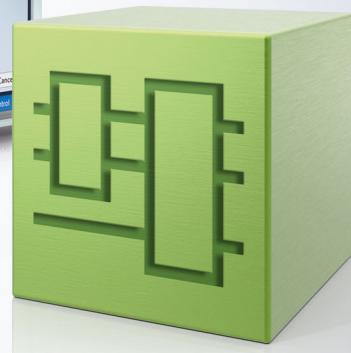
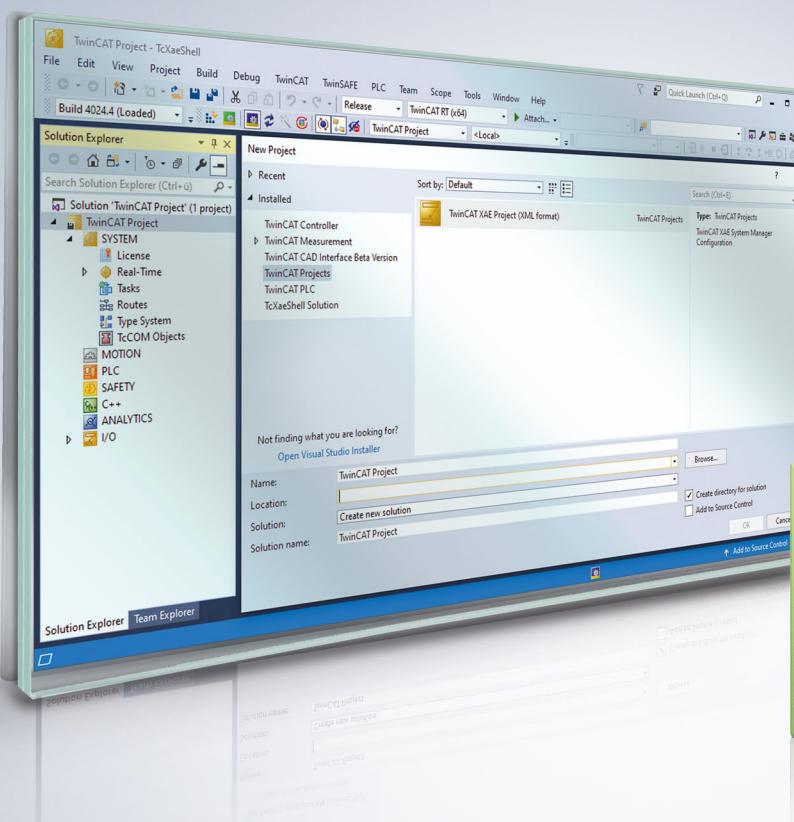


手册 | ZH  
TE1000  
TwinCAT 3 | PLC Library: Tc2\_Standard





# 目录

<b>1 前言</b>	5
1.1 文档说明	5
1.2 安全信息	5
1.3 信息安全说明	6
<b>2 概述</b>	7
<b>3 功能块</b>	8
3.1 双稳态	8
3.1.1 RS	8
3.1.2 SR	8
3.2 计数器	9
3.2.1 CTD	9
3.2.2 CTU	10
3.2.3 CTUD	11
3.3 定时器	11
3.3.1 TOF	11
3.3.2 TON	12
3.3.3 TP	13
3.4 定时器 (LTIME)	14
3.4.1 LTOF	14
3.4.2 LTON	15
3.4.3 LTP	16
3.5 Trigger (触发器)	17
3.5.1 F_TRIGGER	17
3.5.2 R_TRIGGER	18
<b>4 字符串函数</b>	19
4.1 CONCAT	19
4.2 DELETE	19
4.3 FIND	20
4.4 INSERT	20
4.5 LEFT	21
4.6 LEN	21
4.7 MID	22
4.8 REPLACE	22
4.9 RIGHT	23
<b>5 字符串函数 (WSTRING)</b>	24
5.1 WCONCAT	24
5.2 WDELETE	24
5.3 WFIND	25
5.4 WINSET	25
5.5 WLEFT	26
5.6 WLLEN	26

5.7	WMID.....	26
5.8	WREPLACE.....	27
5.9	WRIGHT.....	27
<b>6</b>	<b>全局常量.....</b>	<b>29</b>
6.1	库版本.....	29

# 1 前言

## 1.1 文档说明

本说明仅适用于熟悉国家标准且经过培训的控制和自动化工程专家。

在安装和调试组件时，必须遵循文档和以下说明及解释。

操作人员应具备相关资质，并始终使用最新的生效文档。

相关负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

### 免责声明

本文档经过精心准备。然而，所述产品正在不断开发中。

我们保留随时修改和更改本文档的权利，恕不另行通知。

不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

### 商标

Beckhoff®、ATRO®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、MX-System®、Safety over EtherCAT®、TC/BSD®、TwinCAT®、TwinCAT/BSD®、TwinSAFE®、XFC®、XPlanar® 和 XTS® 是 Beckhoff Automation GmbH

的注册商标并由其授权使用。本出版物中所使用的其它名称可能是商标名称，任何第三方出于其自身目的使用它们可能会侵犯商标所有者的权利。



EtherCAT®是注册商标和专利技术，由 Beckhoff Automation GmbH 授权使用。

### 版权所有

© Beckhoff Automation GmbH。

未经明确授权，不得复制、分发、使用和传播本文档内容。

违者将被追究赔偿责任。Beckhoff Automation GmbH 保留所有发明、实用新型和外观设计专利权。

### 第三方商标

本文档可能使用了第三方商标。有关商标信息，可以访问：<https://www.beckhoff.com/trademarks>。

## 1.2 安全信息

### 安全规范

为了确保您的使用安全，请务必仔细阅读  
并遵守本文档中每个产品的安全使用说明。

### 责任免除

所有组件在供货时都配有适合应用的特定硬件和软件配置。严禁未按文档所述修改硬件或软件配置，否则，德国倍福自动化有限公司对由此产生的后果不承担责任。

### 人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

### 警示性词语

文档中使用的警示信号词分类如下。为避免人身伤害和财产损失，请阅读并遵守安全和警告注意事项。

**人身伤害警告****⚠ 危险**

存在死亡或重伤的高度风险。

**⚠ 警告**

存在死亡或重伤的中度风险。

**⚠ 谨慎**

存在可能导致中度或轻度伤害的低度风险。

**财产或环境损害警告****注意**

可能会损坏环境、设备或数据。

**操作产品信息**

这些信息包括：  
有关产品的操作、帮助或进一步信息的建议。

## 1.3 信息安全说明

Beckhoff Automation GmbH & Co.KG (简称 Beckhoff) 的产品，只要可以在线访问，都配备了安全功能，支持工厂、系统、机器和网络的安全运行。尽管配备了安全功能，但为了保护相应的工厂、系统、机器和网络免受网络威胁，必须建立、实施和不断更新整个操作安全概念。Beckhoff 所销售的产品只是整个安全概念的一部分。客户有责任防止第三方未经授权访问其设备、系统、机器和网络。它们只有在采取了适当的保护措施的情况下，方可与公司网络或互联网连接。

此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 <https://www.beckhoff.com/secguide>。

Beckhoff 的产品和解决方案持续进行改进。这也适用于安全功能。鉴于持续进行改进，Beckhoff 明确建议始终保持产品的最新状态，并在产品更新可用后马上进行安装。使用过时的或不支持的产品版本可能会增加网络威胁的风险。

如需了解 Beckhoff 产品信息安全的信息，请订阅 <https://www.beckhoff.com/secinfo> 上的 RSS 源。

## 2 概述

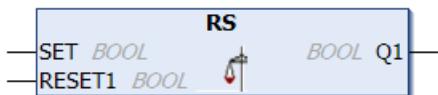
标准库包含所有 IEC61131-3 POU。POU 可以分为以下几类：

- 双稳态功能块
- 触发功能块
- 计数器
- 定时器
- 定时器 (LTIME)
- 字符串函数
- 字符串函数 (WSTRING)

### 3 功能块

#### 3.1 双稳态

##### 3.1.1 RS



功能块 RS 属于双稳态类型，RESET 具有优先权。

$Q1 = RS (SET, RESET1)$

即： $Q1 = \text{NOT RESET1 AND (Q1 OR SET)}$

##### 输入

```
VAR_INPUT
  SET      : BOOL;
  RESET1  : BOOL;
END_VAR
```

名称	类型	描述
SET	BOOL	设置输入
RESET1	BOOL	重置输入

##### 输出

```
VAR_OUTPUT
  Q1 : BOOL;
END_VAR
```

功能块的内部实现：

```
Q1: = NOT RESET1 AND (Q1 OR SET);
```

名称	类型	描述
Q1	BOOL	输出已重置。

##### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

#### 3.1.2 SR



功能块 SR 属于双稳态类型，SET 具有优先权。

$Q1 = SR (SET1, RESET)$

即： $Q1 = (\text{NOT RESET AND Q1}) \text{ OR SET1}$  Q1、SET1 和 RESET 均为 BOOL 变量。

### 输入

```
VAR_INPUT
  SET1 : BOOL;
  RESET : BOOL;
END_VAR
```

名称	类型	描述
SET1	BOOL	设置输入
RESET	BOOL	重置输入

### 输出

```
VAR_OUTPUT
  Q1 : BOOL;
END_VAR
```

功能块的内部实现：

```
Q1 := (NOT RESET AND Q1) OR SET1;
```

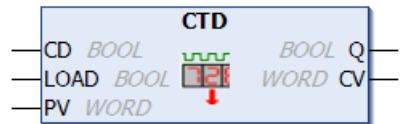
名称	类型	描述
Q1	BOOL	输出已设置。

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.2 计数器

### 3.2.1 CTD



功能块 CTD 是一个减计数器。

### 输入

```
VAR_INPUT
  CD : BOOL; (* Count Down on rising edge *)
  LOAD : BOOL; (* Load Start Value *)
  PV : WORD; (* Start Value *)
END_VAR
```

名称	类型	描述
CD	BOOL	在上升沿递减计数
LOAD	BOOL	加载起始值。
PV	WORD	起始值

### 输出

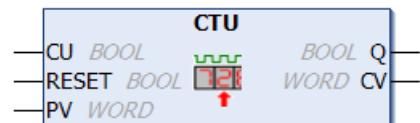
```
VAR_OUTPUT
  Q : BOOL; (* Counter reached 0 *)
  CV : WORD; (* Current Counter Value *)
END_VAR
```

名称	类型	描述
Q	BOOL	如果 LOAD = TRUE, 则计数器变量 CV 将被初始化为上限值 PV。如果 CD 有一个从 FALSE 到 TRUE 的上升沿, 那么, 只要 CV 大于 0, CV 就会递减 1。 (即, 如果不引起下溢的话)
CV	WORD	如果 CV 小于或等于 0, 则 Q 返回 TRUE。

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.2.2 CTU



功能块 CTU 是一个加计数器。

### 输入

```

VAR_INPUT
    CU : BOOL; (* Count Up *)
    RESET : BOOL; (* Reset Counter to 0 *)
    PV : WORD; (* Counter Limit *)
END_VAR

```

名称	类型	描述
CU	BOOL	递增计数
RESET	BOOL	将计数器重置为 0
PV	WORD	计数器限值

### 输出

```

VAR_OUTPUT
    Q : BOOL; (* Counter reached the Limit *)
    CV : WORD; (* Current Counter Value *)
END_VAR

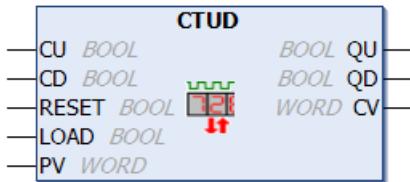
```

名称	类型	描述
Q	BOOL	如果 CV 大于或等于上限值 PV, 则 Q 返回 TRUE。
CV	WORD	如果 RESET = TRUE, 则计数器变量 CV 将被初始化为 0。如果 CU 有一个从 FALSE 到 TRUE 的上升沿, 那么, 只要 CV 小于 PV, 功能块 CV 就会递增 1。 (即, 如果不引起上溢的话)

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

### 3.2.3 CTUD



功能块 CTUD 是一个加/减计数器。

#### 输入

```
VAR_INPUT
  CU : BOOL; (* Count Up *)
  CD : BOOL; (* Count Down *)
  RESET : BOOL; (* Reset Counter to Null *)
  LOAD : BOOL; (* Load Start Value *)
  PV : WORD; (* Start Value / Counter Limit *)
END_VAR
```

名称	类型	描述
CU	BOOL	递增计数
CD	BOOL	递减计数
RESET	BOOL	将计数器重置为 0
LOAD	BOOL	加载起始值。
PV	WORD	起始值/计数器限值

#### 输出

```
VAR_OUTPUT
  QU : BOOL; (* Counter reached Limit *)
  QD : BOOL; (* Counter reached Null *)
  CV : WORD; (* Current Counter Value *)
END_VAR
```

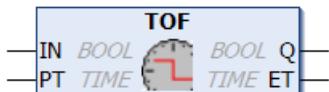
名称	类型	描述
QU	BOOL	如果 CV 已大于或等于 PV，则 QU 返回 TRUE。
QD	BOOL	如果 CV 已小于或等于 0，则 QD 返回 TRUE。
CV	WORD	如果 RESET 生效，则计数变量 CV 将被初始化为 0。如果 LOAD 生效，则 CV 将被初始化为 PV。如果 CU 有一个从 FALSE 到 TRUE 的上升沿，那么，只要 CV 不引起上溢，CV 就会递增 1。如果 CD 有一个从 FALSE 到 TRUE 的上升沿，那么，只要 CV 不引起下溢，CV 就会递减 1。

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

### 3.3 定时器

#### 3.3.1 TOF



功能块 TOF 是一个延时关闭定时器。

### 输入

```
VAR_INPUT
  IN : BOOL; (* starts timer with falling edge, resets timer with rising edge *)
  PT : TIME; (* time to pass, before Q is set *)
END_VAR
```

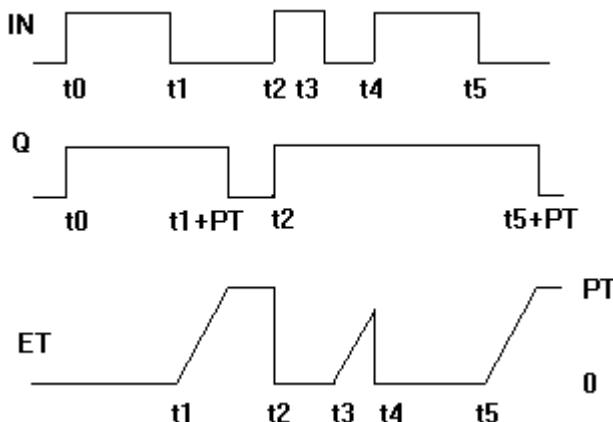
名称	类型	描述
IN	BOOL	通过下降沿启动定时器，通过上升沿重置定时器。
PT	TIME	在设置 Q 之前所经过的时间。

### 输出

```
VAR_OUTPUT
  Q : BOOL; (* is FALSE, PT seconds after IN had a falling edge *)
  ET : TIME; (* elapsed time *)
END_VAR
```

名称	类型	描述
Q	BOOL	当 PT 中设定的毫秒级时长耗尽时，则 Q 产生下降沿。
ET	BOOL	如果 IN = TRUE，则输出为 TRUE 或 0。当 IN = FALSE 时，ET 中的时间就会以毫秒为单位递增，直到该值达到 PT 后保持不变。如果 IN = FALSE 且 ET = PT，则 Q 为 FALSE。否则 Q = TRUE。

随时间变化的 TOF 图：

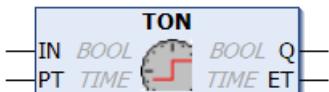


TOF 功能块占用 15 字节的数据存储空间

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.3.2 TON



功能块 TON 是一个延时启动定时器。

### 输入

```
VAR_INPUT
  IN : BOOL; (* starts timer with rising edge, resets timer with falling edge *)
  PT : TIME; (* time to pass, before Q is set *)
END_VAR
```

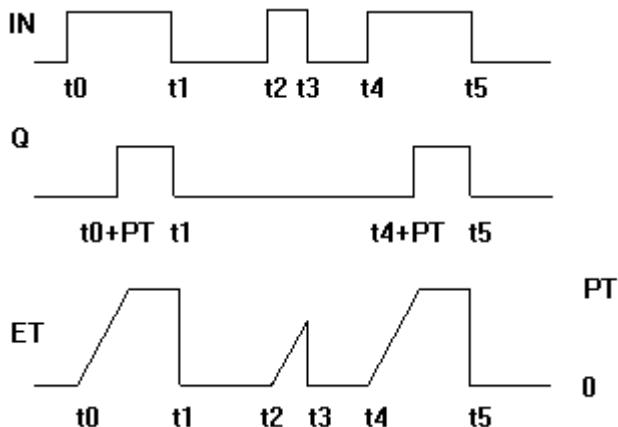
名称	类型	描述
IN	BOOL	通过上升沿启动定时器，通过下降沿重置定时器。
PT	TIME	在设置 Q 之前所经过的时间。

### 输出

```
VAR_OUTPUT
  Q : BOOL; (* is TRUE, PT seconds after IN had a rising edge *)
  ET : TIME; (* elapsed time *)
END_VAR
```

名称	类型	描述
Q	BOOL	当 PT 中设定的毫秒级时间耗尽时，则 Q 产生上升沿。
ET	TIME	如果 IN = FALSE，则输出为 FALSE 或 0。当 IN = TRUE 时，ET 中的时间就会以毫秒为单位递增，直到该值达到 PT 后保持不变。如果 IN = TRUE 且 ET = PT，则 Q 为 TRUE。否则 Q = FALSE。

随时间变化的 TON 图：

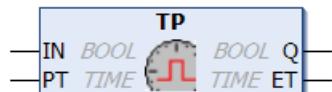


TON 功能块需要 15 字节的数据存储空间

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.3.3 TP



功能块 TP 是一个脉冲发生器，可用于生成具有规定脉冲持续时间的脉冲。

### 输入

```
VAR_INPUT
  IN : BOOL; (* Trigger for Start of the Signal *)
  PT : TIME; (* The length of the High-Signal in ms *)
END_VAR
```

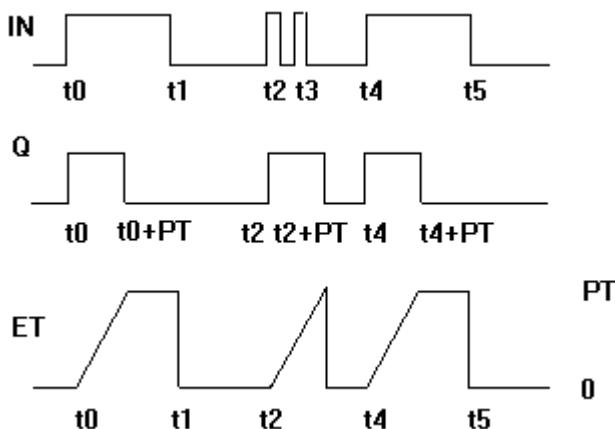
名称	类型	描述
IN	BOOL	信号启动触发器
PT	TIME	高电平信号的长度 (单位: ms)

### ▶ 输出

```
VAR_OUTPUT
  Q : BOOL; (* The pulse *)
  ET : TIME; (* The current phase of the High-Signal *)
END_VAR
```

名称	类型	描述
Q	BOOL	Q 在 PT 指定的时间间隔内提供信号。
ET	TIME	如果 IN = FALSE，则输出为 FALSE 或 0。当 IN = TRUE 时，Q = TRUE，并在脉冲持续时间 PT 内保持 TRUE。只要 Q = TRUE，ET 中的时间就会以毫秒为单位递增，直到该值达到 PT 后保持不变。在脉冲持续时间结束之前，无论输入 IN 的状态如何，输出 Q 始终为 TRUE。

随时间变化的 TP 图：



TP 功能块需要 14 字节的数据存储空间。

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.4 定时器 ( LTIME )

### 3.4.1 LTOF



功能块 LTOF 是一个采用 64 位时间数据类型 (LTIME) 的延时关闭定时器。

### ▶ 输入

```
VAR_INPUT
  IN : BOOL; (*starts timer with falling edge, resets timer with rising edge*)
  PT : LTIME; (*time to pass before Q is reset*)
END_VAR
```

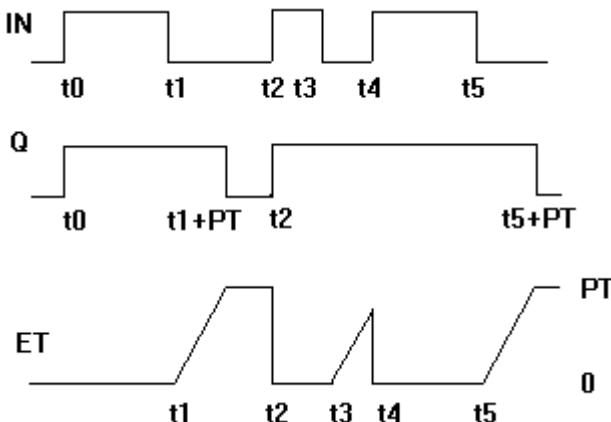
名称	类型	描述
IN	BOOL	通过下降沿启动定时器，通过上升沿重置定时器。
PT	TIME	在设置 Q 之前所经过的时间。

### ▶ 输出

```
VAR_OUTPUT
  Q : BOOL; (*is FALSE, PT seconds after IN had a falling edge*)
  ET : LTIME; (*elapsed time since falling edge at IN*)
END_VAR
```

名称	类型	描述
Q	BOOL	当 PT 中指定的纳秒级时间耗尽时，则 Q 产生下降沿。
ET	LTIME	如果 IN = TRUE，则输出为 TRUE 或 0。当 IN = FALSE 时，ET 中的时间就会以纳秒为单位递增，直到该值达到 PT。然后，该值保持不变。如果 IN = FALSE 且 ET = PT，则 Q 为 FALSE。

随时间变化的 LTOF 图：



### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.4.2 LTON



功能块 LTON 是一个采用 64 位时间数据类型 (LTIME) 的延时启动定时器。

### ▶ 输入

```
VAR_INPUT
  IN : BOOL; (*starts imter with rising edge, resets timer with falling edge*)
  PT : LTIME; (*time to pass before Q is set.*)
END_VAR
```

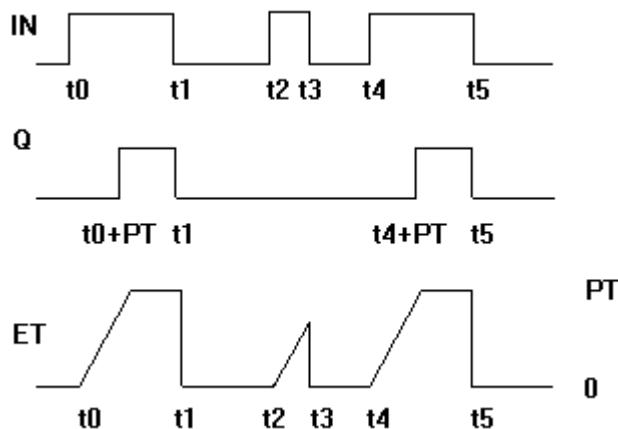
名称	类型	描述
IN	BOOL	通过上升沿启动定时器，通过下降沿重置定时器。
PT	TIME	在设置 Q 之前所经过的时间。

### ▶ 输出

```
VAR_OUTPUT
  Q : BOOL; (*is TRUE, PT seconds after IN had a rising edge*)
  ET : LTIME (*elapsed time since rising edge at IN*)
END_VAR
```

名称	类型	描述
Q	BOOL	当 PT 中指定的纳秒级时长耗尽时，则 Q 产生上升沿。
ET	LTIME	如果 IN = FALSE，则输出为 FALSE 或 0。当 IN = TRUE 时，ET 中的时间就会以纳秒为单位递增，直到该值达到 PT 后保持不变。如果 IN = TRUE 且 ET = PT，则 Q 为 TRUE。否则 Q = FALSE。

随时间变化的 LTON 图：



#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

### 3.4.3 LTP



功能块 LTP 是一个具有 64 位时间数据类型 (LTIME) 的脉冲发生器。该功能块可用于生成具有规定脉冲持续时间的脉冲。

#### 输入

```
VAR_INPUT
  IN : BOOL; (*Trigger for Start of the Signal*)
  PT : LTIME; (*The length of the High- Signal*)
END_VAR
```

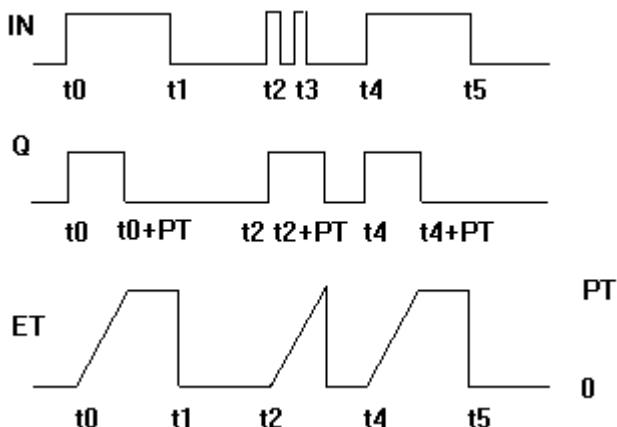
名称	类型	描述
IN	BOOL	信号启动触发器
PT	LTIME	高电平信号的长度 (单位: ms)

#### 输出

```
VAR_OUTPUT
  Q : BOOL; (*The pulse*)
  ET : LTIME (*elapsed time since pulse start*)
END_VAR
```

名称	类型	描述
Q	BOOL	Q 在 PT 指定的时间间隔内提供信号。
ET	LTIME	如果 IN = FALSE，则输出为 FALSE 或 0。当 IN = TRUE 时，Q = TRUE，并在脉冲持续时间 PT 内保持 TRUE。只要 Q = TRUE，ET 中的时间就会以纳秒为单位递增，直到该值达到 PT 后保持不变。在脉冲持续时间结束之前，无论输入 IN 的状态如何，输出 Q 始终为 TRUE。

随时间变化的 LTP 图：



#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.5 Trigger ( 触发器 )

### 3.5.1 F\_TRIG



功能块 F\_TRIG 是一个下降沿检测器。

#### 输入

```
VAR_INPUT
    CLK : BOOL; (* Signal to detect *)
END_VAR
```

名称	类型	描述
CLK	BOOL	待检测信号

#### 输出

```
VAR_OUTPUT
    Q : BOOL; (* Edge detected *)
END_VAR
```

名称	类型	描述
Q	BOOL	若输入变量 CLK = TRUE，则输出 Q = FALSE。当 CLK = FALSE 时，Q = TRUE。这意味着每次调用该函数时，Q 都会返回 FALSE，直到 CLK 产生上升沿且随后出现下降沿。

## 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 3.5.2 R\_TRIG



功能块 R\_TRIG 是一个上升沿检测器。

## 输入

```
VAR_INPUT
    CLK : BOOL; (* Signal to detect *)
END_VAR
```

名称	类型	描述
CLK	BOOL	待检测信号

## 输出

```
VAR_OUTPUT
    Q : BOOL; (* Edge detected *)
END_VAR
```

名称	类型	描述
Q	BOOL	若输入变量 CLK = FALSE，则输出 Q = FALSE。当 CLK = TRUE 时，Q 也将返回 TRUE。这意味着每次调用该函数时，Q 都会返回 FALSE，直到 CLK 产生下降沿且随后出现上升沿。

## 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4 字符串函数

在 [PLC示例](#)页面上，您可以下载有关字符串函数的示例项目。

### 4.1 CONCAT



两个字符串的串联（组合）。

#### FUNCTION CONCAT: STRING (255)

```
VAR_INPUT
  STR1 : STRING(255);
  STR2 : STRING(255);
END_VAR
```

IL示例：

```
LD 'SUSI'
CONCAT 'WILLI'
ST Var1 (* Result is 'SUSIWILLI' *)
```

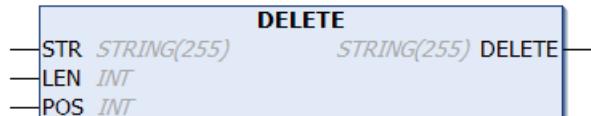
ST中的示例：

```
Var1 := CONCAT ('SUSI', 'WILLI');
```

#### 要求

开发环境	目标平台	待集成的PLC库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

### 4.2 DELETE



函数 DELETE 用于移除一个长字符串指定位置上的子字符串。输入变量 STR 为 STRING 类型，LEN 和 POS 为 INT 类型，函数的返回值为 STRING 类型。

DELETE (STR, LEN, POS) 指：在 STR 字符串中，从位置 POS 对应的字符开始，删除长度为 LEN 的字符。

#### FUNCTION DELETE: STRING (255)

```
VAR_INPUT
  STR : STRING(255);
  LEN : INT;
  POS : INT;
END_VAR
```

IL示例：

```
LD 'SUXYSI'
DELETE 2,3
ST Var1 (* Result is 'SUSI' *)
```

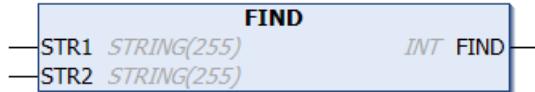
ST中的示例：

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

#### 要求

开发环境	目标平台	待集成的PLC库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.3 FIND



函数 FIND 用于在字符串中搜索子字符串。

FIND (STR1, STR2) 指：在 STR1 字符串中，查找第一个 STR2 的第一个字符的位置。如果在 STR1 中没有找到 STR2，则 OUT:=0。

### FUNCTION FIND: INT

```
VAR_INPUT
  STR1 : STRING(255);
  STR2 : STRING(255);
END_VAR
```

IL 示例：

```
LD 'SUXYSI'
FIND 'XY'
ST Var1 (* Result is 3 *)
```

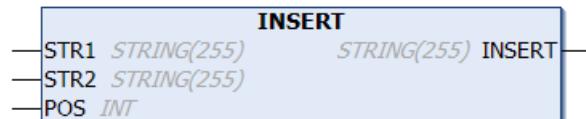
ST 中的示例：

```
Var1 := FIND('SUXYSI', 'XY');
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.4 INSERT



函数 INSERT 在一个指定点将一个字符串插入另一个字符串。

INSERT (STR1, STR2, POS) 指：在 STR1 字符串 POS 位置后，插入 STR2。

### FUNCTION INSERT: STRING (255)

```
VAR_INPUT
  STR1 : STRING(255);
  STR2 : STRING(255);
  POS  : INT;
END_VAR
```

IL 示例：

```
LD 'SUSI'
INSERT 'XY', 2
ST Var1 (* Result is 'SUXYSI' *)
```

ST 中的示例：

```
Var1 := INSERT('SUSI', 'XY', 2);
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.5 LEFT



函数 LEFT 返回给定字符串的左侧起始字符串。

LEFT (STR, SIZE) 指：从字符串 STR 中截取左侧开始的 SIZE 个字符。

### FUNCTION LEFT: STRING (255)

```
VAR_INPUT
  -STR : STRING(255);
  SIZE : INT;
END_VAR
```

IL 示例：

```
LD 'SUSI'
LEFT 3
ST Var1 (* Result is 'SUS' *)
```

ST 中的示例：

```
Var1 := LEFT ('SUSI', 3);
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.6 LEN



函数 LEN 返回字符串的长度。

### FUNCTION LEN: INT

```
VAR_INPUT
  -STR : STRING(255);
END_VAR
```

IL 示例：

```
LD 'SUSI'
LEN
ST Var1 (* Result is 4 *)
```

ST 中的示例：

```
Var1 := LEN ('SUSI');
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.7 MID



函数 MID 从字符串中返回其子字符串。

MID (STR, LEN, POS) 指：从字符串 STR 的 POS 位置开始提取 LEN 个字符。

### FUNCTION MID: STRING (255)

```
VAR_INPUT
  STR : STRING(255);
  LEN : INT;
  POS : INT;
END_VAR
```

IL 示例：

```
LD 'SUSI'
MID 2,2
ST Var1 (* Result is 'US' *)
```

ST 中的示例：

```
Var1 := MID ('SUSI', 2, 2);
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.8 REPLACE



函数 REPLACE 用于将长字符串的某子字符串替换为第三个字符串。

REPLACE (STR1, STR2, L, P) 指：将 STR1 中从位置 P 开始、长度为 L 的字符串替换为 STR2。

### FUNCTION REPLACE: STRING (255)

```
VAR_INPUT
  STR1 : STRING(255);
  STR2 : STRING(255);
  L    : INT;
  P    : INT;
END_VAR
```

IL 示例：

```
LD 'SUXYSI'
REPLACE 'K', 2, 2
ST Var1 (* Result is 'SKYSI' *)
```

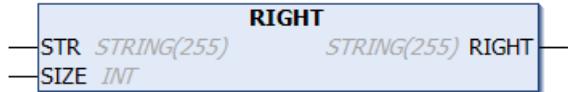
ST 中的示例：

```
Var1 := REPLACE ('SUXYSI', 'K', 2, 2);
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 4.9 RIGHT



RIGHT 函数返回字符串的右侧起始字符串。

RIGHT (STR, SIZE) 指：从字符串 STR 中截取右侧开始的 SIZE 个字符。

### FUNCTION RIGHT: STRING (255)

```
VAR_INPUT
  STR : STRING(255);
  SIZE : INT;
END_VAR
```

IL 示例：

```
LD 'SUSI'
RIGHT 3
ST Var1 (* Result is 'USI' *)
```

ST 中的示例：

```
Var1 := RIGHT ('SUSI', 3);
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5 字符串函数 ( WSTRING )

### 5.1 WCONCAT



两个 WSTRING 的串联 (组合)。

#### FUNCTION WCONCAT: WSTRING (255)

```
VAR_INPUT
  STR1 : WSTRING(255) (*Head part of the concatenated result*)
  STR2 : WSTRING(255) (*Tail part of the concatenated result*)
END_VAR
```

IL 示例:

```
LD "SUSI"
WCONCAT "WILLI"
ST Var1 (*Ergebnis ist "SUSIWILLI"*)
```

ST 中的示例:

```
Var1 := WCONCAT ("SUS", "WILLI");
```

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

### 5.2 WDELETE



函数 WDELETE 用于从指定位置删除 WSTRING 的部分内容。输入 STR 为 WSTRING 类型。LEN 和 POS 均为 INT 类型。函数的返回值为 WSTRING 类型。

WDELETE (STR, LEN, POS) 指: 从 STR 的第 POS 个字符开始删除 LEN 个字符。

#### FUNCTION WDELETE: WSTRING (255)

```
VAR_INPUT
  STR1 : WSTRING(255);
  LEN  : INT;
  POS  : INT;
END_VAR
```

IL 示例:

```
LD "SUXYSI"
WDELETE 2,3
ST Var1 (*Ergebnis ist "SUSI"*)
```

ST 中的示例:

```
Var1 := WDELETE ("SUXYSI", 2, 3);
```

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.3 WFIND



函数 WFIND 用于在 WSTRING 中查找某段字符串。

WFIND (STR1, STR2) 指：查找 STR1 中第一个 STR2 的第一个字符的位置。如果 STR2 在 STR1 中没有出现，则 OUT := 0 适用。

### FUNCTION WFIND: INT

```
VAR_INPUT
  STR1 : WSTRING(255);
  STR2 : WSTRING(255);
END_VAR
```

IL 示例：

```
LD "SUXYSI"
WFIND "XY"
ST Var1 (*Ergebnis ist 3*)
```

ST 中的示例：

```
Var1 := WFIND ("SUXYSI", "XY");
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.4 WINSERT



函数 WINSERT 用于在一个宽字符串的指定位置添加另一个宽字符串。

WINSERT (STR1, STR2, POS) 指：将 STR2 添加到 STR1 的 POS 位置之后。

### FUNCTION WINSERT: WSTRING (255)

```
VAR_INPUT
  STR1 : WSTRING(255);
  STR2 : WSTRING(255);
  POS  : INT;
END_VAR
```

IL 示例：

```
LD "SUSI"
WINSERT "XY", 2
ST Var1 (*Ergebnis ist "UXYSI")
```

ST 中的示例：

```
Var1 := WINSERT ("SUSI", "XY", 2);
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.5 WLEFT



函数 WLEFT 用于提取 WSTRING 左侧的起始字符串。

WLEFT (STR, SIZE) 指：从宽字符串 STR 中截取左侧开始的 SIZE 个字符。

### FUNCTION WLEFT: WSTRING (255)

```

VAR_INPUT
  STR : WSTRING(255);
  SIZE : INT;
END_VAR
  
```

IL 示例：

```

LD "SUSI"
WLEFT 3
ST Var1 (*Ergebnis ist "SUS")
  
```

ST 中的示例：

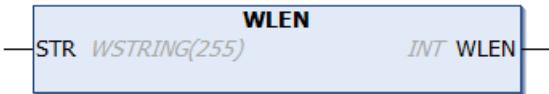
```

Var1 := WLEFT ("SUSI", 3);
  
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.6 WLEN



函数 WLEN 用于输出 WSTRING 的长度。

### FUNCTION WLEN: INT

```

VAR_INPUT
  STR : WSTRING(255);
END_VAR
  
```

IL 示例：

```

LD "SUSI"
WLEN
ST Var1 (*Ergebnis ist 4*)
  
```

ST 中的示例：

```

Var1 := WLEN ("SUSI");
  
```

### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.7 WMID



函数 WMID 用于提取 WSTRING 的子字符串。

WMID(STR, LEN, POS) 指：从宽字符串 STR 的 POS 位置开始提取 LEN 个字符。

#### FUNCTION WMID: WSTRING (255)

```
VAR_INPUT
  STR : WSTRING(255);
  LEN : INT;
  POS : INT;
END_VAR
```

IL 示例：

```
LD "SUSI"
WMID 2,2
ST Var1 (*Ergebnis ist "US")
```

ST 中的示例：

```
Var1 := WMID ("SUSI", 2, 2);
```

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.8 WREPLACE



函数 WREPLACE 用于将 WSTRING 的子字符串替换为另一个 WSTRING。

WREPLACE (STR1, STR2, L, P) 指：将 STR1 中从位置 P 开始、长度为 L 的字符串替换为 STR2。

#### FUNCTION WREPLACE: WSTRING (255)

```
VAR_INPUT
  STR1 : WSTRING(255);
  STR2 : WSTRING(255);
  L    : INT;
  P    : INT;
END_VAR
```

IL 示例：

```
LD "SUXYSI"
WREPLACE "XY",2
ST Var1 (*Ergebnis ist "SKYSI")
```

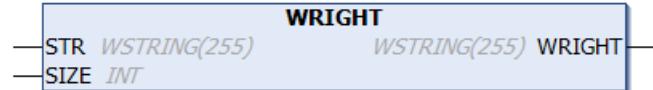
ST 中的示例：

```
Var1 := WREPLACE ("SUXYSI", "K", 2, 2);
```

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 5.9 WRIGHT



WRIGHT 用于 WSTRING 提取右侧的起始字符串。  
WRIGHT (STR, SIZE) 指：从宽字符串 STR 中截取右侧开始的 SIZE 个字符。

#### FUNCTION WRIGHT: WSTRING (255)

```
VAR_INPUT
  STR    : WSTRING(255);
  SIZE   : INT;
END_VAR
```

IL 示例：

```
LD "SUSI"
WRIGHT 3
ST Var1 (*Ergebnis ist "USI")
```

ST 中的示例：

```
Var1 := WRIGHT ("SUSI", 3);
```

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## 6 全局常量

### 6.1 库版本

所有库都有特定版本。该版本信息会显示在 PLC 库的资源库中。  
库版本号在全局常量中存储。

#### Global\_Version

```
VAR_GLOBAL CONSTANT
  stLibVersion_Tc2_Standard : ST_LibVersion;
END_VAR
```

名称	类型	描述
stLibVersion_Tc2_Standard	ST_LibVersion	Tc2_Standard 库的版本号

函数 F\_CmpLibVersion 用于比较现有版本和所需版本。



#### 与 TwinCAT 2 兼容

不再提供 TwinCAT 2 功能库的查询选项。

#### 要求

开发环境	目标平台	待集成的 PLC 库
TwinCAT v3.0.0	PC 或 CX (x86)	Tc2_Standard

## Trademark statements

Beckhoff<sup>®</sup>, ATRO<sup>®</sup>, EtherCAT<sup>®</sup>, EtherCAT G<sup>®</sup>, EtherCAT G10<sup>®</sup>, EtherCAT P<sup>®</sup>, MX-System<sup>®</sup>, Safety over EtherCAT<sup>®</sup>, TC/BSD<sup>®</sup>, TwinCAT<sup>®</sup>, TwinCAT/BS<sup>®</sup>, TwinSAFE<sup>®</sup>, XFC<sup>®</sup>, XPlanar<sup>®</sup> and XTS<sup>®</sup> are registered and licensed trademarks of Beckhoff Automation GmbH.

更多信息：  
[www.beckhoff.com/te1000](http://www.beckhoff.com/te1000)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
电话号码: +49 5246 9630  
info@beckhoff.com  
www.beckhoff.com

