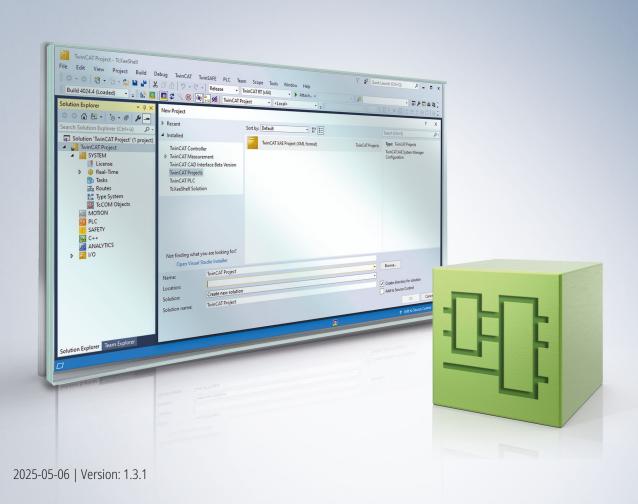
# **BECKHOFF** New Automation Technology

Manual | EN

## TE1000

TwinCAT 3 | PLC Library: Tc2\_Math





## **Table of contents**

1	Fore	word	5
	1.1	Notes on the documentation	5
	1.2	For your safety	6
	1.3	Notes on information security	7
2	Over	view	8
3	Fund	ctions	9
	3.1	CEIL	9
	3.2	FLOOR	10
	3.3	FRAC	12
	3.4	LMOD	13
	3.5	LTRUNC	14
	3.6	MODABS	15
	3.7	MODTURNS	16
4		olete functions]	
	4.1	F_GetVersionTcMath	18
5	Glob	pal constants	19
	5 1	Library version	10





### 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®, 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 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

© 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.

#### Third-party trademarks

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



## 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

#### **A DANGER**

Hazard with high risk of death or serious injury.

#### **A WARNING**

Hazard with medium risk of death or serious injury.

#### **A 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/secquide.

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 <a href="https://www.beckhoff.com/secinfo">https://www.beckhoff.com/secinfo</a>.



## 2 Overview

The Tc2\_Math library contains extended mathematical functions for TwinCAT PLC.

#### **Functions**

FLOOR [▶ 10]	The FLOOR function determines an integral value from a floating point number that is a fraction smaller than or equal that number.
FRAC [▶ 12]	The FRAC function determines the decimal component of a floating point number.
LMOD [▶ 13]	The LMOD function carries out a modulo division and returns the signed divide remainder.
<u>LTRUNC [▶ 14]</u>	The LTRUNC function determines the integral component of a floating point number.
MODABS [▶ 15]	The MODABS function carries out a modulo division and determines the unsigned modulo value within the modulo range.
MODTURNS [▶ 16]	The MODTURNS function carries out a modulo division and determines the signed integral component.
F GetVersionTcMath [ 18]	Returns the version information of the library

## 3 Functions

#### 3.1 **CEIL**



The CEIL function determines an integer value from a floating point number that is lower than or equal to that number. It is always rounded up to the larger number.

The resulting number is of type LREAL and is therefore not limited to the value range of integer variables.

#### **Examples**

FLOOR(2.8) = 3

FLOOR(-2.8) = -2

**FUNCTION CEIL: LREAL** 

#### Inputs

```
VAR_INPUT
lr_in : LREAL;
END VAR
```

Name	Туре	Description
lr_in	LREAL	Function parameters of type LREAL

#### Comparison with other rounding functions

Similar functions: <u>FLOOR</u> [▶ 10], <u>TRUNC</u>, <u>LTRUNC</u> [▶ 14]



Unlike CEIL, the <u>LTRUNC [> 14]</u> function always determines the integral part of a number. For positive values, this number is lower than or equal to the input parameter, for negative values it is greater than or equal to the input parameter.

#### **Example results of various rounding functions**

x	0	0.4	0.5	0.6	1	1.4	1.5	1.6	1.78
FLOOR(x)	0	0	0	0	1	1	1	1	1
CEIL(x)	0	1	1	1	1	2	2	2	2
LTRUNC(x)	0	0	0	0	1	1	1	1	1
TO_LINT(x)	0	0	1	1	1	1	2	2	2

X	-1.78	-1.6	-1.5	-1.4	-1	-0.6	-0.5	-0.4
FLOOR(x)	-2	-2	-2	-2	-1	-1	-1	-1
CEIL(x)	-1	-1	-1	-1	-1	0	0	0
LTRUNC(x)	-1	-1	-1	-1	-1	0	0	0
TO_LINT(x)	-2	-2	-2	-1	-1	-1	-1	0

If you do not want to round to an integer value and instead want to round to a specific decimal place, this can be achieved by multiplying and dividing by a power of ten. The following example results show the rounding to the first decimal place.



x	0	0.04	0.05	0.06	0.1	0.14	0.15	0.16	0.178
FLOOR(x * 10) / 10	0	0	0	0	0.1	0.1	0.1	0.1	0.1
CEIL(x * 10) / 10	0	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2
LTRUNC(x * 10) / 10	0	0	0	0	0.1	0.1	0.1	0.1	0.1
TO_LINT(x * 10) / 10.0	0	0	0.1	0.1	0.1	0.1	0.2	0.2	0.2

х	-0.178	-0.16	-0.15	-0.14	-0.1	-0.06	-0.05	-0.04
FLOOR(x * 10) / 10	-0.2	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1
CEIL(x * 10) / 10	-0.1	-0.1	-0.1	-0.1	-0.1	0	0	0
LTRUNC(x * 10) / 10	-0.1	-0.1	-0.1	-0.1	-0.1	0	0	0
TO_LINT(x * 10) / 10.0	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1	0

#### **Prerequisites**

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1 >= 4026.13	PC or CX (x86)	Tc2_Math >= v3.5.1.0

## 3.2 FLOOR



The FLOOR function determines an integer value from a floating point number that is lower than or equal to that number. It is always rounded down to the lower number.

The resulting number is of type LREAL and is therefore not limited to the value range of an integer variable.

#### **Examples**

FLOOR(2.8) = 2

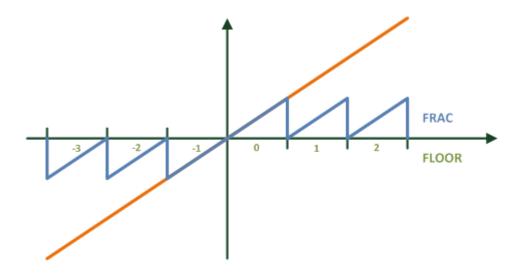
FLOOR(-2.8) = -3

#### **FUNCTION FLOOR: LREAL**

#### Inputs

VAR\_INPUT lr\_in : LREAL; END VAR

Name	Туре	Description
lr_in	LREAL	Function parameters of type LREAL



#### Comparison with other rounding functions

Similar functions: <u>CEIL</u> [▶ 9], <u>TRUNC</u>, <u>LTRUNC</u> [▶ 14]



Unlike FLOOR, the <u>LTRUNC [14]</u> function always determines the integral part of a number. For positive values, this number is lower than or equal to the input parameter, for negative values it is greater than or equal to the input parameter.

#### **Example results of various rounding functions**

x	0	0.4	0.5	0.6	1	1.4	1.5	1.6	1.78
FLOOR(x)	0	0	0	0	1	1	1	1	1
CEIL(x)	0	1	1	1	1	2	2	2	2
LTRUNC(x)	0	0	0	0	1	1	1	1	1
TO_LINT(x)	0	0	1	1	1	1	2	2	2

x	-1.78	-1.6	-1.5	-1.4	-1	-0.6	-0.5	-0.4
FLOOR(x)	-2	-2	-2	-2	-1	-1	-1	-1
CEIL(x)	-1	-1	-1	-1	-1	0	0	0
LTRUNC(x)	-1	-1	-1	-1	-1	0	0	0
TO_LINT(x)	-2	-2	-2	-1	-1	-1	-1	0

If you do not want to round to an integer value and instead want to round to a specific decimal place, this can be achieved by multiplying and dividing by a power of ten. The following example results show the rounding to the first decimal place.

X	0	0.04	0.05	0.06	0.1	0.14	0.15	0.16	0.178
FLOOR(x * 10) / 10	0	0	0	0	0.1	0.1	0.1	0.1	0.1
CEIL(x * 10) / 10	0	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2
LTRUNC(x * 10) / 10	0	0	0	0	0.1	0.1	0.1	0.1	0.1
TO_LINT(x * 10) / 10.0	0	0	0.1	0.1	0.1	0.1	0.2	0.2	0.2

x	-0.178	-0.16	-0.15	-0.14	-0.1	-0.06	-0.05	-0.04
FLOOR(x * 10) / 10	-0.2	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1
CEIL(x * 10) / 10	-0.1	-0.1	-0.1	-0.1	-0.1	0	0	0
LTRUNC(x * 10) / 10	-0.1	-0.1	-0.1	-0.1	-0.1	0	0	0
TO_LINT(x * 10) / 10.0	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1	0



#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math

#### Also see about this

LTRUNC [▶ 14]

## 3.3 FRAC



The FRAC function determines the decimal component of a floating point number.

#### **Examples**

FRAC(2.8) = 0.8

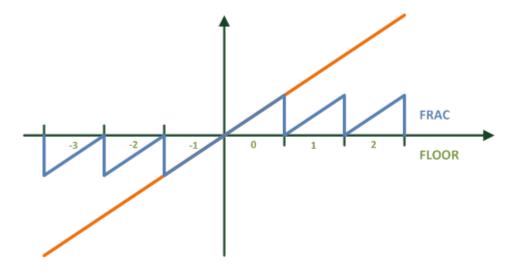
FRAC(-2.8) = -0.8

#### **FUNCTION FRAC: LREAL**

#### Inputs

VAR\_INPUT lr\_in : LREAL; END\_VAR

Name	Туре	Description
Ir in	LREAL	Function parameters of type LREAL

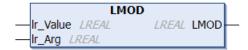


#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math



#### 3.4 **LMOD**



The LMOD function carries out a modulo division and returns the signed divide remainder.

#### **Examples**

LMOD(400.56, 360) = 40.56

LMOD(-400.56, 360) = -40.56

Similar functions: MOD, MODABS [▶ 15]



Unlike MOD, the LMOD function operates with floating point variables and also determines non-integer remainders.

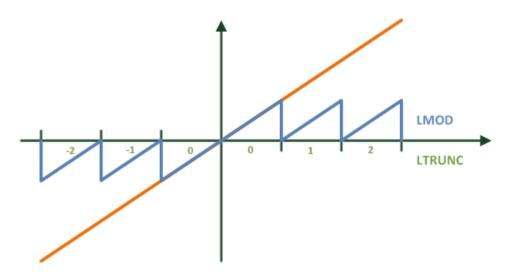
In the context of NC axes, modulo values are usually used unsigned. These can be calculated with the MODABS [▶\_15] function.

#### **FUNCTION LMOD: LREAL**

#### Inputs

VAR\_INPUT
lr\_Value : LREAL;
lr\_Arg : LREAL;
END\_VAR

Name	Туре	Description
Ir_Value	LREAL	Input value
Ir_Arg	LREAL	Modulo range



#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math



## 3.5 LTRUNC



The LTRUNC function determines the integral part of a floating point number.

Unlike TRUNC, the result from LTRUNC is of type LREAL and is therefore not limited to the value range of integer variables.

#### **Examples**

LTRUNC(2.8) = 2

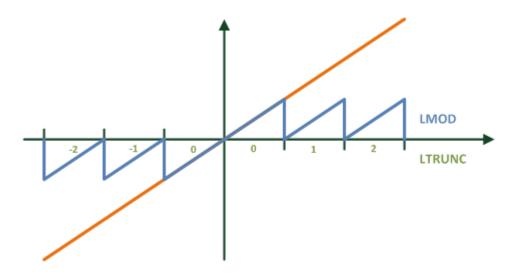
LTRUNC(-2.8) = -2

#### **FUNCTION LTRUNC: LREAL**

#### Inputs

VAR\_INPUT lr\_in : LREAL; END\_VAR

Name	Туре	Description
lr in	IRFAI	Function parameters of type I RFAI



#### Comparison with other rounding functions

Similar functions: TRUNC,  $\underline{CEIL}$  [ $\triangleright$  9],  $\underline{FLOOR}$  [ $\triangleright$  10]

#### **Example results of various rounding functions**

x	0	0.4	0.5	0.6	1	1.4	1.5	1.6	1.78
FLOOR(x)	0	0	0	0	1	1	1	1	1
CEIL(x)	0	1	1	1	1	2	2	2	2
LTRUNC(x)	0	0	0	0	1	1	1	1	1
TO_LINT(x)	0	0	1	1	1	1	2	2	2



x	-1.78	-1.6	-1.5	-1.4	-1	-0.6	-0.5	-0.4
FLOOR(x)	-2	-2	-2	-2	-1	-1	-1	-1
CEIL(x)	-1	-1	-1	-1	-1	0	0	0
LTRUNC(x)	-1	-1	-1	-1	-1	0	0	0
TO_LINT(x)	-2	-2	-2	-1	-1	-1	-1	0

If you do not want to round to an integer value and instead want to round to a specific decimal place, this can be achieved by multiplying and dividing by a power of ten. The following example results show the rounding to the first decimal place.

x	0	0.04	0.05	0.06	0.1	0.14	0.15	0.16	0.178
FLOOR(x * 10) / 10	0	0	0	0	0.1	0.1	0.1	0.1	0.1
CEIL(x * 10) / 10	0	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2
LTRUNC(x * 10) / 10	0	0	0	0	0.1	0.1	0.1	0.1	0.1
TO LINT(x * 10) / 10.0	0	0	0.1	0.1	0.1	0.1	0.2	0.2	0.2

X	-0.178	-0.16	-0.15	-0.14	-0.1	-0.06	-0.05	-0.04
FLOOR(x * 10) / 10	-0.2	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1
CEIL(x * 10) / 10	-0.1	-0.1	-0.1	-0.1	-0.1	0	0	0
LTRUNC(x * 10) / 10	-0.1	-0.1	-0.1	-0.1	-0.1	0	0	0
TO_LINT(x * 10) / 10.0	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1	-0.1	0

#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math

#### Also see about this

CEIL [▶ 9]

FLOOR [▶ 10]

### 3.6 MODABS



The MODABS function performs a modulo division and determines the unsigned modulo value within the modulo range.

#### **Examples**

MODABS(400.56, 360) = 40.56

MODABS(-400.56, 360) = 319.44

Similar functions: MOD, LMOD [▶ 13]



The MODABS function can be used to calculate the modulo set position of an NC axis from its absolute set position.

ModuloSetPosition := MODABS( NcToPlc.fPosSoll, 360 );

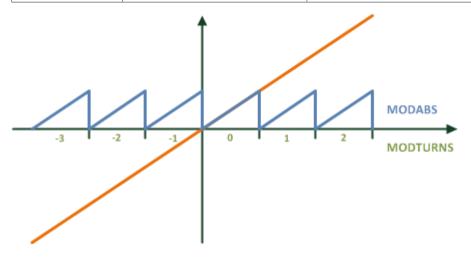


#### **FUNCTION MODABS: LREAL**

#### Inputs

VAR\_INPUT
lr\_val : LREAL;
lr\_mod : LREAL;
END\_VAR

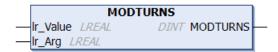
Name	Туре	Description
lr_val	LREAL	Input value
lr_mod	LREAL	Modulo range



#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math

#### 3.7 MODTURNS



The MODTURNS function carries out a modulo division and determines the signed integral component (modulo periods, modulo rotations).



#### Value range of DINT



If the value of the result of MODTURNS lies outside of the value range of DINT, an undefined result will be delivered.

#### **Examples**

MODTURNS (800.56, 360) = 2

MODTURNS (-400.56, 360) = -2



The MODTURNS function can be used to calculate the number of modulo rotations of an NC axis from its absolute set position.

ModuloSetTurns := MODTURNS ( NcToPlc.fPosSoll, 360 );

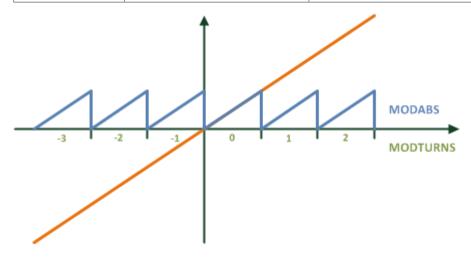


#### **FUNCTION MODTURNS: DINT**

## Inputs

VAR\_INPUT lr\_Value : LREAL; lr\_Arg : LREAL; END\_VAR

Name	Туре	Description
Ir_Value	LREAL	Input value
Ir_Arg	LREAL	Modulo range



#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math



## 4 [obsolete functions]

## 4.1 F\_GetVersionTcMath

```
F_GETVERSIONTCMATH

__nVersionElement : INT F_GetVersionTcMath : UINT___
```

This function can be used to read PLC library version information.

#### FUNCTION F\_GetVersionTcMath: UINT

## Inputs

VAR\_INPUT nVersionElement : INT; END VAR

Name	Туре	Description
nVersionElement	INT	Version element to be read. Possible parameters:
		• 1: major number;
		• 2: minor number;
		3: revision number

#### Requirements

Development environment	Target system type	PLC libraries to include
TwinCAT v3.0.0	PC or CX (x86)	Tc2_Math



## 5 Global constants

## 5.1 Library version

All libraries have a certain version. The version is indicated in the PLC library repository, for example. A global constant contains the information about the library version:

#### Global\_Version

```
VAR_GLOBAL CONSTANT
     stLibVersion_Tc2_Math : ST_LibVersion;
END VAR
```

Name	Туре	Description
stLibVersion_Tc2	ST_LibVersion	Version number of the Tc2_Math library (type:
_Math		ST_LibVersion)

To see if you have the version you need, use the function F\_CmpLibVersion (defined in Tc2\_System).



All other options for comparing library versions, which you may know from TwinCAT 2, are outdated!



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

