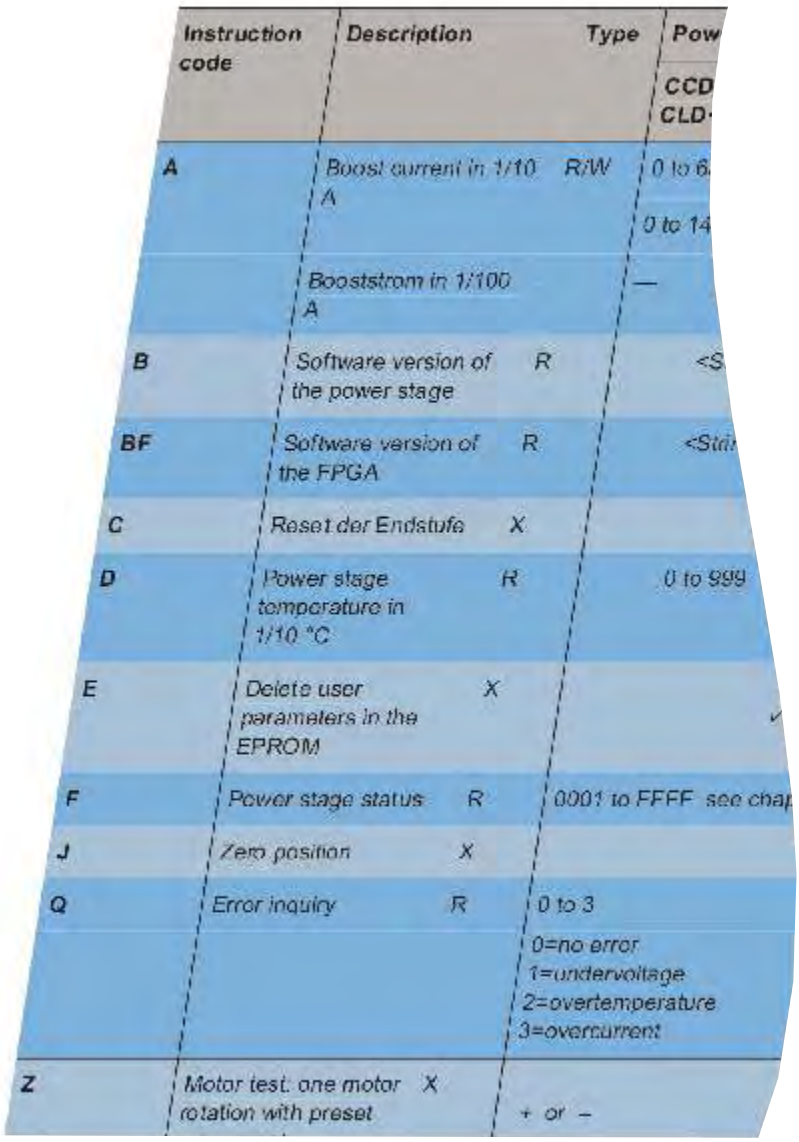
ServiceBus

phytron

Instruction Set for Stepper Motor Power Stages with ServiceBus



Instruction code	Description	Type	Pow
			CCD CLD
A	Boost current in 1/10 A	R/W	0 to 6 0 to 14
	Booststrom in 1/100 A		—
B	Software version of the power stage	R	<S
BF	Software version of the FPGA	R	<Stri
C	Reset der Endstufe	X	
D	Power stage temperature in 1/10 °C	R	0 to 999
E	Delete user parameters in the EPROM	X	✓
F	Power stage status	R	0001 to FFFF see chap
J	Zero position	X	
Q	Error inquiry	R	0 to 3 0=no error 1=undervoltage 2=overtemperature 3=overcurrent
Z	Motor test: one motor rotation with preset	X	+ or –

Manual 1250-A004 GB

customized solutions in motion

Instruction Set for

Stepper Motor Power Stages with ServiceBus

CCD+, CLD+ and ZMX+

Phytron GmbH

Industriestraße 12

82194 Gröbenzell, Germany

Tel.: +49(0)8142/503-0

Fax: +49(0)8142/503-190

Every possible care has been taken to ensure the accuracy of this technical manual. All information contained in this manual is correct to the best of our knowledge and belief but cannot be guaranteed. Furthermore we reserve the right to make improvements and enhancements to the manual and / or the devices described herein without prior notification.

We appreciate suggestions and criticisms for further improvement.

Please send your comments to the following e-mail-address: doku@phytron.de

You´ll find the updated version of this manual on the website of www.phytron.de

Contents

Manual ServiceBus

1 Introduction

1.1 General

All phytron devices and power stages **with ServiceBus** are labelled “+”.

The phytron power stages with the suffix + are particularly service-friendly over the **ServiceBus,** which accesses the power stage settings via PC.

Phytron’s devices can be activated by the **ServiceBus** via **RS 485** bus (SBM_{RS}) or **CAN** bus (SBM_{CAN}) depending on the power stage.

The free ServiceBus-Comm® for Windows® software simplifies the configuration, parameterization and monitoring of the power stage via PC (see ServiceBus-Comm manual).

Dependent on the power stage type, operating parameters like run current, stop current, step resolution, current delay time or more parameters dependent on the power stage type can be configured, saved and transmitted to the power stage by PC. Up to 16 stepper motor axes can be addressed via ServiceBus at the same time.

Individual solutions can also be done without ServiceBus-Comm® software. The ServiceBus instructions and functions can be implemented into the customer’s own environment as ASCII strings– e. g. with LabView, HyperTerminal or in C.

2 To Consider Before Installation

Read all manuals very carefully before installing and operating the device.
Observe the safety instructions in the following chapter!

2.1 Qualified Personnel

Design, installation and operation of systems using the ZMX may only be performed by qualified and trained personnel.

These persons should be able to recognize and handle risks emerging from electrical, mechanical or electronic system parts.

WARNING!

Damage to devices and injury might result by persons without proper training and qualifications!

2.2 Safety Instructions

1. Malfunction can occur during programming of the instruction codes, e.g. the connected motor starts in an uncontrolled way. Test the program sequence step by step!

2. Always switch off the supply voltage before connecting or disconnecting any wires or connectors at the ZMX.
   Do not unplug the connector while powered!
   Danger of electric arcing.

3. Up to 3 minutes after turning off the supply voltage, a hazardous voltage level is present at the motor connector and motor cable, even if the motor is not wired.
   Danger of electric shock!

4. The surface of the ZMX reaches temperatures of more than 85 °C during operation.
   Danger of injury if touching the surface!

3 RS Bus Configuration: SBM_{RS}

The bus connection (SBM_{RS}) is defined as follows:

- RS485: 4 wire connection, also point to point connection possible
- Signal input: R+ R-
- Signal output: T+ T-
- isolated from the motor voltage by means of optocoupler

For the serial ServiceBus connection of CCD+ and CLD+ is applied:
point to point operation by USB connection

A well-defined protocol should be followed to assure a safe data exchange:
Asynchronous transmission, 8 bits/byte, 1 stop bit, 1 parity bit
Transmission rate: 57600 bps
Permanent telegram format:
<STX> <address_H><address_L> <instruction> <value> : <csh> <csl> < ETX>

<STX>	Start-of-text character, 02_H
<address_H>	High-order byte of the power stage address 00_H...$0F_H$, 0_D…31_D; Admissible characters: 0…9 or A…F
<address_L>	Low-order byte of the power stage address 00_H...$0F_H$, 0_D…31_D; Admissible characters: 0…9 or A…F
<instruction>	Instruction byte: A....Z
<value>	Data byte (several)
:	Colon as separator, to distinguish between usable data and checksum
<csh>	Upper byte of the 8 bit checksum value (0...9/A...F)
<csl>	lower byte of the 8 bit checksum value (0...9/A...F)
< ETX>	End-of-text character, 03_H

The checksum is defined by summing up all bytes, beginning with the address byte and including the separator (:) in an exclusive-OR-operation (⊕):

`CS = address_H address_L ⊕ data byte 1 ⊕ data byte 2 . . . ⊕ data byte n ⊕ separator`

The checksum is calculated as one 8-bit binary value (00_h to FF_h). This byte is split into its upper and lower byte (nibbles). After the HEX values of the two nibbles have been transferred to the corresponding two ASCII characters (0 to 9 instead of 0_H to 9_H and A to F instead of A_H to F_H, that means to each nibble 30_H or rather 37_H is mathematically added), the checksum is written into the telegram.

The power stage also calculates (Exclusive OR) the checksum of the received data. The telegram will be rejected if a difference to the received checksum is detected.

If there is no need to validate the contents of the telegram, the checksum monitoring can be set to off. Instead of the checksum bytes, two X characters will be accepted by the power stage. Also,, telegrams without checksum and also without : (separator) will be accepted, e.g.:

Example: `<STX>| 1 | R | 4 | 0 | : | X | X |<ETX>` or

Example: `<STX>| 1 | R | 4 | 0 |<ETX>`

3.1 Operating Parameters

The operating parameters are stored in a permanent memory of the power stage.
In the following table the operating parameters are defined with
W for **w**rite,
R for **r**ead and
X for e**x**ecute.

For current values reading or writing is applied:
Integer value x 1/10 = valid current or voltage value

Example: 190 x 1/100 = 1.9 ($A_{r.m.s}$)

The power stage instruction set consists of two-byte-instructions. This means, that from the third byte the data for the instruction will follow.
The instructions for additional parameters begin with instruction code **P** to define security functions.
L or **U** input after the instruction code sets the instructioln's allowable limits, with **I** input after the instruction code the function of the instruction are displayed. With **S** and **E s**cale and unit (**E**inheit) of current and time values can be read.

The following table gives an overview:

Additional input to the instruction code	Function	Example	Description	Answer
I	**I**nformation about the instruction	FI	Information about the instruction F	fR/– Power Stage Status
U	**U**pper limit of the number range	SU	Stop current: highest value	s250
L	**L**ower limit of the number range	RL	Run current: lowest value	r10
S	Scale of current or temperature values: 1=1:1 10=1:10 100=1:100 1000=1:1000	SS	Read scale value of the stop current	s100
E	Unit of current- and time values: A, mA, ms, s	SE	Read unit of the stop current	eA
?	Read the preset value	R?	Display run current: 2.5 A	r250
<value>	Set the value	S50	Stop current 0.5 A	s50

Important:

- If the instruction input values are faulty, the actual preset value will result.
- If an instruction code is entered which is not implemented, the instruction code is answered by a lower case –character.
  Example: Enter K → Answer: k–

3.2 Instruction Set

✓= implemented

• = not implemented

Instruction set “Motor“					
Instruc-tion code	**Description**	**Type**	**Power Stage**		
			CCD+ **CLD+**	**ZMX+**	
A	Boost current in 1/10 A	R/W	0 to 63 0 to 14	—	
	Boost current in 1/100 A		—	0 to 630	
G	Pref. direction	R/W	1=preferential direction 0=contrary to pref. direction		
I	RMS-value of the current in 1/10 A	R	0 to 63 0 to 14	—	
M	Step width	R/W	0=1/1 1=1/2 2=1/2,5 3=1/4 4=1/5 5=1/8 6=1/10 7=1/16 8=1/20 9=1/32 10=1/64 11=1/128 12=1/256 13=1/512	0=1/1 1=1/2 2=1/2,5 3=1/4 4=1/5 5=1/8 6=1/10 7=1/16 8=1/20 9=1/32 10=1/64 11=1/128 12=1/256 13=1/512	
R	Run current in 1/10 $A_{r.m.s}$	R/W	1 to 63 0 to 14	—	
	Run current in 1/100 $A_{r.m.s}$		—	1 to 630	

Instruction set “Motor“					
Instruc-tion code	**Description**	**Type**	**Power Stage**		
			CCD+ CLD+	**ZMX+**	
S	Stop current in 1/10 $A_{r.m.s}$	R/W	0 to 63 0 to 14	—	
	Stop current in 1/100 $A_{r.m.s}$		—	0 to 630	
T	Delay time in ms	R/W	0 to1000	0=1 1=2 2=4 3=6 4=8 5=10 6=12 7=14 8=16 9=20 10=40 11=60 12=100 13=200 14=500 15=1000	
Z	Motor test: one motor rotation with preset run current	X	+ or –		

Instruction set “Status“					
Instruc-tion code	**Description**	**Type**	**Power Stage**		
			CCD+ CLD+	**ZMX+**	
B	Software version of the power stage	R	<String>		
BF	Software version of the FPGA	R	<String>		
D	Power stage temperature in 1/10 °C	R	0 to 999		
F	Power stage status	R	0001 to FFFF see chap. 3.2.1		
Q	Error inquiry	R	0 to 3 0=no error 1=undervoltage 2=overtemperature 3=overcurrent		
V	Intermediate voltage in 1/10 V	R	0 to 999		

Instruction set “Settings“					
Instruc-tion code	**Description**	**Type**	**Power Stage**		
			CCD+ CLD+	**ZMX+**	
C	Power stage Reset	X	✓		
E	Delete user parameters in the EPROM	X	✓		
J	Enforce the basic position	X	✓		
P	1st byte of the instructions ‚Extra parameters’, to define security functions	R/W/X	see chap. 3.2.2		
U	Power stage deactivation [1]	R/W	0=activated 1=deactivated		
W	Write parameter into EPROM	X	—	✓	

[1] Remark for ServiceBus mode: Please deactivate the power stage only by U=0, if it isn’t still deactivated by the input Activation/Deactivation.

Instruction set “I/O“					
Instruc-tion code	**Description**	**Type**	**Power Stage**		
			CCD+ CLD+	**ZMX+**	
L	Logic level of all inputs	R/W	—	0 = HIGH 1 = LOW	
LA, LB, LD, LR, LT, LX	Logic level of the defined inputs: A= Deactivation B= Boost D=Pref. direction R=Reset T= Control pulses X=Deselect	R/W	A,B,D,R, T,X 0 = HIGH 1 = LOW	—	
O	Logic level of the outputs (Open Collector)	R/W	—	0 = Transistor open 1= Transistor closed	

3.2.1 F Instruction code: Power stage status

The status of the power stage is read by the F instruction code as decimal or hexadecimal value. See example chap.4.2.3.

Hex.	Dec.	Valency	Description
01	**1**	**0**	**Error Undervoltage*)**
02	**2**	**1**	**Error Overtemperature*)**
04	4	2	not implemented
08	8	3	
10	16	4	
20	**32**	**5**	**Power stage in basic position**
40	**64**	**6**	**Error checksum**
80	**128**	**7**	**Reset in the power stage**
100	256	8	not implemented
200	512	9	
400	1024	10	
800	2048	11	
1000	4096	12	
2000	**8192**	**13**	**1= Boost current active**
4000	**16384**	**14**	**0=Boost/stop current active** **1=Boost/run current active**
8000	32768	15	not implemented

*) If both error bits are set, the status means “short circuit in power stage”.

3.2.2 Extra Parameters

Instruc-tion code	Description	Type	Power stage		
			CCD+ CLD+	ZMX+	
PE	Chopper frequency in kHz	R/W	—	0=25 1=50 2=75 3=100	
PH	Overdrive upper switching frequency in Hz		—	225 to 225000	
PI	Information about P instructions	R	✓		
PK	Display contrast	R/W	0 to100	—	
PL	Status display	R/W	0=I+U 1=I+M 2=I+T 3=TEMP 4=NONE	—	
PM	Status display	R/W	0=I+U 1=I+M 2=I+T 3=TEMP 4=NONE	—	
PN	Axis name	R/W	<String>		
PO	Overdrive[2]	R/W	0 = On 1 = Off		

[2] The function Overdrive is deactivated for CLD+.

Instruc-tion code	Description	Type	Power stage		
			CCD+ CLD+	ZMX+	
PS	Status ServiceBus	R	0 = Off 1 = On		
PX	Bus mode eXclusive	R/W	0= Don't ignore rotary switches 1= Ignore rotary switches	—[3]	
	Status ServiceBus	R	—	0 = Off 1 = On	

[3] The ServiceBus mode is activated in ZMX+ power stage by the "SB active" DIP switch. The switch status is read by "PX".

3.2.3 Examples

Instruction code	Action	Answer
A? **A180**	Read Boost current Set Boost current to 1.8 $A_{r.m.s}$	e. g. a160 (= 1.6 $A_{r.m.s}$) a180
B?	Read power stage software version	b<String> (e. g. bV1.0)
C	Reset power stage	c1
D?	Read power stage temperature	e. g.d58 (=58°C)
E	Delete EPROM	e1
F? **FH?**	Read power stage status in decimal mode in hexadecimal mode	e. g. f1 e. g. f0001 (= undervoltage)
G? **G1 or G0**	Read preferential direction Set preferential direction	g0 or g1 g1 or g0
I?	Read RMS-value of the current	e. g. i59 (= 5.9 $A_{r.m.s}$)
J	Enforce the basic position	j1
L? **L0 or L1**	Read input logic level Set input logic level	l0 or l1 l0 or l1
LB? **LT0 or LT1**	Read input logic level “Boost” Set input logic level “Control Pulses”	lb0 or lb1 lt0 or lt1
M0....M12 **M?**	Set step width Read step width	m0...m12 m0...m12
O0...O2 **O?**	Set output function Read output function	o0...o2 o0...o2
PC? **PC0...PC2**	Read current shaping CS Set current shaping CS	pc0...pc2 pc0...pc2
PE? **PE0...PE3**	Read chopper frequency Set chopper frequency	pe0...pe3 pe0...pe3
PH? **PH225...PH225000**	Read Overdrive switching frequency Set Overdrive switching frequency	ph225...ph225000 ph225...ph225000
PI	Information	pi<String>

Instruction code	Action	Answer
PK? **PK0....PK100**	Read display contrast Set display contrast	pk0...pk100 pk0...pk100
PL? **PL0...PL4**	Read status display Set status display	pl0...pl4 pl0...pl4
PM? **PM0...PM4**	Read status display Set status display	pm0...pm4 pm0...pm4
PN?[3] **PNAxis7** **PN/**	Read the axis name Store the axis name Delete the axis name	pn<String> (e. g. Axis4) pnAxis7 pn0
PO? **PO0 or PO1**	Read Overdrive function Set Overdrive function	po0 or po1 po0 or po1
PS?	Read ServiceBus status	ps0 or ps1
PX? **PX0 or PX1**	Read Bus mode eXclusiv Set Bus mode eXclusiv	px0 or px1 px0 or px1
Q?	Error inquiry	q0...q3
R? **R150**	Read run current Set run current to 1.5 $A_{r.m.s}$	e. g. r180 (=1.8 $A_{r.m.s}$) r150
S? **S240**	Read stop current Set stop current to 2.4 $A_{r.m.s}$	e. g. s180 (=1.8 $A_{r.m.s}$) s240
T? **T40**	Read delay time Set delay time to 40 ms	e. g. t40 (=40 ms) t40
U? **U0 or U1**	Read deactivation of the power stage Set deactivation of the power stage	e. g. u0 or u1 u0 or u1
V?	Read intermediate voltage	e. g. v400 (=40 V)
W	Write EPROM	w1
Z+, Z-	Do the motor self test	z1

[3] pn0 is also the answer of PN?, if no name is saved.

4 CAN Bus Configuration: SBM_{CAN}

4.1 General

This manual describes the firmware program (version 0.96) for the SBM CAN (SBM_{CAN}) protocol as implemented in the phytron SBM_{CAN} module. The software allows a register based CAN bus access to the stepper motor power stage ZMX^+. The SBM_{CAN} uses an asynchronous transfer protocol and operates as slave on the CAN bus, it will only send data on request by the host.

4.2 Specification

- Bus controlled configuration of the ZMX^+.
- CAN bus operation at 125kbps (default):
  configurable in 250/500/1000 kbps
- Register oriented access to amplifier settings.

4.3 Operating

The SBM_{CAN} uses dedicated identifiers in the range of $0x240_{hex}$ to $0x25F_{hex}$ (equivalent to decimal 576 to 607). The identifier must be uniquely selected using the **Address** switch on the ZMX^+ front panel. Sixteen different identifiers can be selected this way, enabling the user to access up to sixteen SBM modules via one physical bus connection.

SBM_{CAN} may be accessed by sending a CAN message with its receive identifier. SBM_{CAN} confirms the message by answering its transmit identifier (receive ID + 1), see table below:

Address switch	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Receive ID	240	242	244	246	248	24A	24C	24E	250	252	254	256	258	25A	25C	25E
Transmit ID	241	243	245	247	249	24B	24D	24F	251	253	255	257	259	25B	25D	25F

The length of a CAN message may vary between zero and eight bytes. Two different data lengths are used for write transfers to establish efficient and easy access to the SBM_{CAN}. As most of the registers are 32 bit wide a write operation to the registers uses five bytes data transfers, the first byte is the register address; the next four bytes are interpreted as data. Whereas read operations are initiated by a one byte transfer.

Receive

Received message from a control unit to SBM_{CAN} for read access:

ID	Register address

Received message from a control unit to SBM_{CAN} for write access:

ID	Register address	Byte 1 – lowest	Byte 2	Byte 3	Byte 4 – highest

Transmit

SBM_{CAN} sends data as 32 bit integers.

ID + 1	Register address	Byte 1 – lowest	Byte 2	Byte 3	Byte 4 – highest

Exceptions are only the version information commands: software or FPGA.
They send seven bytes coded as ASCII characters in the form “amplifier / version index”, “FPGA / Version index” e. g. “ZMX1.00” or “FPGA0.4“

Example: Transmitted message from SBM_{CAN} version information:

ID + 1	Register address	‘Z’	‘M’	‘X’	‘1’	‘.’	‘0’	‘0’

4.4 Register Layout

The SBM_{CAN} uses registers as illustrated in the following table. All registers have the same size of four bytes, but some of them are only partial used. Bytes exceeding the register´s data length are written to the register too but are ignored by the software. The index is in the range from 00_H to $3F_H$ (0_D...63_D).

Group 1: Information and Status			
Index	**Register Name**	**Brief Description**	**Example of Answer**
0	Power stage status	Read the power stage status	00|00|00|00|00
1	Error status	Read the Error status	01|00|00|00|00
2	Input Voltage	Read the Input Voltage in 1/10 V	02|8f|02|00|00 → 65.5 V
3	Power stage temperature	Read the power stage temperature in 1/10 °C	03|c8|01|00|00 → 45.6°C
4	Software version (7 Bytes)	Read the SBM_{CAN} controller software version	04|5a|4d|58|31|2e|30|30 → ZMX1.00
5	FPGA-Version (7 Bytes)	Read the ZMX^+ FPGA software version	05|46|50|47|41|30|2e|34 → FPGA0.4
6	Axis ID	Read/Write unique identifier code	06|00|00|00|00
7	ServiceBus switch	Read the ServiceBus switch position at “SB active” 0= Rotary switch mode 1= Servicebus mode	07|00|00|00|00→ 0
8	Reset	Read the Status of the external Reset input 0= passive 1= active	08|00|00|00|00→ 0

Group 2: Basic settings Power stage			
Index	**Register Name**	**Brief Description**	**Example of Answer**
16	Step resolution	Read/Write step resolution: 0=1/1 1=1/2 2=1/2,5 3=1/4 4=1/5 5=1/8 6=1/10 7=1/16 8=1/20 9=1/32 10=1/64 11=1/128 12=1/256 13=1/512	10|07|00|00|00 → 7
17	Boost current	Read/Write Boost current, 0 to 6.3 A in 1/100 A	11|86|01|00|00 → 3.9 A
18	Run current	Read/Write run current, 0 to 6.3 A in 1/100 A	12|04|01|00|00 → 2.6 A
19	Stop current	Read/Write stop current, 0 to 6.3 A in 1/100 A	13|82|00|00|00 → 1.3 A
20	Delay time	Read/Write delay time in ms	14|0a|00|00|00 → 10 ms
21	Pref. direction	Read/Write the Pref. direction 0 = CCW 1 = CW	15|00|00|00|00 → 0

Group 3: Advanced settings Power stage			
Index	**Register Name**	**Brief Description**	**Example of Answer**
32	Reset	Reset of SBM_{CAN}	20|00|00|00|00
33	Basic position	Reset of motor position for basic position	21|00|00|00|00
34	Deactivation	Read/Write Deactivation, The power stage is 1: currentless 0: energized	22|00|00|00|00 → Drive on
35	Current Shaping	Read/Write Current Shaping mode 0 = OFF 1 = ON	23|00|00|00|00 → 0
36	Overdrive	Read/Write Overdrive mode 0 = OFF 1 = ON	24|00|00|00|00 → 0
37	Overdrive frequency	Read/Write upper switching frequency of Overdrive in Hz: 225 to 225 000	25|e8|03|00|00 → 1000 Hz
38	Motor test	A 0 or 1 rotates the motor in or against the pref. direction	26|00|00|00|00

Group 4: Settings and I/O			
Index	**Register Name**	**Brief Description**	**Example of Answer**
48	Logic level	Read/Write the input logic level 0 = normal 1 = inverted	30|00|00|00|00 → 0
49	Output function	Read/Write the output logic level	31|00|00|00|00
52	Bus transmission rate	Change the transmission rate 0 = 1Mbps 1= 500kbps 2= 250kbps 3 = 125kbps A consecutive reset applies the new settings.	34|00|00|00|00 → 1Mbps (max. rate)

Group 5: Parameter storage			
Index	**Register Name**	**Brief Description**	**Example of Answer**
56	Delete ROM	Delete the parameter ROM and save the default values. A consecutive reset applies the new settings.	38|00|00|00|00
57	Write ROM	Save all parameter changes in the parameter ROM. A consecutive reset applies the new settings.	39|00|00|00|00

5 Copyright and Limitation of Warranty

The software ServiceBus-Comm and any documentation delivered with it are protected by copyright law. The manual must not be copied, reproduced, put into machine readable form, neither complete or in parts, without the prior written permission of Phytron-Elektronik GmbH.

ServiceBus-Comm as a freeware product will only give you the right to make backup copies for personal use. However, the program must neither be changed nor be sold.

Limitation of warranty

The software ServiceBus-Comm and its documentation have been made with great care and have been reproduced under effective controlling measures. Each CD-ROM delivered by Phytron-Elektronik GmbH is checked against viruses by means of a well known virus scan program. Nevertheless there might be mistakes.

You'll agree to our delivery and payment conditions, if you install, copy or otherwise use the software. If you disagree to these conditions you aren't authorized to install or use the software.

Trade marks

In this manual several trade marks are used which are no longer explicitly marked as trade marks within the text. The lack of this sign may not be used to draw the conclusion that these products are free of rights of third parties. Some product names used herein are for instance

ServiceBus-Comm is a trade mark of Phytron-Elektronik GmbH.

Microsoft is a registered trade mark and WINDOWS is a trade mark of the Microsoft Corporation in the USA and other countries.

6 Index

Phytron GmbH • Industriestraße 12 • 82194 Gröbenzell, Germany
Tel. +49(0)8142/503-0 • Fax +49(0)8142/503-190 • E-Mail info@phytron.de • www.phytron.de

Phytron, Inc. • 600 Blair Park Road Suite 220 • Williston, VT 05495 USA
Tel. +1-802-872-1600 • Fax +1-802-872-0311 • Email info@phytron.com • www.phytron.com