Protocol description
Serial interfaces

PM 945 • PM 946 • PM 929 • PM 966
RM 45 • RM 46 • RM 29 • RM 66

for version P2.00 to 2.39

BROSE
*Systeme*

Table of Contens

The V.24 (RS-232) interface

Via the V.24, it is possible to transmit measured values from the instrument to a PC or printer, or to set parameters for the instrument. The hardware and software handshake are always supported.

The communication parameters such as baud rate, data bits, parity, stop bits, and address can be set in menu Func (see Operating Instructions, section "Function Settings").

Connecting with a Computer or Printer

The maximum connection shown in Fig. 1 can be minimized by locally bridging the control lines CTS with RTS and DTR with DST as shown in Fig. 2. This allocation type supports only the software handshake.

Figure 1   Figure 2
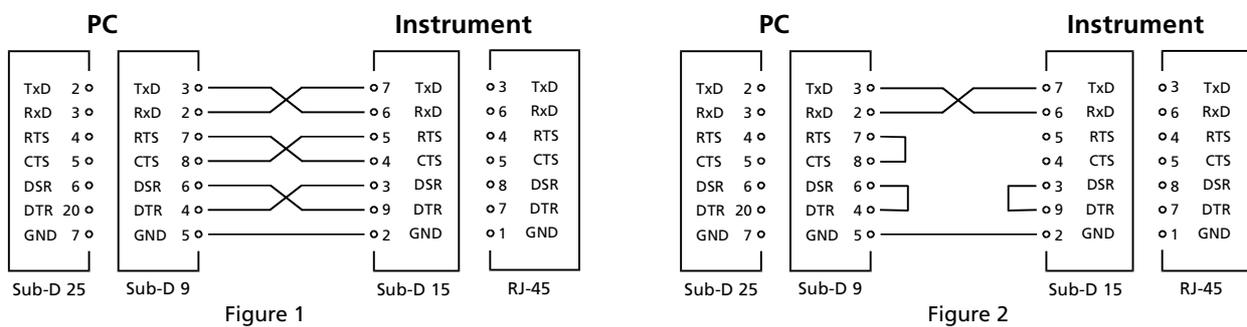
! **Attention:** On the instrument side, the CTS and RTS may not be bridged because the RTS shows the actual interface status and is not static.

Compatibility mode

The instrument's delivery state is set to compatibility mode and is extensively compatible with the operating mode of older instrument types.

This mode employs operating mode 1 and communication address 0, that is, data is sent continuously and there is no addressing support.

Function Description

The states listed below can be realized by software handshake and hardware handshake. The hardware handshake always has priority over the software handshake.

Definition of terms

Term / State	Ctrl. Char. / (Software)	V.24-Signal / (Hardware)	Function
'WAIT'	^S (DC3)	CTS low	Hold sending / measurement
'CONTINUE'	^Q (DC1)	CTS high *	Continue sending / measurement
'TERMINATE'	^T (DC4)	DSR low *	Terminate permanent sending
'RUN'	^R (DC2)	DSR high	Start permanent sending
'TRIGGER'	^F (ACK)	CTS low⇒high	Send 1 measured value in 'TERMINATE' state

* unconnected

WAIT AND CONTINUE STATE

Transmitting "WAIT" to the instrument will hold the sending of measured values and will freeze the current reading on the display. The measurement and the limit monitoring in the instrument are continued and the interface communication is in wait state. To cancel "WAIT" again, transmit "CONTINUE" to the instrument

TERMINATE AND RUN STATE

Transmitting "TERMINATE" to the instrument will stop the permanent sending of measured values. In "TERMINATE" state, the interface communication, with exception of "TRIGGER" and "CONTINUE", is deactivated. Transmitting "RUN" will start the permanent sending and will activate the interface communication.

TRIGGER STATE

In "TERMINATE" state, if only one single measured value is needed, it can be called up by transmitting "TRIGGER". If there is no new measured value yet, only a <CR> command is sent on a inquiry in this state. Only if a new measured is applied, is it sent on a renewed inquiry.

☞ **Note:** The reading on the display is transmitted as measured value; for example, if the reading is switched to maximum value memory then the reading in the maximum value memory is transmitted (see Operating Instructions, section "Function Settings").

BROSE

Normal Operation

This operating mode is set to communication address 0 and operating mode 0. Only after a one/several commands areis given, will a reply be sent with the data be sent in one/several replies (see section Description of Commands).

Addressed Operation

This operating mode is activated by setting the address to >0 in menu $Func.$ under the menu item Adr (see Installation Instructions, section Function Settings).

☞ **Note:** In operating mode 1 and 2, the automatic transmission of data is disabled and the data is sent only on command as in operating mode 0 (see section "Reading / Setting Operation Mode").

Addressing

The address is placed in front of each command line. The first address is A, the second B, then the third C and so forth (according to ASCII characters as of 41Hex), followed by a colon and the actual command; practically like a DOS command line with indication of drive on a PC.

Example: B:?<CR> Read the instrument version with address 2.

Wiring Principle

The instruments are connected as ring bus with a computer, that is, each outgoing line is connected to the reception line of the next instrument. All received characters will be delivered delayed by one character from each instrument. This way, all characters of the computer pass through the instrument and back to the computer again. For the support of the hardware handshake, the control lines RTS and CTS should be looped through analogous to the data lines instead of the bridge (broken line in Fig. 3)
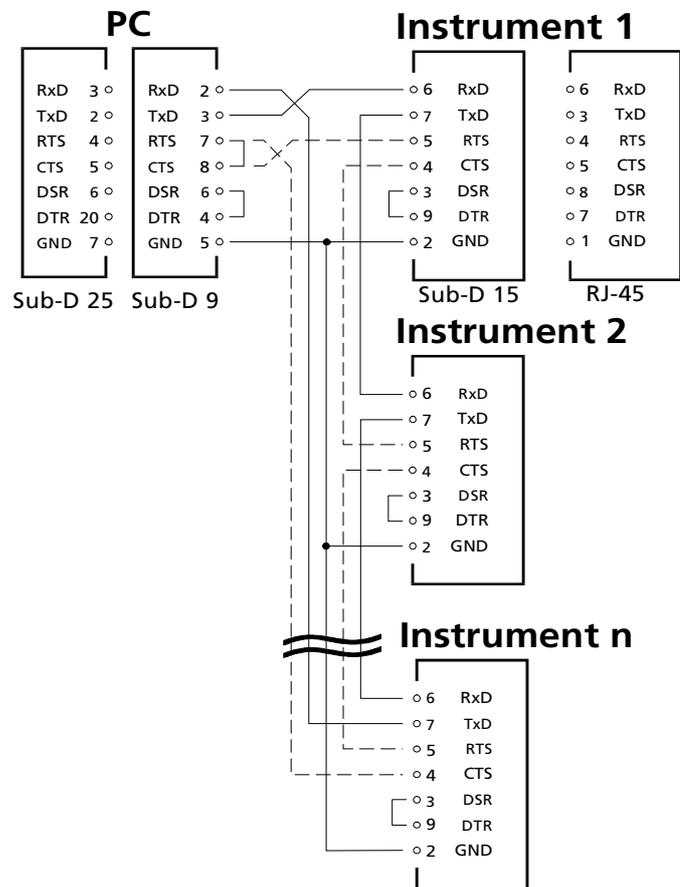

Figure 3

RS-485 Interface with Measurement Bus Protocol

This interface provides secure data transmission with the protocol in accordance with DIN 66348. The RS-485 is designed with 4-conductor technology (similar to RS-422) with identical pin assignment.

Die RS-422-Schnittstelle

This interface provides no internal termination.
The 15 pin Sub-D male connector assignment:

Pin	Signal
2	Receiving data R(A)
9	Receiving data R(B)
4	Transmitting data T(A)
11	Transmitting data T(B)
8	Ground (GND)
15	+ 5 V DC output for aktive termination

Parameterization Program SETBUS

This program enables easy and clear parameterization of the instrument. After starting the program, the parameters can be set for the interfaces:

SETBUS COM <PortNo.>:<Baud>

> <Baud> Transmission speed
> <PortNo.>: Number of computer's interface

Without parameter settings, the interface "COM2" and baud rate 9600 are set automatically for the transmission. After setting the interface parameters, the computer will display the SETBUS screen mask and search a bus user. When an active bus user is detected, the user's instrument type will be displayed. When the instrument type is recognized, the appropriate entry mask will be displayed with all available parameters. Using keys F1 to F10, all basic functions can be called up. Both lines at bottom of entry mask show at any time possible entries and explanations.

Description of Commands

General

For the communication with the instrument, which is carried out in ASCII, a series of commands is available. Each of these commands can be seen as an internal variable of the instrument, which can be accessed via the interface. In principle, each command can be used for reading/writing and setting. During reading, only the command or the variable is given; during writing, this command is assigned with another value with"=".
In principle, the commands are structured as follows:

Command [expansion] channel no.   for reading

Command [expansion] channel no. = parameterfor writing/setting

When writing/setting commands, the instrument confirms the execution of the command by returning "Ok". With reading commands, the reply occurs with the content of variables. It is possible to combine several commands, which are separated by a comma, in one command line. The command lines are processed from left to right. The receiving buffer can only hold a maximum of 20 characters. Therefore, the response to several read commands in one command line are several answers. When several write/set commands are in one command line, only the acknowledgement for the command line is output. Should syntax error appear in the command line, then the message syntax error is output and the processing of the command lines is stopped. If in normal operation, an initialization command is sent, the instruments returns the message "permission denied".

☞ **Note:** To test the communication with the instrument, terminal programs for DOS and Windows are offered on the homepage (see section Customer Service).

Number Representation

In principle, the internal number representation is in integer format, that is, in the value range from +32767 to -32768. The values +32767 and -32768 are interpreted as +OVER or –OVER (over range or overflow of the number format). The + or – sign is always given by the instrument for those number with signs and can be omitted when writing/setting positive numbers. The decimal point is represented as ASCII character (2E Hex).

Priorities

While making settings via the menu, there is no transmission via the interface.

Reading / Setting Operating Mode

Syntax:

M <CNo.> [= <Mode>]

M	Command
<CNo.>	Channel no. 0
=<Mode>	Set optional mode (0 - 255), otherwise read

The response of the instrument's interface is determined by the mode. The used mode is maintained even when switching on again. The indication of the channel number represents only a formal parameter, which should always be 0 with this instrument.
The mode is shown as positive number in the range from 0 to 255.

Example: Set mode to 129

Computer: M0=129
Instrument: Ok

Read mode

Computer: M0
Instrument: 129

The following modes are available:

Mode 0

In this mode, the instrument works on a stand-alone basis. However, it will only reply after receiving a command. The initialization commands are disabled to prevent unintentional conversion (see section "Mode +128").

Mode 1

In this mode, the instrument works on a stand-alone basis. However, it will continuously transmit the current measured value in the form: Measured value_Unit.

Example: +187.5 mV

Mode 2

In this mode, the instrument works as in mode 0. However, in case of a limit violation of one channel, it will continuously transmit the measured value. As long as the limit violation lasts, the current measured value will be transmitted.

Mode +128 (initialisation mode)

To call up the initialization mode, 128 has to be added to each mode, for example 0 and 1 (mode $0 \cong 128$, $1 \cong 129$, etc.). See example in section "Reading/Setting Operating Mode".

In this mode, the instrument works as in <128. However, it is also possible to use the initialization commands, which are:

E	Setting the measurement instrument
S	Scaling the measured value
C	Calibrating the measured value
G	Setting limit values
K	Setting relay configuration register
P	Setting parameter block

BROSE

Reading / Setting Measured Value

Syntax:

W [L I H I M] <CNo.> [= R I <Value>]

W	Command
L	Minimum value
H	Maximum value — optional, otherwise current measured value
M	Average value
<CNo.>	Channel no. 0
= R I <Value>	Set to optional value (integer), otherwise read

The instrument provides the following measured values:

Current measured value
Minimum measured value
Maximum measured value
Average measured value (max. over 93.2 h)

The values are replied in ASCII format with sign followed by decimal point. In addition, the possibility to set and reset the values is given in order to restart or reinitialize the generating of maximum value, minimum value and average values. The values must be given as integer value (range of values +32767 to -32768). An equal number of decimal positions are assumed as the ones that were set for the scaling.

Example: read current measured value

Computer: W0
Instrument: +5788 mm

change back average measured value

Computer: WM0=R
Instrument: Ok

read average measured value

Computer: WM0
Instrument: +3762 m/s

Reading / Setting Relay State

Syntax:

R <RNo.> [= 0 | 1]

    R   Command
    <RNo.>   Relay no. 0 or 1
    = 0 | 1   Set | reset optional relay state, otherwise read

The relay states of instrument can be inquired or influenced by this command. The relay status is indicated in Boolean form, that is, relay OFF = 0, and relay ON = 1.

☞ **Note:** Make sure that the response of the relay is carried out just as with the limit monitoring. If the relays are to respond exclusively via the interface, then the corresponding relay should be configured to OFF (passive), (see section "Reading / Setting Configuration Register" or in Operating Instructions, section "Relay Functions").

    <u>Example:</u>   Read relay state relay 0 (off here !)

                    Computer: R0
                    Instrument: 0

           Set relay state relay 0 to ON

                    Computer: R0=1
                    Instrument: Ok

☞ **Note for example:** The switching of relay may also depend on the limit monitoring, that is, the relay does not necessarily switch.

Reading / Setting Measurement Unit

☞ <u>Note:</u> In a <128 operating mode, setting the measurement unit is disabled (see section "Reading / Setting Operating Mode")

<u>Syntax:</u>

E <CNo.> [= <Unit>]

E	Command
<CNo.>	Channel no. 0
= <Unit>	Set optional unit (max. 8 characters), otherwise read

Using this command, a measurement unit can be given to the measured value. The measured unit is indicated as string with a maximum of 8 characters (ASCII characters 20 Hex - 7F Hex). If no other character is transferred after "="-sign (zero string), no measured unit is not given to the measured value or the entered measurement unit for the measured value is deleted. This command can call up the measurement unit directly. Indirectly, the measurement unit is given with the measured valued when called up (see section "Reading / Setting Measured Value").

   <u>Example:</u>   Read measurement unit

   Computer: E0
   Instrument: m/s

   Set measurement unit to m/s

   Computer: E0=m/s
   Instrument: Ok

BROSE

Reading / Setting Scale Factor

☞ **Note:** In a <128 operating mode, setting the scaling is disabled (see section "Setting / Reading Operating Mode")

Syntax for instruments PM945, PM946, PM929, RM45, RM46, RM29:

S <CNo.> [= SC , W1 , W2 , DP]

S	Command
<CNo.>	Channel no. 0
=	Optional for setting the scaling, otherwise reading
SC	Scale (for PM929: 0 = °C , 1 = °F , 2 = Kelvin)
	(for PM945/PM946: 0 = amplification 1, 1 = ampl. 2 , 2 = ampl. 3)
W1	Display value with measured value 0 (integer)
W2	Display value with measuring limit 19999 (integer)
DP	Amount of decimal positions (0 - 4)

To calibrate the electrical size to the actually measured physical size, a 2-point scaling is carried out. To do this, both measuring points, zero value and limit full scale value are assigned to the display value. The display value is given as a digit without a decimal point; the decimal position is given separately.

<u>Example:</u>	Scaling at	0 mA (0 μA) = 0
		20 mA (20000 μA) = 160.00
	Computer:	S0=0,0,16000,2
	Instrument:	Ok
	Read scaling	
	Computer:	S0
	Instrument:	0,+0,+16000,2

☞ **Note:** Note: As the scale factor at measurement outputs with linearization, for example temperature measurement, is based on internal table values, changing the scale is useless.

Syntax for instruments PM966, RM66:

S <CNo.> [= SC , W1 , W2 , DP]

S	Command
<CNo.>	Channel no. 0
=	Optional for setting the scaling, otherwise reading
SC	Scale / time base (0 = pulses/s , 1 = pulse/min , 2 = kilo-pulse/h)
W1	Factor/multiplier (integer)
W2	Divisor (integer)
DP	Number of decimal places (0 - 4)

To calibrate the measured pulse size to the measured physical size, a factor, the divisor and time base are given. Factor and divisor are given as digits without decimal point; decimal points are given separately.

Reading / Setting Calibration

☞ **Note:** In a < 128 operating mode, setting the calibration is disabled (see "section Reading/Setting Operating Mode").
This command is not available in instruments PM966 and RM66!

C <CNo.> [= SC , W1
      W2 , DP]

C	Command
<CNo.>	Channel no. 0
=	Optional for setting the calibration, otherwise reading
SC	Scale (with PM929: 0 = °C , 1 = °F , 2 = Kelvin)
	(with PM945/PM946: 0 = amplification 1, 1 = ampl. 2 , 2 = ampl. 3)
W1	Display value with first applied measured value (integer)
W2	Display value with second applied measured value (integer)
DP	Amount of decimal positions (0 - 4)

On calibration, the respective reading is allotted to the measured variables, the same as for the zero point and full scale allocation. In contrast to this, the allocation is not, however, carried out for the zero and the full scale but on the currently applied measured variable. From this, the instrument calculates the values for the zero point and full scale. To carry out calibration, the command is subdivided into two parts. Apply the 1st calibration variable to the instrument. The instrument measures the applied calibration variable and sends it back as digit. The second calibration variable must be applied to the instrument after this and the same procedure is carried out as in the case of the first calibration variable. After both calibration variables were applied, the instrument calculates the values for the zero point and full scale, dependent on the internal basic resolution and the scaling.

Calibration differs from scaling only in that the measured values for zero and end value are actually measured rather than already allocated. This command is divided into two parts to enable the input size to be changed. Before a display value is transmitted, the measured quantity in question is set at the measurement input. After the display value has been transmitted, the unit carries out a measurement and then gives the measured value in the internal basic measurement accuracy.

  Example: Calibration   1st measured value 0 V = display 0
                 2nd measured value 8 V = display 237.5

             Set measured value 0 V!
      Computer: C0=0,0
      Instrument: -5   (-5 counts measured)
             Set measured value 8 V!
      Computer: 2375,1
      Instrument: +7995 (7995 counts measured)

  "Read Calibration" and "Read Scaling" are identical,
  see section Reading / Setting Scaling.

☞ **Note:** This function may also be adjusted directly on the instrument with the menu $Func.$ menu item CAL (see Operating Instructions, section "Function Settings - Perform Calibration").

☞ **Note:** So that a high accuracy is achieved, the calibration variables should be selected

as far apart as possible.

Reading / Setting Limit Values

☞ **Note:** In a < 128 operating mode, setting the configuration register is disabled (see section "Reading / Setting Operating Mode")

Syntax:

G <LNo.> [= <1st value> , <2nd value> , <Hyst.>]

G	Command
<LNo.>	Limit value no. 0 or 1
=	Optional for setting the limit values, otherwise reading
<1st value>	1st limit value
<2nd value>	2nd limit value
<Hyst.>	Turn-off hysteresis value (only positive value selectable)

Two limit values are available for each channel, which can be used to control the relays. To avoid the relay from "jittering", a hysteresis can be preset. It indicates the amount less than the value, which must be reached before the relay drops again. All values are given in their basic scaling without a decimal point.

Example: Limit value pair 2 should give an alarm at a measured valued < 0 or > 187.9 with a hysteresis of 1.0.

Computer: G1=0,1879,10
Instrument: Ok

Read limit value pair 2

Computer: G1
Instrument: +0,+1879,10

☞ **Note:** The 2nd limit value will not only be fetched for the limit value monitoring if the configuration of one relay is set configuredto tolerance band monitoring for this limit value pair (see Operating Instructions, section "Function Settings - Adjusting Relay Function").

Reading / Setting Configuration Register

Note: In a < 128 operating mode, setting the configuration register is disabled (see section Reading / Setting Operating Mode)

Syntax:

K <RNo.> [= <Value>]

K	Command
<RNo.>	Relay no. 0 or 1
= <Value>	Set optional config register to value (0 - 255), otherwise read

The response action of each relay can be adjusted via the configuration register. The setting also affects the limit value monitoring. Only if one relay is set to tolerance band monitoring (value > 5) will the 2nd limit value be fetched for processing.

Following options can be selected:

Value	Relay behaviour
0	passive, reacts to no limit values only to R- command
1	relay always active when instrument is in operation
2	activates when limit value 1 is reached
3	activates when limit value 2 is reached
4	activates when limit value 1 is not reached
5	activates when limit value 2 is not reached
6	active as long as measured value is between limit value pair 1
7	active as long as measured value is between limit value pair 2
8	active when measured value is outside limit value pair 1
9	active when measured value is outside limit value pair 2

Example: Relay 0 should react only to interface command

Computer: K0=0
Instrument: Ok

Read configuration relay 0

Computer: K0
Instrument: 0

Note: This function may also be set directly in menu *Func.* menu item *rEL.1/rEL.2.*

Reading the Instrument Version

<u>Syntax:</u>

?

      ?   Command

This command serves the identification of instrument; the model number and software version are given.

    <u>Example:</u>   Read Instrument version

                       Computer: ?
                       Instrument: PM945/H - V1.10

Reading the Instrument Version

Reading / Setting the Parameter Block

☞ **Note:** In a < 128 operating mode, setting the parameter block is disabled (see section "Reading / Setting Operating Mode")

Syntax:

P <CNo.> [= <PBlock>]

    P   Command
    <CNo.>   Channel no. 0
    = <PBlock> Optional for write setting, otherwise read

The reading and writing of the parameter block is for backing-up the data or quickly parameterizing the unit via the interface. By simply writing the stored parameter block, it is thus possible to parameterize an exchanged unit without using the individual commands. The parameter block itself consists of a string of hexadecimal numbers and is transmitted in four part blocks. The block contains a check sum and may therefore be transmitted only in unaltered form.

☞ **Note:** During transmission, the first part blocks must be designated as block, which means, for example they are ended with "end of transmission block" (17 Hex) and not with "end of text" (03 Hex) or via the V.24 (RS232) interface ended with "line feed" (0A Hex) and not with "carriage return" (0D Hex).

    <u>Example:</u> Write parameter block

                    Computer: P0=0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                    Instrument: Ok

            Read parameter block

                    Computer: P0
                    Instrument: 0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF
                                      0000FFFF0000FFFF