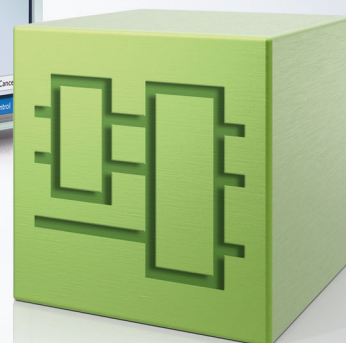
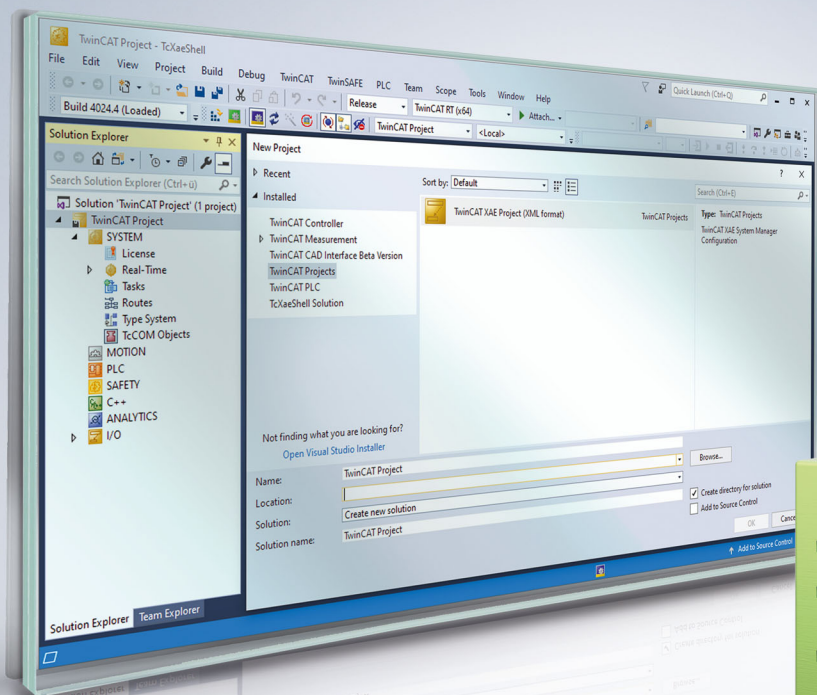


# BECKHOFF New Automation Technology

Handbuch | DE

# TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2\_EnOcean<sup>®</sup>





# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort</b> .....	<b>5</b>
1.1	Hinweise zur Dokumentation .....	5
1.2	Zu Ihrer Sicherheit.....	6
1.3	Hinweise zur Informationssicherheit .....	7
<b>2</b>	<b>Einleitung</b> .....	<b>8</b>
<b>3</b>	<b>EnOcean®</b> .....	<b>9</b>
3.1	Reichweitenplanung .....	9
3.2	Zulassung von EnOcean®-Funk-Technologie .....	10
<b>4</b>	<b>Programmierung</b> .....	<b>11</b>
4.1	POUs.....	12
4.1.1	KL6021-0023.....	13
4.1.2	KL6581.....	24
4.2	DUTs .....	38
4.2.1	KL6021-0023.....	38
4.2.2	KL6581.....	41
4.3	Integration in TwinCAT.....	46
4.3.1	KL6581 mit CX5120.....	46
<b>5</b>	<b>Anhang</b> .....	<b>50</b>
5.1	Support und Service.....	50



# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, stets die aktuell gültige Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiterentwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® und XTS® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Kennzeichnungen führen.



EtherCAT® ist eine eingetragene Marke und patentierte Technologie, lizenziert durch die Beckhoff Automation GmbH, Deutschland.

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwendungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

### Fremdmarken

In dieser Dokumentation können Marken Dritter verwendet werden. Die zugehörigen Markenvermerke finden Sie unter: <https://www.beckhoff.com/trademarks>.

## 1.2 Zu Ihrer Sicherheit

### Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.  
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

### Warnungen vor Personenschäden

#### **GEFAHR**

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

#### **WARNUNG**

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

#### **VORSICHT**

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

### Warnung vor Umwelt- oder Sachschäden

#### **HINWEIS**

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

### Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:  
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

## 1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

## 2 Einleitung

Die BibliothekTc2\_EnOcean ist eine TwinCAT-SPS-Bibliothek zum Datenaustausch mit EnOcean®-Geräten. Diese Bibliothek ist nur in Verbindung mit einer KL6021-0023 oder einer KL6581 einzusetzen.

Für den Nutzer dieser Bibliothek werden folgende Grundkenntnisse vorausgesetzt:

- TwinCAT XAE
- PC und Netzwerkkenntnisse
- Aufbau und Eigenschaften der Beckhoff Embedded-PC und deren Busklemmensystem
- Technologie von EnOcean®-Geräten
- Einschlägige Sicherheitsvorschriften der technischen Gebäudeausrüstung

Diese Softwarebibliothek ist für Gebäudeautomation-Systempartner der Beckhoff Automation GmbH & Co. KG. Die Systempartner sind tätig in dem Bereich Gebäudeautomation und beschäftigen sich mit Errichtung, Inbetriebsetzung, Erweiterung, Wartung und Service von mess-, steuer- und regelungstechnischen Anlagen der technischen Gebäudeausrüstung.

Die Bibliothek Tc2\_EnOcean ist auf allen Hardware-Plattformen einsetzbar, die TwinCAT 3.1 oder höher unterstützen.

Hardware Dokumentation [KL6021-0023](#) und [KL6581](#) im Beckhoff Information System.



### 3 EnOcean®

Die EnOcean®-Funktechnik ermöglicht ein weit reichendes Signal mit geringen Mengen Umgebungsenergie. Mit 50 µWs kann ein serienmäßiges EnOcean®-Funkmodul ohne weiteres ein Signal über eine Distanz von 300 m (im Freifeld) übertragen. Die Signaldauer für ein EnOcean®-Telegramm beträgt ca. 1 tausendstel Sekunde.

- Lizenzfreies 868 MHz Frequenzband mit 1% duty cycle
- Mehrfach-Telegrammaussendung mit Checksumme
- Kurze Telegramme (ca. 1 ms) führen zu geringer Kollisionswahrscheinlichkeit
- Hohe Reichweite: 30 m im Gebäude oder 300 m im Freifeld
- Repeater verfügbar für Erweiterungen
- Uni- und bidirektionale Kommunikation
- Hohe Datenübertragungsraten von 125 kbit/s
- Geringer "Daten-Overhead"
- ASK-Modulation
- Funkprotokoll ist definiert und in Modulen integriert
- Sensorprofile festgelegt und von Nutzern eingehalten
- Eindeutige Sende-ID (32 Bit)
- Keine Interferenz mit DECT-, WLAN-, PMR-Systemen etc.
- Systemdesign verifiziert in Industrieller Umgebung

#### Protokollaufbau

Protokoll	Beschreibung	Länge
ORG	Telegramm Typ	1 Byte
DB_3	Daten Byte 3	1 Byte
DB_2	Daten Byte 2	1 Byte
DB_1	Daten Byte 1	1 Byte
DB_0	Daten 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	Informationsstatus	1 Byte

### 3.1 Reichweitenplanung

Beachten Sie bei der Platzierung der EnOcean®-Geräte die Empfehlungen der EnOcean® Alliance (siehe [www.enocean.de](http://www.enocean.de)). Das Einhalten der Empfehlungen unterstützt eine optimale Reichweite und Störunanfälligkeit.

#### Dämpfung von verschiedenen Materialien

Material	Dämpfung
Holz, Gips, Glas unbeschichtet (ohne Metall)	0...10 %
Backstein, Pressspanplatten	5...35 %
Beton mit Armierung aus Eisen	10...90 %
Metall, Aluminiumkaschierung	90..100 %

**Reichweite**

Material	Reichweite
Sichtverbindung	Typ 30 m in Gängen, bis zu 100 m in Hallen
Rigipswände/Holz	Typ 30 m, durch maximal 5 Wände
Ziegelwände/Gasbeton	Typ 20 m, durch maximal 3 Wände
Stahlbetonwände/-Decken	Typ 10 m, durch maximal 1 Wand/Decke

**Platzierung des KL6583-Moduls**

Das KL6583-Modul beinhaltet Sender, Empfänger sowie die Antenne.

**Abstände**

Der Abstand zu einer Stahlbetondecke sollte mindestens 50 cm betragen, der zu einer Wand 10 cm.

Das KL6583-Modul nicht auf einer Metallplatte verschrauben oder anbringen!

**Umweltbedingungen**

Des Weiteren sind die Umweltbedingungen einzuhalten:

- Luftfeuchtigkeit maximal 95% ohne Betauung
- Umgebungstemperatur 0...55°C

**3.2 Zulassung von EnOcean®-Funk-Technologie****Überprüfen Sie die Zulässigkeit des Betriebs in Ihrem Land**

Die EnOcean®-Transceiver KL6583 sind in folgenden Ländern anmelde- und gebührenfrei zu betreiben: **KL6583-0000: Europäische Union** und **Schweiz**

**KL6583-0100: USA** und **Kanada**

**Die Erlaubnis für den Einsatz in anderen Ländern ist explizit zu klären!**

**KL6583-0100 für USA und Kanada**

**Enthält IC: 5731A-TCM320C**

**Enthält FCC ID: SZV-TCM320C**

Das betroffene Gerät entspricht Teil 15 der FCC Regeln.

Der Betrieb unterliegt den folgenden Bedingungen:

- dieses Gerät darf keine schädlichen Störungen verursachen und
- dieses Gerät muss alle empfangenen Störungen aufnehmen, auch Störungen, die den Betrieb beeinträchtigen.

## 4 Programmierung

### POUs/KL6021-0023

Bausteine	Beschreibung
<a href="#">FB_EnOceanReceive</a> [ <a href="#">▶ 14</a> ]	Kommunikation mit einer KL6021-0023

### POUs/KL6021-0023/Read

Bausteine	Beschreibung
<a href="#">FB_EnOceanPTM100</a> [ <a href="#">▶ 15</a> ]	Empfängt die Signale eines PTM100-Moduls
<a href="#">FB_EnOceanPTM200</a> [ <a href="#">▶ 17</a> ]	Empfängt die Signale eines PTM200-Moduls
<a href="#">FB_EnOceanSTM100</a> [ <a href="#">▶ 18</a> ]	Empfängt die Signale eines STM100-Moduls (veraltet)
<a href="#">FB_EnOceanSTM100Generic</a> [ <a href="#">▶ 21</a> ]	Empfängt die Signale eines STM100-Moduls
<a href="#">FB_EnOceanSTM250</a> [ <a href="#">▶ 22</a> ]	Empfängt die Signale eines STM250-Moduls

### POUs/KL6581

Bausteine	Beschreibung
<a href="#">FB_KL6581</a> [ <a href="#">▶ 24</a> ]	Kommunikation mit einer KL6581

### POUs/KL6581/Read

Bausteine	Beschreibung
<a href="#">FB_Rec_Generic</a> [ <a href="#">▶ 25</a> ]	Empfängt alle Arten von EnOcean®-Telegrammen
<a href="#">FB_Rec_1BS</a> [ <a href="#">▶ 26</a> ]	Empfängt Daten mit ORG-Telegramm 6. Typisches EnOcean®-Gerät: Fensterkontakt
<a href="#">FB_Rec_RPS_Switch</a> [ <a href="#">▶ 27</a> ]	Empfängt Daten mit ORG-Telegramm 5. Typisches EnOcean®-Gerät: Taster
<a href="#">FB_Rec_RPS_Window_Handle</a> [ <a href="#">▶ 28</a> ]	Empfängt Daten mit ORG-Telegramm 5. Typisches EnOcean®-Gerät: Fenstergriff

### POUs/KL6581/Send

Bausteine	Beschreibung
<a href="#">FB_Send_Generic</a> [ <a href="#">▶ 29</a> ]	Sendet beliebige EnOcean®-Telegramme-
<a href="#">FB_Send_4BS</a> [ <a href="#">▶ 30</a> ]	Sendet EnOcean®-Telegramme im 4BS-Format-
<a href="#">FB_Send_RPS_Switch</a> [ <a href="#">▶ 31</a> ]	Sendet EnOcean®-Telegramme im Format eines Tasters.
<a href="#">FB_Send_RPS_SwitchAuto</a> [ <a href="#">▶ 32</a> ]	Sendet EnOcean®-Telegramme im Format eines Tasters.

### POUs/KL6581/Other

Bausteine	Beschreibung
<a href="#">FB_EnOcean_Search</a> [ <a href="#">▶ 33</a> ]	Dieser Baustein erkennt alle EnOcean®-Teilnehmer in seiner Reichweite und zeigt diese an.
<a href="#">FB_Rec_Teach_In</a> [ <a href="#">▶ 34</a> ]	Dieser Baustein zeigt an, wenn in einem EnOcean®-Telegramm das LRN-Bit gesetzt ist unabhängig seiner EnOcean®-ID.
<a href="#">FB_Rec_Teach_In_Ex</a> [ <a href="#">▶ 35</a> ]	Dieser Baustein zeigt an, wenn bei einem EnOcean®-Teilnehmer die Learn-Taste gedrückt wird.

**POUs/KL6581/Function**

Bausteine	Beschreibung
<a href="#">F_Byte_To_Temp [▶ 36]</a>	Diese Funktion wandelt einen Byte-Rohwert in eine REAL-Variable um.
<a href="#">F_Byte_To_TurnSwitch [▶ 37]</a>	Diese Funktion wandelt einen Byte Rohwert in ein Bool Array um.

**DUTs/KL6021-0023/Hardware Types**

Datentypen	Beschreibung
<a href="#">ST_EnOceanInData [▶ 40]</a>	Prozessabbild der Eingänge der KL6021-0023
<a href="#">ST_EnOceanOutData [▶ 40]</a>	Prozessabbild der Ausgänge der KL6021-0023

**DUTs/KL6021-0023**

Datentypen	Beschreibung
<a href="#">E_EnOceanRotarySwitch [▶ 39]</a>	Stellung des Drehschalters am Raumbediengerät
<a href="#">E_EnOceanSensorType [▶ 39]</a>	Sensorentyp
<a href="#">ST_EnOceanReceivedData [▶ 40]</a>	Interne Struktur

**DUTs/KL6581/Hardware Types**

Datentypen	Beschreibung
<a href="#">KL6581_Input [▶ 42]</a>	Prozessabbild der Eingänge der KL6581
<a href="#">KL6581_Output [▶ 43]</a>	Prozessabbild der Ausgänge der KL6581

**DUTs/KL6581**

Datentypen	Beschreibung
<a href="#">AR_EnOceanWindow [▶ 44]</a>	Zustand des Fensters
<a href="#">E_ENOCEAN_Org [▶ 41]</a>	Typ des EnOcean®-Telegramms
<a href="#">E_KL6581_Err [▶ 42]</a>	Fehlermeldungen
<a href="#">STR_EnOceanSwitch [▶ 44]</a>	Zustand der Taster
<a href="#">STR_KL6581 [▶ 45]</a>	Interne Struktur
<a href="#">STR_Teach_In [▶ 46]</a>	Datenstruktur Hersteller-ID, Typ und Profil
<a href="#">STREnOceanTurnSwitch [▶ 46]</a>	Stellung des Drehschalters am Raumbediengerät

## 4.1 POU's

**KL6021-0023**

Bausteine	Beschreibung
<a href="#">FB_EnOceanReceive [▶ 14]</a>	Kommunikation mit einer KL6021-0023

**KL6021-0023/Read**

Bausteine	Beschreibung
<a href="#">FB_EnOceanPTM100 [▶ 15]</a>	Empfängt die Signale eines PTM100-Moduls
<a href="#">FB_EnOceanPTM200 [▶ 17]</a>	Empfängt die Signale eines PTM200-Moduls
<a href="#">FB_EnOceanSTM100 [▶ 18]</a>	Empfängt die Signale eines STM100-Moduls (veraltet)

Bausteine	Beschreibung
<a href="#">FB_EnOceanSTM100Generic [▶ 21]</a>	Empfängt die Signale eines STM100-Moduls
<a href="#">FB_EnOceanSTM250 [▶ 22]</a>	Empfängt die Signale eines STM250-Moduls

**KL6581**

Bausteine	Beschreibung
<a href="#">FB_KL6581 [▶ 24]</a>	Kommunikation mit einer KL6581

**KL6581/Read**

Bausteine	Beschreibung
<a href="#">FB_Rec_Generic [▶ 25]</a>	Empfängt alle Arten von EnOcean®-Telegrammen.
<a href="#">FB_Rec_1BS [▶ 26]</a>	Empfängt Daten mit ORG-Telegramm 6. Typisches EnOcean®-Gerät: Fensterkontakt
<a href="#">FB_Rec_RPS_Switch [▶ 27]</a>	Empfängt Daten mit ORG-Telegramm 5. Typisches EnOcean®-Gerät: Taster
<a href="#">FB_Rec_RPS_Window_Handle [▶ 28]</a>	Empfängt Daten mit ORG-Telegramm 5. Typisches EnOcean®-Gerät: Fenstergriff

**KL6581/Send**

Bausteine	Beschreibung
<a href="#">FB_Send_Generic [▶ 29]</a>	Sendet beliebige EnOcean®-Telegramme.
<a href="#">FB_Send_4BS [▶ 30]</a>	Sendet EnOcean®-Telegramme im 4BS-Format.
<a href="#">FB_Send_RPS_Switch [▶ 31]</a>	Sendet EnOcean®-Telegramme im Format eines Tasters.
<a href="#">FB_Send_RPS_SwitchAuto [▶ 32]</a>	Sendet EnOcean®-Telegramme im Format eines Tasters.

**KL6581/Other**

Bausteine	Beschreibung
<a href="#">FB_EnOcean_Search [▶ 33]</a>	Dieser Baustein erkennt alle EnOcean®-Teilnehmer in seiner Reichweite und zeigt diese an.
<a href="#">FB_Rec_Teach_In [▶ 34]</a>	Dieser Baustein zeigt an, wenn in einem EnOcean®-Telegramm das LRN-Bit gesetzt ist unabhängig seiner EnOcean®-ID.
<a href="#">FB_Rec_Teach_In_Ex [▶ 35]</a>	Dieser Baustein zeigt an, wenn bei einem EnOcean®-Teilnehmer die Learn-Taste gedrückt wird.

**KL6581/Function**

Bausteine	Beschreibung
<a href="#">F_Byte_To_Temp [▶ 36]</a>	Diese Funktion wandelt einen Byte-Rohwert in eine REAL-Variable um.
<a href="#">F_Byte_To_TurnSwitch [▶ 37]</a>	Diese Funktion wandelt einen Byte Rohwert in ein Bool Array um.

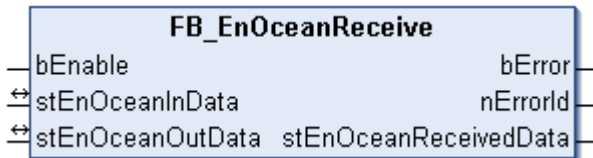
**4.1.1 KL6021-0023**

Bausteine	Beschreibung
<a href="#">FB_EnOceanReceive [▶ 14]</a>	Kommunikation mit einer KL6021-0023

**Read**

Bausteine	Beschreibung
<a href="#">FB_EnOceanPTM100 [► 15]</a>	Empfängt die Signale eines PTM100-Moduls
<a href="#">FB_EnOceanPTM200 [► 17]</a>	Empfängt die Signale eines PTM200-Moduls
<a href="#">FB_EnOceanSTM100 [► 18]</a>	Empfängt die Signale eines STM100-Moduls (veraltet)
<a href="#">FB_EnOceanSTM100Generic [► 21]</a>	Empfängt die Signale eines STM100-Moduls
<a href="#">FB_EnOceanSTM250 [► 22]</a>	Empfängt die Signale eines STM250-Moduls

**4.1.1.1 FB\_EnOceanReceive**



Der Funktionsbaustein *FB\_EnOceanReceive()* ist ein Empfangsbaustein, der die von den EnOcean®-Modulen gesendeten Telegramme in der Struktur *stEnOceanReceivedData* zur Verfügung stellt. Diese Struktur kann dann mit weiteren Bausteinen ausgewertet werden. In der Dokumentation dieser Bausteine sind auch Programmbeispiele aufgeführt, die die Funktionsweise näher erläutern.

**Eingänge**

```

VAR_INPUT
  bEnable      : BOOL := FALSE;
END_VAR
  
```

Name	Typ	Beschreibung
bEnable	BOOL	Ein positives Signal an diesem Eingang setzt den Baustein aktiv. Bei einem negativen Signal am Eingang wird im Baustein keine Funktion ausgeführt.

**Ein-/Ausgänge**

```

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

Name	Typ	Beschreibung
stEnOceanInData	<a href="#">ST_EnOceanInData [► 40]</a>	Wird mit den Eingangsadressen der KL6021-0023 im System Manager verknüpft.
stEnOceanOutData	<a href="#">ST_EnOceanOutData [► 40]</a>	Wird mit den Ausgangsadressen der KL6021-0023 im System Manager verknüpft.

**Ausgänge**

```

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

Name	Typ	Beschreibung
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>nErrorID</i> beschrieben.
nErrorId	UDINT	Beschreibt die Art des <a href="#">Fehlers [► 24]</a> .

Name	Typ	Beschreibung
stEnOceanReceivedData	ST_EnOceanReceivedData [▶ 40]	In dieser Struktur werden die empfangenen Daten abgelegt.

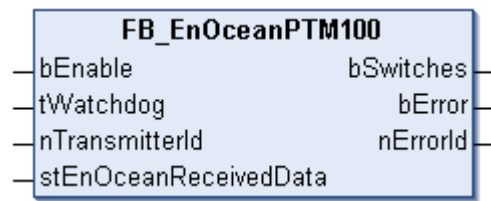
**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.1.2 Read**

Bausteine	Beschreibung
FB_EnOceanPTM100 [▶ 15]	Empfängt die Signale eines PTM100-Moduls
FB_EnOceanPTM200 [▶ 17]	Empfängt die Signale eines PTM200-Moduls
FB_EnOceanSTM100 [▶ 18]	Empfängt die Signale eines STM100-Moduls (veraltet)
FB_EnOceanSTM100Generic [▶ 21]	Empfängt die Signale eines STM100-Moduls
FB_EnOceanSTM250 [▶ 22]	Empfängt die Signale eines STM250-Moduls

**4.1.1.2.1 FB\_EnOceanPTM100**



Der Funktionsbaustein *FB\_EnOceanPTM100()* gibt eine anwenderfreundliche Auswertung über den Zustand eines EnOcean®-PTM100-Moduls. Hierzu ist die Verwendung des Funktionsblocks *FB\_EnOceanReceive()* [▶ 14] notwendig.

Im Unterschied zum PTM200- und PTM250-Modul, kann beim PTM100-Modul nur ein Taster gleichzeitig gedrückt werden. Des Weiteren unterstützt das PTM100-Modul acht, statt vier Taster.



Zu jedem verwendeten Tasten-Modul muss eine neue Instanz dieses Bausteines angelegt werden.

**Eingänge**

```

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

Name	Typ	Beschreibung
bEnable	BOOL	Ein positives Signal an diesem Eingang setzt den Baustein aktiv. Bei einem negativen Signal am Eingang wird im Baustein keine Funktion ausgeführt und alle Ausgänge werden auf 0 bzw. FALSE gesetzt.
tWatchdog	TIME	Überwachungszeit. Innerhalb dieser Zeit müssen neue Informationen über den unten näher beschriebenen Eingang <i>stEnOceanReceivedData</i> in diesen Baustein gelangen. Ist diese Zeit auf <i>t#0s</i> gesetzt, so ist die Watchdog-Funktion inaktiv.

Name	Typ	Beschreibung
nTransmitterId	UDINT	ID des EnOcean®-Modules, auf den der Baustein reagieren soll.
stEnOceanReceivedData	ST_EnOceanReceivedData [► 40]	Informationen und notwendige Verbindung zum EnOcean®-Empfängerbaustein FB_EnOceanReceive() [► 14]. Diese Informationen sind in einer Struktur hinterlegt.

 **Ausgänge**

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

Name	Typ	Beschreibung
bSwitches	ARRAY OF BOOL	Dieses Feld von 8 boolschen Werten beschreibt die Zustände der 8 Taster auf dem Taster-Modul.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable nErrorID beschrieben.
nErrorId	UDINT	Beschreibt die Art des Fehlers [► 24].

Anhand des folgenden Programmbeispiels soll die Funktionsweise des Bausteines näher beschrieben werden:

```
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 diesem Beispielprogramm werden 2 Sendemodule (PTM100) abgefragt, ein Sendemodul mit der Transmitter-Id 16#C4 und ein anderes mit der Transmitter-Id 16#C5. Für beide Sendemodule wird jeweils ein Funktionsbaustein FB\_EnOceanPTM100 angelegt. Beide Funktionsbausteine erhalten ihre Informationen von einem vorangeschalteten Empfängerbaustein FB\_EnOceanReceive und sind nur dann aktiv (Eingang bEnable), wenn der Empfängerbaustein aktiv und nicht in Störung ist. Die Taster des ersten

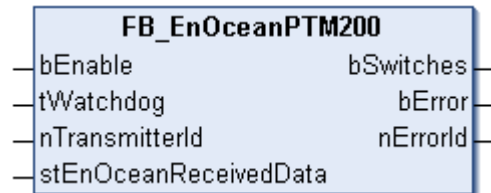


Sendemoduls werden zur weiteren Auswertung einem gleich großen booleschen Array bSwitches1 zugeordnet, während die Taster des zweiten Sendemodules einzelnen booleschen Variablen bSwitches2\_1 bis bSwitches2\_8 zugewiesen werden - beide Möglichkeiten sind denkbar.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.1.2.2 FB\_EnOceanPTM200**



Der Funktionsbaustein *FB\_EnOceanPTM200()* gibt eine anwenderfreundliche Auswertung über den Zustand eines EnOcean®-PTM200- oder PTM250-Moduls. Hierzu ist die Verwendung des Funktionsblocks *FB\_EnOceanReceive()* [▶ 14] notwendig.

Im Unterschied zum PTM100-Modul, können beim PTM200-Module und beim PTM250-Modul bis zu zwei Taster gleichzeitig gedrückt werden. Des Weiteren unterstützt das PTM200- und PTM250-Modul vier, statt acht Taster.



Zu jedem verwendeten Tasten-Modul muss eine neue Instanz dieses Bausteines angelegt werden.

**Eingänge**

```

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

Name	Typ	Beschreibung
bEnable	BOOL	Ein positives Signal an diesem Eingang setzt den Baustein aktiv. Bei einem negativen Signal am Eingang wird im Baustein keine Funktion ausgeführt und alle Ausgänge werden auf 0 bzw. FALSE gesetzt.
tWatchdog	TIME	Überwachungszeit. Innerhalb dieser Zeit müssen neue Informationen über den unten näher beschriebenen Eingang <i>stEnOceanReceivedData</i> in diesen Baustein gelangen. Ist diese Zeit auf t#0s gesetzt, so ist die Watchdog-Funktion inaktiv.
nTransmitterId	UDINT	ID des EnOcean®-Modules, auf den der Baustein reagieren soll.
stEnOceanReceivedData	ST_EnOceanReceivedData [▶ 40]	Informationen und notwendige Verbindung zum EnOcean®-Empfängerbaustein <i>FB_EnOceanReceive()</i> [▶ 14]. Diese Informationen sind in einer Struktur hinterlegt.

**Ausgänge**

```

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

Name	Typ	Beschreibung
bSwitches	ARRAY OF BOOL	Dieses Feld von 4 boolschen Werten beschreibt die Zustände der 4 Taster auf dem Taster-Modul.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>nErrorID</i> beschrieben.
nErrorId	UDINT	Beschreibt die Art des <a href="#">Fehlers</a> (► 24).

Anhand des folgenden Programmbeispiels soll die Funktionsweise des Bausteines näher beschrieben werden:

```

PROGRAM MAIN
VAR
    fbEnOceanReceive : FB_EnOceanReceive;
    fbEnOceanPTM100_1 : FB_EnOceanPTM200;
    fbEnOceanPTM100_2 : FB_EnOceanPTM200;
    bSwitches1 : ARRAY [0..3] OF BOOL;
    bSwitches2_1 : BOOL;
    bSwitches2_2 : BOOL;
    bSwitches2_3 : BOOL;
    bSwitches2_4 : BOOL;
END_VAR

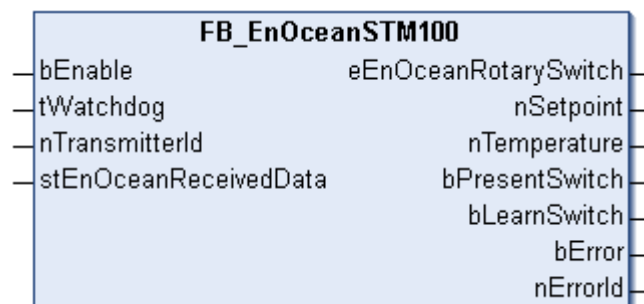
fbEnOceanReceive (
    bEnable := TRUE,
    stEnOceanInData := stEnOceanInData
    stEnOceanOutData := stEnOceanOutData);
fbEnOceanPTM200_1 (
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C6,
    tWatchdog:=t#0s,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
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 diesem Beispielpogramm werden 2 Sendemodule (PTM200/PTM250) abgefragt, ein Sendemodul mit der Transmitter-Id 16#C6 und ein anderes mit der Transmitter-Id 16#C7. Für beide Sendemodule wird jeweils ein Funktionsbaustein *FB\_EnOceanPTM200* angelegt. Beide Funktionsbausteine erhalten ihre Informationen von einem vorangeschalteten Empfängerbaustein *FB\_EnOceanReceive* (► 14) und sind nur dann aktiv (Eingang *bEnable*), wenn der Empfängerbaustein aktiv und nicht in Störung ist. Die Taster des ersten Sendemoduls werden zur weiteren Auswertung einem gleich großen boolschen Array *bSwitches1* zugeordnet, während die Taster des zweiten Sendemoduls einzelnen boolschen Variablen *bSwitches2\_1* bis *bSwitches2\_4* zugewiesen werden - beide Möglichkeiten sind denkbar.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.1.2.3 FB\_EnOceanSTM100**



**i** **Veraltet**

Bei neuen Projekten sollte der Baustein `FB_EnOceanSTM100Generic()` [► 21] verwendet werden!

Der Funktionsbaustein `FB_EnOceanSTM100()` gibt eine anwenderfreundliche Auswertung über die Daten eines EnOcean®-STM100-Moduls. Hierzu ist die Verwendung des Funktionsblocks `FB_EnOceanReceive()` [► 14] notwendig.

**i** Zu jedem verwendeten STM100-Modul muss eine neue Instanz dieses Bausteines angelegt werden.

**🔌 Eingänge**

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

Name	Typ	Beschreibung
bEnable	BOOL	Ein positives Signal an diesem Eingang setzt den Baustein aktiv. Bei einem negativen Signal am Eingang wird im Baustein keine Funktion ausgeführt und alle Ausgänge werden auf 0 bzw. FALSE gesetzt.
tWatchdog	TIME	Überwachungszeit. Innerhalb dieser Zeit müssen neue Informationen über den unten näher beschriebenen Eingang <code>stEnOceanReceivedData</code> in diesen Baustein gelangen. Ist diese Zeit auf <code>t#0s</code> gesetzt, so ist die Watchdog-Funktion inaktiv.
nTransmitterId	UDINT	ID des EnOcean®-Modules, auf den der Baustein reagieren soll.
stEnOceanReceivedData	ST_EnOceanReceivedData [► 40]	Informationen und notwendige Verbindung zum EnOcean®-Empfängerbaustein <code>FB_EnOceanReceive()</code> [► 14]. Diese Informationen sind in einer Struktur hinterlegt.

**🔌 Ausgänge**

```
VAR_OUTPUT
  eEnOceanRotarySwitch : E_EnOceanRotarySwitch;
  nSetpoint             : INT;
  nTemperature          : INT;
  bPresentSwitch       : BOOL;
  bLearnSwitch         : BOOL;
  bError                : BOOL := FALSE;
  nErrorId             : UDINT := 0;
END_VAR
```

Name	Typ	Beschreibung
eEnOceanRotarySwitch	E_EnOceanRotarySwitch [► 39]	Die Ausgabe an diesem Ausgang beschreibt die Stellung des Drehschalters am Raumbediengerät.
nSetpoint	INT	An dieser Ausgangsvariablen liegt der am Gerät eingestellte Sollwert an. Dieser kann Werte im Bereich von -100 bis +100 annehmen.
nTemperature	INT	Hier wird die gemessene Temperatur in 1/10°C ausgegeben mit einem Messbereich von 0°C bis 40°C. Bei ausgelöstem Watchdog vermutet der Baustein einen drahtbruchähnlichen Fehler und der Wert wird fest auf 850°C gesetzt.

Name	Typ	Beschreibung
bPresentSwitch	BOOL	Bei Aktivierung der Anwesenheitstaste am Raumbediengerät wird dieser Ausgang <i>TRUE</i> .
bLearnSwitch	BOOL	Bei Aktivierung der Anlerntaste am Raumbediengerät wird dieser Ausgang <i>TRUE</i> .
bError	BOOL	Der Ausgang wird <i>TRUE</i> , sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>nErrorID</i> beschrieben.
nErrorId	UDINT	Beschreibt die Art des <u>Fehlers</u> [► 24].

Anhand des folgenden Programmbeispiels soll die Funktionsweise des Bausteines näher beschrieben werden:

```
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);

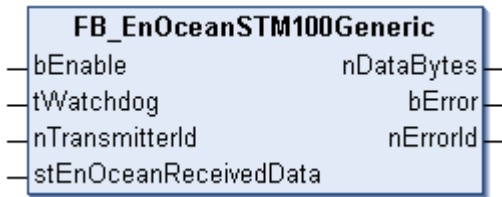
fbEnOceanSTM100_1 (
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceve.bEnable,
    nTransmitterId := 16#000000C4,
    tWatchdog:=t#1h,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData
    nTemperature => Temperature[1],
    nSetpoint => nSetpoint[1] ,
    eEnOceanRotarySwitch => 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 diesem Beispielpogramm werden 2 Raumbediengeräte abgefragt, einer mit der Transmitter-Id 16#000000C4 und ein anderer mit der Transmitter-Id 16#000000C5. Für beide Module wird jeweils ein Funktionsbaustein *FB\_EnOceanSTM100* angelegt. Beide Funktionsbausteine erhalten ihre Informationen von einem vorangeschalteten Empfängerbaustein *FB\_EnOceanReceive* [► 14] und sind nur dann aktiv (Eingang *bEnable*), wenn der Empfängerbaustein aktiv und nicht in Störung ist. Das erste Gerät wird durch die Watchdog-Funktion überwacht, wobei innerhalb von 1 Stunde neue Werte an die Steuerung übertragen werden müssen, das zweite Gerät ist ohne Watchdog-Überwachung programmiert. Zur weiteren Auswertung sind die an den Funktionsbausteinen ausgegebenen Werten Merkern zugewiesen.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

4.1.1.2.4 FB\_EnOceanSTM100Generic



Der Funktionsbaustein *FB\_EnOceanSTM100Generic()* gibt eine anwenderfreundliche Auswertung über die Daten eines EnOcean®-STM100-Moduls. Hierzu ist die Verwendung des Funktionsblocks *FB\_EnOceanReceive()* [► 14] notwendig.



Zu jedem verwendeten STM100-Modul muss eine neue Instanz dieses Bausteines angelegt werden.

Eingänge

```

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

Name	Typ	Beschreibung
bEnable	BOOL	Ein positives Signal an diesem Eingang setzt den Baustein aktiv. Bei einem negativen Signal am Eingang wird im Baustein keine Funktion ausgeführt und alle Ausgänge werden auf 0 bzw. FALSE gesetzt.
tWatchdog	TIME	Überwachungszeit. Innerhalb dieser Zeit müssen neue Informationen über den unten näher beschriebenen Eingang <i>stEnOceanReceivedData</i> in diesen Baustein gelangen. Ist diese Zeit auf <i>t#0s</i> gesetzt, so ist die Watchdog-Funktion inaktiv.
nTransmitterId	UDINT	ID des EnOcean®-Modules, auf den der Baustein reagieren soll.
stEnOceanReceivedData	ST_EnOceanReceivedData [► 40]	Informationen und notwendige Verbindung zum EnOcean®-Empfängerbaustein <i>FB_EnOceanReceive()</i> [► 14]. Diese Informationen sind in einer Struktur hinterlegt.

Ausgänge

```

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

Name	Typ	Beschreibung
nDataBytes	ARRAY OF BYTE	4 Bytes großes Array mit dem Nutzdaten, die das STM100-Modul versendet hat. Die Bedeutung der einzelnen Bytes ist herstellerabhängig.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>nErrorID</i> beschrieben.
nErrorId	UDINT	Beschreibt die Art des Fehlers [► 24].

Anhand des folgenden Programmbeispiels soll die Funktionsweise des Bausteines näher beschrieben werden:

```

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

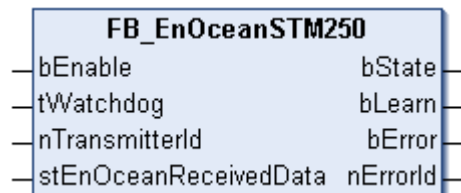
fbEnOceanReceive(
    bEnable := TRUE,
    stEnOceanInData := stEnOceanInData,
    stEnOceanOutData := stEnOceanOutData);
fbEnOceanSTM100_1(
    bEnable := NOT fbEnOceanReceive.bError AND fbEnOceanReceive.bEnable,
    nTransmitterId := 16#000000C4,
    tWatchdog:=t#1h,
    stEnOceanReceivedData := fbEnOceanReceive.stEnOceanReceivedData);
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 diesem Beispielpogramm werden zwei EnOcean®-Sendemodule abgefragt, einer mit der Transmitter-Id 16#000000C4 und ein anderer mit der Transmitter-Id 16#000000C5. Für beide Transmitter wird jeweils ein Funktionsbaustein *FB\_EnOceanSTM100Generic* angelegt. Beide Funktionsbausteine erhalten ihre Informationen von einem vorangeschalteten Empfängerbaustein *FB\_EnOceanReceive* [► 14] und sind nur dann aktiv (Eingang *bEnable*), wenn der Empfängerbaustein aktiv und nicht in Störung ist. Das erste Gerät wird durch die Watchdog-Funktion überwacht, wobei innerhalb von 1 Stunde neue Werte an die Steuerung übertragen werden müssen, das zweite Gerät ist ohne Watchdog-Überwachung programmiert. Zur weiteren Auswertung sind die an den Funktionsbausteinen ausgegebenen Werten Variablen zugewiesen. Für eine weitere Verwendung der Werte müssten diese noch in physikalische Größen skaliert werden. Wie die Umrechnung zu erfolgen hat, ist aus dem Datenblatt des Sensors zu entnehmen.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.1.2.5 FB\_EnOceanSTM250**



Der Funktionsbaustein *FB\_EnOceanSTM250()* gibt eine anwenderfreundliche Auswertung über die Daten eines EnOcean®-STM250-Moduls. Hierzu ist die Verwendung des Funktionsblocks *FB\_EnOceanReceive()* [► 14] notwendig.



Zu jedem verwendeten STM250-Modul muss eine neue Instanz dieses Bausteines angelegt werden.

 **Eingänge**

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

Name	Typ	Beschreibung
bEnable	BOOL	Ein positives Signal an diesem Eingang setzt den Baustein aktiv. Bei einem negativen Signal am Eingang wird im Baustein keine Funktion ausgeführt und alle Ausgänge werden auf 0 bzw. FALSE gesetzt.
tWatchdog	TIME	Überwachungszeit. Innerhalb dieser Zeit müssen neue Informationen über den unten näher beschriebenen Eingang <i>stEnOceanReceivedData</i> in diesen Baustein gelangen. Ist diese Zeit auf <i>t#0s</i> gesetzt, so ist die Watchdog-Funktion inaktiv.
nTransmitterId	UDINT	ID des EnOcean®-Modules, auf den der Baustein reagieren soll.
stEnOceanReceivedData	ST_EnOceanReceivedData <a href="#"> &gt; 40</a>	Informationen und notwendige Verbindung zum EnOcean®-Empfängerbaustein <i>FB_EnOceanReceive()</i> <a href="#"> &gt; 14</a> . Diese Informationen sind in einer Struktur hinterlegt.

 **Ausgänge**

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

Name	Typ	Beschreibung
bState	BOOL	Bei Aktivierung des Reedkontakts am STM250-Modul wird dieser Ausgang <i>TRUE</i> (Kontakt geschlossen).
bLearn	BOOL	Bei Aktivierung der Lerntaste am STM250-Modul wird dieser Ausgang <i>FALSE</i> .
bError	BOOL	Der Ausgang wird <i>TRUE</i> , sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>nErrorID</i> beschrieben.
nErrorId	UDINT	Beschreibt die Art des <a href="#">Fehlers</a> <a href="#"> &gt; 24</a> .

Anhand des folgenden Programmbeispiels soll die Funktionsweise des Bausteines näher beschrieben werden:

```
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 diesem Beispielprogramm wird ein STM250 Modul mit der Transmitter-Id 16#000008CA abgefragt. Hierzu wird der Funktionsbaustein *FB\_EnOceanSTM250* angelegt. Dieser Funktionsbaustein erhält Informationen von einem vorangeschalteten Empfängerbaustein *FB\_EnOceanReceive* [► 14] und ist nur dann aktiv (Eingang *bEnable*), wenn der Empfängerbaustein aktiv und nicht in Störung ist. Zur weiteren Auswertung sind die an den Funktionsbaustein ausgegebenen Werten Variablen zugewiesen.

**Voraussetzungen**

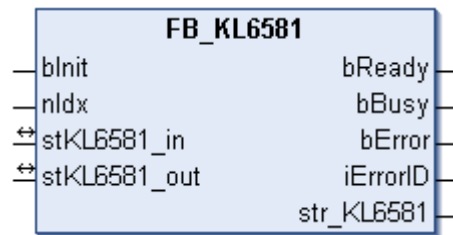
Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.1.3 Fehlercodes**

Wert (hex)	Beschreibung
0x0000	Kein Fehler.
0x0001	Prüfsummenfehler.
0x0002	Watchdogüberwachung.
0x0003	Pufferüberlauf (in der KL6023)
0x0004	Noch keine Daten vom Sensor empfangen.

**4.1.2 KL6581**

**4.1.2.1 FB\_KL6581**



Dieser Funktionsbaustein übernimmt die Kommunikation mit der EnOcean®-Busklemme KL6581. Über diesen Baustein wird die KL6581 konfiguriert und der Datenaustausch mit dem EnOcean®-Netzwerk gestartet.



**Einschränkungen**

- Nur ein Aufruf pro Instanz
- Aufruf muss einmal pro SPS-Zyklus erfolgen
- Instanz muss in derselben SPS-Task aufgerufen werden, wie die ihm zugeordneten Send- und Empfangsbausteine
- Maximal 64 Instanzen pro SPS-Projekt zulässig

**Eingänge**

```

VAR_INPUT
  bInit : BOOL;
  nIdx  : USINT := 1;
END_VAR
  
```

Name	Typ	Beschreibung
bInit	BOOL	Aktiviert den Baustein, der die KL6301 als erstes konfiguriert und dann in den Datenaustausch setzt.
nIdx	USINT	Die idx-Nummer muss beim Einsatz von mehr als einer Busklemme pro SPS-Programm für jede KL6581 eindeutig sein (gültige Werte: 1...64).



 **Ein-/Ausgänge**

```
VAR_IN_OUT
  stKL6581_in : KL6581_Input;
  stKL6581_out : KL6581_Output;
END_VAR
```

Name	Typ	Beschreibung
stKL6581_in	KL6581_Input [▶ 42]	Wird mit den Eingangsadressen der KL6581 im System Manager verknüpft.
stKL6581_out	KL6581_Output [▶ 43]	Wird mit den Ausgangsadressen der KL6581 im System Manager verknüpft.

 **Ausgänge**

```
VAR_OUTPUT
  bReady : BOOL;
  bBusy : BOOL;
  bError : BOOL;
  iErrorID : E_KL6581_Err;
  str_KL6581 : STR_KL6581;
END_VAR
```

Name	Typ	Beschreibung
bReady	BOOL	Der Baustein ist bereit Daten zu senden und zu empfangen.
bBusy	BOOL	Der Baustein ist aktiv. Es können noch keine neuen Daten gesendet werden.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>iErrorID</i> beschrieben.
iErrorID	E_KL6581_Err [▶ 42]	Beschreibt die Art des Fehlers.
str_KL6581	STR_KL6581 [▶ 45]	Wird mit den Send- und Receive-Bausteinen verbunden.

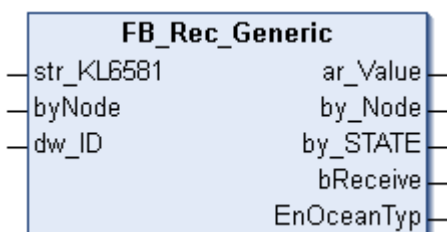
**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.2 Read**

Bausteine	Beschreibung
FB_Rec_Generic [▶ 25]	Empfängt alle Arten von EnOcean®-Telegrammen.
FB_Rec_1BS [▶ 26]	Empfängt Daten mit ORG-Telegramm 6. Typisches EnOcean®-Gerät: Fensterkontakt
FB_Rec_RPS_Switch [▶ 27]	Empfängt Daten mit ORG-Telegramm 5. Typisches EnOcean®-Gerät: Taster
FB_Rec_RPS_Window_Handle [▶ 28]	Empfängt Daten mit ORG-Telegramm 5. Typisches EnOcean®-Gerät: Fenstergriff

**4.1.2.2.1 FB\_Rec\_Generic**



Dieser Funktionsbaustein empfängt alle Daten, die über EnOcean® empfangen wurden. Dieser Baustein kann für alle Arten von EnOcean®-Telegrammen verwendet werden.

Die Daten muss der Anwender selbst interpretieren. Dazu ist die Dokumentation des Herstellers des sendenden EnOcean®-Gerätes notwendig.

**Eingänge**

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

Name	Typ	Beschreibung
str_KL6581	STR_KL6581 [▶ 45]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [▶ 24] verbunden.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.
dw_ID	DWORD	EnOcean®-ID, die empfangen werden soll.

**Ausgänge**

```
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	Typ	Beschreibung
ar_Value	ARRAY OF BYTE	EnOcean®-Daten 4 Byte.
by_Node	BYTE	Node Nummer der KL6583, die das EnOcean®-Telegramm empfangen hat.
by_STATE	BYTE	EnOcean® STATUS Field.
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.
EnOceanTyp	E_EnOcean_Org [▶ 41]	EnOcean® ORG Field.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.2.2 FB\_Rec\_1BS**



Dieser Funktionsbaustein empfängt Daten, die über EnOcean® empfangen wurden. Dieser Baustein wird zum Beispiel zur Anbindung von Fensterkontakten verwendet (ORG FIELD 6).

**Eingänge**

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

Name	Typ	Beschreibung
str_KL6581	STR_KL6581 [▶ 45]	Wird mit der Datenstruktur des Bausteins <b>FB_KL6581()</b> [▶ 24] verbunden.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.
dw_ID	DWORD	EnOcean®-ID, die empfangen werden soll.

**Ausgänge**

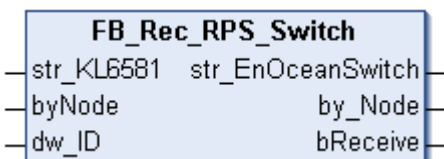
```
VAR_OUTPUT
  bOpen      : BOOL;
  bClose     : BOOL;
  bLRN       : BOOL;
  by_Node    : BYTE;
  bReceive   : BOOL := TRUE;
END_VAR
```

Name	Typ	Beschreibung
bOpen	BOOL	Kontakt offen
bClose	BOOL	Kontakt geschlossen
bLRN	BOOL	LRN-Taste gedrückt
by_Node	BYTE	Node Nummer der KL6583, die das EnOcean®-Telegramm empfangen hat.
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.2.3 FB\_Rec\_RPS\_Switch**



Dieser Funktionsbaustein empfängt Daten eines Schalters, die über EnOcean® empfangen wurden. Der Baustein gibt die Daten in einer Datenstruktur aus (ORG Field 5).

**Eingänge**

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

Name	Typ	Beschreibung
str_KL6581	STR_KL6581 [▶ 45]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [▶ 24] verbunden.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.
dw_ID	DWORD	EnOcean®-ID, die empfangen werden soll.

 **Ausgänge**

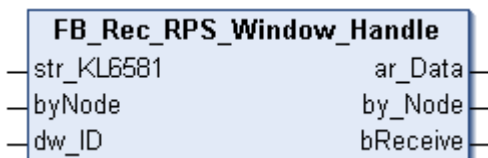
```
VAR_OUTPUT
  str_EnOceanSwitch : STR_EnOceanSwitch
  by_Node           : BYTE;
  bReceive          : BOOL := TRUE;
END_VAR
```

Name	Typ	Beschreibung
str_EnOceanSwitch	STR_EnOceanSwitch [▶ 44]	Daten des Schalters
by_Node	BYTE	Node Nummer der KL6583, die das EnOcean®-Telegramm empfangen hat.
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.2.4 FB\_Rec\_RPS\_Window\_Handle**



Dieser Funktionsbaustein empfängt Daten eines Fenstergriffes (WINDOW HANDLE), die über EnOcean® empfangen wurden. Der Baustein gibt die Daten in einer Datenstruktur aus (ORG Field 5).

 **Eingänge**

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

Name	Typ	Beschreibung
str_KL6581	STR_KL6581 [▶ 45]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [▶ 24] verbunden.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.
dw_ID	DWORD	EnOcean®-ID, die empfangen werden soll.

**Ausgänge**

```
VAR_OUTPUT
  ar_Data : AR_EnOceanWindow;
  by_Node : BYTE;
  bReceive : BOOL := TRUE;
END_VAR
```

Name	Typ	Beschreibung
ar_Data	AR_EnOceanWindow [▶ 44]	Daten des Fenstergriffs
by_Node	BYTE	Node Nummer der KL6583, die das EnOcean®-Telegramm empfangen hat.
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.

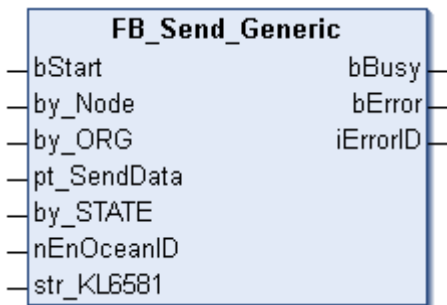
**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.3 Send**

Bausteine	Beschreibung
FB_Send_Generic [▶ 29]	Sendet beliebige EnOcean®-Telegramme.
FB_Send_4BS [▶ 30]	Sendet EnOcean®-Telegramme im 4BS-Format.
FB_Send_RPS_Switch [▶ 31]	Sendet EnOcean®-Telegramme im Format eines Tasters.
FB_Send_RPS_SwitchAuto [▶ 32]	Sendet EnOcean®-Telegramme im Format eines Tasters.

**4.1.2.3.1 FB\_Send\_Generic**



Dieser Funktionsbaustein sendet Daten über EnOcean®. Die Art und der Dateninhalt sind beliebig. Mit diesem Baustein können alle Arten von EnOcean®-Datentelegrammen versendet werden.

**Eingänge**

```
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	Typ	Beschreibung
bStart	BOOL	Positive Flanke sendet die Daten.

Name	Typ	Beschreibung
by_Node	BYTE	Adresse des KL6583 Moduls, an die das Telegramm gesendet werden soll (gültige Werte: 1...8).
by_ORG	E_EnOcean_Org [▶ 41]	ORG Field des EnOcean®-Telegramms
pt_SendData	DWORD	Pointer auf die Daten, die gesendet werden sollen. Mit ADR wird die Pointeradresse ermittelt. Der Pointer muss auf eine 4 Byte Variable zeigen.
by_STATE	BYTE	EnOcean®-STATE. Kann vom TCM-Modul verändert werden.
nEnOceanID	BYTE	Virtuelle EnOcean®-ID. Auf die reale EnOcean®-ID wird ein Wert von 0...127 addiert (gültige Werte: 0...127).
str_KL6581	STR_KL6581 [▶ 45]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [▶ 24] verbunden.

**Ausgänge**

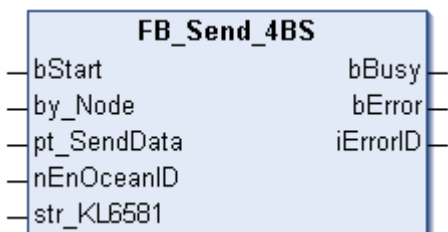
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorID   : E_KL6581_Err;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv. Es können noch keine neuen Daten gesendet werden.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>iErrorID</i> beschrieben.
iErrorID	E_KL6581_Err [▶ 42]	Beschreibt die Art des Fehlers.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.3.2 FB\_Send\_4BS**



Dieser Funktionsbaustein sendet Daten über EnOcean®. Das ORG Field ist fest auf 7 eingestellt.

**Eingänge**

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

Name	Typ	Beschreibung
bStart	BOOL	Positive Flanke sendet die Daten.
by_Node	BYTE	Adresse des KL6583 Moduls, an die das Telegramm gesendet werden soll (gültige Werte: 1...8).

Name	Typ	Beschreibung
pt_SendData	DWORD	Pointer auf die Daten, die gesendet werden sollen. Mit ADR wird die Pointeradresse ermittelt. Der Pointer muss auf eine 4 Byte Variable zeigen.
nEnOceanID	BYTE	Virtuelle EnOcean®-ID. Auf die reale EnOcean®-ID wird ein Wert von 0...127 addiert (gültige Werte: 0...127).
str_KL6581	STR_KL6581 [► 45]	Wird mit der Datenstruktur des Bausteins FB_KL6581() [► 24] verbunden.

**Ausgänge**

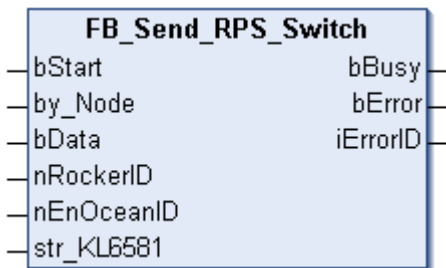
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorID   : E_KL6581_Err;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv. Es können noch keine neuen Daten gesendet werden.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>iErrorID</i> beschrieben.
iErrorID	E_KL6581_Err [► 42]	Beschreibt die Art des Fehlers.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.3.3 FB\_Send\_RPS\_Switch**



Dieser Baustein sendet EnOcean®-Telegramme im Format eines Tasters. Mit der positiven Flanke von *bStart* wird der Wert von *bData* gesendet. Um einen Tastendruck zu simulieren, muss der Baustein üblicherweise 2-mal gestartet werden, einmal mit *bData* = TRUE, und einmal mit *bData* = FALSE. Für eine einfachere Handhabung kann der Baustein *FB\_Send\_RPS\_SwitchAuto()* [► 32] verwendet werden.

**Eingänge**

```
VAR_INPUT
  bStart      : BOOL;
  by_Node     : BYTE;
  bData       : BOOL;
  nRockerID   : INT;
  nEnOceanID  : BYTE;
  str_KL6581  : STR_KL6581;
END_VAR
```

Name	Typ	Beschreibung
bStart	BOOL	Positive Flanke sendet die Daten.
by_Node	BYTE	Adresse des KL6583 Moduls, an die das Telegramm gesendet werden soll (gültige Werte: 1...8).
bData	BOOL	Wert, der übertragen werden soll.

Name	Typ	Beschreibung
nRockerID	INT	Tasternummer, gültige Werte 0..3.
nEnOceanID	BYTE	Virtuelle EnOcean®-ID. Auf die reale EnOcean®-ID wird ein Wert von 0...127 addiert (gültige Werte: 0...127).
str_KL6581	<a href="#">STR_KL6581</a> [ <a href="#">▶ 45</a> ]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [ <a href="#">▶ 24</a> ] verbunden.

 **Ausgänge**

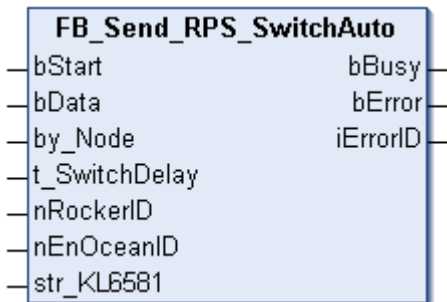
```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  iErrorID   : E_KL6581_Err;
END_VAR
```

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv. Es können noch keine neuen Daten gesendet werden.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>iErrorID</i> beschrieben.
iErrorID	<a href="#">E_KL6581_Err</a> [ <a href="#">▶ 42</a> ]	Beschreibt die Art des Fehlers.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.3.4 FB\_Send\_RPS\_SwitchAuto**



Dieser Baustein sendet EnOcean®-Telegramme im Format eines Tasters. Mit der positiven Flanke von *bStart* wird der Wert von *bData* gesendet. Nach Ablauf der Zeit *t\_SwitchDelay* wird das Signal "Taster loslassen" gesendet.

 **Eingänge**

```
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	Typ	Beschreibung
bStart	BOOL	Positive Flanke sendet die Daten.
bData	BOOL	Wert, der übertragen werden soll.
by_Node	BYTE	Adresse des KL6583 Moduls, an die das Telegramm gesendet werden soll (gültige Werte: 1...8).



Name	Typ	Beschreibung
t_SwitchDelay	TIME	Wie lange der Taster gedrückt werden muss.
nRockerID	INT	Tasternummer, gültige Werte 0..3.
nEnOceanID	BYTE	Virtuelle EnOcean®-ID. Auf die reale EnOcean®-ID wird ein Wert von 0...127 addiert (gültige Werte: 0...127).
str_KL6581	<a href="#">STR_KL6581</a> [ <a href="#">▶ 45</a> ]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [ <a href="#">▶ 24</a> ] verbunden.

**Ausgänge**

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

Name	Typ	Beschreibung
bBusy	BOOL	Der Baustein ist aktiv. Es können noch keine neuen Daten gesendet werden.
bError	BOOL	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>iErrorID</i> beschrieben.
iErrorID	<a href="#">E_KL6581_Err</a> [ <a href="#">▶ 42</a> ]	Beschreibt die Art des Fehlers.

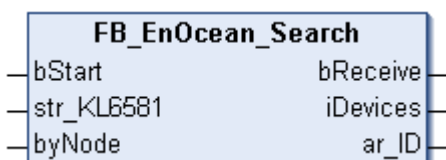
**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.4 Other**

Bausteine	Beschreibung
<a href="#">FB_EnOcean_Search</a> [ <a href="#">▶ 33</a> ]	Baustein erkennt alle EnOcean®-Teilnehmer in seiner Reichweite und zeigt diese an.
<a href="#">FB_Rec_Teach_In</a> [ <a href="#">▶ 34</a> ]	Dieser Baustein zeigt an, wenn in einem EnOcean®-Telegramm das LRN Bit gesetzt ist unabhängig seiner EnOcean®-ID.
<a href="#">FB_Rec_Teach_In_Ex</a> [ <a href="#">▶ 35</a> ]	Dieser Baustein zeigt an, wenn bei einem EnOcean®-Teilnehmer die Learn-Taste gedrückt wird.

**4.1.2.4.1 FB\_EnOcean\_Search**



Dieser Funktionsbaustein zeigt alle EnOcean®-IDs an, die er empfangen hat und trägt diese in ein Empfangsarray ein (*ar\_ID*). Es können bis zu 256 EnOcean®-Teilnehmer erkannt werden. Wahlweise kann der Baustein auch für jede KL6583 einzeln angelegt werden. Damit ist zu erkennen, ob ein EnOcean®-Teilnehmer von mehreren KL6583 empfangen wird.

**Eingänge**

```
VAR_INPUT
  bStart      : BOOL;
  str_KL6581  : STR_KL6581;
  byNode     : BYTE;
END_VAR
```

Name	Typ	Beschreibung
bStart	BOOL	Bei TRUE ist der Baustein aktiviert, bei FALSE deaktiviert.
str_KL6581	STR_KL6581 [ <a href="#">▶ 45</a> ]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [ <a href="#">▶ 24</a> ] verbunden.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.

 **Ausgänge**

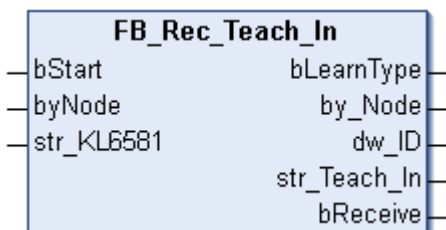
```
VAR_OUTPUT
  bReceive : BOOL := TRUE;
  iDevices : INT;
  ar_ID    : ARRAY [0..255] OF DWORD;
END_VAR
```

Name	Typ	Beschreibung
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.
iDevices	INT	Anzahl an gefundenen EnOcean®-Teilnehmern.
ar_ID	ARRAY OF DWORD	EnOcean®-IDs, die gefunden wurden.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.4.2 FB\_Rec\_Teach\_In**



Dieser Funktionsbaustein zeigt an, wenn bei einem EnOcean®-Teilnehmer die Learn-Taste gedrückt wird. Wenn das Flag *bLearnType* gesetzt ist, können weitere Informationen des EnOcean®-Teilnehmers ausgelesen werden. Dies ist eine Funktion, die das EnOcean®-Gerät liefern muss, die aber bisher von den wenigsten EnOcean®-Geräten unterstützt wird.

 **Eingänge**

```
VAR_INPUT
  bStart    : BOOL;
  byNode    : BYTE;
  str_KL6581 : STR_KL6581;
END_VAR
```

Name	Typ	Beschreibung
bStart	BOOL	Bei TRUE ist der Baustein aktiviert, bei FALSE deaktiviert.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.
str_KL6581	STR_KL6581 [ <a href="#">▶ 45</a> ]	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581()</a> [ <a href="#">▶ 24</a> ] verbunden.

**Ausgänge**

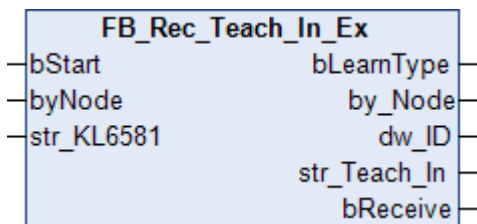
```
VAR_OUTPUT
  bLearnType   : BOOL;
  by_Node      : BYTE;
  dw_ID        : DWORD;
  str_Teach_In : STR_Teach_In;
  bReceive     : BOOL := TRUE;
END_VAR
```

Name	Typ	Beschreibung
bLearnType	BOOL	Ist das Bit gesetzt, finden Sie weitere Daten in der Struktur <i>str_Teach_In</i> .
by_Node	BYTE	Anzahl an gefundenen EnOcean®-Teilnehmern.
dw_ID	DWORD	EnOcean®-ID, bei dem die Learn-Taste gedrückt wurde.
str_Teach_In	<a href="#">STR_Teach_In [▶ 46]</a>	Datenstruktur, Profil, Typ und Hersteller-ID.
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.4.3 FB\_Rec\_Teach\_In\_Ex**



Dieser Funktionsbaustein zeigt an, wenn bei einem EnOcean®-Teilnehmer die Learn-Taste gedrückt wird. Wenn das Flag *bLearnType* gesetzt ist, können weitere Informationen des EnOcean®-Teilnehmers ausgelesen werden. Dies ist eine Funktion, die das EnOcean®-Gerät liefern muss, die aber bisher von den wenigsten EnOcean®-Geräten unterstützt wird.

Zusätzlich zum [FB\\_Rec\\_Teach\\_In\(\) \[▶ 34\]](#)-Funktionsblock wird noch geprüft, ob es sich um ein EEP-Telegramm handelt.

**Eingänge**

```
VAR_INPUT
  bStart       : BOOL;
  byNode       : BYTE;
  str_KL6581   : STR_KL6581;
END_VAR
```

Name	Typ	Beschreibung
bStart	BOOL	Bei TRUE ist der Baustein aktiviert, bei FALSE deaktiviert.
byNode	BYTE	Filter - bei dem Wert Null werden die EnOcean®-Telegramme von allen KL6583 empfangen. Wird ein Wert von 1 bis 8 eingetragen, werden nur die Daten der entsprechenden KL6583 empfangen.
str_KL6581	<a href="#">STR_KL6581 [▶ 45]</a>	Wird mit der Datenstruktur des Bausteins <a href="#">FB_KL6581() [▶ 24]</a> verbunden.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.32	Tc2_EnOcean ab v3.4.6.0

 **Ausgänge**

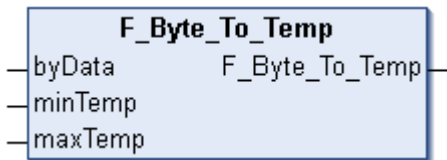
```
VAR_OUTPUT
  bLearnType   : BOOL;
  by_Node      : BYTE;
  dw_ID        : DWORD;
  str_Teach_In : STR_Teach;
  bReceive     : BOOL := TRUE;
END_VAR
```

Name	Typ	Beschreibung
bLearnType	BOOL	Ist das Bit gesetzt, finden Sie weitere Daten in der Struktur <i>str_Teach_In</i> .
by_Node	BYTE	Anzahl an gefundenen EnOcean®-Teilnehmern.
dw_ID	DWORD	EnOcean®-ID, bei dem die Learn-Taste gedrückt wurde.
str_Teach_In	<a href="#">STR_Teach [► 45]</a>	Datenstruktur, Profil, Typ und Hersteller-ID.
bReceive	BOOL	Bei empfangenden EnOcean®-Telegramm wird dieser Wert für einen Zyklus auf FALSE gesetzt.

**4.1.2.5 Function**

Bausteine	Beschreibung
<a href="#">F_Byte_To_Temp [► 36]</a>	Diese Funktion wandelt einen Byte-Rohwert in eine REAL-Variable um.
<a href="#">F_Byte_To_TurnSwitch [► 37]</a>	Diese Funktion wandelt einen Byte Rohwert in ein Bool Array um.

**4.1.2.5.1 F\_Byte\_To\_Temp**



Diese Funktion wandelt einen Byte-Rohwert in eine REAL-Variable um.

Bei EnOcean® werden Temperaturdaten in einem bestimmten Format übertragen, das ein Byte groß ist. Diese Daten sind meist auf einen bestimmten Temperaturwert skaliert.

Zum Beispiel wird ein Wert aus einem Wertebereich von 0...40°C übertragen. Der Funktion wird der minimale und maximale Wert der Daten und der Rohwert übergeben. Der Ausgang der Funktion gibt dann die Temperatur als REAL-Variable aus.

 **Eingänge**

```
VAR_INPUT
  byData   : BYTE;
  minTemp  : REAL := 0;
  maxTemp  : REAL := 40;
END_VAR
```

Name	Typ	Beschreibung
byData	BYTE	Rohdaten
minTemp	REAL	Minimale Temperatur
maxTemp	REAL	Maximale Temperatur

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.5.2 F\_Byte\_To\_TurnSwitch**

**F\_Byte\_To\_TurnSwitch**

byData F\_Byte\_To\_TurnSwitch

Diese Funktion wandelt einen Byte-Rohwert in ein Bool-Array um, das als Datenstruktur vorliegt (siehe STREnOceanTurnSwitch [▶ 46]).

 **Eingänge**

```
VAR_INPUT
  byData : BYTE;
END_VAR
```

Name	Typ	Beschreibung
byData	BYTE	Rohdaten

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.1.2.6 Fehlercodes**

Wert (hex)	Wert (dez)	Wert (enum)	Beschreibung
0x0000	0	NO_ERROR	Am Baustein liegt kein Fehler an.
0x000A	10	KL6581_WrongTerminal	Falsche Klemme angeschlossen.
0x0010	16	KL6581_WatchdogError	Zeitüberschreitung beim Initialisierungsvorgang des Bausteins FB_KL6581() [▶ 24].
0x0011	17	KL6581_NoComWithKL6581	Üblicherweise gibt es bei dieser Meldung keine Verbindung zur Klemme. Klemme im System Manager mit den Variablen verknüpft? Klemme falsch gesteckt? Alles bereinigen, alles Übersetzen und im System Manager neu eingelesen?
0x0012	18	KL6581_idx_number_not_OK	Die Eingangsvariable nIdx des Bausteins FB_KL6581() ist größer als 64.
0x0013	19	KL6581_Switch_to_Stopp	Die Klemme ist aus dem Datenaustausch mit der EnOcean®-Sender und -Empfänger KL6583-0000 gegangen, es wurden keine EnOcean®-Daten gesendet oder empfangen.
0x0014	20	KL6581_not_ready	Interne Meldung für die Funktionsblöcke, die an den FB_KL6581() angeschlossen sind.
0x0015	21	KL6581_No_KL6853_Found	Es ist keine KL6583 an der EnOcean®-Masterklemme KL6581 angeschlossen oder die Kommunikation ist nicht vorhanden!

Wert (hex)	Wert (dez)	Wert (enum)	Beschreibung
0x0016	22	KL6581_TransmissionError	Daten konnten nicht gesendet werden, Adresse der KL6583 prüfen oder KL6583 nicht betriebsbereit.

## 4.2 DUTs

### KL6021-0023/Hardware Types

Datentypen	Beschreibung
<a href="#">ST_EnOceanInData [► 40]</a>	Prozessabbild der Eingänge der KL6021-0023
<a href="#">ST_EnOceanOutData [► 40]</a>	Prozessabbild der Ausgänge der KL6021-0023

### KL6021-0023

Datentypen	Beschreibung
<a href="#">E_EnOceanRotarySwitch [► 39]</a>	Stellung des Drehschalters am Raumbediengerät
<a href="#">E_EnOceanSensorType [► 39]</a>	Sensorentyp
<a href="#">ST_EnOceanReceivedData [► 40]</a>	Interne Struktur

### KL6581/Hardware Types

Datentypen	Beschreibung
<a href="#">KL6581_Input [► 42]</a>	Prozessabbild der Eingänge der KL6581
<a href="#">KL6581_Output [► 43]</a>	Prozessabbild der Ausgänge der KL6581

### KL6581

Datentypen	Beschreibung
<a href="#">AR_EnOceanWindow [► 44]</a>	Zustand des Fensters
<a href="#">E_ENOCEAN_Org [► 41]</a>	Typ des EnOcean®-Telegramms
<a href="#">E_KL6581_Err [► 42]</a>	Fehlermeldungen
<a href="#">STR_EnOceanSwitch [► 44]</a>	Zustand der Taster
<a href="#">STR_KL6581 [► 45]</a>	Interne Struktur
<a href="#">STR_Teach [► 45]</a>	Datenstruktur Hersteller-ID, Typ und Funktion
<a href="#">STR_Teach_In [► 46]</a>	Datenstruktur Hersteller-ID, Typ und Profil
<a href="#">STREnOceanTurnSwitch [► 46]</a>	Stellung des Drehschalters am Raumbediengerät.

### 4.2.1 KL6021-0023

#### Hardware Types

Datentypen	Beschreibung
<a href="#">ST_EnOceanInData [► 40]</a>	Prozessabbild der Eingänge der KL6021-0023
<a href="#">ST_EnOceanOutData [► 40]</a>	Prozessabbild der Ausgänge der KL6021-0023

Datentypen	Beschreibung
<a href="#">E_EnOceanRotarySwitch [► 39]</a>	Stellung des Drehschalters am Raumbediengerät
<a href="#">E_EnOceanSensorType [► 39]</a>	Sensorentyp
<a href="#">ST_EnOceanReceivedData [► 40]</a>	Interne Struktur

## 4.2.1.1 Enums

### 4.2.1.1.1 E\_EnOceanSensorType

Sensorentyp.

```

TYPE E_EnOceanSensorType :
(
  eEnOceanSensorTypePTM      := 5,
  eEnOceanSensorTypeSTM1Byte := 6,
  eEnOceanSensorTypeSTM4Byte := 7,
  eEnOceanSensorTypeCTM      := 8
)
END_TYPE
    
```

Name	Beschreibung
eEnOceanSensorTypePTM	PTM
eEnOceanSensorTypeSTM1Byte	STM 1 Byte
eEnOceanSensorTypeSTM4Byte	STM 4 Byte
eEnOceanSensorTypeCTM	CTM

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

### 4.2.1.1.2 E\_EnOceanRotarySwitch

*E\_EnOceanRotarySwitch* beschreibt die Stellung des Drehschalters am Raumbediengerät.

```

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

Name	Beschreibung
eEnOceanRotarySwitchStep0	Schalter in Stellung „0“
eEnOceanRotarySwitchStep1	Schalter in Stellung „1“
eEnOceanRotarySwitchStep2	Schalter in Stellung „2“
eEnOceanRotarySwitchStep3	Schalter in Stellung „3“
eEnOceanRotarySwitchAuto	Schalter in Stellung „Auto“

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

## 4.2.1.2 Structures

### 4.2.1.2.1 ST\_EnOceanReceivedData

Interne Struktur.

Über diese Struktur wird der Baustein [FB\\_EnOceanReceive\(\)](#) [[▶ 14](#)] mit den Empfangsbausteinen verbunden.

```

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	Typ	Beschreibung
bReceived	BOOL	Daten empfangen
nLength	BYTE	Länge
eEnOceanSensorType	<a href="#">E_EnOceanSensorType</a> <a href="#">▶ 39</a>	Sensortyp
nData	ARRAY OF BYTE	Datenbytes
nStatus	BYTE	Status
nTransmitterId	UDINT	Transmitter-ID

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

### 4.2.1.2.2 ST\_EnOceanInData

Prozessabbild der Eingänge der KL6021-0023.

Wird im System Manager mit der Klemmen verknüpft.

```

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

Name	Typ	Beschreibung
nStatus	BYTE	Status Byte
nData	ARRAY OF BYTE	11 Bytes für die Eingangsdaten

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

### 4.2.1.2.3 ST\_EnOceanOutData

Prozessabbild der Ausgänge der KL6021-0023.

Wird im System Manager mit der Klemmen verknüpft.

```

TYPE ST_EnOceanOutData :
STRUCT
  nCtrl : BYTE;
    
```



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

Name	Typ	Beschreibung
nCtrl	BOOL	Control Byte.
nData	ARRAY OF BYTE	11 Bytes für die Ausgangsdaten

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

## 4.2.2 KL6581

**Hardware Types**

Datentypen	Beschreibung
<a href="#">KL6581_Input [► 42]</a>	Prozessabbild der Eingänge der KL6581
<a href="#">KL6581_Output [► 43]</a>	Prozessabbild der Ausgänge der KL6581
Datentypen	Beschreibung
<a href="#">AR_EnOceanWindow [► 44]</a>	Zustand des Fensters
<a href="#">E_ENOCEAN_Org [► 41]</a>	Typ des EnOcean®-Telegramms
<a href="#">E_KL6581_Err [► 42]</a>	Fehlermeldungen
<a href="#">STR_EnOceanSwitch [► 44]</a>	Zustand der Taster
<a href="#">STR_KL6581 [► 45]</a>	Interne Struktur
<a href="#">STR_Teach [► 45]</a>	Datenstruktur Hersteller-ID, Typ und Funktion
<a href="#">STR_Teach_In [► 46]</a>	Datenstruktur Hersteller-ID, Typ und Profil
<a href="#">STREnOceanTurnSwitch [► 46]</a>	Stellung des Drehschalters am Raumbediengerät.

### 4.2.2.1 Enums

#### 4.2.2.1.1 E\_ENOCEAN\_ORG

Typ des EnOcean®-Telegramms.

```
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	Beschreibung
PTM_TELEGRAM	PTM-Telegramm
STM_1BYTE_TELEGRAM	1 Byte Telegramm
STM_4BYTE_TELEGRAM	4 Byte Telegramm
CTM_TELEGRAM	CTM-Telegramm
MODEM_TELEGRAM	Modem Telegramm
MODEM_ACK_TELEGRAM	Modem Telegramm mit Bestätigung.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.2.2.1.2 E\_KL6581\_Err**

Fehlermeldungen.

```

TYPE E_KL6581_Err :
(
  NO_ERROR           := 16#0,
  KL6581_WrongTerminal := 16#A,
  KL6581_WatchdogError := 16#10,
  KL6581_NoComWithKL6581 := 16#11,
  KL6581_idx_number_not_OK := 16#12,
  KL6581_Switch_to_Stopp := 16#13,
  KL6581_not_ready := 16#14,
  KL6581_No_KL6853_Found := 16#15,
  KL6581_TransmissionError := 16#16
)
END_TYPE
    
```

Name	Beschreibung
NO_ERROR	Am Baustein liegt kein Fehler an.
KL6581_WrongTerminal	Falsche Klemme angeschlossen.
KL6581_WatchdogError	Zeitüberschreitung beim Initialisierungsvorgang des Bausteins "FB_KL6581".
KL6581_NoComWithKL6581	Üblicherweise gibt es bei dieser Meldung keine Verbindung zur Klemme. Klemme im System Manager mit den Variablen verknüpft? Klemme falsch gesteckt? Alles bereinigen, alles Übersetzen und im System Manager neu eingelesen?
KL6581_idx_number_not_OK	Die Eingangsvariable <i>ndx</i> des Bausteins <code>FB_KL6581()</code> [► 24] ist größer als 64.
KL6581_Switch_to_Stopp	Die Klemme ist aus dem Datenaustausch mit der KL6583 gegangen. Es sind keine EnOcean®-Daten gesendet oder empfangen worden.
KL6581_not_ready:	Interne Meldung für die Funktionsblöcke, die an den <code>FB_KL6581()</code> angeschlossen sind.
KL6581_No_KL6853_Found	Es ist keine KL6583 an der KL6581 angeschlossen oder die Kommunikation ist nicht vorhanden!
KL6581_TransmissionError	Daten konnten nicht gesendet werden, Adresse der KL6583 prüfen oder KL6583 nicht betriebsbereit.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.2.2.2 Structures**

**4.2.2.2.1 KL6581\_Input**

Prozessabbild der Eingänge der KL6581.

Wird im System Manager mit der Klemme verknüpft.

```

TYPE KL6581_Input :
STRUCT
  nStatus : BYTE;
  CNODE   : BYTE;
  ORG     : BYTE;
  DB0     : BYTE;
  DB1     : BYTE;
  DB2     : BYTE;
  DB3     : BYTE;
  ID0     : BYTE;
    
```

```

ID1      : BYTE;
ID2      : BYTE;
ID3      : BYTE;
STATUS   : BYTE;
END_STRUCT
END_TYPE
    
```

Name	Beschreibung
nStatus	Status Byte
CNODE	Daten Byte
ORG	Daten Byte
DB0	Daten Byte
DB1	Daten Byte
DB2	Daten Byte
DB3	Daten Byte
ID0	Daten Byte
ID1	Daten Byte
ID2	Daten Byte
ID3	Daten Byte
STATUS	Daten Byte

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.2.2.2 KL6581\_Output**

Prozessabbild der Ausgänge der KL6581.

Wird im System Manager mit der Klemme verknüpft.

```

TYPE KL6581_Output :
STRUCT
  nControl : BYTE;
  CNODE    : BYTE;
  ORG      : BYTE;
  DB0      : BYTE;
  DB1      : BYTE;
  DB2      : BYTE;
  DB3      : BYTE;
  ID0      : BYTE;
  ID1      : BYTE;
  ID2      : BYTE;
  ID3      : BYTE;
  STATUS   : BYTE;
END_STRUCT
END_TYPE
    
```

Name	Beschreibung
nControl	Control Byte
CNODE	Daten Byte
ORG	Daten Byte
DB0	Daten Byte
DB1	Daten Byte
DB2	Daten Byte
DB3	Daten Byte
ID0	Daten Byte
ID1	Daten Byte
ID2	Daten Byte
ID3	Daten Byte

Name	Beschreibung
STATUS	Daten Byte

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.2.2.2.3 AR\_EnOceanWindow**

Diese Struktur zeigt den Zustand des Fensters an.

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

Name	Beschreibung
bUp	Das Fenster ist gekippt.
bOpen	Das Fenster ist offen.
bClose	Das Fenster ist geschlossen.

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

**4.2.2.2.4 STR\_EnOceanSwitch**

Zustand der Taster.

```
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;
END_STRUCT
END_TYPE
```

Name	Beschreibung
bT1_ON	Taster 1 an
bT1_OFF	Taster 1 aus
bT2_ON	Taster 2 an
bT2_OFF	Taster 2 aus
bT3_ON	Taster 3 an
bT3_OFF	Taster 3 aus
bT4_ON	Taster 4 an
bT4_OFF	Taster 4 aus

**Voraussetzungen**

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

### 4.2.2.2.5 STR\_KL6581

Interne Struktur.

Über diese Struktur wird der Baustein FB\_KL6581() mit den Sende-Empfangsbausteinen verbunden.

```

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	Beschreibung
by_Status	Status
by_Node	Node Nummer der KL6583, die das EnOcean®-Telegramm empfangen hat.
by_ORG	Typ des EnOcean®-Telegramms
ar_DB	Daten Bytes
_Dummy	Platzhalter, ohne weitere Bedeutung.
dw_ID	Transmitter-ID
ptData	Pointer
iErrorId	Beschreibt die Art des Fehlers (siehe <a href="#">E_KL6581_Err</a> [▶ 42]).
by_STATE	State
bError	Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable <i>iErrorId</i> beschrieben.
idx	Index

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

### 4.2.2.2.6 STR\_Teach

Datenstruktur Hersteller-ID, Typ und Funktion.

```

STRUCT
  nManufacturerID : WORD;
  nTYPE           : BYTE;
  nFunc           : BYTE;
END_STRUCT
END_TYPE
    
```

Name	Beschreibung
nManufacturerID	Hersteller-ID
nTYPE	Typ
nFunc	Funktion

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.32	Tc2_EnOcean ab v3.4.6.0

### 4.2.2.2.7 STR\_Teach\_In

Datenstruktur Hersteller-ID, Typ und Profil.

```

TYPE STR_Teach_In :
STRUCT
  nManufacturerID : WORD;
  nTYPE           : BYTE;
  nProfile        : BYTE;
END_STRUCT
END_TYPE
    
```

Name	Beschreibung
nManufacturerID	Hersteller-ID
nTYPE	Typ
nProfile	Profil

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

### 4.2.2.2.8 STREnOceanTurnSwitch

*STREnOceanTurnSwitch* beschreibt die Stellung des Drehschalters am Raumbediengerät.

```

TYPE STREnOceanTurnSwitch :
STRUCT
  bStageAuto : BOOL;
  bStage_0   : BOOL;
  bStage_1   : BOOL;
  bStage_2   : BOOL;
  bStage_3   : BOOL;
END_STRUCT
END_TYPE
    
```

Name	Beschreibung
bStageAuto	Schalter in Stellung „Auto“
bStage_0	Schalter in Stellung „0“
bStage_1	Schalter in Stellung „1“
bStage_2	Schalter in Stellung „2“
bStage_3	Schalter in Stellung „3“

#### Voraussetzungen

Entwicklungsumgebung	Einzubindende SPS-Bibliothek
TwinCAT ab v3.1.4020.14	Tc2_EnOcean ab v3.3.5.0

## 4.3 Integration in TwinCAT

### 4.3.1 KL6581 mit CX5120

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für EnOcean® in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es sollen vier Tastersignale eines EnOcean®-Funkschaltmoduls empfangen werden.

Beispiel: [https://infosys.beckhoff.com/content/1031/tcplclib\\_tc2\\_enocean/Resources/6200373771.zip](https://infosys.beckhoff.com/content/1031/tcplclib_tc2_enocean/Resources/6200373771.zip)



Das TwinCAT-Projekt steht als \*.zip-Datei zum Download zur Verfügung. Diese muss zuerst lokal entpackt werden, damit das Archiv (\*.tnzip-Datei) zum Import in das TwinCAT-Projekt zur Verfügung steht.

**Hardware**

**Einrichtung der Komponenten**

Es wird folgende Hardware benötigt:

- 1x Embedded-PC CX5120
- 1x EnOcean®-Masterklemme KL6581
- 1x EnOcean®-Sender und –Empfänger KL6583-0000
- 1x Endklemme KL9010

Richten Sie die Hardware sowie die EnOcean®-Komponenten wie in den entsprechenden Dokumentationen beschrieben ein.

Dieses Beispiel geht davon aus, dass die Id vom Funkschaltmodul bekannt ist.

**Software**

**Erstellung des SPS-Programms**

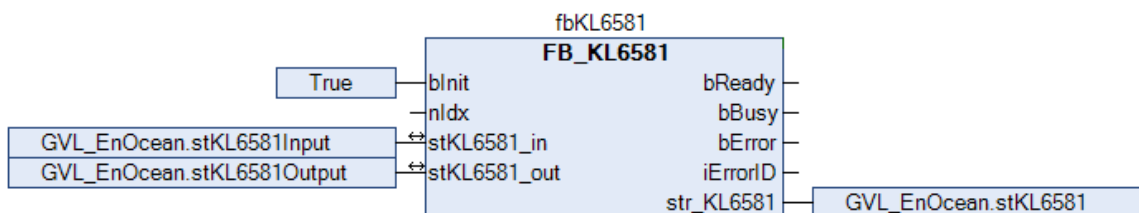
Erstellen Sie ein neues „TwinCAT XAE Project“ und legen Sie ein „Standard PLC Project“ an. Fügen Sie im SPS-Projekt unter „References“ die Bibliothek Tc2\_EnOcean hinzu. Erzeugen Sie eine globale Variablenliste mit den Namen GVL\_EnOcean und legen Sie die folgenden Variablen an:

```
VAR_GLOBAL
    stKL6581Input      AT %I* : KL6581_Input;
    stKL6581Output    AT %Q* : KL6581_Output;
    stKL6581          : STR_KL6581;
END_VAR
```

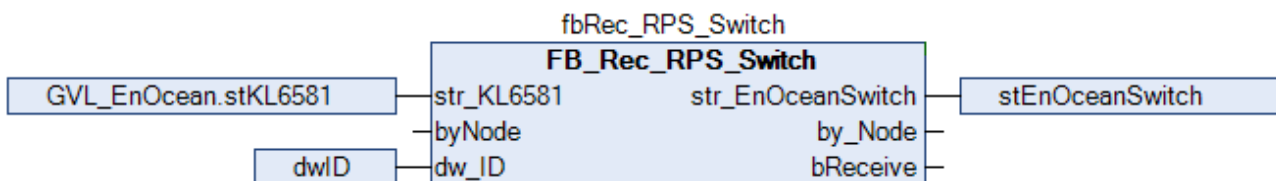
Name	Typ	Beschreibung
stKL6581Input	KL6581_Input [▶ 42]	Eingangsvariable für die EnOcean®-Klemme.
stKL6581Output	KL6581_Output [▶ 43]	Ausgangsvariable für die EnOcean®-Klemme.
stKL6581	STR_KL6581 [▶ 45]	Wird für die Kommunikation mit EnOcean® benötigt.

Alle EnOcean®-Bausteine müssen in derselben Task aufgerufen werden.

Legen Sie ein MAIN-Programm (CFC) an, in dem die Bausteine FB\_KL6581 [▶ 24] und FB\_Rec\_RPS\_Switch [▶ 27] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, ihn mit den Strukturen *stKL6581Input*, *stKL6581Output* und *stKL6581* zu verknüpfen.

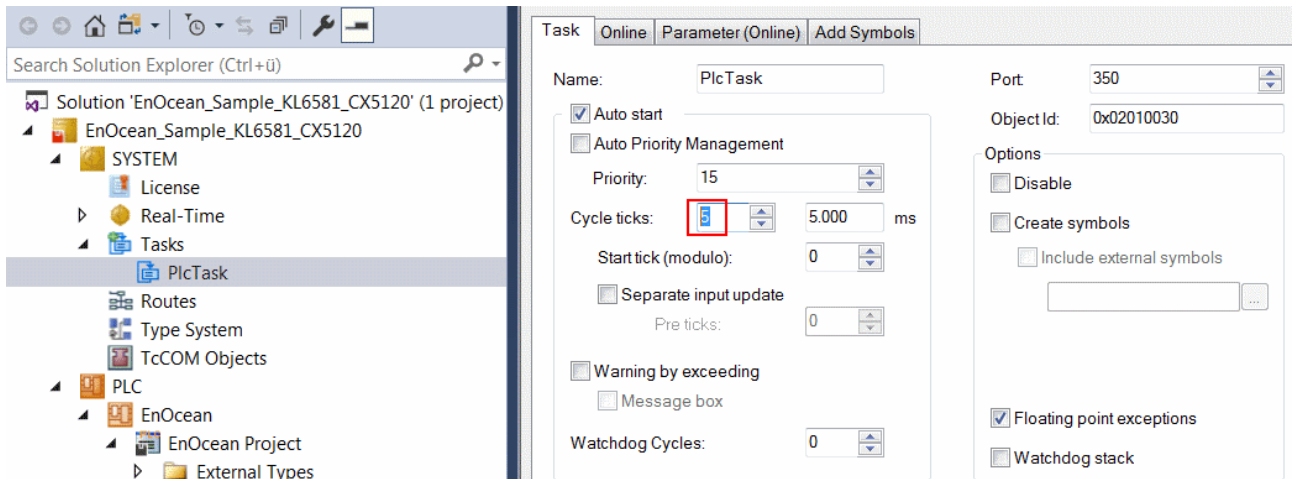


Der Eingang *dw\_ID* des Empfangsbausteins wird mit der lokalen Variable *dwId* (ID vom Funkschaltmodul) verknüpft und *str\_KL6581* mit der globalen Variable *stKL6581*.



Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit.

Weitere Bedingungen finden Sie in der Beschreibung des Bausteins FB\_KL6581 [▶ 24].



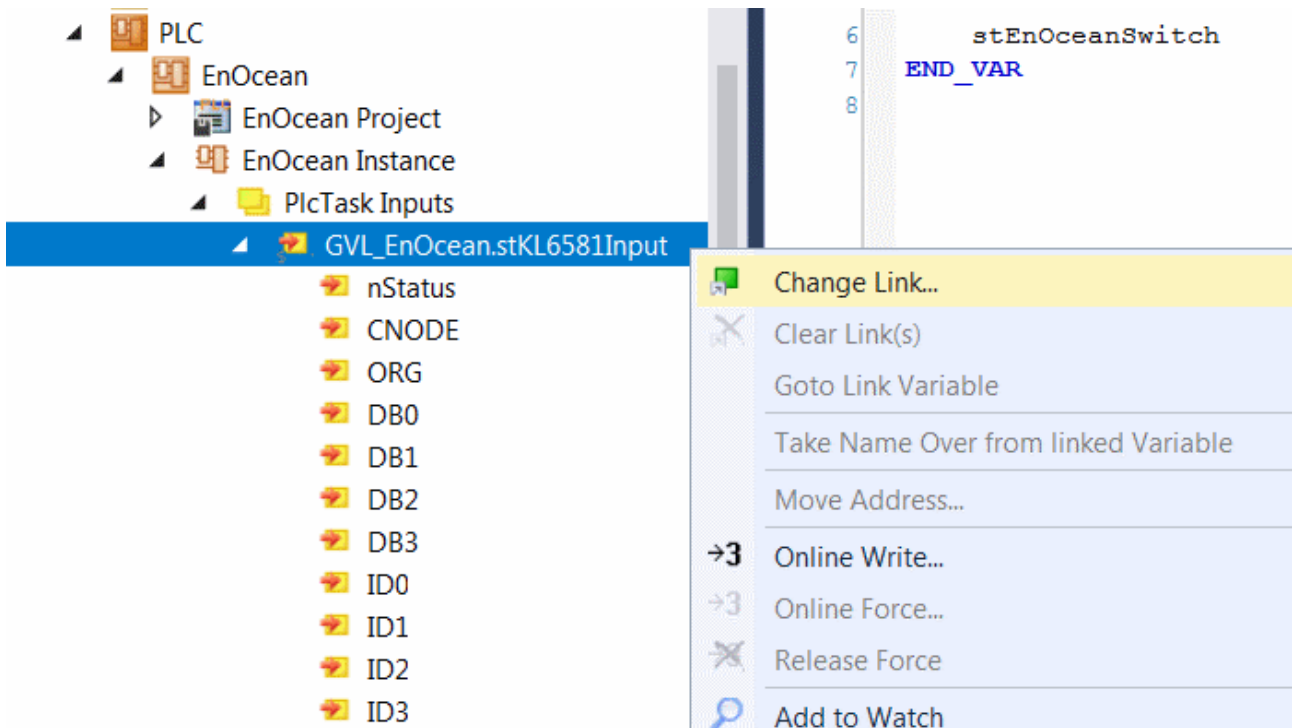
**E/A Konfiguration**

Wählen Sie als Zielsystem den CX und lassen Sie nach dessen Hardware suchen. Im Bereich der SPS, in der Instanz des Projekts sehen Sie, dass die Ein- und Ausgangsvariablen der Task zugeordnet sind.

Verknüpfen Sie die globalen Variablen nun mit den Ein- und Ausgängen der Busklemmen.

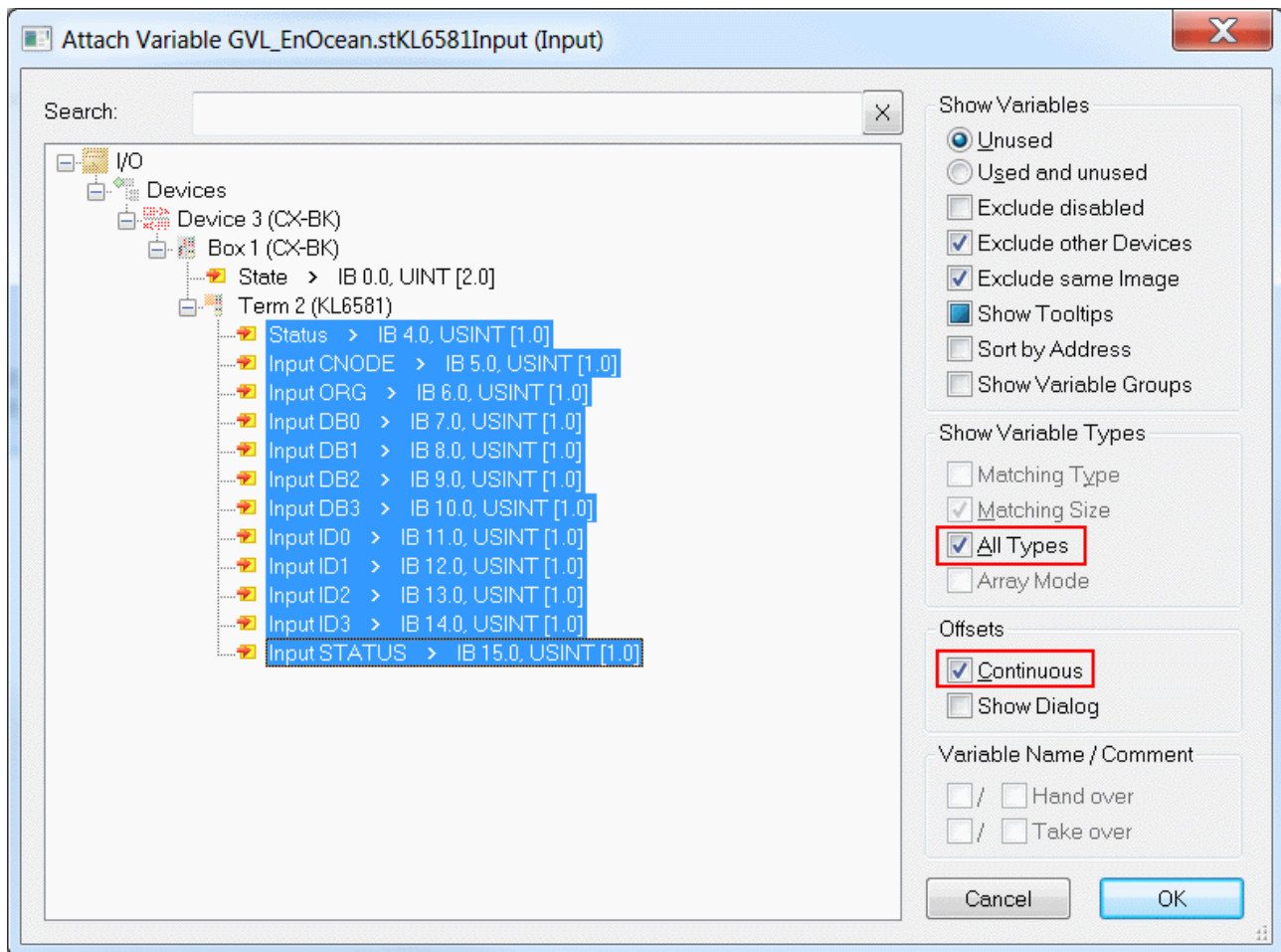
Im Folgenden ist die Verknüpfung der EnOcean®-Variablen ausführlich beschrieben.

Die Struktur *stKL6581Input* mit der rechten Maustaste anklicken und „Verknüpfung ändern“ auswählen.



In der „E/A Konfiguration“ die Klemme auswählen, „Alle Typen“ und „Kontinuierlich“ wählen und von „Status“ bis „InputStatus“ mit der linken Maustaste und >SHIFT< Taste markieren. Anschließend „OK“ drücken.





Sie können die Verknüpfung jetzt kontrollieren. Gehen Sie dazu auf die KL6581 und öffnen Sie diese. Alle Daten der Klemme müssen jetzt mit einem Pfeil markiert sein. Ist dies der Fall, fahren Sie genauso mit den Ausgängen fort.

#### Sehen Sie dazu auch

- 📄 KL6581\_Input [▶ 42]
- 📄 KL6581\_Output [▶ 43]
- 📄 STR\_KL6581 [▶ 45]

## 5 Anhang

### 5.1 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

#### Downloadfinder

Unser [Downloadfinder](#) beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

#### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: [www.beckhoff.com](http://www.beckhoff.com)

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

#### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157  
E-Mail: [support@beckhoff.com](mailto:support@beckhoff.com)

#### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460  
E-Mail: [service@beckhoff.com](mailto:service@beckhoff.com)

#### Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49 5246 963-0  
E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)

Internet: [www.beckhoff.com](http://www.beckhoff.com)

## **Trademark statements**

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

Mehr Informationen:  
**[www.beckhoff.com/te1000](http://www.beckhoff.com/te1000)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

