Fachhochschule Düsseldorf

Fachbereich Elektrotechnik

University of Applied Sciences
Process Control Laboratory

INTERBUS

BASICS

For more information about INTERBUS remote control practical exercises, please refer to http://www.telepraktikum.de.

Prof. Dr.-Ing. Reinhard Langmann

FH Düsseldorf
Process Control Laboratory
http://www.teleautomation.de
E-mail: R.Langmann@t-online.de

1 The History of INTERBUS

In 1983, Phoenix Contact began work on a specification for an industrial fieldbus. Work continued together with computer manufacturers and technical colleges, and a protocol and hardware definition for a realtime-capable sensor/actuator bus was developed and presented at the Hanover Fair in 1987. The primary field of application of this bus system - originally known as *Interbus-S* - was to be the speed-optimized time-deterministic transmission of sensor/actuator data (process data). Over the next few years, an extensive and varied range of bus components and field devices would be developed by Phoenix Contact and other manufacturers. Interbus-S developed into one of the world's leading fieldbus systems in industrial automation.

As the use of open control systems and the development of PC-based automation solutions increased from around the mid-nineties onwards, the bus system was renamed *INTERBUS* and a determined focus was placed on distributed automation using factory-wide uniform and standardized communication structures. Milestones in this phase of its development include:

1995 Development of INTERBUS Loop (also known as the sensor loop, installation local bus) as a logical extension of INTERBUS "downwards", towards the technical process, for the direct connection of sensors/actuators via a two-wire cable

1996 Launch of the new G4 modules (G4 = Generation 4) for connecting INTERBUS "upwards" with the office world (PC, Ethernet, TCP/IP); founding of the Open Control User Group

1997 Launch of INTERBUS INLINE, another component based on INTERBUS technology for individual and networked automation solutions

The advent of industrial Ethernet technology in the year 2000 considerably encouraged the integration of INTERBUS and Ethernet/TCP/IP for the creation of consistent communication structures for industrial automation.

In parallel with the technical development and functional expansion of INTERBUS, manufacturers and users were becoming involved in a range of complementary activities. These included the foundation of the INTERBUS Club e.V. user organization in Germany in 1992 and the development of application profiles (this started in 1992 with the DRIVECOM profile for electrical drives) by working groups in the INTERBUS Club.

In 1993, the INTERBUS Club started to issue a certification symbol for INTERBUS devices to indicate conformance and interoperability.

In 1998, INTERBUS became the world's leading fieldbus system with a market share of 37.4%. More than 2.5 million devices are in use and 2000 device types are available from 1000 manufacturers.

2 General Characteristics

2.1 Main Task

Modern automation requires a continuous flow of information and support for open and flexible control architectures. INTERBUS technology provides an open fieldbus system, which embraces all the process I/Os required for almost any control system.

INTERBUS can be used to connect sensors and actuators via a serial bus cable, to control machines and system parts, to network production cells, and to communicate with control rooms, as well as in production data and machine data acquisition (PDA/MDA). This means that INTERBUS is able to fulfill essential requirements of high-performance control concepts, as it is:

- A cost-effective solution with bus systems, which transmit data serially and reduce the amount of parallel cabling required
- An open and manufacturer-independent networking system, which can easily be connected with existing control systems
- Flexible with regard to future modifications or expansions

INTERBUS is installed in the system to be automated as a compact, single-circuit line following one direction. A controller board provides the interface between a PLC or an industrial PC (IPC) and INTERBUS. The bus system connects all the I/O components present in the system (also known as *INTERBUS devices*) with the control/computer system via a controller board.

The INTERBUS protocol, which has been optimized specifically for the requirements of automation technology, transmits both single-bit data from limit switches or to switching devices (process data) and complex programs or data records to intelligent field devices (parameter data) with the same level of efficiency and safety. Figure 1 illustrates the basic structure of an INTERBUS PLC.

Process data is transmitted in the fixed and cyclic time slot in realtime conditions, while *parameter data* comprises the acyclic transmission of larger volumes of data as and when required.

The continuity of an INTERBUS network for very different tasks within an automation system - ensured in essence by the standard protocol - is supported by additional measures: These include:

- The adaptation of the physical transmission method "downwards", making it easy to install and connect individual sensors and actuators
- The provision of "upwards" interface couplers to connect INTERBUS networks directly with factory and/or company networks (Ethernet networks)

- The guarantee of easy configuration, project planning, and diagnostics with uniform software tools

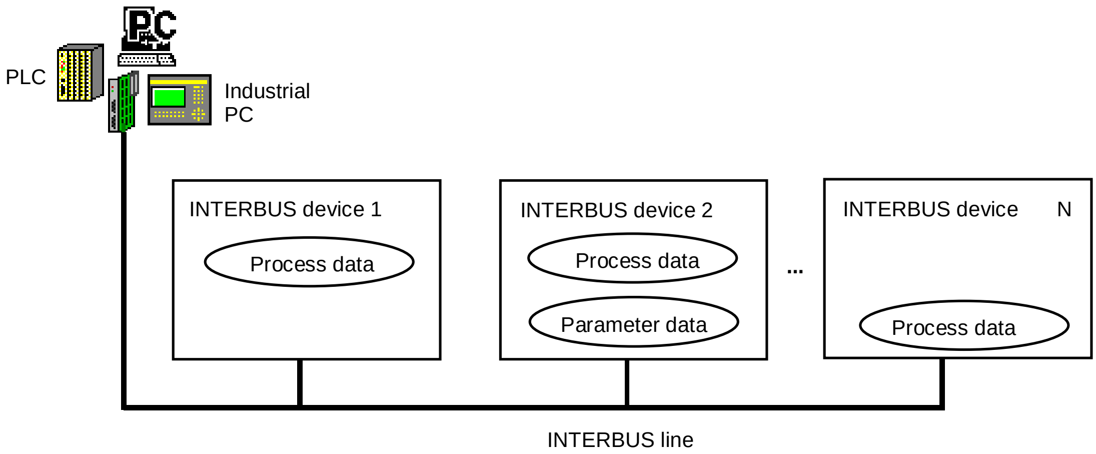


Figure 1: Basic structure of an INTERBUS control system

2.2 Area of Application

With its special features and an extensive product range, INTERBUS has established itself successfully in all sectors of industry. Its traditional field of application is the automotive industry, but INTERBUS is also increasingly being used as an automation solution in other areas such as materials handling and conveying, the paper and print industry, the food and beverage industry, building automation, the wood-processing industry, assembly and robotics applications, general mechanical engineering and, more recently, in process engineering. Today, INTERBUS is used in over 125,000 applications throughout the world.

In addition to standard applications for connecting a large number of sensors/actuators in the field to the higher-level control system via a serial bus system, INTERBUS can also be used to fulfill a variety of special application requirements. The examples below illustrate some of its typical applications.

The Use of Optical Fibers

For applications in critical environmental conditions or environments that are subject to electromagnetic interference, the serial INTERBUS cable can be replaced with optical fibers. Depending on requirements, users can select either a copper or optical fiber transmission medium without having to make any changes to the network topology or system structure. Both transmission media can be combined as desired in the network without restriction.

Distributed Power Distribution

INTERBUS offers distributed power distribution, i.e., control or parameter data transmission coupled with power transmission. Modules have been developed which, in addition to the INTERBUS function, are also able to transmit the three-phase 400 V AC supply voltage or a

24 V DC voltage from module to module via an additional cable. This eliminates the need for the star power cabling that was previously used as standard (Figure 2).

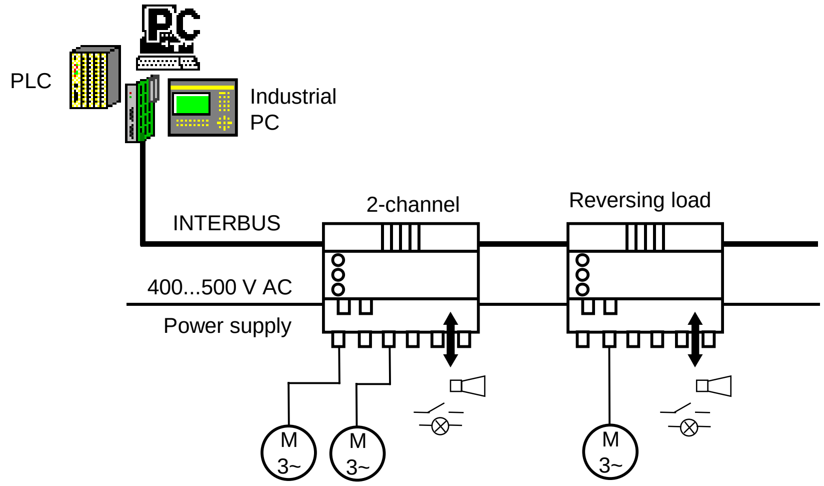


Figure 2: Distributed power distribution with INTERBUS

Drive Synchronization

Many process engineering applications require drive synchronization. An example of this type of application is a mill train with three immediately adjacent rollers (Figure 3).

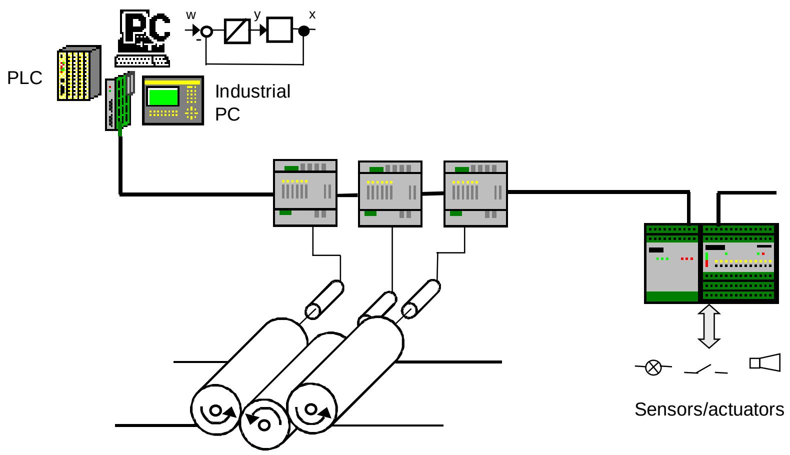


Figure 3: Control loop application in a mill train

Synchronization and the resulting constant material throughput are essential in this type of application. An INTERBUS control device is able to synchronize the drives by synchronizing the INTERBUS cycle and the control cycle.

Dynamic Changes to the Bus Structure

Depending on the application, the bus configuration may change dynamically during an active process. An example of such change is the docking and undocking of tool carriers on a machining center. This requires the activation of various bus parts at one connection point (Figure 4).

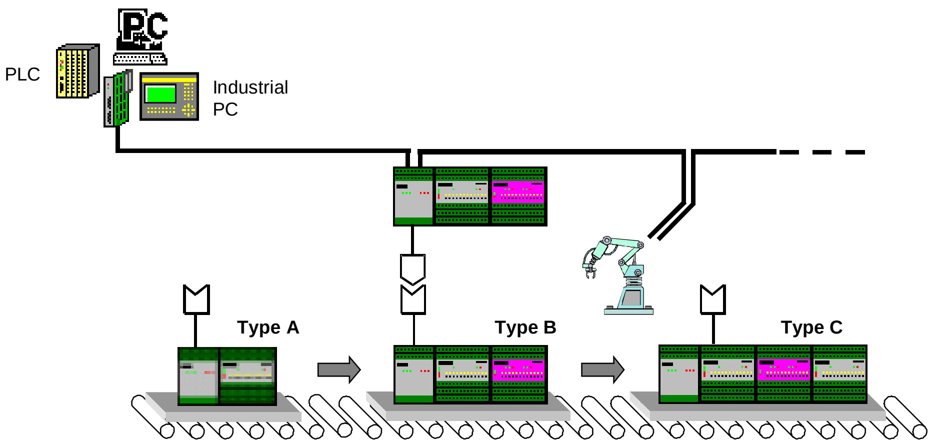


Figure 4: Alternative and changing bus configuration in a machining center

INTERBUS is used to configure the entire configuration (including all alternatives) and save it to the controller board. The selection of the alternative option to be started is controlled via the application program.

2.3 Method of Operation

INTERBUS works with a *master/slave access method*, in which the master also establishes the connection to the higher-level control or bus system. In terms of topology, INTERBUS is a *ring system* with an active connection to communication devices.

Starting at the INTERBUS master, the controller board, all devices are actively connected on the ring system. Each INTERBUS device (slave) has two separate lines for data transmission: one for forward data transfer and one for return data transfer. This eliminates the need for a return line from the last to the first device, necessary in a simple ring system. The forward and return lines run in one cable. From the installation point of view, INTERBUS is similar to bus or linear structures, as only *one* bus cable connects one device with the next.

To enable the structuring of an INTERBUS system, subring systems (bus segments) can be formed on the main ring, the source of which is the master. These subring systems are connected with *bus couplers* (also known as bus terminal modules). Figure 5 illustrates the basic structure of an INTERBUS system with one main ring and two subring systems.

The *remote bus* is installed from the controller board. Remote bus devices and bus couplers are connected to the remote bus. Each bus coupler connects the remote bus with a subring system. There are two different types of subring system, which are available in different installation versions:

- The *local bus* (formally known as the I/O bus) is responsible for local management, connects *local bus devices*, and is typically used to form local I/O compact stations, e.g., in the control cabinet. It is also available as a robust version for direct mounting on machines/systems.

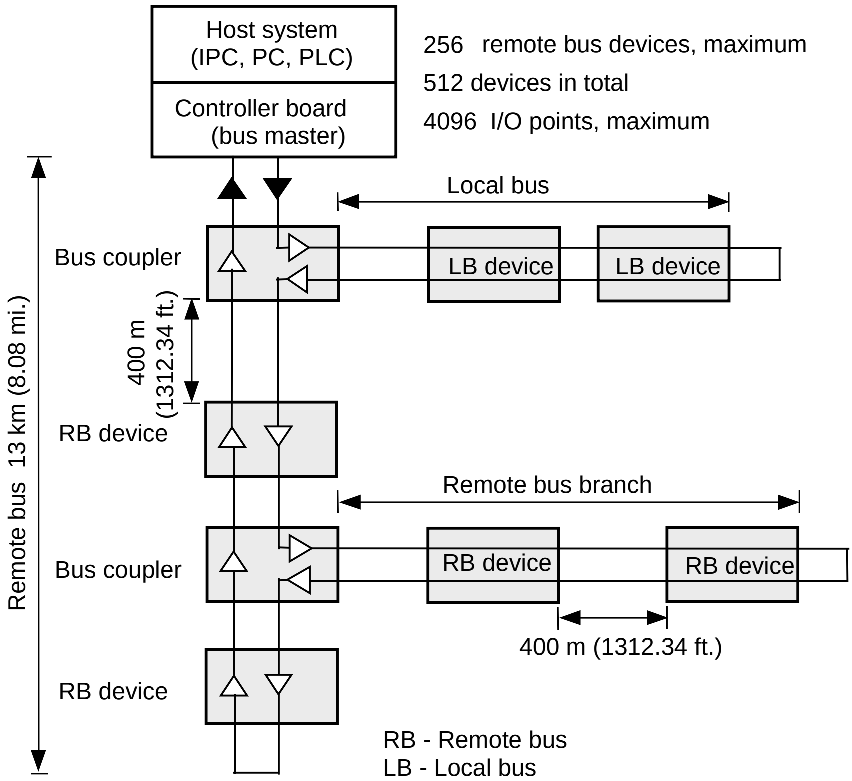


Figure 5: Basic structure of an INTERBUS system

- The *remote bus branch connects remote bus devices* and connects distributed devices over large distances. Remote bus branches can be used to set up complex network topologies, which are ideal for complex technical processes distributed over large distances.

The INTERBUS remote bus cable forms an RS-485 connection and, because of the ring structure and the additional need for an equalizing conductor between two remote bus devices, requires five cables.

Due to the different physical transmission methods, the local bus is available with 9 cables and TTL levels for short distances (up to 1.5 m [4.92 ft.]) and as a two-wire cable with a TTY-based current interface for medium distances (up to 10 m [32.81 ft.]).

Due to the integrated amplifier function in each remote bus device, the total expansion of the INTERBUS system can reach 13 km (8.08 mi.). So that the system is easy to operate, the number of INTERBUS devices is limited to a maximum of 512.

INTERBUS works as a *shift register, which is distributed* across all bus devices and uses the I/O-based summation frame method for data transmission. Each bus device has data memories, which are combined via the ring connection of the bus system to form a large shift register. Figure 6 illustrates the data transmission principle.

A data packet in the summation frame is made available in the send shift register by the master. The data packet contains all data that is to be transmitted to the bus devices (OUT data). The corresponding data registers in the bus devices contain the data to be transmitted to the master (IN data) (Figure 6a).

The OUT data is now transferred from the master to the device and the IN data is transferred from the devices to the master in one data cycle. The master starts by sending the loop-back word through the ring. At the end of the data cycle, the master receives the loop-back word. The loop-back word "pulls" the OUT data along behind it while "pushing" the IN data along in front of it. This is called full duplex data transmission (Figure 6b).

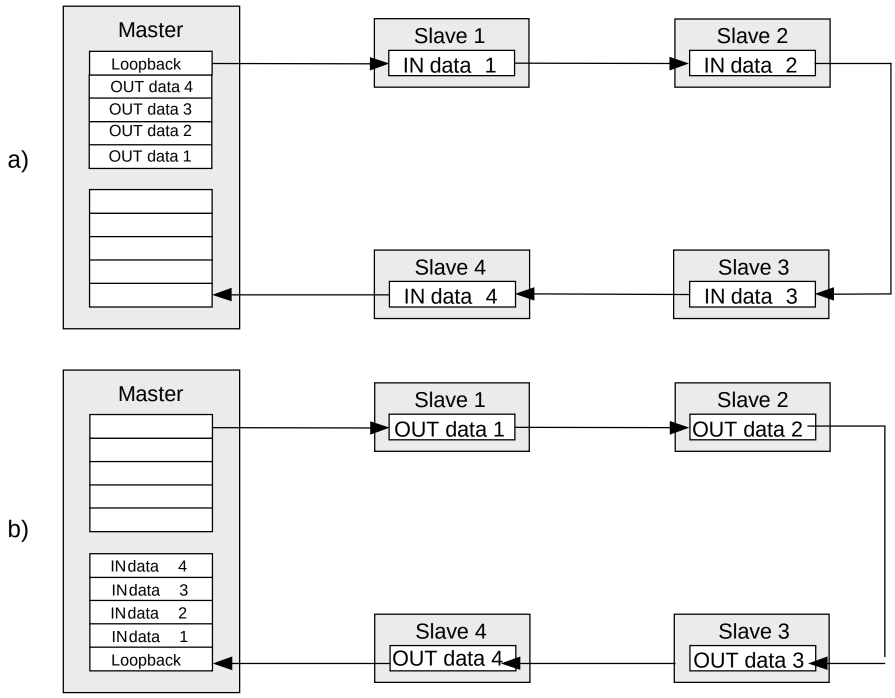


Figure 6: Principle of data transmission on INTERBUS a) Distribution of data before a data cycle b) Distribution of data after a data cycle

The devices do not have to be addressed explicitly, as the physical position of a device in the ring is known and the master can position the information to be transmitted at this point in the summation frame telegram. In the example, the 1st data word after the loop-back word is addressed to slave 4, for example.

The amount of user data to be pushed through the ring corresponds to the total data length of all bus devices. The bus couplers are integrated into the ring but do not provide any user data. Data widths between 1 bit and 64 bytes per data direction are permitted in one INTERBUS device.

3 System Configuration

3.1 Bus Elements

An INTERBUS system comprises the bus elements illustrated in Figure 7: the bus master, bus coupler, bus device, and the remote bus, as well as the local bus connections.

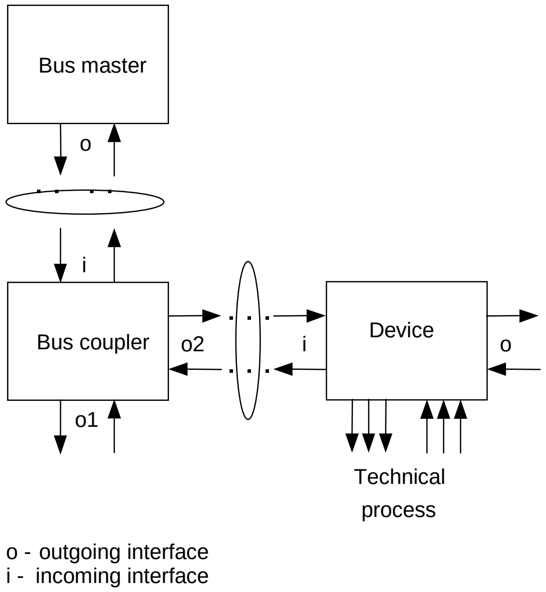


Figure 7: Elements of the topology of an INTERBUS system

The *bus master* is the only communication master in the INTERBUS ring bus system and thus controls all sequences on the ring. It has an outgoing interface (o) on which all other INTERBUS devices acting as communication slaves are connected. The bus master is available as a *controller board* for various host systems (industrial computers, PLCs VME bus systems, transputer systems, etc.) and performs the following tasks:

- Data transfer between the host system and bus device
- Bus management (configuration, error detection, reconfiguration)
- Communication between bus devices

Depending on the type of controller board, it can also take over the processing of the complete control program.

The INTERBUS master tasks require appropriate computing power. The controller boards are therefore fitted with a powerful microprocessor (usually a Motorola 68332), which is responsible solely for the INTERBUS master functions. Essentially, the firmware of this master processor shares the functionality and user-friendliness of the INTERBUS system.

The *bus coupler* is connected to an incoming remote bus cable (i) as shown in Figure 8, thus providing access to the outgoing interfaces (o1, o2). The bus coupler, which is also referred to as the *bus terminal module* in practice, divides the INTERBUS ring system into bus segments and itself operates as a communication slave. Each bus coupler has at least one incoming and one outgoing remote bus interface. Additional interfaces for connecting remote bus or local bus cables are also available. Figure 8 shows the two types of bus coupler that are available.

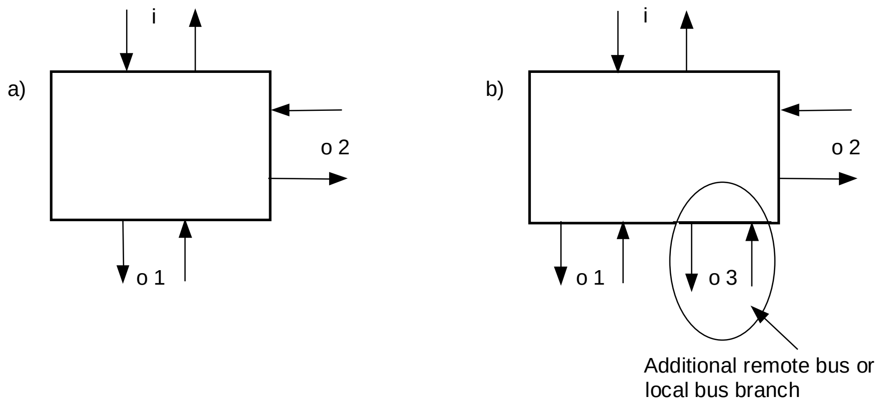


Figure 8: Bus coupler types a) Standard b) Bus coupler with additional interface

The standard bus coupler shown in Figure 8 connects a local bus or remote bus segment to the remote bus. Additional bus couplers cannot be used in the connected bus segment.

The bus coupler with the additional fourth interface shown in Figure 8b can be used to configure complex networks with optimum adaptation to the system configuration.

To support error diagnostics and ring system configuration, bus couplers are able to activate and deactivate the outgoing interfaces under the control of the bus master (Figure 9).

The *bus devices* are communication slaves in the ring bus, which have an incoming and an outgoing interface and also establish the connection to the process signals in the required format (binary, analog, digital). *Remote bus* and *local bus devices* are available depending on whether the device is to operate in a remote bus or local bus segment.

The ring system is configured via the bus couplers and the bus devices. If an outgoing interface on a bus coupler or device is not connected, i.e., a bus cable has not been connected to it, this bus element will automatically short circuit the interface internally (Figure 9).

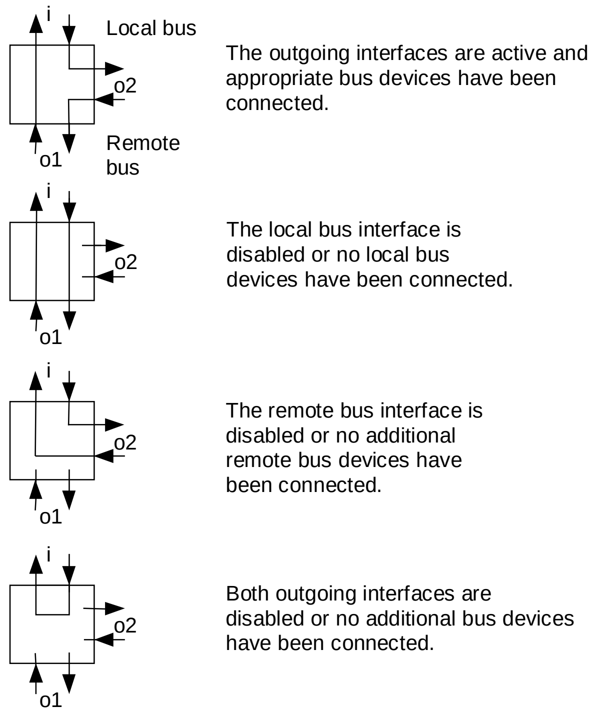


Figure 9: Switching the signal paths using a bus coupler

3.2 Bus Segments

A *bus segment* is a series of bus devices and bus connectors, which starts at the bus master or a bus coupler and ends at the next bus coupler or at whichever bus device does not have an outgoing interface.

INTERBUS distinguishes between the following basic types of bus segment based on the physical transmission method of each bus connector:

- Remote bus segment
- Local bus segment
- INTERBUS Loop segment

Remote Bus Segment

A *remote bus segment* usually starts at an appropriate bus coupler (bus terminal module) and comprises remote bus connectors and a maximum of 255 remote bus devices (Figure 10). Remote bus devices are also referred to as *stations*.

The remote bus segment is a special type, which starts directly at the bus master and comprises a maximum of 256 remote bus devices. The features of a remote bus segment are those given in the basic specifications for the *remote bus* in Table 1.

The maximum number of remote bus devices is not limited by the protocol as the devices do not need to be addressed explicitly. To maintain clarity, the latest master firmware limits the possible number of connected remote bus devices to 256.

Table 1: Basic remote bus specifications

Maximum number of devices	256
Maximum distance between devices	400 m (1312.34 ft.) for copper cables (80 - 3600) m [262.47 - 11811.02 ft.] for optical fibers
Maximum overall system expansion	12.8 km (7.95 mi.) for copper cables > 80 km (49.71 mi.) for optical fibers
Device power supply	Local
Bus connection	9-pos. D-SUB

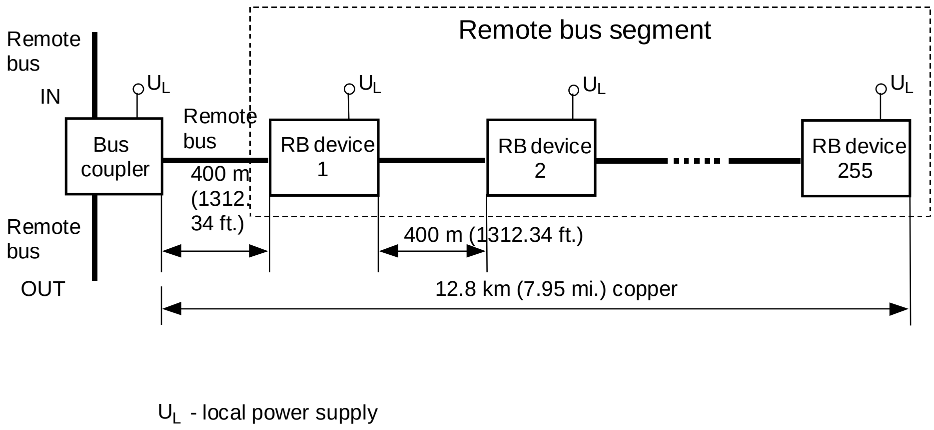


Figure 10: Remote bus segment (RB - remote bus)

In theory, the remote bus can support copper based transmission over a distance of 102.4 km (63.63 mi.). However, the system is currently limited to a maximum expansion of 12.8 km (7.95 mi.), in order to ensure that the testing time for the maximum configuration remains within reasonable limits. The expansion can be increased using optical fibers.

Each remote bus device has a local power supply as well as being electrically isolated from the outgoing bus segment. In line-based transmission, the remote bus signals are transmitted according to the electrical properties of the interface - DIN 66 259 Part 4 (corresponds to RS-485). As well as remote bus terminals, remote bus devices can be simple or intelligent I/O devices.

Local Bus Segment

A *local bus segment* starts at a bus coupler with an outgoing local bus interface and comprises local bus connections and local bus devices. In the local bus segment, a *station* comprises the complete local bus segment together with the corresponding bus coupler.

The *local bus* (formally known as the I/O bus) is designed for transmission over short distances. A typical field of application is in local control cabinets or control boxes. On the local bus, all signal cables for the sensors and actuators are combined and connected to the local bus devices together. Because of the short distance between two devices, local bus devices connected using copper-based cables do not have an RS-485 interface and data transmission takes place with TTL levels. This restricts the spatial expansion of a local bus segment. The maximum distance between two local bus devices is 1.5 m (4.92 ft.) and the total expansion of the local bus segment must not exceed 10 m (32.81 ft.).

In addition to the data lines, the local bus connections also house the power supply lines for the electronics module of the local bus devices. The power supply for all bus devices in a local bus segment is provided centrally by the bus coupler power supply unit. This means that the local bus devices only need one power supply for the connected sensors/actuators.

Local bus segments are available in various installation versions. These include in particular:

- Local bus segments with ST local bus
- INLINE stations
- Optical fiber local bus segment

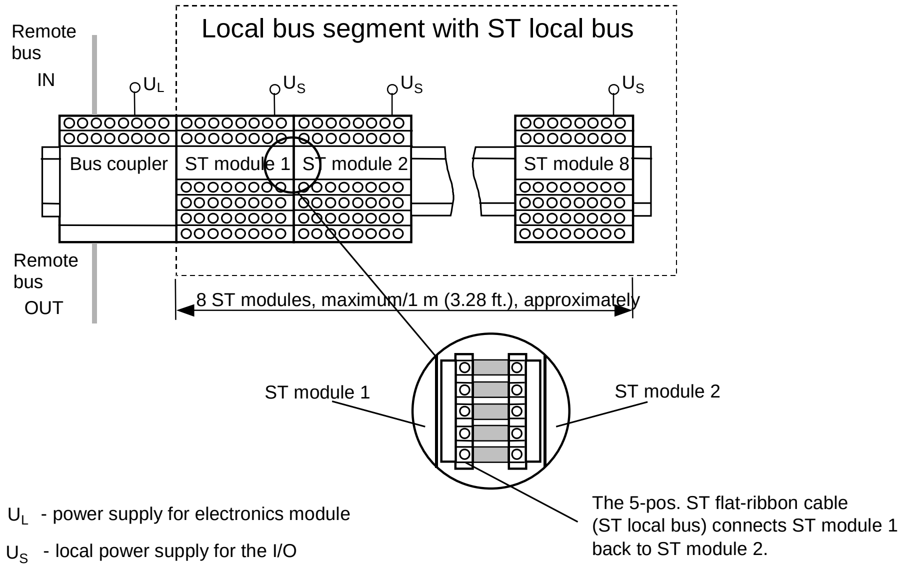


Figure 11: ST station

The standard installation version for INTERBUS devices are local bus segments with ST local bus, which are combined with an appropriate bus terminal module to form an *ST station*, enabling the formation of compact and flexible I/O stations. The individual local bus devices are designed as terminal block devices and are snapped onto DIN rails. The electronics module is a plug-in module, while the sensors/actuators are connected to a purely passive ground terminal block. The ST (Smart Terminal) local bus is a short flat-ribbon cable, which connects local bus devices via rear connectors. The structure of an ST station is shown in Figure 11.

The distance between the ST bus devices and the expansion of an ST station is limited by the length of the ST local bus cable. Therefore, the maximum length of an ST station is approximately 1 m (3.28 ft.).

The central power supply to the local bus devices and the lack of data signal regeneration limits the number of bus devices in the local bus. The physical limit is determined by the maximum current that can be supplied by the bus coupler power supply unit. The master firmware also imposes a logical limit. Currently, a maximum of 8 local bus devices are supported for SL and ST local buses.

The power supply for the I/O (sensors and actuators) is provided locally via external connections.

Inline stations, which, like ST stations, are based on terminal blocks, are another type of local bus segment. However, because they are very precisely graded, they enable the incredibly flexible configuration of compact and application-specific INTERBUS stations. The individual local bus devices are referred to as *automation terminals* and the smallest version available is a digital I/O module with 2 channels. Figure 12 illustrates an example of an INLINE station with 7 digital I/O automation terminals, of which 3 are two-channel input terminals (DI2) and 4 are two-channel output terminals DO2).

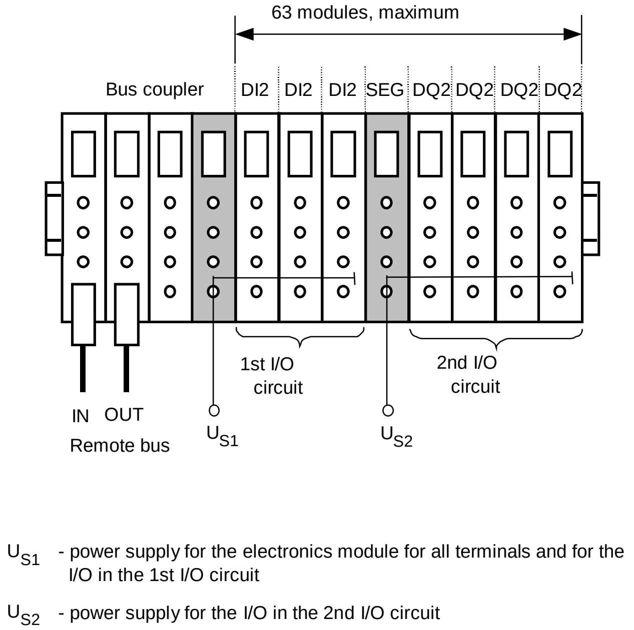


Figure 12: Inline station with 7 two-channel input and output terminals

Connecting the automation terminals by snapping them onto a DIN rail side by side automatically creates a 10-pos. power rail system which can manage the cross wiring required for the bus line and the power supply. The power rail system is referred to as an *Inline potential jumper* and functions as a local bus (Figure 13).

The communications power for all automation terminals is provided centrally through the bus terminal. Electrically isolated I/O circuits can be created using special Inline power terminals (SEG in Figure 12). The power terminals interrupt the potential jumper, making it possible to add a new circuit. The supplied voltage can be between 24 and 230 V. Additional circuit groups (segment circuits) can be created within an I/O circuit via Inline segment terminals.

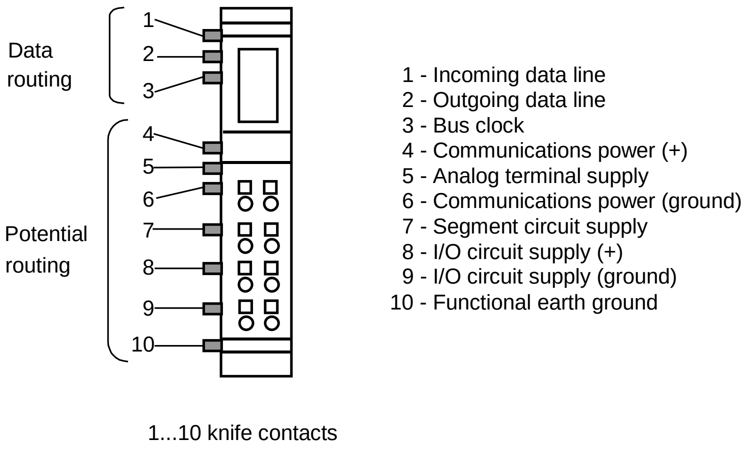


Figure 13: Potential and data routing with the Inline potential jumper

If special Inline local bus terminals are used, the following bus segments can be added to an Inline station:

- INTERBUS Loop segment
- Optical fiber local bus segments
- Remote bus segment

Up to 63 automation terminals can be connected to an Inline bus terminal. This number includes all the devices after the bus terminal, i.e., the Inline terminals and the modules for the connected INTERBUS Loop and optical fiber local bus segments.

Inline automation terminals are available in a universal version for all power ranges. An important feature is the availability of modular pneumatic terminals, which can be used to connect the Inline station directly to pneumatic actuators.

Table 2 contains a summarized overview of the basic specifications for the individual versions of a local bus segment.

Unlike in remote bus segments, it is not possible to program the deactivation of individual local bus devices in local bus segments. If a bus device in the local bus segment is faulty, only the entire bus segment can be disabled via the bus coupler.

Table 2: Basic specifications for local bus segments

	Local Bus Segment With ST Local Bus	**Inline Station**	**Optical Fiber Local Bus Segment**
Maximum number of devices	8	63	63
Maximum distance between devices	Mounted side by side	Mounted side by side	5 m (16.40 ft.)
Maximum overall system expansion	1 m (3.28 ft.), approximately	4 m (13.12 ft.), approximately	25 m (82.02 ft.)
Device power supply	Central via bus coupler	Central via bus coupler	Local
Bus connection	ST flat-ribbon cable	Inline potential jumper	2-wire polymer optical fiber

INTERBUS Loop Segment

Unlike local bus segments, whose components only really differ in terms of installation technology, *INTERBUS Loop* (sensor loop, IP 65 local bus) offers a new physical transmission method. The individual devices are connected via a simple 2-wire unshielded cable to form a ring. The data and the 24 V power supply for up to 32 sensors are also supplied via the cable. Figure 14 shows the configuration of an INTERBUS Loop segment.

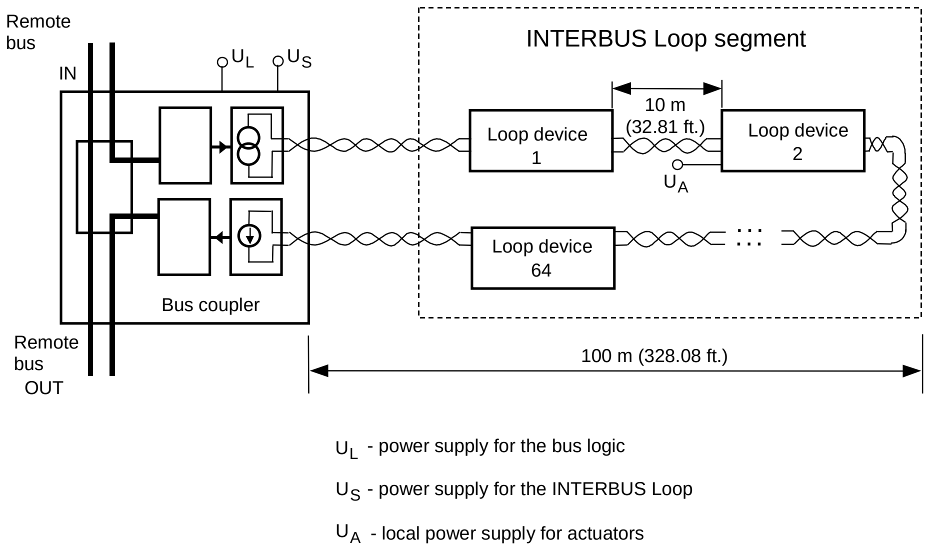


Figure 14: INTERBUS Loop segment

Data is transmitted as load-independent current signals, which have a higher level of immunity to interference than the voltage signals normally used. The data to be transmitted is modulated using Manchester code on the 24 V supply voltage (INTERBUS usually uses the NRZ code). The physical bus characteristics are converted by an appropriate bus terminal module, which can be connected to the INTERBUS ring at any point in a remote bus segment.

INTERBUS Loop has the following characteristic features:

Expansion:	100 m (328.08 ft.), maximum
Distance between two devices:	10 m (32.81 ft.), maximum
Number of devices:	32, maximum
Current consumption of devices:	1.5 A, maximum
Connection medium:	Unshielded two-wire cable, 2 x 1.5 mm² (16 AWG)

One of the main fields of application of INTERBUS Loop is the connection of individual devices with IP 65 and IP 54 connection directly in the system. An extensive range of functions and devices are available as bus devices.

The INTERBUS protocol is not converted in any way in INTERBUS Loop, which means that complex gateways are not required and an INTERBUS Loop segment can be used in conjunction with any other type of INTERBUS device. Data scanning is absolutely synchronous in all parts of the INTERBUS system. Despite this, the high scanning speed is maintained.

3.3 Network Configuration

An INTERBUS system is configured by connecting the bus devices one after the other in a ring. Bus couplers segment the ring according to the application requirements.

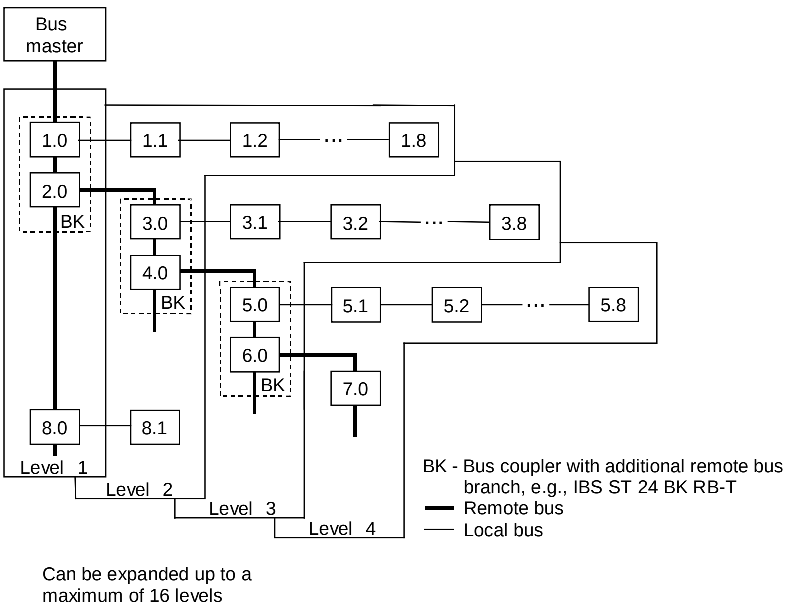


Figure 15: INTERBUS network configuration with four levels

With INTERBUS G4 (Generation 4) and later, it is possible to set up complex network topologies, which can be optimized for the structure of the automation system, by integrating bus couplers with an additional bus connection. There are two ways of structuring the configuration of this type of INTERBUS network:

1. Divide the entire network into various levels
2. Assign segment-specific device numbers

Both configuration methods will be explained using the example of an INTERBUS network configuration with four levels, as illustrated in Figure 15.

The *network is split into four different levels* starting with the bus master on the main remote bus line as the first level. The branching secondary lines are now assigned a second level. The devices connected to these lines can form additional substructures, etc. In this way, a nesting depth of up to 16 levels can be achieved. The sequence is such that a local bus (formally known as the I/O bus) in a remote bus segment is always assigned to the next level.

Segment-specific device numbers are assigned either automatically according to the physical configuration or they can be freely specified by the user. The numbering comprises two components:

<Device number> = <Bus segment number>**. <**Position number in bus segment>

According to this pattern, the second digit of the device number for all remote devices is zero, e.g., 1.0. The second digit is only used by the local bus devices (e.g., I/O modules) connected downstream of the remote device, e.g., 1.1.

Bus couplers with an additional remote bus branch appear as two separate remote bus devices with one local bus/remote bus branch, e.g., bus coupler 1.0/2.0. When physically assigning this type of remote bus device, the remote bus branch is assigned the next consecutive number, e.g., 3.0. Any additional sub-branches on this branch are assigned the next consecutive number, e.g., 4.0, 5.0, etc. The outgoing remote bus from the branch is counted as the last component, e.g., 8.0.

Device numbering is a structuring tool and should not be confused with device addressing. Although the device numbers can be used for addressing purposes, this is not absolutely necessary.

4 Data Transmission

4.1 Protocol Structure

The INTERBUS protocol is based on the OSI reference model and for reasons of efficiency only takes into account layers 1, 2, and 7 (Figure 16). Certain functions from layers 3 to 6 have been included in application layer 7.

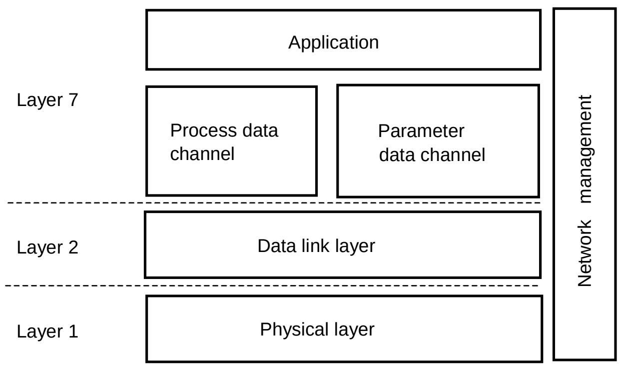


Figure 16: INTERBUS protocol structure

The *physical layer* (layer 1) defines both the time conditions (such as the baud rate, permissible jitter, etc.) and the formats for encoding information. The *data link layer* (layer 2) ensures data integrity and manages cyclic data transfer via the bus using the summation frame protocol. The transmission methods and protocols on layers 1 and 2 can be found in DIN 19 258.

Following on from the data link layer, data access to the INTERBUS devices takes place in the application layer as required via two different data channels:

a) The *process data channel* serves the primary use of INTERBUS as a sensor/actuator bus. The cyclic exchange of I/O data between the higher-level control system and the connected sensors/actuators takes place via this channel.

b) The *parameter channel* supplements cyclic data exchange with individual I/O points in connection-oriented message exchange. This type of communication requires additional data packing, as large volumes of information are being exchanged between the individual communication partners. Data is transmitted using communication services based on the client/server model.

INTERBUS devices almost always have one process data channel. A parameter channel can also be fitted as an optional extra.

During operation, an INTERBUS system requires settings to be made and provides a wide range of diagnostic information. This information is processed by the *network management* on each layer. More detailed information about readiness for operation, error states, and statistical data can also be accessed and evaluated, and network configuration settings can be made.

The *hybrid protocol structure* of INTERBUS for the two different data classes (process data and parameter data) and its independent data transmission via two channels is a decisive factor in the performance of the INTERBUS protocol. The protocol enables the creation of a seamless network comprising control systems and intelligent field devices right down to individual sensors and actuators.

The following sections describe in more detail how the individual protocol components work.

4.2 Bit Transmission

In layer 1, bits are transmitted at a standard data transmission rate of 500 kbps according to the NRZ (non-return to zero) method. The data line is scanned in the INTERBUS devices 16 times faster, in order to maximize the permissible differences in runtime between the rising and falling edges of a bit within a telegram.

If a two-wire INTERBUS cable is used as standard, a clock signal is not transmitted. A 16 MHz clock generator, which provides the internal 500 kHz clock pulse, operates in each device. The device clocks are synchronized internally by a common synchronization marker in the active INTERBUS telegrams.

Data is transmitted on INTERBUS in the form of encoded data bytes. The complete summation frame protocol is split into 8-bit portions and transmitted between two INTERBUS devices in telegrams, the format of which is similar to UART.

The two telegram formats shown in Figure 17 are used for line encoding:

- *Status telegram:* This telegram is 5 bits in length. It is used to generate activity on the bus medium during pauses in transmission and to transmit the status of the SL (select) signal.
- *Data telegram:* Data telegrams transmit user data between two devices. A data telegram comprises 5 bits of header information and 8 bits of user data. To supplement the information in the status telegram, the data telegram header contains an additional bit, which indicates the status of the CR (control) signal.

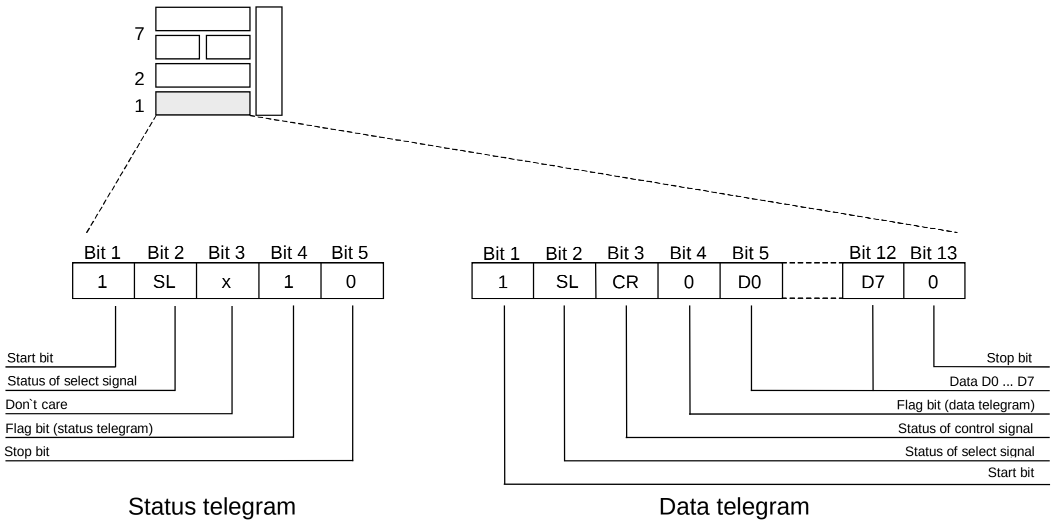


Figure 17: Line encoding in the Physical Layer

Active telegrams are processed in the INTERBUS devices using protocol logic and various shift registers.

4.3 Summation Frame Protocol

Essentially, the cyclic INTERBUS protocol is implemented in the data link layer (layer 2). The protocol is familiar with various operating phases, which are defined by encoding the SL and CR signals in the status and data telegrams (Figure 18).

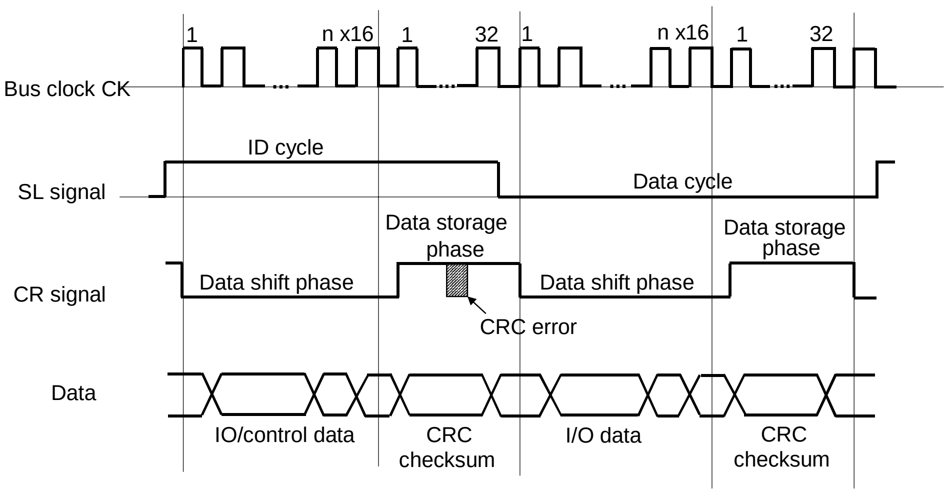


Figure 18: Operating phases in the INTERBUS protocol

A basic distinction is made between the identification and data cycles, and, within these cycles, between the data shift and data storage phases. During the *identification cycle* (ID cycle), the INTERBUS devices switch the ID registers in the INTERBUS data ring via the protocol control and the bus master can identify all the devices. In the *data cycle*, the master sets the SL signal to 0 and the INTERBUS ring is then closed via the data shift register. In terms of the protocol sequence, there is no difference between the data cycle and the ID cycle. The two cycle types only differ in the status of the SL signal and the number and type of data/registers from which the data is being transmitted. The user data is transmitted in the relevant cycle in the *data shift phase* (CR signal = 0).

Once the user data has been transmitted, the CR signal switches the system to the *data storage phase* or *FCS (Frame Check Sequence) phase*. In this operating phase, data is saved using a checksum method with a CRC polynomial according to CCITT.

The bus master uses the information from the ID cycle to ascertain the total amount of data available in the INTERBUS system. The first thing the master does at the beginning of every cycle is to send a *loop-back word* to check that this amount of data, and thus the number of shift registers, is actually still available in the ring. The loop-back word is a special 16-bit data item used in the ID cycle to detect the end of a data shift phase. The telegram shown in Figure 19 is sent in each cycle as the INTERBUS summation frame.

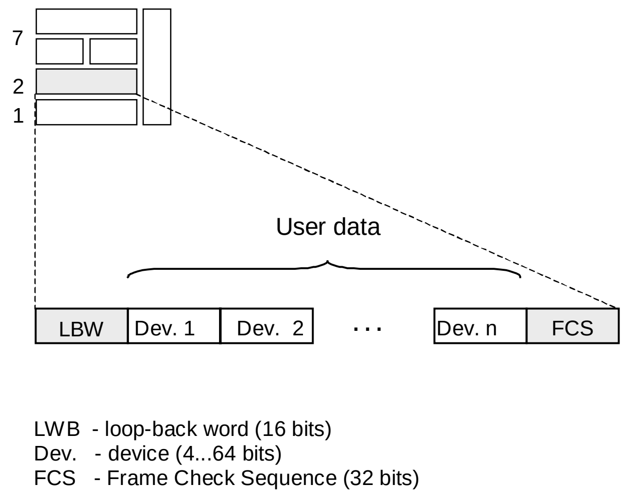


Figure 19: Summation frame telegram in the data link layer

The *summation frame method* used on INTERBUS is a collision-free TDMA transmission method (TDMA - Time Division Multiple Access). The reading in/out of the shift registers in the devices assigns almost every device in the summation frame with a time slot appropriate for its function. The transmission time, which is the total time slot time, can be calculated easily and is a guaranteed value on each INTERBUS system. The summation frame method also ensures that the process image for all devices is consistent, because all the input data originates from the same sampling point and all the output data is accepted by the devices simultaneously.

Data exchange on INTERBUS with the summation frame protocol takes place in the following sequence, as illustrated in Figure 20:

1. All devices are reset and the ID code (including the most recent errors detected) is loaded to the ID shift register.
2. The ID cycle is processed by the master (i.e., all devices are supplied with control information and their current ID codes are read).

3. The ID code is evaluated by the master and data cycles are processed in order to transmit I/O data from/to the devices.
4. If the master or a device detects an error in the data cycles, the master initiates an ID cycle in order to locate the error source.
5. Additional ID cycles to reprogram the network configuration (e.g., disable faulty bus segments) may be initiated to deal with errors that have been detected.

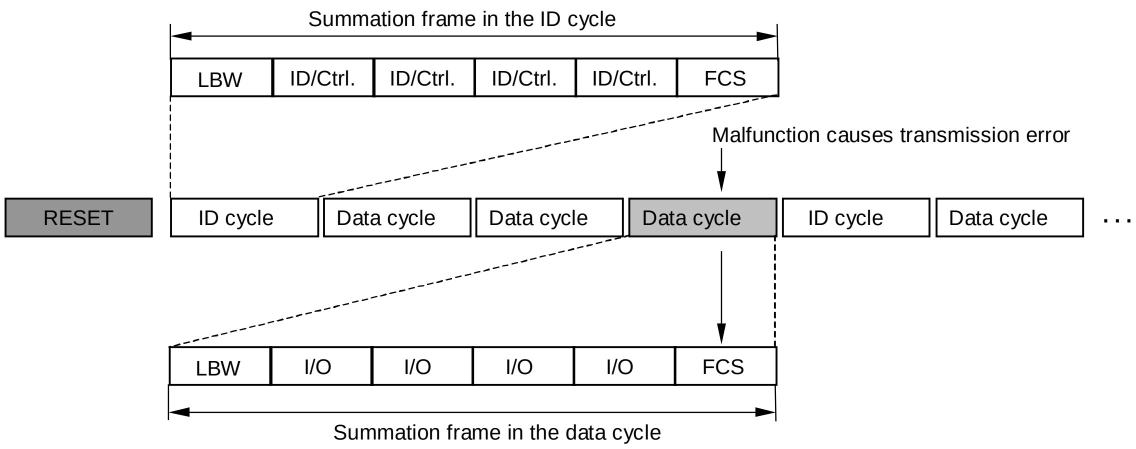


Figure 20: Data exchange with the summation frame protocol

4.4 Process Data Channel

The INTERBUS protocol groups all components of connectionless I/O-based data transmission processes in the *process data channel* (PDC). This channel enables direct and quick access to cyclically transmitted process data.

Process data is characterized by its immediate effect on technical processes. It includes for example switch states, control signals for contactors and valves or setpoint and actual values for sensors.

The complexity of the process data per terminal device (sensor/actuator) is very low and is usually in the range of a few bits. The trend towards distribution requires that the process data is detected at source wherever possible and fed into the INTERBUS system via a bus device. This results in a large number of devices (even up to a few hundred). Although the length of information per device was until recently typically 8 - 16 bits, it is now possible to network individual sensors cost-effectively with 1 or 2 bits. INTERBUS takes this into account by precisely grading the I/O for the Inline device range.

Process data is almost always uniquely identified by its address and/or the sensors/actuators it represents. Additional measures to identify this transmission data are not required. Therefore, the process data is presented to the user program via the process data channel in the form of an I/O process image that is constantly updated. In the INTERBUS protocol, the process data channel is the total amount of I/O data in the summation frame telegram during the data cycles (Figure 21).

From a programmer's point of view, there is in principle no difference between access to process data transmitted on INTERBUS and access to data on parallel I/O boards. After an initialization phase, in which the process data to be transmitted is registered in the application, the data can be updated using a simple read/write procedure.

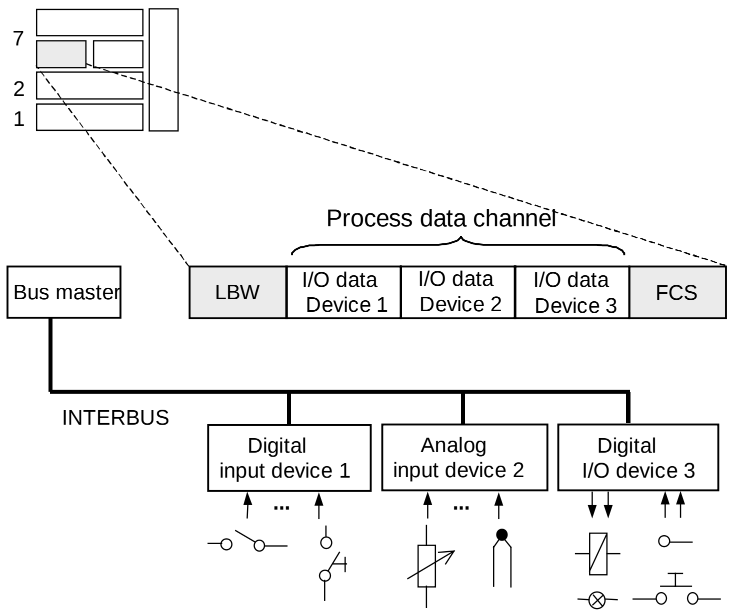


Figure 21: Process data channel in the summation frame protocol

4.5 Parameter Channel

The *parameter channel* is used for manufacturer-independent and standardized transmission of complex parameter data on INTERBUS. In an automated production process, *parameters* are used to set, program, monitor, and control intelligent devices in the process I/O. Unlike process data, parameters are acyclic. Information is only transmitted when required and is thus unique. The complexity of a parameter block in the sensor/actuator range can be between 10 and 100 bytes for the parameterization of devices up to a few hundred kbytes of program information.

Due to the properties of parameter data outlined above, the parameter channel transmits data on a connection or message basis between exactly two INTERBUS devices. The structure and operation of the parameter channel are based on the MAP/MMS automation protocol.

The parameter channel can also be used to transmit TCP/IP application protocols such as FTP (file transfer) on INTERBUS.

Typical INTERBUS devices with parameter channels are frequency inverters, servo amplifiers, positioning controllers, and operating and display units.

Parameter Channel in the Summation Frame Protocol

The basis for parameter data transmission is the standardized INTERBUS summation protocol according to DIN 19 258. "Gaps" are left at certain points in the summation frame so that devices wishing to exchange parameter data can be addressed. The INTERBUS protocol defines a width of a few bytes for the gaps, meaning that they are almost always smaller than the parameter blocks to be transmitted. It is for this reason that each parameter block is divided into individual packets and transmitted cycle for cycle in segments small enough to fit through the gaps, before being reconstructed once it has reached its destination. The load placed by parameter data on the INTERBUS cycle is thus the same as that placed on it by a process data device of the same size. The parameter channel generates the gaps required for parameter transmission, including the associated transmission procedures, in the now *expanded* summation frame protocol. Figure 22 illustrates this concept.

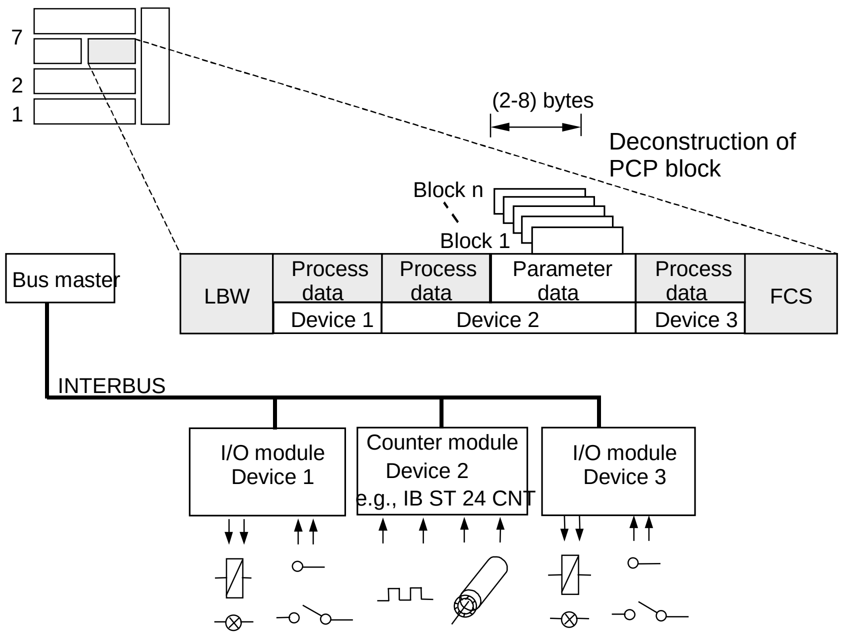


Figure 22: Parameter channel in the expanded summation frame protocol

Parameter transmission always takes place when requested. The required protocol points (gaps) in the expanded summation frame protocol of the corresponding INTERBUS devices are reserved. Parameter data is only inserted at the appropriate protocol points if a device has parameter data to exchange. Otherwise these points remain unused.

The hybrid access process with the expanded summation frame protocol on INTERBUS offers the decisive advantage that process and parameter data can be transmitted in parallel without one affecting the other in any way. Unlike other fieldbus solutions, even when larger parameter records are being transmitted, the determinism and time equidistance of the process data are maintained. The ability of the INTERBUS system to communicate process data almost in realtime is not adversely affected.

Method of Operation of the Parameter Channel

Communication on the parameter channel is based on the principle of the *client/server model*.

- A device in a communication relationship initiates communication by sending a *request*. This device is the client.
- A second communication device receives this request as an *indication* and responds to it. This device is the server. The request may require a reaction from the server. The server sends this as a *response*.
- The client receives the response from the server as a *confirmation*. The communication process is now complete.

INTERBUS' client/server model enables parameter data to be exchanged between the bus master and INTERBUS devices or between two INTERBUS devices.

4.6 Application Interfaces

The interfaces illustrated in Figure 23 are available to the control application for the process data channel and parameter channel as APIs (Application Program Interfaces). In the standard INTERBUS interface concept, the interfaces are based on a multi-port memory (MPM) as hardware interfaces between the host controller and the INTERBUS controller board.

The application exchanges cyclic process data via the *data interface* (DTI). The control system has a defined memory area in the MPM, which corresponds to the I/O image. The DTI also provides access to special registers (diagnostic register, control register). The assignment between the host address area and the required MPM address area takes place during the configuration phase. The DTI is then able to provide random access to preset and configured data areas without an access protocol.

The application sends signals via the *signal interface* (SGI) to activate preset service request sequences, e.g., to start the system, disable bus segments, etc. This enables a host PLC to exchange data and commands with the INTERBUS devices simply by means of bit assignment. The SGI requires an access protocol, which comprises the specified temporal sequence of three signal bits.

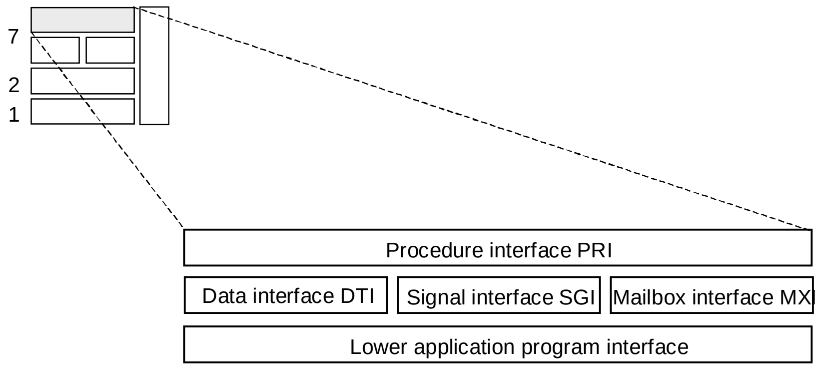


Figure 23: Application interfaces (API)

The *mailbox interface* (MXI) must be used if more complex information is to be exchanged.

An additional interface, which forwards all information to a protocol controller for channel-based transmission, may be present below the DTI, SGI, and MXI interfaces. A *procedure interface* (PRI), which enables optimum adaptation to the host operating system and its programming environment, is usually present above these interfaces.

4.7 Network Management

The complex tasks relating to configuration, error treatment, and power optimization on an INTERBUS system are managed by the PNM (Peripheral Network Management).

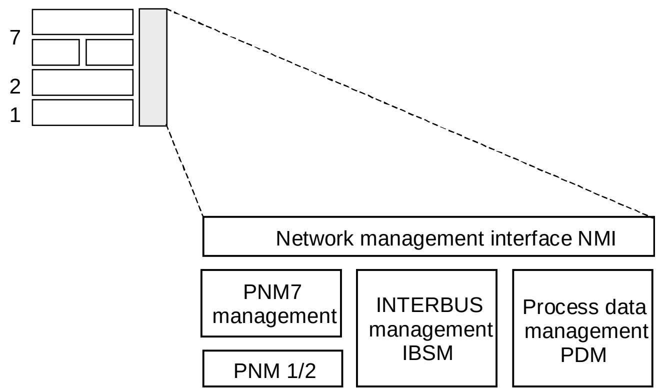


Figure 24: Network management components

The PNM comprises the following components (Figure 24):

- INTERBUS management (IBSM) for global management tasks
- Process data management (PDM) for managing the process data channel
- PNM7 for parameter channel management tasks in layer 7

The *INTERBUS management IBSM* manages all tasks relating to operation, configuration, error detection and messaging, statistics, and safety on the entire INTERBUS system. A distinction is not made between the process data and parameter channels.

Only a small number of IBSM services are generally needed to operate an INTERBUS system.

An INTERBUS system can be started up in just three steps:

Step 1:	`Clear-Display`	-	All error displays are deleted from the master board.
Step 2:	`Configure-Bus`	-	The bus is reconfigured by the master.
Step 3:	`Start-Bus-Cycle`	-	The data cycles are started.

The application interface is provided by the *network management interface* (*NMI*). This can be accessed via the API as illustrated in Figure 23.

5 Electrical Configuration

5.1 Protocol Chip

The most important element in the electrical configuration of an INTERBUS device is the INTERBUS protocol chip, which manages the complete summation frame protocol and provides the physical interface to the INTERBUS ring. The bus master and INTERBUS slave devices use different protocol chips according to their function in the INTERBUS system. Hardware solutions tailored to meet specific technical requirements are available for both INTERBUS master and slave solutions.

Protocol Chips for INTERBUS Slaves

The single most important block for manufacturers of INTERBUS devices is the *INTERBUS slave protocol chip IBS SUPI 2/3* (SUPI - Serial Universal Protocol Interface). It is estimated that 90% of all INTERBUS devices are fitted with these chips.

The SUPI 3 is an ASIC with approximately 15,000 gate equivalents in 0.7-μm format and is available in 84-pos. PLCC or 100-pos. QFP housing. Figure 25 shows a simplified block diagram of the block.

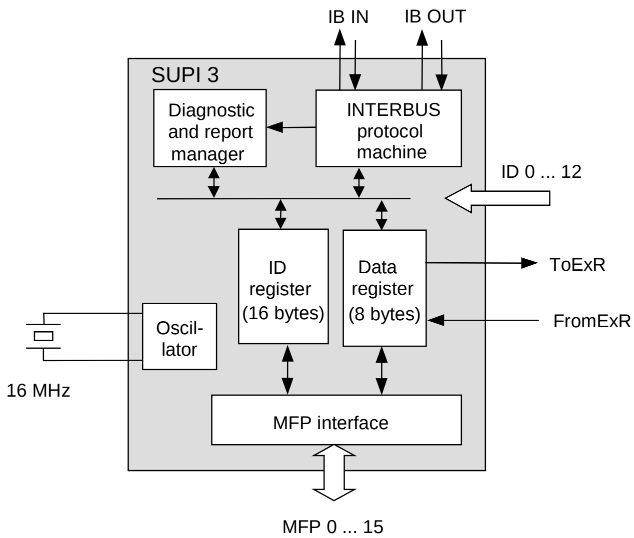


Figure 25: Block diagram of the INTERBUS SUPI 3 protocol chip

The parts of the INTERBUS protocol that correspond to layers 1 and 2 of the OSI reference model are processed entirely in the *protocol machine*. This means that basic devices (e.g., digital I/O modules) can be fitted with the SUPI as an INTERBUS interface without additional software or processing power being required. The protocol machine also provides physical access to the incoming (IB IN) and outgoing (IB OUT) INTERBUS interface.

Both shift registers - the *ID register and data register* - operate as send and receive buffers in the ID and data cycle. The application and/or higher protocol layers can access this buffer via the MPM interface (MPM - Multi-Function Pin). The MFP interface can be set according to interface requirements.

The data registers can be expanded with external registers (ToExR, FromExR). The INTERBUS register chip SRE 1, which, if required, can expand the shift register width of an INTERBUS device to 64 bytes, is used for register expansion. By default, the register width of the SUPI 3 is 8 bytes.

The *diagnostic and report manager* constantly monitors the operation of the SUPI (on-chip diagnostics). Any error descriptions that are received, such as CRC errors, transient loss of medium, voltage dips, etc., are saved to the ID send buffer and can be read from there by the master at any time. This means that unique error locations can even be identified for sporadic errors that are difficult to diagnose.

Using the SUPI as the INTERBUS slave chip enables all INTERBUS device variants for the remote and local bus to be implemented, with the exception of those for INTERBUS Loop. INTERBUS Loop also works with the INTERBUS protocol but uses a different physical transmission medium, which requires the protocol chip on the physical interface to be of the same format.

Protocol Chips for INTERBUS Masters

The standard master protocol chip for INTERBUS masters is the *IPMS* microcontroller. The IPMS is designed to work with a wide range of different processors. The master chip is often used together with the Motorola CPU 68332. Figure 26 illustrates a block diagram for this type of INTERBUS master.

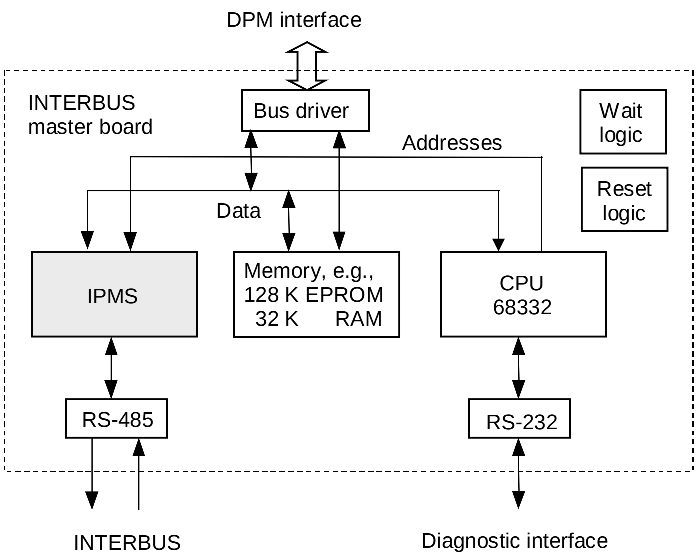


Figure 26: Block diagram of an INTERBUS master with IPMS master protocol chip

The master firmware, which manages the INTERBUS functions, is stored in the EPROM. Only actual bit transmission (layer 1, parts of layer 2) takes place via the IPMS. The IPMS is connected to the relevant host system via a shared memory area, which, in its simplest format, is a DPM (Dual Port Memory) or an MPM (Multi-Port Memory).

INTERBUS masters with IPMS are available in various formats depending on the functions required.

5.2 Local Bus Devices

Figure 27 illustrates an I/O bus interface for the 2-wire protocol with the SUPI 3. This interface is used to configure an ST local bus.

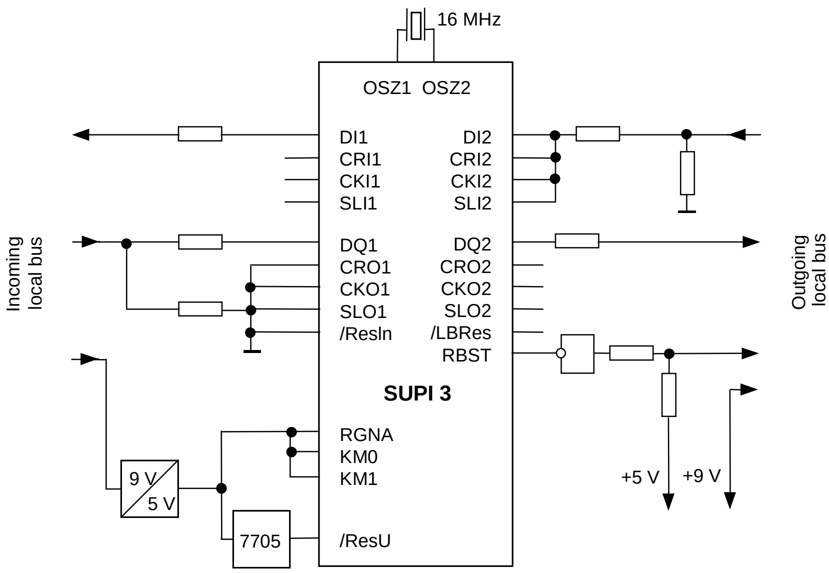


Figure 27: Local bus interface with SUPI 3 (2-wire protocol)

The INTERBUS interface configuration is defined by the wiring of pins KM0, KM1, and RGNA:

KM0	KM1	RGNA	Operating Mode
0	0	1	SL local bus 8-wire protocol
1	1	1	ST local bus 2-wire protocol
1	1	0	Remote bus 2-wire protocol

The *ST local bus* operates with four transmission signals, which, due to the ring format of INTERBUS, are available twice at the incoming and outgoing local bus interface as IN and OUT signal lines (Figure 28). In addition, one incoming and one outgoing reset line are also available in the local bus.

The bus signals can be connected directly to the local bus connectors, as the SUPI 3 meets the INTERBUS specification for the local bus even without external drivers and receivers. Unlike the SUPI 2, the SUPI 3 does not have blocking capacitors on all bus signal inputs.

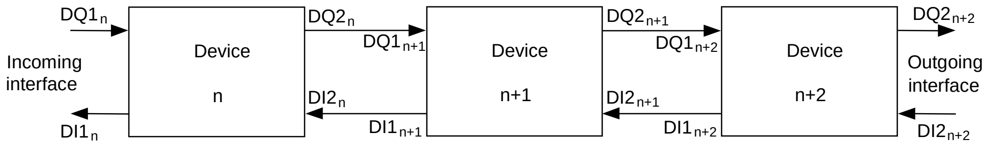


Figure 28: Signal routing in the INTERBUS local bus segment (example: data lines)

The 5 V supply for the device logic is drawn from the 9 V bus connector supply. The Type 7705 monitoring circuit monitors the 5 V supply and generates the initialization reset.

The remote bus connector closes the INTERBUS ring if the device is the last one in that local bus segment. The remote bus connector signal is jumpered in the output connector after +5 V. If the output connector is not present (remote bus connector = 0), the SUPI terminates the signal flow and diverts the outgoing interface to the return path.

The *ST local bus* is distinguished by the lack of an RS-485 driver and the transmission of the protocol with TTL levels. A five-wire flat-ribbon cable is used as the bus cable to connect the modules.

5.3 Remote Bus Devices

If it is used as a remote bus device, the drivers and receivers required for differential signal transmission to RS-485 must be added to the SUPI (Figure 29).

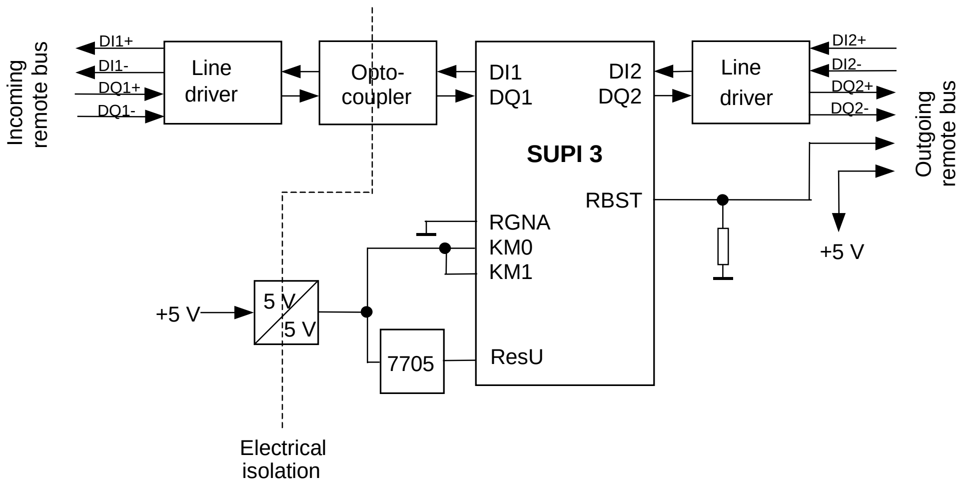


Figure 29: Remote bus interface with SUPI 3 (remote bus device)

On the remote bus, transmission takes place via two twisted pair cables (DO+/DO-, DI+/DI-). Unlike the local bus, remote bus devices require a dedicated power supply for the device logic, as this is no longer provided via the bus cable.

Two optocouplers on the incoming side and one DC/DC converter in the device power supply ensure effective electrical isolation (dotted line in Figure 28). The CR, CK, and

SL signals are generated on the 2-wire remote bus from the transmission telegram and can thus be connected to ground.

If the SUPI is configured as a bus coupler, in addition to the incoming and outgoing remote bus interfaces, an additional outgoing local bus interface is available via the MFP pins. This local bus interface is wired as illustrated in Figure 27.

5.4 INTERBUS Loop Devices

Although INTERBUS Loop devices also operate with the standardized INTERBUS protocol, they do not transmit voltage signals to RS-485, which is usually the case on INTERBUS. Instead, they use load-independent current signals and Manchester coding to transmit the data and supply voltage on one and the same bus line (loop).

Due to the different physical transmission medium, a special *protocol chip*, the *IBS LPC*, is available for INTERBUS Loop. This chip is an ASIC with approximately 7000 gate equivalents and is supplied in QFP-44 housing. Special loop diagnostics are integrated into the LPC 2 to extend the familiar diagnostic functions of the SUPI 3. To simplify the external wiring, the chip also contains a 16 MHz quartz oscillator, overtemperature protection, a 5 V voltage regulator, and a reset generator with undervoltage monitoring.

Figure 30 illustrates the structure of an INTERBUS Loop device with the LPC protocol chip (shaded in the diagram).

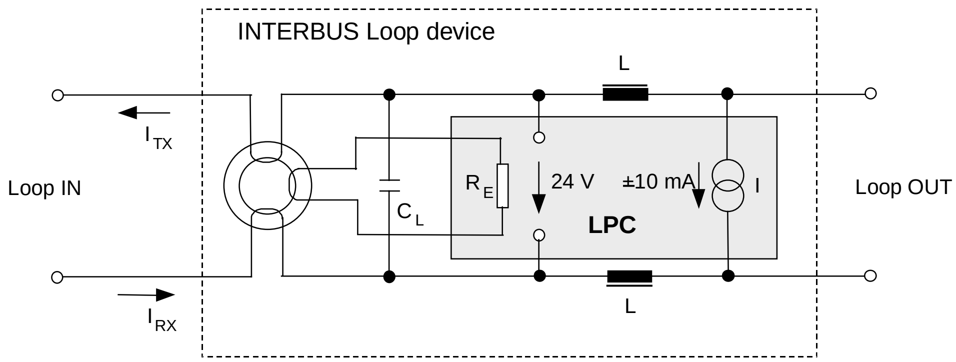


Figure 30: Structure of an INTERBUS Loop device

As on a TTY interface, the data flow on INTERBUS data flow is sent on the line as a current signal (I_{Tx}, I_{Rx}). When the current signal arrives at the next Loop device, it is measured against R_E using a toroidal-core transformer and then short-circuited with the C_L capacitor. The dedicated 24 V DC supply is then available at the capacitor as the supply voltage. The two inductors in the loop lines decouple the output signal from the input signal and forward the 24 V DC voltage to the loop output of the device. Once the data in the LPC protocol chip has run through the INTERBUS shift registers, it is modulated as a current signal (+10 mA / -10 mA) behind the inductors on the loop output line on which the supply voltage is present.
The LPC, like the SUPI, has a configurable 4-bit MFP interface for interfacing with devices, which enables the sensors and actuators to access the INTERBUS system.

A loop bus segment is interfaced with the INTERBUS remote bus using special INTERBUS Loop bus couplers, which also provide the supply voltage for the Loop.

6 System Components

The INTERBUS system components for structuring basic INTERBUS systems essentially comprise controller boards for open computer systems and controller boards for programmable logic controllers.

6.1 Controller Boards for Open Computer Systems

INTERBUS offers various controller boards with graded power ranges for standardized and manufacturer-independent PC technology and for VMEbus systems. Depending on the controller board, both computer systems can operate as INTERBUS bus masters or be integrated as slave devices into an INTERBUS system.

PC Controller Boards

The IBS ISA FC/486DX/I-T module for applications in the high-performance segment will be considered below in more detail as a representative example of a controller board operating as a bus master.

Table 3: Technical data for the IBS ISA FC/486DX/I-T

Control system	IBM-compatible PCs, ISA slot
Dimensions	(222 x 99 x 35) mm [8.740 x 3.898 x 1.378 in.], 2 slots
INTERBUS interface	2-wire remote bus, 9-pos. D-SUB female connector
Diagnostic interface	RS-232C, 9-pos. D-SUB male connector
Supply	5 V DC ±5%, typical; 2 A, typical
Memory address area	4 kbytes, 8000 ... FF000h
I/O address area	8 bytes, 100 ... 3F8h
Number of devices	512
Amount of process data	256 words (= 4096 binary I/O)
PC/104 board	
Coprocessor	80486DX/4, 100 MHz
Memory	4MB DRAM, 1 MB Flash
Operating system	VxWorks, IEC 1131 runtime system
Number of PCP devices	62

The *IBS ISA FC/486DX/I-T* controller, also referred to as the *Field Controller*, is a powerful INTERBUS master, on which control and automation programs according to IEC 1131 can run independent of the host PC. The Field Controller uses an embedded PC board according to the PC/104 standard as an integrated coprocessor board. PC WORX automation software is used with all Field Controllers to provide seamless configuration and programming under IEC 1131. Table 3 contains the general technical data for the IBS ISA FC/486DX/I-T.

On the IBS ISA FC/486DX/I-T Field Controller, the central interface to the host PC is configured as a multi-port memory (MPM) as illustrated in Figure 31, so that the coprocessor can access the process data. There are three MPM devices: the IBS master, the coprocessor board, and the host PC.

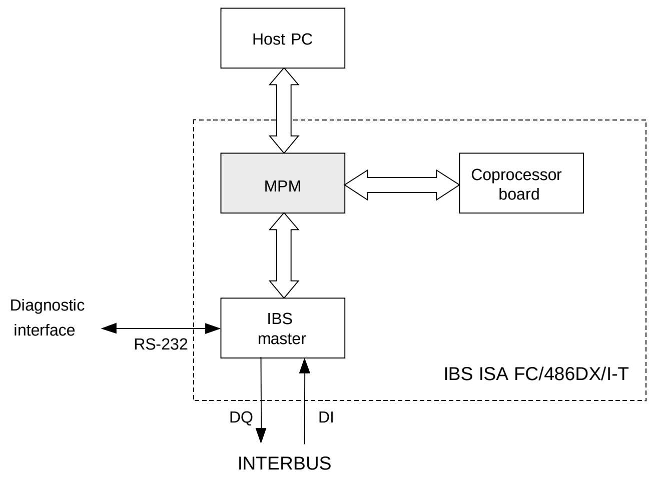


Figure 31: Basic structure of the PC master controller board IBS FC/486DX/I-T (Field Controller)

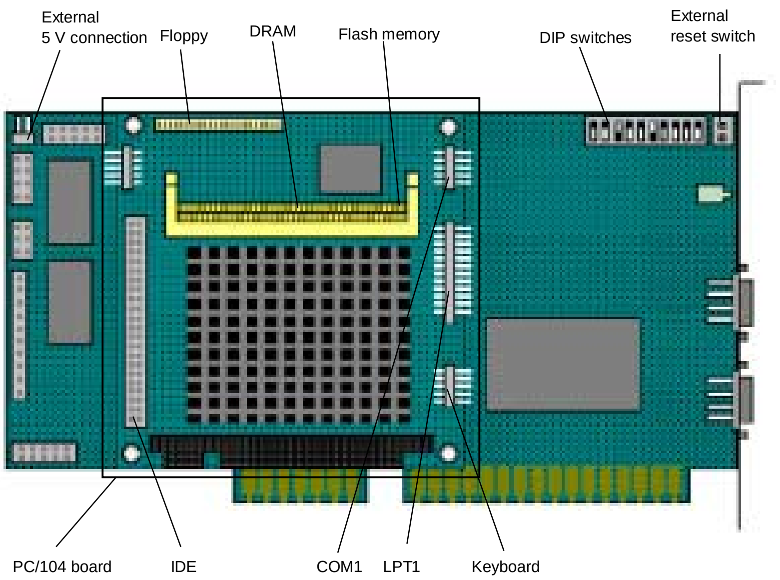


Figure 32: View from above of the Field Controller IBS ISA FC/486DX/I-T

The Field Controller also has a *coprocessor board*, which is a snap-in-place computer module in PC/104 format. It contains all the PC/AT computer functions (without the graphics). Figure 32 (view from above) shows the board and the connections for the keyboard, floppy, HD (IDE interface), DRAM and Flash memories, COM1, COM2, and LPT1.

There are 4 MB of on-board DRAM, which can be expanded to 16 MB via the Dual-SO-DIMM socket. Up to 9 MB Flash memory is available on the board as an integrated, bootable hard disk (1 MB onboard). The realtime operating system Vx Works and an IEC 61131 runtime system are pre-installed on this Flash hard disk.

The control programs are developed and tested on the host PC before being downloaded to the coprocessor board. This frees up the host PC for other tasks such as the visualization of the process signals and parameters connected via INTERBUS.

The IBS ISA FC/486DX/I-T Field Controller has an *external 5 V supply* for implementing safe and independent control solutions. If the host PC voltage supply fails, the Field Controller continues to operate and the host interface is switched to high-resistance.

The Field Controller is also available as IBS ISA FC/I-T without a coprocessor board for standard applications.

6.2 PLC Controller Boards

INTERBUS controller boards are available as bus masters and slave devices for most programmable logic controllers (PLCs) currently available on the market.

A representative example of this type of controller board is the PLC master controller board for SIMATIC S7-400 controllers, which will be considered in more detail below.

PLC Controller Board IBS S7 400 DSC/I-T

Table 4 contains the essential technical data for the board.

Table 4: Technical data for the IBS S7 400 DSC/I-T

Controller	Siemens SIMATIC S7 400
Dimensions	(210 x 290 x 50) mm [8.268 x 11.417 x 1.969 in.], 2 slots
INTERBUS interface	2-wire remote bus, 9-pos. D-SUB female connector
Diagnostic interface	RS-232C, 9-pos. D-SUB male connector
Supply	5 V DC from the host system; 1 A, typical
Address assignment in the PLC	24 bytes I/O, of which 16 bytes for INTERBUS-specific register
Number of devices	512
Amount of process data	256 words (= 4096 I/O)
Emulated S7 board	FM451 FIX SPEED, positioning module for drives

The IBS S7 400 DSC-T controller board connects INTERBUS to the SIMATIC S7-400 PLC. Once the power supply has been connected, the INTERBUS configuration is read automatically and started with the parameterization stored in the Flash parameterization memory. As an INTERBUS master, the board supports the following functions:

- Construction of INTERBUS networks with up to 16 levels
- Synchronization of PLC program and INTERBUS cycle
- Process data preprocessing
- Parameterization of alternative and changing INTERBUS segment parts
- Comprehensive diagnostics and easy operation

Figure 33 illustrates the front view of the IBS S7 400 DSC/I-T.

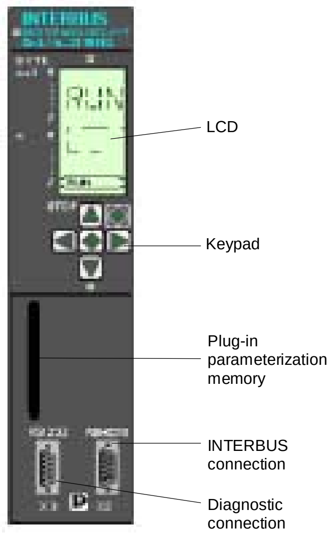


Figure 33: Front view of the IBS S7 400 DSC/I-T controller board

The controller board has a *four-line LCD* for displaying the comprehensive INTERBUS system diagnostics, on which the operating and error states are shown in plain text. The type of message and a parameter with additional information are displayed. The display also indicates the status of the I/O data. The LCD is operated easily and intuitively via the *keypad* on the front plate. As on a PC, the four arrow keys position the cursor in the menu.

The basic parameters (assignment in the S7 system) for the controller board are set with the STEP 7 parameterization tool. In this process, the controller board emulates the Siemens FM 451 I/O module. The powerful INTERBUS CMD tool can be used for INTERBUS parameterization. The plug-in *Flash memory card* can be overwritten numerous times and has been designed as a non-volatile memory.

7 Device Modules

INTERBUS devices integrated into control cabinets or terminal boxes with IP 20 protection are referred to as device modules. Essentially, three installation variants are available, which are described in Table 5.

Table 5: Installation variants with INTERBUS device modules

Installation Variant	Description	INTERBUS Interface	Product Example
Device modules for compact stations	Modules with several I/O points (up to 32 digital I/O), function modules, can be mounted side by side on DIN rails, exchangeable electronics module, potential routing	Local bus	ST family from Phoenix Contact, WINbloc IPS family from Weidmüller
Automation terminals	I/O modules with 1 … 4 digital I/O, function modules, can be mounted side by side on DIN rails, internal potential jumper	Local bus	INTERBUS Inline from Phoenix Contact, IBS modules from WAGO's 750 range
Individual modules	Various compact modules from different manufacturers, can be mounted separately or on DIN rails, up to 32 digital I/O, special functions	Remote bus	RT modules from Phoenix Contact, RIO stand-alone modules from Schleicher

7.1 Device Modules for Compact Stations

These device modules are available with various I/O functions from a variety of manufacturers. Common to all modules is the fact that they can be mounted side by side on a DIN rail. Several I/O points or channels can be connected to each module. Modules are frequently installed in blocks on DIN rails to form *compact stations*, which are then connected to the INTERBUS remote bus via separate bus terminal modules (Figure 34).

The device modules use the local bus with a 2-wire protocol based on TTL technology. Special short bus cables are used for the INTERBUS connection between the modules.

Most devices for compact stations only have one process data channel (PD channel), on which the process image is transmitted in accordance with the defined module-internal signal processing. It is also possible to make limited parameter settings for some analog I/O modules and more complex special function modules such as counters and V 24 modules via a parameter channel (PCP channel).

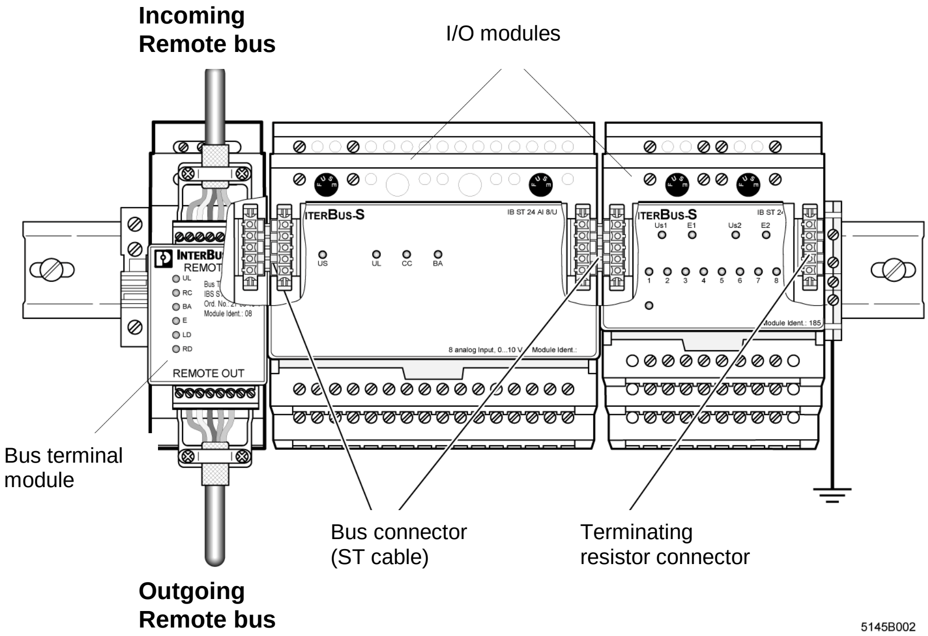


Figure 34: Installing a compact station on a DIN rail

The device modules use the local bus with a 2-wire protocol based on TTL technology. Special short bus cables are used for the INTERBUS connection between the modules.

Most devices for compact stations only have one process data channel (PD channel), on which the process image is transmitted in accordance with the defined module-internal signal processing. It is also possible to make limited parameter settings for some analog I/O modules and more complex special function modules such as counters and V 24 modules via a parameter channel (PCP channel).

7.2 Automation Terminals

Automation terminals are an installation system for INTERBUS device modules. They provide numerous automation and control functions, which are graded specifically, therefore increasing flexibility. Automation terminals are simply plugged into application-specific function units. Snapping the terminals onto a DIN rail automatically wires the INTERBUS connection, power distribution, protective circuit, isolated groups, and the fuse. This user-friendly method of installation almost eliminates the need for wiring in the control cabinet or terminal box. Figure 35 shows an example of automation terminals for the digital input and output (each with 2 channels) from the INTERBUS Inline product range.

The functions offered by the INTERBUS Inline automation terminals can be categorized in the following groups:

- *Digital and analog inputs and outputs:* Most I/O terminals have 2 or 4 channels and are designed for various sensor/actuator connection methods

- *Switching devices:* Power-level terminals that switch, protect, and monitor three-phase standard motors
- *Voltage supply*: Electrically isolated groups can be set up within a block of automation terminals using power terminals and segment terminals
- *Bus terminals*: Bus terminals connect an INTERBUS Inline station with the remote bus or interface branches to INTERBUS Loop

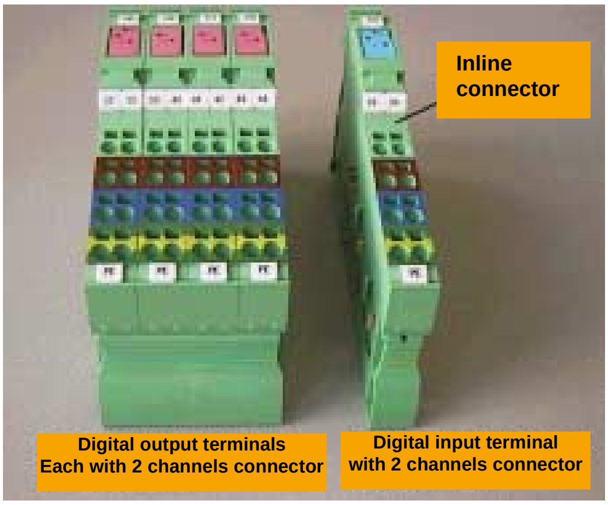


Figure 35: I/O automation terminals from the INTERBUS Inline product range

- *Control terminals:* These terminals can be used to interface distributed mini controllers, programmable according to IEC 1131, with INTERBUS.
- *Pneumatic terminals*: Pneumatic terminals can be used to connect pneumatic actuators directly to an Inline station. The required air is provided via air power terminals.
- *Special functions*: The available products include INTERBUS field multiplexers, which can transmit up to 512 digital signals over large distances (maximum 12 km [7.46 mi.]) between two Inline stations via a two-wire cable. Special terminals with safety functions are available for setting up emergency stop circuits.

The INTERBUS Inline terminals are connected internally via the 2-wire local bus. The connection to the remote bus is made using special Inline bus couplers (bus terminals).

The basic structure of an automation terminal and its integration into a complete Inline station is described below using the example of the IB IL 24 DI 2 digital input terminal.

IB IL 24 DI 2

Two sensors can be connected in 4-wire technology to the IB IL 24 DI 2 digital Inline input terminal with an Inline connector. The Inline connector is a standard connector for the spring-clamp connection of 2 x 4 signals. This connector is snapped onto the electronics base of the automation terminal (Figure 35).

Table 6 contains the essential technical data for the input terminal.

Table 6: Technical data for the IB IL 24 DI 2

Degree of protection	IP 20
Dimensions	(12.2 x 116.1 x 71.5) mm [0.480 x 4.571 x 2.815 in.]
Operating temperature	-25°C to +55°C (-13°F to +131°F)
I/O supply	24 V DC nominal value
Current consumption	30 mA from local bus
Number of inputs	2
Connection method	4-wire
Current per channel	3 mA, typical
1/0 signal	+15 ... +30 V/-30 ... +5 V
Delay time	In the range of μs

The *power supply* for the terminal electronics and the I/O is provided by the internal station potential routing system. Figure 36 shows the contact pins required for the jumpering system on the left-hand side of the Inline terminal.

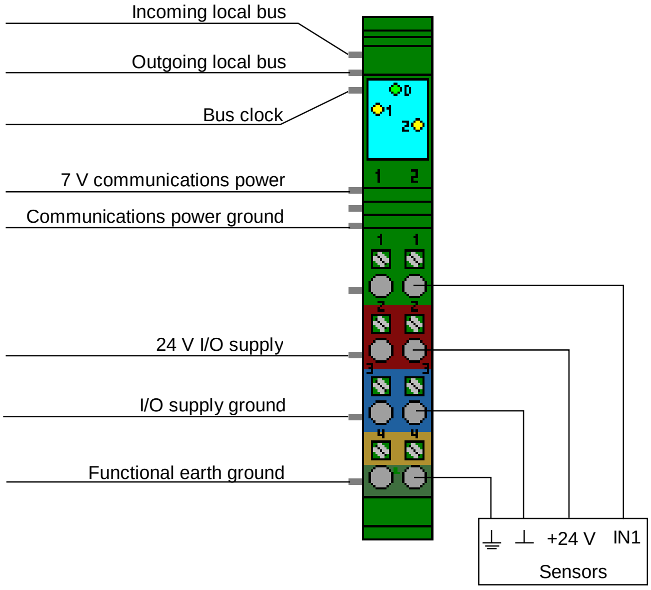


Figure 36: External wiring for the IB IL 24 DI 2 input terminal

The *sensors* are connected via spring-clamp terminals on the Inline connector. The connections are available with ground and functional earth ground.

The automation terminal is connected to INTERBUS via a local bus with a 2-wire protocol. Unlike the ST local bus, this local bus also transmits the bus clock. Physically, the local bus is an Inline potential jumper with three contacts.

The input status of both channels is indicated by two *diagnostic LEDs* (1, 2) on the front panel of the input terminal. An additional LED (D) uses different flashing sequences to indicate the bus status, I/O errors, and faulty devices upstream/downstream of the terminal (Figure 36).

Structure of an Inline Prototype Station With IB IL 24 DI 2

An Inline prototype station with 6 digital inputs and 4 digital outputs will be used as an example. The two-input circuits need to be fused separately for safety reasons. An electrically isolated I/O circuit, which can be switched via INTERBUS, should also be provided for all outputs. The station is constructed using the following automation terminals:

3 x IB IL 24 DI 2 (input terminal, see above)
3 x IB IL 24 DO 2-2A (output terminal, 2 outputs, 4-wire, 24 V/2 A per output)
1 x IB IL 24 SEG/F (segment terminal with fuse, 24 V DC, 5 A, maximum)
1 x IB IL 24 PWR-IN (power terminal, 24 V DC, 10 A, maximum)
1 x IBS IL 24 BK-T (bus terminal, remote bus ↔ Inline potential jumper)

Figure 37 is a diagram of the prototype station and the current wiring required for the I/O circuits.

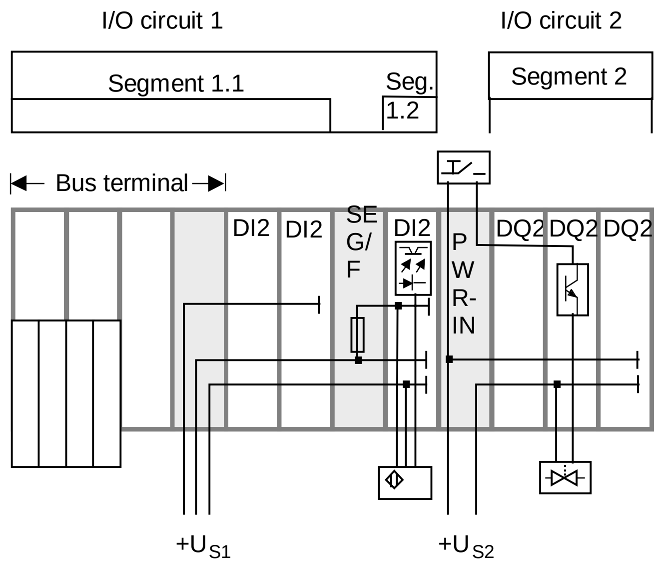


Figure 37: Inline prototype station with 6 digital inputs and 6 digital outputs

A *power and data rail system* (Inline potential jumper) is created automatically when the Inline automation terminals are mounted side by side. This means that the following circuits are created when the prototype station is assembled:

- *Data circuit* (not illustrated in Figure 37): All automation terminals are connected on INTERBUS. Each terminal is a separate INTERBUS device with a separate INTERBUS protocol chip.
- *Logic circuit* (not illustrated in Figure 37): The communications power, with 7 V for all Inline terminals, is supplied centrally via the bus terminal.
- *I/O circuit*: The I/O circuit voltage is used to supply the sensors and actuators. Any number of electrically isolated circuits from 24 V DC to 250 V AC may be present in an Inline station. The prototype station has two I/O circuits: I/O circuit 1 draws its power from the bus terminal supply and I/O circuit 2 is supplied with 24 V DC by the power terminal.

- *Segment circuit*: Segment terminals can be used to set up additional circuits within an I/O circuit. These segment circuits can be individually fused (segment 1.2 in Figure 37) or switched off/on (segment 2 in Figure 37) independently from the rest of the station.

Most automation terminals operate with a single process data channel. Terminals with more complex functions, e.g., IB IL 400 power terminals, can be parameterized via the parameter channel.

7.3 Individual Modules

In addition to the two specific INTERBUS installation device groups described above, numerous device modules are available as *individual modules*. These individual modules are often characterized by the availability of more complex functions and parameterization via the parameter channel.

In accordance with the above classification as device modules, all individual modules are supplied with IP 20 protection. They can be mounted either directly in control cabinets, terminal boxes or similar, or combined with other automation modules and installed as built-in devices in larger device or machine housings.

8 Addressing INTERBUS Data

The process and parameter data acquired in an INERBUS system must be assigned a unique address in the memory of the connected computer or control system (host system). Every INTERBUS device must therefore have a unique address in the host system.

On INTERBUS, this process can basically be described as a *routing level divided into three parts*, through which the INTERBUS data is transmitted in order to reach the host system. These routing levels are illustrated in Figure 38.

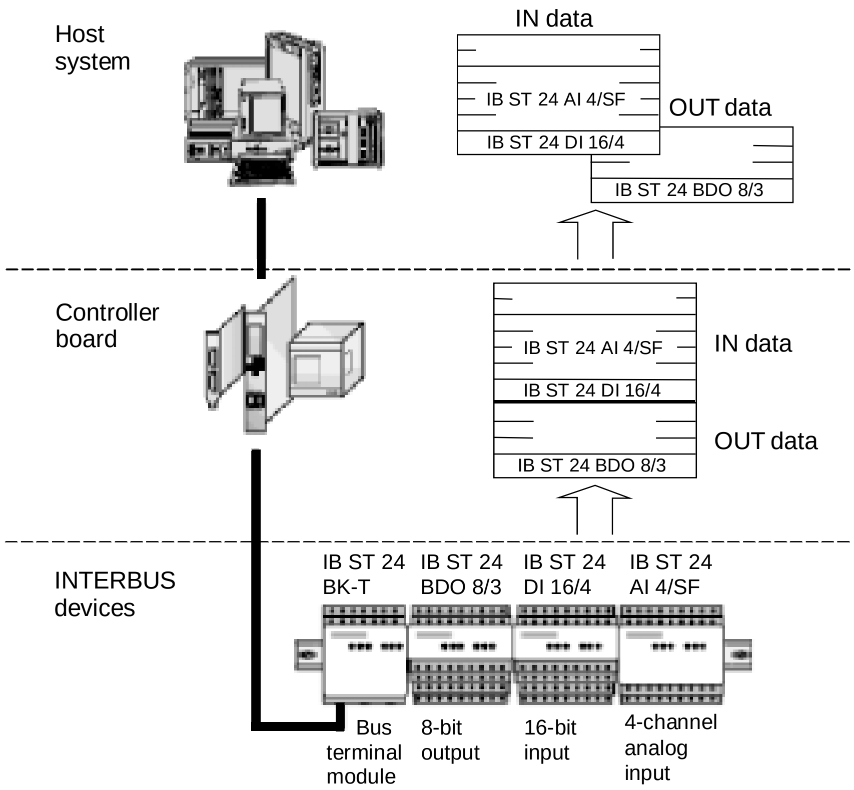


Figure 38: Basic address routing

The upper part of Figure 38 shows the memory area of the host system. Almost every host system organizes its memory area differently. For greater clarity, the illustration shows a memory organized on the basis of input and output data (example: Simatic S5 and S7 PLCs).

The lower part of the Figure shows the controller board, which, as the central function unit, manages the overall operation of the system as well as processing and transferring data between the host and the INTERBUS devices.

On the right of Figure 38 is the INTERBUS system, which receives/forwards the I/O signals (from/to the sensors and actuators).

The INTERBUS cycles, which are controlled by the controller board, send the signals from the I/O to a shared memory area on the controller board (coupling memory), which, depending on the type of controller board, will either be an MPM or DPM. The coupling memory is organized into several areas. Two of these areas are for the input and output data from the INTERBUS system. The controller board splits the INTERBUS device data into input and output data and stores it separately in the coupling memory. The controller board performs this process cyclically. The host system also accesses this coupling memory and copies the data to the host system memory area specified by the user.

The routing (assignment) between the INTERBUS I/O signals from the process I/O and the associated data saved in the host system can now be processed on INTERBUS using two different address methods. Both methods differ in terms of simplicity and flexibility.

8.1 Addressing Methods

With *physical addressing*, the INTERBUS device data is stored in ascending order in the coupling memory of the controller board, depending on its physical location on the INTERBUS data ring. The following conditions apply:

- According to its characteristics, the INTERBUS device data is stored separately as input or output data in the input or output data area of the coupling memory.
- The sequence in the coupling memory is defined from the controller board. The first INTERBUS device after the controller board is the first in the coupling memory. The second follows at the next available address (Figure 39).

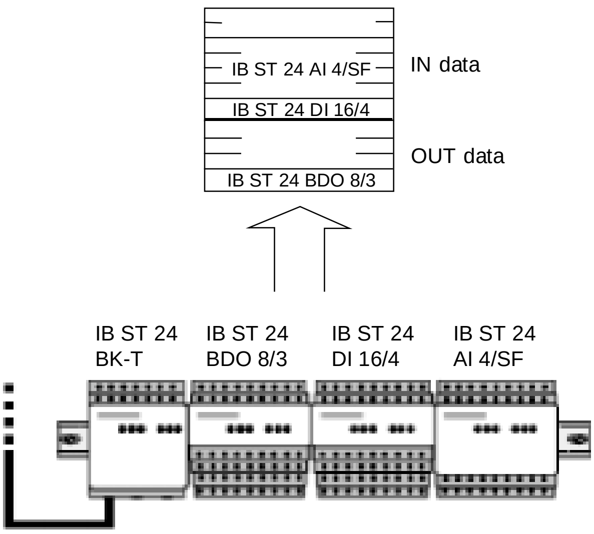


Figure 39: Physical addressing

Physical addressing is configured in two stages:

1. The controller board reads the ID codes to ascertain the configuration, register width, and type of each INTERBUS device and automatically addresses all devices. Only devices detected by the controller board in the INTERBUS system are addressed.
2. Depending on the type of controller board, the INTERBUS configuration read can be written to the parameterization memory (Flash memory) of the controller board together

with a startup sequence preconfigured in CMD. The bus configuration must also be saved to hard disk as a configuration file.

Every time the control system is switched on or the controller board is reset, the startup sequence is used to read and automatically assign the addresses from the controller board parameterization memory.

Very little effort is required for the physical addressing of the overall system with CMD software. However, changes to the bus configuration due, e.g., to the addition or removal of devices, require additional intervention in the user program, as such changes cause the input and output addresses of all connected devices to shift.

Unlike physical addressing, logical addressing or *user-defined addressing* assigns addresses in the host system memory freely to INTERBUS devices, irrespective of their actual physical order on the INTERBUS ring. User-defined addressing prevents addresses from shifting in the event of additions or changes to the INTERBUS system. User-defined addressing also enables the memory segmentation in the host system to be optimized and makes address assignment more transparent.

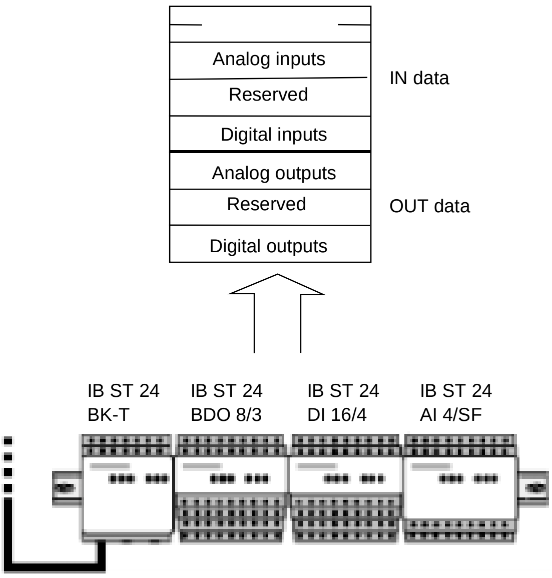


Figure 40: User-defined addressing

For user-defined addressing, the address list must be edited by the user according to requirements. The address list is then saved to the parameterization memory on the controller board. It is activated as described above by means of a preconfigured start sequence. The following issues are relevant for user-defined addressing:

- The free selection of the addresses makes the connection/disconnection of system parts in the bus configuration easier. In the event of changes in the bus configuration, it is only necessary to modify the address list and not all the addresses in the application program.
- System-specific addresses can be assigned, e.g., the bus segment number of a particular system part (e.g., control cabinet) can be set permanently.
- Devices with both input and output data can use the same numerical address in the corresponding input and output process image of the host system.

- Devices that only have one byte address area can be set to odd byte addresses. Addresses that are addressed byte by byte can fill any "byte gaps" arising as a result of physical addressing. Devices that are addressed word by word may only be assigned to even byte addresses.

Because of their flexibility and performance, user-defined addressing is applied almost without exception on complex systems and machines automated with INTERBUS.

8.2 Addressing in Programmable Logic Controllers

Because of their flexibility and performance, *user-defined addressing* is generally applied for programmable logic controllers (PLCs). Depending on the type of host PLC, the INTERBUS data area in the PLC memory can either be segmented randomly or on a block by block basis. For example, the SIMATIC S7-400 offers the following alternative data areas (illustrated in Figure 41) for user-defined addressing of the INTERBUS I/O data:

- *Process image:* This is the standard assignment. It requires a reserved input/output area for INTERBUS in the PLC process image.
- *Bit memory area:* INTERBUS data can also be stored in the bit memory area.
- *Data block:* INTERBUS data can also be stored in data blocks. This storage format is very easy to use and is recommended.

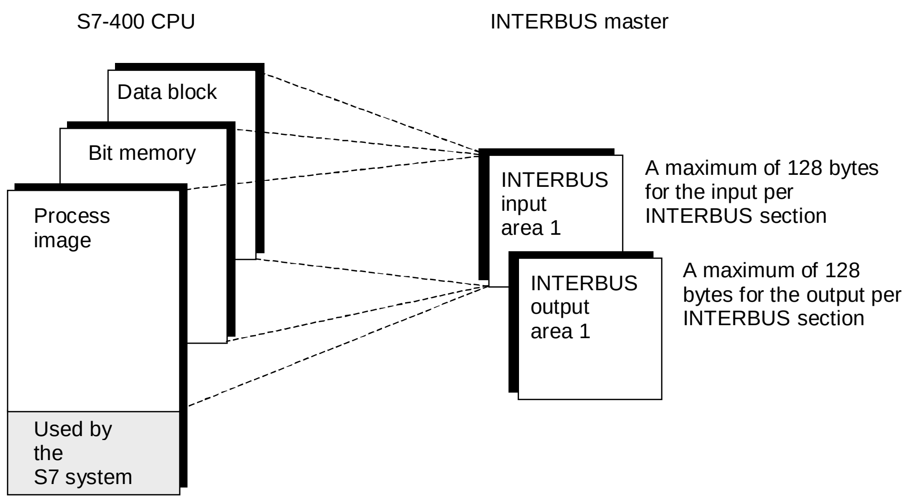
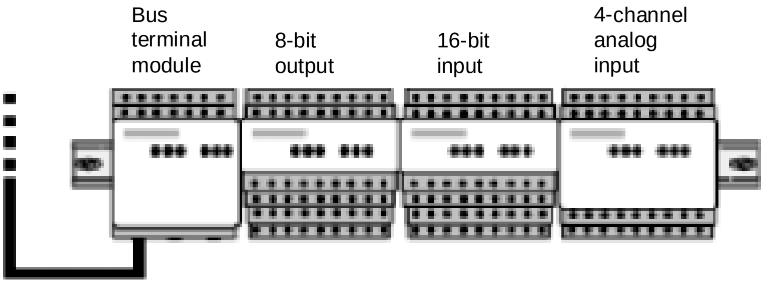


Figure 41: Alternative INTERBUS data areas in the SIMATIC S7-400

User-defined addressing can be configured in two ways:

- Addresses are assigned using the INTERBUS CMD configuration software
- Addresses are assigned via data blocks

Figure 42 shows an example of user-defined addressing. The device data field under the INTERBUS devices in Figure 42 (Table 7) contains all data relevant to user-defined addressing.



IBS ST 24 BK-T	IB ST 24 BDO 8/3	IB ST 24 DI 16/4	IB ST 24 AI 4/SF
1.0	1.1	1.2	1.3
8	185	190	126
0	1	1	4
0	1	1	1
–	–	P32	P34
–	P32	–	–

Figure 42: Example of user-defined addressing on an INTERBUS system

When *assigning addresses using CMD*, the addresses required for each INTERBUS device must be specified by the user in the PLC memory. Other data in the device data field is for information only. Relatively little effort is required for user-defined addressing. The parameterized addresses are saved to the parameterization memory.

Table 7: Device data field of an INTERBUS device

Device Designation
Bus segment number **.** Position in bus segment
Device code (dec.) = bits 0 … 7 of the ID code (see Appendix B)
Device data width in words
Device level number
Address of the input data in the PLC memory
Address of the output data in the PLC memory

Address assignment via data blocks may be advisable for PLC controller boards without a parameterization memory or in configurations subject to change. Data blocks containing lists of information from the device data fields must be created for this purpose.

8.3 Addressing in PC Control Systems

If the PC WORX automation software is used for PC controller boards capable of running control programs (e.g., Field Controllers), addresses can be assigned *without the need for explicit address configuration.*

This integrated system solution, which is based on INTERBUS device descriptions of the connected bus configuration, can use the I/O terminal designations in that configuration directly as process variables in the IEC 61131 control program. Users continue to work with functional components with which they are familiar from the process world, such as I/O terminals and signal designations, and do not require any additional knowledge about internal data paths, routing, and addressing.

9 Device Structure

An INTERBUS system can be set up by following the sequence below:

(1) System design and planning
(2) Device installation and connection
(3) Wiring of devices and process I/O
(4) Configuration and parameterization
(5) Startup and testing
(6) Operation, maintenance, and diagnostics

Steps 1 to 3 are required for the technical structure of the system and can be carried out with the assistance of standard tools and methods used when setting up automation solutions.

In order to be carried out efficiently and reliably, steps 4 and 5 require software tools able to support the activities concerned. INTERBUS generally uses the CMD tool for standard controllers and SYSTEM WORX for Field Controllers. In principle, the system can be configured, parameterized, started up, and tested without special tools in the standard programming environment.

Special tools are also required for step 6 and in particular for diagnostics and error detection, in order to detect errors correctly and in good time, and to rectify them quickly and reliably.

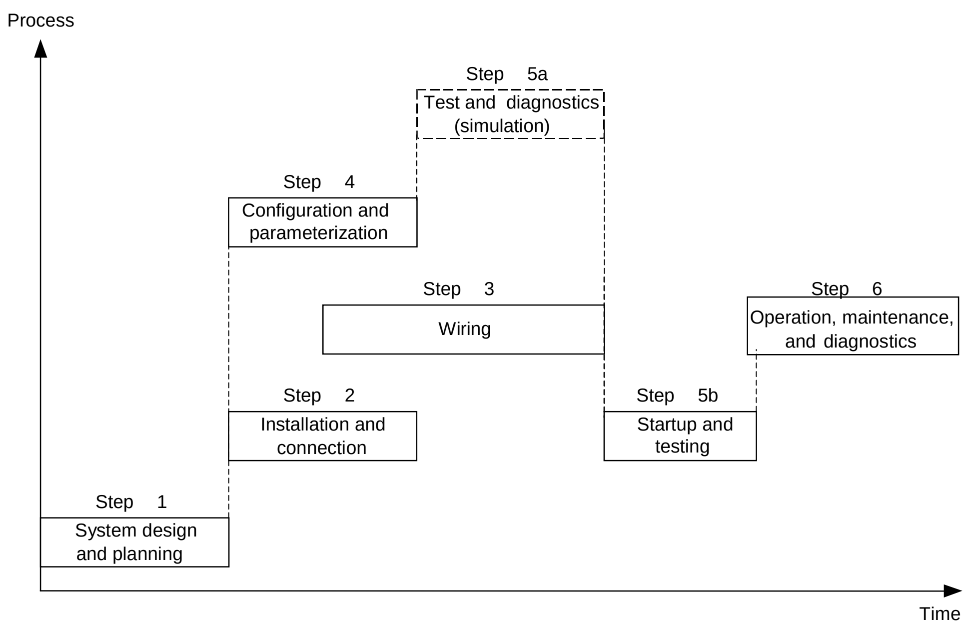


Figure 43: Sequence diagram for setting up an INTERBUS project

9.1 Designing and Planning the System

On the basis of the relevant technical and commercial application requirements, the general conditions for the INTERBUS network and the selection of INTERBUS modules required for the application are defined during the design phase. In addition to clarifying basic issues such as required cycle times, possible number of I/O points, and maximum distances between components (knock-out criteria), other questions have to be answered, including:

- Is the INTERBUS system one network or are there subnetworks?
- How are the levels within the subnetworks to be structured (local bus, remote bus or other bus segments)?
- Which control computer and which operating system will be used to operate the entire system and the subnetworks?
- How will the control programs be created (PLC methods or PC high-level languages)?
- How will data be exchanged between the INTERBUS system and higher-level control/management devices?
- Will parameter data be exchanged on INTERBUS (use of the PCP channel) and have the INTERBUS devices required for this been approved in accordance with the appropriate device profiles?
- Can maintenance and error diagnostics requirements be met with available tools?

Once these fundamental questions and requirements have been clarified, specific INTERBUS devices can be selected from available product ranges. During the planning phase, designs may have to be modified in order for targets to be met. In addition to the required INTERBUS devices, all other accessories and tools are also configured. These include in particular:

- Cables, lines, and connectors
- Power supply
- Terminal boxes and routing distributors
- Control cabinets
- Mechanical accessories (e.g., DIN rails, ground clamps)

As the largest manufacturer of INTERBUS products, Phoenix Contact can support users during the planning phase with an extensive product catalog on CD-ROM as well as an online catalog (http://www.phoenixcontact.com). This catalog can also be used to calculate costs and place orders.

9.2 Mounting

In terms of mechanics, INTERBUS modules are mounted in various ways, depending on the type of module and the class of protection. However, most modules can be classified as one of the following assembly types:

- Mounting on DIN rails with IP 20 protection

- Direct mounting

Modules *mounted on DIN rails* require a standard DIN rail according to DIN EN 50022 onto which the modules can be snapped. The modules are usually mounted with their long sides directly adjacent to one another. They should preferably be mounted horizontally.

For the purpose of illustration, Figure 44 shows the side view of a device module for IB ST DIO 8/8/3-2 A compact stations with the support for DIN rail mounting.

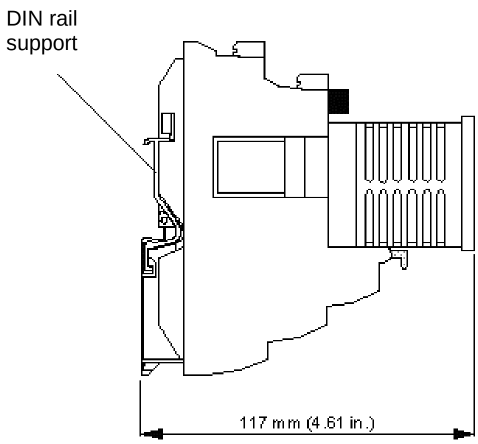


Figure 44: IB ST DIO 8/8/3-2A with support for DIN rail mounting

The DIN rail must be connected to protective earth ground (PE) via grounding terminal blocks. The modules are grounded when they are snapped onto the rail.

Direct mounting is recommended for field modules and special modules with higher degrees of protection (IP 54, IP 65, IP 67). The modules are usually components of remote buses, installation remote buses or INTERBUS Loop segments.

9.3 Connection and Cabling

The following connections and cables are required for the device structure of an INTERBUS system:

- Bus cabling
- PE connection
- Installation of power supply
- Sensor/actuator connection

The *bus cabling* comprises pre-assembled bus cables or bus cables equipped with connectors. In the case of copper-based bus cables, bus cable types can be defined according to individual bus segment types: remote bus cable, installation remote bus cable, and INTERBUS Loop cable. Various types of remote bus and installation remote bus cable are available, e.g., for permanent installation in cable raceways and cable channels, for highly flexible applications (flexible cable tracks), and outdoor and underground installation (UV resistant).

The standard connection technology uses D-SUB connectors (mainly 9-pos.), 8-pos. flat connectors, IP 65 circular connectors, and insulation displacement connectors.

Table 8 provides an overview of the commonly used bus cabling methods.

Table 8: Bus cables for INTERBUS cabling

Bus Segment Type	Connection Method	Cable Type
Remote bus	D-SUB connectors	Remote bus cable with 6 wires (3 x 2 twisted pair for data), shielded
	Flat connector (MINI-COMBICON)	
	IP 65 circular connector	
Installation remote bus	IP 65 circular connector	Installation remote bus cable with 9 wires (3 x 2 twisted pair for data, 3 for power supply), shielded
	Special screw and bolt connection (SAB connection)	
INTERBUS Loop	Insulation displacement connection method (QUICKON)	Twisted pair cable, unshielded

In local bus segments, the bus is cabled (depending on segment type) on segments with ST local bus using a special 5-pos. ST flat-ribbon cable. No specific bus cabling is required within the local bus segment for automation terminals with an INLINE connection, as this is created automatically via the potential jumpers.

The *protective earth ground* of an INTERBUS module is created either by grounding the DIN rail in the case of DIN rail mounting or by means of a separate PE connection for each module in the case of direct mounting.

Depending on the design of the entire system, the *power supply* for the INTERBUS modules can either be central or distributed. The power supply for the modules must be provided as appropriate for the bus segment type, either separately or with the bus cable. The supply voltage is usually at 24 V DC (voltage range 20 V … 30 V DC) with a ripple of 3.6 V_{PP}. Table 9 provides an overview of the power supply in the different bus segments.

Table 9: Power supply for INTERBUS modules

Bus Segment Type	Power Supply for the Electronics Module
ST local bus	Supply via external cables, potential connectors can be used within a compact station
INLINE segment	Internal supply via INLINE potential jumpers
Remote bus	Supply via external cables
Installation remote bus	Internal supply via special lines in the bus cable
INTERBUS Loop	Data and supply voltage via one cable

INTERBUS modules can provide one, two, three, and four-wire connection types for the universal *connection of sensors and actuators* (Figure 45).

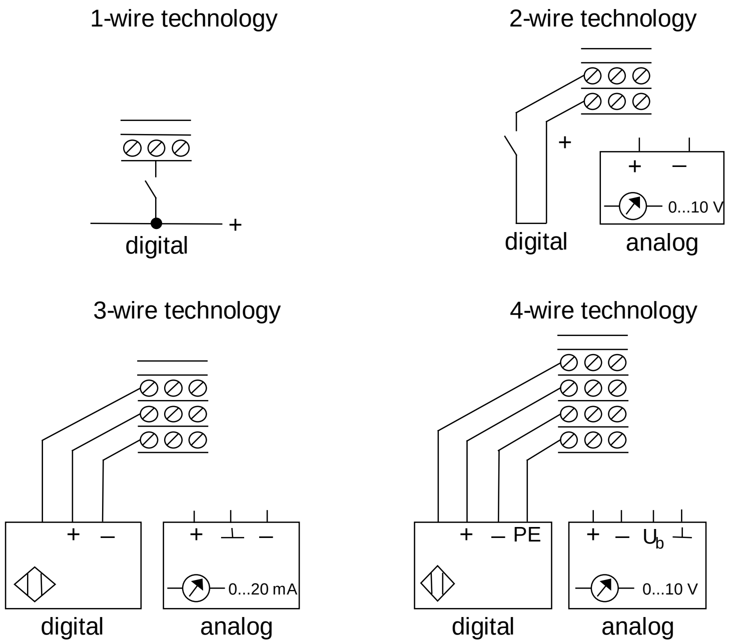


Figure 45: Sensor connection types on INTERBUS input modules

10 Diagnostics in INTERBUS Systems

10.1 INTERBUS Diagnostic Concept

INTERBUS applies a graded diagnostic concept, which is able to monitor and diagnose all devices, from individual sensors and actuators to controller boards (system diagnostics). The diagnostic data is evaluated by the hardware in the individual INTERBUS components or by the software either in the application or using special diagnostic programs.

Table 10 shows the areas covered by system diagnostics in a distributed automation system with fieldbus systems and the areas that can be diagnosed by INTERBUS.

Table 10: System diagnostics in a distributed automation system

Diagnostic Level	Diagnostic Data	Can Be Diagnosed by INTERBUS
Management	Process visualization, database	No
Application	Control program	No
	I/O monitoring	Yes
Controller board	Data flow and bus monitoring	Yes
Bus system	Transmission media, transmission reliability	Yes
I/O modules	Connection of sensors/actuators, transmission functions	Yes
Sensors/actuators	Signal detection	To some extent
	Activation	No

The basis of the diagnostic structure on INTERBUS determines the basic system behavior with powerful mechanisms for intrinsic safety:

- Detection of errors on the transmission path and location of short circuits on the line by dividing INTERBUS into individual bus segments, which can be switched using the switches integrated in the devices
- Location of temporary malfunctions by running a CRC check on each individual transmission path simultaneously in all devices
- Increased system availability through the creation of subsystems/system parts, which can be activated and deactivated via bus terminal modules

In addition, all INTERBUS components, such as controller boards, the bus system, and INTERBUS devices, are integrated into the diagnostic concept.

Diagnostic data is accessed via various diagnostic interfaces. These provide the user with optimized access. INTERBUS differentiates between two basic interface types:

- Visual indicators
- Software diagnostics

10.2 Visual Indicators

Diagnostic data can be displayed visually by LEDs or LCDs on controller boards, bus terminal modules, and bus devices. This type of diagnostics enables service personnel to diagnose INTERBUS systems without the need for additional tools and to detect system states. The distributed structure of the INTERBUS system is reflected in the distribution of the visual indicators: in a central location (on the controller board) and in remote locations (on bus terminal modules and bus devices).

Many Generation 4 PLC controller boards or later have an LCD for *central diagnostic indication*, which indicates every error that occurs in the bus system and on the connected devices with information relating to the cause of the error, its location, and an additional error code.

Figure 46 shows the diagnostic display on the front plate of a PLC controller board.

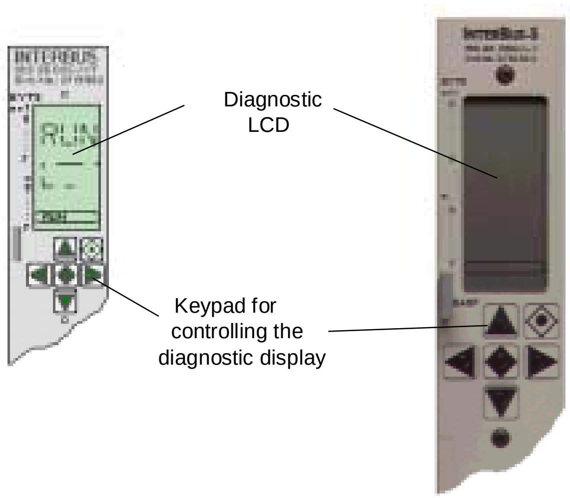


Figure 46: Diagnostic display on the front plate of a PLC controller board

Each diagnostic display is controlled by a menu-driven keypad. In addition to the basic diagnostic display, various other operating modes can be activated via the operator menu for displaying all relevant configuration and diagnostic data relating to the system.

In *distributed diagnostics*, colored LEDs indicate the monitoring results (diagnostic data) as good/bad locally on the bus components. The number and type of local diagnostic LEDs largely depend on the type of INTERBUS device. The basic capabilities of these local diagnostics are however identical for most bus devices, as they are based on diagnostic signals, which are supported by the INTERBUS slave chip used. Typical diagnostic signals for an I/O remote bus device with the SUPI include:

RC: Remote bus check - monitoring of the incoming remote bus cable

BA: Bus active - indicates activity on the bus
E: An error has been located in this INTERBUS module
RD: Remote bus disabled - indicates that the outgoing remote bus has been disabled

Additional LEDs may be used, depending on the module type:

LD: Local bus disabled (on bus terminal modules)
UL: Electronics module power supply present (e.g., all modules in the ST range)
U_{S1}, U_{S2}, ...: Supply voltage for I/O groups (I/O modules)
CC: Cable check for local bus cable OK (local bus modules)

Digital I/O modules also have status LEDs for displaying the binary input and output signals. Figure 47 shows an example layout of diagnostic LEDs on an IBS ST 24 BKM-T bus terminal module.

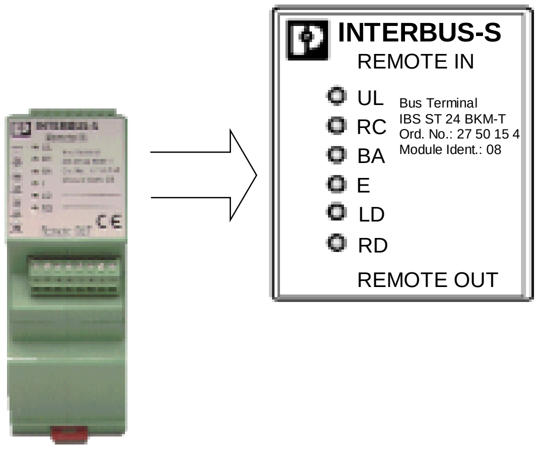


Figure 47: Local diagnostic LEDs on a bus terminal module (type IBS ST 24 BKM-T)

10.3 Software Diagnostics

Software diagnostics means a central diagnostics function for the complete INTERBUS system, which can take two forms:

- User-specific diagnostics in the control program
- Diagnostics with software tools

The diagnostic registers of the control board can be used for *user-specific diagnostics*. These registers map the diagnostic display to the control system.
Specific software components, which are often available to PLC host computers as preconfigured PLC program blocks, are required in the control program for evaluating the registers. It is for this reason that this type of diagnostics function almost always runs together with the control program on the PLC host system.

Diagnostic and error data can also be accessed using INTERBUS management (IBSM) services via the INTERBUS firmware for the relevant controller board. Although it is relatively difficult to integrate these services into a user program, they do enable the system to be diagnosed under all possible conditions.

By far the most frequently used *software tool for diagnostics* on INTERBUS systems is PC WORX/the CMD tool. These tools have a special diagnostics part with which an INTERBUS system can perform simple practical checks for and statistically evaluate messages relating to malfunctions and errors across the entire system and locally (device-specific).

11 Open Automation With INTERBUS

11.1 Industrial PC Technology

Today, mechanical engineers and system builders wishing to sell their products all over the world must equip their products with automation technology appropriate for each intended county of use. This requires a great deal of effort during the planning, configuration, programming, and startup phases. This effort can be reduced considerably by using a piece of control hardware that is available throughout the world - the industrial PC.

Surveys of development trends optimistically predict *an increase in PC-based control systems* on the European market between 1998 and 2004 by approximately 845% (Figure 48).

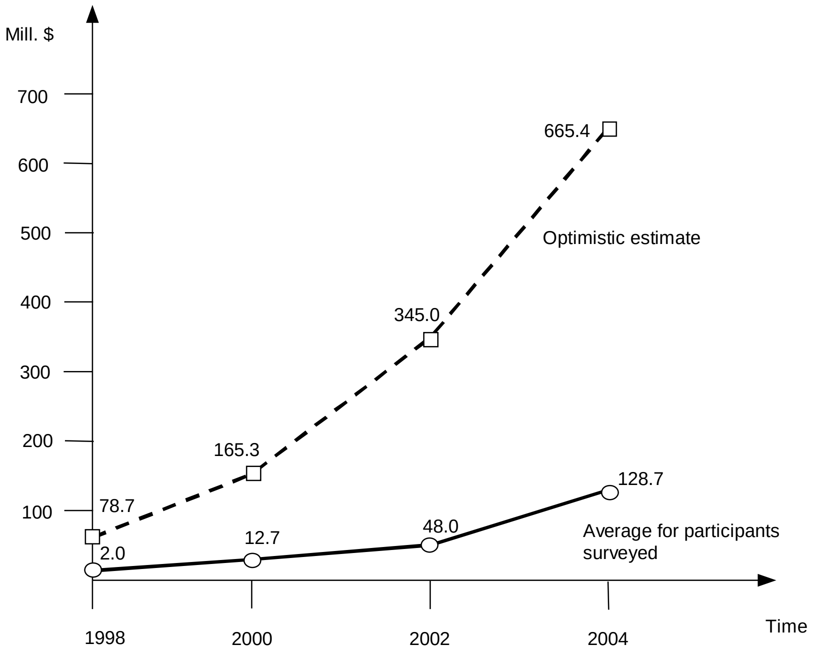


Figure 48: The European market for PC-based control systems (survey of 500 large IT manufacturers)

Similar results to those shown in Figure 48 were recorded by other surveys. Although the actual share of PC control technology is relative low (just a few % of the total volume of control computers), the immense growth rates in the coming years should see the market volume of PC control computers rise to between 10 and 20%. It should also be noted that, in the past, growth rates predicted for microprocessors and subsequently for PC technology have generally proved too low.

As the share of PC technology in the control market increases, development trends are turning their focus towards the following areas:

- Development of industrial PC hardware
- Use of standard software as industrial software

11.2 Data Interfaces

In order to solve the communication problems between applications in a Windows environment and the automation I/O in the world of production, several general de facto standards have been developed for data interfaces, which are already being used to meet numerous requirements:

Dynamic data exchange (DDE): This concept embraces a variety different ways of presenting data and then exchanging it between two devices according to the client/server principle.

- *Object linking and embedding* (OLE): This type of data exchange is also based on the client/server principle. It is designed to group independent objects from various applications and access them in their different formats from another application.

  Microsoft has developed OLE so that in addition to data and methods, the events of an embedded object can now also be exchanged. This is made possible by defining a general model for component objects (COM - Component Object Model). Another model has been developed for distributed COM architectures to support the linking and embedding of objects on different computers (DCOM - Distributed Component Object Model).
- *ActiveX*: The basic idea behind ActiveX is the implementation of small and easy-to-load objects with limited and fixed internal functions, which can be linked to form a complete system. The functions provided are described using individual objects. ActiveX can be used to exchange individual data items or events in the same way as complete visual displays, for example. COM technology is used to convert ActiveX.
- *Dynamic link libraries* (DLL): An appropriate program part is loaded when required during runtime for data exchange with DLLs. This program part accesses the data memory of a second application directly in order to exchange data in individual formats. OLE and ActiveX are often implemented using dynamic libraries.

  In automation technology, DLLs are a standard way of exchanging data under Windows. Data interfaces are also available as DLLs for INTERBUS.
- *Open database connectivity*: Another standardized interface for data exchange between individual components is open database connectivity (ODBC). The primary purpose of this interface is to enable individual items of data to be written to a database in standard format without having to select a specific database for the purposes of data exchange.

In addition to the data exchange interfaces suitable for general use under Windows, recent industry requirements have prompted the development of products to meet the specific needs of automation technology. The focus of these activities has been *OPC* (OLE for process control).

In 1996, many companies, with the participation of Microsoft, formed the *OPC Foundation* in Austin/Texas, USA, and defined an *OPC specification* for data exchange in the process industry based on Microsoft's OLE/COM technology. Clients connected to servers (e.g., visualization and process control systems) can use OPC to access data from those servers. In the same way as a printer is selected on a network, OPC enables the server from which data is to be retrieved, e.g., a PLC or fieldbus system, to be defined. Figure 49 illustrates an OPC client/server connection.

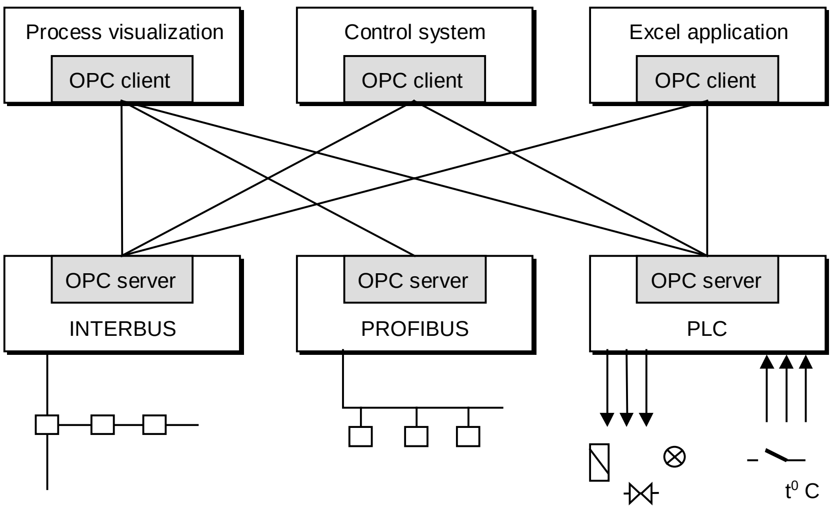


Figure 49: Principle of an OPC client/server connection

Essentially, OPC provides services for reading and writing variables. The OPC server contains a number of items (= variable objects), which represent connections to actual data objects. Within the OPC server, each item has a unique identification. These items can be created in various ways. The items that can be accessed by each client are arranged in groups on the server. Items within a group can be read or written synchronously or asynchronously.

The OPC specification is still being developed. In addition to process data exchange management, new OPC specifications cover alarm and event management and historical trends. OPC clients/servers are already available for the majority of process visualization systems and fieldbus systems.

12 PC WORX Automation Software

The PC WORX automation tool is part of the Open Control concept for PC-based automation solutions with INTERBUS. PC WORX combines three well-known programs to create a harmonized software tool. The required data is available to all PC WORX modules in a shared database. Open software interfaces mean that data can be exchanged easily with other software packages.

As shown in Figure 50, the tool comprises the PC WORX configurator, PC WORX programming, and PC WORX visualization. The entire system can be scaled and adapted to meet the requirements of specific applications.

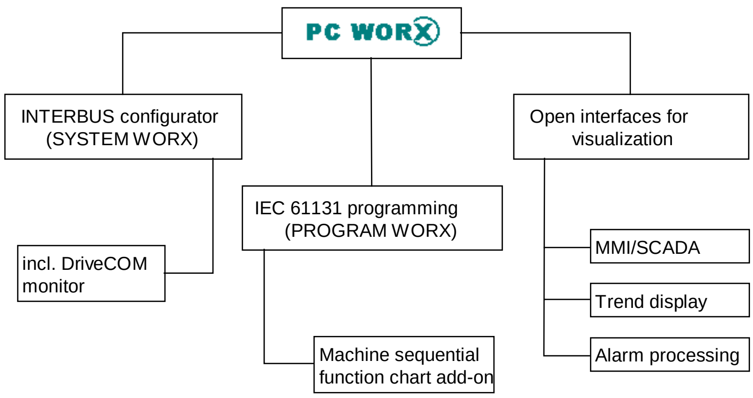


Figure 50: PC WORX component structure

PC WORX is available for all Windows systems and can be used in conjunction with the INTERBUS Field Controller.

12.1 INTERBUS Configurator

The PC WORX configurator (also known as the INTERBUS configurator or SYSTEM WORX) is used to configure, parameterize, and diagnose the entire INTERBUS network and the connected devices. The configurator can be used to configure and parameterize both individual INTERBUS I/O devices and intelligent INTERBUS nodes (system couplers, Remote Field Controllers, etc.) for specific tasks. In terms of its functions and operation, the PC WORX configurator is essentially the same as the CMD tool. However, the configurator provides additional functions as part of the Open Control concept:

- The INTERBUS configuration data is stored in a shared database, which can be accessed by other PC WORX components.

- During configuration, I/O process signals can be assigned as process variables for the control program. These can then be used when developing the control program and do not have to be defined again. Figure 51 illustrates an example of this:

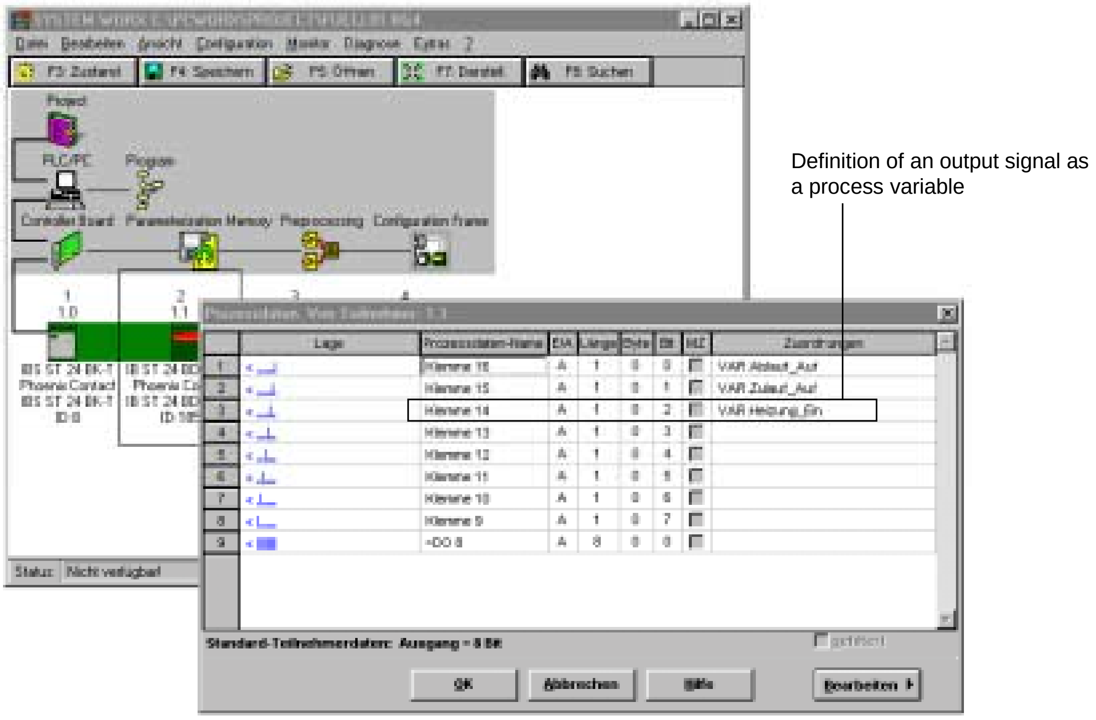


Figure 51: Defining a digital output signal as a process variable with the PC WORX configurator

The digital output signal at terminal 14 for activating the heating in a boiler system (example) is defined once during INTERBUS configuration as process data item Heating_ON and can be used again as process variable VAR.Heating_On in the IEC 61131 control program.

- It is possible to switch directly from the INTERBUS configurator to the PC WORX programming environment (and vice versa).

INTERBUS OPC servers can also be configured with the PC WORX configurator.

12.2 IEC-61131 Programming

PC WORX programming (also known as PROGRAM WORX) is a piece of programming software for INTERBUS control systems based on IEC 61131. The software is based on the VxWorks runtime environment.

Work began on the IEC 61131 "Programmable Controllers" standard in 1979 under the guidance of the IEC. The standard collates experiences gained in the field of control technology and, in recent years, has become a standard programming environment for PC-based control systems.

Taking the hardware and software model of a modern control structure as its basis, Part 3 (IEC 61131-3) describes five powerful languages for structuring and programming:

- Function block diagram (FBD)
- Instruction list (IL)
- Sequential flow chart (SFC)
- Ladder diagram (LD)
- Structured text (ST)

PC WORX programming supports all five IEC 61131 languages as well as an additional application-specific machine sequential function chart language (MSFC).

12.3 Visualization

As an option, PC WORX supports the integration of various process visualization and control systems. The data required for visualization can be imported into all systems directly from the PC WORX programming environment or from standard Windows programs (Excel, Access, etc.) via the Windows data interfaces (DDE, OLE, ODBC). Data exchange via an OPC client/server interface is also possible.

Index

Appendix

Appendix A: Important Addresses

Manufacturer

Phoenix Contact GmbH & Co. KG
Postfach 1341
D-32819 Blomberg, Germany
or
Flachsmarktstr. 8-28
D-32825 Blomberg, Germany
Phone: +49 - 52 35 - 34 12 00
Fax: +49 - 52 35 - 30 0
E-mail: info@phoenixcontact.com
Website: http://www.phoenixcontact.com

INTERBUS Clubs

INTERBUS Club Deutschland e.V.
Head Office
Postfach 1108
D-32817 Blomberg, Germany
Phone: +49 - 52 35 - 34 21 00
Fax: +49 - 52 35 - 34 12 34
E-mail: german@interbusclub.com
Website: http://www.interbusclub.com

Open Control Concept

OPC Foundation
20423 State Road 7
Suite 292
Boca Ralon, FL 33498
Phone: (561) 477 – 1375
Fax: (561) 477 – 0520
E-mail: opc@powerinternet.com
Website: http://www.opcfoundation.org

Appendix B: INTERBUS Documentation

Langmann, Reinhard: INTERBUS-Technologie zur Automation (INTERBUS Technology for Automation), published by Carl Hanser Verlag, Munich, Germany, 1999

Jansen, W., Blome, W.: Interbus, published by Verlag moderne industrie, Bibliothek der Technik, Volume 162, Landsberg/Lech, Germany, 1998

Baginski, A.; Müller, M.: INTERBUS, published by Hüthig Buch Verlag, Heidelberg, Germany, 2nd Edition.,1998

Phoenix Contact (ed.): Grundkurs Sensor/Aktor-Feldbustechnik (Sensor/Actuator Fieldbus Technology Basics),published by Vogel Verlag, Würzburg, Germany, 1997

Nickel, D.: InterBus-S-Installation (InterBus-S Installation), published by Pflaum Verlag, Munich, Germany, 1994

The table below contains a brief summary of each of the specialist books.

Specialist Book Title	**Main Focus**	**Supplied With**	**Target Group**
Process Automation With DIN Measurement Bus and InterBus-S (1993)	Basic knowledge, circuit diagrams for 2-wire remote bus and 8-wire local bus devices with SUPI 2	INTERBUS PC drivers (floppy disk)	Electronics engineers, students at technical colleges and universities
InterBus-S Installation (1994)	Basics, structure of INTERBUS, startup and diagnostics, source text for level control with Siemens S5 and PC (MSDOS)	Control program for level control (floppy disk)	Technicians, maintenance and service personnel
Sensor/Actuator Fieldbus Technology Basics (1997)	General introduction to sensor/actuator fieldbus systems, INTERBUS system description, selected aspects of INTERBUS practice	-	Fieldbus technology beginners, students at technical colleges and universities
INTERBUS (2nd Edition, 1998)	INTERBUS system knowledge, with the focus on PCP communication	-	Students of INTERBUS and engineers, device developers
Interbus (1998)	Brief introduction to the INTERBUS system, marketing-based	-	Technical managers, sales and marketing personnel
INTERBUS Technology for Automation	Complete and systematically ordered overview of INTERBUS technology, focusing on its use in open automation solutions	INTERBUS Learning V2.0 (CD-ROM)	Students of INTERBUS and engineers, lecturers and trainers, users and device developers