BECKHOFF New Automation Technology

Manual | EN

TX1200

TwinCAT 2 | PLC Library: COMlibBC

PLC Libraries

Table of contents

Version: 1.1

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

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

Personnel qualification

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

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER
Serious risk of injury!
Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING
Risk of injury!
Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION
Personal injuries!
Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE
Damage to the environment or devices
Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

● Tip or pointer

i This symbol indicates information that contributes to better understanding.

1.3   Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

The TwinCAT PLC library, COMlibBC, offers function blocks and data structures for serial data communication with the Beckhoff BCxxxx bus controllers. COMlibBC supports the Beckhoff KL6xxx serial bus terminals.

COMlibBC offers blocks for byte-wise transmission and reception of data. All blocks may have multiple instances, so that a number of serial interfaces may be operated in parallel.

The serial bus terminals are available in a 3-byte version and a 5-byte version. In order not to unnecessarily load the memory of the BCxxxx bus controller, a modified ComLibBC5B library is available for 5-byte mode. Use, therefore, one of the two libraries according to the terminal hardware in use. As a rule, the 3-byte version of the serial bus terminal is supplied by the factory.

i   Because there are different versions of the BCxxxx Bus Controllers, there is a variation in the memory mapping of the serial Bus Terminals in 3-byte mode. For this reason, only the 5-byte mode will in future be supported. As far as possible, therefore, only use terminals in 5-byte mode. These can be ordered as KL6xxx-0020.

The PC and the bus controller BC versions of the communication library have been unified since version 2.0. Please compare to the PC documentation**.**

3   Supported Hardware

Serial Bus Terminal

KL6xxx in 3-Byte Mode

The standard version, as supplied, of the Beckhoff serial bus terminal is operated in 3-byte mode. In other words, a bus telegram can transmit or receive 3 data bytes to or from the terminal. Since every data exchange between the PLC and the bus terminal requires 3 PLC cycles, the effective transfer rate is one byte per cycle.

The maximum effective data transfer rate in bits per second depends on the cycle time, T, of the PLC, and on the number of useful bits in each data byte transferred, LB:

$$Bps = LB / T$$

LB = 1 start bit n data bits p parity bits m stop bits

The maximum effective data transmission rate is limited by the physical baud rate programmed into the terminal.

> **ⓘ** Because there are different versions of the BCxxxx Bus Controllers, there is a variation in the memory mapping of the serial Bus Terminals in 3-byte mode. For this reason, this library will in future only support the 5-byte mode. As far as possible, therefore, only use terminals in 5-byte mode. These can be ordered as KL6xxx-0020.

KL6xxx in 5-Byte Mode

The serial bus terminal can be reprogrammed offline by means of a configuration program (Beckhoff KS2000), so that, in 5-byte mode, 5 data bytes at a time can be transferred to or from the terminal. 3 PLC cycles are still necessary for each exchange. The effective data rate for a given cycle time is thus 5/3 greater than in 3-byte mode.

$$Bps = (LB * 5/3) / T$$

The bus terminals cannot be re-programmed while the PLC is running, since the 3-byte and 5-byte modes differ in the register mapping and in the TwinCAT System Manager configuration. For this reason, the required re-programming means that 5-byte operation is only recommended for time-critical applications.

In order not to unnecessarily load the memory of the BCxxxx bus controller, a modified ComLibBC5B library is available for 5-byte mode.

Serial BC8x00 Bus Controller

The BC8x00 bus controller contains a serial interface that works in 5-byte mode as standard. The interface behaves like a serial KL6xxx bus terminal in 5-byte mode. The COMlibBC5B is therefore to be used for programming.

4 Communication Principle

Data buffers of type <u>ComBuffer [▶ 12]</u> are used to decouple the data transfer between the memory and the serial bus terminal. They may be written or read asynchronously.
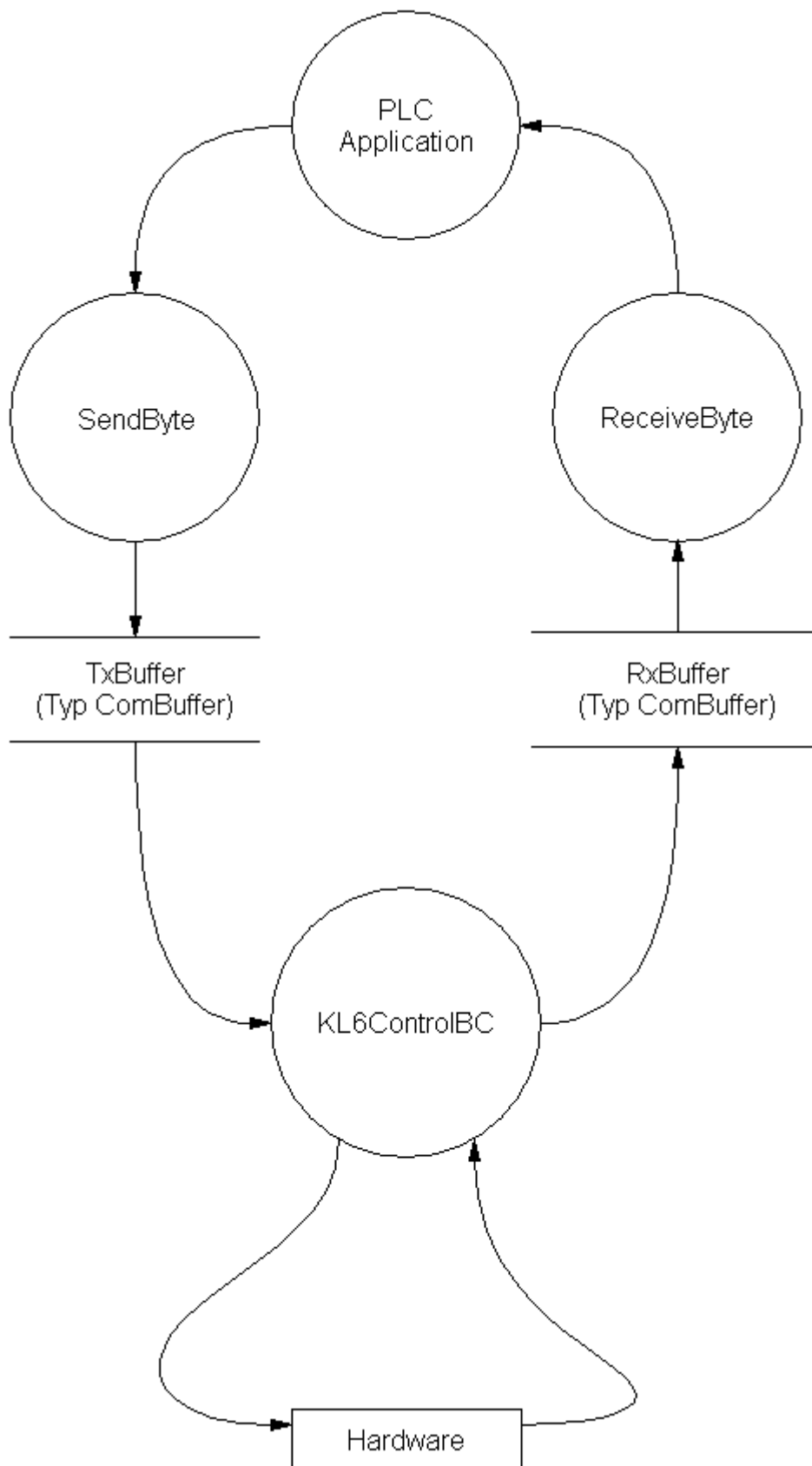
The function blocks described later for receiving and sending data (Send Byte [▶ 14], Receive Byte [▶ 14] etc.) make use of only the data buffers for data exchange, and are thus independent of the hardware being used. In all cases, a communication block (KL6ControlBC [▶ 16]) is called in addition to the send and receive blocks. It handles the data traffic between the data buffers and the hardware with the maximum possible speed in the background.

5 Data Structures

5.1 ComBuffer

The **ComBuffer** data structure is a data buffer that decouples the hardware-dependent communication blocks from the hardware-independent blocks. Data buffers of type ComBuffer are never directly written or read by the user, but are merely used as intermediate storage for the communication blocks.

```
TYPE ComBuffer
STRUCT
    Buffer    : ARRAY[0..300] OF BYTE;
    RdIdx     : INT;
    WrIdx     : INT;
    Count     : INT; (* Number of characters in the ring buffer *)
    FreeByte  : INT;  (* Number of free spaces in the ring buffer *)
    Error     : INT; (* Interface error code *)
    blocked   : BOOL;
END_STRUCT
END_TYPE
```

5.2 Data structures for the KL6xxx serial bus terminal in 3-byte mode

For data exchange by way of the I/O bus, every serial bus terminal needs variables of type KL6inDataBC and KL6outDataBC. These variables are located at fixed addresses in the bus controller's memory map, corresponding to the serial bus terminal's mapping.

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 Data structures for the KL6xxx serial bus terminal in 5-byte mode

For data exchange by way of the I/O bus, every serial bus terminal needs variables of type KL6inData5B and KL6outData5B. These variables are located at fixed addresses in the bus controller's memory map, corresponding to the serial bus terminal's mapping.

KL6inData5B

```
TYPE KL6inData5B
STRUCT
    Status : BYTE;
    D      : ARRAY[0..4] OF BYTE;
END_STRUCT
END_TYPE
```
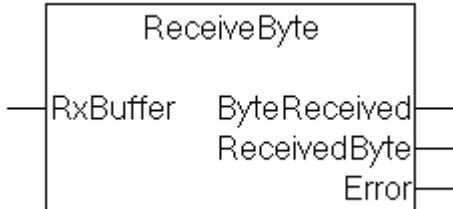
KL6outData5B

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

In order not to unnecessarily occupy the BCxxxx bus controller's memory, the 5-byte mode is only supported by the modified ComLibBC5B library.

6 Function Blocks

6.1 ReceiveByte

```
ReceiveByte

RxBuffer      ByteReceived
              ReceivedByte
                     Error
```

Interface

```
VAR_OUTPUT
    ByteReceived : BOOL;
    ReceivedByte : BYTE;
    Error        : INT;
END_VAR
VAR_IN_OUT
    RxBuffer     : ComBuffer;
END_VAR
```
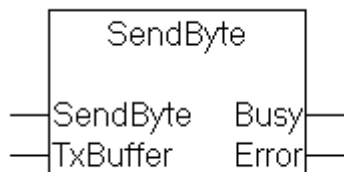
ComBuffer [▶ 12]

Description

The ReceiveByte block receives a single character from the interface corresponding to the input variable RxBuffer. If ByteReceived=TRUE after the call, then the data byte received is available in the output variable ReceivedByte. Otherwise no data has been received.

6.2 SendByte

```
SendByte

SendByte    Busy
TxBuffer    Error
```

Interface

```
VAR_INPUT
    SendByte : BYTE;
END_VAR
VAR_OUTPUT
    Busy    : BOOL;
    Error   : INT;
END_VAR
VAR_IN_OUT
    TxBuffer : ComBuffer;
END_VAR
```
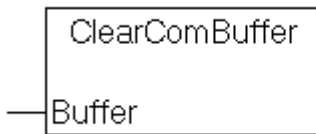
ComBuffer [▶ 12]

Description

SendByte sends a single character to the interface corresponding to the input variable **TxBuffer**. For as long as **Busy**=TRUE, the transmission is not completed. The character was successfully sent when **Busy**=FALSE and Error=0.

> ℹ️ If the send data buffer can still accept data, more than one character can be sent in a single PLC cycle. This is, however, only worthwhile if the buffered characters will be transmitted to the hardware by a faster communication task.

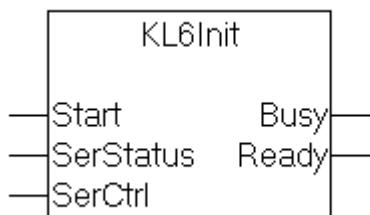# 6.3   ClearComBuffer

Interface

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

ComBuffer [▶ 12]

Description

The communication buffer, **Buffer**, internal to the PLC, is cleared.

6.4   KL6Init

Interface

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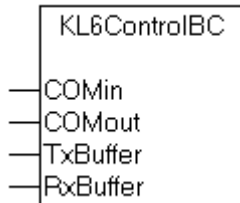
Description

KL6Init initializes the KL6xxx serial bus terminal without altering its configuration. The terminal's internal buffers are cleared.

The **SerStatus** and **SerCtrl** input data correspond to the data linked with the terminal in the TwinCAT System Manager. They are defined in the COMlib <u>KL6inDataBC [▶ 12]</u> and <u>KL6outDataBC [▶ 12]</u> data structures respectively.

The terminal has been successfully initialised after the **Start** signal when Busy=FALSE and Ready=TRUE.

6.5 Background Communication

6.5.1 KL6ControlBC

Interface

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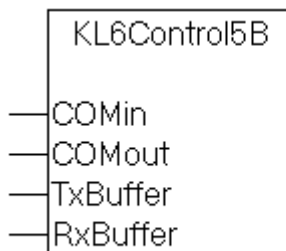
Description

The KL6ControlBC function block controls the data transfer between the data buffers TxBuffer and RxBuffer internal to the PLC and the hardware. The KL6inDataBC and KL6outDataBC data structures are globally declared and are located at a fixed address in the memory map. KL6ControlBC is permanently activated, whether it is intended to send or receive data.

Also see about this

- Data structures for the KL6xxx serial bus terminal in 3-byte mode [▶ 12]
- ComBuffer [▶ 12]

6.5.2 KL6Control5B

Interface

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

Description

KL6Control5B controls the data transfer between the data buffers **TxBuffer** and **RxBuffer**, internal to the PLC and the hardware in a manner similar to the KL6ControlBC [▶ 16] block, but in this case is for the serial bus terminal in 5-byte mode.

In order not to unnecessarily occupy the BCxxxx bus controller's memory, the 5-byte mode is only supported by the modified ComLibBC5B library.

Also see about this

- 📄 Data structures for the KL6xxx serial bus terminal in 5-byte mode [▶ 12]
- 📄 ComBuffer [▶ 12]

7 Linking into a PLC Program

7.1 Installation

Installation is performed by copying the COMlibBC.LB6 and COMlibBC5b.LB6 libraries into the TwinCAT directory TwinCAT\PLC\LIB.

The associated test program should be copied to any project directory of your choice, e.g. to TwinCAT\PLC.

- https://infosys.beckhoff.com/content/1033/tcplclibserialcombc/Resources/12264207755/.zip

Link libraries

Create a new PLC project with TwinCAT PLC Control in order to perform the library linking.

Go to Library Management and add the ComLibBC.LIB library.

7.2 Adapting to the 5-Byte Mode

The ComLibBC5B library is used instead of the ComLibBC library for programming the serial bus terminal in 5-byte mode, and for the serial interface of the BC8x00 bus controller.

The KL6ControlBC function block is thus replaced by the KL6Control5B block. In addition to this, the data structures KL6inDataBC and KL6outDataBC are replaced by KL6inData5B and KL6outData5B respectively.

It should be noted that the KL6xxx serial bus terminals are supplied factory-set for 3-byte mode, and must first be re-programmed using the KS2000 configuration software. Alternatively, appropriately programmed terminals are available as KL6xxx-0020.

7.3 Global Variables

Four global data structures are needed to access a serial interface. Two provide the connection to the hardware in the send and receive directions. Two data buffers are also necessary for intermediate storage.

```
 VAR_GLOBAL
Kl6InData AT %IB0 : Kl6inDataBC;
Kl6OutData AT %QB0: Kl6outDataBC;
RxBuffer         : ComBuffer;
TxBuffer         : ComBuffer;
END_VAR
```

7.4 Background Communication

The communication between the serial hardware and the data buffer, of type ComBuffer [▶ 12], is performed with the aid of a separate block, ComControl.

See also the Communication Principle [▶ 10].

```
 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   Sending and Receiving

Sending data

In the example program included, data is sent via the serial interface at one-second intervals, and the program reacts immediately to an arriving character.

Possible errors

More than one character may be transmitted during one PLC cycle, provided they can be accepted by the send buffer. If the send buffer overflows, the busy output of the send block will remain TRUE after it has been called. In that case the last character is not sent, and the block must be called again with the same input data in the next PLC cycle.

How full a buffer is can be determined at any time (e.g. TxBuffer.Count or 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
```

BECKHOFF

```
    Send(SendByte:=Sb, TxBuffer:=TxBuffer);
    Timer(IN:=FALSE);
END IF
```

8   Example Configuration

The following hardware configuration is used with the included test program.
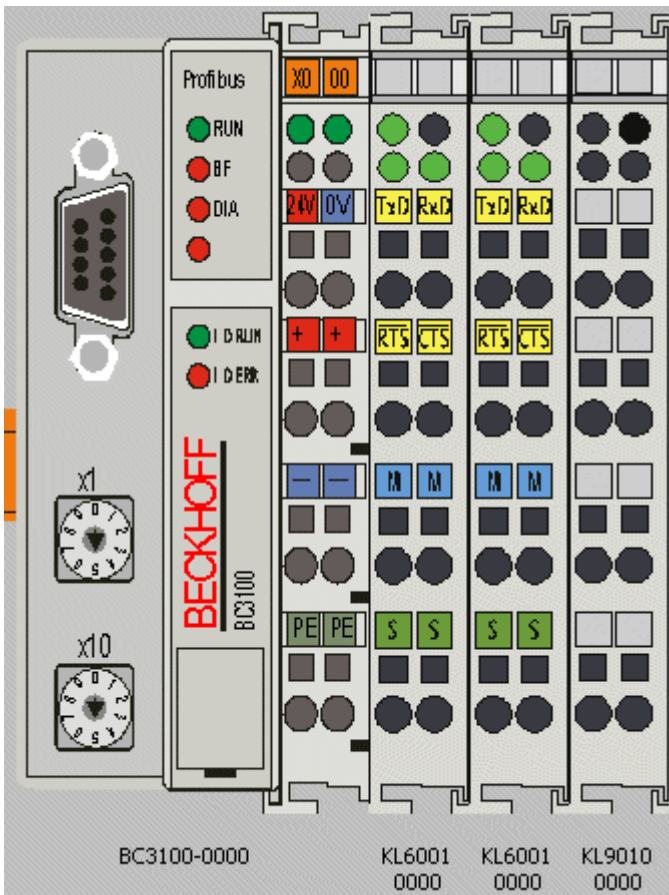
Handshake

Note that the KL6001 terminal is delivered with active RTS/CTS handshake. The terminal can only transmit if the CTS input is active.In the simplest case, RTS and CTS are linked by a wire bridge for the test, in order to deactivate the handshake.

9   Error Codes

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

More Information:
www.beckhoff.com/tx1200