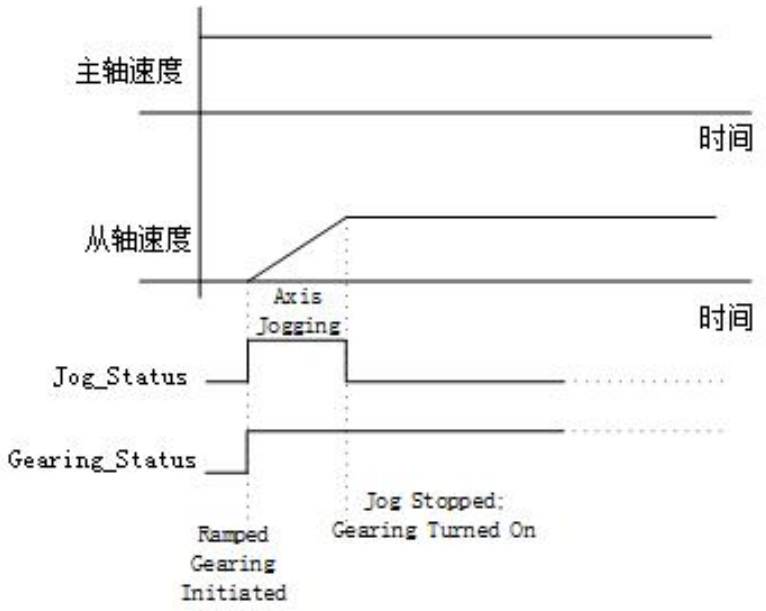
用以补偿材料误差。

要成功执行运动轴传动指令，目标轴必须配置为伺服轴类型，并且必须处于伺服打开状态。如果这些条件中有任何条件没有满足，则指令会出错。

重要事项：

MAG 指令在一次扫描中完成执行，完成后立即设置完成（.DN）位和正在处理（.IP）位，并清除过程完成（.PC）位。正在处理（.IP）位保持为置位状态，直至启动的传动过程由另一个 MAG 指令取代，或由运动轴停止命令、合并操作或伺服故障操作终止为止。

离合功能的工作方式与汽车离合器非常相似，使从轴可以平稳地与主轴啮合，如下所示。



4）注意事项

配置轴时所设置的最大速度、加速度或减速度限制不适用于运动轴电子齿轮。

5）错误说明

MAG 错误代码（.ERR)

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。
Shutdown State Error （关闭状态错误）	7	试图对处于关闭状态的轴执行指令。
Illegal Axis Type （非法轴类型）	8	试图对没有配置用于伺服或虚拟的枚举轴类型执行指令
Overtravel Error （超行程错误）	9	试图以超行程情况的方式执行指令。
Master Axis Conflict （主轴冲突）	10	传递了与从轴参考相同的主轴参考。
Axis Not Configured （轴未配置）	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。
Parameter Out Of Range （参数超出范围）	13	试图使用超出合法范围限制的参数执行指令。
Home in Process （归位正在进行）	16	试图在进行归位时执行指令
Axis Group Not Synchronized （轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Axis in Faulted State （轴处于故障状态）	20	试图对处于故障状态的轴执行指令。

Illegal Controller Mode Operation （非法控制器模式操作）	24	试图在处理器处于测试模式时执行指令
Illegal Axis Data Type （非法轴数据类型）	38	试图对指令不支持的轴数据类型执行指令。
最大减速度值为零	54	轴的减速度设置为零。这是一个非法的减速度值，该值禁止启动运动。

6）示例

梯形图：

ST 示例：

```
MAG(SlaveAxis,          //主轴
    MasterAxis,         //从轴
    miMAG3,             //运动控制
    0,                  //方向
```

MAGRatio1, //比率

1, //从轴比例数

1, //主轴比例数

Command, //主轴参考

real, //传动比格式

Disabled, //离合器

0, //加速度

Unitspersec2); //加速度单位

轴回零指令 MAH 指令

MAH 指令让轴可以以主动、被动等方式寻找轴位置参考点。

1）指令格式

指令	名称	梯形图表现	ST 表现

MAH	轴回零指令		MAH(refAxis, //轴
			miMRP); //运动控制

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。
Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位,在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴成功完成归零或中止归零时置位。
.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如指定了未配置的轴。

.IP （正在处理）	此位在正梯级转换时置位，在运动轴停止完成、或由关闭命令或伺服故障终止之后清除。
.PC （过程完成）	此位在轴归零成功完成时置位。

3）功能说明

运动轴归零（MAH）指令用于校准指定轴的绝对位置。对于配置为伺 服类型的轴，可以使用 Active（主动）、Passive（被动）或 Absolute Homing （绝对归零）模式配置对轴进行归零。对于 Feedback Only （仅反馈）轴，只能使用被动和绝对归零模式。绝对归零模式要求轴装有绝对反馈装置。

主动归零：

如果轴配置为主动归零模式，则首先激活物理轴以进行伺服操作。作为此过程的一部分，会取消正在执行的所有其他运动，并清除相应的状态位。然后，使用配置的归零序列对轴进行归零，归零序列可以为 Immediate （立即）、Switch（开关）、Marker（标志）或 Switch-Marker （开关-标志）。后三种归零序列会导致轴在配置的归零方向上摇动，根据归零事件的检测重新定义位置后，轴再自动移动到配置的归零位置。

被动归零：

如果轴配置为被动归零模式，则 MAH 指令重新定义物理轴在编码器标志下次出现时的实际位置。被动归零最常用于使仅反馈轴与其标志校准，但也可用于伺服轴。对于编码器标志，被动归零与主动归零相同（运动控制器不命令任何轴运动除外）。

启动被动归零后，轴必须移动越过归零序列的编码器标志才能正确完成。对于闭合回路的

伺服轴，这可以使用 MAM 或 MAJ 指令来 完成。对于物理仅反馈轴，运动不能由运动控制器直接命令，而必须通过其他方式才能完成。

绝对归零：

如果运动轴硬件支持绝对反馈装置，则可以使用绝对归零模式。绝对归零模式的唯一有效归零序列是立即类型的。在这种情况下，通过将配置的归零位置应用于绝对反馈装置的报告位置，绝对归零过程建立轴的实际绝对位置。在通过 MAH 指令执行绝对归零过程之前，轴必须处于 Axis Ready（轴就绪）状态，同时禁用伺服回路。

若要在配置使用主动归零模式的轴上成功执行 MAH 指令，目标轴必须配置为伺服轴类型。若要成功执 0 行 MAH 指令，目标轴必须配置为伺服轴或仅反馈轴。如果这些条件中有任何条件没有满足，则指令会出错。

重要事项：

MAH 指令开始执行时，会设置正在处理（.IP）位并清除过程完成（.PC）位。MAH 指令执行可能需要多次扫描来执行，这是因为它需要向运动模块传输多个消息。因此，在成功传输这些消息之前不会设置完成（.DN）位。在设置完成（.DN）位的同时，会清除正在处理（.IP）位并设置过程完成（.PC）位。

4）注意事项

如果检测到归零事件时，旋转轴上在执行单向主动归零并且归零偏移值小于减速距离，则控制将轴移 动到值为零的放卷位置。这样可确保到归零位置的移动是单向的。

5）错误说明

MAH 错误代码（.ERR)

错误消息	代码	说明
Execution Collision（执行冲突）	3	试图在指令已有另一实例在轴上执行时执行。如果不验证完成（.DN）位即执行要求进行消息传递的指令，则可能发生这种情况。
Shutdown State Error（关闭状态错误）	7	试图对处于关闭状态的轴执行指令。
Illegal Axis Type（非法轴类型）	8	试图对没有配置用于伺服或虚拟的枚举轴类型执行指令
Axis Not Configured（轴未配置）	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。
Servo Message Failure（伺服消息失败）	12	到目标运动模块的消息传递失败。
Axis Type Unused（未用轴类型）	18	试图对未根据当前轴类型配置属性进行配置的轴执行指令。
Axis Group Not Synchronized（轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Axis in Faulted State（轴处于故障状态）	20	试图对处于故障状态的轴执行指令。
Group in Faulted State （组处于故障状态）	21	试图对处于故障状态的组中的轴执行指令。

Axis in Motion （轴在运动中）	22	试图对处于运动中的轴执行指令
Illegal Controller Mode Operation （非法控制器模式操作）	24	试图在处理器处于测试模式时执行指令
Registration in Progress（正在处理定位杆）	34	试图在处理定位杆期间执行指令。
Illegal Axis Data Type （非法轴数据类型）	38	试图对指令不支持的轴数据类型执行指令。
Drive Locally Disabled （驱动器在本地禁用）	40	试图在本地禁用驱动器时运行 MAH 命令。
Illegal Homing Configuration （非法归零配置）	41	归零配置是非法的。
最大减速度值为零	54	轴的减速度设置为零。这是一个非法的减速度值，该值禁止启动运动。

6）示例

梯形图：

ST 示例：

```
MAH(refAxis,    //轴
    miMRP);   //运动控制
```

轴点动指令 MAJ 指令

MAJ 指令让轴以指定的速度正向或反向运动。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAJ	轴点动指令		MAJ(ServoAxis, //轴 Motion_MAJ, //运动控制 Direction, //方向 Speed, //点动速度 Unitspersec, //速度单位 Accel, //加速度 Unitspersec2, //加速度单位 Decel, //减速度 Unitspersec2, //减速度单位 S-Curve, //运行曲线 Accel_Jerk, //加速跃度 Decel_Jerk, //减速跃度 Unitspersec3, //跃度单位 Disabled, //合并 0, //合并速度 0, //锁定位置

			0); //锁定方向

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。
Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。
Direction	方向	DINT	立即数变量	点动的方向，二选一： 0 = 向前点动 1 = 向后点动
Speed	速度	REAL	立即数变量	移动轴的速度，单位为百分比或速度单位。
Speed Units	速度单位	DINT	立即数	显示速度值所用的工程单位 二选一： 0 = 单位/ 秒 1 = 最大速度的百分比
Accel Rate	加速度	REAL	立即数变量	轴的加速度，单位为百分比或加速度单位
Accel Units	加速度单位	DINT	立即数	显示加速度值所用的工程单

				位。二选一： 0 = 单位/ 秒 ^{2} 1 = 最大加速度的百分比
Decel Rate	减速度	REAL	立即数 变量	轴的减速度，单位为百分比或减速度单位。
Decel Units	减速度单位	DINT	立即数	显示减速度值所用的工程单位。二选一： 0 = 单位/ 秒 ^{2} 1 = 最大加速度的百分比
Profile	曲线类型	DINT	立即数	选择速度轨迹以运行点动： 0 =梯形 1 = S 形
Accel Jerk	加速跃度	REAL	立即数 变量	必须给加速跃度跟减速跃度操作数输入值。此指令只会在配置为 S 曲线时使用这些值。 0 = 单位/ 秒 ^{3} 1 = 最大跃度的百分比 2 = 时间的百分比
Decel Jerk	减速跃度	REAL	立即数 变量	
Jerk Units	跃度单位	DINT	立即数	
Merge	合并	DINT	立即数	启用时，Merge 指示运动控制将所有当前轴运动（无论当前正在处理哪些运动控制指令）都转换为由此指令控制的

				纯点动。 二选一： 0 = 禁用 1 = 启用
Merge Speed	合并速度	DINT	立即数	启用 Merge 时，这可以确定速度是此指令的指定速度值还是当前轴速度。二选一： 0 =速度字段中的值 1 =当前轴速度
Lock Position	锁定位置	REAL	立即数 变量	当从轴开始运动后，从轴需要开始跟随主轴时，主轴的位置。
Lock Direction	锁定方向	DINT	立即数 变量	指定使用 Lock Position 的条件

MOTION_INSTRUCTION 结构

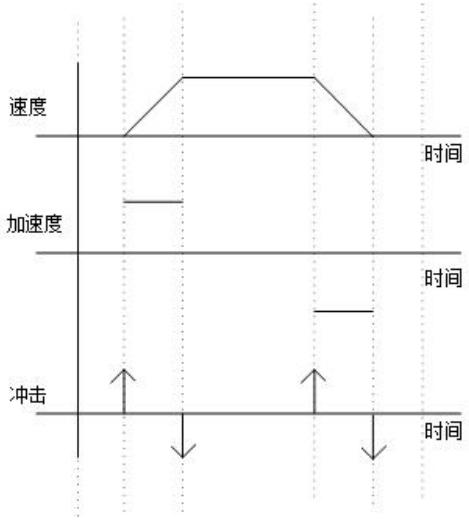
助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位,在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴点动成功启动时置位。
.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。
.IP （正在处理）	此位在正梯级转换时置位，并在点动过程由另一个运动轴点动命令取代，或由运动停止命令、合并、关闭 或伺服故障终止时清除。

3）功能说明

运动轴点动（MAJ）指令使物理轴在指定方向上以指定速度并使用指定加速度和减速度点动（连续移动）。要点动轴，请输入或选择所需的轴和方向，并输入所需速度、加速度和减速度的值或变量变量（以当前配置的最大值的百分比或直接以轴的配置速度和加速度单位表示）。 如果目标轴没有显示在可用轴列表中，则表示该轴尚未配置用于伺服操作。使用变量编辑器创建和配置一个新轴。

梯形：

梯形速度轨迹是最常用的轨迹，因为它在后续运动编程中提供最大的灵活性，并提供最快的加速和减速时间。速度的最大变化由加速度和减速度指定由于冲击不影响梯形轨迹，因此它被视为无限大，在下图中显示为垂直线。



S 形：

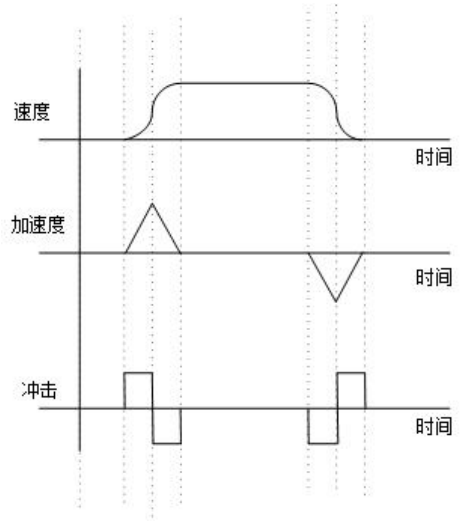
S 形速度轨迹最常用在需要最小化机械系统压力或负荷的情况下。但是，与梯形轨迹相比，S 形轨迹付出了加速和减速时间的代价。加速度或减速度的最大值由冲击进一步限制。

冲击率的计算方式如下：

加速冲击 = (最大加速度)^2 / 最大速度

减速冲击 = (最大减速度)^2 / 最大速度

轴加速度和减速度的计算是在启动轴移动、点动、更改动力或是移动或点动类型的停止时执行的。计算的冲击率生成三角形的加速度和减速度轨迹，如下图所示。



合并的点动：

点动还可以用于终止正在进行的其他运动控制指令产生的运动， 以便轴继续以指定速度移动。此操作的效果是将指定轴上正在进行的所有运动转变为纯点动。若要启动合并的点动，请将 Merge 设置为 Enabled 并定义合并的类型（注意，方向是自动确定的）。输入所需速度、加速度和减速度的值或变量变量。根据需要选择其他选项。

当前速度：

如果选择或输入 At Current Speed （当前速度）作为合并类型， 则点动的速度自动设置为轴的当前实际速度。在这种情况下，则忽略与 MAJ 指令相关联的所有指定速度值或变量变量。

程控速度：

如果选择 Programmed Speed 作为合并类型，则点动的速度设置为输入的速度值或变量变量。如果此速度不同于轴的当前速度，则轴根据指定的值加速或减速到新的速度。

若要成功执行运动轴点动指令，目标轴必须配置为伺服轴且该轴必须处于伺服打开状态。如果这些条件中有任何条件没有满足，则指令会出错。

重要事项：

MAJ 指令在一次扫描中完成执行，完成后立即设置完成（.DN）位和正在处理（.IP）位。正在处理（.IP）位保持为置位状态，直至启动的点动过程由另一个 MAJ 指令取代，或由运动轴停止命令、合并操作或伺服故障操作终止为止。

MAJ 对状态位的更改： 运动状态位

如果 Merge 设置为 Disabled，执行 MAJ 指令会将点动状态位设置为 True。

位名称	状态	含义
JogStatus	TRUE	轴正在点动

如果 Merge 为 Enabled，执行 MAJ 指令会将点动状态位设置为 True， 并清除移动和传动状态位。

位名称	状态	含义
JogStatus	TRUE	轴正在点动
MoveStatus	FALSE	轴不再移动
GearingStatus	FALSE	轴不再传动

4）注意事项

如果使用 S 形曲线，当轴沿 S 形曲线加速或减速时，如需变化加速度、减速度或速度，应格外小心。该操作可能会导致轴发生速度过冲或向相反方向移动。

这是因为跃度限制着 S 形曲线的加速和减速时间。当您减小加速度、减小减速度或增大速度时，也相应减小了跃度。跃度变小可能导致以下后果：

- 加速中的轴发生速度过冲
- 减速中的轴向相反方向移

5）错误说明

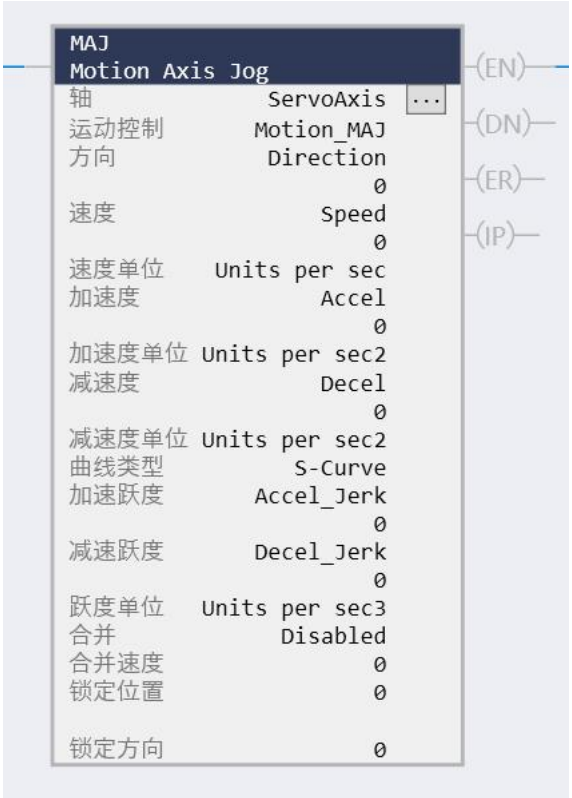
MAJ 错误代码（.ERR)

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。
Shutdown State Error （关闭状态错误）	7	试图对处于关闭状态的轴执行指令。
Illegal Axis Type （非法轴类型）	8	试图对没有配置用于伺服或虚拟的枚举轴类型执行指令
Overtravel Error （超行程错误）	9	试图以超行程情况的方式执行指令。
Axis Not Configured （轴未配置）	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。
Parameter Out Of Range （参数超出范围）	13	试图使用超出合法范围限制的参数执行指令。

Home in Process （归位正在进行）	16	试图在进行归位时执行指令
Axis Group Not Synchronized （轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Axis in Faulted State （轴处于故障状态）	20	试图对处于故障状态的轴执行指令。
Illegal Dynamic Change （非法动力更改）	23	试图进行非法动力更改，例如在 S 形轨 迹上进行合并，将轨迹由梯形动态更改为 S 形，将 S 形轨迹更改为非零速度， 或者更改 S 形轨迹的加速度。
Illegal Controller Mode Operation （非法控制器模式操作）	24	试图在处理器处于测试模式时执行指令
Illegal Axis Data Type （非法轴数据类型）	38	试图对指令不支持的轴数据类型执行指令。
最大减速度值为零	54	轴的减速度设置为零。这是一个非法的减速度值，该值禁止启动运动。

6）示例

梯形图：



ST 示例：

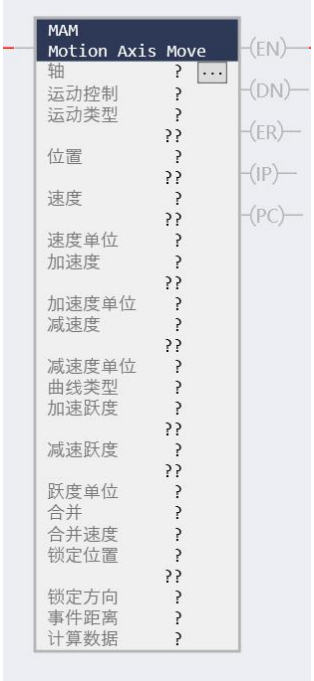
```
MAJ(
    ServoAxis,          //轴
    Motion_MAJ,         //运动控制
    Direction,          //方向
    Speed,              //点动速度
    Unitspersec,        //速度单位
    Accel,              //加速度
    Unitspersec2,       //加速度单位
    Decel,              //减速度
    Unitspersec2,       //减速度单位
```

```
S-Curve,                    //运行曲线
Accel_Jerk,                 //加速跃度
Decel_Jerk,                 //减速跃度
Unitspersec3,               //跃度单位
Disabled,                    //合并
0,                          //合并速度
0,                          //锁定位置
0);                         //锁定方向
```

轴位置指令 MAM 指令

MAM 指令让轴以指定的方式(相对、绝对、旋转等方式)移动到指定位置。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAM	轴位置指令		MAM(ServoAxis, //轴 Wrk_MAM01, //运动控制 diMoveType, //运动类型 MAMDistance,//移动距离 l_rMAMSpd, //移动速度 Unitspersec, //速度单位 Accel, //加速度 Unitspersec2, //加速度单位 Decel, //减速度 Unitspersec2, //减速度单位 S-Curve, //运动曲线 Accel_Jerk, //加加速 Decel_Jerk, //减减速 Unitspersec3, //Jerk 单位 Disabled, //合并

			Programmed, //合并速度 0, //锁定位置 None, //锁定方向 0, //事件距离 0); //计算数据

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。
Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

Move type	运动类型	DINT	立即数 变量	需要的移动操作类型。选择下列值之一： 0 = 绝对移动 1 = 增量移动 2 = 旋转最短路径移动 3 = 旋转正向移动 4 = 旋转负向移动 5 = 绝对主偏移 6 = 增量主偏移
Position	移动距离	REAL	立即数 变量	要移动到的绝对命令位置的值，对于增量移动，则为要从当前命令位置移动的距离的值。
Speed	速度	REAL	立即数 变量	移动轴的速度，单位为百分比或速度单位。
Speed Units	速度单位	DINT	立即数	显示速度值所用的工程单位 二选一： 0 = 单位/ 秒 1 = 最大速度的百分比
Accel Rate	加速度	REAL	立即数 变量	轴的加速度，单位为百分比或加速度单位

Accel Units	**加速度单位**	**DINT**	**立即数**	**显示加速度值所用的工程单位。二选一：** **0 = 单位/ 秒 ^{2}** **1 = 最大加速度的百分比**
Decel Rate	**减速度**	**REAL**	**立即数** **变量**	**轴的减速度，单位为百分比或减速度单位。**
Decel Units	**减速度单位**	**DINT**	**立即数**	**显示减速度值所用的工程单位。二选一：** **0 = 单位/ 秒 ^{2}** **1 = 最大加速度的百分比**
Profile	**曲线类型**	**DINT**	**立即数**	**选择速度轨迹以运行移动：** **0 =梯形** **1 = S 形**
Accel Jerk	**加速跃度**	**REAL**	**立即数** **变量**	**必须给加速跃度跟减速跃度操作数输入值。此指令只会在配置为 S 曲线时使用这些值。** **0 = 单位/ 秒 ^{3}** **1 = 最大跃度的百分比** **2 = 时间的百分比**
Decel Jerk	**减速跃度**	**REAL**	**立即数** **变量**	
Jerk Units	**跃度单位**	**DINT**	**立即数**	
Merge	**合并**	**DINT**	**立即数**	**启用时，Merge 指示运动控制将所有当前轴运动（无论当前正在处理哪些运动控制指令）**

				都转换为由此指令控制的纯移动。 二选一： 0 = 禁用 1 = 启用
Merge Speed	合并速度	DINT	立即数	启用 Merge 时，这可以确定速度是此指令的指定速度值还是当前轴速度。二选一： 0 =速度字段中的值 1 =当前轴速度
Lock Position	锁定位置	REAL	立即数 变量	当从轴开始运动后，从轴需要开始跟随主轴时，主轴的位置。
Lock Direction	锁定方向	DINT	立即数 变量	指定使用 Lock Position 的条件
Event Distance	事件距离	REAL OR 0	变量	从移动结束时测量的移动距离
Event Distance	计算数据	REAL OR 0	变量	从开始移动到最终位置所需的距离（或时间）

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴移动成功启动时置位。

.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如指定了未配置的轴。
.IP （正在处理）	此位在正梯级转换时置位，并在运动轴移动完成，或由另一个运动轴移动命令取代，或由运动停止命令、合并、关闭或伺服故障终止之后清除。
.PC （过程完成）	此位在移动完成（即到达目标）之后置位。

3）功能说明

运动轴移动（MAM）指令使用指定加速度和减速度以指定速度将物理轴移动到指定绝对位置或移动指定增量距离。除了这些绝对和增量移动之外，MAM 指令还可以产生许多其他特殊类型的移动。

若要移动轴，请输入或选择所需的物理轴和移动类型。输入所需位置或距离、速度、加速度和减速度的值或标记变量。可以用当前配置的最大值的百分比的形式或直接以配置的轴速度和加速度单位为单位输入速度、加速度和减速度。

如果目标轴没有显示在可用轴列表中，则表示该轴尚未配置用于伺服操作。使用变量编辑器创建和配置一个新轴。

绝对移动：

如果选择或输入 Absolute（绝对）作为移动类型，则轴以指定速度并使用指定加速度和减速度移动到指定位置。

更改绝对移动的终点：

通过使用另一个 MAM 指令以及新的所需绝对位置，可以在移动时更改绝对移动的终点。使用两种速度轨迹中的任意一种都可以更改终点，但在轴使用 S 形轨迹减速时不能更改终

点。在进行绝对移动期间，使用增量 MAM 指令也可以更改绝对移动的终点。这种情况下，轴的最终目标是初始绝对位置加上新的增量距离。任何情况下，轴都会平稳地移动到新的终点，而不会在旧的终点处停止。

通过指定新位置以及新速度，还可以同时更改正在进行的梯形绝对移动的移动速度与终点。如果指定了新的加速度和减速度值，则这些新值当且仅当轴反转方向时有效。

旋转轴上的绝对移动：

如果轴配置用于旋转操作，则绝对移动的处理方式与在线性轴上的处理方式相同（轴位置超过 Unwind 参数时放卷除外）。采用这种方法，轴位置永远不会大于 Unwind 值，也不会小于零。

指定位置用三角学方法进行说明，可以为正或为负，也可以大于 Unwind 值。负位置值等效于它们对应的正值（即，–90° 与 +270° 相同，等等），在轴旋转过零时十分有用。位置大于放卷值时，轴在停止于某个绝对位置之前通过多个旋转进行移动。

增量移动：

如果选择或输入 Incremental（增量）作为移动类型，则轴以指定速度并使用指定加速度和减速度移动指定距离。

更改移动距离：

使用另一个 MAM 指令以及另一个增量距离，可以在移动时（虽然不能在轴使用 S 形轨迹减速时）更改增量移动的最终目标。 MAM 指令移动的总距离为两个距离之和。

说明：在轴减速时，不能更改 S 形速度轨迹移动的最终目标。

使用另一个 MAM 指令以及新的绝对位置，可以在移动时更改 增量移动的最终目标。这种情况下，轴无需完成增量移动， 即直接移动到指定绝对位置。此操作与绝对 MAM 指令之后执行增量 MAM 指令的情况不同。

带传动的增量移动：

增量移动在从轴上进行时，可以使用运动轴传动（MAG）指令， 将传动运动叠加到移动轨迹上。反之，可以在启用传动后使用增量移动指令产生类似效果。这样可以完成许多复杂的轨迹和复杂的同步。在电子传动之上叠加增量移动对于实现加速/减速控制尤其有用。

旋转轴上的增量移动：

如果轴配置用于旋转操作，则增量移动的处理方式与在线性轴上的处理方式相同（轴位置超过 Unwind 参数时放卷除外）。 如果采用这种方法，则轴命令位置永远不会大于 Unwind 值， 也不会小于零。

指定距离用三角学方法进行说明，可以为正或为负，也可以大于 Unwind 值。当距离大于 Unwind 值时，轴在停止之前通过多个旋转进行移动。

旋转最短路径移动：

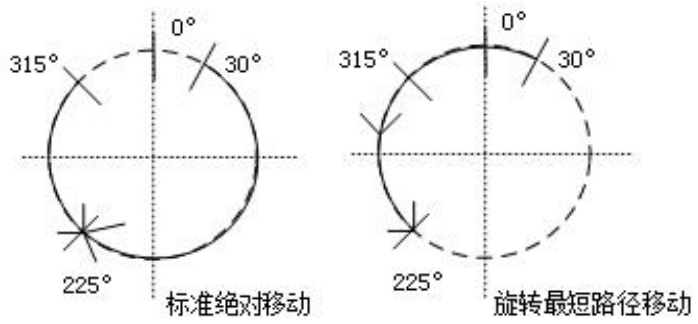
旋转最短路径移动通过最短路径将旋转轴移动到所需绝对位置。这是一种仅用于旋转轴的特殊绝对移动。

如果选择或输入 Rotary Shortest Path （旋转最短路径）作为移动类型，则无论轴的

当前位置如何，轴都以指定速度并使用指定加速度和减速度，在产生最短移动的方向上移动到指定位置。位置必须为正值且小于在运动控制器的机器设置菜单中输入的 Unwind 值，因此使用单个旋转最短路径 MAM 指令不能执行包含多个旋转的移动。

重要事项 只在配置为 Rotary（旋转）的轴上使用旋转最短路径移动。

例如，假定一个位置单位为度的旋转轴要移动到位置 225°。如下图所示，使用标准绝对 MAM 指令时，移动的方向取决于轴的当前位置，不一定是到终点的最短路径。小于终点的起始位置会产生正方向上的运动，而大于终点的起始位置会产生负方向上的运动。



但是，通过旋转最短路径移动，无论当前轴位置如何，轴都会在产生最短移动的方向上移动到指定终点（必要时过 0°）。旋转最短路径移动可以在轴移动或保持静止时使用。

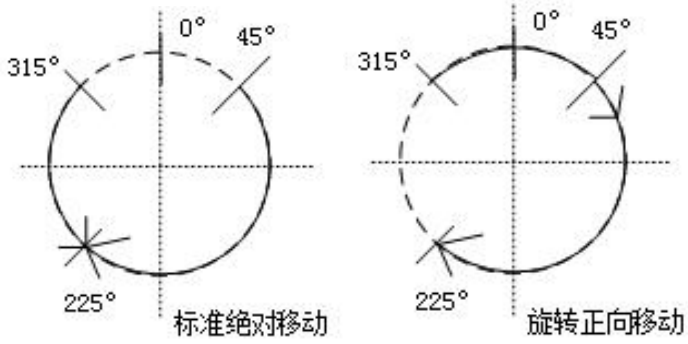
旋转正向移动：

旋转正向移动在正方向上将旋转轴移动到所需绝对位置。这是一种仅用于旋转轴的特殊绝对移动。

如果选择或输入 Rotary Positive （旋转正向）作为移动类型时，则无论轴的当前位置如何，轴都以指定速度并使用指定加速度和减速度在正方向上移动到指定位置。位置必须为

正值，使用单个正向旋转 MAM 指令不能执行包含多个旋转的移动。

例如，假定一个位置单位为度的旋转轴要移动到位置 225°。如下图所示，使用标准绝对 MAM 指令时，移动的方向取决于轴的当前位置，不一定是到终点的最短路径。小于终点的起始位置会产生正方向上的 运动，而大于终点的起始位置会产生负方向上的运动。



但是，通过旋转正向移动，无论当前轴位置如何，轴都会在正方向上移动到指定终点位置（必要时过 0°）。

重要事项 旋转正向移动应仅在轴未移动时使用，以确保运动方向正确。

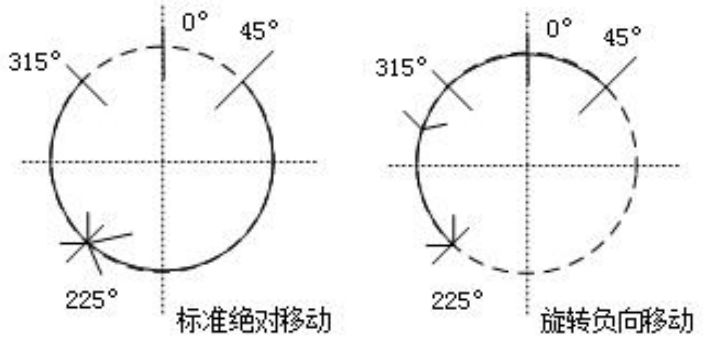
旋转负向移动

旋转负向移动在负方向上将旋转轴移动到所需绝对位置。这是一种仅用于旋转轴的特殊绝对移动。

如果选择或输入 Rotary Negative （旋转负向）作为移动类型，则无论轴的当前位置如何，轴都以指定速度并使用指定加速度和减速度在负方向上移动到指定位置。位置必须为正值，使用单个负向旋转 MAM 指令不能执行包含多个旋转的移动。

重要事项 只在配置为 Rotary（旋转）的轴上使用旋转最短路径移动。

例如，假定一个位置单位为度的旋转轴要移动到位置 225°。如下图所示，使用标准绝对 MAM 指令时，移动的方向取决于轴的当前位置，不一定是到终点的最短路径。小于终点的起始位置会产生正方向上的运动，而大于终点的起始位置会产生负方向上的运动。



但是，通过旋转负向移动，无论当前轴位置如何，轴都会在负方向上移动到指定终点位置（必要时过 0°）。

重要事项　旋转负向移动应仅在轴未移动时使用，以确保运动方向正确。

绝对和增量主偏移移动:

对于主偏移移动，指定的轴定义从轴。位置、加速度和减速度值表示主轴单位。速度单位、加速度单位和减速度单位表示主轴用户单位。最大值指的是主轴最大值。

在绝对主偏移移动的情况下，指定的位置值用作位置偏移目标。对于增量主偏移移动，指定的位置值与当前位置偏移目标相加。因此，当前位置偏移根据指定速度、加速度和减速度，移动到位置偏移目标。

移动轨迹类型:

MAM 指令选择的 Profile（轨迹）设置用于移动的运动轨迹的类型。

合并的移动:

MAM 指令还可以用于终止正在进行的其他运动控制指令产生的运动，以便轴继续以指定速度移动。此操作的效果是将指定轴上正在进行的所有运动转换为纯移动轨迹。若要启动合并的移动，请选中 Merge 复选框并定义合并的类型（注意，方向是自动确定的）。输入所需速度、加速度和减速度的值或标记变量。根据需要选择其他选项。

对于主偏移移动，新的移动被合并到一个活动主偏移移动中。因此，合并主偏移移动不会影响其他运动定制器对象。

当前速度：

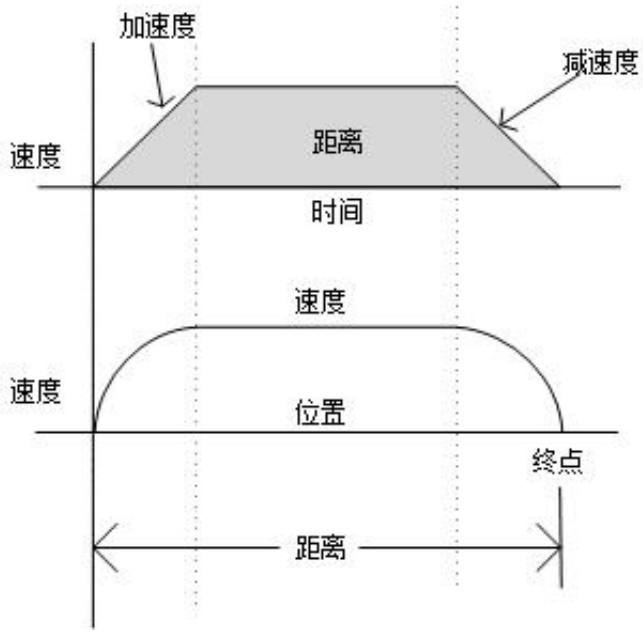
如果选择或输入 At Current Speed （当前速度）作为合并类型，则移动的速度自动设置为轴的当前实际速度。这种情况下，与 MAM 指令相关联的任何指定速度值或标记变量都会被忽略。

程控速度：

如果选择或输入 At Programmed Speed （程控速度）作为合并类型，则移动的速度设置为输入的速度值或标记变量。如果此速度不同于轴的当前速度，则轴根据指定的值加速或减速到新的速度。

若要成功执行运动轴移动指令，目标轴必须配置为伺服轴类型并且处于伺服打开状态。如果这些条件中有任何条件没有满足，则指令会出错。

下图介绍以静止的轴开始的梯形移动的一般形式。



4）注意事项

MAM 指令在一次扫描中完成执行，完成后立即设置完成（.DN）位和正在处理（.IP）位，并清除过程完成（.PC）位。正在处理（.IP）位保持为置位状态，直至启动的移动过程完成，或由另一个 MAM 指令取代， 或由运动轴停止命令、合并操作或伺服故障操作终止为止。仅当在以上所有其他事件终止移动过程并清除正在处理（.IP）位之前完成了启动的移动轨迹时，才设置过程完成（.PC）位。

如果使用 S 形曲线，当轴沿 S 形曲线加速或减速时，如需变化加速度、减速度或速度，应格外小心。该操作可能会导致轴发生速度过冲或向相反方向移动。

这是因为跃度限制着 S 形曲线的加速和减速时间。当您减小加速度、减小减速度或增大速度时，也相应减小了跃度。跃度变小可能导致以下后果：

- **加速中的轴发生速度过冲**
- **减速中的轴向相反方向移**

5）错误说明

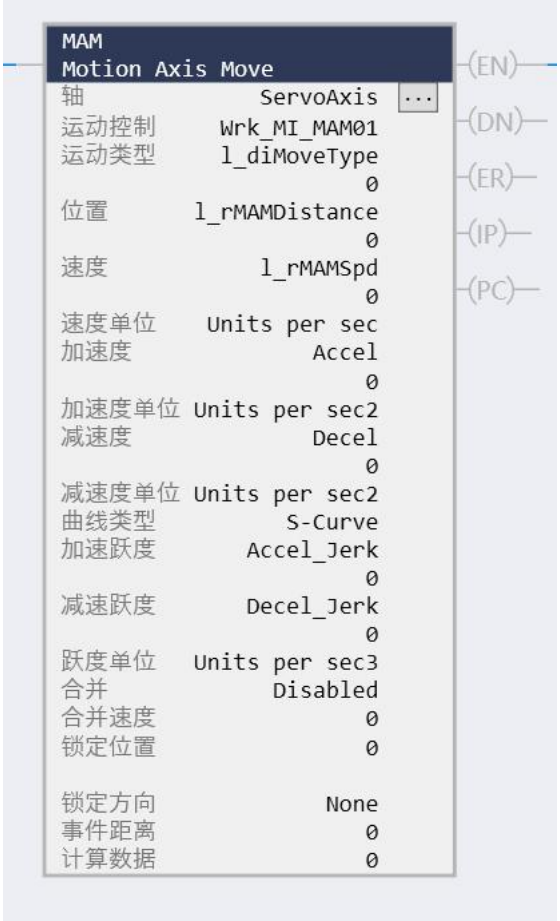
MAM 错误代码（.ERR)

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。
Shutdown State Error （关闭状态错误）	7	试图对处于关闭状态的轴执行指令。
Illegal Axis Type （非法轴类型）	8	试图对没有配置用于伺服或虚拟的枚举轴类型执行指令
Overtravel Error （超行程错误）	9	试图以超行程情况的方式执行指令。
Axis Not Configured （轴未配置）	11	传递的轴值引用了未配置的轴,即该轴没有指定到物理运动模块通道。
Parameter Out Of Range （参数超出范围）	13	试图使用超出合法范围限制的参数执行指令。
Home in Process （归位正在进行）	16	试图在进行归位时执行指令
Axis Group Not Synchronized （轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Axis in Faulted State （轴处于故障状态）	20	试图对处于故障状态的轴执行指令。

Illegal Dynamic Change （非法动力更改）	23	试图进行非法动力更改，例如在 S 形轨 迹上进行合并，将轨迹由梯形动态更改为 S 形，将 S 形轨迹更改为非零速度，或者更改 S 形轨迹的加速度。
Illegal Controller Mode Operation （非法控制器模式操作）	24	试图在处理器处于测试模式时执行指令
Position Cam Not Enabled （未启用位置凸轮）	33	试图在没有执行位置凸轮的情况下执行 MAH
Illegal Axis Data Type （非法轴数据类型）	38	试图对指令不支持的轴数据类型执行指令。
最大减速度值为零	54	轴的减速度设置为零。这是一个非法的减速度值，该值禁止启动运动。
所选轴超出最大系统行程限制 （位置溢出）	65	轴移动得太远，控制器无法存储该位置。位置范围取决于轴的转换常数。

6）示例

梯形图：



ST 示例：

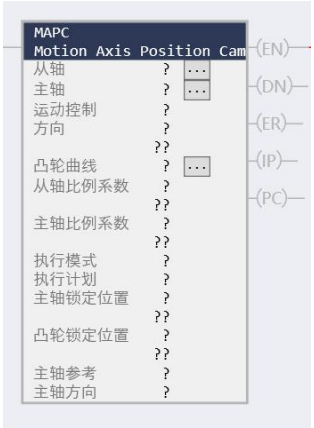
```
MAM( ServoAxis,                 //轴
     Wrk_MI_MAM01,              //运动控制
     l_diMoveType,              //运动类型
     l_rMAMDistance,            //移动距离
     l_rMAMSpd,                 //移动速度
     Unitspersec,               //速度单位
     Accel,                     //加速度
     Unitspersec2,              //加速度单位
     Decel,                     //减速度
```

```
Unitspersec2,                //减速度单位
SCurve,                      //运动曲线
Accel_Jerk,                  //加加速
Decel_Jerk,                  //减减速
Unitspersec3,                //Jerk 单位
Disabled,                    //合并
Programmed,                  //合并速度
0,                         //锁定位置
None,                        //锁定方向
0,                         //事件距离
0);                          //计算数据
```

轴位置凸轮 MAPC 指令

MAPC 指令让从轴以指定方式和指定的凸轮轮廓跟随主轴。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAPC	轴位置凸轮		MAPC(Slave_Axis, //从轴 Master_Axis, //主轴 MAPC, //运动控制 Direction, //方向 Cam_Profile, //凸轮曲线 Slave_Scaling, //从轴比例系数 Master_Scaling, //主轴比例系数 0, //执行模式 0, //执行计划 Master_Lock_Position,//主轴锁定位置 Cam_Lock_Position, //凸轮锁定位置 0, //主轴参考 0); //主轴方向

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明

Slave Axis	从轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	应用凸轮轨迹的轴的名称。
Master Axis	主轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	从轴根据凸轮轨迹跟随的轴。
Motion Control	运动控制	MOTION_ INSTRUCTION	变量	用于访问块状态参数的结构。
Direction	方向	DINT	立即数 变量	从轴相对于主轴的方向： • Same（同向）– 从轴位置值与主轴位置值具有相同的意义。 •Opposite（反向）– 从轴位置值与主轴位置值具有相反的意义。 或者，相对于当前或前凸轮运动方向的方向： •Reverse（反转）– 位置凸轮的当前或上一方向在执行时进行反转。如果选中 Reverse（反转），第一次执行时，控制将方向默认设置为 Opposite（反向）。 •Unchanged（不变）– 允许在不更改当前或前一凸轮运动方向的情况下更改其他凸轮参数。如果选择 Unchanged， 第一次

				执行时，控制将方向默认设置为 Same （同向）。
Cam Profile	凸轮曲线	CAM_PROFILE	变量	计算得出的凸轮轨迹数组的标记名称，用于建立主/从位置 关系。
Slave Scaling	从轴比例系数	REAL	立即数 变量	缩放从轴通过凸轮轨迹所经过的总距离。
Master Scaling	主轴比例系数	REAL	立即数 变量	缩放主轴通过凸轮轨迹所经过的总距离。
Execution Mode	执行模式	DINT	立即数	确定凸轮轨迹仅执行一次还是重复执行： 0 = Once（一次）– 仅当主轴 移动到由凸轮轨迹的起点和终 点定义的范围内时，从轴的凸轮运动才开始。如果主轴移动 到定义的范围之外，则从轴上的凸轮运动停止，过程完成位为置位状态。如果主轴移回凸 轮轨迹范围之内，则从运动不 会恢复。 1 = Continuous（连续）– 一旦开始，凸轮轨迹就无限期 执行。此功能在旋转应用中十 分有

				用，在旋转应用中，凸轮位置以旋转或往复方式连续运行。 2 = Persistent（持久）– 当主轴移动到定义范围之外时，从轴上的凸轮运动停止，PositionCamLockStatus 位为清除状态。如果主轴移回凸轮轨迹范围,PositionCamLockStatus 位为置位状态，则从运动不会恢复。
Execution Schedule	执行计划	DINT	立即数	选择执行凸轮轨迹的方法。选项为： 0 = 立即 – 从轴立即锁定到主轴， 并开始位置凸轮运动过程。 1 = 挂起 – 允许在正在进行的位置凸轮结束后混合新的位置凸轮执行。如果选择挂起，则 会忽略以下参数：Master Axis、Master Lock Position 和 Master Reference。 2 = 仅向前 – 当主位置在向前 方向上

				跨过主锁定位置时，凸轮轨迹开始。 3 = 仅反转 – 当主位置在反转方向上跨过主锁定位置时，凸轮轨迹开始。 4 = 双向 – 当主位置在两个方向中任一方向上跨过主锁定位置时，凸轮轨迹开始。
Master Lock Position	主轴锁定位置	REAL	立即数 变量	从轴锁定到主轴的主轴绝对 位置。如果选择挂起作为执行计划值，则会忽略主锁定位置。
Cam Lock Position	凸轮锁定位置	REAL	立即数 变量	确定凸轮轨迹的起始位置。
Master Reference	主轴参考	DINT	立即数	将主位置参考设置为命令位 置或实际位置。如果选择挂 起作为执行计划值，则会忽 略主参考。 0 = 实际 – 从轴运动从其编码器或其他反馈装置所测定的主轴当前位置发生。 1 = 命令 – 从轴运动从主轴的所需位置或命令位置发生。

Master Direction	主轴方向	DINT	立即数	确定根据凸轮轨迹产生从运动的主轴方向。 选项为： 0 = 双向 – 从轴可以在两个方向中任一方向上跟随主轴。 1 = 仅向前 – 从轴在主轴的向 前方向上跟随主轴。 2 = 仅反转 – 从轴在主轴的相反方向上跟随主轴。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴位置凸轮成功启动时置位。
.ER （错误）	此位在未成功计算出从轴值时置位。它在梯级由 false 转换为 true 时重置。
.IP （正在处理）	例如指定了未配置的轴。 .IP （正在处理）此位在正梯级转变时置位，并在由另一个运动 轴位置凸轮命令取代，或由运动停止命令、 合并、关闭或伺服故障终止时清除。
.PC （过程完成）	此位在正梯级转换时清除，在 once 执行模式下，主轴位置离开主位置范围（由当前活动 的凸轮轨迹定义）时置位。

3）功能说明

运动轴位置凸轮（MAPC）指令执行位置凸轮轨迹，该轨迹由前面的运动计算凸轮轨迹

(MCCP) 指令。 事实上，位置凸轮提供在两个轴之间实现非线性电子传动关系的功能。不使用任何最大速度、加速度或减速度限制。从轴的速度、加速度和减速度完全由主轴的运动以及从关联凸轮表获得的指定凸轮轨迹确定。

凸轮运动方向：

凸轮可以配置为将其增量部分加到从轴命令位置，或从该位置减去增量部分。对此行为的控制是通过 Direction（方向）参数实现的。

同向凸轮运动：

如果选择或输入 Same（同向）作为 MAPC 指令的方向，则将从凸轮轨迹计算得到的从轴位置值加到从轴的命令位置。这是最常见的操作，因为使用的轨迹位置值与原始凸轮表中的输入一样。即，连续增加轨迹值会导致轴在正方向上运动，反之亦然。

反向凸轮运动：

如果选择或输入 Opposite（反向）作为方向，则从从轴的命令位置中减去从凸轮轨迹计算得到的从轴位置值。这样，轴在与原始凸轮表设置的方向相反的方向上运动。即，连续增加轨迹值会导致 轴在负方向上运动，反之亦然。

保持当前凸轮运动方向：

如果选择或输入 Unchanged 作为方向，则可以更改其他位置凸轮参数， 同时保持当前或上一凸轮运动方向（同向或反向）。这在当前方向未 知或不重要时十分有用。如果选中 Unchanged，第一次执行凸轮运动时，控制会将方向默认设置为 Same。

反转当前凸轮运动方向：

如果选择 Reverse（反转），则位置凸轮的当前或上一方向由同向更改为反向，或从反向更改为同向。如果选中 Reverse，第一次执行凸轮时，控制会将方向默认设置为 Opposite。

线性和立方插补：

位置凸轮是完全插补的。这意味着，如果当前主轴位置不完全对应于 与凸轮轨迹相关联的凸轮表中的点，则从轴位置由相邻点之间的线性 或立方插补确定。这种方式提供了最平滑的从运动。

凸轮数组中用于生成凸轮轨迹的每个点都可以配置用于线性或立方插补。

在从轴的摇动或移动过程的任何后续执行过程中，电子凸轮运动保持为活动状态。这样，电子凸轮运动中即可叠加摇动或移动轨迹，从而创建复杂的运动和同步。

缩放位置凸轮：

在执行位置凸轮轨迹时，可以在主方向和从方向上对其进行缩放。要将存储的凸轮轨迹用于确定运动轨迹的一般形式，此缩放功能十分有用。然后，缩放参数可用于定义在其上执行轨迹的总主行程或总从行程。采用这种方法，一个标准凸轮轨迹可以用于生成整个特定凸轮轨迹系列。

默认情况下，Master Scaling 和 Slave Scaling 参数设置为 1。若要缩放位置凸轮轨迹，请输入 1 以外的 Master Scaling 或 Slave Scaling 值。

注意，增加凸轮轨迹的主缩放值会降低轨迹的速度和加速度，而增加从缩放值会提高轨迹

的速度和加速度。要维持经过缩放的轨迹的速度和加速度近似等于未经缩放的轨迹的速度和加速度，主缩放值和从缩放值应相等。例如，如果轨迹的从缩放值为 2，则主缩放值也应为 2， 以便在经过缩放的位置凸轮的执行过程中维持近似相等的速度和加速度。

凸轮轨迹执行模式：

选择 Once 或 Continuous 执行模式，可确定在主位置移动超出由初始凸轮表定义的轨迹的起点和终点时，凸轮运动的行为方式。

如果选择 Once（默认值），则仅当主轴移动到由凸轮轨迹的起点和终点定义的范围内时，从轴的凸轮运动才会开始。主轴移动到轨迹的 范围之外时，从轴上的凸轮运动停止，并设置 MAPC 指令的过程完 成位。注意，与当前 S 类情况相反，当主轴移回起点和终点所指定的 轨迹范围中时，从运动不会恢复。

如果选择 Continuous 模式，指定的凸轮轨迹一旦启动即会无限期执行。对于连续操作，当主轴的位置移动到轨迹范围之外时，轨迹的主位置和从位置被放卷，这会导致凸轮轨迹重复。此功能在旋转应用中 尤其有用，在这些旋转应用中，位置凸轮以旋转或往复方式连续运行。 但是，要使用这种技术生成平滑连续运动，必须小心设计凸轮表的凸轮点，确保在计算得出的凸轮轨迹的起点和终点之间不会出现位置、速度或加速度不连续的情况。

立即执行：

默认情况下，MAPC 指令计划为立即执行。在这种情况下，启用位置凸轮运动过程时没有延迟，并与 Master Lock Position 参数不相关。 从轴立即锁定到从特定凸轮轨迹的凸轮锁定位置开始的主轴上。

更改凸轮锁定位置：

MAPC 指令的 Cam Lock Position 参数确定从轴锁定到主轴时，在凸轮 轨迹中的起始位置。通常，Cam Lock Position 设置为凸轮轨迹的起点。由于大多数凸轮表的起点设置为 0，因此 Cam Lock Position 通常设置为 0。或者，Cam Lock Position 也可以设置为凸轮轨迹主范围中的任何位置。如果指定了超出此范围的 Cam Lock Position，则 MAPC 指令会出错。

仅向前、仅反转或双向执行

如果指令的 Execution Schedule 参数设置为 Forward Only （仅向前）、 Reverse Only （仅反转）或 Bi-directional （双向），只有主轴满足指定条件时，从轴才会锁定到主轴。这种情况下，凸轮运动过程监视主轴以确定主轴何时在指定方向上通过指定的主锁定位置。在旋转轴配 置中，此锁定条件仍然有效，与旋转计数无关。

挂起凸轮执行：

MAPC 指令的执行也可以是当前正在执行的位置凸轮的延期挂起完成。因此，选择 Pending 执行计划可以在不停止运动的情况下无缝混合两个位置凸轮轨迹。

Pending 执行功能在一些应用中尤其有用，如从轴必须锁定到正在移动的主轴和使用特定轨迹加速到适当速度时的高速打包。此加速度轨迹完成时，它必须平稳地混合到正在操作的轨迹（该轨迹通常是连续执行的）中。为了停止从轴，正在操作的轨迹平稳地混合到减速轨 迹中，以便轴在已知位置停止。

通过在当前轨迹仍在执行时将位置凸轮轨迹作为挂起的凸轮轨迹执行，提前设置适当的凸轮轨迹参数。这样可使当前轨迹无缝转换为挂起的轨迹；维持主轴与从轴之间的同步。但是，

要确保转换过程中平稳运动，在设计轨迹时，当前轨迹的终点和新轨迹的起点之间不能出现位置、速度或加速度不连续的情况。

一旦执行了挂起位置凸轮指令，则主轴经过当前轨迹的起点或终点时， 新的凸轮轨迹自动生效（并成为当前轨迹）。如果当前凸轮配置为执行一次，则在通过当前凸轮轨迹之后启动新的轨迹，并设置当前活动 的 MAPC 指令的 PC 位。如果当前凸轮配置为连续执行，则在本次通过当前凸轮轨迹之后启动新的轨迹，并设置当前活动的 MAPC 指令 的 IP 位。运动控制器在发生变化时跟踪主轴和从轴相对于第一个轨迹的位置，并使用此信息保持轨迹之间的同步。

如果 MAPC 指令的 Execution Schedule 设置为 Immediate，并且当前正在运行一个位置凸轮轨迹，则 MAPC 指令会出错。即使轴正在等待锁定到主轴也是如此。

如果在没有运行相应位置凸轮轨迹时选择 Pending 作为执行计划，则 MAPC 指令会执行，但在另一个具有非挂起执行计划的 MAPC 指令启动之前，不会发生任何凸轮运动。这样，在执行初始凸轮之前可以 预载挂起的凸轮轨迹。这种方法解决了立即凸轮在可靠加载挂起的凸轮之前完成这一问题。

配置挂起的位置凸轮之后，指定从轴的运动状态字的位置凸轮挂起状态位设置为 1 (true)。当挂起（新）的轨迹启动并成为当前轨迹时， 位置凸轮挂起状态位立即清除。

主参考：

Master Reference （主参考）参数确定主位置源以链接到凸轮生成器。 此源可以是主轴的实际位置或命令位置。从命令位置获得的运动更平稳。

主方向：

通常，Master Direction （主方向）参数设置为 Bi-directional（默认值）。 但是，如果选择 Forward Only 作为主方向，则从轴在主轴的向前方向上跟踪主轴。如果选择 Reverse only，则从轴在主轴的反向方向上跟踪 主轴。如果主轴更改方向，则从轴不反转方向，而是停留在主轴 反向时它所处的位置。位置凸轮的单向功能用于提供电子滑动离合，可以防止凸轮运动生成器在主轴反转方向时沿凸轮轨迹向后移动。

主轴再次反转，恢复所需方向上的运动时，从轴在主轴到达其最初反向的位置时再次加速。这样，从轴维持与主轴的同步，同时禁止错误方向上的运动。某个方向上的运动可能对机器或产品造成物理损坏的情况下，这特别有用。

凸轮运动时移动：

在凸轮运动时可以执行运动轴移动指令，以便在从轴运行时提供复杂的相位和偏移控制。

增量移动：

增量运动轴移动（MAM）指令可以在操作位置凸轮时用于从轴（或配置用于伺服操作的主轴）。这对于完成加速/减速控制尤其有用。 增量移动距离可以用于消除主轴和从轴之间的任何相位误差，也可以用于创建精确的相位关系。

主偏移移动：

在操作位置凸轮对凸轮的主参考位置进行动态移位时，也可以使用 MAM 指令。与从轴上的增量移动不同，从轴上的主偏移移动使凸轮轨迹相对于主轴进行移位。

停止凸轮：

与其他运动生成器（摇动、移动、传动等）一样，活动凸轮必须通过各种停止指令（MAS 或 MGS）停止。具体而言，MAS 指令必须能够明确停止凸轮运动过程。此行为应当与用于明确停止传动过程的 MAS 功能完全相同。

从凸轮进行合并：

与其他运动生成器（摇动、移动、传动等）一样，活动凸轮必须与运动合并功能兼容。具体而言，移动和摇动必须能够从活动凸轮运动合并。此行为应当与应用于传动过程的合并功能完全相同。

故障恢复：

有时，需要在不失去主轴与从轴（通过凸轮关系锁定）之间同步的情况下响应轴故障条件。通过活动凸轮，有两种方法处理轴故障。

创建一个虚拟轴并向该轴进行所有凸轮运动，如有必要，由机器的实际主轴传动此虚拟主轴。将所有轴的各种故障操作都设置为 Status Only （仅状态）。发生轴故障（如驱动器故障）时，监视轴故障状态 的应用程序会检测到该故障，并通过停止虚拟主轴对所有活动轴进行 受控停止。在轨迹生成器级别，所有一切仍然是完全同步的。使用故障轴的跟踪误差来确定轴偏离位置的距离。在故障轴上重置故障， 使用 MAM 指令和计算得到的跟踪误差使该轴以受控速度进行定位。 最后，开始移动虚拟主轴。

配置与上面相同，但这种方法中，当从轴出现故障时，轴故障操作会禁用驱动器。当然，这会终止从轴上的活动凸轮运动。此时，应用程 序应通过虚拟主轴停止所有其他轴。然后，对故障轴进行重新定位，方法是确定主轴的位置，然后计算从轴在未发生故障的情况下应处的位置。最后，通过将凸轮锁定位置设置为计算得出的值，执行直接锁定 MAPC 进行重新

同步。

4）注意事项

配置轴时所设置的最大速度、加速度或减速度限制不适用于电子凸轮运动。

如果在 MAPC 指令执行之后，还未满足锁定条件之前，重定义主轴的位置参考（如 MRP 指令）， 则凸轮轨迹生成器根据在重定义位置操作之前生效的绝对位置参考系统来监视主轴。

减小位置凸轮的主缩放值或增大其从缩放值会提高轨迹所需的速度和加速度。如果超出驱动器系统的允许范围，这可能会导致运动故障。

5）错误说明

MAPC 错误代码（.ERR)

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。
Shutdown State Error （关闭状态错误）	7	试图对处于关闭状态的轴执行指令。
Illegal Axis Type （非法轴类型）	8	试图对未配置为伺服轴的轴执行指令。
Overtravel Error （超行程错误）	9	试图以恶化当前超行程情况的方式执行指令。
Master Axis Conflict （主轴冲突）	10	传递了与从轴参考相同的主轴参考。
Axis Not Configured	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运

（轴未配置）		动模块通道。
Value Out of Range （值超出范围）	13	试图使用超出范围的输入参数执行指令。
Homing in Process Error （归位正在处理错误）	16	试图在归位过程进行时执行指令。
Axis Group Not Synchronized （轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Axis in Faulted State （轴处于故障状态）	20	试图对处于故障状态的轴执行指令。
Cam Profile Not Calculated （未计算凸轮轨迹）	32	试图执行尚未计算的凸轮轨迹段。
Illegal Axis Data Type （非法轴数据类型）	38	试图对指令不支持的轴数据类型执行指令。
最大减速度值为零	54	轴的减速度设置为零。这是一个非法的减速度值，该值禁 止启动运动。

6）示例

梯形图：

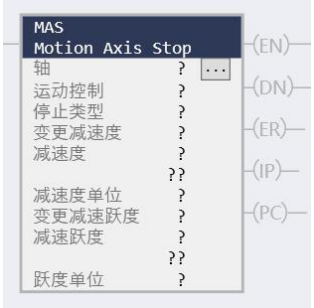
ST 示例：

```
MAPC(
    Slave_Axis,              //从轴
    Master_Axis,             //主轴
    MAPC,                    //运动控制
    Direction,               //方向
    Cam_Profile,             //凸轮曲线
    Slave_Scaling,           //从轴比例系数
    Master_Scaling,          //主轴比例系数
    0,                       //执行模式
    0,                       //执行计划
    Master_Lock_Position,    //主轴锁定位置
    Cam_Lock_Position,       //凸轮锁定位置
    0,                       //主轴参考
    0);                      //主轴方向
```

轴停止指令 MAS 指令

MAS 指令让轴停止指定类型(全部、点动、位移等方式)的运动。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAS	轴停止指令		MAS(ServoAxis, //轴 l_miMAS_MAJ, //运动控制 Jog, //停止类型 Yes, //是否改变减速度 Decel, //减速度 Unitspersec2, //减速度单位 Yes, //是否改变跃度 Accel_Jerk, //跃度值 Unitspersec3);//跃度单位

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。

Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。
Stop Type	停止类型	DINT	立即数	确定在指定轴上要停止的运 动过程。选项为： 0 = 停止所有运动 1 = 停止点动 2 = 停止移动 3 = 停止传动 4 = 停止归位 5 = 停止调整 6 = 停止测试 7 = 停止位置凸轮运动
Change Decel	是否改变减速度	BOOL	立即数	如果为置位状态，则启用指令的 Decel 值，而不使用当前配置的最大减速度。 二选一： 0 = 否 1 = 是
Decel Rate	减速度	REAL	立即数 变量	轴的减速度，单位为百分比或减速度单位。 如果在进行移动时降低

				减速度，则轴可能越过其目标位置。
Decel Units	减速度单位	DINT	立即数	显示 Decel 值所用的工程单位。二选一： 0 = 单位/秒 ^{2} 1 = 最大值的百分比
Change Decel Jerk	是否改变跃度	BOOL	立即数	如果为置位状态，则启用指令的 Decel Jerk 值，而不使用当前配置的最大值。 二选一： 0 = 否 1 = 是
Decel Jerk	跃度值	REAL	立即数 变量	注意：如果在移动中更改，轴可能会超过目标位置。 必须给减速跃度操作数输入值。此指令只会在配置为 S 曲线时使用这些值。 0 = 单位/ 秒 ^{3} 1 = 最大跃度的百分比 2 = 时间的百分比
Jerk Units	跃度单位	DINT	立即数	

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴停止成功启动时置位。
.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如指定了未配置的轴。
.IP （正在处理）	此位在正梯级转换时置位，在运动轴停止完成、或由关闭命令或伺服故障终止之后清除。
.PC （过程完成）	此位在停止操作成功完成后置位。

3）功能说明

运动轴停止（MAS）指令无需禁用伺服回路（如果为激活状态），即可在指定物理轴上停止任何运动。如果需要对任何正在执行的受控运动进行减速停止，则此指令十分有用。使用 MAS 指令可以分别或同时停止由点动、移动、传动、归位、时间凸轮、位置凸轮或主

偏移移动指令启动的运动。

在停止传动或位置凸轮运动过程时，选择从轴以关闭特定过程并停止该轴。如果主轴为伺服轴，则可以停止主轴，进而在不禁用传动或位置凸轮运动的情况下停止从轴。

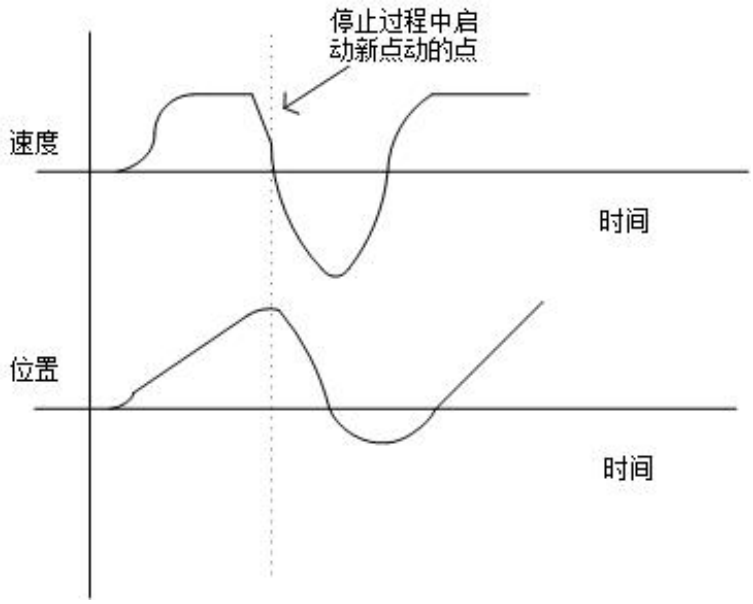
停止所有运动：

如果输入或选择 Stop All Motion （停止所有运动），正在执行的所有运动（即摇动、移动、传动、归位、测试、调整、时间 凸轮、位置凸轮或主偏移移动过程）都会同时停止，使轴处于受控停止状态。

重要事项：

MAS 指令在一次扫描中完成执行，完成后立即设置完成（.DN）位和正在处理（.IP）位，并清除过程完成（.PC）位。正在处理（.IP）位保持为置位状态，直至启动的停止过程完成，或由另一个 MAS 指令取代，或由伺服故障操作终止为止。仅在以上任何其他事件终止停止过程并清除正在处理（.IP）位之前完成了启动的离合轨迹时，才设置过程完成（.PC）位。

下图介绍在运动轴停止过程中启动运动轴点动可能发生的情况。点动的减速度显著小于正在执行的停止的当前减速度。新的减速冲击率变得更小。减速到零所需的时间会导致速度下冲，过零变为负数。轴运动会反转方向，直至速度重新为零。



4）注意事项

如果使用 S 形曲线，当轴沿 S 形曲线加速或减速时，如需变化加速度、减速度或速度，应格外小心。该操作可能会导致轴发生速度过冲或向相反方向移动。

这是因为跃度限制着 S 形曲线的加速和减速时间。当您减小加速度、减小减速度或增大速度时，也相应减小了跃度。跃度变小可能导致以下后果：

- **加速中的轴发生速度过冲**
- **减速中的轴向相反方向移**

5）错误说明

MAS 错误代码（.ERR)

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。

Shutdown State Error（关闭状态错误）	**7**	**试图对处于关闭状态的轴执行指令。**
Illegal Axis Type（非法轴类型）	**8**	**试图对没有配置用于伺服或虚拟的枚举轴类型执行指令**
Axis Not Configured（轴未配置）	**11**	**传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。**
Parameter Out Of Range （参数超出范围）	**13**	**试图使用超出合法范围限制的参数执行指令。**
Axis Group Not Synchronized（轴组未同步）	**19**	**试图对当前与关联轴组不同步的轴执行指令。**
Illegal Axis Data Type（非法轴数据类型）	**38**	**试图对指令不支持的轴数据类型执行指令。**

6）示例

梯形图：

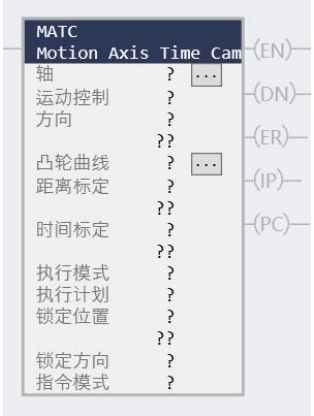
ST 示例：

```
MAS( ServoAxis,          //轴
    l_miMAS_MAJ,         //运动控制
    Jog,                 //停止类型
    Yes,                 //是否改变减速度
    Decel,               //减速度
    Unitspersec2,        //减速度单位
    Yes,                 //是否改变跃度
    Accel_Jerk,          //跃度值
    Unitspersec3);       //跃度单位
```

轴时间凸轮 MATC 指令

MATC 指令让轴以指定方式和指定的凸轮轮廓跟随时间。

1）指令格式

指令	名称	梯形图表现	ST 表现
MATC	轴时间凸轮		MATC(AXIS, //轴 MATC, //运动控制 Direction, //方向 Cam_Profile, //凸轮曲线 Distance_Scaling, //距离标定 Time_Scaling, //时间标定 0, //执行模式 0, //执行计划 Lock_Position, //锁定位置 0, //锁定方向 0); //指令模式

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明

Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	应用凸轮轨迹的轴的名称。
Motion Control	运动控制	MOTION_ INSTRUCTION	变量	用于访问块状态参数的结构。
Direction	方向	DINT	立即数 变量	从轴相对于主轴的方向： •Same（同向）– 从轴位置值与主轴位置值具有相同的意义。 •Opposite（反向）– 从轴位置值与主轴位置值具有相反的意义。 或者，相对于当前或前凸轮运动方向的方向： •Reverse（反转）– 位置凸轮的当前或上一方向在执行时进行反转。如果选中 Reverse （反转），第一次执行时，控制将方向默认设置为 Opposite（反向）。 •Unchanged（不变）– 允许在不更改当前或前一凸

				轮运动方向的情况下更改其他凸轮参数。如果选择 Unchanged， 第一次执行时，控制将方向默认设置为 Same （同向）。
Cam Profile	凸轮曲线	CAM_PROFILE	变量	计算得出的凸轮轨迹数组的标记名称。
Distance Scaling	距离标定	REAL	立即数 变量	缩放轴通过凸轮轨迹所经过的总距离。
Time Scaling	时间标定	REAL	立即数 变量	缩放凸轮轨迹的时间段。
Execution Mode	执行模式	DINT	立即数	确定凸轮运动在时间移动超过凸轮轨迹的终点时的行为方式。选项为： 0 = Once – 当时间凸轮执行时间超过凸轮轨迹的时间范围时，MATC 指令完成，轴运动停止，时间凸轮状态位清除。 1 = Continuous – 凸轮轨迹运动无限期执行。

Execution Schedule	执行计划	DINT	立即数	选择执行凸轮轨迹的方法。 选项为： 0 = Immediate – 指令计划为 立即执行，不延迟启用时间 凸轮运动过程。 1 = Pending – 延迟执行时间凸轮，直至当前或下一立即执行的时间凸轮完成为止。 将新的时间凸轮轨迹与正在进行的过程进行混合以实现无缝转换时，这十分有用。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴时间凸轮成功启动时置位。
.ER （错误）	错误位指示指令检测到错误，如轴未配置。
.IP （正在处理）	正在处理位在正梯级转换时置位，并在由停止命令、 合并、关闭或伺服故障终止时清除。
.PC （过程完成）	过程完成位在正梯级转换时清除，在 Once 执行 模式下，时间离开由当前活动的凸轮轨迹定义的时间范围时置位。

3）功能说明

运动轴时间凸轮（MATC）指令执行时间凸轮轨迹，该轨迹由前一运动计算凸轮轨迹(MCCP) 指令建立。使用时间凸轮可以实现提供的内置梯形和 S 形运动轨迹之外的复杂运动 轨迹。此指令中不使用任何最大速度、加速度和减速度限制。从轴的速度、加速度和减速度完全由从关联凸轮表得到的指定凸轮轨迹确定。

凸轮运动方向：

凸轮可以配置为将其增量部分加到轴命令位置，或从该位置减去增量部分。对此行为的控制是通过 Direction 参数实现的。

同向凸轮运动：

如果选择或输入 Same 作为 MATC 指令的方向，则将从凸轮轨迹计算得到的轴位置值加到轴的命令位置。这是最常见的操作，因为使用的轨迹位置值与原始凸轮表中的输入一样。即，连续增加轨迹值会导致轴在正方向上运动，反之亦然。

反向凸轮运动：

如果选择或输入 Opposite 作为方向，则从轴的命令位置中减去从凸轮轨迹计算得到的轴位置值。这样，轴在与原始凸轮表中设置的方向相反的方向上运动。即，连续增加轨迹值会导致轴在负方向上运动，反之亦然。

更改凸轮轨迹：

如果选择或输入 Unchanged 作为方向，则可以在保持当前或上一凸轮运动方向（同向或反向）时更改其他时间凸轮参数。这在当前方向未知或不重要时十分有用。如果选择 Unchanged，第一次执行凸轮运动时，控制会将方向默认设置为 Same。

更改凸轮运动方向：

如果选择 Reverse，则时间凸轮的当前或上一方向由 Same 更改为 Opposite，或由 Opposite 更改为 Same。如果选择 Reverse，第一次执 行凸轮时，控制会将方向默认设置为 Opposite。

指定凸轮轨迹：

要执行 MATC 指令，必须指定计算得到的凸轮轨迹数据数组标记。使用内置凸轮轨迹编辑器的 MATC 指令，可以创建凸轮轨迹数组标记，通过对现有凸轮数组执行运动计算凸轮轨迹 (MCCP) 指令，也可以创建这些标记。

可以使用凸轮轨迹编辑器在编译时修改凸轮轨迹数组中的数据，也可以使用运动计算凸轮轨迹 (MCCP) 指令在运行时进行修改。如果在运行时修改，必须创建凸轮数组才能使用 MCCP 指令。

可以使用凸轮轨迹编辑器在编译时修改凸轮轨迹数组中的数据，也可以使用运动计算凸轮轨迹 (MCCP) 指令在运行时进行修改。如果在运行时修改，必须创建凸轮数组才能使用 MCCP 指令。

凸轮轨迹数组结构元素的所有元素（状态和类型元素除外）变量编辑器中都是隐藏状态。这些隐藏元素不具有任何值。状态参数用于指示已经计算凸轮轨迹数组元素。如果试图使用凸轮轨迹中的任何未计算元素执行凸轮运动控制指令，则指令将出错。类型参数确定此凸轮

数组元素与下一凸轮数组元素之间所应用的插补类型。

凸轮轨迹数组检查：

凸轮轨迹数组中的第一个元素的 Status（状态）成员比较特殊，用于进行数据完整性检查。因此，MATC 必须始终指定起始索引设置为 0 的凸轮轨迹。

在指定轴上启动凸轮之前，MAPC 指令通过检查第一个凸轮轨迹元素的 Status 成员的值来检查是否已计算凸轮轨迹数组。如果 Status 为 0 或 1，则仍未计算凸轮轨迹，MATC 指令会出错。如果凸轮轨迹数组已经完成计算（Status > 1），则指令递增 Status 成员，指示此轴正在使用该指令。

凸轮完成或终止时，会递减第一个凸轮轨迹数组元素的 Status 成员，以便对使用关联凸轮轨迹的凸轮的数目进行主动跟踪。

线性和立方插补：

时间凸轮是完全插补的。这意味着，如果当前主时间值不完全对应于与凸轮轨迹相关联的凸轮表中的点，则从轴位置由相邻点之间的线性 或立方插补确定。这种方式提供了最平滑的从运动。

凸轮数组中用于生成凸轮轨迹的每个点都可以配置用于线性或立方插补。

在从轴的摇动或移动的任何后续执行过程中，电子凸轮运动保持为活动状态。这样，电子凸轮运动中即可叠加摇动或移动轨迹，从而创建复杂的运动和同步。

缩放时间凸轮：

在执行时间凸轮轨迹时，可以在时间和距离上对其进行缩放。如果要通过缩放（用于定

义执行轨迹所涉及的时间和距离）将存储的轨迹仅 用于运动的形式，则这种缩放十分有用。

如果凸轮轨迹数组由 MATC 指令指定，则凸轮轨迹数组定义的主坐标值采用时间单位（秒），从轴值采用从轴的单位。相反，Time Scaling 和 Distance Scaling 参数是 无单位值，仅用作凸轮轨迹的乘数。

默认情况下，Time Scaling 和 Distance Scaling 参数设置为 1。若要缩放时间凸轮轨迹，请输入不为 1 的 Time Scaling 或 Distance Scaling 值。

增加凸轮轨迹的 Time Scaling 值会降低轨迹的速度和加速度，增加 Distance Scaling 值会提高轨迹的速度和加速度。若要使缩放的轨迹的速度和加速度保持为近似等于未缩放的轨迹的速度和加 速度，Time Scaling 和 Distance Scaling 值应相等。例如，如果轨迹的 Distance Scaling 值为 2，则 Time Scaling 值也应为 2，以便在缩放的时间凸轮的执行过程中维持近似相等的速度和加速度。

凸轮轨迹执行模式：

选择 Once 或 Continuous 执行模式，可确定在时间移动超出由初始凸轮表定义的轨迹的终点时，凸轮运动的行为方式。

如果选择 Once（默认值），则立即启动轴的凸轮轨迹运动。如果时间凸轮执行时间超出凸轮轨迹定义的时间范围，则 MATC 指令完成，轴运动停止，从轴的运动状态字中的时间凸轮状态位也会清除。

如果选择 Continuous 模式，则指定的凸轮轨迹会立即启动并无限期执行。对于连续操作，当时间移动超出凸轮轨迹的终点时，时间被放卷到凸轮轨迹的起点，使凸轮轨迹无限重复。此功能在旋转应 用中尤其有用，在这些旋转应用中，时间凸轮以旋转或往复方式连续 运行。但是，要使用这种技术生成平滑的连续运动，必须小心设计凸轮表的凸轮点，确保在计

算得出的凸轮轨迹的起点和终点之间不会出现位置、速度或加速度不连续的情况。

执行计划

MATC 指令的执行计划是通过 Execution Schedule 参数控制的。

立即执行：

默认情况下，MATC 指令计划为立即执行，这是因为 Execution Schedule 参数的默认设置为 Immediate。在这种情况下，启用时间凸轮运动过程没有任何延迟。

挂起凸轮执行：

或者，MATC 指令的执行实际上也可以是当前正在执行的时间凸轮的延期挂起完成。因此，选择 Pending 执行计划可以在不停止运动的情况下无缝混合两个时间凸轮轨迹。

在轴必须加速到使用特定速度轨迹的速度的应用中，Pending 执行功能尤其有用。此加速度轨迹完成时，它必须平稳地混合到凸轮轨迹中，其中凸轮轨迹通常是连续执行的。为了停止轴，正在操作的轨迹可以平稳地混合到减速轨迹中，以便轴在已知位置停止。

通过在当前轨迹仍在执行时将时间凸轮轨迹作为挂起凸轮轨迹执行，可以提前设置适当的凸轮轨迹参数。这使当前轨迹可以无缝转换为挂起的轨迹 – 保持主时间与从轴位置之间的同步。但是，要确保转换过程中平稳运动，在设计轨迹时，当前轨迹的终点和新轨迹的起点之间 不能出现位置、速度或加速度不连续的情况。

一旦执行挂起时间凸轮指令，则凸轮时间经过当前轨迹的终点时，新的凸轮轨迹即自动生效（并成为当前轨迹）。如果当前凸轮配置为执行一次，则在通过当前凸轮轨迹之后启动新的轨迹，并设置当前活动的 MATC 指令的 PC 位。如果当前凸轮配置为连续执行，则在本

次通过当前凸轮轨迹之后启动新的轨迹，并设置当前活动的 MATC 指令的 IP 位。运动控制器在发生变化时跟踪时间和轴相对于第一个轨迹的位置，并使用此信息保持轨迹之间的同步。

如果 MATC 指令的执行计划设置为 Immediate，而当前正在运行某个时间凸轮轨迹，则 MATC 指令会生成 Illegal Dynamic Change （非法动力更改）错误。

如果在没有运行相应时间凸轮轨迹的情况下选择 Pending 执行计划， 则 MATC 指令会执行，但在另一个具有非挂起执行计划的 MATC 指令启动之前，不会进行任何凸轮运动。这样，在执行初始凸轮之前可以预载挂起的凸轮轨迹。这种方法解决了立即凸轮在可靠加载挂起的凸轮之前完成这一问题。

配置挂起的时间凸轮之后，指定轴的运动状态字的时间凸轮挂起状态位设置为 1 (true)。挂起的（新的）轨迹启动并成为当前轨迹时，时间凸轮挂起状态位立即清除。

停止凸轮：

与其他运动生成器（摇动、移动、传动等）一样，活动凸轮必须通过各种停止指令（MAS 或 MGS）停止。更改 OS 模式时，凸轮运动也必须停止。具体而言，MAS 指令必须能够明确 停止凸轮运动过程。此行为应当与用于明确停止传动过程的 MAS 功能完全相同。

从凸轮进行合并：

与其他运动生成器（摇动、移动、传动等）一样，活动凸轮必须与运动合并功能兼容。具体而言，必须能够从活动凸轮运动合并移动和点动。此行为应当与应用于传动过程的合并功能完全相同。

4） 注意事项

MATC 指令执行在一次扫描中完成，完成后立即设 置完成（.DN）位和正在处理（.IP）位。正在处理（.IP）位保持为置位状态，直至启动的时间凸轮运动过程由另一个 MATC 指令取代，或由运动轴停止命令、 合并操作或伺服故障操作终止为止。

减小时间凸轮的 Time Scaling 值或增大其 Distance Scaling 值会提高轨迹的所需速度和加速度。如果超出了驱动器系统的允许范围，可能导致运动故障。

配置轴时所设置的最大速度、加速度或减速度限制不适用于电子凸轮运动。

5）错误说明

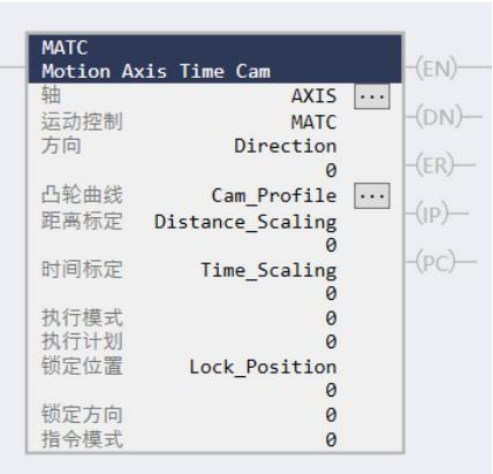
MATC 错误代码（.ERR）

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。
Shutdown State Error （关闭状态错误）	7	试图对处于关闭状态的轴执行指令。
Illegal Axis Type （非法轴类型）	8	试图对未配置为伺服轴的轴执行指令。
Overtravel Error （超行程错误）	9	试图以恶化当前超行程情况的方式执行指令。
Axis Not Configured （轴未配置）	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。
Value Out of Range （值超出范围）	13	试图使用超出范围的输入参数执行指令。
Homing in Process Error	16	试图在归位过程进行时执行指令。

（归位正在处理错误）		
Axis Group Not Synchronized （轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Axis in Faulted State （轴处于故障状态）	20	试图对处于故障状态的轴执行指令。
Illegal Dynamic Change （非法动力更改）	23	试图在其他凸轮轨迹执行时执行指令

6）示例

梯形图：



ST 示例：

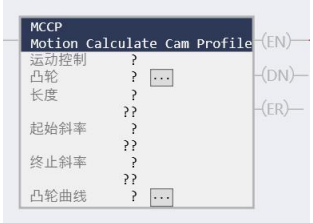
```
MATC(
    AXIS,          //轴
    MATC,          //运动控制
```

```
Direction,              //方向
Cam_Profile,            //凸轮曲线
Distance_Scaling, //距离标定
Time_Scaling,           //时间标定
0,                      //执行模式
0,                      //执行计划
Lock_Position,          //锁定位置
0,                      //锁定方向
0);                     //指令模式
```

轴计算凸轮轮廓 MCCP 指令

MCCP 指令基于凸轮点阵列计算凸轮轮廓。结果可用于 MAPC 和 MATC。

1）指令格式

指令	名称	梯形图表现	ST 表现
MCCP	轴计算凸轮轮廓		MCCP(MCCP, //运动控制 Cam, //凸轮 Length, //长度 Start_Slope, //起始斜率 End_Slope, //终止斜率 Came_Profile);//凸轮曲线

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Motion control	运动控制	MOTION_ INSTRUCTION	变量	用于访问块状态参数的结构。
Cam	凸轮	CAM	变量	用于计算凸轮轨迹的标记名称。 数值数组索引指示在凸轮轨迹计算中使用的数组中的起

				始凸轮元素。省略号启动凸轮轨迹编辑器。
Length	长度	DINT	立即数变量	确定在凸轮轨迹计算中使用的数组中的凸轮元素的数量。
Start Slope	起始斜率	REAL	立即数变量	这是轨迹的初始斜率的边界条件。它仅对立方的第一段有效，用于 指定通过第一个点的斜率。
End Slope	终止斜率	REAL	立即数变量	这是轨迹的结束斜率的边界条件。它仅对立方的最后一段有效，用于指定通过最后一个点的斜率。
Cam Profile	凸轮曲线	CAM_PROFILE	变量	计算得出的凸轮轨迹数组的标记名称，用作 MAPC 和 MATC 指令的输入。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	完成位在成功执行计算凸轮指令和凸轮轨迹数组时置位。
.ER （错误）	如果错误位为置位状态，则指示指令检测到错误，如凸轮数组具有非法长度。

3）功能说明

运动计算凸轮轨迹（MCCP）指令根据指定凸轮数组中的一组给定点计算凸轮轨迹。此指令生成的结果凸轮轨迹可以由后续 MAPC 或 MATC 凸轮运动控制指令用于提供相对于主轴位置或时间的复杂从轴运动。

指定凸轮数组：

为了执行 MCCP 指令，必须使用变量编辑器或凸轮轨迹编辑器创建凸轮数组标记。下图演示如何建立凸轮数组标记以及将其用作 MCCP 指令的输入： 凸轮数组元素由从轴（yp）和主轴（xp）点对以及插补类型组成。由于与 特定轴位置或时间不相关联，因此 x 和 y 点值无单位。每个点的插补类型都可以指定为线性或立方。

指定凸轮轨迹标记：

要执行 MAPC 指令，还必须创建凸轮轨迹数组标记。可以通过变量编辑器或使用内置凸轮轨迹编辑器的 MAPC/MATC 指令创建凸轮轨迹数组标记。

可以使用凸轮轨迹编辑器在编译时修改凸轮轨迹数组中的数据，也可以使用运动计算凸轮轨迹（MCCP）指令在运行时进行修改。如果在运行时修改，必须创建凸轮数组才能使用 MCCP 指令。

状态参数用于指示已经计算凸轮轨迹数组元素。如果试图使用凸轮轨迹中的任何未计算元素执行凸轮运动控制指令，则 MAPC 或 MATC 指令将出错。类型参数确定凸轮数组元素与下一凸轮数组元素之间所应用的插补类型。

凸轮轨迹数组状态成员：

凸轮轨迹数组中的第一个元素的 Status（状态）成员有特殊用途，用于进行数据完整性检查。因此，MCCP 必须始终指定起始索引设置为 0 的凸轮轨迹。

线性或立方样条插补：

计算获得的凸轮轨迹是完全插补的。这意味着，如果当前主位置或时间不完全对应于用于生成凸轮轨迹的凸轮数组中的点，则从轴位置由相邻点之间的线性或立方插补确定。这种方式提供了最平滑的从运动。 通过将系数计算为多项式方程（该方程按照主位置或时间的函数确定从位置），MCCP 指令可实现此操作。

计算凸轮轨迹：

在计算指定轴上的凸轮轨迹之前，MCCP 指令首先通过检查第一个凸轮轨迹元素的 Status 成员的值来检查凸轮轨迹数组是否已计算。如果 Status 值为 0 或 2，则 MCCP 继续计算凸轮轨迹。当已完全计算出凸轮轨迹数组时，MCCP 指令将第一个凸轮轨迹元素的 Status 值设置为 being calculated （正在计算）或 1，然后将所有其他凸轮轨迹元素的 Status 值设置为 being calculated （正在计算）。随着计算的进行， 各个凸轮轨迹成员的 Status 值都设置为 calculated（已计算） 或 2。当凸轮轨迹数组中的所有元素都已计算时，第一个凸轮轨迹元 素的 Status 值也设置为 calculated （已计算）。

但是，如果在初始凸轮轨迹 Status 值为 1 的情况下执行 MCCP 指令， 则当前正在由另一个 MCCP 指令计算该凸轮轨迹，MCCP 指令会出错。 如果 Status 值大于 2，则该凸轮轨迹正在由 MAPC 或 MATC 指令过 程使用，MCCP 指令会出错。

起始斜率和结束斜率：

为了便于平稳进入立方凸轮轨迹以及从其中平稳退出，提供了斜率控制。Start Slope（起始斜率）和 End Slope（结束斜率）参数确定从 轴相对于主轴的初始变化速度。这些值用于对凸轮数组执行的立方样条计算中。

如果开始或结束具有线性插补的凸轮轨迹，则起始斜率和结束斜率值不适用。

4）注意事项

MCCP 指令在一次扫描中完成执行。因此，此指令应放置在独立的任务中以免影响用户程序扫描时间。

5）错误说明

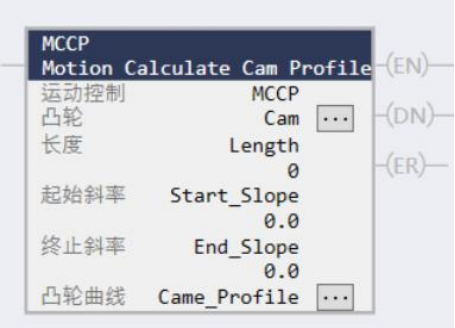
MCCP 错误代码（.ERR)

错误消息	代码	说明
Illegal Cam Length（非法凸轮长度）	26	试图使用非法凸轮数组长度执行指令。
Illegal Cam Profile Length （非法凸轮轨迹长度）	27	试图使用非法凸轮轨迹数组长度执行指令。
Illegal Cam Type（非法凸轮类型）	28	试图使用凸轮元素中的非法段类型执行指令。
Illegal Cam Order（非法凸轮顺序）	29	试图使用非法凸轮元素顺序执行指令。
Cam Profile Being Calculated（正在计算凸轮轨迹）	30	试图对当前正在计算的凸轮轨迹数组执行指令。

Cam Profile Being Used （正在使用凸轮轨迹）	31	试图对当前正在使用的凸轮轨迹数组执行指令。

6）示例

梯形图：



ST 示例：

```
MCCP(MCCP,              //运动控制
     Cam,               //凸轮
     Length,            //长度
     Start_Slope,       //起始斜率
     End_Slope,         //终止斜率
     Came_Profile);     //凸轮曲线
```

轴动态修改运动参数 MCD 指令

MCD 指令可以选择性地改变运动进程中的速度、加速度、jerk 等值。

1）指令格式

指令	名称	梯形图表现	ST 表现
MCD	轴动态修改运动参数		MCD(refAxis, //轴 miMCD, //指令变量 Move, //更改类型 Yes, //是否更改速度 OverrideVelocitySet,//新速度 Yes, //是否更改加速度 I_MCDAcc, //新加速度 Yes, //是否更改减速度 I_MCDDec, //新减速度 Yes, //是否更改加速跃度 I_MCDJerk, //新加速跃度 Yes, //是否更改减速跃度 I_MCDJerk, //新减速跃度 Unitspersec, //速度单位 Unitspersec2,//加速度单位 Unitspersec2,//减速度单位

			Unitspersec3);//跃度单位

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。
Motion Control	指令变量	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。
Motion Type	更改类型	DINT	立即数	要更改的运动轨迹（点动或移动）。二选一： 0 = 摇动 1 = 移动
Change Speed	是否更改速度	BOOL	立即数	设置以启用速度更改。二选一：0 = 否 1 = 是
Speed	新速度	REAL	立即数 变量	移动轴的新速度，单位为百分 比或速度单位。
Change Accel	是否更改加速度	DINT	立即数	设置来启用加速度更改。二选一： 0 = 否 1 = 是

Accel Rate	新加速度	REAL	立即数 变量	轴的加速度，单位为百分比或 加速度单位。
Change Decel	是否更改减速度	BOOL	立即数	设置以启用减速度更改。 二选一： 0 = 否 1 = 是
Decel Rate	新减速度	REAL	立即数 变量	轴的减速度，单位为百分比或 减速度单位。 如果在移动进行时降低减速度， 则轴可能会越过其目标位置。
Change Accel Jerk	是否更改加速跃度	DINT	立即数	设置以启用加速跃度更改。 二选一： 0 = 否 1 = 是
Accel Jerk	新加速跃度	REAL	立即数 变量	必须给减速跃度操作数输入值。此指令只会在配置为 S 曲线时使用这些值。
Change Decel Jerk	是否更改减速跃度.	DINT	立即数	设置以启用减速跃度更改。 二选一：

				0 = 否 1 = 是
Decel Jerk	新减速跃度	REAL	立即数 变量	必须给减速跃度操作数输入值。此指令只会在配置为 S 曲线时使用这些值。
Speed Units	速度单位	DINT	立即数	显示速度值所用的单位。 二选一： 0 = 单位/秒 1 = 最大速度的百分比
Accel Units	加速度单位	DINT	立即数	显示加速度值所用的单位。 二选一： 0 = 单位/秒 ^{2} 1 = 最大加速度的百分比
Decel Units	减速度单位	DINT	立即数	显示减速度值所用的单位。 二选一： 0 = 单位/秒 ^{2} 1 = 最大减速度的百分比
Jerk Units	跃度单位	DINT	立即数	0 = 单位/ 秒 ^{3} 1 = 最大跃度的百分比

				2 = 时间的百分比

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在轴更改动力成功启动时置位。
.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如指定了未配置的轴。

3）功能说明

MCD 指令动态更改梯形轨迹移动的速度，以及梯形轨迹点动的速度、 加速度和减速度。选择所需物理轴和运动类型，并输入速度、加速度和减速度的值或变量。可以按照当前最大配置值的百分比的形式或直接以配置的轴速度或加速度单位为单位输入速度、加速度和减速度。

如果目标轴没有显示在可用轴列表中，则表示该轴尚未配置用于伺服操作。使用变量编辑器创建和配置一个新轴。

更改移动动力：

如果输入或选择 Move 作为移动类型，正在进行的移动的速度、加速度和/或减速度可以更改为指定值。如果新速度高于当前速度，则速度以指定的加速度发生变化；如果新速度小于当前速度，则速度以指定的减速度发生变化。

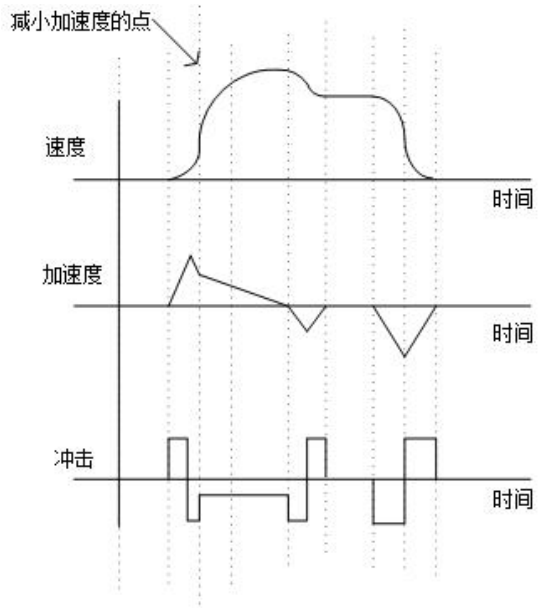
暂停移动：

MCD 指令可以用于临时暂停正在进行的移动，方法是将其速度更改为零。使用另一个 MCD 指令以及一个非零速度值可按照最初指定的方式完成移动。

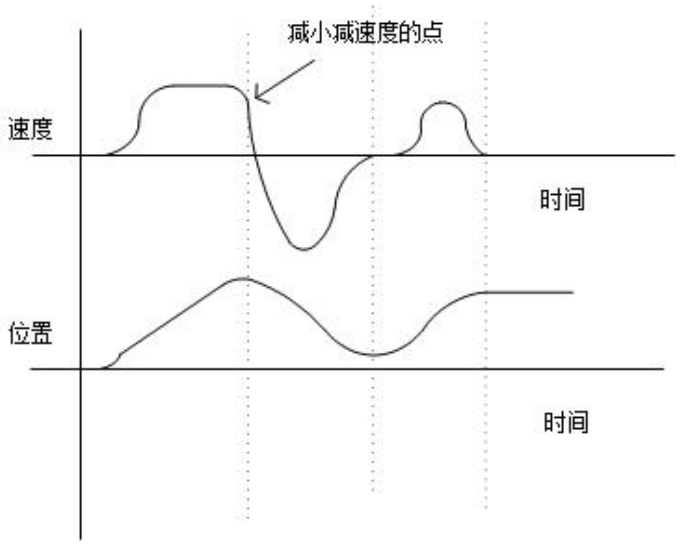
更改点动动力：

如果输入或选择 Jog 作为移动类型，正在进行的点动的速度、加速度或减速度可以更改为指定值。如果新速度高于当前速度，则速度以指定的加速度发生变化；如果新速度小于当前速度，则速度以指定的减 速度发生变化。

下图演示当 MCD 指令用于在速度接近最大值时降低加速度可能发生的情况。新的加速冲击率变得更小，进一步限制了最大加速变化。由于加速到零之前需要额外时间，发生速度下冲。产生另一个轨迹使速度回到程控最大值。



下图演示当 MCD 指令用于在速度和位置接近其目标终点时降低减速度可能发生的情况。新的减速冲击率变得更小。减速到零所需的时间会导致速度下冲，过零变为负数。轴运动也会反转方向，直至速度重新为零。产生另一个轨迹使位置回到程控目标。



4）注意事项

如果使用 S 形曲线，当轴沿 S 形曲线加速或减速时，如需变化加速度、减速度或速度，应格外小心。该操作可能会导致轴发生速度过冲或向相反方向移动。

这是因为跃度限制着 S 形曲线的加速和减速时间。当您减小加速度、减小减速度或增大速度时，也相应减小了跃度。跃度变小可能导致以下后果：

- **加速中的轴发生速度过冲**
- **减速中的轴向相反方向移**

5）错误说明

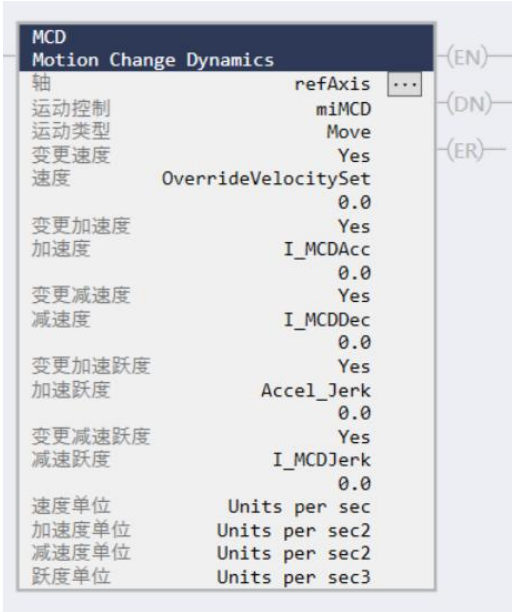
MCD 错误代码（.ERR)

错误消息	代码	说明
Servo Off State Error （伺服关闭状态错误）	5	试图对没有闭合伺服回路的轴执行指令。
Shutdown State Error （关闭状态错误）	7	试图对处于关闭状态的轴执行指令。

Illegal Axis Type （非法轴类型）	8	试图对没有配置用于伺服或虚拟的枚举轴类型执行指令。
Axis Not Configured （轴未配置）	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。
Parameter Out Of Range （参数超出范围）	13	试图使用超出合法范围限制的参数执行指令。
Home in Process （归位正在进行）	16	试图在进行归位操作时执行。
Axis Group Not Synchronized （轴组未同步）	19	试图对当前与关联轴组不同步的轴执行指令。
Illegal Dynamic Change （非法动力更改）	23	试图进行非法动力更改，例如在 S 形轨 迹上进行合并，将轨迹由梯形动态更改为 S 形，将 S 形轨迹更改为非零速度， 或者更改 S 形轨迹的加速度。
Illegal Axis Data Type （非法轴数据类型）	38	试图对指令不支持的轴数据类型执行指令。
最大减速度值为零	54	轴的减速度设置为零。这是一个非法的减速度值，该值禁止启动运动。

6）示例

梯形图：



ST 示例：

```
MCD(
    refAxis,              //轴
    miMCD,                //指令变量
    Move,                 //更改类型
    Yes,                  //是否更改速度
    OverrideVelocitySet, //新速度
    Yes,                  //是否更改加速度
    I_MCDAcc,             //新加速度
    Yes,                  //是否更改减速度
    I_MCDDec,             //新减速度
    Yes,                  //是否更改加速跃度
    I_MCDJerk,            //新加速跃度
    Yes,                  //是否更改减速跃度
    I_MCDJerk,            //新减速跃度
```

Unitspersec,　　//速度单位

Unitspersec2,　　//加速度单位

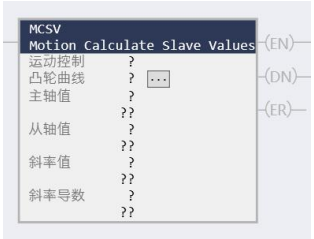
Unitspersec2,　　//减速度单位

Unitspersec3);　　//跃度单位

轴计算凸轮从轴值 MCSV 指令

MCSV 指令从给定凸轮轮廓和主值计算对应从值以及斜率值。

1）指令格式

指令	名称	梯形图表现	ST 表现
MCSV	轴计算凸轮从轴值		MCSV(MCSV, //运动控制 Cam_Profile, //凸轮曲线 Master_Value, //主轴值 Slave_Value, //从轴值 Slope_Value, //斜率值 Slope_Derivative); //斜率导数

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Motion Control	运动控制	MOTION_ INSTRUCTION	变量	用于访问指令状态参数的结构。
Cam Profile	运动凸轮曲线	CAM_PROFILE	变量	元素数组，其数组索引设 置为 0。它定义在计算从轴值时所用的凸轮轨迹。

Master Value	主轴值	REAL	立即数变量	在计算从轴值时所用的凸轮 轨迹在主轴上的精确值。
Slave Value	从轴值	REAL	变量	凸轮轨迹（主轴位于指定主轴值）在从轴上的值。
Slope Value	斜率值	REAL	变量	凸轮轨迹（主轴位于指定主轴值）在从轴上的值的一阶导数。
Slope Derivative	斜率导数	REAL	变量	凸轮轨迹（主轴位于指定主轴值）在从轴上的值的二阶导数。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在成功计算从轴值时置位。它在梯级由 false 转换为 true 时重置。
.ER （错误）	此位在未成功计算出从轴值时置位。它在梯级由 false 转换为 true 时重置。

3）功能说明

运动计算从轴值（MCSV）指令确定给定凸轮轨迹和主轴值的从轴值、斜率值和斜率的微分。作为位置和时间凸轮运动功能的扩展，该指令为凸轮运动操作从故障中恢复提供必要的值。

4）注意事项

如果检测到归零事件时，旋转轴上在执行单向主动归零并且归零偏移值小于减速距离，

则控制将轴移 动到值为零的放卷位置。这样可确保到归零位置的移动是单向的。

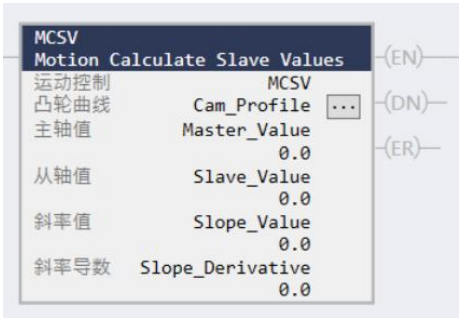
5） 错误说明

MCSV 错误代码（.ERR)

错误消息	代码	说明
Parameter Out Of Range （参数超出范围）	13	试图使用超出范围的输入参数执行指令。
Cam Profile Being Calculated （正在计算凸轮轨迹）	30	试图对当前正在计算的凸轮轨迹数组执行指令。
Cam Profile Not Calculated （未计算凸轮轨迹）	32	试图执行尚未计算的凸轮轨迹段。

6）示例

梯形图：



ST 示例：

MCSV(

MCSV, //运动控制

Cam_Profile, //凸轮曲线

Master_Value, //主轴值

Slave_Value, //从轴值

Slope_Value, //斜率值

Slope_Derivative);//斜率导数

轴位置重新定义 MRP 指令

MRP 指令以指定的方式(绝对/相对)更改轴的命令位置或实际位置。重设位置对正在进行中的运动没有影响。

1）指令格式

指令	名称	梯形图表现	ST 表现
MRP	轴位置重新定义		MRP(refAxis, //轴 miMRP1, //指令变量 Absolute, //类型 Command, //位置选择 mrpPosition); //位置值

2）相关变量

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	变量	要对其执行操作的轴的名称。
Motion Control	指令变量	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

Type	类型	BOOL	立即数	所需的重定义操作工作方式。 二选一： 0 = 绝对 1 = 相对
Position Select	位置选择	BOOL	立即数	选择执行重定义操作的位置。 二选一： 0 = 实际位置 1 = 命令位置
Position	位置	REAL	立即数 变量	用于将轴位置或偏移位置更 改为当前位置的值。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）	此位在梯级由 false 转换为 true 时置位，在伺服消息事务完成并且梯级变为 false 之前，保持为置位状态。
.DN （完成）	此位在成功重定义轴的位置操作时置位。
.ER （错误）	如果此位为置位状态，则指示指令检测到错误，例如指定了未配置的轴。

3）功能说明

运动重定义位置（MRP）指令直接将指定轴的实际或命令位置设置为指定的绝对或相对位置。此指令不会产生任何运动。

如果目标轴没有出现在可用轴列表中，说明该轴没有配置用于操作。 使用变量编辑器创建和配置一个新轴。

轴移动或静止时都可以使用 MRP 指令。MRP 用于为特定定位杆、误差补偿和重校准应用动态重定义位置。

绝对模式：

如果选择或输入 Absolute（绝对）作为 MRP 类型，则新位置指定轴的新绝对位置。不进行任何运动。

相对模式：

如果选择或输入 Relative（相对）作为 MRP 类型，则新位置值用于使轴的当前位置发生偏移。不进行任何运动加上指定的新位置。

在相对模式下，如果轴在移动，重定义轴位置的方法不会产生任何位置误差。对于在程序控制（而不是使用内置旋转轴功能）下放卷轴位置，这尤其有用。

实际位置：

如果选择或输入 Actual（实际）作为 MRP 位置选择，则新位置直接应用于物理轴的实际位置。轴的命令位置也与新实际位置一起进行调整以保留存在的任何位置误差。这样可确保在重新定义位置时，不会出现意外的轴运动。

命令位置：

如果选择或输入 Command（命令）作为 MRP 位置选择，则新位置直接应用于伺服轴

或虚轴的命令位置。由于 Feedback Only 轴没有命令位置，因此始终从 Master Only 轴的 Position 菜单中选择 Actual。伺服轴的实际位置也与新命令位置一起进行调整以保留存在的任何位置误差。这样可确保在重新定义位置时，不会出现意外的轴运动。

命令位置是需要的或命令的伺服位置，由前述任何运动控制指令生成。实际位置是物理轴或虚拟轴的当前位置，由编码器或其他反馈装置测定。位置误差是这两个位置之差，用于驱动电机使实际位置等于命令位置。

4）注意事项

如果使用了软件超行程限制，则新位置必须在最大正行程和最大负行程配置值之间。否则，执行该指令时会产生软件超行程故障。

如果软件超行程限制检查有效，则以绝对模式执行 MRP 可能会使当前最大正行程和最大负行程限制绝对无效。重定义具有行程限制的轴的绝对位置时请小心。

如果轴不移动，则绝对和相对模式 MRP 指令的效果相同。但是，如果轴移动，则绝对模式会产生位置误差，该误差等于轴在执行 MRP 指令和分配新位置期间进行的移动。相对模式不会产生这样的误差，无论轴的速度或位置如何，都能保证位置的精确性。

若要成功执行 MRP 指令，目标轴必须配置为 Servo 或 Feedback Only 轴。否则，指令会出错。

MRP 指令执行可能需要多次扫描来执行，这是因为它需要向运动模块传输多个消息。因此没有立即设置完成（.DN）位，而是仅当成功传输了这些消息后才设置。

5）错误说明

MRP 错误代码（.ERR)

错误消息	代码	说明
Execution Collision	3	试图在该指令已有另一个实例在轴上执 行时执行。如果不

（执行冲突）		验证完成（.DN）位即执行要求进行消息传递的指令，则可能发生这种情况。
Axis Not Configured （轴未配置）	11	传递的轴值引用了未配置的轴，即该轴没有指定到物理运动模块通道。
Servo Message Failure （伺服消息失败）	12	到目标运动模块的消息传递失败。
Parameter Out Of Range （参数超出范围）	13	错误应用于对位置输入参数的检查。 如果轴处于旋转模式，则根据以下内容验证位置： •如果为绝对位置，则限制为小于 unwind 的正值。 •如果为相对位置，则将位置的绝对值限制为小于 unwind（- unwind > MRP 位置 < unwind）。
Home in Process （归位正在进行）	16	试图在进行归位操作时执行。
Axis Type Unused （未用轴类型）	18	试图对未根据当前轴类型配置属性进行配置的轴执行指令。
Axis Group Not Synchronized （轴组未同步）	19	运动组不处于同步状态。伺服模块缺失或配置错误可能导致这种情况。
Illegal Axis Data Type （非法轴数据类型）	38	轴数据类型非法。轴数据类型对于操作不正确。

6）示例

梯形图：

ST 示例：

```
MRP(
    refAxis,        //轴
    miMRP1,         //指令变量
    Absolute,       //类型
    Command,        //位置选择
    mrpPosition); //位置值
```

十一、运动状态指令

轴故障复位 MAFR 指令

使用 MAFR 指令清除轴的所有运动故障。MAFR 指令删除故障状态，但不执行任何其它恢复，例如使能伺服操作。

1）指令格式

指令	名称	梯形图表现	ST 表现
MAFR	轴故障复位		MAFR(Axis, //轴 MotionControl);//指令变量

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
MotionControl	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:

.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴的故障被成功置 0 时，完成位被置 1。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。

3）功能说明

MAFR 指令直接清除指定轴上的指定故障状态。它不会纠正导致错误的条件。如果在执行 MAFR 指令之前未纠正该条件，轴可能会立即再次出现故障，给人一种故障状态未被重置的假象。

此指令最常作为故障处理程序的一部分使用，该程序针对各种潜在的运动控制故障提供特定于应用程序的故障操作。在采取适当的故障操作之后，可以使用 MAFR 指令置 0 所有活动的故障状态位。

执行行为：

情况	行为
Prescan	.EN、.DN 和 .ER 位被置 0。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。
执行条件为 TRUE	将 .EN 置为 True 并执行指令。如果将.EN 位置成 False，则不执行任何操作
Postscan	不适用

4）注意事项

DN 位并非立即置 1。它会在相应的运动模块或驱动器完成所需的重置操作后才会被置 1，这一过程可能需要长达数秒。一旦置 1，轴就处于就绪状态，但这是在请求完成之后。

5）错误说明

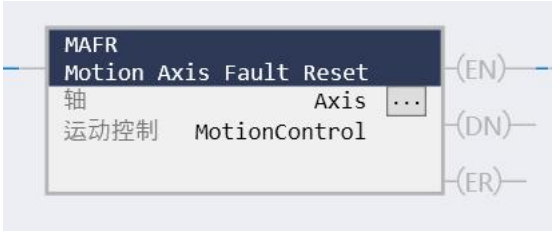
扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MAFR 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

扩展错误代码（十进制）	相关联的错误代码(十进制)	说明
对象模式冲突(12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态
权限被拒绝(15)	SERVO_MESSAGE_FAILURE (12)	使能输入开关错误。
设备处于错误状态(16)	SERVO_MESSAGE_FAILURE (12)	设备状态不适用于该操作。

6）示例

梯形图：

当 MAFR 输入条件为真时，控制器清除 Axis 的所有运动故障。



ST 示例：

当 MAFR 输入条件为真时，控制器清除 Axis 的所有运动故障。

```
MAFR(
        Axis,                   //轴
        MotionControl);         //指令变量
```

轴关断指令 MASD 指令

使用 MASD 指令强制指定轴进入关断状态。轴的关断状态是驱动输出被禁用伺服回路被停用 。在执行轴关断复位之前，轴将保持在关断状态。

1）指令格式

指令	名称	梯形图表现	ST 表现
MASD	轴关断指令		MASD(Axis, //轴 MotionControl); //指令变量

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
MotionControl	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴被成功置于关闭状态时，完成位被置 1。

.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。

3）功能说明

MASD 指令直接且立即禁用驱动器输出，禁用伺服回路，并打开所有相关的 OK 触点。这一操作将轴置于关闭状态。

MASD 指令启动的另一个操作是置 0 所有正在进行的运动过程和所有运动状态位。与这一操作相关，命令还会置 0 针对目标轴当前置 1 的所有运动指令的“IP”位。

MASD 指令强制目标轴进入关闭状态。关闭状态的一个独特特征是，当可用时，运动模块或驱动器的 OK 固态继电器触点是打开的。在可用的情况下，此功能可用于打开控制驱动系统主电源的 E-Stop 环。请注意，通常每个运动模块只有一个 OK 触点，这意味着对给定模块相关的任一轴执行 MASD 指令都会打开 OK 触点。

关机状态的另一个特征是，任何启动轴运动的指令都被阻止执行。尝试这样做会导致执行错误。只有通过执行关闭重置指令之一，才能成功启动运动。

轴将保持在关闭状态，直到执行了运动轴关闭重置（MASR）、运动组关闭重置（MGSR）或运动坐标关闭重置（MCSR）指令。如果轴与坐标系相关联，则在执行运动坐标关闭重置（MCSR）指令时，轴将被重置。

执行行为：

情况	行为
Prescan	.EN、.DN 和 .ER 位被置为 False。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。
执行条件为 TRUE	将 .EN 置为 True 并执行指令。如果将.EN 位置成 False，则不执行任何操作
Postscan	不适用

MASD 对状态位的更改：

位名称	状态	含义
ServoActionStatus	FALSE	轴处于伺服关闭状态，伺服环处于非激活状态。
DriveEnableStatus	FALSE	轴驱动使能输出处于激活状态。
ShutdownStatus	TRUE	轴处于关断状态

4）注意事项

DN 位并非立即置 1。它会在相应的运动模块或驱动器完成所需的重置操作后才会被置 1，这一过程可能需要长达数秒。一旦置 1，轴就处于就绪状态，但这是在请求完成之后。

5）错误说明

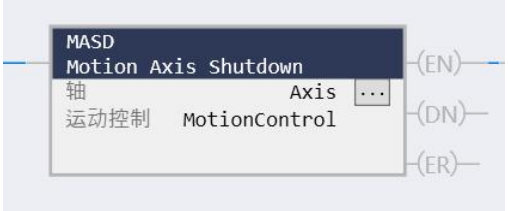
扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MASD 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

扩展错误代码（十进制）	相关联的错误代码(十进制)	说明
对象模式冲突(12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态
权限被拒绝(15)	SERVO_MESSAGE_FAILURE (12)	使能输入开关错误。
设备处于错误状态(16)	SERVO_MESSAGE_FAILURE (12)	设备状态不适用于该操作。

6）示例

梯形图：

当 MASD 输入条件为真时，控制器强制 Axis 进入关闭工作状态



ST 示例：

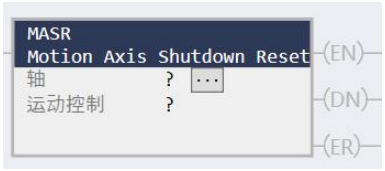
当 MASD 输入条件为真时，控制器强制 Axis 进入关闭工作状态

```
MASD(
        Axis,                   //轴
        MotionControl);         //指令变量
```

轴关断复位 MASR 指令

使用 MASR 指令将轴从现有的关断状态转换为轴就绪状态。

1）指令格式

指令	名称	梯形图表现	ST 表现
MASR	轴关断复位		MASR(Axis, //轴 MotionControl); //指令变量

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
MotionControl	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴被成功从关闭状态中复位时，完成位被置 1。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。

3）功能说明

MASR 指令清除所有轴故障，并将指定轴从关闭状态中移出。如果运动模块支持 OK 触点，并且没有其他模块轴处于关闭状态，MASR 指令将导致模块的 OK 固态继电器触点闭合。无论 OK 触点的状态如何，执行 MASR 指令都会将轴置于轴就绪状态。

正如运动轴关闭（MASD）指令强制目标轴进入关机状态一样，MASR 指令将轴从关闭状态移至轴就绪状态。关闭状态的一个独特特征是，与运动模块相关的任何 OK 固态继电器触点都是打开的。如果由于 MASR 指令的结果，与给定运动模块相关的轴没有处于关闭状态，OK 继电器触点将因 MASR 而闭合。此功能可用于关闭控制驱动系统主电源的 E-Stop 环，从而允许客户重新为驱动系统供电。请注意，通常每个运动模块只有一个 OK 触点，这意味着可能需要对与给定模块相关的所有轴执行 MASR 指令，OK 触点才会闭合。

MASR 指令是一种过程类型命令，从控制器处理，通过相关的运动模块，到达相关的驱动器。

执行行为：

情况	行为
Prescan	.EN、.DN 和 .ER 位被置为 False。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则将.EN 位置为 False。
执行条件为 TRUE	将 .EN 置为 True 并执行指令。如果将.EN 位置成 False，则不执行任何操作
Postscan	不适用

MASR 对状态位的更改：

位名称	状态	含义

ShutdownStatus	FALSE	轴不处于关机状态。

4）注意事项

DN 位并非立即置 1。它会在相应的运动模块或驱动器完成所需的重置操作后才会被置 1，这一过程可能需要长达数秒。一旦置 1，轴就处于就绪状态，但这是在请求完成之后。

5）错误说明

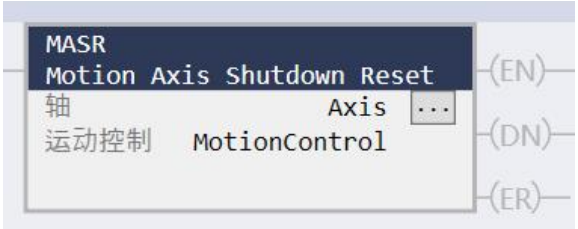
扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MASR 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

扩展错误代码（十进制）	相关联的错误代码(十进制)	说明
对象模式冲突(12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态
权限被拒绝(15)	SERVO_MESSAGE_FAILURE (12)	使能输入开关错误。
设备处于错误状态(16)	SERVO_MESSAGE_FAILURE (12)	设备状态不适用于该操作。

6）示例

梯形图：

当 MASR 输入条件为真时，控制器将 Axis 从关闭工作状态过渡到就绪工作状态。



ST 示例：

当 MASR 输入条件为真时，控制器将 Axis 从关闭工作状态过渡到就绪工作状态。

MASR(

Axis, //轴

MotionControl); //指令变量

伺服断使能 MSF 指令

使用 MSF 指令停用指定轴伺服驱动输出与回路控制。

1）指令格式

指令	名称	梯形图表现	ST 表现
MSF	伺服断使能		MSF(Axis, //轴 MotionControl); //指令变量

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称。
MotionControl	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持置 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴的伺服动作已成功禁用，并且驱动使能和伺服激活状态位已被置 0 时，它被置位。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。

3）功能说明

MSF 指令直接且立即关闭驱动器输出并禁用任何物理伺服轴上的伺服回路。对于非 CIP 运动，这将轴置于“轴就绪”状态，该状态在运动状态指令中有描述。对于 CIP 运动，这将轴置于“停止”状态，该状态也在运动状态指令中有描述。MSF 指令还会禁用在执行时可能处于活动状态的任何运动规划器。MSF 指令不需要参数——只需输入或选择所需的轴。

如果目标轴没有出现在可用轴的列表中，说明该轴尚未被配置为可操作。可以使用变量编辑器（Tag Editor）来创建和配置新的轴。

你可以使用 MSF 指令在必须手动移动轴时关闭伺服动作。即使伺服动作关闭，位置仍然会被跟踪。当再次通过“运动伺服开启”（MSO）指令启用伺服回路时，轴将再次处于闭环控制之下，位于新的位置。

轴的停止行为因驱动器类型而异。在某些情况下，轴会滑行停止；而在其他情况下，轴会利用驱动器的可用停止扭矩减速停止。

对于非 CIP 运动，要成功执行 MSF 指令，目标轴必须被配置为伺服轴。如果未满足此条件，指令将出错。如果轴类型为虚拟（Virtual），指令也会出错，因为对于虚拟轴，伺服动作和驱动器使能状态始终被强制为真。已使用轴数据类型也会出错，因为生产控制器会改变已使用轴的状态。

执行行为：

情况	行为
Prescan	置 0 .EN、.DN 和 .ER 位。 梯级条件输出置为假。
执行条件为 FALSE	如果 .DN 或 .ER 位是 True，则会置 0 .EN 位。
执行条件为 TRUE	更新梯级输出，开始指令执行

Postscan	梯级条件输出被置为 False。

MSF 对状态位的更改：

位名称	状态	含义
ServoActStatus	FALSE	轴处于伺服关闭状态，伺服环处于非激活状态。
DriveEnableStatus	FALSE	轴驱动使能输出处于激活状态。

4）注意事项

该指令的执行可能需要多个扫描周期来完成，因为它需要多次粗略更新才能完成请求。完成位（.DN）并非立即置 1，而是在请求完成后才会被置 1。

5）错误说明

扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MSF 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

扩展错误代码（十进制）	相关联的错误代码(十进制)	说明
对象模式冲突(12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态
权限被拒绝(15)	SERVO_MESSAGE_FAILURE (12)	使能输入开关错误。
设备处于错误状态(16)	SERVO_MESSAGE_FAILURE (12)	设备状态不适用于该操作。

6）示例

梯形图：

当输入条件为真时，控制器禁止伺服驱动器以及 Axis 所配置的轴伺服环。

结构化文本：

当输入条件为真时，控制器禁止伺服驱动器以及 Axis 所配置的轴伺服环。

```
MSF(
    Axis,               //轴
    MotionControl);//指令变量
```

伺服上使能 MSO 指令

使用 MSO 指令激活指定轴伺服驱动输出与回路控制。

1）指令格式

指令	名称	梯形图表现	ST 表现
MSO	伺服上使能		MSO(Axis, //轴 MotionControl);//指令变量

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Axis	轴	AXIS_CIP_DRIVE	变量	要执行操作的实轴名称
Motion Control	运动控制	MOTION_INSTRUCTION	变量	用于访问指令状态参数的结构。

MOTION_INSTRUCTION 结构

助记码:	说明:
.EN （启用）位 31	当梯级进行一次从 False 到 True 的转换时这个位被置 1 并保持 1 状态直到伺服消息转换完成、梯级变为 False 为止。
.DN （完成）位 29	当轴的伺服动作已成功启用，并且驱动使能和伺服激活状态位已被置 1 时，它被置位。
.ER （错误）位 28	如果此位为置位状态，则指示指令检测到错误，例如 指定了未配置的轴。

3）功能说明

MSO 指令直接激活驱动器并启用与物理伺服轴相关联的已配置伺服回路。它可以在程序的任何位置使用，但在轴移动时不应使用。如果尝试这样做，MSO 指令会生成轴运动错误。

MSO 指令通过激活驱动器和相关伺服回路自动启用指定的轴。对于非 CIP 轴，轴的最终状态被称为伺服控制状态。对于 CIP 轴，轴的最终状态被称为运行状态。

这个指令最常见的用途是在准备命令运动之前，激活指定轴在其当前位置的伺服回路。

执行行为：

情况	行为
Prescan	.EN、.DN 和 .ER 位被置为 False
执行条件为 FALSE	如果 .DN 或 .ER 位为真，则将 .EN 位置为 False
执行条件为 TRUE	将 .EN 位置为 True 并执行指令。如果将 EN 位置成 False，则不执行任何操作
Postscan	不适用

MSO 对状态位的更改：

位名称	状态	含义
ServoActStatus	TRUE	轴处于伺服控制状态，伺服环处于激活状态。
DriveEnableStatus	TRUE	轴驱动使能输出处于激活状态。

4）注意事项

该指令的执行可能需要多个扫描周期来完成，因为它需要多次粗略更新才能完成请求。完成位（.DN）并非立即置 1，而是在请求完成后才会被置 1。

5）错误说明

扩展错误代码：扩展错误代码为对于很多指令都通用的错误代码提供更多的指令特定的信息。当 MSO 指令收到伺服消息失败(12)错误消息时，下面的扩展错误代码可帮助确定问题位置。

扩展错误代码（十进制）	相关联的错误代码(十进制)	说明
对象模式冲突(12)	SERVO_MESSAGE_FAILURE (12)	轴处于关闭状态
权限被拒绝(15)	SERVO_MESSAGE_FAILURE (12)	使能输入开关错误。
设备处于错误状态(16)	SERVO_MESSAGE_FAILURE (12)	设备状态不适用于该操作。

6）示例

梯形图：

当输入条件为 True 时，控制器使能伺服驱动器并激活 Axis 所配置的轴伺服环。

ST 示例：

当输入条件为 True 时，控制器使能伺服驱动器并激活 Axis 所配置的轴伺服环。

MSO(

Axis,　　//轴

MotionControl);　//指令变量

十二、程序控制指令

恒假 AFI 指令

AFI 指令将它之后的梯级条件保持为假。

1）指令格式

指令	名称	梯形图表现	ST 表现
AFI	恒假		此指令不适用于结构化文本中

2）相关变量

3）功能说明

使能后，屏蔽 AFI 指令后面的指令执行。

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	将 EnableOut 设置为假。
梯级输入条件为真	将 EnableOut 设置为假

后扫描	不适用

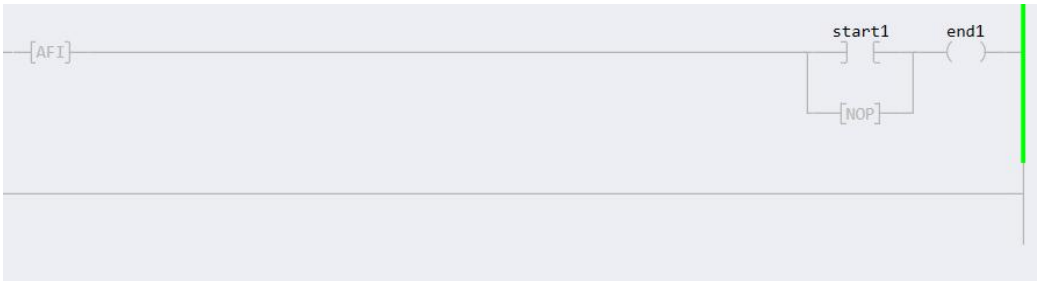
4）注意事项

5）错误说明

6）示例

梯形图：

调试程序时，可使用 AFI 指令临时禁止梯级。 使能后，AFI 将禁止梯级上 AFI 后面的所有指令。



中断 FOR 循环 BRK 指令

BRK 指令中断 FOR 指令调用的例程并跳出 FOR 循环。

1）指令格式

指令	名称	梯形图表现	ST 表现
BRK	中断 FOR 循环		此指令不可用于结构化文本

2）相关变量

3）功能说明

使能后，BRK 指令将退出例程，并将控制权返回到包含最近执行的 FOR 指令的例程中，在该指令之后恢复执行。如果在此扫描期间执行此 BRK 指令之前没有 FOR 指令，则 BRK 不执行任何操作。

如果存在嵌套的 FOR 指令，BRK 指令将控制权返回到最内部的 FOR 指令。

执行行为：

条件/状态	梯形图执行的操作	结构化文本执行的操作
预扫描	不适用	无
梯级输入条件为假	不适用	无
梯级输入条件为真	指令执行	无

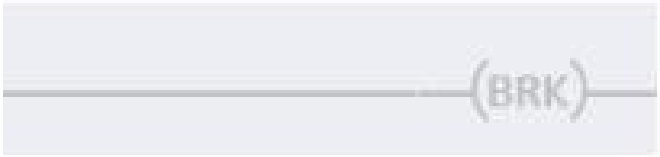
后扫描	不适用	无

4）注意事项

5）错误说明

6）示列

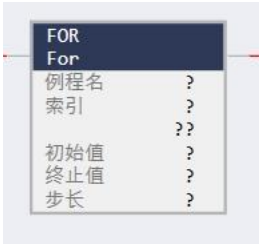
梯形图：



循环调用子例程 FOR 指令

FOR 指令重复执行一个例程。

1）指令格式

指令	名称	梯形图表现	ST 表现
FOR	循环调用子例程		FOR_DO

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Routine name	例程名	ROUTINE	变量	每次 FOR 循环执行时调用的子例程。
Index	索引	DINT	变量	对例程已执行的次数进行计数。

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Initial value	初始值	SINT INT DINT	立即数 变量	索引起始值

		LINT		

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Terminal value	终止值	SINT INT DINT LINT	立即数 变量	达到该值时例程停止执行
Step size	步长	SINT INT DINT LINT	立即数 变量	FOR 指令每次执行例程时索引的增量

3）功能说明

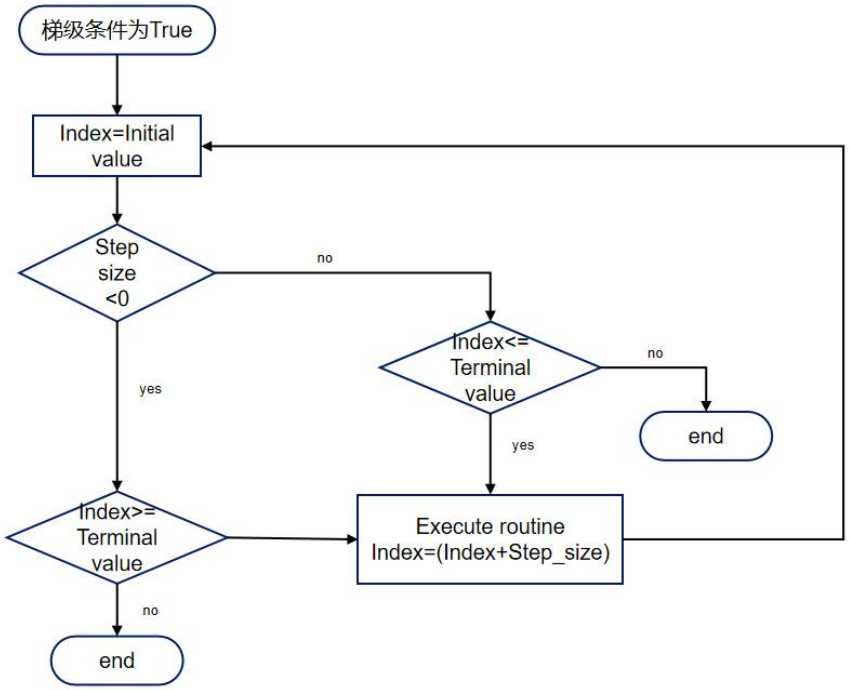
使能后，FOR 指令重复执行例程，直到 Index 值超出 Terminal value。该指令不会将参数传递到例程。

步长值可以是正值，也可以是负值。如果是负数，则当 Index 值小于 Terminal 值时，循环结束。如果是正数，则当索引值大于终止值时，循环结束。

每次 FOR 指令执行例程时，都会向 Index 值加上 Step size 值

执行行为：

条件/状态	梯形图操作
预扫描	如果指定子例程之前未进行过预扫描，则指令会对其进行预扫描。

	提示：如果同一子例程中存在递归 FOR 指令，或者同一子例程中存在多条 FOR 指令（非递归），则该子例程仅预扫描一次。如果下级例程已由 JSR 预扫描，也同样适用。
梯级输入条件为假	**不适用**
梯级输入条件为真	
后扫描	**指令仅对指定子例程执行一次后扫描**

4）注意事项

请注意，在单次扫描中不要循环过多次。重复次数过多可导致控制器的看门狗超时，进而引发严重故障。

5）错误说明

6）示列

梯形图 ：

使能后，FOR 指令重复执行 routine_1，并使 Index 每次加 1。当 Index > 10 或使能 BRK 指令后，FOR 指令不再执行 routine_1。

ST 示列 ：

```
FOR subscript:=0 TO 31 BY 1 DO
    array[subscript] := 0;
END_FOR;
```

跳转到变量 JMP 指令

JMP 和 LBL 指令跳过部分梯形逻辑。

1）指令格式

指令	名称	梯形图表现	ST 表现
JMP	跳转到变量		此指令不适用于结构化文本中

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Lable name	变量名称		变量名称	输入相关 LBL 指令的名称

3）功能说明

使能后，JMP 指令将跳转到参考 LBL 指令，控制器将从那里继续执行。禁止后，JMP 指令不会影响梯形图的执行。

JMP 指令可将梯形图执行位置向前或向后变动。向前跳转到标号可以省略掉一段逻辑并在需要时恢复，从而节省程序的扫描时间。向后跳转可以令控制器重复逻辑的执行。

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用
梯级输入条件为真	执行将跳转到带引用变量名称 LBL 指令所在的梯

	级。
后扫描	不适用

4）注意事项

JMP 和 LBL 指令用于跳过部分梯形图逻辑，需搭配使用。

注意向后跳转的次数不要过多。看门狗计时器可能由于控制器永远无法到达逻辑的末端而超时，从而令控制器出现故障。

控制器不会扫描跳过的逻辑。请将重要逻辑放在跳过的区域之外。

5）错误说明

6）示例

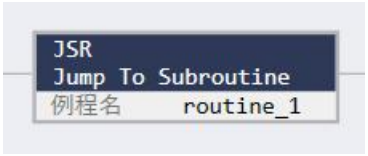
梯形图：

使能 JMP 指令后，执行将跳过连续的逻辑梯级，直至目标梯级，该梯级包含具有 label_20 的 LBL 指令。

跳转到子例程 JSR 指令

JSR 指令跳转到另一个例程执行。

1）指令格式

指令	名称	梯形图表现	ST 表现
JSR	跳转到子例程		JSR(routine_1);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Routine name	例程名称	ROUTINE	名称	要执行的子例程

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Input count	输入计数	SINT INT DINT REAL	立即数	输入参数的数量
Input parameter	输入参数	BOOL SINT	立即数 变量	该例程中要复制到子例程中的变量的数据。

		INT DINT REAL 结构	数组变量	输入参数是可选的。 必要时可输入多个输入 参数。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Return parameter	返回参数	BOOL SINT INT DINT REAL 结构	变量 数组变量	该例程中要将子例程的结 果复制到其上的变量。 返回参数是可选的。 必要时可输入多个返回 参数。

3）功能说明

JSR 指令可启动指定例程(称为子例程)的执行。

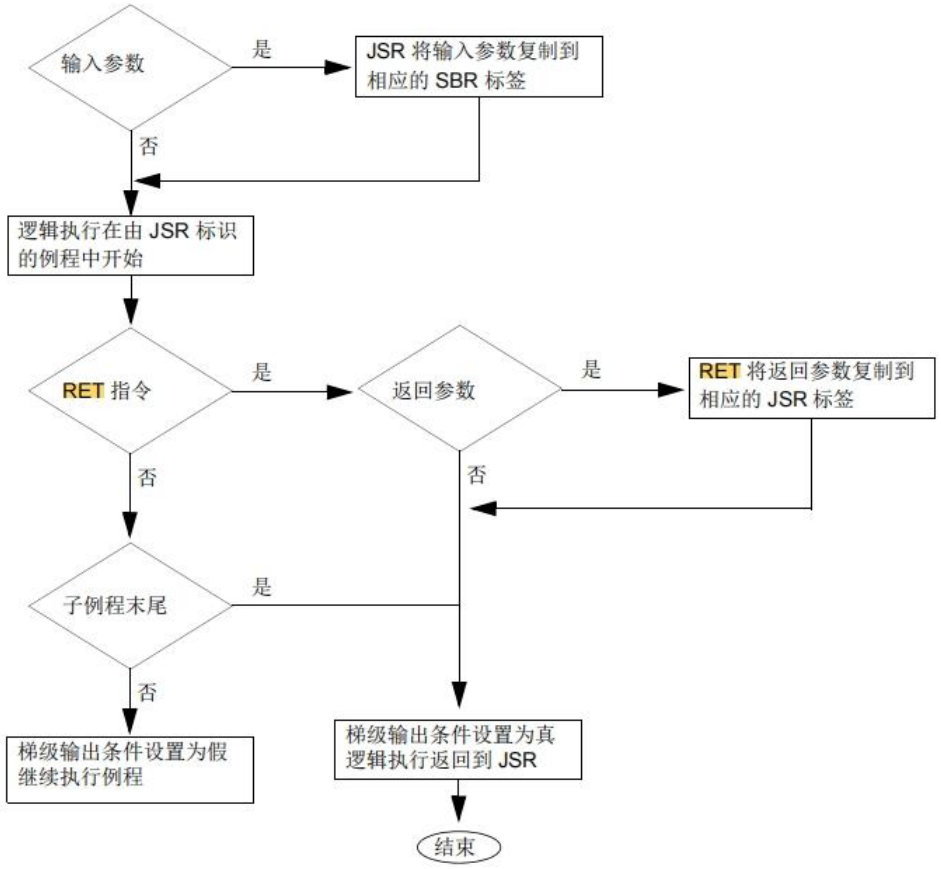
1.子例程将执行一次。

2.子例程执行后，逻辑执行将返回到含有 JSR 指令的例程。

条件	梯形图操作	结构化文本操作
预扫描	梯级设置为假。 控制器执行所有子例程。	同相同条件下梯形图行为

	如果多次调用同一个子例程，则只对其进行一次预扫描。	
梯级输入条件为假	不适用	无
梯级输入条件为真	执行子例程	同相同条件下梯形图行为
后扫描	操作与预扫描相同	同相同条件下梯形图行为

流程图 ：



4）注意事项

不可空引脚。如不指定其他引脚，则关闭其它引脚。

5）错误说明

6）示例

梯形图：

JSR 指令使能后，调用 routine_1 子例程。

ST 示例：

```
JSR(routine_1);
```

JMP 跳转到的变量 LBL 指令

JMP 和 LBL 指令跳过部分梯形逻辑。

1）指令格式

指令	名称	梯形图表现	ST 表现
LBL	变量		此指令不适用于结构化文本中

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Lable name	变量名称		变量名称	执行跳转到具有参考变量名称的 LBL 指令

3）功能说明

当条件为真时，JMP 指令将跳转到引用的 LBL 指令，控制器将从那里继续执行。当条件为假时，JMP 指令不会影响梯形图的执行。

JMP 指令可将梯形图执行位置向前或向后变动。向前跳转到标号可以省略掉一段逻辑并在需要时恢复，从而节省程序的扫描时间。向后跳转可以令控制器重复逻辑的执行。

JMP 及其引用的 LBL 指令必须位于同一个例程中。

条件	梯形图操作
预扫描	不适用

梯级输入条件为假	不适用
梯级输入条件为真	不执行任何操作
后扫描	不适用

4）注意事项

JMP 和 LBL 指令用于跳过部分梯形图逻辑，需搭配使用。

注意向后跳转的次数不要过多。看门狗计时器可能由于控制器永远无法到达逻辑的末端而超时，从而令控制器出现故障。

控制器不会扫描跳过的逻辑。请将重要逻辑放在跳过的区域之外。

LBL 指令必须是梯级中的第一条指令。

在一个例程中，变量名称必须唯一。

5）错误说明

6）示例

梯形：

使能 JMP 指令后，执行将跳过连续的逻辑梯级，直至目标梯级，该梯级包含具有 label_20 的 LBL 指令。

空指令 NOP 指令

NOP 指令为空操作指令。

1）指令格式

指令	名称	梯形图表现	ST 表现
NOP	空指令		此指令不适用于结构化文本中

2）相关变量

3）功能说明

可以将 NOP 指令放在梯级上的任何位置。使能后，NOP 指令不会执 行任何操作。禁止后，NOP 指令不会执行任何操作。

也可用于无输出的逻辑梯级，消除或备用逻辑的编译不通过。

条件	梯形图操作
预扫描	不适用
梯级输入条件为假	不适用

梯级输入条件为真	不适用
后扫描	不适用

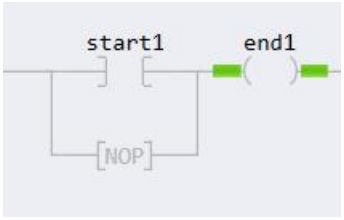
4）注意事项

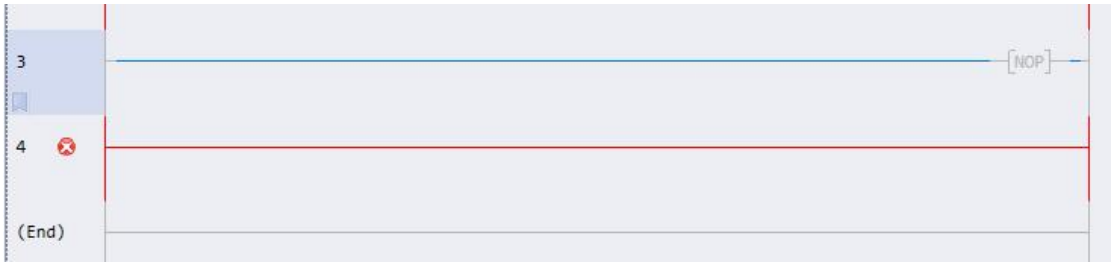
5）错误说明

6）示例

梯形图：

下图将 NOP 指令放在分支上后，该指令可用于定位无条件分支。 NOP 指令可绕过 XIC 指令来使能输出。





子例程返回参数并结束 RET 指令

RET 指令将返回参数传回 JSR 并结束子例程的扫描。

1）指令格式

指令	名称	梯形图表现	ST 表现
RET	子例程返回参数并结束		RET()

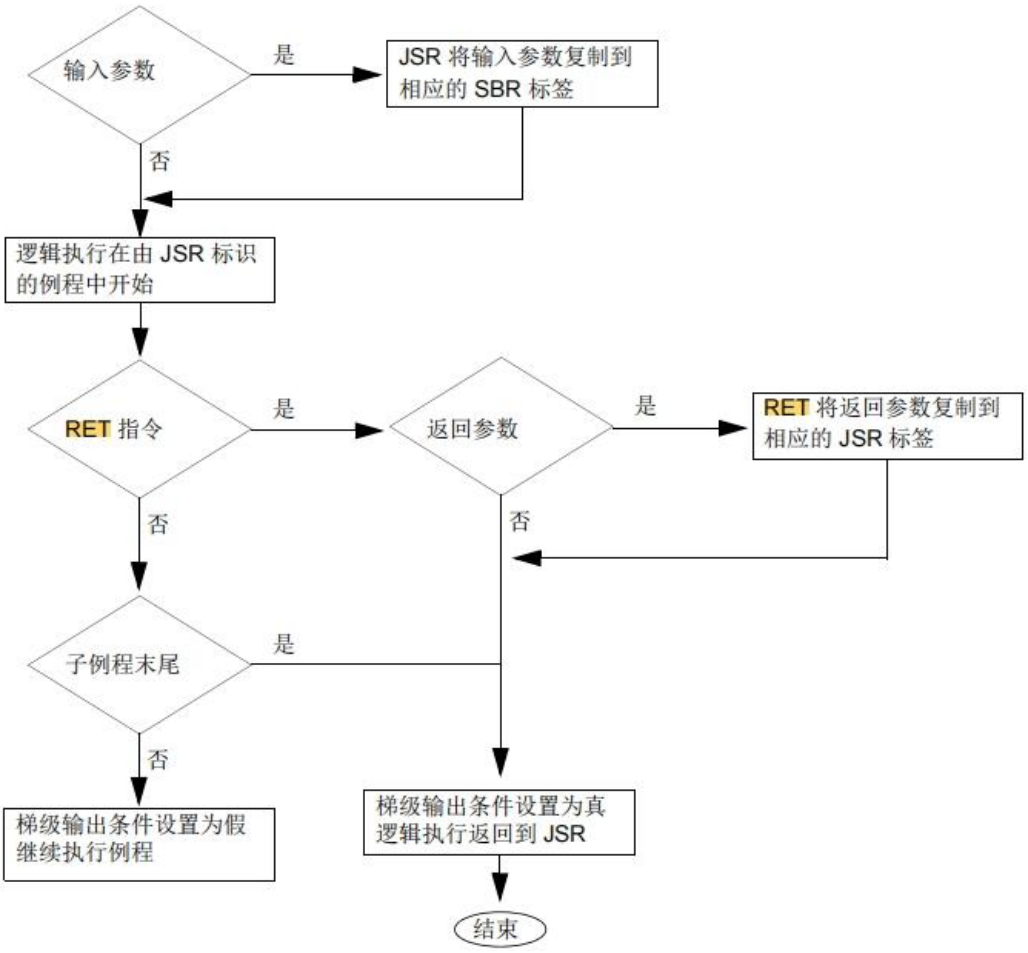
2）相关变量

3）功能说明

条件	梯形图操作	结构化文本操作
预扫描	无论在任何梯级条件下，控制器都将执行所有子例程。为了确保子例程内的所有梯级 都经过预扫描，控制器将忽略 RET 指令。（即， RET 指令不会使子例程退出。） • 在版本 6.*x* 及更早的版本中，传递输入和返回参数。	

	• 在版本 7.*x* 及以后版本中，不传递输入和返回参数。 如果同一子例程存在递归调用，仅在首次执行时预扫描该子例程。如果同一子例程存 在多次调用（ 非递归 ），每次执行时都预扫描该子例程。 梯级输出条件设置为假（ 仅梯形图)。	
梯级输入条件对于 JSR 指令为假	子例程不执行。 子例程中的输出仍保持最后状态。 梯级输出条件设置为假。	N/A
梯级输入条件为真	执行该指令。 梯级输出条件设置为真。	N/A
EnableIn 处于置位状态	N/A	EnableIn 始终置位 执行该指令。

流程图 ：

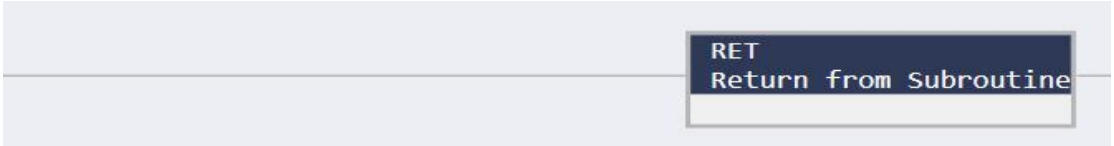


4）注意事项

5）错误说明

6）示列

梯形图 ：



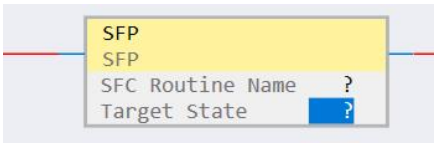
ST 示列 ：

RET();

SFC 暂停 SFP 指令

SFP 指令用于暂停或执行 SFC 例程。

1）指令格式

指令	名称	梯形图表现	ST 表现
SFP	SFC 暂停		SFP(SFCRoutineName,TargetState);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
SFCRoutine Name	SFC 例程名称	ROUTINE	Program 中的例程		名称（Program 中 SFC 例程名字）	要暂停的 SFC 例程
TargetState	目标状态	DINT	Execute/Pause (0-1)		立即数	选择一种状态： 执行（或输入 0） 暂停（或输入 1）

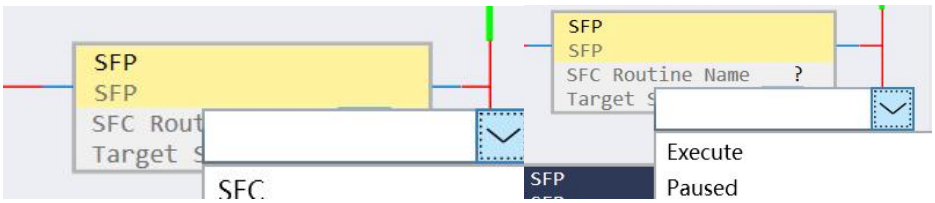
3）功能说明

SFP 指令可暂停一个正在执行的 SFC 例程或恢复一个正在暂停的 SFC 例程。

4）注意事项

1）LD 梯形图上使用与 JSR 指令相似，呈现的样式和操作多增加 TargetState（目标

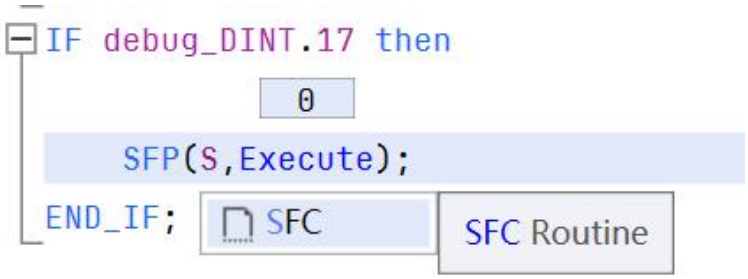
状态)的输入的选择。SFCRoutineName（SFC 例程名称）和 TargetState（目标状态)使用双击后进入编辑选择样式，可通过手动输入(支持自动联想)或通过下拉选择框进行例程名称的选择。下拉选择框内呈现此 Program 中所有的 SFC 例程名称，可通过鼠标选择或上下方向键选中使用 ENTER 键确认选择。TargetState (目标状态)下拉选择框包含 Execute 和 Pause 两种。选择框如下图所示：



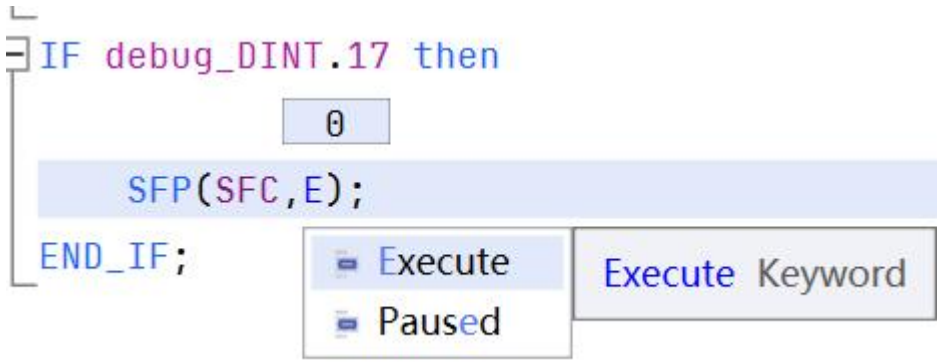
2）ST 结构文本上使用了与 JSR 相同的输入方式

呈现样式 SFP(SFCRoutineName,TargetState);

SFCRoutineName：SFC 例程名字通过输入想暂停或执行的例程的任意字母下方呈现自动联想 Program 中相应的名称。如下图所示：



TargetState：SFC 目标状态通过输入 Execute 和 Pause 或输入立即数 0 或 1 进行 SFC 例程的执行或暂停状态。输入 Execute 和 Pause 其中的字母可进行联想选择框，如下图所示：



编写样式呈现：

1）执行样式

```
SFP(SFC,1);
SFP(SFC,Execute);
```

2）暂停样式

```
SFP(SFC,0);
SFP(SFC,Paused);
```

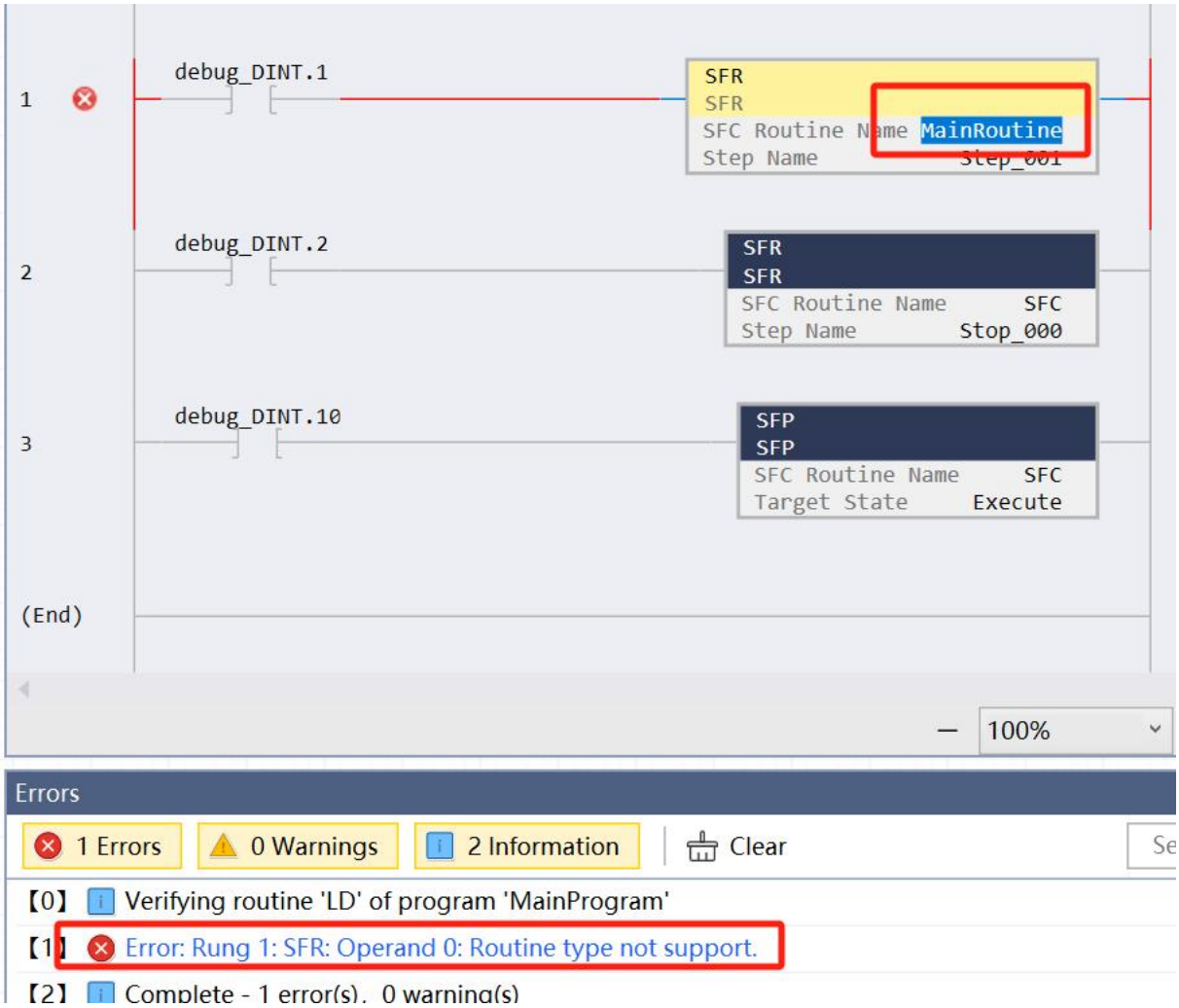
5）错误说明

对于 SFP 指令，包含了所有错误状态内容，详细如下：

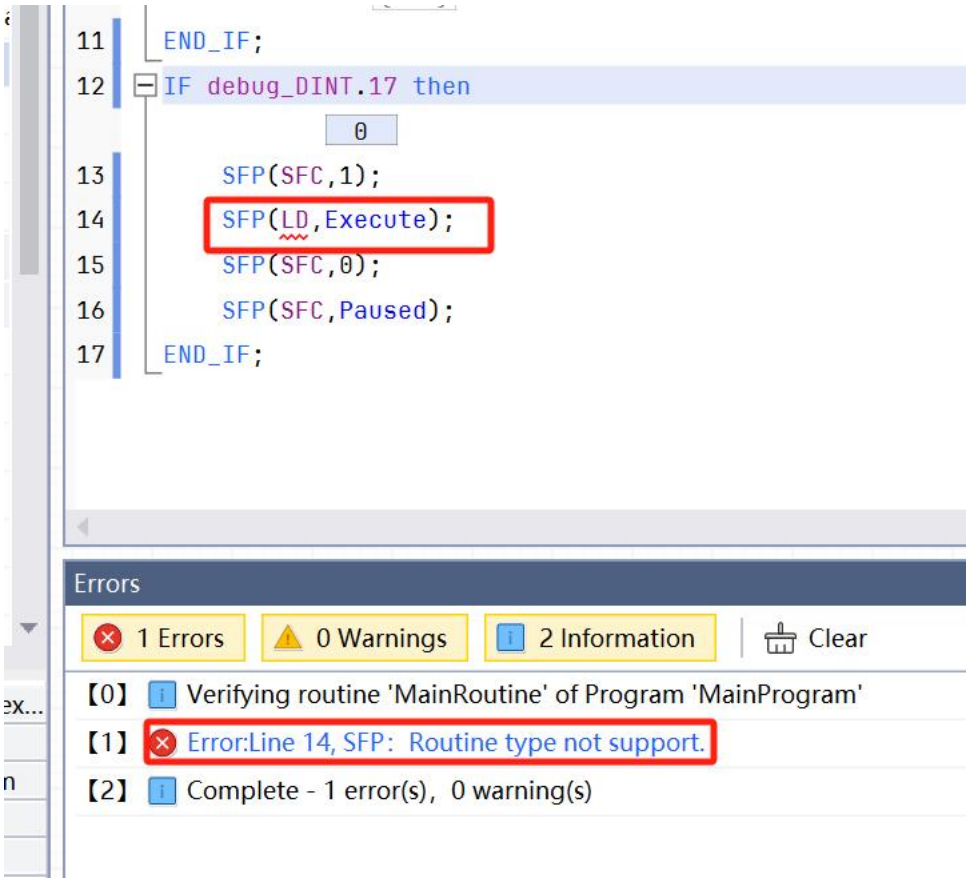
在以下情况下会发生严重故障:	故障类型	故障代码
例程类型不是 SFC 例程	4	85

1. **例程类型不是 SFC 例程：在编辑 SFP 例程名称输入 ST 类型或 LD 类型的例程时会在编辑画面直接报错，自动编译报错不能进行下载。不会产生 PLC 报错情况。如下图显示：**

梯形图画面错误呈现：


1
debug_DINT.1
SFR
SFR
SFC Routine Name MainRoutine
Step Name Step_001
2
debug_DINT.2
SFR
SFR
SFC Routine Name SFC
Step Name Stop_000
3
debug_DINT.10
SFP
SFP
SFC Routine Name SFC
Target State Execute
(End)
100%
Errors
1 Errors
0 Warnings
2 Information
Clear
【0】 Verifying routine 'LD' of program 'MainProgram'
【1】 Error: Rung 1: SFR: Operand 0: Routine type not support.
【2】 Complete - 1 error(s), 0 warning(s)

ST 画面错误呈现：

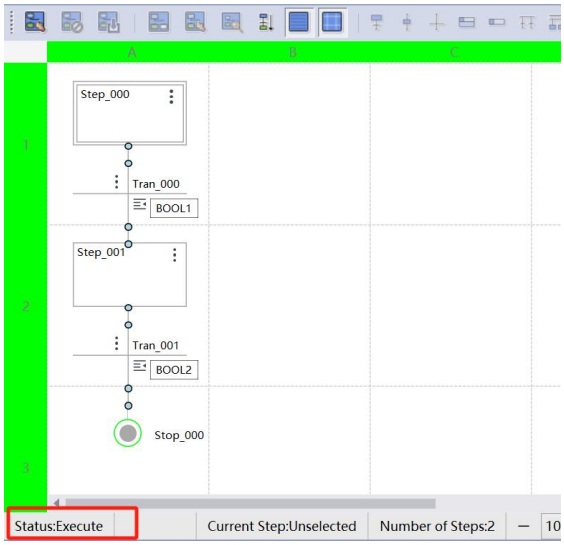


6）示例

梯形图：

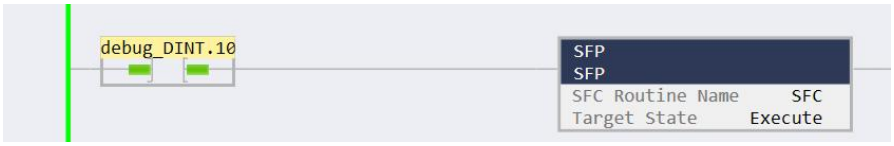
当 SFC 执行，使用 SFP 指令进行 SFC 例程的暂停和执行操作。

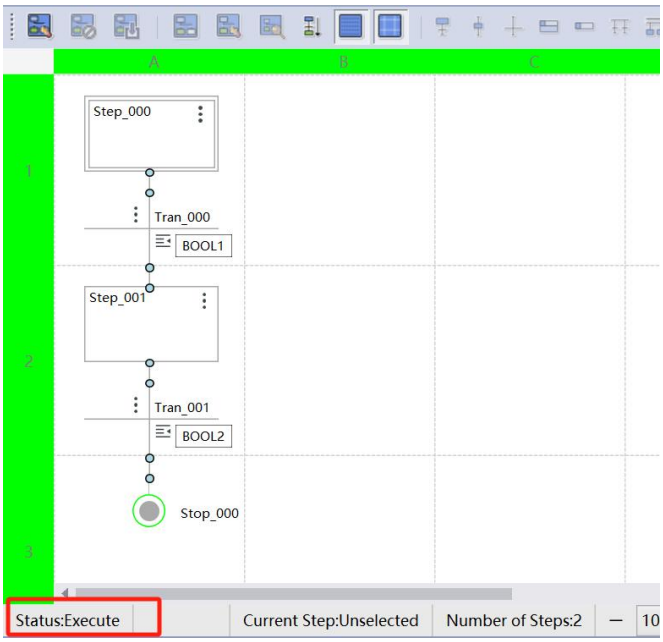
正常 SFC 执行状态，如下图：



使用 SFP 指令进行此例程的暂停操作：

使用 SFP 指令进行此例程的执行操作

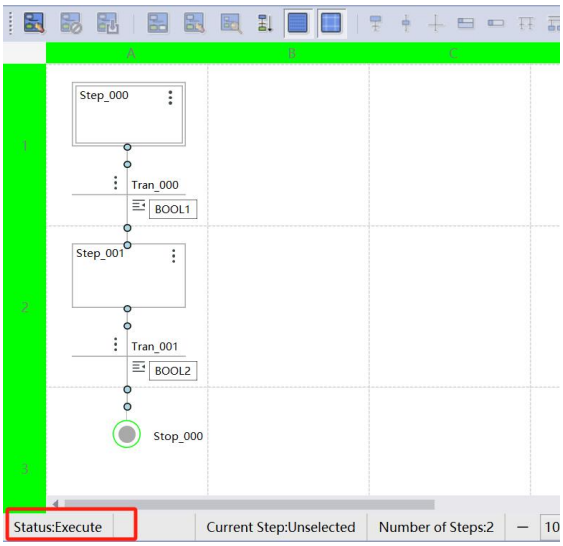




ST 实例 1：

ST 实现前面梯形图实例同样功能。

使用 SFP 指令进行此例程的执行操作：

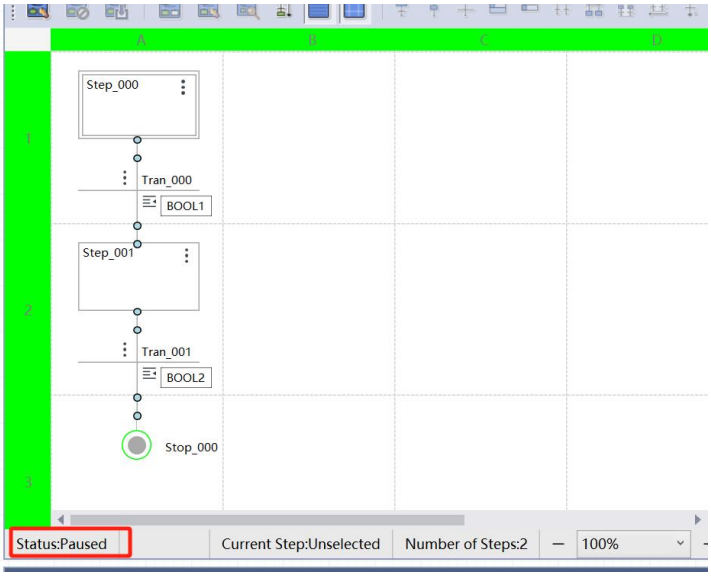


两种书写样式：

```
IF debug_DINT.17 then
    0
    SFP(SFC,Execute);
END_IF;
```

```
IF debug_DINT.19 then
    0
    SFP(SFC,1);
END_IF;
```

使用 SFP 指令进行此例程的暂停操作：



两种书写样式：

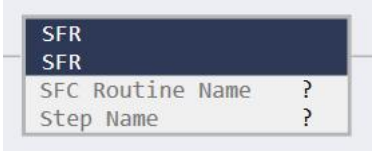
```
IF debug_DINT.18 then
    0
    SFP(SFC,Paused);
END_IF;
```

```
IF debug_DINT.20 then
    0
    SFP(SFC,0);
END_IF;
```

SFC 复位 SFR 指令

SFR 指令可以在指定的步复位 SFC 例程的执行。

1）指令格式

指令	名称	梯形图表现	ST 表现
SFR	SFC 复位		SFR(SFCRoutineName,StepName);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
SFCRoutineName	SFC 例程名称	ROUTINE	Program 中的例程		名称（Program 中 SFC 例程名字）	要复位的 SFC 例程
StepName	步名称	SFC_STEP	复位例程中所有的步序标签名		标签（例程中所有的步标签名字）	目标步，在此处恢复执行

3）功能说明

使能 SFR 指令时：

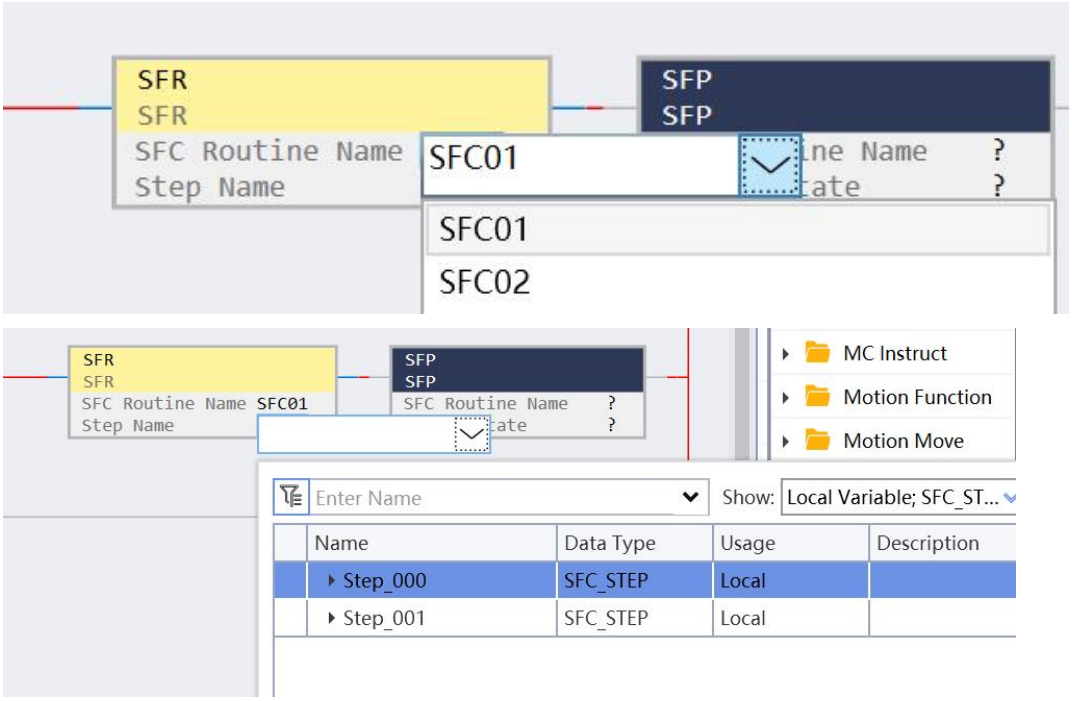
在指定的 SFC 例程内，存储的所有操作都将停止执行（复位）。

SFC 在指定 Step Name(步名称)开始执行。

如果 Step Name(步名称)为 0 时，SFC 顺序功能图将默认复位到初始步。

4）注意事项

1）LD 梯形图上使用与 JSR 指令相似，呈现的样式和操作多增加 Step Name(步名称)的输入的选择。SFCRoutineName（SFC 例程名称）和 stepname(步名称)使用双击后进入编辑选择样式，可通过手动输入（支持自动联想）或通过下拉选择框进行例程名称的选择。下拉选择框内呈现此 Program 中所有的 SFC 例程名称，可通过鼠标选择或上下方向键选中使用 ENTER 键确认选择。选择框如下图所示：

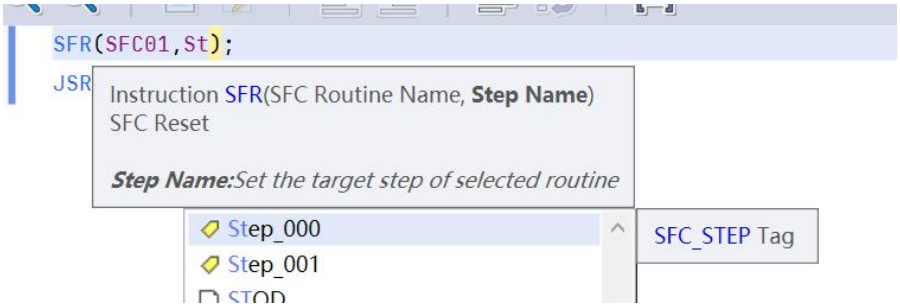


2）ST 结构文本上使用了与 JSR 相同的输入方式

呈现样式 SFR(SFCRoutineName,StepName);

SFCRoutineName：SFC 例程名字通过输入需要复位例程的任意字母下方呈现自动联想 Program 中相应的名称。如下图所示：

StepName：SFC 步序名字通过输入需要复位例程中某一步的任意字母下方呈现自动联想 SFCRoutineName 中相应的名称。如下图所示：



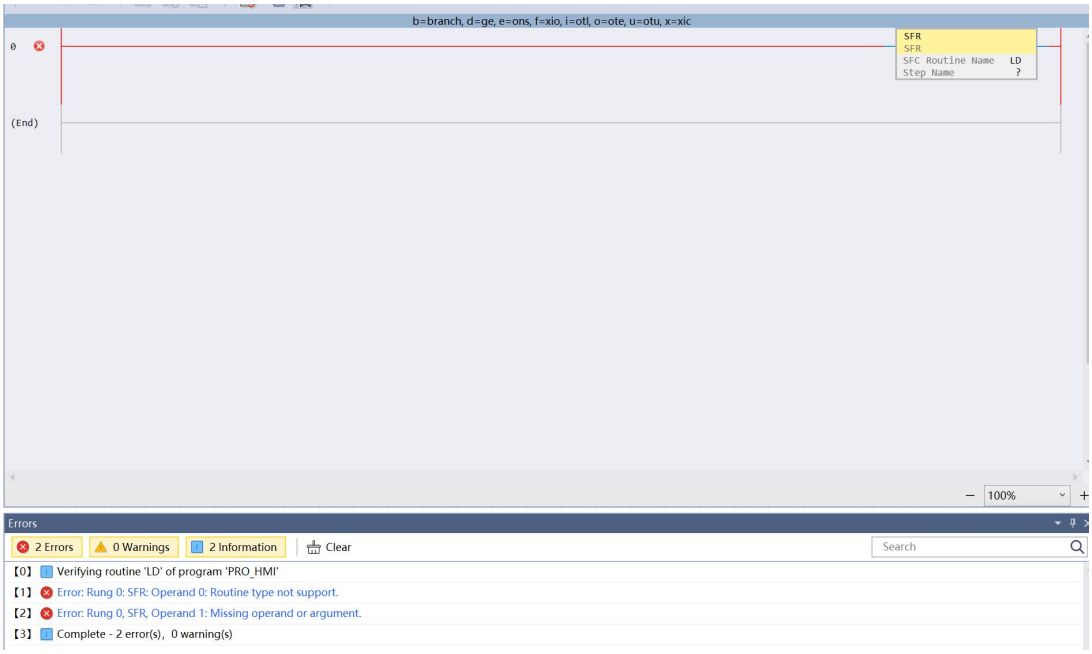
5）错误说明

对于 SFR 指令，包含了所有错误状态内容，详细如下：

在以下情况下会发生严重故障:	故障类型	故障代码
例程类型不是 SFC 例程	4	85
SFC 例程中不存在指定目标步	4	89

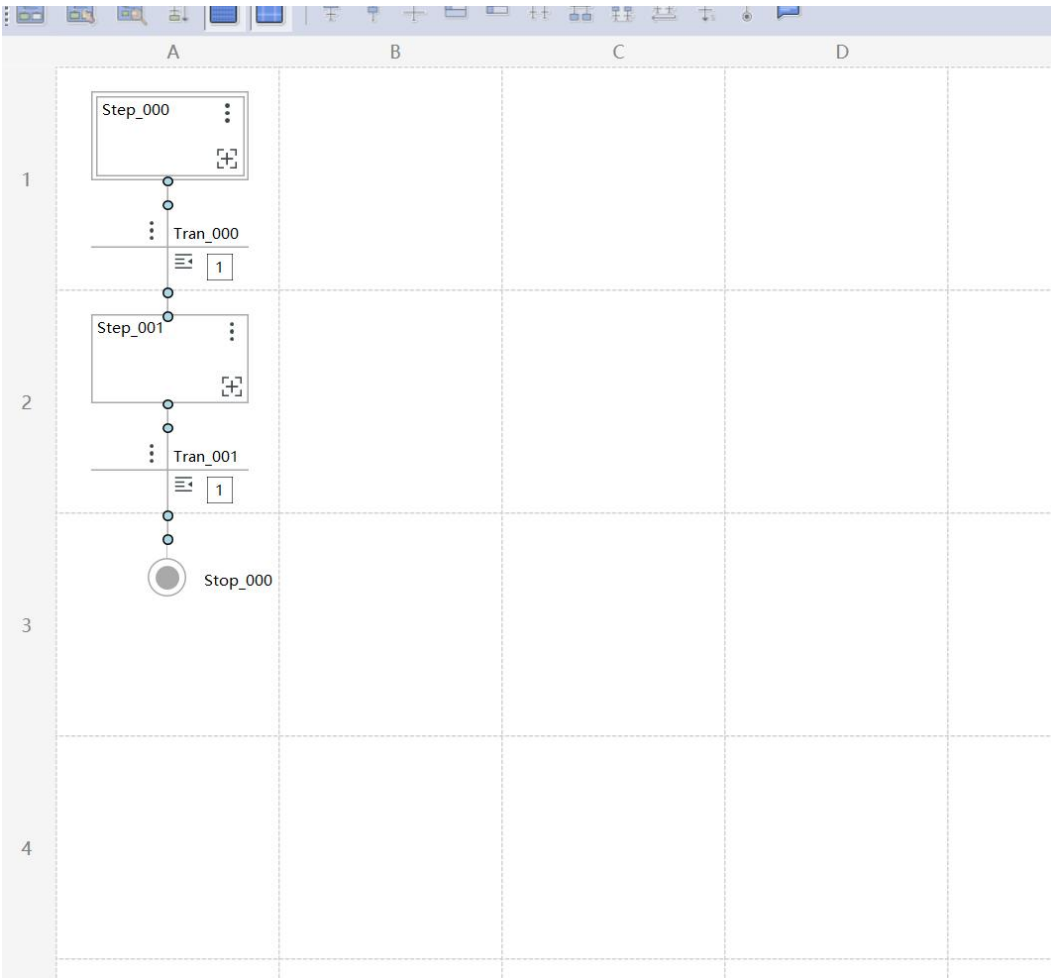
1.例程类型不是 SFC 例程：在编辑 SFR 例程名称输入 ST 类型或 LD 类型的例程时会在编辑画面直接报错，自动编译报错不能进行下载。不会产生 PLC 报错情况。如下图显示：

梯形图画面错误呈现（参考样式）：

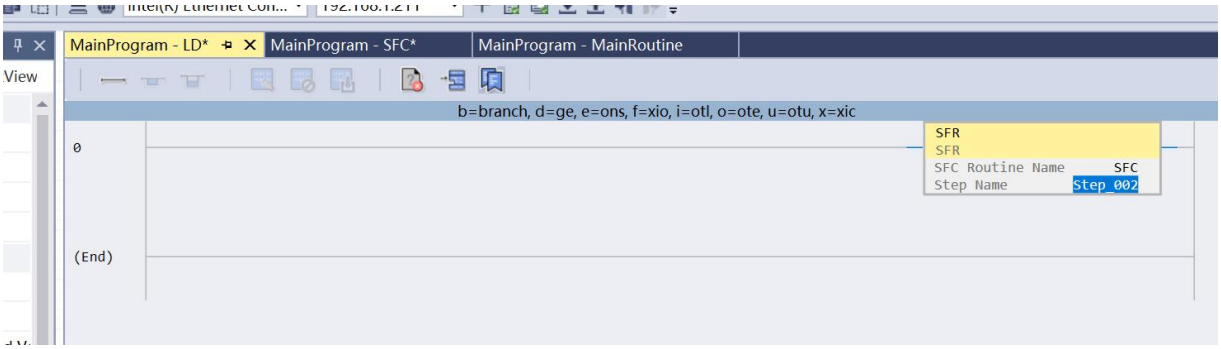


2.SFC 例程中不存在指定目标步：执行 SFR 时如指令中写入了 SFC 中不存在的步名称，在执行 SFR 指令时，PLC 会报错严重故障如下图所示：

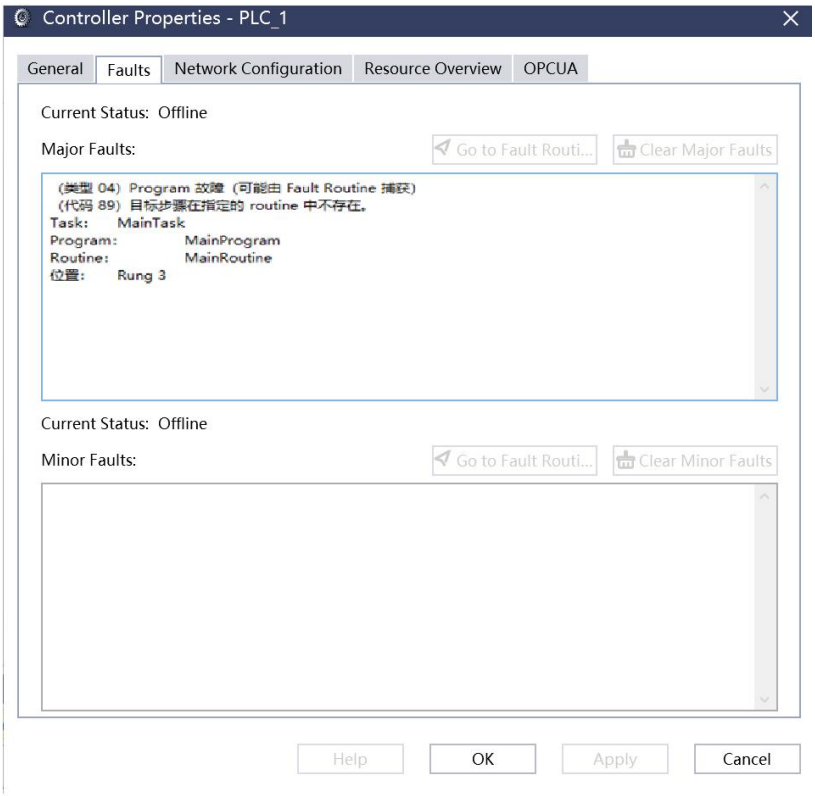
当前 SFC 例程中所有的步序：



编辑一条 SFR 指令执行 SFC 例程中不存在的步序进行复位：



执行指令后 PLC 进入错误状态报错，查看故障显示类型和故障代码显示目标步骤在指定例程中不存在。文案如下：

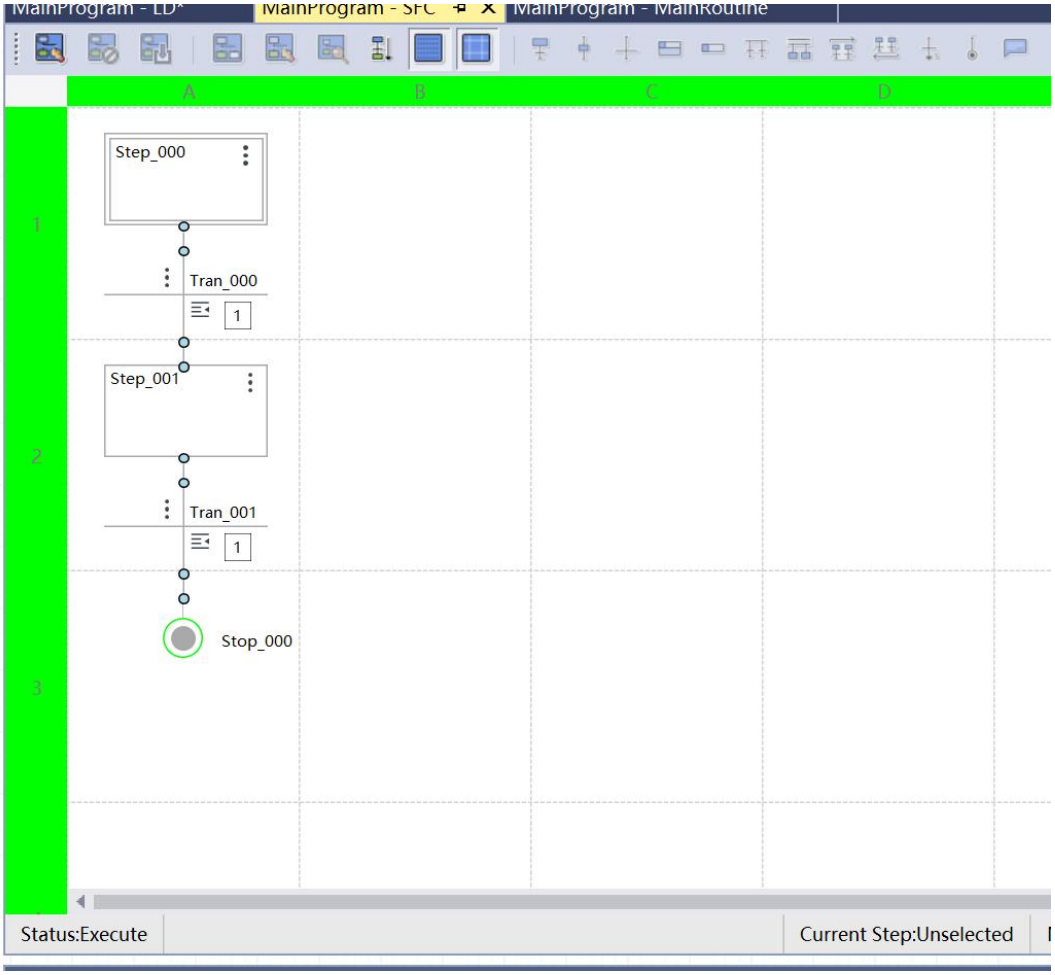


6）示例

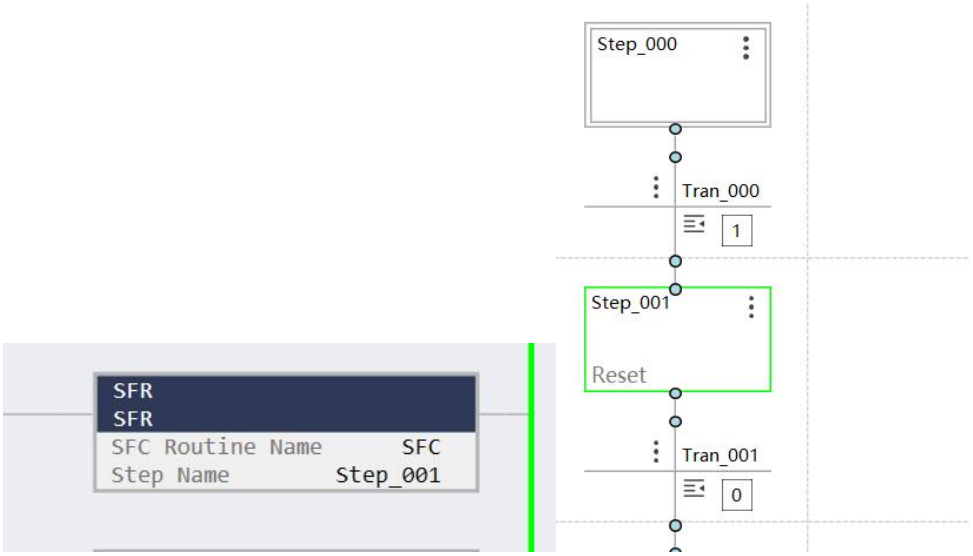
梯形图实例 1：

当 SFC 执行，当前步在 STOP 时执行 SFR 会跳转到相对应的步。当 SFR 步名称写入 0 时默认跳转进入首步。

1）当前步显示在 STOP：



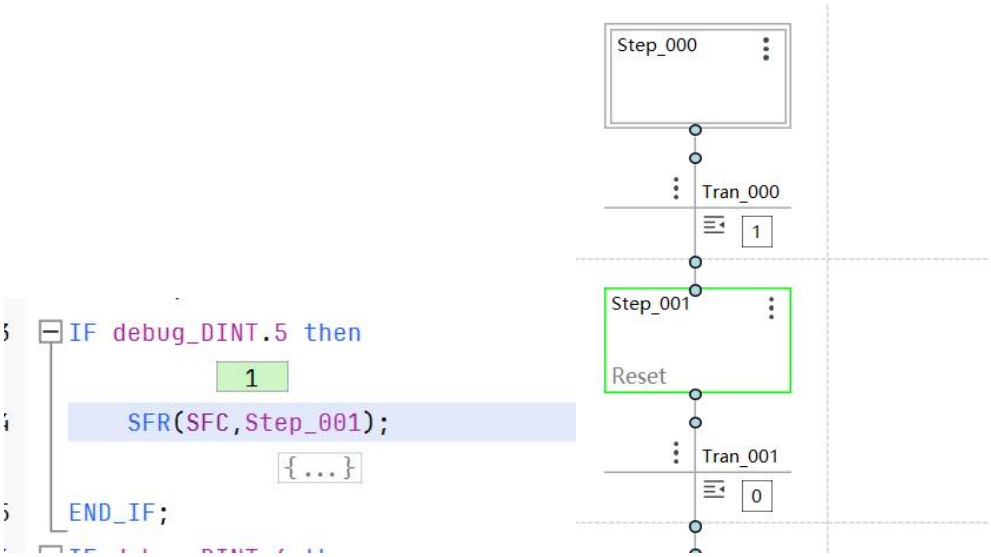
2）使用 SFR 指令进行复位进入 Step_001 步：



ST 实例 1：

ST 实现前面梯形图实例同样功能。

1）使用 SFR 指令进行复位进入 Step_001 步：



临时结束 TND 指令

TND 指令结束当前例程的执行。

1）指令格式

指令	名称	梯形图表现	ST 表现
TND	临时结束		TND()

2）相关变量

3）功能说明

TND 指令用于有条件地结束例程。

使能后，TND 指令将充当例程的末尾。如果 TND 指令位于子例程内，则控制权将返回到调用例程。如果 TND 指令位于主例程内，则控制权将返回到当前任务内的下一个程序。

条件/状态	梯形图执行的操作	结构化文本执行的操作
预扫描	不适用	不适用
梯级输入条件为假	不适用	无
梯级输入条件为真	例程结束	例程结束
后扫描	不适用	不适用

4）注意事项

5）错误说明

6）示列

梯形图 ：

ST 示列 ：

```
TND();
```

禁止用户中断 UID 指令

UID 指令和 UIE 指令可配合使用，以防止少数的重要梯级被其他任务中断。

1）指令格式

指令	名称	梯形图表现	ST 表现
UID	禁止用户中断		UID()

2）相关变量

3）功能说明

当梯级输入条件为真时：

❷ UIE 指令可允许其他任务中断当前任务。

执行行为：

条件/状态	梯形图执行的操作	结构化文本执行的操作
预扫描	不适用	不适用
梯级输入条件为假	不适用	无
梯级输入条件为真	UID 指令可防止涉及的用户任务被中断。	UID 指令可防止涉及的用户任务被中断。
后扫描	不适用	不适用

4）注意事项

要防止一系列梯级被中断：

1. 尽可能将不想被中断的梯级的数量降到最低。如果长时间禁用中断，可能会造成通信丢失。

2. 在不想被中断的第一个梯级上面，输入一个梯级和一条 UID 指令。

3. 在不想被中断的梯级系列的最后一个梯级的后面，输入一个梯级和一条 UIE 指令。

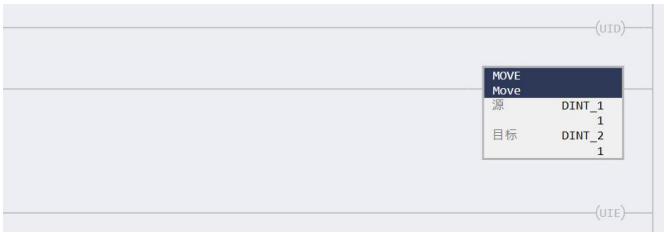
4. 必要时，可以嵌套 UID/UIE 指令对。

当 UID 第一次被调用时，它会改变优先级，保存旧的优先级，并使嵌套计数器值递增。每个后续调用都会使计数递增。UIE 将使嵌套计数器值递减。如果新值为 0，则会恢复保存的优先级。

5）错误说明

6）示列

梯形图 ：



ST 示列 ：

```
UID( );

DINT_1:=DINT_2;

UIE( );
```

允许用户中断 UIE 指令

UID 指令和 UIE 指令可配合使用，以防止少数的重要梯级被其他任务中断。

1）指令格式

指令	名称	梯形图表现	ST 表现
UIE	允许用户中断		UIE()

2）相关变量

3）功能说明

当梯级输入条件为真时：

❷ UID 指令将阻止优先级较高的任务中断当前的任务，但不会禁止故障例程或控制器故障处理器的执行。

执行行为：

条件/状态	梯形图执行的操作	结构化文本执行的操作
预扫描	不适用	不适用
梯级输入条件为假	不适用	无
梯级输入条件为真	UIE 指令在通常情况下允许涉及的用户任务被中断。	UIE 指令在通常情况下允许涉及的用户任务被中断。

后扫描	不适用	不适用

4）注意事项

要防止一系列梯级被中断：

1. 尽可能将不想被中断的梯级的数量降到最低。如果长时间禁用中断，可能会造成通信丢失。

2. 在不想被中断的第一个梯级上面，输入一个梯级和一条 UID 指令。

3. 在不想被中断的梯级系列的最后一个梯级的后面，输入一个梯级和一条 UIE 指令。

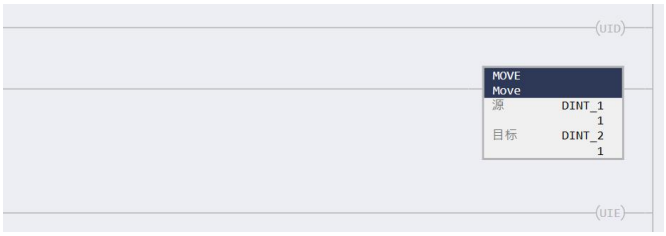
4. 必要时，可以嵌套 UID/UIE 指令对。

当 UID 第一次被调用时，它会改变优先级，保存旧的优先级，并使嵌套计数器值递增。每个后续调用都会使计数递增。UIE 将使嵌套计数器值递减。如果新值为 0，则会恢复保存的优先级。

5）错误说明

6）示列

梯形图：



ST 示列：

```
UID( );
DINT_1:=DINT_2;
```

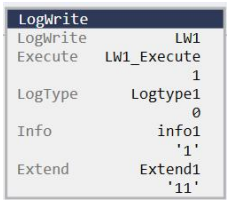
```
UIE( );
```

十三、SD 指令

日志写入 LogWrite 指令

向 SD 卡中写入日志

1）指令格式

指令	名称	梯形图表现	ST 表现
LogWrite	中文: 日志写入 英文: LogWrite		Lw1(execute:=LW1_Execute,LogType:=Logtype1, info:=info1, extend:=extend1);

2）相关变量

输入变量

参数(英文)	参数(中文)	数据类型	有效范围	默认值	格式	说明
Execute	执行	BOOL	0 或 1	0	标签	中文：检查到数据位跳变（0->1）时，向 SD 卡中写入日志。 英文：When a data bit jump (0->1) is detected, write a log to sd card.

LogType	日志类型	DINT	-2,147,483,648 ~ 2,147,483,647	0	标签	中文：日志类型。 英文：Log Type.
Info	信息	String			标签	中文：信息。 英文：Information.
Extend	扩展信息	String			标签	中文：扩展信息。 英文：Extend Information.
Done	指令完成	BOOL	0 或 1	0	标签	中文：指令执行完成。 英文：Instruction Execute Done.
Busy	指令执行中	BOOL	0 或 1	0	标签	中文：指令执行中。 英文：Instruction Is Executing.
Error	指令错误	BOOL	0 或 1	0	标签	中文：指令执行错误。 英文：Instruction Execute Error.
Status	状态	DINT	-2,147,483,648 ~ 2,147,483,647	0	标签	中文：状态码。 英文：Instruction Execute Status.

3）功能说明

1、输入日志类型，信息，扩展信息，将 Execute 置 1，可以向 SD 卡中写入一条日志，

指令的状态有 Done,Busy,Error,Status.

2、将 Execute 置 0， 复位指令状态

4）注意事项

5）错误说明

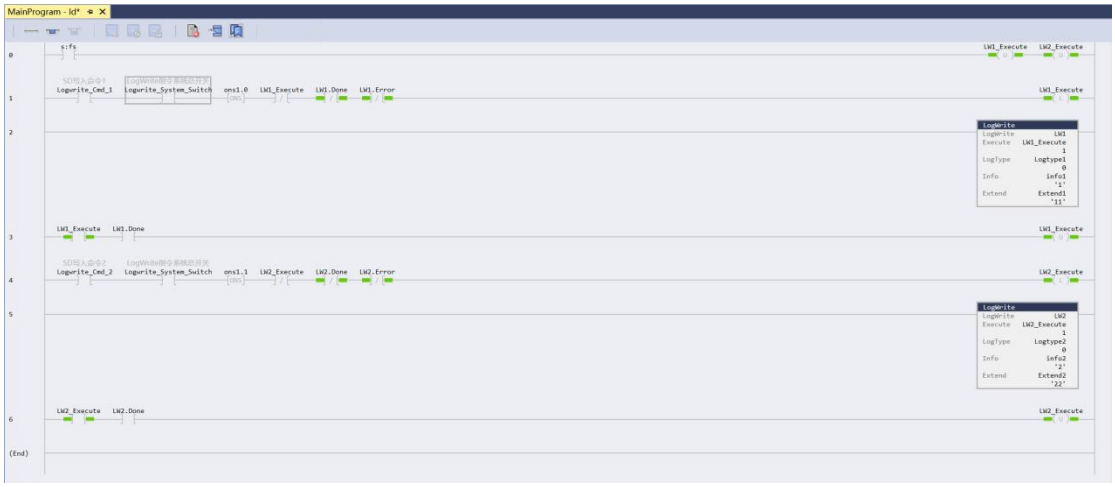
1. 若 SD 卡未插入，或 SD 卡不识别，执行 LogWrite 报错

2. 若 SD 卡满，执行 LogWrite 报错

3. 若 LogWrite 指令的系统缓冲区满,譬如说同时执行了超过 128 条 LogWrite 指令，执行 LogWrite 报错

6）示例

a、下段程序的 LogWrite_Cmd_1 和 LogWrite_Cmd_2 置 1，均可以向 SD 卡写入日志;



十四、过程控制&滤波器指令

报警 ALM 指令-ST

ALM 指令可对任何模拟信号进行报警。

1）指令格式

指令	名称	梯形图表现	ST 表现
ALM	报警	此指令不可用于梯形图中	ALM(ALM_tag)

2）相关变量

结构化文本

变量	数据类型	格式	说明
ALM tag	ALARM	结构	INTG 结构

ALARM 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数 默认置位	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。
IN	指令的模拟	REAL	变量	指令的模拟信号输入。

	信号输入		立即数	有效值 = 任意浮点值 默认值 = 0.0
HHLimit	输入高-高报警限制	REAL	变量 立即数	输入高-高报警限制。 有效=任何实值 Default =最大值
HLimit	输入报警限高	REAL	变量 立即数	输入报警限高。 有效=任何实值 Default =最大值
LLimit	输入端报警下限	REAL	变量 立即数	输入端报警下限。 有效=任何实值 默认值=最大的负值
LLLimit	输入端 low-low 报警限幅	REAL	变量 立即数	输入端 low-low 报警限幅。 有效=任何实值 Default =最大的负值
<Deadband	高-高到低-低阈值的报警死区	REAL	变量 立即数	高-高到低-低阈值的报警死区 有效=大于或等于 0.0 的任何实数 默认值= 0.0
ROCPosLimit	输入正变化的变化率报警极限	REAL	变量 立即数	以单位每秒为单位的输入正（递增）变化的变化率报警极限。设置 ROCPosLimit = 0 禁用 ROC 正告警。如果无效，则该指令假定值为 0.0，并在 Status 中设置相应的位。

				有效=小于或等于 0.0 的任何实数 默认值= 0.0
ROCNegLimit	输入负变化的变化率报警极限	REAL	变量 立即数	以单位每秒为单位的输入负（递减）变化的变化率报警极限。设置 ROCNegLimit = 0，关闭 ROC 负告警。如果无效，则该指令假定值为 0.0，并在 Status 中设置相应的位。 有效=小于或等于 0.0 的任何实数 默认值= 0.0
ROCPeriod	计算变化率值的时间周期	REAL	变量 立即数	计算变化率值的时间周期（采样间隔），以秒为单位。每次采样间隔过期，存储一个新的 In 样本，并重新计算 ROC。而不是像模拟警报中的其他条件一样启用位，变化率检测通过在 ROCPeriod 中放置任何非零值来启用。 Valid = 0.0 到 32767.0 默认值= 0.0。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。

HHAlarm	高-高报警指示灯	BOOL	变量	高-高报警指示灯。 默认值= false
HAlarm	高报警指示灯	BOOL	变量	高报警指示灯。 默认值= false
LAlarm	低报警指示灯	BOOL	变量	低报警指示灯。 默认值= false
LLAlarm	low-low 报警指示灯	BOOL	变量	low-low 报警指示灯。 默认值= false。
ROCPosAlarm	变化率正报警指示灯	BOOL	变量	变化率正报警指示灯。 默认值= false
ROCNegAlarm	变化率负报警指示灯	BOOL	变量	变化率负报警指示灯。 默认值= false
ROC	输出的变化率	BOOL	变量	输出的变化率。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。

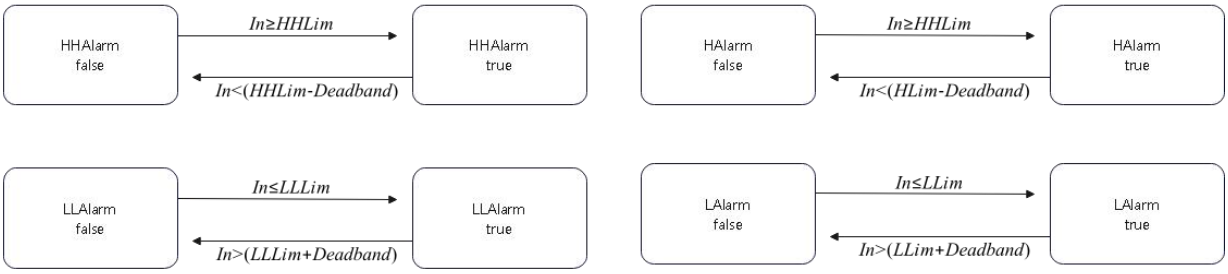
IDeadbandInv (Status.1)	Status.1	BOOL	变量	Deadbandd 无效。
ROCPosLimitInv (Status.2)	Status.2	BOOL	变量	ROCPosLimit 无效。
ROCNegLimitInv (Status.3)	Status.3	BOOL	变量	ROCNegLimit 无效。
ROCPeriodInv (Status.4)	Status.4	BOOL	变量	ROCPeriod 无效。

3）功能说明

ALM 指令为高-高、高、低、低-低、正变化率和负变化率提供报警指示器。high-high、low-low 报警有报警死带。还可以使用用户定义的时间段来执行变化率警报。

高-高到低-低报警

high-high 和 low-low 报警算法将输入值与报警限位值和报警限位值加上或减去限制值进行比较。



变化率报警

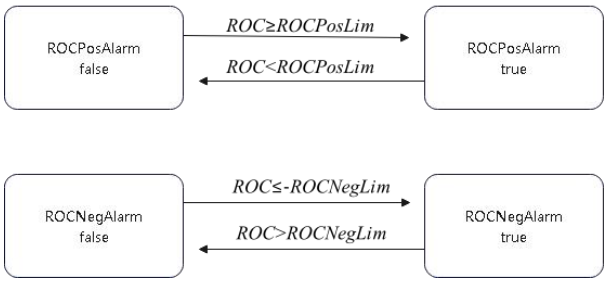
变化率（ROC）警报将在 ROC 周期内输入的变化与变化率限制进行比较。ROCPeriod

为变化率报警提供了一种限制类型。例如，定义一个 ROC 警报限制为 2OF/秒，执行周期为 100ms。如果使用分辨率为 1OF 的模拟输入模块，则每次输入值发生变化时，都会产生 ROC 报警，因为指令计算出 10°F/秒的有效速率。但是，如果输入 1 秒的 ROCPeriod，则该指令仅在速率真正超过 2OF/秒限制时才产生警报。

ROC 报警计算变化率为：

$$ROC = \frac{In(Now) - In(EndofpreviousROCPeriod)}{ROCPeriod}$$

该指令在 ROCPeriod 到期时执行此计算。一旦指令计算出 ROC，它确定警报为：



结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

ALM 指令通常用于不支持板载报警的模拟输入模块（如 1771 I/O 模块），或在计算变量上生成报警。在这个例子中，来自 1771-IFE 模块的模拟输入首先使用 SCL 指令缩放到工程单元。Out of The SCL 指令是 ALM 指令的输入，用于确定是否设置告警。由此产生的报警输出参数可以在您的程序中使用和/或在操作界面显示上查看。

ST 示例：

```
SCL_01.IN := Input0From1771IFE;

SCL(SCL_01);

ALM_01.IN := SCL_01.Out;

ALM(ALM_01);
```

微分 DERV 指令-ST

DERV 指令以每秒为单位计算信号随时间的变化量。

1）指令格式

指令	名称	梯形图表现	ST 表现
DERV	微分	此指令不可用于梯形图中	DERV(DERV_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
DERV tag	DERIVATIVE	结构	DERV 结构

DERV 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0

Gain	导数乘数	REAL	变量 立即数	导数乘数 有效值 = 任意浮点数 默认值 = 1.0
ByPass	请求绕过该算法	REAL	变量 立即数	请求绕过该算法。当 ByPass 为真时，指令集 Out 将等于 In。 默认情况为假。
TimingMode	选择时序执行模式	DINT	变量 立即数	选择时序执行模式。 0 = 周期性模式 1 = 过采样模式 2 = 实时采样模式 有关时序模式的详细信息，请参见“功能块属性”部分。 有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模式的模块更新周期	DINT	变量 立即数	实时采样模式的模块更新周期 有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时	DINT	变量 立即数	实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms

	戳值			默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS | DeltaT - RTSTime | > 1（0.001 秒）时置位。
RTSTimeInv	Status.29	BOOL	变量	RTSTime 值无效。

(Status.29)				
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaT (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

DERV 指令支持一个旁路输入端口，通过该端口您可以停止计算导数，并将信号直接传递至输出端口。

当 Bypass 处于状态时	该指令使用了这个方程式
Cleared and DeltaT > 0	$Out = Gain \frac{In_n - In_{n-1}}{DeltaT}$ $In_{n-1} = In_n$ 其中 ΔT 以秒为单位（DeltaT 表示时间差）
Set	$Out = In_n$ $In_{n-1} = In_n$

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

此示例是 DERV 函数块的最小合法编程，仅用于显示此指令的中性文本和生成的代码。这仅用于内部目的，不是可测试的案例。

ST 示例：

```
DERV_01.In := Speed_Reference;
DERV_01.Gain := Feedforward_Gain;
DERV(DERV_01);

PI_01.In := Speed_Reference - Speed_feedback;
PI_01.Kp := Proportional_Gain;
PI_01.Wld := Integral_Gain;
```

```
PI(PI_01);

regulator_out := DERV_01.Out + PI_01.Out;
```

高通滤波器 HPF 指令-ST

HPF 指令提供一个滤波器来衰减高于截止频率的输入频率。

1）指令格式

指令	名称	梯形图表现	ST 表现
HPF	高通滤波器	此指令不可用于梯形图中	HPF(HPF_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
HPF tag	FILTER_LOW_PASS	结构	HPF 结构

HPF 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟	REAL	变量	指令的模拟信号输入。

	信号输入		立即数	有效值 = 任意浮点值 默认值 = 0.0
Initialize	控制算法初始化请求	BOOL	变量 立即数	控制算法初始化请求。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
WLag	滞后频率	REAL	变量 立即数	滞后频率（单位：弧度/秒）。若 WLag 小于最小值或大于最大值,则指令会在状态寄存器中设置相应的位，并限制 WLag 的取值范围。 有效值 = 参见下文的描述部分以获取有效范围 默认值 = 0.0
Order	滤波器的阶数	REAL	变量 立即数	滤波器的阶数。阶数控制截止频率的锐度。如果阶数无效，则指令会在状态寄存器中设置相应的位，并将阶数设为 1。 有效值 = 1 到 3 默认值 = 1
TimingMode	选择时序执行模式	DINT	变量 立即数	选择时序执行模式。 0 = 周期性模式 1 = 过采样模式 2 = 实时采样模式

				有关时序模式的详细信息，请参见“功能块属性”部分。 有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模式的模块更新周期	DINT	变量 立即数	实时采样模式的模块更新周期 有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时戳值	DINT	变量 立即数	实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。

Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
WLagInv (Status.1)	Status.1	BOOL	变量	WLag > 最大值或 WLag < 最小值。
OrderInv (Status.2)	Status.2	BOOL	变量	无效命令值
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS | DeltaT - RTSTime | > 1（0.001 秒）时置位。
RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效。
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaT (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

HPF 指令通过 Order 参数来控制截止频率的陡峭程度。HPF 指令旨在在一个扫描速率保持恒定的任务中执行。

条件/状态	该指令使用了这样的拉普拉斯传递函数：
Order = 1	$\frac{s}{s+\omega}$
Order = 2	$\frac{s^2}{s^2+\sqrt{2}\times s\times\omega+\omega^2}$
Order = 3	$\frac{s^3}{s^3+\left(2\times s^2\times w\right)+\left(2\times s\ \times w^2\right)\times w^3}$

按照这些参数限制（其中 DeltaT 以秒为单位）：

参数	限制
WLag 第一阶 下限限值	$\frac{0.0000001}{DeltaT}$
WLag 第二阶 下限限值	$\frac{0.00001}{DeltaT}$
WLag 第三阶 下限限值	$\frac{0.001}{DeltaT}$
高限限值	$\frac{0.7\pi}{DeltaT}$

每当计算得出的输出值无效（即为 NAN 值，或为正负无穷大）时，指令将 Out 设为无效值。当计算得出的输出值变为有效值时，指令会初始化内部参数，并将 Out 设为 In。

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

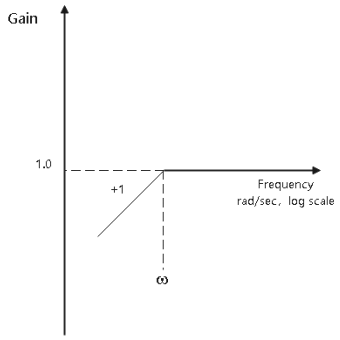
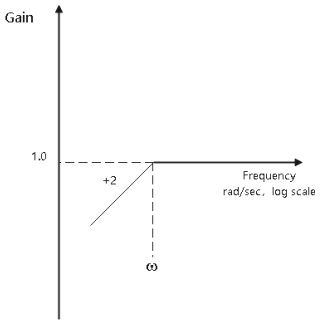
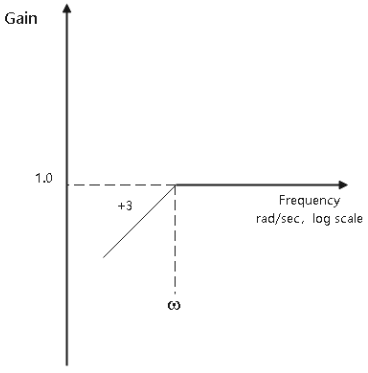
5）错误说明

6）示例

HPF 指令会减弱低于所设截止频率的信号。此指令通常用于过滤低频“噪声”或源自电气或机械源的干扰。您可以选择特定的滤波器阶数以实现不同程度的减弱效果。请注意，阶数越高，滤波指令的执行时间就越长。

以下图表展示了对于给定截止频率而言，滤波器的不同阶数所产生的影响。对于每张图表，均给出了以对数刻度表示增益和频率的理想渐近近似曲线。滤波器的实际响应趋近于这些曲线，但并不完全与这些曲线相吻合。

Filter	Graph

1st order filter	
2nd order filter	
3rd order filter	

此示例为 HPF 功能块的最小合法编程示例，仅用于展示此指令的中性文本及生成的代码。此仅为内部用途，非可测试案例。

ST 示例：

```
HPF_01.In := Velocity_Feedback;
HPF_01.WLead := Cutoff_frequency;
HPF_01.Order := 2;
HPF(HPF_01);
filtered_velocity_output := HPF_01.Out
```

积分器 INTG 指令-ST

INTG 指令实现一个积分操作。该指令设计用于在扫描速率保持恒定的任务中执行。

1）指令格式

指令	名称	梯形图表现	ST 表现
INTG	积分器	此指令不可用于梯形图中	INTG(INTG_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
INTG tag	INTEGRATOR	结构	INTG 结构

INTEGRATOR 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
Initialize	控制算法初	BOOL	变量	控制算法初始化请求。一旦 Initialize

	始化请求		立即数	置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
InitialValue	指令的初始值	REAL	变量 立即数	指令的初始值。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
IGain	积分增益乘数	REAL	变量 立即数	积分增益乘数。如果 IGain < 0，该指令会设置 IGain = 0.0，将 Status 中的相应位置位，并将 Output 保持不变。 有效值 = 0.0 到最大正浮点值 默认值 = 0.0
HighLimit	上限值	REAL	变量 立即数	上限值。这是 Out 的最大值。如果 HighLimit Less than or equal to LowLimit，指令会将 HighAlarm 和 LowAlarm 置位，将 Status 中的相应位置位，并设置 Out = LowLimit。 有效值 = LowLimit < HighLimit Less than or equal to 最大正浮点值 默认值 = 最大正浮点值
LowLimit	下限值	REAL	变量 立即数	下限值。这是 Out 的最小值。如果 HighLimit Less than or equal to

				LowLimit，指令会将 HighAlarm 和 LowAlarm 置位，将 Status 中的相应位置位，并设置 Out = LowLimit。 有效值 = 最大负浮点值 Less than or equal to LowLimit < HighLimit 默认值 = 最大负浮点值
HoldHigh	保持高位命令	BOOL	变量 立即数	保持高位命令。置位时，不允许内部积分器的值增大。 默认清零。
HoldLow	保持低位命令	REAL	变量 立即数	保持低位命令。置位时，不允许内部积分器的值减小。 默认清零。
TimingMode	选择时序执行模式	DINT	变量 立即数	选择时序执行模式。 0 = 周期性模式 1 = 过采样模式 2 = 实时采样模式 有关时序模式的详细信息，请参见“功能块属性”部分。

				有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模式的模块更新周期	DINT	变量 立即数	实时采样模式的模块更新周期 有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时戳值	DINT	变量 立即数	实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
HighAlarm	上限报警指示器	BOOL	变量	上限报警指示器。当 Out 的计算值 大于等于 HighLimit 且输出和积分器强制设为 HighLimit 时置位。

LowAlarm	下限报警指示器	BOOL	变量	下限报警指示器。当 Out 的计算值 小于等于 LowLimit 且输出和积分器强制设为 LowLimit 时置位。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
IGainInv (Status.1)	Status.1	BOOL	变量	IGain > 最大值或 IGain < 最小值。
WldInv (Status.2)	Status.2	BOOL	变量	Wld < 最小值或 Wld > 最大值。
HighLowLimsInv (Status.3)	Status.3	BOOL	变量	HighLimit <= LowLimit。
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS | DeltaT - RTSTime | > 1（0.001 秒）时置位。

RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效。
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaT (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

INTG 指令专用于恒速扫描的任务中。

当 Initialize 清零且 DeltaT > 0 时，INTG 指令将执行该控制算法。

$$\mathrm{Out} = \mathrm{IGain} \times \frac{\mathrm{In} + \mathrm{In}_{\text{n-1}}}{2} \times DeltaT + Out_{n-1}$$

只要计算出的输出值无效（NAN 或 ± INF），指令就会将 Out 设为无效值。内部参数不会更新。在每次的后续扫描过程中，都会使用上一次扫描（输出有效）的内部参数计算输出。

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

下面的示例将说明 INTG 指令的使用方法。在许多实例中，HighLimit 和 LowLimit 输入会限制积分增益分量在调节器总输出中所占的总控制百分比。另一方面，HoldHigh 和 HoldLow 输入也可用来阻止输出向正方向或负方面继续变化。HoldHigh 和 HoldLow 输入会阻止 INTG 指令在已超出受控变量限值的方向达到“积分饱和”。

$$\mathrm{Out} = \mathrm{IGain} \times \frac{\mathrm{In} + \mathrm{In}_{\mathrm{n\text{-}1}}}{2} \times DeltaT + Out_{n-1}$$

下图中，功能块的输入由 0 变为 +200 个单位。在此期间，功能块输出经积分运算后达到 2800 个单位。当 In 由 +200 个单位变为 0 个单位后，Out 保持为 2800 个单位。当 In 由 0 变为 -300 个单位时，Out 的积分值缓慢地减小为 -1400 个单位，直至 In 再变回 0。最后，随着 In 由 0 变为 +100，Out 的积分值再次达到 0，随着 Out 到达 0，In 也设置为 0，二者在此重合。

这种积分器的特点是：在函数存在任何输入时沿特定方向连续积分，并在输入为零时保持在任意水平。这正是促使调节器使用积分增益在一定时间范围内完全消除误差的原因所在。

下面的示例将说明 INTG 指令的使用方法。在许多实例中，HighLimit 和 LowLimit 输入会限制积分增益分量在调节器总输出中所占的总控制百分比。另一方面，HoldHigh 和 HoldLow 输入也可用来阻止输出向正方向或负方面继续变化。HoldHigh 和 HoldLow 输入会阻止 INTG 指令在已超出受控变量限值的方向达到“积分饱和”。

ST 示例：

```
INTG_01.In := Input_Value;
INTG_01.Initialize := Initialize_flag;
INTG_01.InitialValue := Int_Init_Val;
INTG_01.IGain := I_Gain;
INTG_01.HighLimit := Int_saturate_high;
INTG_01.LowLimit := Int_saturate_low;
INTG_01.HoldHigh :=HAlarm_Status;
INTG_01.HoldLow := LAlarm_Status;
INTG(INTG_01);
```

超前/滞后 LDLG 指令-ST

LDLG 指令为输入信号提供相位超前滞后补偿。这条指令通常用于前馈 PID 控制或过程模拟。

1）指令格式

指令	名称	梯形图表现	ST 表现
LDLG	超前/滞后	此指令不可用于梯形图中	ILDLG(LDLG_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
LDLG tag	LEAD_LAG	结构	LDLG 结构

LEAD_LAG 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值

				默认值 = 0.0
Initialize	控制算法初始化请求	BOOL	变量 立即数	控制算法初始化请求。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
Lead	交货时间以秒为单位	REAL	变量 立即数	交货时间以秒为单位。设置引线= 0.0 禁用引线控制算法。如果 Lead < 0.0，则指令在 Status 中设置适当的位，并限制 Lead 为 0.0。如果先导>为最大正浮点数,则该指令在 Status 中设置相应的位。 Valid =任何大于或等于 0.0 的浮点数 默认值= 0.0
Lag	延迟时间	REAL	变量 立即数	延迟时间（以秒为单位）。最小延迟时间为 DeltaT/2。如果 Lag < DeltaT/2，则该指令在 Status 中设置相应的位，并将 Lag 限制为 DeltaT/2。如果 Lag >最大正浮点数，则该指令在 Status 中设置相应的位。 有效=任何大于或等于 DeltaT/2 的浮点数 默认值= 0.0

Gain	过程增益倍增器	REAL	变量 立即数	过程增益倍增器。该值允许模拟过程增益。In 信号乘以这个值。I = (In x Gain) +偏置 Valid =任意浮点数 默认值= 1.0
Bias	进程偏移量级别	REAL	变量 立即数	进程偏移量级别。该值允许模拟环境条件。该值与 In 乘以 Gain 的结果相加。I = (In x Gain) +偏置 Valid =任意浮点数 默认值= 0.0
TimingMode	选择时序执行模式	DINT	变量 立即数	选择时序执行模式。 0 = 周期性模式 1 = 过采样模式 2 = 实时采样模式 有关时序模式的详细信息，请参见“功能块属性”部分。 有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模	DINT	变量	实时采样模式的模块更新周期

	式的模块更新周期		立即数	有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时戳值	DINT	变量 立即数	实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
LeadInv (Status.1)	Status.1	BOOL	变量	Lead<最小值或 Lead>最大值。。

LagInv (Status.2)	Status.2	BOOL	变量	Lag<最小值或 Lag>最大值。。
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS | DeltaT - RTSTime | > 1（0.001 秒）时 置位。
RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效。
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaTInv (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

LDLG 指令支持一个引线和滞后串联。该指令还允许配置增益和偏置因素。LDLG 指令设计用于在扫描速率保持恒定的任务中执行。

LDLG 指令使用这个等式：

$$H(s)=\left[\frac{1+Lead\times s}{1+Lag\times s}\right]$$

使用这些参数限制：

参数	限制
Lead	LowLimit = 0.0 HighLimit = maximum positive float
Lag	LowLimit = DeltaT/2 (DeltaT is in seconds) HighLimit = maximum positive float

每当为输出计算的值无效、NAN 或正负号 INF 时，该指令设置 Out =无效值并设置 Math overflow 状态标志。当为输出计算的值变为有效时，指令初始化内部参数并设置 Out = (In x Gain) + Bias。

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

本例中的 LDLG 指令将一阶延迟添加到模拟过程中。还可以在 LDLG 指令上输入增益来模拟过程增益，也可以输入偏置来模拟环境条件。

ST 示例：

```
DEDT_01.In := SimulatedLoop.CVEU;
DEDT(DEDT_01,DEDT_01_array);
LDLG_01.In := DEDT_01.Out;
LDLG(LDLG_01);
SimulatedLoop.PV := LDLG_01.Out;
PIDE(SimulatedLoop);
```

低通滤波器 LPF 指令-ST

LPF 指令提供一个滤波器来衰减高于截止频率的输入频率。

1）指令格式

指令	名称	梯形图表现	ST 表现
LPF	低通滤波器	此指令不可用于梯形图中	INTG(INTG_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
LPF tag	FILTER_LOW_PASS	结构	LPF 结构

LPF 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值

				默认值 = 0.0
Initialize	控制算法初始化请求	BOOL	变量 立即数	控制算法初始化请求。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
WLag	滞后频率	REAL	变量 立即数	滞后频率（单位：弧度/秒）。若 WLag 小于最小值或大于最大值，则指令会在状态寄存器中设置相应的位，并限制 WLag 的取值范围。 有效值 = 参见下文的描述部分以获取有效范围 默认值 = 0.0
Order	滤波器的阶数	REAL	变量 立即数	滤波器的阶数。阶数控制截止频率的锐度。如果阶数无效，则指令会在状态寄存器中设置相应的位，并将阶数设为 1。 有效值 = 1 到 3 默认值 = 1
TimingMode	选择时序执行模式	DINT	变量 立即数	选择时序执行模式。 0 = 周期性模式 1 = 过采样模式 2 = 实时采样模式

				有关时序模式的详细信息，请参见“功能块属性”部分。 有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模式的模块更新周期	DINT	变量 立即数	实时采样模式的模块更新周期 有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时戳值	DINT	变量 立即数	实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。

Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
WLagInv (Status.1)	Status.1	BOOL	变量	WLag > 最大值或 WLag < 最小值。
OrderInv(Status.2)	Status.2	BOOL	变量	无效命令值
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS | DeltaT - RTSTime | > 1（0.001 秒）时置位。
RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效。
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaT (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

LPF 指令通过 Order 参数来控制截止频率的陡峭程度。LPF 指令旨在在一个扫描速

率保持恒定的任务中执行。

条件/状态	该指令使用了这样的拉普拉斯传递函数：
Order = 1	$\frac{\omega}{s+\omega}$
Order = 2	$\frac{\omega^2}{s^2+\sqrt{2}\times s\times\omega+\omega^2}$
Order = 3	$\frac{\mathrm{w}^3}{s^3+\left(2\times s^2\times w\right)+\left(2\times s\ \times w^2\right)\times w^3}$

按照这些参数限制（其中 DeltaT 以秒为单位）：

参数	限制
WLag 第一阶 下限限值	$\frac{0.0000001}{DeltaT}$
WLag 第二阶 下限限值	$\frac{0.00001}{DeltaT}$
WLag 第三阶 下限限值	$\frac{0.001}{DeltaT}$
高限限值	$\frac{0.7\pi}{DeltaT}$

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

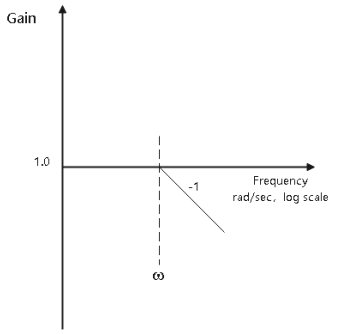
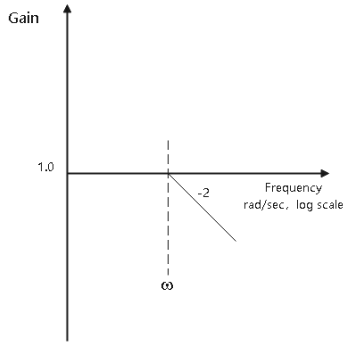
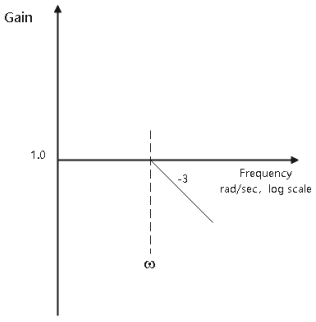
5）错误说明

6）示例

LPF 指令会衰减高于所设截止频率的信号。此指令通常用于滤除源自电气或机械源的高频“噪声”或干扰。您可以选择特定的滤波器阶数以实现不同程度的衰减。请注意，阶数越高，该指令的执行时间就越长。

以下图表展示了对于给定截止频率而言，滤波器的不同阶数所产生的影响。对于每张图表，均给出了以对数刻度表示增益和频率的理想渐近近似曲线。滤波器的实际响应趋近于这些曲线，但并不完全与这些曲线相吻合。

Filter	Graph
1st order filter	

	
2nd order filter	
3rd order filter	

此示例为 LPF 功能块的最小合法编程示例，仅用于展示此指令的中性文本及生成的代码。此仅为内部用途，非可测试案例。

ST 示例：

```
LPF_01.In := Velocity_Feedback;

LPF_01.WLag := Cutoff_frequency;
```

```
LPF(LPF_01);

filtered_velocity_output := LPF_01.Out;
```

比例+积分 PI 指令-ST

PI 指令提供了两种操作方法。第一种方法遵循传统的 PI 算法，因为比例和积分增益在输入信号(误差)的范围内保持恒定。第二种方法使用非线性算法，其中比例增益和积分增益在输入信号的范围内变化。输入信号是过程的设定值和 反馈之间的偏差。

1）指令格式

指令	名称	梯形图表现	ST 表现
PI	比例+积分	此指令不可用于梯形图中	PI(PI_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
PI tag	PROP_INT	结构	PI 结构

PROP_INT 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位

IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
Initialize	控制算法初始化请求	BOOL	变量 立即数	控制算法初始化请求。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
InitialValue	指令的初始值	REAL	变量 立即数	指令的初始值。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
Kp	比例增益	REAL	变量 立即数	比例增益。这会影响比例控制算法和积分控制算法的计算值。如果无效，则指令将 Kp 钳位在限制处，并在 Status 中设置适当的位。 有效 = 任何浮点数 > 0.0 默认值 = 最小正浮点数
Wld	以弧度/秒为单位的超前频率	REAL	变量 立即数	以弧度/秒为单位的超前频率。这会影响积分控制算法的计算值。如果无效，则指令将限制在限制处并在 Status 中设置适当的位。 有效 = 请参阅下面的 描述 部分，了解

				有效范围 默认值 = 0.0
HighLimit	上限值	REAL	变量 立即数	上限值。这是 Out 的最大值。如果 HighLimit 小于或等于 LowLimit，则指令设置 HighAlarm 和 LowAlarm，在 Status 中设置适当的位，并设置 Out = LowLimit。 有效 = LowLimit < HighLimit 小于或等于 最大正浮点数 默认值 = 最大正浮点数
LowLimit	下限值	REAL	变量 立即数	下限值。这是 Out 的最小值。如果 HighLimit 小于或等于 LowLimit，则指令设置 HighAlarm 和 LowAlarm，在 Status 中设置适当的位，并设置 Out = LowLimit。 有效 = 最大负浮点数 小于或等于 LowLimit < HighLimit 默认值 = 最大负浮点数
HoldHigh	保持高位命令	BOOL	变量 立即数	保持高位命令。设置后，内部积分器的值不允许增加。 默认被清除。

HoldLow	保持低位命令	BOOL	变量 立即数	保持低位命令。设置后，内部积分器的值不允许减小。 默认被清除。
ShapeKpPlus	正 Kp 整形增益乘数	REAL	变量 立即数	正 Kp 整形增益乘数。当 In 为大于或等于 0 时使用。如果无效，则指令将 ShapeKpPlus 限制在限制处，并在 Status 中设置适当的位。清除 NonLinearMode 时不使用。 有效 = 0.1 到 10.0 默认值 = 1.0
ShapeKpMinus	负 Kp 整形增益乘数	REAL	变量 立即数	负 Kp 整形增益乘数。当 In < 0 时使用。如果无效，则指令将 ShapeKpMinus 限制在限制处，并在 Status 中设置适当的位。清除 NonLinearMode 时不使用。 有效 = 0.1 到 10.0 默认值 = 1.0
KpInRange	比例增益整形范围	REAL	变量 立即数	比例增益整形范围。定义 In（误差）的范围，在该范围内，比例增益作为 |在 |/ KpInRange 的时间 | 在 |> KpInRange 中，指令使用输入的 Kp 整形增益 x （In - KpInRange） 计算

				比例误差的变化。如果无效，则指令将 KpInRange 限制在限制处，并在 Status 中设置适当的位。清除 NonLinearMode 时不使用。 有效 = 任何浮点数 > 0.0 默认值 = 最大正浮点数
ShapeWldPlus	正将塑造增益乘数	REAL	变量 立即数	正将塑造增益乘数。当 In 为大于或等于 0 时使用。如果无效，则指令将 ShapeWldPlus 限制在限制处，并在 Status 中设置适当的位。清除 NonLinearMode 时不使用。 有效 = 0.0 到 10.0 默认值 = 1.0
ShapeWldMinus	负数将塑造增益乘数	REAL	变量 立即数	负数将塑造增益乘数。当 In < 0 时使用。如果无效，则指令将 ShapeWldMinus 限制在限制处，并在 Status 中设置适当的位。清除 NonLinearMode 时不使用。 有效 = 0.0 到 10.0 默认值 = 1.0
WldlnRange	积分增益整形范围	REAL	变量 立即数	积分增益整形范围。定义 In（误差）的范围，积分增益随 |在 |/ WldInRange

				的 SST 时间 |在 |> WldInRange 中，指令在计算积分误差时将 In 限制为 WldInRange。如果无效，则指令将 WldInRange 限制在限制处，并在 Status 中设置适当的位。清除 NonLinearMode 时不使用。 有效 = 任何浮点数 > 0.0 默认值 = 最大正浮点数
NonLinearMode	使能非线性增益模式	BOOL	变量 立即数	使能非线性增益模式。设置后，该指令使用 ParabolicLinear 选择的非线性增益模式来计算实际的比例和积分增益。当清零时，该指令将禁用非线性增益模式，并使用 Kp 和 would 值作为比例和积分增益。 默认被清除。
ParabolicLinear	选择非线性增益模式	BOOL	变量 立即数	选择非线性增益模式。这些模式是线性的或抛物线的。设置后，该指令使用 y=a * x 的抛物线增益方法 2+ b 计算实际的比例和积分增益。如果清除，则指令使用 y=a * x + b 的线性增益方法。 默认被清除。

TimingMode	选择 timing execution mode	DINT	变量 立即数	选择 timing execution mode。 0 = 周期模式 1 = 过采样模式 2 = 实时采样模式 有关计时模式的更多信息，请参阅 Function Block Attributes。 有效 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	oversample 模式的执行时间。 有效 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模式	DINT	变量 立即数	实时采样模式的模块更新周期 有效 = 1 至 32,767ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时间戳值	DINT	变量 立即数	实时采样模式的模块时间戳值。 有效 = 0 到 32,767 毫秒 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
HighAlarm	最大限值警报指示器	BOOL	变量	最大限值警报指示器。当 Out 大于或等于 HighLimit 以及输出和积分器的计算值限制为 HighLimit 时设置。
LowAlarm	最小限值警报指示器	BOOL	变量	最小限值警报指示器。当 Out 小于或等于 LowLimit 和 output 和 integrator 的计算值限制为 LowLimit 时设置。
DeltaT	更新之间经过的时间	REAL	变量	更新之间经过的时间。这是控制算法用于计算流程输出的运行时间（以秒为单位）。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
KpInv (Status.1)	Status.1	BOOL	变量	Kp <最小值或 Kp >最大值。
WldInv (Status.2)	Status.2	BOOL	变量	将<最小值或将>最大值。

HighLowLimsInv (Status.3)	Status.3	BOOL	变量	HighLimit 小于或等于 LowLimit 的 Limit。
ShapeKpPlusInv (Status.4)	Status.4	BOOL	变量	ShapeKpPlus< 最小值或 ShapeKpPlus >最大值。
ShapeKpMinusInv (Status.5)	Status.5	BOOL	变量	ShapeKpMinus < 最小值或 ShapeKpMinus >最大值。
KpInRangeInv (Status.6)	Status.6	BOOL	变量	KpInRange < 最小值，或者 KpInRange >最大值。
ShapeWldPlusInv (Status.7)	Status.7	BOOL	变量	ShapeWldPlus < 最小值或 ShapeWldPlus >最大值。
ShapeWldMinusInv (Status.8)	Status.8	BOOL	变量	ShapeWldMinus < 最小值或 ShapeWldMinus >最大值。
WldInRangeInv (Status.9)	Status.9	BOOL	变量	WldInRange < 最小值，或者 WldInRange >最大值。
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。 有关计时模式的更多信息，请参阅 Function Block Attributes。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅在实时采样模式下使用。设置 ABS |DeltaT - RTSTime |> 1（.001 秒）。
RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效。

RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaT (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

PI 指令使用 PI 算法的位置形式。这意味着增益项直接应用于 input 信号，而不是 input 信号的变化。PI 指令旨在在扫描速率保持不变的任务中执行。

在非线性算法中，比例增益和积分增益随着输入信号幅度的变化而变化。PI 指令支持两种非线性增益模式：线性和抛物线。在线性算法中，增益随着输入幅度的变化而线性变化。在抛物线算法中，增益根据抛物线曲线随着输入大小的变化而变化。

PI 指令使用以下公式计算 Out：

$$Kp \times \frac{s + Wld}{s}$$

每当为输出计算的值无效、NAN 或 加号或减号 INF 时，指令就会设置 Out = 无效值并设置数学溢出状态标志。内部参数不会更新。在每次后续扫描中，当输出有效时，将使用上次扫描的内部参数计算输出。

4）注意事项

5）错误说明

6）示例

结构化文本：

```
Reference_Select.In1 := Master_Machine_Ref;

Reference_Select.Select1 ： = Master_Machine_Select;

Reference_Select.In2 ： = Section_Jog;

Reference_Select.选择 2 ： = Jog_Select;

SSUM（Reference_Select）；


S_Curve.输入 ： = Reference_Select.输出;

S_Curve.加速度 ： = accel_rate;

S_Curve.减速 ： = accel_rate;

SCRV（S_Curve）；


PMUL_01.In := Resolver_Feedback;

PMUL_01.字大小 ： = 12;

PMUL_01.乘数 ： = 100000;

PMUL（PMUL_01）；


Speed_Feedback ： = PMUL_01.Out;

Velocity_Error ： = S_Curve.出 - Speed_Feedback;


PI_01.In ： = Velocity_Error;

PI_01.初始化 ： = Enable_Regulator;
```

PI_01.Kp ： = Velocity_Proportional_Gain;

PI_01.Wld ： = Velocity_Integral_Gain;

PI（PI_01）;

Torque_Reference ： = PI_01.Out;

脉冲乘法器 PMUL 指令-ST

PMUL 指令通过计算从一次扫描到下一次扫描的输入变化，提供了从位置输入模块(如解析器或编码器反馈模块)到数字系统的接口。通过选择特定的字长，可以配置 PMUL 指令以连续和线性的方式区分滚动边界。

1）指令格式

指令	名称	梯形图表现	ST 表现
IPMUL	脉冲乘法器	此指令不可用于梯形图中	PMUL(PMUL_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
INTG tag	INTEGRATOR	结构	INTG 结构

PULSE_MULTIPLIER 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值

				默认值 = 0.0
Initialize	控制算法初始化请求	BOOL	变量 立即数	控制算法初始化请求。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
InitialValue	指令的初始值	REAL	变量 立即数	指令的初始值。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
Mode	模式输入	BOOL	变量 立即数	模式输入。设置为启用相对模式。清除以启用绝对模式。 设置为默认值。
WordSize	上限值	DINT	变量 立即数	以位为单位的字长。指定在相对模式下计算（Inn-Inn-1）时使用的位数。 绝对模式下未使用 WordSize。 当 in 的变化量大于 1/2 * 2（Wordsize-1）时，Out 改变符号。 当 WordSize 无效时，Out 被保持，并且指令在 Status 中设置适当的位。
Multiplier	下限值	REAL	变量 立即数	乘数。将此值除以 100,000 以控制 In 与 Out 的比率。如果无效，则该指令限制

				该值并在 Status 中设置适当的位。 Valid = -1,000,000 到 1,000,000 默认值= 100,000

输出变量(Output)

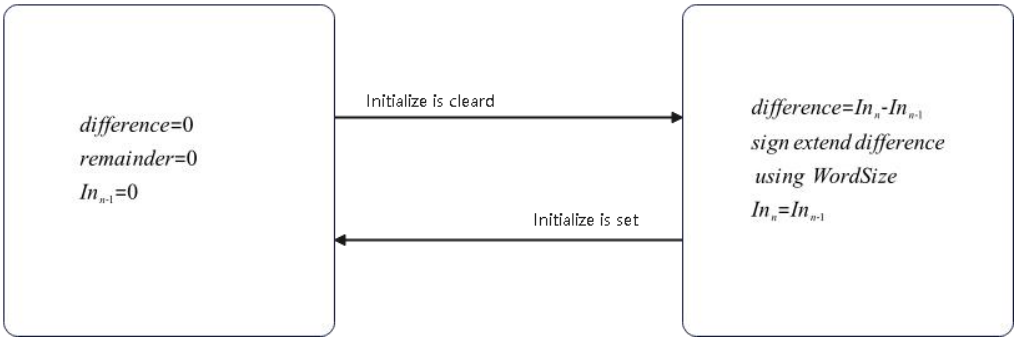
参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
WordSizeInv (Status.1)	Status.1	BOOL	变量	无效的 WordSize 值。
OutOverflow (Status.2)	Status.2	BOOL	变量	内部输出计算溢出。
LostPrecision (Status.3)	Status.3	BOOL	变量	输出< - 2^{24} 或输出> 2^{24} 。当该指令将整数转换为实数时，由于 real 数据类型被限制为 2^{24} ，如果结果大于| 2^{24} |，则会丢失数据。

MultiplierInv (Status.4)	Status.4	BOOL	变量	无效的 MultiplierInv 值。

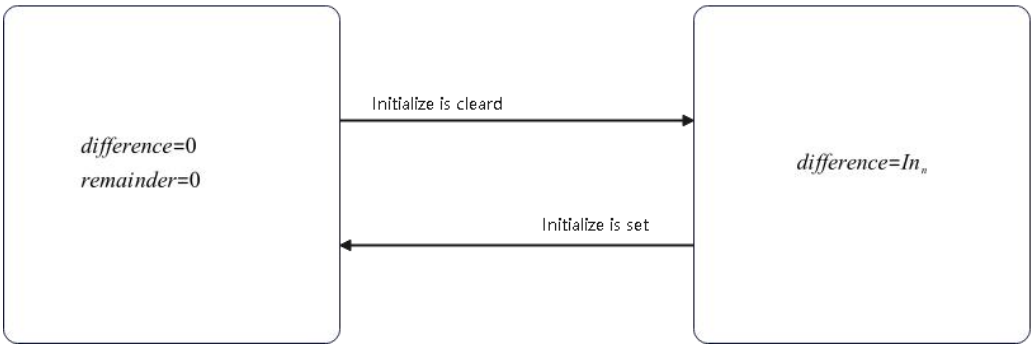
3）功能说明

PMUL 指令以相对或绝对模式运行。

在相对模式下，指令的输出是输入从扫描到扫描的微分，乘以（Multiplier/100,000）。在相对模式下，该指令在扫描中保存除法运算后的余数，并在下一次扫描时将其加回。通过这种方式，位置信息不会在操作过程中丢失。



在绝对模式下，指令可以缩放输入，例如位置，而不会丢失从一次扫描到下一次扫描的任何信息。



计算输出和余数

PMUL 指令使用这些方程在相对或绝对模式下计算 Out：

Ans = （(DiffInput x Multiplier) + INT_Remainder）

INT_Out = Ans / 100,000

INT_Remainder= Ans - （INT_Out * 100,000）

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

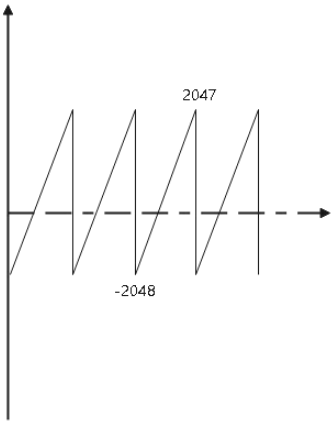
5）错误说明

6）示例

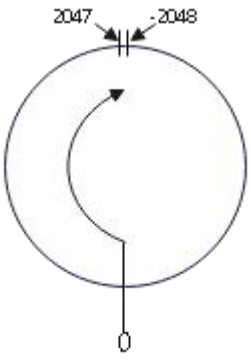
Example 1

PMUL 指令最常见的用法是相对操作模式。在这种模式下，PMUL 指令有几个用途。首先，在相对模式下，PMUL 指令区分它在每次扫描的输入处接收到的信息。当接收到数据时，指令输出从一次扫描到下一次扫描的输入差值。这意味着如果在扫描“n”时 In = 500，然后在扫描“n+1”时 In = 600，在扫描“n+1”时 Out = 100。

其次，在这种操作模式下，PMUL 指令还补偿来自反馈模块的二进制数据的“翻转”值。例如，解析器反馈模块可能具有 12 位分辨率，表示为二进制值，带符号，范围从-2048 到 2047。对于来自反馈模块的原始数据，反馈装置的旋转可以表示为：



在本例中，当反馈数据的值从 2047 移动到-2048 时，位置的有效变化相当于位置上的 4095 个计数的跳跃。然而，实际上，就解析器反馈装置的旋转而言，这种位置变化仅为 1 / 4096。通过了解从反馈模块输入的数据的真实字长，PMUL 指令以旋转方式查看数据，如下图所示：在本例中，当反馈数据的值从 2047 移动到-2048 时，位置的有效变化相当于位置上跳了 4095 个计数。然而，实际上，就解析器反馈装置的旋转而言，这种位置变化仅为 1 / 4096。通过了解从反馈模块输入的数据的真实字长，PMUL 指令以旋转方式查看数据，如下图所示：



通过知道输入到该块的数据的字长，PMUL 指令在块的输入从 2047 移动到-2048 而不是数学计算的 4095 时区分 1 个计数的输出。

请注意

提示：当应用此块时，重要的是要注意，如果要正确区分旋转方向，那么从一次扫描到下一次扫描，反馈数据的变化不应超过单词大小的一半。

在上面的例子中，如果反馈设备沿顺时针方向移动，在扫描“a”时读取 0，然后扫描“B”时读取-2000，则实际位置变化相当于顺时针方向上的+2096 计数。然而，由于这两个值大于单词大小的一半（或大于物理设备旋转的一半），PMUL 指令计算出反馈设备朝相反方向旋转并返回值-2000 而不是+2096。

脉冲倍增器块的第三个属性是，它保留了从一次扫描到下一次扫描的任何余数的小数分量，这些余数是 multiplier /100,000 缩放因子的结果。当块的每次执行完成时，前一次扫描的剩余部分被加回当前值的总数中，这样所有计数或“脉冲”最终都被计算在内，并且系统中没有数据丢失。块的输出 Out 总是产生一个浮点数据类型的整数。

ST 示例：

```
MUL_02.In := Position_feedback;
PMUL_02.Initalize := Initialize_Position;
PMUL_02.WordSize := 12;
PMUL_02.Multiplier := 25000;
PMUL(PMUL_02);
```

```
UPDN_02.Initialize := Initialize_Position;
UPDN_02.InPlus := PMUL_02.Out;
UPDN(UPDN_02);

Total_Position := UPDN_02.Out;
```

Example 2

在此电子线轴应用中，电机 A 的反馈作为电机 B 需要遵循的主参考。电机 A 的反馈别名为“Position_feedback”。电机 B 的反馈别名为“Follower_Position”。由于两个指令的乘数为 1/4 的比率，电机 B 需要在电机 a 每转 4 圈时旋转一次，以保持 UPDN 累加器中的累积值为零。UPDN 指令输出上除零以外的任何值都被视为 Position_error，可以调节并驱动回电机 B，以保持两个电机之间的锁相。

ST 示例：

```
PMUL_02.In := Position_feedback;
PMUL_02.Initalize := Initialize_Position;
PMUL_02.WordSize := 12;
PMUL_02.Multiplier := 25000;
PMUL(PMUL_02);

PMUL_03.In := Follower_Position;
PMUL_03.Initalize := Initialize_Position;
```

```
PMUL_03.WordSize := 12;

PMUL_03.Multiplier := 100000;

PMUL(PMUL_03);

UPDN_02.Initialize := Initialize_Position;

UPDN_02.InPlus := PMUL_02.Out;

UPDN_02.InMinus := PMUL_03.Out;

UPDN(UPDN_02);


Position_error := UPDN_02.Out;
```

标定 SCL 指令-ST

SCL 指令将未缩放的输入值转换为工程单位的浮点值。

1）指令格式

指令	名称	梯形图表现	ST 表现
SCL	标定	此指令不可用于梯形图中	SCL(SCL_tag)

2）相关变量

结构化文本

变量	数据类型	格式	说明
SCL tag	SCALE	结构	SCL 结构

SCL 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0

InRawMax	指令的输入可达到的最大值	REAL	变量 立即数	指令的输入可达到的最大值。如果 InRawMax Less than or equal to InRawMin,则该指令会将 Status 中的相应位置位，并停止更新输出。 有效值 = InRawMax > InRawMin 默认值 = 0.0
InRawMin	指令的输入可达到的最小值	REAL	变量 立即数	指令的输入可达到的最小值。如果 InRawMin Greater than or equal to InRawMax，则指令会将 Status 中的相应位置位，并停止更新输出。 有效值 = InRawMin < InRawMax 默认值 = 0.0
InEUMax	对应于 InRawMax 的输入标定值	REAL	变量 立即数	对应于 InRawMax 的输入标定值。 有效值 = 任意实数值 默认值 = 0.0
Limit	限制	BOOL	变量 立即数	限制选择器。如果为真，会将 Out 限制在 InEUMin 与 InEUMax 之间。 默认值为假。

输出变量(Output)

参数(英文)	数据类型	格式	说明

EnableOut	BOOL	变量	指示指令是否处于启用状态。
Out	REAL	变量	表示模拟量输入标定值的输出。 有效值 = 任意实数值 默认值 = InEUMin
MaxAlarm	BOOL	变量	超出最大输入报警指示器。当 In> InRawMax 时，该值设置为真。
MinAlarm	BOOL	变量	低于最小输入报警指示器。当 In < InRawMin 时，该值设置为真。
状态 (Status)	BOOL	变量	功能块的状态。
InstructFault (Status.0)	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
InRawRangeInv (Status.1)	BOOL	变量	InRawMin Greater than or equal to InRawMax。

3）功能说明

SCL 指令用于不支持标定为全分辨率浮点值的模拟量输入模块。

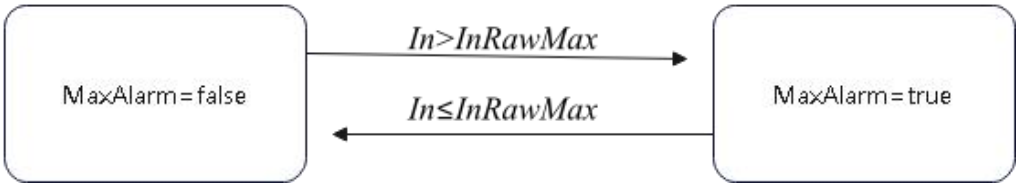
例如，1771-IFE 模块是一个仅支持以整数值进行标定的 12 位模拟量输入模块。如果使用 1771-IFE 模块读取 0-100 加仑/分钟（gpm）的流量，通常不会将模块标定为 0-100，否则会限制该模块的分辨率。相反，应使用 SCL 指令并将模块配置为返回未标定的（0-4095）值，其中 SCL 指令会将值转换为 0-100 gpm（浮点值）而不会损失分辨率。然后，此标定后的值可用作其他指令的输入。

SCL 指令采用以下算法将未标定的输入转换为标定值：

$$Out = (In - InRawMin) \times (\frac{InEUMax - InEUMin}{InRawMasx - InRawMin}) + InEUMin$$

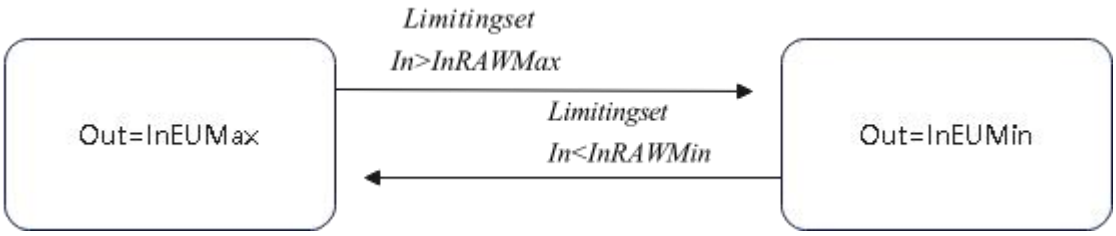
报警

指令计算出 Out 后，将按照以下方式确定 MaxAlarm 和 MinAlarm 的值：



限制

将 Limiting 置位后，会对 Out 进行限制。当 In > InRawMax 时，该指令会设置 Out = InEUMax。当 In < InRawMin 时，该指令会设置 Out = InEUMin。



结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

SCL 指令通常用于不支持板载标定为浮点工程单位的模拟量输入模块。在此示例中，SCL 指令对来自 1771-IFE 模块的模拟量输入进行标定。该指令将结果存储在 Out 中，供 ALM 指令使用。

ST 示例：

```
SCL_01.In := Input0From1771IFE;
SCL(SCL_01);
ALM_01.In := SCL_01.Out;
ALM(ALM_01);
```

S 曲线 SCRV 指令-ST

SCRV 指令执行带有附加跳变率的斜坡函数。跳变速率是用于将输出变为输入的速率的最大变 化率。

1）指令格式

指令	名称	梯形图表现	ST 表现
SCRV	S 曲线	此指令不可用于梯形图中	SCRV(SCRV_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
SCRV tag	S_CURVE	结构	SCRV 结构

S_CURVE 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0

Initialize	控制算法初始化请求	BOOL	变量 立即数	控制算法初始化请求。一旦 Initialize 置位，Output = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
InitialValue	S 曲线的初始值	REAL	变量 立即数	S 曲线的初始值。当 Initialize 置位时，Out = InitialValue。 有效值 = 任意浮点值 默认值 = 0.0
AbsAlgRamp	斜坡类型	BOOL	变量 立即数	斜坡类型。置位时，指令将用作绝对值斜坡。清零时，指令将用作代数斜坡。 默认置位
AccelRate	加速度	REAL	变量 立即数	加速度，单位为输入单位/秒^2。零值可阻止 Out 加速。当 AccelRate < 0 时，该指令会假定 AccelRate = 0，并将 Status 中的相应位置位。 有效值 = 0.0 到最大正浮点值 默认值 = 0.0
DecelRate	减速度	REAL	变量 立即数	减速度，单位为输入单位/秒^2。零值可阻止 Out 减速。当 DecelRate < 0 时，该指令会假定 DecelRate = 0，并将 Status 中的相应位置位。 有效值 = 0.0 到最大正浮点值

				默认值 = 0.0
JerkRate	保持高位命令	REAL	变量 立即数	减速跃度，单位为输入单位/秒^2。零值可阻止 Out 减速。当 DecelRate < 0 时，该指令会假定 DecelRate = 0，并将 Status 中的相应位置位。 有效值 = 0.0 到最大正浮点值 默认值 = 0.0
HoldMode	S 曲线保持模式参数	BOOL	变量 立即数	S 曲线保持模式参数。该参数与 HoldEnable 参数一起使用。如果在 HoldEnable 置位且 Rate = 0 时 HoldMode 置位，指令将使 Out 保持不变。在这种情况下，只要 HoldEnable 置位指令就会使 Out 保持不变，JerkRate 会被忽略，而且 Out 曲线中将生成“拐点”。如果 HoldMode 在 HoldEnable 置位时清零，指令将使用 JerkRate 将 Out 变为常数值。当 Rate = 0 时，Out 保持不变。一旦 HoldEnable 置位，请勿更改 HoldMode，因为指令将忽略此更改。 默认清零。

HoldEnable	S 曲线保持启用参数	BOOL	变量 立即数	S 曲线保持启用参数。置位时，Out 保持不变。清零时，Out 从当前值开始变化，直到等于 In。 默认清零。
TimingMode	选择时序执行模式	DINT	变量 立即数	选择时序执行模式。 0 = 周期性模式 1 = 过采样模式 2 = 实时采样模式 有关时序模式的详细信息，请参见“功能块属性”部分。 有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL	变量 立即数	过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0
RTSTime	实时采样模式的模块更新周期	DINT	变量 立即数	实时采样模式的模块更新周期 有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时戳值	DINT	变量 立即数	实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
S_Mode	S_Mode 输出	BOOL	变量	S_Mode 输出。当 (Jerk * DeltaT) Less than or equal to Rate 且 Rate < Accel 或 Decel 时，S_Mode 置位。否则，S_Mode 清零。
Out	指令输出	REAL	变量	S 曲线指令的输出。该输出的数学状态标志置位。
Rate	Out 的内部变化	REAL	变量	Out 的内部变化，以单位/秒表示。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
AcceRateInv (Status.1)	Status.1	BOOL	变量	AccelRate 为负数。

DecelRateInv (Status.2)	Status.2	BOOL	变量	DecelRate 为负数。
JerkRateInv (Status.3)	Status.3	BOOL	变量	JerkRate 为负数。
TimingModeInv (Status.27)	Status.27	BOOL	变量	TimingMode 无效。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS | DeltaT - RTSTime | > 1（0.001 秒）时置位。
RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效。
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaT (Status.31)	Status.31	BOOL	变量	DeltaT 值无效。

3）功能说明

SCRV 指令的主要要求是确保变化率的变化决不超过指定的急动度。

用户可以将 SCRV 指令配置为针对阶跃输入生成 S 曲线轨迹或斜坡曲线。

S 曲线轨迹

要生成 S 曲线轨迹，可对 JerkRate 进行设置，使 (JerkRate * DeltaT) < AccelRate 和/或 DecelRate。

在 S 曲线轨迹模式下，SCRV 指令可确保变化率的变化决不超过指定的 JerkRate。用于生成 S 曲线轨迹的算法旨在针对阶跃输入生成一条光滑且对称的 S 曲线。为简化此过程，算法中引入了 Out 的梯形积分。因此，曲线各个部分的 Rate 变化都将小于 JerkRate。

当输入发生阶跃变化时，变化率会增加到设定的 AccelRate 或 DecelRate。AccelRate 或 DecelRate 保持不变，直到为了在变化率到达零时使输出等于输入而必须降低变化率。

某些情况下，根据加速度、减速度和急动度值的不同，在变化率必须按急动度开始降低之前，可能还没有达到加速度或减速度。

对于非常小的阶跃变化，SCRV 指令不会尝试生成“S”曲线。在此模式下，将输出整个阶跃，并且 Rate 会反映输出的变化。如果 Out = In 并且 In 的下一个阶跃变化能够以小于或等于所设定 JerkRate 的变化率输出，则会出现这种情况。

SCRV 指令支持代数斜坡和绝对值斜坡。对于代数斜坡，加速条件由朝正向增大的输入定义，而减速条件则由朝负向减小的输入定义。对于绝对值斜坡，加速条件由逐渐远离零的输入定义，而减速条件则由逐渐接近零的输入定义。

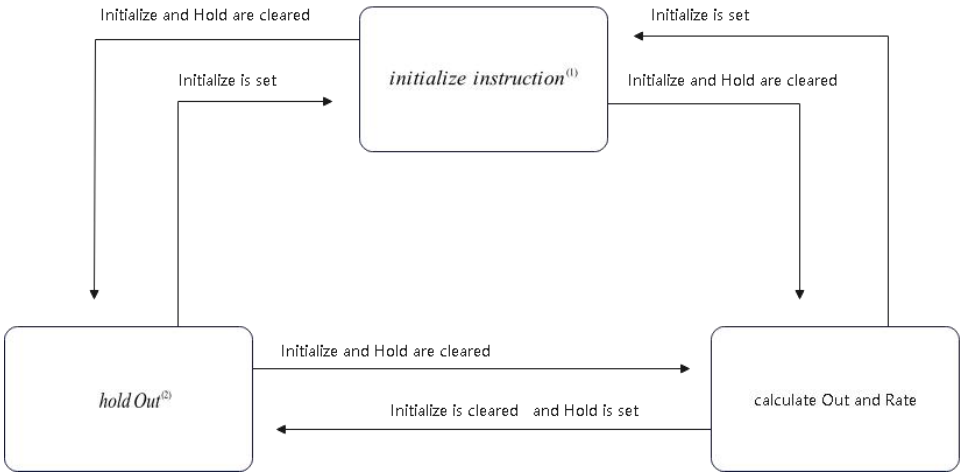
斜坡曲线

要生成斜坡曲线，可对 JerkRate 进行设置，使 (JerkRate * DeltaT) Less than or equal to AccelRate 和/或 DecelRate。

在斜坡曲线模式下，SCRV 指令始终生成与设定的 AccelRate 或 DecelRate 相等的变化率，直到为了到达终点 Out 和 In 的差值需要小于 AccelRate 或 DecelRate。

HoldMode = 0 时的工作方式与 HoldMode = 1 相同。当 HoldEnable 置位时，Out 立即保持不变，Rate 将变为零。

下图展示指令修改 Out 的方式。



(1) 当 Initialize 置位时，指令将设置以下各项：

Outn = InitialValue

Outn-1 = Outn

Raten = 0

Raten-1 = 0

(2) 当 HoldMode 清零时，Out 朝 In 变化并且 HoldEnable 置位，变化率开始按急动度逐渐减小为零。由于 JerkRate 的作用，Out 会保持在变化率达到零时对应的值。当 Out 最终保持不变时，其值与 HoldEnable 置位时的瞬时值不同。

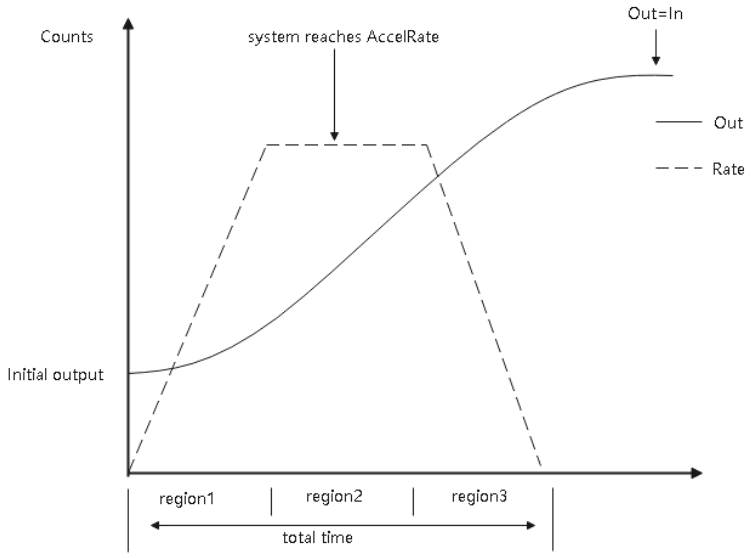
当 HoldMode 置位时，Out 朝 In 变化，HoldEnable 置位，变化率立即设置为 0。Out 保持在 HoldEnable 置位时对应的值。

在转换期间减小 JerkRate 可能会使 Out 超出 In。如果出现这种情况，是因强制使用输入的 JerkRate 引起的。若在调谐过程中以小步长减小 JerkRate，或者在 Out = In（非转换期间）时更改 JerkRate，可避免出现这种情况。

Out 达到与输入变化相等所需的时间与 AccelRate、JerkRate 以及 In 和 Out 之间差值相关。

计算输出值和变化率值

从初始值转换为最终值的过程中，Out 共经过三个区域。在区域 1 和区域 3 中，Out 的变化率取决于 JerkRate。在区域 2 中，Out 的变化率取决于 AccelRate 或 DecelRate。



各个区域的 Out 值计算如下：

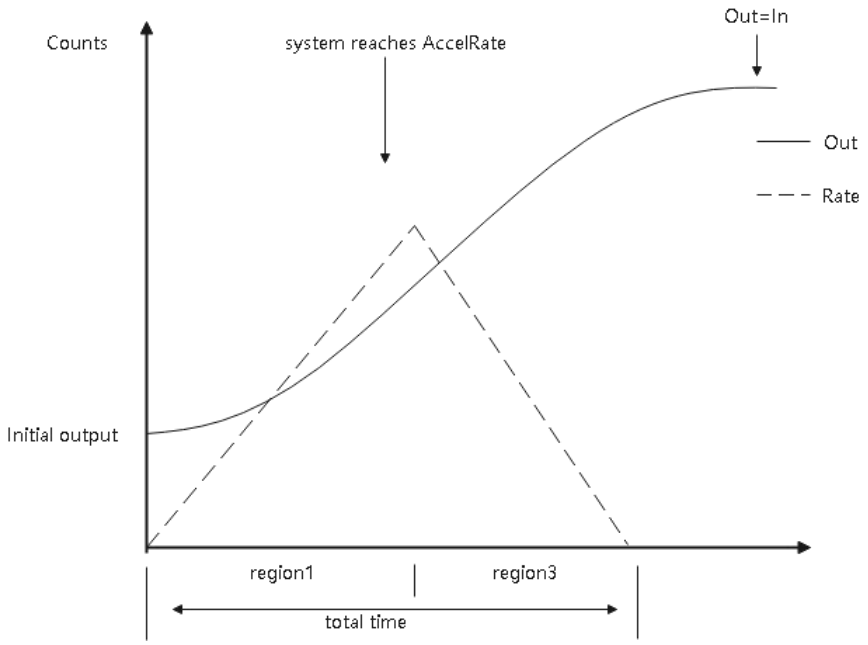
$$TotalTime = \frac{FinalOutput - InitialOutput}{AccelRate} + \frac{AccelRate}{JerjRate}$$

各个区域对应的公式如下

区域	公式
区域 1	$Time_1 = \frac{AccelRata}{JerkRate}$ $Y(Time) = InitialOutput + \frac{1}{2}(JerkRate) \times Time^2$
区域 2	$Time_2 = \frac{JerkRate \times (\text{Final}Output - InitialOutput) - AccelRata^2}{JerkRate \times AccelRata}$ $Y(Time) = InitialOutput + (AccelRata \times Time) - \frac{AccelRata^2}{2 \times JerkRate}$
区域 3	$Time_3 = \frac{AccelRata}{JerkRate}$ $Y(Time) = FinalOutput + \frac{1}{2}(JerkRate) \times \left(Time - \frac{FinalOutput - InitialOutput}{AccelRate} - \frac{AccelRate}{JerkRate} \right)^2$ 。

$$\left|InitialOutput - FinalOutput\right| < \frac{AccelRate^2}{JerkRate}$$

SCRV 功能块未达到 AccelRate 或 DecelRate。Out 如下所示：



其中：

$$TotalTime = 2 \times \sqrt{\frac{\left|InitialOutput - FinalOutput\right|}{JerkRate}}$$

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

在大多数协调驱动应用中，整个驱动器组的线速度都由一个主基准值控制。选择不同基准值后，驱动器的速度基准值不能出现“阶跃”变化，因为负载惯量、电机转矩和调谐存在差异时不允许各个驱动器部分以协调一致的方式工作。SCRV 指令专用于对各驱动器部分的基准信号进行斜坡处理，从而使加速度、减速度和急动度（加速度的导数）得到控制。该指令提供了一种机制，可以使驱动器的基准值达到指定的基准值设置点，同时在此过程中可避免对所连接的机器和设备产生过度压力和影响

ST 示例：

```
SSUM_01.In1 := Master_reference;
SSUM_01.Select1 := master_select;
SSUM_01.In2 := Jog_reference;
SSUM_01.Select2 := jog_select;
SSUM(SSUM_01);

select_out := SSUM_01.Out;
```

```
SCRV_01.In := select_out;

SCRV_01.AccelRate := accel;

SCRV_01.DecelRate := accel;

SCRV_01.JerkRate := jerk_rate;

SCRV(SCRV_01);


scurve_out := SCRV_01.Out
```

十五、选择&限幅指令

增强型选择 ESEL 指令

ESEL 指令允许您选择多达六个输入中的一个。选择选项包括:手动选择(由操作员或程序)，高选择，低选择，中位数选择和平均(平均值)选择。

1）指令格式

指令	名称	梯形图表现	ST 表现
ESEL	增强型选择	此指令不可用于梯形图中	ESEL(ESEL_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
ESEL_tag	SELECT_ENHANCED	结构	ESEL 结构

SELECT_ENHANCED 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
In1	模拟信号 1	REAL	变量 立即数	指令的第一个模拟信号输入。 有效值 = 任意浮点值

				默认值 = 0.0
In2	模拟信号 2	REAL	变量 立即数	指令的第二个模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
In3	模拟信号 3	REAL	变量 立即数	指令的第三个模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
In4	模拟信号 4	REAL	变量 立即数	指令的第四个模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
In5	模拟信号 5	REAL	变量 立即数	指令的第五个模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
In6	模拟信号 6	REAL	变量 立即数	指令的第六个模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
HighLimit	指令初始化命令	REAL	变量 立即数	输入上限。如果 SelectLimit = 0 且 HighLimit Less than or equal to LowLimit，指令会将 Status 中的相应位置位，并会设置 Out = LowLimit。 有效值 = HighLimit > LowLimit

				默认值 = 0.0
n1Fault	In1 不良状况指示器	BOOL	变量 立即数	In1 不良状况指示器。如果 In1 从模拟量输入读取，则 In1Fault 通常由模拟量输入的故障状态控制。如果所有 InnFault 输入均为真，指令会将 Status 中的相应位置位，不执行控制算法，并且不会更新 Out。 默认值 = 假
N2Fault	In2 不良状况指示器	BOOL	变量 立即数	In2 不良状况指示器。如果 In1 从模拟量输入读取，则 In1Fault 通常由模拟量输入的故障状态控制。如果所有 InnFault 输入均为真，指令会将 Status 中的相应位置位，不执行控制算法，并且不会更新 Out。 默认值 = 假
N3Fault	In3 不良状况指示器	BOOL	变量 立即数	In3 不良状况指示器。如果 In1 从模拟量输入读取，则 In1Fault 通常由模拟量输入的故障状态控制。如果所有 InnFault 输入均为真，指令会将 Status 中的相应位置位，不执行控制算法，并且不会更新 Out。

				默认值 = 假
N4Fault	In4 不良状况指示器	BOOL	变量 立即数	In4 不良状况指示器。如果 In1 从模拟量输入读取，则 In1Fault 通常由模拟量输入的故障状态控制。如果所有 InnFault 输入均为真，指令会将 Status 中的相应位置位，不执行控制算法，并且不会更新 Out。 默认值 = 假
N5Fault	In5 不良状况指示器	BOOL	变量 立即数	In5 不良状况指示器。如果 In1 从模拟量输入读取，则 In1Fault 通常由模拟量输入的故障状态控制。如果所有 InnFault 输入均为真，指令会将 Status 中的相应位置位，不执行控制算法，并且不会更新 Out。 默认值 = 假
N6Fault	In6 不良状况指示器	BOOL	变量 立即数	In6 不良状况指示器。如果 In1 从模拟量输入读取，则 In1Fault 通常由模拟量输入的故障状态控制。如果所有 InnFault 输入均为真，指令会将 Status 中的相应位置位，不执行控制算法，并且不会更新 Out。

				默认值 = 假
InsUsed	使用的输入数	DINT	变量 立即数	使用的输入数。此参数定义指令使用的输入数。在最大值选择、最小值选择、中位数选择和平均值选择模式下,指令仅考虑 In1 到 InInsUsed。如果该值无效，指令会将 Status 中的相应位置位。如果 InsUsed 无效、指令未处于手动选择模式并且 Override 清零，指令不会更新 Out。 有效值 =1 到 6 默认值 = 1
Selector Mode	选择器模式输入	DINT	变量 立即数	选择器模式输入。该值将决定指令的操作。 0 = 手动选择 1 = 最大值选择 2 = 最小值选择 3 = 中位数选择 4 = 平均值选择 如果该值无效，指令会将 Status 中的相应位置位，并且不更新 Out。 有效值 = 0 到 4

				默认值 = 0
ProgSelector	程序模式下的选择器输入	DINT	变量 立即数	程序模式下的选择器输入。选择器模式为手动选择并且指令处于程序控制模式时，由 ProgSelector 决定将哪路输入 (In1-In6) 送入 Out。如果 ProgSelector = 0，指令将不更新 Out。如果 ProgSelector 无效，指令会将 Status 中的相应位置位。如果该参数无效且指令处于程序控制模式，并且选择模式为手动选择或者 Override 置位，则指令不会更新 Out。 有效值 = 0 到 6 默认值 = 0
OperSelector	操作员模式下的选择器输入	DINT	变量 立即数	操作员模式下的选择器输入。选择器模式为手动选择并且指令处于操作员控制模式时，由 OperSelector 决定将哪路输入 (In1-In6) 送入 Out。如果 OperSelector = 0，指令将不更新 Out。如果 OperSelector 无效，指令会将 Status 中的相应位置位。如果该参数无效且指令处于操作员控制模式，并且选择

				模式为手动选择或者 Override 置位，则指令不会更新 Out。 有效值 = 0 到 6 默认值 = 0
ProgProgReq	程序发出的程序控制请求	BOOL	变量 立即数	程序发出的程序控制请求。由用户程序设置为真可请求程序控制模式。如果 ProgOperReq 为真，则忽略该值。保持该参数为真且 ProgOperReq 为假可将指令锁定为程序控制模式。 默认值为假。
ProgOperReq	程序发出的操作员控制请求	BOOL	变量 立即数	程序发出的操作员控制请求。由用户程序设置为真可请求操作员控制模式。保持该参数为真可将指令锁定为操作员控制模式。 默认值为假。
ProgOverrideReq	程序超控模式请求	BOOL	变量 立即数	程序超控模式请求。由用户程序设置为真可请求进入超控模式。在超控模式下，指令将起到手动选择的作用。 默认值为假。
OperProgReq	操作员发出的程序控制请求	BOOL	变量 立即数	操作员发出的程序控制请求。由操作员界面设置为真以请求程序控制模式。指令会将此输入设置为假。

				默认值为假。
OperOperReq	操作员发出的操作员控制请求	BOOL	变量 立即数	操作员发出的操作员控制请求。由操作员界面设置为真可请求操作员控制模式。指令会将此输入设置为假。 默认值为假。
ProgValueReset	将程序控制值复位	BOOL	变量 立即数	将程序控制值复位。该值为真时，每次执行指令时，所有程序请求输入都将设置为假。 默认值为假。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。如果 Out 溢出，则设置为假。
Out	指令输出	REAL	变量	计算所得的算法输出。
SelectedIn		DINT	变量	所选输入的编号。指令使用该值指示当前送入输出的是哪路输入。如果选择器模式为平均值选择，指令会设置 SelectedIn = 0。
ProgOpe		BOOL	变量	程序/操作员控制指示器。在程序控制模式下设置为真。在操作员控制模式下设置为假 为假。
Override		BOOL	变量	超控模式。指令处于超控模式时设置为真。

Status		DINT	变量	功能块的状态。
InstructFault (Status.0)		BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
InsFaulted (Status.1)		BOOL	变量	所有使用的 Inn 输入的 InnFault 输入均为真。
InsUsedInv (Status.2)		BOOL	变量	nsUsed 值无效。
SelectorModeInv (Status.3)		BOOL	变量	SelectorMode 值无效。
ProgSelectorInv (Status.4)		BOOL	变量	ProgSelector 值无效。
OperSelectorInv (Status.5)		BOOL	变量	OperSelector 值无效。

3）功能说明

ESEL 指令的工作方式如下：

条件	操作
SelectorMode = 0（手动选择）或 Override 为真、ProgOper 为假，且 OperSelector 不等于 0	Out = In[OperSelector] SelectedIn = OperSelector

SelectorMode = 0（手动选择）或 Override 为真、ProgOper 为真，且 ProgSelector 不等于 0	Out = In[ProgSelector] SelectedIn = ProgSelector
SelectorMode = 1（最大值选择）且 Override 为假	Out = [InsUsed] 的最大值 SelectedIn = 最大输入值对应的索引值
SelectorMode = 2（最小值选择）且 Override 为假	Out = [InsUsed] 的最小值 SelectedIn = 最小输入值对应的索引值
SelectorMode = 3（中位数选择）且 Override 为假	Out = [InsUsed] 的中位数 SelectedIn = 输入值中位数对应的索引值
SelectorMode = 4（平均值选择）且 Override 为假	Out = In[InsUsed] 的平均值 SelectedIn = 0

当 SelectorMode 为 1 到 4 之间的值时，如果任何输入出现不良状况，选择时将会忽略该不良输入。例如，如果 SelectorMode = 1（最大值选择），并且 In6 的值最大但状况不良，则指令会将状况良好的次最大输入值送入输出。

对于最大值或最小值选择模式，如果两路输入相等且同为最大值或最小值，则指令会输出找到的第一路输入。对于中位数选择模式，中位数始终代表从可用输入中选择的值。如果多路输入的值为中位数，则指令会输出找到的第一路输入。

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

该 ESEL 指令根据 SelectorMode 选择 In1、In2 或 In3。在本示例中，SelectorMode = 1，表示最大值选择。指令将找到最大的输入值，并将 Out 值设置为最大 In 值。

ST 示例：

```
ESEL_01.In1 := analog_input1;

ESEL_01.In2 := analog_input2;

ESEL_01.In3 := analog_input3;

ESEL_01.SelectorMode := 1;

ESEL(ESEL_01);


selected_value := ESEL_01.Out;
```

上下限 HLL 指令

HLL 指令将输入值限制在两个值之间。您可以选择高/低、高或低限制

1）指令格式

指令	名称	梯形图表现	ST 表现
HLL	上下限	此指令不可用于梯形图中	HLL(HLL_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
HLL_tag	HL_LIMIT	结构	HL 结构

HL_LIMIT 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
IN	过程误差信号输入	REAL	变量 立即数	过程误差信号输入。这是设置点与反馈之差。 有效值 = 任意浮点值

				默认值 = 0.0
HighLimit	指令初始化命令	REAL	变量 立即数	输入上限。如果 SelectLimit = 0 且 HighLimit Less than or equal to LowLimit，指令会将 Status 中的相应位置位，并会设置 Out = LowLimit。 有效值 = HighLimit > LowLimit 默认值 = 0.0
LowLimit	输入初始值	REAL	变量 立即数	输入下限。如果 SelectLimit = 0 且 LowLimit Greater than or equal to HighLimit，指令会将 Status 中的相应位置位，并会设置 Out = LowLimit。 有效值 = LowLimit < HighLimit 默认值 = 0.0
SelectLimit	比例增益	REAL	变量 立即数	选择限值输入。此输入有三种设置： 0 = 同时使用两个限值 1 = 使用上限 2 = 使用下限 如果 SelectLimit 无效，指令会假定 SelectLimit = 0，并会将 Status 中的相应位置位。 有效值 = 0 到 2

				默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。该输出的数学状态标志置位。
HighAlarm	上限报警指示器	BOOL	变量	上限报警指示器。当 Out 的计算值 Greater than or equal to HighLimit 且输出和积分器强制设为 HighLimit 时置位。
LowAlarm	下限报警指示器	BOOL	变量	下限报警指示器。当 Out 的计算值 Less than or equal to LowLimit 且输出和积分器强制设为 LowLimit 时置位。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。

LimitsInv (Status.1)	Status.1	BOOL	变量	HighLimit 小于或等于 LowLimit。
SelectLimitInv(Status.2)	Status.2	BOOL	变量	SelectLimit 的值不是 0、1 或 2。

3）功能说明

SelectLimit 的值不是 0、1 或 2。

选择	条件	操作
SelectLimit = 0 （使用上限和下限）	In < HighLimit 且 In > LowLimit	Out = In
	In 大于等于 HighLimit	Out = HighLimit
	In 小于等于 HighLimit	Out = LowLimit
	HighLimit 小于等于 LowLimit	Out = LowLimit
SelectLimit = 0 （使用上限和下限）	n < HighLimit	Out = In
	In 大于等于 HighLimit	Out = HighLimit
SelectLimit = 2 （仅使用下限）	In > LowLimit	Out = In
	In 小于等于 LowLimit	Out = LowLimit

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

ST 示例：

```
HLL_01.In := value;
HLL_01.HighLimit := high_limit;
HLL_01.LowLimit := low_limit;
HLL(HLL_01);
limited_value := HLL_01.Out;
high_alarm := HLL_01.HighAlarm;
low_alarm := HLL_01.LowAlarm;
```

变化率限制器 RLIM 指令

RLIM 指令限制信号随时间的变化量。

1）指令格式

指令	名称	梯形图表现	ST 表现
RLIM	变化率限制器	此指令不可用于梯形图中	RLIM(RLIM_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
RLIM tag	RATE_LIMITER	结构	RLIM 结构

RATE_LIMITER 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位
In	指令的模拟	REAL	变量	指令的模拟信号输入。

	信号输入		立即数	有效值 = 任意浮点值 默认值 = 0.0
IncRate	最大输出增量变化率	REAL		最大输出增量变化率，以单位/秒表示。 如果该值无效，指令将设置 IncRate = 0.0 并将 Status 中的相应位置位。 有效值 = 任何 Greater than or equal to 0.0 的浮点值 默认值 = 0.0
DecRate	绕过算法请求	BOOL		绕过算法请求。该参数为真时，Out = In。 默认值为假。
ByPass	绕过算法请求	BOOL		绕过算法请求。该参数为真时，Out = In。 默认值为假。
TimingMode	选择时序执行模式	DINT		选择时序执行模式。 0 = 周期模式 1 = 过采样模式 2 = 实时采样模式 有效值 = 0 到 2 默认值 = 0
OversampleDT	过采样模式的执行时间	REAL		过采样模式的执行时间。 有效值 = 0 到 4194.303 秒 默认值 = 0

RTSTime	实时采样模式的模块更新周期	DINT		实时采样模式的模块更新周期 有效值 = 1 到 32,767 ms 默认值 = 1
RTSTimeStamp	实时采样模式的模块时戳值	DINT		实时采样模式的模块时戳值。 有效值 = 0 到 32,767 ms 默认值 = 0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。该输出的数学状态标志置位。
DeltaT	两次更新间隔的时间	REAL	变量	两次更新间隔的时间。控制算法计算过程输出所用的时间（秒）。
Status	功能块的状态	DINT	变量	功能块的状态。
InstructFault (Status.0)	Status.0)	BOOL	变量	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其他状态位以确定发生的情况。
IncRateInv	Status.1	BOOL	变量	

(Status.1)				IncRate < 0。指令使用值 0。
DecRateInv (Status.2)	Status.2	BOOL	变量	DecRate < 0。指令使用值 0。
TimingModeInv (Status.27)	Status.27	BOOL	变量	无效的 TimingMode 值。 有关时序模式的更多信息，请参见“功能块属性”部分。
RTSMissed (Status.28)	Status.28	BOOL	变量	仅用于实时采样模式。当 ABS(DeltaT – RTSTime) > 1 毫秒时， 设置为真。
RTSTimeInv (Status.29)	Status.29	BOOL	变量	RTSTime 值无效
RTSTimeStampInv (Status.30)	Status.30	BOOL	变量	RTSTimeStamp 值无效。
DeltaTInv (Status.31)	Status.31	BOOL	变量	在过采样模式下，如果 DeltaT <= 0 或 DeltaT > 4194.303，此参数设置为真。

3）功能说明

RLIM 指令提供独立的增量变化率和减量变化率，以单位/秒表示。ByPass 输入用于停止变化率限制，将信号直接传送至输出。

条件	操作
ByPass 为真	Outn = Inn

	Outn-1 = Inn
ByPass 为假，且 DeltaT > 0	$Slope = \frac{In_n - Out_{n-1}}{DeltaT}$ $If\ Slope \le -DecRate\ then\ YSlope = -DecRate$ $If\ -DecRate \le Slope \le IncRate\ then\ YSLope = Slope$ $If\ IncRate \le Slope\ then\ YSlope = IncRate$ $Out_n = Out_{n-1} + DeltaT \times YSlope$ $Out_{n-1} = Out_n$ $WhereDeltaT\ is\ in\ seconds$

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

RLIM 指令通过 IncRate 对 In 进行限制。如果 analog_input1 的变化率大于 IncRate 的值，指令会对 In 进行限制。指令将 Out 设为 In 的变化率限值。

SSUM_01.Select1 := select1;

ST 文本：

```
RLIM_01.In := analog_input1;

RLIM_01.IncRate := value;

RLIM(RLIM_01);

rate_limited := RLIM_01.Out;SSUM(SSUM_01);

selected_add := SSUM_01.Out;
```

取反选择 SNEG 指令

SNEG 指令使用数字输入在输入值和输入值的负值之间进行选择

1）指令格式

指令	名称	梯形图表现	ST 表现
SNEG	取反选择	此指令不可用于梯形图中	SNEG(SNEG_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
SNEG tag	SELECTABLE_NEGATE	结构	SNEG 结构

SNEG 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量	

			立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。默认置位
In	指令的模拟信号输入	REAL	变量 立即数	指令的模拟信号输入。 有效值 = 任意浮点值 默认值 = 0.0
NegateEnable	取反使能	BOOL	变量 立即数	取反使能。当 NegateEnable 为真时，指令会将 Out 设为 In 的取反值。 默认值为真

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。

3）功能说明

SNEG 指令的工作方式如下：

条件	操作
NegateEnable 为真	Out = - In
NegateEnable 为假	Out = In

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。
正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

negate_enable 确定是否对 In 取反。如果 NegateEnable 为假，指令将设置 Out = In。如果 NegateEnable 为真，指令将设置 Out = -In。

ST 示例：

```
SNEG_01.In := analog_input1;
SNEG_01.NegateEnable := negate_enable;
SNEG(SNEG_01);
output_value := SNEG_01.Out;
```

加法选择 SSUM 指令

SSUM 指令使用布尔输入来选择要进行代数求和的输入。

1）指令格式

指令	名称	梯形图表现	ST 表现
SSUM	加法选择	此指令不可用于梯形图中	SSUM(SSUM_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
SSUM tag	SELECTED_SUMMER	结构	SSUM 结构

HL_LIMIT 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableIn	使能输入	BOOL	变量 立即数	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位

In1	要进行求和的第一路输入	REAL	变量 立即数	要进行求和的第一路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain1	第一路输入的增益	REAL	变量 立即数	第一路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select1	第一路输入的选择信号	BOOL	变量 立即数	第一路输入的选择信号。 默认值为假。
In2	要进行求和的第二路输入	REAL	变量 立即数	要进行求和的第二路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain2	第二路输入的增益	REAL	变量 立即数	第二路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select2	第二路输入的选择信号	BOOL	变量 立即数	第二路输入的选择信号。 默认值为假。
In3	要进行求和的第三路输入	REAL	变量 立即数	要进行求和的第三路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain3	第三路输入的增益	REAL	变量 立即数	第三路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0

Select3	第三路输入的选择信号	BOOL	变量 立即数	第三路输入的选择信号。 默认值为假。
In4	要进行求和的第四路输入	REAL	变量 立即数	要进行求和的第四路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain4	第四路输入的增益	REAL	变量 立即数	第四路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select4	第四路输入的选择信号	BOOL	变量 立即数	第四路输入的选择信号。 默认值为假。
In5	要进行求和的第五路输入	REAL	变量 立即数	要进行求和的第五路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain5	第五路输入的增益	REAL	变量 立即数	第五路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select5	第五路输入的选择信号	BOOL	变量 立即数	第五路输入的选择信号。 默认值为假。
In6	要进行求和的第二路输入	REAL	变量 立即数	要进行求和的第二路输入。 有效值 = 任意浮点值 默认值 = 0.0

Gain6	第二路输入的增益	REAL	变量 立即数	第二路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select6	第二路输入的选择信号	BOOL	变量 立即数	第二路输入的选择信号。 默认值为假。
In7	要进行求和的第七路输入	REAL	变量 立即数	要进行求和的第七路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain7	第七路输入的增益	REAL	变量 立即数	第七路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select7	第七路输入的选择信号	BOOL	变量 立即数	第七路输入的选择信号。 默认值为假。
In8	要进行求和的第八路输入	REAL	变量 立即数	要进行求和的第八路输入。 有效值 = 任意浮点值 默认值 = 0.0
Gain8	第八路输入的增益	REAL	变量 立即数	第八路输入的增益。 有效值 = 任意浮点值 默认值 = 1.0
Select8	第八路输入的选择信号	BOOL	变量 立即数	第八路输入的选择信号。 默认值为假。

Bias	偏置信号输入	REAL	变量 立即数	偏置信号输入。指令会将 Bias 与输入之和相加。 有效值 = 任意浮点值 默认值 = 0.0

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
EnableOut	启用输出	BOOL	变量	指示指令是否处于启用状态。
Out	指令输出	REAL	变量	计算所得的算法输出。该输出的数学状态标志置位。

3）功能说明

SSUM 指令的工作方式如下：

条件	操作
未选择任何 In	Out = Bias
已选择一个或多个 In	对于所有 n，Selectn 为真时 Out = Σ (Inn x Gainn) + Bias

结构化文本

条件/状态	执行的操作
预扫描	EnableIn 和 EnableOut 位设置为假。

正常执行	EnableIn 和 EnableOut 位设置为真。 指令执行。
后扫描	EnableIn 和 EnableOut 位设置为假。

4）注意事项

5）错误说明

6）示例

ST 示例：

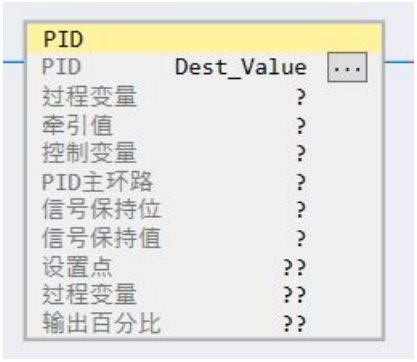
```
SSUM_01.In1 := analog_input1;
SSUM_01.Select1 := select1;
SSUM_01.In2 := analog_input2;
SSUM_01.Select2 := select2;
SSUM(SSUM_01);
selected_add := SSUM_01.Out;
```

十六、特殊指令

比例、积分和微分 PID 指令

PID 指令控制流量、压力、温度或液位等过程变量。

1）指令格式

指令	名称	梯形图表现	ST 表现
PID	比例、积分和微分		PID(PID, ProcessVariable, Tieback, ControlVariable, PIDMasterLoop, InHoldBit, InHoldValue);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
PID	PID	PID	结构体	PID 结构
Process variable	过程变量	REAL	变量	需要控制的值
Tieback	牵引值	SINT INT	立即数 变量	硬件手动/自动站的输出，该输出绕过控制器的输出。

		DINT REAL		如果不使用此参数，请输入 0。
Control variable	控制变量	REAL	变量	将送入最终控制设备（阀，气闸等）的值。 如果使用死区，Control variable 必须为 REAL 型，否则当误差处于死区内时，该值将被强制为 0。
PID master loop	PID 主环路	PID	结构体	主 PID 的 PID 变量 如果您正在进行串级控制,并且这个 PID 是副环，请输入主 PID 的名称。 如果不想使用此参数，则输入 0
Inhold bit	信号保持位	BOOL	变量	支持无扰动重启的输出通道
Inhold value	信号保持值	SINT INT DINT REAL	变量	
SetPoint	设置点	-	-	仅供显示使用 显示设置点的当前值
Process variable	过程变量	-	-	仅供显示使用 标定 Process Variable 的当前值
Output %	输出百分比	-	-	仅供显示之用 显示当前输出百分比值

PID 结构

助记符	数据类型	说明		
.CTL	DINT	.CTL 成员可提供对状态成员（位）的访问，访问时采用一个 32 位字的形式。 位 07-15 由 PID 指令置位		
		位	编号	说明
		.EN	31	
		.CT	30	级联类型（0 = 从；1 = 主）
		.CL	29	级联回路（0 = 否；1 = 是）
		.PVT	28	过程变量跟踪（0 = 否；1 = 是）
		. DOE	27	微分对象（0 = PV；1 = 误差）
		.SWM	26	软件模式（0 = 否-自动；1 = 是 软件手动）
		.CA	25	控制动作（0 = 反向（SP-PV）； 1 = 直接（PV-SP））
		.MO	24	站点模式（0 = 自动；1 = 手动）
		.PE	23	PID 公式（0 = 独立；1 = 相关）
		.NDF	22	微分平滑（0 = 否；1 = 是）
		.NOBC	21	偏置计算（0 = 否；1 = 是）
		.NOZC	20	过零（0 = 否；1 = 非过零死区）
		.INI	15	PID 已初始化（0 = 否；1 = 是）
		.SPOR	14	设置点超出范围（0 = 否；1 = 是）
		.OLL	13	CV 低于输出下限值（0 = 否；1 = 是）
		.OLH	12	CV 高于输出上限值（0 = 否；1 = 是）
		.EWD	11	误差在死区内（0 = 否；1 = 是）

		.DVNA	10	误差下限报警（0 = 否；1 = 是）
		.DVPA	9	误差上限报警（0 = 否；1 = 是）
		.PVLA	8	PV 下限报警（0 = 否；1 = 是）
		.PVHA	7	PV 上限报警（0 = 否；1 = 是）
.SP	REAL	设置点		
.KP	REAL	独立 - 比例增益（无单位）		
		相关 - 控制器增益（无单位）		
.KI	REAL	独立 - 积分增益（1/秒）		
		相关 - 复位时间（分钟/循环）		
.KD	REAL	独立 - 微分增益（秒）		
		相关 - 比率时间（分）		
.BIAS	REAL	前馈或偏置百分比		
.MAXS	REAL	最大工程单位标定值		
.MINS	REAL	最小工程单位标定值		
.DB	REAL	死区工程单位		
.SO	REAL	设置输出百分比		
.MAXO	REAL	输出上限（输出百分比）		
.MINO	REAL	输出下限（输出百分比）		
.UPD	REAL	回路更新时间（秒）		
.PV	REAL	标定 PV 值		
.ERR	REAL	标定误差值		
.OUT	REAL	输出百分比		

.PVH	REAL	过程变量报警上限	
.PVL	REAL	过程变量报警下限	
.DVP	REAL	偏差报警上限	
.DVN	REAL	偏差报警下限	
.PVDB	REAL	过程变量报警死区	
.DVDB	REAL	偏差报警死区	
.MAXI	REAL	最大 PV 值（未标定输入）	
.MINI	REAL	最小 PV 值（未标定输入）	
.TIE	REAL	手动控制的牵引值	
.MAXCV	REAL	最大 CV 值（对应于 100%）	
.MINCV	REAL	最小 CV 值（对应于 0%）	
.MINTIE	REAL	最小牵引值（对应于 100%）	
.MAXTIE	REAL	最大牵引值（对应于 0%）	
.DATA[17]	REAL	.DATA 成员存储：	
		元素	说明
		.DATA[0]	.DATA[0]
		.DATA[1]	微分平滑临时值
		.DATA[2]	前一次 .PV 值
		.DATA[3]	前一次 .ERR 值
		.DATA[4]	前一次有效 .SP 值
		.DATA[5]	百分比标定常数
		.DATA[6]	.PV 标定常数

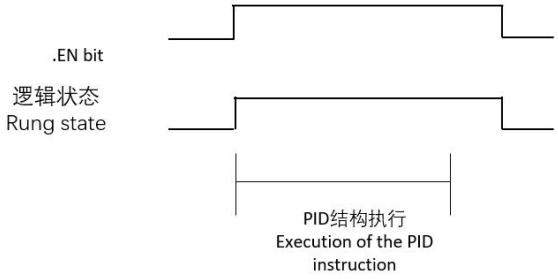
		.DATA[7]	微分标定常数
		.DATA[8]	前一次 .KP 值
		.DATA[9]	前一次 .KI 值
		.DATA[10]	前一次 .KD 值
		.DATA[11]	相关增益 .KP
		.DATA[12]	相关增益 .KI
		.DATA[13]	相关增益 .KD
		.DATA[14]	前一次 .CV 值
		.DATA[15]	.CV 反标定常数
		.DATA[16]	牵引值反标定常数

3）功能说明

PID 指令通常从模拟输入模块接收过程变量（PV），并通过模拟输出模块调节控制变量输出（CV），以使过程变量保持在期望的设定值。

时序图

.EN 位表示执行状态。当 EnableIn 从“假”（False）变为“真”（True）时，.EN 位被置位。当 EnableIn 变为“假”时，.EN 位被清除。PID 指令不使用.DN 位。只要 EnableIn 为“真”，PID 指令就会在每次扫描时执行。



4）注意事项

建议使能指令前，适配好相关的参数，防止输出变化过大或过快，超出预想值或造成风险。

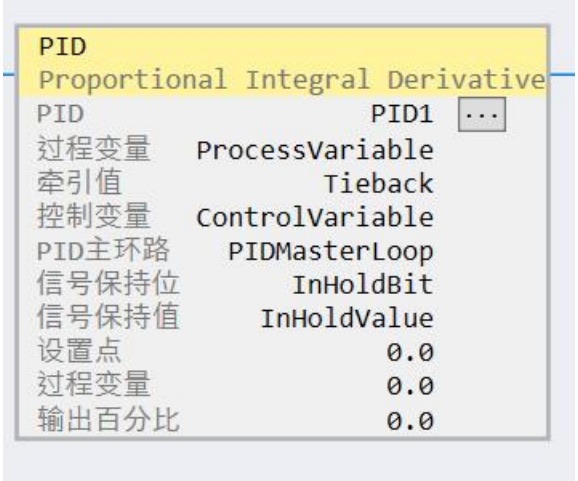
5）错误说明

在以下情况下会发生轻微故障：	故障类型	故障代码
UPD=0	4	35
设置点超出范围	4	36

6）示例

梯形图：

ControlVariable	DINT
InHoldBit	BOOL
InHoldValue	DINT
PID1	PID
PIDMasterLoop	PID
ProcessVariable	DINT
Tieback	DINT



构化文本：

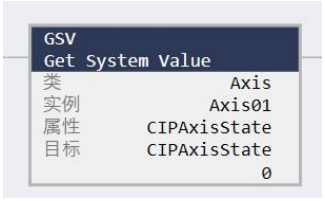
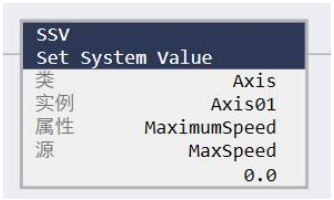
```
PID(PID1,              //PID
    ProcessVariable,//过程变量
    Tieback,           //牵引值
    ControlVariable,//控制变量
    PIDMasterLoop, //PID 主环路
    InHoldBit,         //信号保持位
    InHoldValue);    //信号保持值
```

获取和设置系统值 GSV 和 SSV 指令

GSV 指令获取存储在控制器系统中的对象数据。

SSV 指令设置存储在控制器系统中的对象数据。

1）指令格式

指令	名称	梯形图表现	ST 表现
GSV	获取系统值		GSV(ClassName， InstanceName， AttributeName， Dest);
SSV	设置系统值		SSV(ClassName， InstanceName， AttributeName， Source);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Class Name	类名称	-	名称	对象类的名称
Instance name	实例名称	-	名称	特定对象的名称（当对象需要名称时）
Attribute Name	属性名称	-	名称	对象的属性

				数据类型取决于所选的属性
Destination(GSV)	目标值	SINT INT DINT REAL 结构体	变量	属性数据的目标变量
Source (SSV)	源	SINT INT DINT REAL 结构体	变量	包含要复制到属性的数据的变量

3）功能说明

GSV/SSV 指令用于获取或设置存储在对象中的控制器状态数据。控制器将状态数据存储在对象中。

当条件为真时，GSV 指令会检索指定的信息，并将其放置在目标位置。当条件为真时，SSV 指令会使用来自源的数据设置指定的属性。

当你输入 GSV/SSV 指令时，编程软件会显示每个指令的有效对象类、对象名称和属性名称。对于 GSV 指令，你可以获取所有属性的值。对于 SSV 指令，软件仅显示可以设置的属性（SSV）。

执行行为：

情况/状态	行为

Prescan	无。
执行条件为 FALSE	无
执行条件为 TRUE	指令执行
Postscan	无。

4）注意事项

谨慎使用 SSV 指令。对对象进行更改可能会导致控制器运行异常或对人员造成伤害。

你必须测试并确认这些指令不会更改你不希望更改的数据。

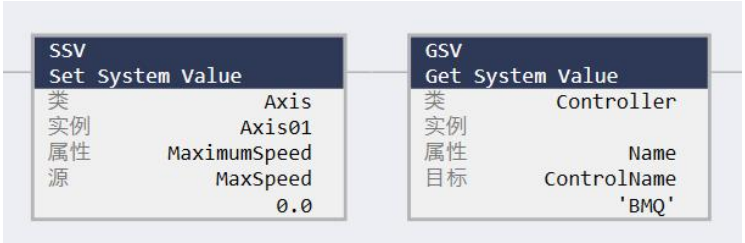
SSV 指令写入数据，而 GSV 指令读取数据，它们可以越过一个成员访问变量中的其他成员。如果变量太小，指令将无法写入或读取数据，而是记录一个轻微故障。

5）错误说明

如果出现以下情况，将发生轻微故障：	故障类型	故障代码
对象地址无效	4	5
指定的对象不支持 GSV/SSV	4	6
属性无效	4	6
为 SSV 指令提供的信息不足	4	6
GSV 指令的目标位置不足以容纳请求的数据	4	7

6）示例

梯形图：

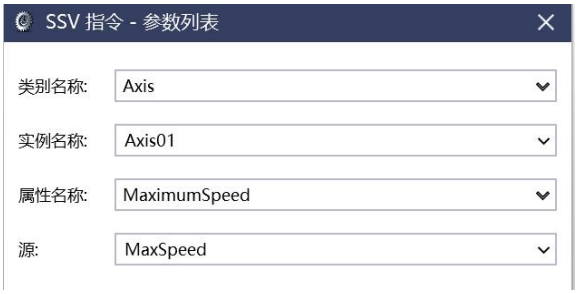


1. **设置 Axis01 的最大速度**

2. **获取控制器名称**

ST 示例：


```
GSV(Controller,      //类
    ,                //实例名称
    Name,            //属性名称
    ControlName);    //目标
```




```
SSV(Axis,
    Axis01,
```

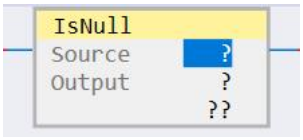
MaximumSpeed,

MaxSpeed);

判断不存在的轴 ISNULL 指令

用于判断轴指针变量是否有指向不存在的轴。(触发方式：上升沿触发)

1）指令格式

指令	名称	梯形图表现	ST 表现
ISNULL	判断轴指针变量是否有指向不存在的轴		IsNull(Source,0utput);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	轴指针变量	AXIS_REF_CIP AXIS_REF_VIRTUAL	变量	存储轴指针的变量
Output	结果输出	BOOL	变量	判断结果，1 或者 0， 1 代表指针指向了空 NULL

3）功能说明

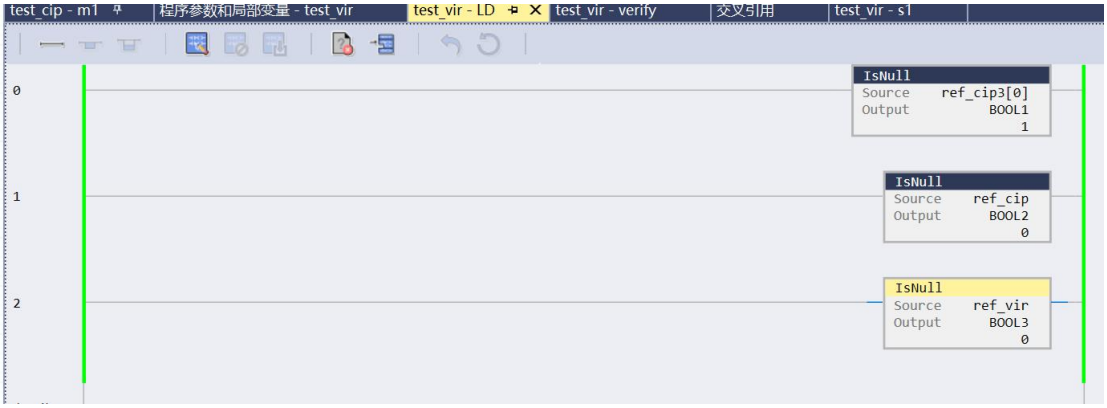
4）注意事项

轴指针指向不存在的轴对象(未经 REF 指令指向对应轴对象空值或者轴指针变量值被人为修改， ICS 规则定为变量表监控页以及代码区监控页轴指针类型变量值不允许人为修改)，实际运行引用到此变量时 PLC 直接进入 Fault 状态。

5）错误说明

6）示例

梯形图：



ST 示例：

下面如图，轴指针变量 ref_cip、ref_vir 分别执行指向了实轴 AX2 和虚轴 vir1，ref_cip3[0]未指向指定轴。

使用 ISNULL 指令后，PLC 在线运行，BOOL1 置 1，BOOL2 和 BOOL3 置 0，证明 ref_cip3[0]指针指向了空 NULL。

```
1
2   REF(   AX2,                ref_cip);
        {...} 16#0000_0000_2983_9300
3   REf(  vir1,                ref_vir);
        {...} 16#0000_0000_2980_4eb0
4
5   IsNull(           ref_cip3[0],BOOL1);//IsNull指令测试
          16#0000_0000_0000_0000  1
6   IsNull(              ref_cip,BOOL2);
          16#0000_0000_2983_9300  0
7   IsNull(              ref_vir,BOOL3);
          16#0000_0000_2980_4eb0  0
8
9
```

消息传递 MSG 指令

MSG 指令与其它模块进行数据异步传输。

1）指令格式

指令	名称	梯形图表现	ST 表现
MSG	消息传递		MSG(message);

2）相关变量

输入输出变量(InOut)

助记符	数据类型	说明
.FLAGS	INT	.FLAGS 成员可提供对状态成员（位）的访问，访问时采用一个 16 位字的形式。
	位 2	对应成员 .EW
	4	.ER
	5	.DN
	6	.ST
	7	.EN
	8	.TO
	9	.EN_CC

	重要事项： 请勿更改 FLAGS 成员的 EW、ER、DN 或 ST 位。例如，请勿清除整个 FLAGS 字。控制器会忽略更改，并使用内部存储的位值。	
.ERR	INT	如果.ER 位置位，错误代码字会标识出 MSG 指令的错误代码。
.EXERR	INT	扩展错误代码字，用于指定部分错误代码的附加错误代码信息。
.REQ_LEN	INT	请求的长度，用于指定消息指令将尝试传输的字数。
.DN_LEN	INT	完成长度，标识实际传输的字数。
.EW	BOOL	当控制器检测到消息请求已进入队列时，使能等待位（.EW）被置位。当.ST 位被置位时，控制器会复位 EW 位。 重要事项： 不要更改.EW 位。控制器会忽略这种更改，并使用内部存储的该位的值。
.ER	BOOL	错误位，当控制器检测到传输失败时置位。EnableIn 下一次由假变为真时，.ER 位复位。 重要事项： 不要更改.ER 位。控制器会忽略这种更改，并使用内部存储的该位的值。
.DN	BOOL	完成位，当消息的最后一个信息包成功传输后置位。EnableIn 下一次由假变为真时，.DN 位复位。

		重要事项： **不要更改.DN 位。控制器会忽略这种更改，并使用内部存储的该位的值。**
.ST	**BOOL**	**开始位，当控制器开始执行 MSG 指令时置位。当.DN 位或.ER 位置位时，.ST 位复位。** **重要事项：** **不要更改.ST 位。控制器会忽略这种更改，并使用内部存储的该位的值。**
.EN	**BOOL**	**使能位，当 EnableIn 变为真时置位，并保持置位直到.DN 位或.ER 位置位且 EnableIn 为假。如果 EnableIn 条件为假，但.DN 位和.ER 位清零，则.EN 位保持置位。** **重要事项：** **不要更改.EN 位。控制器会忽略这种更改，并使用内部存储的该位的值。**
.TO	**BOOL**	**如果手动将.TO 位置位，控制器将停止处理消息，并将.ER 位置位。**
.EN_CC	**BOOL**	**使能缓存位，用于决定 MSG 连接的管理方式。如果希望控制器保留该连接（例如在多次重复执行同一条 MSG 指令时），应将.EN_CC 位置位。如果很少执行 MSG 指令并且需要另外的控制器连接，应将.EN_CC 位清零。** **即使.EN_CC 位置位，由串行端口发出的 MSG 指令连接**

		也不会缓存。
.ERR_SRC	SINT	在“消息配置”(MessageConfiguration)对话框中显示错误路径。
.DestinationLink	INT	要更改 DH+或带源 ID 消息的 CIP 的目标链路，应将此成员设置为所需值。
.DestinationNode	INT	要更改 DH+或带源 ID 消息的 CIP 的目标节点，应将此成员设置为所需值。
.SourceLink	INT	要更改 DH+或带源 ID 消息的 CIP 的源链路，应将此成员设置为所需值。
.Class	INT	要更改 CIP 通用消息的 Class 参数，应将此成员设置为所需值。
.Attribute	INT	要更改 CIP 通用消息的 Attribute 参数，应将此成员设置为所需值。
.Instance	DINT	要更改 CIP 通用消息的 Instance 参数，应将此成员设置为所需值。
.LocalIndex	DINT	如果你使用星号[*]来指定本地数组的元素编号，LocalIndex 将提供该元素编号。要更改元素编号，需将此成员设置为所需的值。
		如果消息为：　则本地数组为：
		读取数据　目标元素
		写入数据　源元素
.Channel	SINT	

<table>
<tr><td>.Rack</td><td>SINT</td><td colspan="2">要更改块传输消息的机架号，需将此成员设置为所需的机架号（八进制）。</td></tr>
<tr><td>.Group</td><td>SINT</td><td colspan="2">要更改块传输消息的组编号，应将此成员设置为所需组编号（八进制）。</td></tr>
<tr><td rowspan="4">.Slot</td><td rowspan="4">SINT</td><td colspan="2">要更改块传输消息的插槽编号，应将此成员设置为所需插槽编号（八进制）。</td></tr>
<tr><td>如果消息通过此网络传输：</td><td>则插槽编号以如下形式指定：</td></tr>
<tr><td>通用远程 I/O</td><td>八进制</td></tr>
<tr><td>ControlNet</td><td>十进制（0-15）</td></tr>
<tr><td>.Path</td><td>STRING</td><td colspan="2">要将消息发送到其他控制器，应将此成员设置为新路径。以十六进制值的形式输入路径。省略逗号[,]例如路径 1,0,2,42,1,3，可输入$01$00$02$2A$01$03。要浏览到某个设备并自动创建部分或完整的新字符串，请右键单击字符串变量，然后选择“转至消息路径编辑器”(GotoMessagePathEditor)。</td></tr>
<tr><td rowspan="4">.RemoteIndex</td><td rowspan="4">DINT</td><td colspan="2">如果使用星号[*]来指定远程数组的元素编号，将由 RemoteIndex 提供元素编号。要更改元素编号，应将此成员设置为所需值。</td></tr>
<tr><td>如果消息为：</td><td>则远程数组为：</td></tr>
<tr><td>读取数据</td><td>源元素</td></tr>
<tr><td>写入数据</td><td>目标元素</td></tr>
</table>

<table>
<tr><td>.RemoteElement</td><td>STRING</td><td colspan="2">要指定将消息发送到控制器中的其他目标变量或地址，应将此成员设置为所需值。以 ASCII 字符形式输入变量或地址。</td></tr>
<tr><td rowspan="3"></td><td rowspan="3"></td><td>如果消息为：</td><td>则远程数组为：</td></tr>
<tr><td>读取数据</td><td>源元素</td></tr>
<tr><td>写入数据</td><td>目标元素</td></tr>
<tr><td>.UnconnectedTimeout</td><td>DINT</td><td colspan="2">非连接型消息的超时，或者建立连接的超时。默认值为 30 秒。如果消息为非连接型消息，则当控制器在 UnconnectedTimeout 时间内未获得响应时，.ER 位置位。如果消息为连接型消息，则当控制器在 UnconnectedTimeout 时间内未获得建立连接的响应时.ER 位置位。</td></tr>
<tr><td>.ConnectionRate</td><td>DINT</td><td colspan="2" rowspan="2">对于已建立连接的消息，响应超时时间是指从另一设备收到响应的等待时间。
此超时时间仅在建立连接后才生效。
超时时间=连接速率（ConnectionRate）×超时倍数（TimeoutMultiplier）。
连接速率的默认值为 7.5 秒。
超时倍数的默认值为 0（相当于乘数因子为 4）。
已连接消息的默认超时时间为 30 秒（7.5 秒×4=30 秒）。
要更改超时时间，可以更改连接速率，同时保持超时倍数为默认值。</td></tr>
<tr><td>.TimeoutMultiplier</td><td>SINT</td></tr>
</table>

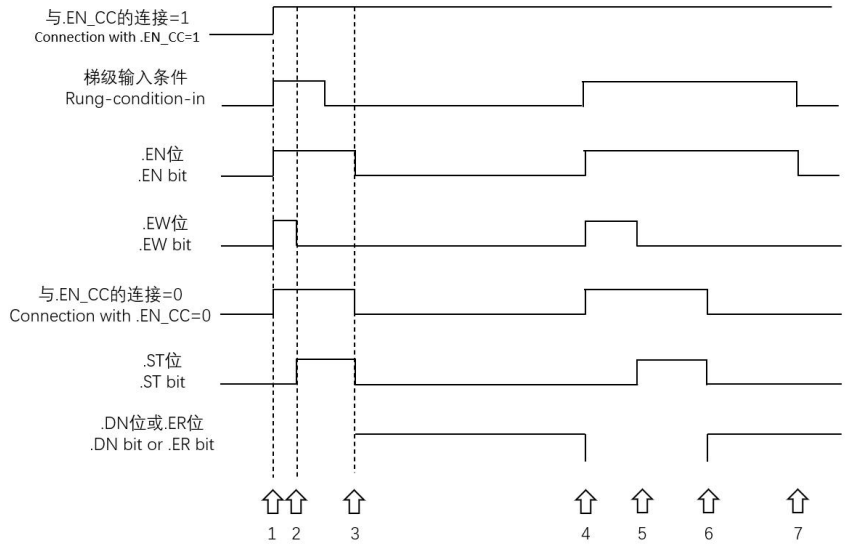
3）功能说明

MSG 指令用于传输数据元素。这是一个过渡性指令：

在梯形图中，每次执行该指令时，需要将使能输入（EnableIn）从清除状态切换为置位状态。

每个数据元素的大小取决于你指定的数据类型以及使用的消息命令类型。

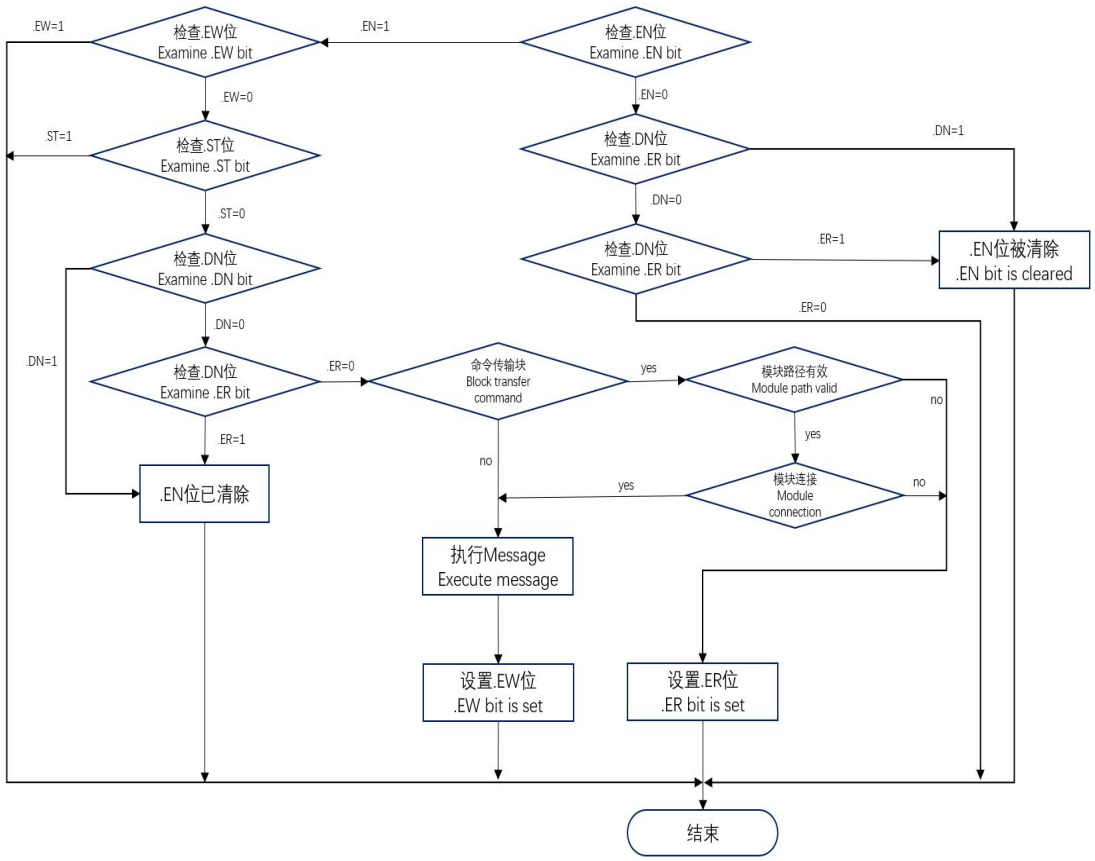
时序图



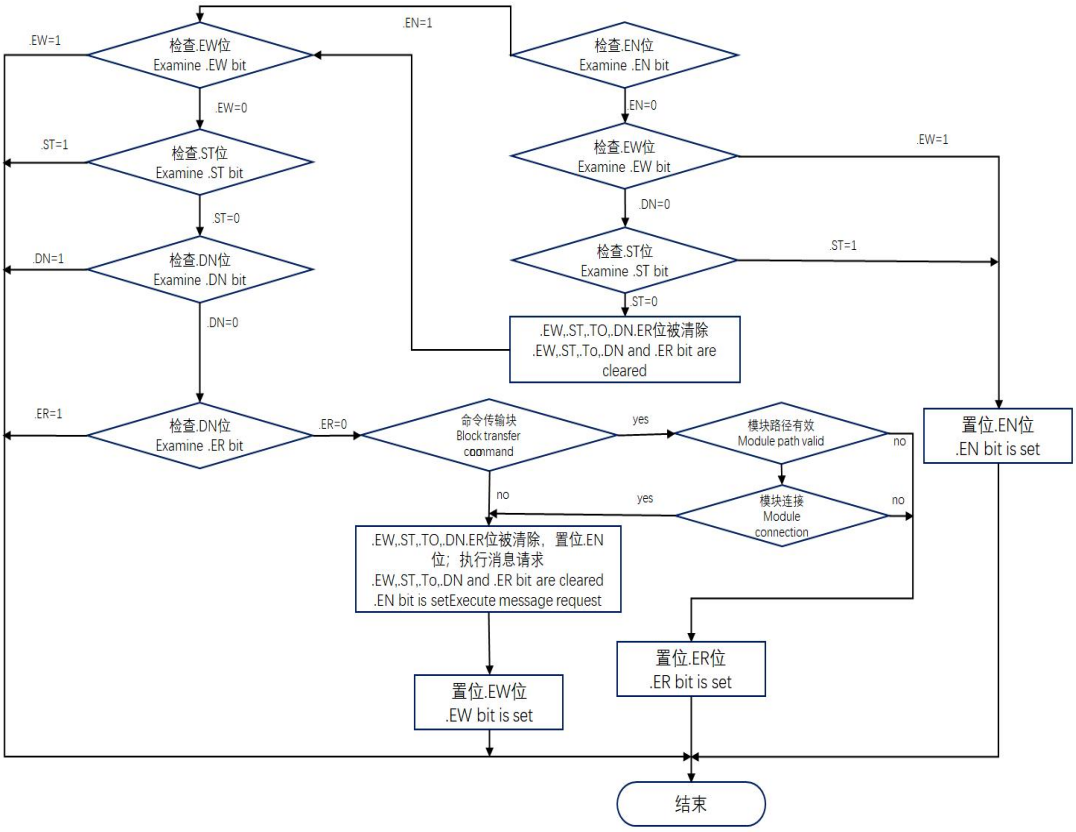
位置	描述
1	EnableIn 为真 .EN 置位 .EW 置位 连接打开
2	消息已发送

	.ST 置位 .EW 清零
3	消息完成或出错，EnableIn 为假 .DN 或 .ER 置位 .ST 清零 连接关闭（如果 .EN_CC = 0） .EN 清零（因为 EnableIn 为假）
4	EnableIn 为真，且 .DN 或 .ER 先前已置位 .EN 置位 .EW 置位 连接打开 .DN 或 .ER 清零
5	消息已发送 .ST 置位 .EW 清零
6	消息完成或出错，EnableIn 仍为真 .DN 或 .ER 置位 .ST 清零 连接关闭（如果 .EN_CC = 0）
7	EnableIn 变为假，且 .DN 或 .ER 置位 .EN 清零

流程图（假）


检查.EW位
Examine .EW bit
.EW=1
.EW=0
检查.EN位
Examine .EN bit
.EN=1
.EN=0
检查.ST位
Examine .ST bit
.ST=1
.ST=0
检查.DN位
Examine .ER bit
.DN=1
.DN=0
.EN位被清除
.EN bit is cleared
.ER=1
.ER=0
检查.DN位
Examine .DN bit
.DN=0
.DN=1
检查.DN位
Examine .ER bit
.ER=0
.ER=1
命令传输块
Block transfer
command
yes
no
模块路径有效
Module path valid
yes
no
模块连接
Module
connection
yes
no
.EN位已清除
执行Message
Execute message
设置.EW位
.EW bit is set
设置.ER位
.ER bit is set
结束

流程图（真）



4）注意事项

多次检查状态位：

如果在逻辑程序的多个位置对状态位进行检查，可使用这些位的副本。否则，在扫描过程中这些位可能发生更改，造成逻辑程序的运行不符合预期。获得副本的一种方法是使用 FLAGS 字。将复制 FLAGS 字到其他变量，然后检查副本中的这些位。

请勿更改 MSG 指令的以下位：

- **DN**
- **EN**
- **ER**
- **EW**
- **ST**

无论是这些位本身还是 FLAGS 字中的这些位都不要更改。否则，控制器可能发生不可恢复的故障。如果发生不可恢复的故障，控制器会将项目从内存中清除。

5）错误说明

请参考指令代码说明

6）示例

梯形图：

ST 示例：

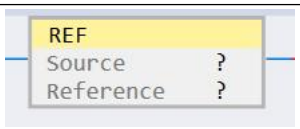
```
MSG(message);
```

获取轴的指针 REF 指令

用于将实轴或者虚轴对象的指针存储到数据类型为 AXIS_REF_CIP 或

AXIS_REF_VIRTUAL 的轴指针变量上。(触发方式：上升沿触发)

1）指令格式

指令	名称	梯形图表现	ST 表现
REF	获取轴的指针		REF(Source,Reference);

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	轴	AXIS_CIP_DRIVE AXIS_VIRTUAL	直接变量	轴对象（运动组里面的实轴或者虚轴）
Reference	轴指针变量	AXIS_REF_CIP AXIS_REF_VIRTUAL	变量	存储轴指针的变量

3）功能说明

执行行为

情况	行为
Prescan	执行指令一次
执行条件为 FALSE	更新梯级输出，停止指令执行
执行条件为 TRUE	更新梯级输出，开始指令执行

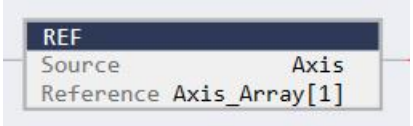
Postscan	执行指令一次

4）注意事项

5）错误说明

6）示例

梯形图：



ST 示例：

REF(Source,Reference);

使用示例 1，轴指针数组下标为常数，数组成员通过^操作符取轴对象传入运动指令使用：

名称	数据类型
Axis_Array	AXIS_REF_CIP[100]

MS0(Axis_Array[1]^,mc_mso)

ActPos_A20:=Axis_Array[1]^.ActualPosition;

其中数据类型为 AXIS_REF_XXX 的变量后面跟着的^符号代表取对应指针的对应类型轴对象；

使用示例 2，轴指针/轴数组可以传入 UDFB 自定义功能块 InOut 类型参数使用

UDFB_AXIS_CTRL(AXIS_CTRL01,Axis_Array);

使用示例 3，UDFB 内部代码可以使用轴指针变量加^操作符访问对应轴对象下成员：

ActPos:=Axis_Ref[1]^,ActualPosition;

使用示例 4，轴指针/轴数组成员加上取对象操作符^后可作为参数传入 GSV/SSV 指令：(UDFB 里面使用 GSV/SSV 指令的时候会用到)

GSV(Axis,Axis_Array[1]^,HomeSpeed,var_HomeSpeed);

SSV(Axis,Axis_Array[1]^,HomeSpeed,var_HomeSpeed);

使用示例 5， 同类型轴指针之间可以互相赋值，不同类型指针之间赋值需要编译

报错：

名称	数据类型
▷ Axis_Array	AXIS_REF_CIP[100]

Axis_Array[1]:=Axis_Array[2];

COP(Axis_Array[1],Axis_Array[2] 1);

CPS(Axis_Array[1],Axis_Array[2],1);

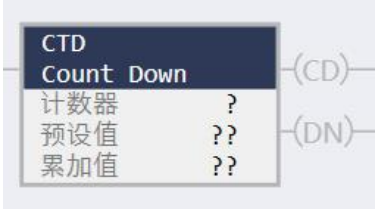
不同类型赋值，编译报错：(举例，实轴指针赋值给了虚轴指针,编译报

十七、定时和计数指令

减计数器 CTD 指令

当梯级输入状态从 0 切换到 1 时，CTD 指令向下计数。

1）指令格式

指令	名称	梯形图表现	ST 表现
CTD	减计数器		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Counter	计数器	COUNTER	变量	计数器结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Preset	预设值	DINT	立即数	Counter.PRE 的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Accum	累加值	DINT	立即数	Counter.ACC 的值

TIMER 结构

助记码:	数据类型	说明:

.CD	BOOL	向下计数使能位，包含指令上次执行时的梯级输入条件。
.DN	BOOL	完成位，清零时指示计数操作完成。
.OV	BOOL	上溢位，置位时指示计数器的值已增至上限 2,147,483,647 以上。
.UN		下溢位，置位时指示计数器的值已减至下限值 -2,147,483,648 以下。
.PRE	DINT	预设值指定了累积值必须达到的数值，只有在达到该数值后，指令才会显示已完成。
.ACC	DINT	累加值，指定指令已计数的跳变次数。

3）功能说明

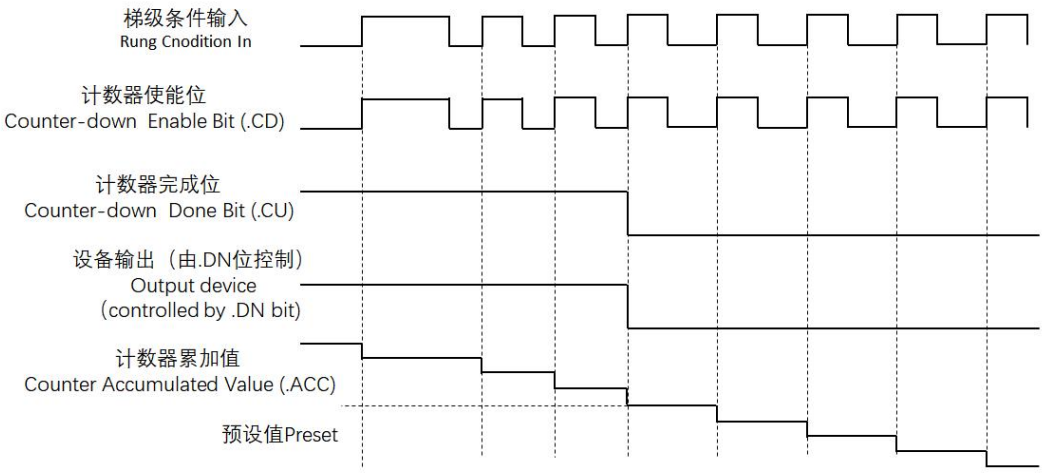
1）CTD 指令通常与引用相同计数器结构的 CTU 指令一起使用。

2）当逻辑行条件输入（rung-condition-in）被置为真（true），且.CD 为假（false）时，.ACC 将减少 1。当逻辑行条件输入为假（false）时，.CD 将被置为假（false）。

执行行为：

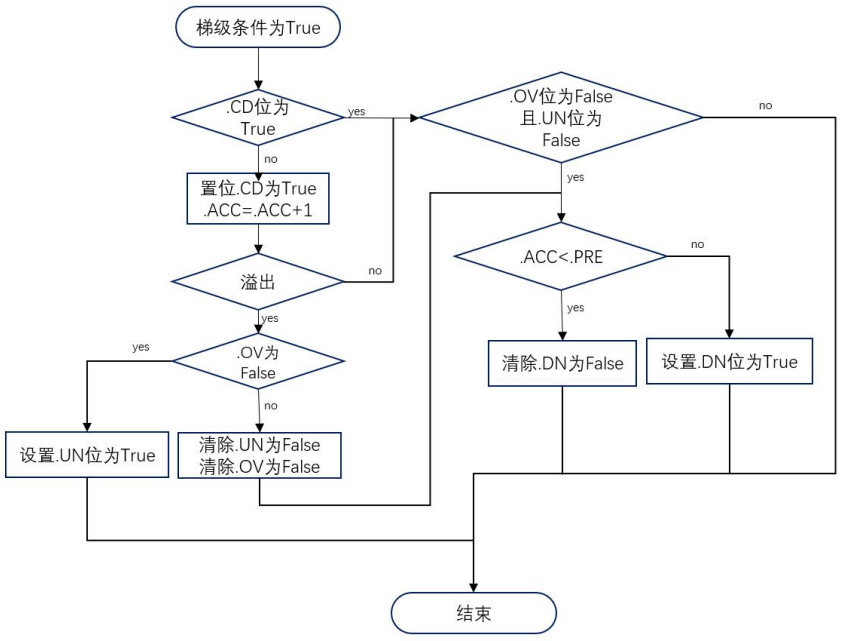
情况/状态	行为
Prescan	将.CD 位置为 True，防止第一次程序扫描期间发生无效递减。
执行条件为 FALSE	将梯级输出条件设置为梯级输入条件 请参见 流程图（真）
执行条件为 TRUE	将梯级输出条件设置为梯级输入条件 请参见 流程图（假）
Postscan	无

时序图

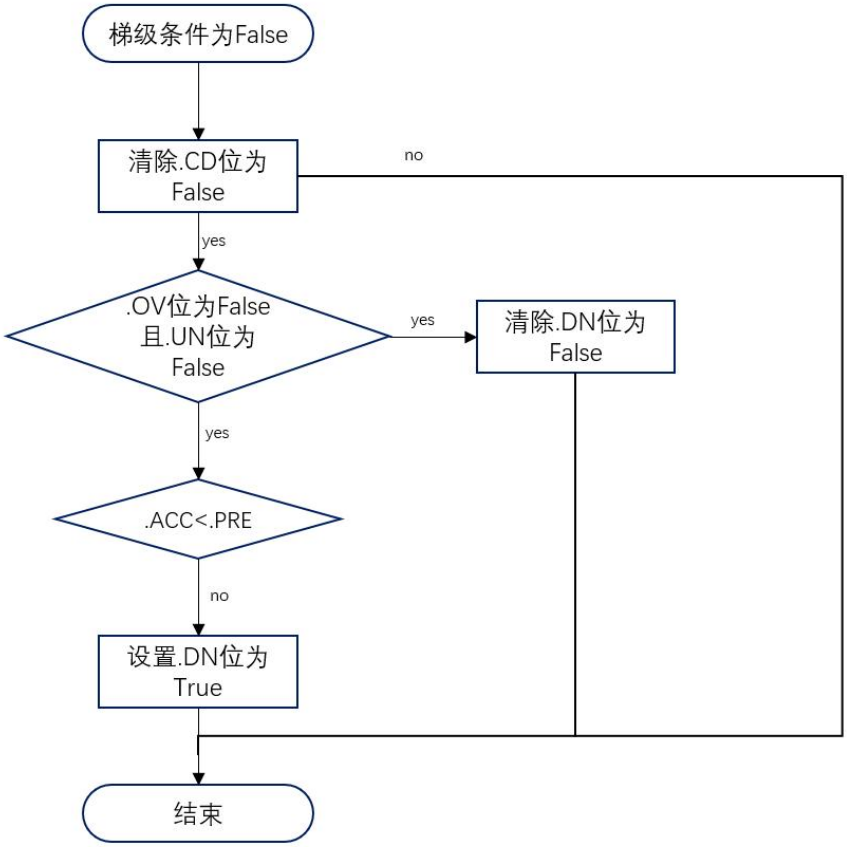


流程图

流程图（真）



流程图（假）



4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

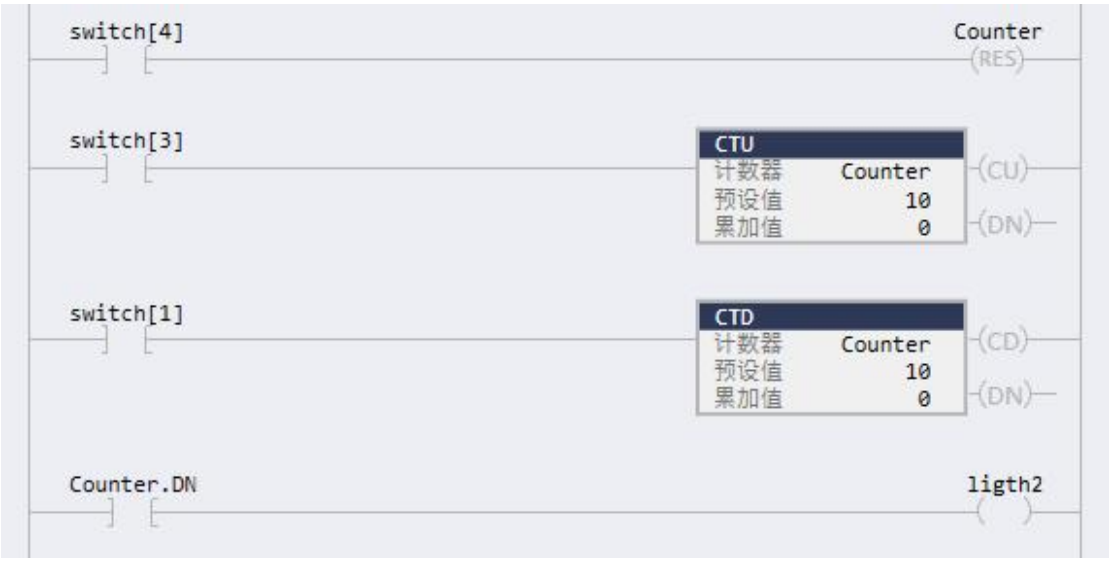
6）示例

梯形图：

传送带将零件传送到缓冲区域中。每次有零件进入缓冲区域，就使能 switch[3]且 Counter 加 1。每次有零件离开缓冲区域，就使能 switch[1]且 counter 减 1。如果缓

冲区域有 100 个零件（Counter.DN 为真），light2 将接通并阻止传送带将更多零件传入缓冲区域，直到缓冲区域有空间容纳更多零件。

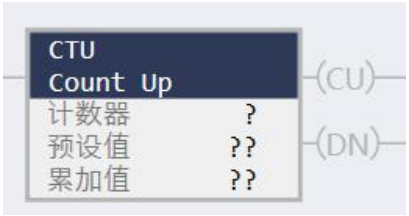
Counter	COUNTER	{...}



加计数器 CTU 指令

当梯级输入状态从 0 切换到 1 时，CTU 指令向上计数。

1）指令格式

指令	名称	梯形图表现	ST 表现
CTU	加计数器		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Counter	计数器	COUNTER	变量	计数器结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Preset	预设值	DINT	立即数	Counter.PRE 的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Accum	累加值	DINT	立即数	Counter.ACC 的值

TIMER 结构

助记码:	数据类型	说明:
.CU	BOOL	向上计数使能位，包含指令上次执行时的 梯级输入条件。

.DN	BOOL	完成位，置位时指示计数操作完成。
.OV	BOOL	上溢位，置位时指示计数器的值已增至上限 2,147,483,647 以上。
.UN		下溢位，置位时指示计数器的值已减至下限值 -2,147,483,648 以下。
.PRE	DINT	预设值指定了累积值必须达到的数值，只有在达到该数值后，指令才会显示已完成。
.ACC	DINT	累加值，指定指令已计数的跳变次数。

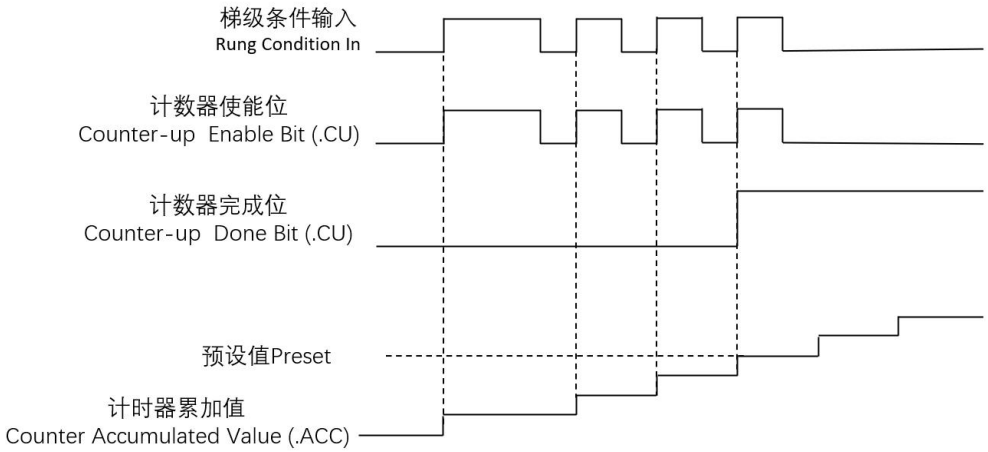
3）功能说明

当逻辑条件输入（rung-condition-in）被置为真（true），且.CU 为假（false）时，ACC 将增加 1。当逻辑条件输入为假（false）时，.CU 将被置为假（false）。

执行行为：

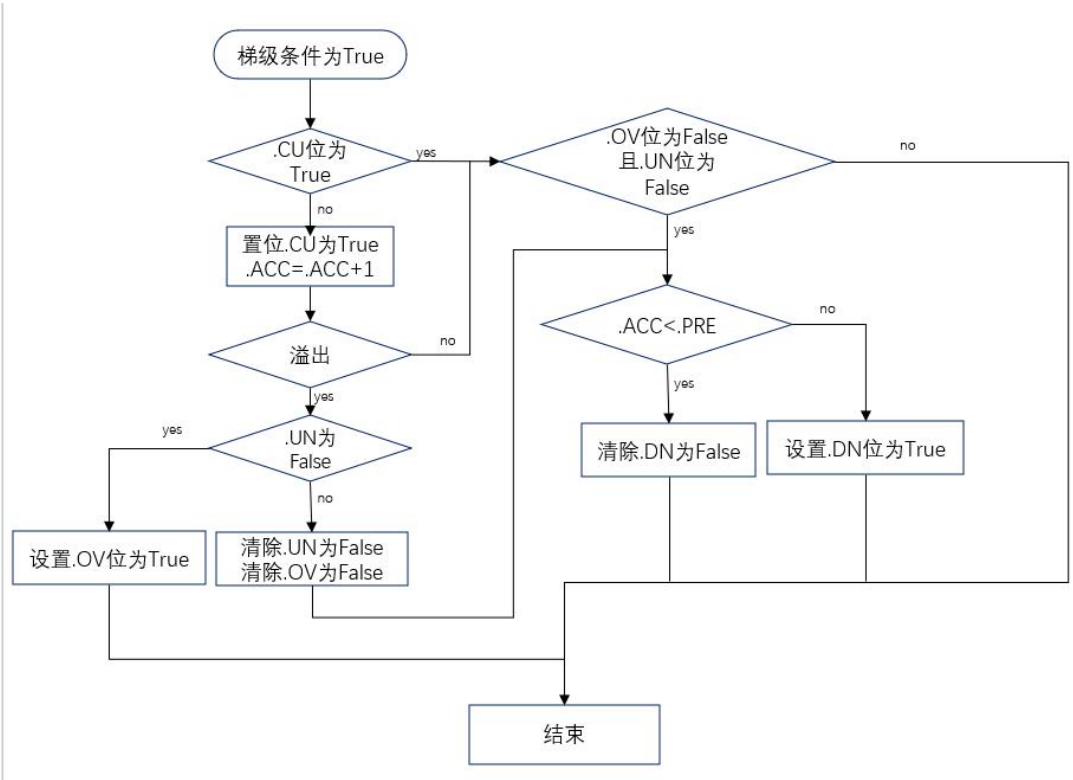
情况/状态	行为
Prescan	将.CU 位置为真（true），以防止在程序的首次扫描期间发生无效的递增操作。
执行条件为 FALSE	将梯级输出条件设置为梯级输入条件 请参见 流程图（真）
执行条件为 TRUE	将梯级输出条件设置为梯级输入条件 请参见 流程图（假）
Postscan	无

时序图

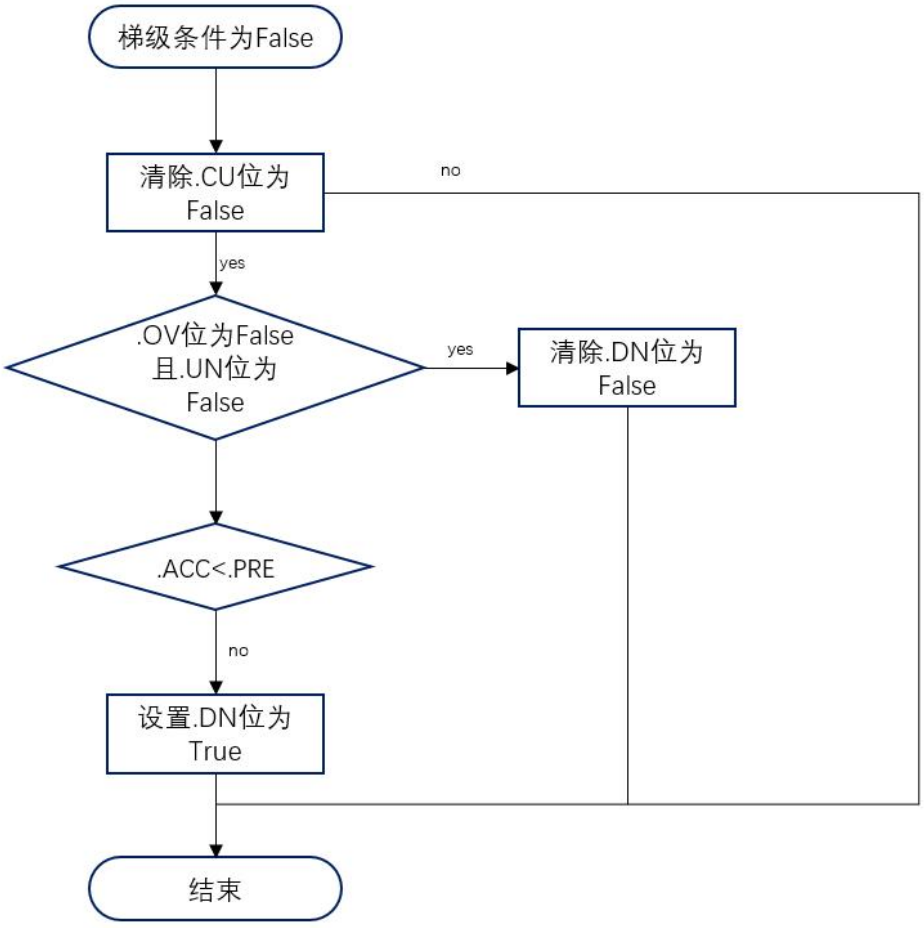


流程图

流程图（真）



流程图（假）



4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

6）示例

梯形图：

当 switch[1]从禁用状态变为使能状态 10 次之后，.DN 位置为真且 light2 接通。

如果 switch[1]继续从禁用状态变为使能状态，则 counter 计数继续增加且.DN 位保持置位状态。当 switch[2]使能后，RES 指令会复位 counter（将状态位清零并清除.ACC 值）且 light2 关闭。

加计数器/减计数器 CTUD 指令-ST

当梯级输入状态从 0 切换到 1 时，CTU 指令向上计数;当梯级输入状态从 1 切换到 0 时，CTU 指令向下计数。

1）指令格式

指令	名称	梯形图表现	ST 表现
CTUD	加计数器/减计数器	此指令不支持梯形图	CTUD(CTUD_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
CTUD_tag	FBD_COUNTER	结构	CTUD 结构

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Counter	计数器	COUNTER	变量	计数器结构
Preset	预设值	DINT	立即数	Counter.PRE 的值
Acc	累加值	DINT	立即数	Counter.ACC 的值

FBD_COUNTER 结构

输入参数	数据类型	说明:
EnableIn	BOOL	如果此参数清零，指令不会执行，也不会更新输出。如果此参数置位，指令执行。 默认置位。

CUEnable	BOOL	启用向上计数。当输入由清零切换为置位时，累加器加 1。 默认清零
CDEnable	DINT	启用向下计数。当输入由清零切换为置位时，累加器减 1。 默认清零
PRE	BOOL	计数器预设值。该值为累加值必须达到的值，当达到该值时，DN 置位。 有效值 = 任意整数 默认值为 0
Reset	BOOL	计时器复位请求。该参数置位时，计数器复位。 默认清零

输出参数	数据类型	说明:
EnableOut	BOOL	指令产生有效结果。
ACC	DINT	累加值。
CU	BOOL	已启用向上计数。
CD	BOOL	已启用向下计数。
DN	BOOL	计数完成。当累加值大于或等于预设值时置位。
OV	BOOL	计数器上溢。指示计数器超出上限值 2,147,483,647。计数器随后将翻转回 -2,147,483,648，并重新开始向下计数。
UN	BOOL	计数器下溢。指示计数器超出下限值 -2,147,483,648。计数器随后

		翻转为 2,147,483,647，并重新开始向下计数。

3）功能说明

当条件为真且 CUEnable 为真时，CTUD 指令将计数器按 1 递增。当条件为真且 CDEnable 为真时，CTUD 指令将计数器按 1 递减。

在同一个扫描期间，CUEnable 和 CDEnable 输入参数均可切换。此时，指令先执行向上计数，后执行向下计数。

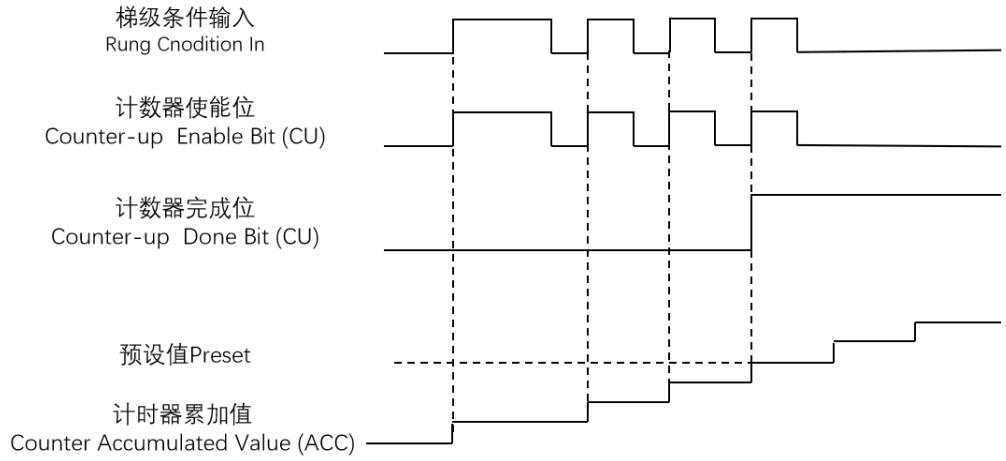
禁用后，CTUD 指令保留其累加值。Reset 输入参数置位时，该指令将复位。

执行行为：

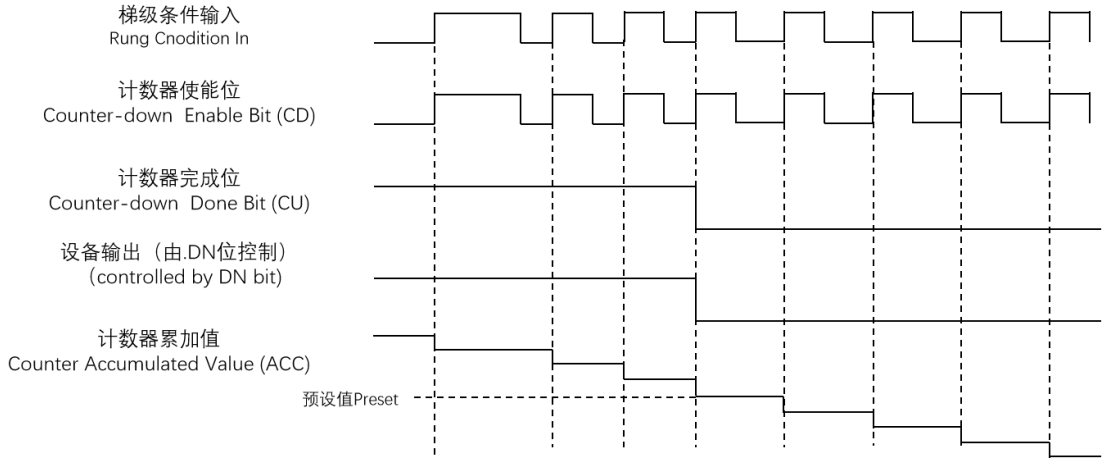
情况/状态	行为
预扫描	EnableIn 和 EnableOut 位设置为假。
EnableIn 处为清零状态	EnableIn 和 EnableOut 位设置为假。初始化数据，以等待 CuEnable 或 CdEnable 的“0 到 1”跳变，来影响 ACC 值。
EnableIn 处为置位状态	EnableIn 和 EnableOut 位设置为真。 指令执行
指令首次运行	初始化数据，以等待 CuEnable 或 CdEnable 的“0 到 1”跳变，来影响 ACC 值。
指令首次扫描	初始化数据，以等待 CuEnable 或 CdEnable 的“0 到 1”跳变，来影响 ACC 值。
后扫描	EnableIn 和 EnableOut 位设置为假。

时序图

递增



递减



4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

6）示例

当 limit_switch1 从清零状态转为置位状态时，为一次扫描设置

CUEnable 且 CTUD 指令按 1 递增 ACC 值。当 ACC≥PRE 时，设置

DN 参数，这将在 CTUD 指令之后启用功能块指令。

ST 示例：

```
CTUD_01.Preset:=500;
CTUD_01.Reset:=Restart;
CTUD_01.CUEnable:=limit_switch1;
CTUD(CTUD_01);
counter_state:=CTUD_01.DN;
```

复位指令 RES 指令

RES 指令重置 TIMER、COUNTER 或 CONTROL 结构体。

1）指令格式

指令	名称	梯形图表现	ST 表现
RES	复位指令		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Structure	结构体	TIMER CONTROL COUNTER	变量	RES 结构

3）功能说明

当条件为真时，RES 指令将这些元素置 0：

当使用一个复位（RES）指令用于:	说明:
TIMER	.ACC 值清 0 控制状态位置为 False。
CONTROL	.ACC 值清 0 控制状态位置为 False。
COUNTER	.POS 值清 0 控制状态位置为 False。

执行行为：

情况/状态	行为
Prescan	无。
执行条件为 FALSE	将梯级输出条件设置为梯级输入条件。
执行条件为 TRUE	将梯级输出条件设置为梯级输入条件。 将指定结构复位。
Postscan	无。

4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

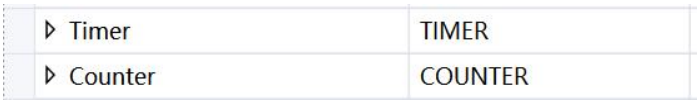
5）错误说明

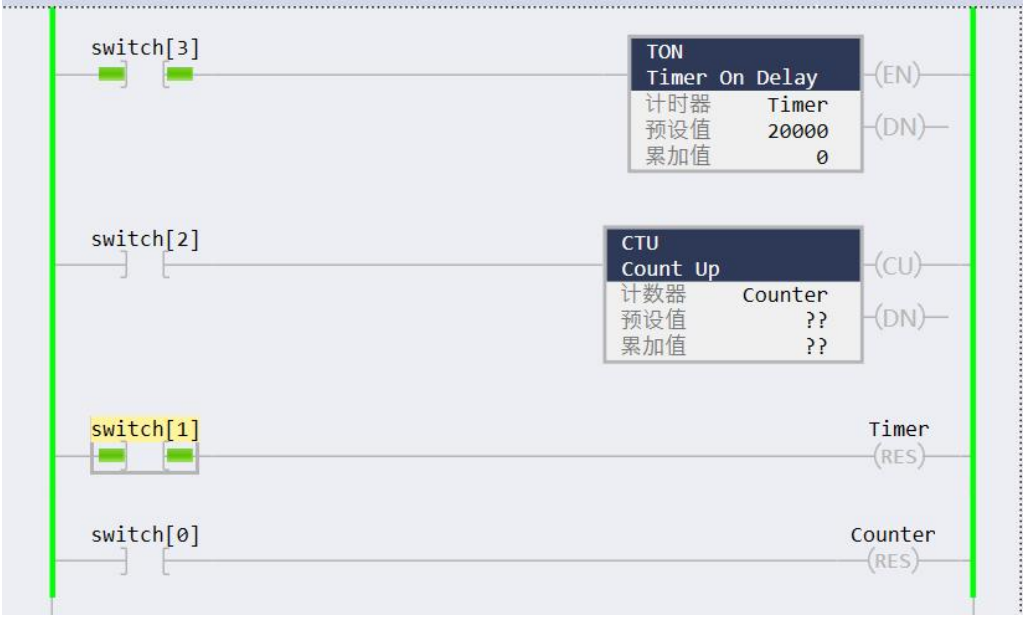
6）示例

梯形图：

Switch[1]接通，Timer 复位；

Switch[0]接通，Counter 复位；

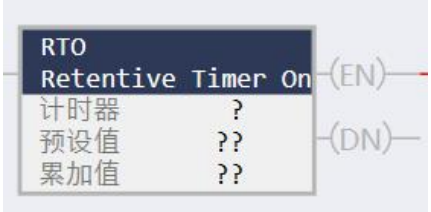



switch[3]
TON
Timer On Delay
计时器 Timer
预设值 20000
累加值 0
(EN)
(DN)
switch[2]
CTU
Count Up
计数器 Counter
预设值 ??
累加值 ??
(CU)
(DN)
switch[1]
Timer
(RES)
switch[0]
Counter
(RES)

保持型定时器 RTO 指令

RTO 指令是一个保持型定时器;当时间使能复位(0→1)时，时间累加。

1）指令格式

指令	名称	梯形图表现	ST 表现
RTO	保持型定时器		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Timer	计时器	TIMER	变量	计时器结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Preset	预设值	DINT	立即数	Timer.PRE 的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Accum	累加值	DINT	立即数	Timer.ACC 的值

TIMER 结构

助记码:	数据类型	说明:
.EN	BOOL	使能位：包含上次执行指令时的梯级输入条件。

.TT	BOOL	计时位，置位时指示计时操作正在进行。
.DN	BOOL	完成位，置位时指示计时操作完成（或暂停）。
.PRE	DINT	预设值指定了累积值必须达到的数值（以 1 毫秒为单位），在达到该数值之前，指令不会显示已完成。
.ACC	DINT	累加值，指定自 RTO 指令使能起经过的时间（毫秒）。

3）功能说明

RTO 指令会累积时间，直到以下情况发生：

- 计时器被禁用。
- 计时器完成计时。

计时器的时间基准始终为 1 毫秒。例如，对于一个 2 秒的计时器，应将.PRE 值设置为 2000。

当计时器完成时，计时器会将.DN 位置为真（true）。

当被使能时，可以通过将.DN 位置为真（true）来暂停计时，并通过将.DN 位置为假（false）来恢复计时。

计时器工作原理：

计时器工作时会用当前时间减去上次扫描的时间：

ACC = ACC + (current_time - last_time_scanned)

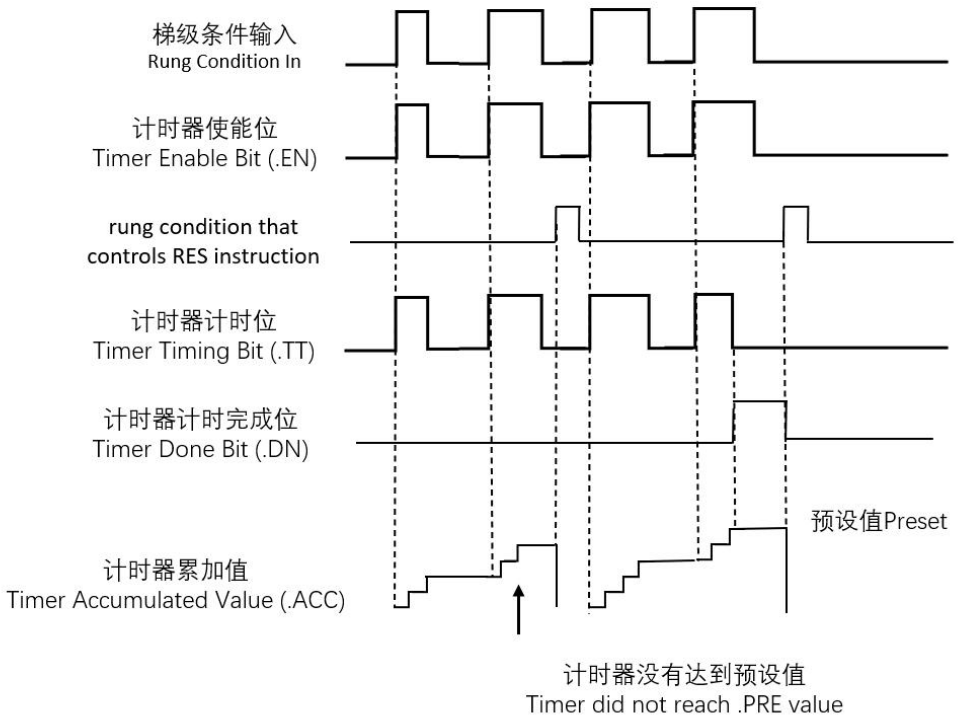
更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

执行行为：

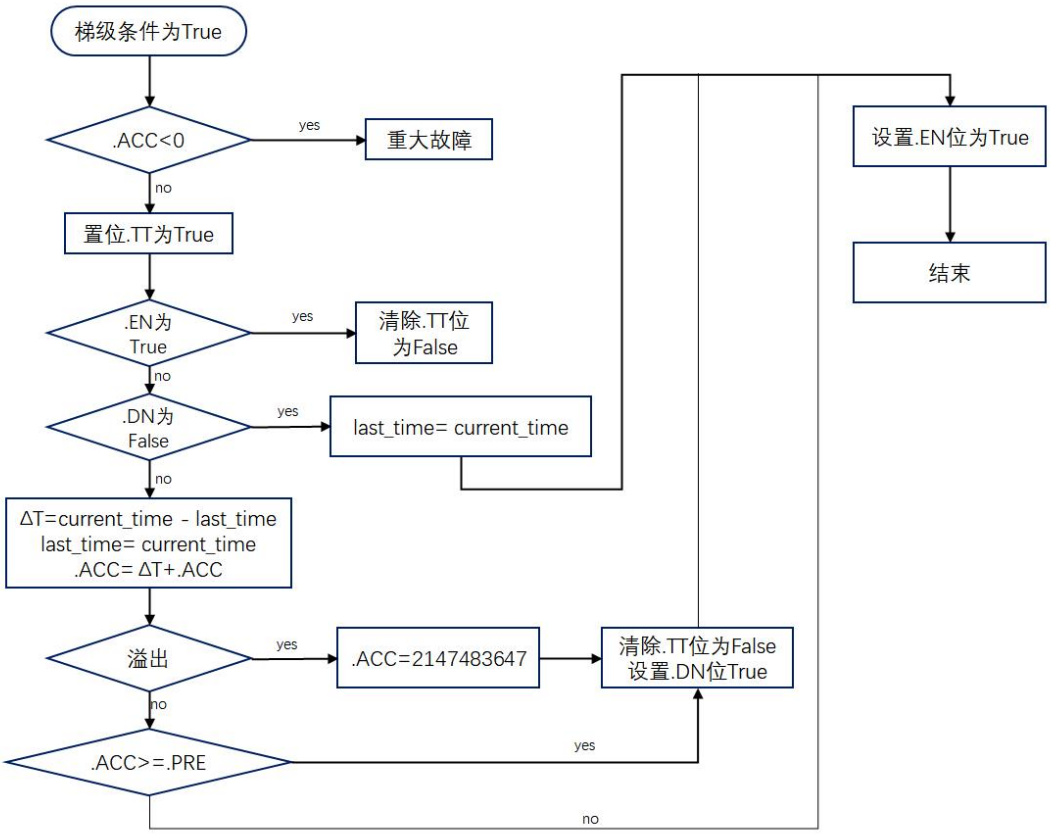
情况/状态	行为

Prescan	**.EN 位置为假（false）。** **.TT 位置为假（false）。**
执行条件为 FALSE	**将梯级输出条件设置为梯级输入条件** **.EN 位置为假（false）。** **.TT 位置为假（false）。**
执行条件为 TRUE	**将梯级输出条件设置为梯级输入条件** **请参见 RTO 流程图（真）**
Postscan	**无**

时序图



流程图（真）



4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

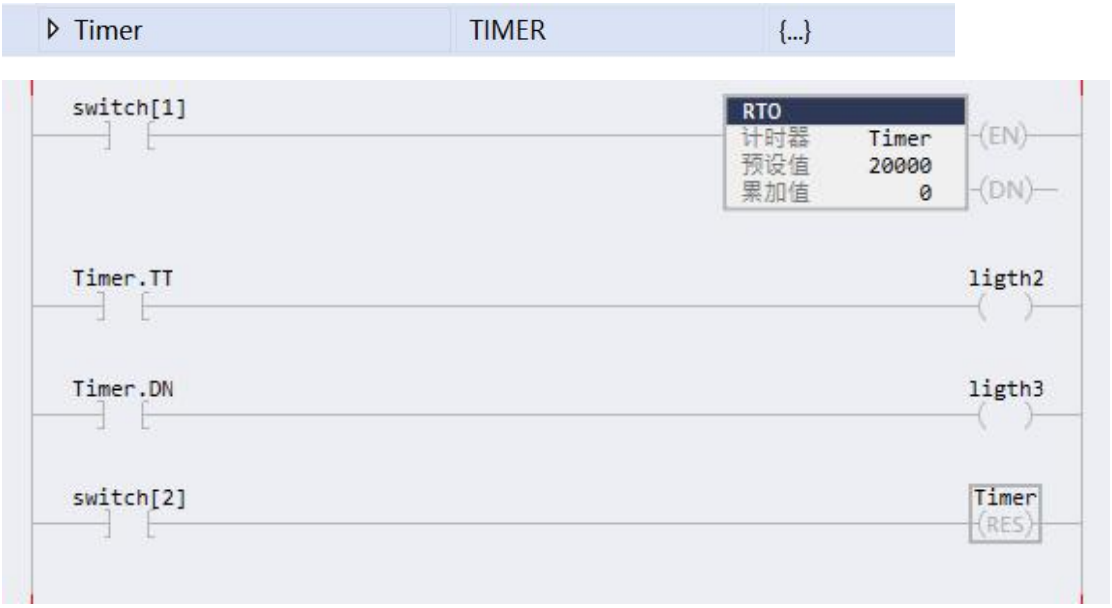
5）错误说明

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE<0	4	34
.ACC<0	4	34

6）示例

梯形图：

当 switch[1]] 置位时，light2 接通并持续 20000 毫秒（Timer 计时 ）。当 Timer.Acc 达到 20000 时，light2 断开，light3 接通。Light3 保持接通，直至 timer3 复位。如果在 Timer 计时期间 switch[1]] 清零，则 light2 断开。当 switch[2]] 置位时，RES 指令会将 Timer 复位（清除状态位和 .ACC 值）。



关断延时定时器 TOF 指令

TOF 指令是一个非保持型定时器;当时间使能复位(1→0)时，时间累加。

1）指令格式

指令	名称	梯形图表现	ST 表现
TOF	关断延时定时器		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Timer	计时器	TIMER	变量	计时器结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Preset	预设值	DINT	立即数	Timer.PRE 的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Accum	累加值	DINT	立即数	Timer.ACC 的值

TIMER 结构

助记码:	数据类型	说明:
.EN	BOOL	使能位：包含上次执行指令时的梯级输入条件。

.TT	BOOL	当计时位被置 1 时，表示计时操作正在进行中
.DN	BOOL	当完成位（Done bit）被置 1 时，表示计时操作已完成（或已暂停）
.PRE	DINT	预设值指定了累积值必须达到的数值（以 1 毫秒为单位），在达到该数值之前，指令不会显示已完成。
.ACC	DINT	累积值指自 TOF 指令被使能以来经过的毫秒数。

3）功能说明

TOF 指令会累积时间，直到以下情况发生：

- 计时器被禁用。
- 计时器完成计时。

计时器的时间基准始终为 1 毫秒。例如，对于一个 2 秒的计时器，应将.PRE 值设置为 2000。

当计时器完成时，计时器会将.DN 位置为假（false）。

当被使能时，可以通过将.DN 位置为假（false）来暂停计时，并通过将.DN 位置为真（true）来恢复计时。

计时器工作原理：

计时器工作时会用当前时间减去上次扫描的时间：

ACC = ACC + (current_time - last_time_scanned)

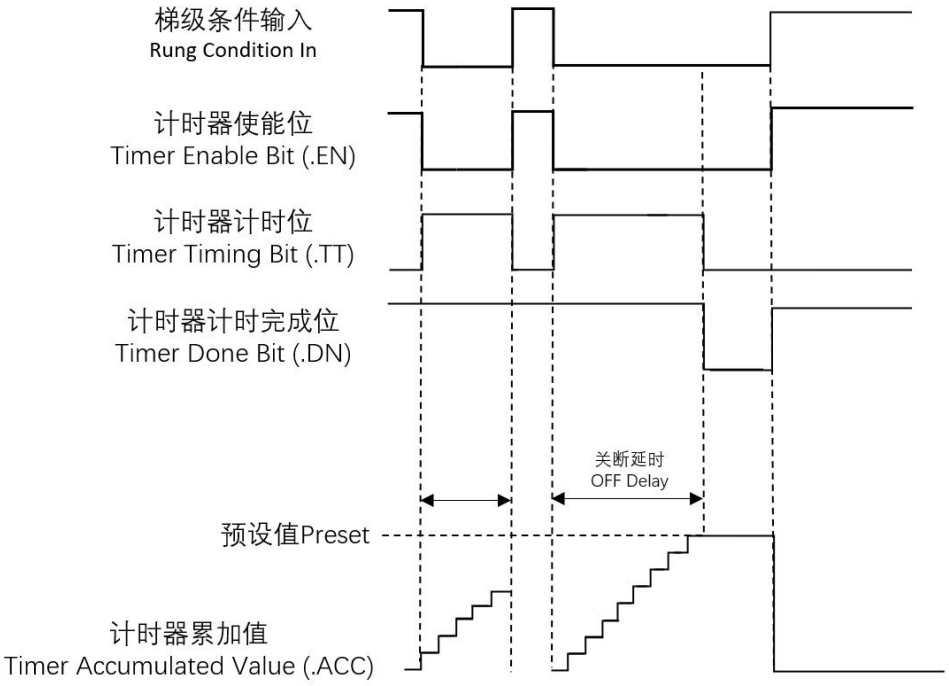
更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

执行行为：

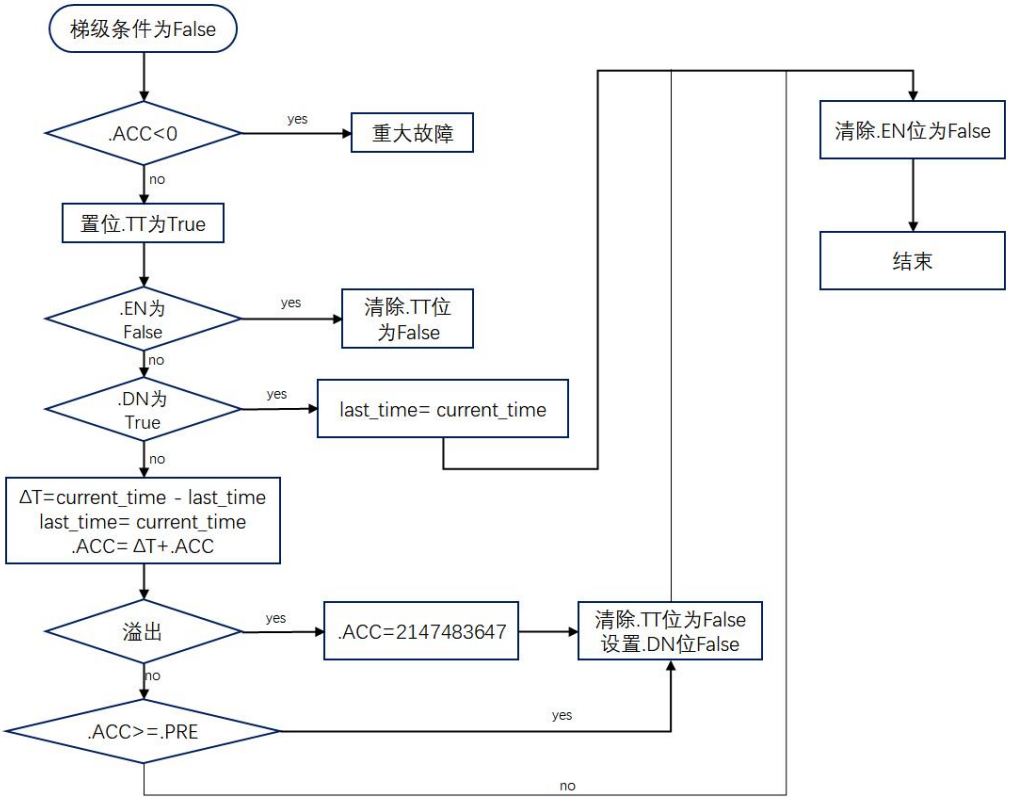
情况/状态	行为

Prescan	**.EN 位置为假（false）。** **.TT 位置为假（false）。** **.DN 位置为假（false）。** **.ACC 值设置为等于.PRE 值。**
执行条件为 FALSE	**将梯级输出条件设置为梯级输入条件** **请参见 流程图（真）**
执行条件为 TRUE	**将梯级输出条件设置为梯级输入条件** **.EN 位置为真（true）。** **.TT 位置为假（false）。** **.DN 位置为真（true）。** **.ACC 值清零。**
Postscan	**.EN 位置为假（false）。** **.TT 位置为假（false）。** **.DN 位置为假（false）。** **.ACC 值清零。**

时序图


梯级条件输入
Rung Condition In
计时器使能位
Timer Enable Bit (.EN)
计时器计时位
Timer Timing Bit (.TT)
计时器计时完成位
Timer Done Bit (.DN)
关断延时
OFF Delay
预设值Preset
计时器累加值
Timer Accumulated Value (.ACC)

流程图（真）


梯级条件为False
.ACC<0
yes
重大故障
no
置位.TT为True
.EN为
False
yes
清除.TT位
为False
no
.DN为
True
yes
last_time= current_time
no
ΔT=current_time - last_time
last_time= current_time
.ACC= ΔT+.ACC
溢出
yes
.ACC=2147483647
清除.TT位为False
设置.DN位False
no
.ACC>=.PRE
yes
no
清除.EN位为False
结束

4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE<0	4	34
.ACC<0	4	34

6）示例

梯形图：

当 switch[1] 清零时，light2 接通并持续 20000 毫秒（Timer 计时 ）。当 Timer.Acc 达到 20000 时，light2 断开，light3 接通。Light3 保持接通，直到 TOF 指令使能。如果在 Timer 计时期间 switch[1] 为真，则 light2 断开。

带复位的关断延时定时器 TOFR 指令-ST

TOFR 指令是非保持计时器，它在清除 TimerEnable 时累计时间。

1）指令格式

指令	名称	梯形图表现	ST 表现
TOFR	带有复位的关断延时定时器	此指令在梯形图中以两个独立指令提供：TOF 和 RES	TOFR(TOFR_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
TOFR_tag	FBD_TIMER	结构	TONR 结构

FBD_TIMER 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	说明:
EnableIn	已使能	BOOL	无影响。指令执行。
TimerEnable	计时启动	BOOL	如果为清零状态，将启用计时器，使之运行并累计时间。 默认为清零状态。
PRE	预设时间	DINT	计时器预设值。这是在计时完成前 ACC 必须达到的值（以毫秒为单位）。如果无效，该指令将设置

			Status 中相应位，计时器不执行。 Valid=0 表示设置为最大的正整数。
Reset	复位	BOOL	请求复位计时器。设置时，计时器复位。 默认为清零状态。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	说明:
EnableOut	使能输出	BOOL	指令产生有效结果。
ACC	累加时间	BOOL	以毫秒表示的累计时间。
EN	已启用	BOOL	启用计时器的输出。指示计时器指令已启用。
TT	计时中	BOOL	计时器计时输出。设置时，表示计时器操作正在进行。
DN	计时完成	BOOL	计时器完成输出。指示累计值何时大于或等于预设值。
Status	状态	DINT	功能块状态。
InstructFault (Status.0)	错误	BOOL	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其余状态位以确定发生的情况。
PresetInv (Status.1)	预设值无效	BOOL	预设值无效。

3）功能说明

TOFR 指令会累积时间，直到以下情况发生：

- 计时器被禁用。
- 计时器完成计时。

计时器的时间基准始终为 1 毫秒。例如，对于一个 2 秒的计时器，应将.PRE 值设置为 2000。

设置 Reset 输入参数可复位该指令。如果在设置 Reset 时清除了 TimerEnable 则 TOFR 指令在清除 Reset 时不再次开始计时。

当计时器完成时，计时器会将.DN 位清除为假（false）。

当被使能时，可以通过将.DN 位清除为假（false）来暂停计时，并通过将.DN 位设置为真（true）来恢复计时。

计时器工作原理：

计时器工作时会用当前时间减去上次扫描的时间：

ACC = ACC + (current_time - last_time_scanned)

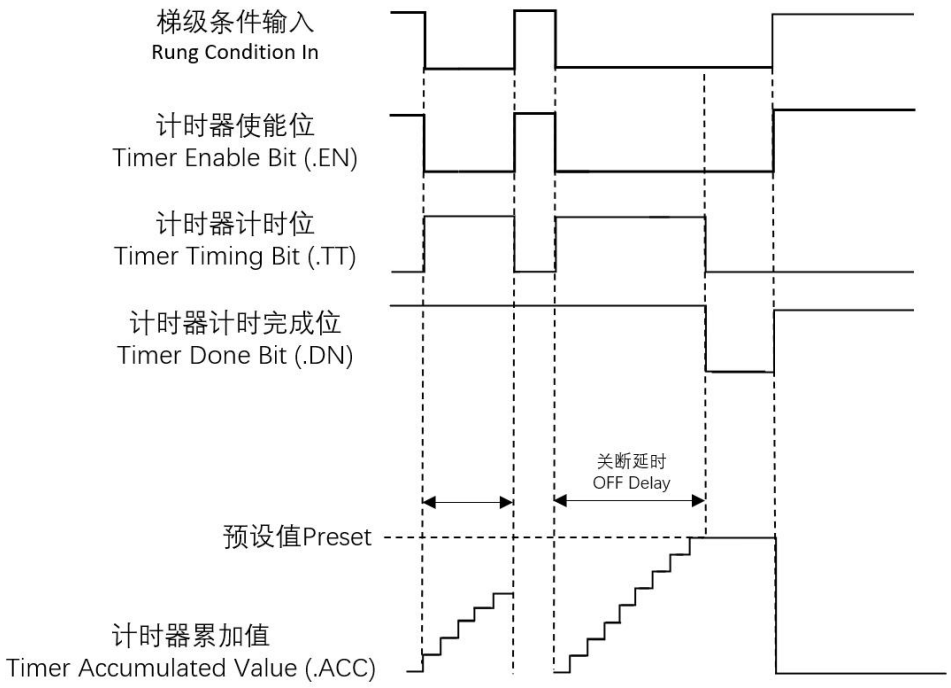
更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

执行行为：

情况/状态	行为
预扫描	不采取任何操作。
指令第一次扫描	清除 EN、TT 和 DN。 将 ACC 值设置为 PRE。
指令第一次运行	清除 EN、TT 和 DN。 将 ACC 值设置为 PRE。
Enableln 处于清零状态	不可用

Enableln 处于置位状态	**Enableln 始终为置位状态。** **指令执行。**
复位	**设置 Reset 输入参数时，该指令清除 EN、** **TT 和 DN 并将 ACC 设置为 PRE。**
后扫描	**不采取任何操作**

时序图



4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE<0	4	34
.ACC<0	4	34

6）示例

ST 示例：

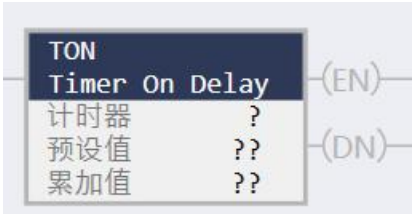
在清除 limit_switch1 后的每次扫描中，TOFR 指令按经过的时间递增 ACC 值，直到 ACC 值达到 PRE 值。当 ACC≥PRE 时，清除 DN 参数，并设置 timer_state2。

```
TOFR_01.Preset :=500;
TOFR_01.Reset:=reset;
TOFR_01.TimerEnable:=limit_switch1;
TOFR(TOFR_01);
timer_state2:=TOFR_01.DN;
```

接通延时定时器 TON 指令

TON 指令是一个非保持型定时器;当时间使能置位(0→1)时，时间累加。

1）指令格式

指令	名称	梯形图表现	ST 表现
TON	接通延时定时器		此指令不支持结构化文本。

2）相关变量

输入输出变量(InOut)

参数(英文)	参数(中文)	数据类型	格式	说明
Timer	计时器	TIMER	变量	计时器结构
Preset	预设值	DINT	立即数	Timer.PRE 的值
Accum	累加值	DINT	立即数	Timer.ACC 的值

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Preset	预设值	DINT	立即数	Timer.PRE 的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Accum	累加值	DINT	立即数	Timer.ACC 的值

TIMER 结构

助记码:	数据类型	说明:
.EN	BOOL	使能位：包含上次执行指令时的逻辑行条件。
.TT	BOOL	当计时位被置 1 时，表示计时操作正在进行中
.DN	BOOL	当完成位（Done bit）被置 1 时，表示计时操作已完成（或已暂停）
.PRE	DINT	预设值指定了累积值必须达到的数值（以 1 毫秒为单位），在达到该数值之前，指令不会显示已完成。
.ACC	DINT	累积值指自 TON 指令被使能以来经过的毫秒数。

3）功能说明

TON 指令从被使能开始累积时间，直到以下情况发生：

- 计时器被禁用。
- 计时器完成计时。

计时器的时间基准始终为 1 毫秒。例如，对于一个 2 秒的计时器，应将.PRE 值设置为 2000。

当计时器完成时，计时器会将.DN 位置为真（true）。

当被使能时，可以通过将.DN 位置为真（true）来暂停计时，并通过将.DN 位置为假（false）来恢复计时。

计时器工作原理：

计时器工作时会用当前时间减去上次扫描的时间：

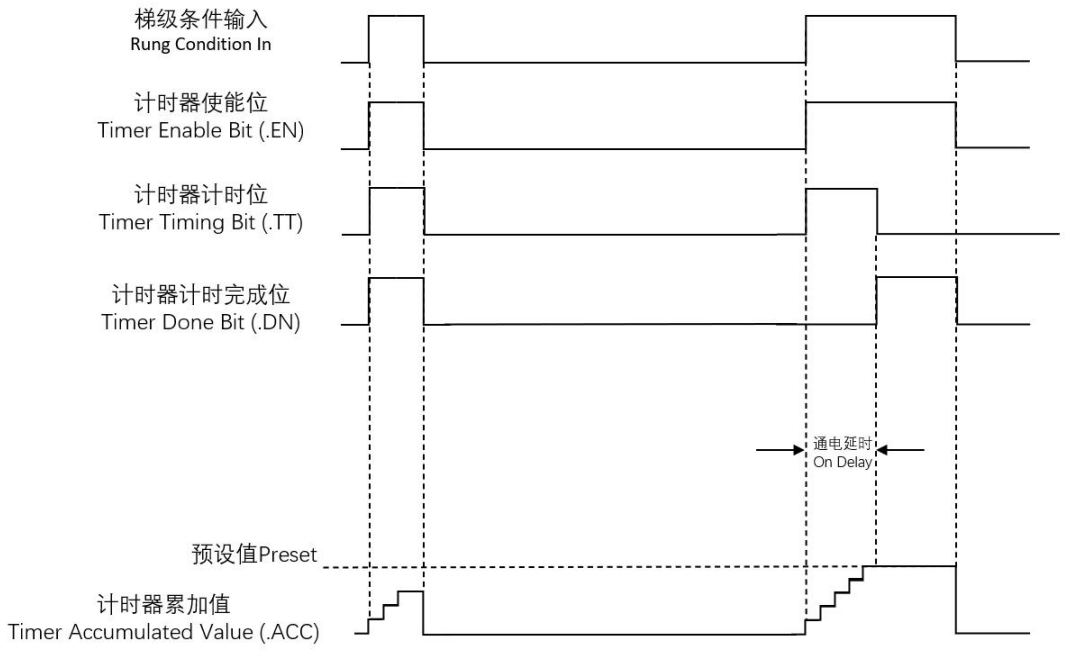
ACC = ACC + (current_time - last_time_scanned)

更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

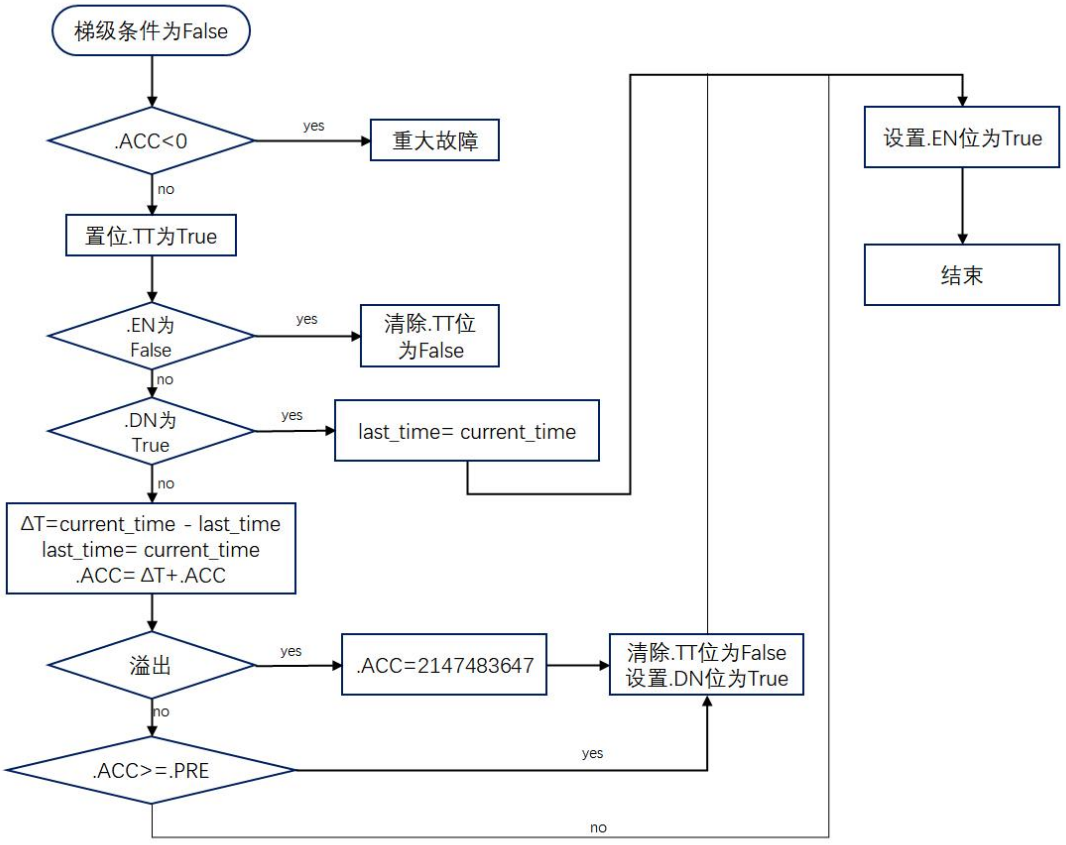
执行行为：

情况/状态	行为
Prescan	.EN 位置为假（false）。 .TT 位置为假（false）。 .DN 位置为假（false）。 .ACC 值清零。
执行条件为 FALSE	将梯级输出条件设置为梯级输入条件 .EN 位置为假（false）。 .TT 位置为假（false）。 .DN 位置为假（false）。 .ACC 值清零。
执行条件为 TRUE	将梯级输出条件设置为梯级输入条件 请参见 流程图（真）
Postscan	.EN 位置为假（false）。 .TT 位置为假（false）。 .DN 位置为假（false）。 .ACC 值清零（false）。

时序图：


梯级条件输入
Rung Condition In
计时器使能位
Timer Enable Bit (.EN)
计时器计时位
Timer Timing Bit (.TT)
计时器计时完成位
Timer Done Bit (.DN)
通电延时
On Delay
预设值Preset
计时器累加值
Timer Accumulated Value (.ACC)

流程图（真）


梯级条件为False
.ACC<0
yes
重大故障
no
置位.TT为True
.EN为
False
yes
清除.TT位
为False
no
.DN为
True
yes
last_time= current_time
no
ΔT=current_time - last_time
last_time= current_time
.ACC= ΔT+.ACC
溢出
yes
.ACC=2147483647
清除.TT位为False
设置.DN位为True
no
.ACC>=.PRE
yes
no
设置.EN位为True
结束

4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE<0	4	34
.ACC<0	4	34

6）示例

梯形图：

当 switch[1] 置为真时，light2 接通并持续 20000 毫秒（ Timer 计时）。当 Timer.Acc 达到 20000 时，light2 断开，light3 接通。如果在 Timer 计时期间 switch[10]设置为假，则 light2 断开。当 switch[1] 设置为假时，Timer 状态位和 .ACC 值复位。

带复位的接通延时定时器 TONR 指令-ST

TONR 指令是非保持计时器，它在设置 TimerEnable 时累计时间。

1）指令格式

指令	名称	梯形图表现	ST 表现
TONR	带有复位的接通延时定时器	此指令在梯形图中以两个独立指令提供：TON 和 RES	TONR(TONR_tag);

2）相关变量

结构化文本

变量	数据类型	格式	说明
TOFR_tag	FBD_TIMER	结构	TONR 结构

FBD_TIMER 结构

输入变量(Input)

参数(英文)	参数(中文)	数据类型	说明:
EnableIn	已使能	BOOL	无影响。指令执行。
TimerEnable	计时启动	BOOL	如果为清零状态，将启用计时器，使之运行并累计时间。 默认为清零状态。
PRE	预设时间	DINT	计时器预设值。这是在计时完成前 ACC 必须达到的值（以毫秒为单位）。如果无效，该指令将设置 Status 中相应位，计时器不执行。

			Valid=0 表示设置为最大的正整数。
Reset	复位	BOOL	请求复位计时器。设置时，计时器复位。 默认为清零状态。

输出变量(Output)

参数(英文)	参数(中文)	数据类型	说明:
EnableOut	使能输出	BOOL	指令产生有效结果。
ACC	累加时间	BOOL	以毫秒表示的累计时间。
EN	已启用	BOOL	启用计时器的输出。指示计时器指令已启用。
TT	计时中	BOOL	计时器计时输出。设置时，表示计时器操作正在进行。
DN	计时完成	BOOL	计时器完成输出。指示累计值何时大于或等于预设值。
Status	状态	DINT	功能块状态。
InstructFault (Status.0)	错误	BOOL	该指令检测到以下执行错误之一。这不是轻微或严重的控制器错误。检查其余状态位以确定发生的情况。
PresetInv (Status.1)	预设值无效	BOOL	预设值无效。

3）功能说明

TONR 指令从被使能开始累积时间，直到以下情况发生：

- 计时器被禁用。
- 计时器完成计时。

计时器的时间基准始终为 1 毫秒。例如，对于一个 2 秒的计时器，应将.PRE 值设置为 2000。

设置 Reset 输入参数可复位该指令。如果在设置 Reset 时设置了 TimerEnable 则 TONR 指令在清除 Reset 时再次开始计时。

当计时器完成时，计时器会将.DN 位设置为真（true）。

当被使能时，可以通过将.DN 位设置为真（true）来暂停计时，并通过将.DN 位清除为假（false）来恢复计时。

计时器工作原理：

计时器工作时会用当前时间减去上次扫描的时间：

ACC = ACC + (current_time - last_time_scanned)

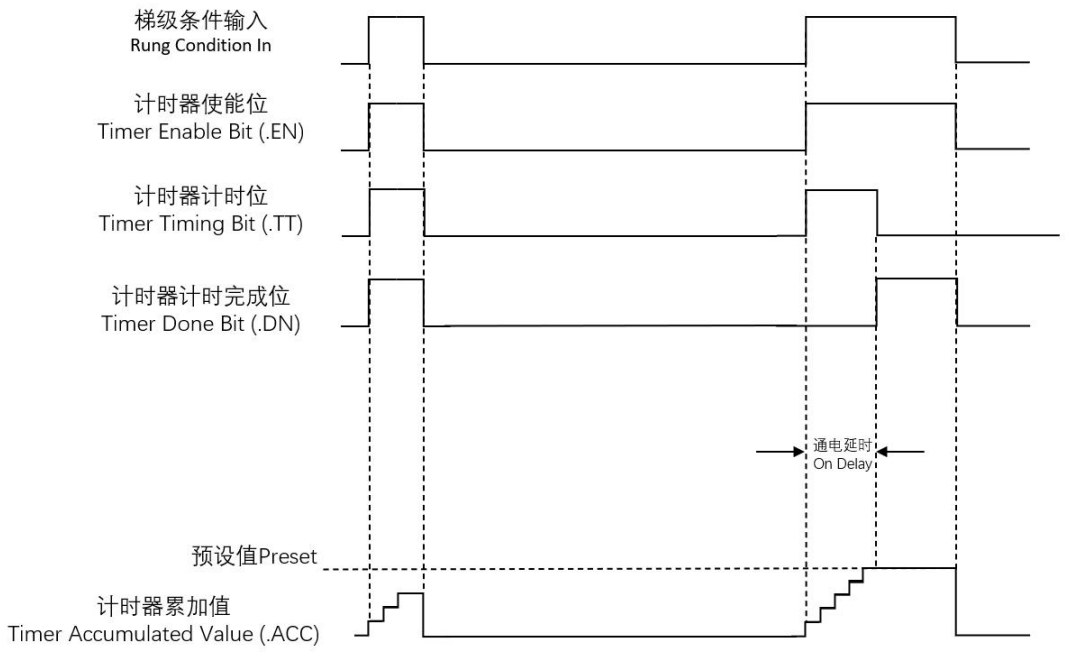
更新 ACC 后，计时器设置 last_time_scanned = current_time，从而使计时器为下一次扫描做好准备。

执行行为：

情况/状态	行为
预扫描	不采取任何操作。
指令第一次扫描	清除 EN、TT 和 DN。 将 ACC 值设置为 0。
指令第一次运行	清除 EN、TT 和 DN。 将 ACC 值设置为 0。
Enableln 处于清零状态	不可用

EnableIn 处于置位状态	**EnableIn 始终为置位状态。** **指令执行。**
复位	**设置 Reset 输入参数时，该指令清除 EN、** **TT 和 DN 并将 ACC 设置为零。**
后扫描	**不采取任何操作**

时序图



4）注意事项

如果出现以下情况，可能会导致意外操作：

- **输出变量操作数被改写。**
- **结构操作数的组成部分被改写。**
- **除非另有说明，否则结构操作数由多条指令共享。**

5）错误说明

在以下情况下会发生严重故障：	故障类型	故障代码
.PRE<0	4	34
.ACC<0	4	34

6）示例

ST 示例：

在设置了 limit_switch1 的每次扫描中，TONR 指令按经过的时间递增 ACC 值，直到 ACC 值达到 PRE 值。当 ACC≥PRE 时，设置 DN 参数，并设置 timer_state。

```
TONR_01.Preset:=500;
TONR_01.Reset:=reset;
TONR_01.TimerEnable:=limit_switch1;
TONR(TONR_01);
timer_state:= TONR_01.DN;
```

脉冲计时器 TP 指令

TP 指令将生成脉冲计时器。

1）指令格式

指令	名称	梯形图表现	ST 表现
TP	脉冲计时器		TP(IN:=bool_in, PT:=dint_in, Q=>bool_out, ET=>dint_out);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
IN	输入	BOOL	立即数 变量	FALSE 变 TRUE 时开始计时
PT	设定时间	UDINT	立即数 变量	设定时间,英文全称 Preset Time

输出变量(Output)

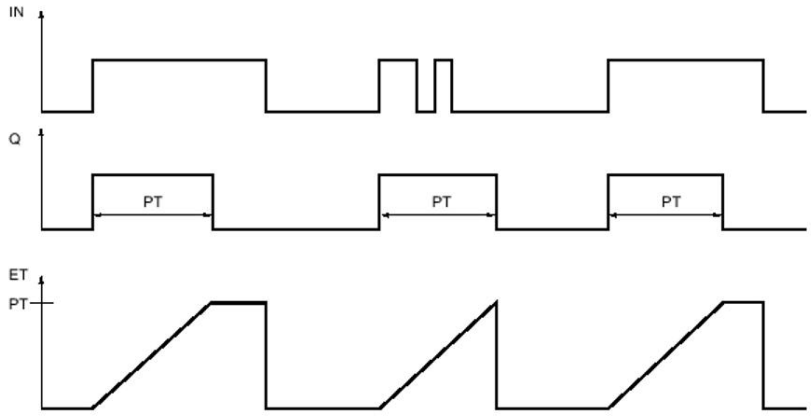
参数(英文)	参数(中文)	数据类型	格式	说明
Q	输出	BOOL	立即数 变量	TRUE 变 FALSE 计时完成, TP 的输出标志位
ET	计时时间	UDINT	立即数 变量	已计时的时间,英文全称 Elapsed Time

3）功能说明

执行行为：

情况	行为
Prescan 预扫描	IN, Q 清零，ET 清零
执行条件为 FALSE	指令内所有状态冻结
执行条件为 TRUE	按 7.3.4 功能说明时序图工作
Postscan 后扫描	IN, Q 清零，ET 清零

时序图：



当"IN"操作数的信号状态从“0"变为"1”时，

按”PT”(Preset Time)参数预设的时间开始计时，

"Q” 操作数置位为 1

当前计时时间值存储在"ET”(Elapsed Time)操作数中。

定时器计时 ET 达到 PT 时计时结束，

操作数“Q”的信号状态复位为 0。

定时器一旦开始计时，中途不受 IN 状态的影响，直到 ET 达到 PT 计时结束；

定时器计时 ET 达到 PT 时计时结束，如果 IN 持续为 1，ET 也不再增加。

4）注意事项

1、参数类型不匹配编译报错；

2、操作数 PT 使用了错误的立即数，例如：负数或者>4294967295；

5）错误说明

6）示例

梯形图：

如下图：Switch[4]切换为 true 时，TP1 并未开始计时，dint_out 依旧为 0，bool_out 也为 false。

当 Switch[4]为 true，并 Switch[1]也为 true 时，TP1 开始计时，bool_out 为 true。

当 TP1 开始计时后，Switch[1]为 false，TP1 依旧继续计时，直到 dint_out 累计值等于 dint_in 设定时间，dint_out 才为 0, bool_out 为 false。

如果 TP1 开始计时后，Switch[4]为 false，则 TP1 暂停计时，直到 Switch[4]重新 true，TP1 恢复计时。

ST 示例：

```
TP(IN:=bool_in,
   PT:=dint_in,
   Q=>bool_out,
   ET=>dint_out);
```

十八、三角函数

计算反正弦值 ACOS 指令

ACOS 指令计算源值的反余弦值（弧度）。

1）指令格式

指令	名称	梯形图表现	ST 表现
ACOS	计算反正弦值		Dest:=ACOS(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	要转换为反余弦的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT	变量	存储计算结果的变

		INT DINT LINT REAL LREAL		量

3）功能说明

使能后，ACOS 指令可计算 Source 值的反余弦值，并将结果存储在 Destination 中（以弧度为单位）

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的反余弦值。
后扫描	不适用

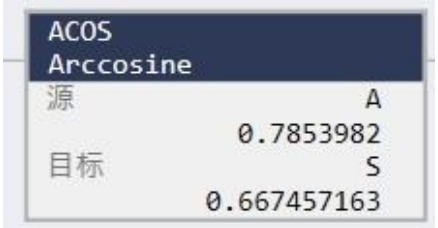
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 0.7853982，ASIN 启用时，ASIN 指令对 A 的值求正切值，并把求到数值“0.667457163”存储到 S 里



ST 示列：

```
S:= ACOS(A);
```

计算反正弦值 ASIN 指令

ASIN 指令计算源值的反正弦值（弧度）。

1）指令格式

指令	名称	梯形图表现	ST 表现
ATAN	计算反正弦值		Dest:=ASIN(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	要转换为反正弦的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明

Dest	目标	SINT INT DINT LINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，ASIN 指令可计算 Source 值的反余弦值，并将结果存储在 Destination 中（以弧度为单位）

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的反正弦值。
后扫描	不适用

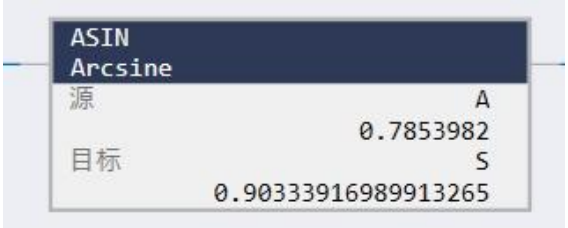
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 0.7853982，ASIN 启用时，ASIN 指令对 A 的值求正切值，并把求到数值“0.90333916989913265”存储到 S 里



ST 示列：

```
S:= ASIN(A);
```

计算反正切值 ATAN 指令

ATAN 指令计算源值的反正切值（弧度）。

1）指令格式

指令	名称	梯形图表现	ST 表现
ATAN	计算反正切值		Dest:=ATAN(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	要转换为反正弦的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT REAL LREAL	变量	存储计算结果的变量

3）功能说明

使能后，ATAN 指令可计算 Source 值的反正切值，并将结果存储在 Destination 中（以弧度为单位）

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的反正切值。
后扫描	不适用

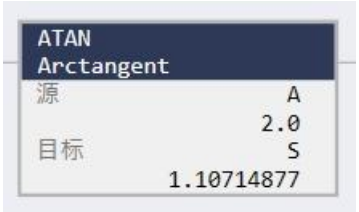
4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 2，TAN 启用时，ATAN 指令对 A 的值求反正切值，并把求到数值“1.10714877”存储到 S 里



ST 示列：

```
S:= ATAN(A);
```

BCD 换为十进制转 BCD_TO 指令

BCD_TO 指令将 BCD 码(源值)转换为十进制码(目标值)

1）指令格式

指令	名称	梯形图表现	ST 表现
BCD_TO	BCD 换为十进制转		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT	变量 立即数	要转换为十进制值的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT LINT	变量	存储计算结果的变量

3）功能说明

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = 带 BCD 值的源的十进制值。
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图示列 ：

A 的值为 2 ，BCD_TO 启用时，BCD_TO 指令将 A 的值转换为十进制值，并把求到数值“2”存储到 S 里

计算余弦值 COS 指令

COS 指令计算源值(弧度)的余弦值。

1）指令格式

指令	名称	梯形图表现	ST 表现
COS	计算余弦值		Dest:=COS(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	计算该值的余弦值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		LINT REAL LREAL		

3）功能说明

COS 指令可计算 Source 值（弧度）的余弦值，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的余弦值。
后扫描	不适用

4）注意事项

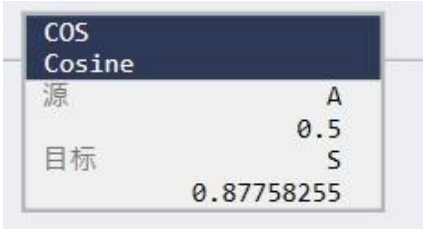
5）错误说明

6）示列

梯形图：

A 的值为 0.5，SIN 启用时，COS 指令对 A 的值求余弦值，并把求到数值

“0.87758255”存储到 S 里



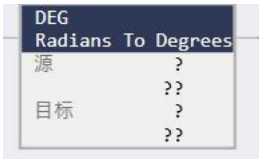
ST 示列：

```
S:= COS(A);
```

弧度转化为角度 DEG 指令

DEG 指令将弧度(源值)转换为角度(目标值)。

1）指令格式

指令	名称	梯形图表现	ST 表现
DEG	弧度转化为角度		Dest:=DEG(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	要转换为角度的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		LINT REAL LREAL		

3）功能说明

启用后，DEG 指令将 Source（弧度）转换为角度，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的角度值。
后扫描	不适用

4）注意事项

DEG 指令使用以下算法：

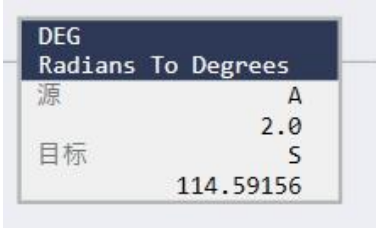
Source*180/pi = Source*57.29578

5）错误说明

6）示列

梯形图示列 ：

A 的值为 2，TAN 启用时,DEG 指令对 A 的值转化为弧度，并把得到数值“114.59156”存储到 S 里



ST 示列：

S:= DEG(A);

角度转化为弧度 RAD 指令

RAD 指令将角度(源值)转换为弧度(目标值)。

1）指令格式

指令	名称	梯形图表现	ST 表现
RAD	角度转化为弧度		Dest:=RAD(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源	SINT INT DINT LINT REAL LREAL	变量 立即数	要转换为弧度的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		LINT REAL LREAL		

3）功能说明

使能后，RAD 指令将 Source（角度）转换为弧度，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的弧度值。
后扫描	不适用

4）注意事项

RAD 指令使用以下算法：

Deg2RadConvFactor=pi/180= 0.017453292

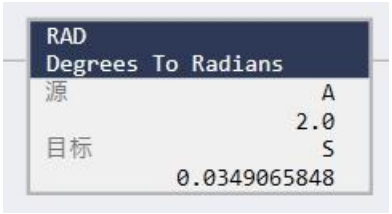
estination = Source * Deg2RadConvFactor

5）错误说明

6）示列

梯形图：

A 的值为 2，TAN 启用时，RAD 指令对 A 的值转化为角度，并把求到数值“0.0349065848”存储到 S 里



ST 示列：

```
S:= RAD(A);
```

计算正弦值 SIN 指令

SIN 指令计算源值(弧度)的正弦值。

1）指令格式

指令	名称	梯形图表现	ST 表现
SIN	计算正弦值		Dest:=SIN(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	计算该值的正弦值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT	变量	存储计算结果的变量

		DINT LINT REAL LREAL		

3）功能说明

使能后，SIN 指令可计算 Source 值（弧度）的正弦值，并将结果存储在 Destination 中。

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的正弦值。
后扫描	不适用

4）注意事项

5）错误说明

6）示列

梯形图：

A 的值为 0.5235988，SIN 启用时，SIN 指令对 A 的值求正弦值，并把求到数值“0.5”存储到 S 里

ST 示列：

S:= SIN(A);

计算正切值 TAN 指令

TAN 指令计算源值(弧度)的正切值。

1）指令格式

指令	名称	梯形图表现	ST 表现
TAN	计算正切值		Dest:=TAN(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	计算该值的正切值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		LINT REAL LREAL		

3）功能说明

TAN 指令可计算 Source 值（弧度）的正切值，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的正切值。
后扫描	不适用

4）注意事项

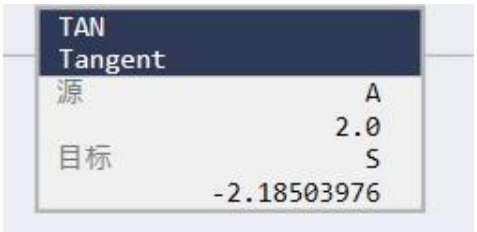
5）错误说明

6）示列

梯形图示列 ：

A 的值为 2，TAN 启用时，TAN 指令对 A 的值求正切值，并把求到数值

“-2.18503976”存储到 S 里



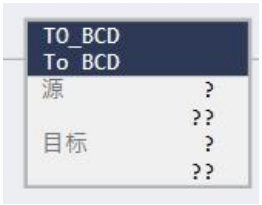
ST 示列：

S:= TAN(A);

十进制转换为 BCD TO_BCD 指令

TO_BCD 指令将十进制码(源值)转换为 BCD 码(目标值)。

1）指令格式

指令	名称	梯形图表现	ST 表现
TO_BCD	十进制转换为 BCD		此指令不可用于结构化文本中

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT	变量 立即数	要转换为 BCD 的值 0 ≤Source Less ≤99,999,999

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		LINT		

3）功能说明

TO_BCD 指令将十进制值（0 ≤ Source L≤99,999,999）转换为 BCD 值，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	不适用。
梯级输入条件为真	控制器将 Source 转换为 BCD 并将结果存储在 Destination 中。
后扫描	不适用。

4）注意事项

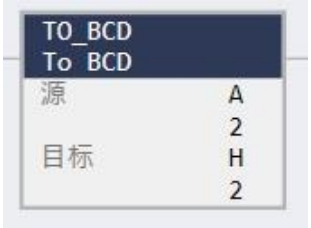
5）错误说明

在以下情况下会发生轻微故障：	故障类型	故障代码
功能已启用并检测到溢出，Source < 0	4	4
功能已启用并检测到溢出，Source > 99,999,999/9,999, 999,999, 999,999	4	4
功能已启用并检测到溢出	4	4

6）示列

梯形图示列 ：

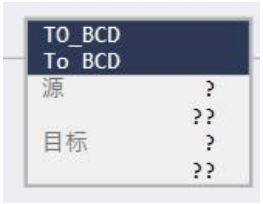
A 的值为 2 ，TO_BCD 启用时，TO_BCD 指令将 A 的值转换为 BCD 值，并把求到数值“2”存储到 S 里



浮点型截断为整数型 TRUNC 指令

TRUNC 指令保留源值的整数部分存储到目标值。

1）指令格式

指令	名称	梯形图表现	ST 表现
TRUNC	浮点型截断为整数型		Dest:=TRUNC(Source);

2）相关变量

输入变量(Input)

参数(英文)	参数(中文)	数据类型	格式	说明
Source	源数据	SINT INT DINT LINT REAL LREAL	变量 立即数	要截断的值

输出变量(Output)

参数(英文)	参数(中文)	数据类型	格式	说明
Dest	目标	SINT INT DINT	变量	存储计算结果的变量

		LINT REAL LREAL		

3）功能说明

TRUNC 启用后，TRN 指令移除（截断）Source 的小数部分，并将结果存储在 Destination 中

条件/状态	执行的操作
预扫描	不适用。
梯级输入条件为假	将梯级输出条件设置为梯级输入条件。
梯级输入条件为真	将梯级输出条件设置为梯级输入条件。 Dest = Source 的截断值。
后扫描	不适用。

4）注意事项

5）错误说明

6）示列

梯形图示列 ：

A 的值为 2 ，TRUNC 启用时，TRUNC 指令将 A 的值截断为整数值，并把求到数值“2”存储到 S 里

附录 A：GSV 和 SSV 控制对象

CLASS	ATTRIBUTE	SSV	GSV	支持的数据类型
AXIS	AccelerationFeedforwardGain	OK	OK	REAL
AXIS	AccelerationLimit	OK	OK	REAL
AXIS	AdaptiveTuningConfiguration	OK	OK	SINT
AXIS	AverageVelocityTimebase	OK	OK	REAL
AXIS	AxisFaultBits	--	OK	DINT
AXIS	AxisStatusBits	--	OK	DINT
AXIS	CIPAxisFaults	--	OK	LINT
AXIS	CIPAxisIoStatus	--	OK	DINT
AXIS	CIPAxisState	--	OK	DINT
AXIS	CIPAxisStatus	--	OK	DINT
AXIS	CommutationOffset	OK	OK	REAL
AXIS	ControlMode	OK	OK	SINT
AXIS	ConversionConstant	OK	OK	REAL
AXIS	CurrentFeedback	--	OK	REAL
AXIS	DecelerationLimit	OK	OK	REAL
AXIS	DynamicsConfigurationBits	OK	OK	DINT
AXIS	Feedback1Polarity	OK	OK	SINT
AXIS	FeedbackMode	OK	OK	SINT
AXIS	FrictionCompensationSliding	OK	OK	REAL
AXIS	HomeConfigurationBits	OK	OK	DINT
AXIS	HomeDirection	OK	OK	SINT
AXIS	HomeMode	OK	OK	SINT
AXIS	HomeOffset	OK	OK	REAL
AXIS	HomePosition	OK	OK	REAL
AXIS	HomeReturnSpeed	OK	OK	REAL

AXIS	HomeSequence	OK	OK	SINT
AXIS	HomeSpeed	OK	OK	REAL
AXIS	HookupTestCommutationOffset	--	OK	REAL
AXIS	InhibitAxis	OK	OK	SINT
AXIS	InterpolatedActualPositon	--	OK	REAL
AXIS	InterpolatedCommandPosition	--	OK	REAL
AXIS	InterpolatedPositionConfiguration	OK	OK	DINT
AXIS	InterpolationTime	OK	OK	DINT
AXIS	LoadObserverBandwidth	OK	OK	REAL
AXIS	LoadObserverConfiguration	OK	OK	SINT
AXIS	LoadObserverFeedbackGain	OK	OK	REAL
AXIS	LoadObserverIntegratorBandwidth	OK	OK	REAL
AXIS	LoadRatio	OK	OK	REAL
AXIS	LoadType	--	OK	SINT
AXIS	MasterInputConfigurationBits	OK	OK	DINT
AXIS	MasterPositionFilterBandwidth	OK	OK	REAL
AXIS	MaximumAcceleration	OK	OK	REAL
AXIS	MaximumAccelerationJerk	OK	OK	REAL
AXIS	MaximumDeceleration	OK	OK	REAL
AXIS	MaximumDecelerationJerk	OK	OK	REAL
AXIS	MaximumSpeed	OK	OK	REAL
AXIS	MechanicalBrakeControl	OK	OK	SINT
AXIS	MechanicalBrakeEngageDelay	OK	OK	REAL
AXIS	MechanicalBrakeReleaseDelay	OK	OK	REAL
AXIS	MotionPolarity	OK	OK	SINT
AXIS	MotionResolution	OK	OK	DINT
AXIS	MotorOverspeedUserLimit	OK	OK	REAL
AXIS	MotorPolarity	OK	OK	SINT
AXIS	OutputCamExecutionTargets	--	OK	DINT

AXIS	PositionErrorTolerance	OK	OK	REAL
AXIS	PositionIntegratorBandwidth	OK	OK	REAL
AXIS	PositionLoopBandwidth	OK	OK	REAL
AXIS	PositionScalingDenominator	--	OK	REAL
AXIS	PositionScalingNumerator	--	OK	REAL
AXIS	PositionTrim	OK	OK	REAL
AXIS	PositionUnwind	OK	OK	DINT
AXIS	RotaryMotorMaxSpeed	--	OK	REAL
AXIS	RotaryMotorRatedSpeed	--	OK	REAL
AXIS	SLATConfiguration	OK	OK	SINT
AXIS	SLATSetPoint	OK	OK	REAL
AXIS	SLATTimeDelay	OK	OK	REAL
AXIS	SoftTravelLimitChecking	OK	OK	SINT
AXIS	SoftTravelLimitNegative	OK	OK	REAL
AXIS	SoftTravelLimitPositive	OK	OK	REAL
AXIS	StoppingTimeLimit	OK	OK	REAL
AXIS	TorqueLeadLagFilterBandwidth	OK	OK	REAL
AXIS	TorqueLeadLagFilterGain	OK	OK	REAL
AXIS	TorqueLimitNegative	OK	OK	REAL
AXIS	TorqueLimitPositive	OK	OK	REAL
AXIS	TorqueLowPassFilterBandwidth	OK	OK	REAL
AXIS	TorqueNotchFilterFrequency	OK	OK	REAL
AXIS	TorqueNotchFilterHighFrequencyLimit	OK	OK	REAL
AXIS	TorqueNotchFilterLowFrequencyLimit	OK	OK	REAL
AXIS	TorqueNotchFilterTuningThreshold	OK	OK	REAL
AXIS	TorqueOffset	OK	OK	REAL
AXIS	TorqueTrim	OK	OK	REAL
AXIS	TransmissionRatioInput	--	OK	DINT
AXIS	TransmissionRatioOutput	--	OK	DINT

AXIS	TravelMode	OK	OK	SINT
AXIS	VelocityErrorTolerance	OK	OK	REAL
AXIS	VelocityFeedforwardGain	OK	OK	REAL
AXIS	VelocityIntegratorBandwidth	OK	OK	REAL
AXIS	VelocityLimitNegative	OK	OK	REAL
AXIS	VelocityLimitPositive	OK	OK	REAL
AXIS	VelocityLoopBandwidth	OK	OK	REAL
AXIS	VelocityTrim	OK	OK	REAL
WallClockTime	LocalDateTime	OK	OK	DINT/DINT 数组
WallClockTime	DateTime	OK	OK	DINT/DINT 数组
WallClockTime	TimeZoneString	OK	OK	INT
Program	DisableFlag	OK	OK	DINT
Controller	Name	--	OK	String
ControllerDevice	DeviceName	--	OK	SINT 数组
ControllerDevice	SerialNumber	--	OK	DINT
ControllerDevice	Type	--	OK	INT
ControllerDevice	Status	--	OK	INT
ControllerDevice	ProductCode	--	OK	INT
ControllerDevice	ProductRev	--	OK	INT
ControllerDevice	Vendor	--	OK	INT
ControllerDevice	IPAddress	--	OK	String
Module	EntryStatus	--	OK	INT

专业·创新·责任·服务

英孚康（浙江）工业技术有限公司

公司地址：嘉兴智创园三期24号楼
联系方式：400-066-8300
公司网址：www.i-con.cn