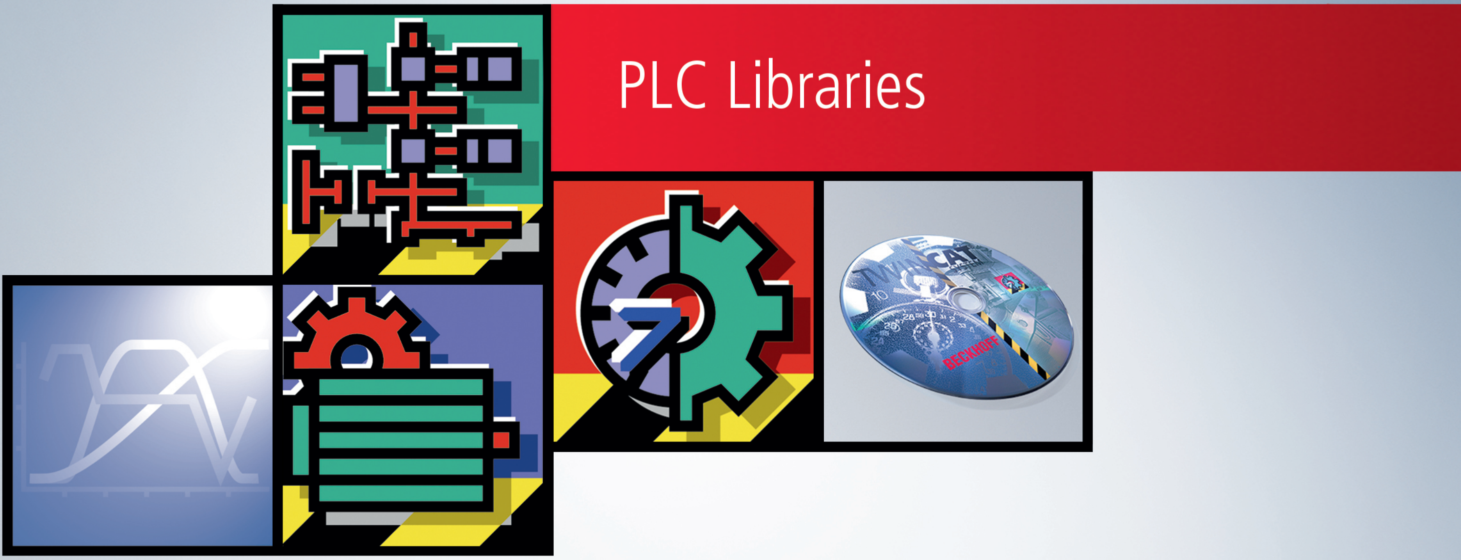
BECKHOFF New Automation Technology

Manual | EN

# TX1200

TwinCAT 2 | PLC Library: Standard



2022-04-12 | Version: 1.0

# Table of contents

**1 Foreword ..... 5**

1.1 Notes on the documentation ..... 5

1.2 Safety instructions ..... 6

1.3 Notes on information security ..... 7

**2 Overview ..... 8**

**3 Function blocks ..... 9**

3.1 Trigger ..... 9

3.1.1 F_TRIG ..... 9

3.1.2 R_TRIG ..... 9

3.2 Bistable ..... 10

3.2.1 RS ..... 10

3.2.2 SR ..... 11

3.2.3 SEMA ..... 11

3.3 Counter ..... 12

3.3.1 CTD ..... 12

3.3.2 CTU ..... 13

3.3.3 CTUD ..... 13

3.4 Timer ..... 14

3.4.1 TOF ..... 14

3.4.2 TON ..... 15

3.4.3 TP ..... 16

**4 String functions ..... 18**

4.1 CONCAT ..... 18

4.2 DELETE ..... 18

4.3 FIND ..... 19

4.4 INSERT ..... 20

4.5 LEFT ..... 20

4.6 LEN ..... 21

4.7 MID ..... 22

4.8 REPLACE ..... 22

4.9 RIGHT ..... 23

# 1 Foreword

## 1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the 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 prior announcement.
No claims for the modification of 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 trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

### Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

### Copyright

## 1.2 Safety instructions

### Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

### Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

### Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

**⚠ DANGER**

**Serious risk of injury!**

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

**⚠ WARNING**

**Risk of injury!**

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

**⚠ CAUTION**

**Personal injuries!**

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

***NOTE***

**Damage to the environment or devices**

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

**Tip or pointer**

This symbol indicates information that contributes to better understanding.

## 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

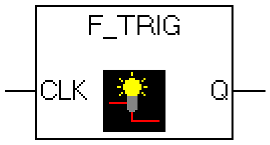
The Standard Library includes all IEC61131-3 POUs. The POUs can be classified in:

- Bistable Function Blocks [▸ 10]
- Trigger Function Blocks [▸ 9]
- Counter [▸ 12]
- Timer [▸ 14]
- String Functions [▸ 18]

# 3 Function blocks

## 3.1 Trigger

### 3.1.1 F_TRIG



Detector for a Falling Edge

#### VAR_INPUT

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

#### VAR_OUTPUT

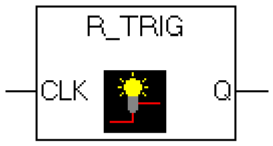
```
VAR_OUTPUT
    Q        : BOOL; (* Edge detected *)
END_VAR
VAR
    M        : BOOL;
END_VAR
```

The output Q and the help variable M will remain FALSE as long as the input variable CLK returns TRUE. As soon as CLK returns FALSE, Q will first return TRUE, then M will be set to 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 include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

### 3.1.2 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
VAR
    M       : BOOL;
END_VAR
```

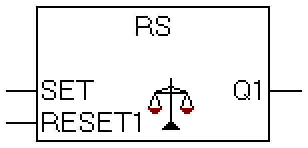
The output Q and the help variable M will remain FALSE if the input variable CLK is FALSE. As soon as CLK returns TRUE, Q will first return TRUE, then M will be set to TRUE. This means each time the function is called up, Q will return FALSE until CLK has falling edge followed by an rising edge.

### Requirements

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.2 Bistable

### 3.2.1 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:

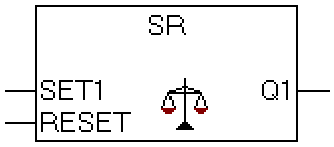
```
Q1: = NOT RESET1 AND (Q1 OR SET);
```

### Requirements

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.2.2 SR



Making Bistable Function Blocks Dominant:

Q1 = SR (SET1, RESET)

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

**VAR_INPUT**

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

**VAR_OUTPUT**

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

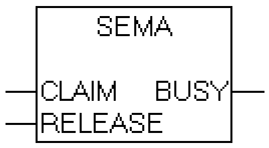
Internal implementation of fb:

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

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.2.3 SEMA



A Software Semaphore (Interruptible).

**VAR_INPUT**

```
VAR_INPUT
    CLAIM    : BOOL;
    RELEASE : BOOL;
END_VAR
```

### VAR_OUTPUT

```
VAR_OUTPUT
    BUSY     : BOOL;
END_VAR
```

If BUSY is TRUE when SEMA is called up, this means that a value has already been assigned to SEMA (SEMA was called up with CLAIM = TRUE). If BUSY is FALSE, SEMA has not yet been called up or it has been released (called up with RELEASE = TRUE).

Internal implementation of fb:

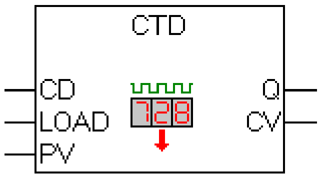
```
BUSY := X;

IF CLAIM THEN
    X:=TRUE;
ELSIF RELEASE
    THEN
        BUSY := FALSE;
        X:= FALSE;
END_IF
```

### Requirements

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.3 Counter

### 3.3.1 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

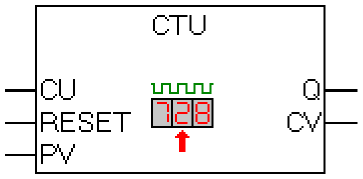
```
VAR_OUTPUT
    Q        : BOOL;  (* Counter reached 0 *)
    CV       : WORD;  (* Current Counter Value *)
END_VAR
```

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 include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.3.2 CTU



Incrementer:

### VAR_INPUT

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

### VAR_OUTPUT

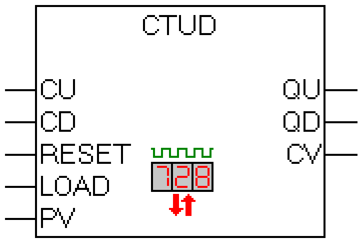
```
VAR_OUTPUT
    Q       : BOOL;  (* Counter reached the Limit *)
    CV      : WORD;  (* Current Counter Value *)
END_VAR
```

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 include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.3.3 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
```

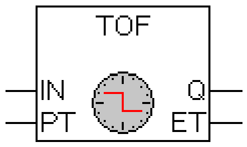
If RESET is valid, the counter variable CV will be initialized with 0. If LOAD is valid, CV will be initialized with PV. If CU has a rising edge from FALSE to TRUE, CV will be raised by 1 pro-vided CV does not cause an overflow. If CD has a rising edge from FALSE to TRUE, CV will be lowered by 1 provided 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 include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.4 Timer

### 3.4.1 TOF



Timer off-delay

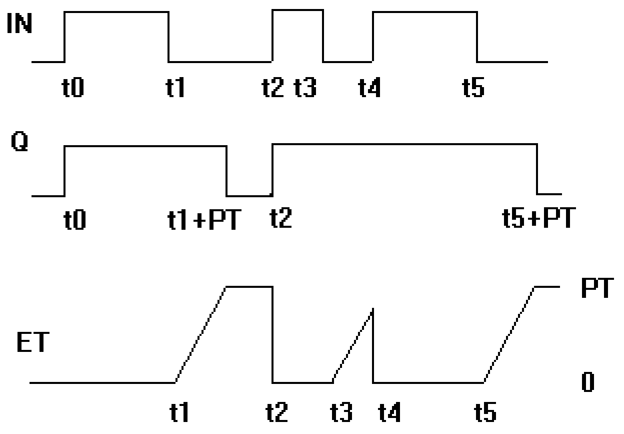
### VAR_INPUT

```
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:

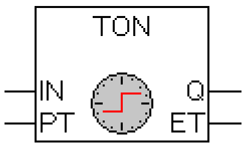


The function TOF requires 15 byte data

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.4.2 TON



Timer on-delay

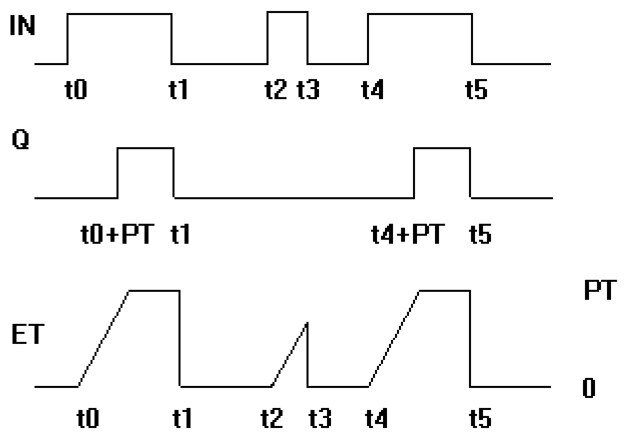
### VAR_INPUT

```
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:

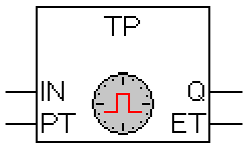


The function TON requires 15 byte data

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 3.4.3 TP



The pulse timer block can be used to generate output pulses of a given time 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, Q is FALSE and ET is 0.

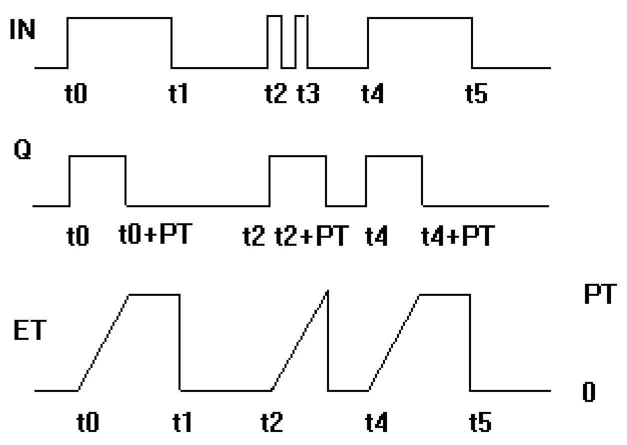
As input IN goes TRUE, tho output Q follows and remains TRUE for the pulse duration as specified by time input PT.

While the pulse output ist TRUE, the elapsed time ET ist increased.

On the termination of the pulse, the elapsed time is held until the beginning of the next pulse, at which point it is reset.

The output Q will remain TRUE until the pulse time has elapsed, irrespective of the state of the input IN.

Graphic Display of the TP Time Sequence:



The function TP requires 14 byte data

**Requirements**

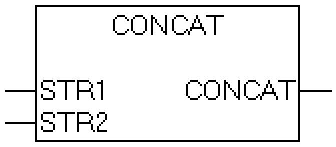
| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

# 4 String functions

**Note:**

Plaese note, string functions are not "thread safe": When using tasks, string functions may only be used in a single task. If the same function is used in different tasks, there is a danger of overwriting.

## 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' *)
```

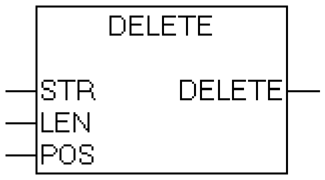
Example in ST:

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

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 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 ist 'SUSI' *)
```

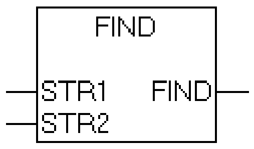
Example in ST:

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

### Requirements

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.3 FIND



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:

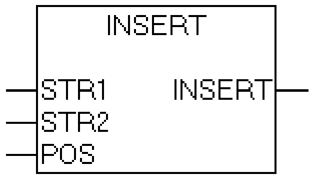
```
Var1 := FIND
('SUXYSI','XY');
```

### Requirements

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.4 INSERT



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' *)
```

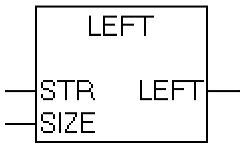
Example in ST:

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

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.5 LEFT



Left returns the left, initial string for a given string. LEFT (STR, SIZE) means: Take the first SIZE character from the right 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' *)
```

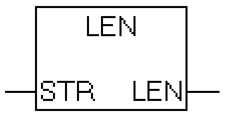
Example in ST:

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

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.6 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 *)
```

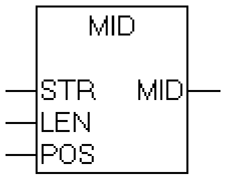
Example in ST:

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

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.7 MID



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' *)
```

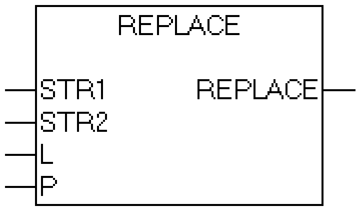
Example in ST:

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

**Requirements**

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.8 REPLACE



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' *)
```

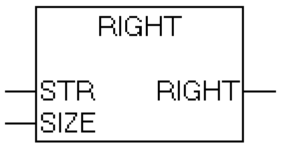
Example in ST:

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

### Requirements

| Development environment | Target system type | PLC libraries to include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

## 4.9 RIGHT



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 include |
|---|---|---|
| TwinCAT Version >= 2.6.0 | PC or CX (x86) | Standard.Lib |
| TwinCAT Version >= 2.6.0 | BC (165) | Standard.Lb6 |
| TwinCAT Version >= 2.9.0 | BCxx50 or BX | Standard.lbx |
| TwinCAT Version >= 2.10.0 Build >= 1301 | CX (ARM) | Standard.lib |

More Information:
**www.beckhoff.com/tx1200**

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