Absolute Rotary Encoder
with Modbus/TCP Interface

User manual

Scancon A/S, Tranevang 1, DK-3450 Alleroed, Denmark
Tlf: +45 48172702 Fax: +45 48172284
www.scancon.dk , info@scancon.dk

Imprint
POSITAL GmbH
Carlswerkstrasse 13c
51063 Köln
Phone +49/221/96213-0

Internet www.posital.com
Fax +49/221/96213-20

e-mail info@fraba.com

Alteration of Specifications reserved
Technical specifications, which are described in this manual, are subject to change due to our permanent strive to improve our products.

Document information

File name:	UME-OCD-EM
Date:	February 2008
Version number:	1.2
Author:	Reiner Bätjer

Service-Phone
For technical support, questions and suggestions for improving our products and documentations call our telephone line: +49/221/96213-0

1 Introduction

1.1 Absolute Rotary Encoders

Absolute rotary encoders provide a definite value for every possible rotary position. All these values are reflected on one or more code discs. The beams of infrared LEDs are sent through the code discs and detected by Opto-Arrays. The output signals are electronically amplified and the resulting value is transferred to the interface.

The absolute rotary encoder has a maximum resolution of 65,536 steps per revolution (16 Bit). The Multi-Turn version can detect up to 16,384 revolutions (14 Bit). Therefore the largest resulting resolution is 30 Bit = 2^{30} = 1,073,741,824 steps. The standard Single-Turn version has 13 Bit, the standard Multi-Turn version 25 Bit.

The encoder sends the data in binary code via standard or fast Ethernet (10 Base T, 100 Base T). At present it supports the following international standardized protocols: TCP, IP (http and SMTP in version A1).

The encoder is able to provide three different kinds of output data: the position value, a velocity value and a time stamp. These can be use in arbitrary combinations for TCP transmitting.

The following functions of the absolute rotary encoder can be programmed directly via the Ethernet connection:

- Used scope of physical resolution
- Total scaled resolution
- Preset value
- Code sequence (Complement)

There is no specific software required for version A1 to initiate and use the absolute rotary encoder because the sensor can be read out and programmed by any standard web browser. For this purpose the absolute rotary encoder contains a web server, which provides HTML documents with embedded Java applets. These documents are a widely self-explanatory graphical user interface (GUI) that is described in detail in chapter 4.2. The automated data transfer with a control system is done with TCP/IP by simple plain text commands and data in ASCII format.

The encoder supports the communication with Modbus/TCP-PLC's and –IPC's. With function code 03 can you read out data. Function code 16 allow to set the parameters. More details see in chapter 3.4.

1.2 Ethernet

The present developments in the field of Industrial Ethernet are based on the vision of an integrated access of all data of a company through a uniform communication system. In higher levels of enterprise communication Ethernet is the main medium of data transfers. Combined with other IT technologies it is internationally standardized. In the long run automation engineers will benefit from the rapid technological progress in the mass markets of IT and web technologies.

Ethernet technically provides a system with higher data transfer rates than common field bus systems. TCP/IP and UDP do have a statistical access method to access the medium thereby prohibiting determined response times. Many developments are intensely done on additional real time mechanisms, e.g. Ethernet Powerlink, Ethernet/IP, Profinet or EtherCat. However, you can already get access times that are sufficient for many applications when using TCP/IP or UDP. If you directly connect the absolute encoder to a computer via a 100 Mbit network card, you will get a cycle time of less than 2 ms. In huge networks the cycle times will depend on the utilization of the network.

1.3 TCP/IP

Even though Ethernet and TCP/IP are often used together and sometimes used interchanged, these are three different kinds of terms and you should carefully separate them. The coherences are based on the ISO/OSI reference model after ISO/IEC 7498 that is needed to basically understand these terms.

Ethernet only describes layer 1 and 2 in this model, nevertheless the term is often used in error in engineering as description of all layers between 1 and 7.

The IP protocol of layer 3 was developed in the 70's by the US military (MIL-STD 1777). It allows a universal addressing independent of the hardware involved in heterogeneous networks. It also manages the transfer of large packets by splitting them up into smaller packets. The well-known TCP protocol (MIL-STD 1778) ensures a reliable data transfer.

Http (RFC 2068) and SMTP (MIL-STD 1781) belong to layer 7 of the OSI model and allow to transfer data and documents via web browser or to send e-mails.

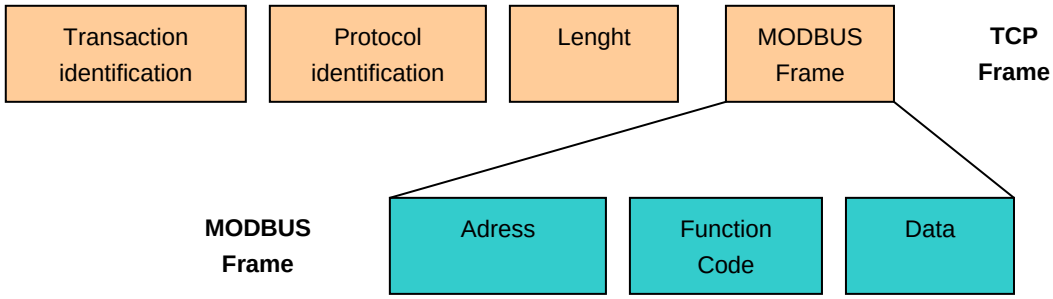
1.4 Modbus/TCP

MODBUS is an application layer messaging protocol, positioned on level 7 of the OSI model, that provides client/server communication between devices connected on different types of buses or networks.
As an industry's standard since 1979, MODBUS continues to enable millions of automation devices to communicate. Today, support for the simple and elegant structure of MODBUS continues to grow. The Internet community can access MODBUS at a reserved system port 502 on the TCP/IP stack.
MODBUS is a request/reply protocol and offers services specified by function codes.
MODBUS function codes are elements of MODBUS request/reply PDUs. The objective of this document is to describe the function codes used within the framework of MODBUS transactions.
MODBUS is an application layer messaging protocol for clients.
For more information's see www.modbus.org.

1.5 OSI-Modell

Layer			
7	Application Layer	Modbus	Application
6	Mapping Layer	Modbus <-> TCP	
5			
4	Transport Layer	TCP	Data transport
3	Network Layer	IP	
2	Ethernet Mac Layer	IEEE 802.3	
1	Physical Layer		Cable

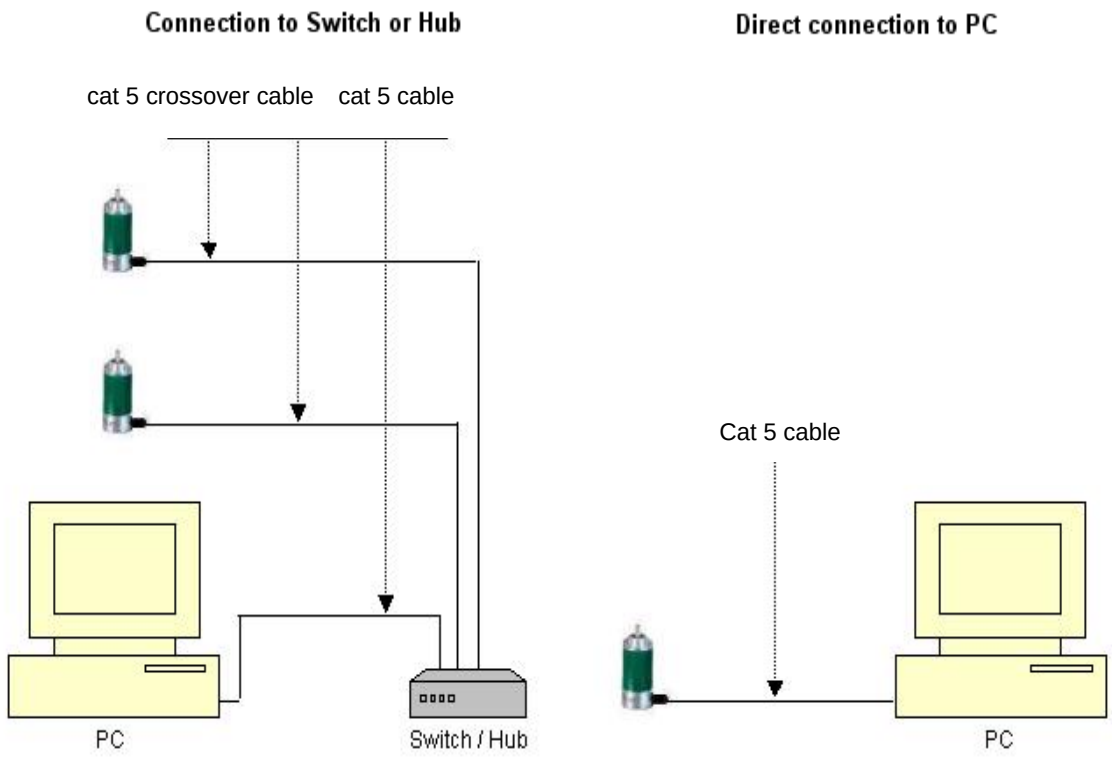
1.6 MODBUS frame



2 Hardware set-up and Ethernet Connection

2.1 Network Topology

Using Ethernet there are different kinds of topologies possible. The connection of the encoder can be made both directly to the computer with a network card or indirectly with a switch, hub or company network, see figure below. If you use a direct connection to a computer without network components in between, you need to use a standard, “straight” network cable (not a crossover cable). You need at least a cable of category 5 to get a data transfer rate up to 100 Mbit. If there is a network component in the network, which does not provide Fast Ethernet, the sensor will automatically switch down to 10 Mbit.



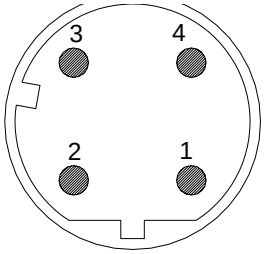
2.2 Connecting an Absolute Encoder

The encoder is connected by a 5 pin M12 connector for the power supply and one 4 pin, D-coded M12 connector for Ethernet.

Connector Ethernet
4 pin female, D-coded

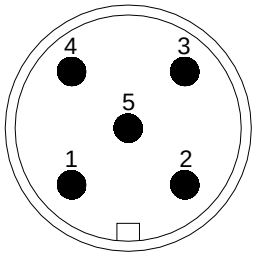
Pin Number	Signal
1	Tx +
2	Rx +
3	Tx -
4	Rx -

Sketch on encoder view



Connector power supply
5 pin male, A-coded

Pin Number	Signal
1	+24 V
2	+24 V
3	0 V
4	0 V
5	PE



2.3 Ethernet Cables

2.3.1 RJ45 – M12 crossed

Signal	RJ45 Pin	M12 Pin	Signal
Tx+	1	2	Rx+
Tx-	2	4	Rx-
Rx+	3	1	Tx+
Rx-	6	3	Tx-

2.3.2 RJ45 – M12 straight

Signal	RJ45 Pin	M12 Pin	Signal
Tx+	3	1	Tx+
Tx-	6	3	Tx-
Rx+	1	2	Rx+
Rx-	2	4	Rx-

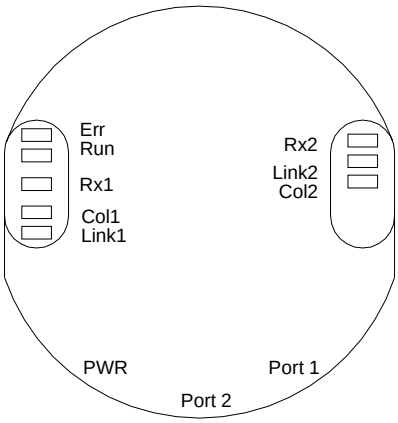
2.3.3 M12 – M12 crossed

Signal	M12 Pin	M12 Pin	Signal
Tx+	1	2	Rx+
Tx-	3	4	Rx-
Rx+	2	1	Tx+
Rx-	4	3	Tx-

2.4 Diagnostic LED's

LED	Color	Description for LED = on
Rx1	Yellow	Incoming and outgoing traffic for port 1
Link1	Green	Link to another Ethernet component for port 1
Collosion1 *	Red	Ethernet collisions on the bus for port 1
Rx2 *	Yellow	Incoming and outgoing traffic for port 2
Link2 *	Green	Link to another Ethernet component for port 2
Collosion2 *	Red	Ethernet collisions on the bus for port 2
Error *	Red	-
Run *	Green	-

* Not available



3 Data transmission

3.1 Values

Position values, velocity and a time stamp are provided.

3.2 Format

	Data type	Sign
Position	32 bit integer	unsigned
Velocity	32 bit integer	signed
Time stamp	64 bit integer	unsigned

3.3 Function code 03

03 (0x03) Read Holding Registers

This function code is used to read the contents of a contiguous block of holding registers in a remote device. The Request PDU specifies the starting register address and the number of registers. In the PDU Registers are addressed starting at zero. Therefore registers numbered i.e. 1-8 are addressed as 0-7.

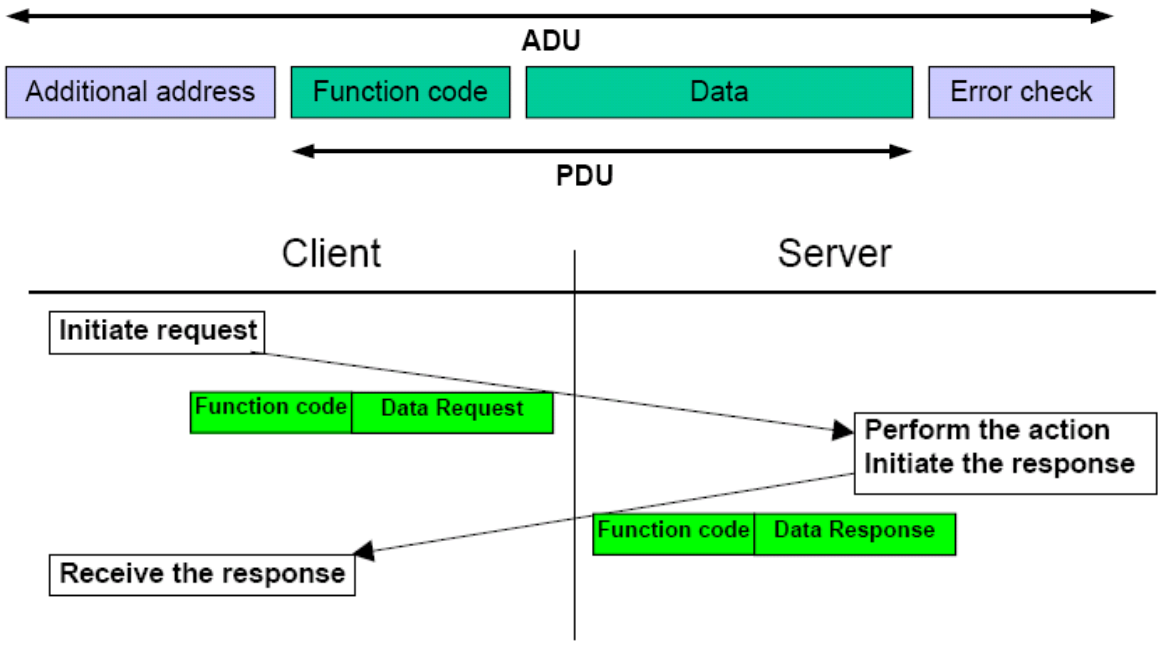
The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits. The Error check in ADU is for Modbus/TCP not available, because TCP use a Error check. For details see www.modbus.org.

3.4 Function code 16

16 (0x10) Write Multiple registers

This function code is used to write a block of contiguous registers in a remote device. The requested written values are specified in the request data field. Data is packed as two bytes per register. The normal response returns the function code, starting address, and quantity of registers written.

Please take care under all circum-stances that the encoder is not turned off while it is writing to the flash !



3.5 Modbus Mapping

Startadress: 0000
Number of registers: 8

Register	Data type	Data
0	Position	Bit 17-32
1	“	Bit 1-16
2	Velocity	Bit 17-32
3	“	Bit 1-16
4	Time stamp	Bit 49-64
5	“	Bit 33-48
6	“	Bit 17-32
7	“	Bit 1-16
8	Not in use	-
9	Not in use	-
10	UsedScopeOfPhysRes	Bit 17-32
11	“	Bit 1-16
12	TotalScaledRes	Bit 17-32
13	“	Bit 1-16
14	Preset	Bit 17-32
15	“	Bit 1-16
16	Offset	Bit 17-32
17	“	Bit 1-16
18	CountingDir CW = 0 CCW = 1	Bit 1-16
19	“	Bit 17-32

Register 10 to 18 are only in use to send the parameters to the encoder.

Notify:

- The write registers will not get a update with changed parameters from the Web applet or TCP commands.
- The velocity value can be wrong during setting some parameters

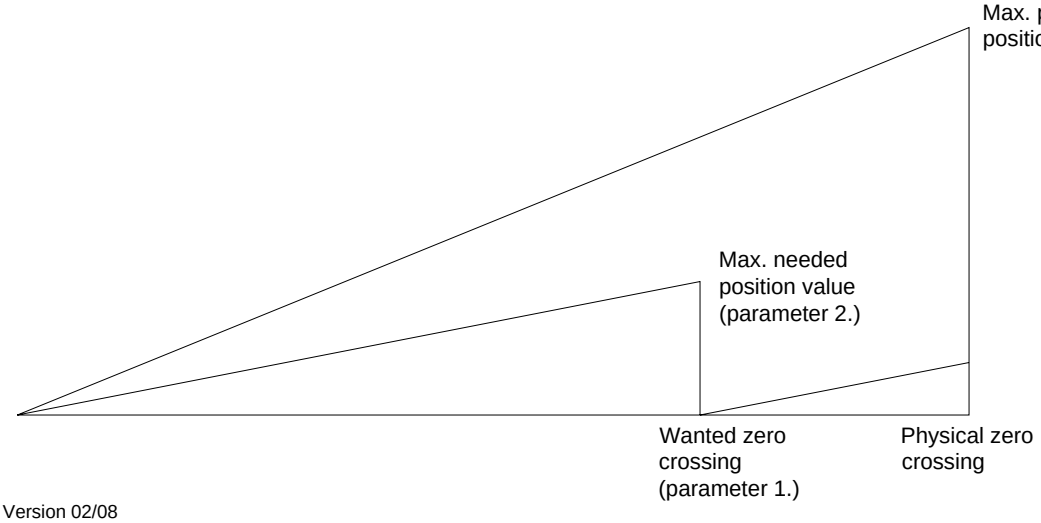
4 Programming

4.1 Programming of Parameters

The encoder is able to provide three different kinds of output data: the position value, a velocity value and a time stamp. These can be used in arbitrary combinations.

Parameter	Description
Used scope of physical resolution (parameter 1.)	Specifies the part of the physical resolution used for the encoder in physical steps. If e.g. for an encoder with a resolution of 8192 steps per revolution 16384 is chosen, the encoder will count 8192 steps per revolution (if “Total scaled resolution” is set to the same value as “Used scope of physical resolution”) and start with zero again after 2 revolutions. If this value is not set to a value which results in an integer division with the total physical resolution, the encoder value will jump to zero when passing the physical zero point.
Total scaled resolution (parameter 2.)	Specifies the scaled resolution which is used over the area of physical steps defined by “Used scope of physical resolution”. If e.g. the encoder is set as described above and “Total scaled resolution” is set to 10, the encoder will count 10 steps over the physical steps defined with “Used scope of physical resolution”, i.e. 5 steps per revolution.
Code sequence	The code sequence (complement) can be programmed as an operating parameter. This parameter determines whether the output code increases or decreases when the axis is turned clockwise.
Preset value	The preset value is the desired output value for the actual position of the axis. The actual output value will be set to this preset value.
Offset value	The offset value can set the offset to physical position of the axis.

The html page, the programmable parameters, and the diagnostics of the encoder are described in the next chapter.

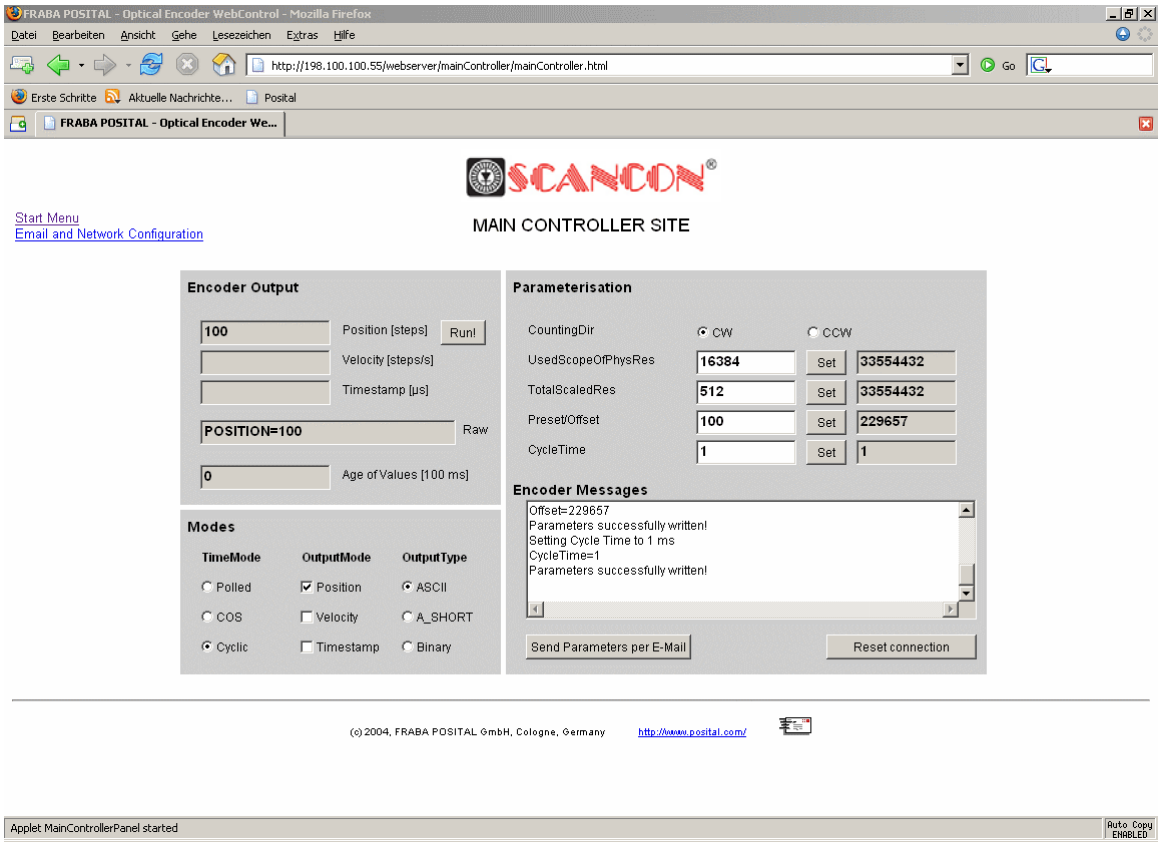


4.2 Operating by the integrated Web Server

The absolute rotary encoder can be addressed by any web browser (e.g. Netscape, Internet Explorer, Opera, etc.). Please enter the IP address of the encoder in the address field of the browser. The factory setting for the IP address is 10.10.10.10. Chapter 4.3 will deal with changing the IP address.

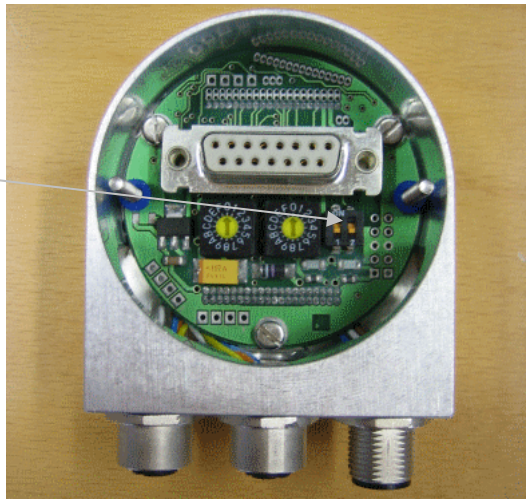
If the encoder has built up a connection to the browser, you can see its start page. To be able to parameterize the encoder
please open the page “Main Controller Site“ (see image below). The other links on the starting page, will open a html page showing all available commands („Information about Commands“) or the page to configure the network settings. Chapter 5 describes these commands in more detail.

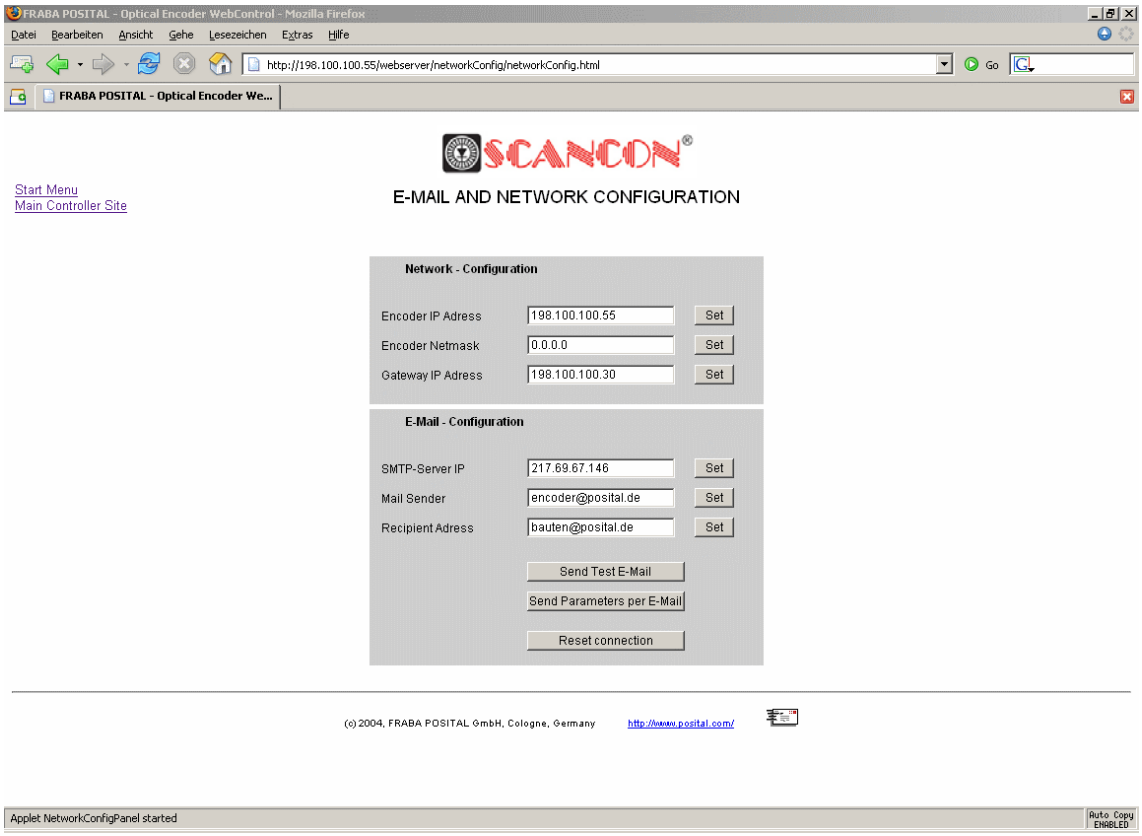
To read, for example, the position value continuously please set the desired cycle time and choose the cyclic mode. Each command to the encoder and messages from the encoder are logged in the encoder message window.



4.3 E-mail and Network Configuration

The rotary encoder can be used either with the wired IP 10.10.10.10 or the software IP address which can be programmed. A switch to choose either option is located in the connection cap. If the switch 2 is in position “off”, the programmable IP has been chosen. Both Hex rotary switches and switch 1 are not in use for this encoder. The configuration window can be accessed via the “Main Controller Site” or the start page.

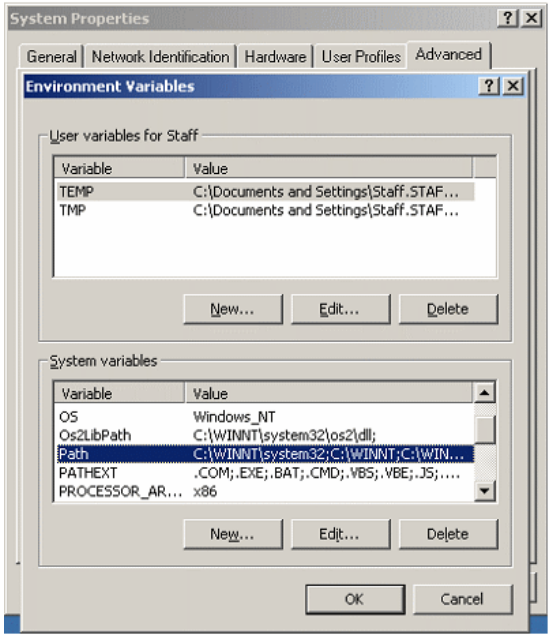




5 Operating by TCP/IP Commands

5.1 Introduction

To use the absolute encoder with a control system platform independent commands and data in ASCII format can be exchanged by TCP/IP. To take a look at the commands and a short description please see chapter 5.6. To find out how to address the TCP/IP interface of your control or operating system please refer to the documentation for these devices



If you use a Windows PC, you can try the following connection to the sensor: Go to the command prompt (DOS) and type in “ping <computer-name>” or “ipconfig”. In response you get the IP address of your computer. If the encoder IP address is not located within your subnet mask, you will need to prepare the data transfer to the encoder by entering the command “route add <IP-sensor> <IP-computer>“. Maybe are administrator rights necessary. Otherwise your PC/control system will try to reach the encoder via your computers standard gateway. The default IP address of the sensor is 10.10.10.10. You can check the connection to the sensor with the command “ping <IP-sensor>“.

5.2 Installation

To communicate with the Encoder using our example tools tcpcl or updcl, a Java runtime environment is required on your PC. If you have not installed Java, you can get it from our CD (look under the section “accessories”). You can also download the latest version from http://java.sun.com/products/j2se. Copy the FRABA-Java programs which you can find on web site http://www.posital.com/de/products/POSITAL/AbsoluteEncoders/AbsoluteEncoders_OCD_IndustrialEthernet_TCP_IP_base.html onto your hard disk, e.g. in the folder c:\fraba\ethernet.
Afterwards you need to set up the PATH variable for the Java installation and the FRABA-Java programs. For a convenient start we also provided batch files to start the java files, depending on the IP addresses you might need to modify them. For TCP will be used port 6000.

5.3 PATH Variable

5.3.1 MS-DOS, Win95, Win98, WinME

Please add the required paths to c:\Autoexec.bat behind the “Path” line. Example:
Path=c:\ms-dos; c:\Program Files\BC\BIN
Path=%Path%;c:\fraba\ethernet\
Path=%Path%;c:\programme\java\bin

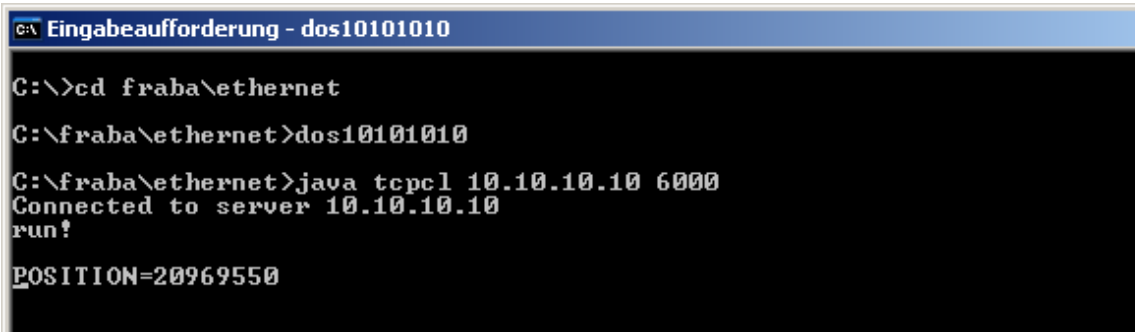
5.3.2 WinNT3.51, WinNT4, Win2000, WinXP

In Start – Settings – Control panel – System – Advanced – Environment Variables you can configure the variable “Path”. Please **do not change** the **other path settings**, but only add the required paths! Depending on the operating system used administrator rights might be necessary.

5.4 Operating

After starting the batch file TCP_10101010.bat the connection to the encoder will be built up. Once you are connected, you can try e.g. “read offset” (please note space) to read out the calculated offset from the encoder. You can see all available commands in the next chapter.

If the encoder is running in cyclic mode, you can see position values coming continuously from the encoder. You can enter a command anyway, although your input will be overwritten by new position values, the command will still be sent once you press enter.
The Java program can be finished with CTRL-C.



5.5 Advanced functionality

In the subdirectory "advanced" in the Zip-file “Software Tools” there is a version of the TCP-client with enhanced functionality:

- the time from the command till the encoder issues an answer can be measured in steps of 10 ms. This can be switched on/off using time / notime.
- the binary values transmitted by the encoder can be transferred to ASCII again, if it does not contain ‘\0’ or ‘\n’. This can be switched by binary / ASCII, it will be automatically switched when the encoder is switched from/to binary mode.
- Scrolling of the output can be turned on/off via scroll / noscroll
- 'new' renews the connection to the encoder
- 'exit' will close the TCPClient application

5.6 Parameters

5.6.1 Commands

Important: Please note spaces, upper and lower case! **<Value>** means the parameter to enter. All commands and parameters have to be entered in one line and started with **<ENTER>**.

“Value“ means the output value. You can change and read the settings of the encoder by using the following commands:

Commands	Remarks
Run!	This command will order the encoder to send a position value, regardless of the time mode.
set <Variable>=<Value>	This command will set a variable to a given value. If successful, the encoder will answer in the form <Variable>=<Value>, else an error message will be issued. All variables/modes are stored in the internal flash a few seconds after they were set. After the value was saved, the message "Parameters successfully written!" is issued to all connected TCP-Clients. If the encoder is turned off while writing to the flash, the process can damage the flash and destroy the encoder program. Please take care under all circumstances that the encoder is not turned off while it is writing to the flash !
read <Variable>	This command will read out a variable from the encoder. The encoder will answer in the form <Variable>=<Value>.

5.6.2 Variables

Variables	Remarks / Values
UsedScopeOfPhysRes	Specifies the part of the physical resolution used for the encoder in physical steps. If e.g. for an encoder with a resolution of 8192 steps per revolution 16384 is chosen, the encoder will count 8192 steps per revolution (if TotalScaledRes is set to the same value as UsedScopeOfPhysRes) and start with zero again after 2 revolutions. If this value is not set to a value which results in an integer division with the total physical resolution, the encoder value will jump to zero when passing the physical zero point. Default value: Physical resolution of the type label. I.e. 4096 resolutions x 8192 steps per revolution = 33,554,432
TotalScaledRes	Specifies the scaled resolution which is used over the area of physical steps defined by UsedScopeOfPhysRes. If e.g. the encoder is set as described above and TotalScaledRes is set to 10, the encoder will count 10 steps over the physical steps defined with UsedScopeOfPhysRes, i.e. 5 steps per revolution. Default value: Physical resolution of the type label. I.e. 4096 resolutions x 8192 steps per revolution = 33,554,432
CountingDir	Specifies the direction to turn the axis which is associated with higher values. • **CW**: denotes that clockwise turning will increase the position value • **CCW**: denotes that counterclockwise turning will increase the position value
Preset	When the preset is set, an internal offset will be calculated, which will be saved and added to all position values afterwards. The value given for the preset denotes the position value the encoder will show at the point where the preset was set.
Offset	This variable makes it possible to directly change the offset calculated and set by the preset function.
TimeMode	Possible time modes are: • **polled**: Encoder will only send output values if asked to do by "Run!" • **cyclic**: Encoder will send output values after time specified by CycleTime. • **change of state**: The Encoder will send the output values only if either the position or the velocity has changed. The values are checked every 5 ms to reduce unwanted network traffic

Variables	Remarks / Values
OutputMode	Possible output modes are: [Position_][Velocity_][Timestamp_] where the components mean: • **Position**: Encoder will send a scaled Position value. • **Velocity**: Encoder will send a velocity Value (steps/s). • **Timestamp**: Encoder will send a timestamp in microseconds, starting with 0 at the startup of the encoder. As the counter is a 32 Bit value, the timestamp will reach zero again after approx. 1.2 hours. This variable has got no effect to the Modbus communication.
OutputType	Possible output types are: • **ASCII**: Encoder will send ASCII-letters in the form "POSITION=<POSITION> VELOCITY=<VELOCITY> TIMESTAMP=<TIME>" • **ASCII_SHORT**: Encoder will send ASCII-numbers in the form "<POSITION> <VELOCITY> <TIME>", separated by spaces • **BINARY**: Encoder will send 32 bit binary values without any separator between the values. This variable has got no effect to the Modbus communication.
CycleTime	States the time in ms for the cyclic time mode. Can have values between 1 ms and 999,999 ms. This variable has got no effect to the Modbus communication.
IP	Sets the IP-address of the encoder and must be a valid IP-address in the form a.b.c.d, with a, b, c, d from 0 to 255. Attention: The IP-address will only be activated after a new power-up when switch 2 is in position “off”.
NetMask	The net mask used by the encoder. Please take care that Encoder and PLC/PC are within the same subnet or specify a working gateway.
Gateway	Gateway to be used by the encoder, if own IP-address and destination IP-address are not within the same subnet specified by the net mask.
OwnEmailAddr	The email-address given as the sender in emails from the encoder.
RmtEmailAddr	The email address emails will be send to.
SMTPServerIP	The IP-address of the SMTP-server which the encoder will send the email by.
Verbose	Level of information output for tracer (0 = only errors, 1 = errors and warnings, 2 = errors, warnings and clues)

5.6.3 Encoder answers

Encoder answers	Remarks
<Variable>=<Value>	If a variable was correctly set, the encoder will answer to all connected TCP-clients with the variable and its new value. This indicates that the Encoder understood the command and now uses the value, it does not indicate that the value was already save to the internal Flash, please allow some additional seconds for that.
ERROR: ...	If something went wrong, the encoder will issue an error, e.g. if it did not understand a command or if a value for a variable was not correct. It will describe the error after the "ERROR:" tag.
WARNING: ...	If a variable was set to a value, which is permitted, but which may result in problems when certain conditions occur, the encoder will issue a warning. This could for example happen, if the variable UsedScopeOfPhysRes is set to a value which does not result in an integer division with the physical resolution of the encoder when dividing the total physical resolution of the encoder. The reason for the warning will be sent following the "WARNING:" tag.
Parameters successfully written!	If any variable was set, it is important to wait until the encoder displays this message before the encoder can be turned off, otherwise the internal flash might be damaged.

6 Technical Data

6.1 Electrical Data

Supply voltage	10 - 30 V DC (absolute limits)
Power consumption	max. 4 Watt
EMC	Emitted interference: EN 61000-6-4
	Noise immunity: EN 61000-6-2
Bus connection	Ethernet
Transmission rate	10/100 MBit
Accuracy of division	± ½ LSB (up to 12 Bit), ± 2 LSB (up to16 Bit)
Step frequency LSB	Max. 800kHz (valid code)
Response time	> 2 ms for MODBUS/TCP
Electrical lifetime	> 10^5 h
Device addressing	Programmable IP-Address and Network parameters

6.2 Mechanical Data

Housing	Aluminum, optional stainless steel
Lifetime	Dependent on shaft version and shaft loading – refer to table
Max. shaft loading	Axial 40 N, radial 110 N
Inertia of rotor	≤ 30 gcm^2
Friction torque	≤ 3 Ncm (without shaft sealing)
RPM (continuous operation)	max. 12,000 RPM
Shock (EN 60068-2-27)	≤ 30 g (half sine, 11 ms)
Permanent shock (EN 60028-2-29)	≤ 10 g (half sine, 16 ms)
Vibration (EN 60068-2-6)	≤ 10 g (10 Hz ... 1,000 Hz)
Weight (standard version)	Singleturn: ≈ 500 g
	Multiturn: ≈ 700 g

Flange	**Synchro (S)**		**Clamp (C)**	**Hollow shaft (B)**
Shaft diameter	6 mm	10 mm	10 mm	15 mm
Shaft length	10 mm	20mm	20 mm	-
hollow shaft depth min. / max.	-	-	-	15 mm / 30 mm

6.3 Minimum (mechanical) lifetime

Flange	Lifetime in 10^8 revolutions with F_a / F_r		
	40 N / 60 N	40 N / 80 N	40 N / 110 N
C10 (Clamp flange 10 x 20)	247	104	40
S10 (Synchro flange 10 x 20)	262	110	42
S6 (Synchro flange 6 x 10) without shaft sealing	822	347	133

S6 (Synchro flange 6 x 10) with shaft sealing: max. 20 N axial, 80 N radial

6.4 Environmental Conditions

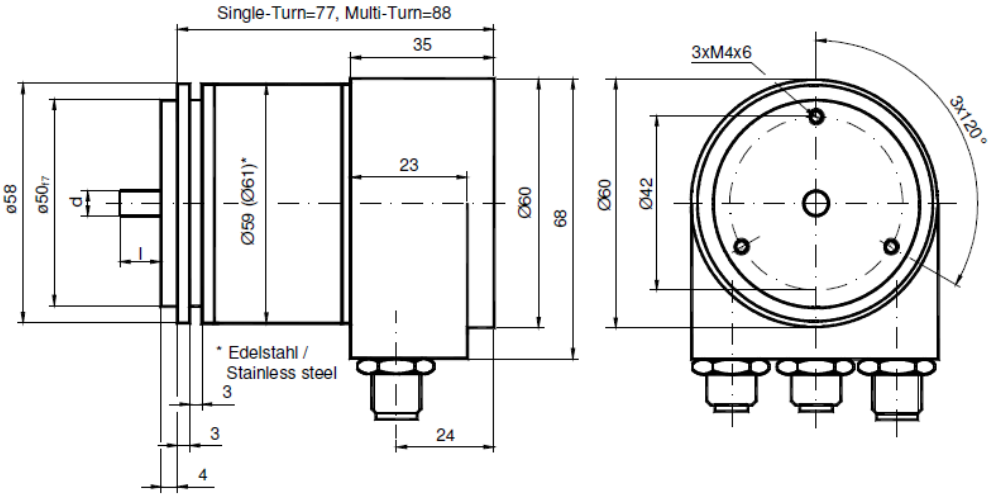
Operating temperature	0 .. +60°C
Storage temperature	- 40 .. + 85 °C
Humidity	98 % (without liquid state)
Protection class (EN 60529)	Casing side: IP 65
	Shaft side: IP 64 (optional with shaft sealing: IP66)

7 Mechanical Drawings

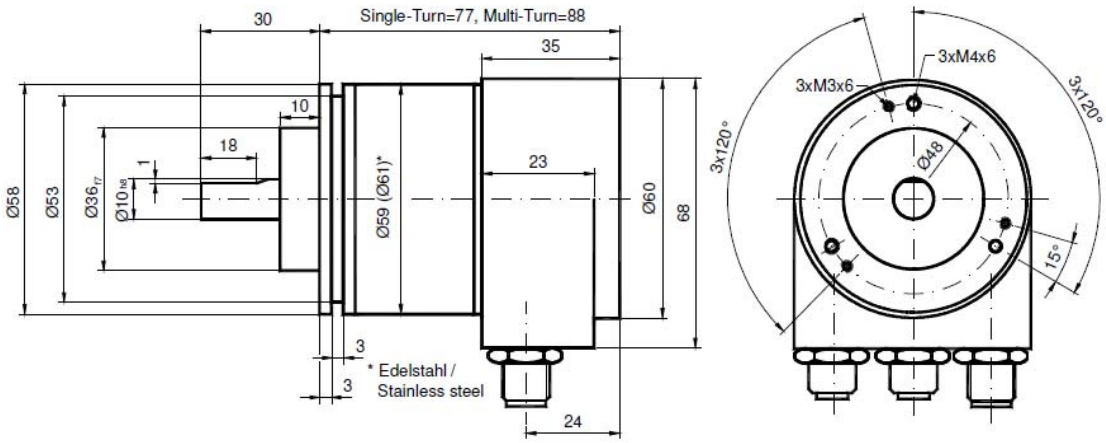
7.1 Synchro Flange (S)

available in 2 versions

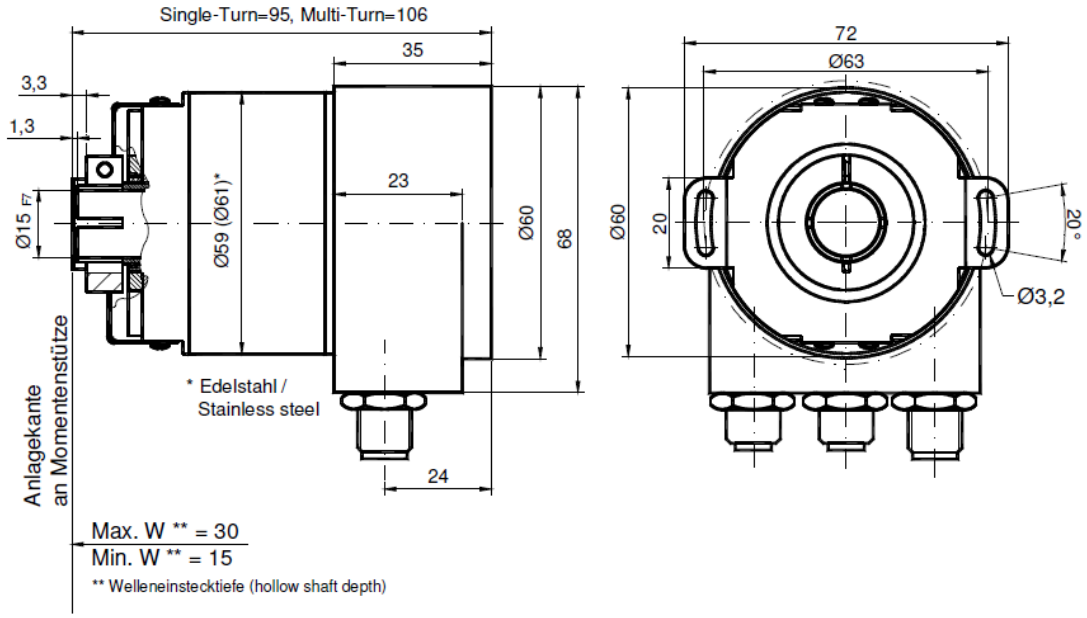
Synchro flange	d / mm	l / mm
Version S06	6_{f6}	10
Version S10	10_{h8}	20



7.2 Clamp Flange (F)



7.3 Hollow shaft (B)



Mounting instructions

The clamp ring should only be tightened after the shaft of the driving element was inserted into the hollow shaft.

The diameter of the hollow shaft can be reduced to 14mm, 12 mm, 11 mm, 10 mm or 8 mm by using an adapter (this reducing adapter can be pushed into the hollow shaft).

Maximum radial and axial misalignment of the drive shaft::

	axial	radial
static	± 0.3 mm	± 0.5 mm
dynamic	± 0.1 mm	± 0.2 mm

8 Models / Ordering Description

Description	Type Key								
Optocode	**SAG-**	**EM**	_ _	B -	_ _	_ _ -	_ _ _	_ -	**PRM**
Interface	Ethernet	**EM**							
Version	2xM12		**00**						
Code	Binary			**B**					
Revolutions (Bits)	Singleturn				**00**				
	Multiturn (4,096 revolutions)				**12**				
	Multiturn (16,384 revolutions)				14				
Steps per	8,192					**13**			
revolution	65,536					16			
Flange /	Clamp flange, full shaft:			Ø 10 mm			**C10**		
Shaft diameter	Synchro flange, full shaft:			Ø 6 mm			**S06**		
				Ø 10 mm			S10		
	Blind hollow shaft, hollow shaft :			Ø 15 mm			**B15**		
Mechanical	Without							**0**	
options	Shaft sealing (IP66)							S	
	Customized							C	
Connection	M12 connector								**PRM**

Standard = bold, further models on request

9 Accessories and Documentation

Description		Type
Male cable connector	M12, 4 pin, D-Coded	PAM4
Female cable connector	M12, 5 pin	PAM5
Coupling *	Drilling: Ø 10 mm	GS 10
	Drilling: Ø 6 mm	GS 06
Clamp disc *	Set = 4 pcs.	SP 15
Clamp half-ring *	Set = 2 pcs.	SP H
Reducing adapter **	15 mm to 14 mm	RR14
	15 mm to 12 mm	RR12
	15 mm to 11 mm	RR11
	15 mm to 10 mm	RR10
	15 mm to 8 mm	RR8
User manual *	Installation / configuration manual, English	UME-OCD-EM00
User manual *	Installation / configuration manual, German	UMD-OCD-EM00

* These can be downloaded free of charge from our homepage www.scancon.dk

** usable only for full shaft

*** usable only for hollow shaft, in stainless steel available too

We do not assume responsibility for technical inaccuracies or omissions. Specifications are subject to change without notice.

10 Glossary

Term	**Explanation**
10 Base T	Transmission line with 10 Mbit data transmission rate
100 Base T	Transmission line with 100 Mbit data transmission rate
ADU	**A**pplication **D**ata **U**nit
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange ASCII describes as code the correlation from digital integers to a normal font described character.
Batch file	Script program for MS-DOS
Baudrate	Transmission rate; it display the transmission bits per second
Binary	Numeric system with value 0 or 1.
Browser	Software program to display HTML-Sides on different operating systems (Linux, Unix, Windows, ...)
CAT5	Terminations for transmission rates up to 100 Mbit.
CRC	The cyclic redundancy check is a method from the information technology to control a checksum for data, to reduce errors by the transmission.
EMC	**E**lectro**m**agnetic **c**ompatibility, there are rules to verifying devices.
Ethernet	Ethernet is a computer network technology based on frames.

Term	Explanation
Fast Ethernet	Transmission technology with 100 Mbit transmission rate.
FCS-Bytes	The **F**rame **C**heck **S**equenz-Bytes are a 32 Bit CRC-Checksum.
Flash	Internal memory, saved data will be available after power down.
HTML	The **H**yper**t**ext **M**arkup **L**anguage is a document format used in the World Wide Web to be displayed by a browser
HTTP	The **H**yper**t**ext **T**ransfer **P**rotocol is a stateless transmission protocol for data transmission.
Hub	The hub connects different network segments e.g. in an Ethernet network.
IP-Adresse	IP-address allow a logic addressing from computer in a network.
IP-Protokoll	The **I**nternet **P**rotocol is widespread in computer networks. It is the implementation of the internet layer of the TCP/IP-model
MODBUS	Is an application layer messaging protocol, positioned at level 7 of the OSI model, that provides client/server communication between devices connected on different types of buses or networks.
MODBUS/TCP	The Internet community can access MODBUS at a reserved system port 502 on the TCP/IP stack.
Mbit	Transmission rate or baud rate, million bits per second
SAG	Acronym: **SAG**, name of an encoder series manufactured by scancon
OSI-Modell	The **O**pen **S**ystem **I**nterconnection reference model is a open layer model for the organisation of a communication.
PDU	**P**rotocol **D**ata **U**nit
PPP-Packet	The **P**oint-to-**P**oint **P**rotocol will be need for a connection establishment. It enables the transmission between different network protocols.
SMTP	**S**imple **M**ail **T**ransfer **P**rotocol managed the transmission of e-mails.
Switch	A switch is an electronic device to connect computers e.g. network segments in a local network. Unlike a hub, a switch uses stacks to avoid network collisions.
TCP	The **T**ransmission **C**ontrol **P**rotocol is a connection orientated transmission protocol, in a network.
TCP-Client	MS-DOS program available from scancon to communicate with the encoder.
UDP	**U**ser **D**atagram **P**rotocol is utilized to send data that does not need to be transferred in a reliable way.