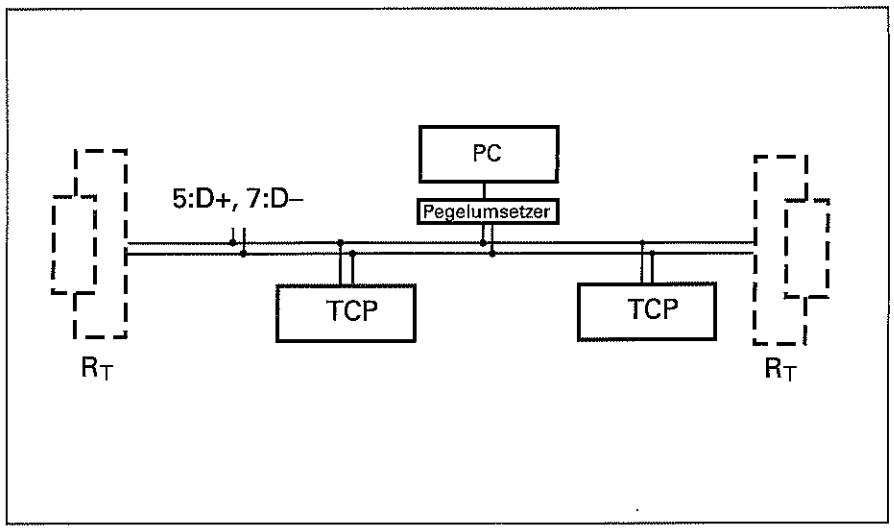
Pfeiffer Vacuum-Protokoll
Pfeiffer Vacuum Protocol



Schnittstelle RS 232, RS 485
Interface RS 232, RS 485

PM 800 488 BN/C (0309)

PFEIFFER VACUUM

1. Pfeiffer Vacuum Protocol

The Pfeiffer Vacuum Protocol is in ASCII format which means that all data bytes are representable symbols with an ASCII code ≥ 32* with the exception of the telegram final symbol carriage return (CR, 13). The transmitted telegrams are located, without exception, within the following frameworks:

General Protocol:

Adress	Action	Parameter Number	Data Length	Data	Checksum	CR

Adress: Adress of the unit addressed or answering, for example "042": The addresses are distinguished as follows:
- Individual addresses: Only specific component is addressed.
- Group addresses: All components of a group, for example all TCM 1601.
- Global address: Address "000", all Pfeiffer Vacuum components are addressed.

Action: "00" = Read parameter
"10" = Describe parameter

Parameter number: Number of the relevant parameter, for example "303".

Data length: For example "06" for six symbols, corresponds to the length of the field "Data".

Data: Data in ASCII format. Format and size of the data are guided by the following considerations:

- Transmission of values ⇨ *2.1 master telegrams and 3. data formats*
- Data request ⇨ *2.2 slave telegrams and 3. data formats*
- Error messages ⇨ *2.2 slave telegrams*

Checksum: The sum of all ASCII symbols up to preceding checksum modulo 256 (decimal), for example sum = 786. 786 modulo 256 = 18 ⇨ Checksum = "018" (converted into ASCII string).

CR: Carriage return (ASCII symbol 13).

As a result of the master/slave behaviour, data exchange always proceeds on the pattern: Master transmits (either position request or request). Slave answers (confirmation or transmission of data/error messages):

2. Telegrams

2.1 Master-Telegrams

The component accepting the communication (master, for example PC) can send two different telegrams:

Position request:

Address			Action		Parameter Number			Data Length		Data	Checksum			CR
a1	a2	a3	1	0	n1	n2	n3	d1	d2	Data	c1	c2	c3	CR

Data request:

Address			Action		Parameter Number			Data Length		Data		Checksum			CR
a1	a2	a3	0	0	n1	n2	n3	0	2	=	?	c1	c2	c3	CR

2.2. Slave-Telegrams

The slave component (for example, Pfeiffer drive unit) cannot independently begin a communication and only answers when it is addressed with a valid individual address. Components which are addressed via the group or global address do not answer. The following telegrams are possible:

***all terms decimal**

Transmission of the requested data (positive response to "Data request":

a1	a2	a3	1	0	n1	n2	n3	d1	d2	Data						c1	c2	c3	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

Confirmation of the received position request (positive response to "Position request"):

a1	a2	a3	1	0	n1	n2	n3	d1	d2	Data						c1	c2	c3	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

A confirmation of the received position request means only at first that the telegram transmitted by the master has been understood. If the operational status of the component permits an alteration, this will be executed. To check, it is recommended to then request the parameter.

The parameter does not exist (error message):

a1	a2	a3	1	0	n1	n2	n3	0	6	N	O	_	D	E	F	c1	c2	c3	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

Transmitted data are outside the permitted range (error message):

a1	a2	a3	1	0	n1	n2	n3	0	6	_	R	A	N	G	E	c1	c2	c3	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

Logic error (for example, the writing of an only readable parameter, error message:

a1	a2	a3	1	0	n1	n2	n3	0	6	_	L	O	G	I	C	c1	c2	c3	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

2.3. Telegram Examples

Reading the actual rotation speed:

master ⇨ slave:

1	2	3	0	0	3	0	9	0	2	=	?	1	1	2	CR
Address			Action		Parameter Number			Data length		Data		Checksum			CR

slave ⇨ master:

1	2	3	1	0	3	0	9	0	6	0	0	0	6	3	3	0	3	7	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

Adjusting the maximum start-up time to 12 minutes:

master ⇨ slave:

0	0	1	1	0	7	0	0	0	6	0	0	0	0	1	2	0	1	8	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

slave ⇨ master:

0	0	1	1	0	7	0	0	0	6	0	0	0	0	1	2	0	1	8	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

Switching on the motor:

master ⇨ slave:

0	4	2	1	0	0	2	3	0	6	1	1	1	1	1	1	0	2	4	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

slave ⇨ master:

0	4	2	1	0	0	2	3	0	6	1	1	1	1	1	1	0	2	4	CR
Address			Action		Parameter Number			Data length		Data						Checksum			CR

3. Data Formats

Depending on the content of the parameter, the data field can present various formats. The following data types are possible:

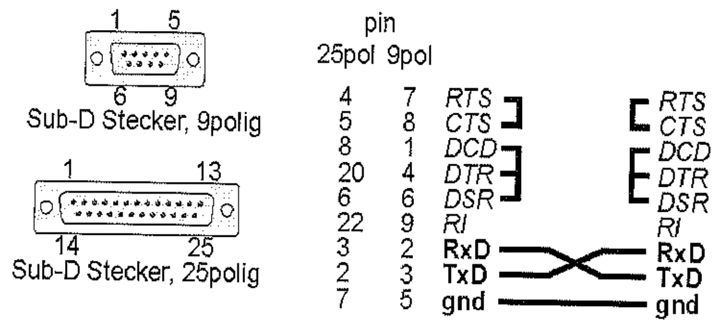
Data type	Description	Size in symbols	Examples
0 - boolean_old	true / false in the form six zeros (ASCII 48) or ones (ASCII 49)	6	000000 corresponds to false 111111 corresponds to true
1 - u_integer	pre-symbol-less integer number with six positions (leading zeros)	6	000042 123456 001200
2 - u_real	fixed comma number with four positions before and two after the comma standardised to 0.01 (leading zeros)	6	001570 corresponds to 15,70 000020 corresponds to 0,2
3 - u_expo	positive exponential number (leading zeros)	6	1.2E-2 corresponds to 1,2x10^{-2} 0005E8 corresponds to 5x10^{8}
4 - string	optional symbol chain with ASCII symbols ≥ 32 (decimal)	6	hallo! TC_600 hgnrfx
5 - vector	several parameter numbers n with values starting with the number of the following parameters (two positions)	n	02001000000702120 value #2 parameter #2 value #1 parameter #1 two parameters follow
6 - boolean_new	true/false in the form of a zero (ASCII 48) or one (ASCII 49)	1	0 corresponds to false 1 corresponds to true
7 - u_short_int	pre-symbol-less integer number with three positions (leading zeros	3	123 042 007
9 - tms_old	TMS control status, first three bytes boolean, last three bytes u-short-int	6	000037 control off, temp = 37°C 111119 control on, temp = 119°C
10 - u_expo_new	Positive exponential number; the first four numbers includes the mantissa multiplied with 1000, the last both the exponent with Offset 20	6	100023 corresponds to 1.000E3 456711 corresponds to 4.567E-9
11 - string16	any character string with ASCII-codes ≥ 32 (decimal)	16	BrezelBier&Wurst 0123456789ABCDEF
12 - string8	any character string with ASCII-codes ≥ 32 (decimal)	8	Pfeiffer 01234567 >Vacuum<

4. General Information On The Serial Interface

The configuration of the serial interface topology, cables and plugs etc. is, for the most part, user specific. However, the following sections will serve as a useful aid.

4.1. RS 232

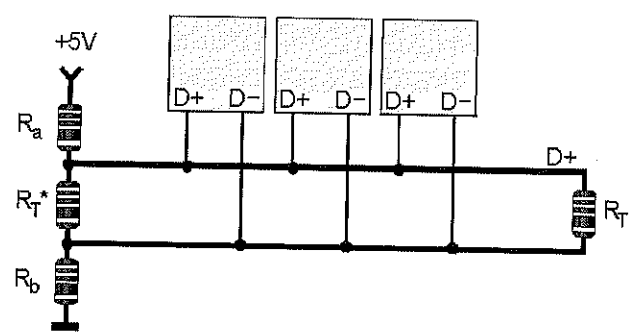
The Serial Interface RS 232 was originally designed as a modem connection and has at its disposal therefore many more signals than are required in many applications without modem. As a rule, the two data cables TxD and RxD related to mass are sufficient.



However, some other signal cables must also be provided with defined levels so as not to simulate current signals on the RS232 driver component where applications without modem are involved. With this type of serial interface only a point to point connection from precisely two components is possible. Illustration 1 shows an example of an RS232 connection. The use of shielded cable is recommended. The bus terminal resistances are often integrated in the driver components. The normal plugs are Sub-D 9 pole or 25 pole.

4.2. RS 485

The RS 485 Serial Interface allows up to 32 components to be connected with each other via two cables whereby never more than one component is able to transmit at any one time. All components are connected with the D+ cable via their D+ connection (or DO/RI) and their D- connection (or DO/RI) via the D- cable. Because several components can be connected to the bus, normally no bus terminal resistances are integrated in the driver components. These have to be connected to the two furthest ends of the bus. The values of the resistances are determined by the wave impedance of the cable in use. To ensure optimal bus driver functioning, the inactive bus must be maintained at a voltage range of 200 mV (logical 1) via external wiring.



For practical purposes the resistance values are in the range R_a=R_b=680 Ω,, R_T=120 Ω, and R_T*=130 Ω, for transposed, shielded, two wire cables (see illustration 2). With respect to bus topology a main cable with the shortest possible dead end feeder is recommended.