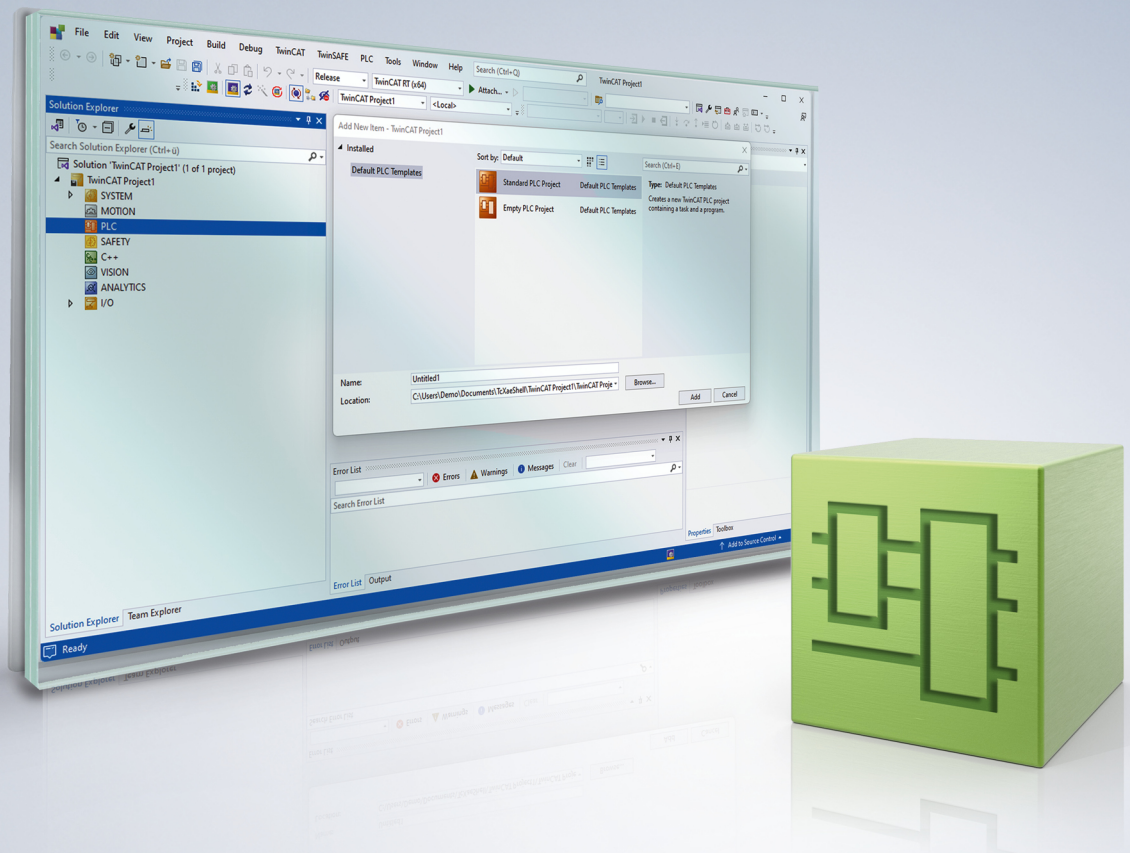
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc2_Standard



2026-04-08 | Version: 1.3.4

Table of contents

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.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled 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.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of the designations or trademarks contained in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

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

Copyright

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

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 in our https://www.beckhoff.com/secguide.

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 at https://www.beckhoff.com/secinfo.

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

The RS function block is bistable and RESET is dominant.

Q1 = RS (SET, RESET1)

I.e.: Q1 = NOT RESET1 AND (Q1 OR SET)

Inputs

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

Name	Type	Description
SET	BOOL	Set input - Q1 is set to TRUE on a rising edge.
RESET1	BOOL	Reset input (dominant) - Q1 is set to FALSE on rising edge

Outputs

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

Name	Type	Description
Q1	BOOL	Output

Internal implementation of the function block:

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

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.1.2 SR

The SR function block is bistable and SET is dominant.

Q1 = SR (SET1, RESET)

I.e.: Q1 = (NOT RESET AND Q1) OR SET1Q1, SET1 and RESET are BOOL variables.

Inputs

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

Name	Type	Description
SET1	BOOL	Set input (dominant) - Q1 is set to TRUE on a rising edge
RESET	BOOL	Reset input - Q1 is set to FALSE on a rising edge

Outputs

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

Name	Type	Description
Q1	BOOL	Output

Internal implementation of the function block:

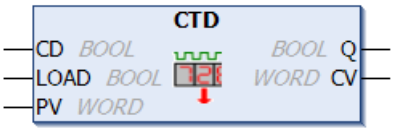
```
Q1 := (NOT RESET AND Q1) OR SET1;
```

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.2 Counter

3.2.1 CTD



The CTD function block is a down counter.

If LOAD = TRUE, the counter variable CV is initialized with the upper limit PV. If the CD input has a rising edge from FALSE to TRUE, the counter variable CV is reduced by one as long as CV is greater than zero. If the counter variable CV reaches the value zero, the Q output is set to TRUE.

Inputs

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

Name	Type	Description
CD	BOOL	On a rising edge: count down by one
LOAD	BOOL	TRUE: Set counter variable CV to the start value PV
PV	Word	Start value

Outputs

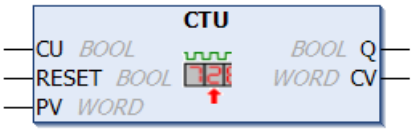
```
VAR_OUTPUT
    Q  : BOOL; (* Counter reached 0 *)
    CV : WORD; (* Current Counter Value *)
END_VAR
```

Name	Type	Description
Q	BOOL	TRUE if the counter value CV is 0.
CV	WORD	Counter variable (current counter value)

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.2.2 CTU



The CTU function block is an up counter.

If RESET = TRUE, the counter variable CV is initialized with zero. If the CU input has a rising edge from FALSE to TRUE, the counter variable CV is increased by one. If the counter variable CV reaches the value of the counter limit PV, the Q output is set to TRUE.

Inputs

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

Name	Type	Description
CU	BOOL	On a rising edge: count up by one
RESET	BOOL	TRUE: Reset counter variable CV to the value 0
PV	WORD	Counter limit

Outputs

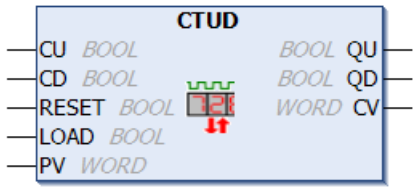
```
VAR_OUTPUT
    Q  : BOOL; (* Counter reached Limit *)
    CV : WORD; (* Current Counter Value *)
END_VAR
```

Name	Type	Description
Q	BOOL	TRUE if the counter value CV is greater than or equal to the counter limit PV.
CV	WORD	Counter variable (current counter value)

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.2.3 CTUD



The CTUD function block is an up and down counter.

If RESET = TRUE, the counter variable CV is initialized with 0. If LOAD = TRUE, the counter variable CV is initialized with the upper limit PV.
If the CU input has a rising edge from FALSE to TRUE, the counter variable CV is increased by one. If the counter variable CV reaches the value of the counter limit PV, the QU output is set to TRUE.
If the CD input has a rising edge from FALSE to TRUE, the counter variable CV is reduced by one as long as CV is greater than 0. If the counter variable CV reaches the value 0, the QD output is set to TRUE.

Inputs

```
VAR_INPUT
    CU    : BOOL; (* Count Up on Rising Edge*)
    CD    : BOOL; (* Count Down on Rising Edge*)
    RESET : BOOL; (* Reset Counter to 0 *)
    LOAD  : BOOL; (* Load Start Value *)
    PV    : WORD; (* Start Value / Counter Limit *)
END_VAR
```

Name	Type	Description
CU	BOOL	On a rising edge: count up by one
CD	BOOL	On a rising edge: count down by one
RESET	BOOL	TRUE: Reset counter variable CV to the value 0
LOAD	BOOL	TRUE: Set counter variable CV to the start value PV
PV	WORD	Start value/counter limit

Outputs

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

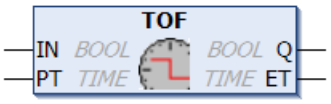
Name	Type	Description
QU	BOOL	TRUE if the counter value CV is greater than or equal to the counter limit PV.
QD	BOOL	TRUE if the counter value CV is 0.
CV	WORD	Counter variable (current counter value)

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.3 Timer

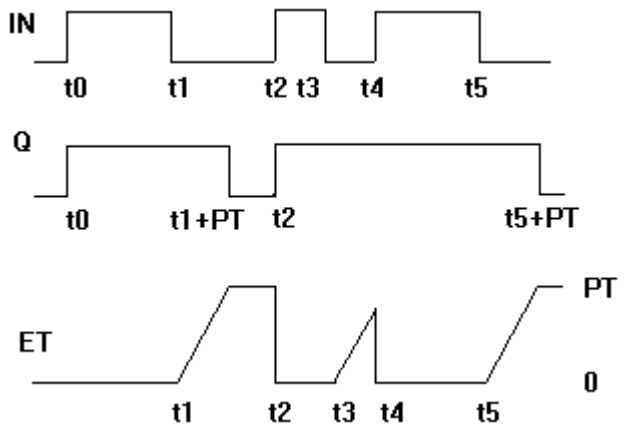
3.3.1 TOF



The TOF function block implements a timer with a switch-off delay.

If IN = TRUE, the Q output has the value TRUE and the ET output has the value zero. As soon as IN is set to FALSE, the time in milliseconds is counted up in ET until the setpoint PT is reached. Q is FALSE, if IN = FALSE and ET = PT. Otherwise Q = TRUE.

Graph showing TOF over time:



The TOF function block requires 15 bytes of data.

Inputs

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

Name	Type	Description
IN	BOOL	Falling edge: Start timer Rising edge: Reset timer
PT	TIME	Delay time (time that elapses before Q is reset)

Outputs

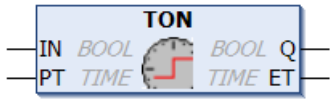
```
VAR_OUTPUT
    Q  : BOOL; (* is FALSE, PT seconds after IN had a falling edge *)
    ET : TIME; (* elapsed time *)
END_VAR
```

Name	Type	Description
Q	BOOL	TRUE if the IN input is TRUE. FALSE if the IN input is FALSE and the time specified in PT has expired.
ET	BOOL	Elapsed time since the falling edge at the IN input.

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

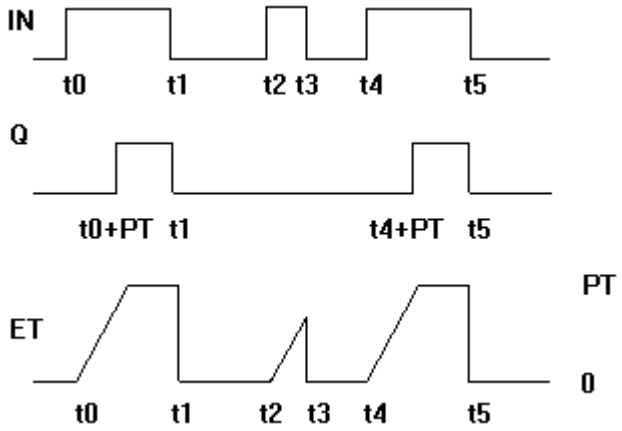
3.3.2 TON



The TON function block implements a timer with a switch-on delay.

If IN = FALSE, the Q output has the value FALSE and the ET output has the value zero. As soon as IN is set to TRUE, the time in milliseconds is counted up in ET until the setpoint PT is reached. Q is TRUE, if IN = TRUE and ET = PT. Otherwise Q = FALSE.

Graph showing TON over time:



The TON function block requires 15 bytes of data.

Inputs

```
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
```

Name	Type	Description
IN	BOOL	Rising edge: Start timer Falling edge: Reset timer
PT	TIME	Delay time (time elapsed before Q is set)

PT	TIME	Delay time (time elapsed before Q is set)

Outputs

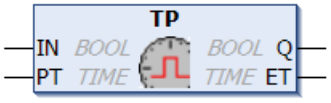
```
VAR_OUTPUT
    Q  : BOOL; (* is TRUE, PT seconds after IN had a rising edge *)
    ET : TIME; (* elapsed time *)
END_VAR
```

Name	Type	Description
Q	BOOL	FALSE if the IN input is FALSE. TRUE if the IN input is TRUE and the time specified in PT has elapsed.
ET	TIME	Elapsed time since the rising edge at the IN input.

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

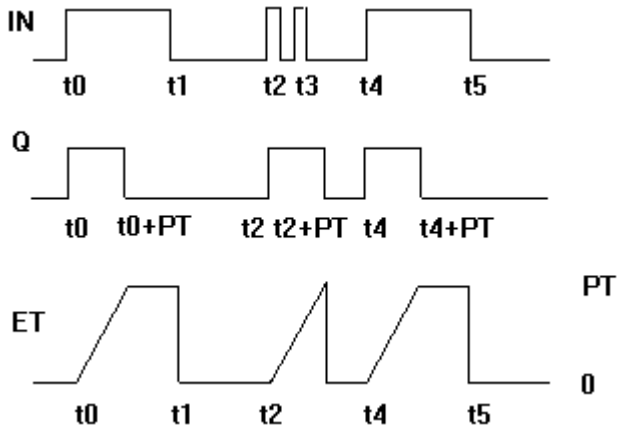
3.3.3 TP



The TP function block, a pulse generator, can be used to generate pulses with a defined pulse duration.

If IN = FALSE and Q = FALSE, the ET output has the value zero. As soon as IN is set to TRUE, Q = TRUE and remains TRUE for the pulse duration PT. As long as Q = TRUE, the time in milliseconds is counted up in ET until the setpoint PT is reached. The Q output remains TRUE until the pulse duration has elapsed, irrespective of the state of the IN input.

Graph showing TP over time:



The TP function block requires 14 bytes of data.

Inputs

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

Name	Type	Description
IN	BOOL	Rising edge: Start pulse timer and set Q to TRUE
PT	TIME	Pulse duration (length of the high signal)

Outputs

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

Name	Type	Description
Q	BOOL	TRUE after a rising edge at the IN input for the pulse duration PT.
ET	TIME	Elapsed time since the start of the pulse timer.

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

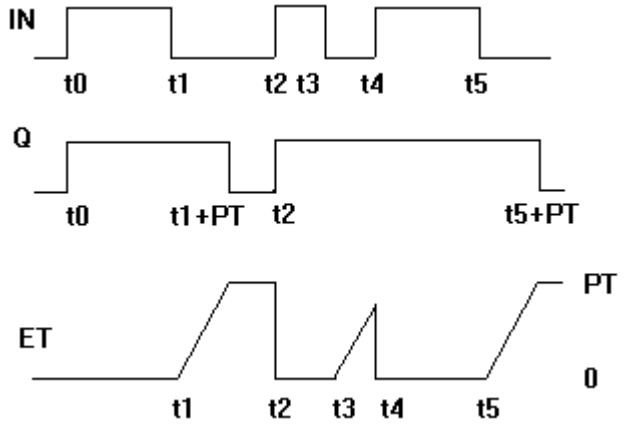
3.4 Timer (LTIME)

3.4.1 LTOF

The LTOF function block implements a timer with switch-off delay with 64-bit time data type (LTIME).

If IN = TRUE, the Q output has the value TRUE and the ET output has the value zero. As soon as IN is set to FALSE, the time in nanoseconds is counted up in ET until the setpoint PT is reached. Q is FALSE, if IN = FALSE and ET = PT. Otherwise Q = TRUE.

Graph showing LTOF over time:



Inputs

```
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
```

Name	Type	Description
IN	BOOL	Falling edge: Start timer Rising edge: Reset timer
PT	TIME	Delay time (time that elapses before Q is reset)

Outputs

```
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
```

Name	Type	Description
Q	BOOL	TRUE if the IN input is TRUE. FALSE if the IN input is FALSE and the time specified in PT has expired.
ET	BOOL	Elapsed time since the falling edge at the IN input.

Requirements

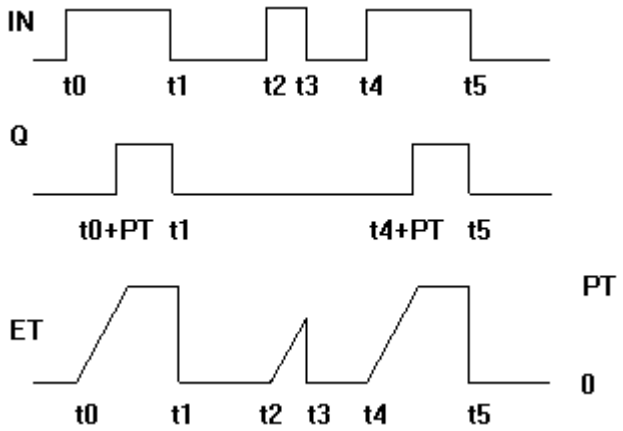
Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.4.2 LTON

The LTON function block implements a timer with a switch-on delay with a 64-bit time data type (LTIME).

If IN = FALSE, the Q output has the value FALSE and the ET output has the value zero. As soon as IN is set to TRUE, the time in nanoseconds is counted up in ET until the setpoint PT is reached. Q is TRUE, if IN = TRUE and ET = PT. Otherwise Q = FALSE.

Graph showing LTON over time:



Inputs

```
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
```

Name	Type	Description
IN	BOOL	Rising edge: Start timer Falling edge: Reset timer
PT	TIME	Delay time (time elapsed before Q is set)

Outputs

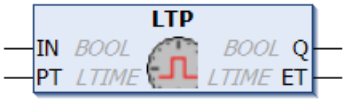
```
VAR_OUTPUT
    Q  : BOOL; (*is TRUE, PT seconds after IN had a rising edge*)
    ET : LTIME (*elapsed time since rising edge at IN*)
END_VAR
```

Name	Type	Description
Q	BOOL	FALSE if the IN input is FALSE. TRUE if the IN input is TRUE and the time specified in PT has elapsed.
ET	TIME	Elapsed time since the rising edge at the IN input.

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

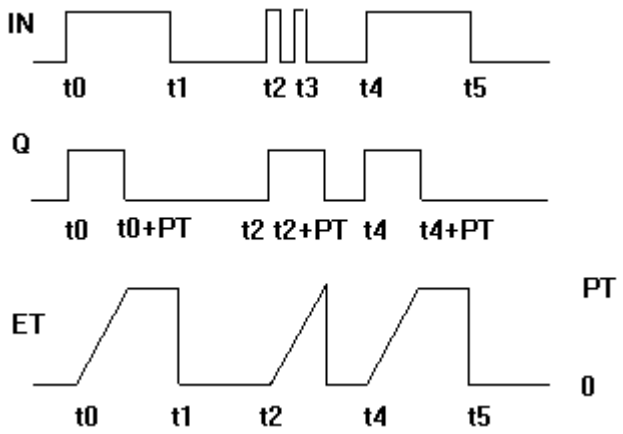
3.4.3 LTP



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

If IN = FALSE and Q = FALSE, the ET output has the value zero. As soon as IN is set to TRUE, Q = TRUE and remains TRUE for the pulse duration PT. As long as Q = TRUE, the time in nanoseconds is counted up in ET until the setpoint PT is reached. The Q output remains TRUE until the pulse duration has elapsed, irrespective of the state of the IN input.

Graph showing LTP over time:



Inputs

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

Name	Type	Description
IN	BOOL	Rising edge: Start pulse timer and set Q to TRUE
PT	TIME	Pulse duration (length of the high signal)

Outputs

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

Name	Type	Description
Q	BOOL	TRUE after a rising edge at the IN input for the pulse duration PT.
ET	TIME	Elapsed time since the start of the pulse timer.

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.5 Trigger

3.5.1 F_TRIG

The F_TRIG function block is a detector for a falling edge.

As long as the CLK input variable = TRUE, the Q output = FALSE. As soon as CLK = FALSE, Q is set to TRUE for one cycle. This means that each time the function is called, Q returns FALSE until CLK has a rising edge and then a falling edge.

Inputs

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

Name	Type	Description
CLK	BOOL	Boolean signal to be checked

Outputs

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

Name	Type	Description
Q	BOOL	TRUE for a falling edge at the CLK input (for one cycle)

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

3.5.2 R_TRIG

The R_TRIG function block is a rising edge detector.

As long as the CLK input variable = FALSE, the Q output = FALSE. As soon as CLK = TRUE, Q is set to TRUE for one cycle. This means that each time the function is called, Q returns FALSE until CLK has a falling edge and then a rising edge.

Inputs

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

Name	Type	Description
CLK	BOOL	Boolean signal to be checked

Outputs

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

Name	Type	Description
Q	BOOL	TRUE for a rising edge at the CLK input (for one cycle)

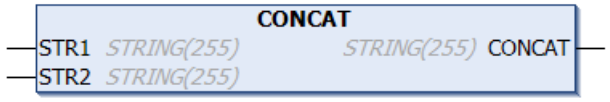
Requirements

Development environment	Target platform	PLC library to include
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 platform	PLC library to include
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 platform	PLC library to include
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 *)
```

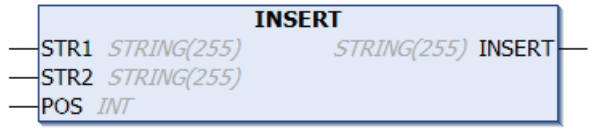
Example in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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 platform	PLC library to include
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' *)
```

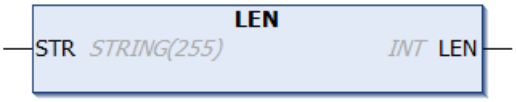
Example in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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_VA
```

Example in IL:

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

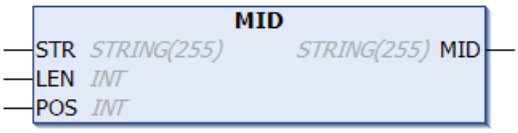
Example in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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' *)
```

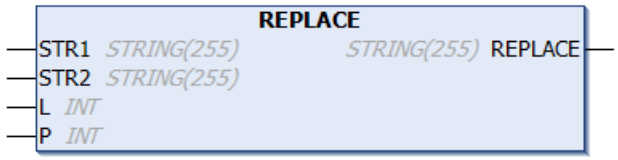
Example in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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' *)
```

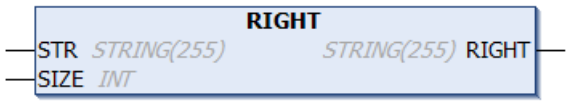
Example in ST:

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

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

4.9 RIGHT



The RIGHT function returns a right-hand start string of a string.
RIGHT (STR, SIZE) means: take the first SIZE characters from the right in the STR string.

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:

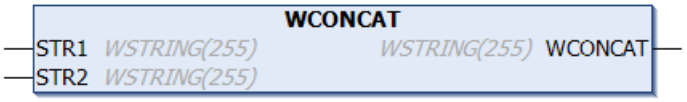
```
Var1 := RIGHT ('SUSI',3);
```

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

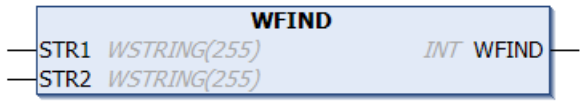
Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

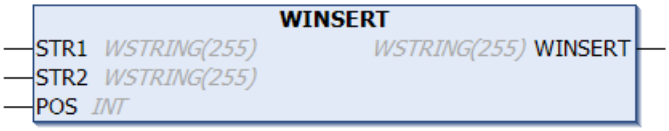
Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

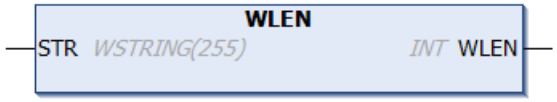
Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
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
```

Sample in IL:

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

Sample in ST:

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

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

6 Global constants

6.1 Library version

All libraries have a certain version. This version is displayed in the repository of the PLC library.
The version number of the library is stored in a global constant.

Global_Version

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

Name	Type	Description
stLibVersion_Tc2_Standard	ST_LibVersion	Version number of the Tc2_Standard library

The function F_CmpLibVersion is used to compare the existing and required version.

Compatibility with TwinCAT 2

Query options for TwinCAT 2 libraries are no longer available.

Requirements

Development environment	Target platform	PLC library to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Standard

Trademark statements

More Information:
www.beckhoff.com/te1000

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