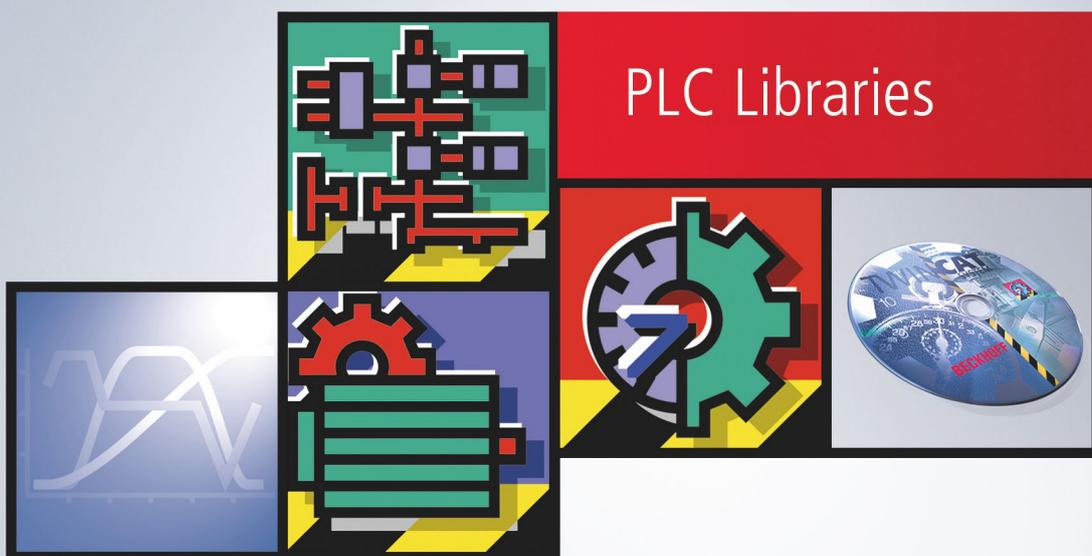


Handbuch | DE

TX1200

TwinCAT 2 | PLC-Bibliothek: COMlibBC



Inhaltsverzeichnis

1	Vorwort.....	5
1.1	Hinweise zur Dokumentation	5
1.2	Sicherheitshinweise	6
1.3	Hinweise zur Informationssicherheit	7
2	Übersicht.....	8
3	Unterstützte Hardware	9
4	Kommunikationskonzept.....	11
5	Datenstrukturen	13
5.1	ComBuffer	13
5.2	Datenstrukturen für serielle Busklemme KL6xxx im 3-Byte Modus	13
5.3	Datenstrukturen für serielle Busklemme KL6xxx im 5-Byte Modus	13
6	Funktionsbausteine	15
6.1	ReceiveByte	15
6.2	SendByte.....	15
6.3	ClearComBuffer	16
6.4	KL6Init	16
6.5	Hintergrundkommunikation	17
6.5.1	KL6ControlBC	17
6.5.2	KL6Control5B.....	17
7	Einbinden in ein SPS-Programm	19
7.1	Installation	19
7.2	Anpassungen an den 5-Byte Modus	19
7.3	Globale Variablen.....	19
7.4	Hintergrundkommunikation	19
7.5	Senden und Empfangen	20
8	Beispielkonfiguration.....	21
9	Fehlercodes	22

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 veröffentlichte 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®

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

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.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig



Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

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 Übersicht

Die TwinCAT SPS Bibliothek COMlibBC bietet Funktionsbausteine und Datenstrukturen für die serielle Datenkommunikation mit den Beckhoff Buscontrollern BCxxxx. Die COMlibBC unterstützt die seriellen Beckhoff Busklemmen KL6xxx.

Die COMlibBC bietet Bausteine für byteweises Senden und Empfangen von Daten. Alle Bausteine sind mehrfach instanzierbar, so dass mehrere serielle Schnittstellen parallel bedient werden können.

Die seriellen Busklemmen sind in einer 3-Byte und in einer 5-Byte Version verfügbar. Um den Speicher des Buscontrollers BCxxxx nicht unnötig zu belasten, steht für den 5-Byte Modus eine modifizierte Bibliothek ComLibBC5B zur Verfügung. Setzen Sie also je nach verwendeter Klemmenhardware eine der beiden Bibliotheken ein. Im Normalfall wird ab Werk die 3-Byte Version der seriellen Busklemme ausgeliefert.

i Aufgrund verschiedener Versionen der Buscontroller BCxxxx ergibt sich ein unterschiedliches Speicher-Mapping der seriellen Busklemmen im 3-Byte Modus. Aus diesem Grunde wird in Zukunft nur noch der 5-Byte Modus unterstützt. Verwenden Sie daher nach Möglichkeit nur Klemmen im 5-Byte Modus, die als KL6xxx-0020 zu bestellen sind.

Ab der Version 2.0 der Kommunikationsbibliothek wurden die Versionen für den PC und die Buscontroller BC vereinheitlicht. Es wird daher auf die [Dokumentation für PC](#) verwiesen.

3 Unterstützte Hardware

Serielle Busklemme

KL6xxx im 3-Byte Modus

Die serielle Beckhoff Busklemme wird in der ausgelieferten Standardausführung im 3-Byte Modus betrieben. D. h. es können in einem Bustelegramm 3 Datenbytes zur Klemme übertragen und von ihr empfangen werden. Da für jeden Datenaustausch zwischen SPS und Busklemme 3 SPS-Zyklen notwendig sind, kann effektiv in jedem Zyklus ein Byte übertragen werden.

Die maximale effektive Datenübertragungsrate, Bps ist abhängig von der Zykluszeit T der SPS, und der Anzahl der Bits pro übertragenem Datenbyte LB:

$$\text{Bps} = \text{LB} / T$$

$$\text{LB} = 1 \text{ Startbit} + n \text{ Datenbits} + p \text{ Paritätsbits} + m \text{ Stopbits}$$

Die maximale effektive Datenübertragungsrate wird nach oben durch die in der Busklemme programmierte physikalische Baudrate begrenzt.



Aufgrund verschiedener Versionen der Buscontroller BCxxxx ergibt sich ein unterschiedliches Speicher-Mapping der seriellen Busklemmen im 3-Byte Modus. Aus diesem Grunde unterstützt diese Library in Zukunft nur noch der 5-Byte Modus. Verwenden Sie daher nach Möglichkeit nur Klemmen im 5-Byte Modus, die als KL6xxx-0020 zu bestellen sind.

KL6xxx im 5-Byte Modus

Die serielle Busklemme kann offline durch ein Konfigurationsprogramm (Beckhoff KS2000) umprogrammiert werden, so daß im 5-Byte Modus jeweils 5 Datenbyte von und zur Klemme übertragen werden können. Dabei sind ebenfalls 3 SPS-Zyklen für einen Austausch notwendig. Die effektive Datenrate liegt bei gleicher Zykluszeit der SPS um 5/3 höher als im 3-Byte Modus.

$$\text{Bps} = (\text{LB} * 5/3) / T$$

Die Umprogrammierung der Busklemmen kann nicht zur Laufzeit der SPS erfolgen, da sich der 3-Byte und der 5-Byte Modus im Register-Mapping und in der Konfiguration im TwinCAT System Manager unterscheiden. Zur Zeit wird daher der 5-Byte Betrieb wegen der notwendigen Umprogrammierung nur für zeitkritische Anwendungen empfohlen.

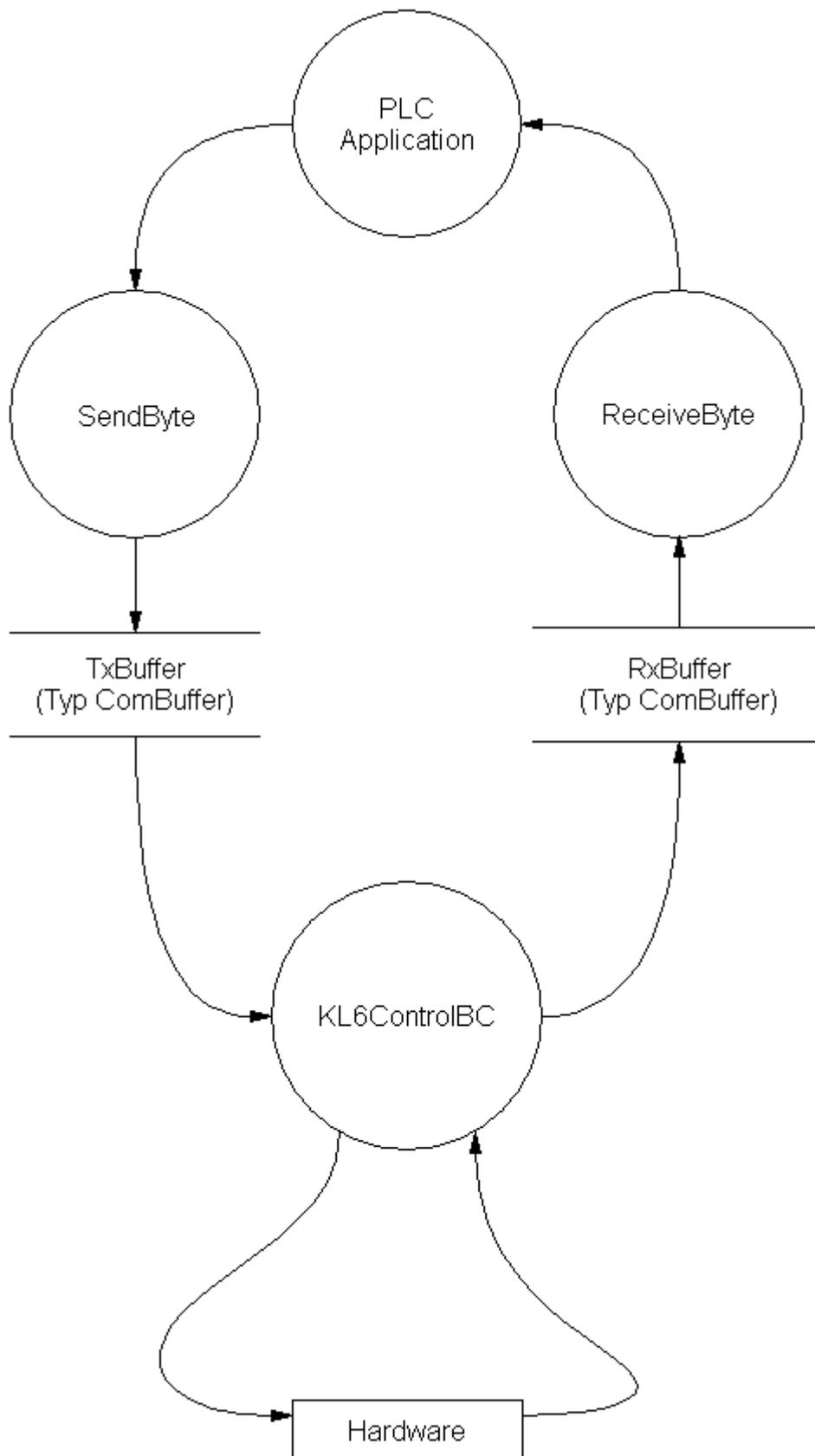
Um den Speicher des Buscontrollers BCxxxx nicht unnötig zu belasten steht für den 5-Byte Modus eine modifizierte Bibliothek ComLibBC5B zur Verfügung.

Serieller Buscontroller BC8x00

Der Buscontroller BC8x00 enthält eine serielle Schnittstelle, die standardmäßig im 5-Byte Modus arbeitet. Die Schnittstelle verhält sich wie eine serielle Busklemme KL6xxx im 5-Byte Modus. Zur Programmierung ist daher die COMlibBC5B einzusetzen.

4 Kommunikationskonzept

Zur Entkopplung des Datentransfers zwischen Speicher und serieller Busklemme werden Datenpuffer vom Typ `ComBuffer` [► 13] verwendet, die asynchron beschrieben und gelesen werden.



Die später beschriebenen Funktionsbausteine zum Empfangen und Senden von Daten (Send Byte [[▶ 15](#)], Receive Byte [[▶ 15](#)] etc.) benutzen nur noch die Datenpuffer zum Datenaustausch und sind damit unabhängig von der verwendeten Hardware. In jedem Fall wird zusätzlich zu den Sende- und Empfangsbausteinen ein Kommunikationsbaustein (KL6ControlBC [[▶ 17](#)]) aufgerufen, der den Datenverkehr zwischen Datenpuffer und Hardware mit maximaler Geschwindigkeit im Hintergrund abwickelt.

5 Datenstrukturen

5.1 ComBuffer

Die Datenstruktur **ComBuffer** ist ein Datenpuffer zur Entkopplung der Hardware-abhängigen Kommunikationsbausteine von den Hardware-unabhängigen Bausteinen. Datenpuffer vom Typ ComBuffer werden vom Anwender niemals direkt beschrieben oder gelesen, sondern dienen nur als Zwischenspeicher für die Kommunikationsbausteine.

```

TYPE ComBuffer
STRUCT
    Buffer      : ARRAY[0..300] OF BYTE;
    RdIdx      : INT;
    WrIdx      : INT;
    Count      : INT; (* Anzahl der Zeichen im Ringpuffer *)
    FreeByte   : INT; (* Anzahl freie Plätze im Ringpuffer *)
    Error      : INT; (* Fehlercode der Schnittstelle *)
    blocked    : BOOL;
END_STRUCT
END_TYPE

```

5.2 Datenstrukturen für serielle Busklemme KL6xxx im 3-Byte Modus

Jede serielle Busklemme benötigt zum Datenaustausch über den I/O-Bus je eine Variable des Typs KL6inDataBC und KL6outDataBC. Diese Variablen werden im Speicherabbild des Buscontrollers auf eine feste Adresse gelegt, die dem Mapping der seriellen Busklemme entspricht.

KL6inDataBC

```

TYPE KL6inDataBC
STRUCT
    SerStatus : BYTE;
    D         : ARRAY[0..2] OF BYTE;
END_STRUCT
END_TYPE

```

KL6outDataBC

```

TYPE KL6outDataBC
STRUCT
    SerCtrl : BYTE;
    D       : ARRAY[0..2] OF BYTE;
END_STRUCT
END_TYPE

```

5.3 Datenstrukturen für serielle Busklemme KL6xxx im 5-Byte Modus

Jede serielle Busklemme benötigt zum Datenaustausch über den I/O-Bus je eine Variable des Typs KL6inData5B und KL6outData5B. Diese Variablen werden im Speicherabbild des Buscontrollers auf eine feste Adresse gelegt, die dem Mapping der seriellen Busklemme entspricht.

KL6inData5B

```

TYPE KL6inData5B
STRUCT
    Status : BYTE;
    D      : ARRAY[0..4] OF BYTE;
END_STRUCT
END_TYPE

```

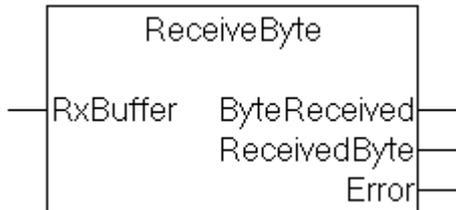
KL6outData5B

```
TYPE KL6outData5B
STRUCT
  Ctrl : BYTE;
  D    : ARRAY[0..4] OF BYTE;
END_STRUCT
END_TYPE
```

Um den Speicher des Buscontrollers BCxxxx nicht unnötig zu belasten, wird der 5-Byte Modus nur durch die modifizierte Bibliothek ComLibBC5B unterstützt.

6 Funktionsbausteine

6.1 ReceiveByte



Interface

```

VAR_OUTPUT
  ByteReceived : BOOL;
  ReceivedByte : BYTE;
  Error       : INT;
END_VAR
VAR_IN_OUT
  RxBuffer    : ComBuffer;
END_VAR

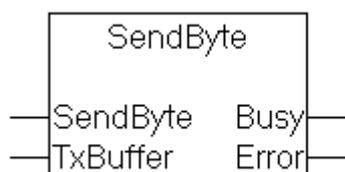
```

[ComBuffer](#) [► 13]

Beschreibung

Der Baustein ReceiveByte empfängt ein einzelnes Zeichen von der mit der Eingangsvariablen RxBuffer korrespondierenden Schnittstelle. Ist nach dem Aufruf ByteReceived=TRUE, dann steht in der Ausgangsvariablen ReceivedByte das empfangene Datenbyte zur Verfügung. Anderenfalls wurden keine Daten empfangen.

6.2 SendByte



Interface

```

VAR_INPUT
  SendByte : BYTE;
END_VAR
VAR_OUTPUT
  Busy     : BOOL;
  Error    : INT;
END_VAR
VAR_IN_OUT
  TxBuffer : ComBuffer;
END_VAR

```

[ComBuffer](#) [► 13]

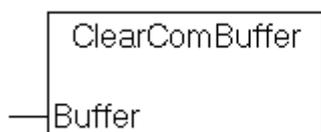
Beschreibung

SendByte sendet ein einzelnes Zeichen an die mit der Eingangsvariablen **TxBuffer** korrespondierenden Schnittstelle. Solange der Ausgang **Busy**=TRUE ist, wurde das Senden noch nicht abgeschlossen. Das Zeichen wurde erfolgreich gesendet, sobald **Busy**=FALSE und **Error**=0 ist.



Solange der Sendedatenpuffer noch Daten aufnehmen kann, können in einem SPS-Zyklus mehrere Zeichen gesendet werden. Das ist jedoch nur sinnvoll, wenn die gepufferten Zeichen durch eine schnellere Kommunikations-Task zur Hardware übertragen werden.

6.3 ClearComBuffer



Interface

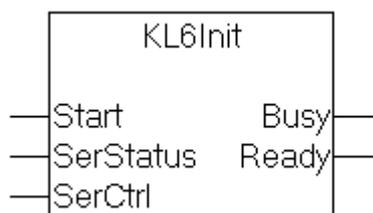
```
VAR_IN_OUT
  Buffer : ComBuffer;
END_VAR
```

[ComBuffer \[► 13\]](#)

Beschreibung

Der SPS-interne Kommunikationspuffer **Buffer** wird gelöscht.

6.4 KL6Init



Interface

```
VAR_INPUT
  Start      : BOOL;
  SerStatus  : BYTE;
END_VAR
VAR_OUTPUT
  Busy       : BOOL;
  Ready      : BOOL;
END_VAR
```

Beschreibung

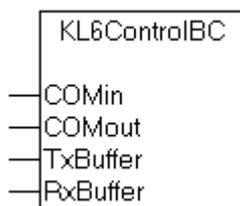
KL6Init initialisiert die serielle Busklemme KL6xxx ohne deren Konfiguration zu verändern. Es werden dadurch die Klemmen-internen Puffer gelöscht.

Die Eingangsdaten **SerStatus** und **SerCtrl** entsprechen den mit der Klemme im TwinCAT SystemManager verknüpften Daten. Sie sind in den Datenstrukturen [KL6inDataBC \[► 13\]](#) bzw. [KL6outDataBC \[► 13\]](#) der COMlib definiert.

Die Klemme wurde nach dem Startsignal **Start** erfolgreich initialisiert, sobald Busy=FALSE und Ready=TRUE ist.

6.5 Hintergrundkommunikation

6.5.1 KL6ControlBC



Interface

```

VAR_INPUT
    COMin      : KL6inDataBC;
END_VAR
VAR_IN_OUT
    COMout     : KL6outDataBC;
    TxBuffer   : ComBuffer;
    RxBuffer   : ComBuffer;
END_VAR
  
```

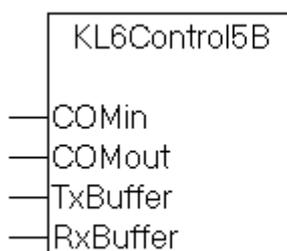
Beschreibung

Der Funktionsbaustein KL6ControlBC kontrolliert die Datenübertragung zwischen den SPS-internen Datenpuffern TxBuffer und RxBuffer und der Hardware. Die Datenstrukturen KL6inDataBC und KL6outDataBC sind global deklariert und werden im Speicherabbild an eine feste Adresse gelegt. KL6ControlBC wird unabhängig von der Absicht Daten zu senden oder zu empfangen permanent aufgerufen.

Sehen Sie dazu auch

- ▣ Datenstrukturen für serielle Busklemme KL6xxx im 3-Byte Modus [► 13]
- ▣ ComBuffer [► 13]

6.5.2 KL6Control5B



Interface

```
VAR_INPUT
  COMin      : KL6inData5B;
END_VAR
VAR_IN_OUT
  COMout     : KL6outData5B;
  TxBuffer   : ComBuffer;
  RxBuffer   : ComBuffer;
END_VAR
```

Beschreibung

KL6Control5B kontrolliert analog zum Baustein [KL6ControlBC](#) [► 17] die Datenübertragung zwischen den SPS-internen Datenpuffern **TxBuffer** und **RxBuffer** und der Hardware, hier jedoch für die serielle Busklemme im 5-Byte Modus

Um den Speicher des Buscontrollers BCxxxx nicht unnötig zu belasten, wird der 5-Byte Modus nur durch die modifizierte Bibliothek ComLibBC5B unterstützt.

Sehen Sie dazu auch

- 📖 Datenstrukturen für serielle Busklemme KL6xxx im 5-Byte Modus [► 13]
- 📖 ComBuffer [► 13]

7 Einbinden in ein SPS-Programm

7.1 Installation

Zur Installation werden die Bibliotheken COMlibBC.LB6 und COMlibBC5b.LB6 in das TwinCAT Verzeichnis TwinCAT\PLC\LIB kopiert.

Das beiliegende Testprogramm kopieren Sie in ein beliebiges Projektverzeichnis, z. B. nach TwinCAT\PLC.

- <https://infosys.beckhoff.com/content/1031/tcplclibserialcombc/Resources/12264207755.zip>

Bibliotheken einbinden

Legen Sie ein neues SPS-Projekt mit dem TwinCAT PLC Control an, um die Einbindung der Bibliothek nachzuvollziehen.

Wechseln Sie in die Bibliotheksverwaltung und fügen Sie die Bibliothek ComLibBC.LIB ein.

7.2 Anpassungen an den 5-Byte Modus

Zur Programmierung der seriellen Busklemme im 5-Byte Modus bzw. der seriellen Schnittstelle des Buscontrollers BC8x00 wird alternativ zur Bibliothek ComLibBC die Bibliothek ComLibBC5B verwendet.

Der Funktionsbaustein KL6ControlBC wird dabei durch den Baustein KL6Control5B ersetzt. Weiterhin ersetzen Sie die Datenstrukturen KL6inDataBC durch KL6inData5B bzw. KL6outDataBC durch KL6outData5B.

Zu beachten ist, daß die seriellen Busklemmen KL6xxx ab Werk im 3-Byte Modus ausgeliefert werden und zunächst mit der Konfigurationssoftware KS2000 umprogrammiert werden müssen. Alternativ sind entsprechend programmierte Klemmen als KL6xxx-0020 erhältlich.

7.3 Globale Variablen

Um eine serielle Schnittstelle anzusprechen, sind vier globale Datenstrukturen notwendig. Zwei stellen die Verbindung zur Hardware in Sende- und Empfangsrichtung her. Zusätzlich werden zwei Datenpuffer als Zwischenspeicher benötigt.

```
VAR_GLOBAL
KL6InData AT %IB0 : KL6inDataBC;
KL6OutData AT %QB0 : KL6outDataBC;
RxBuffer      : ComBuffer;
TxBuffer      : ComBuffer;
END_VAR
```

7.4 Hintergrundkommunikation

Die Kommunikation zwischen der seriellen Hardware und dem Datenpuffer vom Typ ComBuffer [► 13] wird mit einem separaten Baustein ComControl abgewickelt.

Siehe auch Kommunikationskonzept [► 11].

```
PROGRAM MAIN
VAR
    ComControl: KL6ControlBC;
    Timer: TON;
    Receive: ReceiveByte;
    Send: SendByte;
    Rb: BYTE;
    Sb: BYTE;
END_VAR
```

```

ComControl( CoMin:=KL6InData,
            CoMout:=KL6OutData,
            TxBuffer:=TxBuffer,
            RxBuffer:=RxBuffer );

Timer(IN:=TRUE,PT:=T#1S );
IF Timer.Q THEN
  IF Sb=0 THEN
    Sb:=64;
  END IF
  Sb:=Sb+1;
  IF Sb>90 THEN
    Sb:=64;
  END IF
  Send(SendByte:=Sb, TxBuffer:=TxBuffer);
  Timer(IN:=FALSE);
END IF

```

7.5 Senden und Empfangen

Daten senden

Im beiliegenden Beispielprogramm werden Daten über die serielle Schnittstelle im Sekundentakt gesendet und das Programm reagiert sofort auf ein eintreffendes Zeichen.

Mögliche Fehler

Während eines SPS-Zyklus dürfen mehrere Zeichen gesendet werden, solange sie vom Sendepuffer aufgenommen werden können. Bei einem Überlauf des Sendepuffers wird der Busy-Ausgang des Sendebausteins nach dem Aufruf TRUE bleiben. Das letzte Zeichen wurde dann nicht gesendet und der Baustein muß im nächsten SPS-Zyklus mit unveränderten Eingangsdaten erneut aufgerufen werden.

Der Füllstand eines Puffers kann jederzeit ermittelt werden (z. B. TxBuffer.Count bzw. TxBuffer.FreeByte).

```

PROGRAM MAIN
VAR
  ComControl: KL6ControlBC;
  Timer: TON;
  Receive: ReceiveByte;
  Send: SendByte;
  Rb: BYTE;
  Sb: BYTE;
END_VAR

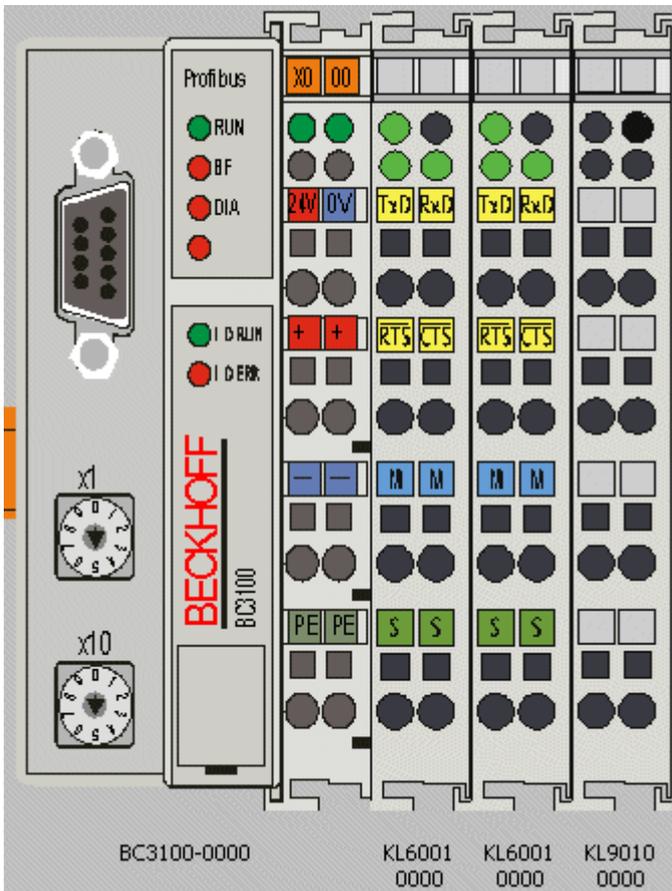
ComControl( CoMin:=KL6InData,
            CoMout:=KL6OutData,
            TxBuffer:=TxBuffer,
            RxBuffer:=RxBuffer );

Timer(IN:=TRUE,PT:=T#1S );
IF Timer.Q THEN
  IF Sb=0 THEN
    Sb:=64;
  END IF
  Sb:=Sb+1;
  IF Sb>90 THEN
    Sb:=64;
  END IF
  Send(SendByte:=Sb, TxBuffer:=TxBuffer);
  Timer(IN:=FALSE);
END IF

```

8 Beispielkonfiguration

Die folgende Hardwarekonfiguration wurde mit dem beiliegenden Testprogramm verwendet.



Handshake

Bei der Klemme KL6001 ist zu beachten, dass sie mit aktivem RTS/CTS Handshake geliefert wird. Die Klemme kann nur senden, wenn der CTS-Eingang aktiv ist. Im einfachsten Fall werden zum Test RTS und CTS durch eine Drahtbrücke verbunden, um das Handshake zu deaktivieren.

9 Fehlercodes

```
COMERROR_NOERROR      : INT := 0;  
COMERROR_TXBUFFOVERRUN : INT := 2;  
COMERROR_STRINGOVERRUN : INT := 10;
```


Mehr Informationen:
www.beckhoff.de/tx1200

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

