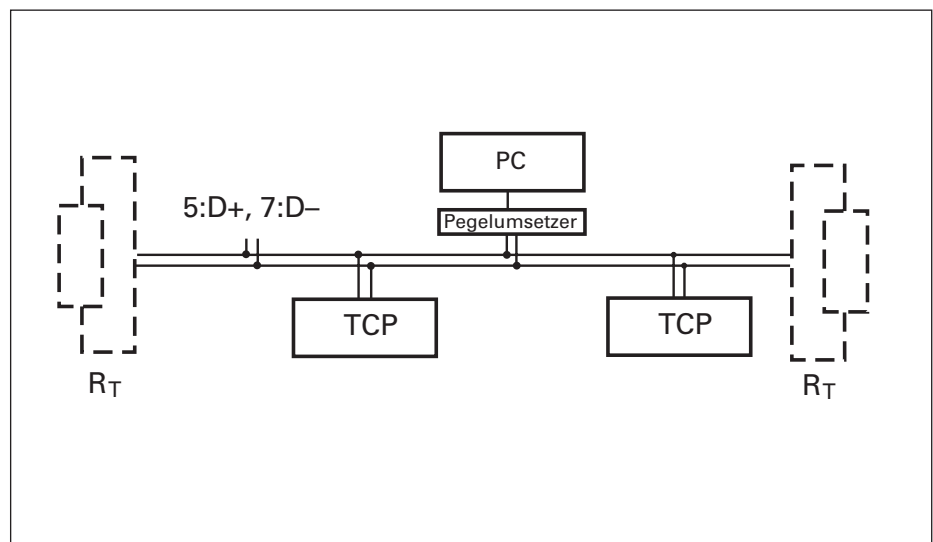


Pfeiffer Vacuum-Protokoll ***Pfeiffer Vacuum Protocol***



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

Inhalt/Contents

	Seite
1. Pfeiffer Vacuum Protokoll.....	3
2. Telegramme	3
2.1. master-Telegramme.....	3
2.2. slave-Telegramme	4
2.3. Telegramm-Beispiele	4
3. Datenformate	5
4. Allgemeines über Schnittstellen.....	5
4.1. RS 232	5
4.2. RS 485	6

	Page
1. Pfeiffer Vacuum Protocol	7
2. Telegrams.....	7
2.1. Master Telegrams	7
2.2. Slave Telegrams	8
2.3. Telegram Examples	8
3. Data Formats	9
4. General Information On The Serial Interface	10
4.1. RS 232.....	10
4.2. RS 485	10

1. Pfeiffer Vacuum-Protokoll

Das Pfeiffer Vacuum-Protokoll bedient sich des ASCII-Formats, d.h. alle Datenbytes sind darstellbare Zeichen mit einem ASCII-Code ≥ 32 mit Ausnahme des Telegramm-Ende-Zeichens carriage return (CR, 13). Die übertragenen Telegramme befinden sich ohne Ausnahme in einem wie folgt gestalteten Rahmen:

allgemeines Protokoll:

Adresse	Aktion	Parameternummer	Datenlänge	Daten	Checksumme	CR
---------	--------	-----------------	------------	-------	------------	----

Adresse: Adresse des angesprochenen bzw. des antwortenden Gerätes, z.B. „042“. Es werden folgende Adressen unterschieden:

- Einzeladressen: es wird nur ein bestimmtes Gerät angesprochen.
- Gruppenadressen: alle Geräte einer Gruppe, z.B. alle TCM1601.
- globale Adresse: Adresse „000“, alle Pfeiffer Vacuum Geräte werden angesprochen.

Aktion: „00“ = Parameter lesen
„10“ = Parameter beschreiben

Parameternummer: Nummer des betreffenden Parameters, z.B. „303“

Datenlänge: z.B. „06“ für sechs Zeichen, entspricht Länge des Feldes „Daten“

Daten: Daten im ASCII-Format. Format und Größe der Daten richten sich nach folgenden Punkten:

- übermitteln von Werten \Rightarrow 2.1 master-Telegramme & 3. Datenformate
- Datenabfrage \Rightarrow 2.2 slave-Telegramme & 3. Datenformate
- Fehlermeldungen \Rightarrow 2.2 slave-Telegramme

Checksumme: Summe aller ASCII-Zeichen bis vor Checksumme modulo 256 (dezimal), z.B. Summe = 786, $786 \bmod 256 = 18 \Rightarrow$ Checksumme = „018“ (umgewandelt in ASCII-string)

CR: carriage return (ASCII-Zeichen 13)

Durch das master-slave-Verhalten verläuft ein Datenaustausch immer nach dem Schema: master sendet (entweder Stellaufruf oder Anfrage), slave antwortet (Bestätigung oder Senden von Daten / Fehlermeldungen).

2. Telegramme

2.1 master-Telegramme

Das die Kommunikation aufnehmende Gerät (master, z.B. PC) kann zwei verschiedene Telegramme verschicken:

Stellaufruf:

a1	a2	a3	1	0	n1	n2	n3	d1	d2	Daten	c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten	Checksumme			CR

Datenabfrage:

a1	a2	a3	0	0	n1	n2	n3	0	2	=	?	c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten	Checksumme			CR	

2.2. slave-Telegramme

Das slave-Gerät (z.B. Pfeiffer Antriebsgerät) kann von sich aus keine Kommunikation starten, sondern antwortet nur, wenn es mit gültiger Einzeladresse angesprochen wird. Geräte, die über die Gruppen- oder globale Adresse angesprochen werden antworten nicht. Folgende Telegramme sind möglich:

*alle Zahlenangaben dezimal

Senden der abgefragten Daten (positive Antwort auf „Datenabfrage“):

a1	a2	a3	1	0	n1	n2	n3	d1	d2							c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

Bestätigung der erhaltenen Stellaufforderung (positive Antwort auf „Stellaufforderung“):

a1	a2	a3	1	0	n1	n2	n3	d1	d2							c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

Eine Bestätigung der erhaltenen Stellaufforderung bedeutet zunächst nur, daß das vom master gesendete Telegramm verstanden worden ist. Wenn der Betriebszustand des Gerätes eine Verstellung erlaubt, wird diese auch ausgeführt. Zur Überprüfung empfiehlt es sich, den Parameter anschließend abzufragen.

Parameternummer existiert nicht (Fehlermeldung):

a1	a2	a3	1	0	n1	n2	n3	0	6	N	O	_	D	E	F	c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

übergebene Daten außerhalb des erlaubten Bereichs (Fehlermeldung):

a1	a2	a3	1	0	n1	n2	n3	0	6	_	R	A	N	G	E	c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

logischer Fehler (z.B. Schreiben eines nur lesbaren Parameters, Fehlermeldung):

a1	a2	a3	1	0	n1	n2	n3	0	6	_	L	O	G	I	C	c1	c2	c3	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

2.3. Telegramm-Beispiele

Lesen der aktuellen Drehzahl:

master ⇒ slave:

1	2	3	0	0	3	0	9	0	2	=	?	1	1	2	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten		Checksumme			CR

slave ⇒ master:

1	2	3	1	0	3	0	9	0	6	0	0	0	6	3	3	0	3	7	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

Einstellen der maximalen Anlaufzeit auf 12 Minuten:

master ⇒ slave:

0	0	1	1	0	7	0	0	0	6	0	0	0	0	1	2	0	1	8	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

slave ⇒ master:

0	0	1	1	0	7	0	0	0	6	0	0	0	0	1	2	0	1	8	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

Motor einschalten:

master ⇒ slave:

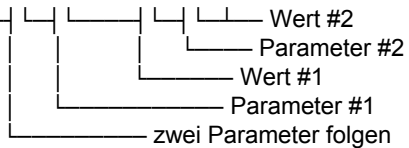
0	4	2	1	0	0	2	3	0	6	1	1	1	1	1	1	0	2	4	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

slave ⇒ master:

0	4	2	1	0	0	2	3	0	6	1	1	1	1	1	1	0	2	4	CR
Adresse			Aktion		Parameternummer			Datenlänge		Daten						Checksumme			CR

3. Datenformate

Abhängig vom Inhalt der Parameter kann das Datenfeld unterschiedliche Formatierungen aufweisen. Folgende Datentypen sind möglich:

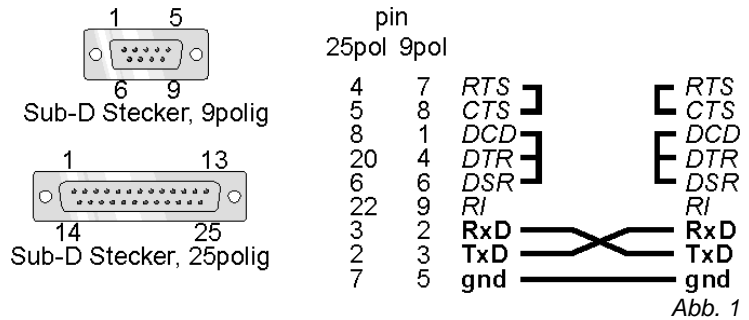
Datentyp	Beschreibung	Größe in Zeichen	Beispiele
0 - boolean_old	wahr / falsch in der Form sechs Nullen (ASCII 48) oder Einsen (ASCII 49)	6	000000 entspricht falsch 111111 entspricht wahr
1 - u_integer	vorzeichenlose Integerzahl mit sechs Stellen (führende Nullen)	6	000042 123456 001200
2 - u_real	Festkommazahl mit vier Vorkomma- und zwei Nachkommastellen, normiert auf 0,01 (führende Nullen)	6	001570 entspricht 15,70 000020 entspricht 0,2
3 - u_expo	positive Exponentialzahl (führende Nullen)	6	1.2E-2 entspricht $1,2 \cdot 10^{-2}$ 0005E8 entspricht $5 \cdot 10^8$
4 - string	beliebige Zeichenkette mit ASCII-Zeichen ≥ 32 (dezimal)	6	hallo! TC_600 hgnrfx
5 - vektor	mehrere Parameternummern mit Werten, angeführt von der Zahl der folgenden Parametern (zwei Stellen)	n	02001000000702120 
6 - boolean_new	wahr / falsch in der Form eine Null (ASCII 48) oder Eins (ASCII 49)	1	0 entspricht falsch 1 entspricht wahr
7 - u_short_int	vorzeichenlose Integerzahl mit drei Stellen (führende Nullen)	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 Exponentialzahl; die ersten vier Stellen beinhalten die mit 1000 multiplizierte Mantisse, die letzten beiden den Exponenten mit Offset 20	6	100023 entspricht 1.000E3 456711 entspricht 4.567E-9
11 - string16	Beliebige Zeichenkette mit ASCII-Zeichen ≥ 32 (dezimal)	16	BrezelBier&Wurst 0123456789ABCDEF
12 - string8	Beliebige Zeichenkette mit ASCII-Zeichen ≥ 32 (dezimal)	8	Pfeiffer 01234567 >Vacuum<

4. Allgemeines über Schnittstellen

Die Konfiguration von Schnittstellentopologien, -kabeln, -steckern usw. ist weitgehend anwendungs-spezifisch. Im Folgenden sollen aber einige unverbindliche Hilfestellungen gegeben werden.

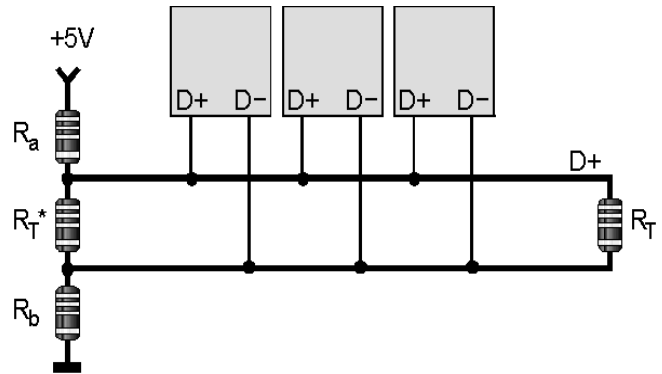
4.1. RS 232

Die RS232 Schnittstelle wurde ursprünglich als Modem-Verbindung konzipiert und verfügt daher über weit mehr Signale, als man in vielen Anwendungen ohne Modem benötigt. In der Regel genügen die beiden auf Masse bezogenen Datenleitungen TxD und RxD. Einige andere Signalleitungen müssen aber ebenfalls mit definierten Pegeln versehen werden, um dem RS232-Treiberbaustein bei Anwendungen ohne Modem nicht vorhandene Signale zu simulieren. Mit dieser Art Schnittstelle ist nur eine Punkt-zu-Punkt-Verbindung von genau zwei Geräten möglich. Abb. 1 zeigt ein Beispiel einer RS232-Verbindung. Es empfiehlt sich, geschirmte Leitung zu verwenden. Die Busabschlußwiderstände sind oft in den Treiberbausteinen integriert. Allgemein übliche Stecker sind Sub-D 9polig oder 25polig.



4.2. RS 485

Mit der RS485-Schnittstelle lassen sich bis zu 32 Geräte über zwei Leitungen miteinander verbinden, wobei nie mehr als ein Gerät zur selben Zeit senden darf. Alle Geräte sind mit ihrem D+ Anschluss mit der D+ (oder DO/RI) Leitung und mit ihrem D- Anschluß mit der D- (oder DO/RI) Leitung verbunden. Da an den bus mehrere Geräte angeschlossen werden können, sind in der Regel keine Busabschlußwiderstände in den Treiberbausteinen integriert. Diese müssen an den beiden entferntesten Enden des Bus angeschlossen werden. Die Werte der Widerstände richten sich nach dem Wellenwiderstand des verwendeten Kabels. Um eine optimale Funktion der Bustreiber zu gewährleisten muß der inaktive Bus oft durch äußere Beschaltung auf einer Spannung von 200mV (logisch 1) gehalten werden.



Für die Praxis ergeben sich dann Widerstandswerte im Bereich $R_a=R_b=680\Omega$, $R_T=120\Omega$ und $R_T^*=130\Omega$ für verdrehte geschirmte Zweidrahtleitungen (vgl. Abb. 2). Als Bustopologie ist eine Hauptleitung mit möglichst kurzen Stichleitungen anzustreben.

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 $\geq 32^*$ 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
--------	--------	------------------	-------------	------	----------	----

Address: Address 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 \Rightarrow [2.1 master telegrams and 3. data formats](#)
- Data request \Rightarrow [2.2 slave telegrams and 3. data formats](#)
- Error messages \Rightarrow [2.2 slave telegrams](#)

Checksum: The sum of all ASCII symbols up to preceding checksum modulo 256 (decimal), for example sum = 786. $786 \text{ modulo } 256 = 18 \Rightarrow$ 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:

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

Data request:

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

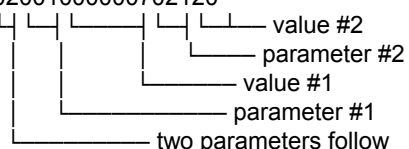
2.2. Slave-Telegrams

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

3. Data Formats

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

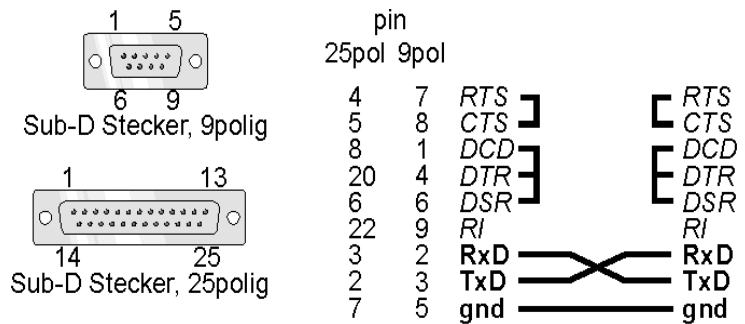
Data type	Description	Size in symbols	Examples
0 - boolean_old	true / false in the form six zeros (ASCII 48) or ones (ASCII 49)	6	000000 corresponds to false 111111 corresponds to true
1 - u_integer	pre-symbol-less integer number with six positions (leading zeros)	6	000042 123456 001200
2 - u_real	fixed 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 $1,2 \times 10^{-2}$ 0005E8 corresponds to 5×10^8
4 - string	optional symbol chain with ASCII symbols ≥ 32 (decimal)	6	hallo! TC_600 hgnrxf
5 - vector	several parameter numbers n with values starting with the number of the following parameters (two positions)	n	02001000000702120 
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.

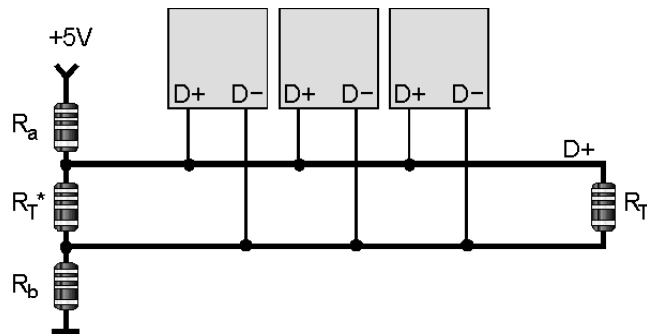


However, some other signal cables must also be provided with defined levels so as not to simulate current signals on the RS232 driver component where

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 $R_a=R_b=680\ \Omega$, $R_T=120\ \Omega$, and $R_{T^*}=130\ \Omega$, 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.

Your Vacuum Technology Experts in



Turbo Pumps



Rotary Vane Vacuum Pumps



Roots Pumps



Dry Vacuum Pumps



Leak Test Units



Valves



Flanges, Feedthroughs



Vacuum Measurement



Gas Analysis



System Technology



Service



Pfeiffer Vacuum Technology AG · Headquarters/Germany

Tel. +49-(0) 64 41-8 02-0 · Fax +49-(0) 64 41-8 02-2 02 · info@pfeiffer-vacuum.de · www.pfeiffer-vacuum.net