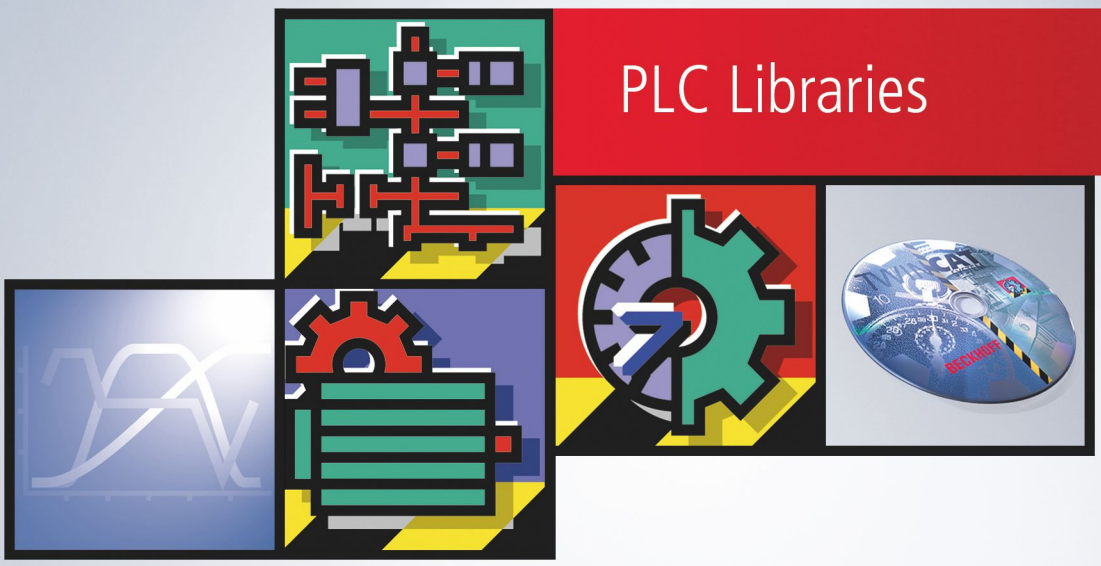
BECKHOFF New Automation Technology

Handbuch | DE

TX1200

TwinCAT 2 | PLC-Bibliothek: TcEIB



13.09.2023 | Version: 1.3

Inhaltsverzeichnis

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, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche 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 weiter entwickelt.
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®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® 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 Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

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

Copyright

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 EIB-Bibliothek ist eine TwinCAT SPS Bibliothek zum Datenaustausch mit EIB Geräten.

Alle Funktionsblöcke aus der Bibliothek müssen in derselben Task aufgerufen werden.

Diese Bibliothek ist nur in Verbindung mit einer KL6301 (EIB-Masterklemme) einzusetzen.

3 Zielgruppen

Für den Nutzer dieser Bibliothek werden folgende Grundkenntnisse vorausgesetzt:

- TwinCAT PLC Control
- TwinCAT System Manager
- PC und Netzwerkkenntnisse
- Aufbau und Eigenschaften der Beckhoff Embedded-PC und deren Busklemmensystem
- Technologie von EIB 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.

4 Arbeitsweise der KL6301

Für das Arbeiten mit der EIB-Busklemme sind Funktionsbausteine notwendig, die in dieser Dokumentation beschrieben werden.

Es stehen ab der Firmware-Version B1 und der Bibliotheksversion V3.000.000 drei Verschiedene Modi zur Verfügung, die in dem KL6301() [▶ 26] Funktionsbaustein aktiviert werden können.

Mode 0: 4 Filter mit jeweils 64 Gruppeneinträgen (kompatibel zur Firmware B0)

Mode 1: 8 Filter mit jeweils 32 Gruppeneinträgen

Mode 2: 8 Filter mit jeweils 32 Gruppeneinträgen invertiert

Mode 100: Monitor Funktion (alle Gruppenadresstelegramme können empfangen werden, die KL6301 sendet kein ACK). In diesem Modus kann nicht gesendet werden.

Senden

Die KL6301 verschickt Daten einzeln. Das bedeutet, eine zur KL6301 übertragene Daten-Variable wird von der Klemme einzeln ins EIB-Netzwerk gesendet. Erst wenn diese erfolgreich verschickt wurde, können Sie die nächsten EIB-Daten zur KL6301 übertragen. Es können 2 Typen von EIB-Telegrammen versendet werden:

- WRITE_GROUP zum Schreiben von Daten auf andere EIB-Teilnehmer
- READ_GROUP_REQ zum Anfordern von Daten von anderen EIB-Teilnehmern

Empfangen

Die KL6301 besitzt maximal 8 Filteradressen. Diese Filter filtern die EIB-Gruppenadressen. Nur EIB-Telegramme, die in den Filter eingetragen sind, werden im Prozessabbild sichtbar und mit einem ACK beantwortet.

Ein Filter kann bis zu 64 Gruppenadressen beinhalten. Bei 4 Filtern in Summe mal 64 Einträge ergibt das 256 Gruppenadressen die Daten annehmen. Bei 8 Filtern in Summe mal 32 Einträge ergibt das 256 Gruppenadressen die Daten annehmen. Die Konfigurierung erfolgt über einen Funktionsbaustein. Bei der Initialisierung der Busklemme werden die Gruppenadressen geladen und sind sofort aktiv.

Es muss mindestens ein Filter parametriert werden. Die Art der Daten hat bei der Filtereinstellung keine Bedeutung.

Monitor Funktion

Ist der Mode 100 aktiviert, dürfen keine Filter eingestellt werden. Die Filter EIB_GROUP_FILTER werden einfach leer gelassen und nicht beschrieben.

5 Integration in TwinCAT

5.1 KL6301 - Verknüpfung mit dem TwinCAT System Manager

Wie verknüpfe ich die KL6301 mit dem System Manager?

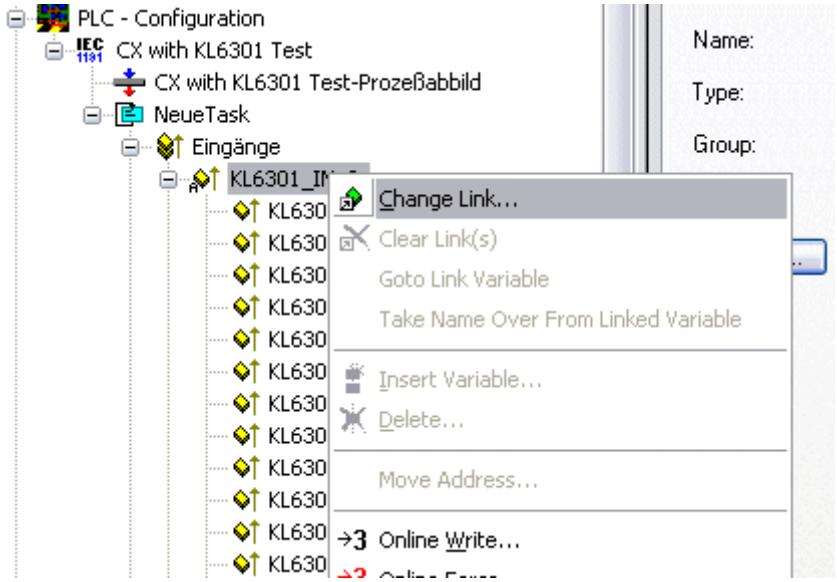


Bild 1

Wählen Sie "All Types" und "Continuous" an (siehe Bild 2).

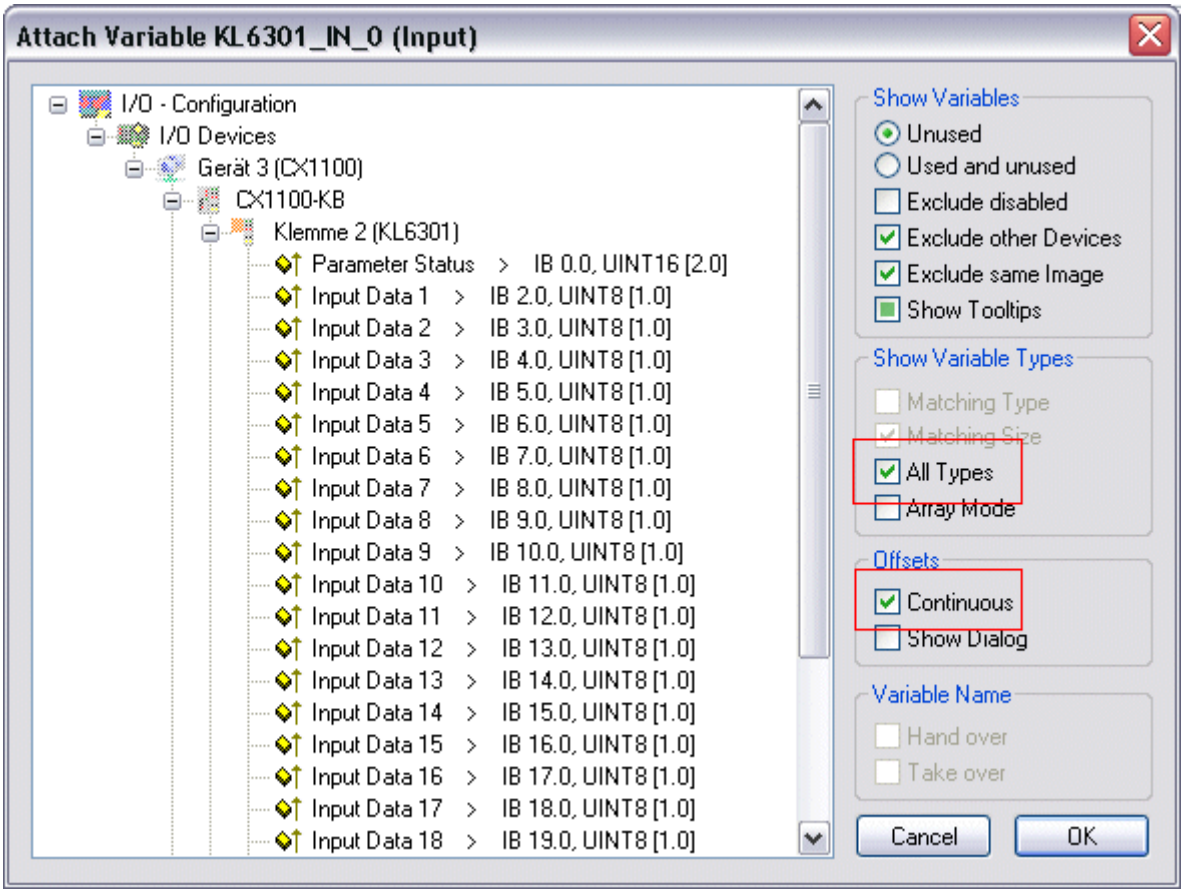


Bild 2

Klicken Sie mit der Maus auf die erste Variable der KL6301 (Parameter Status). Drücken Sie dann die >SHIFT< Taste und halten Sie diese gedrückt. Gehen Sie mir dem Mauszeiger auf die letzte Variable der KL6301 (Input Data 22) und klicken Sie wiederum die linke Maustaste. Jetzt lassen Sie die >SHIFT< Taste wieder los. Alle Daten der Klemme müssen jetzt markiert sein (siehe Bild 3). Anschließend den Button "OK" drücken.

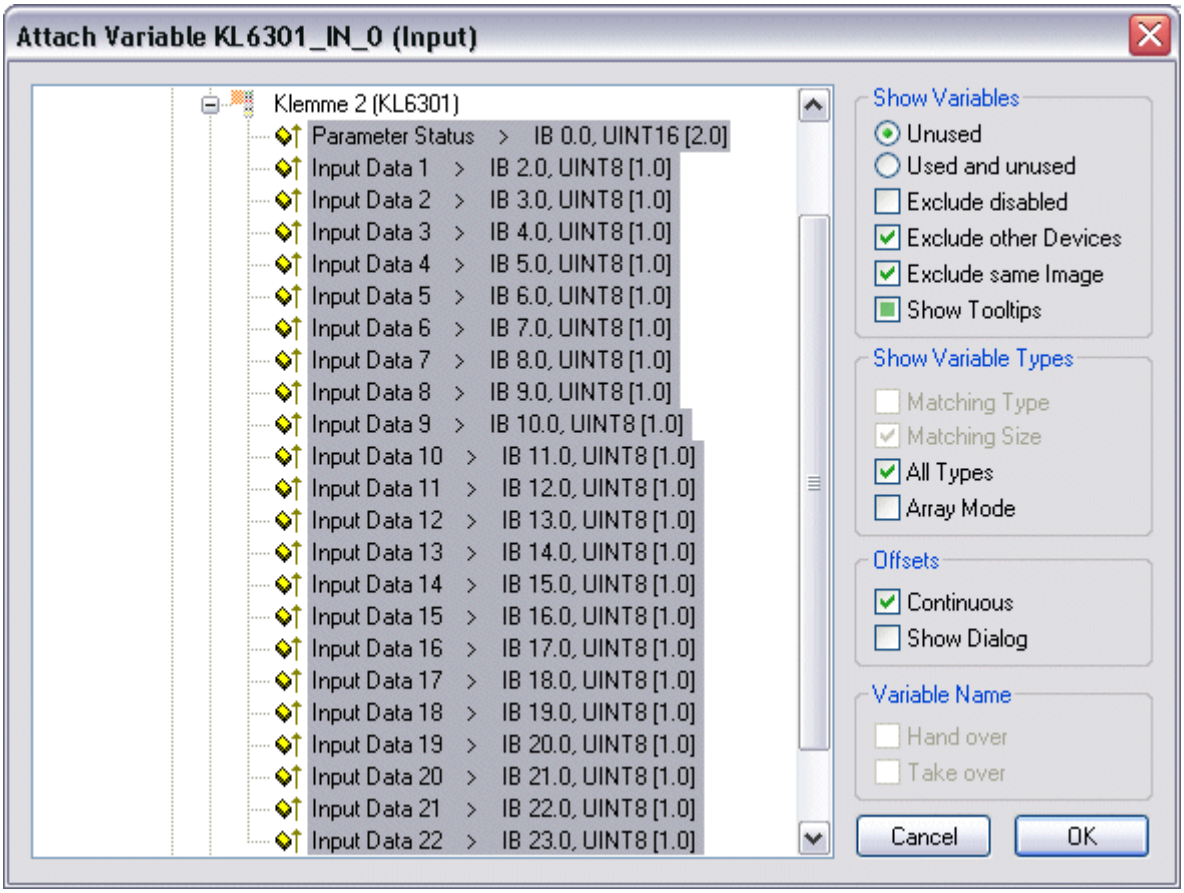


Bild 3

Sie können die Verknüpfung jetzt kontrollieren. Gehen Sie dazu auf die KL6301 und öffnen Sie diese. Alle Daten der Klemme müssen jetzt mit einem kleinen Pfeil markiert sein (siehe Bild 4). Ist dies der Fall fahren Sie genauso mit den Ausgängen fort.

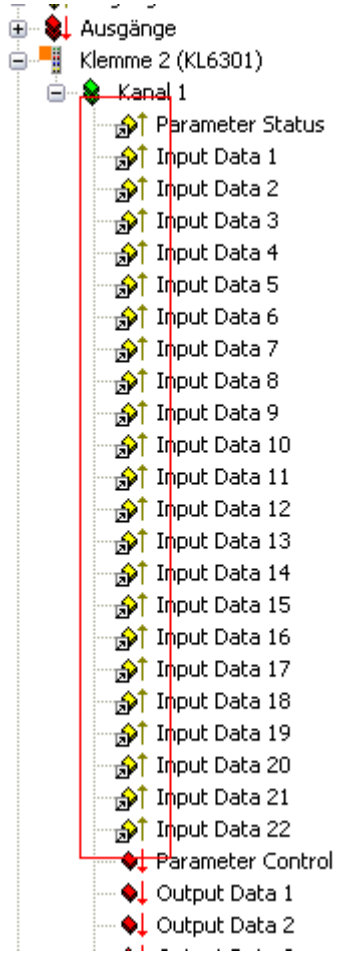


Bild 4

5.2 Integration in TwinCAT (CX9020)

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für EIB in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es soll der Zustand eines Schaltausgangs per Taster verändert werden.

https://infosys.beckhoff.com/content/1031/tcplclibeib/Resources/11993060235.zip https://infosys.beckhoff.com/content/1031/tcplclibeib/Resources/11993060235.zip

Hardware

Einrichtung der Komponenten

Es wird folgende Hardware benötigt:

- 1x Embedded-PC CX9020
- 1x Digitale 2-Kanal-Eingangsklemme KL1002 (für die Setz und Reset Funktion)
- 1x EIB-Klemme KL6301
- 1x Endklemme KL9010

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

Dieses Beispiel geht davon aus, dass ein Setz-Taster auf den ersten und ein Reset-Taster auf den zweiten Eingang der KL1002 gelegt wurde und die EIB Gruppenadresse vom Schaltausgang bekannt ist.

Software

Erstellung des SPS-Programms

Erstellen Sie ein neues SPS-Projekt für PC-basierte Systeme (ARM) und fügen die Bibliothek *TcEIB.lib* hinzu.

Erzeugen Sie als Nächstes die folgenden globalen Variablen:

```
VAR_GLOBAL
    bSet        AT %I*     : BOOL;
    bReset      AT %I*     : BOOL;
    arrKL6301_in    AT %I* : ARRAY[1..24] OF BYTE;
    arrKL6301_out   AT %Q* : ARRAY[1..24] OF BYTE;
    stDataRec              : EIB_REC;
END_VAR
```

bSet : Eingangsvariable für den Setz-Taster.

bReset : Eingangsvariable für den Reset-Taster.

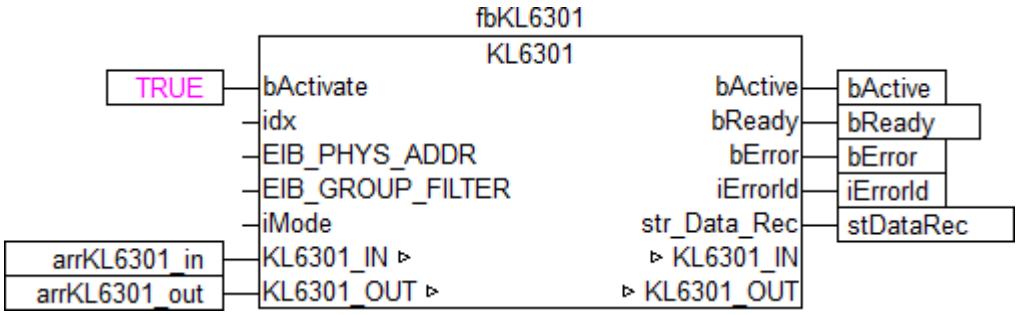
arrKL6301_in : Eingangsvariable für die EIB-Klemme.

arrKL6301_out : Ausgangsvariable für die EIB-Klemme.

stDataRec : Wird für die Kommunikation [▶ 67] mit EIB benötigt.

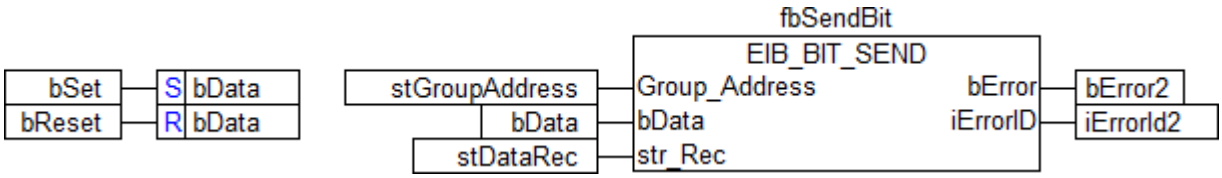
Alle Bausteine bei EIB müssen in derselben Task aufgerufen werden.

Legen Sie daher ein MAIN-Programm (CFC) an in dem die Bausteine KL6301() [▶ 26] und EIB_BIT_SEND() [▶ 57] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, mit *arrKL6301_in*, *arrKL6301_out* und *stDataRec* zu verknüpfen.



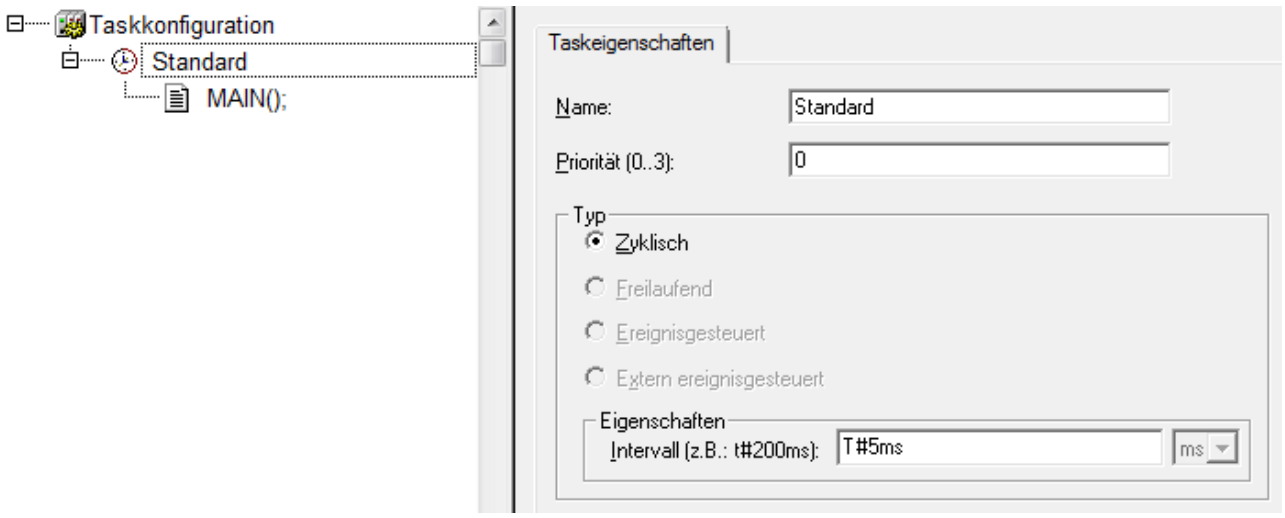
Sample-PC-Comm

Die lokale Variable *bData* wird mit den globalen Variablen *bSet* und *bReset* verknüpft und anschließend mit dem Eingang *bData* des Sendebausteins. Verknüpfen Sie die globale Variable *stDataRec* mit *st_Rec*.



Sample-PC-MAIN

Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit. Genauere Informationen dazu finden Sie in der Beschreibung des Bausteins KL6301() [▶ 26].



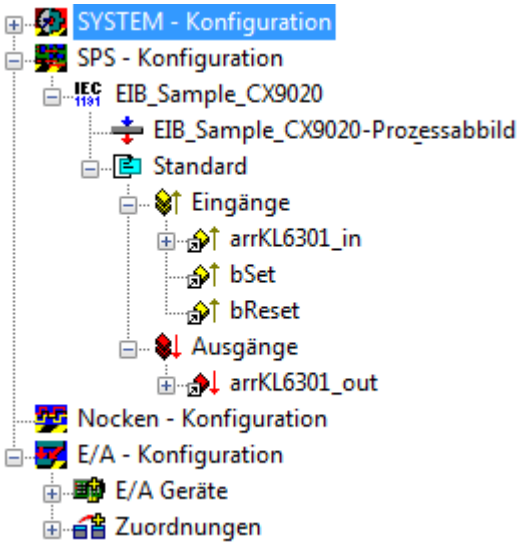
Sample_BC_Task

Laden Sie das Projekt als Bootprojekt auf den CX und speichern Sie es ab.

Konfiguration im System Manager

Legen Sie ein neues TwinCAT System-Manager-Projekt an, wählen Sie als Zielsystem den CX und lassen Sie nach dessen Hardware suchen.

Fügen Sie das oben angelegte SPS-Programm unter SPS-Konfiguration hinzu.



Verknüpfen Sie die globalen Variablen des SPS-Programms nun mit den Ein- und Ausgängen der Busklemmen, erzeugen Sie die Zuordnungen und aktivieren Sie die Konfiguration. Starten Sie dann das Gerät im Run-Modus.
Ihr CX ist jetzt einsatzbereit.

Durch Betätigen der Taster kann der Schaltausgang gesetzt bzw. zurückgesetzt werden.

5.3 Integration in TwinCAT (BC9191)

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für EIB in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es soll der Zustand eines Schaltausgangs per Taster verändert werden.

https://infosys.beckhoff.com/content/1031/tcplclibeib/Resources/11993061643.zip https://infosys.beckhoff.com/content/1031/tcplclibeib/Resources/11993061643.zip

Hardware

Einrichtung der Komponenten

Es wird folgende Hardware benötigt:

- 1x Busklemmen Controller BC9191
- 1x Potenzialeinspeiseklemme 24V DC
- 1x Digitale 2-Kanal-Eingangsklemme KL1002 (für die Setz und Reset Funktion)
- 1x EIB-Klemme KL6301
- 1x Endklemme KL9010

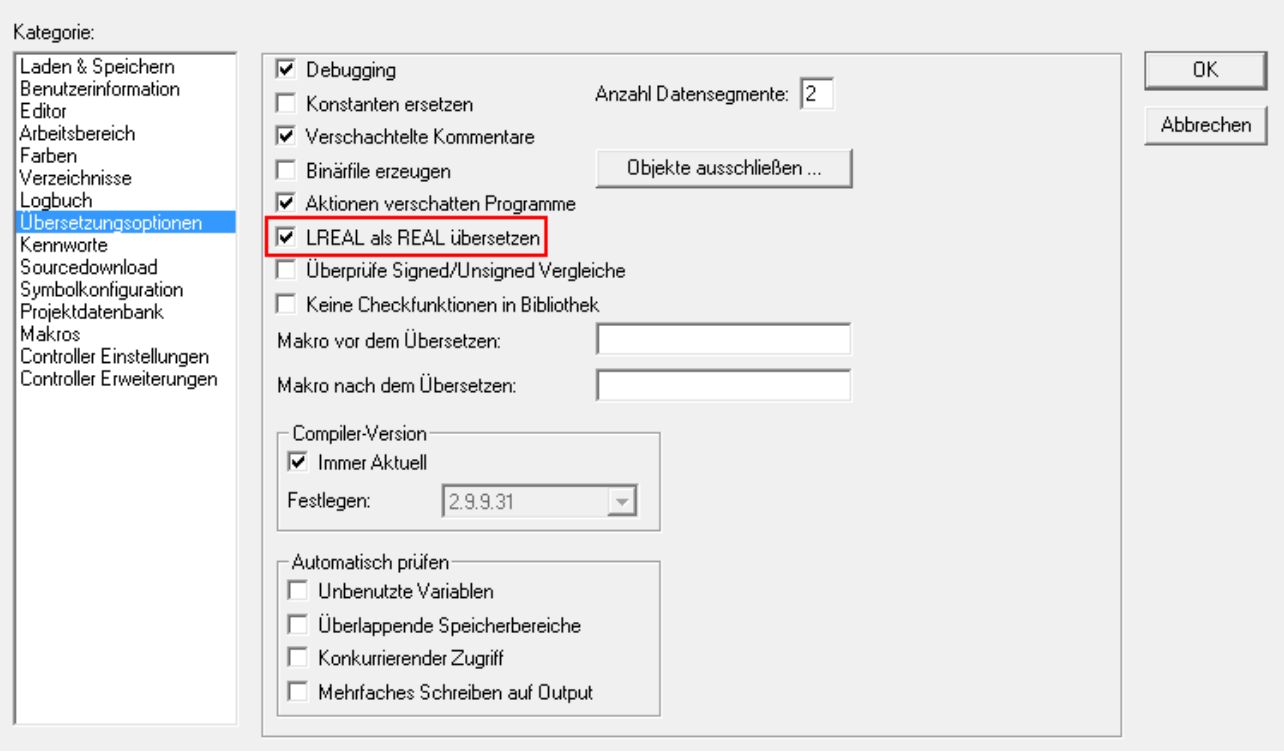
Richten Sie die Hardware sowie die EIB-Komponenten wie in den entsprechenden Dokumentationen beschrieben ein.

Dieses Beispiel geht davon aus, dass ein Setz-Taster auf den ersten und ein Reset-Taster auf den zweiten Eingang der KL1002 gelegt wurde und die EIB Gruppenadresse vom Schaltausgang bekannt ist.

Software

Erstellung des SPS-Programms

Erstellen Sie ein neues SPS-Projekt für BC-basierte Systeme (BCxx50 über AMS) und fügen die Bibliotheken *TcEIB.lbx* und *TcSystemBCxx50.lbx* hinzu. Gehen Sie danach im Menü auf *Projekt → Optionen... → Übersetzungsoptionen* und wählen *LREAL als REAL übersetzen* an.



Erzeugen Sie als Nächstes die folgenden globalen Variablen:

```
VAR_GLOBAL
    bSet        AT %I*     : BOOL;
    bReset      AT %I*     : BOOL;
    arrKL6301_in    AT %I* : ARRAY[1..24] OF BYTE;
    arrKL6301_out   AT %Q* : ARRAY[1..24] OF BYTE;
    stDataRec              : EIB_REC;
END_VAR
```

bSet : Eingangsvariable für den Setz-Taster.

bReset : Eingangsvariable für den Reset-Taster.

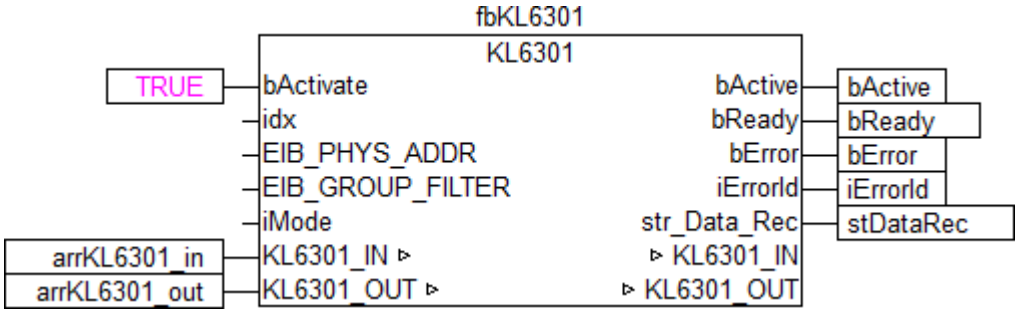
arrKL6301_in : Eingangsvariable für die EIB-Klemme.

arrKL6301_out : Ausgangsvariable für die EIB-Klemme.

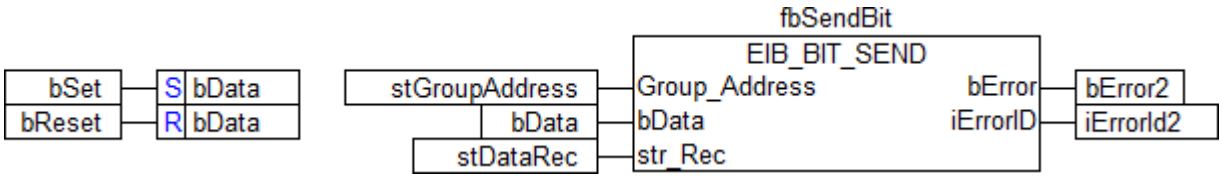
stDataRec : Wird für die Kommunikation [▶ 67] mit EIB benötigt.

Alle Bausteine bei EIB müssen in einer Task ausgeführt werden.

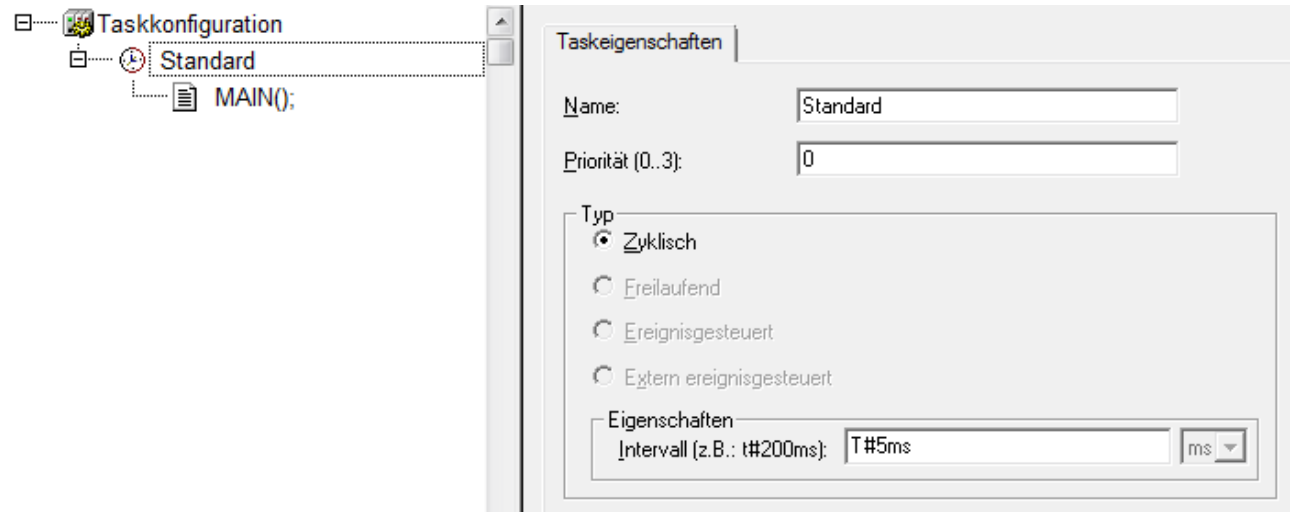
Legen Sie daher ein MAIN-Programm (CFC) an in dem die Bausteine KL6301() [▶ 26] und EIB_BIT_SEND [▶ 57] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, mit *arrKL6301_in*, *arrKL6301_out* und *stDataRec* zu verknüpfen.



Die lokale Variable *bData* wird mit den globalen Variablen *bSet* und *bReset* verknüpft und anschließend mit dem Eingang *bData* des Sendebausteins. Verknüpfen Sie die globale Variable *stDataRec* mit *st_Rec*.



Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit. Genauere Informationen dazu finden Sie in der Beschreibung des Bausteins KL6301() [▶ 26].

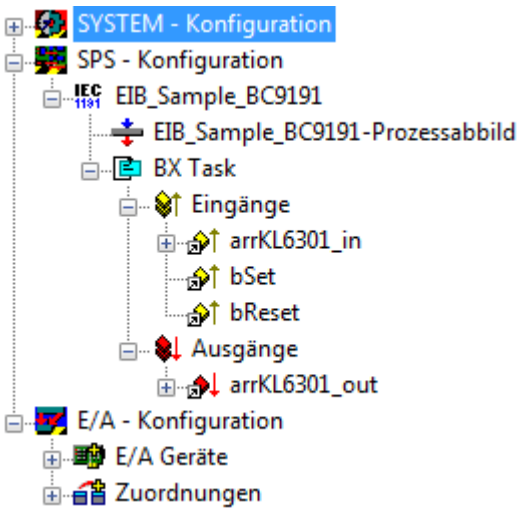


Laden Sie das Projekt als Bootprojekt auf den BC und speichern Sie es ab.

Konfiguration im System Manager

Legen Sie ein neues TwinCAT System-Manager-Projekt an, wählen Sie als Zielsystem den BC und lassen Sie nach dessen Hardware suchen.

Fügen Sie das oben angelegte SPS-Programm unter SPS-Konfiguration hinzu.



Verknüpfen Sie die globalen Variablen des SPS-Programms nun mit den Ein- und Ausgängen der Busklemmen, erzeugen Sie die Zuordnungen und aktivieren Sie die Konfiguration. Starten Sie dann das Gerät im Run-Modus.
Ihr BC ist jetzt einsatzbereit.

Durch Betätigen der Taster kann der Schaltausgang gesetzt bzw. zurückgesetzt werden.

6 Programmierung

Inhalt
Allgemeine Informationen [▶ 22]
KL6301 - Verknüpfung mit dem TwinCAT System Manager [▶ 11]
EIB-Gruppenfilter [▶ 22]

General

Inhalt	Beschreibung
KL6301 [▶ 26]	Kommunikation mit einer KL6301
KL6301_EX [▶ 27]	Kommunikation mit einer KL6301

Read

Bausteine	Beschreibung
EIB_2OCTET_FLOAT_REC [▶ 29]	Empfangen von 2 Byte Float EIB-Daten und konvertieren in REAL
EIB_2OCTET_SIGN_REC [▶ 29]	Empfangen von 2 Byte Sign EIB-Daten und konvertieren in INT
EIB_2OCTET_UNSIGN_REC [▶ 30]	Empfangen von 2 Byte Unsign EIB-Daten und konvertieren in UINT
EIB_3BIT_CONTROL_REC [▶ 30]	Empfangen eines "3 Bit Controlled" Datentypen
EIB_4OCTET_FLOAT_REC [▶ 31]	Empfangen von 4 Byte Float EIB-Daten und konvertieren in REAL
EIB_4OCTET_SIGN_REC [▶ 31]	Empfangen von 4 Byte Sign EIB-Daten und konvertieren in DINT
EIB_4OCTET_UNSIGN_REC [▶ 32]	Empfangen von 4 Byte Unsign EIB-Daten und konvertieren in UDINT
EIB_8BIT_SIGN_REC [▶ 32]	Empfangen von 8 Bit EIB-Daten und konvertieren in INT
EIB_8BIT_UNSIGN_REC [▶ 33]	Empfangen von 8 Bit EIB-Daten und konvertieren in BYTE
EIB_ALL_DATA_TYPES_REC [▶ 33]	Empfängt beliebige EIB Daten
EIB_ALL_DATA_TYPES_REC_EX [▶ 34]	Empfängt beliebige EIB Daten
EIB_BIT_CONTROL_REC [▶ 35]	Empfangen eines "1 Bit Controlled" Datentypen
EIB_BIT_REC [▶ 35]	Empfangen von 1 Bit EIB-Daten und konvertieren in BOOL
EIB_DATE_REC [▶ 36]	Empfangen eines Datums
EIB_TIME_REC [▶ 36]	Empfangen einer Zeit

Send

Bausteine	Beschreibung
EIB_2OCTET_FLOAT_SEND [▶ 37]	Senden eines REAL Wertes (Konvertierung in 2 Byte Float EIB)
EIB_2OCTET_FLOAT_SEND_EX [▶ 38]	Senden eines REAL Wertes (Konvertierung in 2 Byte Float EIB)
EIB_2OCTET_SIGN_SEND [▶ 39]	Senden eines INT Wertes (Konvertierung in 2 Byte Sign EIB)
EIB_2OCTET_SIGN_SEND_EX [▶ 40]	Senden eines INT Wertes (Konvertierung in 2 Byte Sign EIB)

Bausteine	Beschreibung
EIB_2OCTET_UNSIGN_SEND [▶ 41]	Senden eines UINT Wertes (Konvertierung in 2 Byte Unsign EIB)
EIB_2OCTET_UNSIGN_SEND_EX [▶ 42]	Senden eines UINT Wertes (Konvertierung in 2 Byte Unsign EIB)
EIB_3BIT_CONTROL_SEND [▶ 43]	Senden eines "3 Bit Controlled" Datentyypen
EIB_4OCTET_FLOAT_SEND [▶ 44]	Senden eines REAL Wertes (Konvertierung in 4 Byte Float EIB)
EIB_4OCTET_FLOAT_SEND_EX [▶ 45]	Senden eines REAL Wertes (Konvertierung in 4 Byte Float EIB)
EIB_4OCTET_SIGN_SEND [▶ 46]	Senden eines DINT Wertes (Konvertierung in 4 Byte Sign EIB)
EIB_4OCTET_SIGN_SEND_EX [▶ 47]	Senden eines DINT Wertes (Konvertierung in 4 Byte Sign EIB)
EIB_4OCTET_UNSIGN_SEND [▶ 48]	Senden eines UDINT Wertes (Konvertierung in 4 Byte Unsign EIB)
EIB_8BIT_SIGN_SEND [▶ 49]	Senden eines INT Wertes (Konvertierung in 1 Byte Sign EIB)
EIB_8BIT_SIGN_SEND_EX [▶ 50]	Senden eines INT Wertes (Konvertierung in 1 Byte Sign EIB)
EIB_8BIT_UNSIGN_SEND [▶ 51]	Senden eines BYTE Wertes (Konvertierung in 1 Byte Unsign EIB)
EIB_8BIT_UNSIGN_SEND_EX [▶ 52]	Senden eines BYTE Wertes (Konvertierung in 1 Byte Unsign EIB)
EIB_ALL_DATA_TYPES_SEND [▶ 54]	Senden von beliebigen EIB Daten
EIB_BIT_CONTROL_SEND [▶ 56]	Senden eines "1 Bit Controlled" Datentypen
EIB_BIT_SEND [▶ 57]	Senden eines BOOL Wertes
EIB_BIT_SEND_EX [▶ 58]	Senden eines BOOL Wertes
EIB_BIT_SEND_MANUAL [▶ 59]	Senden eines BOOL Wertes
EIB_DATE_SEND [▶ 60]	Senden eines Datums
EIB_READ_SEND [▶ 60]	Senden eines *Read_Group_Req*
EIB_TIME_SEND [▶ 61]	Senden einer Zeit

Functions

Bausteine	Beschreibung
F_CONV_2GROUP_TO_3GROUP [▶ 64]	Umwandlung einer 2 stufigen Gruppenadresse in eine 3 stufige Gruppenadresse
F_CONV_3GROUP_TO_2GROUP [▶ 64]	Umwandlung einer 3 stufigen Gruppenadresse in eine 2 stufige Gruppenadresse

Enums

Datentypen	Beschreibung
EIB_ERROR_CODE [▶ 64]	Fehlermeldungen
EIB_PRIORITY [▶ 66]	Priorität des EIB-Telegramms

Structs

Datentypen	Beschreibung
EIB_GROUP_ADDR [▶ 66]	3-stufige Gruppenadresse
EIB_GROUP_ADDR_2GROUP [▶ 66]	2-stufige Gruppenadresse

Datentypen	Beschreibung
EIB_GROUP_FILTER [▶ 67]	Gruppenfilter
EIB_PHYS_ADDR [▶ 67]	Physikalische Adresse
EIB_REC [▶ 67]	Verbindet die Sende- und Empfangs-Bausteine mit dem Baustein *KL6301*

6.1 Allgemeine Informationen

Installation

Ab TwinCAT 2.11 Build 2229 (R3 und x64 Engineering) werden die Bibliotheken "TcEIB.lib/.lb6/.lbx" standardmäßig mitinstalliert.

Name der Bibliothek

Diese Bibliothek ersetzt die "TcKL6301.lib/.lb6./.lbx". Es hat sich nur der Name der Bibliothek geändert. Die Bausteine sind kompatibel.

Weitere erforderliche Bibliotheken

Für PC-Systeme (x86) und Embedded-PCs (CXxxxx):

- Standard.lib
- TcBase.lib
- TcSystem.lib

Für Busklemmen-Controller der Serie BCxx00:

- Standard.lb6
- PlcHelperBC.lb6

Für Busklemmen-Controller der Serien BCxx50, BCxx20 und BC9191:

- Standard.lbx
- TcBaseBCxx50.lbx
- TcSystemBCxx50.lbx

Für Busklemmen-Controller der Serie BXxx00:

- Standard.lbx
- TcBaseBX.lbx
- TcSystemBX.lbx

Speicherauslastung

Durch Einbinden der Bibliothek wird bereits SPS-Programmspeicher verbraucht. Abhängig vom Applikationsprogramm kann daher der verbleibende Speicher nicht ausreichend sein.

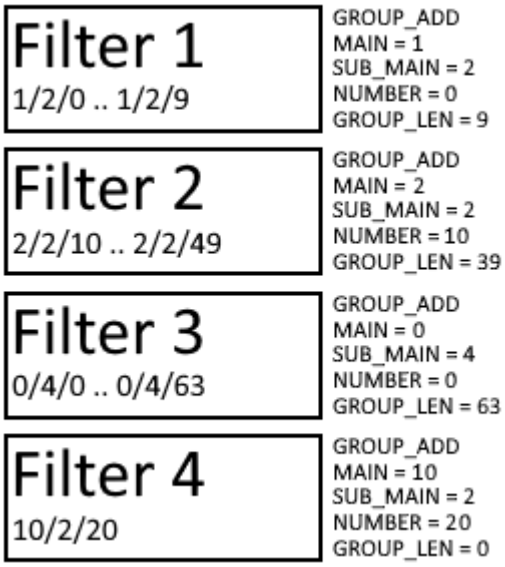
6.2 EIB-Gruppenfilter

Bevor die KL6301 in den Datenaustausch gehen kann, müssen die EIB-Gruppenfilter parametriert sein. Die Filter sind für alle Daten mit einer Gruppenadresse die zur KL6301 geschickt werden notwendig. Jedes Gruppentelegramm, das auch in den Filtern enthalten ist, wird mit einem ACK beantwortet und in die Prozessdaten eingetragen, d.h. in den Funktionsbausteinen sichtbar. EIB-Telegramme mit einer Gruppenadresse, die nicht in den Filtern enthalten sind, werden von der KL6301 verworfen.

Beispiel

Filter 1 Gruppenadresse 1/2/0 Länge: 20
Alle EIB-Telegramme mit der Gruppenadresse 1/2/0 - 1/2/19 passieren den Filter

Es muss immer mindestens ein Filter aktiviert werden.
Der gewählte Mode legt die Anzahl und Länge der Gruppenfilter fest. Die Längenangabe startet bei 0, was genau einem Eintrag entspricht.



Änderung zur Firmware B1 und Bibliothek Version V3

Mit der Firmware-Version B1 und der TwinCAT-Bibliothek TcEIB (V3.000.000) kann man **statt 4 Filtern auch 8 Filter** parametrieren. Die maximale Länge der einzelnen Filter reduziert sich allerdings von 64 Einträgen auf 32 Einträge pro Filtergruppe. So bleibt die Summe der maximal empfangenden Gruppenfilter mit 256 gleich.

6.3 Bausteine

General

Inhalt	Beschreibung
KL6301 [▶ 26]	Kommunikation mit einer KL6301
KL6301_EX [▶ 27]	Kommunikation mit einer KL6301

Read

Bausteine	Beschreibung
EIB_2OCTET_FLOAT_REC [▶ 29]	Empfangen von 2 Byte Float EIB-Daten und konvertieren in REAL
EIB_2OCTET_SIGN_REC [▶ 29]	Empfangen von 2 Byte Sign EIB-Daten und konvertieren in INT
EIB_2OCTET_UNSIGN_REC [▶ 30]	Empfangen von 2 Byte Unsign EIB-Daten und konvertieren in UINT
EIB_3BIT_CONTROL_REC [▶ 30]	Empfangen eines "3 Bit Controlled" Datentypen
EIB_4OCTET_FLOAT_REC [▶ 31]	Empfangen von 4 Byte Float EIB-Daten und konvertieren in REAL
EIB_4OCTET_SIGN_REC [▶ 31]	Empfangen von 4 Byte Sign EIB-Daten und konvertieren in DINT
EIB_4OCTET_UNSIGN_REC [▶ 32]	Empfangen von 4 Byte Unsign EIB-Daten und konvertieren in UDINT
EIB_8BIT_SIGN_REC [▶ 32]	Empfangen von 8 Bit EIB-Daten und konvertieren in INT
EIB_8BIT_UNSIGN_REC [▶ 33]	Empfangen von 8 Bit EIB-Daten und konvertieren in BYTE

Bausteine	Beschreibung
EIB_ALL_DATA_TYPES_REC [▶ 33]	Empfängt beliebige EIB Daten
EIB_ALL_DATA_TYPES_REC_EX [▶ 34]	Empfängt beliebige EIB Daten
EIB_BIT_CONTROL_REC [▶ 35]	Empfangen eines "1 Bit Controlled" Datentypen
EIB_BIT_REC [▶ 35]	Empfangen von 1 Bit EIB-Daten und konvertieren in BOOL
EIB_DATE_REC [▶ 36]	Empfangen eines Datums
EIB_TIME_REC [▶ 36]	Empfangen einer Zeit

Send

Bausteine	Beschreibung
EIB_2OCTET_FLOAT_SEND [▶ 37]	Senden eines REAL Wertes (Konvertierung in 2 Byte Float EIB)
EIB_2OCTET_FLOAT_SEND_EX [▶ 38]	Senden eines REAL Wertes (Konvertierung in 2 Byte Float EIB)
EIB_2OCTET_SIGN_SEND [▶ 39]	Senden eines INT Wertes (Konvertierung in 2 Byte Sign EIB)
EIB_2OCTET_SIGN_SEND_EX [▶ 40]	Senden eines INT Wertes (Konvertierung in 2 Byte Sign EIB)
EIB_2OCTET_UNSIGN_SEND [▶ 41]	Senden eines UINT Wertes (Konvertierung in 2 Byte Unsign EIB)
EIB_2OCTET_UNSIGN_SEND_EX [▶ 42]	Senden eines UINT Wertes (Konvertierung in 2 Byte Unsign EIB)
EIB_3BIT_CONTROL_SEND [▶ 43]	Senden eines "3 Bit Controlled" Datentyypen
EIB_4OCTET_FLOAT_SEND [▶ 44]	Senden eines REAL Wertes (Konvertierung in 4 Byte Float EIB)
EIB_4OCTET_FLOAT_SEND_EX [▶ 45]	Senden eines REAL Wertes (Konvertierung in 4 Byte Float EIB)
EIB_4OCTET_SIGN_SEND [▶ 46]	Senden eines DINT Wertes (Konvertierung in 4 Byte Sign EIB)
EIB_4OCTET_SIGN_SEND_EX [▶ 47]	Senden eines DINT Wertes (Konvertierung in 4 Byte Sign EIB)
EIB_4OCTET_UNSIGN_SEND [▶ 48]	Senden eines UDINT Wertes (Konvertierung in 4 Byte Unsign EIB)
EIB_8BIT_SIGN_SEND [▶ 49]	Senden eines INT Wertes (Konvertierung in 1 Byte Sign EIB)
EIB_8BIT_SIGN_SEND_EX [▶ 50]	Senden eines INT Wertes (Konvertierung in 1 Byte Sign EIB)
EIB_8BIT_UNSIGN_SEND [▶ 51]	Senden eines BYTE Wertes (Konvertierung in 1 Byte Unsign EIB)
EIB_8BIT_UNSIGN_SEND_EX [▶ 52]	Senden eines BYTE Wertes (Konvertierung in 1 Byte Unsign EIB)
EIB_ALL_DATA_TYPES_SEND [▶ 54]	Senden von beliebigen EIB Daten
EIB_BIT_CONTROL_SEND [▶ 56]	Senden eines "1 Bit Controlled" Datentypen
EIB_BIT_SEND [▶ 57]	Senden eines BOOL Wertes
EIB_BIT_SEND_EX [▶ 58]	Senden eines BOOL Wertes
EIB_BIT_SEND_MANUAL [▶ 59]	Senden eines BOOL Wertes
EIB_DATE_SEND [▶ 60]	Senden eines Datums
EIB_READ_SEND [▶ 60]	Senden eines *Read_Group_Req*
EIB_TIME_SEND [▶ 61]	Senden einer Zeit

6.3.1 Funktionsbausteine Details

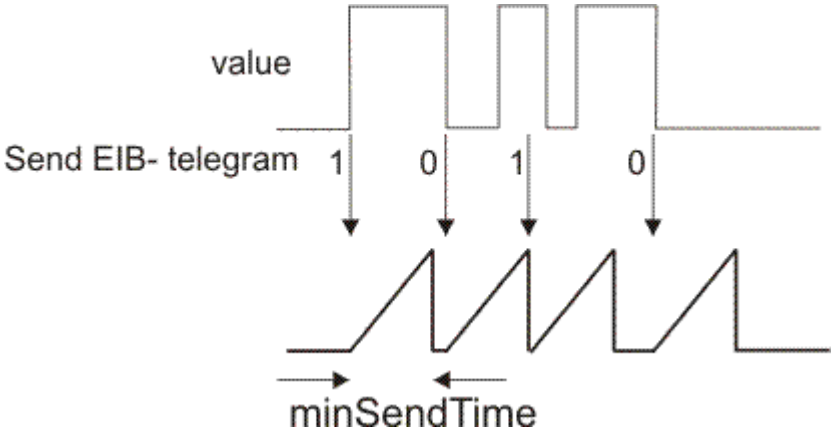
Beschreibung	_Rec	_Send				
		_Send	First Cycle	Delta, min. Send Time	Polling	Auto/Manuell
EIB_BIT	ja [▶ 35]	ja [▶ 57]	nein	200msec	nein	Auto
EIB_BIT_EX	nein	ja [▶ 58]	ja	1sec, variable	10sec, variable	Auto/Manuell
EIB_BIT_MANUAL	nein	ja [▶ 59]	nein	nein	nein	Manuell
EIB_BIT_CONTROL	ja [▶ 35]	ja [▶ 56]	nein	200msec	nein	Auto
EIB_3BIT_CONTROL	ja [▶ 30]	ja [▶ 43]	nein	200msec	nein	Auto
EIB_8BIT_SIGN	ja [▶ 32]	ja [▶ 49]	nein	1sec	nein	Auto
EIB_8BIT_SIGN_EX	nein	ja [▶ 50]	ja	1sec, variable	500msec, variable	Auto/Manuell
EIB_8BIT_UNSIGN	ja [▶ 33]	ja [▶ 51]	nein	1sec	nein	Auto
EIB_8BIT_UNSIGN_EX	nein	ja [▶ 52]	ja	1sec, variable	500msec, variable	Auto/Manuell
EIB_2OCTET_SIGN	ja [▶ 29]	ja [▶ 39]	nein	1sec	nein	Auto
EIB_2OCTET_SIGN_EX	nein	ja [▶ 40]	ja	1sec, variable	500msec, variable	Auto/Manuell
EIB_2OCTET_UNSIGN	ja [▶ 30]	ja [▶ 41]	nein	1sec	nein	Auto
EIB_2OCTET_UNSIGN_EX	nein	ja [▶ 42]	ja	1sec, variable	500msec, variable	Auto/Manuell
EIB_2OCTET_FLOAT	ja [▶ 29]	ja [▶ 37]	nein	1sec	nein	Auto
EIB_2OCTET_FLOAT_EX	nein	ja [▶ 38]	ja	1sec, variable	500msec, variable	Auto/Manuell
EIB_TIME	ja [▶ 36]	ja [▶ 61]	ja	nein	5min	Auto
EIB_DATE	ja [▶ 36]	ja [▶ 60]	ja	nein	5min	Auto
EIB_4OCTET_SIGN	ja [▶ 31]	ja [▶ 46]	nein	1sec	nein	Auto
EIB_4OCTET_SIGN_EX	nein	ja [▶ 47]	ja	1sec, variable	500msec, variable	Auto/Manuell
EIB_4OCTET_UNSIGN	ja [▶ 32]	ja [▶ 48]	nein	1sec	nein	Auto
EIB_4OCTET_FLOAT	ja [▶ 31]	ja [▶ 44]	nein	1sec	nein	Auto
EIB_4OCTET_FLOAT_EX	nein	ja [▶ 45]	ja	1sec, variable	10min, variable	Auto/Manuell
EIB_READ	nein	ja [▶ 60]	nein	nein	nein	Manuell
EIB_ALL_DATA_TYPES	ja [▶ 33]	ja [▶ 54]	nein	1sec, variable	100msec, variable	Auto/Manuell
EIB_ALL_DATA_TYPES_EX	ja [▶ 34]	nein	nein	nein	nein	nein

_Rec: ja - Empfangen wird unterstützt, nein - Empfangen wird nicht unterstützt

_Send: ja - Senden wird unterstützt, nein - Senden wird nicht unterstützt

First Cycle: es wird beim ersten Aufruf des Bausteins ein EIB-Telegramm versendet

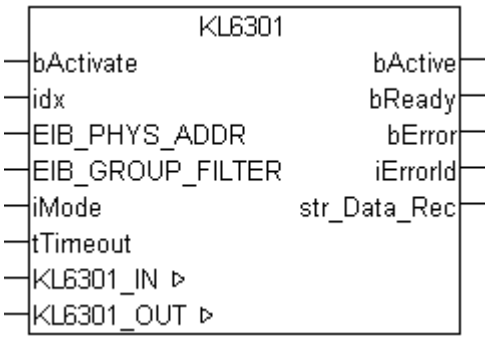
Delta, min. Send Time: es wird nur bei Änderung der Daten ein EIB-Telegramm versendet. Über den Parameter "min. Send Time" ist ein "Sende-Filter" aktiviert, der die Reaktionszeit bei der relativ ersten Eingangsänderung nicht verlängert, aber bei unmittelbar darauf folgenden Änderungen aktiv ist. Die min. Send Time (Sendeverzögerungszeit) beschreibt die Zeitspanne, die zwischen dem Versenden zweier Telegramme mindestens abgewartet werden muss. Die min. Send Time reduziert die Buskommunikation und stellt sicher das auch Sendeaufträge von anderen Bausteinen ihr EIB-Telegramm absetzen können.



Polling: die Daten werden automatisch im angegebenen Zeitabstand versendet, auch wenn sie sich nicht verändert haben

Auto/Manuell: Auto - Beim Aufruf des Bausteins wird automatisch Daten gesendet, Manuell - Daten werden nur gesendet wenn dies beim Baustein angefordert wird

6.3.2 KL6301



Dieser Funktionsbaustein übernimmt die Kommunikation mit der EIB-Busklemme KL6301. Über diesen Baustein wird die KL6301 konfiguriert und der Datenaustausch mit dem EIB-Netzwerk gestartet.

Einschränkungen

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

VAR_INPUT

```
bActivate         : BOOL;
idx               : INT := 1;
EIB_PHYS_ADDR     : EIB_PHYS_ADDR;
EIB_GROUP_FILTER  : ARRAY [1..8] OF EIB_GROUP_FILTER;
iMode             : INT;
tTimeout          : TIME := t#5s;
```

bActivate: Aktiviert den Baustein, der die KL6301 als Erstes konfiguriert und dann in den Datenaustausch setzt.

idx: Die idx Nummer muss beim Einsatz von mehr als einer Busklemme pro PLC Programm für jede KL6301 eindeutig sein. Gültige Werte von 1...64.

EIB_PHYS_ADDR: Physikalische EIB Adresse (siehe EIB_PHYS_ADDR [▶ 67]). Default Adresse ist 1.2.3. Diese Adresse muss im EIB-Netzwerk einmalig sein!

EIB_GROUP_FILTER: Filter für die Gruppenadressen (siehe EIB_GROUP_FILTER [▶ 67]). Es sind maximal 8 Filter möglich.

iMode:
0 - Für Firmware B0 und höher - 4 Filter a 64 Einträge
1 - Für Firmware B1 und höher - 8 Filter a 32 Einträge
2 - Für Firmware B3 und höher - 8 Filter a 32 Einträge invertiert. Bei Querkommunikation von Telegrammen innerhalb des EIB/KNX Netzwerks, die nicht mit der KL6301 verbunden sind, ist darauf zu achten, dass diese Gruppenadressen im Filter enthalten sind, damit die Klemme kein ACK versendet.
100 - Für Firmware B1 und höher - Monitor Funktion, alle Gruppenadresstelegramm werden empfangen. Die Telegramme werden nicht bestätigt (es wird kein ACK gesendet). Im Monitor Betrieb kann nicht gesendet werden.

tTimeout: Zeit, die ein Sende-Funktionsbaustein zum Übermitteln eines EIB-Telegramms hat, bis ein Timeout signalisiert wird.

VAR_OUTPUT

```
bActive             : BOOL;
bReady              : BOOL;
bError              : BOOL;
iErrorId            : EIB_Error_Code;
str_Data_Rec        : EIB_REC;
```

bActive: Der Baustein wurde aktiviert.

bReady: Der Baustein ist bereit Daten zu senden und zu empfangen.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

str_Data_Rec: Datenstruktur die mit den Send- und Receive-Bausteinen verbunden wird (siehe EIB_REC [▶ 67]).

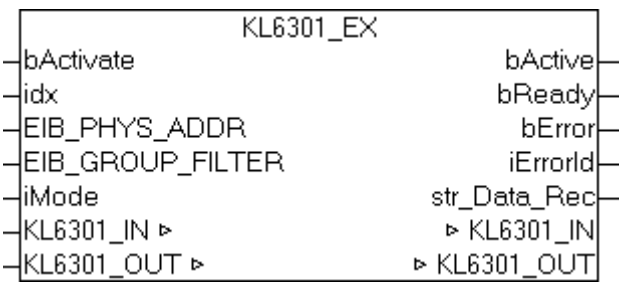
VAR_IN_OUT

```
KL6301_IN           : ARRAY [1..24] OF BYTE;
KL6301_OUT          : ARRAY [1..24] OF BYTE;
```

KL6301_IN: Wird mit den Eingangsadressen der KL6301 verknüpft.

KL6301_OUT: Wird mit den Ausgangsadressen der KL6301 verknüpft.

6.3.3 KL6301_EX



Dieser Funktionsbaustein übernimmt die Kommunikation mit der EIB-Busklemme KL6301. Über diesen Baustein wird die KL6301 konfiguriert und der Datenaustausch mit dem EIB-Netzwerk gestartet.

BETA: ETS Unterstützung für Suche und LED Blinken.

Einschränkungen

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

VAR_INPUT

```
bActivate          : BOOL;
idx                : INT := 1;
EIB_PHYS_ADDR      : EIB_PHYS_ADDR;
EIB_GROUP_FILTER   : ARRAY [1..8] OF EIB_GROUP_FILTER;
iMode              : INT;
```

bActivate: Aktiviert den Baustein, der die KL6301 als Erstes konfiguriert und dann in den Datenaustausch setzt.

idx: Die idx Nummer muss beim Einsatz von mehr als einer Busklemme pro PLC Programm für jede KL6301 eindeutig sein. Gültige Werte von 1...64.

EIB_PHYS_ADDR: Physikalische EIB Adresse (siehe EIB_PHYS_ADDR [▶ 67]). Default Adresse ist 1.2.3. Diese Adresse muss im EIB-Netzwerk einmalig sein!

EIB_GROUP_FILTER: Filter für die Gruppenadressen (siehe EIB_GROUP_FILTER [▶ 67]). Es sind maximal 8 Filter möglich.

iMode:
0 - Für Firmware B0 und höher - 4 Filter a 64 Einträge
1 - Für Firmware B1 und höher - 8 Filter a 32 Einträge
2 - Für Firmware B3 und höher - 8 Filter a 32 Einträge invertiert. Bei Querkommunikation von Telegrammen innerhalb des EIB/KNX Netzwerks, die nicht mit der KL6301 verbunden sind, ist darauf zu achten, dass diese Gruppenadressen im Filter enthalten sind, damit die Klemme kein ACK versendet.
100 - Für Firmware B1 und höher - Monitor Funktion, alle Gruppenadresstelegramm werden empfangen. Die Telegramme werden nicht bestätigt (es wird kein ACK gesendet). Im Monitor Betrieb kann nicht gesendet werden.

VAR_OUTPUT

```
bActive            : BOOL;
bReady             : BOOL;
bError             : BOOL;
iErrorId           : EIB_Error_Code;
str_Data_Rec       : EIB_REC;
```

bActive: Der Baustein wurde aktiviert.

bReady: Der Baustein ist bereit Daten zu senden und zu empfangen.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorId* beschrieben.

iErrorId: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

str_Data_Rec: Datenstruktur die mit den Send- und Receive-Bausteinen verbunden wird (siehe EIB_REC [▶ 67]).

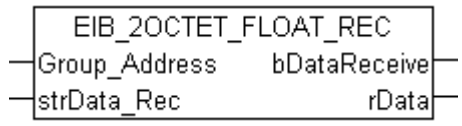
VAR_IN_OUT

```
KL6301_IN          : ARRAY [1..24] OF BYTE;
KL6301_OUT         : ARRAY [1..24] OF BYTE;
```

KL6301_IN: Wird mit den Eingangsadressen der KL6301 verknüpft.

KL6301_OUT: Wird mit den Ausgangsadressen der KL6301 verknüpft.

6.3.4 EIB_2OCTET_FLOAT_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 2 Byte Float EIB-Daten und konvertiert diese in eine IEC61131-3 REAL-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

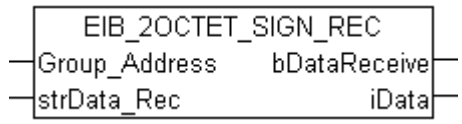
VAR_OUTPUT

```
bDataReceive    : BOOL;
rData           : REAL;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

rData: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

6.3.5 EIB_2OCTET_SIGN_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 2 Byte Sign EIB-Daten und konvertiert diese in eine IEC61131-3 INT-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

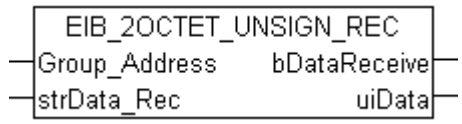
VAR_OUTPUT

```
bDataReceive    : BOOL;
iData           : INT;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

iData: Enthält die Nutzdaten des empfangenden EIB-Telegramms

6.3.6 EIB_2OCTET_UNSIGN_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 2 Byte Unsign EIB-Daten und konvertiert diese in eine IEC61131-3 UINT Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive    : BOOL;
uiData          : UINT;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt, wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

uiData: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

6.3.7 EIB_3BIT_CONTROL_REC

Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 4 Bit EIB-Daten und konvertiert diese in eine IEC61131-3 BOOL-Variable und in eine BYTE-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive    : BOOL;
bControl        : BOOL;
byRange         : BYTE;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

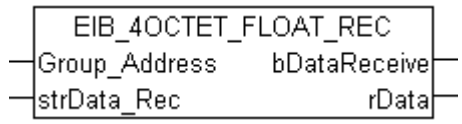
bControl: Gültige Werte (TRUE/FALSE).

byRange: Gültige Werte (000b..111b).

Aufteilung der 4 Bit in die Variablen bControl und byRange.

bControl	byRange.2	byRange.1	byRange.0

6.3.8 EIB_4OCTET_FLOAT_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 4 Byte Float EIB-Daten und konvertiert diese in eine IEC61131-3 REAL Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

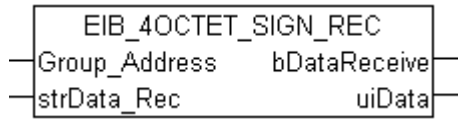
VAR_OUTPUT

```
bDataReceive    : BOOL;
rData           : REAL;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

rData: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

6.3.9 EIB_4OCTET_SIGN_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 4 Byte Sign EIB-Daten und konvertiert diese in eine IEC61131-3 DINT-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

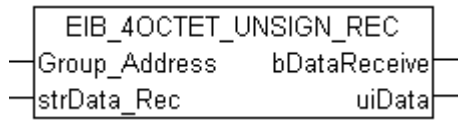
VAR_OUTPUT

```
bDataReceive    : BOOL;
uiData          : DINT;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

uiData: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

6.3.10 EIB_4OCTET_UNSIGN_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 4 Byte Unsign EIB-Daten und konvertiert diese in eine IEC61131-3 UDINT-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

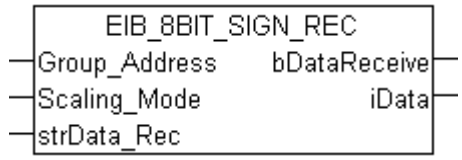
VAR_OUTPUT

```
bDataReceive    : BOOL;
uiData          : UDINT;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

uiData: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

6.3.11 EIB_8BIT_SIGN_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 8 Bit EIB-Daten und konvertiert diese in eine IEC61131-3 INT-Variable. Zusätzlich kann der Wert automatisch umgerechnet werden.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
Scaling_Mode    : INT;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

Scaling_Mode:
0 - Der 8 Bit Wert wird als % Wert ausgegeben 0...100%
1 - Der 8 Bit Wert wird als Grad (Winkel) ausgegeben 0...360°
2 - Der 8 Bit Wert wird als ein Byte Wert ausgegeben 0...255

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

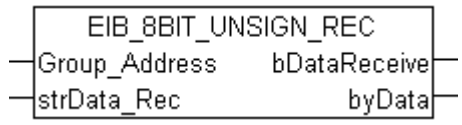
VAR_OUTPUT

```
bDataReceive    : BOOL;
iData           : INT;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

iData: Skalierter Wert, siehe *Scaling_Mode* (-1 - Es wurde ein Scaling Mode außerhalb der Gültigkeit eingegeben).

6.3.12 EIB_8BIT_UNSIGN_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 8 Bit EIB-Daten und konvertiert diese in eine IEC61131-3 BYTE-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

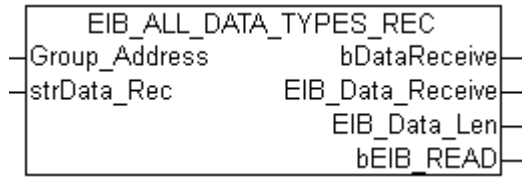
VAR_OUTPUT

```
bDataReceive    : BOOL;
byData          : BYTE;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

byData: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

6.3.13 EIB_ALL_DATA_TYPES_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse variable EIB-Datengrößen und gibt die Rohdaten als ein Byte-ARRAY aus

VAR_INPUT

```
Group_Address      : EIB_GROUP_ADDR;
strData_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive      : BOOL;
EIB_Data_Receive  : ARRAY [1..14] OF BYTE;
EIB_Data_Len      : USINT;
bEIB_READ         : BOOL;
```

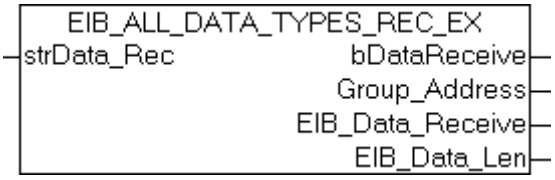
bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt, wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

EIB_Data_Receive: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

EIB_Data_Len: Enthält die Länge der Nutzdaten des empfangenden EIB-Telegramms.
Daten < 8Bit angegebene Länge 1,
Daten >=) 8 Bit angegebene Länge +1
Beispiel: Sie empfangen 1 Bit Daten so ist die Länge in EIB_Data_Len 1. Sie empfangen 2 Byte Daten so ist die Länge in EIB_Data_Len 3.

bEIB_READ: TRUE = EIB Lese Kommando. FALSE = normales EIB-Telegramm (ab V5.2.5).

6.3.14 EIB_ALL_DATA_TYPES_REC_EX



Dieser FB empfängt für alle Gruppenadressen variable EIB Datengrößen und gibt die Rohdaten als Byte-ARRAY aus.

VAR_INPUT

```
strData_Rec       : EIB_REC;
```

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive      : BOOL;
Group_Address     : EIB_GROUP_ADDR;
EIB_Data_Receive  : ARRAY [1..14] OF BYTE;
EIB_Data_Len      : USINT;
```

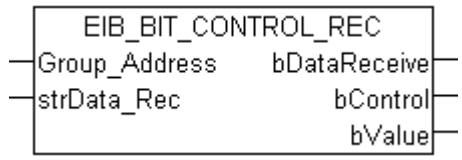
bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

EIB_Data_Receive: Enthält die Nutzdaten des empfangenden EIB-Telegramms.

EIB_Data_Len: Enthält die Länge der Nutzdaten des empfangenden EIB-Telegramms.
Daten < 8 Bit angegebene Länge 1,
Daten >=) 8 Bit angegebene Länge +1
Beispiel: Sie empfangen 1 Bit Daten so ist die Länge in EIB_Data_Len 1. Sie empfangen 2 Byte Daten so ist die Länge in EIB_Data_Len 3.

6.3.15 EIB_BIT_CONTROL_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 2 Bit EIB-Daten und konvertiert diese in zwei IEC61131-3 BOOL-Variablen.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive    : BOOL;
bControl        : BOOL;
bValue          : BOOL;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

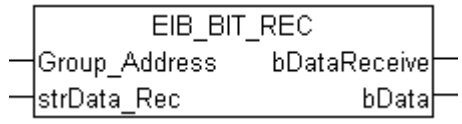
bControl: Gültige Werte (TRUE/FALSE).

bValue: Gültige Werte (TRUE/FALSE).

Aufteilung der 2 Bit in die Variablen bControl und bValue.

bControl	bValue

6.3.16 EIB_BIT_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 1 Bit EIB-Daten und konvertiert diese in eine IEC61131-3 BOOL-Variable.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

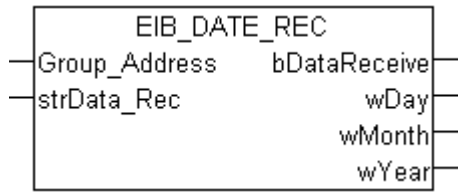
VAR_OUTPUT

```
bDataReceive    : BOOL;
bData           : BOOL;
```

bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

bData: Gültige Werte (TRUE/FALSE).

6.3.17 EIB_DATE_REC



Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 3 Byte EIB-Daten und konvertiert diese in drei IEC61131-3 WORD-Variablen.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive    : BOOL;
wDay            : WORD;
wMonth          : WORD;
wYear           : WORD;
```

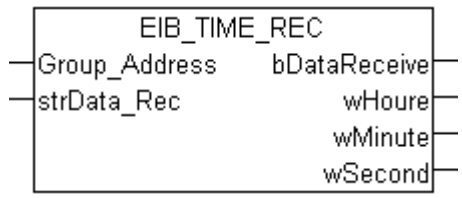
bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

wDay: Datum, Tage [1...31].

wMonth: Datum, Monat [1...12].

wYear: Datum, Jahr [0...99].

6.3.18 EIB_TIME_REC



Anwendung

Dieser Funktionsbaustein empfängt auf der eingestellten Gruppenadresse 3 Byte EIB-Daten und konvertiert diese in drei IEC61131-3 WORD-Variablen.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
strData_Rec     : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

strData_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bDataReceive    : BOOL;
wHoure          : WORD;
wMinute         : WORD;
wSecond         : WORD;
```

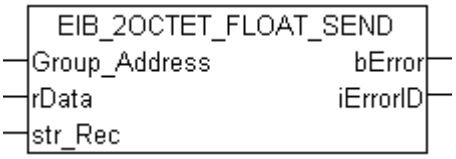
bDataReceive: Dieses Bit wird für genau einen Zyklus auf FALSE gesetzt wenn ein EIB-Telegramm mit der Gruppenadresse empfangen wird.

wHoure: Uhrzeit, in Stunden [0...23].

wMinute: Uhrzeit, in Minute [0...59].

wSecond: Uhrzeit, in Sekunden [0...59].

6.3.19 EIB_2OCTET_FLOAT_SEND



Dieser Funktionsbaustein sendet einen 2 Byte Float EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 REAL Wert zur Verfügung. Die Daten werden nur bei Änderung übertragen. Ändert sich der Wert innerhalb einer Sekunde erneut, werden erst nach Ablauf von einer Sekunde neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
rData           : REAL;
str_Rec         : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

rData: Der Daten-Wert in REAL, wird automatisch in einen EIB 2OCTET FLOAT-Wert gewandelt.

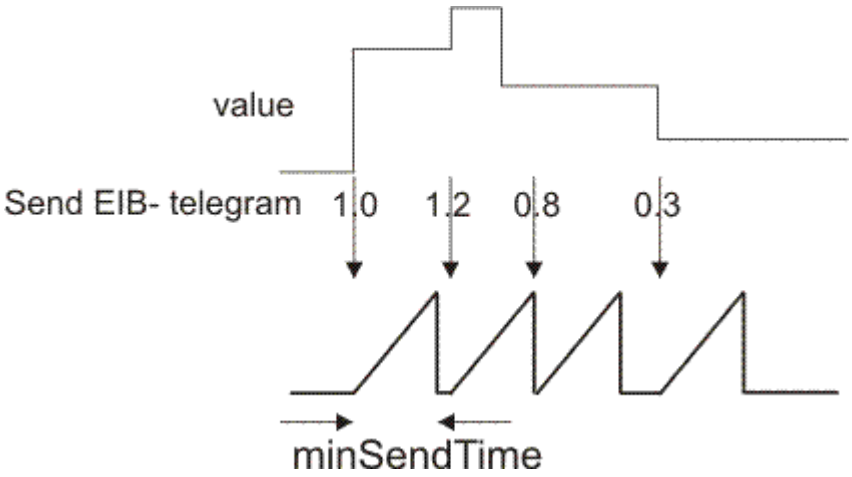
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

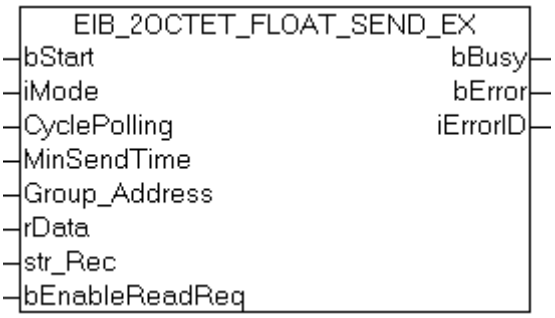
```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.20 EIB_2OCTET_FLOAT_SEND_EX



Dieser Funktionsbaustein sendet einen 2 Byte Float EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 REAL Wert zur Verfügung. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
CyclePolling    : TIME := t#500ms;
MinSendTime     : TIME := t#1s;
Group_Address   : EIB_GROUP_ADDR;
rData           : REAL;
str_Rec         : EIB_REC;
bEnableReadReq : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*).

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

rData: Der Daten-Wert in REAL, wird automatisch in einen EIB 2OCTET FLOAT-Wert gewandelt.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

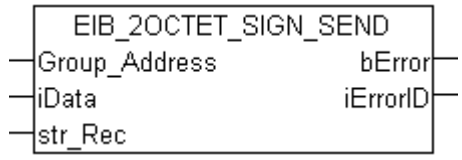
```
bBusy          : BOOL;
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis *bBusy* wieder auf FALSE ist.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.21 EIB_2OCTET_SIGN_SEND



Dieser Funktionsbaustein sendet einen 2 Byte Sign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 INT-Wert zur Verfügung. Die Daten werden nur bei Änderung übertragen. Ändert sich der Wert innerhalb einer Sekunde erneut, werden erst nach Ablauf von einer Sekunde neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung 1). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
iData          : INT;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse von der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]). Diese Gruppenadresse muss in den Filtern eingetragen sein!

iData: Der Daten-Wert in INT, wird automatisch in einen EIB 2OCTET SIGN-Wert gewandelt.

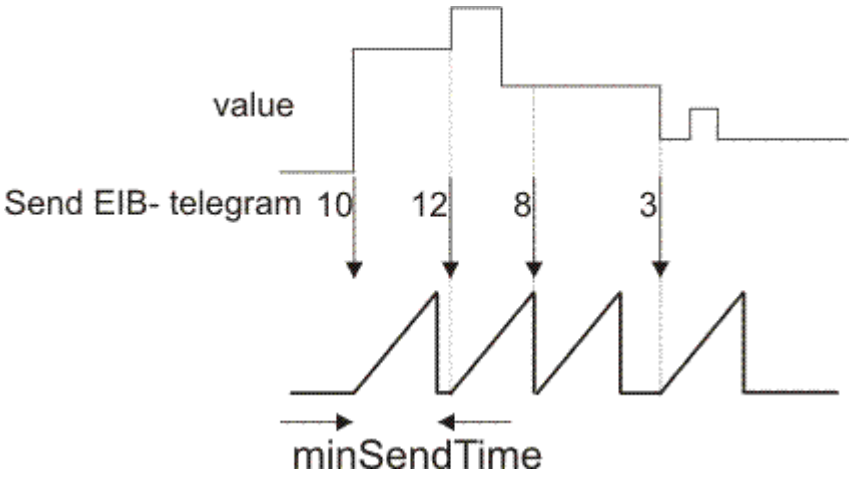
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

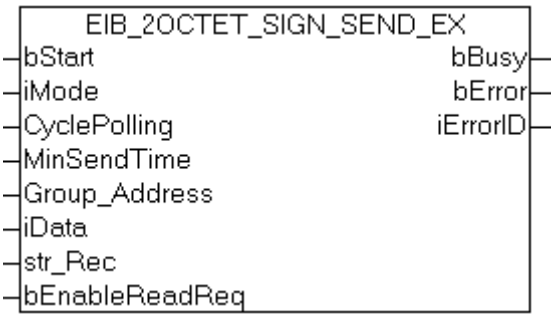
```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.22 EIB_2OCTET_SIGN_SEND_EX



Dieser Funktionsbaustein sendet einen 2 Byte Sign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 INT-Wert zur Verfügung. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
CyclePolling    : TIME := t#500ms;
MinSendTime     : TIME := t#1s;
Group_Address   : EIB_GROUP_ADDR;
iData           : INT;
str_Rec         : EIB_REC;
bEnableReadReq : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*).

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

iData: Der Daten-Wert in INT, wird automatisch in einen EIB 2OCTET SIGN-Wert gewandelt.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

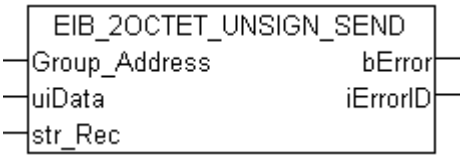
```
bBusy          : BOOL;
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis *bBusy* wieder auf FALSE ist.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.23 EIB_2OCTET_UNSIGN_SEND



Dieser Funktionsbaustein sendet einen 2 Byte Unsign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 UINT-Wert zur Verfügung. Die Daten werden nur bei Änderung übertragen. Ändert sich der Wert innerhalb einer Sekunde erneut, werden erst nach Ablauf von einer Sekunde neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB Telegramm gesendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
uiData         : UINT;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

uiData: Der Datenwert in UINT, wird automatisch in einen EIB 2OCTET UNSIGN-Wert gewandelt.

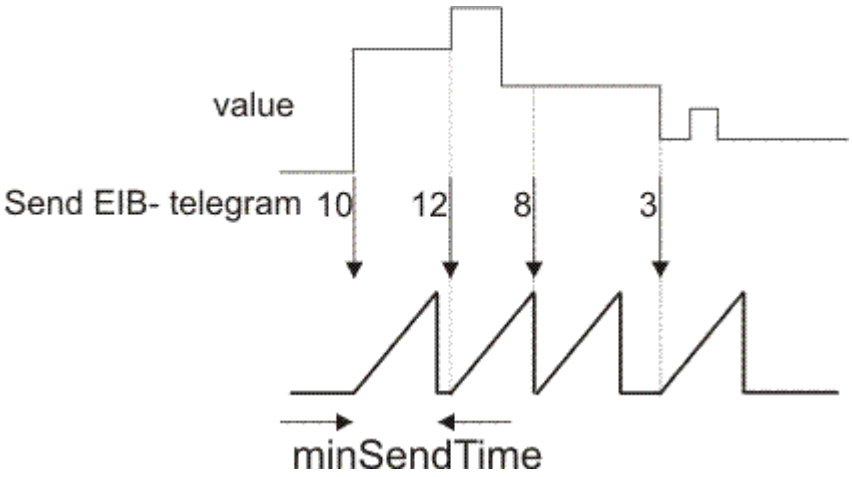
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

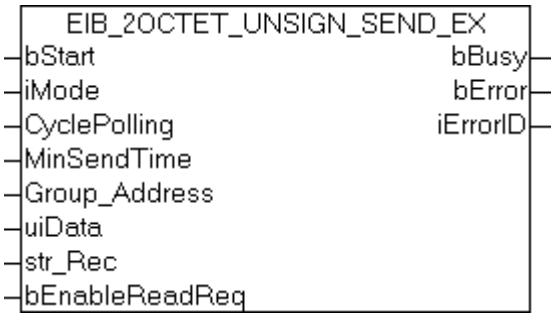
```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.24 EIB_2OCTET_UNSIGN_SEND_EX



Dieser Funktionsbaustein sendet einen 2 Byte Unsign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 UINT-Wert zur Verfügung. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart         : BOOL;
iMode          : INT;
CyclePolling   : TIME := t#500ms;
MinSendTime    : TIME := t#1s;
Group_Address  : EIB_GROUP_ADDR;
uiData         : UINT;
str_Rec        : EIB_REC;
bEnableReadReq : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*).

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

uiData: Der Daten-Wert in UINT, wird automatisch in einen EIB 2OCTET UNSIGN-Wert gewandelt.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

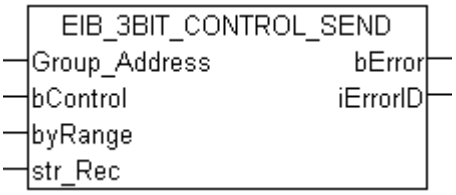
```
bBusy          : BOOL;
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis *bBusy* wieder auf FALSE ist.

bError: Der Ausgang wird TRUE, sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.25 EIB_3BIT_CONTROL_SEND



Dieser Funktionsbaustein sendet einen 4 Bit EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 BOOL und ein BYTE-Wert zur Verfügung. Die Daten werden nur bei Änderung einer der beiden Datentypen übertragen. Ändert sich der Wert innerhalb von 200 ms erneut, wird erst nach Ablauf von 200 ms neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
bControl       : BOOL;
byRange        : BYTE;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

bControl: Wertebereich TRUE/FALSE.

byRange: Wertebereich 000b..111b.

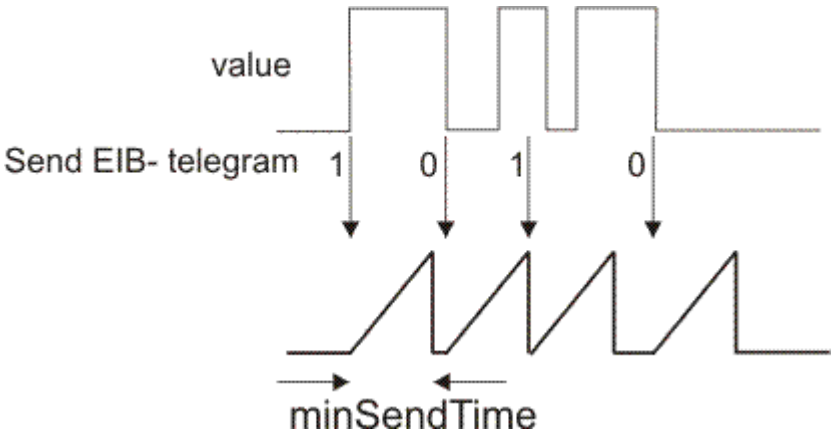
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

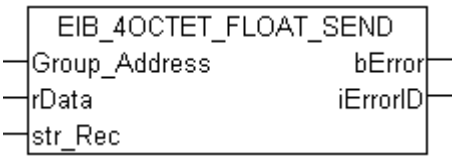
```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.26 EIB_4OCTET_FLOAT_SEND



Dieser Funktionsbaustein sendet einen 4 Byte Float EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 REAL-Wert zur Verfügung. Die Daten werden nur bei Änderung übertragen. Ändert sich der Wert innerhalb einer Sekunde erneut, werden erst nach Ablauf von einer Sekunde neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
rData          : REAL;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

rData: Der Daten-Wert in REAL, wird automatisch in einen EIB 2OCTET FLOAT-Wert gewandelt.

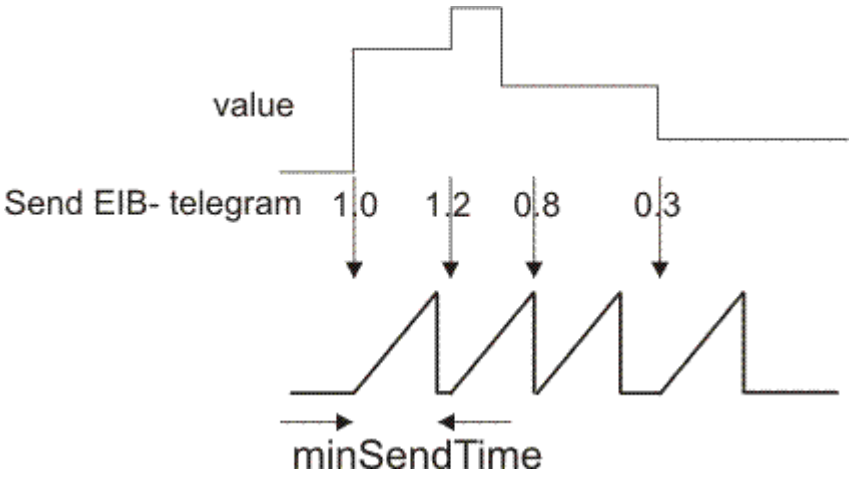
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

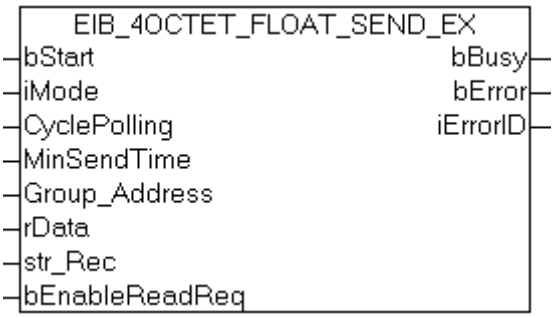
```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.27 EIB_4OCTET_FLOAT_SEND_EX



Dieser Funktionsbaustein sendet einen 4 Byte Float EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 REAL Wert zur Verfügung. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
CyclePolling    : TIME := t#10m;
MinSendTime     : TIME := t#1s;
Group_Address   : EIB_GROUP_ADDR;
rData           : REAL;
str_Rec         : EIB_REC;
bEnableReadReq : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*)

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

rData: Der Daten-Wert in REAL, wird automatisch in einen EIB 2OCTET FLOAT-Wert gewandelt.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

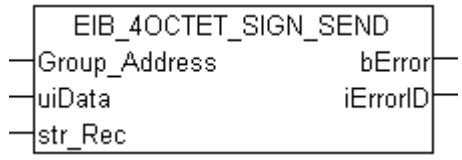
```
bBusy           : BOOL;
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis bBusy wieder auf FALSE ist.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.28 EIB_4OCTET_SIGN_SEND



Dieser Funktionsbaustein sendet einen 4 Byte Sign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 DINT-Wert zur Verfügung. Die Daten werden nur bei Änderung übertragen. Ändert sich der Wert innerhalb einer Sekunde erneut, werden erst nach Ablauf von einer Sekunde neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
uiData          : DINT;
str_Rec         : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

uiData: Der Daten-Wert in DINT, wird automatisch in einen EIB 4OCTET SIGN-Wert gewandelt.

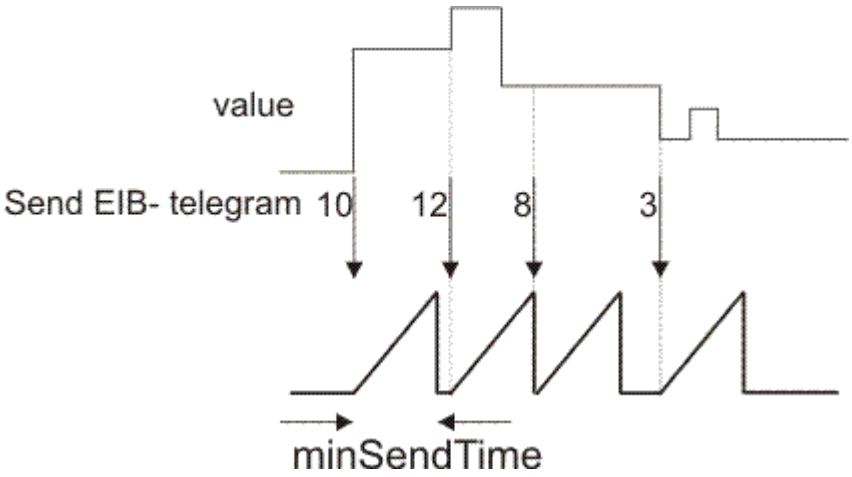
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

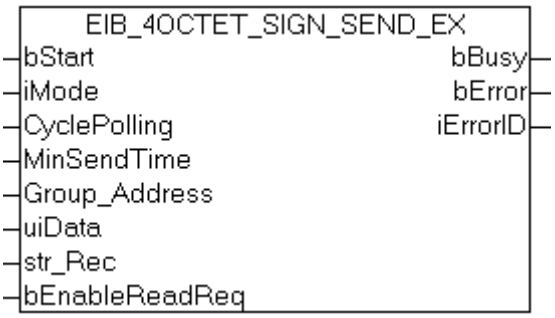
```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.29 EIB_4OCTET_SIGN_SEND_EX



Dieser Funktionsbaustein sendet einen 4 Byte Sign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 DINT-Wert zur Verfügung. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
CyclePolling    : TIME := t#500ms;
MinSendTime     : TIME := t#1s;
Group_Address   : EIB_GROUP_ADDR;
uiData          : DINT;
str_Rec         : EIB_REC;
bEnableReadReq  : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*)

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

uiData: Der Daten-Wert in DINT, wird automatisch in einen EIB 4OCTET SIGN-Wert gewandelt.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

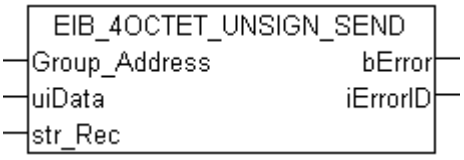
```
bBusy          : BOOL;
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis bBusy wieder auf FALSE ist.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.30 EIB_4OCTET_UNSIGN_SEND



Dieser Funktionsbaustein sendet einen 4 Byte Unsign EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 UDINT-Wert zur Verfügung. Die Daten werden nur bei Änderung übertragen. Ändert sich der Wert innerhalb einer Sekunde erneut, werden erst nach Ablauf von einer Sekunde neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
uiData         : UDINT;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

uiData: Der Daten-Wert in UDINT, wird automatisch in einen EIB 4OCTET SIGN-Wert gewandelt.

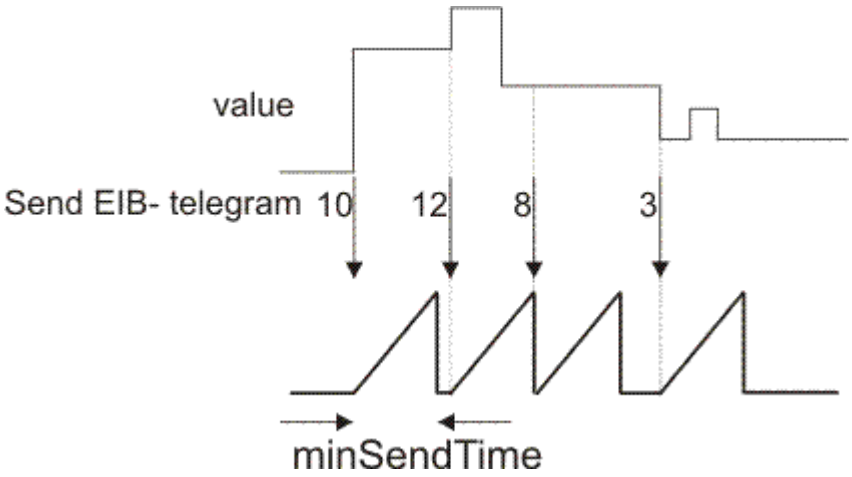
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.31 EIB_8BIT_SIGN_SEND

Dieser Funktionsbaustein sendet einen 8 Bit EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 INT Wert zur Verfügung. Mit dem Scaling_Mode kann der Eingabe-Datenwert skaliert werden. Die Daten werden nur bei Änderung des Datenwertes übertragen. Ändert sich der Wert innerhalb von einer Sekunde erneut, werden erst nach Ablauf von der minSendTime neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
iData           : INT;
Scaling_Mode    : INT;
str_Rec         : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

iData: Daten die versendet werden sollen. Wertebereich abhängig vom *Scaling_Mode*.

Scaling_Mode:
0 - 0...100 [%]
1 - 0...360 [°]
2 - 0...255

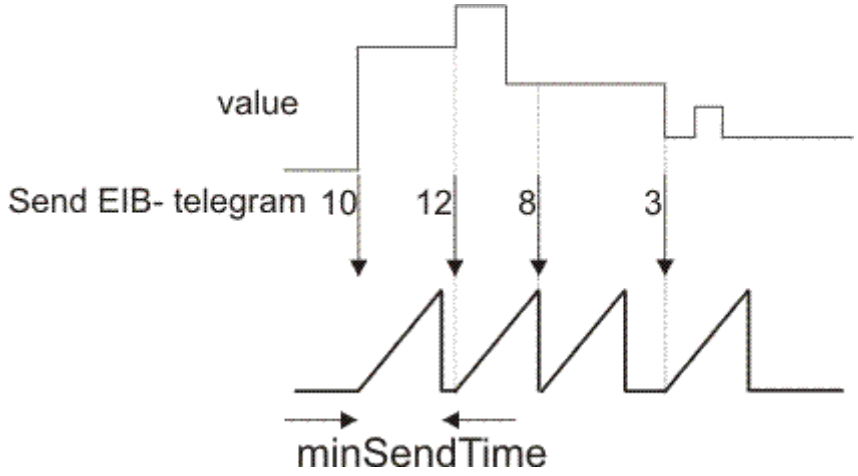
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

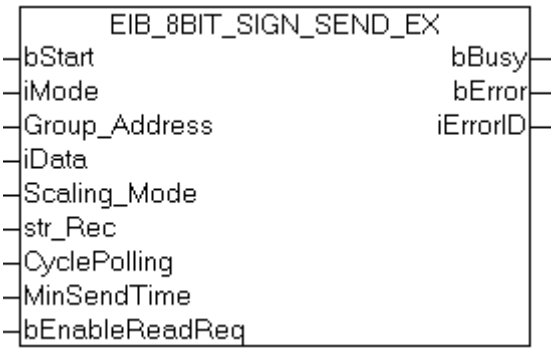
```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.32 EIB_8BIT_SIGN_SEND_EX



Dieser Funktionsbaustein sendet einen 8 Bit EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 INT Wert zur Verfügung. Mit dem *Scaling_Mode* kann der Eingabe-Datenwert skaliert werden. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
Group_Address   : EIB_GROUP_ADDR;
iData           : INT;
Scaling_Mode    : INT;
str_Rec         : EIB_REC;
CyclePolling    : TIME := t#500ms;
MinSendTime     : TIME := t#1s;
bEnableReadReq : BOOL;
```

bStart: Aktiviert den Baustein. Der Baustein fängt an in Abhängigkeit des eingestellten Modus (siehe *iMode*) zu Arbeiten.

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

iData: Daten die versendet werden sollen. Wertebereich abhängig vom *Scaling_Mode*.

Scaling_Mode:
0 - 0...100 [%]
1 - 0...360 [°]
2 - 0...255

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

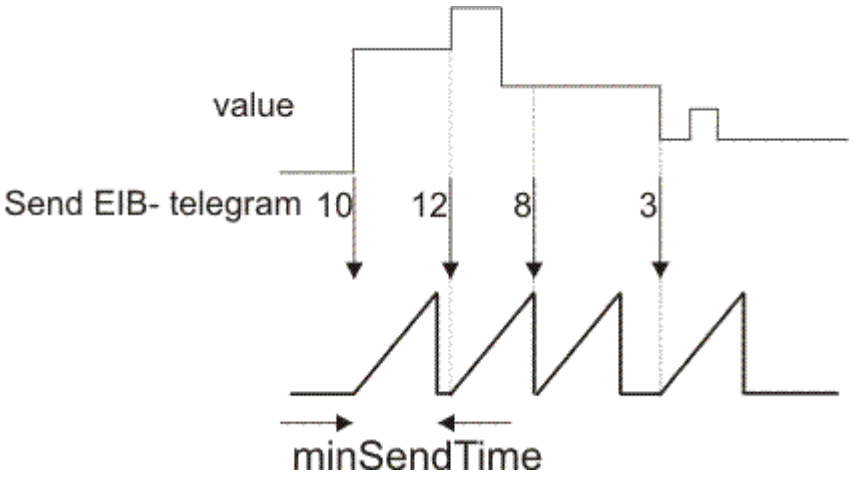
VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

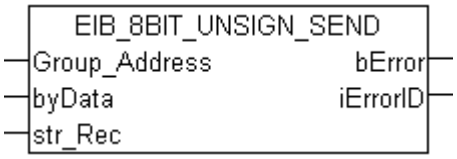
bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis bBusy wieder auf FALSE ist.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.33 EIB_8BIT_UNSIGN_SEND



Dieser Funktionsbaustein sendet einen 8 Bit EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 Byte Wert zur Verfügung. Die Daten werden nur bei Änderung des Datenwertes übertragen. Ändert sich der Wert innerhalb von einer Sekunde erneut, werden erst nach Ablauf von der minSendTime neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
byData          : BYTE;
str_Rec         : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

byData: Daten, die versendet werden sollen. Wertebereich 0x00...0xFF.

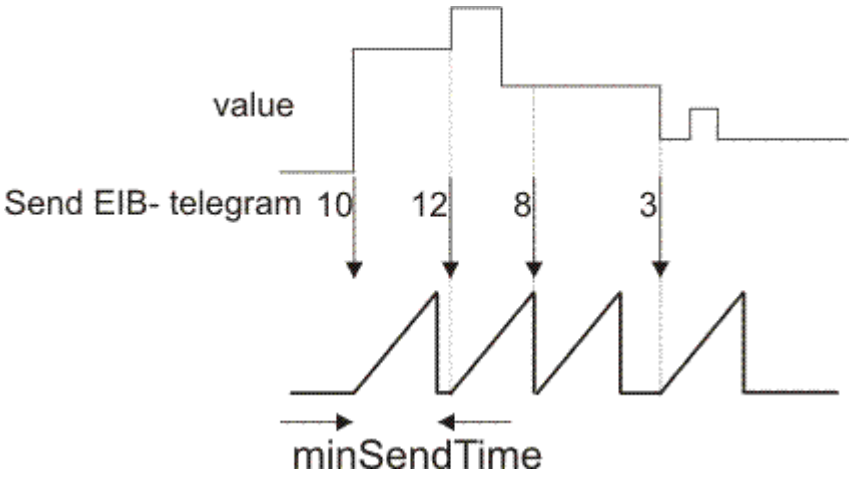
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

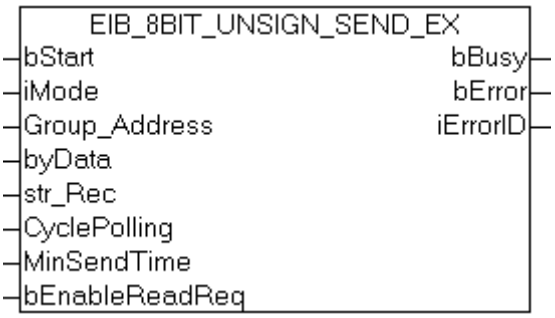
```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.34 EIB_8BIT_UNSIGN_SEND_EX



Dieser Funktionsbaustein sendet einen 8 Bit EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen ein IEC61131-3 Byte Wert zur Verfügung. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
Group_Address   : EIB_GROUP_ADDR;
byData          : BYTE;
str_Rec         : EIB_REC;
CyclePolling    : TIME := t#500ms;
MinSendTime     : TIME := t#1s;
bEnableReadReq  : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*)

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

byData: Daten, die versendet werden sollen. Wertebereich 0x00...0xFF.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

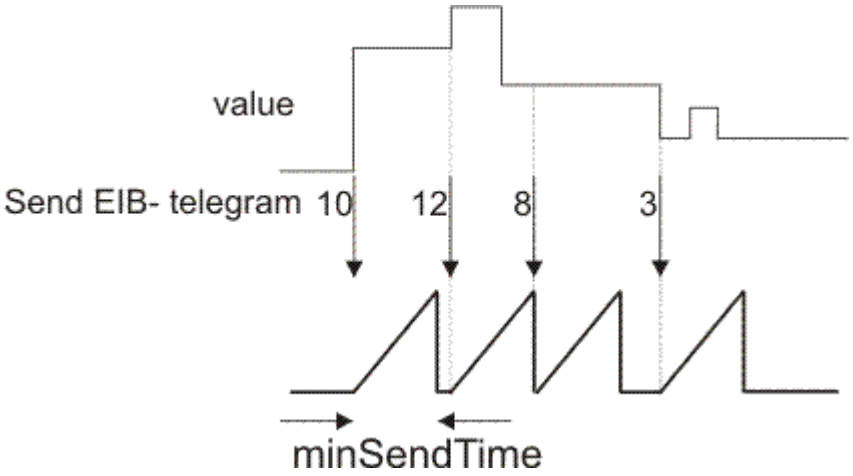
```
bBusy           : BOOL;
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis bBusy wieder auf FALSE ist.

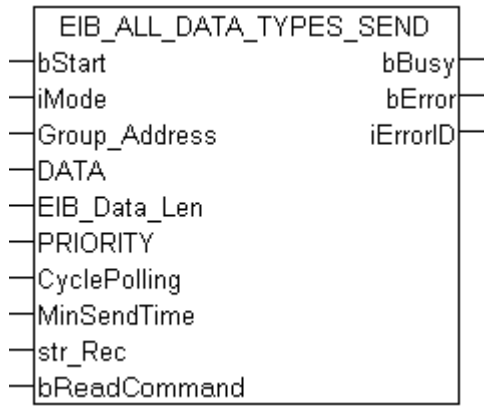
bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

OnChange Mode:



6.3.35 EIB_ALL_DATA_TYPES_SEND



Dieser Funktionsbaustein sendet einen beliebigen EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert steht Ihnen eine IEC61131-3 14 Byte-ARRAY-Variable zur Verfügung. Die Daten werden in Abhängigkeit des eingestellten Modes übertragen.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
Group_Address   : EIB_GROUP_ADDR;
DATA            : ARRAY [1..14] OF OF BYTE;
EIB_Data_Len    : USINT := 1;
PRIORITY        : EIB_PRIORITY := EIB_PRIORITY_LOW;
CyclePolling    : TIME := t#100ms;
MinSendTime     : TIME := t#1s;
str_Rec         : EIB_REC;
bReadCommand    : BOOL;
```

bStart: Ist der Mode auf 0 gestellt, wird ein EIB-Telegramm mit positiver Flanke auf bStart gesendet.

iMode:
0 - Manuell (Abb. 1)
1 - polling (Abb. 2)
2 - OnChange (Abb. 3)

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

DATA: EIB-Datenwerte.

EIB_Data_Len: Länge der EIB-Daten,
EIB Werte >=) 1 Byte die Länge +1 rechnen,
EIB Werte < 1 Byte Länge gleich 1 setzen

PRIORITY: EIB Priorität, Low, High, Alarm.

CyclePolling: Ist der Mode 1 angewählt, wird mit der eingestellten Zeit ein EIB-Telegramm versendet - auch wenn sich die Datenwerte nicht geändert haben.

MinSendTime: Ist der Mode 2 angewählt, werden Daten bei Änderung übertragen Die MinSendTime gibt die minimale Zeit zwischen zwei EIB-Telegrammen an.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bReadCommand: Es kann eine Antwort auf ein EIB READ COMMAND gesendet werden.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bBusy: Ist das Bit gesetzt, so ist der Baustein noch aktiv. Solange das Bit bBusy gesetzt ist, können keine neuen Daten übertragen werden!

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

Übertragungsmodus

Mode 0 Manuell

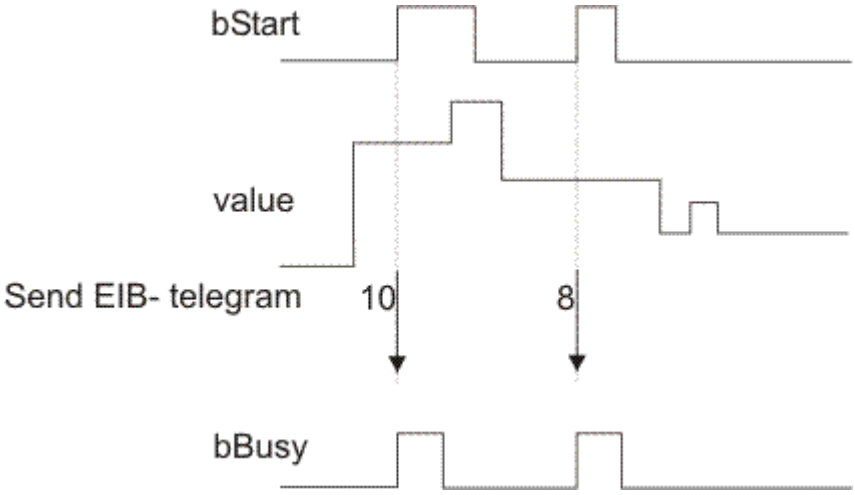


Abbildung 1

Mode 1 Polling

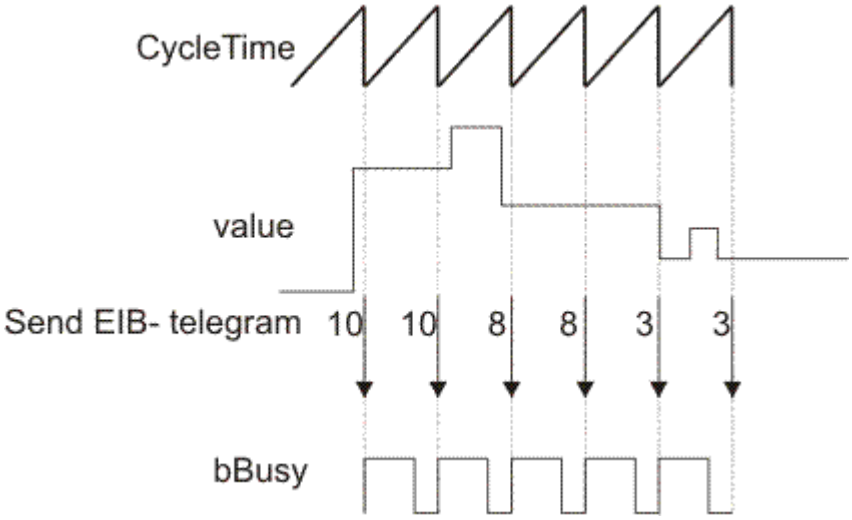


Abbildung 2

Mode 2 OnChange

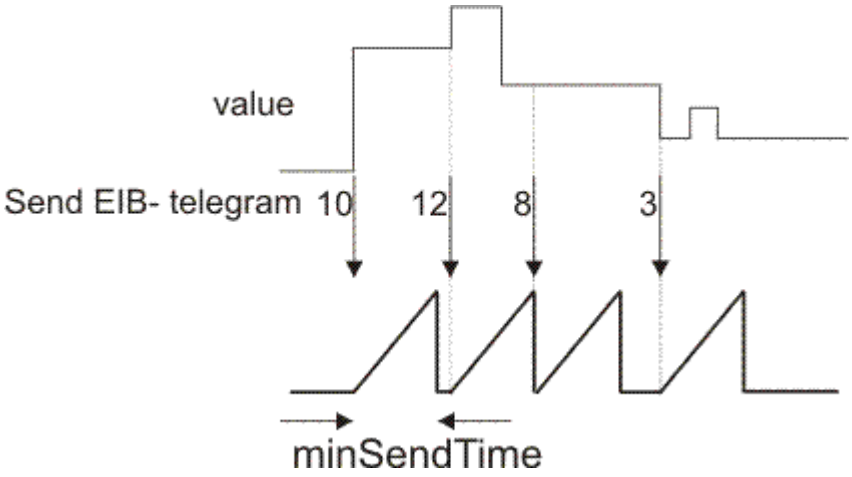
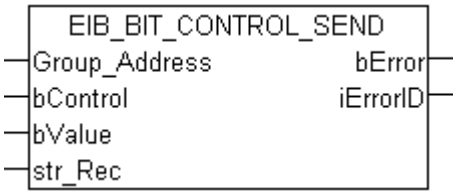


Abbildung 3

6.3.36 EIB_BIT_CONTROL_SEND



Dieser Funktionsbaustein sendet einen 2 Bit EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert stehen Ihnen zwei IEC61131-3 BOOL-Variablen zur Verfügung. Die Daten werden nur bei Änderung einer der beiden Datentypen übertragen. Ändert sich der Wert innerhalb von 200 ms erneut, werden erst nach Ablauf von 200 ms neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
bControl        : BOOL;
bValue          : BOOL;
str_Rec         : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

bControl: Wertebereich TRUE/FALSE.

bValue: Wertebereich TRUE/FALSE.

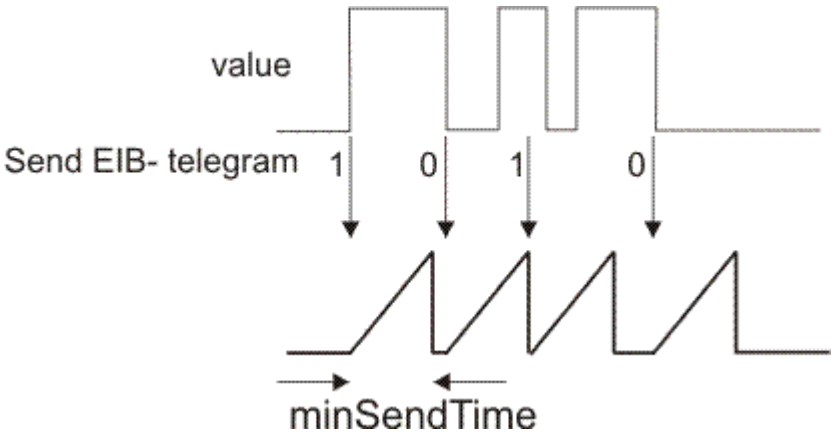
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.37 EIB_BIT_SEND

Dieser Funktionsbaustein sendet einen 1 Bit EIB-Wert zur eingestellte Gruppenadresse. Als Eingabewert steht Ihnen eine IEC61131-3 BOOL-Variable zur Verfügung. Die Daten werden nur bei Änderung des Datenwertes übertragen. Ändert sich der Wert innerhalb von 200 ms erneut, werden erst nach Ablauf von 200 ms neue Daten zum EIB-Teilnehmer gesendet (siehe Abbildung). Ändert sich der Wert innerhalb der "min. Send Time" und fällt dieser Wert auch innerhalb der "min. Send Time" auf den alten schon gesendeten Wert zurück, wird kein neues EIB-Telegramm gesendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
bData          : BOOL;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

bData: Wertebereich TRUE/FALSE.

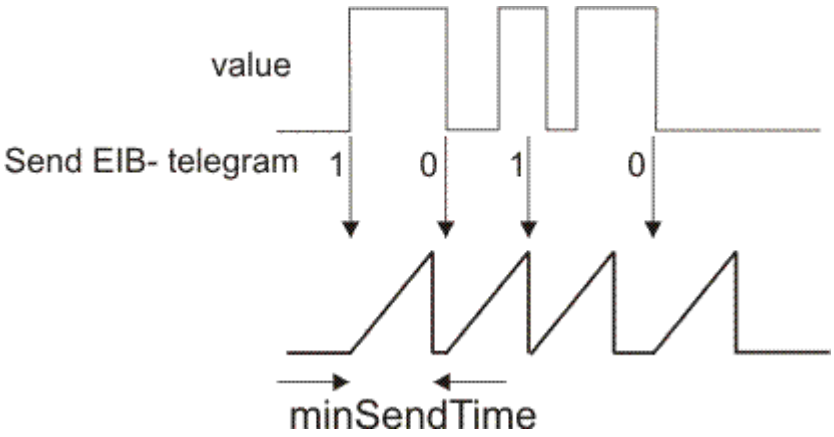
str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

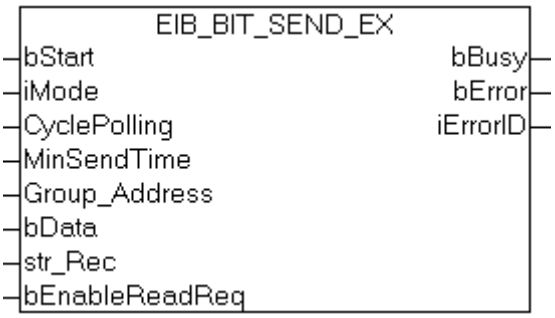
```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.



6.3.38 EIB_BIT_SEND_EX



Dieser Funktionsbaustein sendet einen Bool Wert zur eingestellten Gruppenadresse. In Abhängigkeit des Modus (*iMode*) können die Daten Manuell, Polling oder OnChange gesendet werden.

VAR_INPUT

```
bStart          : BOOL;
iMode           : INT;
CyclePolling    : TIME := t#10s;
MinSendTime     : TIME := t#1s;
Group_Address   : EIB_GROUP_ADDR;
bData           : BOOL;
str_Rec         : EIB_REC;
bEnableReadReq : BOOL;
```

bStart: Aktiviert den Baustein, damit fängt der Baustein an zu Arbeiten in Abhängigkeit des eingestellten Modus (siehe *iMode*)

iMode:
0 - Bei positiver Flanke von *bStart* wird ein EIB-Telegramm gesendet. Ist der Ausgang *bBusy* wieder FALSE, ist der Befehl abgearbeitet.
1 - Polling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet.
2 - OnChange Modus: ist *bStart* auf TRUE, wird bei Änderung der Daten automatisch ein EIB-Telegramm versendet. Mit *MinSendTime* kann der minimale Abstand zwischen zwei EIB Nachrichten parametriert werden, um eine unnötige Belastung des EIB Netzwerkes zu vermeiden.
3 - OnChangePolling Modus: ist *bStart* auf TRUE, werden im Zeitabstand von *CyclePolling* EIB-Telegramme versendet oder automatisch bei Änderung der Daten. Der minimale Abstand zwischen zwei EIB Nachrichten wird mit *MinSendTime* festgelegt.

CyclePolling: Pollzeit für *iMode* = 1 (Polling Mode). Die minimale Zeit ist 200ms.

MinSendTime: Intervallzeit die mindestens vergehen muss bis wieder ein Telegramm im OnChange Mode gesendet wird. Die minimale Zeit ist 200ms.

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

bData: Wertebereich TRUE/FALSE.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

bEnableReadReq: Ermöglicht die Ausführung von Lesebefehlen.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis bBusy wieder auf FALSE ist.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.39 EIB_BIT_SEND_MANUAL

Dieser Funktionsbaustein sendet einen 1 Bit EIB-Wert zur eingestellte Gruppenadresse. Als Eingabewert steht Ihnen eine IEC61131-3 BOOL-Variable zur Verfügung. Die Daten werden bei einer positiven Flanke von *bSend* gesendet. Solange der Baustein aktive ist, ist das *bBusy* gesetzt. *bBusy* wird auf FALSE gesetzt sobald der EIB Befehl gesendet wurde oder ein Fehler vorliegt. Ein Fehler wird mit setzen der *bError* Variable angezeigt. Der Fehlercode ist dann im *iErrorID* angegeben.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
bSend          : BOOL;
bData          : BOOL;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

bSend: Positive Flanke sendet EIB Telegram.

bData: Wertebereich TRUE/FALSE.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

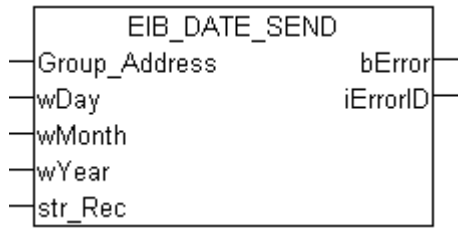
```
bBusy          : BOOL;
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bBusy: Ist der Baustein aktiv ist bBusy TRUE, wenn das EIB Telegramm versendet wurde wird es auf FALSE zurückgesetzt.

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.40 EIB_DATE_SEND



Dieser Funktionsbaustein sendet einen 3 Byte EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert stehen Ihnen drei IEC61131-3 Word-Variablen zur Verfügung. Die Daten werden beim ersten Aufruf des Bausteins und dann alle 5 Minuten erneut versendet.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
wDay           : WORD;
wMonth         : WORD;
wYear          : WORD;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

wDay: Wertebereich 1...31.

wMonth: Wertebereich 1...12.

wYear: Wertebereich 0...99. Wird ein Wert größer 2000 eingegeben, wird automatisch 2000 subtrahiert. So wird zum Beispiel für das Jahr 2005 zum EIB-Knoten nur die 5 übertragen.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

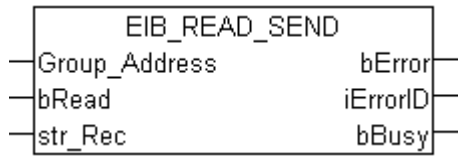
VAR_OUTPUT

```
bError         : BOOL;
iErrorID       : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.41 EIB_READ_SEND



Anwendung

Dieser Funktionsbaustein sendet einen *Read_Group_Req* zur eingestellten Gruppenadresse. Um ein *Read_Group_Res* zu erhalten, muss der Gruppenadressfilter der KL6301 entsprechend parametriert sein.

VAR_INPUT

```
Group_Address  : EIB_GROUP_ADDR;
bRead          : BOOL;
str_Rec        : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

bRead: Positive Flanke startet den Baustein und schickt ein *Read_Group_Req* an den EIB-Teilnehmer.

Um eine Antwort zu erhalten muss die Gruppenadresse im Filter eingetragen sein!

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

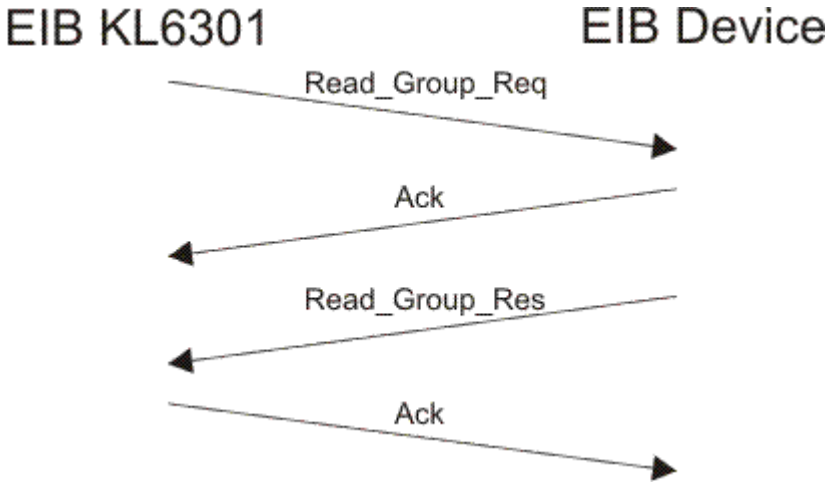
VAR_OUTPUT

```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
bBusy           : BOOL;
```

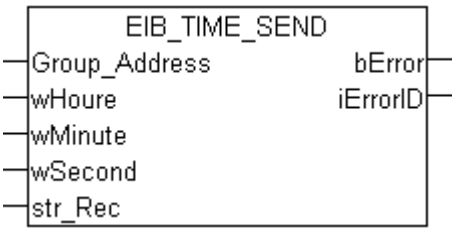
bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

bBusy: Der Baustein ist aktiv. Warten Sie für neue Funktionen, bis bBusy wieder auf FALSE ist.



6.3.42 EIB_TIME_SEND



Anwendung

Dieser Funktionsbaustein sendet einen 3 Byte EIB-Wert zur eingestellten Gruppenadresse. Als Eingabewert stehen Ihnen drei IEC61131-3 Word-Variablen zur Verfügung. Die Daten werden beim ersten Aufruf des Bausteins und dann alle 5 Minuten erneut versendet.

VAR_INPUT

```
Group_Address   : EIB_GROUP_ADDR;
wHoure          : WORD;
wMinute         : WORD;
wSecond         : WORD;
str_Rec         : EIB_REC;
```

Group_Address: Gruppenadresse, zu der die Daten gesendet werden (siehe EIB_GROUP_ADDR [▶ 66]).

wHoure: Wertebereich 0..23.

wMinute: Wertebereich 0..59.

wSecond: Wertebereich 0..59.

str_Rec: Datenstruktur die mit dem Baustein KL6301() [▶ 26] verbunden sein muss (siehe EIB_REC [▶ 67]).

VAR_OUTPUT

```
bError          : BOOL;
iErrorID        : EIB_ERROR_CODE;
```

bError: Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *iErrorID* beschrieben.

iErrorID: Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *bError* TRUE.

6.3.43 Fehlercodes

Wert (hex)	Wert (dez)	Wert (enum)	Beschreibung
0x0000	0	NO_EIB_ERROR	Kein Fehler.
0x0001	1	WRONG_EIB_PHYS_ADDR	Veraltet, wird nicht mehr verwendet.
0x0002	2	WRONG_EIB_GROUP_ADDR	Die Eingangsvariable *EIB_GROUP_FILTER.GROUP_ADDR* ist fehlerhaft. Kontrollieren Sie *GROUP_ADDR* Ihrer Filter. *MAIN* muss kleiner 16 sein, *SUB_MAIN* kleiner 8.
0x0003	3	WRONG_EIB_GROUP_LEN	Die Eingangsvariable *EIB_GROUP_FILTER.GROUP_LEN* ist fehlerhaft. Falsche Länge der Filter, kontrollieren Sie den Mode und die Länge der Filter.
0x0004	4	WRONG_EIB_NO_FILTER	Keine Filter erkannt. Kontrollieren Sie Ihre Filter in *EIB_GROUP_FILTER* und den Mode.
0x0005	5	WRONG_EIB_IDX_RANGE	Die Eingangsvariable *idx* hat einen falschen Wert.
0x000A	10	WRONG_EIB_FIRMWARE	Der Mode wird mit dieser Firmware nicht unterstützt.
0x000B	11	WRONG_EIB_MODE	Nicht unterstützter Modus beim Parametrieren. Kontrollieren Sie *iMode*. Erlaubte Werte sind 0, 1 und 100.
0x000C	12	WRONG_MODE	Die Eingangsvariable *iMode* hat einen falschen Wert.
0x000E	14	WRONG_EIB_FIRMWARE_B1_NECESSARY	Mindestens Firmware B1 oder höher notwendig.
0x000F	15	WRONG_EIB_FIRMWARE_B3_NECESSARY	Mindestens Firmware B3 oder höher notwendig.
0x0014	20	WRONG_EIB_DATA_LEN	Erwartete Datenlänge des EIB Telegramms ist falsch. Telegramm wird verworfen. Kontrollieren Sie die EIB Gruppenadresse und/oder den benutzten Datentyp.

Wert (hex)	Wert (dez)	Wert (enum)	Beschreibung
0x0015	21	ERROR_EIB_SERVICE_NOT_SUPPORT	Dieses EIB Telegramm wird nicht unterstützt.
0x001E	30	KL6301_TP_TOGGLE_ERROR	Klemme reagierte eine Sekunde lang nicht. Kontrollieren Sie die Verbindung zur KL6301. Befindet sich diese noch im Datenaustausch?
0x001F	31	TIME_OUT	Bei der Parametrierung reagierte die Klemme nicht mehr. Kontrollieren Sie die Verbindung zur KL6301.
0x0020	32	KL6301_NO_RESPONSE_FROM_TERMINAL	Keine Verbindung zur KL6301. Entweder Klemme nicht vorhanden oder Mapping fehlerhaft.
0x0028	40	ERROR_SEND_8BIT_WRONG_Scaling_Mode	Falscher oder nicht unterstützter Scaling Mode.
0x0064	100	ERROR_EIB_PHY_ADDR_NOT_SUPPORT	Physikalische Adressierung nicht erlaubt.
0x0065	101	ERROR_EIB_WRITE_DATA	Veraltet. Wird nicht mehr verwendet.
0x0066	102	MONITOR_MODE_LEN_IS_NOT_OK_MUST_0	Für den Monitorbetrieb muss die Länge der Filter 0 sein.
0x0067	103	MONITOR_MODE_ADDR_IS_NOT_OK_MUST_0	Für den Monitorbetrieb müssen die Adressen 0 sein.
0x0068	104	WATCHDOG_ERROR_NO_SEND	Übertragung von Daten nicht möglich. Die Gruppenadresse, an die nicht gesendet werden konnte, befindet sich in der lokalen Variable "NotSendGroup" des Funktionsbausteins KL6301.
0x0BBB	3003	ERROR_EIB_NO_ACK	Kein ACK erhalten.
0xFAFB	64251	ERROR_EIB_NO_COM_TO_TP	Keine Kommunikation mit der EIB Hardware.
0x0FCC	4044	ERROR_TP_TEMP_WARNING	Temperaturüberschreitung in der KL6301.
0x17CC	6092	ERROR_TP_PROTOCOL_ERROR	Protokollfehler auf der EIB Physik.
0x27CC	10188	ERROR_TP_TRANSMITTER_ERROR	Protokollfehler auf der EIB Physik.
0x47CC	18380	ERROR_TP_RECEIVE_ERROR	Protokollfehler auf der EIB Physik.
0x87CC	34764	ERROR_TP_SLAVE_COLLISION	Zu viele Kollisionen auf der EIB Physik. Reduzieren Sie die EIB Last.

6.4 Funktionen

Bausteine	Beschreibung
F_CONV_2GROUP_TO_3GROUP [▶ 64]	Umwandlung einer 2 stufigen Gruppenadresse in eine 3 stufige Gruppenadresse
F_CONV_3GROUP_TO_2GROUP [▶ 64]	Umwandlung einer 3 stufigen Gruppenadresse in eine 2 stufige Gruppenadresse

6.4.1 F_CONV_2GROUP_TO_3GROUP : EIB_GROUP_ADDR

EIB_GROUP_ADDR [▶ 66]

Umwandlung einer 2 stufigen Gruppenadresse in eine 3 stufige Gruppenadresse.

VAR_INPUT

```
IN      : EIB_GROUP_ADDR_2GROUP;
```

IN: 2 stufige Gruppenadresse (siehe EIB_GROUP_ADDR_2GROUP [▶ 66]).

6.4.2 F_CONV_3GROUP_TO_2GROUP : EIB_GROUP_ADDR_2GROUP

EIB_GROUP_ADDR_2GROUP [▶ 66]

Umwandlung einer 3 stufigen Gruppenadresse in eine 2 stufige Gruppenadresse.

VAR_INPUT

```
IN      : EIB_GROUP_ADDR;
```

IN: 3 stufige Gruppenadresse (siehe EIB_GROUP_ADDR [▶ 66]).

6.5 Datentypen

Datentypen	Beschreibung
EIB_ERROR_CODE [▶ 64]	Fehlermeldungen
EIB_PRIORITY [▶ 66]	Priorität des EIB-Telegramms

Datentypen	Beschreibung
EIB_GROUP_ADDR [▶ 66]	3-stufige Gruppenadresse
EIB_GROUP_ADDR_2GROUP [▶ 66]	2-stufige Gruppenadresse
EIB_GROUP_FILTER [▶ 67]	Gruppenfilter
EIB_PHYS_ADDR [▶ 67]	Physikalische Adresse
EIB_REC [▶ 67]	Verbindet die Sende- und Empfangs-Bausteine mit dem Baustein *KL6301*

6.5.1 EIB_ERROR_CODE

Fehlermeldungen der Bibliothek.

```
TYPE EIB_ERROR_CODE :
(
  NO_EIB_ERROR                    := 0,
  WRONG_EIB_PHYS_ADDR             := 1,
  WRONG_EIB_GROUP_ADDR            := 2,
  WRONG_EIB_GROUP_LEN             := 3,
  WRONG_EIB_NO_FILTER             := 4,
  WRONG_EIB_IDX_RANGE             := 5,
  WRONG_EIB_FIRMWARE              := 10,
```

```
  WRONG_EIB_MODE                    := 11,
  WRONG_MODE                        := 12,
  WRONG_EIB_FIRMWARE_B1_NECESSARY   := 14,
  WRONG_EIB_FIRMWARE_B3_NECESSARY   := 15,
  WRONG_EIB_DATA_LEN                := 20,
  ERROR_EIB_SERVICE_NOT_SUPPORT     := 21,
  KL6301_TP_TOGGLE_ERROR            := 30,
  TIME_OUT                          := 31,
  KL6301_NO_RESPONSE_FROM_TERMINAL  := 32,
  ERROR_SEND_8BIT_WRONG_Scaling_Mode := 40,
  ERROR_EIB_PHY_ADDR_NOT_SUPPORT    := 100,
  ERROR_EIB_WRITE_DATA              := 101,
  MONITOR_MODE_LEN_IS_NOT_OK_MUST_0 := 102,
  MONITOR_MODE_ADDR_IS_NOT_OK_MUST_0 := 103,
  WATCHDOG_ERROR_NO_SEND            := 104,
  ERROR_EIB_NO_ACK                  := 16#0BBB,
  ERROR_EIB_NO_COM_TO_TP            := 16#FAFB,
  ERROR_TP_TEMP_WARNING             := 16#0FCC,
  ERROR_TP_PROTOCOL_ERROR           := 16#17CC,
  ERROR_TP_TRANSMITTER_ERROR        := 16#27CC,
  ERROR_TP_RECEIVE_ERROR            := 16#47CC,
  ERROR_TP_SLAVE_COLLISION          := 16#87CC
)
END_TYPE
```

NO_EIB_ERROR**:** Kein Fehler.

WRONG_EIB_PHYS_ADDR**:** Veraltet, wird nicht mehr verwendet.

WRONG_EIB_GROUP_ADDR**:** Die Eingangsvariable *EIB_GROUP_FILTER.GROUP_ADDR* ist fehlerhaft. Kontrollieren Sie *GROUP_ADDR* Ihrer Filter. *MAIN* muss kleiner 16 sein, *SUB_MAIN* kleiner 8.

WRONG_EIB_GROUP_LEN**:** Die Input Variable *EIB_GROUP_FILTER.GROUP_LEN* ist fehlerhaft. Falsche Länge der Filter, kontrollieren Sie den Mode und die Länge der Filter.

WRONG_EIB_NO_FILTER**:** Keine Filter erkannt. Kontrollieren Sie Ihre Filter in *EIB_GROUP_FILTER* und den Mode.

WRONG_EIB_IDX_RANGE: Die Eingangsvariable *idx* hat einen falschen Wert.

WRONG_EIB_FIRMWARE**:** Der Mode wird mit dieser Firmware nicht unterstützt.

WRONG_EIB_MODE**:** Nicht unterstützter Modus beim Parametrieren. Kontrollieren Sie *iMode*. Erlaubte Werte sind 0, 1 und 100.

WRONG_MODE**:** Die Eingangsvariable *iMode* hat einen falschen Wert.

WRONG_EIB_FIRMWARE_B1_NECESSARY**:** Mindestens Firmware B1 oder höher notwendig.

WRONG_EIB_FIRMWARE_B3_NECESSARY**:** Mindestens Firmware B3 oder höher notwendig.

WRONG_EIB_DATA_LEN**:** Erwartete Datenlänge des EIB Telegramms ist falsch. Telegramm wird verworfen. Kontrollieren Sie die EIB Gruppenadresse und/oder den benutzten Datentyp.

ERROR_EIB_SERVICE_NOT_SUPPORT**:** Dieses EIB Telegramm wird nicht unterstützt.

KL6301_TP_TOGGLE_ERROR**:** Klemme reagierte eine Sekunde lang nicht. Kontrollieren Sie die Verbindung zur KL6301. Befindet sich diese noch im Datenaustausch?

TIME_OUT**:** Bei der Parametrierung reagierte die Klemme nicht mehr. Kontrollieren Sie die Verbindung zur KL6301.

KL6301_NO_RESPONSE_FROM_TERMINAL**:** Keine Verbindung zur KL6301. Entweder Klemme nicht vorhanden oder Mapping fehlerhaft.

ERROR_SEND_8BIT_WRONG_Scaling_Mode**:** Falscher oder nicht unterstützter Scaling Mode.

ERROR_EIB_PHY_ADDR_NOT_SUPPORT**:** Physikalische Adressierung nicht erlaubt.

ERROR_EIB_WRITE_DATA**:** Veraltet. Wird nicht mehr verwendet.

MONITOR_MODE_LEN_IS_NOT_OK_MUST_0**:** Für den Monitorbetrieb muss die Länge der Filter 0 sein.

MONITOR_MODE_ADDR_IS_NOT_OK_MUST_0: Für den Monitorbetrieb müssen die Adressen 0 sein.

WATCHDOG_ERROR_NO_SEND: Übertragung von Daten nicht möglich. Die Gruppenadresse, an die nicht gesendet werden konnte, befindet sich in der lokalen Variable "NotSendGroup" des Funktionsbausteins KL6301.

ERROR_EIB_NO_ACK: Kein ACK erhalten.

ERROR_EIB_NO_COM_TO_TP: Keine Kommunikation mit der EIB Hardware.

ERROR_TP_TEMP_WARNING: Temperaturüberschreitung in der KL6301.

ERROR_TP_PROTOCOL_ERROR: Protokollfehler auf der EIB Physik.

ERROR_TP_TRANSMITTER_ERROR: Protokollfehler auf der EIB Physik.

ERROR_TP_RECEIVE_ERROR: Protokollfehler auf der EIB Physik.

ERROR_TP_SLAVE_COLLISION: Zu viele Kollisionen auf der EIB Physik. Reduzieren Sie die EIB Last.

6.5.2 EIB_PRIORITY

Priorität des EIB-Telegrams.

```
TYPE EIB_PRIORITY :
(
  EIB_PRIORITY_LOW    := 1,
  EIB_PRIORITY_HIGH   := 2,
  EIB_PRIORITY_ALARM  := 3,
)
END_TYPE
```

EIB_PRIORITY_LOW: Priorität niedrig.

EIB_PRIORITY_HIGH: Priorität hoch.

EIB_PRIORITY_ALARM: Priorität Alarm.

6.5.3 EIB_GROUP_ADDR

3 stufige Gruppenadresse.

```
TYPE EIB_GROUP_ADDR :
STRUCT
  MAIN      : BYTE;
  SUB_MAIN  : BYTE;
  NUMBER    : BYTE;
END_STRUCT
END_TYPE
```

MAIN: Hauptgruppe (Wertebereich 0..31)

SUB_MAIN: Mittelgruppe (Wertebereich 0..7)

NUMBER: Untergruppe (Wertebereich 0..255)

6.5.4 EIB_GROUP_ADDR_2GROUP

2 stufige Gruppenadresse.

```
TYPE EIB_GROUP_ADDR_2GROUP :
STRUCT
  MAIN      : BYTE;
  SUB_MAIN  : WORD;
END_STRUCT
END_TYPE
```

MAIN: Hauptgruppe (Wertebereiche 0..15).

SUB_MAIN: Untergruppe (Wertebereiche 0..2048).

6.5.5 EIB_GROUP_FILTER

Gruppenfilter.

```
TYPE EIB_GROUP_FILTER :
STRUCT
  GROUP_ADDR  : EIB_GROUP_ADDR;
  GROUP_LEN   : WORD;
END_STRUCT
END_TYPE
```

GROUP_ADDR: Gruppenadresse (siehe EIB_GROUP_ADDR [▶ 66]).

GROUP_LEN: iMode 0 - 0..63. iMode 1 - 0..31.

6.5.6 EIB_PHYS_ADDR

Physikalische Adresse.

```
TYPE EIB_PHYS_ADDR :
STRUCT
  Area    : BYTE := 1;
  Line    : BYTE := 2;
  Device  : BYTE := 3;
END_STRUCT
END_TYPE
```

Area: 0..15.

Line: 0..15.

Device: 0..255.

6.5.7 EIB_REC

Verbindet die Sende- und Empfangs-Bausteine mit dem Baustein *KL6301*.

```
TYPE EIB_REC :
STRUCT
  Rec_Group       : EIB_GROUP_ADDR;
  Rec_Len         : INT;
  Rec_Idx         : INT := 1;
  Rec_Data        : ARRAY[1..15] OF BYTE;
  Rec_bWriteBusy  : BOOL;
  Rec_bReadBusy   : BOOL;
  Rec_bReady      : BOOL;
  Rec_bError      : BOOL;
  Rec_iErrorID    : EIB_Error_Code;
  pStr_Send       : DWORD;
  Rec_Data_rec    : BOOL;
  Rec_Typ         : EIB_Read_Typ;
END_STRUCT
END_TYPE
```

Rec_Group: Gruppenadresse (siehe EIB_GROUP_ADDR [▶ 66]).

Rec_Len: Länge.

Rec_Idx: Index.

Rec_Data: Datenbytes.

Rec_bWriteBusy: Daten werden geschrieben.

Rec_bReadBusy: Daten werden gelesen.

Rec_bReady: Fertig.

Rec_bError: Ist TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *Rec_iErrorID* beschrieben.

Rec_iErrorID: Enthält im Fehlerfall einen Fehlercode (siehe EIB_ERROR_CODE [▶ 64]). Gleichzeitig wird *Rec_bError* TRUE.

pStr_Send: Pointer auf die zu sendenden Daten.

Rec_Data_rec: Signalisiert Datenempfang.

Rec_Typ: Typ des Telegramms.

7 Anhang

7.1 Beispiele

Voraussetzungen

Beispiel	Beschreibung
https://infosys.beckhoff.com/content/1031/tcplclibeib/Resources/11993063051.zip	TwinCAT-PLC-Projekt für die KL6301.
https://infosys.beckhoff.com/content/1031/tcplclibeib/Resources/11993064459.zip	Beispiel für einen Busklemmen-Controller der BCxx00-Serie.

7.2 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

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

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

Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Deutschland

Telefon:	+49 5246 963-0
E-Mail:	info@beckhoff.com
Internet:	www.beckhoff.com

Mehr Informationen:
www.beckhoff.de/tx1200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com