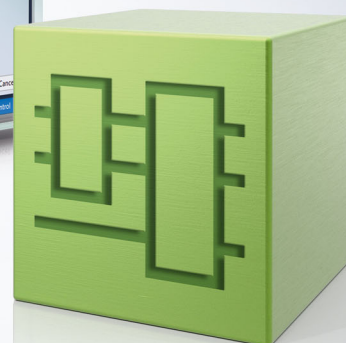
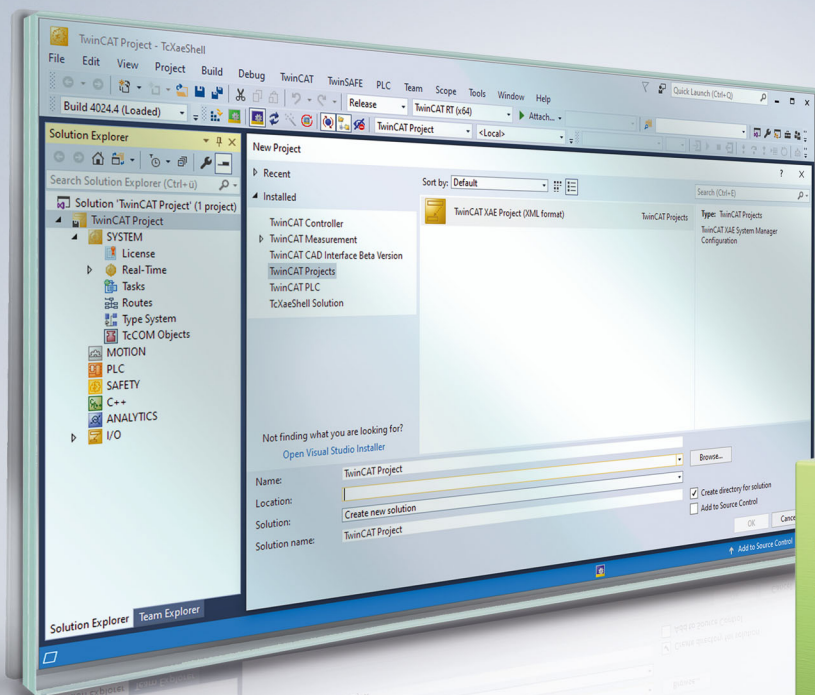


# BECKHOFF New Automation Technology

Manual | EN

# TE1000

TwinCAT 3 | PLC Library: Tc2\_Standard





# Table of contents

<b>1 Foreword</b> .....	<b>5</b>
1.1 Notes on the documentation .....	5
1.2 For your safety .....	5
1.3 Notes on information security.....	7
<b>2 Overview</b> .....	<b>8</b>
<b>3 Function blocks</b> .....	<b>9</b>
3.1 Bistable .....	9
3.1.1 RS .....	9
3.1.2 SR .....	9
3.2 Counter .....	10
3.2.1 CTD.....	10
3.2.2 CTU.....	10
3.2.3 CTUD .....	11
3.3 Timer .....	11
3.3.1 TOF .....	11
3.3.2 TON.....	12
3.3.3 TP.....	13
3.4 Timer (LTIME).....	14
3.4.1 LTOF .....	14
3.4.2 LTON.....	15
3.4.3 LTP.....	16
3.5 Trigger.....	16
3.5.1 F_TRIG .....	16
3.5.2 R_TRIG.....	17
<b>4 String functions</b> .....	<b>18</b>
4.1 CONCAT .....	18
4.2 DELETE .....	18
4.3 FIND.....	19
4.4 INSERT .....	19
4.5 LEFT .....	20
4.6 LEN .....	20
4.7 MID.....	21
4.8 REPLACE .....	21
4.9 RIGHT .....	22
<b>5 String functions (WSTRING)</b> .....	<b>23</b>
5.1 WCONCAT .....	23
5.2 WDELETE.....	23
5.3 WFIND.....	24
5.4 WINSERT .....	24
5.5 WLEFT .....	25
5.6 WLEN.....	25
5.7 WMID .....	25
5.8 WREPLACE .....	26

---

5.9	WRIGHT .....	26
<b>6</b>	<b>Global constants .....</b>	<b>28</b>
6.1	Library version.....	28

# 1 Foreword

## 1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.

For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.

The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

### Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without notice.

No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

### Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.

If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

### Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702  
and similar applications and registrations in several other countries.

## EtherCAT®

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The distribution and reproduction of this document as well as the use and communication of its contents without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

## 1.2 For your safety

### Safety regulations

Read the following explanations for your safety.

Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

**Exclusion of liability**

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

**Personnel qualification**

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

**Signal words**

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

**Personal injury warnings****⚠ DANGER**

Hazard with high risk of death or serious injury.

**⚠ WARNING**

Hazard with medium risk of death or serious injury.

**⚠ CAUTION**

There is a low-risk hazard that could result in medium or minor injury.

**Warning of damage to property or environment****NOTICE**

The environment, equipment, or data may be damaged.

**Information on handling the product**

This information includes, for example:  
recommendations for action, assistance or further information on the product.

## 1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found [here](#).

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the [RSS feed](#).

## 2 Overview

The Standard Library includes all IEC61131-3 POUs. The POUs can be classified in:

- Bistable Function Blocks
- Trigger Function Blocks
- Counter
- Timer
- Timer (LTIME)
- String Functions
- String Functions (WSTRING)



### 3 Function blocks

#### 3.1 Bistable

##### 3.1.1 RS



Resetting Bistable Function Blocks

$Q1 = RS (SET, RESET1)$

means:  $Q1 = NOT\ RESET1\ AND\ (Q1\ OR\ SET)$

**VAR\_INPUT**

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

**VAR\_OUTPUT**

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

Internal implementation of fb:

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

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

##### 3.1.2 SR



Making Bistable Function Blocks Dominant

$Q1 = SR (SET1, RESET)$

means:  $Q1 = (NOT\ RESET\ AND\ Q1)\ OR\ SET1$ , SET1 and RESET are BOOL variables.

**VAR\_INPUT**

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

**VAR\_OUTPUT**

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

Internal implementation of fb:

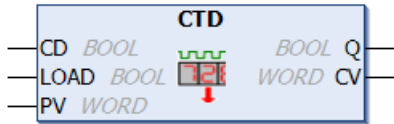
```
Q1 := (NOT RESET AND Q1) OR SET1;
```

## Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 3.2 Counter

### 3.2.1 CTD



Decrementer

#### VAR\_INPUT

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

#### VAR\_OUTPUT

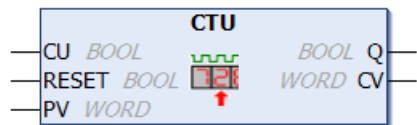
```
VAR_OUTPUT
  Q   : BOOL; (* Counter reached 0 *)
  CV  : WORD; (* Current Counter Value *)
END_VAR
```

When LOAD is TRUE, the counter variable CV will be initialized with the upper limit PV. If CD has a rising edge from FALSE to TRUE, CV will be lowered by 1 provided CV is greater than 0 (i.e., it doesn't cause the value to fall below 0). Q returns TRUE when CV is less than or equal to 0.

## Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 3.2.2 CTU



Incrementer

#### VAR\_INPUT

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

#### VAR\_OUTPUT

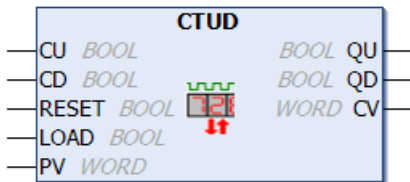
```
VAR_OUTPUT
  Q   : BOOL; (* Counter reached the Limit *)
  CV  : WORD; (* Current Counter Value *)
END_VAR
```

The counter variable CV will be initialized with 0 if RESET is TRUE. If CU has a rising edge from FALSE to TRUE, the function block CV will be raised by 1 provided CV is smaller than PV (i.e., it doesn't cause an overflow). Q will return TRUE when CV is greater than or equal to the upper limit PV.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.2.3 CTUD**



Incrementer and Decrementer

**VAR\_INPUT**

```

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
    
```

**VAR\_OUTPUT**

```

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

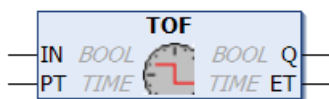
If RESET is valid, the counter variable CV will be initialized with 0. If LOAD is valid, CV will be initialized with PV. If CU has a rising edge from FALSE to TRUE, CV will be raised by 1 provided CV does not cause an overflow. If CD has a rising edge from FALSE to TRUE, CV will be lowered by 1 provided this does not cause the value to fall below 0. QU returns TRUE when CV has become greater than or equal to PV. QD returns TRUE when CV has become less than or equal to 0.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.3 Timer**

**3.3.1 TOF**



Timer off-delay

**VAR\_INPUT**

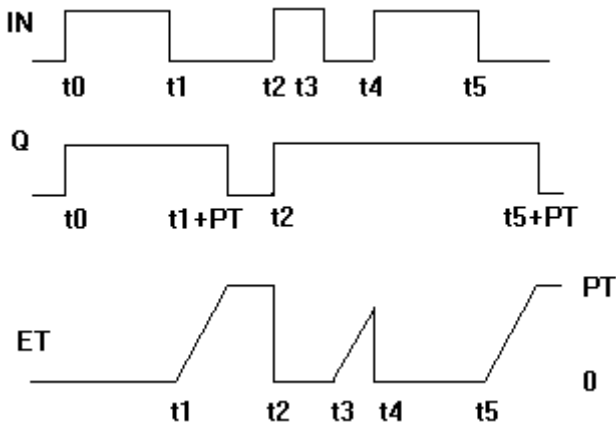
```
VAR_INPUT
  IN : BOOL; (* starts timer with falling edge, resets timer with rising edge *)
  PT : TIME; (* time to pass, before Q is set *)
END_VAR
```

**VAR\_OUTPUT**

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

When IN is TRUE, Q is TRUE and ET is 0. As soon as IN becomes FALSE, the time will begin to be counted in milliseconds in ET until its value is equal to that of PT. It will then remain constant. Q is FALSE when IN is FALSE and ET is equal to PT. Otherwise it is TRUE. Thus, Q has a falling edge when the time indicated in PT in milliseconds has run out.

Graphic display of TOF behavior over time:



The function TOF requires 15 byte data.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.3.2 TON**



Timer on-delay

**VAR\_INPUT**

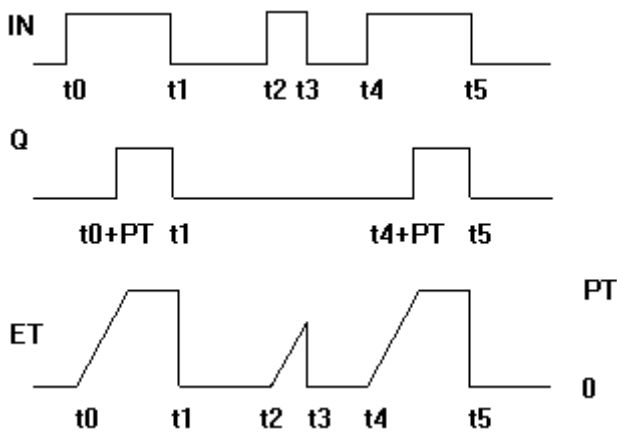
```
VAR_INPUT
  IN : BOOL; (* starts timer with rising edge, resets timer with falling edge *)
  PT : TIME; (* time to pass, before Q is set *)
END_VAR
```

**VAR\_OUTPUT**

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

If IN is FALSE, Q is FALSE and ET is 0. As soon as IN becomes TRUE, the time will begin to be counted in milliseconds in ET until its value is equal to PT. It will then remain constant. Q is TRUE when IN is TRUE and ET is equal to PT. Otherwise it is FALSE. Thus, Q has a rising edge when the time indicated in PT in milliseconds has run out.

Graphic display of TON behavior over time:

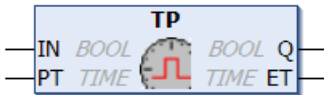


The function TON requires 15 byte data.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.3.3 TP**



Pulse generator. This function block can be used to generate pulses with a defined pulse duration.

**VAR\_INPUT**

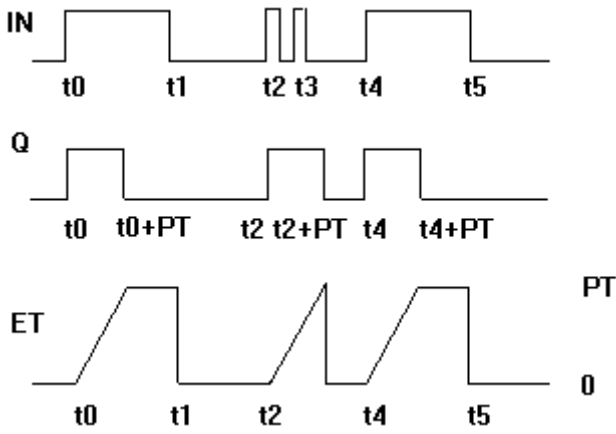
```
VAR_INPUT
  IN : BOOL; (* Trigger for Start of the Signal *)
  PT : TIME; (* The length of the High-Signal in ms *)
END_VAR
```

**VAR\_OUTPUT**

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

If IN is FALSE, the outputs are FALSE or 0. As soon as IN becomes TRUE, Q also becomes TRUE and remains TRUE for the pulse duration PT. As long as Q is TRUE, the time is incremented in milliseconds in ET, until the value reaches PT. The value then remains constant. The output Q remains TRUE until the pulse duration has elapsed, irrespective of the state of the input IN. Q therefore supplies a signal over the interval specified in PT.

Graphic display of TP behavior over time:



The TP function requires 14 bytes of data.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 3.4 Timer (LTIME)

#### 3.4.1 LTOF



Timer-off delay with 64-bit time data type (LTIME).

**VAR\_INPUT**

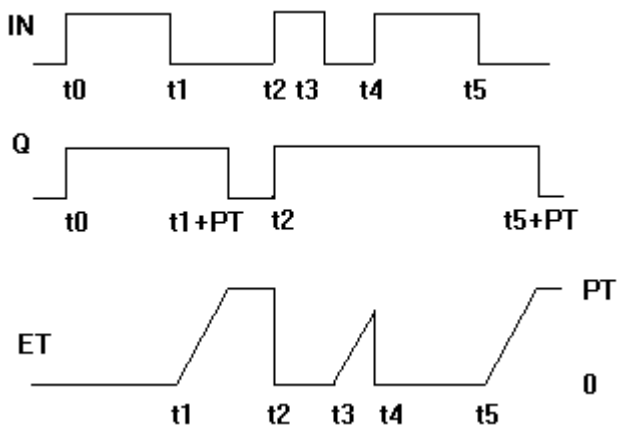
```
VAR_INPUT
  IN : BOOL; (*starts timer with falling edge, resets timer with rising edge*)
  PT : LTIME; (*time to pass before Q is reset*)
END_VAR
```

**VAR\_OUTPUT**

```
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
```

If IN is TRUE, the outputs are TRUE or 0. As soon as IN becomes FALSE, the time is incremented in nanoseconds in ET, until the value equals the value of PT. The value then remains constant. Q is FALSE, if IN = FALSE and ET = PT. Otherwise Q = TRUE. Q thus has a falling edge, once the time specified in nanoseconds in PT has elapsed.

Graph showing LTOF over time:



Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 3.4.2 LTON



Timer-on delay with 64-bit time data type (LTIME).

VAR\_INPUT

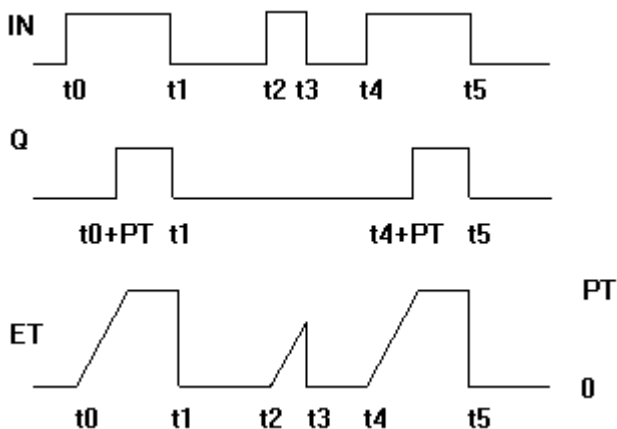
```
VAR_INPUT
    IN : BOOL; (*starts imter with rising edge, resets timer with falling edge*)
    PT : LTIME; (*time to pass before Q is set.*)
END_VAR
```

VAR\_OUTPUT

```
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
```

If IN is FALSE, the outputs are FALSE or 0. As soon as IN becomes TRUE, the time is incremented in nanoseconds in ET, until the value reaches PT. The value then remains constant. Q is TRUE, if IN = TRUE and ET = PT. Otherwise Q = FALSE. Q thus has a rising edge, once the time specified in nanoseconds in PT has elapsed.

Graph showing LTON over time:



**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.4.3 LTP**



Pulse generator with 64-bit time data type (LTIME). This function block can be used to generate pulses with a defined pulse duration.

**VAR\_INPUT**

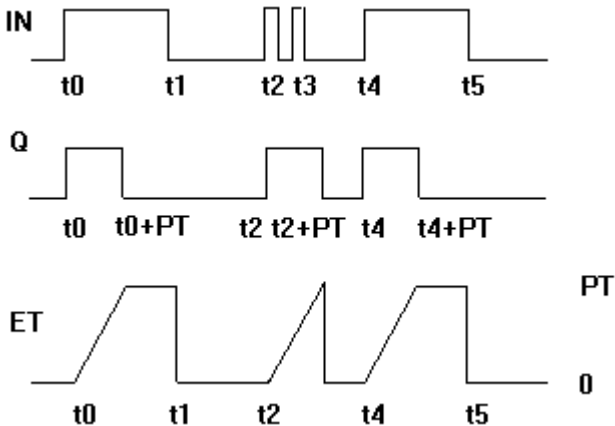
```
VAR_INPUT
  IN : BOOL; (*Trigger for Start of the Signal*)
  PT : LTIME; (*The length of the High- Signal*)
END_VAR
```

**VAR\_OUTPUT**

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

If IN is FALSE, the outputs are FALSE or 0. As soon as IN becomes TRUE, Q also becomes TRUE and remains TRUE for the pulse duration PT. As long as Q is TRUE, the time is incremented in nanoseconds in ET, until the value reaches PT. The value then remains constant. The output Q remains TRUE until the pulse duration has elapsed, irrespective of the state of the input IN. Q therefore supplies a signal over the interval specified in PT.

Graph showing LTP over time:



**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.5 Trigger**

**3.5.1 F\_TRIG**





Detector for a Falling Edge

**VAR\_INPUT**

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

**VAR\_OUTPUT**

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

The output Q will remain FALSE as long as the input variable CLK returns TRUE. As soon as CLK returns FALSE, Q will return TRUE. This means each time the function is called up, Q will return FALSE until CLK has a rising followed by a falling edge.

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**3.5.2 R\_TRIG**



Detector for a Rising Edge

**VAR\_INPUT**

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

**VAR\_OUTPUT**

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

The output Q will remain FALSE as long as the input variable CLK is FALSE. As soon as CLK returns TRUE, Q will return TRUE. This means each time the function is called up, Q will return FALSE until CLK has falling edge followed by a rising edge.

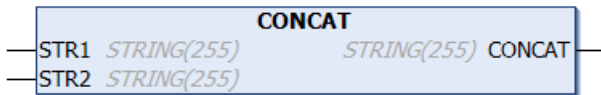
**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 4 String functions

On the [PLC samples page](#) you can download a sample project about string functions.

### 4.1 CONCAT



Concatenation (combination) of two strings.

#### FUNCTION CONCAT: STRING (255)

```

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

Example in IL:

```

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

Example in ST:

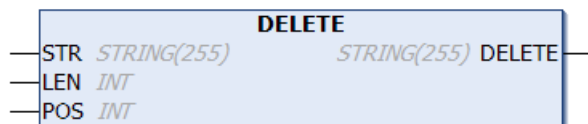
```

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

#### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 4.2 DELETE



The function DELETE removes a partial string from a larger string at a defined position. The input variable STR is type STRING, LEN and POS are type INT, the return value of the function is type STRING. DELETE (STR, LEN, POS) means: Delete LEN characters from STR beginning with the character in the POS.

#### FUNCTION DELETE: STRING (255)

```

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

Example in IL:

```

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

Example in ST:

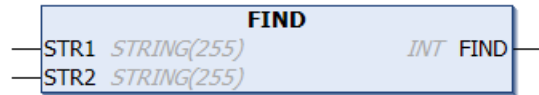
```

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

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 4.3 FIND



The function FIND searches for a partial string within a string. FIND (STR1, STR2) means: Find the position of the first character where STR2 appears in STR1 for the first time. If STR2 is not found in STR1, then OUT:=0.

**FUNCTION FIND: INT**

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

Example in IL:

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

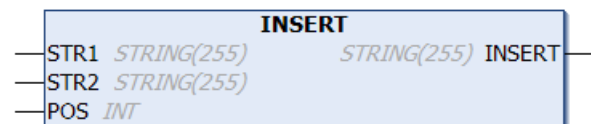
Example in ST:

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

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 4.4 INSERT



The function INSERT inserts a string into another string at a defined point. INSERT (STR1, STR2, POS) means: Insert STR2 into STR1 after position POS.

**FUNCTION INSERT: STRING (255)**

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

Example in IL:

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

Example in ST:

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

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**4.5 LEFT**

The function LEFT returns the left, initial string for a given string.

LEFT (STR, SIZE) means: Take the first SIZE character from the left in the string STR.

**FUNCTION LEFT: STRING (255)**

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

Example in IL:

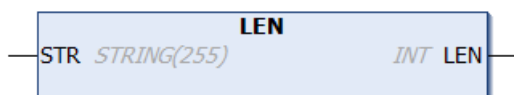
```
LD 'SUSI'
LEFT 3
ST Var1 (* Result is 'SUS' *)
```

Example in ST:

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

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

**4.6 LEN**

The function LEN returns the length of a string.

**FUNCTION LEN: INT**

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

Example in IL:

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

Example in ST:

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

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 4.7 MID



The function MID returns a partial string from within a string.  
 MID (STR, LEN, POS) means: Retrieve LEN characters from the STR string beginning with the character at position POS.

### FUNCTION MID: STRING (255)

```

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

Example in IL:

```

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

Example in ST:

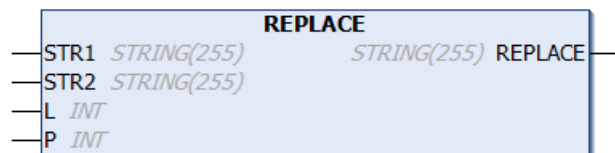
```

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 4.8 REPLACE



The function REPLACE replaces a partial string from a larger string with a third string.  
 REPLACE (STR1, STR2, L, P) means: Replace L characters from STR1 with STR2 beginning with the character in the P position.

### FUNCTION REPLACE: STRING (255)

```

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

Example in IL:

```

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

Example in ST:

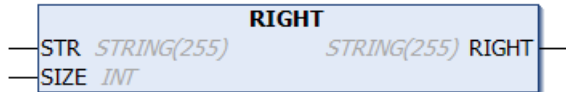
```

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 4.9 RIGHT



The function **RIGHT** returns the right, initial string for a given string.  
**RIGHT** (*STR*, *SIZE*) means: Take the first *SIZE* character from the right in the string *STR*.

### FUNCTION RIGHT: STRING (255)

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

Example in IL:

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

Example in ST:

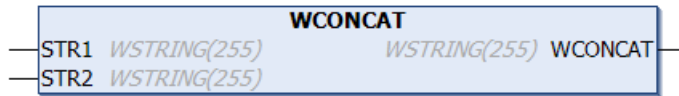
```
Var1 := RIGHT ('SUSI',3);
```

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5 String functions (WSTRING)

### 5.1 WCONCAT



Concatenation (combination) of two WSTRINGs.

#### 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
```

Example in IL:

```
LD "SUSI"
WCONCAT "WILLI"
ST Var1 (*Result is "SUSIWILLI"*)
```

Example in ST:

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

#### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

### 5.2 WDELETE



The function WDELETE deletes part of a WSTRING from a certain point. The input STR is of type WSTRING. LEN and POS are of type INT. The return value of the function is of type WSTRING. WDELETE (STR, LEN, POS) means: Delete LEN characters from STR beginning with the POSth character.

#### FUNCTION WDELETE: WSTRING (255)

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

Example in IL:

```
LD "SUXYSI"
WDELETE 2,3
ST Var1 (*Result is "SUSI"*)
```

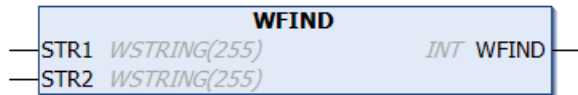
Example in ST:

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

#### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5.3 WFINd



The function WFINd searches for a part in WSTRING. WFINd (STR1, STR2) means: Find the position of the first character of first occurrence of STR2 in STR1. If STR2 does not occur in STR1, then OUT := 0 applies.

### FUNCTION WFINd: INT

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

Example in IL:

```
LD "SUXYSI"
WFINd "XY"
ST Var1 (*Result is 3*)
```

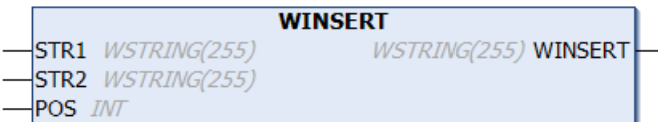
Example in ST:

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5.4 WINSERT



The function WINSERT adds a WString in another WString from a particular location. WINSERT (STR1, STR2, POS) means: Add STR2 in STR1 after the POSth position.

### FUNCTION WINSERT: WSTRING (255)

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

Example in IL:

```
LD "SUSI"
WINSERT "XY", 2
ST Var1 (*Result is "SUXYSI"*)
```

Example in ST:

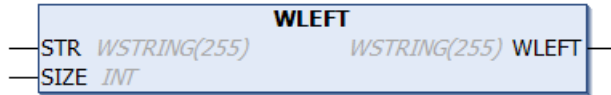
```
Var1 := WINSERT ("SUSI", "XY", 2);
```

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard



## 5.5 WLEFT



The function WLEFT supplies a left-hand start string for a WSTRING.  
 WLEFT (STR, SIZE) means: Take the first SIZE characters from the left in WString STR.

### FUNCTION WLEFT: WSTRING (255)

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

Example in IL:

```
LD "SUSI"
WLEFT 3
ST Var1 (*Result is "SUS"*)
```

Example in ST:

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5.6 WLEN



The function WLEN outputs the length of a WSTRING.

### FUNCTION WLEN: INT

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

Example in IL:

```
LD "SUSI"
WLEN
ST Var1 (*Result is 4*)
```

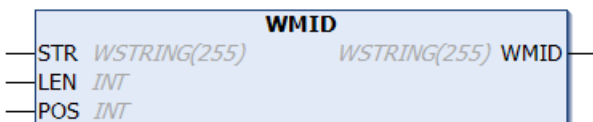
Example in ST:

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5.7 WMID



The function WMID supplies a substring of a WSTRING.

WMID(STR, LEN, POS) means: Fetch LEN characters from WSTRING STR beginning with the character at position POS.

### FUNCTION WMID: WSTRING (255)

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

Example in IL:

```
LD "SUSI"
WMID 2,2
ST Var1 (*Result is "US")
```

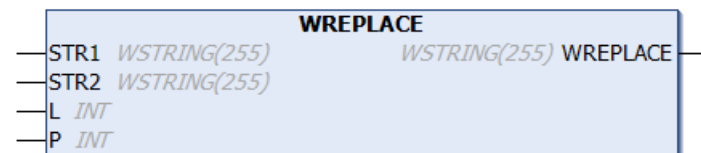
Example in ST:

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5.8 WREPLACE



The function WREPLACE replaces a substring of a WSTRING with another WSTRING.

WREPLACE (STR1, STR2, L, P) means: Replace L characters from STR1 with STR2 beginning with the Pth character.

### FUNCTION WREPLACE: WSTRING (255)

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

Example in IL:

```
LD "SUXYSI"
WREPLACE "XY",2
ST Var1 (*Result is "SKYSI")
```

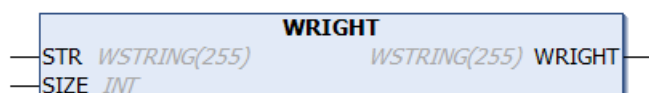
Example in ST:

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

### Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 5.9 WRIGHT



WRIGHT supplies a right-hand start string for a WSTRING.  
 WRIGHT (STR, SIZE) means: Take the first SIZE characters from the right in WString STR.

**FUNCTION WRIGHT: WSTRING (255)**

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

**Example in IL:**

```
LD "SUSI"
WRIGHT 3
ST Var1 (*Result is "USI"*)
```

**Example in ST:**

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

**Requirements**

Development environment	Target system type	PLC libraries to be linked
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

## 6 Global constants

### 6.1 Library version

All libraries have a specific version. This version is shown in the PLC library repository too. A global constant contains the library version information:

#### Global\_Version

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

**stLibVersion\_Tc2\_Standard:** version information of the Tc2\_standard library (type: ST\_LibVersion).

To compare the existing version to a required version use the function F\_CmpLibVersion.



All other possibilities known from TwinCAT2 libraries to query a library version are obsolete!

---



More Information:  
**[www.beckhoff.com/te1000](http://www.beckhoff.com/te1000)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

