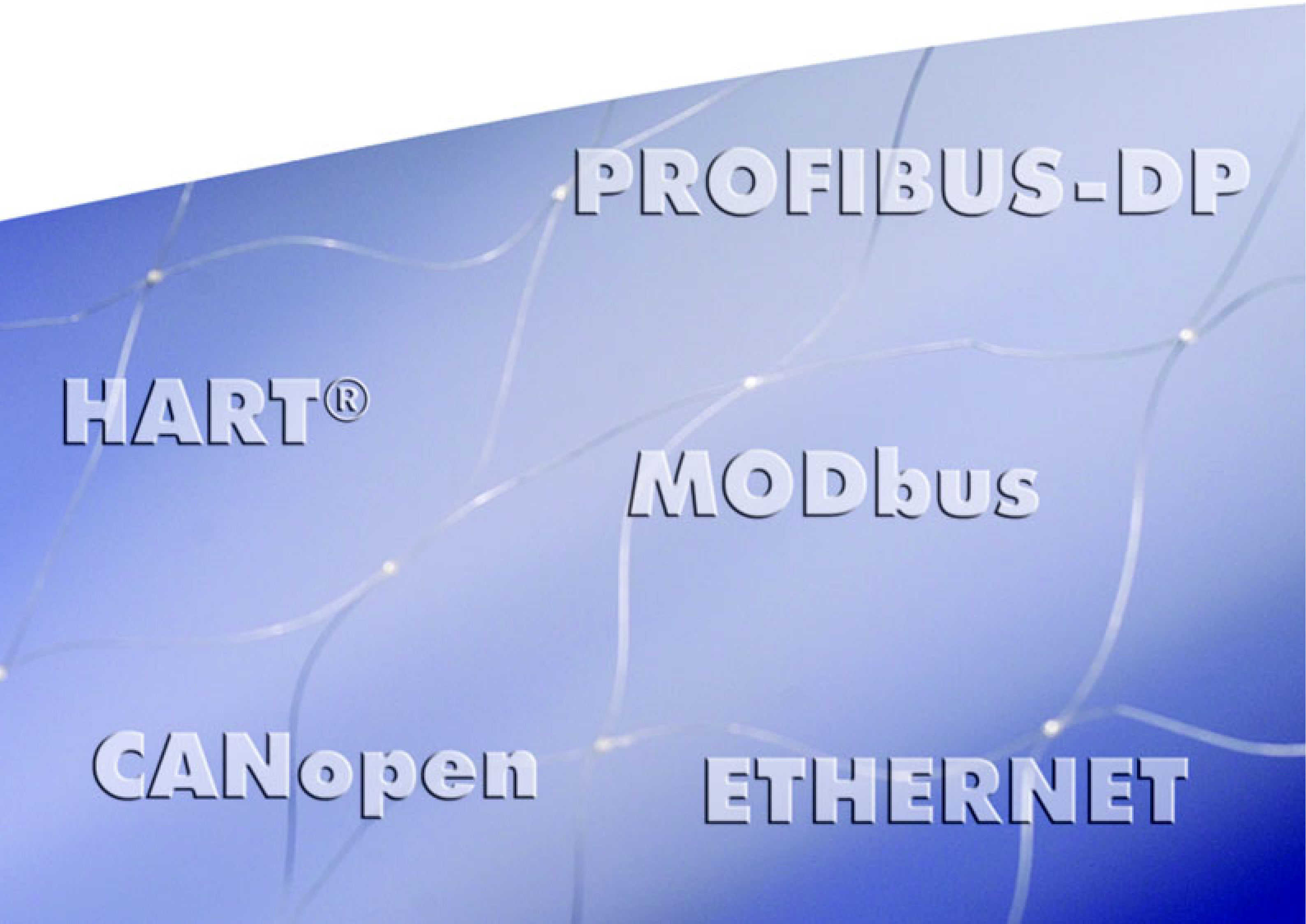
Digital Interfaces and Bus Systems

Fundamentals and practical advice for the connection of field devices

Manfred Schleicher



Digital Interfaces and Bus Systems

Fundamentals and practical advice for the connection of field devices to MODbus, PROFIBUS-DP, ETHERNET, CANopen and HART®

Manfred Schleicher

Preface

For a considerable time, JUMO has been offering components that have the facility to be connected to bus systems.

This book provides you with important fundamentals for making such connections. While writing it, we continually asked ourselves what information the user really needs, and kept to the things that really matter.

Chapters 1 to 3 establish some foundations in the field of digital communications.

And especially in **Chapter 4** we have provided specific advice on linking. However, the contents must be seen as a descriptive guide. For further information, please refer to the corresponding operating manuals.

Chapter 5 shows the principles involved in making a modem connection.

This book has been created with all due care. We cannot, however, accept liability for any errors.

Special thanks are due to all those colleagues who assisted in revising this book.

Fulda, May 2005

Manfred Schleicher

JUMO GmbH & Co. KG
Moltkestraße 13 - 31
D-36039 Fulda, Germany
Phone +49 661 6003-0
Fax: +49 661 6003-500
e-mail: Manfred.Schleicher@jumo.net
Internet: www.jumo.net

5th revised edition, reprinting permitted with source citation!

Part number: 00392023
Book number: FAS603
Date of printing: 06.05
ISBN-10: 3-935742-03-7 (gültig bis 31.12.2006)
ISBN-13: 978-3-935742-03-0 (gültig ab 01.01.2007)

Contents

Contents

Signal types and data encoding

1.1 Analog and digital signals

Manufacturers of automation engineering components have now almost stopped supplying instruments with analog signal processing. The devices offered are almost exclusively microprocessor-controlled. In addition to other advantages, such as programmability, a digital interface can be implemented in such devices. An interface can be used to transmit the process variables.

We will take a look at the principles of analog and digital signals.

A measurement, e. g. a temperature, is converted by a sensor + transmitter into a signal that is proportional to the temperature. The signal could, for instance, be a 4 — 20mA current. Each temperature value corresponds to a unique signal amplitude (e. g. 20 to 200°C / 4 — 20mA). While the temperature continues to change, so does the signal in proportion (Fig. 1).

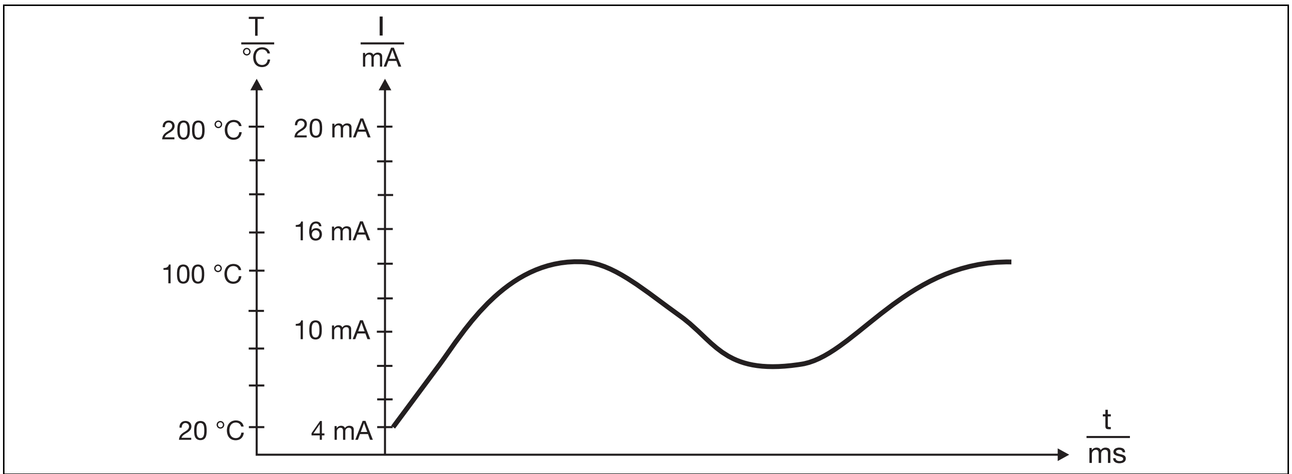


Fig. 1: Analog signal with a continuously changing amplitude

The standard signal (in Fig. 4 — 20mA; signal types 0 — 20mA, 2 — 10V and 0 — 10V are also common) can be used by a wide variety of devices. Any change in the measured value will instantly be detected by the current in all connected devices.

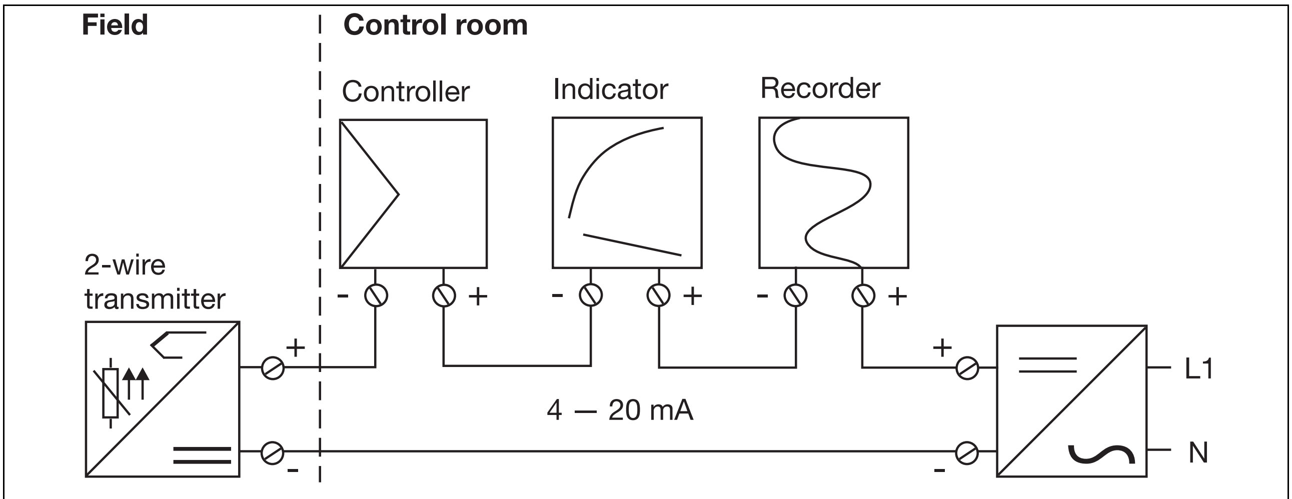


Fig. 2: Analog signal path for a current signal

1 Signal types and data encoding

Note
The signal path shown here is an analog one. Nevertheless, in most cases the transmitters and the attached components operate on the basis of a microprocessor (the 2-wire transmitter acquires the signal in analog form from the sensor, converts it into a digital signal for subsequent processing, and then reconverts it into an analog signal for the output).

If a signal is being digitally processed, then the continuous analog value must be divided into discrete signal levels. The values are polled within a defined time – the sampling time. The conversion of the signal is made by an analog/digital converter (A/D converter).

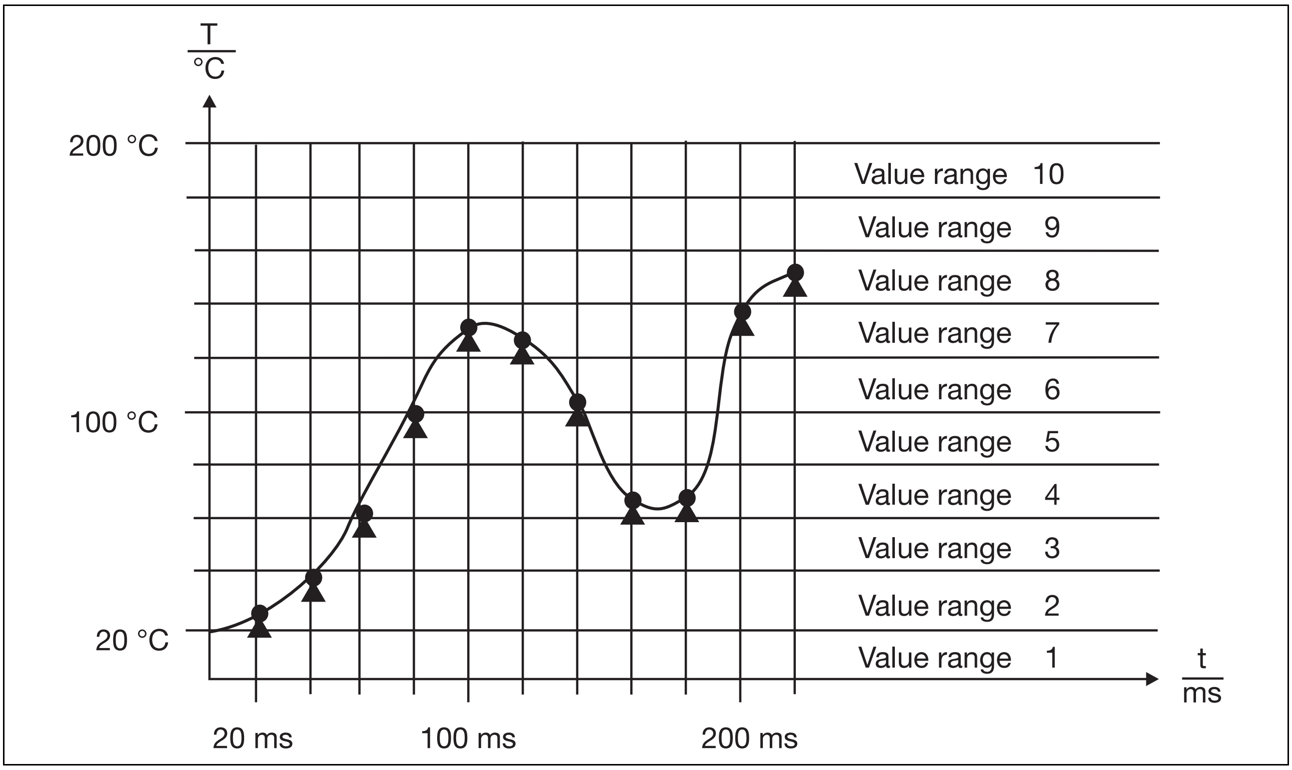


Fig. 3: Digitalized measurement signal

Fig. 3 shows 10 signal levels and a sampling time of 20 ms. Such a digitalized signal would be very imprecise. At the time of printing, JUMO uses, among others, A/D converters that have 16-bit resolution. So there are 2^{16} = 65536 possible signal levels, which demonstrates that, in this case, the quantization error is very small. Typical sampling times are from 50 — 500 ms.

1.2 Data formats and encoding

An interface can be used to transmit only two states (LOW level or HIGH level). This smallest unit of information is called a BIT. The voltage levels that are used for LOW or HIGH depend on the interface.

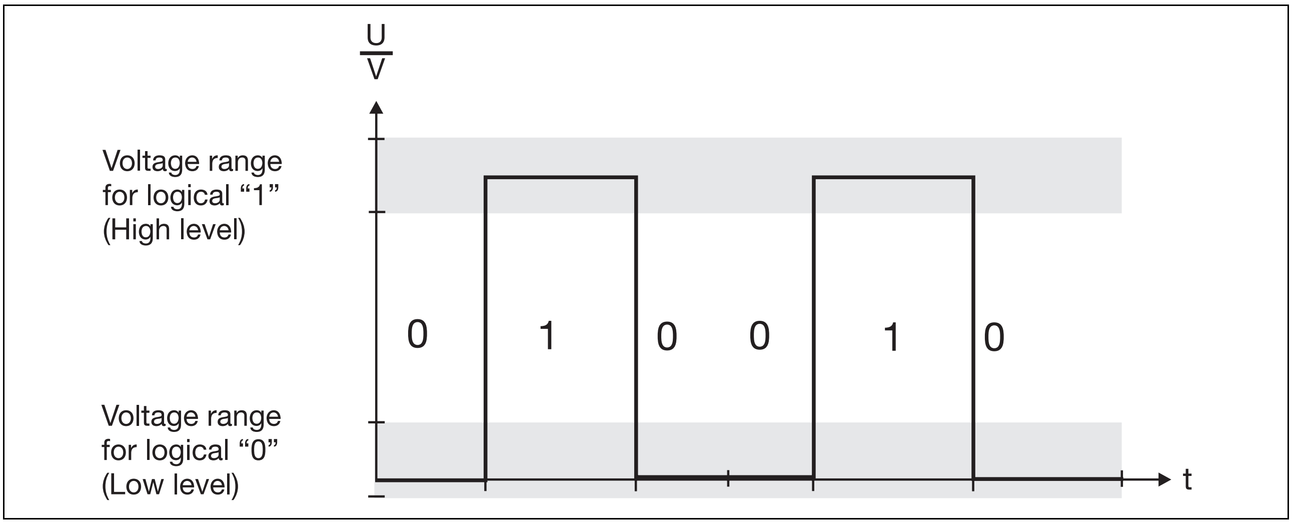


Fig. 4: Data transmission as bits, using different voltage levels

In practice, automation engineering devices never send single bits, but always at least 8 bits in a block. Such a block is known as a BYTE.

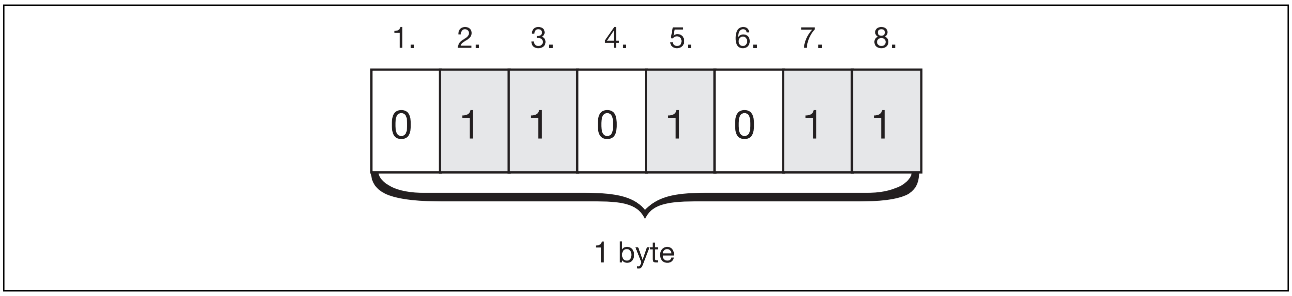


Fig. 5: Combining 8 bits into 1 byte

In the form of a byte, it is possible to transmit, for instance, the states of 8 switching outputs from a controller.

Characters are also transmitted as a byte. In this case, the ASCII code is used. To determine the corresponding value, the byte must be interpreted as a binary number.

Example: the transmitted byte is 00100110_2.

The binary value must now be converted into a decimal number. Each element of the binary number is assigned to a power of 2:

	2^7		2^6		2^5		2^4		2^3		2^2		2^1		2^0
Value of the position	128		64		32		16		8		4		2		1
The byte shown above	0		0		1		0		0		1		1		0
transmits the value	0	+	0	+	32	+	0	+	0	+	4	+	2	+	0
which corresponds to :							38_{10}								

1 Signal types and data encoding

If the transmitter encodes the number as an ASCII value, the following table can be used to determine the corresponding character:

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	nothing	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☹	STX	034	„	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(alarm)	BEL	039	´	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tabulator)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(vertical tab.)	VT	043	+	075	K	107	k
012	(page feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	P
017	◄	DCI	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	¶	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	–	SYN	054	6	086	V	118	V
023	↕	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(position indicator right)	FS	060	<	092	\	124	¦
029	(position indicator left)	GS	061	=	093	]	125	}
030	(position indicator up)	RS	062	>	094	↑	126	~
031	(position indicator down)	US	063	?	095	–	127	△

Table 1: ASCII code

From Table 1 we can see that the transmitted character was an “&”. The 128 characters shown in the table can be encoded with 7 bits. However, there are 8 bits available in one byte. And umlauts and a lot of special characters are not included. The code is therefore expanded to 8 bits, which means that it can encode a total of 2^8 = 256 characters.

The characters which are encoded by the numbers above 128 (extended ASCII code) are not shown in this table.

There are other codes apart from ASCII, e. g. the ANSI (**A**merican **N**ational **S**tandards **I**nstitute) code. Windows uses this ANSI character set, which matches the ASCII character set for the values 32 to 127. This character set is also used in JUMO instrumentation at the time of printing.

Occasionally, numbers are encoded as 2 bytes. Two bytes, which are also known as a WORD, are used to encode integer number in the range 0 — 65535. This is known as the “integer format”, and is used, for instance, for times and numerical values.

Measured values (such as temperature or pressure) are frequently transmitted in automation engineering. These are encoded according to the IEEE 754 standard, and transmitted as 4 bytes. The “float value” encoding covers a value range of 1.0E±38.

1.3 Hexadecimal system

The hexadecimal system is very important in information technology, since 4 bits (a half-byte) are frequently combined. Four bits can be used to encode 16 different states. If this is to be represented by a single digit, then this is not possible in the decimal system (largest number: 10).

All the possible states of 4 bits, and their values in the decimal and hexadecimal systems, are shown in the following table:

4 bits	Decimal value	Hexadecimal value
0000	0_{10}	0_{16}
0001	1_{10}	1_{16}
0010	2_{10}	2_{16}
0011	3_{10}	3_{16}
0100	4_{10}	4_{16}
0101	5_{10}	5_{16}
0110	6_{10}	6_{16}
0111	7_{10}	7_{16}
1000	8_{10}	8_{16}
1001	9_{10}	9_{16}
1010	10_{10}	A_{16}
1011	11_{10}	B_{16}
1100	12_{10}	C_{16}
1101	13_{10}	D_{16}
1110	14_{10}	E_{16}
1111	15_{10}	F_{16}

Table 2: Binary, decimal and hexadecimal numbers

The index 9_{10} or 9_{16} indicates that the number is in the decimal or hexadecimal system respectively. For hexadecimal number, 0x is frequently placed in front of the number.

0xA = A_{16} corresponds to the number A in the hexadecimal system. This has the value 10 in the decimal system (10_{10}).

A byte is often represented as a two-figure hexadecimal number, e.g.

Byte:	0110	1101		
Representation as two hexadecimal figures 0x	6	D	=	0x6D

Working with serial interfaces (RS232, 422, 485)

2.1 Serial and parallel interfaces

As a rule, data is transferred in a bus system as a serial transmission: the individual bits are transmitted one after another, usually over a 2-wire cable.

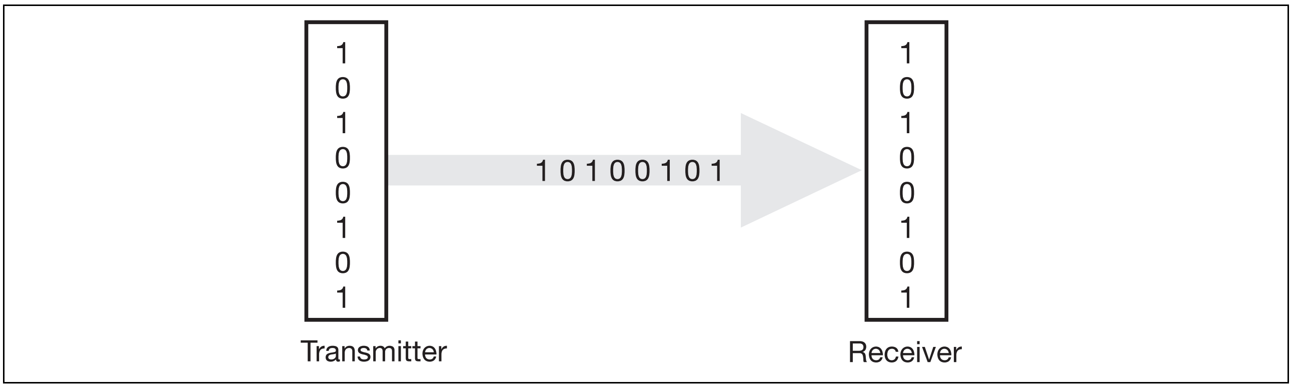


Fig. 6: Serial data transmission

In a few cases, the data in a bus system are transmitted in parallel: in parallel transmission, several bits are always transmitted simultaneously. Parallel bus systems are used in the field of measurement technology (IEC bus IEEE 488).

Fig. 7: Parallel signal transmission

The great advantage of a parallel bus is the high rate of transmission, since several bits are always sent at the same time. Against that must be set the high sensitivity to interference and high wiring costs. Parallel data transmission is suitable for short distances.

The RS232, RS422 and RS485 interfaces that are described below all function as serial transmission methods, and are used by JUMO for interfaces with the MODbus protocol.

2.2 RS232, RS422 and RS485 interfaces

2.2.1 The RS232 interface

A serial interface can be found in just about every PC (each one of the COM ports provides an RS232 interface).

Fig. 8 shows the pin assignments for the most frequently used connector (Sub-D):

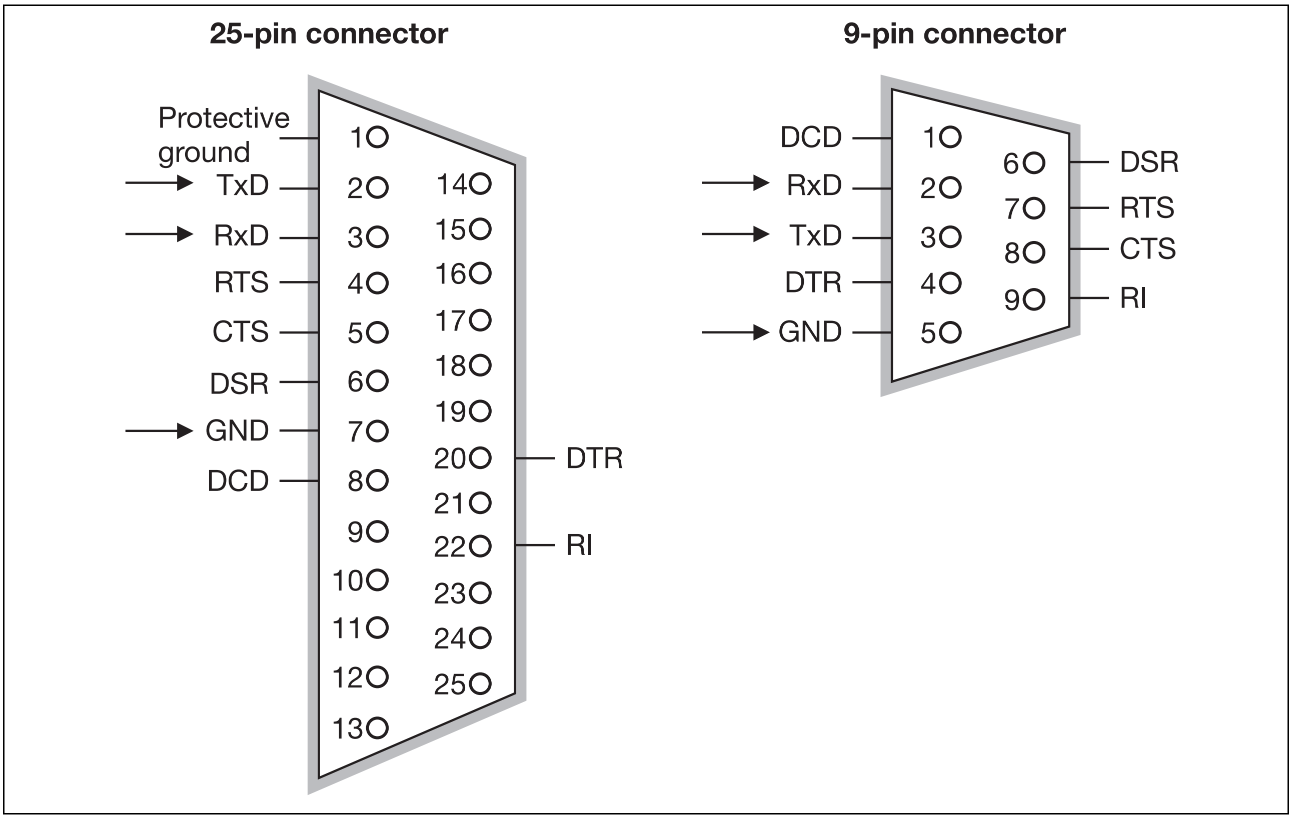


Fig. 8: Typical assignments for SUB-D (25- and 9-pin) for the RS232 interface

The RS232 was originally developed for connecting computers via telephone lines. This is the reason for the large number of signals that are defined.

For automation engineering, only 3 wires of the RS232 interface are normally required: TxD (Transmit Data) for transmitting, RxD (Receive Data) for receiving, and the GND (Ground) wire for the common ground potential (no transmission is possible without the GND wire).

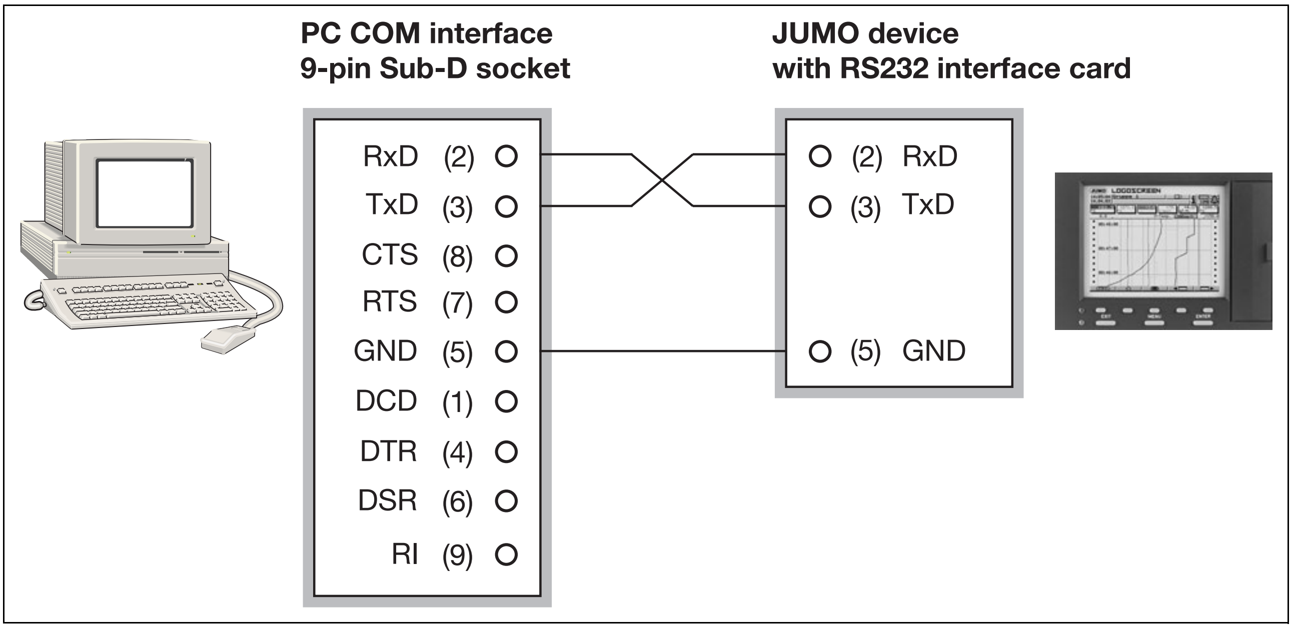


Fig. 9: Wiring a JUMO device to a PC, using the RS232 interface

In Fig. 9 it is noticeable that the TxD from the PC is connected to the RxD for the device. This makes sense, when you consider that the PC transmits on the TxD wire and the device is receiving this signal (therefore RxD). The same applies in the other direction: TxD (from device) to RxD (for the PC).

The LOW level (logical 0) of the RS232 interface lies within the range from 3 to 15V, the HIGH level (logical 1) within the range from -3 to -15V.

The maximum cable length for an RS232 interface is 30m. The RS232 interface has no bus capability, and can only be used for point-to-point connections (two devices connected together).

JUMO uses the RS232 interface for the following applications:

1. Devices that can be configured with the help of a configuration (Setup) program,
   the data are transferred from the PC to the device over a setup cable, via the RS232 interface.

Fig. 10: Controller JUMO DICON 500 with a setup cable

The connector shown in the foreground is plugged into an available COM port on the PC (RS232 interface) and converts the signal levels of the RS232 interface to TTL logic levels. The controller interface uses TTL logic.

2. JUMO supplies paperless recorders that store the measurements from the attached sensors in an internal electronic memory. On request, the measurement data can be transferred under timed control from the recorder to the PC via the interface. If there is only one paperless recorder in the system, then it can be connected to the RS2323 interface.

The RS232 interface is sensitive to interference, as opposed to the RS422 and RS485 interfaces that follow. The reasons for this are explained in the following section.

2.2.2 The RS422 interface

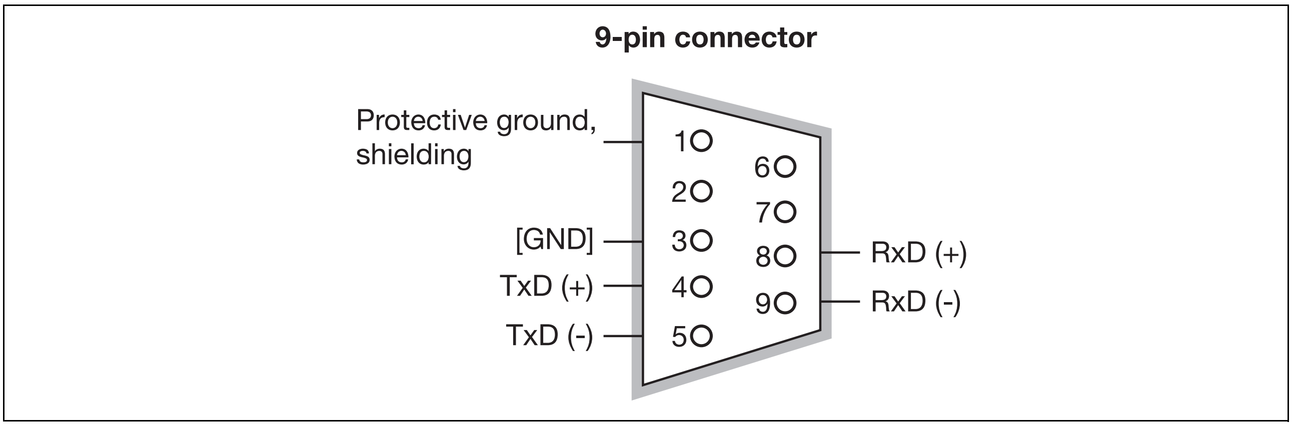


Fig. 11: Typical assignments for a 9-pole Sub-D connector used for an RS422 interface

For the RS422 interface, data transmission takes placed through a 4-wire cable.

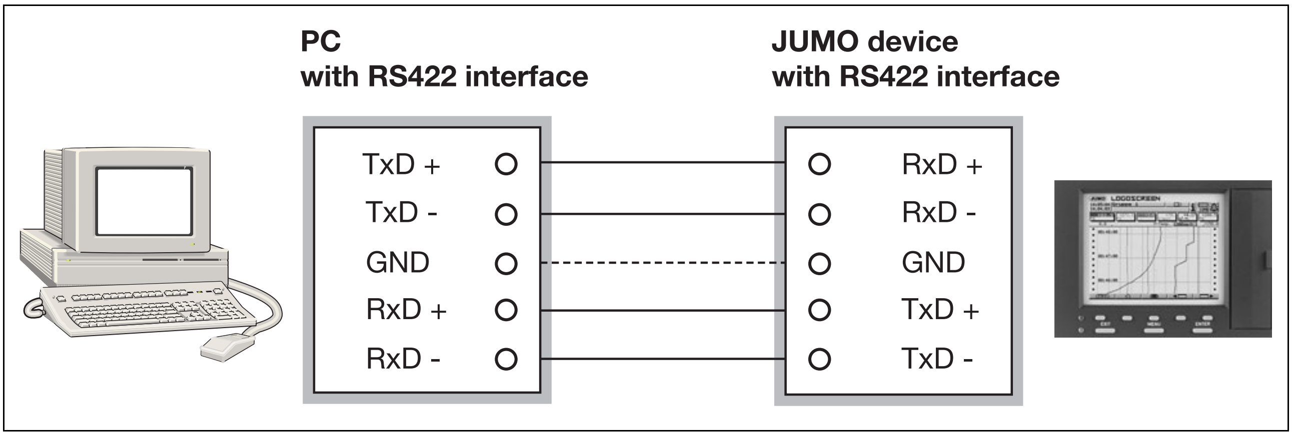


Fig. 12: Data transmission with the RS422 interface

All stations transmit on the TxD + and TxD - wires, and receive on RxD + and RxD -. The GND wire is only necessary for larger potential differences.

The RS422 interface (like the RS485 interface) is less sensitive to interference than the RS232 interface. This is because, the measured signal is always the difference between the levels on the two wires, e. g. if TxD + has +5V and TxD - has -5V, then this corresponds to logical "0". Likewise, if TxD + has -5V and TxD - has +5V, this corresponds to logical "1". Any electromagnetic interference will increase the potential on the two wires by the same amount. So the interference will not have any affect on the differential signal (as a comparison: on an RS232 interface, the interference would only affect the signals, not the ground potential).

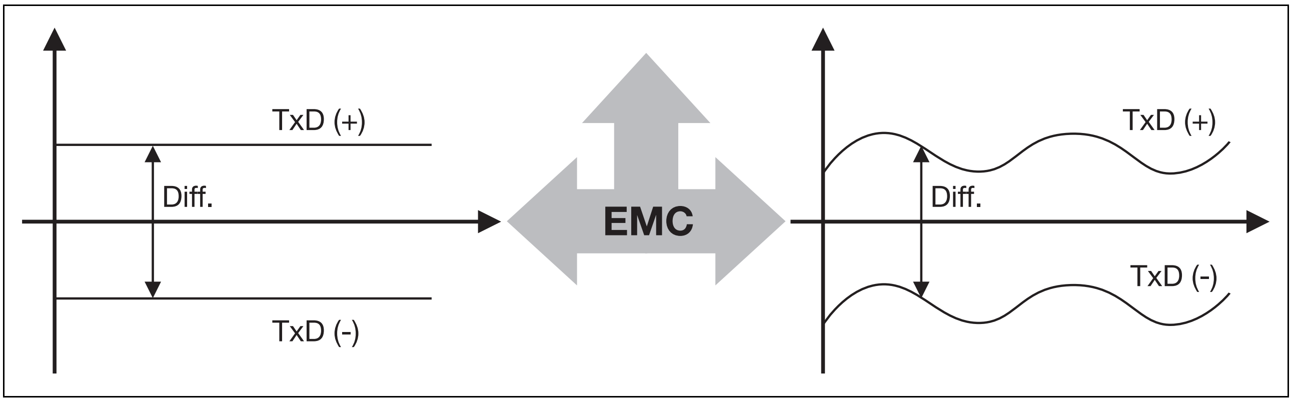


Fig. 13: High interference-immunity RS422 interface

The RS422 interface is also bus capable. Up to 32 stations can be connected together.

How can additional stations be connected to the RS422 interface?
On the field device (Fig. 12), all the terminals (TxD +, TxD -, RxD + and RxD -) are connected to the same terminal on the next device.
The distance from the first station to the last station can be up to 1200m. If there are more than 32 stations available, or there are stations more than 1200m apart, then repeaters (power amplifiers) must be used (Fig. 14). A repeater counts as one device in both segments to which it is connected.

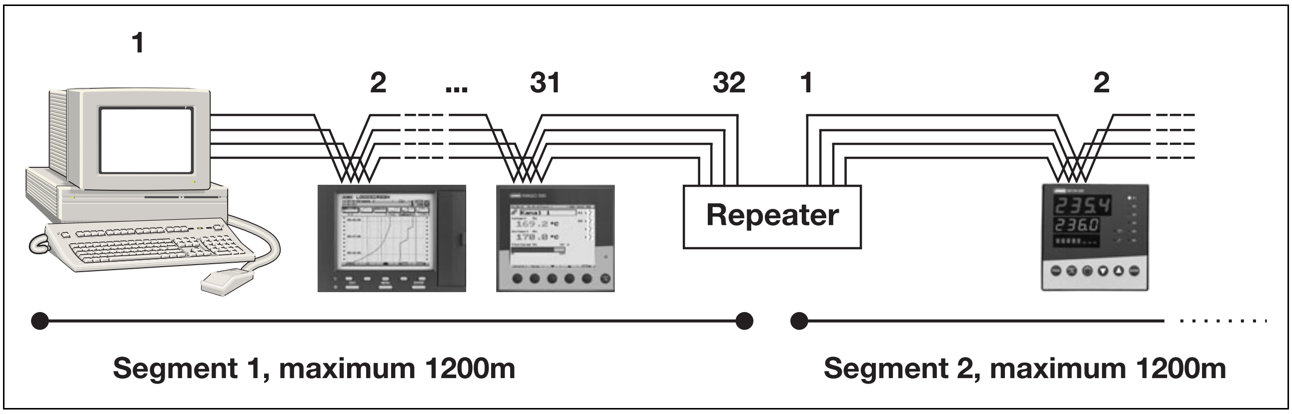


Fig. 14: Using a repeater with an RS422 interface

2.2.3 The RS485 interface

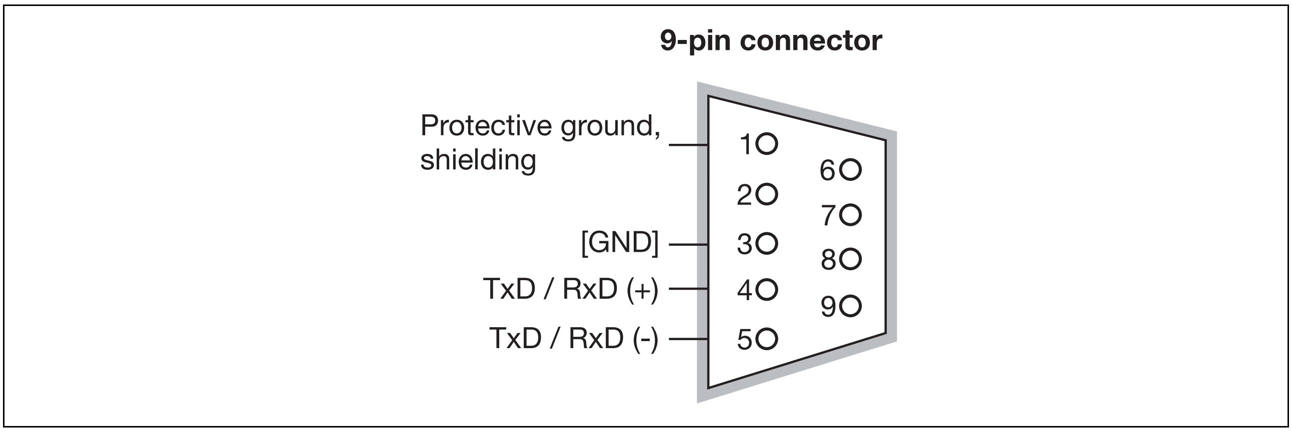


Fig. 15: Typical assignments for a 9-pole Sub-D connector used for an RS485 interface

The RS485 interface has similar characteristics to the RS422 interface. Both have the same signal levels. Distances of up to 1200 m can be achieved, and 32 stations can be connected together. As far as repeaters, high interference immunity and voltage levels are concerned, the same applies as for the RS422 interface.

The major difference to the RS422 interface (and also the reason for its much more widespread application), is that when using the RS485 interface, the stations are connected by just two data wires.

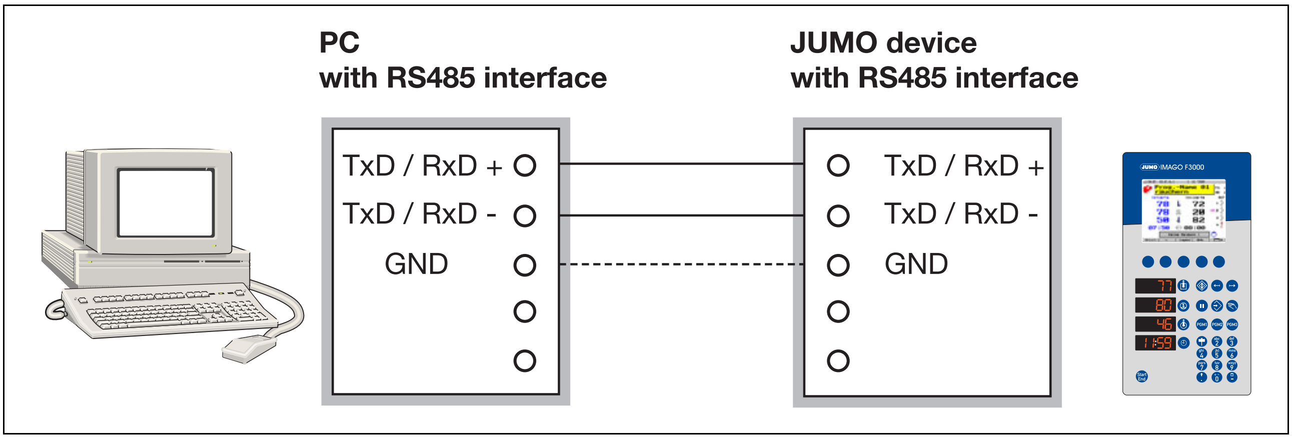


Fig. 16: Data transmission with the RS485 interface

The stations transmit and receive on the TxD / RxD + and TxD / RxD - wires. This results in a considerable reduction in installation costs. With the RS485 interface, the stations have to switch states: for instance, suppose the PC is sending data through the pair of wires to the device. During this time, the device uses the wire pair for receiving. When the message has been transmitted, the device switches state from receive to transmit, and the PC switches state from transmit to receive.

2.2.4 Connecting cable for the RS485/RS422 interface

Chapter 3 *"Access procedure and transmission media"* describes the connecting cable in detail. For RS485 / RS422 interface applications, shielded twisted-pair connecting cables are used. One pair is formed by the RxD/TxD + and RxD/TxD - of the RS485 (or, for the RS422 interface, the first RxD + and RxD - pair) and the second pair is formed by the TxD + and TxD - wires.

The reason for using twisted-pair cables is explained in Chapter 3 *"Access procedure and transmission media"*.

2.2.5 Connecting stations via the RS485 interface

When using the RS422 or RS485 interface, the stations are joined together in a linear or bus topology.

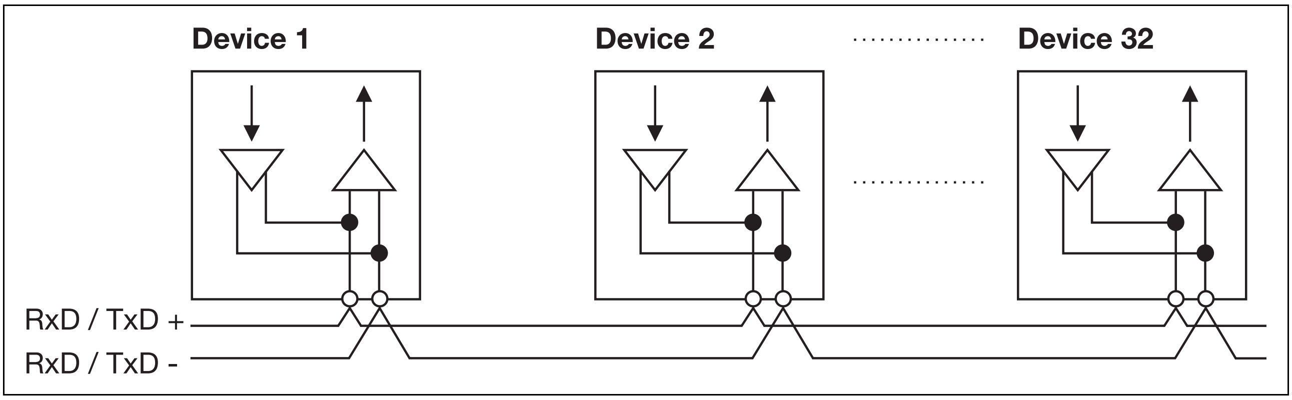


Fig. 17: Linear topology for an RS485 interface

For the RS485 interface, this means that the RxD/TxD + wire is led from the first station to the second, from the second to the third, and so on. Likewise, the RxD/TxD - wire is led from the first station to the second, from the second to the third, and so on.
When using the RS422 interface, the other terminals are linked together in the same way (Fig. 14).

Stub cables must be avoided at all costs. The incoming and outgoing cable must be led right up to the field device (distribution boxes must not be used).

For EMC-conform wiring (with separate grounding/potential equilibration for every switchgear cabinet or field device), best results are usually achieved when the shielding of the bus cable makes a large-area contact to the grounding of the cabinet or device.

A termination resistor must be activated on the first and last stations. If JUMO field devices are used, it is possible to do without the resistor, since for some versions it is already integrated into the interface.

2.2.6 Interface converter

Most PCs are delivered with an RS232 interface (COM port).
If you want to connect up field devices with RS485 interfaces, there are basically three options:

1. Use an interface converter (RS232 - RS485/RS422)
   An interface converter is connected to the COM port. It converts the RS232 signal levels to the levels for the RS485 or RS422 interface.

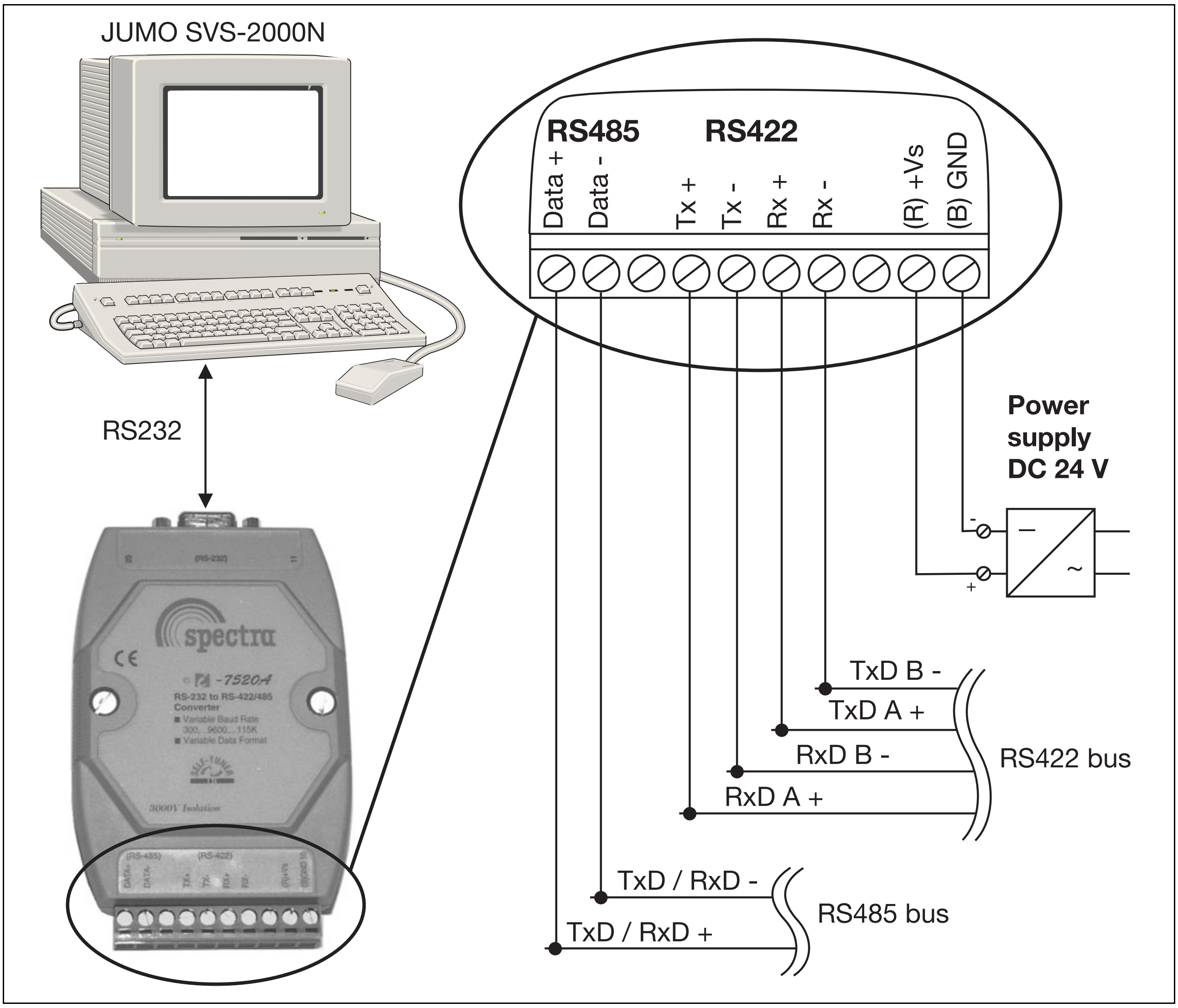


Fig. 18: Interface converters for RS232 to RS422/RS485 interfaces

Fig. 18 shows an interface converter from Spectra. Connecting the interface converter is quite simple: the cable for the RS232 interface is included in the converter package. Now you can either connect field devices with an RS485 interface (RxD/TxD + to Data + and RxD/TxD - to Data -) or appropriate field devices to the RS422 interface. No settings have to be made on the interface converter.
When an interface converter is used, it is important that it automatically detects the direction of the data flow.

2. Using an interface converter (USB - RS485/RS422)
   The converter described here is available with an USB interface, instead of the RS232 interface. The connection of the field devices is identical to that for the converter described under 1

(above). This interface converter is used with PCs and laptops that are not equipped with COM ports. The drivers that are supplied must be installed before the interface converter can be used.

3. Using a card with an RS485 interface
   PC cards with RS485/RS422 interfaces are commercially available. If such cards are installed, the field devices can be connected directly to the PC.

2.2.7 Configuration of interfaces – the UART

Parallel data transmission is used between the components of a field device. Serial transmission is used for the data in a bus system. The UART is the component in a field device that is responsible for converting a parallel transmission into a serial transmission.

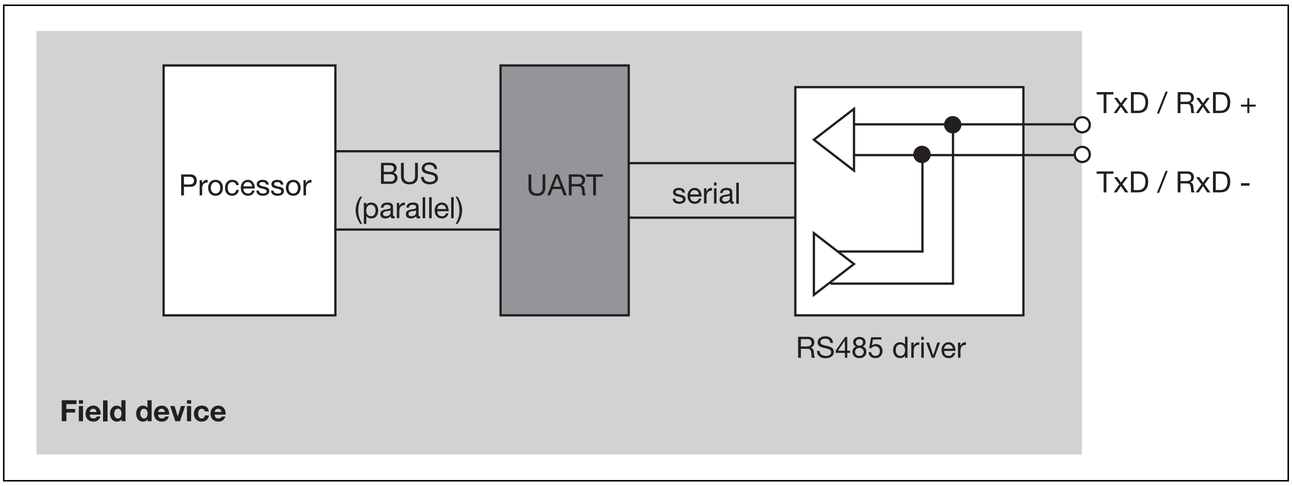


Fig. 19: UART application in an RS485 interface

The UARTs of the field devices that are connected to the bus must all operate at the same rate. The rate of data transmission is defined in baud. The baud rate is the number of changes of state per second. Since, in a serial bus system, one bit can be transmitted per change of state, the baud rate is equivalent to the number of bits transmitted per second (bps).

It is necessary for every station to have an address assigned. There must only be one instance of each address in the system.

In most bus systems, no further configuration of the UART has to be carried out.

Further configuration of the UART
For instance, further settings are made for the MODbus (discussed later). The UART transmits the information bytewise, and even if a measurement (represented by a double word, or 4 bytes) is transmitted, the data are transmitted one byte at a time.

Let’s take a closer look at these bytes.

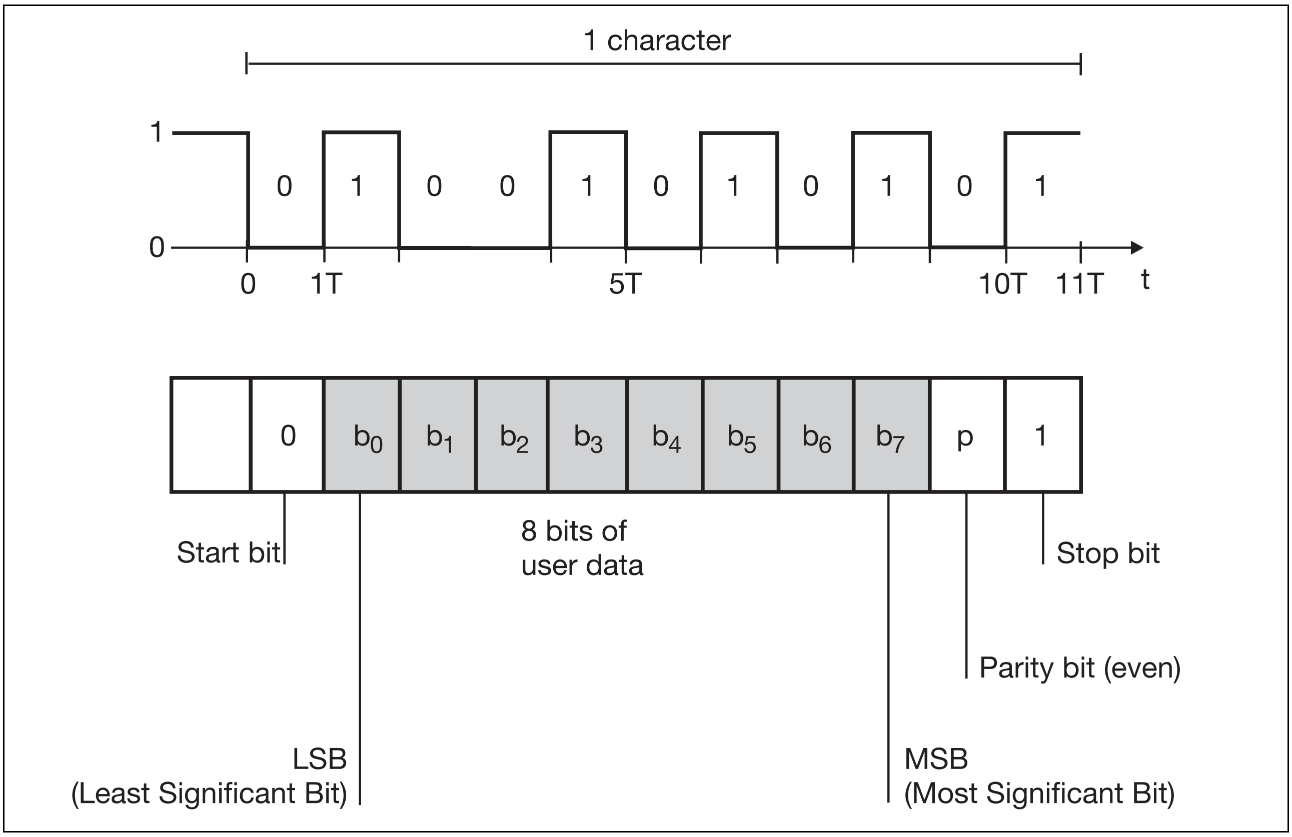


Fig. 20: Transmitting a byte

As mentioned above, all stations must be set to the same baud rate. However, the stations do not necessarily operate synchronously. So a start bit is transmitted before the user data, to establish synchronization (Fig. 20).

This is followed by one byte of user data.

This is followed by the parity bit. It can, for instance, be set to “odd”: the number of bits transmitted in the packet with the value 1 is expanded to an odd number. In this case, the receiver must also be set to “odd” parity: the information will only be accepted if the receiver receives an odd number of bits with the value 1, otherwise it will be rejected. Faulty data transmission can thus be filtered out. The parity check can alternatively be set to “even” (as in Fig. 20). In this case, the number of “1”s is expanded to an even number. If the parity setting is “none”, then there is no parity check. As we will see later, considerably more reliable tests are applied at a higher level, to ensure error-free transmission. Regardless of which setting is made (odd, even, none): all stations must have the same setting.

The stop bit provides the information that the message has now been sent. Here too, different settings can be made (1 or 2 stop bits etc.).
And here too: the same settings must be made for all stations.

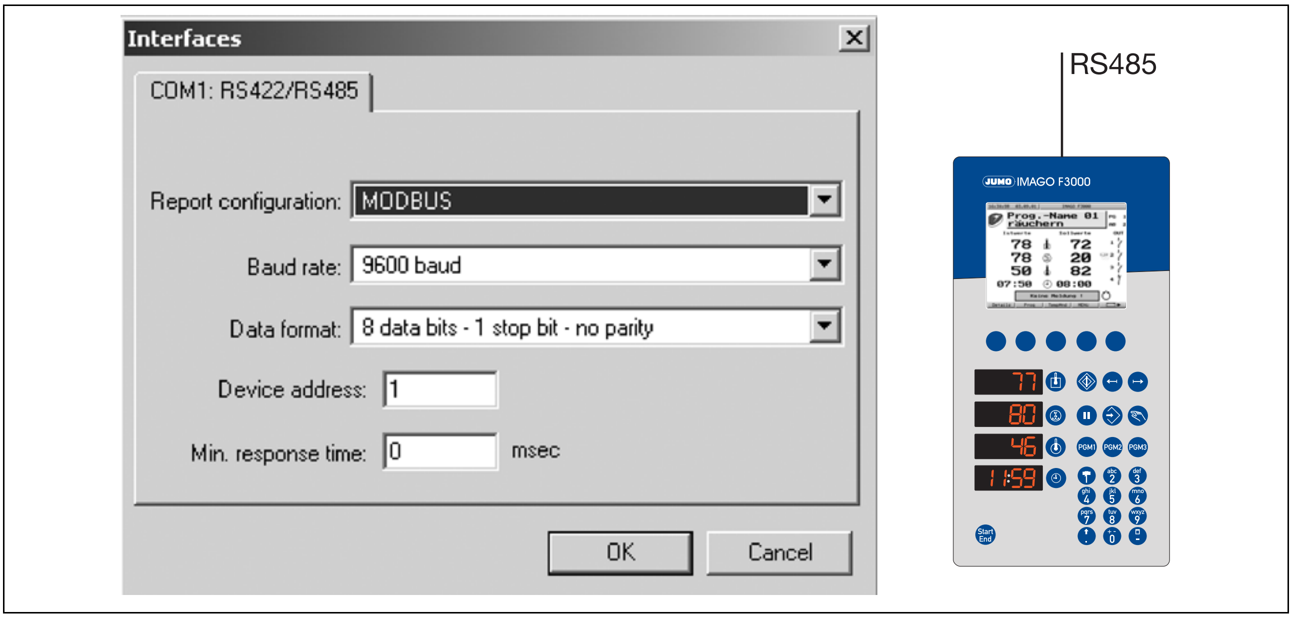


Fig. 21: Configuration of the interface (MODbus via RS485 interface) for the JUMO IMAGO F3000

Note
With an RS485 interface: if a master (e. g. a PC) requests a value from a field device, then the master must be switched over from transmit to receive. The parameter “Minimum response time” (Fig. 21) can be used to set a minimum waiting time for the device before it starts transmitting.

Access procedure and transmission media

3.1 Access procedure

In a bus system, the stations are usually arranged in a bus topology. All components operate, for example, over a 2-wire cable. However, this cable can only be available for one station at a time. The access procedure regulates which station is allowed access to the bus, and at which time.

Let's take a look at the various access procedures.

3.1.1 Master/slave procedure

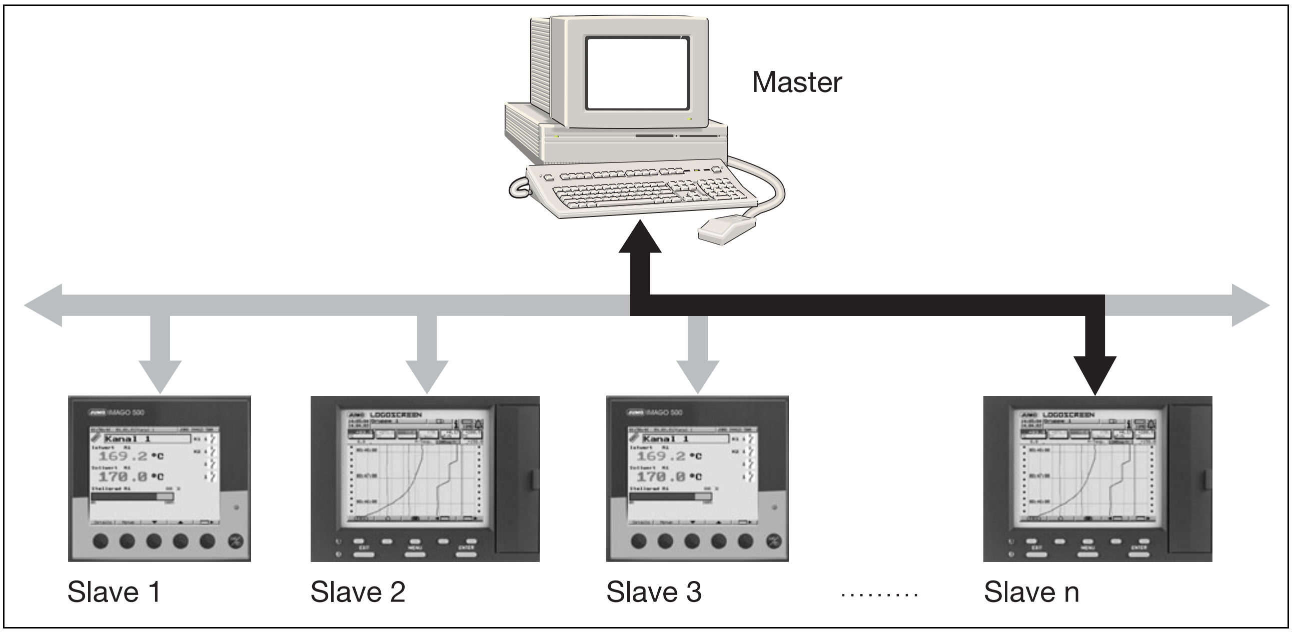


Fig. 22: Master/slave procedure

In the master/slave procedure, a master controls the communication on the bus: starting, for instance, with instructions for Slave 1. This slave follows the instructions and confirms their execution (e. e. acceptance of a setpoint). Next, the master requests relevant values from the same slave (for instance, the slave transmits 2 analog values). The master communicates in this way with all the slaves that are connected. When the data exchange with the last slave has finished, the master starts again with Slave 1. The time that is required for an exchange with all the slaves is known as the bus cycle time. The bus cycle time can vary between milliseconds (PROFIBUS-DP) and several seconds (MODbus).

3 Access procedure and transmission media

3.1.2 Token-passing procedure

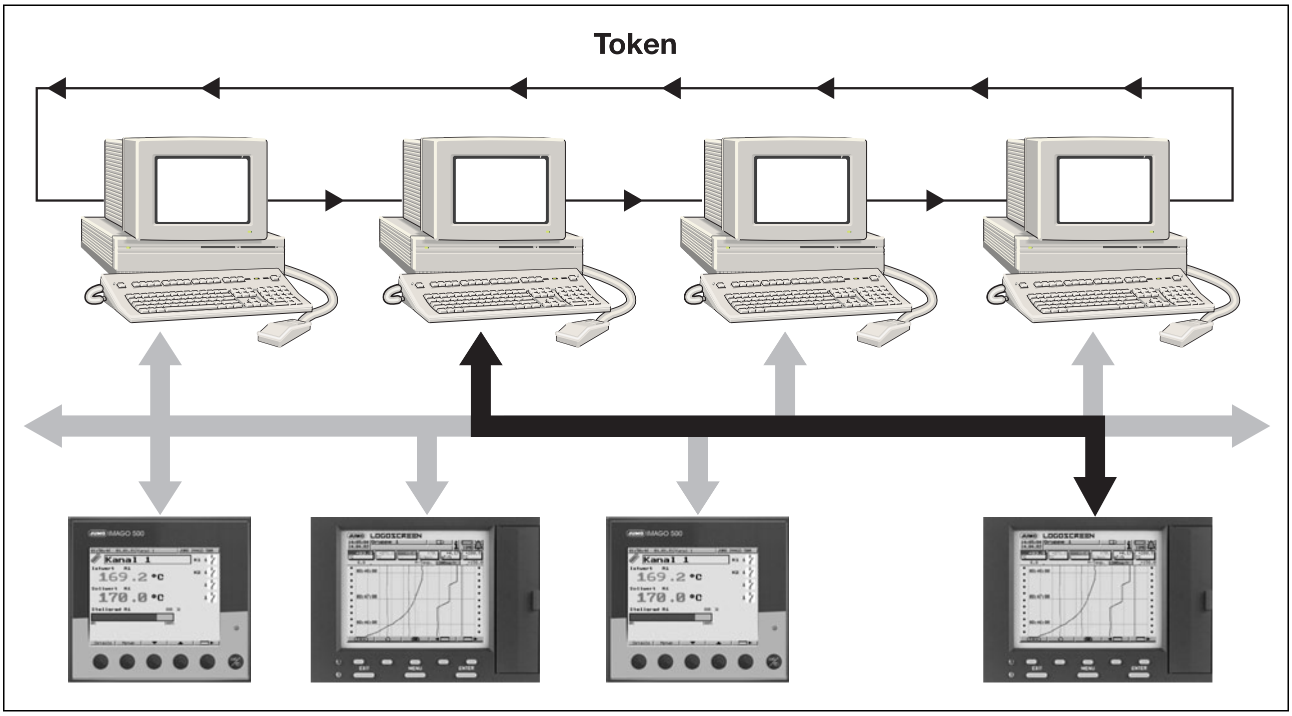


Fig. 23: Token-passing procedure

In the token-passing procedure too, a master simultaneously has control of the bus. The difference is, that in this case there is more than one master in the system. There is a token (access right) in the system. This is passed around from one master to the next. Only the master which holds the token can exchange data with the slaves. The token-passing procedure is used comparatively rarely. One example is in PROFIBUS-DP.

3.1.3 CSMA procedure

If a system operates with the CSMA procedure (**C**arrier **S**ense **M**ultiple **A**ccess), then all stations on the bus have equal rights: any station can start transmitting on its own accord. The principle is, that every station checks whether another station is transmitting at the moment, before it starts transmitting. Problems arise if two stations start transmitting simultaneously. There are various possibilities for dealing with this situation, which is known as a collision:

CSMA-CD

When using the CSMA-CD access procedure (**C**arrier **S**ense **M**ultiple **A**ccess **C**ollision **D**etection), if two stations transmit simultaneously, then the message will be destroyed. The stations recognize the simultaneous transmission, and repeat the transmission on a random basis. If another collision occurs, there will be a fresh transmission etc. ETHERNET, for instance, works on this principle.

CSMA-CA

When using the CSMA-CA procedure (**C**arrier **S**ense **M**ultiple **A**ccess **C**ollision **A**void), the station that is transmitting the message with a lower priority will disconnect from the bus. Thanks to the arrangement of the protocol and the electrical characteristics of the interface, the higher-priority message will obtain access and be transmitted without disturbance.
You can read more about this procedure in Chapter 4.4 *"CAN und CANopen"*.

3.2 Transmission media in automation engineering

In automation engineering, coaxial cables and optical fiber cables are used occasionally. Sometimes, data are also transmitted by radio or infrared signals. However, the standard transmission medium is a shielded, twisted-pair cable.

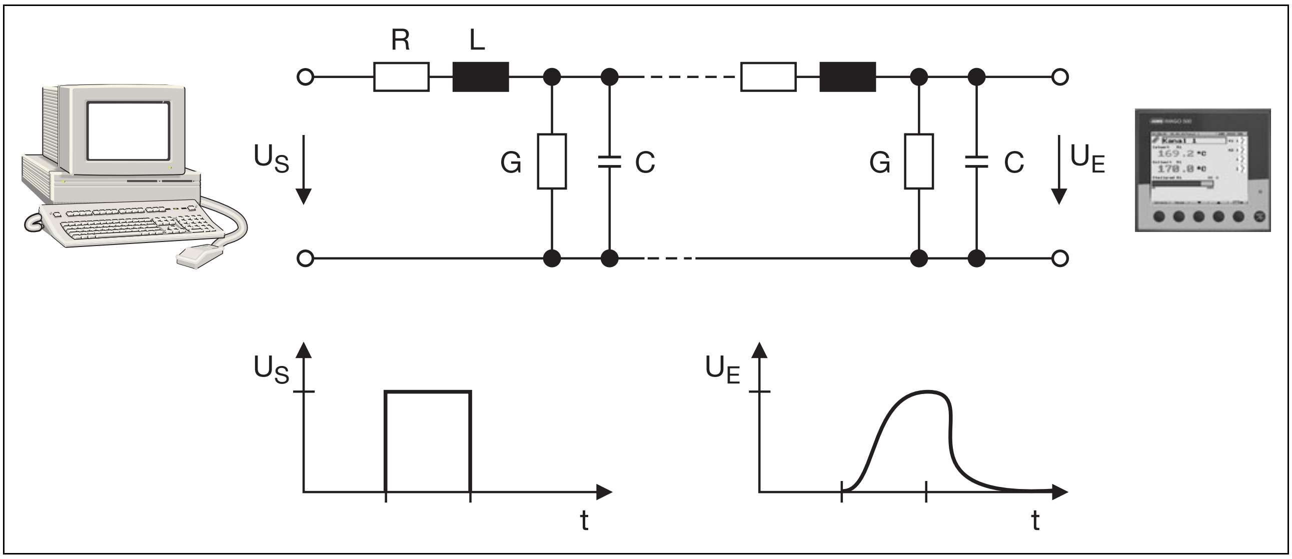


Fig. 24: Equivalent circuit of a cable

The distributed inductance and capacitance of a cable, shown in Fig. 24. effectively form a low-pass filter. This means that high frequencies will be attenuated or not transmitted at all.

The cable inductance, but most of all its capacitance, must be kept to a low level, since the low-pass effect will otherwise be excessive and information will be lost (e. g. the cable for an RS485 interface that is used for PROFIBUS-DP must have a distributed capacitance of <30 pF/meter).

The low-pass effect increases as the cable becomes longer and the baud rate is increased. So as the cable becomes longer, the baud rate will have to be reduced accordingly.

Example:
Later on, we shall see that PROFIBUS-DP operates with the RS485 interface, and the maximum possible baud rate is 12 Mbps. Table 3 shows how the baud rate has to be reduced for increasing lengths of cable:

Segment length [m]	100	200	400	1000	1200
Baud rate [kbps]	12000	1500	500	187.5	9.6

Table 3: Selecting the baud rate for various cable lengths

Note
As described in Chapter 2 *“Working with serial interfaces (RS232, 422, 485)”*, a shielded twisted-pair cable is used in most cases. The cable needs to be a twisted pair so that any voltages induced by an electromagnetic field in the two wires will cancel each other out.

3.3 Termination resistor

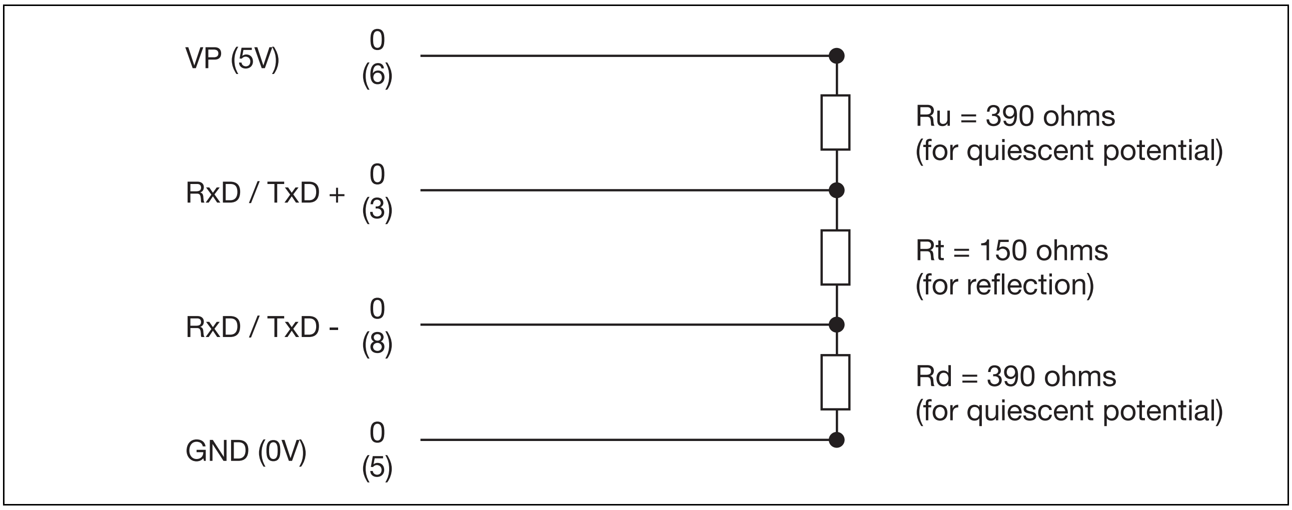


Fig. 25: Example of termination resistors used in an RS485 interface for PROFIBUS-DP

If the ends of the cable are open (at the first and last stations in the bus system) this will cause reflections of the signals. The higher the baud rate, the worse they will be. A termination resistor must be used in order to keep reflected signals as low as possible.

In addition, in PROFIBUS-DP, for example, the connection of the termination resistor to the ground potential provides a defined quiescent potential level.

Bus systems

This section describes those bus systems for which components are supplied by JUMO. Explanations are given for individual system fundamentals, and also advice on making connections. Please note that these explanations must not be considered as a replacement for the appropriate operating instructions.

To begin, we will give you an overview of the fieldbuses described in this chapter:

	MODbus	PROFIBUS-DP	ETHERNET	CANopen	HART®
Manufacturer	Gould-Modicon /	Siemens, Klöckner Möller, Bosch etc.		Bosch and Intel	Rosemount Inc.
User organization	Modbus-Ida Organization	PROFIBUS-Nutzerorganisation (PNO)	Iaona	Can in Automation (CiA)	HART Communication Foundation (HCF)
Access procedure	Master / slave	Master / slave	CSMA-CD	CSMA-CA	Master / slave
Medium	unrestricted	2-wire cable or optical fiber cable	2-wire cable	2-wire cable	2-wire cable
Stations / segment	247	32	16.777.216 (Class A network)	127	15
Data transmission rate	depending on interface, up to 187.5 kbps	9600 bps to 12 Mbps	100 Mbps	50 kbps to 1 Mbps	1200 bps
Bus expansion	depending on interface, up to 1200 m / segment	1200 m / segment	maximum distance between two points is 100 m	up to 5000 m	up to 2000 m
Other information	www.modbus.org	www.profibus.com	www.iaona-eu.com	www.can-cia.de	www.hartcomm.org

Table 4: Summary of the fieldbuses described here

4.1 MODbus

MODbus is a transmission protocol that was developed in 1979 by the American company Gould-Modicon. The primary application for MODbus is in the process visualization sector. It is a simple, reliable protocol, and is specified in the "Modicon Modbus Protocol Reference Guide". The great advantage of MODbus is that this protocol can be implemented by any programmer and no license fees are charged. This leads to commercial advantages in comparison with other bus systems.

In MODbus applications, users can themselves decide which interfaces should be implemented (RS422 / RS485, RS232, optical fiber etc.). In most cases, the RS485 / RS422 interface will probably be used. If RS422 or RS485 interfaces are used, then the stations must be arranged in a linear topology, whereby the first and last stations in the line must be fitted with termination resistors (see also Chapter 2 *"Working with serial interfaces (RS232, 422, 485)"*).

The interfaces for all stations must have the same settings for baud rate, parity check and the number of stop bits. Each station must have an address assigned, and there must only be one instance of each address in the system.

As far as the baud rate is concerned, JUMO uses the following standard rates:
1200, 2400, 4800, 9600 and 19200 bps.

MODbus operates on the master/slave principle. As already described, in this procedure a master (with MODbus, this is often a visualization software on a PC) undertakes a cyclic exchange of data with all the slaves.

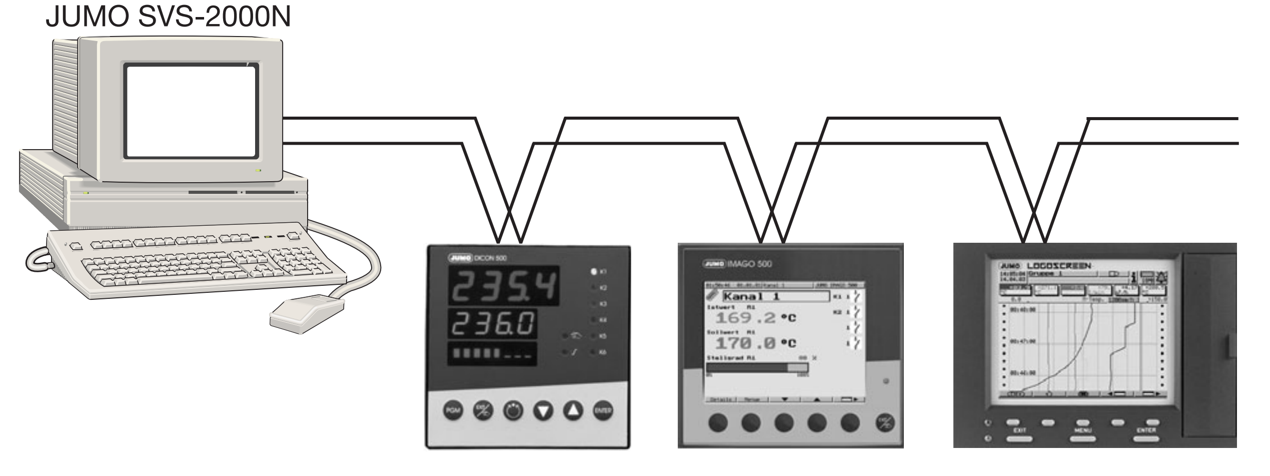


Fig. 26: Several field devices, RS485 interface and a PC with JUMO SVS-2000N

The master begins with the first slave (Fig. 26) and sends its request individually to this station. The slave answers the request (for instance, by transmitting the measurement from analog input 1). The master sends further requests in the same way, which are answered by the slave in the same way. Now the master sends out its instructions, e. g. “Set setpoint 1 to 50”. The station sets the setpoint to 50 and confirms this change for the master. The time taken until the data have been exchanged with all the slaves, and the data transmission can start again from the beginning, is called the cycle time. For MODbus, this may take several seconds, which is usually acceptable for process visualization systems.

4.1.1 Arrangement of the protocol

The requests and instructions have fixed definitions, and are arranged as follows:

Slave address	Function code	Data field	Checksum CRC16
1 byte	1 byte	n x bytes	2 bytes

Let us look at a specific example of the arrangement of an instruction:
The process value for the JUMO DICON 500 controller (this should have the address 1 assigned) should transmit the filtered process value. The request from the PC begins with the slave address. The address is transmitted as 1 byte, so the master will send “00000001”. Expressed as hexadecimal, the value is 0x01 (the 0x indicates that the following number is a hexadecimal number): 0x01 is also decimal 1, but 0x10 is decimal 16).

In the following function code, the controller is instructed as to whether a value should be written or read. The following important function codes are available:
0x03: read n words
0x06: write one word
0x10: write n words

We want to read a value that is formed by several words. For this reason, we choose the function code 0x03.

The previous protocol can be interpreted (as a hexadecimal number) as

01	03
message	read
for Slave 1	n words

The data field follows next, and this begins with the address of the first word.

For this, we need the interface description for the JUMO DICON 500:

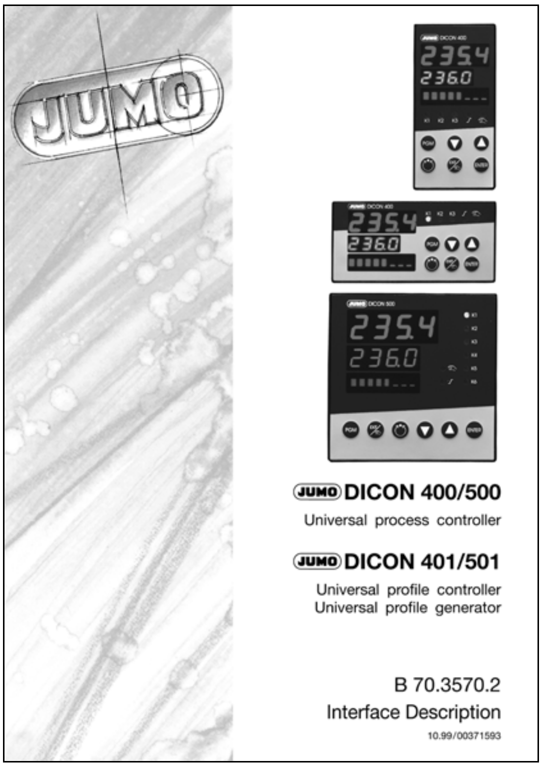
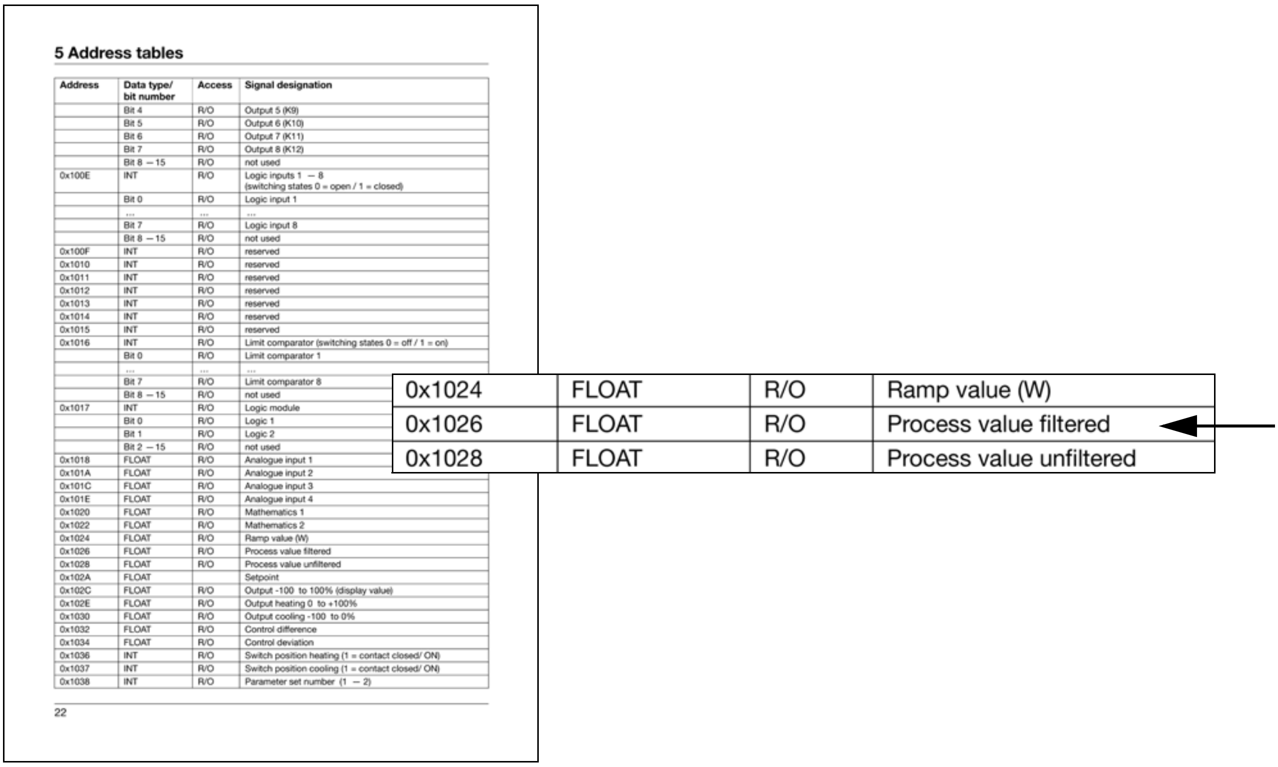


Fig. 27: Interface description for JUMO DICON 500

In the interface description we can see from which address onwards the filtered process value is stored:



5 Address tables

Address	Data type/ bit number	Access	Signal designation
	Bit 4	R/O	Output 5 (K9)
	Bit 5	R/O	Output 6 (K10)
	Bit 6	R/O	Output 7 (K11)
	Bit 7	R/O	Output 8 (K12)
	Bit 8 – 15	R/O	not used
0x100E	INT	R/O	Logic inputs 1 – 8 (switching states 0 = open / 1 = closed)
	Bit 0	R/O	Logic input 1
	...	...	...
	Bit 7	R/O	Logic input 8
	Bit 8 – 15	R/O	not used
0x100F	INT	R/O	reserved
0x1010	INT	R/O	reserved
0x1011	INT	R/O	reserved
0x1012	INT	R/O	reserved
0x1013	INT	R/O	reserved
0x1014	INT	R/O	reserved
0x1015	INT	R/O	reserved
0x1016	INT	R/O	Limit comparator (switching states 0 = off / 1 = on)
	Bit 0	R/O	Limit comparator 1
	...	...	...
	Bit 7	R/O	Limit comparator 8
	Bit 8 – 15	R/O	not used
0x1017	INT	R/O	Logic module
	Bit 0	R/O	Logic 1
	Bit 1	R/O	Logic 2
	Bit 2 – 15	R/O	not used
0x1018	FLOAT	R/O	Analogue input 1
0x101A	FLOAT	R/O	Analogue input 2
0x101C	FLOAT	R/O	Analogue input 3
0x101E	FLOAT	R/O	Analogue input 4
0x1020	FLOAT	R/O	Mathematics 1
0x1022	FLOAT	R/O	Mathematics 2
0x1024	FLOAT	R/O	Ramp value (W)
0x1026	FLOAT	R/O	Process value filtered
0x1028	FLOAT	R/O	Process value unfiltered
0x102A	FLOAT		Setpoint
0x102C	FLOAT	R/O	Output -100 to 100% (display value)
0x102E	FLOAT	R/O	Output heating 0 to +100%
0x1030	FLOAT	R/O	Output cooling -100 to 0%
0x1032	FLOAT	R/O	Control difference
0x1034	FLOAT	R/O	Control deviation
0x1036	INT	R/O	Switch position heating (1 = contact closed/ ON)
0x1037	INT	R/O	Switch position cooling (1 = contact closed/ ON)
0x1038	INT	R/O	Parameter set number (1 – 2)

22

0x1024	FLOAT	R/O	Ramp value (W)
0x1026	FLOAT	R/O	Process value filtered
0x1028	FLOAT	R/O	Process value unfiltered

Fig. 28: Address table for the JUMO DICON 500

As we can see, the filtered process value can be accessed from address 0x1026 onwards. So our data field begins with the value 0x1026. There now follow 2 bytes of information in the data field, which specify how many words have to be read from address 0x1026 onwards. The process value is a float value that consists of two words, so the data field looks like this:

1026 read from address 0x1026	0002 the given address: 2 words

The request can now be interpreted as:

01 message for slave 1	03 read n words	1026 read from address 0x1026	0002 read from the given address 2 words

The master actually transmits the following bits across the bus:

00000001	00000011	0001000000100110	0000000000000010

The CRC check is formed from the request, and transmitted directly after the request. The receiver forms the CRC check in the same way. If it does not come to the same result, the information will be discarded. The way in which the CRC check is formed is shown at the end of this chapter.

As previously mentioned, the MODbus is an old and well-proven protocol. It is still very highly favored by programmers. But of course, it is a disadvantage that programming effort is required. However, one can almost always make use of existing drivers.

JUMO develops and markets a process visualization software that operates on the basis of MODbus. This is JUMO SVS-2000N, and is interesting for companies that use a number of JUMO instruments, and want to be able to show the process variables from these devices on a PC screen (in a control room, for example). The various temperatures, pressures, pH values etc. can not only be displayed, but also saved on the hard disk of the PC.

Users are not confronted with the protocol details as described above. They only have to define the PC interface (COM, baud rate, parity, and the number of stop bits), take the devices from a catalog, and decide which parameters are to be transmitted between the devices and the JUMO SVS-2000N. After incorporating the devices and functions in what are known as “group diagrams”, the variables will be displayed on the PC screen.

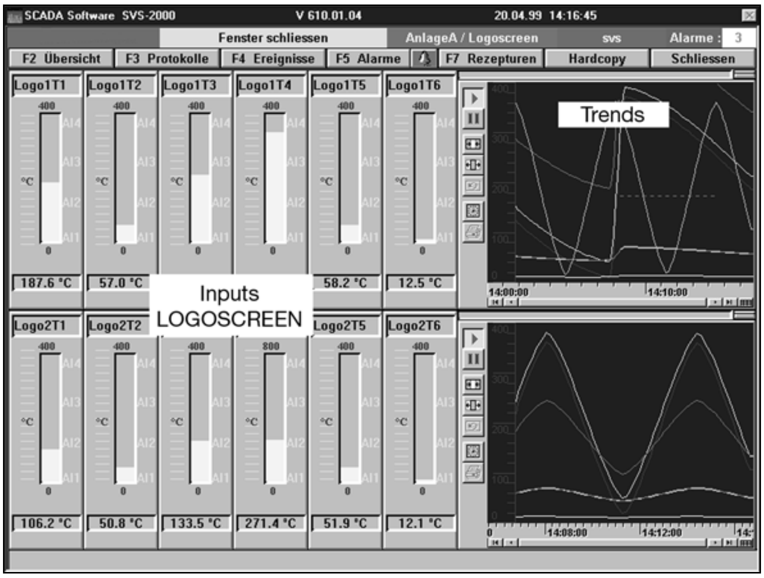


Fig. 29: Group diagram for JUMO SVS-2000N

Fig. 29 shows a group diagram. Here the signals are displayed from two paperless recorders (inputs 1 – 6 in each case). In the example illustrated, the measured values are recorded and displayed in two trend diagrams.

Furthermore, the JUMO SVS-2000N is network-capable, which means that the field devices are, possibly, just connected to a PC in a control room. The JUMO SVS-2000N is installed on other PCs, which are linked to the control room PC through the network. The process data are transmitted across the network to these PCs, where they are also available.

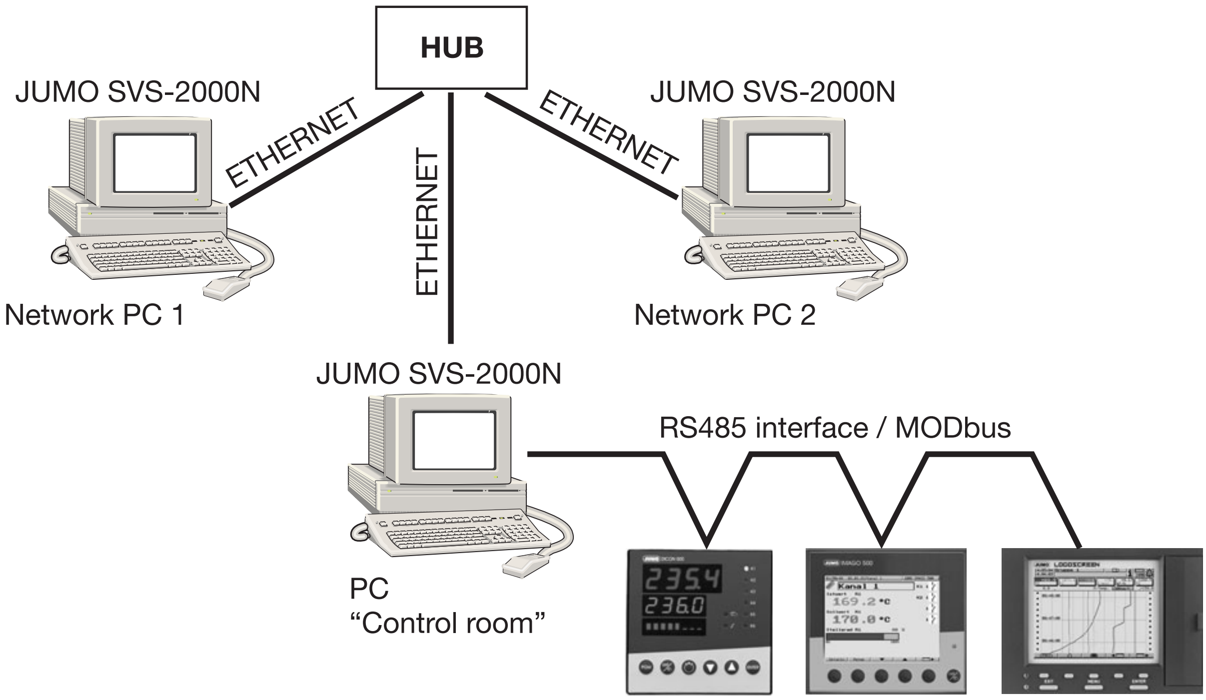


Fig. 30: JUMO SVS-2000N in several PCs within a network

4.1.2 CRC check

The following illustrates how a CRC check is generated for a MODbus request.

Let us take another look at the request that the master has to put out on the bus to request the process value from a DICON 500 with the address 1:

01 03 10260002

The sequence starts with the first byte (01). This is, expressed as a binary number: 00000001_2.

	filled up with 0			
For the 8 bits ...	0000	0000	0000	0001
and ...	1111	1111	1111	1111
the result of a bit-by-bit exclusive-or operation is created:	1111	1111	**1**111	1110

Note
If two bits are subjected to an exclusive-or operation, the result is always 1 if the bits are unequal. Only the first exclusive-or operation is made with 1111 1111 1111 1111 (0xFFFF). As we will now see, the operation is immediately performed with other values.

With the result (1111 1111 **1**111 1110), the value of the rightmost bit is 0. All bits are shifted to the right, until the first 1 falls out. The bits that become vacant on the left are filled up with 0.

This produces: 0011 1111 11**1**1 1111 ~~1~~

For the 8 bits ...	0011	1111	11**1**1	1111
and ...	1010	0000	0000	0001
the result of a bit-by-bit exclusive-or operation is created:	1001	1111	11**1**1	1110

The bits are shifted twice to the right (so that the 1 falls out). The bits that become vacant on the left are filled up with 0. The result is:

	0010	0111	1111	**1**111
Another exclusive-or is performed with	1010	0000	0000	0001
Result:	1000	0111	1111	**1**110

Again, the row is shifted until the first 1 falls off at the right:

	0010	0001	1111	11**1**1
Exclusive-or with	1010	0000	0000	0001
Result:	1000	0001	1111	11**1**0

Again, the row is shifted until the first 1 falls off at the right.
Note that, in all, there are only 8 shifts:

	0010	0000	0111	1111	~~1~~
Another exclusive-or is performed with	1010	0000	0000	0001	
Result:	1000	0000	0111	1110	

Up to now, the CRC check was:	1000	0000	0111	1110
	(or in hexadecimal: 0x807E)			

Now the 2nd byte of the request (0x03 = 00000011) is involved:

For the 2nd byte ...	0000	0000	0000	0011
and the CRC check up to now ...	1000	0000	0111	1110
the result of a bit-by-bit exclusive-or operation is created:	1000	0000	**0**111	1101

Again, the row is shifted until the first 1 falls off at the right:

	0100	0000	0**0**11	1110

The further exclusive-or operations are again performed with

	1010	0000	0000	0001

as before. The row is repeatedly shifted right until a 1 falls off, and then the exclusive-or operation is carried out. After the 8th shift, the exclusive-or operation is only carried out (for the last time) if a 1 falls out, otherwise the result is accepted as it stands.

The result, after involving the 2nd byte, would now be:

0010 0001 0100 0000 (or in hexadecimal: 0x2140)

After involving the 3rd byte (10), the CRC-16 result is:

0011 1100 0010 0001 (or in hexadecimal: 0x3C21)

After involving the 4th byte (26), the CRC-16 result is:

1100 0010 0111 1101 (or in hexadecimal: 0xC27D)

After involving the 4th byte (00), the CRC-16 result is:

0010 0001 0000 0010 (or in hexadecimal: 0x2102)

After involving the 4th byte (02), the CRC-16 result is:

0000 0000 0010 0001 (or in hexadecimal: 0x0021)

The complete request from the master is now as follows:

01	**03**	**10260002**	**2100**
			CRC-16

The low byte of the checksum is transmitted first. For this reason, 0x2100 (and not 0x0021) will be transmitted.

The following diagram shows the calculation method for the CRC check:

CRC = 0xFFFF

CRC = CRC XOR ByteOfMessage

For (1 to 8)

CRC = SHR(CRC)

if (right shifted out Flag = 1)

then CRC = CRC XOR 0xA001	else

while (not all ByteOfMessage processed);

Fig. 31: Calculation method for the CRC check

To conclude, here is an example of a program for calculating the checksum:

PASCAL / DELPHI

```
// Checksum CRC16
function CRC16 (instruction:String): String;
VAR CRC:   word;                    //CRC value
    n,m,i: Integer;
    manz:  Integer;                 // number of bytes
    buff:  array [1..256] of byte;  // message bytes
begin
  manz := ROUND (Length(instruction) / 2);

// conversion of the input string into binary values
  i:=1;m:=1;
  for n:=1 to manz do begin
    buff[m] := HEXtoINT (COPY(instruction,i,2)); m:=m+1; i:=i+2;
  end;

// calculation CRC16
  CRC := $FFFF;
  For m:=1 to manz do begin
    CRC := (CRC xor buff[m]);

    For n:=1 to 8 do begin
      if ((CRC and $0001) = 1) then begin
        CRC := ((CRC shr 1) xor $A001);
      end
      else begin
        CRC := (CRC shr 1);
      end;
      ShowMessage(IntToBin(CRC));
    end;
  end;

// conversion CRC16 into reversed string
  result := COPY (IntToHex(CRC,4),3,2) + COPY (IntToHex(CRC,4),1,2);
end;
```

4.2 PROFIBUS-DP

PROFIBUS (**PRO**cess **FI**eld **BUS**) is an international, open fieldbus standard that has been standardized in the fieldbus standard EN 50 170.

PROFIBUS technology was developed through the cooperation of a number of companies – under the leadership of Siemens, Klöckner Möller and Bosch – and is maintained by the PROFIBUS User Organization (German abbreviation: PNO).

The PROFIBUS family consists of three versions:

- **PROFIBUS-DP (Decentralized Periphery)**
  was conceived as an application for decentralized peripheral areas, where short system response times are important. It is almost always used to link the decentralized automation devices (such as controllers and paperless recorders) to a PLC via a serial interface.
- **PROFIBUS-PA (Process Automation)**
  was specially devised for process engineering, and permits the connection of sensors and actuators in **Ex** areas. PROFIBUS-PA enables communication and energy supply for devices in 2-wire technology, according to the international IEC 1158-2 standard.

PROFIBUS-DP and PA are designed for the fast transmission of small quantities of data.

- **PROFIBUS-FMS (Fieldbus Message Specification)**
  is used for communication at higher levels (e. g. between two programmable logic controllers). PROFIBUS-FMS is no longer very significant these days, since ETHERNET or PROFInet is used for this type of communication.

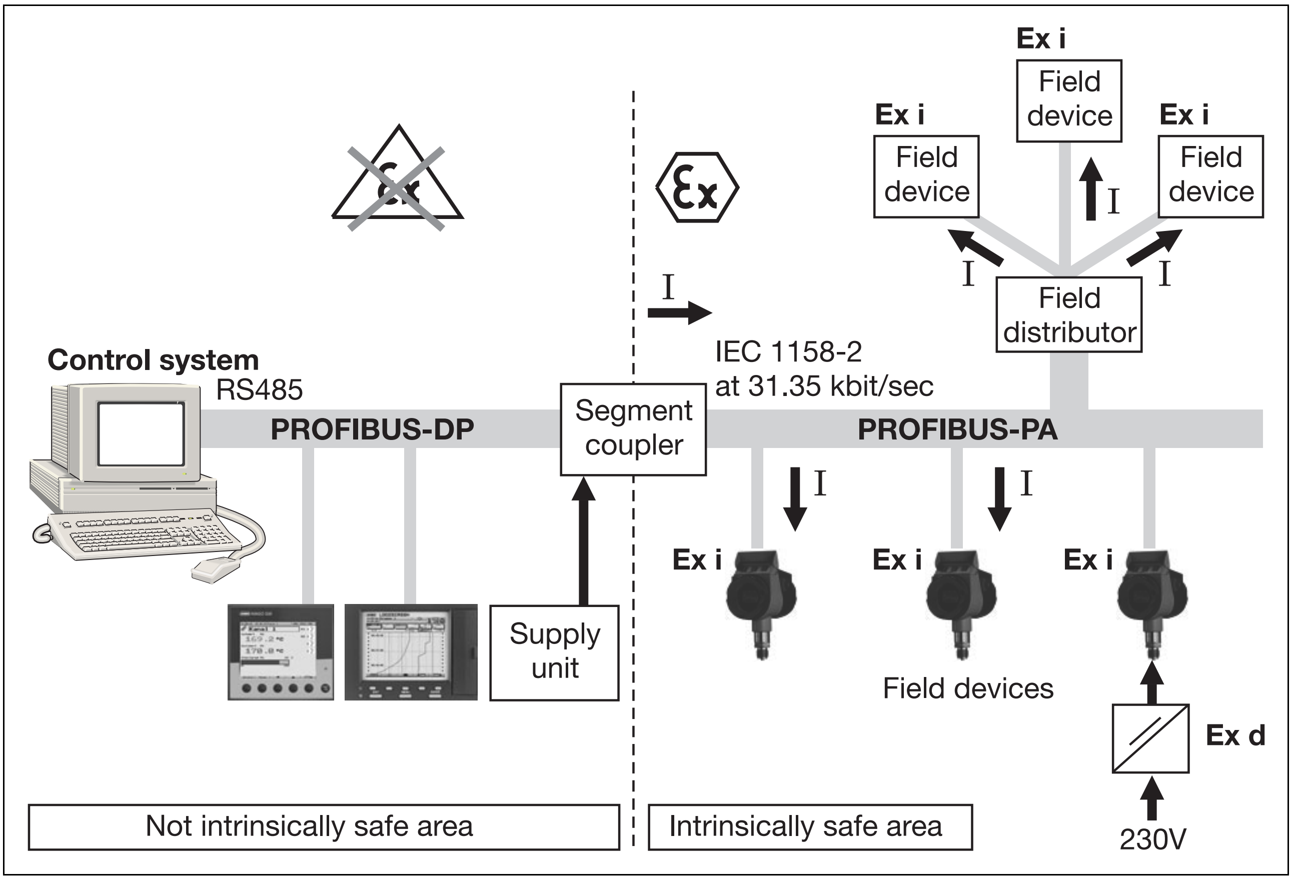


Fig. 32: PROFIBUS-DP and PROFIBUS-PA

4 Bus systems

As far as the PLC is concerned, the bus stations are always connected through PROFIBUS-DP. For this reason, we will begin with this form of PROFIBUS, and would like to mention here, that JUMO only supplies devices with PROFIBUS-DP. A maximum of 32 stations are arranged in a linear topology. If a larger number of stations need to be connected to a single interface, repeaters are available for this purpose.

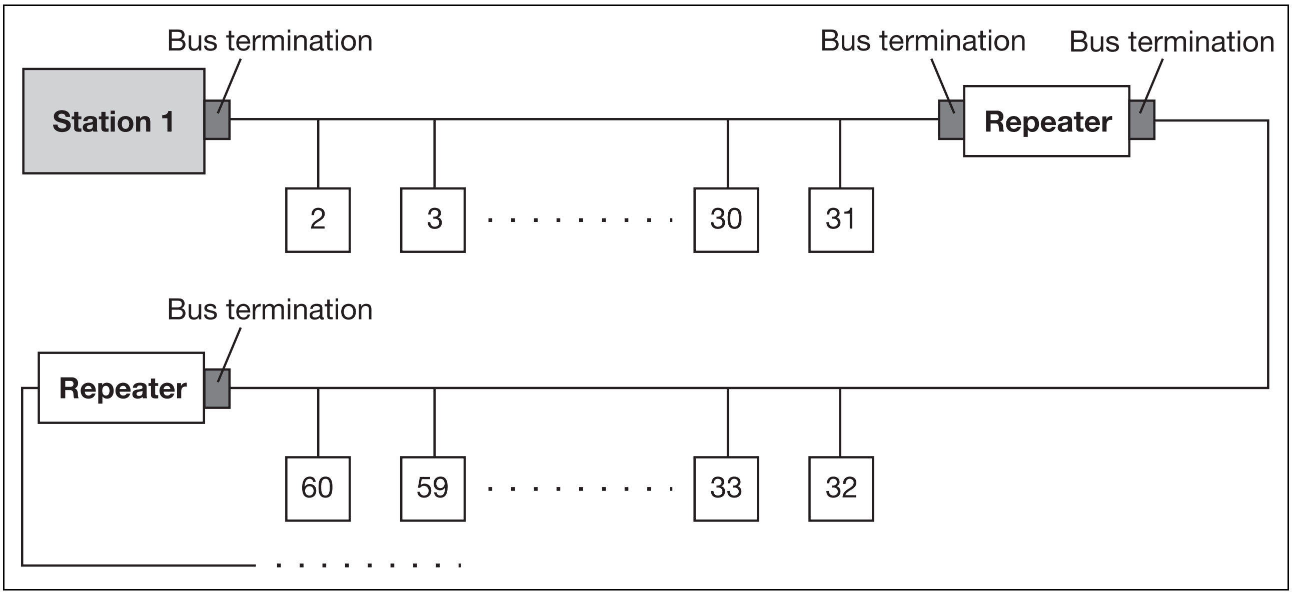


Fig. 33: Linear topology and the use of repeaters

Segments are the sections between repeaters. These are always limited to a maximum of 32 stations (including the repeaters). Overall, up to 127 stations can be connected via the interface. In addition to accessing a larger number of stations, repeaters are also used in order to increase the maximum achievable range of the system. For instance, if the PROFIBUS-DP system is driven at the typical transmission rate of 1.5 Mbits/sec, then the specification permits a range of 200 m. The insertion of a repeater creates two segments. Each segment can have a length of 200 m, so the total length of the network (first to last station) can now be up to 400 m.

The RS485 interface is the one most frequently used for PROFIBUS-DP. The transmission rate can be selected within the range from 9600 bits/sec to 12 Mbits/sec. The maximum achievable length of the network lies in the range from 1200 m (at 9600 bits/sec) to 100 m (at 12 Mbits/sec).

As a rule, the cable used is that defined by EN 50 170 Part 8-2 as Type A cable.
This can, for instance, be ordered directly from Siemens. The SUB-D connector that is used can also be supplied by Siemens.

The assembly of the cabling is very simple: The two cores of the cable (Fig. 34) are colored red and green respectively. The terminals in the connector are marked with the same colors. The incoming and outgoing leads are connected to the correspondingly colored terminals (the terminals for the incoming leads are usually labeled A1 and B1, with A2 and B2 for the outgoing leads).

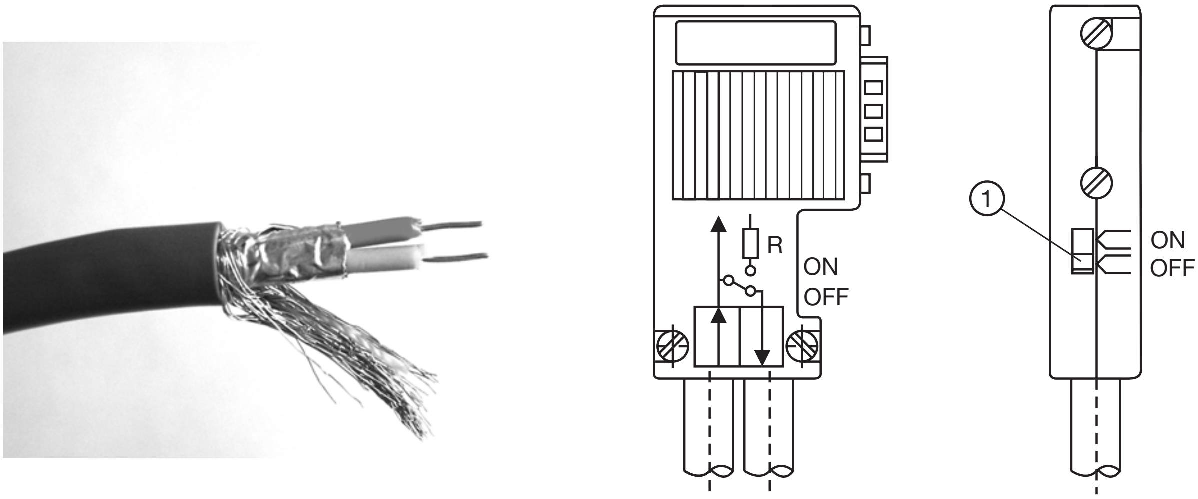


Fig. 34: Type A cable and a schematic representation of the PROFIBUS connector

Fig. 35 shows the pin assignments for the PROFIBUS connector.

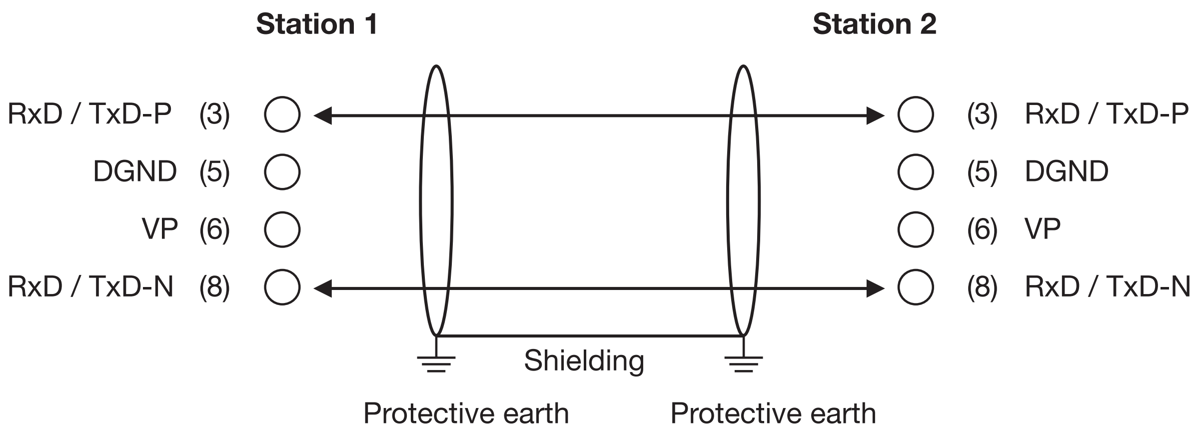


Fig. 35: Pin assignments for the PROFIBUS connector

If an RS485 interface is used, the first and last stations on the line must be fitted with termination resistors. Termination resistors are already incorporated in the connectors (Fig. 34), they only have to be activated by the sliding switch.

PROFIBUS-DP operates mainly on the master-slave principle. The master (usually a PLC) exchanges data on a cyclical basis with the slaves (controllers, paperless recorders etc.), and the bus cycle time is in the millisecond range. However, it is also possible to use several master stations with PROFIBUS-DP. In this case, only one master at a time can communicate with the slaves. The access authorization (the token) is passed from one master to another, and only the master station that possesses the token at the moment is permitted to transmit over the bus.

The interface settings for the slaves in PROFIBUS-DP are limited to the device address. If the slave it fitted with automatic baud rate (transmission rate) detection, then not even this has to be set.

In order to be able to link a slave to a PLC, the manufacturer of the PROFIBUS-DP device must provide a GSD file (**G**eräte-**S**tamm-**D**aten = basic device data). The GSD file is subsequently read in by the configuration tool of the PLC, and specifies, among others, which data are to be transmitted from the PLC to the device, and which data are to be transmitted in the reverse direction.

As an example, let us look at the steps that are necessary to read the process value from a JUMO IMAGO 500 controller into a Siemens S7:

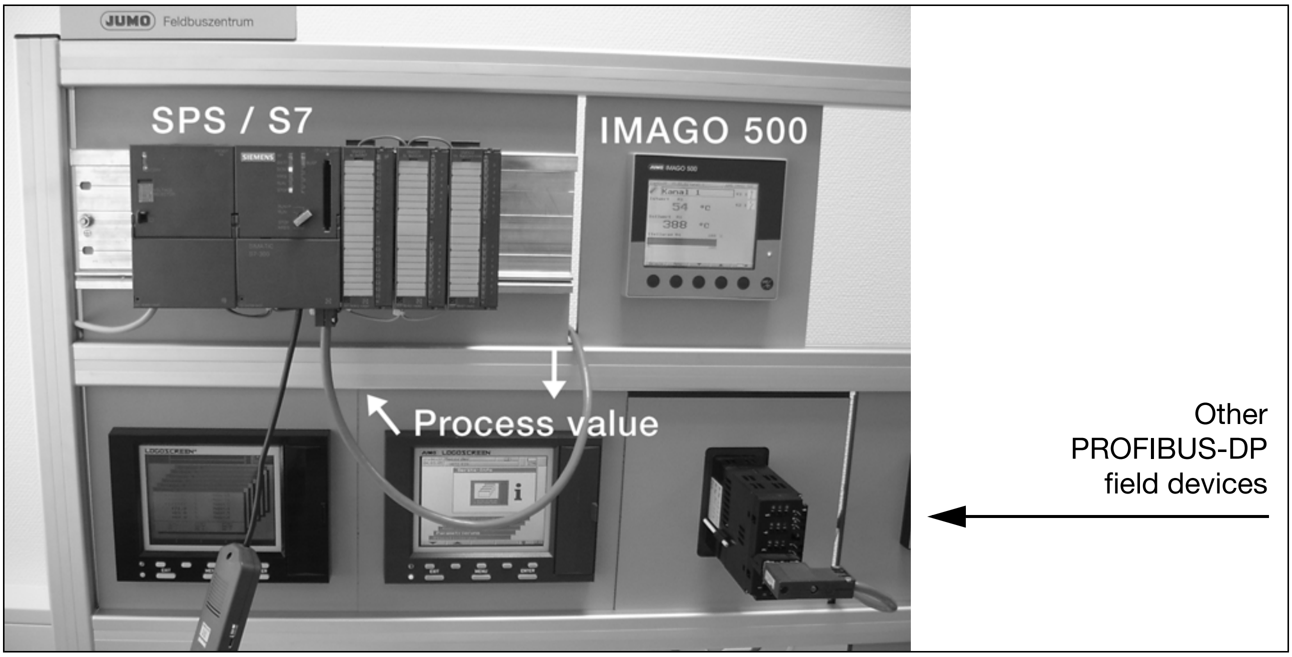


Fig. 36: JUMO IMAGO 500 and PLC

If a PROFIBUS-DP device is acquired from JUMO, then the “GSD Generator” will be supplied with it. This is a program with which the user can create a GSD file for the device and define which data are to be transmitted via PROFIBUS-DP. Using a fixed GSD file would mean that certain defined data would be transmitted – regardless of whether they are actually needed or not. This would place an unnecessary load on the bus system.

In order to create a GSD file for the example mentioned above, the GSD Generator is started up and informed that a GSD file is to be created for a JUMO IMAGO 500.

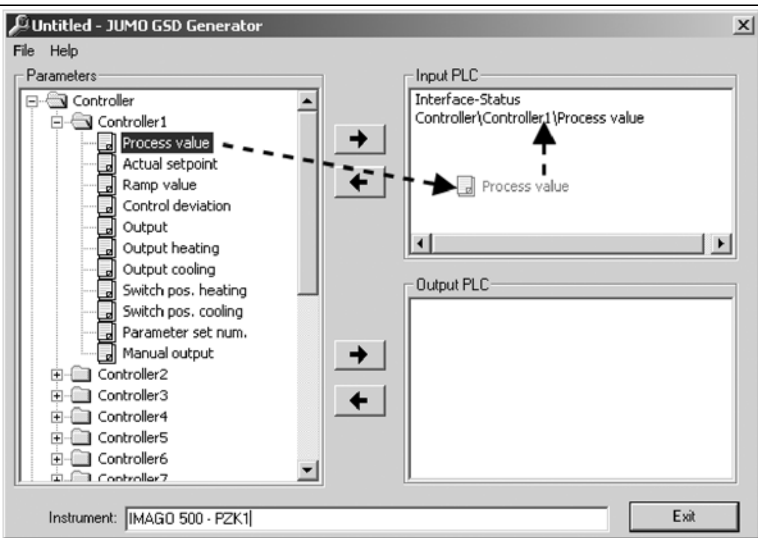


Fig. 37: Creating a GSD file

The “Parameters” window (Fig. 37) lists all the parameters that can be transmitted in this simple fashion via PROFIBUS-DP. Now you can, for example, place the process value from Controller 1 in the “PLC input” window (up to 8 controller channels are possible with the JUMO IMAGO 500). This means that, subsequently, the process value will be read into the input memory of the PLC on a cyclic basis. Other parameters can be set up in the “PLC input” and “PLC output” windows in the same way.

The name “IMAGO 500 - PZK1” that is given for the device is editable, and appears later in the device catalog of SIMATIC Manager.

Finally, save the GSD file.

The address must be defined on the JUMO IMAGO 500. We will set this to “1”. No other settings are required for the interface.

Now we will start working with SIMATIC Manager. This program is used for the hardware and software configuration of the PLC. We will begin with the hardware. Let us assume that the hardware for the PLC has already been configured (CPU, power supply, input/output modules etc.). Furthermore, PROFIBUS-DP has been activated and the transmission rate has been set to 1.5 Mbits/sec.

Let us look at the PLC in the hardware configuration:

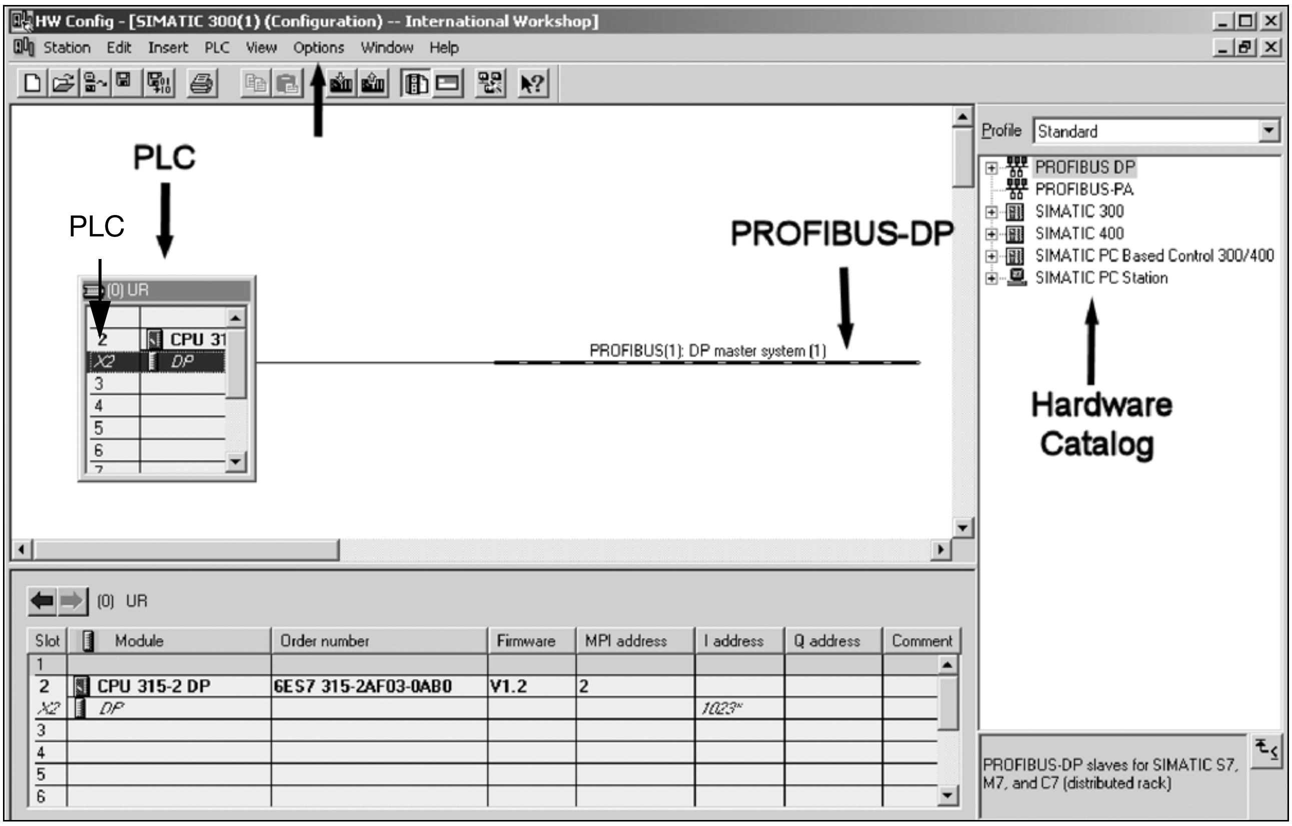


Fig. 38: PLC with PROFIBUS-DP in the hardware configuration

Fig. 38 shows the PLC and the PROFIBUS-DP bus line. In order to link the controller to this bus line, the JUMO IMAGO 500 must be included in the hardware catalog. This is done through the “Extras” menu.

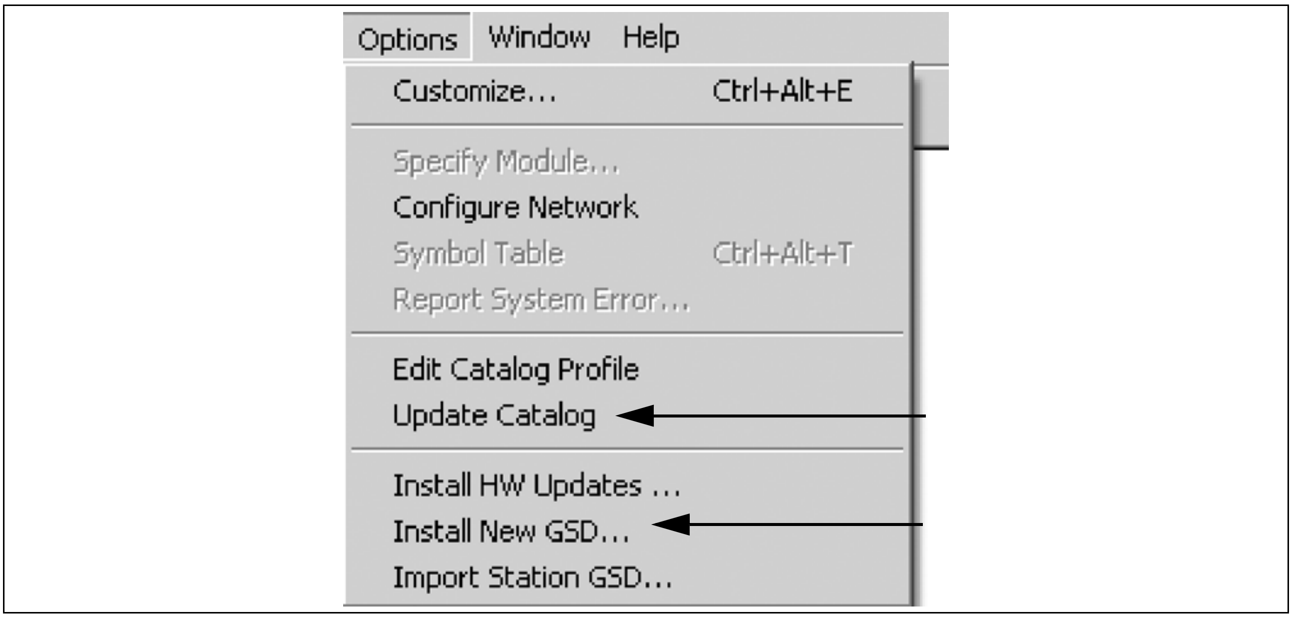


Fig. 39: The "Extras" menu in the hardware configuration

The GSD file that was previously generated is now called up in the sub-menu "Install New GSD". After selecting "Update Catalog", the JUMO IMAGO 500 is set up in the hardware catalog under the designation given above.

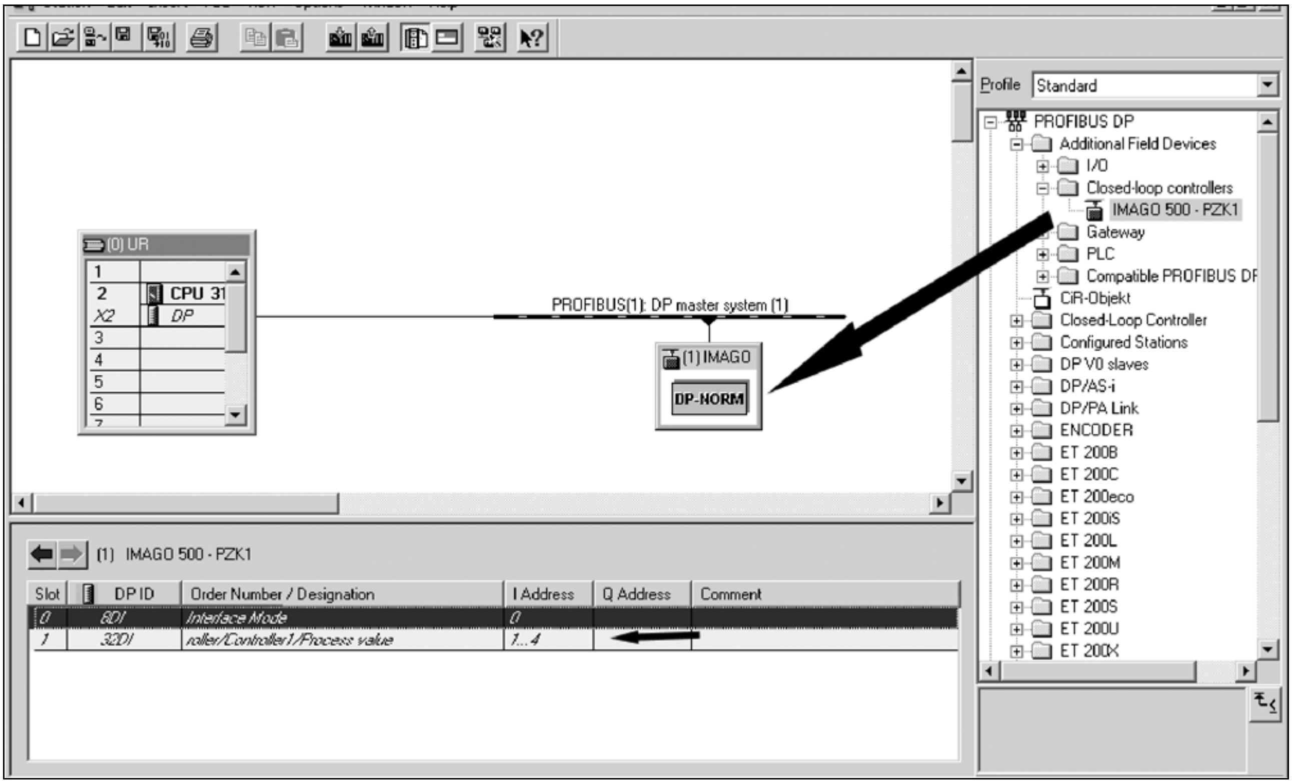


Fig. 40: Linking JUMO IMAGO 500 to PROFIBUS-DP

The controller symbol can now be dragged from the hardware catalog and dropped onto the PROFIBUS-DP line. At this point, the address for the controller must also be defined in SIMATIC Manager (as "1" in our example). If this configuration is transmitted to the PLC, the PLC will now cyclically poll the process value of the controller, and save this value in the input memory, from Byte "1" on (Fig. 40). From this moment on, the process value can be used in the PLC program, through the address in the memory area (saved in the input memory in the range Byte 1 — 4).

As already mentioned, the RS485 interface is the standard medium for PROFIBUS-DP. Optical fiber transmission can also be used, if the environment has a high level of interference or greater distances need to be covered. Special bus connectors are commercially available, with an integrated conversion of RS485 to optical fiber signals and vice versa. This makes it simple to convert from one type of transmission to the other within the system.

In conclusion, some information on PROFIBUS-PA

As can be seen in Fig. 32, the connection between a PLC and PROFIBUS is always made through PROFIBUS-DP (mostly using the RS485 interface). Segment couplers are used to convert the signal level of the RS485 interface into a signal that conforms to IEC 1158-2. This interface meets the requirements of the chemical and petrochemical industries, and makes it possible to use the field devices in the **Ex** area. Networks can be assembled in linear, tree and star topologies. The transmission rate for a 2-wire connection (Fig. 32) is fixed at 31.25 kbits/sec.

With this technology, there is only one supply source (the supply device) in each segment. The supply device is frequently included in the segment coupler, which implements the link between bus segments and the RS485 transmission. The maximum number of stations in one segment is 32, but in practice this number may be further limited by the bus supply and the explosion protection type that is selected.

It is interesting to note that, with PROFIBUS-PA, no energy is fed into the bus when a station is transmitting, since the stations operate as current sinks. Each station draws a constant base current (e. g. 10 mA), and the transmitting device generates the communication signal by a ±9 mA modulation of this current.

4.3 ETHERNET

In the newer implementations of ETHERNET, all stations are connected together via twisted-pair cables and hubs.

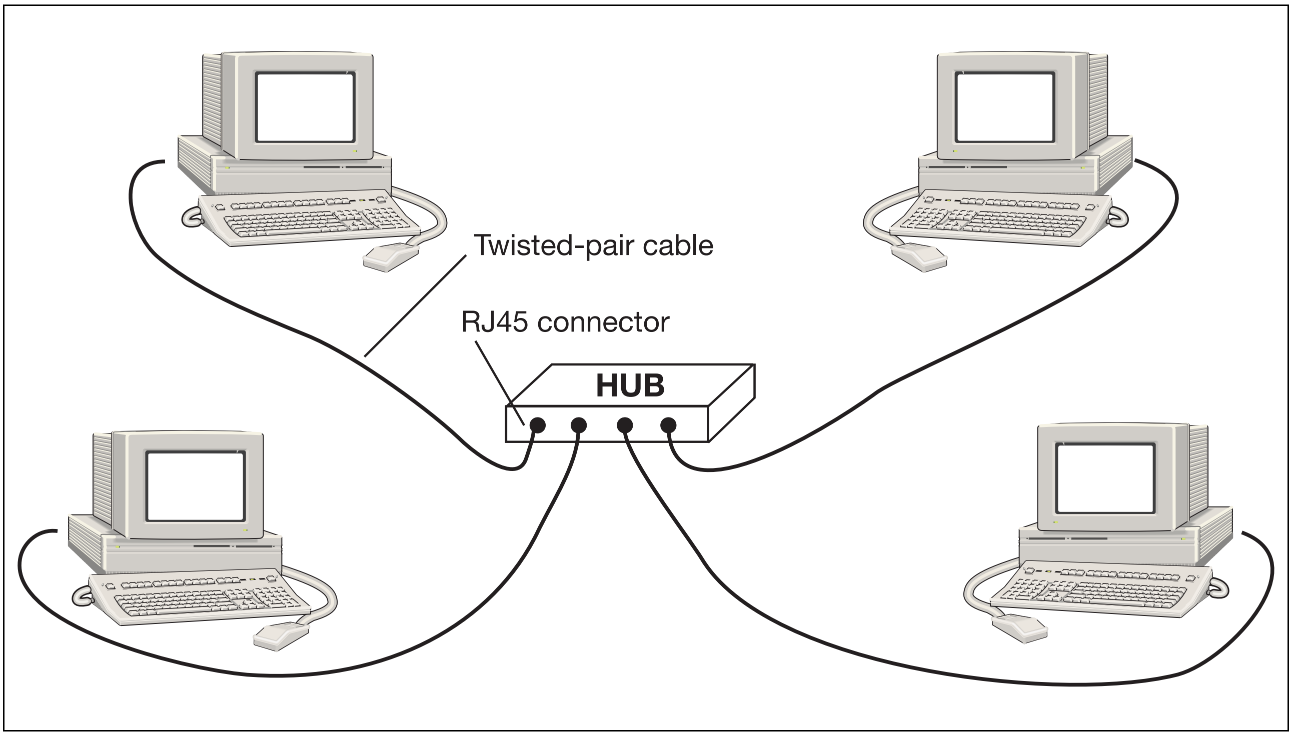


Fig. 41: Connecting stations to ETHERNET via a HUB

A hub (sometimes referred to as a star coupler) provides the facility of connecting several stations together in a star topology. The hub forwards data packets that are received by a port to all other ports. The information is thus made accessible to all the stations. If a station is transmitting data over the network, measures must be taken to ensure that the data are received by the correct computer. The target computer requires a unique address. To achieve this, a world-wide unique address is "burnt" into the card. This is known as the MAC (**M**edia **A**ccess **C**ontrol) address. It has a standardized layout and consists of 6 bytes.

Manufacturer (e. g. JUMO)			Network card		
00	0C	D8	5A	CD	D0

The first three bytes represent the manufacturer. The next three bytes designate the individual card.

The ID numbers for the manufacturer section are allocated centrally by the IEEE (**I**nstitute of **E**lectrical and **E**lectronic **E**ngineers). The IDs for the individual cards are implemented by the manufacturers themselves.

The MAC address is normally printed on a label that can be found on the network card itself. On JUMO paperless recorders with an ETHERNET interface, the address can, for example, be read out with the device information.

ETHERNET provides the physical basis for TCP/IP.

4.3.1 TCP/IP

As early as the 1960s, the American military had already placed an order for the development of a protocol that would enable a standardized exchange of information between any number of various networks, independently of the hardware and software that was used. This requirement led, in 1974, to the TCP/IP protocol.

Although TCP and IP are usually mentioned as if they were one item, they are in fact two stacked protocols. The **I**nternet **P**rotocol (IP) handles the correct addressing and delivery of the data packets, while the **T**ransport **C**ontrol **P**rotocol (TCP) that is stacked on it, is responsible for the transport and safety of the data.

4.3.2 IP – Internet Protocol

The Internet Protocol makes it possible to connect any number of individual devices together to form a complete network. It enables the exchange of data between any two network stations, and these may be located in any subsidiary network.

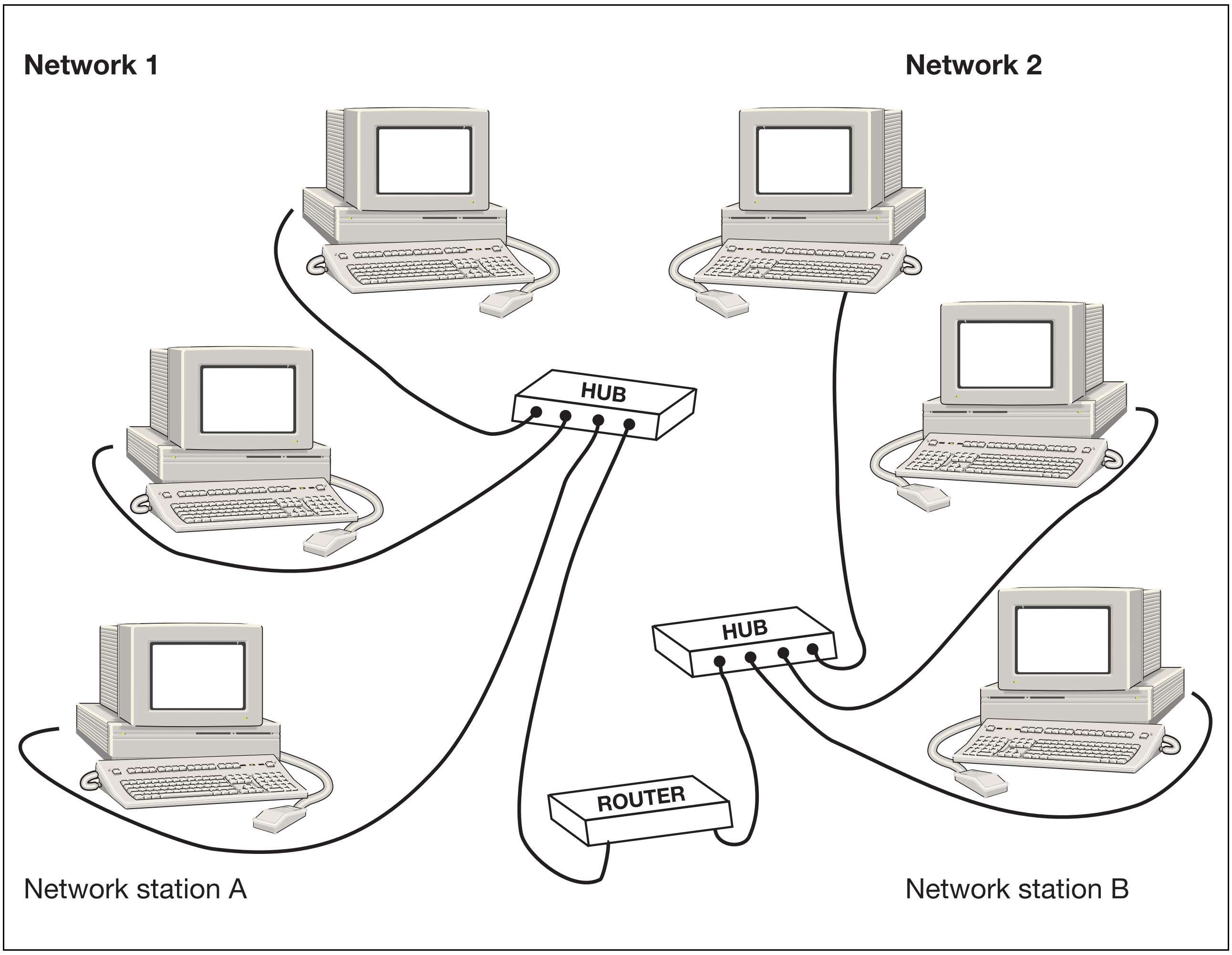


Fig. 42: Two networks connected via a router

The addressing of terminal devices with an ETHERNET interface (mostly PCs) is not, however, made directly through the MAC address, but through the IP address. As we shall shortly see, the use of IP addresses makes it possible, for example, to have logical assignments to separate networks.

The physical network addresses that were mentioned above (the MAC addresses) are assigned to the logical IP addresses.

IP addresses are binary numbers with a length of 32 bits (4 bytes) that are given in "dotted-decimal format" for better legibility: in other words, four decimal numbers (octets) separated by points, e.g. 193.96.1.200.

An octet can be used either for addressing a network or for addressing the participating station (host). We refer to various network classes, depending on how many octets are used for network or station addressing.

In practice, 3 network classes are of major significance: classes A, B and C.

A class A network uses the first octet for addressing the network.
All stations (hosts) in the network have the same network address (the first octet).

Example of a class A network
(in this example, all stations have the network address 1 = network ID):

1.	0.0.1
Network ID	Host ID

If, for example, all hosts are assigned to network 1, then IP 1.0.0.1 could be assigned to the first station, and IP 1.255.255.254 to the last station.

The definition of the network class is made through the subnet mask, which also consists of 4 octets. If the first octet is set to 255, and the remaining three octets to 0 (i.e. 255.0.0.0), this means that the first octet of the IP is used for network addressing, and the remaining three octets for host addressing. With this type of subnet mask (255.0.0.0) we have established a class A network.

For a class B network, the subnet mask must look like this: 255.255.0.0.
In a class B network, the first two octets are used for network addressing, and the remaining two octets for host addressing.

If the stations are located in the first possible class B network, then the first station has the IP

128.1.	0.1
Network ID	Host ID

The last station in this class B network would have the address 128.1.255.254.

In a similar way, a subnet mask 255.255.255.0 defines a class C network.

Addressing rules
Basically, an octet (one byte) can have any value from 0 — 255. However, special addressing rules mean that it is not possible to use the entire range of numbers for IP addresses. These addressing rules will not be discussed in detail here, but the following table illustrates the address options:

Class A					
		Network ID	**Host ID**		
1st network / 1st host	binary	**0**0000001	00000000	00000000	00000001
	decimal	1	0	0	1
1st network / last host	binary	**0**0000001	11111111	11111111	11111110
	decimal	1	255	255	254
last network / 1st host	binary	**0**11111110	00000000	00000000	00000001
	decimal	126	0	0	1
last network / last host	binary	**0**11111110	11111111	11111111	11111110
	decimal	126	255	255	254
Class B					
		Network ID		**Host ID**	
1st network / 1st host	binary	**10**000000	00000001	00000000	00000001
	decimal	128	1	0	1
1st network / last host	binary	**10**000000	00000001	11111111	11111110
	decimal	128	1	255	254
last network / 1st host	binary	**10**111111	11111111	00000000	00000001
	decimal	191	255	0	1
last network / last host	binary	**10**111111	11111111	11111111	11111110
	decimal	191	255	255	254
Class C					
		Network ID			**Host ID**
1st network / 1st host	binary	**110**00000	00000000	00000001	00000001
	decimal	192	0	1	1
1st network / last host	binary	**110**00000	00000000	00000001	11111110
	decimal	192	0	1	254
last network / 1st host	binary	**110**11111	11111111	11111111	00000001
	decimal	191	255	255	1
last network / last host	binary	**110**11111	11111111	11111111	11111110
	decimal	223	255	255	254

Table 5: Address options

Note
Different networks can be linked through bridges or routers.

4.3.3 IP address assignment

There are various options for assigning the IP addresses within a network:

Assignment from an address pool
The DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) provides the network administrator with a tool with which the network settings for the individual terminal devices can be automatically and

centrally configured in a standard manner. At least one DHCP is required in the network to be able to make use of DHCP: a range of IP addresses is defined on the server, and a calling station (usually a PC) will have an IP address assigned that is not otherwise being used at the time. As a rule, this assignment is for a limited time, whereby the time of use (Lease Time) can be defined by the network administrator. This automatic assignment of IP addresses is standard within networks.

Automatic assignment of IP addresses is advantageous if stations can alter their location within the networks. In this case, stations will always receive an IP address that is valid for their location.

Exclusion of specific IP addresses from the DHCP configuration

For terminal devices that are not DHCP-capable, the network administrator has the option of excluding individual IP addresses (or even complete address ranges) from assignment by DHCP. The addresses that have been excluded from the automatic assignment are then available for the corresponding devices

In this case, configuration must be carried out either on the terminal device or through the use of tools that are included in the delivery.

A fixed IP address must be present for JUMO devices with an ETHERNET interface and the COM servers (described below). This means, that the network administrator must exclude the corresponding IP from DHCP assignment, as already described. The IP is reserved for the corresponding field device, and has a fixed setting on this device.

4.3.4 Linking a station to ETHERNET / assembling a class A network

In order to connect a field device to a company network, it is basically only necessary to define the IP address and the subnet mask.

But we want to start right at the beginning.
We want to build up a small network, consisting of two PCs and a JUMO LOGOSCREEN es paperless recorder. The paperless recorder can carry out data recording for up to 36 process variables (temperatures, pressures etc.) and make the measurements available via its ETHERNET interface.

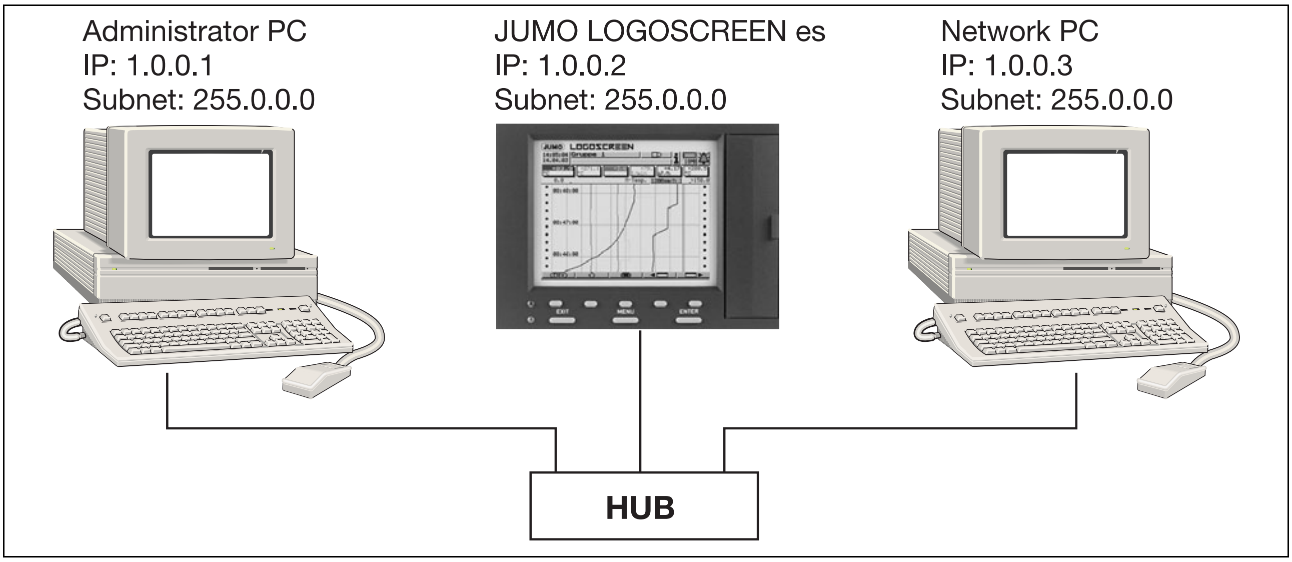


Fig. 43: Basic layout of the network

The PCs are connected to one another by patch cables to a hub.

Fig. 44: Hub front view

Fig. 45: Hub rear view, with the power supply and the patch cables for the three stations

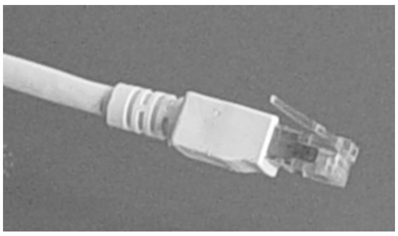


Fig. 46: RJ45 connector on the patch cable

The assignment of the subnet mask and the IP addresses can now be carried out.
A Class A network is to be set up.
We define the subnet mask as: 255.0.0.0 (Fig. 43).
As described previously, the first network address is 1 and the first host address is X.0.0.1.
For this reason, the first station receives the IP address 1.0.0.1.

The assignment of the IP addresses for the PCs can be made in “Control panel ➔ Network Connections”.

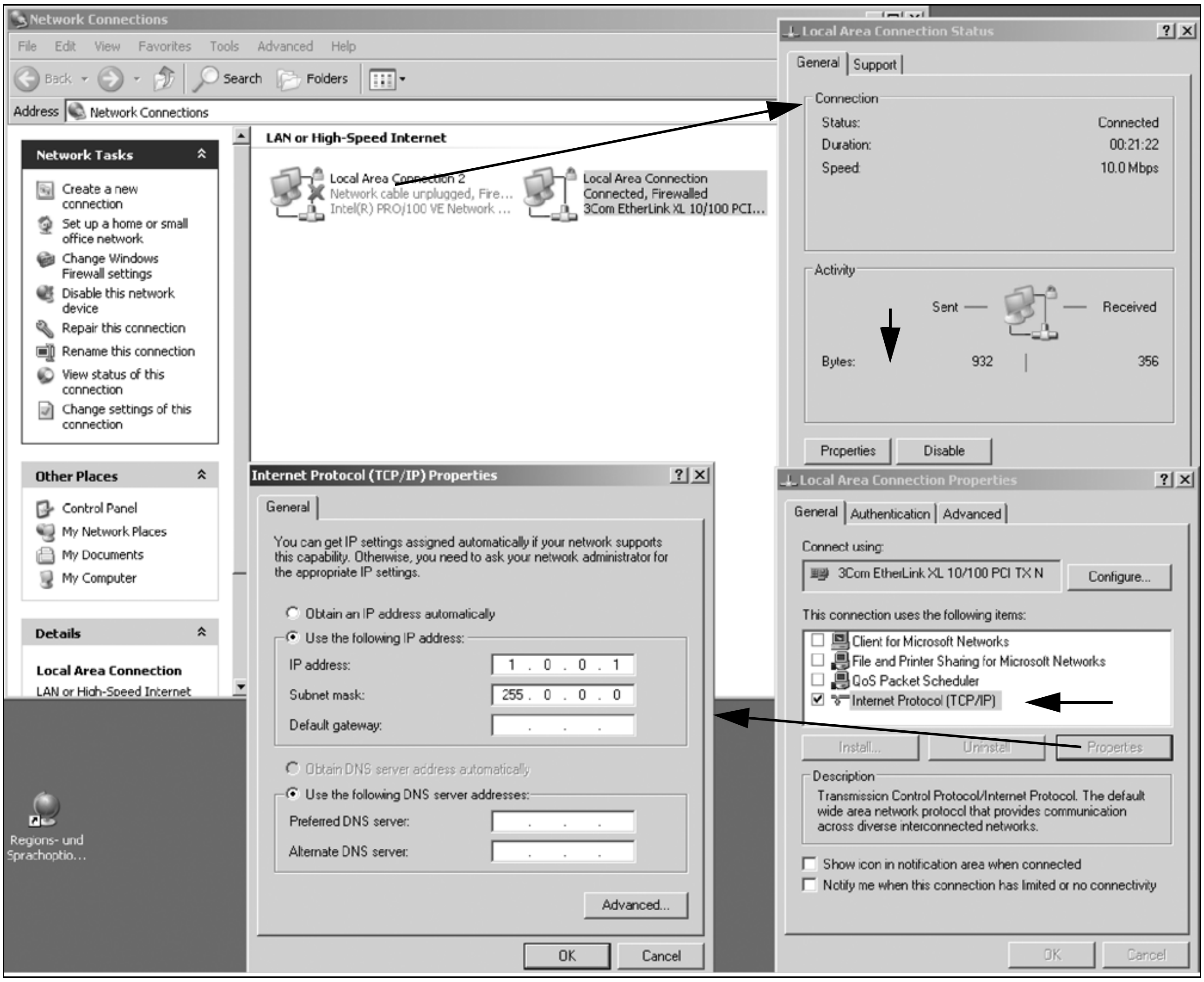


Fig. 47: Assignment of the IP address and subnet mask for the PCs

The IP address and the subnet mask for the PCs can now be entered in the window “Internet protocol properties (TCP/IP)”. ETHERNET is normally composed of partial networks that are connected together via routers or bridges. If a station sends a messages for a terminal station that is not within the same (partial) network, then the message is sent directly to the bridge or router. The IP address of the router or bridge must be defined under “Standard gateway”. Since our network is not connected to any other network, a gateway is not required and we can leave out this entry.

Note

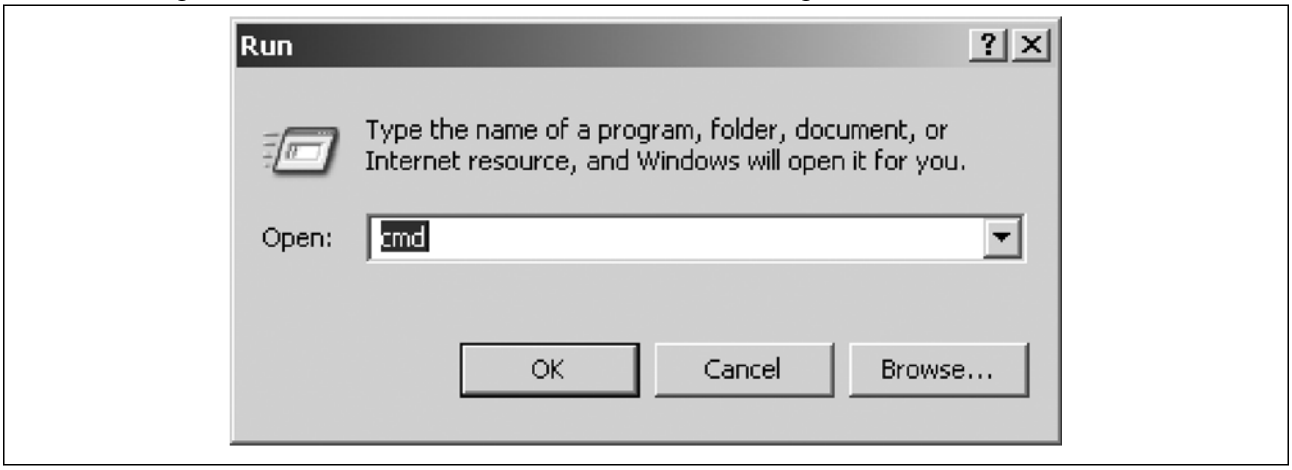
“Automatically assign IP address” must be selected if the IP addresses in the network are to be automatically assigned by DHCP (this is the usual case, but we are going to make fixed assignments for the parameters).

For the LOGOSCREEN es, the assignment of the IP addresses and subnet mask can be made on the device itself or through a configuration program.

After setting the parameters for the interfaces, the question arises:
Are the stations actually communicating with one another?

A way of testing this is shown below.

After selecting “Run” in the Windows start menu, and entering “cmd” ...



... you can test the communication with a station by entering “Ping IP” in the DOS window.

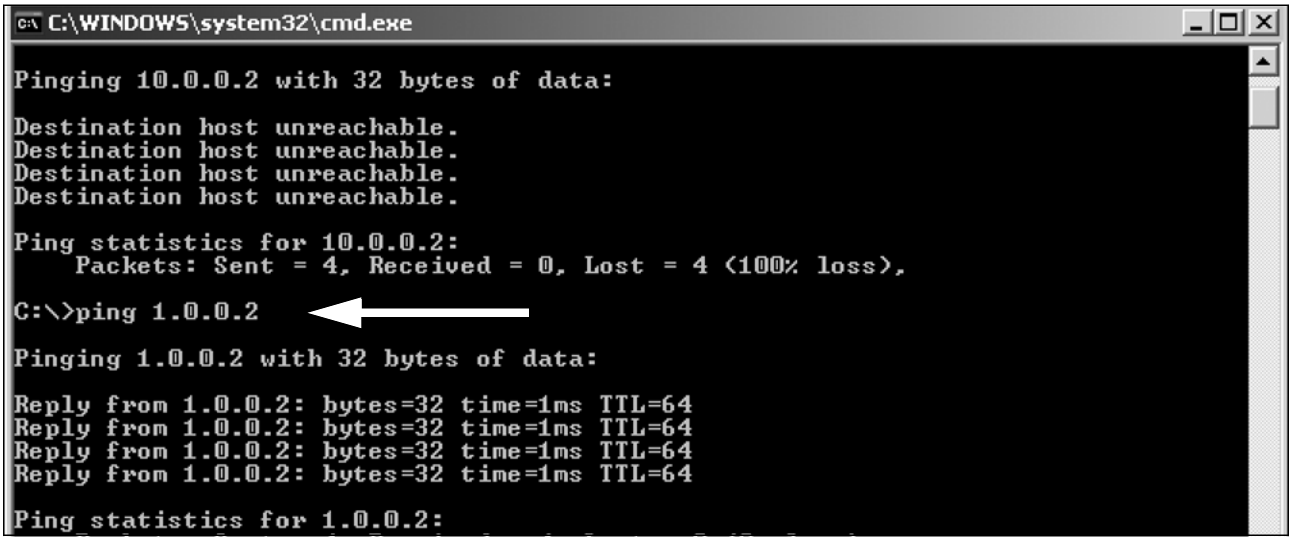


Fig. 48: Testing communication

Fig. 48 shows the response of the station with IP address 1.0.0.2, if communication is active. If the stations can be accessed by using PING, then you may assume that the connection is OK.

The JUMO LOGOSCREEN es can now be accessed from the PCs (all that is necessary is to enter the IP address for the recorder in the corresponding application program).

Integration of JUMO field devices in an existing company network

Basically, it is very easy to integrate an appropriate ETHERNET device.
A free network socket is required. The connection between this and the ETHERNET device is made with a patch cable. The network administrator must enter an IP that is reserved for this device (as already described, this must be excluded from the automatic DHCP assignment). The IP address, subnet mask and standard gateway must all be set up in the JUMO device. The device can now be accessed from any PC in the network, using the appropriate application program.

4.3.5 COM redirection

The COM redirection can be used to link MODbus devices to an ETHERNET.

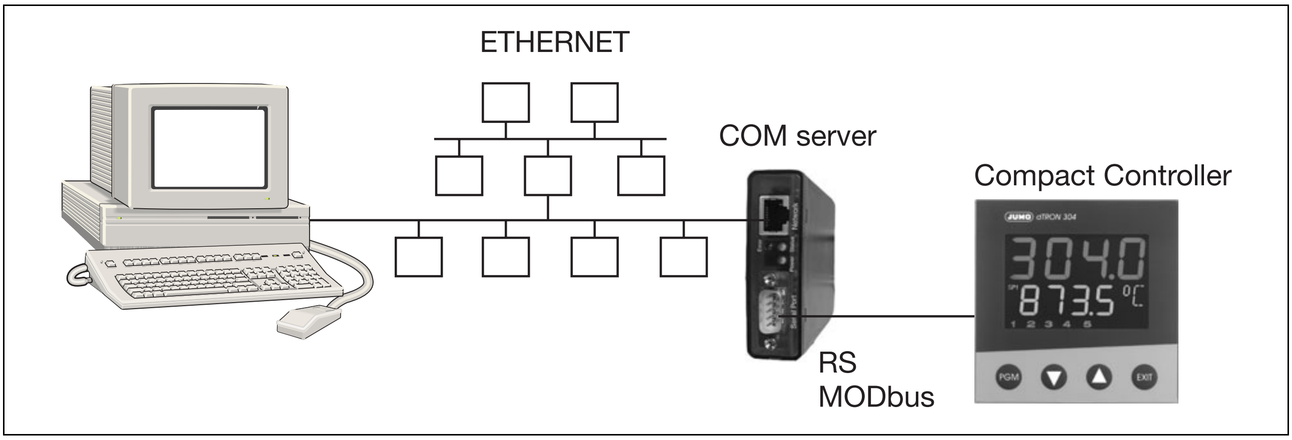


Fig. 49: COM server in the network

COM redirection is made by a COM server (e. g. “COM-Server Highspeed Industry”, see WuT.de) that is incorporated into the network. It has an IP address assigned, and a virtual COM interface is defined for a PC, using an application program (COM redirection). All the data that are now sent from the PC to this virtual COM interface are passed through ETHERNET to the serial interface on the COM server.

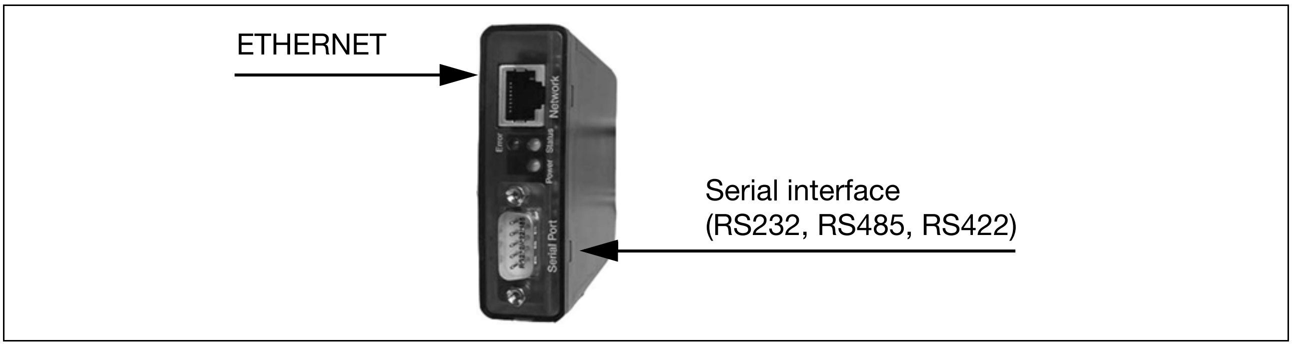


Fig. 50: COM-Server Highspeed Industry

The COM server mentioned above can be used to implement RS232, RS485 and RS422 interfaces. The type of interface is selected by DIP switches in the COM server.

Configuration of the COM server

The ex-factory settings for the COM server do not provide a valid IP address. In this case, the ARP program (**A**dress **R**esolution **P**rotocol) can be used (only once!) to assign a valid IP address.

ARP can be started in Windows, under “Start ➔ Run”. The instruction for a COM server looks like this:

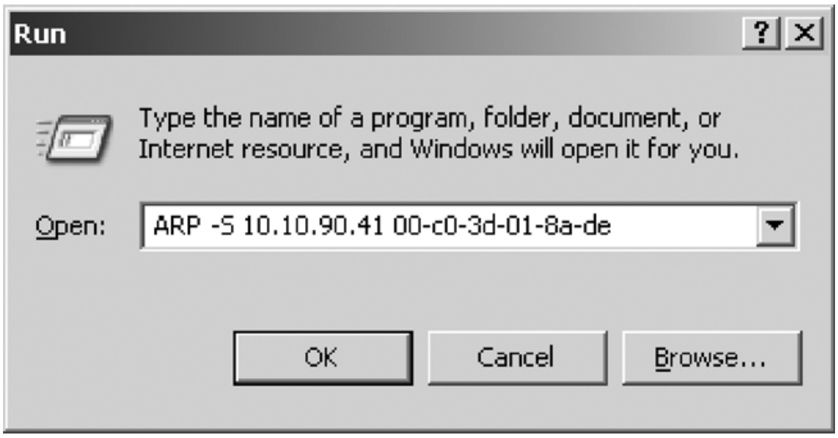


Fig. 51: Assigning an IP address

-S	Assign an IP address to the physical address
10.10.90.41	IP address that is to be assigned to the COM server. The IP address must be entered by the network administrator.
00-c0-3d-01-8a-de	Physical (MAC) address of the COM server. This address is in the COM server.

The COM server must now be configured via Telnet for the format of the data to be transmitted (transmission rate, stop bits etc.). Telnet is a text window (text-oriented program) that can be used for remote configuration of a terminal device in a network.

Telnet is started by entering “Telnet IP 1111”. If we want to use a PC to configure a COM server with the IP 10.10.90.41, then the start of Telnet looks like this:

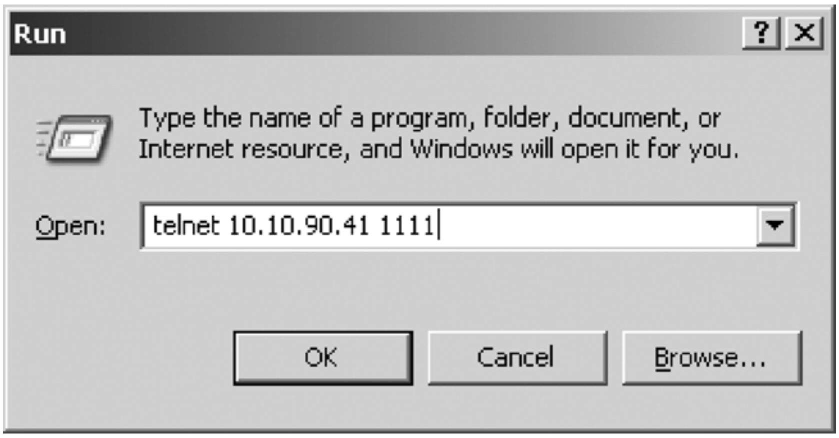


Fig. 52: Starting Telnet

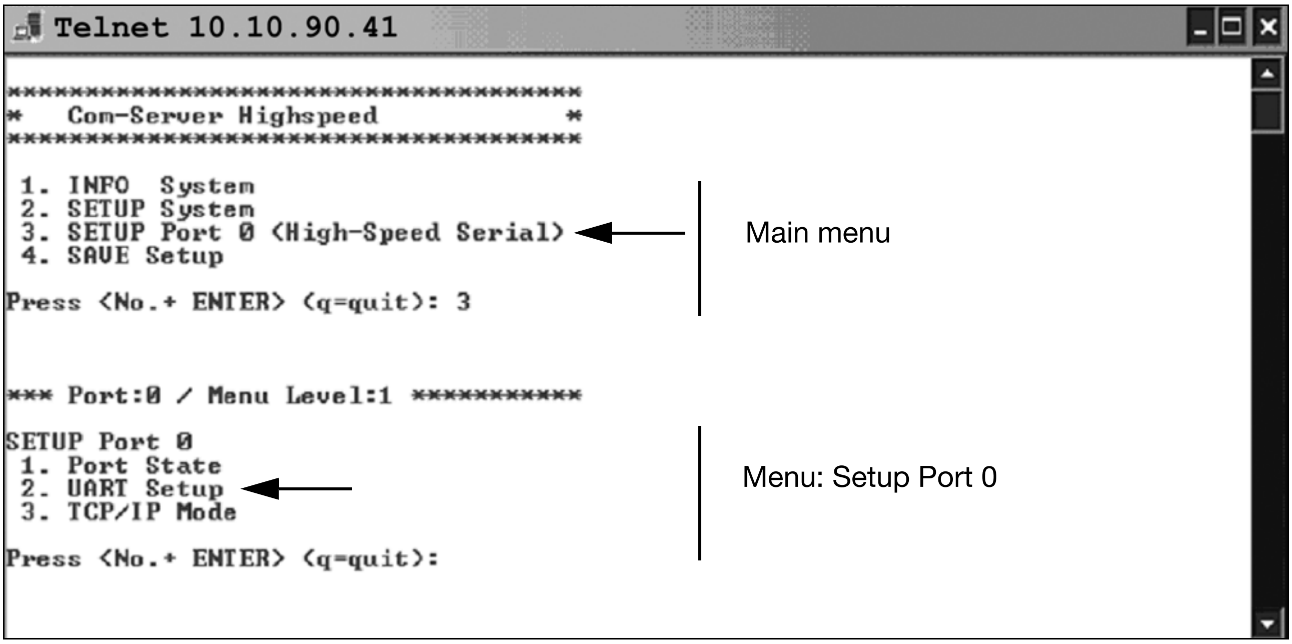


Fig. 53: Main menu for Telnet

After the start, the main menu for Telnet will appear. After selecting the menu “SETUP Port 0”, the sub-menu “UART Setup” can be used to set up the serial interface for the COM server (transmission rate, parity, stop bits etc.). The settings must match the configuration of the device on the MODbus side (RS232, RS485 or RS422 interface).

The COM server supports BootP. BootP can be used by a terminal device to ask for an IP address from a DHCP server. We recommend that you use Telnet to switch off this function when using COM servers in conjunction with JUMO devices.

It will still be possible to use Telnet to make a subsequent alteration of the IP address.

Now the COM server can be installed. After installation, the entry “COM redirection” will be visible in “Control Panel” on the PC.

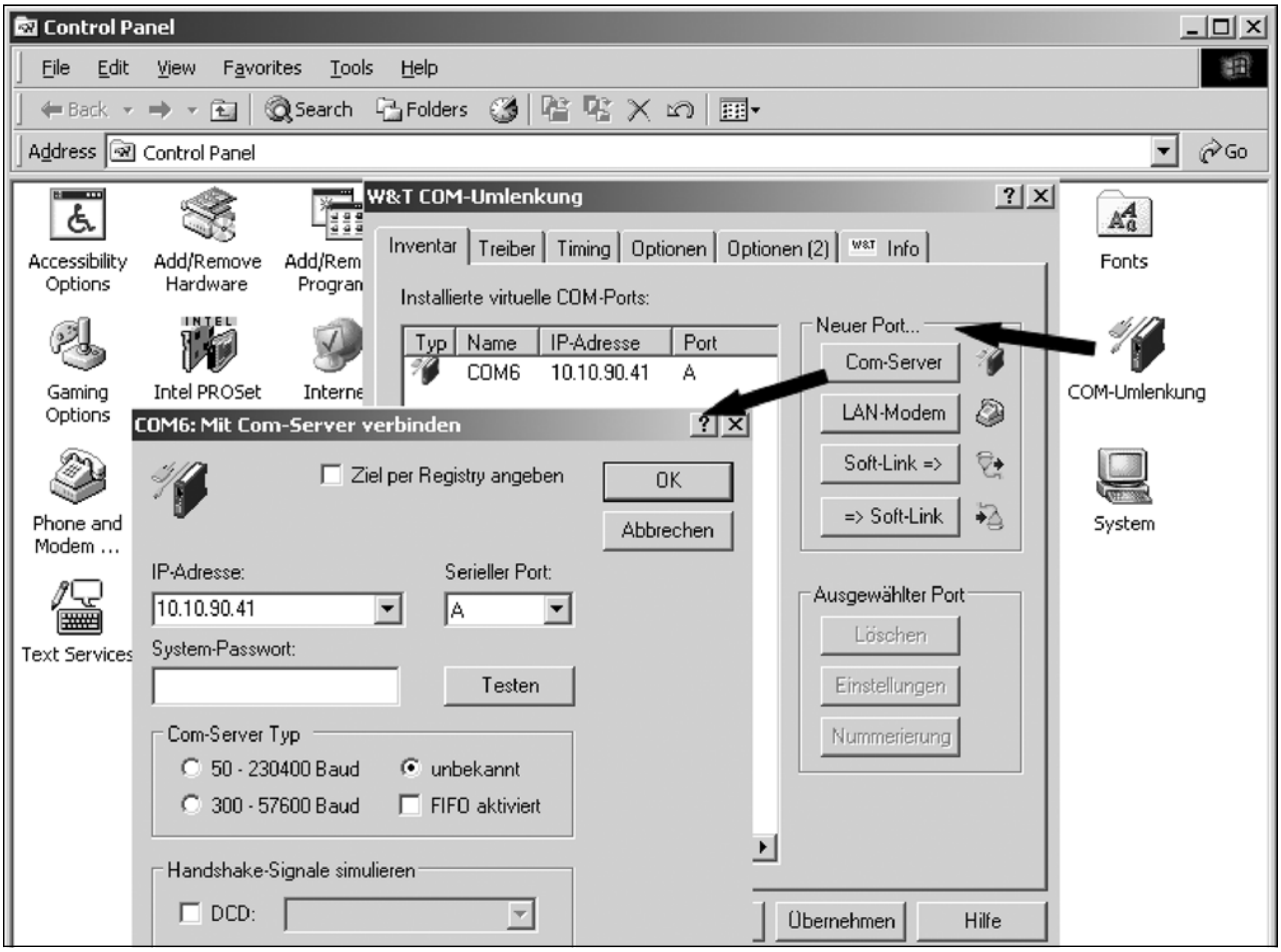


Fig. 54: COM server in “Control Panel”

The IP address for the COM server can now be entered in “COM redirection”.
All the data that are sent to the virtual COM interface (COM6 in this example) are passed through ETHERNET to the serial interface on the COM server.

JUMO has integrated the COM server in several application programs. In these programs you can make a direct entry for a field device that is attached to the serial interface of a COM server, and for which an IP address must be entered. In these cases, it is not necessary to install the COM redirection.

4.3.6 Enabling directories

When working with JUMO devices it will occasionally be necessary to access the data on a PC from another PC. It is conceivable that the measurement data from field devices are stored on PC1, but they need to be available on PC2.
In this case, the appropriate directory on PC1 must be enabled for access. To enable the directory in Windows Explorer, select it with the right mouse button, and then select “Sharing and Security” to enable it.

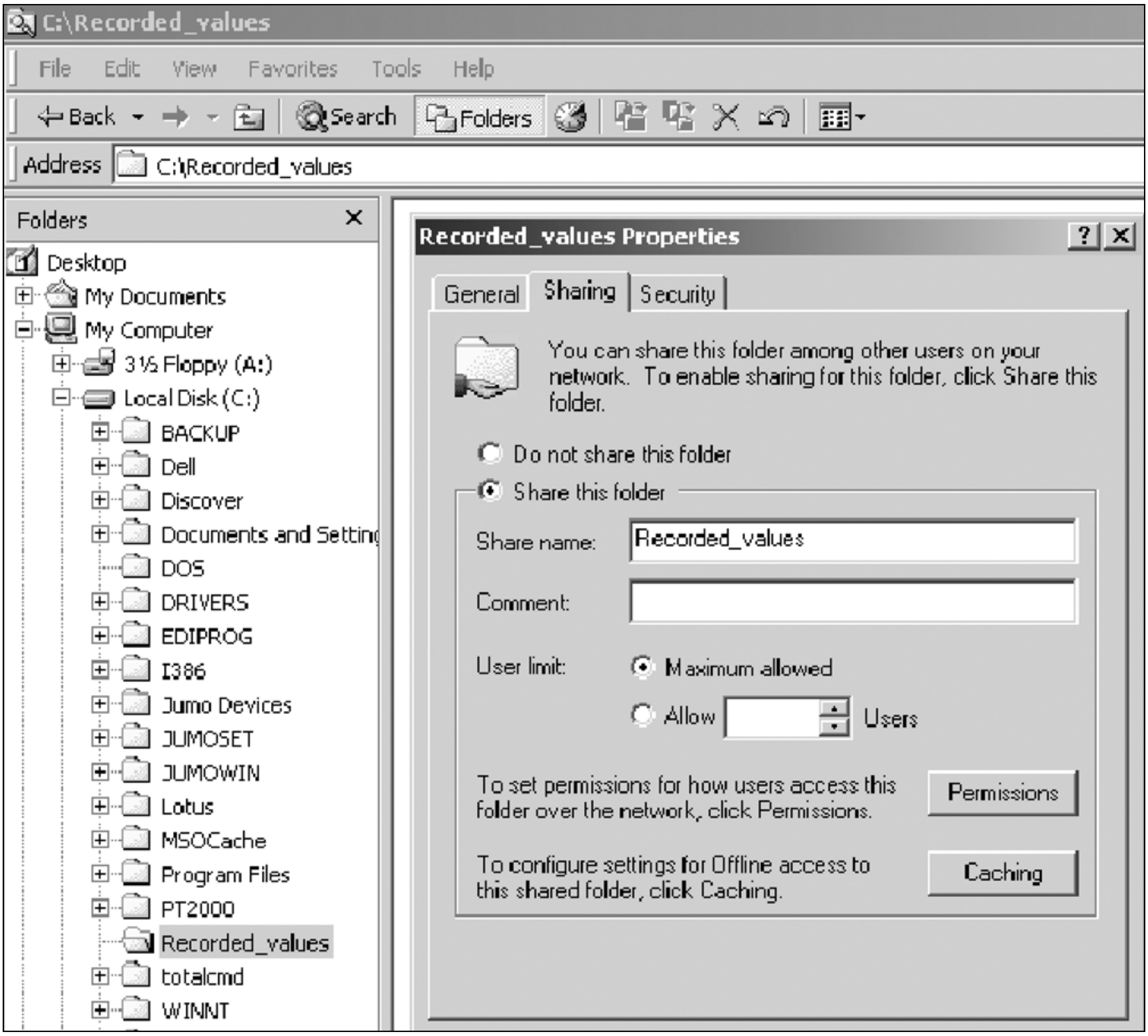


Fig. 55: Enabling folder access

Select “Share this folder” to enable access. After sharing has been enabled, the folder is marked by a hand. The shared directory and its contents are now available under the “Share name” in “Network Neighborhood”.

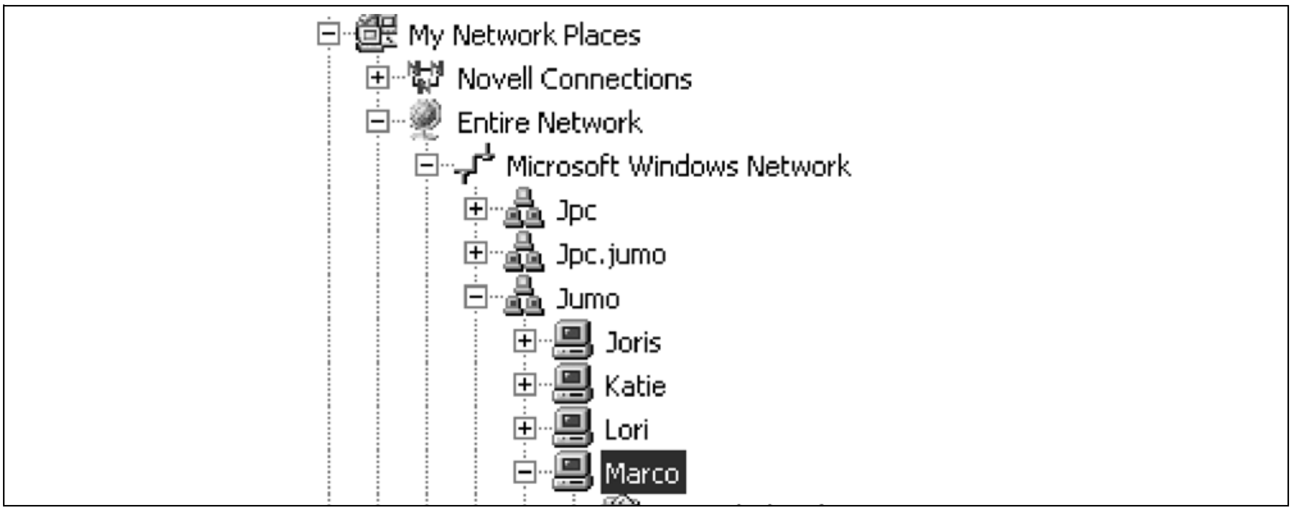


Fig. 56: Accessible folder in the network neighborhood

4.4 CAN and CANopen

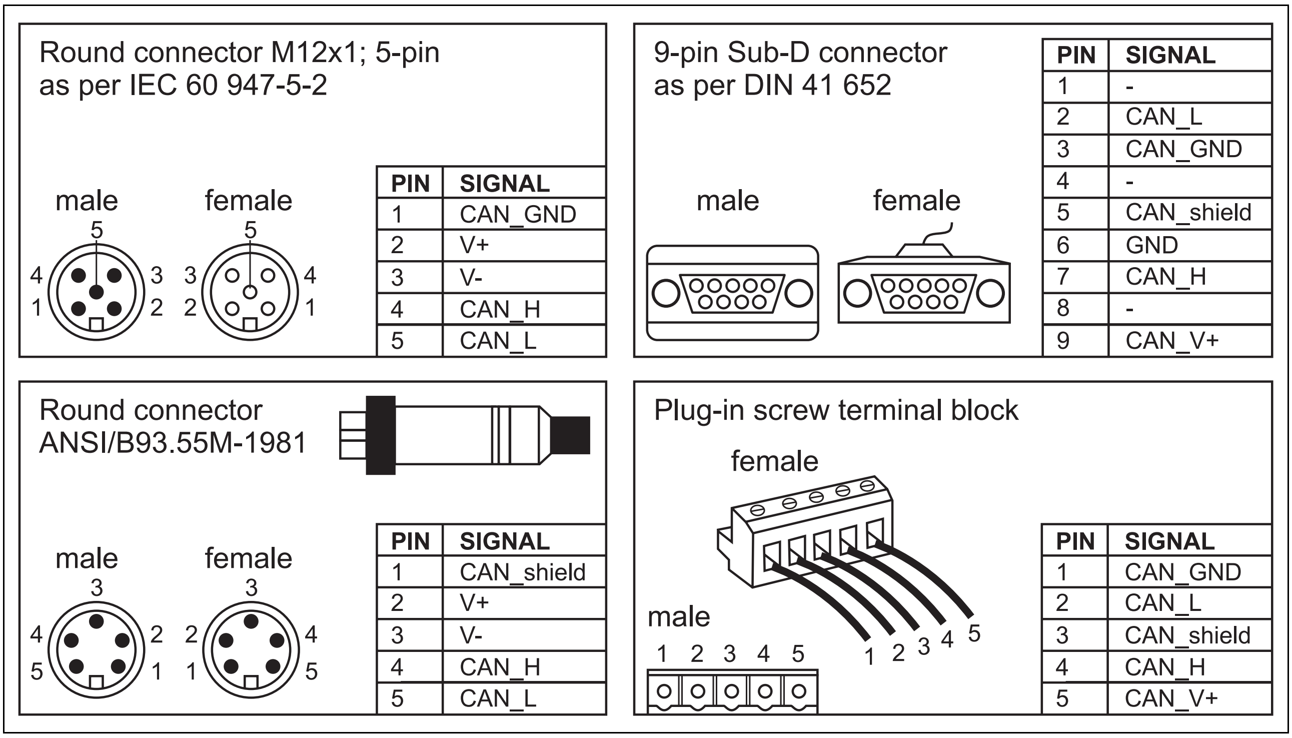
4.4.1 CAN

CAN is a very reliable bus system, that was developed by the Robert Bosch company in 1986. CAN was originally developed for application in vehicles, but is increasingly being used in industrial sectors.

CAN stands for **C**ontroller **A**rea **N**etwork, and is a serial data bus. Short messages are transmitted by CAN without any interference. This interference-free transmission is facilitated by the addressing of the messages, amongst other features.

The signal is transmitted via two wires (CAN_H and CAN_L). The stations are linked together in a linear topology on a shielded two-wire cable, whereby the cable must be fitted with termination resistors at the first and last stations in the line.

Fig. 57: CAN station connection



Round connector M12x1; 5-pin as per IEC 60 947-5-2

PIN	SIGNAL
1	CAN_GND
2	V+
3	V-
4	CAN_H
5	CAN_L

9-pin Sub-D connector as per DIN 41 652

PIN	SIGNAL
1	-
2	CAN_L
3	CAN_GND
4	-
5	CAN_shield
6	GND
7	CAN_H
8	-
9	CAN_V+

Round connector ANSI/B93.55M-1981

PIN	SIGNAL
1	CAN_shield
2	V+
3	V-
4	CAN_H
5	CAN_L

Plug-in screw terminal block

PIN	SIGNAL
1	CAN_GND
2	CAN_L
3	CAN_shield
4	CAN_H
5	CAN_V+

Fig. 58: CAN connector selection

As can be seen in Fig. 58, the CAN connecting cable is also frequently used for the distribution of the supply voltage (V+ und V-) to the stations.

When CAN is used, any station can start transmitting on its own accord. Every message is assigned a priority. There is only one instance of a priority in each system. If two stations start transmitting, the higher-priority message will obtain access and be transmitted. In Fig. 59, two stations start transmitting a message simultaneously. During the transmission, the stations listen in to the bus, to hear whether the information matches the information they are transmitting.

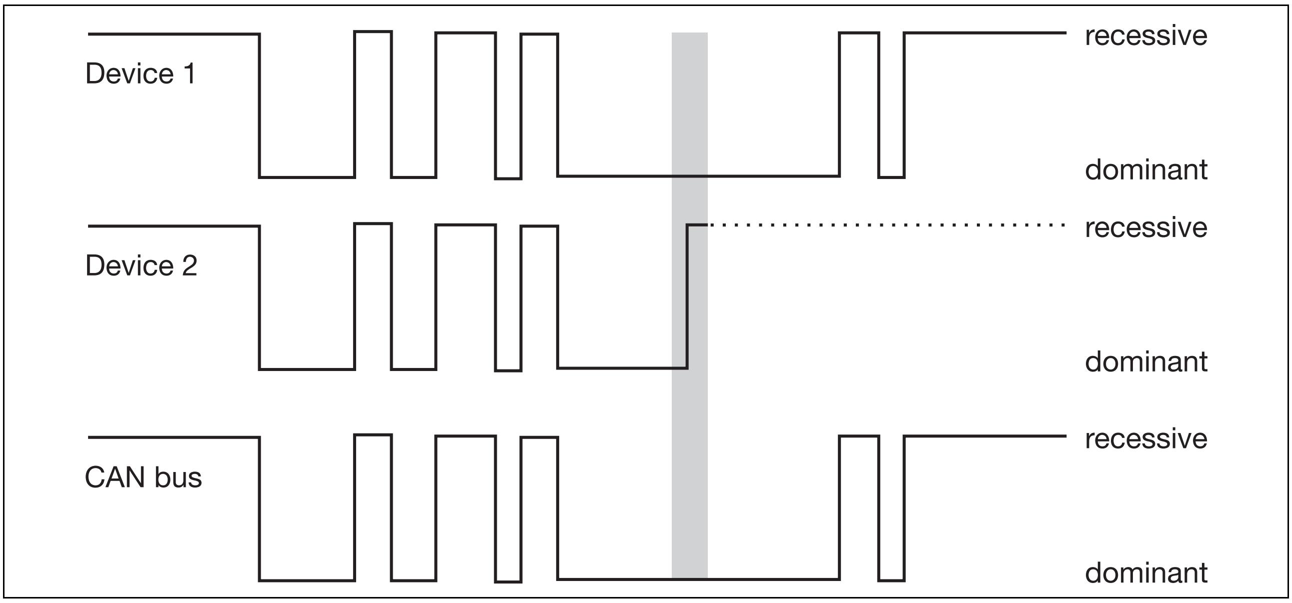


Fig. 59: CSMA-CA procedure

Device 1 and Device 2 start transmitting a message simultaneously. The information is identical up to the marked point, i.e the data on the bus correspond to the message that each station is transmitting. However, the information differs after the marked point (Device 1 is sending level 0, but Device 2 is sending level 1). Level 0 is the dominant level, as this can appear on the bus when only a single station transmits this level. Device 2 detects that the level on the bus (level 0) is different from the level of the bit it is transmitting (level 1). So Device 2 will disconnect from the bus, and send its information again at a later time. In our example, Device 1 is transmitting the message with the higher priority, and this will gain access. As already described in Chapter 3 *“Access procedure and transmission media”*, this is the method known as the CSMA-CA procedure.

The message priority is defined in an “identifier”. Fig. 60 shows the arrangement of a CAN protocol.

Identifier	Data field (0 — 8 bytes)	CRC field

Fig. 60: Protocol structure of a CAN telegram

The identifier (COB-ID) for JUMO sensors consists of 11 bits. In this format it is possible to address up to $2^{11} = 2048$ different messages. As well as this standard format, there is also an expanded format with 29 bits. In this format it is possible to address up to $2^{29} = 536,870,912$ different messages.

The identifier follows the actual information, such as a pressure or temperature measurement. Here you can see that only a very limited amount of information can be transmitted in a telegram (maximum 8 bytes). After the data field comes the CRC field, which is used to transmit the checksum.

This checking makes it possible for the CAN system to detect up to 6 wrongly transmitted bits within one protocol (Hamming distance HD = 6).

The maximum transmission speed for CAN is 1 Mbps with a 25 m bus length. At a transmission rate of 20 kbps, it is possible to have a bus that is up to 2500 meters long.

In order to facilitate communication between devices using CAN, various aspects must be defined. The most important item here is the assignment of the CAN identifiers that are available in the network to the individual devices. In addition, definitions must be made for the contents of the data transmissions. The interpretation of the data contents is most important here. And, last but not least, a higher-level (supervisory) device monitoring is also necessary. All this is gathered together in the application layer.

Protocols that clearly define the above-mentioned items are: CANopen, DeviceNet, CANKingdom, Smart Distributed System, In-Vehicle Networking, J1939.

4.4.2 CANopen

We will now take a look at CANopen – a standard that is widespread in Germany and Europe.

The method of operation of a CANopen node will be described, and the way in which such a component is configured. JUMO supplies pressure and temperature transmitters with CANopen. The explanations are made using the JUMO CANtrans T temperature transmitter as an example.

Fig. 61: JUMO CANtrans T

With CANopen, a maximum of 127 nodes can be joined in a linear topology (as described for CAN).

The configuration of the stations is made with a manufacturer-independent application program, and then transmitted to the stations via the bus. This requires an interface for the connection between the PC and the CAN node.

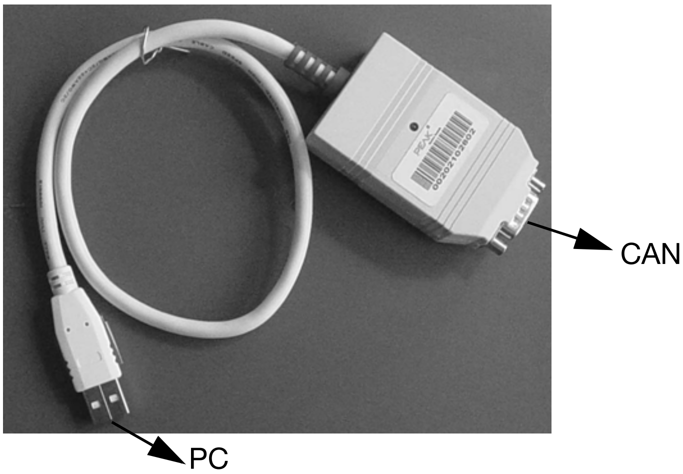


Fig. 62: CAN interface

Fig. 62 shows a CAN interface, which is connected to the USB interface of the PC and enables access to the CAN bus on the other side.

We will now look at the configuration of a node, using the program CANsetter from the Vector company as an example. The configuration parameters are addressed through an object directory. In order to obtain access to this directory, the manufacturer must provide an EDS (Electronic Data Sheet) file for a CANopen device.

In Fig. 63 we “start up” CANsetter and configure the JUMO CANtrans T transmitter. The transmitter operates with the standard setting of 500 kbps. The factory setting for the device address is 125.

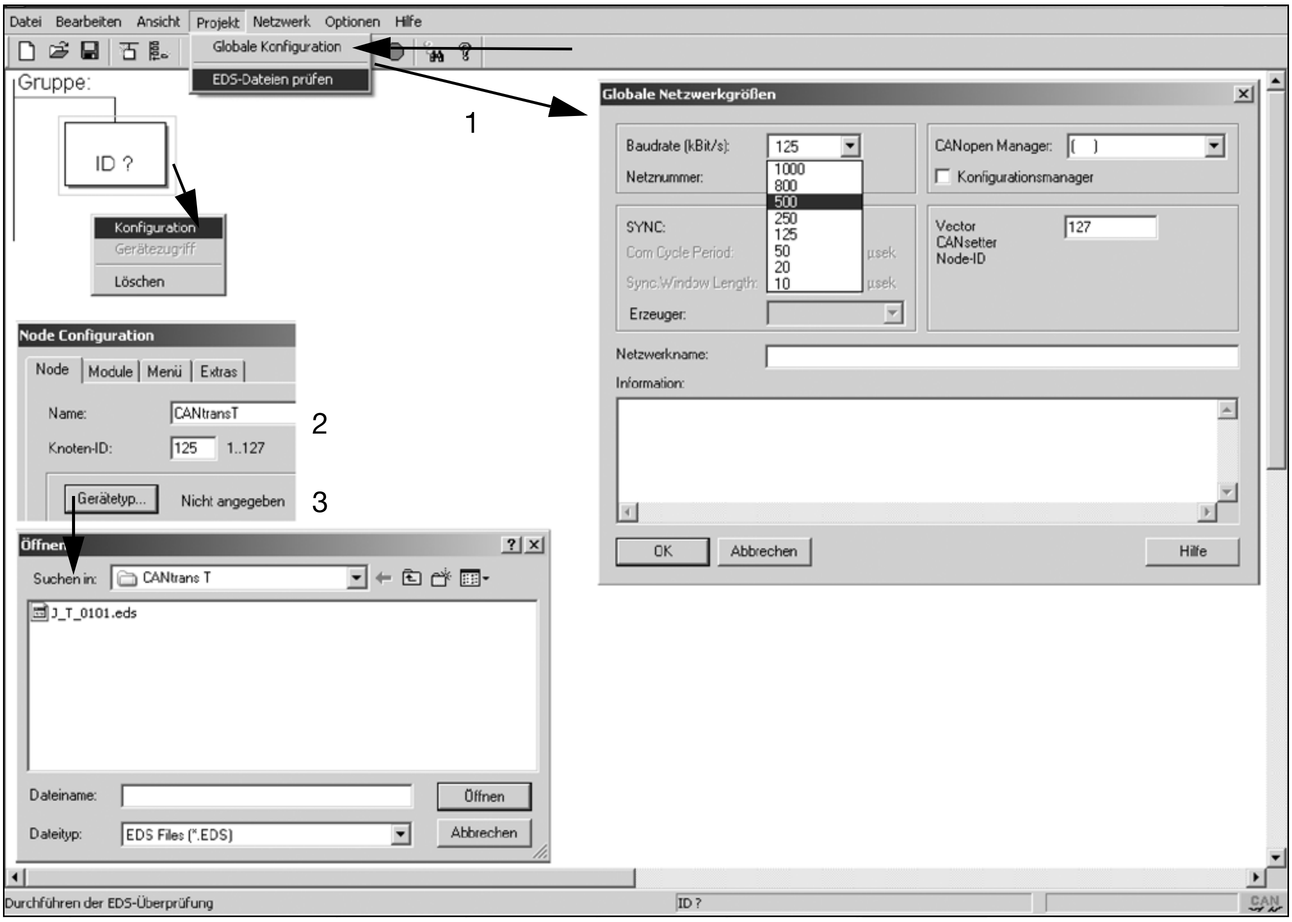


Fig. 63: Access to the object directory for a JUMO CANtrans T

1. In order to access the JUMO CANtrans T, the data transmission rate is set to the same speed as the JUMO CANtrans T (500 kbps) in the menu item “Project - Global Configuration”.
2. A new node (ID?) is created as soon as the program is started.
   This is assigned to the address 125 in the “Node Configuration” window.
3. The EDS file that was provided by the manufacturer is now selected in “Device type”.

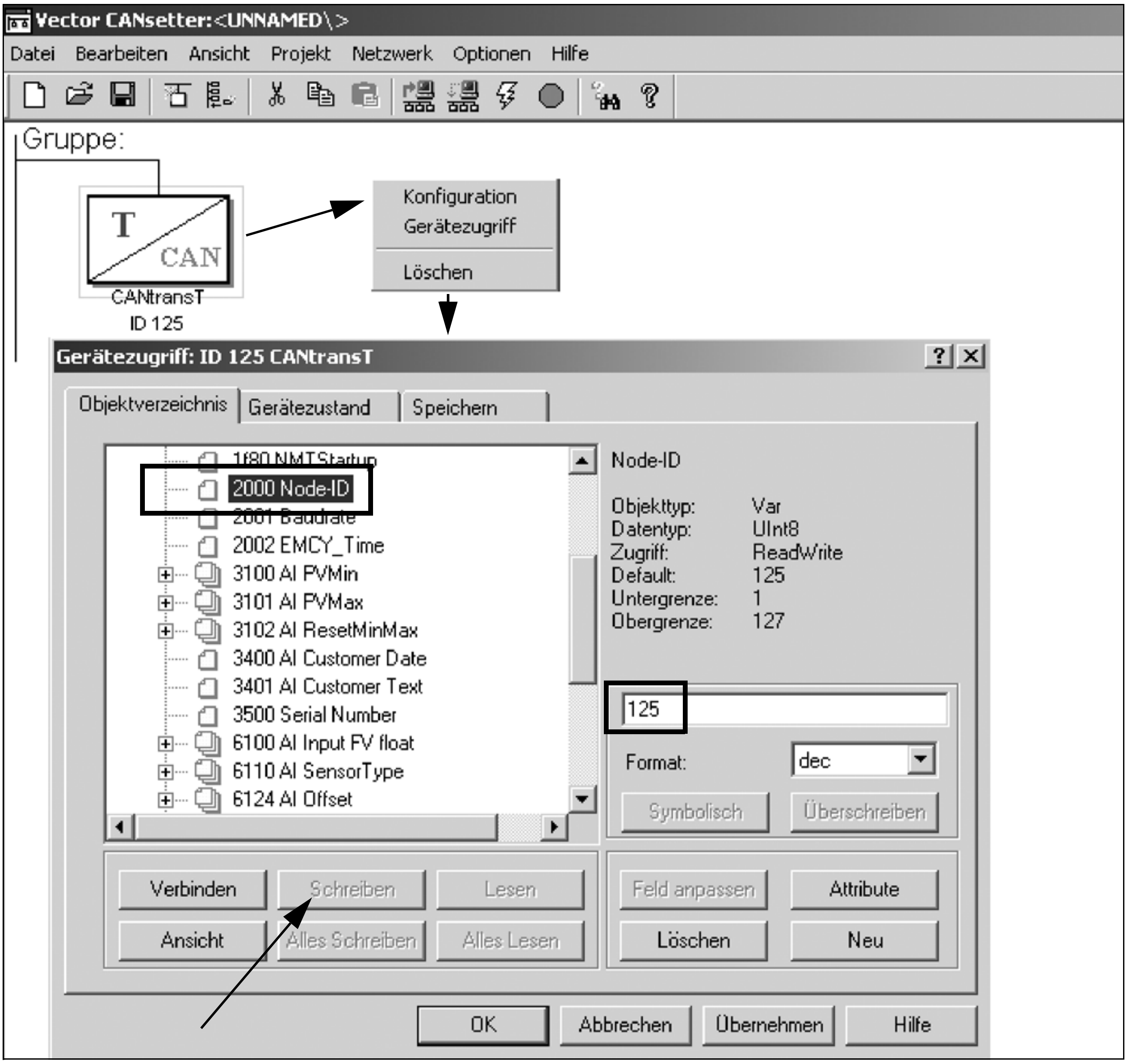


Fig. 64: Configuration of a JUMO CANtrans T

After the node has been created, it is possible to access its object directory by selecting "Configuration". The object directory is a table in which all the configuration parameters can be accessed as 4-figure hexadecimal numbers. As can be seen in Fig. 64, the station address, for instance, can be accessed through 0x2000 (0x = identifier for hexadecimal numbers, Node-ID = station address). The address 125 can be seen in the marked field. This could be altered, and after "Write" has been selected, the new address would then be transmitted to the JUMO CANtrans T. All parameters can be altered in the same way.

Process Data Object (PDO)

When the node configuration has been completed, the transmitter can start transmitting measurements. The measurement (in the case of a JUMO CANtrans T, this will be a temperature) will be transmitted in what is called a "PDO". The PDO is placed in front of the COB-ID (identifier) that has already been described in this chapter. The COB-ID can be configured. The factory setting is 0x180+Node-ID. For a temperature transmitter with the address 125 (decimal), the factory setting for the COB-ID would be 0x180+125 (decimal) = 0x1FD = 00111111101 (binary).

In our example, the telegram would look like this:

0	0	1	1	1	1	1	1	1	0	1	Temperature	CRC field

COB-ID

Fig. 65: Telegram for a JUMO CANtrans T with a COB-ID 0x1FD

When does a transmitter send the measurement – the PDO?

It is possible to configure the time when the transmitter will send the PDO. A distinction is made between event-controlled and synchronous transmission.

An event-controlled transmission is made

- if the measurement has changed by a predefined amount (delta) since the last transmission,
- (repetitively) after the elapse of a preset time (event timer),
- if a hardware fault occurs, or overrange/underrange (toggle),
- on switching over to “Operational” (explanation follows),
- as a response to a request from a master (RTR signal received).

As is clear from this list, there are many possible reasons for transmitting the PDO. So it is therefore possible to set a minimum time between two PDOs (inhibit timer). This parameter can, for example, be used to define that the measurement is not transmitted more frequently than once per second (even if, for instance, the transmit conditions had already been fulfilled 5 times).

As an alternative, synchronized transmission can be used.
The transmitter can be configured in such a way that it only sends out the PDO if it has received a cyclic request from a master (SYNC has been received). In this case, the JUMO CANtrans T does not transmit according to event-controlled conditions, such as delta, event timer, toggle, or change to Operational.

How are two stations connected with CANopen?

The JUMO CANtrans T possesses a PDO with which it transmits the measurement. All stations listen in to the message, but only the station or stations for which the message is intended will process it. Stations that are able to receive a message via CANopen require at least one Receive-PDO. This Receive-PDO can be used, for instance by a controller that has a CANopen interface, to receive the setpoint. The same COB-ID is set for the Input-PDO of the controller as that which is used by a transmitter, for instance, to send the value.

Example

If the controller is supposed to accept the temperature measurement from the JUMO CANtrans T (COB-ID 0x1FD) as the setpoint, then its Input-PDO will also be set to the COB-ID 0x1FD.

Emergency Object (error message)

If there is a change of status with regard to hardware errors, the JUMO CANtrans T will transmit a high-priority error message. The Emergency Object is transmitted repeatedly, with a cycle time that is set (in milliseconds) in the Object Directory.

Assignment of the Node-ID (device address) and baud rate through LSS (Layer Setting Services)

Example:
The system operator is using 10 JUMO CANtrans T temperature transmitters. As already mentioned, these transmitters are factory-set to device address 125. However, it is not permissible for the stations to have the same address and be simultaneously accessed through the bus. In this case, LSS can be used to alter the device addresses (and the baud rates as well). For LSS, the stations can be accessed through the data that are marked on the nameplate

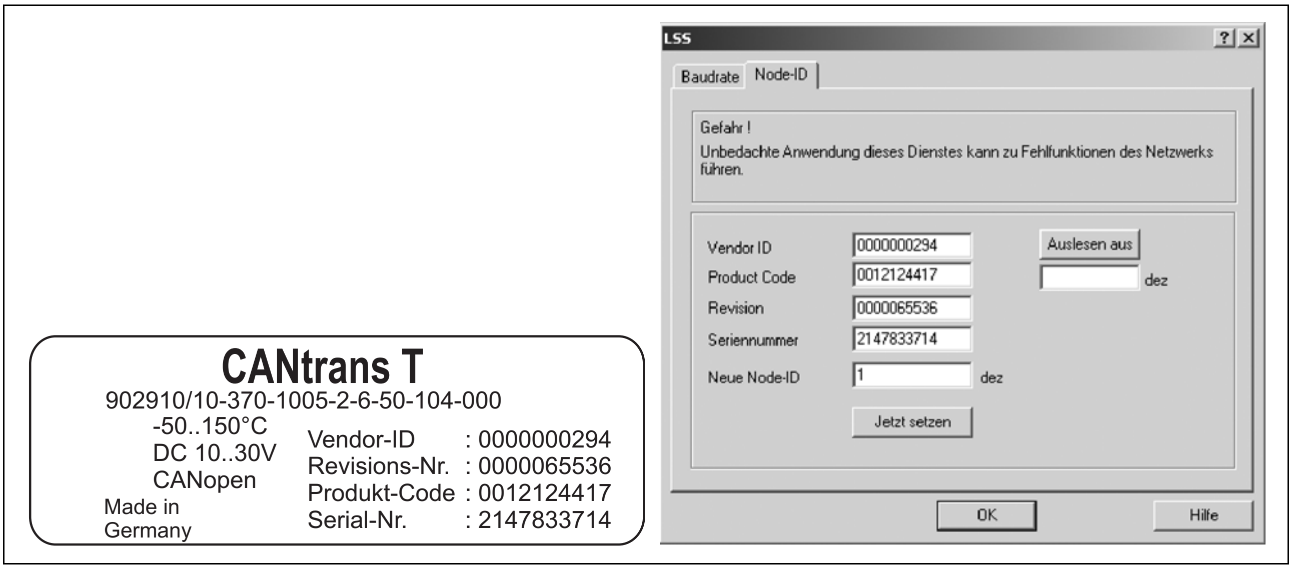


Fig. 66: Nameplate for a JUMO CANtrans T, and the LSS sub-menu from the configuration program

The details on the nameplate (Vendor-ID, Revision-No. etc., Fig. 66, left) are used to identify the transmitter in the manufacturer-independent configuration tool (Fig. 66, right). The transmitter can be accessed, and the baud rate and address can be altered. After restarting the system (an interruption of the supply voltage), the nodes will have accepted the addresses and baud rates that have been set, and further configuration can be carried out.

Pre-Operational, Operational

As a rule, after the supply voltage has been switched on, the transmitter will be in the Pre-Operational mode, which means that it is not yet transmitting data. The sensors will usually be switched into the Operational (= active) state by a higher-level control (= a master). This behavior is programmed in the factory. The transmitter can be configured in such a way that it automatically switches into the Operational state when the electrical supply is switched on (i.e. it will be active, transmitting its PDO and so on).

The entire configuration of the transmitter is made through the Object Directory.
Fig. 67 shows the method of operation of the JUMO CANtrans T.
It also clarifies where the individual parameters can be altered in the Object Directory.

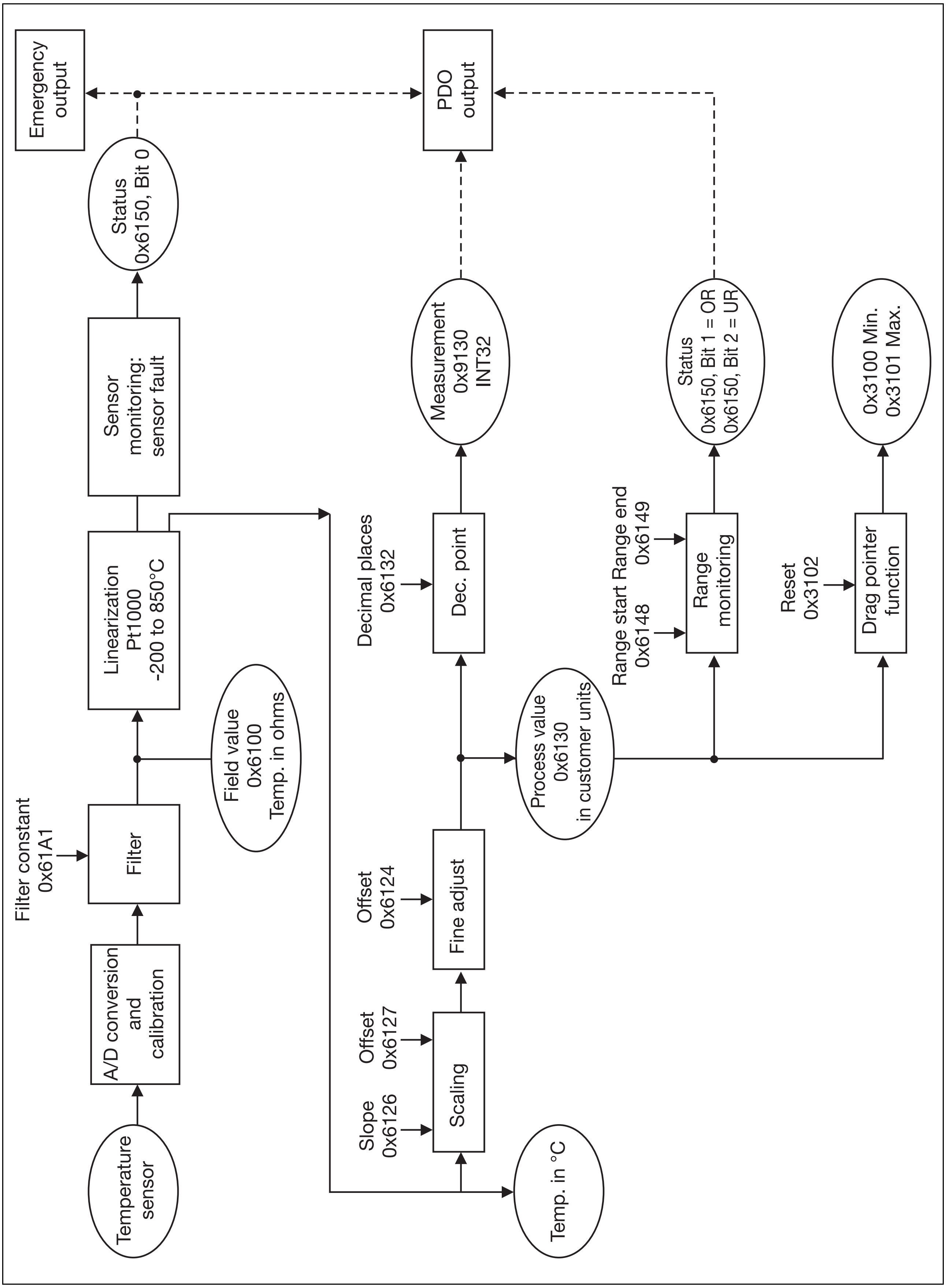


Fig. 67: Data flow for a temperature channel

Follow the path of the measurement from the temperature sensor (Fig. 67). After the filter (the dimensioning of the filter can be made under 0x61A1 in the Object Directory), the measurement is available in ohms (the transmitter contains a Pt1000 resistance sensor). The measurement can be read out under 0x6100 in the Object Directory.

After linearization, the customer can apply a scaling with a fine adjustment. In this way, the temperature (available under 0x6130) can, for instance, be converted from Centigrade to Fahrenheit.

The JUMO CANtrans T can also carry out monitoring of the measurement range. In the event of an overrange (see Range End 0x6149) or underrange (see Range Start 0x6148), the PDO will be transmitted. In the PDO, additional status bits are transmitted as well as the measurement, and they indicate whether an overrange of Range End or an underrange with respect to Range Start has occurred.

4.5 HART®

4.5.1 HART® protocol

The idea behind this technique is that a single cable pair is used to transmit the analog 4 — 20mA signal, as well as the digital communications signals. HART® stands for "**H**ighway **A**dressable **R**emote **T**ransducer" and is a registered trademark of Rosemount Inc.

HART® technology is largely standardized, and offers the advantage that just one universal operating and control device can be used to configure HART® devices from various manufacturers. This is made possible by the fixed definition of commands that are the same for all HART® devices.

HART® is suitable for applications in **Ex** areas.

HART® technology is supported around the world, by the HART Communication Foundation (HCF) that was founded in 1993.

4.5.2 FSK method

The communication method for HART® is based on the FSK principle (**F**requency **S**hift **K**eying). The 2-wire cable carries a 4 — 20mA proportional current signal representing the measured value, on which is superimposed a low-level AC signal (500mV peak-peak, no DC component) that uses logical "0" = 2400Hz and logical "1" = 1200Hz. This digital communication is used not only to transmit the measurement variables, but also the configuration parameters.

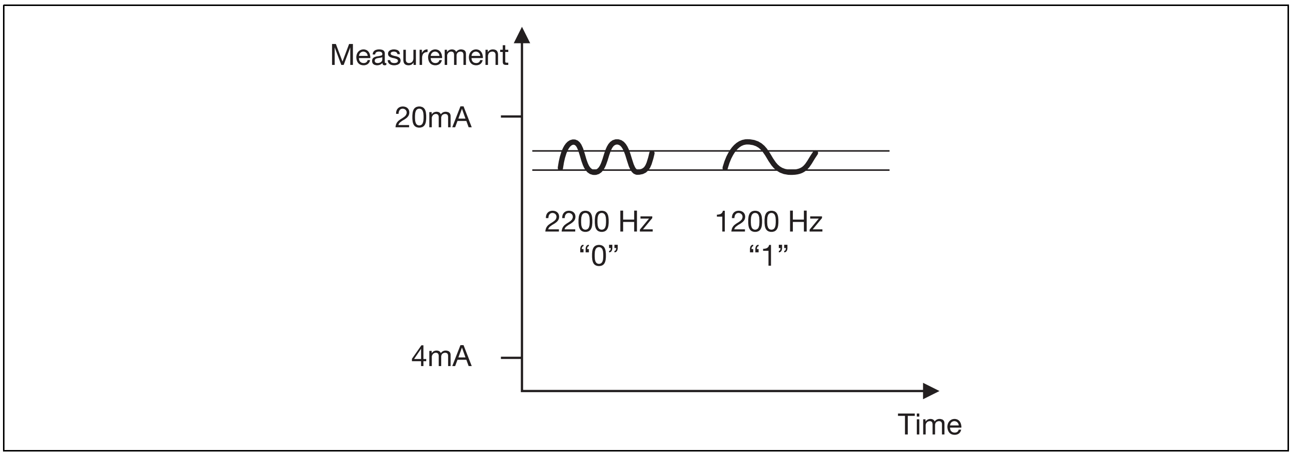


Fig. 68: HART® communication

HART® devices are usually operated in a point-to-point connection, or in a star configuration (using appropriate field multiplexers).

4.5.3 Multidrop operation

HART® can also be used for bus operation:
In the mode known as “Multidrop” operation, up to 15 devices can be connected in parallel on a 2-wire cable and connected to an automation system (linear topology). For this type of operation, each device requires a bus address, and the analog output signal is set to a fixed value of 4mA, which is merely the supply for operating the device. Now there is only a digital exchange of information, using the FSK signal. The master (up to 2 masters may be used) carries out a sequential polling for the values from the devices. One or two telegrams per second can be transmitted via the HART® communication. If the maximum of 15 HART® devices are connected to a bus, the polling cycle will take several seconds!

In general, HART® communication (especially in Multidrop operation) is not particularly suitable for operation and control tasks within a system, but rather for commissioning and maintenance applications. The current signal is the one mostly used for continuous operation.

4.5.4 JUMO devices with HART®

JUMO produces and supplies pressure and temperature transmitters with HART® – also for **Ex** areas. The procedure for configuring a JUMO device that has HART® communication can be demonstrated in the example of the JUMO dTRANS p02 pressure transmitter.

Fig. 69: Pressure transmitter JUMO dTRANS po2 and temperature transmitter JUMO dTRANS T01 HART®

The transmitters are fitted with a 4 — 20 mA analog output. The digital communication signal is superimposed on this output, as is usual for sensors that use the HART® protocol.

In addition to the basic pressure signal, the JUMO dTRANS p02 also transmits other data via the digital signal, such as temperature, maximum pressure etc.

4.5.5 Communication with a PC

Communication between a PC and a JUMO dTRANS p02 requires a normal, commercially available HART® modem that converts the HART® frequency signal to RS232 interface levels. A burden resistor is also inserted in the circuit, so that the varying current signal is made available in the form of an AC voltage signal.

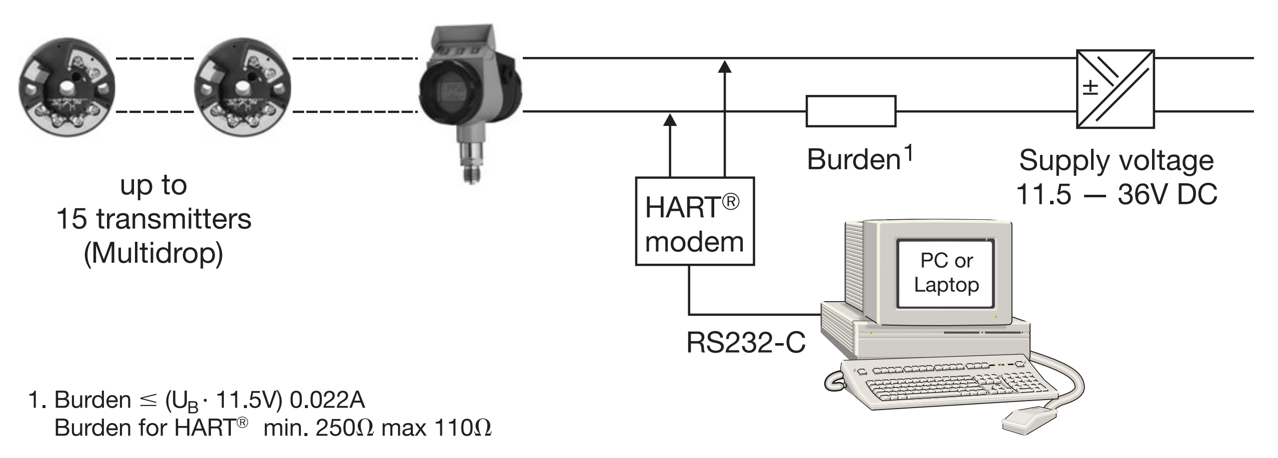


Fig. 70: Communication between PC and transmitter

After communication has been established with the PC, the setup program for the JUMO dTRANS p02 can be used to set the parameters for the transmitter. The setup program features a VDI /VDE2187 user interface for field devices. This user interface is used to set up the parameter data for the device, such as zero point, gain, device address etc., and to transmit them to the device by means of the HART® communication. In addition, measurements from HART® devices can be visualized and recorded.

4.5.6 HHT

An HHT (**H**and**h**eld **T**erminal) can also be used for setting the parameters for the transmitter.

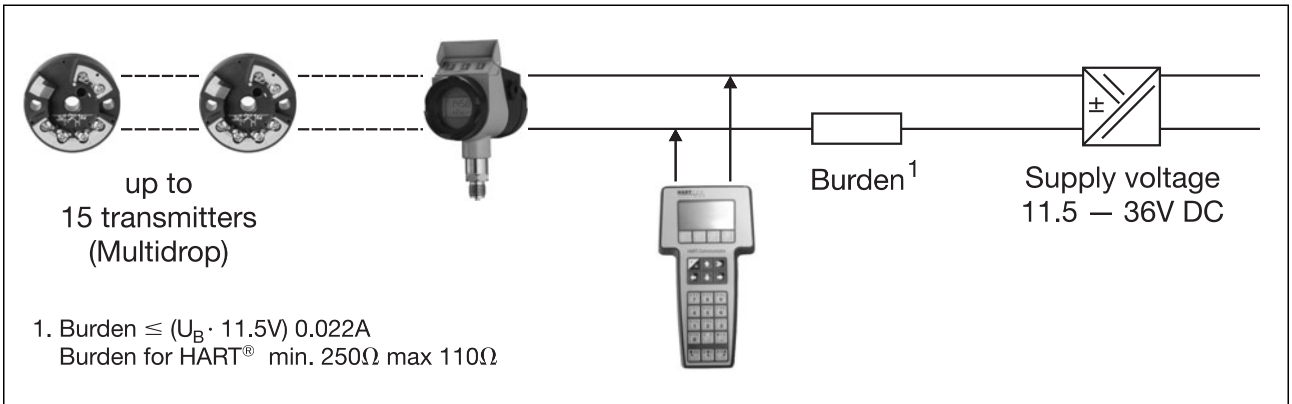


Fig. 71: Communication between HHT and transmitter

The HHT has an integrated modem, and is connected to the transmitter together with a burden resistor. It can operate with all HART® devices, regardless of the manufacturer, and can also be used in **Ex** areas.

4.5.7 Keypad / LCD

If HART® communication is not used, parameter setting for the JUMO dTRANS p02 is made in the usual way, by operating the keys on the device itself.

Modems

The word “modem” is a combination of the terms **M**odulator and **D**emodulator. A modem receives data (e. g. from a PC) in digital form and converts this data into, for instance, an analog signal (in the case where an analog telephone network is used). The receiving modem demodulates the data again, whereby they are once more available for the receiving equipment in digital form.

JUMO uses modems for such applications as

- Teleservice:
  A setup program, for instance, can be used to indicate the states of all the inputs and outputs of a controller, and for remote configuration of the device.
- Applications with the SVS-2000N Process Visualization Software:
  This software provides an overview of the process variables in systems that are located on several different sites. From time to time, a PC dials up the systems and updates the variables that are visualized in the software (such as reserves of consumables etc.).
- The measurement data which have been recorded by a paperless recorder can be downloaded by a PC via a modem link.

Now let’s look at some modem basics, taking the previously mentioned example (remote data transmission of measurement data from a paperless recorder) as a starting point.

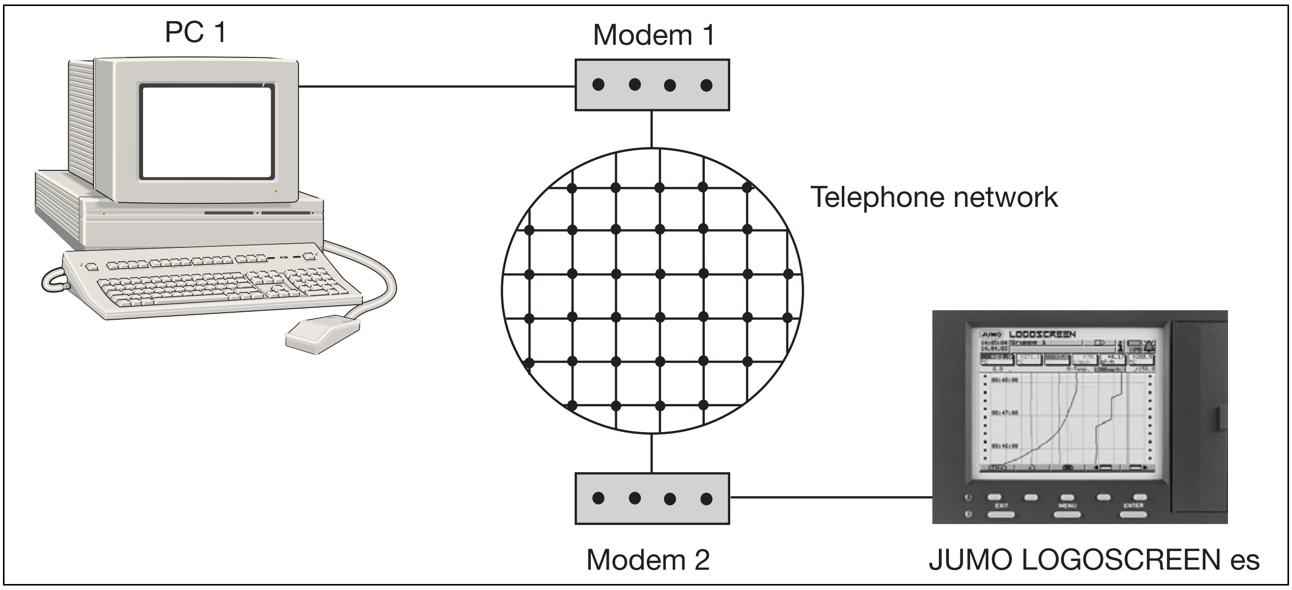


Fig. 72: Remote data transmission of measurement data from the paperless recorder JUMO LOGOSCREEN es

Basically, the link can be made through an analog or ISDN connection. The data transmission could also be made by radio telephony. Appropriate analog, ISDN and GSM modems are commercially available. **However, please take note of the remarks under “JUMO support for modem connections” at the end of this chapter.**

On the PC side, both external modems (RS232 interface, USB etc.) and internal modems (plug-in cards for ISA, PCI etc.) can be used. The appropriate Windows drivers must be used on the PC side. After successful installation, the modem will appear in a list of modems (in Device Manager). In the case of an external modem with an RS232 interface, a cable wired one-to-one is used. Such a cable is normally provided in the modem package as delivered. Likewise, in most cases the cable for connecting to the telephone socket (TAE socket) is also supplied.

5 Modems

On the device side (in this example, the side with the paperless recorder JUMO LOGOSCREEN es) the modem must have a serial interface for connection to the paperless recorder. If one device is attached, then the RS422 interface is adequate, but for more than one device the RS485/RS422 interface will be used.

5.1 Modem commands

The modem is controlled by a de-facto standard that was originally developed by the Hayes company in America, and then made use of by other manufacturers. So you will sometimes hear mention of “Hayes-compatible modems”.

The modem is capable of responding to a set of commands that can be transmitted across the serial interface. These commands always begin with the two letters AT (**At**tention). They are followed by the command codes for certain functions, e.g.

ATH0	= hang up
ATH1	= pick up
ATDP142	= dial number 142

Note
The V.25bis standard is also used. But the Hayes/AT command set is better known, and is recognized by almost all modems.

The connection between two modems is established by calling up the other station: the modem dials the number of the partner station, which then automatically receives the call. When modems with a higher transmission rate and automatic baud rate switching are used, both stations exchange various sound-frequency signals (which can be followed on the modem loudspeaker). The baud rate is negotiated in this way.

Before a modem can be used to make a connection, the modem must be configured. The modem on the PC side (Fig. 72) is only used for dialing into the telephone network. This can usually be done with the factory default configuration.

A configuration must be carried out for the device side modem (in this example, the side with the paperless recorder JUMO LOGOSCREEN es). To do this, the “device modem” is connected up to the serial interface of the PC. The configuration is made through a terminal emulator program, e.g. HyperTerminal (this is a standard accessory in Windows, and can be accessed via “Programs ➔ Accessories ➔ Communication”).

Before HyperTerminal can be used to set up a connection, the settings must be made for the data format (baud rate, number of stop bits, parity check). The modem then takes over these settings for its serial interface. The interface on the paperless recorder must have the same settings.

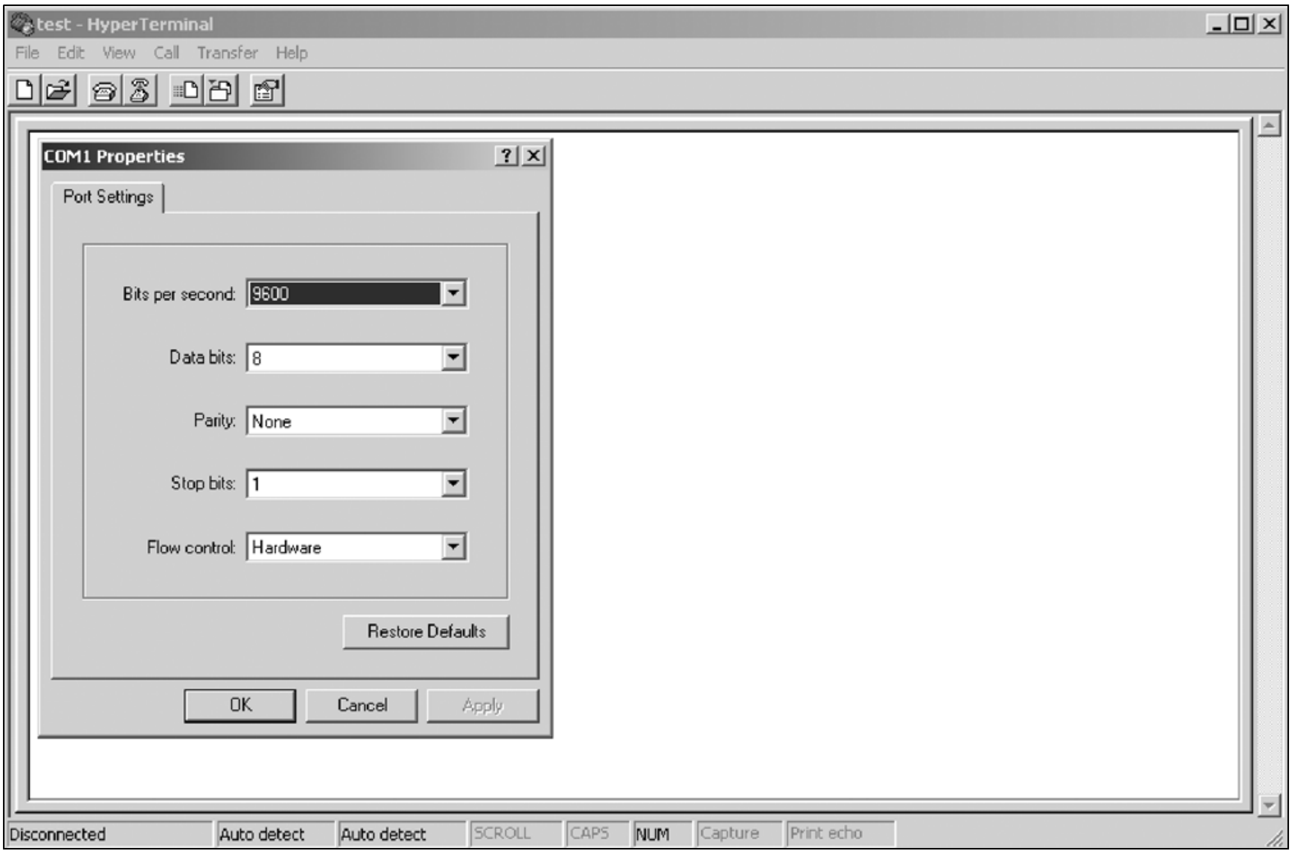


Fig. 73: Setting the baud rate and the data format with HyperTerminal

Now let's look at the AT commands that are required for the modem on the device side (these only apply to the modem type HSM 33.6).

AT&F

The modem is reset to the factory configuration.

AT&K0

If the computer or modem cannot handle the flow of data, then there is an option for halting the data stream

- Hardware handshake
  "Hardware handshake" is the term used to describe the control of the data flow by signal levels on special signaling wires. The modem sets the CTS signal to OFF, and only back to ON when it is ready to accept data. In this way, the data transfer from the computer to the modem can be stopped. The computer can use "Set RTS to OFF" to halt the flow of data from the modem.
- In the connecting cable between the paperless recorder and the modem, the above-mentioned signal pins (CTS and RTS) are not wired up. For this reason, AT&K0 means that the hardware handshake will be ignored.
  In this case – the transmission of measurement data – only short MODbus protocols are used.
  It is therefore unlikely that there will be an excessive quantity of data. However, if an overflow should nevertheless occur, this will be checked (e.g. by the higher-level parity check) and a repeat of the data will be requested, if necessary.

AT&D0

The modem will ignore the DTR signal.

ATS0=1

The modem has an internal register. This will also have an affect on the configuration. ATS0=1 means that the modem will pick up the incoming call after the first ring.

ATT

The tone dialing procedure is selected.

ATE0

The commands that are sent to the modem will not be returned (no “echo”). This ensures that in subsequent operation, the modem will just transfer the data, and will otherwise be “quiet”. This means that the modem does not put out any other data on the bus (RS4232 or RS485 interface).

ATQ1

Result codes OFF. After sending an AT command, e. g. via HyperTerminal, the modem will not answer with OK. This also ensures that in subsequent operation, the modem will just transfer the data, and will otherwise be “quiet”.

ATX0

The modem does not wait for a “free” signal, and will ignore “busy” signals.

AT&W0

The present configuration of the modem will be saved as Profile 0 (there is also a Profile 1).
The values are retained when the modem is switched off.

AT&Y0

The Profile 1 configuration will be loaded when the modem is switched on.

ATZ

The modem will be reset. The connection will be interrupted and – as mentioned above – the Profile 1 configuration will be loaded.

As already explained, the modems negotiate the baud rate before the data transmission. If the connection is of poor quality, it may be advantageous to fix the modem settings at a lower baud rate. To set a fixed baud rate, the following commands apply for the HSM 33.6.

ATB8 for 9600 bps
ATB11 for 19200 bps

Other AT commands will apply to make corresponding settings for other modem types.

If the baud rate is reduced on the receiving side, then the corresponding baud rate on the PC side should also be reduced to match (Fig. 73).

5 Modems

In the connecting cable between the modem and the paperless recorder, the pin assignments must be observed.

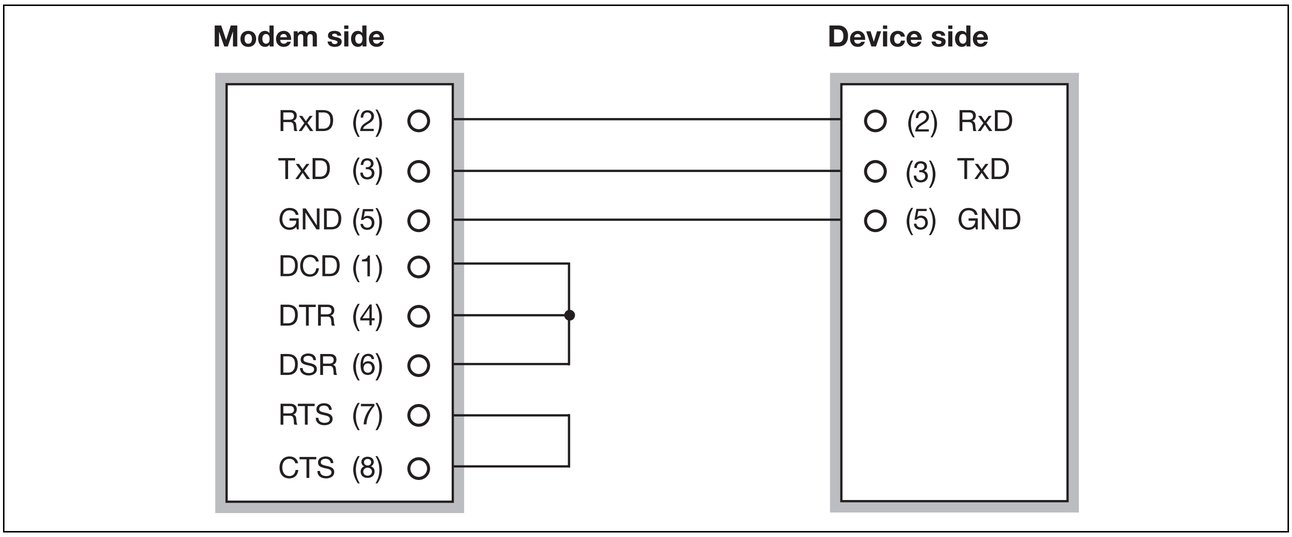


Fig. 74: Connecting cable: modem / paperless recorder when using an RS232 interface

Modem on the PC side

The modem on the PC side is only used for dialing into the telephone network. In most cases, the factory default settings can be accepted.

PCA Communications Software (PCC)

JUMO has developed a software for this example that, under time control, sets up a modem connection and downloads the measurement data that are stored in the paperless recorder. The modems recommended by JUMO can be configured (initialized) with PCC. The modems can be called up in the software and the required AT commands can be transferred. So it is not necessary to work with HyperTerminal.

JUMO support for modem connections

Not all AT commands are standardized. AT commands that can be used for one modem to establish a connection via the telephone network will not necessarily apply for a different type. JUMO has tested examples of analog, ISDN and GSM modems, and publishes the suitable AT commands for these types. On request, JUMO can also supply these modems already preconfigured.

JUMO will not provide support for any modems that were not tested and recommended by JUMO.

Informative material from JUMO -
for beginners and those with some practical experience

Know-how is not just needed to create JUMO products, but also for their later application.
That's why we offer several publications on aspects of measurement and control engineering for our users.

The publications are intended to provide step-by-step familiarization with a wide range of applications, for both beginners and those with some practical experience. They primarily illustrate general topics, with JUMO-specific applications to some extent.

In addition to JUMO Technical Literature and our new Software-Downloads we offer as well the possibility to order our brochures and CD-ROM catalogues online.

Electrical Temperature Measurement
with thermocouples
and resistance thermometers
Matthias Nau

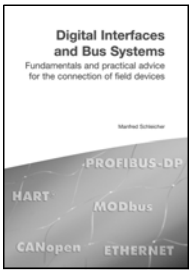
FAS 146
english version
Sales Article No.: 00085081
ISBN 3-935742-07-X
Price: 14,- EUR net

Control Engineering
A guide for beginners
Manfred Schleicher, Frank Blasinger

FAS 525
english version
Sales Article No.: 00323761
ISBN 3-935742-01-0
Price: 14,- EUR net

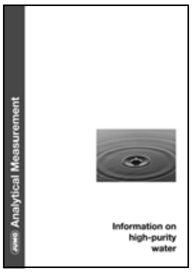
Explosion Protection in Europe
Electrical equipment
fundamentals, guidelines, standards
Jürgen Kuhlmei

FAS 547
english version
Sales Article No.: 000414312
ISBN 3-935742-10-X
Price: 9,- EUR net



Digitale Interfaces and Bus Systems
for Communication
Fundamentals and practical advice
for the connection of field devices
Manfred Schleicher, Frank Blasinger

FAS 603
english version
Sales Article No.: 00392023
ISBN 3-935742-03-7
Price: 14,- EUR net

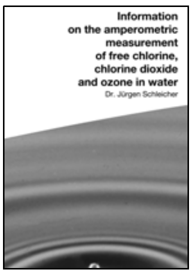


Information on high-purity water
Matthias Kremer, Reinhard Manns,
Dr. Peter John, Dr. Jürgen Schleicher

FAS 614
english version
Sales Article No.: 00403834
free of charge

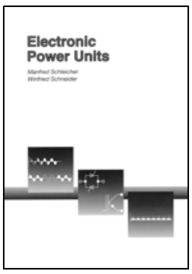
Informationen on
redox voltage measurement
Matthias Kremer, Ulrich Braun,
Dr. Peter John

FAS 615
english version
Sales Article No.: 00398237
free of charge



Informationen on the amperometric
measurement of free chlorine,
chlorine dioxide and ozone in water
Dr. Jürgen Schleicher

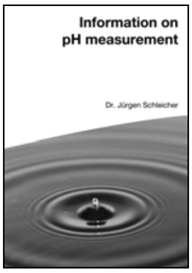
FAS 619
english version
Sales Article No.: 00398147
free of charge



Electronic Power Units
Manfred Schleicher, Winfried Schneider

FAS 620
english version
Sales Article No.: 00400481
ISBN 3-935742-05-3
Price: 9,- EUR net

Informative material from JUMO - for beginners and those with some practical experience

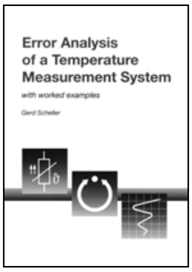


Information on pH measurement
Matthias Kremer, Dr. Peter John

FAS 622
english version
Sales Article No.: 00403233
free of charge

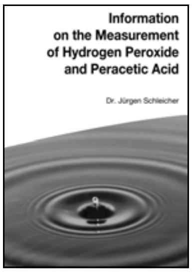
Information on Conductivity Measurement
Reinhard Manns, Dr. Peter John

FAS 624
english version
Sales Article No.: 00411341
free of charge



Error Analysis of a Temperature Measurement System with worked examples
Gerd Scheller

FAS 625
english version
Sales Article No.: 00415704
ISBN 3-935742-13-4
Price: 3,- EUR net



Information on the Measurement of Hydrogen Peroxide and Peracetic Acid
Dr. Jürgen Schleicher

FAS 628
english version
Sales Article No.: 00420697
free of charge

Please visit our German Website **www.jumo.net** (for Austria www.jumo.at, for the Switzerland www.jumo.ch) and assure yourselves of the wide variety of JUMO products for different application fields. Our website provides you with more details and information concerning the contact persons for your requirements, questions, iniciations and orders.

Catalogues on CD-ROM

Our catalogues are - apart from the printed versions - as well available in digital form. Our CD-ROMs with German and/or English data - contain structured catalogues (pdf.files), the JUMO Product Guide, „JUMO International Locations“ - the JUMO contact adresses all over the world - and as well the download of Acrobat Reader free of charge.

JUMO Products

english version
Sales Article No.: 00404116
free of charge

JUMO Produkte + Preise

german version
Sales Article No.: 00397668
free of charge

JUMO

www.jumo.net