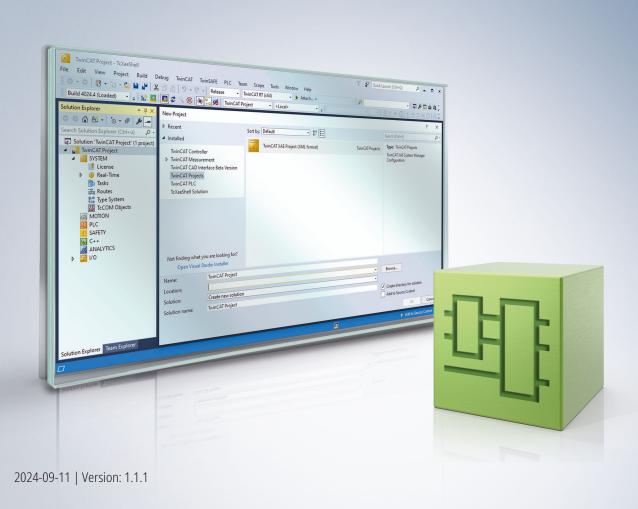
# **BECKHOFF** New Automation Technology

Manual | EN

# TE1000

TwinCAT 3 | PLC Library: Tc2\_Coupler





## **Table of contents**

1	Fore	word	. 5
	1.1	Notes on the documentation	. 5
	1.2	For your safety	. 5
	1.3	Notes on information security	. 7
2	Ove	rview	. 8
3	Fund	ction blocks	. 9
	3.1	ReadWriteTerminalReg	. 9
	3.2	CouplerReset	11
	3.3	FB_ReadCouplerDiag	12
	3.4	FB_ReadCouplerRegs	14
	3.5	FB_WriteCouplerRegs	15
4	[obs	olete functions]	17
4	[obs 4.1	olete functions] F_GetVersionTcPlcCoupler	
	4.1		17
	4.1	F_GetVersionTcPlcCoupler	17 <b>18</b>
	4.1 Data	F_GetVersionTcPlcCouplertypes	17 <b>18</b> 18
	4.1 <b>Data</b> 5.1	F_GetVersionTcPlcCoupler  types  PLCINTFSTRUCT	17 <b>18</b> 18 18
	4.1 <b>Data</b> 5.1 5.2	F_GetVersionTcPlcCoupler  types  PLCINTFSTRUCT  E_CouplerErrType	17 <b>18</b> 18 18 18
	4.1 <b>Data</b> 5.1 5.2 5.3	F_GetVersionTcPlcCoupler  types  PLCINTFSTRUCT  E_CouplerErrType  ST_CouplerDiag	17 18 18 18 18
	4.1 <b>Data</b> 5.1 5.2 5.3 5.4	F_GetVersionTcPlcCoupler  types  PLCINTFSTRUCT  E_CouplerErrType  ST_CouplerDiag  ST_CouplerReg	17 18 18 18 18 19
5	4.1 <b>Data</b> 5.1 5.2 5.3 5.4 5.5 5.6	F_GetVersionTcPlcCoupler  types  PLCINTFSTRUCT  E_CouplerErrType  ST_CouplerDiag  ST_CouplerReg  ST_CouplerTable	17 18 18 18 18 18 19 19





### 1 Foreword

#### 1.1 Notes on the documentation

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

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

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

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

#### **Disclaimer**

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

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

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

#### **Trademarks**

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

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

#### **Patents**

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

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



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

#### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

## 1.2 For your safety

#### Safety regulations

Read the following explanations for your safety.

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



#### **Exclusion of liability**

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

#### Personnel qualification

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

#### Signal words

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

#### Personal injury warnings

#### **▲ DANGER**

Hazard with high risk of death or serious injury.

#### **⚠ WARNING**

Hazard with medium risk of death or serious injury.

#### **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\_Coupler library can be used for the following products: BKxxx couplers, KLxxx terminals, KSxxxx terminals and KMxxx modules.

The Tc2\_Coupler library contains function blocks that provide convenient access to registers in the terminals via the terminal's control/status byte (register communication) and for communication with the Beckhoff couplers via the 2-byte PLC interface. The function blocks can, for instance, be used for parameterization of the terminals by way of the fieldbus.

Only the intelligent terminals have a register structure. The intelligent terminals include, for example, all the analog input and output terminals. The terminal's status/control byte is only visible in the process image if the terminal has been mapped as a complex terminal. Each terminal channel has its own register structure with a maximum of 64 registers. Under a compact mapping the control/status bytes are not visible in the process image.

For register access via the 2-byte PLC interface it is also necessary for the status and control word variables of the PLC interface to be mapped into the process image. In some fieldbusses (Lightbus, Profibus) this can be configured for the particular coupler in the TwinCAT System Manager, but in others (e.g. Interbus S) special configuration software is required for the job (e.g. KS2000). The status and control variables are linked to the function block's corresponding input and output variables.

If any changes made to the registers are to be stored permanently, the power supply to the coupler must be interrupted.

#### NOTICE

#### No cyclic access!

When called, the function blocks of the library carry out write/read access to the registers in the terminals or in the couplers. If they are called cyclically, the EEPROM memory may be destroyed. The function blocks were basically developed to facilitate configuration of the terminals/couplers or fault diagnosis from within the PLC program.

#### **Contents of the Library**

Name	Description
ReadWriteTerminalReg [▶ 9]	Access registers in the terminal via the terminal's control/status byte (register communication)
CouplerReset [▶ 11]	Reset the coupler via the 2-byte PLC interface
FB ReadCouplerDiag [ 12]	Read coupler diagnosis (flash code)
FB ReadCouplerRegs [ 14]	Read coupler registers
FB WriteCouplerRegs [ 15]	Write coupler registers
F GetVersionTcPlcCoupler [ 17]	Returns library version info

### 3 Function blocks

## 3.1 ReadWriteTerminalReg

```
ReadWriteTerminalReg

STATE BYTE BYTE CTRL

DATAIN WORD WORD DATAOUT

REGNO BYTE BOOL BUSY

READ BOOL BOOL BOOL ERR

WRITE BOOL UDINT ERRID

TMOUT TIME WORD CURREGVALUE

NEWREGVALUE WORD
```

The function block ReadWriteTerminalReg enables convenient access to the terminal's registers via the control/status byte of the terminal channel (register communication). In the standard operation mode, the data inputs and outputs of the intelligent terminals (e.g. an analog output terminal) are used to exchange the analog output data. A handshake via the control/status byte permits register access. The data input and output variables are used here to transfer the register values. A positive edge at the READ or WRITE input causes the register with number REGNO to be read or written to. Write protection of the register is disabled by the function block for a write access and enabled once more afterwards. With a write access to a register, the new register value is written first and then read. The value read is available at the output CURRREGVALUE. If changes made to the register values are to be stored permanently, the power supply to the coupler must be interrupted. The variables STATE, DATAIN, CTRL and DATAOUT must be linked in the TwinCAT System Manager to the corresponding I/O variables in the terminal channel.

#### Inputs

```
VAR_INPUT

STATE : BYTE;
DATAIN : WORD;
REGNO : BYTE;
READ : BOOL;
WRITE : BOOL;
TMOUT : TIME;
NEWREGVALUE : WORD;
END_VAR
```

Name	Туре	Description
STATE	BYTE	Status byte of the terminal channel.
DATAIN	WORD	Data input word of the terminal channel.
REGNO	BYTE	Number of the register to read or write.
READ	BOOL	A positive edge at this input activates the function block, and the current register value is read. If successful, the register value is available in the output variable CURREGVALUE.
WRITE	BOOL	A positive edge at this input activates the function block, and the value in the input variable NEWREGVALUE is written into the register REGNO. The current value of the register is then read and, if successful, is available in the output variable CURREGVALUE.
TMOUT	TIME	Specifies the time-out period that must not be exceeded when the function is executed.
NEWREGVALUE	WORD	Data word to be written to the register numbered REGNO during a write access.

### Outputs

```
VAR_OUTPUT

CTRL : BYTE;

DATAOUT : WORD;

BUSY : BOOL;

ERR : BOOL;
```



ERRID : UDINT; CURREGVALUE : WORD; END VAR

Name	Туре	Description
CTRL	BYTE	Control byte of the terminal channel.
DATAOUT	WORD	Data output word of the terminal channel.
BUSY	BOOL	When the function block is activated, this output is set and remains set until the function is completed.
ERR	BOOL	If an error occurs during the execution of the function, this output is set after the BUSY output is reset.
ERRID	UDINT	If the ERR output is set, this parameter supplies the error number.

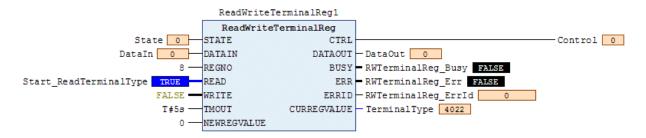
Error number	Error description
0	No error
0x100	Timeout error. The permitted execution time has been exceeded
0x200	Parameter error (e.g. an invalid register number)
0x300	The value read differs from the value written (write access to this register may not be permitted or may have failed)

Name	Туре	Description
CURREGVALUE	WORD	This variable provides the current register value after a
		successful read or write access.

#### **Examples of calls in FBD:**

#### **Example 1**

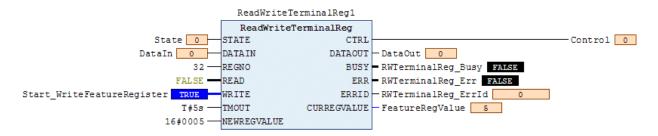
```
VAR
ReadWriteTerminalReg1 : ReadWriteTerminalReg;
State AT%I* : BYTE;
Control AT%Q* : BYTE;
DataIn AT%I* : WORD;
DataOut AT%Q* : WORD;
Start_ReadTerminalType : BOOL;
Start_WriteFeatureRegister : BOOL;
RWTerminalReg_Busy : BOOL;
RWTerminalReg_Err : BOOL;
RWTerminalReg_Err : BOOL;
RWTerminalReg_Err : BOOL;
RWTerminalReg_Err : BOOL;
RWTerminalReg_ErrId : UDINT;
TerminalType : WORD;
FeatureRegValue : WORD;
END_VAR
```



In Example 1 the terminal identifier is read from register 8 of an analog output terminal. The variables *State*, *Control*, *DataIn* and *DataOut* are linked to the terminal's corresponding I/O variables in the TwinCAT System Manager. The terminal identifier is KL**4022**..

#### Example 2





In Example 2 the user-scaling is activated in the feature register (register 32) of a KL4022 analog output terminal. The new value in the feature register is then read by the function block, and can be checked through the output variable **CURREGVALUE**.

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 3.2 CouplerReset

```
CouplerReset

STATE PLCINTFSTRUCT PLCINTFSTRUCT CONTROL

START BOOL
TMOUT TIME

BOOL ERR
UDINT ERRID
```

The function block CouplerReset can be used to execute a reset of the coupler via the 2-byte PLC interface. During a coupler reset, for example, the current terminal configuration is read in again by the coupler via the K-bus (terminal bus) and communication on the K-bus is reinitialized. Existing K-bus error messages for the coupler are reset. The STATE and CONTROL variables are used to perform a handshake with the coupler during the execution of the function block. These variables must therefore be linked to the Control/Status variables of the 2-byte PLC interface in the TwinCAT System Manager.

### Inputs

```
VAR_INPUT
STATE : PLCINTFSTRUCT;
START : BOOL;
TMOUT : TIME;
END VAR
```

Name	Туре	Description
STATE	PLCINTFSTRUCT [▶ 18]	Status word of the 2-byte PLC interface.
START	BOOL	The function block is enabled by a positive edge at this input.
TMOUT	TIME	Specifies the time-out period that must not be exceeded when the function is executed.

### Outputs

```
VAR_OUTPUT

CONTROL : PLCINTFSTRUCT;

BUSY : BOOL;

ERR : BOOL;

ERRID : UDINT;

END VAR
```



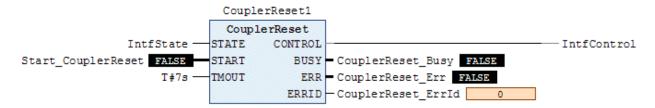
Name	Туре	Description
CONTROL	PLCINTFSTRUCT [▶ 18]	Control word of the 2-byte PLC interface.
BUSY	BOOL	When the function block is activated, this output is set and remains set until the function is completed.
ERR	BOOL	If an error occurs during the execution of the function, this output is set after the BUSY output is reset.
ERRID	UDINT	If the ERR output is set, this parameter supplies the error number.

Error number	Error description
0	No error
0x100	Error during initialization of communication via the 2-byte PLC interface
0x200	Error during communication
0x300	Timeout error. The permitted execution time has been exceeded
0x400	Incorrect parameter value for register number
0x500	Incorrect parameter value for table number

#### Example of a call in FBD:

```
VAR

IntfState AT%I* : PLCINTFSTRUCT;
IntfControl AT%Q* : PLCINTFSTRUCT;
CouplerReset1 : CouplerReset;
Start_CouplerReset : BOOL;
CouplerReset_Busy : BOOL;
CouplerReset_Err : BOOL;
CouplerReset_Err : BOOL;
END VAR
```



The variables *IntfState* and *IntfControl* are linked with corresponding I/O variables in the TwinCAT System Manager.

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 3.3 FB\_ReadCouplerDiag



The function block FB\_ReadCouplerDiag allows reading of the first and second flashing sequences of the error LED on the coupler when a terminal bus or coupler error occurs. The data is transferred to the PLC via the 2-byte PLC interface. This only works if communication is maintained via the fieldbus. It must be possible to transfer data from the coupler to the PLC via the fieldbus without errors. To detect a coupler error that has occurred, the status byte of the coupler can be queried cyclically in the PLC and the function block is activated if an error occurs.



#### Inputs

VAR INPUT

vak\_INPUT
stState : PLCINTFSTRUCT;
bExecute : BOOL;
tTimeout : TIME;
END\_VAR

Name	Туре	Description
stState	PLCINTFSTRUCT [▶ 18]	Status word of the 2-byte PLC interface.
bExecute	BOOL	The function block is enabled by a positive edge at this input.
tTimeout	TIME	Specifies the time-out period that must not be exceeded when the function is executed.

### Outputs

VAR\_OUTPUT

var\_output
stCtrl : PLCINTFSTRUCT;
bBusy : BooL;
bError : BOOL;
nErrId : UDINT;
stDiag : ST\_CouplerDiag;
END\_VAR

Name	Туре	Description
stCtrl	PLCINTFSTRUCT [▶ 18]	Control word of the 2-byte PLC interface.
bBusy	BOOL	When the function block is activated, this output is set and remains set until the function is completed.
bError	BOOL	If an error occurs during the execution of the function, this output is set after the bBusy output has been reset.
nErrId	UDINT	Supplies the error number when the bError output is set.

Error number	Error description
0	No error
0x100	Error at initialisation of the communication via the 2 byte PLC interface
0x200	Error during communication
0x300	Timeout-Error. The permitted execution time was exceeded
0x400	Wrong parameter value at register number
0x500	Wrong parameter value at table number

Name	Туре	Description
stDiag	<u> </u>	Structure containing the coupler's diagnostic information (error type, and the first and second coupler flashing sequences).

#### Requirements

Development environment	Target system type	PLC libraries to include (cate- gory group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)



## 3.4 FB\_ReadCouplerRegs

```
FB_ReadCouplerRegs

stState PLCINTFSTRUCT StCtrl
nTerminal BYTE BOOL bBusy
nTable BYTE BOOL bError
nStartReg BYTE UDINT nErrId
nEndReg BYTE ST_CouplerTable stCouplerTable
bExecute BOOL
tTimeout TIME
```

This function block FB\_ReadCouplerRegs provides read access to the table register in the coupler and the registers of the intelligent terminals. The coupler itself is referred to as terminal 0 (null). All the other terminals in the terminal block, with the exception of passive terminals (such as power supply terminals), are counted in ascending sequence (beginning with 1). All registers or only a subrange (between *nStartReg* and *nEndReg*) can be read. Several seconds are required to read all registers (0.255) of a coupler table. Register values that have been successfully read are found in the structure *stCouplerTable*. The structure is an array of high and low bytes. Each array element corresponds to a register value (e.g.: stCouplerTable[ 5 ] == Register 5).

#### Inputs

```
VAR_INPUT

stState : PLCINTFSTRUCT;

nTerminal : BYTE:= TERM_COUPLER;

nTable : BYTE;

nStartReg : BYTE;

nEndReg : BYTE;

bExecute : BOOL;

tTimeout : TIME;

END VAR
```

Name	Туре	Description
stState	PLCINTFSTRUCT [▶ 18]	Status word of the 2-byte PLC interface.
nTerminal	ВҮТЕ	Terminal number whose table register is to be accessed. The coupler has terminal number null. Passive terminals are not to be counted.
nTable	ВУТЕ	Table number whose register values are to be read. Intelligent terminals only have one table for each terminal channel. A 4-channel terminal has the following table numbers: 0-3. However, an intelligent terminal only has a maximum of 64 register values per terminal channel.
nStartReg	BYTE	The number of the first register to be read.
nEndReg	BYTE	The number of the last register to be read.
bExecute	BOOL	The function block is enabled by a positive edge at this input.
tTimeout	TIME	Specifies the time-out period that must not be exceeded when the function is executed.

### Outputs

```
VAR_OUTPUT
stCtrl : PLCINTFSTRUCT;
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
stCouplerTable : ST_CouplerTable;
END_VAR
```



Name	Туре	Description
stCtrl	PLCINTFSTRUCT [▶ 18]	Control word of the 2-byte PLC interface.
bBusy	BOOL	When the function block is activated, this output is set and remains set until the function is completed.
bError	BOOL	If an error occurs during the execution of the function, this output is set after the bBusy output has been reset.
nErrld	UDINT	Supplies the error number when the bError output is set.

Error number	Error description
0	No error
0x100	Error at initialisation of the communication via the 2 byte PLC interface
0x200	Error during communication
0x300	Timeout-Error. The permitted execution time was exceeded
0x400	Wrong parameter value at register number
0x500	Wrong parameter value at table number

Name	Туре	Description
stCouplerTable	<u> </u>	Structure with the read register values of the terminal or coupler.

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 3.5 FB\_WriteCouplerRegs

```
FB_WriteCouplerRegs

stState PLCINTFSTRUCT PLCINTFSTRUCT stCtrl
nTerminal BYTE BOOL bBusy
nTable BYTE BOOL bError
nStartReg BYTE
stCouplerTable ST_CouplerTable
bExecute BOOL
tTimeout TIME
```

The function block FB\_WriteCouplerRegs can be used for write access to the coupler's table registers and the registers of the intelligent terminals. The coupler itself is referred to as terminal 0 (null). All the other terminals in the terminal block, with the exception of passive terminals (such as power supply terminals), are counted in ascending sequence (beginning with 1). All registers or only a subrange (between *nStartReg* and *nEndReg*) can be written to. Several seconds are required to write all registers (0.255) of a coupler table. The register values to be written are located in the structure *stCouplerTable*. The structure is an array of high and low bytes. Each array element corresponds to a register value (e.g.: *stCouplerTable*[ 5 ] == Register 5).

#### Inputs

```
VAR_INPUT

stState : PLCINTFSTRUCT;
nTerminal : BYTE := TERM_COUPLER;
nTable : BYTE;
nStartReg : BYTE;
nEndReg : BYTE;
bExecute : BOOL;
stCouplerTable : ST_CouplerTable;
tTimeout : TIME;

END_VAR
```



Name	Туре	Description
stState	PLCINTFSTRUCT [▶ 18]	Status word of the 2-byte PLC interface.
nTerminal	ВҮТЕ	Terminal number whose table register is to be accessed. The coupler has terminal number null. Passive terminals are not to be counted.
nTable	ВҮТЕ	Table number whose register values are to be read. Intelligent terminals only have one table for each terminal channel. A 4-channel terminal has the following table numbers: 0-3. However, an intelligent terminal only has a maximum of 64 register values per terminal channel.
nStartReg	BYTE	The number of the first register to be written.
nEndReg	BYTE	The number of the last register to be written.
bExecute	BOOL	The function block is enabled by a positive edge at this input.
stCouplerTable	ST CouplerTable [▶ 19]	Array of the register values to be written.
tTimeout	TIME	Specifies the time-out period that must not be exceeded when the function is executed.

### Outputs

VAR\_OUTPUT

stCtrl : PLCINTFSTRUCT;
bBusy : BOOL;
bError : BOOL;
nErrId : UDINT;
FND VAR

Name	Туре	Description
stCtrl	PLCINTFSTRUCT [▶ 18]	Control word of the 2-byte PLC interface.
bBusy	BOOL	When the function block is activated, this output is set and remains set until the function is completed.
bError	BOOL	If an error occurs during the execution of the function, this output is set after the bBusy output has been reset.
nErrld	UDINT	Supplies the error number when the bError output is set.

Error number	Error description
0	No error
0x100	Error at initialisation of the communication via the 2 byte PLC interface
0x200	Error during communication
0x300	Timeout-Error. The permitted execution time was exceeded
0x400	Wrong parameter value at register number
0x500	Wrong parameter value at table number

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)



## 4 [obsolete functions]

## 4.1 F\_GetVersionTcPlcCoupler

This function is obsolete and should not be used any longer. Use the global constant stLibVersion Tc2 Coupler [ > 20] to read version information from the plc library.

This function reads version information from the plc library.

#### FUNCTION F\_GetVersionTcPlcCoupler : UINT

```
VAR_INPUT
nVersionElement: INT;
END_VAR
```

**nVersionElement**: Version element, that is to be read. Possible parameters:

- 1: major number;
- 2 : minor number;
- 3 : revision number;

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)



## 5 Data types

### 5.1 PLCINTFSTRUCT

```
TYPE PLCINTFSTRUCT:
STRUCT
Byte0: BYTE;
Byte1: BYTE;
END_STRUCT
END_TYPE
```

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 5.2 E\_CouplerErrType

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 5.3 ST\_CouplerDiag

Name	Туре	Description
eErrType	E_CouplerErrType [▶ 18]	General error type
stFlashCode	ST_FlashCode [▶ 19]	The first and second sequence of the flashing code

#### Requirements

Development environment	J , , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 5.4 ST\_CouplerReg

```
TYPE ST_CouplerReg
STRUCT
Lo : BYTE;
Hi : BYTE;
END_STRUCT
END_TYPE
```



A Coupler register has the size of one word. The parameterisation and the configuration of the coupler is discarded in the register.

#### Requirements

Development environment	, , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 5.5 ST\_CouplerTable

```
TYPE ST_CouplerTable : ARRAY[ 0..255 ] OF ST_CouplerReg;
END_TYPE
```

The parameters and configuration of the coupler are stored in the coupler's EEPROM. The memory is divided into tables (Type <u>ST\_CouplerReq [▶ 18]</u>). Each table possesses a maximum of 256 registers.

#### Requirements

Development environment	, , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)

## 5.6 ST\_FlashCode

```
TYPE ST_FlashCode:
STRUCT
ErrType: WORD;
ErrLocation: WORD;
END_STRUCT
END_TYPE
```

Name	Туре	Description
ErrType		Error type. Corresponds to the first flashing sequence of the coupler.
ErrLocation		Error location. Corresponds to the second flashing sequence of the coupler. (The position of the last terminal before the error location. Passive terminals are not counted.)

#### Requirements

Development environment	, , , , , , , , , , , , , , , , , , ,	PLC libraries to include (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Coupler (IO)



## **6** Global Constants

## 6.1 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_Coupler : ST_LibVersion;
END VAR
```

Name	Туре	Description
stLibVersion	ST_LibVersion	Version information of the Tc2_Coupler library
_Tc2_Coupler		

To check whether the version you have is the version you need, use the function F\_CmpLibVersion (defined in the Tc2\_System library).



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

