

取扱説明書

PLCライブラリ: Tc2\_Standard

TwinCAT 3

バージョン: 1.2  
日付: 2020-04-02

**BECKHOFF**



# 目次

<b>1</b>	<b>序文</b> .....	<b>5</b>
1.1	取扱説明書に関する注記 .....	5
1.2	安全に関する指示事項 .....	5
<b>2</b>	<b>概要</b> .....	<b>7</b>
<b>3</b>	<b>ファンクションブロック</b> .....	<b>8</b>
3.1	Bistable .....	8
3.1.1	RS .....	8
3.1.2	SR .....	8
3.2	カウンタ .....	9
3.2.1	CTD .....	9
3.2.2	CTU .....	9
3.2.3	CTUD .....	10
3.3	タイマ .....	11
3.3.1	TOF .....	11
3.3.2	TON .....	11
3.3.3	TP .....	12
3.4	タイマ (LTIME) .....	13
3.4.1	LTOF .....	13
3.4.2	LTON .....	14
3.4.3	LTP .....	15
3.5	トリガ .....	16
3.5.1	F_TRIG .....	16
3.5.2	R_TRIG .....	16
<b>4</b>	<b>Stringファンクション</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</b>	<b>Stringファンクション (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 .....	26
5.8	WREPLACE .....	26

---

5.9	WRIGHT .....	27
<b>6</b>	<b>グローバル定数 .....</b>	<b>28</b>
6.1	ライブラリバージョン .....	28

# 1 序文

## 1.1 取扱説明書に関する注記

この説明は対応する国内規格を熟知した、トレーニングを受けた制御、オートメーションエンジニアリングの専門技術者のみの使用を対象としています。

コンポーネントのインストールとコミッショニングの際には、取扱説明書および以下の注意事項と説明に従うことが重要です。

技術者には各設置およびコミッショニングのそれぞれの時点で、発行された取扱説明書を使用する義務があります。

本製品を使用するうえでの責任者は、本製品の用途および使用方法が、関連するすべての法律、法規、ガイドラインおよび規格を含む、安全に関するすべての要件を満たしていることを確認してください。

### 免責事項

この取扱説明書の記載内容は、一般的な製品説明および性能を記載したものであり、場合により記載通りに動作しないことがあります。

製品の情報・仕様は予告なく変更されます。

この説明書に記載されているデータ、図および説明に基づいて、すでに納品されている製品の変更を要求することはできません。

### 商標

Beckhoff®、TwinCAT®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS®およびXPlanar®は、Beckhoff Automation GmbHの登録商標です。

この取扱説明書で使用されているその他の名称は商標である可能性があり、第三者が独自の目的のために使用すると所有者の権利を侵害する可能性があります。

### 特許出願

EtherCAT Technologyについては、欧州特許 EP1590927、EP1789857、EP1456722およびEP2137893、ドイツ特許DE102015105702

に記載されていますが、これらに限定されるものではありません。

## EtherCAT®

EtherCAT®は、Beckhoff Automation GmbH (ドイツ)によりライセンスを受けた登録商標および特許技術です。

### 著作権

© Beckhoff Automation GmbH & Co. KG, Germany.

明示的な許可なく、本書の複製、配布、使用、および他への内容の転載は禁止されています。

これに違反した者は損害賠償の責任を負います。すべての権利は、特許、実用新案、意匠の付与の際に留保されます。

## 1.2 安全に関する指示事項

### 安全に関する注意事項

この取扱説明書に記載された安全に関する指示や注意事項はよくお読みになり、必ず指示に従ってください。

## 納入仕様

すべての製品は、用途に適した特定のハードウェア構成およびソフトウェア構成を有する状態で供給されます。ハードウェアまたはソフトウェアに取扱説明書に記載されている以外の変更を加えることは許可されていません。許可されていない変更を加えると、Beckhoff Automation GmbH & Co. KGの保証の対象外となります。

## 使用者の資格

この説明書は関連する国内法規を熟知した、制御およびオートメーションエンジニアリングの専門家の使用を目的としています。

## 安全記号の説明

この取扱説明書では、安全に関する指示や注意事項とともに以下の安全記号を使用します。安全に関する指示事項はよくお読みになり、必ず指示に従ってください。

### ⚠ 危険

#### 重大な人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に直ちに危害を及ぼします。

### ⚠ 警告

#### 人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼします。

### ⚠ 注意

#### 人的傷害の恐れ

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼす恐れがあります。

### 📌 注記

#### 物的損害と環境汚染

この記号が付いた安全に関する注意事項に従わないと、物的損害と環境汚染をもたらす恐れがあります。

#### ● ヒントまたはアドバイス

**i** この記号が示す情報により、さらに理解が深まります。

## 2 概要

標準ライブラリには、すべてのIEC61131-3 POUが含まれています。POUは、以下のように分類することができます。

- ・ Bistableファンクションブロック
- ・ Triggerファンクションブロック
- ・ カウンタ
- ・ タイマ
- ・ タイマ(LTIME)
- ・ Stringファンクション
- ・ Stringファンクション(WSTRING)

## 3 ファンクションブロック

### 3.1 Bistable

#### 3.1.1 RS



Rest\_Setフリップフロップファンクションブロック

$Q1 = RS (SET, RESET1)$

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

##### VAR\_INPUT

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

##### VAR\_OUTPUT

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

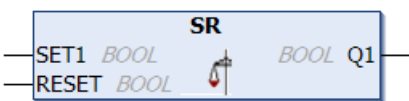
fbの内部実装:

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

##### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

#### 3.1.2 SR



Set\_Resetフリップフロップファンクションブロック

$Q1 = SR (SET1, RESET)$

意味:  $Q1 = (NOT\ RESET\ AND\ Q1)\ OR\ SET1$ 、SET1とRESETはBOOL変数です。

##### VAR\_INPUT

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



### VAR\_OUTPUT

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

fbの内部実装:

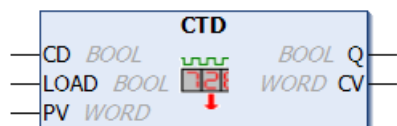
```
Q1 := (NOT RESET AND Q1) OR SET1;
```

#### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 3.2 カウンタ

### 3.2.1 CTD



カウントダウン

### VAR\_INPUT

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

### VAR\_OUTPUT

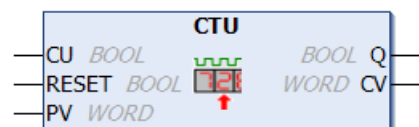
```
VAR_OUTPUT
  Q : BOOL; (* Counter reached 0 *)
  CV : WORD; (* Current Counter Value *)
END_VAR
```

LOADがTRUEのとき、カウンタ変数CVは上限値PVに初期化されます。CDのFALSEからTRUEへの立ち上がりエッジで、CVの値から1が減じられます。ただし、CVは0よりも大きい値となります(つまり、これによって値が0より小さくなることはありません)。CVの値が0以下であるとき、QはTRUEを返します。

#### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 3.2.2 CTU



カウントアップ

### VAR\_INPUT

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

### VAR\_OUTPUT

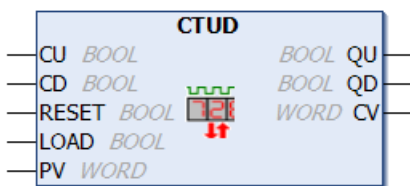
```
VAR_OUTPUT
  Q   : BOOL; (* Counter reached the Limit *)
  CV  : WORD; (* Current Counter Value *)
END_VAR
```

RESETがTRUEのとき、カウンタ変数CVは0に初期化されます。CUのFALSEからTRUEへの立ち上がりエッジで、ファンクションブロックCVの値に1が加えられます。ただし、CVはPVより小さい値となります(つまり、これによってオーバーフローは起こりません)。CVの値が上限値PV以上であるとき、QはTRUEを返します。

### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 3.2.3 CTUD



### カウントアップとカウントダウン

### 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
```

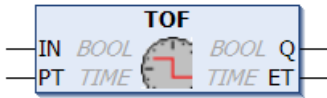
RESETが有効である場合は、カウンタ変数CVは0で初期化されます。LOADが有効である場合は、CVはPVで初期化されます。CUがFALSEからTRUEへの立ち上がりエッジをもつ場合は、CVの値に1が加えられます。ただし、CVはオーバーフローを発生させません。CDがFALSEからTRUEへの立ち上がりエッジをもつ場合は、CVの値から1が減じられます。ただし、これによって値が0より小さくなることはありません。CVの値がPV以上となるとき、QUはTRUEを返します。CVの値が0以下となるとき、QDはTRUEを返します。

### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 3.3 タイマ

#### 3.3.1 TOF



オフディレイタイマ

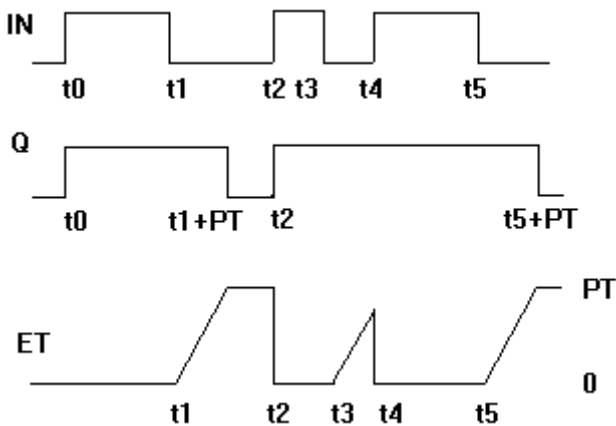
##### VAR\_INPUT

```
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
```

INがTRUEのとき、QはTRUEになり、ETは0になります。INがFALSEになると直ちにETの値がPTの値に等しくなるまで、ETで時間がミリ秒単位でカウントされ始めます。その後は、一定のままになります。INがFALSEでETがPTに等しいとき、QはFALSEです。そうでない場合は、TRUEです。そのため、PTにミリ秒単位で表示される時間が時間切れになると、Qは立ち下がりエッジをもちます。TOFのグラフィックディスプレイは時間の経過に従って次のように動作します。

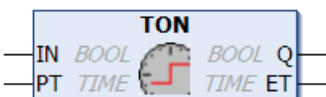


ファンクションTOFには、15バイトのデータが必要です。

##### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

#### 3.3.2 TON



オンディレイタイマ

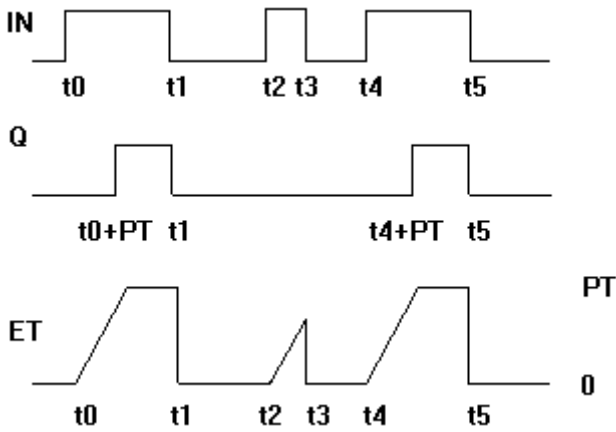
VAR\_INPUT

```
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
```

INがFALSEの場合は、QはFALSEになり、ETは0になります。INがTRUEになると直ちにETの値がPTの値に等しくなるまで、ETで時間がミリ秒単位でカウントされ始めます。その後は、一定のままになります。INがTRUEでETがPTに等しいとき、QはTRUEです。そうでない場合は、FALSEです。そのため、PTにミリ秒単位で表示される時間が時間切れになると、Qは立ち上がりエッジをもちます。TONのグラフィックディスプレイは時間の経過に従って次のように動作します。

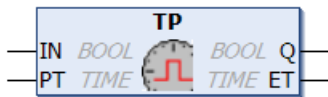


ファンクションTONには、15バイトのデータが必要です。

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

3.3.3 TP



パルスジェネレータこのファンクションブロックを使用して、定義済みのパルスの継続時間をもつパルスを生成することができます。

VAR\_INPUT

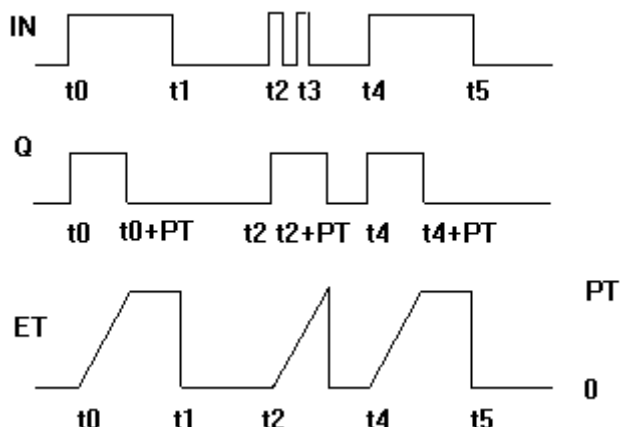
```
VAR_INPUT
  IN : BOOL; (* Trigger for Start of the Signal *)
  PT : TIME; (* The length of the High-Signal in ms *)
END_VAR
```

**VAR\_OUTPUT**

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

INがFALSEである場合は、出力はFALSEまたは0になります。INがTRUEになると直ちに、QもTRUEになり、パルスの継続時間PTの間、TRUEのままになります。QがTRUEである限り、値がPTに達するまで、ETにおいて時間がミリ秒単位でインクリメントされます。その後、値は一定のままになります。出力Qは、パルスの継続時間が経過するまで、入力INの状態に関わりなくTRUEのままになります。そのため、Qは、PTに指定した間隔で信号を供給します。

TPのグラフィックディスプレイは時間の経過に従って次のように動作します。



TPファンクションには、14バイトのデータが必要です。

**要件**

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 3.4 タイマ (LTIME)

#### 3.4.1 LTOF



64ビット時間データタイプのオフディレイタイマ (LTIME).

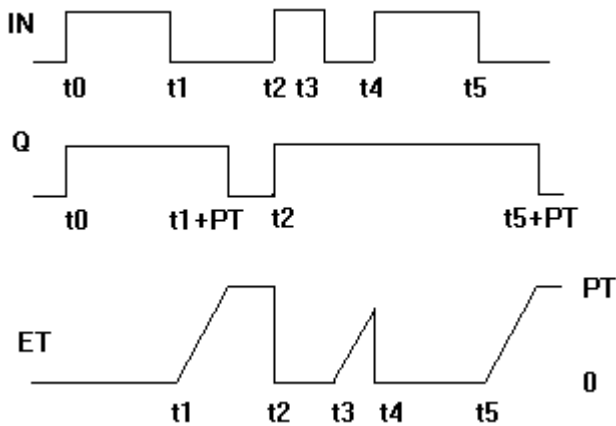
**VAR\_INPUT**

```
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
```

INがTRUEである場合は、出力はTRUEまたは0になります。INがFALSEになると直ちに、値がPTと等しくなるまで、ETにおいて時間がナノ秒単位でインクリメントされます。その後、値は一定のままになります。IN = FALSEかつET = PTである場合は、QはFALSEになります。そうでない場合は、Q = TRUEになります。そのため、PTにナノ秒単位で指定した時間が経過すると、Qは立ち下がりエッジをもちます。次のグラフは、LTOFを時間の経過に従って表示します。



要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 3.4.2 LTON



64ビット時間データタイプのオンディレイタイマ(LTIME).

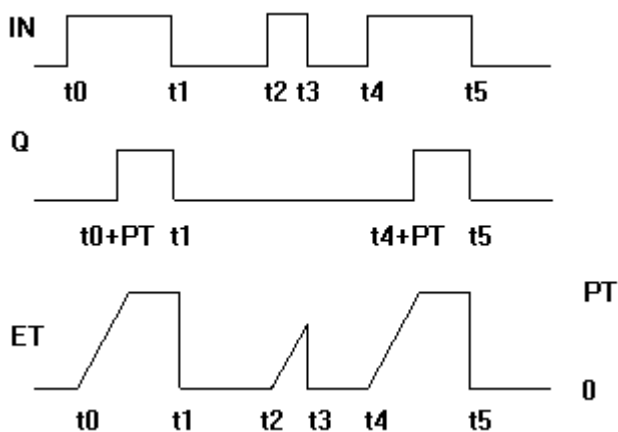
VAR\_INPUT

```
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
```

INがFALSEである場合は、出力はFALSEまたは0になります。INがTRUEになると直ちに、値がPTに達するまで、ETにおいてナノ秒単位で時間がインクリメントされます。その後、値は一定のままになります。IN = TRUEかつET = PTである場合は、QはTRUEになります。そうでない場合は、Q = FALSEになります。そのため、PTにナノ秒単位で指定した時間が経過すると、Qは立ち上がりエッジをもちます。次のグラフは、LTONを時間の経過に従って表示します。



要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 3.4.3 LTP



64ビット時間データタイプのパルスジェネレータ (LTIME)。このファンクションブロックを使用して、定義済みのパルスの継続時間をもつパルスを生成することができます。

**VAR\_INPUT**

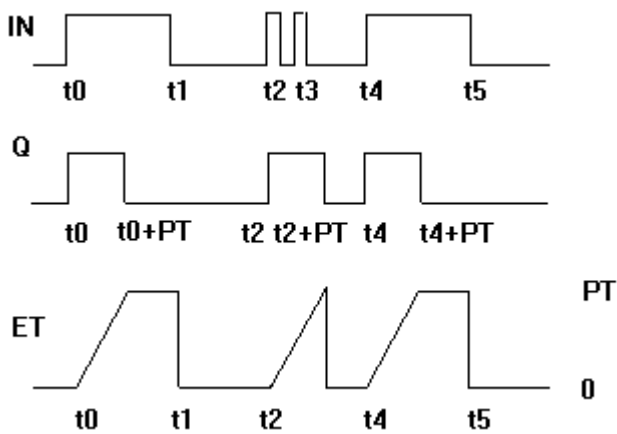
```
VAR_INPUT
    IN : BOOL; (*Trigger for Start of the Signal*)
    PT : LTIME; (*The length of the High- Signal*)
END_VAR
```

**VAR\_OUTPUT**

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

INがFALSEである場合は、出力はFALSEまたは0になります。INがTRUEになると直ちに、QもTRUEになり、パルスの継続時間PTの間、TRUEのままになります。QがTRUEである限り、値がPTに達するまで、ETにおいて時間がナノ秒単位でインクリメントされます。その後、値は一定のままになります。出力Qは、パルスの継続時間が経過するまで、入力INの状態に関わりなくTRUEのままになります。そのため、Qは、PTに指定した間隔で信号を供給します。

次のグラフは、LTPを時間の経過に従って表示します。



要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 3.5 トリガ

#### 3.5.1 F\_TRIG



立ち下がりエッジの検出

**VAR\_INPUT**

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

**VAR\_OUTPUT**

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

入力変数CLKがTRUEのとき、出力QはFALSEのままになります。CLKがTRUEからFALSEを返すと直ちに、Q（1サイクルタイムのみ）はTRUEを返します。

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

#### 3.5.2 R\_TRIG



立ち上がりエッジの検出



**VAR\_INPUT**

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

**VAR\_OUTPUT**

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

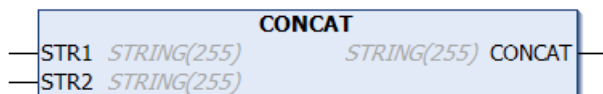
入力変数CLKがFALSEのとき、出力QはFALSEのままになります。CLKがFALSEからTRUEを返すと直ちに、Q（1サイクルタイムのみ）はTRUEを返します。

**要件**

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4 Stringファンクション

### 4.1 CONCAT



2つの文字列の連結(組み合わせ)

#### ファンクションCONCAT: STRING (255)

VAR\_INPUT

STR1 : STRING(255);

STR2 : STRING(255);

END\_VAR

ILの例:

LD 'SUSI'

CONCAT 'WILLI'

ST Var1 (\* Result is 'SUSIWILLI' \*)

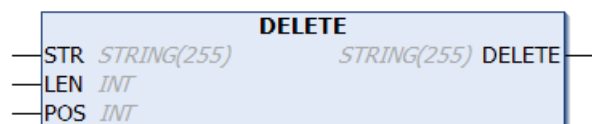
STの例:

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

#### 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 4.2 DELETE



ファンクションDELETEは、与えられた文字列から定義した箇所の部分文字列を削除します。入力変数STRはSTRING型であり、LENとPOSはINT型です。このファンクションの戻り値はSTRING型です。DELETE (STR, LEN, POS)の意味: STRのPOS位置から開始するLEN分の文字を削除します。

#### ファンクションDELETE: STRING (255)

VAR\_INPUT

STR : STRING(255);

LEN : INT;

POS : INT;

END\_VAR

ILの例:

LD 'SUXYSI'

DELETE 2,3

ST Var1 (\* Result is 'SUSI' \*)

STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.3 FIND



ファンクションFINDは、文字列内の部分文字列を検索します。  
 FIND (STR1, STR2)の意味: STR1内で最初にSTR2が出現する先頭文字の位置を探します。STR1内にSTR2が見つからない場合は、OUT:=0になります。

## ファンクションFIND: INT

VAR\_INPUT

STR1 : STRING(255);

STR2 : STRING(255);

END\_VAR

## ILの例:

LD 'SUXYSI'

FIND 'XY'

ST Var1 (\* Result is 3 \*)

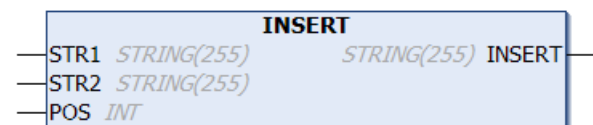
## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.4 INSERT



ファンクションINSERTは、与えられた文字列に定義した位置で他の文字列を挿入します。  
 INSERT (STR1, STR2, POS)の意味: STR1内の位置POSの後にSTR2を挿入します。

## ファンクションINSERT: STRING (255)

VAR\_INPUT

STR1 : STRING(255);

STR2 : STRING(255);

POS : INT;

END\_VAR

## ILの例:

LD 'SUSI'

INSERT 'XY',2

ST Var1 (\* Result is 'SUXYSI' \*)

## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.5 LEFT



ファンクションLEFTは、与えられた文字列の左側先頭文字を返します。  
LEFT (STR, SIZE)の意味：文字列STR内の一番左側からSIZE分の文字を取得します。

## ファンクションLEFT: STRING (255)

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

## ILの例:

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

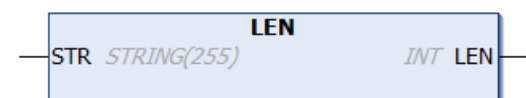
## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.6 LEN



ファンクションLENは、文字列の長さを返します。

## ファンクションLEN: INT

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

## ILの例:

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

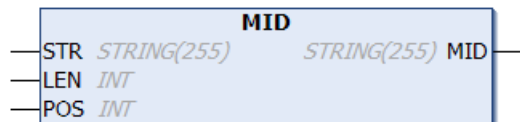
## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.7 MID



ファンクションMIDは、与えられた文字列内の部分文字列を返します。  
MID (STR, LEN, POS)の意味：文字列STRの位置POSの位置からLEN分の文字を取得します。

## ファンクションMID: STRING (255)

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

## ILの例:

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

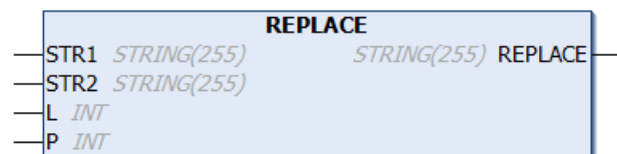
## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.8 REPLACE



ファンクションREPLACEは、与えられた文字列の部分文字列をほかの文字列に置き換えます。  
REPLACE (STR1, STR2, L, P)の意味：STR1のP位置からL文字をSTR2に置き換えます。

## ファンクションREPLACE: STRING (255)

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

## ILの例:

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

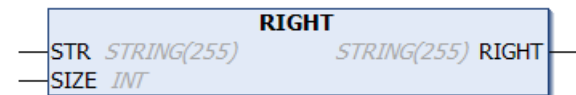
STの例:

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

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 4.9 RIGHT



ファンクションRIGHTは、与えられた文字列の右側先頭文字を返します。  
RIGHT (STR, SIZE)の意味: 文字列STR内の一番右側からSIZE分の文字を取得します。

ファンクションRIGHT: STRING (255)

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

ILの例:

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

STの例:

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

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5 Stringファンクション (WSTRING)

### 5.1 WCONCAT

```

WCONCAT
—STR1 WSTRING(255)      WSTRING(255) WCONCAT—
—STR2 WSTRING(255)

```

2つのWSTRINGの連結 (組み合わせ)

#### ファンクションWCONCAT: WSTRING (255)

```

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

```

ILの例:

```

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

```

STの例:

```

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

```

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

### 5.2 WDELETE

```

WDELETE
—STR WSTRING(255)      WSTRING(255) WDELETE—
—LEN INT
—POS INT

```

ファンクションWDELETEは、与えられた文字列から定義した箇所の部分文字列を削除します。入力変数STRはWSTRING型であり、LENとPOSはINT型です。このファンクションの戻り値はWSTRING型です。WDELETE (STR, LEN, POS)の意味: STRのPOS位置から開始するLEN分の文字を削除します。

#### ファンクションWDELETE: WSTRING (255)

```

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

```

ILの例:

```

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

```

STの例:

```

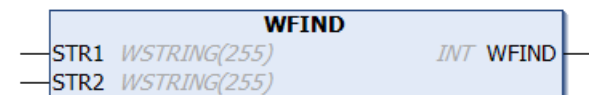
Var1 := WDELETE ("SUXYSI",2,3);

```

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.3 WFINDD



ファンクションWFINDDは、WSTRINGの一部を検索します。

WFINDD (STR1, STR2)の意味: STR1内で最初にSTR2が出現する先頭文字の位置を探します。STR1内にSTR2が存在しない場合は、OUT:=0が適用されます。

## ファンクションWFINDD: INT

```
VAR_INPUT
```

```
  STR1 : WSTRING(255);
```

```
  STR2 : WSTRING(255);
```

```
END_VAR
```

ILの例:

```
LD "SUXYSI"
```

```
WFINDD "XY"
```

```
ST Var1 (*Result is 3*)
```

STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.4 WINSERT



ファンクションWINSERTは、与えられた文字列の定義した位置に他の文字列を挿入します。

WINSERT (STR1, STR2, POS)の意味: STR1内の位置POSの後にSTR2を挿入します。

## ファンクションWINSERT: WSTRING (255)

```
VAR_INPUT
```

```
  STR1 : WSTRING(255);
```

```
  STR2 : WSTRING(255);
```

```
  POS  : INT;
```

```
END_VAR
```

ILの例:

```
LD "SUSI"
```

```
WINSERT "XY",2
```

```
ST Var1 (*Result is "SUXYSI"*)
```

STの例:

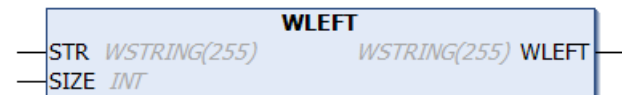


```
Var1 := WINSERT ("SUSI","XY",2);
```

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.5 WLEFT



ファンクションWLEFTは、与えられた文字列の左側先頭文字を返します。  
WLEFT (STR, SIZE)の意味：文字列STR内の一番左側からSIZE分の文字を取得します。

## ファンクションWLEFT: WSTRING (255)

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

## ILの例:

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

## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.6 WLEN



ファンクションWLENは、WSTRINGの長さを出力します。

## ファンクションWLEN: INT

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

## ILの例:

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

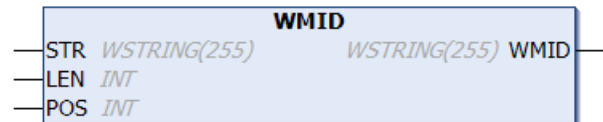
## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.7 WMID



ファンクションWMIDは、与えられた文字列内の部分文字列を返します。  
WMID (STR, LEN, POS)の意味：文字列STRの位置POSの位置からLEN分の文字を取得します。

## ファンクションWMID: WSTRING (255)

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

## ILの例:

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

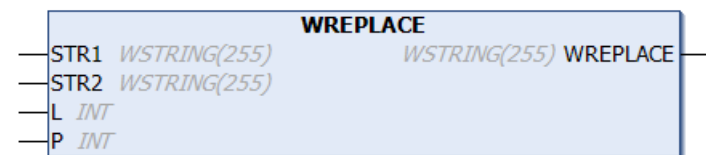
## STの例:

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

## 要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.8 WREPLACE



ファンクションWREPLACEは、与えられた文字列の部分文字列をほかの文字列に置き換えます。  
WREPLACE (STR1, STR2, L, P)の意味：STR1のP位置からL文字をSTR2に置き換えます。

## ファンクションWREPLACE: WSTRING (255)

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

## ILの例:

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

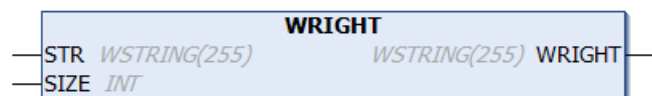
STの例:

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

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 5.9 WRIGHT



ファンクションWRIGHTは、与えられた文字列の右側先頭文字を返します。  
WRIGHT (STR, SIZE)の意味: 文字列STR内の一番右側からSIZE分の文字を取得します。

ファンクションWRIGHT: WSTRING (255)

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

ILの例:

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

STの例:

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

要件

開発環境	ターゲットシステムタイプ	必要ライブラリ
TwinCAT v3.0.0	PCまたはCX (x86)	Tc2_Standard

## 6 グローバル定数

### 6.1 ライブラリバージョン

すべてのライブラリは、固有のバージョンをもっています。このバージョンは、PLCライブラリレポジトリにも表示されています。

グローバル定数には、ライブラリバージョン情報が含まれています。

#### Global\_Version

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

**stLibVersion\_Tc2\_Standard**: Tc2\_標準ライブラリのバージョン情報(タイプ: ST\_LibVersion)。

既存のバージョンを必要なバージョンと比較するには、ファンクションF\_CmpLibVersionを使用します。



TwinCAT2では存在していた他のライブラリについてはすべて廃止されました。