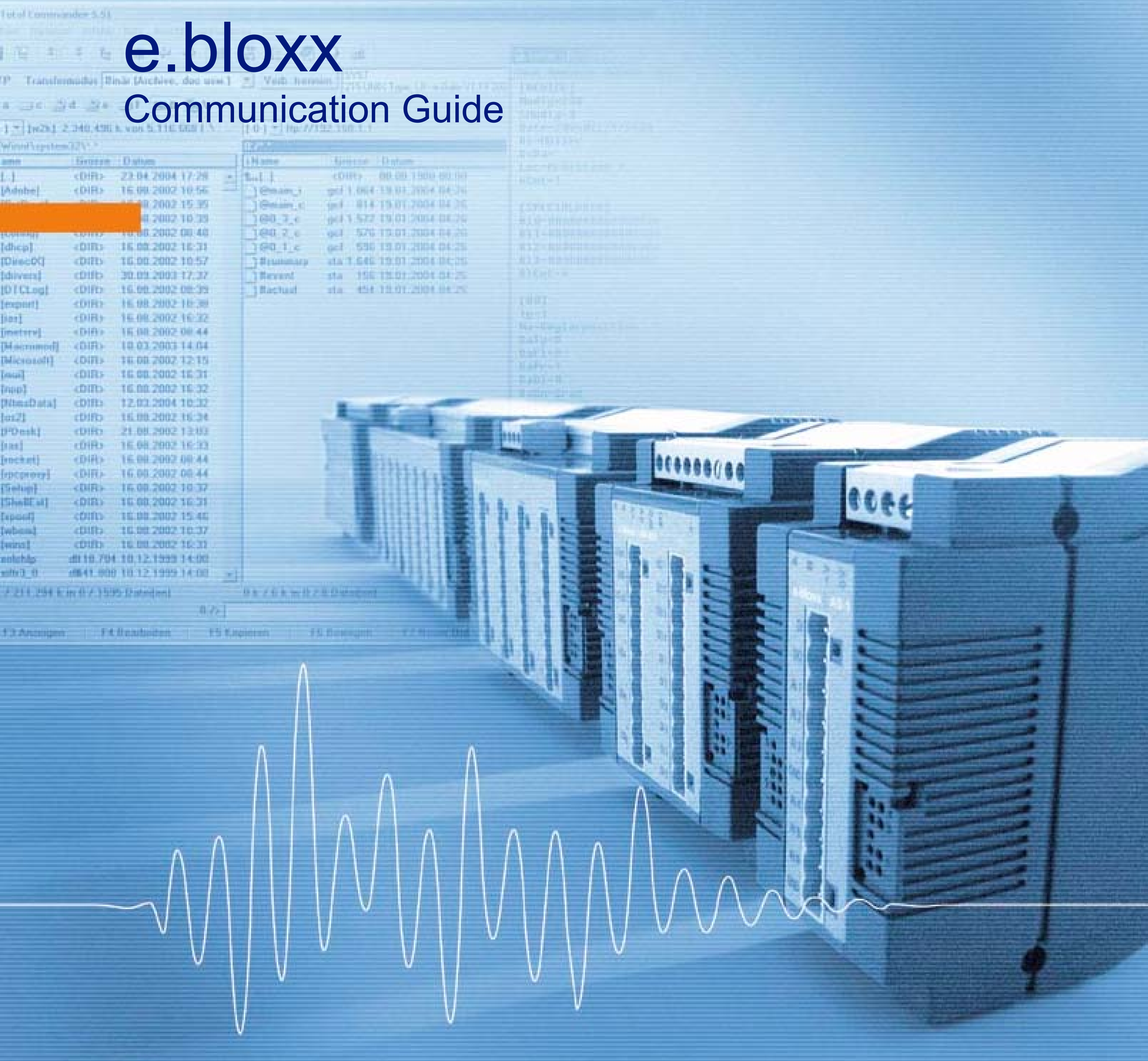
e.bloxx

Communication Guide



Liability: Any claims against the manufacturer based on the hardware or software products described in this manual shall depend exclusively on the conditions of the guarantee. Any further-reaching claims are excluded, and in particular the manufacturer accepts no liability for the completeness or accuracy of the contents of this manual. The right is reserved to make alterations, and alterations may be made at any time without prior notice being given.

Trade marks: Attention is drawn at this point to markings and registered trade marks used in this manual, in particular to those of Microsoft Corporation, International Business Machines Corporation and Intel Corporation.

Important: before commencing installation, commissioning, putting into service and before any maintenance work is carried out, it is essential that the relevant warning and safety instructions in this manual are read!

General warning and safety instructions:

Dear customer,

We congratulate you on having selected a product (device resp. software) of Gantner Instruments Test & Measurement GmbH. So that our product functions in your installation with safety and to your complete satisfaction, we take this opportunity to familiarize you with the following ground rules:

1. Installation, commissioning, operation and maintenance of the product purchased must be carried out in accordance with instructions, i.e. in accordance with the technical conditions of operation, as described in the corresponding product documentation.

2. Before installation, commissioning, operation or maintenance it is therefore essential that you read the corresponding chapter of this manual and observe its instructions.

3. If there are still some points on which you are not entirely clear, please do not take a chance, but ask the customer adviser responsible for you, or ring the Gantner Instruments Test & Measurement GmbH hot line.

4. Where not otherwise specifically laid down, appropriate installation, commissioning, operation and maintenance of the product is the customer's responsibility.

5. Directly on receipt of the goods, inspect both the packaging and the device resp. data carrier itself for any signs of damage. Also check that the delivery is complete (-> accessories, documentation, auxiliary devices, etc.).

6. If the packaging has been damaged in transport or should you suspect that the product has been damaged or that it may have a fault, the product must not be put into service. In this case, contact your customer advisor. He will make every effort to resolve the problem as quickly as possible.

7. Installation, commissioning and servicing of our product must only be carried out by suitably trained personnel. In particular, correspondingly qualified specialists may only make electrical connections. Here, the appropriate installation provisions in accordance with the relative national Electrical Engineers construction regulations (e.g. ÖVE, [Austrian] VDE, [German]...) must be observed.

8. Where not otherwise stated, installation and maintenance work on our appliances is exclusively to be carried out when disconnected from the power supply. This applies in particular to appliances which are normally supplied by low-tension current.

9. It is prohibited to make alterations to the appliances or to remove protective shields and covers.

10. Do not attempt yourself to repair an appliance after a defect, failure or damage, or to put it back into operation again. In such cases, it is essential you contact either your customer adviser or the Gantner Instruments Test & Measurement GmbH hot line. We will make every effort to resolve the problem as quickly as possible.

11. Gantner Instruments Test & Measurement GmbH accepts no responsibility for any injuries or damage caused as a result of improper use of the product.

12. Although every care is taken and we are continuously aiming for improvement, we cannot exclude completely the possibility of errors appearing in our documentation. Gantner Instruments Test & Measurement GmbH therefore accepts no responsibility for the completeness or the accuracy of this manual. The right is reserved to make alterations, and we may carry out alterations at any time without giving prior notice.

13. Should you discover any fault with the product or in its accompanying documentation, or have any suggestions for improvement, you may confidently approach either your customer adviser or Gantner Instruments Test & Measurement GmbH directly.

14. However, even if you just want to tell us that everything has functioned perfectly, we still look forward to hearing from you.

We wish you a successful application of our product. We will be pleased to welcome you as a customer again soon.

Contact address / manufacturer:

Gantner Instruments Test & Measurement GmbH
Montafonerstrasse 8
A - 6780 Schruns/Austria
Tel.: +43 5556 73784 - 410
Fax: +43 5556 73784 - 419
E-Mail: office@gantner-instruments.com
Web: www.gantner-instruments.com

Gantner Instruments Test & Measurement GmbH
Industriestraße 12
D-64297 Darmstadt
Tel.: +49 6151 95136 - 0
Fax: +49 6151 95136 - 26
E-Mail: testing@gantner-instruments.com
Web: www.gantner-instruments.com

TABLE OF CONTENTS

1. ABOUT THIS MANUAL

This manual describes the communication of the modules of the series e.bloxx. It contains detailed information like telegram format, instruction set and parameters of the communication protocols ASCII, PROFIBUS-DP, MODBUS and LOCAL-BUS.

A chapter describes the setup of a Hilscher® Master and a Siemens® S7-300 PLC for communication via PROFIBUS-DP.

2. COMMUNICATION

2.1. Bus Interface

The bus interface of the *series e.bloxx* is a RS 485 interface according to the specifications of the EIA-RS485 USA standard.

2.2. Bus Protocol

With the series *e.bloxx* all protocols are available within one download file. The module itself recognizes the type of protocol for communication depending on the request from the host. Therefore the protocol has not to be selected manually.

- ASCII protocol
- PROFIBUS-DP protocol according to DIN 19245, part 1
- MODBUS-RTU protocol acc. to Reference Guide PI-MBUS-300 Rev. D
- LOCAL-BUS protocol

2.3. Character Formats

The *e.bloxx* supports the following character format:

Format	Start Bit	Data Bit	Parity Bit	Stop Bit	Char. Length
8E1	1	8	E	1	11

Table 2.1 Supported character transfer format

The character format 8E1 with even parity (E=even) corresponds to the PROFIBUS-definitions according to DIN 19245, part 1, and is supported by the sensor modules in the PROFIBUS protocol as well as in the ASCII, MODBUS and LOCAL-BUS protocol. This character format is predefined with the e.bloxx modules and cannot be changed.

2.4. Output Format

The user can preset the format in which data shall be output via the bus with the *Configuration Software ICP100.* The modules adjust the data format accordingly and makes sure that data are available in the selected unit.

For the transmission in the ASCII and PROFIBUS format the format settings listed in table 2.2 and 2.3 can be chosen. At the transmission in the MODBUS format the output format (integer or real) will automatically be confirmed (table 2.4). The coding of a real value in the MODBUS and PROFIBUS format is as follows:

Coding of the real value: x = s ee...ee mmm.....mmm
Value: $(-1)^s \cdot 2^{e-127} \cdot 1{,}m$ # : <1> <- 8 -> <----- 23 ----->

Format Settings	Range of Values
Unit	dependent on sensor
Field Length	1 8
Decimals	0 . . . field length-1 (max 6)

Table 2.2 Format settings for transmission in the ASCII-format

Format Settings	Length	Range of Values
Bool	1 byte	(dec 0: FALSE) and (dec 255: TRUE)
Integer	2 byte	(dec - 32768) ≤ i ≤ (dec +32767)
Real	4 byte	(dec - 2^{129}) ≤ x ≤ (dec + 2^{129})
SET 8	1 byte	(dec 0) ≤ i ≤ (dec 255)

Table 2.3 Format settings for transmission in the PROFIBUS-format

Format Settings	Length	Range of Values
Integer	2 byte	(dec - 32768) ≤ i ≤ (dec +32767)
Real	4 byte	(dec - 2^{129}) ≤ x ≤ (dec + 2^{129})

Table 2.4 Format settings for transmission in the MODBUS-format

Example: The value 50.3094 cm shall be displayed.

Transmission in the ASCII-format:

Decimals	Field Length 6	Field Length 7	Field Length 8
0	____50	_____50	______50
1	__50.3	___50.3	____50.3
2	_50.31	__50.31	___50.31
3	50.309	_50.309	__50.309
4	E.3094	50.3094	_50.3094
5	-	E.30940	50.30940
6	-	-	E.309400

Table 2.5   Output formats for transmission in the ASCII-format ("_":blank).

Transmission in PROFIBUS- and MODBUS-format:

Decimals	Integer	Real
0	00 32 (50)	42 49 3C D3 (50.3094)
1	01 F7 (503)	42 49 3C D3 (50.3094)
2	13 A6 (5030)	42 49 3C D3 (50.3094)
3	xx xx (50309)	42 49 3C D3 (50.3094)
4	xx xx (503094)	42 49 3C D3 (50.3094)
5	xx xx (5030940)	42 49 3C D3 (50.3094)
6	xx xx (50309400)	42 49 3C D3 (50.3094)

Table 2.6   Output formats for transmission in the PROFIBUS- and the MODBUS-format (the decimal notation is given in parentheses).

The following points have to be kept in mind:

- Decimals are not cut off, but are rounded off.

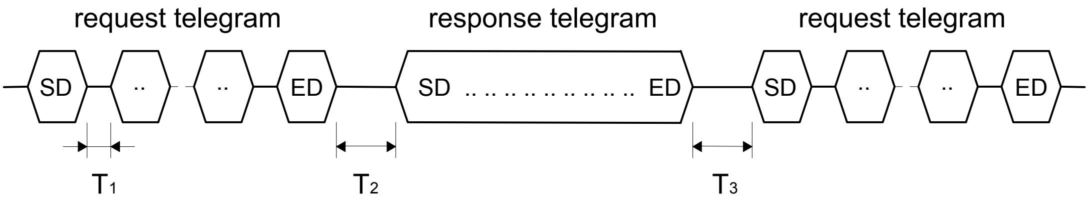
- In case of overflow with a transmission in ASCII format the identification key "E" (for Format Error) is indicated at the first position in the transmission format.

- With a transmission in PROFIBUS and MODBUS format no identification key is given in case of an overflow. The number of decimals must, however, not be selected too large, if the value is to be transmitted in integer format (range of values in integer format limited to -32768 to +32767).

3. ASCII PROTOCOL

3.1. Transmission Sequence

In the ASCII-protocol data are transmitted from and to the e.bloxx module by means of the following sequence:



T_1: Time between two characters
T_2: Time between request-telegram and corresponding response-telegram
T_3: Time between response-telegram and next request-telegram

You will find the minimum and maximum appearing values for T_1, T_2 and T_3 and the adjustment range in the following table 3.1.

Protocol	Baud rate	T_{1min}	T_{1max}	T_{2min}	T_{2max}	T_{3min}	T_{3max}
adjustable		no	no	yes	no	no	yes
A S C I I	19,200 bps 38,400 bps 57,600 bps 115,200 bps	0	1 CT	1... 42 CT 1..85 CT 1..85 CT 1..85 CT	T_{2min} x 1.2	3 CT	0.1 sec to 600 sec

Table 3.1 *Values and adjustment range for the times T_1, T_2 and T_3*
(CT: character time: 1 CT = character length [bit] / baud rate [bps])

Notice: In the ASCII-protocol T_{2max} lasts at least 12 ms.

The values for T_{2min} and T_{3max} and the behavior of the sensor module if the time T_{3max} is exceeded (communication timeout) can be adjusted by means of the *Configuration Software ICP 100*. The default values for the time T_{2min} is 1 CT and for the time T_{3max} 60 sec.

3.2. Telegram Format

For the request and response telegram there is a difference between telegrams without and with check sum in the ASCII protocol. The various telegrams differ by their varying Start-Delimiters (SD). A request telegram without check sum will lead to a response telegram which also will contain no check sum. This is valid for requests with check sum as well. Furthermore there are two short telegrams with a length of one character each. With these telegrams a positive or negative acknowledge respectively can be performed.

Request Telegram Without Checksum

SD	DA	ReqDataUnit	ED
1	2	n	1

n+4 Characters

Response Telegram Without Checksum

SD	ResDataUnit	ED
1	n	1

n+2 Characters

Request Telegram With Checksum

SD	DA	ReqDataUnit	FCS	ED
1	2	n	2	1

n+6 Characters

Response Telegram With Checksum

SD	ResDataUnit	FCS	ED
1	n	2	1

n+2 Characters

Positive Acknowledge

ACK

1 Character

Negative Acknowledge

NAK

1 Character

SD: Start-Delimiter (Length = 1 byte):
The Start-Delimiter SD marks the beginning of a telegram. In an ASCII protocol it has the following values:

SD	Request Telegram	Response Telegram
with check sum	#	>
without check sum	$	=

Table 3.2 Start-Delimiter (SD) in the ASCII-protocol

DA: Destination-Address (Length = 2 byte):
The Destination-Address DA identifies the communication partner's address, to whom data shall be transmitted or from whom data shall be requested. Destination-Address can have a value from 1 to 127 in an ASCII-protocol. Here the value is given as a two-digit ASCII-string (ASCII "01".."7F").

ReqDataUnit: Request-Data-Unit (Length = 1 ... n byte):
The Request-Data-Unit identifies a data field in the request telegram, which contains data for the communication partner with the DA address.
ResDataUnit: Response-Data-Unit (Length = 1 ... n byte):

The Response-Data-Unit identifies a data field in the response telegram, which contains data for the calling communication partner.

FCS: Frame-Check-Sequence (Length = 2 byte):
The Frame-Check-Sequence FCS identifies the running digital sum of the telegram. This is the sum of the ASCII-values in the telegram modulo 256. It is calculated in the ASCII-protocol from Start-Delimiter (SD), Destination Adress (DA) and Data-Unit: CheckSum_ASCII = [SD+DA+DataUnit] mod 256. In the ASCII-protocol the value is indicated as a two-digit ASCII-string (ASCII "00"..."FF").

ED: End-Delimiter (Length = 1 byte):
The End-Delimiter ED identifies the end of the telegram. In an ASCII-protocol it has the value hex 0D ("Cr").

ACK: Acknowledge (length = 1 byte):
With a request, where no data are returned, the proper performance of the instruction is acknowledged by an "Acknowledge"-character (hex 06).

NAK: No-Acknowledge (length = 1 byte):
When a request has not been performed in correct way, a "No Acknowledge" (hex 15) is sent back.

3.3. Instruction Set

Check Sum	Request Telegram	Response With Orderly Performance	Response in Case of Error
read device identification			
with	# aa V cc <cr>	> v..v cc <cr>	NAK
without	$ aa V <cr>	= v..v <cr>	NAK
read device information			
with	# aa S cc <cr>	> s..s cc <cr>	NAK
without	$ aa S <cr>	= s..s <cr>	NAK
read status information			
with	# aa Z cc <cr>	> z..z cc <cr>	NAK
without	$ aa Z <cr>	= z..z <cr>	NAK
read variable information			
with	# aa B kk cc <cr>	> b..b cc <cr>	NAK
without	$ aa B kk <cr>	= b..b <cr>	NAK
read data from a variable			
with	# aa R kk cc <cr>	> d..d cc <cr>	NAK
without	$ aa R kk <cr>	= d..d <cr>	NAK
write data to a variable			
with	# aa W kk d..d cc <cr>	ACK	NAK
without	$ aa W kk d..d <cr>	ACK	NAK

Table 3.3 Instruction set in ASCII-protocol

Char	Meaning	Length	Range
#	start delimiter for request telegram with check sum	1	ASCII "#"
>	start delimiter for response telegram with check sum	1	ASCII ">"
$	start delimiter for request telegram without check sum	1	ASCII "$"
=	start delimiter for response telegram without check sum	1	ASCII "="
<cr>	end delimiter (carriage return)	1	*hex* 0D
ACK	positive acknowledge	1	*hex* 06
NAK	negative acknowledge	1	*hex* 15
aa	destination address	2	ASCII "01".."7F"
cc	check sum	2	ASCII "00".."FF"
kk	variable number	2	ASCII "00".."0F"
v..v	device identification	x	ASCII - String
s..s	device information	27	ASCII - String
z..z	status information	4	ASCII - String
b..b	variable information	29	ASCII - String
d..d	variable value	max. 8	ASCII - String

Table 3.4 Explanation of command characters in ASCII-protocol

3.4. Instruction Parameters

device identification (v...v)			length = x char
<vendor name>	ASCII	("Gantner,")	1..32 char
<model name>	ASCII	(<type of module>,)	1..32 char
<hw version>	ASCII	("xy.yy,")	1..8 char
<sw version>	ASCII	("xy.yy,")	1..8 char

device information (s...s)		length = 27 char
<location>	ASCII	20 char
<serial number>	ASCII	6 char
<number of variables>	ASCII	1 char

status information (z...z)		length = 6 char
<variable status>	ASCII	4 char
<module status>	ASCII	2 char

<variable st.>	= K16..K13	K12..K9	K8..K5	K4..K1	= *hex* 0XYZ
	0	X	Y	Z	*ASCII* "0XYZ"

<module st.>	= M8 M7 M6 M5	M4 M3 M2 M1	= *hex* XY
	X	Y	*ASCII* "XY"

If the bit Kn in the variable status is set it indicates that an error has occurred in variable n. A variable error is given when the measuring value is outside of the linearization, e.g. in consequence of a sensor break down or of a short circuit of transmission.

If the bit Mn in the module status is set it indicates that an error has occurred in the sensor module. Valid is:

M1 = 1:	EEPROM - error	M5 = 1:	(currently not occupied)
M2 = 1:	FLASH - error	M6 = 1:	(currently not occupied)
M3 = 1:	ADC - error	M7 = 1:	(currently not occupied)
M4 = 1:	configuration - error	M8 = 1:	(currently not occupied)

variable information (b...b)		length = 29 char
<variable type>	ASCII	1 char
<variable name>	ASCII	20 char
<data format>	ASCII	1 char
<field length>	ASCII	1 char
<decimals>	ASCII	1 char
<unit>	ASCII	4 char
<host input>	ASCII	1 char

Coding <variable type>:

ASCII "0":	Empty Variable (EM)	"4":	Digital Input Var. (DI)
ASCII "1":	Analog Input Variable (AI)	"5":	Setpoint Variable (SP)
ASCII "2":	Arithmetic Variable (AR)	"6":	Alarm Variable (AL)
ASCII "3":	Digital Output Variable (DO)	"9":	Controller Variable (CO)

Coding <data format>:

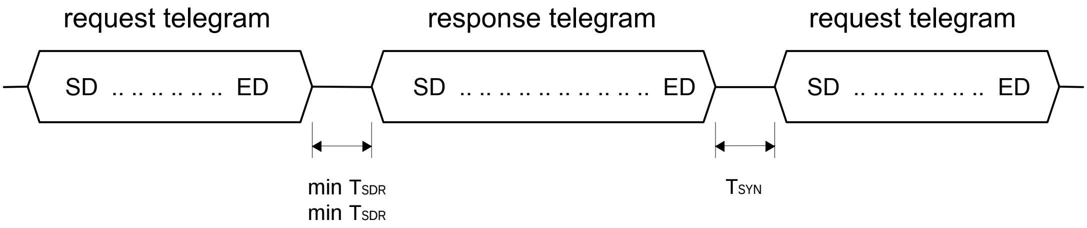
ASCII "0":	no format
ASCII "1":	BOOL
ASCII "4":	INTEGER
ASCII "6":	LONG
ASCII "8":	REAL
ASCII "9":	SET 8

Coding <host input>:
ASCII "0": host input is not possible
ASCII "1": host input is possible (tare/reset/dig.out/setpoint values)

4. PROFIBUS PROTOCOL

4.1. Transmission Sequence

In the PROFIBUS protocol the data are transmitted from and to the sensor module by means of the following sequence:



min TSDR: minimum time between request telegram and corresponding response telegram
max TSDR: maximum time between request telegram and corresponding response telegram
TSYN: time between response-telegram and next request-telegram

You will find the minimum and maximum appearing values for min TSDR, max TSDR and TSYN and the adjustment range in the following table 4.1.

Protocol	Baud rate	min TSDR	max TSDR	TSYN
adjustable		yes	no	yes
PROFIBUS	19.2 kbps	11 tbit	60 tbit	33 tbit
	93.75 kbps	11 tbit	60 tbit	
	187.5 kbps	11 tbit	60 tbit	
	500 kbps	13 tbit	100 tbit	
	1,500 kbps	50 tbit	150 tbit	

Table 4.*1* *Values and adjustment range for the times min TDSR, max TDSR and TSYN*

The PROFIBUS-DP is running on the e.bloxx devices as software stack and is not based on an ASIC-device.

The PROFIBUS ID number for the e.bloxx modules is 6782 in hex format.

4.2. Diagnostic Data

This section describes the coding of the "Extended Diagnostic Data" of PROFIBUS-DP for the e.bloxx series. The extended diagnostic data are added in the section of equipment specific status and error messages of the PROFIBUS-DP "Get-Diagnostic-Data" frame. The displayed messages of a PROFIBUS-DP Master depend on the manufacturer specification.

4.2.1. GSD File

For the e.bloxx series the GSD file GEA6782.gsd, V1.00, date July 8th 2003 is the actual version. With the GSD file a status or error message can be decoded by the PROFIBUS-DP master.

The corresponding part of the GSD file is:

```
" ...
Unit_Diag_Bit (0) = "Modbus Error 1"
Unit_Diag_Bit (1) = "Modbus Error 2"
Unit_Diag_Bit (2) = "Modbus Error 3"
Unit_Diag_Bit (3) = "Modbus Error 4"
Unit_Diag_Bit (4) = "Modbus Error 5"
Unit_Diag_Bit (5) = "Modbus Error 6"
Unit_Diag_Bit (6) = "Modbus Error 7"
Unit_Diag_Bit (7) = "Modbus Error 8"
Unit_Diag_Bit (8) = "Error Var 1"
Unit_Diag_Bit (9) = "Error Var 2"
Unit_Diag_Bit (10) = "Error Var 3"
Unit_Diag_Bit (11) = "Error Var 4"
Unit_Diag_Bit (12) = "Error Var 5"
Unit_Diag_Bit (13) = "Error Var 6"
Unit_Diag_Bit (14) = "Error Var 7"
Unit_Diag_Bit (15) = "Error Var 8"
Unit_Diag_Bit (16) = "Error Var 9"
Unit_Diag_Bit (17) = "Error Var 10"
Unit_Diag_Bit (18) = "Error Var 11"
Unit_Diag_Bit (19) = "Error Var 12"
Unit_Diag_Bit (20) = "Error Var 13"
Unit_Diag_Bit (21) = "Error Var 14"
Unit_Diag_Bit (22) = "Error Var 15"
Unit_Diag_Bit (23) = "Error Var 16"
... "
```

Diagnostic bits 0 to 7 will indicate module specific messages. Bits 8 to 23 are assigned to the variables 1 to 16 of each module. Each module transmits 3 bytes, even though not all bits may be occupied for the module type.

4.2.2. Firmware Versions

"Extended Diagnostic Data" are supported via PROFIBUS-DP by the following firmware version and higher versions.

Device	FW Version	FUP File name
e.bloxx A1-x	X1.00 / A1.03	mk18#_Standard_x100_a103.fup
e.bloxx A4-x	X1.00 / A1.10	Mk26#_Standard_x100_a110.fup
e.bloxx A5-x	X0.30 / A0.60	mk28#_Standard_x030_a060.fup
e.bloxx A9-x	X1.00 / A1.10	Mk30#_Standard_x100_a110.fup
e.bloxx A6-2CF	X1.00 / A1.10	Mk29#_Standard_x100_a110.fup
e.bloxx D1-x	X1.00 / A1.10	Mk31#_Frequency_x100_a110.fup
e.bloxx D1-x	X1.00 / A1.10	Mk31#_Standard_x100_a110.fup

Table 4.2 *- Firmware Versions which support Extended Diagnostic Data*

4.2.3. Diagnostic Data Decoding Table

The error codes in the GSD file are not all supported by each e.bloxx module.

GSD Coding	Meaning of the Error Code	e.bloxx A1-x	e.bloxx A4-x	e.bloxx A5-1	e.bloxx A6-2CF	e.bloxx A9-1	e.bloxx D1-x
Module Error 1	EEPROM – Error	x	x	x	x	x	x
Module Error 2	FLASH – Error	x	x	x	x	x	x
Module Error 3	ADC – Error	x	x	x	x	x	x
Module Error 4	Configuration – Error	x	x	x	x	x	x
Module Error 5	Currently not defined / reserved						
Module Error 6	Currently not defined / reserved						
Module Error 7	Currently not defined / reserved						
Module Error 8	Currently not defined / reserved						
Error Var 1	Error for variable 1 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 2	Error for variable 2 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 3	Error for variable 3 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 4	Error for variable 4 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 5	Error for variable 5 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 6	Error for variable 6 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 7	Error for variable 7 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 8	Error for variable 8 (sensor break, linearisation error)	x	x	x	x	x	x
Error Var 9	Error for variable 9 (sensor break, linearisation error)				x		
Error Var 10	Error for variable 10 (sensor break, linearisation error)				x		
Error Var 11	Error for variable 11 (sensor break, linearisation error)				x		
Error Var 12	Error for variable 12 (sensor break, linearisation error)				x		
Error Var 13	Error for variable 13 (sensor break, linearisation error)				x		
Error Var 14	Error for variable 14 (sensor break, linearisation error)				x		
Error Var 15	Error for variable 15 (sensor break, linearisation error)				x		
Error Var 16	Error for variable 16(sensor break, linearisation error)				x		

Table 4.3 *- Diagnostic Data Decoding Table*

4.3. Launching PROFIBUS-DP Communication with a Hilscher® Master

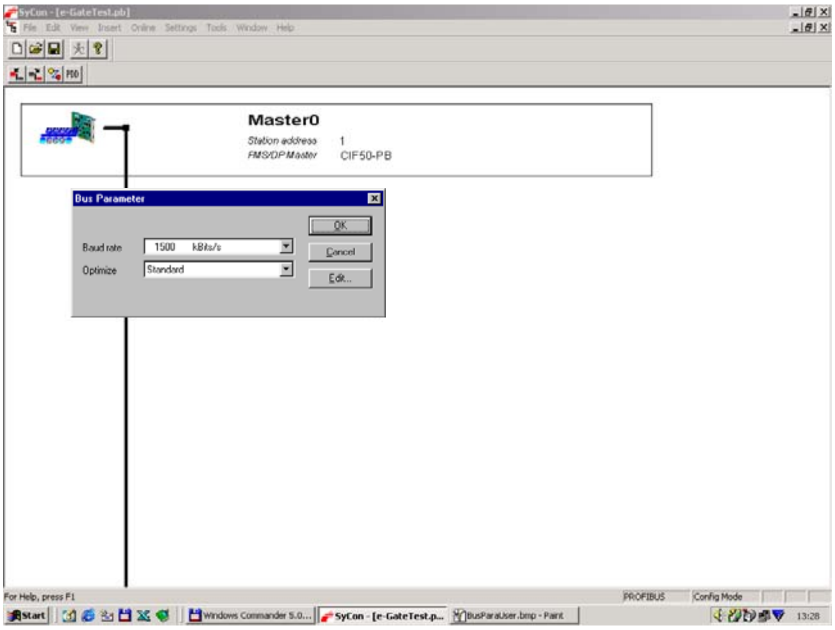
On the following pages the configuration of the Hilscher® Master will be described.

4.3.1. How to Launch PROFIBUS-DP Communication with a Hilscher® Master

The newest e.bloxx GSD file (GEA6782.gsd, shipped with the ICP 100 software and located in the \GSD directory in the installation directory of ICP 100 or directly received from the Gantner Instruments Test & Measurement GMBH) has to be copied to the directory "\Fieldbus\Profibus\GSD" in the SyCon® installation directory.

Now process the steps below in the Hilscher® Master software:

- In the software add one master to the system layout.
- Open the configuration of "Bus Parameters".
- Select the desired Baudrate and than set "Standard" first in the "Optimize" selection field.

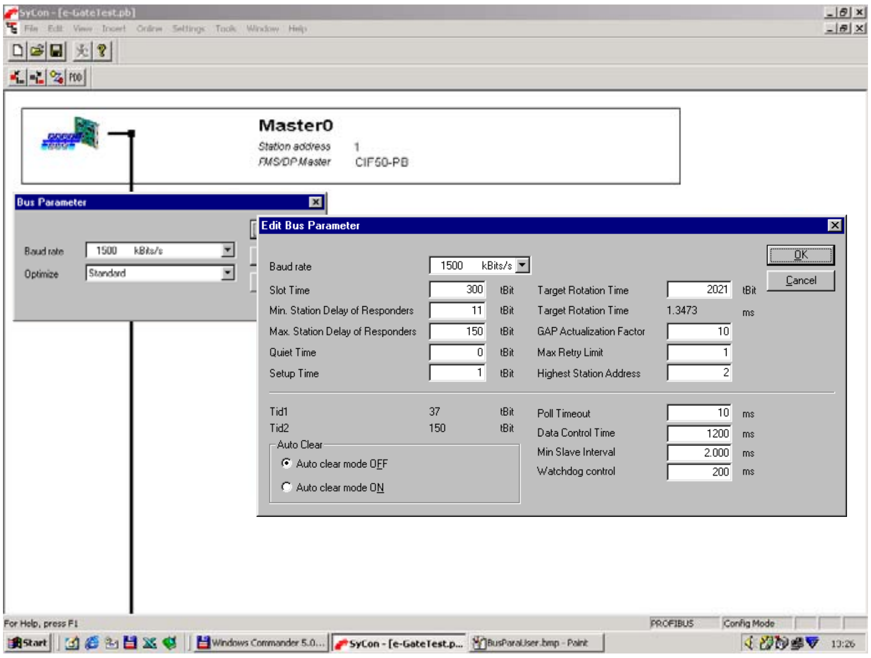


SyCon - [e-GateTest.pb]

Master0
Station address 1
FMS/DP Master CIF50-PB

Bus Parameter
Baud rate 1500 kBits/s
Optimize Standard

- Next select "By User" in the "Optimize" selection field and on the window that opens change the items marked in the next picture.



Master0
Station address 1
FMS/DP Master CIF50-PB

Bus Parameter

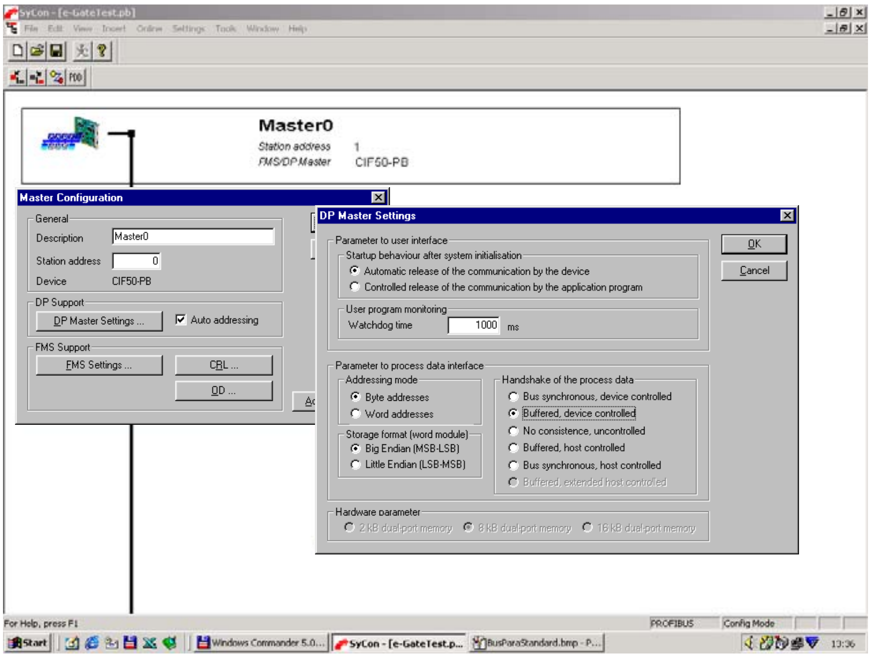
Baud rate 1500 kBits/s
Optimize Standard

Edit Bus Parameter

Baud rate 1500 kBits/s
Slot Time 300 tBit
Min. Station Delay of Responders 11 tBit
Max. Station Delay of Responders 150 tBit
Quiet Time 0 tBit
Setup Time 1 tBit
Target Rotation Time 2021 tBit
Target Rotation Time 1.3473 ms
GAP Actualization Factor 10
Max Retry Limit 1
Highest Station Address 2
Tid1 37 tBit
Tid2 150 tBit
Auto Clear
● Auto clear mode OFF
○ Auto clear mode ON
Poll Timeout 10 ms
Data Control Time 1200 ms
Min Slave Interval 2.000 ms
Watchdog control 200 ms

"Highest Station Address": Highest used slave address in the system
"Min Slave Interval": Needed update rate

- Now you have to set the correct process interface settings. Therefore open the "DP Master Settings" window and in this window you have to mark "Buffered, device controlled" in the "Handshake of the process data" field in order to guarantee data consistency in transfer between master and user application.



Master0
Station address 1
FMS/DP Master CIF50-PB

Master Configuration

General
Description Master0
Station address 0
Device CIF50-PB
DP Support
☑ Auto addressing

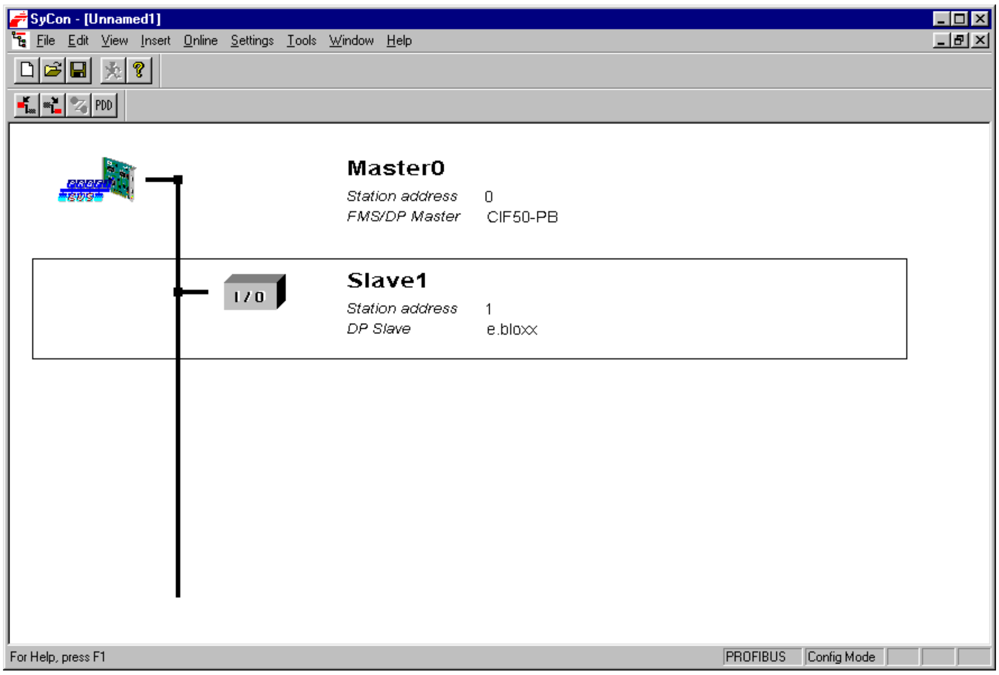
DP Master Settings

Parameter to user interface
Startup behaviour after system initialisation
● Automatic release of the communication by the device
○ Controlled release of the communication by the application program
User program monitoring
Watchdog time 1000 ms

Parameter to process data interface
Addressing mode
● Byte addresses
○ Word addresses
Storage format (word module)
● Big Endian (MSB-LSB)
○ Little Endian (LSB-MSB)
Handshake of the process data
○ Bus synchronous, device controlled
● Buffered, device controlled
○ No consistence, uncontrolled
○ Buffered, host controlled
○ Bus synchronous, host controlled
○ Buffered, extended host controlled

Hardware parameter
○ 2 kB dual-port memory ● 8 kB dual-port memory ○ 16 kB dual-port memory

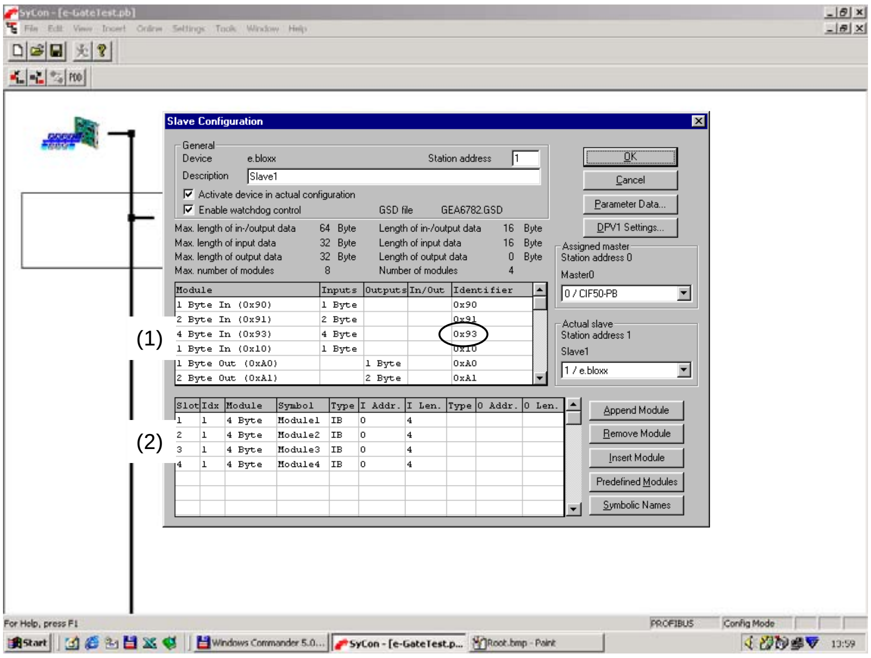
- Add one e.bloxx slave to the system layout and set the address of this slave.



Master0
Station address 0
FMS/DP Master CIF50-PB

Slave1
Station address 1
DP Slave e.bloxx

- For the configuration of PROFIBUS-DP variables information is read from the GSD File. The variable settings will be displayed in the list field (1) in the window "Slave Configuration".



General
Device e.bloxx Station address 1
Description Slave1
☑ Activate device in actual configuration
☑ Enable watchdog control GSD file GEA6782.GSD

Max. length of in-/output data 64 Byte Length of in-/output data 16 Byte
Max. length of input data 32 Byte Length of input data 16 Byte
Max. length of output data 32 Byte Length of output data 0 Byte
Max. number of modules 8 Number of modules 4

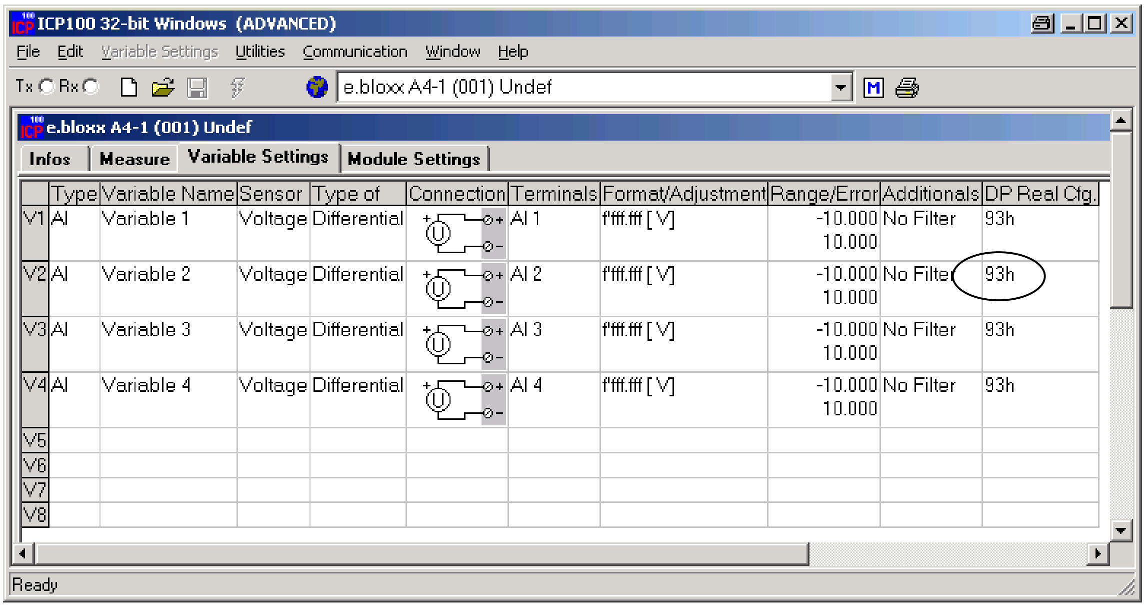
Assigned master
Station address 0
Master0
0 / CIF50-PB

Actual slave
Station address 1
Slave1
1 / e.bloxx

Module	Inputs	Outputs	In/Out	Identifier
1 Byte In (0x90)	1 Byte			0x90
2 Byte In (0x91)	2 Byte			0x91
4 Byte In (0x93)	4 Byte			0x93
1 Byte In (0x10)	1 Byte			0x10
1 Byte Out (0xA0)		1 Byte		0xA0
2 Byte Out (0xA1)		2 Byte		0xA1

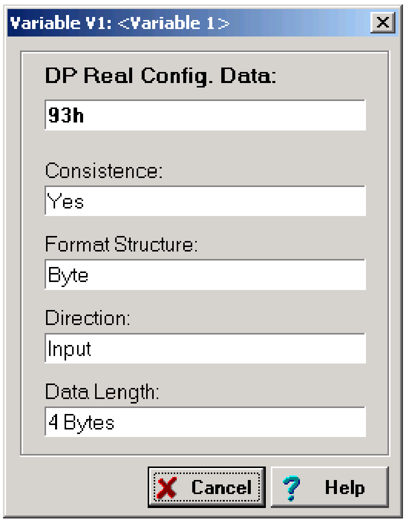
Slot	Idx	Module	Symbol	Type	I Addr.	I Len.	Type	O Addr.	O Len.
1	1	4 Byte	Module1	IB	0	4			
2	1	4 Byte	Module2	IB	0	4			
3	1	4 Byte	Module3	IB	0	4			
4	1	4 Byte	Module4	IB	0	4			

- The configuration of an e.bloxx module is done in the Configuration Table of the *Configuration Software ICP 100* (see next picture).



	Type	Variable Name	Sensor	Type of	Connection	Terminals	Format/Adjustment	Range/Error	Additionals	DP Real Cfg.
V1	AI	Variable 1	Voltage	Differential		AI 1	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V2	AI	Variable 2	Voltage	Differential		AI 2	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V3	AI	Variable 3	Voltage	Differential		AI 3	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V4	AI	Variable 4	Voltage	Differential		AI 4	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V5										
V6										
V7										
V8										

- In this Configuration Table each variable has a "DP Real Cfg." field. The values in these fields have to be the same as in the DP Master (see markings).
- Next add the variables with the buttons "Append Module" and "Insert Module" to the field (2) in the DP Master Configuration. Therefore first mark a variable in the field (1) and then press those buttons to add them to the field (2). The order of the variables has to be the same as in the Configuration Table (from top to bottom).
- If you click with the right mouse button on the field "DP Real Cfg." of a variable in the Configuration Table the following window will open.



DP Real Config. Data: 93h

Consistence: Yes

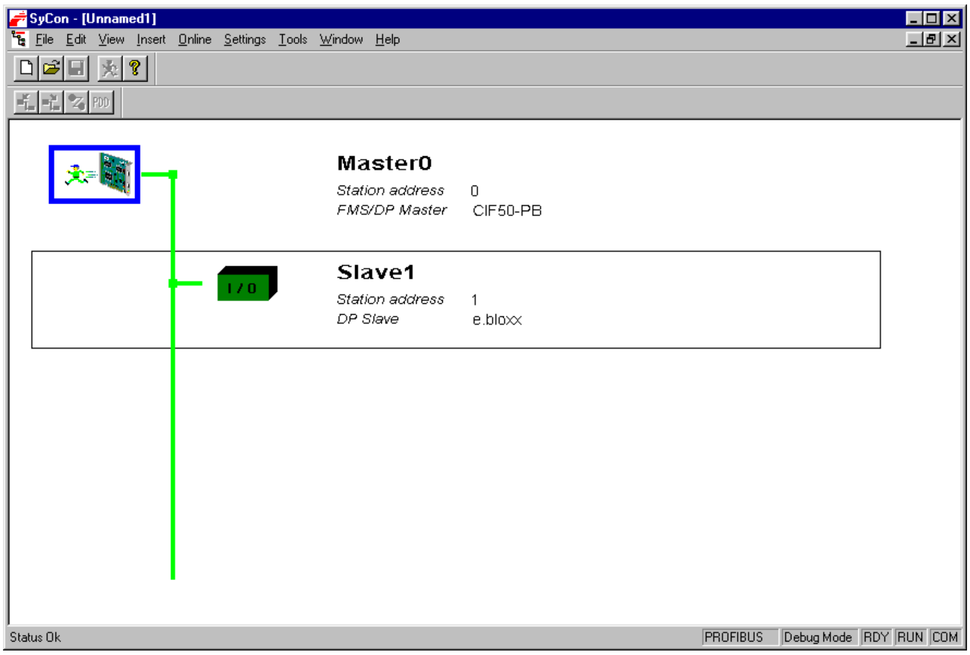
Format Structure: Byte

Direction: Input

Data Length: 4 Bytes

- Here you can see the DP-Settings which must be the same as shown in the DP Master in the lower list field where you added the variables (2).

- In order to activate the configuration, select the master again and perform a "Download ...".
- The last step is to switch on the "Start Debug Mode”, so that successful or erroneous working of system can be viewed.



Master0
Station address 0
FMS/DP Master CIF50-PB

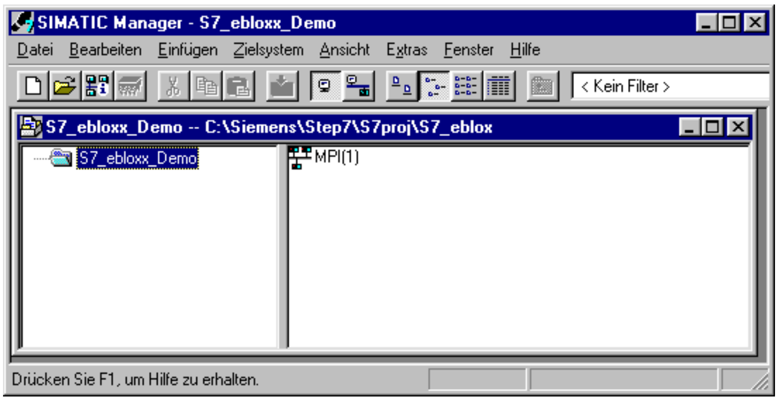
Slave1
Station address 1
DP Slave e.bloxx

4.4. PROFIBUS-DP Configuration for a Siemens® S7-300

In order to set-up a Siemens® S7-300 for a PROFIBUS-DP configuration with e.bloxx the SIMATIC Manager is used. In this software a SIMATIC 300 Station is inserted and the e.bloxx modules are connected to this station via PROFIBUS-DP. Each e.bloxx module is then configured separately according to their variable settings in the Configuration Software ICP 100.

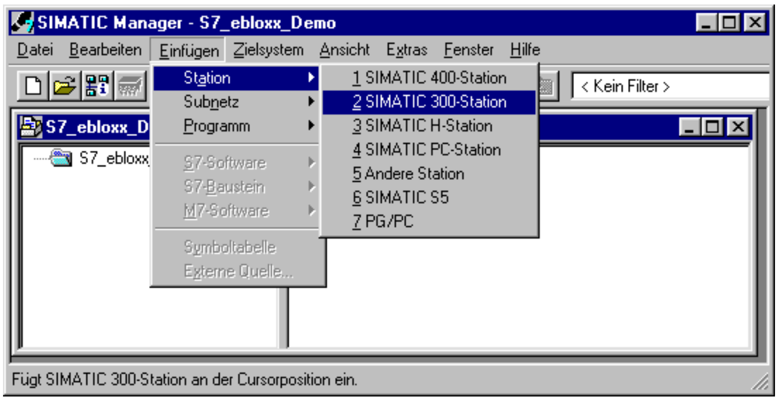
In order to set-up a PROFIBUS-DP configuration for a Siemens® S7-300 follow the next steps:

- Start SIMATIC Manager.
- Begin with a new project (icon "New" or menu item "New Project" in the menu "File").


SIMATIC Manager - S7_ebloxx_Demo
Datei Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe
< Kein Filter >
S7_ebloxx_Demo -- C:\Siemens\Step7\S7proj\S7_eblox
S7_ebloxx_Demo
MPI(1)

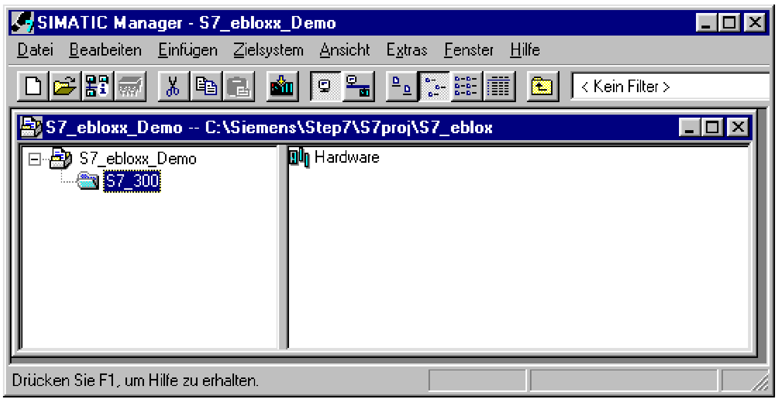
Drücken Sie F1, um Hilfe zu erhalten.

- Insert a new SIMATIC 300 Station. Therefore select the menu item "Station" -> "2 SIMATIC 300-Station" in the menu "Insert".



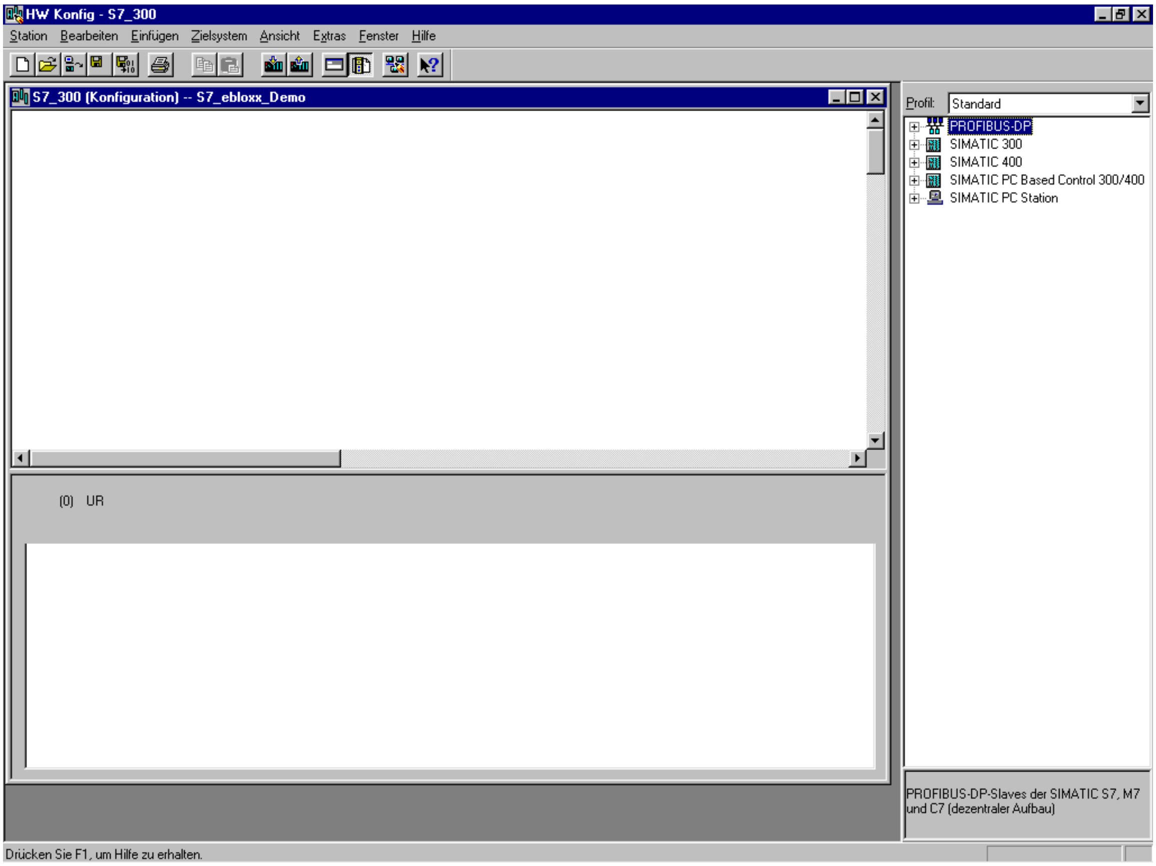
Fügt SIMATIC 300-Station an der Cursorposition ein.

- Now click on the new entry for the SIMANTIC 300 Station.


SIMATIC Manager - S7_ebloxx_Demo
Datei Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe
< Kein Filter >
S7_ebloxx_Demo -- C:\Siemens\Step7\S7proj\S7_eblox
S7_ebloxx_Demo
S7_300
Hardware

Drücken Sie F1, um Hilfe zu erhalten.

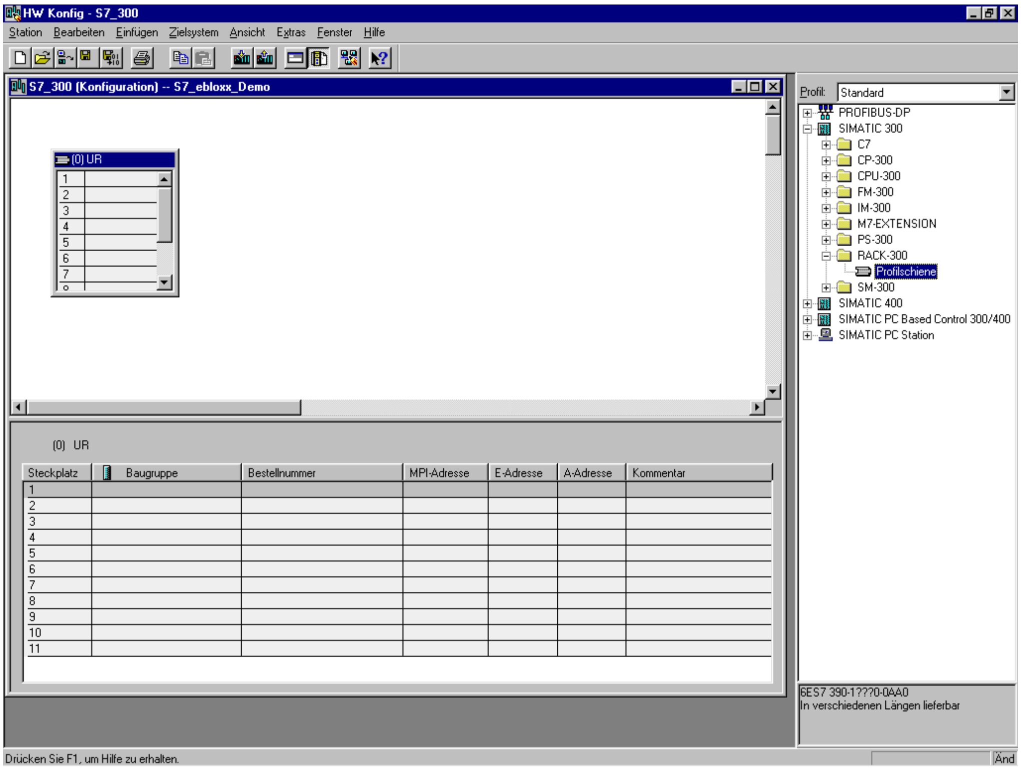
- Double-click on the entry "Hardware" in the right field. A window for the hardware configuration will be opened.


HW Konfig - S7_300
Station Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe
S7_300 (Konfiguration) -- S7_ebloxx_Demo
(0) UR
Profil: Standard
PROFIBUS-DP
SIMATIC 300
SIMATIC 400
SIMATIC PC Based Control 300/400
SIMATIC PC Station

PROFIBUS-DP-Slaves der SIMATIC S7, M7 und C7 (dezentraler Aufbau)

Drücken Sie F1, um Hilfe zu erhalten.

- Open the directory "SIMANTIC 300" in the list field on the right side of this windows. Then go to the sub-directory RACK-300. There you find the entry "Profile Rail". Click on this entry and drag it with the mouse onto the main window (see next picture).



(0) UR

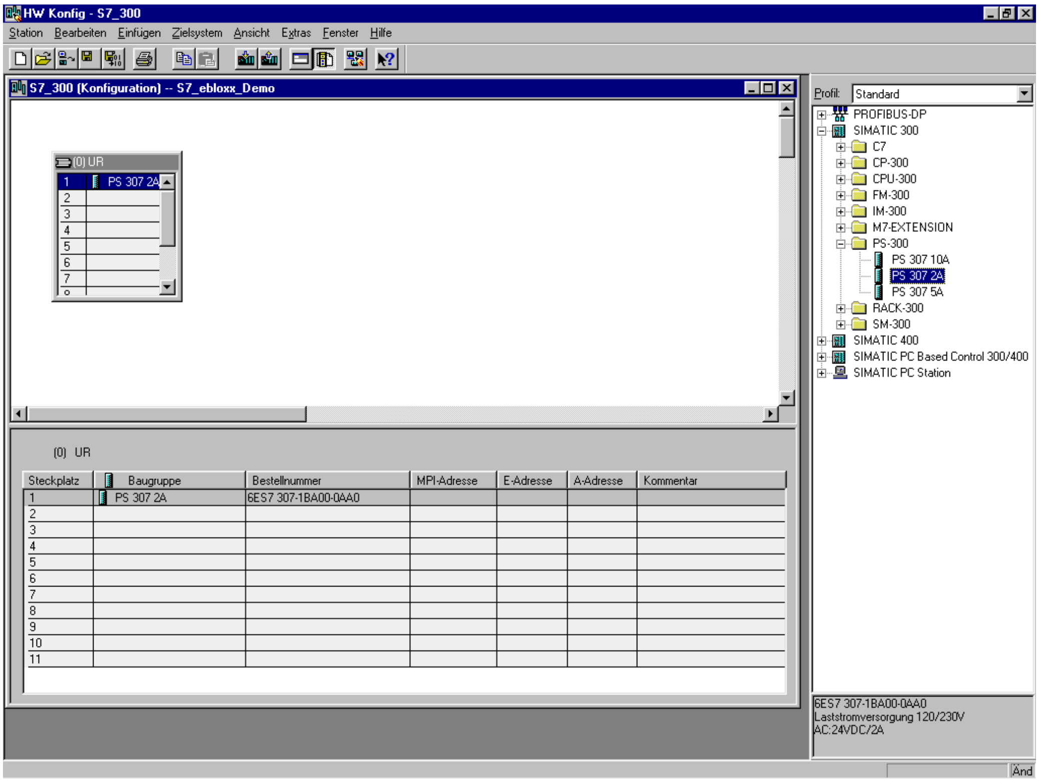
Steckplatz	Baugruppe	Bestellnummer	MPI-Adresse	E-Adresse	A-Adresse	Kommentar
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						

6ES7 390-1???0-0AA0
In verschiedenen Längen lieferbar

Drücken Sie F1, um Hilfe zu erhalten.

Änd

- Now you see the Profile Rail Element in the main window.
- On the right list field open the directory "SIMATIC 300". Then open the sub-directory PS-300. There you find the entry PS 307 2A for the power supply. Drag this entry into the first channel of the Profile Rail Element (see next picture).



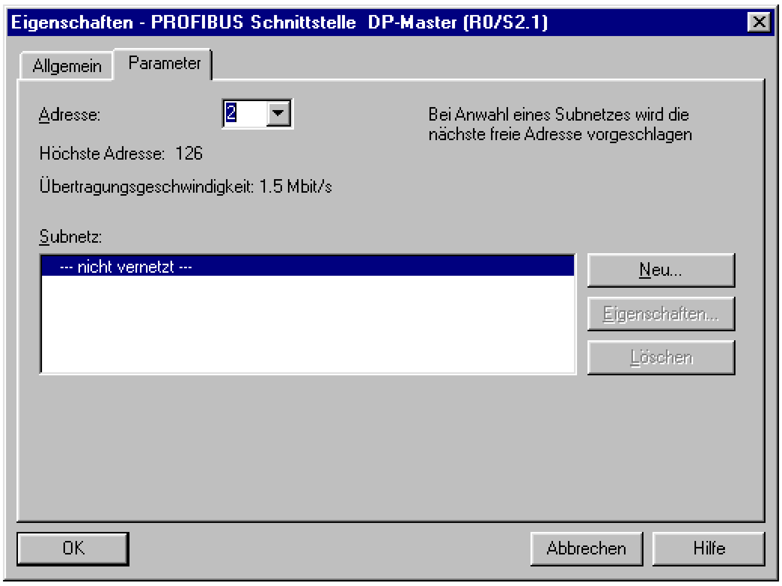
(0) UR

Steckplatz	Baugruppe	Bestellnummer	MPI-Adresse	E-Adresse	A-Adresse	Kommentar
1	PS 307 2A	6ES7 307-1BA00-0AA0				
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						

6ES7 307-1BA00-0AA0
Laststromversorgung 120/230V
AC:24VDC/2A

Änd

- Now the SPS controller has to be inserted in the second channel of the Profile Rail Element. Therefore open the directory "SIMATIC-300" and drag the corresponding controller into the second line of the Profile Rail Element.
- When the SPS controller is dropped into the second line of the Profile Rail Element the following window will be opened.



Adresse: 2

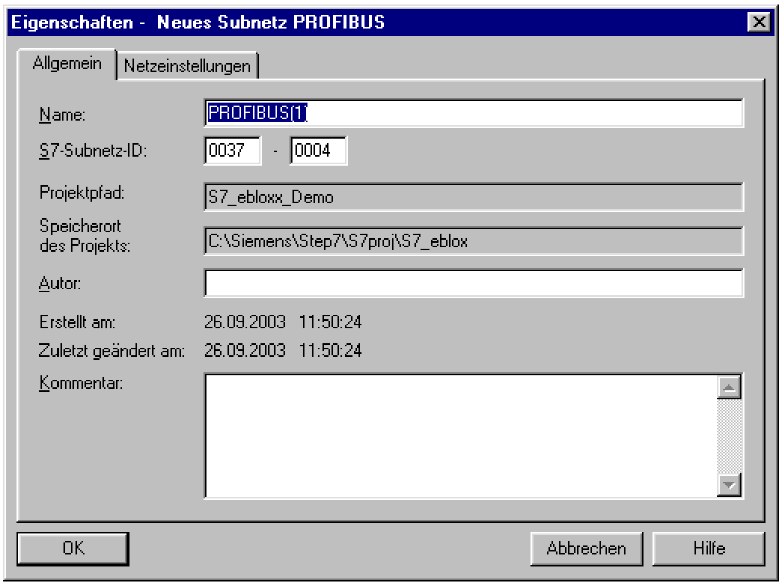
Bei Anwahl eines Subnetzes wird die nächste freie Adresse vorgeschlagen

Höchste Adresse: 126

Übertragungsgeschwindigkeit: 1.5 Mbit/s

Subnetz:
--- nicht vernetzt ---

- In this window change to the register card "Parameters" and press the button "New…". The following window will be opened.



Name: PROFIBUS(1)

S7-Subnetz-ID: 0037 - 0004

Projektpfad: S7_ebloxx_Demo

Speicherort des Projekts: C:\Siemens\Step7\S7proj\S7_eblox

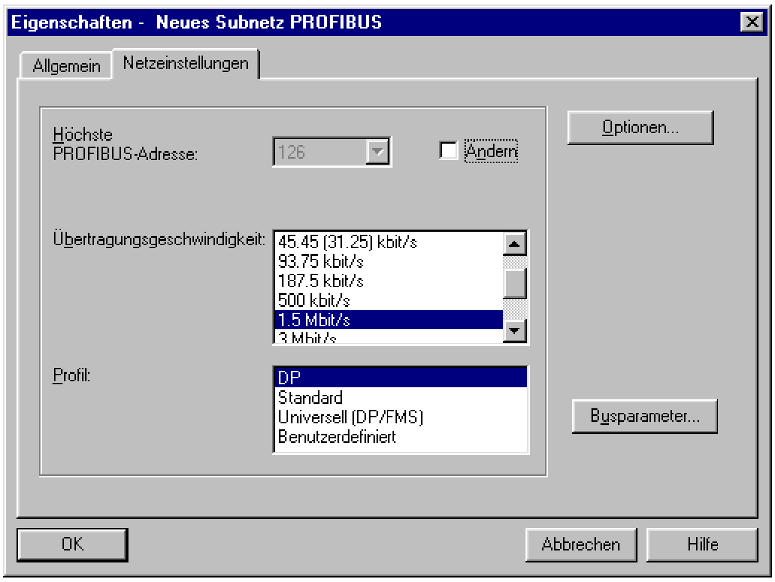
Autor:

Erstellt am: 26.09.2003 11:50:24

Zuletzt geändert am: 26.09.2003 11:50:24

Kommentar:

- Go to the register card "Network Settings".

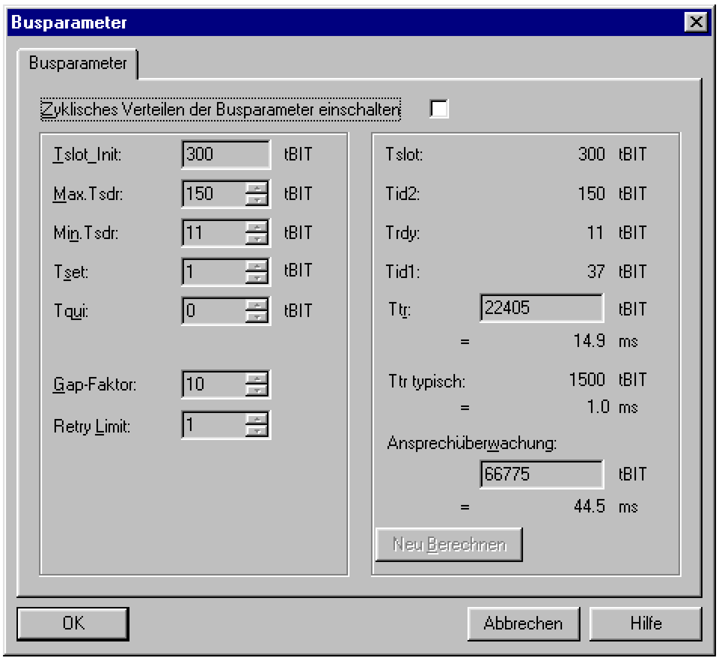

Eigenschaften - Neues Subnetz PROFIBUS

Höchste PROFIBUS-Adresse: 126 □ Ändern

Übertragungsgeschwindigkeit: 45.45 (31.25) kbit/s, 93.75 kbit/s, 187.5 kbit/s, 500 kbit/s, **1.5 Mbit/s**, 3 Mbit/s

Profil: **DP**, Standard, Universell (DP/FMS), Benutzerdefiniert

- Here you have to select the Baud Rate and Profile. Next click on the button "Bus Parameters…". The following window will be opened.

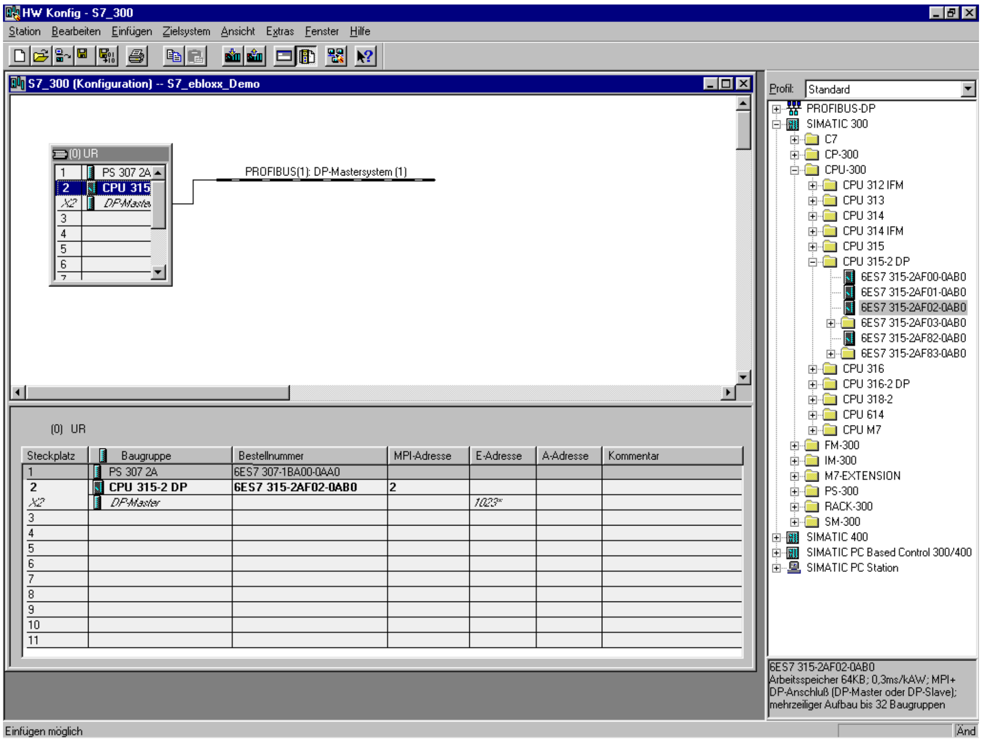

Busparameter

Zyklisches Verteilen der Busparameter einschalten □

Tslot_Init:	300	tBIT
Max.Tsdr:	150	tBIT
Min.Tsdr:	11	tBIT
Tset:	1	tBIT
Tqui:	0	tBIT
Gap-Faktor:	10	
Retry Limit:	1	

Tslot:	300	tBIT
Tid2:	150	tBIT
Trdy:	11	tBIT
Tid1:	37	tBIT
Ttr:	22405	tBIT
=	14.9	ms
Ttr typisch:	1500	tBIT
=	1.0	ms
Ansprechüberwachung:	66775	tBIT
=	44.5	ms

- Here you see the timing settings of the PROFIBUS-DP interface.
  Note: The option "activate cyclic distribution of pus parameters" must not be activated (no check mark). This option may only be used in a pure Siemens$^{®}$-DP system with special care.
- Now confirm all settings with "OK". The main window of the SIMATIC Manager will be shown again.



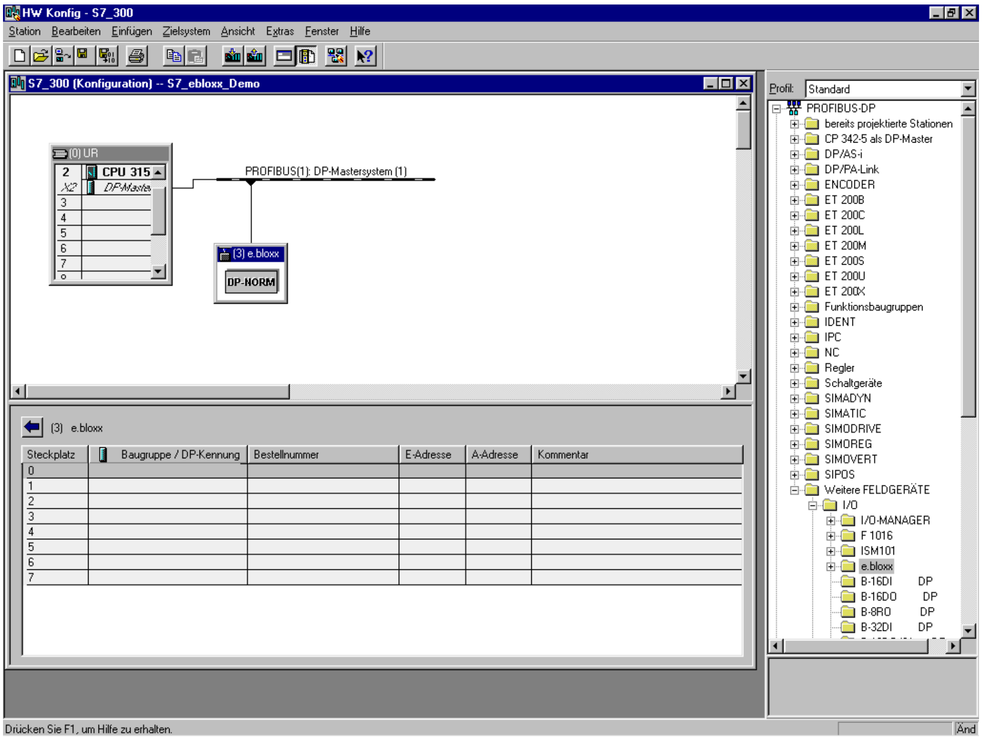
(0) UR

Steckplatz	Baugruppe	Bestellnummer	MPI-Adresse	E-Adresse	A-Adresse	Kommentar
1	PS 307 2A	6ES7 307-1BA00-0AA0				
2	CPU 315-2 DP	6ES7 315-2AF02-0AB0	2			
X2	DP-Master			1023*		
3						
4						
5						
6						
7						
8						
9						
10						
11						

6ES7 315-2AF02-0AB0
Arbeitsspeicher 64KB; 0,3ms/kAW; MPI+ DP-Anschluß (DP-Master oder DP-Slave); mehrzeiliger Aufbau bis 32 Baugruppen

Einfügen möglich

- Now the e.bloxx slaves must be attached to the PROFIBUS-DP line in the main window. Therefore open the directory "PROFIBUS-DP" on the right side of the window. In the sub-directories "Further Fieldbus Devices" -> "I/O" click on "e.bloxx" and drag that entry onto the PROFIBUS-DP line in the main window. A small box will be shown for the e.bloxx and it will be attached automatically to the PROFIBUS-DP line.

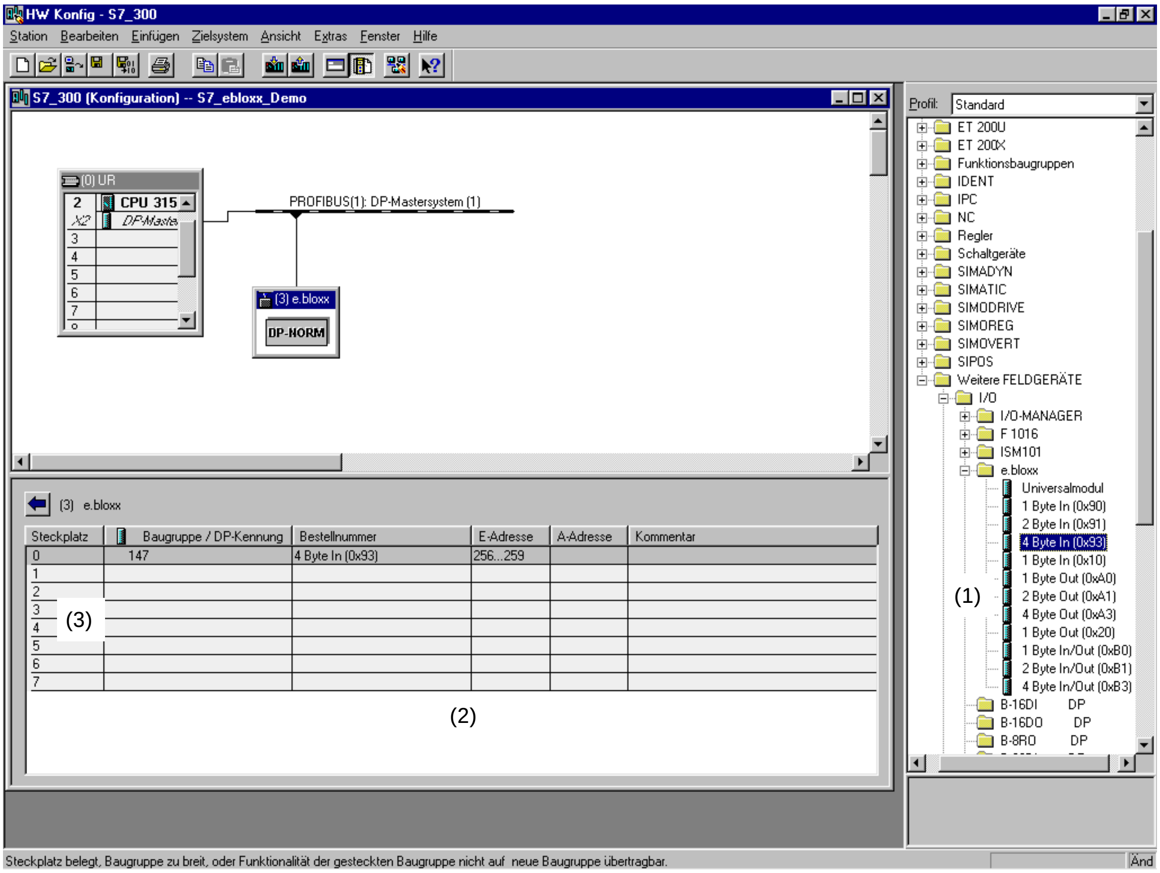


(3) e.bloxx

Steckplatz	Baugruppe / DP-Kennung	Bestellnummer	E-Adresse	A-Adresse	Kommentar
0					
1					
2					
3					
4					
5					
6					
7					

Drücken Sie F1, um Hilfe zu erhalten.

- Now the e.bloxx module must be configured. Therefore open the directory "PROFIBUS-DP" and then go to the sub-directories "Further Fieldbus Devices" -> "I/O" -> "e.bloxx". There you see the different possible configuration values (1).

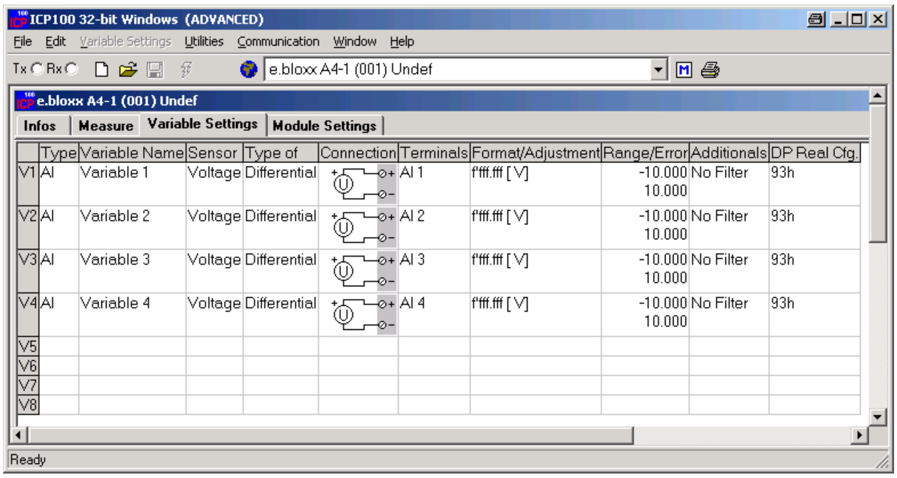


(3) e.bloxx

Steckplatz	Baugruppe / DP-Kennung	Bestellnummer	E-Adresse	A-Adresse	Kommentar
0	147	4 Byte In (0x93)	256...259		
1					
2					
3					
4					
5					
6					
7					

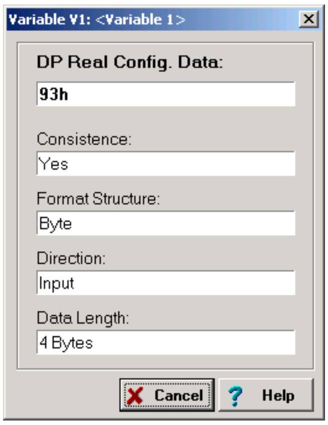
Steckplatz belegt, Baugruppe zu breit, oder Funktionalität der gesteckten Baugruppe nicht auf neue Baugruppe übertragbar.

- The values must be assigned to the correct slots in the table (2) at the bottom of the main window. Therefore drag the corresponding configuration values to the corresponding slots (3). Slot 0 indicated channel 1, slot 1 indicated channel 2, etc..
- The configuration of the channels of an e.bloxx module is done in the Configuration Table of the *Configuration Software ICP 100* (see next picture).


ICP100 32-bit Windows (ADVANCED)
File Edit Variable Settings Utilities Communication Window Help
Tx ○ Rx ○
e.bloxx A4-1 (001) Undef
e.bloxx A4-1 (001) Undef
Infos Measure Variable Settings Module Settings
Ready

	Type	Variable Name	Sensor	Type of	Connection	Terminals	Format/Adjustment	Range/Error	Additionals	DP Real Cfg.
V1	AI	Variable 1	Voltage	Differential		AI 1	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V2	AI	Variable 2	Voltage	Differential		AI 2	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V3	AI	Variable 3	Voltage	Differential		AI 3	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V4	AI	Variable 4	Voltage	Differential		AI 4	f'fff.fff [V]	-10.000 10.000	No Filter	93h
V5										
V6										
V7										
V8										

- In this Configuration Table each variable has a "DP Real Cfg." field. These are the values which have to be the assigned to the corresponding slots in the SIMATIC Master.
- If you click with the right mouse button on the field "DP Real Cfg." of a variable in the Configuration Table the following window will open.



DP Real Config. Data: 93h

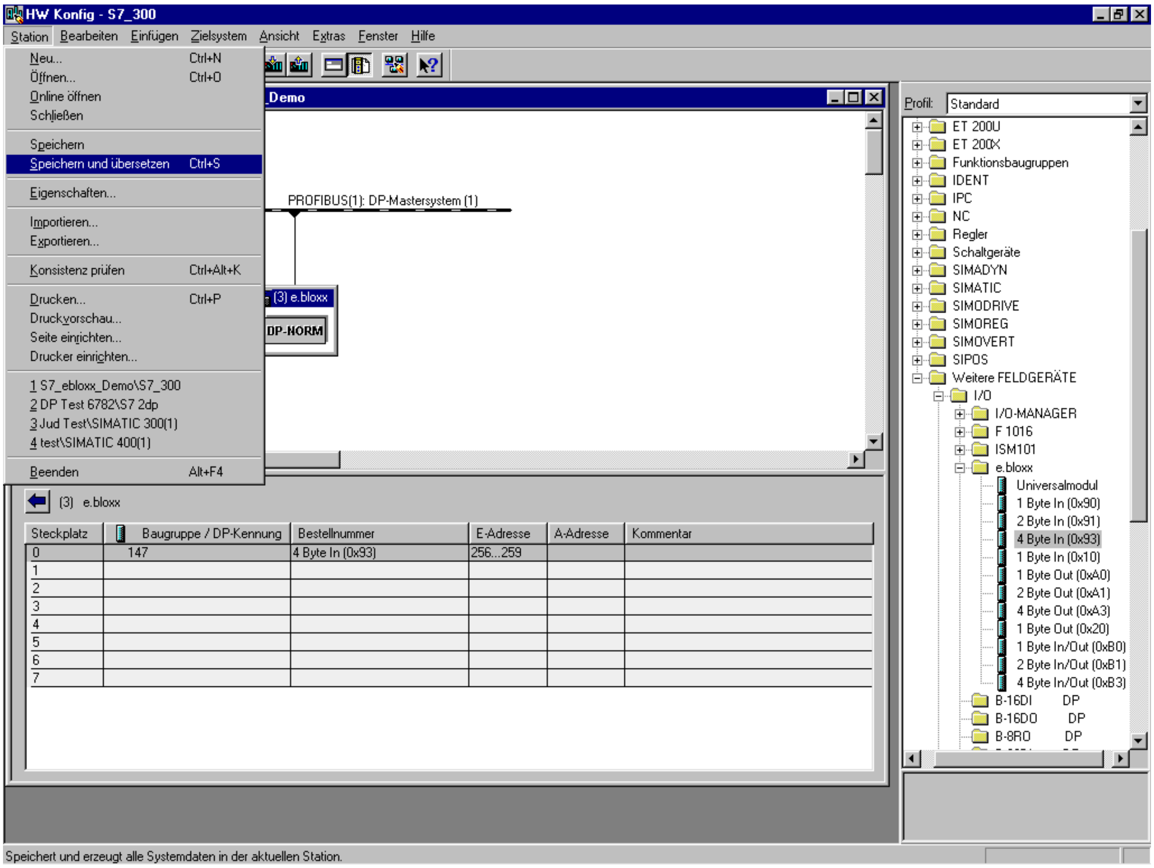
Consistence: Yes

Format Structure: Byte

Direction: Input

Data Length: 4 Bytes

- Here you can see the DP-settings in detail.
- Configure every e.bloxx slave as described above.
- Now you have to save and translate the defined configuration. Therefore select the menu item "Save and Translate" in the menu "Station".

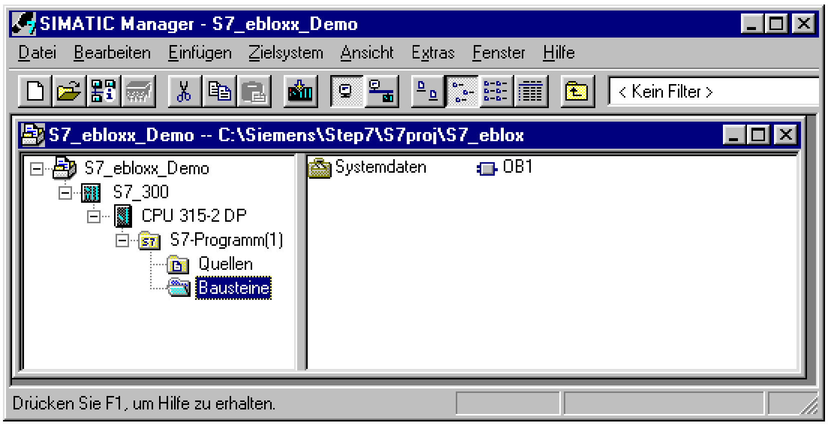


Steckplatz	Baugruppe / DP-Kennung	Bestellnummer	E-Adresse	A-Adresse	Kommentar
0	147	4 Byte In (0x93)	256...259		
1					
2					
3					
4					
5					
6					
7					

Speichert und erzeugt alle Systemdaten in der aktuellen Station.

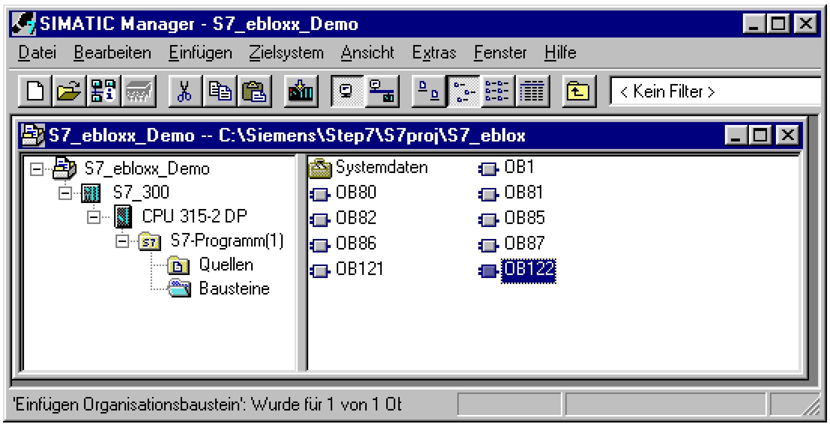
- You get back to the hardware configuration window of the SIMATIC Manager.

- Now select the "Components" level.


SIMATIC Manager - S7_ebloxx_Demo
Datei Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe
< Kein Filter >
S7_ebloxx_Demo -- C:\Siemens\Step7\S7proj\S7_eblox
S7_ebloxx_Demo
S7_300
CPU 315-2 DP
S7-Programm(1)
Quellen
Bausteine
Systemdaten
OB1

Drücken Sie F1, um Hilfe zu erhalten.

- Copy the Component "OB1" eight times. The components are necessary for the SPS to communicate via PROFIBUS-DP. The copies must have the names "OB80", "OB81", "OB82", "OB85", "OB86", "OB87", "OB121" and "OB122".


SIMATIC Manager - S7_ebloxx_Demo
Datei Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe
< Kein Filter >
S7_ebloxx_Demo -- C:\Siemens\Step7\S7proj\S7_eblox
S7_ebloxx_Demo
S7_300
CPU 315-2 DP
S7-Programm(1)
Quellen
Bausteine
Systemdaten
OB1
OB80
OB81
OB82
OB85
OB86
OB87
OB121
OB122

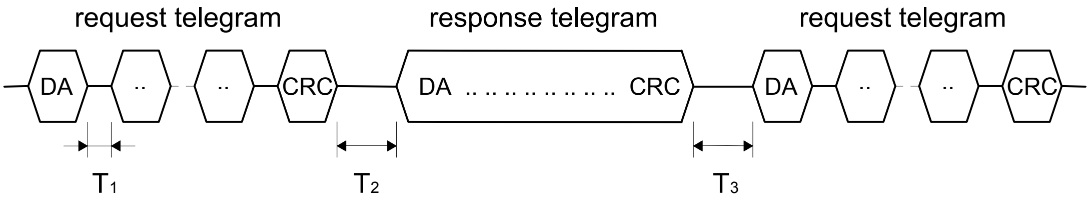
'Einfügen Organisationsbaustein': Wurde für 1 von 1 Ol

- With the button "Load" the complete SIMATEC project must now be sent to the Siemens$^{®}$ S7-300.

5. MODBUS-PROTOCOL

5.1. Transmission Sequence

In the MODBUS-protocol the data are transmitted from and to the sensor module by means of the following sequence:



T_1: time between two characters
T_2: time between request-telegram and corresponding response-telegram
T_3: time between response-telegram and next request-telegram

You will find the minimum and maximum appearing values for T_1, T_2 and T_3 and the adjustment range in the following table 2.14.

protocol	baud rate	T_{1min}	T_{1max}	T_{2min}	T_{2max}	T_{3min}	T_{3max}
adjustable		no	no	yes	no	no	yes
M O D B U S	19,200 bps 38,400 bps 57,600 bps 115,200 bps	0	1.5 CT	3.5 CT	T_{2min} x 1.2	3.5 CT	0.1 sec to 600 sec

Table 5.1 *- Values and adjustment range for the times T1, T2 and T3*
(CT: character time: 1 CT = character length [bit] / baud rate [bps])

Notice: In the MODBUS-protocol T_{2max} lasts at least 12 msec.

The values for T_{2min} and T_{3max} and the behavior of the e.bloxx if T_{3max} is exceeded (communication timeout) can be adjusted by means of the Configuration Software ICP100. The default value for T_{2min} is 1 CT and for T_{3max} it is 60 sec.

5.2. Telegram Format

Request Telegram

Idle Interval	ADR	FNR	Function Parameters / Data	CRC
> 3.5 CT	1 Byte	1 Byte	n Byte	2 Byte

Response Telegram

Idle Interval	ADR	FNR	Function Parameters / Data	CRC
> 3.5 CT	1 Byte	1 Byte	n Byte	2 Byte

The request and response telegrams in the RTU-mode used by the e.bloxx modules are starting with an idle-interval of at least 3.5 character length. Most simple this will be performed by waiting for at least 4 character-times (CT) after receiving the last character of a telegram. The telegrams have no Start-Delimiter and no End-Delimiter too. The first field after that idle-interval is the ISM-Address (ADR) followed by the function number (FNR) and the function parameters or data respectively. At the end the telegrams contain a check sum (CRC) with the length of 16 bits. The check sum is calculated from the whole telegram without the CRC itself. The CRC-polynomial is: $u^{15} + u^{13} + 1$. The start value is *hex* FFFF.

5.3. Instruction Set

With the MODBUS-protocol the data are read and written via register accesses. The following register accesses are defined for the communication with the sensor modules:

Function Number	**Function**
03 hex	Read holding register (read/write register)
04 hex	Read input register (read only register)
06 hex	Preset single register
08 hex	Diagnostic
10 hex	Preset multiple register

***Table 5.2** - MODBUS commands supported by the e.bloxx modules*

Read Holding Register

Description:
With this command input/output registers (read/write registers) can be read.

Request Telegram

ADR	FNR	REGSTA		REGNUM		CRC	
	03	MSB	LSB	MSB	LSB	MSB	LSB

Response Telegram

ADR	FNR	REGSTA		REGNUM		CRC	
	03	MSB	LSB	MSB	LSB	MSB	LSB

ADR ....ISM address (*hex* 00..7F)
FNR ....Function number (*hex* 03)
REGSTA ....Address of the first register to be read
REGNUM ...Number of registers to be read
BYTNUM ....Number of databytes (max. 64)
D0 - Dn ....Data bytes (max. 64)
CRC ....Check sum

CRC polynomial: $u^{15} + u^{13} + 1$
CRC start value: *hex* FFFF

Read Input Register

Description:
With this command input registers (read only registers) can be read.

Request Telegram

ADR	FNR **04**	REGSTA MSB	 LSB	REGNUM MSB	 LSB	CRC MSB	 LSB

Response Telegram

ADR	FNR **04**	BYTNUM	D0	D1	...	Dn	CRC MSB	 LSB

ADR .... ISM address (*hex* 00..7F)
FNR .... Function number (*hex* 04)
REGSTA .... Address of the first register to be read
REGNUM .. Number of registers to be read
BYTNUM ... Number of databytes (max. 64)
D0 - Dn ...... Data bytes (max. 64)
CRC .... Check sum

CRC polynom: $u^{15} + u^{13} + 1$
CRC start value: *hex* FFFF

Preset Single Register

Description:
With this command a single register can be written.

Request Telegram

ADR	FNR	REGADR		DATA		CRC	
	06	MSB	LSB	MSB	LSB	MSB	LSB

Response Telegram

ADR	FNR	REGADR		DATA		CRC	
	06	MSB	LSB	MSB	LSB	MSB	LSB

ADR ....ISM address (*hex* 00..7F)
FNR ....Function number (*hex* 06)
REGADR ....Address of the register to be written
DATA ....Data word (*hex* 0000..FFFF)
CRC ....Check sum

CRC polynomial: $u^{15} + u^{13} + 1$
CRC start value: *hex* FFFF

Diagnostic

Description:
With this command a diagnostic telegram will be sent to the sensor module. If the telegram has been received in correct form the module will send this telegram back unchanged (echo telegram).

Request Telegram

ADR	FNR	SUBFCT		DATA		CRC	
	08	**00**	**00**	**A5**	**37**	MSB	LSB

Response Telegram

ADR	FNR	SUBFCT		DATA		CRC	
	08	**00**	**00**	**A5**	**37**	MSB	LSB

ADR .... ISM address (*hex* 00..7F)
FNR .... Function number (*hex* 08)
SUBFCT .... Subfunction number (*hex* 0000)
DATA .... Data word (*hex* A537)
CRC .... Check sum

CRC polynomial: $u^{15} + u^{13} + 1$
CRC start value: *hex* FFFF

Preset Multiple Registers

Description:
With this command a large, continuous field of registers can be written.

Request Telegram

ADR	FNR	REGSTA		REGNUM		BYTNUM	D0	D1	...	Dn	CRC	
	10	MSB	LSB	MSB	LSB						MSB	LSB

Response Telegram

ADR	FNR	REGSTA		REGNUM		CRC	
	10	MSB	LSB	MSB	LSB	MSB	LSB

ADR ....ISM address (*hex* 00..7F)
FNR ....Function number (*hex* 10)
REGSTA ....Address of the first register to be written
REGNUM ...Number of registers to be written
BYTNUM ....Number of databytes (max. 64)
D0 - Dn ....Data bytes (max. 64)
CRC ....Check sum

CRC polynomial: $u^{15} + u^{13} + 1$
CRC start value: *hex* FFFF

5.4. Register Contents

Variable Values In Integer Format

Register	Type	Content	Range
0000	ro/rw	variable 1 integer value	-32768 ... 32767
0001	ro/rw	variable 2 integer value	-32768 ... 32767
:	:	:	:
000B	ro/rw	variable 8 integer value	-32768 ... 32767
(000F	ro/rw	variable 16 integer value	-32768 ... 32767)*

* ... Only e.bloxx A6-2CF, which has 16 variables.

Read And Write Variable (Real)

Register	Type	Content	Range
0010	ro/rw	variable 1 real value high word	0 ... 65535
0011	ro/rw	variable 1 real value low word	0 ... 65535
0012	ro/rw	variable 2 real value high word	0 ... 65535
0013	ro/rw	variable 2 real value low word	0 ... 65535
:	:	:	:
001E	ro/rw	variable 8 real value high word	0 ... 65535
001F	ro/rw	variable 8 real value low word	0 ... 65535
(002E	ro/rw	variable 16 real value high word	0 ... 65535)*
(002F	ro/rw	variable 16 real value low word	0 ... 65535)*

* ... Only e.bloxx A6-2CF, which has 16 variables.

Attention: The low word and the high word of a variable always have to be read or written simultaneously.

Notice: The possibility of a writing command on the registers 0000 up to 002F depends on the configuration. With the following variable types a writing command is valid if this has been allowed by the Configuration Software ICP100.

Analog Input with Tare Function:
After a writing command for this variable the tare function will be started.

Digital Counter with Reset Function:
After a writing command for this variable the counter will be set to zero.

Arithmetic Variable with min/max-Function and Reset Function:
After a writing command for this variable the pull-pointer will be reset.

Setpoint Variable:
After a writing command for this variable the new set value will be taken over.

Digital Output Variable (Host Output):
A writing command for this variable will set the corresponding variable to '1' or '0' respectively.

Variable Information

Register	Type	Content	Length
1000	ro	variable 1 variable type	2 byte
1001	ro	variable 1 measuring principle	2 byte
1002	ro	variable 1 field length	2 byte
1003	ro	variable 1 decimals	2 byte
1004	ro	variable 1 tare/reset	2 byte
1005..1007	ro	variable 1 units	4 char
1008..1011	ro	variable 1 variable name	20 char
1012..101F		reserve	

1000..101F	variable information for variable 1
1020..103F	variable information for variable 2
1040..105F	variable information for variable 3
1060..107F	variable information for variable 4
1080..109F	variable information for variable 5
10A0..10BF	variable information for variable 6
10C0..10DF	variable information for variable 7
10E0..10FF	variable information for variable 8
1100..111F	variable information for variable 9
1120..113F	variable information for variable 10
1140..115F	variable information for variable 11
1160..117F	variable information for variable 12
1180..119F	variable information for variable 13
11A0..11BF	variable information for variable 14
11C0..11DF	variable information for variable 15
11E0..11FF	variable information for variable 16

Coding <variable type>:

hex 0	Empty Variable (LE)
hex 1	Analog Input Variable (AI)
hex 2	Arithmetic Variable (AR)
hex 3	Digital Output Variable (DO)
hex 4	Digital Input Variable (DI)
hex 5	Setpoint Variable (VO)
hex 6	Alarm Variable (AL)
hex 9	Controller Variable (CO)

Coding <measuring principle>:

Analog Input:

hex 0	voltage single-ended
hex 1	voltage differential
hex 2	current
hex 3	resistance 2-wire technique
hex 4	resistance 3-wire technique
hex 5	resistance 4-wire technique
hex 6	bridge 4-wire technique
hex 7	bridge 6-wire technique
hex 8	thermocouple internal
hex 9	thermocouple external
hex A	cold junction terminal 1
hex B	potentiometric

hex C	cold junction terminal 2
hex D	cold junction terminal 3
hex E	cold junction terminal 4

Digital Input:

hex 0	no
hex 1	host input
hex 2	frequency
hex 3	progressive counter
hex 4	quadrature counter
hex 5	up/down counter

Digital Output:

hex 0	no
hex 1	host output
hex 2	PWM output
hex 3	process output

Coding <tare/reset>:

hex 0	no tare/reset
hex 1	tare/reset valid

Device Information

Register	Type	Content	Length
0300	ro	number of variables	2 byte
0301..0303	ro	serial number	6 char
0304..030D	ro	location	20 char

Device Identification

Register	Type	Content	Length
0400..XXXX	ro	vendor name ("Gantner,")	1..32 char
XXXX..XXXX	ro	model name (<type of module>,)	1..32 char
XXXX..XXXX	ro	hw version ("xy.yy,")	1..8 char
XXXX..XXXX	ro	sw version ("xy.yy,")	1..8 char

status information

Register	Type	Content	Length
0500	ro	module status	2 byte
0501	ro	variable status	2 byte

<module st.>	= M16..M13	M12..M9	M8..M5	M4..M1	= *hex* 0XYZ
	0	X	Y	Z	*ASCII* "0XYZ"
<variable st.>	= K16..K13	K12..K9	K8..K5	K4..K1	= *hex* 0XYZ
	0	X	Y	Z	*ASCII* "0XYZ"

If the bit Mn in the module status is set it indicates that an error has occurred in the sensor module. Valid is:

M1 = 1:	EEPROM - Error
M2 = 1:	FLASH - Error
M3 = 1:	ADC - Error
M4 = 1:	Configurations - Error

If the bit Kn in the variable status is set it indicates that an error has occurred in variable n. A variable error is given when the measuring value is outside of the linearization, e.g. in consequence of a sensor break down or of a short circuit of transmission.

6. LOCAL-BUS PROTOCOL

6.1. General

The protocol described in this paper is used for the communication with e.bloxx devices:
- e.bloxx A1-x
- e.bloxx A4-x
- e.bloxx A5-1
- e.bloxx A9-1
- e.bloxx A6-2CF
- e.bloxx D1-x

6.2. Character Format

The general character format for the LOCAL-BUS will be 8e1 (1 Start Bit / 8 Data Bits / Even Parity / 1 Stop Bit).

6.3. Transmission Order

The transmission order is MS-Byte for LSB-Byte with LS-Bit for MS-Bit

6.4. Primary Build-Up Of The Frame

6.4.1. Request Frame

SYNC2	ADDR	L	CMDXX	RequDATA	FCS

SYNC2 .... Start delimiter for addressed communication: **0xA6** CHAR
ADDR .... Address of the slave: 0x01 ... 0x7F CHAR
L .... Length of the flowing Data: CHAR
L = (CMDXX + RequDATA[...])
L = [0x01 ... 0xFF]
CMDXX: ..... The command specifies the sense of data which are following: 0x00 ... 0xFF CHAR
RequDATA: Request data string: 0x00 ... 0xFE character CHAR STRING
FCS: .... Frame check sequence of the frame: CHAR
FCS = (Addr + L + CMDXX + RequDATA[...]) mod 256

6.4.2. Response Frame

Positive Without Data

SQ

SQ .... Short quit; one single character: **0xE5** CHAR

Positive With Data

ACK	ADDR	L	RespDATA	FCS

ACK.............Start delimiter for a positive response with data: 0xB6 CHAR
ADDR..........Address of the slave: 0x01 ... 0x7F CHAR
L ....Length of the flowing Data: L = RespData[...] CHAR
L = [0x01 ... 0xFF]
RespDATA..Response data string: 0x00 ... 0xFF character CHAR STRING
FCS.............Frame check sequence of the frame:
FCS = (Addr + L + RequData[...]) mod 256 CHAR

Negative

ACK	ADDR	L	ERROR-CODE	FCS

NAK.............Start delimiter for a negative response with data: 0xC6 CHAR
ADDR..........Address of the slave: 0x01 ... 0x7F CHAR
L ....Length of the flowing Data: L = ErrorCode[...] CHAR
L = [0x01 ... 0xFF]
ERROR-CODE Error code of the responding slave: CHAR
0x01: Command not available
0x02: Invalid parameter received
FCS.............Frame check sequence of the frame: CHAR
FCS = (Addr + L + ErrorCode) mod 256

6.5. Data Format Coding

CHAR.................. 0 .. 255
BOOLEAN........... FALSE = 0; TRUE <> 0
INTEGER .... -32768 … + 32767
LONGINTEGER.. -2E32 ... 2E32-1
FLOAT .... IEEE 754 (4 Byte): ±1.4E-45 ... ±3.4E38
CHAR STRING ... a different number of following CHAR

6.6. Baud Rate

For communication the baud rates 19.2 kbit/s, 38.4 kbit/s, 57.6 kbit/s, 93.75 kbit/s, 115.2 kbit/s, 187.5 kBit, 500 kbit/s and **1.5 Mbit/s** are available.

6.7. Communication Procedure

The communication will always be initiated by a master system (PC, e.gate 01, ...). The slave will answer in a proper way with a defined maximum answer delay (AnswerDelay). The minimum answer delay is one character time (e.g.: 9.2 kbit/s, 8e1, one character equal 11 bit -> 1.2 ms)

Each request must be answered with a response. If no response will be received within a certain time (maximum answer delay) the master must recognize it and start a user defined error procedure.

An incorrect command – e.g. command not available - for the slave will be answered with an NAK.

6.8. Command Table

The command table is showing the general used commands of the e.bloxx system.

Command	Type	Command Number	Description
GetDiag	M	2 (0x02)	Get diagnostic
GetAllVar	M	10 (0x0A)	Get all variables
GetSingleVar	M	11 (0x0B)	Get single variable
SetSingleVar	M	12 (0x0C)	Set single variable
GetDeviceIdent	M	13 (0x0D)	Get device identification
GetSingleVarEx	M	20 (0x14)	Get single variable extended
SetSingleVarEx	M	21 (0x15)	Set single variable extended

Table 6.1 *- Command Table*

Type: M ... Mandatory
O ... Obligatory

Get Diagnostic Data (GetDiag)

Description:
This command is used to read the actual diagnostic data of an e.bloxx equipment.

Request Telegram

SYNC2	ADDR	L	CMD2	FCS

L ....Length of the data: 0x01 CHAR
CMD2..........Command 2 "Get Diagnostic Data": 0x02 CHAR
FCS............Frame check sequence of the request CHAR

Response Telegrams

Acknowledge:

ACK	ADDR	L	DiagData [...]	FCS

L ....Length of the data: min 0x02 CHAR
DiagData .....Diagnostic data: CHAR STRING
DiagData[0] = Module State MSB
DiagData[1] = Module State LSB
DiagData[3] = Variable State 15 … 8
DiagData[4] = Variable State 7 … 0
…
DiagData[n] = Variable State x … y
If a bit is set, an error is available for this variable.
FCS............Frame check sequence of the ACK response CHAR

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L ....Length of the data: 0x01 CHAR
ErrorCode ...Command not available/invalid: 0x01 CHAR
FCS............Frame check sequence of the NAK response CHAR

Get Variable All (GetVarAll)

Description:
This command is used to read the actual value of all variables of an e.bloxx equipment.

Request Telegram

SYNC2	ADDR	L	CMD10	FCS

L	Length of the data: 0x01	CHAR
CMD10	Command 10 "Get Variable All": 0x0A	CHAR
FCS	Frame check sequence of the request	CHAR

Response Telegrams

Acknowledge:

ACK	ADDR	L	VALUE [0]	VALUE [1]	...	VALUE [n]	FCS

L	Length of the data: 0x01 ... 0xFF	CHAR
VALUE [0]	Values 0 to be transmitted	BOOLEAN / INTEGER / FLOAT
...		
VALUE [n]	Values n to be transmitted n depends on the e.bloxx equipment e.bloxx A1-x: max. n = 0x07 e.bloxx A4-x: max. n = 0x07 e.bloxx A5-1: max. n = 0x07 e.bloxx A6-2CF: max. n = 0x0F e.bloxx A9-1: max. n = 0x07 e.bloxx D1-x: max. n = 0x07	BOOLEAN / INTEGER / FLOAT
FCS	Frame check sequence of the ACK response	CHAR

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L	Length of the data: 0x01	CHAR
ErrorCode	Command not available/invalid: 0x01	CHAR
FCS	Frame check sequence of the NAK response	CHAR

Get Single Variable (GetSingleVar)

Description:
This command is used to read the actual value of one defined variable of an e.bloxx equipment.

Request Telegram

SYNC2	ADDR	L	CMD11	VarIndex	FCS

L	Length of the data: 0x01	CHAR
CMD11........	Command 11 "Get Single Variable": 0x0B	CHAR
VarIndex......	Variable number to be read: 0x00 ... n	CHAR
	e.bloxx A1-x: max. n = 0x07	
	e.bloxx A4-x: max. n = 0x07	
	e.bloxx A5-1: max. n = 0x07	
	e.bloxx A6-2CF: max. n = 0x0F	
	e.bloxx A9-1: max. n = 0x07	
	e.bloxx D1-x: max. n = 0x07	
FCS.............	Frame check sequence of the request	CHAR

Response Telegrams

Acknowledge:

ACK	ADDR	L	VALUE	FCS

L	Length of the data: 0x01 (BOOLEAN) up to 0x04 (FLOAT)	CHAR
VALUE	The value to be transmitted	BOOLEAN / INTEGER / FLOAT
FCS.............	Frame check sequence of the ACK response	CHAR

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L	Length of the data: 0x01	CHAR
ErrorCode ...	Command not available/invalid: 0x01	CHAR
FCS.............	Frame check sequence of the NAK response	CHAR

Set Single Variable (SetSingleVar)

Description:
This command is used to set the value of one defined variable of an e.bloxx equipment.

Request Telegram

SYNC2	ADDR	L	CMD12	VarIndex	VALUE	FCS

L	Length of the data: 0x03 up to 0x06 The length depends on the data format of the value	CHAR
CMD12	Command 12 "Set Single Variable": 0x0C	CHAR
VarIndex	Variable number to be set: 0x00 ... n e.bloxx A1-x: max. n = 0x07 e.bloxx A4-x: max. n = 0x07 e.bloxx A5-1: max. n = 0x07 e.bloxx A6-2CF: max. n = 0x0F e.bloxx A9-1: max. n = 0x07 e.bloxx D1-x: max. n = 0x07	CHAR
VALUE........	The value to be set	BOOLEAN / INTEGER / FLOAT
FCS	Frame check sequence of the request	CHAR

Response Telegrams

Acknowledge:

SQ

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L	Length of the data: 0x01	CHAR
ErrorCode...	Command not available/invalid: 0x01	CHAR
FCS	Frame check sequence of the NAK response	CHAR

Get Device Identification (GetDevIdent)

Description:
This command is used to read the device identification parameters of an e.bloxx equipment.

Request Telegram

SYNC2	ADDR	L	CMD13	FCS

L	Length of the data: 0x01	CHAR
CMD13	Command 13 "Get Device Identification": 0x0D	CHAR
FCS	Frame check sequence of the request	CHAR

Response Telegrams

Acknowledge:

ACK	ADDR	L	L1	VENDOR	L2	DEVTYPE	L3	HWR	L4	SWR	FCS

L	Length of the data: max. 0xFF	CHAR
L1	Length of the vendor string: max. 0x14	CHAR
VENDOR	The vendor of the slave: e.g. "Gantner"	CHAR STRING
L2	Length of the device type string: max. 0x14	CHAR
DEVTYPE	The type of the device: e.g. "e.bloxx A1-1"	CHAR STRING
L3	Length of the hardware release string: max. 0x05	CHAR
HWR	The hardware release string in the format "XX.XX": e.g. "x0.06"	CHAR STRING
L3	Length of the firmware release string: max. 0x05	CHAR
SWR	The firmware release string in the format "YY.YY": e.g. "a1.00"	CHAR STRING
FCS	Frame check sequence of the request	CHAR

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L	Length of the data: 0x01	CHAR
ErrorCode	Command not available/invalid: 0x01	CHAR
FCS	Frame check sequence of the NAK response	CHAR

Get Single Variable Extended (GetSingleVarEx)

Description:
This command is used to read special/extended values of an e.bloxx A6-2CF.

Request Telegram

SYNC2	ADDR	L	CMD20	VarIndex	SelIndex	FCS

L .... Length of the data: 0x03 CHAR
CMD20 .... Command 20 "Get Single Variable Extended": 0x14 CHAR
VarIndex ..... Variable number to be read: 0x00 ... n CHAR
e.bloxx A1-x: max. n = 0x07
e.bloxx A4-x: max. n = 0x07
e.bloxx A5-1: max. n = 0x07
e.bloxx A6-2CF: max. n = 0x0F
e.bloxx A9-1: max. n = 0x07
e.bloxx D1-x: max. n = 0x07
SelIndex ..... Selection index for the special values: CHAR
SelIndex_NetValue: 0
SelIndex_TareValue: 1
SelIndex_GrossValue: 2
SelIndex_ZeroValue: 3
SelIndex_UnbalancedVaule: 4
Meaning of the values:
UnbalancedValue + ZeroValue = GrossValue
GrossValue + TareValue = NetValue
FCS .... Frame check sequence of the request CHAR

Acknowledge:

ACK	ADDR	L	VALUE	FCS

L .... Length of the data: 0x01 (BOOLEAN) up to 0x04 (FLOAT) CHAR
VALUE........ The value to be transmitted BOOLEAN / INTEGER / FLOAT
FCS .... Frame check sequence of the ACK response CHAR

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L .... Length of the data: 0x01 CHAR
ErrorCode.... Command not available/invalid: 0x01 CHAR
FCS .... Frame check sequence of the NAK response CHAR

Set Single Variable Extended (SetSingleVarEx)

Description:
This command is used to set special/extended values of the e.bloxx A6-2CF.

Request Telegram

SYNC2	ADDR	L	CMD21	VarIndex	SelIndex	VALUE	FCS

L ....Length of the data: 0x03 up to 0x06 CHAR
The length depends on the data format of the value

CMD21........Command 21 "Set Single Variable Extended": 0x15 CHAR

VarIndex......Variable number to be set: 0x00 ... n CHAR

e.bloxx A1-x:	max. n = 0x07
e.bloxx A4-x:	max. n = 0x07
e.bloxx A5-1:	max. n = 0x07
e.bloxx A6-2CF:	max. n = 0x0F
e.bloxx A9-1:	max. n = 0x07
e.bloxx D1-x:	max. n = 0x07

SelIndex ......Selection index for the special values: CHAR

SelIndex_TareValue:	1
SelIndex_ZeroValue:	3

VALUE ....The value to be set BOOLEAN / INTEGER / FLOAT

FCS.............Frame check sequence of the request CHAR

Response Telegrams

Acknowledge:

SQ

Not Acknowledge:

NAK	ADDR	L	ErrorCode	FCS

L ....Length of the data: 0x01 CHAR

ErrorCode ...Command not available/invalid: 0x01 CHAR

FCS.............Frame check sequence of the NAK response CHAR

Notice:

Information in this manual are valid from June 17th 2005 until revocation.
Further changes and completion of the manual are reserved and possible without notice.

www.gantner-instruments.com

Gantner Instruments Test & Measurement GMBH
Montafonerstraße 8 • A-6780 Schruns/Austria
Tel.: +43 (0)5556-73784-410 • Fax: +43 (0)5556-73784-419
E-Mail: office@gantner-instruments.com

Gantner instruments