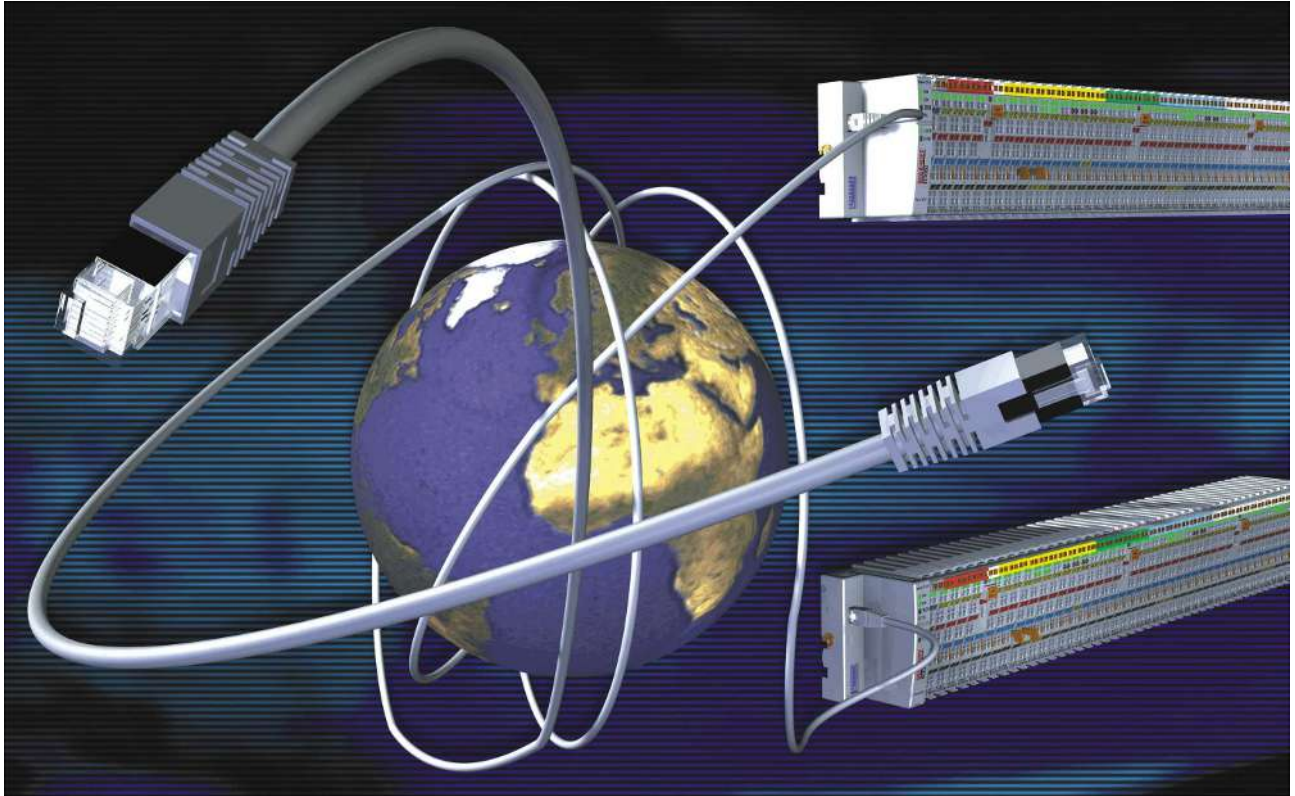




Documentation

BC9000 and BC9100

Bus Terminal Controller for Ethernet

Version: 4.1.0
Date: 2020-01-08

BECKHOFF

Table of contents

1 Foreword	5
1.1 Notes on the documentation.....	5
1.2 Safety instructions	6
1.3 Documentation issue status	7
2 Product overview.....	9
2.1 Principle of the Bus Terminal.....	9
2.2 The Beckhoff Bus Terminal system	9
2.3 Technical data	11
2.4 Technical Data for the PLC	12
2.5 Ethernet	12
3 Fitting and wiring.....	14
3.1 Mechanical installation	14
3.1.1 Dimensions	14
3.1.2 Installation on mounting rails	15
3.2 Wiring.....	16
3.2.1 Power supply, potential groups.....	16
3.2.2 Ethernet topologies.....	18
3.2.3 Ethernet connection.....	20
3.2.4 Ethernet cable.....	21
4 Parameterization and Commissioning	23
4.1 Start-up behavior of the Bus Terminal Controller	23
4.2 Parameterization of the Bus Coupler using DIP switches	23
4.3 Network classes	24
4.4 IP address	25
4.4.1 IP address.....	25
4.4.2 Configuration with KS2000	25
4.4.3 Setting the IP address using the ARP table.....	26
4.4.4 Setting the IP Address Using the Beckhoff BootP Server	27
4.4.5 Setting the address using a DHCP server	28
4.4.6 Addressing by name	29
4.4.7 Subnet mask.....	30
4.4.8 Testing the IP address.....	30
4.4.9 Reading the MAC-ID.....	31
4.4.10 Security settings	31
5 Configuration	32
5.1 Configuration using the System Manager	32
5.2 The IP Address tab.....	33
5.3 PLC tab.....	34
5.4 Configuration for ModbusTCP	36
6 Programming	37
6.1 TwinCAT PLC	37
6.2 PLC Cycle Time.....	37
6.3 PLC Variables.....	38

6.4	Mapping the Bus Terminals	39
6.5	Local process image	39
6.6	Allocated flag area %MB.	40
6.6.1	Remanent and persistent data	40
6.6.2	Diagnostics	41
6.6.3	Cycle Time Measuring	41
6.7	Fieldbus process image	41
6.7.1	Fieldbus process image	41
6.7.2	ModbusTCP Process Image	41
6.7.3	ADS Process Image	42
6.8	Program transfer	42
6.8.1	Program transfer via the serial interface	42
6.8.2	Program Transfer via Ethernet	43
6.9	Libraries	46
6.9.1	Libraries	46
6.9.2	TcUtilitiesBC9000	47
6.9.3	TclpUtilitiesBC	50
7	Fieldbus system	52
7.1	Ethernet	52
7.2	Topology	54
7.3	Reaction times	54
7.4	ADS-Communication	55
7.4.1	ADS-Communication	55
7.4.2	ADS protocol	56
7.4.3	ADS services	57
7.5	ModbusTCP	65
7.5.1	ModbusTCP Protocol	65
7.5.2	Modbus TCP interface	66
7.5.3	ModbusTCP slave error answer (BK9000, BX/BC9xx0, IP/ILxxxx-B/C900, EK9000)	68
7.5.4	ModbusTCP functions	68
7.5.5	Examples for ModbusTCP	75
7.6	Description of parameters	76
7.6.1	Properties of the Bus Terminal Controller	76
8	Error handling and diagnosis	79
8.1	Diagnostic LEDs	79
8.2	General errors	82
8.3	ADS diagnostics	82
8.4	ModbusTCP diagnostic	84
9	Appendix	85
9.1	General operating conditions	85
9.2	Approvals	86
9.3	Test standards for device testing	87
9.4	Bibliography	87
9.5	List of Abbreviations	87
9.6	Support and Service	89

1 Foreword

1.1 Notes on the documentation

Intended audience

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 documentation and the following notes and explanations are followed when installing and commissioning these components.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

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

Patent Pending

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

In this documentation the following instructions are used.

These instructions must be read carefully and followed without fail!

DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.



Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Modifications
4.1	• Update download link
4.0	• Migration
3.12	• HTML documentation for firmware versions from BB (BC9000) and B1 (BC9100)
3.11	• HTML documentation for firmware versions from BB (BC9000) and B1 (BC9100)
3.10	• HTML documentation; valid as from firmware BA
3.9	• HTML documentation; valid as from firmware B9
3.8	• HTML documentation; valid as from firmware B8

Firmware BC9000/BC9100

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 the 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.de.

Firmware	Description
BB (BC9000) B1 (BC9100)	• Restructuring of the internal Ethernet communication • Adjustment of the cycle tick counter to 1 ms per tick
BA	• Variable size increased from 6 kbyte to 16 kbyte • Data storage in <u>flash memory</u> [► 60] • SNTP/Time Protocol implemented • Access via name resolution • Optimization: 25% performance increase through code optimization
B9	Error in the B8 version related to mapping the KL60xx Bus Terminals corrected
B8	Internal software reset modified (this does not have any effects that concern the user)
B7	• Settings such as the IP address are retained during a firmware update • ADS indication function block implemented • ADS "State" supported • DHCP for "Nortel Switch" modified • Cycle time monitoring • Fast ModbusTCP implemented
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
B5	• Addressing via ARP and BootP possible • Access to the flags area via ModbusTCP
B3	Implementation of ModbusTCP
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
B1	Released version
B0	Preliminary version

2 Product overview

2.1 Principle of the Bus Terminal

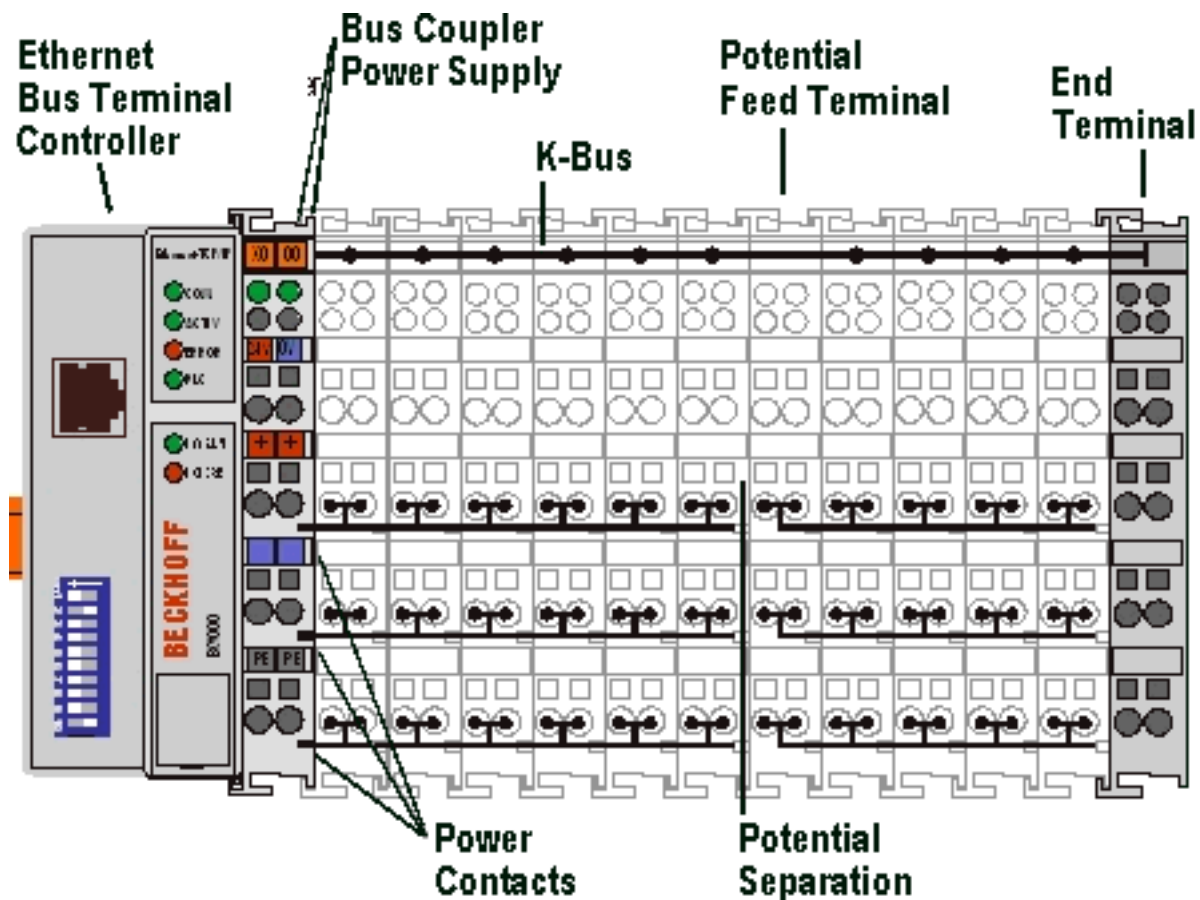


Fig. 1: Principle of the Bus Terminal

2.2 The Beckhoff Bus Terminal system

Up to 256 Bus Terminals, with 1 to 16 I/O channels per 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. Up to 255 Bus Terminals can be connected via the K-bus extension. For each technical signal form, terminals are available with one, two, four or eight I/O channels, which can be mixed as required. All the terminal types have the same mechanical construction, so that difficulties of planning and design are minimized. 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.

Mounting on standardized mounting rails

The installation is standardized thanks to the simple and space-saving mounting on a standardized mounting rail (EN 60715, 35 mm) and the direct wiring of actuators and sensors, without cross connections between the terminals. The consistent labelling scheme also contributes.

The small physical size and the great flexibility of the Bus Terminal system allow 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 Bus Terminals can be used in a terminal block, with optional K-bus extension for up to 256 Bus Terminals. 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.

2.3 Technical data

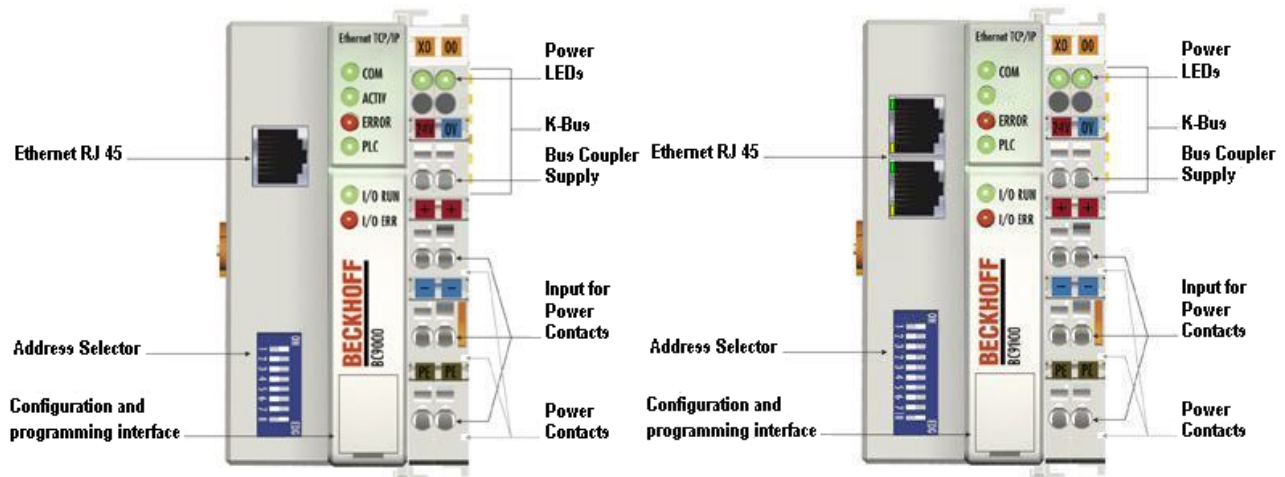


Fig. 2: BC9000 and BC9100

System data	Ethernet (BC9000)	Ethernet (BC9100)
Number of I/O modules	only limited by the IP address space	
Number of I/O points	depending on controller	
Data transfer medium	4 x 2 twisted-pair copper cables; category 3 (10 Mbit/s), category 5 (100 Mbit/s)	
Distance between modules	100 m (distributor hub to BC9000)	Line: max. 100 m between two BC9100
Data transfer rate	10 / 100 Mbit/s	
Topology	star wiring	star wiring, up to 20 BC9100 in one line

Technical data	BC9000	BC9100
Number of Bus Terminals	64	
Digital peripheral signals	256 inputs/outputs	
Analog peripheral signals	128 inputs/outputs	
Protocols	Beckhoff ADS, ModbusTCP	
Configuration possibility	via the KS2000 configuration software or the controller (TwinCAT)	
Maximum number of bytes	512 bytes I and 512 bytes O	
Bus connection	1 x RJ 45	2 x RJ45
Internal switch	-	3-port switch (two external connections)
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	max. 24 V _{DC}	
Power contact current load	10 A max.	
Dielectric strength	500 V (power contact/supply voltage/fieldbus)	
Weight approx.	170 g	
Operating temperature	-25°C ... +60°C	
Storage temperature	-40°C ... +85°C	
Relative humidity	95% no condensation	
Vibration/shock resistance	conforms to IEC 68-2-6 / IEC 68-2-27	
EMC immunity/emission	conforms to EN 50082 (ESD, burst) / EN 50081	
Mounting position	variable	
Protection class	IP20	

2.4 Technical Data for the PLC

PLC data	BC9000	BC9100
Programmability	via programming interface (TwinCAT) 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)	Approx. 1.2 ms for 1000 IL commands (without I/O cycle)
Programming languages	IEC 6-1131-3 (IL, LD, FBD, ST, SFC)	

2.5 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 fiber (10BaseF) or twisted pairs (10BaseT) with screen (STP) or without screen (UTP). Using Ethernet, different topologies can be built such as ring, line or star.

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 numbers of different networks. TCP/IP was developed to provide standardised, reliable data exchange between any numbers of different networks. 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 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 Fitting and wiring

3.1 Mechanical installation

3.1.1 Dimensions

The system of the Beckhoff Bus Terminals is characterized 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.

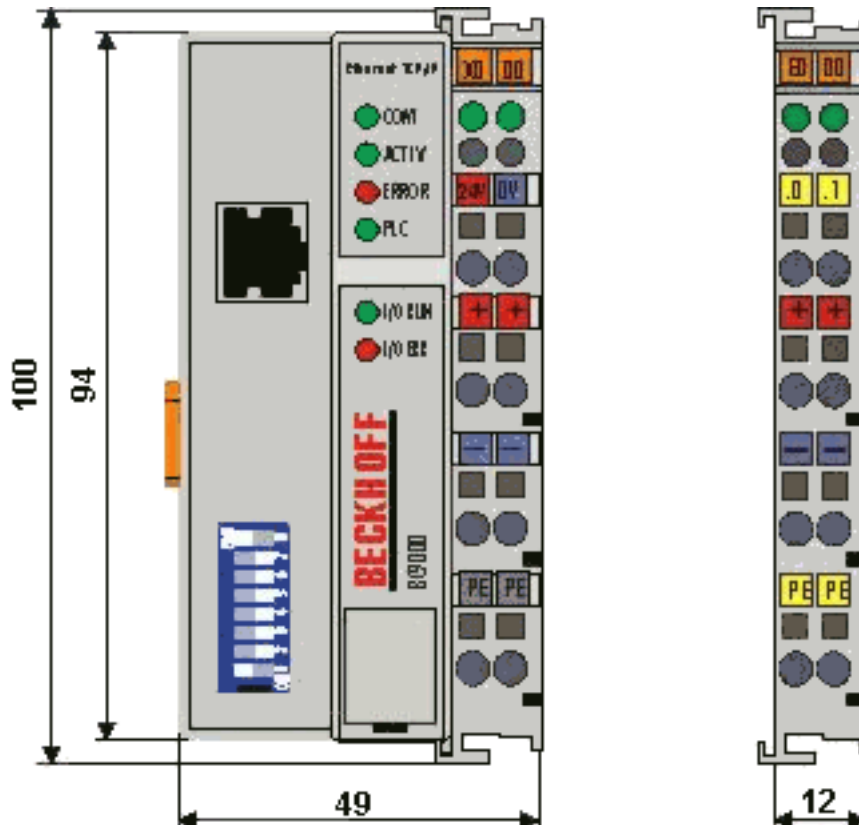


Fig. 3: Dimensions

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 mm or 24 mm wide. The front wiring increases the total height of 68 mm by about 5 mm to 10 mm, depending on the wire thickness.

3.1.2 Installation on mounting rails

WARNING

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminals system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals.

Mounting

The Bus Couplers and Bus Terminals are attached to commercially available 35 mm mounting rails (DIN rail according to EN 60715) 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 slot and key 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.

Fixing of mounting rails

The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

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 terminal to be disassembled, 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 realized by joining the components:

- The six spring contacts of the K-Bus/E-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.

Power contacts

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 and EL91xx, EL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

PE power contact

The power contact labelled 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.

NOTE**Risk of damage to the device**

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.

⚠ WARNING**Risk of electric shock!**

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 safely and permanently.

**Shielding**

Analog sensors and actuators should always be connected with shielded, pair-wise twisted cables.

3.2 Wiring

3.2.1 Power supply, potential groups

Power supply for the Bus Coupler

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/E-Bus, the Bus Terminals. The power supply for the Bus Coupler electronics and that of the K-Bus/E-Bus are electrically separated from the potential of the field level.

Power supply for the 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 cannot 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 (not for BK1250, EK1x00)

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/E-Bus electronics in the Bus Coupler and the K-Bus/E-Bus itself in an electrically isolated manner. The supply voltage is also used to generate the operating voltage for the fieldbus.

Note: All Bus Terminals are electrically isolated from the K-Bus/E-bus. The K-Bus/E-bus is therefore completely electrically isolated.

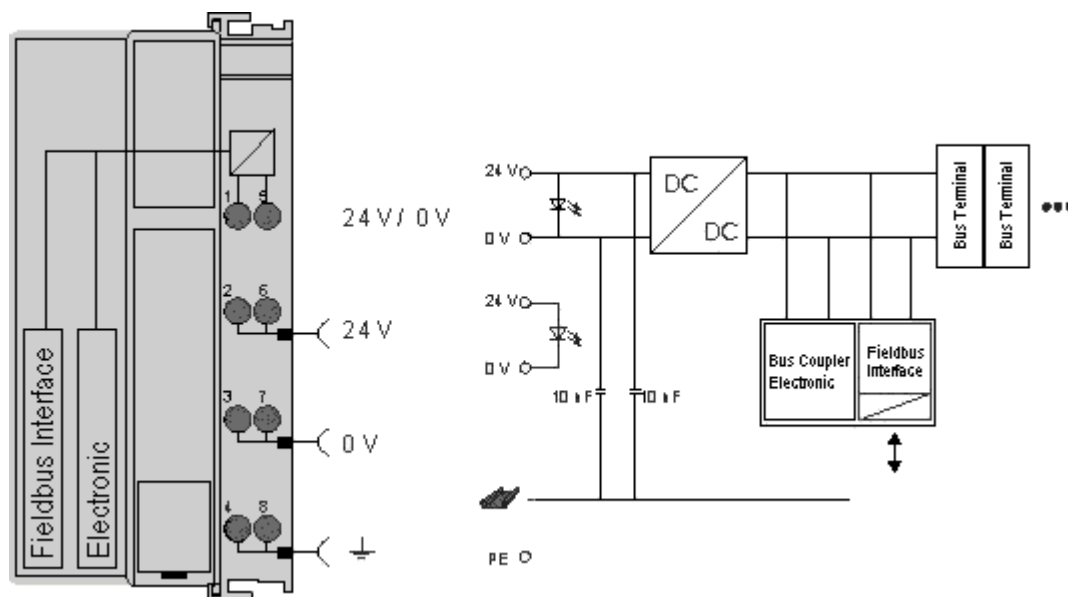


Fig. 4: Electrical isolation

3.2.2 Ethernet topologies

BK9000, BK9050, BC9000, BC9020, BC9050

These Bus Couplers and Bus Terminal controllers have a single Ethernet connection. This can be connected directly to an external switch. This makes it possible to construct the typical Ethernet star topology.

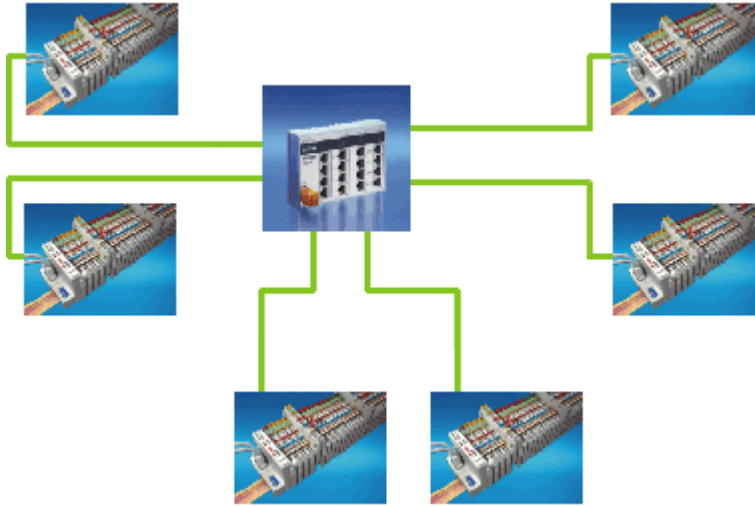


Fig. 5: Ethernet layout in star topology

BK9100, BC9100, BC9120, BC9191

These Bus Couplers and Bus Terminal controllers have an internal triple switch with one internal and two external ports. The internal switch enables the simple construction of a linear topology. A maximum of 20 BK9100/BC91x0/BC9191 can be connected in series in a physical line. However the distance between two Ethernet devices may not exceed 100 m. The maximum overall line length is therefore 2 km. No further switches may be included in this line.

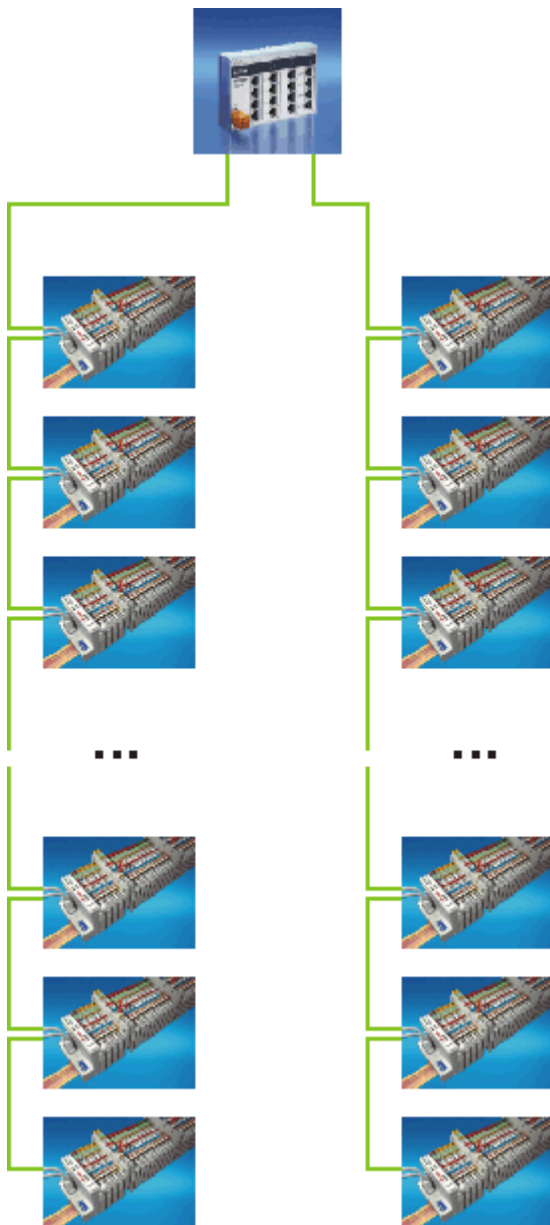


Fig. 6: Ethernet layout in linear topology

Of course, the construction of a classic star topology is also possible with these Bus Couplers and Bus Terminal controllers.

3.2.3 Ethernet connection

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

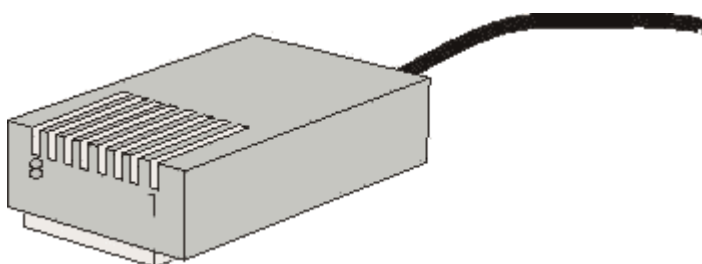


Fig. 7: RJ45 connector (Western plug)

Cabling

Connection via hub or switch

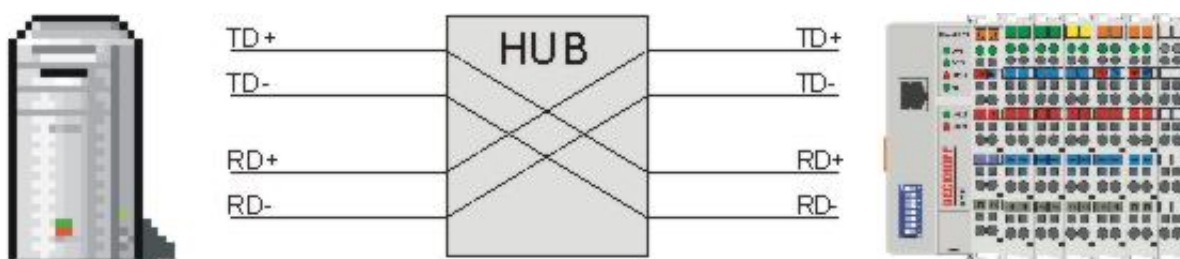


Fig. 8: Ethernet connection via hub or switch

Connect the PC's network card to the hub/switch 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

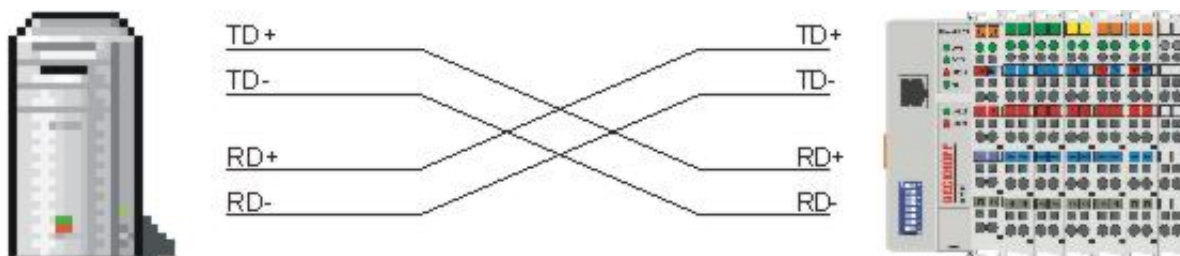


Fig. 9: Direct Ethernet connection (crossover cable)

Use a crossover Ethernet cable to connect the PC directly with the Bus Terminal Controller.

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

3.2.4 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 (Cheaper net) 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 fiber 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 meters.

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 an overall shield 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 Parameterization and Commissioning

4.1 Start-up behavior of the Bus Terminal Controller

When the Bus Terminal Controller is switched on it checks its state, configures the K-bus, creates a configuration list based on the connected Bus Terminals and starts its local PLC.

The I/O LEDs flash when the Bus Terminal Controller starts up. If the system is in an error-free state, the I/O LEDs should stop flashing after approx. 2-3 seconds. In the event of a fault the error type determines which LED flashes (see chapter *Diagnostic LEDs*).

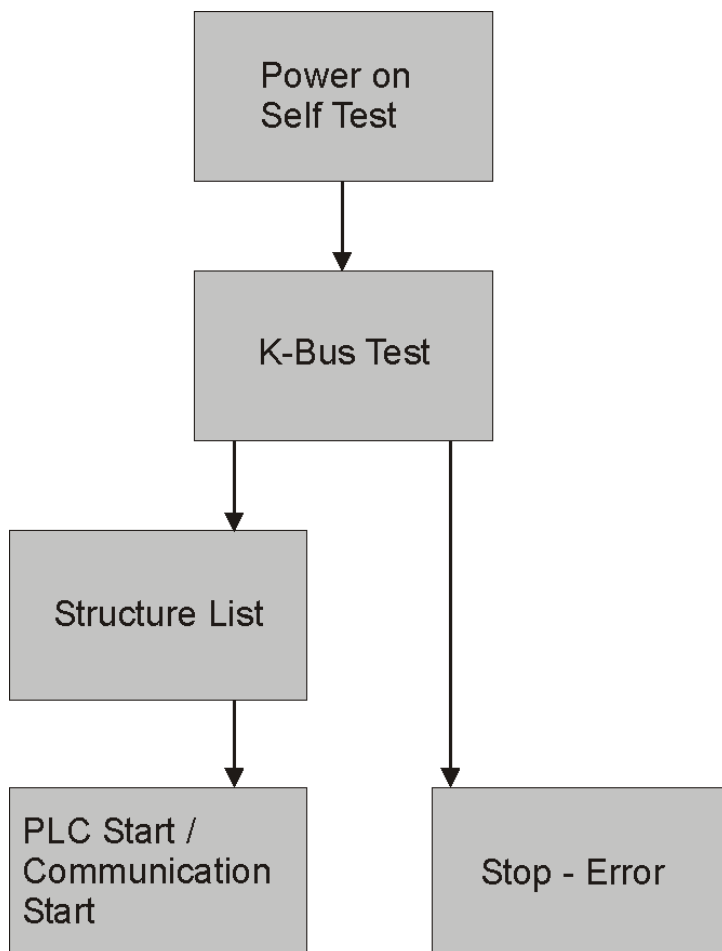


Fig. 10: Start-up behavior of the Bus Terminal Controller

4.2 Parameterization of the Bus Coupler using DIP switches

The following parameterizations can be carried out without using configuration software, with the aid of the DIP switches and the end terminal (KL9010).

This paramétrisation mode is only active if only one end terminal (KL9010) is inserted. Otherwise, the normal settings apply.

Restoring the manufacturer's settings

- Switch the Bus Coupler off and then append 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 ERR* LEDs flash alternately.

- You can then switch the Bus Coupler off, connect the Bus Terminals, and continue as usual.

Deleting the boot project (BC9000 only)

- Switch the Bus Coupler off and then append 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 ERR* LEDs flash alternately.
- You can then switch the Bus Coupler off, connect the Bus Terminals, and continue as usual.

Setting the Ethernet parameters

- Switch the Bus Coupler off and then append 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 ERR* LEDs light steadily.
- Make the desired setting in accordance with the following table.

DIP switch	Parameter	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.

4.3 Network classes

Three different network classes are distinguished. They determine how many address bits are reserved for the network ID and how many for the node number (e.g. for PCs or Bus Couplers). The difference is located in the first three bits of the IP address.

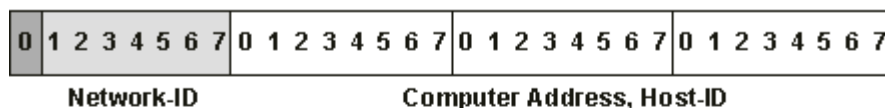
Network class	Number of bits for the network ID	Enables no. of networks	No. of bits for the node address	Enables no. of nodes per network
A	7	126	24	16 777 214
B	14	16 382	16	65 536
C	21	2 097 150	8	254



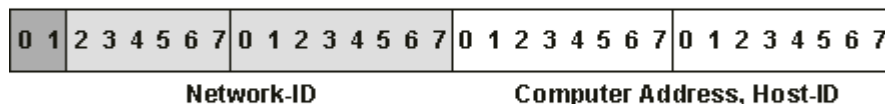
Unique IP address

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

Network class A, Addresses: 1.xxx.xxx.xxx - 126.xxx.xxx.xxx



Network class B, Addresses: 128.0.xxx.xxx - 191.255.xxx.xxx



Network class C, Addresses: 192.0.0.xxx - 223.255.255.xxx

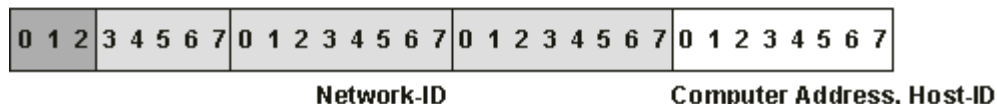


Fig. 11: Network classes

● Identical network class

i In a communication with another Ethernet devices, the IP address set must have the same network class. Sample: Your PC has the address 172.16.17.55, which means the coupler must have the address 172.16.xxx.xxx (xxx stands for a number between 0 and 255. 0 is usually used by the router/switch and should therefore remain reserved).

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

4.4 IP address

Usually the IP address is specified by the PROFINET controller and therefore does not need to be set in the bus coupler. The bus couplers are addressed via the PROFINET-Name which can be set using various procedures (see DIP switch settings).

In exceptional cases it makes sense to give the bus coupler an IP address in advance (without connecting PROFINET), e.g. to access the couplers via KS2000 Software and Ethernet.

4.4.1 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 [► 25]	KS2000 configuration software and KS2000 cable
ARP	Addressing via the ARP table [► 26]	PC with network
BootP	Addressing via BootP server [► 27]	BootP server
DHCP	Addressing via DHCP server	DHCP server

4.4.2 Configuration with KS2000

The KS2000 configuration software (from version 3.2.8) can be used to set the TCP/IP address in a dialog box, or it can be written directly into the registers. DIP switches 9 and 10 (for BK9050 DIP switches 1 and 2 in blue) should both be 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})

Sample

Fig. 12: BK9000, BK9100, BC9000, BC9100



Fig. 13: BK9050

Switch no.	1	2	3	4	5	6	7	8	9 (1)	10 (2)	
Valence	1	2	4	8	16	32	64	128	-	-	
In this example	ON	OFF	OFF	ON	OFF	OFF	ON	ON	OFF	OFF	
Value	1	0	0	8	0	0	64	128	-	-	Total=201

**Software reset**

A software reset of the BK9500 is required in order to save changes in the tables of a BK9000. A hardware reset (power on/off) is not sufficient!

4.4.3 Setting the IP address using the ARP table

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 [31]>" 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.

● Changing the IP address

i If the IP address is changed, all dynamic ARP entries must be deleted. Only one ping with a length of 123 bytes is permitted for the reconfiguration of the IP address (>ping -l "IP-Address<).

Sample

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

4.4.4 Setting the IP Address Using the Beckhoff 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 reports LED error code 6-4 (see diagnostics LEDs). 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 [► 23]).

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. The installation version of the Beckhoff TwinCAT CD can be found under the folder >Unsupported Utilities< or under https://download.beckhoff.com/download/software/TwinCAT/TwinCAT2/Unsupported_Utilities/TcBootP_Server/

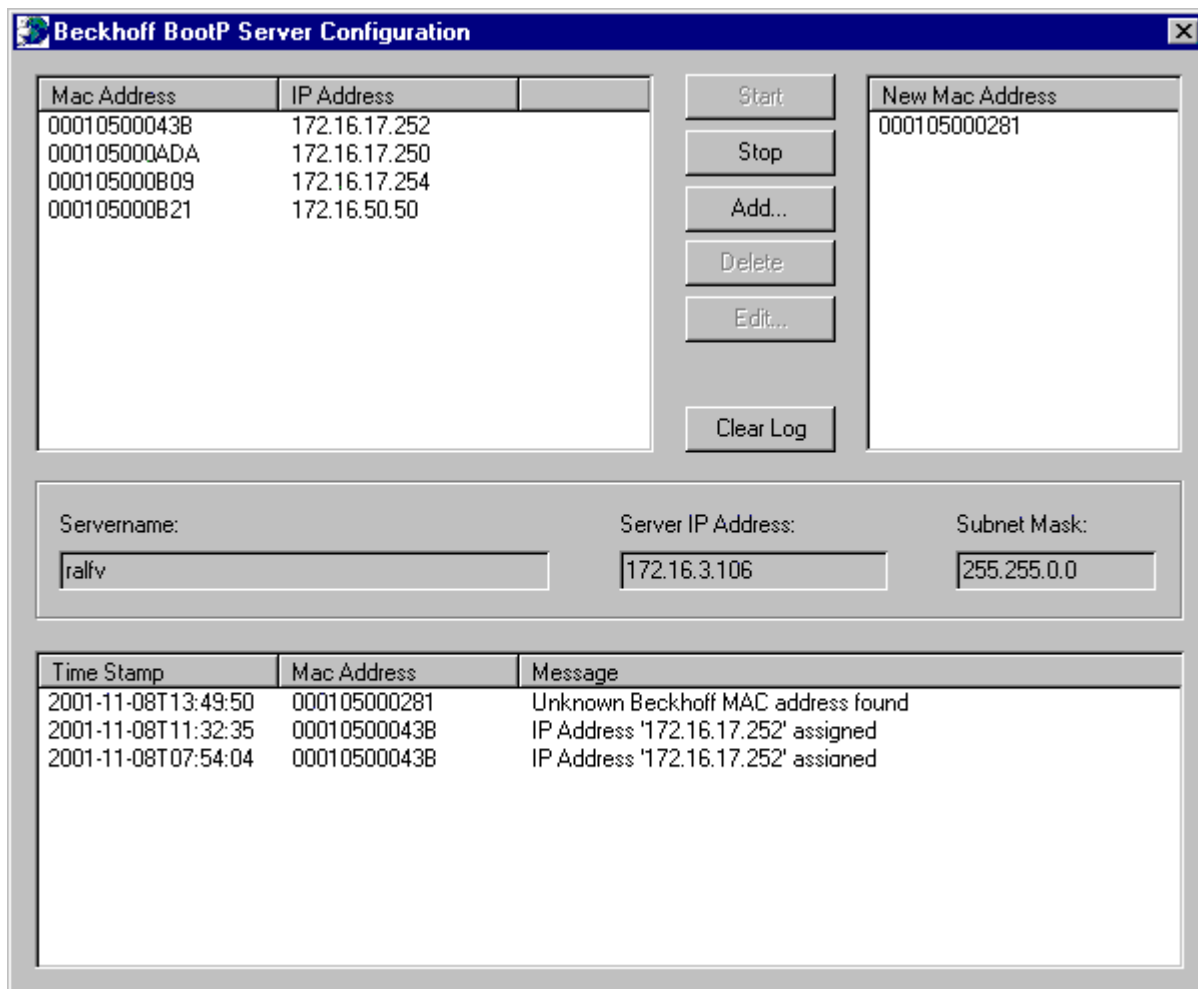


Fig. 14: Configuration of the Beckhoff BootP server

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 [► 31] 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).

4.4.5 Setting the address using a 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 assigned an IP address by the DHCP server.

The DHCP server must know the MAC ID of the Bus Coupler and should assign the same IP address to this MAC ID on each startup.

The *TCP/IP error* LED flashes while the address is being allocated.

4.4.6 Addressing by name

Name resolution requirements:

- Enabling DHCP: Set DIP switch 10 to ON (DHCP and DNS server available)
- Name setting via KS2000 on the BC/BK9000

The default setting for a BC9000 name is *BC9000-4*, for example. The number following the hyphen, in this case 4, originates from the address selection switches 1 to 8 of the BC9000. The name can also be set with the KS2000 configuration software. The maximum length is 18 characters.

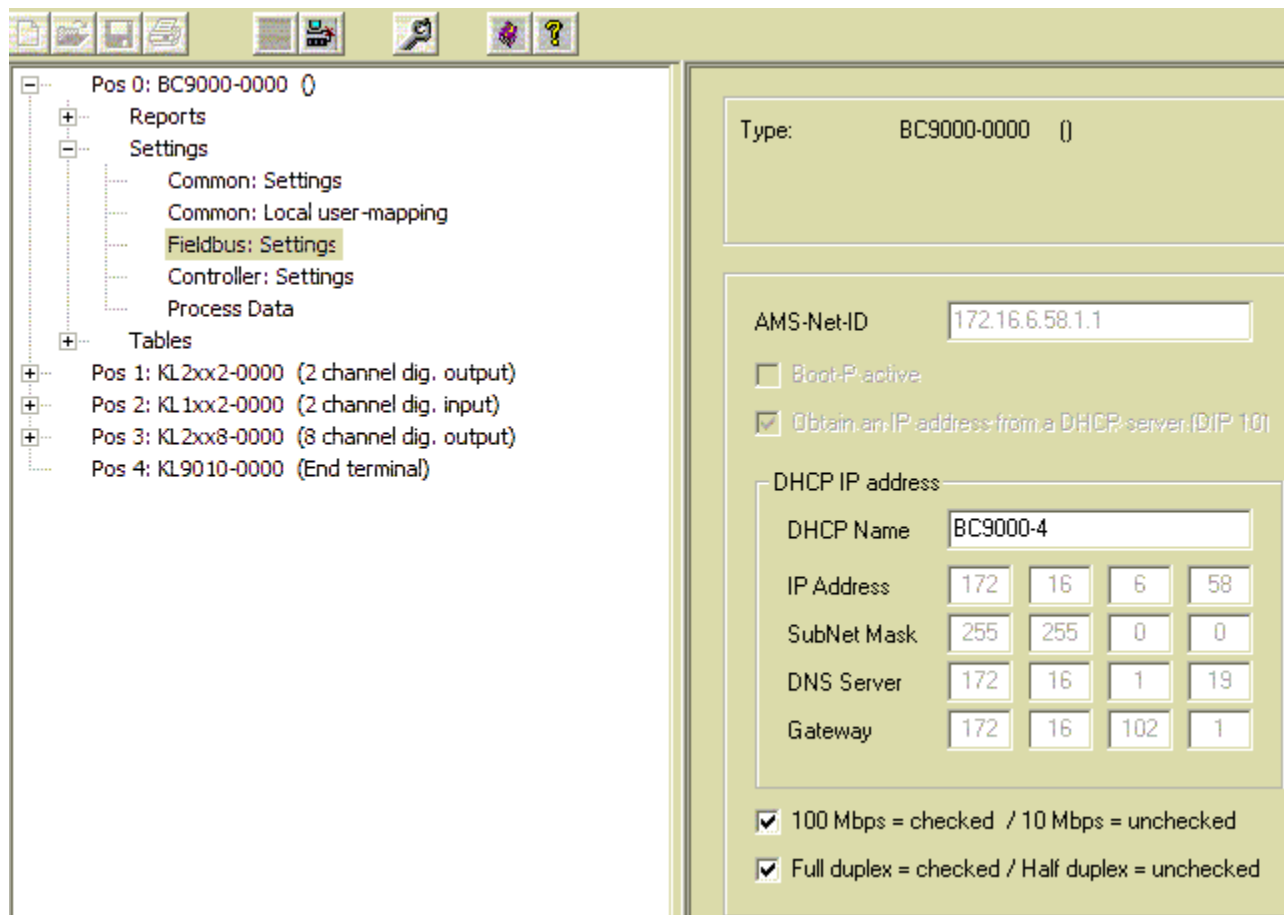


Fig. 15: Addressing by name

Once the IP address was allocated by the DHCP server, the Bus Coupler can be addressed by name.

C:>ping BC9000-4

Addressing via the system manager

In the System Manager an Ethernet Coupler can be found by searching for *GetHostByName*. The DNS server is queried first. If no DNS server is present, or if the DNS server does not recognize the Bus Coupler, the coupler is queried via NetBIOS. If the name of the coupler does not match the default name, the coupler responds with its IP address^{*)}.

For name allocation via NetBIOS, the coupler can be addressed and allocated a name via the KS2000 configuration software (see table 100 [► 78]).

^{*)} From firmware version B8 (BK9000) or firmware version BA (BC9000).

4.4.7 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 space.
- Zeros indicate that part of the address space 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

i Subnets and host number

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 transmitted also by the server.

4.4.8 Testing the IP address

To test the IP address you can use the *Ping* command in a Windows prompt.

```

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>ping 172.16.18.1

Ping wird ausgeführt für 172.16.18.1 mit 32 Bytes Daten:

Antwort von 172.16.18.1: Bytes=32 Zeit<10ms TTL=128
Antwort von 172.16.18.1: Bytes=32 Zeit<10ms TTL=128
Antwort von 172.16.18.1: Bytes=32 Zeit<10ms TTL=128
Antwort von 172.16.18.1: Bytes=32 Zeit<10ms TTL=128

C:\>

```

Fig. 16: Testing the IP address using the Ping command

4.4.9 Reading the MAC-ID

Proceed as follows to read the MAC-ID.

- Change the IP address of your PC to 172.16.x.x. and the subnet mask to 255.255.0.0
In the delivery condition of the BC9000 or BK9000, the IP address is 172.16.17.255 (DIP switches 1 to 8 set to ON).
- Start the DOS window
- Send a *ping >ip-address<* to the IP address 172.16.17.255
- Read the MAC-ID with *arp -a*.

4.4.10 Security settings

The security setting causes the Bus Terminal controller to accept and process TCP/IP or UDP/IP telegrams only from certain TCP/IP devices. If a TCP/IP device is not included in the table, the Bus Terminal controller refuses connections with this device. UDP telegrams from devices that are not entered in the table are rejected. In delivery state the table is empty, i.e. all devices have access to the Bus Coupler.

KS2000 dialog

From KS2000 version 4.3.0.39 security table entries can be made via dialog.

Security settings (IP/UDP):

List of accessibility Computers

Device Name:	IP address:	Flags:
PC 1	172.16.6.183	TCP/IP & UDP/IP
PC 2	172.16.2.2	TCP/IP & UDP/IP
PC 3	0.0.0.0	
PC 4	0.0.0.0	
PC 5	0.0.0.0	
PC 6	0.0.0.0	
PC 7	0.0.0.0	
PC 8	0.0.0.0	
PC 9	0.0.0.0	
PC 10	0.0.0.0	
PC 11	0.0.0.0	
PC 12	0.0.0.0	

Exit

Clean all

Settings

Device Name:

IP address:

Flags

☐ TCP/IP ☐ UDP/IP ☒ TCP/IP & UDP/IP

Remove

Add

Fig. 17: Security settings

5 Configuration

5.1 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.

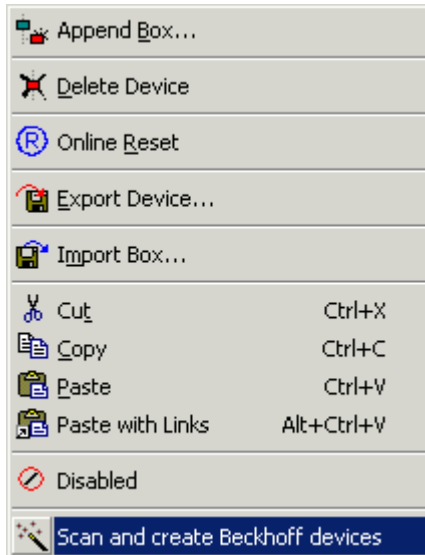


Fig. 18: Reading the connected Bus Couplers and Bus Terminals

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 kB there for each Bus Coupler. This will give, for instance, 3.5 MB (10 x 350 kB) for 10 Bus Couplers.

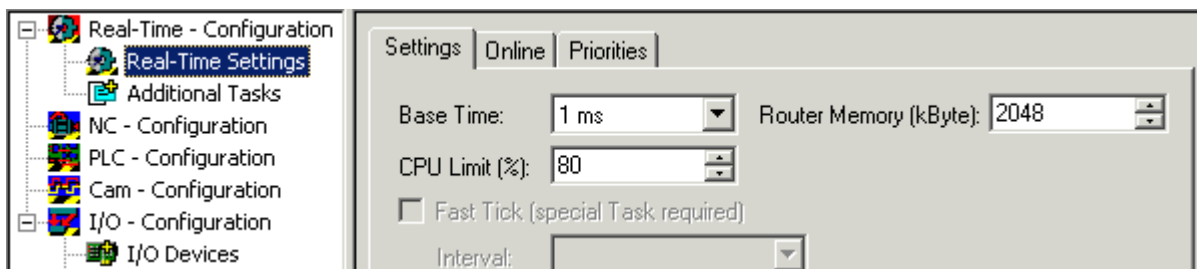


Fig. 19: Setting the router memory

Note: The router memory depends on your PC's main memory, and there is a limit to how high it can be set. Calculation of the maximum possible router memory:

$$\text{max. router memory [MB]} = (1 \text{ MB} + ((\text{RAM [MB]} - 4 \text{ MB}) \times 0.4))$$

Sample

Suppose your PC has 128 MB of RAM:

$$1 \text{ MB} + (128 \text{ MB} - 4 \text{ MB}) \times 0.4 = 50.6 \text{ MB}$$

So the maximum size of the router memory must not exceed 50.6 MB if there is 128 MB of RAM.

5.2 The 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.

Fig. 20: The IP Address tab

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 (as from TwinCAT Version 2.8).

New MAC Addresses

If the [Beckhoff BootP server](#) [► 27] 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 *IO/Reset* function. No outputs will be output as long as no reset has been triggered.

5.3 PLC tab

PLC tab

For Bus Terminal Controllers (BC), special data exchange variables for fieldbus-based communication between PC and BC can be defined under *PLC variables* (see sample configuration of a Bus Terminal Controller).

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

Fig. 21: PLC tab

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 older Bus Terminal Controllers (e.g. BC2000 and BC3100) for compatibility reasons. 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 remanent marker area within the Bus Terminal Controller (%Mx0 ... %Mxyy).

K-bus 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 K-bus update will be carried out before executing the PLC cycle.

After PLC: A K-bus 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 entry *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.

Sample configuration of a Bus Terminal Controller

The image below shows both data exchange variables and fieldbus variables. The Bus Terminal Controller managed terminals 4 to 9 locally.

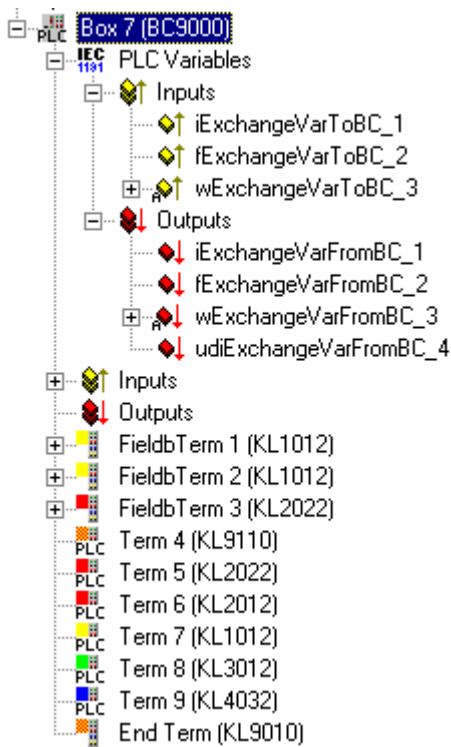


Fig. 22: Sample configuration of a Bus Terminal Controller

5.4 Configuration for ModbusTCP

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

6 Programming

6.1 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 port for software download and for debugging.

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.

6.2 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 runs 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 optimize 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.

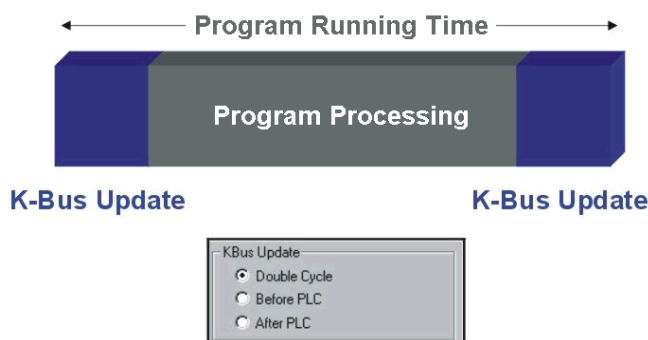


Fig. 23: Program running time

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. **Following the processing of the program, a second K-bus update takes place 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.

Task Time

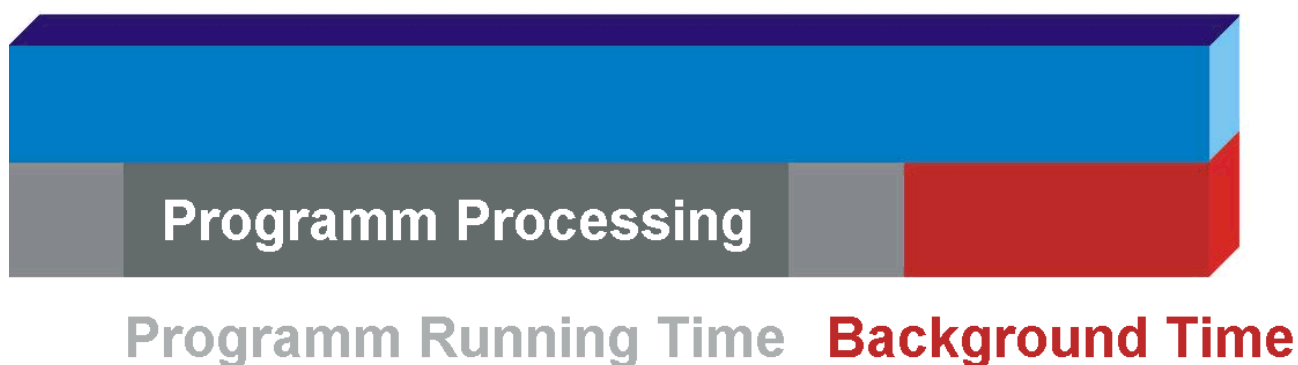


Fig. 24: Task Time

6.3 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. The PLC variables are located from address 128 onwards. You can change both this starting address and the length of these data (default 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.

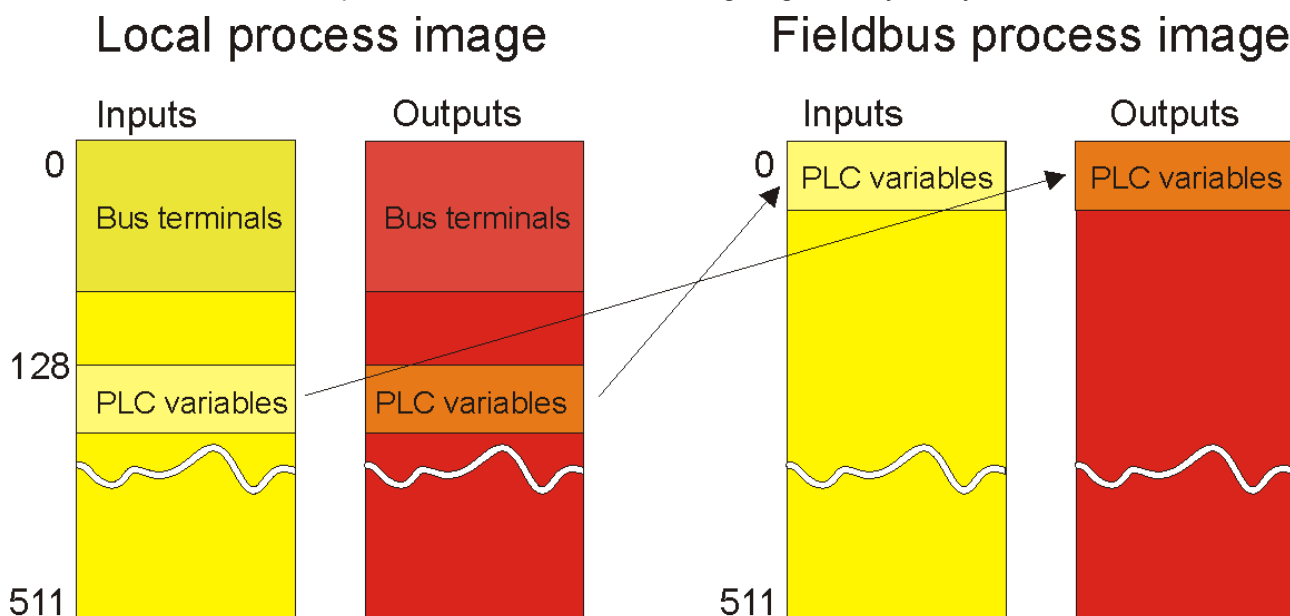


Fig. 25: Bus Terminals in the BC's local process image

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.

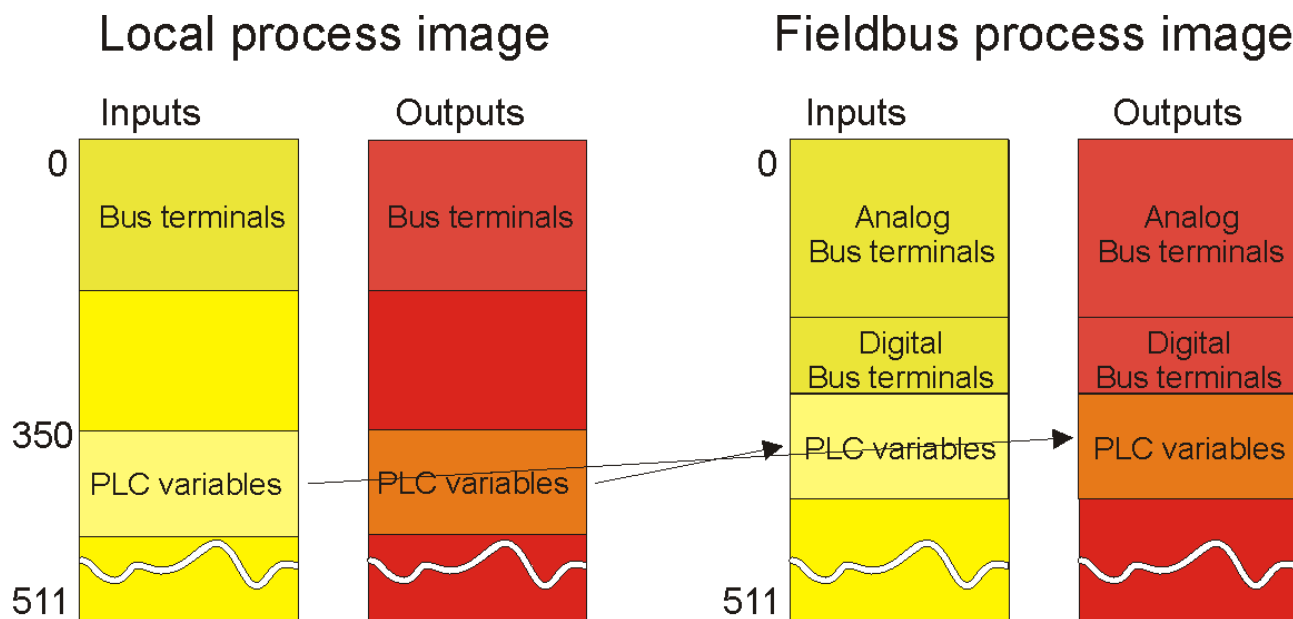


Fig. 26: Assign bus terminals to the higher level controller

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

6.4 Mapping 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.de>.

Byte oriented Bus Terminals	Bit oriented Bus Terminals
KL15x1	KL10xx, KL11xx, KL12xx, KL17xx, KM1xxx
KL25xx	KL20xx, KL21xx, KL22xx, KL26xx, KL27xx, KM2xxx
KL3xxx	
KL4xxx	
KL5xxx	
KL6xxx	
KL7xxx	
KL8xxx	
	KL9110, KL9160, KL9210, KL9260

6.5 Local process image

All connected bus terminals are assigned in the default setting to the local process image of the BC/BX.

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 or BXxxxx
 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 to %IX30.1	-
KL1104	2	%IX30.1 to %IX30.5	-
KL2012	3	-	%QX30.0 to %QX30.1
KL2034	4	-	%QX30.2 to %QX30.5
KL9010	9	-	-

i Address of the Bus Terminals, which you have assigned to local PLCs (BCxx00)

If you don't know the address at which the Bus Terminals are located, which you have assigned to the local PLC of your BC/BX:

Perform your hardware configuration in the System Manager.

Once you have entered all Bus Terminals and PLC variables, right-click on the BC/BX in the hardware tree of the System Manager and select the menu item Export Variable Info...

A file is saved that you can insert in the PLC control under Import Project.

Under the global variables you now have the entry TwinCAT Import in which all variables are located that you have assigned to the local PLC of your BC/BX.

6.6 Allocated flag area %MB..

6.6.1 Remanent and persistent data

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

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 number of persistent data must be greater than or equal to the number 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).

Sample

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

6.6.2 Diagnostics

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
	Bit 1 ... 14	reserved
	Bit 15	No connection to switch: The <i>Link</i> LED does not light up (the transfer of this error information is disabled in delivery state. To enable it, set <u>register 25</u> [► 78] in table 100 to 1 _{bin}).
%MB4094-4095	Bit 0	K-bus error
	Bit 1	Configuration error
	Bit 2	K-bus cycle time exceeded (see table 1, register 17)
	Bit 3 ... 15	reserved

6.6.3 Cycle Time Measuring

The flags area includes a time measurement 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 overflow occurs after approx. 48 days.

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

6.7 Fieldbus process image

6.7.1 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.

6.7.2 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 with the KS2000 configuration software ([register description table 1 \[p. 76\]](#)).

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

6.7.3 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. The setting can be made with the KS2000 configuration software, or, for some BCs that support ADS, can be changed by means of start-up parameters.

6.8 Program transfer**6.8.1 Program transfer via the serial interface**

The Bus Terminal Controller is programmed using TwinCAT.

The program can also be downloaded to the Bus Terminal Controller via the RS232 interface of the PC. A special cable, which is included in the scope of delivery of the KS2000 configuration software, should be used for this purpose.

To this end, select the serial interface in TwinCAT PLC Control.

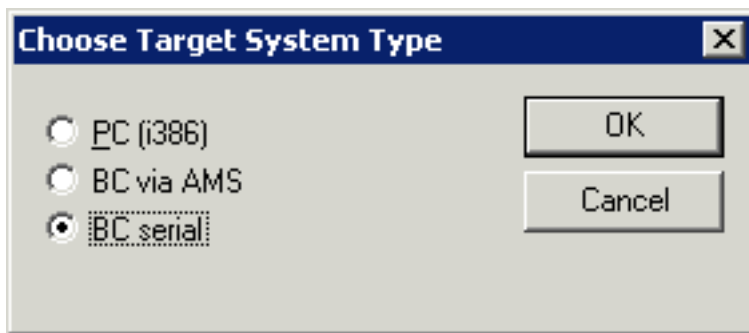


Fig. 27: Selection of the target platform

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

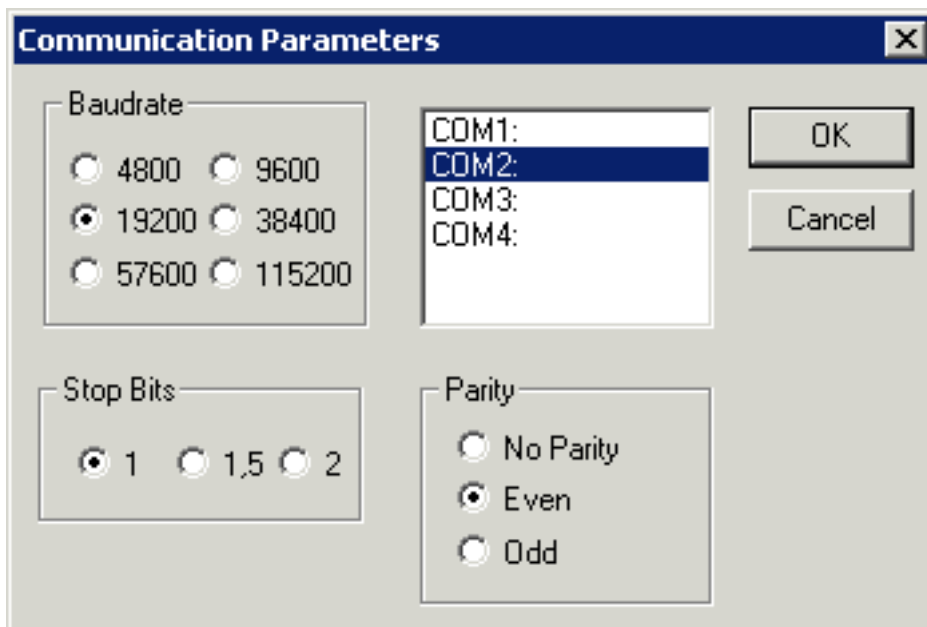


Fig. 28: Communication parameters

6.8.2 Program Transfer 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.

Initializing 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.

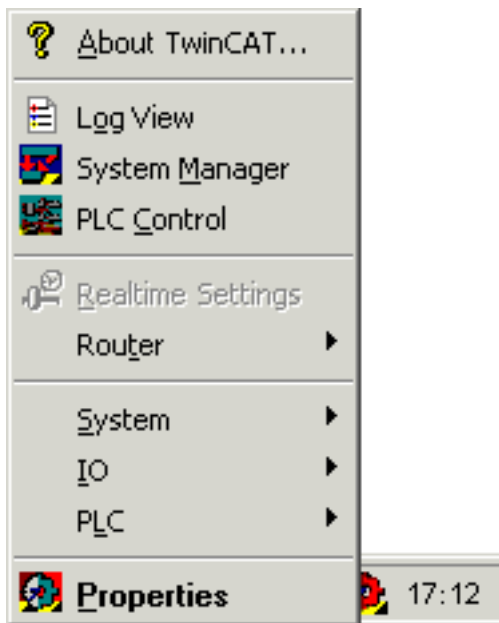


Fig. 29: The menu behind the TwinCAT icon

Select the properties menu item.

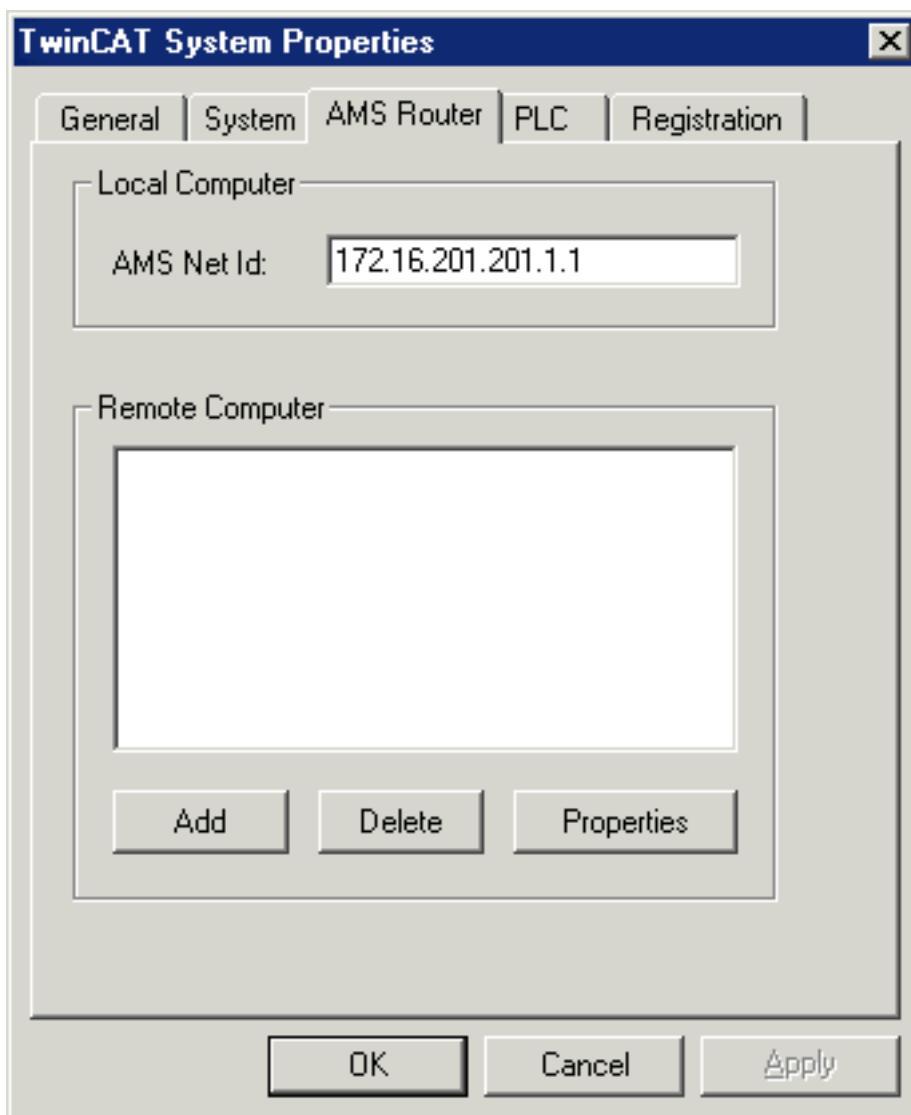


Fig. 30: TwinCAT system properties

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

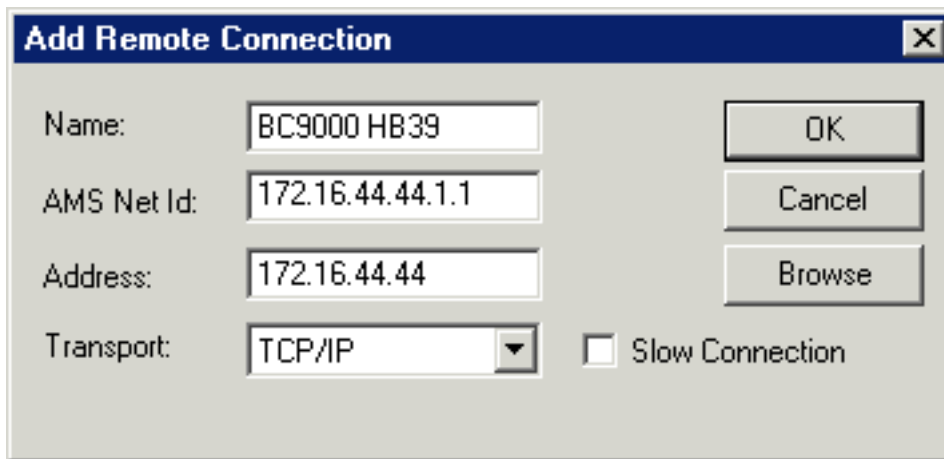


Fig. 31: Add remote connection

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

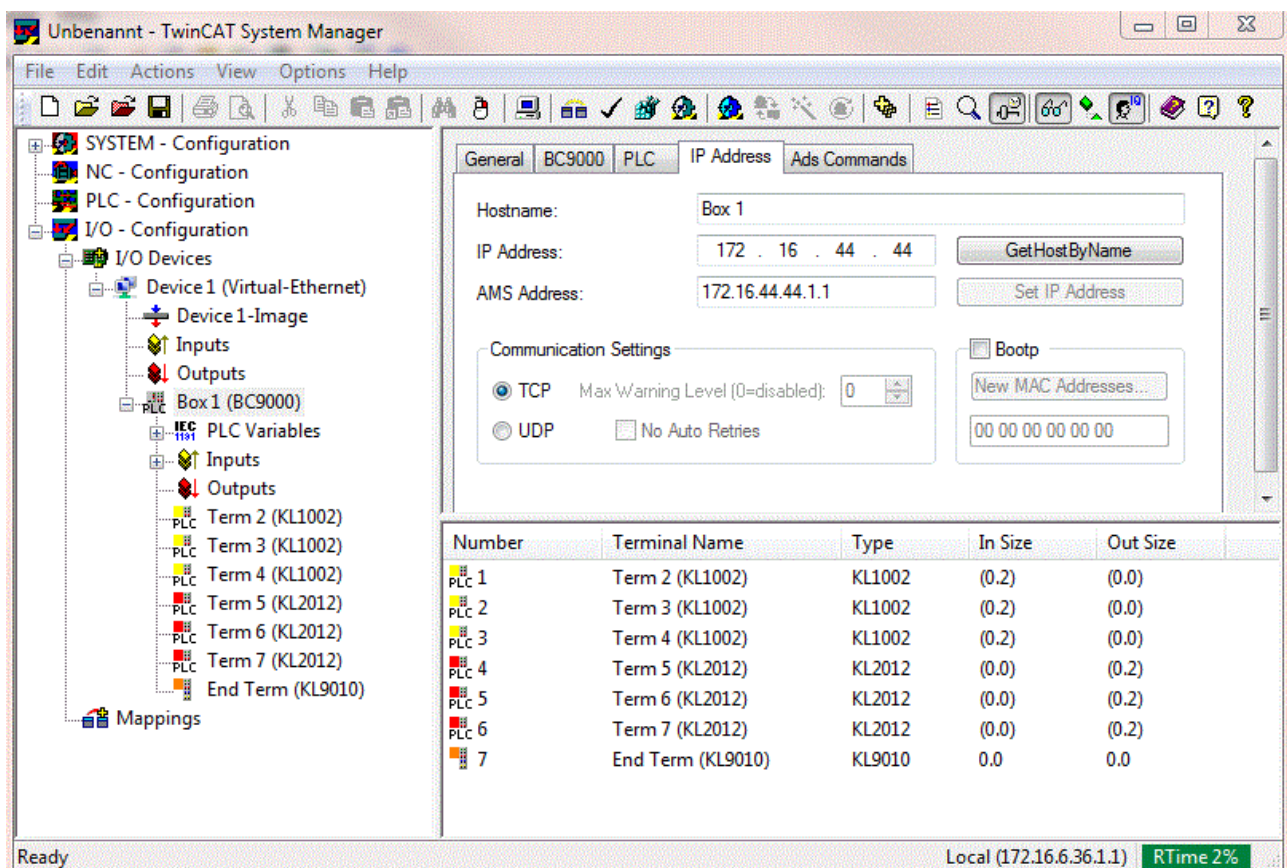


Fig. 32: TwinCAT System Manager

Start the System Manager, and enter your configuration there. In contrast to the first method, this configuration is written to the coupler when TwinCAT is started (e.g. Bus Terminal mapping, PLC cycle time, number of retain data etc.). Other settings 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 as controller, the PC or the Bus Terminal Controller.

Two options 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](#) [► 42] and the Bus Terminal Controller's programming interface

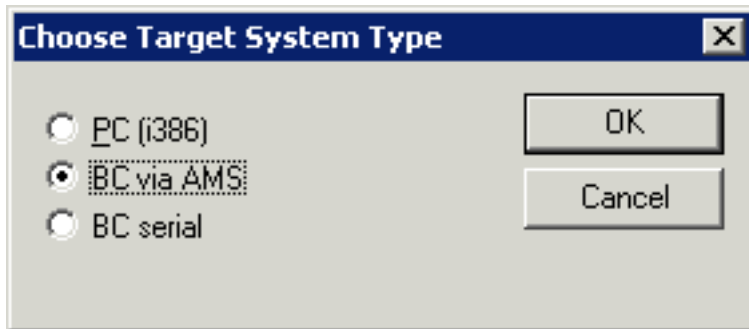


Fig. 33: Selection of the target platform

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

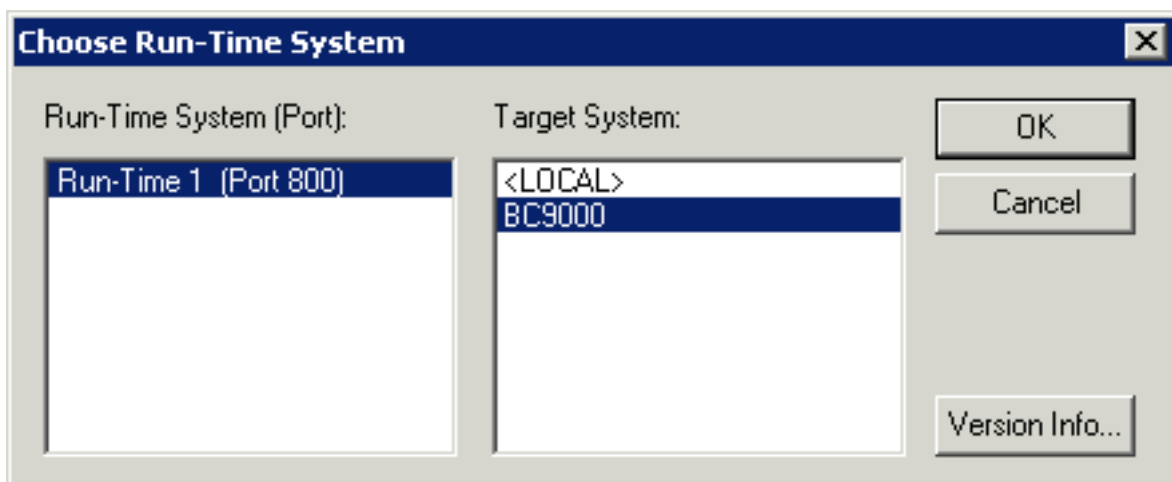


Fig. 34: Choose Target System

6.9 Libraries

6.9.1 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 blocks, 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.lib6* 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 function 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 minimizes 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 target variables. Symbol names are also supported.

Watchdog function 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 (Simple Mail Transfer Protocol).

6.9.2 TcUtilitiesBC9000

6.9.2.1 TcUtilities BC9000



Download library: (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351403531.zip>)

FUNCTION_BLOCK RW_FieldbusPrmBC9000

This function block can be used to modify fieldbus parameter data. Only one instance of this function block is permitted.

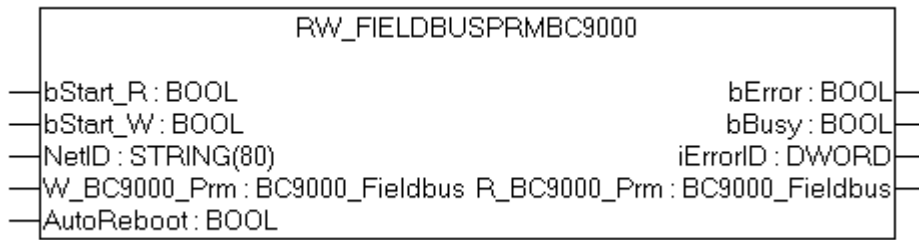


Fig. 35: Function block RW_FIELDBUSPRMBC9000

VAR_INPUT

```

bStart_R      :BOOL;
bStart_W      :BOOL;
NetID         :STRING;
W_BC9000_Prm  :BC9000_Fieldbus;
AutoReboot    :BOOL;

```

bStart_R: Reads the data and outputs it to R_BC9000_Prm

bStart_W: Reads and writes the data of the data structure W_BC9000Prm

NetID: String - AMSNetId

W_BC9000_Prm: Data structure for the parameters

AutoReboot: "TRUE" - the coupler starts and reboots automatically once the parameters have been set successfully, after which the modified settings apply.

VAR_OUTPUT

```

bBusy         :BOOL;
bError        :BOOL;
iErrorId      :DWORD;
R_BC9000_Prm  :BC9000_Fieldbus;
RebootNecessary :BOOL;

```

bBusy: The function block is active as long it is TRUE.

bError: Error bit.

iErrorId: Error number.

W_BC9000_Prm: Fieldbus data structure.

RebootNecessary: The parameters are new, a restart of the BC Controller is required.

```

TYPE BC9000_Fieldbus:
STRUCT
    Disable_ADS      :BOOL;
    WD_ADS           :WORD;
    CloseIP_Port_ADS :WORD;
    DisableModbus     :BOOL;
    WD_Modbus         :WORD;
    Close_IP_Port_Modbus :WORD;
    FastModbus        :BOOL;
    IP_Port_Number_Modbus :WORD;
END_STRUCT
END_TYPE

```

Key

Disable_ADS: Disables the ADS protocol (table 100, offset 15.1)

WD_ADS: ADS watchdog: if no telegram was received after this time, the outputs are set to zero (table 100, offset 14)

Close_IP_Port_ADS: The TCP/IP service is closed after this time, if no more telegrams reach the coupler (table 100, offset 27)

Disable_Modbus: Disables the Modbus protocol (table 100, offset 15.0)

WD_Modbus: ModbusTCP watchdog: if no telegram was received after this time, the outputs are set to zero (table 100, offset 24)

Close_IP_Port_Modbus: The TCP/IP service is closed after this time, if no more telegrams reach the coupler (table 100, offset 28)

FastModbus: Enables Fast Modbus (table 100, offset 29.0)

IP_Port_Number_Modbus: TCP port number (table 100 offset 26)

Return parameter iErrId	Meaning
0	No error
> 0	Error number (#not documented#)

FUNCTION_BLOCK RW_ParameterBC9000

This function block can be used to modify the PLC parameters. Only one instance of this function block is permitted.

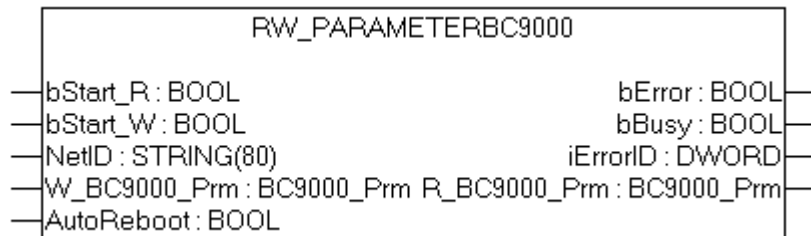


Fig. 36: Function block RW_ParameterBC9000

VAR_INPUT

```
bStart_R      :BOOL;
bStart_W      :BOOL;
NetID         :STRING;
W_BC9000_Prm  :BC9000_Prm;
AutoReboot    :BOOL;
```

bStart_R: Reads the data and outputs it to R_BC9000_Prm

bStart_W: Reads and writes the data of the data structure W_BC9000Prm

NetID: String - AMSNetId

W_BC9000_Prm: Data structure for the parameters

AutoReboot: "TRUE" - the coupler starts and reboots automatically once the parameters have been set successfully, after which the modified settings apply.

VAR_OUTPUT

```
bBusy         :BOOL;
bError        :BOOL;
iErrorId      :DWORD;
R_BC9000_Prm  :BC9000_Prm;
RebootNecessary :BOOL;
```

bBusy: The function block is active as long it is TRUE.

bError: Error bit.

iErrorId: Error number.

W_BC9000_Prm: Fieldbus data structure.

RebootNecessary: The parameters are new, a restart of the BC Controller is required.

```
TYPE BC9000_Fieldbus:
STRUCT
    Byte_Offset_Output      :WORD;
    Byte_Offset_Input       :WORD;
    PLC_Offset_Input        :WORD;
    PLC_Len_Input           :WORD;
    PLC_Offset_Output       :WORD;
    Task_Time               :WORD;
    Background_Time         :WORD;
    RetainData              :WORD;
    PersistentData          :WORD;
    Terminal_Refresh        :WORD;
    Check_Terminals         :BOOL;
END_STRUCT
END_TYPE
```

Key

Byte_Offset_Output: Specifies the offset for the digital output terminals (table 0, offset 19)
Byte_Offset_Input: Specifies the offset for the digital input terminals (table 0, offset 20)
PLC_Offset_Input: Specifies the offset for the fieldbus input data (PLC variables) (table 1, offset 0)
PLC_Len_Input: Specifies the length of the fieldbus input data (PLC variables) (table 1, offset 1)
PLC_Offset_Output: Specifies the offset for the fieldbus output data (PLC variables) (table 1, offset 2)
PLC_Len_Output: Specifies the length of the fieldbus output data (PLC variables) (table 1, offset 3)
Task_Time: Task time (table 1, offset 12)
Background_Time: Background time (table 1, offset 13)
RetainData: Number of retain data (table 1, offset 15)
PersistentData: Number of persistent data (table 1, offset 18)
Terminal_Refresh: K-bus refresh (table 1, offset 17)
Check_Terminals: Deactivate Bus Terminal verification during start-up (table 1 offset 14.6)

Return parameter iErrId	Meaning
0	No error
> 0	Error number (#not documented#)



Download sample: (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351405707.zip>)

6.9.3 TcIpUtilitiesBC

6.9.3.1 Time protocol (SNTP)

(from firmware version BB)

The Simple Network Time Protocol is used to synchronize clocks via the internet. The BC9000 can be synchronized with a time server.



Download library: (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351407883.zip>)



Fig. 37: Function block FB_Sntp

FUNCTION_BLOCK FB_Sntp

From firmware version BA:

If an empty string is entered in variable sSntpServer, the BC9000 will search for a time server. If no server is found the process is terminated. If a time server is found, the IP address can be displayed via the cCurrSntpServer variable. A rising edge of bStart activates the function block and starts the time server query.

From firmware version BB:

If an IP address is entered the Bus Terminal Controller uses the SNTP protocol. If an empty string is transferred, the time protocol (UDP port 37) is used.

VAR_INPUT

```
bStart      :BOOL;  
sSntpServer :STRING(15);  
tTimeOut    :TIME;
```

bOpen: A positive edge starts the function block

sSntpServer: SMTP server entry. If an empty string is entered, the time protocol is used (UDP port 37)*.

tTimeOut: TMOut after which the process is to be terminated

VAR_OUTPUT

```
bBusy      :BOOL;  
bError     :BOOL;  
iErrorId   :WORD;  
tTime      :DT;  
cCurrSntpServer :STRING(15);
```

bBusy: The function block is active as long it is TRUE.

bError: Error bit.

iErrorId: Error number.

tTime: Time and date.

sCurrSnptServer: IP address of the SMTP server

Return parameter iErrId	Meaning
0	No error
> 0	Error number (#not documented#)



Download sample: (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351410059.zip>)

7 Fieldbus system

7.1 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 fiber (10BaseF) or twisted pairs (10BaseT) with screen (STP) or without screen (UTP). Coaxial cable (10Base5), optical fiber (10BaseF) or twisted pairs (10BaseT) with screen (STP) or without screen (UTP).

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 standardized, reliable data exchange between any numbers of different networks. TCP/IP was developed to provide standardized, reliable data exchange between any numbers of different networks. Although the term is often used as if it were a single concept, a number of protocols are layered together: z. B. 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 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.

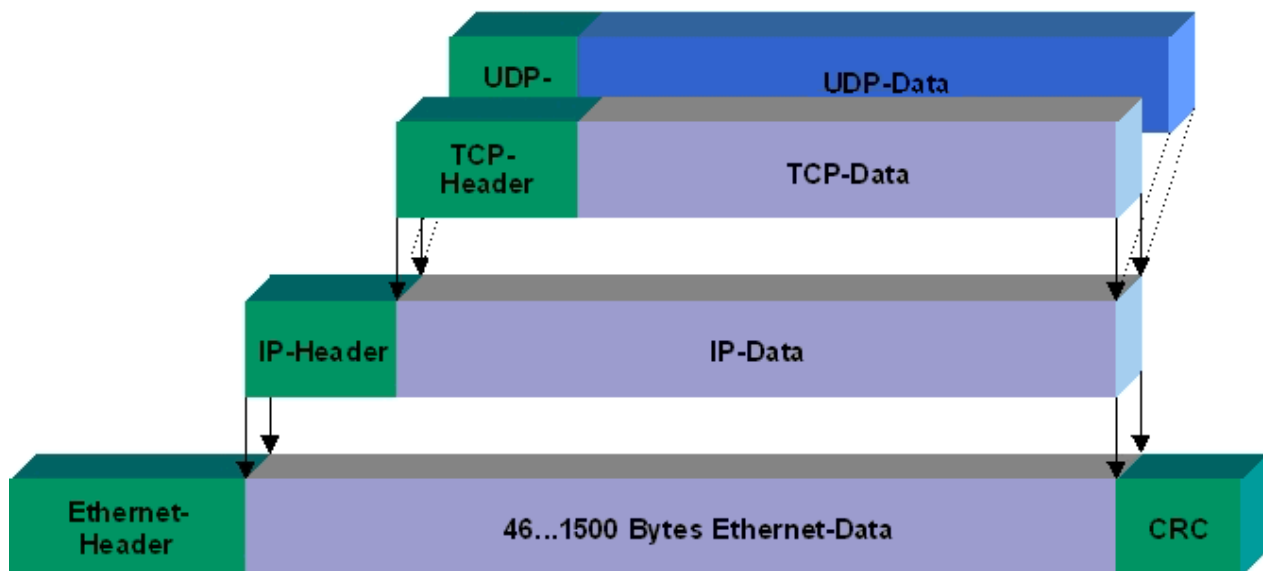


Fig. 38: User Datagram Protocol (UDP)

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.

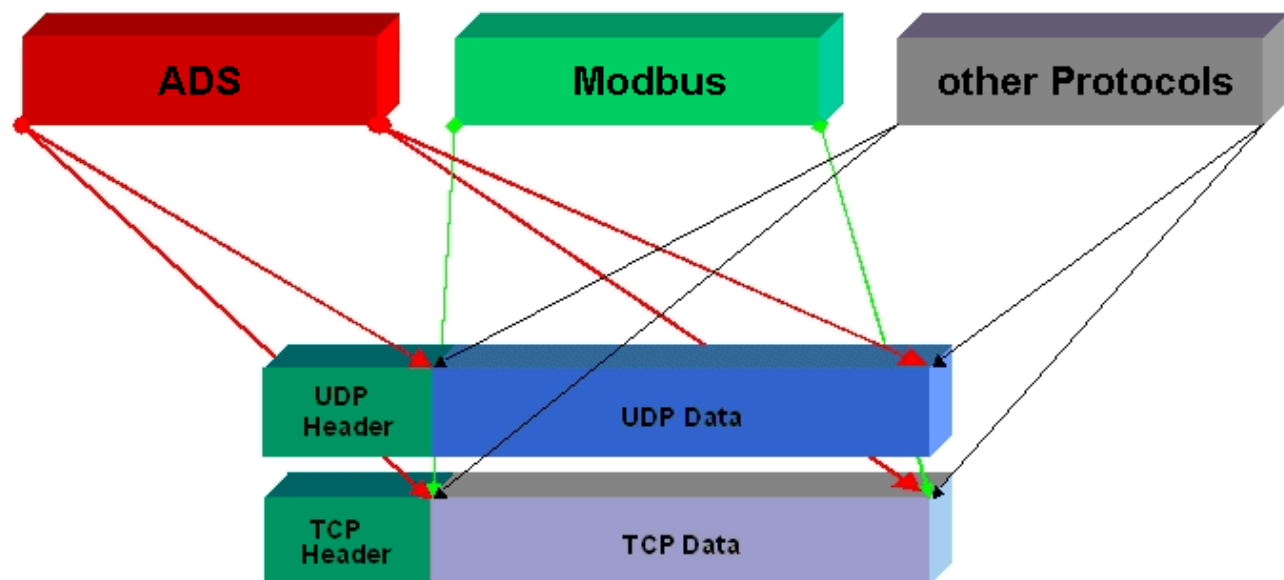


Fig. 39: Protocols running on top of TCP/IP and UDP/IP

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

7.2 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. Standardized 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 meters, 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.

7.3 Reaction times

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](#)) [► 66]

BK9000 reaction time

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

RT-Ethernet**	Time [ms]
RAW	< 1 ms + K-bus (depending on the K-bus propagation delay)

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) [► 66]** from firmware B8

7.4 ADS-Communication

7.4.1 ADS-Communication

The ADS protocol (ADS: Automation Device Specification) is a transport layer within the 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 enables communication with other tools from any point within the TwinCAT. If communication with other PCs or devices is required, the ADS protocol can use TCP/IP as a basis. Within a networked system it is thus possible to reach all data from any point.

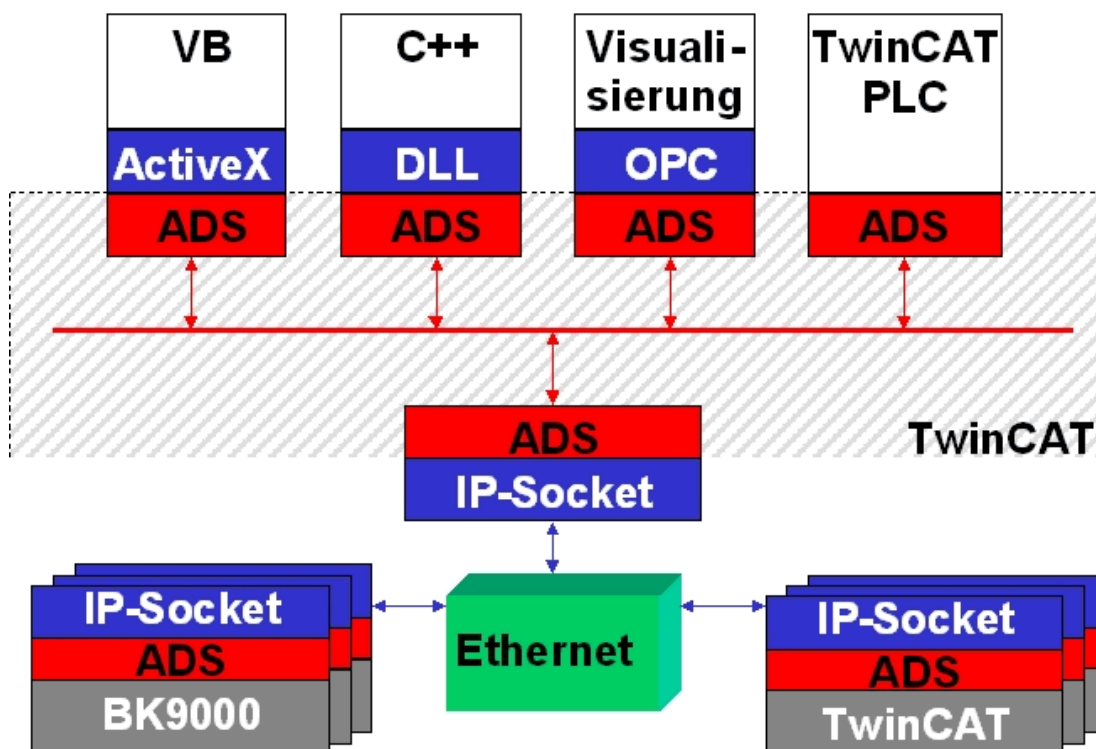


Fig. 40: The ADS protocol as a transport layer within the TwinCAT system

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 parameterize 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.

7.4.2 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 *TcSystem.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.

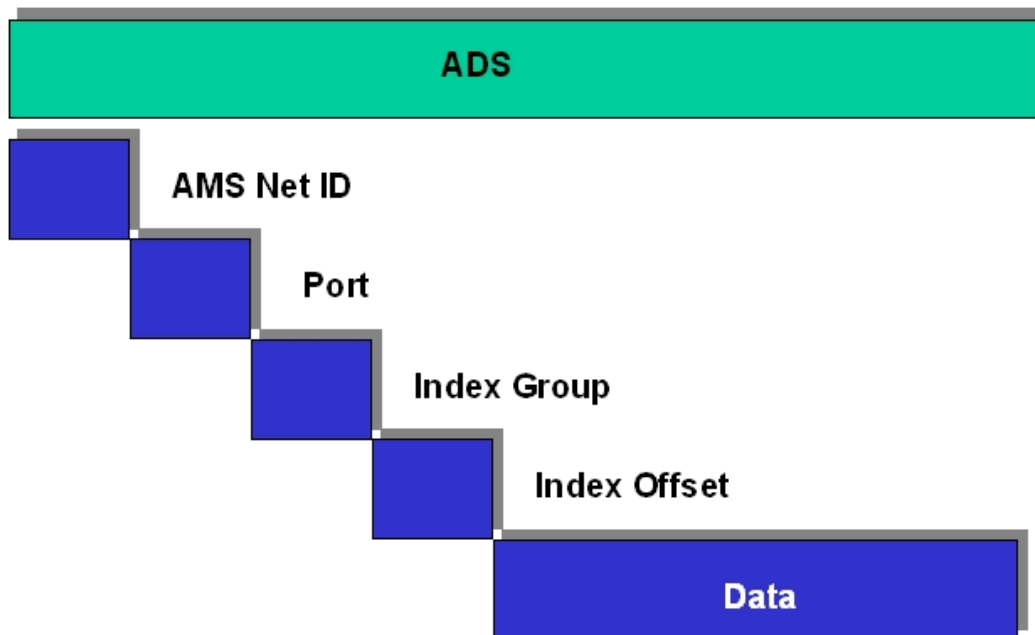


Fig. 41: Structure of the ADS protocol

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 cannot be altered.

Sample:

IP address 172.16.17.128

AMSNetID 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, C900 only)

Index group

The index group distinguishes different data within a port.

Index offset

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

Len

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

TCP port number

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

7.4.3 ADS services

7.4.3.1 ADS services

**User data of an ADS telegram**

The maximum size of the user data in an ADS telegram is 1900 bytes.

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, C900 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 are 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, C900 only)	0...4096

ADS services**AdsServerAdsStateB7**

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

AdsServerDeviceStateB7

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

AdsServerTypeB7

Data type (read only)	Meaning
String	Coupler_PLC

ADSWriteControlBA

Data type (write only)	Meaning
NetID	Net ID of the BC9000,C900
Port	800
ADSSTATE	5 - RUN / 6 - STOP
DEVSTATE	0
LEN	0
SRCADDR	0
WRITE	rising edge starts the block
TMOU	example: t#1000 ms

● Acyclic data communication

i With acyclic data communication it is important to note that about 20 to 30 ms are required in order to establish a TCP/IP connection. Following the successful establishment of a connection the ADS data are sent or read. An ADS read of 1000 bytes takes about 50 ms. If no data are transmitted for 10 seconds, the TCP/IP connection is disconnected from the BC/BK9000, B/C900.

Register port 100

The ADS port number in the BK/BC9000, B/C900 for register communication is fixed, being set at 100.

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

● Timeout of the ADS function block

i When reading the register, the time out of the ADS block has to be set to a time longer than 1 second.



Setting the password

When writing to the registers, the password has to be set (see the documentation for the particular Bus Terminal).

Access control and IP - AMS-Net ID assignment^{B6}

The *AMS Net-Id* table permits access control to the BC 9000, C900 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. Furthermore, an assignment of the AMS-Net ID to the IP address of the node is explicitly done 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 byte
IP address	4 byte
Reserve	2 byte
Reserve	4 byte
Reserve	4 byte

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 byte
1..10	Entry n (1...10)	Data structure, 20 bytes



First entry

The first entry must be 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

)² as from firmware BA

See the [example](#) [► 60].

7.4.3.2 Further ADS services

Flash data storage*

(from firmware version BA)

Process data port 0x1234

This service enables further data to be stored in flash memory. A range of 16,000 bytes is available to the user.

Index group	Meaning	Index offset (value range)
0x4000_FE00	Flash data memory	0...0x3EFF



read and write data

- Two IP sockets are required for the call. Please ensure that the maximum number of open IP sockets is not exceeded.
- A flash has limited write capability! Please ensure that data are only written to the flash if actually required. Otherwise use remanent data.
- A maximum of 1500 bytes can be read or written. The byte address of the memory can be specified via the index offset.

Writing of data requires a certain amount of time:

During normal operation, writing of 200 bytes takes approx. 120 ms, 600 bytes approx. 150 ms.

Reading of 200 bytes takes approx. 50 ms, 600 bytes approx. 60 ms.



Sample Program in ST (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351412235.zip>)

7.4.3.3 Samples

7.4.3.3.1 AMS routing table

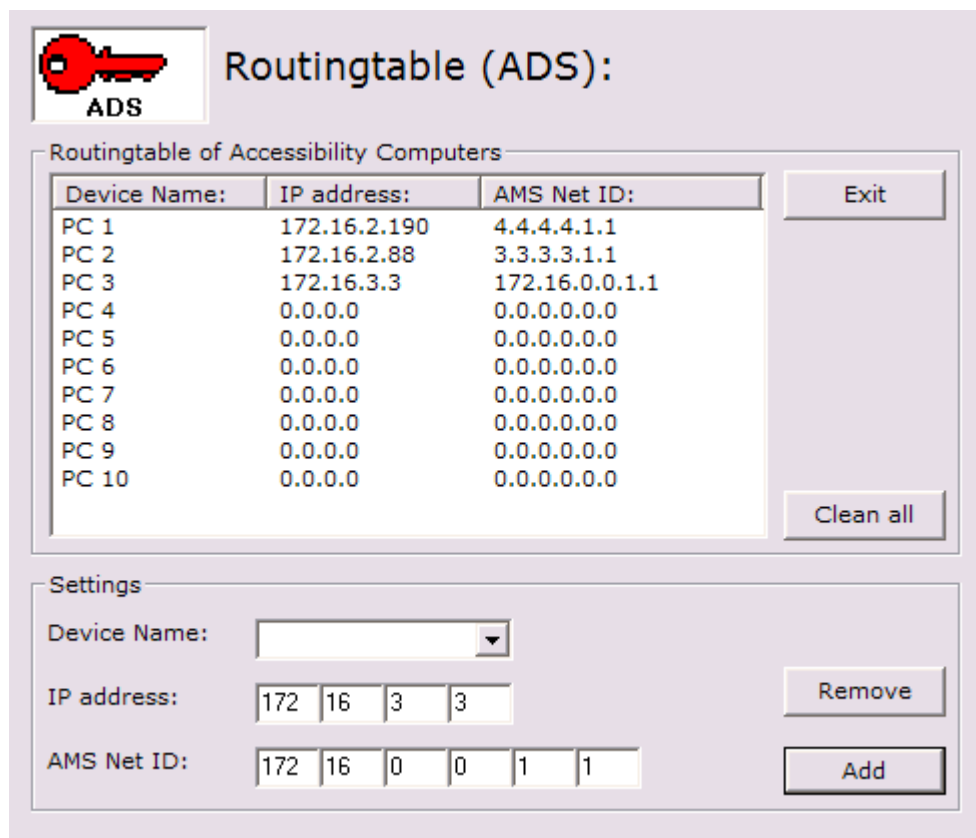
The AMS routing table can be used for two functions:

1. If the AMS Net ID does not match the TCP/IP address, the link between the two non-matching numbers can be entered here.
The AMS Net ID matches the TCP/IP address if the first 4 bytes are identical and the AMS Net ID ends with ".1.1".
2. Only AMS Net IDs stored in the table can access the BC9000 via ADS.

Please ensure that all ADS devices with which communication is required are entered in the table. If the table is empty (default), all ADS devices can access the BC9000.

Processing the AMS routing table using the KS2000 configuration software

From KS2000 version 4.3.0.39 entries in the AMS routing table can be made via dialog.



Routingtable (ADS):

Routingtable of Accessibility Computers

Device Name:	IP address:	AMS Net ID:
PC 1	172.16.2.190	4.4.4.4.1.1
PC 2	172.16.2.88	3.3.3.3.1.1
PC 3	172.16.3.3	172.16.0.0.1.1
PC 4	0.0.0.0	0.0.0.0.0.0
PC 5	0.0.0.0	0.0.0.0.0.0
PC 6	0.0.0.0	0.0.0.0.0.0
PC 7	0.0.0.0	0.0.0.0.0.0
PC 8	0.0.0.0	0.0.0.0.0.0
PC 9	0.0.0.0	0.0.0.0.0.0
PC 10	0.0.0.0	0.0.0.0.0.0

Exit

Clean all

Settings

Device Name:

IP address:

AMS Net ID:

Remove

Add

Fig. 42: Displaying the AMS routing table using the KS2000 configuration software

Entering values in the routing table via ADS

The AMS routing table can be loaded into the BC9000 via ADS. Ensure that the first entry is that of the sender, since this setting becomes effective immediately and may otherwise prevent further entries in the AMS routing table.

 Download (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/6744240267.zip>)

7.4.3.3.2 Example communication between BC9000s

ADS-Communication

Data can be sent directly to another BC9000 using the ADS function blocks from the TcADSBC.lbx library. ADSREADEX can be used to read data from another BC9000, ADSWRITE can be used for writing. A maximum of 3 BC9000 can be accessed at the same time. In order to communicate with further BC9000s, the connection to at least one BC9000 has to be closed via ADSCLOSE. Make sure that only a maximum of 3 connections to a BC9000 exist (the 4th connection should be reserved for programming). These connections may also be established by another remote station.

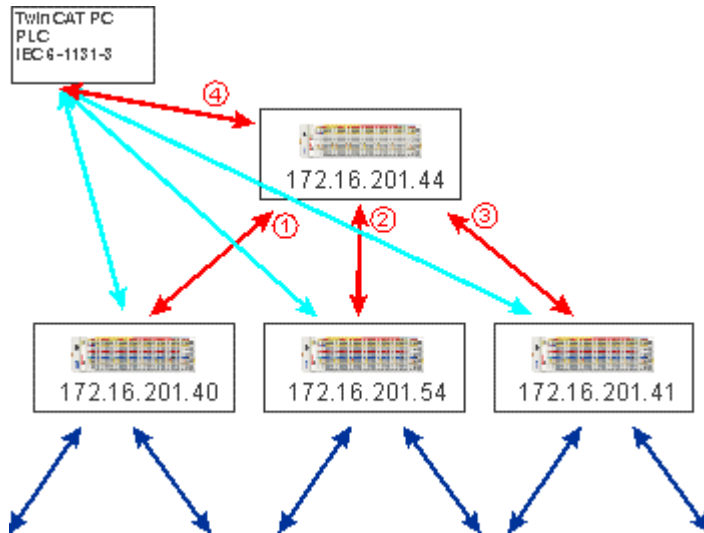
Example for a functioning ADS connection

Fig. 43: ADS connection

Sample program for connecting BC9000 to BC9000

 Sample program in ST Master (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351416587.zip>)

 Sample program in ST Slave (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351418763.zip>)

**Cross-communication better between BCs than to the BK**

Please note that a BK9000 only allows write access from a master, and the BK9000 must be polled regularly at intervals of one second. It is therefore advisable to implement cross communication only between BC9000 and BC9000.

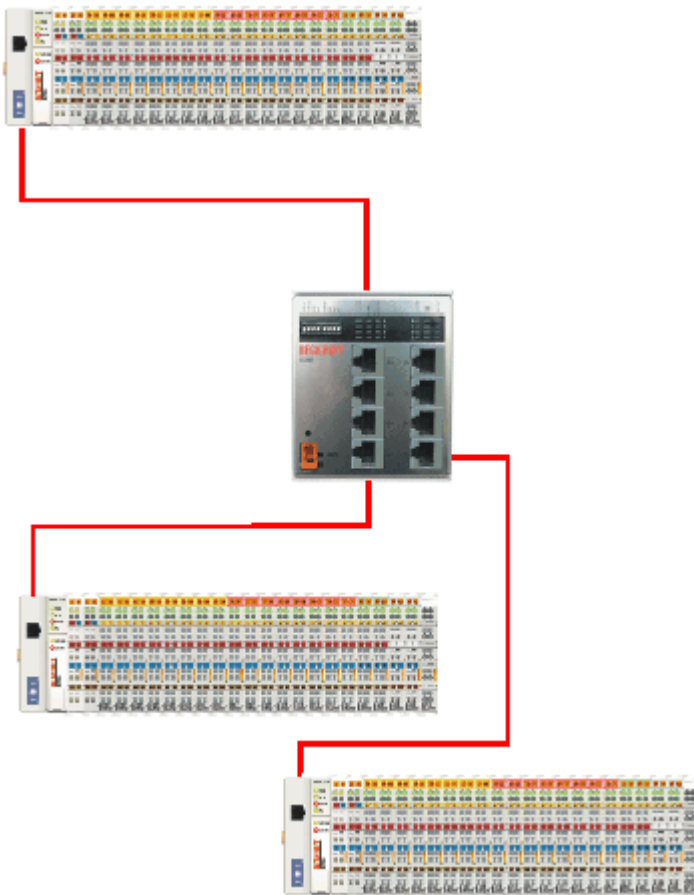


Fig. 44: Bus structure for cross communication

7.4.3.3.3 Example communication from BC9000 to a TwinCAT PC system

In some applications polling operation (master-slave principle) is not wanted. The ADS function blocks of the BC9000 enable both cross communication (from BC9000 to BC9000) and communication to the higher-level TwinCAT PC system. This means that the BC9000 application can decide when data are sent to the higher-level system. Any fault information, for example, is transferred immediately to the TwinCAT PC.

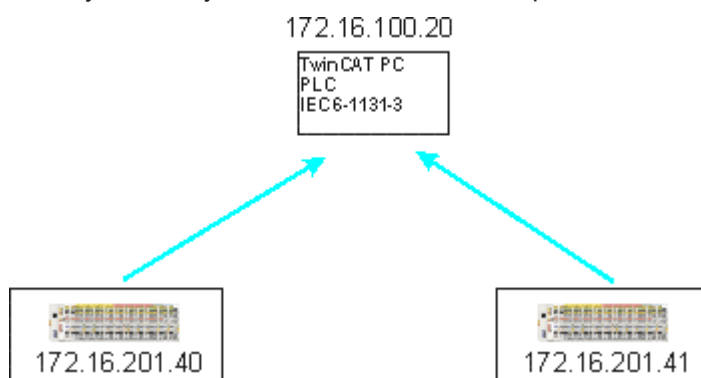


Fig. 45: Cross communication from one PC to two BC9000s

Requirement

TwinCAT 2.9 Build 1025

The IP address matches the AMS Net ID plus ".1.1"

An >ipconfig< command issued in a DOS window can be used to find out the IP address of your PC.

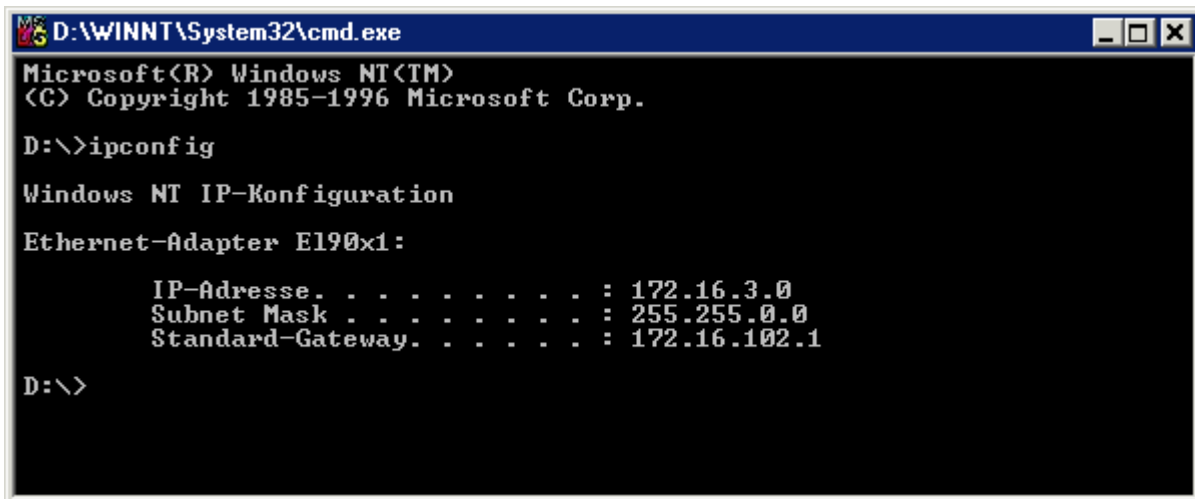


Fig. 46: Command >ipconfig< in the DOS window

The AMS Net ID is displayed under Properties/AMS router.

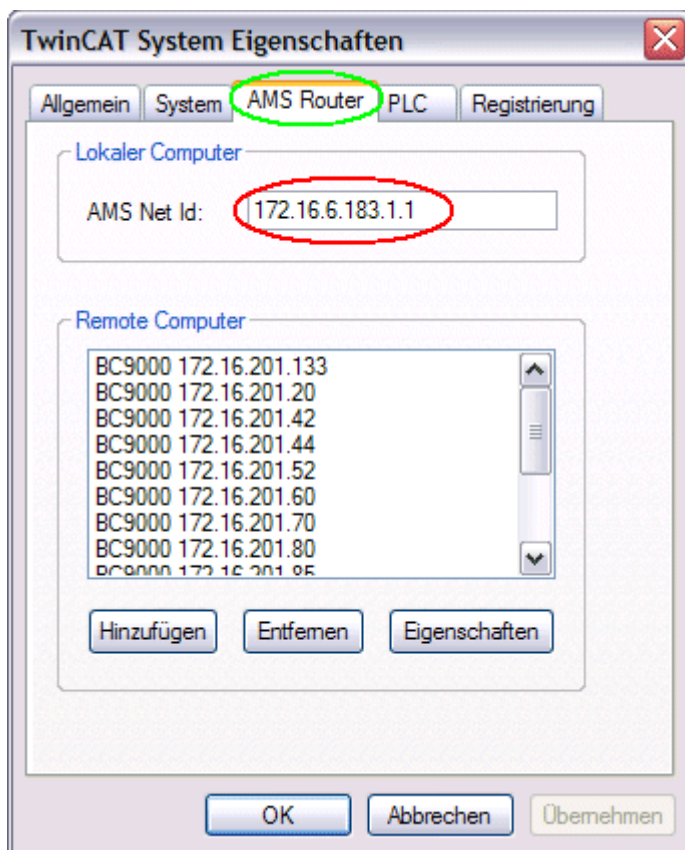


Fig. 47: Display of the AMS Net ID

If the AMS Net ID does not match the TCP/IP address, the link can be entered in the routing table (table 2).

Example for AMS Net ID OK

AMS Net ID: 172.16.100.10.1.1

TC/IP: 172.16.100.10

Required library: AdsBC.Ib6

sample programs for the connection of BC9000 to a TwinCAT PC

 Sample program in ST Master (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351420939.zip>)

 Sample program in ST Slave (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351423115.zip>)

7.4.3.3.4 Sample: Changing the Ethernet Coupler settings via ADS

Requirement

For this example a BC9000 is required. Please note two local sockets are required for reading or writing!

Required library:

AdsBC.lb6
TcPlcUtilitiesBC.lb6
PlcSystemBC.lb6
PlcHelperBC.lb6
TcUtilitiesBC9000.lb6

Description

The function block enables setting of various parameters (see [sample \[► 47\]](#)). A more detailed description of the function block can be found in the *Library* chapter.

Example:

 **Download sample program in ST** (<https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351425291.zip>)

7.5 ModbusTCP

7.5.1 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 parameterized 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.

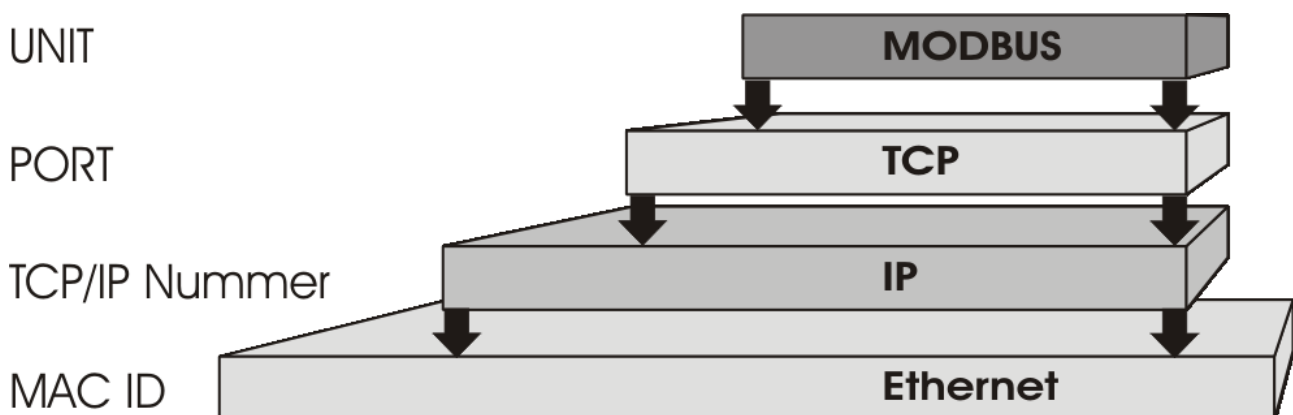


Fig. 48: ModbusTCP Protocol

TCP port number

The TCP port number for ModbusTCP has been standardised to 502.

Modbus-Unit

The unit is returned by the slave.

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	returned by the slave
7	Modbus	Modbus protocol beginning with the function follows

7.5.2 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..)*			

* all Bus Terminal controllers BC9xx0 and BX9000

** for BC9x00 from firmware B7 and BK9000 from firmware B5 and all unlisted BK9xxx and BC/BX9xxx

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

Legend

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 K-bus reset to be carried out. The 2-byte PLC interface requires two bytes each of input and output data. They are handled using a special protocol. 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. K-bus diagnostics must however be activated for this purpose. The 2-byte diagnostic interface occupies two bytes each of input and output data. A special protocol is processed via these two bytes. A description of the 2 byte-diagnostic interface can be supplied on request.

7.5.3 ModbusTCP slave error answer (BK9000, BX/BC9xx0, IP/ILxxxx-B/C900, EK9000)

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 EK9000: E-bus error
6	SLAVE DEVICE BUSY	Output data is already been received from another IP device

7.5.4 ModbusTCP functions

7.5.4.1 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 [► 68]	1	Read digital outputs
Read input status [► 69]	2	Read digital inputs
Read holding register [► 69]	3	Read analog outputs and inputs / GPR
Read input register [► 70]	4	Reading the inputs / GPR
Force single coil [► 71]	5	Writing a digital output
Preset single register [► 71]	6	Writing an output / GPR
Diagnostics [► 72]	8	Diagnostics
Force multiple coils [► 73]	15	Write a number of digital outputs
Preset multiple register [► 74]	16	Writing several outputs / GPRs
Read / write registers [► 74]	23	Write and read a number of process data outputs / GPRs

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

7.5.4.2 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	Sample
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 is set in the output process image, and the fieldbus coupler shows the value 4 in the first data byte.

Response

Byte name	Sample
Function code	1
Byte count	2
Data bits 0...7	4
Data bits 8...18	0

7.5.4.3 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	Sample
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 is set in the output process image, and the fieldbus coupler shows the value 4 in the first data byte.

Response

Byte name	Sample
Function code	2
Byte count	2
Data bits 0..7	1
Data bits 8..18	0

7.5.4.4 Read holding register (Function 3)

The *Read holding register* function can be used to read the input and output words and the registers. Inputs from offset 0 - 0xFF and outputs from offset 0x800 - 0x8FF, and for controllers (BC, BX) the flag area from offset 0x4000.

In this example the first two analog outputs (or two output words) are read. The analog outputs (or output words) start at offset 0x800. The length indicates the number of channels (or words) to be read.

Query

Byte name	Sample
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	Sample
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

7.5.4.5 Read input register (Function 4)

The function *Read input register* reads the inputs on a word basis.

In this example the first two analog inputs (or the first two input words) are read. The analog inputs (or input words) start at an offset of 0x0000. The length indicates the number of words to be read. A KL3002, for example, has two words of input data. Therefore, the length to be entered at *Number low* is two.

Query

Byte name	Sample
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	Sample
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

7.5.4.6 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	Sample
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	Sample
Function code	5
Start address high	0
Start address low	2
Data high	255
Data low	0

7.5.4.7 Preset single register (Function 6)

The function *Preset singles register* can be used to access the output or flag process image (only for controllers) and the [Modbus TCP interface \[► 66\]](#).

Function 6 writes the first output word. The 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	Sample
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	Sample
Function code	6
Start address high	8
Start address low	0
Data high	63
Data low	255

7.5.4.8 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 sub-function code defining the test that is to be carried out. The slave returns the function code and the sub-function 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	Sample
Function code	8
Sub-function high	0
Sub-function low	0
Data high	2
Data low	3

Response

Byte name	Sample
Function code	8
Sub-function high	0
Sub-function 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)

The coupler is reset with sub-function 1, error counters are reset and the controller performs a self-test. No telegrams are either received or sent while the controller is being reset. The IP socket is closed.



Last reply with the sub-function 1

Before the Controller restarts it sends a reply with sub-function 1, after which the IP socket is closed.

Sub-function	Data field (query)	Data field (response)
0x0001	0x0000	0x0000

Delete all counter contents (sub-function 10)

When this sub-function is called the controller clears all error counters.

Sub-function	Data field (query)	Data field (response)
0x000A	0x0000	Echo query data

Bus Communication Answer Counter (Subfunction 11)

Returns the number of communication answers.

Sub-function	Data field (query)	Data field (response)
0x000B	0x0000	Counter value

Error Answer Counter (Subfunction 13)

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

Sub-function	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 coupler is accessed by different masters.

Slave Answers (Subfunction 14)

Contains the number of answers that the slave has sent.

Sub-function	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.

Sub-function	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.

Sub-function	Data field (query)	Data field (response)
0x0010	0x0000	Counter value

7.5.4.9 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 composed from the combination all the bytes that are to be written.

Sample: 20 bits count is 3 bytes (round up to the nearest byte)

The data bytes contain the values for the individual bits. In this example, the bits 0 to 15 are set to TRUE, while bits 17 to 20 are set to FALSE.

Query

Byte name	Sample
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	Sample
Function code	15
Start address high	0
Start address low	0
Length high	0
Length low	20

7.5.4.10 Preset single register (Function 16)

The *Preset multiple register* function can be used to write a number of outputs. The first two analog output words are written in this example. The 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.

Sample: 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	Sample
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	Sample
Function code	16
Start address high	8
Start address low	0
Length high	0
Length low	2

7.5.4.11 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 / write 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. Sample: 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	Sample
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	Sample
Function code	23
Byte count	4
Data 1 high	0
Data 1 low	56
Data 2 high	63
Data 2 low	11

7.5.5 Examples for ModbusTCP

Examples for Modbus TCP are contained in this help text.

Simple example: Modbus TCP via VB6.0 <https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351427467.zip>

Example: ModbusTCP via VB6.0, set and reset of the watchdog: <https://infosys.beckhoff.com/content/1033/bc9000/Resources/zip/4351429643.zip>

7.6 Description of parameters

7.6.1 Properties of the Bus Terminal Controller

7.6.1.1 Register settings, Table 0

Using the KS2000 configuration software you can set the register values via the dialog window or write them directly into the registers.

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

7.6.1.2 Register settings, Table 1

Using the KS2000 configuration software you can set the register values via the dialog window or write them directly into the registers.

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 1st Bus Terminal	11 _{bin}	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
	00 _{bin} Fieldbus process image complex		
	10 _{bin} Fieldbus process image compact		
	11 _{bin} Local process image		
4.2...4.3	Assignment of the 2st 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
12	PLC cycle time	5 ms (20 ms in the case of BC9000/ BC9100)	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
13	Background Time	2 ms (10 ms in the case of BC9000/ BC9100)	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
14.0	Enable remanent data	1 _{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		
14.6	Switch on Bus Terminal checking	1 _{bin}	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
15	Size of the NOVRAM (remanent data)	64 [bytes]	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
17.0...17.7	Terminal bus autorefresh: Cycle in [ms]	0x0000 Disable (0x005F for BC9000) (0x035F for BC9100)	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
17.8...17.15	Terminal bus autorefresh: max. retries		
18	Size of the NOVRAM Persistent data R18<R15	0	BC2000, BC3100, BC4000, BC7300, BC8x00, BC9000
27	Byte offset KL6xx1 bus terminal emulation %Q...	500	BC8x00
28	Byte offset KL6xx1 bus terminal emulation %I...	500	BC8x00
32	Baud rate	2	BC8x00
	0 38400 Bd		
	1 19200 Bd		
	2 9600 Bd		
	3 57600 Bd		
	4 1200 Bd		
	5 2400 Bd		
	6 4800 Bd		
33	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		
34	Number of stop bits	0	BC8x00
	0 one stop bit		
	1 two stop bits		

7.6.1.3 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, BC9100, BK9000, BK9050, BK9100
2-3	IP mask	0xFF, 0xFF, 0x00, 0x00	BC9000, BC9100, BK9000, BK9050, BK9100
4-13	Device Name	BC9000	BC9000, BC9100, BK9000, BK9050, BK9100
14	Watchdog AMS/ADS	1000 ms	BC9000, BC9100, BK9000, BK9050, BK9100
15.0	0 _{bin} enable ModbusTCP	0 _{bin}	BC9000, BC9100, BK9000, BK9050, BK9100
	1 _{bin} disable ModbusTCP		
15.1	0 _{bin} enable AMS/ADS	0 _{bin}	BC9000, BC9100, BK9000, BK9050, BK9100
	1 _{bin} disable AMS/ADS		
16.8	0 _{bin} Ethernet mode half duplex	1 _{bin}	BC9000, BC9100, BK9000, BK9050, BK9100
	1 _{bin} Ethernet mode full duplex		
16.12	0 _{bin} Autobaud off	1 _{bin}	BC9000, BC9100, BK9000, BK9050, BK9100
	1 _{bin} Autobaud on		
16.13	0 _{bin} 10 Mbaud	1 _{bin}	BC9000, BC9100, BK9000, BK9050, BK9100
	1 _{bin} 100 Mbaud		
17-18	Default Gateway	0x00, 0x00, 0x00, 0x00	BC9000, BC9100, BK9000, BK9050, BK9100
24	Watchdog ModbusTCP	1000 ms	BC9000, BC9100, BK9000, BK9050, BK9100
25.0	Entry of a fieldbus error (connection to switch interrupted) in the <u>flags area diagnosis</u> [► 40]		BC9000, BC9100
	0	Error was not entered	
	1	Error was entered	
26	ModbusTCP port no.	502	BC9000, BC9100, BK9000, BK9050, BK9100
27	ADS connection service life	10 s	BC9000, BC9100, BK9000, BK9050, BK9100
28	Modbus connection service life	10 s	BC9000, BC9100, BK9000, BK9050, BK9100
29.0	ModbusTCP mode		BC9000, BC9100, BK9000, BK9050, BK9100
	1 _{bin}	Fast ModbusTCP	

8 Error handling and diagnosis

8.1 Diagnostic LEDs

BK9000, BK9100, BC9000, BC9100

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.

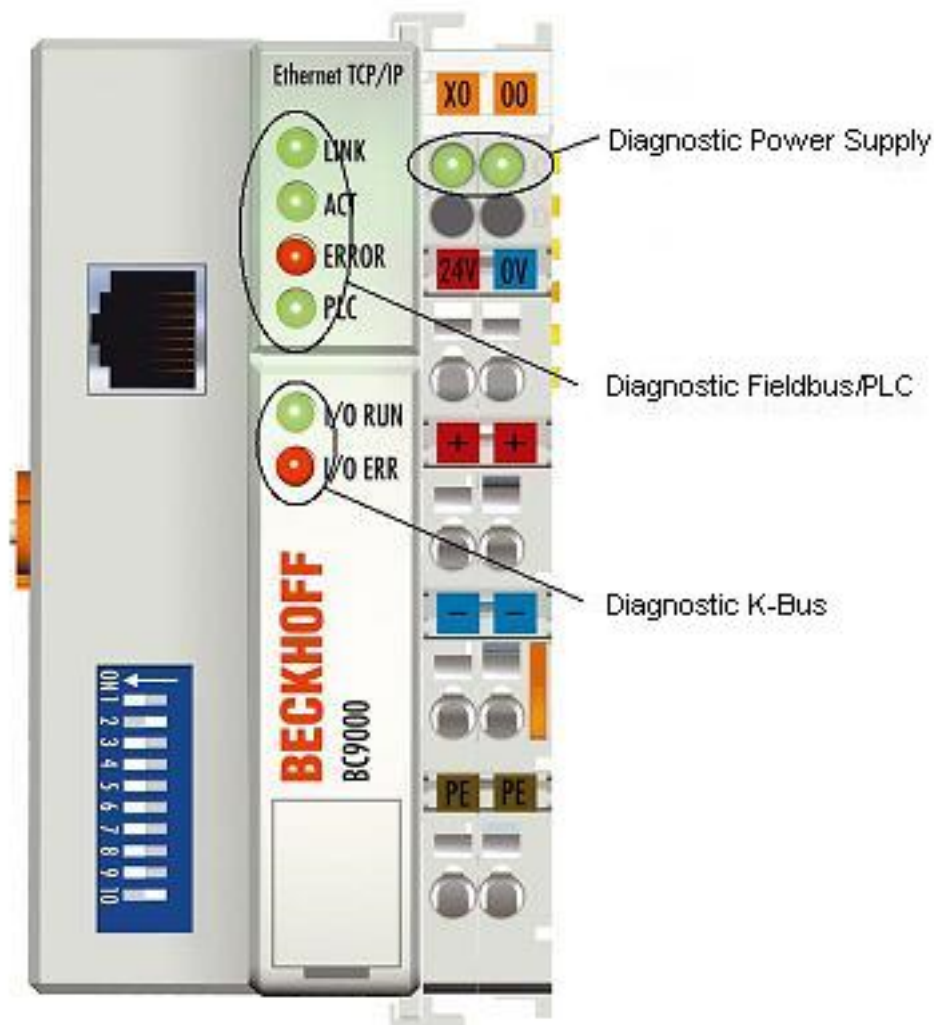


Fig. 49: BC9000 - LEDs

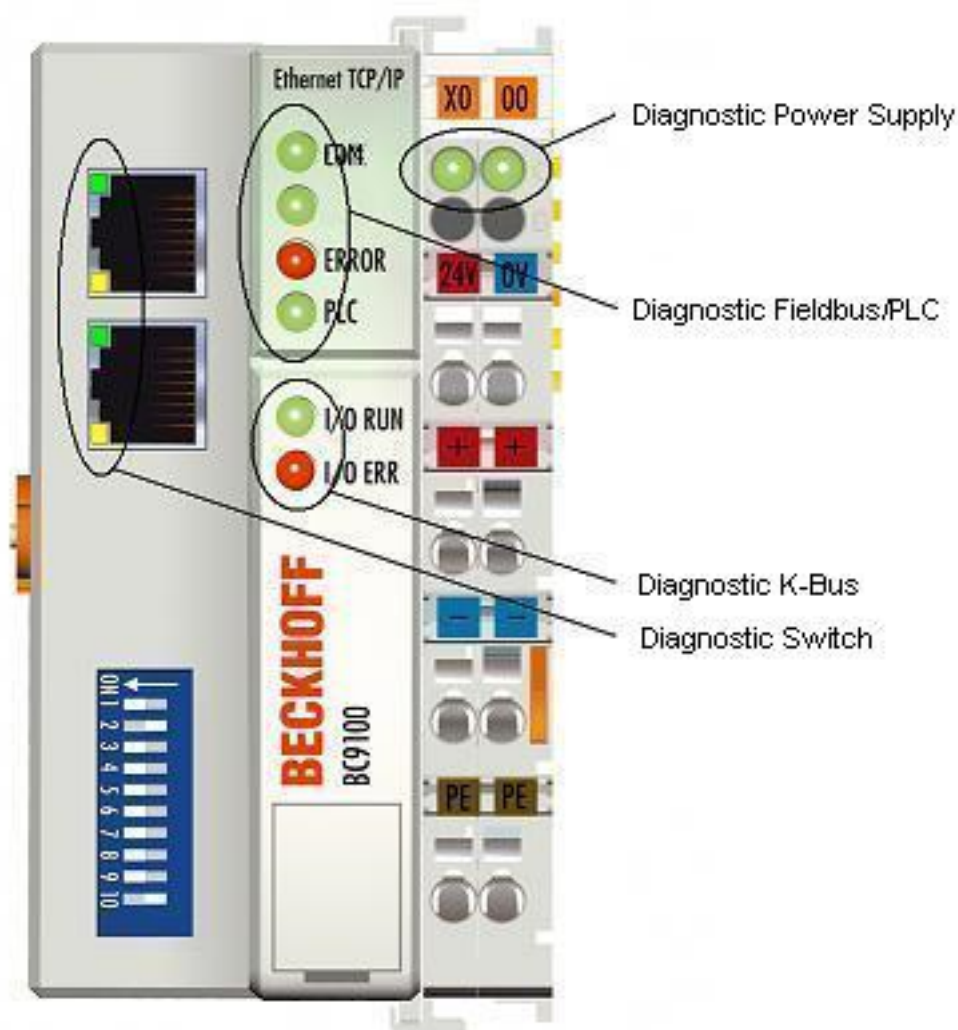


Fig. 50: BC9100 - LEDs

LEDs for power supply diagnostics

LED	Meaning
Left LED off	Bus Coupler has no power
Right LED off	No power supply 24 V _{DC} connected at the power contacts

LEDs for fieldbus diagnostics

LED	on	off
LINK (BK9000/BC9000 only)	Physical connection present	No physical connection present
ACT (BK9000/BC9000 only)	Flashing: Bus traffic present	No bus traffic (bus idle)
COM (BK9100/BC9100 only)	Flashing: data received at the controller	no data are received
ERROR	LED flashes slowly if DHCP or BootP is enabled and the Bus Coupler has not yet been allocated an IP address LED flashes quickly (5 times, only when switching on): the Bus Coupler was addressed with ARP. The settings on the DIP SWITCH are not valid. LED permanently on (only in RT-Ethernet mode) Note: task time set too fast	No error
PLC (BK9000/BC9100 only)	PLC program is in RUN mode LED flashes while the boot project is being saved.	PLC program is in stop mode
WDG (BK9000 only)	Watchdog is triggered: - No error	Watchdog expired or not triggered: - Start communication - Reset watchdog error

LEDs for K-bus diagnostics

Error code	Error code argument	Description	Remedy
Persistent, continuous flashing		EMC problems	- Check power supply for undervoltage or overvoltage peaks - Implement EMC measures - If a K-bus error is present, it can be localized by a restart of the coupler (by switching it off and then on again)
1 pulse	0	EEPROM checksum error	Enter factory settings with the KS2000 configuration software
	1	Code buffer overflow	Insert fewer Bus Terminals. Too many entries in the table for the programmed configuration
	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 initialization	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 software reset from firmware BC9000 BB and BC9100 B0: from firmware BK9000 BA and BK9100 B0: IP address already exists in the network	Hardware reset of the Bus Coupler (switch off and back on again) Assign another IP address
	4	DIP switch incorrect for BootP	Set 1-8 to <i>on</i> or <i>off</i> , see BootP ▶ 76
	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 (BC9000/BC9100 only)	0	Note: Cycle time was exceeded (see Table 1 Register 17 ▶ 76)	Warning: the set cycle time was exceeded. This note (flashing of the LED) can only be reset by re-booting the BC. Remedy: increase the cycle time
9 pulses (BC9000/BC9100 only)	0	Checksum error in Flash program	Transmit program to the BC again
	1	Incorrect or faulty library implemented	Remove the faulty library
10 pulses (BC9000/BC9100 only)	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

LEDs for switch diagnosis (BK9100/BC9100 only)

LED	on	Flashes	off
LINK/ACT	Physical connection present	Communication available	No physical connection present
10 / 100 Mbit/s	100 Mbit/s	-	10 Mbit/s

8.2 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.

Sample: >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.

* BC9000, IL230x-C900 only

8.3 ADS diagnostics

Status inputs

It is possible to monitor the BK/BC9000, B/C900 Bus Coupler's communication in the system manager. Each Bus Coupler has status inputs that can be found in the hardware tree.

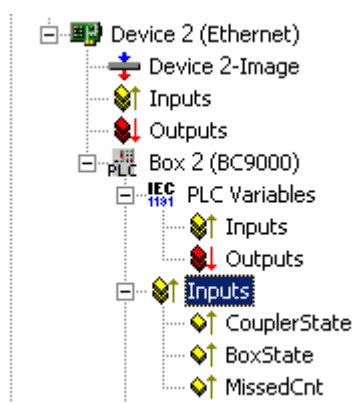


Fig. 51: Status inputs of the BC9000

Coupler state

Value	Meaning
0x0000	No error
0x0001	Bus Terminal error; there is a K-bus error
0x0002	Configuration error; the parameterized 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.

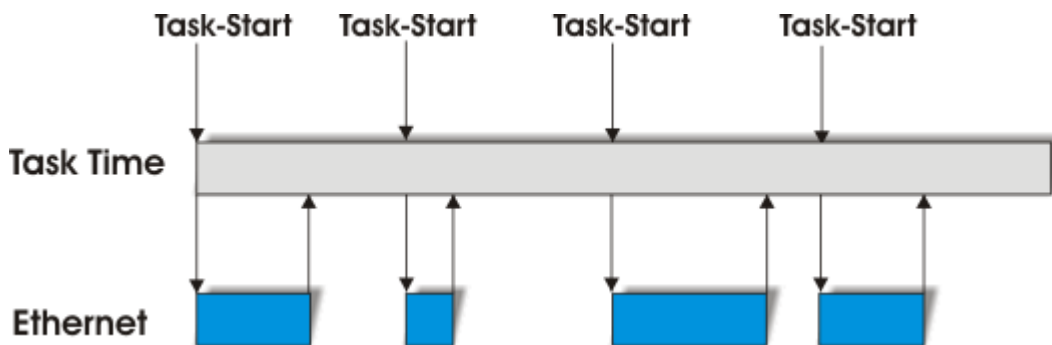


Fig. 52: Task time longer than Ethernet propagation delay

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.

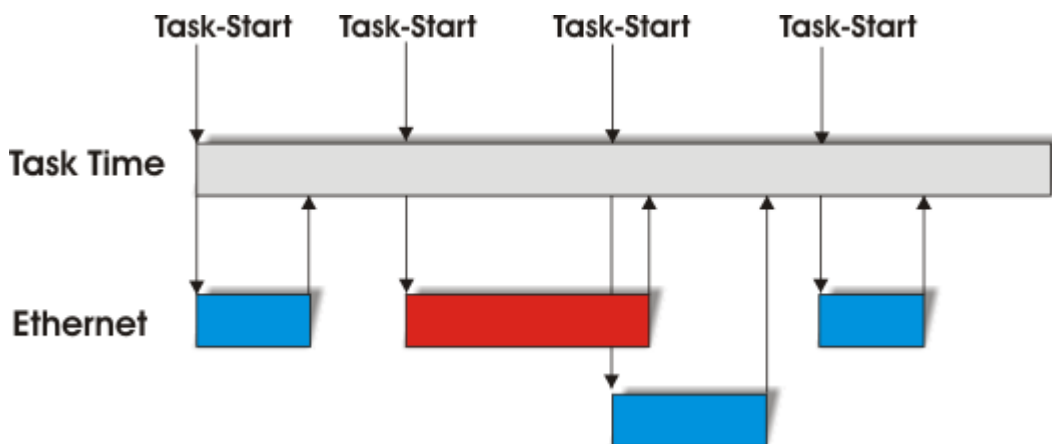


Fig. 53: Task time shorter than Ethernet propagation delay

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, C900: this consumes additional processing capacity in the 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
0xxxxz	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.

8.4 ModbusTCP diagnostic

- [See Modbus diagnostic function \[► 72\]](#)
- [See ModbusTCP error answers \[► 68\]](#)

9 Appendix

9.1 General operating conditions

The following conditions must be met in order to ensure flawless operation of the fieldbus components.

Environmental conditions

Operation

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

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

Condition	Permissible range
Permissible ambient temperature during operation	see technical data
Installation position	variable
Vibration resistance	According to EN 60068-2-6
Shock resistance	According to EN 60068-2-27
EMC resistance	According to EN 61000-6-2
Emission	According to EN 61000-6-4

Transport and storage

Condition	Permissible range
Permissible ambient temperature during storage	-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 conforms 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:

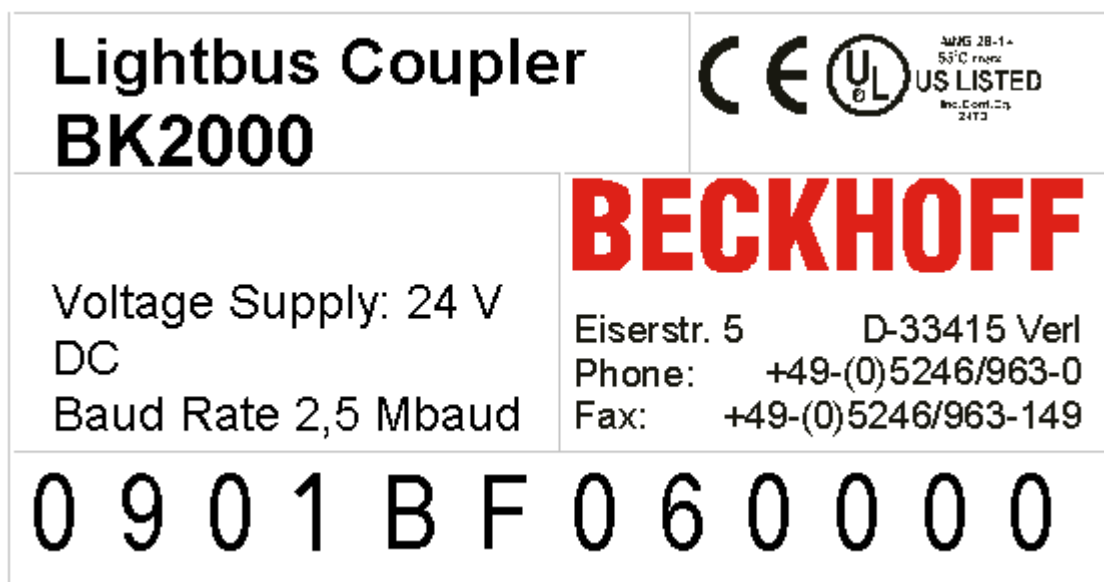


Fig. 54: Sticker with information about the BK2000 Bus Coupler certification

The following information is printed on the label:

Printed item	Meaning for this label
Precise product identification	Lightbus Coupler BK2000
Supply voltage	24 V _{DC}
Data transfer rate	2.5 Mbit/s
Manufacturer	Beckhoff Automation GmbH
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 week of production (2 characters), the year of production (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 - in the year 2001 - containing the BF firmware version - and using the 6th hardware version - with no special indications

9.2 Approvals

Underwriter laboratories

UL E172151

Conformity mark

CE

Protection class

IP20 conforms to EN60529

9.3 Test standards for device testing

EMC

EN 61000-6-2

EN 61000-6-4

Vibration resistance

EN 60068-2-6 vibration test

EN 60068-2-27/29 shock test

9.4 Bibliography

TCP/IP

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

TCP/IP (English)

Illustrated, Volume1 The Protocols

by W. Richard Stevens

Publisher: ADDISON-WESLEY Longmann Verlag

Modbus/TCP

<http://www.modicon.com/>

<http://www.modbus.org>

TwinCAT

BECKHOFF Information System

<http://infosys.beckhoff.com>

9.5 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 for Bus Terminals, Bus Couplers, Bus Terminal Controllers, fieldbus box modules, etc.

PE

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

TwinCAT

The Windows Control and Automation Technology

9.6 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's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages:

<http://www.beckhoff.com>

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

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

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
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 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

List of illustrations

Fig. 1	Principle of the Bus Terminal	9
Fig. 2	BC9000 and BC9100	11
Fig. 3	Dimensions	14
Fig. 4	Electrical isolation	17
Fig. 5	Ethernet layout in star topology	18
Fig. 6	Ethernet layout in linear topology	19
Fig. 7	RJ45 connector (Western plug)	20
Fig. 8	Ethernet connection via hub or switch	20
Fig. 9	Direct Ethernet connection (crossover cable)	20
Fig. 10	Start-up behavior of the Bus Terminal Controller	23
Fig. 11	Network classes	25
Fig. 12	BK9000, BK9100, BC9000, BC9100	26
Fig. 13	BK9050	26
Fig. 14	Configuration of the Beckhoff BootP server	28
Fig. 15	Addressing by name	29
Fig. 16	Testing the IP address using the Ping command	30
Fig. 17	Security settings	31
Fig. 18	Reading the connected Bus Couplers and Bus Terminals	32
Fig. 19	Setting the router memory	32
Fig. 20	The IP Address tab	33
Fig. 21	PLC tab	34
Fig. 22	Sample configuration of a Bus Terminal Controller	36
Fig. 23	Program running time	37
Fig. 24	Task Time	38
Fig. 25	Bus Terminals in the BC's local process image	38
Fig. 26	Assign bus terminals to the higher level controller	39
Fig. 27	Selection of the target platform	43
Fig. 28	Communication parameters	43
Fig. 29	The menu behind the TwinCAT icon	44
Fig. 30	TwinCAT system properties	44
Fig. 31	Add remote connection	45
Fig. 32	TwinCAT System Manager	45
Fig. 33	Selection of the target platform	46
Fig. 34	Choose Target System	46
Fig. 35	Function block RW_FIELDBUSPRMBC9000	48
Fig. 36	Function block RW_PARAMETERBC9000	49
Fig. 37	Function block FB_SNTP	50
Fig. 38	User Datagram Protocol (UDP)	53
Fig. 39	Protocols running on top of TCP/IP and UDP/IP	53
Fig. 40	The ADS protocol as a transport layer within the TwinCAT system	55
Fig. 41	Structure of the ADS protocol	56
Fig. 42	Displaying the AMS routing table using the KS2000 configuration software	61
Fig. 43	ADS connection	62
Fig. 44	Bus structure for cross communication	63

Fig. 45	Cross communication from one PC to two BC9000s.....	63
Fig. 46	Command >ipconfig< in the DOS window	64
Fig. 47	Display of the AMS Net ID	64
Fig. 48	ModbusTCP Protocol.....	65
Fig. 49	BC9000 - LEDs	79
Fig. 50	BC9100 - LEDs	80
Fig. 51	Status inputs of the BC9000	82
Fig. 52	Task time longer than Ethernet propagation delay	83
Fig. 53	Task time shorter than Ethernet propagation delay	83
Fig. 54	Sticker with information about the BK2000 Bus Coupler certification.....	86