SINUMERIK 840/840C
SINUMERIK 880/880 GA2
Computer Link
General Description

Planning Guide 09.95 Edition

Manufacturer Documentation

SINUMERIK 840/840C
SINUMERIK 880/880 GA2
Computer Link
General Description

Planning Guide

Manufacturer Documentation

Valid for:

Control	*Software version*
SINUMERIK 840	from 1
SINUMERIK 840C	from 1
SINUMERIK 880 T/M	from 3
SINUMERIK 880 GA2 T/M	from 1

September 1995 Edition

SINUMERIK® documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the ”Remarks” column.

Status code in ”Remarks” *column*:

A . . . New documentation **B** . . . Unrevised reprint with new Order No.
C . . . Revised edition with new status. If factual changes have been made on the page since the last edition, this is indicated by a new edition coding in the header on that page.

Edition	Order No.	Remarks
08.89	6ZB5 410-0CB02-0BA0	**A**
06.91	6ZB5 410-0CB02-0AA1	**C**
09.95	6FC5 197-0AB20-0BP0	**C**

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

This publication was produced on the Siemens 5800 Office System.
Subject to change without prior notice.

Preliminary Remarks

Reader guidance

This documentation is intended for manufacturers and users who wish to obtain information on the computer link with SINUMERIK 840/880.

The description provides a general overview of the computer link, focussing on the computer link with SINUMERIK 840/880.

It is meant as a first introduction for users who are not yet familiar with the computer link with SINUMERIK 880 as it describes the basics for the message frame traffic. The manual "SINUMERIK 840/880 Computer Link General Description" is part of a documentation set on the 840/880 computer link that is organized as follows:

- General documentation
- Configuration documentation
- Installation and service documentation

The term ”SINUMERIK 8X0” is used in this description as a short form for ”SINUMERIK 840/880”.

Technical information

This documentation applies to software version 1 and higher in the case of SINUMERIK 840 and 840C to software version 3 and higher in the case of SINUMERIK 880 GA2 to software version 1.

Contents

1 General

Communication systems are used in various areas and, depending on their application and environment, they must meet specific quality requirements in order to fulfil their tasks.

Local communication networks are used in three major fields of application:

- Laboratory
- Factory
- Office

This description deals only with communication in the factory environment via industrial bus and serial connections.

Industrial bus according to MAP

Let us briefly review the history of the MAP industrial bus. General Motors (GM) approached all electronics manufacturers at the end of the seventies asking them to cooperate. GM have worked on LANs since 1979. As early as 1980 they started to develop the MAP protocol. During the NCC 84 US computer show in July 1984, GM introduced a local area network on the basis of a broad band token bus. They showed a MAP network up to the fourth (transport) layer, using different computer and control systems of IBM, Hewlett Packard, Digital Equipment, Motorola, Allen Bradley, Gould and Concord Data Systems. At the September 1985 Autofact in Detroit, Siemens was represented on the bus with a programmable controller together with other companies. The SIMATIC S5-150S was connected to the token bus via a 301 EU interface module and the 310 Intel development system.

SINEC H1

The SINEC H1 industrial bus (ETHERNET) has been implemented by the Siemens Automation Group. This bus ensures the heterogeneous integration of Siemens programmable controllers.

A uniform protocol is necessary to enable communication to take place with different automation equipment such as numerical controls, personal computers, process control computers etc. in a production automation environment.

For SINEC H1 the framework protocol architecture has been implemented using international standards in the transport-oriented layers (1 to 4) and the SINEC-AP1 Siemens standard in the application-oriented layers (5 to 7).

Industrial buses

General quality specifications
• Security Data must not be falsified or get lost without an error message being displayed • Performance The communication performance must meet the requirements • Functionality The functions (services) required by the user have to be provided in the required quality • Uptime Communication systems are often major parts of a system, and the uptime of the entire system can depend on them to a large extent • Cost-effectiveness The cost / performance ratio must be reasonable.

In addition to the general quality specifications, additional requirements apply to communication systems in the process automation environment.

Additional quality specifications for process data communication systems
• Ability to respond Data must proceed from the transmitter to the receiver within a defined period of time, which is usually very short. • Trouble-free operation Operation in close proximity to power installations must be possible. The system must adequately resist mechanical and climatic stresses.

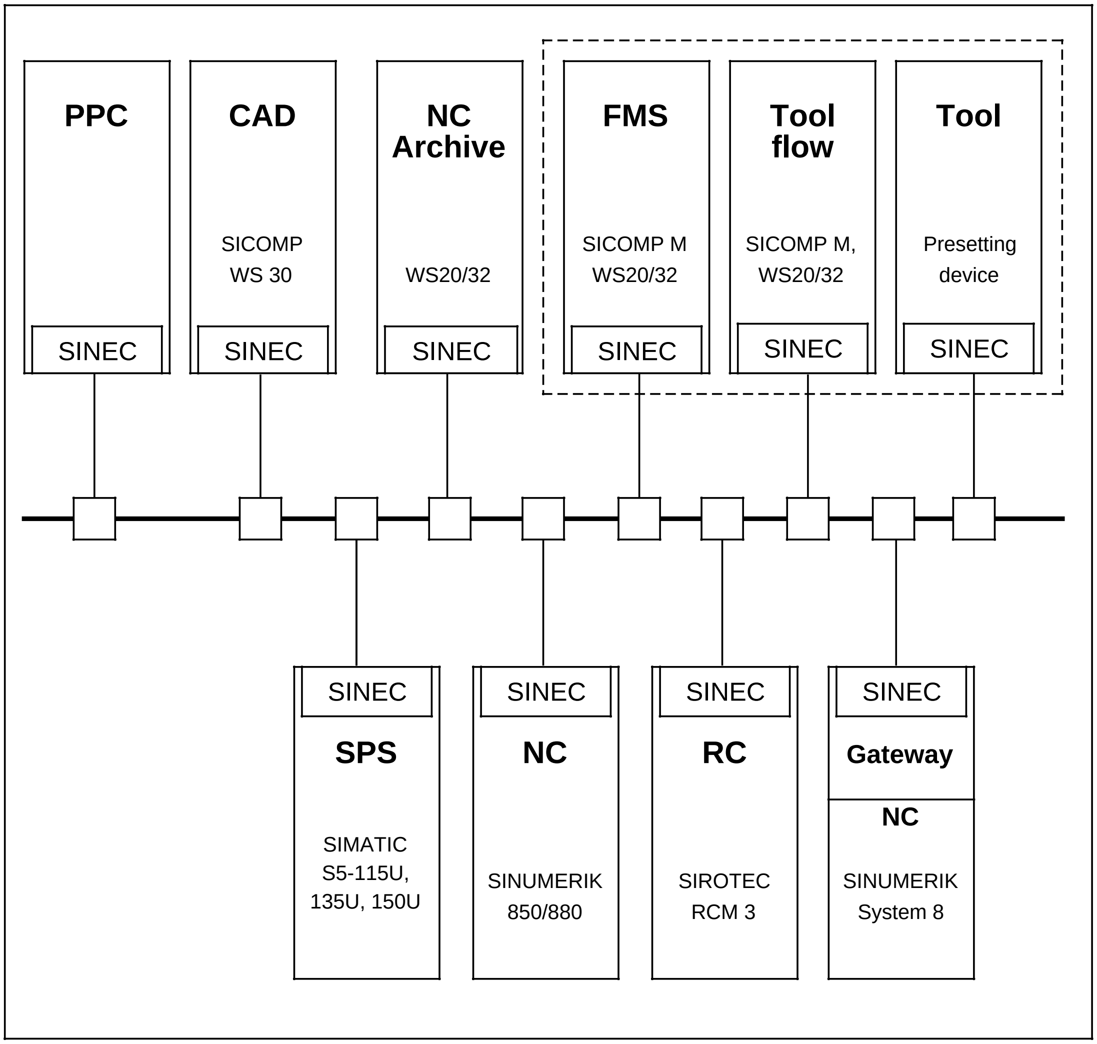
1.1 Definition of terms

1.1.1 Computer link

A computer link enables the exchange of data and control information between computers and controls (e.g. CAD, FMS, ... , NC, RC, SPS ... , as shown in figure SINEC H1) on the initiative of a computer, control system or operator. The data exchange mainly covers workpiece programs, tool data, workpiece data, control functions (e.g. NC start, NC stop), operating data such as status messages, alarm and error messages.

A computer link can be realised using a bus connection or using one of many serial connections.

The following figure shows the SIEMENS computer systems used for CAD, FMS, NC Archive and PPC as well as the RC, NC and PLC systems on a common bus cable. If older systems are to be connected to the bus, e.g. SINUMERIK System 8, this is done via a gateway station.



Bus system: SINEC H1

1.1.2 ISO reference model

For the data exchange between systems various prerequisites have to be met by the two communication partners. Both stations must have compatible physical interfaces, and the form of information routing, addressing, the type of error checking and the format of the information have to be agreed by both communication partners.

For this reason, the International Standards Organization (ISO) has established a model for communication procedures known as the ISO reference model.

It consists of a seven-layer hierarchy which is organized as follows:

- Layer 1 Physical layer
- Layer 2 Data Link layer
- Layer 3 Network layer
- Layer 4 Transport layer
- Layer 5 Session layer
- Layer 6 Presentation layer
- Layer 7 Application layer

Further information on the individual layers and their functions are contained in Section 2.1.1, ”The ISO seven layer model”.

1.1.3 LAN

ISO defines local area networks (LAN) as networks for bit-serial transmission of information between interconnected independent devices (e.g. computers, microcomputers, programmable controllers, numerical controls), i.e. the interconnection of autonomous systems.

”These are completely controlled by the users and limited to their site”.

In factory plants, communication and exchange of information between the individual devices is to take place in such a way that direct data access to all stations is ensured.

1.1.4 MAP

MAP (Manufacturing Automation Protocol) is a new communication protocol which is based on international and US standards as well as additional specifications by GM. This protocol allows an open communication in the factory (open networks) i.e. devices of different manufacturers can be interconnected (opposite: closed networks).

1.1.5 SINEC AP 1

SINEC AP 1 is the Siemens Network Architecture Automation Protocol Version 1, a Siemens automation protocol. It was developed for the integration of the following systems in a factory automation system:

- SICOMP computers
- SINUMERIK
- SIROTEC
- SIMATIC S5
- Programmers
- MMC 216
- Concentrators
- Personal computers
- Workstations

The protocol covers the OSI layers 5 to 7. For further information please refer to Section 3.3.1, ”SINEC AP 1 protocol”.

1.1.6 Transport connection TPV

The term transport connection describes a virtual connection between two nodes in a network which, in accordance with the ISO reference model, is also referred to as layer 4 connection since it reaches to layer 4 of the respective communication partner.

In the case of a serial link, the transport connection need not be specified since an unambiguous hardware allocation is made by the interface used.

The transport connection is determined by addresses which are clearly assigned when the network is configured. By means of the addresses the two end points of the transport connection are allocated.

In the case of a computer link with SINUMERIK 8X0, a transport connection is determined by the address pair's own Ethernet address, the internal TSAP (Transport Service Access Point) and the Ethernet and TSAP address of the partner. The addresses or the allocations for the corresponding transport connection are stored in the interface module of the SINUMERIK 8X0.

The transport connection provides a reliable communication connection to transfer the application-oriented information and it manages the transfer-specific services.

Several transport connections can be configured for one system. For example, a different transport connection can be used for the file transfer (NC data exchange) than for the loading/unloading of tools.

Access to the transport connection is from the application-oriented layers via the transport service access point (TSAP), which has its own unique address (TSAP address).

1.1.7 Logical peer

Logical peer is the symbolic designation for a technological function module in a system, e.g. in the SINUMERIK 8X0.

A system can contain one or several function modules.

In the case of a bus connection, a "logical peer" is identified by the combination of transport connection and addressing in the application layer.

One or several logical peers can be served via a TPV transport connection. (When connecting the SINUMERIK 8X0 to the Siemens FMS only one logical peer per transport connection is allowed.)

The "logical peer" is determined by the addressing in the application layer alone in the case of a serial connection.

If the AS 512 protocol is used, several logical peers can be allocated to one interface.

If an interface is configured without the AS 512 protocol, only one logical peer can be allocated to it.

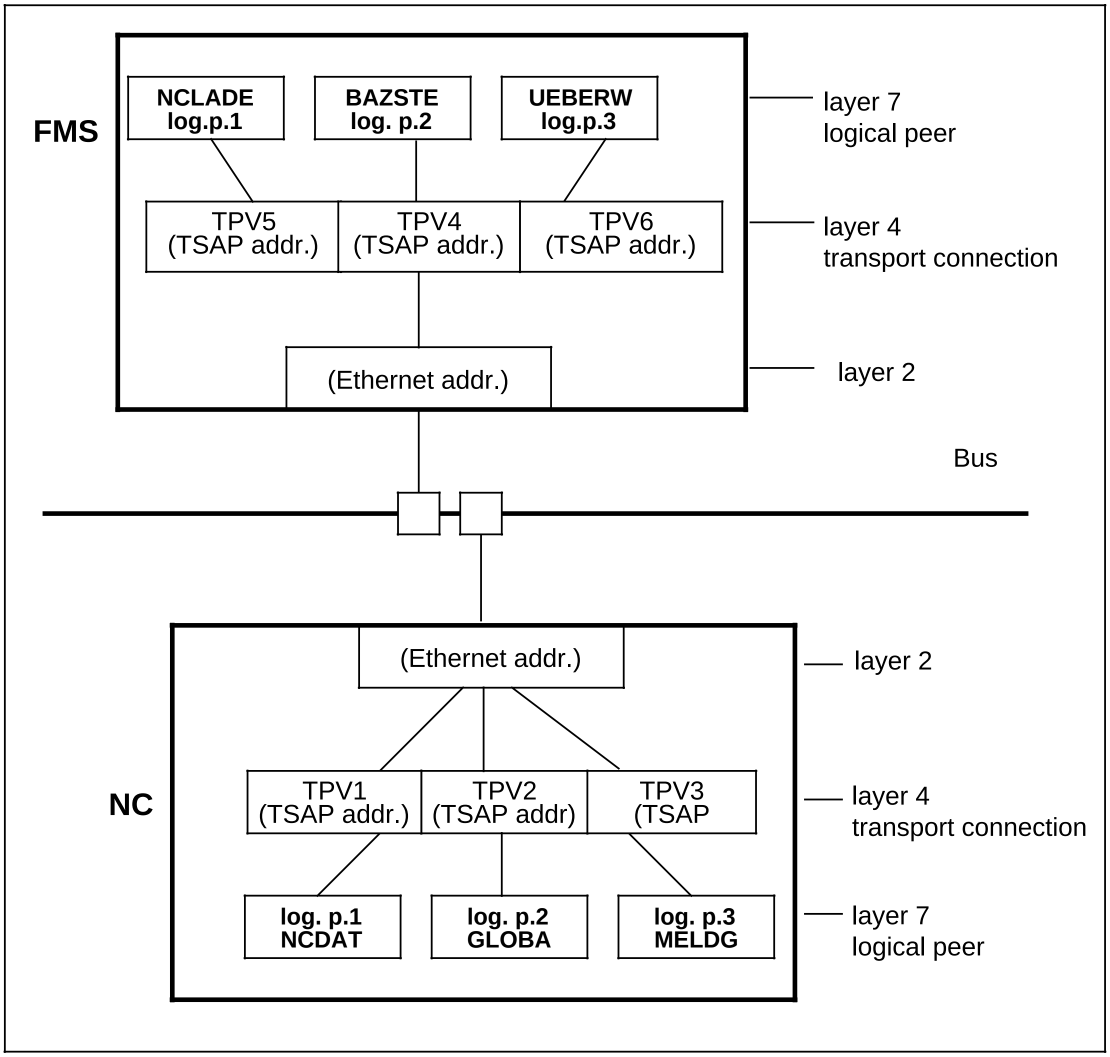
In the communication process, a distinction is made between the transmitting and the receiving logical peer.

The transmitting logical peer is the symbolic designation for the functional unit by which the message frame was sent whereas the receiving logical peer refers symbolically to the function module by which the message frame is to be received.

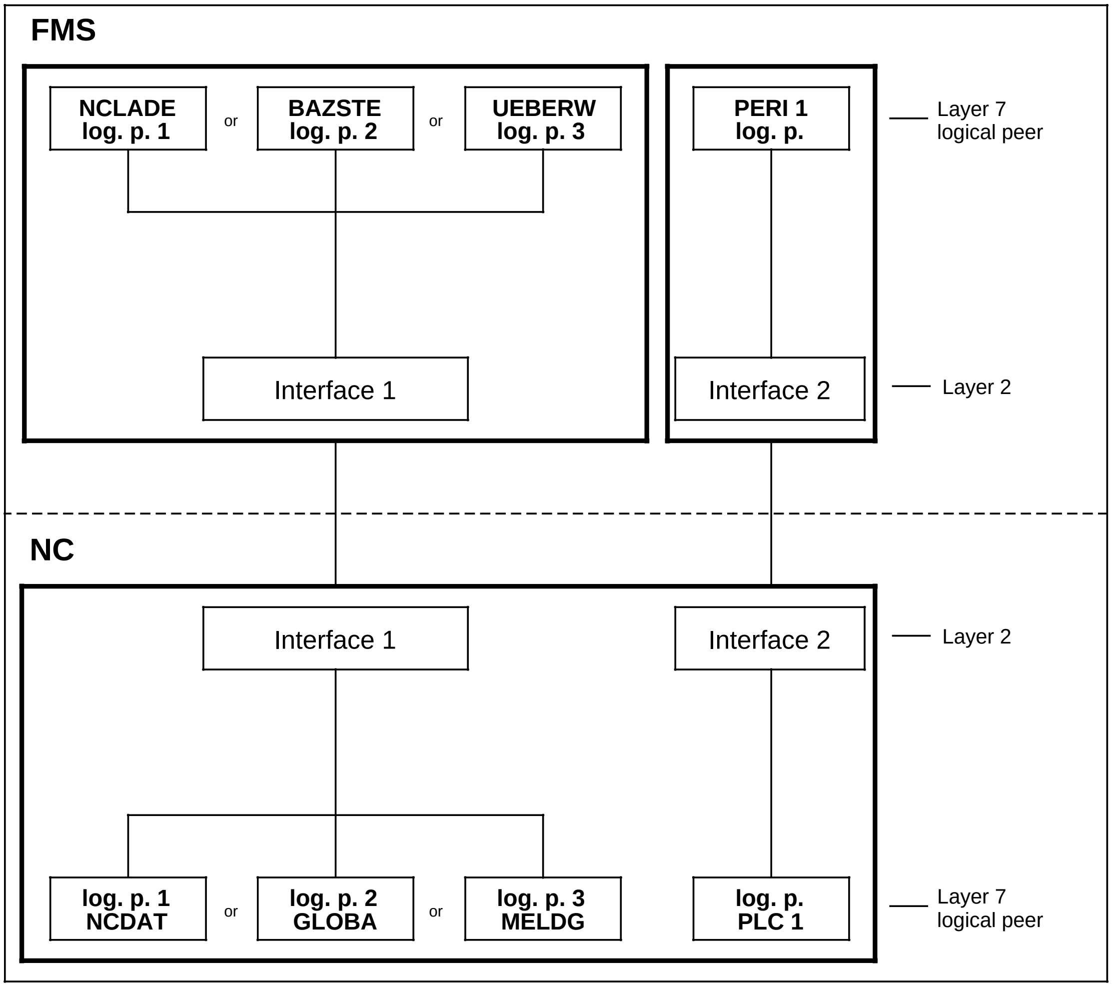
In input direction, the SINUMERIK 8X0 recognizes the sender and/or receiver by means of the address information contained in the application part of the message frames.

In output direction, the logical peer to which the message frame is to be sent is assigned to the corresponding connection.

The diagrams below give examples of logical peers both for a production control computer and for an NC, first for the bus and then for the serial connection.



Logical peer in bus connection



Logical peer in serial connection

Configuring interface 1:

- with AS 512 protocol
- with system 800 ID

Configuring interface 2: (depends on peripheral device)

- with/without AS 512 protocol
- with/without system 800 ID

For the computer (FMS):

Logical peer 1: "Functional unit load NC program,
retransmit and erase **NCLADE**"

Logical peer 2: "Functional unit manufacturing session, synchronization, end-of-work, **BAZSTE** message"

Logical peer 3: "Functional unit line monitoring **UEBERW**"

For the numerical control:

Logical peer 1: "Functional unit transfer of programs / data to the NC,
messages, initiation of transfer using **NCDAT** screen form"

Logical peer 2: "Functional unit synchronization, end-of-work, mode switchover **GLOBA**"

Logical peer 3: "Functional unit alarms, operator interrupt, status message **MELDG**"

Logical peer: **PLC1**

For the peripheral device:

Logical peer: PERI1 (any name may be selected).

2 Local Area Networks

As automation in manufacturing increases, so does the necessity to exchange data between different manufacturing systems such as controls, robots, conveyor systems and higher-level systems as for instance host computers.

In the past, information was transferred between the individual systems by people. This method is time-consuming and prone to errors.

Local area networks (LANs), interconnecting all systems involved in the manufacturing and planning process, provide a remedy for this problem. LANs ensure secure and quick data transfer within manufacturing facilities as well as to higher-level systems.

Special requirements must be met by LANs in the tough environment of industrial manufacturing:

- A universal basis for different stations (terminals)
- Insensitivity to external mechanical and electric stresses
- High availability
- Simple extendability

Various configurations are possible for local area networks. They are different in the type of connection and of data transmission.

For the computer link with SINUMERIK 8X0, bus and star connections are of special importance. These are described below in greater detail.

2.1 Bus connection

Networks with bus architectures generally use serial data transmission and often a time-division multiplex method to maintain the data flow. The bus topology has the following advantages:

- Passive transport medium, only needs a coaxial cable terminating with the correct impedance;
- Depending on the allocation procedure, the network control is concentrated at one station or distributed over all connected stations;
- Easy reconfiguration, stations can be connected or disconnected as desired;
- The failure of one station does not affect the operation of the network, provided the allocation method is suited for the purpose.

The only disadvantage of the bus topology is that it uses a common transport medium: if the cable is damaged, all nodes are affected.

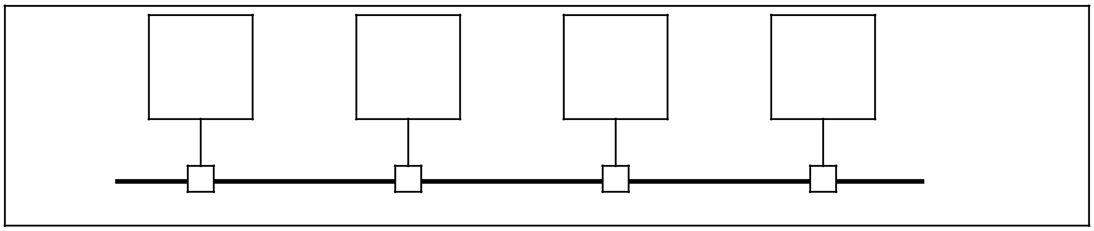
Suitable transport media for local networks are:

- twisted copper lines,
- coaxial cables,
- broad band coaxial cables
- fibre optic cables.

The bus consists of a common resource, the bus cable, and branches to each station.

Since the bus cable can be run throughout the facility, only short branch cables are necessary. The cable costs are thus reduced, and the expenditure for extending the system remains linear.

Basically, each connected station can communicate with another one over the common bus cable. However, since the bus is a common resource for all stations, its use has to be controlled by an access method in order to prevent one station from interfering with the operation of another one.



Process communication bus topology

Advantages of bus systems:

- Low cable expenses
- Cost-saving even with few nodes connected
- Extension costs are linear
- Any traffic between all units is possible simultaneously
- Bus is a common resource   access method

2.1.1 The ISO seven layer model

Various specifications have to be observed for the communication between systems. Both partners must have compatible physical interfaces, they must speak the same "language", and they must "mean" the same thing when they say something.

To ensure this between the peers on a local area network, ISO (International Standardisation Organization) has defined a hierarchical layer model.

In this model, a layer is defined as a program or a process that communicates with the corresponding process of its peer on the same layer .

The layers function as follows:

A layer receives a job from the next higher layer. It processes the job with its own resources and then passes a job to the next lower level. This means that to handle the job the layer uses its own resources as well as the services made available by the next lower layer.

Example: 3 layer model

A "humanized" example of a 3 layer model may help to explain the layer structure:

Two scientists would like to exchange their findings. Scientist 1 lives in Japan and speaks Japanese, scientist 2 lives in Spain and speaks Spanish. The two scientists represent layer 3.

In order to exchange information, each of them hires a translator. To understand each other, the translators must agree on a common language, e.g. English. The translators constitute layer 2.

In order to transmit the information, the translators deliver their texts to the respective post office.

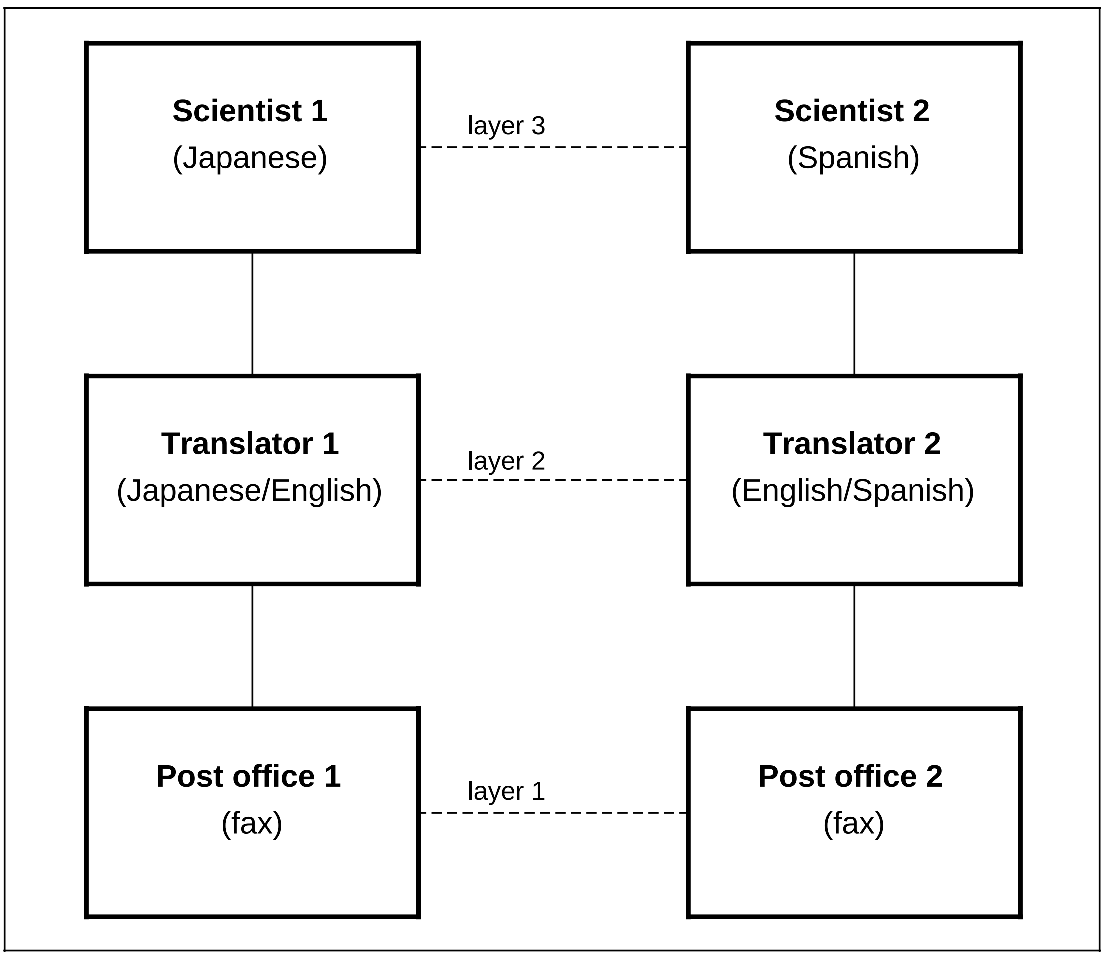
To send the texts, the employees of the post offices agree on a common transmission method (e.g. telex or fax). They constitute layer 1.

The procedure is as follows:

Scientist 1 (layer 3) hands the message written in Japanese to his translator (layer 2). The translator translates the text into English and delivers it to the post office (layer 1). The post office in Japan transmits the text by fax to the respective post office in Spain. The translator in Spain (layer 2) receives the fax from the post office employee and translates it from English into Spanish. Scientist 2 (layer 3) receives a Spanish text from his translator which contains the thoughts of his Japanese colleague.

Irrespective of the common language the translators have chosen (e.g. French instead of English) or the type of transmission selected by the post (e.g. telex instead of fax) the scientists receive the message in a language they understand.

This example shows that individual layers can change their agreements (protocols) without affecting the other layers. This is the main reason for using the layer structure for networks.



3 layer model

	Layer	Designation	Function	Example
Appli-cation-oriented layer	**7**	Application Layer	Application functions (Information processing)	• File transfer • Remote Control
	6	Presentation Layer	Data representation language	• Common
	5	Session Layer	Communication control Synchronization	• Coordination (open, end, abort)
CP / Transport-oriented layer	**4**	Transport Layer	Transport control Flow control Blocking Acknowledgement Connection establishment/release	• Transmission of conversation text
	3	Network Layer	Network connections Addressing of other networks	X.25
	2	Data Link Layer	Reliable transfer	CRC check
			Allocation method	CSMA/CD (**IEEE* 802.3**)
	1	Physical Layer	Physical characteristics of the transmission path	Yellow Cable (**IEEE* 802.3**)

*** Ethernet**

Protocol architecture for local area networks according to ISO

ISO layers

The layer model defined by ISO includes 7 layers:

The transport-oriented layers (1 to 4) and the user-oriented layers (5 to 7). Each of these 7 layers provides services. In order to make its services available, a layer of a network entity must communicate with the same layer of its peer. This communication process, including the means used for it, is referred to as a protocol.

Layer 1: Physical Layer = Physical characteristics of the transmission path, e.g. Yellow Cable

This layer defines the physical operation such as electrical (current-voltage level) and mechanical parameters between 2 peers, in order to be able to transmit data.

It provides services in order to activate, maintain and deactivate physical connections for transmission of data between two data-link entities.
"A medium must be available, for example air, telephone line etc.".

Layer 2: Data Link Layer = 2a Access method, such as CSMA/CD 2b Reliable transfer, e.g CRC check

On this level, the access mechanism is defined that ensures that only one node sends data over the bus at a given time. The layer provides functional and procedural services in order to establish, maintain and release data link connections. It detects and possibly corrects errors which may occur in the subordinate physical layer. It enables the network layer to control the interconnection of data circuits within the physical layer, e.g. to interconnect partial sections and to switch to alternate paths using different combinations.

Layer 1 and 2 ensure that information can be transmitted and that access to the transmission medium is controlled.

Layer 3: Network Layer = Network connections/addressing of other networks

The network layer provides the means to establish, maintain and terminate connections between systems as well as networks.

"Rules for routing must be specified when using several transmission sections, e.g. route switching of several sections in a communication network".

Layer 4: Transport Layer = Transport control: Flow control, blocking, acknowledgement, establishment and release of connection, transmission of the conversation text.

The function of the transport layer is to provide a universal transport service in cooperation with the services of the lower-level layers.

It separates the application-oriented from the transmission-oriented layers.

The transport layer ensures data security and data consistency. It manages the transport control requirements and provides the following services:

- Flow control:
  A message frame is sent only if the remote station is able to receive it, e.g. if sufficient receive buffer is available.
- Blocking:
  Message frames longer than e.g. 512 bytes are blocked automatically so that long message frames do not block the line.
- Acknowledgement:
  Acknowledgement of a message frame received without error by the transport layer (no user acknowledgement).

To implement these functions transport connections (logical channels) are established. Establishment/release of these connections are effected automatically by the services of layer 4.

Layer 5: Session Layer = Communication control/synchronization e.g. coordination of the conversation (opening, end, abort)

The session layer provides means necessary for cooperating entities of the higher-level presentation layer (layer 6) to organize and synchronize their dialogue and manage their data exchange. For this purpose the session layer provides services to establish a session connection between two entities of the higher-level presentation layer and to support actions required for a proper data exchange.

The session layer can be designed with more or less comfortable functions.

There are only user-specific aspects for these and the higher layers.

Layer 6: Presentation Layer = Data representation e.g. common language

The Presentation Layer is concerned with the representation

- of the data which are transmitted between corresponding entities of the higher-level application layer. This data representation is also called data syntax.
- of the data structure to which entities of the application layer refer in order to perform actions.

The presentation layer is concerned only with the representation (syntax), but not with its meaning (semantics). The semantics is only known to the application layer.

"Receiver and transmitter must have the information represented in a form known to everyone (common language). If the representation forms differ (the transmitter speaks Japanese, the receiver speaks Spanish), a translator is required, e.g. an interpreter for Japanese and Spanish. However, an intermediate language, for example English, might be chosen, and both the transmitter and the receiver employ an interpreter; the transmitter for Japanese-English and the receiver for English-Spanish".

Layer 7: Application Layer = Application function (information processing) e.g. File Transfer, Remote Job Control

The application layer is the highest of the 7 layers of the OSI model and does not provide any services to a higher layer with regard to the integration of open systems (OSI environment). It rather provides the means for the application processes, so that these can access the OSI environment. The application layer serves as a window between corresponding application processes which are using the integration of (open) systems to exchange meaningful information.

Each application process is represented to its peer by the application layer entity.

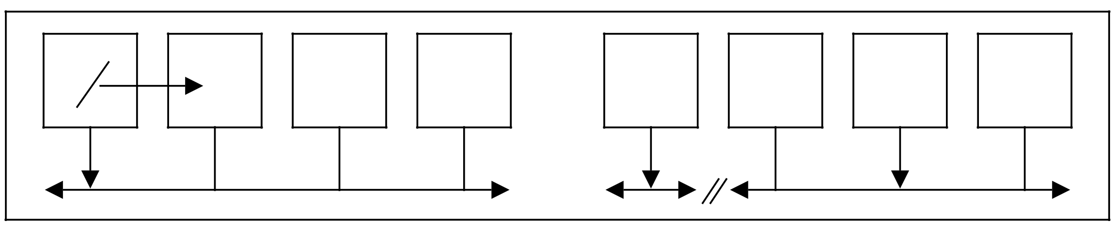
"Even when the same language is spoken and the same terms are used, a term can have different meanings, when for example the same terms are used in a different environment. It must therefore be ensured that the meaning of one or a combination of terms is identical for both the transmitter and the receiver".

2.1.2 Access methods

Two access methods are today at the centre of both theory and practice:

The token principle and the collision principle (CSMA/CD, Ethernet). There are already several variants of both procedures, however one variant of each has been standardized.

Process communication/access methods



Allocation systems/
token procedures

- An authorization to send (token) exists, which is held by one station onlyat any time.
- The authorization to transmit is passed on to another station after a given period of time.
- Each station receives one authorization to transmit after a defined time interval.

Contention systems/
collision principle
(CSMA/CD, Ethernet)

- Theoretically, all stations in a network can transmit simultaneously.
- In case of a collision, this is recognized and the stations repeat transmission after different times.
- High transmission capacity keeps the number of collisions low
- A probability factor can be specified with which a message will be transmitted.

2.1.3 SINEC H1

SINEC (Siemens Network Architecture) stands for the integrated communication system of the Siemens automation systems.

This includes:

- The communication network
- The respective module which establishes the connection to the communication network (interface module)
- Protocols and services used for transmission of data between the systems.

SINEC H1 is a baseband transmission communication system to IEEE 802.3 (Ethernet).

It is based on the reference model of the international standard committee ISO for open communication aiming at operating different types of devices together.

The SINEC H1 layers 1 and 2a (transmission procedure and network access) are based on the IEEE 802.3 standard. These layers are concerned with data transmission and the access of the nodes to the network. The access method is based on the collision principle used by Ethernet. SINEC H1 provides for the higher layers a standardardized interface which is independent of the access method.

Layer 3 of the ISO reference model is required for large networks. This layer is inactive with SINEC H1.

For layer 4, SINEC H1 has implemented the ISO transport protocol (ISO-8073 class 4).

The functions of layers 1 to 4 are implemented by SINEC H1 on communication processors which are located on the interface module. They offer the user a series of services which relieve him/her from work-intensive procedures.

At present there are no uniform international standards for layers 5 to 7 of the OSI model for the NC applications. As a first step to standardization of OSI layers 5 to 7 within the company, Siemens have therefore defined the AP 1 automation protocol. It is suitable for use on all Siemens automation devices.

For more details on the SINEC AP 1 protocol please refer to Section 3.4, Protocols.

The most important features of SINEC H1 are:

- Topology: bus (line) consisting of one or several segments
- Transmission medium: shielded coaxial cable; 50 ohms
- Segment length: max. 500 metres
- Connectable nodes per segment: max. 100
- Transmission method: baseband
- Data transfer rate: 10 Mbits/sec.
- Bus access method: collision principle

2.2 Serial connection

In contrast to bus connection, with which all nodes are physically connected in parallel to a common bus cable, every connection between two peers is implemented with a separate connection cable in the case of serial connection.

Procedures agreed between the two nodes ensure the controlled data exchange between the connected peers.

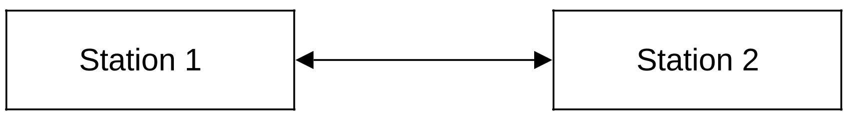
2.2.1 Point-to-point connection

The point-to-point connection ensures the communication between two peers. This is the simplest form of process communication if it is established as direct connection (separate line) between two nodes.

Complex communication relations can be logically traced back (to a large extent) to a combination of these simplest relationships.

If exactly this case occurs in practice, even the simplest type of connection is adequate.

Process communication point-to-point connection



- Data transfer between two peers
- Simplest case of process communication
- Simplest type of connection is adequate (4-wire)
- Example: PLC/PG traffic on SIMATIC systems

Exactly one physical point-to-point connection corresponds to each connection i.e. each existing request of a unit to exchange data with a partner station.

Depending on the number of nodes involved, cabling expenditure is high and costs will increase progressively if a further station is to be connected and wishes to communicate with all other stations. Larger networks cannot be implemented due to the required number of interfaces per unit increasing considerably.

Fault diagnosis is relatively simple since an unambiguous allocation is possible according to communication request and physical implementation.

2.2.2 Star connection

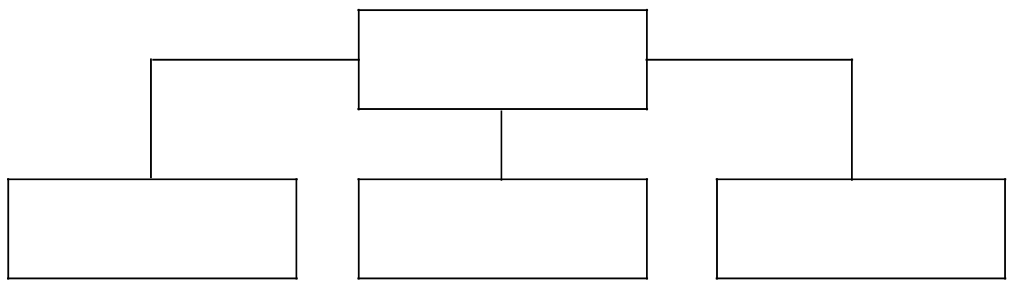
The star topology is today probably the most widely used type of connection in a hierarchically organized process.

Although the cabling expenditure is lower than with meshed networks, it is still relatively high, as in most cases the distances between master and the lower-level controls are rather long.

The master station must have one interface for every node; with larger systems, the masters themselves can form a hierarchical structure.

The internode communication between two lower-level systems has to be executed over the master which results in an additional burden and an availability bottleneck.

Process communication star topology



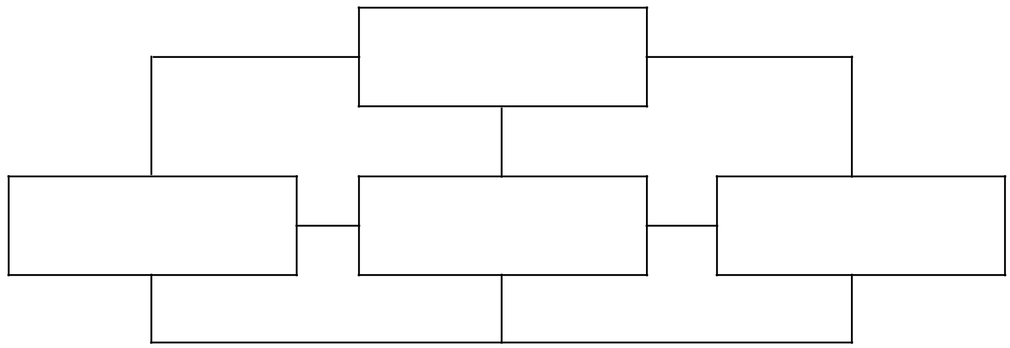
- Corresponds to the hierarchical organization of the process
- Linear extension cost
- Only at the master station are many interfaces necessary
- Internode communication is executed via the master node
- Master station is availability bottleneck
- Simple diagnostics
- Simple to configure
- Different transmission media can be used

2.2.3 Meshed network

Meshed networks have the advantage that the individual stations are connected with several nodes via point-to-point connections.

Internode communication is possible, due to which the availability increases, since - in contrast to the star network - the master does not cause an availability bottleneck.

Process communication meshed network



- 1 to 1 allocation of logical connections to physical communication channels
- High cable cost
- Extension costs highly progressive
- Very large number of interfaces required per unit
- Simple diagnostics
- Failsafe
- Internode communication is possible

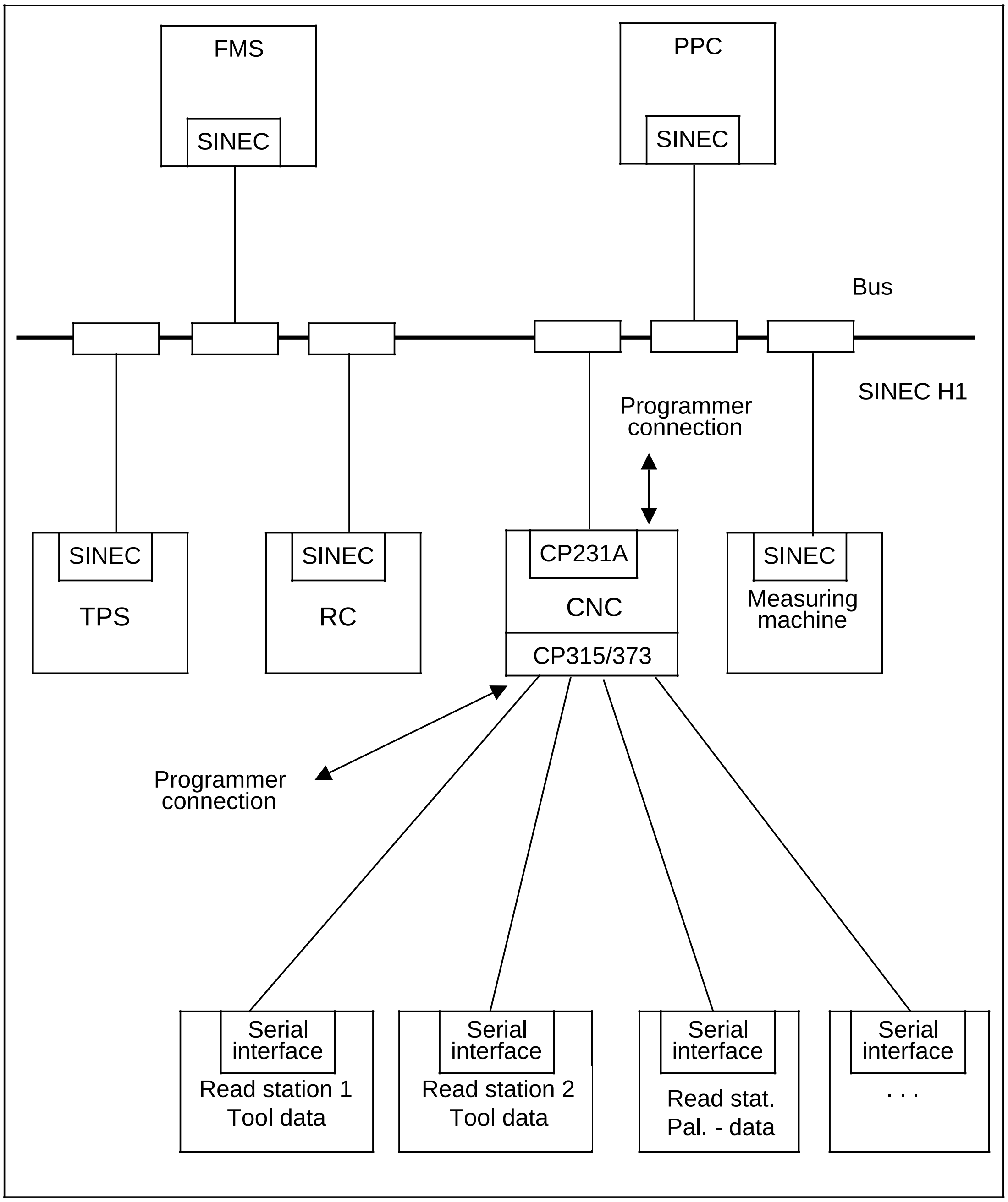
2.3 Bus and serial connections

In addition to local area networks featuring either only bus connections or only serial connections, there are mixed forms of networks.

The following figure shows a network with different connection types, using the example of a NC system. In this example the CNC is the master station of a star network and at the same time a node on the bus network.

The CNC is connected to the higher-level host systems (e.g. host computer) and the systems at the same level (e.g. RC controls) via bus connection.

Data transfer between the CNC and the secondary systems (e.g. read stations) is via serial connections.



3 Interface Modules

For the computer link with SINUMERIK 8X0, there are corresponding interface modules which are plugged in the COM area.

The modules are the interface between the local area networks or point-to-point connections and the control system. The services at the physical layer through to the application layer such as procedures, test methods, access mechanisms etc. required for communication are implemented on the interface modules.

The connection-specific and application-oriented parameters are stored on the interface modules in lists which are compiled by means of a configuring software executable on the PG 685.

The following interface modules are available:

- Bus interface module (CP 231A)
  The bus interface module is referred to as CP 231A. It allows connection to the SINEC H1 bus.
- Basic board for serial connection (CP 315)
  The basic board for serial connections provides a serial interface conforming with RS 232C/TTY. The connector pin assignment determines whether the serial interface is used as RS 232C or TTY interface. The board is called CP 315.
- Expansion board for serial connection (CP 373)
  To complement the CP 315 basic board there is an expansion board with the designation CP 373. It is mounted on the side of the basic board and can be operated only together with it. The CP 373 offers three further serial interfaces.

Dependent on the frame variant or the SINUMERIK 880 configuration, various configurations are possible for the interface modules.

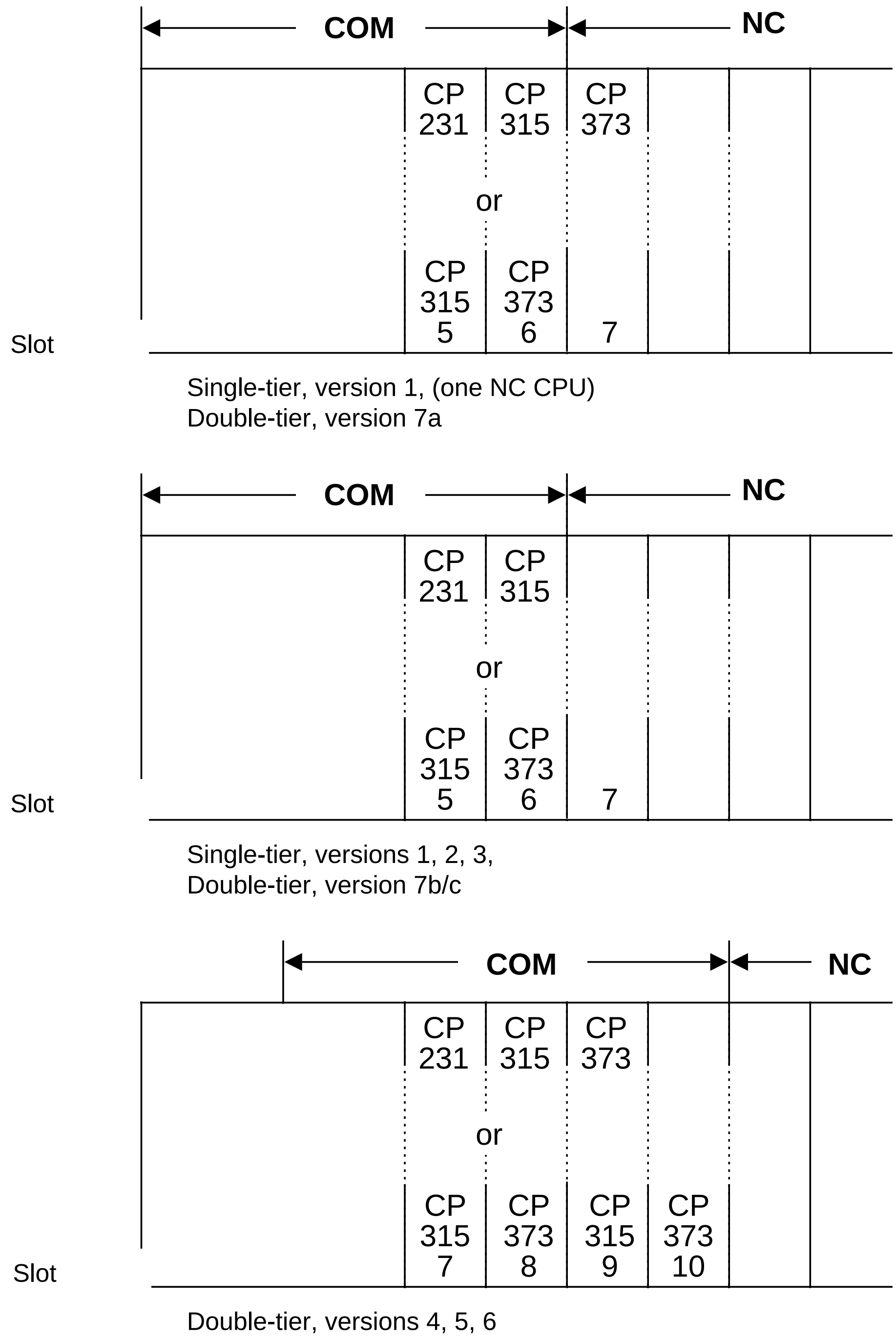
The following figure and table show examples of typical variants.

At application layer, the addressing concept of the computer link message frames is identical for both bus and serial interface modules. It is implemented on the interface module by means of SINUMERIK-specific input and output lists. In the case of a serial interface module, message frames not corresponding to the SINUMERIK specifications can also be processed. The addressing is carried out via internal lists.

Message frames can be transmitted between the bus interface and the serial interface module, i.e. the message frames are not processed in the SINUMERIK but are passed from one interface module to the other.

Rack type	Max. complement with interface modules	Number of serial interfaces	Number of bus interfaces
Single-tier Version 1 (with one NC CPU) **Double-tier** Version 7a	1 CP 231 1 CP 315 1 CP 373	 1 3	1
Single-tier Version 1 (with one NC CPU) **Double-tier** Version 7a	1 CP 315 1 CP 315 1 CP 373	1 1 3	–
Single-tier Version 1 Version 2 Version 3 **Double-tier** Version 7b Version 7c	1 CP 231 1 CP 315	 1	1
Single-tier Version 1 Version 2 Version 3 **Double-tier** Version 7b Version 7c	1 CP 315 1 CP 373	1 3	–
Double-tier Version 4 Version 5 Version 6	1 CP 231 1 CP 315 1 CP 373	 1 3	1
Double-tier Version 4 Version 5 Version 6	1 CP 315 1 CP 373 1 CP 315 1 CP 373	1 3 1 3	–

Examples of complements with interface modules for SINUMERIK 880

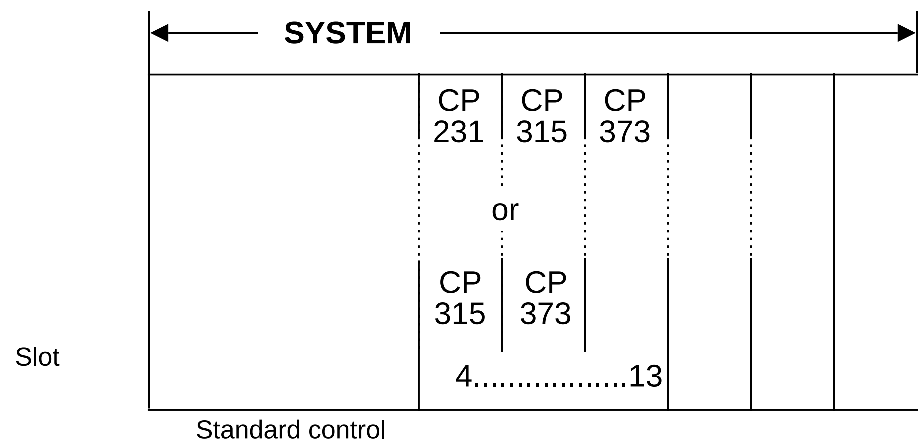


CP 231: bus interface module
CP 315: serial interface module (basic board, 1 interface)
CP 373: serial interface module (expansion board, 3 interfaces)

SINUMERIK 800 rack configuration

Rack type	Max. complement with interface modules	Number of serial interfaces	Number of bus interfaces
	1 CP 231 1 CP 315 1 CP 373	 1 3	1

Examples of complements with interface modules for SINUMERIK 840



CP 231: Bus connection
CP 315: Serial connection (basic board, 1 SS)
CP 373: Serial connection (expansion board, 3 SS)

Rack configuration for SINUMERIK 840

3.1 Bus interface module

The SINEC H1 interface module CP231A is used to link the SINUMERIK 8X0 to a SINEC H1 bus system.

The figure "SINEC H1 interface module" shows the front view of the interface module with the EPROM submodule, the programmer interface and the bus interface.

The firmware required for the interface module is stored on the EPROMs.

Testing and configuring functions can be executed locally with the PG 685 or PC AT via the programmer interface.

The programmer interface conforms to the RS 232C/TTY standard, i.e. it can be used as RS 232C interface or TTY interface, as required. Its use is determined by the assignment of pins in the connector of the connecting cable.

The hardware interface to the SINEC H1 bus is a 15-way subminiature connector which accommodates the transceiver cable.

Data transmission between the interface module and the SINEC H1 bus is executed with a transmission speed of 10 Mbit/sec.

The CP 231A module uses the ISO seven layer model. An outline configuration of the layer model on the CP 231 is shown in figure "CP 231: interface module configuration".

Layer 1 to 2a consist of the physical components of the LAN and parts of the interface module hardware.

Roughly, these are: the SINEC H1 bus cable, the transceivers and the interface module hardware for the bus access method, in this case collision monitoring.

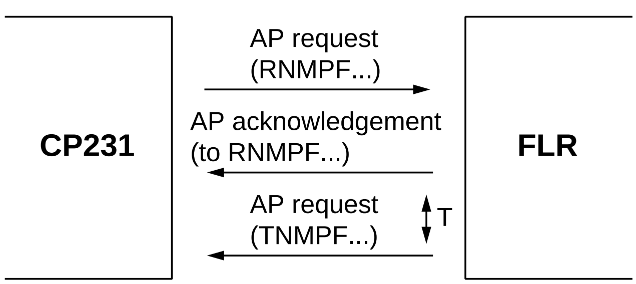
Layer 2b through 4 are implemented using a real-time operating system and a network software.

The essential services are: addressing, blocking, error detection and flow control.

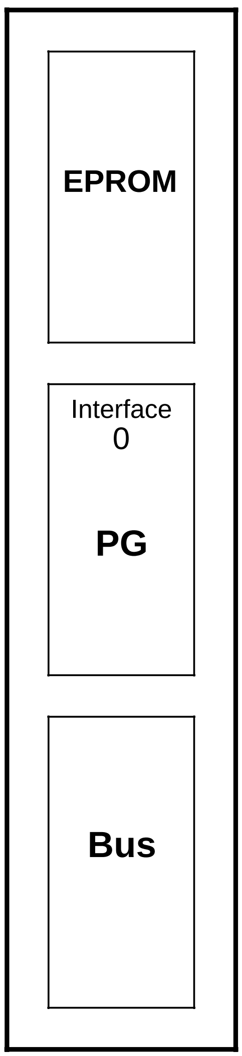
The SINEC AP 1.0 software package uses the network software interface and covers layer 5 through 7.

Note:

To ensure error-free data transfer between communications processor CP 231A and host computer, the minimum time delay between two message frames, i.e. between AP acknowledgement and AP request, should be approx. 30 - 50 ms.



CP 231 A



Technical data

1. Interfaces

   1 x PG interface (RS 232/TTY)

   (25-way connector)

   1 x bus interface (SINEC H1)

   (15-way connector)

2. Transmission rate

   10 Mbit / sec. (SINEC H1)

   9600 bits / sec. (programmer interface)

3. Access method

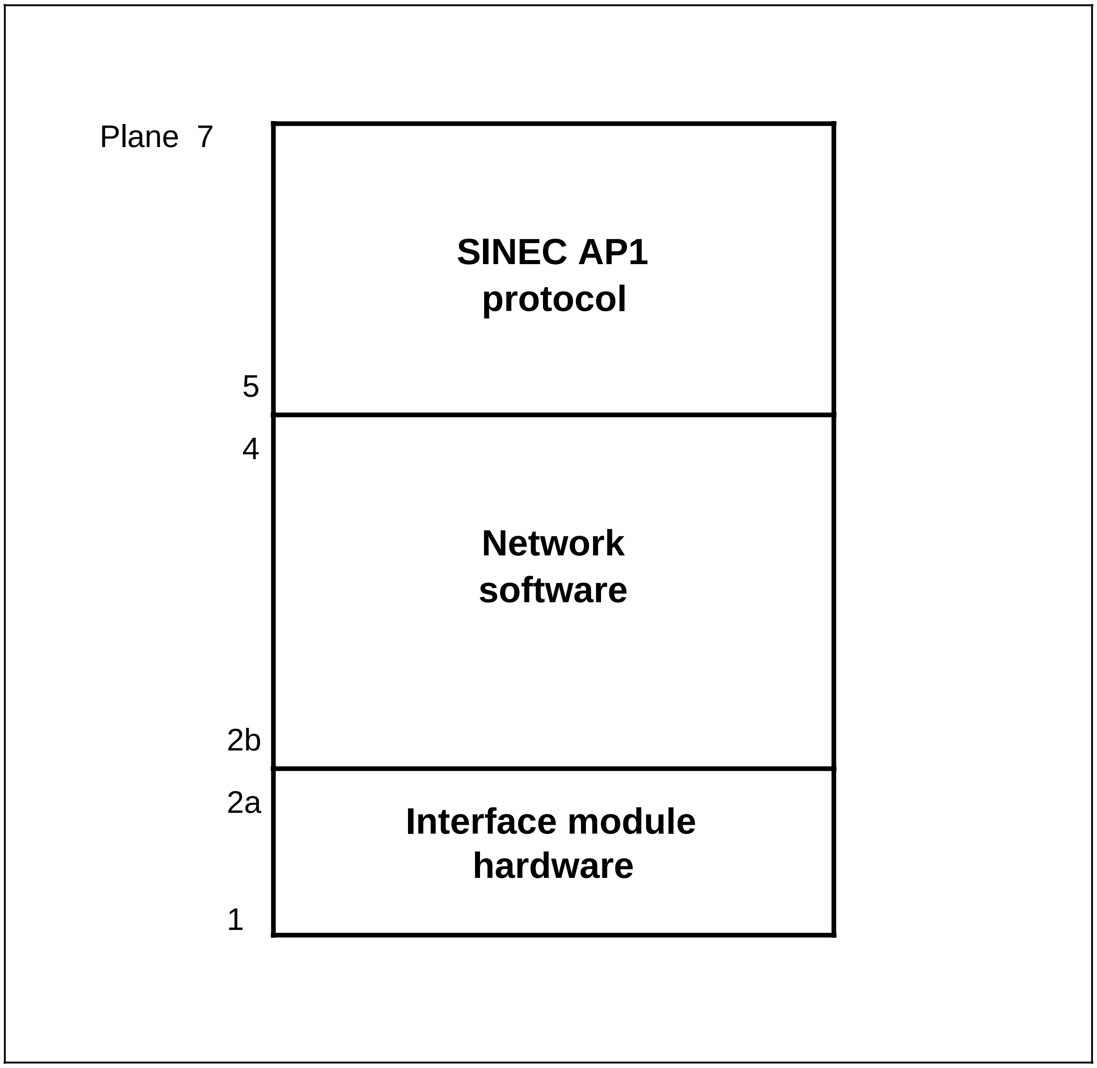
   CSMA/CD (IEEE 802.3)

4. Protocols

   ISO-8073 class 4

   SINEC AP 1.0

SINEC H1 interface module



CP 231: interface module configuration

3.2 Serial interface modules

The active serial interface module CP315 is used for point-to-point connections to computers (FMS) as well as to tool or pallet stations, measuring machines, tool coding systems, transport controls and other systems.

It consists of a basic board with EPROM submodule, a programmer interface and an unassigned interface.

An expansion board which is referred to as CP 373 is available for the basic board. The expansion board offers three additional serial interfaces. It is mounted on the side of the basic board and can only be operated together with the basic board.

The front view of the two boards is shown in the figure "Active serial interface".

The firmware for the basic board and the expansion board is stored on the EPROM submodule of the basic board.

The programmer interface is used for connecting the PG 685 or PC AT, or for transmitting the configuring data generated on the PG 685 or PC AT for CP 315/373. Furthermore, test functions can be activated via the programmer interface with the aid of the PG 685 or PC AT.

The hardware configuration of the PG interface and the serial interfaces for the computer link are compatible with the RS 232C/TTY, i.e. they can be used as RS 232C or TTY, depending on the connector assignment of the connecting cable.

Each of the offered serial interfaces can also be operated as RS 422 when using an external adapter.

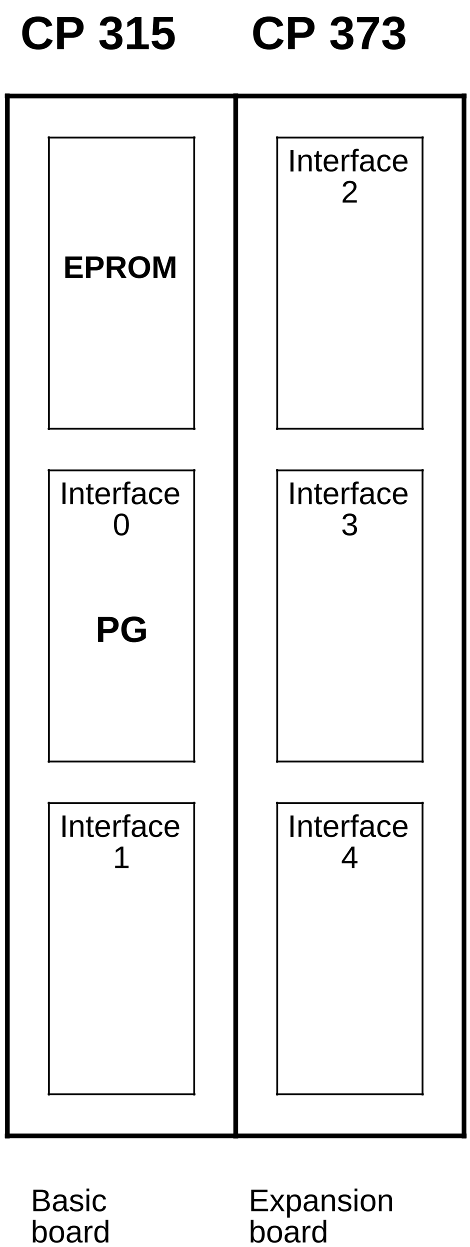
The cumulative data rate of the offered serial interfaces is 19200 bauds.

Procedures 3964R, LSV 2, XON/XOFF, RTS-LINE and TRANSPARENT are offered as a standard for the serial interfaces which are to be used for the connection to remote stations. Which procedure is to be used for the specific interface, is determined when configuring the interface module.

The figure "Configuration of the active serial interface modules" shows the outline configuration of the software modules contained on the interface module.

The link between procedure driver and SINUMERIK driver is the device management which in turn must supply or interpret the AS 512 header, if configured.

For configuring and test purposes, independent software modules are provided on the CP.



Technical data

1. Basic board:

   1 x PG interface (RS 232/TTY)

   1 x RS232/TTY

   1 x EPROM submodule with firmware

2. Expansion board:

   3 x RS 232/TTY

   RS 422 via adapter

3. Procedures:

   a) 3964 R

   b) LSV 2

   c) XON/XOFF

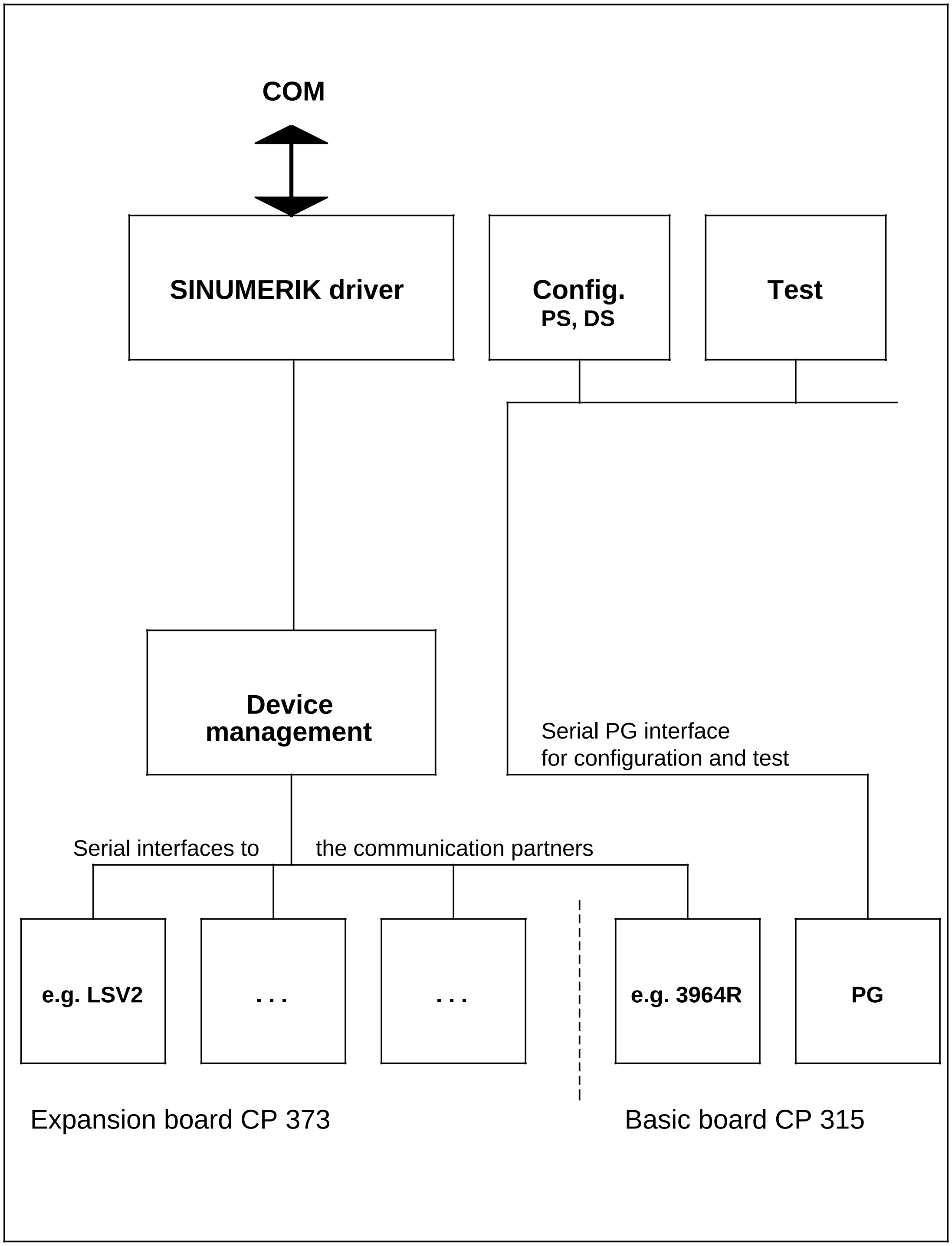
   d) RTS-LINE

   e) TRANSPARENT

4. Cumulative data rate

   19200 baud

Active serial interface modules



Configuration of the active serial interface modules

3.2.1 Communication with the CP 315

When using the CP 315/373 interface modules, the user can select from a choice of eleven combinations of procedure, protocol and message frame specification for interface assignment. The protocol and message frame specifications in the column "* with AS 512 * SINUMERIK 850/880 message frames" is compatible with existing configurations of SINUMERIK 850 systems. The possible combinations indicated in the remaining columns were added subsequently and are referred to as "flexible communication".

The following figure shows the assignment of procedures to protocols or message frame specifications:

Protocol or message frame specification / Procedure	* with AS 512 * SINUMERIK 850/880-message frames	Flexible communication		
		* w/o AS 512 * SINUMERIK 850/880 mess. frames	* with AS 512 * Free message frames	* w/o AS 512 * Free message frames
LSV2	Possible	Possible	Possible	Possible
3964 R	Possible	Possible	Possible	Possible
XON/XOFF	–	–	–	Possible
RTS-LINE	–	–	–	Possible
TRANSPARENT	–	–	–	Possible

Allocation of procedure to protocol or message frame specification

Brief description of the message frame specification:

- SINUMERIK 850/880 message frames are message frames with System 800 identification.
- Free message frames are message frames without System 800 identification.

The message frames are described in more detail in Section 6.
Brief description of the procedures:

- LSV2 is an asynchronous synchronous data transmission procedure (start-stop procedure) for point-to-point connections (DIN 66019).
- The 3964R procedure is an asynchronous, bit-serial transmission method.
- The XON/XOFF procedure is a a character-oriented transmission method which is controlled by the characters XON and XOFF.
- RTS-LINE is a line-controlled transmission procedure. The RTS and CTS signal lines replace the control characters XON and XOFF.
- The TRANSPARENT procedure does not restrict the user to any guidelines. Only one control line can be connected.

The procedures can be configured independently of each other for each activated interface.
For a detailed description of the procedures see Section 3.3.

3.2.2 Requirements for the use of "flexible communication"

- Hardware requirements:
  CP 315/373, all versions.
- Software requirements:
  CP 315/373 from firmware release 2A onwards (Order No. 6FX1841-0BX01-2A)
  SINPS 315 from release 7.0 onwards.

"Flexible communication" has an extended scope of functions compared to the existing interface module software and forms an integral part of the latter from the firmware releases stated.

3.2.3 Brief description of functions

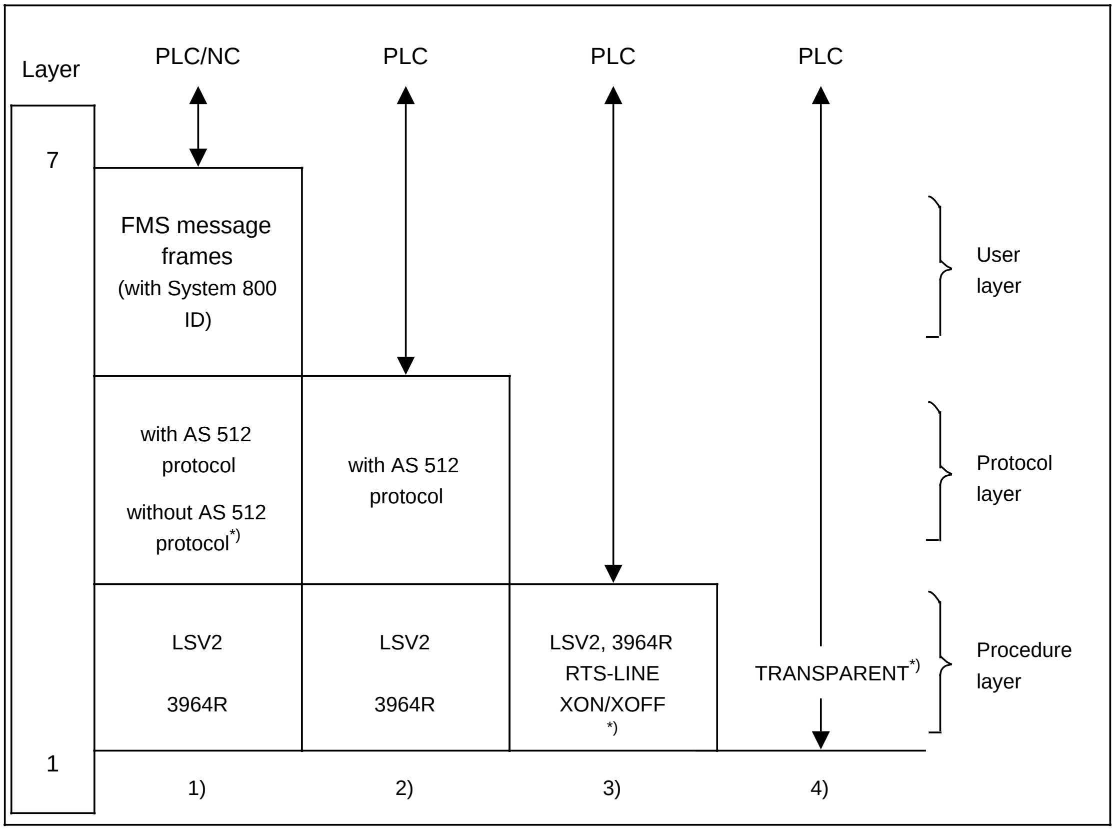
"Flexible communication" enables the user to assign interfaces of the communications processor (CP 315) to open data communication. The use of these interfaces is **not** restricted to the AS 512 protocol and the SINUMERIK 850/880 message frame specification (i.e. with System 800 identification). As a result, any peripheral units, i.e. operating with/without the AS 512 protocol or with/without FMS message frame specification, can be connected. In addition to LSV2 and 3964R, the procedures XON/XOFF, RTS-LINE and TRANSPARENT can also be configured on the procedure level. This ensures a high degree of freedom of choice with regard to communications between the SINUMERIK controls and the peripheral units connected.

There are four possible combinations of protocols and message frame specifications:

- SINUMERIK 850/880 message frames
  - with AS 512 protocol
  - without AS 512 protocol
- Free message frames
  - with AS 512 protocol
  - without AS 512 protocol

The user interface (DB 101/DB 102) in the PLC forms the internal interface for "flexible communication".

3.2.4 Possible communication links within the framework of the ISO 7-layer model



The above drawing shows the possible communication links within the framework of the ISO 7-layer model.

The user can select individual layers of the ISO reference model for establishing a communication link. Additionally, the user can utilize the permanently implemented user, protocol or procedure layers, individual ISO layers or change these layers for user-specific definitions.

In column 1), the user is tied to the SINUMERIK 850/880 message frame specification. Message frames can be used with or without AS 512 protocol. On the procedure level, the procedures LSV2 and 3964R are available to the user.

Column 2) is independent of the SINUMERIK 850/880 message frame specification; the AS 512 protocol is however obligatory. The LSV2 or 3964R procedure must be selected just as in column 1.

In column 3), the user can transfer any message frames, irrespective of the SINUMERIK 850/880 message frame specification and the AS 512 protocol. On the procedure level, the user can choose among four procedures.

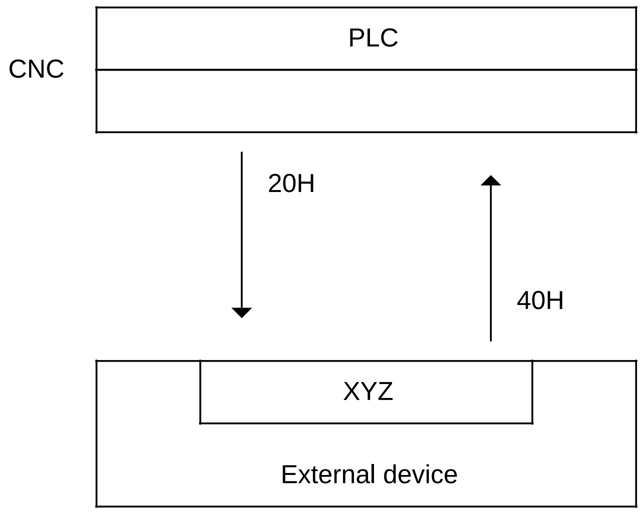
In column 4), the user is, just as in column 3), neither restricted to the SINUMERIK 850/880 message frame specification nor to the AS 512 protocol. The TRANSPARENT procedure involves no limitations, not even on the procedure level. Message frame transfer is completely **transparent,** i.e. the message frames remain unchanged while being passed on.

*) *Extended scope of functions: "Flexible communication"*

Note for the user:

The more implemented layers are replaced by user-specific definitions the more provisions the user has to make for proper message frame transfer.

The following diagram shows a possible application for variable configuring of message frames:



An external device uses the imaginary procedure XYZ. This procedure waits for the 20H control character to be transmitted; it acknowledges reception of this character with the 40H character.

When using this procedure for data transfer, the user must therefore proceed as follows:

1) The user sends a message frame with only one usable data - namely, 20H - which is neither embedded in the SINUMERIK 850/880 message frame specification nor in the AS 512 protocol. For this purpose, the user selects the TRANSPARENT procedure to be able to transfer this data in a completely transparent manner (i.e. unmodified) to the external device.

2) The external device recognizes the 20H character as a control character and sends the 40H character as acknowledgement.

3) The user receives the 40H character, interprets it as a control character and can now start the data exchange. The user has thus adapted to the guidelines of the XYZ procedure and created an identical procedure on his/her side.

The disadvantage of such an application is the relatively large amount of time involved. The user must send a separate message frame for every control character. In the given example, two message frames (20H and 40H) must be sent in addition to the actual message traffic.

3.3 Procedures

3.3.1 3964R procedure

The 3964R procedure processes all operations necessary to control the data flow between the CP 315 interface module and the connected node (e.g. computer).

The 3964R procedure is an asynchronous, bit-serial transmission method. Transmit and receive clock (baud rate) must be set to the same value on the CP 315 interface module and in the connected peripheral unit since no clock cable is provided between the two devices (asynchronous). Control characters and useful information characters are transmitted through the junctions.

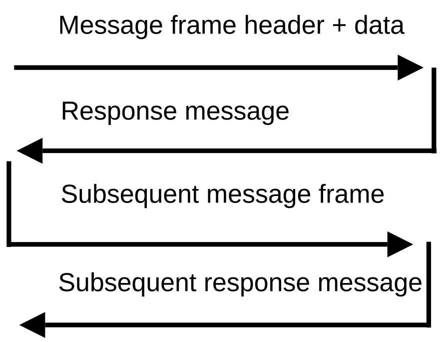
When addressing an interface of the CP 315, to which a computer link section is connected, the computer link program modules in the CP 315 are activated.

Two or more units of an identical or different type can be connected with each other. Each interface of the basic board and of the expansion board can be assigned to one computer link channel.

A message frame sequence is shown below in a global manner between two nodes which are connected with each other over a point-to-point connection. A more detailed description is given on the following pages.

- Output commands - data from node I to node II

  Node I Node II



In the case of large data volumes which cannot be transported in one message frame a subsequent message frame is sent.

The block length of the message frame is configurable. Block lengths between 128 and 256 characters are possible.

Transmitting using the 3964R procedure

(CP 315 acting as transmitter = active CP 315)

To establish the connection, the 3964R procedure sends the STX control character (request to send). If the peripheral unit responds within the timeout period with the DLE character, the procedure switches to the transmit mode. If the peripheral unit responds with NAK, any other character (except DLE) or a mutilated character, or if the timeout period elapses without a response, the connection establishment attempt has failed. The driver then attempts again to establish the connection. After failed repeated attempts, the number of which is configurable, the driver aborts the process and enters an error message in the exception trace.

If connection establishment is successful, the useful information contained in the output buffer are sent to the peripheral unit at the selected data transmission rate. The peripheral unit monitors the intervals in which the incoming characters arrive.The interval between two characters must not be longer than the character delay period (ZVZ).

Every DLE character retrieved from the buffer is sent as two DLE characters.

After the buffer content has been transmitted, the transmit driver adds the characters DLE, ETX (end-of-text) and BCC (block check character) to specify end-of-text and waits for an acknowledgement character.

The data block has been received without error if the peripheral unit transmits the DLE character (acknowledgement) within the timeout period. If the peripheral unit responds with NAK, any other character (except DLE) or a mutilated character, or if the acknowledgement delay (timeout) elapses without response, the driver repeats transmission of the complete data block. With the CP 315, the number of transmission attempts can be configured. After several failed attempts to send the data block, the procedure aborts the process and the driver sends the identification character "procedure error".

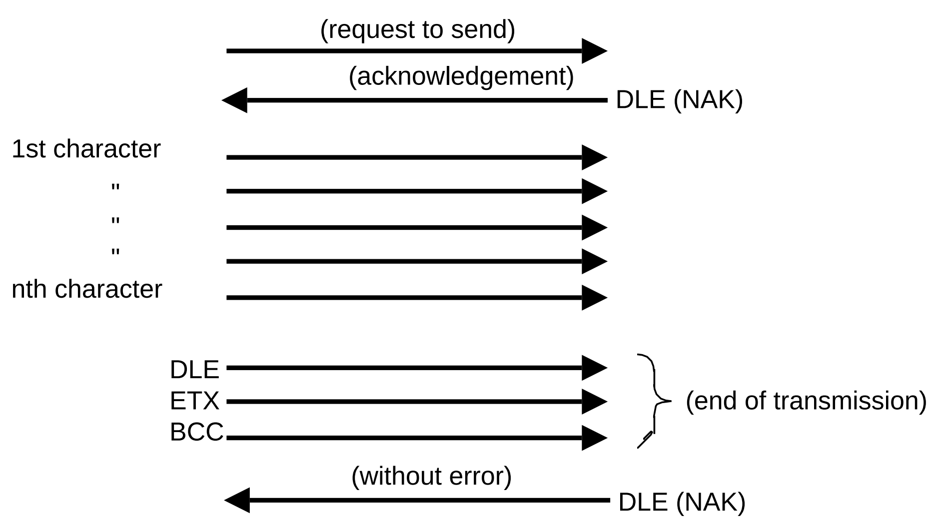
If the peripheral unit sends the NAK character during a transmission session, the driver terminates the block and repeats the procedure as described above.

Example:

Example of an error-free data traffic using the 3964R procedure:

SINUMERIK 8X0 with CP 315
3964R procedure

peripheral unit
(e.g. read/write station)



For more detailed information on data reception using 3964R, initialization conflict, character frames, data protection and monitoring times please refer to the procedure description in the appendix (see Section 9, Appendix).

The following parameters of the 3964R procedure are configurable:

Transmission rate	(speed of data transmission)
Priority	(master or slave; only valid in the event of a conflict)
Repetitions	(number of repetitions after negative or incorrect acknowledgement)
Monitoring times	(acknowledgement monitoring time for transmitter and character delay or monitoring time)
Block length	(128 bytes or 256 bytes)
Acknowledgement delay	(with/without)

3.3.2 LSV2 procedure

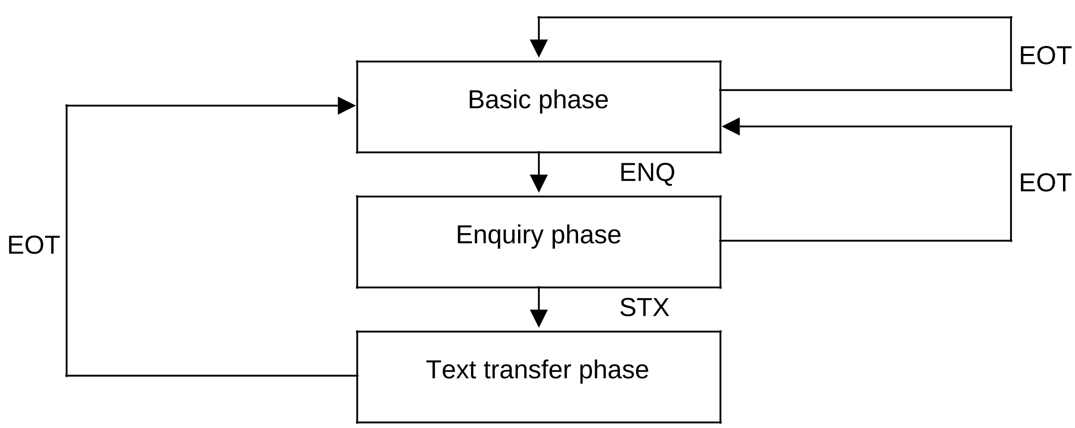
This procedure is based on the German DIN standard 66019.

The transmission procedure is an agreement on the data exchange procedure, the operating mode and other parameters. It ensures reliable transmission of data. LSV2 is an asynchronous synchronous data transmission procedure (start-stop procedure) for point-to-point connections.

Both stations are authorized to start transmission.

Phases of the procedure

3 phases are possible for the procedure:



Control characters initiating a change of phase are entered in the phase diagram. With the transmitting station, the change of phase is caused by transmission of a control character, and with the receiving station by reception of that character.

The ENQ character also occurs in the enquiry and text transfer phases but does not cause a change of phase.

Transmission is idle during the basic phase. The station moves to the enquiry phase through a request to send or when it receives the ENQ character.

In the enquiry phase, the station informs the other station that it wishes to send, where upon the station at the other end assumes its status as receiving station. In the text transfer phase, the text is transmitted in the form of a message frame.

The following control characters are used:

ENQ **Enquiry**
The ENQ data transmission control character is transmitted to a station as "request to respond". ENQ requests the receiver to send an acknowledgement character.

DLE **Data Link Escape**
With the DLE data transmission control character it is possible to give another meaning to a limited number of directly following characters. In conjunction with characters 0 and 1 the character combinations DLEO and DLE1 are used as positive acknowledgement. Due to the DLE control character, the next character is interpreted as control character if it has been coded as data transmission control character. If DLE occurs as character in the text, it has to be doubled and is interpreted as text character.
In the receiving station this is undone. If a single DLE occurs in the text with any character, the block is rejected as incorrect and acknowledged with NAK.

DLE/ENQ **Block abort**
If the transmitting station sends DLE/ENQ anywhere in the text, this means that the block is aborted. The receiving station must respond with NAK.

DLE/STX **Start of Text**
The DLE/STX transmission control character directly precedes the text. During block securing, the adding procedure to obtain the block check character on STX begins with resetting the block check unit to ZERO.

DLE/ETX **End of Text**
DLE/ETX signifies the end of a text and indicates that the next character is a block check character (BCC). ETX is always included in the adding process for block checking.

EOT **End of Transmission**
This transmission control character is used in order to identify the end of transmission of texts. The text transmitting station sends EOT in order to either indicate transmission end or abort. The transmitter abandons control of transmission. Both stations (text transmitting and receiving stations) return to the basic phase.

NAK **Negative Acknowledgement**
The text receiving station sends NAK as negative acknowledgement to the text transmitting station. This is an acknowledgement to a request to receive in order to indicate that the station is not ready to receive. NAK is sent during the text transfer phase as acknowledgement to the reception of a transmitted text if the transmission has not been received error-free and the text receiving station is ready to receive this block once again.

Data exchange sequence

The procedure can be split up into three phases:

- Enquiry phase
- text transfer phase
- completion phase

Enquiry phase		Text transfer phase		Completion phase
MA	SL	MA	SL	MA
ENQ	DLEO	DLE STX Text DLE ETX BCC	DLE1	EOT

MA : Master station = text transmitting station
SL : Slave station = text receiving station

For more detailed information on the enquiry phase, text transfer phase, completion phase, repetitions, monitoring times, collisions and data protection please refer to the LSV2 procedure description in the appendix (Section 9).

The following values of the LSV2 procedure are configurable:

Transmission rate	(speed of data transmission)
Priority	(master or slave; only valid in case of a conflict)
Repetitions	(number of repetitions after negative or incorrect acknowledgement in the enquiry and text transfer phases)
Monitoring times	(acknowledgement monitoring time for transmitter/receiver and character delay or character monitoring time)
Block length	(128 bytes or 256 bytes)
Acknowledgement delay	(with/without)

3.3.3 XON/XOFF procedure

The XON/XOFF line procedure is a character-oriented transmission method. The receiving system controls transmission of data of the transmitting system by means of XON/XOFF. Data is generally transmitted in half-duplex mode and in blocks. The user need not take into consideration block lengths or data overflow.

XON/XOFF is generally represented by the control characters DC1/DC3 (11H/13H); however, both control characters are user-configurable. The DTR, DSR, RTS and CTS control and signal lines need not be connected for the XON/XOFF procedure. The DSR signal line is an exception. The evaluation of the DSR line (Data Set Ready) depends on the configuration. If the parameter DSR evaluation has been initialized with "with", the DSR status is interrogated when transmitting the first character.

The XON/XOFF procedure distinguishes between two (configurable) variants.

3.3.3.1 Communication between "intelligent" and "non-intelligent" (unequal) nodes

The receiving station sends the XON character after initiation of the "Data Start" command. As a precondition, the receiving station must be configured as a **slave**.

The transmitting station must wait for the XON character to be transmitted before starting transmission. For this scope of functions, an additional configuring parameter is available - namely, "Waiting for 1st XON". "Waiting for 1st XON" is independent of the priority of the station.

After the XON character has been sent by the receiving station, the station on the other end starts transmission. If transmission is terminated by a valid end criterion, another transmission can be initiated only by the "Data Start" command.

The receiving station sends XOFF if it is no longer ready to receive. With regard to the printer, this would be the case if the print buffer is full or the OFF line key is pressed or if the printer enters the faulted state.

The CP 315 sends an XOFF character if the receiving scratch (input buffer for the respective procedure) is disabled, e.g. in the case of equipment bottlenecks, or if the receiving scratch is full or previous jobs cannot be passed on due to missing acknowledgements.

The station sends an XOFF character even if the interface is not yet assigned.

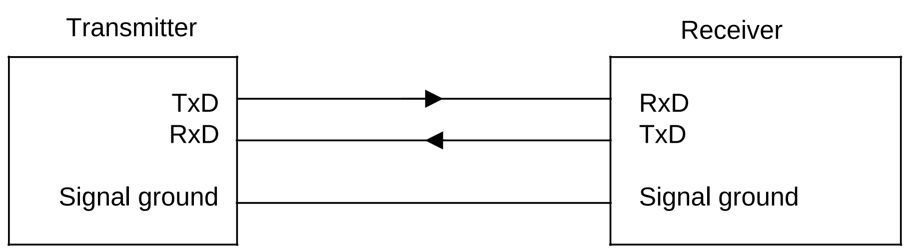
3.3.3.2 Communication between "intelligent" (equal) nodes

In the initial state, both nodes are ready to send and ready to receive; no control characters are transmitted. The procedure does not prevent any initialization conflict (both stations simultaneously attempt to send). This must be achieved by the application. In addition to the control characters XON and XOFF, the STX character is recognized as a control character.

If a node that is configured as slave receives an STX character during transmission, it stops transmitting and waits for the receiving job. This receiving job must arrive within a specified waiting period (line reservation time). If the waiting period expires without a message being received, the station goes into the initial state.

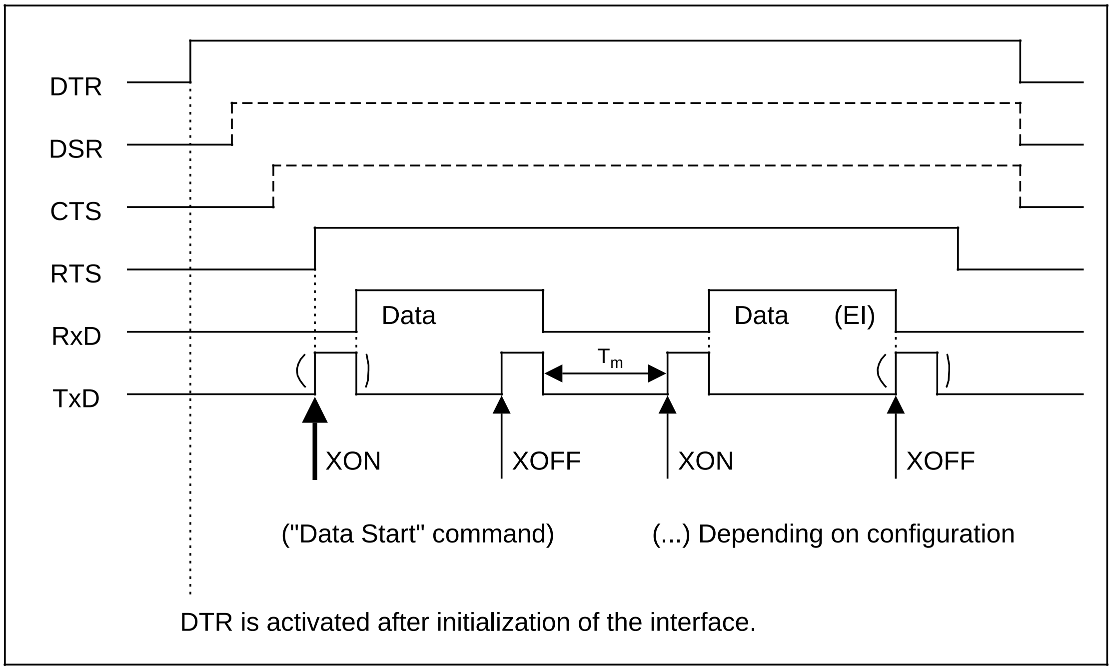
The transmission job interrupted by STX is then resumed.

Minimum wiring required for the XON/XOFF procedure



3.3.3.3 Signal chart for the XON/XOFF procedure

3.3.3.3.1 Read data



The pin assignments refer to the receiver in the previous figure (EI = End identifier)

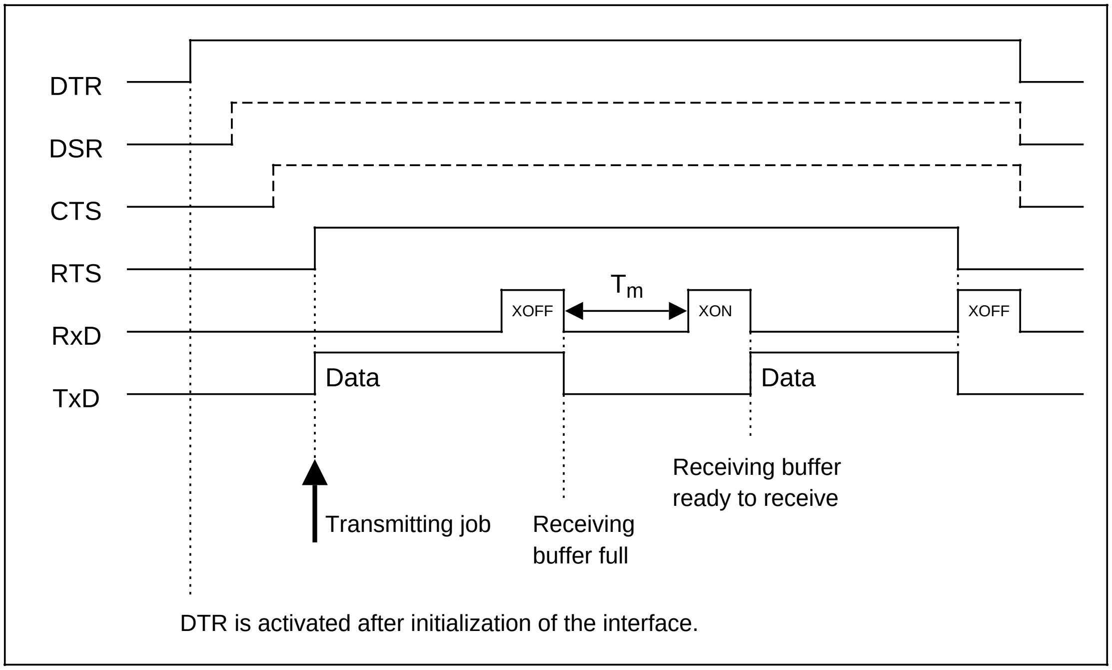
After initialization of the interface, the DTR signal line is activated. After the command "Data Start" has been issued, the station transmits the XON control character.

If no free receiving scratch is available to the driver, it transmits the XOFF control character. Under these circumstances, however, the driver is still able to receive and buffer up to ten characters until a free scratch is available again. The driver checks every 100 ms whether a new scratch is available. As soon as the receiving scratch is free, the buffered characters are copied into the scratch and an XON character is sent. The station is now again ready to receive.

A receiving job is terminated if the end criterion is met. The driver then terminates the receiving scratch and passes it on to the higher-level buffer manager of the device management level. Receiving jobs where the end criterion is not complied with are rejected by the driver.

The end identifiers for the input direction can be configured by the user who can decide whether to assign no, one or two end identifiers.

3.3.3.3.2 Data output



The pin assignments refer to the transmitter shown in the figure of Section 3.3.3.2.

After initializing the interface, the DTR signal line is activated. After a transmitting job has been transferred to the transmitting scratch (output buffer of the respective procedure), the procedure driver is started. Depending on the configuration, the DSR status is checked before transmitting the first character. A waiting period for the first XON character can also be configured.

Data transmission is interrupted only after XOFF is transmitted by the receiver. The procedure driver waits for the XON control character to be transmitted until the monitoring time T_m has.elapsed.

If the monitoring time T_m expires without an XON character received, the transmitting job is returned to the transmitter with a negative acknowledgement.

The monitoring time T_m is necessary for enabling the interface after a failure of the peripheral unit.

The following parameters of the XON/XOFF procedure are configurable:

Transmission rate	(Speed of data transmission)
Monitoring times	(Character delay or character monitoring times and device monitoring time)
DSR evaluation	(Check if node is active)
Number of end identifiers	(Number)
Node states equal	(Yes/no)
Waiting for the 1st XON	(Transmitting station waits for the receiver to transmit the first XON before starting transmission)
XON character	(Hexadecimal value)
XOFF character	(Hexadecimal value)
Line reservation time	(Monitoring time after initialization conflict)

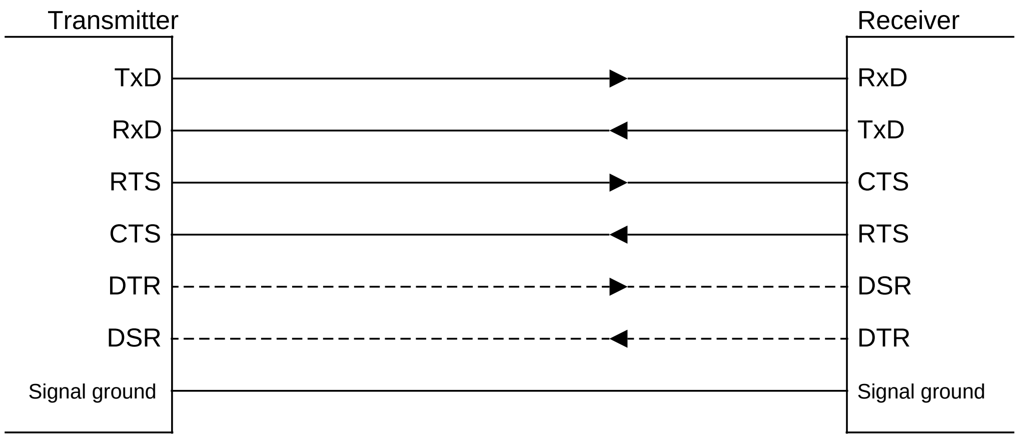
3.3.4 RTS-LINE (RTS/CTS)

The RTS-LINE transmission procedure is a line-controlled transmission method, i.e. the control characters XON and XOFF are not required. The functions of these control characters are carried out by the RTS and CTS signal lines. This procedure has full-duplex capabilities, i.e. data can be transmitted and received simultaneously. As in the case of the XON/XOFF procedure, the user need not take into consideration block lengths or data overflow. This is managed by the procedure.

If the procedure is configured as slave, the RTS signal line is activated for reading in data only after the "Data start" command is received. In this way, the transmitter is informed that it may start transmitting. If the procedure is assigned the priority "master", it is activated as soon as the RTS interface is assigned.

Depending on the configuration, the driver checks the status of the DSR line and evaluates it. For this purpose, however, the DTR and DSR modem control lines must be connected.

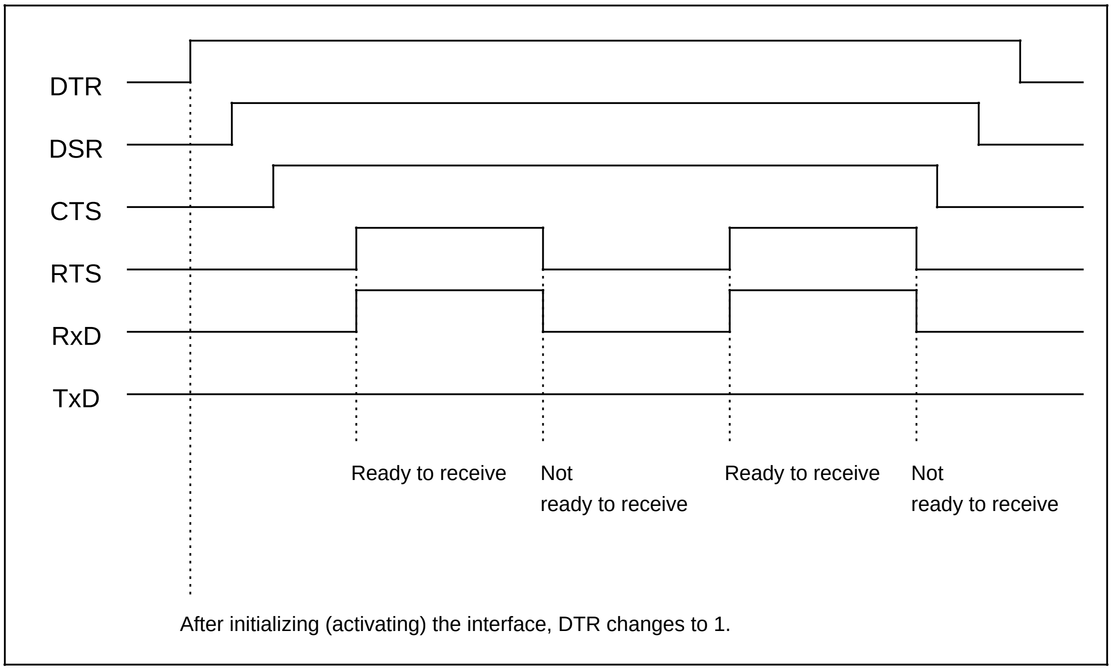
DTR is set and/or reset by the CP 315; DSR, however, is merely evaluated by the CP 315. As in the case of XON/XOFF, the end identifiers for the RTS-LINE procedure driver can be configured. For normal applications, however, these end identifiers are not required. If the procedure is configured "without" end identifiers, the expiry of the character monitoring time is used as the end criterion.



Wiring for the RTS-LINE procedure

3.3.4.1 Signal charts for the RTS/CTS procedure

3.3.4.1.1 Data input



The pin assignments refer to the receiver shown in the figure of Section 3.3.4.

After initializing the interface, the DTR control line is activated. The "Data start" command enables the procedure driver to activate the RTS control line (priority = slave). The CP is ready to receive from this instant, i.e. the receiving scratch is enabled.

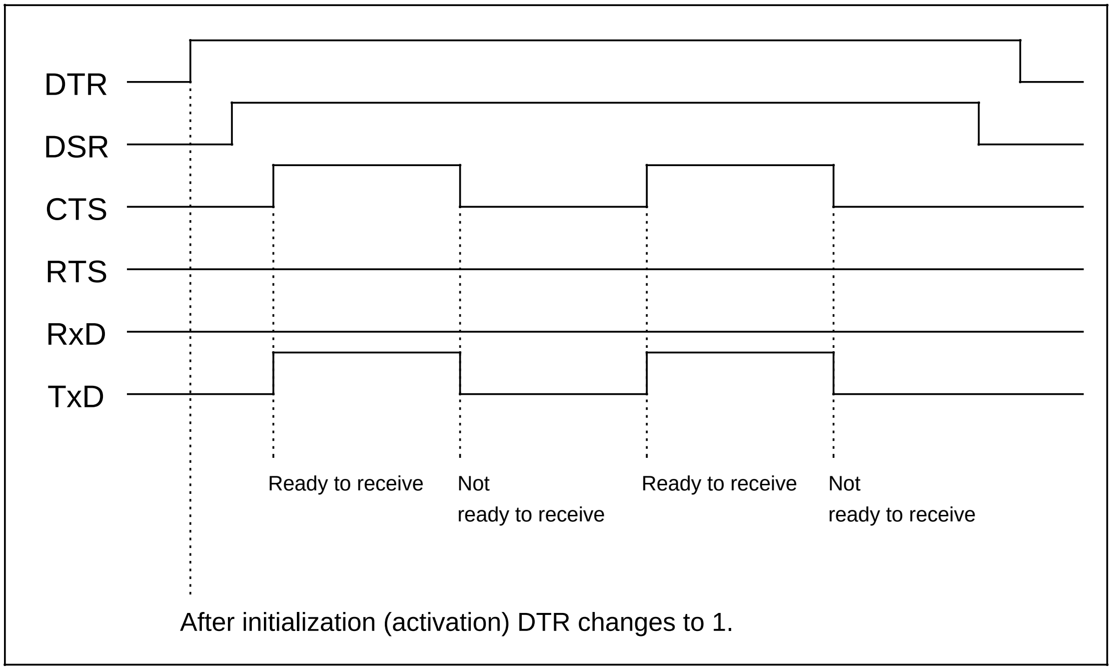
As soon as the receiving scratch is full but no new scratch is available, the driver deactivates the RTS. The output system must then stop transmitting. After clearing and/or after enabling the receiving scratch, RTS is again activated and the CP can continue reading of the data.

After deactivating the RTS procedure, the receiving node is still able to buffer up to ten characters.

If a new receiving scratch is available, the characters are copied into this scratch and RTS reactivated.

Transmitting jobs can also be processed in parallel to this job.

3.3.4.1.2 Data output



The pin assignments refer to the transmitter shown in the figure of Section 3.3.4.

After initialization of the interface, the DTR signal line is activated. The signal level of DSR is checked prior to transmission of the first character (depending on configuration). This interrogation is subject to time monitoring. If the DSR level does not change to 1 within a period of three minutes, the transmitting job is returned with a negative acknowledgement.

When a character is transmitted, the signal level at the CTS pin is checked first. If CTS is active, the character is sent. If CTS is inactive, the monitoring time T_m is started. The CTS status is then checked every 100 ms until the monitoring time expires. If the CTS signal does not change to the active state within the specified monitoring time, the driver returns the transmission job with negative acknowledgement.

The following parameters of the RTS-LINE procedure are configurable:

Transmission rate	(Speed of data transmission)
Monitoring times	(Character delay or character monitoring time and device monitoring time)
DSR evaluation	(Check if connected node is active)
Number of end identifiers	(Number)
Node states equal	(Yes/no)

3.3.5 Communication control in the Transparent Mode

Communication control as in the case of the XON/XOFF and RTS-Line procedures is not possible in the Transparent Mode. In its functionality, the Transparent Mode resembles the RTS driver, apart from the fact that the RTS and CTS signal lines are not used. The only signal line available is the DSR line which is scanned depending on the configuration.

Like the RTS driver, the Transparent Mode driver has full-duplex capabilities. The user can connect both an input device (e.g. keyboard) and an output device (e.g. screen) to the interface.

Configuring of up to two end identifiers is also possible here. In order to avoid any loss of data, however, the user must adhere to the relevant specifications (buffer size = 246 bytes) as no driver can be controlled. Flow control (initiating and stopping data input) must therefore be ensured by the user.

The following parameters of the TRANSPARENT procedure are configurable:

Transmission rate	(Speed of data transmission)
Monitoring times	(Character delay or character monitoring time)
DSR evaluation	(Check if node is active)
Number of end identifiers	(Number)

3.4 Protocols

The message frames in a computer link with the SINUMERIK 8X0 system, both bus and point-to-point links, consist of up to four parts (see Figure "Protocols for computer link"):

- Start of procedure
- Protocol header
- Message frame data
- Procedure termination

The differences between bus and point-to-point links differ with regard to the procedure (start, termination), protocol header and - depending on the configuration - the message frame data.

The procedure frame is described in Sections 3.1 and 3.2 or 3.3 whereas the protocol header is explained in this section.

The protocol header of a bus link is called AP 1 header and has a length of 22 bytes including the message frame data length.

In a serial link, each interface can be configured with or without AS512 protocol. The AS512 header has a length of ten bytes. The message frame data length requires two bytes.

The contents of the message frame data are identical for both bus and point-to-point links if the relevant serial interface is configured with a System 800 identification. A distinction is made between the SINUMERIK header (identification and error number) and the SINUMERIK useful data part on the one hand and the FMS message frames on the other hand. The description "SINUMERIK 850/880 Message Frames, Computer Link" includes a detailed description of SINUMERIK message frames.

If a serial interface is configured without System 800 identification, the data part of the message frame consists of useful data only.

For a detailed description of the message frame data, please refer to Section 6, Message frames.

Start of procedure	Protocol header		Message frame data			End of procedure
SINEC H1 (bus link)	AP 1 header 20 bytes	Message frame data length 2 bytes	Identification 8 bytes	Error number 2 bytes	Useful data 2)	SINEC H1 (bus link)
3964R/LSV2 (serial link)	AS 512 header 1) 10 bytes	Message frame data length 2 bytes	Identification 1) 8 bytes	Error number 2 bytes	Useful data 2)	3964R/LSV2 (serial link)
XON/XOFF, RTS-LINE, TRANSPARENT (serial link)	———		Useful data (max. 244 bytes)			XON/XOFF, RTS-LINE, TRANSPARENT (serial link)

Protocols for computer link

3.4.1 SINEC AP1 protocol

AP SINEC defines the sum of the rules by which communication between two AP applications is executed.

This sum of rules (i.e. the protocol) covers:

- Protocol sequences
  How does the node respond when it receives a job, and which session types exist.
- Coding and syntax
  How is an item of information represented and what is the structure of the.message.
- Meaning, semantics
  What can be communicated with the aid of the sequences and code in a standardized form.

SINEC AP standardizes these rules for SINEC in layers 5 to 7 of the protocol architecture. In this way nodes on different systems can exchange standardized information. Due to standardization of messages, it is possible to fully integrate layers 5 to 7 into the network management. As a result, complete control of the local area network system is ensured up to the AP application interface.

1) *configurable in interfaces*
with/ewithout AS 512 protocol
with/without System 800 identification

2) *Max. 244 bytes; when connected to Siemens FMS systems, max. 224 bytes.*

The following data transfer services are to be added to the above system aspects of the protocol:

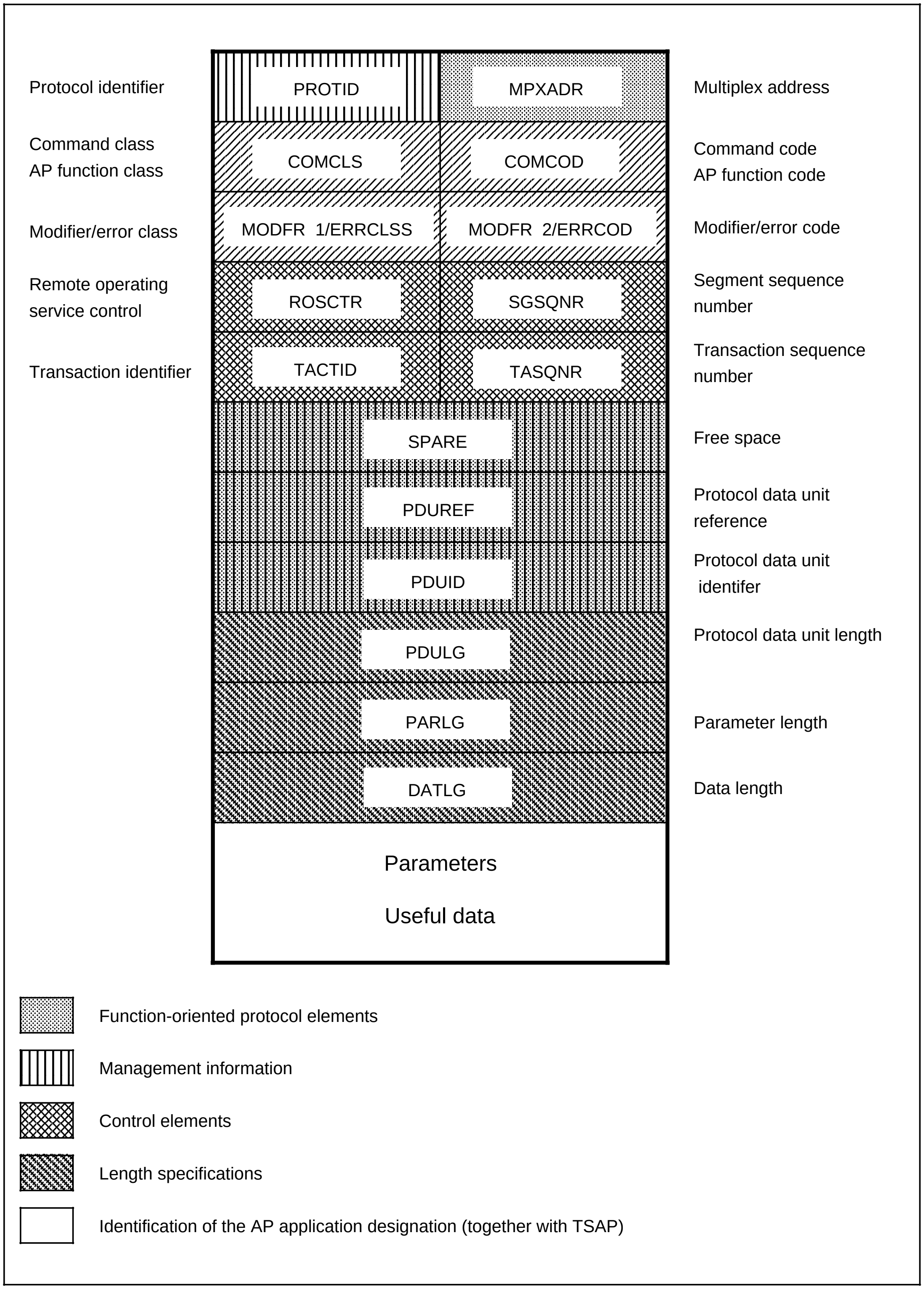
- Identification of the communication relation of an AP application
- Multiplexing at layer 7, i.e. a multiple use of lower-level resources, in this case a layer 4 link.
- Efficient job management with independent, configurable error correction measures.
- Standardization of the technological functions, i.e. standardization of "what" one node requests from the other node.
- SINEC AP is a protocol between AP applications, i.e. reliable data transfer ensured by a user acknowledgement is also a service.
- Segmenting, i.e. division of useful data into units which can be transmitted by the communication system and which can be managed by the AP applications as a unit.
- Transmission of information on error states in the communication system at layers 5 to 7.
- Modular, hierarchical structure (syntax) of the protocol, i.e. extendibility is ensured and simple implementation is possible.

The AP1 header is 22 bytes long (see figure "SINEC AP1 header (system part)”) and it can be split up into the following protocol element groups:

- Function-oriented protocol elements
  They determine the meaning of the message contents.
- Control elements
  These control the sequences and provide information on the session elements.
- Management elements
  These provide information for the administration of AP applications.
- Identification elements
  By means of these elements, a communication relation on AP application level is identified in the protocol.
- Size information
  They contain information on the data structure of the AP PDU.

The system part is the crucial part of the message. It transports in the protocol elements the codes which can be interpreted by the receiver or by the communication system.

For further specifications on SINEC AP, please refer to the "SINEC AP manual"
(Order no: 6 GK 1972-7AA02-0AA0).



SINEC AP1 header (system part)

3.4.2 Transparent data exchange

The SINEC H1 interface module of the SINUMERIK 8X0 uses the function "transparent data exchange" of the SINEC AP1 protocol.

The function "transparent data exchange" allows the user

- to transmit useful data with the job PDU from the client to the server
- and to transmit useful data from the server to the client with the acknowledgement PDU.

For transparent data exchange, a fixed AP1 header is used, the entries of which are constant except for the message frame length specifications.

The message frames are sent with a job PDU (figure "Message frame structure for job PDU") and acknowledged by the peer with an acknowledgement PDU (figure "Message frame structure of acknowledgement PDU"). The figures show the entries in the header fields.

The following is agreed by the entries in the AP1 header:

Job PDU:

PROTID $=00_{Hex}$ (Protocol Identifier)
The PROTID characterizes the protocol version. For SINEC AP1.0 it is preset with $00_{Hex.}$

MPXADR $=00_{Hex}$ (Multiplex Address)
Multiplexing in layer 7 is not used with connection to Siemens FMS systems.

COMCLS $=04_{Hex}$ (Command Class)
COMCLS describes global functions;04_{Hex} means serial transfer.

COMCOD $=02_{Hex}$ (Command Code)
To each global function belong single functions which are entered into COMCOD in a coded form.

COMCLS and COMCOD must always be considered in conjunction.
When COMCLS = 04_{Hex} COMCOD = 02_{Hex} transparent data exchange

ROSCTR $=01_{Hex}$ (Remote Operating Service)
ROSCTR is a control information. 01_{Hex} stands for job with acknowledgement.

SGSQNR $=00_{Hex}$ (Segment Sequence Number)
00_{Hex} determines that no segmenting will take place.

PDUREF (Protocol Data Unit Reference)

PDUID (Protocol Data Unit Identifier)
These values are freely assigned by the AP application.

PDULG (Protocol Data Unit Length)
This length describes the total length of the SINEC AP1 PDU.

Job PDU

AP 1.0 Header	PROTID 00_{Hex}	MPXADR 00_{Hex} not used
	COMCLS 04_{Hex} Serial transfer	COMCOD 02_{Hex} Transparent data exchange
	MODFR 1 00_{Hex} not used	MODFR 2 00_{Hex} not used
	ROSCTR 01_{Hex} Job with acknowledgement	SGSQNR 00_{Hex} no segmenting
	TACTID 00_{Hex} not used	TASQNR 00_{Hex} not used
	SPARE 0000_{Hex}	
	PDUREF xx	
	PDUID xx	
	PDULG xx Length of PDU in bytes	
	PARLG 0000_{Hex} not used	
	DATLG xx Length of message frame data in bytes	
SINUMERIK Messageframe	Identification	
	Error numbers	
	Useful data (max. 244 bytes, with connection to SIEMENS FMS systems max. 224 bytes)	

Message frame structure for job PDU

Acknowledgement PDU:

COMCLS= 00_{Hex} (Command Class)
00_{Hex} means acknowledgement without data

COMCOD (Command Code)
With AP acknowledgements, the COMCOD line describes the result of a job processing.
00_{Hex} = job executed properly
01_{Hex} = job executed properly, requested information is transferred with response job.
02_{Hex} = job executed properly, special conditions have occurred during execution. Additional information in the cells ERRCLS and ERRCOD. Entries greater than or equal to 10_{Hex} indicate error.

ERRCLS (Error Class)

ERRCOD (Error Code)
When errors occur, correspondingly coded entries are made in these cells. For entries in ERRCLS, ERRCOD see appendix (Section 9.3, XON/XOFF procedure))

ROSCTR= 02_{Hex} (Remote Operating Service Control)
02_{Hex} stands for acknowledgement without additional field, only with AP header.

PDULG This length defines the total length of the PDU.

PROTID	00_{Hex}	MPXADR Not used	00Hex
COMCLS Acknowledgement with data	00_{Hex}	COMCOD Result code	xx
ERRCLS	xx	ERRCCOD	xx
ROSCTR Acknowledgement without data	02_{Hex}	SGSQNR No segmenting	00_{Hex}
TACTID Not used	00_{Hex}	TASQNR Not used	00_{Hex}
SPARE			0000_{Hex}
PDUREF			xx
PDUID			xx
PDULG Length of PDU in bytes			xx
PARLG Not used			0000_{Hex}
DATLG			0000_{Hex}

Message frame structure of acknowledgement PDU

3.4.3 AS 512 protocol

For the active serial interface module, AP 512 is used as protocol (AP 512 header) in all procedures.

This protocol corresponds to the protocol of interface module 512 for SIMATIC.

In case of the computer link with the SINUMERIK 8X0, it is interpreted or generated on the interface module CP 315 by the device management.

SINUMERIK 8X0 uses the AP 512 header only to multiplex on one interface.

It is therefore possible to address several logical partners on one serial interface with the aid of the addressing parameters in the AP 512 header.

The parameters of the AP 512 header are not used for addressing within the SINUMERIK 8X0 system, this is done by means of the identification section of the SINUMERIK message frames.

By using the AP 512 protocol, the SINUMERIK 8X0 can be connected easily to other programmable controllers which use the AP 512 protocol, as e.g. SIMATIC.

The AP 512 header of the initial message frames consists of 10 bytes, followed by another two bytes for the message frame length. The message frame header is abbreviated in the case of response message frames, subsequent message frames and subsequent response message frames.

The response message frames can be suppressed if required.

With initial message frames the AP 512 header consists of 5 values at one word each:

- Identification,
- Command,
- Destination,
- Data volume and
- Coordination flag

The fields of the AP 512 header are used as follows:

- Identification
  0000_{Hex} for initial message frame
  $00FF_{Hex}$ for subsequent message frame
- Command configurable
  ASCII AD = 4144_{Hex} output of a DB
  ASCII AX = 4158_{Hex} data block output of an expanded DX data block
- Destination configurable
  0408_{Hex} starting address in destination data block
  (data block no., data word no.) e.g. DB4DW08
- Data word quantity variable
  The variable value specifies the number of the message frame data words,
  e.g. $0064A_{Hex}$, i.e. 100 data words
- Co-ordination flag configurable
  The byte and bit address of the respective flag is specified,
  e.g. 0607_{Hex} , i.e. flag byte 6 flag bit 7.

Computer link using CP 315

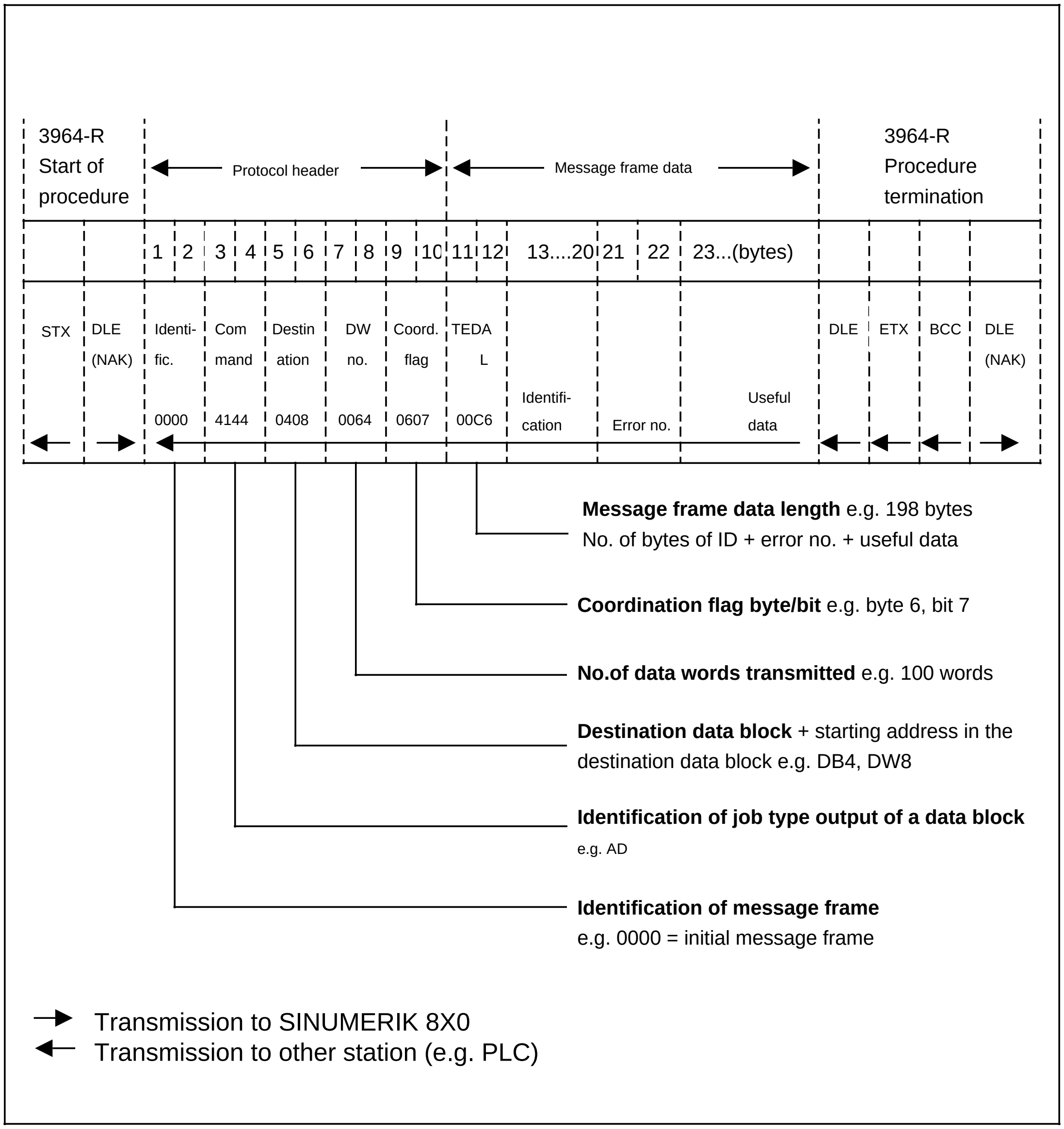
The session can be initiated by the SINUMERIK 8X0 or by its communication partner.

The possible message frame sequences between the SINUMERIK 8X0 and a communication partner (e.g. PLC) are shown in the following table.

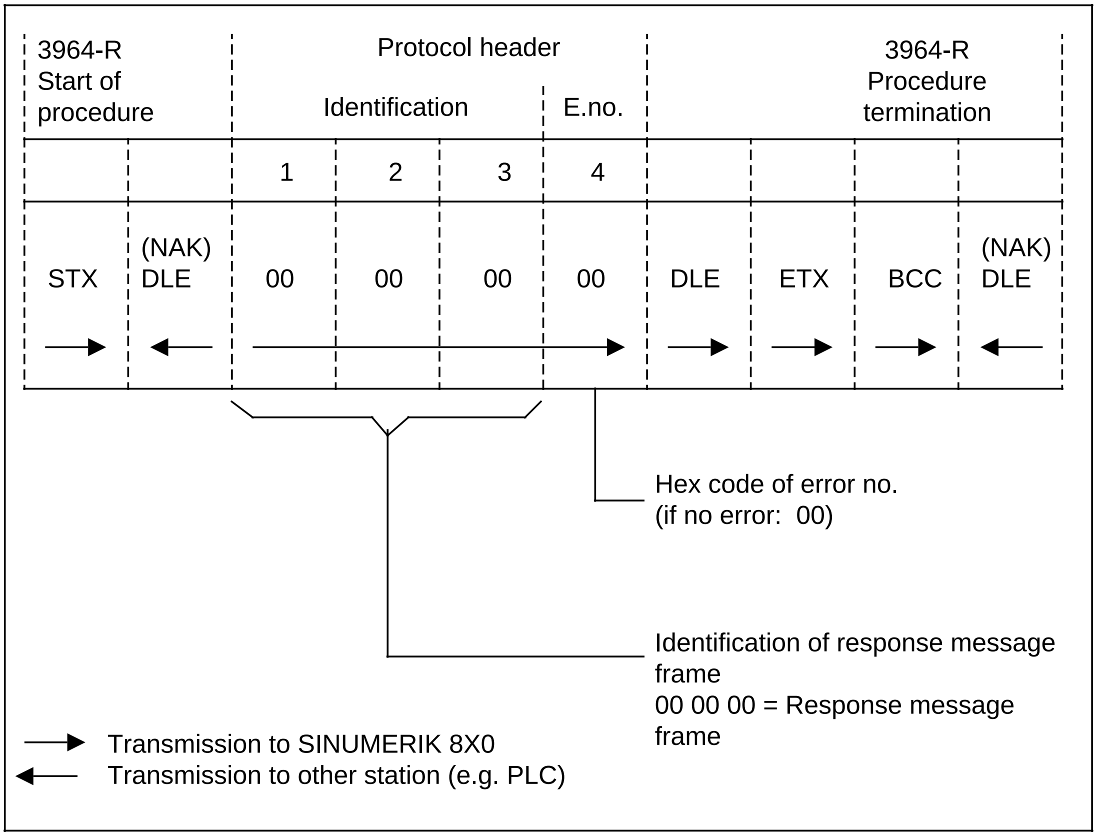
	Data sequence	PLC | SINUMERIK 8X0
1. Other station (PLC) is active PLC sends data to SINUMERIK 8X0		A S512 header + message frame data → Response message frame ←
2. SINUMERIK 8X0 is active SINUMERIK 8X0 sends data to PLC		AP 512 header + message frame data ← Response message frame →

In the following example, the complete message frame structure with the 3964R procedure is shown and details of the AP 512 header are explained.

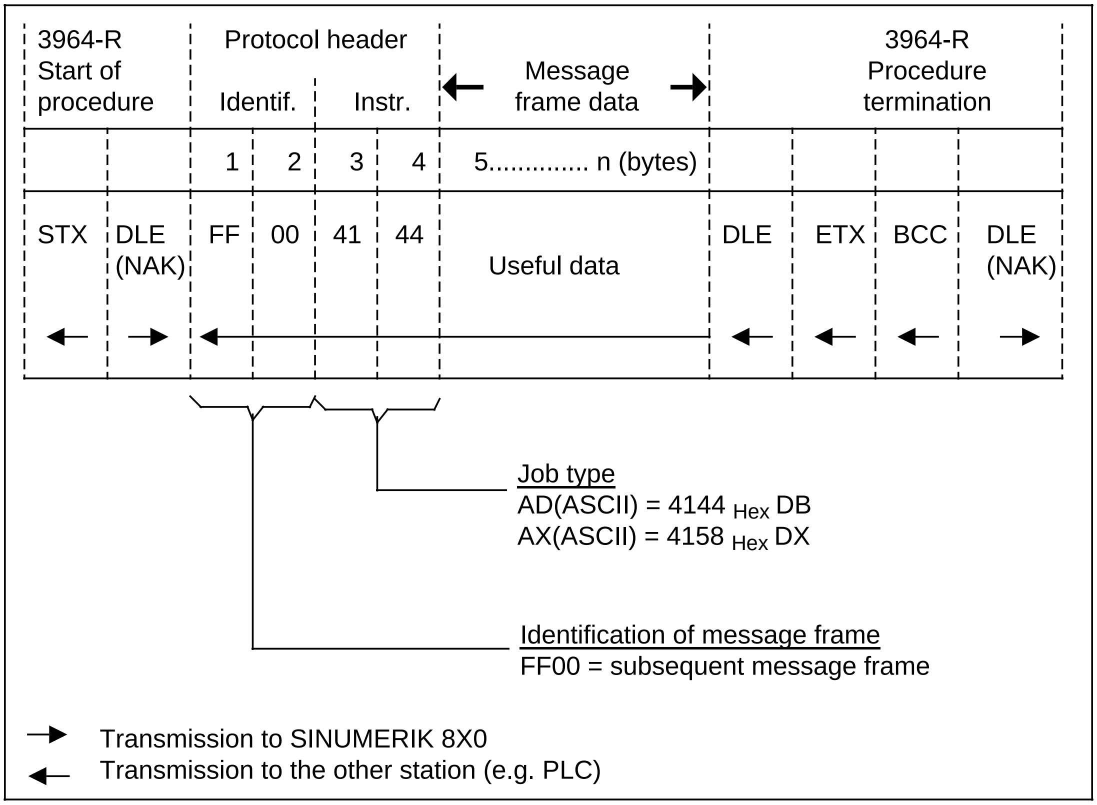
) Transmitting data from other station (PLC) to SINUMERIK 8X0
) Transmitting data from SINUMERIK 8X0 to the other station (PLC)



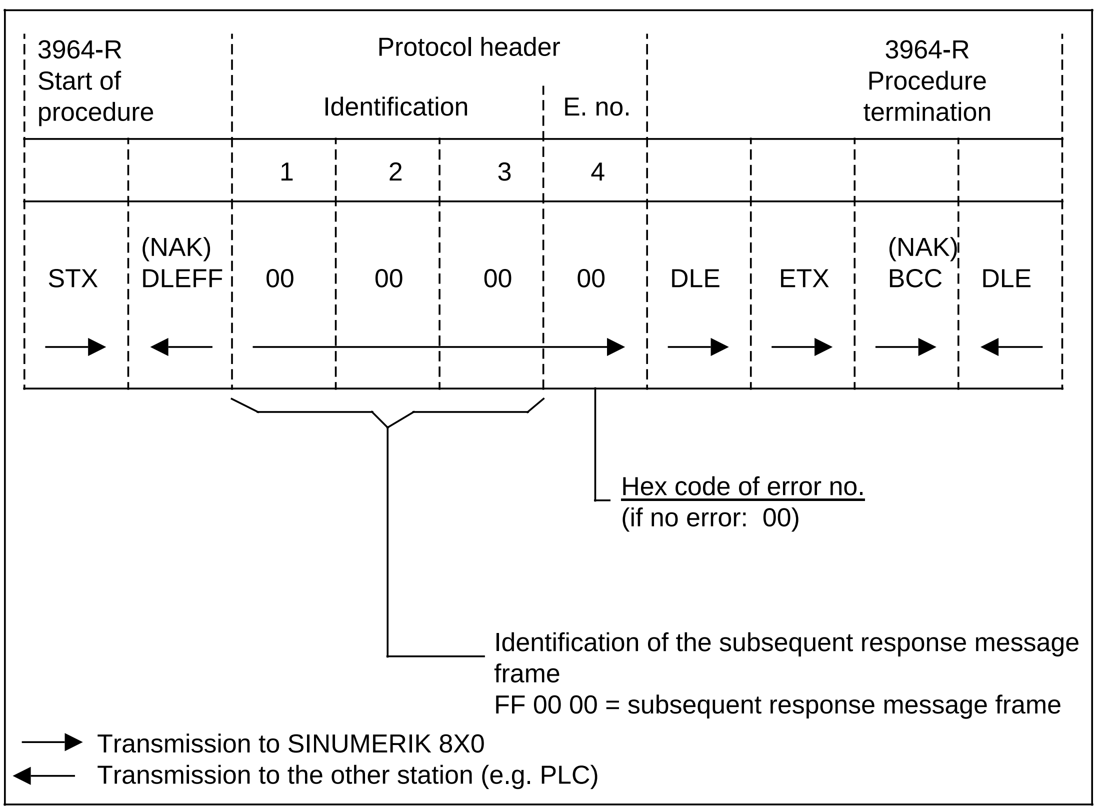
AS 512 action message frame



AS 512 response message frame



AS 512 subsequent message frame



AS 512 subsequent response message frame

Example:

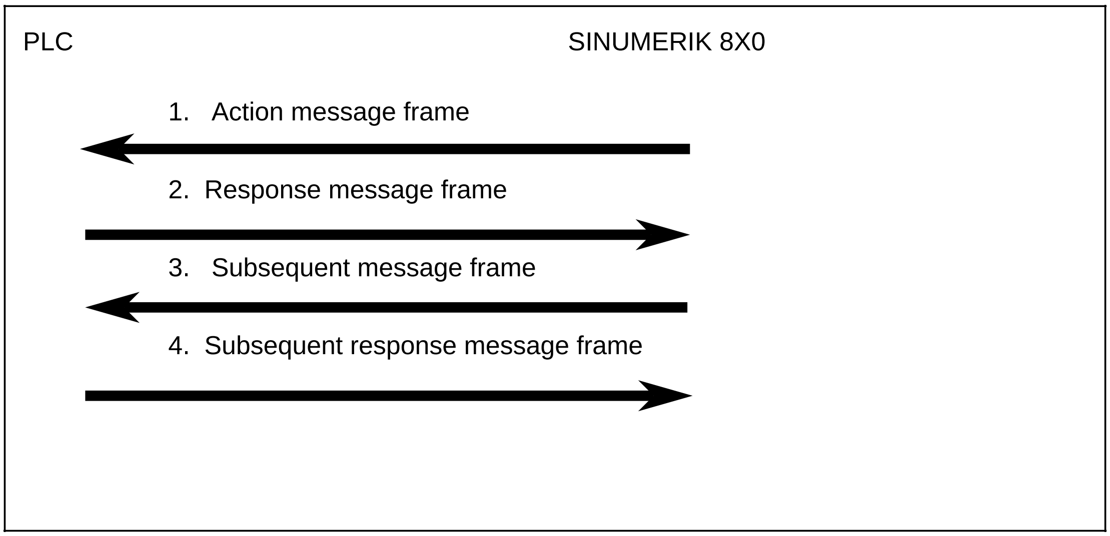
Send message frames from SINUMERIK 8X0 to another station (e.g. PLC)

In figures "AS 512 action message frame", "AS 512 response message frame", "AS 512 subsequent message frame" and "AS 512 subsequent response message frame", a message frame sequence between SINUMERIK 8X0 and another station (PLC) is shown as an example.

For these figures the following assumptions were made:

- SINUMERIK 8X0 sends message frames to the PLC (SINUMERIK is active)
- Response message frames are used
- The block length set on the CP 315 is 128 bytes
- The data volume to be sent is more than 128 bytes (e.g. 200 bytes)

On the basis of the above assumptions, the following sequence results for the session:



The session begins with an action message frame which is sent by SINUMERIK 8X0 to the PLC.

The message frame consists of the protocol header and the message frame data. The header includes the identification, the command (data transmission direction and addressing mode), the destination address, the data quantity and the coordination flag.

- Identification
  The 0000_{Hex} entry in the identification field informs the message frame receiver that this is an initial message frame
- Command
  In the command field, the job type is specified. In the case of a data block output, this field contains for example AD (ASCII) = 4144_{Hex}
- Destination
  In the destination field the destination data block of the receiver is specified in which the data is to be stored. Furthermore, in the destination field the data word is specified from which the entries are made in the destination data block. In figure "AP 512 action message frame" DB4, DW 8 = 0408_{Hex} has been entered.

- Data word number (DW number)
  The entry in the field DW number specifies the number of message frame data words to be transmitted.
  200 bytes are transmitted in the example, therefore 100 words = 0064_{Hex} are entered here.

- Coordination flag (Ko-merk)
  The entry in the coordination flag field specifies which flag is set in the destination system after arrival of the message frame.
  The destination device recognizes by means of this flag that a message frame has been received.
  In figure "AS 512 action message frame", for example, byte 6, bit 7 = 0607_{Hex} has been entered.

- Message frame data length (TEDA-L)
  The message frame data length in bytes is specified again following the AP 512 header. This length specification is not part of the AP 512 protocol. It indicates how many bytes follow after this value. In the example, the data volume is 200 bytes (including TEDA-L), therefore a value of 198 bytes =$00C6_{Hex}$ is entered in the TEDA-L field.

Response message frame (figure "AS 512 response message frame")

The response message frame is an acknowledgement at protocol layer. The protocol header is split up into the identification and error number fields.

- Identification
  The identification field is three bytes long. The entry 00 00 00 means response message frame.

- Error number (E. no.)
  In the case of a negative acknowledgement, an error number is entered in this field.
  In figure "AS 512 response message frame" an error-free transmission is assumed, therefore 00_{Hex} is entered.

Subsequent message frame (figure "AS 512 subsequent message frame")

In the example, a subsequent message frame has to be sent. The protocol header of the subsequent message frame is split up into the fields identification and job type.

- Identification
  The identification field is two bytes long. The entry $FF00_{Hex}$ indicates a subsequent message frame.

- Job type
  As with the action message frame, the job type has to be identified here, and AD (ASCII)= 4144_{Hex} is entered.

Subsequent response message frame (figure ”AS 512 subsequent response message frame”)

The subsequent response message frame structure is similar to that of the response message frame. The protocol header contains the identification and error number fields.

- Identification
  The identification field is three bytes long. The entry FF 0000_{Hex} identifies a subsequent response message frame.

- Error number (E.no.)
  Errors during transmission of the subsequent message frame are indicated in this field.

3.4.4 Communication with SIMATIC S5 systems

Direct communication between S5 programmable controllers and the PLC 135 WB integrated in the SINUMERIK 8X0 systems permits the use of different protocols on the application layer. For this purpose, the SINEC Technological Functions (STF) have been defined by Siemens as a common level-7 protocol. The SINEC Technological Functions are fully compatible with ISO 9506 (MMS), parts 1 and 2 with regard to the scope of functions and can thus easily be integrated in MAP 3.0. On the SIMATIC S5 side, the CP 143 enables the STF functions to be utilized within SIMATIC S5 systems.

3.4.4.1 STF extension

The implementation of the STF extension in the SINEC H1 module permits direct communication between the PLC 135 WB integrated in the SINUMERIK 8X0 system and the SIMATIC S5 programmable controller via STF.

STF (SINEC Technological Functions) offers the following functions

- Communication with SIMATIC S5 using jobs and acknowledgements
- Utilization of all STF services (variable services, domain services, ...)
- Segmentation of jobs and acknowledgements > 246 bytes (up to max. 2 Kbytes)

The STF extension is capable of simultaneously processing five jobs in the output direction and five jobs in the input direction if different logical links are used.

Only one job at a time can be transferred

via a logical link.

The relevant configuring parameters must be entered in input list 1 and the assignment list (logical partner/link list).

Entries in input list 1 for STF:

- Sub-address position: 255 (TF identification)
- Sub-address length: 4
- Sub-address contents: Logical partner target (4 ASCII characters)
  (logical partner target for input direction must also be entered in the assignment list.)

No fixed AP header is used for the STF extension. The SINUMERIK driver implemented in the interface module reads the COMCLS parameter in the AP 1.0 header to determine whether a TF message frame is transmitted and to determine the type of service required.

In the PLC, the STF message frames (including AP header) are made available via the user interface or useful data DBs to which user- configurable function blocks have access. These user function blocks control data transfer and/or flow control and interpret the STF message frames. No standard FBs are offered for this purpose. The function blocks required for the SINEC Technological Functions (e.g. variable services, domain services, ...) must be generated by the user and implemented in the PLC.

STF message frames

Identification bytes 6 to 10 are used for message frame identification, segmenting and error identification since they are available for all communications partners. At the user interfaces on the PLC side, this information is stored in DB 101 for input and in DB 102 for output.

The parameters are assigned as follows:

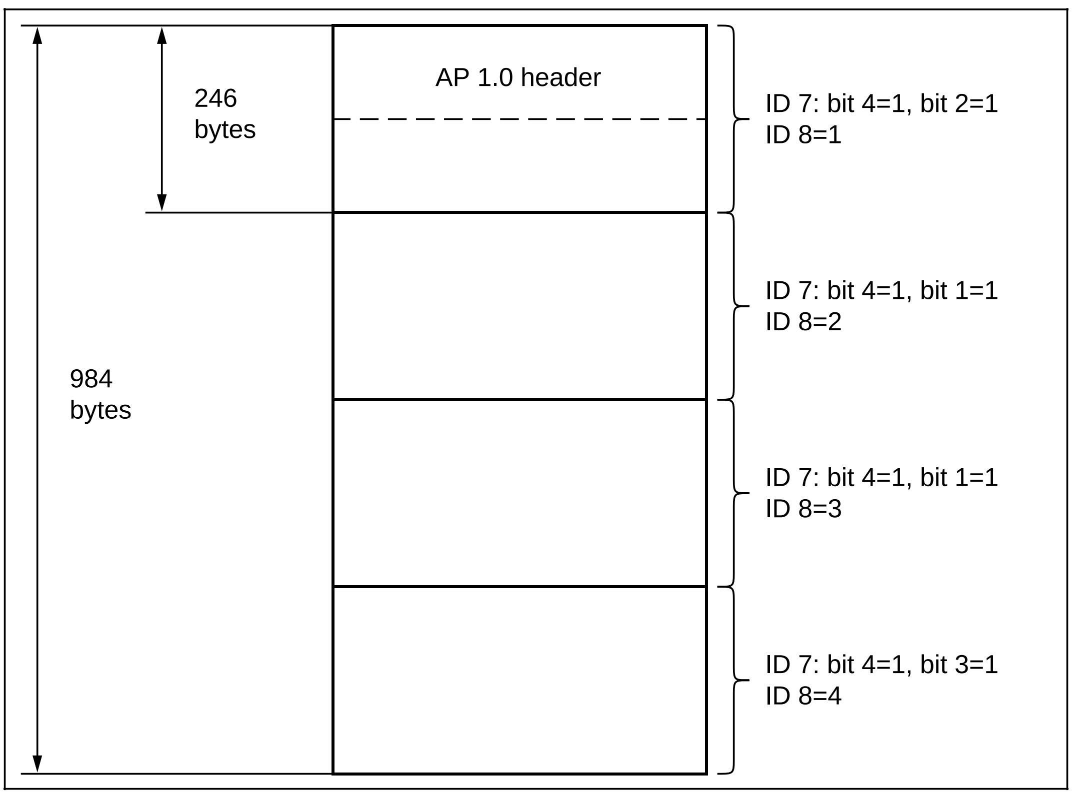
- The function number is not used (i.e. identification bytes 1 to 5 are not used).
- Identification byte 6: Not used
- Identification byte 7: Identifies the job type:

  Bit 0: 1=Single block   Bit 1: 1=Subsequent block
  Bit 2: 1=First block   Bit 3: 1=Last block
  Bit 4: 1=Job   Bit 5: 1=Acknowledgement
- Identification byte 8: Segment counter

  The first segment is assigned number 1, the second segment number 2, ...
- Identification byte 9/10: Error identifier or error number

  Error numbers which can be interpreted by means of an error list are entered here.
- DB-No./DW No.: Number of the first data block and first data word in which the useful data is entered.

Example:



The segmenting information enables the receiver of a message to recompile the job correctly.

This information is especially important if one buffer "overtakes" another within the transportation mechanism. Theoretically, the distribution mechanism of the system cannot prevent buffers belonging to the same job from overtaking one another. It is possible, for example, that a buffer with identification byte 8 = 3 arrives earlier than the buffer with identification byte 8 = 2. The segment number, however, ensures that the data can be allocated unambiguously to the relevant message frame.

3.4.4.2 Requirements for the application of STF

- Hardware requirements
  CP 231, any version
- Software requirements
  CP 231 firmware release from 2A (Order No. 6FX1 840-0BX01-2A)
  NML, any version

3.4.5 Remote communication with programmers via SINEC H1

(under development)

3.4.5.1 Scope of functions

The local programmer interface in the COM area is used for commissioning and configuring the SINUMERIK 8X0. The PLC integrated in the SINUMERIK 8X0 system and the programmer connected communicate via this serial link by means of the AS512 protocol. The utilization of programmer functions via SINEC H1 is being planned for a project in the field of material handling and storage system automation.

On the programmer side, an interface has been defined for remote commissioning of SIMATIC S5 systems via SINEC H1.

On the basis of this interface definition, the SINEC H1 interface module and the internal software of the SINUMERIK 8X0 shall be expanded to provide remote programmer functions for SINUMERIK 8X0 via SINEC H1.

The PG REMOTE function is optionally available and must be installed when required.

The BUS SELECTION utility program can be used to establish a bus link between a programmer and the relevant SINUMERIK system. A password (maximum of 8 ASCII characters) can be entered.

A password can also be included in the machine data MD 234/235 of the SINUMERIK system which is then checked by the CP 231 A interface module for congruence with the password entered via the programmer in the BUS SELECTION utility program. This feature ensures selective protection against unauthorized access via the PG REMOTE function.

The only prerequisite for the PG REMOTE mode is the assignment of an Ethernet address in the SINUMERIK system (specified either by means of the machine data MD 5141 to 5146 or by configuring the computer link with NML).

The local programmer port has priority over the PG REMOTE mode and can also be interrupted if necessary. The scope of functions of the CP 231 A "computer link" and PG Remote mode are to a large extent independent of one another.

3.4.5.2 Requirements for the use of the PG REMOTE function

- Hardware requirements:
  CP 231, any version
  PG with CP 536, from S5-DOS level 5
- Software requirements:
  CP 231 firmware release from 3A (Order No. 6FX1840-0BX01-3A)

3.5 Addressing lists

With the bus and the point-to-point connection, the header information of the message frames is evaluated on the interface modules.

3.5.1 Addressing lists of the bus interface module

The network software, the AP monitor and the SINUMERIK driver evaluate and generate the headers of the message frames (figure "Message frame processing using CP 231A"). The information necessary for this purpose is stored in lists. These lists are on the interface module and are configured correspondingly on start-up of the system.

Processing in input direction

The addressing data of plane 1 to 4 (Ethernet address, TSAP address) which are stored in the network header of the incoming message frame are interpreted by the network software.

Based on the address, it passes the message frame via a corresponding interface to the AP monitor.

The AP monitor analyses the entries in the AP header and transmits the message frames to the appropriate interface of the SINUMERIK driver.

The information required by the AP monitor for interpretation of the AP header are listed in the application relation table and function distribution table.

The SINUMERIK driver evaluates the SINUMERIK header (identification) of the SINUMERIK message frame. It assigns a SINUMERIK internal coding (function number) to the header information and determines the addressing data to be passed on within the SINUMERIK. The SINUMERIK driver takes the necessary information from input list 1.

The SINUMERIK driver routes the message frame to the dual port RAM and there stores the relevant addressing data which have been drawn from input list 1.

Using these address data, the message frame is passed on by the COM-CPU within the SINUMERIK.

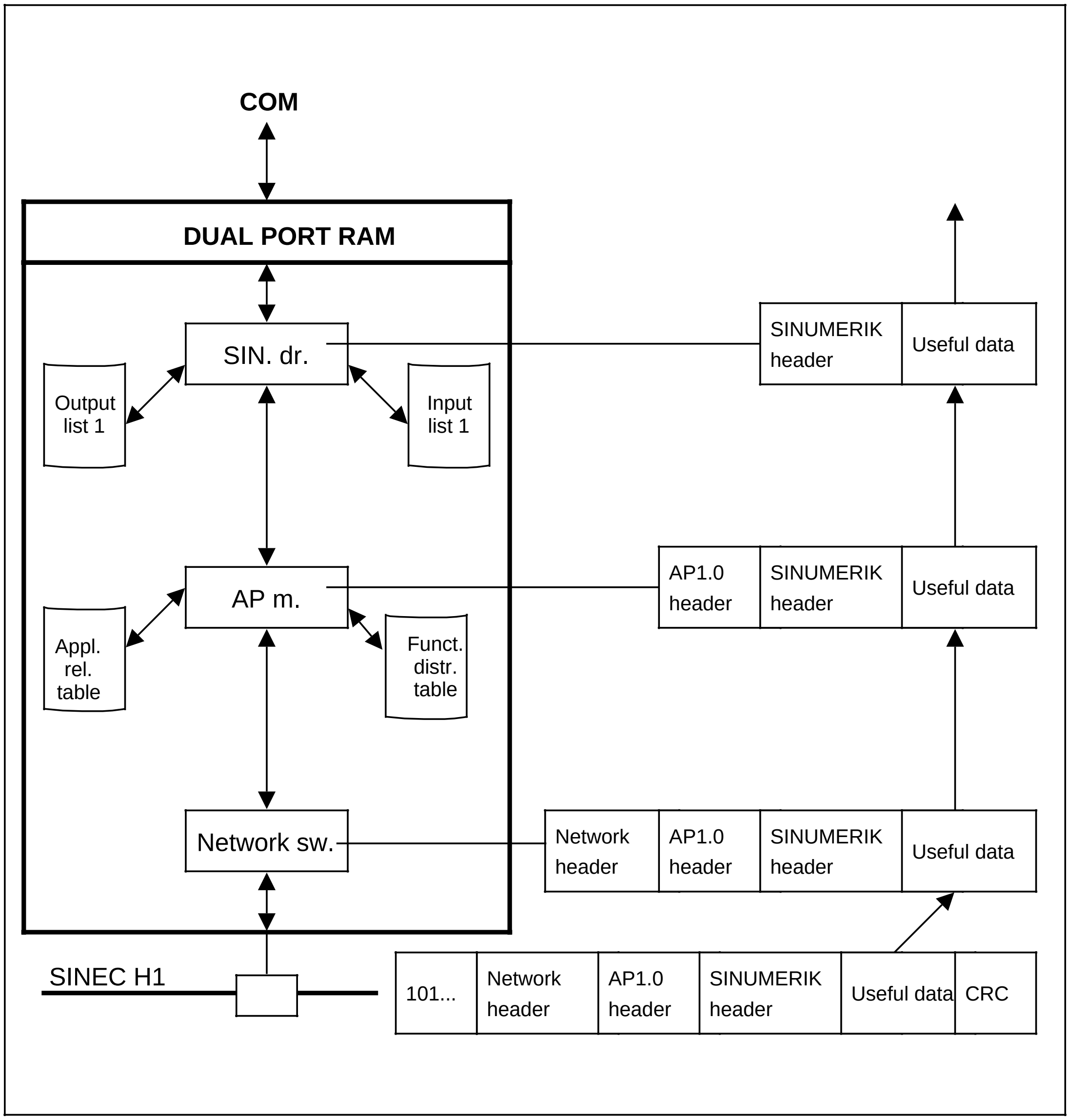
Processing in output direction

A message frame to be sent is first stored by the COM-CPU in the dual port RAM. The system-internal addressing data belonging to the message frame are also stored by the COM-CPU in the dual port RAM. The SINUMERIK driver is then notified via a control bit that a message frame which is ready to be sent is waiting in the dual port RAM.

Then the SINUMERIK driver generates the SINUMERIK header for this message frame. It takes the necessary information from output list 1. The SINUMERIK driver transfers the message frame to the AP monitor.

The AP monitor generates the AP header for the message frame and allocates the message frame to the required connection, i.e. the corresponding interface of the network software. The necessary data and/or allocations takes the AP monitor from the application relation table and function distribution table.

The network software receives the message frame to be sent from the AP monitor via a specific interface and then generates the network header containing the network addressing data (Ethernet address, TSAP address). Then it sends the message frame to the SINEC H1 bus.



Message frame processing using CP 231 A

3.5.2 Addressing lists of the serial interface module

The procedure driver, the device management and the SINUMERIK driver evaluate and generate the protocol and message frame header on the serial interface module if these are configured (figure "Message frame processing using CP 315").

The necessary data for this purpose are stored in lists which are located on the interface module. These lists are configurable.

3.5.2.1 For message frames with AS 512 protocol and with identification (FMS message frames)

Processing in input direction

The procedure driver of a serial interface recognizes that the connected communication partner wants to send a message frame to the SINUMERIK and manages the processes necessary to control the data flow. After error-free reception of the message frame at procedure level, the procedure driver passes the message frame data to the device management.

The device management evaluates the entries of the AS 512 header and makes internal allocations from the combination of interface number and AS 512 parameter. The device management takes the necessary information from the interface list and the addressing list. Then the device management routes the message frame to the SINUMERIK driver.

The SINUMERIK driver evaluates the SINUMERIK header (identification) of the SINUMERIK message frame. It assigns a SINUMERIK-internal code (function number) to the header information and determines the address data to be passed on within the SINUMERIK. The SINUMERIK driver takes the necessary information from input list 1.

The SINUMERIK driver routes the message frame to the dual port RAM and stores there the corresponding address data which were determined from input list 1. On the basis of these address data the message frame is passed on within the SINUMERIK by the COM area.

Processing in output direction

A message frame which is to be sent is first stored by the COM area in the dual port RAM. The COM area also stores the system-internal addressing data belonging to the message frame in the dual port RAM. Then the SINUMERIK driver is informed via a control bit that a message frame which is ready to be sent is waiting in the dual port RAM whereupon the SINUMERIK driver generates the SINUMERIK header for this message frame. It takes the necessary information from output list 1. Then the SINUMERIK driver passes the message frame on to the device management.

The device management generates the AS 512 header and uses the internal addressing data by assigning the message frame to the corresponding interface to which the addressed communication partner is connected. The information required by the device management is stored in the interface list and the addressing list. The device management transfers the message frame to be sent to the procedure driver.

The procedure driver informs the connected station that the SINUMERIK want to send a message frame and manages the necessary processes to transmit data at procedure level.

3.5.2.2 For message frames without AS 512 protocol and with identification

Processing in input direction

After error-free reception of the message frame, the procedure driver transfers the message frame data to the device management (no protocol header is available).

The device management interprets the entries with the aid of the interface and addressing list and transmits the message frame to the SINUMERIK driver.

The SINUMERIK driver evaluates the SINUMERIK header (identification) with the aid of input list 1. It enters the message frame data and the corresponding address data in the dual port RAM. The COM area fetches this data and passes it on.

Processing in output direction

The SINUMERIK driver calls a message frame stored by the COM area in the dual port RAM. The SINUMERIK driver generates the SINUMERIK header with the aid of output list 1. Then it passes the message frame on to the device management.

The device management evaluates the entries with the aid of the information from the interface and addressing list and passes the message frame on to the procedure driver.

The procedure driver transmits the data to the connected station.

3.5.2.3 For message frames with AS 512 protocol and without identification

Processing in input direction

After error-free reception of the message frame, the procedure driver transfers the protocol header and the message frame data (exclusively useful data without identification and error number) to the device management.

The device management interprets the AS 512 header and evaluates the entries with the aid of the interface and addressing list. Then it passes the useful data on to the SINUMERIK driver.

The SINUMERIK driver checks the message frame data length and the status of the interface (reserved/free) and enters the data into the dual port RAM where they are routed by the COM area to the multi-port RAM and then by the PLC software to the user interface.

Processing in output direction

The SINUMERIK driver fetches a message frame to be transmitted from the dual-port RAM. It checks the message frame data length and the status of the interface before passing the message frame on to device management.

The device management generates the AS 512 header with the aid of the interface and addressing list and transmits the message frame to the procedure driver, which makes sure that the data is properly transmitted to the connected station.

3.5.2.4 For message frames without AS 512 protocol and without identification

Processing in input direction

The procedure driver of a serial interface recognizes the request to send of the node connected and handles the operations necessary for data transmission. If a message has been received without any error, the procedure driver transfers the message frame data (only useful data since neither protocol nor message frame header are available) to the device management.

The device management evaluates the entries by means of the interface and addressing list and transfers the message frame to the SINUMERIK driver.

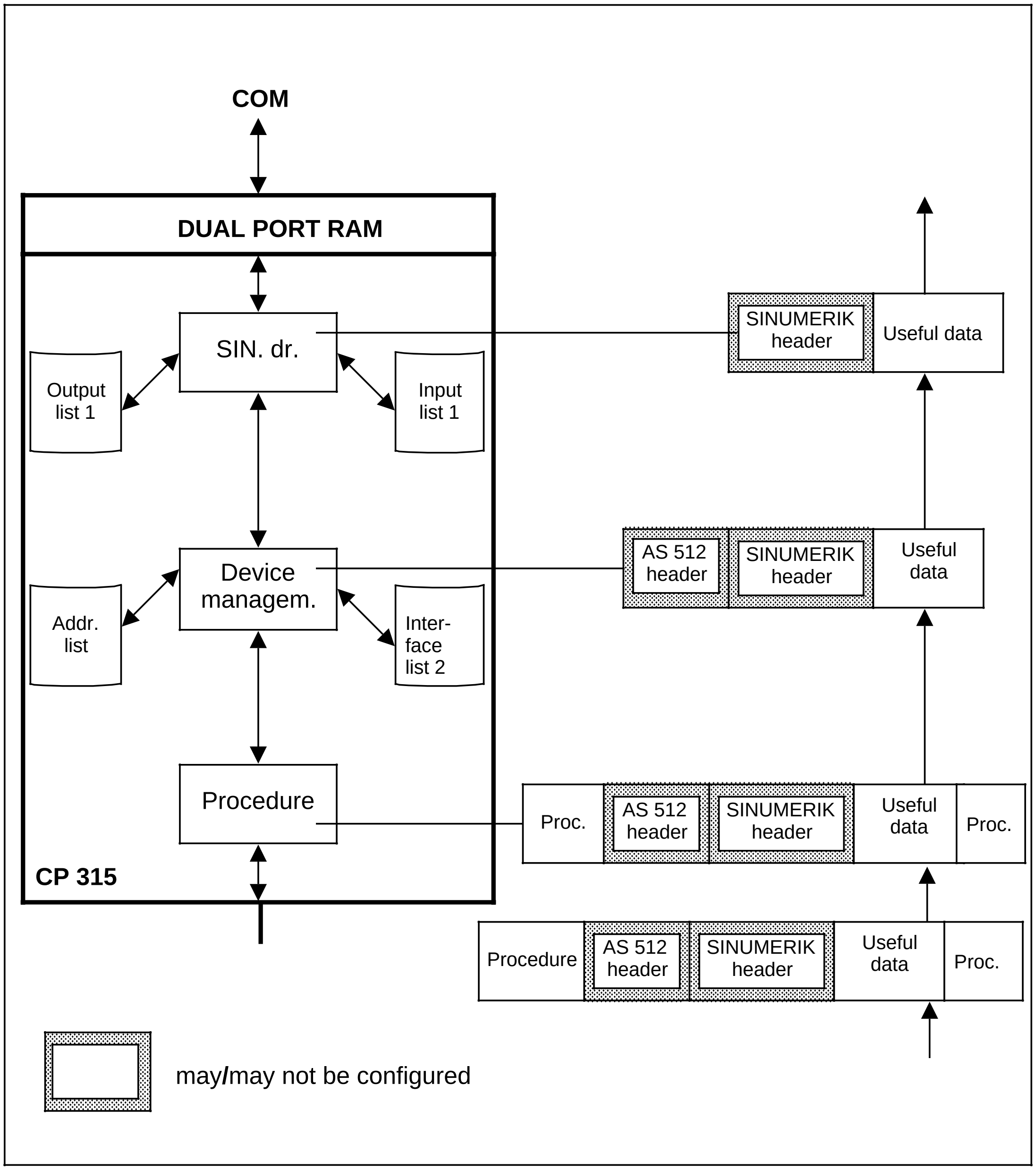
The SINUMERIK driver checks the message frame data length and interface status and enters the relevant message frame data, together with the associated address data received from the COM area at an earlier time and stored in an internal list, in the dual port RAM. There the COM area calls this data and passes it on.

Processing in output direction

A message frame to be transmitted is stored by the COM area in the dual-port RAM where it is called by the SINUMERIK driver.

The SINUMERIK driver checks the message frame data length and the status of the interface and transmits the message frame to the device management. The device management evaluates the entries in the interface and addressing list that are associated with the message frame and passes the message frame on to the procedure driver.

The procedure driver informs the node connected of the request to send and handles the operations necessary for data transmission.



Message frame processing using CP 315

4 COM

4.1 Structure

The outline structure of the COM area is shown in figure "COM structure". The task of the COM area is to distribute data or to process data by means of routines. The interface between interface module and COM area is the common dual port RAM. Using the local bus of the NC control the COM area can either retrieve data from the dual port RAM or store data there. The interface modules are passive, i.e. they do not access other memories via the local bus.

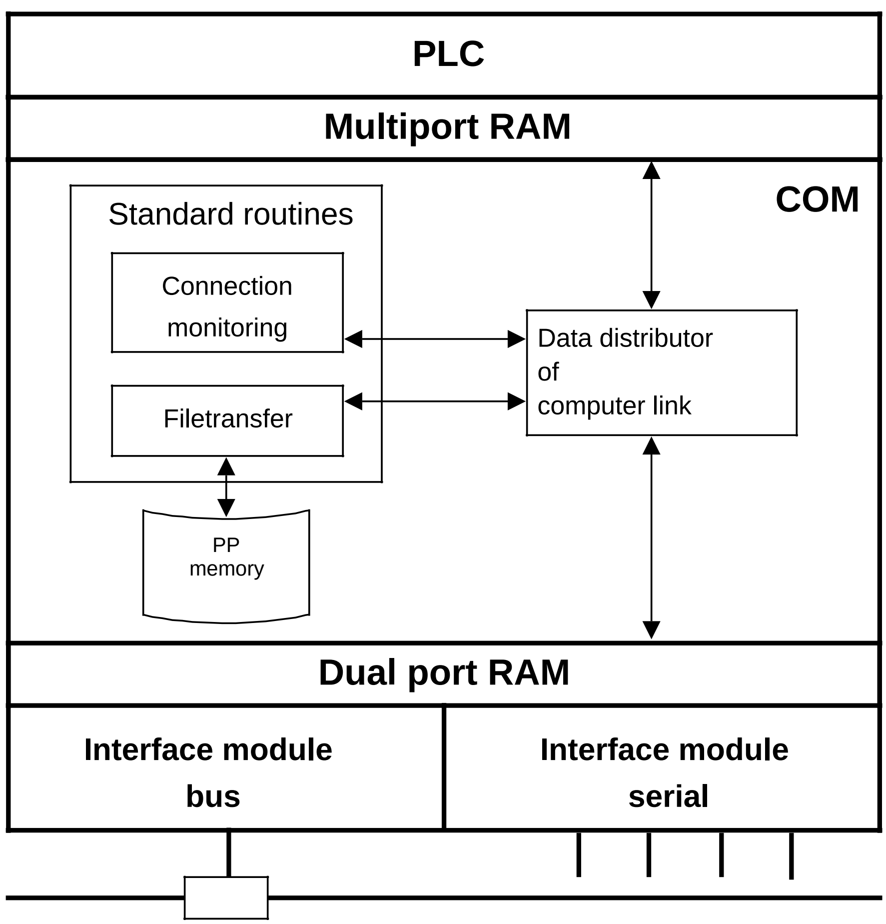
The interface between COM area and PLC is the multiport RAM. Data which is processed in the PLC is stored by the COM area in the multiport RAM.

Three processes are executed in the COM area:

- Store files in or retrieve them from the message frame memory
- Pass data to the PLC
- Process functions by means of standard routines.

The SINUMERIK 880 contains its own COM CPU.

The COM area in the SINUMERIK 840 is built into the NC CPU.



COM structure

4.2 Routines

The input buffer processing task cyclically calls on the interface module and PLC input buffers. If the cyclic monitoring fails between interface module and COM, the input processing task blocks itself.

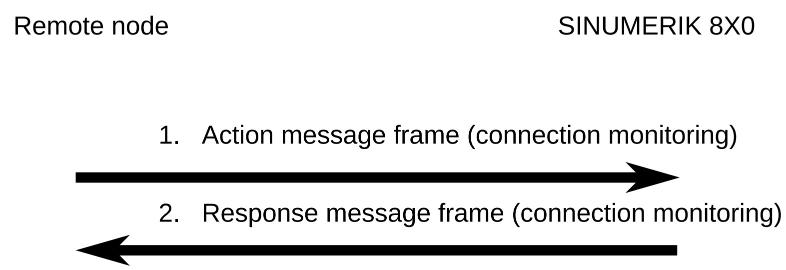
If the task finds a valid entry, it calls the data distribution module. On the basis of the address information belonging to the message frame, this module performs the distribution to the corresponding unit.

These processing units are the PLC, the interface modules and the standard routines.

4.2.1 Standard routines

Connection monitoring

The function "connection monitoring" is executed on the COM CPU (communication CPU) via a standard routine. With this function the connection to the CNC control can be monitored by a remote node. Connection monitoring consists of a simple sequence of message frames.



Connection monitoring is always initiated by the remote partner.

The response message frame is always output to this remote partner.

The distributor on the COM area activates the standard routine and enters the address of the input buffer into the transfer buffer.

The standard routine retrieves the function number from the input buffer in the dual port RAM and evaluates it. If this has the value 10, the standard routine generates a response message frame. If not, the input buffer is acknowledged with error number.

To generate the response message frame the standard routine also transmits the address data "sender location" and "logical partner sender" from the input buffer into the processing buffer. The "sender location" is required in order to send the response message frame to the interface module from which the initiation message frame came.

"Sender location"

CP1 - 1st interface module (CP: Communication processor)
CP2 - 2nd interface module

The following data is entered into the processing buffer:

	Value
Function number:	13
Receiver location:	"Sender location" from the initiation message frame
Log. partner destination:	"Log. partner sender" from the initiation message frame
Identifiers 7-10:	Id. 7 8 9 10
	- - 0 0

File transfer

The "file transfer" function is executed in the COM area by a standard routine. Loading, retransmitting and clearing of NC files is possible in this way.

The file transfer can be initiated by the partner node (e.g. production control computer) or by the CNC.

In order to achieve unambiguity, the system mode determines which node is authorized to initiate:

- In the normal mode only the partner node (e.g. production control computer) can initiate the file transfer.
- In the special/maintenance mode only the operator at the machine can inititate the transfer Only SINUMERIK 880).

5 PLC

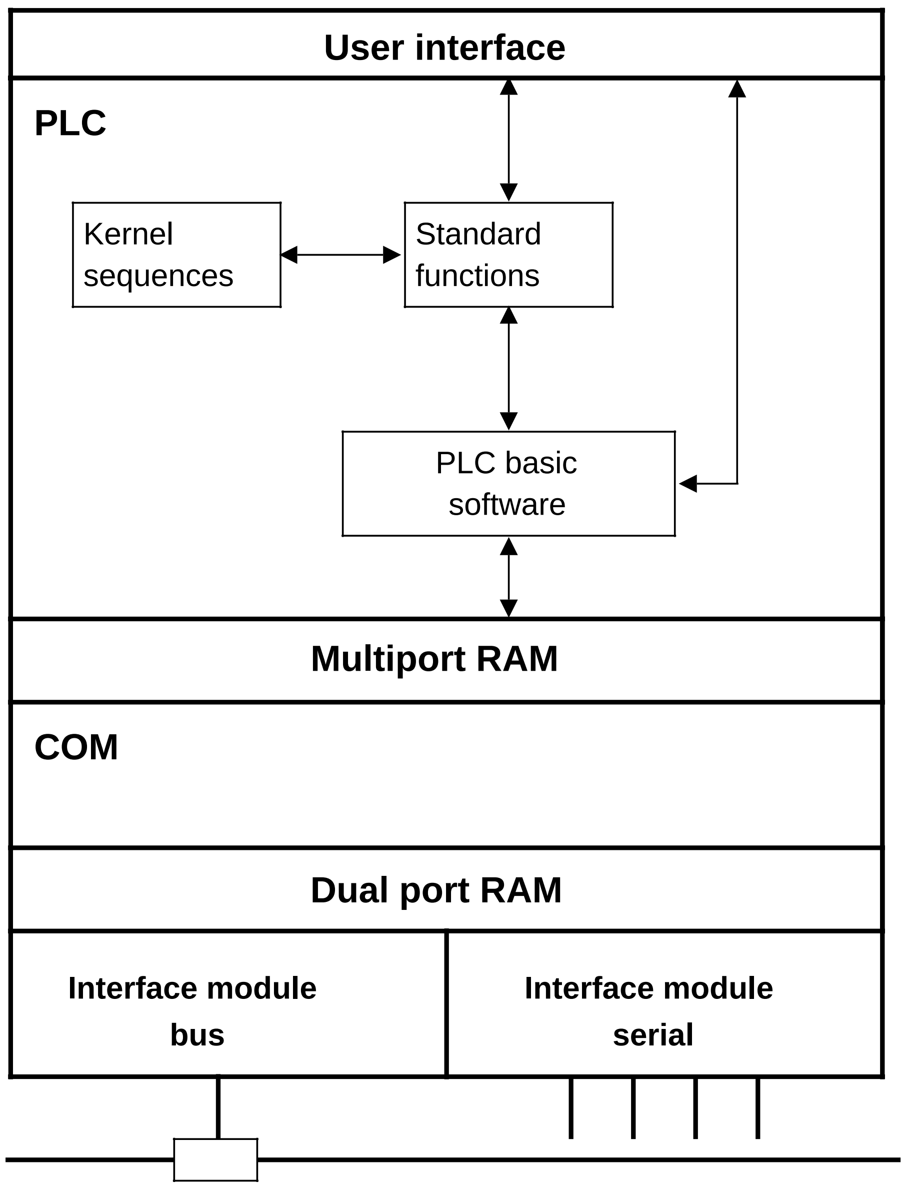
The PLC area is necessary for the input to the user interfaces and the processing of PLC standard message frames.

The PLC basic software supplies the user interfaces from the multiport RAM.

The standard function blocks process the standard message frames.

5.1 Structure

An outline overview of the PLC is given in figure "PLC structure".



PLC structure

The PLC software for the computer link with SINUMERIK 8X0 consists of the PLC basic software, the standard functions and the kernel sequences.

Processing in input direction:

If a message frame is addressed to a PLC, the COM area searches in the multiport RAM for a free buffer to the PLC and stores the message frame data there.

If the data have been assessed to be valid, the PLC basic software accesses the input buffer in the MPR (multiport RAM) and starts processing.

In case of a user message frame, the PLC basic software passes the message frame to the user interface and sets a flag there, by means of which the user software recognizes that a user message frame has arrived.

In case of a standard message frame, this is transmitted by the PLC basic software to the processing program for standard functions. The processing result is communicated to the user interface after the standard function has been processed.

Certain standard message frames are processed by kernel sequences.

Kernel sequences are fixed message frame sessions which cannot be changed by the PLC user program.

Processing in output direction:

If a message frame is to be sent via the user interface, this is indicated to the PLC basic software by means of a control flag. Then the PLC basic software copies the message frame into the multiport RAM and informs the COM area that a message frame which is ready to be sent is waiting in the multiport RAM.

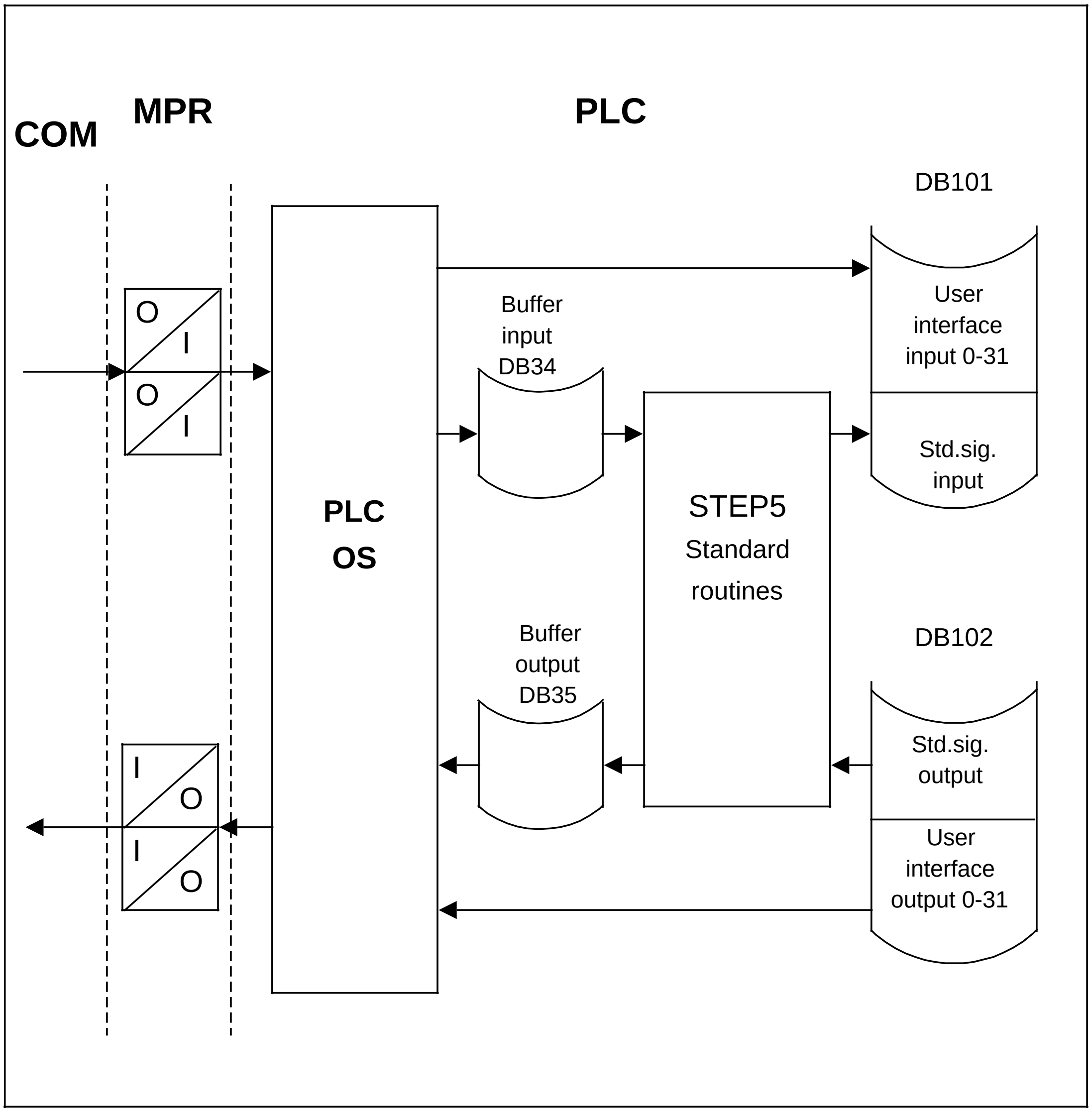
If a standard message frame is to be output, this is communicated to the PLC in a specific area of the user interface. Then processing and forwarding of the standard message frame is effected automatically by the standard function blocks.

5.1.1 User interfaces

The PLC provides to the user 32 input and output interfaces each (see figure "User interfaces").

The input interfaces are in DB 101 and the output interfaces in DB 102.

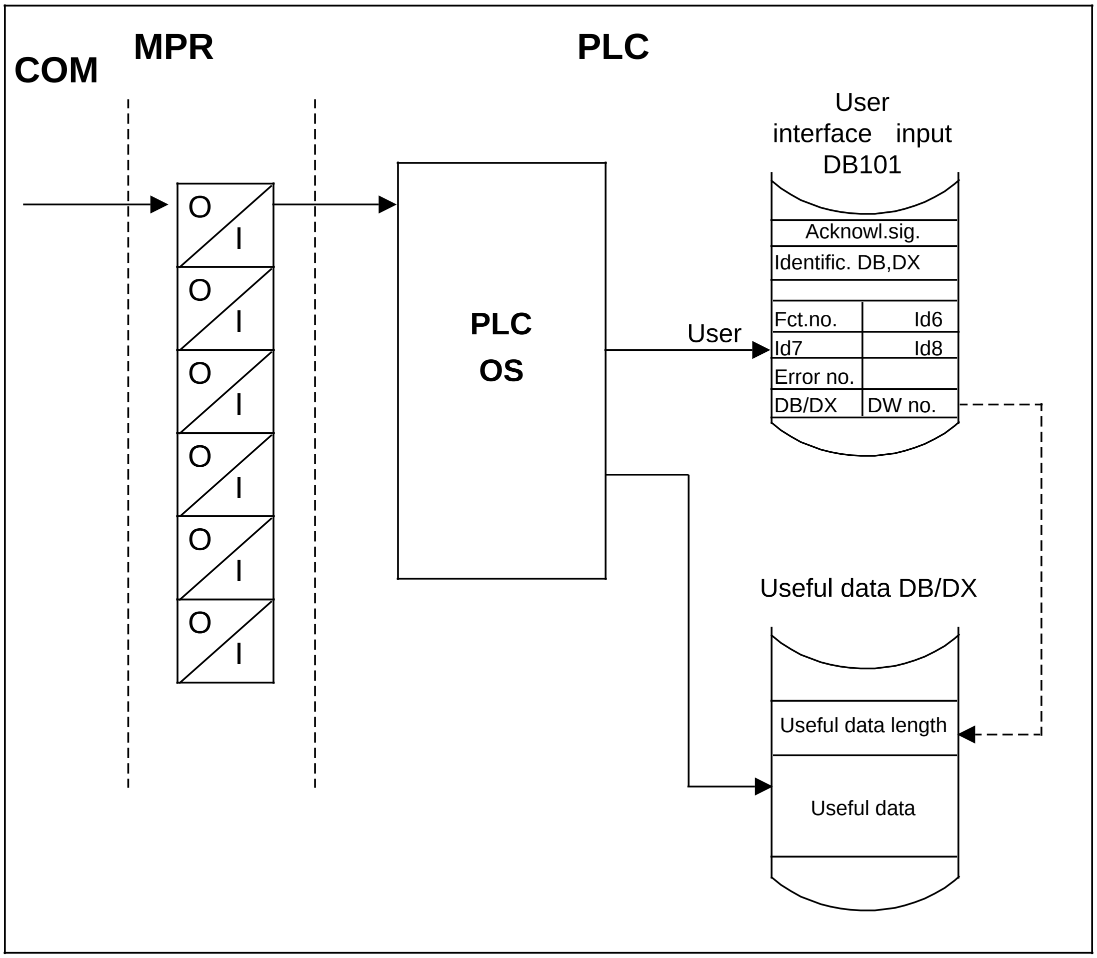
The computer link software of the PLC is responsible for the data transfer between the user interfaces (DB 101, DB 102) and the input/output buffers in the multiport RAM.



User interfaces

In the following, data transfer from the input/output buffer of the multiport RAM to the user interface and vice versa is described:

Input buffer user interface (figure "Transfer to user interface")



Transfer to user interface

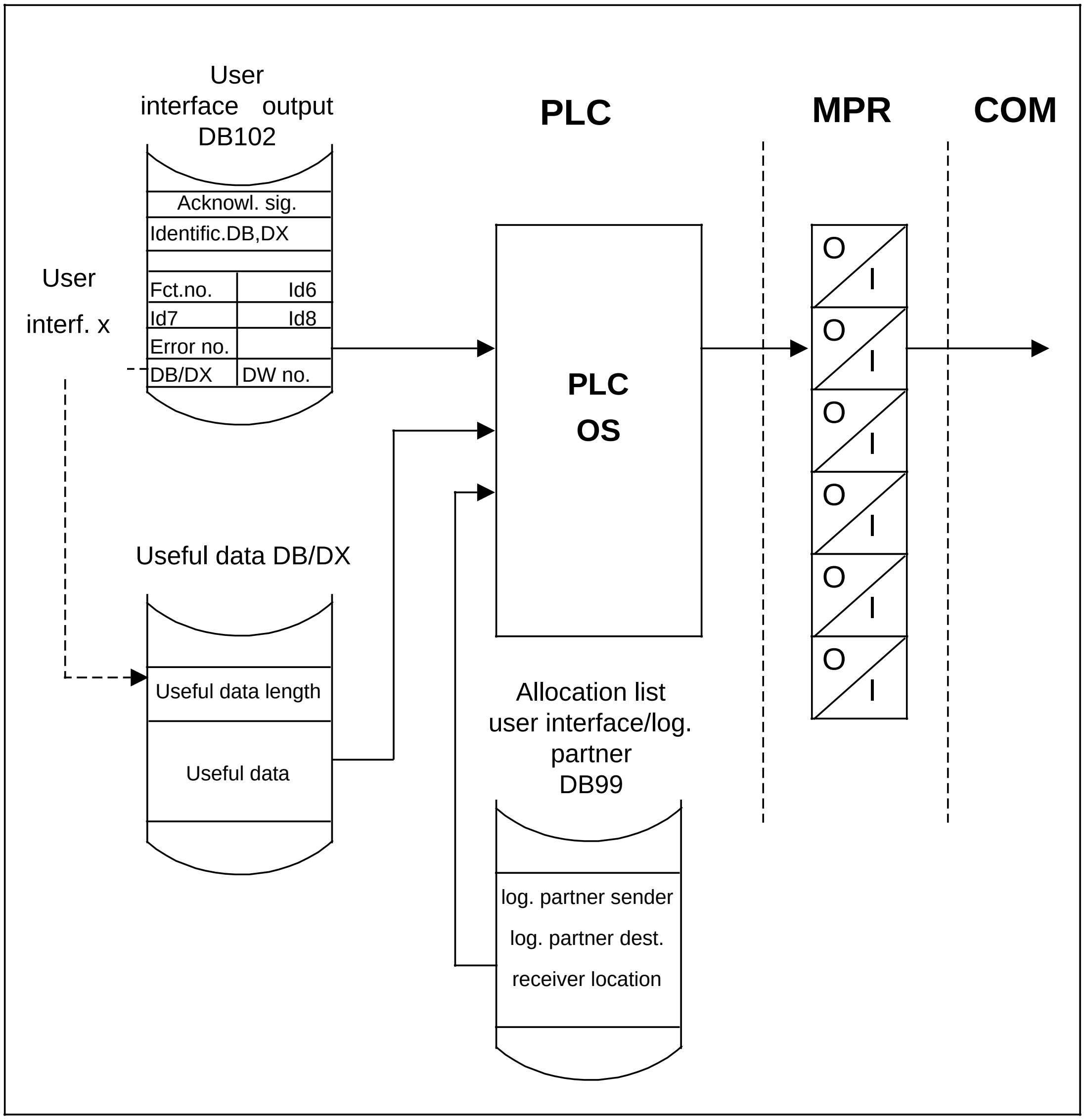
The PLC cyclically checks the 6 input buffers in the multiport RAM for incoming message frames. Assignment of the input/output buffers is dynamic, i.e. there is no fixed allocation between the buffers and the user interfaces. If valid data is in an input buffer, the buffer processing program checks whether the message frame is to be processed by the corresponding PLC and makes a distinction between configurable and standard message frames. In the case of configurable message frames, it checks whether the addressed user interface is free, i.e. whether the data can be copied.

If the user interface is free, the function number, the 6th to 8th character of the identification (identification structure see Section 6, Message frames), the error number (9th and 10th character) and the reference to the useful data from the input buffer are entered into that user interface in data block 101.

The useful data are then copied into the specified data block (DB/DX).

After all data has been copied, the acknowledgement signal for the user interface is set and the input buffer in the multiport RAM is acknowledged. The user has to release the user interface by resetting the acknowledgement signal after he/she has initiated all necessary responses to the incoming message frame.

User interface   Output buffer (Figure "Transfer from user interface")



Transfer from user interface

For output of a message frame, the user supplies the corresponding user interface in DB102, i.e. he/she enters the function number, the 6th to 8th identification character, the error number and the reference to the useful data. Thereafter, he/she sets the acknowledgement signal "output" for the corresponding user interface.

The PLC operating system evaluates the acknowledgement signal and transmits the data into a free output buffer in the MPR. The useful data are copied by the useful data DB/DX into the data field of the output buffer. In order to generate the addressing data, the specifications of the allocation table in data block 99 are loaded into the output buffer.

By setting the data validity bit it is indicated to the COM area that a message frame to be sent is waiting in the output buffer.

If the message frame was able to be be sent, the acknowledgement signal output is deleted and the user interface cleared.

5.1.2 Standard functions

Standard functions are only processed in one PLC. The corresponding PLC is activated via a machine data.

On input by the computer link basic software, standard function message frames are copied by the MPR into the DB 34 input buffer (figure "Transfer of standard message frames"). Decoding and processing of the message frame are done subsequently by the STEP 5 standard function blocks.

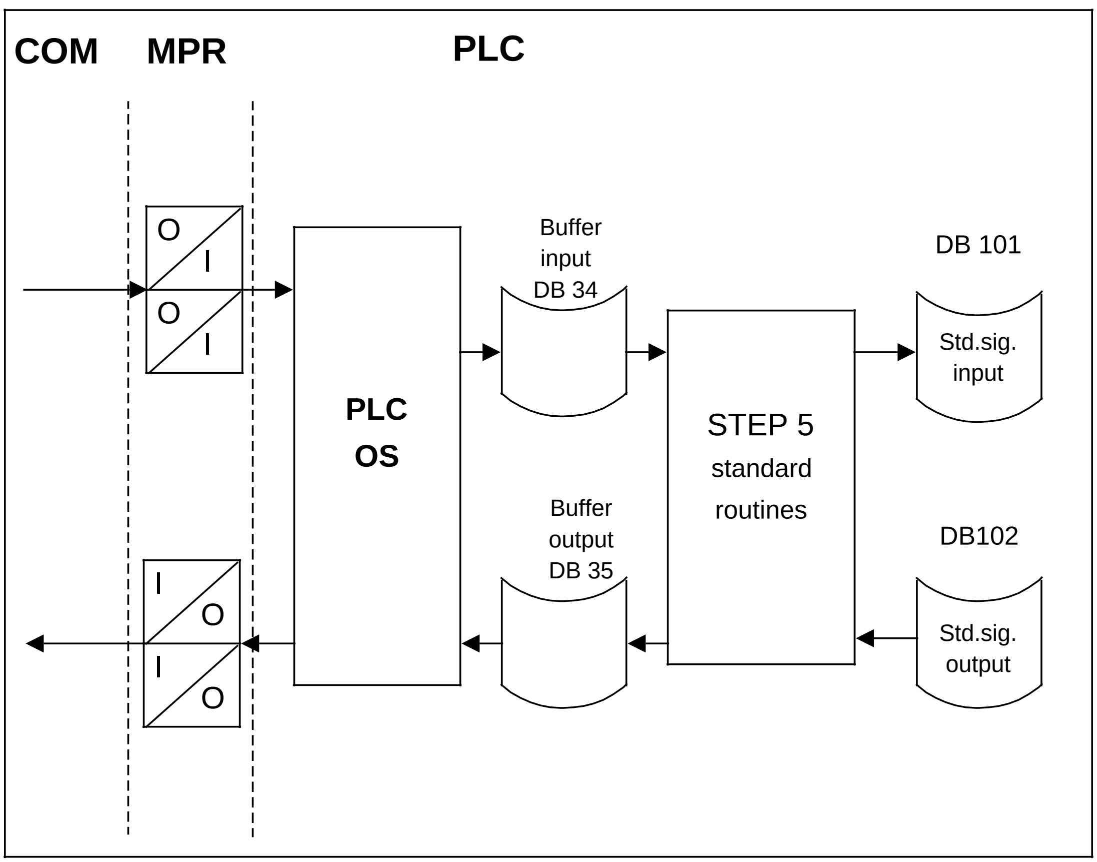
The processing result is entered as a signal in the standard area of the user interface for the input (DB 101, DW 0 to 15).

All other PLCs which do not process the standard functions must retrieve the information from the buffer for the standard signals in the MPR and copy them into their standard interfaces for input.

Output to the production control computer from the user interface standard area is only possible by the PLC which is determined by machine data for processing of the standard message frames.

The output from the standard interface can take place on initiative of the user (action) or as response of the user to the production control computer's initiative.

For the output, the message frame and addressing data are processed due to a signal in the standard output interface in DB 102. Then this block is entered by the STEP5 standard function blocks into the output buffer DB 35. The OS part of the PLC-CL basic software transfers the message frame from the output buffer for the STEP 5 standard function blocks into a PLC output buffer (in the MPR).



Transfer of standard message frames

Kernel sequences

Message frames for kernel sequences are a subset of the standard function message frames.

Kernel sequences are fixed message frame sessions which cannot be changed by the PLC user program.

With SINUMERIK 8X0 the tool sessions are executed via kernel sequences.

Therefore kernel sequences are appropriate only when the tool management package for SINUMERIK 8X0 is used.

The kernel sequences for the tool sessions are processed in one PLC. Via MD is specified in which PLC the kernel sequences for the tool sessions are executed. The DBs for the magazine assignment must also be provided in this PLC.

The message frames of the kernel sequences for the tool sessions are transferred to the processing PLC using the addressing data "receiver location" which is configured in input list 1 of the correponding interface module.

The following functions are implemented for the tool sessions:

- Inquiry of all tools
- Inquiry of tools used
- Reporting a tool
- Random loading/unloading

6 Message Frames

Message frames are data blocks which are exchanged between the nodes. The message frame structure is shown in figure "Protocols for computer link".

Procedure start	Protocol header		Message frame data			Proc. termination
SINEC H1 (Bus connection)	AP 1 header 20 bytes	Message frame length 2 bytes	Identification 8 bytes	Error number 2 bytes	Useful data [2]	SINEC H1 (Bus connection)
3964R/LSV2 (Serial connection)	AS 512 header[1] 10 bytes	Message frame length 2 bytes	Identification[1] 8 bytes	Error number 2 bytes	Useful data [2]	3964R/LSV2 (Serial connection)
XON/XOFF, RTS-LINE, TRANSPARENT (Serial connection)	————		Useful data (max. 244 bytes)			XON/XOFF, RTS-LINE, TRANSPARENT (Serial connection)

Protocols for computer link

A message frame consists of the procedure start and procedure termination, the protocol header and the message frame data.

With the bus connection, the procedure start and termination of the SINEC H1 procedure are used as well as the AP 1.0 header as protocol header. If connection is via serial interfaces, either the 3964R, LSV2, XON/XOFF, RTS-LINE or TRANSPARENT procedure can be used.

The protocol header of the serial connection is identical with that of the AS512 header of the S5 control.

For a more detailed description of the procedures and protocols, please refer to Section 3, Interface Modules.

Message frame data

The message frame data is described in greater detail below.

In the case of a bus connection, message frames have the following structure:

- Identification
- Error number
- Useful data

1) *configurable in interface list*
with/without AS 512 protocol
with/without System 800 identification

2) *max. 244 bytes, when connected to SIEMENS FMS SYSTEM max. 224 bytes*

In the case of a serial connection, the message frames in the 3964R and LSV2 procedures consist of either

- Identification
- Error number
- Useful data (same structure as with the bus connection)
  or only of
- useful data.

The message frames in the procedures XON/XOFF, RTS-LINE and TRANSPARENT are always useful data only (without identification and error number).

If the message frames consist of useful data only, the identification bytes are used in another way. They are used to communicate between the interface module and the user interface, but are not transmitted over the line.

Whether communication takes place with or without identification and error number in the case of 3964R and LSV2 is determined in the configuration stages. The parameter ”System 800 ID: with/without” in the interface list is used for this purpose. The parameters in the interface list may be configured for each individual interface.

The identification is required for SINUMERIK-internal addressing and function identification.

It consists of 8 ASCII characters.

In all message frames a spaceholder for an error number is provided between identification and useful data. The error number is also called identification no. 9 and 10.

The useful data length of the message frames is limited to 244 bytes.

Note:

For reasons of compatibility a maximum message frame length of 224 bytes must be used with links to Siemens FMS systems or SINUMERIK 850. The maximum message frame length can be set via machine data.

In case of larger data volumes, subsequent message frames are sent until all data has been transmitted. The message frames defined for the computer link with SINUMERIK 8X0 (see Section 6.5, Message frame overview) cover the most important functions for a large number of machine types.

Furthermore, it is quite simple to define or to configure additional message frames which are tailored specifically to the requirements of the customer.

There are three types of message frames:

- Standard message frames
- Kernel sequences
- Configurable message frames

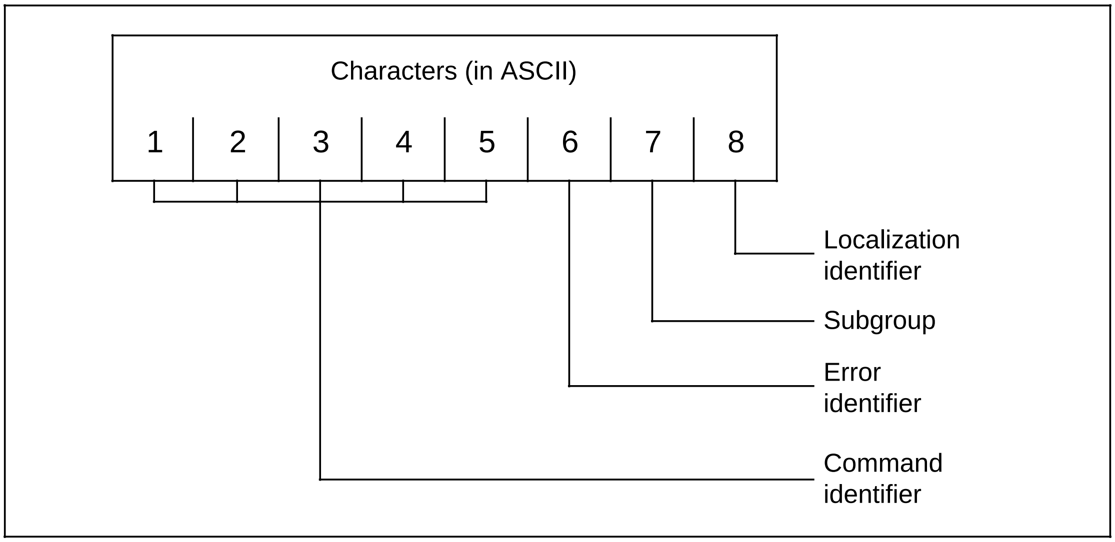
The sessions required for manufacturing can be compiled by concatenating the message frames.

6.1 Identification

The identification of the message frames is described hereunder in greater detail in order to explain the tables in Section 6.5, Message frame overview. Please refer to the Description "SINUMERIK 850/880 Computer Link/Description of Message Frames" for further information on message frame structure and sessions.

The message frame identification always consists of 8 characters (format = ASCII).

6.1.1 Meaning of the characters in the message frame identification for SINUMERIK message frames



Identification structure

Command identifier (characters 1 to 5)

1st character:
The first character defines the type of command

C (Command)	=	Command call
Q (Acknowledgement)	=	Acknowledgement of an executed job
P (Prepare)	=	Request to prepare
R (Receive)	=	Request to receive data
T (Transmit)	=	Request to send data

2nd to 5th character:

The next 4 characters can be selected freely and should represent an abbreviation of the message frame function, as e.g. R_WBC for "load tool with code carrier".

This facilitates an allocation between identification and meaning of the message frame, which is especially advantageous during installation and for debugging.

Error identifier (character 6)

F (error) =	If a received message frame cannot be processed or contains an error, the initiating message frame is returned to the sender (production control computer or PLC) with the identification "F" as 6th character.

otherwise blank (space)

Subgroup identifier (character 7)

Within a session it must be possible to clearly distinguish between the individual message frames. With the 7th character of the message frame header a session step is defined with which specific message frames repeatedly occurring within a session are additionally identified. In this way, the message frame becomes unambiguous within the session.

7th character:

A (Start)	=	Start of a session
B (Machining stat.)	=	Arrival of a pallet at the machining station
E (end)	=	End or temporary end of a session
G (Off)	=	Event off
K (On)	=	Event on
M (Machine)	=	Arrival of a pallet at the machine

otherwise blank (space)

Localization identifier (character 8)

In the case of message frames for specific mode groups, the mode group is defined by means of the 8th character. If the message frame shall apply to all mode groups, 0 is entered as 8th character.

When alarms are reported, identification "P" is specified as the 8th character for PLC alarms or "N" for NC alarms resp.

8th character:

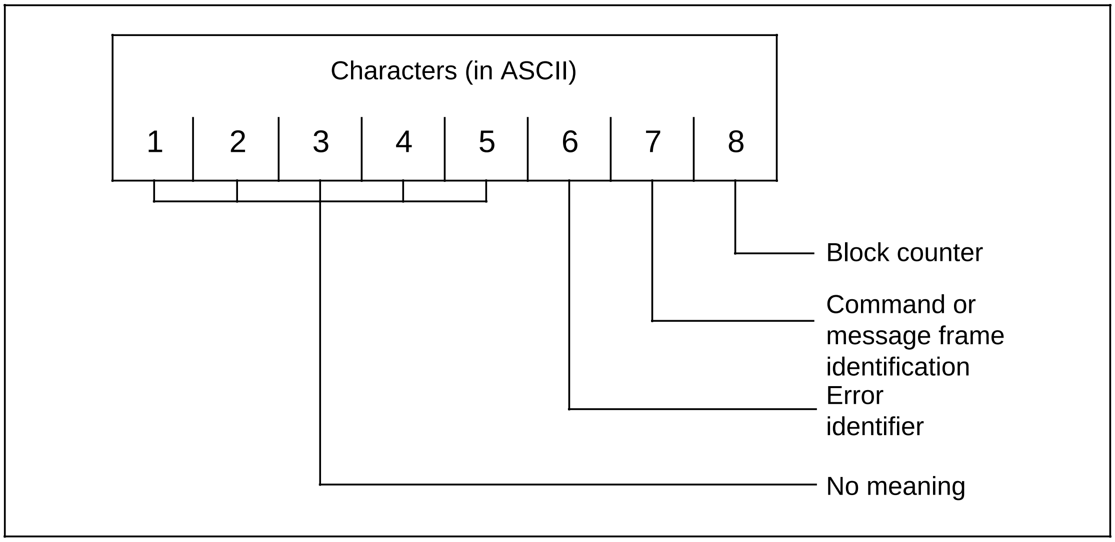
g	=	Number of the mode group (1...8, 0=all mode groups)
N(NC alarms)	=	Reporting NC alarms
P(PLC alarms)	=	Reporting PLC alarms

A blank (space character) is used as 8th character for message frames for which the allocation to mode groups makes no sense, such as message frames for connection monitoring (T_OK____)

6.1.2 Meaning of the characters used in the identifications of "free message frames"

"Free message frames" are user message frames that do not comply with the System 800 message frame specification, i.e. there is no message frame identification.

The identification bytes of "free message frames" are used in a different manner. They are used for communications between the interface module and the user interface but are not transmitted via the link.



Structure of the identification in "free message frames"

Characters 1 to 5 are not used.

Error identifier (character 6)

F (error) = If a command or message frame is not valid or contains errors, it is returned to the transmitter with identification "F" entered as the 6th character.

Command or message frame identifier (character 7)

For "free message frames", a distinction is made between commands and message frames. Commands are used in the first place for the administration of the interfaces (reserving/enabling the interface) or for specifically triggering the connected station ("DATA START"). Commands are generally issued by the user.

7th Character

B= Command "Reserve interface"

Prior to the actual data transfer, the user must reserve the interface and include the parameters for addressing in the reverse direction in the useful data before transferring the data to the interface module. These parameters are "receiver type, address of user interface, the DB/DX address, DW address".

F= Command "Enable interface"

On completion of a data transfer, the user must enable the interface to provide access for other users, too. In the process, the address data stored for this interface are deleted on the interface module.

The command requires no useful data.

D= Command "DATA START"

For the XON/XOFF and RTS-LINE procedures (configuration: node states equal: NO), data input must be initiated by the user via the "DATA START" command.

The command requires no useful data.

E = Single message frame (valid both in input and output direction)

N = Initial message frame of a message frame sequence (valid in input direction only)

M = Subsequent message frame of a message frame sequence (valid in input direction only)

I = Terminating message frame of a message frame sequence (valid in input direction only)

Block counter (character 8)

The block counter is operated by the procedure driver. All message frames received are numbered consecutively. The block counter enables the user to join the blocks of a message frame sequence in the correct sequence and thus detect the loss of any block. The counter is implemented as a ring counter, i.e. a message frame sequence need not begin with 0.

6.2 Standard message frames

Standard message frames are those message frames which are processed by SINUMERIK 880 standard software in the COM or PLC area.

Standard message frames are:

- Event message frames
- Message frames for connection monitoring
- Mode switchover message frames
- Restart message frame
- Message frames for synchronization
- Message frames for file transfer

On receipt of a standard message frame a signal to the user interface is provided if required.

Some standard message frames have to be initiated via the standard interface (example: Operator interrupt reporting ON/OFF).

6.3 Kernel sequences

Kernel sequences are defined message frame sessions between host computer and PLC which are executed autonomously on the PLC side via the standard function blocks. Since the kernel sequences are processed by the standard, these sessions are fixed and cannot be changed by the PLC user program.

The message frames of the kernel sequences represent a subset of standard message frames. Initiation of some kernel sequences is via the user interface "output".

If a message frame is acknowledged within a kernel sequence with an error, this results always in the kernel sequence being aborted.

The tool data is entered through kernel sequences into the corresponding tool memories of the SINUMERIK 8X0 on "random loading".

Use of the kernel sequences is only appropriate when the tool management software package is integrated in the SINUMERIK 8X0 system.

6.4 Configurable message frames

Configurable message frames are those message frames the data and information of which must be processed in the PLC user program in the case of incoming message frames or which have to be provided by the PLC user program in the case of message frames to be sent.

Incoming message frames are communicated to the user via the "input" user interface. The transmitted data is stored in a specific data area.

Initiation for transmitting a message frame is via the "output" user interface. The data to be transmitted has to be provided in specified data areas.

6.5 Message frame overview

The following tables contain an overview of message frames, covering all identifications currently agreed which are required with FMS systems between the SINUMERIK 8X0 (CNC) and the production control computer.

Furthermore, message frames are listed which are exchanged between the SINUMERIK (CNC) and subordinate units such as write/read devices.

In addition to the message frame identification the following is specified in the tables:

- the direction of the message frame with the aid of an arrow which always points from the transmitter to the receiver of the respective message frame.
- the type of message frame with the aid of the symbols
  S, K and P:

  S = Standard message frame
  K = Message frame of a kernel sequence
  P = Configurable message frame

Function	Identific-ation	Prod. ctrl CNC	Message fr. type
Partner: Computer			
Message frames for operator-process communication			
Computer link cold restart by CNC			
Computer link cold restart by CNC	R_NS___		S
Synchronization			
Start of synchronization	T_SY___		S
Rejection/abort by CNC (neg. acknowledgement)	T_SY_F_		S
Transmission of status data (pos. acknowledgement)	R_SY___		S
End of synchronization message	T_SY_E_		S
Abort by production control	R_SY_F_		S
Interrogation of "all" pallets at machine			
Pallet data request	T_SP__g		P
Rejection by CNC (neg. acknowledgement)	T_SP_F_g		P
Pallet data transmission (pos. acknowledgement)	R_SP__g		P
Data of last pallet transmitted (end)	R_SP_Eg		P
Abort of pallet data transmission by production control	R_SP_F_g		P
End-of-work			
Action: Change end-of-work status	C_FA___		S
Response: Rejection by CNC (neg. acknowledgement)	C_FA_F_		S
End-of-work status changed (pos. acknowledgement)	Q_FA___		S
Pallet arrival messages			
Action: Pallet delivered to machine	R_PN_Mg		P
Response: negative acknowledgement	R_PN_FMg		P
positive acknowledgement	Q_PN_Mg		P
Pallet delivered to machining station	R_PN_Bg		P
NC program selection/start/end message			
Action: NC program selection with start enable	C_CS_Ag		P
Response: negative acknowledgement	C_CS_FAg		P
positive acknowledgement	Q_CS_Ag		P
Report start of machining	R_CS_Ag		P
Report end of machining	R_CS_Eg		P
Remove pallet from machining station			
Action: Remove pallet	C_PA__g		P
Response: negative acknowledgement	C_PA_F_g		P
positive acknowledgement	Q_PA__g		P
Event message frames			
NC alarms on	R_AL_KN		S
PLC alarms on	R_AL_KP		S
All alarms off	R_AL_G_		S
Operator interrupt on	R_BU_K_		S
Operator interrupt off	R_BU_G_		S
Status message on	R_BM_K_		S
Status message off	R_BM_G_		S
Continuation message	R_FS___		S

Function	Identific-ation	Prod. ctrl CNC	Message fr. type
Connection monitoring			
Test message frame from production control	T_OK__		S
Acknowledgement by CNC	R_OK__		S
Production control computer mode switchover			
by production control	C_MO__		S
by CNC (activate manual FMS mode)	R_MO__		S
by CNC (manual FMS mode ended)	R_MO_E_		S
Tool message frames			
Report all tool magazine assignments			
Tool block request	T_TG__		K
Tool block request (neg. acknowledgement)	T_TG_F_		K
Tool block transmission (pos. acknowledgement)	R_TG__		K
End of session message: all available tools reported	R_TG_E_		K
Session abort by production control	R_TG_F_		K
Report all tools used			
Tool block request	T_TQ__		K
Tool block request (neg. acknowledgement)	T_TQ_F_		K
Tool block transmission (pos. acknowledgement)	R_TQ__		K
End of session message: all tools used reported	R_TQ_E_		K
Session abort by production control	R_TQ_F_		K
Reporting a tool			
Reporting a tool with data 1st cutting edge	R_TS__		P
Report a tool all cutting edges			
Report data of first cutting edge	R_TS__A_		P
Positive acknowledgement from production control	T_TS___		K
Negative acknowledgement from production control	R_TS__FA_		K
Report data of the next cutting edge	R_TS____		K
Negative acknowledgement from PLC	T_TS__F_		K
Report data of the last cutting edge	R_TS__E_		K
Production control acknowledges positively receipt of data last cutting edge	T_TS__E_		K
Random loading without code carrier			
Loading initiation - request tool load	T_WB_A_		P1)
Rejection by production control (negative acknowledgem.)	T_WB_FA_		K
Transmit tool data (positive acknowledgement)	R_WB__		K
Request tool data next cutting edge	T_WB__		K
Transmit tool data last cutting edge	R_WB_E_		K
Acknowl.: Receipt of the tool data last cutting edge	T_WB_E_		K
Session abort by CNC	R_WB_F_		K
Session abort by production control	T_WB_F_		K
1) necessary for the kernel sequences that follow			
Random loading - report loaded tool			
Tool is loaded	Q_WB__		P
Tool is not loaded	Q_WB_F_		P

Function	Identific-ation	Prod. ctrl CNC	Message fr. type
Random loading with code carrier			
Transmit tool data 1st cutting edge	R_WBC_A_		K
Rejection by production control (neg. acknowledgement)	R_WBCFA_		K
Request tool data of next cutting edge	T_WBC__		K
Transmit tool data next cutting edge	R_WBC__		K
Transmit tool data of last cutting edge	R_WBC_E_		K
Acknowl.: Receipt of the tool data last cutting edge	T_WBC_E_		K
Session abort by CNC	T_WBCF_		K
Session abort by production control	R_WBCF_		K
Random unloading with/without code carrier			
Unloading initiation - report tool data first cutting edge	R_WE_A_		P1)
Rejection by production control (negative acknowledgement)	R_WE_FA_		K
Request tool data next cutting edge (pos. acknowl.)	T_WE___		K
Report tool data next cutting edge	R_WE___		K
Report tool data last cutting edge	R_WE_E_		K
Acknowledgement: Receipt of tool data last cutting edge	T_WE_E_		K
Session abort by CNC	T_WE_F_		K
Session abort by production control	R_WE_F_		K
1) necessary for the kernel sequences that follow			
Random unloading - report unloaded tool			
Tool is unloaded	Q_WE___		P
Tool is not unloaded	Q_WE_F_		P
Load magazine assignment data			
Production control transmits the data first cutting edge	R_MB_A_		K
Positive acknowledgement from CNC	T_MB_A_		
Negative acknowledgement from CNC and session abort	R_MB_FA_		K
Production control transmits the data next cutting edge	R_MB___		K
End of session, data last cutting edge are transmitted	R_MB_E_		K
Positive acknowledgement from CNC and session abort	T_MB_E_		K
Negative acknowledgement from CNC and session abort	R_MB_FE_		K
Abort message frame, so that production control can abort session at any time	T_MB_F_		K
Tool cartridge sessions Message frames			
Tool cartridge sessions			
1. Loading list			
• Production control transmits the assignment for the tool cartridge to CNC	R_TL___		K
• Acknowledgement from CNC	Q_TL___		K
• Production control transmits the last message frame with cartridge assignment data to CNC	R_TL_E_		K
• Acknowledgement loading list received in full	Q_TL_E_		K
2. Unloading list			
• Production control transmits unloading list to CNC	R_TU___		K
• Acknowledgement from CNC	Q_TU___		K
• Production control transmits the last unloading list message frame	R_TU_E_		K
• Acknowledgement unloading list received in full	Q_TU_E_		K

Function	Identific-ation	Prod. ctrl CNC	Message fr. type
Tool cartridge sessions Message frames			
Tool cartridge sessions			
3. Machining enable of a tool cartridge			
• Production control issues machining enable	C_TE___		P
• Negative acknowledgement from CNC	C_TE_F_		P
• Positive acknowledgement from CNC	Q_TE___		P
4. Transmission of cartridge assignment to production control			
• CNC transmits cartridge assignment to production	R_TP___		K
control	Q_TP___		K
• Positive acknowledgement from production control	R_TP_F_		K
• Negative acknowledgement from production control	R_TP__E_		K
• CNC transmits last message frame of cartridge assignment to production control			
• Positive acknowledgement from production control	Q_TP__E_		K
• Negative acknowledgement from production control	R_TP_FE_		K
Message frames for the NC file transfer			
Transmit NC program to CNC (production control CNC)			
Request to load from production control			
Production control requests ready-to-receive state	PRaaa__		S
Rejection by CNC (negative acknowledgement)	PRaaaF_		S
CNC acknowledges ready-to-receive state positively or	TNaaa__		S
requests transmission of the next data block			S
NC program data transmission to CNC	RNaaa__		S
Message: End of NC program transmission	RNaaa_E_		S
Acknowledgement for final message: negative	RNaaaFE_		S
Acknowledgement for final message: positive	REaaa__		S
Abort by production control	TNaaaF_		S
Abort by CNC	RNaaaF_		S
Request to load by operator via screen form			
Request to production control to send NC data	TDaaa__		S
Rejection by production control (negative acknowledgem.)	TDaaaF_		S
Retransmit NC program to production control (CNC Production control			
Request from production control			
Production control computer requests CNC to send NC data	PTaaa__		S
NC program data transmission to production control	RNaaa__		S
Rejection by CNC	PTaaaF_		S
Request to transmit next data block	TNaaa__		S
Message: End of the NC program transmission	RNaaa_E_		S
Acknowledgement for final message: negative	RNaaaFE_		S
Acknowledgement for final message: positive	REaaa__		S
Abort by CNC	TNaaaF_		S
Abort by production control	RNaaaF_		S

Function	Identific-ation	Prod. ctrl CNC	Message fr. type
Request by operator via screen form			
Request to production control to receive NC data	RDaaa__		S
Rejection by production control (negative acknowledgem.)	RDaaaF_		S
Rejection by CNC	PTaaaF_		S
Request to send next data block	TNaaa__		S
Message: End of NC program transmission	RNaaa_E_		S
Acknowledgement for end message: positive	RNaaaFE_		S
Acknowledgement for end message: negaitive	REaaa__		S
Abort by CNC	TNaaaF_		S
Abort by production control	RNaaaF_		S
Notes:			
aaa = MPF (Main Program File)			
aaa = SPF (Sub Program File)			
aaa = ZOA (Zero Offset Active)			
aaa = RPA (R Parameter Active)			
aaa = TE1, TE2, TE4 Machine Data			
aaa = TOA Tool Offsets			
aaa = SEA, SE4 Setting Data			
If the transmission initiative is taken by the computer, the complete scope of functions can be used.			
Transmission of main programs (MPF) and subroutines (SPF) can be requested by the production control computer or by the operator.			
NC erase program			
Production control requests ready-to-receive state for erase program	PRCLF__		S
Rejection by CNC (negative acknowledgement)	PRCLFF_		S
CNC requests transmission of erase program	TNCLF__		S
Rejection by production control	TNCLFF_		S
Transmission of erase program to CNC	RNCLF__		S
End-of-transmission message for erase program	RNCLF_E_		S
Abort by production control	TNCLFF_		S
Abort by CNC	RNCLFF_		S
Acknowledgement to NC erase program			
NC erase program successfully executed	RNKOM_		S
NC erase program not executed	RNKOMF_		S

Function	Identific-ation	Coding station	CNC	Message fr. type
Message frames for writing and reading pallet data carriers				
Partner: Pallet data carrier system				
Read data carrier				
CNC requests reading of pallet data				P
Rejection by data carrier system				P
Transmit pallet data to CNC				P
Rejection by CNC				P
Write data carrier				
CNC transmits the pallet data to be written to the data carrier system	R_CW___			P
Rejection by data carrier system	R_CW_F_			P
Pallet data written on data carrier	T_CW___			P
Rejection by CNC	T_CW_F_			P
Reset data to carrier				
CNC aborts pending command	C_CR___			P
End session with a data carrier				
CNC requests session end with a data carrier	C_CN___			P
Rejection by data carrier system	C_CN_F_			P
Data carrier system ends session for a data carrier	Q_CN___			P
Rejection by CNC	Q_CN_F_			P
Message frames for writing on and reading of tool code carriers				
Partner: Tool code carrier system				
Read data carriers				
CNC requests reading of tool data	T_CB___			P
Rejection by code carrier system	T_CB_F_			P
Code carrier system transmits tool data	R_CB___			P
Rejection by CNC	R_CB_F_			P
Write data to carrier				
CNC transmits tool data to be written to the code carrier system	R_CE___			P
Rejection by code carrier system	R_CE_F_			P
Tool data written on code carrier	T_CE___			P
Rejection by CNC	T_CE_F_			P
Delete data carrier				
Delete trigger from data carrier	R_CL___			P
Positive acknowledgement	T_CL___			P
Negative acknowledgement	R_CL_F_			P

6.6 Addressing

Bus connection and connection via serial interface feature a similar addressing concept. The only differences are in the interpretation of the procedure and protocol header data on the respective interface module. The addressing via message frames is identical with both connections if the serial interface is configured with ”with system 800 ID”. In the following, the addressing philosophy shall be explained by means of a message frame which is transferred to the user interface. The figure "Addressing" shows schematically the route of the message frame and the evaluation of the message frame headers.

The information of the lower OSI layers is contained in the procedure header of the message frames. With the bus interface, these are the Ethernet and TSAP address on the basis of which the message frame is routed through the local area network to the interface module, up to layer 4. In the procedure header for the serial interface module, addresses are not necessary since an unambiguous hardware allocation is made by the interface used.

Logical partners are assigned to the address information in the protocol header on the interface module. In case of the bus interface module, this is done by the AP monitor whereas in case of the serial interface module the device management is responsible for this task. The necessary information is stored in lists.

If no protocol header is configured for a serial interface (without AS 512 protocol) the logical partner is determined solely by the information from the addressing list (interface number logical partner). In this case there is only one logical partner per interface. The logical partner recognized from the address information is transferred to the dual port RAM. The procedure header and the protocol header completely terminate on the interface module on input and are generated on the interface module on output.

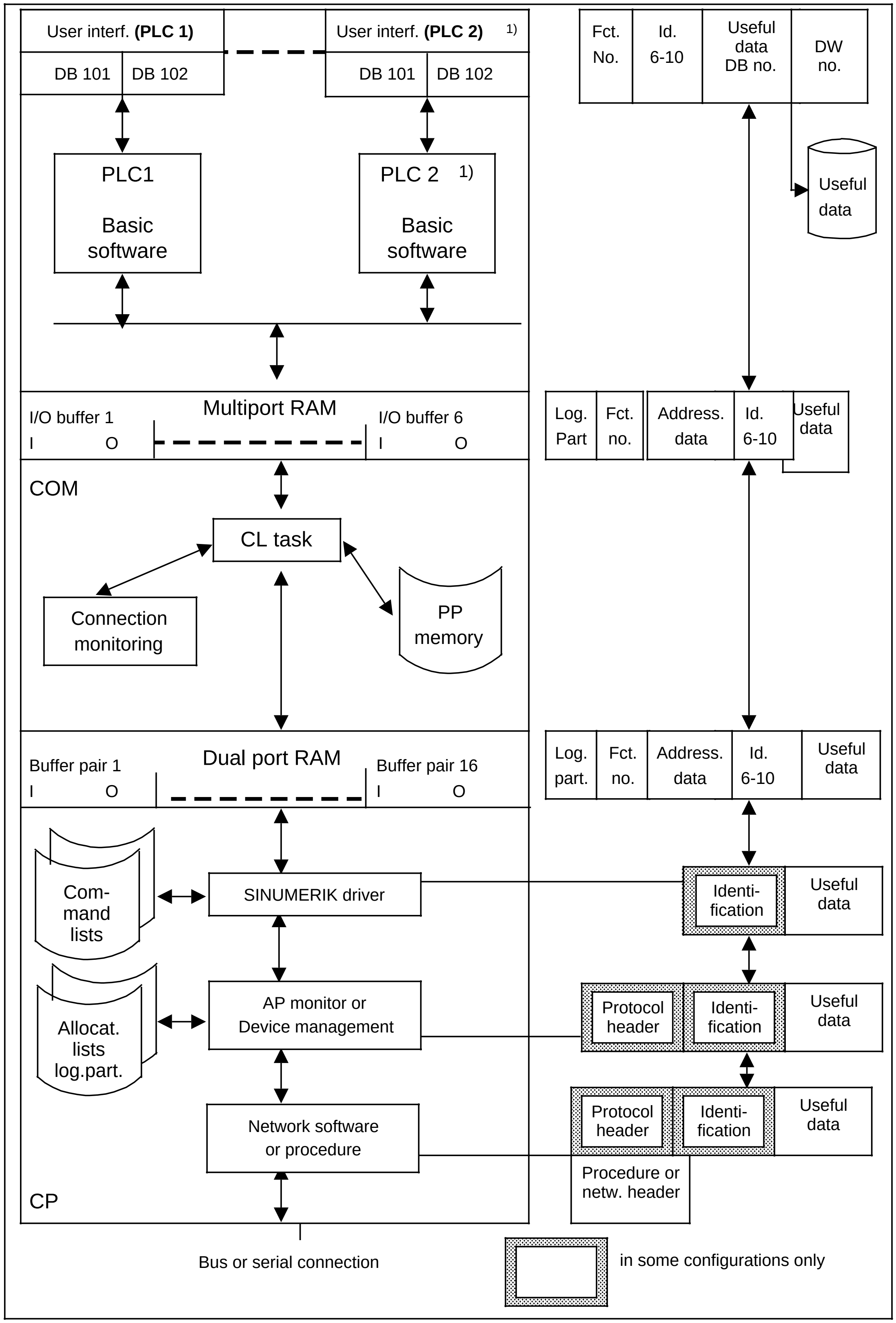
The SINUMERIK driver takes over the remaining addressing including the entry in the dual port RAM. In the case of bus connections and serial connections which are configured with System 800 identification, the SINUMERIK driver generates the function number from the first 6 characters of the identification. It evaluates the remaining SINUMERIK-specific address information on the basis of allocations stored in lists and enters the data in dual port RAM.

If a serial interface is configured without System 800 identification, a special command must be assigned to it before data exchange. After data exchange, it is enabled once more by another command. When a serial interfrace is assigned the addressing data are transferred to the SINUMERIK driver, which stores them in an internal list. In the event of data exchange the SINUMERIK driver examines the status of the interface (assigned/unassigned) and the message frame length and stores the message frame data in dual port RAM. Identification bytes 6 to 10 are used for transmitting the command, message frame and error information between the CP 315 and the user interfaces. In this way the structure of dual port RAM can remain intact. Above this layer there is no difference in addressing between serial connection and bus connection.The COM forwards the message frame on the basis of the addressing data in the dual port RAM, in this case to the PLC or to the multiport RAM.

The addressed PLC recognizes that a message frame has been stored for it in the multiport RAM. It transmits the data necessary for the user to the user interface addressed and stores the useful data in the specified useful data block. The following data is transmitted to the user interface:

- Function number, characters 6 to 10 of the identification,
- Useful data DB no. and
- Data word no.

The function number is not used for message frames without System 800 identification with a serial interface and therefore has a default value of 0.



Addressing

---

1) *only for SINUMERIK 880*

7 Configuration and Test

The SINUMERIK 8X0 computer link uses an identical configuration and test concept for both bus and point-to-point connection.

The connections and the SINUMERIK-specific message frames are defined on configuration. The configuring data is stored in lists on the interface module.

It is possible to record for test purposes message frames within the SINUMERIK or on the transmission medium - bus cable or serial connection cable.

Furthermore, test devices are available with which either the control system or the computer part in sessions between control system and computers can be simulated.

7.1 Configuration

The following table gives an overview of the available configuration means which support configuration of the bus interface module and the serial interface module. They are used to define the connections and to generate the SINUMERIK-specific lists.

Config. means	Config. device	Application		Task
		CP 231	CP 315	
SINEC NM (Network Management)	PG685T	X		Configuration of the SINEC interface modules in the local area network via the SINEC H1 bus from the central configuring station.
SINEC NML (Network Management Local)	PG685/ PG685T PG 750	X		Configuration of the SINEC interface module via the programmer interface of the SINEC interface module or via the SINEC H1 bus from a local configuring station.
SINPS231 (SINUMERIK Configuring service)	PG685/ PG685T PG 750	X		Generation of the SINUMERIK-specific lists (input list 1, output list 1, format list) for the bus interface module.
SINPS315 (SINUMERIK Configuring service)	PG685/ PG685T/ PC AT PG 750		X	Generation of the SINUMERIK-specific lists (input list 1, output list 1, format list) and connection-oriented lists (list of interfaces, addressing list) for the serial interface module.

7.1.1 SINEC NM

SINEC NM (SINEC Network Mangement) is a software package which is used on a central configuring station (see figure "Central configuring"). The central configuring station is a PG 685T which is connected to the SINEC H1 network.

SINEC NM supplies all network-related functions required for the configuration, optimization and monitoring of the systems integrated in SINEC and operated on SINEC H1 networks.

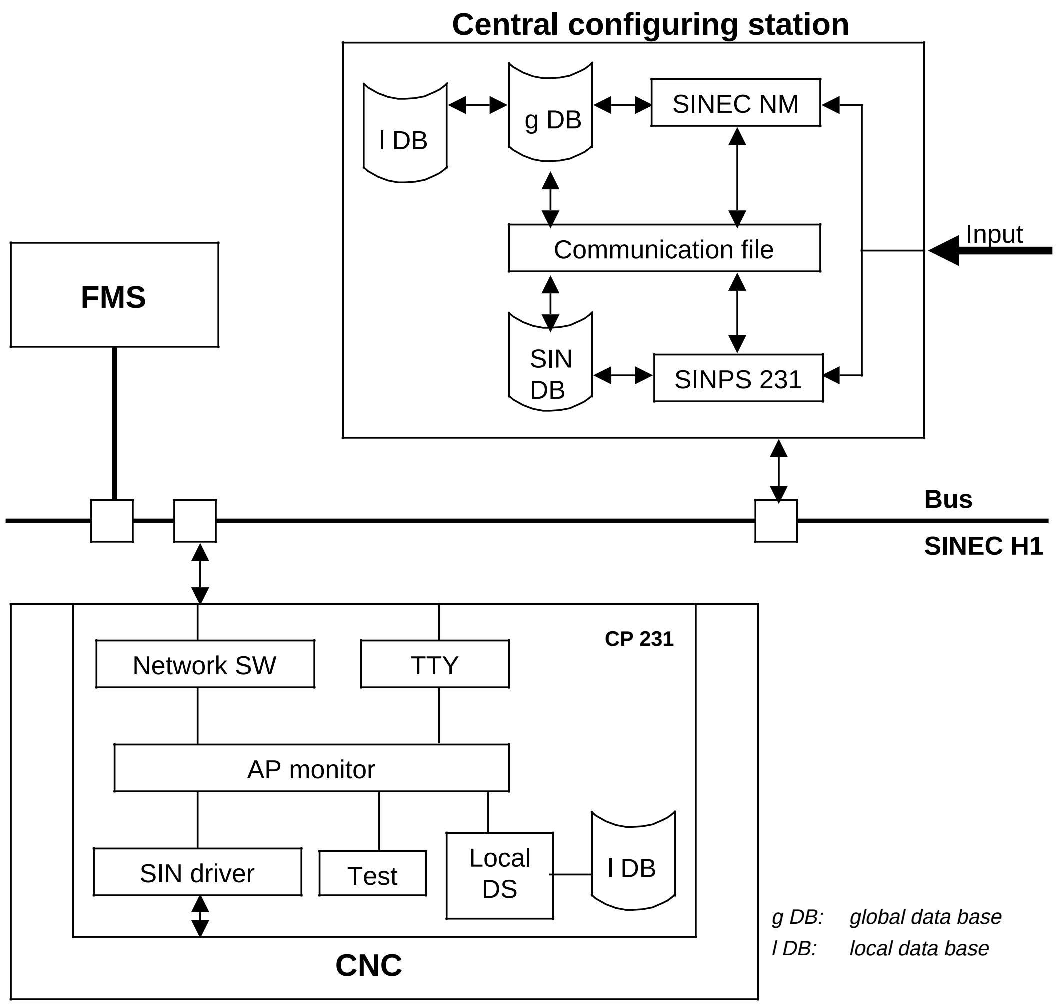
On the SINUMERIK bus interface module only the Ethernet address is set locally whereas the central station is responsible for the other configuration parameters of the interface module.

The relations between the network nodes are defined and the configuring data of the network nodes are generated from the central configuring station. The configuring data of the entire network is stored centrally in a global data base.

In addition to the global data base local data bases are generated in the central configuring station and distributed. Local data bases contain the relevant configuring data for a network node. They are structured as internally required by the node.

When generating local data bases, communication files are used as a link between SINEC configuration and system-specific configuration, e.g. SINUMERIK-specific configuration (input list 1, output list 1, format list).

Distribution of the local data bases is via the SINEC H1 bus or, alternatively, via the TTY interface.



Central configuring

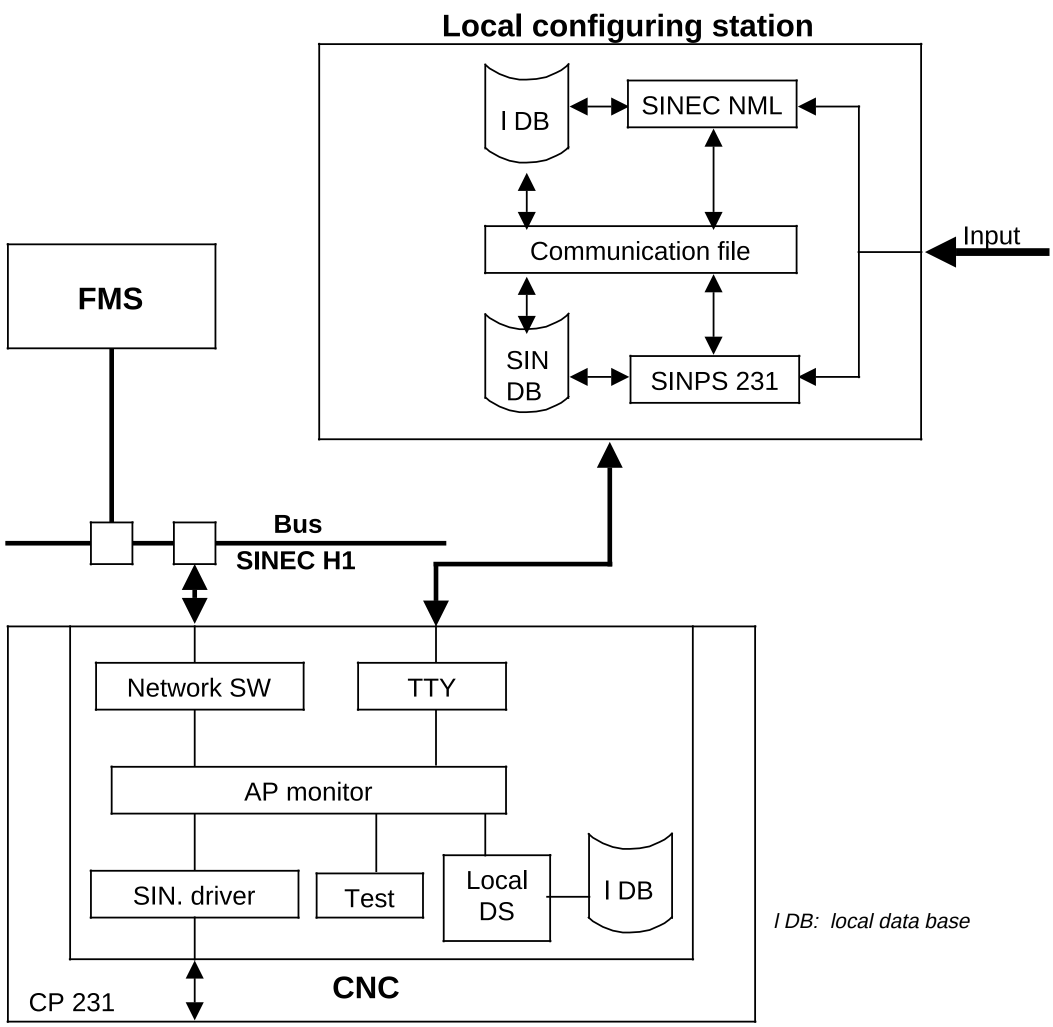
7.1.2 SINEC NML

SINEC NML (SINEC Network Management Local) is a software package which runs on a local configuring station (see figure "Local configuration"). The local configuring station is a PG 685 or PG 685 T which is connected to the bus interface module via the TTY interface or can communicate with the bus interface module via the SINEC H1 bus. With SINEC NML local data bases can be generated which are structured so that they can be read by the bus interface module of the SINUMERIK. However, SINEC NML does not create a global data base for the entire network configuration nor does it check whether e.g. identifiers have been assigned in the network more than once.

The local data base covers all connection-oriented and SINUMERIK-specific configuring data necessary for the bus interface module.

On generation of the local data base, communications files are used as link between the SINEC configuration such as definition of the connections, and the SINUMERIK-specific configuration (input list 1, output list 1, format list).

Transfer of the local data base to the bus interface module is via the TTY interface or the SINEC H1 bus.



Local configuration

7.1.3 SINPS 231

SINPS 231 (SINUMERIK configuration service for CP 231) is a software package with which the SINUMERIK-specific configuring data (input list 1, output list 1, format list) is generated. This software runs on PG 685 and PG 685 T and is used together with the SINEC NM or SINEC NML software packages (see figures "Central configuration" and "Local configuration").

Via a communication file the SINUMERIK-specific configuration is linked to the SINEC configuration.

Input list 1, output list 1 and the format list are generated by SINPS 231. In these lists the information is stored by means of which the SINUMERIK driver interprets and/or generates the SINUMERIK-specific part of the message frames.

The lists are stored on the interface module.

Input list 1

In input list 1, for incoming message frames, the characters 1 to 6 of the identification and the sub-address are related to a function number, the SINUMERIK-internal addressing data and the format identifier. The figure "Structure of input list 1" shows the entries of input list 1 as an example for two message frames of the synchronization session.

The search key in input list 1 consists of characters 1 to 6 of the identification and the sub-address.

Example: Synchronization session (automatic synchronization by production control)

	ID 1-6	Sub-address pos.	length	Max. length useful. data	Fct. no.	Cont. sub-addr.	Log. part. dest.	Locat. receiver	Type receiv.	Addr.. user	Addr DB/DX	DW	Form. ident.
Mess.fr. No. 1	T-SY	--	--	10	11	--	e.g. GLOB	PLC1	S	--	--	--	--
Mess.fr. No. 3	T-SP	--	--	14	e.g. 150	--	e.g. GLOB	PLC1	P	e.g. 20	e.g. DB240	e.g. DW10	e.g. SYNCH
	•	•	•	•	•	•	•	•	•	•	•	•	•
	•	•	•	•	•	•	•	•	•	•	•	•	•
	•	•	•	•	•	•	•	•	•	•	•	•	•

Configuration of input list 1 for two message frames of the synchronization session

S: Standard
P: User, configurable

Structure of input list 1

Output list 1

Output list 1 is relevant for message frames in output direction.

In output list 1 the corresponding characters 1 to 6 of the identification are related to the function number.

Furthermore, the repetition times and rates as well as a reference to the format identifier are included in the list. The function number is used as a search key in output list 1. The figure "Structure of output list 1" shows an example of an entry in output list 1 by means of two message frames of the synchronization session.

Example: Synchronization session (automatic synchronization by production control)

	Fct. No.	ID 1-6	TIME-INST (instruction)	TIME-RESP (response)	REPETITION flag	Format identifiers
Mess.fr. No. 2	18	R-SY--				e.g. acknowledgement
	17	T-SY-F				e.g. abort
	• • •	• • •	Default value (specified by SINEC NM or NML)	Default value (specified by SINEC NM or NML)	Default value (specified by SINEC NM or NML)	• • •

Configuration of output list 1 for two message frames of the synchronization session

Structure of output list 1

Format list

The format of the useful data is described in the format list. A cross reference to the format description in the format list is made by means of a symbol- the format identifier - in input list 1 and output list 1. The format list is required for the representation of the useful data which is checked by means of test functions.

7.1.4 SINPS 315

The SINUMERIK-specific and connection-oriented configuring data is stored on the serial interface module CP 315 in lists which have to be configured by the user.

SINPS 315 (SINUMERIK configuration service for CP 315) is a software package for the generation of the configuring data which are required by the serial interface module.

The SINPS 315 configuration software is executable on:

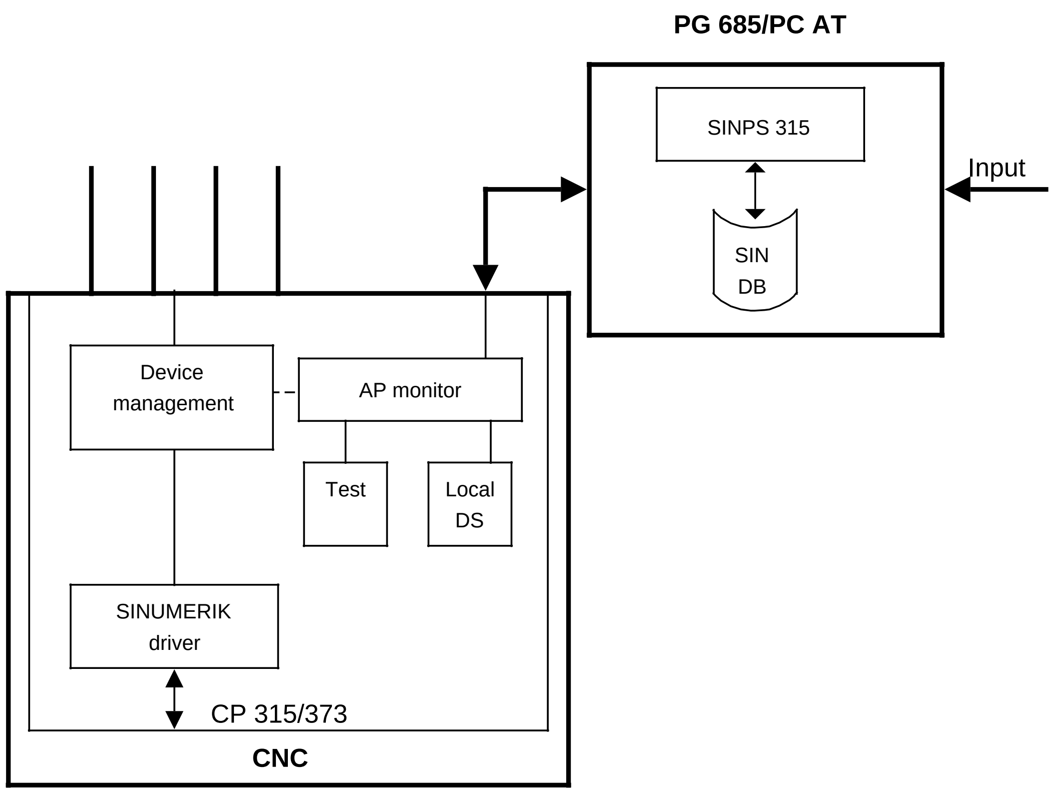
- PG 685 and PG 685T under PCP/M86
- PC AT under MS-DOS

With SINPS 315, the SINUMERIK-specific configuring data (input list 1, output list 1, format list) and connection-oriented configuring data (addressing list, list of interfaces) are generated on the PG or PC AT.

The configuring data is transferred from the PG or PC AT to the interface module via the programmer interface of the serial interface module CP315 (figure "Configuration of CP315").

The data of input list 1, output list 1, format list, addressing list and list of interfaces is valid for the CP 315 basic board and the CP 373 expansion board.

It is stored on the CP 315 basic board.



Configuration of CP 315

The information on the basis of which the SINUMERIK driver interprets and/or generates the SINUMERIK-specific part of the message frames is stored In input list 1, output list 1 and in the format list. The structure of these lists is identical with that of input list 1, output list 1 and the format list of the bus interface module (see figures "Structure of input list 1" and "Structure of output list 1"). They have been explained in Section 7.1.3 above.

The connection-related data which is required by the device management is stored in the addressing list and in the list of interfaces.

Addressing list

In this list the SINUMERIK-internal connection identifier, which is symbolized by logical nodes, is assigned to the corresponding interface and its defined AS 512 parameters, if the AS 512 protocol is implemented. The interface list specifies whether or not the serial interface is configured with the AS 512 protocol. The addressing parameters and the data for the AS 512 header, if any, are entered in the addressing list. The structure of the addressing list is represented in the diagram ”Structure of addressing list”.

If a serial interface is configured without the AS 512 protocol, only one logical partner may be assigned to this interface. Several logical partners can be assigned to an interface by means of the AS 512 parameters (co-ordination flag, data block, data word).

In input direction the interface number and the corresponding AS 512 parameters - if configured - assigned to a logical partner serve as search key.

Interface no. (AS 512 parameter)	Log. partner

In output direction the logical partner which is assigned to an interface with the corresponding AS 512 parameters - if configured - serves as search key.

Log. partner	Interface no. (AS 512 parameter)

Designation	Parameter
Log. partner destination / sender	e.g. production control computer
Interface no.	1......4
AS 512 parameters:	
Command	AD / AX
Destination	DB, DW
Coordination flag	byte, bit
[dotted box] configurable in the interface list: with/without AS 512 protocol	

Structure of the addressing list

List of interfaces

The procedure driver data and protocol parameters of the CP315 / CP373 serial interfaces are defined in the list of interfaces.

The list of interfaces covers 4 data blocks which corresponds to the maximum number of serial interfaces.

For each interface one block is configured by which the procedure and protocol parameters of the corresponding interface are determined, such as driver type, baud rate etc.

The figure "Structure of list of interfaces" shows the contents of the block for one interface.

Designation	Parameter
Interface status	inactive / active
Type of driver	3964 R / LSV 2
Character frame	Number of data bits
	Number of stop bits
	Parity
Baud rate	110.....19200 bauds
Procedure parameters	(see below)
Protocol type	AS 512
Block size AS 512	1 x 128 / 2 x 128 bytes
Response message frame	with / without
Monitoring time response message frame	x times seconds
Monitoring time subsequent message frame	x times seconds
Max. number of single message frames	0 / 1/ ....
Max. number of subsequent message frames	0 / 1/ ....

Structure of the list of interfaces

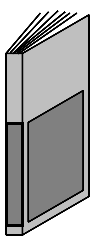
Procedure parameters in the interface list

Parameter	3964R	LSV2	XON/XOFF	RTS-LINE	TRANS-PARENT
Priority (Master/Slave)	x	x			
Acknowledgement delay (with/without)	x	x			
Repetitions (number)	x				
Repetitions in the enquiry phase (number)		x			
Repetitions in the text transfer phase (number)		x			
Acknowledgement monitoring time, sender (x times 10 ms)	x	x			
Acknowledgement time, receiver (x times 10 ms)		x			
Character monitoring time (x times 10 ms)	x	x	x	x	x
DSR evaluation (yes/no)			x	x	x
Termination identifiers (number)			x	x	x
Partner status equal (yes/no)			x	x	
Waiting for 1st. XON: (with/without)			x		
XON character (Hex value)			x		
XOFF character (Hex value)			x		
Hold-off interval (x times 10 ms)			x		
Device monitoring time (x times sec.)			x	x	

7.2 Test and simulation

For the installation, initial tests and fault detection, various test means are available which are used for simulation and monitoring (see figures "Test configuration for SINEC H1 bus" and "Test configuration for point-to-point connection"). The following table gives an overview of the test services offered.

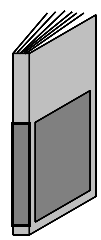
Test means	Test device	Application CP 231	CP 315	Task
SINT (SINUMERIK test software)	PG 685 / PG 685T PG 750 PC AT	X	X	Configuring, read-out and interpreting the trace buffers available on the interface modules.
SIM 850	PC 16-11 PC 16-20 WS 20-32	X	X	Simulation of the interface session between SINUMERIK 8X0 and production control computer for bus and point-to-point connection.
FOX PG-S	PG 675 / PG 685		X	Simulation of the interface session between SINUMERIK 8X0 and production control computer for point-to-point connection.
FOX-PGM	PG 675/ PG 695		X	Monitoring the message frame traffic on the serial transmission line for point-to-point connection.
SIPRA H1(SINEC Protocol Analyzer)	PG 685 / PG 695	X		Recording the activity on the SINEC H1 bus.

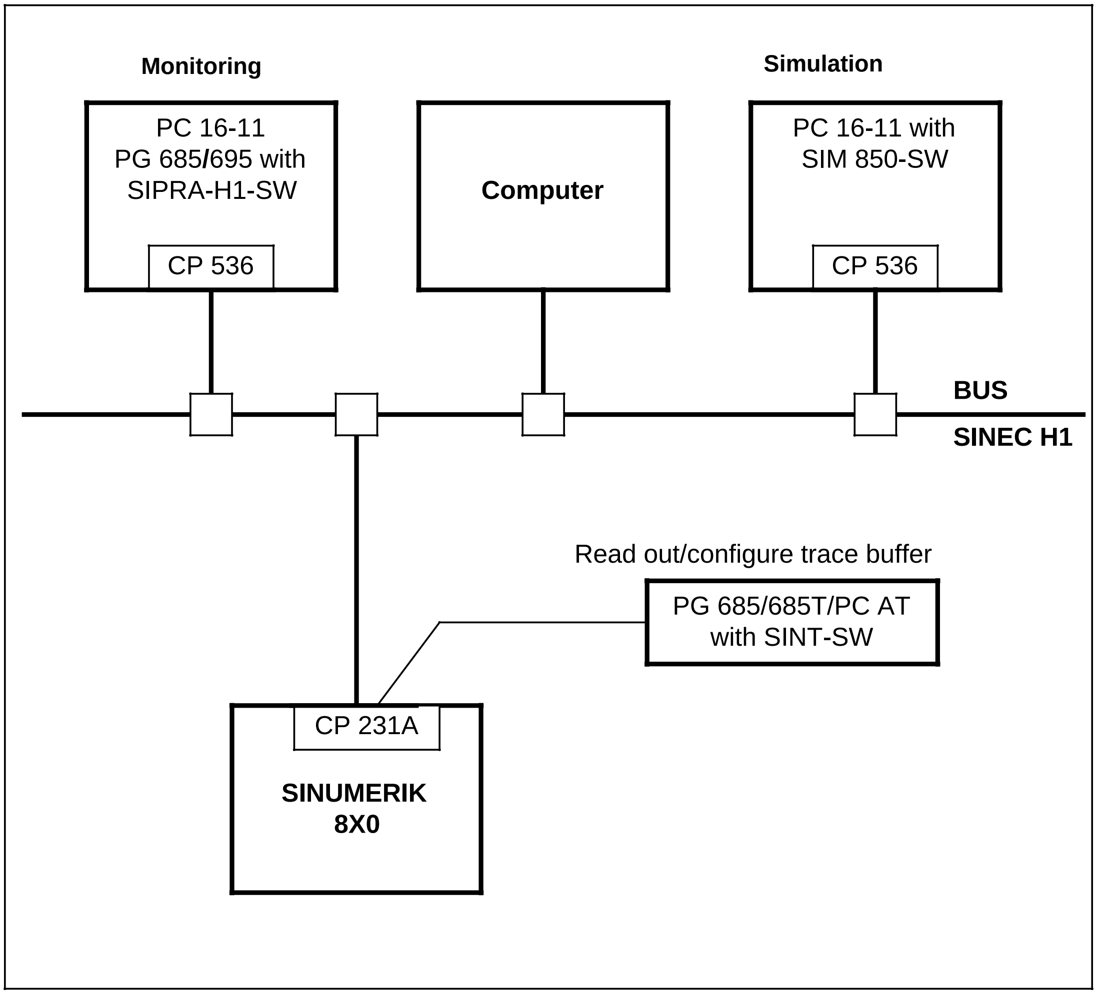


The configuring data can be stored with the SINT software on the EPROM system module of the CP 231A or CP 315.

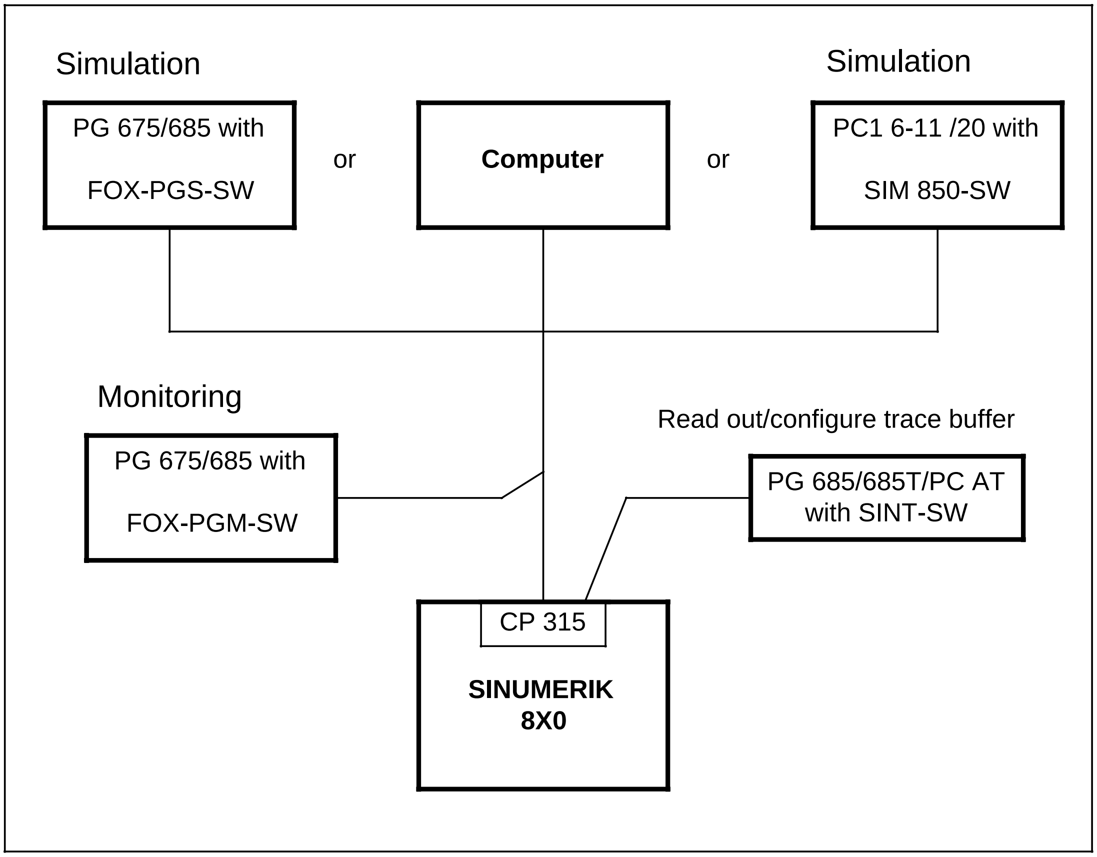
Further explanations are available in the following documentation:

SINUMERIK 840/880 Planning Guide
Computer Link
Test software SINT





Test configuration for SINEC H1 bus



Test configuration for point-to-point connection

7.2.1 SINT

With the aid of the SINUMERIK test software SINT the message frames can be traced on application plane in input and output direction and represented in readable form on the PG or PC AT.

For this purpose, the trace buffers on the interface modules are read out. It is possible to define message frame selection, connection selection and various trigger conditions so that the message frames and trace data can be indicated selectively. This is achieved via the menu-driven screen form software.

By assigning parameters the communication data to be entered into the trace buffers and displayed on the PG or PC AT screen is defined.

With the aid of the format lists the data is displayed in a comprehensible form.

The traced data can be printed out. Communication between the PG/PC AT and the interface modules is via the TTY interface of the interface modules.

Furthermore, the exception trace entries in the interface modules can be read out with SINT.

7.2.2 SIM 850

SIM 850 makes it possible to simulate the interface session between SINUMERIK 850 and production control computer. One node (NC or computer) is replaced by a personal computer with the program "SIM 850". Single message frames and message frame sequences can be tested.

The link between the communicating nodes can be a point-to-point connection or a bus connection.

The message frames are created by the user with a text editor (e.g. Wordstar, VEDIT) on the PC. The message frame traffic is recorded in a log file and is displayed on the screen.

The log file can be printed out.

7.2.3 FOX PG-S

The FOX PG-S simulator is used for the installation of control systems and computers with serial asynchronous links.

The simulator is connected to the object to be tested (control, computer) and simulates the peer node.

The following functions are supported:

- Receive data from connected controls,
- The simulator sends data to connected controls.

The necessary response messages and/or data message frames can be sent either automatically or by using function keys.

Initiation conflicts are resolved automatically. The simulation program executes necessary message frame repetitions automatically. Message frames are repeated max. twice.

Representation is either in ASCII or HEX code.

7.2.4 FOX PG-M

This program is used to monitor the message frame traffic which is executed via the point-to-point connection. For this purpose, an adapter looped into the transmission circuit and a FOX PG-M program which is executable on PG 675 or PG 685 are required. The adapter ensures opto-decoupled monitoring of the message frame traffic thus preventing interference of the data traffic. The FOX PG-M evaluation program serving the PLC and printer interface of the programmer, is called by the CP/M-86 operating system.

During recording of the data traffic, a ring buffer in the PG (reserved memory capacity in the RAM area) is written in parallel. It is only erased again when the FOX PG-M monitoring program is quitted.

The contents of the ring buffer can be stored on a formatted floppy disk. The floppy disk files can also be output on the screen within the FOX PG-M program using a function key or to a printer, e.g. PT 88.

7.2.5 SIPRA H1 (SINEC Protocol Analyzer)

The SIPRA H1 bus test software is used to monitor the message frame traffic on the SINEC H1 bus. For this purpose, a PG 685 with a CP 536 communication processor is required. The connection can be made on a transceiver which is already available in the system.

SIPRA H1 makes it possible to monitor and store the message frames sent over the SINEC H1 bus (layers 1 to 4 in plain text including useful data). Various conditions can be specified for this process, e.g. the entire bus traffic or only the traffic between 2 particular stations can be recorded.

Furthermore, trigger conditions can be specified.

For display on the screen the headers of layers 2 to 4 of the OSI seven layer model are represented already decoded. The headers of the other protocol layers and the useful data are displayed both in hexadecimal and in ASCII format.

8 Abbreviations

AP	Automation Protocol
AP PDU	Application Protocol Data Unit
AS512	Point-to-point connection for SIMATIC
ASCII	American Standard Code of Information Interchange
CL	Computer Link
CNC	Computerized Numerical Control
COM	Communication CPU
COMCLS	Command Class
COMCOD	Command Code
CP	Communication Processor
CP231	SINEC H1 interface module for SINUMERIK (Ethernet)
CP315	Basic board of the point-to-point connection for SINUMERIK, active serial interface module
CP373	Expansion board of the point-to-point connection for SINUMERIK,
	3 additional interfaces
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access Collision Detect, Ethernet
DB	Data block
DLE	Data Link Escape
DNC	Direct Numerical Control
DPR	Dual port RAM
DS	Directory services
DX	Data block (extended area with PLC 135W)
ECMA	European Computer Manufacturers Association
	Electronic Industries Association
ENQ	Enquiry
EOT	End of transmission
ERRCLS	Error Class
ERRCOD	Error Code
ETX	End of Text
EU interface	
module	Expansion unit interface module
File server	Central data base

FMS	Flexible manufacturing system
GM	General Motors
HW	Hardware
ID	Identification
ISO	International Standardisation Organization
K	Kernel sequence
LAN	Local Area Network
LSV2	Low Speed Version 2, procedure
MAP	Manufacturing Automation Protocol
MMFS	Manufacturing Message Format Standard
MPF	Main Program File
MPR	Multi Port RAM
MPXADR	Multiplex Address
NAK	Negative Acknowledgement
NC	Numerial Control
NC Archiv	Archiving and distribution of NC programs
OSI model	Open Systems Interconnection communication of open systems to the ISO 7-layer model, ISO reference model
P	Configurable message frame
PDU	Protocol Data Unit
PDUID	Protocol Data Unit Identifier
PDUREF	Protocol Data Unit Reference
PG	Programmer
PG 685	Programmer type 685
PG 685 T	Programmer type 685 turbo
PLC	Programmable Logic Control
PP	Part Program
PPS	Product Planning and Control
PROTID	Protocol Identifier
PS	Configuration service
RC	Robot Control
ROSCTR	Remote Operating Service Control
S	Standard message frame
S5	SIMATIC control with STEP 5 language
SCP	SINEC Communication Processor

SGSQNR	Segment Sequence Number
SINEC	Siemens Network Architecture
SINEC AP1	Siemens Network Architecture Automation Protocol Version 1
SINEC H1	Communication network of the Siemens programmable controllers
SINEC NM	SINEC Network Management
SINEC NML	SINEC Network Management Local
SINPS231	SINUMERIK Configuration Service for CP 231
SINPS315	SINUMERIK Configuration Service for CP 315
SINT	SINUMERIK test software
STR	Standard Routine
SW	Software
TIME - INST	Time Instruction
TIME - RESP	Time Response
TOA	Tool Offset, tool offset in main memory
TPS	Transport Control
TPV	Transport connection, logical channel
TSAP	Transport Service Access Point
ZOA	Zero Offset, in main memory
ZOF	Zero Offset
ZVZ	Character delay period

9 Appendix

9.1 Reception using the 3964R procedure

(CP315 as receiver = passive CP315)

The receiver waits in idle state for the connection being established by the peripheral unit. When the driver is ready to receive and the STX character is received, a DLE response is given and the driver is switched to receive mode.

Then following receiving characters are stored in the input buffer. If two consecutive DLE characters are received, these are loaded into the input buffer as one DLE character only.

Reception of the incoming characters is monitored using the character delay period. If the character delay period elapses without a character being received, the NAK character is sent to the peripheral unit.

When recognizing the character sequence DLE, ETX and BCC, the driver terminates reception, loads ETX as end character into the buffer (DLE is suppressed) and sends DLE to the peripheral unit to indicate an error-free (or NAK for a block containing errors) received block. Data exchange is thus ended. The driver returns into the passive state.

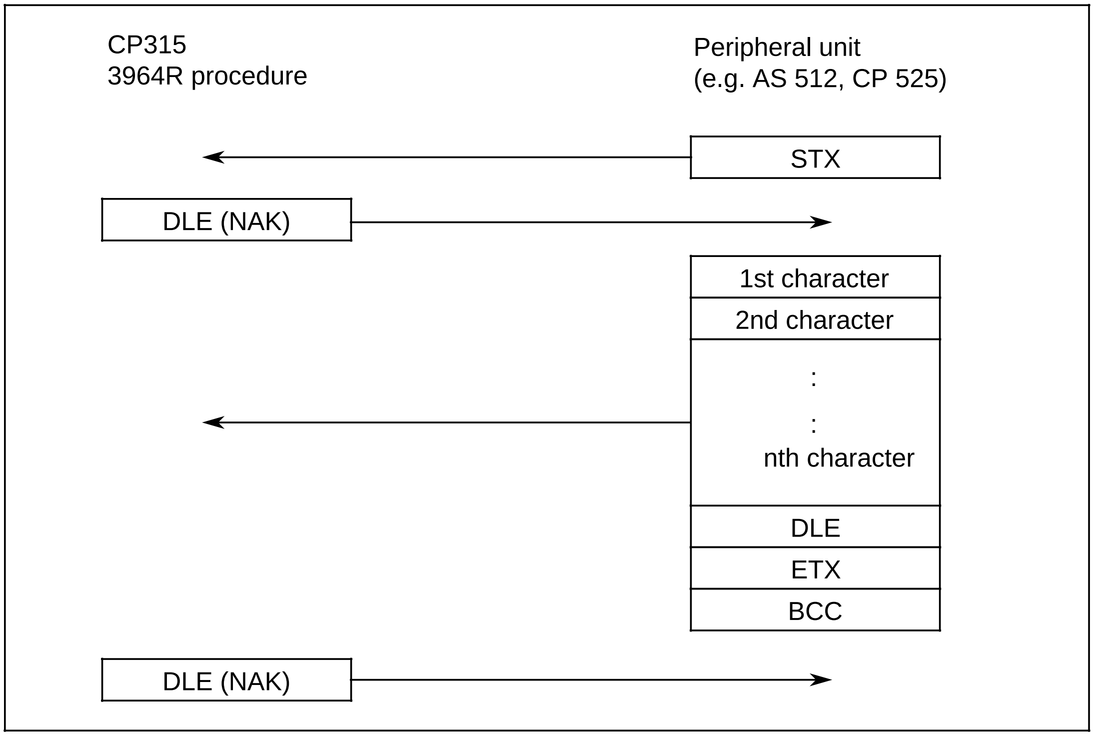
If the driver status is unclear upon receipt of STX (e.g. contents of input buffer not yet retrieved), the driver sends NAK to the peripheral unit.

If during reception a break condition on the input line occurs, or the character delay period is exceeded or a buffer overflow occurs (>128 characters), the receiving driver aborts reception, sends NAK and returns to the passive state.

If transmission errors occur during reception (lost character, frame error, parity error) reception is continued, and after arrival of the DLE, ETX, BCC characters the received block is acknowledged negatively by transmitting NAK to the peripheral unit. A repetition of the block is then expected.

If the block cannot be received even after several attempts (value is configurable) without error, the 3964R procedure aborts the reception and reports an error.

Example : Error-free data communication with the 3964R procedure:

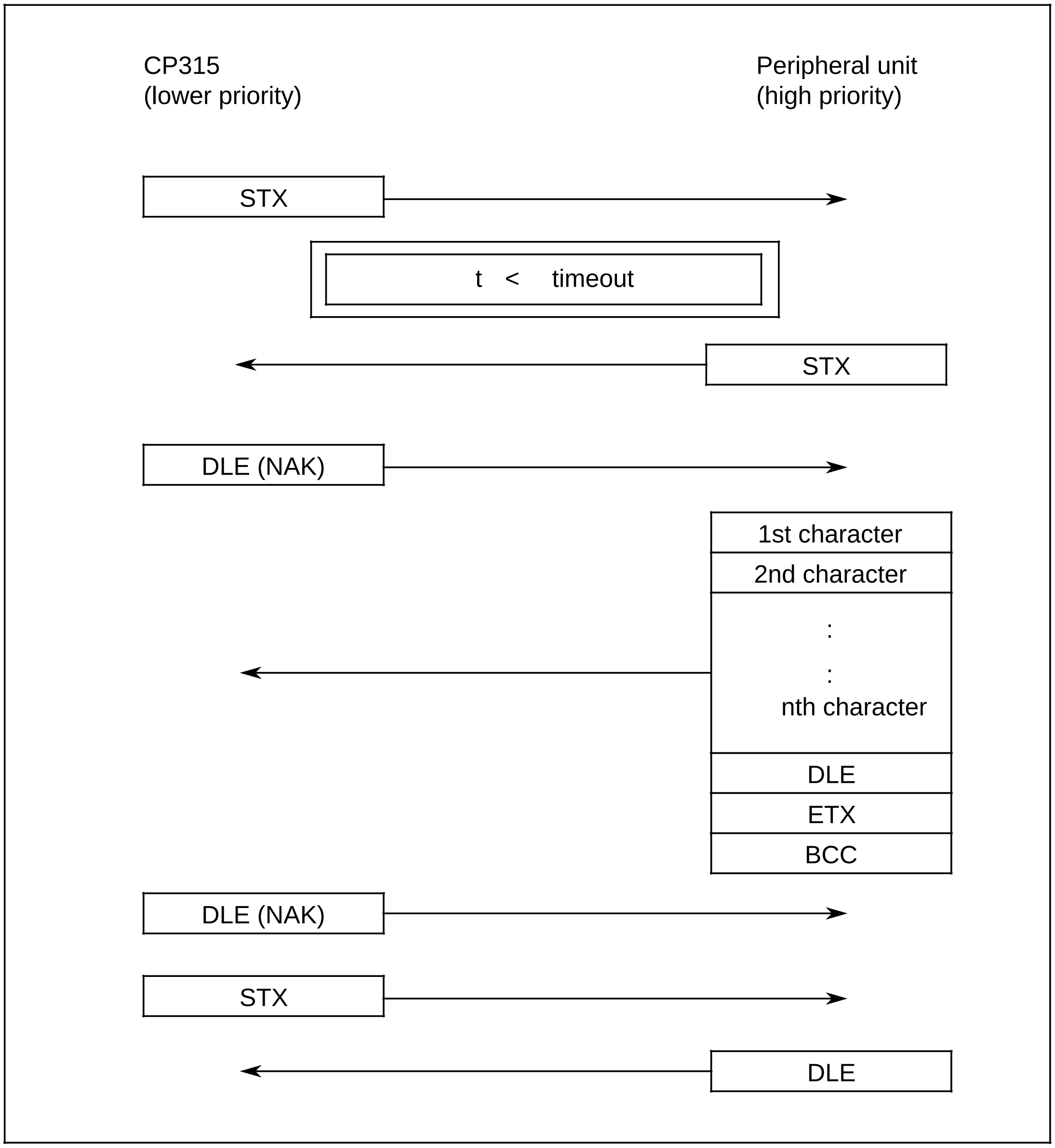


Initialization conflict

If a device does not respond to a request to send (STX character) by its peripheral unit within the timeout period with the acknowledgement DLE or NAK but also sends the STX character, this means that an initialization conflict exists. The two devices wish to execute a transmission task at the same time.

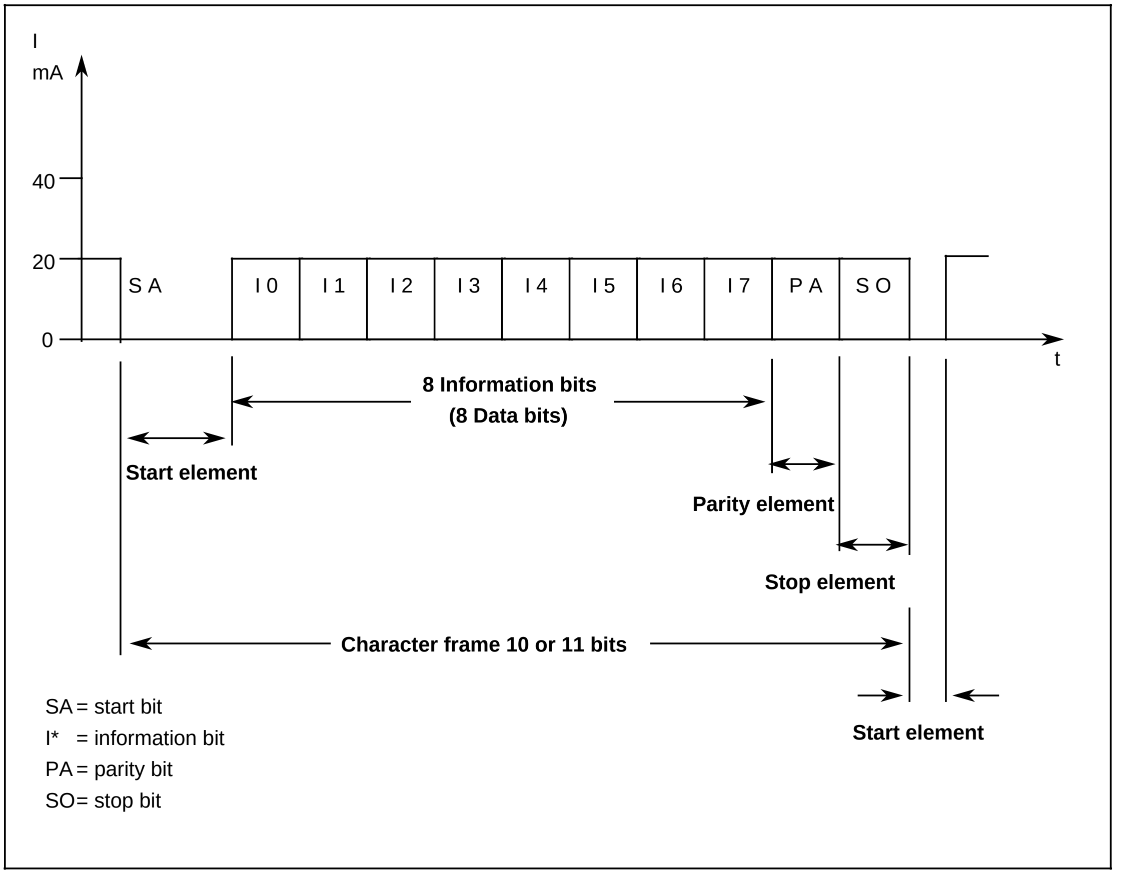
The device with the lower priority cancels its request and responds with the DLE character. Then the device with the higher priority sends its data as described above. After release of the connection, the device with the lower priority can execute its transmission task.

Example: Procedure in the case of an initialization conflict:



Character frame

In order to recognize each character again at the receiving end and to check error-free transmission, further bits are added to the characters sent, either as prefix or suffix. Each character to be transmitted begins with a start element and ends with at least one stop element. The 3964R procedure uses 8 data bits. The parity bit is used to protect the data against transmission errors.



The control characters for the 3964R procedure are in accordance with the DIN 66003 standard for the 7-bit code. At the end of each data block a block check character (BCC) is sent to secure the data.

For the information characters a code is not specified (code transparency).

Data protection

A parity bit is appended to the character to be transmitted for protecting the data against transmission errors. Additionally, with the 3964R procedure, the sum of bits of equal value of all characters in a block is complemented by a further bit. The resulting block check character BCC itself is secured using the character parity method and transmitted at the end of the block. All characters of the block except the STX start control character are covered by that procedure.

Transmission rate

The data transmission rate is specified in bits/sec. (bauds). At the configuration stage, the following values can be selected from the table below.

19200	bauds
9600	"
4800	"
2400	"
1200	"
600	"
300	"
150	"
110	"

Priority

The priority of the 3964R procedure in case of an initialization conflict is configured. For the other node, the opposite priority must be set accordingly.

The following can be selected

- high priority
- low priority

Monitoring times

The following monitoring times are used for monitoring the data transfer:

- Acknowledgement monitoring time (= timeout)

  The acknowledgement monitoring time is started after transmission of the STX or DLE ETX BCC control characters.

  If no valid positive acknowledgement arrives within the timeout period, the driver is requested to repeat the same job. If even after several repeated attempts (configurable) a correct transmission was not possible, an error is entered in the exception trace.

  The timeout period is programmable to integrate several components. The acknowledgement monitoring period can be specified in intervals of 10 ms (1 word, max. FF FF).

- Character monitoring time (character delay time)

  The receiving CP 315 monitors the arrival of the individual characters within the character delay time.

  If a character does not arrive within the character delay time, the receiving driver aborts reception, sends NAK and returns to the passive state.

  Since, in general, different types of units will communicate with each other, the character delay period can be parameterized. It can be specified in intervals of 10 ms (1 word, max. FF FF).

9.2 LSV2 procedure

In the following, the individual phases of the LSV2 procedure are described in greater detail.

Procedure representation

The control character sequence is represented in three-column diagrams.

In the left column the station requesting to send or the text sending station is entered. The centre column contains the response of the station on the other end. In the right column, references to other responses of the station requesting to send or the text transmitting station are entered.

The control characters entered into the individual columns are sent by the station which is in the heading of the column.

If this character is received mutilated by the other station, the control character is represented with a line through it.

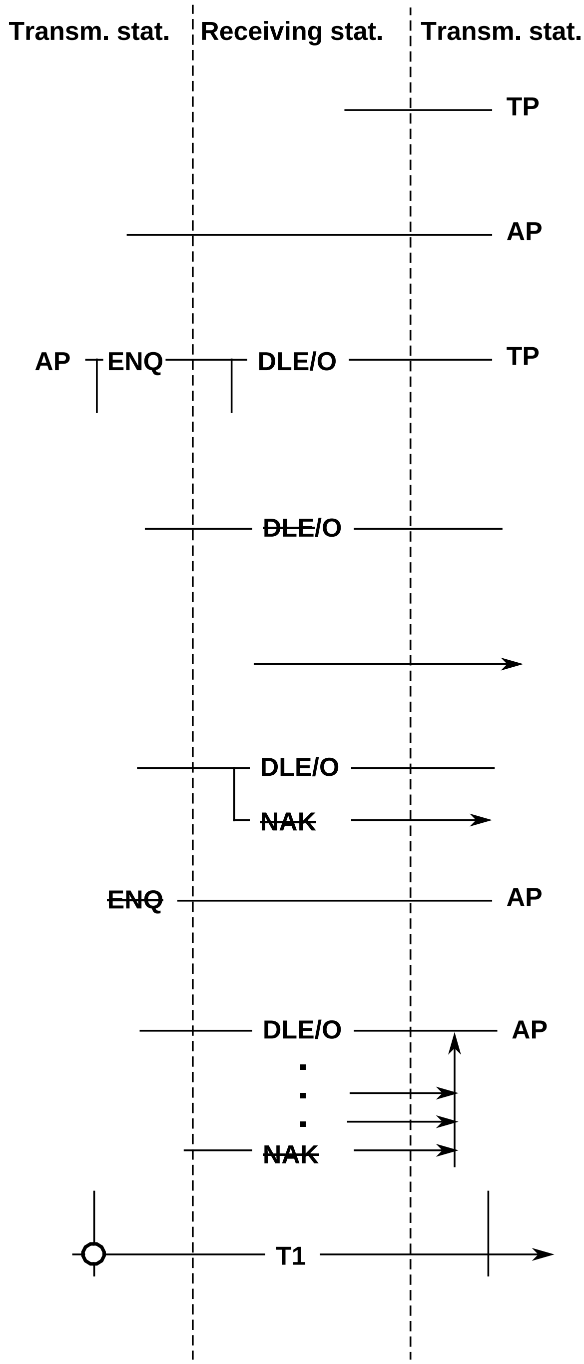
The course of the procedure is represented by a line.

The line branches in the case of possible errors.

If a station does not respond, the line is drawn without a character being entered.

If a station responds to different error possibilities in the same way, the corresponding lines are drawn together.

The diagrams also show monitoring times. The setting of a monitoring period is marked by a small circle. The arrowhead marks the end of this period and points to the response of the station which has set the period.



TP =Text transfer phase (data communication phase)
AP =Request phase

Data exchange process

The procedure can be divided into three phases:

- Enquiry phase,
- Text transfer phase and
- Completion phase.

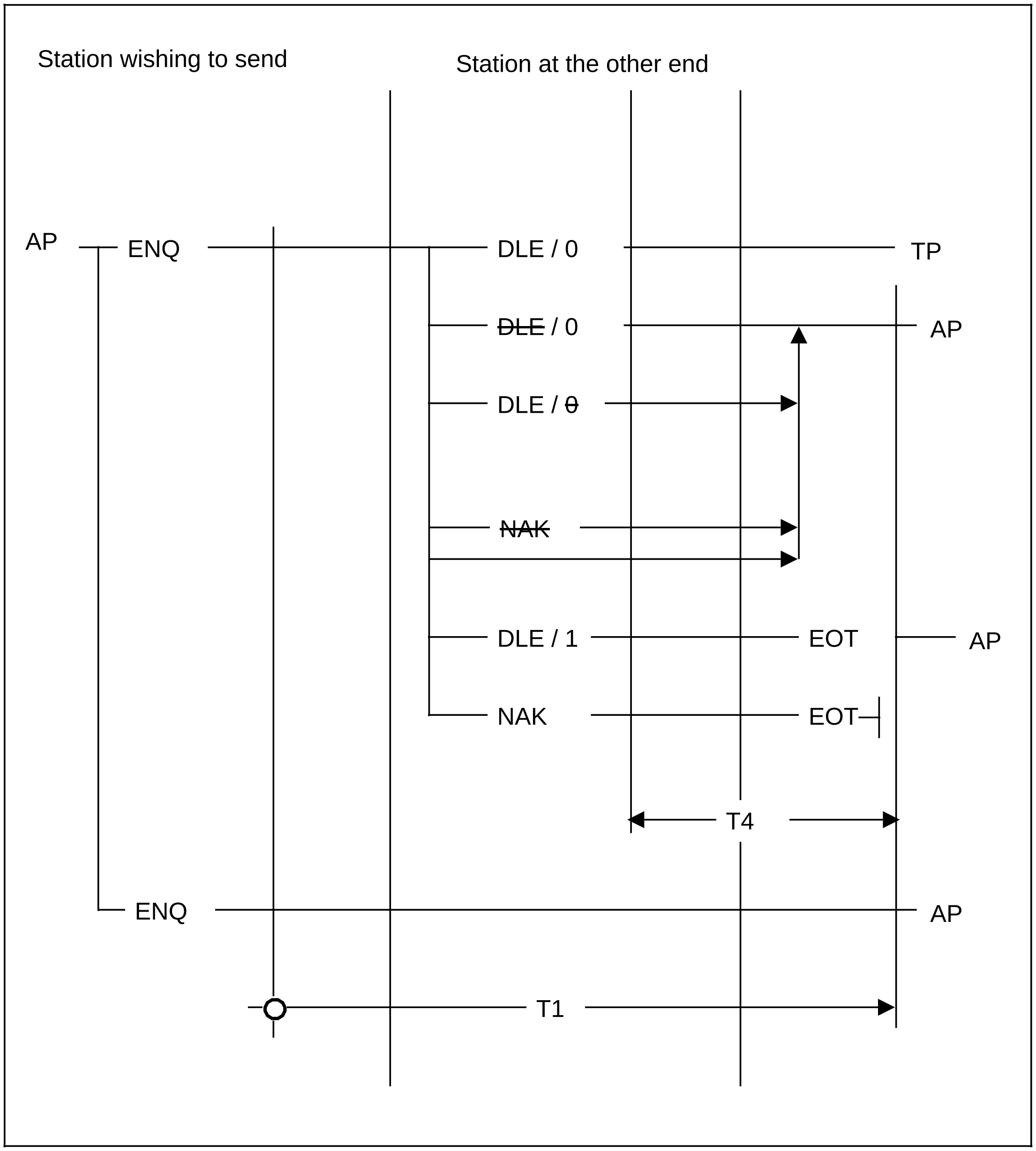
Enquiry phase		Text transfer phase		Com.ph
MA	SL	MA	SL	MA
ENQ	DLE0	DLE STX Text DLE ETX BCC	DLE1	EOT

MA : Master station = Text transmitting station
SL : Slave station = Text receiving station

Enquiry phase

At the beginning of the first phase, the enquiry phase, both stations are in a neutral state. If a station wishes to send, it requests the station on the other end to receive text by transmitting the ENQ control character and waits for the DLE 0 control character sequence. If the other station sends this character sequence and it is received by the station wishing to send without error, the station wishing to send moves into the text transfer phase. The figure on the next page shows the enquiry phase and also covers all possible error conditions. If the station wishing to send receives a character sequence containing errors, it ignores these characters. Only after monitoring timeout, the station recognizes the error condition and repeats the ENQ enquiry.

If the station wishing to send receives a negative acknowledgement NAK, the station on the other end is not ready to receive text, and the station wishing to send aborts the procedure with EOT. Upon receipt of the DLE 1 character sequence, it is assumed that the other station end is still in the completion phase. Then the station wishing to send sends EOT (thus causing the station on the other end to return to the initial state) and, with ENQ, requests it again to receive text. By sending STX, the station wishing to send moves into the text transfer phase and becomes a text transmitting station. By receiving STX, the other station also moves into the text transfer phase and becomes a receiving station.



Enquiry phase

Text transfer phase

The transfer is executed in the transparent mode. In this mode, all text characters can be coded as desired within a fixed character frame.

The DLE STX character sequence specifies the switchover to the transparent mode. Switchback to the standard mode is by means of the DLE ETX character sequence. After an error-free reception the receiving station acknowledges with DLE 1.

When the transmitting station receives this acknowledgement, it moves into the completion phase and ends the transmission with EOT. On reception of EOT, the receiving station also moves into the completion phase.

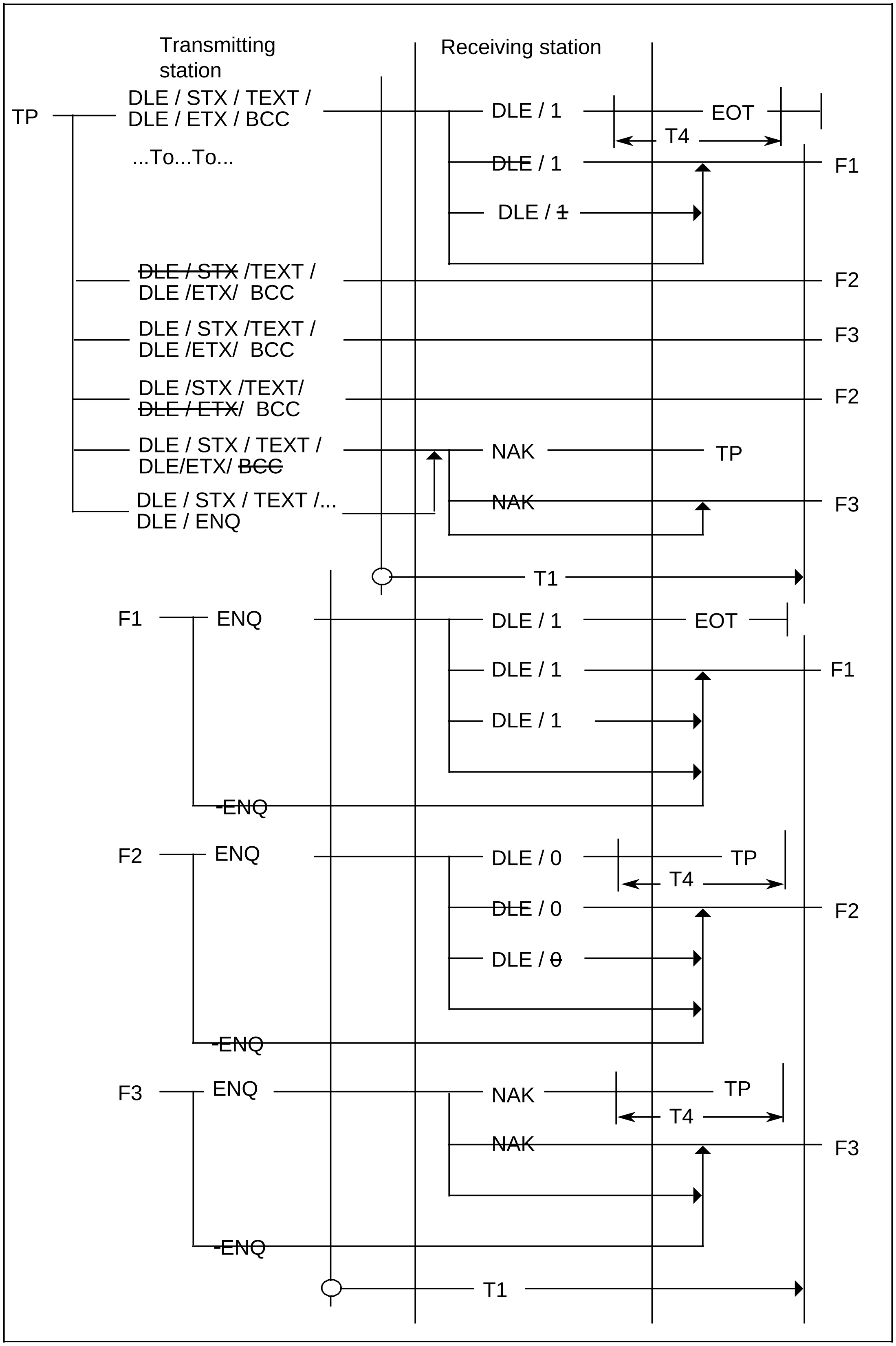
If the text is mutilated during transmission, the receiving station acknowledges with NAK and waits for a new text transfer.

If the transmitting station receives a mutilated acknowledgement, this character is ignored and the transmitting station sends a further acknowledgement request after the T1 monitoring time has elapsed.

If on the other hand, the receiving station receives a mutilated control character, it does not respond, and this error is only noticed after the monitoring time in the transmitting station has expired.

Completion phase

In the completion phase, the transmitting station sends EOT thus moving into the initial state (neutral state). If the receiving station receives the EOT undistorted, it also returns into the initial state. If EOT is received by the receiving station with an error, either the T4 monitoring time responds at the receiving station and the receiving station moves into the initial phase, or the transmitting station moves again into an enquiry phase with ENQ and the receiving station responds with DLE 1.



Text transfer phase

Monitoring times

To avoid undefined conditions which can occur due to errors (mutilation of control characters or control character sequences), monitoring periods are used. These periods are used to initiate rerun routines if errors have occurred or to disable the connection.

TO: e.g. 2s — On reception of DLE/STX the receiving station starts the TO monitoring time (value configurable).

In order to permit the transmission of transmission character sequences of different lengths, the T0 period is reset with every character received. The T0 period is reset and stopped on reception of the DLE/ETX transmission control character. If no character arrives within the T0 monitoring time, the receiving station moves into the basic phase.

T1: e.g. 3s — The transmitting station or the station wishing to send starts the T1 monitoring time by transmitting ENQ, DLE/ENQ and BCC. If within this period no valid acknowledgement comes from the station on the other end, the latter is requested with ENQ to send an acknowledgement (value configurable). The periods T0 and T1 must be configured so that T0 expires before T1.

T4: e.g. 24s — The receiving station sets the T4 monitoring time with transmission of acknowledgement DLE0 DLE1 or NAK. If within this period a valid response of the transmitting station does not come, the receiving station moves into the basic phase (value configurable)

Repetitions (values configurable)

Repetitions are possible in the enquiry phase and in the text transfer phase. The number of repetitions is configurable.

The station wishing to send transmits ENQ in the enquiry phase. If the transmitting station receives no or a mutilated acknowledgement, the request is repeated with ENQ after expiration of the T1 monitoring time.

If no DLE0 is received after the configured number of repetitions (e.g 4 times), the station sends EOT and aborts the procedure.

Repetitions also are possible in the text transfer phase (value is configurable). For this purpose, text repetitions and ENQ acknowledgement requests are added up (e.g.: one text repetition after NAK, two acknowledgement requests ENQ after mutilated acknowledgement). If a DLE1 is not received after the configured number of repetitions (e.g. 3 times), the transmitting station aborts the procedure with EOT.

Data protection

In order to guarantee reliable data transmission, a character and horizontal parity protection is used. In this way, multiple errors can be recognized. For every transmitted character a parity bit can be included in the transmission.

In addition, the transmitted text is saved on a block basis. For this purpose, a logical "XOR" is generated covering the message frame from ID to ETX inclusively. The character obtained in this way is the block check character (BCC). The block check function is set to zero for STX.

In the receiving station, BCC is also computed and compared with the transmitted BCC.

Example:

	STX	S	I	N	U	M	E	R	I	K	ETX		BCC
Bit position b1	0	1	1	0	1	1	1	0	1	1	1	→	0
b2	1	1	0	1	0	0	0	1	0	1	1	→	1
b3	0	0	0	1	1	1	1	0	0	0	0	→	0
b4	0	0	1	1	0	1	0	0	1	1	0	→	1
b5	0	1	0	0	1	0	0	1	0	0	0	→	1
b6	0	0	0	0	0	0	0	0	0	0	0	→	0
b7	0	1	1	1	1	1	1	1	1	1	0	→	1
b8	0	0	0	0	0	0	0	0	0	0	0	→	0

Parity generation

Character parity bit	1	0	1	0	0	0	1	1	1	0	0

DLE is not included into block checking, if it is transmitted with another data transmission control character such as DLE STX, DLE ETX.

In the case of the DLE DLE character sequence, the first DLE is not included in the block check.

If an irregularity is recognized, the receiver acknowledges with NAK. The text can be repeated after a negative acknowledgement.

Collision

A collision occurs if an ENQ comes after transmission of an ENQ instead of the expected DLE0.

In this case the station with the higher priority is allowed to transmit.

The station with the higher priority ignores the ENQ received and the other station cancels its request to send and transmits DLE0.

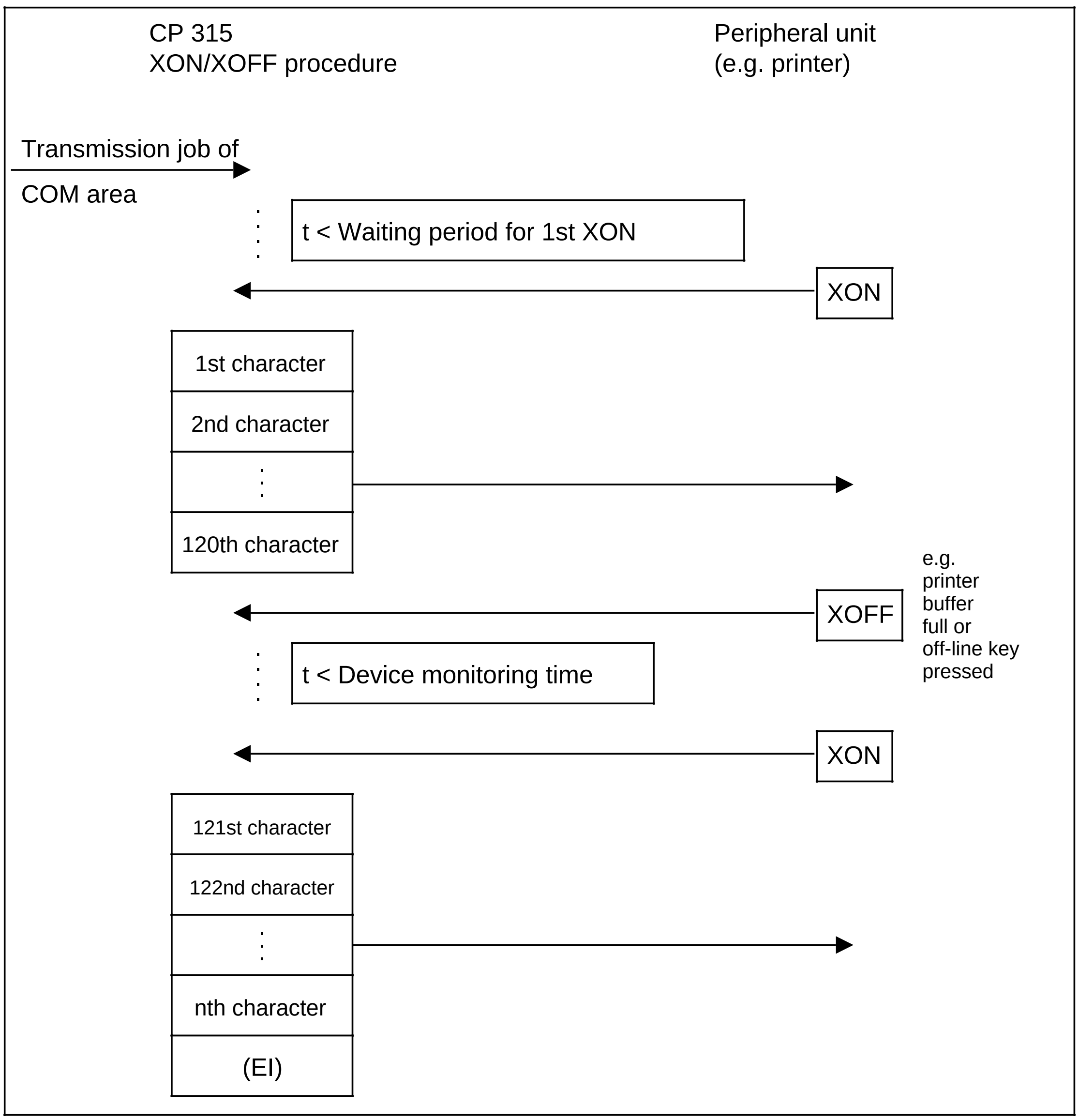
9.3 XON/XOFF procedure

If a monitoring time (device, character monitoring time or line reservation time) has expired, the driver sends the XOFF character and goes into the initial state.

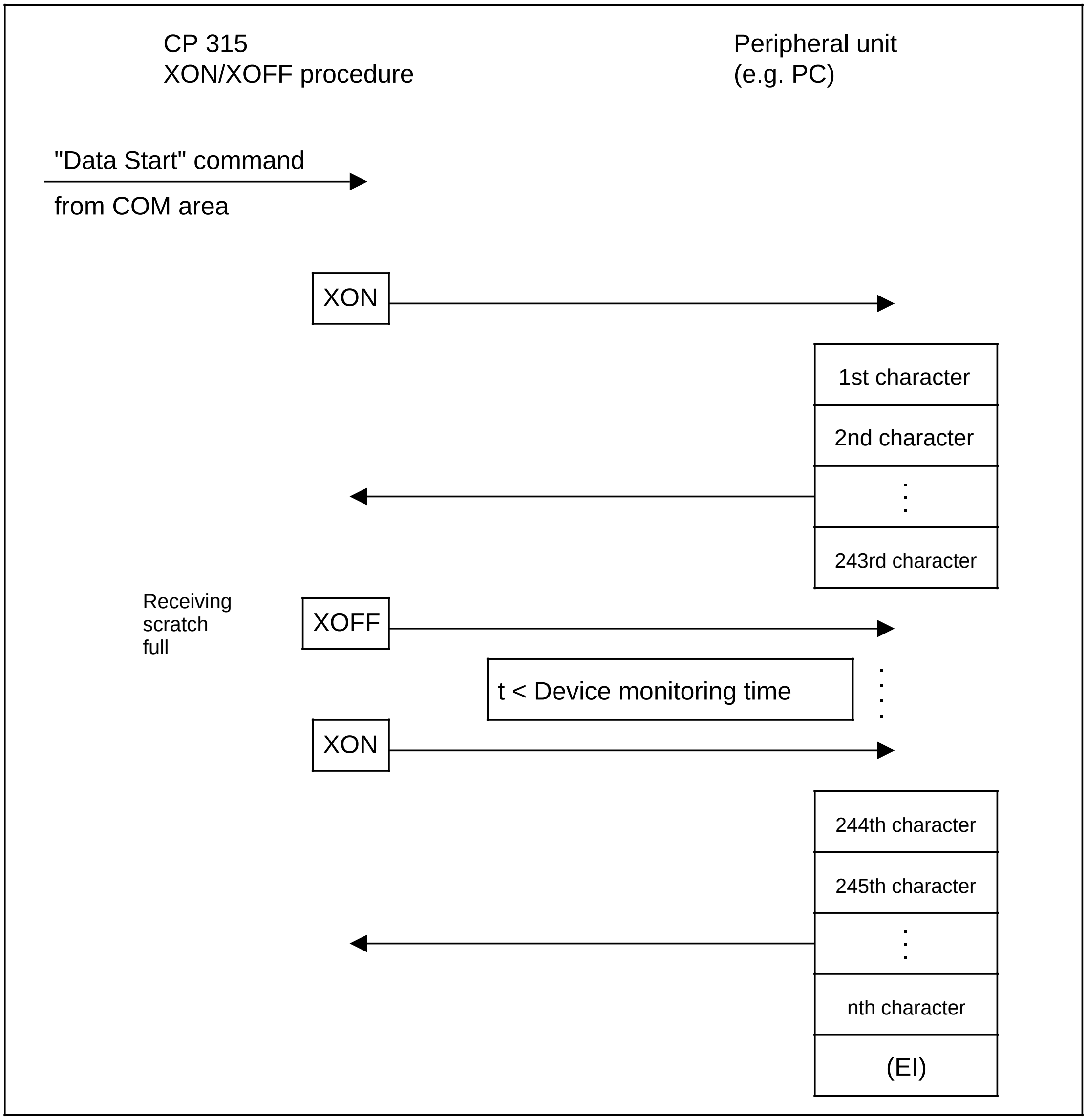
The configuring parameter "Node states equal" is insignificant unless both stations want to transmit at the same time. The node configured with "Node states equal=Yes" is the master and is entitled to interrupt the transmission initiated by the other node to transmit its data. Optionally, both nodes can be configured as master. In this case, they have the same priority and neither of them can interrupt or be interrupted during transmission.

a) Configuration: Node states equal: No
Waiting for 1st XON: With (EI)=End identifiers, if configured

Data output

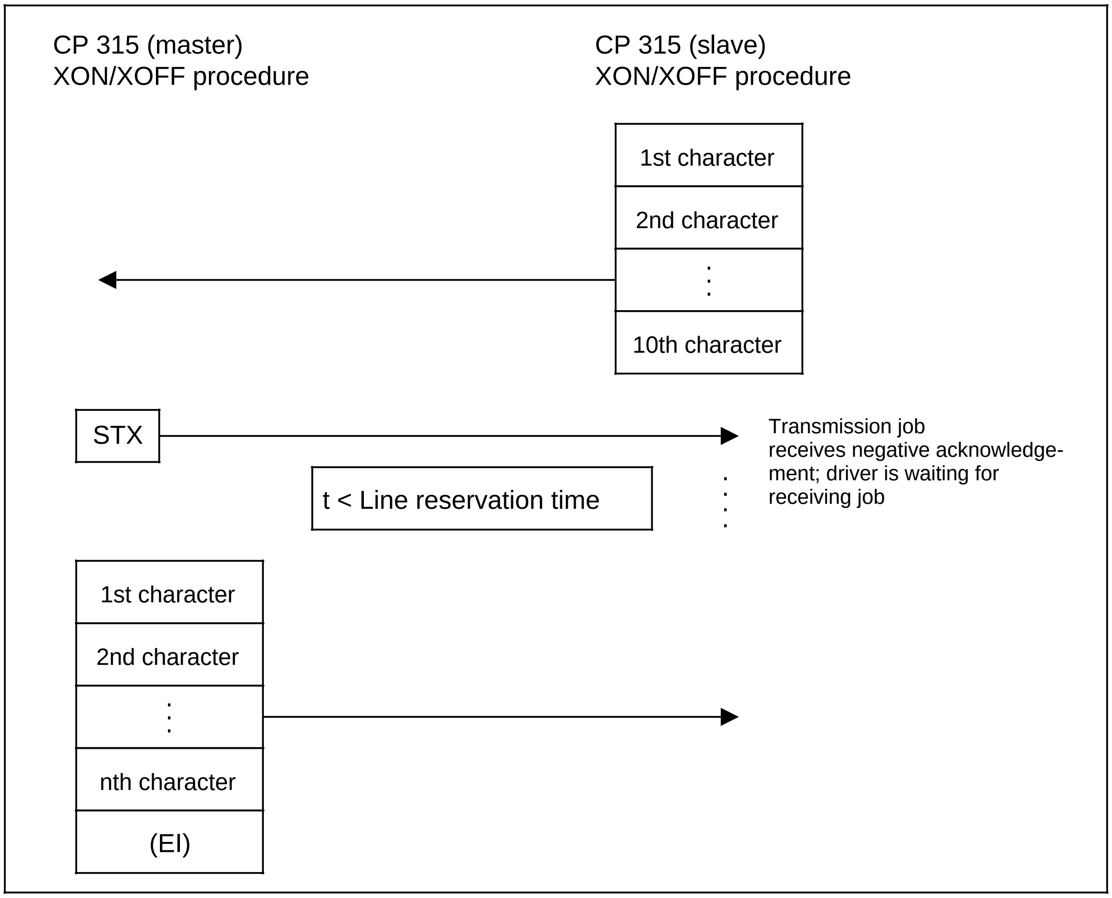


Data input


CP 315
XON/XOFF procedure
Peripheral unit
(e.g. PC)
"Data Start" command
from COM area
XON
1st character
2nd character
243rd character
Receiving
scratch
full
XOFF
t < Device monitoring time
XON
244th character
245th character
nth character
(EI)

b) Configuration: Node states equal: YES (CP 315 on the left=Master)
NO (CP 315 on the right=Slave)
Waiting for 1st XON: Without

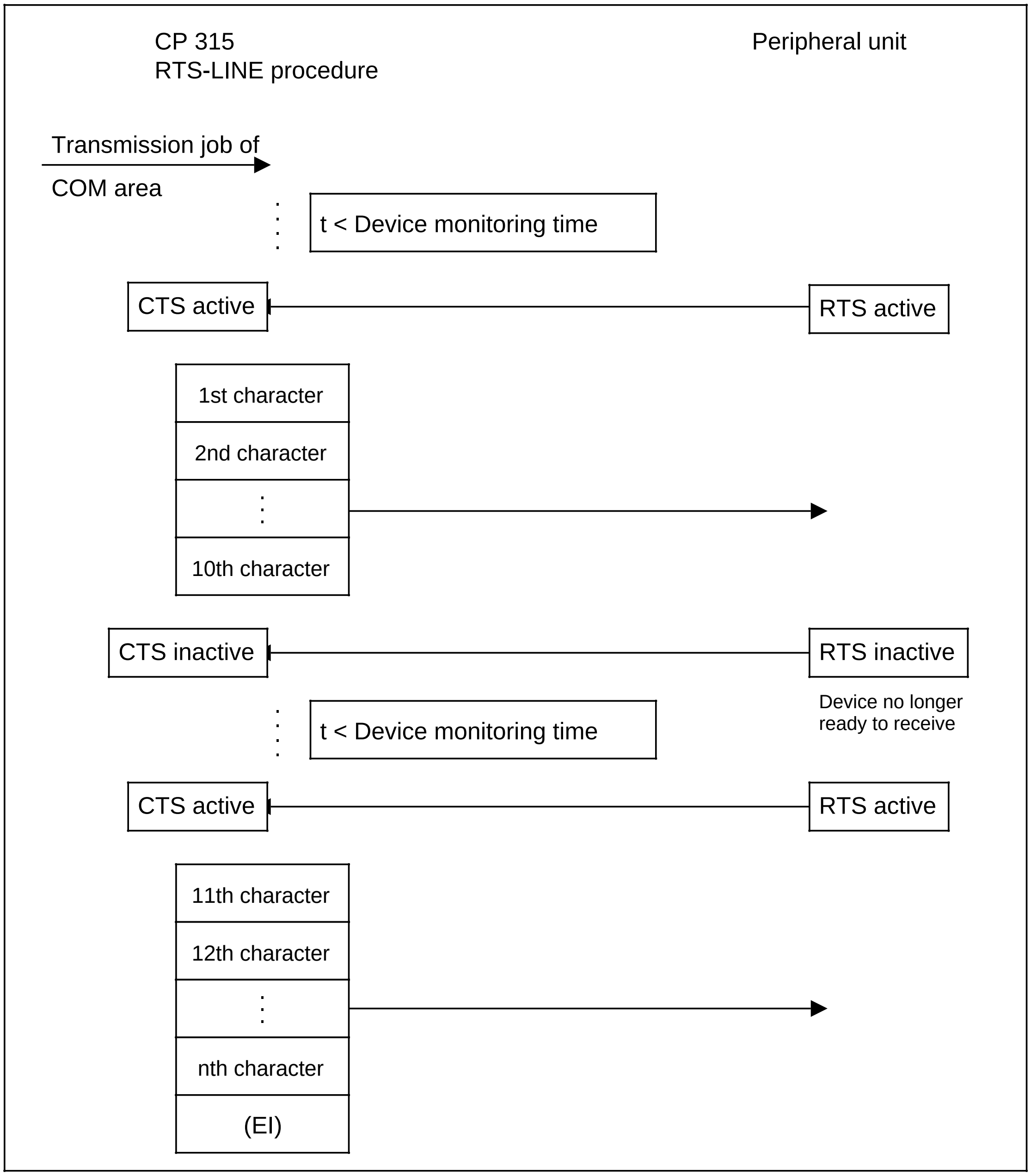
Initialization conflict



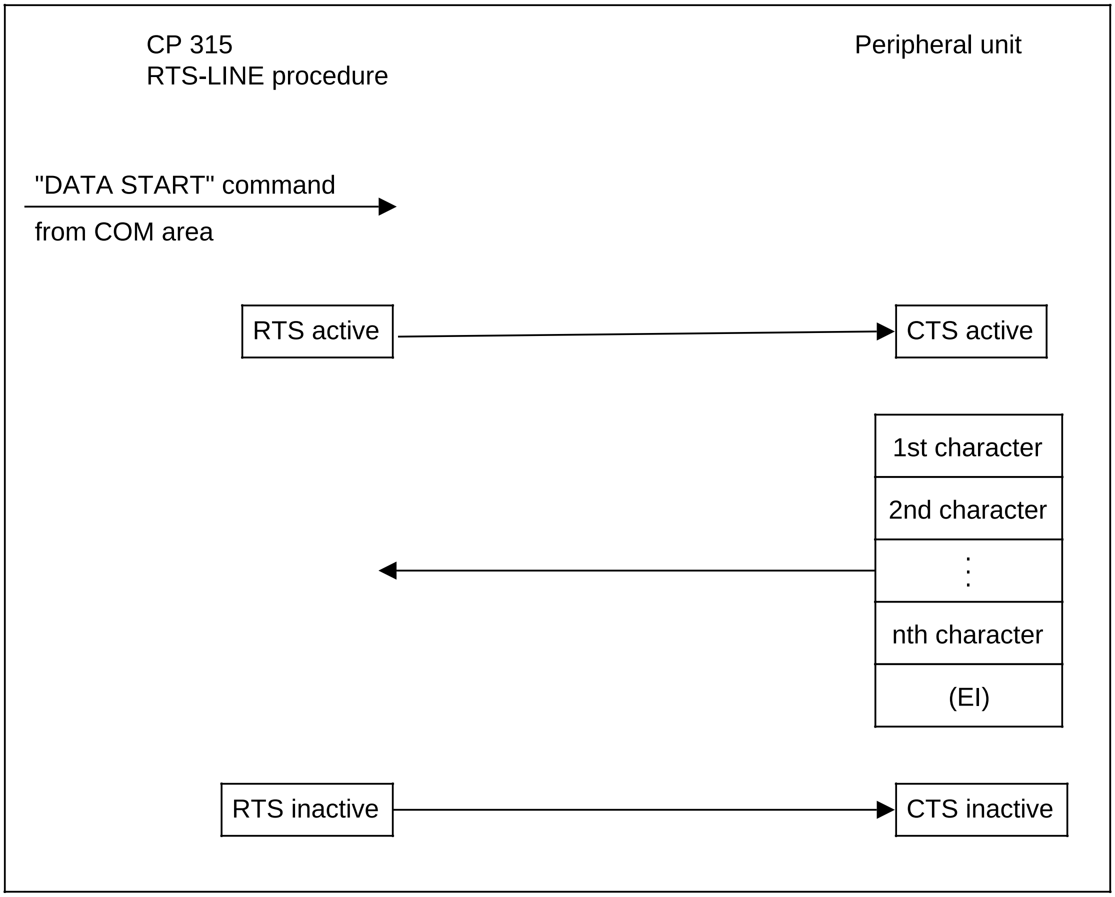
9.4 RTS-LINE procedure

(EI)=End identifiers - if configured

Data output



Data input


CP 315
RTS-LINE procedure
Peripheral unit
"DATA START" command
from COM area
RTS active
CTS active
1st character
2nd character
nth character
(EI)
RTS inactive
CTS inactive

9.5 Error coding of remote AP protocol processing systems and AP users

ERRCLS, ERRCOD	Error description
09 H C1 H	Server not configured
09 H C2 H	No server defined for this function
09 H C3 H	No resources for activation of the AP peer object available
09 H C4 H	AP PDU cannot be transmitted due to max. lack of resources period with (remote) node being exceeded.
69 H C1 H	AP peer object (function server) not implemented
69 H C2 H	Specified PROTID not supported by remote node
69 H C3 H	Specified COMCLS value not implemented for this MPXADR
69 H C4 H	Specified COMCOD value not implemented for this COMCLS and MPXADR
69 H C5 H	TACTID does not have implemented value
69 H C6 H	TSQNR has permissible value
69 H C7 H	Function not completely/partially contained in distributing table
61 H C1 H	Value of ROSCTR not supported by remote node
61 H C2 H	Specified MODFR1 value not implemented for the assigned function
61 H C3 H	Specified MODFR2 value not implemented for the assigned function
91 H C1 H	SGSQNR has permissible value
99 H C1 H	Processing aborted due to resources being assigned too long
51 H C1 H	APM (remote) in permissible condition

ERRCLS , ERRCOD	Error description
29 H C1 H	AP PDU volume too large for local processing
29 H C2 H	PDULG not equal to PARLG+DATLG+22
29 H C3 H	AP PDU is smaller than expected by remote and cannot be processed
29 H C4 H	PARLG is not valid
29 H C5 H	AP acknowledgement to be sent too long
29 H C6 H	AP acknowledgement to be sent too small
29 H C7 H	PDULG in AP acknowledgement to be sent is <> PARLG+DATLG+22

Additional error codes function class "serial transfer"

ERRCLS , ERRCOD	Error description
2A H 01 H	Transferred or requested useful data volume too large
2A H 02 H	Transferred or requested useful data volume too small
9A H 02 H	Format of the requested data not in accordance with format agreed
9A H 01 H	Data not in accordance with agreement