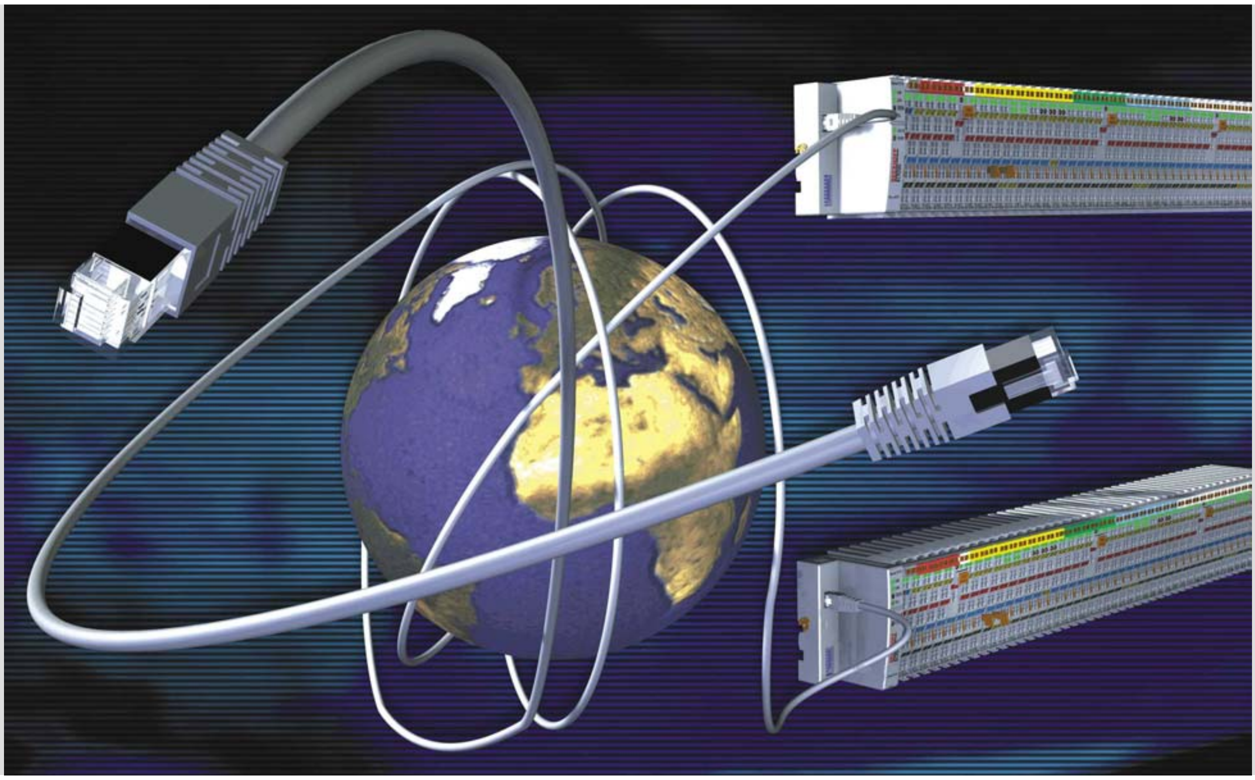


Documentation for the Beckhoff BC9000

Bus Terminal Controller for Ethernet

Version: 3.9
Date: November, the 28th 2002

BECKHOFF

Inhaltsverzeichnis

1. Foreword

Notes on the Manual

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards. It is essential that the following notes and explanations are followed when installing and commissioning these components.

Liability Conditions

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.

The documentation has been prepared with care. The products described are, however, constantly under development. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics, and does not represent an assurance of characteristics in the sense of § 459, Para. 2 of the German Civil Code. In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. 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.

Safety Instructions

Safety Rules

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.

State at Delivery

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 Elektro BECKHOFF GmbH.

Personnel Qualification

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

Description of safety symbols

The following safety symbols are used in this operating manual. They are intended to alert the reader to the associated safety instructions.

Danger	This symbol is intended to highlight risks for the life or health of personnel.

Warning	This symbol is intended to highlight risks for equipment, materials or the environment.

Note	This symbol indicates information that contributes to better understanding.

Documentation Issue Status

Version	Modifications
3.8	HTML documentation; valid as from firmware B8
3.9	HTML documentation; valid as from firmware B9

Firmware BC9000

You can determine which firmware was fitted when the Bus Coupler left the factory from the adhesive label underneath (see the fifth and sixth figures of the production number).

Example:
3200B2020000
The firmware in this example is B2.

In order to update your firmware, you require the KS2000 configuration software and the serial cable included with that software; as from software version B6, you can update your coupler using the TwinCAT 2.8 system manager. You will find the firmware under www.beckhoff.com.

Firmware	Description
B0	Preliminary version
B1	Released version
B2	- Problems with the ASC/CHR functions corrected - The fault which caused error code 1 to be generated after TwinCAT starts up was corrected. - Implementation of PLCSystemBC.lb6
B3	Implementation of ModbusTCP
B5	- Addressing via ARP and BootP possible - Access to the flags area via ModbusTCP
B6	- Implementation of the ModbusTCP, ADS, SMTP and IP-Config function blocks - Firmware update over Ethernet possible - Reading out the configuration with the system manager (as from TwinCAT 2.8) possible - ADS access control through table 2 implemented
B7	- Settings such as the IP address are retained during a firmware update - ADS indication block implemented - ADS "State" supported - DHCP for "Nortel Switch" modified - Cycle time monitoring - Fast ModbusTCP implemented
B8	- Internal software reset modified (this does not have any effects that concern the user)
B9	Error in the B8 version related to mapping the KL60xx Bus Terminals corrected

Firmware BK9000

You can determine which firmware was fitted when the Bus Coupler left the factory from the adhesive label underneath (see the fifth and sixth figures of the production number).

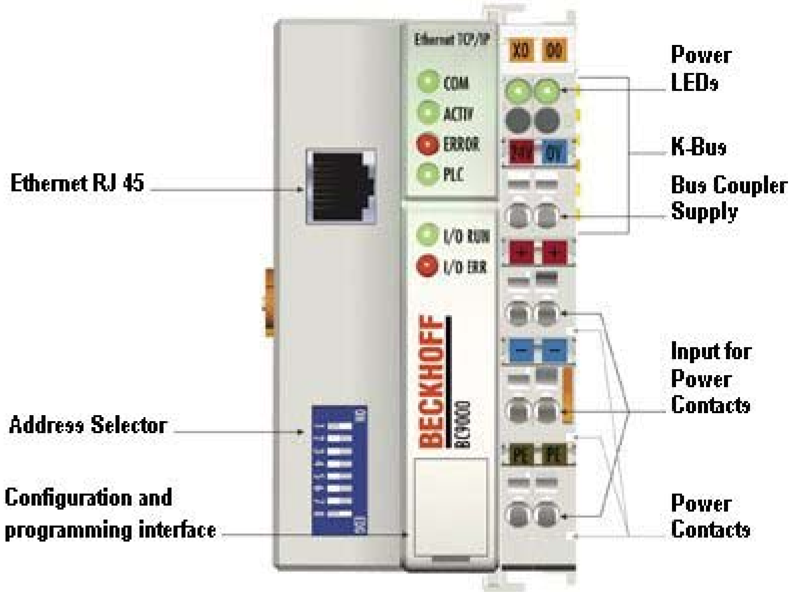
Example:
3200B2020000
The firmware in this example is B2.

In order to update your firmware, you require the KS2000 configuration software and the serial cable included with that software; as from software version B6, you can update your coupler using the TwinCAT 2.8 system manager. You will find the firmware under www.beckhoff.com.

Firmware	Description
B0	Released version
B1	Watchdog LED triggered under ModbusTCP
B2	- Addressing via ARP, DHCP and BootP possible - Access with FC3 ModbusTCP to the process inputs implemented
B3	- Firmware update over Ethernet possible - Reading out the configuration with the system manager (as from TwinCAT 2.8) possible - ADS access control through table 2 implemented
B4	- Settings such as the IP address are retained during a firmware update
B5	- Fast ModbusTCP implemented
B6	- Internal software reset modified (this does not have any effects that concern the user)
B7	- Error in the B6 version related to mapping the KL60xx Bus Terminals corrected

2. Product Overwiew

Technical Data



System data	Ethernet (BC9000)
Number of I/O modules	only limited by the IP address space
Number of I/O points	depending on the controller
Data transfer medium	4 x 2 twisted pair copper cable; category 3 (10 MBaud), category 5 (100 MBaud)
Distance between mod-ules	100 m (distributor hub to BC9000)
Data transfer rate	10/100 MBaud
Topology	star wiring

Technical data	BC9000
Number of Bus Terminals	64
Digital peripheral signals	256 inputs/outputs
Analog peripheral signals	128 inputs/outputs
Protocol	Beckhoff ADS, ModbusTCP
Configuration possibility	Via the KS2000 configuration software or the controller (TwinCAT / TwinCAT BC)
Maximum number of bytes	512 bytes I and 512 bytes O
Bus connection	RJ 45
Power supply	24 V DC (-15%/+20%)
Input current	70 mA + (total K-Bus current)/4
Starting current	approx. 2.5 x continuous current
Recommended fuse	≤ 10 A
K-Bus power supply up to	1750 mA
Power contact voltage	24 V DC max.
Power contact current load	10 A max.
Dielectric strength	500 Veff (power contact / supply voltage / fieldbus)
Weight approx.	170 g
Operating temperature	0°C ... +55°C
Storage temperature	-25 °C ... +85°C
Relative humidity	95 % no condensation
Vibration / shock resistance	conforms to IEC 68-2-6/IEC 68-2-27
EMC resistance burst / ESD	conforms to EN 50082 (ESD, burst)/EN 50081
Installation position	variable
Protection class	IP20

Technical Data for the PLC

PLC data	BC9000
Programmability	Via the programming interface (TwinCAT, TwinCAT BC) or via Ethernet (TwinCAT)
Program memory	64/96 kbyte
Data memory	64/128 kbyte
Remanent flags	4 kbyte
PLC cycle time	Approx. 1.5 ms for 1000 IL commands (without I/O cycle)
Programming languages	IEC 6-3 (IL, LD, FBD, ST, SFC)

System Overview

Up to 64 Bus Terminals each having 2 I/O channels for each signal form

The Bus Terminal system is the universal interface between a fieldbus system and the sensor / actuator level. A unit consists of a Bus Coupler as the head station, and up to 64 electronic series terminals, the last one being an end terminal. For each technical signal form, terminals are available each having two I/O channels, and these can be mixed in any order. All the terminal types have the same mechanical construction, so that difficulties of planning and design are minimised. The height and depth match the dimensions of compact terminal boxes.

Decentralised wiring of each I/O level

Fieldbus technology allows more compact forms of controller to be used. The I/O level does not have to be brought to the controller. The sensors and actuators can be wired decentrally, using minimum cable lengths. The controller can be installed at any location within the plant.

Industrial PCs as controllers

The use of an Industrial PC as the controller means that the operating and observing element can be implemented in the controller's hardware. The controller can therefore be located at an operating panel, in a control room, or at some similar place. The Bus Terminals form the decentralised input/output level of the controller in the control cabinet and the subsidiary terminal boxes. The power sector of the plant is also controlled over the bus system in addition to the sensor/actuator level. The Bus Terminal replaces the conventional series terminal as the wiring level in the control cabinet. The control cabinet can have smaller dimensions.

Bus Couplers for all usual bus systems

The Beckhoff Bus Terminal system unites the advantages of a bus system with the possibilities of the compact series terminal. Bus Terminals can be driven within all the usual bus systems, thus reducing the controller parts count. The Bus Terminals then behave like conventional connections for that bus system. All the performance features of the particular bus system are supported.

Assembly on standardised C mounting rails

The easy, space-saving, assembly on a standardised C-rail, and the direct wiring of actuators and sensors, without cross-connections between the terminals, standardises the installation. The consistent labelling scheme also contributes.

The small physical size and the great flexibility of the Bus Terminal system allows it to be used wherever a series terminal is also used. Every type of connection, such as analog, digital, serial or the direct connection of sensors can be implemented.

Modularity

The modular assembly of the terminal strip with Bus Terminals of various functions limits the number of unused channels to a maximum of one per function. The presence of two channels in one terminal is the optimum compromise of unused channels and the cost of each channel. The possibility of electrical isolation through potential feed terminals also helps to keep the number of unused channels low.

Display of the channel state

The integrated LEDs show the state of the channel at a location close to the sensors and actuators.

K-Bus

The K-Bus is the data path within a terminal strip. The K-Bus is led through from the Bus Coupler through all the terminals via six contacts on the terminals' side walls. The end terminal terminates the K-Bus. The user does not have to learn anything about the function of the K-Bus or about the internal workings of the terminals and the Bus Coupler. Many software tools that can be supplied make project planning, configuration and operation easy.

Potential feed terminals for isolated groups

The operating voltage is passed on to following terminals via three power contacts. You can divide the terminal strip into arbitrary isolated groups by means of potential feed terminals. The potential feed terminals play no part in the control of the terminals, and can be inserted at any locations within the terminal strip.

Up to 64 terminals can be used within one terminal strip. This count does include potential feed terminals, but not the end terminal.

Bus Couplers for various fieldbus systems

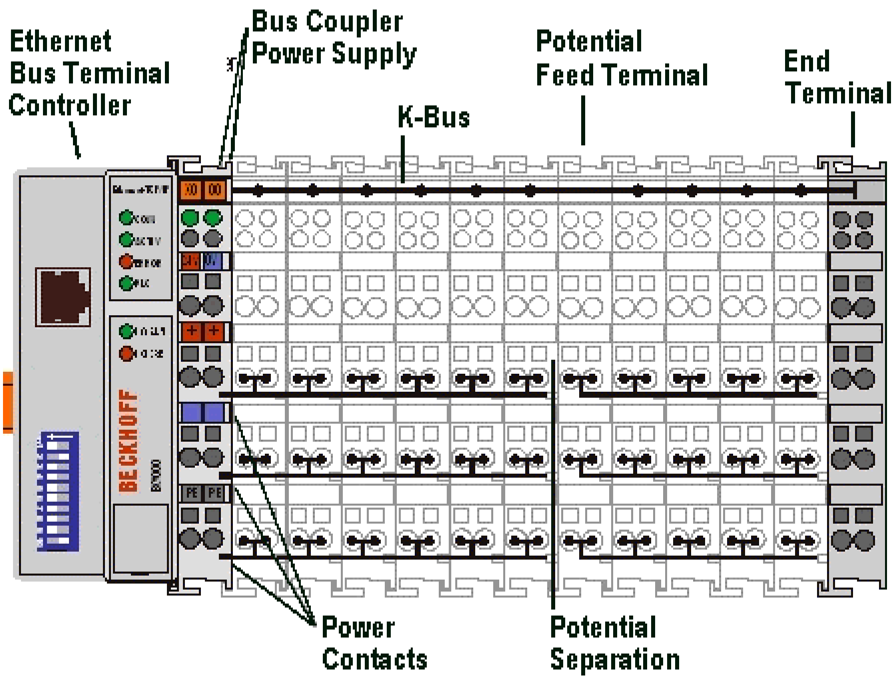
Various Bus Couplers can be used to couple the electronic terminal strip quickly and easily to different fieldbus systems. It is also possible to convert to another fieldbus system at a later time. The bus coupler performs all the monitoring and control tasks that are necessary for operation of the connected Bus Terminals. The operation and configuration of the Bus Terminals is carried out exclusively by the Bus Coupler. Nevertheless, the parameters that have been set are stored in each Bus Terminal, and are retained in the event of voltage drop-out. Fieldbus, K-Bus and I/O level are electrically isolated.

If the exchange of data over the fieldbus is prone to errors or fails for a period of time, register contents (such as counter states) are retained, digital outputs are cleared, and analog outputs take a value that can be configured for each output when commissioning. The default setting for analog outputs is 0 V or 0 mA. Digital outputs return in the inactive state. The timeout periods for the Bus Couplers correspond to the usual settings for the fieldbus system. When converting to a different bus system it is necessary to bear in mind the need to change the timeout periods if the bus cycle time is longer.

The interfaces

A Bus Coupler has six different methods of connection. These interfaces are designed as plug connectors and as spring-loaded terminals.

The Principle of the Bus Terminal



Fieldbus Overview

Ethernet was originally developed by DEC, Intel and XEROX (as the "DIX" standard) for passing data between office devices. The term nowadays generally refers to the *IEEE 802.3 CSMA/CD* specification, published in 1985. Because of the high acceptance around the world this technology is available everywhere and is very economical. This means that it is easy to make connections to existing networks.

There are now a number of quite different transmission media: coaxial cable (10Base5), optical fibre (10BaseF) or twisted pairs (10BaseT) with screen (STP) or without screen (UTP). A variety of topologies such as ring, line or star can be constructed with Ethernet.

Ethernet transmits Ethernet packets from a sender to one or more receivers. This transmission takes place without acknowledgement, and without the repetition of lost packets. To achieve reliable data communication, there are protocols, such as TCP/IP, that can run on top of Ethernet.

Basic Principles

The Internet Protocol (IP)

The internet protocol (IP) forms the basis of this data communication. IP transports data packets from one device to another; the devices can be in the same network, or in different networks. IP here looks after the address management (finding and assigning MAC-IDs), segmentation and routing. Like the Ethernet protocol, IP does not guarantee that the data is transported - data packets can be lost, or their sequence can be changed.

TCP/IP was developed to provide standardised, reliable data exchange between any number of different networks. TCP/IP is thus substantially independent of the hardware or software being used. Although the term is often used as if it were a single concept, a number of protocols are layered together: e.g. IP, TCP, UDP, ARP and ICMP.

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) which runs on top of IP is a connection-oriented transport protocol. It includes error detection and error handling mechanisms. Lost telegrams are repeated.

User Datagram Protocol (UDP)

UDP is connectionless transport protocol. It provides no control mechanism when exchanging data between sender and receiver. This results in a higher processing speed than, for example, TCP. Checking whether or not the telegram has arrived must be carried out by the higher-level protocol.

Internet Control Message Protocol (ICMP)

It is used by end devices, to exchange information about the current status of the internet protocol.

Address Resolution Protocol (ARP)

Performs conversion between the IP addresses and MAC addresses.

BootP

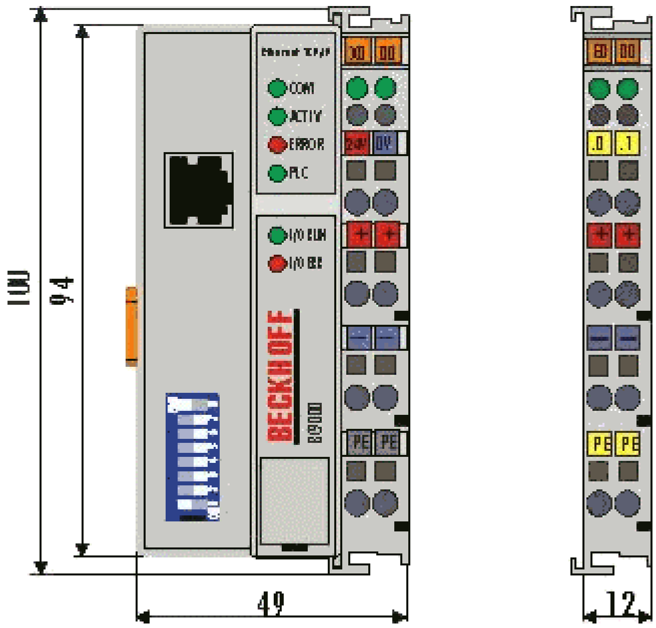
The BootP protocol allows the TCP/IP address to be set or altered, by addressing the network device with its MAC-ID.

3. Mounting and Wiring

Mecanical Installation

Dimensions

The system of the Beckhoff Bus Terminals is characterised by low physical volume and high modularity. When planning a project it must be assumed that at least one Bus Coupler and a number of Bus Terminals will be used. The mechanical dimensions of the Bus Couplers are independent of the fieldbus system.



The total width in practical cases is composed of the width of the Bus Coupler, the KL9010 bus end terminal and the width of the Bus Terminals in use. Depending on function, the Bus Terminals are 12 or 24 mm wide. The front wiring increases the total height of 68 mm by about 5 to 10 mm, depending on the wire thickness.

Installation of Bus Terminals on C mounting rails

Danger	Bring the bus system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Assembly

The Bus Coupler and Bus Terminals are attached to commercially available 35 mm C mounting rails (EN 50022) by applying slight pressure:

1. First attach the Fieldbus Coupler to the mounting rail.
2. The Bus Terminals are now attached on the right-hand side of the Fieldbus Coupler. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.
   If the Terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

During the installation of the Bus Terminals, the locking mechanism of the terminals must not come into conflict with the fixing bolts of the mounting rail.

Disassembly

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

1. Carefully pull the orange-colored lug approximately 1 cm out of the disassembled terminal, until it protrudes loosely. The lock with the mounting rail is now released for this terminal, and the terminal can be pulled from the mounting rail without excessive force.
2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal away from the mounting rail.

Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realised by joining the components:

- The six spring contacts of the K-Bus deal with the transfer of the data and the supply of the Bus Terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminals on the Bus Coupler.

Note	During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (KL91xx, KL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

⚠ **Warning**	Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

The PE power contact must not be used for other potentials!

Wiring

Up to eight connections enable the connection of solid or finely stranded cables to the Bus Terminals. The terminals are implemented in spring force technology. Connect the cables as follows:

1. Open a spring-loaded terminal by slightly pushing with a screwdriver or a rod into the square opening above the terminal.
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal closes automatically when the pressure is released, holding the wire securely and permanently.

Wiring

Power Supply, Potential Groups

Bus Coupler power supply

The Bus Couplers require a 24 V DC supply for their operation. The connection is made by means of the upper spring-loaded terminals labelled *24 V* and *0 V*. The supply voltage feeds the Bus Coupler electronics and, over the K-Bus, the Bus Terminals. The power supply for the Bus Coupler electronics and that of the K-Bus are electrically separated from the potential of the field level.

Input for Power Contacts

The bottom six connections with spring-loaded terminals can be used to feed the supply for the peripherals. The spring-loaded terminals are joined in pairs to a power contact. The feed for the power contacts has no connection to the voltage supply for the Bus Coupler. The design of the feed permits voltages of up to 24 V. The assignment in pairs and the electrical connection between feed terminal contacts allows the connection wires to be looped through to various terminal points. The current drawn from the power contacts must not exceed 10 A for long periods. The current carrying capacity between two spring-loaded terminals is identical to that of the connecting wires.

Power contacts

On the right hand face of the Bus Coupler there are three spring contacts for the power contact connections. The spring contacts are hidden in slots so that they can not be accidentally touched. By attaching a Bus Terminal the blade contacts on the left hand side of the Bus Terminal are connected to the spring contacts. The tongue and groove guides on the top and bottom of the Bus Coupler and of the Bus Terminals guarantees that the power contacts mate securely.

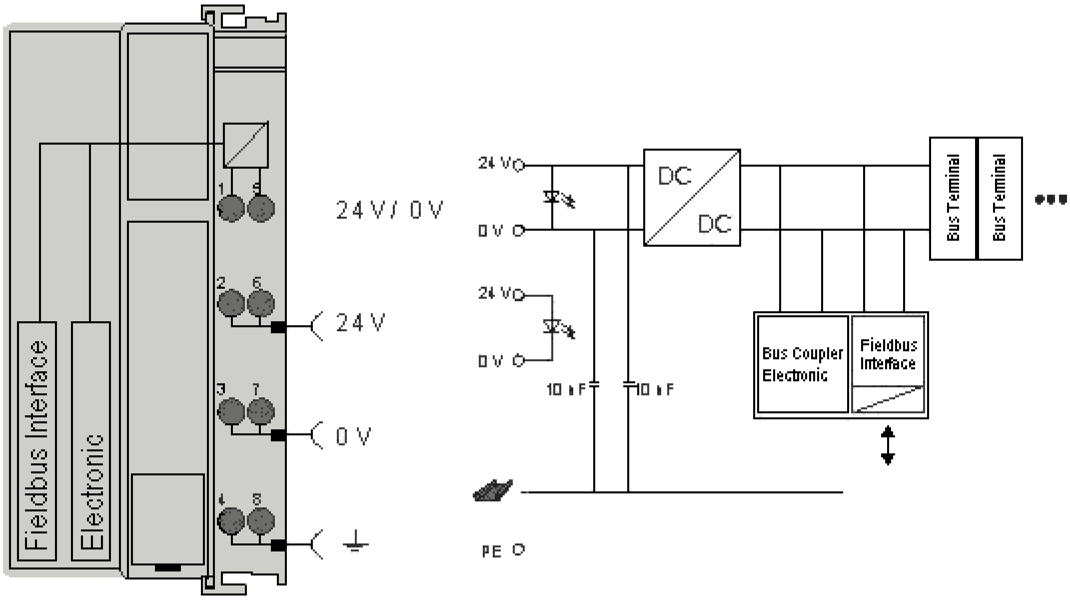
Configuration Interface

The standard bus couplers have an RS232 interface at the bottom of the front face. The miniature connector can be joined to a PC with the aid of a connecting cable and the KS2000 configuration software. The interface permits the Bus Terminals to be configured, for example adjusting the amplification factors of the analog channels. The interface can also be used to change the assignments of the bus terminal data to the process image in the Bus Coupler. The functionality of the configuration interface can also be reached via the fieldbus using string communication facility.

Electrical isolation

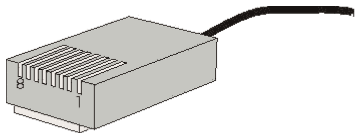
The Bus Couplers operate by means of three independent potential groups. The supply voltage feeds the K-Bus electronics in the Bus Coupler and the K-Bus itself, which are electrically isolated. The supply voltage is also used to generate the operating voltage for the fieldbus.

Remark: All the Bus Terminals are electrically isolated from the K-Bus. The K-Bus is thus electrically isolated from everything else.



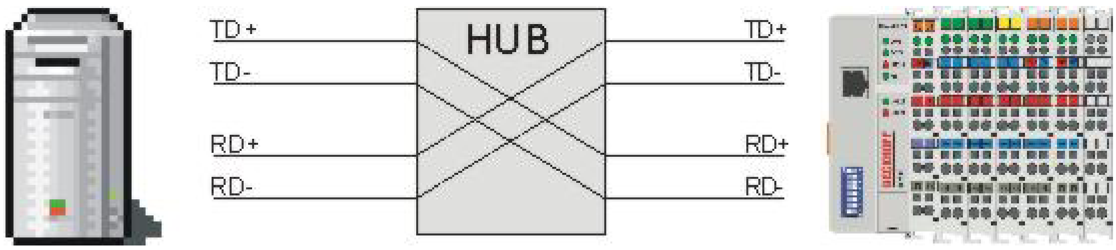
Ethernet Connection

The connection to the Ethernet bus is made via an RJ45 connector (a Western plug).



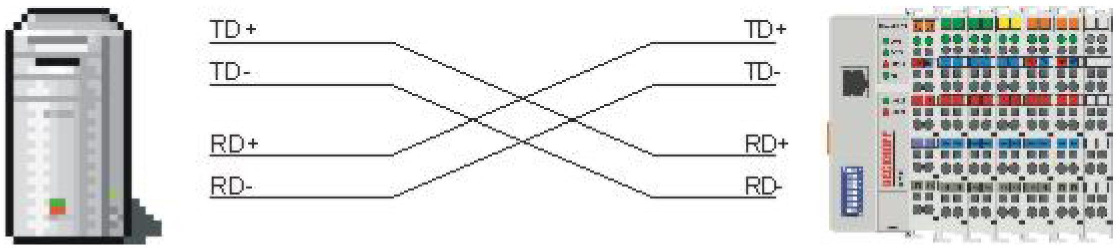
Cabling

Connection via hub or switch



Connect the PC's network card to the hub using a standard Ethernet cable, and connect the hub, again using a standard Ethernet cable, to the Bus Terminal controller. Connection via a switch is done in the same way.

Direct connection between PC with Ethernet card and BC9000



To connect the PC directly to the Bus Terminal Controller, you must use an Ethernet cable in which the pairs of cores have been crossed (a crossover cable).

Pin assignment of the RJ45 plug

PIN	Signal	Description
1	TD +	Transmit +
2	TD -	Transmit -
3	RD +	Receive +
4	-	reserved
5	-	reserved
6	RD -	Receive -
7	-	reserved
8	-	reserved

Ethernet Cable

Transmission standards

10Base5

The transmission medium for 10Base5 consists of a thick coaxial cable ("yellow cable") with a max. transmission speed of 10 MBaud arranged in a line topology with branches (drops) each of which is connected to one network device. Because all the devices are in this case connected to a common transmission medium, it is inevitable that collisions occur often in 10Base5.

10Base2

10Base2 (Cheapernet) is a further development of 10Base5, and has the advantage that the coaxial cable is cheaper and, being more flexible, is easier to lay. It is possible for several devices to be connected to one 10Base2 cable. It is frequent for branches from a 10Base5 backbone to be implemented in 10Base2.

10BaseT

Describes a twisted pair cable for 10 MBaud. The network here is constructed as a star. It is no longer the case that every device is attached to the same medium. This means that a broken cable no longer results in failure of the entire network. The use of switches as star couplers enables collisions to be reduced. Using full-duplex connections they can even be entirely avoided.

100BaseT

Twisted pair cable for 100 MBaud. It is necessary to use a higher cable quality and to employ appropriate hubs or switches in order to achieve the higher data rate.

10BaseF

The 10BaseF standard describes several optical fibre versions.

Short description of the 10BaseT and 100BaseT cable types

Twisted pair copper cable for star topologies, where the distance between two devices may not exceed 100 metres.

UTP
Unshielded twisted pair
This type of cable belongs to category 3, and is not recommended for use in an industrial environment.

S/UTP
Screened/unshielded twisted pair (screened with copper braid)
Has a general screen of copper braid to reduce influence of external interference. This cable is recommended for use with Bus Couplers.

FTP
Foiled shielded twisted pair (screened with aluminium foil)
This cable has an outer screen of laminated aluminium and plastic foil.

S/FTP
Screened/foiled-shielded twisted pair (screened with copper braid and aluminium foil)
Has a laminated aluminium screen with a copper braid on top. Such cables can provide up to 70 dB reduction in interference power.

STP
Shielded twisted pair
Describes a cable with an outer screen, without defining the nature of the screen any more closely.

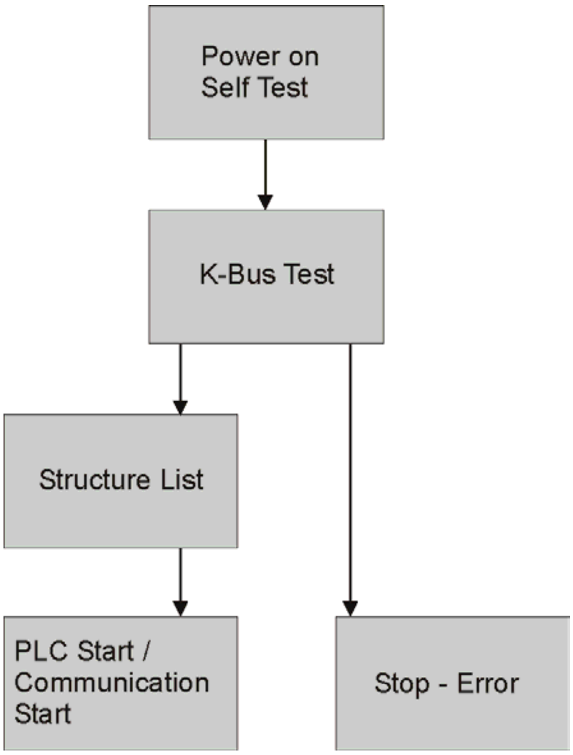
S/STP
Screened/shielded twisted pair (wires are individually screened)
This identification refers to a cable with a screen for each of the two wires as well as an outer shield.

ITP
Industrial Twisted-Pair
The structure is similar to that of S/STP, but, in contrast to S/STP, it has only one pair of conductors.

4. Parameterisation and Start-up

Start-up Behaviour of the Bus Coupler

After being switched on, the Bus Coupler checks its state, configures the K-Bus, and creates a structure list on the basis of the inserted bus terminals. If the Bus Coupler contains a decentralised controller (BCxxxx) the local PLC is started once the structure list has successfully been created.
The I/O LEDs illuminate and flash as the Bus Coupler starts up. If there are no errors, the I/O LEDs should stop flashing within about 2-3 seconds. If the event of an error, then the LED that flashes will depend on the type of that error (see Diagnostic LEDs).



Parameterisation of the Bus Coupler using DIP Switches

The following parameterisations can be carried out without using configuration software, with the aid of the DIP switches and the end terminal (KL9010).
This parameterisation mode is only active if only one end terminal (KL9010) is inserted. Otherwise, the normal settings apply.

Restoring the manufacturer's settings

- Switch off the Bus Coupler, and plug in just the end terminal (KL9010).
- Set all the DIP switches to ON, and switch the Bus Coupler on again.
- Once the default parameters have successfully been set, the *Error* LED lights, and the *I/O Run* and *I/O Error* LEDs flash alternately.
- You can then switch the Bus Coupler off, connect the Bus Terminals, and continue as usual.

Deleting the boot project

- Switch off the Bus Coupler, and plug in just the end terminal (KL9010).
- Set DIP switches 1 to 9 to ON, DIP switch 10 to OFF, and switch the Bus Coupler on again.
- Once the boot project has been successfully deleted, the *I/O Run* and *I/O Error* LEDs flash alternately.
- You can then switch the Bus Coupler off, connect the Bus Terminals, and continue as usual.

Setting the Ethernet parameters

- Switch off the Bus Coupler, and plug in just the end terminal (KL9010).
- Set all the DIP switches to OFF, and switch the Bus Coupler on again.
- The *I/O* RUN and *I/O Error* LEDs light steadily.
- Make the desired setting in accordance with the following table.

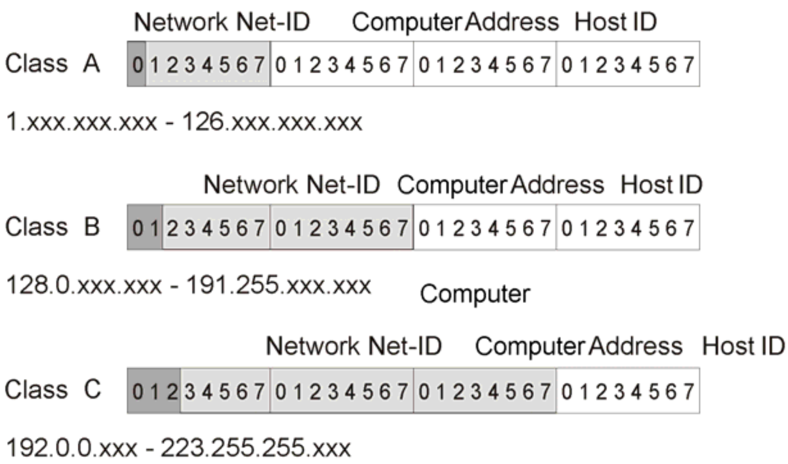
DIP switch	Parameters	Selection	Setting	Comment
1	Baud Rate			
		10 MBaud	OFF (0)	
		100 MBaud	ON (1)	Default
2	Auto-Baud-Rate			
		Disable	OFF (0)	
		Enable	ON (1)	Default
3	Transmission type			
		Half duplex	OFF (0)	
		Full duplex	ON (1)	Default

- To accept the values, set DIP switch 10 to ON.
  By flashing the *I/O RUN* and *I/O ERR* LEDs the Bus Coupler indicates that it has accepted the parameters.
- You can then switch the Bus Coupler off, connect the Bus Terminals, and continue as usual.

Network Classes

Three different addressing classes are distinguished. They specify how many address bits are reserved for the network number and how many for the computer number (or node number). The difference is located in the first 3 bits of the IP address. Class A has 7 bits of the network address. This makes 126 class A networks possible in the world. The networks can then posses 224 computers. Class B networks can contain 65536 computers, but 16384 networks are possible. Class C networks have a maximum of 256 computers, but 221 networks can be created.

Warning	An IP address must be unique within the entire connected network!



Note	In a communication with another Ethernet devices, the IP address set must have the same network class. Example: Your PC has address 172.16.17.55, which means that the Bus Coupler must have address 172.16.xxx.xxx (each xxx stands for a number between 0..255 - "0" is normally used for routers, and should therefore be reserved).

In order to see the PC's own address, the command "ipconfig" can be entered into a DOS window under Windows NT/2000.

IP Address

The IP address can be set using four different procedures, and these will be described in more detail below.

Procedure	Explanation	Necessary components
KS2000	Addressing using the KS2000 configuration software and DIP switches	KS2000 configuration software and KS2000 cable
ARP	Addressing via the ARP table	PC with network
BootP	Addressing via BootP server	BootP server
DHCP	Addressing via DHCP server	DHCP server

Address Configuration with KS2000

Using the KS2000 configuration software you can set the TCP/IP address via a dialog box (as from version 3.2.8) or write directly into the registers. DIP switches 9 and 10 must be set to OFF (0) before switching on.

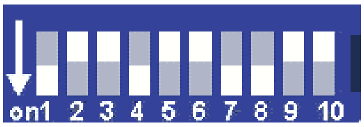
Table 100

Register	High byte	Low byte
0	IP byte 2	IP byte 1
1	Not used	IP byte 3

Default

Byte	Default value (hex)	Default value (dec)
1	0xAC	172_{dec}
2	0x10	16_{dec}
3	0x11	17_{dec}
4	(DIP switch)	(0 to 255_{dec})

Example:



Switch no.	1	2	3	4	5	6	7	8	9	10	
Weight	20	21	22	23	24	25	26	27	-	-	
In this example	ON	OFF	OFF	ON	OFF	OFF	ON	ON	OFF	OFF	
Value	1	0	0	8	0	0	64	128	-	-	Total=201

Address Configuration via ARP

An easy method of modifying the IP address is to set the address using the DOS window. It is, however, only possible to alter addresses within the same network class. The new IP address that has been set remains stored even after the Bus Coupler has been switched off.

Procedure:

- Set DIP switches 9 and 10 to OFF. DIP switches 1-8 then no longer have any address function.
- Open a DOS box on your PC.
- Enter the command "ping <OLD IP address>" to create an entry in the ARP table.
- Read the table with the command "ARP -a".
- Enter "ARP -d <OLD IP address>" to remove the Bus Coupler from the table.
- Use "ARP -s <NEW IP address> <MAC-ID>" to make an entry manually.
- With "ping -l 123 <NEW IP address>" the new IP address becomes valid.

A short flash from the ERROR LED at the moment of switching on indicates that the Bus Coupler is being addressed by ARP, and that DIP switches 1-8 give no indication of the address that is set.

Note	When the IP address is changed, all the dynamic ARP entries should be cleared, because on receipt of an ICMP (Internet Control Message Protocol) IP telegram (e.g. a ping) with the correct MAC-ID and the old IP address, the coupler will again accept the old IP address and will reconfigure itself. To avoid this, only one ping with the length of 123 bytes is permitted for reconfiguration of the IP address (<ping -l "IP address>).

Example:

1. C:>ping 172.16.17.255
2. C:>arp -a
   172.16.17.255 00-01-05-00-11-22
3. C:>arp -d 172.16.17.255
4. C:>arp -s 172.16.44.44 00-01-05-00-11-22
5. C:>ping -l 123 172.16.44.44

Address Configuration via BootP Server

If the address is to be set by the Beckhoff BootP server, then set DIP switch 9 to ON (1) and DIP switch 10 to OFF (0). DIP switches 1-8 then no longer have any address function. If this is not the case, the Bus Coupler will report error code 6 -4. The *TCP/IP ERROR* LED flashes while the address is being allocated.

IP address save modes

DIP switches 1-8 in the ON position

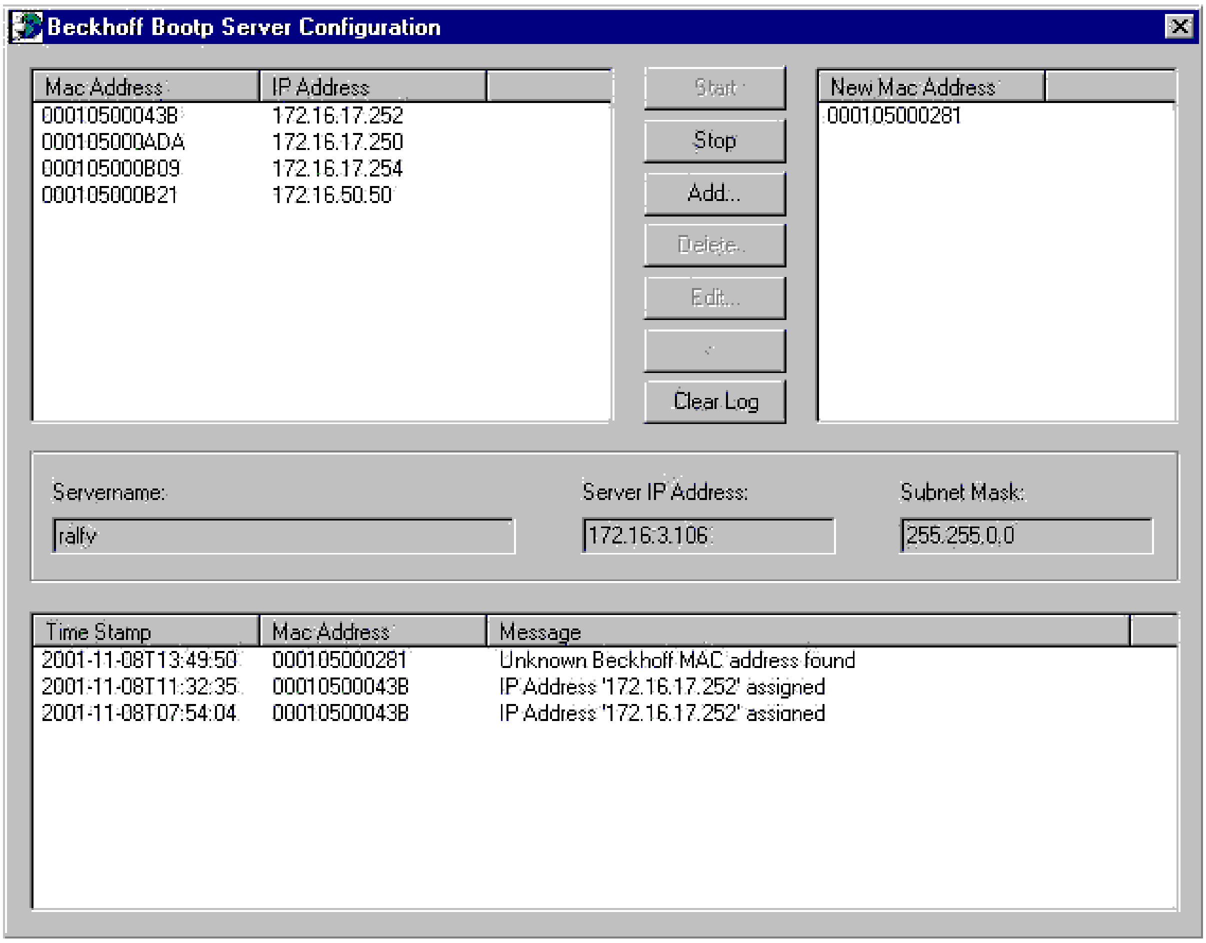
The address assigned by the BootP server is stored, and the BootP service will not be restarted after the next cold start.
The address can be cleared again by reactivating the manufacturers' settings (using the KS2000 software or by DIP switch and end terminal).

DIP switches 1-8 in the OFF position

The IP address assigned by the BootP server is only valid until the Bus Coupler is switched off. The BootP server must assign a new IP address to the Bus Coupler at the next cold start.
The address is, however, retained through a software reset of the Bus Coupler.

Beckhoff BootP server

Beckhoff supply a BootP server for Windows 98, ME, NT4.0, NT2000 and XP.



As soon as the BootP server has started, the *New MAC Address* window shows all the Beckhoff nodes that are working in BootP mode and still have not received an IP address. The assignment of the MAC-ID to IP address is made with the "<<" button. Successful assignment is displayed in the log window.
To start the BootP server automatically when your PC boots, it is only necessary to provide a shortcut in the Windows autostart folder. Include the */Start* parameter in the shortcut (.../TcBootPDlg.exe/start).

Address Configuration via DHCP Server

To set the address by means of a DHCP server, set DIP switch 9 to OFF (0) and DIP switch 10 to ON (1). In this state, the DHCP service is switched on, and the Bus Coupler is automatically assigned an IP number by the DHCP server. For this purpose the DHCP server must know the Bus Coupler's MAC-ID. The IP address should be set statically. The *TCP/IP Error* LED flashes while the address is being allocated.

Subnet Mask

The subnet mask is subject to the control of the network administrator, and specifies the structure of the subnet.

Small networks without a router do not require a subnet mask. The same is true if you do not use registered IP numbers. A subnet mask can be used to subdivide the network with the aid of the mask instead of using a large number of network numbers.

The subnet mask is a 32-bit number.
- Ones in the mask indicate the subnet part of an address region.
- Zeros indicate that part of the address region which is available for the host IDs.

Description	Binary representation	Decimal representation
IP address	10101100.00010000.00010001.11001000	172.16.17.200
Subnet mask	11111111.11111111.00010100.00000000	255.255.20.0
Network ID	10101100.00010000.00010000.00000000	172.16.16.0
Host ID	00000000.00000000.00000001.11001000	0.0.1.200

Standard subnet mask

Address class	Standard subnet mask (decimal)	Standard subnet mask (hex)
A	255.0.0.0	FF.00.00.00
B	255.255.0.0	FF.FF.00.00
C	255.255.255.0	FF.FF.FF.00

Note	Neither subnet 0 nor the subnet consisting only of ones may be used. Neither host number 0 nor the host number consisting only of ones may be used! If the IP address is set using the KS2000 configuration software, it is necessary for the subnet mask also to be changed with the KS2000 configuration software. If ARP addressing is used, the associated standard subnet mask, based on the IP address, is entered. Under BootP or DHCP the subnet mask is entered by the server.

Testing the IP Address

Use the *Ping* command to test the IP address.

```
D:\WINNT\System32\cmd.exe
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

D:\>ping 172.16.17.202

Ping wird ausgeführt für 172.16.17.202 mit 32 Bytes Daten:

Antwort von 172.16.17.202: Bytes=32 Zeit=10ms TTL=60
Antwort von 172.16.17.202: Bytes=32 Zeit=10ms TTL=60
Antwort von 172.16.17.202: Bytes=32 Zeit=10ms TTL=60
Antwort von 172.16.17.202: Bytes=32 Zeit=10ms TTL=60

D:\>
```

Reading the MAC-ID

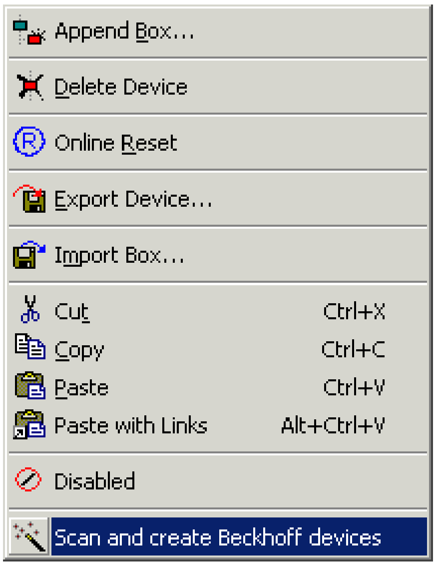
Proceed as follows to read the MAC-ID.

- Change the IP address of your Bus Coupler to 172.16.x.x.
  The IP address when supplied by the manufacturer is 172.16.17.255 (DIP switches 1 to 8 are set to ON).
- Send a Ping to IP address 172.16.17.255
- Read the MAC-ID with *arp -a*.

5. Configuration

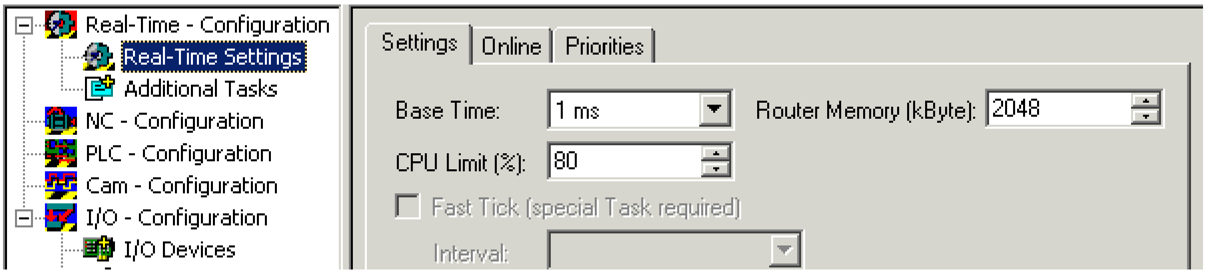
Configuration using the System Manager

Enter a general Ethernet card in the TwinCAT System Manager under devices. If the Bus Couplers are already connected to the network and have IP addresses, you can also read these. This will cause all the Bus Couplers with Bus Terminals and their configurations to be loaded. You can then adapt these to meet your requirements.



Commissioning with 6 or more Ethernet nodes

If you enter more than five nodes in the System Manager, you will have to increase the router memory. You will find this setting in the System Manager under *Real Time Settings*. Enter 350 kbyte there for each Bus Coupler. This will give, for instance, 3.5 MB (10 x 350 kbyte) for 10 Bus Couplers.

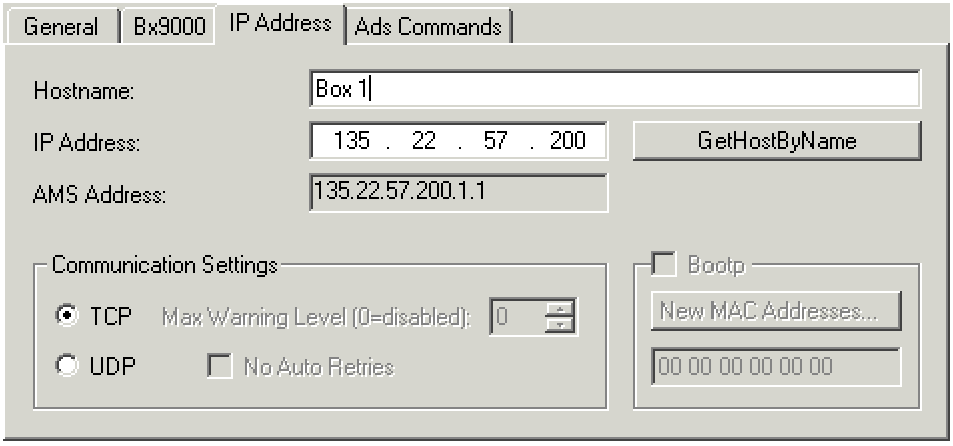


Note: The router memory depends on your PC's working memory, and there is a limit to how high it can be set. Calculation of the maximum possible router memory:
Max. router memory [MB] = (1 MB +((RAM [MB] - 4 MB) x 0.4)

Example:
Suppose your PC has 128 MB of RAM:
1MB +(128MB - 4MB) x 0,4 = 50,6 MB
So the maximum size of the router memory must not exceed 50.6 MB if there is 128 MB of RAM.

IP Address Tab

Ethernet components require an unambiguous IP address in the network. The setting dialog necessary for configuration of the fieldbus nodes in the TwinCAT system is described below. The tab illustrated appears after you have selected a BK9000, BC9000 or other Ethernet fieldbus device with the right mouse button (Add box) under your Ethernet cards in the System Manager.



Host name
The name of the Bus Coupler station can be edited here.

IP address
Enter the IP address of the Bus Coupler here.

GetHostByName
By clicking this button you can (as from BK9000 firmware version *B2*) obtain an IP address from a Windows 2000 DHCP (Dynamic Host Configuration Protocol) server.

AMS address
Reports the Bus Coupler's AMS Net ID. It is automatically generated from the IP address by adding two additional bytes (e.g. ".1.1").

BootP
This checkbox can be activated if the Beckhoff BootP server has been installed and started or (as from TwinCAT Version 2.8).

New MAC Addresses
If the Beckhoff BootP server has been started, the MAC addresses (Media Access Controller) of Ethernet fieldbus devices that have newly been connected can be displayed by clicking this button. If only one new Ethernet fieldbus device has been connected, then only one new MAC address will accordingly be displayed. You can then assign it to the desired IP address in the *IP address field* described above. Each time another Ethernet fieldbus component is added you can repeat this process on the tab described here for the new device.

Communication Settings
Settings for IP (Internet Protocol) communication.

TCP
If this option is active (which is the default) the communication is handled by TCP (*Transmission Control Protocol*). This means that telegram packets to and from Bus Couplers are exchanged using a reliable (acknowledged) mechanism.

UDP
If this option is selected, communication is handled by UDP (*User Datagram Protocol*). In contrast to TCP, the individual telegram packets are not exchanged using a reliable mechanism. They are, in other words, not individually acknowledged by the receiver after they have arrived. Telegram packets that are damaged or whose sequence has become disturbed are neither re-sent nor sorted. The advantage of UDP is that the telegram transfer times can be calculated, because it is not necessary to wait for an answer from the receiver. It is therefore possible to talk of UDP in terms of a limited real-time capability.

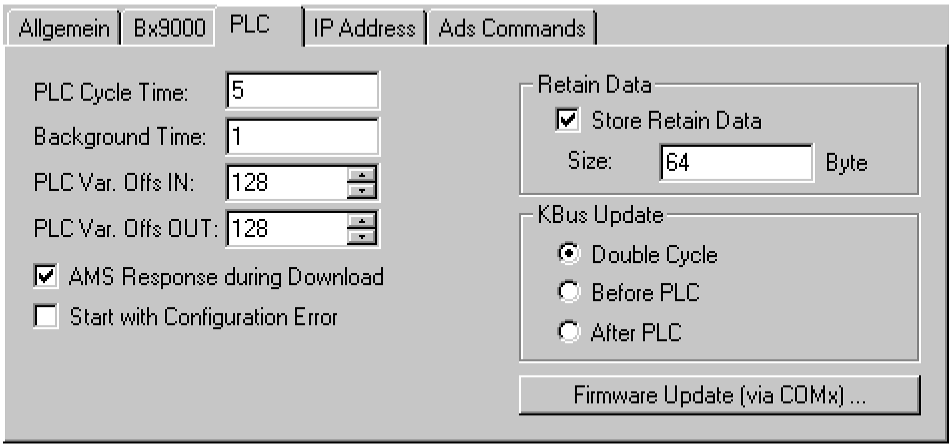
Max. Warning Level
This setting option is only active with *UDP*. The maximum value of the error counter can be entered here. Waiting for response telegrams from the fieldbus node stops when the set maximum value has been reached. Instead, read telegrams only, based on a higher cycle time, are sent to the fieldbus nodes.

No Auto Retries
This checkbox also can only be selected when the *UDP* option is active. If this checkbox is active, then when the level set under *Max. Warning Level* has been reached it will be necessary to execute the *Online Reset* function.

PLC tab

Special data exchange variables for the fieldbus-based communication between the PC and the Bus Terminal Controllers (BC) can be defined for Bus Terminal Controllers under *PLC variables* (see Example of a Bus Controller Configuration).

The following tab, amongst others, appears when a BC is selected in the TwinCAT System Manager's tree view:



PLC Cycle Time
The small controller's reserved cycle time. During this period, no data will be sent by the BC to the fieldbus.

Background Time
Time for the processing of the communication via the fieldbus (includes program download, debugging, etc.).

PLC Var. Offs IN
Start offset of the data exchange variables in the input area of the BC Process Image

PLC Var. Offs OUT
Start offset of the data exchange variables in the output area of the BC process image

AMS Response during download
Can be disabled for compatibility reasons in old Bus Terminal Controllers (BC3100 and BC2000). Newer bus controller firmware versions acknowledge the receipt of individual download sequences. This checkbox must therefore be activated (in all cases for BC9000).

Start with Configuration Error
Even if the configured BC9000 is not connected to the network, it is possible to start the TwinCAT system if this checkbox is enabled.

Retain Data
Store Retain Data: Activates the support of remanent data in the Bus Terminal Controller.

Size: Size of the residual marker area within the Bus Terminal controller (%M x 0 .. %Mxyy).

KBus Update
Double Cycle: Activates execution of I/O cycles (K-Bus update) within the BC station before and after each PLC cycle.
Before PLC: A KBus update will be carried out before executing the PLC cycle.
After PLC: A KBus update will be carried out after executing the PLC cycle.

Firmware Update (via COMx)
This function can be used to update the BC firmware.

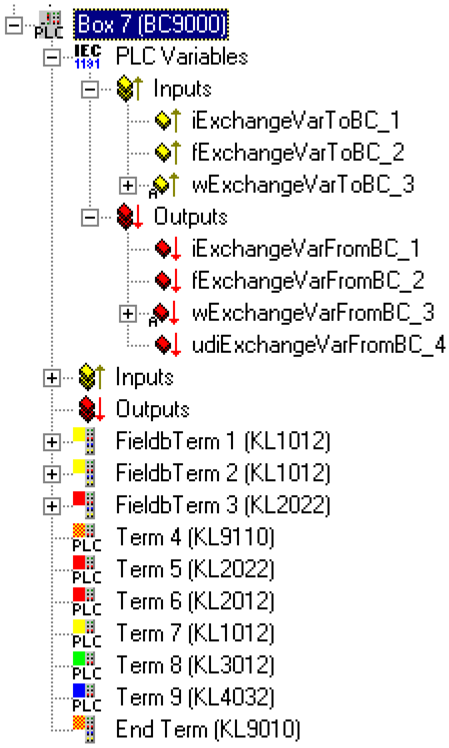
Variable tab

After adding variables below the tree node *PLC Variables* you will see the dialog for the selected variable on the right-hand side, as described in the *Variables Information* section.

Choose meaningful names for the data exchange variables, because they will be used as variable names in the local BC process image when Variable information is exported. After this you can link the data exchange variables with existing tasks.

Example of a Bus Controller configuration

The image below shows both data exchange variables and fieldbus variables. The BC manages terminals 4 to 9 locally.



ModbusTCP

Configuration for ModbusTCP

No special configuration is necessary for communication over ModbusTCP. The default setting has 16 bytes input and output (PLC variables). These can be reached through the Modbus functions.

6. Programming

TwinCAT PLC

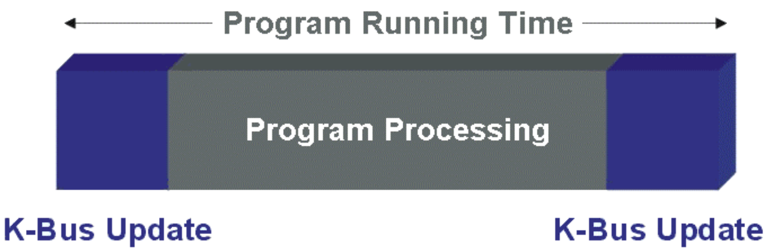
The Beckhoff TwinCAT Software System turns any compatible PC into a real-time controller with a multi-PLC system, NC axis control, programming environment and operating station. The TwinCAT programming environment is also used for programming the BC9000. If you have installed TwinCAT PLC (WinNT4, Win2000, XP), you can use the Ethernet connection or the serial interface for software download and for debugging. If you are programming with TwinCAT BC (also compatible with Win95/98/ME), the connection to the BC9000 is made exclusively via the serial interface.

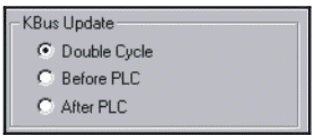
TwinCAT I/O or TwinCAT PLC can also be used as the Ethernet Master (host), in order to exchange process data with the Bus Terminal Controller. TwinCAT provides you with the System Manager as a configuration tool, as well as the drivers and the ADS protocol.

PLC Cycle Time

The PLC cycle time determines the program's repetition frequency. This time is not deterministic. This means that the PLC cycle time can become longer in some programs, and can exceed the cycle time that has been set. If the program requires less time than has been set, it is repeated within the set cycle time, and the remaining duration is padded with background time.
The processing of the Ethernet and the serial interface is executed in the background time. This should be set to somewhere between 20 and 50 % of the PLC cycle time.

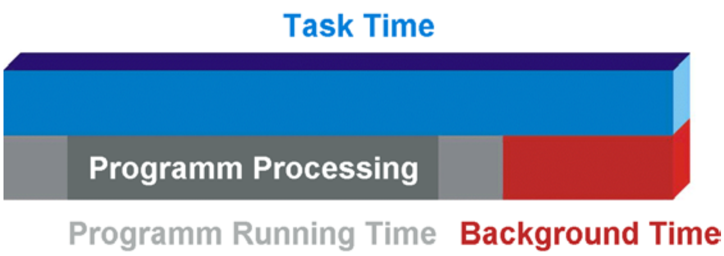
The *mean cycle* time is measured in order to optimise the system. You will find this in PLC Control under the *Online/Coupler* menu item. About 50% is added to the time determined there, and the result entered as the PLC cycle time. Set the background time to between 20 and 50% of the PLC cycle time.





The program running time is composed of the program processing and the K-Bus time. Before the program is called, the Bus Terminal Controller carries out a K-Bus update, and reads the current inputs. After the program has executed, a second K-Bus update is executed in order to write the outputs. The K-Bus time depends on the number and type of Bus Terminals inserted.

The program running time can be shortened by operating only one K-Bus cycle, and this may either be done before or after program processing. In that case, inputs are read and outputs written in a single K-Bus cycle. You can make these settings either with the aid of the KS2000 configuration software, or through ADS.



Remanent and persistent Data

There are special flags for remanent and persistent data in the higher memory region.

Remanent Data

The remanent data is located in the flags area. The default setting is for 64 remanent bytes, i.e. from %MB0 to %MB63. This region can be expanded to a maximum size of 4 kbyte. It should be borne in mind that the task time increases with the quantity of remanent data (approx. 0.5 ms for 512 bytes). The settings for the remanent data can be made via the (TwinCAT) System Manager, or with the KS2000 configuration software (Table 1, Register 15, default value 64 max. 4 kbyte).

Persistent data

Persistent data is even more stable, and is retained even through a program download. The quantity of persistent data must be smaller than or equal to the quantity of remanent data! Like the remanent data, the persistent data is located in the region allocated for flags (%MBxx). You can make the settings for the persistent data using the KS2000 configuration software (Table 1, Register 18, default value 0, max. 4 kbyte).

Example

400 bytes of remanent data, of which 200 bytes are to be persistent
Register 15 400 (%MB200 - %MB399)
Register 18 200 (%MB0 - %MB199)

Cycle Time Measuring

The flags area includes a timing system with a resolution of one millisecond for each digit. The data type is UDINT (unsigned double integer). This value can be overwritten by the program, in order to align with a controller. The timer rolls over in about 48 days.

Flag byte	Meaning	
%MB4088-4091	4 bytes	
	Data type	UDINT
		0 .. 4.22 billion ms / 0 .. about 48 days
	Resolution	1 ms/digit

Diagnostic

It is possible to read the diagnostic data in the Bus Terminal Controller. This information is located in the flags area.

Flag byte	Meaning	
%MB4092-4093	Bit 0	ADS communication watchdog
	Bit 1	ModbusTCP communication watchdog
	Bits 1-14	reserved
	Bit 15	No connection to switch: the *Link* LED does not illuminate. (Transmission of this error information is not active under the factory settings. To activate it, set Register 25 to 1_{bin} in Table 100.)
%MB4094-4095	Bit 0	K-Bus error
	Bit 1	Configuration error
	Bits 2-15	reserved

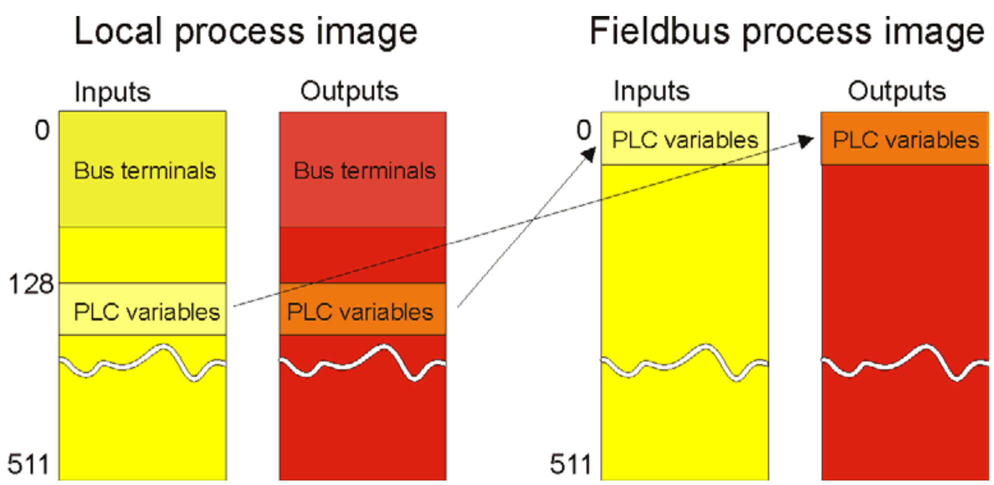
PLC Variables

The PLC variables act as the interface between the local process image of the bus controller and the fieldbus process image for a higher-level controller. This will first be explained on the basis of the default setting.

Bus Terminals in the BC's local process image

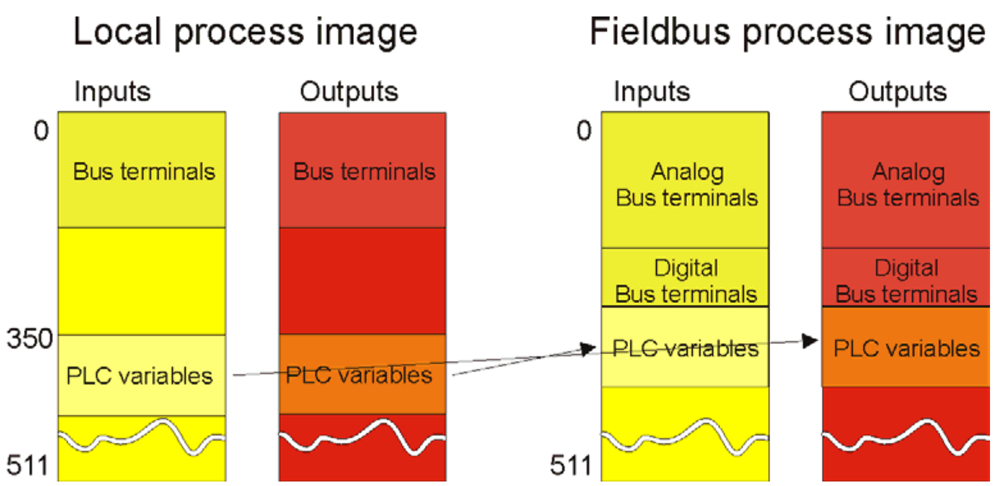
All the connected Bus Terminals are assigned to the local process image. PLC variables are located starting at address 128. You can change both the starting address and the length of the data (default value 16 bytes).

Data that is to be read by a higher level controller is written into the output process image. This is input data for the higher level controller. Data that is to be transferred from the higher level controller to the BC is output data for the controller and input data for the BC. The following diagram may clarify this.



Assign bus terminals to the higher level controller

You can also assign the Bus Terminals directly to the higher level controller. The general scheme of the fieldbus process image is such that the analog Bus Terminals are mapped into this process image first. The digital Bus Terminals then follow, while the PLC variables come last.



Remember that none of the process images in use may be larger than 512 bytes.

Mapping of the Bus Terminals

The precise assignment of the byte-oriented Bus Terminals may be found in the configuration guide for the particular bus terminal. This documentation is available on the Beckhoff *Products & Solutions* CD or on the Internet under http://www.beckhoff.com.

Byte oriented Bus Terminals	Bit oriented Bus Terminals
KL1501	KL10xx, KL11xx, KL12xx, KL17xx
KL25xx	KL20xx, KL21xx, KL22xx, KL26xx, KL27xx
KL3xxx	
KL4xxx	
KL5xxx	
KL6xxx	
KL8xxx	
	KL9110, KL9160, KL9210, KL9260,

Local Process Image

The default setting is for all the connected Bus Terminals to be assigned to the local process image. Mapping within the Bus Terminal Controller is carried out according to the following rule:
First come all the complex Bus Terminals, in whatever sequence they are physically inserted, followed by the digital Bus Terminals which are padded to a whole byte. The default mapping of the complex Bus Terminals is:

- complete evaluation
- Intel format
- word alignment

Example:
1 x BCxxxx
1 1 x KL1012
2 1 x KL1104
3 1 x KL2012
4 1 x KL2034
5 1 x KL1502
6 1 x KL3002
7 1 x KL4002
8 1 x KL6001
9 1 x KL9010

Bus Terminal	Position	Input image	Output image
KL1502	5	%IB0...%IB5	%QB0...%QB5
KL3002	6	%IB6...%IB13	%QB6...%QB13
KL4002	7	%IB14...%IB21	%QB14...%QB21
KL6001	8	%IB22...%IB29	%QB22...%QB29
KL1012	1	%IX30.0..30.1	-
KL1104	2	%IX30.1..30.5	-
KL2012	3	-	%QX30.0..30.1
KL2034	4	-	%QX30.2..30.5
KL9010	9	-	-

Note

If you do not know the address at which the Bus Terminals that you have assigned to the local PLC (BC9000) are located, then configure your hardware in the System Manager. After you have entered all the Bus Terminals and PLC variables, click with the right mouse button on the BC9000 in the hardware tree, and select the menu item *Export variables information...*. A file is saved, and this file can be inserted in the System Manager under *Project - Import*. Now you will have the entry *TwinCAT import* under the global variables, and you will find here all the variables that you have assigned to the local PLC (BC9000).

Fieldbus Process Image

16 bytes input and output are assigned to the fieldbus by default in all the BCxxxx devices. These variables are known as the *PLC variables*. They are located by default in the BCxxxx process image starting at address 128. This address can be shifted. The setting can be made with the KS2000 configuration software, or, for some BCs, can be changed by means of start-up parameters.

ADS Process Image

16 bytes input and output are assigned to the fieldbus by default in all the BCxxxx devices. These variables are known as PLC variables, and are located, under the BCxxxx default setting, starting from address 128 in the process image. This setting can be made with the KS2000 configuration software, or, for some BCs, can be changed by means of start-up parameters.

ModbusTCP Process Image

The ModbusTCP process image makes a fundamental distinction between digital and byte-oriented (word-oriented) signals. This will be clarified by examples.

Example 1: (Default setting)
All the connected Bus Terminals are assigned to the local PLC. The 16 bytes of PLC variables are available for communication with the ModbusTCP master. PLC variables are part of the byte-oriented data packets, and can therefore not be accessed by the digital Modbus functions, even though the variables are digitally mapped in the BCxxxx. Difficulties are most often associated with whether signals are considered as inputs or outputs. PLC variables that are outputs for the BCxxxx are inputs for the Modbus master, and vice versa. The default setting can be changed using the KS2000 software (Table 1 register description).

Local PLC inputs			Local PLC outputs		
Modbus function	**Modbus address/offset**	**Local Variables**	**Modbus function**	**Modbus address/offset**	**Local Variables**
3 read	0x800..807	%IB128..%IB143	3 read	0x000..007	%QB128..%QB143
			4 read	0x000..007	%QB128..%QB143
6 write	0x800..807	%IB128..%IB143			
16 write	0x800..807	%IB128..%IB143			
23 read/write	0x800..807	%IB128..%IB143	23 read/write	0x000..007	%QB128..%QB143

Example 2:
In addition, the Bus Terminals can be assigned once to the BC or to the local PLC. This creates additional complications in the process image. The rule governing how the process image is assembled is, however, simple.
- Entry of all the byte-oriented Bus Terminals (such as the analog Bus Terminals)
- Entry of all the digital Bus Terminals (rounded up to 16 bits)
- Entry of the PLC variables

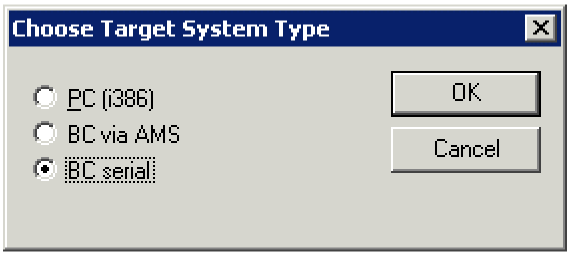
Programming

Transmission via the Serial Interface

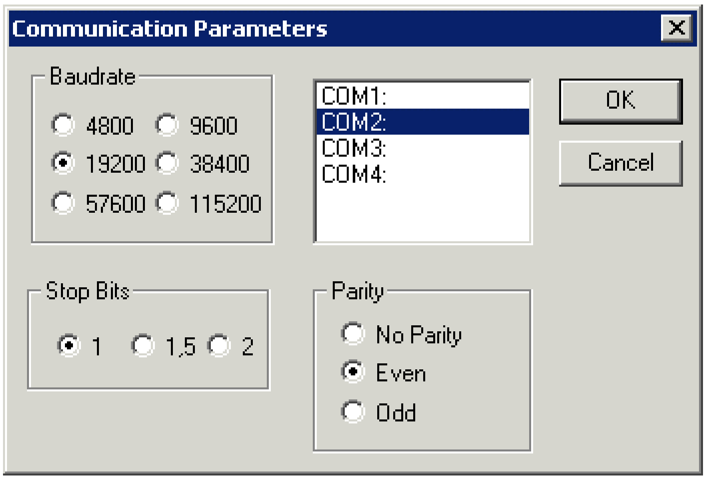
Everything that can be programmed and parameterised with TwinCAT BC can also be handled with TwinCAT. A special feature of TwinCAT BC is that it operates both under Windows 95/98/ME or under NT/2000/XP. Program download via Ethernet, however, is not possible with TwinCAT BC.

Every Bus Terminal Controller can be programmed via the PC's RS232 interface. A special cable to be used for this purpose is supplied along with TwinCAT BC and the KS2000 configuration software.

Select the serial interface in TwinCAT PLC Control.



The settings for the serial interface, port number, baud rate etc. are found under Online/Communication parameters in PLC Control. The Bus Terminal Controller requires the following setting:
Baud Rate: 19200
Stop bits: 1
Parity: even



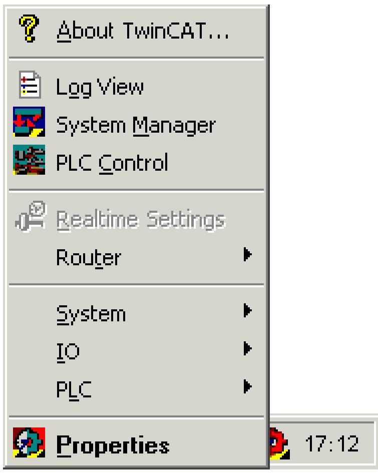
Transmission via Ethernet

TwinCAT offers a facility for transferring the user program to the BC9000 over Ethernet. The BC9000 can be selected as the target system in PLC Control, after saving in the registry and restarting the TwinCAT system. The TwinCAT-level TwinCAT PLC is necessary.

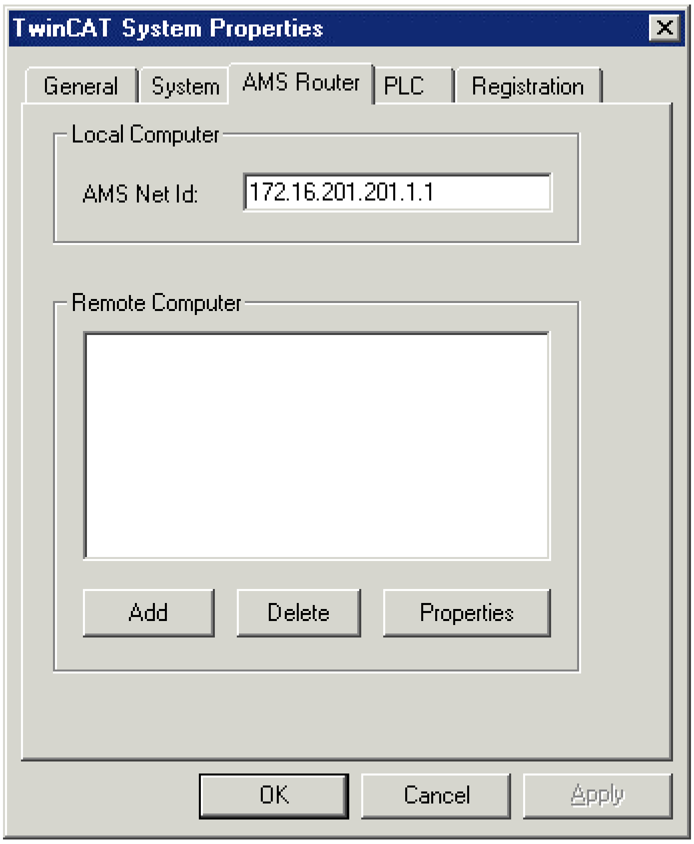
Initialising the BC9000

The coupler must first be made known to the system before it can be selected in PLC Control. There are two ways of doing this. The method you choose to use depends on your needs.

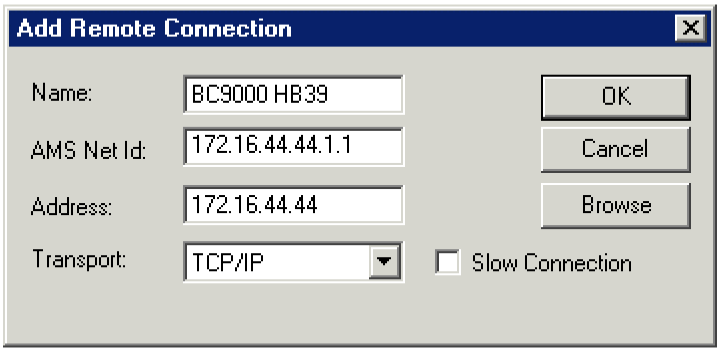
1. You only use TwinCAT PLC for programming.
Click with the left mouse button on the TwinCAT icon in the bottom right corner of the screen.



Select the properties menu item.

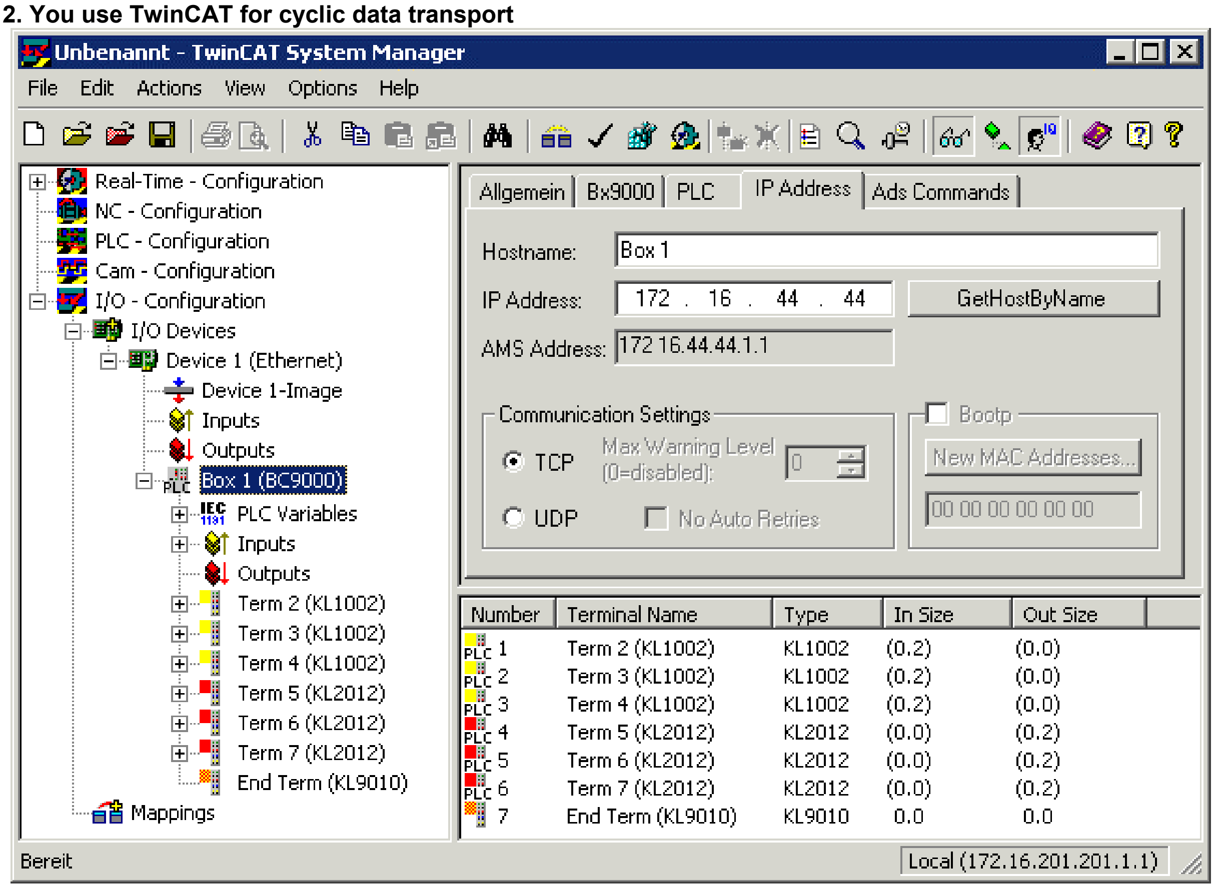


Under the *AMS Router* menu item, enter the BC9000 as a remote computer. Click *Add* to do this.



Enter the name, which you can choose freely. The AMS Net ID is the IP address with an additional ".1.1". This is the IP address set for the Bus Coupler. Confirm with OK. After entering the desired Bus Coupler, you must start the TwinCAT system (the TwinCAT icon is green). The Ethernet coupler is now known to the system, and can be selected in the PLC Control.

2. You use TwinCAT for cyclic data transport



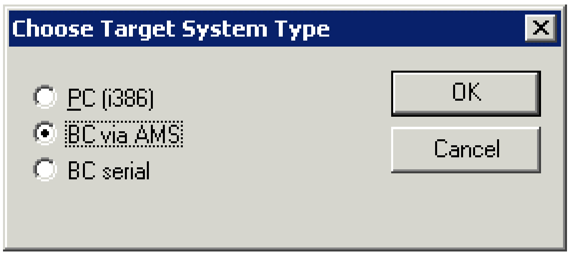
Start the system manager, and enter your configuration there. In contrast to the first method, this configuration (e.g. Bus Terminal mapping, PLC cycle time, amount of retain data etc.) is downloaded to the coupler when TwinCAT starts. Other settings that might have been made with the KS2000 configuration software are overwritten.

PLC control

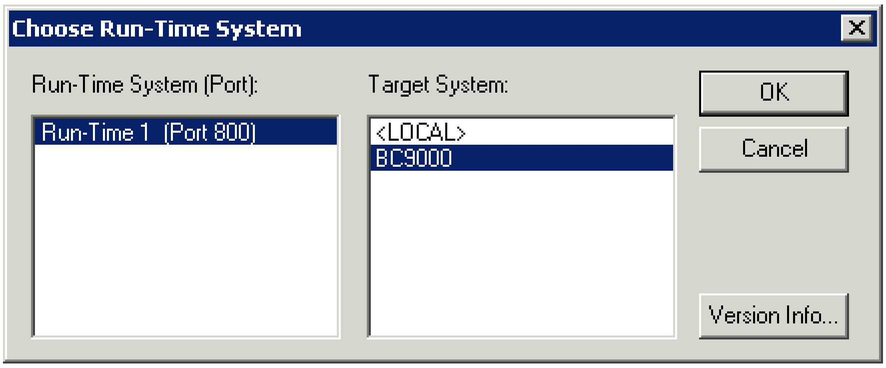
When TwinCAT PLC Control is restarted, TwinCAT asks for the target platform, i.e. the device on which the user program is later to run. TwinCAT offers two target platforms for the controller, the PC or the Bus Terminal Controller.

Two routes are available to you for transmission to the Bus Terminal Controller:

- AMS for communication via the fieldbus
- the serial cable for communication via the PC's RS232 interface and the Bus Terminal Controller's programming interface



After your program has been created, select the target system under the *Online* toolbar. TwinCAT must be running to do this. In the example, this is the Ethernet card with Box 1 and the Runtime 1 of the Bus Terminal Controller.



Libraries

There are a number of libraries for the Bus Terminal Controllers (Bus Couplers with PLC functionality, named BCxxxx) (see the TwinCAT InfoSys).

TwinCAT PLC Library: System BC

The library contains function blocks for access to the Bus Terminal Controller's (BCxxxx) system functions.

TwinCAT PLC Library: Utilities BC

The library contains useful function blocks for the Bus Terminal Controller (BCxxxx). In addition to the RTC block, the library contains a function block for decoding the DCF-77 time signal, along with a number of conversion functions. Internally, the Bus Terminal Controller's system functions are called.

TwinCAT PLC Library: Helper BC

The *PlcHelper.lb6* library contains a number of functions which provide direct access to memory areas in the Bus Terminal Controller's (BCxxxx) PLC runtime system.

TwinCAT PLC Library: ADS BC

The library contains function blocks for acyclic client-server communication between a **BC9xxx** Bus Terminal Controller and other ADS devices in the network.The way in which these blocks operate is not significantly different from that for the PLC runtime system on the PC.

TwinCAT PLC Library: Event driven BC transmission blocks

The present function blocks simplify event-driven data exchange between the Bus Terminal Controllers and/or other ADS devices (TwinCAT NC, Bus Terminal Controllers, ...).
The FB_WriteXXXOnDelta() function blocks implement a write procedure when the input signal rises above or falls below a specified limit value. The frequency with which the input signal is examined can be set. Event-driven data writing minimises the loading on the fieldbus. If an error occurs during transmission, the process is repeated until the connection is established once more. All data types supported in the TwinCAT PLC are permitted as source and destination variables. Symbol names are also supported.
Watchdog blocks are available to monitor individual communication partners. The device that is to be monitored cyclically transmits an incrementing counter. A check is made at the receiver to see that the counter state changes within a specific time.

TwinCAT PLC Library: ModbusTCP BC

The library contains function blocks that permit the exchange of data between the Bus Terminal Controller (BC9xxx) and a remote partner on the MODBUS/TCP port.

TwinCAT PLC Library: SMTP BC

The library contains useful function blocks for the Bus Terminal Controller (BC9xxx) for use of the SMTP protocol (**S**imple **M**ail **T**ransfer **P**rotocol).

7. Fieldbussystem

Ethernet

Ethernet was originally developed by DEC, Intel and XEROX (as the "DIX" standard) for passing data between office devices. The term nowadays generally refers to the *IEEE 802.3 CSMA/CD* specification, published in 1985. Because of the high acceptance around the world this technology is available everywhere and is very economical. This means that it is easy to make connections to existing networks.

There are now a number of quite different transmission media: coaxial cable (10Base5), optical fibre (10BaseF) or twisted pairs (10BaseT) with screen (STP) or without screen (UTP). A variety of topologies such as ring, line or star can be constructed with Ethernet.

Ethernet transmits Ethernet packets from a sender to one or more receivers. This transmission takes place without acknowledgement, and without the repetition of lost packets. To achieve reliable data communication, there are protocols, such as TCP/IP, that can run on top of Ethernet.

MAC-ID

The sender and receiver of Ethernet packets are addressed by means of the MAC-ID. The MAC-ID is a 6 byte identification code unique to every Ethernet device in the world. The MAC-ID consists of two parts. The first part (i.e. the first 3 bytes) is a manufacturer identifier. The identifier for Beckhoff is 00 01 05. The next 3 bytes are assigned by the manufacturer and implement a unique serial number. The MAC-ID can, for example, be used for the BootP protocol in order to set the TCP/IP number. This involves sending a telegram containing the information such as the name or the TCP/IP number to the corresponding node. You can read the MAC-ID with the KS2000 configuration software.

The Internet Protocol (IP)

The internet protocol (IP) forms the basis of this data communication. IP transports data packets from one device to another; the devices can be in the same network, or in different networks. IP here looks after the address management (finding and assigning MAC-IDs), segmentation and routing. Like the Ethernet protocol, IP does not guarantee that the data is transported - data packets can be lost, or their sequence can be changed.

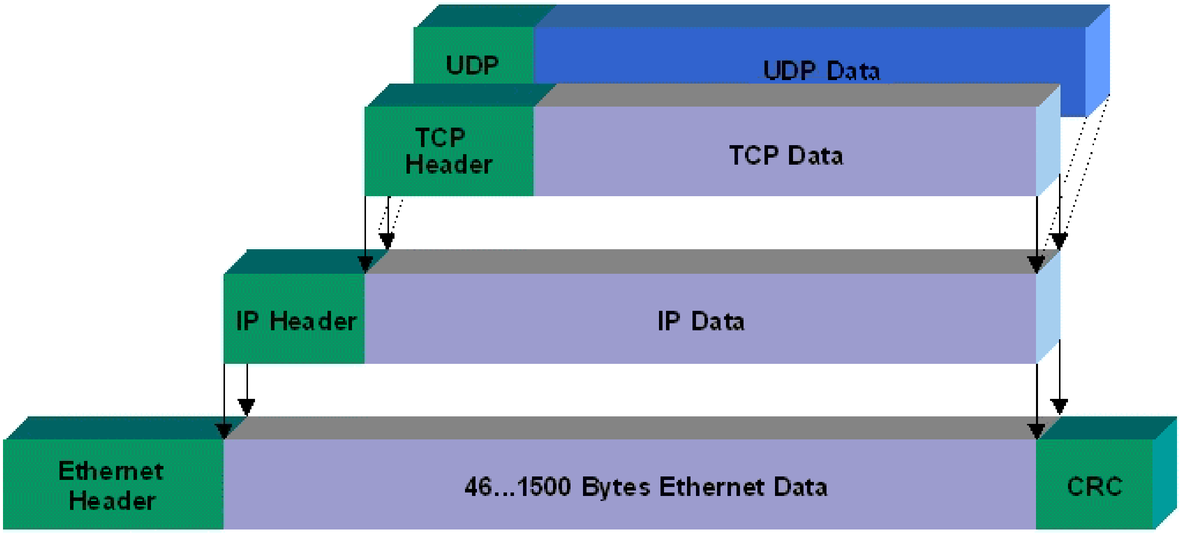
TCP/IP was developed to provide standardised, reliable data exchange between any number of different networks. TCP/IP is thus substantially independent of the hardware or software being used. Although the term is often used as if it were a single concept, a number of protocols are layered together: e.g. IP, TCP, UDP, ARP and ICMP.

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) which runs on top of IP is a connection-oriented transport protocol. It includes error detection and error handling mechanisms. Lost telegrams are repeated.

User Datagram Protocol (UDP)

UDP is connectionless transport protocol. It provides no control mechanism when exchanging data between sender and receiver. This results in a higher processing speed than, for example, TCP. Checking whether or not the telegram has arrived must be carried out by the higher-level protocol.

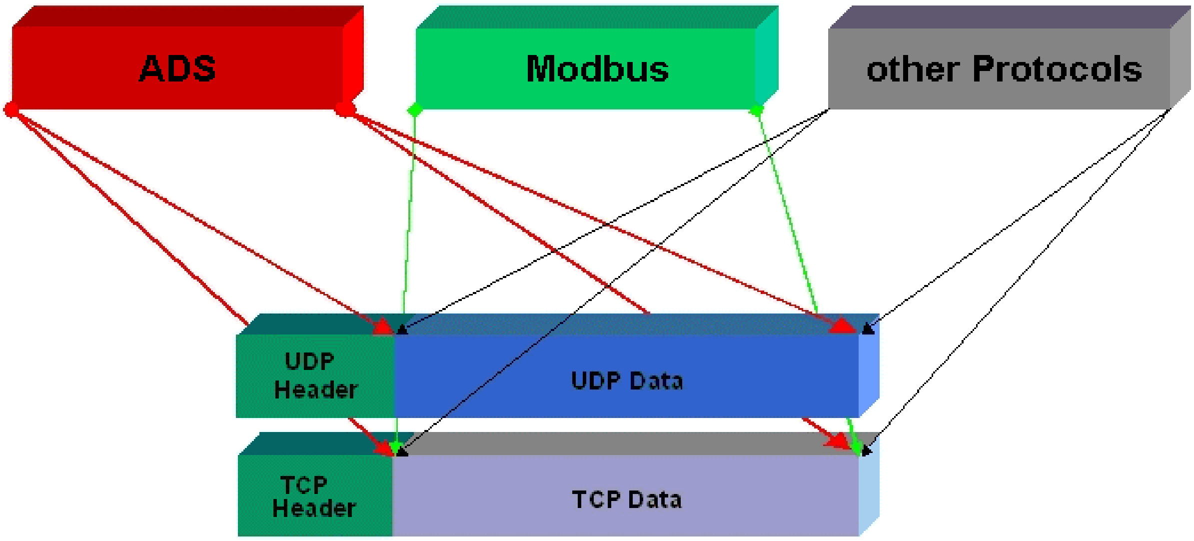


Protocols running on top of TCP/IP and UDP/IP

The following protocols can run on top of TCP/IP or UDP:

- ADS
- ModbusTCP

Both of these protocols are implemented in parallel on the Bus Coupler, so that no configuration is needed to activate the protocols.



ADS can be used on top of either TCP or UDP, but ModbusTCP is always based on TCP/IP.

Topology

In 10BaseT and 100BaseT a number of stations are star connected according to the Ethernet standard.

Star topology

The simplest form of a star LAN consists of a single point-to-point connection. All messages pass via a central node (the hub or switch), which then passes the information to the desired device according to the destination address.

Tree topology

A tree topology consists of a number of connected star networks. As soon as the network contains a number of hubs or switches, the topology is classified as a tree. Ideally the connections between the star couplers have a particularly wide bandwidth, since these transport the most traffic. When constructing tree topologies, the repeater rule must be observed. This is also known as the 5-4-3 repeater rule. There must be no more than two pairs of repeaters (or of hubs) in the transmission path between any two stations, unless they are separated by bridges, switches or routers. A transmission path may consist of at most five segments and four repeater sets (two repeater pairs). Up to three of these segments may be coaxial segments to which the stations are connected. The remaining segments must consist of point-to-point connections; these are also known as IRL (inter repeater link) connections.

Cabling guidelines

Structured cabling provides general guidelines for constructing the cabling for a LAN. It specifies maximum permitted cable lengths for the wiring within the grounds, building or floor. Standardised in EN 50173, ISO 11801 and TIA 568-A, *structured cabling* provides the basis for an advanced, application-independent and economical network infrastructure. The wiring standards are applicable to a range defined as having a geographical extent of up to 3 km and an office area of up to one million sq metres, with between 50 and 50,000 end devices. Recommendations for the structure of a cabling system are also given. The figures can vary, depending on the topology selected, the transmission media and coupling modules used under industrial conditions, and on the use of components from various manufacturers in one network. The given figures should therefore only be considered as recommendations.

BC9000 Reaction Time

ADS	Time [ms]
TCP	23 to 50 ms + task time on the BC9000 (jitter)
UDP	2 to 3 ms + task time on the BC9000 (jitter)

Modbus	Time [ms]
TCP	12 to 15 ms + task time on the BC9000 (jitter) (default)
Fast TCP*	1 to 3 ms + task time on the BC9000 (jitter)

* as from firmware B7, Table 100, Register 29 - "1" Fast TCP / "0" TCP (see Modbus Interface)

BK9000 reaction time

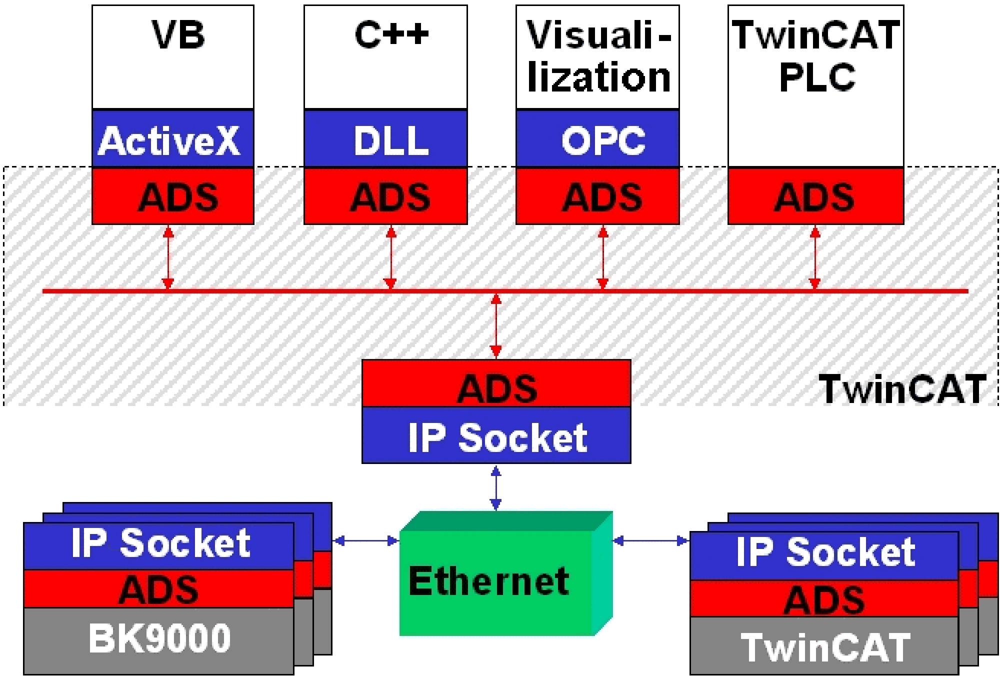
ADS	Time [ms]
TCP	23 to 50 ms
UDP	2 to 3 ms

Modbus	Time [ms]
TCP	12 to 15 ms (default)
Fast TCP*	1 to 3 ms

* as from firmware B5, Table 100, Register 29 - "1" Fast TCP / "0" TCP (see Modbus Interface)

ADS Communication

The ADS protocol (ADS: Automation Device Specification) is a transport layer within the Beckhoff TwinCAT system. It was developed for data exchange between the different software modules, for instance the communication between the NC and the PLC. This protocol offers the freedom of using other tools to communicate with any point in TwinCAT. If it is necessary to communicate with another PC or device, the ADS protocol is used on top of TCP/IP. This means that in a networked system, all the data is accessible from any desired point.



The ADS protocol runs on top of the TCP/IP or UDP/IP protocols. It allows the user within the Beckhoff system to use almost any connecting route to communicate with all the connected devices and to parameterise them. Outside the Beckhoff system a variety of methods are available to exchange data with other software tools.

Software interfaces

ADS-OCX

The ADS-OCX is an Active-X component. It offers a standard interface to, for instance, Visual Basic, Delphi, etc.

ADS-DLL

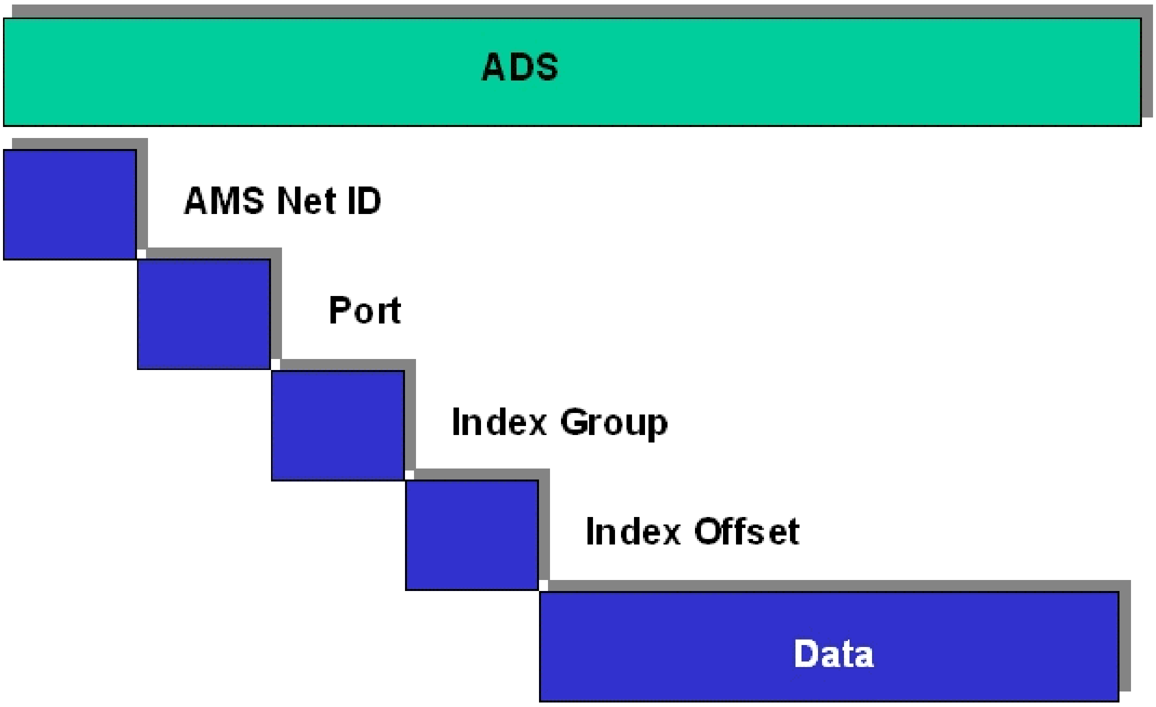
You can link the ADS-DLL (DLL: Dynamic Link Library) into your C program.

OPC

The OPC interface is a standardised interface for communication used in automation engineering. Beckhoff offer an OPC server for this purpose.

ADS Protocol

The ADS functions provide a method for accessing the Bus Coupler information directly from the PC. ADS function blocks can be used in TwinCAT PLC Control for this. The function blocks are contained in the *PLCSystem.lib* library. It is also equally possible to call the ADS functions from AdsOCX, ADSDLL or OPC. It is possible to access all the data through ADS port number 300, and to access the registers of the Bus Coupler and Bus Terminals through ADS port number 100.



AMSNetID

The AMSNetID provides a reference to the device that is to be addressed. This is created from the set TCP/IP address and an additional 2 bytes. These additional 2 bytes consist of "1.1", and can not be altered.
Example:
IP address 172.16.17.128
AMS Net ID 172.16.17.128.1.1

Port number

The port number distinguishes sub-elements in the connected device.
Port 100: Register access
Port 300: Fieldbus process data
Port 800: Local process data (BC90x0 only)

Index group

The index group distinguishes different data within a port.

Index offset

Indicates the offset, the byte from which reading or writing is to start.

Len

Gives the length of the data, in bytes, that that is to be read or written.

TCP port number

The TCP port number for the ADS protocol is 48898 or 0xBF02.

ADS Services

Process data port 300

The fieldbus data is accessed via ADS port number 300. The data is monitored by a watchdog. If no further telegram arrives within 1000 ms the outputs will be switched to the safe state.

Index group	Meaning	Index offset (value range)
0xF020	Inputs	0..511
0xF030	Outputs	0..511

Local process image port 800 (BC9000 only)

Data can be read from and written to the local process image. If it is necessary for outputs to be written, it is important to ensure that they are not used by the local PLC, because the local controller will overwrite these values. The data is not associated with a watchdog, and therefore must not be used for outputs that would have to be switched off in the event of a fault.

Index group	Meaning	Index offset (value range)
0xF020	Inputs	0..511
0xF030	Outputs	0..511
0x4020	Flags (BC 9000 only)	0..4096

ADS services

AdsServerAdsState**

Data type (read only)	Meaning
String	Start - the local PLC is running Start - the local PLC is stopped

AdsServerDeviceState**

Data type (read only)	Meaning
INT	0 – Start - the local PLC is running 1 – Stop - the local PLC is stopped

AdsServerType**

Data type (read only)	Meaning
String	Coupler_PLC

Register port 100

The ADS port number in the BK/BC9000 for register communication is fixed, being set at 100.

Index group	Index offset (value range)		Meaning
	Hi-Word	Low Word	
0	0..127	0..255	Registers in the Bus Coupler High word, table number of the Bus Coupler Low word, register number of the table
1-64	0-3	1-64	Register of the Bus Terminal High word, channel number Low word, register number of the Bus Terminal

Note	Note when reading the register that the time out for the ADS block is set to a time longer than 1 second.

Note	Note when writing to the registers that the password is set (see the documentation for the particular Bus Terminal).

Access control*

The AMS *Net-Id* table permits access control to the BC 9000 via AMS. As soon as this table has entries, only those AMS devices that have been entered will be able to access the BC 9000. An explicit association of the MAC-ID and the IP address of the node is also made here.
The *AMS Net-Id* table can be filled with ADS write commands:
A maximum of 10 entries is possible.

The structure

AMS Net ID	Size
AMS Net ID	6 bytes
IP address	4 bytes
Reserve	2 bytes
Reserve	4 bytes
Reserve	4 bytes

Access takes place via port number: 10.000

Index group: 700

Index Offset (Write)	Comment	Data
0	Add an entry	Data structure, 20 bytes
1	Delete an entry	-
2	Delete all entries	-
10	Save the table in Flash memory	-

Index Offset (Read)	Comment	Data
0	Number of entries	2 bytes
1..10	Entry n (1..10)	Data structure, 20 bytes

Note	The first entry must be that for the device that his writing into the table, because the settings have immediate effect. Make sure that all the settings are correct. The table can also be deleted if the end terminal only is inserted, and the DIP switches 1 to 7 are set to ON.

* as from firmware B6
** as from firmware B7

See the example

Access Control Example

```
PROGRAM MAIN
VAR
	i: INT;
	strAc121:ARRAY[0..19] OF BYTE:=172,16,2,121,1,1,172,16,3,0,0,0,0,0,0,0,0,0,0,0;
					(*AMS Net ID + IP-Address + 0000000000*)
	strAc122:ARRAY[0..19] OF BYTE:=172,16,2,122,1,1,172,16,2,123,0,0,0,0,0,0,0,0,0,0;
					(*AMS Net ID + IP-Address + 0000000000*)
	fbADSWRITE1: ADSWRITE;
	Save_in_Flash:BOOL:=FALSE;		(* Save this access True = yes , False :=no*)
	TableReset: BOOL;			(* Reset Table *)
	fbFlankeTableReset: R_TRIG;
END_VAR
```

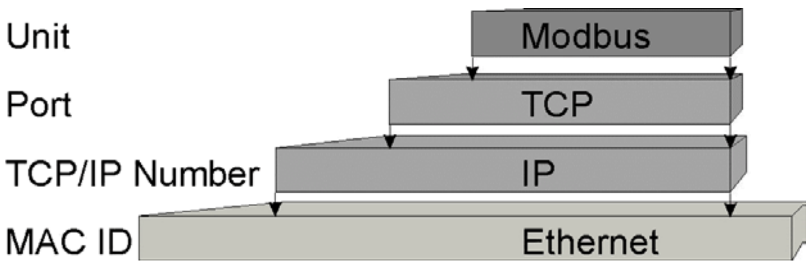
```
fbFlankeTableReset(CLK:=TableReset);
IF i=100 AND fbFlankeTableReset.Q THEN
	i:=0;
END_IF

CASE i OF
0:	IF TableReset THEN
		i:=500;
	ELSE
		i:=1;
	END_IF
1:	fbADSWRITE1(NETID:='172.16.17.151.1.1' , PORT:=10000 , IDXGRP:=700 , IDXOFFS:=0 ,
		LEN:=SIZEOF(strAc121) , SRCADDR:=ADR(strAc121) , WRITE:=TRUE , TMOUT:=t#1s  );
	IF NOT fbADSWrite1.Busy THEN
		i:=i+1;
	END_IF
2:	fbADSWRITE1(Write:=FALSE);
	IF fbADSWRITE1.Err THEN
		i:=200;
	ELSE
		i:=i+1;
	END_IF
3:	fbADSWRITE1(NETID:='172.16.17.151.1.1' , PORT:=10000 , IDXGRP:=700 , IDXOFFS:=0 ,
		LEN:=SIZEOF(strAc122) , SRCADDR:=ADR(strAc122) , WRITE:=TRUE , TMOUT:=t#1s  );
	IF NOT fbADSWrite1.Busy THEN
		i:=i+1;
	END_IF
4:	fbADSWRITE1(Write:=FALSE);
	IF fbADSWRITE1.Err THEN
		i:=200;
	ELSE
		IF Save_in_Flash THEN
			i:=i+1;
		ELSE
			i:=100;
		END_IF
	END_IF
5:	fbADSWRITE1(NETID:='172.16.17.151.1.1' , PORT:=10000 , IDXGRP:=700 , IDXOFFS:=10 , WRITE:=TRUE ,
		 TMOUT:=t#1s  );
	IF NOT fbADSWrite1.Busy THEN
		i:=i+1;
	END_IF
6:	fbADSWRITE1(Write:=FALSE);
	IF fbADSWRITE1.Err THEN
		i:=200;
	ELSE
		i:=100;
	END_IF
100: ;	(*End without error*)
200: ;	(*Error*)
```

```
(* Reset Table and Save*)
500:        fbADSWRITE1(NETID:='172.16.17.151.1.1' , PORT:=10000 , IDXGRP:=700 , IDXOFFS:=2 ,
                                WRITE:=TRUE , TMOUT:=t#1s  );
            IF NOT fbADSWrite1.Busy THEN
                        i:=i+1;
            END_IF
501:        fbADSWRITE1(Write:=FALSE);
            IF fbADSWRITE1.Err THEN
                        i:=200;
            ELSE
                        i:=502;
            END_IF
502:        fbADSWRITE1(NETID:='172.16.17.151.1.1' , PORT:=10000 , IDXGRP:=700 , IDXOFFS:=10 ,
                                WRITE:=TRUE , TMOUT:=t#1s  );
            IF NOT fbADSWrite1.Busy THEN
                        i:=i+1;
            END_IF
503:        fbADSWRITE1(Write:=FALSE);
            IF fbADSWRITE1.Err THEN
                        i:=200;
            ELSE
                        i:=100;
            END_IF
END_CASE
```

ModbusTCP

ModbusTCP Protocol

The Ethernet protocol is addressed by means of the MAC-ID. The user does not normally need to be concerned about this address. The IP number has a length of 4 bytes, and must be parameterised by the user on the Bus Coupler and in the application. In ModbusTCP, the TCP port is set to 502. The UNIT can be freely selected under ModbusTCP, and does not have to be configured by the user.



TCP port number

The TCP port number for ModbusTCP has been standardised to 502. This TCP port number can also be changed with the KS2000 configuration software: table 100, register 26.

Modbus-Unit

The unit is only of any significance if a number of stations are accessing the BK9000. The first unit that accesses the Bus Coupler has write access. All the other devices can only read from the BK9000.

ModbusTCP Protocol

Byte	Name	Description
0	Transaction identifier	Is returned by the slave
1	Transaction identifier	Is returned by the slave
2	Protocol identifier	Always 0
3	Protocol identifier	Always 0
4	Length field	0 (if the message is less than 256 bytes in length)
5	Length field	Number of following bytes
6	UNIT identifier	This should be understood as a protocol number, and is returned by the slave
7	Modbus	The Modbus protocol with the function follows

Modbus TCP interface

Address		Description		
0x0000 0x00FF		Process data interface Inputs		
0x0800 0x08FF		Process data interface Outputs		
0x1000 0x1006	Read only	Bus Coupler identification		
0x100A		2 byte PLC interface		
0x100B		Bus terminal diagnosis		
0x100C		Bus Coupler status		
0x1010		Process image length in bits, analog outputs (without PLC variables)		
0x1011		Process image length in bits, analog inputs (without PLC variables)		
0x1012		Process image length in bits, digital outputs		
0x1013		Process image length in bits, digital inputs		
0x1020		Watchdog, current time in [ms]		
0x110A	Read/Write	2 byte PLC interface		
0x110B		Bus terminal diagnosis		
0x1120		Watchdog, pre-defined time in [ms] (Default value: 1000)		
0x1121		Watchdog Reset Register		
0x1122		Type of watchdog	1	Telegram watchdog (default)
			0	Write telegram watchdog
0x1123		ModbusTCP mode**	1	Fast Modbus
			0	Normal Modbus (default)
0x4000 0x47FF		Flags area (%MB..)*		

* BC 9000 only
** BC9000 B7 - BK9000 B5 firmware

Watchdog

The watchdog is active under the factory settings. After the first write telegram the watchdog timer is initiated, and is triggered each time a telegram is received from this device. Other devices have no effect on the watchdog. A second approach, which represents a more sensitive condition for the watchdog, is for the watchdog only to be re-triggered after each write telegram. To do this, write a zero into register 0x1122 (default value "1").
The watchdog can be deactivated by writing a zero to offset 0x1120. The watchdog register can only be written if the watchdog is not active. The data in this register is retained.

Watchdog register
If the watchdog timer on your slave has elapsed it can be reset by writing twice to register 0x1121. The following must be written to the register: 0xBECF 0xAFFE. This can be done either with function 6 or with function 16.

The Bus Coupler's status register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FB	-	-	-	-	-	-	-	-	-	-	-	-	-	CNF	KB

Key

Bit	Name	Value	Description
15	FB	1_{bin}	Fieldbus error, watchdog time elapsed
14...2	-	-	reserved
1	CNF	1_{bin}	Bus Coupler configuration error
0	KB	1_{bin}	Bus Terminal error

ModbusTCP mode

The fast Modbus mode should only be used in small local networks. The fast ModbusTCP is not active under the default settings. If problems are found to occur with this type of communication, the Bus Coupler should be switched to "normal" ModbusTCP communication. The mode is set in the Modbus interface, offset 0x1123. It is necessary to reset the coupler (e.g. using ModbusTCP function 8) after the change. It is not permitted to send more than one Modbus service within one Ethernet frame in fast Modbus mode.

2 byte PLC interface

Registers in the complex terminals and Bus Terminal Controller registers can be both read and written using the 2 byte PLC interface. The complex terminal registers are described in the associated terminal documentation. The Bus Coupler registers can be used, for example, to read terminal bus diagnostics data, the terminal composition or the cycle times, and the programmed configuration can be written. It is also possible for a manual terminal bus reset to be carried out. The 2-byte PLC interface requires two bytes each of input and output data; a special protocol is processed via these 2 bytes. A description of the 2 byte PLC interface, the registers available in the Bus Couplers and of function blocks for various PLC systems that support the 2 byte PLC interface can be supplied on request.

2 byte diagnostic interface

The terminals' error messages can be sent over the 2-byte diagnostic interface. Terminal bus diagnostics must however be activated for this purpose. The 2-byte diagnostic interface occupies two bytes each of output and input data. A special protocol is processed via these two bytes. A description of the 2 byte-diagnostic interface can be supplied on request.

ModbusTCP slave error answer (BK9000, BC9000, IP/IL230x-B/C900)

When the user sends the slave either a request or information that the coupler does not understand, the slave responds with an error report. This answer contains the function and the error code. 0x80 is added to the value returned by the function.

Code	Name	Meaning
1	ILLEGAL FUNCTION	Modbus function not implemented
2	ILLEGAL DATA ADDRESS	Invalid address or length
3	ILLEGAL DATA VALUE	Invalid parameter - Diagnostic functions - Incorrect register
4	SLAVE DEVICE ERROR	Watchdog or K-Bus error
6	SLAVE DEVICE BUSY	Output data is already been received from another IP device

ModbusTCP Functions

In the Modbus protocol, the functions determine whether data is to be read or written, and what kind of data is involved.

Function	Code	Description
Read coil status	1	Read digital outputs
Read input status	2	Read digital inputs
Read holding register	3	Read analog outputs and inputs / GPR
Read input register	4	Read analog inputs / GPR
Force single coil	5	Writing a digital output
Preset single register	6	Write one analog output / GPR
Diagnostic	8	Diagnostic
Force multiple coils	15	Write a number of digital outputs
Preset multiple register	16	Write a number of analog outputs / GPRs
Read / Write Registers	23	Write and read a number of process data outputs / GPRs

GPR (General Preset Register) - register structure of the Modbus interface (see Appendix)

Read Coil Status (Function 1)

The *Read Coil Status* function can be used to read the digital outputs that have been set.

The first 10 digital outputs are read in this example. The start address is zero. An offset can be entered in the *Start address* field

Query

Byte Name	Example
Function code	1
Start address high	0
Start address low	0
Count high	0
Count low	10

The fieldbus coupler answers with *byte count* 2, i.e. 2 bytes of data are returned. The query was for 10 bits, and these are now distributed over 2 bytes. The third bit in the output process image of the BK9000 is set, and the fieldbus coupler returns the value 4 in the first data byte.

Response

Byte Name	Example
Function code	1
Byte Count	2
Data bits 0...7	4
Data bits 8...18	0

Read Input Status (Function 2)

The *Read Input Status* function can be used to read the digital input data. The first 10 digital inputs are read in this example. The start address is zero. An offset can be entered in the *Start address* field

Query

Byte Name	Example
Function code	2
Start address high	0
Start address low	0
Count high	0
Count low	10

The fieldbus coupler answers with *Byte count* 2, i.e. two bytes of data are returned. The query was for 10 bits, and these are now distributed over 2 bytes. The third bit in the output process image of the BK9000 is set, and the fieldbus coupler returns the value 4 in the first data byte.

Response

Byte Name	Example
Function code	2
Byte Count	2
Data bits 0...7	1
Data bits 8...18	0

Read Holding Register (Function 3)

The *Read Holding Register* function can be used to read the input and output words and the registers. The inputs have offsets 0 - 0xFF while the outputs have offsets 0x800 - 0x8FF.

The first two analog outputs are read in this example. The analog outputs begin at offset 0x800. The length indicates the number of channels to be read.

Query

Byte Name	Example
Function code	3
Start address high	8
Start address low	0
Count high	0
Count low	2

The fieldbus coupler answers with byte count 4, i.e. 4 bytes of data are returned. The query was for two analog channels, and these are distributed over two words. In the analog output process image, the first channel has the value 0x3FFF, while the second channel has the value 0x0.

Response

Byte Name	Example
Function code	3
Byte Count	4
Data 1 high byte	63
Data 1 low byte	255
Data 2 high byte	0
Data 2 low byte	0

Read Input Register (Function 4)

The *Read Input Register* function reads the analog inputs.

In this example the first two analog inputs of slave number 11 are read. The analog outputs start at an offset of 0x0000. The length indicates the number of words to be read. A KL3002 has two words of input data, which is why the value to be entered in *Count low* is two.

Query

Byte Name	Example
Function code	4
Start address high	0
Start address low	0
Count high	0
Count low	2

The fieldbus coupler answers with byte count 4, i.e. four bytes of data are returned. The query was for two analog channels, and these are now distributed over 2 words. In the analog input process image, the first channel has the value 0x0038, while the second channel has the value 0x3F1B.

Response

Byte Name	Example
Function code	4
Byte Count	4
Data 1 high byte	0
Data 1 low byte	56
Data 2 high byte	63
Data 2 low byte	11

Force Single Coil (Function 5)

The *Force Single Coil* function can be used to write a digital output. The third digital output is written in this example. The digital outputs start at address 0x0000. The digital value is located in *Data high*. To switch the output on, *Data high* must contain the value 0xFF, while 0x00 is used to switch the output off again. *Data low* must contain the value 0x00.

Query

Byte Name	Example
Function code	5
Start address high	0
Start address low	2
Data high	255
Data low	0

The coupler answers with the same telegram.

Response

Byte Name	Example
Function code	5
Start address high	0
Start address low	2
Data high	255
Data low	0

Preset Single Register (Function 6)

The *Preset Single Register* function can be used to access the output process image and the interface.

The first analog output is written with function 6. The analog outputs start at an offset of 0x0800. Here again the offset always describes a word. This means offset 0x0803 refers to the fourth word in the output process image.

Query

Byte Name	Example
Function code	6
Start address high	8
Start address low	0
Data high	63
Data low	255

The Fieldbus Coupler replies with the same telegram and confirmation of the received value.

Response

Byte Name	Example
Function code	6
Start address high	8
Start address low	0
Data high	63
Data low	255

Diagnosis (Function 8)

The diagnosis function provides a series of tests for checking the communication system between the master and the slave and for examining a variety of internal error states within the slave. A broadcast telegram is not supported.

The function uses two bytes in the query to specify a subfunction code defining the test that is to be carried out. The slave returns the function code and the subfunction code in the response.

The diagnostic queries use a two-byte data field to send diagnostics data or control information to the slave.

Query

Byte Name	Example
Function code	8
Subfunction high	0
Subfunction low	0
Data high	2
Data low	3

Response

Byte Name	Example
Function code	8
Subfunction high	0
Subfunction low	0
Data high	2
Data low	3

Echo a request (Subfunction 0)

Subfunction 0 causes the data that is sent to the slave by the master to be returned.

Coupler reset (Subfunction 1)

Subfunction 1 re-initialises the BC9000. Error counters are reset, and the Bus Terminal Controller executes a self-test. No telegrams are either received or sent while the Bus Terminal Controller is being reset. The IP socket is closed.

Note	Before the Bus Terminal Controller restarts it sends one more answer with subfunction 1, after which the IP socket is closed.

Subfunction	Data field (query)	Data field (response)
0x0001	0x0000	0x0000

Delete All Counter Contents (Subfunction 10)

Calling this subfunction deletes the contents of all error counters in the Bus Terminal Controller.

Subfunction	Data field (query)	Data field (response)
0x000B	0x0000	Echo query data

Bus Communication Answer Counter (Subfunction 11)

Returns the number of communication answers.

Subfunction	Data field (query)	Data field (response)
0x000C	0x0000	Counter value

Error Answer Counter (Subfunction 13)

This counter contains the number of error answer telegrams that the coupler has sent.

Subfunction	Data field (query)	Data field (response)
0x000D	0x0000	Counter value

The following functions contain the counter states for various units. This means that the Modbus telegrams can be distinguished through their units, if, for example, a Bus Coupler is accessed by different masters.

Slave Answers (Subfunction 14)

Contains the number of answers that the slave has sent.

Subfunction	Data field (query)	Data field (response)
0x000E	0x0000	Counter value

Number of unanswered telegrams (Subfunction 15)

Contains the number of answers that the slave has not sent.

Subfunction	Data field (query)	Data field (response)
0x000F	0x0000	Counter value

Number of Error Answers (Subfunction 16)

Contains the number of error answers that the slave has sent.

Subfunction	Data field (query)	Data field (response)
0x0010	0x0000	Counter value

Force Multiple Coils (Function 15)

The *Force multiple coils* function can be used to set or reset a number of digital outputs at the same time.

The first 20 digital outputs are written in this example. The digital outputs start at an offset of 0x0000. Here the offset always describes a bit. Offset 0x0003 writes to the fourth bit in the output process image. The length indicates the number of bits, and the *Byte count* is formed from the combination all the bytes that are to be written.

Example: 20 bits yield a byte count of 3 (rounded up to a byte boundary).

The data bytes contain the values for the individual bits. In this example, bits 0 to 15 are set to TRUE, while bits 16 to 23 are FALSE.

Query

Byte Name	Example
Function code	15
Start address high	0
Start address low	0
Length high	0
Length low	20
Byte Count	3
Data 1 bit 0...7	255
Data 2 bit 8...15	255
Data 3 bit 16...23	0

Response

The Bus Coupler answers with the same telegram.

Byte Name	Example
Function code	15
Start address high	0
Start address low	0
Length high	0
Length low	20

Preset Multiple Register (Function 16)

The *Preset Multiple Register* function can be used to write a number of analog outputs. The first two analog output words are written in this example. The analog outputs start at an offset of 0x0800. Here the offset always describes a word. Offset 0x0003 writes to the fourth word in the output process image. The length indicates the number of words, and the *Byte count* is formed from the combination of all the bytes that are to be written.

Example: 4 words – correspond to a byte count of 8

The data bytes contain the values for the analog outputs. In this example, two words are to be written. The first word is to receive the value 0x7FFF, and the second word is to receive the value 0x3FFF.

Query

Byte Name	Example
Function code	16
Start address high	8
Start address low	0
Length high	0
Length low	2
Byte Count	4
Data 1 byte 1	127
Data 1 byte 2	255
Data 2 byte 1	63
Data 2 byte 2	255

Response

The coupler replies with the start address and the length of the transmitted words.

Byte Name	Example
Function code	16
Start address high	8
Start address low	0
Length high	0
Length low	2

Read / Write Registers (Function 23)

A number of analog outputs can be written and a number of analog inputs read with one telegram using the *Read / Rrite Registers* function. In this example the first two analog output words are written, and the first two analog inputs are read. The analog outputs start at offset 0x0800, while the inputs start at offset 0x0000. Here the offset always describes a word. Offset 0x0003 writes to the fourth word in the output process image. The length indicates the number of words, and the *Byte count* is formed from the combination of all the bytes that are to be written. Example: 4 words – correspond to a byte count of 8

The data bytes contain the values for the analog outputs. In this example, two words are to be written. The first word is to receive the value 0x3FFF, and the second word is to receive the value 0x7FFF.

Query

Byte Name	Example
Function code	23
Read start address high	0
Read start address low	0
Read length high	0
Read length low	2
Write start address high	8
Write start address low	0
Write length high	0
Write length low	2
Byte Count	4
Data 1 high	63
Data 1 low	255
Data 2 high	127
Data 2 low	255

Response

The coupler replies with the start address and the length of the bytes to be transferred in *Byte count*. The data information follows. In this example the first word contains 0x0038 while the second word contains 0x3F0B.

Byte Name	Example
Function code	23
Byte Count	4
Data 1 high	0
Data 1 low	56
Data 2 high	63
Data 2 low	11

Examples for ModbusTCP

Examples for Modbus TCP are contained in this help text.

Simple example: ModbusTCP via VB6.0
Extract ModbusTCP.zip

Example: ModbusTCP via VB6.0, set and reset of the watchdog:
Extract BK9000Modbus.zip

Description of the Parameters

Busterminal Controller Properties

Register Settings, Table 0

The registers can be set through a dialog using the KS2000 configuration software, or it is possible to write into the registers directly.

Register	Description	Default	Bus Terminal Controller
19	Byte offset digital inputs %I...	0	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
20	Byte offset digital outputs %Q...	0	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000

Register Settings, Table 1

The registers can be set through a dialog using the KS2000 configuration software, or it is possible to write into the registers directly.

Register	Description		Default	Bus Terminal Controller
0	Byte offset PLC variables %I.		128	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
1	Length of the PLC variables %I...		16	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
2	Byte offset PLC variables %Q...		128	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
3	Length of the PLC variables %Q...		16	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
4-11	Bus Terminal assignment			BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
4.0-4.1	Assignment of the first Bus Terminal		00_{bin}	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
	00_{bin}	Local process image		
	10_{bin}	Fieldbus process image compact		
	11_{bin}	Fieldbus process image complex		
4.2-4.3	Assignment of the second Bus Terminal		00_{bin}	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
m.n-1..m.n	Assignment of the nth Bus Terminal		00_{bin}	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
14.2-14.3	Execution of the K-Bus update		00_{bin}	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
	00_{bin}	before and after the PLC		
	01_{bin}	before the PLC		
	10_{bin}	after the PLC		
12	PLC cycle time		5 ms 20 ms for the BC9000)	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
13	Background Time		2 ms 10 ms for the BC9000)	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
15	Size of the NOVRAM (remanent data)		64 kB	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
17.0-17.7	Terminal bus autorefresh: Cycle in [ms]		0x0000 Disable (0x005F for the BC9000)	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
17.8-17.15	Terminal bus autorefresh: max. retries			

Register	Description		Default	Bus Terminal Controller
18	Size of the NOVRAM Persistent data R18<R15		0	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
19	Byte offset digital inputs %I...		0	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
20	Byte offset digital outputs %Q...		0	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
27	Byte offset KL6xx1 bus terminal emula-tion %Q...		500	BC8x00
28	Byte offset KL6xx1 bus terminal emula-tion %I...		500	BC8x00
31	Baud Rate		2	BC8x00
	0	38400Bd		
	1	19200Bd		
	2	9600Bd		
	3	57600Bd		
	4	1200Bd		
	5	2400Bd		
	6	4800Bd		
32	Mode		2	BC8x00
	0	7 data bits, even parity		
	1	7 data bits, odd parity		
	2	8 data bits, no parity		
	3	8 data bits, even parity		
	4	8 data bits, odd parity		
33	Number of stop bits		1	BC8x00
	0	one stop bit		
	1	two stop bits		

Register Settings, Table 100

The registers can be set through a dialog using the KS2000 configuration software, or it is possible to write into the registers directly.

Register	Description		Default	Bus Terminal Controller
0-1	IP address		0xAC, 0x10, 0x11, 0x00	BC9000, BK9000
2-3	IP mask		0xFF, 0xFF, 0x00, 0x00	BC9000, BK9000
4-13	Device name		BC9000	BC9000, BK9000
14	Watchdog AMS/ADS		1000 ms	BC9000, BK9000
15.0	0_{bin}	Enable ModbusTCP	0_{bin}	BC9000, BK9000
	1_{bin}	Disable ModbusTCP		
15.1	0_{bin}	Enable AMS/ADS	0_{bin}	
	1_{bin}	Disable AMS/ADS		
16.8	0_{bin}	Ethernet mode half duplex	1_{bin}	BC9000, BK9000
	1_{bin}	Ethernet mode full duplex		
16.12	0_{bin}	Auto baud rate off	1_{bin}	
	1_{bin}	Auto baud rate on		
16.13	0_{bin}	10 MBaud	1_{bin}	
	1_{bin}	100 MBaud		
17-18	Default gateway		0x00, 0x00, 0x00, 0x00	BC9000, BK9000
24	Watchdog ModbusTCP		1000 ms	BC9000, BK9000
25.0	Entry of a fieldbus error (connection to switch interrupted) in the flags area diagnosis		0_{bin}	BC9000
	0	Error was not entered		
	1	Error was entered		
26	ModbusTCP port no.		502	BC9000, BK9000
27	ADS connection duration		10 s	BC9000, BK9000
28	Modbus connection duration		10 s	BC9000, BK9000
29.0	ModbusTCP mode		0_{bin}	BC9000, BK9000
	1_{bin}	Fast ModbusTCP		

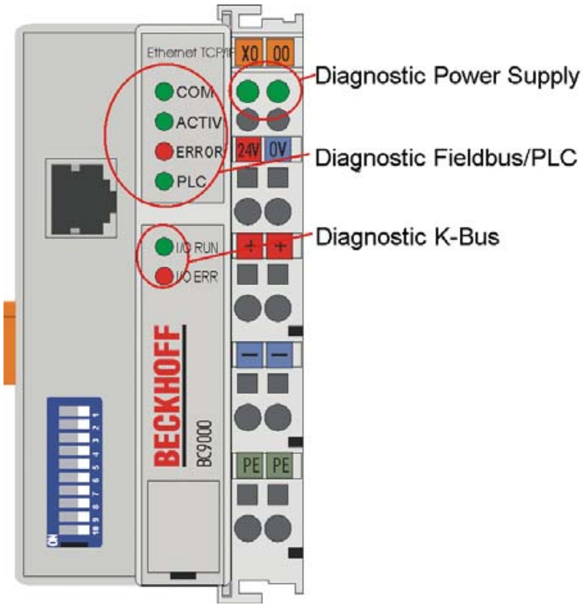
8. Error Handling and Diagnosis

Diagnostic LEDs

After switching on, the Bus Coupler immediately checks the connected configuration. Error-free start-up is indicated when the red *I/O ERR* LED goes out. If the *I/O ERR* LED blinks, an error in the area of the terminals is indicated. The error code can be determined from the frequency and number of blinks. This permits rapid rectification of the error.

The Bus Coupler has two groups of LEDs for the display of status. The upper group with four LEDs indicates the status of the respective fieldbus. The significance of the fieldbus status LEDs is explained in the appropriate sections of this manual. It corresponds to the usual fieldbus display.

On the upper right hand side of the Bus Couplers are two more green LEDs that indicate the supply voltage. The left hand LED indicates the presence of the 24 V supply for the Bus Coupler. The right hand LED indicates the presence of the supply to the power contacts.



LEDs for power supply diagnosis

LED	Meaning
Left LED off	Bus coupler has no power
Right LED off	No 24V DC power connected to the power contacts

LEDs for fieldbus diagnosis

LED	On	Off
LINK	Physical connection present	No physical connection present
ACT	Flashing: Bus traffic present	No bus traffic (bus idle)
ERROR	The LED flashes slowly if DHCP or BootP is active but the Bus Coupler has not yet received an IP address The LED flashes rapidly (5 times, only when switching on); the Bus Coupler is addressed with ARP. The settings on the DIP SWITCH are not valid.	No error.
PLC	PLC program is in RUN mode The LED flashes while the boot project is being saved.	PLC program is in stop mode

LEDs for K-Bus diagnosis

Error code	Error code argument	Description	Remedy
Persistent, con-tinuous blinking		EMC problems	- Check power supply for overvoltage or undervoltage peaks - Implement EMC measures - If a K-Bus error is present, it can be localised by a restart of the coupler (by switching it off and then on again)
1 pulse	0	EEPROM checksum error	Set manufacturer's setting with the KS2000 configuration software
	1	Code buffer overflow	Insert fewer Bus Terminals. The programmed configuration has too many entries in the table
	2	Unknown data type	Software update required for the Bus Coupler
2 pulses	0	Programmed configuration has an incorrect table entry	Check programmed configuration for correctness
	n (n > 0)	Table comparison (Bus Terminal n)	Incorrect table entry
3 pulses	0	K-Bus command error	- No Bus Terminal inserted - One of the Bus Terminals is defective; halve the number of Bus Terminals attached and check whether the error is still present with the remaining Bus Terminals. Repeat until the defective Bus Terminal is located.
4 pulses	0	K-Bus data error, break behind the Bus Coupler	Check whether the n+1 Bus Terminal is correctly connected; replace if necessary.
	n	Break behind Bus Terminal n	Check whether the Bus End Terminal 9010 is connected.
5 pulses	n	K-Bus error in register communication with Bus Terminal n	Exchange the nth bus terminal
6 pulses	0	Error at initialisation	Exchange Bus Coupler
	1	Internal data error	Perform a hardware reset on the Bus Coupler (switch off and on again)
	2	DIP switch changed after a software reset	Perform a hardware reset on the Bus Coupler (switch off and on again)
	4	DIP switch incorrect for BootP	Set 1-8 to *on* or *off*, see BootP
	8	Internal data error	Perform a hardware reset on the Bus Coupler (switch off and on again)
	16	Error in IP socket	Perform a hardware reset on the Bus Coupler (switch off and on again)
7 pulses	0	Note: Cycle time was exceeded (see Table 1, Register 17)	Warning: the set cycle time was exceeded. This indication (flashing LEDs) can only be cleared by booting the Bus Coupler again. Remedy: increase the cycle time
9 pulses	0	Checksum error in Flash program	Transmit program to the BC again
	1	Incorrect or faulty library implemented	Remove the faulty library

Error code	Error code argument	Description	Remedy
10 pulses	n	Bus Terminal n is not consistent with the configuration that existed when the boot project was created	Check the nth Bus Terminal. The boot project must be deleted if the insertion of an nth bus terminal is intentional
14 pulses	n	nth Bus Terminal has the wrong format	Start the Bus Coupler again, and if the error occurs again then exchange the Bus Terminal
15 pulses	n	Number of Bus Terminals is no longer correct	Start the Bus Coupler again. If the error occurs again, restore the manufacturers setting using the KS2000 configuration software
16 pulses	n	Length of the K-Bus data is no longer correct	Start the Bus Coupler again. If the error occurs again, restore the manufacturers setting using the KS2000 configuration software

General Errors

No data exchange after replacing a bus coupler

You have exchanged the Ethernet Bus Coupler and set the same TCP/IP number, but data is not exchanged.

Every Ethernet device has its own, unique MAC-ID. This number is saved when connecting to an Ethernet node, and stored in a table. This table contains the correspondences between the MAC-ID and the TCP/IP number. You must delete this table. Do this in a DOS window, by entering the command "arp -d" and the TCP/IP number of the Bus Coupler.
Example: >arp -d 172.16.17.203<
If the DHCP protocol or the BootP protocol is active it is necessary to set the MAC-ID of the new coupler in the DHCP server or BootP server after changing the Bus Coupler.

Communication errors when online

After a period in the online state (logged in via Ethernet/AMS) the message *Communication error - logging out* always occurs.

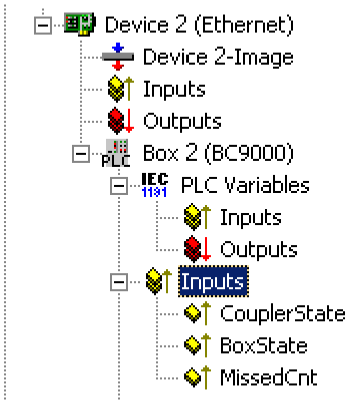
The data traffic through the Ethernet interface is jamming. Remedy:

- Reduce the level of data communication.
- Stop the cyclical data traffic, or lengthen the task time.
- Reduce the number of windows open in the online display.
- Log in via the serial interface.

ADS Diagnostic

Status inputs

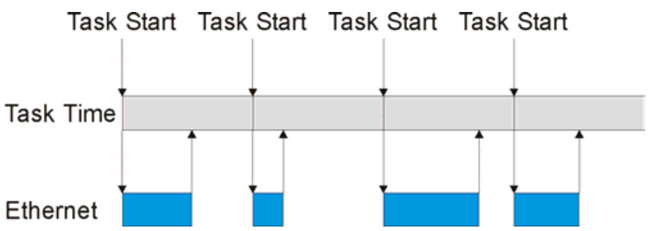
It is possible to monitor the BK/BC9000 Bus Coupler's communication in the system manager. Each Bus Coupler has status inputs that can be found in the hardware tree.



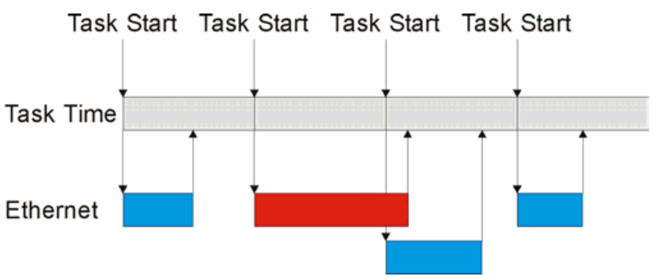
Coupler state

Value	Meaning
0x0000	No error
0x0001	Bus Terminal error, there is a K-Bus error
0x0002	Configuration error; the parameterised configuration does not match the actual configuration

MissedCnt
Ideal configuration:
The task time is always longer than the Ethernet transmission time. An Ethernet telegram is transmitted at the beginning of the task, and it returns it to the PC again after a period of time, t_{Eth}. If the time t_{Eth} is always smaller than the task time that has been set, the value in the *MissedCnt* counter remains constant.



If, after the task time has elapsed, an Ethernet telegram has still not arrived at the PC, only reaching it after the next task has started, then TwinCAT will continue to work with the old input data. In addition, the *MissedCnt* counter is incremented.



This can have the following causes:

- The chosen task time is too short. Set
  - 100 ms or more for TCP
  - 20 ms or more for UDP
- Too many collisions in the network: use switches instead of hubs!
- The bus loading is too high: change to 100 MBaud!
- You are logged in to the BC9000: this consumes additional processing capacity in the Bus Terminal Controller, which lengthens the response time.

The two following diagnostic inputs have different meanings, depending on the transmission type.

TCP/IP diagnosis

BoxState

Value	Meaning
0x0000	No error
0x0001	No current inputs

MissedCnt

Value	Meaning
0xyyyy	Number of telegrams that have not returned in time for the start of the task. This value should remain almost constant. If the value keeps rising, the task time should be lengthened.

UDP/IP diagnosis

BoxState

Value	Meaning
0x0000	No error
0x0001	No current inputs
0x0002	Outputs are written as zero
0xxxzz	xx - warning level. The value here is incremented by one each time the data is not received in time. When data is exchanged correctly, the value is decremented by one. When the maximum warning level (default value 100) is reached, zero is written to the output data, and it is only enabled again for the normal process image when the warning level has reached zero.

MissedCnt

Value	Meaning
0xyyyy	Number of telegrams that have not returned in time for the start of the task. This value should remain almost constant. If the value keeps rising, the task time should be lengthened.

ModbusTCP Diagnostic

See Modbus diagnostic function

See ModbusTCP error answers

9. Appendix

General Operating Conditions

The following operating conditions must be observed if the components are to function without error.

Environmental conditions

The components may not be used without additional protection in the following locations:

- in difficult environments, such as where there are corrosive vapours or gases, or high dust levels
- in the presence of high levels of ionising radiation.

Condition	Permissible range
Operating temperature	0°C ... +55°C
Vibration resistance	conforms to IEC 68-2-6
Shock resistance	conforms to IEC 68-2-27
Installation position	variable
EMC immunity	conforms to EN50082 (ESD, Burst)
Emission	conforms to EN50081

Transport and storage conditions

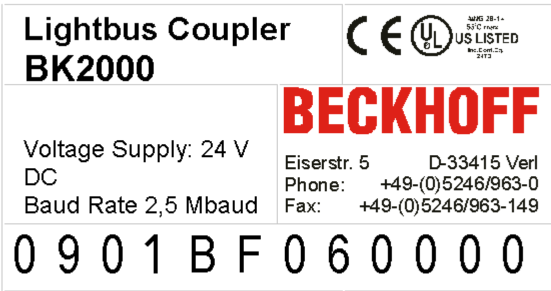
Condition	Permissible range
Storage temperature	-25 °C ... +85°C
Relative humidity	95 %, no condensation
Free fall	up to 1 m in the original packaging

Protection classes and types

Condition	Permissible range
Protection class in accordance with IEC 536 (VDE 0106, Part 1)	A protective conductor connection to the mounting rail is necessary!
Protection class according to IEC 529	IP20 (protection against contact with a standard test finger)
Protection against foreign objects	Less than 12 mm in diameter
Protection against water	No protection

Component identification

Every supplied component includes an adhesive label providing information about the product's approvals. For example, on the BK2000:



The following information is printed on the label:

Printed item	In this particular example:
Precise product identification	Lightbus Coupler BK2000
Supply voltage	24 V_{DC}
Data transfer rate	2.5 MBaud
manufacturer	Elektro Beckhoff GmbH, Industry Electronics Division
CE mark	Conformity mark
UL mark	Mark for UL approval. UL stands for the Underwriters Laboratories Inc., the leading certification organisation for North America, based in the USA. C = Canada, US = USA, LISTED 22ZA (the test results can be inspected under this entry)
Production identification	From left to right, this sequence of characters indicates the production week (2 characters), the production year (2 characters), the software version (2 characters) and hardware version (2 characters), along with any special indications (4 characters). This case therefore is a BK2000 - produced in the 9th calendar week - of the year 2001 - containing the BF firmware version - and using the 6th hardware version - with no special indications

Approvals

UL E172151

Conformity mark

CE

Protection class

IP20 in accordance with EN60529

Test Standards for Device Testing

EMC

EN 50082-2

EN 50081-2

Vibration resistance

EN 60 68-2-2 Vibration Testing

EN 60 68-2-27 Shock Testing

Bibliography

TCP/IP

TCP/IP (English)
Illustrated, Volume1 The Protocols
by W. Richard Stevens
Publisher: ADDISON-WESLEY Longmann Verlag

TCP/IP (German)
Aufbau und Betrieb eines TCP/IP Netzes (Structure and Operation of a TCP/IP Network)
by Kevin Washburn and Jim Evans
Publisher: ADDISON-WESLEY Longmann Verlag

Modbus/TCP

http://www.modicon.com/
http://www.modbus.org

TwinCAT

TwinCAT information documentation
http://tcinfosys.beckhoff.com

List of Abbreviations

ADS
Automation Device Specification.

IP (20)
Bus Terminal protection class

IPC
Industrial PC

I/O
Inputs and outputs

K-Bus
Terminal bus

KS2000
Configuration Software

PE
The PE power contact can be used as a protective earth.

TwinCAT
The Windows Control and Automation Technology

Support and Service

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

BECKHOFF Support

Support offers you comprehensive technical assistance, helping you no only with the application of individual BECKHOFF products, but also with other, wide-ranging services:

- world-wide support
- design, programming and commissioning of complex automation systems
- and extensive training program for BECKHOFF system components

Hotline:	+49(0)5246/963-157
Fax:	+49(0)5246/963-199
e-mail:	support@beckhoff.com

BECKHOFF Service

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

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

Hotline:	+49(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

You will find further support and service addresses on our Internet pages under http://www.beckhoff.com. You will also find documentation for BECKHOFF components there.