# Digital Interfaces and Bus Systems for Communication

*Practical fundamentals*

*Manfred Schleicher*
*Frank Blasinger*

# Digital Interfaces and Bus Systems for Communication

*Practical fundamentals*

*Frank Blasinger*

*Manfred Schleicher*

# Preface

Digital communication confronts us every day in modern process engineering:

- it is used for configuring and setting the parameters for microprocessor instruments
- serial bus systems, with minimum wiring requirements, are able to acquire a large amount of decentralized information and distribute it to the process equipment. Intelligent field and automation devices can communicate directly with one another via a digital bus.

This book is intended as a step-by-step introduction to the subject of digital communications, for practical engineers and those new to this field. The emphasis is on clarifying generalized topics, as well as including some JUMO-specific applications.

In this revised edition, the material on bus systems has been extensively updated. The method of operation of bus systems for which JUMO has field devices available is explained in a practical manner.

Special thanks are due to all our colleagues, who helped us to prepare this book with their cooperation and professional input.

Fulda, March 2001

Manfred Schleicher Frank Blasinger

English translation of the 3rd. edition (revised)

M.K. JUCHHEIM GmbH & Co, Fulda

Part No. 00392023
Book No. FAS 603
Printed: 03.01
ISBN 3-935742-03-7

# Contents

# 1 Basic principles of digital interfaces and networks

This chapter deals first of all with some basic principles. The aim here is to achieve this without over-complex theoretical or mathematical treatment. Amongst other things, the basic facts about data encoding, types of data transmission, properties of different interfaces, construction of networks etc. are explained for practical engineers, who are increasingly faced with the subjects of digital communication and bus systems in modern automation engineering.

## 1.1 Analog/digital signals

In today's automation engineering, more and more devices operate digitally. This is in contrast with the more familiar analog measurement technology and data transmission. This means that digital process instruments are increasingly replacing analog type instruments in modern process control, part because of the technological advances and the advantages offered. Nowadays, digital transmission is even superseding the use of familiar standard signals such as 4 — 20mA, 0 — 10V, etc. for the transfer of analog measurements. The main features of different data transmission technologies are explained in more detail below.

**Analog signals**

A measurement, a temperature for example, is converted into a signal corresponding to this temperature by a measuring device. The signal could be, for instance, a 4 — 20mA current. Every temperature value corresponds clearly to a value of electrical current. If the temperature changes continuously, the analog signal also changes continuously. In other words, a characteristic feature of analog transmission is that the amplitude of the selected signal varies continuously over time (see Fig. 1).

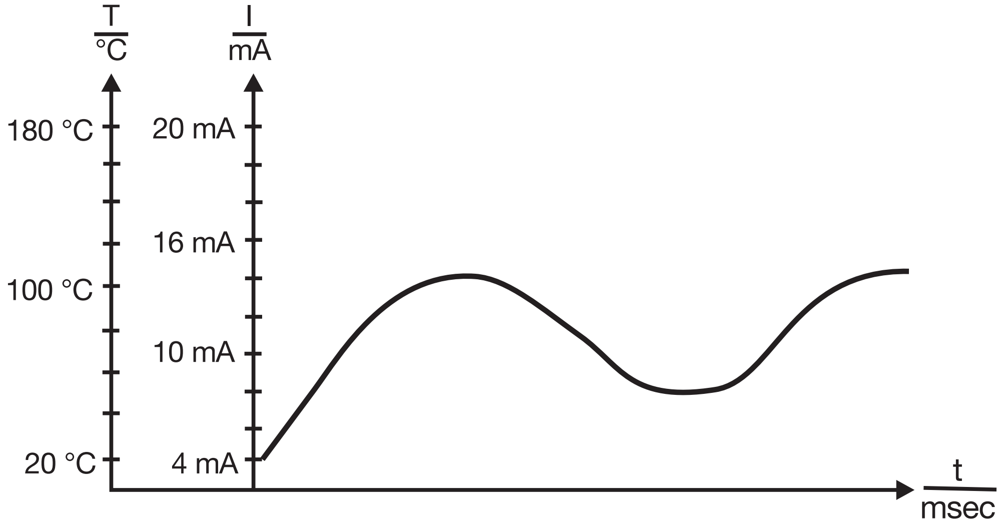


Fig. 1: Analog signal with continuously changing amplitude

In automation engineering, such standard signals (4 — 20 mA) are transmitted in pure analog form as a normalized current signal. A temperature value is measured by a Pt100 resistance thermometer, then converted into a current proportional to the measurement by a transmitter, and subsequently transmitted to a controller, indicator and recorder (see Fig. 2). By means of the current, every change in measurement value is immediately recorded by each instrument connected in the circuit.

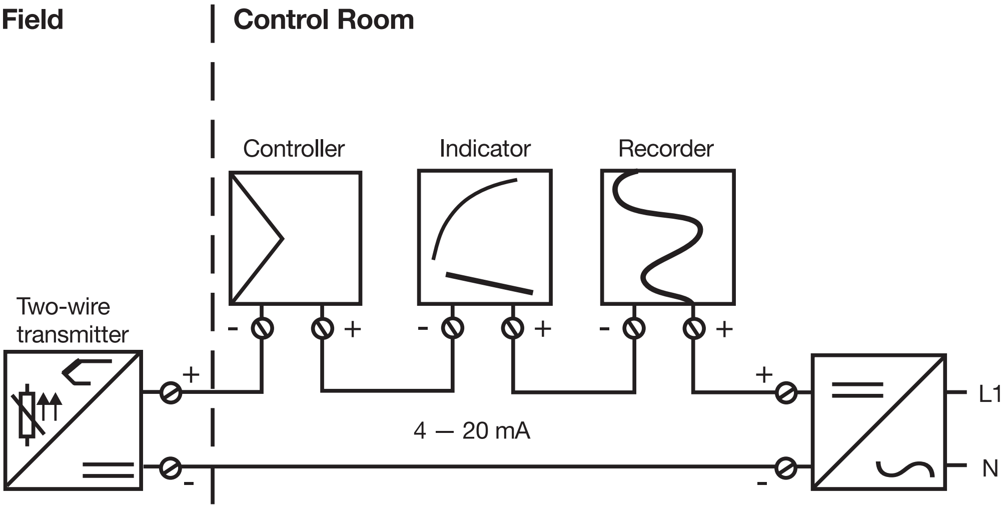


Fig. 2: Analog signal transmission

In measurement engineering, the information content of an analog signal is very limited in comparison with acoustic (sound) or optical (light) data transmission. Apart from the advantages of an unambiguous, continuously reproduced measurement, with simultaneous supply of power to the measurement recorder (e.g. two-wire transmitter), the information content of the analog signal consists only of the magnitude of the measurement, and whether or not the signal is available at the connected device.

**Digital signals**

The term “digital” is derived from the word “digit” and comes originally from the Latin “digitus = finger”. Digital means sudden or step changes, i.e. a digital signal does not vary continuously.

In the example of temperature measurement, this means that the analog measurement is divided into specific value bands, within which no intermediate values are possible. The values are read at fixed time intervals, the sampling time. The task of conversion is carried out by an analog to digital converter (or ADC). Here, the accuracy or resolution of the signal depends on the number of value bands and the sampling frequency.

In the example shown in Fig. 3, samples are taken every 20 msec, with a subdivision into 10 value bands.

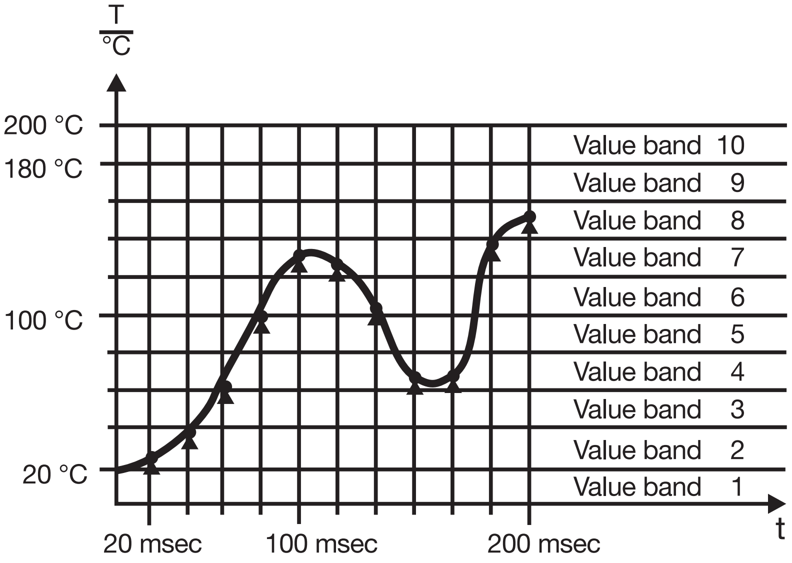


Fig. 3: Digitized measurement signal

The digitized quantity has only the two values “high = 1” and “low = 0” and must now be transmitted as a data packet by a microprocessor (µP)-transmitter with an interface (see Fig. 4). The measurement is transmitted encoded as a packet, and has to be decoded by the receiver (see also data encoding, Chapter 1.2). The transmission mode can vary: by different voltage levels, light pulses or a sequence of notes.

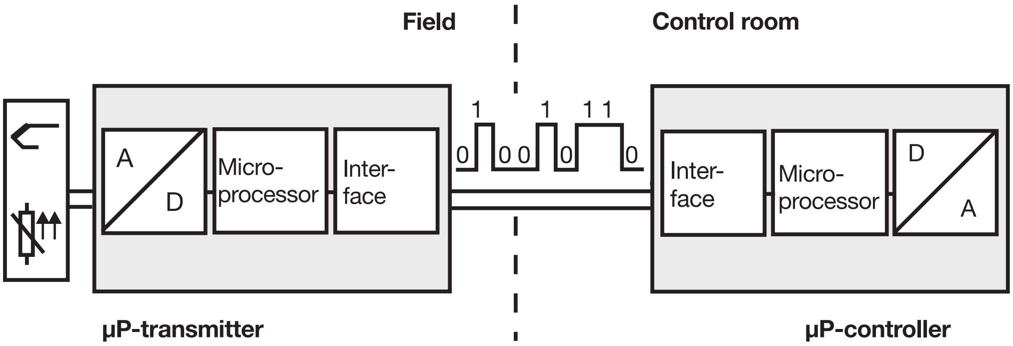


Fig. 4: Digital signal transmission

Digital data transmission has a number of advantages compared with conventional analog technology. As well as the actual measured value, the field device, with its integral microprocessor, can also transmit additional information (designation, dimensions, limit values, service interval etc.) to the automation system. Furthermore, data can be transmitted to the field device. The fact that several devices can communicate with the automation system over one cable results in a reduction in materials and less expenditure on installation, hence reducing the overall costs (see Fig. 5).

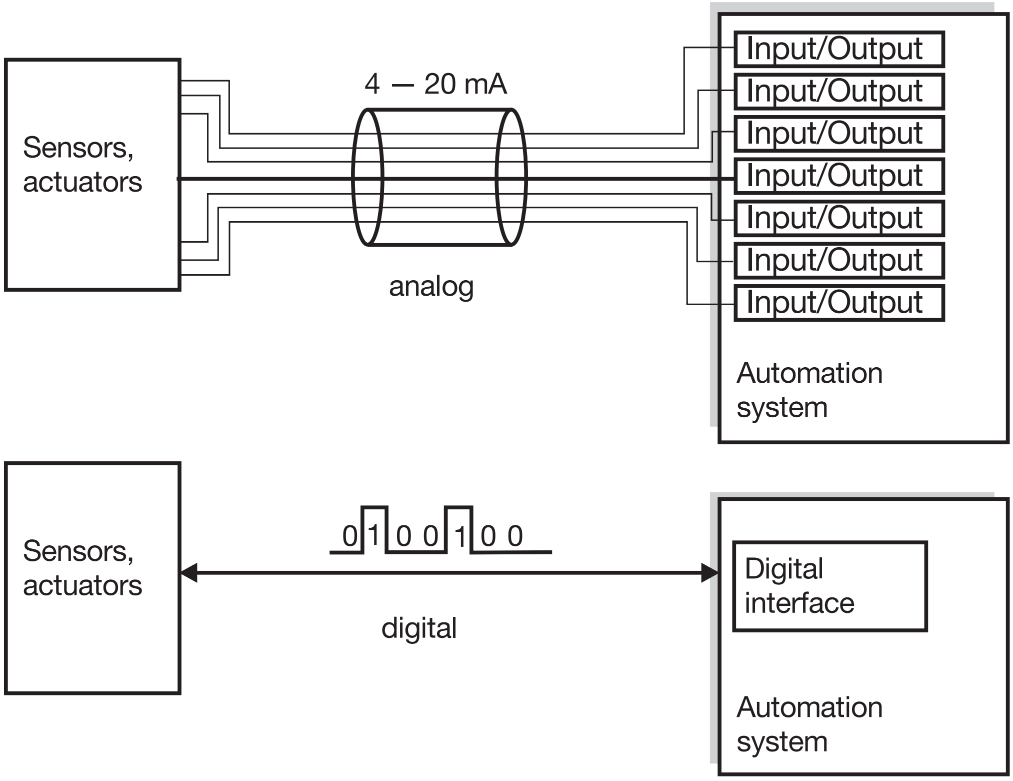


Fig. 5: Analog and digital cabling

One disadvantage of the conventional technology using 4 — 20 mA signals with µP-devices is the unnecessary D/A conversion. A digitized value available in the microprocessor must be converted to an analog current signal and then digitized once again in the automation device for further processing.

## 1.2 Data encoding

We have already noted that some form of encoding into electrical signals is required first of all for data transmission. With analog transmission, the information is transmitted by means of the amplitude. Digital technology has only two states, “On = logic 1” and “Off = logic 0”, which are usually transmitted by means of different voltage levels. Various codes and protocols, which must be understood by all communication partners in the data system, are used for transmission of digital quantities.

The most common codes and protocols are investigated in more detail in this chapter and later on in the book.

**Bit and byte**

**Bit**

The bit is the unit for a binary (two value) signal, corresponding to a single digital data unit that has the value “1” or “0”, The term bit, an abbreviation for binary digit, is very commonly used as the smallest unit in information technology.

As already described, the signals logic “1” and logic “0” are usually represented by voltage signals with different levels (see Fig. 6). The voltage levels used depend on the type of interface employed.

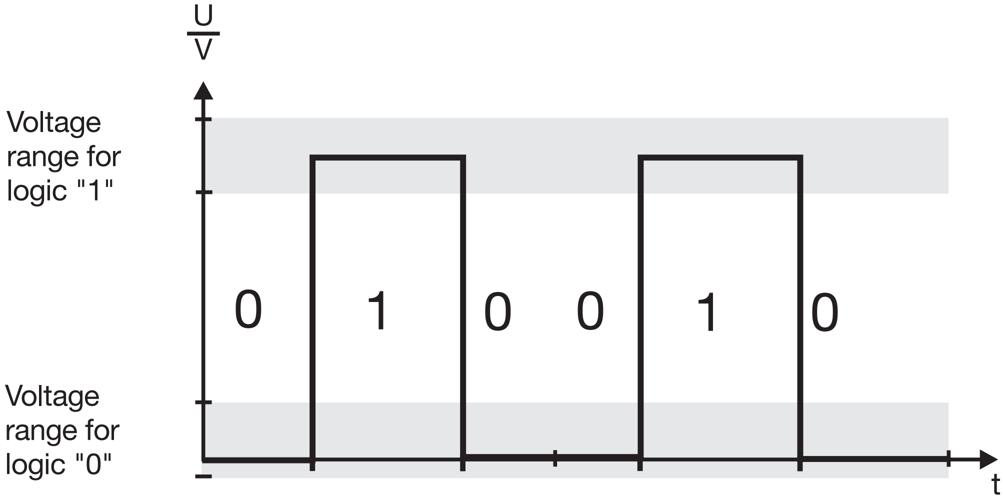


Fig. 6: Binary data transmission in “bits” using different voltage levels

**Byte**

The term byte was introduced for a unit with 8 binary characters. A byte has a length of 8 bits (see Fig. 7). In an automation device (PLC), for example, the signal statuses of 8 logic inputs or outputs are combined into either an input byte or an output byte. Correspondingly, for example, 64 bits = 8 bytes, 72 bits = 9 bytes etc.

Larger units, which will be encountered when dealing with computers, are the kilobyte (kB) = 1024 bytes and the megabyte (MB) = 1024 kB.

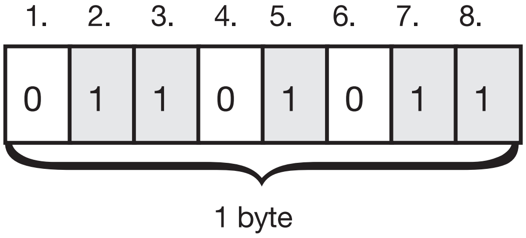


Fig. 7: Combination of “8 bits” to give “1 byte“

**Word**

A sequence or string of binary characters that is regarded as a unit in a particular context is designated as a word. A control instruction in a PLC or a command in a communications protocol has, for example, 1 word, 2 bytes or 16 bits. In many automation devices, 16 logic inputs or outputs are combined to give either an input word or an output word (see Fig. 8).

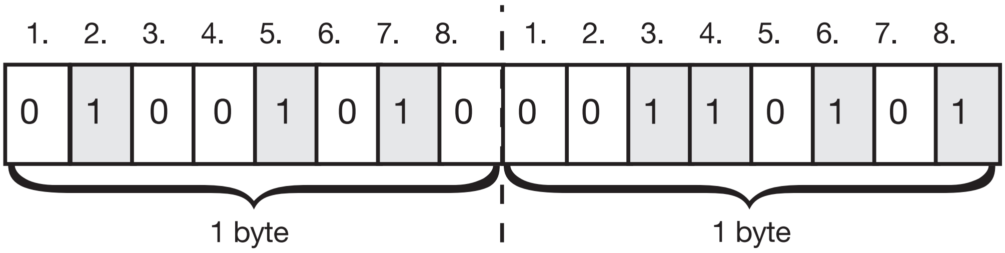


Fig. 8: Representation of a data word using “2 bytes“

**Double word**

A double word has 2 words, 4 bytes or 32 bits.

**Binary system**

The best known and most important number system is the binary number system, usually referred to as the binary system. Each position in a binary number is assigned a power of 2. Fig. 9 shows the basic structure of the system. If a place value is zero, a “0” is entered instead of a “1”.

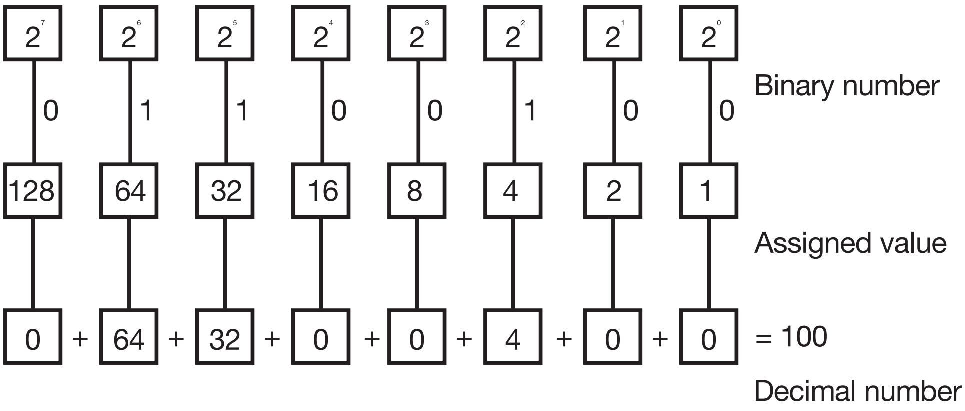


Fig. 9: Representation of decimal numbers in the binary system

**BCD code (8-4-2-1 code)**

The name BCD (binary-coded decimal) means a decimal number expressed in binary code. The main aim of this frequently used code is to represent the decimal numbers by 0 and 1, using the binary system. In this case, the highest value decimal number is 9. To express the number 9 in binary code requires 4 power of 2 places in all (see Table 1).

| Decimal number | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
|---|---|---|---|---|
| | 8 | 4 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |

Table 1: Representation of the decimal numbers up to 9 in BCD code

Four binary places (also referred to as a tetrad) are required to represent the highest decimal number. The BCD code is thus a 4 bit code. We can see that it very easy to transmit a decimal number to an automation device, a PLC for example, by expressing the number as a bit muster in 8-4-2-1 code.

**ASCII code**

However, not only numbers have to be transmitted, but also letters, punctuation marks etc. For this we use coding tables in which each number is assigned a letter, number or symbol. The ASCII code (American Standard Code for Information Interchange) is very widely used in this connection. In addition to characters and numbers, special characters and control characters are also defined here (see Table 2).

| ASCII value | Charac-ter | Control charac-ter | ASCII value | Charac-ter | ASCII value | Charac-ter | ASCII value | Charac-ter |
|---|---|---|---|---|---|---|---|---|
| 000 | null | 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 | (Home) | VT | 043 | + | 075 | K | 107 | k |
| 012 | (Form 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 | ◄ | DC1 | 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 | (Cursor right) | FS | 060 | < | 092 | \ | 124 | ¦ |
| 029 | (Cursor left) | GS | 061 | = | 093 | ] | 125 | ⌇ |
| 030 | (Cursor up) | RS | 062 | > | 094 | ↑ | 126 | ~ |
| 031 | (Cursor down) | US | 063 | ? | 095 | – | 127 | △ |

Table 2: ASCII code

It is noticeable that the ASCII standard only caters for 128 characters, and these can be represented by using 7 bits ($2^7 = 128$). Bear in mind that the zero counts as a number, hence 128 characters. The eighth bit has no significance in the ASCII code; originally it was assigned as a parity bit (see forms of data transmission, Chapter 1.3).

In this code, the characters from decimal 0 to 31 are control codes, from 32 to 63 numbers and from 64 to 127 mainly upper and lower case letters. However, with 128 characters, some special characters such as umlauts, “§” etc. cannot be represented. For this reason, the code is extended to 8 bits, and can then represent a total of 256 characters ($2^8 = 256$). Today, the extended 8 bit form is used almost exclusively.

Here is a short example of transmission of a character using the ASCII code. The letter “J” is to be transmitted to an automation device. This takes place as follows:

- Conversion into ASCII code:
  “J” = 74
- Conversion into a bit muster and transmission:
  74 = 1 0 0 1 0 1 0
- Decoding the bit muster in the receiving device:
  $$\begin{aligned}1\,0\,0\,1\,0\,1\,0 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = \\ &= 64 + 0 + 0 + 8 + 0 + 2 + 0 = 74\end{aligned}$$
- Conversion using ASCII code:
  74 = “J”

This may all appear to be very time-consuming, but modern microprocessor devices can carry out the encoding and decoding in an extremely short time.

There are other codes as well as the ASCII code, for example IEEE 754 (Standard for Binary Floating-Point Arithmetic) issued by ANSI (American National Standard Institute) in 1985. It deals with the representation of signs, exponents and mantissas of a floating-point number.

**Hexadecimal system**

In computers, each binary number is expressed in groups of four bits, giving rise to very long bit strings. For this reason another system is used, the hexadecimal system, also called hex code. Place values use the powers of 16. Including the zero, 16 numbers are required altogether. The decimal system is used for the numbers 0 to 9, and the letters A, B, C, D, E and F are used for the numbers 10 to 15 (see Table 3).

| Decimal | Hexadecimal | Binary |
|---|---|---|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0002 | 0002 |
| 3 | 0003 | 0011 |
| 4 | 0004 | 0100 |
| 5 | 0005 | 0101 |
| 6 | 0006 | 0110 |
| 7 | 0007 | 0111 |
| 8 | 0008 | 1000 |
| 9 | 0009 | 1001 |
| 10 | 000A | 1010 |
| 11 | 000B | 1011 |
| 12 | 000C | 1100 |
| 13 | 000D | 1101 |
| 14 | 000E | 1110 |
| 15 | 000F | 1111 |
| 16 | 0010 | 10000 |

Table 3: Comparison of decimal, hexadecimal and binary numbers

We can see that the carry occurs at decimal 16 = 0010 hex. When converting a hexadecimal number to a binary number, each hex. number corresponds to a four bit binary number with the equivalent value, and vice versa. A microprocessor would interpret a binary number that it receives as follows:

| | | | | |
|---|---|---|---|---|
| Binary | 0001 | 1011 | 1111 | 1100 |
| Hex | 1 | B | F | C |

## 1.3 Types of data transmission

Having explained the principles of data encoding, we now turn to the question of how the individual data are transmitted.

**Parallel transmission**

The most obvious solution is to transmit the data in bytes over a minimum of 8 lines, so that each line is assigned a switching state or bit. This is referred to as parallel transmission or parallel interface (see Fig. 10). With the parallel interface, the 8 data bits that, for example, are transmitted from the computer to the printer for each letter, are transmitted simultaneously over 8 lines. The Centronics and IEEE 488 interfaces are familiar versions here.

Parallel data traffic permits a high transfer rate, as each block of 8 bits (and even 16 or 32 bits with more advanced systems) is transmitted simultaneously. The disadvantages are the high cabling costs and the susceptibility of parallel transmission to interference as the distance increases. For this reason, the parallel interface is only used over short distances.

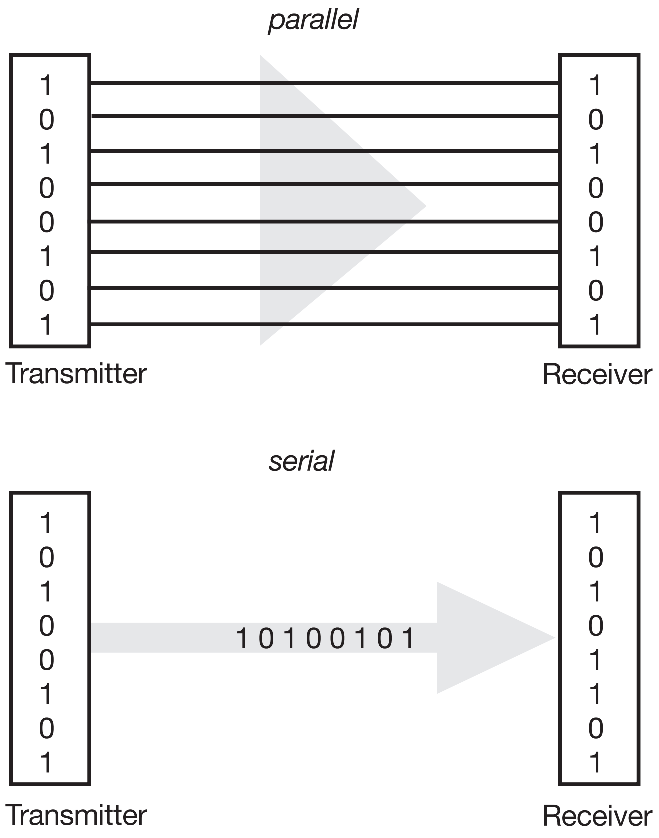


Fig. 10: Parallel and serial data transmission

**Serial transmission**

It is advisable to use serial data transmission (interface) for longer routes and because of its reduced susceptibility to interference (see Fig. 10). Here the 8 data bits are sent one after the other, e.g. over a two-wire circuit. It is significantly slower than the parallel method, but has been accepted in automation engineering because of its suitability for bus systems. The RS485 interface is well known here, and will be described in more detail later in the book.

The electronic circuitry of a suitable interface has the task of transmitting the digital information via a medium. One method is the D.C. current / D.C. voltage pulse method, the outstanding feature of which is the very simple transmitter and receiver construction. Information can be transferred by:

- switching a current on and off
- switching a voltage on and off
- switching a voltage from a negative to a positive potential
- changing the sign of a voltage difference (see Fig. 11).

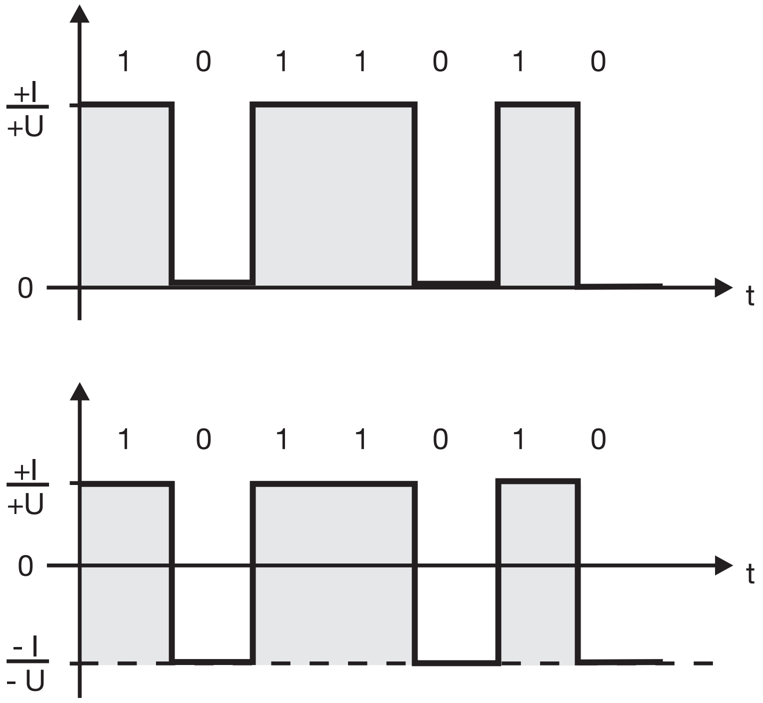


Fig. 11: D.C. current and voltage pulse methods

There is also the familiar A.C. methods, predominantly used for data transmission in the telephone network. The signals are normally transmitted as:

- amplitude modulated (AM) signals
- frequency modulated (FM) signals
- phase modulated alternating current (PM) signals (see Fig. 12).

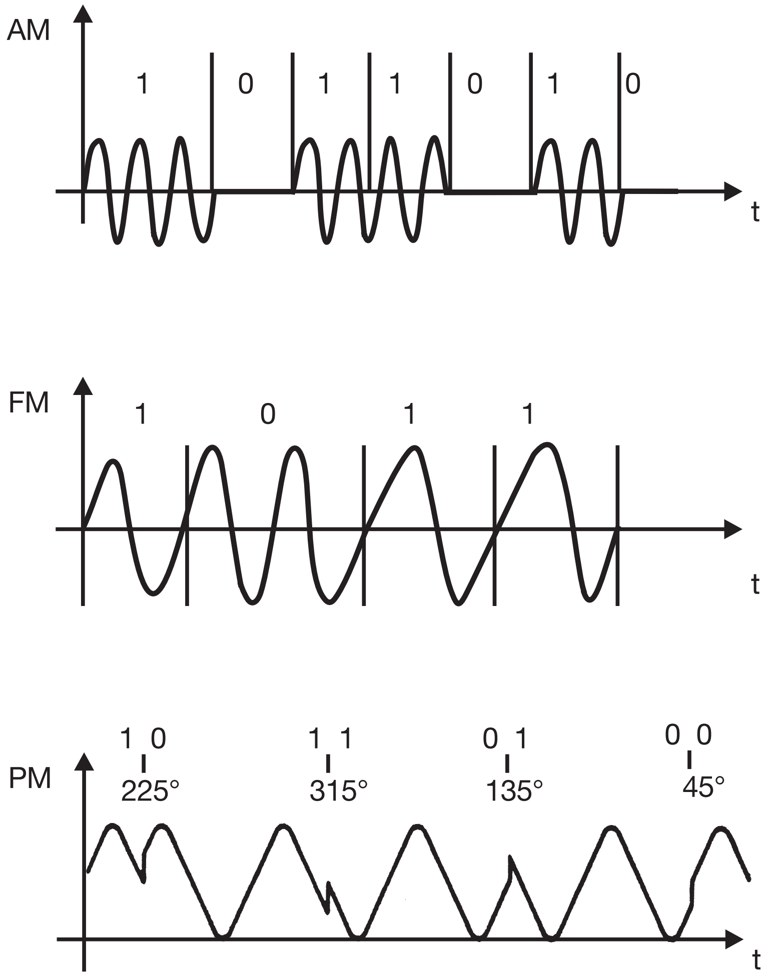


Fig. 12: A.C. current pulse methods

**Pulse code modulation**

Digital data transmission is becoming increasingly important in modern data transmission. With pulse code modulation, no analog signals are used, and only digital signals are transmitted. Amplifier noise suppression is simplified using this method. The technique is widely used to improve quality in home electronics, e.g. with the CD (compact disk).

Where telephone channels are transmitted over digital networks, this method is used in the ISDN (Integrated Services Digital Network) for voice transmission etc.

When data is transmitted, it must be ensured that the receiving device records a message at the same time rate as it was sent out by the transmitting device. This procedure is known as synchronization of the communication partners, and means that the internal clocks must be synchronized with one another. In practice, two distinct methods are used: **synchronous and asynchronous transmission**. The difference between the two methods is, that with synchronous data transmission the two subscribers or nodes are synchronized throughout the entire operation, whereas with asynchronous transmission they are only synchronized for a short period of time.

### Asynchronous transmission

With the asynchronous method, each transmitted byte or code word is enclosed between two synchronization characters. Each character transmitted begins with a “start bit” followed by the actual information (one or even several bytes). The termination is made up of the “parity bit” test character and one or two “stop bits” (see Fig. 13).

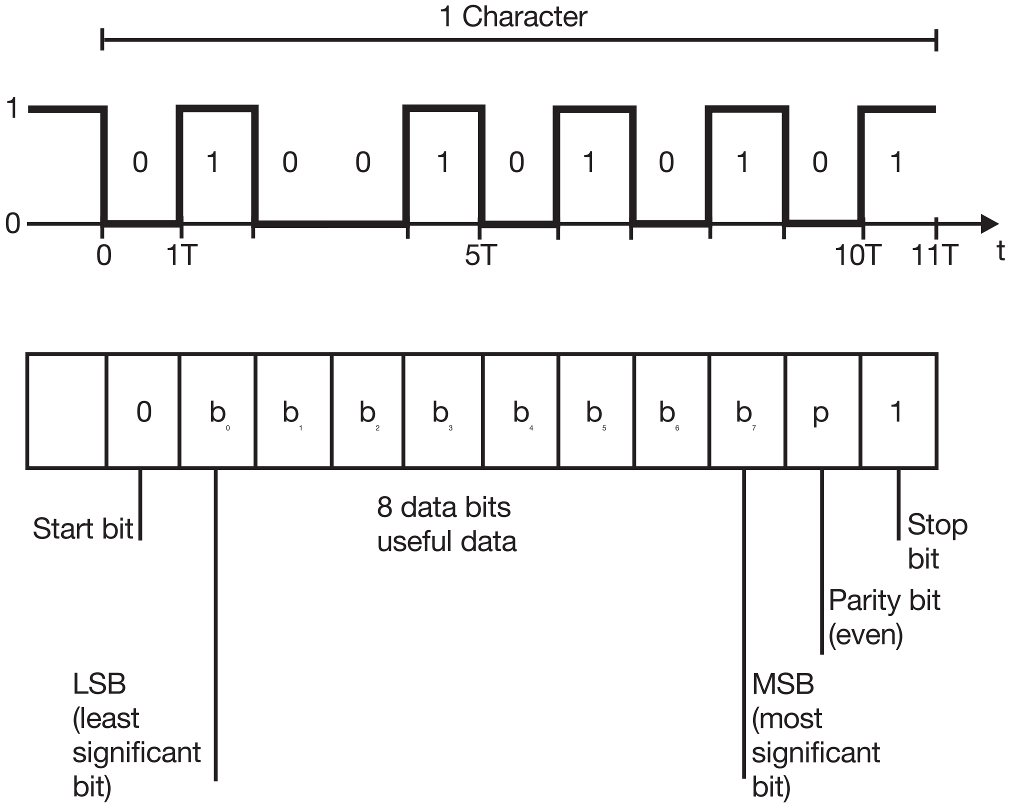


Fig. 13: Composition of a telegram with asynchronous data transmission

The task of the start bit is to establish synchronism between the transmitter and the receiver. The start bit is issued by the transmitter and indicates that a character of data now follows. The data line is held in a mark or “1” condition when no data is being transmitted, and is set to the space or “0” condition by the start bit.

The “stop bit” reports that the transmission of the data is complete and resets the transmission medium from “0” to “1”. Depending on the system used, 1 or 2 stop bits are possible.

In other words, this means that for transmission of a character, the receiver synchronizes its clock with the start bit. To a large extent, the transmitter and receiver are in synchronism for the duration of the transmission; synchronism is lost again after the stop bit. As the clocks only run in synchronism for a short time, this is referred to as asynchronous data transmission.

A data transmission does not take place without interference. For this reason, some method of checking a received character for correctness is required (see Chapter 1.8.3 “Error checking”). The simplest method uses a parity bit for this, set at the transmitter, and dependent on the 7 or 8 data bits. With even parity checking, the parity bit is set so that checksum of the data and parity bits produces an even number, and with odd parity checking an odd number. For its part, the receiver forms the checksum of the digits of the character received and compares it with the status of the accompanying parity bits. The example in Fig. 13 illustrates even parity checking, as the sum of the digits is four with the parity bit set at 0. Error correction is clearly not possible with this method. However, there are rather more complicated methods which do allow error correction.

The advantages of this type of transmission lie in the low hardware costs and a simple, easily understood data protocol. It is particularly suitable for short messages and is often used for communication in fieldbus systems. The disadvantage is the additional bus loading, caused by the constantly recurring start and stop bits.

**Synchronous transmission**

With synchronous transmission, a timing signal (character) is generated by the transmitter and used for control of the transmission. The receivers are synchronized with one another by means of this timing signal, i.e. the point at which they transfer data is determined. This character must be repeated at regular intervals (every 100 to 1024 data bits).

The advantage of synchronous transmission is that a large data block can be transferred within a short time window. Disturbances on the transmission medium can be a disadvantage, and can lead to loss of synchronism. Another disadvantage is the higher hardware cost of the transmitter and receiver.

### 1.3.1 Operating modes of a transmission medium

The direction in which data can be transmitted between various subscribers or nodes is of considerable importance. In general terms, three distinct operating modes for data traffic can be identified (see Fig. 14).

**Simplex operation**

Transmit or receive operation. Data transfer is possible in one direction only. This operating mode is rarely used in data processing, as the receiver has no possibility of replying. Good examples are radio and television transmitters, which all operate in simplex mode. The transmitter sends out a program, irrespective of whether it is received or not (see Fig. 14a).

**Half-duplex operation**

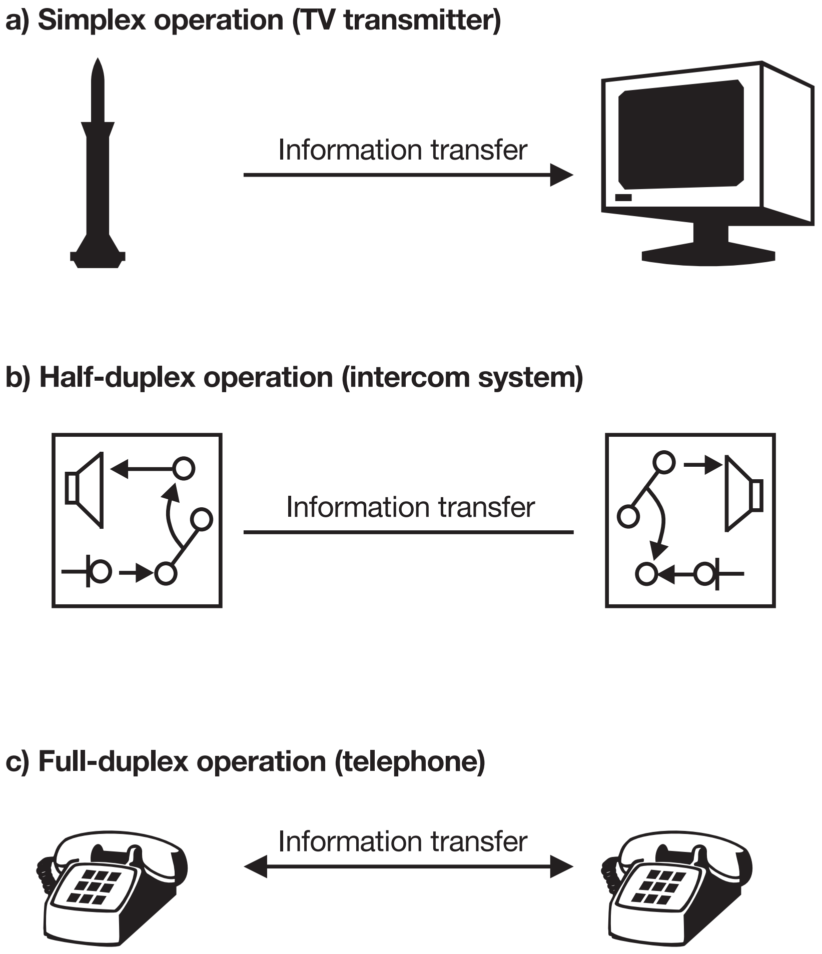
Alternate operation. Data transfer is possible in both directions, not simultaneously however, but alternately, one after the other. This operating mode is most often used in data processing and fieldbus systems. Examples are: intercom systems, CB radio or remote scanning via telefax (see Fig. 14b).

This mode is also known as bidirectional (in both directions). JUMO instruments fitted with an interface can operate bidirectionally in half-duplex mode, i.e. they can transmit and receive data, but cannot do both at the same time.

**Full-duplex operation**

Two-way operation. Simultaneous transmission in both directions is possible. This transmission mode will find increasing application in the future, in modern data processing networks. Telephoning is an example of this mode, where it is possible to speak and hear at the same time (see Fig. 14c).

Full-duplex operation may not necessarily be possible, even if separate send and receive lines are available to interconnect two automation devices. This is because the hardware and software devices are unable to send and receive data simultaneously. Separate lines are more likely to be used, to avoid the cost of the switching equipment required to change over from transmit to receive mode.


**a) Simplex operation (TV transmitter)**

**b) Half-duplex operation (intercom system)**

**c) Full-duplex operation (telephone)**

Fig. 14: Data transmission operating modes

### 1.3.2 Speed of data transmission

Another criterion for data transmission is the speed (baud rate). By this is meant the number of binary signals transmitted per unit time i.e. how many bits per second (bit/sec or bps) are transmitted between transmitter and receiver. To be able to understand one another, all automation devices connected together must communicate at the same speed.

**Baud**

DIN 66 348 defines speeds of 110 — 19200 bit/sec for serial transmission of measurement data in a point-to-point network. The terms baud and baud rate are often used. A baud is defined as one pulse or bit per second, named after the French engineer, Baudot.

The speed of transmission cannot be indefinitely high, as there are physical limits. For example, the longer the transmission line is, the higher is the probability of distortions and disturbances, and hence the lower the possible baud rate. Today, depending on the fieldbus technology employed and the standard used, normal speeds range from 1200 bps to 500 kbps. Speeds of over 1 Mbps can be achieved with special transmission technologies.

## 1.4 Media for data transmission

The transmission medium or cable itself represents a further important aspect, as transmission speeds are continually increasing. The correct cable is vital in industrial data transmission too, particularly with regard to interference. The main differences between the various transmission media are the transmission speed, the material costs and the ease of laying.

Primarily, the medium used depends on the protocol employed. This determines the physical characteristics of the interface and with it the transmission medium that can be used. In the same way, the possible cable length is specified, which depends mainly on the interface's susceptibility to interference, the baud rate, and the resistance and capacitance of the cable. For practical applications, the following media are presently available.

**Shielded and unshielded twisted-pair cable**

**Unshielded twisted-pair cable**

**Unshielded twisted-pair** cable (UTP) offers the most cost-effective solution with the easiest installation. It is relatively well-known from telephone technology. However, it is not widely used as a transmission medium, because of the low transmission speed and limited transmission distance achievable. In addition, this type of cable is susceptible to EMC interference.

Using this transmission medium, a range of 100 – 200m at a speed of up to 167 kbps can be attained.

**Shielded twisted-pair cable**

**Shielded twisted-pair** cable (STP) is a better alternative. This is a cable with two individually shielded twisted-pairs and an overall outer shield or screen. Depending on the physical interface, twisted-pair cables with multiple cores are also used.

In today's automation technology, the shielded twisted-pair cable is the most widely used transmission medium. The range is from 1000 — 3000m at a transmission speed of over 1Mbps.

**Coaxial cable**

A number of messages can be transmitted simultaneously with a coaxial cable. The main area of use for this transmission medium is in computer networks, e.g. Ethernet in office communications. It is rarely used at field level, as it is more expensive than the twisted-pair cable, and also more difficult to install. The advantages of this type of cable are the high transmission speed and good noise immunity.

The range can be up to several thousand meters, but the planned length of each bus segment should not normally exceed about 500m, because of the specific attenuation. The transmission capacity is up to 300Mbps.

**Optical fiber waveguide**

The optical fiber waveguide or fiber-optic cable is tremendously efficient and is becoming more and more significant in modern data transmission. Pulses of light, rather than electrical pulses, are transmitted through this cable. Hence this cable is immune to electromagnetic interference.

A fiber-optic cable consists of a thin glass or plastic fiber core, surrounded by various protective layers for reasons of stability. There are two different types, the multi-mode and the mono-mode fiber (see Fig. 15). With the mono-mode fiber, there is one beam of light which is guided parallel to the axis, so that the signal appears almost unchanged at the end of the cable. The multi-mode fibre operates in a different way. The light splits into a number of beams that travel through the cable by reflection at the boundary. This gives rise to transit-time differences and the signals appear somewhat more dispersed at the end of the cable. Because of this, the bandwidth available for signal transmission is rather limited with this type of cable.

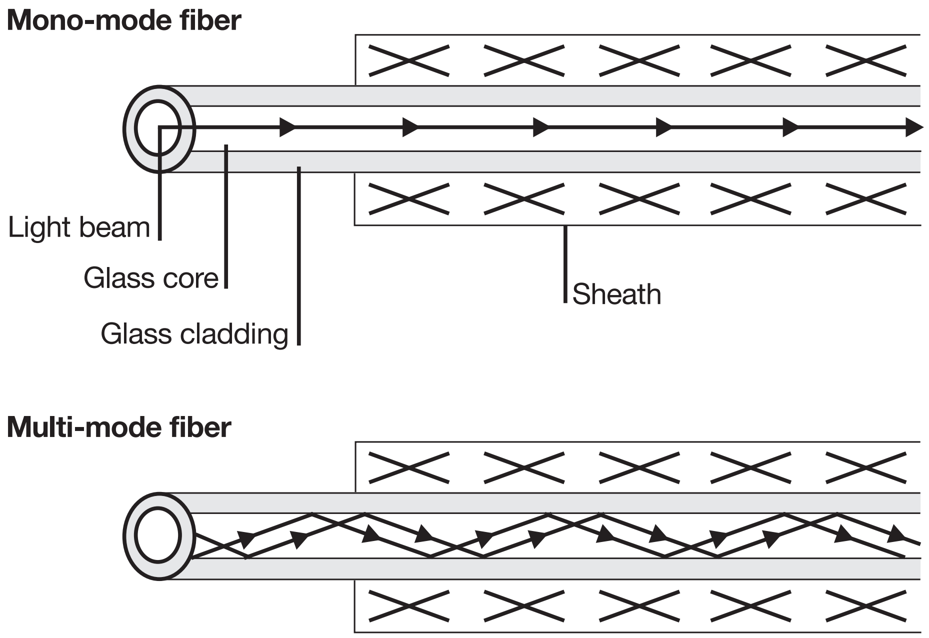


Fig. 15 Principle of an optical fiber waveguide

One small disadvantage of this technology is that the application is limited to point-to-point connections.

Because light represents a form of energy, in exactly the same way as does an electric current, it too cannot be transmitted without losses. Ranges of up to several kilometers can be achieved with real glass fibers, but amplifiers must be fitted every 2 – 3 km. Transmission speeds of the order of gigabit/sec are possible with this medium.

Development of this technology is advancing all the time, and an increasing number of networks are being equipped with fiber-optic cable. The cost of this technology will reduce further in the future, so that this medium will be used more and more widely for field cabling in automation engineering. It has already attracted a great deal of interest for applications in hazardous (Ex) areas.

### 1.4.1 Transmission quality and cable terminating resistance

The quality of transmission relative to the cable length depends on a number of factors:

**Noise susceptibility**

- Noise susceptibility of the physical interface type used.

**Noise effects**

- Effects of noise on the cable and their avoidance through the use of shielded twisted-pair cable etc. When a shielded bus cable is used, it is recommended that the screen is connected to the protective earth at both ends of the cable, using a low inductance connection, in order to achieve the best possible electromagnetic compatibility (EMC). Of course, the installation of interface cables should be sensibly planned, so that they do not run in close proximity to contactors, thyristor controllers, inverters, motors, transformers, or parallel to A.C. power cables or D.C. cables that are used for switching inductive loads, etc.

**Transmission speed**

- Transmission speed (baud rate). High transmission rates demand an optimally installed transmission medium and, for the most part, shorter transmission paths.

**Capacitance**

- Capacitance of the cable. The cable is not a valueless medium, whose electrical properties can be neglected. It is a passive four-terminal network with constant electrical values (series resistance R, inductance L, cross resistance G, capacitance C). In other words, with these values, the cable represents a lowpass filter, and the higher the transmitted frequency, the more the transmitted signal is distorted.
  For this reason, the characteristic values for the bus cable, as laid down by the standards or specifications, should always be observed (see, for example, DIN 19 245 Part 1, Para. 3.1.2.3).

**Cable termination**

Another important point for data transmission is the cable termination. At higher transmission speeds (< 200 kbps), the use of a cable terminating resistor is recommended, in order to retain signal rise times and to keep reflections as small as possible. The terminating resistor “Rt” should be as nearly as possible equal to the characteristic impedance of the conductor pair in the frequency band of the signal spectrum.

In some bus systems (see DIN 19 245 Part 1), in addition to the terminating resistor of the data lines, a pull-down resistor Rd is connected to the reference potential GND and a pull-up resistor connected to the supply voltage +VP (see Fig. 16). This measure ensures a defined quiescent potential on the cable, even when no subscriber is transmitting.

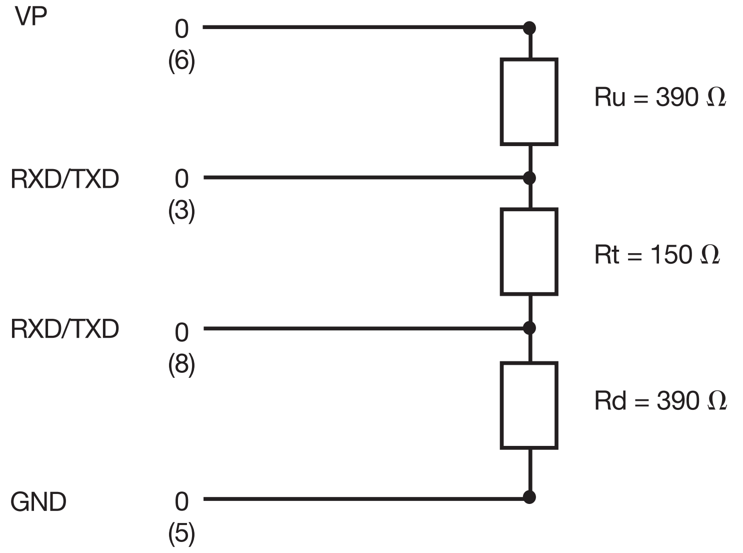


Fig. 16: EIA RS485 Standard cable termination to DIN 19 245 Part 1

### 1.4.2 Modem

The word "modem" is a combination of the terms MOdulator/DEModulator and describes a device that processes binary data for transmission over a telephone line. Consequently, modems allow data to be transmitted over substantial distances. The signals are forwarded over the telephone network, by radio or by satellite.

**Modem**

A modem receives data in digital form, for example from a computer, and converts this to an analog signal, so that it can be carried over the telephone line, demodulated at the other end by a second modem, and thus made available in digital form again (see Fig. 17).

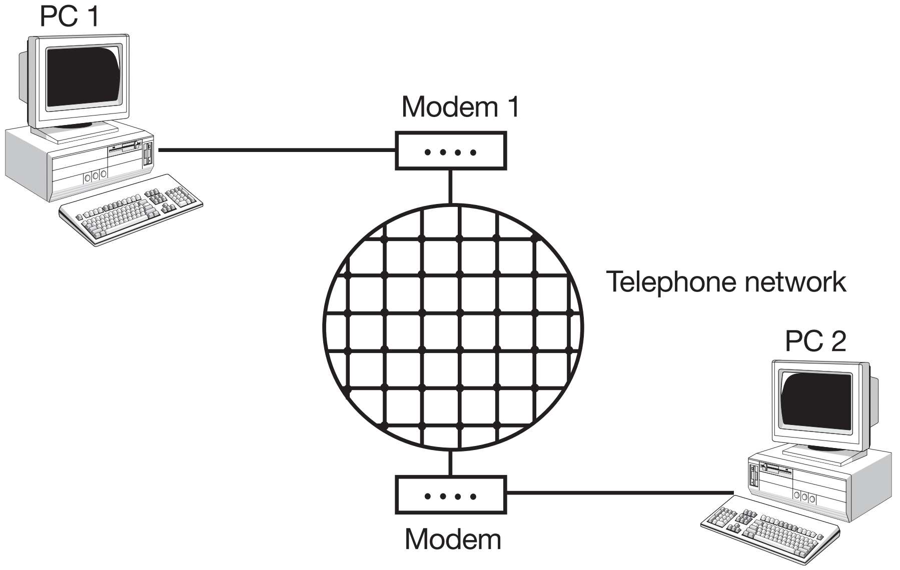


Fig. 17: Communication via a modem

# 1 Basic principles of digital interfaces and networks

As long as the majority of installed telephone connections are based on conventional analog technology and do not make use of the new ISDN network, then, for the time being, modems will retain their importance for data transmission. However, the ISDN (Integrated Services Digital Network) will increasingly replace the existing analog telephone network in Europe. It combines the use of speech, text, graphics and data in a single network. The difference between ISDN and the existing network is that, in ISDN, the signals are transmitted exclusively in digital form to the terminal device. The existing telephone lines can still be used for this digital transmision.

This means that, instead of using a modem to convert the digital signals of a PC to analog form, it is possible to use an ISDN controller card for direct data communication via ISDN. Normally, the connection takes place via the Basic-Rate Interface (BRI). Subscribers are given access to the S0 bus via an S0 interface (four-wire interface) for up to eight terminal devices, of which two can be used simultaneously. The bus has two 64 kbps user channels (B-channels) and one 16 kbps signalling channel (D-channel).

Other possible connections e.g. the primary multiplex connection (PMx) and the various communication possibilities are not discussed in detail here.

## 1.5 Properties of various interfaces

To establish communication in automation engineering, different devices from various manufactures have to be connected together. For successful data transfer, the correct cable must be chosen, and the interfaces of the devices must be identical. To this end, several interface standards have been established and specified in various standards.

Table 4 sets out a number of properties of some well-known interfaces.

Parallel interfaces, such as the IEC bus, also known as IEEE 488 or IEC 625, are not suitable for industrial use, because their noise susceptibility is too high and the cabling is cost-intensive. RS232 (V.24) and TTY (20 mA) interfaces are still very widely used for point-to-point connections. The physical properties of the RS422 and RS485 interfaces make them very robust, and they are now accepted as serial interfaces in the field area, because of their suitability for bus operation (see Chapter 1.6 “Networks and bus operation in automation”) and their high noise immunity. The suitability for bus operation reduces the cost of connection and cabling, and the noise immunity permits long cable lengths.

| | Serial interfaces | | | | Parallel interfaces |
|---|---|---|---|---|---|
| | TTY (20 mA) | RS232 (V24) | RS422 | RS485 | IEC<br>(IEEE 488) |
| Signal states logic 0/logic 1 | 20 mA/0 mA | +3V to/ -3V to<br>+15V / -15V | -5V / +5V<br>+5V / -5V | -5V / +5V<br>+5V / -5V | 5V / 0V |
| Possible transmission modes | asynchronous full-duplex | asynchronous full-duplex | asynchronous full-duplex | asynchronous half-duplex | asynchronous full-duplex |
| Maximum cable length | 1000 m | 30 m | 1200 m | 1200 m | 2 — 20 m |
| Number of lines | 4 | min. 3<br>2 data<br>1 ground | 4 (5)<br>2 transmitter<br>2 receiver<br>(1 ground) | 2 (3)<br>2 data<br>(1 ground) | 16<br>8 data<br>3 handshake<br>5 control |
| Subscribers per interface transmitters/ receivers | 1/1 | 1/1 | 1/10<br>without Repeater | 32<br>transmitters/ receivers without Repeater | 1/15 |
| Maximum transmission rate | 19.2 kbps | 19.2 kbps | 10 Mbps | 10 Mbps | 2 Mbps |
| Application | Teleprinters, displays, hazardous (Ex) areas, CNC machines | PC peripherals, automation engineering devices | PC peripherals, automation engineering devices | PC peripherals, fieldbuses in automation | Instruments and control devices for use in laboratories |
| Properties | insensitive to noise, suitable for bus operation, secure data transmission | insensitive to noise,<br>widely used | insensitive to noise,<br>suitable for bus operation | insensitive to noise,<br>suitable for bus operation,<br>widely used | susceptible to noise,<br>very fast |

Table 4: Properties of various interfaces

In principle, the type of interface has nothing to do with the method of data transmission, i.e. character-coding according to ASCII, the protocol used etc. Of course, the interface does determine whether bus operation, handshaking etc. are at all possible.

**UART**

Interface drivers are integrated circuits which process the digital signals into interface-specific signals. A V.24 driver, for instance, gives the signal the required maximum voltage levels of -15V and +15V etc. The processor output signals to the driver are always the same.

The interface driver itself is controlled by a “UART” (universal asynchronous receiver/transmitter) module (see Fig. 18). This converts the parallel data format of the processor into serial format, adds start and stop bits, calculates the parity bit etc. The data format is determined by the UART, the electrical format by the interface driver.

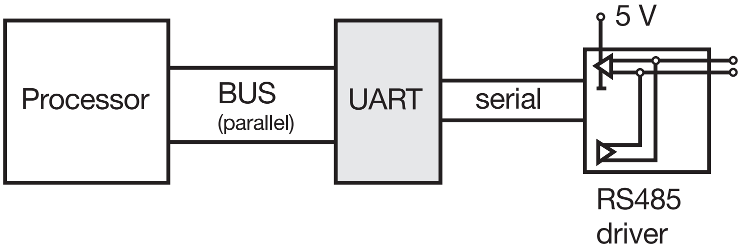


Abb. 18: Principle of interface processing into RS485 format

For this reason, DIN 66 258 Part 1, for example, refers to both TTY and V.24 in the same breath, although at first glance they appear completely different. However, if we set aside the handshake lines of the V.24 and ignore the different signal levels of the two interfaces, they are absolutely identical.

In contrast to TTY and V.24, the RS422 and RS485 interfaces are suitable for bus operation, requiring special software to manage the bus. Once again, the two interfaces are very similar to one another.

To check whether two devices can be connected via a digital interface, the information in the handbook should be used to establish whether the various parameters, e.g. the data format, of each device are identical:

- Electrical characteristics
  (determined automatically by the interface designation)
- Data format (baud rate, parity bit, stop bits, word length)
- Addresses
- Protocol, etc.

Although these are the essential conditions for a data transfer, they are not sufficient in themselves. The situation is comparable with two telephones, which have established a connection, after the above conditions are fulfilled. Whether the two callers can not only hear one another, but also understand one another, depends on whether they speak the same language. The data and bus protocols used must be identical and be correctly interpreted, and this is the area where the “software” takes over.

**Interface converters**

There are interface converters, some of them devices no bigger than a cigarette packet, which facilitate a conversion. For instance, if a device with a V.24 interface is required to transmit over a longer distance, or is to be connected to a number of devices, a conversion to RS422/RS485 can be carried out with this type of device. In the simplest type of interface converter, only the signal level is changed. As explained earlier, the data format, character-coding etc. are clearly not interface-specific (see Chapter 3.5.5 "Connection via an interface converter" ).

The most important data and properties of the serial interfaces (Table 4) used in today's automation engineering are described below.

**TTY interface**

There are a number of different designations for the TTY (20 mA) interface: line current, current loop, current interface. The TTY interface was originally developed for teleprinters ("TeleTYpes"). Its official name is the C_S interface. The design guidelines are laid down in DIN 66 258 Part 1.

A current is used for data transmission. A constant current, normally 20 mA, flows in the passive state.

As a result of this:

passive state (logic 0) = current "ON""

active state (logic 1) = current "OFF""

At first glance, the fact that the interface inverts may appear irrational.

Clearly, for a full-duplex operation, two current loops are required. When two devices with TTY interfaces are linked together, only one device provides the loop current, whilst the second device is "passive". Interconnecting two active devices can lead to malfunctions. In general, the current source is electrically isolated from the device.

As a general rule, in industrial versions of the TTY, the signals are coupled in and out using opto-couplers (see Fig. 19). A current source (often integral in the device) supplies the line current, normally defined as 20 mA. The open-circuit voltage of the current source must not exceed 24 V max. (see DIN 66 258 Part 1).

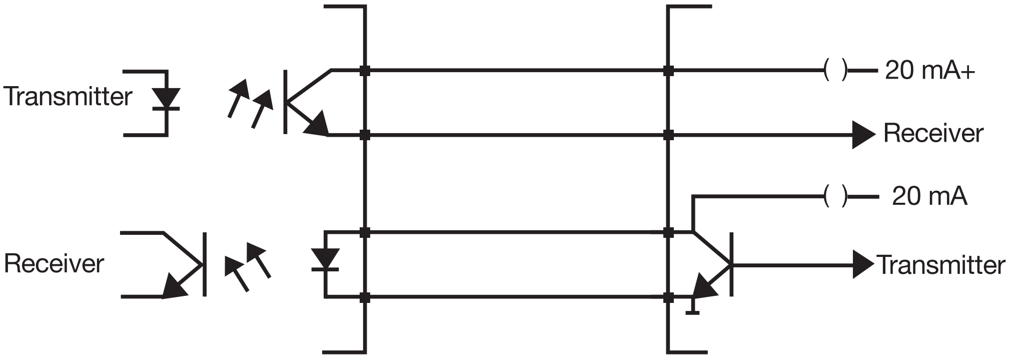


Fig. 19: Current loop with opto-coupler on a TTY interface

Because of the low internal resistance of the TTY interface, a number of devices can be connected in series in a ring formation. The central computer then becomes a component part of the ring. Of course, this presupposes that suitable software is available to facilitate system management (device numbers etc.).

There is no standardized plug connection for this interface. Hence, the most suitable connector can be chosen for each specific application. A good solution is a 9-way sub-D connector, as this is also used for many other interfaces (see Fig. 20). Table 5 shows the pin assignment and the corresponding functions. The transmit and receive signals are positioned opposite one another on the connector, in order to offer good termination possibilities for twisted-pair cables.

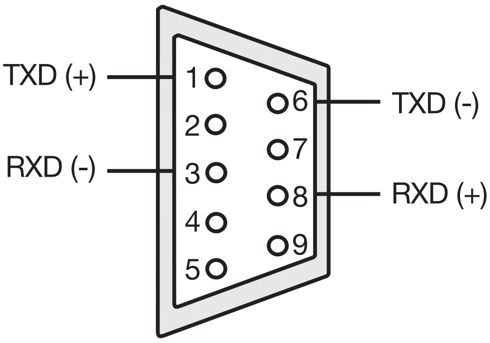


Fig. 20: Example of a connector used with the TTY interface

| Pin | Function | |
|---|---|---|
| 1 | TXD(+) | Transmitted Data(+) |
| 2 | - | - |
| 3 | RXD (-) | Received Data(-) |
| 4 | - | - |
| 5 | - | - |
| 6 | TXD (-) | Transmitted Data(+) |
| 7 | - | - |
| 8 | RXD (+) | Received Data(+) |
| 9 | - | - |

Table 5: TTY interface pin assignment and signals

Because of its high noise immunity, this interface can be used in a relatively “rough” environment and will achieve very high reliability. Table 4 provides additional technical data.

**RS232C interface**

The RS232C (V.24) interface is one of the best known serial interfaces. In Germany, it is often found under the designation V.24. In fact, the V.24 standard applies only to the functional values and the V.28 to the electrical values of the American RS232C standard. Generally speaking, "V.24" is taken to mean the electrical values as well. The electrical and mechanical data are laid down by the German Industrial Standards Organization (DIN) in the standards, DIN 66 020, 66 021 and 66 259.

The RS232C interface was originally designed to connect computers over telephone lines. Here, the computer (referred to as DTE, data terminal equipment) is connected by means of the interface to a modem (referred to as DCE, data communications equipment). The same arrangement is used at the other end to decouple the signals.

The RS232C interface defines a total of 20 lines. However, many of these are specially designed for modem features, such as the acknowledgement of signal quality and the like. They are not required for data transfer, so that most of the 20 lines can be ignored.

RS232C and DIN standards use different nomenclatures. The DIN nomenclatures are based on differentiating between ground, status, control, and data lines. However, the standard American abbreviations are used in the following description.

The pin assignments given in Table 6 refer to a 25-pin and a 9-pin sub-D connector (see Fig. 21). Normally a 25-pin sub-D connector is used. The pin assignments are applicable to data terminal equipment (DTE), see above. As only a few lines are used in practice, the 9-pin connector is used increasingly nowadays; it has a non-standard pin assignment and was originally introduced by IBM.

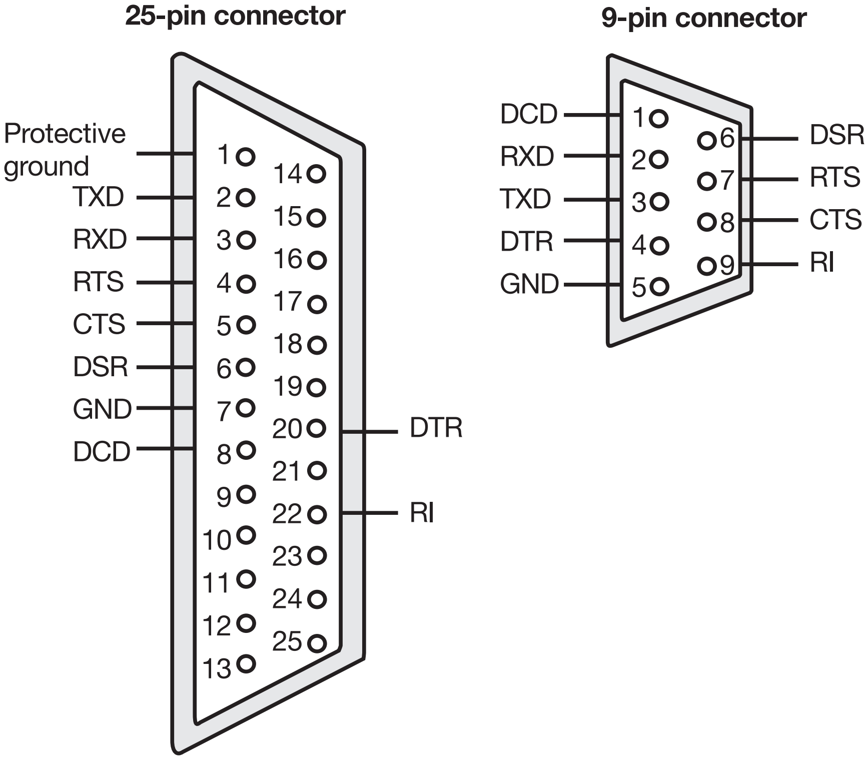


Fig. 21: Examples of connectors used with the RS232C interface

| Pin number | | Signal abbreviation | | Function | Direction |
|---|---|---|---|---|---|
| 9-pin | 25-pin | RS232 | | | DTE |
| 1 | 8 | DCD | M5 | Data carrier detect | In |
| 2 | 3 | RXD | D2 | Received data | In |
| 3 | 2 | TXD | D1 | Transmitted data | Out |
| 4 | 20 | DTR | S1 | Data terminal ready | Out |
| 5 | 7 | GND | E2 | Signal ground | |
| 6 | 6 | DSR | M1 | Data set ready | In |
| 7 | 4 | RTS | S1 | Request to send | Out |
| 8 | 5 | CTS | M2 | Clear to send | In |
| 9 | 22 | RI | M3 | Ring indicator | In |
| | 1 | CG | E1 | Protective ground | |
| | 9 | TV+ | | Test voltage + | Out |
| | 10 | TV- | | Test voltage – | Out |
| | 11 | CK | S5 | High transmit frequency | In |
| | 12 | S DCD | HM5 | Secondary channel data carrier detect | In |
| | 13 | S CTS | HM2 | Secondary channel clear to send | In |
| | 14 | S TXD | HD1 | Secondary channel transmitted data | Out |
| | 15 | TXC | T2 | Transmit clock | Out |
| | 16 | S RXD | HD2 | Secondary channel received data | In |
| | 17 | RXC | T4 | Receive clock | In |
| | 18 | nc | | Unassigned | |
| | 19 | S RTS | HS2 | Secondary channel request to send | Out |
| | 21 | SQD | M6 | Signal quality detector | In |
| | 23 | CH | S4 | High receive frequency | Out |
| | 24 | nc | | Unassigned | |
| | 25 | nc | | Unassigned | |

Table 6: Pin assignment and signals for the RS232C interface

The signal levels employed are a negative voltage of between –3V and –15V to represent the passive state (high) and a positive voltage of between +3V and +15V to represent the active state (low). The range between –3V and +3V is undefined (see Fig. 22).

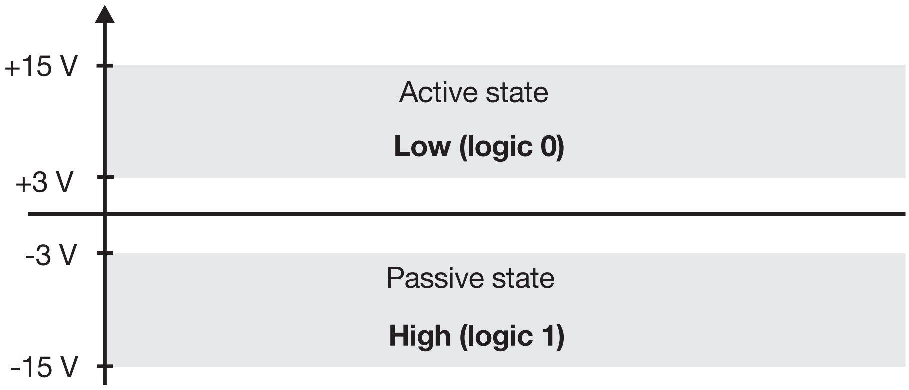


Fig. 22: Voltage levels used with the RS232C interface

The V.24 and TTY interfaces are analogous in that the passive state corresponds to logic 0 and the active state to logic 1, so the V.24 interface also inverts in the same way.

**Handshake**

The RS232C interface also facilitates both hardware and software handshaking. The CTS and RTS lines can be used to implement a hardware handshake. A device can then only transmit when its CTS input is activated. The RTS output is activated to signal that the device is ready to receive.

(The ready to receive signal can be interpreted as meaning that the device is requesting another device to transmit data. This explains the expression “Switch on transmitter”)

The DSR and DTR lines are sometimes used for hardware handshaking.

In addition, there is also the possibility of a software handshake, but this will not be discussed in detail here.

**DTE,**
**DCE**

The V.24 interface has been adversely affected by the fact that it was originally only designed for connecting a computer to a modem, and not for the direct connection of two computers. For this reason, there are two different connector arrangements, one for data terminal equipment (DTE) such as printers, computers, diskette drives etc., and another for data communications equipment (DCE) such as modems.

**In general, it can be assumed that data terminal equipment is present.**

The transmission installations are only explained for the sake of completeness, and, as a rule, the information already given still applies. In every case, two devices are wired such that an output of one is connected to the input of the other. As the devices are normally DTE and thus have the same connector arrangement, this results in **“crossed lines”**. Some writers also refer to a “null modem connection”, as it originates from the interconnection of two computers with the modem omitted.

Today, as far as the PC sector is concerned, termnial devices such as mice, printers, plotters, etc. are largely connected to the RS232 interface.

Table 4 gives additional technical details.

**Connection of JUMO instruments with RS232 interfaces**

In modern process automation with its bus systems and networks, the possibility of connecting process devices via the RS232 interface now plays only a relatively unimportant role. The main reason for this is that only one receiving device can be connected to a transmitter. Hence, this type of communication, used for instance with a PC, will only be explained briefly.

On the latest versions of JUMO process instruments, this form of data communication is normally only used to configure the device from a PC using the setup program (see Chapter 1.9.1 “Configuration software”). With earlier versions, e.g. controllers in the JUMO-DICON series, the serial interface is offered as a separate hardware item, for communication using suitable visualization software. The protocol for data transmission has a very simple structure and was developed using only the ASCII character strings.

JUMO instruments used 5 lines at the most, and normally only 3 lines of the V.24 interface (RXD, TXD, GND, CTS, RTS). A hardware handshake is possible with the CTS/RTS lines. These options can be checked in the appropriate interface description of these devices.

This results in the following wiring, for example, for interconnection of a JUMO DICON controller and a PC via the COM port (the example in Fig. 23 shows a connection without handshake). The pin arrangement refers to a 9-pin sub-D female connector.

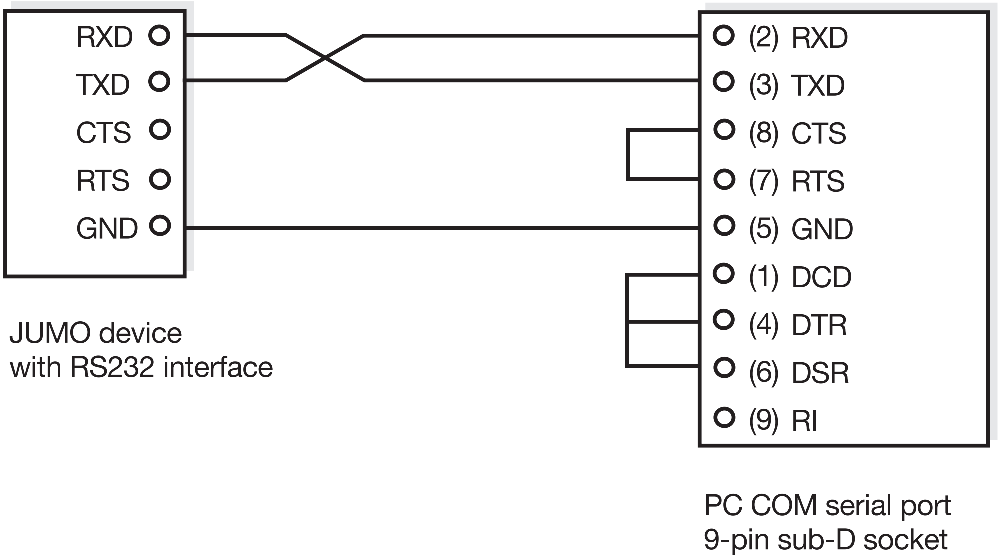


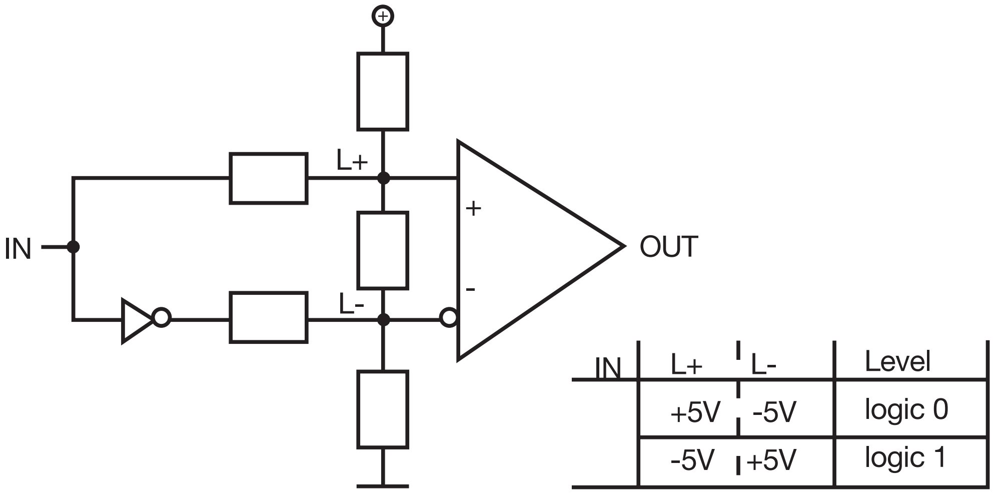
Fig. 23: Interconnection of JUMO devices with RS232C interfaces, without handshake

**RS422 interface**

The RS422 interface was drawn up by the EIA (Electronic Industries Association) in cooperation with the ISO. There is no DIN standard. Only the electrical and physical requirements are specified in this standard. The advantage of this interface in comparison with the RS232 is its suitability for bus operation, as a number of subscribers can be attached. The individual technical data are summarized in Table 4.

The interface has symmetrical inputs and outputs, giving it a high noise immunity, as interference voltages coupled into the ground line have no effect on the signal.

Normally, this interface uses voltage differentials of 5V as the signal level, identical to the TTL signal level. The voltage differential is always the same, only the sign of the differential is changed. If ground (GND) is not selected as the reference point, then when a logic 1 is transmitted, the voltage level is approximately -5V, and when a logic 0 is transmitted, the instrument reading would be +5V. The basic data connection consists of two lines, one for the normal signal (L+), and one for the inverted signal (L-) (see Fig. 24).



| IN | L+ | L- | Level |
|---|---|---|---|
|  | +5V | -5V | logic 0 |
|  | -5V | +5V | logic 1 |

Fig. 24: Voltage levels with an abstracted RS422 interface

Discriminating between a positive and a negative voltage differential to read the logic level has a favorable effect on the noise susceptibility of the transmission. Even though the received signal voltages may be very low, a reliable signal level can always be assigned. A voltage differential of 0.3 V between the two inputs is sufficient for safe assignment of signal levels; the voltage differential should not exceed 7V:

$$0.3V < \left| U(+) - U(-) \right| < 7V$$

There is no standardized connector for this interface either. As with the TTY interface, a 9-pin sub-D connector can be recommended, and it is often used (see Fig. 25). Table 7 shows the pin arrangement and the associated functions.

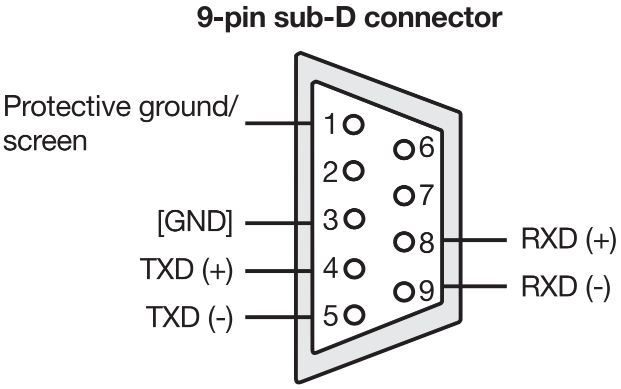


Fig. 25: Possible connector for use with an RS422 interface

| Pin | Function | |
|---|---|---|
| 1 | Protective ground/screen | ⇨ The connection is not essential |
| 2 | - | |
| 3 | GND/signal ground | ⇨ The connection is not essential |
| 4 | TXD (+)/Out (+) | Transmitted Data (+) |
| 5 | TXD (-)/Out (-) | Transmitted Data (-) |
| 6 | - | |
| 7 | - | |
| 8 | RXD (+)/IN (+) | Received Data (+) |
| 9 | RXD (-)/IN (-) | Received Data (-) |

Table 7: Pin arrangement and signals of the RS422 interface

However, the connector arrangement should be taken from the corresponding handbooks in each case. The remaining free pins can be used to transmit control signals. Here, as is the case with the TTY interface, neither ground nor supply voltage lines are normally carried. Although it is not essential in theory, it does however make sense to connect the signal ground (GND), if very different potentials develop on the devices as a result of inadequate electrical isolation.

To increase the noise immunity over the transmission path, twisted-pair cables should be used, or better still, shielded twisted-pair cables. Depending on the application or requirement, the shield can then be connected to the protective ground or technical ground of the device or devices at one or both ends of the cable. Here, care should be taken to maintain strict segregation from the signal ground (GND).

On JUMO process instruments with an RS422 serial interface, in addition to the four signal lines, the signal ground is always brought out to the connection terminals. The interconnection with a remote device (PC etc.) could then look like that shown in Fig. 26.

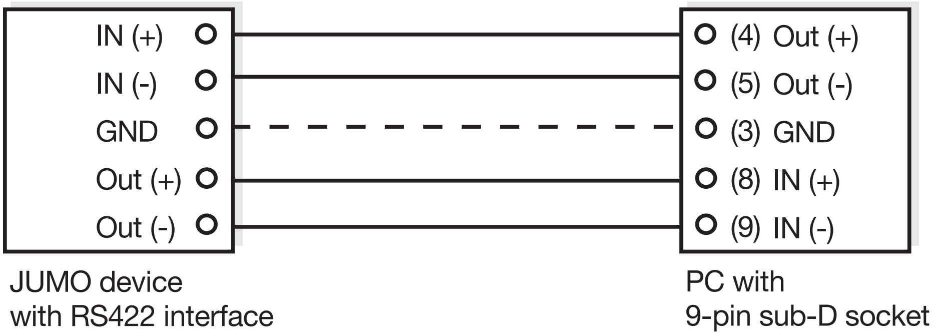


Fig. 26: Interconnection of a JUMO device with RS422 interface

**RS485 interface**

The RS485 interface, an EIA standard, once again lays down only the electrical and physical requirements. It was developed so that a higher number of bus subscribers could be connected for bus operation (see Chapter 1.6 “Networks and bus operation...”). Here, up to 32 subscribers, either transmitters or receivers, can be connected to a 2-wire bus cable for a balanced data transmission. Because of the balanced arrangement, a high noise immunity is achieved, as with the RS422.

There are no differences whatsoever between the method of transmission used by the RS485 interface and that used by the RS422 interface covered earlier. Both use the voltage differential method for the logic level. There are also no differences concerning the technical properties such as transmission speed and cable lengths. Even the same twisted-pair cable can be used.

The single difference between the two interfaces is that, with the RS485, data can only ever be transmitted in one direction at a time, because, of course, only one bus is available. All devices, transmitters and receivers, are connected to the same pair, i.e. they are all connected in parallel (see Fig. 27).

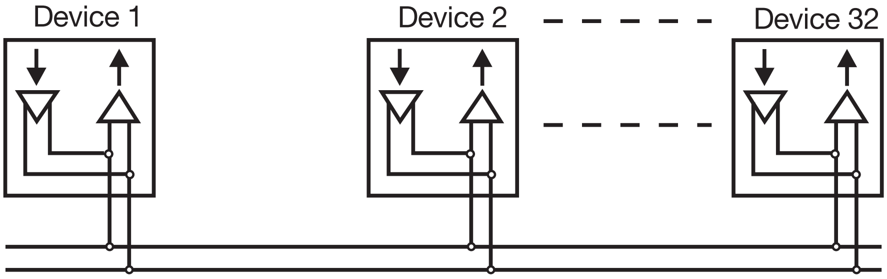


Fig. 27: Bus system with the RS485 interface

A PC is equipped with an RS232 interface as standard. An interface converter is required to operate an RS422 or RS485 interface (see Chapter 3.5.5 “Connection via an interface converter”).

As it is necessary to switch between transmit and receive modes, care must always be taken to ensure that only one transmitter accesses the bus at any one time. This is the job of the PC, bus system and protocol used, and will be discussed in more detail later.

The master device (e.g. a PC) undertakes the switching between transmit and receive modes. To ensure a reliable application structure, the PC application software must have direct access to the register of the interface module. The specialized knowledge needed for this is not required when the RS422 interface is used. In view of this, it is recommended that the RS422 interface be used.

Here again, there is no standardized connector arrangement. Fig. 28 and Table 8 show a suggested pin arrangement for a 9-pin sub-D connector.

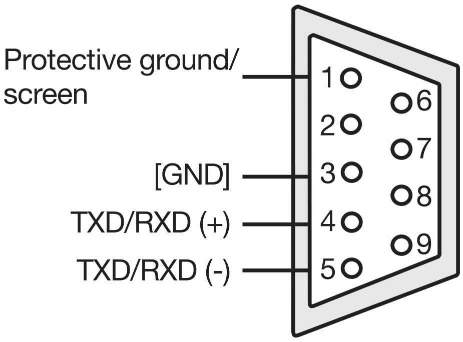


Abb. 28: Possible connector for use with an RS485 interface

| Pin | Function | |
|---|---|---|
| 1 | Protective ground/screen | ⇨ The connection is not essential |
| 2 | - | |
| 3 | GND/signal ground | ⇨ The connection is not essential |
| 4 | TXD/RXD (+) | Transmitted/received data (+) |
| 5 | TXD/RXD (-) | Transmitted/received data (-) |
| 6 | - | |
| 7 | - | |
| 8 | - | |
| 9 | - | |

Table 8: Pin arrangement and signals of the RS485 interface

The interface cards used in JUMO process instruments have a common driver for the RS422 and RS485 interfaces. Automatic switching between RS485 and RS422 takes place internally, depending on whether the device is connected for RS422 with 4 data lines, or RS485 with 2 data lines. Fig. 29 shows an example of the interconnection with the RS485 interface.

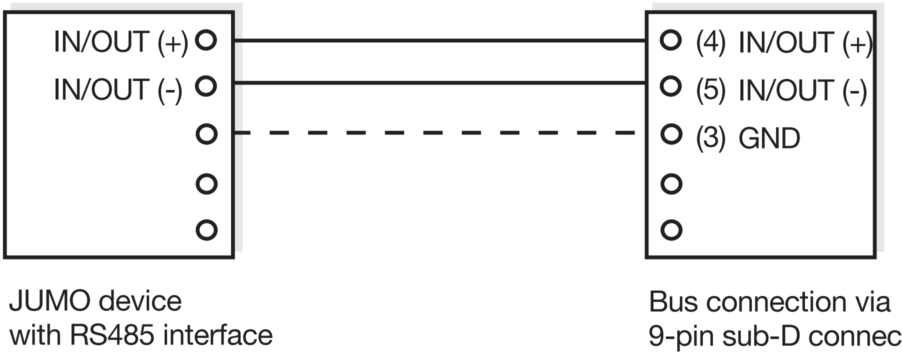


Fig. 29: Interconnection of a JUMO device with RS485 interface

Because of its high noise immunity and its suitability for bus operation, this interface is very often used for fieldbuses such as Profibus, Modbus, LON (see Chapter 2 “Important fieldbus systems”).

## 1.6 Networks and bus operation in automation

In today's computer-oriented automation engineering, the data bus or fieldbus is of central importance for communication between the individual measurement and control components in the process. A bus can be said to exist, if a number of units use the same signal lines to transfer data. Basically, any subscriber can communicate with any other subscriber over the bus. The bus can be inside a digital process device, used for data exchange e.g. between CPU, memory, and I/O devices. Both parallel communication, where data is exchanged over a number of parallel lines, and serial communication are possible.

Here, we will consider especially the exchange of data between individual automation components. Normally, this takes place over a serial bus, the fieldbus. In this context, it represents a cost-effective data network for active and passive subscribers at field level, for transmission of distributed input/output information.

Because such a bus system has to cope with many different tasks, depending on the application, and in recent years many firms have used their own bus systems, the number of different bus systems has steadily increased. The idea of having a unified fieldbus standard has been dropped as a result of industrial political infighting. Today, there are a number of user clubs or user organizations who would like to push their proprietary bus system through as the "standard". Some well-known bus systems are presented in Chapter 2.

A fieldbus represents the only way to fully utilize all the possibilities of modern digital field devices within a system.

**Field multiplexer**

One of the advantages of such databuses is lower cabling costs and the associated reduction in installation and material costs. In total, in a typical conventional installation, information relating to 40 inputs/outputs has to be exchanged between two control cabinets several meters apart, via 80 lines and their corresponding terminals (see Fig. 30). With this arrangement, known as **point-to-point connection**, each sensor or actuator has only a single connection to the terminals of one other device. Nowadays, the control cabinet may contain a field multiplexer that collects the analog and/or digital signals and transmits them serially, bit by bit, to the central control unit via a two-core cable, the bus.

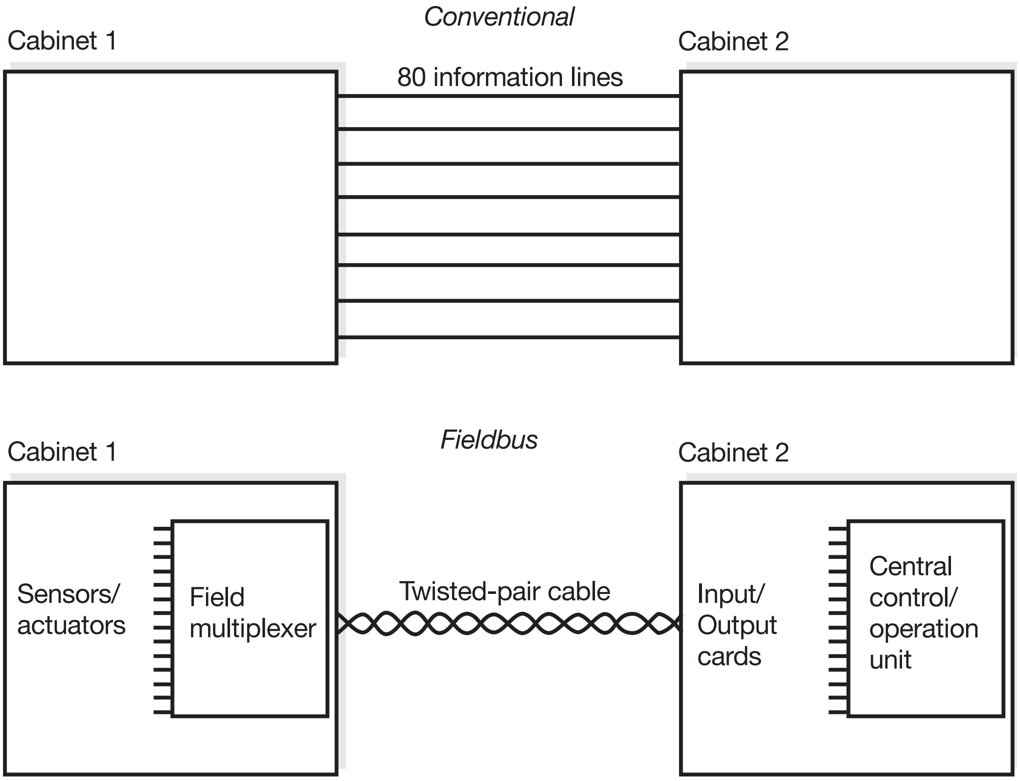


Fig. 30: Conventional control and networked bus communication

A further advantage is that the existing bus structure can be easily expanded or modified. The installation also remains transparent during such expansions, and in most cases, it is possible to upgrade bus subscribers with the system in operation. In the event of a fault, for example, a subscriber can be removed from the bus without switching off the system. There is no resulting shutdown, hence plant availability is improved. In the same way, the use of fieldbus systems for data interconnection also offers the possibility of using the PC as a subscriber on the bus, to collect and visualize process data at the central point, or even to operate the process.

Additional demands which are made on a fieldbus today, through the widely different applications, are:

- high security of the data transmission
- simple programming and diagnostic possibilities of the bus system
- deterministic dynamic response on the bus (e.g. real-time response for controls)
- interoperability (use of devices from different manufactures through standardized parameters)
- transmission of auxiliary power over the bus cable
- safe operation in zones with an explosion hazard (Ex zones)

In comparison with the conventional technology, when a bus system is used, the user is initially confronted with a number of new considerations that bring certain costs with them. This does not simply mean the hardware costs of the process devices (PLC, process controller, etc.). When such a system is set up, various costs are incurred right at the start, during the information and discussion phases, i.e. during the pre-planning stage. Then there are the project planning and programming costs incurred relating to data exchange on the bus (software costs), which nowadays, at around 50 %, make up the largest cost component of an installation. As well as this, the production of system documentation, the commissioning phase and the after-care and maintenance of the installation must all be taken into account. Specially trained, skilled staff are required for all these tasks.

Nevertheless, when all factors are considered together over a number of years, automation represents a considerable cost reduction for the user, in comparison with conventional installations, in spite of its initial costs. Because of this, fieldbus systems play a central role in automation, and they can be described as the backbone of future industrial communication.

### 1.6.1 Communication networks and levels

**WAN**

The range that a communication network or bus system has to cover constitutes a distinguishing feature of networking. Distances range from a few meters to kilometers. Networks that allow connections over longer distances are called wide area networks (WANs). Here we must distinguish between the public telephone networks and the private networks i.e. remote control networks (signal transmission via high voltage grids, gas or oil pipelines for monitoring and control of widely distributed processes). The public analog telephone network is familiar to everyone, and is by far the most comprehensive national/international WAN. This network deals primarily with voice communication by means of dialled connections. The use of this network to transmit data via a modem or in the form of a fax transmission is still of great importance, even though the future clearly belongs to the ISDN referred to earlier.

In the local area, up to several kilometers, three other types of network are in common use, the MAN, LAN and HAN (see Fig. 31).

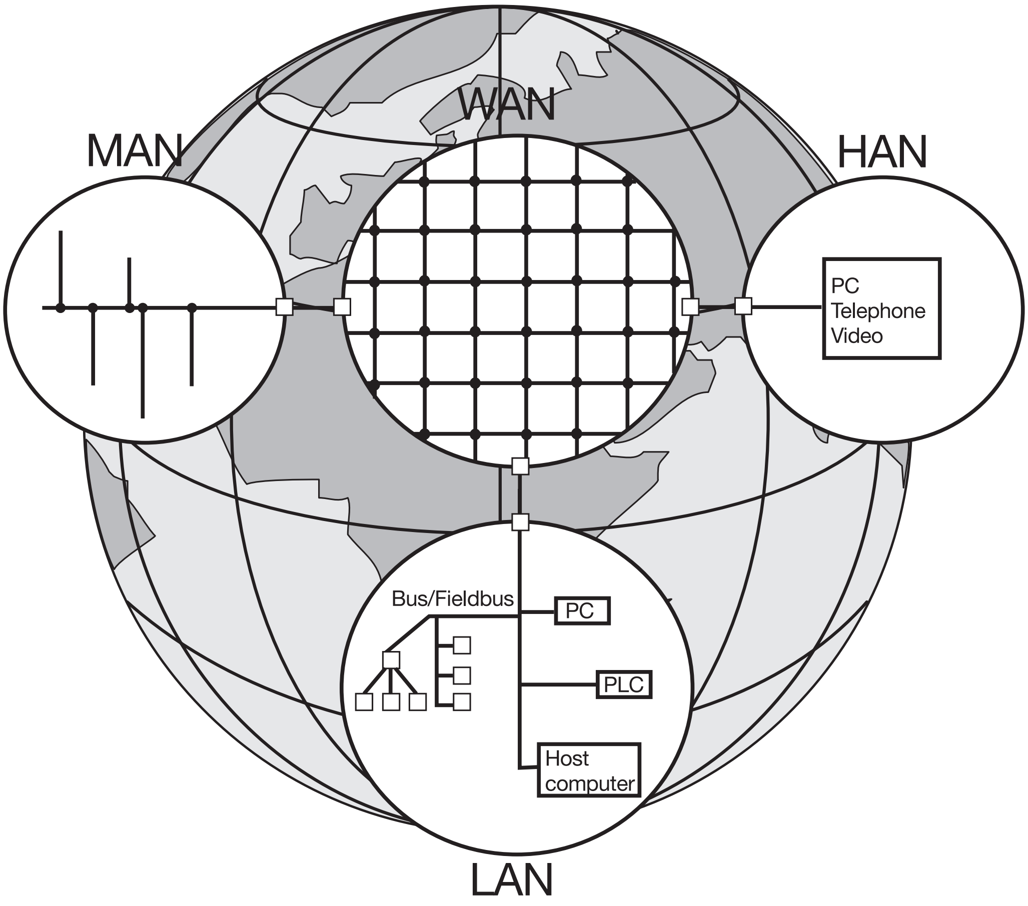


Fig. 31: Types of network in local and wide areas

**MAN** The MAN (Metropolitan Area Network) has a range of up to several hundred kilometers and should cope with communication in urban conurbations or towns. Of course, this form of network can also be connected with larger, more comprehensive networks, or even with smaller local subnetworks such as HAN.

**HAN** HAN (Home Area Network) should be used in the home or in the office. Here, various media such as telephone, PC, video and hi-fi system etc. can be networked together.

**LAN** The LAN (Local Area Network) is well known in industrial production plants. Its range is limited to a few kilometers. This type of network, for example, can link together various process devices (PC, PLC, controller etc.) within a production plant. The nature of this connection is largely determined by the actual application.

**Field levels** Networking within this type of system can extend from the lowest field level up to company management levels. The main objective of such a system is to make all events transparent, within a firm for instance, so that production/ manufacturing sequences can be rationalized. A typical communication hierarchy is split into five levels: field level, group control level, process control level, production control level and company control level (see Fig. 32). Occasionally, in smaller firms and with simple manufacturing sequences, only the lower levels are present. The processes are then normally operated by a PLC and/or a PC from a central control room.

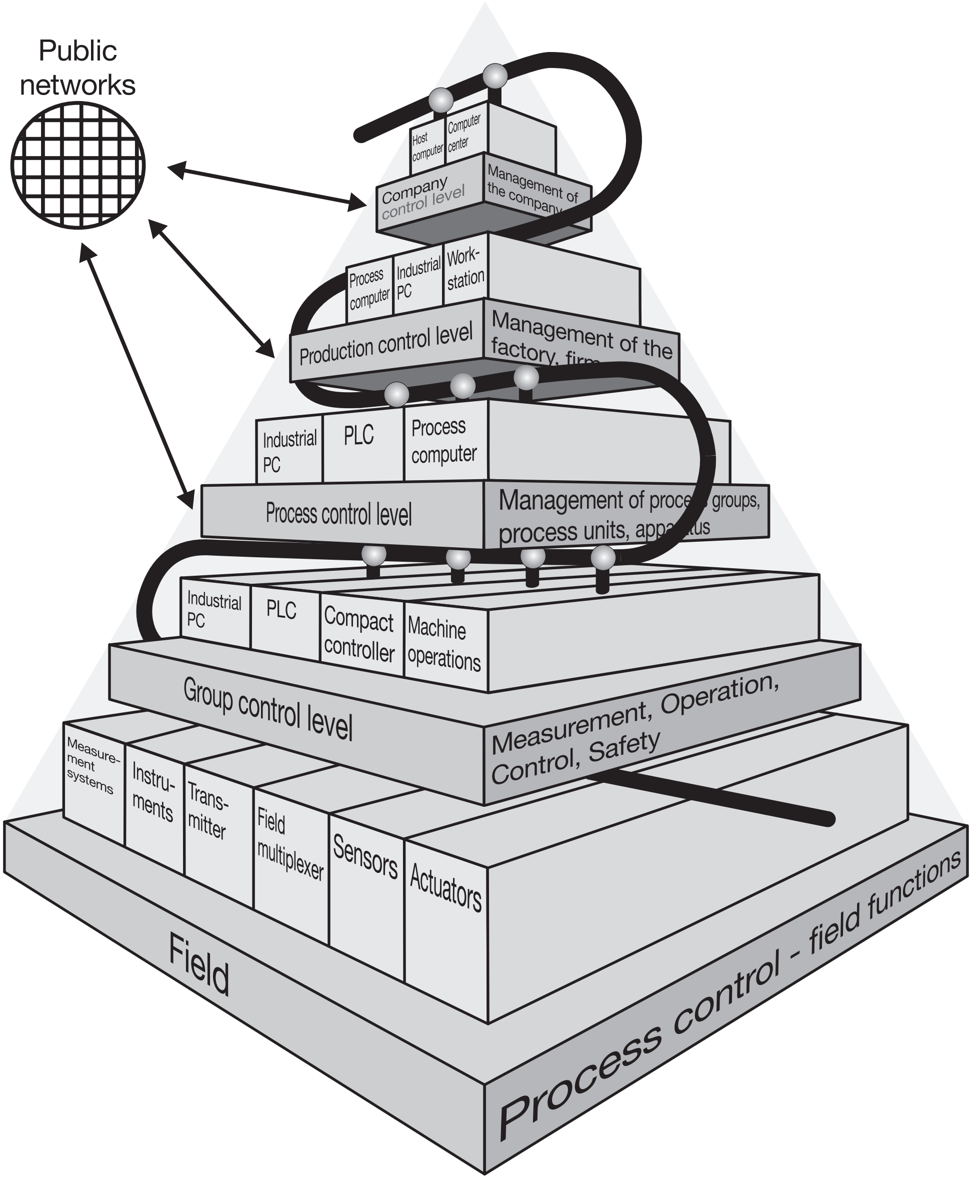


Fig. 32: Communication pyramid

Quite clearly, the actual bus system used must fulfill various performance criteria, because of the different data processing requirements at the individual levels. This is most noticeable in automation engineering, because the communicating devices have very different performance capabilities and functions. The data present at the individual levels are extremely varied and not of the same type. For example, the further one moves towards the lower levels of the pyramid, the shorter are the data sets transmitted, and the faster the data must influence the process. The term real-time response is often used in this context.

**Backbone**

Within the factory, the individual buses or local networks and their host computers are interconnected by a main data transmission line, known as the backbone. The connection of these, to some extent dissimilar, data networks is handled by a protocol converter, the gateway (see Chapter 1.8.4 “Connection of networks via repeater, bridge, router and gateway”). Standard LANs are used for these networks, such as Ethernet or even Fast Ethernet operating at 100 Mbps. There are two further network types, FDDI (Fiber Distributed Data Interface) or ATM (Asynchronous Transfer Mode), but these will not be discussed in detail here, because of their complexity.

**Sensor/actuator level**

The different protocol structures, especially with relation to the backbone, can be seen most clearly at the lowest level, the field or sensor/actuator level. Here, in the extreme case, information is transmitted simply as a single bit. The information may be binary control commands or alarm messages that are transmitted over a suitable fieldbus (e.g. Bitbus, ASI, PROFIBUS etc.). These can be either transmitted between individual field devices, or forwarded to the next higher level, the group control level.

**Group control level**

Process devices such as PCs, PLCs, compact controllers etc. are found at the group control level. These devices also pass information such as program settings or setpoint values in the opposite direction. In addition, individual groups of machines or production units are controlled and managed from this level. Examples of the fieldbuses employed here are Interbus-S, PROFIBUS, Modbus etc.

**Process control level**

Entire process groups consisting of individual production units are then supervised from the higher process control level. This level is occupied by the individual control centers with their PCs and host computers. Similarly, at this level, bus systems such as PROFIBUS are used to interconnect the host computer and the real-time computer.

**Production control level**

The production control level is responsible for the management of the firm or factory. Here, the entire production is supervised and controlled, and communication established with other sectors such as order preparation, purchasing, marketing and dispatch. This task is undertaken by process computers and workstations with their PPS (product planning system) packages.

**Company control level**

The highest level, the company control level, is responsible for the management of the entire company, and its operation is management-oriented. These strategic computer centers are used for the performance of administrative activities.

Here again, data links from different sectors to the public networks can be established, via the individual communication levels. In conclusion, it should be noted that, even at the two lowest levels (field and group levels), there are complex automation tasks using digital field devices. These tasks (such as visualization, operation, setpoint setting) are best carried out by suitable fieldbus systems with their connected automation devices (PC, PLC, controller). The performance capabilities of modern automation devices can only be fully utilized by exchanging data over this type of bus system.

### 1.6.2 Fieldbus topologies

In the previous section we mentioned that in a LAN within a factory, several data processing devices and automation units can be linked together in a network. The structure of this network is largely determined by the actual automation concept used. The structure and classification of such networks are identified by various characteristics.

**Topology**

These characteristics are the transmission medium, access methods (see Chapter 1.6.4 “Access methods”) and the topology. In general, topology denotes the network architecture, by which is meant the arrangement of the individual communication partners (terminal devices, PCs, etc.).

There are a number of topologies (structures) which differ with regard to the requirement criteria, such as availability, expandability etc. The basic forms are star, ring or bus, and in practice some networks are made up of a mixture of these forms. The basic forms are briefly compared in Table 9.

| | Star structure | Ring structure | Bus structure |
|---|---|---|---|
| Operation and access authorization | Central operation, access authorization controlled by central intelligence | Distributed operation, access authorization transmitted from device to device | Both central and distributed operation possible |
| Availability and redundancy | Failure of the central intelligence means that the network fails too. | If the cable fails, the network fails too. Bypass switches are required to avoid losing a device. | Dependent on the bus control mode. For central operation, see star structure,<br>for distributed operation, see ring structure. Individual device failure has no effect on the functioning of the network. |
| Expandability | Limited by the number of lines to the central control | Theoretically unlimited, in practice limited by the token circulation time (see 1.6.4), which determines the reply time. | Theoretically unlimited, in practice limited by the network polling time. |

Table 9: Comparison of various network topologies

**Star topology**

In the star structure, all the information is routed via a central node (e.g. a computer). All subscribers are grouped in star formation round this central computer and are connected directly to it (see Fig. 33).

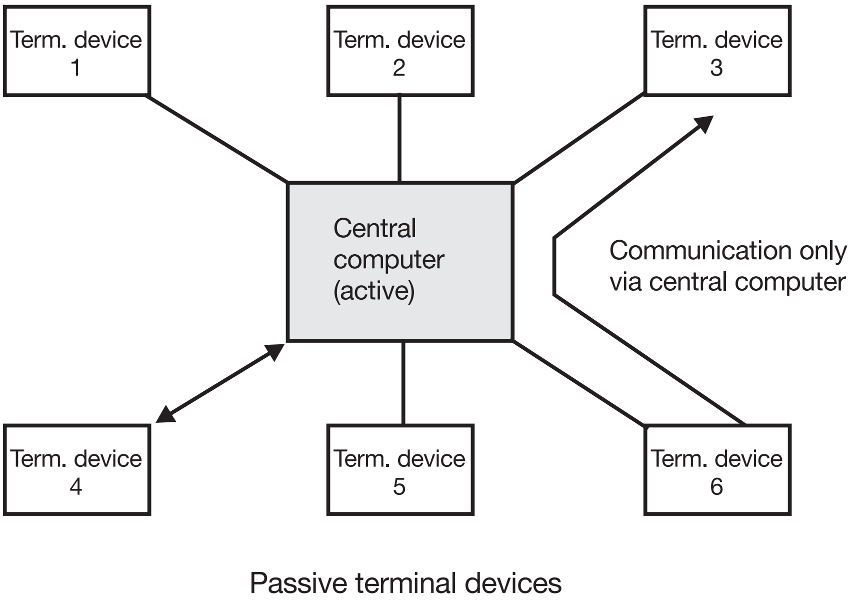


Fig. 33: Star topology

All communication takes place via the central unit, i.e. the individual subscribers cannot communicate directly with one another. Because of this, there is a high dependency on the central node, as a failure there means that the entire system is out of service. However, a fault on one cable normally only affects one subscriber, and the connection can be switched on and off to exchange a terminal device with the system still in service.

With this structure, any requirement to expand the installation by adding a subscriber can only be met by means of a new cable connection. Hence, adequate capacity for laying cables should be designed in at the project planning stage.

This topology is used predominantly in office communication, where a central computer or server is connected to a large number of PCs.

**Ring topology**

With the ring structure, the information is passed on between several stations that are connected to one another in a ring (see Fig. 34). Each station has a transmitter and a receiver. If a station receives data, these are copied, marked with an acknowledgement and then passed on via the ring to the next station. In this way, the corresponding transmitter recognizes that the data have been correctly received. There is no central operation, each station takes control of the ring at the time of transmission.

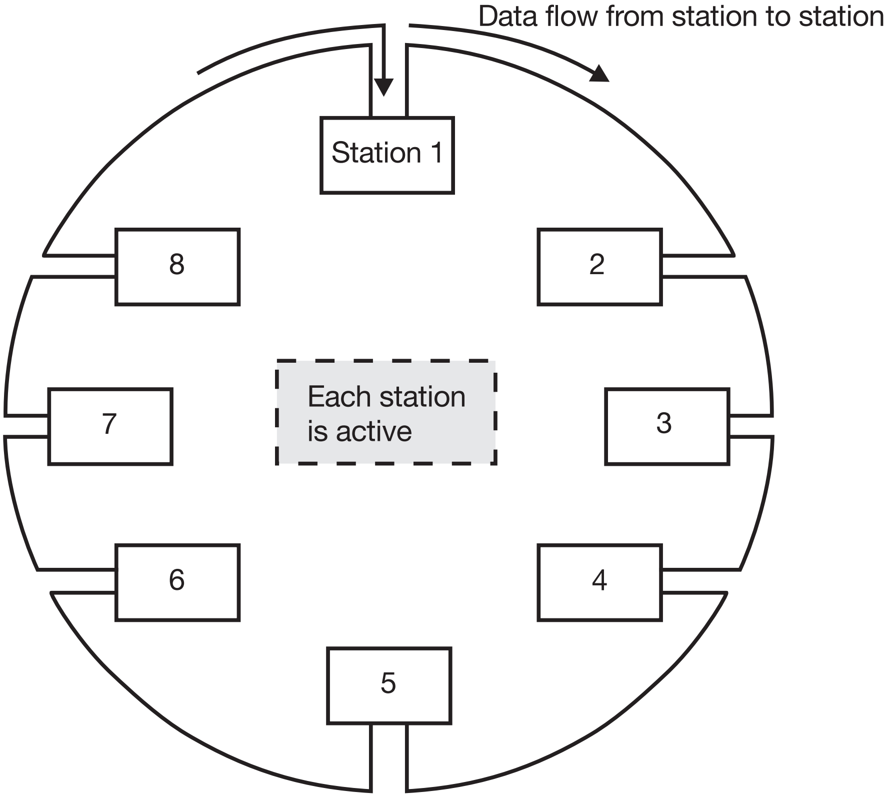


Abb. 34: Ring topology

Here again, data exchange is brought to a standstill when one station fails, as the ring is now broken. To avoid this, bridging switches have been introduced, which reclose the ring when a station is out-of-order. The use of these switches with this structure once more allows stations to be exchanged with the system still in service.

A typical representative of ring topology is, for example, the Interbus (see Chapter 2.6 “Interbus”). Here, the outgoing and return lines are combined in one cable. This type of topology can also be implemented with the LON technology used by JUMO. The requirements for such systems are mainly with regard to cost savings on cabling, commissioning and maintenance.

**Bus topology**

The best known and most often used LAN structure is the bus structure (sometimes referred to as line structure). All stations or subscribers are connected to a common data line, the bus, via short spur connections (see Fig. 35). Here, in theory, each bus subscriber can communicate with any other subscriber, i.e. all stations would have equal rights. However, this would lead to a data collision on the bus, and the information content would be destroyed. To avoid this occurrence, access methods (see Chapter 1.6.4 “Access methods”) are defined, which regulate the data traffic on the bus.

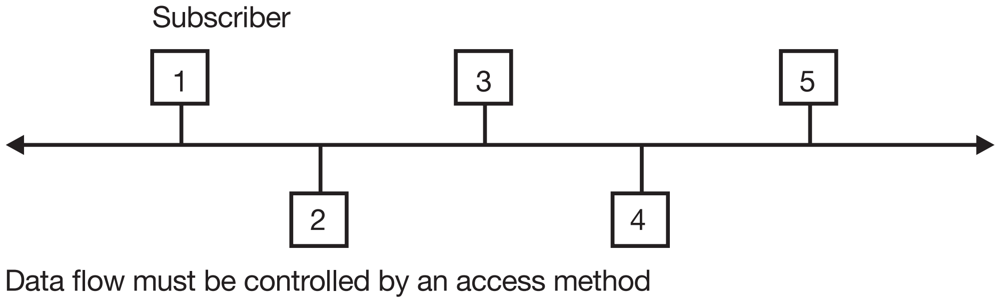


Fig. 35: Bus topology

The installation can be easily expanded to accommodate additional devices, called bus subscribers. Cabling costs here are also very low, however they do depend on the physical interface used. Very often, the RS485 interface is used with a twisted-pair cable as the transmission medium. The range of a bus system is limited because of this physical structure. With the RS485, for example, the maximum range is around 1200 meters.

**Repeater**

To achieve greater ranges, individual bus segments can be connected via repeaters (see Chapter 1.8.4 “Connection of networks via repeaters...”). A repeater is a bidirectional line amplifier used to extend the data transmission range. A distinction should be made here between regenerative and non-regenerative repeaters. Regenerative repeaters regenerate the data signal and can therefore be used to expand the topology (see Fig. 36). These repeaters give rise to branches, also known as tree structures. In addition, different topologies can be combined together, giving a mixed topology.

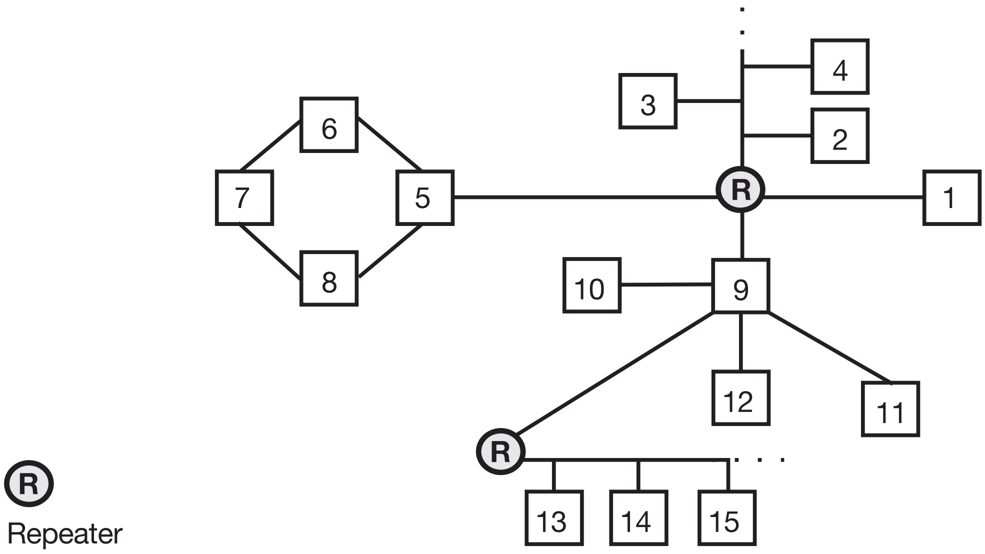


Fig. 36: Expanded/mixed topology using repeaters

If a repeater is defective, only one bus segment or section is lost. PROFIBUS is a typical example of this bus topology. Similarly, the LON technology employed by JUMO uses the bus topology or the mixed (free) form.

### 1.6.3 Centralized and distributed arrangement of automation devices

**Centralized arrangement**

When an automation device is arranged centrally, all the signals from the process are transmitted to a central operation and control facility. A typical example of this arrangement is the PLC, the central operating unit to which all sensors and actuators have to be connected. This is still standard practice with conventional analog technology (20 mA). Another possibility, in this age of the bus system, is to collect all analog and/or digital signals in a field multiplexer, and then transmit them via a fieldbus (bus cable) to the central control room using a digital protocol (see Fig. 37). Here, there are no independent intelligent automation units, simply sensors and actuators in the process. In this case, the central unit takes on all the data calculations, the operation and control sequences.

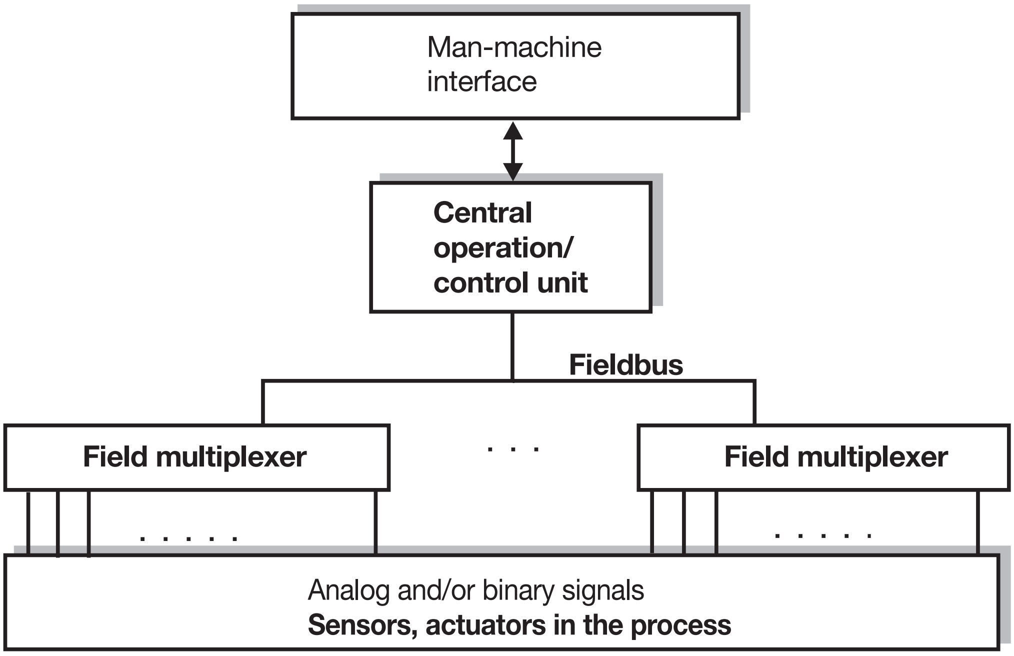

Man-machine interface
Central operation/ control unit
Fieldbus
Field multiplexer
. . .
Field multiplexer
. . . . . .
. . . . . .
Analog and/or binary signals
Sensors, actuators in the process

Fig. 37: Central automation device with fieldbus connection

The features of this centralized arrangement with digital data transmission via a field multiplexer are:

- short signal paths for critical signals
- lower installation costs compared with conventional methods
- higher accuracy through digital transmission
- quick modifications and simple expansion

**Distributed arrangement**

The current trend is to move away from the often functionally overloaded central unit towards networked **distributed independent automation devices** with local intelligence. The functions previously incorporated in the central unit are now transferred to distributed devices, with the aim of creating clearly defined and reusable units (see Fig. 38).

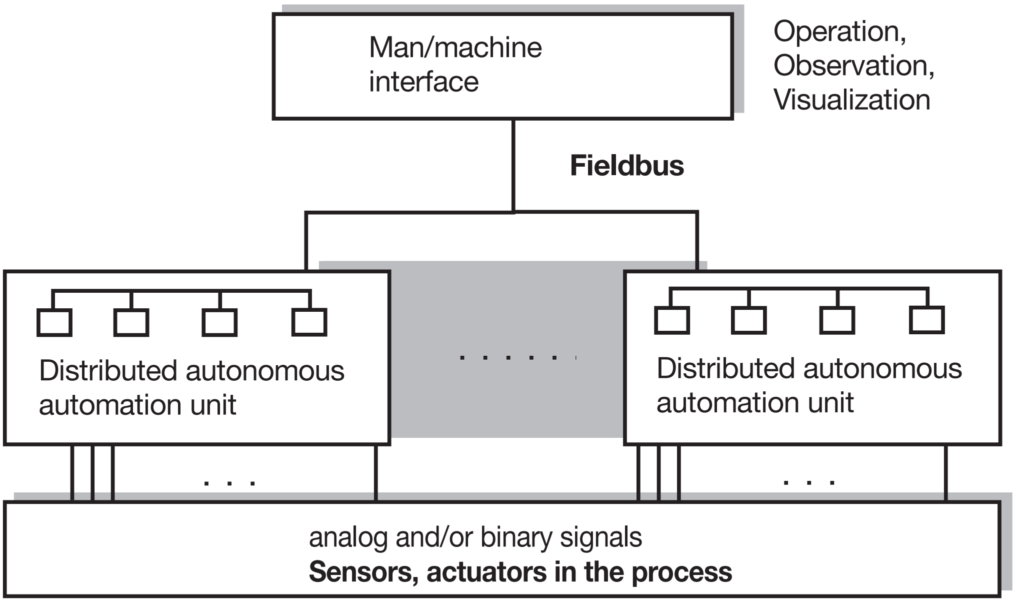

Man/machine interface
Operation, Observation, Visualization
Fieldbus
Distributed autonomous automation unit
. . . . . .
Distributed autonomous automation unit
. . .
. . .
analog and/or binary signals
Sensors, actuators in the process

Fig. 38: Distributed independent automation units, networked via fieldbus

Such a distributed device can, for instance, measure a temperature, compare it locally with the required setpoint, calculate the manipulating variable through an integrated controller function, and output it to the actuator. All intelligent devices are connected to one another by a bus (network). The automation highway is used to interchange the signals, parameter data and configuration data defined in the system design, in order to solve the set automation tasks on site.

The characteristics of distributed independent automation units are:

- short reaction times without dependency on the bus transit time
- higher system availability through independent units
- quick modifications and simple expansion
- task-oriented and user-oriented system structure
- simple, transparent programming, configuring and system parameter setting

### 1.6.4 Access methods

When a number of devices are linked on a data highway, an order of calling has to be organized, as otherwise all devices could transmit at the same time. As in a round of talks, only one subscriber should speak at a time. If two messages are superimposed on the bus, both sets of information will be mutually destroyed. The various access methods are responsible for ensuring an orderly flow of data.

**Arbiter**

With this process of bus allocation, also called *bus arbitration*, two basic methods are apparent. Referred to time, a distinction can be made between the deterministic (fixed) and the stochastic (random) access methods. The most important methods are described briefly below.

**Master/slave**

With the simple master/slave method, there is only one master on the bus. The master controls the access rights of the bus subscribers/slaves from a central point. This central control is also known as **"fixed master"**. Only the active subscriber (master) on the bus has the full right to transmit, whereas the passive subscribers (slaves) are only allowed to transmit when requested to do so by the master (see Fig. 39).

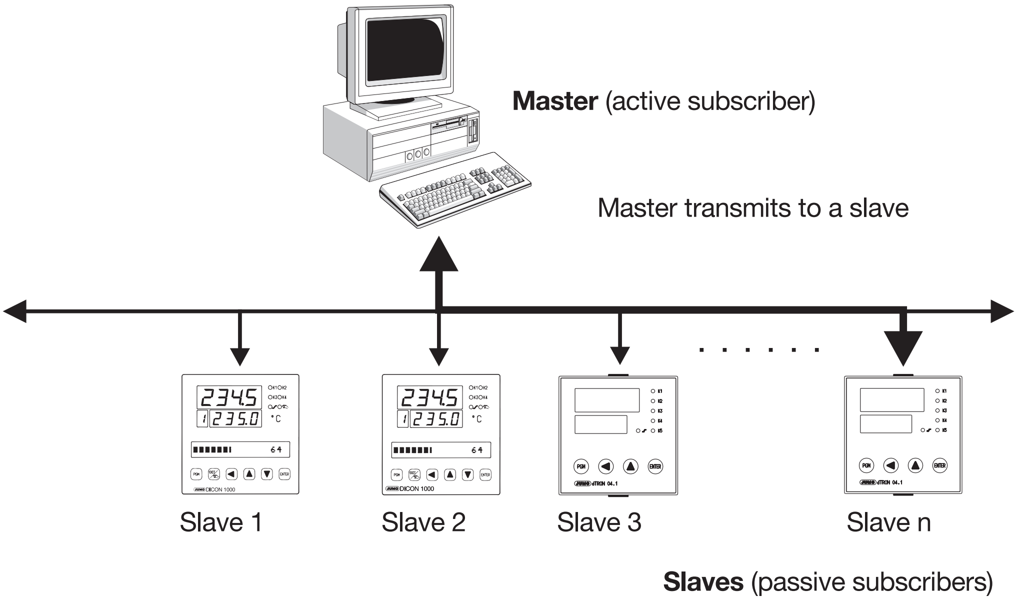

**Master** (active subscriber)
Master transmits to a slave
Slave 1
Slave 2
Slave 3
Slave n
**Slaves** (passive subscribers)

Fig. 39: Master/slave principle

**Polling method**

The master interrogates each slave in turn, in accordance with a cyclic schedule. Important subscribers can also be taken into account several times in this schedule. The continuously repeated interrogation, from the first to the last slave, is known as the polling method or polling cycle. One advantage of this principle is that a maximum reply time can be calculated, within which the data exchange between master and slave will have taken place in every case.

This communication, under the control of the master, is very simple, but there are also some reliability problems here, as no further communication is possible if the master fails. Other factors such as security and speed of data transmission depend on the physical interface employed and the protocol type used.

An example of the use of the master/slave method by JUCHHEIM, is the linking of compact controllers to a PC via the RS422/485 interface, using Modbus/Jbus protocol (see Chapter 3.5 "JUMO instruments with Modbus/Jbus").

**Token passing**

With the token passing method, there is no explicit master, and the right to transmit (right to talk) is passed on cyclically from subscriber to subscriber. This signal or right to transmit is known as the token. Whoever is in possession of the token has control of the bus, and must pass the token on to its next subscriber after a maximum transmission time. During this time, none of the other bus subscribers have the right to transmit.

This type of distributed bus control is also called “flying master”, as each subscriber can act as bus master at some stage, on receipt of the token. It then takes over control of the other non-active subscribers. The advantage of this method is that even with a heavy bus loading, a uniform bus allocation is ensured by the presence of a predetermined time (token circulation time). Time delays can ensue however, under fault conditions, especially when tokens are lost or doubled due to error.

There are two forms of the token passing method:

**Token ring**

Here the subscribers are connected together in the ring topology. The token cycles round the ring from subscriber to subscriber, and can be seized by a station wishing to transmit (see Fig. 40). The bus master discharges its message and waits until it arrives back there again. The token is now released once more.

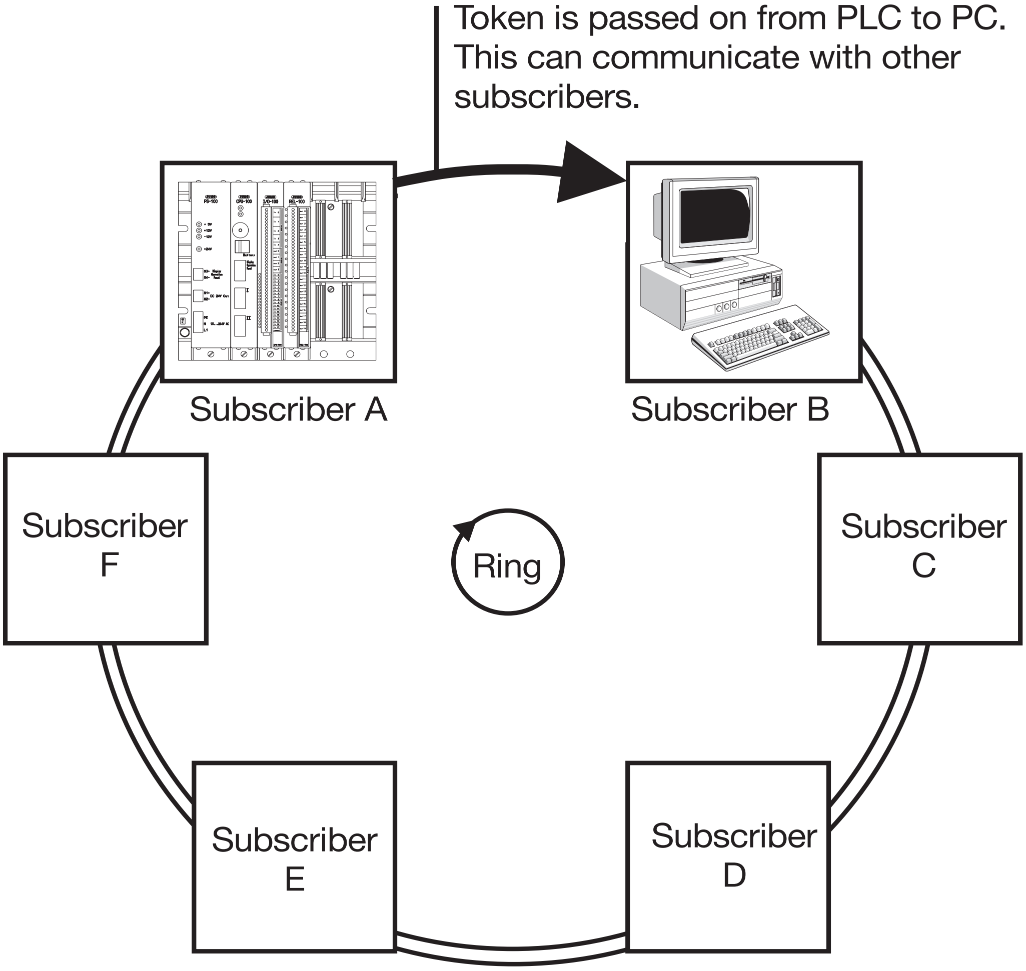

Token is passed on from PLC to PC.
This can communicate with other subscribers.
Subscriber A
Subscriber B
Subscriber F
Ring
Subscriber C
Subscriber E
Subscriber D

Fig. 40: Token ring principle

**Token bus**

Here all subscribers hang on a bus (see Fig. 41). As the concept of the token is retained with this arrangement as well, a “logical ring” is created. In other words, the sequence of subscribers is arranged in a list, and the token cycles round the planned logical ring in accordance with this list.

The difference between this and the token ring method is that here the transmitted message can be sent direct to the target station, and does not have to be looped via the individual subscribers.

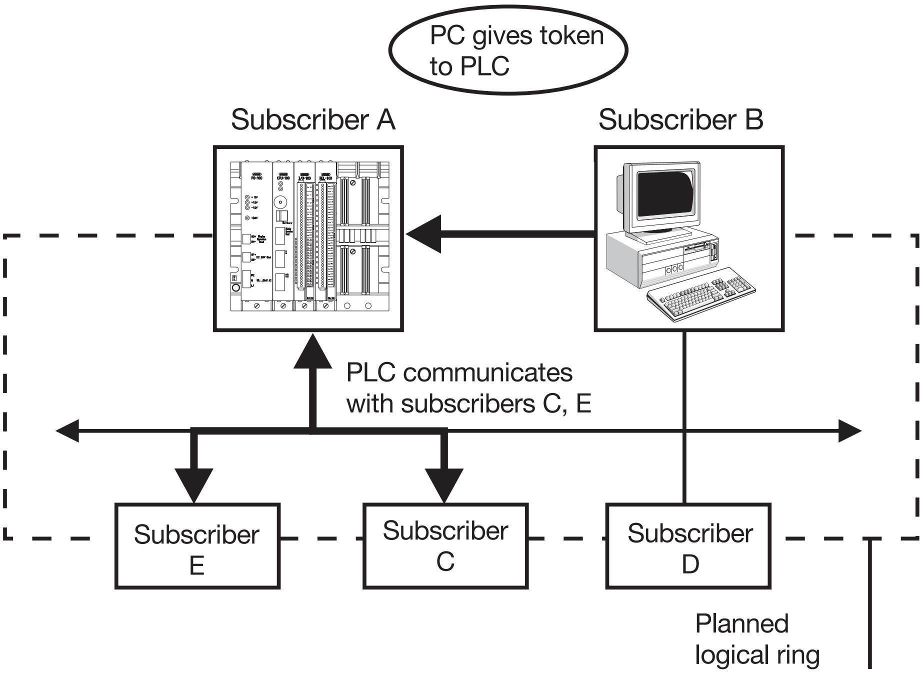
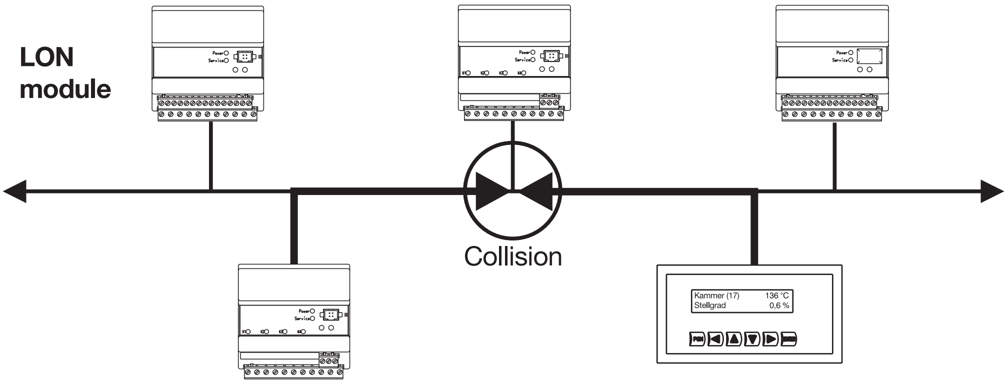


Fig. 41: Token bus principle

**CSMA method**

With the classic stochastic CSMA method (Carrier-Sense Multiple Access), all bus subscribers have the right of access. They all continually check the bus and wait until it is free. If the bus is free, a subscriber is able to access it (see Fig. 42).

It is possible for a number of subscribers wishing to transmit at almost the same time to determine that the bus is free and start to transmit. When this situation arises, there is a a collision between the messages. The messages are corrupted by the collision, and the subscribers have to make another attempt to transmit at a later stage.



Two stations transmitting at the same time

Fig. 42: CSMA principle

Suitable methods are employed to detect such collisions on the bus or even to avoid them altogether.

**CSMA/CD**

(Carrier-Sense Multiple Access with Collision Detection)

This method tries to detect collisions on the bus very quickly, by listening in and comparing with the transmitted signal. If a collision does occur, this is detected by the corresponding stations and they break off their attempts to transmit. A random generator in each station determines a standby time, after which a fresh attempt is started. This method works very well with lower bus loadings, but with higher bus loadings, the time spent waiting for access to the bus increases sharply.

This technique is used for example in Ethernet (in office communication) and also in the field of automation engineering.

**CSMA/CA**

(Carrier-Sense Multiple Access with Collision Avoidance)

With this technique, there are several different approaches to the problem of avoiding collisions in advance, by the use of various access methods.

One possibility is the allocation of a source address. When two subscribers transmit at roughly the same time, they both transmit their address. Now the subscriber with the lowest (or highest) address is accepted, and the other breaks off its attempt to transmit.

In the other method, each subscriber is assigned a fixed time interval at the end of its last attempt to transmit, after which it is allowed to access the medium once again, or the access right is allowed to travel cyclically, in much the same way as the token.

The JUMO mTRON system under LON (see Chapter 3.4 “JUMO instruments with LON”) also employs a type of CSMA/CA technique. As soon as the bus is quiet, the stations ready to send work out a form of sequence for the transmission, based on random probability. In this case, a series of slots are calculated, always in multiples of 16, up to a maximum value of 1024. The subscriber with the highest calculated value is allowed to transmit first (see Fig. 43).

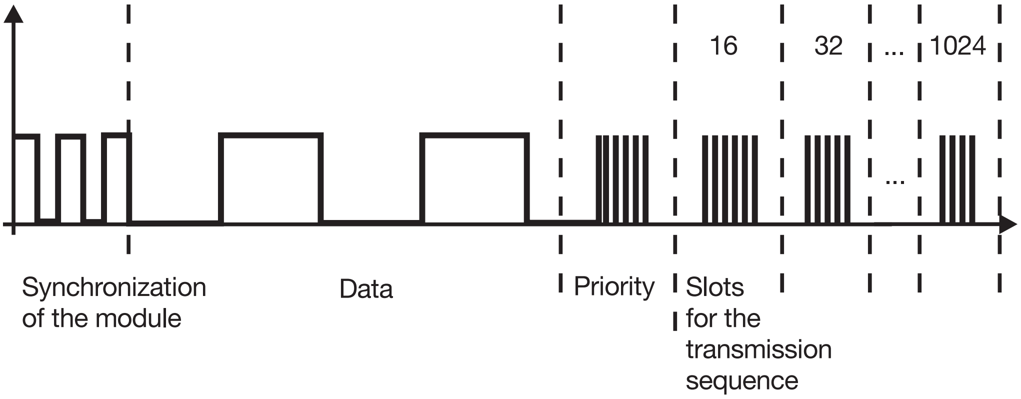

16
32
...
1024
...
Synchronization of the module
Data
Priority
Slots for the transmission sequence

Fig. 43: Access method with the JUMO mTRON system

The other subscribers have then to wait for their own “slot time”. The higher the bus loading, the higher can be the calculated number for the sequence (max. 1024). The possibility of two modules “throwing” the same number (e.g. 16) and the consequent data collision must be avoided. The disadvantage here is that once again the time spent waiting for access rights is longer at higher bus loadings.

With the JUMO mTRON system, there is the additional possibility of allocating a fixed priority, so that, in every case, a subscriber with an important message gains acceptance, in preference to the other subscribers.

### 1.6.5 Bus communication

There are two typical forms of communication over a bus, the parallel and serial bus. Here, parallels can be drawn with the parallel and serial interfaces described in Chapter 1.3.

**Parallel bus**

The parallel bus is used for the transport of data over extremely short paths. Its advantage is the high transmission speed when information is exchanged. The disadvantages are the high susceptibility to noise and the higher cabling costs compared with the serial bus. For this reason, parallel buses are normally used internally within devices such as PCs or microprocessor-based process devices (see Fig. 44). The address bus, data bus, control bus and supply bus are familiar examples here.

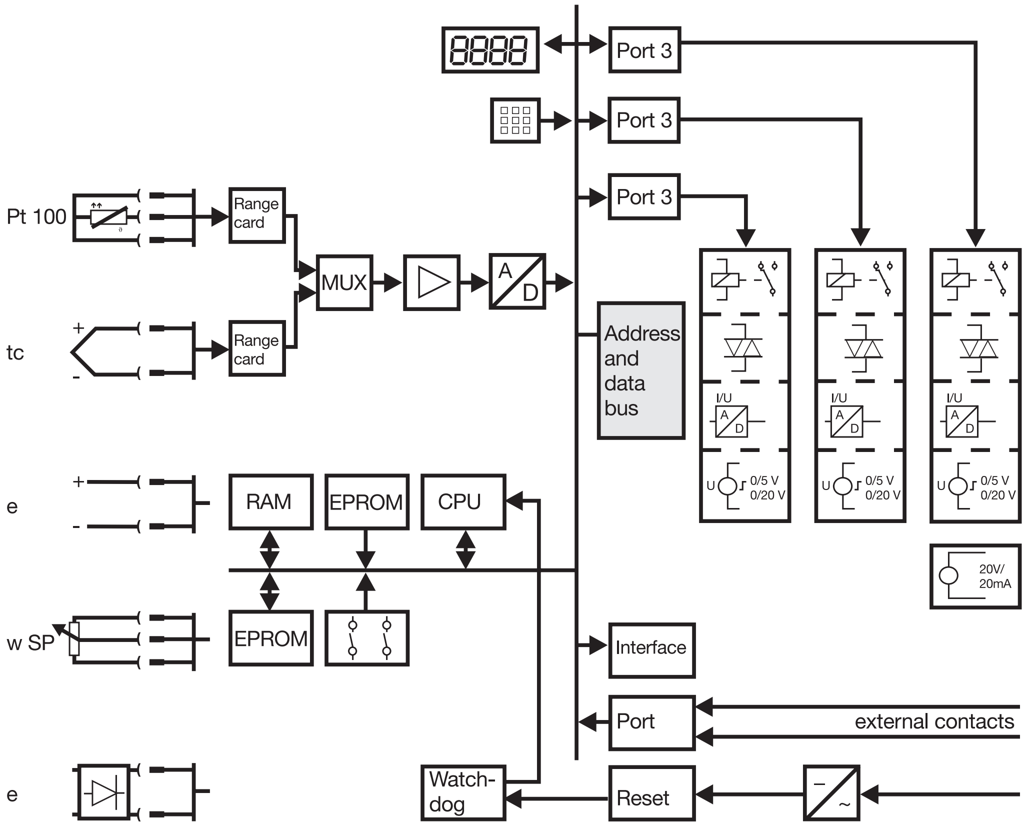


Fig. 44: Block schematic of a microprocessor-based compact controller with 16-bit address bus and 8-bit data bus

**Serial bus**

The serial bus is predominantly used as a peripheral bus for the operation of automation devices. It facilitates communication over longer distances with relatively high noise immunity. A further advantage is the lower cabling costs. One disadvantage compared with the parallel system is the lower transmission speed.

With serial transmission, the individual bits of a message are transmitted over the medium, one after the other. Depending on the particular bus system, the data are transmitted in accordance with a fixed set of rules, the protocol. In the last section covering access methods, we saw that only one packet of information can be transmitted at any one time and at a specific transmission speed. In communication technology however, the diverse applications dictate that different information with different transmission speeds must be handled at the same time. This necessitates different forms of transmission mode.

The three following forms can be distinguished:

**Baseband**

A baseband only transmits one character per unit time. The information is injected into the transmission medium in the form of pulses (e.g. in digital form as 0 or 1). This means that only one transmission channel is available for all bus subscribers.

Typical characteristics are maximum transmission speeds of 10 Mbps, a spatial range of up to 1000 meters, a maximum number of about 30 stations per segment, low connection costs and bidirectional use of the cable. Twisted-pair or coaxial cable is used as the transmission medium. Important representatives of this technology are Ethernet and practically all fieldbuses.

**Carrier band**

The information is modulated onto a carrier signal and transmitted in analog form (e.g. sinusoidal). Once again, as with the basis band, only one signal is transmitted at a time. The technique facilitates high transmission rates and security of transmission. Typical transmission rates are around 35 Mbps.

**Broadband**

This technique makes several transmission channels available. Several characters can be transmitted per unit time, as the information is modulated onto different carrier frequencies. Various frequency-division multiplexing methods can be used here. The data can be in analog or digital form. Coaxial or fiber-optic cable is used as the transmission medium. Transmission rates of 88 Mbps and beyond are achieved.

Typical characteristics are a high noise immunity, a range of several kilometers and the possibility of connecting several hundred nodes. This technology is used mainly in office and communication networks.

## 1.7 OSI reference model

In today's modern automation engineering, a wide variety of process devices from many different manufacturers have to be linked. They must all communicate with one another and jointly operate and control a system. Unfortunately however, not all these devices speak the same language, i.e. the individual devices may operate with different protocols (see Chapter 1.8 "Network management"). The protocol defines the data format and the operating procedure (code, transmission mode, etc.) for the exchange of data between two devices.

**Closed system**

Normally, there is no problem of understanding when devices from one manufacturer are used. This type of manufacturer-dependent system is often called a closed system, as there is normally no need to consider linking with devices from other sources.

**Open system**

Nowadays it is really necessary to interconnect very different devices, and the individual manufacturers are trying to follow the example of specific standards and guidelines. In this way, the different process devices, which are then compatible with one another, can be combined in an open system.

**ISO/OSI/ seven-layer model**

In 1978, in America, the ISO (International Organization for Standardization) took the first steps towards drawing up a standard that would achieve defined interfaces in the protocol, and hence independence from hardware and software. In 1984, the OSI (open systems interconnection) reference model that was developed, was adopted as ISO Standard 7498 for the abstract description of inter-process communication between physically separated communication partners. Some writers also use the terms ISO/OSI layer model or seven-layer model etc. Since this development, corresponding standardization work for bus and network protocols has been and is being carried out worldwide, based on the ISO standard.

**OSI reference model**

The OSI reference model controls and defines the sequence of the data communication at seven levels (see Fig. 45). Each level takes on a clear task to implement the communication. In principle, these seven levels can be split into two basic zones.

The lower or transporting (communication-oriented) layers 1-4. The exchange of data between process devices and their processors is controlled here.

The higher or user-oriented layers 5-7. They control the exchange of data between applications within the individual processors.

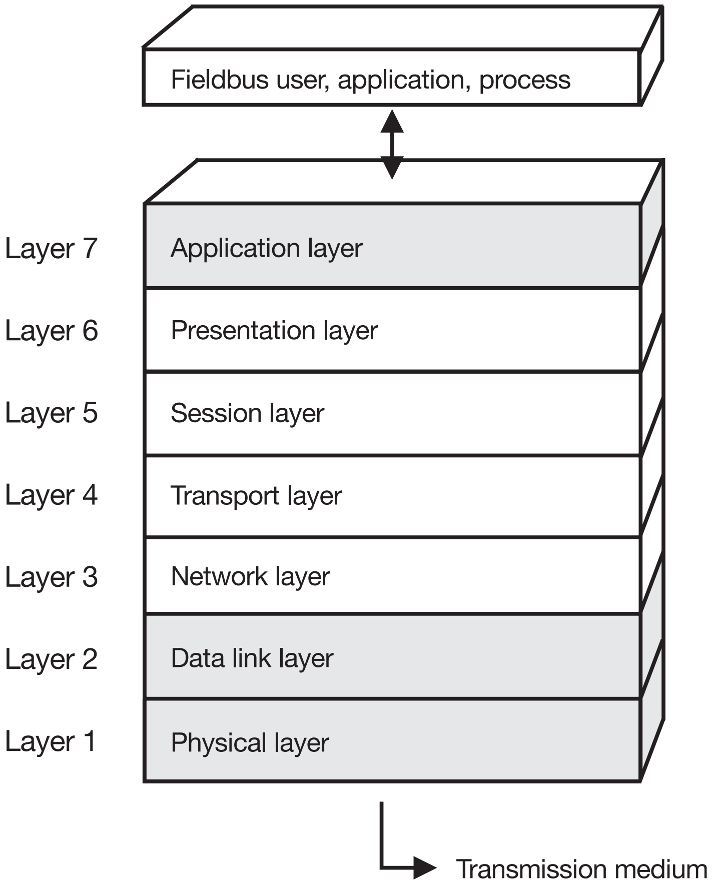


Fig. 45: OSI reference model layers

The seven protocol layers each have interfaces to the neighboring layers. When information is transmitted from PC “X” to PC “Y”, for example, the message originates in the highest layer, layer 7 (see Fig. 46). The actual application process takes place here at the highest level, in the application program. The message is forwarded through all the layers as far as layer 1. The message is now transmitted at the physical layer level, and is then reconverted by layers 1-7 of the receiver (PC “Y”). Seen from the user’s point of view, the data transmission occurs totally in the background. In other words, he operates his application program in the foreground, and the software implements the exchange of data in the background, as described.

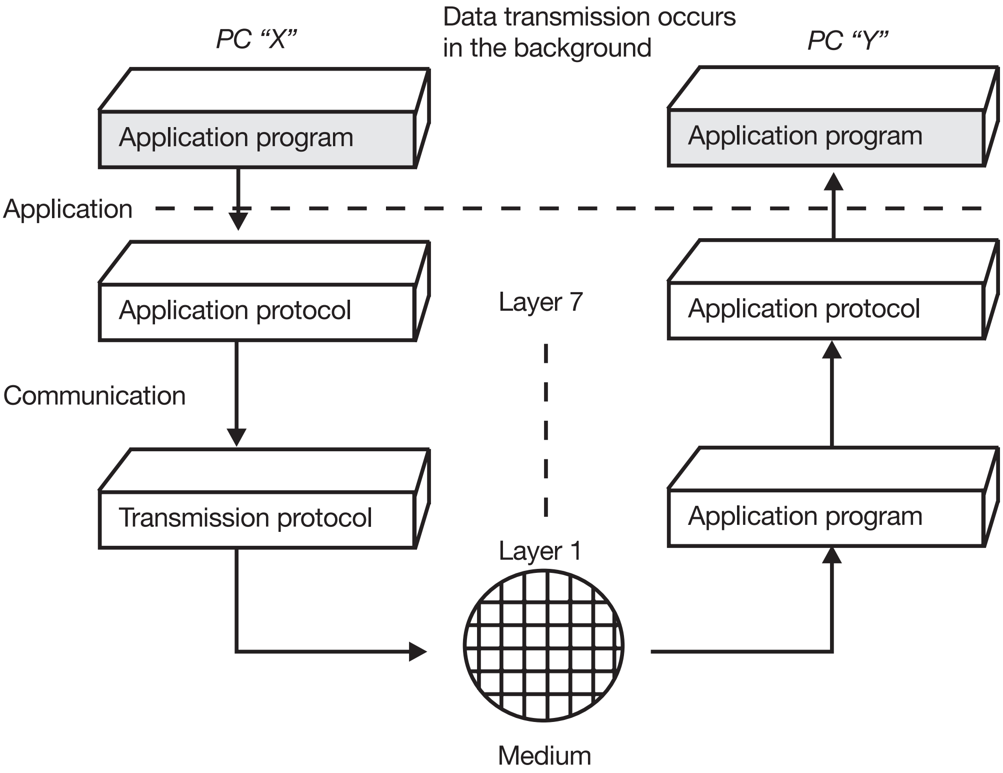


Fig. 46: Principle of data exchange between two PCs

For the time being, the reference model does not lay down how the individual layers are implemented in a specific case. It only marks out the broad framework; special implementations are left to the designer/user.

Normally, with a fieldbus system, layers 3-6 are dispensed with in favor of a fast, real-time capable, cost-effective communication, and those functions that are definitely required are integrated in layers 1, 2 and 7. A minimal construction of a bus system must consist of at least the two lowest layers.

The individual layers can be described as follows:

**Physical layer**

**Layer 1: Physical layer**
At this level, the individual bits are transmitted between subscribers via appropriate interfaces and an associated medium. Here, for instance, the encoding, connector type, and the connector pin arrangement are laid down. In addition, the type of cable (twisted-pair cable, coaxial cable, fiber-optic cable, etc.), its length, its composition and the mode of transmission (e.g. radio transmission) are specified. In the same way, the electrical specification (e.g. RS485) and the data rate are described here.

**Data link layer**

**Layer 2: Data link layer**
Here, data security is provided during the transmission, and the transmission protocol defined. This layer is sub-divided into MAC (medium access control) and LLC (logical link control). The MAC layer defines the access method, and the LLC layer handles the security protocol (on this subject, see also Chapter 1.8 "Network management").

**Network layer**

**Layer 3: Network layer**
The connection between two subscribers in a network is managed in this layer. With a complex network, the selection of the most favorable communication path (the “routing”) takes place here.

**Transport layer**

**Layer 4: Transport layer**
This layer facilitates full-duplex communication between two subscribers, as it provides the services for the data transport. These are services for error detection and correction, as well as requests for repetition.

**Session layer**

**Layer 5: Session layer**
It provides services that allow both subscribers to share common environments. These are aids for establishing and terminating the connection, or the arrangement of common data areas.

**Presentation layer**

**Layer 6: Presentation layer**
The task of this layer is the conversion of the machine-oriented view used at the lower levels, into a problem-oriented one. It defines procedures for conversion and format modification, and ensures a correct interpretation of the data.

**Application layer**

**Layer 7: Application layer**
It forms the interface between the user and the network. The application layer provides a wide range of services that the application program can use directly, e.g. a database, file transfer, job transfer, communication systems etc.

Many standards have already been developed in connection with this model. The RS232 and RS485 interfaces, for example, based on the transmission or physical layer, or the access methods such as CSMA used in the data link layer, which to some extent had already been introduced in advance. In addition, both PROFIBUS (which takes in layers 1, 2 and 7) and Modbus (covering layers 1 and 2 of this model) constitute a type of standard based on the reference model.

## 1.8 Network management

Network management is a versatile expression that is used in several areas in the context of communication within networks. For example, within a complex integrated network that is made up of several sub-networks, network management is used to describe the management of the sub-networks amongst themselves. In this case, network management means the linking of different network architectures, with different protocol applications.

Similarly, the expression can be used to describe the networking of individual subscribers with one another, within a sub-network. In addition, there is a partial or full network management structure in each subscriber. This is where the data exchanges between the different transmission layers/levels (see OSI reference model) in a device must be dealt with.

Broadly speaking, the tasks of network management for all application areas can be split into three areas:

1. Control of the data exchange and correction of errors. This also means the preparation of the signal to be transmitted, and the interpretation of the received signal.
2. Operation and monitoring of the network status and network configuration. This includes making regular checks on the bus, which subscribers are connected, which messages are sent, etc.
3. Translation and interpretation of commands from user and application programs, i.e. the data of the application program must be brought into a form that can be processed in the network.

**MAC**

How the individual functions are executed depends on the composition (type, size, etc.) of the network. For example, in a small sub-network with only a few subscribers and a two-wire bus cable, the MAC (medium access control) ensures that there is only one message on the bus.

### 1.8.1 Functions of MAC and MAP

The following section will now look at network management between the processing levels in an automation device (see Fig. 47).

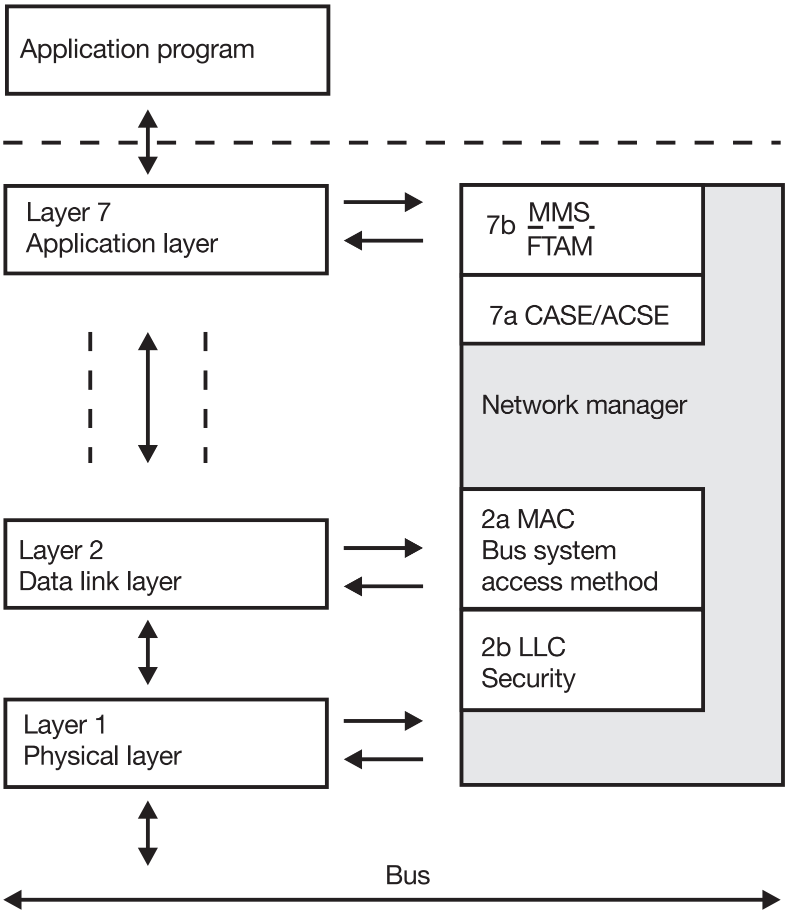


Fig. 47: Network management between the processing levels within a device

The OSI reference model has become a vital foundation for network technology. Because layers 1, 2 and 7 in particular are important for a minimum communication profile, these will now be considered in more detail. In the physical layer, layer 1, the individual bits are transmitted between the individual subscribers over a medium via a suitable interface, whereas layer 2 is concerned with the bus access and security methods required for a safe transmission of data.

**LLC**

These aspects are the job of the MAC and the LLC (logical link control), which are part of the network management. The two profiles divide layer 2, the data link layer, into zones 2a and 2b (see Fig. 47). This includes standardized methods such as the CSMA access method, as well as the data security protocol “CRC-16” (see Chapter 1.8.3 “Error checking”), i.e. measures are taken here to ensure that the connection between receiver and network output is as near perfect as possible. As well as error detection and correction, this also means a combination of the data into blocks, allocation of the addressing etc. These are all defined in the different bus protocols.

**TOP/ MAP**

We should also look again at layer 7, with its user protocols. Two different standards and directions have developed here. On the one hand, we have the TOP (technical office protocol) in the office environment, and on the other hand the MAP (manufacturing automation protocol) in the automation and process field. The MAP standard, as used in the automation field, will now be considered in a little more detail.

MAP was introduced by General Motors in 1980, as a result of the setting up of the MAP task force. It is a communication model which establishes appropriate protocols for each layer of the OSI reference model. Based on this, three "communication profiles" (combination of all the standards for each layer) have emerged:

**MAP (full MAP)**

This operates on the basis of all 7 layers.

**Mini-MAP**

This system operates only on the basis of three layers. Here, as in many fieldbus systems, layers 3-6 are omitted. This necessitates certain restrictions concerning transmission security, but these do lead to an increase in speed.

**MAP/EPA**

If there is a requirement to combine a mini-MAP protocol stack (3-layer model) with a full MAP protocol stack (7-layer model), a certain structure is needed, and this can be implemented through MAP/EPA (MAP/enhanced performance architecture). This structure contains all the components necessary for joining/ linking. Hence, the user has the possibility of communicating right the way through from the highest level (backbone) to the lowest level (field level).

A detailed description of the MAP concepts is outside the scope of this book. However, a number of specialized terms used in connection with MAP will be explained briefly.

**CIM**

As well as reducing manufacturing costs and increasing flexibility in manufacturing technology, MAP should create a foundation for CIM (computer-integrated manufacturing). CIM means the integration of all computers into a single unit, to maintain a continuous flow of information from receipt of order, through production and design, up to the delivery stage. This includes, for example, all machines such as CNC machines. The MAP standard even makes it possible to link components from different manufacturers in a common system.

To produce this type of communication infrastructure, certain standardized services must be made available to the user. On this subject, look again at layer 7 of the reference model. Under MAP, layer 7 is also split up into two subranges 7a and 7b (see Fig. 47).

**CASE ACSE**

Layer 7a contains general services, and is referred to as CASE (common applications service elements). These services, e.g. establishment, termination and cancellation of the connection, are available to any user. Either the user or the CASE service provider, referred to as ACSE (association control service elements), are responsible for these duties.

**MMS/FTAM**

Layer 7b is concerned with the special services that are not available to every user. These include, for example, protocols such as MMS, FTAM and the network management. The MMS (manufacturing messaging specification) protocol ensures a uniform communication between the process devices (PLCs, etc.). The terms “client” and “server” are also used in the context of MMS. In an office network, the software needed by each subscriber (client) resides, for example, in the server. The individual subscriber then requests specific services, which the server provides. In the FTAM (file transfer access and management), services are provided that facilitate transfer of data between two users. In other words, opening and closing of files, reading and writing of data records, etc.

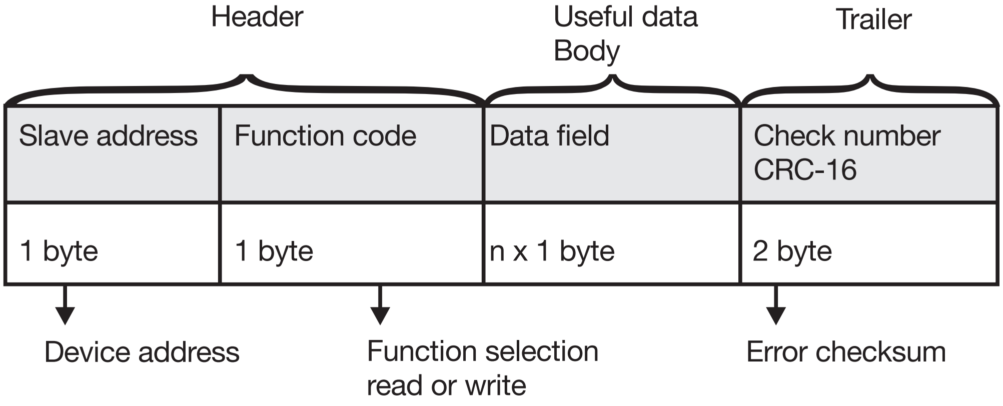
The network management acts as a general control agent over all the layers referred to here. With this, the user can access the services of the lower layers, and hence can analyze possible hardware and software errors very closely.

### 1.8.2 The data structure

**Telegram**

When a message is transmitted between two subscribers over the bus, the message can be a combination of very different types of data. For example, numerical values, texts, special characters or control signals can be transmitted. The most important aspect of the transmission procedure is that a uniform agreed format for the data to be transmitted is laid down in advance. Defining this format ensures that all subscribers understand one another, i.e. “speak the same language”. Such a format, meaning the framework of the message to be transmitted, is referred to as a telegram. The build-up of such a telegram is described, for example, through the bus protocol actually used in the individual field bus systems (see Chapter 2 “Important fieldbus systems”).

In principle, such a telegram can be split into three sub-areas, the **Header, Body and Trailer**. Fig. 48 shows the basic construction of such a data block, taking as an example the Modbus/Jbus format used by JUMO.



| Slave address | Function code | Data field | Check number<br>CRC-16 |
|---|---|---|---|
| 1 byte | 1 byte | n x 1 byte | 2 byte |

Fig. 48: Structure of the data block with Modbus

**Header** The telegram header contains protocol information that is transmitted before the useful data. This is made up of the destination address of the required bus subscriber (slave) and a function code that specifies whether data are to be read or written. With Modbus/Jbus, each subscriber has its own defined address. If the master places a message on the bus, all the slave listen in, but only the one whose address matches the destination address, actually processes the telegram.

**Body** The body contains the corresponding useful data. In the Modbus/Jbus example, this data field contains the information about word address, word count and the actual word value.

**Trailer** The trailer, the final or data security section, contains data that are attached to the useful data for validity checking or as an end of frame identifier. The transmitter calculates a test character in accordance with a set method (in this case, CRC-16), and this is then entered in the trailer. The receiver calculates this test character for itself, using the same method, and then compares it with the received test character. If the two agree, the telegram was transmitted without errors.

Within a network, the messages are normally transmitted in bit-oriented format. A command is generated in a application program and then converted to a function code or numeric code (see Fig. 49).

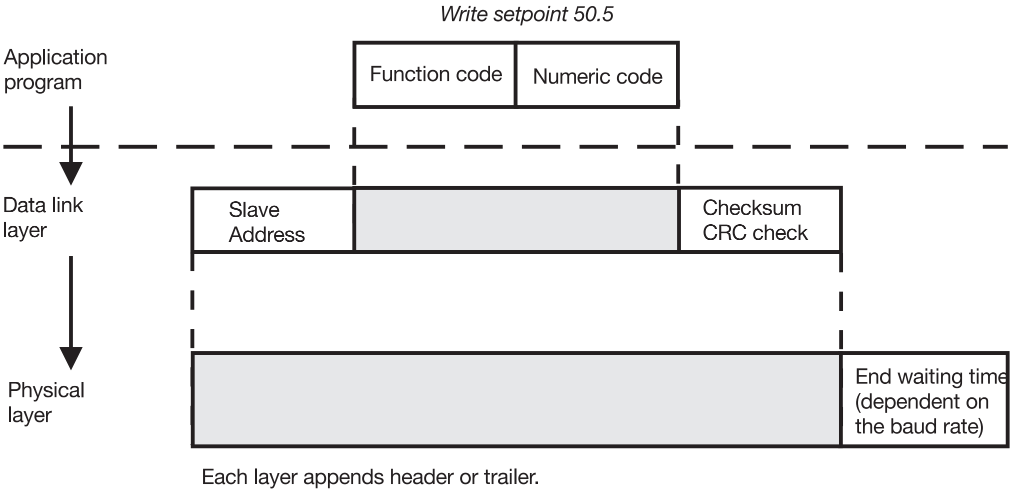


Fig. 49: Modbus: processing levels in the device (OSI model)

The message is forwarded to the data link layer. This layer calculates the checksum and sets the destination address. This telegram is then forwarded to the physical layer. Depending on the physical interface and protocol used, certain control characters (start/end identifiers) may also be added to the data block. This is known as setting the appropriate “flags”. In our example, there is simply a delay time at the end of the telegram, the length of which depends on the baud rate used. The signal is now transmitted and then stripped down again at the receiver, in the reverse of this sequence.

### 1.8.3 Error checking

**CRC**

When messages are routed, various methods can be used to check for errors in transmission. The methods vary according to the degree of complexity involved and security offered. Familiar methods are the parity check and the CRC (cyclic redundancy check).

The simple parity check has already been mentioned in Chapter 1.3.2. (see Fig. 50).

| Start bit | ASCII code for “J” | Parity bit | Stop bit |
|---|---|---|---|
| 1 | 1001010 | 1 | 00 |

**Parity**

Test for even parity: 1 + 0 + 0 + 1 + 0 + 1 + 0 = 3
3 + 1 (parity bit) = 4 even

Fig. 50: Error detection using parity bit with the ASCII code

In the following section we will deal briefly with the CRC method, also used by JUMO. This method is frequently used for checking entire message blocks. Here, all the useful data and a certain number of check bits are considered to represent the coefficients of a polynomial. This sequence of bits is divided by a “generator polynomial” (see Fig 51,). All communication partners must know this polynomial. Before the division takes place, the check segment places are filled with either “0”s (see example) or “1”s (Modbus).

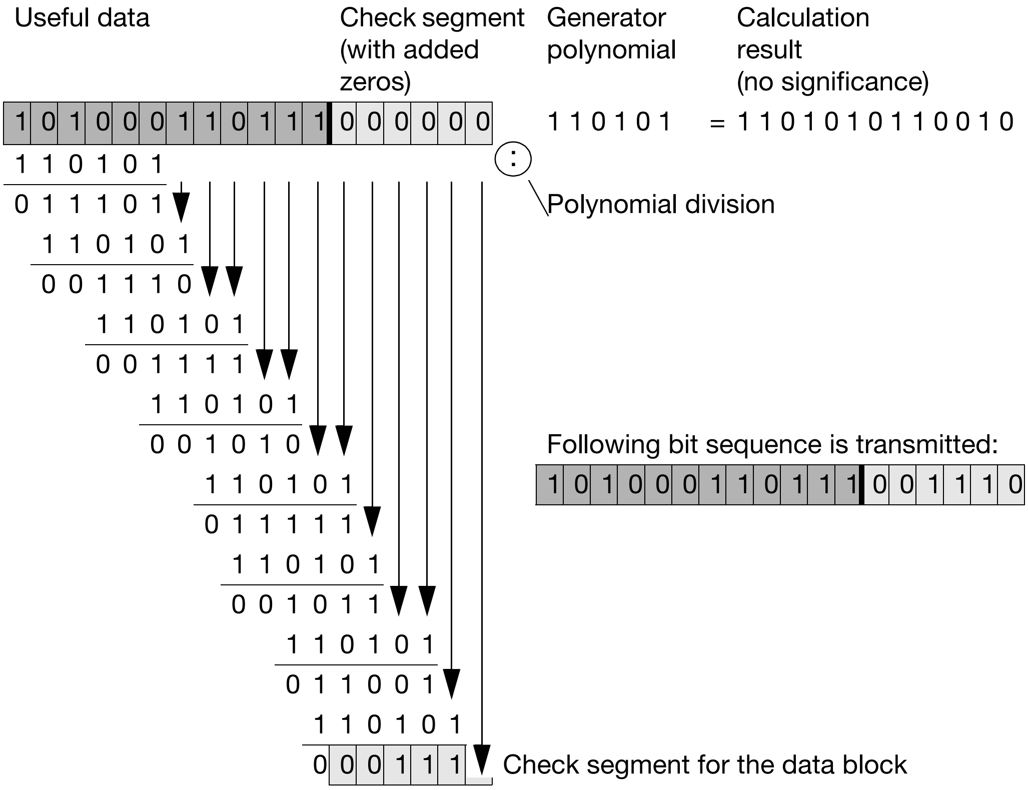


Fig. 51: Polynomial division to calculate the check segment with the CRC method

In practice, the division is normally carried out via exclusive-OR gates (half-adder). A detailed description of the method is given in the corresponding interface description for JUMO process instruments with Modbus/Jbus or in the Modicon Reference Guide.

The remainder of the division (not the result of the calculation) is then appended to the useful data as a check segment. Fig. 51 shows a simple but unrealistic example. Here, a 10-bit message is safeguarded using a 6-bit generator polynomial in the CRC method ($X^5 + X^4 + X^2 + 1$). The bit sequence in the check segment here comes to (001110).

Very often in practice, generator polynomials with a highest power of 16 are used. This is also referred to as the CRC-16 method, in which a 16-bit check segment is appended to the message. This method is used with Modbus as well. Here the CRC-16 checksum is appended to the data block as a two-byte check segment (see Fig. 48). In this case, the protocol-specific generator polynomial used equates to ($X^{16} + X^{14} + 1$ = 10100000000000001).

In the same way, the receiver then splits up the message, consisting of useful data and check segment, using the generator polynomial already known to it (see Fig. 52).

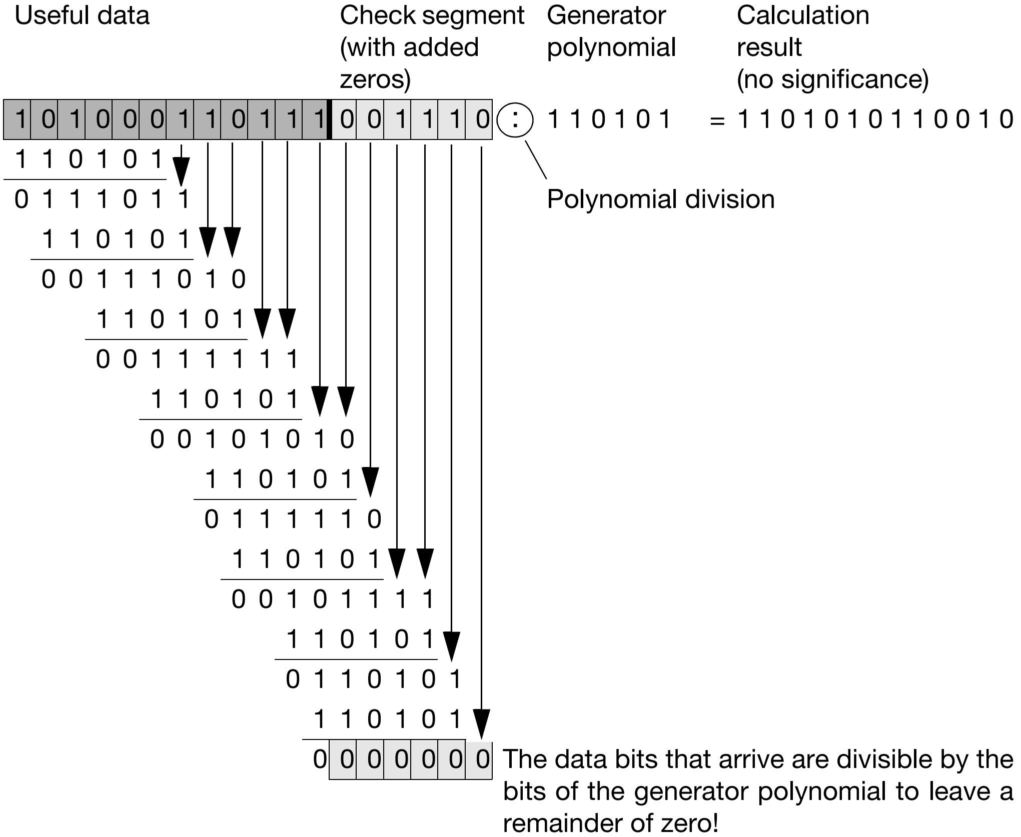


Fig. 52: Checking the message at the receiver

Because the transmitter has appended the remainder of the division as the check segment, the bit sequence received can now be divided at the receiver to leave no remainder. If this division results in anything other than a zero as a remainder, a transmission error has occurred.

### 1.8.4 Connection of networks via repeater, bridge, router and gateway

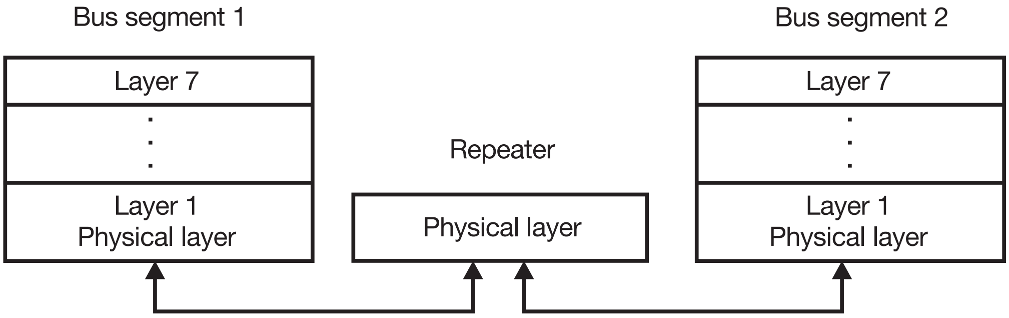
When we looked at physical digital interfaces and network structures (topologies) earlier, it was clear that only a limited range was possible. For example, the maximum range allowed by the RS422 or RS485 interfaces is only 1200m, and this is insufficient for certain applications. Another application may call for a connection between individual sub-networks via different communication levels.

Inevitably, because of all these requirements, the need arises for special network elements. Examples of these network elements are the repeater, bridge, router or gateway, and they will be described briefly below.

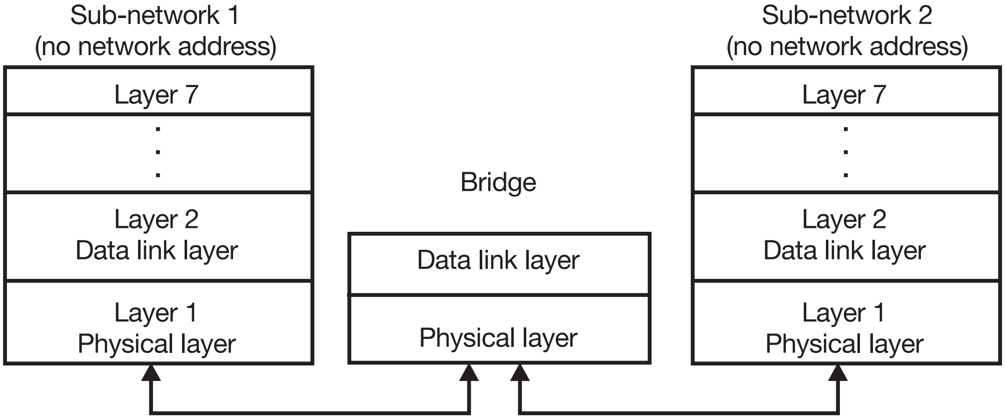
**Repeater**

The repeater is defined as a device that works as a type of amplifier to regenerate signals, so that the bus topology can be extended. It should be noted here that there are non-regenerative as well as regenerative repeaters. These digital amplifier stations can copy all the bits that arrive from one bus segment, into another segment. It therefore connects different networks at layer 1 (see Fig. 53 a). The important thing here is that networks/bus segments in question must use the same protocol. The maximum length of a network segment can therefore be increased using a repeater.

**a) Repeater**



**b) Bridge**



**c) Router**

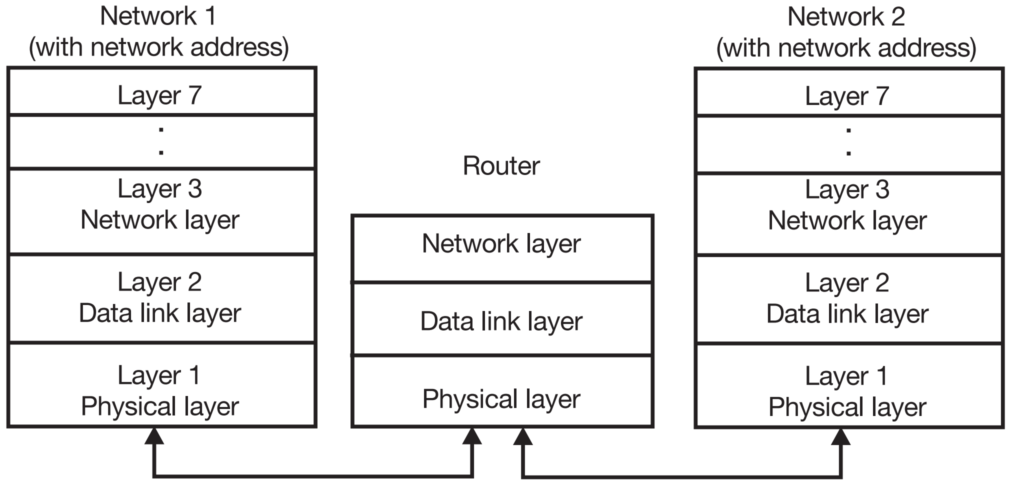


Fig. 53: Structure of a repeater, a bridge or a router

The advantage of repeaters is that they are independent of the network protocol, and are available for a wide variety of media, such as fiber-optic cable.

Non-regenerative repeaters merely pass the signal on, without amplifying it. For this reason, both the increase in range possible and the maximum number of repeaters that can be used are limited.

Regenerative repeaters boost the transmitted signal and remove any possible superimposed interference. Theoretically, the maximum number of these repeaters that can be used is unlimited, however in practice it is limited by the dynamic response on the bus. In other words, as the length of the transmission medium increases, more time is taken up for the data transmission. However, a slave has to reply to the master within a certain fixed time window, otherwise it is signed off. Problems can arise from this, when the number of repeaters installed exceeds a certain number. As well as the time aspect, the maximum number of repeaters must also comply with the protocol used and the topology form.

**Bridge**

The bridge connects two similar sub-networks together, but does not have a network address of its own. Here, only networks with the same protocol and topology form can be connected together. In this case, the connection of the sub-networks occurs via the MAC level of layer 2 of the OSI reference model (see Fig. 53 b). A bridge stores complete frames of data blocks, and forwards them to its data link layer for validation checking.

Nowadays, a distinction is drawn between the **remote bridge**, which connects widely separated networks via fixed lines or satellites using modems, and the **local bridge**, which only connects together geographically adjacent networks.

It can be a good idea to use a bridge, for instance, in an overloaded network, so that the network is split into two networks with equal activity levels. If a suitable separation point is chosen, the traffic over the bridge is not very high, and each of the two remaining networks now has only to handle half the load. This is a form of logical decoupling of network segments. Another reason for using a bridge is the increased network reliability, as the effects of an interference source are not noticeable beyond the bridge. Furthermore, the bridge also offers all the advantages mentioned earlier for the repeater.

**Router**

The router has an individual address and likewise can connect two similar sub-networks together. Such devices are necessary when data packets have to pass through a number of networks or nodes with a number of branches. The router's job is to transmit data via such branches as far as the destination network.

Connection of the network segments occurs via layer 3 (network layer) of the OSI reference model (see Fig. 53c). Each sub-network receives its own network address and each subscriber its own subscriber address. Also, each router is regarded as a type of subscriber and receives its own address. The transmission of data to the destination address is left up to the router with its router protocol of layer 3. Here it is quite possible for the data to be passed on via several routers.

**Broadcast**

One advantage of a router is the distribution of broadcasts. Broadcasting here means sending a message to all connected stations in a network, where no reply to the message is required. Routers are equipped with a measure of intelligence, which is concerned with finding the optimal route through the entire network. If there is a large number or an increased number of data sets queuing up, the router is also in a position to share them out over various routes. This therefore gives rise to a very efficient utilization of the cable routes.

The router has one slight disadvantage however, in that it is limited to one protocol, exactly like the other exchange devices. Connection of networks with different protocols necessitates the use of devices that carry out a protocol conversion. The gateway is an example of such a device.

**Gateway**

The gateway connects two or more dissimilar networks and, in so doing, takes on the protocol conversion. Hence, it is in a position to connect together different network architectures.

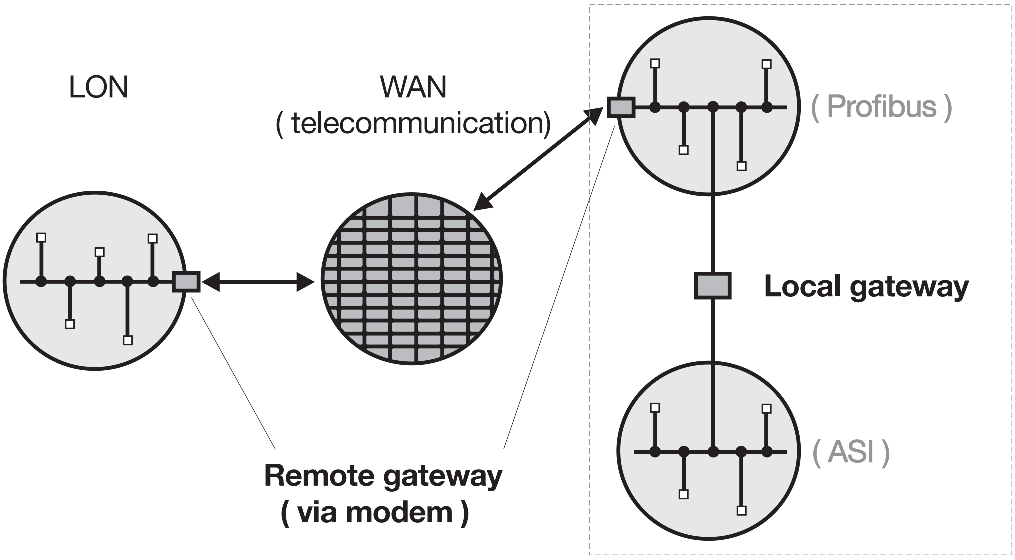


Fig. 54: The gateway

Once again a distinction can be drawn between the **local gateway**, which, within a company, for example, deals with the data exchange from the field level up to the company control level, and the **remote gateway**, which connects different networks over longer distances, via a modem attachment (see Fig. 54).

In general terms, it could also be said that the gateway is an assembly of hardware and software, that facilitates inter-network communication. Here, different demands are imposed on the router, depending on the application, and these must be met on an individual basis.

## 1.9 Operation through application programs

Some of the major advantages of digital field devices are the diagnostic possibilities in the event of a fault, and the facility for programming, communication and data exchange over longer distances using special terminals or the PC. The disadvantages for someone using devices from various manufactures are the different ways of using the process devices and the different concepts for operation, parameter setting and configuring.

Just as there is a wide variety of standards with fieldbus systems, so the operator has to work with many different concepts with the application programs as well. It would be a mistake for a user to think that, once a fieldbus standard has been decided on, he would only have to work with one unified user interface, provided that the process devices chosen were compatible with this standard. This is even the case when one considers the OSI reference model with its seven different layers. Although this model safeguards communication between the individual subscribers, it does not concern itself with the use of data which leaves via layer 7. However, it is precisely the application and handling of data via suitable software which is of interest to the user. Here, the user could be said to be operating in a form of “layer 8”, a layer that is no longer defined by the OSI model, because instrument functions that no longer belong to the application layer are determined here.

**Interoperability**

Attempts to find a unified solution here are handicapped by the wide variety of process devices with very different functionalities, and the varying demands of the user. Although all manufacturers follow certain guidelines, they tend to go their own way when converting their applications, as, quite naturally, they would not want to lose their individual identity. This means that problems will arise here if the user would like to operate a process controller from manufacturer X with a controller from manufacturer Z on the same bus, or exchange one device for another. Hence, for the moment the desire for “interoperability”, as it is known, remains mostly just a thought for the future. Interoperability is the capability of integrating products from different manufacturers in flexible, functional systems, without having to develop customized hardware, software or tools.

**DDL**

Each device (each software) must know how it should edit or process the individual data. To establish a certain guideline here, a data description language has been developed, for example, in the field of measurement, known as DDL (device description language). It describes individual device parameters with the help of data description codes, and is used a great deal with smart transmitters or process instruments.

**GMA guideline**

Another guideline concerned with the definition of unified operating concepts is, for example, the GMA guideline VDI/VDE 2187 (unified PC display and user interfaces for digital field devices, issued in 1993). This lays down specific rules for the PC operation of field devices capable of communication (sensor and actuator systems).

### 1.9.1 Configuration software (setup program)

Modern microprocessor instruments have become extremely complex, and programming them from the front of the instrument is correspondingly laborious. When the instrument is operated from the front, it is very difficult to provide a comprehensive picture of the current configuration data of the complete instrument. Should an instrument develop a fault, it would be an expensive process if the configuration data for the replacement instrument had to be input manually.

For this reason, PC application programs offered by the individual manufacturers for their automation devices are used for easy configuration of digital process devices. However, these programs are normally manufacturer-specific.

JUMO offers setup programs for configuration of its microprocessor instruments. Setup programs offer the advantage that the configuration data of the device can be read into the PC and displayed clearly. The configuration data of a device are stored in a file, and can be used to similarly configure additional devices.

Each device (controller, recorder, transmitter, etc.) that can be programmed using a setup program, has a 4-pin connector as standard, so that it can be connected to the communication port (COM1 or COM2) of a PC. This arrangement uses a simple TTL interface, where the appropriate voltage level for the data transmission is generated in an interface. As an option, the setup connection can also be established via a PC card (RS422 or RS485) integral to the device (see Chapter 3 “Organization of the data system for JUMO”). A protocol specific to JUMO is used as the data structure in this case.

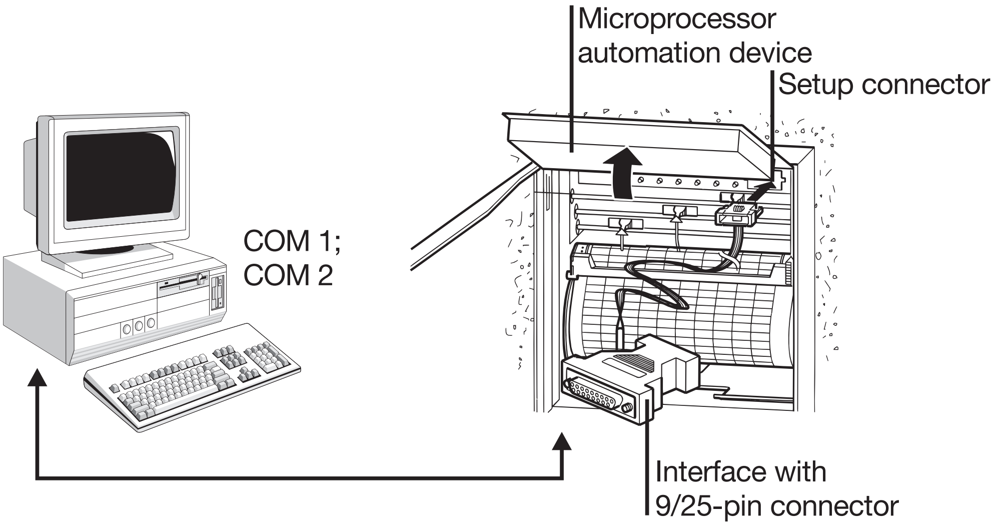
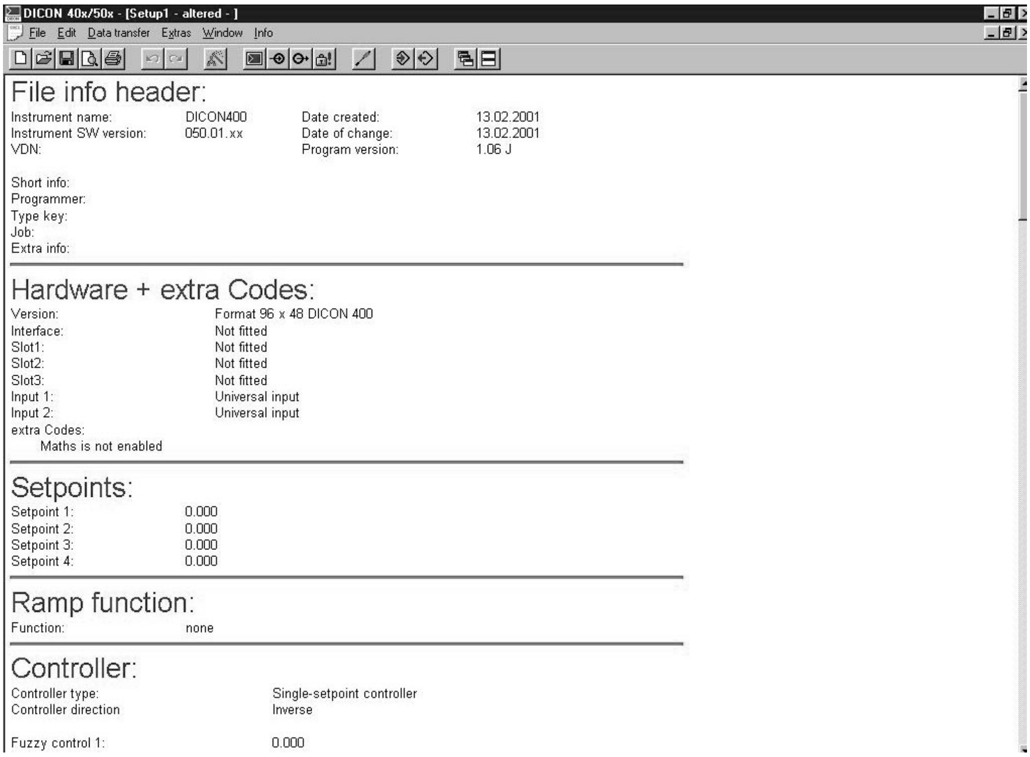

Microprocessor automation device
Setup connector
COM 1;
COM 2
Interface with 9/25-pin connector

Fig. 55: Connection of a microprocessor instrument with a PC via the setup interface

Whereas in the past setup programs ran under the MS-DOS operating system, programs nowadays run under Windows 95, Windows 98 and Windows NT. The menu structure of the programs corresponds to the accepted systematics for MDI (multi-document interface) programs running under Windows.

The user interface under Microsoft Windows has developed as a quasi-standard in industrial automation engineering as well. One of the reasons for this is undoubtedly that many application programs are offered for the MDI architecture, and the user is familiar with the workings of this type of program.

JUMO offers software products developed in accordance with the Windows style guideline. With JUMO, when such programs are installed, a new program group is started up under the name “JUMO instruments”. This group contains the individual setup programs of the different devices. When a program is started, the user sees a program window with all the important elements, that, for the most part, he is already familiar with and knows how to use. As well as configuration of connected devices, newer programs from JUMO also offer the facility of checking inputs, outputs and displays etc., using an “adjust and test” mode.



File info header:

| | | | |
|---|---|---|---|
| Instrument name: | DICON400 | Date created: | 13.02.2001 |
| Instrument SW version: | 050.01.xx | Date of change: | 13.02.2001 |
| VDN: | | Program version: | 1.06 J |

Short info:
Programmer:
Type key:
Job:
Extra info:

Hardware + extra Codes:

| | |
|---|---|
| Version: | Format 96 x 48 DICON 400 |
| Interface: | Not fitted |
| Slot1: | Not fitted |
| Slot2: | Not fitted |
| Slot3: | Not fitted |
| Input 1: | Universal input |
| Input 2: | Universal input |

extra Codes:
Maths is not enabled

Setpoints:

| | |
|---|---|
| Setpoint 1: | 0.000 |
| Setpoint 2: | 0.000 |
| Setpoint 3: | 0.000 |
| Setpoint 4: | 0.000 |

Ramp function:

Function: none

Controller:

| | |
|---|---|
| Controller type: | Single-setpoint controller |
| Controller direction | Inverse |
| Fuzzy control 1: | 0.000 |

Fig. 56: Setup program of the DICON 40x/50x programmable controller

In addition to the setup programs mentioned, JUMO also offers setup programs with the VDE/VDI 2187 interface for sensors in the chemical industry.

With these programs, the user sees two interfaces (windows):
General settings such as the selection of the interface are made in the main window. Measurements can be displayed and logged here, and diagnostic checks made on the device. The configuration of the device takes place in the device window.

This standard deviates somewhat from the “JUMO standard”, however those working in these branches of the industry meet familiar features, which makes it easier to work with these programs.

In the past, JUMO has supplied setup programs that made it possible to display a limited number of process variables during commissioning. Whereas until now, process variables had to be observed and/or logged by a recorder during some commissionings, there will be no need for this in future, as all the recordings can be made by PC.

### 1.9.2 Project design software

When considering a complete installation or a complete system consisting of several components (process devices) that exchange data with one another, it is not sufficient merely to configure the devices for their tasks. A suitable automation system and network must be designed for such communication.

Project design software such as the JUMO mTRON-iTOOL is used for this application. With this software package, for example, the JUMO mTRON distributed automation system can be graphically designed on the PC (see Chapter 3.4 “JUMO instruments with LON”). Nowadays, such software packages are also very often integrated in the Microsoft Windows user interface. This is transparent to the user, who is already familiar with the menu structure, which is very much oriented towards standard Windows architecture. Using this program, the project designer is able to link individual components together, and in the example shown, even configure application-specific parameters using an embedded setup program, and then load the project into the automation system (see Fig. 57).

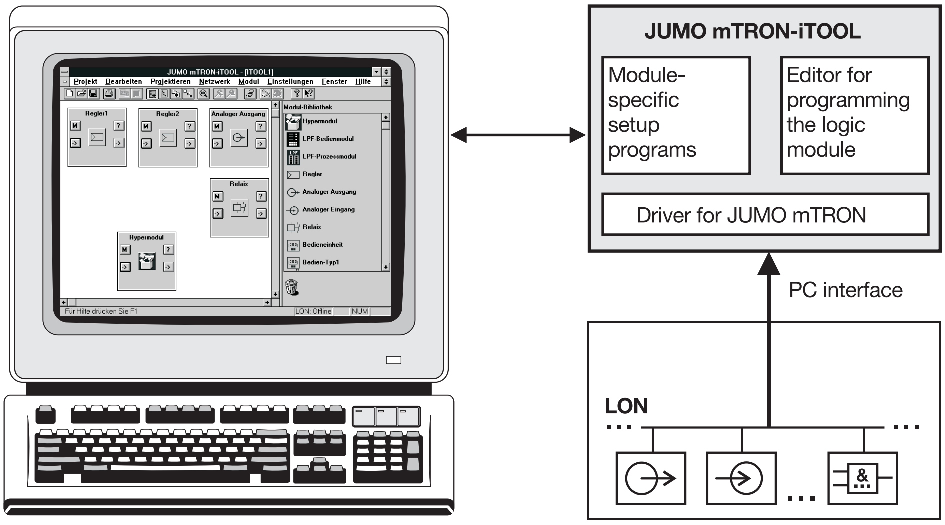


Fig. 57: JUMO mTRON-iTOOL project design software

For the user, the use of this type of software package results in the following advantages:

- The individual process modules are held in a library and can be freely positioned on the design sheet as graphic objects using a "drag and drop" technique.
- Any name can be assigned to these objects, so that the name bears a direct relationship to the real installation.
- Integrated device-specific setup program within a user interface.
- When the plant is operational, certain dialog windows can be used to display the values of network variables online. This allows the design to be checked on the real installation.
- A project navigator allows large projects to be split into smaller structured projects, so that the user still retains an overview.

### 1.9.3 Measurement display and operation using visualization/evaluation software

Efforts are also being made in the area of measurement display, to lay down, through specific guidelines (GMA guideline), how a user interface must appear, or how it should be partitioned, as in the Windows programs already mentioned, where the individual menu items are almost totally uniformly structured. Here, for example, it is recommended that the individual measurements to be visualized, and the data for parameter setting and configuration are displayed in different levels, which can then be allocated different access rights using passwords. These are certain general requirements that are normally clearly set down by the user in the specification for the application.

In this field, there is a large number of different software packages on the market. In many respects, these packages are very similar in their functionality, depending on the area of use. In most cases, the packages provide software tools with which the user (client) can either design his specific application himself, or have it designed by a service provider (software house), to his specification.

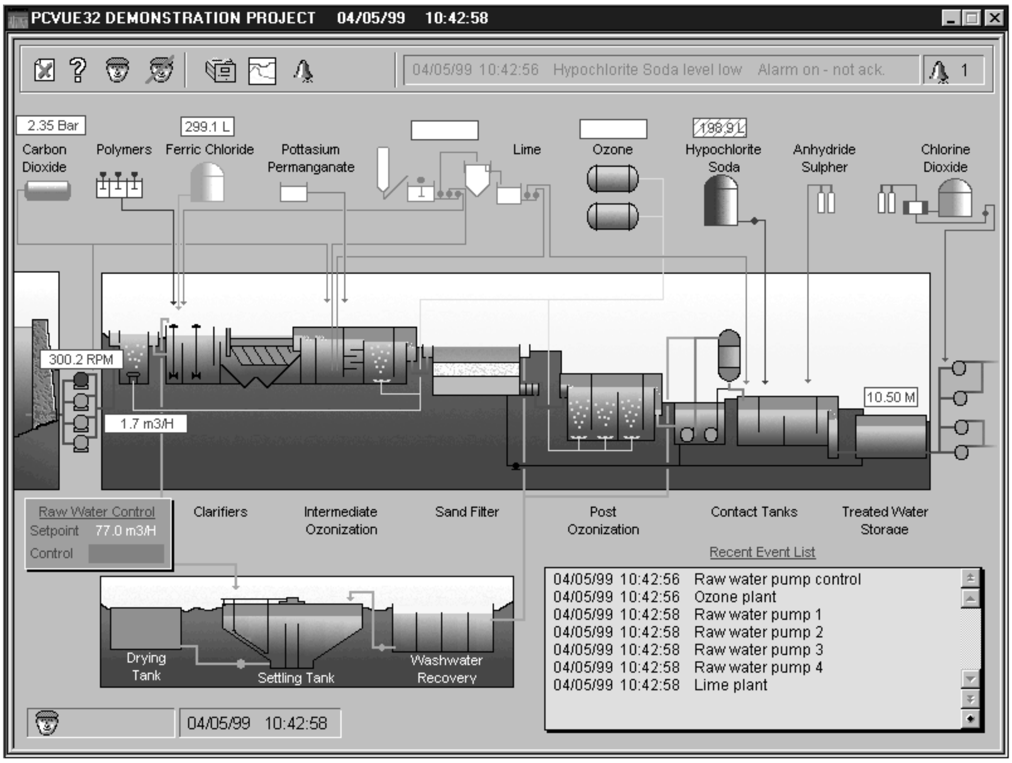
**SCADA programs**

SCADA programs (Supervisory Control and Data Acquisition) are mainly used for centralized operation, management and observation of processes. Normally there is a flow diagram where the technical installation is represented in graphical format as a block schematic. Such a process diagram contains all the apparatus, machines, and measurements necessary for the process, as well as major material flows (see Fig. 58).

The most important tasks and advantages of such user software in the field of automation can be described as follows:

- Display of all important process variables and faults
- Centralized user-friendly operation of the installation directly from a PC
- Simplified process management through the combination of individual operating instructions accessed via recipe functions;
  avoidance of operator error
- Communication with process devices from different manufacturers via suitable driver software

- Independent generation of manufacturing records (batch reports) to accompany products for quality assurance etc.
- Clear representation of measurement changes through trend diagrams etc.
- Possibility of historical trend derivation



PCVUE32 DEMONSTRATION PROJECT 04/05/99 10:42:58

04/05/99 10:42:56 Hypochlorite Soda level low Alarm on - not ack. 1

Raw Water Control
Setpoint 77.0 m3/H
Control

Recent Event List

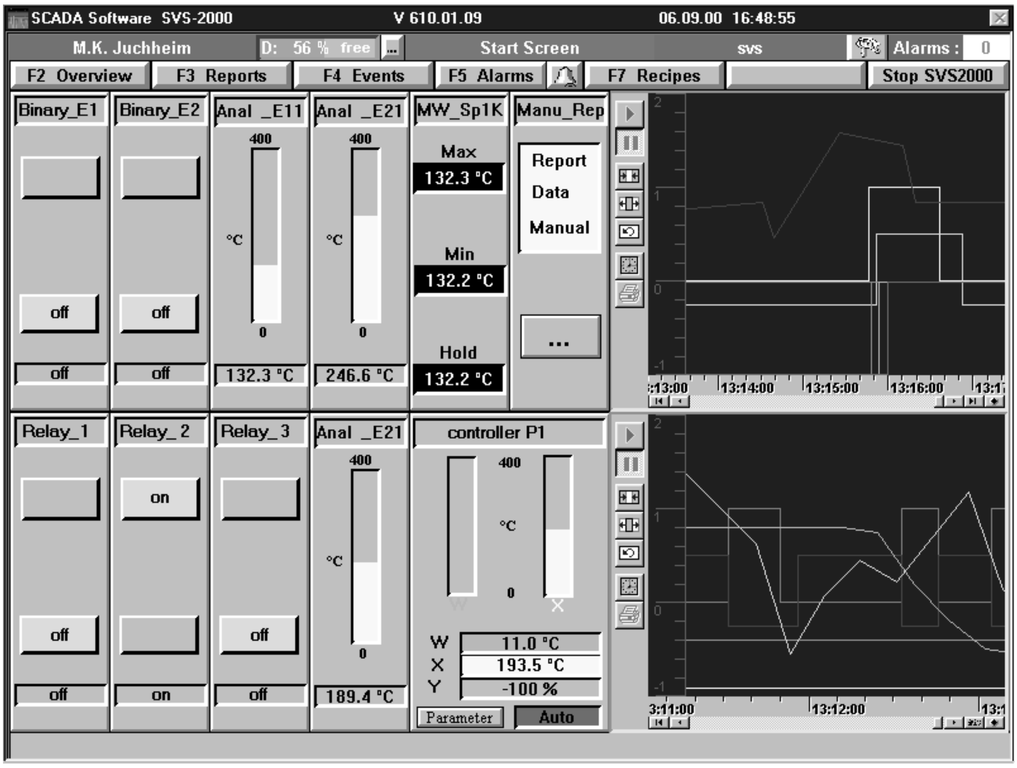
```
04/05/99 10:42:56  Raw water pump control
04/05/99 10:42:56  Ozone plant
04/05/99 10:42:58  Raw water pump 1
04/05/99 10:42:58  Raw water pump 2
04/05/99 10:42:58  Raw water pump 3
04/05/99 10:42:58  Raw water pump 4
04/05/99 10:42:58  Lime plant
```

04/05/99 10:42:58

Fig. 58: PCVUE process visualization software

JUMO is one of the firms that supply the 32-bit based process visualization software program, PCVUE 32. This was developed using Microsoft software technologies, and follows the user interface concept and security method developed for Windows NT.

PCVUE 32 has its own graphical interface in which the graphical representation of the application is generated.



Max 132.3 °C

Min 132.2 °C

Hold 132.2 °C

Anal _E11 132.3 °C

Anal _E21 246.6 °C

Anal _E21 189.4 °C

W 11.0 °C

X 193.5 °C

Y -100 %

Fig. 59: Group diagram for furnace monitoring using SVS-2000 process visualization software

The SVS-2000 process visualization software was developed based on PCVUE 32, to simplify the visualization and operation of JUMO instruments using a process visualization software program. The SVS-2000 makes it possible for the user to call up instruments manufactured by JUMO, such as recorders, controllers, transmitters, indicators and mTRON modules, in the configuration level of the SVS-2000, and then to link these in a group diagram in the parameter level (see Fig. 59). Hence, the SVS-2000 process visualization software is ideal for a fast design of applications using JUMO instruments. As the software is based on PCVUE 32, an application designed using the SVS-2000 can be expanded to include devices from other manufacturers (see Fig. 60).

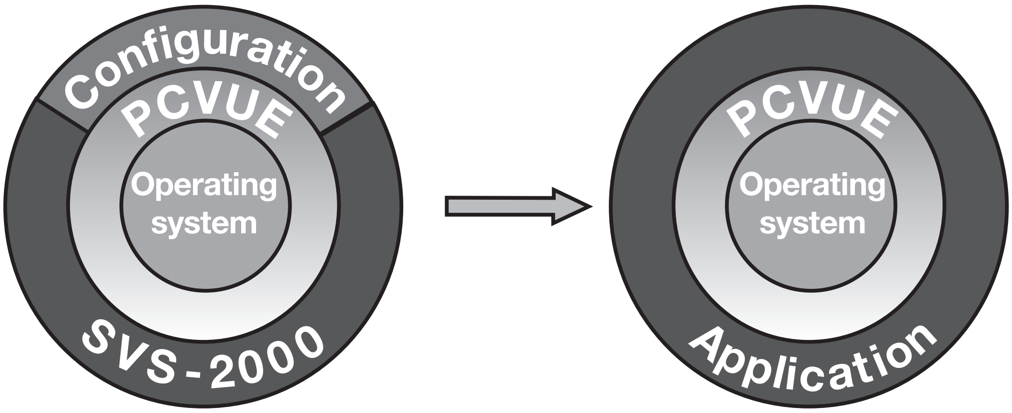
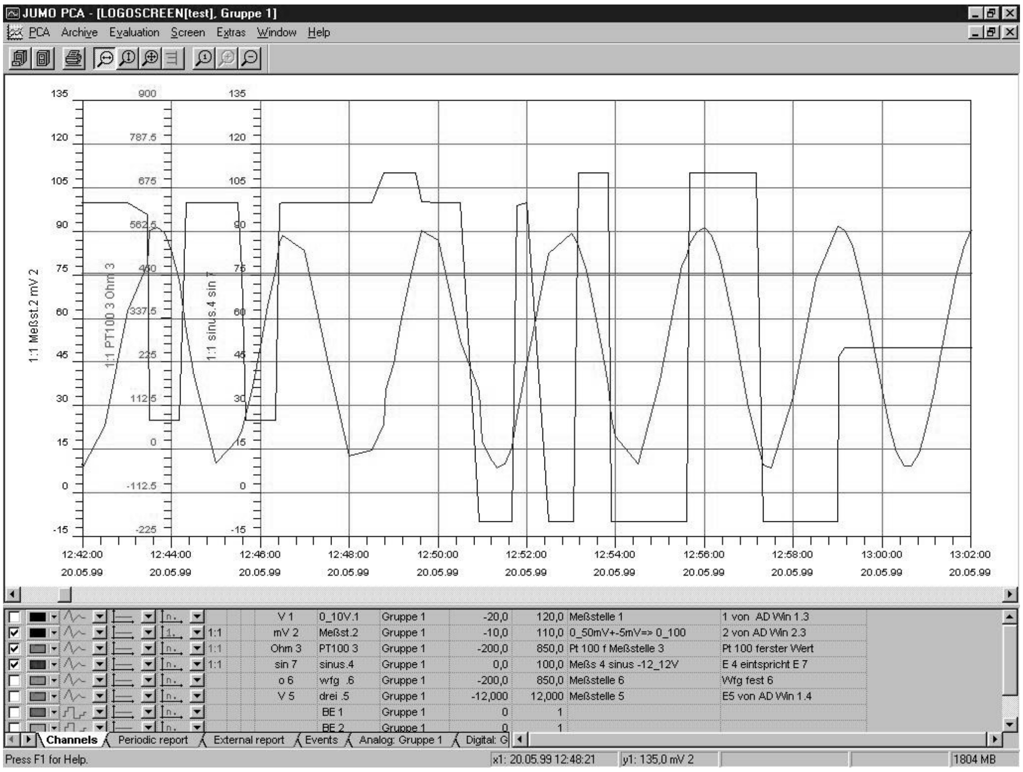


Fig. 60: PCVUE-SVS-2000

**Evaluation software**

With the PC evaluation program (PCA), JUMO offers a program that runs under Windows 95/98 or Windows NT, and can be used to manage, archive, visualize and evaluate the data from the LOGOSCREEN screen recorder.



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| □ | | V 1 | 0_10V.1 | Gruppe 1 | -20,0 | 120,0 | Meßstelle 1 | 1 von AD Win 1.3 |
| ☑ | 1:1 | mV 2 | Meßst.2 | Gruppe 1 | -10,0 | 110,0 | 0_50mV+-5mV=> 0_100 | 2 von AD Win 2.3 |
| ☑ | 1:1 | Ohm 3 | PT100 3 | Gruppe 1 | -200,0 | 850,0 | Pt 100 f Meßstelle 3 | Pt 100 ferster Wert |
| ☑ | 1:1 | sin 7 | sinus.4 | Gruppe 1 | 0,0 | 100,0 | Meßs 4 sinus -12_12V | E 4 eintspricht E 7 |
| □ | | o 6 | wfg .6 | Gruppe 1 | -200,0 | 850,0 | Meßstelle 6 | Wfg fest 6 |
| □ | | V 5 | drei .5 | Gruppe 1 | -12,000 | 12,000 | Meßstelle 5 | E5 von AD Win 1.4 |
| □ | | | BE 1 | Gruppe 1 | 0 | 1 | | |
| □ | | | BE 2 | Gruppe 1 | 0 | 1 | | |

Press F1 for Help. x1: 20.05.99 12:48:21 y1: 135,0 mV 2 1804 MB

Fig. 61: Evaluation of screen recorder data with PCA

- The data from different screen recorders are stored in an archive database
- Any channels of a device can be brought together in PCA groups
- Using an export filter, the data can be exported into other programs, such as Microsoft Excel
- The data can be read out from the screen recorder either via the serial port or using a diskette
- The evaluation program supports the networking capability, which means that several users can obtain data from the same database in the network, independently of one another.

The original purpose of a fieldbus was to be used for networking at the field level, as a simple means of connecting all the sensors and actuators to the network. Nowadays a fieldbus is not only used for the attachment of sensors, actuators and other devices at the field level, but also for communication at group and process/plant control levels. The components are mostly connected by a 2-core cable.

A fieldbus system must be based on certain requirements, such as:

- high noise immunity
- a short reaction time, appropriate to the application
- unambiguous and simple diagnosis
- uncomplicated installation and low costs
- simple configuration and programming

In addition, the connection of devices to the bus must not have any repercussions on other bus subscribers, and the failure of individual bus devices must not affect its operation. For process technology applications it is also desirable that the fieldbus is implemented as “intrinsically safe”. In other situations there is a requirement that the supply for the sensors and actuators at the field level is provided through the bus.

Compared with the previous analog technology, the introduction of a fieldbus into automation engineering also brings financial advantages, through:

- lower installation costs for the signal/data transmission system, through
  - linear cabling on site, instead of individual wiring from the control center
  - fewer cable terminals, distribution panels are completely unnecessary
  - fewer instruments, thanks to multifunctional usage
- savings in everyday operation, in planning, realization and documentation of alterations and the recording of production data

**Total costs of ownership**

Naturally, field devices cost more for the initial purchase than analog devices. In some application, these higher prices may even cancel the price advantage through the lower installation costs of the signal/data transmission system. But the savings in daily operation are enormous, and must also be considered when making the decision between analog technology and a bus system. It is the total costs of ownership that have to be considered.

Before deciding on a bus system, the system planner must face the following questions:

- can the job be done with the bus system?
- is the data transmission fast enough?
- is the bus system the most economic one for the application?
- does the bus system work compatibly with existing components, or can this compatibility be achieved?
- is transmission reliability and noise immunity adequate for the application?
- can the network be expanded to the size required?
- is it easy to extend the system?

- are all the required components available with this bus interface?

At present, there are over 50 different fieldbuses on the market, which can be taken seriously, that have implementation or standardization recommendations and/or products available. These available fieldbuses either meet national standards, or are in effect industry standards.

An internationally standardized fieldbus would be very welcome for automation in process engineering, one which could replace the conventional point-to-point wiring of field devices in analog technology. This highly desirable fieldbus would have to fulfil all the requirements of the various industrial sectors and be able to connect all field devices together, regardless of the manufacturer. However, since company interests lie behind almost every fieldbus, and the various sectors make extremely varied demands on a bus system, this ideal international fieldbus for all applications will probably never exist. These days, people are instead trying to develop simple communication mechanisms for differing bus systems.

This chapter will therefore provide an abbreviated explanation of some of the most important fieldbus systems and their typical characteristics.

## 2.1 HART communication

**HART protocol**

The basic idea for this technique is that a single cable pair is used to transmit digital communication signals, as well as the analog 4 — 20mA signal. HART stands for “Highway Addressable Remote Transducer”, and is a registered name of Rosemount Inc.

HART technology is largely standardized, and offers the advantage that just one universal operating/control instrument is able to operate HART devices from various manufacturers. This is made possible by use of defined commands which are the same for all HART devices.

HART is suitable for the use in areas with an explosion hazard (Ex areas).

HART is already spread around the world, and receives global support from the HART Communication Foundation (HCF) that was set up in 1993.

**FSK method**

The communication method for HART is based on FSK (Frequency Shift Keying). The proportional 4 — 20mA current in the 2-wire cable represents the measured value, and the communication signal is overlaid as an pure AC voltage (with no DC component) with a low amplitude (500mV pk-pk). Logic “0” = 2400Hz and logic “1” = 1200Hz. The digital communication channel enables transmission of not only the primary measurement, which is that represented by the analog signal, but also other measurements and instrument parameters.

| | |
|---|---|
| Manufacture/<br>user organization | Rosemount Inc. /<br>HART Communication Foundation (HCF) |
| Medium | 2-wire cable 4 — 20mA |
| Subscribers | max. 15 (multidrop operation) |
| Data transmission rate | 1200 bit/sec |
| Transmission distance | max. 2000 m |

Table 10: Summary of HART technology

HART devices are usually operated with point-to-point connections, or in a star configuration using appropriate field multiplexers.

**Multidrop operation**

Bus operation is also possible with HART: in *multidrop* operation up to 15 devices can be connected in parallel to a 2-core cable and attached to an automation system. Here a bus address is assigned to each device, and the analog signal is set to a fixed value of 4 mA, which is just the supply current for the devices. Information is now only exchanged in digital form, through the FSK signal. The master (up to 2 masters are permitted) requests the values from the devices one after another. At this point it should be mentioned that HART communication can transmit 1-2 telegrams per second. If the maximum number of 15 HART devices is attached to the bus, one polling cycle takes several seconds! HART communication is generally (especially in multidrop operation) not exactly suitable for operating and controlling a system, but rather for setting parameters and monitoring measurements.

**Communication** The first and simplest step is communication using the appropriate HHT (hand-held terminal) for setting parameters, transferring status information and so on, or a connection to a PC.

The HART protocol supports levels 1 and 2 of the OSI reference model. The communication structure corresponds to the master/slave procedure, whereby the telegrams can be fairly long. HART communication is not very suitable for transmitting process-control measurement data, but is very suitable for conveying status information.

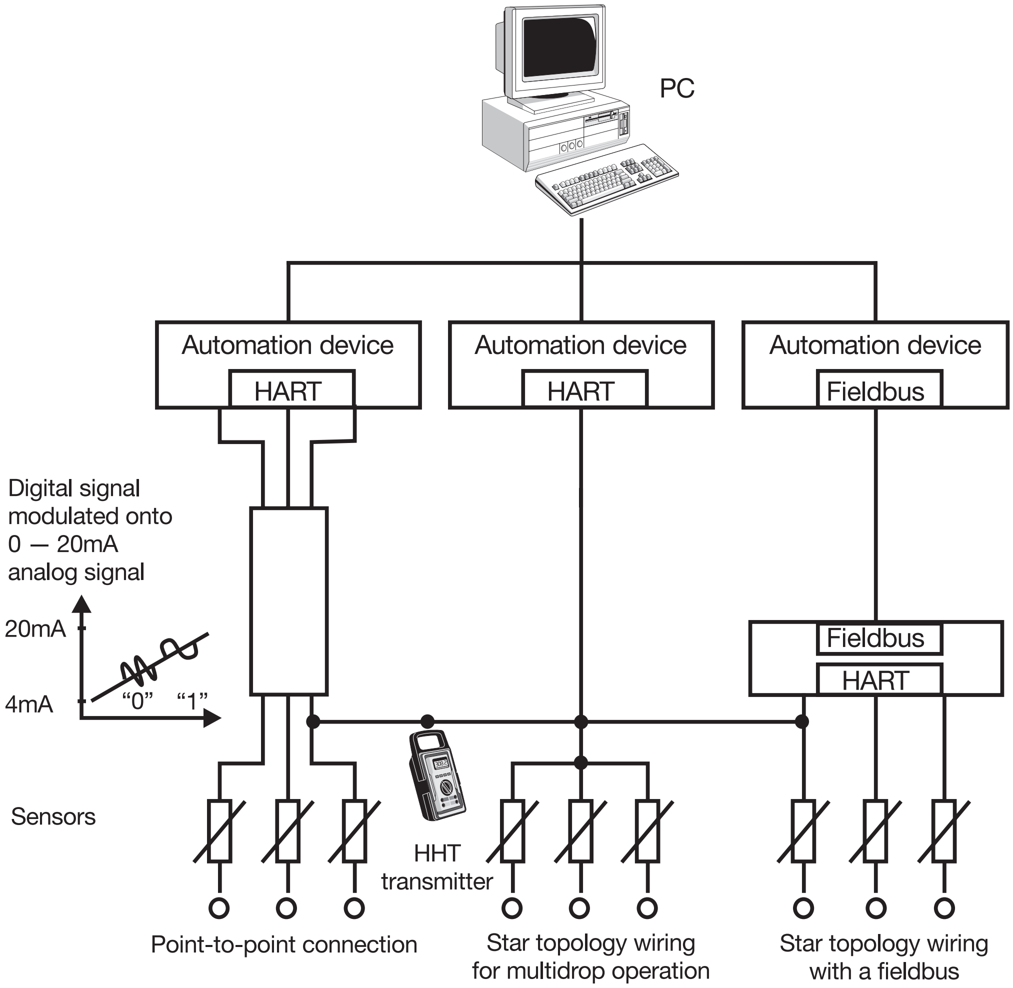

PC
Automation device
HART
Automation device
HART
Automation device
Fieldbus
Digital signal modulated onto 0 — 20mA analog signal
20mA
4mA
"0"
"1"
Fieldbus
HART
Sensors
HHT transmitter
Point-to-point connection
Star topology wiring for multidrop operation
Star topology wiring with a fieldbus

Fig. 62: Possible communication paths with the HART protocol

Figure 62 shows possible communication paths using HART:

With a *point-to-point connection*, the cable from each sensor is led to an automation device.

With a *star-topology wiring*, using multidrop, all sensors are connected in parallel and fed by a constant 4mA current. The measurement can only be evaluated through the digital signal.

With *star-topology fieldbus wiring*, the HART signals are processed on the spot and integrated into a different fieldbus protocol. The transmission to the automation equipment is thus made through this other fieldbus.

## 2.2 ASI bus

The AS (actuator-sensor) interface does not attempt to replace a complete control system, but just provide a simple method of connecting sensors and actuators at the lowest field level. Transmission is limited to a simple binary state information ("On" + "Off"). It is therefore not very suitable for more complex systems where several items of data must be communicated, but it is suitable for simple tasks, such as measuring filling levels or pressure sensing. The system has a master/slave structure, and all activities are initiated by the master.

| | |
|---|---|
| Manufacturer/<br>user organization | a group of several sensor/actuator manufacturers (Festo, ifm, Pepperl+Fuchs, Siemens...)/<br>ASI Verein e.V. |
| Medium | unshielded 2-core cable (ASI-specific) |
| Subscribers | max. 31 |
| Data transmission rate | 167 kbit/sec |
| Transmission distance | max. 100 m |

Table 11: A summary of the ASI bus

**ASI bus**

The full implementation of an ASI system includes a master and 31 slaves, whereby each slave can handle up to 4 binary state devices (input or output) – see Fig. 63. The slaves are connected to each other and to the master by a 2-core cable that is used to supply power (up to 100mA per slave) as well as for the data transmission. The cycle time for handling 31 slaves is approx. 5 milliseconds.

The ASI bus is a simple, economic system which can be used to replace previous I/O cable looms.

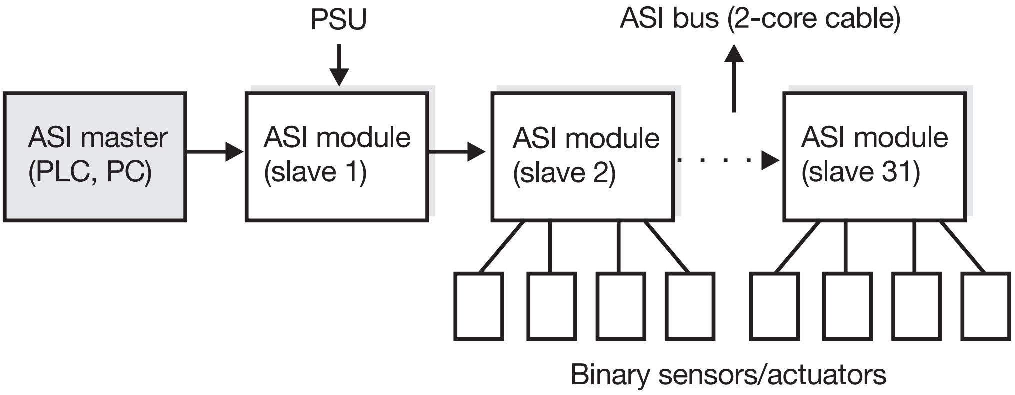


Fig. 63: Basic layout of an ASI system

## 2.3 Bitbus

**Bitbus**

Bitbus is a master/slave system that has been optimized for the field level. The fundamental task of the Bitbus is the exchange of information over a robust network within a manufacturing and production process. Its main application is therefore networking at the process computing level.

It is a serial fieldbus system which was developed by Intel in 1984. It has been specified in an international standard (IEEE 1118) since 1991. Layers 1 and 2 of the OSI model are clearly covered by the standard. Layer 7, the application level, takes on some of the tasks of OSI layers 3-6. The transmission protocol is defined here, amongst other items.

The system has now been applied in industry for several years, and is widespread in manufacturing technology, with the emphasis on networking of robotic equipment. Support is provided by the Bitbus European Users Group (BEUG). This organization is also supported by Phoenix Contact, which offers Bitbus under the name of Interbus-C.

| | |
|---|---|
| Manufacturer/ user organization | Intel (IEEE 1118) /<br>Bitbus European Users Group (BEUG)<br>from Phoenix Contact as Interbus-C |
| Medium | RS485 |
| Subscribers | max. 250 |
| Data transmission rate | 62.5 kbit/sec to 2.4 Mbit/sec |
| Transmission distance | 30 m to 13.2 km |

Table 12: A summary of Bitbus

**Polling procedure**

Network access is regulated by the master, through a polling procedure. The hierarchy can cover several levels (Fig. 64), by linking a slave from a high level with the master at the next level down, but the software for the operating system only supports one level of the hierarchy directly. The data transfer (from one level to another) must be handled by application software.

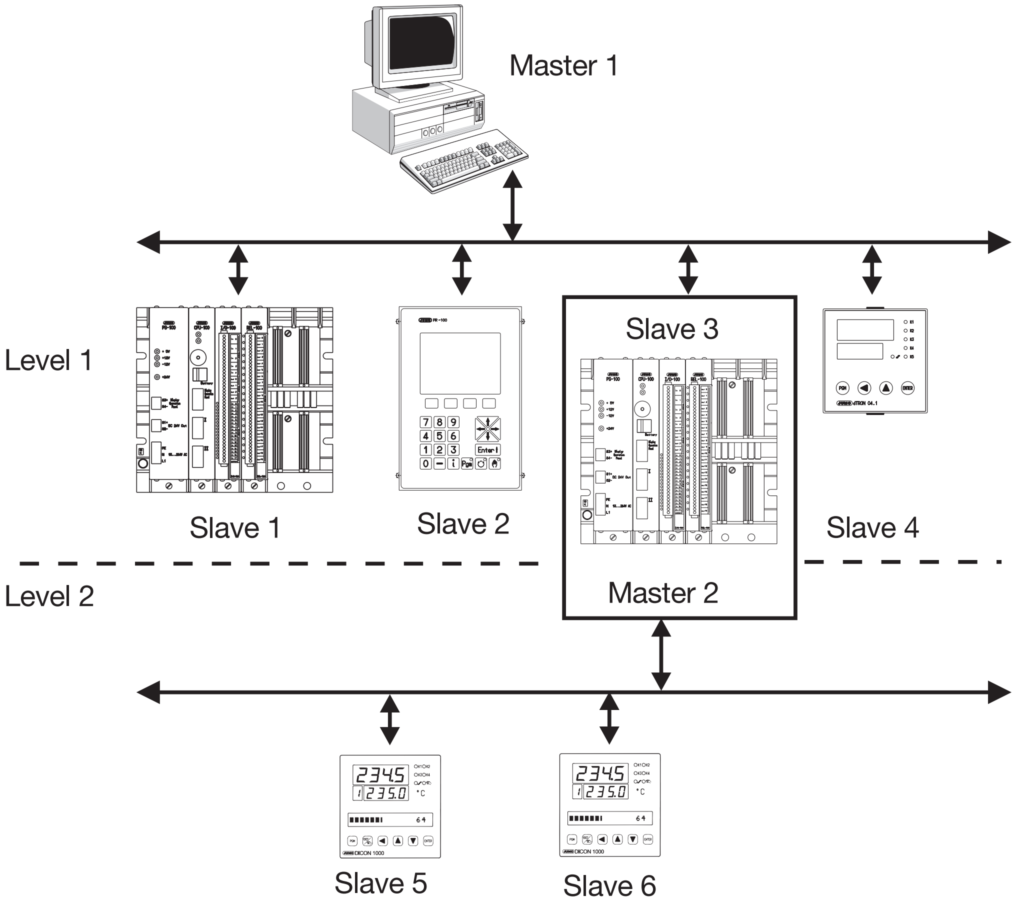


Fig. 64: Bitbus system with more than one hierarchical level

For developing Bitbus modules, Intel processor chips can be used (8044 or 80C152).

## 2.4 CAN bus

**CAN**

The CAN (Controller Area Network) bus was initially developed by Bosch and Intel in 1987, for applications in the automobile sector. For some years now, CAN has also been present in industrial, process, and building automation. Major CAN characteristics are very secure data transmission and extremely fast response times. The transmission medium is a twisted-pair cable (shielding is recommended) with a termination resistance. CAN is not usable for Ex areas. The CAN bus is internationally standardized in ISO 11898.

**CiA**
**CAN in Automation**

In March 1992, manufacturers and users set up the international association CiA (CAN in Automation). It supports its members worldwide from offices in Erlangen. CANopen was developed within CiA, and is supported by the association. CiA also publishes manufacturer-independent information of all kinds around the world.

| | |
|---|---|
| Manufacturer/ user organization | Bosch and Intel (ISO 11898) /<br>CAN in Automation (CiA) |
| Medium | twisted-pair cable |
| Subscribers | practical: up to 64 nodes<br>theoretical: unlimited |
| Data transmission rate | 50 kbit/sec to 1 Mbit/sec |
| Extent of bus | typical: 40 m at 1 Mbit/sec transmission rate,<br>up to 1 km with a reduced transmission rate |

Table 13: Summary of the CAN bus

**Multi-master system**

CAN is a multi-master system. This means that all the nodes can communicate with one another without a specific master. This is also achieved through the *broadcasting* procedure. In this procedure, no specific subscribers or nodes are addressed, as in the usual fieldbus systems, but prioritized messages are instead sent from one sender to all other nodes. Before a station transmits a telegram, with a maximum of 8 bytes of data (such as a temperature or pH value) it sends a message number (Identifier). The identifier shows what priority the message has, and which station is sending it. During the system design, the priorities are assigned through corresponding binary values, and cannot then be altered. A message with a lower identifier has a higher priority than messages with a higher identifier. If a CAN device transmits a message on the bus, then all other stations become receivers of this message.

Each station receives the transmitted message, and then decides, on the basis of the identifier, whether the data that have been received are relevant for it or not. Data which are relevant will be processed, but irrelevant data will be discarded.

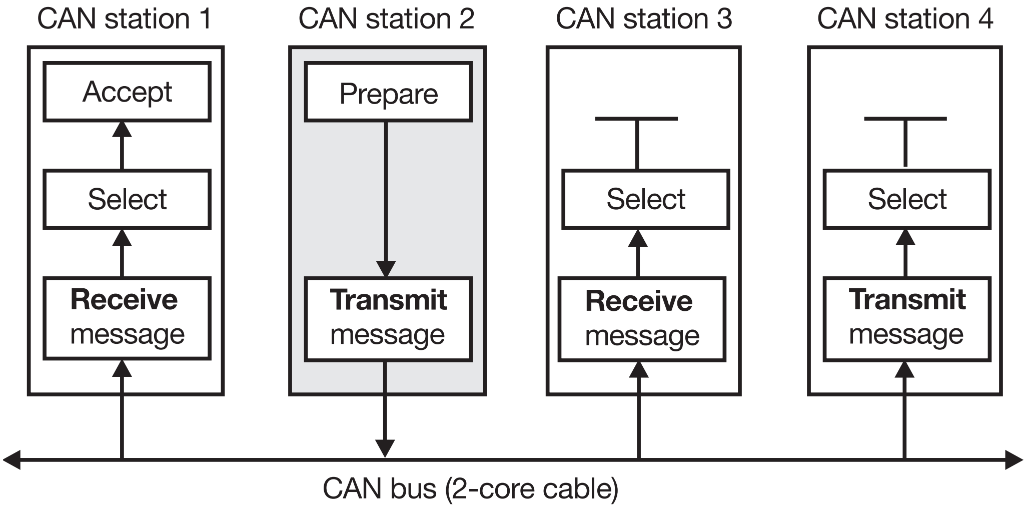


Fig. 65: Principle of data transmission on the CAN bus (station 2 sending to station 1)

**Bus arbitration**

Since all the CAN stations are permitted to transmit, and so have access to the bus if required, bus access must be on request. With the procedures used (CSMA, CSMA/CD, and CSMA/CA) there may be simultaneous attempts at access from more than one sender (access collisions). In CAN, the telegram with the higher priority is always granted access, and this access regulation is known as bus arbitration.

The message with the lowest binary number following the identifier has the highest priority in CAN. Possible bus access collisions are thus resolved without loss of data:

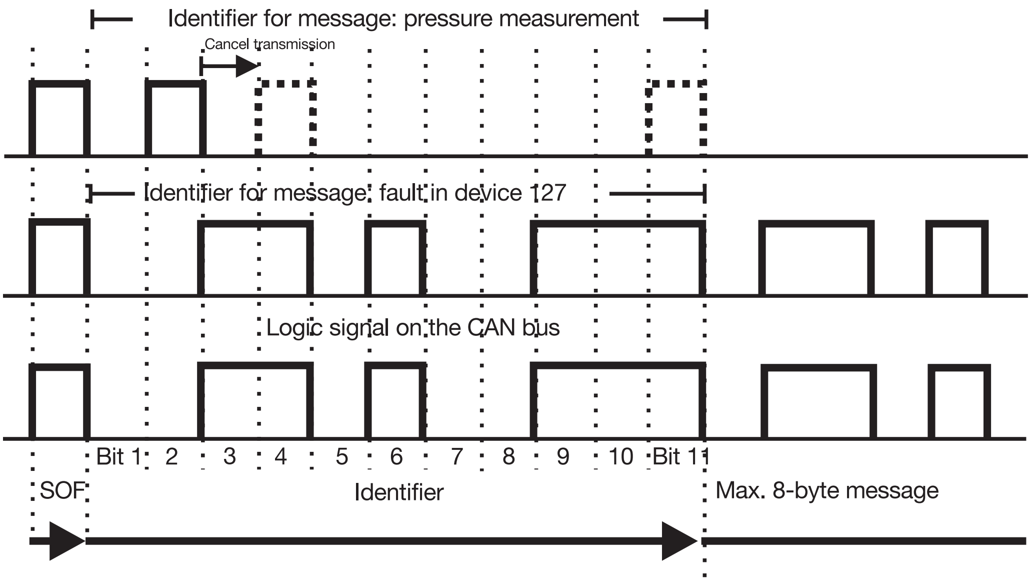

Identifier for message: pressure measurement
Cancel transmission
Identifier for message: fault in device 127
Logic signal on the CAN bus
Bit 1
2
3
4
5
6
7
8
9
10
Bit 11
SOF
Identifier
Max. 8-byte message

Fig. 66: An example of bitwise arbitration, without loss of data

If two stations, in this example a pressure transducer and another device with the bus address 127, start transmitting data, this begins with the start bit (SOF). This is followed by the identifier for the telegram (the pressure transducer wants to transmit the pressure, but the other device with the address 127 wants to transmit a fault signal with a higher priority). As the bus line is arranged as a kind of open-collector circuit, the logic LOW level overrides the HIGH level if both are present at the same time. Both devices look at the bus while they are transmitting their data. When the pressure transducer sees that the identifier on the bus is different from that which it transmitted (see identifier bit 2 in Fig. 66), it stops transmitting and also becomes a receiver for the present message. The pressure transmitter will repeat its transmission later.

**CANopen**

As defined in ISO 11898, CAN only covers the first and second layers of the OSI reference model. In the past, this made it very difficult or even impossible to establish a connection between modules from different manufacturers. In order to achieve an open network it is necessary to operate through a standardized method of communication. For this reason, several open CAN protocols have been developed, such as the CANopen protocol, which is very widely used in Europe. An open protocol makes is possible for devices from different manufacturers to operate together:

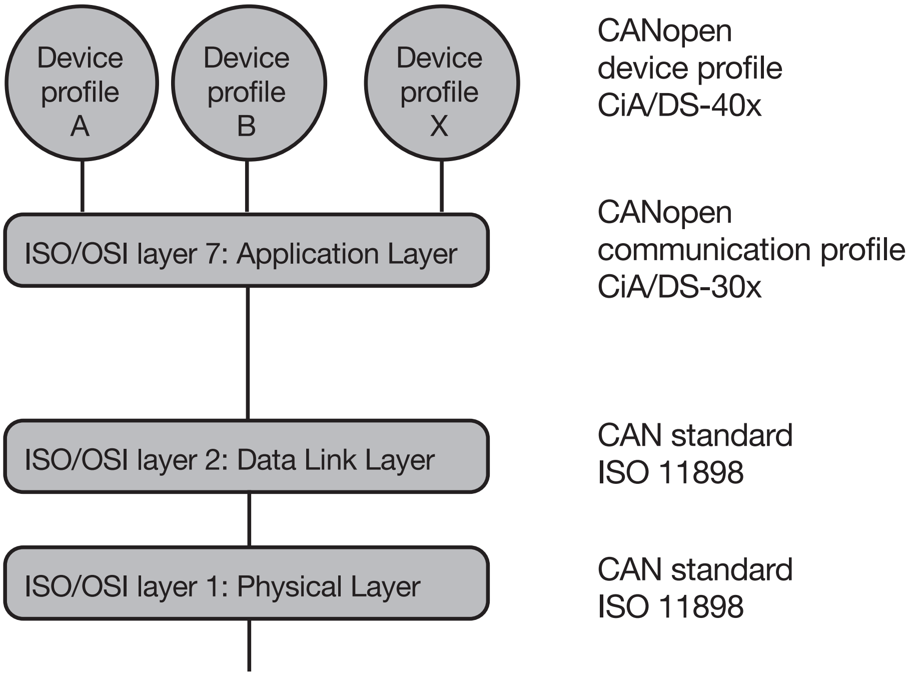

Device profile A
Device profile B
Device profile X
CANopen device profile CiA/DS-40x
ISO/OSI layer 7: Application Layer
CANopen communication profile CiA/DS-30x
ISO/OSI layer 2: Data Link Layer
CAN standard ISO 11898
ISO/OSI layer 1: Physical Layer
CAN standard ISO 11898

Fig. 67: Open communication (devices from different manufacturers are able to communicate with one another) through CANopen

CANopen is a family of profiles, consisting of the communication profiles (CiA DS-30x) and a number of device profiles (CiA DS-40X).

**Communication profile**

The CANopen communication profile represents the ISO/OSI layer 7 (Application Layer) and specifies how data are exchanged between network nodes.

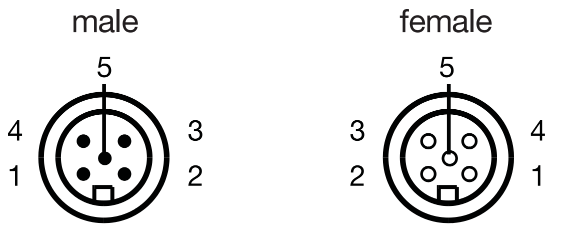
**Device profile**

The CANopen device profile defines standardized device functions. The device profile defines precisely how the data from a device are organized at the CAN interface. Device profiles make is possible to produce standardized devices for CANopen that can operate in networks together with devices from other manufacturers, without requiring any adaptation.

Device profile are available for such equipment types as:

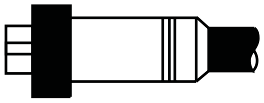
- I/O modules (CiA DS-401)
- drives and motion control (CiA DS-402)
- measuring services and closed-loop control (CiA DS-404)

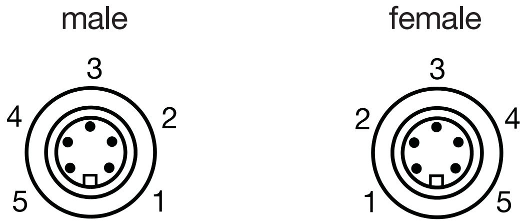
**Connectors for CAN**



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

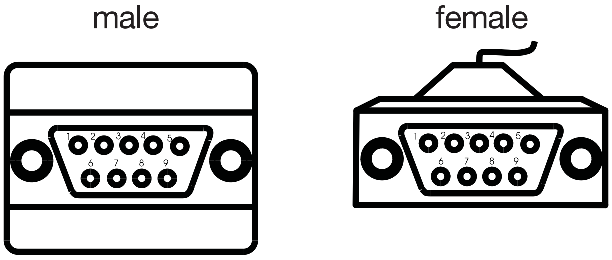
Fig. 68: Round connector M12x1; 5-pole as per IEC 60947-5-2





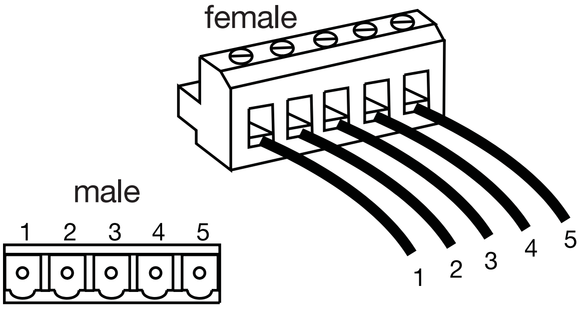
| PIN | SIGNAL |
|---|---|
| 1 | CAN_shield |
| 2 | V+ |
| 3 | V- |
| 4 | CAN_H |
| 5 | CAN_L |

Fig. 69: Round connector to ANSI/B93.55M-1981



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

Fog. 70: 9-pole sub-D connector to DIN 41652



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

Fig. 71: Screw terminal block

**EDS file**

The configuration and linking of a device into a CAN bus system requires a setup program. For CANopen devices, manufacturer-independent programs are available, which an be used to handle all CANopen devices together. Handling the device in the setup program requires the addition of the EDS (Electronic Data Sheet) that is supplied with the device.

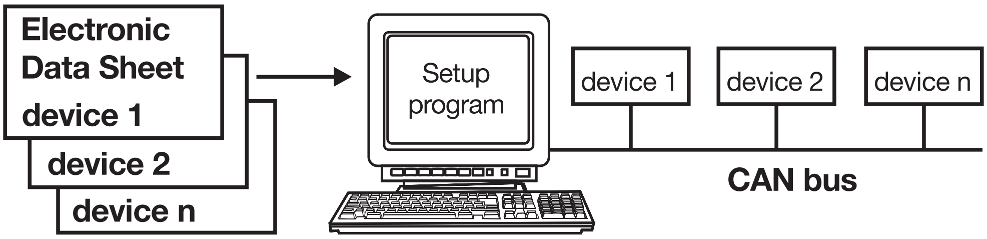


Fig. 72: CANopen EDS file

There are other open profiles in addition to CANopen. Mention must be made of the **DeviceNet**, developed by Rockwell Automation and Allen-Bradley for industrial automation, published in 1994, and supported and maintained today by the ODVA (Open Device-Net Vendor Association), and the **Smart Distributed System** SDS, developed in 1993 by Honeywell Microswitch in North America and optimized for the sensor/actuator communication sector.

## 2.5 FIP bus

**FIP bus**

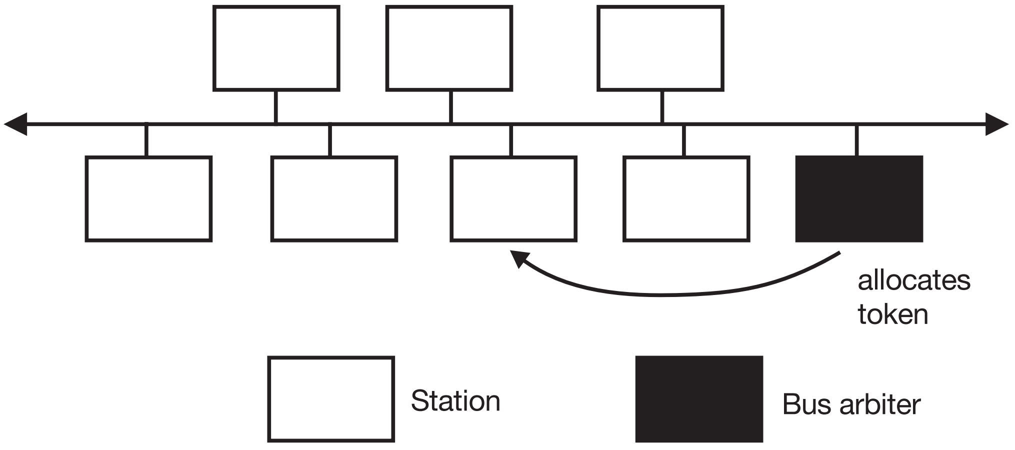
FIP (Flux d'Information vers le Processus) is a French standard, in effect the French equivalent of the German PROFIBUS, which has also been implemented by some Italian companies. It has been standardized as a French national standard (UTE 4660) since 1991, and included (as one of three fieldbus solutions) in the European fieldbus standard EN 50 170 since 1996. It uses a centrally controlled bus access procedure, whereby a master station passes the transmission authorization sequentially to the other stations, but always keeps control over the access to the medium.

| | |
|---|---|
| Manufacturer/<br>user organization | Telemechanique (Standard UTE 4660; EN 50 170)/<br>Cub World FIP Europe |
| Medium | FIP-specific (2-core, shielded) |
| Subscribers | max. 256 stations |
| Data transmission rate | up to 1Mbit/sec |
| Transmission distance | 1000 m |

Table 14: Summary of FIP

**Bus arbiter**

The FIP bus has a bus arbitrator to manage the bus according to the *delegated token* principle. This bus master can also have a redundant implementation. It delegates the token sequentially to the individual bus stations (Fig. 73).



- the bus arbiter manages the bus centrally (logical ring)
- station can only transmit after delegation through the bus arbiter

Fig. 73: Principle of decentralized control with FIP

**Token**

The subscriber which has received the token puts out its data onto the bus, and all other subscribers which are interested in the data listen in. As soon as the data have been output to the bus, the token is returned to the master, which can now delegate it once more.

## 2.6 Interbus

Interbus was developed by Phoenix Contact, and is defined in the standard DIN 19 258, so it is not a manufacturer-specific network. It is an open sensor/ actuator bus. Its main area of application is in production and process engineering, transport and storage technology. It is particularly strongly represented in the automobile industry and drive technology. It is supported nationally and internationally by the Interbus-Club, which follows the aims of further standardization and expansion. Furthermore, several user groups have been formed, such as DRIVECOM and ENCOM.

| | |
|---|---|
| Manufacturer/<br>user organization | Phoenix Contact (DIN 19 258) /<br>Interbus-Club, DRIVECOM, ENCOM, etc. |
| Medium | RS485 (twisted cable) or fiber-optic |
| Subscriber | max. 256; 31 per bus segment |
| Data transmission rate | 500 kbit/sec (designed for up to 2 Mbit/sec) |
| Transmission distance | max. distance between 2 stations is 400 m,<br>up to approx. 13 km with repeaters (remote bus) |

Table 15: Summary of Interbus

The bus system is constructed as a data ring, with a centralized master/slave access procedure. The bus master, in the form of a plug-in card, can also perform the function of coupling into a higher-level control system. Unlike a bus, the ring offers the possibility of simultaneous transmission and reception (full-duplex operation). The Interbus interface is implemented through an ASIC of the Interbus protocol chip (SµPI). This can be used to couple 16 input/output logic signals through internal registers.

**Interbus**

The Interbus data ring is built up like a closed-loop shift register. The data are transferred cyclically, with a fixed telegram length, using the summed frame procedure (Fig. 74).

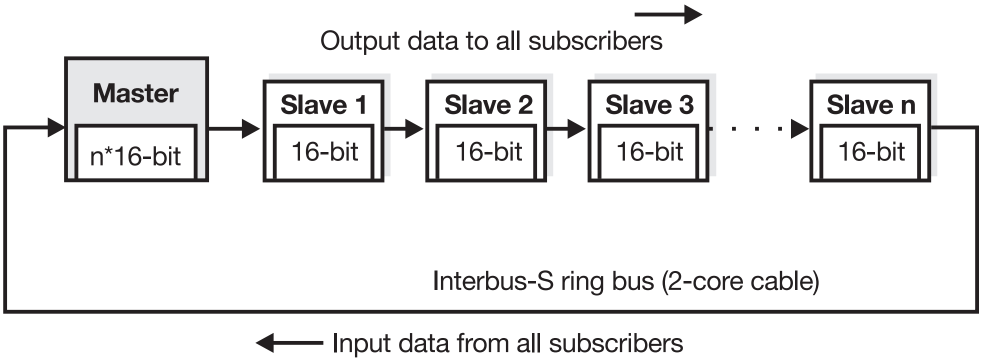

Output data to all subscribers
Master
n*16-bit
Slave 1
16-bit
Slave 2
16-bit
Slave 3
16-bit
Slave n
16-bit
Interbus-S ring bus (2-core cable)
Input data from all subscribers

Fig. 74: Principle of data transmission with Interbus, using the summed frame procedure

The data ring starts at the master, leads through the slaves, and finishes once more at the master. The slaves form the decentralized peripheral level. The master pushes data into the shift-register ring and receives data from it at the same time. This procedure is repeated cyclically, thus continually exchanging data between the master and the slaves. The data therefore pass along the bus in a constant and calculable cycle time. Interbus is a deterministic fieldbus, which always transfers the data at the same constant time intervals to and from the sensors and actuators.

As an example, the cycle time of a demonstration system in the fieldbus test center at Phoenix Contact, with 48 stations and a 300 kbit/sec transmission rate, is T = 3.5 msec. Since the system was devised for a maximum transmission rate of 2 Mbit/sec, this time could be further reduced by a factor of 5.

## 2.7 LON bus

LON (Local Operation Networks) is a development from the American company Echelon. The complete technology package from Echelon, including the development tools, carries the name LonWorks. The individual elements of LonWorks are the Neuron chip, the communication protocol LonTalk, the Neuron-C programming language, and various network coupler components. The LON technology uses a wide range of transmission media, so that the most suitable transmission medium can be utilized for every one of the measurement and control devices which are used.

This fieldbus concept is the only one in which all seven layers of the OSI model are available. Up to now it has mainly been used in building automation, but its extremely decentralized structure means that it will be particularly interesting in the future for automation engineering. The possibility of transmitting the data through a fiber-optic connection also makes it interesting for the chemical industry, for Ex areas.

In Germany, this concept is supported by the LNO user organization. This is divided into various working groups from the building automation and industrial automation sectors.

| | |
|---|---|
| Manufacturer/<br>user organization | Echelon /<br>LNO user organization |
| Medium | various: 2-wire, fiber-optic, power line, radio... |
| Subscribers | max. 255 subnets × 127 LON nodes |
| Data transmission rate | 9 kbit/sec to 1.25 Mbit/sec |
| Transmission distance | max. 2000 m |

Table 16: Summary of the LON bus

The individual LON modules can be connected together in any arrangement, i.e. in a free topology (linear, star, ring structures etc.). The user does not have to define addresses for the devices. Individual stations can be considered as quasi-masters, which access the bus according to demand (Fig. 75). This demand-dependent bus allocation between the individual bus stations is made according to the CSMA/CA method. Further details on communication via the LON bus are presented in Chapter 3.

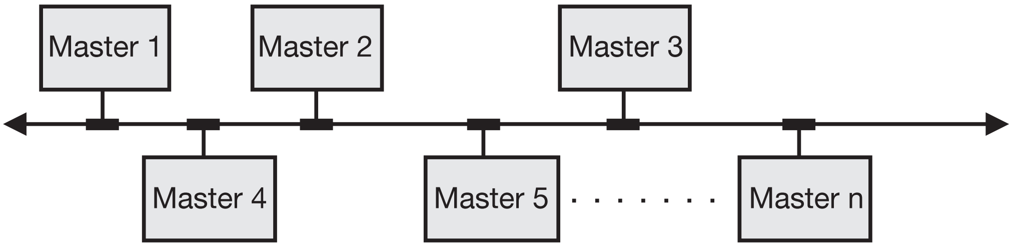


Fig. 75: Demand-dependent bus allocation between individual bus stations

## 2.8 Modbus

**Modbus protocol**

Modbus is a transmission protocol, which was developed in 1979 by the American company Gould-Modicon. It is a simple and safe protocol, and is specified in the "Modicon Modbus Protocol Reference Guide". Compared with other fieldbus systems, no physical interface is defined here. Users can decide themselves which interface is most suitable for them (RS422/485, RS232, TTY, fiber-optic, etc.).

| | |
|---|---|
| Manufacturer/<br>user organization | Gould-Modicon / supported by<br>international control system manufacturers |
| Medium | free: RS422/485, RS232, fiber-optic, TTY |
| Subscribers | max. 247 |
| Data transmission rate | up to 187.5 kbit/sec, depending on the interface |
| Transmission distance | up to 1200 m, depending on the interface |

Table 17: Summary of Modbus

Communication takes place according to the master/slave principle, in the form: request/instruction/response. The master (e.g. a PC), which identifies the slaves by their addresses, controls the entire data traffic.

The Modbus protocol can be looked upon as a quasi-industrial standard which is available for every kind of control system. Modbus is strongly represented in the American and French markets. In Germany it is represented by AEG in Seligenstadt.

**Modbus-Plus**

Since 1989 a further development has been available: Modbus-Plus, for applications with higher requirements. Unlike Modbus, it is registered as a trade name, and is a multi-master system with up to 4000 subscribers and a transmission rate of up to 1 Mbit/sec.

## 2.9 P-Net

**P-Net**

P-Net (a fieldbus) is a system which has been developed by the Danish company PROCES-DATA since 1984. It is provided to users without a licence being required. In 1991 the P-Net user organization A.p.S. was set up in Denmark. Since March 1996 it has been incorporated into the new European fieldbus standard EN 50 170.

P-Net is a multi-master system, with the special feature that several systems can be networked together (as bus segments). The system is conceived not so much as a fast sensor/actuator bus, but rather for the interconnection of intelligent sensor units or a PLC with industrial PCs. The applications are predominantly in quality assurance and process engineering. From data acquisition and control systems on trucks in the food and non-food sectors, to machines and plant in the textile, plastic and food production industries, in dairies, breweries and the like, P-Net has demonstrated its capabilities around the world in a large number of varied applications.

| | |
|---|---|
| Manufacturer/<br>user organization | Process Data (DSF 21906, Denmark, EN 50 170) /<br>P-Net user organization A.p.S. |
| Medium | RS485<br>(shielded twisted pair cable) |
| Subscribers | max. 125 |
| Data transmission rate | 76.8 kbit/sec |
| Transmission distance | 1200 m |

Table 18: Summary of the P-Net fieldbus

The well-known OSI reference model is also used to describe the P-Net standard. Layers 1, 2, 3, 4 and 7 are used. In addition to the typical subscribers for a master/slave mode (i.e. masters and slaves), the P-Net standard also defines a third type, the *controller* or *gatewa*y (see Fig. 76).

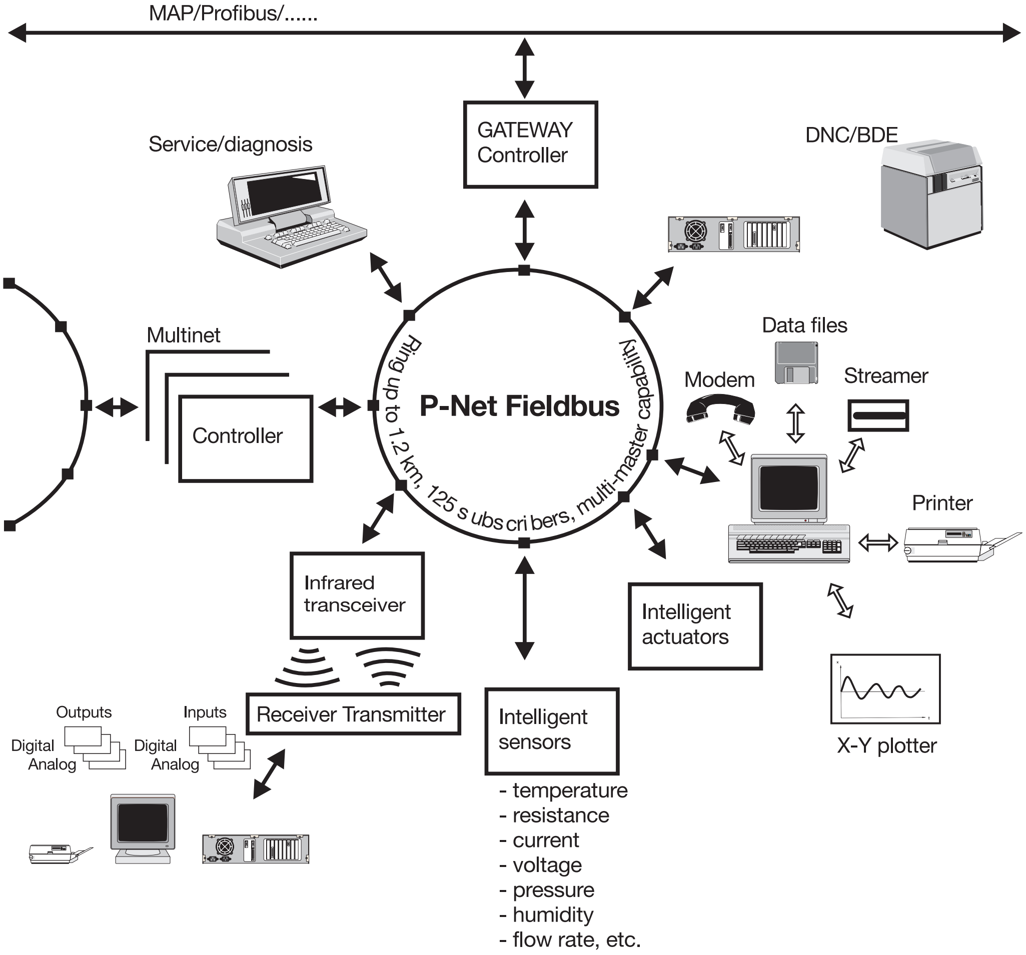


Fig. 76: Networking options with P-Net

The controllers have the task of linking two networks (bus segments) together. The decisive feature of these controllers is that they function in one direction as a slave, and in the other as a master. In their capacity as masters they are subject to the multi-master access procedure that is applied, which can also be described as “virtual token-passing”. Each master has a given numbering, and the bus access is delegated cyclically by ascending numbers. The system will continue to operate even if one or more masters fails.

## 2.10 PROFIBUS

PROFIBUS (PROcess FIeld BUS) is an international, open fieldbus standard, that has been standardized in the fieldbus standard EN 50 170.

The PROFIBUS technology was developed as a cooperation between several companies, led by Siemens, Klöckner Möller and Bosch, and is cared for by the PROFIBUS User Organization (PNO). The members of PNO include manufacturers, users and research establishments. Regional user groups in 16 major industrial countries provide support in the local language. All the regional user groups are members of PI, the overall organization.

**PROFIBUS consists of a family with 3 versions:**

**PROFIBUS-DP** PROFIBUS-DP (Decentralized Periphery) is designed for the area of decentralized peripherals, where rapid system response times are decisive. It is used to connect the central automation equipment (e.g. a PLC) to the decentralized input and output devices (sensors and actuators) via a fast serial interface.

**PROFIBUS-PA** PROFIBUS-PA (Process Automation) was specially designed for process engineering, and even permits the attachment of sensors and actuators in Ex areas. PROFIBUS allows data communication and power feed to the devices over a 2-core cable, in accordance with the international standard IEC 1158-2.

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

**PROFIBUS-FMS** PROFIBUS-FMS (Fieldbus Message Specification) is used for communication tasks at the group-control level. PROFIBUS-FMS includes layer 7 of the OSI reference model, as well as the layers 1 and 2 which are included in PROFIBUS-DP and PROFIBUS-PA. FMS contains the application protocol and provides communication services for the user. PROFIBUS-FMS was developed for universal transmission of large volumes of data.

| | |
|---|---|
| Manufacturer/ user organization | Bosch, Klöckner Möller, Siemens etc. (EN 50 170)/<br>PROFIBUS User Organization (PNO) |
| Medium | RS485 (twisted cable) or fiber-optic,<br>IEC 1158-2 specific for PROFIBUS-PA |
| Subscribers | max. 127 addressable; 32 per bus segment (with limitations for PA) |
| Data transmission rate | PROFIBUS-DP and -FMS: 9600 bit/sec to 12 Mbit/sec<br>PROFIBUS-PA: 31.25 kbit/sec, fixed |
| Transmission distance | up to 1200 m,<br>up to 1900 m (remote-fed bus segment) for PROFIBUS-PA |

Table 19: Summary of PROFIBUS profiles

**Transmission method**

Three different transmission methods are provided for PROFIBUS:

The **RS485 transmission**, often referred to as H2, is applied with DP and FMS. This method of transmission is the one most frequently used, and covers all areas where a high transmission rate and simple, cost-effective installation are required. The medium used is a shielded twisted-pair copper cable, with a transmission rate between 9.6 kbit/sec and 12 Mbit/sec. This technology permits up to 32 subscribers to be connected together in one bus segment. If there are more than 32 subscribers, then repeaters (cable drivers) must be used to join the individual bus segments. The bus is terminated by an active termination at the start and end of each segment (see Fig. 77). A 9-pole D-Sub connector is preferred for PROFIBUS networks which use the RS485 transmission method, and the pin assignment is shown below:

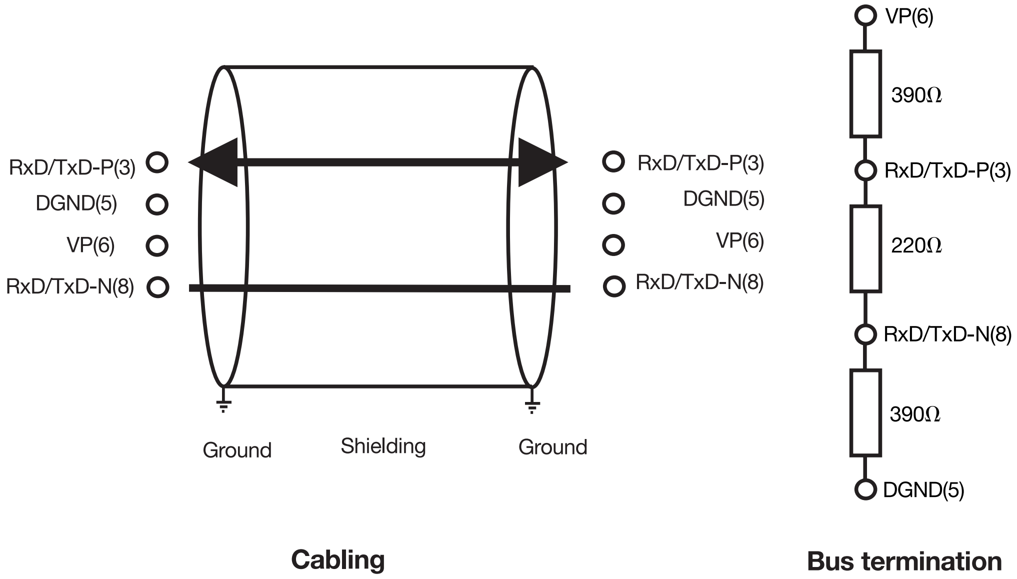


Fig. 77: Cabling and bus termination for PROFIBUS-DP and PROFIBUS-FMS

The **IEC 1158-2 transmission for PA** fulfills the requirements of the chemical and petrochemical industries, enabling PROFIBUS to be used in Ex areas (see Fig. 78). It provides both intrinsic safety, and the power feed to the field devices through the bus. The protocol which is used is often referred to as H1. Networks can be set up in linear, tree, or star topologies. The transmission medium is a 2-core cable (shielded or unshielded). The transmission rate is fixed at 31.25 kbit/sec, and cannot be altered.

With this technology there is only one feed point in each bus segment, the feeder unit, which is usually contained in the segment coupler which is used to link the bus sections to segments that are using RS485 transmission. The number of subscribers which can be connected to one segment is limited to 32, but it may be further limited by the level of Ex protection and the feed to the bus, in accordance with transmission as per IEC 1158-2.

When a subscriber transmits something, no power is fed into the bus, since the devices function as current sinks. Each bus subscriber draws a constant basic current (e.g. 10 mA). The communication signals are then generated by applying a modulation of +/-9 mA to the device which is transmitting.

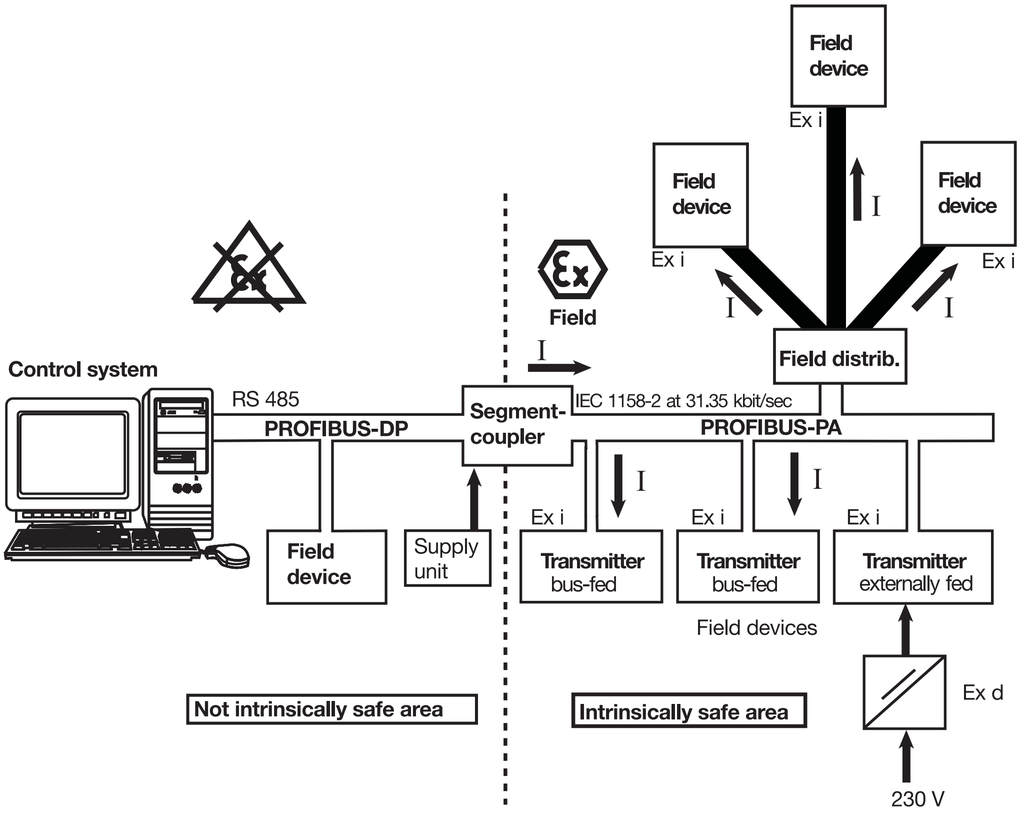


Fig. 78: Typical system configuration for process automation

**Fiber-optic**

Fiber-optic connections are ideal for applications in environments with a high level of interference and to increase the achievable transmission distance at high transmission rates. Special bus connectors are commercially available which have the conversion of RS485 signals to an optical fiber (or the other way around) integrated into the connector. This makes it possible to change easily from one transmission medium to another within an installation.

**Master and slave**

A central control that exchanges information with decentralized stations (slaves) in a fixed cycle is known as a **DP-master,** Class 1, cyclical data exchange. A typical example of such a device is a PLC. The programming, project-planning and operator devices which are used to configure the system are known as **DP-master(s),** Class 2, acyclical data exchange. A **DP-slave** is a peripheral device (I/O, drive, valve etc.) which acquires information and transmits it over the bus on request.

**Multi-master system**

All three PROFIBUS versions use the same bus-access protocol: this is the *token-passing* procedure, whereby the token (access right) is passed around a logical ring from one master to another. Each master possesses the token for a precisely predetermined length of time. It can only communicate with the other subscribers (slaves) during this period (see Fig. 79). If the bus includes more than one master, it is referred to as a multi-master system.

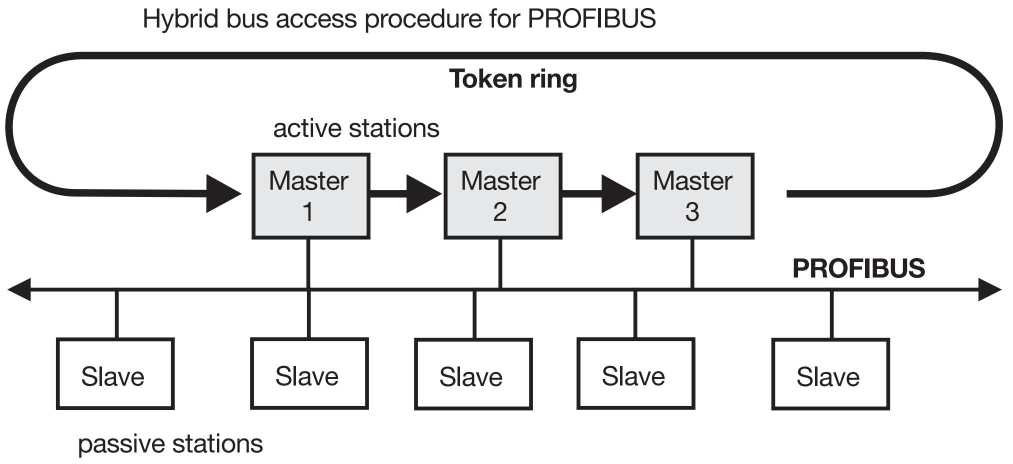


Fig. 79: Bus access in a logical ring

If only one master is operating at the moment, then it is a mono-master system (see Fig. 80).

Mono-master systems have the shortest cycle times, which is why they are more frequently used than multi-master systems.

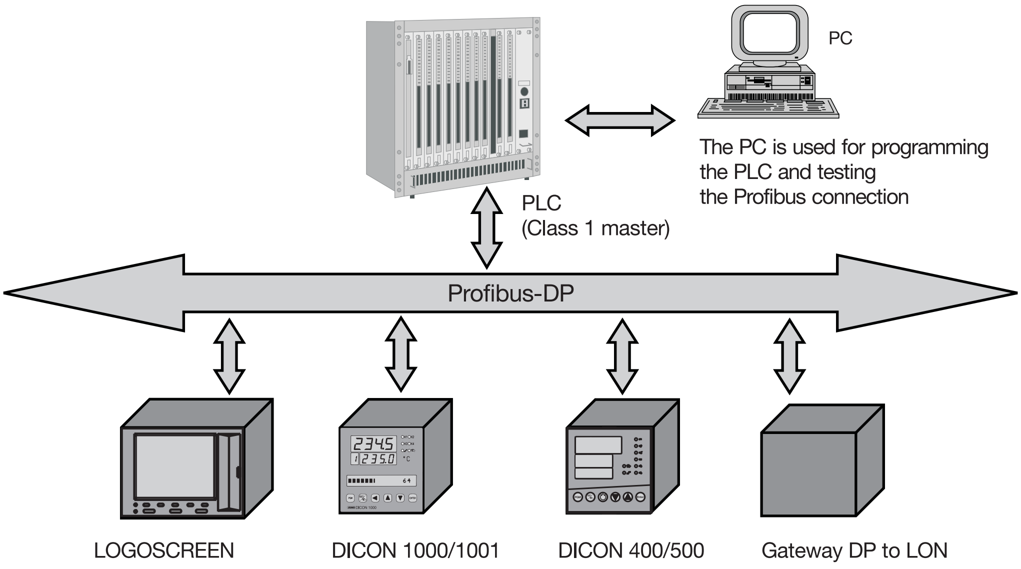


Fig. 80: PLC in mono-master operation with PROFIBUS-DP instruments from JUMO

**GSD file**

PROFIBUS devices have different performance features. They differ in the extent of the functions which they offer (e.g. number of I/O signals, diagnostic messages) or the possible bus parameters such as baud rate and time monitoring. These parameters are specific to each manufacturer and each type of device, and are usually documented in the instrument manual. In order to achieve a simple plug-and-play configuration for PROFIBUS, the instrument characteristics are recorded in the form of an electronic data sheet (GSD, from the German *Gerätestammdaten Datei* = device data base file). The precisely defined file format for the GSD means that the project-planning system can simply read in the fundamental data and take it into account during the configuration of the bus system (see Fig. 81).

Every manufacturer of a DP-slave device is obliged to provide a GSD file for the customer with the device.

User-friendly editors are commercially available for creating GSD files according to the standards.

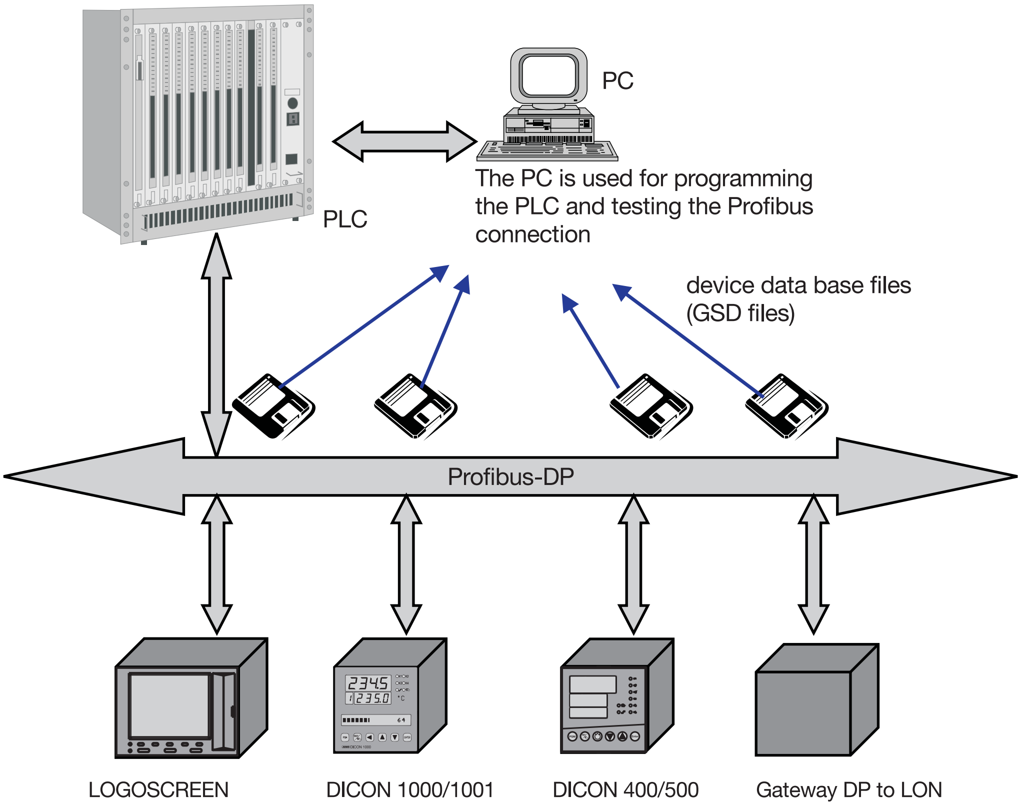

PC
PLC
The PC is used for programming the PLC and testing the Profibus connection
device data base files (GSD files)
Profibus-DP
LOGOSCREEN
DICON 1000/1001
DICON 400/500
Gateway DP to LON

Fig. 81: Configuration through GSD files

## 2.11 FOUNDATION fieldbus

The Fieldbus Foundation was created in September 1994 by the merger of WorldFIP North America and Interoperable Systems Project. It has existed since 1999 as a non-profit making organization of more than 120 suppliers and users from the automation industry. The members of this organization are the major players in the development of the FOUNDATION fieldbus.

**FOUNDATION fieldbus**

Fisher-Rosemount is the principal supplier of FOUNDATION fieldbus products. In the FOUNDATION fieldbus, layers 1, 2 and 7 of the OSI reference model are emphasized.

| | |
|---|---|
| Manufacturer/<br>user organization | Fisher-Rosemount<br>Fieldbus Foundation |
| Medium | twisted cable, with a bus termination resistor<br>2-core cable with H1<br>4-core cable with H2 |
| Subscribers | 32 |
| Data transmission rate | 31.25 kbit/sec with H1,<br>1 or 2.5 kbit/sec with H2 |
| Bus expansion<br>(main cable) | up to 1900 m with H1,<br>750 m with H2 |

Table 20: Summary of the FOUNDATION fieldbus

The signal transmission is not bitwise, but a Manchester Coding is used instead (see Chapter 2.12).

Two physically different types of bus are distinguished:

**H1 Fieldbus**

The H1 fieldbus is used for control tasks, such as temperature, filling level, or flow rate. Since the connection is made by a 2-core cable, the field devices can operate with the same cabling as conventional 4 — 20 mA instruments. If safety barriers are provided between the power supply in the safe area and the devices in the Ex area, then the devices may be operated in Ex areas.

The method of signal transmission is that a sender produces a current of +/– 10 mA. As a result, there is a 1 V (pk-pk) voltage drop across the 50 Ω termination resistor, as a modulation on the DC of the supply voltage. The transmission rate is 31.25 kbit/sec.

The bus allows devices to be arranged in a bus topology, but branching is also permitted. The fieldbus cable can have a maximum length of 1900 meters per segment when shielded twisted-pair cable is used. The total length is given by adding together the length of the main cable plus all branching cables. The termination resistors are installed at the ends of the main cable.

**H2 fieldbus**

The H2 fieldbus is used for more demanding automation tasks, detached I/O subsystems, and fast control applications. The devices are connected via a 4-core cable (with 2 separate supply feeds), but the devices often have an on-site supply.

The sender creates a +/– 60mA current through a 75Ω termination resistor, thus producing a voltage drop of 9V (pk-pk). The transmission rate is 1 or 2.5 Mbit/sec.

The H2 fieldbus can also transmit data in an Ex area, whereby the fieldbus signal is a modulation on 16 kHz AC.

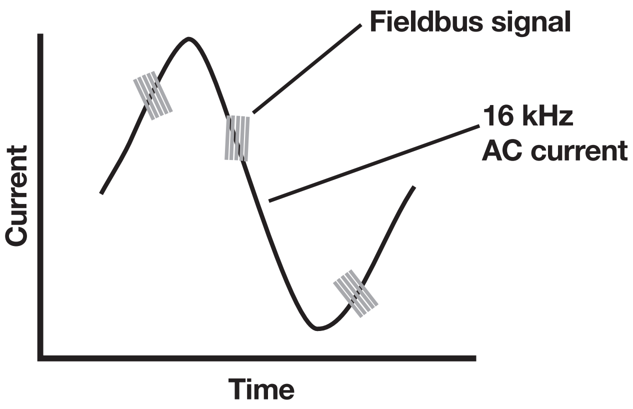


Fig. 82: H2 signal transmission via a current interface

Because of the high rate of transmission, H2 can only be implemented as a network in proper bus topology. Branching is not permitted.

**Cyclical (clocked) data transmission**

The FOUNDATION fieldbus includes a *Link Master* to control communication. The active master, also known as the *Link Active Scheduler* (LAS), has a list which defines the transmission times for all the data buffers of the devices which are on the bus. The contents of these data buffers must be transmitted at definite intervals. When the time comes for a device to transmit its buffer contents, the LAS sends a message to the device concerned. On receiving the message, the device transmits the contents of its buffer over the bus. Clocked data transmission is used to ensure regular, cyclical data traffic over the fieldbus.

**Free data transmission**

In addition to the cyclical polling of the data buffers of all devices, they also have the option of sending unclocked data over the bus, between the cyclical polling queries. To do this, the LAS transmits a *Pass Token* (PT) message to a device, and so gives it the opportunity of accessing the fieldbus. The device is allowed to send messages until it has finished, but only up to the limit of the token-holding time.

**Device description**

The manufacturers provide device descriptions with the fieldbus devices, to enable communication between the field devices and a control system or a higher-level computer.

## 2.12 Ethernet

Ethernet is primarily used in the area of office communication. However, efforts are being made to utilize this network technology for industrial applications as well, right down into the field area.

**Ethernet**

Ethernet is based on the IEEE802.3 standard. This is a part of the IEEE802 set of standards which describes the CSMA/CD access procedure, Token-Bus and Token-Ring in LANs. ISO took over these standards as the foundation for international standards (ISO 8802). The IEEE (Institute of Electrical and Electronics Engineers) is the world's largest professional body. It is concerned with the development of standards for local networks and the publication of test reports in professional journals, and the organization of many annual conferences. IEEE802 is the most important standard for local area networks (LANs).

| | |
|---|---|
| User organization | The IEEE<br>(Institute of Electrical and Electronics Engineers) |
| Medium | 2 coax cable types (10Base-5 and 10Base-2)<br>twisted pair cable (10Base-T)<br>fiber-optic (10Base-F) |
| Subscribers/segment | 100 for 10Base-5<br>30 for 10Base-2<br>1024 for 10Base-T and 10Base-F |
| Data transmission rate | 10 Mbit |
| Bus length | 500 m for 10Base-5<br>200 m for 10Base-2<br>100 m for 10Base-T<br>2000 m for 10Base-F |

Table 21: Summary of Ethernet

Networks to IEEE 802.3 work with CSMA/CD as the access procedure, and are the most widespread systems at present, with an enormous installed base and a correspondingly large fund of experience.

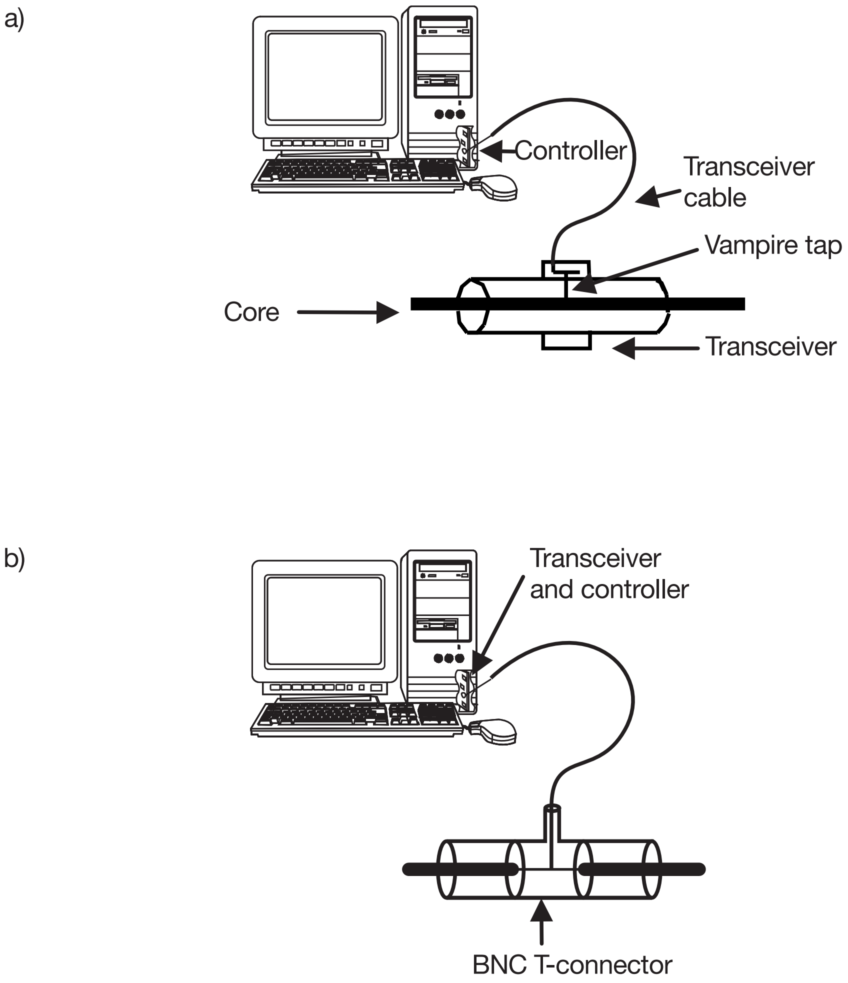
Four types of cable are used:
The 10Base-5, generally known as *Thick Ethernet*, is the first cable, from the historical point of view (see Fig. 83 a). It is as thick as a garden hose, and has markings at intervals of 2.5 m, to indicate access points. Connections are generally made through "vampire taps", whereby a needle is cautiously inserted into the core of the coax cable to about the midway point. The cable operates at 10 Mbit/sec, and a maximum of 100 subscribers can communicate across the cable. The maximum length of a segment is 500 meters.

The second type of cable was the 10Base-2 or *Thin Ethernet*, which, unlike Thick Ethernet, can be easily bent (see Fig. 83 b). With this type of cable, connections are made via standard industrial BNC connectors in the form of T-pieces. Connections are simpler and more reliable than with Thick Ethernet.

Thin Ethernet is also much cheaper and simpler to install, but can only cover up to 200 m, and only manages 30 devices per segment.

With 10Base-T there is no bus cable, but a *hub* (see Fig. 83c). In this configuration it is very easy to add or remove a station. Cable breaks can also be detected very quickly. The disadvantages of 10Base-T are that the maximum cable length from hub to the station is only 100 to 150 m, and the hub is relatively expensive. Nevertheless, 10Base-T is enjoying increasing popularity, mainly because of its ease of maintenance.

The fourth cable option is 10Base-F, using fiber-optic cable as the transmission medium. This expensive alternative has excellent noise immunity, and is often used to connect widely separated hubs.



c)

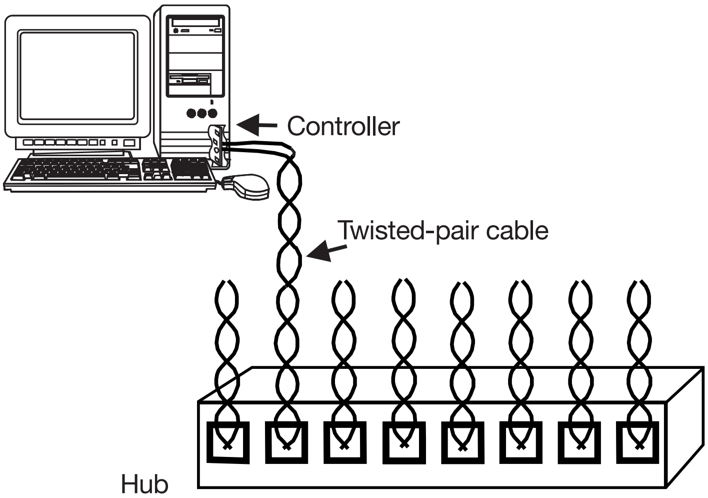


Fig. 83: Three cable types to 802.3; (a) 10Base-5; (b) 10Base-2; (c) 10Base-T

Ethernet permits almost any kind of network topology.

**Manchester coding**

802.3 does not use plain binary encoding, such as 0V for logical “0” and 5V for logical “1”, because this leads to ambiguity. If one station transmits the bit sequence 0001000, this might be erroneously interpreted by other stations as 10000000 or 01000000, because they cannot distinguish between a quiescent sender (0V) and a “0” bit (0V).

So a method is required to allow the receiver to recognize the start or middle of every bit, without having to use a separate clock signal. One such method is Manchester coding. In this case, every bit period is divided into two intervals. For a logical “1” the voltage is set high in the first interval and low in the second interval. For a logical “0” the voltage is set low in the first interval and high in the second interval.

The fact that every bit period has a transition in the middle makes it easier to synchronize a receiver to the sender. However, Manchester coding has the disadvantage that, for a given data flow rate, twice the bandwidth is required, since two pulses are transmitted per bit.

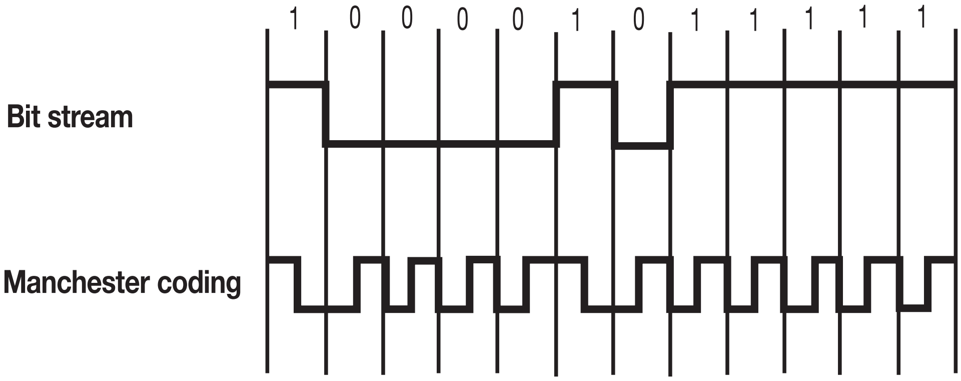


Fig. 84: Binary and Manchester coding

**Frame length**

With 802.3, a maximum of 1500 data bytes can be transmitted in a protocol. But there is also an important reason to define a minimum frame length: if a protocol frame was very short, a station might finish its transmission before the first bit had even reached the receiving station. The transmitting station would then assume that the reception had been OK, but a collision could be produced:

**Example:** Station A sends a message from one end of the network to Station B, and this message requires a time $\tau$ to reach the other end of the network. Another Station, C, is at this end, and starts transmitting before it has detected the message from Station A. Station C listens to the cable while it is transmitting, and detects a higher power level on the network than it is sending out (and thus the collision). So Station C generates a signal to warn all the other stations. But it also takes the same time $\tau$ until this warning reaches Station A, which then stops transmitting. But if a station sends out a message, it assumes that the reception was correct if it does not receive a collision warning from another station during its transmission. So the transmitting time for all data frames must be at least $2\tau$ long.

**Random control for collision handling**

After a collision, caused by two stations transmitting at the same time, the time is divided into individual slots. The duration of a time slot is 51.2 µsec. After a collision the stations wait for 0 or 1 slot times before starting to transmit again. If the station choose the same slot, there will again be a collision. After this (second) collision they each pick a number 0, 1, 2 or 3 at random, and start transmitting again after this random number (0, 1, 2 or 3) of slots. The more collisions, the higher the maximum random number is set, until after 10 successive collisions the maximum number is limited to 1023, and after 16 further collisions an error is signalled. This algorithm is called binary exponential back-off.

Ethernet is therefore not deterministic (so real-time capability is possible, but limited, depending on the circumstances). Furthermore, there is no priority arrangement.

## 2.13 Summary of the fieldbus systems

| Bus system | Topology | Type | Medium | No. of subs. | Data rate kbit/sec | Distance max. in m |
|---|---|---|---|---|---|---|
| ASI | any | master/ slave | 2-wire unshielded | 31 | 167 | 100 |
| Bitbus | bus | master/ slave | RS485 | 250 | 62.5 — 2400 | 13200 — 30 |
| CAN | bus | CSMA/ CD | RS485 twisted | eff. 64 | 50 — 1000 | 1000 — 40 |
| FIP | bus | multi-master | 2-wire shielded | 256 | 1000 | 1000 |
| Interbus | ring | master/ slave | RS485 twisted, fiber-optic | 256 | 500 | 400 |
| LON | any | CSMA/ CA | any | any | 9 — 1250 | 2000 |
| Modbus | bus | master/ slave | any, shielded | 247 | 187.5 | 1200 |
| PROFIBUS | bus | master/ slave | RS485 twisted, fiber-optic to IEC 1158-2 | 127 | 9.6 — 12000 | 1200 |
| HART | bus | master/ slave | 2-wire unshielded | 15 | 1.2 | 1500 |
| FOUNDATION fieldbus | bus, limited branching | master/ slave | 2- or 4-wire, shielded | 32 | 31.25<br>1000<br>2500 | 1900 |
| Ethernet | bus or tree structure | CSMA/ CD | coax, 2-wire, fiber-optic | 400 per segment | 10000, 100000 with fast Ethernet | 2000 |

Table 22: Summary of important fieldbus systems

# 3 Organization of the data system for JUMO

Appropriate hardware conditions (physical interface) and software conditions (transmission protocol) must be fulfilled in order to ensure the proper functioning of data communication with a computer or a higher-level (supervisory) control system. In other words, automation devices that have to communicate with one another must speak the same language. The µP-controlled process equipment from JUMO supports various standards. The different interfaces and protocols are described briefly in this chapter, with examples. Detailed information can be found in the operating instructions and interface descriptions for the individual items of process control equipment.

# 3 Organization of the data system for JUMO

## 3.1 The various communications options

### 3.1.1 Physical interfaces

**COM port**

Each computer/PC is equipped as standard with one or more serial communications interfaces, known as COM ports. These are usually implemented as RS232 serial interfaces (also known as V.24). Such interfaces merely permit the attachment of a process device that meets the same hardware specification. This was the physical interface that was offered in the first generation of µP-controlled equipment.

But data systems for modern automation technology requires the networking of various devices, sometimes over considerable distances, and, for instance, via a bus. This can be achieved with bus-capable serial interfaces, such as RS422 and RS485, which can also cover greater distances than RS232. If no separate interface card is used in the computer, then instruments with this type of interface can only be operated from the computer COM port through an interface converter (see Chapter 3.5.5 “Connection via an interface converter”). Since these types of interface are supported by many fieldbus systems, JUMO offers a combined RS422/485 interface card for process control instruments that have communication capabilities.

As well as this type of interface, most instruments also have an additional serial TTL interface, for use as a setup interface for configuration from a PC (see Chapter 1.9.1 “Configuration software”).

As a rule, twisted screened cables should be used for interface connections. However, this depends on the maximum cable length and the electromagnetic interference to which it is subjected.

### 3.1.2 Transmission protocols and fieldbus systems

Before unified fieldbus standards with defined transmission protocols were even considered, every manufacturer of process control equipment had already basically specified his own input/output syntax for those instruments which had communication capabilities. So it was with JUMO too, where the data exchange for the first generation of µP equipment was implemented through a JUMO-specific protocol, consisting of a string of ASCII characters. These days, this type of data transfer is no longer offered.

At present, the orientation is towards certain industrial fieldbus standards. For its communication-capable equipment (such as compact controllers, recorders etc.) JUMO is thus offering options for the Modbus protocol, PROFIBUS, CAN and HART. These transmission protocols and fieldbuses are supported by a very large number of manufacturers of automation equipment and visualization/process control software packages. These instruments and products then include drivers that can be used to set up the data communication with the JUMO instrument. For example, communication can be established in this way between a PLC with a Modbus or PROFIBUS driver (from AEG, April, Siemens, Telemechanique, etc.) and a universal and autonomously operating compact controller which is installed at the process site (see Fig. 85).

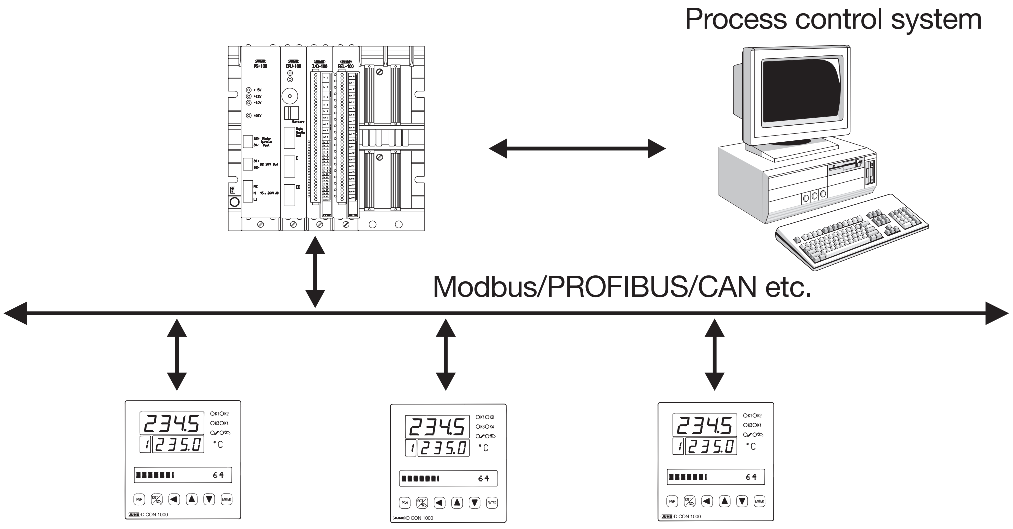


Fig. 85: Connecting JUMO instruments to a PLC by Modbus, etc.

In principle, any type of fieldbus and/or transmission protocol can be used to implement a distributed automation design, as long as the method that is chosen meets the economic and physical specifications. However, if equal handling of intelligent units is required, then pure master/slave procedures (such as the previously mentioned Modbus) must be ruled out. For this reason, JUMO has selected a modern fieldbus technique for implementing automation concepts with distributed intelligence for the JUMO mTRON automation system. This is the fieldbus design known as LON (see Chapter 3.4 “JUMO instruments with LON”).

## 3.2 JUMO instruments with HART

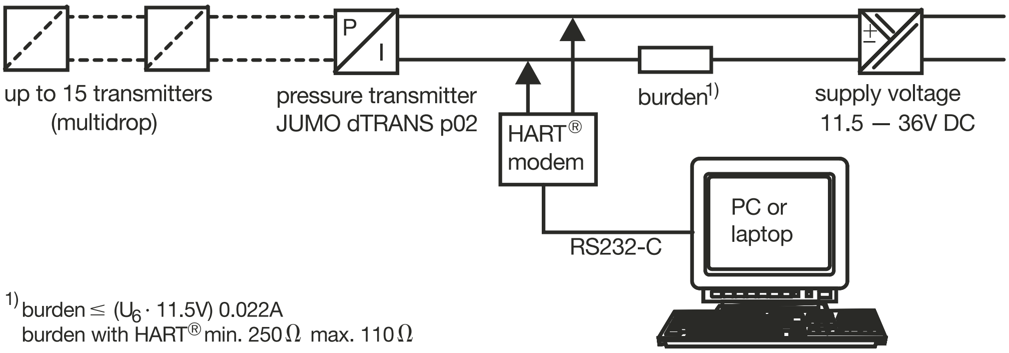
The procedure for configuring a JUMO instrument that uses HART communication is illustrated by the example of the JUMO dTRANS p02 pressure transmitter.

The pressure transmitter type JUMO dTRANS p02 is fitted with a 4 — 20mA analog output. The digital communication signal is superimposed on this, as is usual for sensors that use the HART protocol.

In addition to the basic parameter of pressure, the dTRANS p02 can also transmit the data for temperature, max. pressure and min. pressure via the digital signal.

**Communication with a PC**

Communication between a PC and dTRANS p02 requires a normal commercial HART modem, which converts the frequency signal from HART to the RS232 interface. A burden resistor is also inserted in the circuit, so that the varying current signal develops a varying voltage signal across the resistor (see Fig. 86).



[1)] burden ≤ ($U_6$ · 11.5V) 0.022A
burden with HART® min. 250 Ω max. 110 Ω

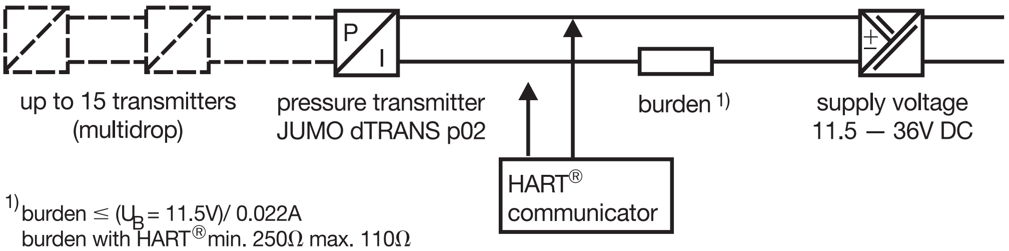
Fig. 86: Communication between a PC and a transmitter

**Setup program**

The setup program for the JUMO dTRANS p02 can be used to set the parameters of the pressure transmitter, after communication has been achieved with the PC. The setup program has a user interface for field units, as per VDI/ VDE2187. With this, the parameter data for the instrument can be defined, such as zero point, gain, unit address etc. and then transferred to the instrument via the HART communication. In addition, the measured values from HART instruments can be visualized and recorded.

**HHT**

Furthermore, a *hand-held terminal* (HHT) can be used for setting the parameters of the pressure transmitter:



[1)] burden ≤ ($U_B$ = 11.5V)/ 0.022A
burden with HART® min. 250Ω max. 110Ω

Fig. 87: Communication between a HART communicator and a transmitter

The HHT already has a modem built-in, and is also wired up to the transmitter with a burden resistor (see Fig. 87).

A HHT can be used to operate all HART devices, regardless of the manufactuer. The HHT can also be used in areas with an Ex (explosion) hazard.

**Keypad/LCD**

If HART communication is not used, the parameters for the dTRANS p02 can also be set up in the usual way by using the keys on the instrument.

## 3.3 JUMO instruments with CANopen

JUMO produces several instruments with a CANopen interface (transmitters, controllers, recorders).

The following illustrates the use and operation of the JUMO dTRANS p01 CAN pressure transmitter.

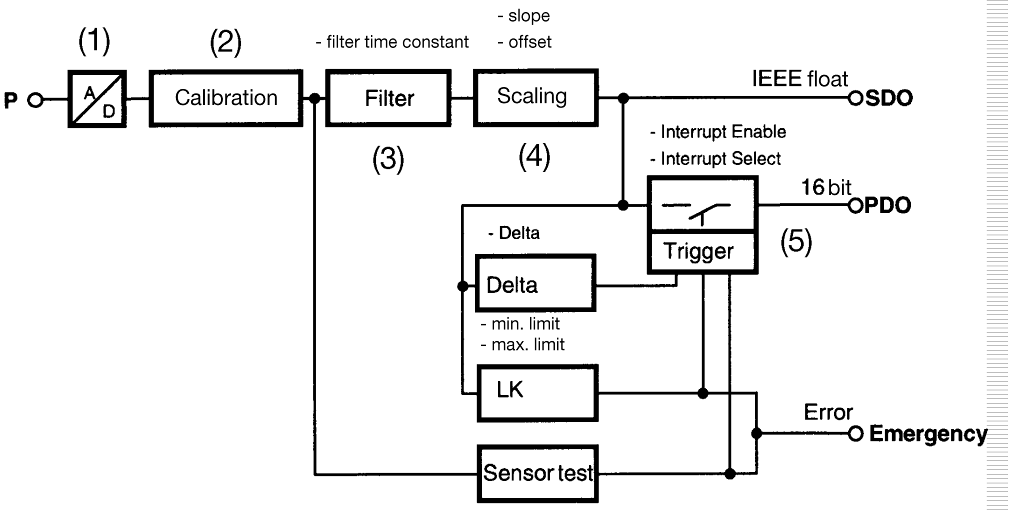


Fig. 88: Block diagram of the JUMO dTRANS p01 CAN pressure transmitter

The analog signal from the pressure sensor/transducer is digitalized (1) and digitally adjusted (ex-factory). This is followed by filtering (3) of the signal, and a subsequent scaling of the measurement value (4).

With **CANopen** there are **various types of telegram**:
In an **emergency telegram**, error messages are transmitted with a very high priority. For instance, the pressure sensor transmits an **emergency telegram**, if the sensor test or limit comparator produces an error.

The **PDO** (Process Data Object) **telegram** is used to transmit process data, typically measurements. In the PDO telegram, the pressure sensor transmits the pressure measurement, if one of the following trigger (5) conditions is fulfilled:

- Delta: if a preset pressure change has occurred
- LK: if the limit comparator has been activated
- Sensor test: if the pressure sensor has a fault

In the **SDO** (Service Data Object) telegram, acyclic data (such as parameters) are transmitted on request. The pressure sensor transmits the pressure measurement as a floating value, as per IEEE.

**Bus address**
**Baud rate**

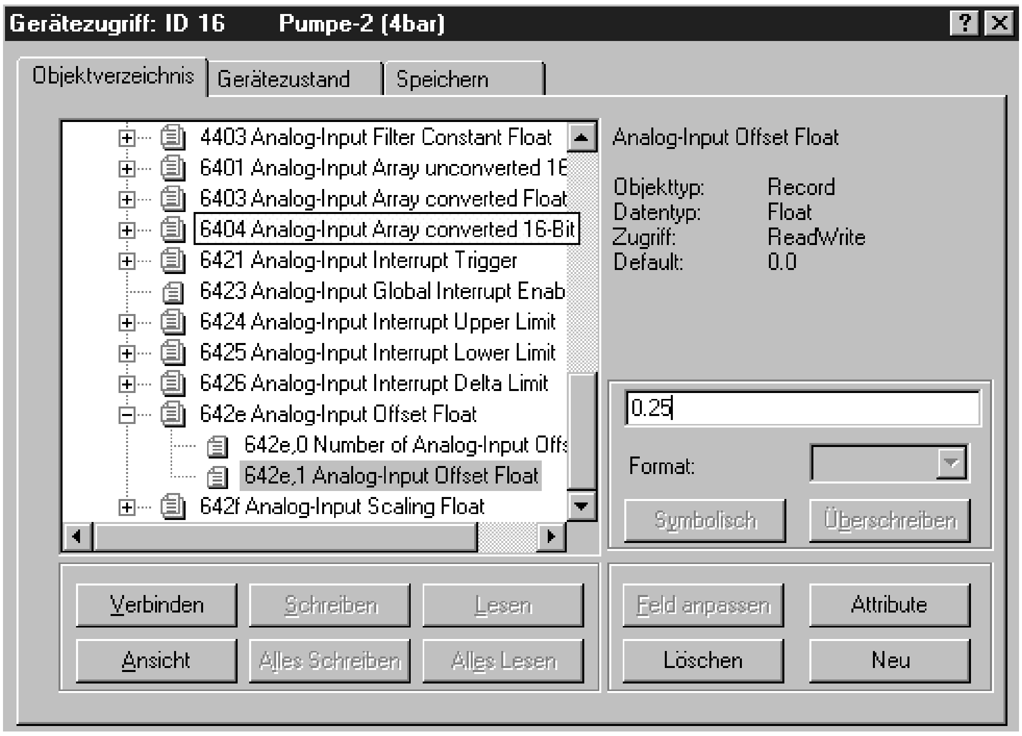
To configure a dTRANS p01 CAN pressure transmitter for connection to a CAN bus system, the following procedure must be applied:

First of all, the bus address (module ID) and the baud rate must be set in the pressure transmitter. The module ID can be set by DIP switches, in the range from 1 to 127. Alternatively, if all the DIP switches are OFF, then the module ID can be defined through the setup program. The baud rate can also be set within the range from 10kbit/sec (10 kbaud) to 1Mbit/sec (1 Mbaud) by using the DIP switches or the setup program.

**EDS file**

An EDS (= Electronic Data Sheet) file is attached to the pressure transmitter, so that all the other parameters can be adjusted through normal commercial set-up programs.

**Setup program**



Gerätezugriff: ID 16 Pumpe-2 (4bar)

4403 Analog-Input Filter Constant Float
6401 Analog-Input Array unconverted 16
6403 Analog-Input Array converted Float
6404 Analog-Input Array converted 16-Bit
6421 Analog-Input Interrupt Trigger
6423 Analog-Input Global Interrupt Enab
6424 Analog-Input Interrupt Upper Limit
6425 Analog-Input Interrupt Lower Limit
6426 Analog-Input Interrupt Delta Limit
642e Analog-Input Offset Float
642e,0 Number of Analog-Input Offs
642e,1 Analog-Input Offset Float
642f Analog-Input Scaling Float

Analog-Input Offset Float

Objekttyp: Record
Datentyp: Float
Zugriff: ReadWrite
Default: 0.0

0.25

Format:

Fig. 89: Setup program for a CANopen instrument

All the instrument parameters for the object directory are presented, with index and name. After a parameter has been selected (e.g. pressure, zero, module ID, baud rate...) it can be read out and altered, if required.

## 3.4 JUMO instruments with LON

At the field level, measurement and automation technology has been in a state of flux for some years. The structural concepts for signal transmission between sensors, actuators, and the centralized automation intelligence are changing from the single-signal oriented transmission to networked, decentralized peripheral devices for information processing close to the process. The introduction of Local Operating Network technology at the sensor/actuator level has led to the establishment of distributed information processing systems that are made up of independently operating units. In addition to the concept for the JUMO mTRON automation system, this section describes the layout of LON networks and their data communication.

### 3.4.1 The JUMO mTRON concept

When selecting the fieldbus, JUMO decided on a new technology for the implementation of automation designs with distributed intelligence. This is the fieldbus network concept known as LON, a development from the American company Echelon (see Chapter 2.7 “LON bus”).

The mTRON system consists of autonomous units (controller, logic module, relay module, etc.), each of which has an integrated LON interface.

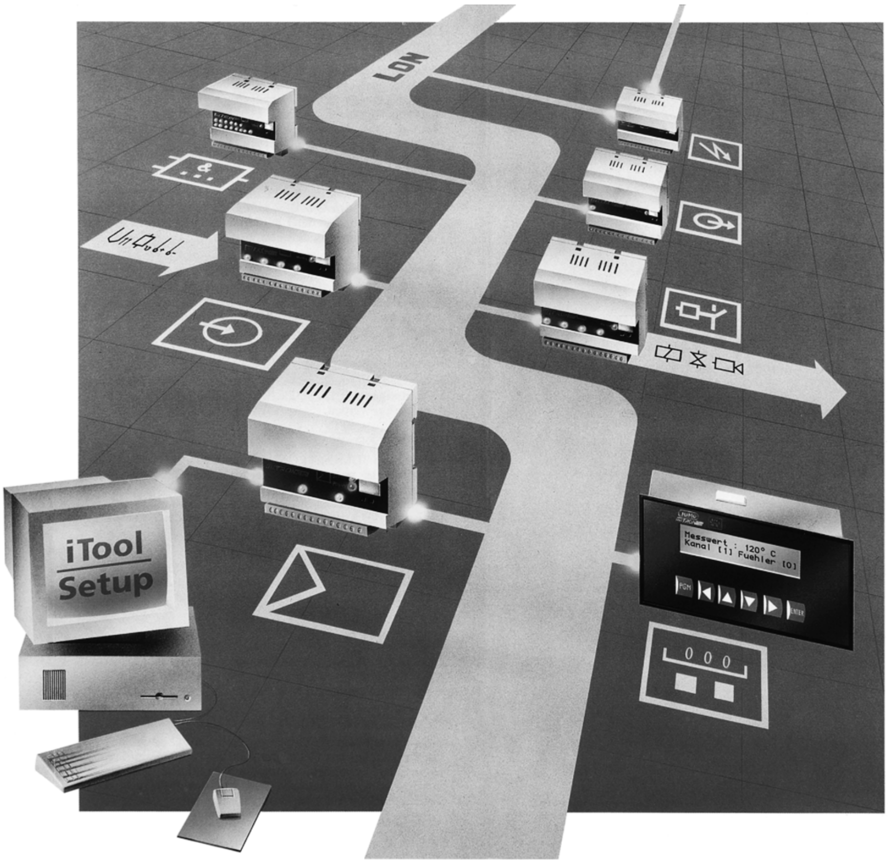


Fig. 90: The JUMO mTRON concept

The housings and technical dimensions of the individual modules are identical. The connections between the sensors/actuators and the instruments, as well as between one module and another, are made by plug-in screw terminals. An operating terminal that can be built into a cabinet door also enables direct on-site process control, as well as from the PC.

The JUMO mTRON system uses a screened twisted pair for signal transmission (Fig. 91).

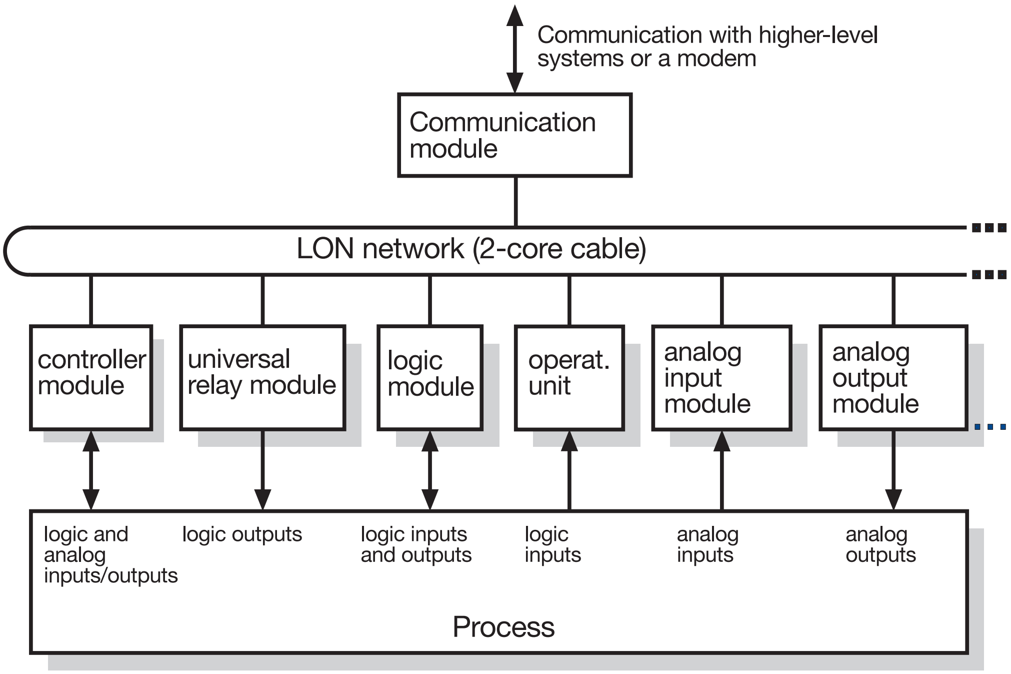
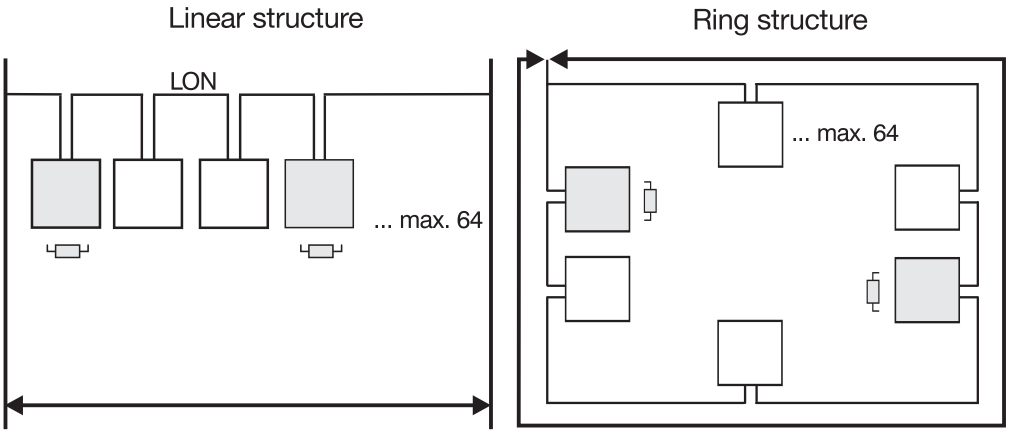


Fig. 91: JUMO mTRON automation system instruments in process control

The topology can be arranged as a linear structure, ring, star, or a mixed structure (free topology). For instance, at a transmission rate of 78 kbit/sec the linear structure can have a maximum extent of approx. 2700 meters (Fig. 92). With ring, star or mixed structures, the maximum extent is around 500 meters.



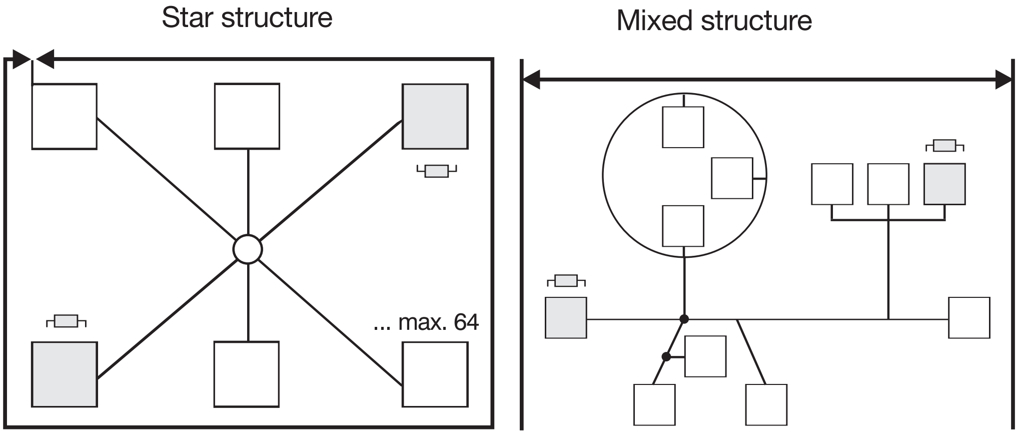


Fig. 92: Various network topologies

The type FTT-10 transceiver that is used by JUMO (Table 23) can be used to network up to 64 modules. Expansion of the network through additional amplifier elements (routers, bridges), as described below, is not part of the JUMO mTRON system.

### 3.4.2 Network structure

The basic idea behind the implementation of a LON network is to handle instruments which have a LON interface as if they were components in an electronic circuit. The linking of the network variables of various LON devices makes it possible to integrate the individual functions into whatever complex automation structures are required.

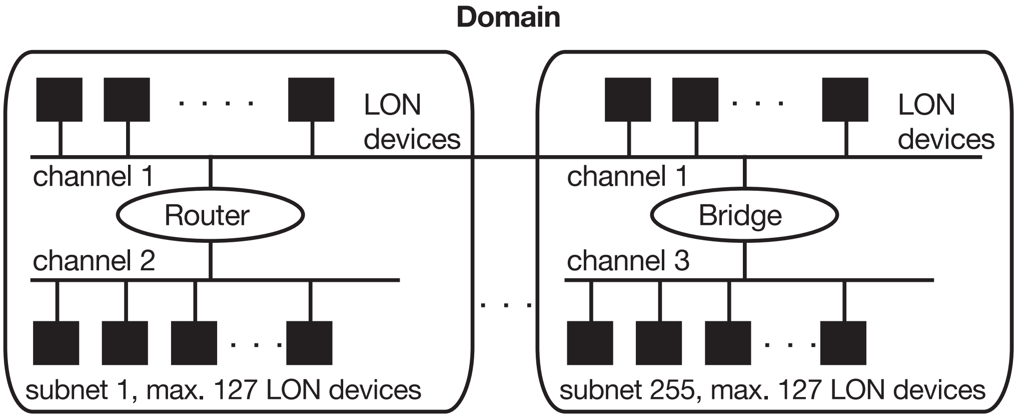


Fig. 93: LON network structure, with coupling components

Quite apart from the opportunities that are provided by the JUMO application of LON technology in the mTRON system, LON also offers the following theoretical forms of network expansion.

**Domain**

The highest level of a network structure is known as a *domain*. A domain represents a logical set of LON devices using one or more physical means of transmission. Each LON device has a 48-bit neuron-ID as its address. So a LON network can contain a theoretical maximum of $2^{48}$ domains. The second level is called a *subnet*. A domain can (theoretically) include up to 255 subnets, whereby each subnet can consist of a maximum of 127 LON devices. Networks that use different physical means of transmission are linked by routers or bridges.

### 3.4.3 Hardware architecture of a LON device

Each unit with distributed intelligence in a LonWorks network is a LON device. Such a device requires the components shown in Fig. 80 for its implementation.

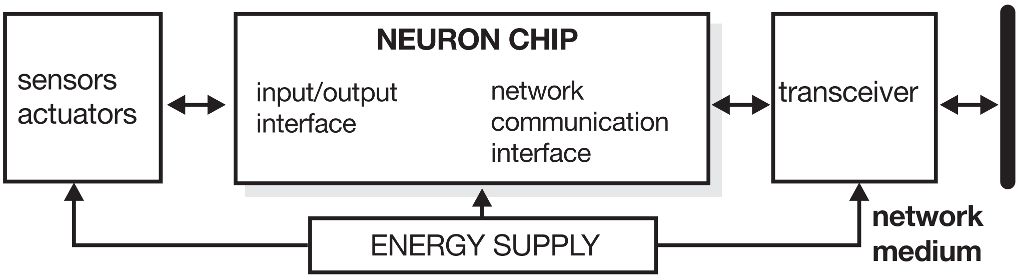


Fig. 94: Block diagram of a LON device

**Neuron chip**

The *neuron chip* is the heart of the distributed intelligence. It contains three 8-bit CPUs for communication and application tasks, ROM, RAM and EEPROM memories, an input/output interface for connection to the instrument application, and a network communication interface that includes the integrated LonTalk protocol.

**Transceiver**

If the network does not cover a large area, then the communication interface of the neuron chip can be connected directly to the network medium (e.g. twisted pair cabling). For substantial distances between the LON devices, and of course where there is a risk of electromagnetic interference in the network cabling, transceivers (combined transmitter/receiver signal drivers) must be used. The application of a transceiver provides electrical isolation between the LON device and the network cabling. Table 23 shows examples of LonWorks transceivers, and the most important data for the user. Other transceiver types are available for such media as: wireless transmission, optical fiber, infrared transmission etc.

| Type | Medium | Data rate<br>(kbits/sec) | Range<br>(meters) | Topology |
|---|---|---|---|---|
| *Twisted Pair* | 2-wire | 78 | | |
| TPT/XF-78 | | 78 | 2700 | bus (linear) |
| TPT/XF-1250 | | 1250 | 500 | bus |
| *Free Topology* | 2-wire | | 2700 | bus (linear) |
| FTT-10 | | 78 | 500 | any |
| *Link Power* | 2-wire | | | |
| LPT-10 | | 78 | 320 | any |
| *Power Line* | 230 V AC | | | |
| PLT-20 | or DC | 4.8 | < 5000 | any |
| PLT-30 | | 2.0 | < 5000 | any |

Table 23: A selection of LonWorks transceivers

The Link-Power tranceiver permits communication and the supply feed for the electrical energy through a common polarized 2-core cable. A switchmode power supply that is integrated into the transceiver provides 5V for the LON device (at max. 100 mA). A centralized power supply produces a supply voltage at approx. 48V DC which powers a bus segment that can be up to 320m long. Link-Power transceivers and Free-Topology transceivers can be freely combined within a network. LON devices which use Free-Topology transceivers require a separate power supply feed.

The Power-Line transceiver permits data transmission over the 230V AC or DC power line. The information is transmitted as a wide-band signal over a large frequency range. Since the pickup of interference signals from devices such as electric motors, switchmode power supplies, solenoid valves etc. is usually over a limited bandwidth, the interference does not affect the transmission of data over the entire modulation bandwidth, and so the data can be reliably received and demodulated.

### 3.4.4 Communication procedure

The neuron chip contains three identical 8-bit processors, which have clearly defined tasks assigned to them (Fig. 95).

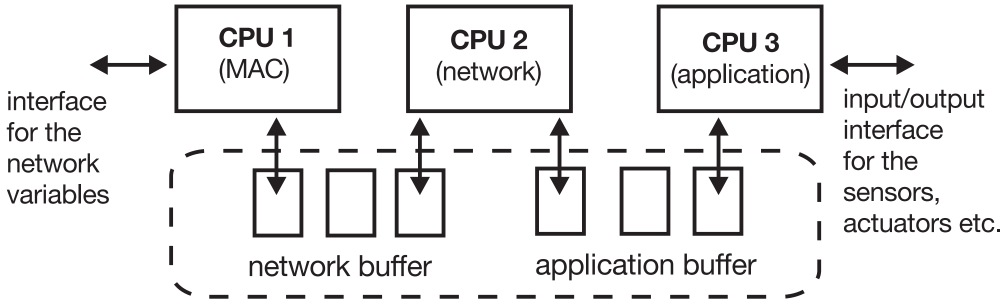

CPU 1
(MAC)
CPU 2
(network)
CPU 3
(application)
interface for the network variables
input/output interface for the sensors, actuators etc.
network buffer
application buffer

Fig. 95: Data communication of the three processors in a neuron chip

CPU 1 is the processor for Media-Access-Control (MAC). It handles layers 1 and 2 of the 7-layer LonTalk protocol. Layer 1 establishes the electrical connection to the attached communication hardware. Layer 2 carries out such tasks as frame checking, data decoding, data security and collision detection. CPU 1 communicates with CPU 2 through the network buffers. CPU 2 is responsible for layers 3 to 6 of the LonTalk protocol. These cover such tasks as address allocation and checking for the network variables, transmission management – for instance, for repeats and feedback of acknowledgements, time monitoring functions, and a lot more. CPU 2 communicates with CPU 1 through the network buffers and with CPU 3 through the application buffers. CPU 3 executes the application program, which is written in Neuron C.

## 3.5 JUMO instruments with Modbus/Jbus

The Modbus protocol is specified at JUMO under the designation *Modbus/ Jbus*. The reason for this is that the Jbus protocol, defined by the French CNOMO organization, is compatible with the Modbus protocol. The structure of the data blocks is identical in both protocols. The difference lies in the absolute address and the data format. With Jbus, the addresses and the formats for the parameters (data) to be transmitted are specified by CNOMO. With Modbus, which is defined by the “Modbus Protocol Reference Guide” from the Modicon company, there is a certain degree of freedom, inasmuch as the addresses are not predetermined.

JUMO implements both applications.

### 3.5.1 Physical interface and data flow

It has already been mentioned that Modbus does not define a particular physical interface, and JUMO uses the RS422/485 serial interface for this application. There are two hardware options available for exchanging data.

The first option implements the communication via a simple interface card with an RS422/485 driver, with direct access through Modbus to the system processor of the automation device (Fig. 96 a).

**Dual-port RAM**

In the second option, the data flow is through an intelligent interface card with its own processor and a dual-port RAM (Fig. 96 b).

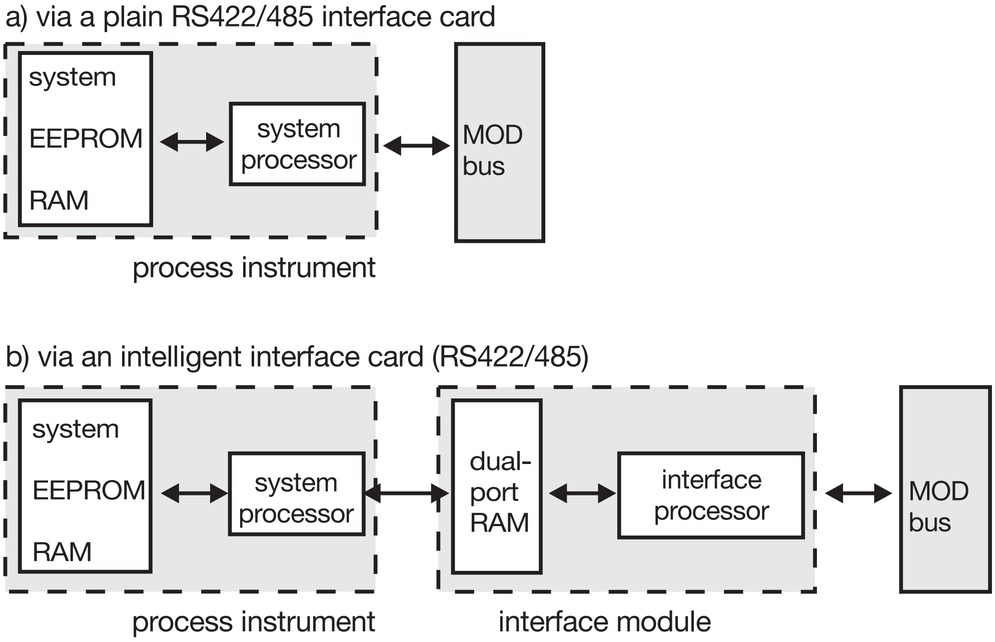


Fig. 96: Data flow

In this version, the system processor writes the process values to the dual-port RAM, for transmission through Modbus. This RAM is divided into two areas: one area for cyclical data and one for acyclical data.

**Cyclical data**

Cyclical data are system variables (e.g. instrument data, process data) which are regularly updated in the dual-port RAM at each sampling cycle. The Modbus driver can read out or write to these variables directly.

**Acyclical data**

Acyclical data are variables such as configuration data, code numbers, mathematical formulae, etc. which are not cyclically updated in the dual-port RAM by the system processor. Data in this area have to be requested by the Modbus driver. They are only available after handling by the system processor.

### 3.5.2 Master/slave principle

Communication between a master (e.g. a PC) and a slave (JUMO process instrument with Modbus/Jbus) is carried out according to the master/slave principle, in the form of a data query/instruction – response (see Fig. 97).

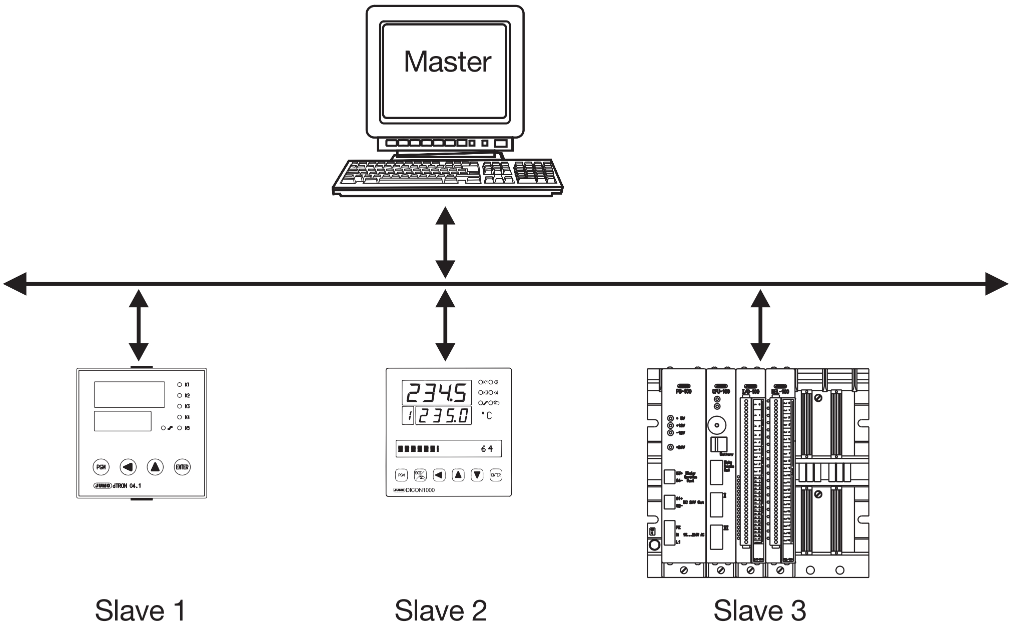


Fig. 97: Master/slave principle with Modbus/Jbus

As already described, the master controls the exchange of data, and the slaves only have a response function. They are identified through their device addresses. The device address can be set from 1 to 255 for Modbus/Jbus. The device address 0 is reserved for a broadcast call to all the slaves.

**At this point, it must be noted that a maximum of 31 slaves can be physically addressed when using the RS422/485 interface, if repeaters are not used.**

Two different methods must be distinguished for the exchange of data between the master and the slaves:

**Query**

A data query/request from the master to a slave, through the slave device address. The corresponding slave will respond.

**Broadcast**

An instruction from the master to all attached slaves, through the device address 0. For instance, a certain parameter (e.g. a setpoint) could be transmitted simultaneously to all the slaves.

### 3.5.3 Transmission mode

**RTU-Modbus**

The Modbus protocol allows two modes of transmission: the ASCII mode and the RTU mode (Remote Terminal Unit). The ASCII mode of operation is not supported by JUMO process instruments.

**LSB**

In the RTU mode, the data are transmitted in binary format (hexadecimal) with 8 bits, 16 bits for integer values, or 32 bits for float values. The LSB (least significant bit) is transmitted first. One great advantage of this transmission method is that the higher packing density results in a better throughput of data than when using ASCII mode at the same baud rate.

A variety of data transmission speeds (baud rates) can be achieved when using JUMO automation instruments (see Table 24).

| Baud rate [bits/sec] |
|---|
| 187k |
| 125k |
| 38400 |
| 19200 |
| 9600 |
| 4800 |
| 2400 |
| 1200 |
| 600 |
| 300 |
| 150 |

Table 24: Possible baud rates

Each byte that is to be transmitted has a characteristic layout, i.e. a defined data format. The adjustable/optional elements in the format are: data word, parity bit and stop bit, as shown in Table 25.

| Data word<br>8 bits | Parity bit | Stop bit<br>1 or 2 bits | No. of bits |
|---|---|---|---|
| x | - | 1 | 9 |
| x | - | 2 | 10 |
| x | even | 1 | 10 |
| x | odd | 1 | 10 |

Table 25: Data format

The individual characteristic interface parameters must first be programmed in the process instruments. These settings for the serial interface can be entered directly from the keypad on the instrument, or from a PC, using the appropriate setup software. In most cases, the standard settings are used: 9600 bits/sec, 1 stop bit, no parity. As well as these parameters, it is also necessary to set the device address (address 0 is reserved for broadcast instructions).

### 3.5.4 Format of the data blocks

The data blocks for communication between master and slave all have the same structure. They are assembled from four fields (Table 26).

| Slave address | Function code | Data field | Checksum<br>CRC16 |
|---|---|---|---|
| 1 byte | 1 byte | n x 1 bytes | 2 bytes |

Table 26: Format of the data blocks with Modbus/Jbus

Slave address: Device address of the slave concerned

Function code: Function selection between reading data and writing data.

Data field: The proper functional information is included here

Checksum: The CRC16 is used to detect transmission errors

**Query-response cycle**

The *query-response cycle* (i.e. the cycle of data query/request and data response between the master and the slaves) produces the following interaction (Fig. 98).

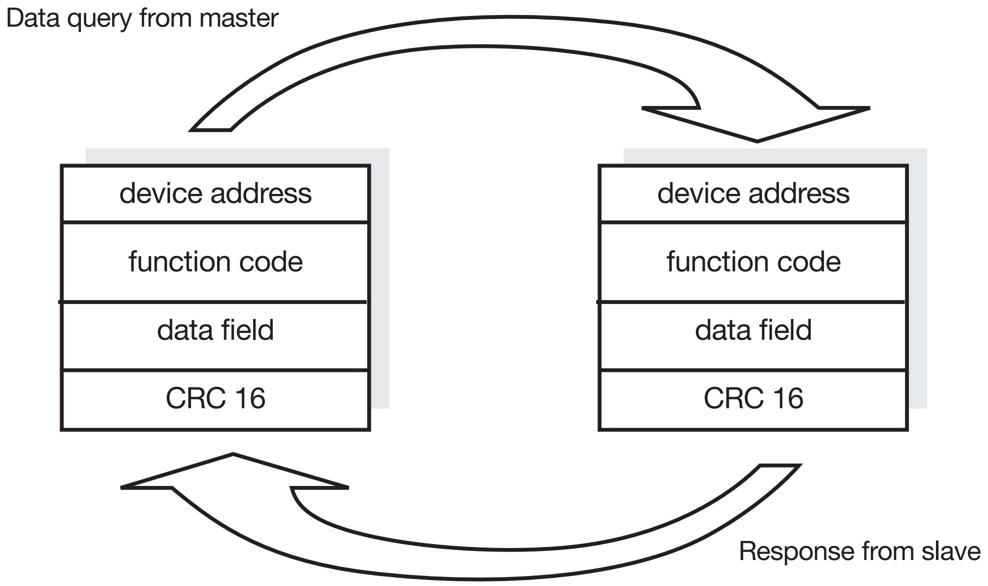


Fig. 98: Query-response cycle

It can be seen, that the slave answers with its device address and the function code that was used by the master, i.e. the data query and the data response have the same structure.

The slave will always respond, unless one of several fault conditions is present:

- the checksum (CRC16) is incorrect
- the instruction from the master is incomplete or incorrectly defined
- the number of words to be read/written is zero

Furthermore, the master must not make any data queries during the internal processing time of the slave. Data queries which are made during this time will be ignored by the slave. Similarly, no data queries may be made over the bus during the response time of the slave. Making a query within this period will invalidate all the data on the bus.

It must be noted that all data (also device addresses, function codes, data addresses, etc.) are presented in hexadecimal form when using the Modbus/Jbus Protocol. Hexadecimal numbers are designated by a prefix “0x” (example: 0x 0010 = 16 decimal).

### 3.5.5 Connection via an interface converter

It has already been mentioned that personal computers are usually fitted as standard with one or more serial communication interfaces (COM1, COM2 ...) that have an RS232 physical interface. If it is necessary to operate a process instrument with a serial RS485 or RS422 interface from a PC, then an interface converter is required. This converts the RS232 signal levels at the interface output of the PC into the levels for RS422/485.

Various items of equipment are available on the market, and the supplier’s wiring instructions must always be observed. In general, the following remarks are appropriate:

1. Most converters require a separate power supply, and are therefore delivered with their own power supply unit. Without this power supply, the equipment cannot function and it will not be possible to establish communication.

**COM ports**

2. The COM ports on a PC are usually implemented as 9-pin Sub-D sockets. The RS232 input on the converter is often implemented as a 25-pin plug. This means that an adapter must be wired up, thus offering the first opportunity for wiring errors to occur.
3. If the RS422 interface is used, then one pair of cores is required for transmitting the data, and one for receiving. As a rule, the transmitting pair (TxD) from the converter is wired to the receiving pair (RxD) of the instrument, and the receiving pair (RxD) of the converter is wired to the transmitting pair (TxD) of the instrument (see Fig. 25/26).

   However, in many converters this crossover of the signals is made internally, so that transmit must be joined to transmit and receive to receive. Once again, the supplier’s information must be carefully observed.
4. With the RS485 interface, the master (e.g. a PC) must perform the changeover between transmitting and receiving, since only one physical channel is available for both directions. On some interface converters this is controlled by the RTS signal (e.g. RTS = high > transmit; RTS = low > receive). If this changeover is not carried out, errors will occur. To avoid this source of errors, it is recommended that the RS422 interface is used.

Fig. 99 shows an example of the wiring for operating a JUMO process instrument with an RS422 serial interface via the COM port (V.24) of the computer. In this example, the converter used is the type PSM-V24/V11/BB from Phoenix Contact.

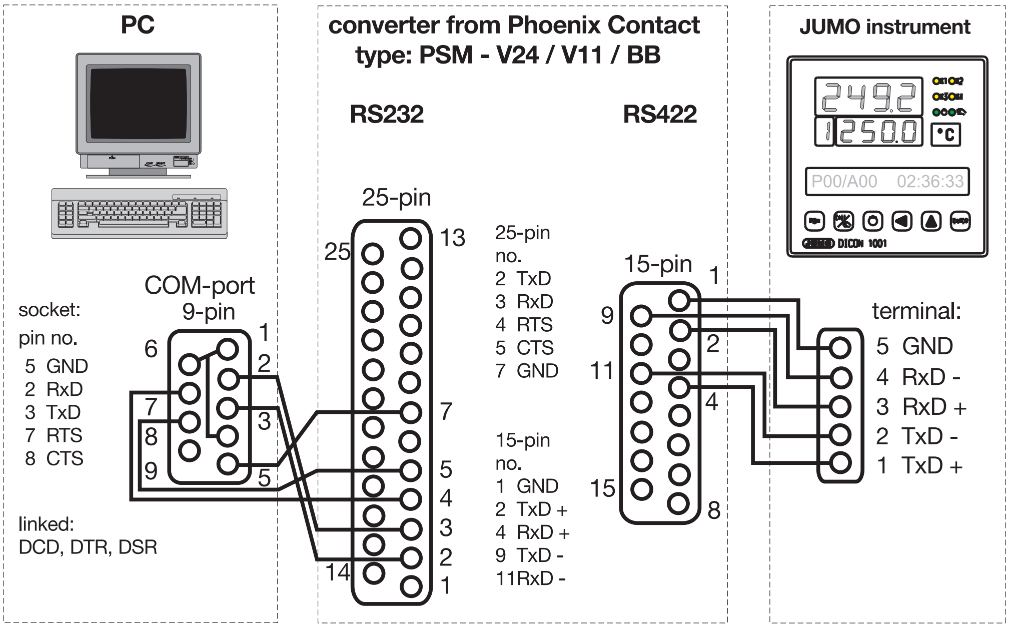


Fig. 99: Wiring a PC to a JUMO instrument via an RS232/RS422 converter

## 3.6 JUMO instruments with PROFIBUS

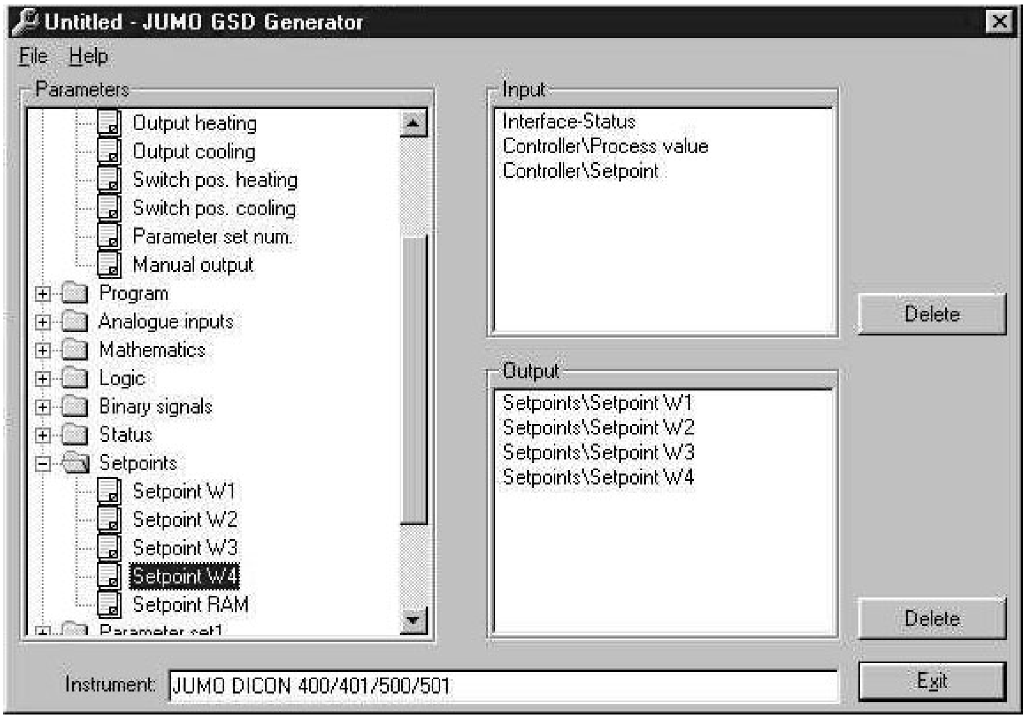
JUMO supplies several instruments with PROFIBUS as well a “DP to LON” gateway that converts the protocols from PROFIBUS-DP to LON and vice versa. The “DP to LON” gateway can be used to connect all JUMO mTRON instruments to PROFIBUS.

**DP slave GSD file**

The procedure described below for the creation of the GSD file is not valid for the “DP to LON” gateway that is available from JUMO. The GSD file for this gateway cannot be created with the GSD generator. JUMO supplies a special GSD file in this instance.

The following example describes the connection of the JUMO controller type DICON 40X/50X JUMO to an existing PROFIBUS system:

In order to couple the controller onto the PROFIBUS, the corresponding GSD file will be needed. JUMO delivers a GSD generator for the creation of the GSD file (see Fig. 100). After starting the GSD configurator, the user merely has to select the instrument that is to be coupled on. The DICON 40X/50X controller has a large number of input and output parameters. The user applies the GSD generator to select only those parameters which are relevant for his application, and only these will be transmitted across the PROFIBUS.



Input
Interface-Status
Controller\Process value
Controller\Setpoint

Output
Setpoints\Setpoint W1
Setpoints\Setpoint W2
Setpoints\Setpoint W3
Setpoints\Setpoint W4

Instrument: JUMO DICON 400/401/500/501

Fig. 100: Generating a GSD file for a DICON 40X/50X controller

A PC configuration program (PLC manager: a part of the PLC) is available for the configuration of a master (see Fig. 101). The configuration of the PROFIBUS system can be set up with this program and transmitted to the PLC. The previously generated GSD file is stored in a defined path in this configuration program. After starting the configuration program, the DICON 40X/50X can be added to the device catalog.

The DICON 40X/50X can now be chosen in the device catalog and receive a permanent hardware assignment to the PLC (see Fig. 101). Finally, the PROFIBUS device address is allocated.

Depending on the version, the PLC can have a larger or smaller I/O memory. The memory area allocation for the input and output data for DICON 40X/50X must also be made in the configuration program. Later, the PLC programmer will work with these addresses.

After loading the program from the PC to the PLC, the parameterization data that are contained in the GSD file will be transferred from the PLC to the DICON 40X/50X.

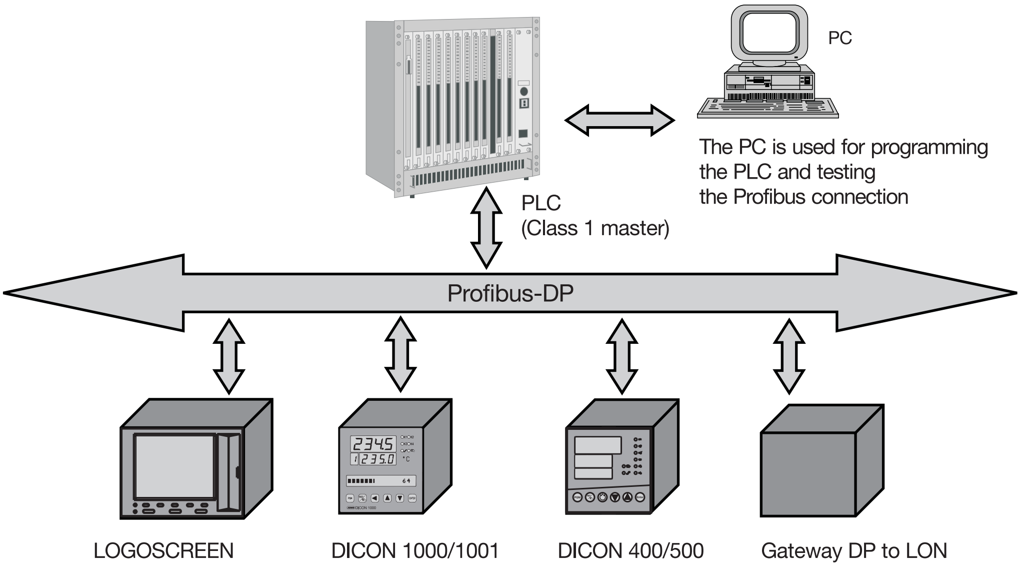


Fig. 101: Configuration of a PROFIBUS system

## 3.7 Checklist for fault-finding in serial interfaces

When working with process instruments via digital interfaces, you will frequently be disappointed to discover that the first attempt at communication does not produce any result. The reason could be anywhere: hardware, software, cabling, etc. In most case extensive investigation is required to localize the fault. About 80 % of the time the reason is to be found in faulty wiring.

The table below presents some typical fault conditions that occur again and again when operating with digital interfaces. It has been collected from everyday experience at JUMO, to a large extent from user feedback. The checklist is therefore based on the software and hardware used by JUMO, but certainly has some general application.

| Source of fault | What could be the cause? |
|---|---|
| Interface parameters | ... are they set the same in the master and the slave? (data format, baud rate, word length, parity, stop bit(s)?<br>... echo mode (on older equipment) may have to be switched off (terminal=off) |
| Wiring or connection of the instruments | ... is it correctly connected?<br>... are any cables mixed up?<br>... is the connector pin assignment correct?<br>... does the master end address the correct interface (COM1, COM2 ...)? |
| Handshake | RTS and CTS must not be linked in JUMO instruments (only relevant for RS232)<br>RTS and CTS may need to be linked at the master end |
| Interface mode | ... have you set the correct mode on the device? |
| Syntax, protocol | ... is the correct syntax being used? (options are: ASCII, setup, Modbus/Jbus) |
| Device address | ... have you set the correct address on the device?<br>... is there another device on the bus with the same address?<br>... if you set the device address to “00” with Modbus, there will be no reponse (broadcast mode)<br>... if the checksum was incorrectly calculated, there will also be no response |
| End character (for ASCII) | ... has the instruction got the correct termination character (e.g. “CR“ = 0DH)?<br>... try leaving out the “LF” (possible problem with RS485) |

| Source of fault | What could be the cause? |
|---|---|
| Initialization before each instruction (ASCII) | ... if you are using it – is it the correct initialization character (“EOT”=04)? |
| Interface converter | ... is your converter OK? (check whether characters are transferred before/after the converter!)<br>... with RS422: try linking transmit/receive lines to test the converter!<br>... is the power supply unit switched on?<br>... is the switch setting correct as per the manual? |
| Interface type | ... is the assignment RS232, RS422, RS485 between master and slave correct?<br>... if you are using RS422 or RS485, are you waiting for at least 20ms after the slave response before sending a new instruction to the slave?<br>... if using RS485, is the changeover between transmit and receive functioning at the master end? (if communication via RS485 is not working, try RS422. It only requires connecting two more lines)<br>... take care with the universal interface card! (select RS422 or RS485 by solder link or jumper) |
| Instrument software | ... are you using the correct software version? |
| Interface cards | ... is the interface card on the master OK?<br>... is the interface card on the slave OK? |
| JUMO; universal interface | ... cyclical data can be read directly!<br>... acyclical data can only be read indirectly, i.e. the data exchange between the system-EEPROM and the dual-port RAM of the interface must be started by an additional command! |
| Representation protocol | ... hexadecimal/decimal not correctly observed? |
| Data format | ... are the data available in the expected format (string, bit, integer, float)?<br>... are data types with more than one byte in the correct sequence? Some processors may require exchanging High and Low bytes! |
| Modbus address | ... many Modbus drivers on the market put out an address on the bus that is one lower than entered. Adjust for this when making the entry! |

Table 27: Checklist for fault-finding

## 4.1 Standards and technologies in automation engineering

The advances made in microelectronics and in information technology in general, have also altered the technical boundary conditions in the field of automation engineering. The possibilities for using information technology to support and optimize automation processes appear almost unlimited. It was already recognized some years ago, that in the systems engineering sector, where various subsystems are joined together to perform a higher-level task, the open publication of the interface specifications leads basically to an improvement of the functionality of the entire system. User acceptance is increased, and this opens up additional markets.

This chapter describes the application of Ethernet at the field level, as well as systems with open interfaces.

### 4.1.1 NOAH (Network Oriented Application Harmonization)

During the commissioning of fieldbus systems, considerable effort and expense sometimes arises, because the tools that are needed (e.g. for configuration or commissioning) have difficulty in operating together, or have special features which are not supported by all the devices. In heterogeneous systems, a unification mode of operation for communication links, independent of the fieldbus and manufacturer, would lead to increased effectivity. The aim of the EU joint project NOAH is therefore to create a generally valid interface, which can be used for the configuration and commissioning of field devices to EN 50 170 through a single software tool (see Fig. 102).

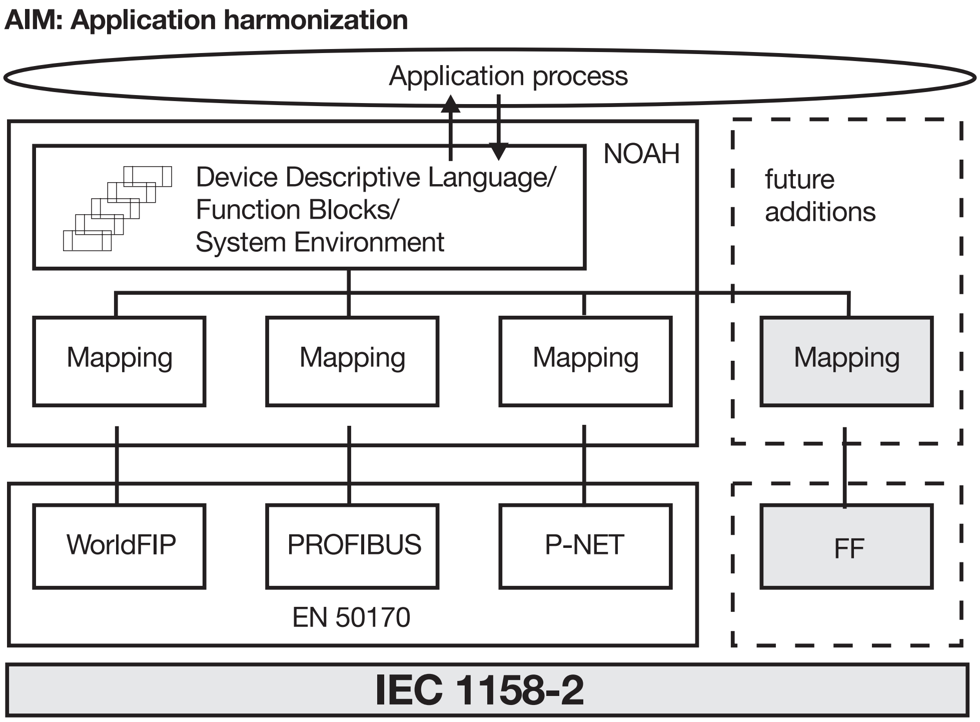


Fig. 102: The NOAH project

NOAH will use the application layer of the ISO/OSI reference model to ensure that the application is always accessed in the same way (parameter setting, diagnosis, functions) – regardless of the fieldbus on which it is running.

Automation equipment, such a sensors and transmitters, actuator drives, simple I/O devices etc. have a clearly defined basic range of functions to perform, derived from the requirements of a particular industrial sector. However, the implementation of these functions may be very different from one manufacturer to another. This means that a free choice between comparable products is hardly possible, since a device which might otherwise be economically or functionally favorable could prove to be impractical, or cause a number of consequential alterations if the type of device is changed. Practical experience with fieldbus-specific equipment profiles has shown that, typically, only a limited portion of the required device functionality can be standardized. For this reason, it is necessary to provide electronic data sheets for every device that offers an extension to the scope of the profile. These electronic device descriptions must be built up on a standardized descriptive method. Defining this descriptive method is part of the aims of NOAH.

As a result of the considerable delays in defining a unified international standardized fieldbus within the IEC, the European Standard EN 50170 “General Purpose Fieldbus Communication System” was finalized by CENELEC (European Committee for Electrotechnical Standards) in 1995. This standard includes (and thus favors) the PROFIBUS, FIP and P-Net fieldbuses. Since NOAH, in effect, is intended to create a unified application layer for fieldbuses to EN 50170, the activities of NOAH are a consistent continuation from EN 50170. The organizations participating in NOAH were able to commence work at the start of 1998, intending to present the results during 2000.

The results of the NOAH project will be significant for both equipment manufacturers and system providers. The profiles which are to be defined in NOAH could achieve considerable significance as an European Standard. System providers would then have the opportunity of unrestricted integration of field devices with a NOAH profile into the present-day manufacturer-specific systems control technology.

### 4.1.2 OPC (OLE for Process Control) for communication

In modern production plants, electronic information processing at various levels is growing together more and more (see Chapter 1.6.1, “Communication networks and levels”). Programs at various levels must communicate with one another, and this usually involves different types of automation components from different manufacturers, as well as different bus systems.

The transfer of data from one program to another and the interconnection of the automation components is set up by software specialists, and is very expensive.

This was the reason for the definition of *OLE for Process Control* (OPC), a standardized interface on the basis of COM/DCOM from Microsoft.

**OLE**

OLE (Object Linking and Embedding) is a software interface, which can link information from one program into another application and display it.

**COM/DCOM**

COM (Component Object Model) is the standard for the operation of components with one another.

For software calls outside the host computer, the term used is DCOM (Distributed COM)

**OPC Foundation**

OPC was developed by the OPC Foundation (formed in 1996), in which Microsoft is also an active participant. The OPC Foundation is a collaboration between manufacturers of automation and visualization components which supports users who wish to develop programs on this basis.

**OPC**

If a software house provides software which is to be linked to automation components from another manufacturer, then it develops a software driver for these components. This means that more than one software driver will be created for a single device – indeed, there will be one from each software provider who uses that device in their program. These drivers do not support any expansion of the device functions, and would require modification in such cases. A simultaneous access of the device by two software applications is also not possible, since they would be accessing via two separate drivers, with a resulting software conflict.

These days, many programs in the industrial information processing sector are developed on personal computers using the MS-Windows platform. Designs based on Windows technology with OPC provide a unified access procedure and a standardized interface.

The aim of OPC is to couple components from different manufacturers into one program, as an automation solution, without the programmers having to be concerned about communication with individual automation components.

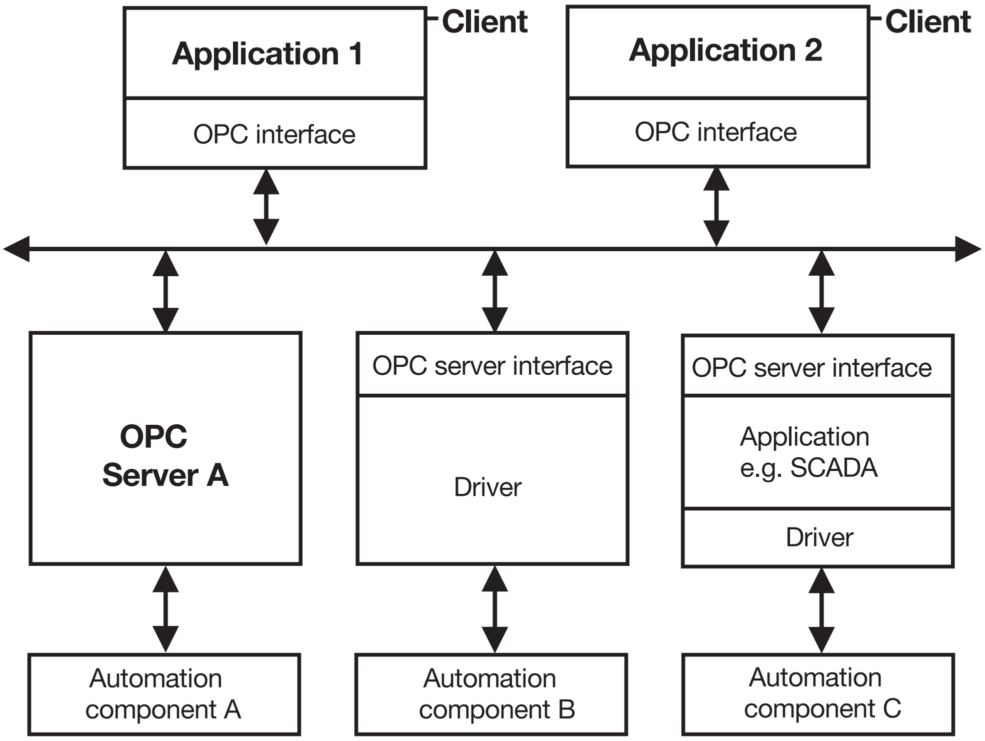


Fig. 103: OPC$^{TM}$ interfaces in an automation solution

**Mode of operation**

OPC uses the Client-Server model. Programs that require data from the automation level request this data through *services*, so they are *clients*. The automation components, on the other hand, make the information available through OPC, so they are servers. A client application only requires a standardized OPC interface in order to be able to communicate with any other OPC-capable server, such as are provided by various suppliers of automation components, to be able to communicate with these components. An OPC interface is described precisely in the OPC specification.

As OPC servers one primarily thinks of drivers which are implemented as OPC servers, to exchange data with the automation components through communication system links (see Fig. 103). But program packages which have preprogrammed functions, such as SCADA systems, can also be OPC servers. The automation components can be linked to the OPC system through the usual driver techniques, as before. A client application can then request the appropriate data for further processing through the OPC interface of such an OPC server. In this way, it is only necessary to add an OPC interface to an existing program or device driver to create an OPC server.

The services which are offered to the OPC clients from the servers, via the OPC interface, are typically: reading, altering and monitoring process variables.

With OPC, industrial software developers are striving to achieve something similar to the standardization of the Windows printer interface. These days, printers can be connected to personal computers by all kinds of hardware; these printers are declared to the Windows system by the installation of a driver, and are from then on available for all Windows programs, without the particular application requiring any information other than that provided by the driver. OPC behaves in a similar fashion: once the Windows installation is known, then the OPC services are available for all programs that can make use of them.

Since years of effort have passed without a unified fieldbus being defined, OPC offers a possibility for integrating automation components from various suppliers at a higher level, in the software which handles the communication with the components. Even if the communication and fieldbus systems differ, a common OPC interface makes it possible to incorporate the variegated automation system components which are available on the market into a system solution in an economic way (see Fig. 104). The basic idea is that each manufacturer of automation components should also provide a standardized driver, to enable the integration of their components into a system configuration.

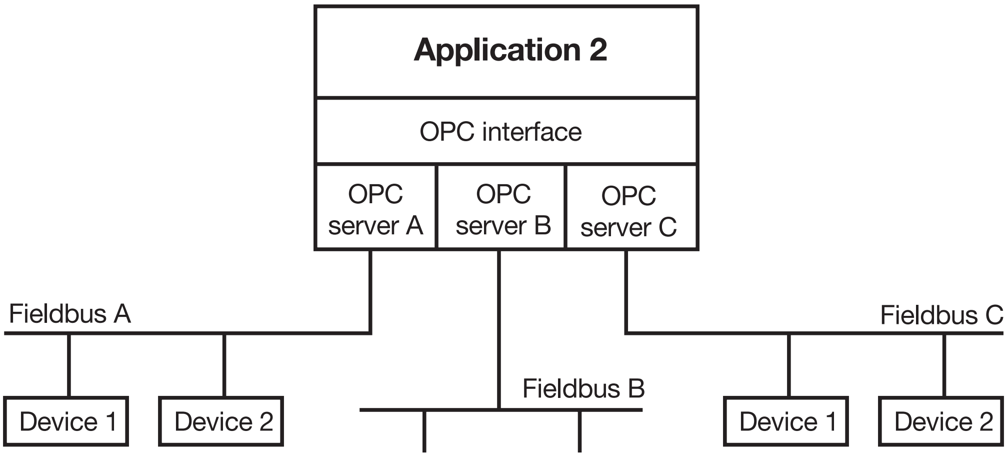


Fig. 104: OPC in the fieldbus environment

### 4.1.3 Ethernet fieldbus equals system bus

For some time now, companies have been developing systems which utilize Ethernet at the field level. Ethernet (already described above) offers the following advantages compared with the classic control system structures:

- high transmission capacity (now up to 100 Mbit/sec with Fast Ethernet)
- almost no restrictions on the network topology
- with switches, practically unlimited network expansion is possible
- established standard
- link to the world of information technology, world-standard TCP/IP
- considerable worldwide development effort in this technology

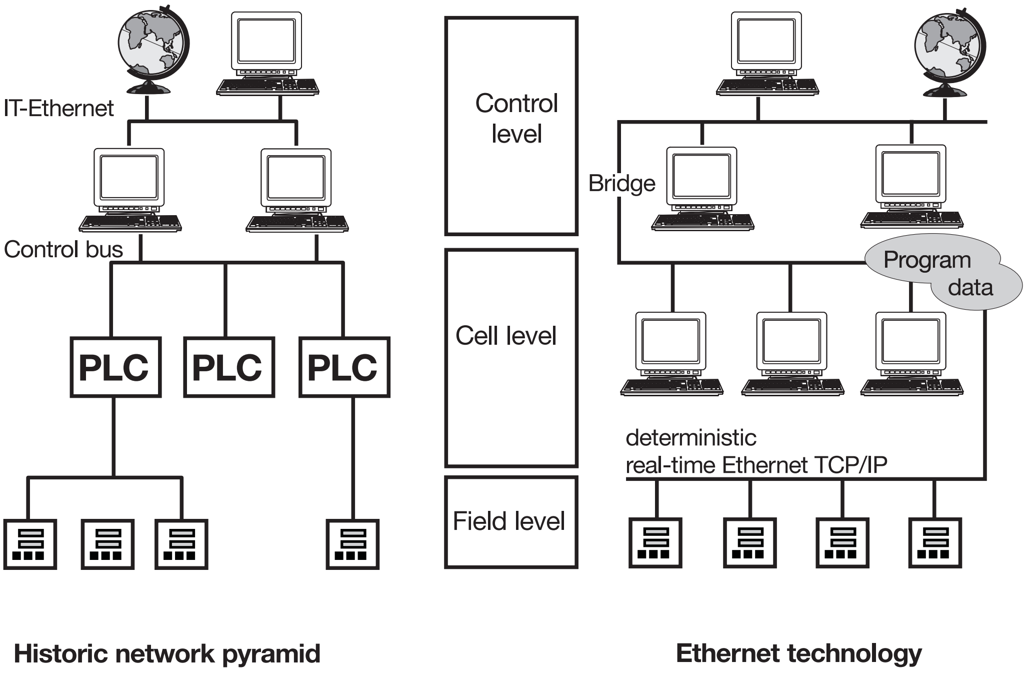


Fig. 105: The historic network pyramid and Ethernet technology

The traditional systems consists of CPUs and slave modules connected through an internal system bus. Data have to pass through several CPUs between the sensor and the control system levels. A distinction is made between internal and external data traffic.

If the system is implemented as an Ethernet network, then Ethernet forms part of the control system and thus becomes the system bus. There is no longer a distinction between internal and external data traffic (see Fig. 105).

**TCP/IP**

The protocol which is used in these networks – TCP/IP (Transmission Control Protocol/Internet Protocol) – is defined in the TCP/IP reference model, and is a standard for data transmission in networks and the standard for the Internet. Since this is the protocol that is used for Ethernet networks, it enables a transparent communication with the world of IT and the Internet.

A switch is incorporated into each control system, to provide a physical separation between the physical network (domain) and other networks, while maintaining the logical connection. A long as communicating partners are within this control area, their data does not go outside the domain, and so it does not load the entire network. Data that have to be transmitted across this boundary can be buffered in the switch until the outgoing connection into the network is available. All the network data which are not intended for the domain will not pass beyond the switch. This means that the data traffic within the domains can be estimated, and the Ethernet has real-time capability within this section of the network.

## 4.2 Long-distance data transmission

**Long-distance data transmission**

Long-distance data transmission or remote control is understood as a regulated and continuous monitoring (or the facility for remote parameter setting and operation) of a distant process. Up to now, remote control and operation was, as a result of the very high implementation costs, primarily reserved for large-scale distributed process control systems, such as public utility supplies for gas, electricity, water, and long-distance heating.

But these days, more and more users who have strongly decentralized processes require long-distance data transmission for a modern fieldbus concept and/or a visualization/monitoring or control system. Advantages such as reduced wiring requirements, centralized operation and monitoring directly from the control room are no longer enough. Users are looking for options for remote parameter setting, diagnosis and monitoring of a system from a central but remote location, without having to be continuously on site.

**Data exchange**

Long-distance data transmission enables the exchange of data between IT systems. For instance, a branch office can regularly send turnover figures to the company's central computer, so that automatic deliveries can be initiated if stocks are below a minimum threshold, or statistical evaluation can be carried out.

**Working on a remote host computer**

Long-distance data transmission not only offers the facility of directly transferring data between two computer systems. It is also possible, via a remote connection, to become a participant in a remote network or to work on a remote host computer (Remote Access). In this way, field sales staff can access company data at any time or a *Home Office* workplace can be set up.

**Networked computers**

Long-distance transmission can be used to set up networks of computers which permit worldwide cheap access to IT systems or the exchange of information between computer users. Such networks are available on a commercial basis (e.g. CompuServe) as well as on a non-commercial basis (Internet). These networks are experiencing enormous growth at the moment.

The integration of remote control functions into modern automation systems and fieldbus concepts makes it possible to cover an ever wider market for small and medium installations in industrial automation. Examples are decentralized metering and measuring points for boreholes, pumping stations, weather stations, water treatment plants (rainwater reservoirs), and so on. An application for mobile remote monitoring of cooling/storage rooms is shown in Fig. 106. Here a mobile phone function has also been integrated to enable continuous monitoring.

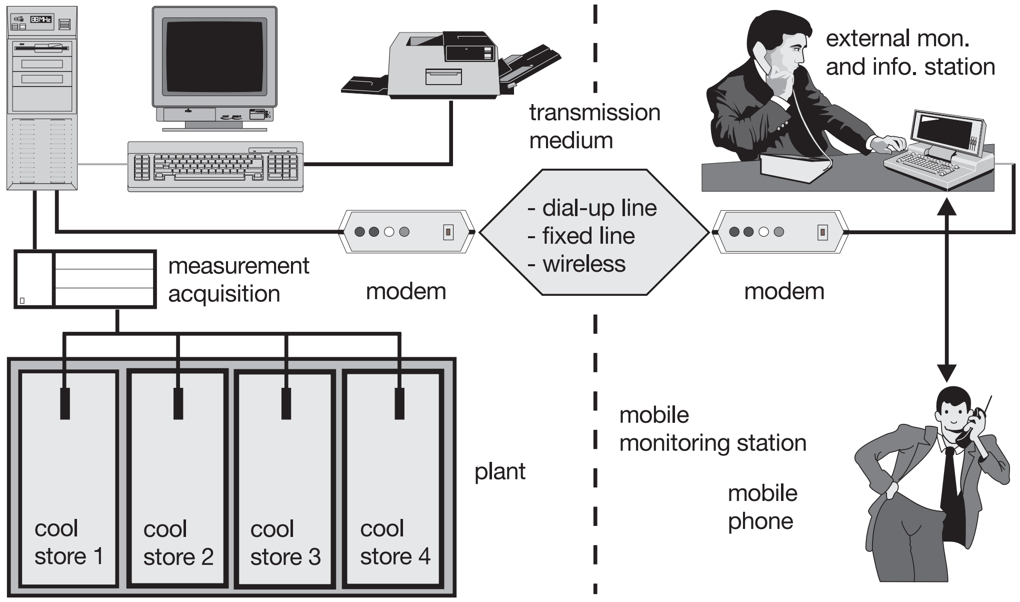


Fig. 106: Decentralized process monitoring

If a threshold is exceeded at a measurement point, this triggers a mobile phone call. The person carrying the mobile phone can then use a PC to call up the current process data from the external monitoring and information center. Faults can thus be analyzed very quickly, and countermeasures to eliminate the fault can be initiated without delay.

For some years now, transmission via a modem and the public telephone network has established itself as the medium for long-distance data transmission. The distance between the decentralized station and the control center is no longer decisive. Two fundamental modes of operation can be distinguished with this method:

**Dial-up operation**

If a permanent connection is not necessary, and only small quantities of data have to be transmitted, at lengthy intervals, e.g. for alarm signals, then dial-up operation is recommended. In this type of operation, a master station calls another station only when it is necessary to transfer data.

**Fixed line**

If a lot of data have to be transmitted at frequent intervals, such as process data for archiving or further processing, then a fixed connection is recommended. In this case, there is a permanent connection between the participants. If the public telephone network is used, then the fixed lines must be rented, which results in higher running costs.

In addition to these modes, wireless transmission is becoming ever more important. The lack of telephone sockets and the unavailability of fixed lines often means that wireless is the only feasible transmission medium. The continually sinking costs are another factor in favor of this technology.

Today, these options for long-distance data transmission allow the operation and control of decentralized installations and systems, giving the user considerably more flexibility when planning the installation site.

## 4.3 Distributed systems

The microelectronics which is available for modern automation systems, and the various communication facilities, are strengthening the trend towards the use of decentralized intelligent field devices and systems. The term distributed intelligence or distributed intelligent controls means the division of the individual control functions among several networked systems. In this case, the complete system must be divided into suitable functional elements that can work together.

Increasing requirements and the individual technical possibilities are today resulting in ever more complex systems. The reasons include:

- an increasing number of sensors and actuators
- extensive exchange of data with other systems
- shorter reaction times
- voluminous data acquisition for records or quality control
- complicated operating and control algorithms, etc.

It is well known that the effort required to manage such complex systems (planning, project management, programming, installation, operation, maintenance, documentation etc.) increases out of all proportion to the complexity. The resulting costs can only be reduced by reducing the complexity, by using effectively subdivided components which are networked together.

If a suitable application is available, and the task is properly divided into autonomous function blocks, then the user will reap the benefits:

- the resulting open system structure leads to reduced costs
- the system structure can be more flexibly expanded or adapted to changed requirements
- the maintenance and diagnosis facilities are improved, and system downtimes are reduced accordingly
- the achievable productivity is increased.

However, there is no doubt that the increase in distributed systems, individual intelligent components (field devices) and their matched interoperability will become more important. Tools for the planning, programming, commissioning and maintenance of the installations must combine the individual solutions and variegated components into a complete system. This has to be implemented through an open fieldbus technology. A number of innovative steps are still required, but they will then contribute to the rationalization of manufacturing and production.

# Index