

# M系列 可编程逻辑控制器

基础逻辑

## 指令手册



## 目录

目录	I
前言	V
第一章 数据类型	1
第二章 指令说明	3
2.1 时序输入输出指令	4
2.1.1 R_TRIG (上升沿检测)	4
2.1.2 F_TRIG (下降沿检测)	6
2.1.3 RS (复位优先)	7
2.1.4 SR (置位优先)	9
2.1.5 SEMA (置位延时输出)	11
2.1.6 JMP (跳转)	13
2.2 数据搬移指令	14
2.2.1 MOVE (搬移指令)	14
2.2.2 MoveBit (位搬移指令)	16
2.2.3 TransBit (多位搬移指令)	18
2.2.4 MoveDigit (数位搬移指令)	20
2.2.5 Exchange (数据交换指令)	22
2.2.6 Swap (高低字节数据交换指令)	23
2.3 比较运算	24
2.3.1 EQ (等于)	24
2.3.2 NE (不等于)	26
2.3.3 GT/GE/LT/LE (大于/大于等于/小于/小于等于)	27
2.4 定时器	30
2.4.1 TON (通电延时定时器)	30
2.4.2 TOF (断电延时定时器)	33
2.4.3 TP (脉冲延时定时器)	36
2.4.4 通过触摸屏修改定时器定时时间示例程序	38
2.5 计数器	39
2.5.1 CTU (加计数器)	39
2.5.2 CTD (减计数器)	41
2.5.3 CTUD (加减计数器)	43
2.6 数学函数	46
2.6.1 ADD (加法)	46
2.6.2 SUB (减法)	48
2.6.3 MUL (乘法)	50
2.6.4 DIV (除法)	52
2.6.5 ABS (计算绝对值)	54
2.6.6 MOD (整数取余)	55
2.6.7 MODREAL (浮点数取余)	57
2.6.8 MODTURNS (计算圈数)	58
2.6.9 MODABS (计算相位)	60

2.6.10	RadToDeg (弧度转角度)	62
2.6.11	DegToRad (角度转弧度)	63
2.6.12	SIN/COS/TAN (三角函数)	64
2.6.13	ASIN/ACOS/ATAN (反三角函数)	67
2.6.14	LN (自然对数)	70
2.6.15	LOG (自然对数)	71
2.6.16	SQRT (求平方根)	72
2.6.17	EXP (自然指数)	73
2.6.18	EXPT (幂指数)	74
2.6.19	TRUNC/FLOOR (浮点数取整数部分)	75
2.6.20	FRACTION (浮点数取小数部分)	77
2.6.21	RAND (随机数)	78
2.7	逻辑运算指令	79
2.7.1	AND (与)	79
2.7.2	OR (或)	82
2.7.3	NOT (取反)	84
2.7.4	XOR (异或)	86
2.7.5	XORN (同或)	88
2.8	数据移位	90
2.8.1	SHL (向左移位) / SHR (向右移位)	90
2.8.2	ROL (向左循环移位) / ROR (向右循环移位)	93
2.9	选择运算	96
2.9.1	MAX/MIN (最大值/最小值)	96
2.9.2	SEL (位选择)	98
2.9.3	MUX (多路复用器)	100
2.9.4	LIMIT (上下限限制)	102
2.9.5	BAND (死区限制)	104
2.9.6	ZONE (输入偏移)	106
2.10	数据类型转换	108
2.10.1	BOOL_TO_*** (BOOL 转换为其它数据类型)	108
2.10.2	***_TO_*** (位串转换为其它数据类型)	110
2.10.3	***_TO_*** (整数转换为其它数据类型)	113
2.10.4	REAL/LREAL_TO_*** (实数转换为其它数据类型)	117
2.10.5	TIME/DATE_TO_*** (时间、日期转换为其它数据类型)	120
2.10.6	STRING_TO_*** (STRING 转换为其它数据类型)	121
2.11	字符串指令	123
2.11.1	CONCAT (连接字符串)	123
2.11.2	DELETE (删减字符串)	124
2.11.3	INSERT (字符串插入)	125
2.11.4	LEFT/RIGHT (左/右截取字符串)	127
2.11.5	MID (字符串截取)	129
2.11.6	REPLACE (字符替换)	130
2.11.7	LEN (计算字符串长度)	131

2.11.8	FIND (查找字符串) .....	132
2.12	IO 刷新和 PID.....	133
2.12.1	UpdataInput (输入点立即刷新) .....	133
2.12.2	UpdataOutput (输出点立即刷新) .....	134
2.12.3	PWM_S (可变占空比脉冲输出) .....	135
2.12.4	PID (自整定 PID) .....	137
2.13	校验函数.....	141
2.13.1	CRC16(CRC16 校验函数).....	141
2.13.2	LRC(LRC 校验函数).....	143
2.14	位字转换.....	144
2.14.1	GetBitofWord(读取变量中指定位的值) .....	144
2.14.2	SetBitofWord(设定变量中的指定位的值).....	146
2.15	扩展模块通讯 .....	148
2.15.1	EXT_ReadParameter(读取扩展模块参数).....	148
2.15.2	EXT_WriteParameter (设定扩展模块参数) .....	150
2.16	系统功能.....	152
2.16.1	SYS_GetTotalWorkTime (累计通电时间) .....	152
2.16.2	SYS_GetWorkTime (单次通电时间) .....	153
2.16.3	SYS_GetRTCTime (读取实时时钟) .....	154

## 前言

非常感谢您购买M系列控制器。该手册主要介绍控制器逻辑指令，如数学运算指令、定时器、计数器、数据转换指令等。

### 阅读对象

本手册阅读对象：M系列控制器编程和调试的技术人员，读者需要具备一定的可编程控制器相关的基础知识和编程思维。

### 手册修订说明

版本	变更时间	修订内容
V1.00	2023/8/23	初版

### 其他说明

- 本手册内容基于产品信息和客户需求编辑，用户对手册内容有疑问或错误之处，欢迎致电禾川或发送邮件至 [400@hcfa.cn](mailto:400@hcfa.cn)，并按照封面标注版本号协助说明。
- 本手册内容，包括文字、图片、标识、表格等，未经授权时，不得以任何形式复制和传递本手册中的内容，否则，我司将依法追究违规者的法律责任。





# 第一章 数据类型



M系列控制器支持的数据类型及对应的数据范围如下表所示：

分类	数据类型	数据宽度	数据范围	初始值
布尔	BOOL	1个位	TRUE或者FALSE	FALSE
位串	BYTE	1字节	16#00 ~ 16#FF	0
	WORD	2字节	16#0000 ~ 16#FFFF	0
	DWORD	4字节	16#00000000 ~ 16#FFFFFFFF	0
	LWORD	8字节	16#0000000000000000 ~ 16#FFFFFFFFFFFFFFFF	0
整数	USINT	1字节	0 ~ 255	0
	UINT	2字节	0 ~ 65535	0
	UDINT	4字节	0 ~ 4294967295	0
	ULINT	8字节	0 ~ 18446744073709551615	0
	SINT	1字节	-128 ~ +127	0
	INT	2字节	-32768 ~ +32767	0
	DINT	4字节	-2147483648 ~ +2147483647	0
浮点数	LINT	8字节	-9223372036854775808 ~ +9223372036854775807	0
	REAL	4字节	-3.402823e+38 ~ -1.175495e-38, 0, 1.175495e-38 ~ 3.402823e+38	0.0
时间、日期	LREAL	8字节	-1.79769313486231e+308 ~ -2.22507385850721e-308, 0, 2.22507385850721e-308 ~ 1.79769313486231e+308,	0.0
	TIME	8字节	显示格式：T#XXXXXdXXhXXmXXsXXX.XXXms。 范围：T#0ns~213503d23h34m33s709.551ms。 单位：纳秒 (ns)。	T#0ms
	DATE	4字节	显示格式：D#年-月-日。 范围：D#1970-01-01~D#2106-02-07。 单位：秒 (s)。	D#1970-01-01
	TOD	4字节	显示格式：TOD#时：分：秒.毫秒。 范围：TOD#00:00:00~23:59:59.999。 单位：毫秒 (ms)。 数值为0时对应的值为TOD#00:00:00， 数值为1000时对应的值为TOD#00:00:01.000。	TOD#00:00:00
字符串	DT	4字节	显示格式：DT#年-月-日-时-分-秒。 范围：DT#1970-01-01-0:0:0~2106-02-07-6:28:15。 单位：秒 (s)。	DT#1970-01-01-0:0:0
	STRING	0~80个字节	0~80个字符	''



## 第二章 指令说明



第二章 指令说明 .....	3
2.1 时序输入输出指令 .....	4
2.2 数据搬移指令 .....	14
2.3 比较运算 .....	24
2.4 定时器 .....	30
2.5 计数器 .....	38
2.6 数学函数 .....	46
2.7 逻辑运算指令 .....	79
2.8 数据移位 .....	90
2.9 选择运算 .....	96
2.10 数据类型转换 .....	108
2.11 字符串指令 .....	123
2.12 IO 刷新和 PID .....	133
2.13 校验函数 .....	141
2.14 位字转换 .....	144
2.15 扩展模块通讯 .....	148
2.16 系统功能 .....	152



## 2.1 时序输入输出指令

### 2.1.1 R\_TRIG (上升沿检测)

本指令检测到输入的上升沿时，输出“Out”为TRUE1个任务周期。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
R_TRIG	上升沿触发	FB		<pre>R_TRIG_instance (   CLK= 《参数》,   Q=&gt; 《参数》, );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
CLK	输入检测	输入	输入信号	FALSE或者TRUE
Q	输出状态	输出	输出信号	FALSE或者TRUE

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 该指令检测到输入变量“CLK”的上升沿从 FALSE 变为 TRUE 时，仅 1 个任务周期输出变量“Q”为 TRUE，即输出变量“Q”的值为 TRUE1 个任务周期后变为 FALSE，其余情况下，输出变量“Q”的值为 FALSE。
- 示例程序如下所示：

梯形图(LD):

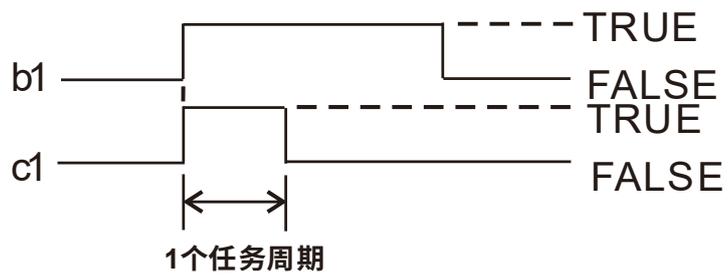


结构化文本(ST):

```
R_TRIG1(CLK:=b1,Q=>c1);
```

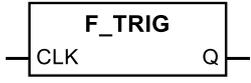
- 时序图如下所示：

时序图：



### 2.1.2 F\_TRIG (下降沿检测)

本指令检测到输入的下降沿时，输出“Out”为TRUE 1个任务周期。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
F_TRIG	下降沿检测	FB		<pre>F_TRIG_instance ( CLK:=《参数》, Q=&gt;《参数》, );</pre>

■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
CLK	输入检测	输入	输入信号	FALSE或者TRUE
Q	输出状态	输出	输出信号	FALSE或者TRUE

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

■ 功能说明

- 该指令检测到输入变量“CLK”的下降沿（从 TRUE 变为 FALSE）时，即输出变量“Q”的值为 TRUE，1 个任务周期后变为 FALSE，其余情况下，输出变量“Q”的值为 FALSE。
- 示例程序如下所示：

梯形图(LD):

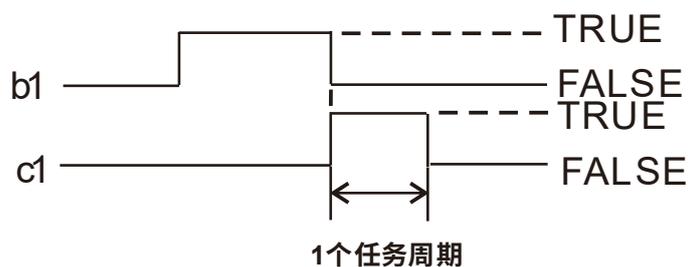


结构化文本(ST):

```
F_TRIG1(CLK:=b1,Q=>c1);
```

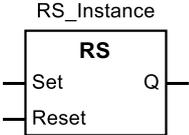
- 时序图如下所示：

时序图:



### 2.1.3 RS (复位优先)

复位输入信号为TRUE时，复位输入信号优先，输出信号为FALSE。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
RS	复位优先	FB		<pre>RS_instance ( SET := 《参数》, Reset=&gt;《参数》, Q =&gt;《参数》, );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Set	置位输入	输入	置位输入	FALSE或者TRUE
Reset	复位输入		复位输入	FALSE或者TRUE
Q	输出	输出	输出	FALSE或者TRUE

	布尔	位串				整数							实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Set	<input type="radio"/>																			
Reset	<input type="radio"/>																			
Q	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于置位或复位操作，其中复位操作优先于置位操作，即复位操作执行时，置位操作无效。
- 参数“Set”为TRUE时执行置位操作（将“Q”设置为TRUE），参数“Reset”为TRUE时执行复位操作（将“Q”设置为FALSE）。
- 当置位和复位均为TRUE时，复位操作优先，输出“Q”为FALSE；当置位和复位均为FALSE时，输出“Q”的值保持不变。
- 输入和输出逻辑关系如下表所示：

“Set” 的值	“Reset” 的值	“Q” 的值
FALSE	TRUE	FALSE
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	FALSE	保持不变

- 示例程序和时序图如下所示：

梯形图(LD):

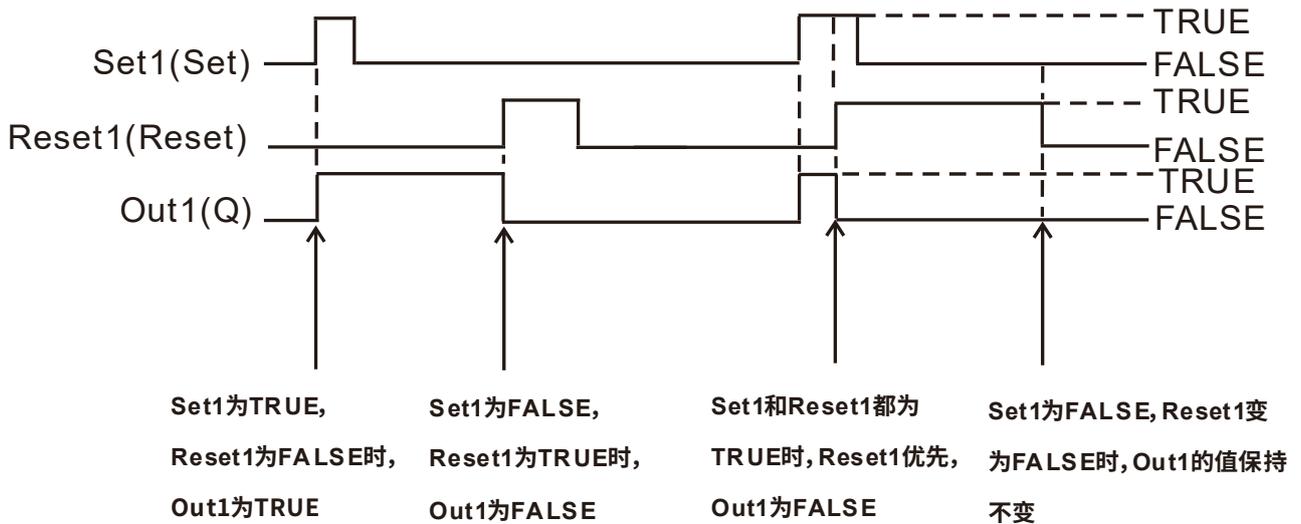
名称	类别	名称	分配到	数据类型	在线值	准备值
44	VAR	RS0		RS		
46	VAR	set1		BOOL	TRUE	
47	VAR	reset1		BOOL	TRUE	
48	VAR	out1		BOOL	FALSE	

结构化文本ST(LD):

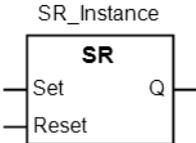
RS0(EN:= TRUE,Set:=Set1 ,Reset:=Reset1 ,ENO=> ,Q=>out1 );

时序图:



## 2.1.4 SR (置位优先)

置位输入信号为TRUE时，置位输入信号优先，输出信号为TRUE。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SR	置位优先	FB		<pre>SR_instance ( SET := 《参数》, Reset=&gt;《参数》, Q =&gt;《参数》, );</pre>

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Set	置位输入	输入	置位输入	FALSE或者TRUE
Reset	复位输入		复位输入	FALSE或者TRUE
Q	输出	输出	输出	FALSE或者TRUE

	布尔	位串				整数							实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Set	<input type="radio"/>																			
Reset	<input type="radio"/>																			
Q	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于置位或复位操作，其中置位操作优先于复位操作，即置位操作执行时，复位操作无效。
- 参数“Set”为TRUE时执行置位操作（将“Q”设置为TRUE），参数“Reset”为TRUE时执行复位操作（将“Q”设置为FALSE）。
- 当置位和复位均为TRUE时，置位操作优先，输出“Q”为TRUE；当置位和复位均为FALSE时，输出“Q”的值保持不变。
- 输入和输出逻辑关系如下表所示：

“Set”的值	“Reset”的值	“Q”的值
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	保持不变

- 示例程序和时序图如下所示：

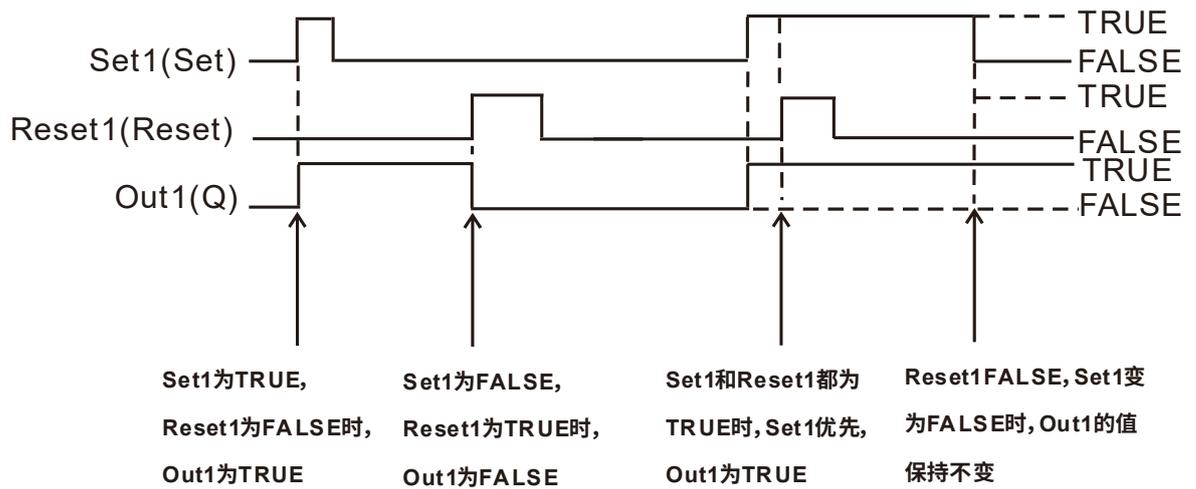
梯形图(LD)：



结构化文本ST(LD)：

```
SR0(EN:= TRUE,Set:=Set1,Reset:=Reset1,ENO=>,Q=>out1);
```

时序图：



### 2.1.5 SEMA (置位延时输出)

置位输入为TRUE时，置位输入优先，输出延迟1个周期为TRUE。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SEMA	置位延时输出	FB		<pre>SEMA_instance ( CLAIM := 《参数》, RELEASE=&gt; 《参数》, Q =&gt; 《参数》, );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Claim	置位输入	输入	置位输入	FALSE或者TRUE
Release	复位输入		复位输入	FALSE或者TRUE
Q	输出	输出	输出信号	FALSE或者TRUE

	布尔	位串					整数							实数		时间, 日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Claim	<input type="radio"/>																			
Release	<input type="radio"/>																			
Q	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

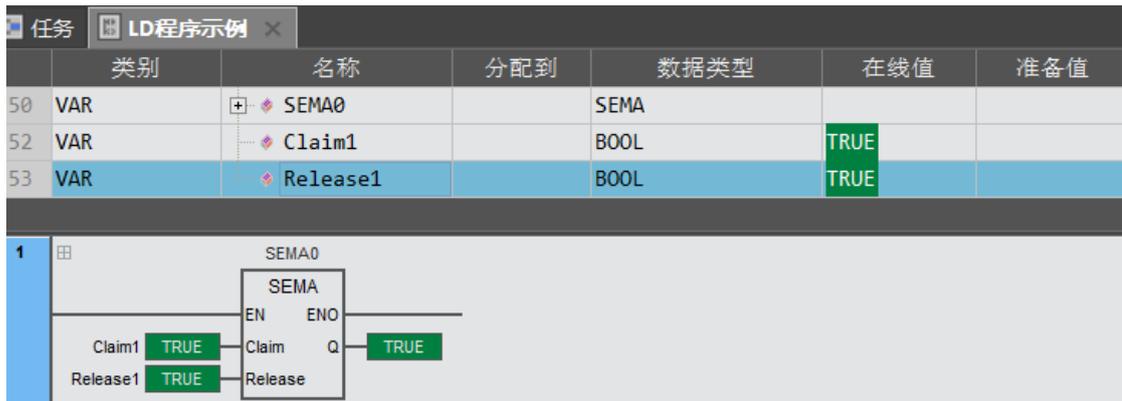
#### ■ 功能说明

- 本指令用于置位或复位操作，其中置位操作优先于复位操作，即置位操作执行时，复位操作无效。
- 参数“Claim”为TRUE时执行置位操作（将“Q”设置为TRUE），参数“Release”为TRUE时执行复位操作（将“Q”设置为FALSE）
- 当置位和复位均为TRUE时，置位操作优先，输出“Q”延时1个周期为TRUE；当置位和复位均为FALSE时，输出“Q”的值保持不变。
- 输入和输出逻辑关系如下表所示：

“Claim”的值	“Release”的值	“Q”的值
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	保持不变

- 示例程序和时序图如下所示：

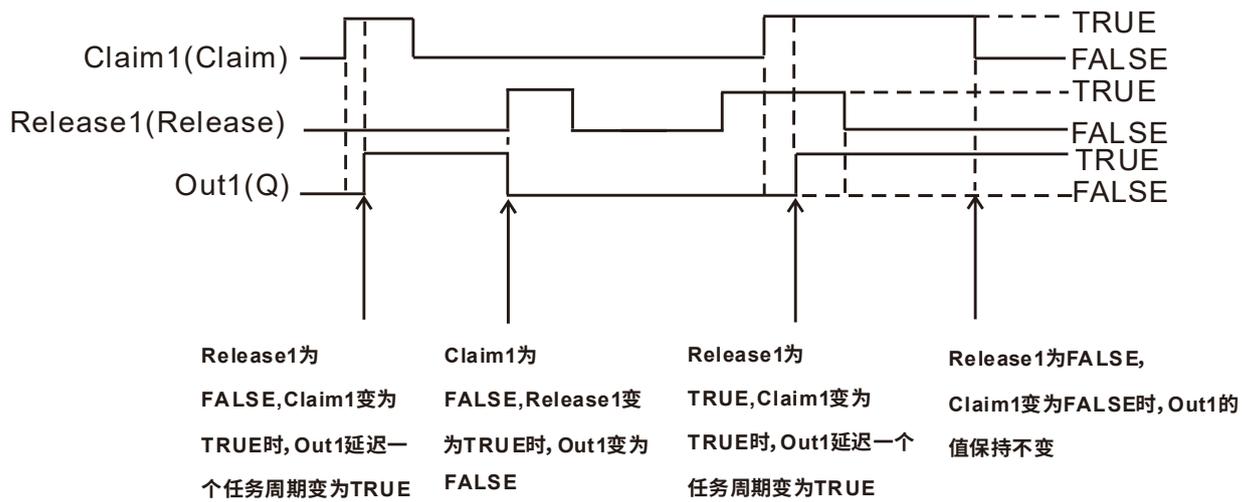
**梯形图(LD)：**



**结构化文本ST(LD)：**

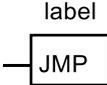
SEMA0(EN:= TRUE,Claim:=Claim1,Release:=Release1,ENO=>,Q=>out1);

**时序图：**



## 2.1.6 JMP (跳转)

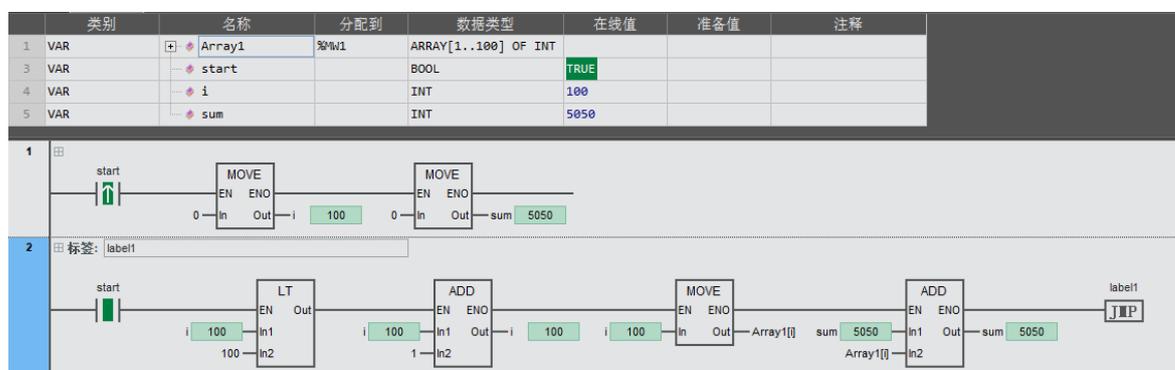
跳转到指定标签位置。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
JMP	跳转	FUN		GOTO <标签名>;

### ■ 功能说明

- 当输入条件成立时，“JMP”指令生效，跳转到指定的标签执行。当输入条件不成立时，执行 JMP 后面的指令。
- 示例程序功能说明：网络 1 通过 start 变量的上升沿给 i 和 sum 变量赋值为 0，网络 2 通过 JMP 指令和 LT 指令配合，实现 1 个周期内循环 100 次，将 1~100 对应放入%MW1~%MW100 装置内，并求 1~100 累加和。
- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

label1:

```

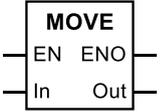
IF i<100 THEN
    i:=i+1;
    Array1[i]:=i;
    sum:=sum + Array1[i];
    GOTO label1;
END_IF;

```

## 2.2 数据搬移指令

### 2.2.1 MOVE (搬移指令)

本指令用于将单个变量或常量的值搬移至其他变量内。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MOVE	数据搬移	FUN		Out:=In;

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	数据来源	输入	数据来源	变量的数据类型决定
Out	搬移目标	输出	数据输出	变量的数据类型决定

	布尔	位串					整数							实数		时间, 日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	<input type="radio"/>																			
Out	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将数据来源“In”中变量或常量的值搬移至“Out”。
- 数据来源“In”可以为枚举量、数组变量名、数组成员、结构体、结构体中的1个成员。
- “In”和“Out”的数据类型须一致。“In”和“Out”为数组变量名时，可以将“In”数组所有成员的数据对应搬移到“Out”数组的所有成员内，数组的数据类型和成员个数需要相同。
- 示例程序1：将变量a的值搬移至变量b内。

#### 梯形图(LD):

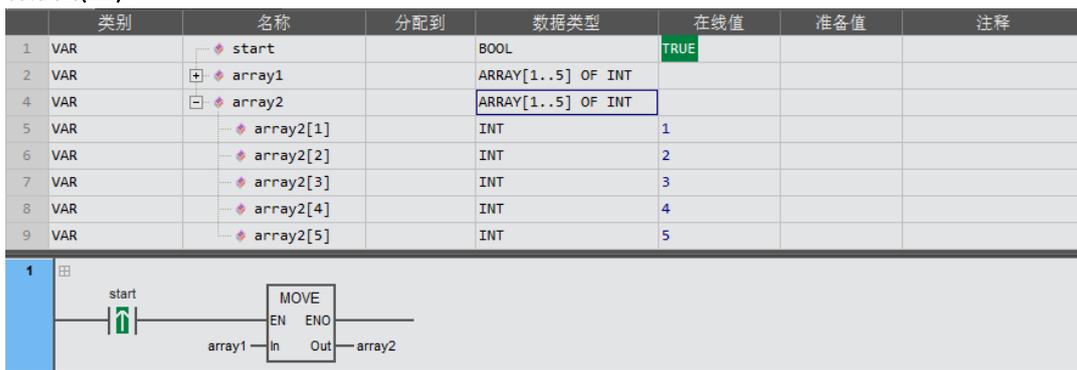


#### 结构化文本(ST):

b:=a;

- 示例程序 2：将数组变量 array1 的所有成员的值对应搬移至数组变量 array2 的所有成员。

#### 梯形图(LD):

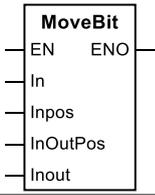


#### 结构化文本(ST):

```
array2:=array1;
```

## 2.2.2 MoveBit (位搬移指令)

本指令用于将1个变量中某1位的值搬移至另1个变量中某1位。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MoveBit	位搬移	FUN		<pre>MoveBit (   IN:= 《参数》,   InPos:= 《参数》,   InOutPos:= 《参数》,   InOut:= 《参数》, );</pre>

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	数据来源	输入	数据来源	变量的数据类型决定
InPos	数据来源中指定位置	输入	数据来源中指定位置	0 ~ “In” 包含的位数-1
InOutPos	搬移目标指定位置	输入	搬移目标指定位置	0 ~ “Out” 包含的位数-1
InOut	搬移目标	输入	搬移目标	变量的数据类型决定

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

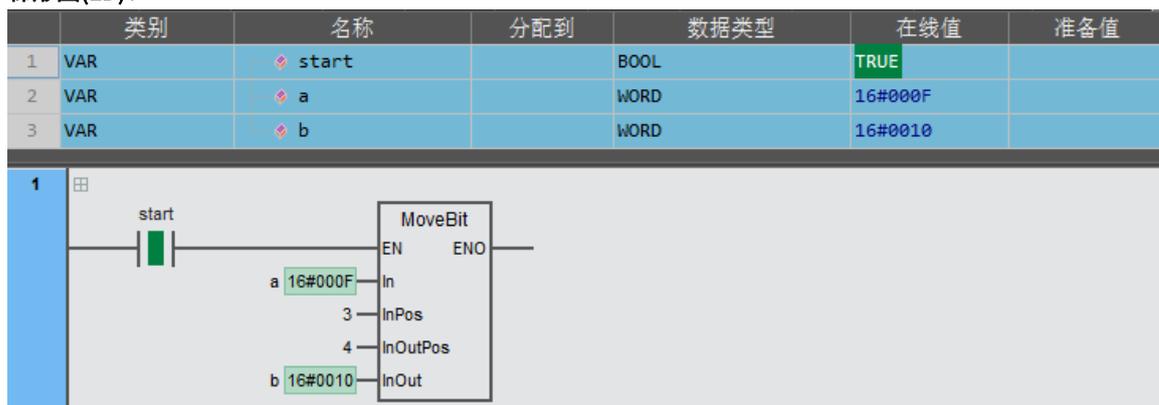
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于将输入变量中指定位置某1位的值搬移至 InOut 变量中指定位置某一位，InOut 变量中其它位的值保持不变。
- “InPos”表示数据来源中（“In”）指定位置， “InPos”的值从“In”的位0（bit0）开始计算。如 “InPos”的值为0时表示指定“In”的位0（bit0）， “InPos”的值为1时表示指定“In”的位1（bit1），以此类推。
- “InOutPos”表示搬移目标中（“InOut”）指定位置， InOutPos 的值从“InOut”的位0（bit0）开始计算。如 “InOutPos”的值为0时表示指定“InOut”的位0（bit0）， “InOutPos”的值为1时表示指定“InOut”的位1（bit1），以此类推。
- 下图所示为将两个 WORD 类型的变量进行位搬移的示例，变量 a 的值为 16#0F，“InPos”的值为3，“InOutPos”的值为4，该指令执行后，变量 b 的值为 16#10。

- 示例程序和示意图如下所示:

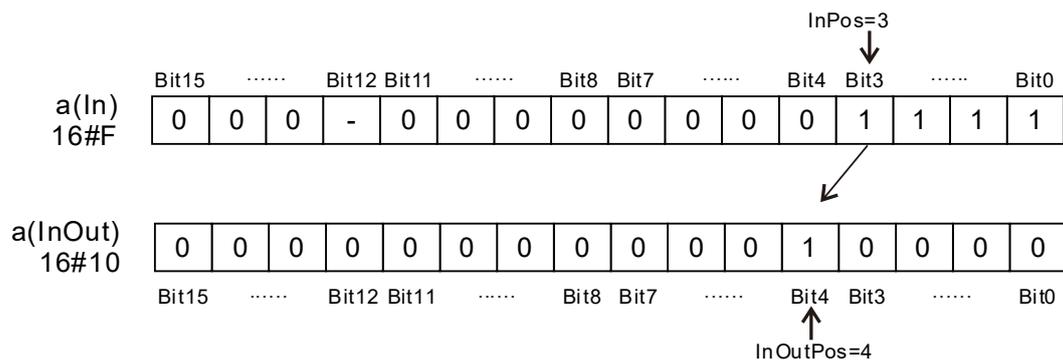
#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
    MoveBit(In:=a ,InPos:=3 ,InOutPos:=4 ,InOut:=b );
END_IF;
```

#### 执行结果示意图:



#### △注意

- InPos 指定的位置超过 In 的位置范围时, 不进行位搬移, InOut 值不变。
- InOutPos 指定的位置超过 InOut 的位置范围时, 不进行位搬移, InOut 值不变。

### 2.2.3 TransBit (多位搬移指令)

本指令用于将输入变量中指定位置多个位的值搬移至另外1个变量中指定位置多个位。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
TransBit	多位搬移指令	FUN		<pre>TransBit (   IN:= 《参数》,   InPos:= 《参数》,   InOutPos := 《参数》,   Size := 《参数》,   InOut := 《参数》, );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	数据来源	输入	数据来源	变量的数据类型决定
InPos	数据来源中指定位置	输入	数据来源中指定位置	0 ~ “In” 包含的位数-1
InOutPos	搬移目标指定位置	输入	搬移目标指定位置	0 ~ “Out” 包含的位数-1
Size	位的数量	输入	搬移位的数量	变量的数据类型决定
InOut	搬移目标	输入	搬移目标	变量的数据类型决定

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

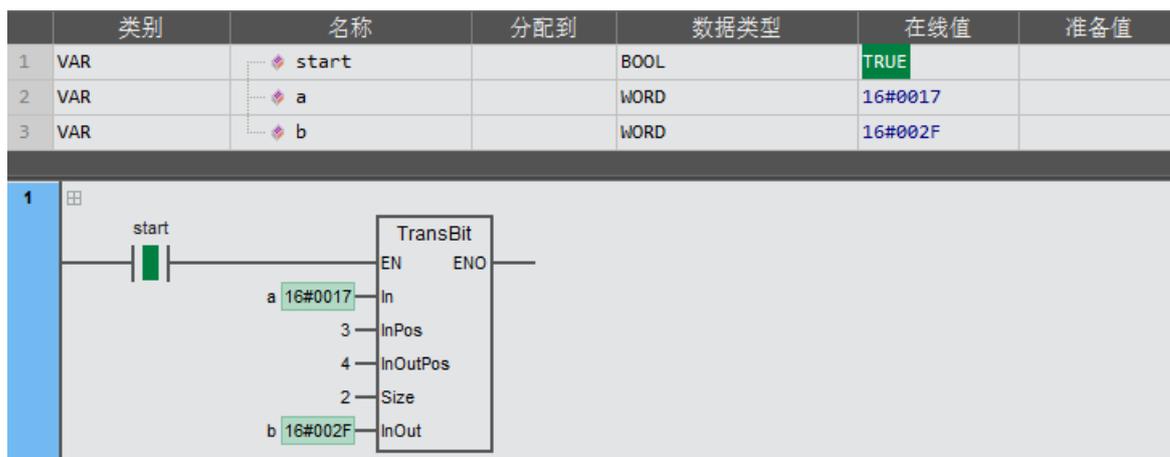
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将输入变量中指定位置多个位的值搬移至 InOut 变量中指定位置多个位，InOut 变量中其它位的值保持不变。
- “InPos”表示数据来源中（“In”）指定位置， “InPos”的值从“In”的位0（bit0）开始计算。如“InPos”的值为0时表示指定“In”的位0（bit0）， “InPos”的值为1时表示指定“In”的位1（bit1），以此类推。
- “InOutPos”表示搬移目标中（“InOut”）指定位置， InOutPos 的值从“InOut”的位0（bit0）开始计算。如“InOutPos”的值为0时表示指定“InOut”的位0（bit0）， “InOutPos”的值为1时表示指定“InOut”的位1（bit1），以此类推。
- Size 表示搬移位的数量。
- 下图所示为将两个 WORD 类型的变量进行多位搬移的示例，“IN”的值为 16#17，“InPos”的值为 3，“InOutPos”的值为 4，“Size”的值为 2，该指令执行后，b 的值为 16#2F。

- 示例程序和示意图如下所示:

#### 梯形图(LD):



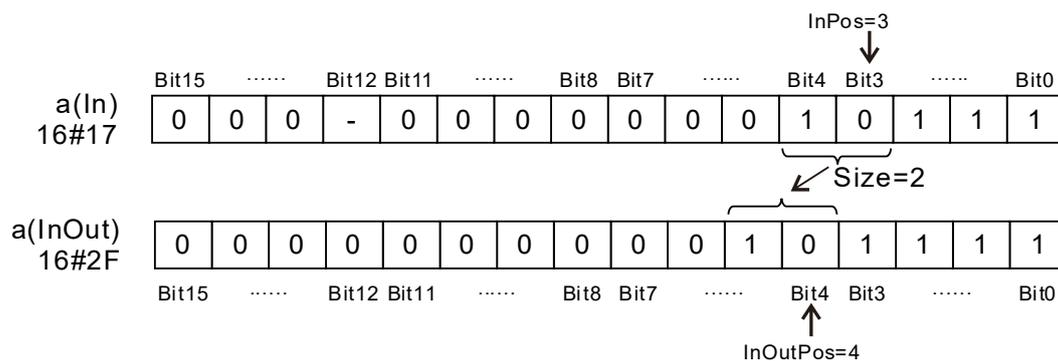
#### 结构化文本(ST):

```
IF start THEN
```

```
    TransBit(In:=a, InPos:=3, InOutPos:=4, Size:=2, InOut:=b);
```

```
END_IF;
```

#### 执行结果示意图:

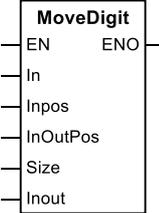


#### △注意

- InPos 指定的位置超过 In 的位置范围时, 不进行位搬移, InOut 值不变。
- InOutPos 指定的位置超过 InOut 的位置范围时, 不进行位搬移, InOut 值不变。
- Size 的值超过 In 或 InOut 变量的范围时, 不进行位搬移, InOut 值不变。

## 2.2.4 MoveDigit (数位搬移指令)

本指令用于将1个变量中指定位置和数量的数位值搬移至另1个变量中指定位置开始的数位中。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MoveDigit	位搬移	FUN		<pre>MoveDigit ( In:= 《参数》, InPos:= 《参数》, InOutPos := 《参数》, Size := 《参数》, InOut := 《参数》, );</pre>

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	数据来源	输入	数据来源	变量的数据类型决定
InPos	数据来源中指定位置	输入	数据来源中指定位置	0 ~ “In” 包含的位数-1
InOutPos	搬移目标指定位置	输入	搬移目标指定位置	0 ~ “Out” 包含的位数-1
Size	数位的数量	输入	数位的数量 (4个位组成1个数位)	变量的数据类型决定
InOut	搬移目标	输入	搬移目标	变量的数据类型决定

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

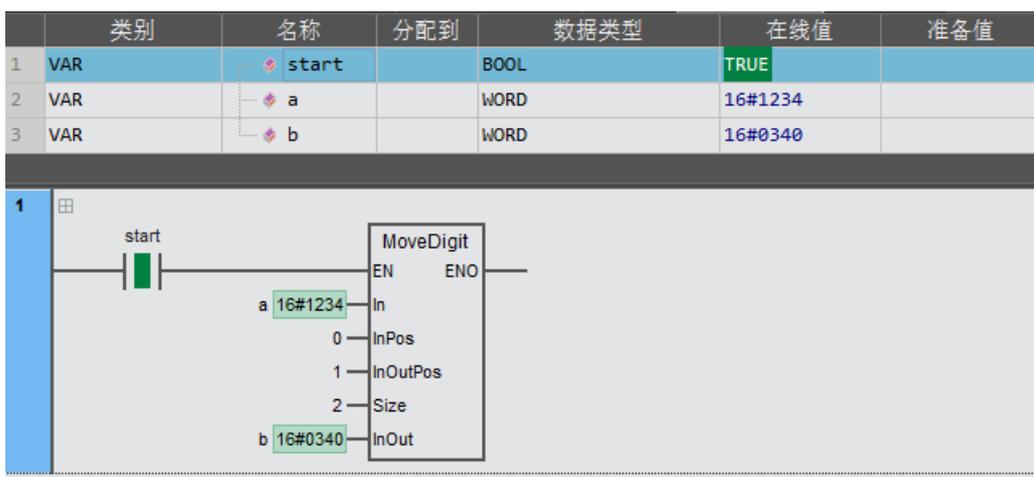
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 将输入变量中指定位置及数量的数位值搬移至 InOut 变量指定位置开始的数位中，In 变量和 InOut 变量中 4 个位组成 1 个数位。
- InPos 表示输入变量数位的位置。输入变量的位 0~位 3 组成数位 0，对应 InPos 的值为 0；输入变量的位 4~位 7 组成数位 1，对应 InPos 的值为 1，以此类推。InOutPos 表示 InOut 变量数位的位置。InOut 变量的位 0~位 3 组成数位 0，对应 InOutPos 的值为 0；InOut 变量的位 4~位 7 组成数位 1，对应 InOut 的值为 1，以此类推。
- Size 表示数位的数量。
- 下图所示为将两个 WORD 类型的变量进行数位搬移的示例。“IN”的值为 16#1234 “InPos”的值为 0，“InOutPos”的值为 1，“Size”的值为 2，b 的初始值为 0，该指令执行后，b 的值为 16#340。

- 示例程序和示意图如下所示:

#### 梯形图(LD):



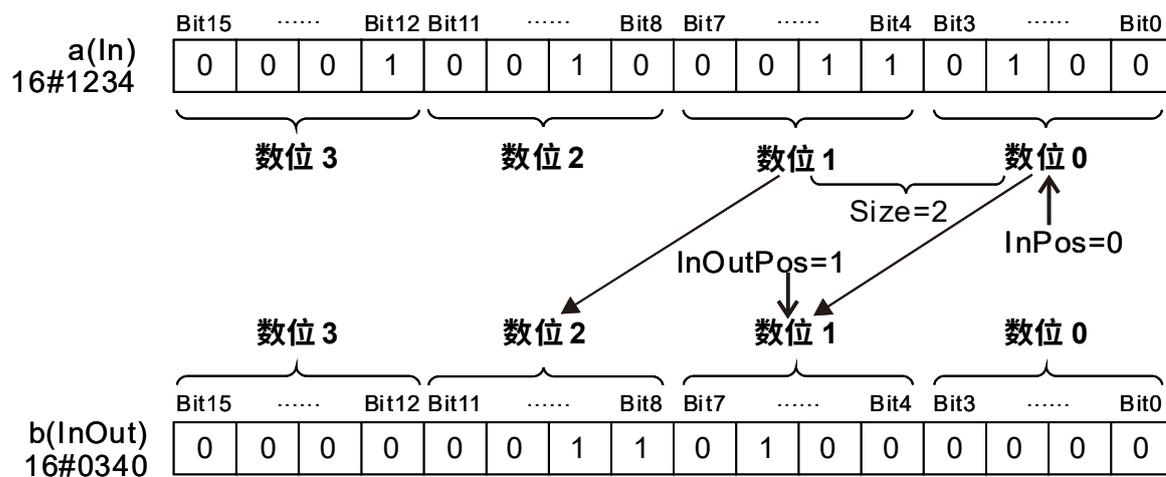
#### 结构化文本(ST):

```
IF start THEN
```

```
    MoveDigit(In:=a ,InPos:=0 ,InOutPos:=1 ,Size:=2 ,InOut:=b);
```

```
END_IF;
```

#### 执行结果示意图:



#### △注意

- InPos 指定的位置超过 In 的位置范围时, 不进行数位搬移, InOut 值不变。
- InOutPos 指定的位置超过 InOut 的位置范围时, 不进行数位搬移, InOut 值不变。
- Size 的值超过 In 或 InOut 变量的范围时, 不进行数位搬移, InOut 值不变。

## 2.2.5 Exchange (数据交换指令)

本指令用于将两个变量的值进行交换。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
Exchange	数据交换	FUN		Exchange(In1:=,In2:=);

### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	输入1	输入	输入1	变量的数据类型决定
In1	输入2	输入	输入2	变量的数据类型决定

	布尔	位串					整数						实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	<input type="radio"/>																			
In2	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

- 本指令用于将两个变量中的值进行交换。
- 两个变量的数据类型须一致。
- 该指令执行时，建议该指令的执行条件为上升沿。该指令的执行条件一直为 TRUE，则两个变量的值会一直进行交换。
- 示例程序如下所示：

#### 梯形图(LD)：

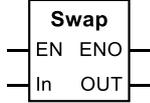


#### 结构化文本(ST)：

```
IF EDGEPOS(start) THEN
    Exchange(In1:=a,In2:=b);
END_IF;
```

## 2.2.6 Swap（高低字节数据交换指令）

本指令用于输入变量中的值进行高低字节交换后放入输出变量中。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
Swap	高低字节数据交换指令	FUN		Out:=Swap(In);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	数据来源	输入	数据来源	变量的数据类型决定
Out	交换结果	输出	交换结果	变量的数据类型决定

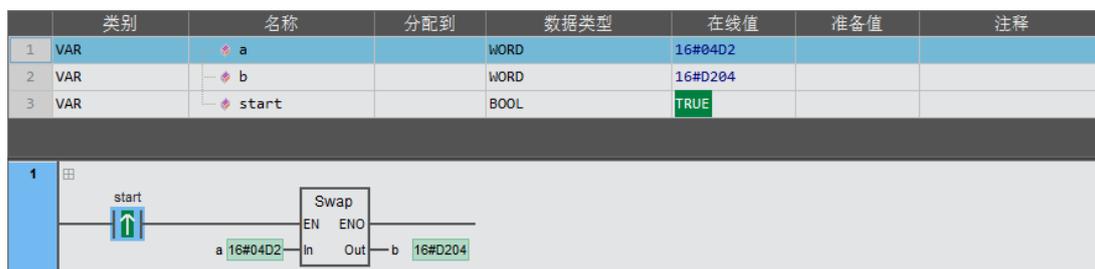
	布尔	位串					整数							实数		时间, 日期					字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
In			○						○												
Out			○						○												

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于输入变量中的值进行高低字节交换后放入输出变量。
- 两个变量的数据类型须一致。
- 该指令执行时，建议该指令的执行条件为上升沿。该指令的执行条件一直为 TRUE，则两个变量的值会一直进行交换。
- 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
IF EDGEPOS(start) THEN
    b:=Swap(In:=a);
END_IF;
```

## 2.3 比较运算

### 2.3.1 EQ (等于)

本指令用于比较多个变量或常量是否全部相等，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
EQ	等于	FUN		$\text{Out} := (\text{In1} = \text{In2}) \& (\text{In2} = \text{In3}) \& (\text{InN-1} = \text{InN});$

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	比较对象	输入	比较对象	变量的数据类型决定
In2至InN	比较对象	输入	比较对象，程序编写时，梯形图时可通过编程软件增加或减少运算对象，即N = 2 ~ 8	变量的数据类型决定
Out	比较结果	输出	比较结果	变量的数据类型决定

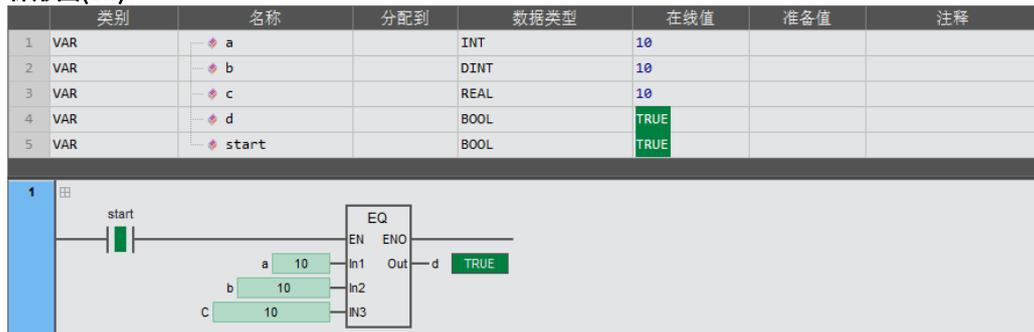
	布尔		位串					整数					实数		时间, 日期				字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
In1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
In2至InN	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Out	○																				

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于比较 In1 至 InN 个变量或常量是否全部相等，结果输出至“Out”，即  $\text{Out} := (\text{In1} = \text{In2}) \& (\text{In2} = \text{In3}) \& (\text{InN-1} = \text{InN})$ 。输入值全部相等时，比较结果“Out”的值为 TRUE，其余情况下，比较结果“Out”的值为 FALSE。
- 输入变量类型为位串、整数和实数类型时，本指令允许输入变量“In1”~“InN”为不同数据类型的变量。当“In1”~“InN”为不同数据类型的变量时，以能包含“In1”~“InN”所有取值范围的数据类型进行运算。如“In1”的数据类型为 INT，“In2”的数据类型为 DINT，则以 DINT 数据类型进行运算处理。
- 当输入变量的数据类型为 BOOL、TIME、DATE、TOD、DT、STRING 中的一种时，要求输入变量 In1~InN 均为该数据类型。如 In1 的数据类型为 DATE，则 In2~InN 的数据类型必须为 DATE，否则软件编译时将报错。
- 示例程序如下所示：

#### 梯形图(LD)：



**结构化文本(ST):**

IF start THEN

d:=(a=b)&amp;(b= c);

END\_IF;

### 2.3.2 NE (不等于)

本指令用于输入变量中的值进行高低字节交换后放入输出变量中。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
NE	不等于	FUN		Out:=In1<>In2;

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	比较对象	输入	比较对象	变量的数据类型决定
In2	比较对象	输入	比较对象	变量的数据类型决定
Out	比较结果	输出	比较结果	变量的数据类型决定

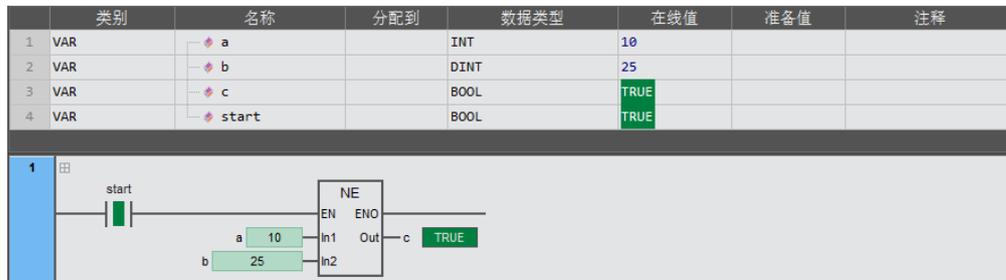
	布尔		位串					整数						实数		时间, 日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	<input type="radio"/>																			
In2	<input type="radio"/>																			
Out	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于比较 2 个变量或常量是否不相等，结果输出至“Out”，即 Out:=In1<>In2。输入值不全部相等时，比较结果“Out”的值为 TRUE，其余情况下，比较结果“Out”的值为 FALSE。
- 输入变量类型为位串、整数和实数类型时，本指令允许输入变量“In1”~“In2”为不同数据类型的变量。当“In1”~“In2”为不同数据类型的变量时，以能包含“In1”~“In2”所有取值范围的数据类型进行运算。如“In1”的数据类型为 INT，“In2”的数据类型为 DINT，则以 DINT 数据类型进行运算处理。
- 当输入变量的数据类型为 BOOL、TIME、DATE、TOD、DT、STRING 中的一种时，要求输入变量 In1~In2 均为该数据类型。如 In1 的数据类型为 TIME，则 In2 的数据类型必须为 TIME，否则软件编译时将报错。
- 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
    c:=a<>b;
END_IF;
```

### 2.3.3 GT/GE/LT/LE (大于/大于等于/小于/小于等于)

这些指令用于对多个变量或常量进行比较，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
GT	大于	FUN		$Out := (In1 > In2) \& (In2 > In3) \& (InN-1 > InN);$
GE	大于等于	FUN		$Out := (In1 \geq In2) \& (In2 \geq In3) \& (InN-1 \geq InN);$
LT	小于	FUN		$Out := (In1 < In2) \& (In2 < In3) \& (InN-1 < InN);$
LE	小于等于	FUN		$Out := (In1 \leq In2) \& (In2 \leq In3) \& (InN-1 \leq InN);$

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	比较对象	输入	比较对象	变量的数据类型决定
In2至InN	比较对象	输入	比较对象，程序编写时，梯形图时可通过编程软件增加或减少运算对象，即N = 2 ~ 8	变量的数据类型决定
Out	比较结果	输出	比较结果	变量的数据类型决定

	布尔		位串					整数					实数		时间, 日期				字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	<input type="radio"/>																			
In2至InN	<input type="radio"/>																			
Out	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

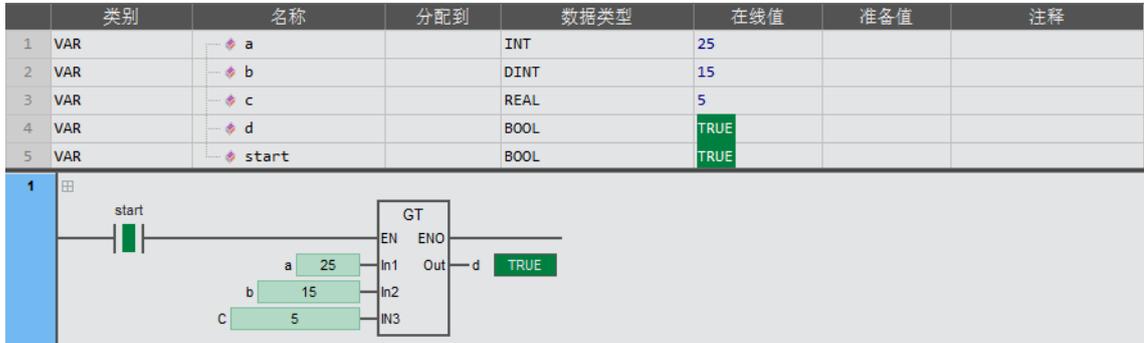
#### ■ 功能说明

- 这些指令用于对 In1~InN 个变量或常量进行比较，结果输出至“Out”，比较结果“Out”的值定义如下表所示：

指令	“Q” 的值
GT	In1>In2>...>InN时，“Out”的值为TRUE，否则为FALSE
GE	In1>=In2>=...>=InN时，“Out”的值为TRUE，否则为FALSE
LT	In1<In2<...<InN时，“Out”的值为TRUE，否则为FALSE
LE	In1<=In2<=...<=InN时，“Out”的值为TRUE，否则为FALSE

- 输入变量类型为位串、整数和实数类型时，本指令允许输入变量 “In1” ~ “InN” 为不同数据类型的变量。当 “In1” ~ “InN” 为不同数据类型的变量时，以能包含 “In1” ~ “InN” 所有取值范围的数据类型进行运算。如 “In1” 的数据类型为 INT，“In2” 的数据类型为 DINT，则以 DINT 数据类型进行运算处理。
- 当输入变量的数据类型为 BOOL、TIME、DATE、TOD、DT、STRING 中的一种时，要求输入变量 In1~InN 均为该数据类型。如 In1 的数据类型为 TIME，则 In2~InN 的数据类型必须为 TIME，否则软件编译时将报错。
- GT 示例程序如下所示：

**梯形图(LD):**

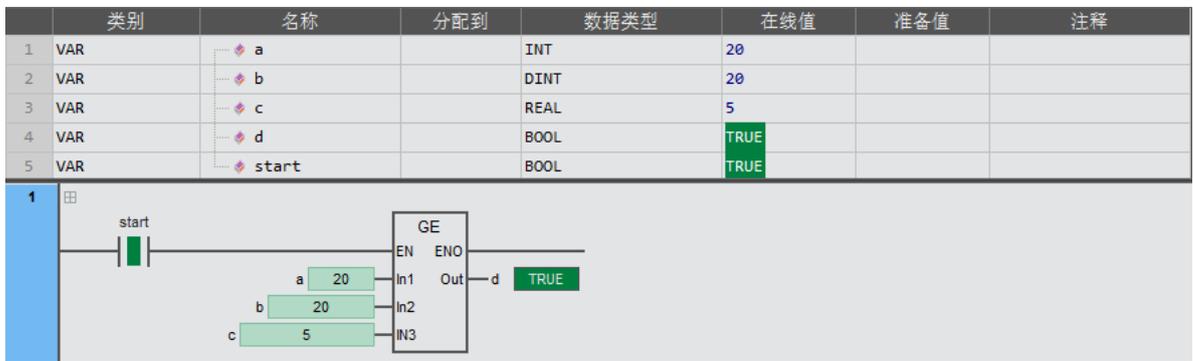


**结构化文本(ST):**

```
IF start THEN
    d:=(a>b)&(b>c);
END_IF;
```

- GE 示例程序如下所示：

**梯形图(LD):**

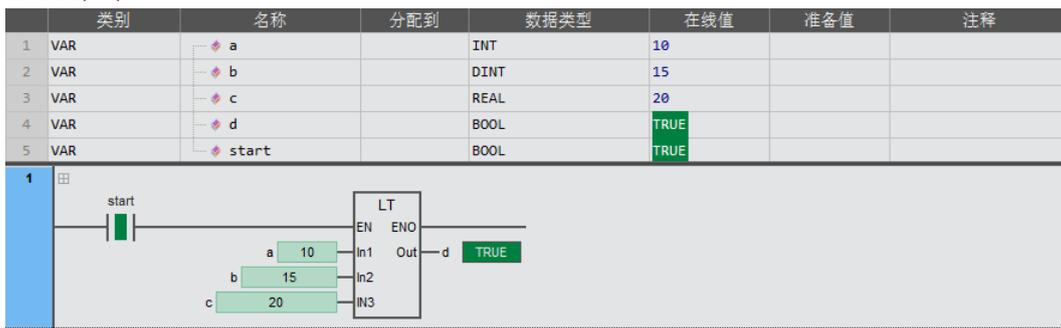


**结构化文本(ST):**

```
IF start THEN
    d:=(a>=b)&(b>=c);
END_IF;
```

- LT 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

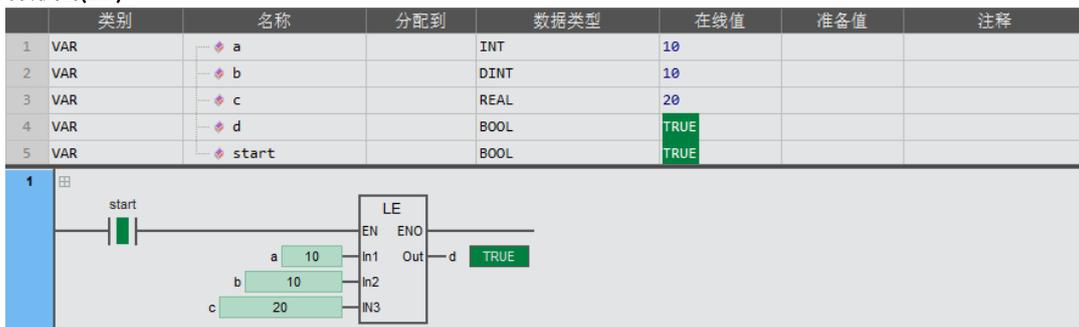
```
IF start THEN
```

```
    d:=(a<b)&(b<c);
```

```
END_IF;
```

- LE 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
```

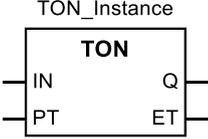
```
    d:=(a<=b)&(b<=c);
```

```
END_IF;
```

## 2.4 定时器

### 2.4.1 TON (通电延时定时器)

定时器定时条件成立时，经过设定的定时时间后输出为TRUE。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
TON	通电延时定时器	FB		<pre>TON_instance ( In := 《参数》, PT := 《参数》, Q =&gt; 《参数》, ET =&gt; 《参数》, );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	定时器输入	输入	从FALSE变为TRUE时定时器计时开始时，持续为TRUE计时时到“Q”变为TRUE。为FALSE时定时器计时复位为0	FALSE或者TRUE
PT	设定定时时间		设定从“In”从FALSE变为TRUE并持续到“Q”变为TRUE的定时时间	参考下方*处的说明
Q	定时时间到标志	输出	定时器设定时间到达时为TRUE，其它为FALSE	FALSE或者TRUE
ET	当前计时时间		定时器“In”从FALSE变为TRUE后，TRUE持续的时间	参考下方*处的说明

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

\*: T#0ns ~ 213503d23h34m33s709ms551us615ns

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

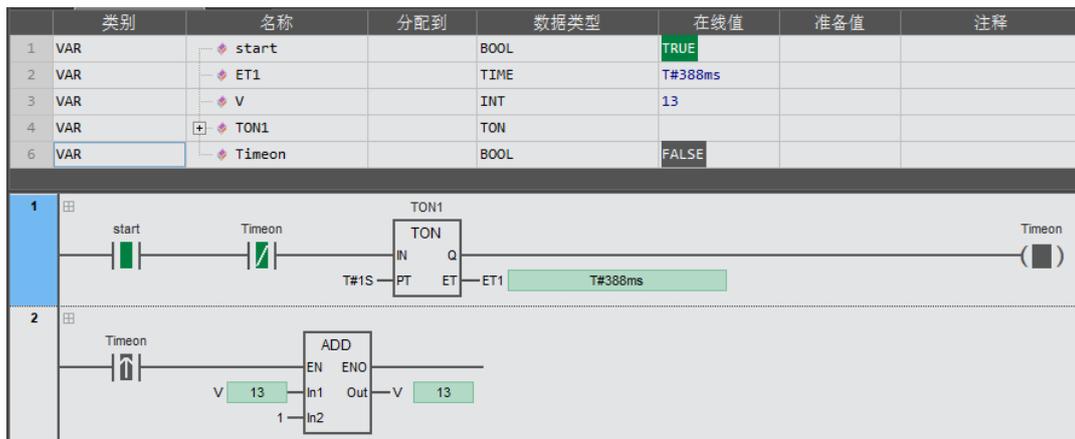
#### ■ 功能说明

- 定时器定时条件成立时，经过设定的定时时间后输出变为 TRUE 的定时器。定时器相关时间单位为纳秒（ns）。
- 定时器“In”为 TRUE 时，定时器开始计时，“ET”显示当前计时的时间。定时器到达“PT”设定的时间时，输出“Q”变为 TRUE，“ET”值不再变化。定时时间到达后，“In”变为 FALSE 时，输出“Q”变为 FALSE，“ET”值复位为 0。
- 定时器启动后，如果“ET”值没有计时到“PT”值设定时间时，“In”变为 FALSE，则输出“Q”变为 FALSE，“ET”值复位为 0。

- 示例程序如下所示：

### 梯形图(LD):

1秒触发1次，变量加1的示例。Start为TRUE时，V的值间隔1秒加1。



### 结构化文本(ST):

```

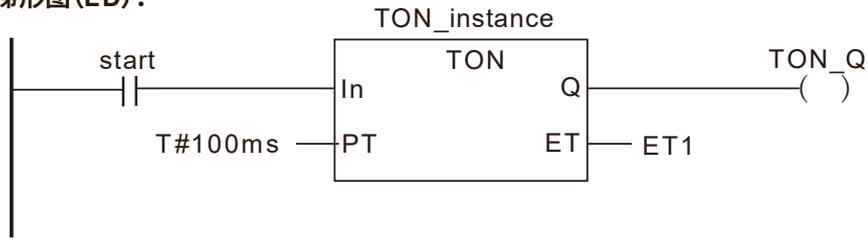
TON1 (IN:=start AND (NOT Timeon) ,PT:= T#1s ,Q=>Timeon ,ET=>ET1);
IF EDGEPOS(Timeon) THEN;
    V:=V+1;
END_IF;

```

- 时序图如下所示：

时序图：

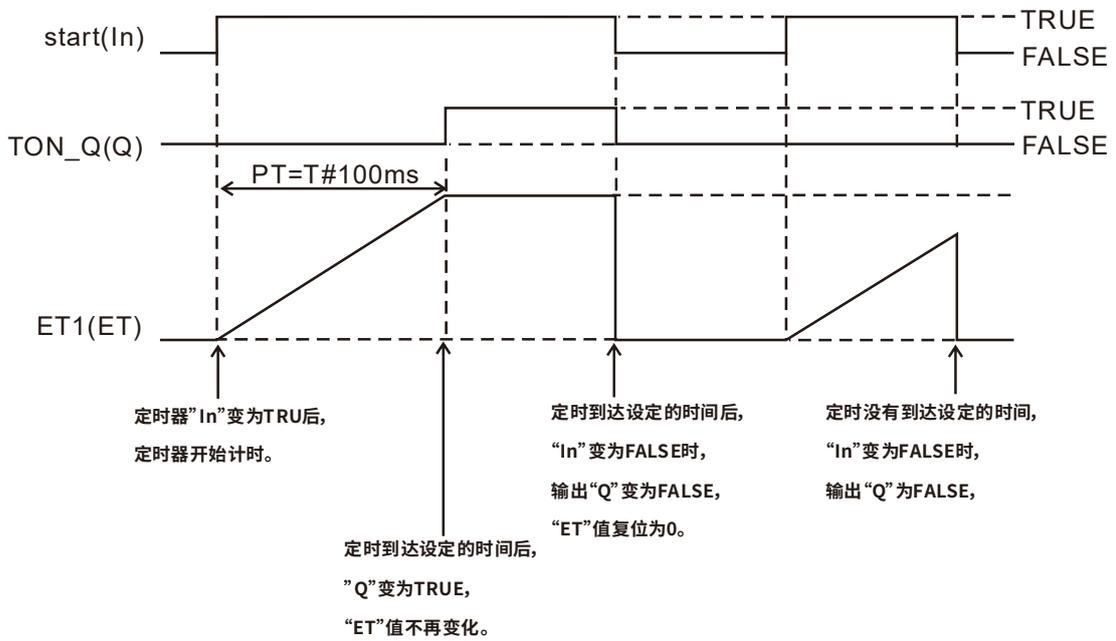
梯形图(LD)：



结构化文本(ST)：

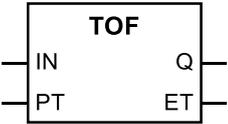
TON\_instance(IN:=start ,PT:= T#100ms ,Q=>TON\_Q ,ET=>ET1 );

时序图：



## 2.4.2 TOF (断电延时定时器)

定时器定时条件成立时，经过设定的定时时间后输出为FALSE。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
TOF	断电延时定时器	FB		<pre>TOF_instance (   In := 《参数》,   PT := 《参数》,   Q =&gt; 《参数》,   ET =&gt; 《参数》 );</pre>

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	定时器输入	输入	从 TRUE 变为 FALSE 时定时器开始计时，持续为 FALSE 时计时到“Q”变为 FALSE。为 TRUE 时定时器计时复位为 0	FALSE 或者 TRUE
PT	设定时间		设定从“In”从 TRUE 变为 FALSE 并持续到“Q”变为 FALSE 的定时时间	参考下方*处的说明
Q	定时时间到标志	输出	当前计时时间到达设定时间为 FALSE，其它为 TRUE	FALSE 或者 TRUE
ET	当前计时时间		定时器“In”从 TRUE 变为 FALSE 后，FALSE 持续的时间	参考下方*处的说明

\*: T#0ns ~ 213503d23h34m33s709ms551us615ns

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

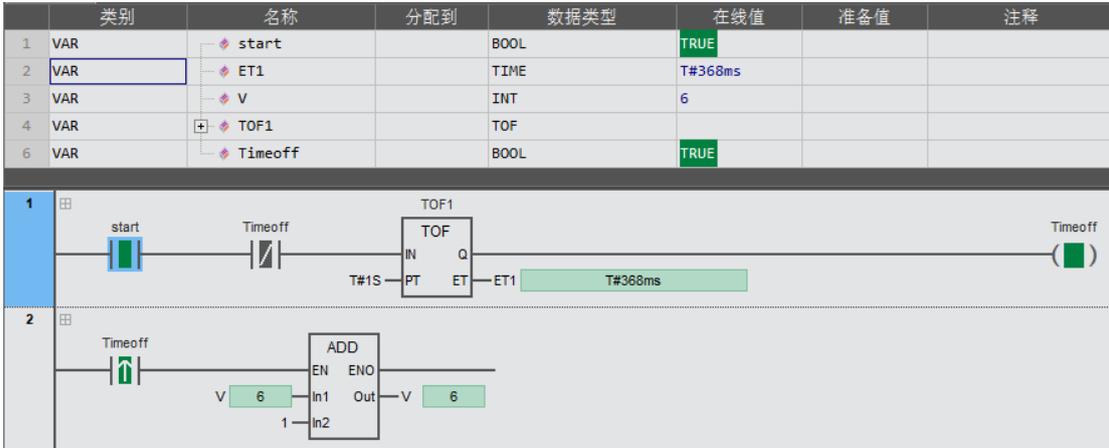
### ■ 功能说明

- 定时器定时条件成立时，经过设定的定时时间后，输出由 TRUE 变为 FALSE 的定时器。
- 定时器相关时间单位为纳秒（ns）。
- 定时器“In”为 TRUE 时，输出“Q”为 TRUE，“ET”值复位为 0。
- 定时器“In”从 TRUE 变为 FALSE 时，定时器开始计时，输出“Q”为 TRUE，“ET”显示当前计时的时间。定时器到达“PT”设定的时间时，输出“Q”变为 FALSE，“ET”值不再变化。定时时间到达后，“In”变为 TRUE 时，输出“Q”为 TRUE，“ET”值复位为 0。
- 定时器启动后，如果“ET”值没有计时到“PT”值设定时间时，“In”变为 FALSE，输出“Q”仍为 TRUE，“ET”值复位为 0。

- 示例程序如下所示：

**梯形图(LD):**

1秒触发1次变量加1的示例。Start为TRUE时，V的值间隔1秒加1。



**结构化文本(ST):**

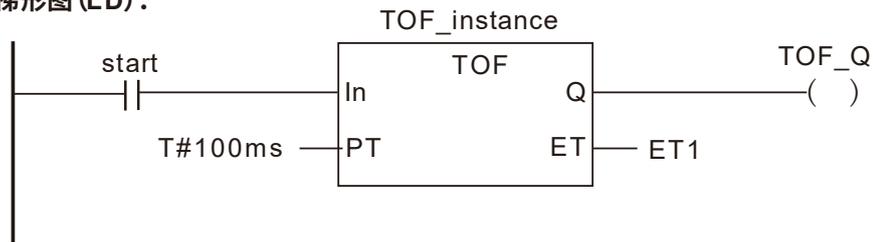
```

TOF1 (IN:=start AND (NOT Timeoff) ,PT:= T#1s ,Q=>Timeoff ,ET=>ET1 );
IF EDGEPOS(Timeoff) THEN;
    V:=V+1;
END_IF;
    
```

- 时序图如下所示：

时序图：

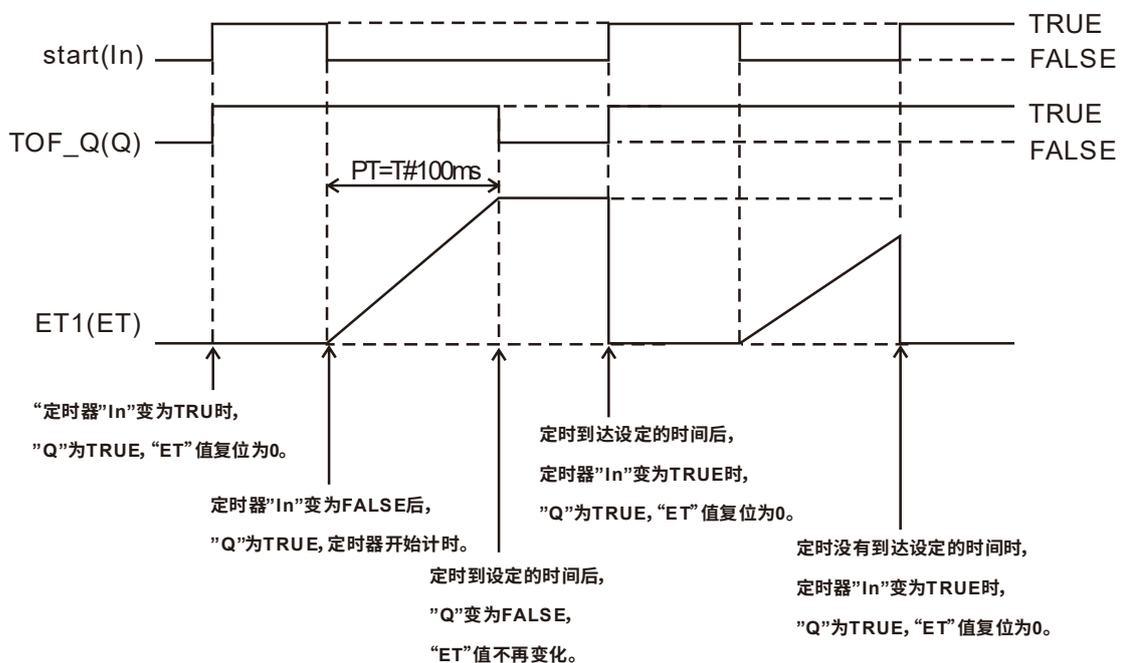
梯形图 (LD)：



结构化文本 (ST)：

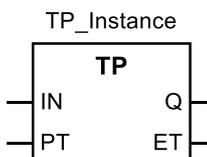
```
TOF_instance(IN:=start,PT:= T#100ms ,Q=>TOF_Q ,ET=>ET1 );
```

时序图：



### 2.4.3 TP (脉冲延时定时器)

检测到输入的上升沿后开始计时，经过设定的定时时间后输出从TRUE变为FALSE的定时器。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
TP	脉冲延时定时器	FB		<pre>TP_instance (   In := 《参数》,   PT := 《参数》,   Q =&gt; 《参数》,   ET =&gt; 《参数》 );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	定时器输入	输入	“In” 从FALSE变为TRUE时，定时器开始计时	FALSE或者TRUE
PT	设定定时时间		设定 “In” 从FALSE变为TRUE到 “Q”从TRUE 变为FALSE的定时时间	参考下方*处的说明
Q	定时时间到标志	输出	定时器设定时间到达时从TRUE 变为FALSE；定时器未启动时为FALSE	FALSE或者TRUE
ET	当前计时时间		“In” 从FALSE变为TRUE时定时器启动，定时器启动后经过的时间	参考下方*处的说明

\*: T#0ns ~ 213503d23h34m33s709ms551us615ns

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

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 定时器启动后，经过设定的定时时间后，输出由 TRUE 变为 FALSE 的定时器。定时器相关时间单位为纳秒（ns）。
- 定时器 “In” 检测到上升沿时，定时器开始计时，同时输出 “Q” 变为 TRUE，当前计时时间 “ET” 显示当前计时的时间。定时器到达 “PT” 设定的时间时，输出 “Q” 变为 FALSE，“ET” 值停止增加。定时器计时到达设定时间后，“In” 从 TRUE 变为 FALSE 时，输出 “Q” 仍为 FALSE，当前计时时间 “ET” 值复位为 0。
- 定时器启动后，没有到达 “PT” 设定的时间时，即使 “In” 从 TRUE 变为 FALSE，定时器会继续计时到达 “PT” 设定的时间。定时器到达 “PT” 设定的时间时，输出 “Q” 变为 FALSE，“ET” 值复位为 0。
- 示例程序如下所示：

#### 梯形图(LD):

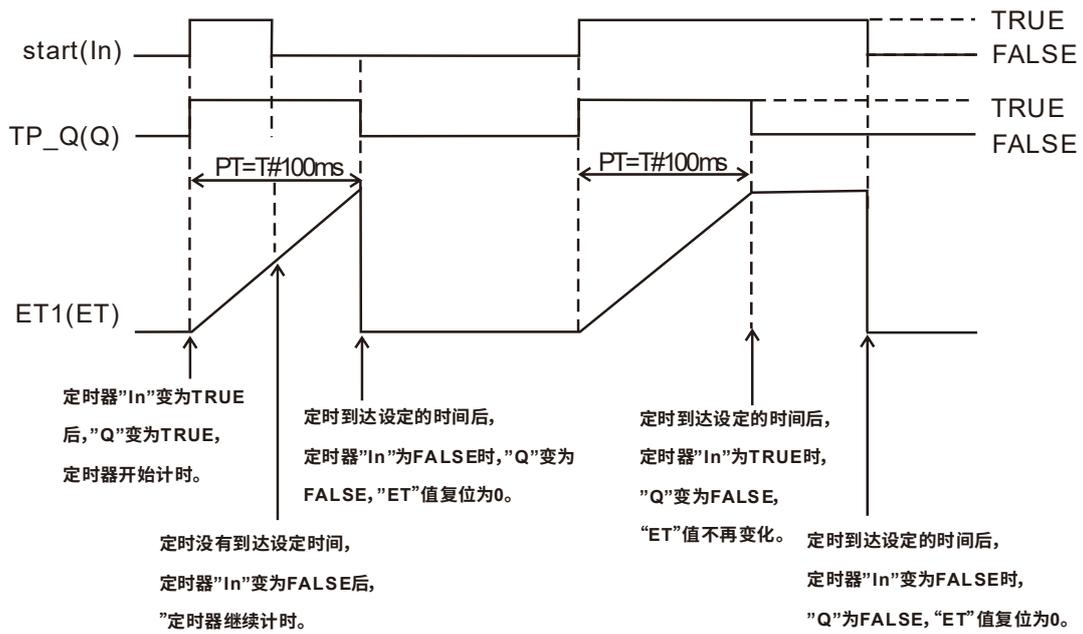


#### 结构化文本(ST):

```
TP1(EN:= TRUE,IN:=start,PT:=T#100ms,ENO=>,Q=>TP_Q,ET=>ET1);
```

- 时序图如下所示：

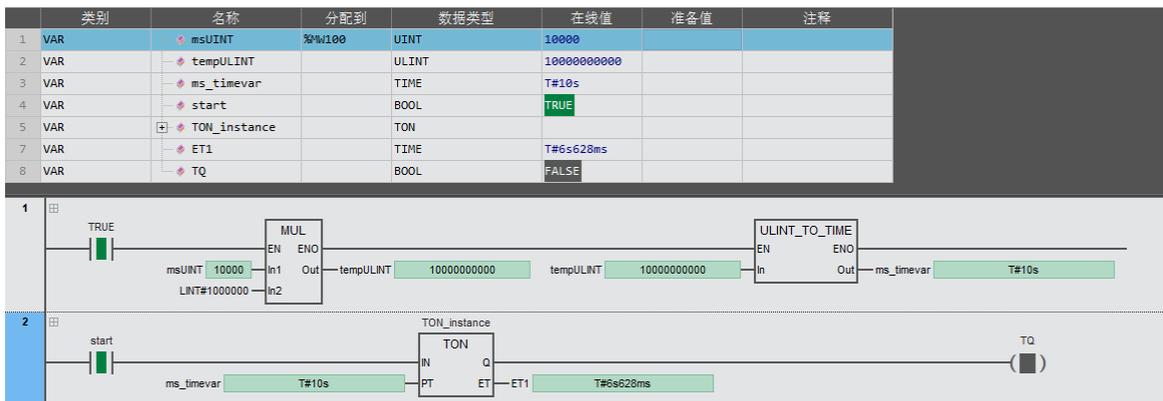
时序图：



### 2.4.4 通过触摸屏修改定时器定时时间示例程序

- 定时器相关指令输入参数“PT”为 TIME 类型数据，单位为 ns(纳秒)。当触摸屏和控制器通讯时，如果触摸屏不支持 TIME 类型，而且客户想通过触摸屏设定以 ms（毫秒）为单位的定时时间，可以按照该示例程序进行转换。转换时将变量 msUINT（UINT 类型）和控制器的装置绑定，下图所示的示例程序中将 msUINT 变量和%MW100 绑定（通过变量表中“分配到”指定装置）。
- $1\text{ms}=10^6\text{ns}$ 。该示例程序中先将变量 msUINT 的值乘以 1000000（ms 转换为 ns），所得的结果放入 ULINT 类型变量 tempULINT 内，然后再通过 ULINT\_TO\_TIME 指令将 ULINT 类型变量转换为 TIME 类型的变量，因为定时器指令输入参数“PT”为 TIME 类型数据，从而实现 ms 到 ns 的转换。
- 当需要定时的时间为 10S 时，只需要将 msUINT 设置为 10000 即可。示例程序如下：

#### 梯形图(LD):



#### 结构化文本(ST):

```
tempULINT:=msUINT*LINT#1000000;
mstimevar:=ULINT_TO_TIME(tempULINT);
TON_instance(In:=start, PT:= mstimevar,ET=>ET1, Q=>TQ);
```

## 2.5 计数器

### 2.5.1 CTU (加计数器)

每次检测到输入信号的上升沿时进行加1运算的计数器。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
CTU	加计数器	FB		<pre>CTU_instance (   CU := 《参数》,   Reset := 《参数》,   PV := 《参数》,   Q =&gt; 《参数》,   CV =&gt; 《参数》 );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
CU	计数输入信号	输入	CU从FALSE变为TRUE时, CV值加1	FALSE或者TRUE
Reset	复位信号		为TRUE时将 CV值复位为0	FALSE或者TRUE
PV	设定值		计数器的设定值	0 ~ 4294967295
Q	计数完成标志	输出	当前计数值到达设定值标志 TRUE: CV值等于PV值时 FALSE: CV值不等于PV值时	FALSE或者TRUE
CV	当前计数值		计数器的当前计数值	0 ~ 4294967295

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

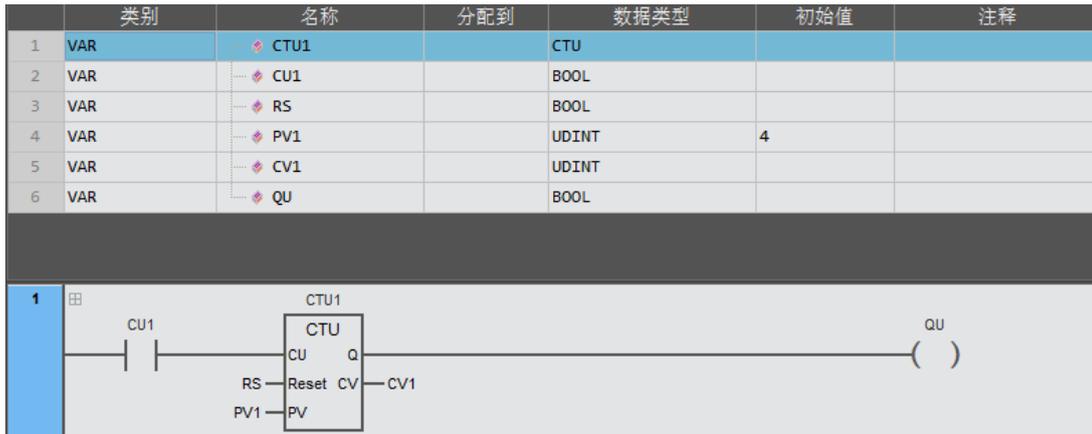
\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 每次检测到输入信号的上升沿时进行加 1 运算的计数器。
- 计数器输入信号 CU 从 FALSE 变为 TRUE 时, “CV” 值加 1, “CV” 值等于 “PV” 设定值时, “CV” 值停止增加, 同时 “Q” 从 FALSE 变为 TRUE。
- 计数器复位信号 “Reset” 为 TRUE 时, 计数完成标志 “Q” 变为 FALSE, 当前计数值 CV 值复位为 0。计数器复位信号 “Reset” 为 TRUE 时, 输入信号 CU 值无效, 即 CV 值不能通过输入信号 CU 进行计数。

- 示例程序如下所示：

**梯形图(LD):**

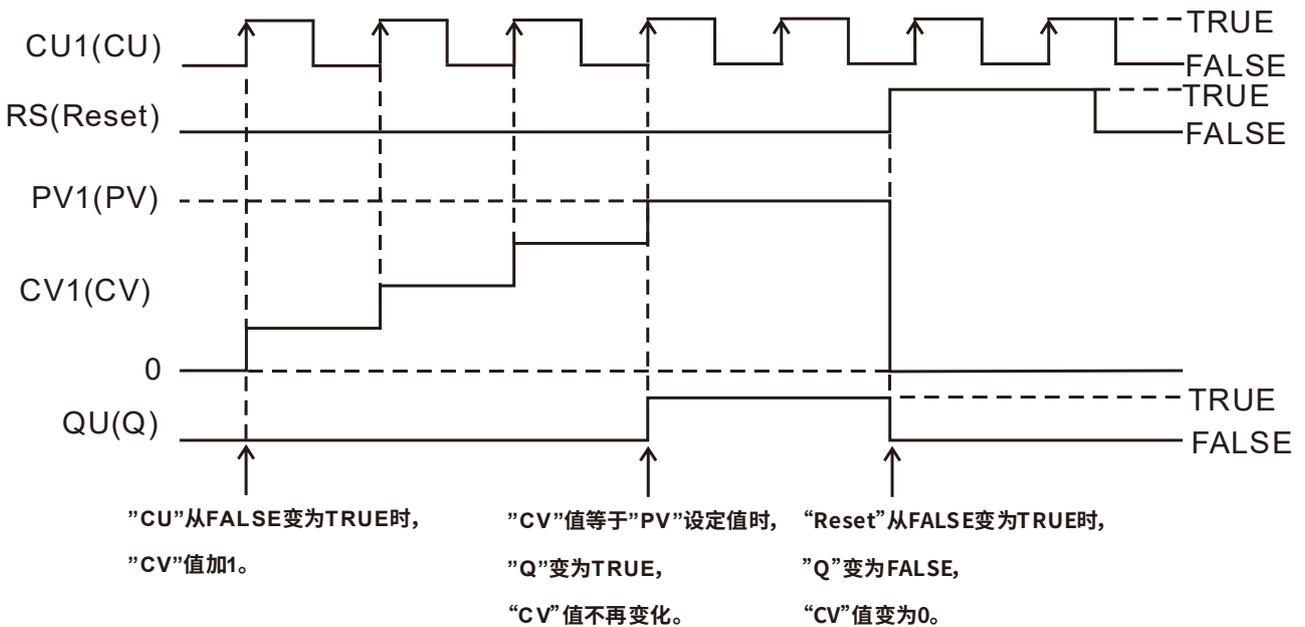


**结构化文本(ST):**

CTU1(CU:=CU1,Reset:=RS,PV:=PV1 ,Q=>QU ,CV=>CV1 );

- 时序图如下所示：

**时序图:**



## 2.5.2 CTD (减计数器)

每次检测到输入信号的上升沿时进行减1运算的计数器。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
CTD	减计数器	FB		<pre>CTD_instance (   CD := 《参数》,   Load := 《参数》,   PV := 《参数》,   Q =&gt; 《参数》,   CV =&gt; 《参数》 );</pre>

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
CD	计数输入信号	输入	CD从FALSE变为TRUE时, CV值减1	FALSE或者TRUE
Load	装载信号		为TRUE时将 PV值搬运到CV内	FALSE或者TRUE
PV	设定值		计数器的设定值	0 ~ 4294967295
Q	计数完成标志	输出	当前计数值减小到0标志 TRUE: CV值等于0时 FALSE: CV值不等于0时	FALSE或者TRUE
CV	当前计数值		计数器的当前计数值	0 ~ 4294967295

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

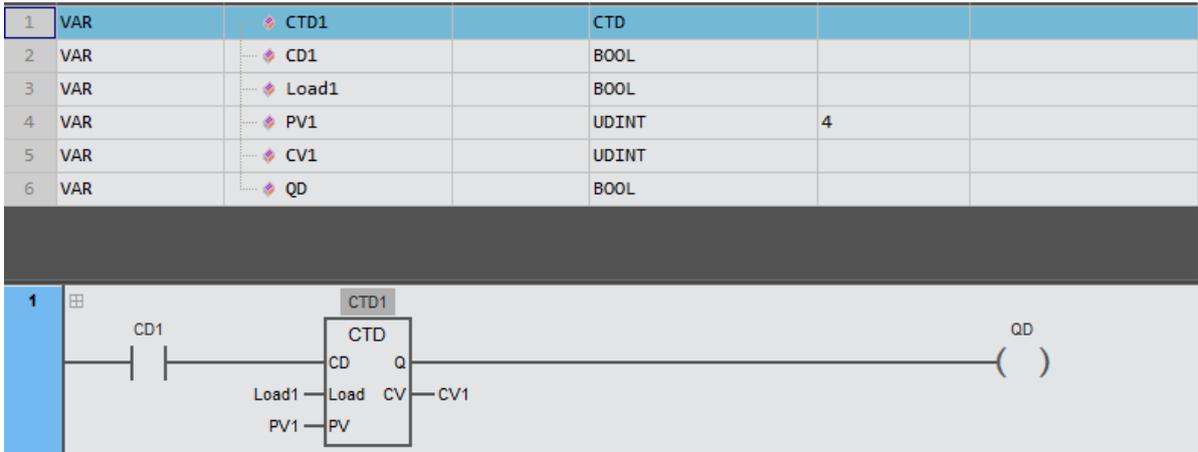
\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 每次检测到输入信号的上升沿时进行减 1 运算的计数器。
- 计数器装载信号“Load”为 TRUE 时, PV 值搬运到 CV 内, 计数完成标志“Q”为 FALSE。计数器装载信号“Load”为 TRUE 时, 输入信号 CD 值无效, 即 CV 值不能通过输入信号 CD 进行计数, 计数时需要将装载信号“Load”变为 FALSE。
- 计数器装载信号“Load”从 TRUE 变为 FALSE 后, 计数器输入信号 CD 从 FALSE 变为 TRUE 时, “CV”值减 1, CV 值等于 0 时, CV 值停止减小, 同时“Q”从 FALSE 变为 TRUE。

- 示例程序如下所示：

**梯形图(LD):**

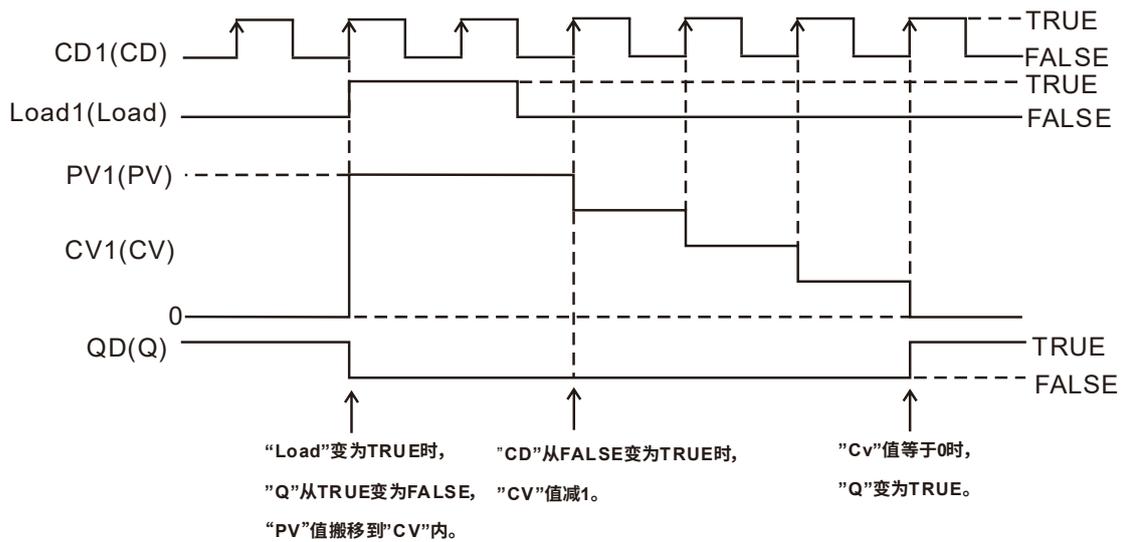


**结构化文本(ST):**

CTD1(CD:=CD1,Load:=Load1,PV:=PV1,Q=>QD,CV=>CV1);

- 时序图如下所示：

**时序图:**



### 2.5.3 CTUD (加减计数器)

根据加法计数输入信号的上升沿和减法计数输入信号的上升沿进行加1或减1运算的计数器。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
CTUD	加减计数器	FB		<pre>CTUD_instance (   CU := 《参数》,   CD := 《参数》,   Reset := 《参数》,   Load := 《参数》,   PV := 《参数》,   Q =&gt; 《参数》,   CV =&gt; 《参数》 );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
CU	加计数输入信号	输入	CU从FALSE变为TRUE时, CV值加1	FALSE或者TRUE
CD	减计数输入信号		CD从FALSE变为TRUE时, CV值减1	FALSE或者TRUE
Reset	复位信号		为TRUE时将 CV值复位为0	FALSE或者TRUE
Load	装载信号		为TRUE时将 PV值搬移到CV内	FALSE或者TRUE
PV	设定值		计数器的设定值	0 ~ 4294967295
QU	增计数完成标志	输出	当前计数值到达设定值标志 TRUE: CV值等于PV值时 FALSE: CV值不等于PV值时	FALSE或者TRUE
QD	减计数完成标志		当前计数值减小到0标志 TRUE: CV值等于0时 FALSE: CV值不等于0时	FALSE或者TRUE
CV	当前计数值		计数器的当前计数值	0 ~ 4294967295

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

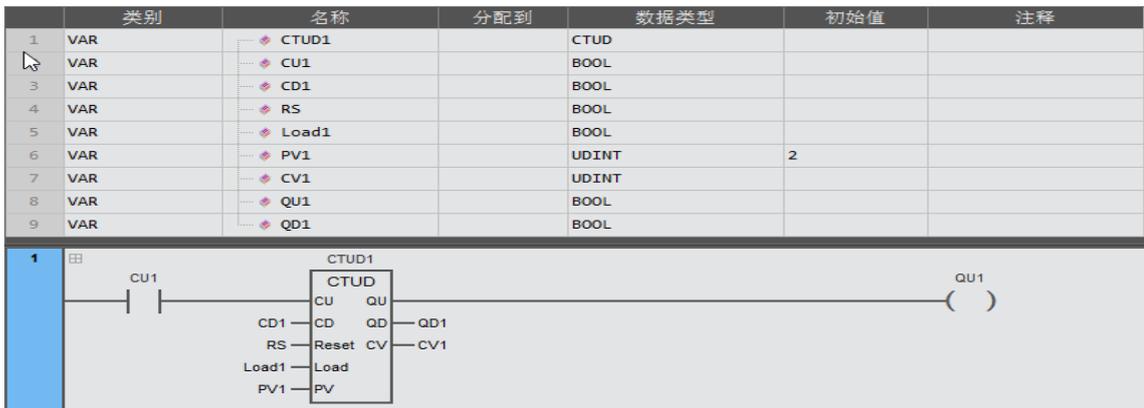
\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 根据加法计数输入信号的上升沿和减法计数输入信号的上升沿进行加 1 或减 1 运算的计数器。
- “Reset”和“Load”为 FALSE 时, 加计数输入信号 CU 从 FALSE 变为 TRUE 时, CV 值加 1, CV 值等于 PV 设定值时, CV 值停止增加, 同时“QU”从 FALSE 变为 TRUE。

- “Reset”和“Load”为FALSE时，减计数输入信号CD从FALSE变为TRUE时，CV值减1，CV值等于0时，CV值停止减小，同时“QD”从FALSE变为TRUE。
- 计数器复位信号“Reset”为TRUE时，CV值变为0，增计数完成标志“QU”为FALSE，减计数完成标志“QD”为TRUE。
- 计数器装载信号“Load”为TRUE时，PV值搬移到CV内，计数完成标志“QD”变为FALSE，“QU”变为TRUE。计数器装载信号“Load”为TRUE时，输入信号CD值无效，即CV值不能通过输入信号CD进行计数，计数时需要把装载信号“Load”变为FALSE。
- “Reset”和“Load”同为TRUE时，CV值变为0，增计数完成标志“QU”为FALSE，减计数完成标志“QD”为TRUE。
- 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

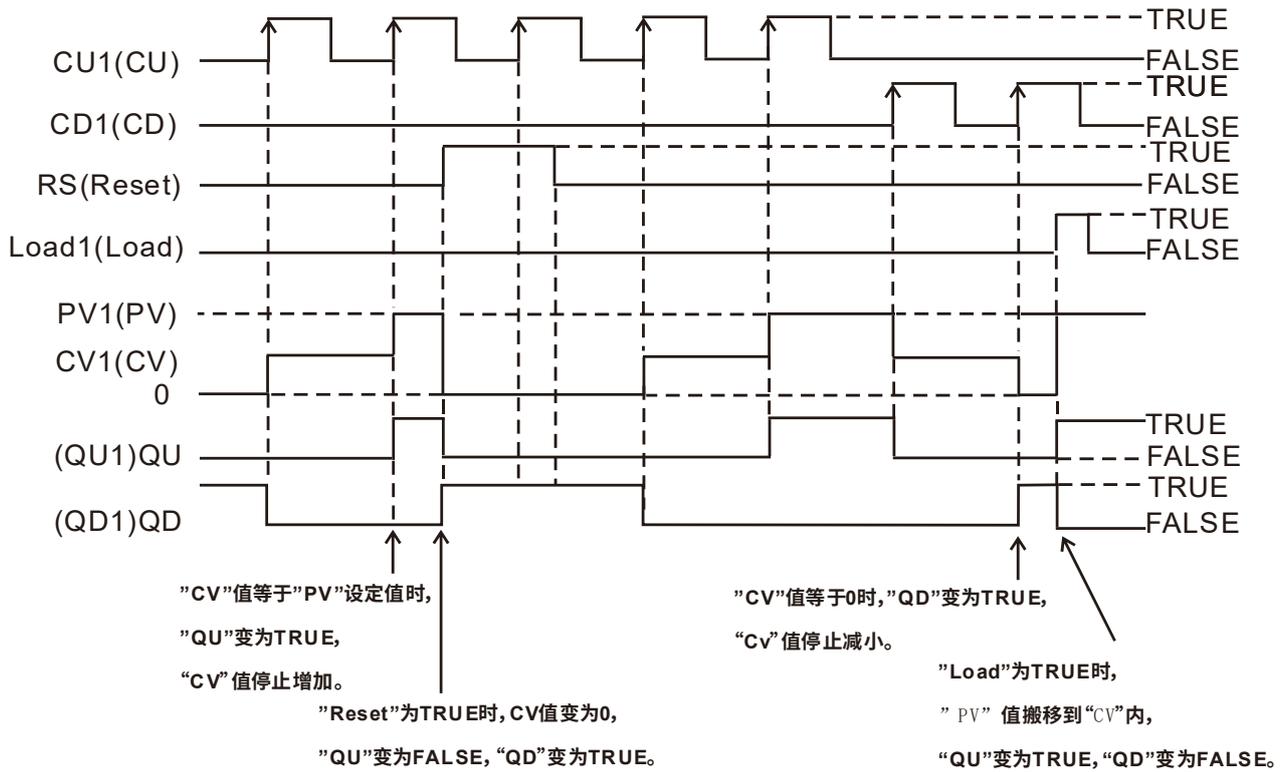
```

CTUD1(
    CU:=CU1 ,
    CD:=CD1 ,
    Reset:=RS ,
    Load:=Load1 ,
    PV:=PV1 ,
    QU=>QU1 ,
    QD=>QD1 ,
    CV=>CV1 );

```

- 时序图如下所示：

时序图：



## 2.6 数学函数

### 2.6.1 ADD (加法)

对位串、整数、实数、时间、日期类型的数据进行加法运算。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
ADD	加法	FUN		Out:=In1+ In2+...+ InN;

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	被加数	输入	被加数	变量的数据类型决定
In2至InN	加数	输入	程序编写时，可通过编程软件增加或减少加数，N的范围为2~8	变量的数据类型决定
Out	和	输出	In1 ~ InN的代数和	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

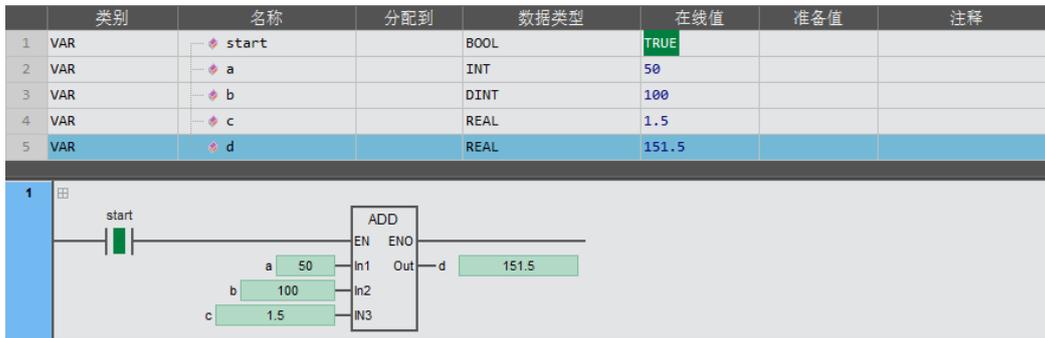
#### ■ 功能说明

- 对位串、整数、实数、时间、日期类型的输入数据进行加法运算，结果输出至“Out”，即  $Out = In1 + In2 + \dots + InN$ 。
- 输入变量类型为位串、整数和实数类型时，允许输入变量“In1”~“InN”数据类型不同。当“In1”~“InN”数据类型不同时，以包含“In1”~“InN”数据类型进行运算。如“In1”的数据类型为SINT，“In2”的数据类型为INT，则以INT进行运算处理。
- “Out”数据类型宽度须大于等于“In1”~“InN”类型的最大宽度，否则软件编译报错。如“In1”和“In2”的数据类型分别为INT和SINT，则“Out”的数据类型应该为INT、DINT等。
- 时间和日期类型的数据进行加法运算时，仅支持以下三种组合：
  1. In1、In2、Out数据类型都为TIME类型。
  2. In1数据类型为TOD (TIME\_OF\_DAY)，In2数据类型为TIME，Out数据类型为TOD。
  3. In1数据类型为DT (DAY\_AND\_TIME)型，In2数据类型为TIME，Out数据类型为DT。

#### △注意

- “In1”~“InN”之和可能超出“Out”数据类型有效范围。如In1、In2数据类型均为INT，数值分别为32767和1，如果Out变量的数据类型为INT，Out的值为-32768；如果Ou数据类型为DINT，则值为32768。

- 示例程序如下所示：

**梯形图(LD):****结构化文本(ST):**

```
IF start THEN  
    d:=a+b+c;  
END_IF;
```

## 2.6.2 SUB (减法)

对位串、整数、实数、时间、日期类型的数据进行减法运算。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SUB	减法	FUN		Out:=In1- In2;

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	被减数	输入	被减数	变量的数据类型决定
In2	减数	输入	减数	变量的数据类型决定
Out	差	输出	In1与In2的差	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

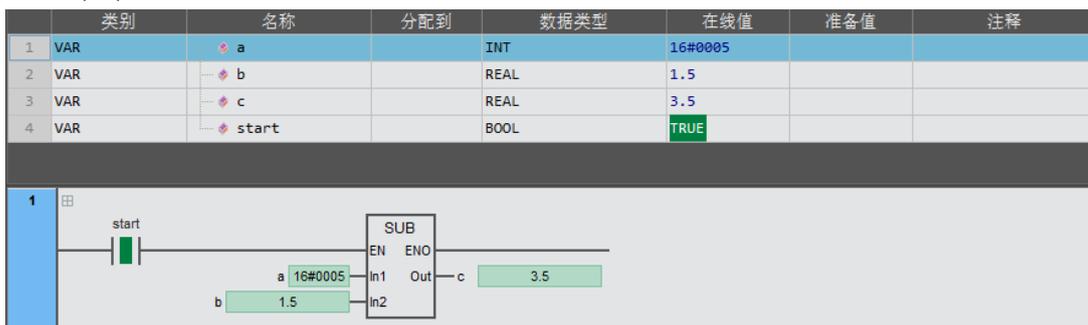
- 将被减数“In1”减去减数“In2”结果输出至“Out”，即  $Out = In1 - In2$ 。
- 输入变量类型为位串、整数和实数类型时，允许输入变量“In1”~“In2”数据类型不同。当“In1”~“In2”数据类型不同时，以包含“In1”~“In2”数据类型进行运算。如“In1”的数据类型为 SINT，“In2”的数据类型为 INT，则以 INT 进行运算处理。
- 对于时间和日期类型，仅支持以下几种类型相减：
  1. In1、In2、Out 数据类型都为 TIME 类型。
  2. In1 数据类型为 TOD，In2 数据类型为 TIME，Out 数据类型为 TOD。
  3. In1、In2 数据类型为 TOD，Out 数据类型为 TIME。
  4. In1、In2 数据类型为 DATE，Out 数据类型为 TIME。
  5. In1、In2 数据类型为 DT，Out 数据类型为 TIME。
  6. In1 数据类型为 DT，In2 数据类型为 TIME，Out 数据类型为 DT。

### △注意

- “In1”与“In2”之差可能超出“Out”数据类型的有效范围。如 In1、In2 数据类型均为 INT，值分别为 -32768 和 1，如果 Out 数据类型为 INT 时，Out 的值为 32767；如果 Out 数据类型为 DINT，则 Out 的值为 -32769。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

```

IF start THEN
    c:=a-b;
END_IF;

```

### 2.6.3 MUL (乘法)

对位串、整数、实数类型的数据进行乘法运算。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MUL	乘法	FUN		Out:=In1* In2*…* InN;

#### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	被乘数	输入	被乘数	变量的数据类型决定
In2至InN	乘数	输入	可通过编程软件增加或减少乘数，N的范围为2~8	变量的数据类型决定
Out	积	输出	In1 ~ InN的积	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

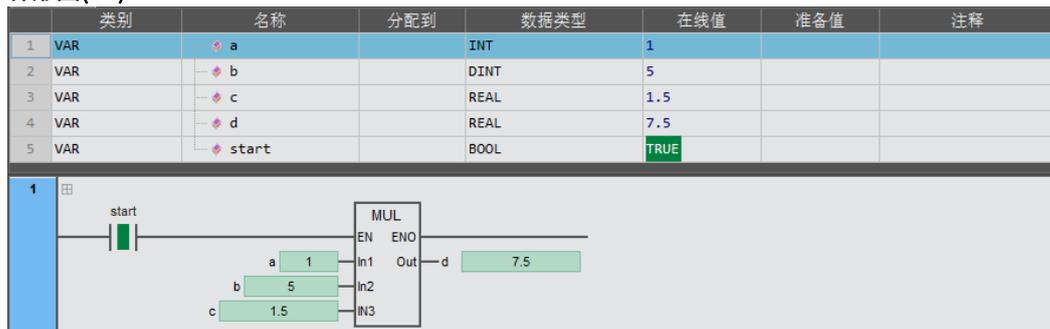
#### 功能说明

- 对位串、整数、实数类型的输入数据进行乘法运算，结果输出至“Out”，即  $Out = In1 * In2 * \dots * InN$ 。
- 输入变量类型为位串、整数和实数类型时，允许输入变量“In1” ~ “InN”数据类型不同。当“In1” ~ “InN”数据类型不同时，以包含“In1” ~ “InN”数据类型进行运算。如“In1”的数据类型为 SINT，“In2”的数据类型为 INT，则以 INT 进行运算处理。
- “Out”数据类型宽度须大于等于“In1” ~ “In2”类型的最大宽度，否则软件编译报错。如“In1”和“In2”的数据类型分别为 INT 和 SINT，则“Out”的数据类型应该为 INT、DINT 等。

#### 注意

- “In1” ~ “InN”之积可能超出“Out”数据类型的有效范围。如 In1、In2 数据类型均为 INT，值分别为 20000 和 2，如果 Out 数据类型为 INT 时，Out 的值为- 25536；如果 Out 数据类型为 DINT，则 Out 值为 40000。
- 示例程序如下所示：

#### 梯形图(LD)：



**结构化文本(ST):**

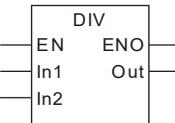
IF start THEN

d:=a\*b\*c;

END\_IF;

## 2.6.4 DIV (除法)

对位串、整数、实数类型的数据进行除法运算。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
DIV	除法	FUN		Out:=In1/In2;

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	被除数	输入	被除数	变量的数据类型决定
In2	除数	输入	除数	变量的数据类型决定
Out	商	输出	In1与In2的商	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

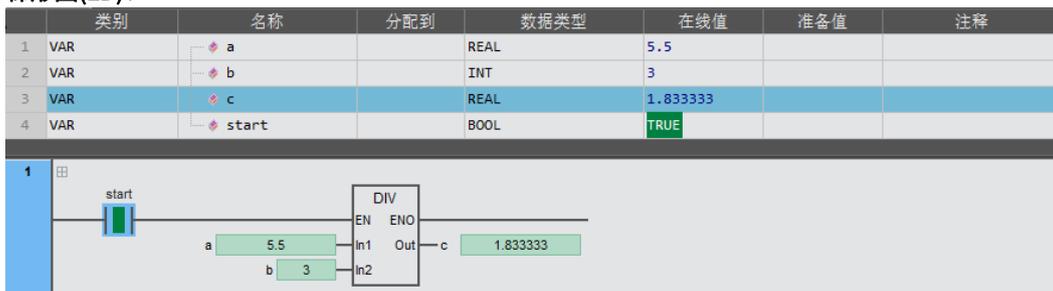
### ■ 功能说明

- 将被除数“In1”除以除数“In2”，结果输出至“Out”，即  $Out = In1 / In2$ 。
- 输入变量类型为位串、整数和实数类型时，允许输入变量“In1”~“In2”数据类型不同。当“In1”~“In2”数据类型不同时，以包含“In1”~“In2”数据类型进行运算。如“In1”的数据类型为 SINT，“In2”的数据类型为 INT，则以 INT 进行运算处理。
- “Out”数据类型宽度须大于等于“In1”~“In2”类型的最大宽度，否则软件编译报错。如“In1”和“In2”的数据类型分别为 INT 和 SINT，则“Out”的数据类型应该为 INT、DINT 等。
- 时间和日期类型的数据进行加法运算时，仅支持以下三种组合：
  1. In1、In2、Out 数据类型都为 TIME 类型。
  2. In1 数据类型为 TOD (TIME\_OF\_DAY)，In2 数据类型为 TIME，Out 数据类型为 TOD。
  3. In1 数据类型为 DT (DAY\_AND\_TIME) 型，In2 数据类型为 TIME，Out 数据类型为 DT。

### ⚠注意

- “In1”与“In2”之商可能超出“Out”数据类型的有效范围。如 In1、In2 数据类型均为 INT，值分别为 -32768 和 -1，如果 Out 数据类型为 INT 时，Out 的值为 -32768；如果 Out 数据类型为 DINT，则 Out 的值为 32768。
- 当除数为 0 时，“Out”的值为 0。

- 示例程序如下所示：

**梯形图(LD):****结构化文本(ST):**

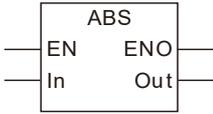
```
IF start THEN
```

```
    c:=a/b;
```

```
END_IF;
```

### 2.6.5 ABS (计算绝对值)

计算位串、整数、实数类型数据的绝对值。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
ABS	绝对值	FUN		Out:= ABS(In);

■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	原始值	输入	原始值	变量的数据类型决定
Out	绝对值	输出	绝对值	变量的数据类型决定

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

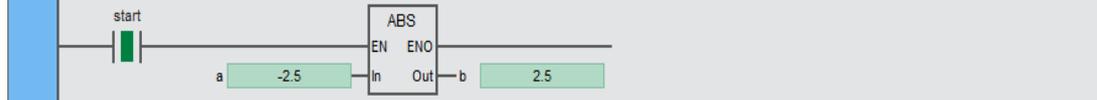
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

■ 功能说明

- 计算“In”的绝对值，结果输出至“Out”。
- 当“In”和“Out”数据类型不同时，“Out”数据类型的宽度须大于或等于“In”数据类型的宽度，否则软件编译报错。
- 示例程序如下所示：

梯形图(LD):

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		REAL	-2.5		
2	VAR	b		REAL	2.5		
3	VAR	start		BOOL	TRUE		

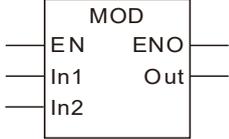
  


结构化文本(ST):

```
IF start THEN
    b:=ABS(a);
END_IF;
```

## 2.6.6 MOD (整数取余)

计算对位串、整数类型数据进行除法运算后的余数。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MOD	整数取余	FUN		Out:=In1 MOD In2;

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	被除数	输入	被除数	变量的数据类型决定
In2	除数	输入	除数	变量的数据类型决定
Out	余	输出	In1除以In2的余数	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 将被除数 In1 除以除数 In2 的余数输出至 Out。
- 输入变量类型为位串、整数和实数类型时，允许输入变量 In1~In2 数据类型不同。当 In1~In2 数据类型不同时，以包含 In1~In2 数据类型进行运算。如 In1 的数据类型为 SINT，In2 的数据类型为 INT，则以 INT 进行运算处理。
- Out 数据类型宽度须大于等于 In1~In2 类型的最大宽度，否则软件编译报错。如 In1 和 In2 的数据类型分别为 INT 和 SINT，则 Out 的数据类型应该为 INT、DINT 等。

### ⚠注意

- 当 In2 的值为 0 时，Out 的值为 0。
- 示例程序如下所示：

#### 梯形图(LD)：



**结构化文本(ST):**

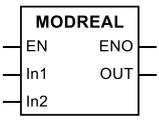
IF start THEN

    c:= a MOD b;

END\_IF;

## 2.6.7 MODREAL (浮点数取余)

计算对实数进行除法运算后的余数。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MODREAL	浮点数取余	FUN		Out:=MODREAL(In1,In2);

### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	被除数	输入	被除数	变量的数据类型决定
In2	除数	输入	除数	变量的数据类型决定
Out	余	输出	In1除以In2的余数	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

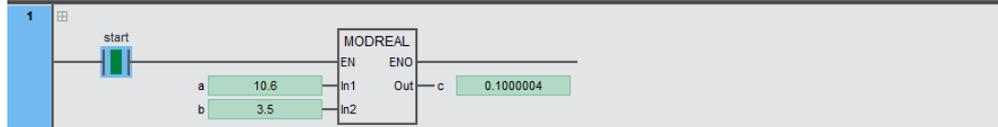
- 将被除数 In1 除以除数 In2 的余数输出至 Out。
- 允许输入变量 In1~In2 数据类型不同。当 In1~In2 数据类型不同时，以包含 In1~In2 数据类型进行运算。如 In1 的数据类型为 REAL，In2 的数据类型为 LREAL 时，则以 LREAL 进行运算处理。
- Out 数据类型宽度须大于等于 In1~In2 类型的最大宽度，否则软件编译报错。

### △注意

- 当 In2 的值为 0 时，Out 的值为 0。
- 示例程序如下所示：

#### 梯形图(LD):

类别	名称	分配到	数据类型	在线值	准备值	注释
1 VAR	a		REAL	10.6		
2 VAR	b		REAL	3.5		
3 VAR	c		REAL	0.1000004		
4 VAR	start		BOOL	TRUE		

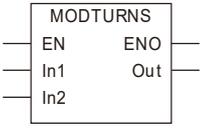
  


#### 结构化文本(ST):

```
IF start THEN
    c:=MODREAL(In1:=a,In2:=b);
END_IF;
```

## 2.6.8 MODTURNS (计算圈数)

对实数进行除法运算后的商取整数。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MODTURNS	计算圈数	FUN		Out:=MODTURNS(In1,In2);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	输入值	输入	输入值	变量的数据类型决定
In2	模数范围	输入	模数范围	变量的数据类型决定
Out	圈数	输出	圈数	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 将“In1”和“In2”进行除法运算，并将所得的商取整后输出至“Out”。“In1”、“In2”及“Out”的值，示例如下表所示。
- 取整规则：向下取整，当商为整数时直接去除小数部分取整；当商为负数去掉小数后需要再-1取整。

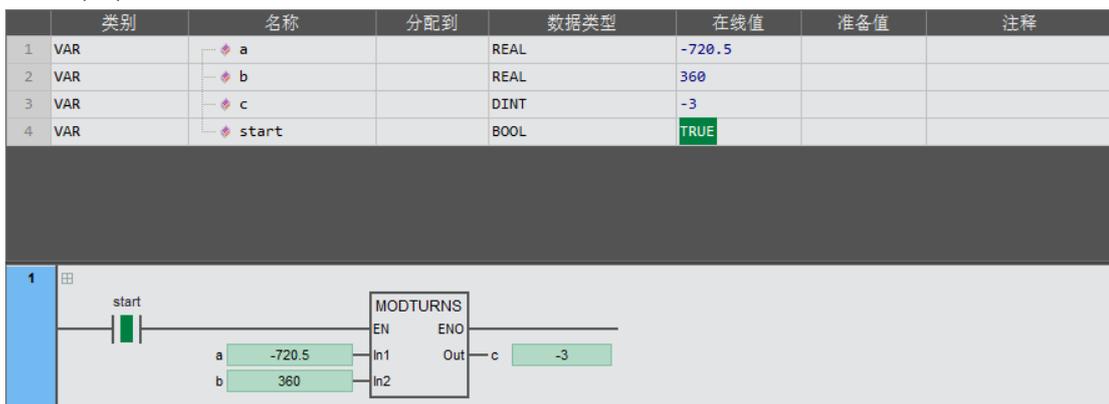
“In1”的值	“In2”的值	“Out”的值
720	360	2
-720	360	-2
721.8	360	2
-721.8	360	-3

#### ⚠注意

- 本指令输入变量与输出变量数据类型不同。
- 当In2的值为0时，Out的值为0。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

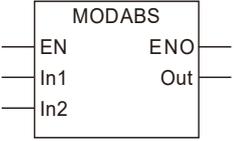
```
IF start THEN
```

```
    c := MODTURNS(In1:=a, In2:=b);
```

```
END_IF;
```

## 2.6.9 MODABS (计算相位)

两个浮点数变量或常量进行运算后所得的相位。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MODABS	计算相位	FUN		Out:= MODABS(In1, In2);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	输入值	输入	输入值	变量的数据类型决定
In2	模数范围	输入	模数范围	变量的数据类型决定
Out	相位	输出	In1除以In2所得的相位	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于计算两个浮点数变量或常量进行运算后所得的相位，结果输出至“Out”。“Out”的值为正数。可通过该指令计算轴的相位或凸轮相位。“In1”、“In2”及“Out”的值示例如下表所示：

“In1”的值	“In2”的值	“Out”的值
540	360	180
-540	360	180
560	360	200
-560	360	160

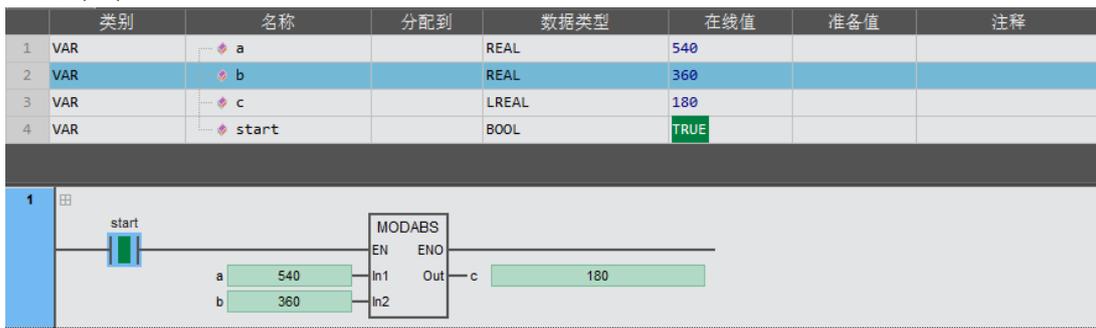
- 允许输入变量“In1”与“In2”为不同数据类型的变量。当“In1”与“In2”为不同数据类型的变量时，以包含“In1”和“In2”所有取值范围的数据类型进行运算。如“In1”的数据类型为REAL，“In2”的数据类型为LREAL，则以LREAL数据类型进行运算处理。
- “Out”数据类型的宽度须大于或等于“In1”和“In2”类型的宽度，否则，软件编译时会报错。如“In1”和“In2”的数据类型分别为REAL和LREAL，则“Out”的数据类型须为LREAL等；如果“Out”链接变量的数据类型为REAL，软件编译时会报错。

### △注意

- 当In2的值为0时，Out的值为0。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

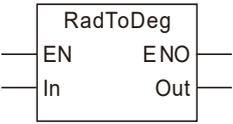
```
IF start THEN
```

```
    c:=MODABS(In1:=a,In2:=b);
```

```
END_IF;
```

### 2.6.10 RadToDeg (弧度转角度)

本指令用于将弧度转换为角度。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
RadToDeg	弧度转角度	FUN		Out:= RadToDeg(In1);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	弧度	输入	待转换的弧度数值	变量的数据类型决定
Out	角度	输出	转换后的角度值	变量的数据类型决定

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

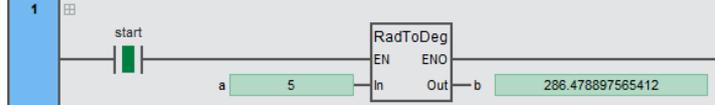
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将输入变量“In”中的弧度(rad)值转换为角度(°)后输出至“Out”。转换公式  $Out = In * 180 / \pi$ 。
- 示例程序如下所示：

#### 梯形图(LD):

类别	名称	分配到	数据类型	在线值	准备值	注释
1 VAR	a		REAL	5		
2 VAR	b		LREAL	286.478897565412		
3 VAR	start		BOOL	TRUE		

#### 结构化文本(ST):

```
IF start THEN
    b:= RadToDeg(a);
END_IF;
```

## 2.6.11 DegToRad (角度转弧度)

本指令用于将角度转换为弧度。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
DegToRad	角度转弧度	FUN		Out:= DegToRad (In1);

### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	角度	输入	待转换的角度数值	变量的数据类型决定
Out	弧度	输出	转换后的弧度值	变量的数据类型决定

	布尔		位串					整数						实数		时间, 日期			字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		<input type="radio"/>																		
Out														<input type="radio"/>	<input type="radio"/>					

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

- 本指令用于将输入变量“In”中的角度(°)值转换为弧度(rad)后输出至“Out”。

转换公式  $Out = (In / 180) * \pi$ 。

- 示例程序如下所示：

梯形图(LD):

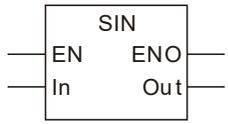
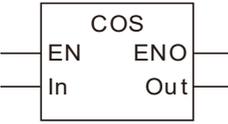
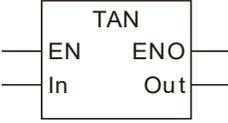


结构化文本(ST):

```
IF start THEN
    b:= DegToRad (a);
END_IF;
```

## 2.6.12 SIN/COS/TAN (三角函数)

对输入变量“ln”进行三角函数计算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SIN	正弦	FUN		Out:= SIN(ln);
COS	余弦	FUN		Out:= COS(ln);
TAN	正切	FUN		Out:= TAN(ln);

### 输入、输出变量说明及数据类型

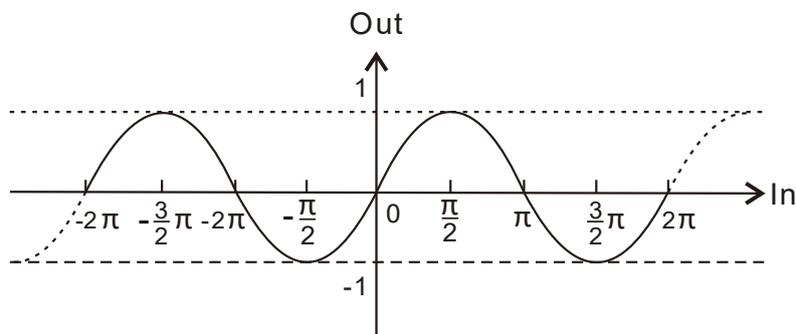
引脚	名称	输入/输出	描述	参数取值范围
In	输入值	输入	待转换的数值，单位：弧度	变量的数据类型决定
Out	输出值	输出	通过三角函数运算后的结果	变量的数据类型决定

	布尔	位串					整数						实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		<input type="radio"/>																		
Out														<input type="radio"/>						

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

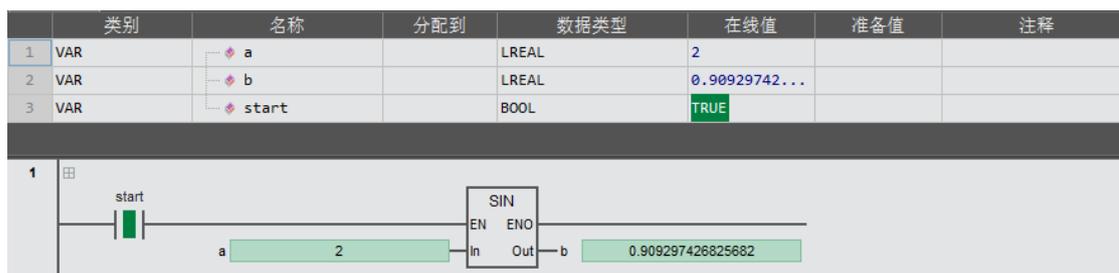
### 功能说明

- SIN 指令用于对“ln”的值进行正弦运算，结果输出至“Out”，“ln”的单位为弧度。



- 示例程序如下所示：

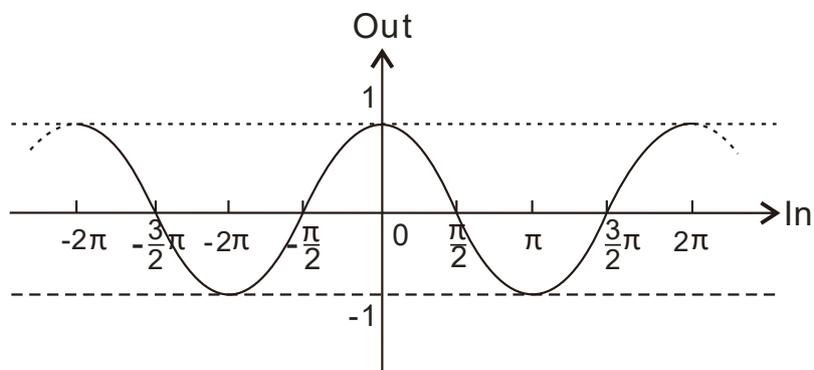
#### 梯形图(LD):



#### 结构化文本(ST):

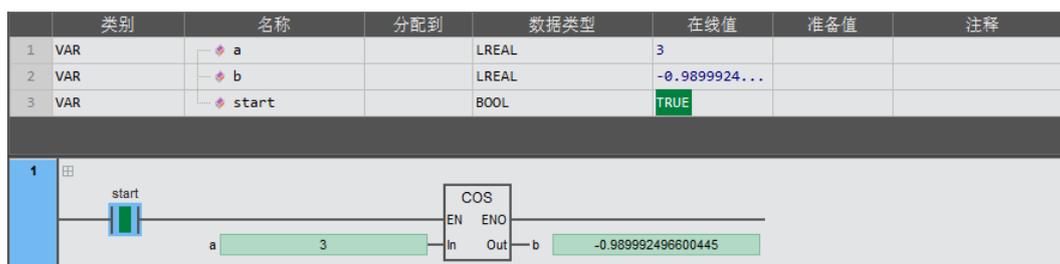
```
IF start THEN
    b:= SIN(a);
END_IF;
```

- COS 指令用于对 “In” 的值进行余弦运算，结果输出至 “Out”，“In” 的单位为弧度。



- 示例程序如下所示：

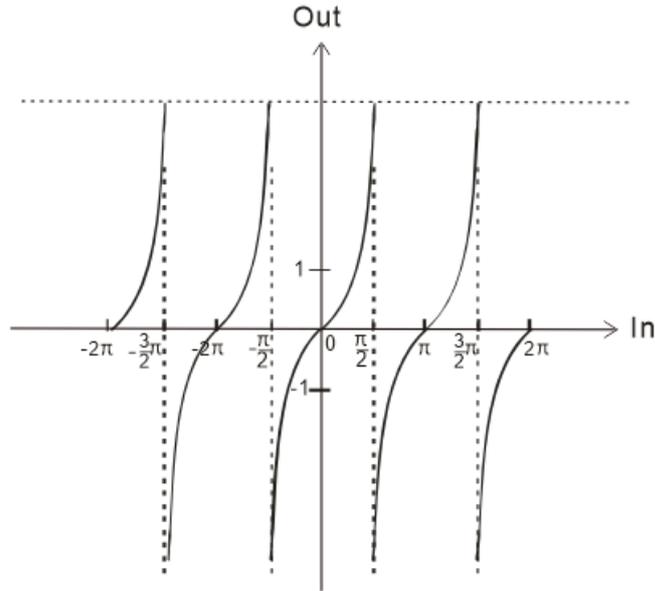
#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
    b:= COS(a);
END_IF;
```

- TAN 指令用于对 “In” 的值进行正切运算，结果输出至 “Out”，“In” 的单位为弧度。



- 示例程序如下所

示：

**梯形图(LD):**

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		LREAL	2		
2	VAR	b		LREAL	-2.18503986326152		
3	VAR	start		BOOL	TRUE		

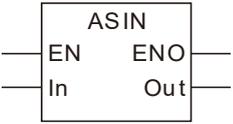
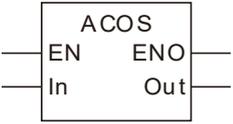
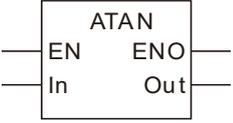
**结构化文本(ST):**

```

IF start THEN
    b:= TAN(a);
END_IF;
    
```

## 2.6.13 ASIN/ACOS/ATAN (反三角函数)

用于对输入变量进行反三角函数计算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
ASIN	反正弦	FUN		Out:=ASIN(In);
ACOS	反余弦	FUN		Out:=ACOS(In);
ATAN	反正切	FUN		Out:=ATAN(In);

### 输入、输出变量说明及数据类型

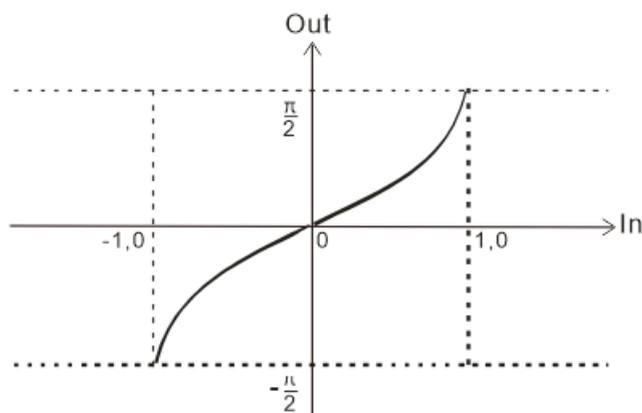
引脚	名称	输入/输出	描述	参数取值范围
In	输入值	输入	待转换的数值	变量的数据类型决定
Out	输出值	输出	通过反三角函数运算后的结果，单位：弧度。	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

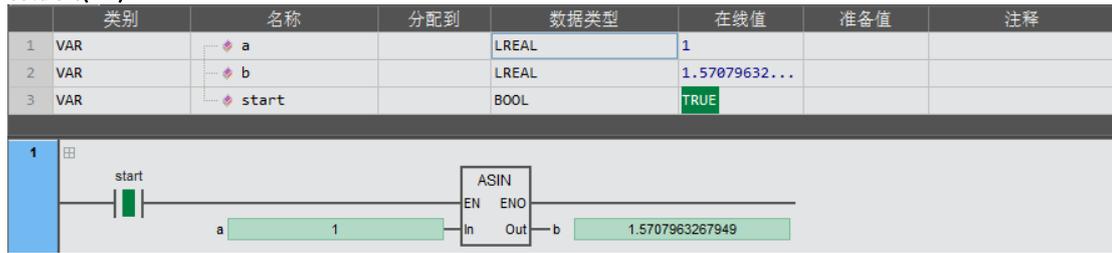
### 功能说明

- ASIN 指令用于对“In”的值进行反正弦运算，结果输出至“Out”，“Out”的单位为弧度。



- 示例程序如下所示：

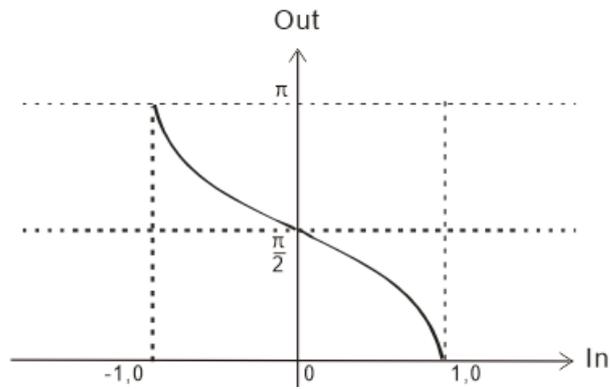
**梯形图(LD):**



**结构化文本(ST):**

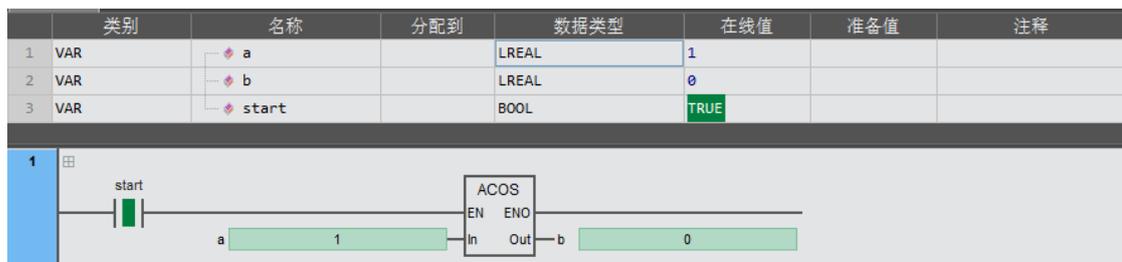
```
IF start THEN
    b:=ASIN (a);
END_IF;
```

- ACOS 指令用于对 “In” 的值进行反余弦运算，结果输出至 “Out”，“Out” 的单位为弧度。



- 示例程序如下所示：

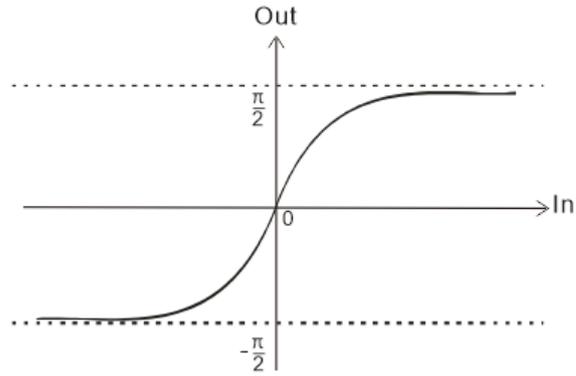
**梯形图(LD):**



**结构化文本(ST):**

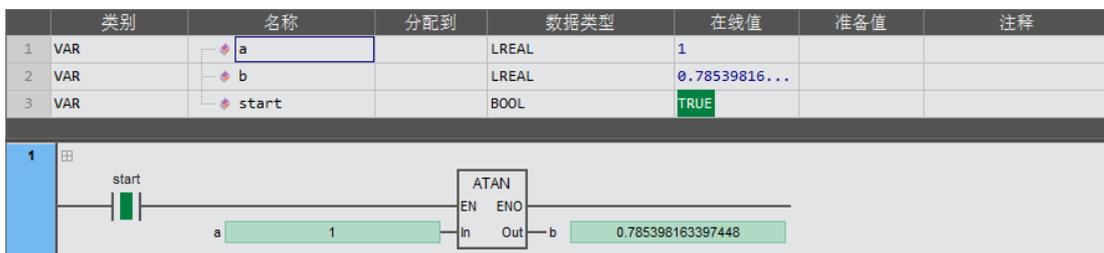
```
IF start THEN
    b:= ACOS(a);
END_IF;
```

- ATAN 指令用于对“In”的值进行反正切运算，结果输出至“Out”，“Out”的单位为弧度。



- 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

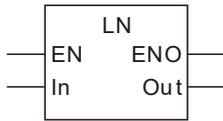
```

IF start THEN
    b:= ATAN(a);
END_IF;

```

### 2.6.14 LN (自然对数)

本指令用于对输入变量“ln”进行自然对数计算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
LN	自然对数	FUN		Out:= LN(ln);

■ 输入、输出变量说明及数据类型

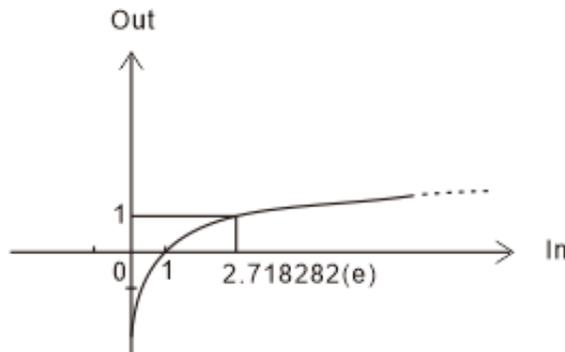
引脚	名称	输入/输出	描述	参数取值范围
In	输入值	输入	待转换的数值	变量的数据类型决定
Out	对数	输出	自然对数	变量的数据类型决定

	布尔		位串					整数					实数		时间, 日期				字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		<input type="radio"/>																		
Out														<input type="radio"/>						

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

■ 功能说明

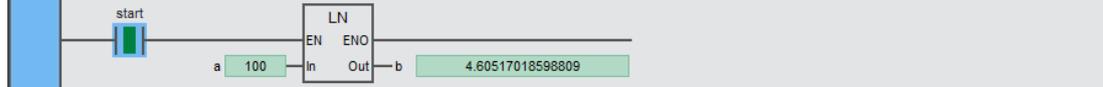
- LN 指令用于对“ln”的值进行自然对数运算，即计算以 e (e=2.718282) 为底“ln”的对数，结果输出至“Out”。



- 示例程序如下所示：

梯形图(LD):

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		INT	100		
2	VAR	b		LREAL	4.60517018...		
3	VAR	start		BOOL	TRUE		

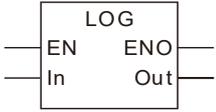
  


结构化文本(ST):

```
IF start THEN
    b:= LN(a);
END_IF;
```

## 2.6.15 LOG (自然对数)

本指令用于对输入变量“ln”进行常用对数计算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
LOG	常用对数	FUN		Out:= LOG(ln);

### 输入、输出变量说明及数据类型

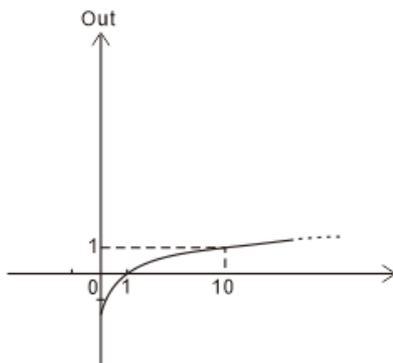
引脚	名称	输入/输出	描述	参数取值范围
In	输入值	输入	待转换的数值	变量的数据类型决定
Out	输出值	输出	常用对数	变量的数据类型决定

	布尔		位串					整数						实数		时间, 日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		<input type="radio"/>																		
Out														<input type="radio"/>						

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

- LOG 指令用于对“ln”的值进行对数运算，即计算以 10 为底“ln”的对数，结果输出至“Out”。



- 示例程序如下所示：

#### 梯形图(LD):

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		INT	100		
2	VAR	b		LREAL	2		
3	VAR	start		BOOL	TRUE		

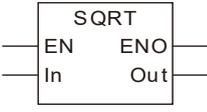
  

#### 结构化文本(ST):

```
IF start THEN
    b:= LOG (a);
END_IF;
```

### 2.6.16 SQRT (求平方根)

本指令用于对输入变量“ln”进行平方根计算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SQRT	平方根计算	FUN		Out:= SQRT(ln);

■ 输入、输出变量说明及数据类型

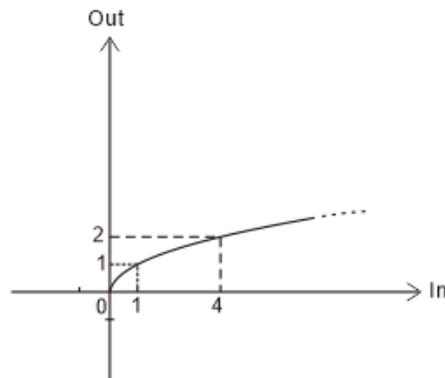
引脚	名称	输入/输出	描述	参数取值范围
In	运算值	输入	运算值	变量的数据类型决定
Out	平方根	输出	平方根	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

■ 功能说明

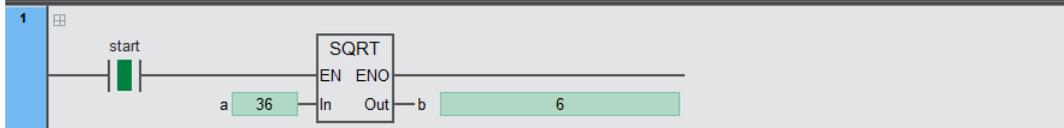
- 本指令用于对输入变量“ln”进行平方根计算，结果输出至“Out”。



- 示例程序如下所示：

梯形图(LD):

	类别	名称	分配到	数据类型	在线值	准备值	注释
46	VAR	a		INT	36		
47	VAR	b		LREAL	6		
48	VAR	start		BOOL	TRUE		

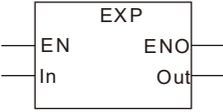
  


结构化文本(ST):

```
IF start THEN
    b:= SQRT(a);
END_IF;
```

## 2.6.17 EXP (自然指数)

本指令用于计算以e为底，“In”为指数的幂运算，并将结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
EXP	自然指数	FUN		Out:= EXP(In);

### 输入、输出变量说明及数据类型

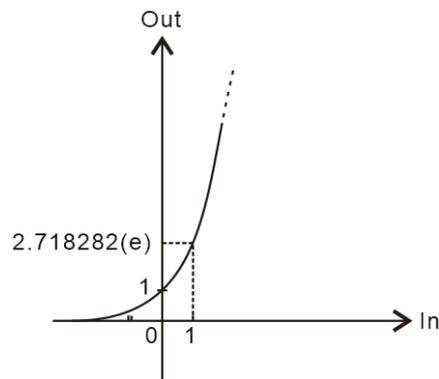
引脚	名称	输入/输出	描述	参数取值范围
In	指数	输入	指数	变量的数据类型决定
Out	运算结果	输出	e为底，“In”为指数的结果	变量的数据类型决定，非负数

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

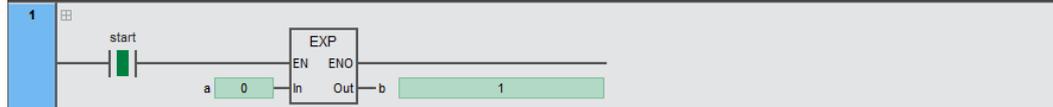
- 本指令用于本指令用于计算以 e 为底，“In”为指数的幂运算，并将结果输出至“Out”。



- 示例程序如下所示：

#### 梯形图(LD):

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		INT	0		
2	VAR	b		LREAL	1		
3	VAR	start		BOOL	TRUE		

#### 结构化文本(ST):

```
IF start THEN
    b:=EXP (a);
END_IF;
```

### 2.6.18 EXPT (幂指数)

本指令用于计算以 “In” 为底，“Pwr” 为指数的幂运算，并将结果输出至 “Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
EXPT	幂指数	FUN		Out:= EXPT(In);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	底	输入	底数	变量的数据类型决定
Pwr	指数		指数	变量的数据类型决定
Out	运算结果	输出	以 “In” 为底，“Pwr” 为指数的计算结果	变量的数据类型决定

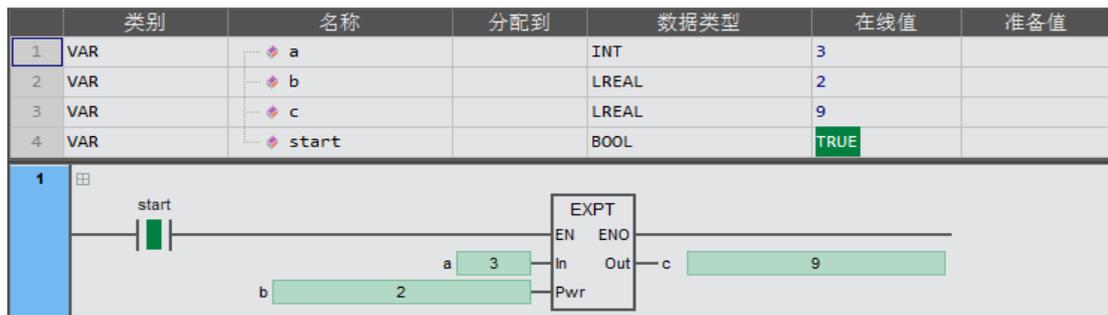
	布尔		位串					整数						实数		时间, 日期			字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		○	○	○	○	○	○	○	○	○	○	○	○	○	○					
Out														○	○					

\*说明：上表中的 “○” 表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于本指令用于计算以 “In” 为底，“Pwr” 为指数的幂运算，并将结果输出至 “Out”。
- 示例程序如下所示：

##### 梯形图(LD):



##### 结构化文本(ST):

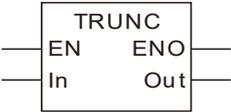
```
IF start THEN
    c:=EXPT (In:=a,Pwr:=b);
END_IF;
```

## 2.6.19 TRUNC/FLOOR (浮点数取整数部分)

TRUNC用于取出输入变量“ln”的整数部分,舍弃小数部分,结果输出至“Out”。

FLOOR用于取出输入变量“ln”的整数部分(正数和负数时区别对待),舍弃小数部分,结果输出至“Out”。

所属库: Standard

指令	名称	FB/FUN	梯形图样式	ST样式
TRUNC	浮点数取整	FUN		Out:= TRUNC(ln);
FLOOR	浮点数取整后运算	FUN		Out:= FLOOR(ln);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	输入值	输入	输入值	变量的数据类型决定
Out	运算结果	输出	运算后的整数部分	变量的数据类型决定

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

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。输出变量的数据类型只能为LINT。

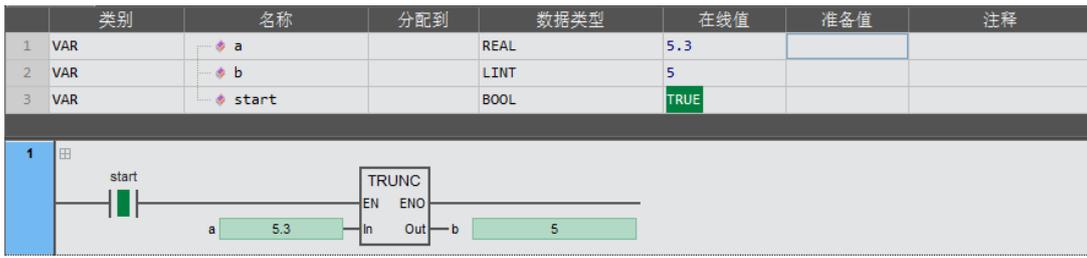
### ■ 功能说明

- TRUNC 用于取出输入变量“ln”的整数部分,舍弃小数部分,结果输出至“Out”。
- FLOOR 用于取出输入变量“ln”的整数部分,舍弃小数部分,结果输出至“Out”。该指令的输入整数部分为正数时,“Out”为对应的正数;该指令的输入整数部分为负数时,“Out”为对应的整数减 1。

“ln”的值	TRUNC指令“Out”的值	FLOOR指令“Out”的值
5.3	5	5
5	5	5
-5.3	-5	-6
-5	-5	-5

- 示例程序如下所示：

**梯形图(LD):**

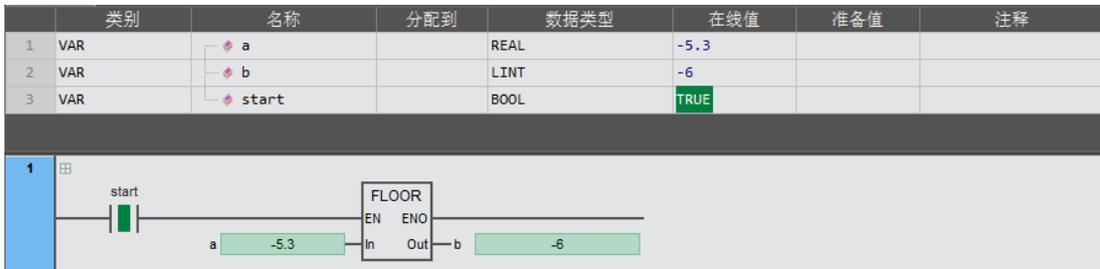


**结构化文本(ST):**

```
IF start THEN
    b:=TRUNC (a);
END_IF;
```

- 示例程序如下所示：

**梯形图(LD):**



**结构化文本(ST):**

```
IF start THEN
    b:=FLOOR (a);
END_IF;
```

## 2.6.20 FRACTION (浮点数取小数部分)

本指令用于取出输入变量“In”的小数部分，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
FRACTION	浮点数取小数	FUN		Out:= FRACTION(In);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	输入值	输入	输入值	变量的数据类型决定
Out	运算结果	输出	运算后的小数部分	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。输出变量的数据类型只能为LREAL。

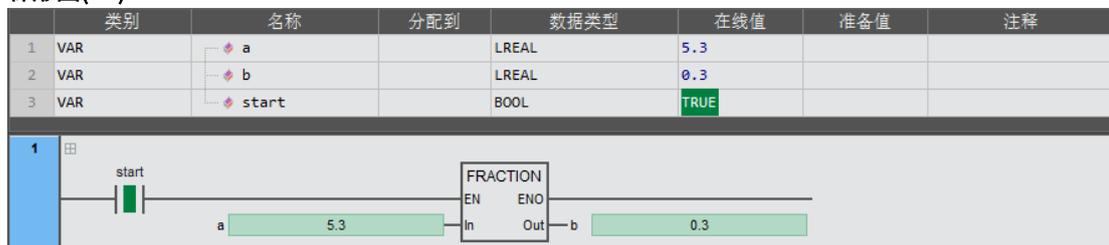
### ■ 功能说明

- 本指令用于取出输入变量“In”的小数部分，结果输出至“Out”。“Out”的符号和“In”的符号一致，如“In”的值为正数时，“Out”的值为正数；“In”的值为负数时，“Out”的值为负数。

“In” 的值	“Out” 的值
5.3	0.3
-5.3	-0.3

- 示例程序如下所示：

#### 梯形图(LD):

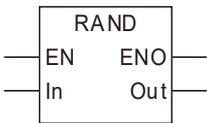


#### 结构化文本(ST):

```
IF start THEN
    b:=FRACTION(a);
END_IF;
```

### 2.6.21 RAND (随机数)

本指令用于产生1个随机数。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
RAND	随机数	FUN		Out:= RAND(In);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	保留	输入	保留	变量的数据类型决定
Out	随机数	输出	随机数	变量的数据类型决定

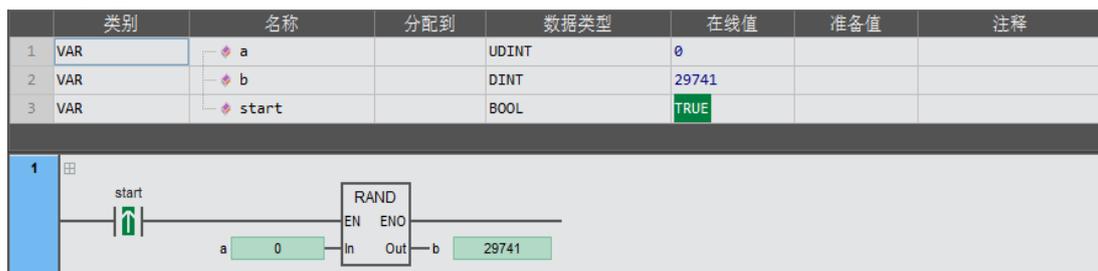
	布尔		位串				整数						实数		时间, 日期			字符串			
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
In								○													
Out												○									

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于产生 1 个 0 - 32767 范围的随机数。
- 该指令的输入变量的值对输出结果的值没有影响。
- 示例程序如下所示：

梯形图(LD):



结构化文本(ST):

```
IF start THEN
    b:=RAND(a);
END_IF;
```

## 2.7 逻辑运算指令

### 2.7.1 AND (与)

本指令用于多个变量或常量进行与运算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
AND	与	FUN		Out:=In1 and In2 and ...and InN; Out:=In1& In2 & ...& InN;

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	运算对象	输入	运算对象	变量的数据类型决定
In2至InN	运算对象		运算对象，程序编写时，梯形图时可通过编程软件增加或减少运算对象，即N=2~8	变量的数据类型决定
Out	运算结果	输出	运算结果	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将两个或多个变量或常量按对应的位进行与运算，结果输出至 Out，即  $Out = In1 \& In2 \& \dots \& InN$ 。按位与的运算规则为：输入变量的对应位均为 TRUE 时，输出变量的对应位为 TRUE，否则为 FALSE。

例如 16#37 AND 16#42 ,结果为 16#02。

- 输入变量类型为位串、整数时，本指令允许输入变量“In1”~“InN”为不同数据类型的变量。当 In1~InN 数据类型不同时，使用包含 In1~InN 取值范围的数据类型进行运算，例如 In1 的数据类型为 BYTE，In2 的数据类型为 DWORD，则 Out 的数据类型为 DWORD；运算时将 In1 由 BYTE 转换成 DWORD（补齐位全为 0）后再与 In2 按位作与运算。
- 如果输入变量的数据类型为 BOOL 时，要求所有输入和输出变量的数据类型均为 BOOL，否则软件编译时将报错。





## 2.7.2 OR (或)

本指令用于多个变量或常量按对应的位进行或运算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
OR	或	FUN		Out:=In1 OR In2 OR ...OR InN;

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	运算对象	输入	运算对象	变量的数据类型决定
In2至InN	运算对象		运算对象，程序编写时，梯形图时可通过编程软件增加或减少运算对象，即N = 2 ~ 8	变量的数据类型决定
Out	运算结果	输出	运算结果	变量的数据类型决定

	布尔	位串					整数							实数		时间, 日期			字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1至InN	<input type="radio"/>																			
Out	<input type="radio"/>																			

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于将两个或多个变量或常量按位或运算，结果输出至 Out，即 Out = In1 OR In2 OR...OR InN。运算规则为输入变量的对应位只要有 1 个为 TRUE 时，输出变量的对应位为 TRUE，输入变量的对应位都为 FALSE 时，输出变量的对应位为 FALSE。例如 16#02 OR 16#03,结果为 16#03。
- 当输入变量类型为位串、整数时，允许输入变量“In1”~“InN”为不同数据类型的变量。当 In1~InN 数据类型不同时，以包含 In1~InN 所有取值范围的数据类型进行运算。例如 In1 数据类型为 BYTE，In2 数据类型为 DWORD，则 Out 的数据类型为 DWORD；运算时将 In1 由 BYTE 提升至 DWORD（补齐位全为 0）后再与 In2 按位作或运算。
- 当输入变量的数据类型为 BOOL 时，则要求所有输入和输出变量的数据类型均为 BOOL，否则软件编译报错。
- 示例程序 1 和示意图如下所示：

#### 梯形图(LD)：

下图所示为将三个WORD类型的变量进行相或的示例,。“In1”的值为16#9009，“In2”的值为16#A00A，“In3”的值为16#C00C,该指令执行后，Out的值为16#F00F。执行时为“In1”和“In2”相或后的结果再和“In3”相或，最终结果输出至“Out”。

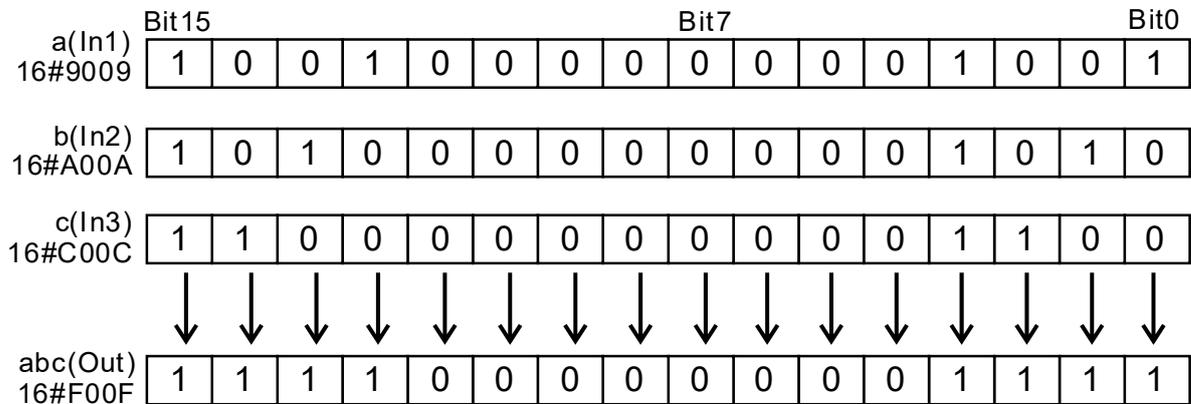
	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		WORD	16#9009		
2	VAR	b		WORD	16#A00A		
3	VAR	c		WORD	16#C00C		
4	VAR	d		WORD	16#F00F		
5	VAR	start		BOOL	TRUE		

**结构化文本(ST):**

IF start THEN

d:=a OR b OR c;

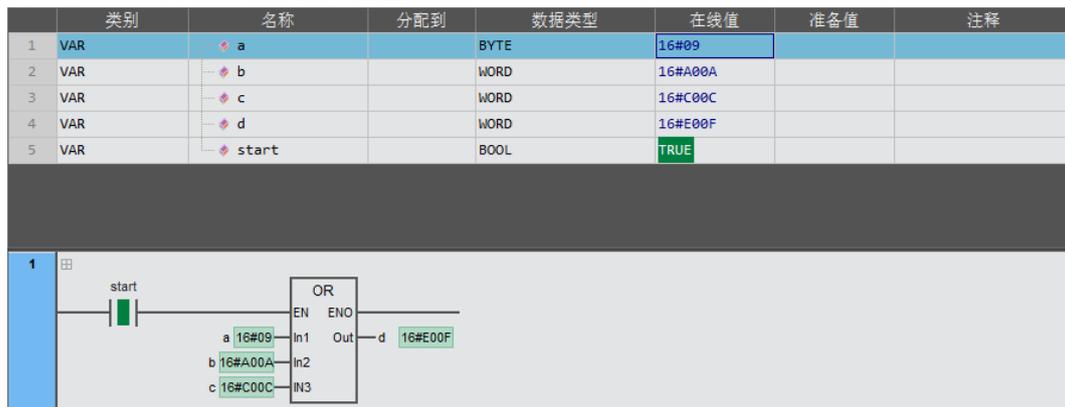
END\_IF;

**示意图:**

- 示例程序 2 和示意图如下所示:

**梯形图(LD):**

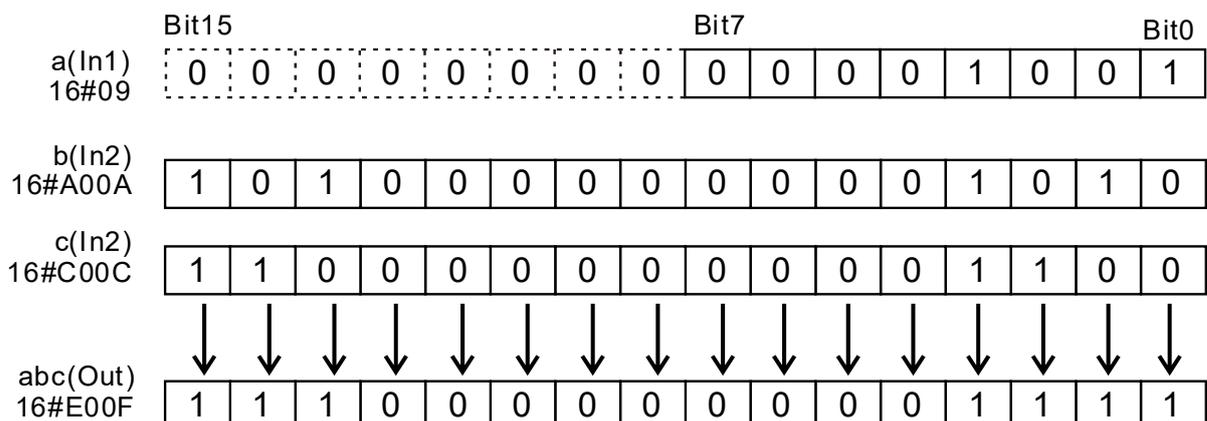
下图所示为1个BYTE类型的变量和两个WORD类型变量进行相或的示例。“In1”的值为16#9，“In2”的值为16#A00A，“In2”的值为16#C00C,该指令执行后，Out的值为16#E00F。执行时为“In1”和“In2”相或后的结果再和“In3”相或，最终结果输出至“Out”。

**结构化文本(ST):**

IF start THEN

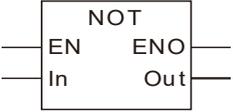
d:=a OR b OR c;

END\_IF;

**示意图:**

### 2.7.3 NOT (取反)

本指令用将变量或常量进行按位取反运算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
NOT	取反	FUN		Out:=NOT(In);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	运算对象	输入	运算对象	变量的数据类型决定
Out	运算结果	输出	运算结果	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将变量或常量按位取反运算，结果输出至 Out，即 Out = NOT(In)。运算规则为将输入变量为 TRUE 的位转成 FALSE，将输入变量 FALSE 的位转成 TRUE,并把最终结果输出至 Out。
- “Out”数据类型宽度必须大于等于“In”数据类型的宽度，否则软件编译报错。以“Out”的数据类型进行运算。例如 In 数据类型为 BYTE，Out 的数据类型为 DWORD，则 Out 的数据类型为 DWORD。运算时先进行取反运算然后将 In 由 BYTE 提升至 DWORD（补齐位全为 0）。
- 当输入变量的数据类型为 BOOL 时，则要求所有输入和输出变量的数据类型均为 BOOL，否则软件编译报错。
- 示例程序和示意图如下所示：

#### 梯形图(LD)：

下图所示为1个WORD类型的变量进行取反的示例。“In”的值为16#5555，该指令执行后，Out的值为16#AAAA。

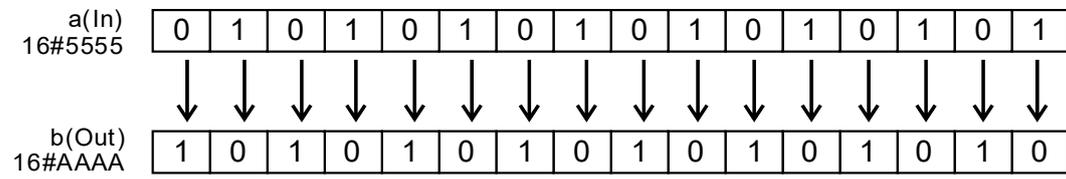
	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		WORD	16#5555		
2	VAR	b		WORD	16#AAAA		
3	VAR	start		BOOL	TRUE		

**结构化文本(ST):**

IF start THEN

b:=NOT a;

END\_IF;

**示意图:**

## 2.7.4 XOR (异或)

本指令用于多个变量或常量按对应的位异或运算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
XOR	异或	FUN		Out = In1 XOR In2 XOR...XOR InN;

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	运算对象	输入	运算对象	变量的数据类型决定
In2至InN	运算对象		运算对象，程序编写时，梯形图时可通过编程软件增加或减少运算对象，即N = 2 ~ 8	变量的数据类型决定
Out	运算结果	输出	运算结果	变量的数据类型决定

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

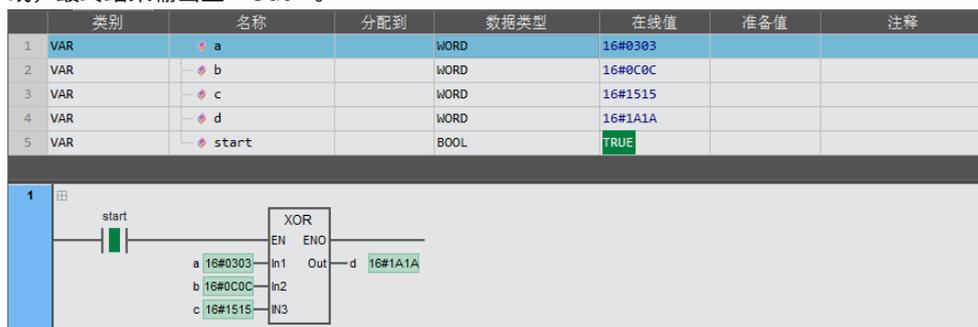
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将两个或多个变量或常量按对应的位异或运算，结果输出至 Out，即 Out = In1 XOR In2 XOR...XOR InN。运算规则为如果输入变量对应位的值不同，输出变量的对应位为 TRUE，如果输入变量对应位的值相同，输出变量的对应位为 FALSE。
- 当输入变量类型为位串、整数时，允许输入变量“In1”~“InN”数据类型不同。以包含 In1~InN 所有参数取值范围的数据类型进行运算。例如 In1 数据类型为 WORD，In2 数据类型为 DWORD，则 Out 的数据类型为 DWORD；运算时将 In1 由 WORD 提升至 DWORD（补齐位全为 0）后再与 In2 按位作异或运算。
- 当输入变量的数据类型为 BOOL 时，要求所有输入和输出变量的数据类型均为 BOOL，否则软件编译报错。
- 示例程序 1 和示意图如下所示：

#### 梯形图(LD)：

下图所示为将三个WORD类型的变量进行异或的示例。“In1”的值为16#0303，“In2”的值为16#0C0C，“In3”的值为16#1515,该指令执行后，Out的值为16#1A1A。执行时为“In1”和“In2”异或后的结果再和“In3”异或，最终结果输出至“Out”。

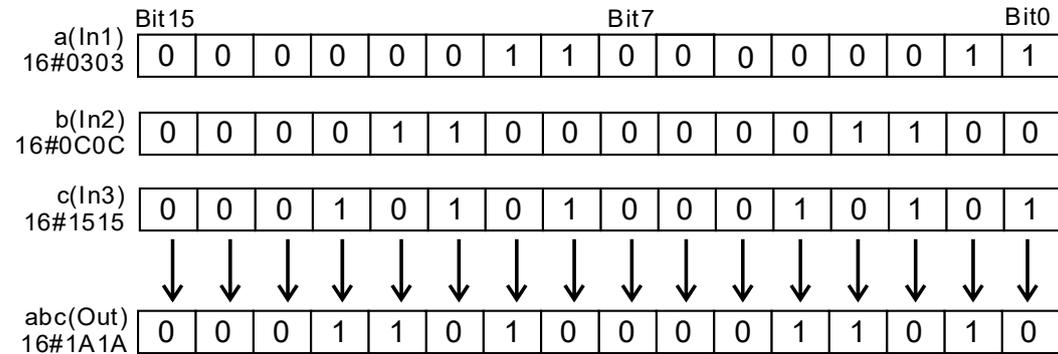


**结构化文本(ST):**

```
IF start THEN
```

```
    d:=a XOR b XOR c;
```

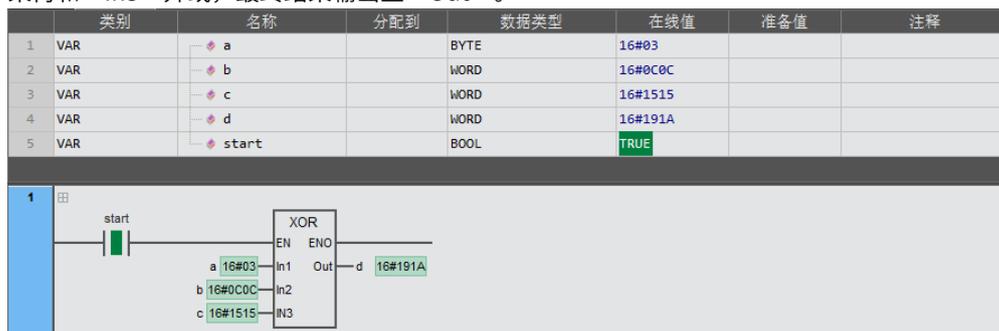
```
END_IF;
```

**示意图:**

- 示例程序 2 和示意图如下所示:

**梯形图(LD):**

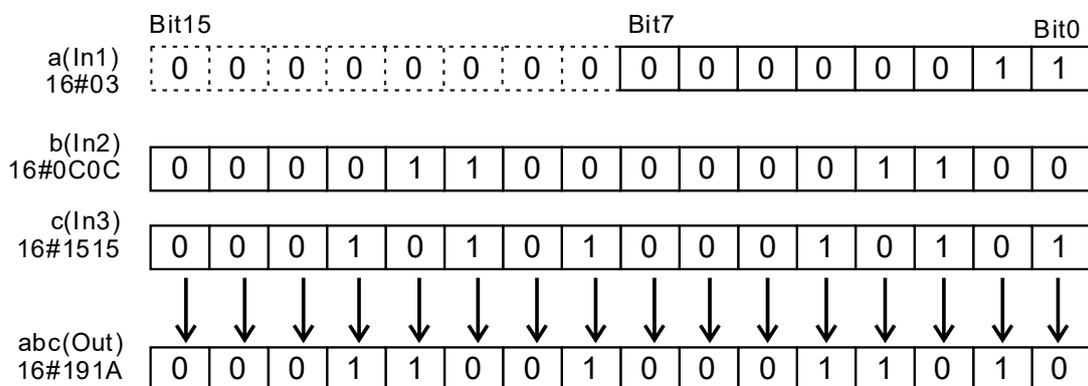
下图所示为1个BYTE类型的变量和两个WORD类型变量进行异或的示例。“In1”的值为16#03，“In2”的值为16#0C0C，“In3”的值为16#1515,该指令执行后，Out的值为16#191A。执行时为“In1”和“In2”异或后的结果再和“In3”异或，最终结果输出至“Out”。

**结构化文本(ST):**

```
IF start THEN
```

```
    d:=a XOR b XOR c;
```

```
END_IF;
```

**示意图:**

## 2.7.5 XORN (同或)

本指令用于多个变量或常量按对应的位同或运算，结果输出至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
XORN	同或	FUN		Out = In1 XORN In2 XORN...XORN InN;

### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	运算对象	输入	运算对象	变量的数据类型决定
In2至InN	运算对象		运算对象，程序编写时，梯形图时可通过编程软件增加或减少运算对象，即N = 2 ~ 8	变量的数据类型决定
Out	运算结果	输出	运算结果	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

- 本指令用于将两个或多个变量或常量按对应的位同或运算，结果输出至 Out，即 Out = In1 XORN In2 XORN...XORN InN。运算规则为当输入变量对应位全部相同，输出变量的对应位为 TRUE，当输入变量对应位不同时，输出变量的对应位为 FALSE。
- 输入变量类型为位串、整数时，允许输入变量“in1”~“inN”数据类型不同。当 In1~InN 数据类型不同时，以包含 In1~InN 取值范围的数据类型进行运算。例如 In1 数据类型为 WORD，In2 数据类型为 DWORD，则 Out 的数据类型为 DWORD；运算时将 In1 由 WORD 转换成 DWORD（补齐位全为 0）后再与 In2 按位作同或运算。
- 当输入变量的数据类型为 BOOL 时，则要求所有输入和输出变量的数据类型均为 BOOL，否则软件编译报错。
- 示例程序 1 和示意图如下所示：

#### 梯形图(LD)：

下图所示为将三个WORD类型的变量进行同或的示例。“In1”的值为16#0303，“In2”的值为16#0C0C，“In3”的值为16#1515,该指令执行后，Out的值为16#1A1A。执行时为“In1”和“In2”同或后的结果再和“In3”同或，最终结果输出至“Out”。

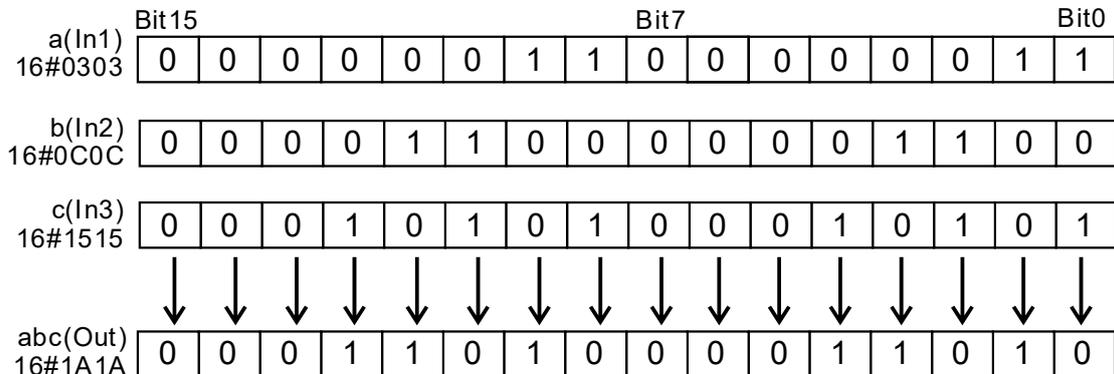
	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	a		WORD	16#0303		
2	VAR	b		WORD	16#0C0C		
3	VAR	c		WORD	16#1515		
4	VAR	d		WORD	16#1A1A		
5	VAR	start		BOOL	TRUE		

**结构化文本(ST):**

IF start THEN

d:=a XORN b XORN c;

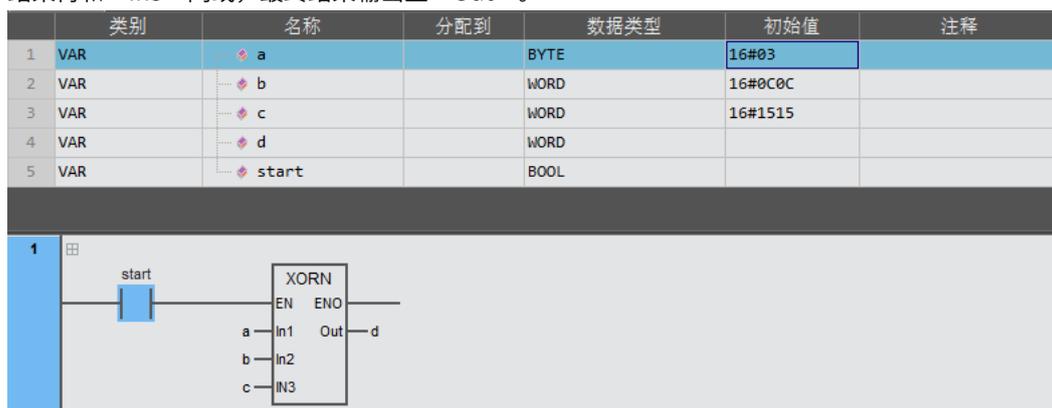
END\_IF;

**示意图:**

- 示例程序 2 和示意图如下所示:

**梯形图(LD):**

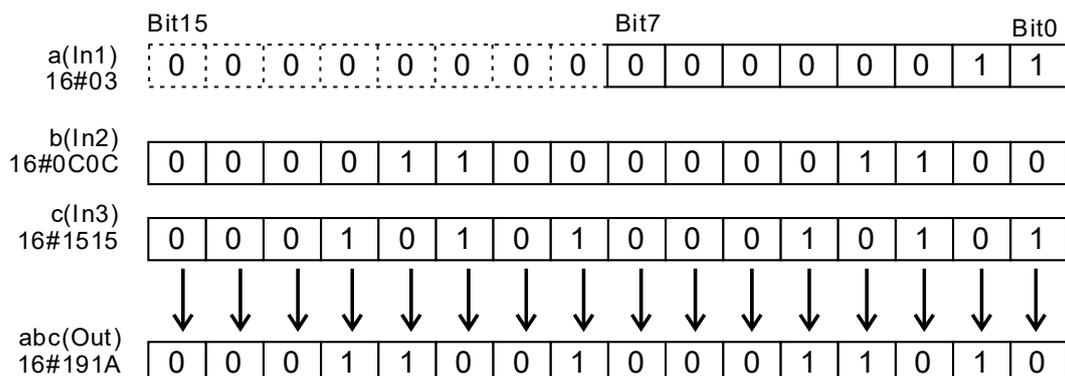
下图所示为1个BYTE类型的变量和两个WORD类型变量进行同或的示例。“In1”的值为16#03，“In2”的值为16#0C0C，“In3”的值为16#1515，该指令执行后，Out的值为16#191A。执行时为“In1”和“In2”同或后的结果再和“In3”同或，最终结果输出至“Out”。

**结构化文本(ST):**

IF start THEN

d:=a XORN b XORN c;

END\_IF;

**示意图:**

## 2.8 数据移位

### 2.8.1 SHL (向左移位) / SHR (向右移位)

SHL(向左移位): 用于将输入变量或常量中的所有位按指定的位数从右向左依次移位, 结果输出至“Out”。

SHR(向右移位): 用于将输入变量或常量中的所有位按指定的位数从左向右依次移位, 结果输出至“Out”。

所属库: Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SHL	向左移位	FUN		Out:=SHL(In,Num);
SHR	向右移位	FUN		Out:=SHR(In,Num);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	移位对象	输入	移位对象	变量的数据类型决定
Num	移位数量		移位数量	变量的数据类型决定
Out	移位结果	输出	移位结果	变量的数据类型决定

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

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

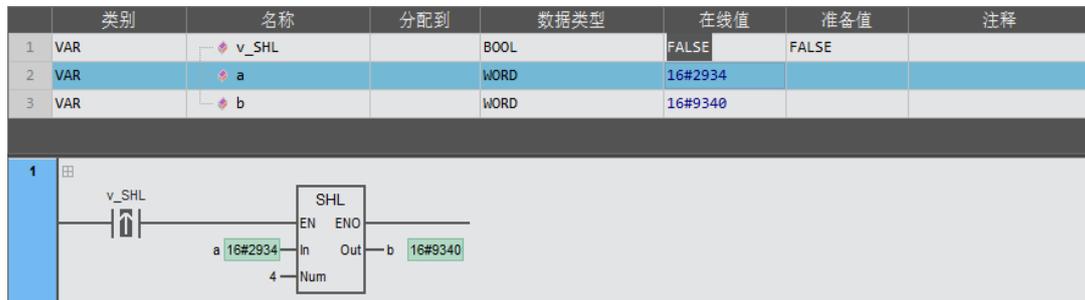
#### ■ 功能说明

- SHL (向左移位): 用于将输入变量或常量中的所有位按 Num 指定的位数从右向左(从低位到高位)依次移位, 最右侧 Num 位的值全部变为 0, 移位后结果输出至“Out”。
- SHR (向右移位): 用于将输入变量或常量中的所有位按 Num 指定的位数从左向右(从高位到低位)依次移位, 最左侧 Num 位的值全部变为 0, 移位后结果输出至“Out”。

- SHL 示例程序和示意图如下所示：

#### 梯形图(LD):

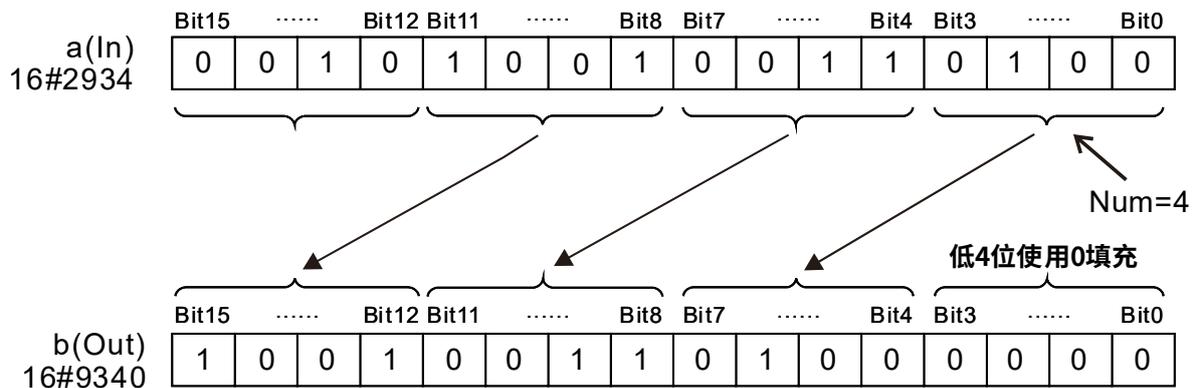
下图所示为将WORD类型的变量进行向左移位的示例。v\_SHL变量从FALSE变为TRUE时，SHL指令执行1次。“In1”的值为16#2934，“Num”的值为4，该指令执行后，Out的值为16#9340。



#### 结构化文本(ST):

```
IF EDGEPOS(v_SHL) THEN
    b:=SHL(In:=a ,Num:=4);
END_IF;
```

#### 示意图:



- SHR 示例程序和示意图如下所示：

#### 梯形图(LD):

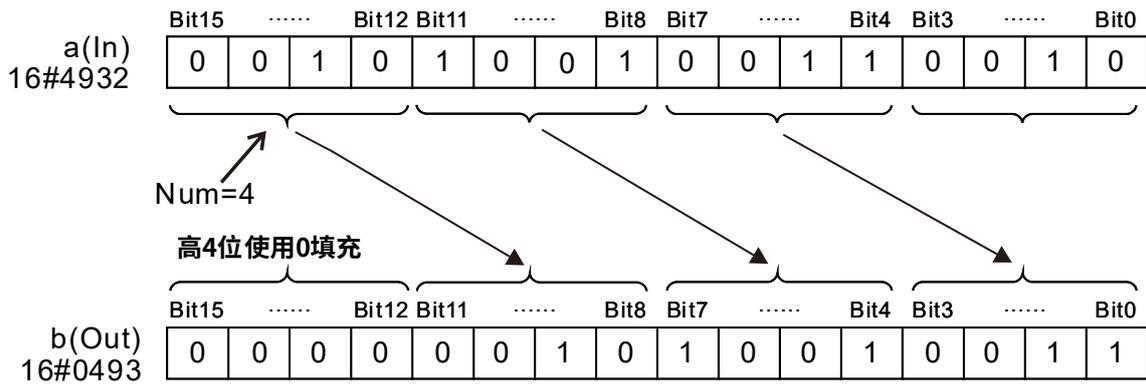
下图所示为将WORD类型的变量进行向右移位的示例。v\_SHR变量从FALSE变为TRUE时，SHR指令执行1次。“In1”的值为16#4932，“Num”的值为4，该指令执行后，Out的值为16#0493。



#### 结构化文本(ST):

```
IF EDGEPOS(v_SHR) THEN
    b:=SHR(In:=a ,Num:=4);
END_IF;
```

示意图:



## 2.8.2 ROL (向左循环移位) / ROR (向右循环移位)

ROL(向左循环移位): 用于将输入变量或常量中的所有位按指定的位数从右向左依次循环移位, 结果输出至“Out”。

ROR(向右循环移位): 用于将输入变量或常量中的所有位按指定的位数从左向右依次循环移位, 结果输出至“Out”。

所属库: Standard

指令	名称	FB/FUN	梯形图样式	ST样式
ROL	向左循环移位	FUN		Out:=ROL(In,Num);
ROR	向右循环移位	FUN		Out:=ROR(In,Num);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	移位对象	输入	移位对象	变量的数据类型决定
Num	移位数量		移位数量	变量的数据类型决定
Out	移位结果	输出	移位结果	变量的数据类型决定

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

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

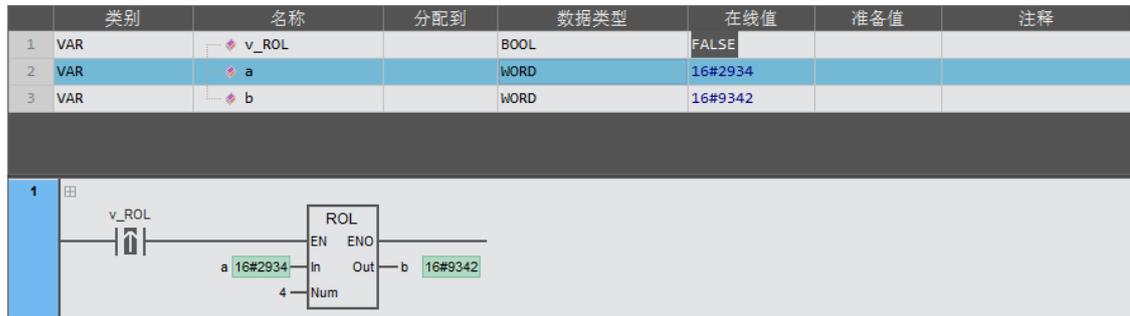
### ■ 功能说明

- ROL (向左循环移位): 用于将输入变量或常量中的所有位按 Num 指定的位数从右向左(从低位到高位)依次移位, 最右侧 Num 位的值变为最左侧 Num 位的值, 移位后结果输出至“Out”。
- ROR (向右循环移位): 用于将输入变量或常量中的所有位按 Num 指定的位数从左向右(从高位到低位)依次旋转移位, 最左侧 Num 位的值变为最右侧 Num 位的值, 移位后结果输出至“Out”。

- ROL 示例程序和示意图如下所示：

**梯形图(LD):**

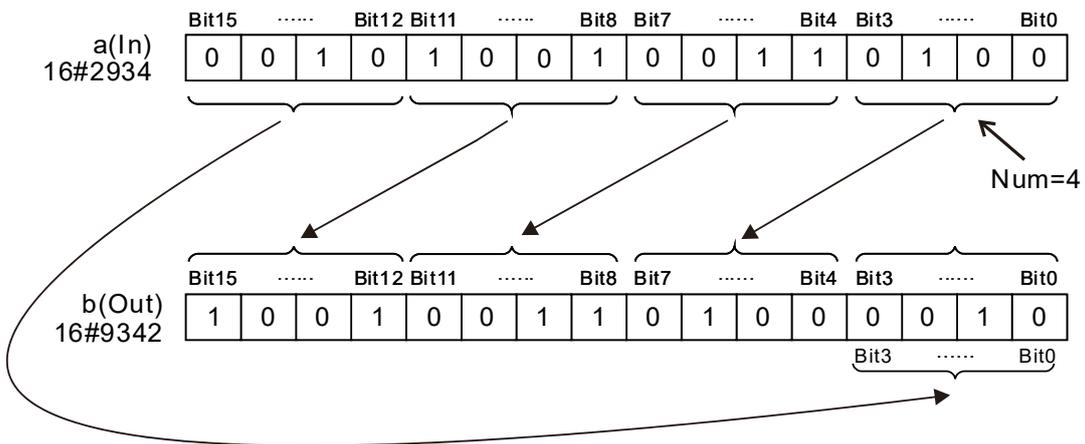
下图所示为将WORD类型的变量进行向左循环移位的示例。v\_ROL变量从FALSE变为TRUE时，ROL指令执行1次。“In1”的值为16#2934，“Num”的值为4,该指令执行后，Out的值为16#9342。



**结构化文本(ST):**

```
IF EDGEPOS(v_ROL) THEN
    b:=ROL (In:=a ,Num:=4);
END_IF;
```

**示意图:**



- ROR 示例程序和示意图如下所示：

**梯形图(LD):**

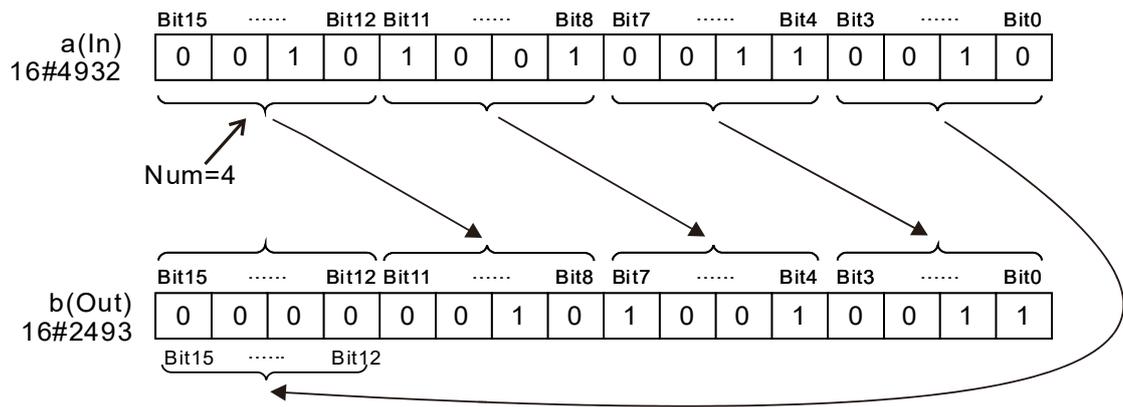
下图所示为将WORD类型的变量进行向右循环移位的示例。v\_ROR变量从FALSE变为TRUE时，ROR指令执行1次。“In1”的值为16#4932，“Num”的值为4,Out的值为16#2493。



**结构化文本(ST):**

```
IF EDGEPOS(v_ROR) THEN
    b:=ROR (In:=a ,Num:=4);
END_IF;
```

示意图:

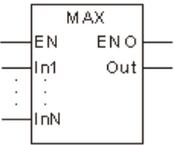
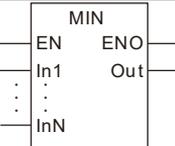


## 2.9 选择运算

### 2.9.1 MAX/MIN (最大值/最小值)

MAX(最大值):用于比较寻找多个变量或常量中的最大值,结果输出至“Out”。

MIN(最小值):用于比较寻找多个变量或常量中的最小值,结果输出至“Out”。所属库: Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MAX	最大值	FUN		Out:=MAX(In1,In2);
MIN	最小值	FUN		Out:=MIN(In1,In2);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	比较对象	输入	比较对象	变量的数据类型决定
In2至InN	比较对象		比较对象,程序编写时,梯形图时可通过编程软件增加或减少运算对象,即N=2~8	变量的数据类型决定
Out	最大值	输出	最大值	变量的数据类型决定

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

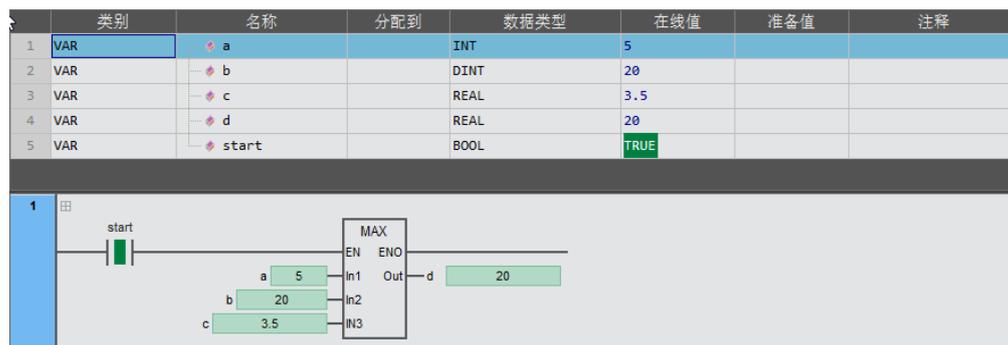
\*说明:上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- MAX(最大值):本指令用于比较寻找 In1~InN 个变量或常量中的最大值,结果输出至“Out”。
- MIN(最小值):本指令用于比较寻找 In1~InN 个变量或常量中的最小值,结果输出至“Out”。
- 当“In1”~“InN”及“Out”变量数据类型为位串、整数、实数时,允许“In1”~“InN”及“Out”为不同类型的变量,运算时以能包含“In1”~“InN”及“Out”所有取值范围的数据类型进行运算。如“In1”的数据类型为INT,“In2”的数据类型为INT,“Out”的数据类型为DINT,则运算时以DINT数据类型进行处理。
- “Out”数据类型的宽度须大于或等于“In1”~“In2”类型的宽度,否则,软件编译时会报错。如“In1”和“In2”的数据类型分别为INT和DINT,则“Out”的数据类型须为DINT、LINT等;如果“Out”所连变量的数据类型为SINT、INT等,软件编译时会报错。
- 当输入变量的数据类型有BOOL、TIME、DATE、TOD、DT、STRING数据类型时,要求“In1”~“InN”及“Out”均为相同类型数据类型。如“In1”的数据类型为DATE,“In2”及“Out”的数据类型也必须为DATE,否则软件编译时将报错。

- MAX 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
```

```
    d:=MAX(MAX(a,b),c);
```

```
END_IF;
```

- MIN 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
```

```
    d:=MIN(MIN (a,b),c);
```

```
END_IF;
```

## 2.9.2 SEL (位选择)

通过选择位将两个输入变量或常量中的1个搬移至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
SEL	位选择	FUN		Out:=SEL(G,In0, In1);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
G	选择位	输入	选择位	FALSE或者TRUE
In0、In1	选择选项		选择选项	变量的数据类型决定
Out	选择结果	输出	选择结果	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 通过选择位将两个输入变量或常量中的1个搬移至“Out”。选择哪个输入变量由选择位“G”决定，“G”为FALSE时，“In0”的值搬移至“Out”；“G”为TRUE时，“In1”的值搬移至“Out”。
- 当“In0”、“In1”及“Out”变量数据类型为位串、整数、实数时，允许“In0”、“In1”及“Out”为不同类型的变量，运算时以能包含“In0”、“In1”及“Out”所有取值范围的数据类型进行运算。如“In0”的数据类型为INT，“In1”的数据类型为INT，“Out”的数据类型为DINT,则运算时以DINT数据类型进行处理。
- “Out”数据类型的宽度须大于或等于“In0”、“In1”类型的宽度，否则，软件编译时会报错。如“In0”和“In1”的数据类型分别为INT和DINT，则“Out”的数据类型须为DINT、LINT等；如果“Out”所连变量的数据类型为SINT、INT等，软件编译时会报错。
- 当输入变量的数据类型有BOOL、TIME、DATE、TOD、DT、STRING数据类型时，要求“In0”、“In1”及“Out”均为相同类型数据类型。如“In0”的数据类型为DATE，“In1”及“Out”的数据类型也必须为DATE，否则软件编译时将报错。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

```
IF start THEN
```

```
    c:=SEL (G:=TRUE ,In1:=a ,In2:=b);
```

```
END_IF;
```

### 2.9.3 MUX (多路复用器)

通过选择条件将指定输入变量或常量的值搬移至“Out”。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MUX	多路复用器	FUN		Out:=MUX(K,In1, In2);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
K	选择条件	输入	选择条件	变量的数据类型决定
In0至InN	选择选项		选择选项 程序编写时，梯形图时可通过编程软件增加或减少选择选项参数，即N=0~7	变量的数据类型决定
Out	选择结果	输出	选择结果	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

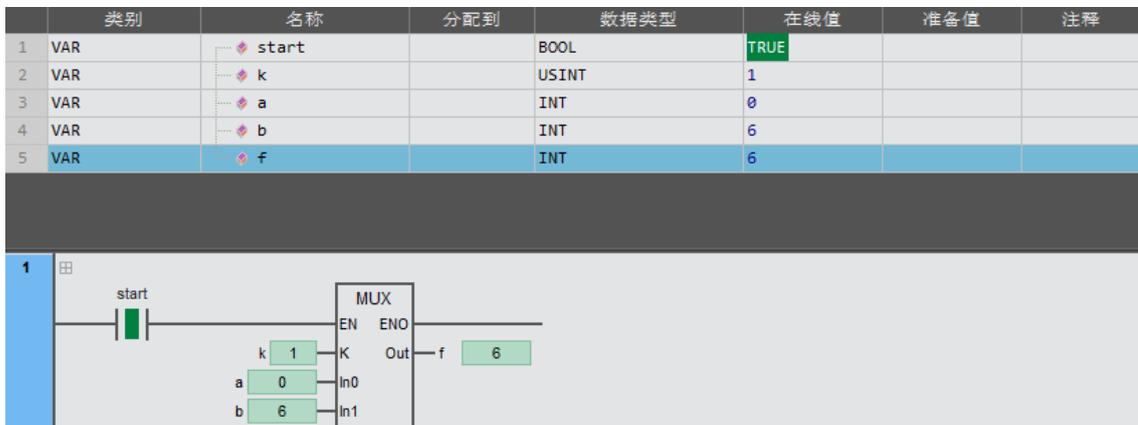
- 通过选择条件将指定输入变量或常量的值搬移至“Out”，选择哪个输入变量由选择条件“K”决定。详细如下表所示。

K取值	Out取值
0	In0 (Out:=In0)
1	In1 (Out:=In1)
2	In2 (Out:=In2)
3	In3 (Out:=In3)
4	In4 (Out:=In4)
5	In5 (Out:=In5)
6	In6 (Out:=In6)
7	In7 (Out:=In7)

- 当“In0”~“InN”及“Out”变量数据类型为位串、整数、实数时，允许“In0”~“InN”及“Out”为不同类型的变量，运算时以能包含“In0”~“InN”及“Out”所有取值范围的数据类型进行运算。如“In0”的数据类型为INT，“In1”的数据类型为INT，“Out”的数据类型为DINT,则运算时以DINT数据类型进行处理。
- “Out”数据类型的宽度须大于或等于“In0”~“InN”类型的宽度，否则，软件编译时会报错。如“In0”和“In1”的数据类型分别为INT和DINT，则“Out”的数据类型为须为DINT、LINT等；如果“Out”所连变量的数据类型为SINT、INT等，软件编译时会报错。

- 当输入变量的数据类型有 BOOL、TIME、DATE、TOD、DT、STRING 数据类型时，要求 “In0” ~ “InN” 及 “Out” 均为相同类型数据类型。如 “In1” 的数据类型为 DATE， “In2” 及 “Out” 的数据类型也必须为 DATE，否则软件编译时将报错。
- 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```

IF start THEN
    f:=MUX (K:=k ,In0:=a ,In1:=b);
END_IF;

```

## 2.9.4 LIMIT (上下限限制)

本指令用于限制输出值在输入值设定的上限和下限范围内。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
LIMIT	上下限限制	FUN		Out:=LIMIT(MN,In, MX);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
MN	下限值	输入	下限值	变量的数据类型决定
In	限制对象		限制对象	变量的数据类型决定
MX	上限值		上限值	变量的数据类型决定
Out	输出值	输出	输出值	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

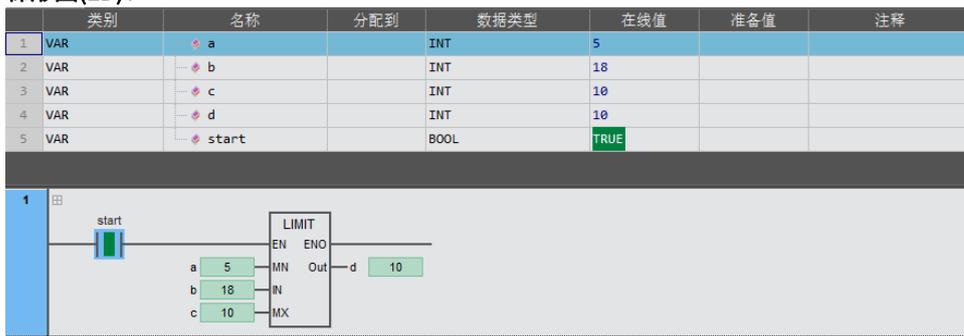
- 本指令用于限制输出值在输入值设定的上限和下限范围内，详细如下表所示。

In的值	Out的值
“In” < “MN”	“MN”
“MN” ≤ “In” ≤ “MX”	“In”
“MX” < “In”	“MX”

- 本指令允许输入和输出变量为不同类型的变量，运算时以能包含所有取值范围的数据类型进行运算。例如“MN”的数据类型为 REAL，“In”的数据类型为 REAL，“MX”的数据类型为 REAL，“Out”的数据类型为 LREAL，则运算时以 LREAL 数据类型进行处理。
- “Out”数据类型的宽度须大于或等于输入变量数据类型的宽度，否则，软件编译时会报错。例如“MN”、“In”、“MX”的数据类型分别为 LREAL、REAL 和 REAL，则“Out”的数据类型须为 LREAL；如果“Out”所连变量的数据类型为 REAL 等，软件编译时会报错。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

```
IF start THEN
```

```
    d:=LIMIT(MN:=a,In:=b, MX:=c);
```

```
END_IF;
```

## 2.9.5 BAND (死区限制)

本指令用于死区限制。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
BAND	死区限制	FUN		Out:=BAND(MN,In, MX);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
MN	下限值	输入	下限值	变量的数据类型决定
In	限制对象		限制对象	变量的数据类型决定
MX	上限值		上限值	变量的数据类型决定
Out	输出值	输出	输出值	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于死区限制，详细如下表所示。

In的值	Out的值
“In” < “MN”	“In” - “MN”
“MN” ≤ “In” ≤ “MX”	0
“MX” < “In”	“In” - “MX”

- 本指令允许输入和输出变量为不同类型的变量，运算时以能包含所有取值范围的数据类型进行运算。例如“MN”的数据类型为 REAL，“IN”的数据类型为 REAL，“MX”的数据类型为 REAL，“Out”的数据类型为 LREAL,则运算时以 LREAL 数据类型进行处理。
- “Out”数据类型的宽度须大于或等于输入变量数据类型的宽度，否则，软件编译时会报错。例如“MN”、“In”、“MX”的数据类型分别为 LREAL、REAL 和 REAL，则“Out”的数据类型须为 LREAL；如果“Out”所连变量的数据类型为 REAL 等，软件编译时会报错。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

```
IF start THEN
```

```
    d:=BAND(MN:=a,In:=b, MX:=c);
```

```
END_IF;
```

## 2.9.6 ZONE (输入偏移)

本指令用于输入值偏移。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
ZONE	输入偏移	FUN		Out:=ZONE(BiasN,In1, BiasP);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
BiasN	负偏移值	输入	负偏移值	变量的数据类型决定
In	控制对象		控制对象	变量的数据类型决定
BiasP	正偏移值		正偏移值	变量的数据类型决定
Out	输出值	输出	输出值	变量的数据类型决定

	布尔		位串			整数							实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
BiasN														<input type="radio"/>	<input type="radio"/>					
In														<input type="radio"/>	<input type="radio"/>					
BiasP														<input type="radio"/>	<input type="radio"/>					
Out														<input type="radio"/>	<input type="radio"/>					

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

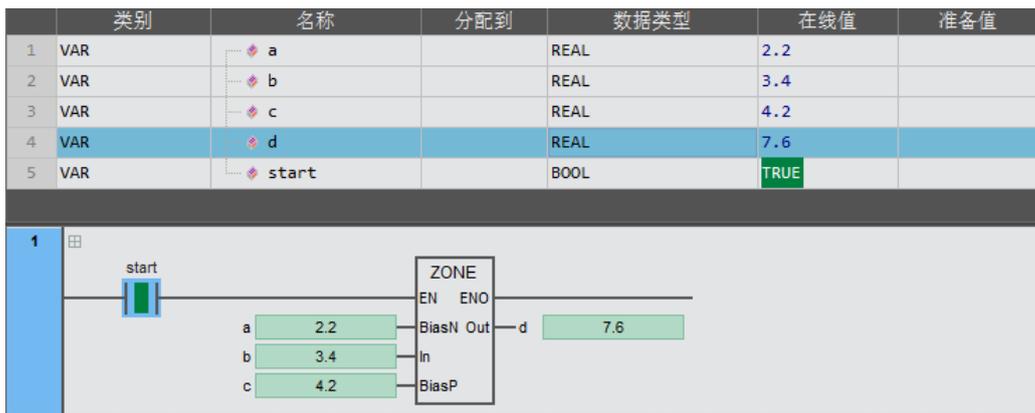
- 本指令用于输入值偏移，详细如下表所示。

In的值	Out的值
“In” < 0	“In” + “BiasN”
“In” = 0	0
“In” > 0	“In” + “BiasP”

- 本指令允许输入和输出变量为不同类型的变量，运算时以能包含所有取值范围的数据类型进行运算。如“BiasN”的数据类型为 REAL，“IN”的数据类型为 REAL，“BiasP”的数据类型为 LREAL，“Out”的数据类型为 LREAL,则运算时以 LREAL 数据类型进行处理。
- “Out”数据类型的宽度须大于或等于输入变量数据类型的宽度，否则，软件编译时会报错。如“BiasN”、“In”、“BiasP”的数据类型分别为 REAL、REAL 和 LREAL，则“Out”的数据类型必须为 LREAL；如果“Out”所连变量的数据类型为 REAL，软件编译时会报错。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

```
IF start THEN
```

```
    d:=ZONE(BiasN:=a,In:=b, BiasP:=c);
```

```
END_IF;
```

## 2.10 数据类型转换

### 2.10.1 BOOL\_TO\_\*\*\* (BOOL转换为其它数据类型)

本指令用于将布尔数据类型的数据转换为其它类型的数据。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
BOOL_TO_***	BOOL转换指令	FUN		<pre>Out:= BOOL_TO_*** (In) ; ***表示不同的数据类型，如 Out:=BOOL_TO_INT (In) 等</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	转换对象	输入	转换对象	FALSE或者TRUE
Out	转换结果	输出	转换结果	变量数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

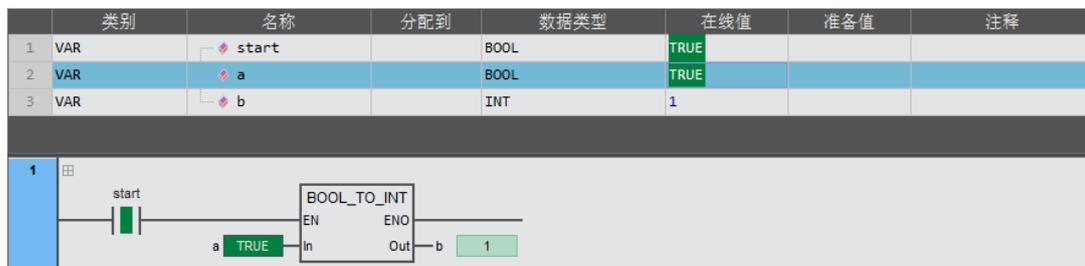
- 本指令用于将布尔数据类型的数据转换为其它类型的数据。
- 指令名称因“Out”的数据类型不同而不同，如“Out”为INT型，则指令名称为BOOL\_TO\_INT。
- 布尔数据类型数据和其他类型数据对应关系如下表所示。

“In”和“Out”的数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
BOOL	BOOL	FALSE	FALSE
		TRUE	TRUE
BOOL	BYTE	FALSE	0
		TRUE	1
BOOL	WORD	FALSE	0
		TRUE	1
BOOL	DWORD	FALSE	0
		TRUE	1
BOOL	LWORD	FALSE	0
		TRUE	1
BOOL	USINT	FALSE	0
		TRUE	1
BOOL	UINT	FALSE	0
		TRUE	1
BOOL	UDINT	FALSE	0
		TRUE	1

“In”和“Out”的数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
BOOL	ULINT	FALSE	0
		TRUE	1
BOOL	SINT	FALSE	0
		TRUE	1
BOOL	INT	FALSE	0
		TRUE	1
BOOL	DINT	FALSE	0
		TRUE	1
BOOL	LINT	FALSE	0
		TRUE	1
BOOL	REAL	FALSE	0
		TRUE	1
BOOL	LREAL	FALSE	0
		TRUE	1
BOOL	TIME	FALSE	T#0ms
		TRUE	T#0ms
BOOL	DATE	FALSE	D#1970-1-1
		TRUE	D#1970-1-1
BOOL	TOD	FALSE	TOD#0:0:0.000
		TRUE	TOD#0:0:0.000
BOOL	DT	FALSE	DT#1970-01-01-00:00:00
		TRUE	DT#1970-01-01-00:00:00
BOOL	STRING	FALSE	'FALSE'
		TRUE	'TRUE'

- 示例程序和时序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```

IF start THEN
    b:=BOOL_TO_INT(In:=a);
END_IF;

```

### 2.10.2 \*\*\*\_TO\_\*\*\* (位串转换为其它数据类型)

本指令用于将位串数据类型的数据转换为其它类型的数据。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
***_TO_***	位串转换指令	FUN		Out:= ***_TO_*** (In) ***表示不同的数据类型, 如 Out:= BYTE_TO_INT (In) 等

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	转换对象	输入	转换对象	变量数据类型决定
Out	转换结果	输出	转换结果	变量数据类型决定

	布尔	位串					整数							实数		时间, 日期			字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>															
Out	<input type="radio"/>																			

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将位串数据类型的数据转换为其它类型的数据。
- 指令名称因“In”和“Out”的数据类型不同而不同, 如“In”为 WORD 型, “Out”为 DINT 型, 则指令名称为 WORD\_TO\_DINT。
- 位串类型数据转换为布尔类型数据。

将位串转换为 BOOL 型数据时, 如果位串类型数值为 0, 则转换结果为 FALSE, 如数值不为 0, 则转换结果为 TRUE。详细规则如下表:

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
BYTE	BOOL	0	FALSE
		16#1~16#FF	TRUE
WORD	BOOL	0	FALSE
		16#1~16#FFFF	TRUE
DWORD	BOOL	0	FALSE
		16#1~16#FFFF_FFFF	TRUE
LWORD	BOOL	0	FALSE
		16#1~16#FFFF_FFFF_FFFF_FFFF	TRUE

- 位串类型数据转换为整数类型数据。

位串类型数据与整数类型数据转换关系如下表所示：

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
BYTE	USINT	0~16#FF	0~255
	UINT	0~16#FF	0~255
	UDINT	0~16#FF	0~255
	ULINT	0~16#FF	0~255
	SINT	0~16#7F	0~127
	INT	0~16#FF	0~255
	DINT	0~16#FF	0~255
	LINT	0~16#FF	0~255
WORD	USINT	0~16#FF	0~255
	UINT	0~16#FFFF	0~65535
	UDINT	0~16#FFFF	0~65535
	ULINT	0~16#FFFF	0~65535
	SINT	0~16#7F	0~127
	INT	0~16#7FFF	0~32767
	DINT	0~16#FFFF	0~65535
	LINT	0~16#FFFF	0~65535
DWORD	USINT	0~16#FF	0~255
	UINT	0~16#FFFF	0~65535
	UDINT	0~16#FFFF_FFFF	0~4294967295
	ULINT	0~16#FFFF_FFFF	0~4294967295
	SINT	0~16#7F	0~127
	INT	0~16#7FFF	0~32767
	DINT	0~16#7FFF_FFFF	0~2147483647
	LINT	0~16#FFFF_FFFF	0~4294967295
LWORD	USINT	0~16#FF	0~255
	UINT	0~16#FFFF	0~65535
	UDINT	0~16#FFFF_FFFF	0~4294967295
	ULINT	0~16#FFFF_FFFF_FFFF_FFFF	0~18446744073709551615
	SINT	0~16#7F	0~127
	INT	0~16#7FFF	0~32767
	DINT	0~16#7FFF_FFFF	0~2147483647
	LINT	0~16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807

- 位串类型数据转换为浮点数类型数据。

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
BYTE	REAL	0~16#FF	0~255
	LREAL	0~16#FF	0~255
WORD	REAL	0~16#FFFF	0~65535
	LREAL	0~16#FFFF	0~65535
DWORD	REAL	0~16#FFFF_FFFF	0~4.294967e+9
	LREAL	0~16#FFFF_FFFF	0~4294967295

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
LWORD	REAL	0~16#FFFF_FFFF_FFFF_FFFF	0~1.844674e+19
	LREAL	0~16#FFFF_FFFF_FFFF_FFFF	0~1.84467440737096e+19

- 位串类型数据转换为时间或日期类型数据。

位串转换为时间或日期对应关系如下表所示：

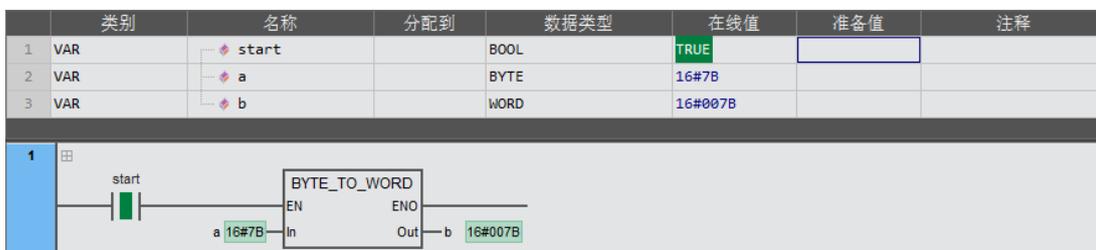
“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
BYTE	TIME	0~16#FF	T#0ns~ T#255ns
	DATE	0~16#FF	D#1970-1-1~ D#1970-1-1
	TOD	0~16#FF	TOD#0:0:0~ TOD#0:0:0.255
	DT	0~16#FF	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15
WORD	TIME	0~16#FFFF	T#0ns~65us535ns
	DATE	0~16#FFFF	D#1970-1-1~ D#1970-1-1
	TOD	0~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	0~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15
DWORD	TIME	0~16#FFFF_FFFF	T#0ms~4s294ms967us295ns
	DATE	0~16#FFFF_A500	D#1970-1-1~ D#2106-2-7
	TOD	0~16#526_5BFF	TOD#0:0:0~ TOD#23:59:59.999
	DT	0~16#FFFF_FFFF	DT#1970-1-1-0:0:0~ DT#2106-2-7-06:28:15
LWORD	TIME	0~16#FFFF_FFFF_FFFF_FFFF	T#0ms~213503d23h34m33s709.551ms
	DATE	0~16#FFFF_A500	D#1970-1-1~ D#2106-2-7
	TOD	0~16#526_5BFF	TOD#0:0:0~ TOD#23:59:59.999
	DT	0~16#FFFF_FFFF	DT#1970-1-1-0:0:0~ DT#2106-2-7-06:28:15

- 位串类型数据转换为字符串类型数据。

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
BYTE	STRING	0~16#FF	'0' ~ '255'
WORD	STRING	0~16#FFFF	'0' ~ '65535'
DWORD	STRING	0~16#FFFF_FFFF	'0' ~ '4294967295'
LWORD	STRING	0~16#FFFF_FFFF_FFFF_FFFF	'0' ~ '18446744073709551615'

- 示例程序如下所示：

**梯形图(LD):**



**结构化文本(ST):**

```
IF start THEN
    b:=BYTE_TO_WORD(In:=a);
END_IF;
```

### 2.10.3 \*\*\*\_TO\_\*\*\* (整数转换为其它数据类型)

本指令用于将整数数据类型的数据转换为其它类型的数据。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
***_TO_***	整数转换指令	FUN		Out:= ***_TO_*** (In) ***表示不同的数据类型, 如 Out:= =UINT_TO_BOOL (In) 等

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	转换对象	输入	转换对象	变量数据类型决定
Out	转换结果	输出	转换结果	变量数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将数据类型为整数的“In”转换为“Out”指定数据类型的数据。
- 指令名称因“In”和“Out”的数据类型不同而不同, 如“In”为WORD型, “Out”为DINT型, 则指令名称为WORD\_TO\_DINT。
- 整数类型数据转换为布尔类型数据。

将整数转换为BOOL型数据时, 如果整数值为0, 则转换结果为FALSE, 如整数值不为0, 则转换结果为TRUE。详细规则如下表:

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
USINT	BOOL	0	FALSE
		1~255	TRUE
UINT	BOOL	0	FALSE
		1~65535	TRUE
UDINT	BOOL	0	FALSE
		1~4294967295	TRUE
ULINT	BOOL	0	FALSE
		1~18446744073709551615	TRUE
SINT	BOOL	0	FALSE
		-128~-1, 1~127	TRUE
INT	BOOL	0	FALSE
		-32768~-1, 1~32767	TRUE
DINT	BOOL	0	FALSE
		-2147483648~-1, 1~2147483647	TRUE

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
LINT	BOOL	0	FALSE
		-9223372036854775808~-1, 1~9223372036854775807	TRUE

- 整数类型数据转换为位串类型数据。

整数类型数据与位串类型数据转换关系如下表所示：

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
USINT	BYTE	0~255	16#00~16#FF
	WORD	0~255	16#0000~16#00FF
	DWORD	0~255	16#0000_0000~16#0000_00FF
	LWORD	0~255	16#0000_0000_0000_0000~16#0000_0000_0000_00FF
UINT	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~65535	16#0000_0000~16#0000_FFFF
	LWORD	0~65535	16#0000_0000_0000_0000~16#0000_0000_0000_FFFF
UDINT	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~4294967295	16#0000_0000~16#FFFF_FFFF
	LWORD	0~4294967295	16#0000_0000_0000_0000~16#0000_0000_FFFF_FFFF
ULINT	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~4294967295	16#0000_0000~16#FFFF_FFFF
	LWORD	0~18446744073709551615	16#0000_0000_0000_0000~16#FFFF_FFFF_FFFF_FFFF
SINT	BYTE	0~127	16#00~16#7F
	WORD	0~127	16#0000~16#007F
	DWORD	0~127	16#0000_0000~16#0000_007F
	LWORD	0~127	16#0000_0000_0000_0000~16#0000_0000_0000_007F
INT	BYTE	0~127	16#00~16#7F
	WORD	0~32767	16#0000~16#7FFF
	DWORD	0~32767	16#0000_0000~16#0000_7FFF
	LWORD	0~32767	16#0000_0000_0000_0000~16#0000_0000_0000_7FFF
DINT	BYTE	0~127	16#00~16#7F
	WORD	0~32767	16#0000~16#7FFF
	DWORD	0~2147483647	16#0000_0000~16#7FFF_FFFF
	LWORD	0~2147483647	16#0000_0000_0000_0000~16#0000_0000_7FFF_FFFF
LINT	BYTE	0~127	16#00~16#7F
	WORD	0~32767	16#0000~16#7FFF
	DWORD	0~2147483647	16#0000_0000~16#7FFF_FFFF
	LWORD	0~9223372036854775807	16#0000_0000_0000_0000~16#7FFF_FFFF_FFFF_FFFF

- 整数类型数据转换为浮点数类型数据。

整数类型数据与浮点数值数据转换关系如下表所示：

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
USINT	REAL	0~255	0~255
	LREAL	0~255	0~255
UINT	REAL	0~65535	0~65535
	LREAL	0~65535	0~65535
UDINT	REAL	0~4294967295	0~4.294967e+9
	LREAL	0~4294967295	0~4294967295
ULINT	REAL	0~18446744073709551615	0~1.844674e+19
	LREAL	0~18446744073709551615	0~1.84467440737096e+19
SINT	REAL	-128~127	-128~127
	LREAL	-128~127	-128~127
INT	REAL	-32768~32767	-32768~32767
	LREAL	-32768~32767	-32768~32767
DINT	REAL	-2147483648~2147483647	-2.147484e+9~2.147484e+9
	LREAL	-2147483648~2147483647	-2147483648~2147483648
LINT	REAL	-9223372036854775808~9223372036854775807	-9.223372e+18~9.223372e+18
	LREAL	-9223372036854775808~9223372036854775807	-9.22337203685478e+18~9.22337203685478e+18

- 整数类型数据转换为时间或日期类型数据。

整数转换为时间或日期对应关系如下表所示：

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
USINT	TIME	0~255	T#0ns~ T#255ns
	DATE	0~255	D#1970-1-1~ D#1970-1-1
	TOD	0~255	TOD#0:0:0~ TOD#0:0:0.255
	DT	0~255	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15
UINT	TIME	0~65535	T#0ns~65us535ns
	DATE	0~65535	D#1970-1-1~ D#1970-1-1
	TOD	0~65535	TOD#0:0:0~ TOD#0:1:5.535
	DT	0~65535	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15
UDINT	TIME	0~4294967295	T#0ms~4s294ms967us295ns
	DATE	0~4294944000	D#1970-1-1~ D#2106-2-7
	TOD	0~86399999	TOD#0:0:0~ TOD#23:59:59.999
	DT	0~4294967295	DT#1970-1-1-0:0:0~ DT#2106-2-7-06:28:15
ULINT	TIME	0~18446744073709551615	T#0ms~213503d23h34m33s709.551ms
	DATE	0~4294944000	D#1970-1-1~ D#2106-2-7
	TOD	0~86399999	TOD#0:0:0~ TOD#23:59:59.999
	DT	0~4294967295	DT#1970-1-1-0:0:0~ DT#2106-2-7-06:28:15
SINT	TIME	0~127	T#0ns~ T#127ns
	DATE	0~127	D#1970-1-1~ D#1970-1-1
	TOD	0~127	TOD#0:0:0~ TOD#0:0:0.127
	DT	0~127	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:2:07
INT	TIME	0~32767	T#0ns~ T#32us767ns

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
	DATE	0~32767	D#1970-1-1~ D#1970-1-1
	TOD	0~32767	TOD#0:0:0~ TOD#0:0:32.767
	DT	0~32767	DT#1970-1-1-0:0:0~ DT#1970-1-1-09:06:07
DINT	TIME	0~2147483647	T#0ns~ T#2s147ms483ms647ns
	DATE	0~2147483647	D#1970-1-1~ D#2038-1-19
	TOD	0~86399999	TOD#0:0:0~ TOD#23:59:59.999
	DT	0~2147483647	DT#1970-1-1-0:0:0~ D#2038-1-19-03:14:07
LINT	TIME	0~9223372036854775807	T#0ms~106751d23h47m16s854.776ms
	DATE	0~4294944000	D#1970-1-1~ D#2106-2-7
	TOD	0~86399999	TOD#0:0:0~ TOD#23:59:59.999
	DT	0~4294967295	DT#1970-1-1-0:0:0~ DT#2106-2-7-06:28:15

- 整数类型数据转换为字符串类型数据。

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
In	Out	“In” 的数据有效范围	“Out” 的数据有效范围
USINT	STRING	0~255	'0' ~ '255'
UINT	STRING	0~65535	'0' ~ '65535'
UDINT	STRING	0~4294967295	'0' ~ '4294967295'
ULINT	STRING	0~18446744073709551615	'0' ~ '18446744073709551615'
SINT	STRING	-128~127	'-128' ~ '-127'
INT	STRING	-32768~32767	'-32768' ~ '-32767'
DINT	STRING	-2147483648~2147483647	'-2147483648' ~ '2147483647'
LINT	STRING	-9223372036854775808~ 9223372036854775807	'-9223372036854775808' ~ '9223372036854775807'

- 示例程序如下所示：

**梯形图(LD):**



**结构化文本(ST):**

```

IF start THEN
    b:=INT_TO_DINT(In:=a);
END_IF;
    
```

## 2.10.4 REAL/LREAL\_TO\_\*\*\* (实数转换为其它数据类型)

本指令用于将实数数据类型的数据转换为其它类型的数据。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
***_TO_***	实数转换指令	FUN		Out:= ***_TO_*** (In) ***表示不同的数据类型，如 Out:= REAL_TO_DINT (In) 等

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	转换对象	输入	转换对象	变量数据类型决定
Out	转换结果	输出	转换结果	变量数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于将实数数据类型的数据转换为其它类型的数据。
- 指令名称因“In”和“Out”的数据类型不同而不同，如“In”数据类型为 REAL，“Out”数据类型为 DINT 型，则指令名称为 REAL\_TO\_DINT。
- 实数类型数据转换为布尔类型数据。

将实数类型数据转换为 BOOL 类型数据时，如果输入实数数值为 0，输出值为 FALSE；如果输入实数数值不为 0，输出值为 TRUE。详细规则如下表：

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的值	“Out”的值
REAL	BOOL	-3.1	TRUE
		0	FALSE
		3.1	TRUE
LREAL	BOOL	-6	TRUE
		0	FALSE
		6	TRUE

- 实数转换为整数或者位串时转换规则。

实数转换为整数或者位串时，“In”值小数部分进行四舍五入处理，“In”值四舍五入后的整数为“Out”的值。相关示例如下表所示。

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的值	“Out”的值
REAL	SINT	1.49	1
REAL	SINT	1.50	2

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的值	“Out”的值
REAL	SINT	1.55	2
REAL	SINT	-1.49	-1
REAL	SINT	-1.50	-2
REAL	SINT	-1.55	-2
REAL	SINT	127	127
REAL	BYTE	127	16#7F

- 实数类型数据转换为位串类型数据。

将实数类型数据转换为位串类型数据时，先将输入实数数值进行四舍五入，输出值为四舍五入后的整数部分。实数类型数据和位串类型数据转换时，输入和输出数据有效范围内的对应关系如下表所示：

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
REAL	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~4.294967e+9	16#0000_0000~16#FFFF_FFFF
	LWORD	0~1.844674e+19	16#0000_0000_0000_0000~16#FFFF_FFFF_FFFF_FFFF
LREAL	BYTE	0~255	16#00~16#FF
	WORD	0~65535	16#0000~16#FFFF
	DWORD	0~4294967295	16#0000_0000~16#FFFF_FFFF
	LWORD	0~1.84467440737095e+19	16#0000_0000_0000_0000~16#FFFF_FFFF_FFFF_FFFF

- 实数类型数据转换为整数类型数据。

将实数类型数据转换为整数类型数据时，先将输入实数数值进行四舍五入，输出值为输入值四舍五入后的整数部分。实数类型数据和整数类型数据转换时，输入和输出数据有效范围内的对应关系如下表所示：

“In”和“Out”数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
REAL	USINT	0~255	0~127
	UINT	0~65535	0~32767
	UDINT	0~4.294967e+9	0~4294967295
	ULINT	0~1.844674e+19	0~18446744073709551615
	SINT	-128~127	-128~127
	INT	-32768~32767	-32768~32767
	DINT	-2.147484e+9~2.147484e+9	-2147483648~2147483647
	LINT	-9.223372e+18~9.223372e+18	-9223372036854775808~9223372036854775807
LREAL	USINT	0~255	0~255
	UINT	0~65535	0~65535
	UDINT	0~4294967295	0~4294967295
	ULINT	0~1.84467440737095e+19	0~18446744073709551615
	SINT	-128~127	-128~127
	INT	-32768~32767	-32768~32767
	DINT	-2147483648~2147483648	-2147483648~2147483647
	LINT	-9.22337203685477e+18~9.22337203685477e+18	-9223372036854775808~9223372036854775807

- 实数类型数据转换为时间或日期类型数据。

将实数类型数据转换为时间相关类型数据时，先将实数类型数据转换为 ULINT 类型数据，ULINT 类型数据再按照输出类型的单位转换为对应的时间。ULINT 转换为时间或日期相关可以参考“整数类型转换为其它数据类型”指令中的说明。

- 实数类型数据转换为字符串类型数据。

将实数类型数据转换为字符串类型数据时，输入数据中的字符和符号都转换为对应的字符。实数类型数据和字符串类型数据转换时例子如下表所示：

“In” 和 “Out” 数据类型		“In” 和 “Out” 数据对应关系	
In	Out	“In” 的值	“Out” 的值
REAL	STRING	123.123	123.123
REAL	STRING	-123.123	-123.123
REAL	STRING	1.23e+07	1.23e+07

示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
IF start THEN
    b:=REAL_TO_INT(In:=a);
END_IF;
```

### 2.10.5 TIME/DATE\_TO\_\*\*\* (时间、日期转换为其它数据类型)

本指令用于将时间、日期数据类型的数据转换为其它类型的数据。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
TIME/DATE_TO_***	时间、日期转换指令	FUN		Out:= ***_TO_*** (In) ; ***表示不同的数据类型, 如 Out:=TIME_TO_INT (In) 等

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	转换对象	输入	转换对象	变量数据类型决定
Out	转换结果	输出	转换结果	变量数据类型决定

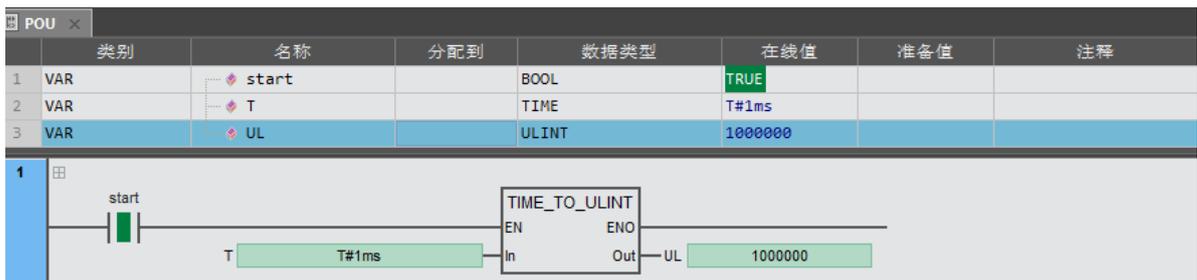
	布尔	位串					整数							实数		时间, 日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																○	○	○	○	
Out	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将时间、日期数据类型的数据转换为其它类型的数据。
- 指令名称因“In”和“Out”的数据类型不同而不同，如“In”数据类型为 TIME，“Out”为 ULINT 型，则指令名称为 TIME\_TO\_ULINT。
- 示例程序和时序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```

IF start THEN
    UL:=TIME_TO_ULINT(In:=T);
END_IF;
    
```

## 2.10.6 STRING\_TO\_\*\*\* (STRING转换为其它数据类型)

本指令用于将STRING数据类型的数据转换为其它类型的数据。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
STRING_TO_***	STRING 转换指令	FUN		Out:= STRING_TO_*** (In) ; ***表示不同的数据类型, 如 Out:=STRING_TO_DINT (In) 等

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	转换对象	输入	转换对象	FLASE或者TRUE
Out	转换结果	输出	转换结果	变量数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于将 STRING 数据类型的数据转换为其它类型的数据。
- 指令名称因“Out”的数据类型不同而不同，如“Out”为 INT 型，则指令名称为 STRING\_TO\_INT。
- STRING 数据类型数据和其他类型数据对应关系如下表所示。

“In”和“Out”的数据类型		“In”和“Out”数据对应关系	
“In”	“Out”	“In”的数据有效范围	“Out”的数据有效范围
STRING	BOOL	其他字符串	FALSE
		‘TRUE’ 或 ‘true’	TRUE
STRING	BYTE	‘0’ ~ ‘255’	0~16#FF
STRING	WORD	‘0’ ~ ‘65535’	0~16#FFFF
STRING	DWORD	‘0’ ~ ‘4294967295’	0~16#FFFF_FFFF
STRING	LWORD	‘0’ ~ ‘18446744073709551615’	0~16#FFFF_FFFF_FFFF_FFFF
STRING	USINT	‘0’ ~ ‘255’	0~255
STRING	UINT	‘0’ ~ ‘65535’	0~65535
STRING	UDINT	‘0’ ~ ‘4294967295’	0~4294967295
STRING	ULINT	‘0’ ~ ‘18446744073709551615’	0~18446744073709551615
STRING	SINT	‘-128’ ~ ‘-127’	-128~127
STRING	INT	‘-32768’ ~ ‘-32767’	-32768~32767

“In” 和 “Out” 的数据类型		“In” 和 “Out” 数据对应关系	
“In”	“Out”	“In” 的数据有效范围	“Out” 的数据有效范围
STRING	DINT	‘-2147483648’ ~ ‘2147483647’	-2147483648~2147483647
STRING	LINT	‘-9223372036854775808’ ~ ‘9223372036854775807’	-9223372036854775808~ 9223372036854775807
STRING	REAL	‘-3.402823e+38’ ~ ‘-1.175495e-38’ , ‘0’ , ‘1.175495e-38 ~ ‘3.402823e+38’	-3.402823e+38 ~ -1.175495e-38, 0, 1.175495e-38 ~ 3.402823e+38
STRING	LREAL	‘-1.79769313486231e+308’ ~ ‘-2.22507385850721e-308’ , ‘0’ , ‘2.22507385850721e-308’ ~ ‘1.79769313486231e+308’	-1.79769313486231e+308 ~ - 2.22507385850721e-308, 0, 2.22507385850721e-308 ~ 1.79769313486231e+308,
STRING	TIME	‘T#0ms’ ~ ‘T#213503d23h34m33s709.551ms’	T#0ms~ T#213503d23h34m33s709.551ms
STRING	DATE	‘D#1970-1-1’ ~ ‘D#2106-2-7’	D#1970-1-1~ D#2106-2-7
STRING	TOD	‘TOD#0:0:0’ ~ ‘TOD#23:59:59.999’	TOD#0:0:0~ TOD#23:59:59.999
STRING	DT	‘DT#1970-1-1-0:0:0’ ~ ‘DT#2106-2-7-06:28:15’	DT#1970-1-1-0:0:0~ DT#2106-2-7-06:28:15

- 示例程序和时序如下所示：

**梯形图(LD)：**



**结构化文本(ST)：**

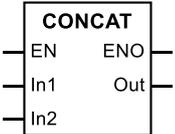
```

IF start THEN
    b:=STRING_TO_INT(In:=a);
END_IF;
    
```

## 2.11 字符串指令

### 2.11.1 CONCAT (连接字符串)

用于将2个字符串类型的变量或常量连接成1个字符串。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
CONCAT	连接字符串	FUN		Out := CONCAT(In1:=,In2:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	字符串	输入	输入字符串	变量的数据类型决定
In2	字符串		输入字符串	变量的数据类型决定
Out	字符串	输出	连接结果字符串	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将 In1 和 In2 字符串依次连接组成 1 个新的字符串并输出至 Out。
- 示例程序如下所示：

##### 梯形图(LD):

例如In1为‘abc’，In2为‘456’，连接的结果为‘abc456’。



##### 结构化文本(ST):

```
Str3:=CONCAT(In1:=Str1,In2:=Str2);
```

### 2.11.2 DELETE (删减字符串)

本指令用于将输入字符串类型从指定位置开始删除指定长度后，产生新的字符串。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
DELETE	删减字符串	FUN		Out := DELETE(In:=,L:=,P:=);

#### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	字符串	输入	输入字符串	变量的数据类型决定
L	个数		删减字符个数	变量的数据类型决定
P	开始位置		删减字符的开始位置	变量的数据类型决定
Out	字符串	输出	删减连接结果字符串	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### 功能说明

- 本指令用于将字符串 (In) 从位置 P 开始删减 L 个字符。将结果输出 “Out”。
- 示例程序如下所示：

##### 梯形图(LD):

例如In为 ‘ABCDEFGH’ ,L为3, P为3, 结果为 ‘AEFGH’。



##### 结构化文本(ST):

Str2:=DELETE(In:=Str1,L:=UINT#3,P:=UINT#3);

### 2.1.1.3 INSERT (字符串插入)

本指令用于向字符串指定位置插入1串字符，并将产生的新字符串输出值Out。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
INSERT	字符串插入	FUN		Out := INSERT(In1:= ,In2:= ,P:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	字符串	输入	输入字符串	变量的数据类型决定
In2	插入的字符串		插入的字符串	变量的数据类型决定
P	插入位置		插入位置	0~字符串In1最大长度
Out	产生的字符串	输出	产生的字符串	变量的数据类型决定

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

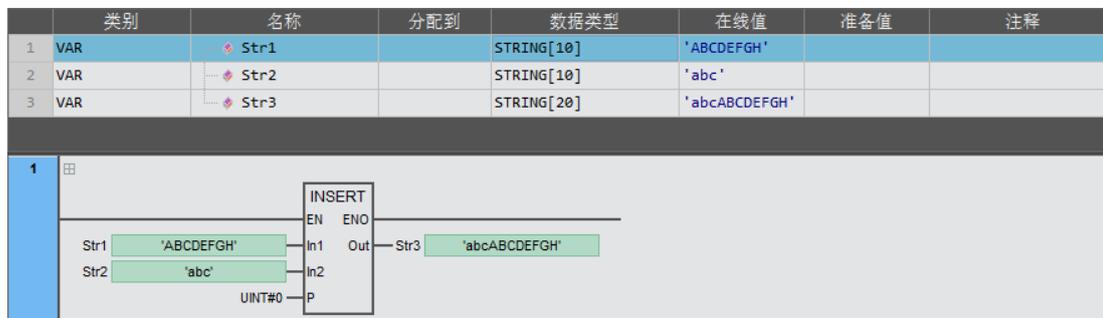
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于向字符串（In1）的指定位置插入另1字符串（In2），将产生的新字符串输出至 Out。例如 In1 为‘ABCDEFGH’，In2 为‘abc’，P 为 0，结果为‘abcABCDEFGH’。
- 示例程序 1 如下所示：

##### 梯形图(LD):

例如 In1 为‘ABCDEFGH’，In2 为‘abc’，P 为 0，结果为‘abcABCDEFGH’。



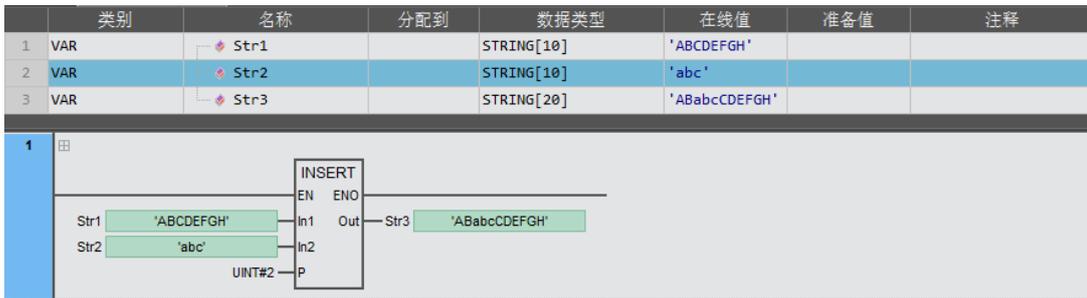
##### 结构化文本(ST):

```
Str3:=INSERT(In1:= Str1 ,In2:= Str2 ,P:=UINT#0);
```

- 示例程序 2 如下所示：

#### 梯形图(LD):

例如In1为 'ABCDEFGH' ,In2为 'abc' ,P为2, 结果为 'ABabcCDEFGH' 。



#### 结构化文本(ST):

```
Str3:=INSERT(In1:= Str1 ,In2:= Str2 ,P:=UINT#2);
```

### 2.1.1.4 LEFT/RIGHT (左/右截取字符串)

本指令用于向字符串指定位置插入1串字符，并将产生的新字符串输出值Out。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
LEFT	左截取字符串	FUN		Out := LEFT(In:=,L:=);
RIGHT	右截取字符串	FUN		Out := RIGHT(In:=,L:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	字符串	输入	原始字符串	变量的数据类型决定
L	截取长度		截取长度	0~字符串In最大长度
Out	产生的字符串	输出	产生的字符串	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 指令 LEFT 和 RIGHT 用于从字符串 (In) 左或右截取指定长度的字符串，并将结果输出至 Out。
- LEFT 从 In 的左侧开始截取，例如 In 为 ‘ABCDEFGH’，L 为 4，结果为 ‘ABCD’。
- LEFT 示例程序如下所示：

##### 梯形图(LD):

LEFT从In的左侧开始截取，例如In为‘ABCDEFGH’，L为4，结果为‘ABCD’。



##### 结构化文本(ST):

```
Str2:= LEFT(In:=Str1,L:=UINT#4);
```

- RIGHT 示例程序如下所示：

**梯形图(LD):**

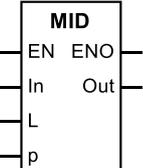
RIGHT从In的右侧开始截取，例如In为‘ABCDEFGH’，L为4，结果为‘EFGH’。

**结构化文本(ST):**

```
Str2:= RIGHT(In:=Str1 ,L:=UINT#4);
```

### 2.1.1.5 MID (字符串截取)

本指令用于从字符串指定位置截取指定长度并生成新字符串。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
MID	字符串截取	FUN		Out := MID(In:=,L:=,P:=);

#### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	字符串	输入	字符串	变量的数据类型决定
L	截取个数		截取个数	0~字符串In最大长度
P	起始位置		起始位置	1~字符串In最大长度
Out	产生的字符串	输出	产生的字符串	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### 功能说明

- 此指令用于从字符串 (In) 中从第 P 个字符开始截取 L 个字符，并将产生的字符串输出至 Out。
- 示例程序如下所示：

##### 梯形图(LD):

例如字符串In为‘ABCDEFGH’,P为2,L为3,结果为‘BCD’。



##### 结构化文本(ST):

```
Str2:= MID(In:=Str1 ,L:=UINT#3 ,P:=UINT#2);
```

### 2.11.6 REPLACE (字符替换)

本指令用于将字符串中指定位置、指定长度的字符串替换成另1字符串。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
REPLACE	字符替换	FUN		Out := REPLACE(In1:= ,In2:= ,L:= ,P:= );

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	原始字符串	输入	原始字符串	变量的数据类型决定
In2	替换字符串		替换字符串	变量的数据类型决定
L	替换个数		替换个数	0~字符串In最大长度
P	起始位置		起始位置	1~字符串In最大长度
Out	产生的字符串	输出	产生的字符串	变量的数据类型决定

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

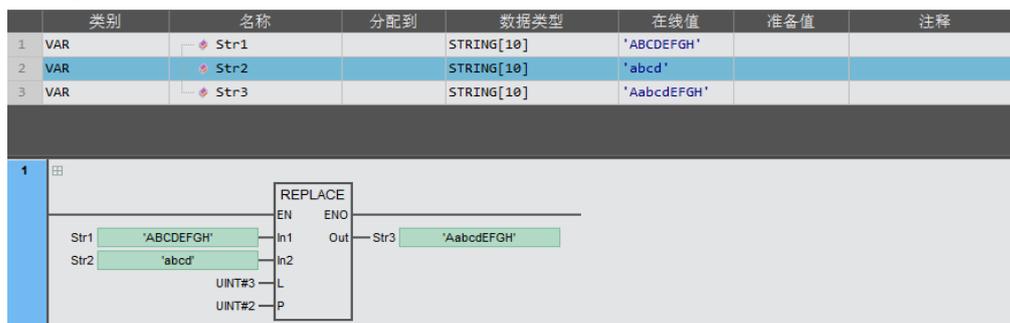
\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于将字符串 (In1) 中从第 P 个字符开始的 L 个字符替换成另 1 字符串 (In2)。
- 示例程序如下所示：

##### 梯形图(LD):

产生的字符串输出至 Out。例如字符串 In1 为 'ABCDEFGH', In2 为 'abcd', L 为 3, P 为 2, 结果为 'AabcdEFGH'。

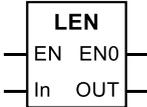


##### 结构化文本(ST):

Str3:=REPLACE(In1:=Str1 ,In2:=Str2 ,L:=UINT#3 ,P:=UINT#2 );

### 2.1.1.7 LEN (计算字符串长度)

本指令用于将字符串中指定位置、指定长度的字符串替换成另1字符串。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
LEN	计算字符串长度	FUN		Out := LEN(In:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In	字符串	输入	字符串	变量的数据类型决定
Out	字符个数	输出	字符个数	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

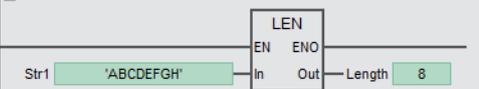
#### ■ 功能说明

- 本指令用于计算字符串的字符数输出至 Out。
- 示例程序如下所示：

##### 梯形图(LD):

例如In为ABCDEFGH时，Out为8。

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	Str1		STRING[10]	'ABCDEFGH'		
2	VAR	Length		UINT	8		

##### 结构化文本(ST):

```
Length := LEN(In:=Str1);
```

### 2.1.1.8 FIND (查找字符串)

本指令用于将字符串中指定位置、指定长度的字符串替换成另1字符串。所属库：Standard

指令	名称	FB/FUN	梯形图样式	ST样式
FIND	计算字符串长度	FUN		Out := FIND(In1:= ,In2:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
In1	原字符串	输入	原字符串	变量的数据类型决定
In2	查找的关键字符		查找的关键字符	变量的数据类型决定
Out	字符起始位置	输出	字符起始位置	变量的数据类型决定

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令将 In2 作为关键字符，在 In1 中查找其所在位置，并输出至 Out。
- 若无法在 In1 内查找到 In2，Out 为 0。若 In1 中存在多个 In2，Out 为 In2 第 1 次被查到的位置。
- 示例程序如下所示：

##### 梯形图(LD):

例如In1为ABCDEFGH, In2为DE, 则Out为4。

	类别	名称	分配到	数据类型	在线值	准备值	注释
1	VAR	Str1		STRING[10]	'ABCDEFGH'		
2	VAR	Str2		STRING[10]	'DE'		
3	VAR	Out		UINT	4		

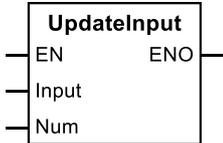
##### 结构化文本(ST):

Out:= FIND(In1:=Str1 ,In2:=Str2);

## 2.12 IO刷新和PID

### 2.12.1 UpdateInput (输入点立即刷新)

本指令用于立即刷新输入点的状态。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
UpdateInput	输入点立即刷新	FUN		UpdateInput(Input:=,Num:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Input	输入点	输入	控制器本体输入点	0~15 (0:%IX0.0, 1:%IX0.1, 7:%IX0.7, 8:%IX1.0, 14:%IX1.6, 15:%IX1.7)
Num	数量	输入	指定刷新输入点的数量, 以Input指定输入点开始计算	1~16

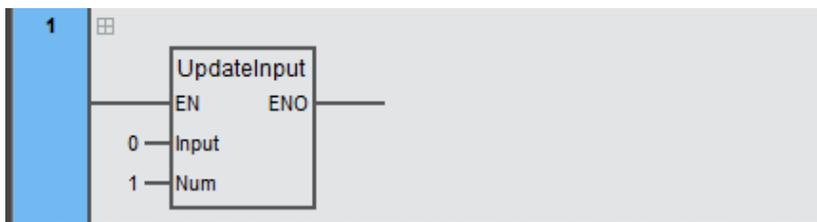
	布尔		位串			整数							实数		时间, 日期			字符串			
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
Input							○														
Num						○															

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于控制器本身的输入点立即刷新(扩展模块输入点不支持)。当执行该指令时, 可以立即检测到外部输入点的状态; 如果不使用本指令, 需要到下个扫描周期开始时才刷新外部输入点的状态。
- 示例程序如下所示:

梯形图(LD):

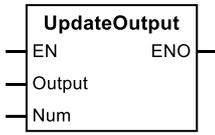


结构化文本(ST):

```
UpdateInput(Input:=0,Num:=1);
```

## 2.12.2 UpdateOutput (输出点立即刷新)

本指令用于立即刷新输入点的状态。所属库: Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
UpdateOutput	输出点立即刷新	FUN		Out:=UpdateOutput (Output:=,Num:=);

### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Output	输出点	输入	控制器本体输出点	0~15 (0:%QX0.0, 1:%QX0.1, 7:%QX0.7, 8:%QX1.0, 14:%IX1.6, 15:%IX1.7)
Num	数量	输入	指定刷新输出点的数量, 以output指定输出点开始计算	1~16

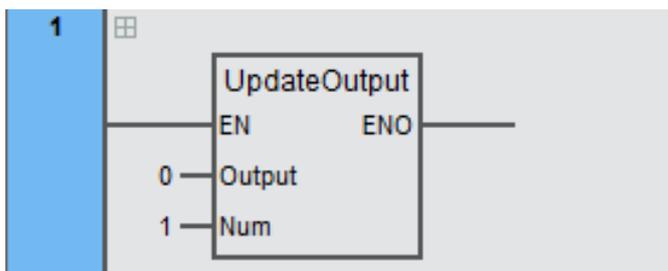
	布尔		位串			整数							实数		时间, 日期			字符串			
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
Output							○														
Num						○															

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### ■ 功能说明

- 本指令用于控制器本身的输出点立即刷新。当执行该指令时, 可以立即检测到外部输出点的状态; 如果不使用本指令, 需要到下个扫描周期开始时才刷新外部输入点的状态。
- 示例程序如下所示:

梯形图(LD):



结构化文本(ST):

```
UpdateOutput (Input:=0 ,Num:=1 );
```

### 2.12.3 PWM\_S (可变占空比脉冲输出)

可调整占空比脉冲输出。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
PWM_S	脉冲调变	FB		<pre>PWM_S__instance (Enable:=,OnTime:=,CycleTime:=,Out=&gt;);</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Enable	执行位	输入	执行位	TRUE或FALSE
OnTime	脉冲宽度		脉冲输出宽度 (ms)	0~32767
CycleTime	脉冲周期		脉冲输出周期 (ms)	1~32767
Out	脉冲输出	输出	脉冲输出变量或装置	TRUE或FALSE

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

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 该指令用于控制输出点进行脉冲输出。OnTime 为输出脉冲的高电平时间，单位为毫秒 (ms)。CycleTime 为输出脉冲周期，即输出脉冲高电平和低电平的时间和，单位为毫秒 (ms)。
- OnTime 和 CycleTime 可在指令执行时更改。
- OnTime 大于等于 CycleTime 时，Out 的值一直为 TRUE。
- 该指令控制输出点进行脉冲输出时，OnTime 和 CycleTime 设定时间不是特别精准，如果需要 OnTime 和 CycleTime 精准的场所，不建议使用该指令。
- 本指令输出可以使用变量或任意位装置。
- 示例程序和时序图如下所示：

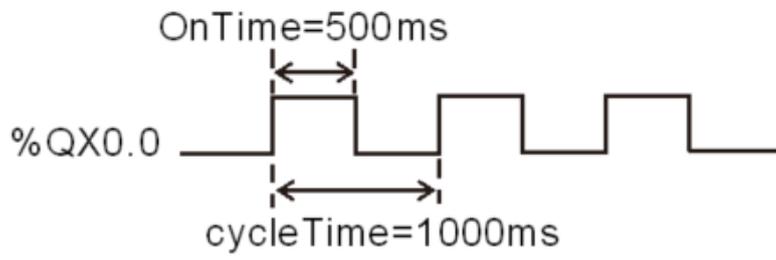
#### 梯形图(LD):



结构化文本(ST):

```
PWM_S0(Enable:=TRUE,OnTime:= onnTime, CycleTime:= cycletime ,Out=> output );
```

时序图:



## 2.1.2.4 PID (自整定PID)

PID(自整定PID)。所属库: Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
PID	自整定PID	FB		<pre> PID(     RUN:=,     SV:=,     PV:=,     Mode:=,     ManualCtrl:=,     Cycle:=,     Kp:=,     Ki:=,     Kd:=,     Tf:=,     DIR:=,     ERR_DBW:=,     MaxMV:=,     MinMV:=,     ManualMV:=,     FeedForward:=,     Option1:=,     Option2:=,     Option3:=,     Option4:=,     MV=&gt; ); </pre>

## ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Run	执行位	输入	执行位	由变量数据类型决定
SV	目标值		目标值	由变量数据类型决定
PV	当前值		当前值	由变量数据类型决定
Mode	模式		0: PID正常模式; 1: PID自整定模式, 自整定过程会计算出合适Kp,Ki,Kd,Tf等参数, 自动填入输入变量中。自整定流程结束会自动进入模式0, (正常工作模式)。	0,1
ManualCtrl	A/M模式		TRUE:手动模式 FALSE:自动模式	由变量数据类型决定
Cycle	采样周期		采样周期 (Ts)	1-4000ms
Kp	比例系数		比例系数	由变量数据类型决定
Ki	积分系数		积分系数	由变量数据类型决定
Kd	微分系数		微分系数	由变量数据类型决定
Tf	微分时间常数		微分时间常数	由变量数据类型决定
DIR	方向		TRUE:正向 (E=SV-PV) FALSE:反向 (E=PV-SV)	由变量数据类型决定
ERR_DBW	偏差忽略范围		偏差忽略范围 (E)	由变量数据类型决定
MaxMV	输出饱和上限		输出(MV)饱和上限	由变量数据类型决定
MinMV	输出饱和下限		输出(MV)饱和下限	由变量数据类型决定
ManualMV	MV手动值		MV手动值	由变量数据类型决定
FeedForward	前馈控制输出		前馈控制输出	由变量数据类型决定
Option1	保留	保留	由变量数据类型决定	

引脚	名称	输入/输出	描述	参数取值范围
Option2	保留		保留	由变量数据类型决定
Option3	保留		保留	由变量数据类型决定
Option4	保留		保留	由变量数据类型决定
MV	输出值	输出	输出值, 介于MaxMV和MinMV之间的数值	由变量数据类型决定

	布尔	位串					整数							实数		时间, 日期				字符串
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Run	<input type="radio"/>																			
SV														<input type="radio"/>						
PV														<input type="radio"/>						
Mode											<input type="radio"/>									
ManualCtrl	<input type="radio"/>																			
Cycle											<input type="radio"/>									
Kp														<input type="radio"/>						
Ki														<input type="radio"/>						
Kd														<input type="radio"/>						
Tf														<input type="radio"/>						
DIR	<input type="radio"/>																			
ERR_DBW														<input type="radio"/>						
MaxMV														<input type="radio"/>						
MinMV														<input type="radio"/>						
ManualMV														<input type="radio"/>						
FeedForward														<input type="radio"/>						
Option1	<input type="radio"/>																			
Option2	<input type="radio"/>																			
Option3	<input type="radio"/>																			
Option4	<input type="radio"/>																			
MV														<input type="radio"/>						

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

## ■ 功能说明

- 建议 PID 指令在固定周期任务中使用。一般该指令 Cycle 的设定值和固定周期的周期值相同。。
- 当 Mode 设置为 0 (正常模式) 时,用户手动调整 Kp、Ki 及 Kd 三个主要参数时, 请先将 Kp 值依经验值设定, Ki 及 Td 值先设定为 0, 待 Kp 值确定且系统运行相对平稳, 再由小到大调整 Ki、Kd 值。Kp 值为 1 则表示 100%, 即对偏差值的增益为 1, 小于 100%将对偏差值衰减, 大于 100%将对偏差值放大。
- 当 Mode 设置为 1 (自整定模式) 时, 建议建议将主要输入变量设置为停电保持属性, 以免自整定过的参数因掉电而消失。有些场合自整定参数可能并非最优参数, 使用者可适当调整, 通常建议只修改 Ki 或 Kd 数值即可。

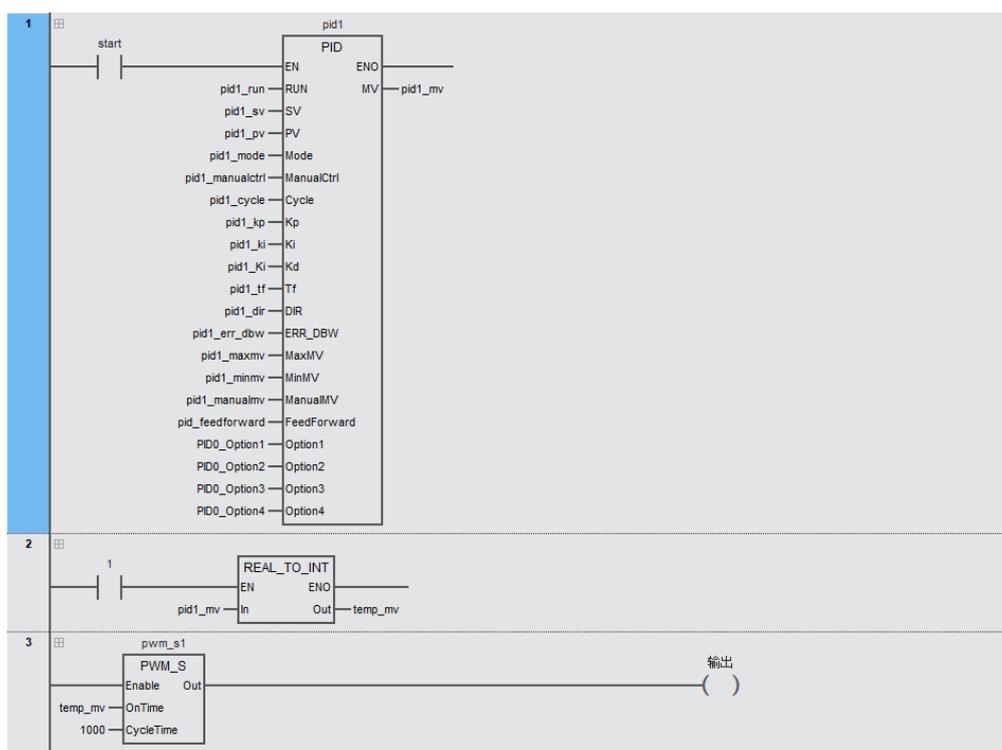
- 示例程序如下所示：

PID功能块调节温度。

变量表：

类别	名称	分配到	数据类型	初始值	注释
VAR	pid1		PID		
VAR	pid1_en		BOOL		
VAR	pid1_run		BOOL		
VAR	pid1_sv		REAL	180	
VAR	pid1_pv		REAL		
VAR	pid1_mode		DINT	1	
VAR	pid1_manualctrl		BOOL		
VAR	pid1_cycle		DINT	100	
VAR RETAIN	pid1_kp		REAL		
VAR RETAIN	pid1_ki		REAL		
VAR RETAIN	pid1_kd		REAL		
VAR RETAIN	pid1_tf		REAL		
VAR	pid1_dir		BOOL	1	
VAR	pid1_err_dbw		REAL		
VAR	pid1_maxmv		REAL	1000	
VAR	pid1_minmv		REAL	0	
VAR	pid1_manualmv		REAL		
VAR	pid1_feedforward		REAL		
VAR	pid1_mv		REAL		
VAR	pwm_s1		PWM_S		
VAR	temp_mv		REAL		
VAR	输出	%QX0.0	BOOL		

梯形图(LD)：



**结构化文本(ST):**

```
pid1(RUN:=pid1_en ,
      SV:= pid1_sv,
      PV:=pid1_pv ,
      Mode:=pid1_mode ,
      ManualCtrl:= pid1_manualctrl,
      Cycle:=pid1_cycle,
      Kp:=pid1_kp,
      Ki:=pid1_ki ,
      Kd:=pid1_kd,
      Tf:=pid1_tf ,
      DIR:=pid1_dir ,
      ERR_DBW:=pid1_err_dbw,
      MaxMV:=pid1_maxmv ,
      MinMV:=pid1_minmv ,
      ManualMV:=pid1_manualmv ,
      FeedForward:=pid_feedforward ,
      MV=>pid1_mv );
temp_mv:=REAL_TO_INT(pid1_mv);
pwm_s1(Enable:=TRUE ,OnTime:=pid1_mv, CycleTime:=1000 ,Out=>输出 );
```

**程序说明:**

将合适的Kp,Ki,Kd,Tf等参数填入到变量中，EN设置为TRUE，Mode参数设置为0，PID指令根据Mode,ManualCtrl,PV,SV,MaxMV,MinMV等参数的值开始工作，MV的输出值范围在MinMV-MaxMv之间。一般通过PWM\_S（脉冲宽度调整）指令控制控制器本体输出点做脉冲输出，如此来控制外部加热装置进行加热或冷却；PID指令MV的值一般作为PWM\_S指令的脉冲输出宽度参数（OnTime），PID指令的Cycle一般做为PWM\_S指令的脉冲输出周期参数（CycleTime）。如当PID指令MV=500，Cycle=1000时，则PWM\_S脉冲输出周期为1000ms，输出点ON的时间为500ms。

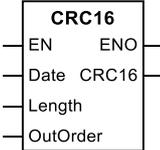
**△注意**

在使用PID指令时，一般先将MODE参数设置自整定模式（Mode=1），自整定过程会计算出合适Kp,Ki,Kd,Tf等参数，自动填入输出变量中。自整定流程结束会自动进入模式0（正常工作模式）。SV的值一般需要将温度模块的值进行计算转化为以°C为单位。

## 2.13 校验函数

### 2.13.1 CRC16(CRC16校验函数)

本指令用于计算1组数据的CRC16校验码。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
CRC16	CRC16校验函数	FUN		CRC16( Data:=, Length:=, OutOrder:= );

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Data	数据指针	输入	数据指针	-
Length	计算长度		计算长度(单位: 字节)	变量数据类型决定
OutOrder	校验码字节模式		FALSE:校验码高字节在前, 低字节在后 TRUE:校验码低字节在前, 高字节在后	变量数据类型决定
CRC16	校验码	输出	校验码	变量数据类型决定

	布尔		位串					整数					实数		时间, 日期			指针			
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	Pointer To BYTE	
Data																					○
Length							○														
OutOrder	○																				
CRC16			○																		

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

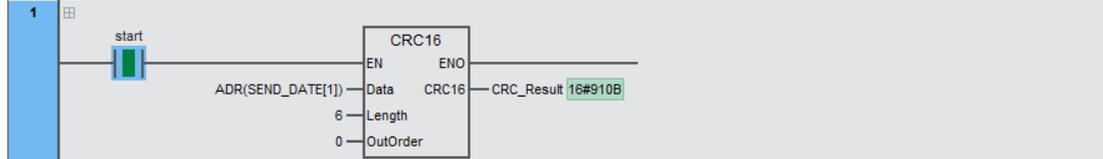
#### ■ 功能说明

- 本指令用于计算 1 组数据的 CRC16 校验码。
- 示例程序如下所示:

#### 梯形图(LD):

例如: 1组数据为{16#01,16#03,16#10,16#01,16#00,16#02}, 计算长度为6, 校验码为16#910B。

类别	名称	分配到	数据类型	在线值	准备值	注释
1 VAR	SEND_DATE		ARRAY[1..6] OF BYTE			
2 VAR	SEND_DATE[1]		USINT	16#01		
3 VAR	SEND_DATE[2]		USINT	16#03		
4 VAR	SEND_DATE[3]		USINT	16#10		
5 VAR	SEND_DATE[4]		USINT	16#01		
6 VAR	SEND_DATE[5]		USINT	16#00		
7 VAR	SEND_DATE[6]		USINT	16#02		
8 VAR	CRC_Result		WORD	16#910B		
9 VAR	start		BOOL	TRUE		

**结构化文本(ST):**

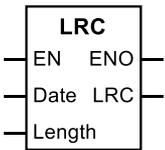
IF start THEN

    CRC\_Result:=CRC16(Data:=ADR(SEND\_DATE[1]),Length:=6,OutOrder:=0);

END\_IF;

## 2.13.2 LRC(LRC校验函数)

本指令用于计算1组数据的LRC校验码。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
LRC	LRC校验函数	FUN		LRC(Data:=,Length:=);

### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Data	数据指针	输入	数据指针	-
Length	计算长度		计算长度(单位: 字节)	变量数据类型决定
LRC	校验码	输出	校验码	变量数据类型决定

	布尔		位串					整数						实数		时间, 日期			指针	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	Pointer To BYTE
Data																				○
Length							○													
LRC			○																	

\*说明: 上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

- 本指令用于计算 1 组数据的 LRC 校验码。
- 示例程序如下所示:

#### 梯形图(LD):

例如: 1组数据为{16#01,16#03,16#10,16#01,16#00,16#02}, 计算长度为6, 校验码为16#E9。

类别	名称	分配到	数据类型	在线值	准备值	注释
1 VAR	SEND_DATE		ARRAY[1..6] OF BYTE			
2 VAR	SEND_DATE[1]		USINT	16#01		
3 VAR	SEND_DATE[2]		USINT	16#03		
4 VAR	SEND_DATE[3]		USINT	16#10		
5 VAR	SEND_DATE[4]		USINT	16#01		
6 VAR	SEND_DATE[5]		USINT	16#00		
7 VAR	SEND_DATE[6]		USINT	16#02		
8 VAR	LRC_Result		WORD	16#00E9		
9 VAR	start		BOOL	TRUE		


#### 结构化文本(ST):

```
IF start THEN
```

```
    LRC_Result:=CRC16(Data:=ADR(SEND_DATE[1]),Length:=6);
```

```
END_IF;
```

## 2.14 位字转换

### 2.14.1 GetBitofWord(读取变量中指定位的值)

读取位串或者整数类型变量中指定位置1个位 (bit) 的状态。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
GetBitofWord	读取位串或者整数类型变量中指定位的值	FUN		GetBitofWord:=GetBitofWord(StartDevice:=,Offset:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
StartDevice	参数指针	输入	参数指针	-
Offset	位偏移		位偏移	0-1023
GetBitofWord	位状态	输出	位状态	变量数据类型决定

	布尔		位串			整数							实数		时间, 日期			指针 Pointer To UINT			
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE		TOD	DT	
StartDevice																					○
Offset			○				○														
GetBitofWord	○																				

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 读取位串或者整数类型变量中指定位置 1 个位 (bit) 的状态。
- StartDevice 为 Pointer to UINT，因此一般变量不能直接作为输入变量，需要通过 ADR 转换。详细可参考范例。
- Offset 为 Bit 偏移序号，0 为 Bit 0,1 为 Bit1，以此类推。

- 示例程序如下所示：

#### 梯形图(LD):

范例如下：读取“参数”变量中bit1的状态。当“参数”值为2，“偏移”值为1时，该指令执行后，“Bit状态”值为TRUE。当“参数”值为0，“偏移”值为1时，该指令执行后，“Bit状态”值为FALSE。



#### 结构化文本(ST):

IF 执行 THEN

    Bit状态:=GetBitofWord(StartDevice:=ADR(参数),Offset:=偏移);

END\_IF;

当需要读取更大位状态的值时，如“offset”的值超过15时，可以将范例中“参数”的数据类型设置为ARRAY[1..10] OF UINT，“StartDevice”处填写ADR(参数[1])，通过指定“offset”的值，读取数组成员中指定定位的值。

### 2.14.2 SetBitofWord(设定变量中的指定位的值)

设定位串或者整数类型变量中指定位置1个位 (bit) 的值。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
SetBitofWord	设定位串或者整数类型变量中指定位的值	FUN		SetBitOfWord(StartDevice:=,Offset:=,Value:=);

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
StartDevice	参数指针	输入	参数指针	变量数据类型决定
Offset	位偏移		位偏移	变量数据类型决定
Value	设定状态		设定状态	变量数据类型决定

	布尔		位串					整数							实数		时间, 日期				指针
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	Pointer To UINT	
StartDevice																					○
Offset			○				○														
Value	○																				

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

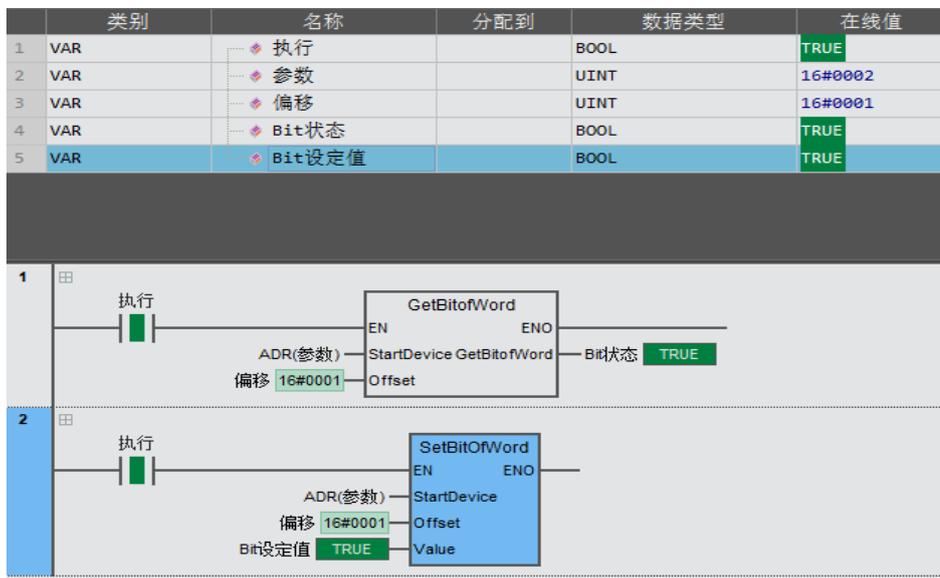
#### ■ 功能说明

- 设定位串或者整数类型变量中指定位置 1 个位 (bit) 的值，设定位 TRUE 或者 FALSE，设定值通过输入参数 Value 指定。
- StartDevice 为 Pointer to UINT，因此一般变量不能直接作为输入变量，需要通过 ADR 转换。详细可参考范例。
- Offset 为 Bit 偏移序号，0 为 Bit0，1 为 Bit1，以此类推。

- 示例程序如下所示：

### 梯形图(LD):

范例：设置“参数”变量中bit1的值，并读取“参数”变量中bit1的状态。当“偏移”值为1，“Bit设定值”为TRUE时，该指令执行后，“参数”的bit1为TRUE；当“偏移”值为1，“Bit设定值”为FALSE时，该指令执行后，“参数”的bit1为FALSE。



### 结构化文本(ST):

IF 执行 THEN

    Bit状态:=GetBitofWord(StartDevice:=ADR(参数),Offset:=偏移);

END\_IF;

IF 执行 THEN

    SetBitOfWord(StartDevice:= ADR(参数),Offset:=偏移,Value:=Bit设定值);

END\_IF;

当需要设置更大位的值时，如“offset”的值超过15时，可以将范例中“参数”的数据类型设置为ARRAY[1..10] OF UINT，“StartDevice”处填写ADR(参数[1])，通过指定“offset”的值，设置数组成员中指定位的值。

## 2.15 扩展模块通讯

### 2.15.1 EXT\_ReadParameter(读取扩展模块参数)

本指令用于读取扩展模块的参数值。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
EXT_ReadParameter	读取扩展模块参数	FB		<pre>EXT_ReadParameter_instance (   ModularID:=,   Excecute:=,   ParameterIndex:=,   Done=&gt;,   Busy=&gt;,   Active=&gt;,   Error=&gt;,   ErrorID=&gt;,   Value=&gt; );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
ModularID	模块序号	输入	模块序号	1-32
Excecute	执行位		执行位	TRUE / FALSE
ParameterIndex	参数编号		模块参数对应的编号	模块参数支持范围决定
Done	完成位	输出	完成位	TRUE / FALSE
Busy	执行状态位		执行状态位	TRUE / FALSE
Active	激活位		激活位	TRUE / FALSE
Error	错误位		错误位	TRUE / FALSE
ErrorID	错误码		错误码	0~65535
Value	读取数值		读取数值	由变量数据类型决定

	布尔	位串					整数						实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
ModularID						○														
Excecute	○																			
Index						○														
Done	○																			
Busy	○																			
Active	○																			
Error	○																			
ErrorID			○																	
Value			○																	

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

## ■ 错误代码说明

当指令执行出错时，指令ErrorID（错误代码）有对应的值，通过下表查询ErrorID值的含义及处理方法。

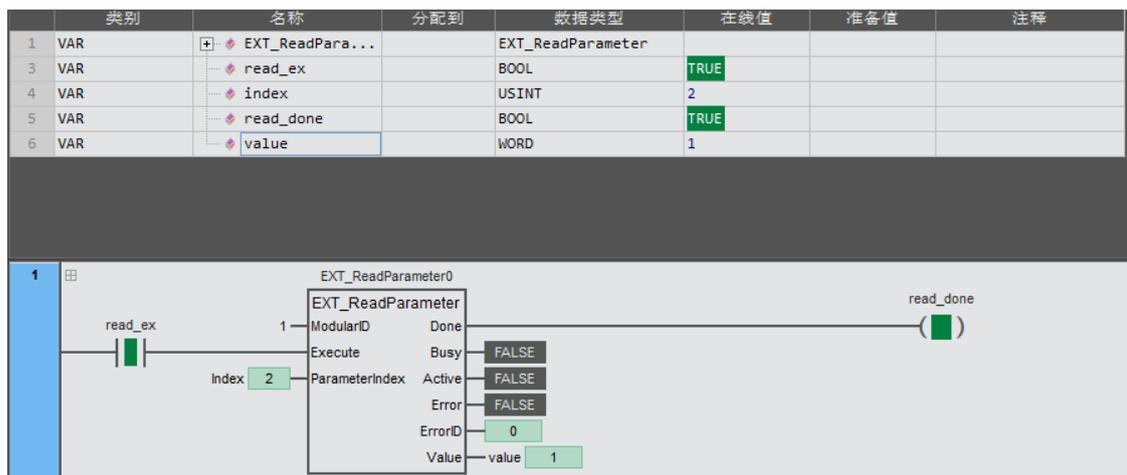
错误代码		ErrorID值含义	处理方法
十六进制	十进制		
4000	16384	扩展模块编号超出范围	将扩展模块编号(ModularID)设置在允许范围内
4001	16385	主机与扩展模块通讯超时	检查并确保主机与扩展模块连接正常
4002	16386	主机对模块参数读或写异常	1.检查模块参数编号（ParameterIndex）是否存在 2.检查并确保主机与扩展模块连接正常 3.避免将产品置于强干扰的环境中

## ■ 功能说明

- 本指令用于读取扩展模块的参数值。
- 控制器右侧第一台模块编号为 1，右侧第二台模块编号为 2，其他依次类推。
- 参数编号含义请参阅具体模块手册。
- 示例程序如下所示：

### 梯形图(LD):

下图范例为读取控制器右侧第一台模块、参数编号为2的参数值，读到的数值为1。



### 结构化文本(ST):

```
EXT_ReadParameter0 (
  ModularID:=1,
  Execute:= read_ex,
  ParameterIndex:= Index,
  Done=> read_done,
  Value=> value
);
```

### 2.15.2 EXT\_WriteParameter (设定扩展模块参数)

本指令用于设定扩展模块的参数值。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
EXT_WriteParameter	设定扩展模块参数	FB		<pre>EXT_WriteParameter_instance (     ModularID:= ,     Excecute:= ,     ParameterIndex:= ,     Value:= ,     Done=&gt; ,     Busy=&gt; ,     Active=&gt; ,     Error=&gt; ,     ErrorID=&gt; );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
ModularID	模块序号	输入	模块序号	1-32
Excecute	执行位		执行位	TRUE / FALSE
ParameterIndex	参数编号		模块参数对应的编号	模块参数支持范围决定
Value	设定数值		读取数值	变量数据类型决定
Done	完成位	输出	完成位	TRUE / FALSE
Busy	执行状态位		执行状态位	TRUE / FALSE
Active	激活位		激活位	TRUE / FALSE
Error	错误位		错误位	TRUE / FALSE
ErrorID	错误代码		错误代码	0~65535

	布尔	位串				整数							实数		时间, 日期			字符串			
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
ModularID						○															
Excecute	○																				
Index						○															
Value			○																		
Done	○																				
Busy	○																				
Active	○																				
Error	○																				
ErrorID			○																		

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

## ■ 错误代码说明

当指令执行出错时，指令ErrorID（错误代码）有对应的值，通过下表查询ErrorID值的含义及处理方法。

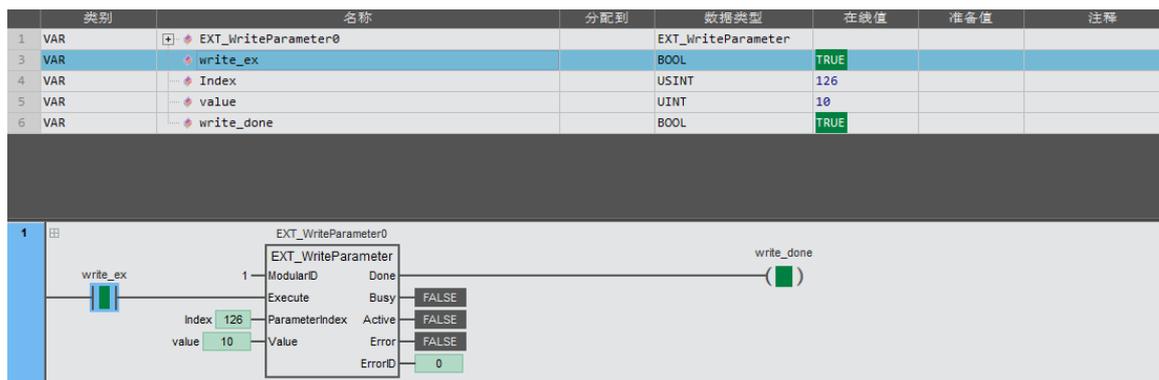
错误代码		ErrorID值含义	处理方法
十六进制	十进制		
4000	16384	扩展模块编号超出范围	将扩展模块编号(参数ModularID)设置在允许范围内
4001	16385	主机与扩展模块通讯超时	检查并确保主机与扩展模块连接正常
4002	16386	主机对模块参数读或写异常	1.检查模块参数编号（ParameterIndex）是否存在 2.Value(设定参数值)是否超过模块参数允许的范围 3.检查并确保主机与扩展模块连接正常 4.避免将产品置于强干扰的环境中

## ■ 功能说明

- 本指令用于设定扩展模块的参数。
- 控制器右侧第一台模块序号为 1，右侧第二台模块序号为 2，其他依次类推。
- 参数编号含义请参阅具体模块手册。
- 示例程序如下所示：

### 梯形图(LD)：

下图范例为设定控制器右侧第一台模块、参数编号为126的参数值，设定数值为10。



### 结构化文本(ST)：

```
EXT_WriteParameter0(
  ModularID:=1,
  Execute:= write_ex ,
  ParameterIndex:= Index,
  Value:= value ,
  Done=> write_done
);
```

## 2.16 系统功能

### 2.16.1 SYS\_GetTotalWorkTime (累计通电时间)

本指令用于读取控制器累计通电时间。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
SYS_GetTotalWorkTime	控制器累计通电时间	FB		<pre>SYS_GetTotalWorkTime_instance( Enable:=, Valid=&gt;, WorkTime=&gt; );</pre>

#### ■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Enable	执行位	输入	执行位	变量数据类型决定
Vaild	执行状态	输出	执行状态	变量数据类型决定
WorkTime	累计上电时间	输出	上电时间	变量数据类型决定

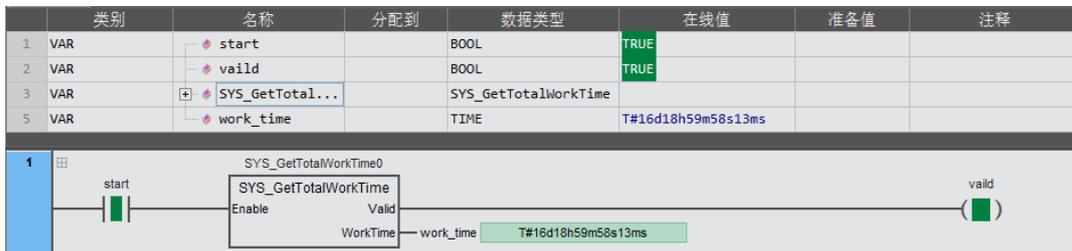
	布尔	位串					整数							实数		时间, 日期			字符串		
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
Enable	○																				
Vaild	○																				
WorkTime																○					

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

#### ■ 功能说明

- 本指令用于获取控制器自制造厂出厂后，累计工作时间。
- 示例程序如下所示：

梯形图(LD)：



结构化文本(ST)：

```
SYS_GetTotalWorkTime0(Enable:=start ,Valid=>vaild ,WorkTime=>work_time );
```

## 2.16.2 SYS\_GetWorkTime (单次通电时间)

本指令用于读取控制器每次通电后的通电时间。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
SYS_GetWorkTime	控制器每次通电后的通电时间	FB		SYS_GetWorkTime_instance (Enable:=, Valid=>, WorkTime=>);

### 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Enable	执行位	输入	执行位	变量数据类型决定
Vaild	执行状态	输出	执行状态	变量数据类型决定
WorkTime	上电时间	输出	上电时间	变量数据类型决定

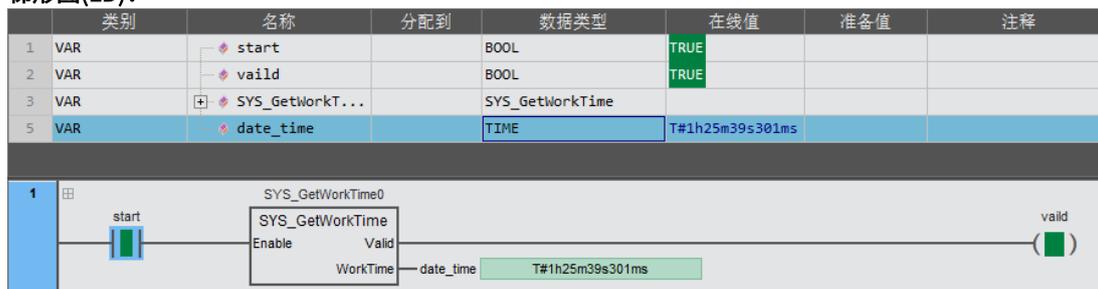
	布尔		位串					整数						实数		时间, 日期				字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	
Enable	○																				
Vaild	○																				
WorkTime															○						

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

### 功能说明

- 本指令用于获取控制器单次上电工作时间。该时间在控制器每次上电后从 0 开始计时。
- 示例程序如下所示：

#### 梯形图(LD):



#### 结构化文本(ST):

```
SYS_GetWorkTime0 (Enable:=start ,Valid=>vaild ,WorkTime=>date_time );
```

### 2.16.3 SYS\_GetRTCTime (读取实时时钟)

本指令用于读取控制器实时时钟时间。所属库：Standard\_Part2

指令	名称	FB/FUN	梯形图样式	ST样式
SYS_GetRTCTime	读取控制器实时时钟时间	FB		<pre> SYS_GetRTCTime_instance(   Enable:= ,   Valid=&gt; ,   Year=&gt; ,   Month=&gt; ,   Day=&gt; ,   Week=&gt; ,   Hour=&gt; ,   Minute=&gt; ,   Second=&gt; );                     </pre>

■ 输入、输出变量说明及数据类型

引脚	名称	输入/输出	描述	参数取值范围
Enable	执行位	输入	执行位	由变量数据类型决定
Valid	执行状态	输出	执行状态	由变量数据类型决定
Year	年	输出	年	1970-2106
Month	月	输出	月	1-12
Day	日	输出	日	1-31
Week	星期	输出	周	1-7
Hour	时	输出	时	0-23
Minute	分	输出	分	0-59
Second	秒	输出	秒	0-59

	布尔		位串				整数							实数		时间, 日期			字符串	
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	○																			
Valid	○																			
Year							○													
Month							○													
Day							○													
Week							○													
Hour							○													
Minute							○													
Second							○													

\*说明：上表中的“○”表示指令参数允许与该数据类型的变量或常量连接。

■ 功能说明

- 本指令用于获取控制器本身实时时钟时间，请注意实时时钟由电池或本体电容储电才能继续计时。

- 示例程序如下所示：

### 梯形图(LD):



### 结构化文本(ST):

```

SYS_GetRTCTime0 (
  Enable:=start ,
  Valid=>vaild ,
  Year=>year ,
  Month=>month ,
  Day=>day ,
  Week=>week ,
  Hour=>hour ,
  Minute=>minute ,
  Second=>second );

```



禾川科技HCFA



禾川自动化中心ATC

## 浙江禾川科技股份有限公司

浙江省衢州市龙游县工业园区阜财路9号

## 杭州研发中心

浙江省杭州市余杭区衢州海创园D座4楼

 **400热线电话-400-012-6969**

 **禾川官网网址-[www.hcfa.cn](http://www.hcfa.cn)**

本手册中记载的其它产品,产品名称以及产品的商标或注册商标归各公司所有,并非本公司产品;  
本手册中所有信息如有变更,恕不另行通知。