SIEMENS

SINEC

CP 5431 FMS with COM 5431 FMS

Volume 2 of 2

6GK1970-5AB01-0AA1 C79000-G8976-C048 Release 02

SIEMENS

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in der Druckschrift werden jedoch regelmäßig überprüft. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

Technische Änderungen vorbehalten.
Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

We have checked the contents of this manual for agreement with the hardware described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcome.

Technical data subject to change.

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility or design, are reserved.

Nous avons vérifié la conformité du contenu du présent manuel avec le matériel et le logiciel qui y sont décrits. Or, des divergences n'étant pas exclues, nous ne pouvons pas nous porter garants pour la conformité intégrale. Si l'usage du manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition. Veuillez nous faire part de vos suggestions.
Nous nous réservons le droit de modifier les caractéristiques techniques.

Toute communication ou reproduction de ce support d'informations, toute exploitation ou communication de son contenu sont interdites, sauf autorisation expresse. Tout manquement à cette règle est illicite et expose son auteur au versement de dommages et intérêts. Tous nos droits sont réservés, notamment pour le cas de la délivrance d'un brevet ou celui de l'enregistrement d'un modèle d'utilité.

Siemens Aktiengesellschaft

Elektronikwerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SINEC

CP 5431 FMS with COM 5431 FMS

Description C79000-B8976-C062/02

Note

We would point out that the contents of this product documentation shall not become a part of or modify any prior or existing agreement, commitment or legal relationship. The Purchase Agreement contains the complete and exclusive obligations of Siemens. Any statements contained in this documentation do not create new warranties or restrict the existing warranty.We would further point out that, for reasons of clarity, these operating instructions cannot deal with every possible problem arising from the use of this device. Should you require further information or if any special problems arise which are not sufficiently dealt with in the operating instructions, please contact your local Siemens representative.

General

This device is electrically operated. In operation, certain parts of this device carry a dangerously high voltage.

WARNING !

!

Failure to heed warnings may result in serious physical injury and/or material damage.

Only appropriately qualified personnel may operate this equipment or work in its vicinity. Personnel must be thoroughly familiar with all warnings and maintenance measures in accordance with these operating instructions.

Correct and safe operation of this equipment requires proper transport, storage and assembly as well as careful operator control and maintenance.

Personnel qualification requirements

Qualified personnel as referred to in the operating instructions or in the warning notes are defined as persons who are familiar with the installation, assembly, startup and operation of this product and who posses the relevant qualifications for their work, e.g.:

– Training in or authorization for connecting up, grounding or labelling circuits and devices or systems in accordance with current standards in saftey technology;
– Training in or authorization for the maintenance and use of suitable saftey equipment in accordance with current standards in safety technology;
– First Aid qualification.

Contents

1 Introduction

This volume of the manual "CP 5431 FMS and COM 5431 FMS under SINEC NCM" describes the protocol and services for open, heterogeneous PROFIBUS (PROcess FIeld BUS) communication with the CP 5431 FMS communications processor.

PROFIBUS is a bus system for applications in automation engineering in areas closely associated with the process.

With the PROFIBUS, SIMATIC S5 programmable controllers, programming devices, AT compatible PCs and other control systems and, of course, PROFIBUS-compatible devices from various vendors can be networked.

The CP 5431 FMS communications processor is used to connect programmable controllers of the SIMATIC S5 range to the local area network SINEC L2/L2FO and complies with the PROFIBUS standard (DIN 19 245) in Parts 1, 2 and 3 as an active station on the bus (PROFIBUS multivendor network).

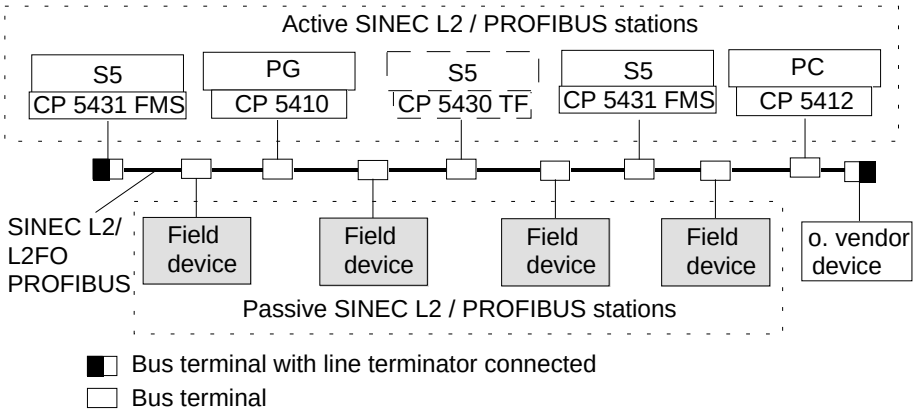


Fig. 1.1 Example of PROFIBUS L2 Configuration

Communications processors with TF protocol architecture (CP 5430 TF) and FMS protocol architecture (CP 5431 FMS) and the CP 5430 (A0) can co-exist on the same bus.

The manual describes the functions of PROFIBUS communication for SIMATIC S5 PLCs.

You configure the network using COM 5431 FMS under SINEC NCM (network and communication management). The configuration tool can be run on the PG 710, 730, 750 and 770 under the S5 DOS/ST operating system.

The PROFIBUS communications system provides the user system with a wide variety of services for using open communication in many applications. This variety of services is, of course, not required in all units and areas. For this reason, the standard remains flexible allowing various application areas and various system configurations and functional structures to be implemented. Profiles are therefore stipulated for different application areas (e.g. building automation, manufacturing, automation engineering) which contain part of the functions within the standard. The range of services is therefore adapted to the area of application of the field bus.

The information in this manual is intended for the following users:

- the planner and designer of a communications network,
- programmers of application processes that need to communicate with each other,
- customers wishing to use SINEC L2/L2FO in the SIMATIC S5 system.

General symbols


Active star coupler

Twisted pair


Bus terminal (terminating resistor connected)


Bus terminal (terminating resistor disconnected)

Data Terminal Equipment


Fiber optic cable


Optical bus terminal


SF repeater adapter

RS 485 repeater RS 485

✔ This character indicates an activity or operation for you to perform.

☞ **This symbol highlights special features and dangers.**

Requirements of the user:

To understand the examples, you should have the following:

- ➢ Knowledge of programming with STEP 5
- ➢ Basic knowledge of the use of handling blocks (HDBs). The description of the HDBs can be found in the manual for your programmable controller or in separate descriptions of the programmable controllers.

Training offer
Siemens provides SINEC users with a comprehensive range of training opportunities.

For more detailed information contact

Informations- und Trainings-Center
für Automatisierungstechnik

AUT 95 Kursbüro
Postfach 21 12 62
76181 Karlsruhe

or your local Siemens office.

Order numbers for the products mentioned in this manual can be found in the current catalogs.

To help you find your way through this manual the remainder of this section outlines the chapters briefly.

Chapter 2
Basics of the PROFIBUS
This chapter provides an introduction to the communications model by explaining terminology and inter-relationships and illustrates the interface to the SIMATIC S5 user.

Chapter 3
Selecting the Type of Communication
This chapter helps you to select the type of communication for your specific task by briefly outlining the essential characteristics of different types of communication. The detailed descriptions of the possible types of communication can then be found in Chapters 4 and 5, each of which contains a specific description of configuring as well as a simple example.

Chapter 4
Acyclic Communication with SINEC Services
This chapter describes acyclic communication both with a detailed description of the individual FMS services and configuration procedures. At the end of the chapter there is an example to illustrate an application.

Chapter 5
Cyclic Communication
This chapter describes cyclic communication and configuration. At the end of the chapter there is an example to illustrate an application.

Chapter 6
Documentation and Testing
This chapter contains a description of the test and documentation functions referred to in earlier chapters.

Chapter 7
Request Editor
This chapter introduces the request editor utility. This tool is used for creating job buffers for the specific services for acyclic communication.

Chapter 8
Appendix
Here, you will find important information you require regularly, for example, the significance of error messages.

Chapter A and B
Abbreviations and Index
The list of abbreviations will help you considerably when working with this manual since you can check the meaning of unknown abbreviations. You can use the index to find a theme quickly.

Chapter C
Further Reading
This section lists publications and manuals dealing with related aspects (marked in the text with ***/x/***).❑

2 Basics of the PROFIBUS

To understand and to be able to work with the services correctly, you should be familiar with the PROFIBUS standard.
For further information refer to the DIN standard 19245.

The basic idea behind "open communication" is to allow automation systems of different vendors to communicate with each other. The specification describes how a message is exchanged. The meaning (semantics) of the information exchanged depends on the context (application or application processes). The negotiations about the application task must, for example, include the following:

- Which local objects (data) should be available to the other device with which services (operations on a defined class of objects).

- Which application associations are made available between the applications.

A measured value is, for example, an object belonging to the class of variables which can be addressed with variable services read, write and information report.

The PROFIBUS protocol stipulates the following:

- Useful data length: for read 237 bytes, for write and information report 233 bytes.

- No blocking, i.e. putting together of a number of short messages to form a long message.

- No routing via the network layer (layer 3 of the ISO/OSI model).

Within this framework, the following possibilities are open to the user:

➢ Access protection for objects is optional.

➢ Services can be staggered starting from a minimum configuration (profile endorsed by the PROFIBUS Users' Organization).

2.1 PROFIBUS Architecture <-> OSI Environment

The structure of the architecture which can be roughly divided into three layers (application, data link and physical layer) is illustrated in Fig. 2.1. The components are explained here briefly to clarify the meaning of the terms used.

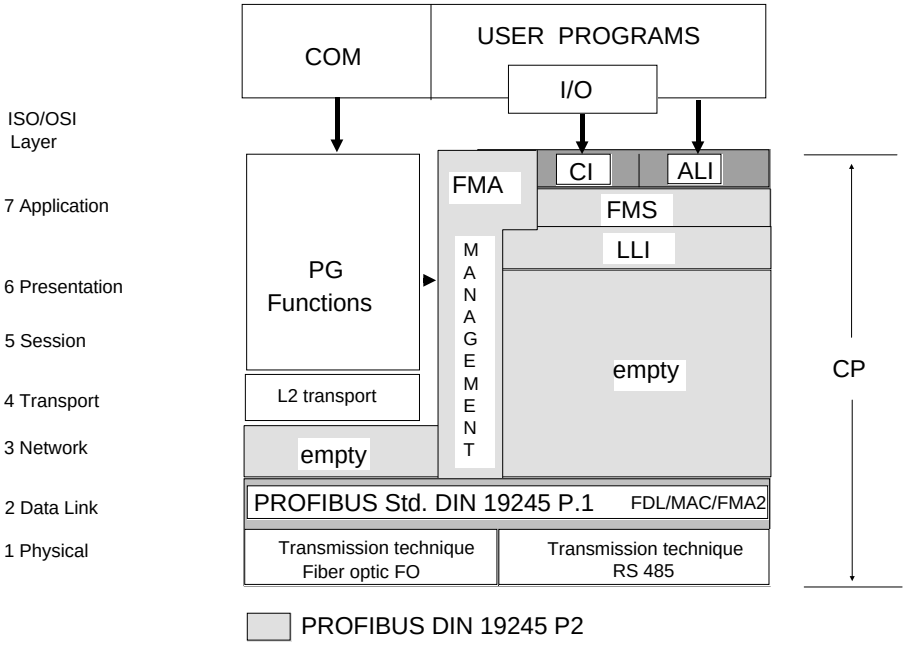


Fig. 2.1 Protocol Architecture for the FMS Part of the CP

Key:

ALI: Application Layer Interface
Driver forming the interface for the acyclic services between the SIMATIC S5 PLC and the application layer. This consists of two entities, FMS (field bus message specification) and LLI (lower layer interface). (Chapter 4)

CI: Cyclic Interface
Driver which forms the interface for the cyclic services between the I/Os of the SIMATIC S5 PLC and the application layer. (Chapter 5)

FMS: Fieldbus Message Specification
The FMS describes communication objects, services and the models resulting from them.

LLI: Lower Layer Interface
This entity forms the interface between FDL and FMS/FMA. The essential tasks of the LLI are as follows:
- simulation of FMS and FMA7 services on the FDL services
- application association establishment/termination
- application association monitoring

FMA: Fieldbus Management Layer
Describes objects and management services. The objects are manipulated locally or remotely with the management services divided into three groups, as follows:

Context management:
These are services for establishing/terminating a management application association (only remote).

Configuration management:
Identifies communication components of a station for loading and reading the application association list (AAL) and for accessing variables, statistics counters and parameters of layers 1/2.

Fault management:
Provides services for detecting and clearing errors and faults.

MAC: Medium Access Control

FDL: Fieldbus Data Link

2.1.1 PROFIBUS Communications Model (FMS)

The PROFIBUS communication model allows distributed application processes to be united as a virtual total process via application associations (refer to Fig. 2.2) (/9/).

From the point of view of communication, an application process includes all the programs, resources and tasks not assigned to a communications layer. This includes, for example, operating systems, real application processes, application programs, communication drivers and communications processors (ALI - Application Layer Interface, CI - Cyclic Interface).

In this example, the application processes are distributed on several different devices. More than one application process can exist on one device. An application process works with process objects (variables, programs etc.) which are described by attributes, rules and operations that can be used on it. The PROFIBUS communications model therefore supports modern, object-oriented functions of the application process, i.e. not "reading the field content of station B with address n via the bus" is the aim, but rather "reading the setpoint level of tank 1".

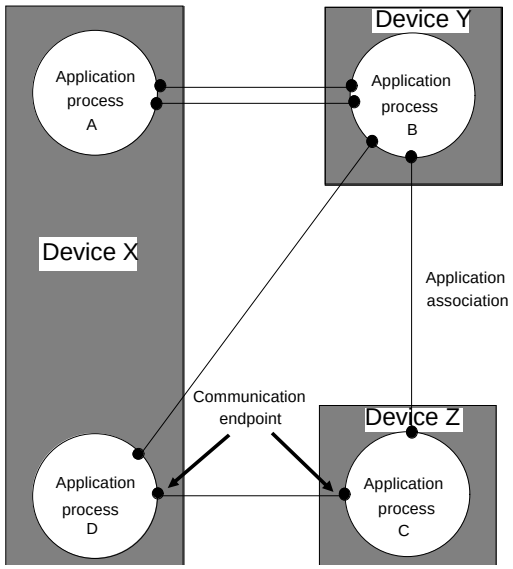


Fig. 2.2 Process Comprising Sub-processes A, B, C, D

2.1.1.1 Relationship between Application Processes

Logical application associations exist between application processes and are used to exchange information. These application associations must all be specified in the application association list (AAL) before data exchange begins. Access to an application process for communication is implemented via communication end points. One or more communication end points are assigned to an application process. The application process can address the communication end points using local device-specific communication references (address of the communication end point) (refer to Fig. 2.2).

2.1.1.2 Client and Server Associations

With message-oriented protocols, the application associations are based on the client server principle. Since the programmable controllers can be configured differently both physically and logically, these differences must be disguised to allow communication. The automation components or application processes are therefore concealed behind an abstract device (the virtual field device (VFD)) which allows the user to consider the components as uniform in terms of communication.
This principle defines two communications partners (refer to Fig. 2.3):

Client:
For communication purposes, a client is an application process which uses the functions of a virtual field device (VFD) of a remote application process.

Server:
The server is the application process that makes the functions of its virtual field device (VFD) available to the client.

☞ **An application process can always be both client and server.**

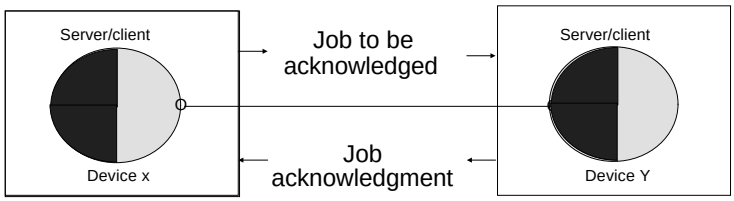


Fig. 2.3 Client - Server

2.1.1.3 Logical Data Exchange

Logical data exchange means the sending of messages (PDUs) via application associations (logical channels) to the communications partner. Jobs are issued using the FMS services. The data exchange is implemented using jobs on layers 7, 2 and 1.

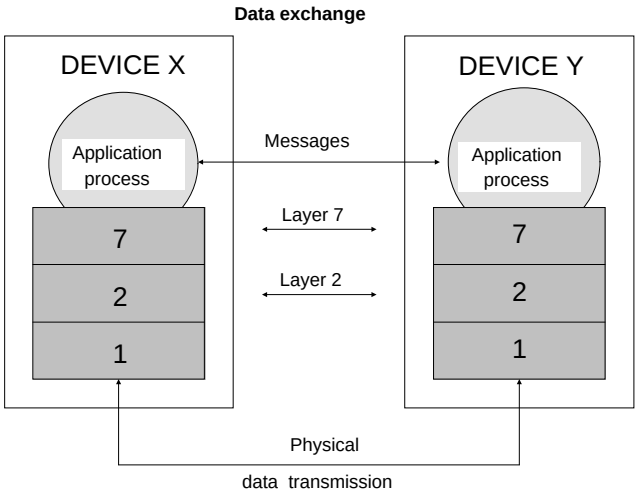


Fig. 2.4 Logical Data Exchange

The data on one side are passed down through all the layers from 7 to 2, transferred via the physical medium and passed up through all the layers from 2 to 7. This is invisible to the user. To the user, it appears as if the application processes exchange data directly with each other. On the same route (seen from the current layer) the peer layers also exchange data with each other.

2.2 PROFIBUS Application Layer with the CP 5431 FMS

One application layer of PROFIBUS is FMS. FMS is a protocol language for message-oriented communication. The FMS protocol not only standardizes the meaning of the message (semantics) but also the data formats for creating the message (syntax).

At the forefront of the model, there are terms which are essential for understanding PROFIBUS communication, as follows:

- Application Layer Interface (ALI), Cyclic Interface (CI)
- The virtual field device (VFD)
- The communication objects
- Access to objects
- The application associations
- Services

The following sections explain these terms and how they are simulated on the SIMATIC S5 PLC.

2.2.1 Application Layer Interface (ALI) and Cyclic Interface (CI)

The interface of the CP to FMS is divided into two blocks:

Application Layer Interface (ALI)

The ALI simulates the interface of the individual application process on the standardized interface of the application layer and vice-versa. ALI also includes the service management and management of objects for acyclic services. This aspect is the most important for the user and is therefore dealt with in detail.

Cyclic Interface (CI)

The CI provides an interface for cyclic services without object management. For each input area, an FMS read request is generated automatically and for each output area an FMS write request in each case on application associations with cyclic data exchange.

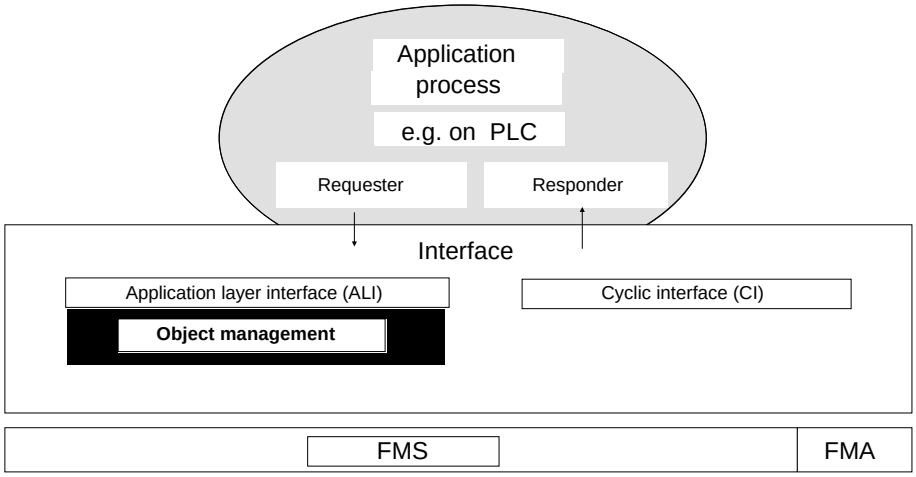


Fig. 2.5 Interface Between the Application Process and FMS

When an application process communicates with an application process on a different device, the local objects must be made available to the other process by means of services. Services are operations performed on objects of a defined class. For example: a measured value is an object belonging to the class of variables. The services allowed with this class are "FMS read" and "FMS write" (FMS information report).

ALI tasks are therefore the simulation of the local objects (process objects) on objects known to PROFIBUS (communication objects) and the specific coding of service jobs according to the PROFIBUS standard.

The task of the **CI** is to read or write to remote objects cyclically. Locally, the data are either written to the I/O area or read from it.

2.2.2 The Virtual Field Device (VFD Model)

The uniform view of a device is known as the virtual field device (VFD).This uniform view is characterized by a "public" object list provided by all stations on the bus, standardized device characteristics and identical services. The functions of the VFD are adapted by communication processes (e.g. ALI/CI for SIMATIC) for each device individually and simulated on the real process. This visible part of the application process that can be manipulated consists of process objects which are declared as communication objects by entering them in an object list. The user can address these communication objects of a partner on the bus.

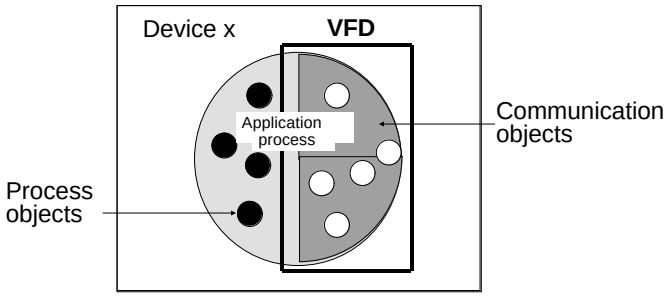


Fig. 2.6 Assignment of the VFD to the Application Process

Example: the object "variable" is described by an index, a data type and its access rights. The content of this object can be read or written only using the services "read" and "write" and specifying the index. VFDs are addressed using the addresses stored in the FMS application association list (see Section 2.2.3). The communication between the application processes of the devices results in the exchange of process objects on the communication system. These manipulations via the network can only be made on this virtual device which in terms of the network responds identically to all other devices, using the FMS services. This means that the process objects used for communication must be made known to the communications system. This is achieved by entering all the communication objects in an object list (OL).

The VFD therefore represents a model-like, standardized simulation of the physical device.

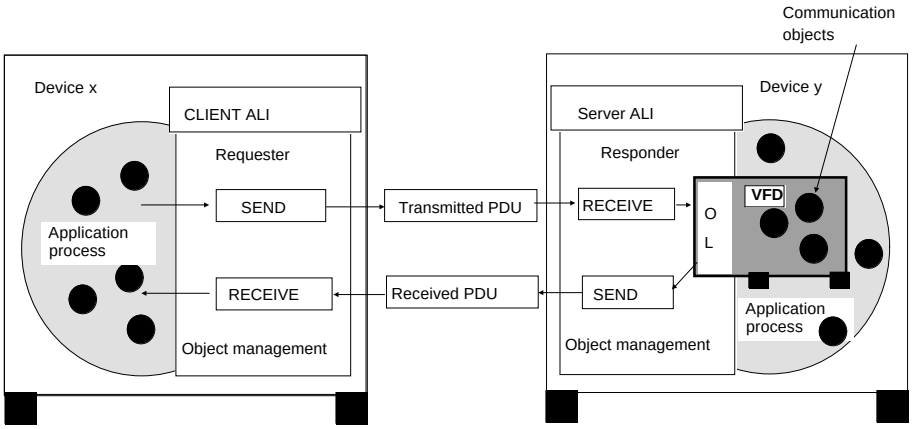


Fig. 2.7 VFD Model

SIMATIC S5 and VFD

The essential function of the communications processor (CP) is the simulation of the VFD on the programmable controller.

Viewed from the network, each communications processor and the programmable controller in which it is installed is considered as a single VFD. A VFD always contains one communications processor and (in the case of multiprocessor PLCs) up to four CPUs. It is, nevertheless, possible to use several CPs in one PLC rack. Since the communications processors all work independently of each other, each can be considered a VFD itself.

2.2.2.1 Communication Objects

Communication objects form the basis of a common exchange of information via communication systems. Objects have attributes (data type, access rights, etc.). Specific services operate on each communication object. If a service processes a communication object, the response of the object is defined by the rules of the PROFIBUS model.

An object description is assigned to the communication objects. The object description of all communication objects is entered in the object list (OL) of a station (refer to Section 2.2.2.3). There is one or more VFDs per object list. The CP 5431 FMS only supports one VFD per object list. By entering a process object in an object list, it becomes a communication object and can be used by remote application processes.

FMS distinguishes between two types of object:

- Static types of object
  Static communication objects are created when the device starts up. They can neither be modified nor deleted during operation.

- Dynamic types of object
  These are objects that can be created during operation. **These types of object are not supported on the CP 5431 FMS.**

A VFD consists of the following communication objects:

Variable:

Variables are objects that simulate the data of a user program. Variables are identified by names (indexes) and contain a description of their structure. The description allows a standardized representation of the data, uniform throughout the system. The structure of the data can be either simple (simple variable) or complex (array or record). Variables can be read or written to via the communication system.

Data type

A type is assigned to each variable object. There are standard data types and freely configured types. Data types are created when the variables are configured and are linked to the variable.

The static variables and the data types assigned to them are configured with COM 5431 FMS.

In the SIMATIC S5, data are accessed by a STEP 5 user program exclusively using variables assigned an index.

The CP 5431 FMS supports the following static communication objects:

- Variable with standard data type:
  these elements are of a predefined data type.

- Variable (record) with structured data type:
  a group of structured components.

- Variable (array) of all standard data types:
  a series of elements with the same structure.

- With complex variables, a nesting level up to 2 levels is possible.
  Records with structures as components or arrays with structures as components are permitted.
  Variables of the array object type or record object type with components of the array type are not permitted.

Standard data types:
The standard data types described below are standardized in the PROFIBUS and are supported by the CP 5431 FMS:

- Boolean

- Integer (8, 16, 32 bits)

- Unsigned (8, 16, 32 bits)

- Floating-point (IEEE Std. 754 - short real number with 32 bits on the bus)

- Visible string (ISO 646)

- Octet string (binary coding)

➢ Bit string (according to PROFIBUS standard converted from octet units to bit units)

On the bus, the variables consist of one or more octets, the number being decided by the type. An octet is 8 bits long, the bits being numbered from 1 to 8. Bit 1 is the LSB (least significant bit).

Object attributes

The characteristics of the objects are the object attributes. They are represented by a formal description and form the basis of the standardized access to objects in the open communication system.

For the CP 5431 FMS these are:

➢ Index:
logical address of the object

➢ Password:
contains the password for access protection mechanisms

➢ Access Rights:
information about the permitted type of access, e.g. read for all

2.2.2.2 Access to Objects

The index is the key for accessing objects and object descriptions. These indexes are listed on the field device in the object list. Apart from the index, the data type and possibly also the length of the object is stored. The exchange of useful data therefore only requires the index for the service job. First order components (nesting level 1) of a complex object can be accessed directly using the subindex.

The services always address the object of the server, since that is where the real object exists.

2.2.2.3 The Object List

The object list (OL) is divided into a local source OL and a remote OL.

The **source OL** contains the object description of the locally defined objects.

The **remote OLs** are copies of source OLs from other partners. They are requested implicitly from the partner with the FMS service Get OL.

The object descriptions are defined when the network is configured and stored on the station in which the object exists (source OL). They can be requested later by any other station. The length of the object description may vary for different objects. The communications partners maintain a complete or partial copy of the requested remote object descriptions (remote OL). This allows every station to have a source OL of locally existing communication objects and one or more OLs for remote objects.

If a source OL is modified by a remote communications partner, the station must inform all other partners using the same source OL to ensure consistency. This is achieved by terminating all the application associations involved.

With the CP 5431 FMS, a remote object list is read automatically each time an application association is established. This assumes that the index of the objects to be read is specified when the application association is configured, otherwise Get OL is not used.

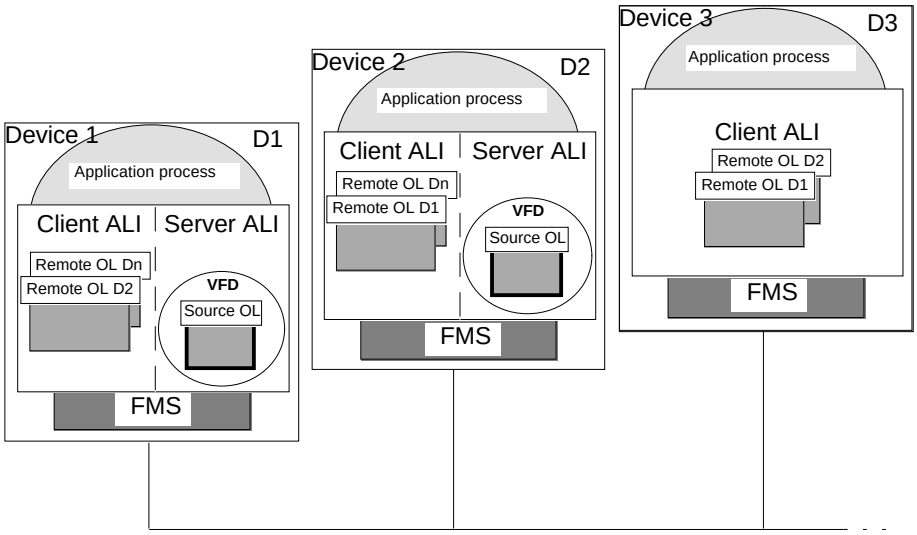


Fig. 2.8 Source und Remote Object Lists

2.2.3 Application Associations

From the point of view of the user, communication with the application processes of the communication partners takes place via logical channels. These logical channels to the communication partners are defined in the configuration phase in the application associations list (AAL).

The application associations list (AAL) of a station contains the description of all the application associations of this station with other stations regardless of when they are used. The AAL is created during configuration for each specific PROFIBUS station and is loaded locally or remotely using the management services. For SINEC CPs the NCM transfer services are used.

For each application association, the following information is stored in the AAL:

➢ address of the remote station

➢ local and remote service access point

➢ type of application association (with CP 5431 FMS acyclic or cyclic application associations.)

➢ application association attribute (with CP 5431 FMS only defined application association)

➢ type of application association (e.g. master/slave for cyclic data exchange without slave initiative (CI))

To access this information, communication references assigned automatically by the configuring tool are used.

CP 5431 FMS and communication with application processes

When you configure the CP 5431 FMS, you assign an SSNR and ANR and allocate them to a communication reference. The combination of these two (SSNR/ANR) forms the interface to the application program.

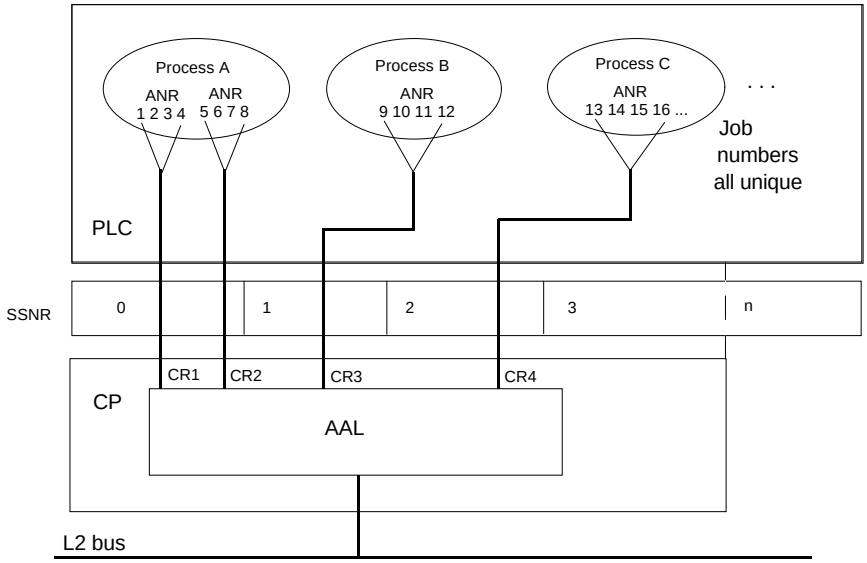


Fig. 2.9 Addressing Model for Communication on the CP

The CP can handle application associations of the following types:

- "One-to-one", connection-oriented.
- "One-to-all", connectionless.

2.2.3.1 Connection-Oriented Application Association

With a connection-oriented application association, a logical "one-to-one" connection between two communications partners is established as defined in the configuring phase (defined application association).

This connection-oriented communication involves the following phases:

➢ the establishment or initiation phase,

➢ The data transfer phase and

➢ The termination phase.

The sequence is as follows:

During the establishment phase, an application association establishment request is sent to the remote application process (initiate service). The establishment request includes the services to be used in the data transfer phase, the maximum frame size, the number of parallel services (context) supported by the partner and the required type of application association as well as any other options required on the application association.

If the remote application process agrees to the establishment request, it sends a confirmation to the initiator. Following this, both application processes are in the data transfer phase and can communicate with each other according to the terms negotiated (context).

The availability of an existing application association is monitored by the LLI monitoring functions. The selected monitoring times must be the **same** on both the local and remote end points.

An application association is cleared by a termination function. This may be triggered automatically if errors occur.

2.2.3.2 Connectionless Application Association

The connectionless application association corresponds to "one-to-all" (broadcast) communication. "One-to-many" (multicast) communication is not supported by the CP 5431 FMS.
The broadcast application association is always in the data transfer phase.
The context check is omitted in connectionless application associations. The application association is not monitored. The execution of a broadcast job cannot be acknowledged by the remote application processes, so that only unconfirmed services (information report) are permitted.
You can only configure one application association of the broadcast type on a virtual field device.

2.2.3.3 Types of Communication

The CP 5431 FMS supports the following communication types stipulated by PROFIBUS:

➢ with master-slave application associations:

- Application associations for cyclic data exchange without slave initiative.
- Application associations for acyclic data exchange without slave initiative.
- Application associations for acyclic data exchange with slave initiative.

➢ With master-master application associations:

- Application associations for acyclic data exchange.

➢ Connectionless broadcast application associations.

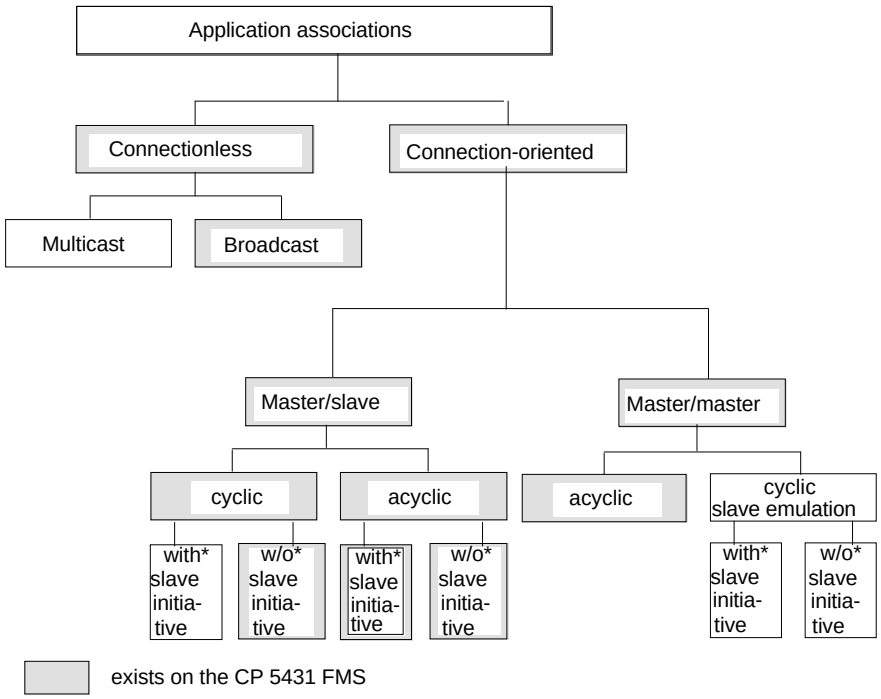


Fig. 2.10 Types of Application Association

* Slave initiative:
A slave device can send an unconfirmed service request.

A **master**
can access the bus actively, i.e. it can send information on its own initiative.

Slaves without initiative
are only permitted passive bus access, i.e. they can only transmit information at the request of a master.

Slaves with initiative
can also initiate unconfirmed FMS services with selectable priority. The CP 5431 FMS supports this type of communication as receiver of an FMS information report (see Section 4.2.2.1 "Information Report").

If there is more than one master, rules must be established for bus access. In the PROFIBUS, this is achieved by a token passing technique, in which the right to bus access (token) is passed on from master to master.

During active bus access, a master can poll slaves, this means that the slaves receive the passive bus token one after the other.

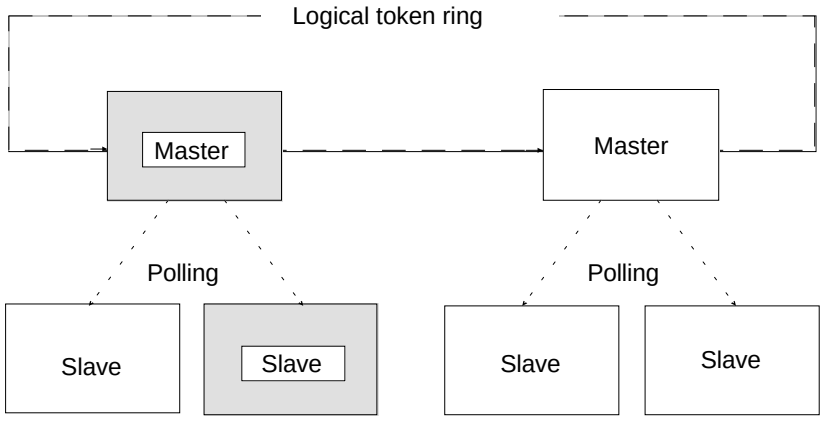


Fig. 2.11 Hybrid Bus Access Technique with PROFIBUS

2.2.4 FMS Services and How They are Modeled on SINEC Services

With the FMS services, an application process can use the server functions of a remote application process. The FMS services are transferred to the communications partner with protocol data units (PDUs). The individual FMS services are part of the communication models which define the sequence and rules for using services. In a communication model, the FMS services operate on communication objects. Certain services use or process only certain communication objects. There are services which are explicitly confirmed by remote application processes following execution (confirmed services) and others which are not confirmed after execution by the application process (unconfirmed services). The parameters required to use the service are either provided by the application process or are configured in advance.

The following services must be available as server services in all PROFIBUS devices:

Management services

- Initiate
- Abort
- Reject
- Status
- Identify
- Get OL (short form)

A service can only be used to address one object. This object itself can, however, consist of several objects. With a read job, a maximum of 237 bytes of useful data can be transferred and with a write or information report job, a maximum of 233 bytes.

In the ALI, the SINEC services triggered by the SIMATIC user are re-coded as FMS services.

In the CI, an FMS read request is generated for each input and an FMS write request for each output, in each case with cyclic data exchange.

Simulation on SINEC services:

SINEC service		**FMS service**
Read variable	<->	FMS read
Write variable	<->	FMS write
Information report	<->	FMS information report
Identify	<->	FMS identify
Status	<->	FMS status

2.2.5 Access Protection Mechanisms

Access protection within automation systems is intended to provide operational safety in the plant. The PROFIBUS standard defines various access protection mechanisms to meet these needs.

The communication object can be protected by configuring protection mechanisms in the object list. The CP 5431 FMS only supports access protection by means of a password. ❑

NOTES

3 Selecting the Type of Communication

In Chapter 2.2, it was mentioned that there are various mechanisms available for data transmission which can, in practical terms, be divided into two different types as follows:

➢ Data transmission using cyclic communication (with CI)

  This type of communication is always suitable when values of a CI slave are only to be read or written cyclically. The jobs for cyclic processing must be configured. Only the variable values are exchanged between the PLC and CP. The CP automatically generates the FMS-PDUs.

➢ Data transmission using acyclic communication (with ALI)

  This type of communication is more suitable when the selection and timing of the required services is controlled by the user program. Job buffers with a job description and possibly data are exchanged between the PLC and CP. The PDU is created based on the content of the job buffer

This chapter contains basic information about the different types of communication to help you select the type of data transmission most suitable for your applications.

3.1 Data Transmission using Cyclic Communication

For data transfer via the I/Os, the CP 5431 FMS provides the cyclic interface (CI). When using the services, the FMS infrastructure is available to the user. This ensures, for example, increased reliability by logical acknowledgment of frames and the timed and logical monitoring of FMS jobs. The jobs to be processed cyclically are entered in the polling list.

The cyclic interface has the following characteristics:

- Only connection-oriented application associations are supported between the master and slave without slave initiative.
- With this type of application association, the CP as master can access the bus actively and therefore transmit messages itself.
- The CP can only be the client.
- All PROFIBUS-compatible stations which support master-slave operation with cyclic data exchange can be used as communications partners.
- The data structures are variables of standard types including arrays and records with data field lengths of 1 to 32 bytes.
- The data transmission is always synchronized with the cycle, i.e. the time at which the data transmission is made is dictated by an HDB call in the PLC program.

From the point of view of the client, the following FMS services are required of the server:

FMS Initiate, FMS Abort, FMS Get OL, FMS Reject

FMS Read: When this service is called, the client requests data from the server. The job frame itself must not contain data. The client obtains the data from the server in the acknowledgment.

FMS Write: With this service, the client transfers data to a server. The client receives an acknowledgment of the received data.

FMS Status: The client instructs the server to inform the client of the server's current status.

FMS Identify: Prompted by the client, the server informs the client of its device identifier.

This is the minimum range of functions of a server according to the PROFIBUS standard DIN 19245 Part 2, extended by the READ and WRITE services.

3.2 Data Transmission using Acyclic Communication

For acyclic communication with the SINEC services, the CP 5431 FMS provides the ALI. With this interface, general access to data is possible in a communications network consisting of various PROFIBUS systems. To allow this general access despite different end systems, the data is simulated on variable objects (known simply as variables).

With the SINEC services, the user can write or read variables defined on a server via the communications system. In the server role, the device can inform the client of the content of its variables. The variables are accessed using their indexes (logical addresses).

For this reason, a distinction must be made between two situations:

- The CP 5431 FMS is a client wishing to access variables defined on a different station.

- The CP 5431 FMS is a server managing variables defined locally on its own station.

The ALI has the following features:

- The CP can be both server and client.

- The application association can be either a master/master or master/slave relationship.

- When acting as master, the CP can access the bus actively and send frames on its own initiative.

- Acyclic application associations (master/master or master/slave) without slave initiative or master/slave with slave initiative are handled by the ALI as one-to-one application associations between two communications partners.

- The ALI can transmit objects to all stations in the ring on a connectionless broadcast application association (SINEC service "information report") or can have objects reported to it from every active station. As the client, the master always uses the FMS unconfirmed service "information report.request" with high priority.
  On its server interface, the master can receive an "information report.indication" with low or high priority.
  A maximum of 16 bytes of data can be transmitted.

- All PROFIBUS-compatible stations can act as communications partners of the CP 5431 FMS.

- The data structures are variables of the standard types, as well as arrays and records with data field lengths of 1 to 237 bytes. Records can contain up to a maximum of 63 components.

- Alternative access to single components of arrays and records are possible in the subindex range from 1 to 255. Subindex 0 identifies the complex variable itself. Components of nesting level 1 of a complex variable are be referenced using the subindex.

- The CP allows up to 32 acyclic (master-master, broadcast or master-slave) and up to 32 cyclic application associations (master-slave), however, in total only 48 application associations.

- A maximum of 256 local (VFD) variables can be configured. The exact number depends on the variable types, on the available RAM memory with such types and the size of the memory submodule you are using.

- The assignment of a variable to the real object in the SERVER (e.g. a data block) is configured on the SIMATIC PLC using COM 5431 FMS (VFD variables). The conversion when the variable is accessed is the responsibility of the CP.

- To allow the SIMATIC S5 (client) to access variables, the index and description of the object are necessary. The index must be specified in the job buffer. The object description can be either contained in the job buffer directly or is loaded automatically by the remote communications partner using the "get OL" service when the application association is established. In the second case, the type description does not need to be specified in the job buffer, however, the index of the variable must

be specified when the application association is configured on the client.

➢ A maximum of 32 different communication objects can be reported to the CP 5431 FMS. The CP requires a configures assignment table (report variable) of the remote indexes to the local virtual field device (VFD).

➢ If the SIMATIC PLC is acting as the server, the variables are managed by the CP, i.e. the variable attributes are stored on the CP.

The following SINEC services are implemented on the CP and they are simulated on FMS as follows:

Read variable: FMS read
When this service is called, the client requests data from the server. The job frame itself must not contain data. The client obtains the data from the server in the acknowledgment.

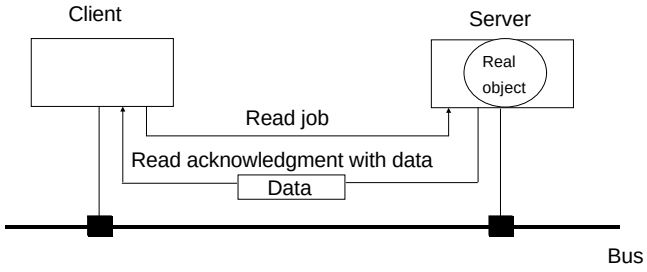


Fig. 3.1 "Read" Variable Service

Write variable: FMS write
With this service, the client transfers data to a server. The client receives an acknowledgment of the received data.

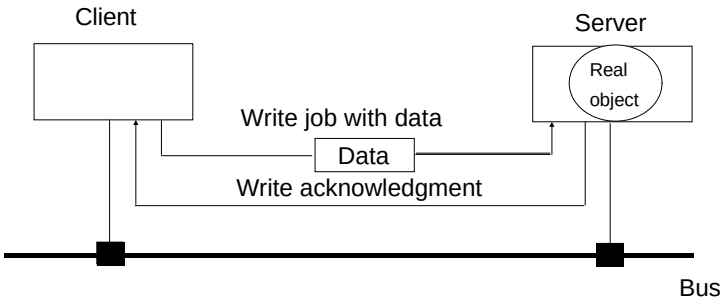


Fig. 3.2 "Write" Variable Service

Information report: FMS information report
The server reports the content of one of its local objects to the client without being requested to do so by the client. The most important feature of this service is that the server initiates the service (requester). This service is not acknowledged by the client as the service receiver (unconfirmed FMS service).

The CP 5431 FMS supports this service as requester and receiver.
A typical application:
A slave with initiative (requester) reports a process or event variable of the slave to the CP (receiver).

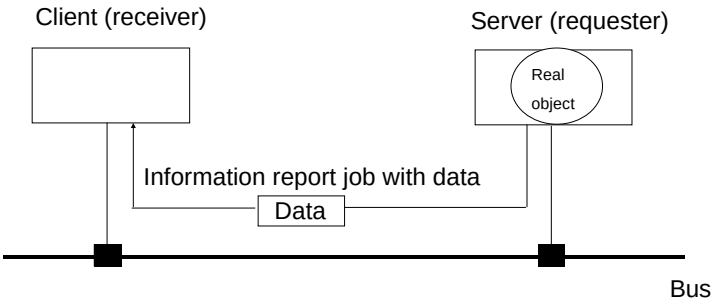


Fig. 3.3 "Information Report" Variable Service

General services for VFD:

These general services allow a client to request and then process information about the status or attributes of the virtual field device (VFD) on the server. The following services are included in the client and server functions.

Read status: FMS status
With this service, a client requests information about the physical status of the VFD. The server returns the required information

Identify virtual device: FMS identify
With this service, a client can request information about attributes of a VFD, e.g. the vendor's identifier for the VFD, the device ID and the version of the CP.

FMS initiate, FMS abort, FMS get OL, FMS reject are implicitly supported by the CP, however, they are not simulated on SINEC services.❑

NOTES

4 Acyclic Communication with SINEC Services

The communication between user programs and the ALI makes use of SINEC services. The conversion of SINEC services to FMS services and the management of the object list for the bus takes place in the ALI. If the ALI receives a job (always specified in a job buffer) from the PLC (client interface), it creates an FMS request field and sends this for further processing to the remote ALI (server interface).

The ALI provides the interfaces "CLIENT" and "SERVER".

ALI client interface

The services initiated by the client (S5 program) using job buffers are simulated on FMS services. The CP provides the SINEC variable services **read**, **write** and **information report**. Since FMS does not support addressing using names for SIMATIC S5 but only addressing using the index, this part of the job buffer has the appropriate structure. The job buffer must be completed by the user. To make the creation of the job buffer as easy as possible, COM 5431 FMS includes a request editor (Chapter 7).

To allow the PLC to issue jobs, it must call the SEND direct handling block with the appropriate parameters (SSNR and ANR which define the application association) for each job. The ALI then handles each job on the corresponding application association. Up to 4 jobs can be activated simultaneously on an application association.

The first step is to transfer the job buffer to the CP with the HDB "SEND direct". If the job buffer is longer than 256 bytes of data, the job is aborted by the CP. With certain jobs, the job buffer may already contain the data required for the service. The CP does not segment data and the data for a job must therefore not exceed 237 bytes for read and 233 bytes for write and information report.

After calling the "Send direct", the CP sets the status of the job (status byte in the dual-port RAM) to "job active". Depending on the service (see sequence of the individual services), the CP requires data from the PLC or must transfer data to the PLC. To achieve this, appropriate jobs are transferred to the CP and processed by the PLC when the SEND ALL or RECEIVE ALL block is called. On completion of the job, the status is set to "job complete with/without error" by the CP. A confirmed service is only complete when the layer 7 acknowledgment of the remote ALI is received.

☞ **To allow all services to run correctly, the SEND ALL and RECEIVE ALL block must be called at least once in the PLC cycle per interface.**

Before data can be transferred from the PLC to FMS, or from FMS to the PLC, a type conversion may be necessary (refer to Table 4.1).

The structure of the status word ANZW is represented in Fig. 4.1 and is described in greater detail in Section 4.3. In case of errors, the third word contains the FMS error code.

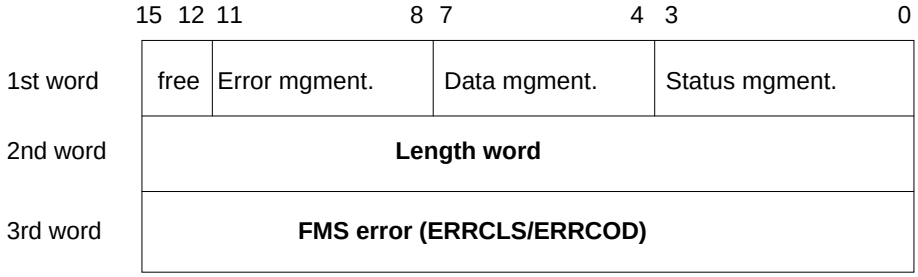


Fig. 4.1 Status Word

	FMS representation	S5 representation
Boolean	8 bits	16 bits
Floating point	IEEE	MC 5 (KG)
Bit string	ASN1 "bit string"	SIMATIC "bit string" KY

Table 4.1 Type Conversion on Server

☞ **You can disable or enable access to local variables using the status word specified when the variables were configured.**

Server as requester

The "information report" job represents a special case in the client/server model. The server triggers a job using the ALI client interface. In terms of the VFD model, it nevertheless remains server, since the communication object to be reported is formed from local process objects.

The communication object must not exceed a data length of 233 bytes (with alternative access 231 bytes)(low priority). With unconfirmed jobs with high priority, the maximum data length is 16 bytes. The reported data is transferred to the application process by the receiver using the ALI server interface.

ALI server interface

If the ALI receives a job from the FMS (server interface), it interprets the job and executes it. For variable services in the PLC program, only the required SEND ALL and RECEIVE ALL handling blocks must be called. The general services are interpreted and executed by the CP alone.

Interrupt-driven processing

Reported data can be transferred to the PLC by the CP using the PLC interrupts. The CP triggers an interrupt in a CPU depending on the selection made when configuring (Interrupts: IRA-IRB-IRC-IRD). The PLC is instructed to interrupt cyclic processing and to run a specific routine in an interrupt block. The interrupt routine contains the call for a receive direct handling block. The receive direct block fetches the received data from the CP and transfers it to the PLC. The destination address is supplied by the CP. The configuration steps and description of the sequence are explained in the "information report" SINEC service description and in Section 4.2.2.1.

4.1 Basics of the SINEC Services

4.1.1 Read Variable, Client Interface

Job buffer "Read"

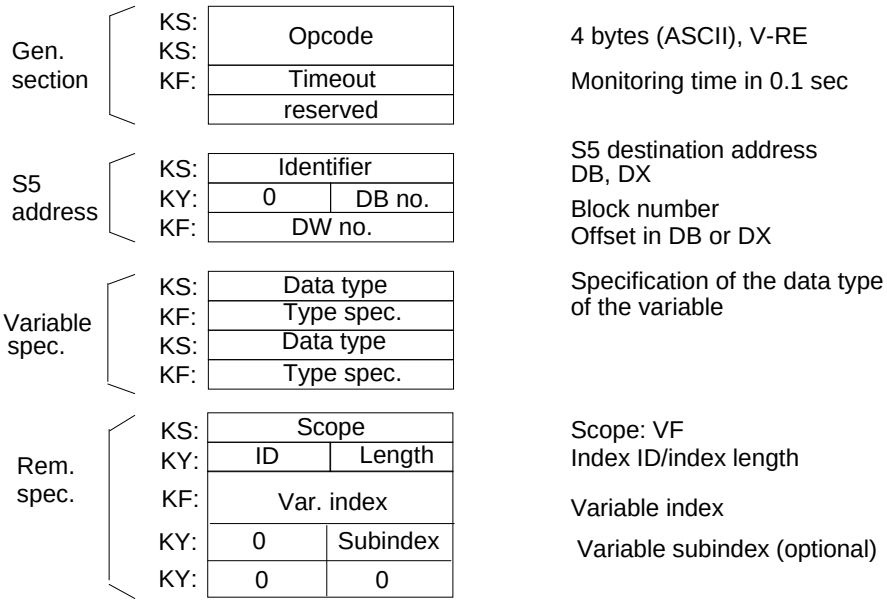


Table 4.2 Structure of the Job Buffer for the "Read" SINEC Variable Service

Description of the call

General section:

Opcode: V-RE

Timeout: 1 word, format: KF
Specifies the maximum length of time the user program will wait for an acknowledgment for the service (i.e. the maximum dwell time of the job in the CP). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be

completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 address:

Identifier: 1 word, format: KS
range of values: DB, DX
DB for data block
DX for extended data block
local object
Description of the local destination address for the service, the DA identifier is not allowed.

DB no.**:** 1 word, format: KY
range of values: high byte: 0, low byte: 1-255
low byte: DB or DW number

With identifier DB or DX the DB no. must never be 0. In PLCs that use DB 1 for system settings, you must make sure that DB 1 is not overwritten by an FMS service.

DW no.: 1 word, format: KF
range of values: 0 - 2042

This is an offset within the data block or extended data block. Owing to the current PG software, the maximum DW number is restricted to 2042. The "length" parameter required for the complete definition of an S5 address is not specified. This is calculated in the CP from the type information for the variable.

Specific to the variable:

Data type: 1 word, format: KS
permitted values: see Table 4.3 "SINEC types"

This defines the data type of the addressed communication object. In the job buffers, only simple types (standard types) and arrays of simple types are supported. These can be coded in four words. If the variable is more complex, the type description must remain unspecified (blanks entered) In this case, the user must enter the variable index in the "access to variables" field in the configuration screen. The client fetches the type description of the variable directly from the server with a "Get OL". In the client configuration data for the application association, the required variable index must be specified. When the application association is established, the type description of the required index on the server is loaded automatically with a "Get OL". The variable description is then also known to the client.

Type specification	1 word, format:	KF
	Basic data types:	number of bits in an object of this type
	String types:	number of bytes in a string
	Arrays:	number of elements in an array
	permitted values:	see Table 4.3 "SINEC types"

The following more complex data types are supported but must be configured for the client access in the application association screen by specifying the index:

- Structures (records) with components of the basic data type.
- Structures with components containing structures with components of the basic type.
- Arrays with elements that are structures whose components are of the basic data type.

SINEC type	Type description	Meaning	Corresp. to S5 type
BO	no entry	Boolean	-
IN	8	integer, 8 bits	-
	16	integer, 16 bits	KF
	32	integer, 32 bits	-
UN	8	unsigned, 8 bits (unsigned number))	-
	16	16 bits	KH
	32	32 bits	KD
FP	32	floating point no. in MC 5 format, 32 bits	KG
BS	1 to 1864 (1896)*	bit string, no. of bits in string	KM
OS	1 to 233 (237)*	octet string, no. of bytes in string	KY
VS	1 to 233 (237)*	visible string, no. of bytes in string	KS

* Values in brackets apply to read jobs

Table 4.3 SINEC Types

Explanation of the SINEC types / representation of the types in SIMATIC S5:

BO:
Boolean

Boolean variables are modeled on the CP on a data word in the data block.
The following values are permitted:
0000_H "False"
$FFFF_H$ "True"

IN:
Integer 8

The FMS data type integer 8 (representation 1 byte) is modeled by the CP on a data word in a data block (format KF). The range of values for variables of this type is from -128 to +127.

UN:
Unsigned 8

The FMS data type unsigned 8 is modeled by the CP on a data word in the data block. The range of values is from 0 to 255.

FP:
floating point

Floating point numbers are stored by the CP in MC5 format in the SIMATIC PLC.

BS:
bit string

The "bit string" data type is modeled on the type "KM" known in the SIMATIC PLC. If the number of valid bits can be divided by 16 (number of bits in a data word), it is modeled directly on the resulting number of data words. Otherwise, the string is extended to the next word limit, and the bits appended must not be used in the PLC program. Every byte of the S5 representation is the "mirror image" of the FMS-(=ASN1) representation.

☞ **All string types are stored in the SIMATIC PLC in ascending memory addresses. This means with data blocks that first the high byte (this is located at the lower memory address) and then the low byte is written to.**

OS:
Octet string

The octet string represents a string of bytes (with any content). It is modeled on the SIMATIC format (KY). Storage is word-oriented and if necessary a padding byte is added.

VS:
visible string

The handling in the CP is the same as with the octet string data type, however, the range of values is restricted to representable ASCII characters (format KS).

The format and type conversions specified here are also carried out at the server interface, so that the user is unaware of differences in the data representation.

The user can also define not only a single element of one of the types specified above, but also arrays of these types. To do this, the ASCII characters "AR" (for array) must be entered in the first data word of the type description in the job buffer and the number of elements in the array (repetition factor) in the second data word. The third and fourth data words contain the definition of the data type of the array elements.

With standard data types, the third and fourth words are irrelevant.

Examples of configuring variables

➢ 16 bit integer:
1st index: 100
type: IN 16

➢ Array of 21 elements 32 bit floating point numbers:
2nd index: 101
type: AR 21
type: FP 32

➢ Boolean value:
3rd index: 102
type: BO

If the type of variable is not specified in the job buffer, the variable index must be specified when you configure the application association on the client. This is configured with the COM (application association configuration).

Specific to the remote device:

Scope: 1 word, format KS, scope of the variable, here only VF.

Index ID and index length: 1 word, format: KY. The IDs indicate whether or not the object access uses variable subindex addressing.

	Index ID	Length of the following index description in bytes
Only index	2	2
Index and subindex	3	4

Var. index: 1 word, format KH, index of the variable

Subindex: 1 word, format KY
range of values: high byte: 0, low byte: 0 - 255
Subindex: component of a variable. Specifying the subindex is optional. Subindex 0 identifies the variable itself.

Description of the sequence (Read, positive acknowledgment)

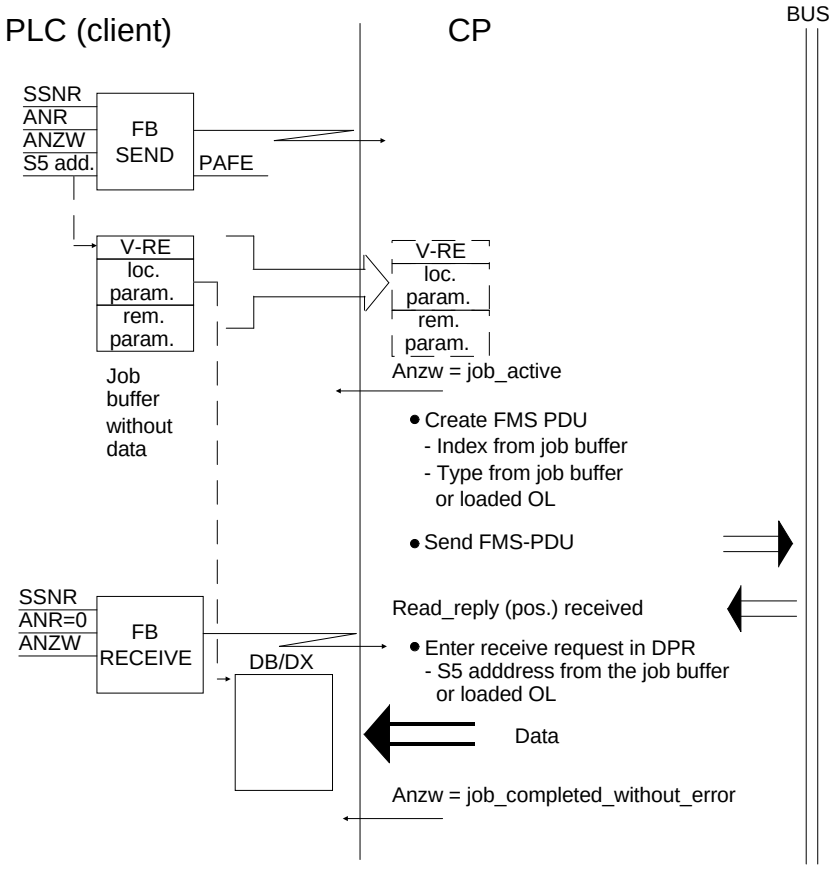


Fig. 4.2 Sequence Description (Read, Positive Acknowledgment)

Description of the sequence (read, negative acknowledgment)

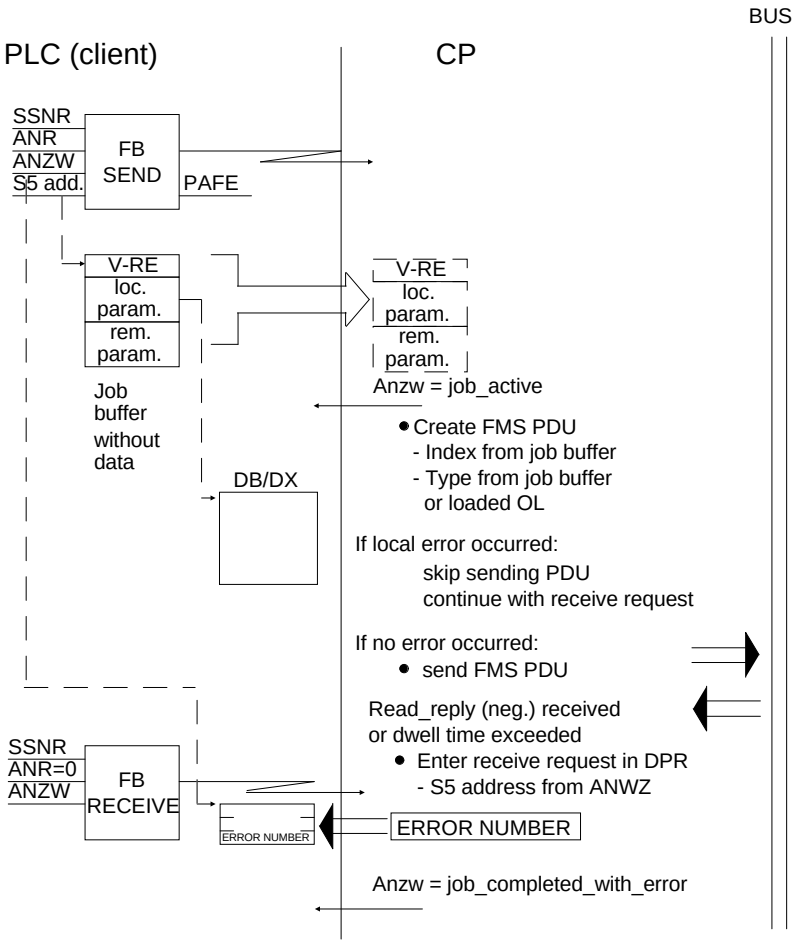


Fig. 4.3 Sequence Description (Read, Negative Acknowledgment)

4.1.2 Write Variable, Client Interface

Job buffer "Write"

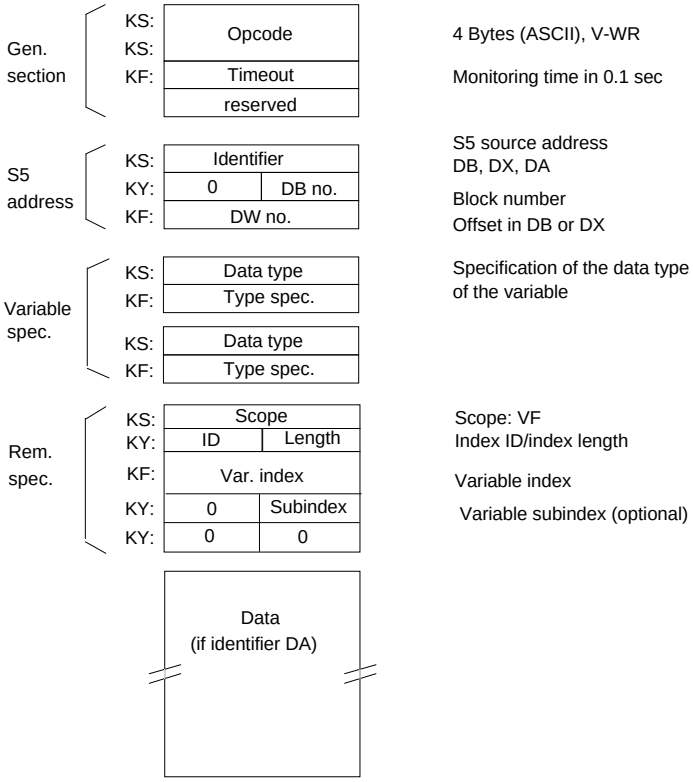


Fig. 4.4 Job Buffer for "Write" SINEC Variable Service

Description of the call

General section:

Opcode: V-WR

Timeout**:** 1 word, format: KF
Specifies the maximum length of time the user program will wait for an acknowledgment for the service (i.e. the maximum dwell time of the job in the

CP). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 address: Description of the local source address or specification that the data are located in the job buffer.

Identifier**:** 1 word, format: KS
range of values: DB, DX, DA
DB for data block
DX for extended data block
DA for data in job buffer

Note on the "DA" identifier:
Apart from the SINEC service specification and the required parameters, the S5 user program can also transfer the data completely to the CP. This is possible when the specified S5 address is a data source. This allows a considerable increase in the data throughput, as can be seen in the description of the sequence (see Fig. 4.6). When using this option, remember that a job buffer must not exceed 256 bytes. The data must follow on immediately after the last valid parameter of the job buffer.

If the DA identifier is specified the next two words are invalid:

DB no**.:** 1 word, format: KY
range of values: high byte: 0, low byte: 1-255
low byte: DB or DW number

With identifier DB or DX the DB no. must never be 0. In PLCs that use DB 1 for system settings, you must make sure that DB 1 is not overwritten by an FMS service.

DW no.: 1 word, format: KF
range of values: 0 - 2042

This is an offset within the data block or extended data block. Owing to the current PG software, the maximum DW number is restricted to 2042. The "length" parameter required for the complete definition of an S5 address is not specified. This is calculated in the CP from the type information for the variable.

Specific to the variable:

Data type: 1 word, format: KS
permitted values: see Table 4.3 "SINEC Types"

This defines the data type of the addressed communication object. In the job buffers, only simple types (standard types) and arrays of simple types are supported. These can be coded in four words. If the variable is more complex, "OS" must be specified as the type in the job buffer. The length of the octet string must correspond to the length of the variable accessed on the server. If the type description remains unspecified (value 0), the client must fetch the type description of the variable directly from the server with a "Get OL". In the client configuration of the application association, the required variable index must be specified. When the application association is established, the type description of the required index on the server is loaded automatically with a "Get OL". The variable description is then also known to the client. This allows the type description to remain unspecified in the job buffer.

Type specification: 1 word, format: KF
Basic data types: number of bits in an object of this type
String types: number of bytes in a string
Arrays number of elements in an array
Permitted values: see Table 4.3 "SINEC Types"

Specific to remote device:

Scope: 1 word, format: KS
Scope of the variable, here only VF.

Index ID and index length: 1 word, format: KY. The IDs describe whether or not the object access uses variable subindex addressing.

	Index ID	Length of the following index description in bytes
Only index	2	2
Index and subindex	3	4

Var. index: 1 word, format: KH (with certain restrictions KF also possible), variable index

Subindex: 1 word, format KY
range of values: high byte: 0, low byte: Subindex: 0 -255,component of a variable. Specifying the subindex is optional. Subindex 0 identifies the variable itself.

Data: (Only with source ID = DA), the CP expects the data values to be transmitted according to the data type description.

Description of the sequence (write with DB or DX, positive acknowledgment)

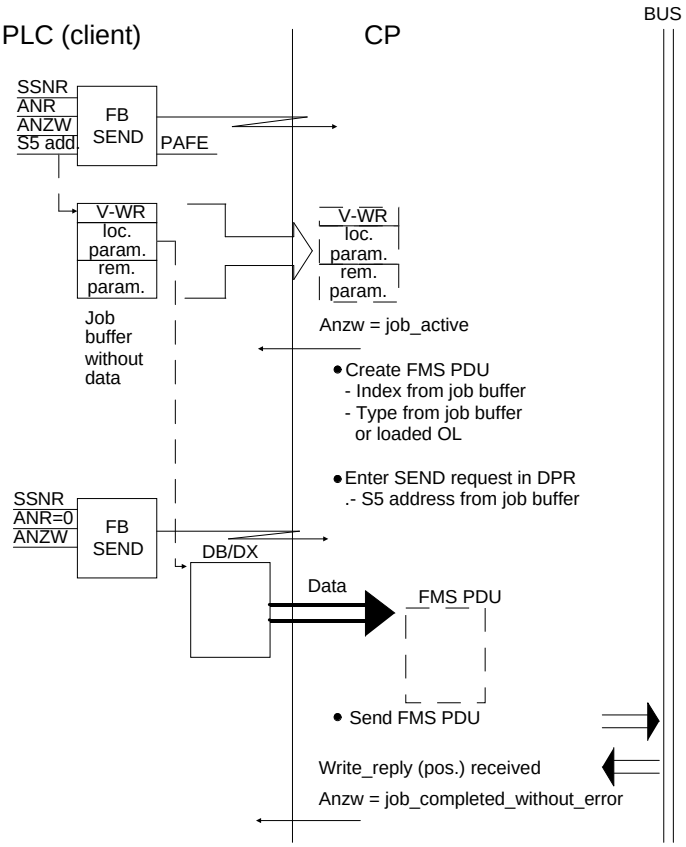


Fig. 4.5 Sequence Description (Write with DB or DX, Positive Acknowledgment)

Description of the sequence (write with DA, positive acknowledgment)

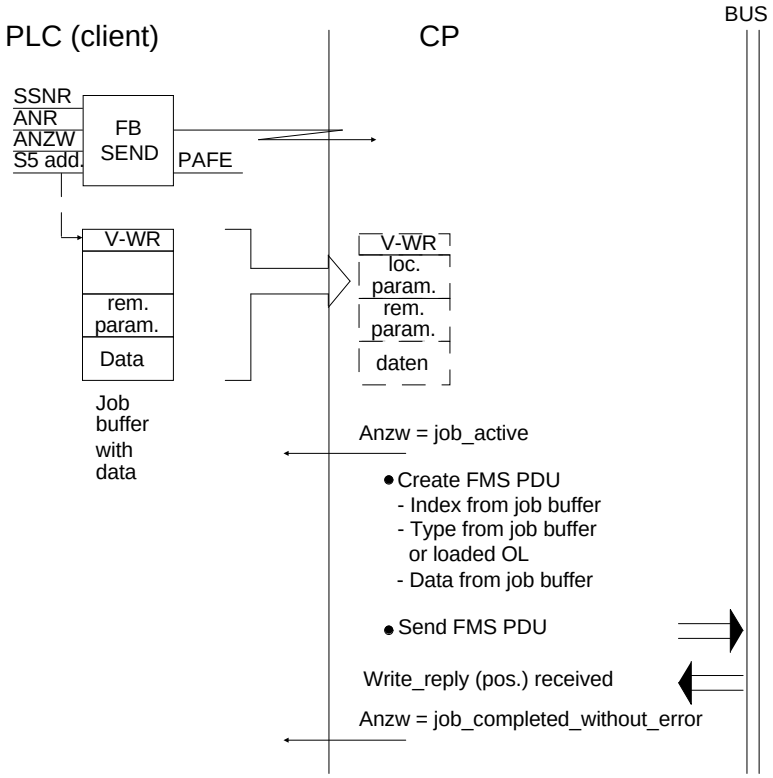


Fig. 4.6 Sequence Description (Write with DA, Positive Acknowledgment)

Description of the sequence (write, negative acknowledgment)

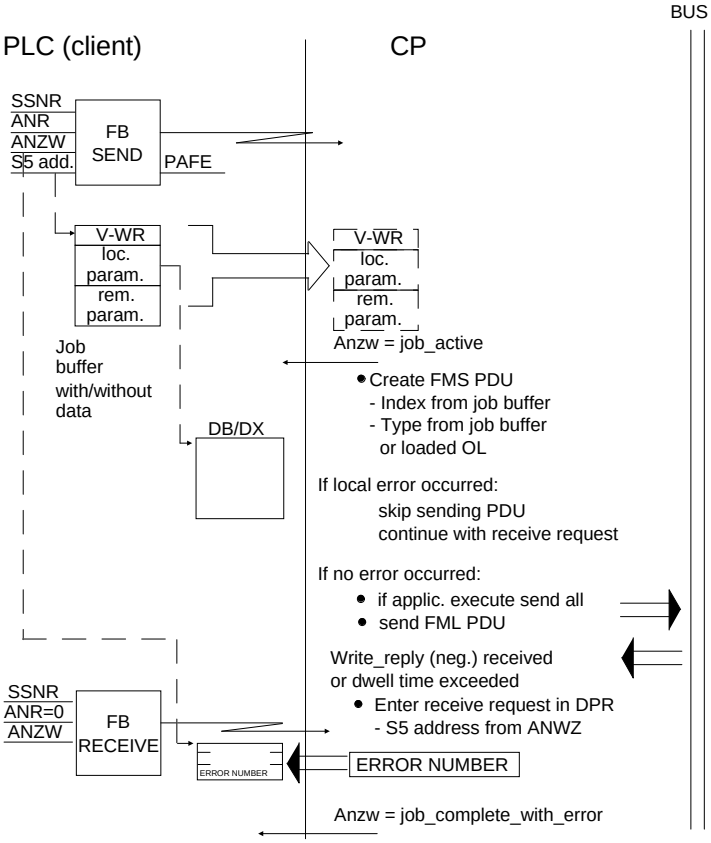


Fig. 4.7 Sequence Description (Write, Negative Acknowledgment)

4.1.3 Information Report, Client Interface

Job buffer "information report"

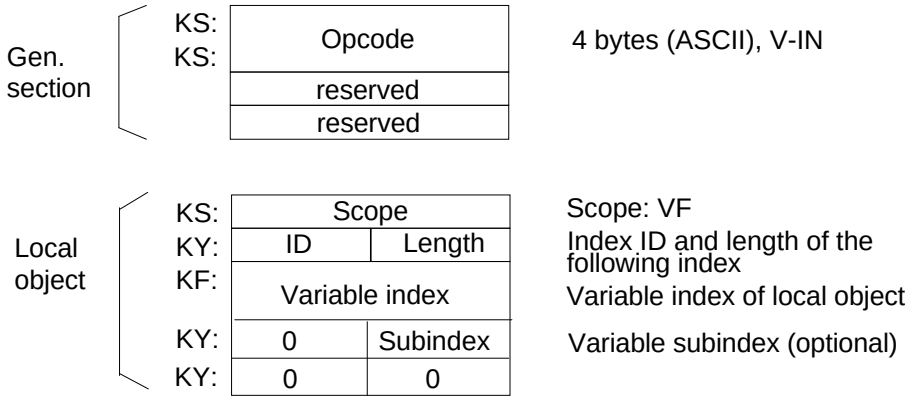


Fig. 4.8 Structure of the Job Buffer for the "Information Report" SINEC Variable Service

Description of the call

General section:

Opcode: V-IN

Local object: With the SINEC variable service "information report", the server transmits a local object (or its value) to another station. The local object is identified by the index and with alternative access by the subindex. The object itself must always be configured locally with the VFD variables editor.

Scope: 1 word, format KS, scope of the variable: VF.

Index ID and index length: 1 word, format: KY. The index identifier and length of the index description depend on the type of object access. If alternative access is used both the index and subindex must be specified in the job buffer.

	Index ID	Length of the following index description in bytes
Only index	2	2
Index and subindex	3	4

Var. index: 1 word, format: KH, index of the local object

Subindex: 1 word, format KY
range of values: high byte: 0, low byte: Subindex: 0 - 255. Specifying the subindex is optional. Subindex 0 identifies the object itself. Subindex > 0 identifies a component of the object at nesting level 1.

Description of the sequence (information report, positive acknowledgment)

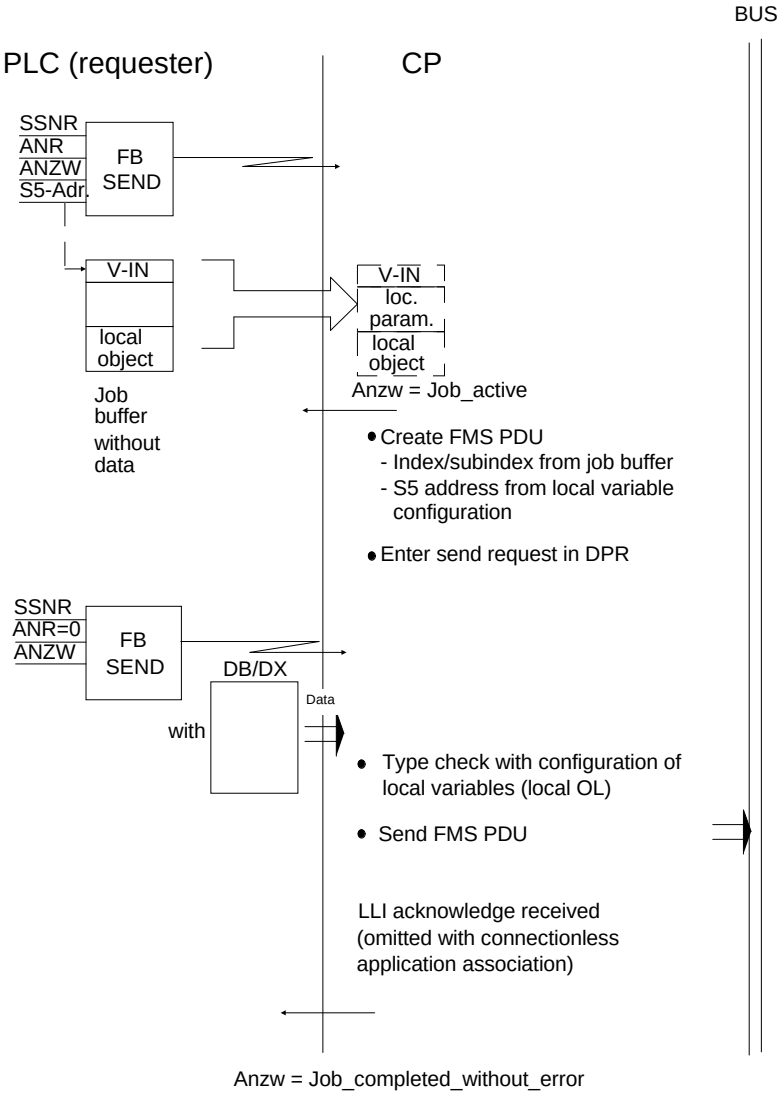


Fig. 4.9 Sequence Description (Information Report, Positive Acknowledgment)

Description of the sequence (information report, negative acknowledgment, local error)

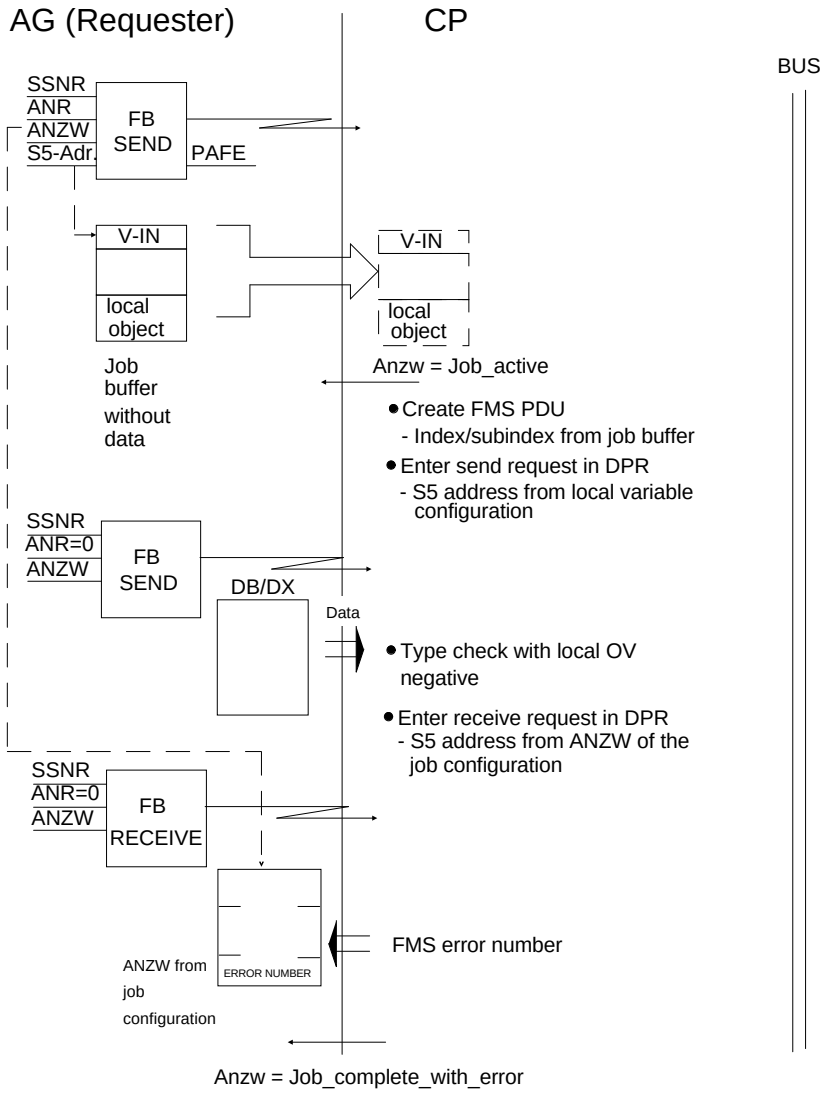


Fig. 4.10 Sequence Description (Information Report, Negative Ack., Local Error)

4.1.4 Variable Services, Server Interface

4.1.4.1 Configuring Variables

To be able to execute FMS variables services in the CP, you must define these variables in the COM during the configuring phase. You can only select the scope VFD-specific for the variables.

VFD-specific
VFD (virtual field device) indicates the sum of the objects and operations on objects managed by a CP. With FMS, access to objects by unauthorized communications partners can be disabled. The CP provides access protection using a password.

A communication object protected by a password can only be accessed on one single application association on which the client has the identical password configured. The password of the variables is assigned during the configuring of the variables, the password for an application association is assigned when the application association of the **remote** station is configured. The access rights must then be set to read with password or write with password or read and write with password.

Since up to four CPUs can exist in the SIMATIC PLC in a station (e.g. S5-155U, S5-135U) in the multiprocessor mode, when configuring, not only the S5 address of the VFD-specific variables must be defined but also the interface on which the CP will access the variable (i.e. the physical CPU in which the variable is located) (see Section 2.2.3).

Apart from defining the index and the type of the variable and the S5 address at which the variable is stored by the PLC program, you can also specify an address for a status word. The PLC program can then read information about access to the variable. It is, for example, possible for the PLC program to recognize whether the value of a variable has been updated by another station or whether no access has taken place. Using the status word, the PLC program can also block access to the variable temporarily for another station. Handling the status word and the significance of the individual bits are explained in the descriptions of the CP handling blocks.

During the configuring phase, you can also globally prevent all other stations writing to a variable ("read only variable").

4.1.4.2 Processing Variable Services in the CP

The SINEC services read and write are interpreted and executed at the server end in the CP largely without support of the PLC CPU. Only the CP handling blocks "SEND ALL" and "RECEIVE ALL" must be called in the PLC program.

The CP processes a received variable job as follows:

➢ It searches through the configured list for the index specified in the PDU.

➢ It triggers the job (if permitted by the statically configured access rights)

- Write
  CP triggers a "RECEIVE ALL" job with the configured S5 address.
- Read
  CP triggers a "SEND ALL" with the configured S5 address.

➢ On completion of the data exchange with the PLC, the reply PDU is generated and transmitted.

4.1.4.3 Configuring the Report Variables

The report variables must be configured on the receiver for the SINEC service "information report". The report variables allow a received communication object (remote variable) to be modeled on a process object in the programmable controller.

The communication object referenced with "information report" is defined in the local object list of the server (requester with "information report"). The Client (receiver with "information report") does not initially have a description of the communication object. In the report variables, it is therefore possible to assign the type description, destination S5 address and method of receiving the data to the variables received.
A type description of the remote index (variable type, length) must always exist since only data whose type has been checked are passed on to the PLC. If a type description is not or cannot be configured (for example with structured variables), the remote index is entered in the list "access to variables (GET OL)". The CP 5431 FMS then fetches the type description itself from the object list of the remote VFD.
The transfer of the reported data to the S5 destination address configured in the report variables can be speeded up by using PLC interrupts. If the reported data are transferred synchronized with the cycle, a RECEIVE-ALL handling block call in the PLC cycle is sufficient.
If a remote index is reported, which has an interrupt channel assigned to it (interrupt A, ..., interrupt D), the CP triggers the CPU interrupt.
The S5 program in the programmable controller then branches from the processing cycle to an interrupt routine. In the interrupt routine, a "RECEIVE DIRECT" handling block must be started to transfer the data from the CP to the PLC. The "RECEIVE DIRECT" must be supplied with the interface number from the CP basic configuration data and with the job number and status word from the configured report variables. The S5 destination address configured in the report variables and the data length in the PLC are passed on to the receive direct handling block by the CP.

In the cyclic part of the PLC program, a "RECEIVE-ALL" handling block must be called. With the "RECEIVE ALL", the CP transfers data to the PLC when the interrupt channel is not used and, if an error occurs, transfers an FMS error word to the status word configured in the report variables.

Description of the sequence (server interface, interrupt channel)

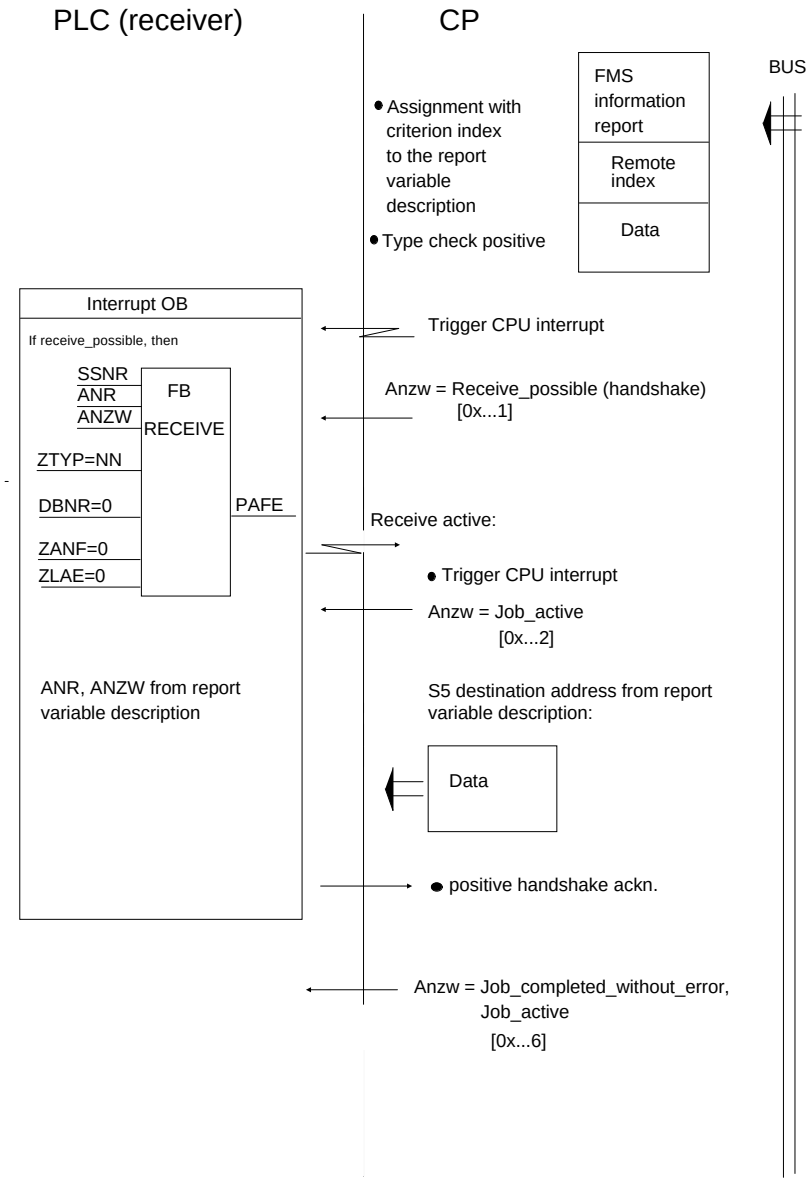


Fig. 4.11 Sequence Description (Interrupt Channel, Positive Acknowledgment)

Description of the sequence (server interface, interrupt channel)

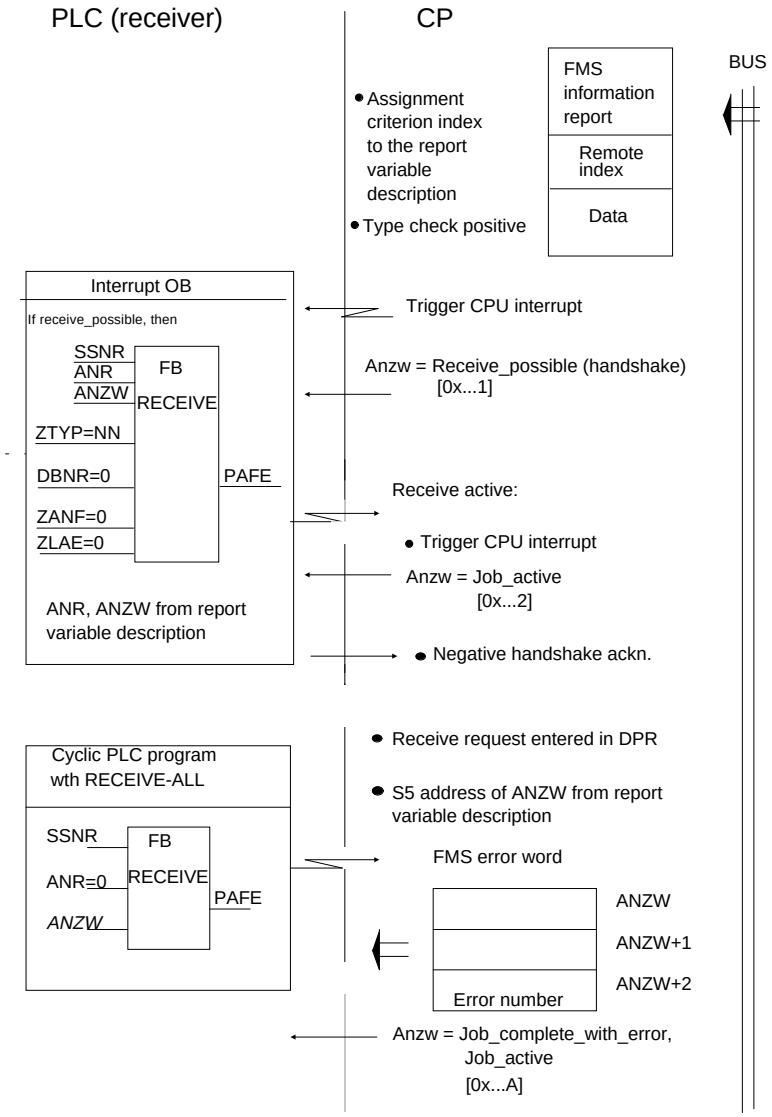


Fig. 4.12 Sequence Description (Interrupt Channel, Negative Acknowledgment)

4.1.4.4 Type Conversion and Type Check

The following type conversions are performed:

	FMS representation	S5 representation
Boolean	8 bits	16 bits
Floating point	IEEE	MC 5 (KG)
Bit string	ASN1 "bit string"	SIMATIC "bit string" KY

Table 4.4 Type Conversion on Server

With all values, remember that each standard type of variable or element is saved at the next word boundary.

☞ **You can use the status word specified during configuration of the variables to disable or enable access to local variables.**

Strings of variable length are not supported, requests for such strings are acknowledged negatively by the CP.

On the server side, there is no type check when reading and writing. However, as the client, the CP provides the possibility of a type check when the type description has been loaded (specification of the corresponding variable index in the configuring screen) and if it was additionally specified in the job buffer. Otherwise, the only check made is whether the variable lengths match and whether the range of values of the variables have been exceeded. If a device from a different vendor is used as the client, you may have to make sure that the types are consistent. The access rights are also checked. If there is an access conflict, a negative acknowledgment is generated.

4.1.5 General Services for Virtual Field Devices

The general FMS services for virtual field devices allow a client to request information about the status or attributes of the VFD in the server. The information can then be further processed on the client, for example to provide a supervisory control center with an overview of the whole status of the plant.

4.1.5.1 Status of the Virtual Device

Using the "status" service, a client requests information about the physical and logical status of the virtual field device. The server sends the requested information in the acknowledgment (e.g. whether the "real" field device or the communications processor of the server is in the RUN or STOP mode or whether the PLC and CP are synchronized or not).

Job buffer "VFD Status"

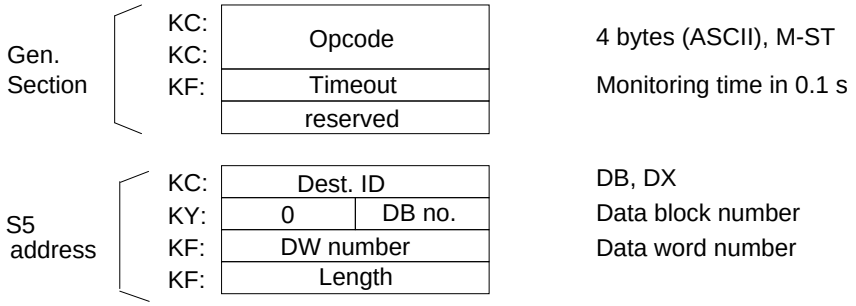


Fig. 4.13 Structure of Reply Data in "VFD Status"

Description of the call

General section:

Opcode: M-ST

Timeout: 1 word, format: KF
Specifies the maximum length of time the user

program will wait for an acknowledgment for the service (i.e. the maximum dwell time of the job in the CP). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 adress: Address at which the information about the status will be stored.

Identifier: 1 word, format: KS
range of values: DB, DX
DB for data block
DX for extended data block
Local object:
Description of the local destination address for the service, DA identifier not permitted.

DB no.: 1 word, format: KY
range of values: high byte: 0, low byte: 1-255
low byte: DB or DW number

With identifier DB or DX the DB no. must never be 0. In PLCs that use DB 1 for system settings, you must make sure that DB 1 is not overwritten by an FMS service.

DW no.: 1 word, format: KF
range of values: 0 - 2042

This is an offset within the data block or extended data block. Owing to the current PG software, the maximum DW number is restricted to 2042. The "length" parameter required for the complete definition of an S5 address is not specified. This is calculated on the CP from the type information for the variable.

Length: format: KF
range of values: 4..2043, -1

Meaning: Length of the data block area in which the VFD status can be stored; the value -1 means that all the data in the acknowledgment from the DW number to the end of the data block can be accepted.

Description of the sequence "VFD status"

The sequence of the "VFD status" service is analogous to the sequence of the "read" SINEC variable service.

Processing on the server

The service is executed by the CP without the support of the PLC. However, the status of the PLC is also included in the data of the reply as follows:

Structure of the reply data sent following a VFD status request:

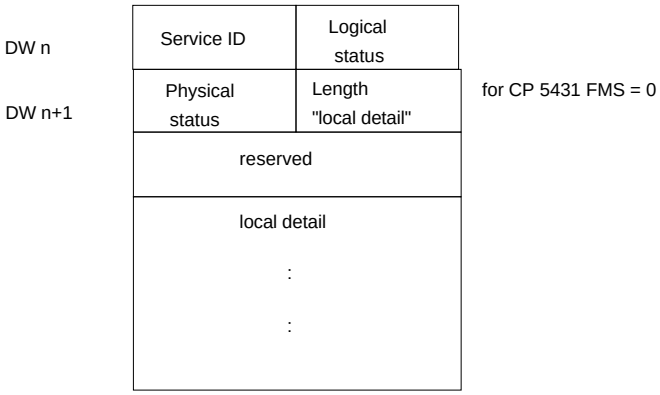


Fig. 4.14 Structure of the Reply Data

The reply data stored by the CP at the STEP 5 address specified in the job buffer has the following structure:

Service identifier: to assign the reply to the required service uniquely (0H).

Logical status:

Value	Logical status	CP 5431 FMS
0	operational	CP RUN and PLC RUN
2	restricted services	CP STOP or PLC STOP

Physical status:

Value	Physical status	CP 5431 FMS
10_H	operational	PLC RUN
11_H	partially operational	-
12_H	not operational	-
13_H	maintenance required	PLC STOP

4.1.5.2 Identify

With this service, a client can request information about the attributes of the VFD from a server.

Job buffer "Identify

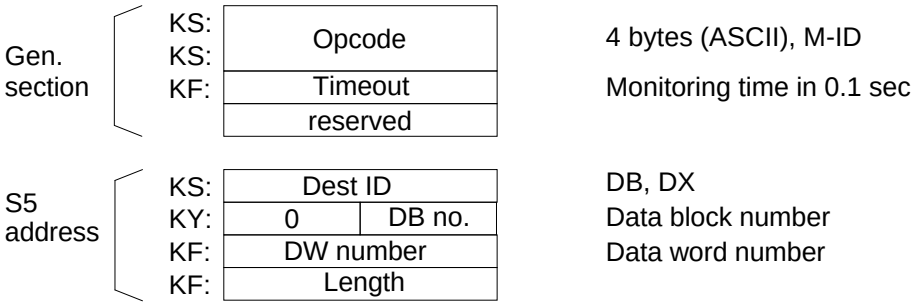


Fig. 4.15 Structure of the Job Buffer "Identify"

Description of the call

General section:

Opcode: M-ID

Timeout: 1 word, format: KF
Specifies the maximum length of time the user program will wait for an acknowledgment for the service (i.e. the maximum dwell time of the job in the CP). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 address: Address at which the information about the status will be stored.

Dest. identifier	format:	KS
	range of values:	DB, DX
DB no.:	format:	KY
	range of values:	high byte: 0 low byte: 1..255
DW number:	format:	KF
	range of values:	0..2042
Length:	format:	KF
	range of values:	33..2043, -1

Meaning: Length of the data block area in which the VFD status can be stored; the value -1 means that all the data in the acknowledgment from the DW number to the end of the data block can be accepted.

Description of the sequence "Identify"

The sequence of the "identify" service is analogous to the sequence of the "read" SINEC variable service.

The service cannot be used on the local PLC.

Processing on the server

The service is executed by the CP without support of the PLC.

Structure of the reply data

The reply data stored by the CP at the S5 address specified in the job

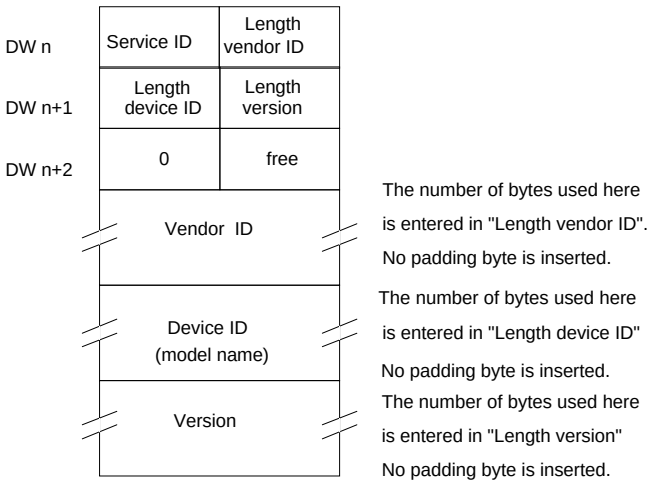


Fig. 4.16 Structure of the Reply Data in the Data Block

buffer has the following structure:

Service identifier:	2h (to assign the reply to the requested service uniquely).
Length of vendor's identifier:	Length of the vendor identifier contained in the reply data.
Length of the device identifier:	length of the device identifier contained in the reply data.
Length of the version:	length of the version contained in the reply data.

For the CP 5431 FMS e.g.:

Vendor Name: "Siemens AG"

Device ID: "CP 5431 FMS"

Version: "V X.Y"

4.2 Configuration

To assign parameters to CP, the PG package COM 5431 FMS is used under SINEC NCM.

The screens required for configuring are provided by SINEC NCM as shown in Fig. 4.17:

- Application Association Configuration
- VFD Variables Editor
- Request Editor
- Test Functions.

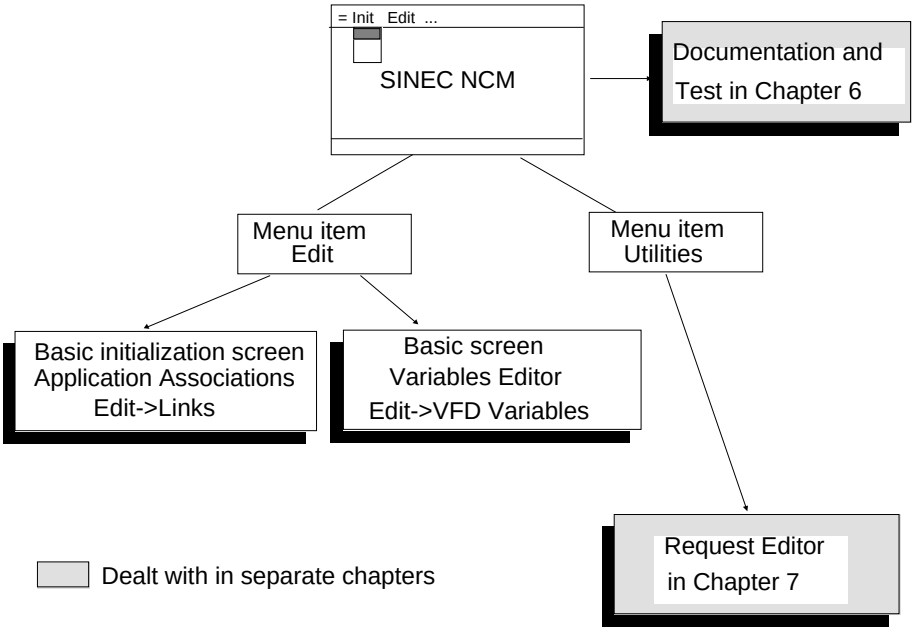


Fig. 4.17 Overview: Configuring Acyclic Communication

4.2.1 Logical Connections (Application Associations)

Up to four jobs are possible on an application association (for example, between two PLCs). An application association is identified by a communication reference (CR), The jobs (for all application associations on a device) are identified uniquely with numbers.Each application association functions on only one interface, i.e. only one CPU is addressed.

The Context of an application association:
The context contains the attributes of the application association which must be compatible with the corresponding attributes of the communications partner. These include the following:

- Application association type
- Max. SCC (max. value for Send Confirmed Request Counter)
- Max. RCC (max. value for Receive Confirmed Request Counter)
- Max. SAC (max. value for Send Acknowleged Request Counter)
- Max. RAC (max. value for Receive Acknowleged Request Counter)
- Monitoring interval (Acyclic Control Intervall, ACI)

The context of connection-oriented application associations is checked when the association is established. The context is irrelevant for connectionless application associations.

The application association type and monitoring interval must be identical on both partners in the application association. The values of the send request counter at one end must correspond to the values at the other.

The following conventions apply to the CP 5431 FMS:

Client jobs (SCC, SAC) and server jobs (RCC) are specified by assigning the job numbers. The odd job numbers are used for active jobs, i.e. client jobs while the even job numbers are used for passive jobs, i.e. server jobs.
A distinction must also be made between confirmed and unconfirmed jobs.

Relationship between job numbers and the request counters:

	SCC	SAC	RCC	RAC
Without unconfirmed job	o	0	e	0...4
With unconfirmed job	o-1	1*	e	0...4

e = number of even ANRs
o = number of odd ANRs
* = The first configured odd job number of the application association is reserved for a send unconfirmed request (SINEC service "information report").

The unconfirmed server jobs (RAC) can be set independently of the job numbers of the application association. If report variables are configured, the maximum RAC can be set in the range 1 to 4, otherwise the setting is max. RAC = 0.

This division allows client and server jobs to be processed on an application association simultaneously. With application associations, the job numbers specified here must be unique. This means that an ANR assigned to a particular application association cannot be used with any other application association.

Application associations and the indexes of the variable descriptions to be loaded are stored in data link blocks (VBs). These data link blocks are either saved in a database file (offline mode) or written directly to the database of the CP and modified there (online mode). In this way, database files created offline can be loaded on the CP or the content of the CP database can be saved in a file.

4.2.2 Configuring Application Associations

Select Edit -> Links to call the following screen. The screen has the following structure:

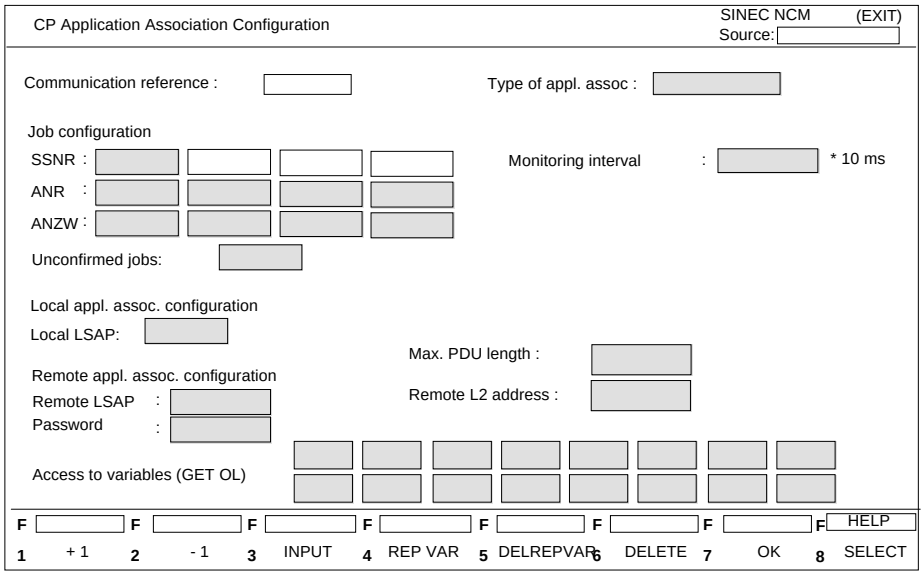


Fig. 4.18 Configuring Application Associations

Input fields:

Type of appl. assoc.**:** Specifies the ALI application association type. The following are permitted:

MMAC: acyclic master-master-application association
The confirmed services read and write are possible. The master be client and server. The unconfirmed service information report is possible. The master can be requester and receiver (configure report variables!). Each application association must be assigned a unique SAP.

MSAC: acyclic master-slave-application associations
The confirmed services read and write are possible. The master is always client. The master always uses the poll SAP, the slave needs its own SAP for every application association.

MSAC_SI: acyclic master-slave-application association with slave initiative
Like MSAC, but the master is also receiver for the unconfirmed service information report (configure report variables!).

Broadcast: connectionless broadcast application association
The master is requester for the unconfirmed service information report. For the receiver functions, report variables must be configured. Confirmed services are not allowed. A maximum of 16 bytes of data can be transmitted. The remote LSAP is always the broadcast SAP, the remote address is the broadcast address. Only one broadcast application association can be configured. Fields on the screen that are only intended for connection-oriented application associations or only for confirmed jobs are not displayed.

Monitoring interval:

Acyclic application associations can be monitored from the CP end. The value entered specifies the monitoring time. It you specify the value 0, there is no monitoring, (range of values: 0, 52 to 99999x10ms).

☞ **For master-master application associations, the monitoring time entered should be at least three times greater than the target rotation time (TTR), and for master-slave application associations at least three times greater than the actual polling time otherwise aborts of the type "FDL: no remote resources" or "LLI: timeout" can occur.**

☞ **Changing the data rate influences the TTR.**

SSNR: The interface number is the page number of the CP. This forms the CPU - CP interface. The interface number must be uniform for all jobs on an application association. It can therefore only be entered in the first field and is automatically repeated when further parallel services are defined (range of values: 0..3).

ANR: For each application association, up to four jobs with different values for job number (ANR) and status word (ANZW) can be defined (range of values: 1 ... 199).

Odd ANR: for each job, a SEND handling block with the corresponding combination of ANR/SSNR must be run.

☞ **Odd job number: client jobs**
Even job number: server jobs

ANZW: Status word for the defined job:

Format: <DB number> <word number>
This status word (DX, DB, FW) along with the SSNR and the ANR forms the interface to the PLC program (parameter for send direct and control HDB). (range of values: DB number 1...255, word number 0...250).

Unconfirmed jobs:

Configuration for the SINEC service "information report". The first odd ANR of this application association is reserved for an unconfirmed client job (job buffer for "information report").
Range of values and influence on parallel use and data length of the application association:

<blank>: no unconfirmed service possible;
max. SAC = 0;
max. SCC = number of odd ANRs.

UNCFLOW: One unconfirmed service with "LOW" priority;
max. SAC = 1;
max. SCC = number of odd ANRs -1.

UNCHIGH: One unconfirmed service with "HIGH" priority;
max. SAC = 1
max. SCC = number of odd ANRs -1;
max. PDU length for high priority frames to be sent = 24 bytes (otherwise: 0 bytes).

Local LSAP/ remote LSAP:

With the CP 5431 FMS, a maximum of 32 SAPs can be activated simultaneously for acyclic application associations. With MSAC application associations, the poll SAP (value 58) is entered automatically. If the application association is an MMAC application association, the local LSAP must not be the same as the poll SAP (range of values local: 2 .. 53 remote: 0, 2 .. 62).

☞ **Make sure that no overlapping of job numbers and LSAPs for S5-S5 and FDL communication is possible.**

Remote L2 address:

Here, the layer 2 address of the partner station is specified. This parameter must always be equal to or less than the parameter "highest L2 station address" (HSA) in the local network parameters (range of values: 1 to 126).

Password: The password is a number (index). If you do not make an entry in this field or enter a 0, this application association will not support access using a password. It is not permitted for two application associations to have the same remote password. If an attempt is made to establish such an application association, it is rejected (range of values: 0 to 255).

Max. PDU length: Limits the length of the transmitted FMS PDU for frames with low priority.
(range of values: 24, ... , 241 bytes)

The length of a transmitted FMS-PDU with high priority is as follows:
- no unconfirmed high service: 0 bytes
- with unconfirmed service: 24 bytes
  (data length max. 16 bytes).

The received FMS-PDU length is fixed at 241 bytes (low priority).

Access to variables: If a type check is required with the jobs (comparison of type specified in job buffer and configured type on the remote station) or if a type cannot be specified in the job buffer (e.g. access to a structure), the indexes of the corresponding variables must be specified here. A maximum of 16 indexes of remote variables can be specified per application association. The total number of programmable remote variable indexes is dynamic and depends on the type of the variables being used. With variables without standard types (e.g. records or structures), a type check is only performed when there are still adequate resources (range of values of the variable index: 15 .. 65535).

Output field:

Communication reference: The communication reference is required to identify the application associations uniquely. This value is generated automatically by SINEC NCM and it is displayed for information.

Function keys:

F1 +1	Page one communication reference forwards.
F2 -1	Page one communication reference backwards.
F3 INPUT	With this function key the next free communication reference is made available for configuring.
F4 REP VAR	Enter the assignment list for report variables.
F5 DELREPVAR	Delete report variables
F6 DELETE	With this function key, the current communication reference is deleted. You must confirm your intention before it is deleted.
F7 OK	The "OK" function key enters the data. If the database does not exist, it is created after you confirm that you want to create it.
F8 SELECT	If you press this key, a list of possible entries is displayed for every input field that cannot be freely edited. You select the values in the list with the cursor and enter them in the input field directly with the return key.

4.2.2.1 Assignment List for Report Variables

Report variables must be configured when a remote server is going to report local objects on this application association (connection-oriented and connectionless).

The reported data are assigned using the assignment criterion "remote variable index":

- S5 destination address for storing data in the local PLC
- Type description (optional: "access to variable" application association configuration screen)
- Priority and transfer procedure on the S5 interface
- Status word (optional).

The report variables are valid within the application association to which they are assigned.
If you do not specify a type description for the expected remote variable, the index is entered in the "Access to variables" list in the application association configuration screen. After the application association has been established, the CP reads the type description from the remote object list (Get OL service).

If the type is completely configured and the remote index is also entered in the "Access to variables" list, the CP checks that the local configuration data in the information report screen matches the type description in the object list of the remote slave.

☞ **The Get OL service cannot be used with broadcast application associations; with broadcast, only objects of standard types or arrays of standard types can be used.**

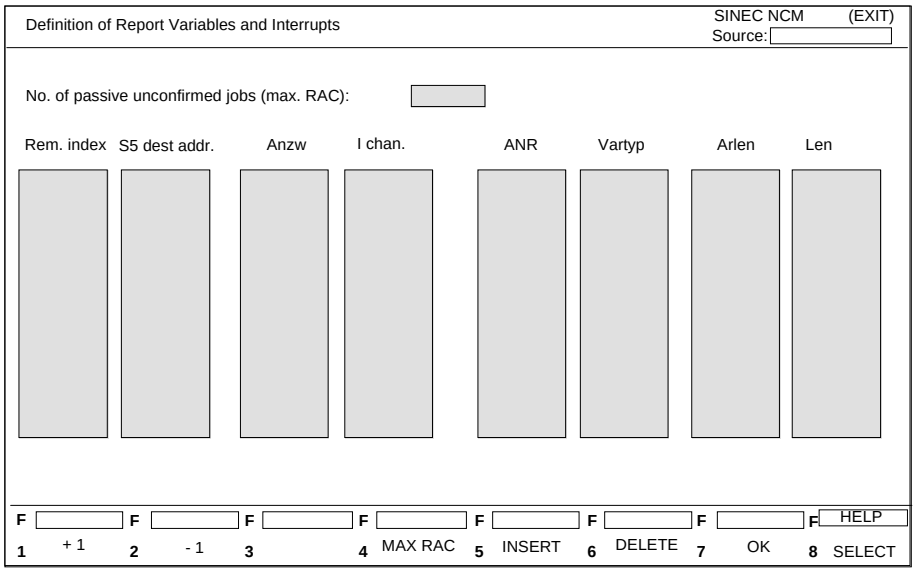


Fig. 4.19 Report Variables Screen

Input fields:

No. of passive unconfirmed jobs (max. RAC): Setting for parallel passive unconfirmed jobs (context of the application association!)
range of values: 1 - 4 passive jobs
The setting for maximum RAC is only effective when report variables have been configured, otherwise the max. RAC counter is set to 0.

Rem. index: Index of a variable from the object list of a remote station.
Only the indexes specified here can be reported by the remote station.
(range of values: 15 to 65535)

S5 dest. address: S5 destination address in the local PLC where the received data will be stored.
Format: <ORG_identifier><ERW_identifier><start address>
(range of values:
ORG_identifier: DB or DX for data block or extended data block.
ERW_identifier: 1 .. 255 data block number
Start address.: 0..2042 data word number at which the value of the variable starts)

ANZW: Status word for job status and FMS errors (optional).

Format: <TYPE><DBNO><DWNO> or <FWNO>
(range of values: with TYPE = FW, DB, DX
TYPE: FW,DB,DX
DBNO: 1 .. 255 if TYPE = DB or DX (data block number)
FWNO = 0..250, if TYPE = FW (flag word number)
DWNO = 0..255, if TYPE = DB, DX (data word number) DWNO = empty, if TYPE = FW).

I chan.: Accelerated transfer of the received data from the CP to the CPU by non-cycle synchronized triggering of a CPU interrupt.
" ": no CPU interrupt
"A": trigger CPU interrupt A
"B": trigger CPU interrupt B
"C": trigger CPU interrupt C
"D": trigger CPU interrupt D.

ANR: Job number for "RECEIVE" handling block in the interrupt routine (only when interrupt occurs).
range of values: 2, ... , 198 (even numbers).

Vartyp: Type description of the remote object (optional).
Standard types and arrays of standard types can be configured.

- BO: Boolean
- IN08: Integer (8 bits)
- IN16: Integer (16 bits)
- IN32: Integer (32 bits)
- UN08: Unsigned integer (8 bits)
- UN16: Unsigned integer (16 bits)
- UN32: Unsigned integer (32 bits)
- FP: Floating point (32 bits)
- VS: Visible string (length in bytes)
- OS: Octet string (length in bytes)
- BS: Bit string (length in bits)

Configurable standard types

With the standard types Boolean, integer, unsigned integer and floating point, the length of the variable is coded in the type. With the string types (VS, OS, BS), the length must be coded in the "Len" input field.
To configure an array of standard types, you must specify the number of elements "Arlen" field.
With structures and arrays of structures, the type description is not configured. If the type description is missing, the index is entered in the "access to variables" list of the application association configuration screen. The type description of the object referenced with the index is then fetched from the server with the Get OL service and saved internally on the CP.

Arlen: Number of elements for an array of standard types.
range of values: 1 ... 255.

Len: Length for string types,
in bytes (OS,VS) or in bits (BS).
range of values: 1 ... 233 (OS, VS)
1 ... 1864 (BS).

Function keys:

F1 LINE +1	Move one line forward.
SHIFT F1 PAGE +1	Move one page forward.
F2 LINE -1	Move one line back.
SHIFT F2 PAGE - 1	Move one page back.
F4 MAX RAC	Move to Max. RAC input field.
F5 INSERT	Insert an empty line at the cursor position.
F6 DELETE	Delete the input line selected with the cursor.
F7 OK	The "OK" function key enters the data. If the database does not exist, it is created after you confirm that you want to create it.
F8 SELECT	If you press this key, a list of possible entries is displayed for every input field that cannot be freely edited. You select the values in the list with the cursor and enter them in the input field directly with the return key.

Example of configuration

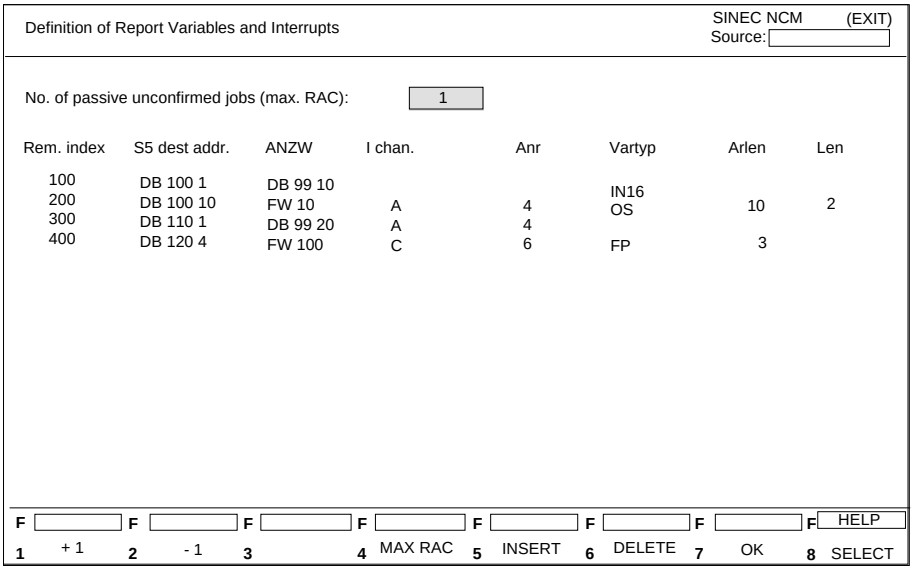


Fig. 4.20 Definition of Report Variables and Interrupts Screen

If the CP receives a SINEC "information report" request from a communications partner on this application association, and if the type check is positive, the received data are transferred to the S5 destination address in the PLC and the status is updated in the ANZW.

If data with index 100 from the example are reported, these must be of the integer type (16 bits). They are then transferred to the data destination with a RECEIVE-ALL handling block call in the cyclic S5 program.
Data with the index 200 belong to an array object with 10 single components of the octet string data type, each individual component is 2 bytes long. If the type check is positive, the CP triggers PLC interrupt "A". A receive handling block assigned the SSNR from the basic configuration data, with ANR=4, ANZW=FW10 and ZTYP=NN is called in the interrupt routine and saves the data at the configured S5 destination address.

If a single component (subindex) of the remote object is reported, the data is stored with the offset of the single component from the configured start address in the S5 PLC.

The type description for the remote index 300 is known to the CP. The index 300 was entered in the "Access to variables" list of the application association configuration screen and the type description was read out of the object list of the remote partner by the CP after the application association was established (not possible on broadcast application associations).

A frame arriving with the remote variable index 400 means that the data are checked for the type array object with 3 single components of the floating point type (32 bits). The interrupt routine in the CPU with interrupt "C" can trigger a RECEIVE-DIRECT handling block call with SSNR, ANR=6, ANZW=FW100, ZTYP=NN.

4.2.3 Configuring Variables

With FMS, the scope of a variable is always the largest possible, i.e. the virtual fieldbus device (VFD). The scope is restricted by specifying a password for the application associations, that have access authorization when the application association is established. All remote passwords for application associations to a VFD must be different, otherwise the application association is not established.

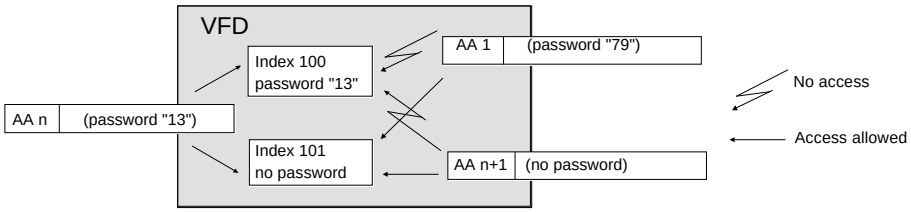


Fig. 4.21 Access Rights with FMS

Possible access to application associations (example)

Variable		Application associations			
		Read with passw.13	Write with passw. 13	Read with / w/o passw. 79	Write with / w/o passw. 79
Read	with passw. 13	+	-	-	-
Write	with passw. 13	-	+	-	-
Read/Write	with passw. 13	+	+	-	-
Read All	w/o passw.	+	-	+	-
Write All	w/o passw.	-	+	-	+
	w/o passw.	+	+	+	+

All the variables are assigned to the total virtual (fieldbus) device (see table above).

With the CP 5431 FMS, it is not possible to put variables into groups (according to PROFIBUS standard "variables lists").

Select Edit -> VFD Variables Editor to call the COM screen. The screen has the following structure:

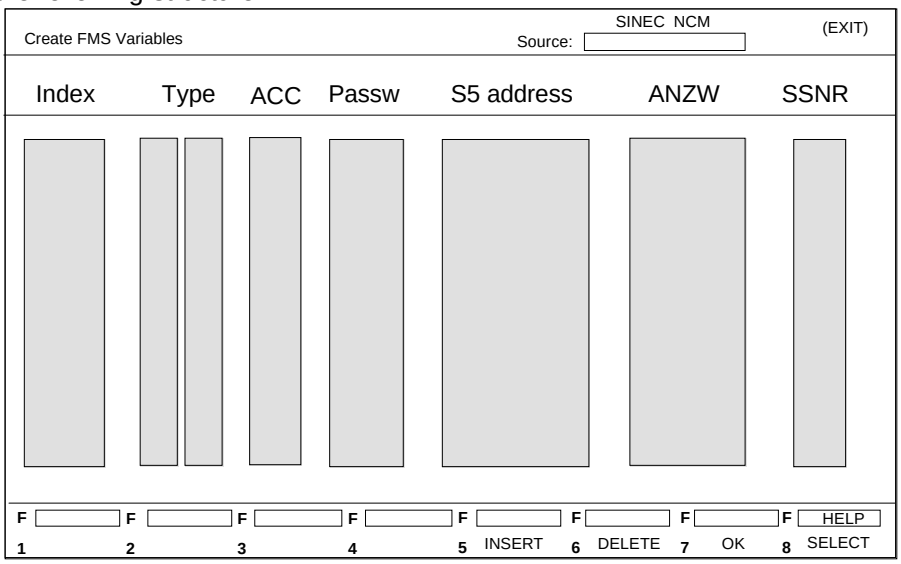


Fig. 4.22 FMS Variables Screen

Input fields:

Index:

➢ The variable index identifies the local variable object uniquely.

- Index 0 is only possible for the OL object description, i.e. it must not be used.
- Indexes 1 to 14 for standard types, i.e. must not be used.
- Indexes 15 to 99 reserved for configured types.

➢ This leaves the remaining area for the indexes (range of values: 100 .. 65535).

Type: Specifies the type of variable: the first input field (2 characters long) identifies the actual variable type and the 2nd input field (4 characters long) the variable length

The following more complex data types are supported, but must be configured for client access in the application association configuration screen by specifying the index:

structures (records) with components of the standard data type

structures with components that contain other components of a standard data type

arrays with elements that are structures with components of a standard data type.

BO : Boolean	-
IN : integer	8,16,32 bits
UN : unsigned integer	8,16,32 bits
FP : floating point	32 bits
OS : octet string	length in bytes
VS : visible string	length in bytes
BS : bit string	length in bits
{ : start of structure	number of components (calculated by COM)
} : end of structure	data type index 15 .. 63 (irrelevant for configuring CP 5431)
AR : array	number of elements in an array

Fig. 4.23 Configurable Variable Types

ACC: This parameter describes the type of access:

no entry:
read and write access possible for all

R:
read only access with password

RA:
read only access for all

W:
only write access with password

WA:
only write access for all

RW:
read and write access with password

Password: If access to the variable is only allowed for clients which have specified a certain password when the application association was established, this must be declared here. No entry or the value 0 means that all clients have access (reading and writing) (range of values: 0 .. 255).

Access protection using groups according to PROFIBUS is not supported, i.e. clients of all groups (0...7) can access the variable.

S5 address: Format: <ORG_identifier><ERW_identifier><start address>
(range of values:

ORG_identifier:	DB or DX for data block or extended data block.
ERW_identifier:	1 .. 255 data block number start address.: 0..2042 data word number at which the value of the variable starts.)

ANZW: Configured status word for server functions with these variables

Format: <TYPE><DBNO><DWNO> or <FWNO>
(range of values: with TYPE = FW, DB, DX
TYPE: FW,DB,DX
DBNO: 1 .. 255 if TYPE = DB or DX (data block number)
FWNO = 0..250, if TYPE = FW (flag word number)
DWNO = 0..255, if TYPE = DB, DX (data word number) DWNO = empty, if TYPE = FW).

SSNR: The interface number is the page number of the CP. This forms the CPU - CP interface. The interface number must be uniform for all jobs on an application association. It can therefore only be entered in the first field and is automatically repeated when further parallel services are defined (range of values: 0..3).

Function keys:

F5 INSERT	Inserts an empty line at the cursor position.
F6 DELETE	Deletes an input line marked by the cursor.
F7 OK	The "OK" function key enters the data. If the database does not yet exist, it is created after you confirm that you want to create it.
F8 SELECT	If you press this key, a list of possible entries is displayed for every input field that cannot be freely edited. You select the values in the list with the cursor and enter them in the input field directly with the return key.

4.3 Status and Error Information

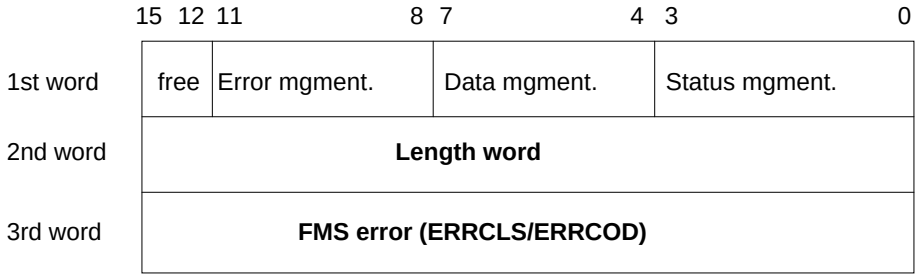
4.3.1 Structure and Meaning of the Information

The CP supplies status and error information as follows:

- in the status word for client jobs in the S5 user program,
- in the status word for variables (server function) in the S5 user program,
- with the test functions "Status of an application association", "Status/ trace of all application associations" and "Overview of all application associations" on the PG.

While the error and status code is partially explained by text in the COM, the S5 user only receives error information in hexadecimal form. Here, only the codes for the S5 user, i.e. the bits in the status word, are explained.

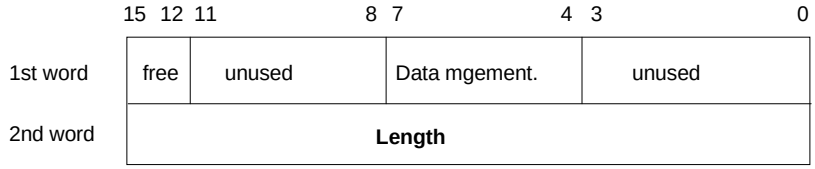
The status word for client jobs contains the status of the job currently being processed with an odd job number (ANR) of an application association. Using the ANR, only one job is processed on an application association at any one time, i.e. a second job with this ANR can only be transferred to the CP with a send direct when the first job is completely processed. The status word is updated by the control and send direct HDBs.
The status words for client jobs have the following structure:



The third word (FMS error) contains the error status of the FMS job currently being processed (refer to Appendix).

In the length word, the handling blocks (SEND, RECEIVE) enter the length of the data already transferred for the job, i.e. with receive jobs, the data already received and with transmit jobs, the data already transmitted.

The variable status words do not have all the parameters of the status words for client jobs. They consist of only two words.



In addition to this, the block status word of the SEND and RECEIVE ALL contains the server job number of the job currently being processed.

The status word itself (1st word) is divided into four nibbles as follows:

Nibble 1, bits 0 to 3, status bits of the job (only client):
This indicates in code form whether a job has been started, whether errors have occurred, or whether the job is disabled, e.g. because the virtual circuit is not established. The COM 5431 FMS status display provides more precise information about the status of a job. Here, intermediate statuses such as "waiting for frame" or "application association establishment active" are indicated.

Nibble 2, bits 4 to 7, data management of the job:
This indicates whether the data transfer for the job is still active, or whether the data transfer or data acceptance is already completed. With the "enable/disable" bit, the data transfer for the job can be blocked (disable = 1; enable = 0). The data management nibble is not represented with COM 5431 FMS.

Nibble 3, bits 8 to 11, error bits of the job:
This contains the error bits of the job. These error bits are only valid when the bit "job completed with error" is set at the same time in the status nibble. The error numbers indicated here also appear in the same form in COM 5431 FMS as transport errors, however in plain text.

Nibble 4, bits 12 to 15
Reserved

Status bits nibble 1, bits 0-3 (only Client)

Bit 0: **Handshake possible**

Bit 1: **Job active**

Set: By the handling blocks when a job is transferred to the CP.

This bit is set in the ANZW of an information report job with data reception using an interrupt until a new information report is received by the CP 5431 FMS.

Cleared: By the handling blocks when a job has been processed by the CP (e.g. acknowledgment received).

Evaluated:
By the handling blocks
A new job is only issued when the "old" job has been processed.
By the user:
To find out whether a new job can be triggered.

Bit 2: **Job completed without error**

Set: By the handling blocks when a job was completed without errors

Cleared: By the handling blocks when the job is triggered again.

Evaluated: By the user: to check whether the job was completed without errors.

Bit 3: **Job completed with error**

Set: By the handling blocks when the job was completed with errors. The cause of the error is then encoded in the high byte of the status word.

Cleared: By the handling blocks when the job is triggered again.

Evaluated:
By the user:
To check whether the job was completed with errors. If the identifier "job completed with error" is set, the cause of the error is written to the high byte of the status word.

Data management Nibble 2, bits 4 - 7

Bit 4: **Data acceptance/data transfer active**

Set: By the handling blocks SEND and RECEIVE when the transfer or acceptance of data for a job has begun, e.g. when data are being transferred via the ALL function although the job was triggered with SEND DIRECT.

Cleared: By the handling blocks SEND and RECEIVE when the data exchange for a job is complete (last sub-block transferred)

Evaluated:
By the user:
During the data transfer CP-PLC, the user must not change the data record of a job. Larger amounts of data can, however, only be transferred in data units and the fragmentation of the total data is distributed over several PLC cycles. To ensure the consistency of the data, the user must first check whether the data unit has just been transferred before changing the data of a job.

Bit 5: **Data transfer completed**

Set: By the SEND handling block, when the data transfer for a job is completed.

Cleared: By the SEND handling block when the transfer of data is begun for a new job (data transfer for new job triggered). By the user, following evaluation (signal edge).

Evaluated:
By the user:
With this bit, the user can determine whether the data record for a job has already been transferred to the CP or when a new data record for a currently active job can be prepared (e.g. cyclic transfer).

Bit 6: **Data acceptance complete**

Set: By the RECEIVE handling block when the acceptance of data for a job has been completed.

Cleared: By the RECEIVE handling block, when the transfer of data to the PLC for a new job has begun. By the user following evaluation (signal edge).

Evaluated:
By the user:
With this bit, the user can determine whether the data record of a job has already been transferred to the PLC or when a new data record for a currently active job was transferred to the PLC.

Bit 7: **Disable block of data.**

Set: By the user, to prevent an area being written to by the RECEIVE block or an area being read from by the SEND block (only with 1st block of data).

Cleared: By the user, to release the data area.

Evaluated: By the SEND and RECEIVE handling blocks. If bit 7 is set, the blocks do not exchange data but signal the error to the CP.

Error bits nibble 3, bits 8 - 11 (only client)

0: **No error**, If bit 3 is nevertheless set in the ANZW, this indicates that the CP has gone through a cold restart

1: **Incorrect type specified in handling block**

Cause: The QTYP/ZTYP parameter of the handling block call had an invalid TYPE identifier assigned to it

Remedy: Supply the block with the correct type identifiers (refer to the manual for HDBs).

2: **Memory area does not exist**

Cause: the source or destination area specified in the HDB call does not exist in the PLC (e.g. DB not set up).

Remedy: use the correct type or set up the DB.

3: **Memory area too small**

Cause: the memory area specified in the HDB call (parameters QTYP/ZTYP, Q/Z-LAE) is too small for the data transmission.

Remedy: set up a data block with sufficient length or correct the parameters for the HDB call.

4: **Timeout**

Cause: during data transfer, a memory cell in the transfer area has not acknowledged.

Remedy: check the memory submodule of the CPU and, if necessary, replace or check the source/destination parameters for the HDB call and correct (with types QB, PY, OY).

5: **Error in status word**

Cause: the parameter "ANZW" was specified incorrectly.

Remedy: correct the parameter or set up the data block in which the ANZW is located correctly (DB no, DB length).

6: **Invalid source/destination parameter**

Cause: an illegal parameter was used.

Remedy: use the correct type

A: **Application association error**

Cause: local station or the communications partner not connected to the L2 bus.

Remedy: switch on the system or check the bus application associations.

B: **Handshake error**

Cause: HDB executed incorrectly
HDB execution time exceeded.

Remedy: check HDB version or if the monitoring time is exceeded, restart the job.

C: **System error**

Cause: error in system program.

Remedy: inform Siemens service.

D: **Disabled block of data**

Cause: the data transmission is/was disabled while the HDB was executed (control bit disable/enable in ANZW set to disable).

Remedy: set the disable/enable control bit in the ANZW to enable (0) and repeat the communications job.

E: **Passive job cannot be processed**

Cause: The even ANR (passive ANR) for an application association cannot be triggered with a job.

Remedy: select an odd job number

F: **Application association or ANR not specified.**

Cause: the job (parameter SSNR/ANR) is not defined on the CP.

Remedy: configure job (application association) or correct the SSNR/ANR parameter in the HDB call.

The length word (2nd word) :

The length word immediately follows the status word (in the next memory location). This indicates the number bytes of data exchanged between the PLC and CP.

Written: By the SEND and RECEIVE blocks during the data exchange. The length word is calculated from the following:
number of data bytes currently being transmitted + number of data bytes already transmitted.

Cleared: Overwritten by each new SEND, RECEIVE

Evaluated:
By the user:

When the "job completed without error" or "data transfer/acceptance completed" bit is set, the current source or destination length is indicated in the length word

When the "job completed with error" bit is set, the length word contains the number of data transferred before the error.

The FMS error word (3rd word) (client only)

Explanations of the FMS error word can be found in the appendix. (see Section 8.1).

Structure of the parameter assignment error byte

The parameter assignment error byte (PAFE) informs you about various parameter assignment errors. When assigning parameters to individual blocks, you specify the address at which the information can be found. The significance of the individual bits is explained in Fig.4.24.

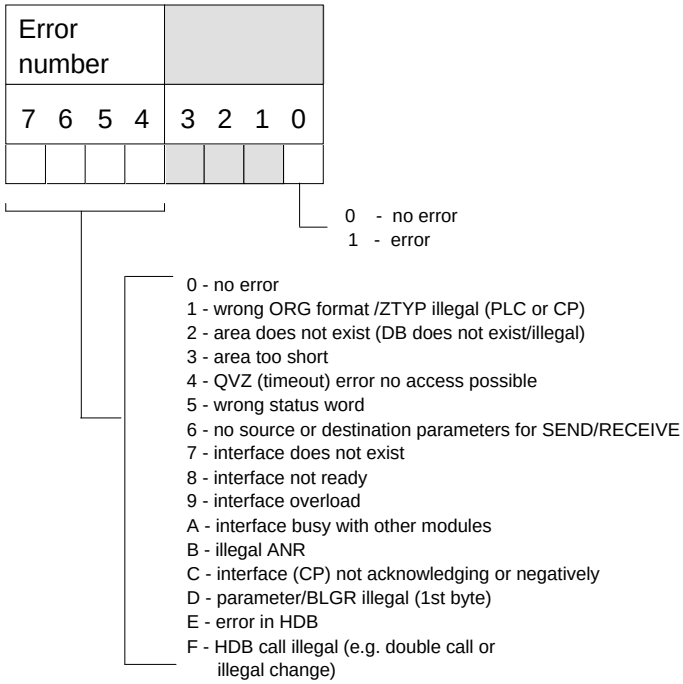


Fig. 4.24 Structure of the Parameter Assignment Error Byte "PAFE"

4.4 Example of a Program: CP 5431 Master-Master Acyclic Data Exchange

In the example, two SIMATIC S5 115U programmable controllers each with a CP 5431 FMS are used.
PLC 1 represents a manufacturing device producing parts for various production stores. PLC 2 represents one of these stores.

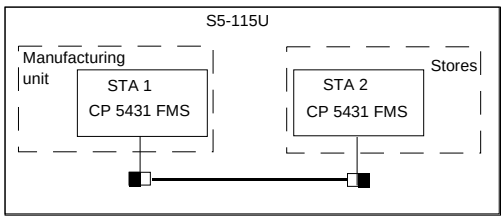


Fig. 4.25 Master-Master Acyclic Data Exchange

Sequence of the data exchange

The manufacturing controller fetches the type and number of parts to be produced from the store controller using a READ job.
Each of the parts produced by the manufacturing controller is transferred individually to the store controller with a WRITE job.

The complete example is designed so that both controllers (manufacturing and store) with the appropriate SINEC L2 CP 5431 FMS communications processors can also be simulated with a 155U controller.

Storing the data in the SIMATIC PLC:

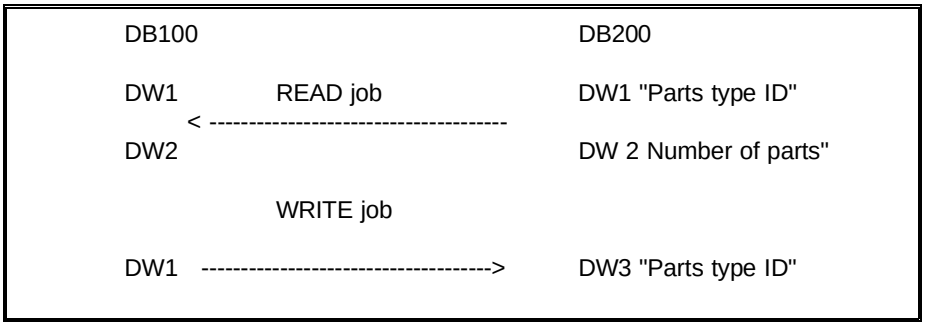


Fig. 4.26 Data Storage Overview

The simulation works as follows:

By setting a bit (FB200...ANST) a READ job is sent from the manufacturing controller to the store controller. This READ job fetches the required parts type and number of parts (DB200 DW1/2) from the store controller and stores it in DB100 DW1/2 of the manufacturing controller.
Once the READ job is completed, i.e. the required data are in DB100 DW1/2, the trigger bit for the READ job is automatically reset by FB200.

By setting the trigger bit for FB100, the configured WRITE job is sent to the store controller. The WRITE job sends a word (DB100 DW1) with the type ID of the required part to the store controller. Here, the transferred type ID is stored in DB200 DW3.

FB201 sends the exact number of WRITE frames as number of parts received from the store. The number was read previously with a read job (DB100 DW2).

These frames are transmitted one after the other to the store controller (WRITE jobs). When all the frames have been successfully transmitted, FB100 automatically resets the trigger bit (refer to activating the program).

By setting the READ trigger bit again followed by the WRITE trigger bit, this data exchange between the two controllers can be repeated as often as necessary.

S5 program description:

Function blocks FB230/FB231:

The blocks contain the calls for the SEND and RECEIVE-ALL handling blocks for both CP 5431 processors (SSNR "0" and SSNR "4").
These are required for the following:

- So that the RECEIVE-ALL in the manufacturing controller can accept the data requested in the READ job.
  So that the SEND-ALL in the stores controller can transfer the data requested in the READ job.

- So that the SEND-ALL in the stores controller can transfer the data requested in the READ job.
  So that the RECEIVE-ALL can receive the data received by the stores controller as a result of the WRITE job.

Function block FB200:

This block manages the READ job of the manufacturing controller. The user (program) must simply set the trigger bit "ANST". When the service is successfully completed, the "ANST" bit is reset by FB200. If errors occur indicated in ANZW or PAFE, FB200 repeats the job automatically.

Function block FB201:

This block manages the WRITE job of the manufacturing controller. The user (program) must simply set the trigger bit "ANST". When the service is successfully completed, the "ANST" bit is reset by FB201. If errors occur indicated in ANZW or PAFE, FB201 repeats the job automatically.

Function block FB1:

With this FB, the manufacturing and transfer of the parts to the store controller is simulated. The FB sends a frame with the corresponding parts ID for each part transferred to the store controller. The number of parts to be transferred is determined previously with the READ job.

To be able to try out this practical example, of using the MMAC-FMS service, follow the procedure outlined below:(see also Volume 1, Chapter 16).

➢ Transfer the following COM 5431 FMS database files to the two CPs.

- Under the network file FMS1@NCM.NET the file QFERTIG.TN1 for station 1.
- Under the network file FMS1@NCM.NET the file QLAGER.TN2 for station 2.

➢ transfer the STEP 5 files FERTIGST.S5D (station 1) and LAGER@ST.S5D (station 2) to the programmable logic controllers you are using (S5-115U).

The example files are on the COM/application examples diskette.

Activating the program:

Testing the application associations to controller 1
With the help of the PG function force VAR, input the elements shown in Table 4.5.

Monitoring the application associations to controller 2
Using the PG function force VAR, input the elements shown in Table 4.6.

In the manual mode, setting flag bit "F1.0" using FORCE VAR triggers the READ job. Once the job is successfully completed, the bit is automatically reset by the program. By setting bit "F1.1" using FORCE VAR, the WRITE job is activated. How often the WRITE job is activated depends on the "number of parts" previously read in from the store. Bit "F1.1" is only reset again by the program when all the parts have been transferred.
In the automatic mode of the simulation program flag bit "F1.0" must simply be set once. After this, the READ and WRITE jobs alternate automatically depending on the process simulation.

Meaning	Screen	
	Operands	Signal states
- Simulation aux. flag	FY1	KM=00000000
- Anzw v-read / active - Anzw v-read length word	FW100 FW102	KH=0008 KH=0000
- Anzw v-write / active - Anzw v-write length word	FW100 FW102	KH=0008/0004 KH=0000
- Applic. assoc. error	FW104	KH=0000
Working DB for production	DB100	
- Stores part type ID - Parts job counter	DW1 DW2	KH=0000 KF=+0

Table 4.5 Application Association Monitoring for Station 1

Meaning	Screen	
	Operands	Signal states
- Variable status word (200)	FW150	KH=0000
- Variable status word (201)	FW160	KH=0000
- Working DB for stores	DB200	
- Stores part type ID - Number of required parts - Type of transferred part	DW1 DW2 DW3	KH=1111 KF=+100 KH=0000

Table 4.6 Application Association Monitoring for Station 2

4.5 Example of a Program for CP the 5431 FMS "Information Report"

For this example, two CP 5431 FMS modules are required in a SIMATIC S5-115U.
As in the example of acyclic data exchange (master-master), station 1 is a manufacturing unit (client) producing parts for various production stores. Station 2 represents one of these stores (server).

Two communication tasks "request parts" and "report fault" are implemented using the "information report" job (server -> client)lö.

Sequence of the data exchange (request parts)

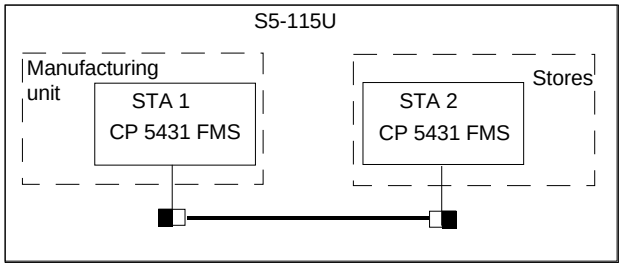


Fig. 4.27 Master-Master Acyclic Data Exchange

If parts are required urgently, the stores controller (station 2) reports this to the manufacturing controller (station 2) with an information report job. The frame that arrives there triggers an interrupt that interrupts the cyclic program. The required data are fetched by the CP 5431 FMS using a RECEIVE DIRECT job in interrupt OB2.

"Request parts"				
Manuf. controller STA1 (client)			Stores controller STA2 (server)	
S5 address (dest.)	Unconfirmed request variable		Local VFD variable	S5 address (source)
DB100, DW5, DW6	Received in int. OB2	< ---------- Info report	Index 300, Type AR2, IN16	DB200, DW5, DW6

Sequence of the data exchange ("report fault")

A problem occurring in the stores controller (STA2) is reported to the manufacturing unit (STA1) using an information report job.
In contrast to the "request parts" job", cyclic program execution is not interrupted. The received data are received within the cyclic user program using the RECEIVE-ALL job.

The stores controller uses the variable subindex to report the problem.

"Report fault"				
Manuf. controller STA1 (client)			Stores controller STA2 (server)	
S5 address (dest)	Unconfirmed request variable		Local VFD-variable	S5 address (source)
DB100, DW7, **DW8**, DW9, DW10	received in cyclic program using receive all	< ---------- Info report	Index 300, Type AR4, Subindex: 2, IN16 (corr. to DW8)	DB200, DW7, **DW8**, DW9, DW10

Data storage on the SIMATIC controller

STA1 Manufacturing controller		STA2 Stores controller
DB100		DB200
DW5		DW5 "Parts type ID"
DW6	< ---------- Information report with interrupt	DW6 "Number of parts"
DW7		DW7 "fault plant section A"
DW8	< ---------- Information report	**DW8 "fault plant section B"**
DW9		DW9 "fault plant section C"
DW10		DW10 "fault plant section D"

The simulation is as follows:

The stores controller (STA2/server) informs the manufacturing controller of the parts type ID and number of urgently required parts using FB201 "REQUEST" (trigger bit F1.1) with the configured information report job (report variable, index 300).

When the information report job is received in the controller, this triggers an interrupt to stop the cyclic program. The received data are fetched using FB202 in interrupt OB2, they are written to DB100 DW5/6 using a RECEIVE DIRECT job.
When the job has been sent without an error occurring, the trigger flag F1.1 is reset by the program.

The example also allows a problem in a section of plant to be reported to the manufacturing unit.
This makes use of FB200 "FAULT" (trigger flag F1.0) again with an information report job (report variable, index 301,2).
The reception of this data on the controller of the manufacturing unit does not trigger an interrupt. The data is saved in DB100 DW8 during the cyclic program using the RECEIVE ALL job.

Since a problem only in plant section B must be reported, this job uses subindex addressing.
Once the job has been sent without an error occurring, the trigger flag F1.0 is reset by the program.

S5 Program Description:

Function blocks FB230/FB231:

The blocks contain the calls for the SEND and RECEIVE ALL handling blocks for both the CP 5431 FMS modules (SSNR "0" and SSNR "4").

Function block FB200 (stores):

This block manages the information report job "FAULT" of the stores controller. The trigger bit "ANST" must be set by the user (program).
Once the service has been completed successfully, the "ANST" bit is reset by FB200. If errors occur (reported in ANZW or PAFE), FB200 automatically repeats the job.

Function block FB201 (stores):

This block manages the information report job "REQUEST" (with interrupt) of the stores controller. The trigger bit "ANST" must be set by the user (program).

Function block FB202 (manufacturing):

This block called in interrupt OB2 fetches the data received from the stores controller.

List of files:

➢ Transfer the following COM 5431 FMS database files to both CPs you are using.

– Under the network file "INFORNCM.NET" the file "QINFOREP.CLI" for station 1 (client -> manufacturing controller).

– Under the network file "INFORNCM.NET" the file "QINFOREP.SER" for station 2 (server -> stores controller).

➢ Transfer the STEP 5 file INFOREST.S5D (stations 1 and 2) to the programmable logic controller (S5-115U).❑

NOTES

5 Cyclic Communication

This chapter contains the following information:

- The devices and applications for which data transmission with the cyclic interface is suitable.
- How this type of data transmission functions.
- How to assign parameters to the CP for this type of data transmission when an S5 programmable controller is to exchange data with a field device.
- How to use this communication based on an example including a STEP 5 program.

5.1 Basics of the Cyclic Interface (CI)

5.1.1 Applications for Data Transmission Using the Cyclic Interface (CI)

Only client interface

Data transmission with the cyclic interface is suitable for communication between SIMATIC S5 PLCs and PROFIBUS-compatible field devices. Field devices are passive stations on the bus which cannot access the bus themselves and must be constantly (normally cyclically) polled by active stations.

"Cyclic data exchange" means that the CP sends the whole of the output area assigned to the cyclic interface cyclically and updates the whole input area assigned to the CI with the received data. You can use these virtual I/Os in the dame way as proper inputs or outputs. These addressed areas are processed normally with STEP 5 language commands. An HDB must be called at the checkpoints required by the user to ensure the consistency of inputs and outputs. This HDB also serves to trigger a group job for data transmission.

The cyclic interface provides data transmission via the I/O area using cyclic application associations. The main feature of data transmission with the CI is that it is easy to use, i.e. it involves far less programming compared with other types of data transmission, for example ALI services. The service does not need to be triggered with a job buffer as in acyclic communication (refer to Chapter 4).

The CP can only be client in this communication. A slave emulation is not supported with the cyclic interface.

The variables to be read and the variables to be written are addressed via their own cyclic application association. The CI always expects the configured variable from the slave. This variable is simulated on the I/O area as described earlier.

Data transmission with CI is suitable for simple variables such as the following:

- small volume of data up to 32 bytes
- cyclically changing data

This would, for example, include measured values.

5.1.2 Functions for Data Transmission with the Cyclic Interface

When you have specified a SIMATIC S5 PLC with a CP 5431 FMS as being an active station, you can configure data transmission via the CI for this PLC and exchange (poll) data with PROFIBUS-compatible field devices. The communication between the SIMATIC S5 PLC and field device functions according to the master slave method (refer to Chapter 2).

In data transmission via the cyclic interface the data exchange takes place via the I/Os of the SIMATIC PLC, as follows:

The data for transmission is assigned to the output area of the I/Os in the control program.

The received data is stored in the input area of the I/Os. Transmitted and received data can be processed with STEP 5 operations

This means:

➢ The data for transmission is transferred to the CP directly using the control program and STEP5 commands or using the operating system functions PIQ (process image of the outputs).

➢ The received data is fetched from the CP directly using the control program and STEP5 commands or using the operating system functions PII (process image of the inputs).

All I/O bytes via which you want to transmit and all I/O bytes via which you want to receive must be designated in the configuration as CI I/Os using the COM (Chapter 5.2.1).

☞ **Communication with the cyclic interface is only permitted using the base interface number (base SSNR).**

☞ **Simultaneous operation of DP and CI is not possible.**

➢ Simulation of the I/O areas on variables

These input and output areas are assigned to different field devices (slaves). The I/Os are simulated on variables. The definition of the variable object for outputs and inputs must be made on the slave (field device) or be fixed there by the vendor. The variables are always addressed by the CP using the object list (OL) index. The simulation of the I/O areas on variables is configured in COM 5431 FMS (Section 5.2.2).

➢ Application association definition with L2 address and SAP

The field device (slave) can only be addressed by the CP when it knows both the L2 address and the corresponding service access point (SAP) of this field device. Both the L2 address of the slave and the SAP number must be specified using the CI editor of the COM 54301 FMS software package. The variable is accessed using the index.

➢ Access to variable using the index.

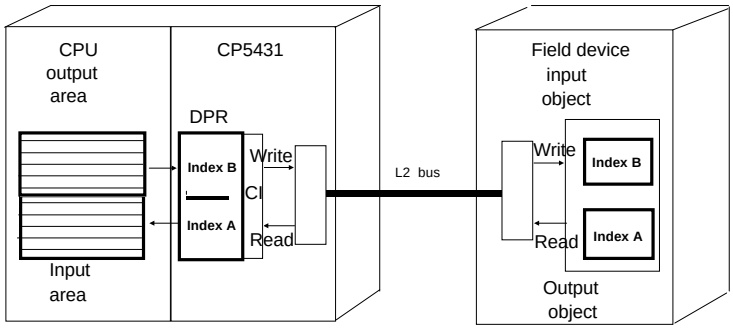


Fig. 5.1 Data Transmission with I/Os, Simulation on FMS Variable Services

Principle of operation:

The CP becomes the "distributor" after it is configured with the CI editor.

Transmitting:

- ➢ data transmission triggered by HDB
- ➢ CI reads the output area of the PLC
- ➢ allocates the area to a field device using the L2 address and destination SAP
- ➢ simulates output bytes on variables and "packs" output bytes in frames
- ➢ sends these frames to the addressed field devices
- ➢ requests response frames from these field devices
- ➢ receives the response frames

Receiving

- ➢ data acceptance triggered by HDB
- ➢ CI allocates the input area to a field device
- ➢ enters the programmed variable index in the request frames
- ➢ sends these frames to the addressed field devices
- ➢ requests response frames from these field devices
- ➢ receives the response frames
- ➢ transfers the received data to the input area addressed using the index

The following information is important:

➢ With field devices, different data are only transmitted via the poll SAP.

➢ The CI transmits and sends only via the poll SAP.

➢ The CI uses only the read and write FMS services.

☞ **The frames of these services always have low priority. This means that when there is a large volume of traffic on the bus with higher priority from other stations, it cannot be guaranteed that CI frames are transmitted during one token rotation. The data transmission cycle is then extended.**

Updating the input and output areas of the CI

The times at which the CP updates the CI bytes to be transmitted are determined by the PLC control program by means of a SEND handling block call with job number 210.
When you assign parameters for the SEND (ANR 210), the parameters QTYP, DBNR, QANF, QLAE are irrelevant.
The times at which the CP transfers the received bytes to the CPU input area are also determined by the PLC control program using a RECEIVE handling block with job number 211.
When you assign parameters for the RECEIVE (ANR 211) the parameters QTYP, DBNR, QANF, QLAE are irrelevant.

The CI update points are independent of the communication via the bus. Communication between CP and slaves takes place constantly (cyclically).

If a slave station breaks down or no successful data exchange was possible, the input bytes assigned to this station are reset (to the value 0) and the station marked in the CI station list as failed (refer to Section 5.1.3) The CI also attempts to re-establish an application association to the failed station.

If the PLC changes from the RUN to the STOP mode, its CI output bytes are also reset to the value "0" and all application associations aborted. This is also the reaction to the change from CP RUN to CP STOP.

Procedure with the mode used

The reaction of CI for inputs and outputs is illustrated by the following diagrams.

The diagram below illustrates how the cycle-synchronized mode for output bytes functions.

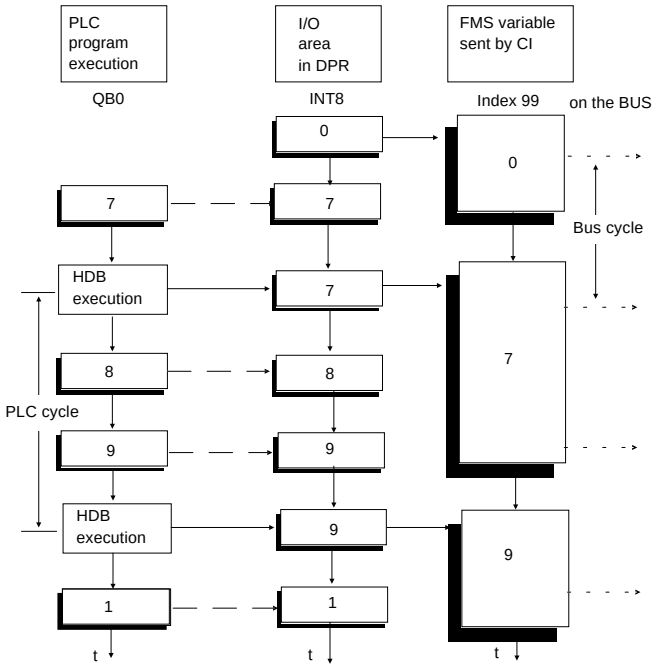


Fig. 5.2 Sequence of the Data Transfer from PLC to FMS Variable

The output area can be written to at any time. The transfer of the value to the variable is, however, not dependent upon this. The current data from the I/O area are entered in the FMS variable only when the send direct HDB (210) is run through. The data are accepted by the CI blocking further send directs, simulating the output bytes on the variables and transmitting the variables with a suitable number of write services to the corresponding field devices. The next acceptance of data can only be triggered for the output bytes after the PLC status byte "job complete" is set.

After a stoppage on the PLC, the CI resets the variables of the passive stations connected to the outputs to "0" and then aborts all cyclic application associations.

The diagram below illustrates how the cycle-synchronized mode for input bytes functions.

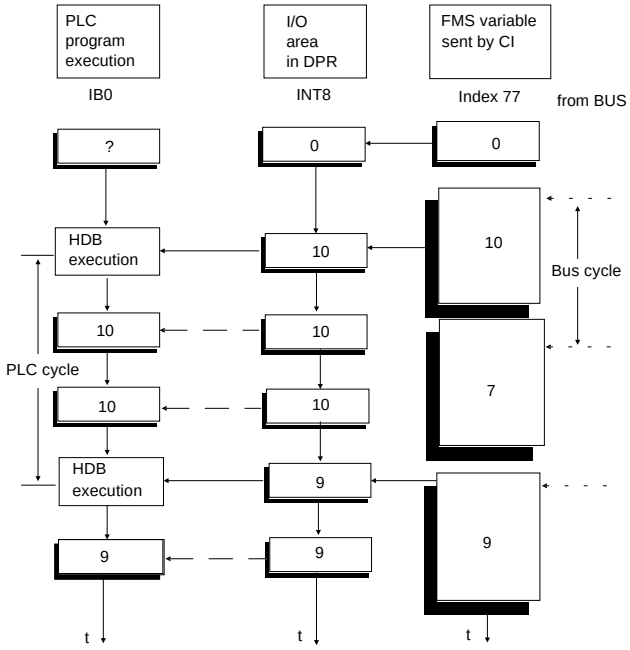


Fig. 5.3 Sequence of FMS Variable Transfer to the PLC

The FMS variables are read cyclically by the CI. The variable values are only accepted in the I area when a receive direct HDB (211) is run through. The acceptance is implemented by the CI blocking further receive directs, reading the variables with a suitable number of read services and simulating the variables on the input bytes. The next acceptance of output bytes can only be triggered after the PLC status byte has been set to "job complete". The PLC can access the updated input values at any time. The values are only updated with the next HDB call.

If the passive field device fails, the CI resets all the inputs assigned to it to "0".

The following applies to data transmission on the bus with CI:
Since the cycle time on the bus differs from that on the PLC, the data transmission on the bus is asynchronous to the data transfer by the PLC.

5.1.3 Status and Error Codes for the Cyclic Interface

The status word of the send (ANR 210) and receive (ANR 211) HDBs are available for status and group error bits. A detailed error message can be obtained by reading out the CI station list using a receive direct HDB (ANR 202).

Structure of the status word for HDB SEND (ANR 210) and RECEIVE (ANR 211) and CI station list (ANR 202)

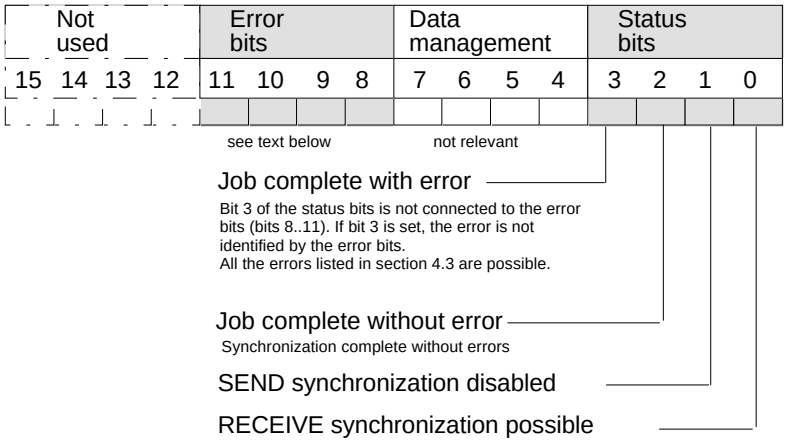


Fig. 5.4 Structure of the Status Word

If there is a group error message, bit 3 of the status word (status bit) is not set.
If a station has failed, the corresponding CI input bytes on the other stations are automatically reset by the CP 5431 FMS (to value 0). This is also the reaction to a start up!

Bits 8 to 11 are group error bits, bit 8 being used for the RECEIVE HDB and bit 11 for the HDB for the CI error list. All other bits are irrelevant. More detailed information about failed stations can be found in the CI station list.

Error bits of the CI station list (ANR 202)

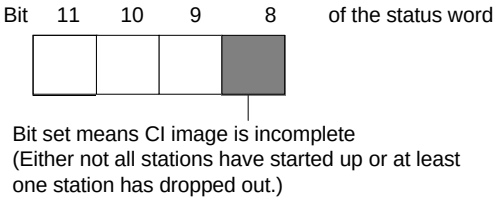


Fig. 5.5 Error Bits for the CI Station List (ANR 202)

Error bits for the RECEIVE HDB (ANR 211)

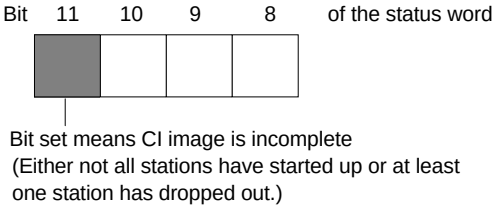
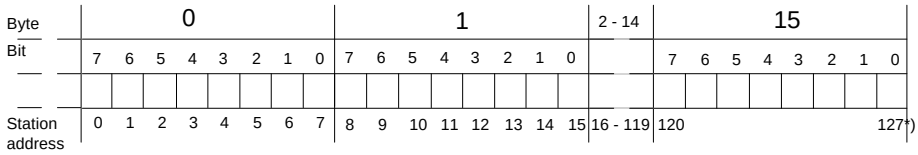


Fig. 5.6 Error Bits for the RECEIVE HDB (ANR 211)

Structure of the CI station list

The station list has a length of 16 bytes, with each bit assigned to a station address

All stations without cyclic application associations and stations for which all application associations are in the data transfer phase, are marked with "0". If any application association to a station is functioning incorrectly, this is marked with a "1" in the station list.



* The last bit in the station list is irrelevant since the permitted station addresses on the L2 bus are in the range from 0 to 126.

Fig. 5.7 Structure of the CI Station List

Error bits and updating the station list during data transmission

During start-up, the station list is initialized with "0". After the RECEIVE HDB has been executed the first time, all stations whose application associations are not in the data transfer phase are marked with bit value "1".

As long as one application association is not functioning correctly, the corresponding group error message is set both in the status word of the RECEIVE HDB (ANR 211) and in the status word of the CI station list (ANR 202).

When all the cyclic read and write application associations for a station are in the data transfer phase, the station is cleared from the station list and the bit in the station list is set to "0".

The station list is always updated during the execution of the RECEIVE HDB: This means that the station list read out is always up-to-date with the currently valid input image.

If an error occurs, (one or more application associations not in the data transfer phase), the CI station list can be read out at any time. If all the application associations are functioning correctly in the data transfer phase, the RECEIVE HDB for the CI station list is disabled.

Structure of the Parameter Assignment Error Byte:

The parameter assignment error byte (PAFE) informs you about various parameter assignment errors. When assigning parameters to individual blocks, you specify the address at which the information can be found. The significance of the individual bits is explained in Fig. 5.8.

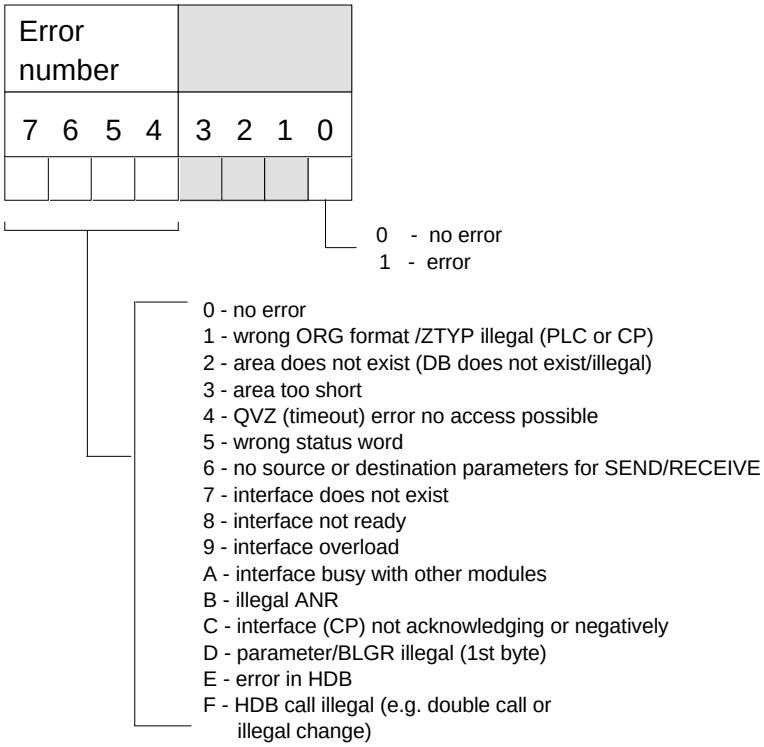


Fig. 5.8 Structure of the Parameter Assignment Error Byte "PAFE"

5.1.4 Handling Blocks in the S5 Program

When incorporating HDBs, remember that they should be called to suit the method of transmission to ensure updated values. If the inputs and outputs defined for the CI are located in the process image of the PLC, the outputs should be updated (SEND HDB) at the beginning and the inputs (RECEIVE HDB) at the end of the control program (refer to Fig. 5.9).

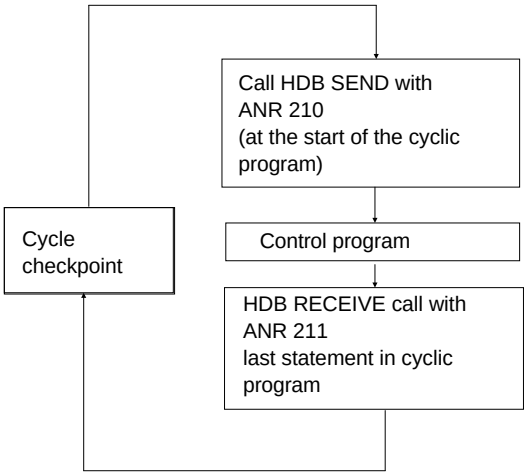


Fig. 5.9 Updating at the Cycle Checkpoint

If the I/Os are accessed directly, i.e. the inputs and outputs are not in the process image, the SEND HDB should be called directly after the updating of the outputs and the RECEIVE HDB directly before access to the inputs.

If the CI station list is to be read out in addition to the data transmission, this is possible after calling the RECEIVE HDB.

Evaluation of the CI station list is only useful when a group error bit is set in the RECEIVE HDB (refer to Fig. 5.6) or in the status word of the CI error list (refer to Fig. 5.5), since the CI station list is updated only in these cases.

5.2 Configuring

The communications processor provides the functions of Layer 7 for cyclic data transfer. The PG package SINEC NCM with COM 5431 FMS is used to configure the functions.

The screens you require for configuring are provided by SINEC NCM as shown in Fig. 5.10.

- I/O areas
- Cyclic application associations

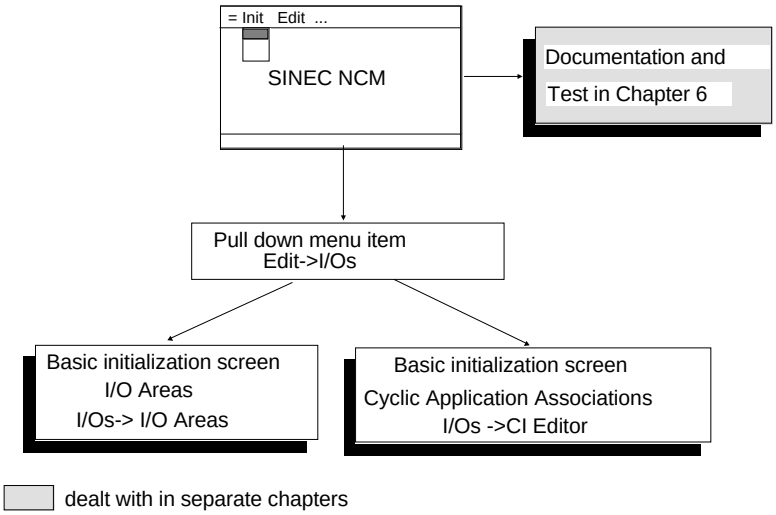


Fig. 5.10 Cyclic Communication (Configuring Cyclic Communication)

5.2.1 I/O Areas

Input and output areas of the SIMATIC PLC for the cyclic interface are assigned in the Input/Output Areas screen.

☞ **The configured areas are only valid after a power off -> power on transition.**

☞ **Simultaneous operation of DP and CI is not possible.**
Simultaneous operation of GP and CI is not possible.

Select Edit -> I/Os -> I/O Areas to call the following screen. The screen has the following structure:

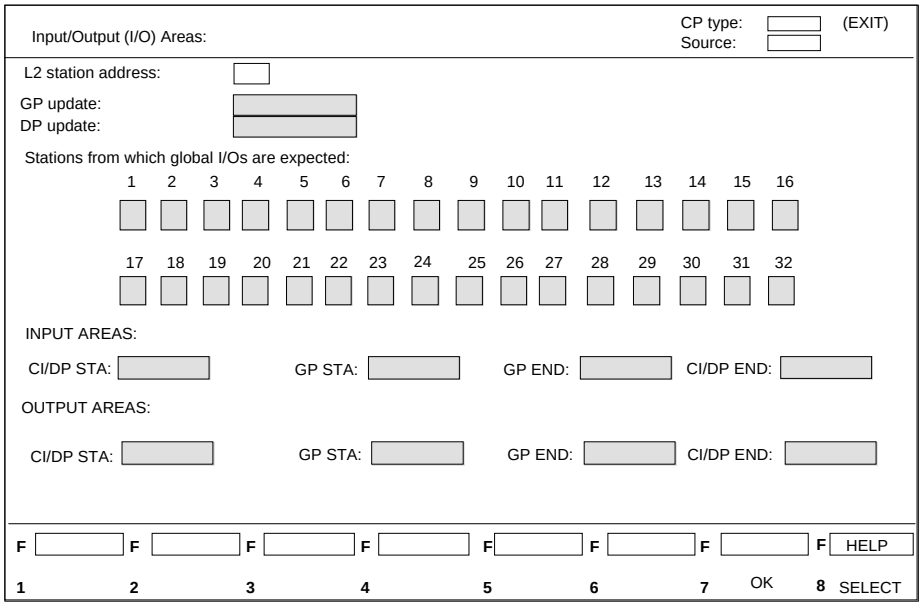


Fig. 5.11 Screen for Assigning Input/Output Areas

Output fields

L2 station address	The address of the current station is displayed.

Input fields:

GP-/DP update:	**Cycle-synchron**: updating at the cycle checkpoint by HDB. **Free**: updating of the I/O area by the CP.

Input areas:

CI/DP STA	Beginning of the (continuous) input area for the CI. (Range of values PY 0 .. 254, OY 0 .. 254 - only even numbers possible)
CI/DP END	End of the (continuous) input area for the CI. (Range of values PY 1 .. 255, OY 1 .. 255 - only odd numbers possible)

Output areas:

CI/DP STA:	Beginning of the (continuous) output area for the CI. (Range of values PY 0 .. 254, OY 0 .. 254 - only even numbers possible)
CI/DP END:	End of the (continuous) output area for the CI. (Range of values PY 1 .. 255, OY 1 .. 255 - only odd numbers possible)

☞ **The input or output area must always begin with an even byte number and must always end with an odd byte number. The fields remain empty if no input or output areas are required for the CI. The input area/output area for CI must not exceed a maximum of 256 bytes (each).**

☞ **The I/O area reserved for the CI must not be used for I/O modules! Online changes to the CI area only become effective after a POWER OFF/POWER ON transition!**

Function keys:

F7 OK	The "OK" key enters the data. If the database file does not yet exist, it is created when you confirm the entries.
F8 SELECT	If you press this key, a selection list is displayed with possible entries for fields which cannot be edited freely. Select values from the list with the cursor keys and enter them in the field with the return key.

5.2.2 Cyclic Application Associations

Once you have reserved the input/output areas for cyclic I/Os, you must now assign part of the reserved area to each field device (slave) using the CI editor.

The CI editor simulates the input and output fields on the variable objects of the slave with FMS. These are transferred via master-slave application associations with cyclic data exchange.

Select Edit -> I/Os -> CI Editor to call the following screen.

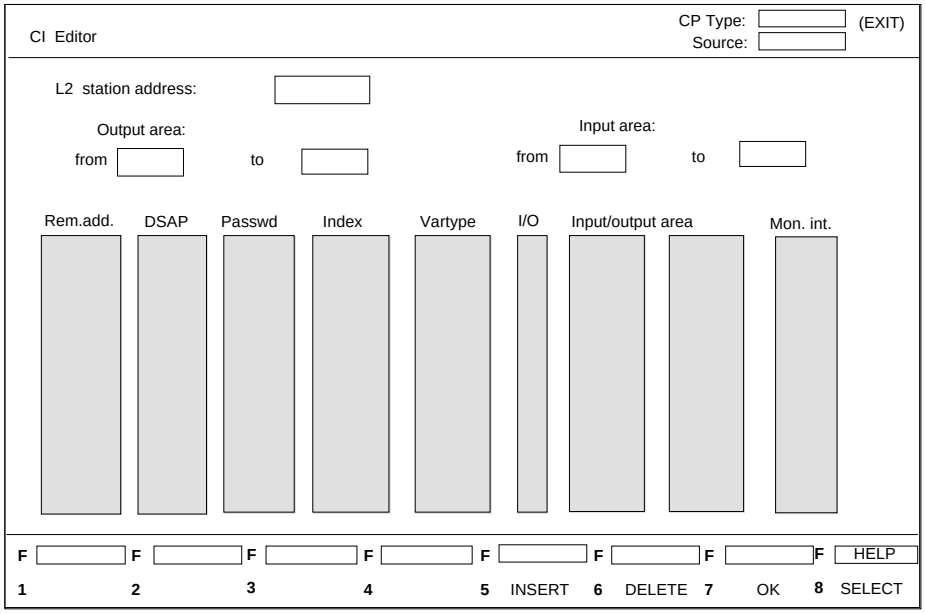


Fig. 5.12 Screen for CI Editor

Input fields

Rem. add.: In this column, you specify the L2 address of the slave station (remote address). (range of values: 0..126)

DSAP: The SAP of the slave station must also be specified. (range of values: 0, 2 .. 62)

Passwd: If a slave with access protection is accessed, the password must be specified here, otherwise the column remains empty.
(range of values: 1 .. 255)

Index: Index of the slave variable, to be accessed using the FMS services read or write.
(range of values: 15 .. 65535)

Vartype: Type of requested variable.

1st input field:
Selection of the variable type, see Table 5.1. Only standard types allowed.

2nd input field:
Size of the type specified in the 1st input field, see Table 5.1.
(range of values: default: IN 16 otherwise Table 5.1)

If the variable configured on the slave is a complex data type (e.g. structure), "OS" must be entered as the type.

Basic type Inp field 1	Size Inp field 2	Meaning	Corr. to S5 type
BO	no entry	Boolean	-
IN	8 16 32	integer, 8 bits integer, 16 bits integer, 32 bits	- KF -
UN	8 16 32	unsigned, 8 bits (unsigned number) 16 bits 32 bits	- KH KD
FP	32	floating point no. in MC 5 format, 32 bits	KG
BS	n	bit string, n fixed by the length of the I/O area	KM
OS	n	octet string, n fixed by the length of the I/O area	KY
VS	n	visible string, n fixed by the length of the I/O area	KS

Table 5.1 Variable Types

I/O: Specifies whether the reserved area is an input or output. (range of values: "I" or "Q")

Input/output area: Here, the I/O area is specified on which the variable will be simulated. (range of values: as displayed in input or output area)
from
This is the first byte of the field.
to
The last byte of the field.
(maximum 32 bytes for a field)

☞ **The values must be within the displayed input/output areas. The variable can be a maximum of 32 bytes long. Input areas must not overlap. The same output areas can be used as often as necessary, but make sure that they have the same variable type.**

Mon.int. — Here, you can set the monitoring time for variable updating. If no update occurs in this period, the application association is terminated.
(range of values: 18 ... 99999 *10 ms)

Output fields:

L2 station address — The address of the currently addressed station is displayed.

Input/ output area: — Displays the I/O area on which the variables to be configured will be simulated.
from
This is the first byte of the field.
to
The last byte of the field.

Function keys

F5 INSERT — An empty line is inserted at the current cursor position.

F6 DELETE — Deletes the input line marked by the cursor.

F7 OK — The "OK" key enters the data. If the database file does not yet exist, it is created when you confirm the entries.

F8 SELECT — If you press this key, a selection list is displayed with possible entries for fields which cannot be edited freely. Select values from the list with the cursor keys and enter them in the field with the return key.

5.3 Example of Data Transfer using Cyclic Communication

Introduction (master CP 5431 FMS with SIMOCODE slave)

This example illustrates the use of the CP 5431 FMS CI service in a simple application.

The data exchange with a SIMOCODE FMS device in the SIMATIC is directly via input and output bytes.
The CI service of the CP 5431 FMS actually handles the data exchange in the background using READ and WRITE application associations to the SIMOCODE.
For the user, this means that the transfer interface for data exchange with the CI service is the I/O area of the programmable controller.

In the example, QB 100 is used as the standard byte (transmitting commands to SIMOCODE) and IB 100-108 as message bytes (reading out the current status messages from the SIMOCODE).

A list of the control and message bits is shown in Table 5.2 and 5.3.
The detailed explanations of these control and message bits can be found in the manual for the SIMOCODE FMS.

Structure of the index 52 "control byte"

Access using password 1.

	Byte no.	Bit no.	Description
QB 100	Q100	7	On 2
	Q100	6	Off
	Q100	5	On 1
	Q100	4	Remote reset
	Q100	3	Emer. start
	Q100	2	Manual enable
	Q100	1	Change fetched
	Q100	0	Enable

Table 5.2 CP 5431 FMS - Example of CI Service

Structure of the index 46 "current status messages"

	Byte no.	Bit no.	Description
IB 100	I100	7	On 2
	I100	6	Off
	I100	5	On 1
	I100	4	Group fault
	I100	3	Changes
	I100	2	Interlock time active
	I100	1	RMON
	I100	0	RMOFF
IB 101	I101	7	Overload warning
	I101	6	Man/off
	I101	5	Man/on
	I101	4	CPU fault
	I101	3	RM 1
	I101	2	RM 2
	I101	1	RM 3
	I101	0	RM 4

Table 5.3 CP 5431 FMS - Example of CI Service

Assigning parameters for the CP 5431 FMS and the SIMOCODE FMS slave

With the CI service of the CP 5431 FMS used in the example, the output byte "100" is transferred as a control byte to the SIMOCODE using a write job. The updated status messages fetched by SIMOCODE are written to the input bytes 100 - 108.

With the read function, the CI service accesses index 46 and with the write function index 52 of the SIMOCODE FMS.

The variable type used is "BS".

The write access to index 52 "control byte" uses password "1".

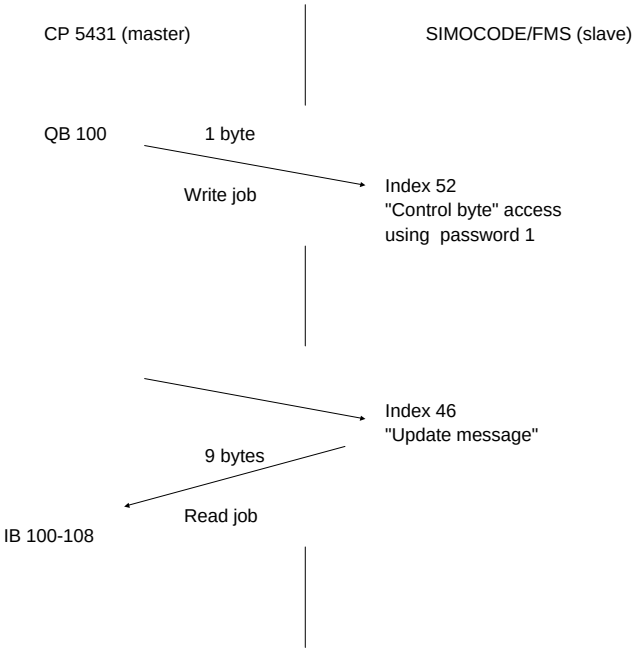


Fig. 5.13 CI Service "Overview of Sequence"

The description of the parameter assignment for the SIMOCODE FMS slave can be found in the SIMOCODE manual.
The following setting on the SIMOCODE is required for the example:
station address: 3
Bus parameter settings as on the master CP 5431 FMS.

To try out the CI service with the example program explained here, follow the steps below (see also Chapter 16, Volume 1).

➢ Under the network file FMS2@NCM.NET, transfer the CP 5431 FMS database file QZIBEIS.TN1 to the CP you are using CP.

➢ Transfer the STEP 5 file ZIBEISST.S5D to the programmable controller you are using (S5-115U).

The example files are on the COM/application example diskette.

Explanation of the example program

FB 210:
Cycle checkpoint for CI send

FB 211:
Cycle checkpoint for CI receive

FB 202:
Example of reading out the CI status list

FB 202 checks the group error bit for CI errors (F 240.3) in the ANZW of the cycle checkpoint FB call.
This bit is set to "1" if there is a CI error.
As soon as FB 202 recognizes that there is a CI error, the CI status/error list is read out by the receive function with ANR 202 and written to DB 202.

The recognition and reading of the CI error list is indicated by setting bit "FEHL" in FB 202.❑

6 Documentation and Testing

The screens required for documentation or testing are provided by SINEC NCM as shown in Fig. 6.1 and Fig. 6.2.

6.1 Documentation Functions

To give you the opportunity of producing listings of your configuration, the following documentation and print functions are integrated.

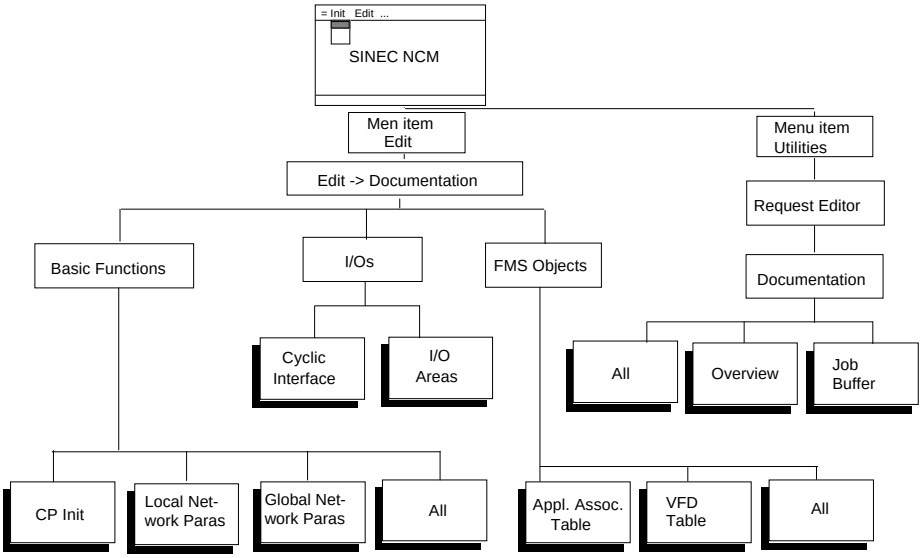


Fig. 6.1 Menu Structure for Documentation

With footer on/off in the "Init -> Edit" screen (refer to Chapter 6, Fig. 6.7 in Volume 1) you can specify a footer file in which you have already created a footer for the printout using the S5 footer editor.

With "Printer output on/off" option in the screen (refer to Chapter 6, Fig. 6.7 in Volume 1) you can decide whether to output solely on the screen or on both printer and screen.

6.2 Test

To allow you to test your program, the test functions shown in Fig. 6.2 are integrated in CI (cyclic interface) and ALI (FMS application associations).

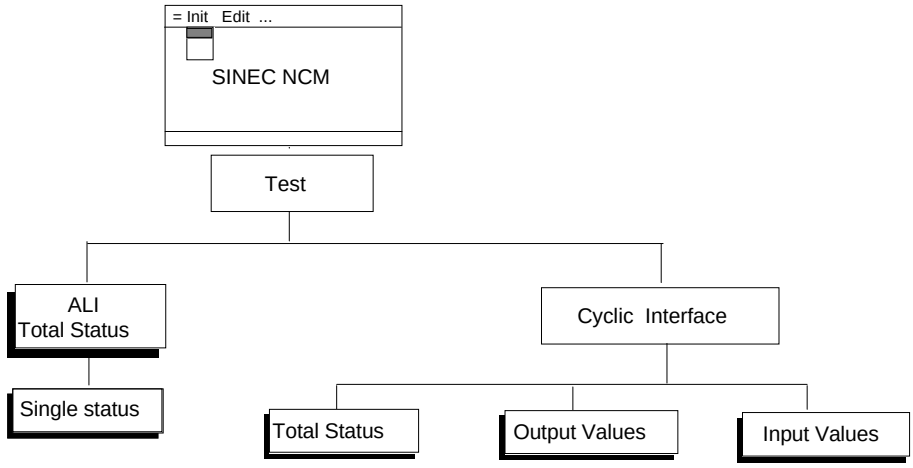


Fig. 6.2 Test Menu Structure

6.2.1 CI Test Functions

With the CI test functions on the PG, you can determine the statuses of individual system components during communication and, if necessary, localize errors.

6.2.1.1 Total Status of the CI Jobs

The total status of the CI jobs provides you with an overview of application association errors and FMS errors. The application association and FMS errors are displayed in hexadecimal code. By pressing a function key you obtain a more detailed explanation of the error. By selecting lines, the display can be restricted to the most important items. Further functions are explained under the function keys.

The screen has the following structure (examples of parameters):

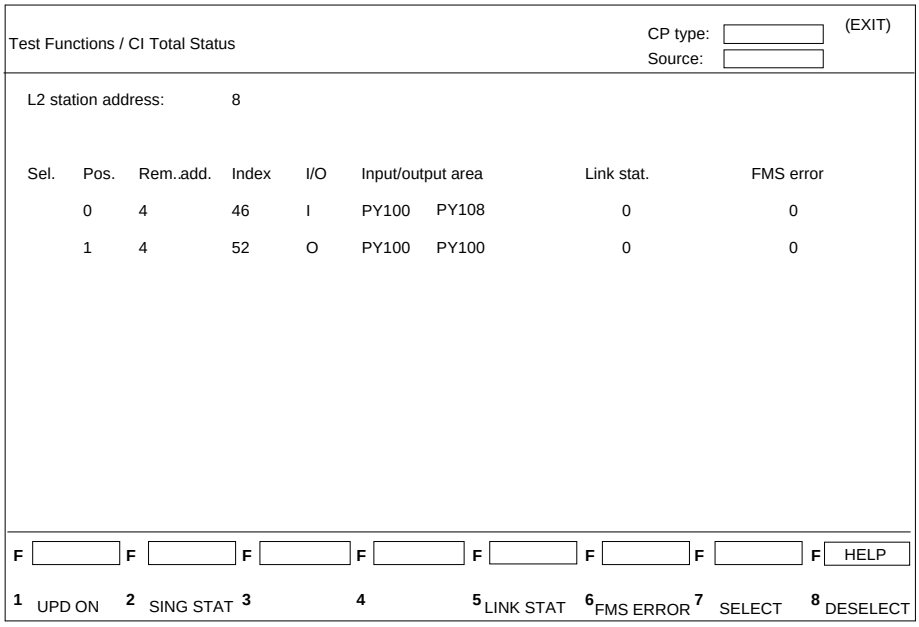


Fig. 6.3 CI Total Status

Output fields

L2 station address — The L2 address of the master station is displayed here.

Sel.: — Indicates with an asterisk that a line is selected.

Pos:	Consecutive line index.
Rem.add.:	Address of remote station on bus.
Index:	Index of the remote variable (decimal)
I/O:	Specifies whether an input or output area is involved.
Input/output area:	I/O area to which the application association is assigned (decimal)
Link stat:	Status of the job (hexadecimal code number). Press F5 to obtain an explanatory text.
FMS error:	FMS error number (hexadecimal code number). Press F6 to obtain an explanatory text.

Function keys

F1 UPD ON	Using this key, you can update the content of the screen. Pressing this key activates the automatic, cyclic updating of the screen data, pressing it again deactivates the automatic updating.
F2 SING STAT	This calls the screen for the single status of the variable. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the parameters.
F5 LINK STAT	With this function key, you obtain information about the status of a job. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the application association status (hexadecimal code).

F6 FMS ERROR	With this function key, you obtain information about the FMS error. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the error (hexadecimal code).
F7 SELECT	Using this key, or the enter key, you can select lines from the complete list of the total status screen by marking them with the inverse bar controlled by the cursor keys. These selected lines are then the only lines displayed after pressing the update key F1. You exit this mode with the ESC key.
F8 DESELECT	With this key, you can cancel the selection made with F7.

6.2.1.2 Single Status of CI Jobs

The single status of CI jobs function provides you with more detailed information about the status of a single job. The application association and FMS errors are coded in hexadecimal and more detailed information can be obtained with a function key.

The screen (with example parameters) can be called in the CI total status

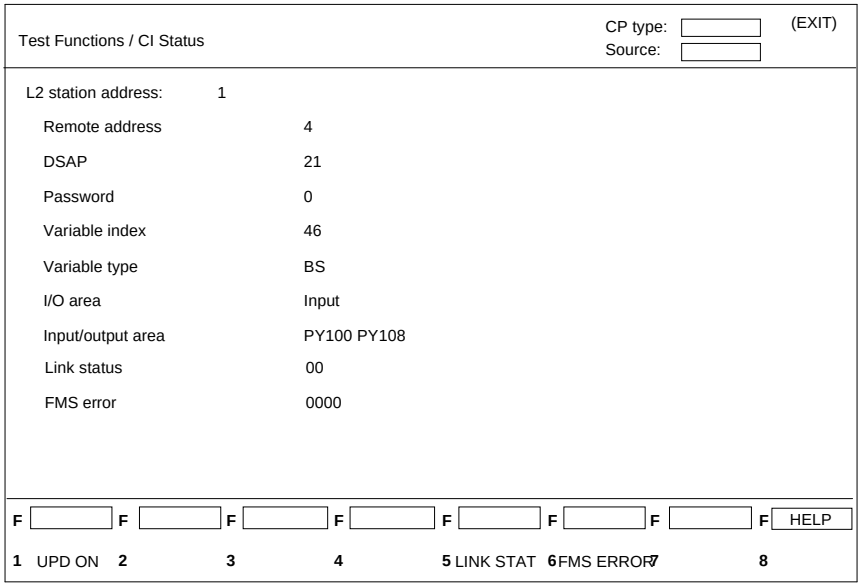


Fig. 6.4 CI Single Status

screen and has the following structure:

Output fields

L2 station address: The L2 address of the master station is displayed here.

Remote address: Address of remote station on bus (slave).

DSAP: SAP of remote station on bus (slave).

Password:	If the server object is protected by a password, this is displayed here.
Variable index:	Index of the server object.
Variable type:	Specifies the type of object.
I/O area:	Specifies whether an input or output area is involved.
Input/output-area:	Input/output area as I/O byte.
Link status:	Status of the job (hexadecimal code numbers).
FMS error:	FMS error identifier (hexadecimal code numbers)

Function keys

F1 UPD ON	Using this key, you can update the content of the screen. Pressing this key activates the automatic, cyclic updating of the screen data, pressing it again deactivates the automatic updating.
F5 LINK STAT	With this function key, you obtain information about the status of a job. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the application association status (hexadecimal code).
F6 FMS ERROR	With this function key, you obtain information about the FMS error. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the error (hexadecimal code).

6.2.1.3 Display of the CI Output Values

The CI output values are displayed per output byte. You cannot see the defined variable as a whole. The output bytes are displayed in ascending order.
The screen has the following structure (examples of parameters):

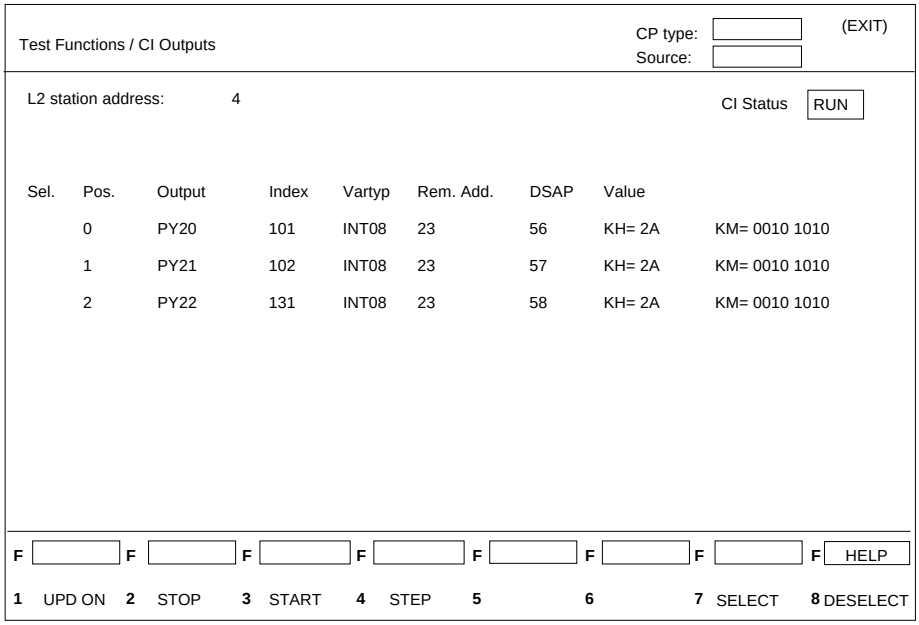


Fig. 6.5 CI Output Values

Output fields

L2 station address	The L2 address of the master station is displayed here.
Sel.:	Indicates with an asterisk that a line is selected.
Pos.:	Display position: application associations are numbered in ascending order (0 to 255).
Output:	Physical assignment of the output bytes of this station.

Index: Index of the variable assigned to the peripheral byte.

Vartyp: Type of the variable assigned to the peripheral byte.

Rem.add.: Address of remote station on bus (slave).

DSAP: Destination SAP of remote station.

Value: Value of the output in KH (hex.) and KM (bit).

Function keys

F1 UPD ON	Using this key, you can update the content of the screen. Pressing this key activates the automatic, cyclic updating of the screen data, pressing it again deactivates the automatic updating.
F2 STOP	With this function key, you send a stop message to CI. The output values are then no longer updated. The status field is then in the STOP status.
F3 START	With this function key, you send a start message to CI (warm restart). The status field is then in the RUN status.
F4 STEP	With this function key, you update the CI output byte once. The status field then changes to the STOP status.
F7 SELECT	Using this key, or the enter key, you can select lines from the complete list of the output values screen by marking them with the inverse bar controlled by the cursor keys. These selected lines are then the only lines displayed after pressing the update key F1. You exit this mode with the ESC key.

F8 DESELECT	With this key, you can cancel the selection made with F7.

With the page up and page down keys, you can page through all the lines of the list when there are too many lines for one screen page.

6.2.1.4 Display of the CI Input Values

The CI input values, just as the output values, are displayed per input byte. You cannot see the defined variable as a whole. The input bytes are displayed in ascending order

The screen has the following structure (examples of parameters):

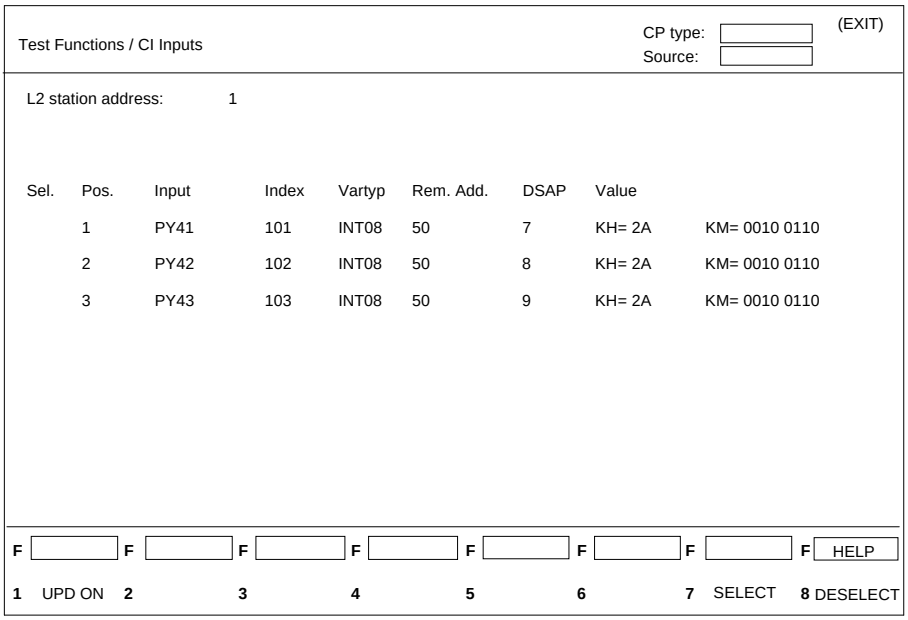


Fig. 6.6 CI Input Values

Output fields

L2 station address:	The L2 address of the master station is displayed here.
Sel.:	Indicates with an asterisk that a line is selected.
Pos:	Display position: application associations are numbered in ascending order (0 to 255).
Input:	Physical assignment of the input bytes of this station.

Index: Index of the variable assigned to the peripheral byte.

Vartyp: Type of the variable assigned to the peripheral byte.

Rem.add.: Address of remote station on bus (slave).

DSAP: Destination SAP of remote station.

Value: Value of the output in KH (hex.) and KM (bit).

Function keys

F1 UPDATE	Using this key, you can update the content of the screen. Pressing this key activates the automatic, cyclic updating of the screen data, pressing it again deactivates the automatic updating.
F7 SELECT	Using this key, or the enter key, you can select lines from the complete list of the input values screen by marking them with the inverse bar controlled by the cursor keys. These selected lines are then the only lines displayed after pressing the update key F1. You exit this mode with the ESC key.
F8 DESELECT	With this key, you can cancel the selection made with F7.

With the page up and page down keys, you can page through all the lines of the list when there are too many lines for one screen page.

6.2.2 ALI Test functions

With the ALI test functions on the PG, you can determine the status of individual system components in the communication and, if necessary, localize errors.

6.2.2.1 Total Status of ALI Jobs

The total status of ALI jobs test function displays the status of the ALI jobs as a list sorted according to the communication reference.

The screen has the following structure (examples of parameters):

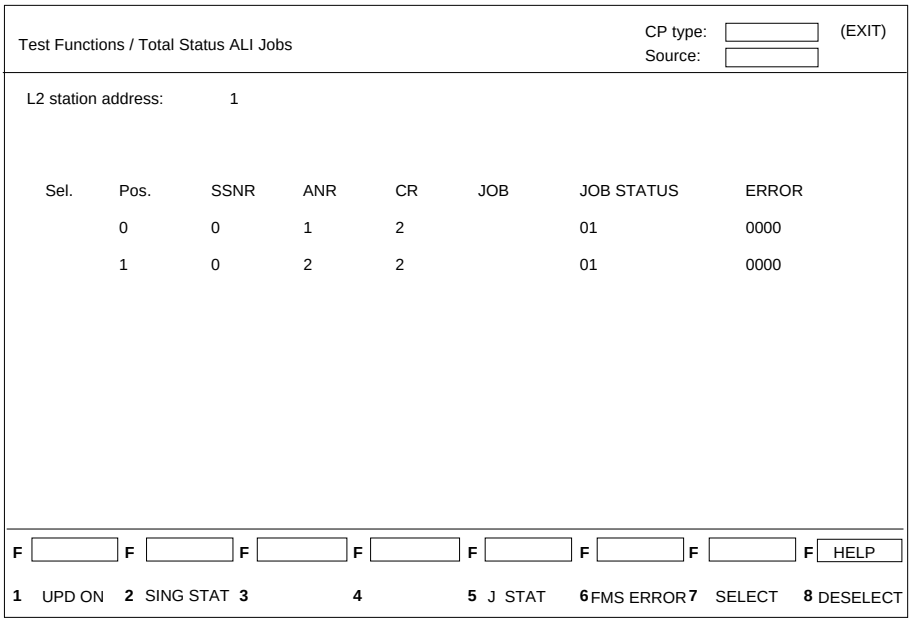


Fig. 6.7 ALI Total Status

Output fields

L2 station address: The L2 address of the master station is displayed here.

Sel.:	Indicates with an asterisk that a line is selected.
Pos:	Display position: application associations are numbered in ascending order (0 to 255).
SSNR:	Interface number
ANR:	Configured job number for client hjobs and confirmed server jobs. With unconfirmed server jobs the CP assigns an ANR with the value (200+CR).
CR:	Communication reference of the application association assigned during programming.
JOB:	Provides information about the action currently being performed by the CP or PLC. Job: READ, WRITE, STATUS, IDENTIFY.
JOB STATUS:	Indicates the job status (hex.)
ERROR:	Indicates error in job (ERRCLS/ERRCOD) (hex.) (see Appendix).

Function keys

F1 UPD ON	Using this key, you can update the content of the screen. Pressing this key activates the automatic, cyclic updating of the screen data, pressing it again deactivates the automatic updating.
F2 SING STAT	This calls the screen for the single status of the application association. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information.
F5 J STAT	With this function key, you obtain information about the status of a job. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the application association status (hexadecimal code).

F6 FMS ERROR	With this function key, you obtain information about the FMS error. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the error (hexadecimal code).
F7 SELECT	Using this key, or the enter key, you can select lines from the complete list of the total status screen by marking them with the inverse bar controlled by the cursor keys. These selected lines are then the only lines displayed after pressing the update key F1. You exit this mode with the ESC key.
F8 DESELECT	With this key, you can cancel the selection made with F7.

6.2.2.2 Single Status of ALI Jobs

This test function informs you about a single ALI job. The screen can be called in the ALI total status screen with F2 and has the following structure:

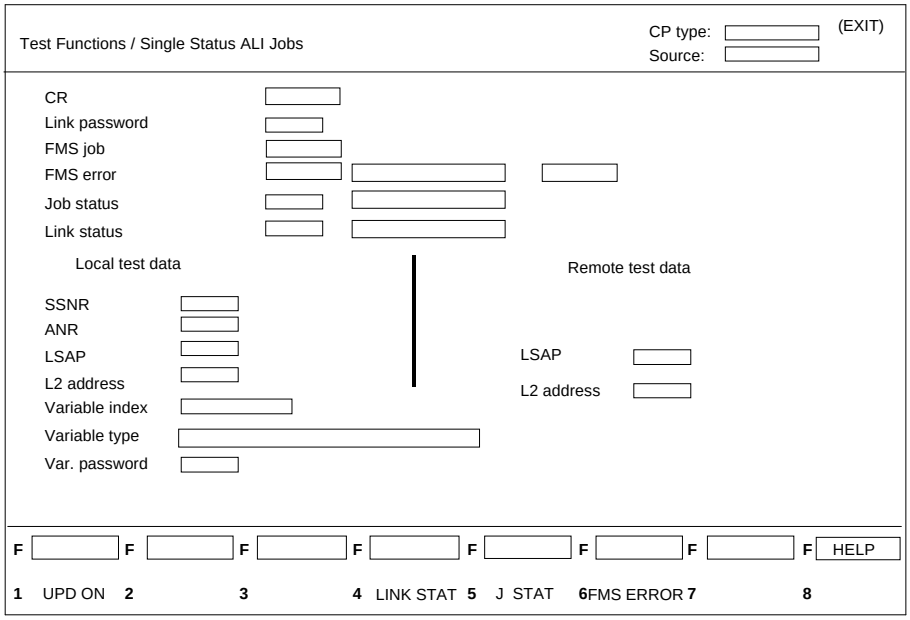


Fig. 6.8 ALI Single Status

Output fields

CR: Communication reference of the application association via which the job is executed.

Link password: Displays the password belonging to the application association.

FMS job:	Jobs with corresponding opcodes and texts are as follows: M-ST Status M-ID Identify V-RE Read V-WR Write V-IN Information report
FMS error:	Indicates error in job (ERRCLS/ERRCOD) (hex. and plain language) and provides information about origin of error. If the error message is from the remote partnere, "Remote" is displayed.
Job status:	Indicates the job status (hex.)
Link status:	Indicates the application association status (hex.).

Local/remote test data:

SSNR:	Interface number
ANR:	Configured job number for client hjobs and confirmed server jobs. With unconfirmed server jobs the CP assigns an ANR with the value (200+CR).
LSAP (local/ remote):	Local/remote SAP with which the job is executed
L2 address (local/remote):	The L2 address of the station is displayed here.
Variable index:	With variable services, the index or subindex of the variable addressed.

Variable type: With the local test data of the client, the type specified in the job buffer (if it is entered there) is displayed. If no type is specified in the job buffer, then when the application association is established, the loaded type description is taken as the basis. To achieve this, the appropriate index must be entered in the "Access to variables" field in the CP configuration. If there is no entry here, there is no type specified. On the server, the type actually configured is listed if the variable object exists.

Var. password: The password configured for the variable

Function keys

F1 UPD ON	Using this key, you can update the content of the screen. Pressing this key activates the automatic, cyclic updating of the screen data, pressing it again deactivates the automatic updating
F4 LINK STAT	With this function key, you obtain information about the status of a application association. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the application association status (hexadecimal code).
F5 J STAT	With this function key, you obtain information about the status of a job. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the job status (hexadecimal code).

F6 FMS ERROR	With this function key, you obtain information about the FMS error. Using the cursor keys, you can select a line with the inverse bar and obtain more detailed information about the error (hexadecimal code). ❑

7 Request Editor User-Friendly Interface for Generating Job Buffers

To support you when creating job buffers, there is a tool that runs under SINEC NCM which has a user interface similar to control system flowchart and with which you can configure the individual services and the corresponding job buffers stored in a data block of the S5 program file.

Help texts are available for each input field in which the possible inputs are explained. These are obtained by pressing the HELP key on the PG to open a window in which the help text for the current input field is displayed.

In fields in which only certain options are valid, you can select the options using the cursor keys.

7.1 Structure of the Job Buffer

The following diagram (Fig. 7.1) illustrates the data structure of a job buffer.

The following points are significant:

1. Each job buffer is preceded by its length (in words).

2. Each job buffer begins with a fixed structure
   - Opcode (2 words, 4 characters)
   - Timeout (1 word, 16 bits fixed point)
   - Reserve (1 word)

3. The structure following this is matched to the type dynamically and its length depends both on the type and parameters.

4. No job buffer can exceed the maximum length of 256 bytes.

A data block can contain several job buffers. The number of job buffers in the DB is limited by its length (2042 words).

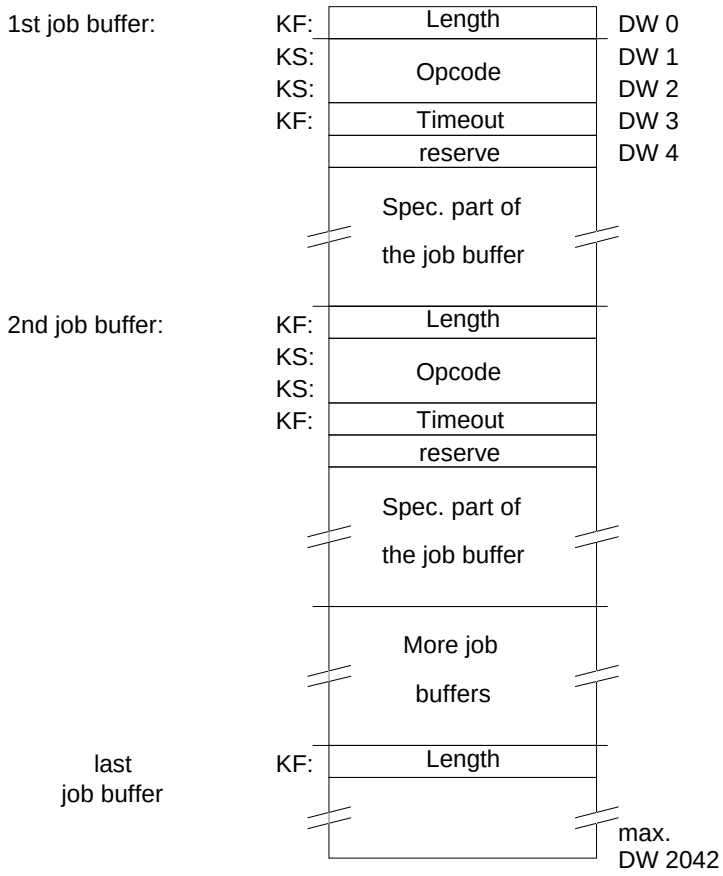


Fig. 7.1 Data Structure of the Job Buffer

7.2 Description of the Request Editor

The tool has the following structure:

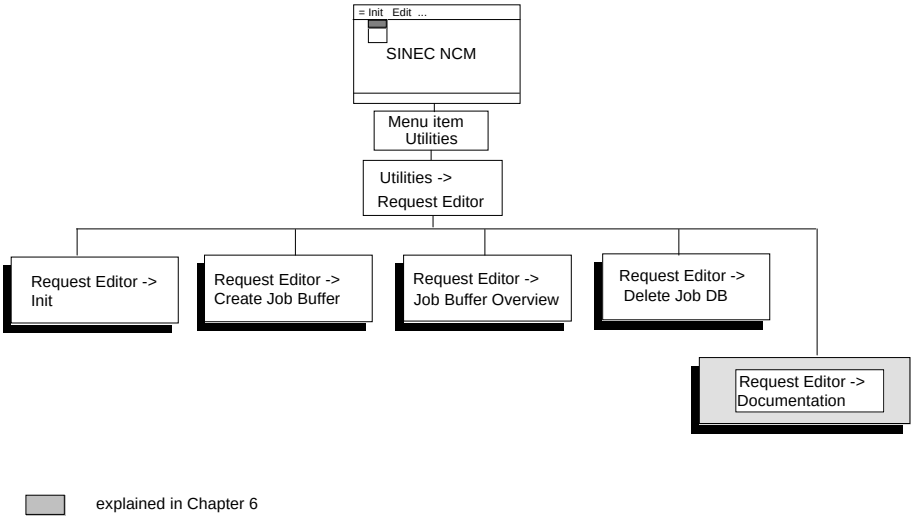


Fig. 7.2 Structure of the Request Editor

In the COM screen, called with the menu item "Utilities->Request Editor->Init", you select an S5 program file and a data block which are assigned to the job buffer (Section 7.2.1).

With the menu item **"Utilities->Request Editor->Create Job Buffer"** you call the type selection screen for setting up a job buffer for a specific service (Section 7.2.2).

With the menu item **"Utilities->Request Editor->Job Buffer Overview**" you call a screen which provides you with an overview of the programmed job buffers (Section 7.2.3).

With the menu item **"Utilities->Request Editor -> Delete Job DB"** you call a screen with which you can delete job buffer DBs (Section 7.2.4).

With the menu item **"Utilities->Request Editor->Documentation"** there is a submenu described in Chapter 6 "Documentation and Testing".

7.2.1 Initializing the Request Editor

Select Request Editor->Init to call the following screen.

The screen for initialization has the following structure

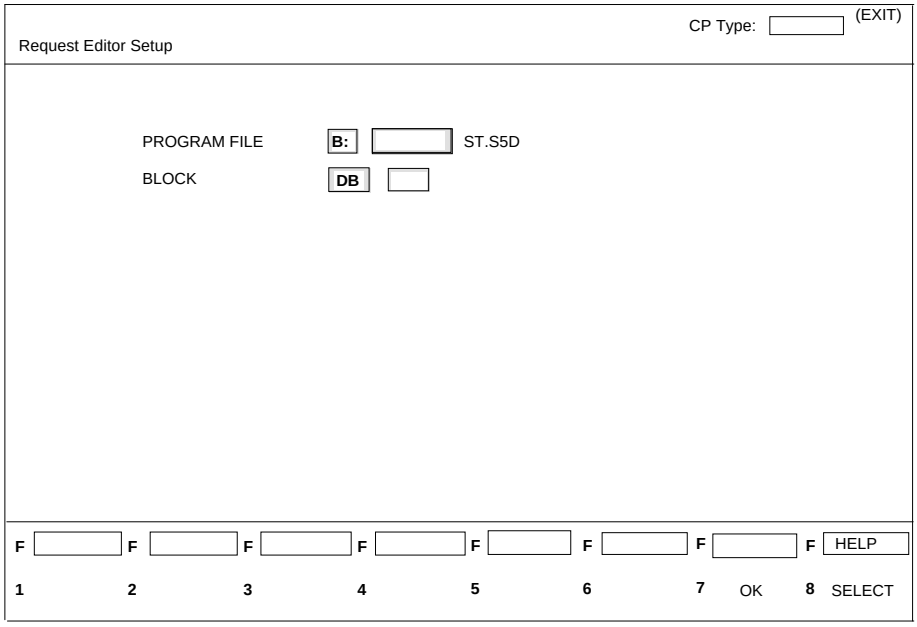


Fig. 7.3 Request Editor -> Init Screen

Input fields:

PROGRAM FILE:	Specifies the S5 program file to which the job buffer will be assigned. If the file does not exist it is created. If the specified file is read-only, an appropriate message is displayed in the message line. In this case, no new job buffers can be edited, but only existing buffers output.
BLOCK:	1st input field: Specifies the type of block containing the job buffer or that will contain the job buffer. Possible values: DB, DX (In the following descriptions, both block types are simply described as "data blocks". Default: DB 2nd input field: Number of the data block containing the job buffer or that will contain the job buffer. If the data block does not yet exist in the program file, it is created. In this case the following message is displayed in the message line: BLOCK DOES NOT EXIST Neither a data block preheader nor comment block is created.

Function keys:

F7 OK	Enters the data you have input.
F8 SELECT	Possible parameters are displayed for selection.

7.2.2 Input Screen Form

Select "Request Editor-> Create Job Buffer" to call the following screen. The screen has the following structure:

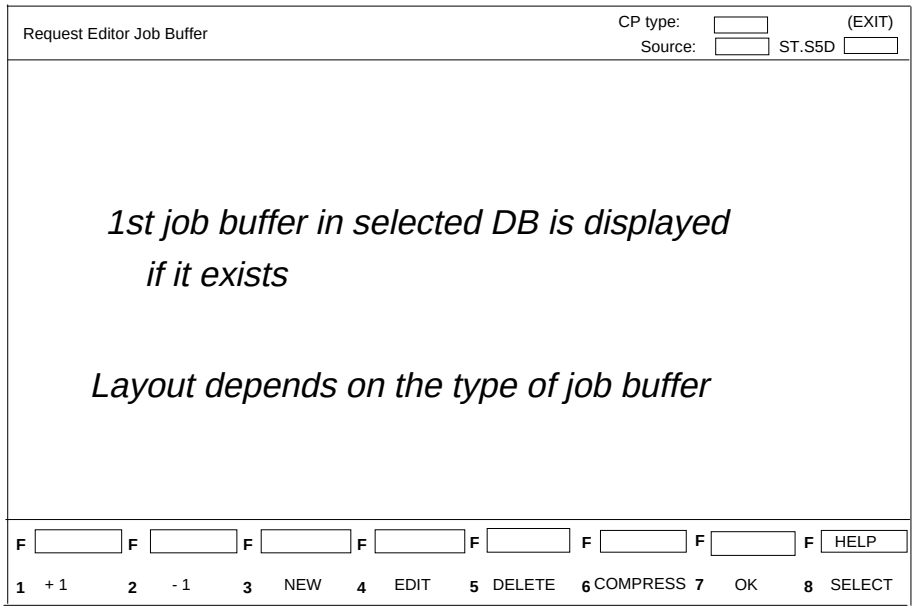


Fig. 7.4 Input Screen

If there is no job buffer in the selected DB, the following message is displayed in the message line:
NO JOB BUFFER EXISTS

If the data block exists, but does not contain a job buffer, the following message is output:
ERROR IN DATA BLOCK, DELETE?

Here, you must decide whether to abort the function or whether the selected data block should be deleted before editing the job buffer.

The functions of the screen are described in the explanation of the function keys.

Function keys:

F1 +1	Finds the next job buffer in the data block and displays it.
F2 -1	Finds and displays the previous job buffer.
F3 NEW	Input of a new job buffer at the end of the current data block. Next screen: service selection. (refer to Fig. 7.5) If the selected data block cannot accept any more job buffers, but sufficient space would result from compressing the block (see below), the following message appears: BLOCK TOO LARGE, FIRST COMPRESS. If compressing the block would still not provide sufficient space for a further job buffer, the following message is displayed: BLOCK TOO LARGE, COMPRESSING NO HELP.
F4 EDIT	You can modify an existing job buffer. The original job buffer is automatically deleted and a new buffer is appended to the end of the block. **The call parameters for the "SEND DIR" for triggering the service are changed.**
F5 DELETE	Deletes the current job buffer from the data block. To prevent the remaining job buffers in the data block from being shifted together, the job buffer is not deleted but declared invalid, it can nevertheless no longer be restored. To prevent you accidentally deleting a job buffer, you must confirm the prompt: delete (YES/NO). You remain in the input screen.

F6 COMPRESS	Compresses the selected data block. This means that all invalid job buffers are removed and the valid job buffers are shifted together. The following message then appears in the message line: CAUTION: X-REF WILL CHANGE, PLEASE CONFIRM (xxx BYTES FREE) xxx indicates the number of free bytes in the data block. To prevent accidental compressing of the data block, you must confirm your intention. On completion of the function, the following text appears in the message line: COMPRESSING DONE, xxx BYTES FREE. If the data block does not contain any invalid job buffers, the following text appears in the message line: NO INVALID JOB BUFFER EXISTS, xxx BYTES FREE. Following this, you can decide whether you want to compress the data block or abort the function.
F7 OK	Completes the entry of new job buffers and writes the data block back to the program file.
F8 SELECT	Possible parameters are displayed for selection.

The service selection screen lists all the possible services for the user to select from.

The screen is called with F3 NEW in the input screen and has the following structure:

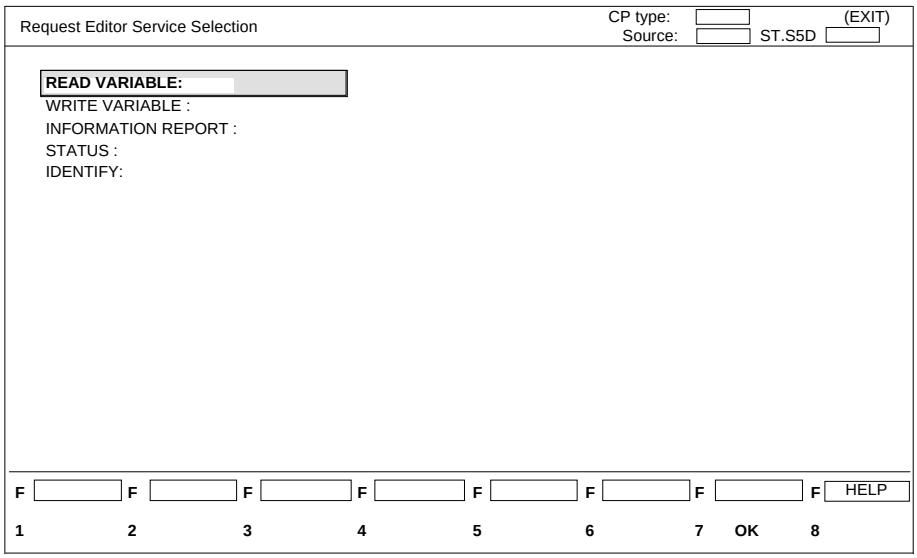


Fig. 7.5 Service Selection Screen

The following options are available:

READ VARIABLE
Read a variable from another station.

WRITE VARIABLE
Transfer the current value of a variable to another station.

INFORMATION REPORT
Report the current value of a local variable to a remote station.

STATUS
Request the status (physical and logical) of another CPU.

IDENTIFY
Request information about the type and characteristics of a remote station.

Select the function you require using the cursor keys, the currently selected function is displayed inversely on the screen.

Default: READ VARIABLE

Function keys:

F7 OK	Select a job buffer for the currently selected function Next screen: (dependent on the selected function) READ VARIABLE (Section 7.2.2.1) WRITE VARIABLE (Section 7.2.2.2) INFORMATION REPORT (Section 7.2.2.3) STATUS (Section 7.2.2.4) IDENTIFY (Section 7.2.2.5)

7.2.2.1 Read Variable

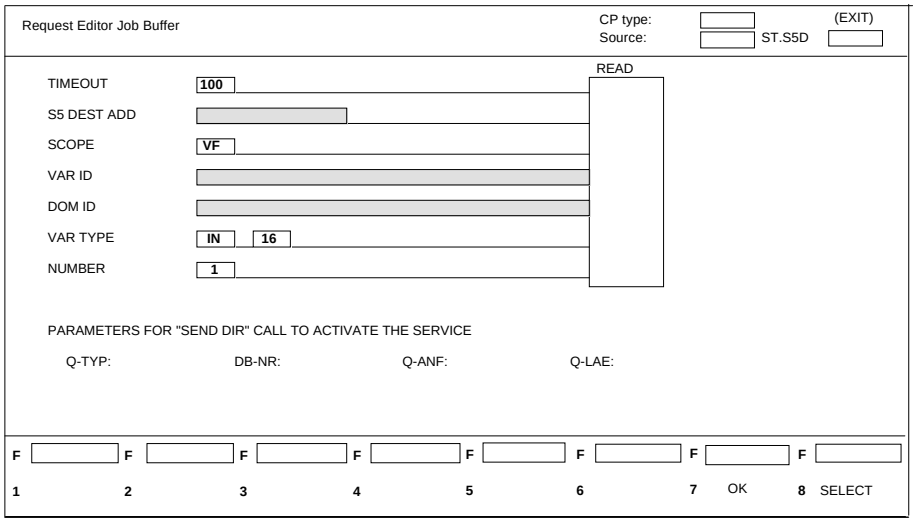


Fig. 7.6 Read Variable Screen

Input fields

TIMEOUT: 1 word, format: KF
Specifies the maximum length of time the user program will wait for an acknowledgement for the service (i.e. the maximum dwell time of the job in the PC). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 DEST ADD: The address in the S5 system at which the value of the requested variable will be stored by the CP.

Dest type: DB, DX
DB no.: 1..255
Start: 0..2042

It is not possible to specify the length, this is determined implicitly by the type of variable.

The address is input in the usual way in COM programming, i.e. by separating the individual parameters of the S5 address with a blank.

SCOPE — Specifies the validity of the requested variable in the other system. For FMS this is always "VF" which cannot be changed.

VAR ID — Index, subindex of the variable of a variable on the partner. When using a subindex, the index and subindex are separated by a comma.

DOM ID — When configuring the CP 5431 FMS, no entry can be made here.

VARTYP — Specifies the type of the requested variable. (default: IN 16)

1st input field:
Input of the basic type, see Fig. 7.7.

2nd input field:
Length of the type specified in the first input field.

Basic type 1st inp. field	Size 2nd input field	Comment
BO	no entry	Boolean, 0=False, $FFFF_H$=True
IN	8, 16, 32	Integer, size = no. of bits per element
UN	8, 16, 32	Unsigned integer, size as above
FP	32	Floating point, size as above
BS	8...1864 (1896)*	Bit string, size = no. of bits in string
OS	1...233 (237)*	Octet string, size = no. of bytes in string
VS	1...233 (237)*	Visible string, size = no. of bytes in string

* Values in brackets apply to read jobs.

Fig. 7.7 Basic Types

If nothing is specified as the basic type, the index of the variable in the job buffer must also be specified when programming the application association on the client. The client then obtains the type description of the variable automatically from the partner with the Get OL service when the application association is established

NUMBER**:** Number of elements for arrays
Default: 1 (no array)

Function keys:

F7
OK

Completes the input and stores the newly edited job buffer in the main memory of the programmer.
Next screen: input.

The parameters of the last edited job buffer are displayed on the screen. In addition to this, call parameters for the "SEND DIR" call to trigger the service are also displayed.

F8
SELECT

Possible parameters are displayed for selection.

Example of a "selection menu"

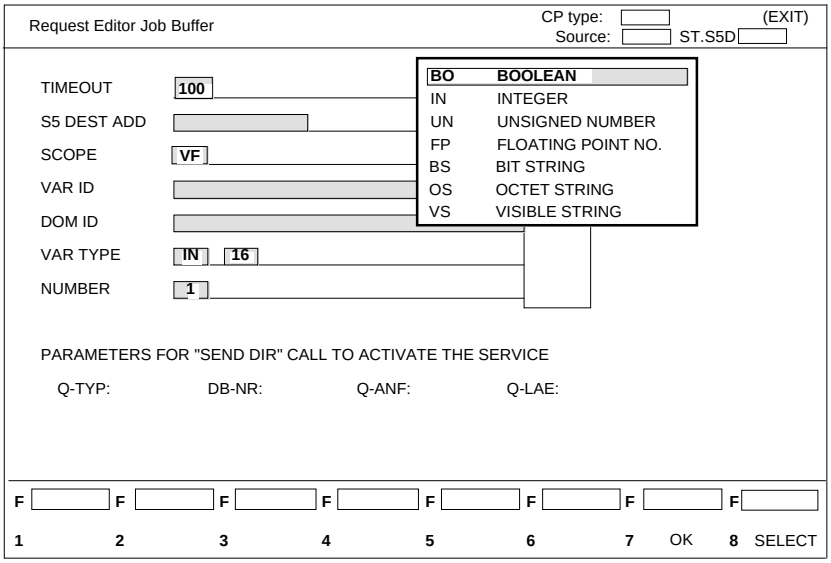


Fig. 7.8 Read Variable Screen

Assumption:
You want to make an entry in the "VAR TYPE" input field and you cannot remember the abbreviations for the types (press the SELECT key on the PG). By moving the inversely displayed line in the help window with the cursor up and cursor down keys, you can select the required type and enter it in the field by pressing the enter key or carriage return key. The help window then disappears.

For the second input field specifying the type, a help window is displayed to help you make the input, only the options allowed for the selection made in the first window are displayed.

7.2.2.2 Write Variable

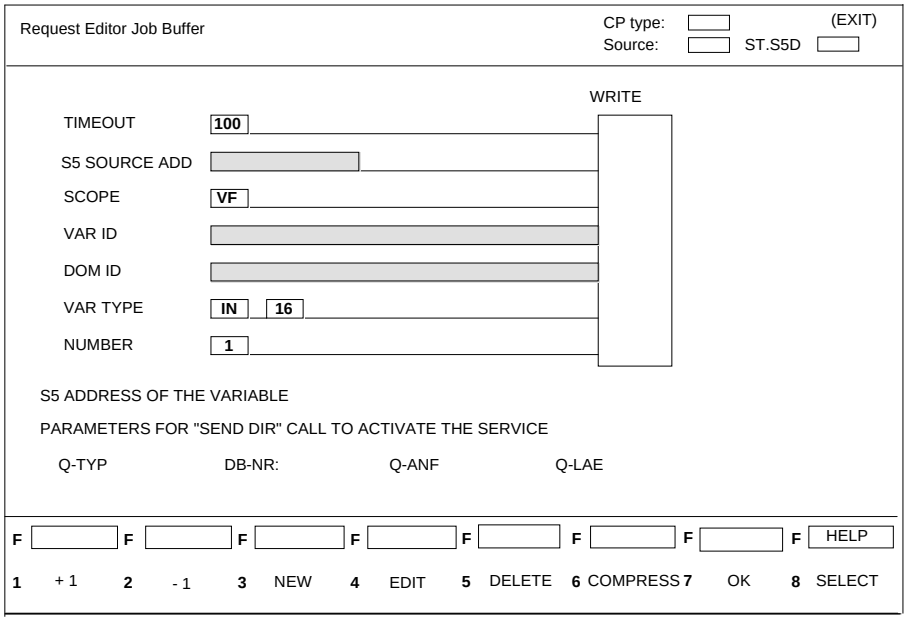


Fig. 7.9 Write Variable Screen

Input fields:

TIMEOUT: 1 word, format: KF
Specifies the maximum length of time the user program will wait for an acknowledgement for the service (i.e. the maximum dwell time of the job in the PC). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 SOURCE ADDRESS: Address in the S5 system at which the user program has stored the value of the variable to be sent.
Source type: DB, DX, DA
DB no.: 1..255
Start: 0..2042

It is not possible to specify the length, this is determined implicitly by the type of variable

Note on the source type "DA":
This means that the user program stores the value of the variables after the parameters. In this case, the parameters "DB no" and "start" are invalid.

SCOPE Specifies the validity of the requested variable in the other system. For FMS this is always "VF" which cannot be changed.

VAR ID: Index, subindex of the variable of a variable on the partner. When using a subindex, the index and subindex are separated by a comma.

DOM ID**:** When configuring the CP 5431 FMS, no entry can be made here.

VARTYP: Specifies the type of the requested variable.(default: IN 16

1st input field:
Input of the basic type, see Fig. 7.7.

2nd input field:
Length of the type specified in the first input field.

NUMBER: Number of elements for arrays.
Default: 1 (no array)

Function keys

F7 OK	Completes the input and stores the newly edited job buffer in the main memory of the programmer. Next screen: input. The parameters of the last edited job buffer are displayed on the screen. In addition to this, call parameters for the "SEND DIR" call to trigger the service are also displayed.
F8 SELECT	Possible parameters are displayed for selection.

7.2.2.3 Information Report

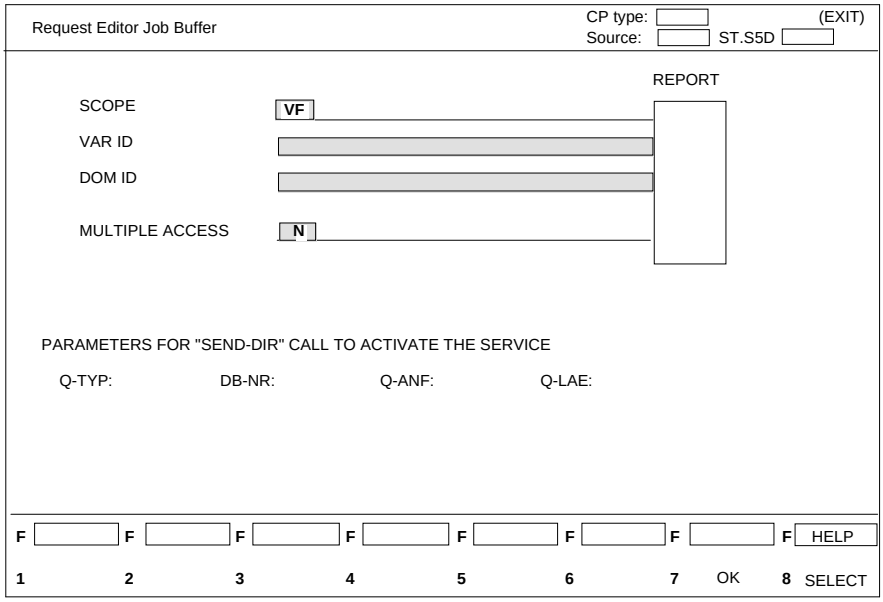


Fig. 7.10 Information Report Screen

Input fields:

SCOPE	Specifies the validity of the requested variable in the other system. For FMS this is always "VF" which cannot be changed.
VAR ID:	Index, subindex of the variable of a communication object (VFD variable). When using a subindex, the index and subindex are separated by a comma.
DOM ID:	When configuring the CP 5431 FMS, no entry can be made here.
MULTIPLE ACCESS:	Multiple access is not supported on the CP 5431 FMS.

Function keys:

F7 OK	Completes the input and stores the newly edited job buffer in the main memory of the programmer. Next screen: input. The parameters of the last edited job buffer are displayed on the screen. In addition to this, call parameters for the "SEND DIR" call to trigger the service are also displayed.
F8 SELECT	Possible parameters are displayed for selection.

7.2.2.4 Status

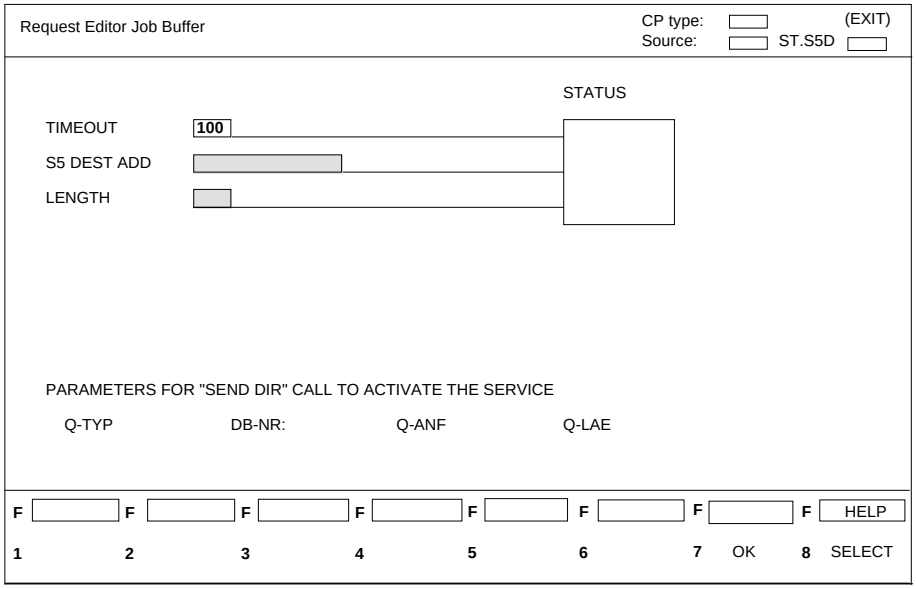


Fig. 7.11 Status Screen

Input fields

TIMEOUT: 1 word, format: KF
Specifies the maximum length of time the user program will wait for an acknowledgement for the service (i.e. the maximum dwell time of the job in the PC). This is specified in multiples of 0.1 sec. The value 0 means no time monitoring. If the job cannot be completed within the specified time, the job and all others active on this application association (communication reference) are deleted. The application association is then terminated and established again.

S5 DEST ADD:	Address in the S5 system at which the required status information will be stored by the CP. Dest. type: DB, DX DB no.: 1..255 Start: 0..2042
LENGTH:	The "length" parameter specifies how many data bytes can be written to the data block by the CP. The value -1 means that all the data of the acknowledgement can be entered.

Function keys

F7 OK	Completes the input and stores the newly edited job buffer in the main memory of the programmer. Next screen: input. The parameters of the last edited job buffer are displayed on the screen. In addition to this, call parameters for the "SEND DIR" call to trigger the service are also displayed.
F8 SELECT	Possible parameters are displayed for selection.

7.2.2.5 Identify

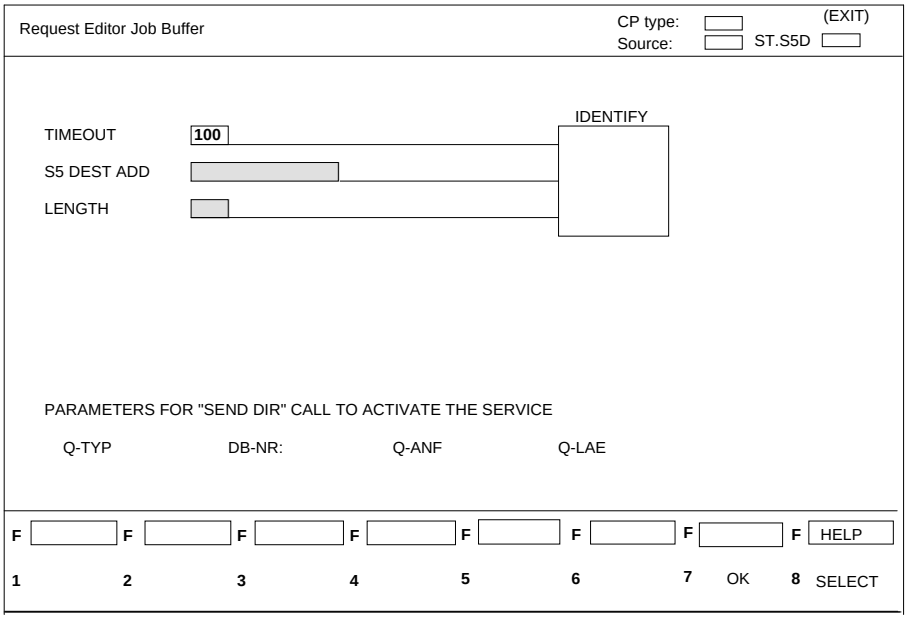


Fig. 7.12 Identify Screen

Input fields

TIMEOUT: Acknowledgement monitoring time for the job in units of 0.1 sec. Default: 10 sec. If the job is not completed within this time, the CP aborts the job. If you do not enter a value in the field, the CP assumes that no time monitoring is required for the job.

S5 DEST ADDRESS: Address in the S5 system at which the required information will be stored by the CP.
Dest. type: DB, DX
DB no.: 1..255
Start: 0..2042

LENGTH: The "length" parameter specifies how many data bytes can be written by the CP into the data block. The value -1 means that all the data of the acknowledgement can be entered.

Function keys:

F7 OK	Completes the input and stores the newly edited job buffer in the main memory of the programmer. Next screen: input. The parameters of the last edited job buffer are displayed on the screen. In addition to this, call parameters for the "SEND DIR" call to trigger the service are also displayed.
F8 SELECT	Possible parameters are displayed for selection.

7.2.3 Job Buffer Overview

Here, a list is displayed containing all the call parameters for the job buffers edited in the data block.

These are as follows:

- Type of job buffer (opcode)
- S5 address of the job buffer
  (parameter of the send direct HDB for triggering the service in the PLC)
- Name of the object
- S5 address of the object

The screen you activate with Request Editor -> Job Buffer Overview has the following structure:

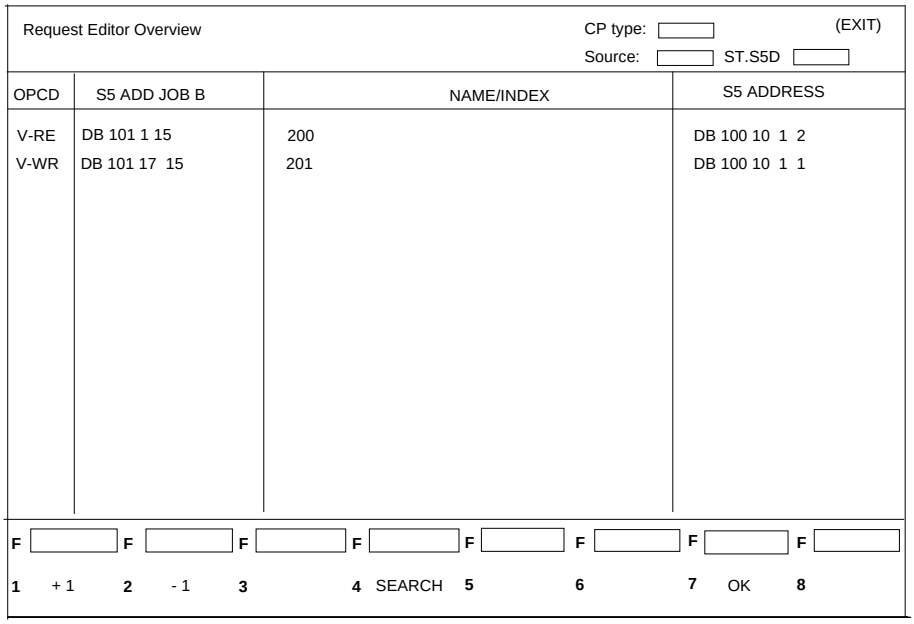


Fig. 7.13 Job Buffer Overview

Each output line corresponds to a job buffer in the data block.

Output fields

OPCD: Output of the selected service of the job buffer

- V-RE: Read variable
- V-WR: Write variable
- V-IN: Information report
- M-ST: Status
- M-ID: Identify

S5 ADD JOB B: Display of the source address for calling the "SEND DIR" to trigger a service.

NAME/INDEX Display of the index contained in the job buffer.

S5 ADDRESS: Display of the S5 address contained in the job buffer. With variables services this is the source or destination address of the variable.

Function keys

F1 +1	Page forward to the next job buffers if there are more job buffers than fit on one screen page.
F2 -1	Page backward to previous job buffers if there are more job buffers than fit on one screen page.
F4 SEARCH	Using the cursor keys (up/down) you can select a job buffer (inversely displayed line) and display it with the function key F4.

F7 OK

Using the cursor keys (up/down) you can select a job buffer (inversely displayed line) and display it with the function key F7.

7.2.4 Delete Job DB

Select Request Editor -> Delete Job DB to call the following COM screen:

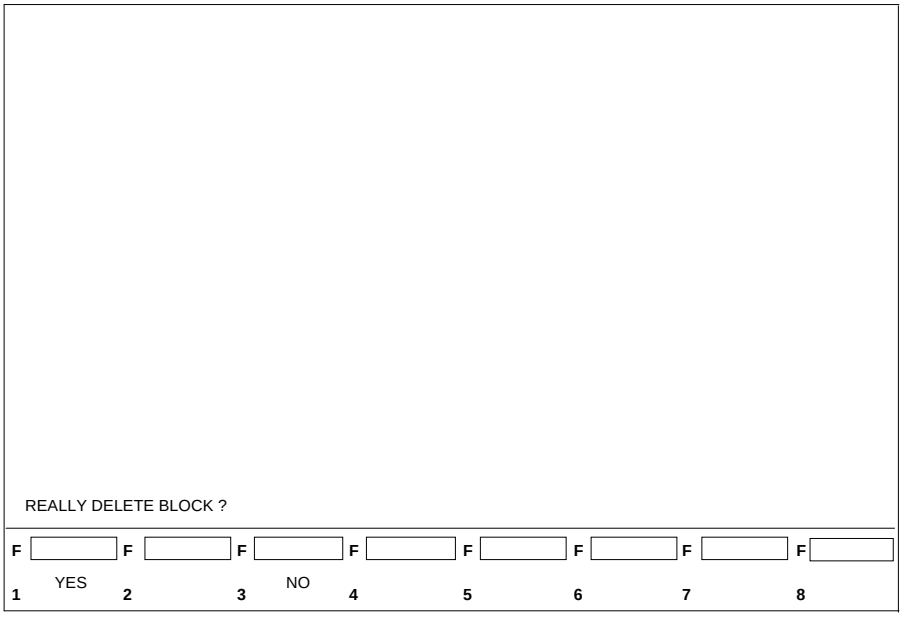


Fig. 7.14 Delete Job DB Screen

In the message line the following prompt appears:

REALLY DELETE BLOCK?

which you can answer with the function keys F1 "YES" or F3 "NO".

Function keys

F1 YES	Block is deleted
F3 NO	Block is retained ❑

8 Appendix

8.1 Errors

This appendix describes the error numbers and the causes of errors that can occur when operating the CP 5431 FMS. The error numbers consist of the parameters ERRCLS and ERRCOD of the FMS protocol. There are two possible reasons for errors occurring:

- The CP receives a job from the PLC (S5 as requester) that is either incorrect or cannot be executed at the present time. The job is acknowledged negatively to the PLC (job completed with error) without an FMS-PDU being transferred.

- The CP receives an FMS indication (S5 as responder) that is incorrect or cannot be executed. The CP 5431 FMS sends a response with the corresponding error number back to the requester.

The error number is made available to the user either via the test functions in the COM or at the requester end by transferring it to the PLC. When the error number is transferred to the PLC, it is written to the second word after the programmed or configured status word. It remains entered here until it is overwritten by the application, by a second error number or by a correct JOB 0H. An error number is transferred to the PLC in the following situations:

- When an error occurs in a job from the PLC.

- When an FMS response is received with ERRCLS and ERRCOD.

8.1.1 FMS Errors

Error Class / Code	Meaning
0x0100	current VFD status preventing job from being executed
0x0200 0x0201	error other than following application association error user process not obtainable
0x0300 0x0301 0x0302 0x0303	error other than following object definition errors object with required index/name does not exist specified object attributes inconsistently configured object index or name already exists
0x0400 0x0401	error other than following capacity exceeded error no memory available to execute job
0x0500 0x0501 0x0502 0x0503 0x0504 0x0505	error other than follwing service errors current object status preventing execution FMS-PDU too long current object restrictions preventing execution service has inconsistent parameters a parameter has an illegal value
0x0600 0x0601 0x0602 0x0603 0x0604 0x0605 0x0606 0x0607 0x0608 0x0609	error other than the following access errors object with undefined reference attribute access unsuccessful due to hardware fault client does not have adequate access rights physical address outside valid range object with inconsistent reference attribute object not defined for intended access object does not exist access with wrong data type or destination DB too short access with name not supported
0x0700 0x0701 0x0702 0x0703 0x0704 0x0705 0x0706	error other than following OL errors permitted name length exceeded permitted OL length exceeded object list is read-only permitted extension length exceeded length of single object description exceeded processing problem with OL
0x0800	error cannot be defined as one of the above errors

8.1.2 Errors Establishing an Application Association

Error Class / Code	Meaning
0x0f00	error other than following initiate errors
0x0f01	selected PDU length not adequate
0x0f02	required service not supported by responder
0x0f03	OL version of requester not compatible with that of responder
0x0f04	remote FMS user rejects applic. assoc. establishment
0x0f05	application association with same password already exists on responder
0x0f06	profile number of the requester not compatible

8.1.3 Abort Codes

Error Class / Code	Meaning
0x1000	application association abort by FMS user, e.g. due to timeout
0x1001	OL version not compatible
0x1002	application association with same password already exists
0x1003	profile of responder not supported
0x1004	Device status "Limited Services Permitted"
0x1100	FMS AAL entry absent
0x1101	illegal, incorrect or unknown service from user
0x1102	incorrect or uknown FMS-PDU received from LLI
0x1103	illegal LLI service primitive
0x1104	incorrect or unknown service primitive
0x1105	illegal PDU size
0x1106	service not allowed on this application association
0x1107	invoke ID of response cannot be assigned
0x1108	maximum number of confirmed services exceeded
0x1109	conflict in FMS application association status
0x110a	service of request and response do not match
0x110b	invoke ID of the request exists already
0x1200	context check between local and remote LLI negative
0x1201	illegal LLI-PDU during associate or abort
0x1202	illegal LLI-PDU in data transfer phase
0x1203	reception of unknown or incorrect LLI-PDU
0x1204	DTA-ACK-PDU received and SAC = 0
0x1205	maximum number of services exceeded
0x1206	unknown invoke Id
0x1207	priority violation
0x1208	local error on remote partner
0x1209	timeout during associate
0x120a	timeout of cyclic control timers
0x120b	timeout of idle receive time
0x120c	timeout during LSAP activation
0x120d	illegal FDL primitive during associate or abort
0x120e	illegal FDL primitive in the data transfer phase
0x120f	unknown FDL primitive
0x1210	unknown LLI primitive
0x1211	illegal LLI primitive during associate
0x1212	illegal LLI primitive in the data transfer phase
0x1213	AAL entry incorrect
0x1214	conflict in application association establishment status
0x1215	error during cyclic data transfer
0x1216	maximum number of parallel services exceeded

Abort Codes

Error Class / Code	Meaning
0x1217	AAL is loaded by FMA7, LLI disabled
0x1218	error in confirmation or indication mode
0x1219	illegal FMA 1/2 primitive received
0x121a	illegal service on a cyclic application association
0x121b	wrong FMS-PDU size on a cyclic application association
0x1300	error other than following errors after pos. remote ack.
0x1301	error on remote FDL / FMA 1/2 interface
0x1302	no resources available on remote partner for send data
0x1303	service not activated on remote SAP
0x130c	no resources for low priority data response
0x130d	no resources for high priority data response
0x1310	service not activated on local SAP
0x1311	no reaction from remote station
0x1312	local station not connected
0x1313	error in local FDL service
0x1314	no local resources available
0x1315	error in local request parameters

8.1.4 FMS Reject

Error Class / Code	Meaning
0x2000	error other than following causes of reject
0x2001	invoke ID of the confirmed service req. already exists
0x2002	maximum number of outstanding services client exceeded
0x2003	Service is not supported connection-oriented as client
0x2004	Service is not supported connectionless as client
0x2005	PDU length > maximum PDU length
0x2006	illegal or incorrect service primitive from FMS user

8.2 Protocol Implementation Conformance Statement (PICS)

The Protocol Implementation Conformance Statements (PICS) provides the user with further information about FMS implementation (range and complexity) of the CP 5431 FMS.

Using the PICS, the user can determine the following:

- which services are supported by the CP 5431 FMS
- with which degree of complexity the supported services are available..

The PICS consists of four parts:

- Part 1 provides information about the implementation and the system.
- Part 2 lists the services supported.
- Part 3 lists the parameters and options supported.
- Part 4 lists the local implementation values.

PICS Serial Number:1 PICS Part 1 Implementation in the system	Date Issued: 12/94
System Parameters	Detail
Implementation's Vendor Name	Siemens AG
Implementation's Model Name	CP 5431 FMS
Implementation's Revision Identifier	V _._ 1)
Vendor Name of FMS	Siemens AG
Controller Type of FMS	NEC 70325
Hardware Release of FMS	A _._ 2)
Software Release of FMS	V _._ 1)
Profile Number	
Calling FMS User (enter "Yes" or "No")	Yes
Called FMS User (enter "Yes" or "No")	Yes

1) to be read out with the COM
2) the release can be found on the type plate

PICS Part 2 Supported Services			
Service	Primitive		Supported
Initiate	.req, .con		Yes
Status	.req, .con		Yes
Identify	.req, .con		Yes
Get-OL	.req, .con		Yes
Get-OL (long form)	.req, .con	.ind,.res	No
Unsolicited-Status	.req	.ind	No
Initiate-Put-OL Put-PV Terminate-Put-OL	.req, .con .req, .con .req, .con	.ind, .res .ind, .res .ind, .res	No
Initiate-Download-Sequence Download-Segment Terminate-Download-Sequence	.req, .con .req, .con .req, .con	.ind, .res .ind, .res .ind, .res	No
Initiate-Upload-Sequence Upload-Segment Terminate-Upload-Sequence	.req, .con .req, .con .req, .con	.ind, .res .ind, .res .ind, .res	No
Request-Domain-Download	.req, .con	.ind, .res	No
Request-Domain-Upload	.req, .con	.ind, .res	No
Create-Program-Invocation Delete-Program-Invocation	.req, .con .req, .con	.ind, .res .ind, .res	No
Start Stop Resume Reset	.req, .con .req, .con .req, .con .req, .con	.ind, .res .ind, .res .ind, .res .ind, .res	No
Kill	.req, .con	.ind, .res	No
Read	.req, .con	.ind, .res	Yes
Write	.req, .con	.ind, .res	Yes
Read-With-Type	.req, .con	.ind, .res	No
Write-With-Type	.req, .con	.ind, .res	No
Phys-Read	.req, .con	.ind, .res	No

PICS Part 2 Supported Services			
Service	Primitive		Supported
Phys-Write	.req, .con	.ind, .res	No
Information-Report	.req	.ind	Yes
Information-Report-With-Type	.req	.ind	No
Define-Variable-List Delete-Variable-List	.req, .con .req, .con	.ind, .res .ind, .res	No
Event-Notification	.req	.ind	No
Event-Notification-With-Type	.req	.ind	No
Acknowledge-Event-Notification	.req, .con	.ind, .res	No
Alter-Event-Condition-Monitoring	.req, .con	.ind, .res	No

"Yes" is entered in every line when all service primitives are supported.

PICS Part 3	
FMS Parameters and Options	Detail
Addressing by names	No
Maximum length for names	-
Access-Protection Supported	Yes
Maximum length for Extension	0
Maximum length for Execution Arguments	0

PICS Part 4	
Local Implementation Values	Detail
Maximum length of FMS-PDU	241
Maximum number of Services Outstanding Calling	4
Maximum number of Services Outstanding Called	4
Syntax and semantics of the Execution Argument	-
Syntax and semantics of Extension	-

❑

NOTES

A Abbreviations

Abbreviations

A

ACI	Acyclic Control Interval
ALI	Application Layer Interface. FMS interface for applications
ANR	Job number (for handling blocks)
ANZW	Status word
AP	Automation protocol layers 5 to 7 of the ISO/OSI reference model
AS	Active star coupler
ASCII	American Standard Code of Information Interchange

B

B	Block
BCD	Binary coded decimal
BE	Block end

C

CC	Central controller
CI	Cyclic interface
CIM	Computer Integrated Manufacturing
COM	Abbreviation for programming software for SIMATIC S5 CPs

COR	Coordination module
CP	Communications Processor
CPU	Central Processing Unit
CSF	Control System Flowchart, graphical representation of automation tasks with symbols
CSMA/CD	Carrier sense multiple access with collision detect
CTS	Clear To Send
D	
DA	Destination Address
DB	Data block
DCE	Data Communication Equipment
DIN	Deutsches Institut für Normung (German Standards Institute)
DIR	Directory of data medium and files
DMA	Direct Memory Access
DOS	Operating system
DP	Distributed I/Os
DPR	Dual Port RAM
DTE	Data Terminal Equipment
DW	Data word (16 bits)
DX	Extended data block

E

EG/EU	Expansion unit
EIA	Electronic Industries Association
EPROM	Erasable Programmable Read Only Memory
ET 200	Electronic Terminal 200

F

F	Flag bit
FB	Function block
FD	Floppy Disk (data medium)
FD	Flag double word
FDDI	Fiber Distributed Data Interface
FDL FDL2	Fieldbus Data Link (subfunction of layer 2) Free layer 2 communications
FlexOs	Multitasking operating system
FMA	Fieldbus Management Layer
FMS	Fieldbus Message Specification (complying with PROFIBUS)
FO	Fibre Optic
FW	Flag word
FY	Flag byte

G

GO	Global Object
GP	Global I/Os
GPW	Global Peripheral Word
GPY	Global Peripheral Byte
GRAPH 5	Software package for planning and programming sequence controllers

H

HDB	Handling blocks
HSA	Highest Station Address

I

IB	Input byte
IEC	International Electronics Commission
IEEE	Institution of Electrical and Electronic Engineers
IP	Intelligent peripheral module
ISO	International Standardization Organization
IW	Input word

K

KOMI	Command interpreter

L

LAD	Ladder Diagram, graphical representation of the automation task with symbols of a circuit diagram
LAN	Local Area Network
LB	Link block
LED	Light Emitting Diode
LEN	Length of a block
LLC	Logical Link Control
LLI	Lower Layer Interface
LSB	Least Significant Bit

M

MAC	Medium Access Control
MAP	Manufacturing Automation Protocol
MMS	Manufacturing Message Specification

N

NCM	Network and Communication Management

O

OB	Organization block
OSI	Open System Interconnection
OW	Word from the extended I/Os
OY	Byte from the extended I/Os

P

PAFE	Parameter assignment error
PB	Program block
PC	Personal Computer
PCI	Protocol Control Information (for coordinating a protocol)
PCP/M-86	Operating system Personal CP/M-86
PDU	Protocol Data Unit (frames consisting of PCI and SDU)
PG	Programmer
PI	Program invocation
PI	Process image
PII	Process image of the inputs
PIQ	Process image of the outputs
PLC	Programmable controller
PNO	PROFIBUS user organization
PRIO	Priority
PROFIBUS	PROcess Field BUS
PW	Peripheral word
PY	Peripheral byte

Q

QB	Output byte
QW	Output word

R

RAM	Random Access Memory
RLO	Result of logic operation (code bits)
RS	Recommended Standard
RS 485	EIA standard (multipoint capability) standard for electrical data transmission

S

S5-S5	Special type of communication PLC with PLC
SA	Source Address
SAP	Service Access Point. Logical interface points on the interface between the layers via which the PDUs are exchanged between service users.
SB	Sequence block
SDA	Send Data with Acknowledge
SDN	Send Data with No Acknowledge
SDU	Service Data Unit. Information about the service used and the user data contained within it.
SINEC	Siemens network architecture for coordination and engineering
SINEC NCM	Network management for LANs
SINEC TF	SINEC technological functions
SRD	Send and Request Data
SSNR	Interface number

STEP 5	Programming language for programming programmable controllers of the SIMATIC S5 range
STL	Statement List, STEP 5 method of representation as a series of mnemonics of PLC commands (complying with DIN 19239)
Sub-D	Subminiature D (connector)
SYM	Symbolic addressing
SYSID	Block for system identification
S5-KOMI	S5 command interpreter
S5-DOS/MT	S5 operating system based on FlexOS

T

TF	Technological functions
TSAP	Transport Service Access Point
TSAP-ID	Transport Service Access Point Identifier
TSET	Set-up time
TSDR	Station delay
TSL	Slot-time
TTR	Target rotation time
TPDU	Transport Protocol Data Unit (size of the block of data transferred by the transport system)
TSDU	Transport Service Data Unit (size of the block of data transferred to the transport system with a job for transportation via a transport relation)

TSEL	Transport selector, term used as an alternative for TSAP-ID

V

VB	Code for application association-specific and abbreviation (code) for data link block.
VFD	Virtual Field Device
VMD	Virtual Manufacturing Device

NOTES

B Index

D

E

F

G

H

I

J

L

M

O

P

R

S

T

U

V

C Further Reading

/1/ N.N.:
PROFIBUS Standard DIN 19245, Part 1
Beuth-Verlag Berlin 1988

/2/ N.N.:
PROFIBUS Standard DIN 19245 Part 2
Beuth-Verlag, Berlin, 1990

/3/ Klaus Bender:
PROFIBUS Der Feldbus für die Automation
Hanser Verlag, München Wien, 1990
ISBN 3-446-16170-8

NOTES