Operating Instructions

RS485 - Option (F1, F5)

For software versions V1.20

September 97

HAUSER
We automate motion

CERTIFIED DIN EN ISO 9001 QUALITY SYSTEM
Reg. Nr. 36 38

Parker Hannifin GmbH
EMD HAUSER
Robert-Bosch-Str. 22
D-77656 Offenburg, Germany
Phone: +49 (0)781 / 509-0
Fax: +49 (0)781 / 509-176
http://www.parker-emd.com

Parker Hannifin plc
EMD Digiplan
21 Balena Close
Poole, Dorset. BH17 7DX UK
Phone: +44 (0)1202 / 69-9000
Fax: +44 (0)1202 / 69-5750
http://www.parker-emd.com

Table of contents

This documentation applies to the following devices:

- COMPAX 25XXS with option F1 or F5
- COMPAX 45XXS with option F1 or F5
- COMPAX 85XXS with option F1 or F5
- COMPAX P1XXM with option F1, or F5 without option N1
- COMPAX 02XXM with option F1 or F5
- COMPAX 05XXM with option F1 or F5
- COMPAX 15XXM with option F1 or F5

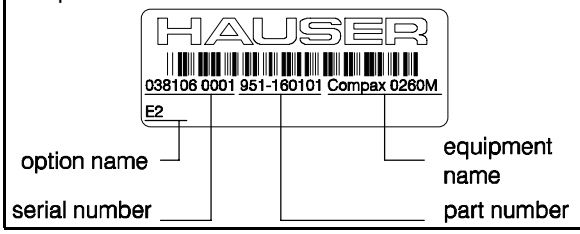
"XX": any version number
"F1/F5": RS485 option
"N1": single-phase supply (not possible in combination with F1)

Key to device designation:

e.g.: COMPAX 0260M:
COMPAX: Name
02: rated power class
60: version, e.g. "00" = standard device
M: device type where "M" = multi-axes device and "S" = single-axis device

HAUSER-name-plate

The name-plate is located on the top of the device and comprises:



1. Description of the RS485-interface

You can, via the RS232 interface of a industrial computer and an RS232-RS485 bus converter, communicate with up to 31 COMPAX devices. The following functions are supported:

- direct command input and execution in on-line mode,
- reading out status values,
- reading and writing program blocks (full command set available),
- reading and writing parameters, and
- transmission of control statements.

Interface parameter of the ASCII protocol:

Interface:	**F1: 4 wire for RS232-master via RS232/RS485 - converter.** **F5: 2 wire - RS485**
Baud rate:	150 up to 115200, set by parameter P195
Maximum line length:	1,2 kilometers
Devices on bus:	up to 31 COMPAX devices can be connected
Address:	set by parameter P194
Character length:	8 bits
Start bit:	1 bit
Stop bit:	1 bit
Parity:	none / even / odd, set by parameter P196
Hardware handshake:	no
Software handshake:	XON/XOFF, set by parameter P196
Timeout monitoring	can be activated, by parameter P196
Block-Check Character (BCC)	can be activated, improves transmission reliability
Input buffer:	command string, maximum 40 characters
Output buffer:	status string, maximum 40 characters
Data format:	ASCII, except for BCC
End-of-text character (ETX):	C_R: carriage return (0D hex)

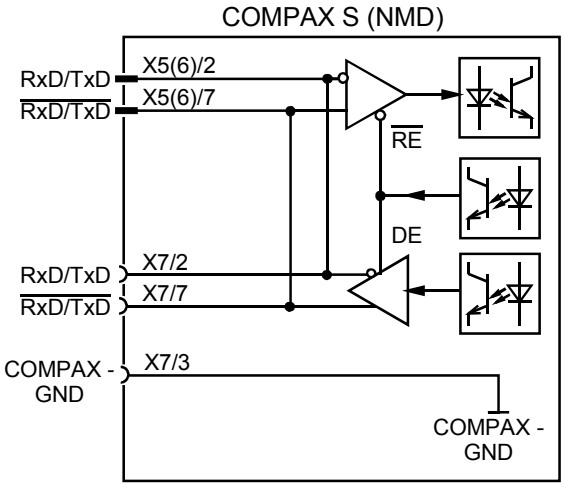
2. RS485-Bus

2.1 Bus wiring

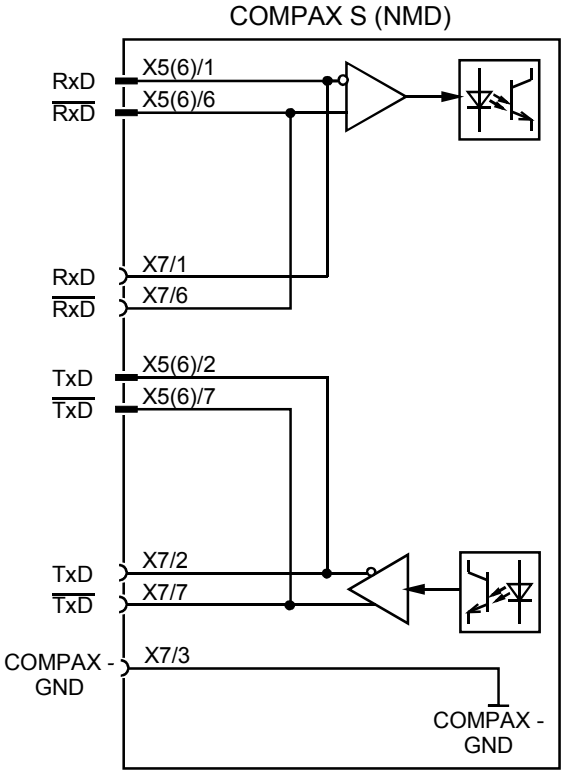
plug assignment of RS485-bus

COMPAX S: X5 / X7
NMD: X6 / X7

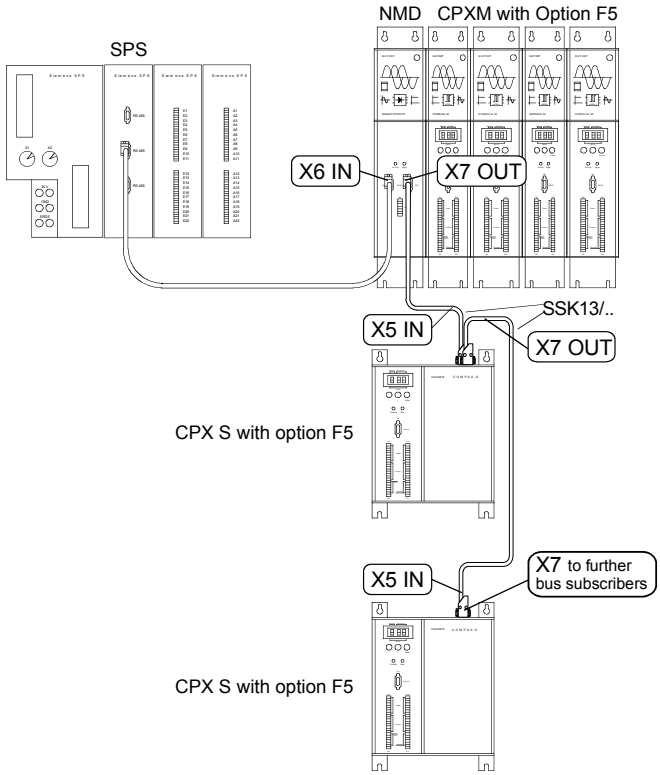
option F5



option F1



2.1.1 RS485 master at option F5



Within a system group of COMPAX M and a power module, the CAN-bus-signals are transmitted via the already existing ribbon cable connection.

possible connections with SSK13/..:

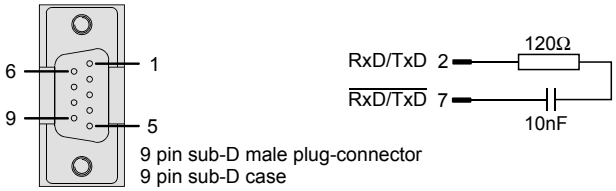
- power module X7 (OUT) → power module X6 (IN)
- power module X7 (OUT) → COMPAX S X5 (IN)
- COMPAX S X7 (OUT) → power module X6 (IN)
- COMPAX S X7 (OUT) → COMPAX S X5 (IN)

bus termination option F5

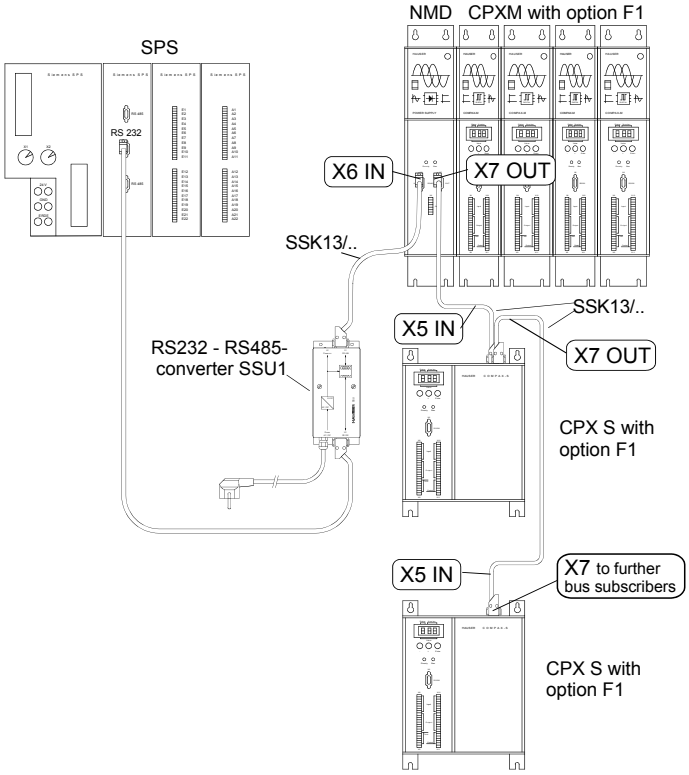
the last device of the bus system must be terminated by a resistor in order to guarantee interference-free operation.

F5: 120Ω and 10nF between X7/2 and X7/7 at power module or at COMPAX S.

X7



2.1.2 Via RS232-converter and option F1



In a system group comprising a number of COMPAX devices and a power module (NMD), the RS485 signals are transferred over the already existing ribbon-cable connections.

possible connections with SSK13/..:

- converter SSU1 → power module X6 (IN)

or

- converter SSU1 → COMPAX S X5 (IN)
- power module X7 (OUT) → power module X6 (IN)
- power module X7 (OUT) → COMPAX S X5 (IN)
- COMPAX S X7 (OUT) → power module X6 (IN)
- COMPAX S X7 (OUT) → COMPAX S X5 (IN)

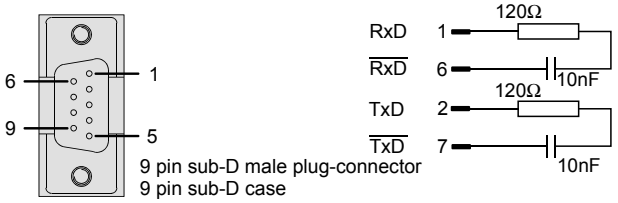
bus termination option F1

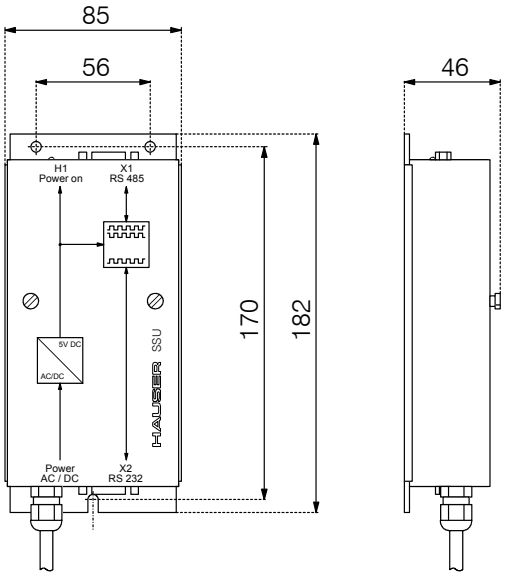
In order to guarantee interference-free operation, the last device of the bus system must be terminated by 2 resistors.

F1: 120Ω and 10nF between X7/1 and X7/6 and X7/2 and X7/7 at power module or at COMPAX S.

RS232 - RS485-converter: SSU1/01

X7

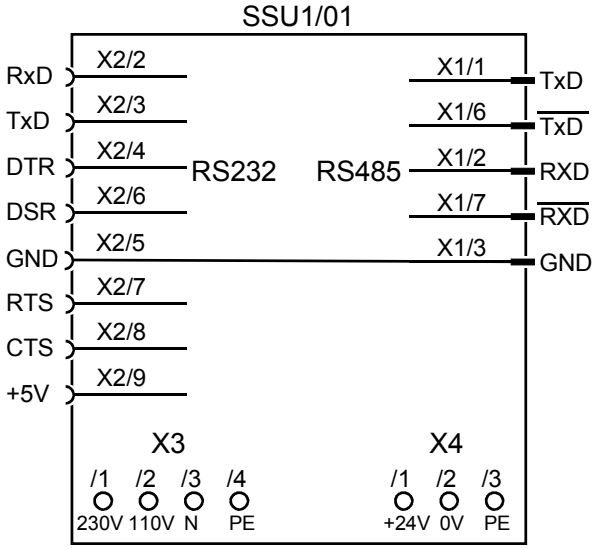




Attention!

The converter can only be used together with option F1 (4 wire)!

plug assignment SSU1/01



Fixing: 3 screws M4.
the converter is fed via a power line (3m) by 230V AC.
the power supply can be adjusted to 110 V AV and 24 V DC.
. For doing this, the housing must be opened and the feed lines have to be clamped as follows:

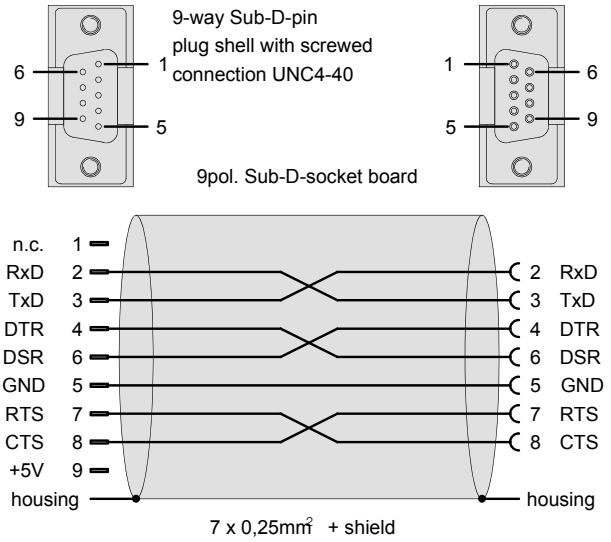
230V AC: X3/1 (L1); X3/3 (N) X3/4 (PE) (standard)
110V AC: X3/2 (L1); X3/3 (N) X3/4 (PE)
24V DC: X4/1 (+24V); X4/2 (0V) X4/3 (PE)

2.1.3 Position plan of the cables

position plan of the cables: SSK1

from RS232 - RS485 - converter to PC

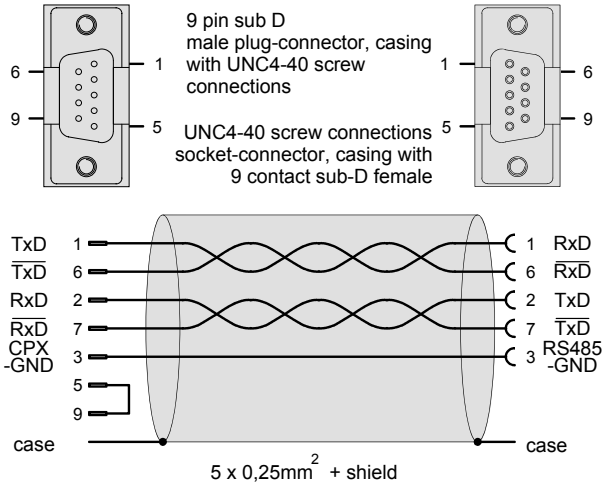
X6 PC / terminal



position plan of the cables: SSK13

converter i.e.. COMTAC to power module and power module to power module

NMD X7 / COMPAX S X7 SSU1 X1 — SSK 13/.. — NMD X6 / COMPAX S X5



3. Device settings

The RS485 interface supports a number of functions with selectable settings. These are set by means of definable parameters.

3.1 Device address

The device address on each COMPAX can be set by means of parameter P194.
The range is 0: 31, where "0" is the default setting. Before putting any COMPAX device into operation, an address from 1 to 31 must be set.

3.2 Baud rate

The baud rate on each COMPAX can be set by means of parameter P195. The range is 150 - 115200 bit/s.

⇨ Important: In all devices connected, the same baud rate must be set.

3.3 RS485 operating mode

3.3.1 Adjustment of operation mode at ASCII - transmission

to be adjusted via parameter P196

function	adjustment	valence
handshake:	none: with XON/XOFF	0 1
parity:	none: with: odd: even:	0 2 0 4
block-check-character:	none: with:	0 8
time-out-supervision:	none: with:	0 16
end-sign of COMPAX-reply	C_R L_F > C_R	0 32
COMPAX - reply	without address with preceding address of COMPAX	0 64

The desired adjustment can be carried out by entering the valence sum into parameter P196.

3.3.2 Adjustment of field-bus protocol by option F5

In order to be able to communicate with COMTAC, the field-bus protocol must be adjusted. . For doing this, please set **P196 = 164**.

Reference during an application with COMTAC, field-bus (option F5) and COMPAX:
Previous adjustment: P196=164
When using
◆F5 (2 wire-RS485) Software ≥V1.22 and
◆COMPAX - Version > V2.0
P196 has to be adjusted to 165.

Baudrate

At field-bus operation, 4 different baud rates (to be adjusted via P195) are possible:
◆ 28 800 baud
◆ 57 600 baud
◆ 172 800 baud
◆ 345 600 baud

3.3.3 Block-Check Character (BCC)

A byte-by-byte logical XOR of all characters in a transmission up until - but excluding - the C_R (0D hex) is sent after the last character; (C_R in the direction from PC to COMPAX, and the prompt in the direction from COMPAX to IPC). This performs an additional check on the transmitted data.
If the BCC function has been activated, COMPAX performs a block check, and reports any detected transmission errors in the form of error message E72. This error message does not have to be acknowledged; you simply send the same message again. If the BCC is error-free, COMPAX acknowledges the error message itself.

3.3.4 Timeout monitoring

As soon as COMPAX is activated via the assigned address, a time monitoring check is performed. For a definite period (namely the transmission time required for five characters), COMPAX awaits receipt of the next character. If COMPAX does not receive this character, it generates error message E73.
This error message does not have to be acknowledged; you simply send the same message again. If transmission is error-free, COMPAX acknowledges the error message itself.
If the BCC function has been activated, timeout monitoring is performed up until the BCC, and otherwise right up until the last character.

4. Transmission protocol

Transmission via the RS485 interface takes place in ASCII format (American Standard Code for Information Interchange), the same as via the COMPAX's RS232 interface, i.e. all transmitted characters are genuinely representable ASCII characters - with the exception of C_R, L_F, XON/XOFF, and the Block-Check Character (BCC).

4.1 Frame format for transmitting from PC to COMPAX

Necessary data	Optional extra characters
Device address	
Command code or execute character ("!" ≡ 21 hex)	
	Buffer character ("," ≡ 2C hex)
	No ackn. character ("/" ≡ 2F hex)
	Echo character ("?" ≡ 3F hex)
End-of-text character ("C_R"≡0Dhex)	
	Block-Check Character BCC

4.2 Meaning of the various transmission elements

Device address: Possible settings are from 1 to 32.
It is possible, by means of address 32 (broadcast address), to address all connected devices simultaneously. The addressed devices, however, do not send a reply.

Command code: The possible commands and their functions are explained in detail in the COMPAX-M Operating Instructions, in the description of the RS232 interface, in the Section "Interface functions".

Execute: character "!" ≡ 21 hex.
When this character is sent, a command previously stored in the buffer is executed.

⇨ When transmitting, you can send either a command code or the execute character.

Buffer: (character "," ≡ 2C hex)
If this character is appended to a command code, the command in question will not be executed immediately; it will be stored in the COMPAX buffer.

No acknowledgement: (character "/" ≡ 2F hex)
If this character is appended to a command code, COMPAX will not send a reply or acknowledgement.

Echo: (character "?" ≡ 3F hex)
If this character is appended to a command code, COMPAX will send the received data back in full, completing the return transmission with C_R L_F >.

End-of-text : (character "C_R" ≡ 0D hex)
"C_R" indicates the end of the character string being transmitted.

BCC If the BCC function has been activated, you must, in this position, transmit the byte-by-byte logical XOR of the characters you have transmitted (up until but not including C_R).

It is possible, for the purposes of particularly time-critical applications, to transmit a series of commands in COMPAX's specific binary form. Time can be saved by thus omitting the format conversion from ASCII to COMPAX-internal binary. Transmission in normal ASCII format is, even in these circumstances, still possible (i.e. mixed mode).

4.3 Frame format of the COMPAX response

Transmission, if error-free, is concluded with:

- C_R L_F >

or

- the desired reply and C_R L_F > BCC
  (BCC only if this function has been activated)

Transmission, if subject to error, is concluded with:

- Exx C_R L_F > BCC
  (BCC only if this function has been activated)
  where "Exx" = error message resulting from detected error

⇨ As first part of the reply there will be sent, if adjusted via P196, the address of the answering device.

4.4 Binary data transmission via RS485

As far as time-critical applications are concerned it is possible to transmit certain commands in the internal COMPAX binary format. By leaving out the format conversion from ASCII to the binary format, time can be saved; nevertheless, transmission in the normal ASCII-format is still possible (mixed operation)

At binary data transmission the frame format changes when transmitting from PC to COMPAX:

necessary data	characters that can be added optionally.
device address (ASCII-format)	
binary command code	
	block-check-character BCC

- **Please note that the device address will still be transmitted in ASCII-format!**
- **The end-sign may not be sent!**
- **The binary format has always to be transmitted in its whole length!**

4.4.1 Meaning of the binary command code

command	binary format (hexadecimal)	number formats
POSA value	88 41 xx xx xx xx xx xx LSB MSB	**number formats of "xx xx xx xx xx xx"** *[2] 3 byte of places behind/ 3 byte of places in front of the decimal point.
POSR value	88 52 xx xx xx xx xx xx LSB MSB	**valence:** 2^{-24} 2^{-23} ... 2^{-2} 2^{-1} , 2^{0} 2^{1} 2^{2} ... 2^{22} 2^{23}
SPEED value	88 53 xx xx xx xx xx xx LSB MSB	transmission order.e.g.: "88 41 LSB....MSB"
ACCEL value	84 4C yy yy MSB LSB	**number formats of "yy yy"** 2 byte of places in front of the decimal point. no places behind the decimal point.
ACCEL- value	84 44 yy yy MSB LSB	**valence:** *[1] 2^{15} 2^{14} ... 2^{2} 2^{1} 2^{0}.
OUTPUT Ayy=0	85 4F yy yy 30 MSB LSB	transmission order e.g..: "84 4C MSB LSB".
OUTPUT Ayy=1	85 4F yy yy 31 MSB LSB	
POSR value OUTPUT Ayy=0	8C 52 xx xx xx xx xx xx 4F yy yy 30 LSB MSB MSB LSB	
POSR value OUTPUT Ayy=1	8C 52 xx xx xx xx xx xx 4F yy yy 31 LSB MSB MSB LSB	
POSR value1 SPEED value2	8F 52 xx xx xx xx xx xx 53 xx xx xx xx xx xx LSB MSB LSB MSB	

*[1] Negative numbers

- Negative numbers are shown in a two's complement. - Formation of a two's complement:
- the bit combination of the positive numerical value has to be determined.
- the binary value must be negated.
- there has to be added "1".

*[2] Format transformation

Out of any optional number disposing of places behind the decimal point, you can generate this format as follows:

example: number = 450,5

1. Number to be multiplied by 2^{24}
   $450{,}5 * 2^{24} = 7558135808$.
2. 8388608 to be changed into a hexadecimal number (possibly change into integer before) => .0x00 01 C2 80 00 00 ≡ places in front of the decimal point,
   places behind the decimal point ≡ MSB,.... LSB, MSB,.... LSB.
3. These bytes have now to be filled into the command in the indicated order. The order of bytes is reversed. The order of bits is not allowed to be reverserd.

This conversion is also valid for negative numbers.

Example for the number format of xx xx xx xx xx xx"

number	MSB					LSB
10	00	00	0A	00	00	00
256	00	01	00	00	00	00
400,5	00	01	C2	80	00	00
-1	FF	FF	FF	00	00	00
	places in front of the decimal point			places behind the decimal point		

For **POSA 256,0** you get the following string:

"88 41 00 00 00 00 01 00"

Please note, that all places have to be transmitted

4.5 Program examples

In the following examples, the PC reads status S1 from two COMPAX devices, for which the following parameters have previously been set:
Device 1: P194 = 1 P195 = 9600bit/s **Device 2:** P194 = 5 P195 = 9600bit/s

4.5.1 BASIC program with BCC

◆P196 = 8 in both devices

```
OPEN "com2:9600,N,8,1" FOR RANDOM AS #1 ----- Initialization of interface as #1 (channel 1)
                                                com2:    serial interface COM2
                                                9600:    9600 baud transmission rate
                                                N:       no parity bit
                                                8:       8 bit word length
                                                1:       1 stop bit
text$ = "1S1" ------------------------------------------------- device 1 is to be interrogated for status 1
GOSUB output -------------------------------------------------- calls up subprogram "Output"
GOSUB read ---------------------------------------------------- calls up subprogram "Read"
PRINT text$ --------------------------------------------------- outputs read value to screen
text$ = "5S1" ------------------------------------------------- device 5 is to be interrogated for status 1
GOSUB output -------------------------------------------------- calls up subprogram "Output"
GOSUB read ---------------------------------------------------- calls up subprogram "Read"
PRINT text$ --------------------------------------------------- outputs read value to screen
END ----------------------------------------------------------- program end

output: ------------------------------------------------------- start of subprogram "Output"
bcc = 0 ------------------------------------------------------- bcc is set to 0
FOR a = 1 TO LEN(text$) --------------------------------------- loop performed for each character in text$
 bcc = bcc XOR ASC(MID$(text$, a, 1)) ------------------------ XOR operation on all characters to bcc
NEXT a -------------------------------------------------------- end of loop
PRINT #1, text$ ----------------------------------------------- text$ is output with concluding C_R to #1
PRINT #1, CHR$(bcc); ------------------------------------------ BCC in bcc is output as character without C_R to #1
RETURN -------------------------------------------------------- branch back to main program

read:--------------------------------------------------------- start of subprogram "Read"
INPUT #1, text$ ----------------------------------------------- value is read from #1 to text$
a$ = INPUT$(2, 1) --------------------------------------------- characters L_F and ">" are read from #1
bbcc = 0 ------------------------------------------------------ bbcc is set to 0
FOR a = 1 TO LEN(text$) --------------------------------------- loop performed for each character in text$
bbcc = bbcc XOR ASC(MID$(text$, a, 1))------------------------ XOR operation on all characters to bbcc
NEXT a -------------------------------------------------------- end of loop
ebcc = ASC(INPUT$(1, 1)) -------------------------------------- BCC is read from #1 to ebcc
IF ebcc = bbcc THEN RETURN------------------------------------ BCC read into ebcc is compared with
                                                               calculated BCC in bbcc; if values are same,
                                                               branch back to main program
PRINT "Transmission error !!!" -------------------------------- error message output on screen
RETURN -------------------------------------------------------- branch back to main program
```

4.5.2 BASIC program without BCC

◆P196 = 0 in both devices

```
OPEN "com2; 9600,N,8,1" FOR RANDOM AS #1-----Initialization of interface as #1 (channel 1)
                                                com2:    serial interface COM2
                                                9600:    9600 baud transmission rate
                                                N:       no parity bit
                                                8:       8 bit word length
                                                1:       1 stop bit
text$ = "1S1" ------------------------------------------device 1 is to be interrogated for status 1
GOSUB output -------------------------------------------calls up subprogram "Output"
GOSUB read ---------------------------------------------calls up subprogram "Read"
PRINT text$ --------------------------------------------outputs read value to screen
text$ = "5S1"------------------------------------------device 5 is to be interrogated for status 1
GOSUB output -------------------------------------------calls up subprogram "Output"
GOSUB read ---------------------------------------------calls up subprogram "Read"
PRINT text$--------------------------------------------outputs read value to screen
END----------------------------------------------------program end

output:------------------------------------------------start of subprogram "Output"
PRINT #1, text$----------------------------------------text$ is output with concluding C_R to #1
RETURN-------------------------------------------------branch back to main program

read: --------------------------------------------------start of subprogram "Read"
INPUT #1, text$----------------------------------------value is read from #1 to text$
a$ = INPUT$(2, 1)--------------------------------------characters L_F and ">" are read from #1
RETURN-------------------------------------------------branch back to main program
```

5. Parameters for the RS485 interface

No.	Meaning	Units	Minimum	Default	Maximum	Valid from
Parameters for the start-up of the RS485 interface						
P194	Device address		0	0	31	power-on
P195	Baud rate	bit/s	150	9600	115200	power-on
	at fieldbus protocol (P196 = 164/165)	bit/s	28800 • 57600 • 172800 • 345600			
P196	Operating mode		0	0	31	power-on

6. Special error messages relating to the RS485 interface

No.	Cause	Measures per cause	Ackn.	Drive disabled
E70	Parity error	Retransmit characters	[1]	no
E71	COMPAX buffer overflow; Retransmit more than 40 characters sent with fewer characters	Retransmit with fewer characters	[1]	no
E72	Block-Check Character error	Retransmit characters	[1]	no
E73	Timeout error	Retransmit characters	[1]	no
E74	BUSY error		power on	no

[1] No acknowledgement is needed; the error message will be erased automatically with the next error-free transmission.