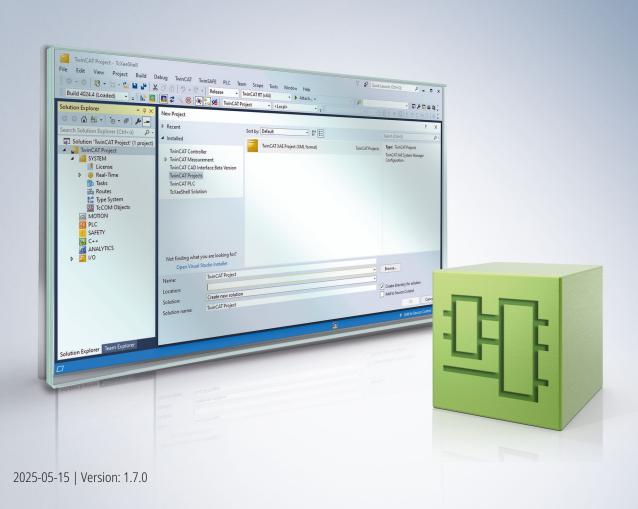
# **BECKHOFF** New Automation Technology

Manual | EN

# TE1000

TwinCAT 3 | PLC Libary: Tc2\_EnOcean





# **Table of contents**

1	Fore	word		5
	1.1	Notes o	n the documentation	5
	1.2	For you	r safety	5
	1.3	Notes o	n information security	7
2	Intro	duction .		8
3	EnO	cean®		9
	3.1	Range p	olanning	9
	3.2	Approva	al of EnOcean <sup>®</sup> radio technology	10
4	Prog	ramming	J	11
	4.1	POUs		12
		4.1.1	KL6021-0023	13
		4.1.2	KL6581	23
	4.2	DUTs		37
		4.2.1	KL6021-0023	38
		4.2.2	KL6581	40
	4.3	Integrati	ion into TwinCAT	46
		4.3.1	KL6581 with CX5120	46
5	Appe	endix		49
	5.1	Support	and Service	40



## 1 Foreword

### 1.1 Notes on the documentation

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

The documentation and the following notes and explanations must be complied with when installing and commissioning the components.

The trained specialists must always use the current valid documentation.

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

#### **Disclaimer**

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

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

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

#### **Trademarks**

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

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



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

#### Copyright

© 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: https://www.beckhoff.com/trademarks.

# 1.2 For your safety

#### Safety regulations

Read the following explanations for your safety.

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

#### **Exclusion of liability**

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



#### Personnel qualification

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

#### Signal words

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

#### Personal injury warnings

#### **▲ DANGER**

Hazard with high risk of death or serious injury.

#### **⚠ WARNING**

Hazard with medium risk of death or serious injury.

#### **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 Introduction

TheTc2\_EnOcean library is a TwinCAT PLC library for data exchange with EnOcean® devices. The library should only be used in conjunction with a KL6021-0023 or KL6581.

The user of this library requires basic knowledge of the following:

- TwinCAT XAE
- · PC and network knowledge
- · structure and properties of the Beckhoff Embedded PC and its Bus Terminal system
- · technology of EnOcean® devices
- · relevant safety regulations for technical building equipment

This software library is intended for building automation system partners of Beckhoff Automation GmbH & Co. KG. The system partners operate in the field of building automation and are concerned with the installation, commissioning, expansion, maintenance and servicing of measurement, control, and regulating systems for the technical equipment of buildings.

The Tc2\_EnOcean library is usable on all hardware platforms that support TwinCAT 3.1 or higher.

Hardware documentation for <u>KL6021-0023</u> and <u>KL6581</u> in the Beckhoff Information System.



# 3 EnOcean®

The EnOcean® radio technology makes a far-reaching signal with low quantities of ambient energy possible. With 50  $\mu$ Ws, a standard EnOcean® radio module can easily transmit a signal over a distance of 300 m (in the free field). The signal period for an EnOcean® telegram is approx. 1 thousandth of second.

- · License-free 868 MHz frequency band with 1% duty cycle
- · Multiple telegram transmission with checksum
- · Short telegrams (approx. 1 ms) lead to a small probability of collision
- · Long range: 30 m indoors or 300 m outdoors
- · Repeater available for extensions
- · Unidirectional and bidirectional communication
- · High data transmission rates of 125 kbit/s
- · Low "data overhead"
- · ASK modulation
- · Radio protocol is defined and integrated in modules
- · Sensor profiles specified and adhered to by users
- Unique transmission ID (32-bit)
- · No interference with DECT, WLAN, PMR systems etc.
- · System design verified in industrial environment

#### **Protocol structure**

Protocol	Description	Length
ORG	Telegram type	1 byte
DB_3	Data byte 3	1 byte
DB_2	Data byte 2	1 byte
DB_1	Data byte 1	1 byte
DB_0	Data byte 0	1 byte
ID_3	Transmitter ID byte 3	1 byte
ID_2	Transmitter ID byte 2	1 byte
ID_1	Transmitter ID byte 1	1 byte
ID_0	Transmitter ID byte 0	1 byte
STATUS	Information status	1 byte

# 3.1 Range planning

Please follow the recommendations of the EnOcean® Alliance (see <a href="www.enocean.org">www.enocean.org</a>) when placing the EnOcean® devices. Adherence to the recommendations supports optimum range and immunity to interference.

#### Attenuation of different materials

Material	Attenuation
Wood, plaster, uncoated glass (without metal)	010 %
Brick, chipboard	535 %
Concrete with iron reinforcement	1090 %
Metal, aluminum cladding	90100 %



#### Range

Material	Range
Line of sight	Typically 30 m in corridors, up to 100 m in halls
Plasterboard walls/wood	Typically 30 m, through max. 5 walls
Brick walls/aerated concrete	Typically 20 m, through max. 3 walls
Reinforced concrete walls/ceilings	Typically 10 m, through max. 1 wall/ceiling

#### Placement of the KL6583 module

The KL6583 module contains transmitter, receiver and antenna.

#### **Distances**

The distance to a reinforced concrete ceiling should be at least 50 cm and to a wall 10 cm.

Do not attach or screw the KL6583 module to a metal plate!

#### **Environmental conditions**

Furthermore, the environmental conditions are to be adhered to:

- Maximum air humidity 95% without condensation
- Ambient temperature 0 55°C

# 3.2 Approval of EnOcean® radio technology

Check that operation is permitted in your country

1

The EnOcean® transceivers KL6583 can be operated free of registration and fees in the following countries: **KL6583-0000: European Union** and **Switzerland** 

KL6583-0100: USA and Canada

Permission for use in other countries must be clarified explicitly!

#### KL6583-0100 for USA and Canada

Contains IC: 5731A-TCM320C
Contains FCC ID: SZV-TCM320C

This device complies with Part 15 of the FCC Rules. Operation is subject to the following conditions:

- · this device must not cause adverse interference, and
- this device must absorb all received interference, including interference that would impair the operation.



# 4 Programming

#### POUs/KL6021-0023

POUs	Description
FB_EnOceanReceive [▶ 14]	Communication with KL6021-0023

#### POUs/KL6021-0023/Read

POUs	Description
FB EnOceanPTM100 [ 15]	Receives the signals of a PTM100 module
FB EnOceanPTM200 [ 17]	Receives the signals of a PTM200 module
FB EnOceanSTM100 [ 18]	Receives the signals of a STM100 module (obsolete)
FB EnOceanSTM100Generic [ > 20]	Receives the signals of a STM100 module
FB EnOceanSTM250 [ 22]	Receives the signals of a STM250 module

#### POUs/KL6581

Function blocks	Description
FB KL6581 [ <b>&gt;</b> 23]	Communication with a KL6581

#### POUs/KL6581/Read

Function blocks	Description
FB Rec Generic [▶ 25]	Receives all types of EnOcean® telegrams
FB_Rec_1BS [▶ 26]	Receives data with ORG telegram 6.Typical EnOcean® device: window contact
FB Rec RPS Switch [ 27]	Receives data with ORG telegram 5.Typical EnOcean® device: switch
FB Rec RPS Window Handle [▶ 27]	Receives data with ORG telegram 5. Typical EnOcean® device: window handle

#### POUs/KL6581/Send

Function blocks	Description
FB Send Generic [ > 28]	Sends any EnOcean® telegrams-
FB Send 4BS [ > 29]	Sends EnOcean® telegrams in 4BS format-
FB Send RPS Switch [▶ 30]	Sends EnOcean® telegrams in the format of a switch.
FB Send RPS SwitchAuto [▶ 31]	Sends EnOcean® telegrams in the format of a switch.

#### POUs/KL6581/Other

Function blocks	Description
FB EnOcean Search [▶ 32]	This function block recognizes all EnOcean® devices within its range and displays them.
FB_Rec_Teach_In [▶ 33]	This function block indicates if the LRN bit in an EnOcean® telegram is set, independent of its EnOcean® ID.
FB Rec Teach In Ex [▶ 34]	This function block indicates when the Learn button is pressed on an EnOcean® device.



## POUs/KL6581/Function

Function blocks	Description
<u> </u>	This function converts a byte raw value into a REAL variable.
<u> </u>	This function converts a raw byte value to a Boolean array.

#### DUTs/KL6021-0023/Hardware Types

Data types	Description
ST EnOceanInData [ > 39]	Process image of the KL6021-0023 inputs
ST EnOceanOutData [ • 40]	Process image of the KL6021-0023 outputs

#### DUTs/KL6021-0023

Data types	Description
E EnOceanRotarySwitch [▶ 38]	State of the rotary-switch on the transmitting-module
E EnOceanSensorType [▶ 38]	Sensor type
ST EnOceanReceivedData [ > 39]	Internal structure

#### DUTs/KL6581/hardware types

Data types	Description
<u>KL6581_Input [▶ 41]</u>	Process image of the KL6581 inputs
KL6581 Output [▶ 42]	Process image of the KL6581 outputs

#### DUTs/KL6581

Data types	Description
AR EnOceanWindow [▶ 43]	State of the window
E ENOCEAN Org [▶ 40]	Type of EnOcean® telegram
E KL6581 Err [▶ 41]	Error messages
STR EnOceanSwitch [ 43]	State of the switches
STR KL6581 [ • 44]	Internal structure
STR Teach In [▶ 45]	Data structure manufacturer ID, type and profile
STREnOceanTurnSwitch [ > 45]	Position of the rotary switch at the room control unit

# 4.1 POUs

#### KL6021-0023

POUs	Description
FB EnOceanReceive [ 14]	Communication with KL6021-0023

#### KL6021-0023/Read

POUs	Description
FB EnOceanPTM100 [▶ 15]	Receives the signals of a PTM100 module
FB EnOceanPTM200 [▶ 17]	Receives the signals of a PTM200 module
FB_EnOceanSTM100 [▶ 18]	Receives the signals of a STM100 module (obsolete)
FB EnOceanSTM100Generic [> 20]	Receives the signals of a STM100 module



POUs	Description
FB EnOceanSTM250 [ 22]	Receives the signals of a STM250 module

#### KL6581

POUs	Description
1 B RE0301 [	Main block for the communication with the KL6581 and the connected KL6583 modules

#### KL6581/Read

Function blocks	Description
FB Rec Generic [ > 25]	Receives all types of EnOcean® telegrams.
FB Rec 1BS [▶ 26]	Receives data with ORG telegram 6.Typical EnOcean® device: window contact
FB Rec RPS Switch [▶ 27]	Receives data with ORG telegram 5.Typical EnOcean® device: switch
FB_Rec_RPS_Window_Handle [▶ 27]	Receives data with ORG telegram 5. Typical EnOcean® device: window handle

#### KL6581/Send

Function blocks	Description
FB Send Generic [ > 28]	Sends any EnOcean® telegrams.
FB Send 4BS [ > 29]	Sends EnOcean® telegrams in 4BS format.
FB_Send_RPS_Switch [▶ 30]	Sends EnOcean® telegrams in the format of a switch.
FB Send RPS SwitchAuto [▶ 31]	Sends EnOcean® telegrams in the format of a switch.

#### KL6581/Other

Function blocks	Description
FB EnOcean Search [▶ 32]	This function block recognizes all EnOcean® devices within its range and displays them.
FB Rec Teach In [▶ 33]	This function block indicates if the LRN bit in an EnOcean® telegram is set, independent of its EnOcean® ID.
FB Rec Teach In Ex [▶ 34]	This function block indicates when the Learn button is pressed on an EnOcean® device.

#### KL6581/Function

Function blocks	Description
<u> </u>	This function converts a byte raw value into a REAL variable.
F Byte To TurnSwitch [▶ 36]	This function converts a raw byte value to a Boolean array.

## 4.1.1 KL6021-0023

Function blocks	Description
FB EnOceanReceive [ 14]	Communication with a KL6021-0023



#### Read

Function blocks	Description
FB_EnOceanPTM100 [ 15]	Receives the signals of a PTM100 module
FB EnOceanPTM200 [ 17]	Receives the signals of a PTM200 module
FB EnOceanSTM100 [ 18]	Receives the signals of a STM100 module (outdated)
FB EnOceanSTM100Generic [ 20]	Receives the signals of a STM100 module
FB EnOceanSTM250 [ 22]	Receives the signals of a STM250 module

### 4.1.1.1 FB\_EnOceanReceive

```
FB_EnOceanReceive

— bEnable bError —

stEnOceanInData nErrorId —

stEnOceanOutData stEnOceanReceivedData —
```

The function block FB\_EnOceanReceive() is a receive block, which makes the telegrams sent by the EnOcean® modules available in the structure stEnOceanReceivedData. This structure can then be analyzed with further function blocks. The documentation for this function blocks also includes sample programs, which illustrate the operating principle.

#### Inputs

```
VAR_INPUT
bEnable : BOOL := FALSE;
END VAR
```

Name	Туре	Description
bEnable		A positive signal at this input activates the function block. A negative signal at the input disables the function block functionality.

#### **₹/** Inputs/outputs

```
VAR_IN_OUT
stEnOceanInData : ST_EnOceanInData;
stEnOceanOutData : ST_EnOceanOutData;
END_VAR
```

Name	Туре	Description
stEnOceanInData	<u> </u>	Is linked to the input addresses of the KL6021-0023 in the System Manager.
stEnOceanOutData	ST_EnOceanOutData [▶ 40]	Is linked to the output addresses of the KL6021-0023 in the System Manager.

#### Outputs

```
VAR_OUTPUT
bError : BOOL := FALSE;
nErrorId : UDINT := 0;
stEnOceanReceivedData : ST_EnOceanReceivedData;
END_VAR
```

Name	Туре	Description
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the variable <i>nErrorID</i> .
nErrorld	UDINT	Describes the type of <u>error</u> [▶ <u>23</u> ].
stEnOceanReceived Data	ST_EnOceanReceivedData [▶_39]	The received data is stored in this structure.



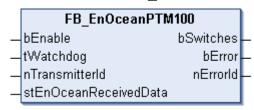
#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.1.2 Read

Function blocks	Description
FB EnOceanPTM100 [▶ 15]	Receives the signals of a PTM100 module
FB EnOceanPTM200 [ 17]	Receives the signals of a PTM200 module
FB EnOceanSTM100 [▶ 18]	Receives the signals of a STM100 module (outdated)
FB EnOceanSTM100Generic [ > 20]	Receives the signals of a STM100 module
FB EnOceanSTM250 [ 22]	Receives the signals of a STM250 module

#### 4.1.1.2.1 FB\_EnOceanPTM100



The function block *FB\_EnOceanPTM100()* provides a user-friendly evaluation of the state of an EnOcean® PTM100 module. The function block <u>FB\_EnOceanReceive()</u> [▶ 14] is required for this purpose.

In contrast to the PTM200 and PTM250 modules, only one switch at a time can be pressed in the PTM100 module. In addition, the PTM100 module supports eight switches, not four.



A new instance of this function block must be created for each switch module used.

#### Inputs

VAR\_INPUT

bEnable : BOOL := FALSE;

tWatchdog : TIME;

nTransmitterId : UDINT;

stEnOceanReceivedData : ST\_EnOceanReceivedData;

END\_VAR

Name	Туре	Description
bEnable	BOOL	A positive signal at this input enables the function block. A negative signal at the input disables the function block functionality, and all outputs are set to 0 or FALSE.
tWatchdog	TIME	Monitoring time. Within this time, new information must reach this function block via the input stEnOceanReceivedData described below. If this time is set to t#0s, the watchdog function is inactive.
nTransmitterId	UDINT	ID of the EnOcean® module, to which the function block should respond.
stEnOceanReceived Data	ST_EnOceanReceivedData [▶ 39]	Information and required connection to the EnOcean® receive function block <u>FB EnOceanReceive()</u> [▶ 14]. This information is stored in a structure.



#### Outputs

```
VAR_OUTPUT
bSwitches : ARRAY [0..7] OF BOOL;
bError : BOOL := FALSE;
nErrorId : UDINT := 0;
END VAR
```

Name	Туре	Description
bSwitches	ARRAY OF BOOL	This field of 8 Boolean values describes the states of the 8 push buttons on the push button module.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the variable <i>nErrorID</i> .
nErrorld	UDINT	Describes the type of <u>error</u> [▶ <u>23</u> ].

The following sample program illustrates the operating principle of this function block:

```
PROGRAM MAIN
VAR
    fbEnOceanReceive : FB_EnOceanReceive;
    fbEnOceanPTM100 1 : FB EnOceanPTM100;
    fbEnOceanPTM100 2 : FB EnOceanPTM100;
    bSwitches1 : ARRAY
                               [0..7] OF
                                                 BOOL;
    bSwitches2 1 : BOOL;
    bSwitches2_2 : BOOL;
    bSwitches2 3 : BOOL;
    bSwitches2 4 : BOOL;
    bSwitches2_5 : BOOL;
bSwitches2_6 : BOOL;
    bSwitches2_7 : BOOL;
    bSwitches2 8 : BOOL;
END VAR
fbEnOceanReceive(
    bEnable := TRUE,
    stEnOceanInData := stEnOceanInData,
    stEnOceanOutData := stEnOceanOutData);
fbEnOceanPTM100 1(
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C4,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
bSwitches1 := fbEnOceanPTM100 1.bSwitches;
fbEnOceanPTM100 2(
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C5,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
bSwitches2_1 := fbEnOceanPTM100_2.bSwitches[0];
bSwitches2_3 := fbEnOceanPTM100_2.bSwitches[1];
bSwitches2_6 := fbEnOceanPTM100_2.bSwitches[2];
bSwitches2 5 := fbEnOceanPTM100 2.bSwitches[3];
bSwitches2 8 := fbEnOceanPTM100 2.bSwitches[4];
bSwitches2_2 := fbEnOceanPTM100_2.bSwitches[5]; bSwitches2_7 := fbEnOceanPTM100_2.bSwitches[6];
bSwitches2 4 := fbEnOceanPTM100 2.bSwitches[7];
```

In this sample program two transmitter modules (PTM100) are queried: a transmitter module with the transmitter ID 16#C4 and another module with the transmitter ID 16#C5. A function block FB\_EnOceanPTM100 is created for each of the two transmitter modules. Both function blocks obtain their information from the upstream receive function block FB\_EnOceanReceive and are only active (input bEnable) if the receive function block is active and not in error. The push buttons of the first transmitter module are assigned to a Boolean array bSwitches1 for further analysis, while the push buttons of the second transmitter module are assigned to individual Boolean variables bSwitches2\_1 to bSwitches2\_8; both options are conceivable.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0



#### 4.1.1.2.2 FB EnOceanPTM200

	FB_EnOceanPTM200		l
_	bEnable	bSwitches	H
_	tWatchdog	bError	H
_	nTransmitterId	nErrorld	H
_	stEnOceanReceivedData		l

The function block *FB\_EnOceanPTM200()* provides a user-friendly evaluation of the state of an EnOcean® PTM200 or PTM250 module. The function block <u>FB\_EnOceanReceive()</u> [▶\_14] is required for this purpose.

In contrast to the PTM100 module, the PTM200 and PTM250 modules allow up to two switches to be pressed simultaneously. In addition, the PTM200 and PTM250 modules support four instead of eight switches.



A new instance of this function block must be created for each switch module used.

#### Inputs

```
VAR_INPUT
bEnable : BOOL := FALSE;
tWatchdog : TIME;
nTransmitterId : UDINT;
stEnOceanReceivedData : ST_EnOceanReceivedData;
END VAR
```

Name	Туре	Description
bEnable	BOOL	A positive signal at this input enables the function block. A negative signal at the input disables the function block functionality, and all outputs are set to 0 or FALSE.
tWatchdog	TIME	Monitoring time. Within this time, new information must reach this function block via the input stEnOceanReceivedData described below. If this time is set to t#0s, the watchdog function is inactive.
nTransmitterId	UDINT	ID of the EnOcean® module, to which the function block should respond.
stEnOceanReceived Data	ST EnOceanReceivedData [▶ 39]	Information and required connection to the EnOcean® receive function block <u>FB EnOceanReceive()</u> [▶ 14]. This information is stored in a structure.

#### Outputs

```
VAR_OUTPUT
  bSwitches : ARRAY [0..3] OF BOOL;
  bError : BOOL := FALSE;
  nErrorId : UDINT := 0;
END_VAR
```

Name	Туре	Description
bSwitches	ARRAY OF BOOL	This field of 4 Boolean values describes the states of the 4 push buttons on the push button module.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the variable <i>nErrorID</i> .
nErrorld	UDINT	Describes the type of error [▶ 23].

The following sample program illustrates the operating principle of this function block:

```
PROGRAM MAIN

VAR

fbEnOceanReceive : FB_EnOceanReceive;
fbEnOceanPTM100_1 : FB_EnOceanPTM200;
```



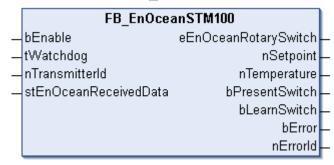
```
fbEnOceanPTM100 2 : FB EnOceanPTM200;
    bSwitches1 : \overline{ARRAY} [0..3] OF BOOL;
    bSwitches2 1 : BOOL;
    bSwitches2_2 : BOOL;
    bSwitches2_3 : BOOL;
    bSwitches2 4 : BOOL;
END VAR
fbEnOceanReceive(
    bEnable := TRUE,
    stEnOceanInData := stEnOceanInData
    stEnOceanOutData := stEnOceanOutData);
fbEnOceanPTM200 1(
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C6,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
    bSwitches1 := fbEnOceanPTM200 1.bSwitches;
fbEnOceanPTM200 2(
   bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C7,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
bSwitches2 1 := fbEnOceanPTM200 2.bSwitches[0];
bSwitches2 2 := fbEnOceanPTM200 2.bSwitches[1];
bSwitches2_3 := fbEnOceanPTM200_2.bSwitches[2];
bSwitches2 4 := fbEnOceanPTM200 2.bSwitches[3];
```

In this sample program two transmitter modules (PTM200/PTM250) are queried; one with the transmitter ID 16#C6, another one with the transmitter ID 16#C7. A function block FB\_EnOceanPTM200 is created for each of the two transmitter modules. Both function blocks obtain their information from the upstream receive function block FB\_EnOceanReceive [\rightarrow\_14] and are only active (input bEnable) if the receive function block is active and not in error. The push buttons of the first transmitter module are assigned to a Boolean array bSwitches1 for further analysis, while the push buttons of the second transmitter module are assigned to individual Boolean variables bSwitches2\_1 to bSwitches2\_4; both options are conceivable.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.1.2.3 FB\_EnOceanSTM100



#### Outdated



For new projects the function block <u>FB EnOceanSTM100Generic()</u> [▶ 20] should be used!

The function block *FB\_EnOceanSTM100()* provides a user-friendly evaluation of the data from an EnOcean® STM100 module. The function block <u>FB\_EnOceanReceive()</u> [▶\_14] is required for this purpose.



A new instance of this function block must be created for each STM100 module used.



#### Inputs

```
VAR_INPUT
bEnable : BOOL := FALSE;
tWatchdog : TIME;
nTransmitterId : UDINT;
stEnOceanReceivedData : ST_EnOceanReceivedData;
END VAR
```

Name	Туре	Description
bEnable	BOOL	A positive signal at this input enables the function block. A negative signal at the input disables the function block functionality, and all outputs are set to 0 or FALSE.
tWatchdog	TIME	Monitoring time. Within this time, new information must reach this function block via the input stEnOceanReceivedData described below. If this time is set to t#0s, the watchdog function is inactive.
nTransmitterId	UDINT	ID of the EnOcean® module, to which the function block should respond.
stEnOceanReceived Data	ST EnOceanReceivedData [ > 39]	Information and required connection to the EnOcean® receive function block <u>FB EnOceanReceive()</u> [▶ 14]. This information is stored in a structure.

#### Outputs

```
VAR_OUTPUT

eEnOceanRotarySwitch : E_EnOceanRotarySwitch;

nSetpoint : INT;

nTemperature : INT;

bPresentSwitch : BOOL;

bLearnSwitch : BOOL;

bError : BOOL := FALSE;

nErrorId : UDINT := 0;

END VAR
```

Name	Туре	Description
eEnOceanRotarySwi tch	E EnOceanRotarySwitch [▶ 38]	The output at this output describes the position of the rotary switch on the room control unit.
nSetpoint	INT	This output variable indicates the setpoint value set on the device. The values can range from -100 to +100.
nTemperature	INT	The measured temperature is output here in 1/10 °C with a measuring range of 0 °C to 40 °C. If the watchdog is triggered, the function block assumes a wire break-like fault and the value is fixed at 850 °C.
bPresentSwitch	BOOL	If the presence button at the room control unit is activated, this output becomes <i>TRUE</i> .
bLearnSwitch	BOOL	If the learn button at the room control unit is activated, this output becomes <i>TRUE</i> .
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the variable <i>nErrorID</i> .
nErrorld	UDINT	Describes the type of error [▶ 23].

The following sample program illustrates the operating principle of this function block:

```
PROGRAM MAIN

VAR

fbEnOceanReceive : FB_EnOceanReceive;
fbEnOceanSTM100_1 : FB_EnOceanSTM100;
fbEnOceanSTM100_2 : FB_EnOceanSTM100;
nTemperature : ARRAY [1..2] OF INT;
nSetpoint : ARRAY [1..2] OF INT;
nStateRotarySwitch : ARRAY [1..2] OF E_EnOceanRotarySwitch;
bPresentSwitch : ARRAY [1..2] OF BOOL;

END_VAR
```



```
fbEnOceanReceive(
   bEnable := TRUE,
   stEnOceanInData := stEnOceanInData,
   stEnOceanOutData := stEnOceanOutData);
fbEnOceanSTM100 1(
   bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceve.bEnable,
   nTransmitterId := 16#000000C4,
   tWatchdog:=t#1h,
   stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData
   nTemperature => Temperature[1],
   nSetpoint => nSetpoint[1]
   eEnOceanRotarySwitch => nStateRotarySwitch[1],
   bPresentSwitch => bPresentSwitch[1]);
fbEnOceanSTM100 2(
   bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
   nTransmitterId := 16\#000000C5.
   tWatchdog:=t#0s,
   stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData
   nTemperature => Temperature[2],
   nSetpoint => nSetpoint[2]
   eEnOceanRotarySwitch => nStateRotarySwitch[2],
   bPresentSwitch => bPresentSwitch[2]);
```

In this sample program two room control units are queried; one with the transmitter ID 16#00000C4 and another one with the transmitter ID 16#00000C5. A function block FB\_EnOceanSTM100 is created for each of the two function blocks. Both function blocks obtain their information from the upstream receive function block FB EnOceanReceive [ \( \) 14] and are only active (input bEnable) if the receive function block is active and not in error. The first device is monitored by the watchdog function, whereby new values must be transmitted to the control unit within 1 hour; the second device is programmed without watchdog monitoring. The values output to the function blocks are assigned to flags for further evaluation.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.1.2.4 FB EnOceanSTM100Generic

```
FB_EnOceanSTM100Generic

— bEnable nDataBytes —
tWatchdog bError —
nTransmitterId nErrorId —
stEnOceanReceivedData
```

The function block *FB\_EnOceanSTM100Generic()* provides a user-friendly evaluation of the data of an EnOcean® STM100 module. The function block FB EnOceanReceive() [▶ 14] is required for this purpose.



A new instance of this function block must be created for each STM100 module used.

#### Inputs

```
VAR_INPUT

bEnable : BOOL := FALSE;

tWatchdog : TIME;

nTransmitterId : UDINT;

stEnOceanReceivedData : ST_EnOceanReceivedData;

END_VAR
```

Name	Туре	Description
bEnable	BOOL	A positive signal at this input enables the function block. A negative signal at the input disables the function block functionality, and all outputs are set to 0 or FALSE.



Name	Туре	Description
tWatchdog	TIME	Monitoring time. Within this time, new information must reach this function block via the input stEnOceanReceivedData described below. If this time is set to t#0s, the watchdog function is inactive.
nTransmitterId	UDINT	ID of the EnOcean® module, to which the function block should respond.
stEnOceanReceived Data	ST EnOceanReceivedData [ > 39]	Information and required connection to the EnOcean® receive function block <u>FB EnOceanReceive()</u> [▶ 14]. This information is stored in a structure.

#### Outputs

```
VAR_OUTPUT
  nDataBytes : ARRAY [0..3] OF BYTE;
  bError : BOOL := FALSE;
  nErrorId : UDINT := 0;
END_VAR
```

Name	Туре	Description
nDataBytes	ARRAY OF BYTE	4-byte array with the user data sent by the STM100 module. The purpose of the individual bytes is manufacturer-specific.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the variable <i>nErrorID</i> .
nErrorld	UDINT	Describes the type of error [▶ 23].

The following sample program illustrates the operating principle of this function block:

```
PROGRAM MAIN
VAR
    fbEnOceanReceive : FB EnOceanReceive;
    fbEnOceanSTM100 1 : FB EnOceanSTM100Generic;
    fbEnOceanSTM100 2 : FB EnOceanSTM100Generic;
    nTemperature : ARRAY [1..2] OF BYTE;
    nSetpoint : ARRAY [1..2] OF BYTE;
    nStateRotarySwitch : ARRAY [1..2] OF BYTE;
    nPresentSwitch : ARRAY [1..2] OF BYTE;
END VAR
fbEnOceanReceive(
    bEnable := TRUE,
    stEnOceanInData := stEnOceanInData,
    stEnOceanOutData := stEnOceanOutData);
fbEnOceanSTM100 1(
   bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C4,
    tWatchdog:=t#1h,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
nTemperature[1] := fbEnOceanSTM100_1.nDataBytes[0];
nSetpoint[1] := fbEnOceanSTM100 1.nDataBytes[1];
nStateRotarySwitch[1] := fbEnOceanSTM100 1.nDataBytes[2];
nPresentSwitch[1] := fbEnOceanSTM100_1.nDataBytes[3];
fbEnOceanSTM100 2(
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C5,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
nTemperature[2] := fbEnOceanSTM100 2.nDataBytes[0];
nSetpoint[2] := fbEnOceanSTM100 2.nDataBytes[1];
nStateRotarySwitch[2] := fbEnOceanSTM100_2.nDataBytes[2];
nPresentSwitch[2] := fbEnOceanSTM100_2.nDataBytes[3];
```

In this sample program, two EnOcean® transmitter modules are queried, one with the transmitter ID 16#00000C4 and another with the transmitter ID 16#00000C5. A function block FB\_EnOceanSTM100Generic is created for both transmitters. Both function blocks obtain their information from the upstream receive function block FB EnOceanReceive [▶ 14] and are only active (input bEnable) if the receive function block is active and not in error. The first device is monitored by the watchdog function, whereby new values must be transmitted to the control unit within 1 hour; the second device is programmed



without watchdog monitoring. The values output to the function blocks are assigned to variables for further evaluation. Before the values can be used further, they have to be scaled to physical values. Details of the conversion can be found in the data sheet for the sensor.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.1.2.5 FB\_EnOceanSTM250

	FB_EnOceanSTM2	50
_	bEnable	bState
_	tWatchdog	bLearn
_	nTransmitterId	bError
_	stEnOceanReceivedData	nErrorld

The function block *FB\_EnOceanSTM250()* provides a user-friendly evaluation of the data from an EnOcean® STM250 module. The function block <u>FB\_EnOceanReceive()</u> [▶\_14] is required for this purpose.



A new instance of this function block must be created for each STM250 module used.

#### Inputs

```
VAR_INPUT
bEnable : BOOL := FALSE;
tWatchdog : TIME;
nTransmitterId : UDINT;
stEnOceanReceivedData : ST_EnOceanReceivedData;
END_VAR
```

Name	Туре	Description
bEnable	BOOL	A positive signal at this input enables the function block. A negative signal at the input disables the function block functionality, and all outputs are set to 0 or FALSE.
tWatchdog	TIME	Monitoring time. Within this time, new information must reach this function block via the input stEnOceanReceivedData described below. If this time is set to t#0s, the watchdog function is inactive.
nTransmitterId	UDINT	ID of the EnOcean® module, to which the function block should respond.
stEnOceanReceived Data	ST EnOceanReceivedData [• 39]	Information and required connection to the EnOcean® receive function block <u>FB EnOceanReceive()</u> [▶ 14]. This information is stored in a structure.

#### Outputs

```
VAR_OUTPUT
bState : BOOL;
bLearn : BOOL;
bError : BOOL := FALSE;
nErrorId : UDINT := 0;
END_VAR
```

Name	Туре	Description
bState		When the reed contact on the STM250 module is activated, this output becomes <i>TRUE</i> (contact closed).
bLearn	BOOL	This output becomes <i>FALSE</i> if the learn button at the STM250 module is activated.



Name	Туре	Description
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the variable <i>nErrorID</i> .
nErrorld	UDINT	Describes the type of error [▶ 23].

The following sample program illustrates the operating principle of this function block:

```
PROGRAM MAIN
VAR
    fbEnOceanReceive : FB EnOceanReceive;
    fbEnOceanSTM250 : FB EnOceanSTM250;
    bState : BOOL;
   bLearn : BOOL;
END_VAR
fbEnOceanReceive(
    bEnable := TRUE,
    stEnOceanInData := stEnOceanInData,
    stEnOceanOutData := stEnOceanOutData);
fbEnOceanSTM250(
   bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000008CA,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData
    bState => bState.
  bLearn => bLearn);
```

In this sample program an STM250 module with the transmitter ID 16#000008CA is queried. To this end the function block *FB\_EnOceanSTM250* is created. This function blocks obtains its information from the upstream receive function block <u>FB\_EnOceanReceive [\rightarrow 14]</u> and is only active (input *bEnable*) if the receive function block is active and not in error. The values output to the function block are assigned to variables for further evaluation.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.1.3 Error codes

Value (hex)	Description
0x0000	No error.
0x0001	Checksum error.
0x0002	Watchdog monitoring.
0x0003	Buffer overflow (in the KL6023)
0x0004	No data received yet from sensor receive.

#### 4.1.2 KL6581

#### 4.1.2.1 FB\_KL6581

```
FB_KL6581

— blnit bReady —
nldx bBusy —
stKL6581_in bErrorlD —
stKL6581_out iErrorlD —
```

This function block takes care of communication with the KL6581 EnOcean® Bus Terminal. This function block is used to configure the KL6581 and start data exchange with the EnOcean® network.





#### Restrictions

- Only one call per instance
- · Call must be made once per PLC cycle
- Instance must be called in the same PLC task as the send and receive blocks assigned to it
- Maximum 64 instances per PLC project allowed

#### Inputs

```
VAR_INPUT
bInit : BOOL;
nIdx : USINT := 1;
END VAR
```

Name	Туре	Description
blnit		Activates the function block that configures the KL6301 and then activates the data exchange.
nldx		The idx number must be unique for each KL6581 (valid values: 164) if more than one bus terminal is used per PLC program.

### **₹/** Inputs/outputs

```
VAR_IN_OUT
   stKL6581_in : KL6581_Input;
   stKL6581_out : KL6581_Output;
END VAR
```

Name	Туре	Description
stKL6581_in	itteoso i inipat [/ _ i i]	Is linked to the input addresses of the KL6581 in the System Manager.
stKL6581_out	112000: 0 010010 1 1 1 1 1 1	Is linked to the output addresses of the KL6581 in the System Manager.

#### Outputs

```
VAR_OUTPUT

bReady : BOOL;

bBusy : BOOL;

bError : BOOL;

iErrorID : E_KL6581_Err;

str_KL6581 : STR_KL6581;

END VAR
```

Name	Туре	Description
bReady	BOOL	The function block is ready for sending and receiving data.
bBusy	BOOL	The function block is active. No new data can be sent yet.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the <i>iErrorID</i> variable.
iErrorID	E_KL6581_Err [▶ 41]	Describes the type of error.
str_KL6581	STR_KL6581 [▶ 44]	Connected with the send and receive function blocks.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0



#### 4.1.2.2 Read

Function blocks	Description
FB Rec Generic [ > 25]	Receives all types of EnOcean® telegrams.
FB Rec 1BS [▶ 26]	Receives data with ORG telegram 6.Typical EnOcean® device: window contact
FB Rec RPS Switch [ 27]	Receives data with ORG telegram 5.Typical EnOcean® device: switch
FB Rec RPS Window Handle [▶ 27]	Receives data with ORG telegram 5. Typical EnOcean® device: window handle

#### 4.1.2.2.1 FB\_Rec\_Generic

```
FB_Rec_Generic

— str_KL6581 ar_Value —
byNode by_Node —
dw_ID by_STATE —
bReceive —
EnOceanTyp —
```

This function block receives all data that were received via EnOcean®. This function block can be used for all types of EnOcean® telegrams.

The user must interpret the data himself. This requires the documentation of the manufacturer of the transmitting EnOcean® device.

#### Inputs

```
VAR_INPUT
str_KL6581: STR_KL6581;
byNode : BYTE;
dw_ID : DWORD;
END_VAR
```

Name	Туре	Description
str_KL6581	STR KL6581 [▶ 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].
byNode		Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.
dw_ID	DWORD	EnOcean® ID to be received.

#### Outputs

```
VAR_OUTPUT

ar_Value : ARRAY [0..3] OF BYTE;

by_Node : BYTE;

by_STATE : BYTE;

bReceive : BOOL := TRUE;

EnOceanTyp : E_EnOcean_Org;

END_VAR
```

Name	Туре	Description
ar_Value	ARRAY OF BYTE	EnOcean® data 4 bytes.
by_Node		Node number of the KL6583 that has received the EnOcean® telegram.
by_STATE	BYTE	EnOcean® STATE Field.
bReceive		On receiving an EnOcean® telegram this value is set to FALSE for one cycle.
EnOceanTyp	E EnOcean Org [▶ 40]	EnOcean® ORG Field.



#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.1.2.2.2 FB\_Rec\_1BS

```
FB_Rec_1BS

-- str_KL6581 bOpen --
-- byNode bClose --
-- dw_ID bLRN --
by_Node --
bReceive --
```

This function block receives data that were received via EnOcean®. This function block is used for integration of window contacts, for example (ORG FIELD 6).

#### Inputs

```
VAR_INPUT
str_KL6581 : STR_KL6581;
byNode : BYTE;
dw_ID : DWORD;
END VAR
```

Name	Туре	Description
str_KL6581	311 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Is connected to the data structure of the function block FB KL6581() [▶ 23].
byNode		Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.
dw_ID	DWORD	EnOcean® ID to be received.

#### Outputs

```
VAR_OUTPUT
bOpen : BOOL;
bClose : BOOL;
bLRN : BOOL;
by_Node : BYTE;
bReceive : BOOL := TRUE;
END VAR
```

Name	Туре	Description
bOpen	BOOL	Contact open
bClose	BOOL	Contact closed
bLRN	BOOL	LRN button pressed
by_Node	BYTE	Node number of the KL6583 that has received the EnOcean® telegram.
bReceive	BOOL	On receiving an EnOcean® telegram this value is set to FALSE for one cycle.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0



#### 4.1.2.2.3 FB\_Rec\_RPS\_Switch

This function block receives data from a switch that were received via EnOcean®. The function block outputs the data in a data structure (ORG Field 5).

#### Inputs

VAR\_INPUT
 str\_KL6581 : STR\_KL6581;
 byNode : BYTE;
 dw\_ID : DWORD;

END\_VAR

Name	Туре	Description
str_KL6581	STR KL6581 [▶ 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].
byNode		Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.
dw_ID	DWORD	EnOcean® ID to be received.

#### Outputs

VAR\_OUTPUT str\_EnOceanSwitch : STR\_EnOceanSwitch

by\_Node : BYTE;

bReceive : BOOL := TRUE;

END\_VAR

Name	Туре	Description
str_EnOceanSwitch	STR_EnOceanSwitch	Switch data
	[ <u>\_43]</u>	
by_Node		Node number of the KL6583 that has received the EnOcean® telegram.
bReceive		On receiving an EnOcean® telegram this value is set to FALSE for one cycle.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.2.2.4 FB\_Rec\_RPS\_Window\_Handle

```
FB_Rec_RPS_Window_Handle

- str_KL6581 ar_Data -

- byNode by_Node -

- dw_ID bReceive -
```

This function block receives data from a window handle that were received via EnOcean®. The function block outputs the data in a data structure (ORG Field 5).



## Inputs

VAR INPUT

str\_KL6581 : STR\_KL6581; byNode : BYTE;
dw\_ID : DWORD;
END\_VAR

Name	Туре	Description
str_KL6581	3111 (KE0301 ) 111	Is connected to the data structure of the function block FB KL6581() [▶ 23].
byNode		Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.
dw_ID	DWORD	EnOcean® ID to be received.

#### Outputs

VAR OUTPUT

ar\_Data : AR\_EnOceanWindow;
by\_Node : BYTE;
bReceive : BOOL := TRUE;

END VAR

Name	Туре	Description
ar_Data	AR_EnOceanWindow	Window handle data
	[ <u>*43]</u>	
by_Node	BYTE	Node number of the KL6583 that has received the EnOcean® telegram.
bReceive	BOOL	On receiving an EnOcean® telegram this value is set to FALSE for one cycle.

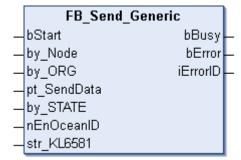
#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.2.3 Send

Function blocks	Description
FB Send Generic [ > 28]	Sends any EnOcean® telegrams.
FB Send 4BS [ > 29]	Sends EnOcean® telegrams in 4BS format.
FB Send RPS Switch [▶ 30]	Sends EnOcean® telegrams in the format of a switch.
FB Send RPS SwitchAuto [ > 31]	Sends EnOcean® telegrams in the format of a switch.

#### FB\_Send\_Generic 4.1.2.3.1





This function block sends data via EnOcean®. The type and the data contents are arbitrary. All types of EnOcean® data telegrams can be sent with this function block.

#### Inputs

```
VAR_INPUT
bStart : BOOL;
by_Node : BYTE;
by_ORG : E_EnOcean_Org;
pt_SendData : DWORD;
by_STATE : BYTE;
nEnOceanID : BYTE;
str_KL6581 : STR_KL6581;
END_VAR
```

Name	Туре	Description
bStart	BOOL	Positive edge sends the data.
by_Node	BYTE	Address of the KL6583 module to which the telegram is to be sent (valid values: 18).
by_ORG	E_EnOcean_Org [▶ 40]	ORG Field of the EnOcean® telegram
pt_SendData	DWORD	Pointer to the data to be sent. ADR is used to determine the pointer address. The pointer must point to 4-byte variable.
by_STATE	BYTE	EnOcean®-STATE. Can be changed by the TCM module.
nEnOceanID	BYTE	Virtual EnOcean® ID. A value of 0127 is added to the real EnOcean® ID (valid values: 0127).
str_KL6581	STR KL6581 [▶ 44]	Is connected to the data structure of the function block FB_KL6581() [▶ 23].

#### Outputs

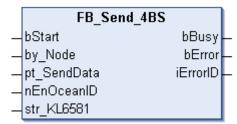
```
VAR_OUTPUT
bBusy : BOOL;
bError : BOOL;
iErrorID : E_KL6581_Err;
END_VAR
```

Name	Туре	Description
bBusy	BOOL	The function block is active. No new data can be sent yet.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the <i>iErrorID</i> variable.
iErrorID	E KL6581 Err [▶ 41]	Describes the type of error.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.1.2.3.2 FB\_Send\_4BS



This function block sends data via EnOcean®. The ORG field is set permanently to 7.



#### Inputs

```
VAR_INPUT
bStart : BOOL;
by_Node : BYTE;
pt_SendData : DWORD;
nEnOceanID : BYTE;
str_KL6581 : STR_KL6581;
END_VAR
```

Name	Туре	Description
bStart	BOOL	Positive edge sends the data.
by_Node	BYTE	Address of the KL6583 module to which the telegram is to be sent (valid values: 18).
pt_SendData	DWORD	Pointer to the data to be sent. ADR is used to determine the pointer address. The pointer must point to 4-byte variable.
nEnOceanID	BYTE	Virtual EnOcean® ID. A value of 0127 is added to the real EnOcean® ID (valid values: 0127).
str_KL6581	STR KL6581 [▶ 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].

#### Outputs

VAR\_OUTPUT
bBusy : BOOL;
bError : BOOL;
iErrorID : E\_KL6581\_Err;
END\_VAR

Name	Туре	Description
bBusy	BOOL	The function block is active. No new data can be sent yet.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the <i>iErrorID</i> variable.
iErrorID	E KL6581 Err [▶ 41]	Describes the type of error.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

### 4.1.2.3.3 FB\_Send\_RPS\_Switch

```
FB_Send_RPS_Switch

-- bStart bBusy --
by_Node bError --
-- bData iErrorID --
-- nRockerID
-- nEnOceanID
-- str_KL6581
```

This function block sends EnOcean telegrams in the format of a switch. The value of bData is sent with a positive edge of bStart. In order to simulate a button press, the function block usually has to be started twice, once with bData = TRUE, once with bData = FALSE. For simpler handling the function block FB Send RPS SwitchAuto() [ 31] can be used.

#### Inputs

VAR\_INPUT
bStart : BOOL;
by\_Node : BYTE;



```
bData : BOOL;
nRockerID : INT;
nEnOceanID : BYTE;
str_KL6581 : STR_KL6581;
END VAR
```

Name	Туре	Description
bStart	BOOL	Positive edge sends the data.
by_Node	BYTE	Address of the KL6583 module to which the telegram is to be sent (valid values: 18).
bData	BOOL	Value to be transmitted.
nRockerID	INT	Push button number, valid values 03.
nEnOceanID	BYTE	Virtual EnOcean® ID. A value of 0127 is added to the real EnOcean® ID (valid values: 0127).
str_KL6581	STR_KL6581 [▶ 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].

#### Outputs

VAR\_OUTPUT
bBusy : BOOL;
bError : BOOL;
iErrorID : E\_KL6581\_Err;
END VAR

 Name
 Type
 Description

 bBusy
 BOOL
 The function block is active. No new data can be sent yet.

 bError
 BOOL
 This output goes TRUE as soon as an error occurs. This error is described via the iErrorID variable.

#### **Prerequisites**

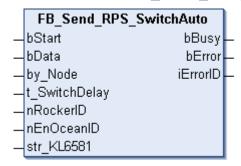
iErrorID

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

Describes the type of error.

#### 4.1.2.3.4 FB\_Send\_RPS\_SwitchAuto

E KL6581 Err [▶ 41]



This function block sends EnOcean telegrams in the format of a switch. The value of *bData* is sent with a positive edge of *bStart*. The signal "Release switch" is sent once the time *t\_SwitchDelay* has elapsed.

#### Inputs

VAR\_INPUT

bStart : BOOL;

bData : BOOL;

by\_Node : BYTE;

t\_SwitchDelay : TIME := T#100ms;

nRockerID : INT;

nEnOceanID : BYTE;

str\_KL6581 : STR\_KL6581;

END VAR



Name	Туре	Description
bStart	BOOL	Positive edge sends the data.
bData	BOOL	Value to be transmitted.
by_Node	BYTE	Address of the KL6583 module to which the telegram is to be sent (valid values: 18).
t_SwitchDelay	TIME	How long the push button has to be pressed.
nRockerID	INT	Push button number, valid values 03.
nEnOceanID	BYTE	Virtual EnOcean® ID. A value of 0127 is added to the real EnOcean® ID (valid values: 0127).
str_KL6581	STR KL6581 [▶ 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].

#### Outputs

VAR OUTPUT

bBusy : BOOL; bError : BOOL; iErrorID : E\_KL6581\_Err;

END\_VAR

Name	Туре	Description
bBusy	BOOL	The function block is active. No new data can be sent yet.
bError	BOOL	This output goes TRUE as soon as an error occurs. This error is described via the <i>iErrorID</i> variable.
iErrorID	E_KL6581_Err [▶ 41]	Describes the type of error.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.2.4 Other

Function blocks	Description
FB EnOcean Search [▶ 32]	Function block recognizes all EnOcean® devices within its range and displays them.
FB Rec Teach In [▶ 33]	This function block indicates if the LRN bit in an EnOcean® telegram is set, independent of its EnOcean® ID.
FB_Rec_Teach_In_Ex [▶ 34]	This function block indicates when the Learn button is pressed on an EnOcean® device.

#### 4.1.2.4.1 FB\_EnOcean\_Search



This function block displays all EnOcean® IDs that it has received and enters them in a receive array (ar\_ID). Up to 256 EnOcean® devices can be recognized. Alternatively the function block can also be created separately for each KL6583. This can be used to recognize whether an EnOcean® device is being received by several KL6583s.



#### Inputs

VAR INPUT bStart : BOOL; str\_KL6581 : STR\_KL6581; byNode : BYTE;

END VAR

Name	Туре	Description
bStart	BOOL	If TRUE the function block is enabled, if FALSE it is disabled.
str_KL6581	STR KL6581 [▶ 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].
byNode	ВҮТЕ	Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.

#### Outputs

VAR OUTPUT bReceive : BOOL := TRUE;

iDevices : INT;

: ARRAY [0..255] OF DWORD; ar\_ID END\_VAR

Name	Туре	Description
bReceive	BOOL	On receiving an EnOcean® telegram this value is set to FALSE for one cycle.
iDevices	INT	Number of EnOcean® devices found.
ar_ID	ARRAY OF DWORD	EnOcean® IDs that were found.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.2.4.2 FB\_Rec\_Teach\_In

```
FB Rec Teach In
 bStart
                     bLearnType
 byNode
                       by_Node
_str_KL6581
                          dw_ID
                    str_Teach_In
                       bReceive
```

This function block indicates when a Learn button is pressed on an EnOcean® device. If the flag bLearnType is set, further information about the EnOcean® device can be read. This function must be provided by the EnOcean® device. So far, however, it is only supported by very few EnOcean® devices.

#### Inputs

END VAR

VAR INPUT bStart : BOOL; byNode : BYTE; str\_KL6581 : STR\_KL6581; bStart

Name	Туре	Description
bStart	BOOL	If TRUE the function block is enabled, if FALSE it is
		disabled.



Name	Туре	Description
byNode		Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.
str_KL6581	311 K K L O 3 O 1 [ / 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].

#### Outputs

```
VAR_OUTPUT

bLearnType : BOOL;

by_Node : BYTE;

dw_ID : DWORD;

str_Teach_In : STR_Teach_In;

bReceive : BOOL := TRUE;

END_VAR
```

Name	Туре	Description
bLearnType	BOOL	If the bit is set you will find further data in the structure str_Teach_In.
by_Node	BYTE	Number of EnOcean® devices found.
dw_ID	DWORD	EnOcean® ID for which the Learn button was pressed.
str_Teach_In	STR Teach In [▶ 45]	Data structure, profile, type and manufacturer ID.
bReceive	BOOL	On receiving an EnOcean® telegram this value is set to FALSE for one cycle.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.2.4.3 FB\_Rec\_Teach\_In\_Ex

```
FB_Rec_Teach_In_Ex

-bStart bLearnType -

-byNode by_Node-

-str_KL6581 dw_ID -

str_Teach_In -

bReceive -
```

This function block indicates when a Learn button is pressed on an EnOcean® device. If the flag *bLearnType* is set, further information about the EnOcean® device can be read. This function must be provided by the EnOcean® device. So far, however, it is only supported by very few EnOcean® devices.

In addition to the <u>FB Rec Teach In()</u> [▶ <u>33]</u> function block, the system also checks whether it is an EEP telegram.

## Inputs

```
VAR_INPUT

bStart : BOOL;

byNode : BYTE;

str_KL6581 : STR_KL6581;

END_VAR
```

Name	Туре	Description
bStart	BOOL	If TRUE the function block is enabled, if FALSE it is disabled.



Name	Туре	Description
byNode		Filter - if the value is zero, the EnOcean® telegrams from all KL6583s are received. If a value of 1 to 8 is entered, only the data from the corresponding KL6583 are received.
str_KL6581	311 K K L O 3 O 1 [ / 44]	Is connected to the data structure of the function block FB KL6581() [▶ 23].

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.32	Tc2_EnOcean from v3.4.6.0

#### Outputs

VAR\_OUTPUT

blearnType : BOOL;

by\_Node : BYTE;

dw\_ID : DWORD;

str\_Teach\_In : STR\_Teach;

bReceive : BOOL := TRUE;

END VAR

Name	Туре	Description
bLearnType	BOOL	If the bit is set you will find further data in the structure $str\_Teach\_In$ .
by_Node	BYTE	Number of EnOcean® devices found.
dw_ID	DWORD	EnOcean® ID for which the Learn button was pressed.
str_Teach_In	STR_Teach [▶_44]	Data structure, profile, type and manufacturer ID.
bReceive	BOOL	On receiving an EnOcean® telegram this value is set to FALSE for one cycle.

#### **4.1.2.5** Function

Function blocks	Description
F Byte To Temp [ > 35]	This function converts a byte raw value into a REAL variable.
F_Byte_To_TurnSwitch [▶ 36]	This function converts a raw byte value to a Boolean array.

#### 4.1.2.5.1 F\_Byte\_to\_Temp : REAL



This function converts a byte raw value into a REAL variable.

Temperature data transmitted with EnOcean® is one byte in size and is sent in a specific format. This data is usually scaled to a specific temperature value.

For example, a value is transmitted from a range of values from 0 to 40°C. The minimum and maximum data value and the raw value are transferred to the function. The output of the function then outputs the temperature as REAL variable.



#### Inputs

VAR\_INPUT
byData : BYTE;
minTemp : REAL := 0;
maxTemp : REAL := 40;

Name	Туре	Description
byData	BYTE	Raw data
minTemp	REAL	Minimum temperature
maxTemp	REAL	Maximum temperature

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.1.2.5.2 F\_Byte\_to\_TurnSwitch

```
F_Byte_To_TurnSwitch
—byData F_Byte_To_TurnSwitch
```

This function converts a raw byte value to a Boolean array in the form of a data structure (see <a href="STREnOceanTurnSwitch">STREnOceanTurnSwitch</a> [> 45]).

#### Inputs

VAR\_INPUT
byData: BYTE;
END\_VAR

Name	Туре	Description
byData	BYTE	Raw data

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.1.2.6 Error codes

Value (hex)	Value (dec)	Value (enum)	Description
0x0000	0	NO_ERROR	No error at the function block.
0x000A	10	KL6581_WrongTerminal	Wrong terminal connected.
0x0010	16	KL6581_WatchdogError	Timeout during initialization of function block FB KL6581() [ > 23].
0x0011	17	KL6581_NoComWithKL6581	This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Clean all, rebuild all and read again in the System Manager?
0x0012	18	KL6581_idx_number_not_OK	The input variable nldx of function block FB_KL6581() is greater than 64.



Value (hex)	Value (dec)	Value (enum)	Description
0x0013	19	KL6581_Switch_to_Stopp	The terminal has stopped exchanging
			data with the KL6583-0000 EnOcean®
			transmitter and receiver, no EnOcean®
			data has been sent or received.
0x0014	20	KL6581_not_ready	Internal message for the function blocks connected to the FB_KL6581().
0x0015	21	KL6581_No_KL6853_Found	No KL6583 is connected to the EnOcean®
			master terminal KL6581 or communication is not available!
0x0016	22	KL6581_TransmissionError	Data could not be sent, check address of KL6583 or KL6583 not ready for operation.

# 4.2 DUTs

# KL6021-0023/Hardware Types

Data types	Description
ST EnOceanInData [ > 39]	Process image of the inputs
ST EnOceanOutData [▶ 40]	Process image of the outputs

## KL6021-0023

Data types	Description
E EnOceanRotarySwitch [▶ 38]	State of the rotary-switch on the transmitting-module
E EnOceanSensorType [▶ 38]	Sensor type
ST EnOceanReceivedData [ > 39]	Internal structure

# KL6581/Hardware Types

Data types	Description
<u>KL6581_INPUT [▶ 41]</u>	Process image of the inputs
KL6581 OUTPUT [▶ 42]	Process image of the outputs

## KL6581

Data types	Description
AR EnOceanWindow [▶ 43]	State of the window
E ENOCEAN Org [▶ 40]	Type of EnOcean® telegram
E KL6581 Err [▶ 41]	Error messages
STR EnOceanSwitch [ > 43]	State of the switches
STR KL6581 [▶ 44]	Internal structure
STR Teach [ > 44]	Data structure manufacturer ID, type and function
STR Teach In [▶ 45]	Data structure manufacturer ID, type and profile
STREnOceanTurnSwitch [▶ 45]	Position of the rotary switch at the room control unit.



## 4.2.1 KL6021-0023

#### **Hardware types**

Data types	Description
ST EnOceanInData [ > 39]	Process image of the KL6021-0023 inputs
ST EnOceanOutData [▶ 40]	Process image of the KL6021-0023 outputs

Data types	Description
E EnOceanRotarySwitch [▶ 38]	Position of the rotary switch at the room control unit
E EnOceanSensorType [▶ 38]	Sensor type
ST EnOceanReceivedData [ > 39]	Internal structure

#### 4.2.1.1 Enums

# 4.2.1.1.1 E\_EnOceanSensorType

#### Sensor type.

Name	Description
eEnOceanSensorTypePTM	PTM
eEnOceanSensorTypeSTM1 Byte	STM 1 byte
eEnOceanSensorTypeSTM4 Byte	STM 4 byte
eEnOceanSensorTypeCTM	СТМ

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.2.1.1.2 E\_EnOceanRotarySwitch

*E\_EnOceanRotarySwitch* describes the position of the rotary switch at the room control unit.

```
TYPE E_EnOceanRotarySwitch :
  (
   eEnOceanRotarySwitchStep0 := 0,
   eEnOceanRotarySwitchStep1 := 1,
   eEnOceanRotarySwitchStep2 := 2,
   eEnOceanRotarySwitchStep3 := 3,
   eEnOceanRotarySwitchAuto := 4
)
END_TYPE
```

Name	Description
eEnOceanRotarySwitchStep 0	Switch in position "0"
eEnOceanRotarySwitchStep	Switch in position "1"



Name	Description
eEnOceanRotarySwitchStep 2	Switch in position "2"
eEnOceanRotarySwitchStep 3	Switch in position "3"
eEnOceanRotarySwitchAuto	Switch in position "Auto"

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.2.1.2 Structures

## 4.2.1.2.1 ST\_EnOceanReceivedData

Internal structure.

This structure is used to link the <u>FB\_EnOceanReceive()</u> [▶ 14] block with the receive blocks.

```
TYPE ST_EnOceanReceivedData:

STRUCT

bReceived : BOOL;

nLength : BYTE;

eEnOceanSensorType : E_EnOceanSensorType;

nData : ARRAY[0..3] OF BYTE;

nStatus : BYTE;

nTransmitterId : UDINT;

END_STRUCT

END_TYPE
```

Name	Туре	Description
bReceived	BOOL	Data received
nLength	BYTE	Length
eEnOceanSensorTy	E EnOceanSensorType	Sensor type
ре	[ <u>\_38]</u>	
nData	ARRAY OF BYTE	Data bytes
nStatus	BYTE	State
nTransmitterId	UDINT	Transmitter ID

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.2.1.2.2 ST\_EnOceanInData

Process image of the KL6021-0023 inputs.

Linked to the terminals in the System Manager.

```
TYPE ST_EnOceanInData:
STRUCT

nStatus: BYTE;
nData: ARRAY[0..10] OF BYTE;
END_STRUCT
END_TYPE
```

Name	Туре	Description
nStatus	BYTE	Status byte
nData	ARRAY OF BYTE	11 bytes for the input data



Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

# 4.2.1.2.3 ST\_EnOceanOutData

Process image of the KL6021-0023 outputs.

Linked to the terminals in the System Manager.

```
TYPE ST_EnOceanOutData:
STRUCT
nCtrl: BYTE;
nData: ARRAY[0..10] OF BYTE;
END_STRUCT
END_TYPE
```

Name	Туре	Description
nCtrl	BOOL	Control byte.
nData	ARRAY OF BYTE	11 bytes for the output data

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.2.2 KL6581

#### **Hardware Types**

Data types	Description
<u>KL6581 Input [▶ 41]</u>	Process image of the KL6581 inputs
KL6581 Output [▶ 42]	Process image of the KL6581 outputs
Data types	Description
AR EnOceanWindow [ > 43]	State of the window
E ENOCEAN Org [▶ 40]	Type of EnOcean® telegram
E KL6581 Err [▶ 41]	Error messages
STR EnOceanSwitch [ 43]	State of the switches
STR KL6581 [ <b>&gt;</b> 44]	Internal structure
<u>STR Teach</u> [▶ 44]	Data structure manufacturer ID, type and function
STR Teach In [ • 45]	Data structure manufacturer ID, type and profile
STREnOceanTurnSwitch [ • 45]	Position of the rotary switch at the room control unit.

#### 4.2.2.1 Enums

# 4.2.2.1.1 E\_ENOCEAN\_ORG

Type of EnOcean® telegram.

```
TYPE E_ENOCEAN_Org :

(
PTM_TELEGRAM := 5,
STM_1BYTE_TELEGRAM := 6,
STM_4BYTE_TELEGRAM := 7,
CTM_TELEGRAM := 8,
MODEM_TELEGRAM := 16#A,
```



```
MODEM_ACK_TELEGRAM := 16#B
)
END_TYPE
```

Name	Description
PTM_TELEGRAM	PTM telegram
STM_1BYTE_TELEGRAM	1-byte telegram
STM_4BYTE_TELEGRAM	4-byte telegram
CTM_TELEGRAM	CTM telegram
MODEM_TELEGRAM	Modem telegram
MODEM_ACK_TELEGRAM	Modem telegram with acknowledgement.

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

# 4.2.2.1.2 E\_KL6581\_Err

#### Error messages.

Name	Description
NO_ERROR	No error at the function block.
KL6581_WrongTerminal	Wrong terminal connected.
KL6581_WatchdogError	Timeout during initialization of function block "FB_KL6581".
KL6581_NoComWithKL6581	This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Clean all, rebuild all and read again in the System Manager?
KL6581_idx_number_not_OK	The input variable <i>nldx</i> of function block <u>FB_KL6581()</u> [▶ 23] is greater than 64.
KL6581_Switch_to_Stopp	The terminal has left the data exchange with the KL6583. No EnOcean® data was sent or received.
KL6581_not_ready:	Internal message for the function blocks connected to the FB_KL6581().
KL6581_No_KL6853_Found	No KL6583 is connected to the KL6581 or communication is not available!
KL6581_TransmissionError	Data could not be sent, check address of KL6583 or KL6583 not ready for operation.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.2.2.2 Structures

## 4.2.2.2.1 KL6581\_INPUT

Process image of the KL6581 inputs.



## Linked to the terminal in the System Manager.

```
TYPE KL6581_Input :
STRUCT
  nStatus : BYTE;
  CNODE : BYTE;
        : BYTE;
: BYTE;
: BYTE;
  ORG
  DB0
  DB1
         : BYTE;
  DB2
  DB3
          : BYTE;
         : BYTE;
  ID0
         : BYTE;
: BYTE;
  ID1
  ID2
 ID3 : BYTE;
STATUS : BYTE;
END STRUCT
END_TYPE
```

Name	Description
nStatus	Status byte
CNODE	Data byte
ORG	Data byte
DB0	Data byte
DB1	Data byte
DB2	Data byte
DB3	Data byte
ID0	Data byte
ID1	Data byte
ID2	Data byte
ID3	Data byte
STATUS	Data byte

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.2.2.2.2 KL6581\_Output

Process image of the KL6581 outputs.

Linked to the terminal in the System Manager.

```
TYPE KL6581_Output :

STRUCT

nControl : BYTE;
CNODE : BYTE;
ORG : BYTE;
DB0 : BYTE;
DB1 : BYTE;
DB2 : BYTE;
DB3 : BYTE;
ID0 : BYTE;
ID0 : BYTE;
ID1 : BYTE;
ID1 : BYTE;
ID2 : BYTE;
END STRUCT
END TYPE
```

Name	Description
nControl	Control byte
CNODE	Data byte
ORG	Data byte
DB0	Data byte



Name	Description
DB1	Data byte
DB2	Data byte
DB3	Data byte
ID0	Data byte
ID1	Data byte
ID2	Data byte
ID3	Data byte
STATUS	Data byte

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

# 4.2.2.2.3 ar\_EnOceanWindow

This structure indicates the state of the window.

```
TYPE AR_EnOceanWindow:
STRUCT
bUp : BOOL;
bOpen : BOOL;
bClose : BOOL;
END_STRUCT
END_TYPE
```

Name	Description
bUp	The window is tilted.
bOpen	The window is open.
bClose	The window is closed.

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

## 4.2.2.2.4 str\_EnOceanSwitch

State of the push buttons.

```
TYPE STR_EnOceanSwitch:

STRUCT

bT1_ON: BOOL;

bT1_OFF: BOOL;

bT2_ON: BOOL;

bT2_OFF: BOOL;

bT3_ON: BOOL;

bT3_OFF: BOOL;

bT4_ON: BOOL;

bT4_OFF: BOOL;

bT4_OFF: BOOL;
```

Name	Description
bT1_ON	Push button 1 on
bT1_OFF	Push button 1 off
bT2_ON	Push button 2 on
bT2_OFF	Push button 2 off
bT3_ON	Push button 3 on
bT3_OFF	Push button 3 off



Name	Description
bT4_ON	Push button 4 on
bT4_OFF	Push button 4 off

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

# 4.2.2.2.5 str\_KL6581

Internal structure.

This structure is used to link the function block FB\_KL6581() with the send/receive blocks.

```
TYPE STR_KL6581:

STRUCT

by_Status: BYTE;

by_Node: BYTE;

by_ORG: BYTE;

ar_DB: ARRAY[0..3] OF BYTE;

_Dummy: BYTE;

dw_ID: DWORD;

ptData: PVOID;

iErrorId: E_KL6581_Err;

by_STATE: BYTE;

berror: BOOL;

idx: USINT;

END_STRUCT

END_TYPE
```

Name	Description
by_Status	State
by_Node	Node number of the KL6583 that has received the EnOcean® telegram.
by_ORG	Type of EnOcean® telegram
ar_DB	Data bytes
_Dummy	Placeholder, no other purpose.
dw_ID	Transmitter ID
ptData	Pointer
iErrorld	Describes the error type (see <u>E_KL6581_Err</u> [▶ 41]).
by_STATE	State
bError	This output goes TRUE as soon as an error occurs. This error is described via the <i>iErrorld</i> variable.
idx	Index

## **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

# 4.2.2.2.6 STR\_Teach

Data structure manufacturer ID, type and function.

```
STRUCT

nManufacturerID : WORD;

nTYPE : BYTE;

nFunc : BYTE;

END_STRUCT

END_TYPE
```



Name	Description
nManufacturerID	Manufacturer ID
nTYPE	Туре
nFunc	Function

Development environment	PLC library to include
TwinCAT from v3.1.4020.32	Tc2_EnOcean from v3.4.6.0

# 4.2.2.2.7 str\_Teach\_In

Data structure manufacturer ID, type and profile.

```
TYPE STR_Teach_In :
STRUCT

nManufacturerID : WORD;
nTYPE : BYTE;
nProfile : BYTE;
END_STRUCT
END TYPE
```

Name	Description
nManufacturerID	Manufacturer ID
nTYPE	Туре
nProfile	Profile

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0

#### 4.2.2.2.8 STREnOceanTurnSwitch

STREnOceanTurnSwitch describes the position of the rotary switch at the room control unit.

```
TYPE STREnOceanTurnSwitch:

STRUCT

bStageAuto: BOOL;

bStage_0: BOOL;

bStage_1: BOOL;

bStage_2: BOOL;

bStage_3: BOOL;

END_STRUCT

END_TYPE
```

Name	Description	
bStageAuto	Switch in position "Auto"	
bStage_0	Switch in position "0"	
bStage_1	Switch in position "1"	
bStage_2	Switch in position "2"	
bStage_3	Switch in position "3"	

#### **Prerequisites**

Development environment	PLC library to include
TwinCAT from v3.1.4020.14	Tc2_EnOcean from v3.3.5.0



# 4.3 Integration into TwinCAT

#### 4.3.1 KL6581 with CX5120

This sample shows how to write a simple PLC program for EnOcean® in TwinCAT and how to link it with the hardware. The task is to receive four switch signals of an EnOcean® wireless switch module.

Sample: https://infosys.beckhoff.com/content/1033/tcplclib tc2 enocean/Resources/6200373771.zip



The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tnzip file) is available for import into the TwinCAT project.

#### Hardware

#### Setting up the components

The following hardware is required:

- 1x CX5120 Embedded PC
- 1x KL6581 EnOcean® master terminal
- 1x KL6583-0000 EnOcean® transmitter and receiver
- 1x KL9010 end terminal

Set up the hardware and the EnOcean® components as described in the associated documentation.

This example assumes that the ID of the wireless switch module is known.

#### **Software**

## Creation of the PLC program

Create a new "TwinCAT XAE Project" and create a "Standard PLC Project".

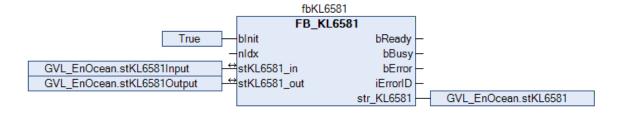
Add the Tc2\_EnOcean library to the PLC project under "References".

Create a global variable list with the name GVL EnOcean and create the following variables:

Name	Туре	Description
stKL6581Input	KL6581 INPUT [▶ 41]	Input variable for the EnOcean® terminal.
stKL6581Output	KL6581 Output [▶ 42]	Output variable for the EnOcean® terminal.
stKL6581	str_KL6581 [▶ 44]	Required for the communication with EnOcean®.

All EnOcean® function blocks must be called in the same task.

Create a MAIN program (CFC) in which the function blocks <u>FB\_KL6581 [▶ 23]</u> and <u>FB\_Rec\_RPS\_Switch [▶ 27]</u> are called. Ensure that the communication block is linked with the structures *stKL6581Input*, *stKL6581Output* and *stKL6581*.



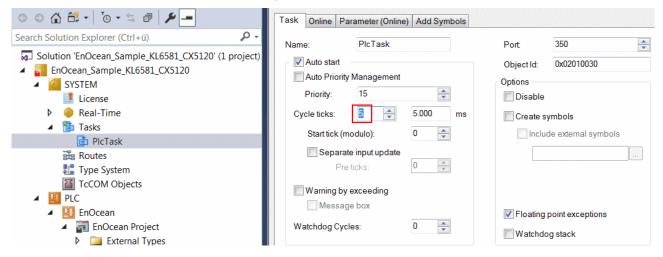


The input  $dw_ID$  of the receive block is linked with the local variable dwId (ID from wireless switch module) and  $str_KL6581$  with the global variable stKL6581.



Go to the task configuration and give the task a lower interval time.

Further conditions can be found in the description of the function block FB KL6581 [▶ 23].



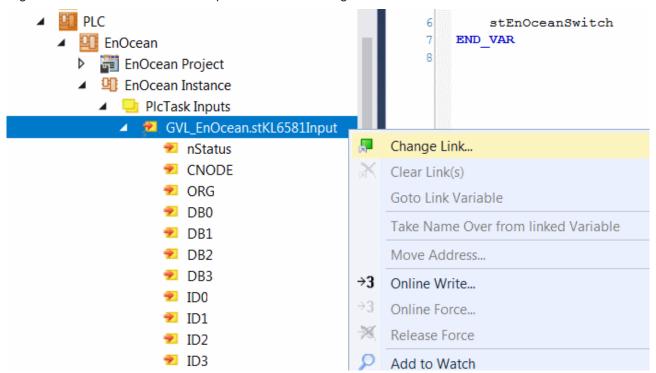
#### I/O configuration

Select the CX as target system and initiate a search for its hardware. In the project instance within the PLC section, you can see that the input and output variables are assigned to the task.

Now link the global variables with the inputs and outputs of the bus terminals.

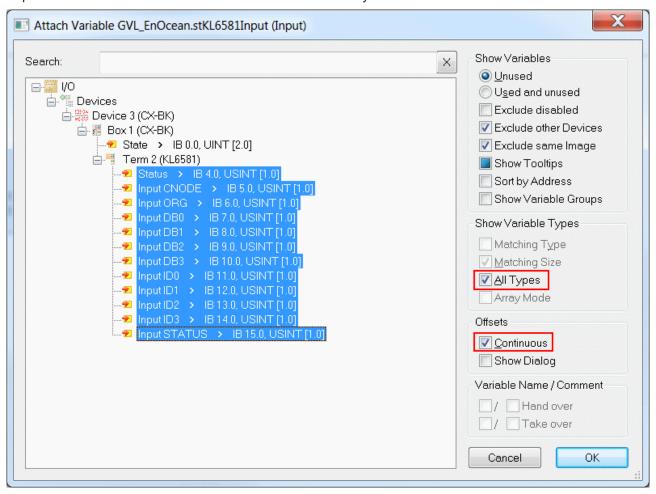
The linking of the EnOcean® variables is described in detail below.

Right-click the structure stKL6581Input and select "Change link".





Under "I/O Configuration" select the terminal, select "All Types" and "Continuous", then select "Status" to "InputStatus" with the left mouse button and the >SHIFT< key. Then click "OK".



You can now check the link. To do this go onto the KL6581 and open it. All terminal data must now be marked with an arrow. If that is the case, then proceed in exactly the same way with the outputs.

#### Also see about this

- KL6581\_INPUT [▶ 41]
- KL6581\_Output [▶ 42]
- str\_KL6581 [▶ 44]



# 5 Appendix

# 5.1 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

#### **Download finder**

Our <u>download finder</u> contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

#### Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for <u>local support and service</u> on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: <a href="https://www.beckhoff.com">www.beckhoff.com</a>

You will also find further documentation for Beckhoff components there.

## **Beckhoff Support**

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- · design, programming and commissioning of complex automation systems
- · and extensive training program for Beckhoff system components

Hotline: +49 5246 963-157 e-mail: support@beckhoff.com

#### **Beckhoff Service**

The Beckhoff Service Center supports you in all matters of after-sales service:

- · on-site service
- · repair service
- · spare parts service
- · hotline service

Hotline: +49 5246 963-460 e-mail: service@beckhoff.com

#### **Beckhoff Headquarters**

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20 33415 Verl Germany

Phone: +49 5246 963-0
e-mail: info@beckhoff.com
web: www.beckhoff.com



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

