SIEMENS

SIMATIC S5

COM 525
Programming Package for the Communications Processors CP 524 and CP 525 (S5-DOS)

Manual

Volume 1/2

Order No. 6ES5 998-1DB21
Release 08

We have checked the contents of this manual for agreement with the hardware and software 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 welcomed.

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 model or design, are reserved.

Copyright © Siemens AG 1991 All rights reserved

Siemens Aktiengesellschaft

6ES5 998-1DB21
Elektronikwerk-Karlsruhe
Printed in the Federal Republic of Germany

Contents		
Warning Information Suggestions/Corrections	C79000-R8576-C499	
Volume 1		
		1
		2
How to Use the Manual	C79000-D8576-C499-03	3
CP 525-2 Communications Processor Instructions	C79000-B8576-C537-07	4
CP 524 Communications Processor Instructions	C79000-B8576-C538-06	5
–		6
Computer Link with RK 512 User's Guide	C79000-B8576-C539-08	7
Event Output and Listing with the PT88/PT89 Printer User's Guide	C79000-B8576-C541-05	8
		9
		10
Volume 2		

SIEMENS

Warnhinweis

Gefahren beim Einsatz sogenannter SIMATIC-kompatibler Baugruppen fremder Hersteller

"Den Hersteller eines Produktes (hier SIMATIC) trifft die Produktbeobachtungspflicht, d. h. er muß generell vor Gefahren des Produktes warnen. Diese Produktbeobachtungspflicht wurde von der neueren Rechtssprechung auch auf fremde Zubehörteile erstreckt. Der Hersteller hat danach die Verpflichtung, auch solche Gefahren zu beobachten und zu erkennen, die aus der Verbindung des Produktes mit Produkten anderer Hersteller entstehen.

Aus diesem Anlaß sehen wir uns verpflichtet, unsere Kunden, die SIMATIC-Produkte einsetzen, zu warnen, sogenannte SIMATIC-kompatible Baugruppen fremder Hersteller als Ersatz- oder Zusatzbaugruppen in das Automatisierungssystem SIMATIC einzusetzen.

Unsere Produkte werden einer anspruchsvollen Qualtiätssicherung unterworfen. Uns ist nicht bekannt, ob die fremden Hersteller sogenannter SIMATIC-kompatibler Baugruppen überhaupt oder eine annähernd gleichwertige Qualitätssicherung durchführen. Diese sogenannten SIMATIC-kompatiblen Baugruppen kommen nicht im Einvernehmen mit uns auf den Markt; es gibt **keine** Empfehlung der Siemens AG, sogenannte SIMATIC-kompatible Baugruppen fremder Hersteller einzusetzen. Die Werbung der fremden Hersteller sogenannter SIMATIC-kompatibler Baugruppen erweckt irrtümlich den Eindruck, als sei der Inhalt der Werbung in Fachzeitschriften, Katalogen oder Ausstellungen mit uns abgesprochen. Werden sogenannte SIMATIC-kompatible Baugruppen fremder Hersteller mit unserem SIMATIC-Automatisierungssystem verbunden, handelt es sich um einen empfehlungswidrigen Gebrauch unseres Produkts. Wegen der universellen Vielfalt der Einsatzmöglichkeiten unserer SIMATIC-Automatisierungssysteme und der hohen Zahl der weltweit vermarkteten Produkte, können wir die konkrete Gefahrenanalyse durch diese sogenannten SIMATIC-kompatiblen Baugruppen nicht konkret beschreiben.

Es geht über die tatsächlichen Möglichkeiten des Herstellers hinaus, alle diese sogenannten SIMATIC-kompatiblen Baugruppen in ihrer Wirkung auf unser SIMATIC-Produkt überprüfen zu lassen. Treten Mängel bei der Verwendung von sogenannten SIMATIC-kompatiblen Baugruppen in einem SIMATIC-Automatisierungssystem auf, werden wir für solche Systeme jede Gewährleistung ablehnen.

Im Fall von Produkthaftpflichtschäden verursacht durch den Einsatz von sogenannten SIMATIC-kompatiblen Baugruppen sind wir nicht haftbar, da wir die Anwender rechtzeitig vor den potentiellen Gefahren der Benutzung sogenannter SIMATIC-kompatibler Baugruppen gewarnt haben."

Warning

Risks involved in the use of so-called SIMATIC-compatible modules of non-Siemens manufacture

"The manufacturer of a product (SIMATIC in this case) is under the general obligation to give warning of possible risks attached to his product. This obligation has been extended in recent court rulings to include parts supplied by other vendors. Accordingly, the manufacturer is obliged to observe and recognize such hazards as may arise when a product is combined with products of other manufacture.

For this reason, we feel obliged to warn our customers who use SIMATIC products not to install so-called SIMATIC-compatible modules of other manufacture in the form of replacement or add-on modules in SIMATIC systems.

Our products undergo a strict quality assurance procedure. We have no knowledge as to whether outside manufacturers of so-called SIMATIC-compatible modules have any quality assurance at all or one that is nearly equivalent to ours. These so-called SIMATIC- compatible modules are not marketed in agreement with Siemens; we have never recommended the use of so-called SIMATIC-compatible modules of other manufacture. The advertising of these other manufacturers for so-called SIMATIC-compatible modules wrongly creates the impression that the subject advertised in periodicals, catalogues or at exhibitions had been agreed with us. Where so-called SIMATIC-compatible modules of non-Siemens manufacture are combined with our SIMATIC automation systems, we have a case of our product being used contrary to recommendations. Because of the variety of applications of our SIMATIC automation systems and the large number of these products marketed worldwide, we cannot give a concrete description specifically analyzing the hazards created by these so-called SIMATIC-compatible modules.

It is beyond the manufacturer's capabilities to have all these so-called SIMATIC-compatible modules checked for their effect on our SIMATIC products. If the use of so-called SIMATIC-compatible modules leads to defects in a SIMATIC automation system, no warranty for such systems will be given by Siemens.

In the event of product liability damages due to the use of so-called SIMATIC-compatible modules, Siemens are not liable since we took timely action in warning users of the potential hazards involved in so-called SIMATIC-compatible modules."

Avertissement

Risques liés à l'utilisation de modules de constructeurs tiers commercialisés sous la désignation de "modules compatibles SIMATIC"

≪Le constructeur d'un produit (dans le cas présent SIMATIC) a l'obligation d'observer le produit, c'est-à-dire qu'il est obligé, d'une manière générale, d'attirer l'attention sur les dangers inhérents au produit. Ces derniers temps, la jurisprudence a étendu cette obligation d'observation du produit aux éléments accessoires issus de constructeurs tiers. En foi de quoi, le constructeur a aussi l'obligation d'observer son produit pour déceler les dangers susceptibles de survenir dans le cadre de l'association de son produit avec des produits de constructeurs tiers.

Pour cette raison, nous nous voyons obligés d'attirer l'attention de nos clients, utilisateurs de produits SIMATIC, sur les risques liés à l'utilisation de "modules compatibles SIMATIC" de constructeurs tiers à titre de modules de remplacement ou de complément dans les produits de notre système d'automatisation SIMATIC.

Nos produits font l'objet d'une assurance qualité très poussée. Il nous est impossible de savoir si les constructeurs tiers de "modules compatibles SIMATIC" mettent en œuvre un système qualité et, dans l'affirmative, si leurs dispositions d'assurance qualité permettent d'obtenir le niveau de qualité requis. Les "modules compatibles SIMATIC"ne sont pas commercialisés avec notre consentement : Siemens AG n'a émis **aucune** recommandation concernant l'utilisation de "modules compatibles SIMATIC" de constructeurs tiers. La publicité des constructeurs tiers de "modules compatibles SIMATIC" laisse penser à tort que les textes publicitaires dans les revues, les catalogues ou les expositions ont été convenus avec nous. L'utilisation conjointe de "modules compatibles SIMATIC" de constructeurs tiers et de produits de notre système d'automatisation SIMATIC constitue un cas d'utilisation de nos produits qui est contraire à nos recommandations. Considérant la grande diversité d'emploi de notre système d'automatisation SIMATIC ainsi que l'importance du parc mondial des produits installés, il nous est impossible de donner une description concrète de l'analyse des risques liés à l'emploi des "modules compatibles SIMATIC".

Nous n'avons pas la possibilité matérielle de procéder au contrôle de l'interaction de notre produit SIMATIC avec les "modules compatibles SIMATIC" de constructeurs tiers. Nous rejetons tout appel en garantie pour les vices survenant dans un système d'automatisation SIMATIC mettant aussi en œuvre des "modules compatibles SIMATIC" de constructeurs tiers.

Nous déclinons toute responsabilité pour les sinistres relevant de la Responsabilité Civile Produits, étant donné que nous avons attiré à temps l'attention des utilisateurs sur les risques potentiels inhérents à l'utilisation de "modules compatibles SIMATIC" de constructeurs tiers. ≫

Safety-Related Guidelines for the User

1 General

This manual provides the information required for the intended use of the particular product. The documentation is written for technically qualified personnel such as engineers, programmers or maintenance specialists who have been specially trained and who have the specialized knowledge required in the field of instrumentation and control.

A knowledge of the safety instructions and warnings contained in this manual and their appropriate application are prerequisites for safe installation and commissioning as well as safety in operation and maintenance of the product described. Only qualified personnel as defined in section 2 have the specialized knowledge that is necessary to correctly interpret the general guidelines relating to the safety instructions and warnings and implement them in each particular case.

This manual is an inherent part of the scope of supply even if, for logistic reasons, it has to be ordered separately. For the sake of clarity, not all details of all versions of the product are described in the documentation, nor can it cover all conceivable cases regarding installation, operation and maintenance. Should you require further information or face special problems that have not been dealt with in sufficient detail in this documentation, please contact your local Siemens office.

We would also 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.

2 Qualified Personnel

Persons who are **not qualified** should not be allowed to handle the equipment/system. Non-compliance with the warnings contained in this manual or appearing on the equipment itself can result in severe personal injury or damage to property. Only **qualified personnel** should be allowed to work on this equipment/system.

© Siemens AG 1991 C79000-D8576-C336-01

Qualified persons as referred to in the safety guidelines in this manual as well as on the product itself are defined as follows:

- System planning and design engineers who are familiar with the safety concepts of automation equipment;
- Operating personnel who have been trained to work with automation equipment and are conversant with the contents of the manual in as far as it is connected with the actual operation of the plant;
- Commissioning and service personnel who are trained to repair such automation equipment and who are authorized to energize, deenergize, clear, ground and tag circuits, equipment and systems in accordance with established safety practices.

3 Danger Notices

The notices and guidelines that follow are intended to ensure personal safety, as well as protecting the product and connected equipment against damage.

The safety notices and warnings for protection against loss of life (the users or service personnel) or for protection against damage to property are highlighted in this manual by the terms and pictograms defined here. The terms used in this manual and marked on the equipment itself have the following significance:

Danger

indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.

Warning

indicates that death, severe personal injury or substantial property damage can result if proper precautions are not taken.

Caution

indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note

is an important information about the product, its operation or a part of the manual to which special attention is drawn.

© Siemens AG 1991 C79000-D8576-C338-01

4 Proper Usage

- The equipment/system or the system components may only be used for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers as far as this is recommended or permitted by Siemens.
- The product described has been developed, manufactured, tested and the documentation compiled in keeping with the relevant safety standards. Consequently, if the described handling instructions and safety guidelines described for planning, installation, proper operation and maintenance are adhered to, the product, under normal conditions, will not be a source of danger to property or life.



Warning

- After opening the housing or the protective cover or after opening the system cabinet, certain parts of this equipment/system will be accessible, which could have a dangerously high voltage level.
- Only suitably qualified personnel should be allowed access to this equipment/system.
- These persons must be fully conversant with any potential sources of danger and maintenance measures as set out in this manual.
- It is assumed that this product be transported, stored and installed as intended, and maintained and operated with care to ensure that the product functions correctly and safely.

© Siemens AG 1991 C79000-D8576-C336-01

5 Procedures for Maintenance and Repair

If measurement or testing work is to be carried out on an active unit, the rules and regulations contained in the "VBG 4.0 Accident prevention regulations" of the German employers liability assurance association (Berufsgenossenschaften) must be observed. Particular attention is drawn to paragraph 8 "Permissible exceptions when working on live parts". Use only suitable electrical tools.

Warning

- Repairs to an item of automation equipment may only be carried out by **Siemens service personnel** or an **authorized Siemens repair center**. For replacement purposes, use only parts or components that are contained in the spare parts list or listed in the "Spare parts" section of this manual. Unauthorized opening of equipment and improper repairs can result in loss of life or severe personal injury as well as substantial property damage
- Before opening the equipment, always remove the power plug or open the disconnecting switch.
- Only use the fuse types specified in the technical specifications or the maintenance instructions of this manual.
- Do not throw batteries into an open fire and do not carry out any soldering work on batteries (danger of explosion). Maximum ambient temperature 100°C. Lithium batteries or batteries containing mercury should not be opened or recharged. Make sure that the same type is used when replacing batteries.
- Batteries and accumulators must be disposed of as classified waste.
- The following points require attention when using monitors:
  Improper handling, especially the readjustment of the high voltage or fitting of another tube type can result in excessive X-ray radiation from the unit. The license to operate such a modified unit automatically lapses and the unit must not be operated at all.

The information in this manual is checked regularly for updating and correctness and may be modified without prior notice. The information contained in this manual is protected by copyright. Photocopying and translation into other languages is not permitted without express permission from Siemens.

© Siemens AG 1991 C79000-D8576-C336-01

Guidelines for Handling Electrostatically Sensitive Devices (ESD)

1 What is ESD?

VSLI chips (MOS technology) are used in practically all SIMATIC S5 and TELEPERM M modules. These VLSI components are, by their nature, very sensitive to overvoltages and thus to electrostatic discharge:

They are therefore defined as
"Electrostatically Sensitive Devices"

"ESD" is the abbreviation used internationally.

The following warning label on the cabinets, subracks and packing indicates that electrostatically sensitive components have been used and that the modules concerned are susceptible to touch:



ESDs can be destroyed by voltage and energy levels which are far below the level perceptible to human beings. Such voltages already occur when a component or a module is touched by a person who has not been electrostatically discharged. Components which have been subjected to such overvoltages cannot, in most cases, be immediately detected as faulty; the fault occurs only after a long period in operation.

An electrostatic discharge
- of 3500 V can be felt
- of 4500 V can be heard
- must take place at a minimum of 5000 V to be seen.

But just a fraction of this voltage can already damage or destroy an electronic component.

The typical data of a component can suffer due to damage, overstressing or weakening caused by electrostatic discharge; this can result in temporary fault behavior, e.g. in the case of

- temperature variations,
- mechanical shocks,
- vibrations,
- change of load.

Only the consequent use of protective equipment and careful observance of the precautions for handling such components can effectively prevent functional disturbances and failures of ESD modules.

2 When is a Static Charge Formed?

One can never be sure that the human body or the material and tools which one is using are not electrostatically charged.

Small charges of 100 V are very common; these can, however, very quickly rise up to 35 000 V.

Examples of static charge:

- Walking on a carpet	up to 35 000 V
- Walking on a PVC flooring	up to 12 000 V
- Sitting on a cushioned chair	up to 18 000 V
- Plastic desoldering unit	up to 8 000 V
- Plastic coffee cup	up to 5 000 V
- Plastic bags	up to 5 000 V
- Books, etc. with a plastic binding	up to 8 000 V

3 Important Protective Measures against Static Charge

- Most plastic materials are highly susceptible to static charge and must therefore be kept as far away as possible from ESDs.
- Personnel who handle ESDs, the work table and the packing must all be carefully grounded.

© Siemens AG C79000-D8076-C339-01

4 Handling of ESD Modules

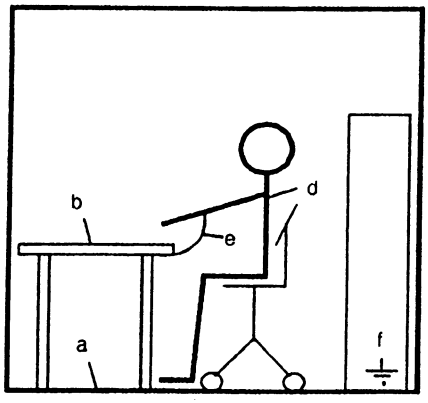
- One basic rule to be observed is that electronic modules should be touched by hand only if this is necessary for any work to be done on them. Do not touch the component pins or the conductors.
- Touch components only if
  - the person is grounded at all times by means of a wrist strap

  or
  - the person is wearing special anti-static shoes or shoes with a grounding strip.
- Before touching an electronic module, the person concerned must ensure that (s)he is not carrying any static charge. The simplest way is to touch a conductive, grounded item of equipment (e.g. a blank metallic cabinet part, water pipe, etc.) before touching the module.
- Modules should not be brought into contact with insulating materials or materials which take up a static charge, e.g. plastic foil, insulating table tops, synthetic clothing, etc.
- Modules should only be placed on conductive surfaces (table with anti-static table top, conductive foam material, anti-static plastic bag, anti-static transport container).
- Modules should not be placed in the vicinity of monitors, TV sets (minimum distance from screen > 10 cm).

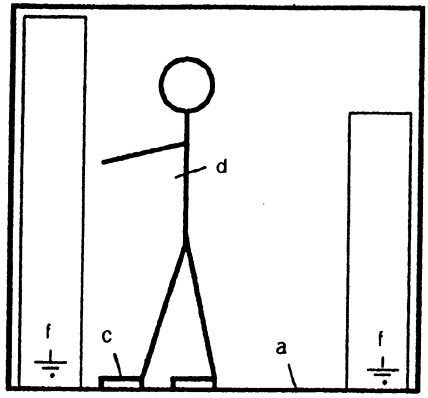
The diagram on the next page shows the required protective measures against electrostatic discharge.

5 Measurements and Modification to ESD Modules

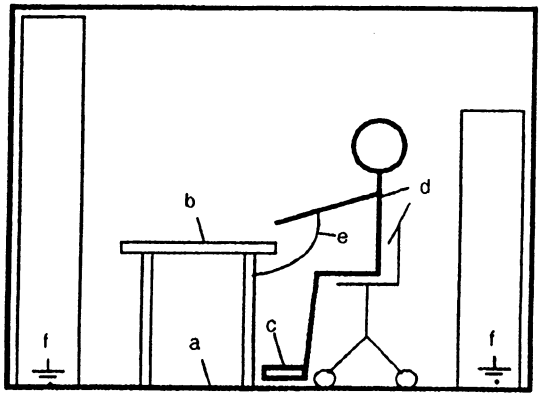
- Measurements on modules may only be carried out under the following conditions:
  - the measuring equipment is grounded (e.g. via the PE conductor of the power supply system) or
  - when electrically isolated measuring equipment is used, the probe must be discharged (e.g. by touching the metallic casing of the equipment) before beginning measurements.
- Only grounded soldering irons may be used.



Sitting position



Standing position



Standing/sitting position

a Conductive flooring
b Anti-static table
c Anti-static shoes
d Anti-static coat
e Grounding wrist strap
f Grounding connection of the cabinets

6 Shipping of ESD Modules

Anti-static packing material must always be used for modules and components, e.g. metalized plastic boxes, metal boxes, etc. for storing and dispatch of modules and components.

If the container itself is not conductive, the modules must be wrapped in a conductive material such as conductive foam, anti-static plastic bag, aluminium foil or paper. Normal plastic bags or foils should not be used under any circumstances.

For modules with built-in batteries ensure that the conductive packing does not touch or short-circuit the battery connections; if necessary cover the connections with insulating tape or material.

© Siemens AG C79000-D8076-C339-01

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please do not forget to state the title, order number and release of your manual.

Title of Your Manual: .....
Order No. of Your Manual: ..... Release: .....

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Siemens AG
AUT E 1163

D-76181 Karlsruhe
Federal Republic of Germany

From:

Your Name: . . . . . .

Your Title: . . . . . .

Company Name: . . . . . .

Street: . . . . . .

City, Zip Code: . . . . . .

Phone: . . . . . .

Please check any industry that applies to you:

- ☐ Automotive
- ☐ Chemical
- ☐ Electrical Machinery
- ☐ Food
- ☐ Instrument and Control
- ☐ Nonelectrical Machinery
- ☐ Pharmaceutical
- ☐ Plastic
- ☐ Pulp and Paper
- ☐ Textiles
- ☐ Transportation
- ☐ Petrochemical

Other . . . . . .

C79000-V8576-C066-01

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please do not forget to state the title, order number and release of your manual.

Title of Your Manual: .....

Order No. of Your Manual: ..... Release: .....

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Siemens AG
AUT E 1163

D-76181 Karlsruhe
Federal Republic of Germany

From:

Your Name: . . . . . .

Your Title: . . . . . .

Company Name: . . . . . .

Street: . . . . . .

City, Zip Code: . . . . . .

Phone: . . . . . .

Please check any industry that applies to you:

- ☐ Automotive
- ☐ Chemical
- ☐ Electrical Machinery
- ☐ Food
- ☐ Instrument and Control
- ☐ Nonelectrical Machinery
- ☐ Pharmaceutical
- ☐ Plastic
- ☐ Pulp and Paper
- ☐ Textiles
- ☐ Transportation
- ☐ Petrochemical

Other . . . . . .

C79000-V8576-C066-01

SIEMENS

SIMATIC S5

How to Use the Manual

C79000-D8576-C499-03

Contents **Page**

1	Things to Remember	3
2	Where Do I Find This in the Manual?	5
3	Abbreviations	9
4	Ordering Data	12
5	Further Relevant Documentation	14

1 Things to Remember

This manual is intended to help you use the **CP 525-2 and CP 524** communications processors as follows:

- for the "Computer link with RK 512" and
- for "Event output and logging with the PT88/PT89 printer"

It is also intended to help you program the associated CP 525 user program with the programming package COM 525.

The manual is divided into **two** parts.

CP 525-2

The CP 525-2 is often abbreviated to CP 525.

CP 524

The CP 524 can be used just as the CP 525 when you wish to communicate with only **one** partner.

The CP 524 has only **one** serial device interface; it has **no PG interface**. Therefore only EPROM submodules that have been programmed on the PG can be used on the CP 524.

The serial device interface of the CP 524 corresponds to the device interface IF 1 on the CP 525. An EPROM submodule programmed for IF 1 can be used both on the CP 525 and CP 524.

The CP 524 is not referred to in the parts of this manual; however, almost all the information referring to IF 1 on the CP 525 also applies to the CP 524, except the CP 525 functions available when you are using the PG online (e.g. start, stop, transfer).

You should first develop your user program using a CP 525. You can then connect the PG online and transfer your CP 525 user program directly to a RAM submodule plugged into the CP 525. Then you can carry out the start-up. Any corrections which may be needed can then be carried out quickly. Once you have completed these corrections, you can then program the EPROM for use with the CP 524.

!IMPORTANT!

The CP 525 and CP 524 **cannot** be used in association with the programmable controller **S5-135U/S processor** (CPU 921).

We wish you every success with your communications processor!

2 Where Do I Find This in the Manual?

This manual

COM 525 programming package for the communications processors CP 524 and CP 525 (S5-DOS)

is divided into **two** parts. Both parts contain a table of contents of the whole manual.

Part 1

Section 1 **This section is not used**

Section 2 **This section is not used**

Section 3 **Instructions on using the manual**

These are general instructions that you should bear in mind; this section also includes
- abbreviations and their explanations
- ordering data
- further relevant documentation

Section 4 **Communications processor CP 525-2** Instructions

This is an introductory description; it contains important information about installing and running the module.

Section 5 **Communications processor CP 524**
Instructions

This is an introductory description; it contains important information about installing and running the module.

Section 6 **This section is not used**

Section 7 **Computer link with RK 512**
User's guide

This guide provides the following:
- basic facts
- important, general information about programming
- help in correcting errors in the computer link component

Section 8 **Event output and logging with the PT88/PT89 printer**
User's guide

This guide provides you with the following:
- basic facts
- important, general information about programming
- help in correcting errors in the logging printer

Part 2

Section 1 **Computer link with RK 512**
Example of application

This is an introduction. When you have worked through this example you will have generated a CP 525 user program in which all the important functions of the computer link are implemented. This section concerns programming the STEP 5 user program for the CPU.

Section 2 **Event output and logging with the PT88/PT89 printer**
Example of application

This is an introduction. When you have worked through this example you will have generated a CP 525 user program in which all the important functions of the logging printer are implemented. This section also concerns programming the STEP 5 user program for the CPU.

Section 3 **Notes on the operating systems PCP/M-86 and S5-DOS**
User's guide

This is information you should be familiar with before you install the COM 525 programming package.

Section 4 **Programming package COM 525**
User's guide

This is an introductory guide for the programming package COM 525 as follows:
- contains all the screen masks output by COM 525
- includes instructions about the entries to be made in the fields of the screen masks.

Section 5 **COM 525 messages**
Reference manual

This is a description of the error messages which are displayed on the screen of your programmer by the COM 525 programming package.

Section 6 **PROM 525**
User's guide

This is an introduction to programming EPROMs as follows:
- includes all the screen masks output by PROM 525
- contains information about the entries to be made in the fields of the screen masks.

Section 7 **Using the handling blocks**
Reference manual

This reference manual provides an overview of the functions provided by the handling blocks, with particular emphasis on the computer link or the logging printer.

3 Abbreviations

A-NR	Job number
ANZW	Condition codeword
AS 512C	Interface module 512C
B	Bell
BLGR	Field length
C	Condensed print
CF	Coordination flag
CL	Chained log
CL	Computer link
CML	Current message log
COM 525	Programming package COM 525 for the communications processors CP 525 and CP 524
CP	Communications processor
CPU	Central processing unit
CR	Carriage return
DB	Data block
DBNR	Data block number
DL	Left-hand data byte in a data word
DR	Right-hand data byte in a data word
DW	Data word
DX	Expanded data block
E	Expanded print
FB	Function block
FD	Disk drive
FIFO	First in first out buffer
FW	Flag word
FY	Flag byte
GIB	Group inhibit bit
H	Heavy print
HDB	(data) Handling block
IF	Device interface
JB	Jumper block

LAUFPT88	Procedure LAUFPT88 (logging printer PT88/PT89 function)
MEP adapter	Module-EPROM-programming adapter
OB	Organization block
PAFE	Condition code byte for parameter assignment errors
PB	Program block
PC	Programmable controller
PG	Programmer
PSEUDO R/W function	PSEUDO READ/WRITE function
PSL	Process status log
PT88/PT89	PT88/PT89 printer
QANF	Start address of source
QLAE	Source length = number of pieces of source data
QTYP	Source type = type of data source
QVZ	Timeout
REPTEL	Reply telegram
RS422-A	EIA standard RS422-A
RS485	EIA standard RS485
R/W function	READ/WRITE function
S5-DOS	Operating system
S5 program	STEP 5 program
SML	Sequential message log
SSNR	Interface number
SYSID	Identification area SYSID
SYSTAT	Error message area of the SYSTAT
TTY	Current loop signals (20 mA)
U	Underlined
UPDATE SML	Update sequential message log

V.24	V.24 signals
ZANF	Start address of destination
ZLAE	Destination length = number of pieces of destination data
ZTYP	Destination type = type of data destination
ZVZ	Character timeout

4 Ordering Data

Memory submodules for the CP 525-2 and CP 524:

EPROM submodules	Order number
16 Kwords	6ES5 373-1AA41
32 Kwords	6ES5 373-1AA61
64 Kwords	6ES5 373-1AA81

Memory submodules for CP 525-2:

RAM submodules	Order number
16 Kwords	6ES5 377-0AB21
32 Kwords	6ES5 377-0AB31
64 Kwords	6ES5 377-0AB41
64 Kwords, 377 RAM submodule, buffered	6ES5 377-0BB41

Interface modules for the CP 524:

V.24	6ES5 752-0AA22
TTY (20 mA)	6ES5 752-0AA12
RS422-A/485	6ES5 752-0AA42

Installation guidelines for SIMATIC compact devices:	C79000-B8576-C252
Signalling functions for standard CP's:	6ES5 848-7WA01 (7=MS-DOS) 6ES5 842-8WA01 (8=PCP/M)
Compact signalling functions:	6ES5 848-7WE01 (7=MS-DOS) 6ES5 848-8WE01 (8=PCP/M)
Programming package COM PMC:	6ES5 895-4SF21 (9=5.25") 6ES5 835-4SF21 (3=3.5")

Handling blocks:	Order number
S5-135U / R processor	6ES5 842-xCBx0(-yy)
S5-150U	6ES5 844-xCAx0(-yy)
S5-155U	6ES5 846-xCA01

Please enquire at your nearest SIEMENS branch or the nearest technical office about the numbers or letters marked x or (-yy).

Standard cable connectors	Order number
CP 525 - PG 6xx (TTY)	6ES5 726-0xxx0
CP 525/524 - CP 525/524 (TTY)	6ES5 726-1xxx0
CP 525/524 - PROMEA M (TTY)	6ES5 725-8xxx0
CP 525/524 - AS 512C (TTY)	6ES5 726-2xxx0
CP 525/524 - PROMEA R (TTY)	6ES5 726-3xxx0
CP 525/524 - PT88/PT89 (TTY/V.24)	6ES5 726-5xxx0
CP 525/524 - Modem N10 (V.24)	6ES5 726-7xxx0
CP 525/524 - CP 525/524 (V.24)	6ES5 726-8xxx0
CP 525/524 - PROMEA R (V.24)	6ES5 726-4xxx0
CP 524 - CP 524 (V.11/RS422-A)	6ES5 725-7xxx0
CP 525 IF2 - PG 7xx	6ES5 734-5xxx0

(xxx = SIMATIC length key)

5 Further Relevant Documentation

/1/ See the ordering data for the handling blocks

/2/ PT88/PT89 printer - description

SIEMENS

Siemens Aktiengesellschaft

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 model or design, are reserved.

© Siemens AG 1988

Order No.: C79000-B8576-C541
Order from: Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany
AG 0689 0,5 AB 198 En

SIEMENS

SIMATIC S5

Communications Processor CP 525-2

Instructions C79000-B8576-C537-07

Contents

		Page
1	**Technical Description**	**1**
1.1	Application	1
1.2	Mechanical Design	4
1.3	Serial Device Interface	5
1.4	Dual-port RAM, Interface to the CPU	6
1.5	Memory Submodules	12
1.6	Hardware Clock	14
1.7	Technical Data	15
2	**Installation**	**16**
2.1	Slots in the Programmable Controller	16
2.2	Plugging in and Removing the CP 525-2	17
2.3	Installation Guidelines	17
3	**Operation**	**18**
3.1	Controls and Displays	18
3.2	Operating Modes	19
3.3	Status Displays	19
3.4	Jumper Settings	20
3.4.1	Module Address/Interface Number	21
3.4.2	Enabling/Disabling IPC Flags	23
3.4.3	Settings of the Device Interfaces	26
3.4.4	Permanent Jumper Settings	28
4	**Setting up the Hardware**	**29**
5	**Pin Assignments**	**30**
5.1	Backplane Connector 1	30
5.2	Backplane Connector 2	31
5.3	Memory Submodule Interface	32
5.4	Device Interface 1 (IF 1)	33
5.5	Device Interface 2 (IF 2)	34

4

6 Standard Cable Connectors 35

6.1 Connection CP 525 - PG 6xx (TTY) 35
6.2 Connection CP 525/524 - CP 525/524 (TTY) 36
6.3 Connection CP 525/524 - PROMEA M (TTY) 36
6.4 Connection CP 525/524 - AS 512C (TTY) 37
6.5 Connection CP 525/524 - PROMEA R (TTY) 38
6.6 Connection CP 525/524 - PT88/PT89 (TTY/V.24) 39
6.7 Connection CP 525/524 - Modem N10 (V.24) 40
6.8 Connection CP 525/524 - CP 525/524 (V.24) 40
6.9 Connection CP 525/524 - PROMEA R (V.24) 41
6.10 Connection CP 525,IF2 - PG 7xx (TTY) 42

1 Technical Description

1.1 Application

The **communications processor CP 525-2** can be used in the programmable controllers of the SIMATIC 5 U range (S5-115U, S5-135U, S5-150U and S5-155U). **Note that the CP 525-2 cannot be used in conjunction with the S5-135U / S processor.**

Using the communications processor CP 525-2 and the accompanying programming package COM 525 you can accomplish the following:

- **log process states and messages originating in the process** and/or
- **link your programmable controller with other programmable controllers or computers**

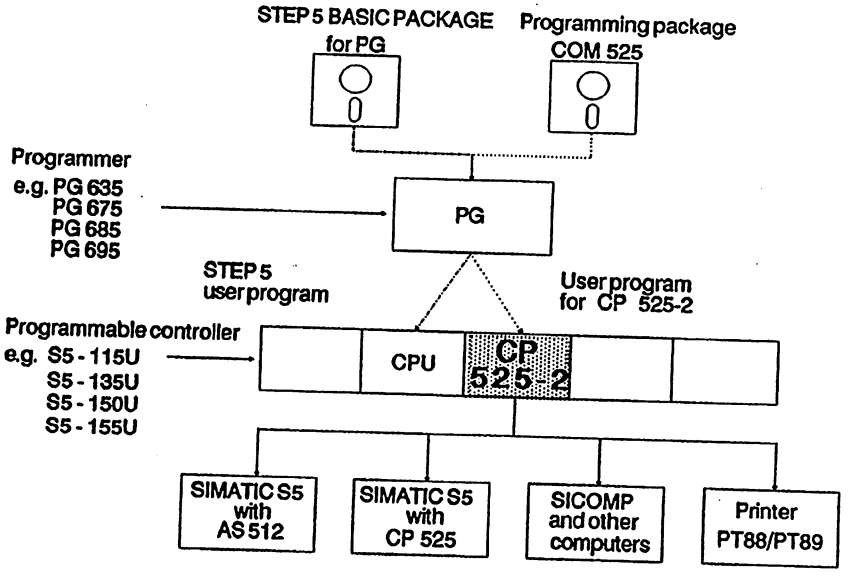
By connecting the printer PT88/PT89, you can log process states and messages. The communications processor CP 525-2 initiates the compilation of message texts and their output on the printer.

If the programmable controller (PC) is part of an automation network, it can exchange data with one or more programmable controllers or with a central process computer (computer link). These tasks are also carried out independently by the CP 525-2.

The data exchange between the CP 525-2 and the central processor (CPU) of the programmable controller is restricted to a minimum. Only the variable data (current process values) is exchanged between the CPU and the CP 525-2. The CP 525-2 has its own memory for invariable data, which you program using the programming package COM 525.

CP 525-2:

- listing process statuses and messages originating in the process
- linking programmable controllers with each other or with computers



The communications processor CP 525-2 can be plugged into the S5 programmable controllers of the U range. By using a programmer (PG) and the programming package COM 525, you can write your user program for the CP 525-2 (e.g. message texts to be output on the PT88/PT89 printer). Once written, this user program is stored either in an EPROM submodule or in a RAM submodule.

If you decide to store it in the RAM submodule, connect the PG to device interface 2 of the CP 525-2 and transfer the user program for the CP 525-2 to the RAM submodule. This submodule must already be plugged into the memory submodule receptacle of the CP 525-2.

You also program your STEP 5 user program for the CPU on the PG using the STEP 5 BASIC PACKAGE.

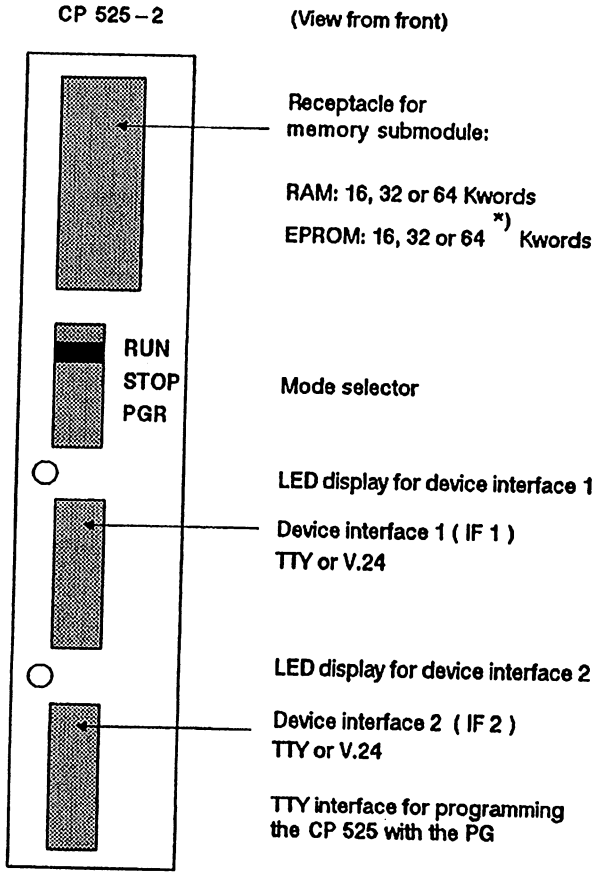
The following devices can be connected directly to the CP 525-2:

- other programmable controllers
- central computer
- the PT88/PT89 printer

Note:

With the CP 524 and with special drivers you can connect other manufacturers' equipment to your programmable controller (e.g. computers, printers or intelligent subsystems). For more information about special drivers contact your nearest Siemens branch or technical office.

1.2 Mechanical Design



*): when programming the 64 Kword memory submodule on the PG you need an adapter (MEP adapter), since a different programming voltage is required for EPROMs of the type 27256.

On its front panel, the communications processor has two 25-pin female Cannon connectors with serial device interfaces for connecting devices (printer PT88/PT89, other programmable controllers or central computers). Instead of these devices, device interface 2 (IF 2) can also be used to connect the programmer (PG).

Each of the device interfaces has an LED display.

Using the mode selector the following operating modes can be selected:

- RUN
- STOP = PGR (no jobs will be processed)

The function of the switch setting STOP is identical with that of PGR.

The memory submodule for the user program is plugged into the submodule receptacle. You can use either **EPROM submodules** or **RAM submodules**.

The connection to the S5 bus in the PC frame is established via two 48-pin backplane connectors.

1.3 Serial Device Interfaces

Transmission via both device interfaces is serial and asynchronous. Both loop current (TTY) and V.24 signals are available on the two device interfaces.

The loop current (20 mA) can be supplied by both the CP 525-2 and from the partner device. Only the end which provides the current is non-floating. The CP supplies the current (20 mA) via jumpers in the connector of the device interface. The 24 V required for generating loop current is taken from the PC bus. With a closed circuit, a current of 20 mA must flow (logical 1) if the line current circuit is functioning correctly. When the current is interrupted, logical 0 results.

The CCITT recommendations V.24/V.28 apply for the V.24 signals. A logical zero is a voltage greater than +3 V; a logical 1 a voltage less than -3 V.

The foregoing technical details are irrelevant if you use the standard cable connectors to connect the PT88/PT89 printer, other programmable controllers or certain computers. These connectors are designed for V.24 or TTY signals (see Section 6).

When you write the user program with the COM 535 programming package you decide the following:

- which partner is to be linked on which interface
- which settings are required for the transmission parameters (transmission rate, frame format)

For further details refer to the user's guide for the COM 525 programming package.

1.4 Dual-port RAM, Interface to the CPU

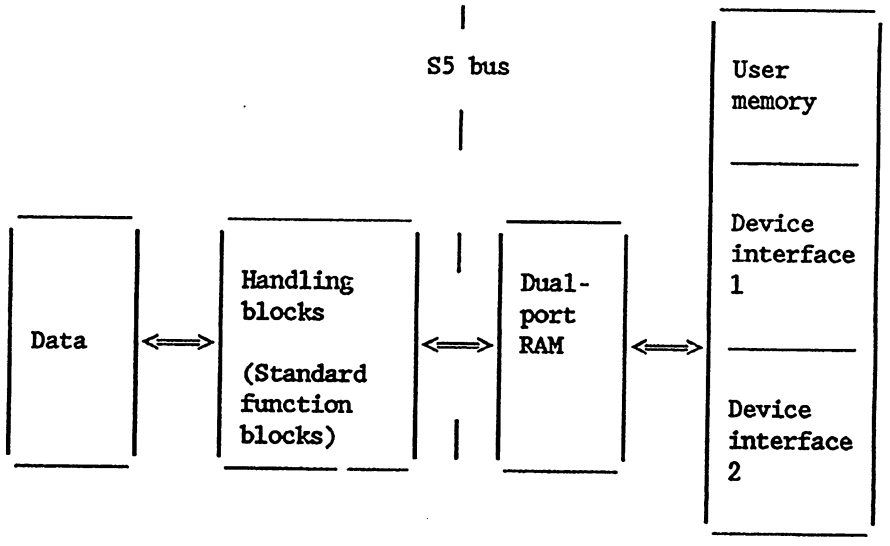
The data exchange between the CP 525-2 and the CPU is restricted to a minimum. Only the variable data (current process values) is exchanged between the CPU and the CP. The CP 525-2 has its own user memory for invariable data, e.g., message texts.

Data is exchanged between the CPU and the CP 525-2 using a common memory area: the **dual-port RAM**.

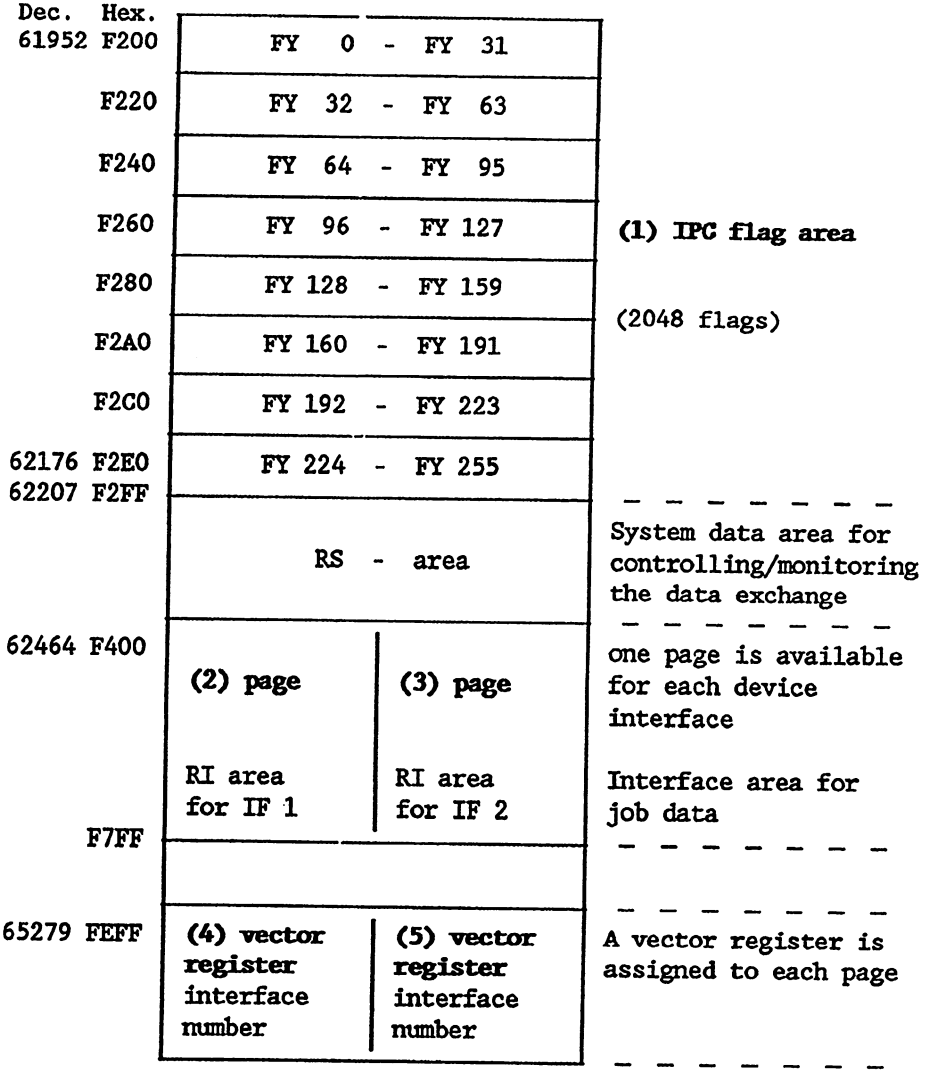
The dual-port RAM is on the CP 525-2 and is similar to a mailbox. The CP 525-2 and CPU can leave messages (data, requests, etc.) for each other in this mailbox.

The CPU of the PC always acts as master for data exchange. The CP 525-2 must "wait to be asked" by the CPU whether it wants to transfer data. This polling function is carried out by standard function blocks (**"handling blocks"**) in the CPU. These are the same handling blocks that are used by other communications processors.

The actual exchange of data between the CPU and CP 525-2 is also managed by handling blocks. For more details, refer to the description "Using the handling blocks" in this manual.



The **dual-port RAM** of the CP 525-2 is structured as follows:



This diagram is explained on the following page.

(1) Interprocessor communication (IPC) flag area:

IPC flags are defined as follows:

- IPC flags are part of the flag area of a CPU
  or
- if they are specified as IPC output bytes (in terms of the computer link) they are part of the dual-port RAM of the CP 525-2. These IPC flags are also called coordination flags.

IPC flags are required because they are used as follows:

- for the cyclic exchange of data between CPUs in bytes
- for the data exchange between CPUs and the CP 525-2. The IPC flag area in the dual-port RAM of the CP 525-2 is responsible for this function.

The IPC flag area in the CP 525-2 covers 256 IPC flag bytes (2048 IPC flags).

With the computer link and RK 512 the IPC flags (coordination flags) can be used to **inhibit and enable the data exchange between the CPU and CP**. By inhibiting this exchange, you can prevent data that has not yet been evaluated from being overwritten. These IPC flags must be designated as output flag bytes. For further details see the user's guide for the computer link with RK 512.

When you are using the printer PT88/PT89, you do **not** need the IPC flags.

The IPC flags are transferred once per STEP 5 program cycle between the flag area of the CPU and the IPC flag area of the CP 525-2 (similar to the process image for digital inputs and outputs).

To prevent multiple addressing when using several CPs or CPs and a coordinator in one PC, you must divide the IPC flags into groups (jumper settings see 3.4.2). This also applies if there is more than one CPU in a PC.

Note:

S5-115U: the IPC flags must be listed in DB1 as IPC output flags.

S5-135U:

S5-155U: the IPC flags must be entered as IPC output flags in the DB1 of each CPU. This is also the case when only one CPU is being used along with a CP 525-2.

S5-150U: since the IPC flags are in a different address area, the function block FB TRANSFER handles the cyclic exchange of IPC flags between the CPU and CP. The IPC flags must be specified when the block is called. The block call must be at the end of the cycle of your STEP 5 program.

(2)(3) Data transfer area (page)

Each device interface of the CP 525-2 has its own page assigned to it. This means that each CP 525-2 used in a programmable controller has **two pages**.

The page is the area within the dual-port RAM in which **data** is exchanged between the CP 525-2 and the CPU. This data may, e.g., be numerical values, to be logged on the printer. In addition, the programmable controller can also transfer system messages to the CP 525-2 via the page.

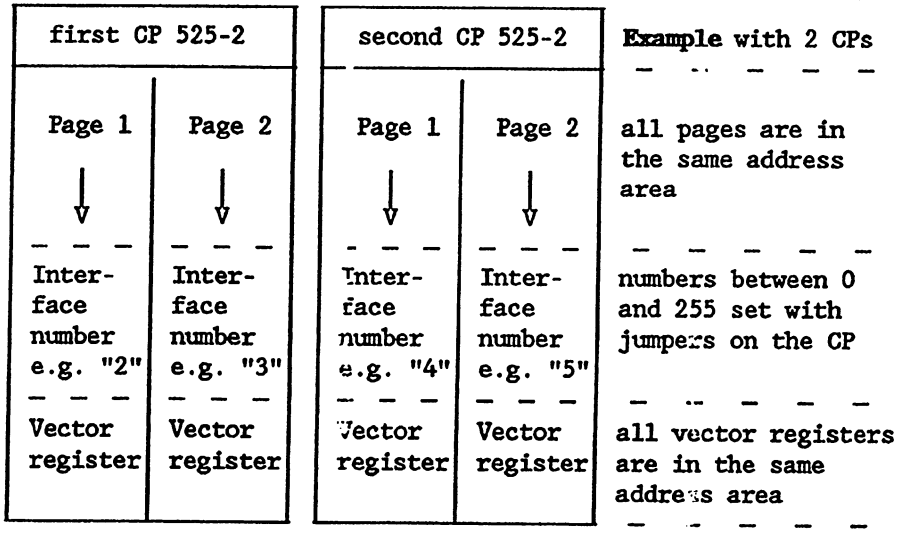
(4)(5) Vector register - interface number

The pages of all the CPs in a PC are in the same address area. To ensure that only the required page is addressed when data are exchanged, each page can only be addressed by its **interface number**. This is between 0 and 255 and is set on the CP by means of jumpers.

Each page has a **vector register** allocated to it. The vector registers of all the CPs have the same address.

If one of the two pages of a particular CP is to be addressed by the CPU (e.g., to transfer data to the CP 525-2), the CPU writes the interface number of the required page in all the **vector registers** available. Each CP compares the contents of the vector register with its own two interface numbers. If the CP recognizes that the number matches an interface number, the required page is enabled for data transfer. This function is carried out automatically by handling blocks.

Since the interface numbers of the CP 525-2 can only be set in pairs, you set the even interface number of IF 1 on the CP 525-2. The interface number of IF 2 is automatically set with the next higher number (for more details see 3.4.1 Module address/interface number).



1.5 Memory Submodules

The memory submodule of the CP 525-2 contains your CP 525 user program. The user program includes the following:

- the specific programs for the device interfaces ("interpreters and procedures"). These programs decide which function (computer link or printer) is carried out via which interface on the CP

- the "actual" user program, i.e., the message texts for the logging printer and/or the telegram parameters for the computer link

The interpreters and procedures of the two interfaces require a total of 8 Kwords (2 x 4 Kwords) in the memory submodule, regardless of whether one or both interfaces are being used. The CP 525-2 also stores a user data manager for each interface in the memory submodule. This manager is dynamic, i.e., it grows with the number of items of user data; per interface it can be a maximum of 4 Kwords long.

Computer link with RK 512: per interface a maximum 1 Kword is required for the telegram parameters. A **16 Kword memory submodule** is adequate for one computer link per CP 525-2 interface

Printer PT88/PT89: the available memory space depends on the capacity of the memory submodule and on whether one or two interfaces are being used.

You can use either **EPROM submodules or RAM submodules**. When it is plugged in, the RAM is battery-backed via the backplane bus of the programmable controller.

!IMPORTANT!: if the **RAM submodule** is unplugged from the CP 525-2 or the CP is disconnected from the PC, the user program stored in the RAM is lost!!!

The following memory submodules are available for the CP 525-2:

EPROM submodules

16 Kwords
32 Kwords
64 Kwords

When you are programming this submodule on the PG, you must use an adapter (MEP adapter), since a different programming voltage is required for EPROMs of the type 27256.

RAM submodules

16 Kwords
32 Kwords
64 Kwords

In the CP 525-2, you may also use the buffered 377 RAM submodule (64 Kwords).

The order numbers for these submodules can be found in the ordering data in this manual.

If you wish to use **both** interfaces the following submodules can be used:

IF1 and IF2	16 Kw submodule	32 Kw submodule	64 Kw submodule
PT88/PT89 / PT88/PT89	no	yes	yes
PT88/PT89 / CL	yes	yes	yes
CL / PT88/PT89	yes	yes	yes
CL / CL	yes	yes	yes

PT88/PT89 - printer PT88/PT89
CL - computer link with RK 512

If you only require **one** interface you can use all the memory submodules for the printer and the computer link.

1.6 Hardware Clock

In contrast to the CP 524 (software clock) the CP 525 has a battery-backed hardware clock to display the date and time.

Leap years and changes of year are automatically taken into account by the software of the CP. Even if the CP 525 is switched off (for up to 1 year), with data supported only by the battery back-up, changes of year and leap years are taken into account when it is switched on again.

As soon as the CP 525-2 module has been plugged into the programmable controller, the clock starts running with the settings: January 1st, 0.00 hours. The year is set at random.

Further details about the hardware clock can be found in the user's guide "Computer link with RK 512" or in the user's guide "CP 525: signalling and listing with the PT88/PT89 printer" (CP 525: reading and writing the date and time).

The hardware clock is accurate to +/- 50 ppm,
i.e. +/- 4.3 seconds per day.

1.7 Technical Data

Degree of protection according to DIN 40050 and IEC 144	IP00
Perm. ambient temperature	
during operation	0°C to 55°C
during transport and storage	-40°C to +70°C
Humidity class DIN 40040	code letter F
Operating altitude	up to 3,500 m above sea level
Power supply	5 V +/- 5%; 24 V for current sources
Current input 5 V	2.1 A max; 1.8 A measured statically
Current input 24 V	60 mA
Back-up voltage	min. 2.7 V
Back-up current (at 60°C and without RAM submodule)	140 µA max.
Transmission procedure	asynchronous, procedures can be loaded via PG
Transmission rate	110 to 19200 bps with TTY max. 9600 bps
Transmission cable	shielded 4-wire cable with braided shield and metal connector: earthing required both ends
Transmission link TTY	max. 1 km at 9600 bps
Transmission link V24	max. 20 m
Design	
dimensions (W x H x D)	20.32 mm x 160 mm x 233.4 mm
backplane connector (packaging system)	ES 902 / row 2, 48-pin
Weight	approx. 0.4 kg

2 Installation

2.1 Slots in the Programmable Controller

The CP 525-2 can be plugged into the following programmable controllers (fan required):

S5-115U with subrack CR 700-2
Slot number

		0	1	2	3	4	5		

S5-115U with subrack ER 701-3

	0	1	2	3	4	5	6		

The interface modules 304 and 314 are required

S5-115U with subrack CR 700-3:

		0	1	2	3	4	5			

The interface modules 304 and 314 are required
Fan required

S5-135U

	11	19	27	35	43	51	59	67		

With the CP 525-2, *no* coordination processor is required if only one CPU is used.

S5-150U

3	11	19		107	115	123	131	139	147	

Further details about your programmable controller can be found in the appropriate manual.

S5-155U

	19		35	43	51	59		75	83	91	99	107	115	123	131	139	147

The CP 525-2 can be plugged into the following expansion unit (fan required):

EU 185U

19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139

The interface modules 304 and 314 are required.

2.2 Plugging in and Removing the CP 525-2

Do not remove or plug in the CP 525-2 when the power is on! This also applies when you are replacing the memory submodule!

2.3 Installation Guidelines

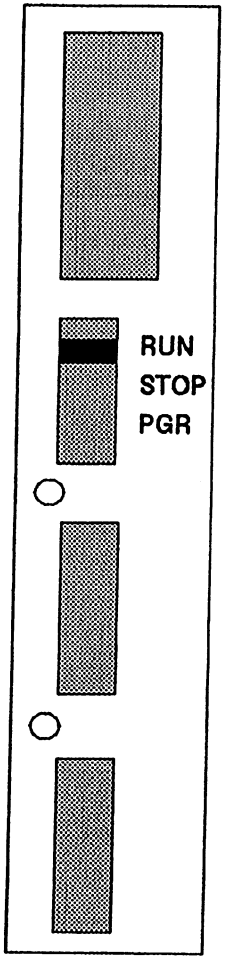
The connecting cable between the CP 525-2 and a partner device must be a cable with a shield earthed at both ends and with metal connectors. The standard cable connectors listed in the catalog and in Section 6 meet these requirements.

The shield must make large area contact with the casing of the metal connector (not on pin 7). If the partner device does not have a metal connector, the cable shield must make large area contact with ground. If there is a large difference in ground potential between the CP 525-2 and the partner device, a potential equalizing cable should be included. The cable should be kept away from thyristor drives and power cables carrying more than 500 V. Further information about installation can be found in the "Installation guidelines for SIMATIC compact devices" (see ordering data).

3 Operation

3.1 Controls and Displays

CP 525-2 (View from front)



Mode selector

LED display for device interface 1

LED display for device interface 2

3.2 Operating Modes

Mode selector

RUN	The processor runs in the normal operating mode, the user functions which have been programmed are processed. The LEDs are not lit (if LEDs are lit, see 3.3)
STOP = PGR	No jobs are processed. The function of the switch position "STOP" is identical with that of "PGR". The LEDs are lit.

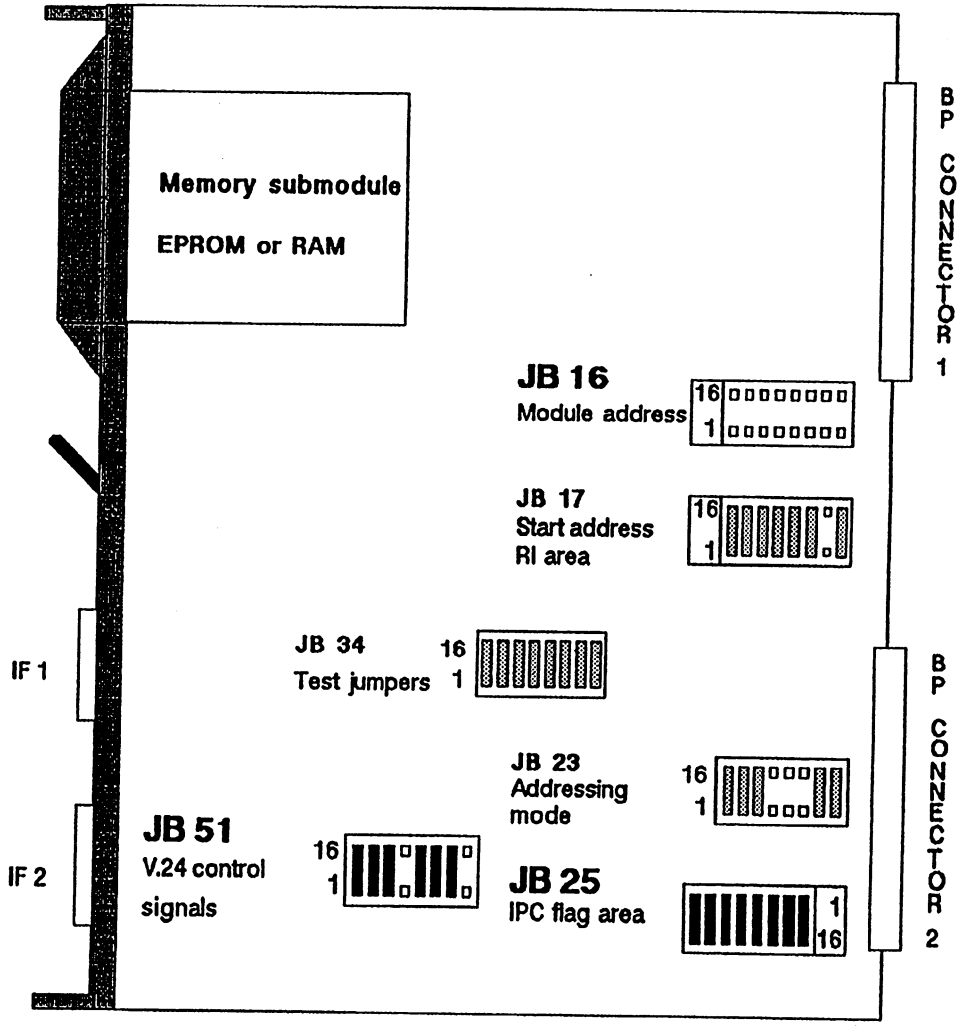
3.3 Status Displays

Each interface has a red LED display. If the LED display of a device interface is lit, one of the following has occurred:

- device driver and procedure for the interface not loaded
- mode selector switched to STOP/PGR
- following the loading of the user program, drive and procedure, no CP 525-2 cold restart was carried out
- CP 525-2 still in start-up phase following a power failure (takes max. 10 s)
- "serious" error recognized during start-up (e.g., wrong memory submodule; submodule type not permitted).

3.4 Jumper Settings

The diagram below shows the position of the jumper blocks on the module. Adjustments need to be made only on JB 16 / JB 25 / JB 51. The jumpers on the remaining jumper blocks are set in the factory and should not be changed.

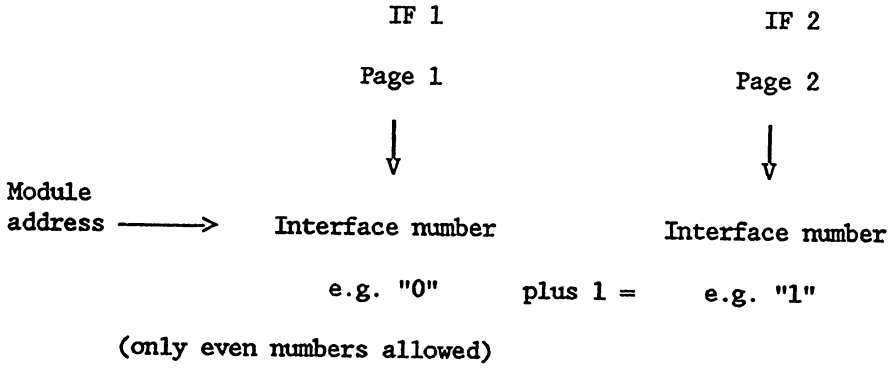


IF... = Device interface ... JB... = jumper block ...

3.4.1 Module Address / Interface Number

As described in 1.4, the data is transferred via the page area of the dual-port RAM. Each of the two device interfaces of the CP 525-2 has its own page and is allocated its own interface number (values from 0 to 255).

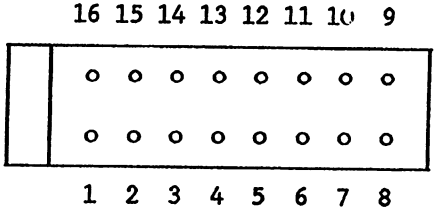
You specify the interface numbers by setting the **module address** of the CP 525-2 at **JB 16**. For example, if you select the module address "0", (no jumpers inserted), then device interface 1 (IF 1) has the interface number "0". Device interface 2 (IF 2) is automatically assigned the next higher number - interface number "1".



!IMPORTANT! The jumper setting is binary. Only **even numbers** can be selected for **IF 1**; IF 2 is then automatically assigned the next higher odd number.

Note: to ensure that the CPU addresses the CP correctly, you must specify the same interface number (parameter SSNR) when assigning parameters to the handling blocks as that set on the CP.

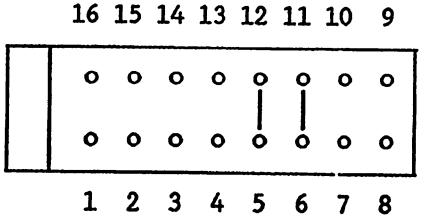
Jumper block no. 16:



Jumper	Value	
8-9	irrelevant	2^0
7-10	2	2^1
6-11	4	2^2
5-12	8	2^3
4-13	16	2^4
3-14	32	2^5
2-15	64	2^6
1-16	128	2^7

Jumper open: $0 \times 2^n = 0$
Jumper inserted: 1×2^n

Example: selected interface number equals 12



12 decimal = $0x2^0+0x2^1+1x2^2+1x2^3+0x2^4+0x2^5+0x2^6+0x2^7$

Jumpers 6-11 and 5-12 are inserted; the other jumpers remain open.

The upper device interface (i.e. IF 1) now has interface number 12 and the lower device interface therefore automatically has address 13.

3.4.2 Enabling/Disabling IPC Flags

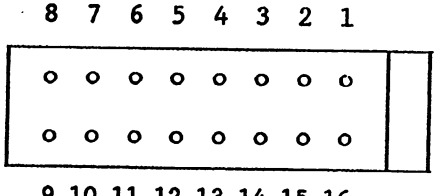
As described in 1.4 and in the user's guide "Computer link with RK 512" you can enable or inhibit the data transfer between the CPU and CP 525-2 by using the IPC flags. These IPC flags must be specified as output flags; in this situation the term "coordination flag" can be used instead of the term IPC flag.

The printer PT88/PT89 does not require any IPC flags.

A total of 256 IPC flag bytes are available in the programmable controller (2048 IPC flags).

In multiprocessor operation, IPC flags are a means of exchanging data between the processors. If several CP 525-2s or several CPUs (in the S5-135U) are installed in the programmable controller the IPC flags must be divided up among the CPs and the coordinator (see also instructions for the S5-135U). The IPC flags must then be enabled on the modules by means of jumpers. On the CP 525-2 there are eight jumpers available at jumper block 25. With the appropriate jumper setting on this block, 32 contiguous IPC flag bytes can be enabled or disabled.

Jumper block no. 25:

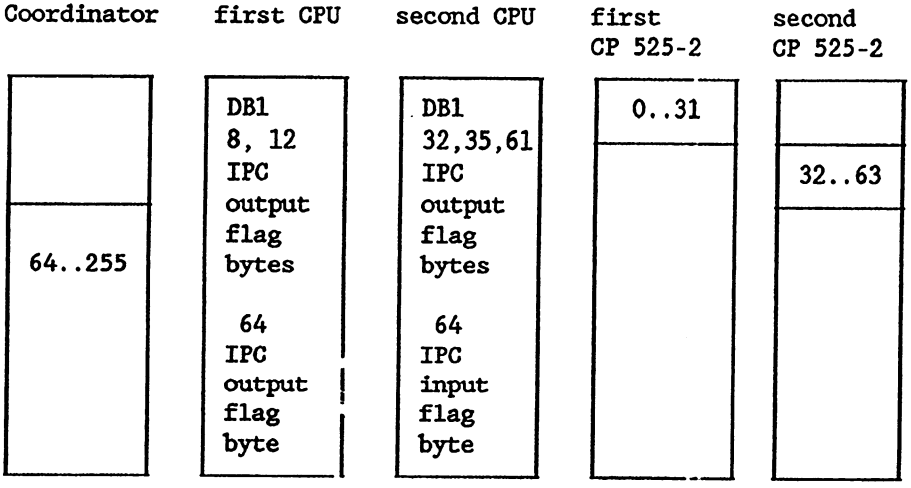


Jumper	IPC flag byte
8- 9	0 ... 31
7-10	32 ... 63
6-11	64 ... 95
5-12	96 .. 127
4-13	128 .. 159
3-14	160 .. 191
2-15	192 .. 223
1-16	224 .. 255

Jumper inserted: the corresponding IPC flag bytes are enabled.

!IMPORTANT!: the corresponding IPC flag bytes must also be entered in the DBls of the CPUs in the S5-135U and 115U! These can then be addressed as flags in the STEP 5 program.

The following is an example of the distribution of IPC flags between two CP 525-2s and one coordinator with the S5-135U:



This example specifies that the first CPU "coordinates" the data exchange with the first CP 525-2 via one (or several) bits of the IPC flag bytes 8 and 12. Therefore the area 0 to 31 is enabled on the first CP 525-2. (An area of 32 contiguous IPC flags must always be enabled or disabled). Jumper 8-9 is inserted on jumper block 25. In DB1 of the first CPU, you must enter bytes 8 and 12 as IPC output flag bytes.

In the same way the second CPU coordinates the data exchange with the second CP 525-2 via IPC flags 32, 35, 61. On the second CP 525-2 the area 32 to 63 must therefore be enabled (jumper 7-10 inserted). You must then enter the bytes 32, 35 and 61 as IPC output flags in DB1 of the second CPU.

Since the IPC flag area exists only once in each PC, only the IPC flags 64 to 255 can be used for other purposes. In this example we selected the IPC flag 64 for CPU - CPU communication (IPC output flag in DB1 of the first CPU - IPC input flag in DB1 of the second CPU). The area which contains byte 64 must therefore also be enabled in the coordinator.

3.4.3 Settings of the Device Interfaces

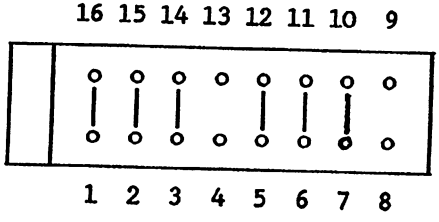
Each device interface has both 20 mA current loop signals and V.24 signals. According to CCITT, control signals are required for V.24. The evaluation of the control signals depends on the device procedure installed. The jumpers are set at the factory so that the signals are not evaluated by the procedure software. This setting means that all devices intended for connection to the CP 525-2 can be operated.

The control signals are as follows:

Pin		
6	Data set ready	DSR
5	Clear to send	CTS

On each device interface one of these signals can be used (as required).

The factory setting of the jumpers is as follows:



Jumper block no. 51

The significance of the jumpers is as follows:

Device interface 1:

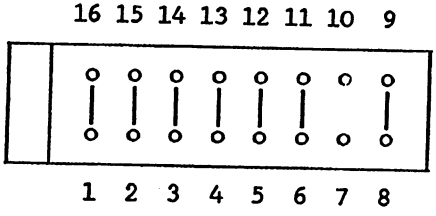
8- 9 inserted and 7-10 open	= DSR (pin 6) selected
7-10 inserted and 8- 9 open	= CTS (pin 5) selected
6-11 inserted	= CTS/DSR input not used
6-11 open	= the signal selected with jumpers 8-9 and 7-10 can be evaluated by the software
5-12 open	= the signal is inverted for evaluation
5-12 inserted	= the signal is evaluated directly

Device interface 2:

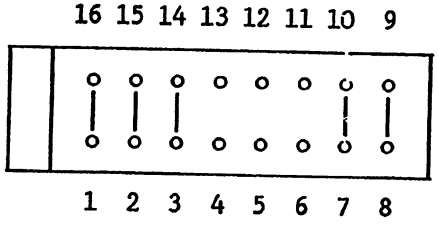
4-13 inserted and 3-14 open	= DSR (pin 6) selected
3-14 inserted and 4-13 open	= CTS (pin 5) selected
2-15 inserted	= CTS/DSR input not used
2-15 open	= the signal selected with jumpers 4-13 and 3-14 can be evaluated by the software
1-16 open	= the signal is inverted for evaluation
1-16 inserted	= the signal is evaluated directly

3.4.4 Permanent Jumper Settings

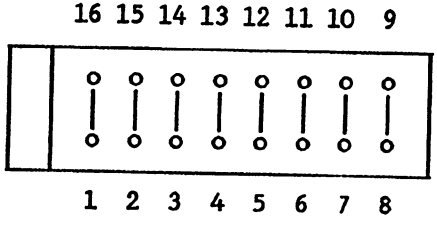
The following jumpers are set in the factory and must not be changed:



Jumper block no. 17



Jumper block no. 23



Jumper block no. 34

4 Setting up the Hardware

When installing the CP 525-2 module, follow the checklist below:

1. Check the jumper setting on the module according to 3.4.

2. Insert a memory submodule in the CP submodule receptacle.

3. Plug the CP 525-2 module into one of the permitted slots in your device. The power supply must be switched off (see 2.1).

4. Connect the partner device (e.g. PT88/PT89) and the CP 525-2 (see Section 6).

5. Check the position of the mode selector on the CP 525-2. It must be switched to "RUN", to allow data exchange with the partner device.

6. Make the correct settings at the partner device. These settings can be found in the user's guide "Computer link with RK 512" and "Printer PT88/PT89" in the section describing installation.

   Make sure that the parameters required for transmission (data rate, parity, stop bit, character length) are the same in both the partner device and the CP 525-2.

7. Switch on the power supply to your CP. Both LED displays on the front of the CP 525-2 light up. The operating system carries out a self-test. The LED of the device interface being started goes out if no error has been detected and you have done one of the following:

   - plugged a programmed EPROM into the CP (device driver and transmission procedure must be programmed for each device interface)
   -transferred your user program from the PG to the RAM plugged into the CP 525-2 and have then carried out a **cold restart**.

5 Pin Assignments

5.1 Backplane Connector 1

	d	b	z
2		Ground	+ 5 V
4	U BAT		
6	ADB 12	ADB 0	/CPKL
8	ADB 13	ADB 1	/MEMR
10	ADB 14	ADB 2	/MEMW
12	ADB 15	ADB 3	/RDY
14		ADB 4	DB 0
16		ADB 5	DB 1
18		ADB 6	DB 2
20		ADB 7	DB 3
22		ADB 8	DB 4
24		ADB 9	DB 5
26		ADB 10	DB 6
28	/DSI	ADB 11	DB 7
30		BASP	
32		Ground	

5.2 Backplane Connector 2

	d	b	z
2		Ground	+ 5 V
4			
6			
8			
10			
12			
14			/NAU
16			/BAU
18			
20	/HOLDA		/DMAFA
22	TXD	/STOPPA	
24			GEP
26		RXD	
28			
30			GROUND 24V
32		Ground	+ 24 V

5.3 Memory Submodule Interface

	c	b	a
1	ADB 12	Ground	+ 5 V
2	ADB 0	ADB 1	ADB 2
3	ADB 3	ADB 4	ADB 5
4	ADB 6	ADB 7	ADB 8
5	ADB 9	ADB 10	ADB 11
6	ADB 13	ADB 14	/RD
7	/WR	DB 0	DB 1
8	DB 2	DB 3	DB 4
9	DB 5	DB 6	DB 7
10	DB 0	DB 1	DB 2
11	DB 3	DB 4	DB 5
12	DB 6	DB 7	K1
13	/CS	/CS	K2
14	/CS	/CS	K3
15	UCMOS	PSW	K4
16	+ 5 V	Ground	K5

5.4 Device Interface 1 (IF 1)

Pin	Designation
1 -	ground (G_{ex})
2 -	transmitter TXD V24 IF1
3 -	receiver RXD V24 IF1
5 -	clear to send CTS V24 IF1
6 -	data set ready DSR V24 IF1; receiver must be converted from CTS to DSR with jumper.
7 -	TTL 0 V
8 -	not used
9 -	not used
10 -	transmitter (+) TTY IF
11 -	not used
12 -	current source (+) 20 mA
13 -	receiver (+) TTY IF1
14 -	receiver (-) TTY IF1
15 -	not used
16 -	current source u(+) 20 mA
17 -	not used
18 -	not used
19 -	transmitter (-) TTY IF1
20 -	not used
21 -	current source ( ) 20 mA
22 -	not used
23 -	not used
24 -	current source (-) 20 mA
25 -	not used

5.5 Device Interface 2 (IF 2)

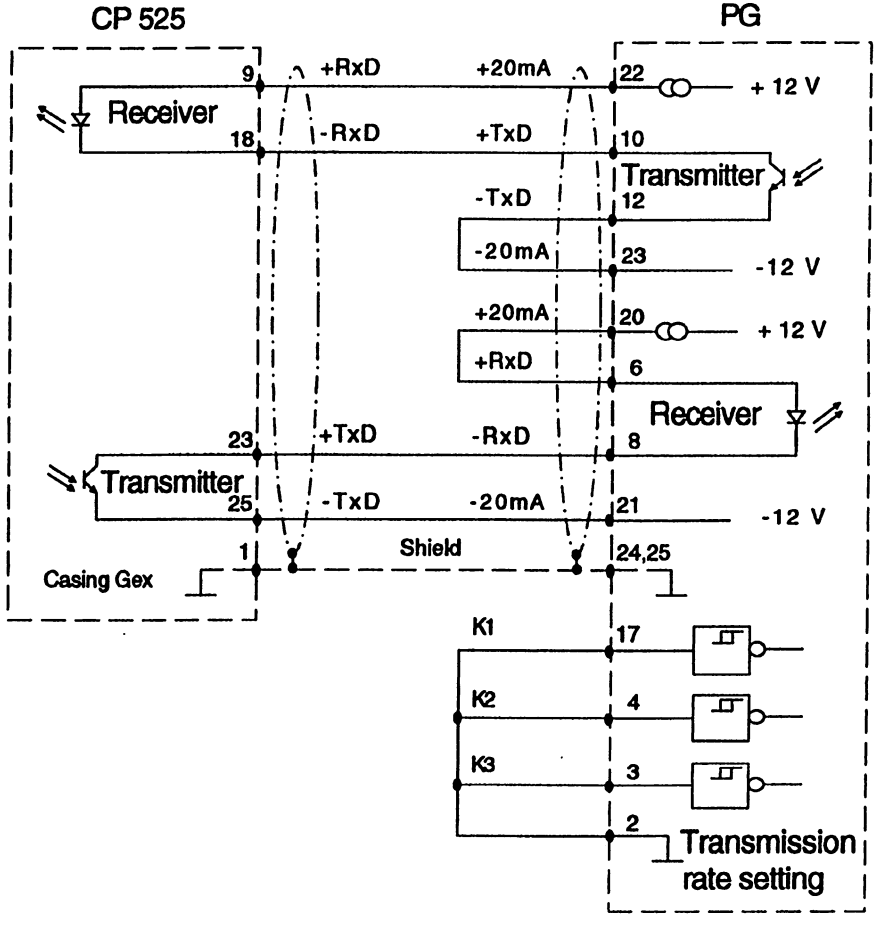
Pin	Designation	
1 -	ground (G_{ex})	
2 -	transmitter TXD V24 IF2	
3 -	receiver RXD V24 IF2	
5 -	clear to send CTS V24 IF2	
6 -	data set ready DSR V24 IF2;	receiver must be converted from CTS to DSR with a jumper.
7 -	TTL 0 V	
8 -	not used	
9 -	receiver (+) PG TTY	
10 -	transmitter (+) TTY IF2	
11 -	not used	
12 -	current source (+) 20 mA	
13 -	receiver (+) TTY IF2	
14 -	receiver (-) TTY IF2	
15 -	not used	
16 -	current source (+) 20 mA	
17 -	not used	
18 -	receiver (-) PG TTY	
19 -	transmitter (-) TTY IF2	
20 -	not used	
21 -	current source (-) 20 mA	
22 -	not used	
23 -	transmitter (+) PG TTY	
24 -	current source (-) 20 mA	
25 -	transmitter (-) PG TTY	

6 Standard Cable Connectors

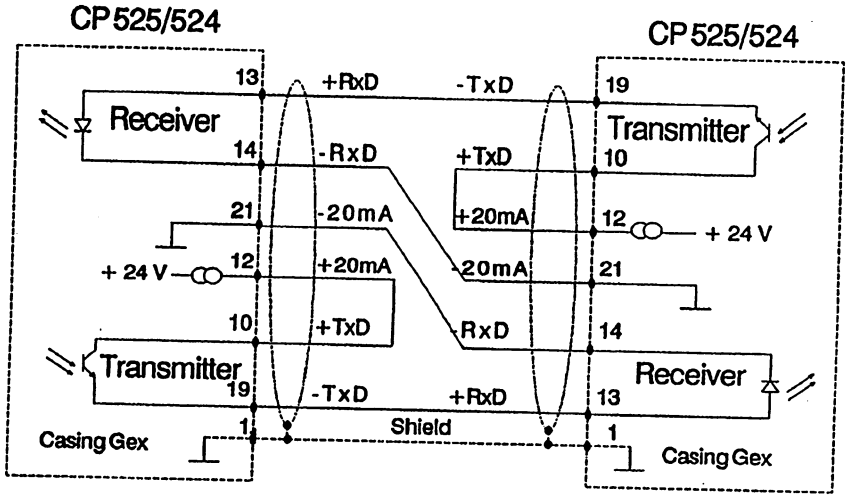
For the most common connections between devices there are standard cable connectors available (see ordering data).

6.1 Connection CP 525 - PG 6xx (TTY)

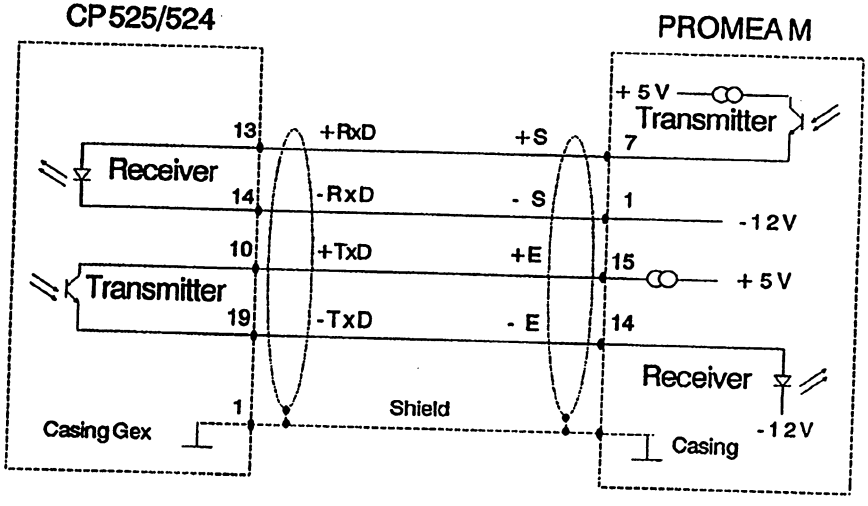
The cable can only be connected at the lower jack of the CP 525.



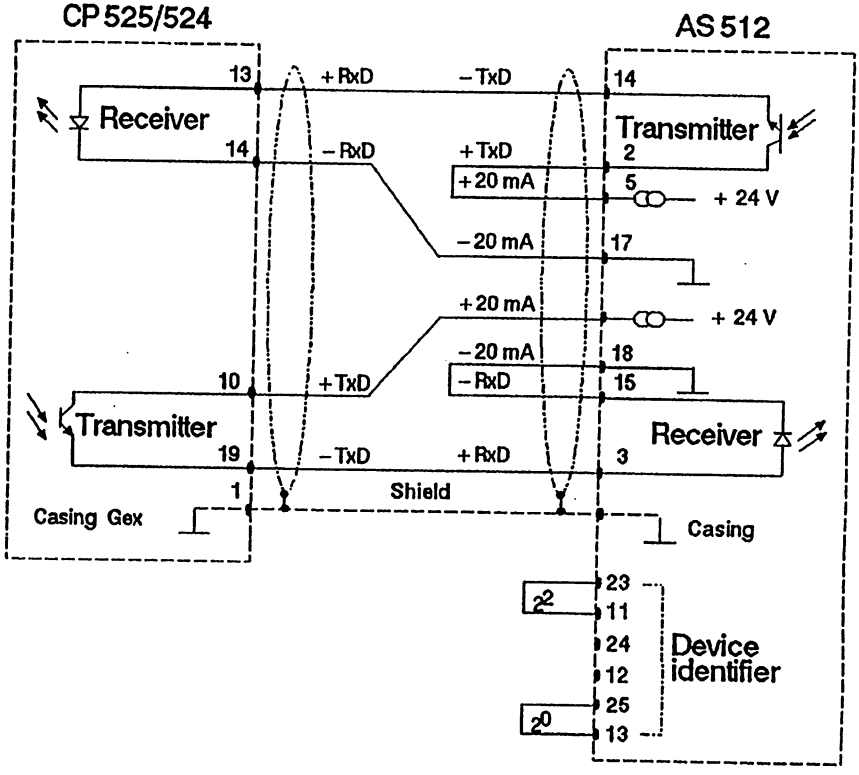
6.2 Connection CP 525/524 - CP 525/524 (TTY)



6.3 Connection CP 525/524 - PROMEA M (TTY)

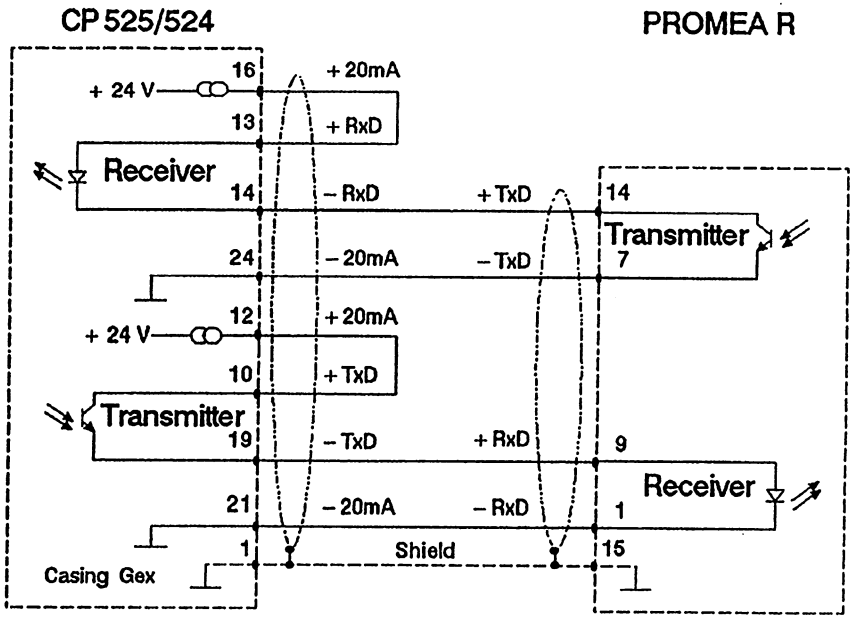


6.4 Connection CP 525/524 - AS 512C (TTY)

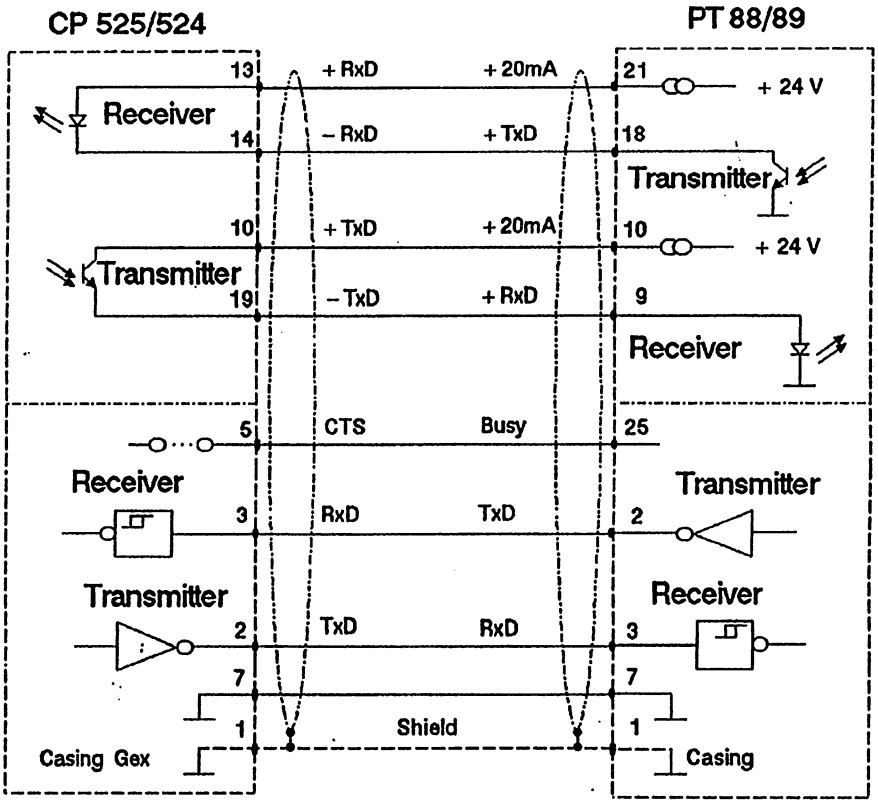


Note: to generate a current loop the AS 512 must be supplied with 24 V via the AMP connectors on the front panel.

6.5 Connection CP 525/524 - PROMEA R (TTY)



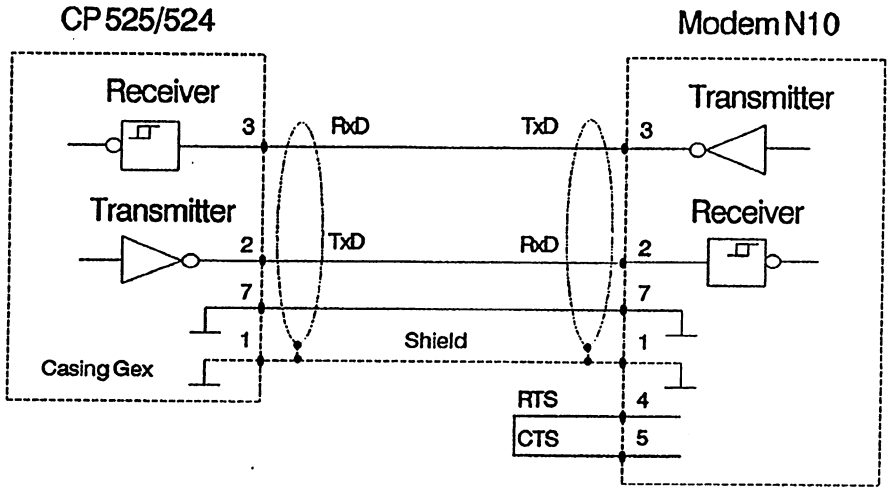
6.6 Connection CP 525/524 - PT88/PT89 (TTY/V.24)



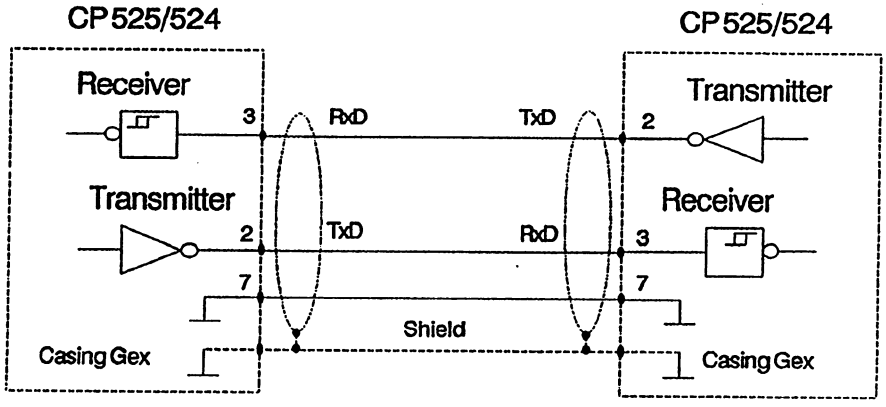
Note: The cable is designed for both TTY and V.24 signals. It can therefore be used regardless of the type of interface used in the printer.

With the CP 524, make sure that you use the same interface type as in the printer.

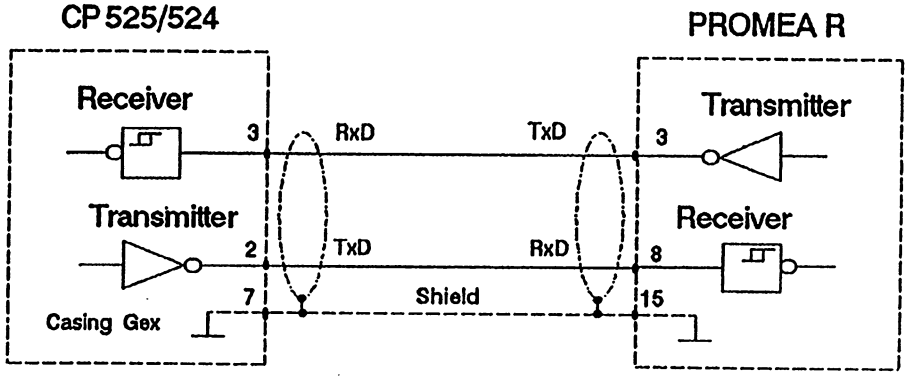
6.7 Connection CP 525/524 - Modem N10 (V.24)



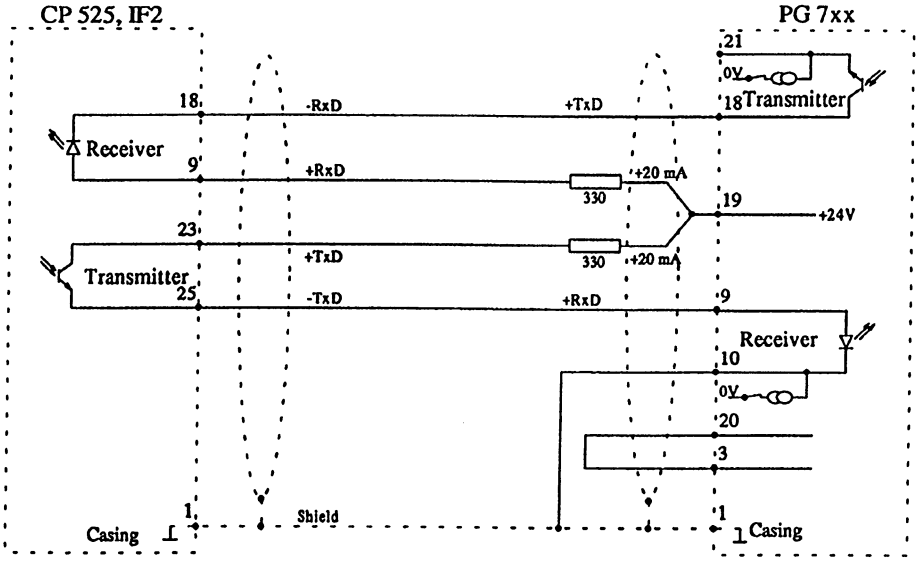
6.8 Connection CP 525/524 - CP 525/524 (V.24)



6.9 Connection CP 525/524 - PROMEA R (V.24)



6.10 Connection CP 525, IF2 - PG 7xx (TTY)



SIEMENS

Siemens Aktiengesellschaft

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 model or design, are reserved.

© Siemens AG 1988

Order No.. C79000-B8576-C537

Order from: Gerätewerk Karlsruhe

Printed in the Federal Republic of Germany
AG 0689 0.5 AB 46 En

SIEMENS

SIMATIC S5

Communications Processor CP 524

Instructions C79000-B8576-C538-06

Contents Page

1	**Technical Description**	**1**
1.1	Application	1
1.2	Mechanical Design	3
1.3	Serial Device Interface	4
1.3.1	V.24 Module	5
1.3.2	TTY Module	8
1.3.3	RS422-A/485 Module	10
1.4	Dual-port RAM: Interface to the CPU	13
1.5	Memory Submodules	19
1.6	Technical Data	20
2	**Installation**	**22**
2.1	Slots in the Programmable Controller	22
2.2	Plugging in and Removing the CP 524	24
2.3	Installation Guidelines	24
3	**Operation**	**25**
3.1	Controls and Displays	25
3.2	Operating Modes	26
3.3	Status Displays	26
3.4	Jumper Settings	27
3.4.1	Module Address/Interface Number	31
3.4.2	Enabling/Disabling IPC Flags	32
3.4.3	Jumpers on the V.24 Module	35
3.4.4	Jumpers on the TTY module	36
3.4.5	Jumpers on the RS422-A/485 Module	37
4	**Setting up the Hardware**	**40**
5	**Pin Assignments**	**41**

6 Standard Cable Connectors 44

6.1 Connection CP 524/525 - CP 524/525 (TTY) 44
6.2 Connection CP 524/525 - PROMEA M (TTY) 45
6.3 Connection CP 524/525 - AS 512C (TTY) 46
6.4 Connection CP 524/525 - PROMEA R (TTY) 47
6.5 Connection CP 524/525 - Modem N10 (V.24) 47
6.6 Connection CP 524/525 - CP 524/525 (V.24) 48
6.7 Connection CP 524/525 - PROMEA R (V.24) 48
6.8 Connection CP 524/525 - PT 88/PT 89 (TTY/V.24) 49
6.9 Connection CP 524 - CP 524 (RS422-A or V.11) 50

1 Technical Description

1.1 Application

The **communications processor CP 524** can be used in the programmable controllers of the SIMATIC 5 U range (S5-115U, S5-135U, S5-150U and S5 155U). **Note that it cannot be used in conjunction with the S5-135U / S processor.**

Using the communications processor CP 524 and the accompanying programming package COM 525 you can perform the following functions:

- **logging process statuses and messages from the process** and/or
- **link your programmable controller to other programmable controllers or computers.**

In association with the PT 88/89 printer

- process statuses and messages can be logged. The communications processor CP 524 initiates the preparation of messages to be output on the printer.

If the programmable controller (PC) is part of an automation network, it can

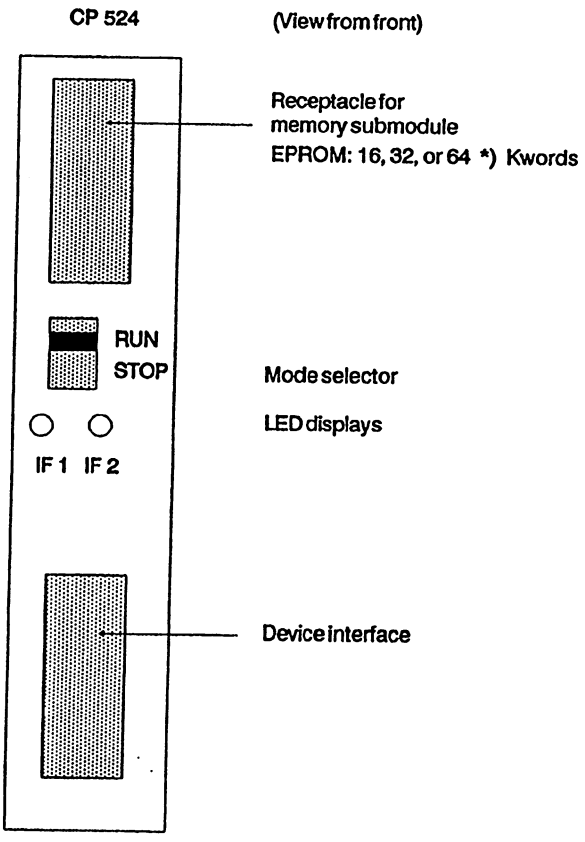
- exchange data with one or more programmable controllers or with a supervisory process computer (computer link). The CP 524 performs these tasks independently.

The data exchange between the CP 524 and the central processor (CPU) of the programmable controller is restricted to a minimum. Only the variable data (current process values) are exchanged between the CPU and the CP 524. The CP 524 has its own memory for invariable data, which you program using the software package COM 525.

Note:

With the CP 524 and with special drivers you can connect other manufacturers' equipment to your programmable controller (e.g. computers, printers or intelligent subsystems). For more information about special drivers contact your nearest Siemens branch or technical office.

1.2 Mechanical Design



*): when programming the 64 Kword memory submodule on the programmer, you need an adapter (MEP adapter), since a different programming voltage is required for EPROMs of the type 27256.

The CP 524 has one device interface; the interface module is plugged into the lower receptacle of the CP 524. You can select a V.24, TTY or RS422-A/485 interface.

The mode selector is used to select the operating mode "RUN" or "STOP".

More detailed information on the two LED displays can be found in 3.1 "Controls and displays".

The memory submodule containing your CP 524 user program is plugged into the upper receptacle of the CP 524. In the CP 524, **only EPROM submodules** can be used. These must first be programmed with the programmer using the programming package COM 525.

The connection to the S5 bus in the PC frame is established via two 48-pin backplane connectors.

1.3 Serial Device Interface

The serial device interface is plugged into the CP 524 and secured by means of two screws. The following interface modules are available:

TTY module
V.24 module
RS422-A/485 module

Connecting the CP 524 to the partner device is made easier by using standard cable connectors (see also Section 6). If you do not wish to use these standard cables, note the pin assignment of the interface module in 1.3.1, 1.3.2 or 1.3.3.

The following **standard cable connectors** are available:

CP 524 connected to:	Module in CP 524:	Interconnection in CP 524: transmitter, receiver
CP 524	TTY - module	active, passive
CP 525	TTY - module	active, passive
AS 512C	TTY - module	passive, passive
PROMEA M	TTY - module	passive, passive
PROMEA R	TTY - module	active, active
Modem N10	V.24 - module	
CP 524	V.24 - module	
PROMEA R	V.24 - module	
PT 88/89	TTY module	passive, passive
PT 88/89	V.24 module	
CP 524	RS422/A-485 module	

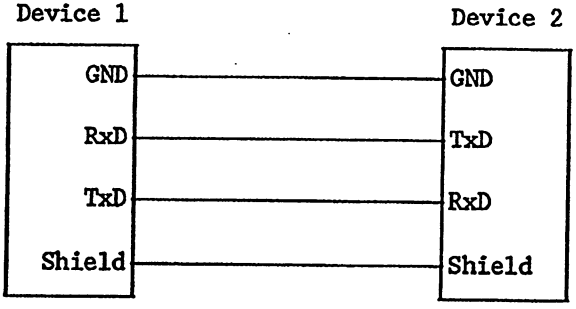
The order numbers can be found in the ordering data section of this manual.

When you write your CP 524 user program using the programming package COM 525 on the programmer, you set the transmission parameters (transmission speed, frame format).

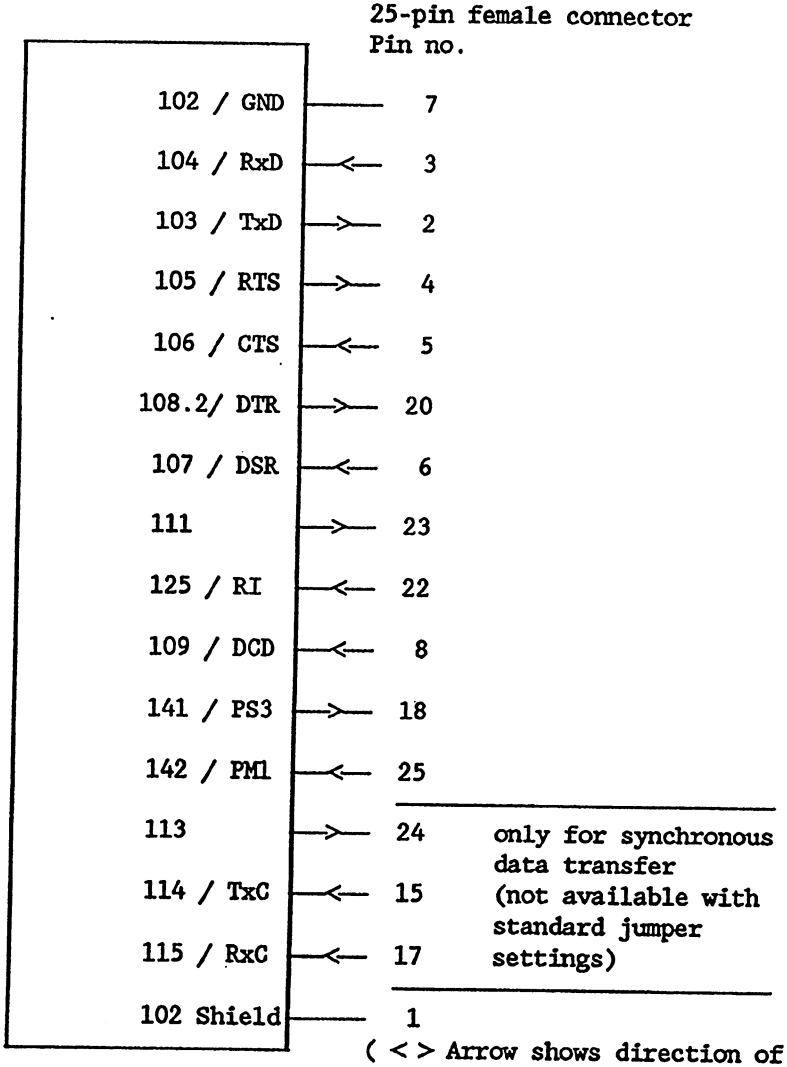
1.3.1 V.24 Module

The V.24 interface module not only has lines for transmitting and receiving but also has a series of control and signal lines in accordance with the CCITT recommendations V.24/V.28. For the computer link with RK 512 and the PT 88/89 printer these control signals are, however, not required and do not need to be connected. If you want to prepare your own cable connectors, remember that unused inputs may have to be connected to open circuit potential on other devices. For more information refer to the appropriate manuals and the CCITT recommendations V.24 or V.28.

The following diagram shows the basic interconnection of V.24 interfaces. In the example only the transmit and receive lines are connected. With V.24 signals, logical 0 is a voltage greater than +3 V; logical 1 is less than -3 V.



The following figure shows the pin assignment of the 25-pin female Cannon connector on the front panel of the V.24 module:



Signal numbering is according to CCITT recommendation V.24, signal designation is according to internationally accepted abbreviations (RS 232 C).

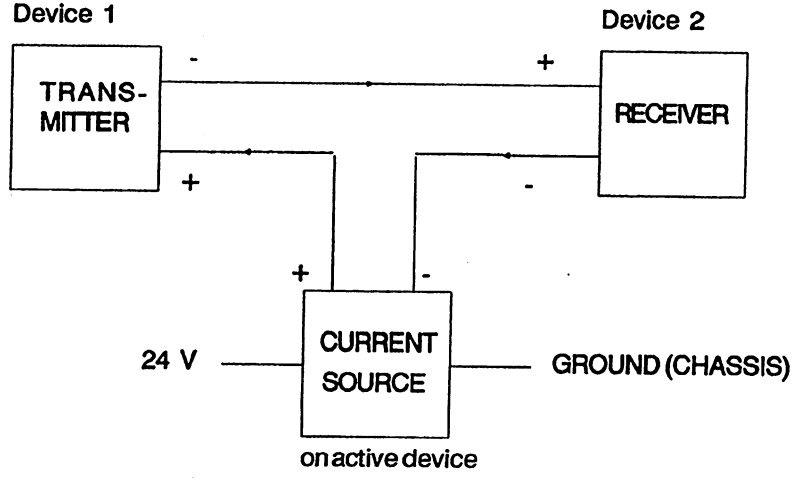
The position and significance of the jumpers on the V.24 module can be found in 3.4 in these instructions.

When you program your CP 524 user program using the programming package COM 525, you specify the setting of the transmission parameters (transmission rate, frame format).

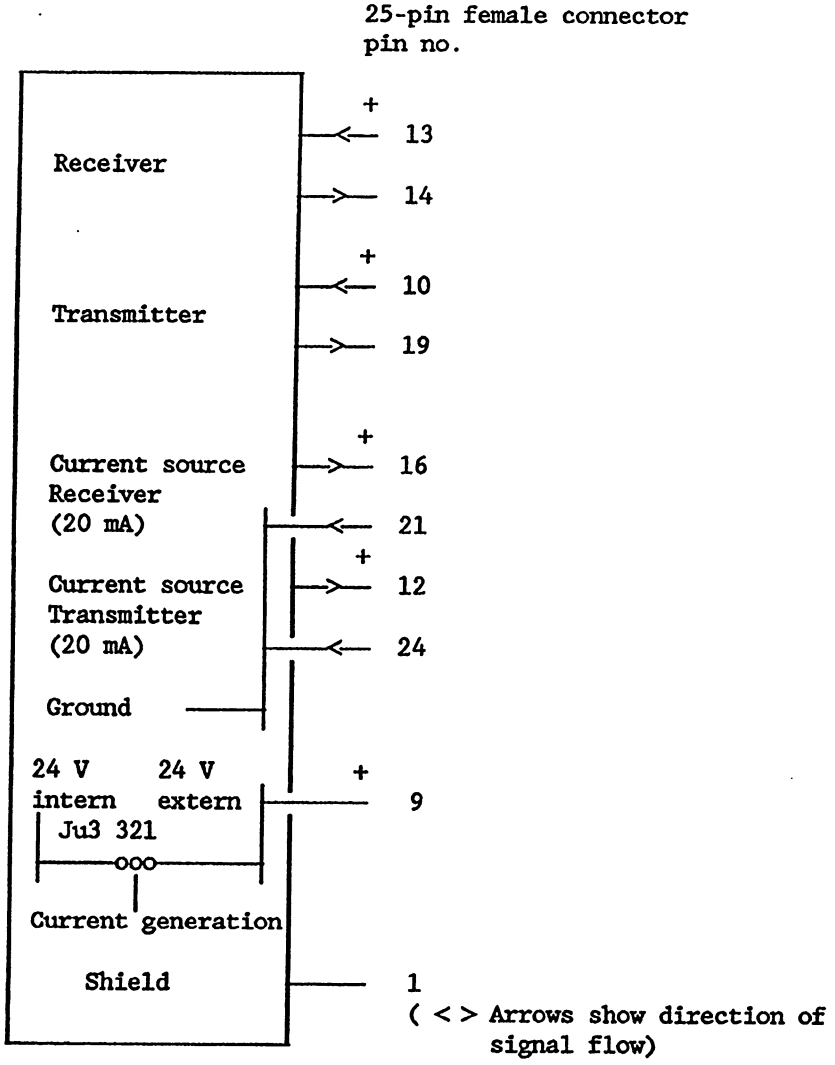
1.3.2 TTY Module

The TTY module is equipped with a transmitter and receiver for 20 mA current loop signals. The current loop itself can be supplied both by the CP 524 and the partner device. Only the end which supplies the current is non-floating. The CP supplies the current (20 mA) via jumpers in the front connector. The 24 V required for generating the current loop are taken from the PC bus. With a closed circuit, a current of 20 mA must flow (logical 1) if the line current circuit is functioning correctly. When the current is interrupted logical 0. results.

The following diagram shows the interconnection of current loop signals.



The following figure shows the pin assignment of the 25-pin female Cannon connector on the front panel of the TTY module:



5

The position and significance of the jumpers on the TTY module can be found in 3.4 in these instructions.

You set the parameters for the data transmission (data rate, parity, stop bits) using the programming package COM 525 on the PG. A maximum data rate of 9600 bps is possible with the TTY module.

1.3.3 RS422-A/485 Module

The RS422-A/485 module can be used in the **computer link** to transfer data in full or in half duplex. The jumper settings can be found in Section 3.4. In half duplex operation, you cannot use the standard procedures 3964 and 3964R. In this case, you must use special drivers.

The RS422-A/485 module can also be used with the **logging printer PT88/PT89**, there are, however, no standard cable connectors available.

The RS422-A/485 module electrically isolates the interface signals from the power supply. The electrical characteristics of the RS422-A/485 module are as follows:

- in full duplex mode: according to the EIA standard RS422-A (CCITT recommendation V.11)
- in half duplex mode: according to the EIA standard RS485.

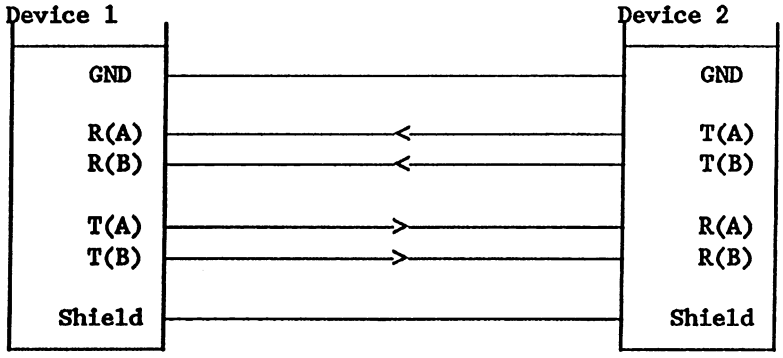
The difference between the two standards is that only one transmitter may be available with the EIA standard RS422-A, whereas several transmitters and receivers are permitted with the EIA standard RS485. In half duplex mode, the data exchange can only take place via the two-wire line R.

Apart from the transmit and receive lines, the RS422-A/485 module is equipped with a series of control and monitoring lines according to the CCITT recommendation X.24 and ISO 8481. For the computer link and the logging printer PT88/PT89, these control and monitoring signals are not required and do not need to be connected.

In the following examples, only transmit and receive lines are connected. The following applies to signals according to the EIA standard RS422-A (V.11) and EIA standard RS485:

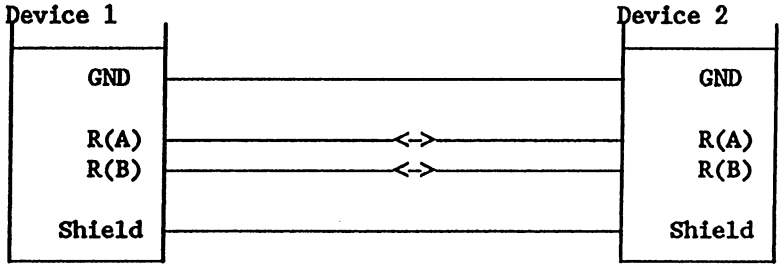
- logic 0 (ON) corresponds to $V_A > V_B$ V_A = greater than +3 V
- logic 1 (OFF) corresponds to $V_A < V_B$ V_B = smaller than -3 V.

Basic wiring of the interface in **full duplex mode**:



(< > signal direction)

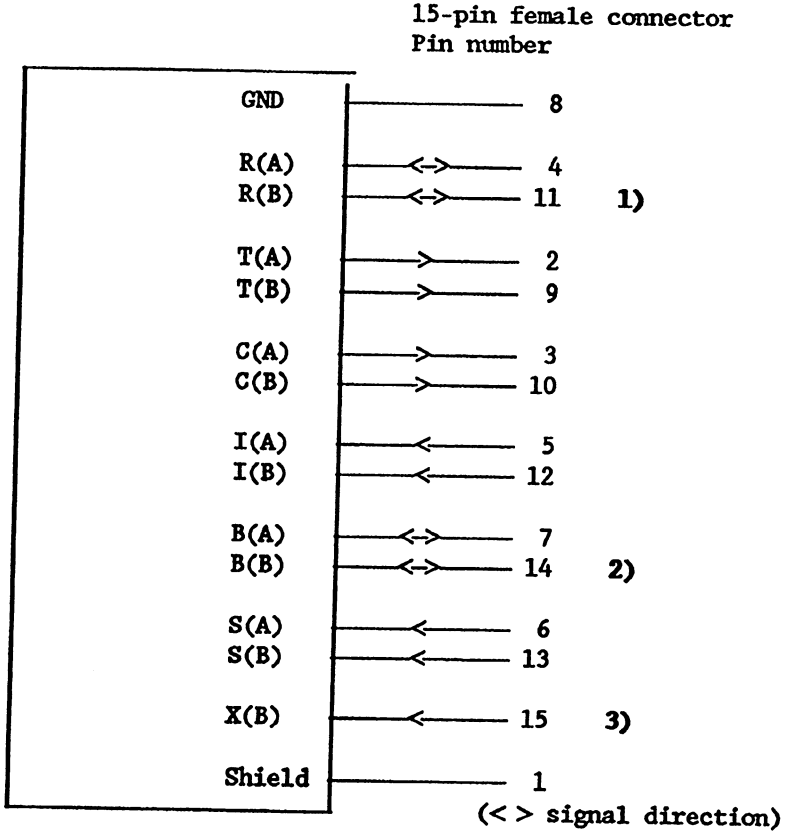
Basic wiring of the interface in **half duplex mode**:



(< > signal direction)

Caution: Do not connect the shield with the GND. Otherwise the electrical isolation is disabled.

The following diagram shows the pin assignment of the 15-pin Cannon connector in the front panel of the RS422-A/485 module:



Signal designation according to CCITT recommendation X.24

1) If you operate the interface in full duplex, data can only be received on this 2-wire line. In half duplex, data can be received and transmitted.

2) This signal can be connected either as an input or output (see jumper settings).

3) This signal can only be used when signal B is used as an output. In this case, the line X(A) is connected to the frame and not to the interface.

The position and use of the jumpers on the RS422-A/485 module can be found in Section 3.4.

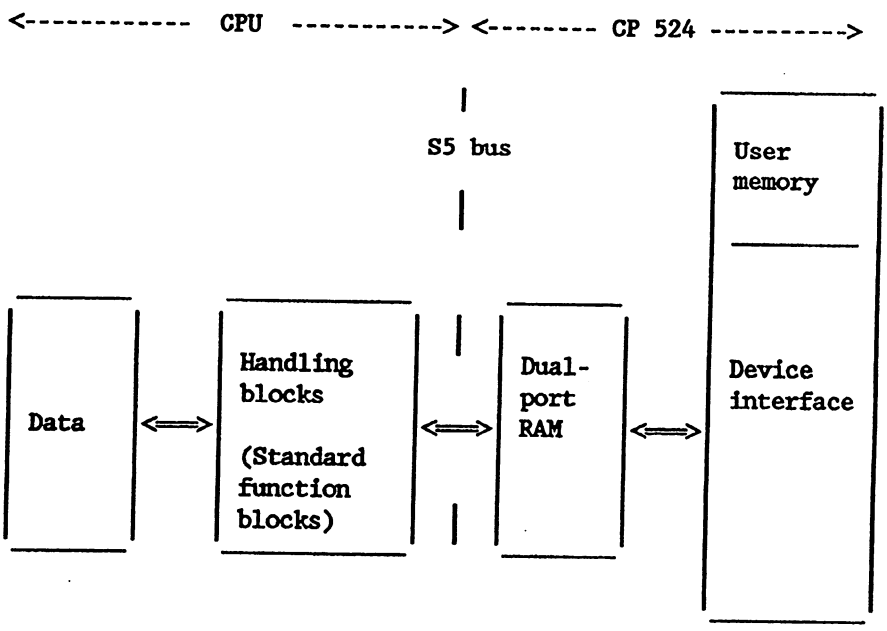
When programming your CP 524 user program using the COM 525 programming package, you select the required transmission parameters (transmission speed, character frame).

1.4 Dual-port RAM: Interface to the CPU

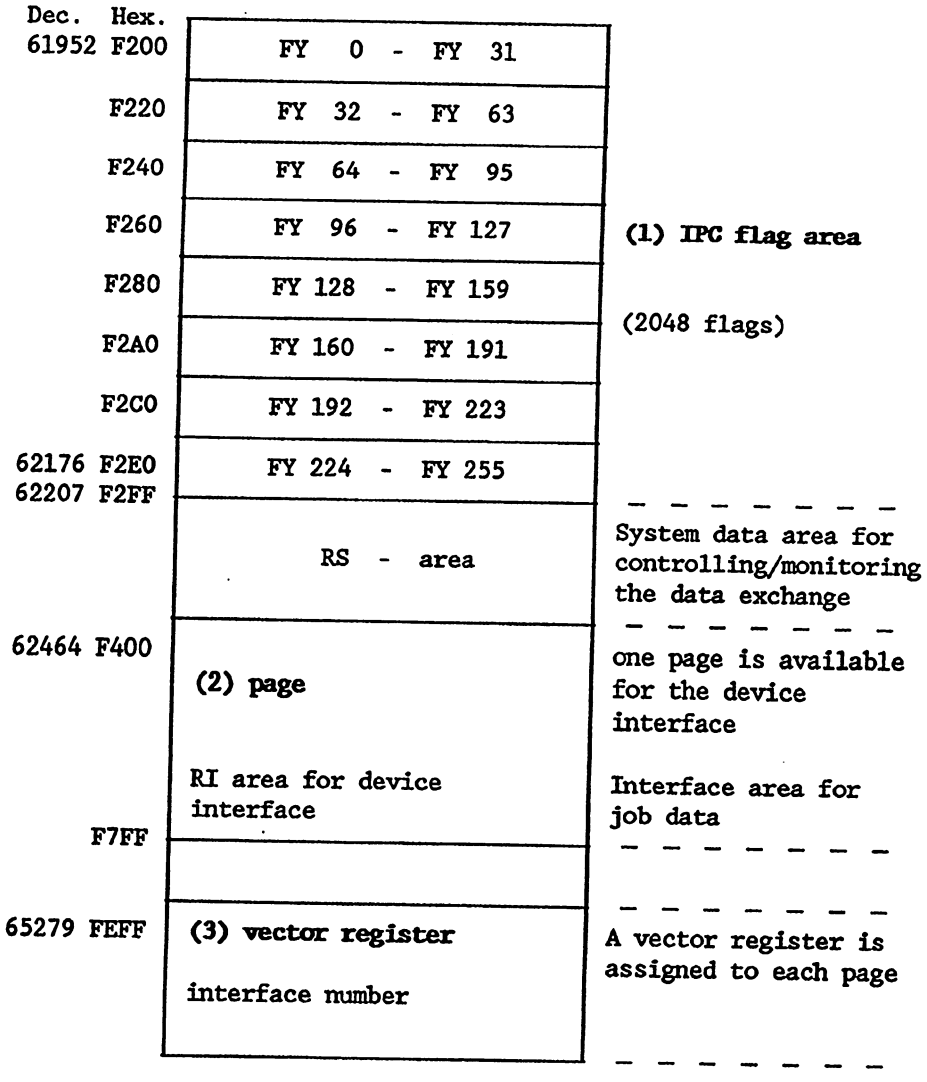
Data is exchanged between the CPU and the CP 524 via a common memory area: the **dual-port RAM**. The dual-port RAM is on the CP 524 and is similar to a mailbox. The CP 524 and CPU can leave messages (data, requests etc.) for each other in this mailbox.

The CPU of the PC always acts as master for data exchange. The CP 524 must "wait to be asked" by the CPU whether it wants to transfer data. This polling function is carried out by standard function blocks (**"handling blocks"**) in the CPU. These are the same handling blocks that are used by other communications processors.

The actual exchange of data between the CPU and CP 524 is also managed by handling blocks. For more details, refer to the description "Using the handling blocks" in this manual.



The **dual-port RAM** of the CP 524 is structured as follows:



This diagram is explained on the following page.

(1) Interprocessor communication (IPC) flag area

IPC flags are defined as one of the following:

- IPC flags may be part of the flag area of a CPU.
- If they are specified as IPC output bytes (in terms of the computer link), they are part of the dual-port RAM of the CP 524. These IPC flags are also called coordination flags.

IPC flags are needed because of the following:

- IPC flags are used for the cyclic exchange of data between CPUs in bytes.
- They can also be used for the data exchange between CPUs and the CP 524. The IPC flag area in the dual-port RAM of the CP 524 is responsible for this function.

The IPC flag area in the CP 524 covers 256 IPC flag bytes (2048 IPC flags).

With the computer link and RK 512, the IPC flags (coordination flags) can be used to **inhibit and enable the data exchange between the CPU and CP**. By inhibiting this exchange, you can prevent data that has not yet been evaluated from being overwritten. These IPC flags must be designated as output flag bytes. For further details, see the user's guide for the computer link with RK 512.

The IPC flags are not used, if you are working with the PT88/PT89 logging printer.

The IPC flags are transferred once per STEP 5 program cycle between the flag area of the CPU and the IPC flag area of the CP 524 (similar to the process image for digital inputs and outputs).

To prevent multiple addressing when you are using several CPs or CPs and a coordinator in one PC, you must divide the IPC flags into groups (jumper settings see 3.4.2). This also applies if there is more than one CPU in a PC.

Note:

S5-115U: the IPC flags must be listed in DB1 as IPC output flags.

S5-135U: the IPC flags must be entered as IPC output flags in the DB1 of each CPU. This also applies when only one CPU is being used along with a CP 524.

S5-150U: since the IPC flags are in a different address area, the function block FB TRANSFER handles the cyclic exchange of IPC flags between the CPU and CP. The IPC flags must be specified when the block is called. The block call must be at the end of the cycle of your STEP 5 program.

S5 155U: the IPC flags must be entered in DB1 of the CPU as output flags. This is also the case when only one CPU and one CP 524 are being used.

(2) Data transfer area (page)

The device interface of the CP 524 has a page assigned to it. The page is the area within the dual-port RAM in which the **data** are exchanged between the CP 524 and CPU.

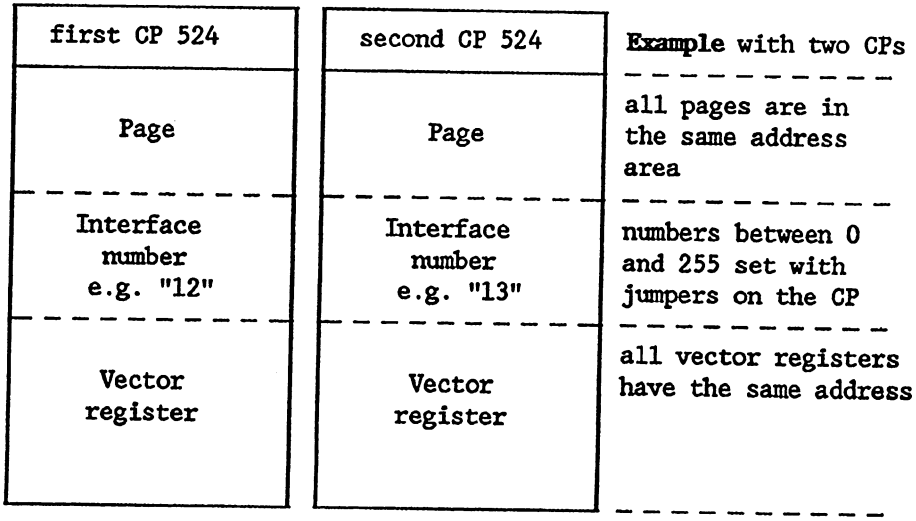
(3) Vector register - interface number

The pages of all the CPs in a PC are in the same address area. To ensure that only the required page is addressed when data is exchanged, each page can only be addressed by using its **interface number**. This is between 0 and 255 and is set on the CP by means of jumpers.

The page has a **vector register** allocated to it. The vector registers of all the CPs have the same address.

If the page of a particular CP is to be addressed by the CPU (e.g., to transfer data to the CP 524) the CPU writes the interface number of the required page in all the **vector registers** available. Each CP compares the contents of the vector register

with its own interface number. If the CP recognizes that the number which has been entered matches its interface number the required page is enabled for data transfer. This function is carried out automatically by data handling blocks.



1.5 Memory Submodules

The memory submodule of the CP 524 contains your **CP 524 user program**, which includes the following:

- the specific programs for the device interface ("interpreter and procedure"). These programs determine which function (computer link or printer) is performed.

- the "actual" user program, i.e., the messages for the printer and/or telegram parameters for the computer link.

The interpreter and procedure require a total of 8 Kwords in the memory module. The CP also sets up a manager for the user data in the module. This manager is dynamic, i.e. it grows with the number of user data. It can occupy a maximum of 4 Kwords.

For the computer link with RK 512 a maximum of 1 Kword is required.
To operate a computer link a **16 Kword** EPROM module is required.

For the PT 88/PT 89, the required memory space depends on the size of your user progam.

Only **EPROM submodules** can be used in the CP 524, since (in contrast to the CP 525) there is no interface to the programmer required to load the software in the RAM.

The following memory submodules are available for the CP 524:

EPROM submodules

16 Kwords 32 Kwords 64 Kwords	When programming this submodule on the PG you must use an adapter (MEP adapter), since a different programming voltage is required for EPROMs of the type 27256.

The order numbers for these submodules can be found in the ordering data section of this manual.

1.6 Technical Data

Degree of protection according to DIN 40050 and IEC 144	IP00
Perm. ambient temperature	
during operation	0°C to 55°C
during transport and storage	-40°C to +70°C
Humidity class (DIN 40040)	code letter F
Operating altitude	up to 3,500 m above sea level
Power supply	5 V +/- 5%; 24 V +25%/-15%
Current input 5 V	1.5 A max.
CP 524	0.8 A, measured statically
TTY module	0.1 A, measured statically
V.24 module	0.2 A, measured statically
RS422-A/485 module	0.5 A, measured statically
Current input 24 V	
TTY module	60 mA
Transmission procedure	asynchronous, exchangeable procedures
Transmission rate	110 to 19200 bps with TTY max. 9600 bps
Transmission cable	shielded 4-wire cable with braided shield and metal connector: earthing required both ends
Transmission link TTY	max. 1 km at 9600 bps
Transmission link V24	max. 20 m
Transmission link RS422/A-485	max. 1200 m

Design	
dimensions (W x H x D)	20.32 mm x 160 mm x 233.4 mm
backplane connector (packaging system)	ES 902 / row 2, 48-pin
Weight	approx. 0.4 kg

2 Installation

2.1 Slots in the Programmable Controller

The CP 524 can be plugged into the following programmable controllers (fan required):

S5-115U with subrack CR 700-2
Slot number

		0	1	2	3	4	5		

S5-115U with subrack ER 701-3
Slot number

	0	1	2	3	4	5	6		

The interface modules 304 and 314 are required

S5-115U with subrack CR 700-3
Slot number

		0	1	2	3	4	5		

The interface modules 304 and 314 are required

S5-135U
Slot number

	11	19	27	35	43	51	59	67		

With the CP 524, *no* coordination processor is required if only one CPU is used.

S5-150U
Slot number

3	11	19		107	115	123	131	139	147	

S5-155U
Slot number

	19	27	35	43	51			115	123	131	139	147	

Further details about your programmable controller can be found in the appropriate manual.

The CP 524 can be plugged into the following expansion unit (fan required):

EU 185U
Slot number

19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139

The interface modules 304 and 314 are required.

2.2 Plugging in and Removing the CP 524

The CP 524 must not be removed or plugged in when the power is on! This also applies when replacing the submodules!

2.3 Installation Guidelines

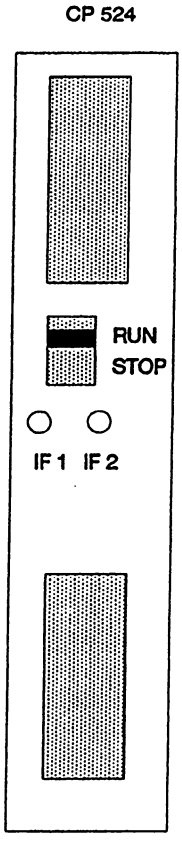
The connecting cable between the CP 524 and a partner device must be a cable with a shield earthed at both ends and with metal connectors. The standard cable connectors listed in the catalog and in Section 6 meet these requirements.

The shield must make large area contact with the casing of the metal connector (not on pin 7). If the partner device does not have a metal connector, the cable shield must make large area contact with ground. If there is a large difference in ground potential between the CP 524 and the partner device, a potential equalizing cable should be included. The cable should be kept away from interfering a.c. power controller cables, thyristor drives and power cables carrying more than 500 V.

Further information about installation can be found in the "Installation guidelines for SIMATIC compact devices" (see ordering data).

3 Operation

3.1 Controls and Displays



3.2 Operating Modes

Mode selector

RUN	The processor runs in the normal operating mode, the user functions which have been programmed are processed. The LEDs are not lit (if LEDs are lit, see 3.3)
STOP	No jobs are processed. The LED IF1 is lit.

3.3 Status Displays

If the LED IF1 is lit, one of the following has occurred:

- device driver and procedure not loaded
- mode selector switched to STOP
- CP 524 still in start-up phase following a power failure (takes max. 10 s)
- serious error recognized during start-up (e.g., incorrect memory submodule - submodule type not permitted).

If the LED IF2 is lit, one of the following has occurred:

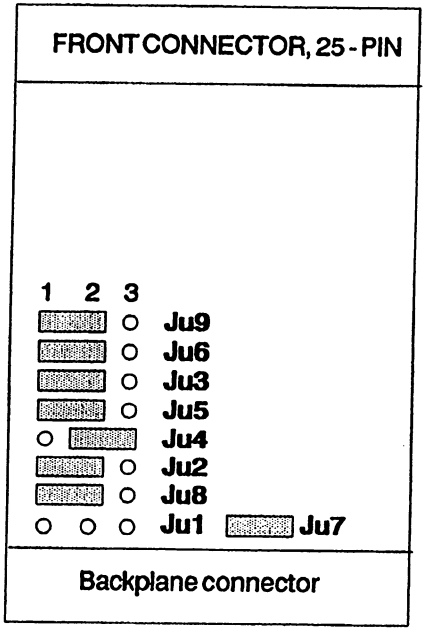
- serious error recognized during start-up (e.g., incorrect memory submodule - submodule type not permitted)
- CP 524 still in start-up phase following a power failure (takes max. 10 s)

3.4 Jumper Settings

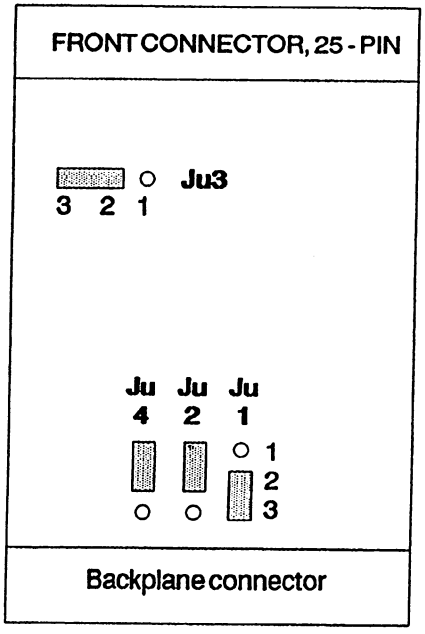
The jumpers on the interface modules (V.24, TTY and RS422/A-485) and on the CP 524 are preset in the factory. You can use this setting to operate the CP 524 and the interface modules immediately (if only one communications processor is used in the PC). The interface number is then 0. All the IPC flags are enabled.

To change the interface number (module address) on the CP 524 or enable/inhibit IPC flags, make the appropriate jumper settings. For more information refer to 3.4.1 and 3.4.2.

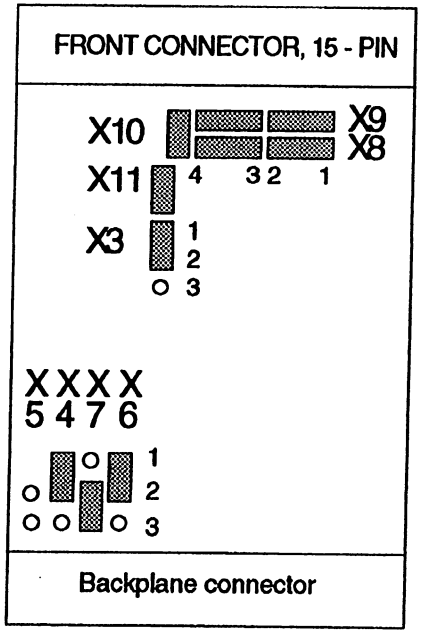
V.24 Module:
Jumpersettingsassupplied



TTY-Module:
Jumpersettingsassupplied

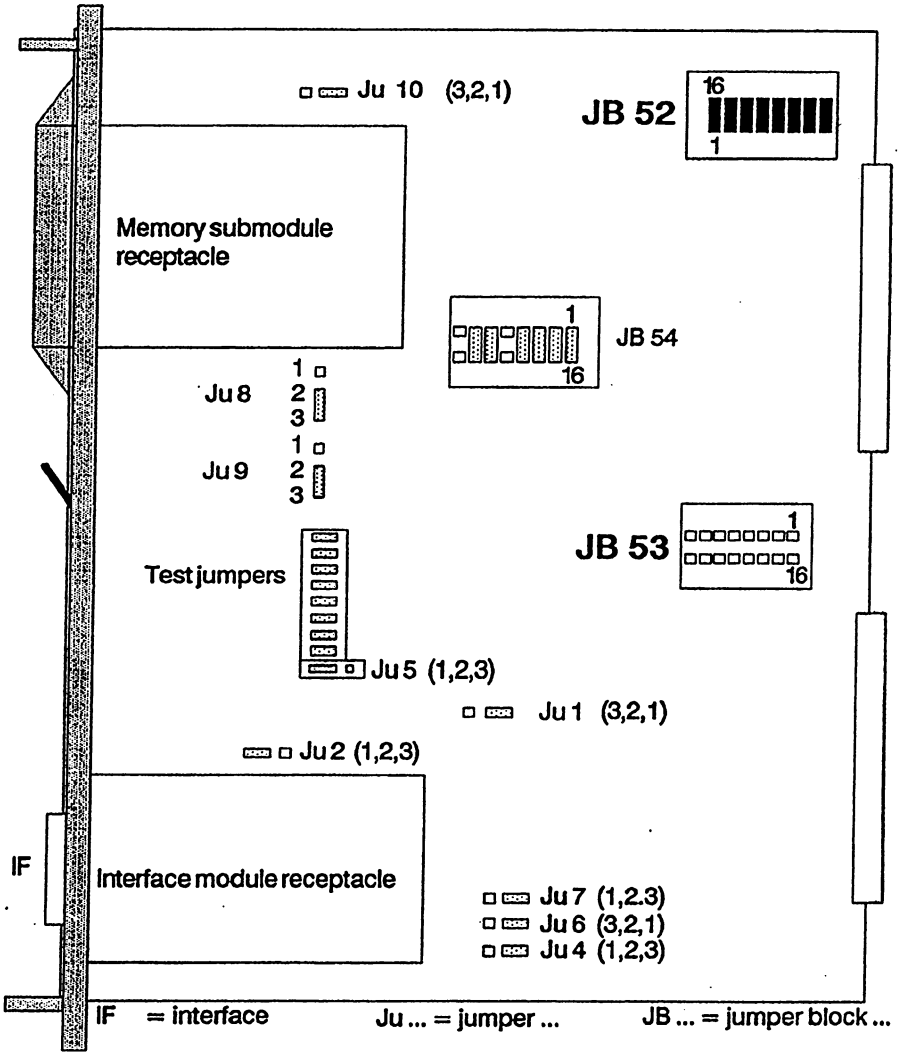


RS422-A/485-Module:
Jumper settings as supplied



5

CP 524: factory jumper settings



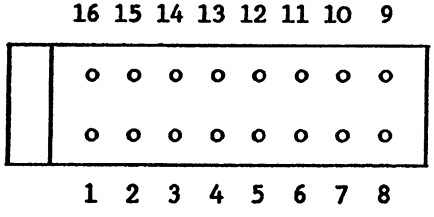
3.4.1 Module Address/Interface Number

As described in 1.4 the data is transferred via the pages of the dual-port RAM. The page is assigned an interface number ranging from 0 to 255. With the CP 524, the interface number is identical with the module address.

The module address (interface number) is set in binary on the CP 524 at jumper block **JB53**.

Note: to ensure that the CPU addresses the CP correctly, you must specify the same interface number (parameter SSNR) when assigning parameters to the handling blocks that is set on the CP.

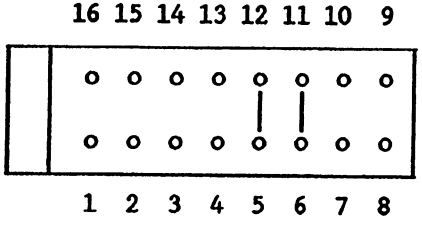
Jumper block J53:



Jumper	Value	
8-9	1	2^0
7-10	2	2^1
6-11	4	2^2
5-12	8	2^3
4-13	16	2^4
3-14	32	2^5
2-15	64	2^6
1-16	128	2^7

Jumper open: $0 \times 2^n = 0$
Jumper inserted: 1×2^n

Example: selected interface number equals 12.



12 decimal = $0x2^0+0x2^1+1x2^2+1x2^3+0x2^4+0x2^5+0x2^6+0x2^7$

Jumpers 6-11 and 5-12 are inserted, the other jumpers remain open.

3.4.2 Enabling/Disabling IPC Flags

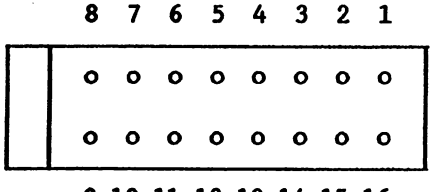
As described in 1.4 and in the user's guide "Computer link with RK 512" you can enable or inhibit the data transfer between the CPU and CP 524 by using the IPC flags. These IPC flags must be designated as output flag bytes. In this situation, the term "coordination flag" can also be used instead of IPC flag.

A total of 256 IPC flag bytes are available (2048 IPC flags).

No IPC flags are used with the PT 88/89 printer.

In multiprocessor operation, IPC flags allow the exchange of data between the processors. If several CP 524s or several CPUs (in the S5-135U) are installed in the programmable controller, the IPC flags must be divided up among the CPs and the coordinator (see also instructions for the S5-135U). The IPC flags must then be enabled on the modules by means of jumpers. On the CP 524 there are eight jumpers available at location JB52. By appropriate jumper setting at this location, 32 contiguous IPC flag bytes can be enabled or disabled.

Jumper block JB52:



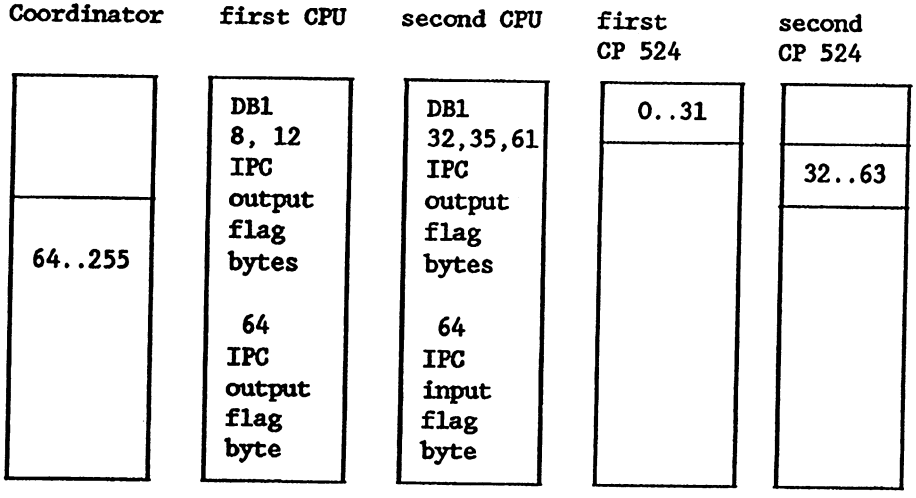
Jumper	IPC flag byte
8- 9	0 ... 31
7-10	32 ... 63
6-11	64 ... 95
5-12	96 .. 127
4-13	128 .. 159
3-14	160 .. 191
2-15	192 .. 223
1-16	224 .. 255

Jumper inserted: the corresponding IPC flag bytes are enabled.

5

!IMPORTANT!: the corresponding IPC flag bytes must also be entered in the DBls of the CPUs in the S5-135U and 115U! These can then be addressed as flags in the STEP 5 program.

The following example shows the distribution of IPC flags between two CP 524s and one coordinator with the S5-135U:



This example specifies that the first CPU "coordinates" the data exchange with the first CP 524 via one (or several) bits of the IPC flag bytes 8 and 12. Therefore the area 0 to 31 is enabled on the first CP 524. (An area of 32 contiguous IPC flags must always be enabled or disabled!). Jumper 8-9 is inserted on JB 52. In DB1 of the first CPU, you must enter bytes 8 and 12 as IPC output flag bytes.

In the same way the second CPU coordinates the data exchange with the second CP 524 via IPC flags 32, 35, 61. On the second CP 524, area 32 to 63 must therefore be enabled (jumper 7-10 inserted). You must then enter bytes 32, 35 and 61 as IPC output flags in DB1 of the second CPU.

Since the IPC flag area exists only once in each PC, only IPC flags 64 to 255 can be used for other purposes. This example uses IPC flag 64 for CPU - CPU communication (IPC output flag in DB1 of the first CPU - IPC input flag in DB1 of the second CPU). The area which contains byte 64 must therefore also be enabled in the coordinator.

3.4.3 Jumpers on the V.24 Module

Jumpers **Ju1** and **Ju2** determine whether the internal clock pulse of the CP 524 is used as the transmit and receive clock pulse on the CP 524 or whether the pulse is supplied by the line

	1 2 3	
Ju1	o—o o	Receive clock pulse internal
	o o—o	Receive clock pulse from line
Ju8	o—o o	Receive clock pulse = transmit pulse
	o o—o	Receive clock pulse depends on Ju1
Ju2	o—o o	Transmit clock pulse internal
	o o—o	Transmit clock pulse from line
Ju4	o—o o	Transmit clock pulse of CP 524 connected to line 113 (pin 24)
	o o—o	Transmit clock pulse not supplied by CP 524

5

With jumpers **Ju3** and **Ju5**, the polarity of the transmitted and received data can be reversed as follows:

	1 2 3	
Ju3	o—o o	Transmitted data with normal polarity
	o o—o	Transmitted data negated
Ju5	o—o o	Received data with normal polarity
	o o—o	Received data negated

With jumper **Ju6**, all V.24 receivers can be connected so that all the signals are in the positive voltage range as follows:

	1 2 3	
Ju6	o—o o	All received signals must have the V.24 signal level.
	o o—o	All received signals can be in the positive voltage range

With jumper **Ju9**, CTS can be permanently at open circuit potential or switched through from the front plug as follows:

	1	2	3	
Ju9	o——	o	o	CTS at open circuit potential
	o	o——	o	CTS at pin 5

3.4.4 Jumpers on the TTY Module

With jumpers **Ju1** and **Ju2**, the polarity of the transmitted or received data can be reversed as follows:

	1	2	3	
Ju1	o——	o	o	Transmitted data negated
	o	o——	o	Transmitted data with normal polarity
Ju2	o——	o	o	Received data with normal polarity
	o	o——	o	Received data negated

With jumper **Ju3**, the 24 V source voltage for generating the current loop can be reconnected as follows:

Ju3	o——	o	o	24 V supplied by pin 9 of the front connector
	o	o——	o	24 V supplied by backplane connector (internally)
Ju4	o——	o	o	Receive clock pulse from transmit pulse
	o	o——	o	Receive clock pulse supplied by CP 524.

3.4.5 Jumpers on the RS422-A/485 module

The jumper **X3** is used to switch over the data direction on the two-wire line R.

	1 2 3	
X3	o—o o	Setting for full duplex mode. Data can only be received on the two-wire line R.
X3	o o—o	Setting for half duplex mode. With the signal /RTS, data can either be transmitted or received on the two-wire line R. /RTS = 0 --> transmit /RTS = 1 --> receive

Using the **jumper X4** you can apply either the signal /PS3 or the internal transmit clock pulse (TxCint) to the two-wire line B, providing that the two-wire output B is set as an input by jumpers X8 and X9.

	1 2 3	
X4	o—o o	Signal /PS3 can be transmitted on the two-wire line B.
X4	o o—o	The internal transmit clock pulse TxCint can be sent on the two-wire line B.

With **jumper X5** you can use a clock pulse transmitted on the two-wire line S as the receive clock pulse.

X5	o o	Clock pulse at input S is not used as receive clock pulse.
X5	o—o	Clock pulse at input S is used as the receive clock pulse.

With **jumpers X6 and X7** you can select the transmit and receive clock pulse.

	1 2 3	
X6	o—o o	The internal transmit clock pulse
X7	o o—o	(TxCint) is used as transmit and receive clock pulse.
X6	o o—o	The internal transmit clock pulse
X7	o o—o	(TxCint) is used as the transmit clock pulse and is used as the receive clock pulse.
X6	o—o o	The clock pulse transmitted via the
X7	o—o o	two-wire line S is used as the transmit clock pulse and the internal receive clock pulse used as the receive clock pulse.
X6	o o—o	The clock pulse transmitted via the
X7	o—o o	two-wire line S is used as the transmit clock pulse, the internal receive clock pulse (RxCint) is used as the receive clock pulse.

The **jumpers X4, X5, X6 and X7** can be used for synchronous transmission with clock pulse control according to the DIN draft ISO 8481. The two-wire line B (designation X according to ISO 8481) must be set as an output by jumpers X8 and X9.

	1 2 3	
X4	o o—o	The internal transmit clock pulse
X5	o—o o	(TxCint) is used as the transmit
X6	o o o	clock pulse. The internal clock
X7	o o—o	pulse is simultaneously transmitted on the two-wire line B.

With **jumpers X8 and X9** you can set the two-wire line B as an input or ouptut.

	1 2 3 4	
X8	o—o o—o	Two-wire line B is connected as
X9	o—o o—o	an output. The line X(B) can be used as an input. X(A) is connected to frame.
X8	o o—o o	Two-wire line B is connected as
X9	o o—o o	an input. The line X(B) cannot be used.

With **jumpers X10 and X11** you can suppress the control signal on the two-wire line R for recognizing the break status.

5

X10	o o	No control signal on the two-wire
X11	o o	line R - recognition of the break status cannot be guaranteed.
X10	o—o	Two-wire line R has the control
X11	o—o	signal, the break status will be recognized.

4 Setting up the Hardware

When setting up the CP 524 module follow the checklist below:

1. Check the jumper setting on the module according to 3.4.

2. Insert a programmed EPROM submodule in the receptacle.

3. Insert an interface module in the receptacle. Check the jumper setting according to 3.4. The interface module must be secured by the two screws.

4. Plug the CP 524 module into one of the permitted slots in your device (see 2.1). The power supply must be switched off.

5. Connect the partner device to the CP 524 (see Section 6).

6. Check the position of the mode selector on the CP 524. It must be switched to "RUN", to data exchange with the partner device.

7. Make the correct settings at the partner device. These settings can be found in the user's guide "Computer link with RK 512" and "Event output and logging with the PT 88/PT 89 printer" in the sections describing installation. Make sure that the parameters required for transmission (data rate, parity, stop bit, character length) are the same in both the partner device and the CP 524.

8. Switch on the power supply to your PC. Both LED displays on the front of the CP 524 light up. The operating system carries out a self-test. The LEDs go out if no error has been detected and if the device driver and transmission procedure for the serial device interface are present in the EPROM.

5 Pin Assignments

Backplane connector 1

	d	b	z
2		0 V	+ 5 V
4			
6	ADB 12	ADB 0	$\overline{\text{CPKL}}$
8	ADB 13	ADB 1	$\overline{\text{MEMR}}$
10	ADB 14	ADB 2	$\overline{\text{MEMW}}$
12	ADB 15	ADB 3	$\overline{\text{RDY}}$
14		ADB 4	DB 0
16		ADB 5	DB 1
18		ADB 6	DB 2
20		ADB 7	DB 3
22		ADB 8	DB 4
24		ADB 9	DB 5
26		ADB 10	DB 6
28		ADB 11	DB 7
30			
32		0 V	

Backplane connector 2

	d	b	z
2		0 V	+ 5 V
4			
6			
8			
10			
12			
14			
16			
18			
20			
22		$\overline{\text{STOPPA}}$	
24			
26			
28			
30		M 24 V	M 24 V
32		0 V	+ 24 V

5

Memory submodule connector

	c	b	a
1	SADB 12	M	+ 5 V
2	SADB 0	SADB 1	SADB 2
3	SADB 3	SADB 4	SADB 5
4	SADB 6	SADB 7	SADB 8
5	SADB 9	SADB 10	SADB 11
6	SADB 13	SADB 14	OE
7	WRITE	SDB 8	SDB 9
8	SDB 10	SDB 11	SDB 12
9	SDB 13	SDB 14	SDB 15
10	SDB 0	SDB 1	SDB 2
11	SDB 3	SDB 4	SDB 5
12	SDB 6	SDB 7	K 1
13	CS 1	CS 3	K 2
14	CS 2	CS 4	K 3
15	UCMOS +5 V	PSW	K 4
16	+ 5 V	M	K 5

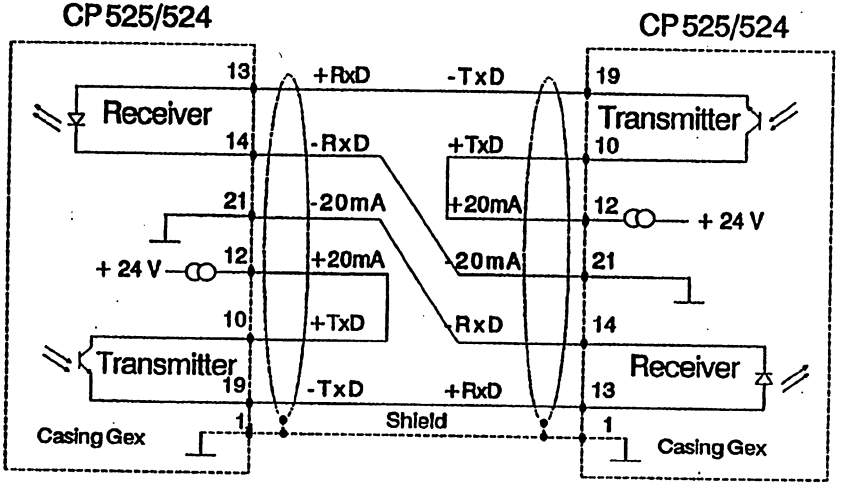
Interface module connector

	c	b	a
1			
2	M	M	M
3	+ 5 V	+ 5 V	+ 5 V
4	SSK 1	PM 1	RxD
5	SSK 2	PS 3	TxD
6	SSK 3	HUE	CTS
7			RTS
8			DSR
9			DTR
10			DCD
11			RI
12			TxC
13			TxC Int.
14		RxD'	RxC
15	M 24 V	M 24 V	RxC Int.
16	+ 24 V	+ 24 V	+ 24 V

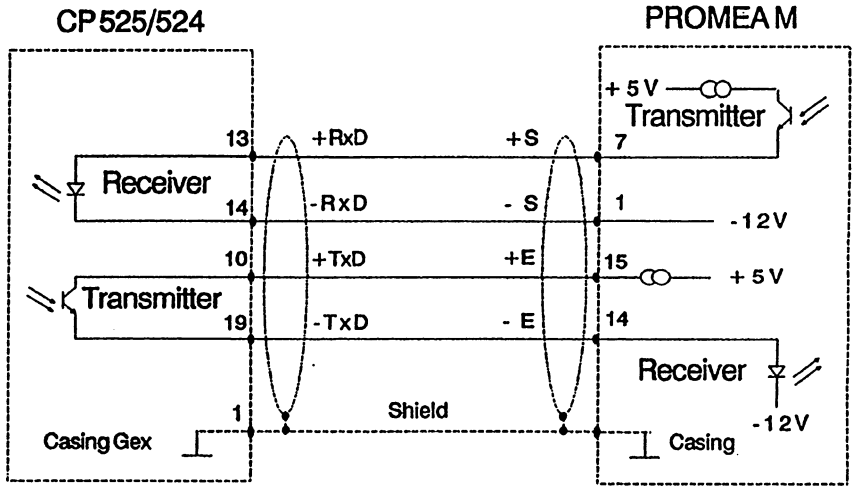
6 Standard Cable Connectors

For the most common connections between devices there are standard cable connectors available (see ordering data).

6.1 Connection CP 524/525 - CP 524/525 (TTY)

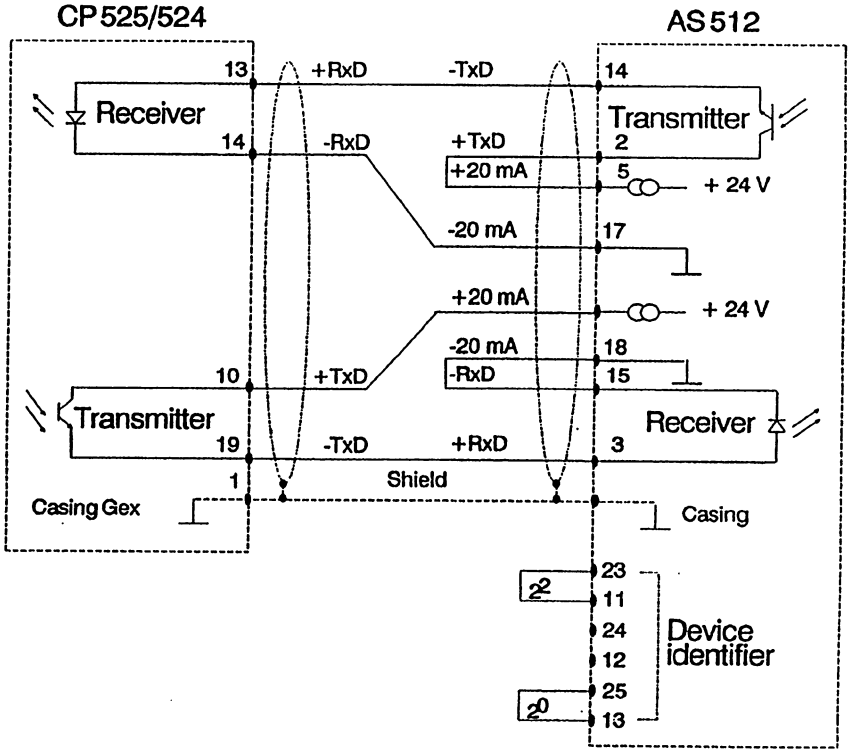


6.2 Connection CP 524/525 - PROMEA M (TTY)



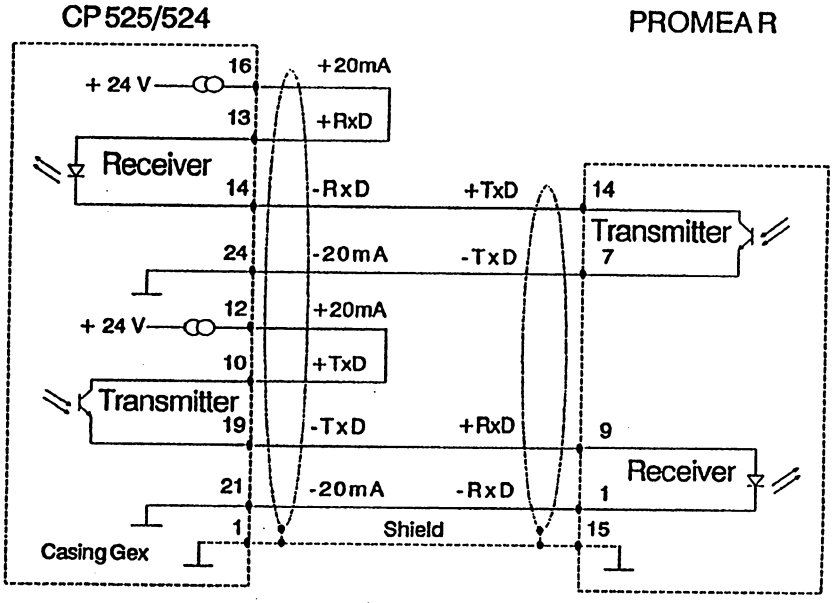
5

6.3 Connection CP 524/525 - AS 512C (TTY)

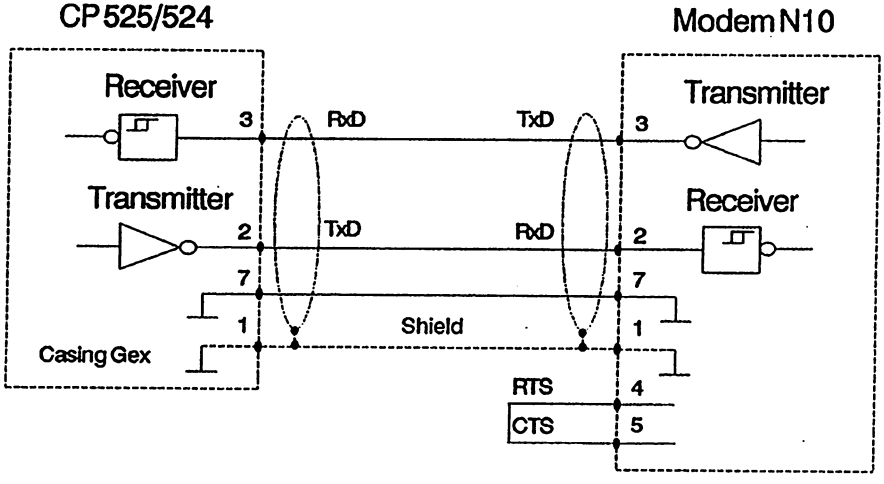


Note: to generate a current loop the AS 512 must be supplied with 24 V via the AMP connectors on the front panel.

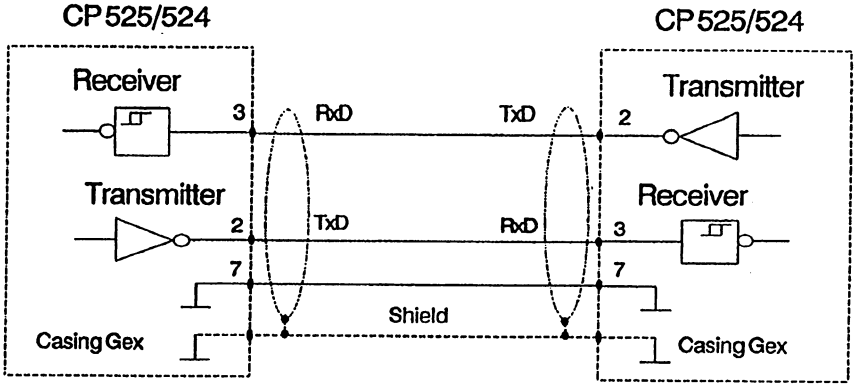
6.4 Connection CP 524/525 - PROMEA R (TTY)



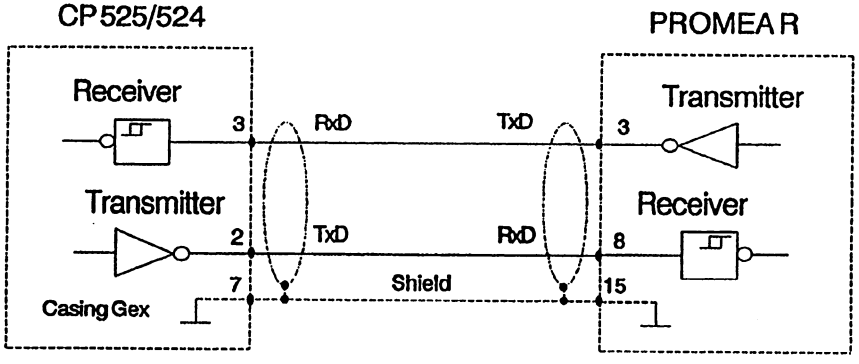
6.5 Connection CP 524/525 - Modem N10 (V.24)



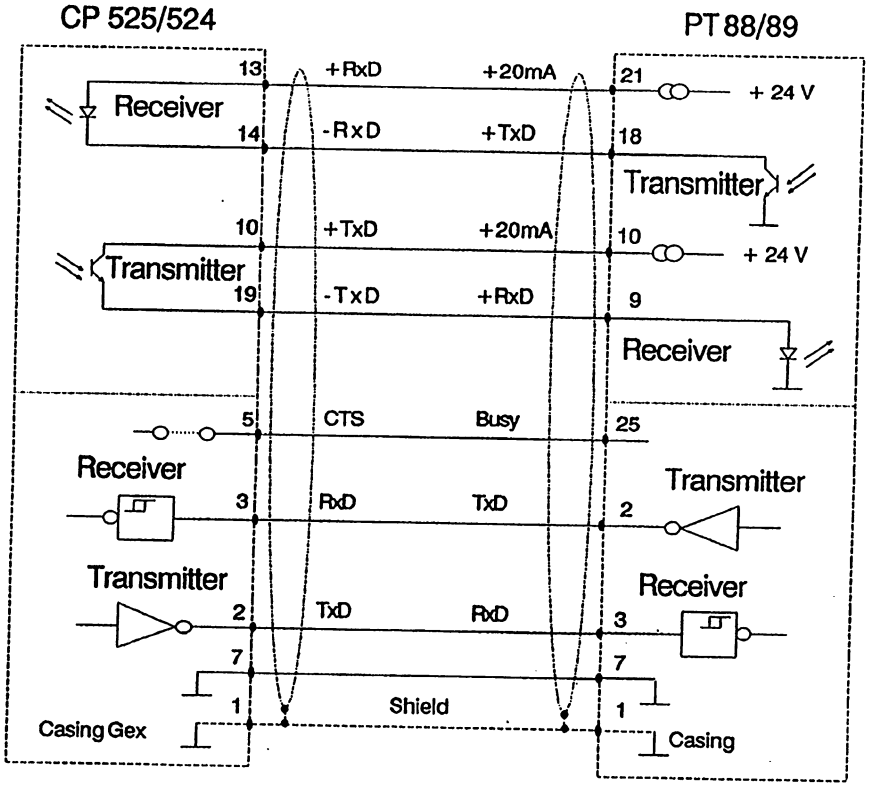
6.6 Connection CP 524/525 - CP 524/525 (V.24)



6.7 Connection CP 524/525 - PROMEA R (V.24)



6.8 Connection CP 524/525 - PT 88/PT 89 (TTY/V.24)

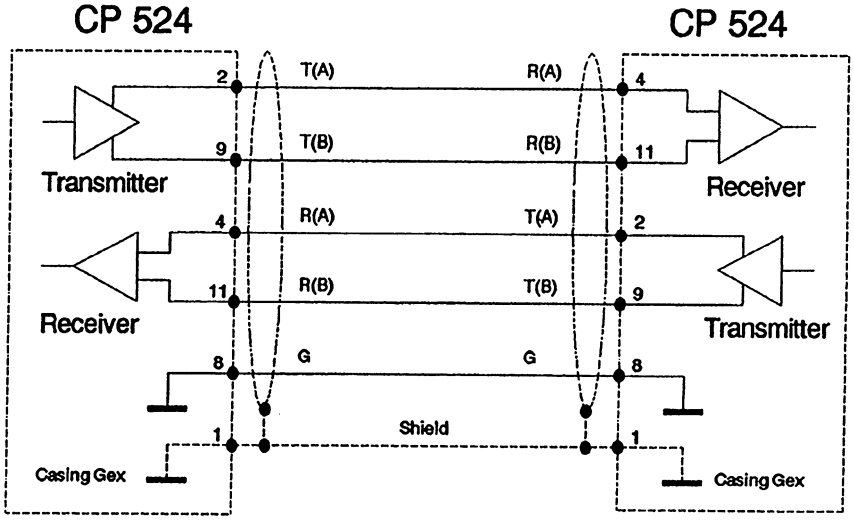


Note: The cable is designed for both TTY and V.24 signals. It can therefore be used regardless of the type of interface used in the printer.

With the CP 524, make sure that you use the same interface type as in the printer.

5

6.9 Connection CP 524 - CP 524 (RS422-A or V.11)



Siemens Aktiengesellschaft

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 model or design, are reserved.

© Siemens AG 1988

Order No.: C79000-B8576-C538-05
Order from: Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany
AG 0689 0.5 AB 54 En

SIEMENS

SIMATIC S5

Computer Link with RK 512

User's Guide C79000-B8576-C539-08

Contents Page

1 Functions 1

1.1 Transmitting Data (SEND Job) 5
1.2 Fetching Data (FETCH Job) 7

2 Programming the Functions 9

2.1 User Memory of the CP 525 12
2.1.1 Memory Requirements of the Job Block 12
2.2 Assigning Parameters to the Job Block for the CP 525 13
2.3 Assigning Parameters to the Interpreter and Procedure 17
2.4 Parameters for the Handling Blocks 18
2.5 Example of a Complete Parameter Set 21
2.6 Job Tables 26

3 Settings and Start-up 38

3.1 Settings on the CP 525 38
3.1.1 Jumper Settings 38
3.1.2 Setting Interpreters and Procedures 38
3.2 Settings on the Partner 38
3.3 Start-up Routine 39
3.3.1 Routine Following Power-up 39
3.3.2 SYNCHRON Job from the CPU 39
3.3.3 Mode Selector Settings RUN/STOP/PGR 41
3.3.4 Transferring from the PG 42

4 How Is Data Transferred? 44

4.1 Telegram Level 45
4.1.1 Sending Data 48
4.1.2 Fetching Data 51
4.2 Quasi Full Duplex Operation 54

7

5 Procedures 55

5.1 Procedure 3964 55
5.1.1 Procedure Data 55
5.1.2 Transmitting with the 3964 Procedure 57
5.1.3 Receiving with the 3964 Procedure 59
5.1.4 Initialization Conflict 60
5.1.5 Procedure Errors 61
5.2 3964R Procedure 63
5.2.1 Procedure Data 63
5.2.2 Transmitting with the 3964R Procedure 65
5.2.3 Receiving with the 3964R Procedure 67
5.2.4 Initialization Conflict 68
5.2.5 Procedure Errors 69

6 Multiprocessor Operation 71

7 Special Jobs Carried out by the CPU 72

7.1 PSEUDO READ/WRITE Function 72
7.2 Reading the Error Message Area of the SYSTAT 81
7.3 Resetting (Clearing) the Error Message Area 85
7.4 Reading the Whole SYSTAT 85
7.5 Reading the SYSID 87
7.6 CP 525: Reading and Writing the Date and Time 90

8 Error Handling 94

8.1 Error Messages for SYSTAT, ANZW and REPTEL Arranged According to the Numbers for ANZW 100
8.2 Error Messages for SYSTAT, ANZW and REPTEL Arranged According to the Numbers for SYSTAT 124
8.3 Error Messages for SYSTAT and REPTEL Arranged According to the Numbers for REPTEL 162

9 Time Measurements - Processing a DIRECT Job 176

1 Functions

In automation engineering **data** must often be **exchanged**

- **between programmable controllers**
  or
- **between a programmable controller and a central process computer**

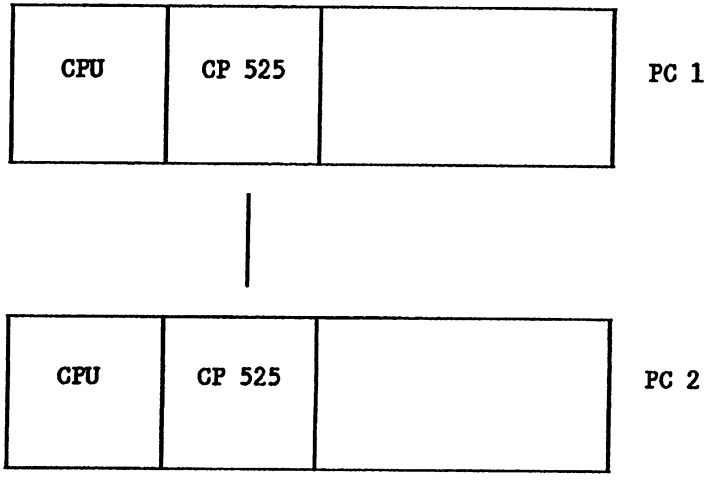
The data exchange can be implemented using the communications processor CP 525 with its interpreter RK 512 and the procedures 3964 or 3964R.
This means that you can link the following programmable controllers:

- the programmable controllers of the SIMATIC S5 U range (115U, 135U, 150U, 155U) with each other
- the programmable controllers S5-110S, S5-130W, S5-150A/K, S5-150S (if they are equipped with an AS 512C module) with the PCs, S5-115U, S5-135U, S5-150U and S5 155U

Note that the CP 525 cannot be used with the S5-135U/S processor.

The RK 512 interpreter uses the same message format as the interface module AS 512C in SIMATIC S5.

The data is exchanged via a point-to-point link as shown in the following diagram.



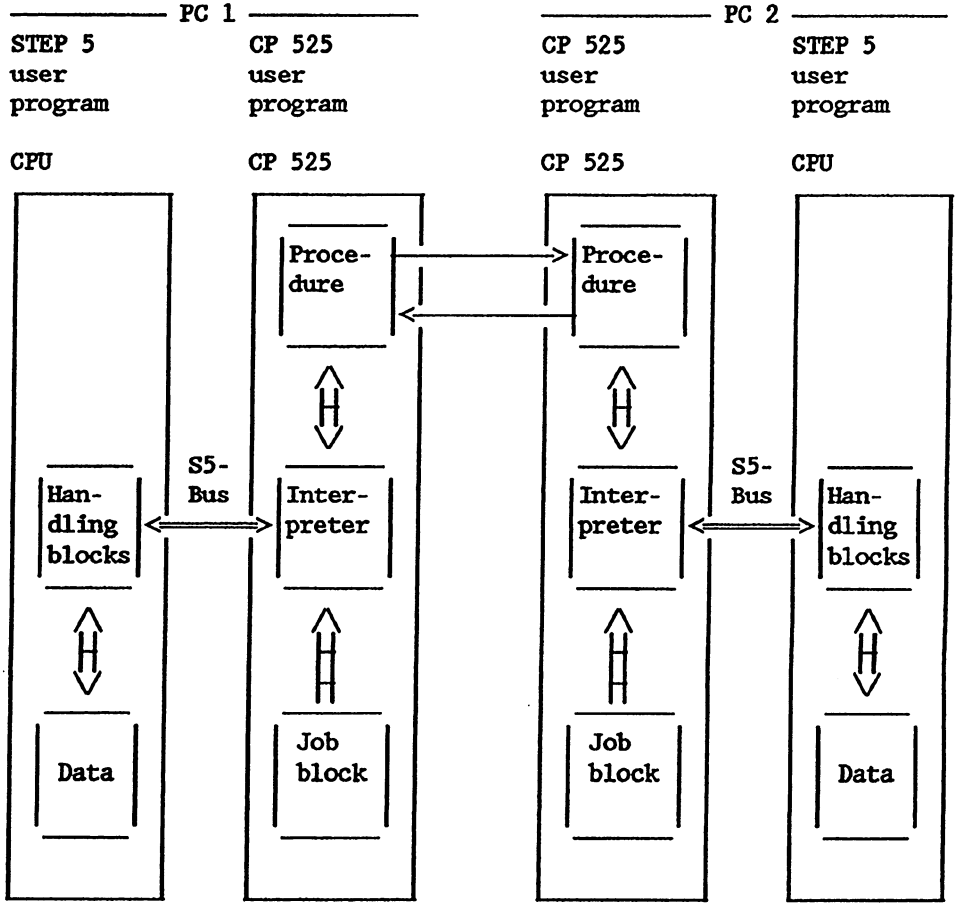
The exchange of data can be carried out in two ways as follows:

- Programmable controller PC 1 can **transmit data** on its own "initiative" to programmable controller PC 2. In this case CPU 1 must call a **"SEND job"** on its "own" CP 525.
- PC 1 can **fetch data** on its own "initiative" from PC 2. In this case CPU 1 must call a **"FETCH job"** on its "own" CP 525.

In the same way, PC 2 can also transmit data to or fetch data from PC 1.

Remember that per CP interface a maximum of **10 jobs (SEND and FETCH jobs)** can be processed "simultaneously". A new job can be started only when one of the 10 jobs has been terminated with or without errors.

The interaction during data exchange between programmable controller 1 and 2 is shown in the following diagram.



Data handling blocks (HDBs) in the STEP 5 user program on the CPU transfer the data from the CPU to the CP 525 and vice-versa. HDBs also initiate the jobs to be executed - a SEND job (send data) or a FETCH job (fetch data). The HDB is assigned a **job number** under which the corresponding **job** is stored in the job block on the CP 525. All the jobs together are known as the job block.

7

The **job block** on the CP 525 contains all specifications regarding the destination or source of the data in the **other PC.**

The **data handling block** in the CPU contains all the specifications regarding the source or destination of the data in **its own PC.**

To make use of the computer link, you require two programs for each programmable controller as follows:

- the **STEP 5 user program** in the CPU
  including the HDBs with parameters assigned according to their jobs

  and

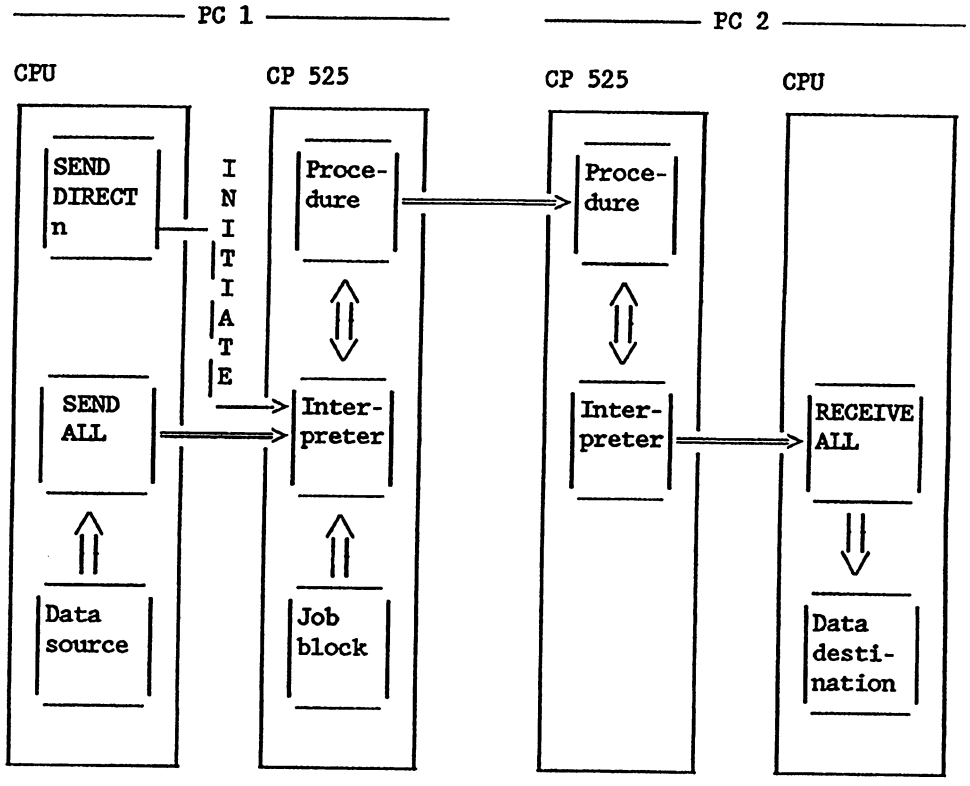
- the **CP 525 user program** in the CP 525
  including the jobs in the job block and the procedure and interpreter
  Interpreter: manages the data exchange
  Procedure: implements the transfer

The programming package **COM 525** supports you when you are generating the CP 525 user program.

1.1 Transmitting Data (SEND Job)

The CPU of PC 1 is to send data on its own initiative to the CPU of PC 2 (SEND job).

SEND job



n = job number

7

First the HDB **SEND DIRECT n** (n = job number) is called in the CPU of PC 1. This initiates the call-up of the data from the data source in the CPU of PC 1 by the interpreter of the CP (PC 1). The SEND DIRECT requires the following parameters to be specified:

- interface number of the CP 525 in PC 1
- the number of the corresponding SEND job on the CP 525 (PC 1)
- details of the data source (where the data to be sent is stored in the CPU of PC 1)

The data transfer from the CPU to the CP 525 in PC 1 is handled by the HDB **SEND ALL**.

The following parameters are assigned to the SEND job in the job block on the CP 525 of PC 1:

- the number of the corresponding SEND DIRECT in the CPU
- details of the data destination: i.e., where the data is to be stored in the receiver (PC 2)

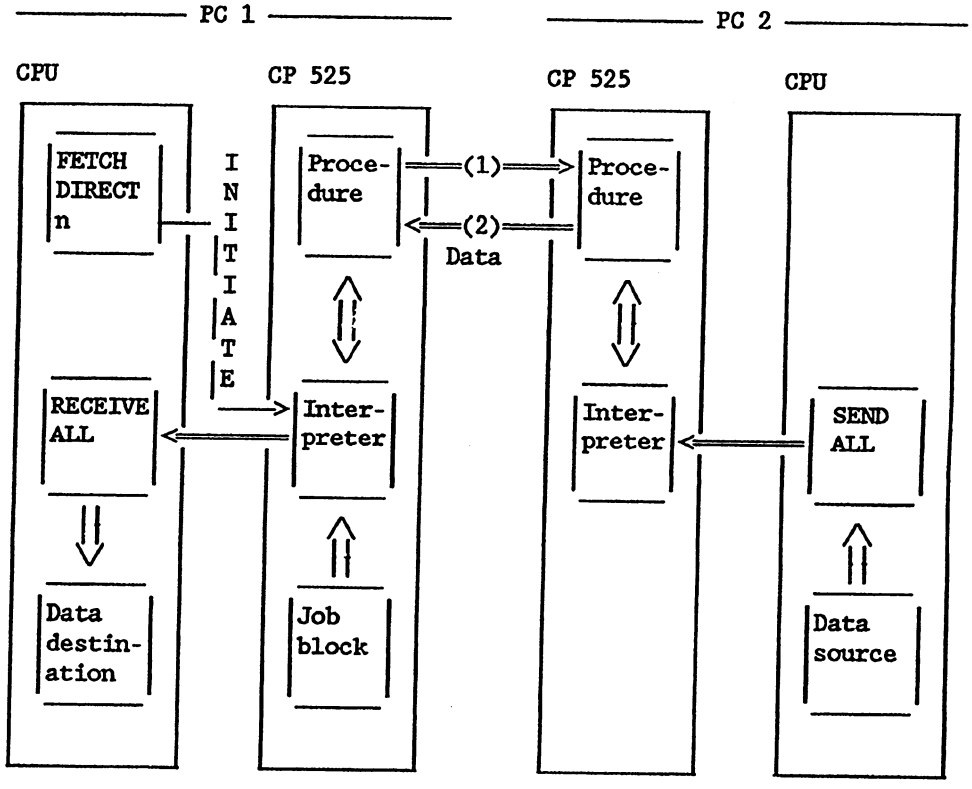
This information is sent to the receiver PC 2 along with the data.

No job block is necessary on the CP 525 in PC 2. In the CPU of PC 2, the handling block **RECEIVE ALL** is called cyclically. This handling block transfers the data received by the CP 525 to the CPU. The RECEIVE ALL is assigned the interface number of the CP 525 plugged into PC 2. (To set the interface number, see the instructions for the CP 525 in this manual).

1.2 Fetching Data (FETCH Job)

PC 1 is to fetch data on its own initiative from the CPU of PC 2 (FETCH job).

FETCH job



n = job number

The HDB **FETCH DIRECT n** (n = job number) is called in the CPU of PC 1. It must have the following parameters specified:

- interface number of the CP 525 in PC 1
- number of the corresponding FETCH job on CP 525 (PC 1)
- information about the data destination (where the data fetched from PC 2 is to be stored in the CPU of PC 1)

The FETCH job in the job block on the CP 525 of PC 1 contains information about the location of the data to be fetched from the CPU of PC 2. This information is sent to PC 2 in a **request telegram** (1).

No job block is required on the CP 525 of PC 2. The handling block **SEND ALL** is called cyclically in the CPU of PC 2. This HDB transfers the requested data from the CPU to the CP 525 of PC 2. It is assigned the interface number of the CP 525 in PC 2.

Once the requested data (2) is received in the CP 525 of PC 1 it is transferred to the CPU of PC 1. This is performed by the handling block **RECEIVE ALL**. RECEIVE ALL is only assigned the interface number of this CP 525 and is called up cyclically in the user program. (To set the interface number, see the instructions for the CP 525 in this manual).

2 Programming the Functions

To use the computer link, you must install **two** programs as follows:

- the **STEP 5 user program** for the CPU (with handling blocks)
- the **CP 525 user program** for the CP 525

You generate the CP 525 user program using the programming package COM 525. A detailed description of the functions of COM 525 can be found in the user's guide "Programming package COM 525" in this manual.

The STEP 5 user program for the CPU contains the handling block calls. More detailed information concerning the handling blocks in conjunction with the computer link can be found in the description "Using the handling blocks" in this manual.

The next pages provide an overview of the following:

- the parameters required during programming
- where these parameters are to be specified (STEP 5 or CPU 525 user program)
- the significance of the parameters in a wider context

In addition, Section 2.6 contains the job tables that list the permissible combinations of parameters. Use these tables to check whether the transfer you require can be implemented.

When generating the user programs you should proceed as follows:

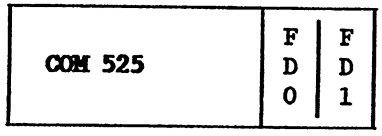
1. First ask the following questions:

- with which partners is data to be exchanged?
- which device interfaces are the partner devices to be connected to?
- which data is to be exchanged?
- which jobs are required and which handling blocks are needed?

2. Generate the CP 525 user program as follows:

- assign parameters to the jobs (Section 2.2)
- assign parameters to the interpreter and procedure (Section 2.3)
- load the job block, interpreter and procedure in the user memory of the CP 525 (RAM or EPROM submodule).

PG





assign parameters to the jobs
assign parameters to the interpreter and procedure



load these in the user memory of the CP 525:

either in an EPROM submodule
or in a RAM submodule (serial transmission)

3. Generate the STEP 5 user program for the CPU (Section 2.4) as follows:

PG

STL	F	F
LAD	D	D
CSF	0	1

↓

assign parameters to the handling blocks

↓

load them in the memory submodule of the CPU:

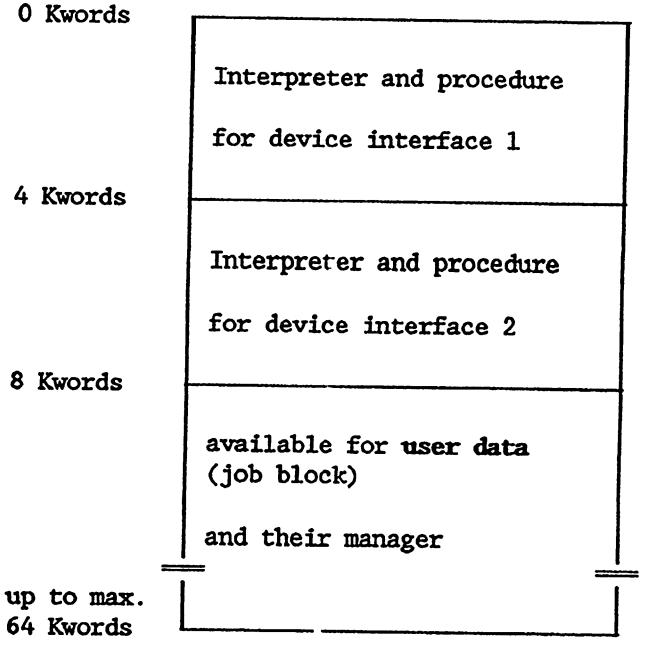
either in an EPROM submodule
or in a RAM submodule (serial transmission)

7

2.1 User Memory of the CP 525

You insert either a RAM or an EPROM submodule (see ordering data) in the memory submodule receptacle of the CP 525. The user memory of the CP 525 is divided into two areas:

- one area for the interpreter and procedures of the two device interfaces
- one area for the "actual" user data and their manager



2.1.1 Memory Requirements of the Job Block

If you operate the computer link on a device interface of the CP, the required job block occupies **1 Kword** in the memory module, regardless of how many jobs have been programmed. If the CP is used only for the computer link, a **16 Kword** memory submodule is adequate.

2.2 Assigning Parameters to the Job Block for the CP 525

Using the software package COM 525 you must assign parameters to a job block if your CP 525 is to transmit data to a partner, or fetch data from a partner on its **own initiative**.

A CP is **active** when it transmits on its own initiative. A CP is **passive** when it cannot operate on its own initiative; i.e., data can only be sent to it or fetched from it. If a CP is only required to be passive then it needs only an interpreter and procedure (no job block!) in the CP 525 user program.

It is assumed throughout this description that a CP is active and its partner passive.

The jobs in the job block contain information about the **data destination or data source in the partner**. A maximum of **189** jobs can be programmed in an active CP 525.

When assigning parameters to the job you must specify the following:

(1) **The job number** - must match the number specified in the HDB (SEND DIRECT or FETCH DIRECT). The job number can have values between 1 and 189.

(2) **Job type** - SEND or FETCH

(3) **Type of data to be transferred** - a SEND job can send data to the following types of destination:

data blocks
system data
absolute addresses
extended data blocks

a FETCH job can fetch data from the following types of source:

data blocks
flag bytes
input bytes
output bytes
I/O bytes
timer locations
counter locations
system data
absolute addresses
extended data blocks
extended I/Os

(4) **Destination address** for SEND or **source address** for FETCH - limit values depend on the data type.

(5) **CPU number** - if a particular CPU is to be addressed by this job, then the range of values is 1 to 4. No specification is made if the partner only has one CPU.

(6) **DB number** - assigned if "data block" or "extended data block" was specified as the type. With a SEND job this is the destination data block, with a FETCH job it is the source data block.

(7) **Coordination flag** (CF) - specified if the source or destination is a data block or an extended data block

Notes on **coordination flags**:

In conjunction with the computer link, the term "IPC flag" can be replaced by "coordination flag".

Using coordination flags you can enable or inhibit the data exchange between the CPU and CP.

In the SEND or FETCH telegram the byte and bit number of the coordination flag are transferred to the partner, which then evaluates them. This prevents as yet unprocessed data from being overwritten or read. The partner device then sends a reply message with an error number to the active CP 525.

You can write the STEP 5 user program in the CPU of the active CP so that it evaluates the received error number and repeats the SEND or FETCH message later.

For more information on coordination flags refer to the section in the instructions for the CP 525 ("Enabling/ disabling IPC flags"). The coordination flags must first be enabled by means of jumpers on the module. In some PCs the coordination flags must be defined as output IPC flags in DB 1.

Coordination flags with the link CP 525 ---> AS 512C

If the CP 525 sends a telegram that contains not only data but also the byte and bit number of a coordination flag, the AS 512C (as passive partner) sets the corresponding flag to "1" when this message is received. The data is accepted or read.

If the CP 525 sends another telegram with the same byte and bit number for a coordination flag to the AS 512C, the latter recognizes that the flag is already set. The data transfer/reception is disabled. The AS 512C sends a reply telegram with an error number to the active CP. In the active PC an error number is indicated in the condition codeword. If the the HDB is called again (SEND DIRECT or FETCH DIRECT) the telegram is sent again.

In the passive PC (with the AS 512C) the coordination flag must be reset by the STEP 5 program after the first telegram has been processed or when new data is ready to be fetched. The exchange of data is then once again enabled.

The range of values of the coordination flags is as follows:

byte number	0 to 223
bit number	0 to 7 or no specification

Coordination flags with the link CP 525 ---> CP 525

If the CP receives a telegram with the byte and bit number of the coordination flag, it checks whether the coordination flag with this number is set. If it is, the CP rejects the exchange of data with the active CP by means of a reply telegram with an error number. The data that has been sent is not tranferred to the CPU or is ignored by the CPU. If the coordination flag has already been reset when the telegram is received the data exchange takes place.

With the last data transfer between the CPU and CP 525, the byte number of the coordination flag is indicated in the condition codeword of the SEND ALL or RECEIVE ALL. (The number of data transfers depends on the number of items of data transferred and the field length for the data transfer - see parameter (field length BLGR for the 'SYNCHRON' HDB). This byte number can be evaluated by the STEP 5 program and the appropriate coordination flag is then set.

To allow the coordination flag to be evaluated by the passive CP, the byte number must be in the range between 1 and 223.

2.3 Assigning Parameters to the Interpreter and Procedure

Each CP interface being used requires an interpreter and procedure in the CP 525 user memory. For the computer link these are as follows:

- **interpreter RK 512**
- **procedure 3964** or **3964R**

The interpreter requires no parameters. You must specify the following for the procedure:

Data rate

19200 bps (*)
9600 bps
4800 bps
2400 bps
1200 bps
600 bps
300 bps
150 bps
110 bps

(*) 19200 bps are only allowed if you use the V.24 interface with the CP 525/CP 524 or the RS422-A/485 interface with the CP 524. In this case you can only load and operate **one** interface with the CP 525.

Priority

If both devices want to place a telegram on the transmission line simultaneously, one device must back off. The CP with the higher priority transmits its telegram first and the partner with lower priority, transmits its telegram second.. You can select the following:

higher priority
lower priority

Other parameters

The following parameters are **fixed** for the 3964 and 3964R procedures:

- character length in bits: **8**
- parity: **even**
- number of stop bits in bits: **1**

2.4 Parameters for the Handling Blocks

To start a SEND job, you must call the **SEND DIRECT** HDB in the STEP 5 user program. For a FETCH job you call the **FETCH DIRECT** HDB.

The following parameters must be specified:

(1) **Interface number** (SSNR)

Every CP (or IP) in the PC is allocated one or more interface numbers. The SSNR is set by means of jumpers on the module. The CP 525 has two interface numbers. The even number addresses device interface 1; the next higher odd number addresses device interface 2.

Range of values: 0 to 255

More details can be found in the instructions for the CP 525 in this manual.

(2) **Job number** (A-NR)

Every job (SEND DIRECT/FETCH DIRECT) is assigned a number that must match the number of the corresponding job in the user memory of the CP 525.

Range of values: 1 to 189

(3) **Source type** (QTYP) with SEND DIRECT or
Dest. type (ZTYP) with FETCH DIRECT

The following source types can be sent with SEND DIRECT:

data blocks	DB
flag bytes	FY
input bytes	IB
output bytes	QB
I/O bytes	PB
timer locations	TB
counter locations	CB
system data	RS
absolute addresses	AS
extended data blocks	DX
extended I/Os	OB

The following destination types are possible when receiving (FETCH DIRECT):

data blocks	DB
system data	RS
absolute addresses	AS
extended data blocks	DX

(4) **Data block number** (DBNR)

If you selected a data block or an extended data block as the source or destination type, you enter the number of the data block here.

Range of values: 3 to 255

(5) **Source start address** (QANF) with SEND DIRECT or
Dest. start address (ZANF) with FETCH DIRECT

The range of values permitted depends on the data type.

[1]) previously PB [2]) previously OB

(6) **Source length** (QLAE) with SEND DIRECT or
Dest. length (ZLAE) with FETCH DIRECT

(6) **Source length** (QLAE) with SEND DIRECT or
Dest. length (ZLAE) with FETCH DIRECT

You enter the number of items of data to be transferred. Whether this is specified in bytes or words depends on the data type (3); e.g., flags are specified in bytes, data blocks in words. The range of values permissible also depends on the data type and is listed in the job tables in Section 2.6.

(7) Other parameters are the condition codeword **ANZW** and the parameter assignment error byte **PAFE**. These indicate whether a data transfer was performed sucessfully or whether errors occurred. For more details see the description "Using the handling blocks" in this manual.

2.5 Example of a Complete Parameter Set

Task I: SEND job

Ten data words from data block 20 (DB20) starting at data word 10 (DW10) are to be transferred from PC 1 to PC 2 and be entered in data block 5 (DB5) starting at data word 1 (DW1). Make sure that PC 1 and PC 2 are each equipped with a CP 525.

In PC 1 you require the following:

- the **SEND DIRECT** HDB, which triggers the job. It must have the following parameters assigned:

SSNR	0	interface number is 0
A-NR	1	job number is 1
ANZW	FW12	condition codeword is flag word 12
QTYP	DB	source is a data block
DBNR	20	with number 20 and
QANF	10	with the start address 10
QLAE	10	10 data words will be sent
PAFE	FY11	FY11 is selected for parameter assignment errors

- the **SEND ALL** HDB that transfers the data from the PC to the CP must also be called. It requires the following parameters:

SSNR	0	interface number is 0
A-NR	0	ALL function is selected
ANZW	FW16	condition codeword is FW16
PAFE	FY19	FY19 is selected for parameter assignment errors

Note: the condition codeword of the ALL job must be different from the ANZW for SEND DIRECT, since other information is written to it.

Before beginning with the programming read the description "Using the handling blocks" in this manual to find the form in which these parameters must be specified.

For the CP in PC 1 you require the following:

- one job in the job block

job number	1	number as for HDB parameters
job	SEND	send has been selected
job type	DB	dest. is a data block
dest. word address	1	with start address 1
CPU number	-	only one CPU exists
DB no.	5	the dest. DB no. is 5
coordination flags	-	coordination flags are not being used

- the interpreter RK 512 and the procedure (e.g., 3964). For the procedure you must set the following:

data rate	9600 bps	data rate for this example
priority	low	optional

In PC 2 you require the following:

- the **RECEIVE ALL** HDB, which transfers the data and enters the following in the destination DB:

SSNR	0	interface number is 0
A-NR	0	job number is 0
ANZW	FW6	ALL function is selected
PAFE	FY4	condition codeword is FW 6
		FY4 is selected for parameter assignment errors

For the CP in PC 2 you require the following:

- **no** job block, as long as no active jobs are called up in PC 2 (SEND DIRECT, FETCH DIRECT).

- the interpreter RK 512 and the procedure 3964 for this CP. Set the following:

data rate	9600 bps	data rate for this example
priority	higher	

Task II: FETCH job

Six flag bytes starting at FY7 are to be fetched from PC 2. These bytes are to be written into data block 20 starting at data word 30 in PC 1. A FETCH job is therefore programmed in PC 1.

In PC 1 you require the following:

- the call for the **FETCH DIRECT** HDB to trigger the job:

SSNR	0	as for task I
A-NR	2	job number here is 2
ANZW	FW34	condition codes in FW34
ZTYP	DB	destination is a DB
DBNR	20	with no. 20 and
ZANF	30	start address 30
ZLAE	3	fetch 6 bytes = 3 words
PAFE	FY33	parameter assignment errors

- the **RECEIVE ALL** HDB to receive the data fetched:

SSNR	0	interface number is 0
A-NR	0	job number is 0
ANZW	FW27	ALL function is selected
PAFE	FY26	condition codeword in FW27
		parameter assignment errors in FY26

For the CP in PC 1 you require the following:

- the job with the number 2 as follows:

job number	2	number as for HDB
job	FETCH	
job type	flags	source is the flag area
source byte address	7	from byte 7
CPU no.	-	only one CPU present
DB no.	-	no parameter possible
coordination flags	-	no parameter possible

- the interpreter and procedure are already programmed and available if you carried out task I:

data rate	9600 bps
priority	low

In PC 2 you require the following:

- a **SEND ALL** HDB as follows, which transfers the flag bytes to its CP:

		interface number is 0
SSNR	0	job number is 0
A-NR	0	ALL function is selected
ANZW	FW40	FW40 is selected as ANZW
PAFE	FY39	FY39 selected for parameter assignment errors

For the CP in PC 2 you require the following:

- **no** job block, however, interpreter and procedure (if not already available from task I). For the procedure you need the following:

data rate	9600 bps
priority	higher

2.6 Job Tables

The following tables contain **all the transferable data types** and their parameter assignments in the HDB and in the job block. Specifications regarding the addresses depend on the PC and are not always the same for different types. This is particularly important with absolute addresses. Also refer to the documentation for your specific PC.

Because of the wide variety of parameters, each table is spread over two pages. Remember that the two tables are, in fact, one.

1. Transmit data: SEND

As the source, all data types stored on the partner in a destination DB or DX (exception AS/RS) can be used.

The parameter QLAE (source length) represents a number of bytes, if the source area is organised in bytes; if not, it represents a number of words. If 0 is specified there will be no exchange of telegrams.

Be careful with odd numbers of bytes; since the destination area is a DB or DX, only full words can be stored there. If the partner is a CP 525, it writes 0s to the right-hand data byte (DR) if it receives an odd number of bytes.

Notes on the table
On the right-hand side there are three columns with specifications for the telegram header. These specifications need to be taken into account only if you are using a device of a different make as the partner, and you are implementing the communications software yourself; or if you wish to monitor the data exchange on the line with an interface test device (FOXPG). The specifications refer to the contents of bytes 3 to 8 in the header of the SEND telegram.

Explanation of the abbreviations:

parameter assignment of the HDB

QTYP	= source type
DBNR	= data block number
QANF	= source start address
QLAE	= source length

parameter assignment in COM 525

Z-Typ	= destination type
Z-DB	= data block number of the destination
Z-Adr	= destination start address
CF-poss.	= coordination flags possible?
header, bytes	= telegram header of the data transmission on the line

Source, Send from PC 1	Dest., to PC 2	Param. assignment of HDB in PC 1			
		QTYP	DBNR	QANF	QLAE
Data block	Data block	DB	3-255	0-2047	1-2048
Ext. DB	Data block	DX	3-255	0-2047	1-2048
Flags	Data block	FY	irrel.	0-255	1-256
Inputs	Data block	IB	irrel.	0-127	1-128
Outputs	Data block	QB	irrel.	0-127	1-128
Counters 115U	Data block	CB	irrel.	0-127	1-128
Counters 135U	Data block	CB	irrel.	0-127	1-128
Counters 150U	Data block	CB	irrel.	0-255	1-256
Timers 115U	Data block	TB	irrel.	0-127	1-128
Timers 135U	Data block	TB	irrel.	0-127	1-128
Timers 150U	Data block	TB	irrel.	0-255	1-256
I/Os	Data block	PY	irrel.	0-255	1-256
Sy-Adr.135U	Sys. addr.	RS	irrel.	0-255	1-256
Sy-Adr.150U	Sys. addr.	RS	irrel.	0-511	1-512
Abs. addr.	Abs. addr.	AS	irrel.	0-+32767 -32768	1-32767
ext. I/Os only for PC 150U	Data block	OB	irrel.	0-255	1-256

Param. assign. in COM 525 in PC 1				Telegram header, bytes		
Z-Typ	Z-DB	Z-Adr	CF. poss.	3/4 comm.	5/6 Z-DB/Z-Adr	7/8 Num. in
DB	3-255	0-255	yes	AD	DB/DW	Words
DB	3-255	0-255	yes	AX	DB/DW	Words
DB	3-255	0-255	yes	AM	DB/DW	Bytes
DB	3-255	0-255	yes	AE	DB/DW	Bytes
DB	3-255	0-255	yes	AA	DB/DW	Bytes
DB	3-255	0-255	yes	AZ	DB/DW	Words
DB	3-255	0-255	yes	AZ	DB/DW	Words
DB	3-255	0-255	yes	AZ	DB/DW	Words
DB	3-255	0-255	yes	AT	DB/DW	Words
DB	3-255	0-255	yes	AT	DB/DW	Words
DB	3-255	0-255	yes	AT	DB/DW	Words
DB	3-255	0-255	yes	AP	DB/DW	Bytes
RS	———	0-255	no	AB	Address	Words
RS	———	0-511	no	AB	Address	Words
AS	———	0-65535	no	AS	Address	Words
DB	3-255	0-255	yes	AQ	DB/DW	Byte

7

Source, **Send from** PC 1	Dest., to PC 2	Param. assignment of HDB in PC 1			
		QTYP	DBNR	QANF	QLAE
Data block	Ext. DB	DB	3-255	0-2047	1-2048
Ext. DB	Ext. DB	DX	3-255	0-2047	1-2048
Flags	Ext. DB	FY	irrel.	0-255	1-256
Inputs	Ext. DB	IB	irrel.	0-127	1-128
Outputs	Ext. DB	QB	irrel.	0-127	1-128
Counters 115U	Ext. DB	CB	irrel.	0-127	1-128
Counters 135U	Ext. DB	CB	irrel.	0-127	1-128
Counters 150U	Ext. DB	CB	irrel.	0-255	1-256
Timers 115U	Ext. DB	TB	irrel.	0-127	1-128
Timers 135U	Ext. DB	TB	irrel.	0-127	1-128
Timers 150U	Ext. DB	TB	irrel.	0-255	1-256
I/Os	Ext. DB	PB	irrel.	0-255	1-256
ext. I/Os only for PC 150U	Ext. DB	OB	irrel.	0-255	1-256

Param. assign. in COM 525 in PC 1				Telegram header, bytes		
Z-Typ	Z-DB	Z-Adr	CF. poss.	3/4 comm.	5/6 Z-DB/Z-Adr	7/8 Num. in
DX	3-255	0-255	yes	OD	DX/DW	Words
DX	3-255	0-255	yes	OX	DX/DW	Words
DX	3-255	0-255	yes	OM	DX/DW	Bytes
DX	3-255	0-255	yes	OE	DX/DW	Bytes
DX	3-255	0-255	yes	OA	DX/DW	Bytes
DX	3-255	0-255	yes	OZ	DX/DW	Words
DX	3-255	0-255	yes	OZ	DX/DW	Words
DX	3-255	0-255	yes	OZ	DX/DW	Words
DX	3-255	0-255	yes	OT	DX/DW	Words
DX	3-255	0-255	yes	OT	DX/DW	Words
DX	3-255	0-255	yes	OT	DX/DW	Words
DX	3-255	0-255	yes	OP	DX/DW	Bytes
DX	3-255	0-255	yes	OQ	DX/DW	Byte

7

2. Fetch data: FETCH

All data types can be specified as the source in the partner PC. The destination in the HDB can only be a **data block** or an extended data block (exception RS/AS). The destination length (ZLAE) can be specified only in words. If, for example, 5 input bytes are to be fetched, "3" = three words = six bytes must be specified in the HDB. If 0 is specified there is no telegram exchange.

Notes on the table
On the right-hand side there are three columns with specifications for the telegram header. These specifications need to be taken into account only if you are using a device of a different make as the partner, and you are implementing the communications software yourself; or if you wish to monitor the data exchange on the line with an interface test device (FOXPG). The specifications refer to the contents of bytes 3 to 8 in the header of the FETCH telegram.

Explanation of the abbreviations:

parameter assignment of the HDB

ZTYP	= destination type
DBNR	= data block number
ZANF	= destination start address
ZLAE	= destination length

parameter assignment in COM 525

Q-Typ	= source type
Q-DB	= data block number of the source
Q-Adr	= source start address
CF-poss.	= coordination flags possible?
header, bytes	= telegram header of the data transmission on the line

Source: fetch from PC 2	Dest.: in PC 1	Param. assignment in HDB in PC 1			
		ZTYP	DBNR	ZANF	ZLAE
Data block	Data block	DB	3-255	0-2047	1-2048
Ext. DB	Data block	DB	3-255	0-2047	1-2048
Flags	Data block	DB	3-255	0-2047	1-128
Inputs	Data block	DB	3-255	0-2047	1-64
Outputs	Data block	DB	3-255	0-2047	1-64
Counters 115U	Data block	DB	3-255	0-2047	1-128
Counters 135U	Data block	DB	3-255	0-2047	1-128
Counters 150U	Data block	DB	3-255	0-2047	1-256
Timers 115U	Data block	DB	3-255	0-2047	1-128
Timers 135U	Data block	DB	3-255	0-2047	1-128
Timers 150U	Data block	DB	3-255	0-2047	1-256
I/Os	Data block	DB	3-255	0-2047	1-128
Sys.Ad.135U	Sys. addr.	RS	irrel	0-255	1-256
Sys.Ad.150U	Sys. addr.	RS	irrel.	0-511	1-512
Abs. addr. 115 U	Abs. addr.	AS	irrel.	0-+32767 -32768	1-32767
Abs. addr. 135U/150U	Abs. addr.	AS	irrel.	0-+32767 -32768	1-32767
ext. I/Os only for S5-150U	Data block	DB	3-255	0-2047	1-128

Param. assign. in COM 525 in PC 1				Telegram header, bytes		
Q-Typ	Q-DB	Q-Adr	CF poss.	3/4 Comm.	5/6 Q-DB/Q-Adr	7/8 Num. in
Data bl.	0-255	0-255	yes	ED	DB/DW	Words
Ext. DB	0-255	0-255	yes	EX	DX/DW	Words
Flag	———	0-254	no	EM	byte addr.	Bytes
Input	———	0-126	no	EE	byte addr.	Bytes
Output	———	0-126	no	EA	byte addr.	Bytes
Counter	———	0-127	no	EZ	Counter no	Words
Counter	———	0-127	no	EZ	Counter no	Words
Counter	———	0-255	no	EZ	Counter no	Words
Timer	———	0-127	no	ET	Timer no.	Words
Timer	———	0-127	no	ET	Timer no.	Words
Timer	———	0-255	no	ET	Timer no.	Words
I/O	———	0-254	no	EP	I/O addr.	Bytes
Sys. dat.	———	0-255	no	EB	Sys.addr.	Words
Sys. dat.	———	0-511	no	EB	Sys.addr.	Words
Abs. addr	———	0-65534	no	ES	Abs.addr.	Words
Abs. addr	———	0-65535	no	ES	Abs.addr.	Words
ext. I/O	———	0-254	no	EQ	I/O addr.	Byte

7

Source: fetch from PC 2	Dest.: in PC 1	Param. assignment in HDB in PC 1			
		ZTYP	DBNR	ZANF	ZIAE
Data block	Ext. DB	DX	3-255	0-2047	1-2048
Ext. DB	Ext. DB	DX	3-255	0-2047	1-2048
Flags	Ext. DB	DX	3-255	0-2047	1-128
Inputs	Ext. DB	DX	3-255	0-2047	1-64
Outputs	Ext. DB	DX	3-255	0-2047	1-64
Counters 115U	Ext. DB	DX	3-255	0-2047	1-128
Counters 135U	Ext. DB	DX	3-255	0-2047	1-128
Counters 150U	Ext. DB	DX	3-255	0-2047	1-256
Timers 115U	Ext. DB	DX	3-255	0-2047	1-128
Timers 135U	Ext. DB	DX	3-255	0-2047	1-128
Timers 150U	Ext. DB	DX	3-255	0-2047	1-256
I/Os	Ext. DB	DX	3-255	0-2047	1-128
ext. I/Os only for S5-150U	Ext. DB	DX	3-255	0-2047	1-128

Param. assign. in COM 525 in PC 1				Telegram header, bytes		
Q-Typ	Q-DB	Q-Adr	CF poss.	3/4 Comm.	5/6 Q-DB/Q-Adr	7/8 Num. in
Data bl.	0-255	0-255	yes	ED	DB/DW	Words
Ext. DB	0-255	0-255	yes	EX	DX/DW	Words
Flag	———	0-254	no	EM	byte addr.	Bytes
Input	———	0-126	no	EE	byte addr.	Bytes
Output	———	0-126	no	EA	byte addr.	Bytes
Counter	———	0-127	no	EZ	Counter no	Words
Counter	———	0-127	no	EZ	Counter no	Words
Counter	———	0-255	no	EZ	Counter no	Words
Timer	———	0-127	no	ET	Timer no.	Words
Timer	———	0-127	no	ET	Timer no.	Words
Timer	———	0-255	no	ET	Timer no.	Words
I/O	———	0-254	no	EP	I/O addr.	Bytes
ext. I/O	———	0-254	no	EQ	I/O addr.	Byte

7

3 Settings and Start-up

Chapter 3 contains information about the following:

- settings on the CP 525
- settings on the partner
- start-up routines

3.1 Settings on the CP 525

3.1.1 Jumper Settings

As described in the instructions for the CP 525 in this manual, you must set jumpers on the CP 525 (to assign interface numbers and enable the interprocessor communication (IPC) flags). **No** jumper settings are required for interpreters and procedures.

3.1.2 Setting Interpreters and Procedures

No settings are required on the interpreter RK 512; it is simply loaded in the CP 525.

You set the **data rate, transmission format** and **priority** for the procedure when programming the CP 525 user program with COM 525. The permissible values can be found in Section 2.3.

3.2 Settings on the Partner

If the partner is also a CP 525 the settings are as described in Section 3.1.

If your data exchange is performed with an AS 512C interface module, you must set the required priority and data rate (see instructions AS 512C).

If the partner is a PROMEA or a DUST 3964 the settings can also be found in the corresponding instructions.

3.3 Start-up Routine

3.3.1 Routine Following Power-up

After you switch on the power supply the two **LEDs** on the front panel of the CP 525 light up. The CP 525 now checks the following:

- the RAM and EPROM are functional
- the interpreter and procedure are present and complete in the memory submodule

Following the start-up phase which takes approximately two seconds, the LEDs of the loaded interfaces go off. If the selector is switched to STOP/PGR or a hardware error has been recognized the LEDs remain lit.

Following the start-up phase, the CP 525 expects the **SYNCHRON** job from the CPU.

3.3.2 SYNCHRON Job from the CPU

The SYNCHRON job is initiated after the start-up phase

The SYNCHRON HDB is required for handling the SEND, FETCH, RECEIVE jobs so that the length of the fields for the data transfer between the CPU and CP 525 is established. The SYNCHRON HDB is usually called in the STEP 5 user program in organization blocks OB20, OB21 and OB22.

When the SYNCHRON job runs without errors, the CP is ready to process jobs (SEND, FETCH, RECEIVE) from the CPU and from the partner.

Possible reactions before SYNCHRON runs are as follows:
- the partner sends a telegram before the SYNCHRON job has run; the CP sends a reply telegram containing the error number 2AH. 4FH is entered in the error message area of the SYSTAT.

Jobs sent from the CPU (except SYNCHRON) before SYNCHRON has run are handled as follows:
- they are rejected with "not ready" (error message number 81H in the PAFE byte of the corresponding job).

A new SYNCHRON job is received by a functional interface:
(e.g., the reset switch on the CPU has been pressed)

If a **partner job** is being processed it is be aborted with a reply telegram containing the error number 2AH. The corresponding error number in the error message area of the SYSTAT is 4FH. Partner jobs are then ignored until the CP 525 has gone through its cold restart routine.

If a **PC job** is being processed, it is terminated with the error number 7H in the condition codeword. The error number 16H is entered in the error area of the SYSTAT.

If SYNCHRON runs **without** errors, the job statuses (and therefore the condition codewords) of all jobs are cleared. The error number 7H, is therefore no longer visible. The entries in SYSTAT are retained.

If SYNCHRON runs **with** errors, all the PC jobs set to "running" during the cold restart phase are also terminated with the error number 7H. (16H is again entered in SYSTAT).

New PC jobs (except for SYNCHRON) are then blocked by the "not ready" message (number 81H in the PAFE byte of the corresponding job).

Note:

Aborting a PC job by means of SYNCHRON can also cause a further error status.

Reason: if a job is stopped immediately, this fact cannot be signalled to the partner. The partner still expects follow-on telegrams connected with the current job. It therefore interprets the first telegram following SYNCHRON as a follow-on telegram. The partner, however, recognizes that it is not a follow-on telegram associated with the current job and answers with a reply telegram with the number 36H. This causes the first job following the SYNCHRON to be aborted with the error number CH in the condition codeword. The number 37H is entered in SYSTAT.

3.3.3 Mode Selector Settings RUN/STOP/PGR

The settings of the mode selector on the front panel of the CP 525 are as follows:

o **RUN**

The RK 512 interpreter processes jobs arriving from the CPU and from the partner (if a SYNCHRON job was performed without errors).

The LEDs of the loaded interfaces on the front panel of the CP 525 are no longer lit.

o **STOP** or **PGR**

These settings are identical. They are used to stop the transmission of data on the serial transmission line. Any jobs arriving are rejected with an error message.

- PC jobs (SEND/FETCH) are terminated with the error number CH in the condition codeword. The error number 27H is written in SYSTAT.
- Partner jobs are rejected with a reply telegram containing the error number 2AH. The number 4EH is entered in SYSTAT.

Switching the mode selector from RUN to STOP/PGR

If a job is currently being executed it will be completed, i.e., without an error message!

Depending on the number of items of data to be transmitted, it may take some time before the STOP/PGR state is reached. This state is shown by the LEDs on the front panel.

Switching the mode selector from STOP/PGR to RUN

The interpreter once again begins processing jobs. The LED on the front panel goes off (only if the interpreter and procedure are loaded). The CP 525 does **not** go through a cold restart when the mode selector position is changed.

Exception

If part or all of the interface has just been loaded or reloaded (transfer from PG) the interface requires a cold restart. In this case, switching the selector position from STOP/PGR to RUN brings about a cold restart, or you can press the key "cold restart" on the PG. The selector on the PC must be set to RUN.

3.3.4 Transferring from the PG

If the transfer of the interpreter or procedure or job block is initiated at the PG, the software of the CP 525 goes to the wait state. In the wait state the LED of the corresponding interface is lit on the front panel.

The transfer is started only when the program to be transferred is different from the program already loaded in the CP 525. For this reason the PG first reads the program in the CP and compares the two programs byte by byte. The position of the mode selector is irrelevant for this transfer.

Before the software of the CP 525 goes to the wait status, the job currently being processed by the partner and CPU is completed without an error message. Depending on the number of items of data to be transmitted, it may therefore take some time before the wait status is actually adopted.

During the wait status no jobs sent by the partner (partner jobs) can be processed. The procedure does not react to characters arriving via the serial interface. During the wait status the LEDs on the front panel are lit.

During the wait status the CP 525 also rejects any jobs from its own CPU (PC jobs), instead "overload" is indicated. This is recognized by the error number 91H in the PAFE byte of the job affected. In rare cases, a PC job might be completed (terminated with error) before the PAFE error message (number 6H in the condition codeword and 11H in SYSTAT). In this case the PC was unable to recognize the overload early enough.

Following the transfer, the software remains in the wait status until you initiate a cold restart on the interface, either with the "cold restart" key on the PG, or by switching the mode selector on the CP from STOP/PGR to RUN.

Following the cold restart the LEDs on the front panel go off if the transferred program is correct and complete. Following the transfer, a SYNCHRON job is no longer required.

4 How is Data Transferred?

The data transfer between the partners takes place on the following three levels:

1. **Telegram level**

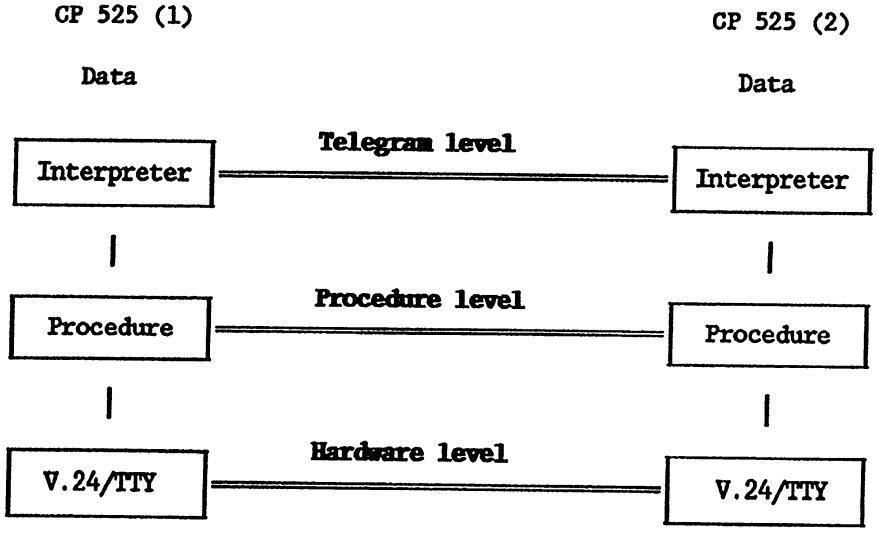
The interpreters of the two CP 525s exchange data in the form of fixed **telegrams**. The interpreter of the CP 525 (1) sends the interpreter of the CP 525 (2) a command telegram. CP 525 (2) answers with a reply telegram.

2. **Procedure level**

The telegrams are changed into a particular **format** and integrated into the handshaking.

3. **Hardware level**

The hardware level covers the **physical conditions** of the data transfer. The CP 525/CP 524 can operate either with TTY (current loop 20 mA) or with a voltage level V.24. The front connectors are designed for both types of signal. In addition, the RS422-A/485 module can be used in the CP 524.



4.1 Telegram Level

The following sections provide detailed information about the telegram level. The various telegrams are defined as follows:

CP 525 (1) sends a **command telegram** to CP 525 (2).
CP 525 (2) sends a **reply telegram** to CP 525 (1).

Command telegrams are either **SEND telegrams** (send data)
or **FETCH telegrams** (fetch data).

With SEND telegrams the passive partner sends a
reply telegram without data,
With FETCH telegrams the passive partner sends a
reply telegram with data.

If the volume of data exceeds 128 bytes then a **follow-on telegram** is sent.

Structure and contents of a command telegram (SEND telegram/FETCH telegram

A SEND telegram consists of a telegram header and data.
A FETCH telegram consists only of the telegram header.

The telegram header is made up of **10 bytes** and contains (in a SEND telegram) details of the data destination; (in a FETCH telegram) details of the data source.

Byte

1	2	3	4	5	6	7	8	9	10
00H (FFH)	00H	command		high dest./source	low	high number	low	CPU no./CF	

Significance of the bytes:

1 : telegram identifier (00H or FFH with foll. comm. tels)
2 : telegram identifier (00H)
3 : command, SEND ('A'or `0`) or FETCH ('E')
4 : command type, i.e. type of data to be transferred
'D' = data block 'X' = extended DB
'E' = input bytes 'A' = output bytes
'M' = flag bytes 'P' = I/O bytes
'Z' = counter locations 'T' = timer locations
'S' = absolute addresses 'B' = system addresses
'Q' = extended I/Os

5 and 6: destination address for SEND or source address for FETCH, e.g.: byte 5 = DB no. and byte 6 = DW no.

7 and 8: number of pieces of information data to be transferred, depending on the type in bytes or in words 9: byte number of the coordination flag. If no CF is specified, this contains FFH

10: bits 0 to 3: bit number of the coordination flag (CF)
If no CF is specified, FH is entered
bits 4 to 7: CPU number, a number from 1 to 4.
If no CPU number is specified, but rather a coordination flag, this contains 0H. If no CPU no. and no CF is specified, byte 10 contains FFH. In both cases all CPUs are possible at the partner.

The letters specified in bytes 3 and 4 are ASCII characters.

The telegram header of the follow-on command telegram includes only bytes 1 to 4.

After the command telegram is transferred, the interpreter expects a reply telegram from the partner within the monitoring time. The length of the monitoring time depends on the data rate, i.e. the baud rate.

Table of monitoring times in the CP 525:

19200 - 1200 bps = approx. 5 seconds
600 bps = approx. 7 "
300 bps = approx. 10
150 bps = approx. 15
110 bps = approx. 20

Tolerance: +/- 500 ms

Structure and contents of the reply telegram

The reply telegram consists of four bytes and contains information about the processing of the job.

Byte

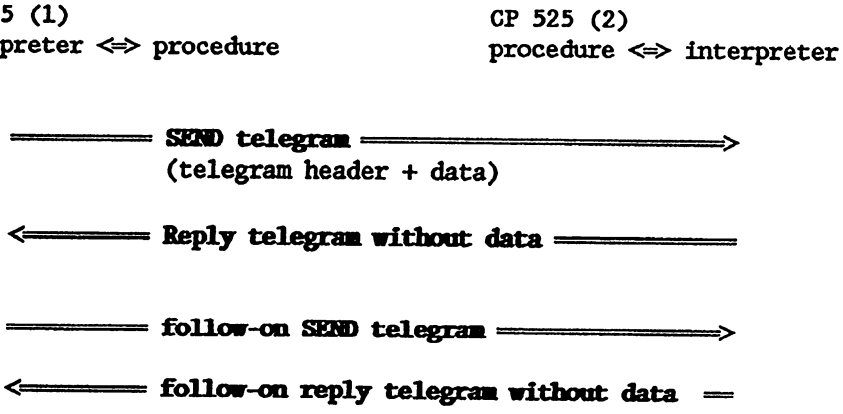
1	2	3	4
00H / FFH	00H	00H	error number

Significance of the bytes is as follows:

1: telegram identifier (00H or FFH for follow-on reply telegrams)
2: telegram identifier (00H)
3: contains 00H
4: error number of the partner (see Chapter 8, Error handling).

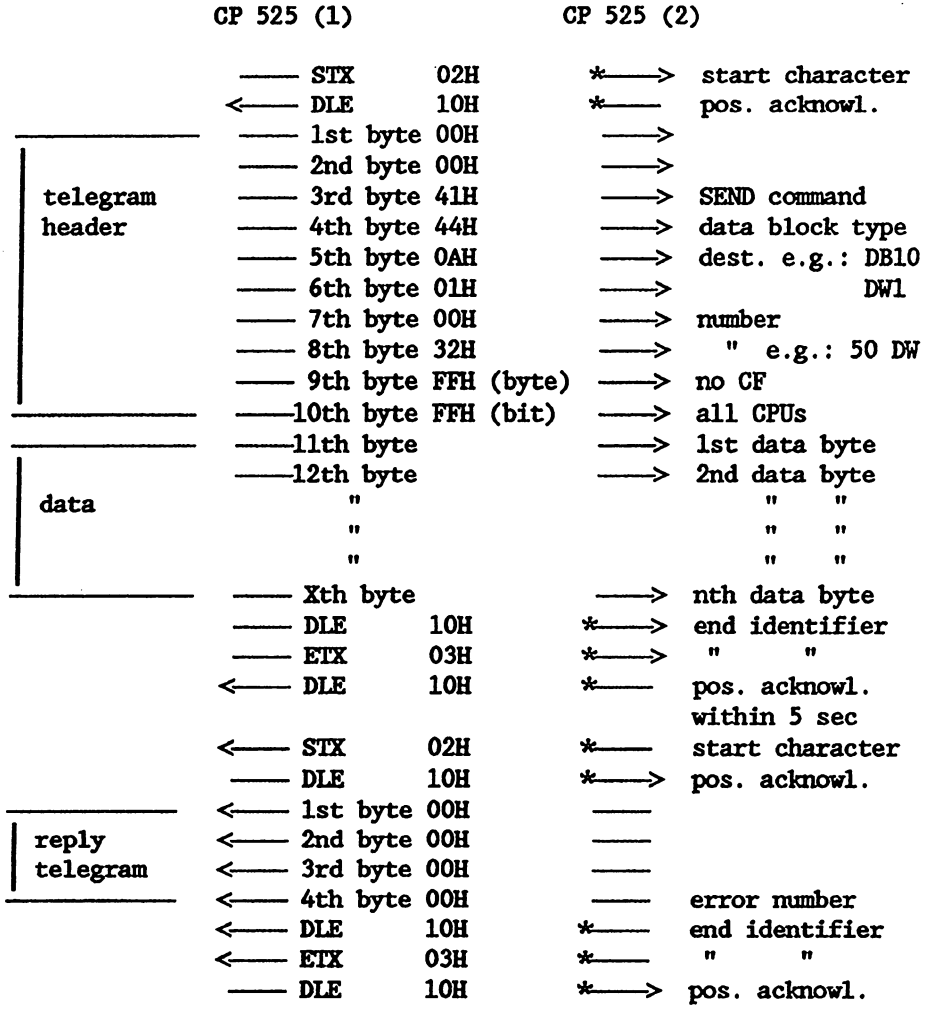
4.1.1 Sending Data

SEND job: CP 525 (1) sends data to CP 525 (2).



Follow-on SEND telegrams and follow-on reply telegrams are sent when the volume of information data exceeds **128 bytes**.

Sequence of events for a SEND telegram:

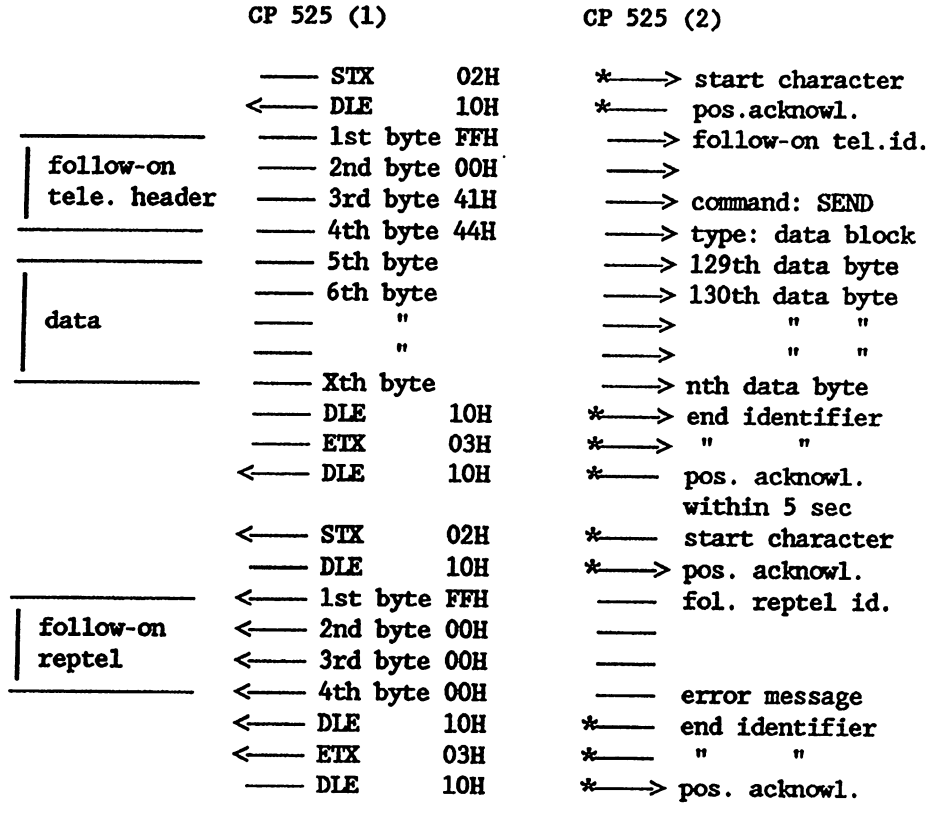


The characters marked * are added to the data field with the telegram header during the transfer by the 3964 procedure.

The telegram exchange for a SEND telegram without errors and with a length less than 129 bytes is completed (the fourth byte in the reply telegram does not contain an error number).

Follow-on SEND telegrams

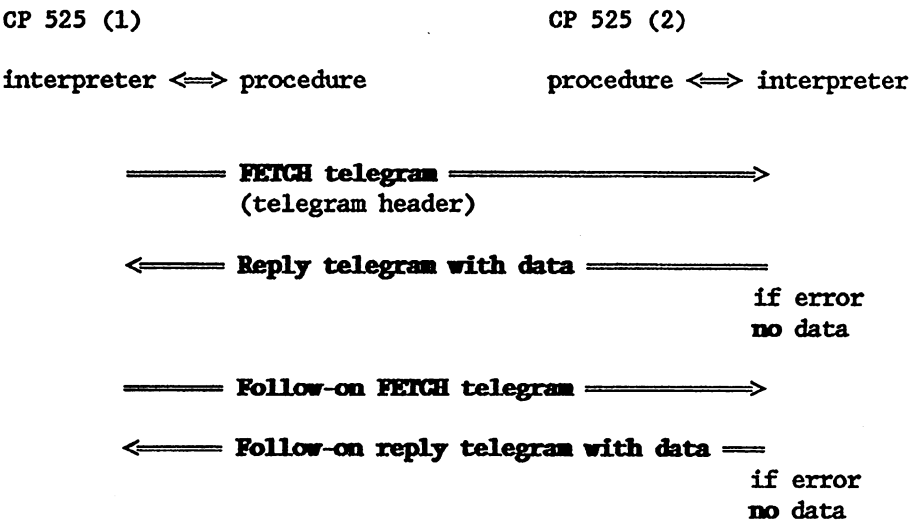
A follow-on SEND telegram is started if the volume of data to be sent exceeds **128 bytes**. The sequence of events corresponds to the SEND telegram.



Another follow-on SEND telegram is started if there are more than 256 bytes to be transferred.

4.1.2 Fetching Data

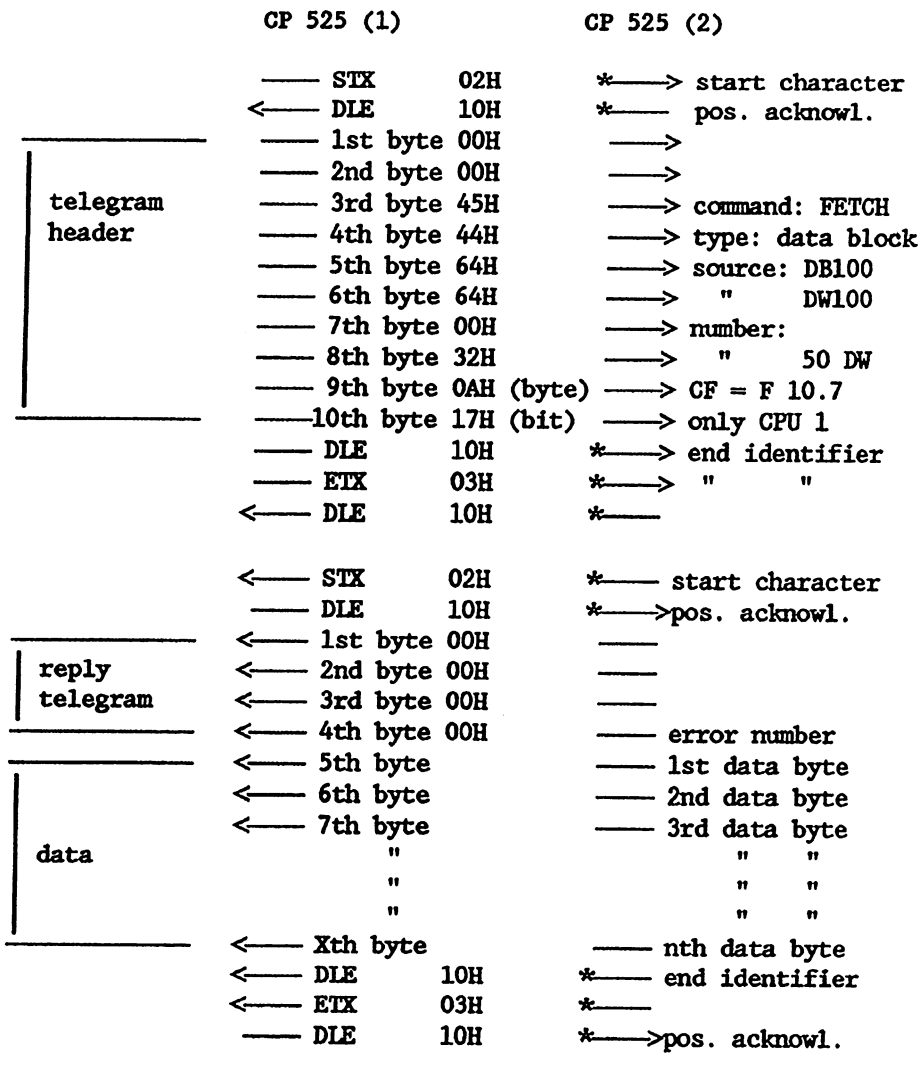
FETCH job: CP 525 (1) requests data from CP 525 (2).



Follow-on FETCH telegrams and follow-on reply telegrams with data are sent only if the volume of information data exceeds 128 bytes.

If there is an error number $\neq 0$ in the 4th byte of the reply telegram, then no data will be added.

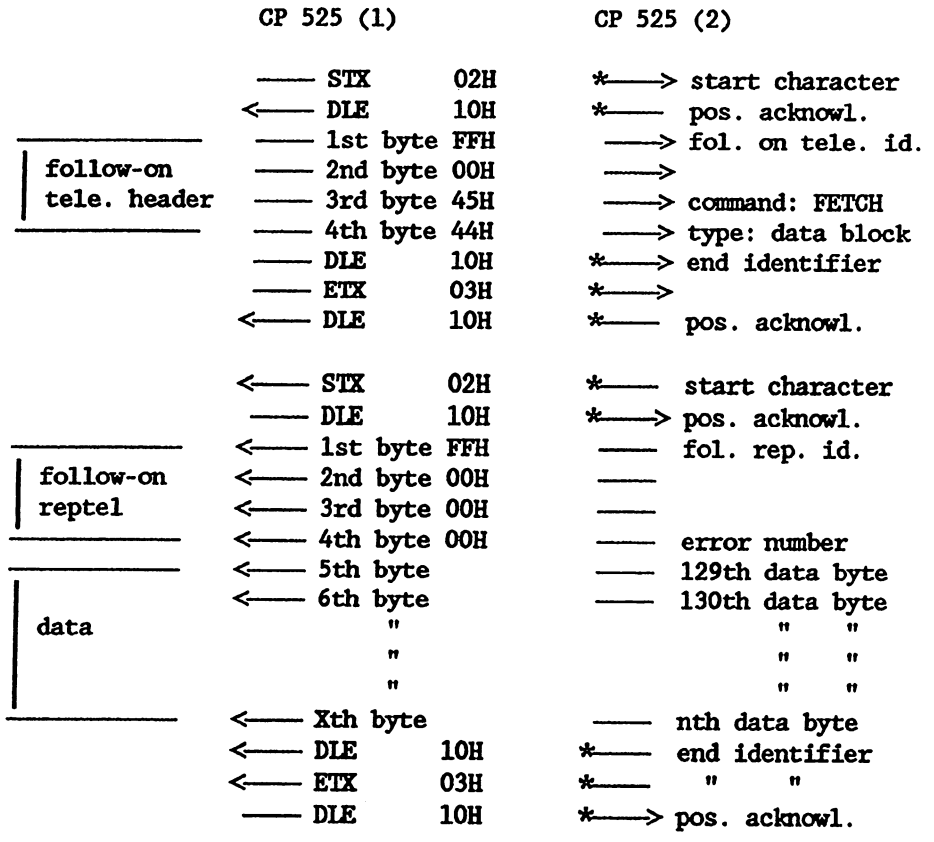
Sequence of events for a FETCH telegram: (request telegram)



This is the sequence of events for a FETCH telegram without errors and with a length of less than 129 bytes.

If more than 128 bytes are requested, these are automatically fetched in one or more follow-on telegrams.

Follow-on FETCH telegram with follow-on reply telegram:



Another follow-on FETCH telegram is started if the volume of information data exceeds 256 bytes.

4.2 Quasi Full Duplex Operation

Full duplex means that telegrams can normally be sent at any time, except in the following situation:

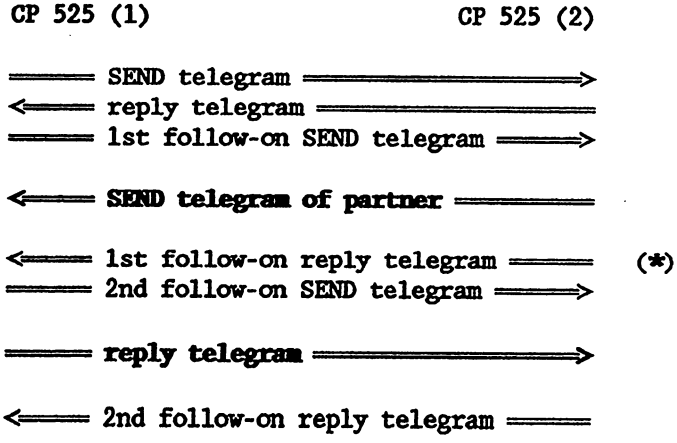
- CP 525 (1) must **not** send information data, when CP 525 (2) is sending information data.

- CP 525 (1) must wait until the telegram is completed with <DLE><ETX> from CP 525 (2) and <DLE> from CP 525 (1).

Under certain circumstances, (if both CPs wish to send), a SEND telegram can be transferred by the partner before the reply telegram; for example, if a SEND telegram from the partner has been entered in the output buffer of CP 525 (1) before the reply telegram.

In the following example, the follow-on reply telegram (*) for the first SEND telegram is sent only after the **SEND telegram of the partner**.

Example of a possible telegram exchange:



5 Procedures

Section 5 contains information about the procedures 3964 and 3964R.

5.1 Procedure 3964

5.1.1 Procedure Data

The procedure 3964 controls the flow of data between your programmable controller and the partner, e.g., a central process com-puter, another CP 525 or an AS 512C. In the CP 525, the interpreter transfers the data to be sent into **output buffers** for the procedure. The procedure sends this data along with the transmission protocol 3964 to the partner; if necessary repeats the transmission, and signals non-recoverable errors to the interpreter.

Data from the partner are stored in **input buffers**. If the data is received without errors, it is transferred to the interpreter for further processing.

The 3964 procedure is an asynchronous, bit serial transfer procedure. The transmission rate both sending and receiving must be the same on the CP 525 and on the partner (asynchronous because there are no timing circuits between the two devices).

Control and information characters are transmitted on the link. To ensure that every character is recognized by the receiver and to allow the transmission to be checked for errors, the characters are encapsulated by other bits.

The order of the bits is as follows:

SA	I0	I1	I2	I3	I4	I5	I6	I7	PA	SO

SA = start bit
I* = information bit no.
PA = parity bit
SO = stop bit

The control characters for the 3964 procedure are taken from the DIN standard 66003 for the 7-bit code. However, the character length used is **8-bit** (bit 7 = 0). There is no code prescribed for the information characters (code transparency).

You can set the following procedure parameters when writing the CP 525 user program with COM 525:

o **Data rate**

The speed of the data transmission is specified in bps. You can select the following values:

19200 bps *
9600 bps
4800 bps
2400 bps
1200 bps
600 bps
300 bps
150 bps
110 bps

(*) 19200 bps are only allowed if you use the V.24 interface with the CP 525/CP 524 or the RS422-A/485 interface with the CP 524. In this case you can only load and operate **one** interface with the CP 525 since the maximum sum of the data rates is also 19200 bps.

o **Priority**

To resolve initialization conflicts, you can set the priority of the CP 525. At the partner the opposite priority must be set. You can choose between the following:

higher priority
lower priority

The following are **fixed** for the 3964 procedure:

character length: 8 bits
number of stop bits: 1
parity bit, i.e., checking for even parity

5.1.2 Transmitting with the 3964 Procedure

To establish the link, the 3964 procedure sends the control character STX. If the partner replies within the time allowed of 550 ms (i.e., no QVZ = timeout) the procedure starts the transmission. If the partner answers with NAK, any other character (except DLE) or if there is a timeout, then the attempt to establish a link has failed. After a total of six unsuccessful attempts, the procedure gives up, signals the interpreter that there is an error establishing the link, and sends the character NAK to the partner.

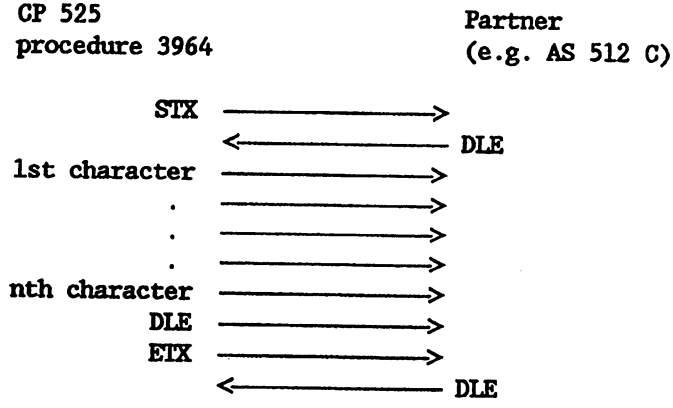
If the link is successfully established, the information data in the current output buffer is sent to the partner at the selected data rate. The partner monitors the time between the characters as it receives them. The time between two characters must not exceed the character delay time of 220 ms (ZVZ = character timeout).

Each DLE character found in the buffer is sent as two DLE characters (DLE doubling), i.e., the data (10H) is sent twice.

After sending the contents of the buffer, the procedure adds the characters DLE and ETX as end identifier and waits for an acknowledgement. If the partner sends the character DLE within the time allowed (QVZ), the data field was received without errors. If the partner replies with NAK, any other character (except DLE), a corrupted character, or if there is a timeout, the procedure begins again with the link establishment STX. After a total of six unsuccessful attempts to send the data field, the procedure breaks off the attempt, signals the error to the interpreter and sends NAK to the partner.

If, while data is being transmitted, the partner sends the character NAK, the procedure breaks off the field and repeats it as previously described. If any other character is received, the procedure waits for the character delay time (ZVZ) to elapse and then sends NAK to reset the partner. Then the procedure begins the transmission again with the link establishment STX.

Example of a data exchange without errors:



5.1.3 Receiving with the 3964 Procedure

In the idle state, i.e., when there is no send job to be processed, the 3964 procedure waits for the link to be established by the partner.

If the procedure receives any character (except STX) while in the idle state, it waits until the character delay time (ZVZ) has elapsed and then sends a NAK character. The error is signalled to the interpreter.

If the procedure receives the character STX and if it has an empty input buffer available, it answers with DLE. The characters then received are entered in the input buffer. If two DLE characters are received in succession, only one DLE character is entered in the input buffer. If the input buffer is full before the partner has started to terminate the link, the full buffer is transferred to the interpreter and the next characters arriving are entered in the second input buffer.

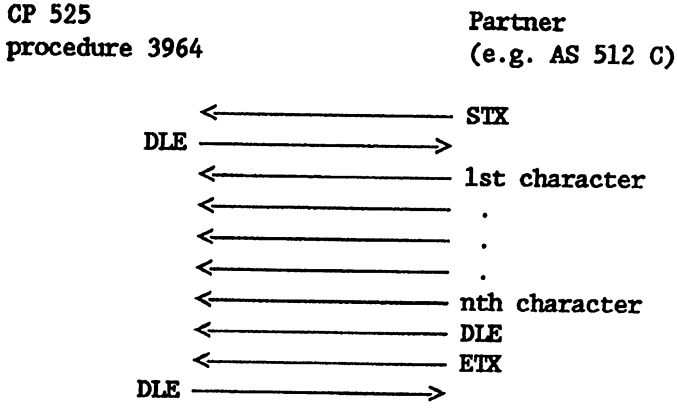
After each character received, the procedure waits for the next character to arrive within the character delay time. If there is a character timeout, the NAK character is sent to the partner and the error signalled to the interpreter.

If the procedure recognizes the character sequence DLE ETX, it stops receiving and sends DLE to the partner to indicate that the field was received without errors (or NAK with errors). The procedure transfers the contents of the input buffer to the interpreter and returns to the idle state.

If there is no empty input buffer available when the link is established with STX, a waiting time of 400 ms is started. If, after this time has elapsed, there is still no empty input buffer, the error is signalled to the interpreter. The procedure sends a NAK character and returns to the idle state. If an input buffer becomes available, the procedure sends the DLE character and receives the data as described above.

If transmission errors occur during the reception (characters lost, frame error, parity error), reception is continued until the link is terminated and then NAK is sent to the partner. Then a repetition is expected. If the field cannot be received without errors after six attempts, or the repetition or repetitions by the partner do not take place within a waiting time of 4 s, the 3964 procedure stops receiving and signals the error to the interpreter.

Example of a data exchange without errors:

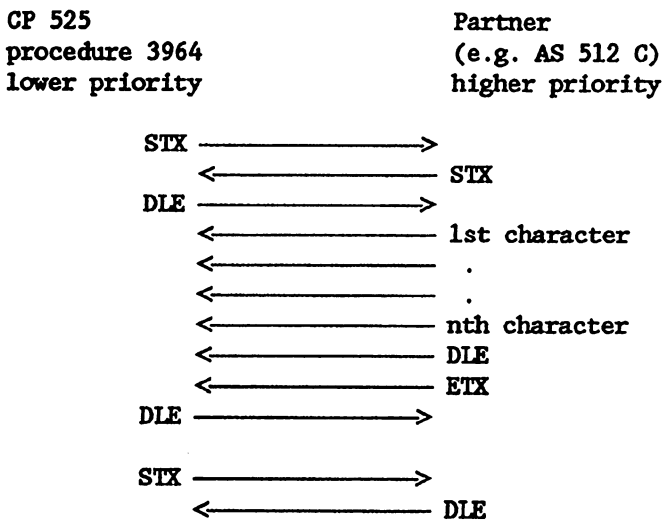


5.1.4 Initialization Conflict

If a device receives a send request (STX character) from its partner and does not answer within the acknowledgement delay time QVZ with DLE or NAK, but also sends the character STX, there is an initialization conflict. Both devices are attempting to carry out a send job. The device with the lower priority desists and answers with the DLE character. The device with the higher priority sends its data as described above. After the link has been terminated the lower priority device can then run its send job.

You set the "priority" parameter when writing the CP 525 user program with COM 525 in the ASSIGN PROC. PARA. mask. Remember that one device must be assigned higher, and one lower priority.

Example of resolving an initialzation conflict:



5.1.5 Procedure Errors

The procedure recognizes errors caused by the partner's reacting incorrectly, and errors caused by disturbances on the line.

In both cases an attempt is made to send or receive the data field correctly. The first repetition is signalled to the interpreter. If the data field cannot be sent or received within the maximum number of repetitions (or a new error occurs), the procedure breaks off the transmission or reception. It signals a specific error number to the interpreter for the first error detected and then goes to the idle state.

The interpreter enters the error number received from the procedure in the error message area of the SYSTAT. The significance of the individual error numbers can be found in the error tables in Section 8. If the error occurs during transmission for a PC job, the job status also has an error entry.

If you find that there is often an entry for send or receive repetitions in the error message area of the SYSTAT, but it does not have an exact error number, you can assume sporadic disturbances of the data exchange. These disturbances are compensated by the number of repetitions. You should check the transmission line for sources of interference, since the effective data rate and reliability of transmission are reduced by large numbers of repetitions. The cause of the problem could also be incorrect reaction by the partner.

Note that a link to an AS 512C module using a higher data rate occasionally causes repetitions.

Note:

- If there is a BREAK on the receive line, the BREAK error number is signalled to the interpreter immediately. No repetitions are started. The error signal remains active until the problem is eliminated.

- When transmission errors are detected (lost character, frame error, parity error), the same number is signalled, regardless of whether the error was detected when a data field was transmitted or received. The error is, however, signalled only when the repetitions are unsuccessful. If the corrupted character is received while the procedure is in the idle state, the error number is signalled to the interpreter immediately to indicate serious interference on the transmission line.

5.2 3964R Procedure

5.2.1 Procedure Data

The procedure 3964R controls the flow of data between your programmable controller and the partner, e.g., a central process com-puter, another CP 525, or an AS 512C. In the CP 525, the interpreter transfers the data to be sent into **output buffers** for the procedure. The procedure sends this data along with the transmission protocol 3964R to the partner; if necessary repeats the transmission, and signals non-recoverable errors to the interpreter.

Data from the partner is stored in **input buffers**. If the data is received without errors, it is transferred to the interpreter for further processing.

The 3964R procedure is an asynchronous, bit serial transfer procedure. The transmission rate both sending and receiving must be **the same** on the CP 525 and on the partner (asynchronous because there are no timing circuits between the two devices).

Control and information characters are transmitted on the link. To ensure that every character is recognized by the receiver and to allow the transmission to be checked for errors, the characters are encapsulated by further bits.

The order of the bits is as follows:

SA	I0	I1	I2	I3	I4	I5	I6	I7	PA	SO

SA = start bit
I* = information bit no.
PA = parity bit
SO = stop bit

The control characters for the 3964R procedure are taken from the DIN standard 66003 for the 7-bit code. However, the character length used is **8-bit** (bit 7 = 0). At the end of each data field a block check character (BCC) is added for data protection. The block check character BCC is the even longitudinal parity of the information bits of all data bytes of a field that has been sent or received. The characters begin with the first byte of information data following the establishment of the link and finish with the DLE ETX character when the link is terminated. No code is prescribed for the information characters (code transparency).

The following procedure parameters can be set when writing the CP 525 user program with COM 525:

Data rate

The speed of the data transmission is specified in bps. You can select the following values:

19200 bps *
9600 bps
4800 bps
2400 bps
1200 bps
600 bps
300 bps
150 bps
110 bps

(*) 19200 bps are only allowed if you use the V.24 interface with the CP 525/CP 524 or the RS422-A/485 interface with the CP 524. In this case you can only load and operate **one** interface with the CP 525 ince the maximum sum of the data rates is also 19200 bps.

Priority

To resolve initialization conflicts you can set the priority of the CP 525. At the partner the opposite priority must be set. You can choose between the following:

higher priority
lower priority

The following are **fixed** for the 3964R procedure:

character length: 8 bits
number of stop bits: 1
parity bit, i.e., checking for even parity

5.2.2 Transmitting with the 3964R Procedure

To establish the link, the 3964R procedure sends the control character STX. If the partner replies within the time allowed of 2000 ms (i.e., no QVZ = timeout), the procedure starts the transmission. If the partner answers with NAK, any other character (except DLE) or if there is a timeout, then the attempt to establish a link has failed. After a total of six unsuccessful attempts, the procedure gives up, signals the interpreter that there is an error establishing the link, and sends the character NAK to the partner.

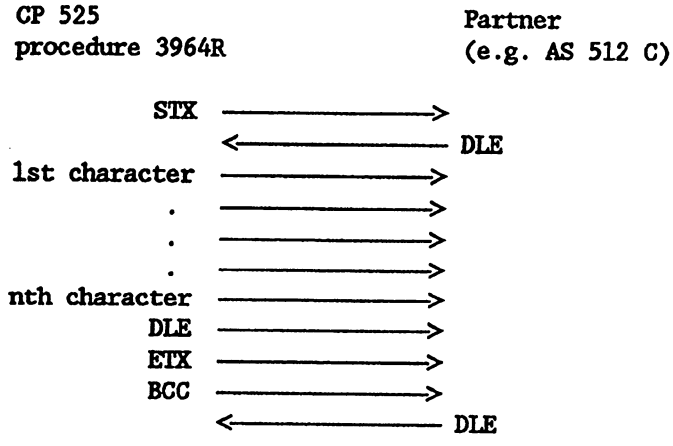
If the link is successfully established, the information data in the current output buffer is sent to the partner at the selected data rate. The partner monitors the time between the characters as it receives them. The time between two characters must not exceed the character delay time of 220 ms (ZVZ = character timeout).

Each DLE character found in the buffer is sent as two DLE characters (DLE doubling), i.e., the data (10H) is sent twice.

After sending the contents of the buffer, the procedure adds the characters DLE ETX and BCC as end identifier and waits for an acknowledgement. If the partner sends the character DLE within the time allowed (QVZ), the data field was received without errors. If the partner replies with NAK, any other character (except DLE), a corrupted character, or if there is a timeout, the procedure begins the link establishment again with STX. After a total of six unsuccessful attempts to send the data field, the procedure breaks off the attempt, signals the error to the interpreter and sends NAK to the partner.

If, while data is being transmitted, the partner sends the character NAK, the procedure breaks off the field and repeats it as previously described. If any other character is received, the procedure waits for the character delay time (ZVZ) to elapse and then sends NAK to reset the partner. Then the procedure begins the transmission again with the link establishment STX.

Example of a data exchange without errors:



5.2.3 Receiving with the 3964R Procedure

In the idle state, i.e., when there is no send job to be processed, the 3964R procedure waits for the link to be established by the partner.

If the procedure receives any character (except STX) while in the idle state, it waits until the character delay time (ZVZ) has elapsed and then sends a NAK character. The error is signalled to the interpreter.

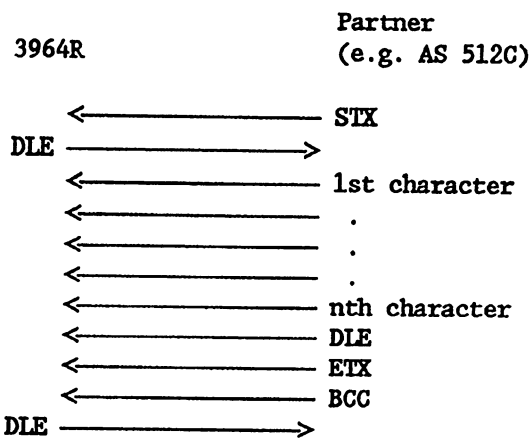
If the procedure receives the character STX and if it has an empty input buffer available, it answers with DLE. The characters then received are entered in the input buffer. If two DLE characters are received in succession, only one DLE character is entered in the input buffer. If the input buffer is full before the partner has started to terminate the link, the full buffer is transferred to the interpreter and the next characters arriving are entered in the second input buffer.

After each character received, the procedure waits for the next character to arrive within the character delay time. If there is a character timeout, the NAK character is sent to the partner and the error signalled to the interpreter.

If the procedure recognizes the character sequence DLE ETX and BCC, it stops receiving. It compares the received block check character BCC with the longitudinal parity calculated internally. If the block check character is correct and no other errors have occurred during reception, the procedure sends DLE. The procedure transfers the input buffer to the interpreter and returns to the idle state. If the BCC does not correspond, NAK is sent to the partner. Then a repetition is expected. If the block cannot be received without errors following six attempts or if the repetition by the partner does not take place within the waiting time of 4 s, the 3964R procedure breaks off the reception and signals the error to the interpreter.

If transmission errors occur during the reception (characters lost, frame error, parity error), reception is continued until the link is terminated and then NAK is sent to the partner. Then a repetition is expected as previously described.

Example of a data exchange without errors:

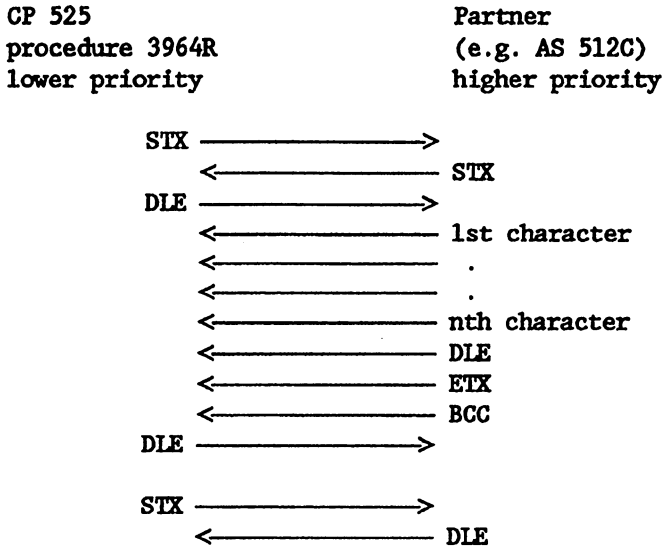


5.2.4 Initialization Conflict

If a device receives a send request (STX character) from its partner and does not answer within the acknowledgement delay time QVZ with DLE or NAK, but also sends the character STX, there is an initialization conflict. Both devices are attempting to perform a send job. The device with the lower priority desists and answers with the DLE character. The device with the higher priority sends its data as previously described. After the link has been terminated, the lower priority device can then run its send job.

The priority parameter is set when you are writing the CP 525 user program with COM 525 in the ASSIGN PROC. PARA. mask. Remember that one device must be assigned higher, and one lower, priority.

Example of resolving an initialization conflict:



5.2.5 Procedure Errors

The procedure recognizes errors caused by the partner reacting incorrectly, and errors caused by disturbances on the line.

In both cases an attempt is made to send or receive the data field correctly. The first repetition is signalled to the interpreter. If the data field cannot be sent or received within the maximum number of repetitions (or a new error occurs), the procedure breaks off the transmission or reception. It signals a specific error number to the interpreter for the first error detected and then goes to the idle state.

The interpreter enters the error number received from the procedure in the error message area of the SYSTAT. The significance of the individual error numbers can be found in the error tables in Section 8. If the error took place during transmission for a PC job, the job status also has an error entry.

If you find that there is often an entry for send or receive repetitions in the error message area of the SYSTAT, but there is no exact error number, you can assume sporadic disturbances of the data exchange. This is, however, compensated by the number of repetitions. In this case you should check the transmission line for sources of interference, since the effective data rate and reliability of transmission is reduced by large numbers of repetitions. The cause of the problem could also be incorrect reaction by the partner.

Note that a link to an AS 512C module using a higher data rate occasionally causes repetitions.

Note:

- If there is a BREAK on the receive line, the BREAK error number is signalled to the interpreter immediately. No repetitions are started. The error signal remains active until the problem is eliminated.

- When transmission errors are detected (lost character, frame error, parity error) the same number is signalled, regardless of whether the error was detected when a data field was transmitted or received. The error is only signalled when the repetitions are unsuccessful. If the corrupted character is received while the procedure is in the idle state, the error number is signalled to the interpreter immediately to indicate serious interference on the transmission line.

6 Multiprocessor Operation

Compared with single processor operation, multiprocessor operation requires even more that the programming of the individual CPUs (STEP 5 user programs) and of the CP 525 (CP 525 user programs) is carefully matched.

The more complex the application, the greater the care that must be taken to ensure the numbers of the individual CPUs are correctly specified in the CP 525 user program.

!IMPORTANT!

If several CPUs are to supply jobs to only one interface of a CP 525, the STEP 5 user programs of these CPUs must be coordinated.

Remember that a maximum of 10 jobs can be processed simultaneously on one interface. If an eleventh job is initiated, it is rejected with an error number in the PAFE byte. Another job can be initiated only when one of the ten jobs has been completed with or without errors.

If jobs with the same job number are programmed on more than one CPU the you should design the STEP 5 user programs so that at any one time only **one** CPU can start one of these jobs. All other CPUs must wait until this job is signalled as "completed" (with or without errors).

For more information on specific jobs see the corresponding sections in this user's guide.

7 Special Jobs Carried out by the CPU

Section 7 contains information about the following:

- the PSEUDO READ/WRITE function
- the error message area of the SYSTAT
- the identification area SYSID
- the reading and writing of the date and time

7.1 PSEUDO READ/WRITE Function

The READ/WRITE function (R/W function) with which you are familiar as a user of the handling blocks must **not** be used in connection with the CP 525. This is true for the following reasons:

- This R/W function does **not** allow the specification of a coordination flag and the CPU number.
- It overwrites part of the data area in the dual-port RAM of the CP 525, which currently contains data to be transmitted.

Therefore a **PSEUDO R/W function** is used, based essentially on the function and structure of the known R/W function.

The pseudo R/W function means that you can act as follows:

- can change source and/or destination specifications of a job dynamically while the program is running
- do not have to change source and/or destination specifications of a job dynamically while the program is running
- do not have to change source and/or destination specifications of a job dynamically while the program is running
- do not have to program this job with COM 525

All the parameters for the source and destination are stored in a data block (DB) or extended data block (DX).

The job numbers 190 to 199 are reserved for the PSEUDO READ/WRITE function.

A SEND job with the pseudo R/W function is started by one of the following HDBs:

SEND DIRECT 190 to SEND DIRECT 199

The pseudo R/W job is entered in the internal job queue of the CP 525 just as other direct jobs. When the job is processed the CP 525 uses the SEND ALL HDB to request the specified data block (or extended data block) with the assigned parameters and checks the source and destination parameters specified in it. If these are within the permitted limits, the CP 525 requests the first source data by means of a second SEND ALL HDB and then processes the SEND job as usual.

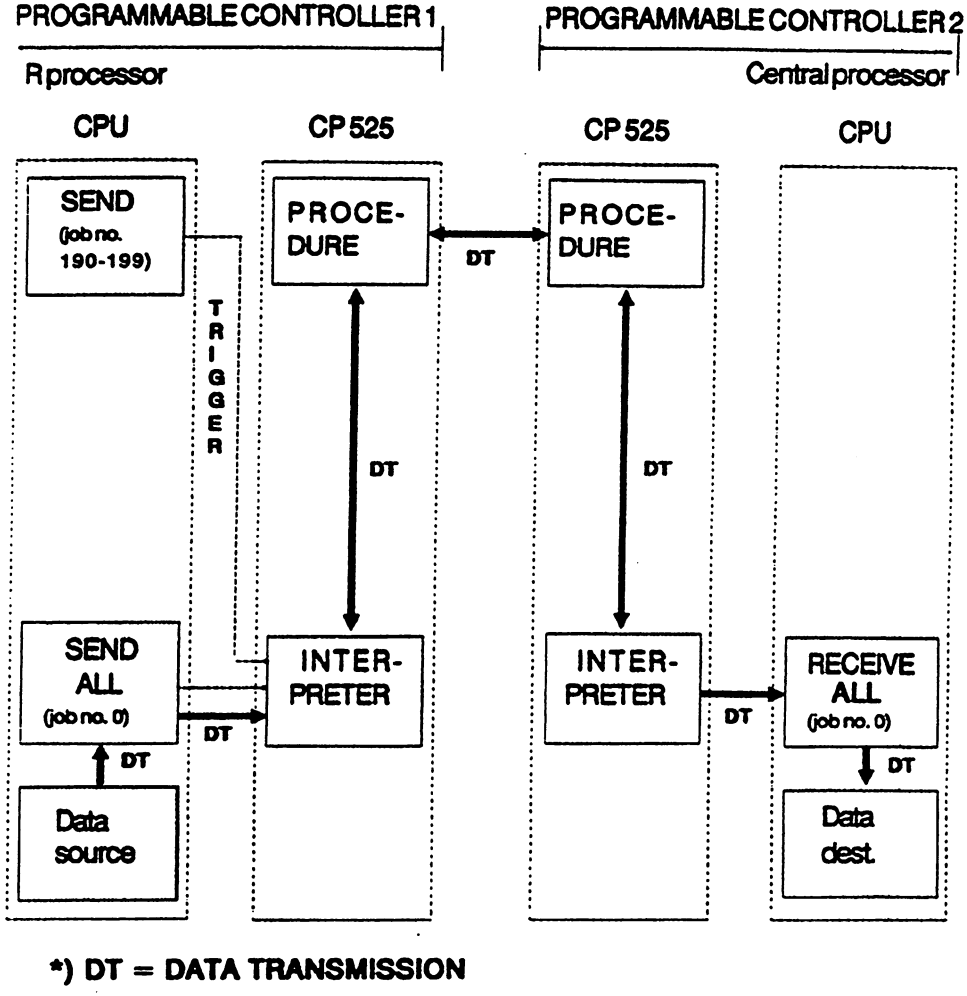


Fig. 1 WRITE function

A FETCH job with the PSEUDO R/W function is started by one of the following HDBs:

FETCH DIRECT 190 to FETCH DIRECT 199

As with the SEND job, the CPU fetches the specified data block using SEND ALL and evaluates the source/destination parameters. Then the FETCH telegram is sent to the partner. Once the data is received from the partner, the RECEIVE ALL HDB enters it in the specified destination.

In contrast to normal direct jobs, the PSEUDO R/W function requires an additional exchange of data between the CPU and CP. This data exchange is handled by the SEND ALL, which transfers the source/destination parameters to the CP 525.

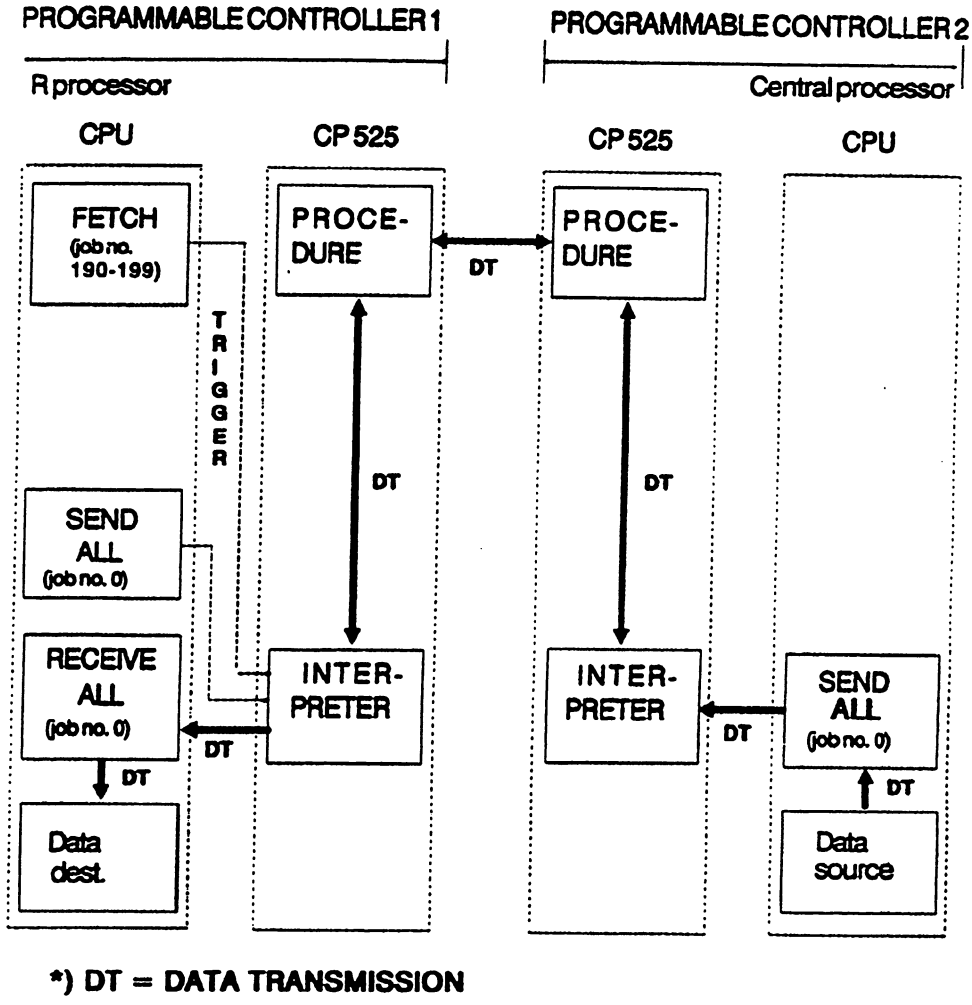


Fig. 2 READ function

The following pages show the structure of the data block with the source/destination parameters (parameter data block). The form and order are those you see on the PG. The letters x and y are variables. You must replace them with values.

The addresses of the data are relative to the start address QANF, which is specified in the HDB.

1.) Structure of the parameter data block for SEND DIRECT 190 to SEND DIRECT 199

```
QANF +
0:   KS= xx          ;source type (in own PC)
                      value range: = DB,DX,CB,TB,RS,AS,FY,QB
                                     IB,PB,OB
                                     not: XX,RW,NN
1:   KY=   0,   y;   ;source data block number
                      value range:y= 3 to 255 for source type
                                     DB/DX irrelevant for other
                                     source types
2:   KF=        x;   ;source start address
                      value range:x= dependent on source type
                                     and PC type (see job table
                                     in Section 2.6)
3:   KF=        x;   ;source length
                      value range:x= dependent on source type
                                     and PC type (see job table
                                     Section 2.6)
4:   KS= xx          ;dest. type (for partner)
                      value range:xx=DB,RS,AS,DX
                                     dependent on source type
                                     (see job tables in Section
                                     2.6)
5:   KY=   0,   y;   ;dest. data block number
                      value range:y= 3 to 255 for dest. type DB
                                     or DX irrelevant for other
                                     dest.types
6:   KF=        x;   ;dest. start address
                      value range:x= dependent on dest. type and
                                     PC type (see job tables in
                                     Section 2.6)
7:   KF=        x;   ;(dest. length) irrelevant!!
                      value range:x= any

8:   KY=   x,   y;   ;coord. flags (dependent on dest. type)
                      value range:x= 0 to 255: byte number
                                  y= 0 to   7: bit number
                      no coord. flags: x=255, y=255 !!!
9:   KF=        x;   ;CPU no. for partner
                      value range:x= 1 to 4 or
                      x= 0 for PC's with only one CPU
```

2.) Structure of the parameter data block for FETCH DIRECT 190 to FETCH DIRECT 199

```
ZANF +
0:   KS= xx          ;source type (in partner)
                      value range: = DB,DX,CB,TB,RS,AS,FY,QB
                                     IB,PB,OB
                                     not: XX,RW,NN
1:   KY=   0,   y;   ;source data bl. no. for source type DB, DX
                      value range:y= 3 to 255 for source type
                                     DB/DX irrelevant for other
                                     source types
2:   KF=        x;   ;source start address
                      value range:x= dependent on source type
                                     and PC type (see job table
                                     in Section 2.6)
3:   KF=        x;   ;(source length) irrelevant!!
                      value range:x= any

4:   KS= xx          ;dest. type (in own PC)
                      value range:xx=DB,RS,AS,DX
                                     dependent on source type
                                     (see job tables in Section
                                     2.6)
5:   KY=   0,   y;   ;dest. data block number
                      value range:y= 3 to 255 for dest. type DB
                                     or DX irrelevant for other
                                     dest. types
6:   KF=        x;   ;dest. start address
                      value range:x= dependent on dest. type and
                                     PC type (see job tables in
                                     Section 2.6)
7:   KF=        x;   ;dest. length
                      value range:x= dependent on dest. type
                                     and PC type (see job table
                                     Section 2.6)
8:   KY=   x,   y;   ;coord. flags (dependent on dest. type)
                      value range:x= 0 to 255: byte number
                                  y= 0 to   7: bit number
                      no coord. flags: x=255, y=255 !!!
9:   KF=        x;   ;CPU no. for partner
                      value range:x= 1 to 4 or
                      x= 0 for PC's with only one CPU
```

7

Transmissions of possibilities can be seen in the job table in Section 2.6.

For SEND DIRECT 190 to 199 the data source is in your own PC. You must specify the source length in the parameter data block (enter this in bytes if the QTYP is a byte area). The specification of the dest. length (ZLAE) is unnecessary. All specifications of the data destination refer to the partner.

With FETCH DIRECT 190 to 199 the data destination is in your own PC. Therefore the specification of the length ZLAE in the parameter data block is vital. Any specification of the parameter QLAE is not evaluated. All the parameters involving the data source refer to the partner.

The jobs SEND DIRECT 190 to 199 and FETCH 190 to 199 are called in the STEP 5 program just as "normal" jobs with "JU". In the specifications in the block : QTYP/ZTYP, DB no., QANF/ZANF, QLAE/ZLAE, you name your data block with the RW parameters. The specification of QLAE/ZLAE is unnecessary since the CP 525 always reads 10 data words (with SEND ALL).

If no more than 10 jobs are reported as running simultaneously, the pseudo R/W jobs can be called mixed in with normal SEND and FETCH jobs. If more than 10 jobs are called, the initiation of the jobs must be interlocked.

Since each job number can be assigned only once, it is not possible to activate a SEND DIRECT 190 and a FETCH DIRECT 190 at the same time. If the SEND DIRECT job 190 is set to running in the condition codeword, no other job with the number 190 (e.g., FETCH 190) can be started until SEND DIRECT 190 is completed with or without errors.

7.2 Reading the Error Message Area of the SYSTAT

The error message area of the SYSTAT is a data area in the dual-port RAM of the CP 525. After the CP 525 recognizes an error, it enters the corresponding error number in the error message area of the SYSTAT. You decide how the CPU is to react in your STEP 5 user program on the CPU.

The CPU reads the **error message area** of the SYSTAT for the specified CP interface by means of the job

RECEIVE DIRECT 200

The area in the CPU (destination area) to which the error message area of the SYSTAT is to be transferred, is fixed when you assign parameters to RECEIVE DIRECT 200. These parameters should be a data block. The destination length must be at least two words.

RECEIVE DIRECT 200 is executed only when an error number is entered in the error message area of the SYSTAT, since only then is the bit "RECEIVE - job ready" set in the ANZW (bit 0).

Note: in some cases (e.g., if no SYNCHRON has been run or if start-up errors have occurred in the CP 525), the error message area cannot be read by RECEIVE DIRECT 200 (cf. note in Section 7.4 Reading the whole SYSTAT).

Structure of the destination area:

If error numbers are entered in the error message area they are located (after they have been read by RECEIVE DIRECT 200) in the selected area in the CPU (e.g., in the data block).

		Bit 7	6	5	4	3	2	1	0
Byte	0 (DL0)	-				E	0	B	R
	1 (DR0)	Error number 1							
	2 (DL1)	Error number 2							
	3 (DR1)	Error number 3							

This diagram is explained on the following page.

Byte	Bit	Status	Significance
byte 0	bit 3 (E)	0	no error in SYSTAT (not following executed RECEIVE DIRECT 200)
		1	error entered in SYSTAT note: this bit is always set if there is an error entry - i.e., use this bit to check whether data was transferred with RECEIVE DIRECT 200
	bit 2 (O)	0	no error overflow (max. 3 error entries)
		1	error overflow (more than 3 errors)
	bit 1 (B)	0	no BREAK currently on the interface
		1	BREAK on the interface note: bit 1 shows only the instantaneous status and does not always mean an error. The error status BREAK is determined only by the error number in DR0, DL1 and DR1. The current BREAK status can only be indicated if the interpreter and procedure are loaded and have been started.
	bit 0 (R)		unnecessary for computer link
	bit 7, 6, 5, 4		unnecessary
byte 1			error number 1
byte 2			error number 2
byte 3			error number 3 00 = no error

For more information on error messages refer to Section 8.

Example of a command sequence within a STEP 5 user program for the CPU 992 (R processor) / S5-135U:

```
     : .
     :O   F 1.0        ; to initiate the reading out from SYSTAT
     :ON  F 1.0        ; result of logic operation must be "1"
     :
     :JU  FB121
     NAME:RECEIVE      ; data transfer only in error entry
     SSNR: KY0,0       ; interface number 0
     A-NR: KY0,200     ; fixed job number for reading SYSTAT
     ANZW: FW10
     ZTYP: KSDB
     DBNR: KY0,10      ; dest. for error numbers is DB10
     ZANF: KF+1        ; from data word 1
     ZLAE: KF+2        ; 2 data words
     PAFE: FY5
     :
     :C   DB10
     :AN  D 1.11       ; test bit 3 of group byte
     :JC  =M001        ; jump if no error entered
     :.
     :.                ; error evaluation
     :.
     :.
     :JU  FB124        ; an error was entered, i.e., the SYSTAT
     NAME:RESET        ; register on the CP 525 should be reset
     SSNR: KY0,0       ; with RESET DIRECT 200 (see following
     A-NR: KY0,200       pages)
     PAFE: FY6
     :
     :C   DB10         ; reset error entry in DB
     :L   KH0000
     :T   DW1
     :T   DW2
     :
M001:.
     :etc.
     :BE
```

7.3 Resetting (Clearing) the Error Message Area

The CPU resets the error message area of the SYSTAT (based on interface numbers) with the job

RESET DIRECT 200

All the entries are then cleared except the BREAK bit (B).

!IMPORTANT!: the error message area of the SYSTAT is not cleared by a SYNCHRON call.

The CP 525 can enter up to three error numbers in the error message area. More errors cannot be signalled unless the "old" entries have been cleared. Therefore the error message area must be reset in good time by a RESET DIRECT 200.

The destination area in the CPU must also be reset.

7.4 Reading the Whole SYSTAT

In addition to the reading of the error message area of the SYSTAT, the entire SYSTAT of the CP can be read by means of the job

RECEIVE DIRECT 221

The area read then contains the error messages for both interfaces (in bytes 4 to 10). This job can be executed only if there are no error numbers entered. A destination length greater than or equal to 16 bytes (8 words) is necessary.

Note: in some cases (e.g. no SYNCHRON executed, or error during the start-up of the CP 525) the SYSTAT cannot be read by means of RECEIVE DIRECT 221. It must then be read directly from the dual-port RAM (from address F660H onwards).

The area transferred with this job is structured as follows:

Byte	Bit 7	6	5	4	3	2	1	0
0	irrelevant						RUN	STOP
1	irrelevant							
2	"sign of life"							
3	irrelevant							PGR
4	E2	02	B2	R2	E1	01	B1	R1
5	error number 1					IF1		
6	error number 2					IF1		
7	error number 3					IF1		
8	error number 1					IF2		
9	error number 2					IF2		
10	error number 3					IF2		
11 12 13 14 15	reserved							

The individual bits or bytes have the following significance:

- RUN = 1 mode selector switched to RUN
- STOP = 1 mode selector switched to STOP
- PGR = 1 mode selector switched to PGR
- "Sign of life", this byte is incremented by 8 at regular intervals on the CP

- Bytes 4 to 10: cover the error message area for both interfaces (see Section 7.2 - Reading the error message area of the SYSTAT).

- Error message area for IF1: bytes 5 to 7 and byte 4 (bits 0 to 3)
- Error message area for IF2: bytes 8 to 10 and byte 4 (bits 4 to 7)

7.5 Reading the SYSID

An identification area is specified on each CP 525. Information is written into the identification area SYSID of the CP 525; i.e., you can find information about the following:

- module
- memory submodule
- interface assignment
- issue/version of the loaded software etc.

The CPU can read the SYSID of the CP 525 by means of the job

RECEIVE DIRECT 223

This is possible if the field length is greater than or equal to 128 bytes; i.e., the BLGR parameter in the SYNCHRON is KY 0,4 / KY 0,5 / KY 0,6.

A destination length greater than or equal to 128 bytes is also necessary.

If the field length is less than 128 bytes the message "completed with errors" and the error number 7 are entered in the status byte and the number 1E in the error message area of the SYSTAT. The SYSID area is not then transferred.

Note: the SYSID can also be read directly from the dual-port RAM, i.e., without handling blocks (from address F680H onwards). For more information refer to the manual for your PC.

Structure of the identification area SYSID

The SYSID contains a number of parameters which identify the CP. Each parameter is completed with carriage return (CR = 0DH) that is also counted in the byte specification. Parameters not assigned consist simply of CR. Some parameters have a different significance depending on the memory submodule types. All the parameters are represented in ASCII.

With the CP 525 these parameters are structured as follows:

Parameter	Byte	Description
0	0 to 9	Mem. submodule identifier/capacity: EPROM: e.g. for order number 6ES5373-0AA41 shown as:373-0AA41 RAM: memory capacity of the submod. RAM xxxKw e.g. xxx = 64
1	10 to 18	Module identifier: CP 525-2 / CP 524
2	19 to 21	Version of the CP firmware
3	22 to 41	Plant designation from mem. submodule: EPROM: designation of programmed plant RAM: replaced by space (20H)
4	42 to 50	Generated on: EPROM: date of EPROM- programming RAM: replaced by space (20H)
5 to 9	51 to 55	Not used with CP 525/CP 524
10	56 to 64	PROM 525 - version: PROM 525 = programming program for EPROM submodules EPROM: PROM 525 - version RAM: replaced by space (20H)

Interface 1

Parameter	Byte	Description	Example
11	65 to 67	component type	RK
12	68 to 76	interpreter name (*)	RK512
13	77 to 79	interpreter version	nn
14	80 to 88	procedure name (*)	P3964R
15	89 to 91	procedure version	nn

Interface 2

Parameter	Byte	Description	Example
16	92 to 94	component type	PT
17	95 to 103	interpreter name (*)	PT88
18	104 to 106	interpreter version	nn
19	107 to 115	procedure name (*)	LAUFPT88
20	116 to 118	procedure version	nn
	119	end identifier ETX (03H)	

nn = version number
* = name of interpreter and procedure correspond to the designations in the COM library

If the interface is not assigned, the parameters specific to the interface are replaced by spaces (20H).

7.6 CP 525: Reading and Writing the Date and Time

A CP 525 must be designated either time master or time slave.

Time slave: after powering up the CP reacts at first as a time slave. It can update its own date and time, however, the CPU cannot read them.

Time master: the master provides the system time; i.e., the master must make this available at any time.

To synchronize the CPs (at regular intervals or when certain events occur) the CPU reads the system time from the time master and sets all the time slaves to this time. This relieves the time slaves of unnecessary updating tasks.

The CPU can set the date and time on a CP 525 at any time. It can, however, only read the date and time of a CP 525 that is designated as time master.

Note:
The CP 524 does not have a hardware clock.
When using the CP 524, the software clock is not updated or buffered if the PLC is set to STOP or the mains voltage is switched off. Thus, the CP 524 clock must be updated when the CPU is restarted (cold restart).

The date/time area transferred during writing or reading is structured as follows:

Byte	Significance
0	bit 0: master id. 1 = master 0 = slave
1	0
2	$^1/_{10}$ s $^1/_{100}$ s
3	s
4	min
5	h
6	day
7	month
8	year
9	0

In a word transfer byte, 0 corresponds to the left half of the first data word; byte 1 the right half of the first data word. The date and time are coded in BCD.

The accuracy of the clock is $^1/_{10}$ s. This means that the position $^1/100$ s in byte 2 of the date/time area is always 0. When the time is written, the $^1/100$ s is not accepted.

The job number 218 is fixed on the CP 525 for the jobs "read / write date and time".

SEND DIRECT 218

Using this job, the CPU declares the CP as master or slave. At the same time the CPU can also transfer the date and time to the CP.

Whether the CPU transfers the date and time depends on the source length specified (QLAE = length of the data to be transferred).

QLAE = 1 word

The CP 525 is declared as master/slave by means of the contents of bit 0 (least significant bit) in the first byte; master: bit 0 = 1 / slave: bit 0 = 0.

During the start-up or in a warm restart of the CPU, the SEND DIRECT 218 with QLAE = 1 and master bit = 1 for the CP that is to be master must be called up. The date and time on the CP 525 are not changed by QLAE = 1.

QLAE = 5 words

The CP is declared as master/slave by means of the master identifier and the date and time are transferred.

Remember to set the master identifier when transferring the time to a master which is to remain as master.

Also remember that the date and time are only adopted by the CP during the handshake (synchronization via inputs or after reading a different clock).

Note: the source area from which the data is transferred to the CP should be a data block.

Note:

The CP 525 can be declared as time master on one or both interfaces. The date and time can only be read on the interface on which the CP was declared as master.

If the CP is declared as master on an interface and then as slave on the other interface, the CP becomes a time slave on both interfaces.

For example, the following situation occurs: when the date/time from the first interface is read, the error number 1F is entered in SYSTAT. When reading from the second interface the handshake is not performed, since the bit "RECEIVE - job ready" in the ANZW (bit 0) is not set. No error message is output.

A SYNCHRON does not influence the master-slave function.

RECEIVE DIRECT 218

By means of this job the CPU can read the date and time from the CP (if the CP has already been declared time master with SEND DIRECT 218.

The destination length must be five words. The destination area should be a data block.

Note: the data read only has meaning if the date and time have been set at least once since the module was plugged in (with SEND DIRECT 218, length: five words).

8 Error Handling

Errors occurring while the computer link is in operation, may have been caused at various points.

If the CP detects the errors, they are signalled to the CPU with as much detail as possible. The way in which the CPU then reacts is decided by your STEP 5 user program on the CPU (for more details see Section 7).

For example you can have the following:

- an error message output on the printer PT88/PT89 (either via another CP 525 or via the same CP 525 using the other interface)
- the error displayed as messages on a monitor (e.g. with the CP 526)
- the job repeated

The error is signalled by the following:

- an error number in the **PAFE byte** of the handling block (1.)
- an error number in the condition codeword **ANZW** of the handling block (2.)
- an error number in the error message area of the **SYSTAT** (3.)
- the CP 525 signals an error to the partner by an error number in the reply telegram in **REPTEL** (4.)

1. The error numbers in the PAFE byte of the handling block (HDB)

The PAFE byte is a parameter you specify when calling an HDB. In the PAFE byte, the handling blocks indicate when serious errors occur in connection with a CP or when parameters have been specified incorrectly. A detailed description of PAFE byte error messages can be found in the description "Using the handling blocks" in this manual.

If you receive a PAFE error number, this means that the data exchange between the CPU and CP either was not started or was aborted after several unsuccessful attempts.

In the following exceptional cases the error numbers 91H and C1H can mean that a serious error has occurred:

- after a program section is transferred from the PG to the CP 525
- if coordination flags have been used incorrectly

!IMPORTANT!: if these situations occur on your system you should first repeat the corresponding job.

2. The error numbers in the condition codeword ANZW

The condition codeword ANZW is also a parameter you specify when calling an HDB. With direct jobs (SEND DIRECT, FETCH DIRECT, RECEIVE DIRECT), ANZW indicates whether the job is running or was completed.

If the job was terminated with an error, an error number is entered in **bits 8 to 11** in the ANZW. Bit 3 is also set: "job terminated with error".

!IMPORTANT!: an error number is only entered in ANZW if the error occurred during the processing of a DIRECT job!

Since only four bits are available for the error numbers in ANZW, one error number normally indicates one of several error states. Owing to the limited number of error numbers the ANZW is suitable for evaluation by the STEP 5 user program.

The errors are distinguished by the various sources of error as follows:

No. for ANZW	Source
1 to 5	parameter assignment error recognised by the CPU of the PC and signalled to the CP
6	error in data exchange between CPU and CP detected by CP
7	job cannot be carried out, error in job parameters
9,A,B,C,D	error during the execution of a job, caused by incorrect reaction or error messages from the partner
E, F	error in data exchange between CP 525 and partner

Descriptions of the errors can be found in the tables on the following pages.

Note: a status byte is reserved in the dual-port RAM for each of the possible jobs. If the CP detects an error in connection with a DIRECT job, it writes the error number and the bit "job terminated with error" in the status byte of the corresponding job in the dual-port RAM. To have the job status shown in the condition codeword one of the following must occur:

- the CONTROL HDB in the STEP 5 program must be called with the corresponding job number in the cycle (OB1)
- the DIRECT job (SEND DIRECT) must be called with the command JU and RLO=0.

See also the description "Using the handling blocks" in this manual.

3. The error numbers in the error message area of the SYSTAT

The error message area of the SYSTAT is a data area in the dual-port RAM; the CPU reads this area using the special job RECEIVE DIRECT 200. For each interface the error message area of the SYSTAT includes three error message bytes and an additional status byte. A detailed description can be found in Section 7 "Special jobs carried out by the CPU".

All errors recognized by the CP are entered in the error message area of the SYSTAT. If the error occurs in connection with a DIRECT job, an error number is also entered in the condition codeword. If the error occurs in connection with a partner job, an error number is transferred in the reply telegram to the partner.

The error entry in the SYSTAT is so detailed that it is particularly suitable for exact error analysis during commissioning.

A programmed evaluation of individual error numbers is also possible. Remember that not only serious errors are entered in the SYSTAT. For example, an entry is made in SYSTAT when the procedure repeats parts of the telegram exchange. A repetition is not an error but simply points to minor problems in the telegram exchange that can be corrected by repetitions.

The special job RECEIVE DIRECT 200 should be called up in OB1 to read the error message area in the SYSTAT during the commissioning phase. You can recognize the occurrence of errors when bit 0 in the ANZW of the RECEIVE DIRECT 200 is set to 1 by the CP and the error numbers are then copied into the data block specified as a parameter. If a programmer is connected to the CPU of the PC, you can display the errors directly on the monitor in the operating mode "STATUS variable" or "CONTROL variable".

To clear the error entries in the SYSTAT, call the special job RESET DIRECT 200 (for more details see Section 7).

The errors are distinguished by the various error sources as follows:

No. for SYSTAT	Source
1 to 6	parameter assignment errors detected by the CPU of the PC, and signalled to the CP
10 to 14	errors in the data exchange between CPU and CP detected by the CP
15 to 1F	system errors at the beginning of the job processing
20 to 2F	errors in the processing of a PC job, detected by the CP
30 to 3A	errors in the processing of a PC job, detected by the partner and signalled to the CP by means of an error number in the reply telegram
40 to 53	errors during the processing of a partner job, recognised by the CP
5E to 5F	reception of an uninterpretable telegram
B0 to B8	error in data exchange between PG and CP
C0 to CA	hardware error on the module
D0 to FF	error in the data exchange between CP and partner

4. Error numbers in reply telegrams

If an error occurs in connection with a SEND or FETCH telegram from the partner, an error number is entered in the fourth byte of the reply telegram. The partner job is then terminated.

Note:

The numbers of the error messages are specified as hexadecimal values.

The following tables contain a detailed description of the error numbers

- arranged according to the numbers for ANZW
- arranged according to the numbers for SYSTAT
- arranged according to the numbers for REPTEL

and **how you can remedy the problem.**

The tables generally contain three columns with error numbers (no. for ANZW/SYSTAT/REPTEL). The error numbers in the second and third columns correspond to the number in the first column.

Note that error numbers in ANZW and REPTEL are entered only if the job is running at the time when the error occurs. For example:

Case 1: an interference pulse is generated on the transmission line during a break in transmission; the procedure recognizes a corrupted character. It therefore generates the error code FEH which is entered in SYSTAT. The ANZW is not changed since the error did not occur during the actual transmission.

Case 2: the interference pulse occurs while a PC job is being sent and the repetitions do not succeed in achieving an error-free transfer; an error number (EH) is entered in the ANZW as well as the entry (FEH) in the SYSTAT.

8.1 Error Messages for SYSTAT, ANZW and REPTEL Arranged According to the Numbers for ANZW

No. for ANZW	No. for SYSTAT	No. for REPTEL	Description
			Parameter assignment errors detected by the CPU of the PC, and signalled by the CP
1H	01H	0AH	a) Source/dest. type (e.g. ext. DB) illegal area (start address, length) illegal (negative value)
2H	02H	14H 0CH	b) DB/DX not present or illegal (e.g. DB0, DB1, DX0) or other data type not present or illegal
3H	03H	14H 0CH	c) DB/DX too short: (start address + length > area or area too short with other data type (start address + length > area
4H	04H	0AH	d) Access to area not possible for user: e.g. hardware memory too small or I/O module not plugged in
5H	05H	0CH	e) Wrong condition codeword: group message for all errors attributable to this ANZW
6H	06H	0AH	Error code from the CPU to the CP, which the CP cannot interpret

Remedy

- Check parameter assignment on CPU and CP and if necessary correct or
- Partner has illegal parameters in the telegram header

Find the permissible data types in the job tables (Section 2.6).

- Check parameter assignment on CPU and CP, if necessary set up block
- Partner has wrong parameters in the telegram header

Find the permissible data types in the job tables (Section 2.6).

- Check parameter assignment on CPU and CP if necessary check block/area or
- Partner has wrong parameters in the telegram header

7

- Check parameter assignment on CPU and CP or
- Partner has wrong parameters in the telegram header

Find the permissible start addresses and lengths in the job tables (Section 2.6)

Check whether condition codeword is valid.

Evaluate PAFE in handling block (STEP 5 user program).

No. for ANZW	No. for SYSTAT	No. for REPTEL	Description
			Errors detected by CP during data exchange between CPU and CP:
6H	10H	0AH	a) Timeout for handshake on CP
	11H		b) ALL job without request or direct job despite overload
	12H		c) Wrong sequence for handshake by CPU
	13H		d) Unknown job type or illegal job number for direct job
	14H		e) Illegal acknowledgement from CPU during handshake

Remedy

Check whether RECEIVE ALL or SEND ALL are called in your STEP 5 program; evaluate the PAFE in the handling block.

Check STEP 5 program, e.g., errors in handling block
(Message also possible after transfer from PG to CP).

Check STEP 5 program, e.g., errors in handling block.

Check STEP 5 program, e.g., wrong parameters for handling block.

Check STEP 5 program.

7

No. for ANZW	No. for SYSTAT	No. for REPTEL	Description
			System errors at start of job processing:
7H	15H		a) Number of direct jobs which can be processed simultaneously too large for CP 524/CP 525
	16H		b) Current job was terminated with error status when CP was cold restarted (power on) or cold restarted from PG
	17H		c) Area is masked by STEP 5 program (bit 7 in ANZW = 1)
	1EH		d) Field length for "read SYSID" (RECEIVE DIRECT 223) too small
	1FH		e) "Read date/time" (RECEIVE DIRECT 218) illegal, CP is set as time slave

Remedy

Modify STEP 5 program so that maximum 10 direct jobs run simultaneously.

There is no remedy with power on. When cold restarting the CP from the PG, you should make sure that no more CPU jobs are running before writing to an interface. The SYNCHRON HDB clears the job statuses in the dual-port-RAM but not the SYSTAT entry.

Reset bit 7 in the ANZW.

For SYNCHRON the field length should be greater than or equal to 128 bytes or do without "Read SYSID" (data is nevertheless available in the dual-port RAM).

Before reading the date/time, the CP must be designated as master - this then applies to this module. Check whether the CP on the other interface has been set to master.

No. for ANZW	No. for SYSTAT	No.from REPTEL	Description
			Parameter ass. errors in the STEP 5 prog. detected by the CP (e.g. in the pseudo R/W parameters) or job block (CP 525 user program)
7H	20H		a) No parameters were stored in the CP 525 user program for this direct job (under this job number)
	21H		b) Error in data type (DB, OB...): - unknown data type or - data type not allowed (e.g. ext. DB) or - data type not allowed in HDB parameters in connection with data type in the job block (CP 525 user program)
	22H		c) Start address specified too high for required data type or start address or DB/DX no. too low
	23H		d) Coordination flags not allowed with this data type or wrong bit number specified
	24H		e) CPU number specified too high in pseudo R/W parameters
	25H		f) Transmission length > 64 Kbytes too long for CP 525
	26H		g) Length of telegram header too great or length too great for type specified (in own PC)

Remedy

Create parameters for this job with COM 525 or select other job number.

Find the permissible data types and their combinations in the job tables (Section 2.6).

Find the permissible start addresses and DB/DX nos. in the job tables (Section 2.6) that can be specified in the CP 525 user program or STEP 5 program (with pseudo R/W parameters).

Coordination flags are not permissible for all data types (see Section 2.6). Check whether the bit number (0 to 7) has been specified correctly when using the pseudo R/W function.

CPU nos. permitted: none, 1, 2, 3 or 4

If necessary split the job into several jobs with smaller lengths.

Find the permissible lengths in the job tables (Section 2.6) that can be specified in the HDB or for the pseudo R/W parameters. If necessary split the job into several jobs with smaller lengths.

No. for ANZW	No. for SYSTAT	No. for REPTEL	Description
			If the partner detects an error when processing a job from another device, it sends a reply telegram back to the CP 525 with the error number in the 4th byte. The CP generates a no. for SYSTAT and for ANZW from this:
9H	30H	No.from REPTEL 32H	a) DB / DX disabled by coordination flags
AH	31H	No.from REPTEL 0AH	b) Hardware (DMA) error at the partner or with CP 525 as partner: - source/dest. type illegal - or memory error in partner PC - or error in handshake between CP/CPU in partner or - partner PC is in STOP state
	32H	No.from REPTEL 0CH	c) Memory access error at partner (memory not present) with CP 525 as partner: - wrong area for condition code-word or - data area does not exist (except DB/DX) or - data area too small (except DB/DX)

Remedy

In the partner program after processing the last transmission data reset the coordination flags.
In the STEP 5 program repeat the job!

- Check whether the partner can transfer the required data type.
- Check partner's hardware configuration.
- Switch mode selector at partner PC to RUN.

Check whether the required data area exists on the partner and whether it is long enough, or check the COM 525 job block or the pseudo R/W parameters. Check the length specified in the HDB.

No. for ANZW	No. for SYSTAT	No. from REPTEL	Description
AH	33H	14H	d) DB/DX access error at partner (DB/DX does not exist or too short) with CP 525 as partner: - DB/DX does not exist or - DB/DX too short or - DB/DX no. illegal
BH	2AH		**Errors in reply telegram from partner, detected by CP** a) Error in format of reply telegram received: first byte not 00H or FFH
	2BH		b) Reply telegram received after FETCH job has too many data
	2CH		c) Reply telegram received after FETCH job has too few data
	2DH		d) Reply telegram received after SEND job arrived with data

Remedy

Check whether the required data block exists on the partner and whether it is long enough; or check the job block (CP 525 user program); or check the pseudo R/W parameters (STEP 5 user program). Check the length specified in the HDB.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for ANZW	No. for SYSTAT	No.from REPTEL	Description
			If the partner detects an error when processing a job from another device, it sends a reply telegram back to the CP 525 with the error number in the 4th byte. The CP generates a no. for SYSTAT and for ANZW from this:
BH	34H	16H	a) Errors in the telegram header, detected by partner: first command letter in header wrong
	35H	10H	b) Error in telegram header: second command letter in header wrong
	36H	34H	c) Partner detects wrong telegram length (total length)
	39H	12H	d) Partner signals "system command illegal"
	3AH	??H	e) Unknown error number received in reply telegram
	5FH	34H	A correct telegram with a length > 128 bytes was received. The telegram could not be interpreted. The job being processed is aborted.

Remedy

Check whether the error is being caused by a disturbance or an incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

Check whether the error is being caused by a disturbance or an incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

Check whether the error is being caused by a disturbance or an incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

Incorrect reaction by partner; the CP 525 never outputs a system command.

Check whether the error is being caused by a disturbance or an incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for ANZW	No. for SYSTAT	No.from REPTEL	Description
CH	27H		Selector on STOP/PGR during PC job
CH	29H		**Errors in the reply telegram from the partner, detected by the CP:** a) Synchron error by partner, since - reply telegram arrived, though no job running or - the (follow-on) FETCH telegram has not yet been sent or - a follow-on reply telegram was received after a FETCH telegram was sent or - a reply telegram was received after a follow-on FETCH telegram was sent
	37H	36H	**If the partner detects an error when processing a job from another device, it sends a reply telegram back to the CP 525 with the error number in the 4th byte.** **The CP generates a no. for SYSTAT and for ANZW from this:** a) Partner detects synchron error - order of telegrams is wrong

Remedy

Switch the selector to RUN

This error can be signalled following a cold restart at your PC or at the partner. It is a normal reaction during start-up; you do not need to do anything. During normal operation the error can also occur as a result of older error statuses that were detected by the partner. Otherwise you can assume that the partner is reacting incorrectly. It is possible that the error will not occur with jobs < 128 bytes.

7

This error can be signalled during a cold restart at your PC or at the partner. It is a normal reaction during start-up; you do not need to do anything. During normal operation the error can also occur as a result of older error statuses that were detected by the partner. Otherwise you can assume that the partner is reacting incorrectly.

No. for ANZW	No. for SYSTAT	No.from REPTEL	Description
CH	38H	2AH	b) Up to now there has been no cold restart at the partner. with CP 525 as partner: Since the power up no "SYNCHRON" HDB has run or the mode selector is switched to STOP/PGR.
DH	2FH		No reply telegram came from the partner within the monitoring time after a FETCH telegram was sent.

Remedy

Carry out a cold restsrt at the partner PC or switch the selector on the CP to RUN.

Is the partner a very slow device?
This error is often indicated as the result of an older error. For example, procedure receive errors (F0H to FFH) can be indicated after a FETCH telegram has been sent because the reply telegram could not be received owing to a disturbance —> there is a timeout. This error can sometimes occur when a cold restart has been carried out at the partner before it could reply to the last FETCH telegram received.

No. for ANZW	No. for SYSTAT	Description
		Errors in data exchange CP <—> partner:
EH		**Errors detected by the procedure during the sending of a telegram and signalled to the interpreter:**
	E1H	a) Errors during the link establishment - after STX was sent, NAK or some other character (except DLE or STX) was received or - the reply came too soon or - an initialization conflict has occurred
	E2H	b) Timeout (QVZ): after sending STX the partner did not reply within the acknowledgement time

Remedy

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Partner is too slow or not ready to receive or there is a line break. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for ANZW	No. for SYSTAT	Description
EH	E3H	c) Aborted by partner: during transmission one or more characters were received from the partner (incl. NAK)
	E4H	d) Errors at end of link: - the telegram was rejected at the end by the partner with NAK or some other character (except DLE) or - the acknowledgement character was received too soon
	E5H	e) Timeout at end of link: after link termination with DLE ETX no reply was received from the partner within the QVZ time
	FEH	Transmission error: a transmission error (parity error, stop bit error, overflow error) was detected. If this occurred during transmission or reception, repetitions will be started. If a corrupted character is received in the idle state, the error is signalled immediately to ensure that disturbances on the line are detected as early as possible.

Remedy

- Check whether the partner is also indicating an error, it is possible that not all the data sent have arrived (e.g., break in the line) or
- Partner is reacting incorrectly. Verify this if necessary with an interface test device ('FOXPG') tapped into the transmission line.

- Check whether the partner is also indicating an error; it is possible that serious disturbances have occurred or
- Partner is reacting incorrectly. Verify this if necessary with an interface test device ('FOXPG') tapped into the transmission line.

- Partner is too slow or has a fault. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

- Disturbances on the line result in repetitions of the telegram and threfore reduce the actual data throughput. The chance of an undetected error increases. Change the configuration or re-lay the cable.
- Check whether the data rates, parity and number of stop bits are the same at both devices.

No. for ANZW	No. for SYSTAT	Description
FH	FFH	BREAK The communication cable (receiver cable) to the partner has a break. This error number is repeated until the fault is cleared.

Remedy

Re-establish the connection between the devices or switch on the partner. Check whether the current flows in TTY operation.

8.2 Error Messages for SYSTAT, ANZW and REPTEL Arranged According to the Numbers for SYSTAT

No. for SYSTAT	No. for ANZW	No. for REPTEL	Description
			Parameter assignment errors recognized by the CPU of the PC and signalled to the CP:
01H	1H	0AH	a) Source/dest. type (e.g. ext. DB) illegal area (start address, length) illegal (negative value)
02H	2H	14H 0CH	b) DB/DX not present or illegal (e.g. DB0, DB1, DX0) or other data type not present or illegal
03H	3H	14H 0CH	c) DB/DX too short: (start address + length > area or area too short with other data type (start address + length > area
04H	4H	0AH	d) Access to area not possible for user: e.g., hardware memory too small or I/O module not plugged in
05H	5H	0CH	e) Wrong condition codeword: group message for all errors attributable to this ANZW
06H	6H	0AH	Error code from the CPU to the CP that the CP cannot interpret

Remedy

- Check parameter assignment on CPU and CP and if necessary correct or
- Partner has illegal parameters in the telegram header

Find the permissible data types in the job tables (Section 2.6).

- Check parameter assignment on CPU and CP; if necessary set up block
- Partner has wrong parameters in the telegram header

Find the permissible data types in the job tables (Section 2.6).

- Check parameter assignment on CPU and CP if necessary check block/area or
- Partner has wrong parameters in the telegram header

- Check parameter assignment on CPU and CP or
- Partner has wrong parameters in the telegram header

Find the permissible start addresses and lengths in the job tables (Section 2.6).

Check whether condition codeword is valid.

Evaluate PAFE in handling block (STEP 5 user program).

No. for SYSTAT	No. for ANZW	No. for REPTEL	Description
10H	6H	0AH	**Errors detected by CP during data exchange between CPU and CP:** a) Timeout for handshake on CP
11H	6H		b) ALL job without request or direct job despite overload
12H	6H		c) Wrong sequence for handshake by CPU
13H	6H		d) Unknown job type or illegal job number for direct job
14h	6H		e) Illegal acknowledgement from CPU during handshake
15H	7H		**System errors at start of job processing:** a) Number of direct jobs that can be processed simultaneously too large for CP 524/CP 525
16H	7H		b) Current job was terminated with error status when CP was cold restarted (power on) or cold restarted from PG
17H	7H		c) Area is masked by STEP 5 program (bit 7 in ANZW = 1)
1EH	7H		d) Field length for "Read SYSID" (RECEIVE DIRECT 223) too small
1FH	7H		e) "Read date/time" (RECEIVE DIRECT 218) illegal, CP is set as time slave

Remedy

Check whether RECEIVE ALL or SEND ALL are called in your STEP 5 program; evaluate the PAFE in the handling block

---

Check STEP 5 program, e.g., errors in handling block
(Message also possible after transfer from PG to CP).

Check STEP 5 program, e.g., errors in handling block.

Check STEP 5 program, e.g., incorrect parameters for handling block.

Check STEP 5 program.

Modify STEP 5 program so that maximum 10 direct jobs run simultaneously.

There is no remedy with power on. When cold restarting the CP from the PG you make sure that no more CPU jobs are running before writing to an interface. The SYNCHRON HDB clears the job statuses in the dual-port RAM, but not the SYSTAT entry.

Reset bit 7 in the ANZW.

For SYNCHRON the block size should be greater than or equal to 128 bytes or do without "Read SYSID" (data are nevertheless available in the dual-port RAM).

Before reading the date/time, the CP must be designated as master - this then applies to this module; check whether the CP on the other interface has been set to slave.

No. for SYSTAT	No. for ANZW	No.from REPTEL	Description
			Parameter ass. errors in the STEP 5 prog. detected by the CP (e.g. in the pseudo R/W parameters) or job block (CP 525 user program)
20H	7H		a) No parameters were entered in the CP 525 user program for this direct job (under this job number)
21H	7H		b) Error in data type (DB, OB...): - unknown data type or - data type not allowed (e.g. ext. DB) or - data type not allowed in HDB parameters in connection with data type in the job block (CP 525 user program)
22H	7H		c) Start address specified too high for required data type or start address or DB/DX no. too low
23H	7H		d) Coordination flags not allowed with this data type or wrong bit number specified
24H	7H		e) CPU number specified too high with pseudo R/W parameters
25H	7H		f) Transmission length > 64 Kbytes too long for CP 525
26H	7H		g) Length of telegram header too great or length too great for type specified (in own PC)

Remedy

Create parameters for this job with COM 525 or select other job number.

Find the permissible data types and their combinations in the job tables (Section 2.6).

Find the permissible start addresses and DB/DX nos. in the job tables (Section 2.6), which can be specified in the CP 525 user program or STEP 5 program (with pseudo R/W parameters).

Coordination flags are not permissible for all data types (see Section 2.6). Check whether the bit number (0 to 7) has been specified correctly when the pseudo R/W function is used.

CPU nos. that are permitted: none, 1, 2, 3 or 4.

If necessary, split the job into several jobs with smaller lengths.

Find the permissible lengths in the job tables (Section 2.6) to be specified in the HDB or for the pseudo R/W parameters. If necessary split the job into several jobs with smaller lengths.

No. for SYSTAT	No. for ANZW	No. in REPTEL	Description
27H	CH		Selector on STOP/PGR during PC job
28H			Error when sending a command telegram. A corresponding procedure error number was entered in SYSTAT immediately before (see error numbers E0 to E5 and F8 to FF).
29H	CH		**Errors in the reply telegram from the partner, detected by the CP:** a) Synchron error by partner, since - reply telegram arrived, though no job running or - the (follow-on) FETCH telegram has not yet been sent or - a follow-on reply telegram was received after a FETCH telegram was sent or - a reply telegram was received after a follow-on FETCH telegram was sent
2AH	BH		b) Error in format of reply telegram received: first byte not 00H or FFH
2BH	BH		c) Reply telegram received after FETCH job has too many data
2CH	BH		d) Reply telegram received after FETCH job has too few data
2DH	BH		e) Reply telegram received after SEND job arrived with data

Remedy

Switch the selector to RUN.

See remedy for previous procedure error number.

This error can be signalled following a cold restart at your PC, with long telegrams, or at the partner. It is a normal reaction during start-up; you do not need to do anything. During normal operation the error can also occur as a result of older error statuses that were detected only by the partner. Otherwise you can assume that the partner is reacting incorrectly. It is possible that the error will not occur with jobs < 128 bytes.

7

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for SYSTAT	No. for ANZW	No.from REPTEL	Description
2EH			When processing a PC job: Errors in transfer/reception of data via the dual-port-RAM —> handshaking error with the CPU. A more detailed error number was entered immediately before in systat (error numbers 1 to 5 and 10, 14, 16, 17 - see previous pages)
2FH	DH		No reply telegram came from the partner within the monitoring time after sending a FETCH telegram.
			If the partner detects an error when processing a job from another device, it sends a reply telegram back to the CP 525 with the error number in the 4th byte. The CP generates a no. for SYSTAT and for ANZW from this:
30H	9H	32H	a) DB / DX disabled by coordination flags
31H	AH	0AH	b) Hardware (DMA) error at the partner or with CP 525 as partner: - source/dest. type illegal - or memory error in partner PC - or error in handshake between CP/CPU at partner or - partner PC is in STOP state

Remedy

See the remedy for the preceding error number.

Is the partner a very slow device?
This error is often indicated as the result of an older error. For example, procedure receive errors (FOH to FFH) can be indicated after a FETCH telegram has been sent because the reply telegram could not be received owing to a disturbance —> there is a timeout. This error can sometimes occur when a cold restart has been carried out at the partner before it could reply to the last FETCH telegram received.

7

In the partner program: after processing the last transmission data reset the coordination flags.
In the STEP 5 program: repeat the job.

- Check whether partner can transfer the required data type
- Check partner's hardware configuration
- Switch mode selector at partner PC to RUN

No. for SYSTAT	No. for ANZW	No.from REPTEL	Description
32H	AH	0CH	c) Memory access error at partner (memory not present) with CP 525 as partner: - wrong area for condition code-word or - data area does not exist (except DB/DX) or - data area too small (except DB/DX)
33H	AH	14H	d) DB/DX access error at partner (DB/DX does not exist or too short) with CP 525 as partner: - DB/DX does not exist or - DB/DX too short or - DB/DX no. illegal
34H	BH	16H	e) Errors in the telegram header, detected by partner: first command letter in header wrong
35H	BH	10H	f) Error in telegram header: second command letter in header wrong
36H	BH	34H	g) Partner detects wrong telegram length (total length)
37H	CH	36H	h) Partner detects synchron error - order of telegrams is wrong

Remedy

Check whether the required data area exists on the partner and whether it is long enough; or check the job block (CP 525 user program); or the pseudo R/W parameters (STEP 5 user program). Check the length specified in the HDB.

Check whether the required data block exists on the partner and whether it is long enough; or check the job block (CP 525 user program); or the pseudo R/W parameters (STEP 5 user program). Check the length specified in the HDB.

Check whether the error is being caused by disturbances or incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

Check whether the error is being caused by disturbances or incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

Check whether the error is being caused by disturbances or incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

This error can be signalled during a cold restart at your PC or at the partner. It is a normal reaction during start-up; you do not need to do anything. During normal operation the error can also occur as a result of older error statuses that were detected only by the partner. Otherwise you can assume that the partner is reacting incorrectly.

No. for SYSTAT	No. for ANZW	No.from REPTEL	Description
38H	CH	2AH	i) Up to now there has been no cold restart at the partner. with CP 525 as partner: since the power up no "SYNCHRON" HDB has run or the mode selector is switched to STOP/PGR.
39H	BH	12H	j) Partner signals "system command illegal"
3AH	BH	??H	k) Unknown error number received in reply telegram

Remedy

Carry out a cold restart at the partner PC or switch the selector on the CP to RUN.

This is an incorrect reaction by partner, the CP 525 never outputs a system command.

Check whether the error is being caused by disturbances or incorrect reaction of the partner. Verify with an interface test device ('FOXPG') tapped into the transmission line.

No. for SYSTAT	No. for ANZW	No. for REPTEL	Description
			Error in the command telegram of the partner. A reply telegram is sent to the partner with the specified error no. in the 4th byte..
40H		10H	a) Error in 1st command byte not 00H or FFH
41H		16H	b) Error in 3rd command byte: command letter is not "A" or "O" or "E" or "X"
42H		16H	c) Error in 3rd command byte in follow-on telegram: command letter not as in 1st command telegram
43H		10H	d) Error in 4th command byte: command letter wrong
44H		10H	e) Error in 4th command byte in follow-on telegram: command letter not as in first command telegram

Remedy

For more information on command telegrams, see Section 4 in this user's guide.

Basic format error in partner's header. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Basic format error in partner's header. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Basic format error in partner's header. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Basic format error in partner's header or a command combination is requested, which is not allowed with the CP525. Check the permissible commands in the appropriate tables (Section 2.6) Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Basic format error in partner's header. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for SYSTAT	No. for ANZW	No. for REPTEL	Description
45H		14H	f) Error in 5th command byte: DB/DX no. illegal (e.g. 00)
46H		0CH	g) Error in 5th command byte: start address too high
47H		34H	h) Error in 7th + 8th comm. bytes: specified length too high
48H		0CH	i) Error in 9th + 10th comm. bytes: coordination flag cannot be specified with this data type or the bit number of the coor-ination flag is too high (perm. value range 0 to 7)
49H		0CH	j) Error in 10th command byte: CPU no. too high (permitted values 0,1,2,3,4,F)
4AH		12H	k) System command "XF" not allowed
4BH		34H	l) Send telegram was longer than expected, i.e., more data was received (poss. with follow-on telegrams) than indicated in the telegram header
4CH		34H	m) Send telegram was too short or < 128 bytes; i.e. less data received (poss. with follow-on telegrams) than indicated in the telegram header
4DH		34H	n) Fetch telegram received with information data.

Remedy

Find the permissible DB/DX numbers in the job tables (Section 2.6).

Find the permissible start addresses in the job tables (Section 2.6).

Find the permissible lengths in the job tables (Section 2.6).

Basic error in partner's header structure. Find the in the job tables (Section 2.6) when coordination flags are allowed.

Basic error in partner's header structure

System command only meaningful for AS 512C.

Must be corrected at partner

Must be corrected at partner

Must be corrected at partner

No. for SYSTAT	No. for ANZW	No. for REPTEL	Description
			Other errors when processing the partner job
4EH		2AH	a) CP has received command telegram, while mode selector was set to STOP/PGR
4FH*)		2AH	b) Command telegram was received, although the CPU of the PC has not run a "SYNCHRON" HDB
50H		-	c) Error sending the (follow-on) reply telegram, a corresponding procedure error number was entered in SYSTAT immediately before (error numbers E0 to E5, F8 to FF)
51H		36H	d) - synchron error by partner, new (follow-on) command telegram arrived, although previous (follow-on) reply telegram had not been sent by CP 525 or - a normal command telegram was received, although a follow-on command telegram was expected or - a follow-on command telegram arrived, although a command telegram was expected

*) The CP has not yet been synchronized. The SYSTAT cannot be read by the handling blocks. These error messages must be read direct from the dual-port RAM. (Structure from address F660H as for "read SYSTAT").

Remedy

Switch mode selector on the CP to RUN

In the STEP 5 program, "SYNCHRON" calls must be programmed in the start-up organization blocks (OB20, OB21, OB22). Following this a cold restart must be carried out (the PAFE byte in the SYNCHRON HDB may have to be evaluated).

See remedy for error number entered immediately before in SYSTAT (SYSTAT no. E0 to E5 and F8 to FF).

This error can be signalled following a cold restart at your PC if long telegrams are to be sent or during a cold restart at the partner. This is a normal reaction during start-up; you do not need to do anything. During normal operation the error can also occur as a result of older error statuses which were only detected by the partner. Otherwise you can assume that the partner is reacting incorrectly. It is possible that the error will not occur with jobs less than 128 bytes.

No. for SYSTAT	No. for ANZW	No. for REPTEL	Description
52H		32H	DB/DX disabled by coordination flags
53H			When processing a partner job: errors during transfer/reception of data via the dual-port RAM; handshake error with the CPU of the PC; a more detailed error number was entered in SYSTAT immediately before (Error numbers 1 to 5 and 10, 14, 16 - number for REPTEL - see previous pages).
5EH			A correct telegram with a length less than the header length in the (follow-on) command telegram was received (< length 4 with follow-on telegrams or reply telegrams; < length 10 with command telegrams). The telegram is ignored. Jobs currently running are not aborted.
5FH		34H	A correct telegram with a length > 128 bytes was received. The telegram could not be evaluated. The job being processed is aborted.

Remedy

In your own STEP 5 program: reset the coordination flags again after processing the last transmitted data.

In the partner program: repeat the job

When starting up for the first time, make sure that the coordination flags are distributed on the individual processors, CPs, IPs and the coordinator, the jumper settings are correct and the coordination flags used in the start-up OBs (OB20, OB21, OB22) are reset (see also instructions CP 524/CP 525).

See remedy for the error number entered immediately before in SYSTAT.

Verify the incorrect reaction of the partner, if necessary, with an interface test device ('FOXPG') tapped into the transmission line.

Verify the incorrect reaction of the partner, if necessary. with an interface test device ('FOXPG') tapped into the transmission line.

No. for SYSTAT	No. on PG	Description
		Error messages when sending a telegram from the CP to the PG:
B0H	-	a) Link establishment with STX was either not acknowledged by the PG within 300 ms acknowledgement monitoring time) or was not acknowledged with DLE ACK or DLE NAK
B1H	Err. 722 transfer error	b) Transmission (CP -> PG) was aborted by PG with NAK
B2H		c) The link termination with DLE ETX was acknowledged by the PG with DLE NAK
B3H		d) The link termination with DLE ETX was not acknowledged by the PG with DLE ACK or DLE NAK within 300 ms (acknowledgement monitoring time).
		Error messages when receiving a telegram from the PG:
B4H	Err. 722 transfer error	a) Transfer error when receiving (CP aborts with DLE NAK).

For further information see the description "COM 525 messages" in this manual.

Remedy

- Check cable
- Load COM 525
- Hardware error at PG or CP

Repeat transfer (otherwise: see BOH)

Repeat transfer (otherwise see BOH)

- Check cable
- Load COM 525
- Hardware error at PG or CP

Repeat transfer (otherwise: see BOH)

No. for SYSTAT	No. on PG	Description
		Errors carrying out a job from the PG:
B5H	IF.007 CP funct. unknown	a) A function is to be carried out which is unknown to the CP 525 or there is a transfer error in the exchange of telegrams
B6H	ERR. 709 CP memory full	b) The user memory of the CP 525 is full
B7H	ERR. 027 EPROM plugged in	c) The write access by the PG cannot be carried out, because there is no RAM submodule plugged in as the user memory of the CP 525 (either EPROM or no submodule plugged in)
B8H	ERR. 721 CP switch set to "STOP"	d) The mode selector on the CP is set to "STOP" or "PGR" (during cold restart from the PG)

Remedy

- COM 525 loaded?
- Repeat transfer

- Plug in submodule with greater capacity
- Erase unnecessary module files on the CP

Plug in RAM

Switch mode selector on the CP to "RUN"

No. for SYSTAT	No. for ANZW	Description
		Hardware errors on the module:
C0H*)	-	a) Error testing the module's operating system EPROM
C1H	-	b) RAM on CP defect
C2H	-	c) Dual-port RAM on module defect
C3H	-	d) RAM on memory submodule card defect
C4H*)	-	e) Memory submodule not permitted
C5H	-	f) Error in memory submodule while program is running
C6H*)	-	g) EPROM memory submodule empty or contents unknown
C7H*)	-	h) Applies only to CP 524: interpreter not permitted for this CP

*): These errors are signalled only on interface 1. The CP was not synchronized, the data cannot be read by the handling block. These error messages must then be read directly from the dual-port RAM (structure from address F660H onwards as for "read SYSTAT").

Remedy

Wrong or defective operating system EPROM
Replace EPROM.

Replace CP.

Replace CP.

Replace RAM submodule.

- Use permitted submodule type
- Submodule or submodule controller defect

Contact customer service.

- Check whether this is the correct EPROM module
- Check whether module is programmed

With submodules which are programmed for the CP 525-2 and used in the CP 524, this error number appears if interface 1 is not programmed.

Only use permitted interpreter.

No. for SYSTAT	No. for ANZW	Description
C8H*)	-	i) Only applies to CP 524: interface module (identifier) unknown or module not plugged in
C9H	-	j) Interface not loaded
CAH*)	-	k) Memory submodule not plugged into CP

*): These errors are only signalled on interface 1. The CP was not synchronized, the data cannot be read by the handling block. These error messages must then be read directly from the dual-port RAM (structure from address F660H onwards as for "read SYSTAT").

Remedy

- Check whether correct module, if "yes": module defect
- Check whether module plugged in correctly

Load interface if necessary
Caution: the entry in SYSTAT is not cleared by loading the interface, but only by RESET DIRECT 200

Check whether submodule is plugged in; if "yes": submodule defect (or submodule controller defect) - replace submodule

No. for SYSTAT	No. for ANZW	Description
D0H	-	Incompatible procedure - interpreter: procedure and interpreter on the CP 525 cannot work together.
E0H		**Errors in data exchange CP <—> partner:** **1. Errors detected by the procedure during the sending of a telegram and signalled to the interpreter:** a) The first repetition has been sent: - when sending a telegram an error was recognised or - the partner requests a repetition by means of a negative acknowledgement character (NAK)
E1H	EH	b) Errors during the link establishment - after STX was sent, NAK or any other character (except DLE or STX) was received or - the reply came too soon or - an initialization conflict has occurred
E2H	EH	c) Timeout (QVZ): after sending STX the partner did not reply within the acknowledgement time

Remedy

Check procedure/interpreter on the CP 525 and if necessary replace.

A repetition is not an error. It can, however, point to disturbances on the transmission line or to an incorrect reaction by the partner. If, after the maximum number of repetitions, the telegram could not be transferred, an error number is signalled that describes the error which occurred first.

Note:
when coupled to an AS 512C module the AS 512C module sometimes requests a repetition owing to overload at high data rates.

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Partner is too slow or not ready to receive or there is a line break. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for SYSTAT	No. for ANZW	Description
E3H	EH	d) Aborted by partner: during transmission one or more characters were received from the partner (incl. NAK)
E4H	EH	e) Errors at end of link: - the telegram was rejected at the end by the partner with NAK or some other character (except DLE) or - the acknowledgement character was received too soon
E5H	EH	f) Timeout at end of link: after link termination with DLE ETX no reply was received from the partner within the QVZ time
		2. Errors occurring when a telegram is received, detected by the procedure and signalled to the interpreter:
F0H	-	a) Expecting the first repetition: when receiving a telegram an error was detected and the CP 525 requested a repetition by sending a negative acknowledgement (NAK).

Remedy

- Check whether the partner is also indicating an error; it is possible that not all the data sent has arrived (e.g. break in the line) or
- partner is reacting incorrectly. Verify this, if necessary, with an interface test device ('FOXPG') tapped into the transmission line.

- Check whether the partner is also indicating an error; it is possible that serious disturbances have occurred or
- partner is reacting incorrectly. Verify this, if necessary with an interface test device ('FOXPG') tapped into the transmission line.

- Partner is too slow or has a fault. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

A repetition is not an error. It can, however, point to disturbances on the transmission line or to an incorrect reaction by the partner. If, after the maximum number of repetitions, the telegram could not be transferred, an error number is signalled which describes the error which occurred first.

No. for SYSTAT	No. for ANZW	Description
F1H		b) Errors during link termination: - in the idle state one or more characters (except NAK or STX) were received or - following reception of STX the partner sent further characters without waiting for the reply DLE
F2H		c) Logical error while receiving: after receiving DLE a further character was received (except DLE, ETB, ETX)
F3H		d) Character timeout (ZVZ): - two successive characters were not received within the ZVZ or - during link establishment the 1st character after DLE was not received within the ZVZ
F4H		e) Block check character (BCC) error: the value of the BCC calculated internally does not agree with the one received from the partner at the end of the link (only with 3964R procedure)
F5H		f) Block monitoring time has expired. After a command telegram was sent no reply telegram was received from the partner within the monitoring time.

Remedy

Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Check whether the partner always doubles DLE in the telegram header and in the data string or terminates the link with DLE ETX. Verify incorrect reaction of partner, if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Partner is too slow or has a fault. Verify this if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Check whether the connection is badly affected by interference; if so, error codes FE are sometimes obtained. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

Assign the same repetition amount of parameters to the partner as to the CP. Verify incorrect reaction of partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for SYSTAT	No. for ANZW	Description
F6H		f) There is no free receive buffer available: after receiving STX there was no empty receive buffer available for the procedure during link establishment and during a further waiting period
F8H		g) Wrong end identifier (ETB) was received: a received telegram was completed with the end identifier ETB. The telegram is ignored, the job being processed is not aborted
FEH	EH	Transmission error: a transmission error (parity error, stop bit error, overflow error) was detected. If this occurred during transmission or reception repetitions will be started. If a corrupted character is received in the idle state, the error is signalled immediately to ensure that disturbances on the line are detected as early as possible.
FFH	FH	BREAK The communication cable (receiver cable) to the partner has a break. This error number will be repeated until the fault is cleared.

Remedy

This error can only occur if the transmission is extremely badly disturbed because of short term overload of the RK 512 interpreter. Correct the problem by changing the system configuration or re-laying the cables.

Modify the partner since RK 512 does not recognise the data control character ETB.

Disturbances on the line result in repetitions of the telegram and therefore reduce the actual data throughput. The chance of an undetected error increases. Change the configuration or re-lay the cable.
Check whether the data rates, parity and number of stop bits are the same at both devices.

Re-establish the connection between the devices or switch on the partner. Check whether the current flows in TTY operation.

8.3 Error Messages for SYSTAT and REPTEL Arranged According to Numbers for REPTEL

No. for REPTEL	No. for SYSTAT	Description
		Parameter assignment errors detected by the CPU of the PC and signalled to the CP:
0AH	01H	a) Source/dest. type (e.g. ext. DB) not permissible area (start address, length) not permissible (negative value)
	04H	b) Access to area not possible for user: e.g. hardware memory to small or I/O module not plugged in
	06H	Error indication from the CPU to the CP, which the CP cannot interpret
		Errors detected by the CP during data exchange between CPU and CP:
	10H	a) Monitoring time for handshake elapsed on CP

Remedy

- Check parameter assignment on CPU and CP and if nec. correct or
- Partner has illegal parameters in the telegram header

Find the permissible data types in the job tables (Section 2.6).

- Check parameter assignment on CPU and CP or
- Partner has wrong parameters in the telegram header

Find the permissible start addresses and lengths in the job tables (Section 2.6).

Evaluate PAFE in handling block (STEP 5 user program).

Check whether RECEIVE ALL or SEND ALL is called in your STEP 5 program; evaluate PAFE in the handling block.

No. for REPTEL	No. for SYSTAT	Description
		Parameter assignment errors detected by the CPU of the CP and signalled to the CP:
0CH	02H	a) Data type does not exist or not permissible
	03H	b) Area too short (start address + length) > area

Remedy

- Check parameter assignment of CPU and CP, if nec. set up block
- Partner has wrong parameters in telegram header

Find the permissible data types in the job tables (Section 2.6).

- Check parameter assignment on CPU and CP; if necessary check block/area or
- Partner has incorrect parameters in telegram header

No. for REPTEL	No. for SYSTAT	Description
		Errors in the command telegram of the partner; reply telegram is sent to the partner with the specified error number in the 4th byte
OCH	46H	a) Error in 5th command byte: start address too high
	48H	b) Error in 9th + 10th command byte: specifying a coordination flag is not permissible with this data type or the bit number of the coordination flag is too high (permitted value range 0 to 7)
	49H	c) Error in 10th command byte: CPU no. too large (permitted value range 0,1,2,3,4,F)

Remedy

Find the permissible start addresses in the job tables (Section 2.6).

Basic format error in the partner header. Find in the job tables (Section 2.6) when a coordination flag is permissible.

Basic format error in the partner header.

No. for REPTEL	No. for SYSTAT	Description
10H	40H	d) Error in 1st command byte: is not 00H or FFH
	43H	e) Error in 4th command byte: command letter wrong
	44H	f) Error in 4th command byte in follow-on telegram: command letter not as in the 1st command telegram

Remedy
Basic format error in the partner header. Verify the incorrect reaction of the partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.
Basic format error in the partner header or a combination of commands was requested which is not allowed with the CP 525. Find the permissible commands in the command tables (Section 2.6). Verify the incorrect reaction of the partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.
Basic format error in the partner header. Verify the incorrect reaction of the partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for REPTEL	No. for SYSTAT	Description
12H	4AH	g) System command "XM" not permissible
		Parameter assignment errors detected by the CPU of the PC and and signalled to the CP:
14H	02H	a) DB/DX not present or illegal (e.g. DB0, DB1, DX0)
	03H	b) DB/DX too short: (start address + length > area
		Errors in the command telegram from the partner. A reply telegram is sent to the partner with the error no. in the 4th byte:
	45H	a) Error in 5th command byte: DB/DX no. illegal (e.g. 00)
16H	41H	b) Error in 3rd command byte: command letter is not "A" or "E" or "X" or "O"
	42H	c) Error in 3rd command byte of a follow-on telegram: 1st command letter not as in the 1st command telegram

Remedy
System command is only meaningful for AS 512C
- Check parameter assignment on CPU and CP, if necessary set up block or - Partner has wrong parameters in the telegram header Find the permissible data types in the job tables (Section 2.6).
- Check parameter assignment on CPU and CP if necessary check block/area or - Partner has wrong parameters in the telegram header
Find the permissible DB/DX numbers in the job tables (Section 2.6)
Basic format error in the partner header. Verify the incorrect reaction of the partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.
Basic format error in the partner header. Verify the incorrect reaction of the partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

No. for REPTEL	No. for SYSTAT	Description
		Other errors during the processing of the partner job
2AH	4EH	a) CP has received command telegram, while mode selector was set to STOP/PGR
	4FH	b) Command telegram was received, however, the CPU of the PC has not yet run the "SYNCHRON" HDB
32H	52H	DB/DX disabled by coordination flags
		Errors in command telegram of the partner. A reply telegram is sent to the partner with the specified error in the 4th byte.
34H	47H	a) Error in 7th + 8th command byte: length specified too great
	4BH	b) Send telegram was longer than expected, i.e. more data were received (possible with follow-on telegrams) than were indicated in the telegram header.

Remedy

Switch the mode selector to RUN.

In the STEP 5 program, "SYNCHRON" calls must be programmed in the start-up organization blocks (OB20, OB21, OB22). Following this a cold restart must be carried out (the PAFE byte in the SYNCHRON HDB may have to be evaluated).

In your own STEP 5 program: reset the coordination flags again after processing the last tranmission data

In the partner program: repeat the job

When starting up for the first time, make sure that the coordination flags are distributed on the individual processors, CPs, IPs and the coordinator; that the jumper settings are correct; and that the coordination flags used in the start-up OBs (OB20, OB21, OB22) are reset (see also instructions CP 524/CP 525).

Find the permissible lengths in the job tables (Section 2.6).

Correction must be carried out on the partner.

No. for REPTEL	No. for SYSTAT	Description
34H	4CH	c) Send telegram was too short or < 128 bytes , i.e. less data were received (possible with follow-on telegrams) than were indicated in the telegram header
	4DH	d) Fetch telegram received with information data
	5FH	A correct telegram with a length > 128 bytes was received. The telegram cannot be evaluated. The job will be aborted.
36H	51H	**Other errors during processing of the partner job:** a) - synchron error on the partner, since a new (follow-on) command telegram arrived, although the previous (follow-on) reply telegram had not been sent by the CP 525 or - a normal command telegram was received although a follow-on command telegram was expected or - a follow-on command telegram arrived, although a command telegram was expected

Remedy

Carry out correction on the partner.

Carry out correction on the partner.

Verify the incorrect reaction of the partner if necessary with an interface test device ('FOXPG') tapped into the transmission line.

This error can be signalled following a cold restart at your PC if long telegrams are to be sent or during a cold restart at the partner. This is a normal reaction during start-up; you do not need to do anything.
During normal operation the error can also occur as a result of older error statuses were detected only by the partner. Otherwise you can assume that the partner is reacting incorrectly. It is possible that the error will not occur with jobs less than 128 bytes.

9 Time Measurement - Processing a DIRECT Job

The results of these time measurements for a SEND DIRECT job are examples of the processing times for DIRECT jobs.

Two S5 135Us, each with a CP 922 and a CP 525 were used for the time measurement.

The interpreter RK 512 and procedure 3964 (low priority) are loaded in the CP 525 of PC 1 on IF 1.

The interface IF 2 is not programmed.
A SEND job is programmed in the job block.

The interpreter RK 512 and procedure 3964 (high priority) are loaded in the passive CP 525 of PC 2 on IF 2.

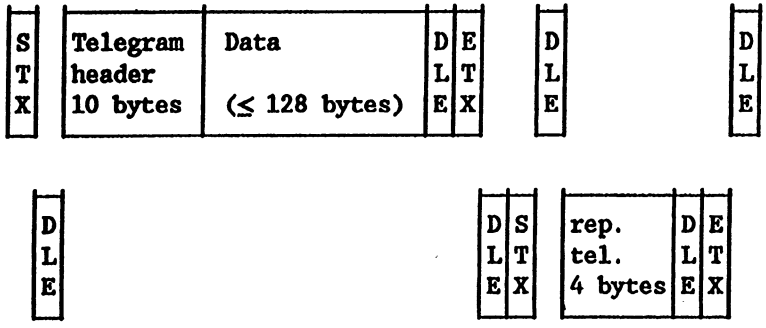
The interface IF 2 is not programmed.
No job is programmed in the job block.

The following times are measured:

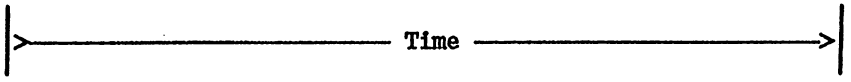
- from initiation of the SEND DIRECT job in PC 1
  (the bit "job running" is set)
- to the end of the job in PC 1
  (the bit "job finished without error" is set.

Telegrams are exchanged in a SEND DIRECT job with 128 or less bytes of information as follows:

active CP 525 in PLC 1



Passive CP 525 in PLC 2



If there is more than 128 bytes of data, then follow-on SEND telegrams are started. The header of a follow-on SEND telegram is 4 bytes long.

For more information on telegrams refer to the section "How is data transferred?".

Table: time measurement - processing a SEND DIRECT job

The time measurements are carried out with different baud rates and with different amounts of useful data.

bps / data	19200	9600	4800	2400	1200	600	300	150	110
1 DW	42ms	57ms	87ms	148ms	258ms	490ms	960ms	1.92s	2.63s
2DW	44ms	60ms	91ms	158ms	277ms	527ms	1.04s	2.08s	2.84s
4DW	46ms	64ms	100ms	175ms	313ms	600ms	1.19s	2.37s	3.23s
5DW	48ms	67ms	106ms	183ms	323ms	637ms	1.26s	2.52s	3.44s
8DW	50ms	73ms	118ms	212ms	387ms	750ms	1.48s	2.96s	4.04s
10DW	58ms	83ms	133ms	236ms	429ms	830ms	1.63s	3.25s	4.43s
16DW	66ms	97ms	163ms	295ms	548ms	1.07s	2.11s	4.20s	5.74s
20DW	75ms	109ms	184ms	332ms	617ms	1.20s	2.36s	4.72s	6.44s
32DW	93ms	143ms	244ms	447ms	840ms	1.64s	3.24s	6.48s	8.83s
50DW	132ms	200ms	343ms	628ms	1.18s	2.32s	4.59s	9.12s	12.4s
64DW	152ms	238ms	412ms	760ms	1.45s	2.83s	5.61s	11.2s	15.2s
80DW	196ms	310ms	541ms	1.01s	1.93s	3.76s	7.46s	14.9s	20.3s
100DW	220ms	356ms	634ms	1.19s	2.27s	4.49s	8.92s	17.8s	24.3s
128DW	252ms	420ms	760ms	1.45s	2.80s	5.53s	11.0s	21.9s	29.9s
192DW	353ms	601ms	1.11s	2.13s	4.15s	8.22s	16.4s	32.7s	44.5s

The time required to process a **FETCH DIRECT job** is approximately the same as that required for the SEND DIRECT job.

The times for the **3964R procedure** are approximately as those for processing a DIRECT job.

We have checked the contents of this manual for agreement with the hardware and software 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 welcomed.
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 model or design, are reserved.

Copyright © Siemens AG 1989 All Rights Reserved

Siemens Aktiengesellschaft

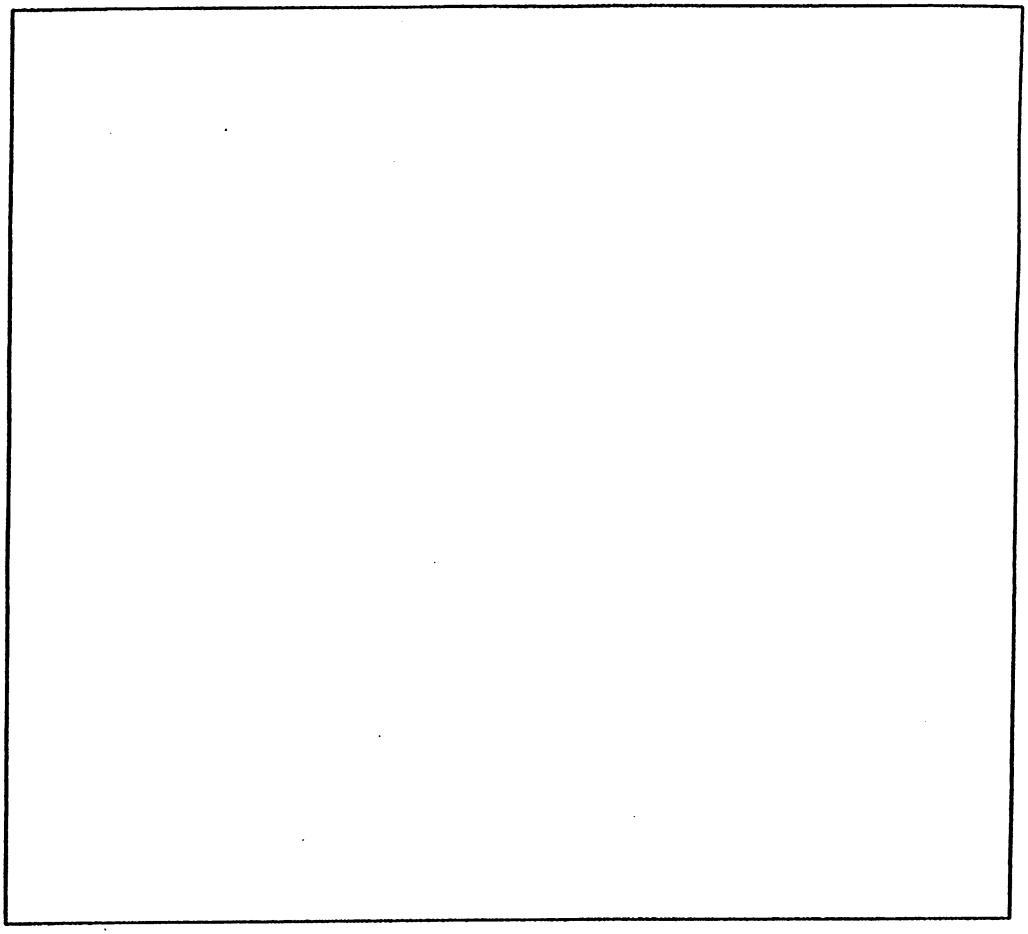
C79000-B8576-C539
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5
CP 525: Event Output and Listing with the PT88/PT89 Printer

User's Guide

Order No. C79000-B8576-C541-05



Contents

		Page
1	Overview	1
2	Sequential Message List (SML)	5
2.1	Processing the Messages on the CPU	6
2.2	Interface Between CPU and CP 525	9
2.2.1	Structure of the DB M RECORD	9
2.2.2	Example of the Structure of a DB M RECORD	19
2.3	Processing Messages on the CP 525	22
2.3.1	Buffering the Message Records	22
2.3.2	Preparing and Outputting the Messages	25
2.3.3	FIFO Overflow	26
2.3.4	Print Attributes	27
2.4	Frames with SMLs	28
2.5	Priority of the SML	29
2.6	Limit Values for the SML	31
3	Current Message List (CML)	32
3.1	Processing the Messages on the CPU	33
3.2	Interface Between the CPU and CP 525	33
3.3	Processing the Messages on the CP 525	36
3.4	Frames with CML	37
3.5	Priority of the CML	37
3.6	Limit Values for the CML	38
4	Process Status List (PSL)	39
4.1	Processing the PSL on the CPU	40
4.2	Interface Between the CPU and CP 525	40
4.3	Processing the PSL on the CP 525	41
4.4	Frames with PSLs	43
4.5	Priority of the PSL	43
4.6	Limit Values for the PSL	44

5	Chained Lists (CL)	45
5.1	Processing the CL on the CPU	46
5.2	Interface Between CPU and CP 525	47
5.3	Processing the CL on the CP 525	47
5.4	Frames with the Chained List	48
5.5	Limit Values for Chained Lists	49
6	Frames (List Headers/Trailers)	50
6.1	Processing the Data on the CPU	51
6.2	Interface Between CPU and CP 525	51
6.3	Processing the Data on the CP 525	51
6.4	Limit Values for Frames	53
7	Update Sequential Message List	54
7.1	Processing the Job on the CPU	55
7.2	Interface Between the CPU and CP 525	56
7.3	Processing the Job on the CP 525	56
8	Update Group Inhibit Bits	57
8.1	Processing the Group Inhibit Bits on the CPU	58
8.2	Interface Between CPU and CP 525	59
8.3	Processing the Group Inhibit Bits on the CP 525	59
9	New Page	60
10	Resetting the Page Number	60

11	Print Attributes	61
11.1	Print Types	62
11.1.1	Normal Print	62
11.1.2	Compressed Print (C)	63
11.2	Expanded Print	63
11.2.1	Normal Character Width	63
11.2.2	Double Character Width (E)	63
11.3	Bold Print (B)	64
11.3.1	Bold Print with the Impact Matrix Printer (Needle)	64
11.3.2	Bold Print with the Inkjet Printer	64
11.4	Underlining (U)	64
11.5	Acoustic Signal (A)	64
12	Setting up	66
12.1	Requirements	66
12.2	Settings on the CP 525	67
12.2.1	Jumper Settings on the CP 525	67
12.2.2	Assigning Parameters to the Procedure	67
12.2.3	Parameter Assignment for the Interpreter	68
12.3	Settings on the PT88/PT89 Printer	69
12.3.1	Coding Switch on the Central Controller	69
12.3.2	Interface Adapter SAP-S1 (V.24/V.28)	72
12.3.3	Interface Adapter SAP-S2 (TTY/20 mA)	74
12.3.4	Interface Adapter SAP-S3 (TTY/20 mA; V.24)	76
12.4	Start-up Phase of the CP 525	78
12.4.1	Starting Up after Switching On the Power	78
12.4.2	SYNCHRON Job from the CPU	78
12.4.3	Switch Settings RUN/STOP/PGR	80
12.4.4	Transferring from the PG	81
13	LAUFPT88 Procedure	83
13.1	Procedure Data	83
13.2	Transmitting with the LAUFPT88 Procedure	85
13.2.1	Data Transfer	85
13.2.2	Mode of Control of the LAUFPT88 Procedure	85
13.3	Receiving with the LAUFPT88 Procedure	87

14	Multiprocessor Operation	88
14.1	Jobs Programmed Only Once on the CP 525	89
14.1.1	Jobs without Data Transport	89
14.1.2	Jobs with Data Transport	90
14.2	Jobs which Can Be Programmed More Than Once on the CP 525, Including Frames	92
15	Memory Assignment, Memory Space Requirements	93
15.1	Loadable Software	96
15.1.1	Memory Requirements of the Loadable Software	96
15.2	User Programs for Specific Tasks	97
15.2.1	Directory (Memory Management)	97
15.2.1.1	Subdirectory of the Messages	98
15.2.1.2	Directory of PC Jobs and Frames	99
15.2.2	Element Types and Elements	103
15.2.2.1	Messages	106
15.2.2.2	Frames (Headers, Trailers)	108
15.2.2.3	PC Jobs	115
16	Special Jobs Carried Out by the CPU	124
16.1	Reading the Error Message Area of the SYSTAT	124
16.2	Resetting (Clearing) the Error Message Area	128
16.3	Reading the Whole SYSTAT	128
16.4	Reading the SYSID	130
16.5	CP 525: Reading and Writing the Date and Time	133
17	Error Handling	138
17.1	Error Messages for SYSTAT and ANZW - Arranged According to the Numbers for ANZW	144
17.2	Error Messages for SYSTAT and ANZW - Arranged According to the Numbers for SYSTAT	162
18	What Does It Mean When?	188

1 Overview

With the component "event output and listing with the PT88/PT89 printer" the CP 525 can **supervise** or **monitor** your **process/a part of the process/your plant**.

The following functions are available **for supervising your process:**

- **Sequential message list (SML)** (for detailed description see Section 2)

Each process event to be recorded is output on the printer in a message line with the following:

- static text (programmed by user) e.g. motor on
- time at which event occurred e.g. 11.11.86 13:59:00
- message status e.g. coming
- max. one parameter (process value) e.g. +135.2

The SML is therefore a continuous list of all the events (disturbances) to be recorded as they occur in the process over a longer period.

- **Current message list (CML)** (for detailed description see Section 3)

When the CML is called, the CP 525 has all the disturbances and events currently "active" on the CPU printed out. CML therefore records the overall status of events and disturbances in the process at a defined time (start time of the CML).

The dynamic data found in the SML is not included in the CML. The dynamic data includes the following:

- time
- status
- parameter (process value)

Question marks are printed out in their place.

For monitoring your process the following functions are available:

- **Process status list (PSL)** (for detailed description see Section 4)

When the PSL is called, the CP 525 fetches the current process data from the CPU and compiles a list by adding these data to static texts stored in the CP user memory. This list is then output on the printer.

PSL is therefore an image of the process/part of the process at a defined time (starting point of the PSL).

- **Chained lists (CL)** (for detailed description see Section 5)

When the CPU starts a chained list, several PSLs (maximum **eight**) are printed out automatically one after the other. In this way, i.e., by chaining the images of individual parts of the process, a process image of the whole system can be obtained.

You can have the lists supplemented by a

- **List frame** (for detailed description see Section 6)

This is part of a list that can be printed out at the beginning and/or at the end of each list.

The frame is not a list in its own right and cannot be started by the CPU. It can be assigned to the SML, CML and PSL when they are programmed.

- **Update sequential message list** (for detailed description see Section 7)

The dynamic values in the header and trailer of a current SML are updated at any time you select (e.g. change of shift).

- **Update group inhibit bits** (for detailed description see Section 8)

Individual messages in the SML or CML and any line of a PSL can be assigned to a **message group** (maximum **16** groups). These groups can be released or inhibited by the CPU with the job "update group inhibit bits". **None** of the messages or PSL lines belonging to an inhibited group are listed!

Your STEP 5 program on the CPU can send a data word at any time in which the CP 525 can determine the status of the message group (inhibited or released).

You can also bring "order" into your lists:

- **New page** (for detailed description see Section 9)

You can have a form feed carried out on the printer (printer header is then on the next page).

- **Reset page number** (for detailed description see Section 10)

When printing out on the PT88/PT89 you can have the page number reset to 1.

With the exception of the list frame, all the functions mentioned previously can be started by the CPU of the PC (i.e. by your STEP 5 program). This is achieved by calling the handling block **SEND DIRECT** with a job number.

2 Sequential Message List (SML)

Certain events within the process or plant (from now on referred to simply as **disturbances**) can be listed by the CP 525 quickly, in detail and at the exact time of the occurrence on the PT88/PT89 printer.

The sequential message list consists of a series of individual messages which are output continuously whenever a disturbance is detected (see Section 2.1). The SML can, if required, have a header and/or trailer added.

```
+-------------------------------------------------------+
! S e q u e n t i a l   m e s s a g e   l i s t   !
+-------------------------------------------------------+
!              Date:     12.08.86                       !
!              Time:     08:08:00 am                    !
+-------------------------------------------------------+
12.08.86 08:08:00 am  Temperature too high GOING
12.08.86 08:08:00 am  Motor is OFF         GOING
12.08.86 08:09:00 am  Load = 1024 kg       GOING
12.08.86 08:09:00 am  Belt no. +5 defect   GOING
12.08.86 08:10:00 am  !! Overflow !!       COMING
12.08.86 08:10:00 am  Process stable!      ACKNOWL
```

Fig. 1 Example: sequential message list

8

The disturbances that are listed can come from the following:

- the process (inputs, outputs, flags, data)
- the bus (e.g. SINEC H1) (telegrams)
- a station, via a telecontrol link (telegrams)
- etc.

Your STEP 5 program on the CPU of the CP must collect and prepare the data and pass them on to the CP 525. There is **an interface between the CPU and CP 525** for this purpose (see Section 2.2).

2.1 Processing the Messages on the CPU

Your STEP 5 program on the CPU must make the process data available to the CP 525. Before the CPU transfers the process data to the CP, the data is placed into **message records** on the CPU. The individual message records are entered in a data block known as **DB M RECORD** (for the structure of the DB M RECORD see Section 2.2).

At certain intervals or when the DB M RECORD is full of message records, the CPU sends a copy of the DB M RECORD to the CP 525 by calling the handling block SEND DIRECT.

!IMPORTANT! The structure of the DB M RECORD must be adhered to. The length of the "body" must be entered correctly!

The STEP 5 program for processing messages should be called up cyclically and must meet the following requirements:

1. The S5 program must detect disturbances.

To detect disturbances, the S5 program must do the following:
- define areas from which the messages originate (input, output, flag or data areas)
- assign the message number to the cause of a message. This is achieved using a data block (referred to from now on as DB M OLDNEW). The structure of DB M OLDNEW is described in Section 3.2
- compare the areas NEW and OLD cyclically in DB M OLDNEW (see 3.2). This comparison makes it possible to recognize whether new disturbances have "come" and older disturbances have "gone".

Area OLD		Area NEW	Message status
1	->	0	disturbance going
0	->	1	disturbance coming

2. The S5 program must generate message records when disturbances are detected.

To generate messages, the program must do the following:
- determine and enter the message number from the status changes in DB M OLDNEW (see 3.2)
- enter the message status (edge evaluation!)
- enter the data/time

8

- enter the parameter type and the parameter (process value). If no parameter is entered the parameter type must be specified as OFFH.

3. The S5 program must enter the message records in DB M RECORD.

To enter the records, the program must do the following:
- count the body length according to the length of the message record entered.

4. The S5 program must transfer DB M RECORD (or preferably a copy of the DB M RECORD called DB M SEND) to the CP 525.

To transfer the DB M RECORD, the program must do the following:
- guarantee that the DB M SEND is transferred to the CP 525 by means of a SEND DIRECT when the record is full and/or at certain intervals.

!IMPORTANT!: the job number of the SEND DIRECT must be the same as the job number of the SML programmed on the CP 525! The number of the DB M SEND must be specified when you are assigning parameters to the SEND DIRECT.

- purge the DB M RECORD, i.e., make space for new entries (body length = 0; pointer = 1st message record).

2.2 Interface Between CPU and CP 525

The interface between the CPU and the CP 525 is the data block DB M RECORD.

2.2.1 Structure of the DB M RECORD

The DB M RECORD is structured according to certain rules to ensure that the CP 525 can evaluate the DB M RECORD correctly. The DB M RECORD consists of a header and body as follows:

Byte	DW no.	Content
0 1	0	header
2 . . . 510 511	1 . . . 255	body

The **header** contains organizational information as follows:

Byte	DB addr.	Content	Comment
0	DL 0	33H	identifier for messages
1	DR 0	length (HEX)	length of the body in words

8

The **body** begins at byte 2 of the DB M RECORD. It contains a string of message records. The length of a message record, shown as follows, depends on whether it contains a parameter:

12 bytes = 6 words	-	without parameters
16 bytes = 8 words	-	with parameters

A DB M RECORD can be a maximum **256 words** long. It can therefore contain a maximum of the following:

42 message records	-	without parameters
31 message records	-	with parameters

Message record

The current information about any disturbance is entered in a message record in the following order:

1. parameter type
2. message status
3. message type
4. message number
5. time
6. date
7. parameter (optional)

Structure of a message record

Byte	S5 addr.	Content
0	DL 0	parameter type
1	DR 0	message status
2	DL 1	message type, message number
3	DR 1	message number
4	DL 2	10 ms – time
5	DR 2	s – time
6	DL 3	min – time
7	DR 3	h – time
8	DL 4	day – date
9	DR 4	month – date
10	DL 5	year – date
11	DR 5	reserved
12	DL 6	– parameter
13	DR 6	parameter
14	DL 7	parameter
15	DR 7	– parameter

8

1. Parameter type - byte 0

The parameter type determines whether a parameter is included in the message record and what type of parameter it is. The following SIMATIC data types are possible:

Type (hex)	Significance
0	binary
1	character
2	byte
3	16-bit BCD number
4	32-bit BCD number
5	16-bit fixed point number
6	32-bit fixed point number
7	32-bit floating point number
8	timer
9	counter
FF	no parameter included

The length of the message record is as follows:

12 bytes - without parameters
16 bytes - with parameters

2. Message status - byte 1

The status of the disturbance is described in more detail as follows:

coming
- the event just occurred (the disturbance has just been detected)
- the disturbance is still "active"
- the message has not yet been acknowledged by the user (e.g. via the CP 526)

going
- a previously signalled disturbance is no longer "active". The event has "gone".

acknowledged
- the user has noticed the signalled disturbance and has acknowledged it. The disturbance is, however, still "active".

The status of a disturbance must be represented in the message record shown as follows:

byte 1, bit 2^0 and 2^1:

Bit number 7	6	5	4	3	2	1	0	Significance
0	0	0	0	0	0	0	0	end of disturbance (going)
0	0	0	0	0	0	0	1	disturbance (coming)
0	0	0	0	0	0	1	0	acknowledged
0	0	0	0	0	0	1	1	not allowed

3. Message type - byte 2

The message type indicates whether it is a message caused by a disturbance.

byte 2, bit 2^4 to 2^7:

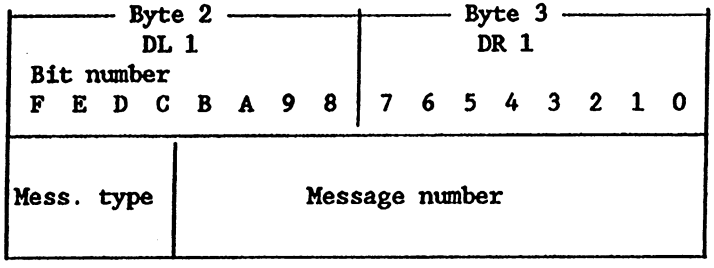
Bit number 7	6	5	4	3	2	1	0	Significance DEC	
0	0	0	0					0	disturbance message
0	0	0	1	see				1 –	
0	0	1	0	message				2	
.	.	.	.	number				.	reserved for extensions
.	.	.	.					.	
1	1	1	1					15 –	

4. **Message number** - byte 2 and byte 3

The messages are numbered to ensure that the following occurs:

- the correct values are assigned to each disturbance on each **CPU**
- the correct message text is assigned to each message on the **CP 525** (i.e. the text programmed for this message).
  If no message text is stored for a particular message number, nothing can be printed out on the printer.

The higher order part of the message number is in byte 2, bits 2^3 to 2^0 and the lower order in byte 3.



Limits

numbers that can be assigned:	1 to 2000
per CP 525 interface maximum possible:	1000 messages
per CPU of the PC maximum possible:	2000 messages

8

5./6. Date and time - byte 4 to byte 10

You can have the time at which the disturbance occurs or at which the status of the disturbance changes entered in the message record. You can, for example, read the date and time from the CP 525 or from the CP 526.

When you program the SML (with COM 525) you determine at which point and in which order the date and/or time is entered in the message text. This applies to **all** messages in the sequential message list (!), i.e. you cannot specify the date/time for each individual message. This process ensures a **uniform, clear layout** for all SMLs.

The date and time must be entered in BCD code as follows:

Byte	S5 addr.	Content		Example
				23.05.87 14.15 / 16s 700ms
4	DL 2	10 ms	–	70H
5	DR 2	s	|	16H
6	DL 3	min	| **time**	15H
7	DR 3	h	–	14H
8	DL 4	day	–	23H
9	DR 4	month	| **date**	05H
10	DL 5	year	–	87H

!IMPORTANT!: the CP 525 checks whether there is a zero in the field for the day (byte 8). If there is, the date and time are not entered in the message record. The CP 525 then enters the current value of the CP 525 clock in the message record.

Note that you must set the clock of the CP 525 using the CPU clock. You can set the date and time on the CPU using COM 526 for example.

7. Parameter - byte 12 to byte 15

You can reserve space for **one** current parameter in each disturbance message to obtain detailed information about the disturbance. The current parameter is acquired at the time when the disturbance occurs.

Example:

25.10.85 19:30 inlet pressure exceeded : **15** bar

---

15 = parameter

The parameter is optional.
The dynamic parts of this message text are printed in **bold face.**

.heB8576541-03 **Printer PT88/PT89**

Writing the parameters in the message record

Type	Meaning	DL 6 Byte 12 FEDC BA98	DR 6 Byte 13 7654 3210	DL 7 Byte 14 FEDC BA98	DR 7 Byte 15 7654 3210
0	Bit	yyyy yyyy	yyyy yyyy	aaaa	
1	Character	yyyy yyyy	yyyy yyyy	bbbb	
2	Byte	yyyy yyyy	yyyy yyyy	bbbb	
3	BCD 16	zzzz zzzz	zzzz zzzz		
4	BCD 32	zzzz zzzz	zzzz zzzz	zzzz zzzz	zzzz zzzz
5	FXP 16	zzzz zzzz	zzzz zzzz		
6	FXP 32	zzzz zzzz	zzzz zzzz	zzzz zzzz	zzzz zzzz
7	FLP 32	zzzz zzzz	zzzz zzzz	zzzz zzzz	zzzz zzzz
8	Timer	zzzz zzzz	zzzz zzzz		
9	Counter	zzzz zzzz	zzzz zzzz		

z = location of the parameter with formats that require at least one data word

y = location of the parameter with formats that do not require a whole data word. Nevertheless, a whole data word is stored. The selection of the bits or bytes is made via a or b.

a = bit address in data word

b = 0H: right-hand byte in DW 6

b = 8H: left-hand byte in DW 6

Example - parameters in the message record

Parameter type		Value	Byte 12	Byte 13	Byte 14	Byte 15
0	BIN	Bit 14=1	40	00	0E	xx
1	CHAR	'A'	41	00	08	xx
2	BYTE	10	00	0A	00	xx
3	BCD 16	100	01	00	xx	xx
4	BCD 32	987654	00	98	76	54
5	FXP 16	1024	04	00	xx	xx
6	FXP 32	1024	00	00	04	00
7	FLP 32	+1234567+02	04	62	C3	EF
8	Timer	123.3	31	23	xx	xx
9	Counter	123	01	23	xx	xx

xx = not used

2.2.2 Example of the Structure of a DB M RECORD

```
DB206                                          LEN=260     ABS
                                                          PAGE 1
        0 :      KH= 330E;
        1 :      KH= 0001;
        2 :      KH= 0065;
        3 :      KH= 6056;
        4 :      KH= 3715;
        5 :      KH= 2205;
        6 :      KH= 8700;
        7 :      KH= 0001;
        8 :      KH= 0100;
        9 :      KH= FF00;
       10 :      KH= 0064;
       11 :      KH= 6056;
       12 :      KH= 3715;
       13 :      KH= 2205;
       14 :      KH= 8700;
       15 :      KH= 0000;
       16 :240 <KH= 0000;>
```

This is the printout of DB 206, which is used in this example as DB M RECORD. Only data words DW 0 to DW 14 are of interest since only two messages have been entered.

Explanation of DB M RECORD

DW no	Significance
0	33 : identifier for DB M RECORD 0E =14D : body length in words, DW 1 to DW 14
1 to 8	message record of the first message
1	00 = parameter type = BINARY 01 = message status = COMING
2	0 = message type = disturbance message 065 = message number = message 101

DW no.	Significance
3	60 = time : 10 milliseconds 56 = time : seconds
4	37 = time : minutes 15 = time : hours
5	22 = date : day 05 = date : month
6	87 = date : year 00 = reserved
7	0001 = parameter (bit 0 = 1, bit 1 to bit 15 = 0)
8	01 = bit number of the parameter in DW 7 (bit 1) 00 = irrelevant for parameter type BINARY
9 to 14 message record of the second message	
9	FF = parameter type = no parameter 00 = message status = GOING
10	0 = message type = disturbance message 064 = message number = message 100
11	60 = time : 10 milliseconds 56 = time : seconds
12	37 = time : minutes 15 = time : hours
13	22 = date : day 05 = date : month
14	87 = date : year 00 = reserved
15 to xx irrelevant, since only two messages	

Two messages were entered:

```
05.22.87  15:37:56  COM 101  binary
05.22.87  15:37:56  GON 100  static
```

2.3 Processing Messages on the CP 525

The message records are buffered immediately when they arrive at the CP.

2.3.1 Buffering the Message Records

The PT88 and PT89 can output messages at the following speeds:

maximum 150 characters/s (inkjet)
maximum 80 characters/s (impact matrix)

This is **considerably slower** than the acquisition of process data on the CPU and the transfer of data from the CPU to the CP 525. These differences in speeds are compensated by having all the message records that are transferred from the CPU to the CP 525 buffered on the CP 525. The data is output to the printer only after it has been buffered.

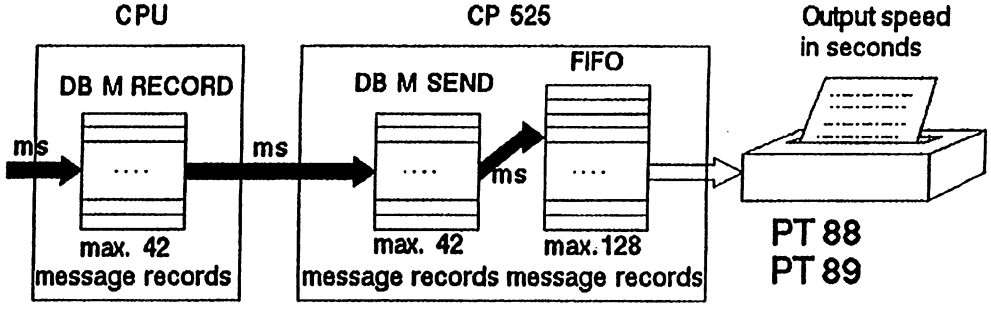


Fig. 2 Buffering data on the CP 525

The DB M SEND is transported to the CP 525 at **high speed** (ms = milliseconds). It is then buffered since the PT88 (PT89) can output a maximum 150 characters/s (80 characters/s).

The message buffer on the CP 525 operates as a **FIFO (first in first out)** memory. This means that all messages are processed in the order in which they are entered.

The CP 525 ensures that the following occurs:

- the messages are output quickly on the printer
- even if a large number of messages occur, none are lost

The CP 525 therefore only transfers the message records to the FIFO when the following conditions are met:

- there is still space in the FIFO
  (maximum 128 message records can be entered)
- a message text is stored on the CP 525 for the message number entered in the message record (you programmed these with COM 525 - see Fig. 3).

This process avoids the FIFO's being "cluttered up" by unlistable messages that might otherwise cause an overflow if many messages occur at one time.

From the message records provided for the CP 525 by the CPU, the CP 525 selects those that have message texts programmed for them. These are then buffered in the FIFO and printed out on the printer.

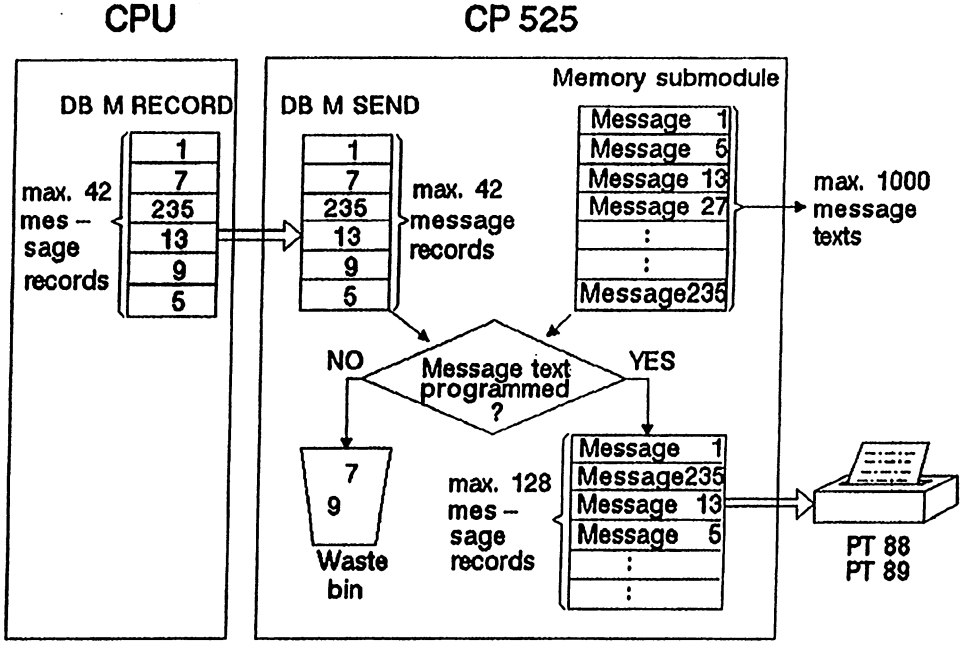


Fig. 3 Processing the messages on the CP 525

2.3.2 Preparing and Outputting the Messages

The CP 525 processes the message records buffered in the FIFO continuously as follows:

- the dynamic data from the message record is added into the preprogrammed static message texts at the positions specified. This dynamic data includes the status, date, time and the parameter
- the control data for the printer attributes (compressed print, expanded print, bold print, underlined, acoustic signal) is added (see Section 2.3.4).

The entire message line is printed out.

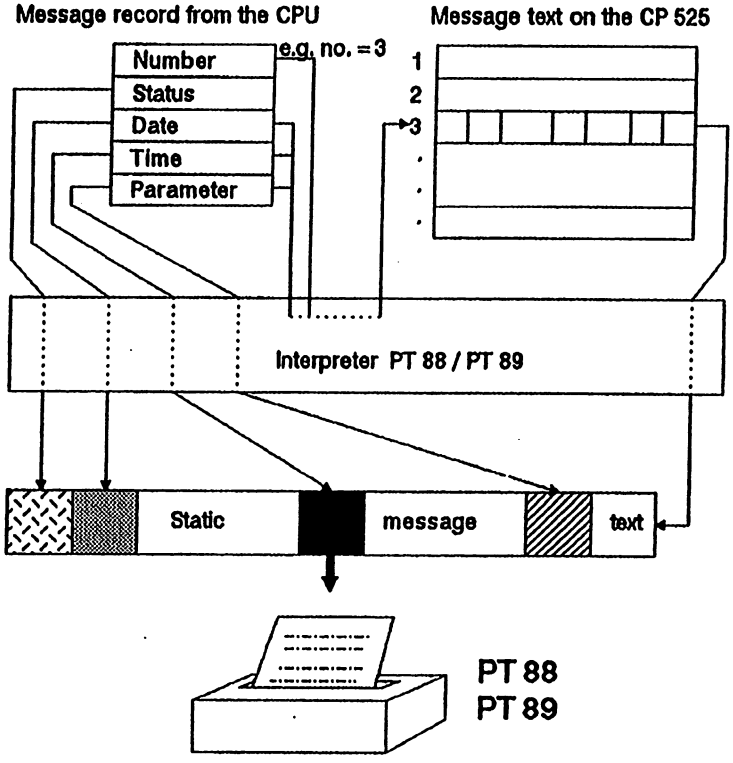


Fig. 4

8

Note: if the CP recognizes that the message record and the static message text do not match (e.g., the message record does not contain a parameter although a dynamic field is programmed for one in the static message text) then instead of the dynamic field, the CP causes filling characters to be printed out.

2.3.3 FIFO Overflow

Although the FIFO has the capacity for 128 messages, it can become full. Space for new entries can only be created if messages are printed out. The FIFO can overflow because of the following:

- The printer
  is switched offline,
  has run out of paper,
  has run out of ink

- A very long list that cannot be interrupted is being printed out.

- A very large number of messages arrive within a very short time.

If the CP 525 recognizes that there is a FIFO overflow, the CP signals the error to the CPU and the last DB M SEND to be sent is **not** entered in the FIFO. Until the cause of the error is remedied, all further DB M SENDs are rejected by the CP 525 with an error. These DB M SENDs are **not** buffered on the CP.

DB M SENDs are accepted again only when the FIFO has space for a complete DB M SEND.

Your STEP 5 program can react in the following ways:

- The CPU ignores the error message and sends further DB M SENDs to the CP 525.

  Reaction of the CP: as long as the error is not corrected, all further DB M SEND's are rejected with an error and not buffered in the FIFO.

- The CPU evaluates the error message, buffers the DB M SEND that was terminated with an error and sends it to the CP 525 again until the CP completes the job without error.

  Reaction of the CP: the DB M SEND that caused the overflow is entered after the error is corrected and is then printed out.

  If the CPU recognizes new disturbances in the meantime, these disturbances are either lost or the S5 program has its own buffer area for temporarily storing these messages.

2.3.4 Print Attributes

When programming with COM 525, you can assign individual print attributes to the static parts and dynamic parts of each message. The following attributes are available:

C = compressed print
E = expanded print
B = bold face
U = underlined
A = acoustic signal of the PT88/PT89

The print attributes are described in more detail in Section 11.

	C	E	B	U	A
fixed texts	yes	yes	yes	yes	yes
dynamic texts	no	yes	yes	yes	yes

!IMPORTANT!: if you select different print attributes for the static and dynamic parts of individual messages the layout of the SML can become confusing. The date, time and status may not be printed out in uniform columns.

!IMPORTANT!: if the messages are longer than the printout lines, there is no guarantee that the print attributes will bring about the expected result on the printout.

2.4 Frames with SMLs

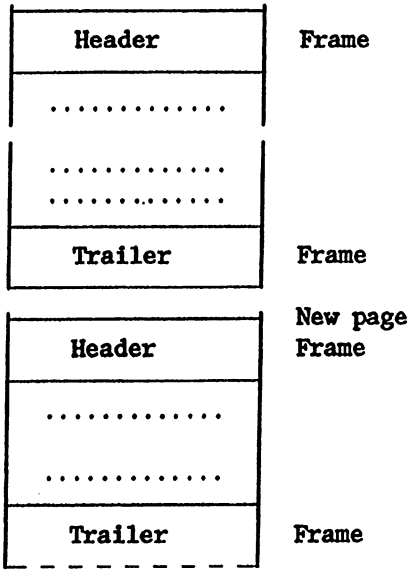
An SML (as well as the other types of lists) can have a list **header** and/or **trailer** known as a frame.

Header = a **frame** printed out at the beginning of the SML and at the beginning of every new page

Trailer = a **frame** printed at the end of every page and at the end of the SML

You can use the same or a different frame for the header and trailer. You can select the frame when you program the SML. All the frames must be programmed under the list type **"frame"**.

Note: frames can only be printed out correctly
- when the single message is not longer than one line (i.e., no line overflow) and
- and when the number of lines in the header plus the number of lines in the trailer < the page length (see Section 12.2.3).



When the single messages that make up the SML are output, the CP 525 counts the lines. It then recognizes that the end of a page is close and outputs the trailer (if this is programmed).

Then a **form feed** is triggered and, if programmed, the header is printed at the top of the next page. Then the single messages are continued.

For more details on frames see Section 6.

8

2.5 Priority of the SML

Messages belonging to the SML are printed out **with highest priority as standard**. CMLs or PSLs are printed out only when all the messages for the SML have been listed (i.e. the message FIFO is empty).

CMLs and PSLs can, however, be interrupted at any point in the list (i.e., between two protocol lines) by SML messages. You can prevent this when programming the PSL or CML with COM 525. If you do, then PSLs and CMLs are printed out completely. The SML messages are buffered in the FIFO until the printer is free.

You decide whether or not the CMLs and PSLs are to be interrupted by SMLs:

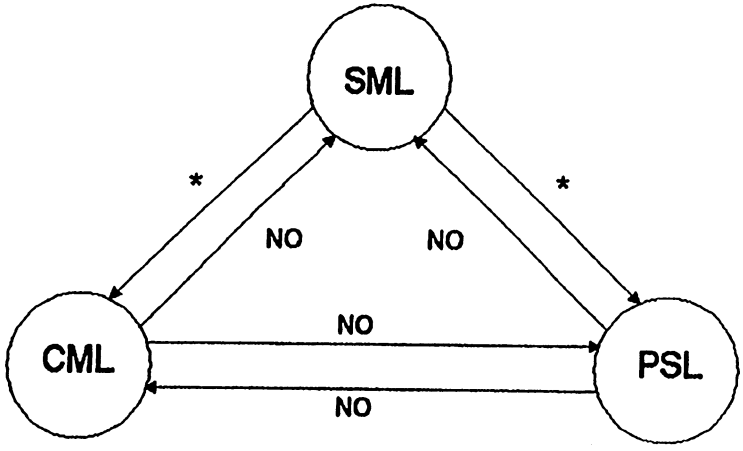


Fig. 5 Priority of the SML

———> means: e.g. SML **has priority over CML**
= SML can interrupt CML

no
———> means: e.g. CML has **no** priority over SML

* means: when programming the PSL or CML you can prevent them being interrupted.

2.6 Limit Values for the SML

The following limit values apply to the messages in the SML:

max. number of SML's per CP 525 interface	1
messages numbers	1 to 2000
max. number of messages which can be stored per CP 525 interface	1000
characters per line	PT88 normal print 80 characters PT88 compressed print 136 characters PT89 normal print 136 characters PT89 compressed print 136 characters
parameters	date time status one parameter (process value) per line

8

3 Current Message List (CML)

Disturbances in the process occur spontaneously. They must be detected by your STEP 5 program and recorded in a data block - **DB M OLDNEW** (status "coming" - bit set). If the disturbance is corrected, the entry must be cleared again in DB M OLDNEW (status "going" - bit cleared).

At any time DB M OLDNEW therefore contains information about the disturbances which are currently affecting the plant. You can have this information printed out as messages on the printer (PT88/PT89).

```
+-------------------------------------------------------+
!              Current message list                     !
+-------------------------------------------------------+
!              Date:     08.12.86                       !
!              Time:     13:14:31 am                    !
+-------------------------------------------------------+
??.??.?? ??:??:?? Temperature too high ???????
??.??.?? ??:??:?? Motor is ?           ???????
??.??.?? ??:??:?? Load = ???? kg       ???????
??.??.?? ??:??:?? Belt no. ?? defect   ???????
??.??.?? ??:??:?? !! Overflow !!       ???????
??.??.?? ??:??:?? Process stable!      ???????
+-------------------------------------------------------+
!           Current message list end                    !
+-------------------------------------------------------+
!              Time:    13:14:31 am                     !
+-------------------------------------------------------+
```

Fig. 6 Example: current message list

3.1 Processing the Messages on the CPU

By sending a SEND DIRECT job to the CP 525 your STEP 5 program can initiate the start of a CML. The CP 525 then requests the **DB M OLDNEW** from the CPU.

The number of the SEND DIRECT job must first be specified as follows:

- in the CP 525 user program (with COM 525)
- in the STEP 5 user program

The data exchange between the CPU and CP 525 takes place using the following handling blocks:

- job for output of a CML:	**SEND DIRECT with job number**
- transfer of the DB from the CPU to the CP 525:	**SEND ALL**

Further information on the handling blocks can be found in the description "Using the handling blocks" in this manual and in /1/ in the list of relevant documentation.

Examples of programming the CPU and the CP 525 can be found in the example of an application "CP 525: event output and listing with the PT88/PT89 printer" in this manual.

8

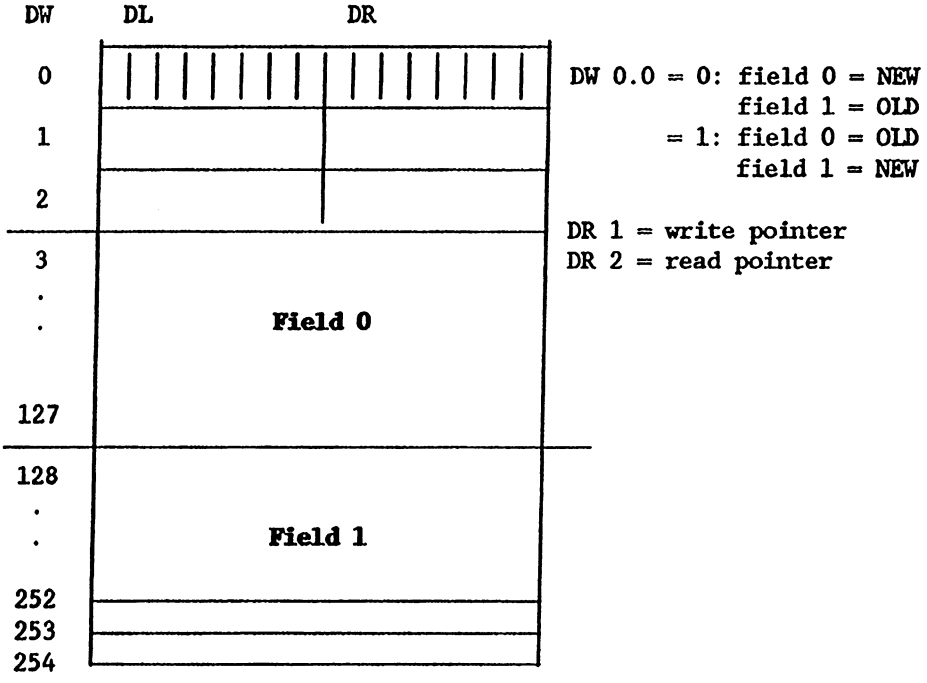
3.2 Interface Between the CPU and CP 525

The interface between the CPU and the CP 525 is the data block **DB M OLDNEW**. A bit in DB M OLDNEW is assigned to every disturbance to be signalled as follows:

- bit set - status "coming"
- bit cleared - status "going"

Structure of DB M OLDNEW

DB M OLDNEW is divided into the areas **OLD** and **NEW**. The NEW area contains the current status of disturbances; the OLD area contains the status as it was in the previous processing cycle on the CPU.



253: this data word contains the lowest **message number**, i.e., the number of the message assigned bit 3.0 or 128.0 in the areas OLD/NEW.

254: contains the **number** of words containing bits to which messages are assigned.

!IMPORTANT!: the maximum possible message number is 2000; therefore, the following error messages may be sent to the CPU:

MESSAGE NUMBER > 2000 or NUMBER > 125 => error

MESSAGE NUMBER + (16 * NUMBER) - 1 > 2000 => error

After each processing cycle the areas NEW/OLD are swapped over when DW 0.0 is complemented as follows:

Cycle	n-3	n-2	n-1	n	n+1	n+2	n+3
DW 0.0	0	1	0	1	0	1	0
Field 0	NEW	OLD	NEW	OLD	NEW	OLD	NEW
Field 1	OLD	NEW	OLD	NEW	OLD	NEW	OLD

Each bit in fields 0 and 1 has a message number assigned to it. This assignment is made according to the contents of DW 253. For DW 253 = 1 (i.e. the lowest message number = 1) the following assignment results:

Relative DW number Field 0/1	Bit number 15	14	13	12	11	10	.	.	2	1	0
0	16	15	14	13	12	11	.	.	3	2	1
1	32	31	30	29	28	27	.	.	19	18	17
2	48	47	46	45	44	43	.	.	35	34	33
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
123	1984	1983	.	.	.	.	.	.	.	.	.
124	2000	1999	1998	1997	1996	1995	.	.	1987	1986	1985

The specification of the DW number for field 0 (and for field 1) is relative to the start of the field. For field 0 the distance to the DB start is 3 data words; for field 1 it is 128 data words.

Example: the message with the number 27 corresponds to the bit with the address DW 4.10 in field 0 and DW 129.10 in field 1.

If there is an active disturbance ("coming") the bit assigned to the message must be set to 1 in the new value area in DB M OLDNEW.

The CP 525 evaluates only the new value part of DB M OLDNEW, which is determined by the value of DR 0. This contains the image of the disturbance status on the CPU at the time when the DB M OLDNEW is transferred from the CPU to the CP 525.

3.3 Processing the Messages on the CP 525

When your STEP 5 program starts, the CML (with SEND DIRECT to the CP 525), the CP requests the DB M OLDNEW from the CPU. As soon as it receives the CP the DB buffers the new value part for the whole of the time that the CML is output on the CP 525. This ensures that the CML is a complete image of the disturbance status at a defined time.

The CML is structured and output in lines. The CP 525 proceeds as follows:

- searches the DB M OLDNEW (new value part) for set disturbance bits
- ascertains the number of this disturbance
- checks whether there is a message text programmed for this number on the CP 525
  if "no" the disturbance is ignored
  if "yes" the corresponding message text is output on the PT88/PT89
- the programmed print attributes are assigned to the static parts of the message line in the list
- ensures that **instead of the dynamic parts** (date, time, status, parameter),
  "?" will be printed out to match the number of characters programmed for the field.

Note: all the dynamic parts of a message relate to the time at which the event being signalled was detected. When the CML is output it is known only that disturbances exist, but not, however, when they were detected and which dynamic values were current at the time! As a result the dynamic values for the date, time and parameter are missing. These are not listed in the CML; instead a corresponding number of "?" is printed out.

IMPORTANT!: "?" is printed out correctly only if an SML has been triggered by the CPU at least once before the CML is started.

3.4 Frames with CML

As with the other list types, the CML can have a list **header** and/or **trailer**. The header and trailer are selected when the CML is programmed and first programmed as list type "**frame**".

Header = a **frame** printed out at the beginning of the CML and at the beginning of every new page.

Trailer = a **frame** printed out at the end of every page and at the end of the CML.

The header and trailer are both optional. You can use the same frame or a different one for the header and trailer.

The CP 525 lists the individual messages of the CML and counts the lines. Before the end of a page is reached, the CP recognises this condition and outputs the trailer. Then a **form feed** is triggered and the header (if programmed) is printed out at the beginning of the next page. Following this, the output of the single messages is continued.

Note: frames can be correctly printed out only
- when the individual message is not longer than one line (i.e. no line overflow) and
- when the number of lines in the header plus the number of lines in the trailer < the page length (see Section 12.2.3).

For further information on frames see Section 6.

3.5 Priority of the CML

CMLs can be interrupted at any time between two lines by SML disturbance messages. The CML itself cannot interrupt a PSL or an SML. The CML can also not be interrupted by a PSL.

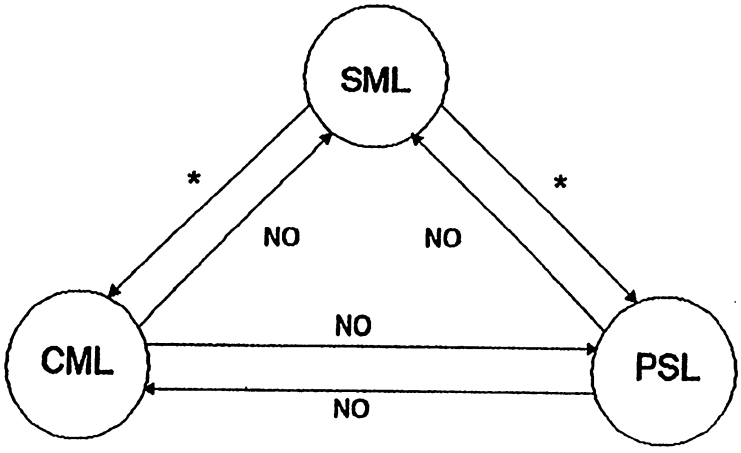


Fig. 7 Priority of the CML

————> means: e.g. SML **has priority over CML**
= SML can interrupt CML

no
————> means: e.g. CML has **no** priority over SML
= CML cannot interrupt SML

* means: when programming the PSL or CML you can prevent their being interrupted.

3.6 Limit Values for the CML

The following limit values apply to the messages in the CML:

max. number of CMLs per CP 525 interface: 1

max. number of list lines (= messages) per CML: 1000
(header, trailer, not included)

4 Process Status List (PSL)

The CP 525 can output data and statuses from the process (e.g. temperatures, pressures, counts) in process status lists on the printer. A PSL consists of a sequence of a maximum **99 list lines**, which together provide an image of the process at a defined time.

If required, a PSL can be printed out with a frame header and/or trailer.

```
+-------------------------------------------------------+
! Process status list no. 1                            !
! Date :  12.09.86  Time :  01:29:12 pm                !
+-------------------------------------------------------+

Temperature in tank :                      +520 Grd/C
Temperature of outer tank wall :           +70 Grd/C
Level in tank                  :           +12 m
Discharge rate                 :            30 kg/h
Deviation from setpoint        :            -2 %

+-------------------------------------------------------+
! Process status list no. 1 end                        !
+-------------------------------------------------------+
```

Fig. 8 Example: process status list

A PSL consists of static and dynamic parts. The static parts are determined when you program the PSL with COM 525. The dynamic values must be acquired by the STEP 5 program on the CPU, collected in a data block and transferred to the CP 525 when the CP requests the data. There is an interface between the CPU and CP 525 for this purpose.

8

4.1 Processing the PSL on the CPU

The CPU must acquire all the dynamic parts of the PSL and store them in a pre-assigned data block. Before the CPU sends the job to the CP to output a PSL, the DB must be available on the CPU and completely updated, since the CP requests this DB from the CPU immediately.

The data exchange between the CPU and CP 525 takes place using the following handling blocks:

- job for output of a PSL: **SEND DIRECT with job no.**
- transfer of the DB from the CPU to the CP 525: **SEND ALL**

Further information on the handling blocks can be found in the description "Using the handling blocks" in this manual and in /1/ in the list of relevant documentation.

Examples of programming the CPU and the CP 525 can be found in the example of an application "CP 525: event output and listing with the PT88/PT89 printer" in this manual.

4.2 Interface Between the CPU and CP 525

The interface between the CPU and CP 525 is the **data block** assigned for this purpose. This **can** be different from PSL to PSL. The number of the data block is fixed when you pro-gram the PSL in the CP 525 user program. This DB must also be de-fined in the STEP 5 user program.

4.3 Processing the PSL on the CP 525

Your STEP 5 program starts a PSL by sending a SEND DIRECT job with a job number to the CP 525. The CP then requests the assigned DB from the CPU. When the CP 525 receives the DB, it buffers it on the CP 525 for the entire time the PSL is output. This guarantees that all the data for the PSL are acquired at one time and that they belong together.

The PSL is structured and output in lines as follows:

- the static parts are assigned "their" print attributes and
- the dynamic parts are assigned "their" print attributes

and both parts are listed together.

You select print attributes when you program the PSL with COM 525. The print attributes of the static parts are reset for each list line. They apply to all static texts in this list line.

The print attributes of the dynamic parts can be set individually for each field. They then only apply to that particular dynamic field (i.e. not for the whole line). After this field the attributes for the static part apply once again.

In the assigned DB the dynamic values can be stored in the usual SIMATIC S5 formats as follows:

Format	Significance	Range of values
BINARY	1 bit binary	0,1
CHAR	character	ASCII
BYTE	1 byte binary	0 ... 255
16BIT BCD	16 bits in BCD code	-999 ... +999
16BIT FXP	16 bits in fixed point format	-32768 ..... +32767
32BIT BCD	32 bits in BCD code	-9999999 .... +9999999
32BIT FXP	32 bits in fixed point format	-2^{31}... 2^{31} -1
32BIT FLP	32 bits in floating point format	approx.$-999999 * 10^{9}$ ... $-999999 * 10^{-9}$ 0 $+999999 * 10^{-9}$ ... $+999999 * 10^{9}$
TIMER	16 bits BCD code + time base (example: 154.2)	0 ... +999 0 ... 3
COUNTER	16 bits BCD code	0 ... +999

!IMPORTANT! Make sure when programming the STEP 5 user program and the CP 525 user program that you use the **same** data types and same addresses in the DB!

4.4 Frames with PSLs

As with the other list types, the PSL can have a list header and trailer. You select the header and trailer when programming the PSL and program them as a "frame" list type.

In contrast to SMLs and CMLs, the header for a PSL is printed only **at the start** and the trailer only **at the end** of a list (i.e. not on every page).

Further information about frames can be found in Section 6.

4.5 Priority of the PSL

PSLs can be interrupted at any time between two list lines by SML messages.

The PSL cannot interrupt an SML or a CML. The PSL cannot be interrupted by a CML. When programming with COM 525, you can prevent the PSL being interrupted.

4.6 Limit Values for the PSL

max. no. of PSL's per CP 525 interface, theoretical *)	199
max. no. of list lines per PSL (header, trailer not included)	99
max. length of a dynamic field in characters	80
max. no. of dynamic fields per PSL line	40
max. no. of dynamic fields in one PSL *) **)	approx. 1400
max. length for process status variables in characters	60
max. no. of statuses for PSL variables	8

*) In practice this number depends on
- the capacity of the memory submodule,
- the capacity of the DB (max. 512 bytes),
- the type of dynamic data and their space requirements in the DB,
- the number of places the dynamic fields require in the list lines.

**) If process status variables are programmed, the maximum number of dynamic fields is less since the space requirements for process status variables are much greater (up to 8 alternatives per dynamic field).

5 Chained Lists (CL)

The current status of your plant or part of the plant can be printed out in a process status list (PSL). PSLs can be a maximum of 99 lines long.

If you require more than 99 lines, you should program and start a **chained list**.

```
+-------------------------------------------------------+
! Process status list no. 1                             !
! Date :  12.09.86  Time :  01:29:12 pm                 !
+-------------------------------------------------------+

Temperature in tank :                      +520 Grd/C
Temperature of outer tank wall :            +70 Grd/C
Level in tank                  :            +12 m
Discharge rate                 :             30 kg/h
Deviation from setpoint        :             -2 %

+-------------------------------------------------------+
! Process status list no. 1 end                         !
+-------------------------------------------------------+

+-------------------------------------------------------+
! Process status list no. 2                             !
+-------------------------------------------------------+

Throughout of distributor network :

     valve 1   :   129 l/min
     valve 2   :    20 l/min
     valve 3   :    15 l/min

+-------------------------------------------------------+
! Process status list no. 2 end                         !
+-------------------------------------------------------+
```

Fig. 9 Example: chained lists

8

Chaining lists simply means that **up to 8 process status lists** are started automatically. This process saves code and job numbers on the CPU. There are a maximum 199 job numbers per CP interface which allows you to do the following:

- print out the **status of the whole plant** since all the PSL's of the different parts of the plant are output automatically
- structure a process status list with **more than 99 lines**, by spreading the list over several PSLs.

All the parameters of the individual PSLs such as the

- numbers of the individual data blocks
- the names of the list headers
- the names of the list trailers

are fixed when you program the **PSL**. When programming the chained list, simply specify the names of the individual PSLs that are to be chained.

For more information on PSLs see Section 4.

5.1 Processing the CL on the CPU

Your STEP 5 program starts a chained list by sending a SEND DIRECT job with a job number to the CP 525. The job number must be specified in advance

- in the CP 525 user program (with COM 525)
- in the STEP 5 user program

Then the CP 525 requests the data block from the CPU for the first PSL to be chained. The CPU then sends all the DBs for the individual PSLs to the CP 525 in succession as they are requested.

The data exchange between the CPU and CP 525 takes place using the following handling blocks:

- job for output of a CL: **SEND DIRECT with job no.**
- transfer of the DB's from the CPU to the CP 525: **SEND ALL**

Further information of the handling blocks can be found in the description "Using the handling blocks" in this manual and in /1/ in the list of relevant documentation.

5.2 Interface Between CPU and CP 525

The interface between the CPU and CP 525 is formed by the data blocks you assign when programming the individual PSLs. This can be the same for all PSLs or differ from PSL to PSL.

For more information on PSLs see Section 4.

5.3 Processing the CL on the CP 525

Your STEP 5 program starts a chained list by sending one SEND DIRECT job with a job number to the CP 525. The CP then processes, in succession, all the PSLs specified in the CL.

This processing is described in detail in Section 4.

Note: with the chained list:
when the CPU starts a CL with a SEND DIRECT job, the CP 525 automatically processes the individual PSLs you specified when programming with COM 525.

with single process status lists:
the CPU must start each PSL with a SEND DIRECT job.

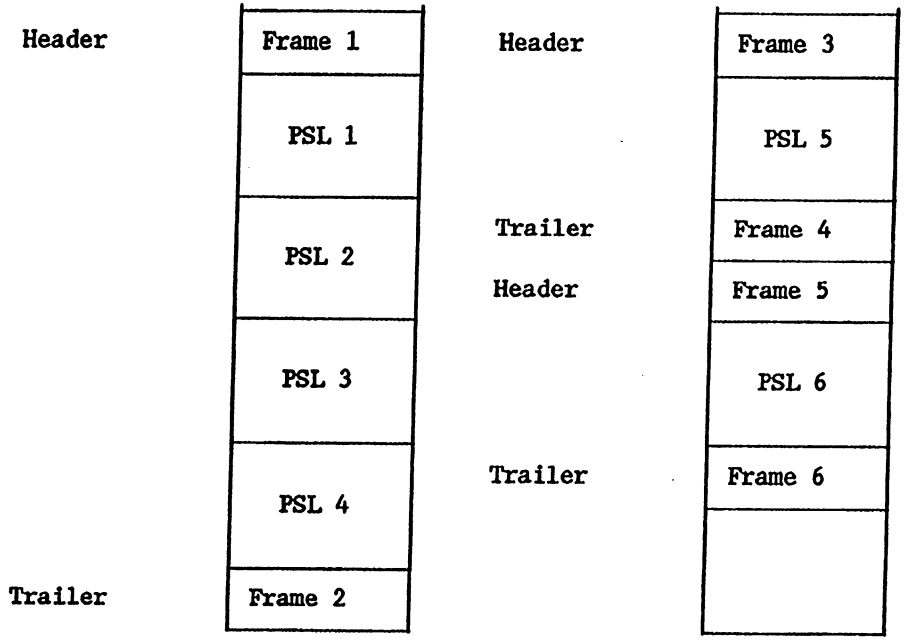
5.4 Frames with the Chained List

When programming the chained list with COM 525 you **cannot** specify a header or trailer specially for the CL. The headers and trailers must be assigned for each individual PSL.

By programming the PSLs accordingly you have a wide variety of options as follows:

- each PSL has its own header and trailer
- only the first PSL has a header
- only the last PSL has a trailer
- any of the chained PSLs can have a header and/or trailer
- each PSL begins with form feed
- only the first PSL begins with form feed

Example:



For more information about frames see Section 6.

5.5 Limit Values for Chained Lists

max. no. of PSLs per CP 525 interface, theoretical *) 199

max. no. of chained PSLs per CL 8

max. no. of variables per CL, theoretical *) approx. 8 * 1400
(per PSL approx. 1400)

max. no. of DBs per chained list: 8
(DB's for headers/trailers are not included)

*) In practice this number depends on
- the capacity of the memory submodule
- the capacity of the DBs (max. 512 bytes per DB)
- the type of data and their space requirements in the DB
- the number of places the dynamic fields require in the list lines

6 Frames (List Headers/Trailers)

All three types of list (SML, CML, PSL) can have a list **header** and/or **trailer**. These are parts of lists which are programmed independently from the list under the term **"frame"**.

Header = a **frame** printed at the beginning of the list and at the beginning of each new page (except PSL: only at the beginning of the list)

Trailer = a **frame**, printed at the end of every page or at the end of the list (except PSL: only at the end of the list)

You can decide whether to have a header or trailer printed out by entering the required frame name when you program the SML, CML or PSL under the header or trailer. The structure of the frame is similar to that of the PSLs.

!IMPORTANT! Frames **cannot** be started by the CPU. Headers and trailers can only be printed out in conjunction with SML, CML and PSL.

Frames can be used as required, e.g., as follows:

- a frame can be used as a header in one list and a trailer in another
- all lists can have the same frame as header and/or trailer
- each list can be supplemented by its own header and/or trailer, i.e. different frames for different lists

When programming a frame with COM 525, you select a data block that is to contain the current dynamic values of the frame.

!IMPORTANT! Only data words 0 to 127 are evaluated in this DB.

6.1 Processing the Data on the CPU

If the STEP 5 program starts a list, the CP 525 checks whether the list requires a frame and if so, requests the data block from the CPU that contains the dynamic values for the frame.

The data exchange between the CPU and CP 525 takes place using the following handling blocks:

transfer of the DBs from the CPU to the CP 525: **SEND ALL**

Further information on the handling blocks can be found in the description "Using the handling blocks" in this manual and in /1/ in the list of relevant documentation.

Examples of programming the CPU and the CP 525 can be found in the example of an application "CP 525: event output and listing with the PT88/PT89 printer" in this manual.

6.2 Interface Between CPU and CP 525

The interface between the CPU and CP 525 is the assigned data block and can be different from frame to frame. You decide the number of this DB in the CP 525 user program. You must also define this DB in the STEP 5 user program.

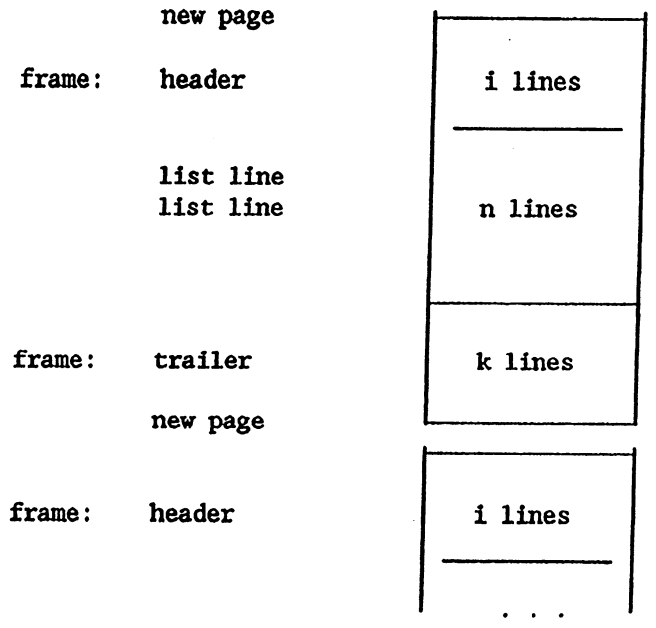
6.3 Processing the Data on the CP 525

After the CPU has started an SML, CML or PSL, the CP first checks whether a header and/or trailer has been programmed for the list. If 'yes' it requests the DB for the frame from the CPU - first the DB for the header and then the DB for the trailer.

The DBs transferred from the CPU with SEND ALL (only data words 0 to 127) are buffered on the CP 525 until the complete list is printed out. This ensures that the dynamic data of the header or trailer remain the same during the output of the list (status as at the start of the list).

The CP 525 counts the lines as they are printed out on the printer. The CP 525 then recognizes the end of a page, adds the trailer and triggers a form feed to the next page where the header is then printed (if programmed). With PSLs the header and trailer are only output at the beginning and end of the list.

Example of frames with SML and CML:



The programmed page length = i + n + k lines, where i, k = 0 is possible if no header or trailer has been programmed.

6.4 Limit Values for Frames

max. no. of frames:	unlimited
can be started by CPU:	no
max. no. of lines per frame:	9
max. length of a dynamic field in characters	80
max. no. of dynamic fields per frame line:	40
max. no. of dynamic fields in a frame, theoretical *):	9 * 40

in the DBs for frames only the data words **0 to 127** are evaluated

*) In practice this number depends on
- the capacity of the memory submodule
- the capacity of the DB (max. 128 words)
- the type of dynamic data and their space requirements in the DB
- the number of places the dynamic fields require in the list lines

7 Update Sequential Message List

The function 'update sequential message list (UPDATE SML)' is used when you want to change the dynamic values in list headers/ trailers at certain times, (e.g. change of shift) while the SMLs are running. For example, you may want to change the name of the shift, or shift engineer, etc.

```
+-------------------------------------------------------+
! Sequential message list for: 02.13.87 at: 02:29 pm!
!                                                       !
! Early shift    Shift engineer :  Bownes               !
+-------------------------------------------------------+
02.13.87 02:29:26 pm  Tank 1 :  max. level     COMING
02.13.87 02:29:27 pm  Tank 1 :  overflow       COMING
02.13.87 02:29:29 pm  Tank 1 :  overflow       GOING
02.13.87 02:29:30 pm  Tank 1 :  max. level     GOING
+-------------------------------------------------------+
! 02:29 pm  Early shift  Shift engineer : Bownes       !
+-------------------------------------------------------+

+-------------------------------------------------------+
! Sequential message list for: 02.13.87 at: 02:31 pm!
!                                                       !
! Late shift     Shift engineer :  Richmond             !
!-------------------------------------------------------!
02.13.87 02:31:03 pm  Tank 1 :  max. level     COMING
02.13.87 02:31:06 pm  Tank 1 :  max. level     GOING
```

Fig. 10 Example: update sequential message list

Basic information about SMLs:
The sequential message list can have a list header and/or trailer (see Section 2.4). If you program a header and/or trailer, they are printed out at the start and end of each new page and at the start and end of the SML. If there are dynamic values in the header and trailer, you can select a DB as the data source for these values, both for the header and for the trailer.

If the SML is started for the first time, the CP 525 requests these data blocks from the CPU and stores their contents. This guarantees that within one SML the dynamic values of the header and trailer are not changed. This also means that the data blocks need not be requested from the CPU for each SML job. This reduces the load on the CPU and CP.

Update SML:
If you wish to change the dynamic values for the list header and trailer while an SML is running, you can have **new dynamic values** sent from the CPU to the CP in DBs. You use the job **UPDATE SML** that is triggered by the CPU when the CPU calls the handling block SEND DIRECT with a job number.

7.1 Processing the Job on the CPU

The dynamic values for the header and trailer of the SML are stored on the CPU in a DB (one for header and one for trailer, although it is possible to have dynamic values for both in one DB).

If the CPU starts the job, the DBs must already contain the updated values that are to appear in the header/trailer of the SML. Immediately after the start of the job the CP 525 requests these DBs from the CPU.

The data exchange between the CPU and the CP 525 is carried out by means of handling blocks as follows:

- job "update SML": **SEND DIRECT with job number**
- transfer of the data blocks from the CPU to the CP 525: **SEND ALL**

More information on handling blocks can be found in the description "Using the handling blocks" in this manual and under /1/ in the list of relevant documentation.

7.2 Interface Between the CPU and CP 525

The interface between the CPU and CP 525 is formed by one or two data blocks. The numbers of these data blocks are specified when you program the frame with COM 525. These DBs must also be defined in the STEP 5 user program.

!IMPORTANT! With frames, only data words DW 0 to a maximum DW 127 are evaluated.

7.3 Processing the Job on the CP 525

Your STEP 5 user program starts the 'update SML' by sending a SEND DIRECT job with a job number to the CP 525. This job checks whether a sequential message list with header and/or trailer has already been started. If this is the case, the job requests the DBs that you specified when programming the frame for this SML. The CPU transfers these with a SEND ALL to the CP 525. These DBs contain the most recent dynamic values. The CP buffers the values transferred (DW 0 - DW 127 for each block). From this point onwards these values are used as the source for the output of dynamic values in the header/trailer of the SML.

So that you can recognize whether the dynamic values of the header and trailer for the SML have been updated after the UPDATE SML has started the following occurrs:

- the previous messages of the SML are completed with the trailer which still contains the "old" dynamic values
- new messages are output with the header which already contains the "new" dynamic values. If you programmed it, a form feed for the beginning of the list is performed.

8 Update Group Inhibit Bits

Very often a plant is divided into various parts. These may

- be started up
- have maintenance work carried out, or
- be switched off

at different times. The corresponding messages and process statuses are irrelevant during these times and only increase the load. While a part of a plant is starting up there are usually a lot of disturbance messages since the plant is usually unstable during the start-up phase.

For this reason it is possible to prevent the listing of such messages or process statuses by assigning each list line to a **group** and release or inhibit the listing of this group.

The messages are assigned to groups when programming the lists with COM 525. There are max. **16 groups per CP 525 interface** available. Each of the 16 groups is assigned a bit in a defined data word. These bits are known as **group inhibit bits** (GIB).

With the GIB's you can control the output of list lines.

GIB status = **1**: group **inhibited**
All messages and PSL lines which are assigned to this group will **not** be listed.

GIB status = **0**: group **released**
All messages and PSL lines which are assigned to this group will be listed.

The group inhibit bits (GIB) can be set or reset by the CPU at any time. The **update group inhibit bits** job is available for this purpose.

!IMPORTANT! After a cold restart carried out from the PG and following a SYNCHRON, the group inhibit bits on the CP 525 are reset, i.e. all groups are enabled.

8.1 Processing the Group Inhibit Bits on the CPU

The STEP 5 program maintains the current status of the group inhibit bits in a defined data word. Both the DW address and the DB number have already been programmed

- in the CP 525 user program (with COM 525)
- in the STEP 5 user program

By setting or resetting a group inhibit bit, a group can be inhibited or enabled at any time. The CPU sends the job "update group inhibit bits" to the CP 525 by means of a SEND DIRECT job. The CP 525 immediately requests the assigned DB from the CPU.

The data exchange between the CPU and the CP 525 is carried out by means of handling blocks:

job "update SML":	**SEND DIRECT with job number**
transfer of the DB from the CPU to the CP 525:	**SEND ALL**

More information on handling blocks can be found in the description "Using the handling blocks" in this manual and under /1/ in the list of relevant documentation.

8.2 Interface Between CPU and CP 525

The interface between the CPU and CP 525 is the defined data word in the declared data block. This contains the group inhibit bits for the 16 groups of messages. The following applies:

	DL n	15	14	13	12	11	10	9	8
	group no.	15	14	.	.	.	.	9	8
DW n									
	DR n	7	6	5	4	3	2	1	0
	group no.	7	6	.	.	.	.	1	0

DW = data word
DL = left data byte of DW
DR = right data byte of DW
n = number of the data word

Bit = 1 means message group inhibited
Bit = 0 means message group released

8.3 Processing the Group Inhibit Bits on the CP 525

Immediately after the SEND DIRECT job has been started the CP 525 requests the data block with the defined data word from the CPU. This sends the DB with a SEND ALL to the CP. The CP 525 stores the data word and interprets the contents immediately as group inhibit bits for the 16 groups.

The output of each list line is dependent on the status of the group inhibit bits for its group, which is assigned when programming the list line.

Inhibit bit = **1**: list line or message **not** output
Inhibit bit = **0**: list line or message output on PT88/PT89

9 New Page

The CP 525 automatically initiates a **form feed** when the programmed page length is reached.

Your STEP 5 user program can also output a form feed **at any time**. Using a SEND DIRECT job with a job number, the CPU starts the "new page" job on the CP 525.

Note: this job can only become effective as follows:

- when all the jobs in the queue of the CP 525 are processed
- when all the texts previously transferred to the PT88/PT89 and buffered in its input buffer are printed out
- when the printer head is not already at the start of a page, i.e. the form feed has just been performed.

Further information can be found in the example of an application "CP 525: event output and listing with the PT88/PT89 printer" in this manual.

10 Resetting the Page Number

The CP 525 counts the list pages printed out on the PT88/PT89. It begins with page 1 after the CP 525 has been switched on, after a cold restart of the interface, or after BREAK.

Your STEP 5 user program can reset the page counter on the CP 525 to one **at any time**. By means of a SEND DIRECT job, the CPU starts the job "reset page number" on the CP. A form feed is automatically carried out at the same time.

11 Print Attributes

Each list line (of an SML, CML or PSL) can be printed out individually. You can select from among the following print attributes when programming the lists with COM 525:

C = Compressed print = 17 characters per inch (instead of 10 characters per inch)

E = Expanded print = double character width

B = Bold face

U = Underlined

A = Acoustic signal of the PT88/PT89

You can assign the print attributes individually to the static parts of the list lines (fixed texts) as well as to the dynamic part (date, time, process variable).

The selected attributes are then valid as follows:

- **for the whole list line** for all **static** parts
  (if you selected these attributes when programming the static parts)

- **only for a particular field** (within the line) for all **dynamic** parts
  (if you selected these attributes when programming the dynamic parts)

The attribute "compressed print" is **not** allowed for dynamic parts. The following attributes are allowed:

	C	E	B	U	A
fixed texts	yes	yes	yes	yes	yes
dynamic texts	no	yes	yes	yes	yes

You can combine the print attributes as you wish. The text is then output according to the print attributes assigned to it.

Example: the attributes **ECHUB** are assigned to a particular text. This text is then output as follows:

- in compressed print (C), however
- with double character width (E)
- in bold print (B)
- underlined (U)
- with the acoustic signal of the PT88/PT89 (A)

If dynamic parts with their own attributes are printed out within a static text, the attributes of the static text are used again after the dynamic part has been printed out.

11.1 Print Types

The PT88/PT89 can print out in two print types as follows:

normal print = **10** characters per inch

compressed print = **17** characters per inch

11.1.1 Normal Print

Normal print = 10 characters per inch.
Unless you select the print attribute "C" (compressed print) the text is printed out in normal print. In normal print with normal character width you can print the following:

- maximum 80 characters per line on a PT88
- maximum 136 characters per line on a PT89

11.1.2 Compressed Print (C)

Condensed print = 17 characters per inch.
If you select the print attribute "C", the texts are printed in compressed print. In compressed print with normal character width you can print the following:

- maximum 136 characters per line on a PT88
- maximum 136 characters per line on a PT89

This attribute is accepted only by the PT88/PT89 at the beginning of a line. It then remains valid for the whole line, i.e. for static and dynamic text.

11.2 Expanded Print

The PT88/PT89 can print out as follows:

- normal character width = character matrix 5 * 9 dots
- double character width = character matrix 2 * 5 * 9 dots

11.2.1 Normal Character Width

Unless you select the print attribute "E", the text is printed out in normal character width. A character is then represented by a 5 * 9 dot character matrix.

11.2.2 Double Character Width (E)

If you select the print attribute "E", the text is printed with double character width. Each dot in the character matrix is represented by two dots side by side. A character is then represented by a 2 * 5 * 9 dot character matrix.

11.3 Bold Print (B)

With bold print, each character is printed more than once. There is, however, a difference between heavy print with the impact matrix printer and heavy print with the inkjet printer.

11.3.1 Bold Print with the Impact Matrix Printer (Needle)

A bold face character is produced when the printer prints the character **three times**.

11.3.2 Bold Print with the Inkjet Printer

A bold face character is produced when the printer prints the character **twice**.

11.4 Underlining (U)

If you select the print attribute "underlined" (U), the text is underlined. The underlining is part of the character. It is printed on the bottom position of the character (ninth position). Spaces are also underlined.

Note: the tails of letters such as g, j, p, q become unclear!
Example: doggy becomes doggy.

11.5 Acoustic Signal (A)

The attribute "A" triggers the acoustic signal of the PT88/PT89 (lasts one second). The CP 525 ensures that it is triggered only once per list line even if it is programmed more than once in a line (e.g. for several dynamic parts).

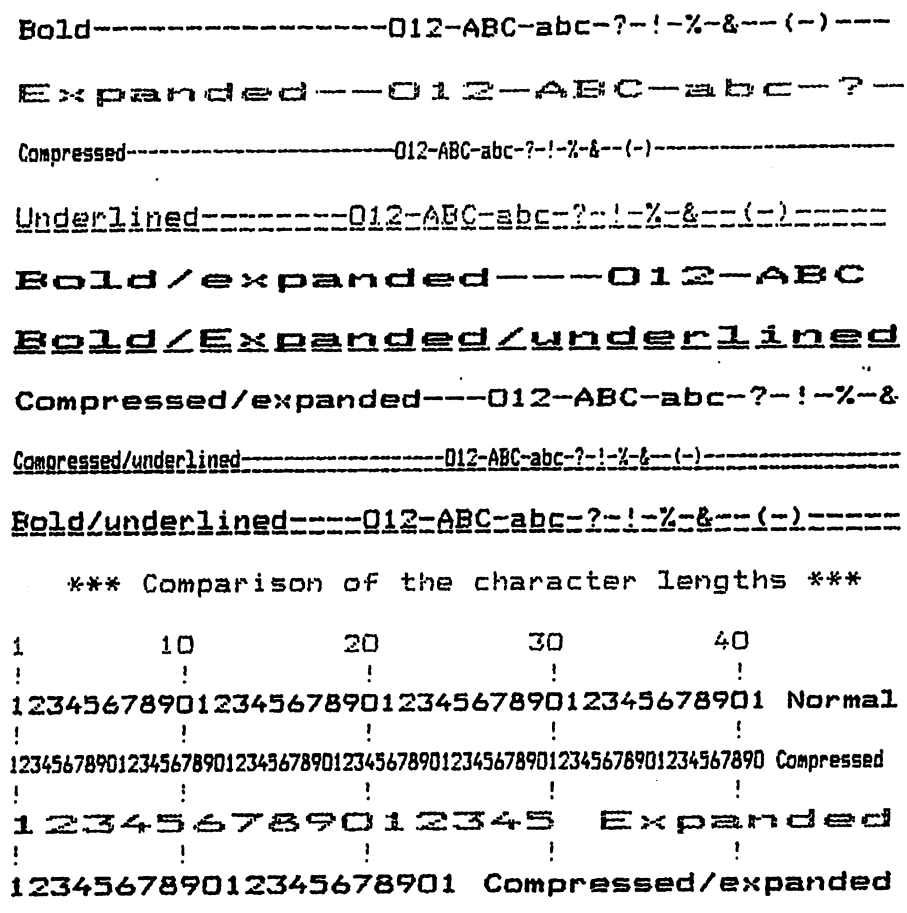


Fig. 11 Example: different combinations of print attributes

8

12 Setting Up

12.1 Requirements

Work through the following check list to see whether your system is complete and capable of operation.

	REQUIRED	MORE INFO
Hardware		
- Programmable controller	x	PC manual
- Communications processor CP 525 (check jumper settings!)	x	Instructions CP 525 (in this manual)
- Programmer PG 685 or PG 635 / PG 675 / PG 695	x	PG manual
- Printer PT88/PT89 (check settings!)	x	PT88/PT89 - User's guide
- Cable connector PG <—> CP 525	x	Instructions CP 525 (in this manual)
- Cable connector CP 525 <—> PT88/PT89	x	Instructions CP 525 (in this manual)
Loadable software		
- Programming software COM 525 Procedure (param. assigned!) Interpreter (param. assigned!)	x	User's guide COM 525 and example of application PT88/PT89 (in this manual)
- Programming software STEP 5	x	Literature on programming with STEP 5
Handling blocks	x	User's guide on the handling blocks for the particular PC and processor type

12.2 Settings on the CP 525

Check the settings of the jumpers on the CP 525 and the parameter assignment of the loadable software.

12.2.1 Jumper Settings on the CP 525

On the CP 525 you must set the jumpers as described in the instructions for the CP 525 in this manual (assignment of interface numbers). **No** jumper settings are required for the interpreter and procedure.

12.2.2 Assigning Parameters to the Procedure

The data rate and data format are set using the COM 525 programming package. These specifications must agree with the corresponding settings on the PT88/PT89 printer.

You can select the following:

- data rate: 110, 200, 300, 600, 1200, 2400, 4800, 9600 bps

- parity: even, odd

- number of stop bits: 2, 1.5, 1

More information can be found in Section 13 "Procedure LAUFPT88".

12.2.3 Parameter Assignment for the Interpreter

When assigning parameters to the interpreter with COM 525, set the following:

printer model	PT88 or PT89
printer type	inkjet or impact matrix (needle) inkjet: bold print equals double print impact: bold print equals treble print
lines per inch	3, 4, 6
page length	in lines 1 to 99
page width	in characters 1 to 80 with PT88 1 to 136 with PT89
format (date)	D, M, Y
format (time)	H, M, S
separator for date	. * + - , ' / : ;
separator for time	. * + - , ' / : ;
time format	German = 24 hours English = 12 hours am/pm
character set	German English International ASCII French/Belgian Spanish Norwegian/Danish Swedish/Finnish

The character set applies only to the printer and has no effect on the representation on the PG monitor. On the PG you can switch over to the German character set by pressing LOCK + SHIFT simultaneously.
More information can be found in the user's guide for the programming package COM 525 in this manual.

12.3 Settings on the PT88/PT89 Printer

The settings required for the PT88/PT89 are made at the coding switch on the printer (see description of PT88/PT89).

12.3.1 Coding Switch on the Central Controller

The coding switch is under the cover on the central controller. It is accessible after the front cover has been opened.

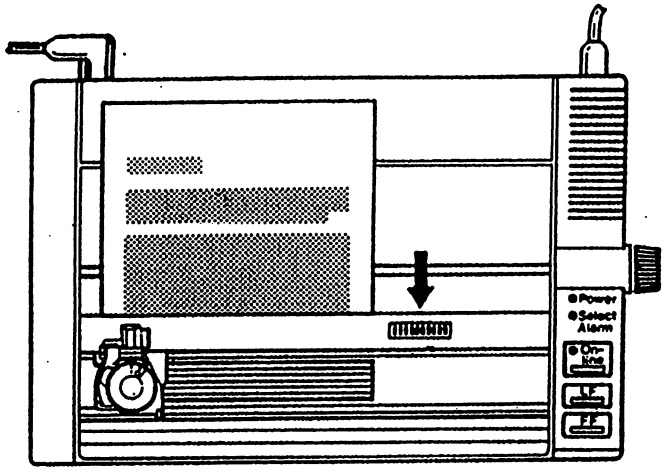
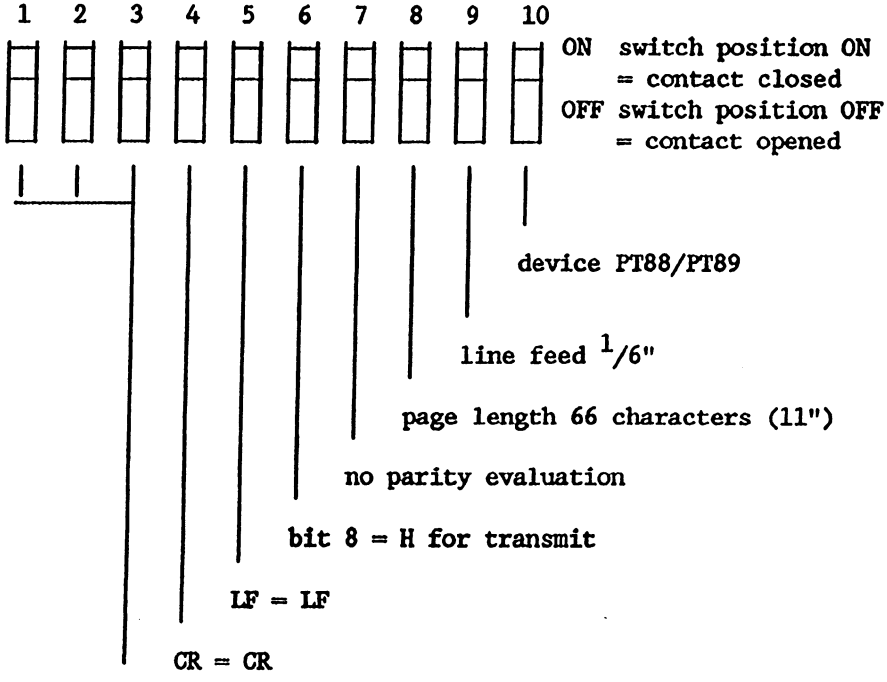


Fig. 11 PT88/PT89 position of the coding switch

Using the coding switch on the central controller you can select certain standard functions. The settings of the coding switch are read by the printer controller after the power has been switched on. Changing the switch setting while the printer is switched on has no effect.

Note: when you cold restart the interface, if break is set or if you switch the printer on/off, the variables set during the parameter assignment for the interpreter (character set, lines per inch and page length) are assigned by the software.

Factory setting of the coding switch



Switch functions		Switch 1	2	3	4	5	6	7	8	9	10	
switches 1 to 3 no significance - character set determined by COM 525 "interpreter parameters"		x	x	x								2)
paper handling	CR = CR LF = LF				*	 *						1)
parity	even odd						* o	o o				
page length	11" 12"								x x			2)
line feed	1/6"									*		1)
device	PT 88 PT 89										* o	3)

* = ON; o = OFF; x = no significance

1) means: fixed setting (for more details see description of the PT88/PT89)

2) means: is not evaluated; set by means of the interpreter parameters with COM 525

3) means: with PT89: switch must be ON, if you use paper with 80 characters per line

8

12.3.2 Interface Adapter SAP-S1 (V.24/V.28)

The following figure shows the position of the operating mode switches S1 and S2 on the SAP-S1.

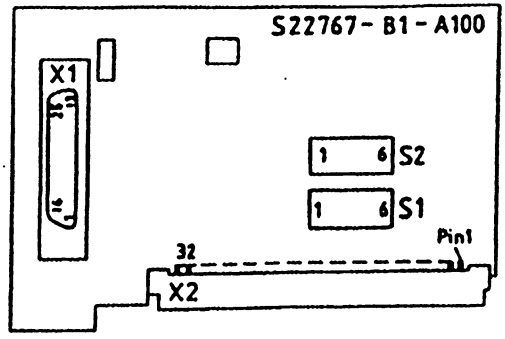


Fig. 12 Position of the operating mode switches S1 and S2 on the SAP-S1

Settings of the operating mode switches S1 and S2 for the various operating modes

Operating mode	Switch S1 1	2	3	4	5	6	Switch S2 1	2	3	4	5	6
data transmission rate bps 110	OFF	OFF	OFF									
200	ON	OFF	OFF									
300	OFF	ON	OFF									
600	ON	ON	OFF									
1200	OFF	OFF	ON									
2400	ON	OFF	ON									
4800	OFF	ON	ON									
9600	ON	ON	ON									*)
operation with X-ON/X-OFF protocol line 103, 104, 108.2				OFF	OFF	ON	OFF	ON	OFF	ON	OFF	OFF

*) presetting for the LAUFPT88 procedure in COM 525

In order to operate the PT88/PT89 as the listing printer with the CP 525 only the X-on/X-OFF protocol is permitted.

8

12.3.3 Interface Adapter SAP-S2 (TTY/20 mA)

The following figure shows the position of the operating mode switches on the SAP-S2.

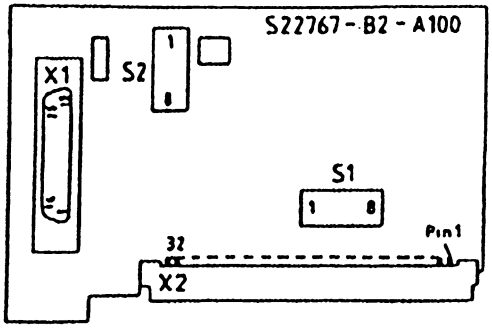


Fig. 13 Position of the operating mode switches S1 and S2 on the SAP-S2

Settings of the operating mode switches S1 and S2

Operating mode	Switch S1								Switch S2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
data transmission rate bps 110	OFF	OFF	OFF													
200	ON	OFF	OFF													
300	OFF	ON	OFF													
600	ON	ON	OFF													
1200	OFF	OFF	ON													
2400	ON	OFF	ON													
4800	OFF	ON	ON													
9600	ON	ON	ON													*)
operation with X-ON/X-OFF protocol and own supply				ON	OFF	ON	OFF	OFF	OFF	OFF	ON	ON	OFF	OFF	ON	ON

*) presetting of the LAUFPT88 procedure in COM 525

For operating the PT88/PT89 as the listing printer with the CP 525 only operation with the X-ON/X-OFF protocol and own supply is permitted.

8

12.3.4 Interface Adapter SAP-S3 (TTY/20 mA; V.24)

The following figure shows the position of the operating mode switches on the SAP-S3.

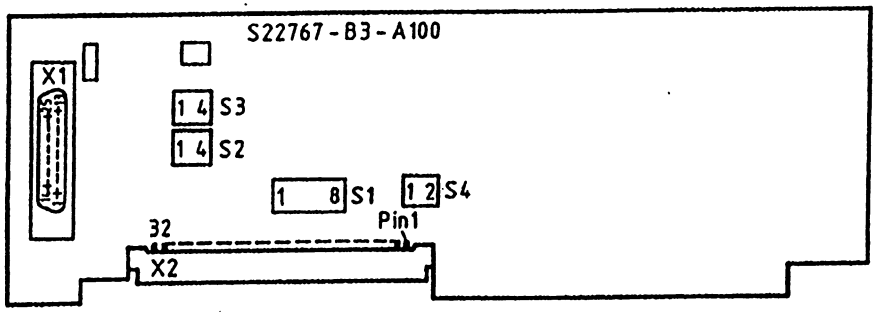


Fig. 14 Position of the operating mode switches S1, S2, S3 and S4 on the SAP-S3

Settings of the operating mode switches

Operating mode switch S4 - setting the interface variant as follows:

Interface	Position
RS-232-C (V.24/V.28)	1
TTY	2

Settings for the data transmission rate and the protocol

Operating mode	Operating mode switch S1							
	1	2	3	4	5	6	7	8
data transmission rate . bps								
110	OFF	OFF	OFF					
200	ON	OFF	OFF					
300	OFF	ON	OFF					
600	ON	ON	OFF					
1200	OFF	OFF	ON					
2400	ON	OFF	ON					
4800	OFF	ON	ON					
9600	ON	ON	ON					*)
operation with X-ON/X-OFF protocol				ON	OFF	OFF	ON	OFF

*) presetting for the LAUFPT88 procedure in COM 525

To operate the PT88/PT89 as a listing printer with the CP 525, use only the X-ON/X-OFF protocol.

Operating mode switches S2 and S3 - setting is required only for TTY operation (with V.24 no significance)

S2				S3				
ON	ON	ON	ON	OFF	OFF	OFF	OFF	own power (from PT88) required

12.4 Start-up Phase of the CP 525

12.4.1 Starting Up after Switching On the Power

After you have switched on the power the two **LEDs** on the front panel of the CP 525 light up. The CP 525 then checks whether

- the RAM and EPROM are functional and
- the interpreter and procedure are present and complete in the memory submodule

Following the start-up phase which takes approximately two seconds the CP switches off the LEDs of the loaded interfaces. If the mode selector is set to STOP or PGR the LEDs do not go off.

Following the start-up phase the CP 525 expects the **SYNCHRON** job from the CPU.

12.4.2 SYNCHRON Job from the CPU

The SYNCHRON job arrives after the start-up phase

The SYNCHRON HDB is necessary for processing the jobs, SEND and RECEIVE (to ensure that the field length for data transfer CPU ---> CP 525 is agreed). The SYNCHRON HDB is usually called up in the STEP 5 user program in the organization blocks OB 20, OB 21, OB 22.

When the SYNCHRON job runs through without errors, the CP is ready to process jobs (SEND, RECEIVE) from the CPU.

One possible reaction before SYNCHRON is as follows:

- jobs (except SYNCHRON) could arrive from the CPU before the SYNCHRON has run; these are rejected with "not ready" (error number 81H in the PAFE byte of the corresponding job).

A new SYNCHRON job arrives for an interface which is already in operation (for example as a result of pressing the RESET button on the CPU)

If a **PC job** is currently being processed, it will be terminated with the error number 7H in the condition codeword. The error number 16H is signalled in the error message area of the SYSTAT.

If the SYNCHRON is run through **without** errors the job statuses (and therefore the condition codewords) of all jobs are cleared. The error number 7H mentioned also disappears. The entries in the error message area of the SYSTAT remain.

If the SYNCHRON is run through **with** errors, all the PC jobs which were set to "running" during the restart phase are also terminated with the error number 7H. (16H is once again entered in SYSTAT).

Following this, new PC jobs (except SYNCHRON) are blocked with the message "not ready" (no. 81H in the PAFE byte of the corresponding job).

12.4.3 Switch Settings RUN/STOP/PGR

The settings of the mode selector on the front panel of the CP 525 have the following significance:

- **RUN**

The PT88 interpreter processes jobs arriving from the CPU (condition: a SYNCHRON job has run through without errors).

The LEDs of the loaded interfaces are no longer lit on the front panel of the CP 525.

- **STOP** or **PGR**

These switch settings are identical. Jobs arriving will be rejected with an error message as follows:

- PC jobs (SEND) are terminated with the error number CH in the condition codeword. The error number 27H is written into SYSTAT.

Switching the selector from RUN to STOP/PGR

If a job is currently being processed it will be completed, i.e. processed without an error message.

Depending on the number of pieces of data to be output, it may take some time before the CP 525 goes over to the STOP/PGR state. This transition is indicated by the LEDs on the front panel lighting up.

Switching the selector from STOP/PGR to RUN

The interpreter once again starts to process jobs. The LED on the front panel goes off (if the interpreter and procedure are loaded). Changing the selector setting does **not** trigger a cold restart on the CP 525.

Exception

If a part of the interface or the whole interface has been reloaded (transferred from the PG) the interface must be cold restarted. In this case, change the selector setting from "STOP/PGR" to "RUN", to perform a cold restart, or you can press the "cold restart" key on the PG. The mode selector on the CP must be set to "RUN".

12.4.4 Transferring from the PG

If the transfer of the procedure, interpreter, or PC jobs is triggered at the PG, the software on the CP 525 goes into a wait state. In the wait state the LED of the interface affected lights up on the front panel.

The transfer is started only when the program to be transferred differs from the program already loaded in the CP 525. For this reason the PG first reads the program in the CP and compares it byte for byte with the program to be transferred. The transfer can take place regardless of the setting of the mode selector.

Before the CP 525 software goes into the wait status (indicated by the lighting up of the LEDs on the front panel), the job currently being processed is completed without an error message. Depending on the number of pieces of data to be output, this may take some time.

With the software in the wait state no PC jobs can be processed - "overload" will be indicated. This can be recognized by error message number 91H in the PAFE byte of the corresponding job. In rare cases it can happen that a PC job will be "terminated with error" before the PAFE error message (no. 6H in the condition codeword and 11H in SYSTAT). In this case, the PC was unable to recognize the "overload" early enough.

Following the transfer, the software remains in the wait state until you cold restart the interface - either with the "cold restart" key on the PG or by using the mode selector on the CP (switching from STOP/PGR to RUN).

Following the cold restart, the LEDs on the front panel go off if the program transferred is correct and complete. A SYNCHRON job is no longer required after the transfer.

13 LAUFPT88 Procedure

Section 13 contains information about the procedure LAUFPT88 as follows:

- procedure data
- transmitting with the LAUFPT88 procedure
- receiving with the LAUFPT88 procedure

13.1 Procedure Data

The LAUFPT88 procedure controls the data flow between your CP 525 and the PT88/PT89 printer. Within the CP 525, the interpreter transfers the data to be transmitted to the procedure into output buffers. The procedure then has the data output on the printer.

If the printer sends characters to the CP 525, these are handled by the LAUFPT88 procedure as follows:

- they are recognised as control characters and processed, or
- the characters cannot be recognized and the procedure signals a non-recoverable error to the interpreter which passes this on to the CPU. An error number is entered in the condition code-word or in the error message area of the SYSTAT (see Section 17 "Error handling").

The LAUFPT88 procedure is an asynchronous bit serial transmission procedure. The transmit and receive data rates (baud rates) on the CP 525 and on the connected printer must be the same (i.e., there are no timing lines between the two devices).

Both **control and information characters** are transmitted on the data lines. The control characters used, **XOFF** (13H) and **XON** (11H), are described in Section 13.2.2 "Mode of control with the LAUFPT88 procedure".

To check that the transfer takes place without errors, the information bits are "encapsulated" by further bits.

The order of the bits are as follows:

SA	I0	I1	I2	I3	I4	I5	I6	PA	SO

SA = start bit
I* = information bit no.
PA = parity bit
SO = stop bit

The following parameters are **fixed** and cannot be changed with COM 525:

- character length: 7 bits (fixed)
- priority: low (fixed)

The following procedure parameters can be adjusted with COM 525:

- number of stop bits:

1 stop bit
1.5 stop bits
2 stop bits

- Data rate

The speed of the data transmission is specified in bps.

You can set the following values:

9600 bps
4800 bps
2400 bps
1200 bps
600 bps
300 bps
200 bps
110 bps

- **Parity**

Both odd and even parity are possible.

Refer to the instructions for the CP 525 in this manual and the user's guide for the PT88/PT89 printer.

13.2 Transmitting with the LAUFPT88 Procedure

13.2.1 Data Transfer

If the interpreter transfers a SEND job to the LAUFPT88 procedure, the procedure goes into the transmit mode; i.e., it transmits the data in the current output buffer to the printer (if the printer is ready to receive). Once the procedure has output the entire contents of the buffer, it terminates the transmit mode and waits for new jobs.

13.2.2 Mode of Control of the LAUFPT88 Procedure

To guarantee the correct transfer of data between the CP 525 and the PT88/PT89 printer, the following mode of control is used:

- control with the character XOFF (13H) and XON (11H)
  "XOFF" means: printer not ready to receive
  "XON" means: printer ready to receive

The PT88/PT89 sends these control characters to the CP 525 to signal its current status to the CP. These characters are transferred by the printer regardless of the interrface used (V.24/V.28 or TTY).

Controlling data transmission with the XOFF/XON protocol:

The LAUFPT88 procedure can adopt the following statuses:

- idle state
- transmit mode

In both statuses the procedure is capable of receiving XOFF or XON.

If the LAUFPT88 procedure receives either of the control characters while in the idle state, it records the status of the printer internally. If a data transmission is then to be started (procedure is to change from idle state to transmit mode), it can take place only if the printer is in the XON state, i.e., if

- the printer has paper and ink
- it is switched online

If the LAUFPT88 procedure receives the control character XOFF while in the transmit mode, the transmit mode is interrupted and a maximum of two further characters output. At the same time, a waiting time of one minute is started. If the procedure receives the control character XON during this time, it continues the output. If it does not receive the character within this time it signals the interpreter that the waiting time has elapsed and that the printer is not ready to receive. An error number is entered in the condition codeword or in the error message area of SYSTAT (see Section 17 "Error handling"). This situation could have the following causes:

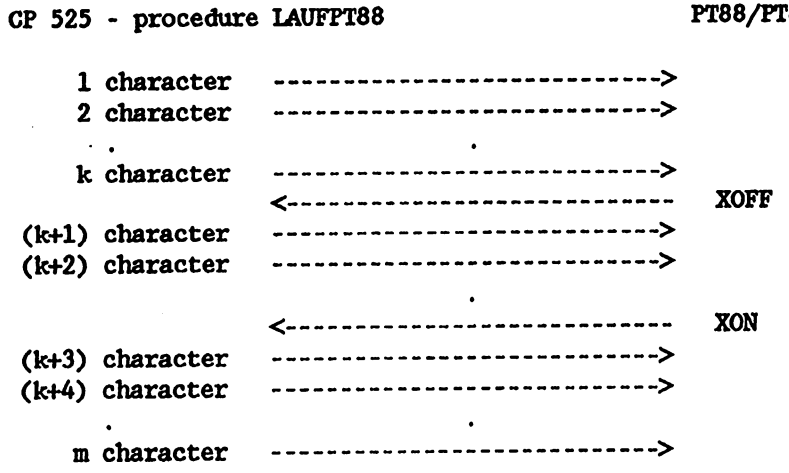
- no more paper
- no more ink
- printer is switched offline

If the printer is once again ready to receive (XON is received), the data transfer is continued.

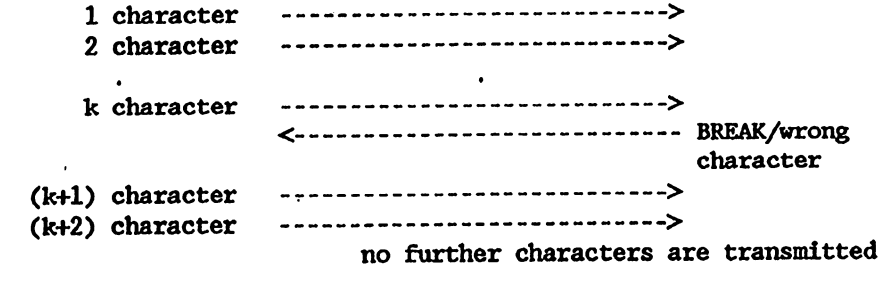
13.3 Receiving with the LAUFPT88 Procedure

Only the control characters XON and XOFF are received and processed by the procedure. Other characters result in error messages being sent to the interpreter (check the condition codeword or error message area of the SYSTAT). A break on the line is also signalled as an error. Following these error messages, all the jobs signalled to the procedure are aborted and the procedure goes over to the wait state.

Example of transmitting m characters **without** errors:



Example of transmitting m characters **with error**:



8

14 Multiprocessor Operation

Compared with single processor operation, multiprocessor operation requires even more that the programming of the individual CPUs and of the CP 525 be carefully matched.

The more complex the application, the greater the care that must be taken to ensure that the CPU numbers and data block numbers for the dynamic values of lists and frames in the CP 525 are correctly specified.

!IMPORTANT!

If several CPUs are to supply jobs to only one interface of the CP 525, the STEP 5 user programs of these CPUs must be coordinated.

If jobs with the same job number are programmed on more than one CPU (e.g. SMLs), the STEP 5 user programs must be designed so that at any one time only one CPU can start one of these jobs. All other CPUs must wait until this job is signalled as "completed" with or without errors. Only then can this job be started by another CPU.

The jobs (SML, CML, PSL, new page, etc.) can be divided into the following two groups:

- jobs programmed only once on the CP 525
- jobs programmed more than once on the CP 525

Recommendations for multiprocessor operation are arranged according to these groups.

For more information on the individual jobs see the corresponding sections in this user's guide.

14.1 Jobs Programmed Only Once on the CP 525

The following jobs belong to this group:

- sequential message list (SML)
- current message list (CML)
- new page
- reset page number
- update PSL

The common factor among these jobs is that each job has a fixed name and can therefore be programmed only once on a CP 525 interface.

SML and CML differ from other jobs in that data must be transferred from the CPU to the CP 525 before it is executed.

14.1.1 Jobs without Data Transport

When using the jobs "new page, reset page number and update SML" remember that the same job number is used for the same job on each CPU.

14.1.2 Jobs with Data Transport

With the SML and CML jobs, remember that you must use the same job number for the same job on each CPU.

The following points should also be remembered.

Once the CPU has triggered a job, it makes a data block available to the CP 525 that contains the dynamic values for the message. With an SML, this is the DB M RECORD. With CML, it is the DB M OLDNEW, which images the disturbances in the plant. The CPU informs the CP 525 by means of the SML or CML job that the data block can be fetched.

If the CP 525 receives this job, it determines which CPU sent the job and requests the data block from this CPU. As soon as it receives the data block, the CP has the corresponding message texts output on the PT88/PT89.

Any CPU can have the messages output via any CP 525 if it has message texts programmed on the CP 525.

Note:

- On one CP 525, one message number is assigned to one message text. This is output regardless of which CPU initiated the job.
- When programming the messages with COM 525, do not specify a CPU number.
- Make sure that the message numbers are distributed efficiently on the individual CPUs.

Depending on the application the following procedure is recommended:

- **Using several CPU's with one CP 525 in one PC**

- Select a CPU on which the messages from all the CPUs in the PC can be collected.
- Have all the messages prepared on this CPU and passed on to the CP 525.

This process avoids having the same message number assigned on more than one CPU.

- **Using one CPU with several CP 525's in one PC**

If you wish to output various message texts via the individual CPs proceed as follows:

- program the CPs so that you store only one message text for each message number **on only one CP 525** (although all the message numbers appear on all the CPs)
- implement your STEP 5 user program on the CPU so that the data blocks (DB M RECORD or DB M OLDNEW) are sent to all CP 525s.

Since the CP 525 ignores all messages for which it does not have a message text programmed, the required message text is output only via one CP 525, if the CP has a printer connected.

If you want to output the same message texts via several CP 525s, you must allocate the same message text to the same message numbers.

- **Using several CPU's and several CP 525s in one PC**

See recommendations for the following:

- using several CPUs with one CP 525 in one PC
- using one CPU with several CP 525s in one PC

14.2 Jobs which Can Be Programmed More Than Once on the CP 525, Including Frames

This group includes the following:

- process status list (PSL)
- chained list (CL)
- update group inhibit bits (update GIB)

- frame

When programming the CP 525 with COM 525 you determine for this group of jobs

- which data block is to be fetched
- from which CPU

(for the output of the dynamic values or for updating the group inhibit bits).

As soon as **any** CPU has sent a job to the CP 525, the CP 525 requests the **programmed** data block from the CPU, which has been specified for this job in the CP 525 user program. The specified CPU does not need to be the one which initiated the job.

It is advisable, to select the same CPU for the start of a job and for the supply of data. This avoids

- attempting to fetch data for a job from a non-existent CPU or
- the data block fetched from another CPU not matching the job initiated.

15 Memory Assignment, Memory Space Requirements

The CP 525 requires the following:

- **loadable software (for the specific component)**
- **user programs for specific tasks**

Loadable software and user programs for specific tasks are stored on the plug-in memory submodule (RAM or EPROM). The amount of memory space required depends on the particular component and the extent of your user programs.

The following sections provide you with the following:

- an overview of the space requirements (graphic representation)
- a detailed description with tables
- examples you can use to calculate your own memory space requirements

!IMPORTANT!

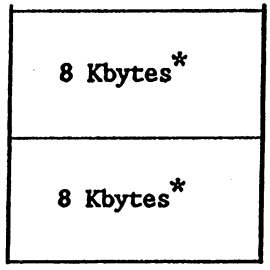
The information can only take the form of guidelines with which you can estimate the memory space you require. The figures ought to help you determine the minimum capacity your memory submodule needs to handle your requirements. The figures quoted in the following pages are arranged so that the memory requirements calculated tend towards the maximum requirement.

Using the COM 525 information function you can obtain detailed information about programs which have already been generated (see user's guide "Programming package COM 525" in this manual) or using special PROM 525 functions (see user's guide "PROM 525").

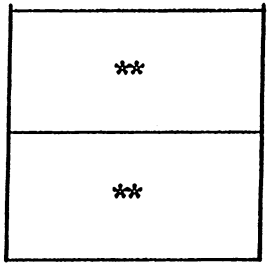
Overview

The following is an overview of the memory space required on the memory submodule.

Loadable software:



User programs for specific tasks:



* always assigned, even if the interface is not programmed
** only assigned if this interface is programmed.

The following also applies:

- if the **computer link with RK 512** is programmed on the interface, **2 Kbytes** of memory are required

- if the **printer PT88/PT89** is programmed on the interface, you will require the following memory space for the directory and elements:

more exact information about the sub-areas can be found in the sections with the graphic representations:

Space required for the directory

1370 bytes see 15.2.1	basic requirements for the directory
5 x m see 15.2.1.1	m = number of messages programmed
32 bytes 32 bytes see 15.2.1.2	if frames are programmed if PC jobs are programmed
19 x k see 15.2.1.2	k = number of programmed elements, e.g. frame 1, PSL5, PSL21, trailer 5

Space required by the programmed elements

see 15.2.2.1	space required for messages
see 15.2.2.2	space required for frames
see 15.2.2.3	space required for PC jobs

15.1 Loadable Software

You can use the CP 525 for the following:

- the computer link with RK 512
- event output and listing on the PT88/PT89 printer

For each of these components there is special software. To avoid putting unnecessary load on the limited amount of memory capacity of the CP 525, this software is not usually stored on the CP 525. The software you require for your particular application is loaded in plug-in memory submodules (RAM or EPROM).

The loadable software both for the computer link and for the printer PT88/PT89 includes the following:

- procedure implements the data exchanges with the connected partner (computer, other PC, PT88 printer ...) and
- interpreter processes the user programs that are programmed with COM 525

15.1.1 Memory Requirements of the Loadable Software

The loadable software always occupies **2 x 8 Kbytes** of memory on the plug-in memory submodule of the CP 525. This means that 8 Kbytes are occupied per interface regardless of whether the CP 525 is being operated on one or both interfaces, and also regardless of whether the component computer link and/or printer PT88/PT89 is being used.

15.2 User Programs for Specific Tasks

The loadable software for the printer component alone does not allow the CP 525 to execute the printing function. It does not yet know what it should list for your special application,

- messages or process status list
- which static texts go with which dynamic values
- from which CPU and from which DB it will obtain the data ...

This is determined when you program your CP 525 user program with the COM 525 programming package. After programming, you can use the PG to load the user program into the memory submodule of the CP 525 (RAM or EPROM).

15.2.1 Directory (Memory Management)

The printer PT88/PT89 component includes the user program made up of individual program elements (from now on called simply **elements**). These are

- **messages**
- **frames**
- **PC jobs**

To enable the CP 525 as well as COM 525 to access these elements quickly, they are stored "systematically" with their name, address, length, type etc. All these specifications are contained in an "address book" called **directory** from now on. The directory also occupies space in the memory submodule as follows:

- basic requirement
- subdirectory for messages
- subdirectory of the PC jobs and frames

Memory required (in bytes)	For
1370	basic requirements for the directory of an interface
5 x m	subdirectory of messages: m = number of messages programmed
32 x i	subdirectory of PC jobs and frames: i = number of programmed elements, e.g. - PSL, SML - frame
19 x k	k = number of programmed elements, e.g. - frame 1, header 25, trailer PSL - chain 2, rolling mill, PRESS 1, BELT 1, - updateGIB15, GIB3b

15.2.1.1 Subdirectory of the Messages

The subdirectory of the messages is designed for the following:

Optimum use of memory
With up to 1000 messages per interface, it is extremely important that no byte is wasted, although the messages can have different lengths (1 - 132 characters).

Short access times
The messages must be found quickly and output on the printer.

!IMPORTANT!: messages are not identified by names or job numbers but rather by the message numbers.

The subdirectory of the messages requires **over and above the basic requirements of approximately 1370 bytes**, the following space in the memory:

Space required (in bytes)	For
5 x m	m = number of programmed messages
	totalled up therefore:
0	if no message is programmed
5	if one single message is programmed
5000	if 1000 messages are programmed (max.)

15.2.1.2 Directory of PC Jobs and Frames

The subdirectory of PC jobs and frames is designed to accommodate the great variety of sizes of the elements: e.g.,
- a PSL can be a maximum 99 lines long
- the PSL line can contain up to 40 variables
- the PSL line can be up to 132 characters long

A PSL can be 100 characters long or 40000 characters long.

Note:

PC jobs are identified in two ways as follows:
- by their names, when they are processed using COM 525 (when they are first specified, modified or deleted)
- by their job numbers when they are processed on the CP 525. (Calls by the CPU use only the job number).

The directory of PC jobs and frames must contain all this information and consequently differs from the directory of messages.

The memory requirements for the subdirectory of the PC jobs and frames can be calculated as follows:

Memory required (in bytes)	For
32 x i	i = number of programmed element types, e.g., - PSL, SML, update GIB - frames
19 x k	k = number of programmed elements, e.g., - frame 1, header 25, trailer PSL - chain 2, rolling mill, PRESS 1, BELT 1, - updateGIB15, GIB3b

Example:

The following is programmed:

Number	Element types	Different element types
1 each	SML, UPDATE SML, CML	–
1 each	NEW PAGE, PAGE NO. = 1	1
9	PSL's	
3	update GIB	–
5	frames	1
22	elements	2 types

and:

150	messages	-

The directory for this requires the following space in the memory submodule:

Memory required (in bytes)	For
1370	basic requirements for the directory of this interface
750	subdirectory of messages: = 5 x 150 for 150 programmed messages
64	subdirectory for PC jobs and frames: = 2 x 32 for 2 different element types
418	= 22 x 19 for 22 programmed elements
2602	total requirements for the directory

8

!IMPORTANT!

Note on programming the two interfaces of the CP 525 with printer PT88/PT89 is as follows:

The user memory of the CP 525 is divided into memory pages, each with 8 Kbytes. The directory of the user program is established on one memory page and the elements are continuously written to the other memory pages.

The printer on one interface can use only memory pages on which the printer of the other interface has not yet written data. For example, if printer 1 has used 1 Kbyte of a memory page, printer 2 cannot use the 7 Kbytes remaining on this page; i.e., these 7 Kbytes remain unused.

When you calculate the memory requirements the calculated program lengths must be rounded up to whole multiples of 8 Kbytes.

15.2.2 Element Types and Elements

The CP 525 user program contains everything the CP 525 requires to carry out the tasks selected by the user. For the printer PT88/PT89 component the user program includes the following individual element types:

- **messages**
- **PC jobs** (PSL, SML, update GIB ...)
- **frames**

The element types are further divided into single elements. The structure of the individual **elements** is similar. They always include the following:

- one **element header**
- one or more **commands**

Element header

This header contains all the central data and commands for the element as follows:
- control information
- name of the list header and/or list trailer
- CPU no., DB no. (source address for the dynamic data)
- etc.

The memory required for the element header varies depending on the element type.

Commands

Commands include the **command code** and **parameters**.

Command code

The command code instructs what is to be done with the parameters which follow.

Examples:

command: PRINT "today is Monday"
therefore:
command code = PRINT (stored as code)
parameter: today is Monday"

command: SET ATTRIBUTES TO: "bold on", "acoustic signal on"
therefore:
command code = SET ATTRIBUTES TO (stored as code)
parameter: "bold on" and "acoustic signal on" (stored as code)

Parameters

Parameters can be the following:
- texts
- CPU number
- DB number
- address, e.g. DR 123, DW 15
- name (e.g. of frames) etc.

This brief introduction illustrates that not only the texts to be output require space in the memory, but also that large amounts of additional information (commands, parameters) are also required before a message can be printed out as you require.

In addition, another directory is required to ensure that the elements can be found quickly.

The more dynamic data you bring into a text, the more often you change the print attributes; the more elements (PSLs, frames, messages) you program, the more memory space will be required.

The following sections show you how much space the various elements require and how much space the corresponding directory takes up. The information must be considered as a guideline so that you can estimate the total memory space required. The guidelines help you to decide whether the memory submodule installed is sufficient for your requirements. The calculated requirements always tend towards maximum values. Exact formulae are extremely complex and impossible to use because of the large number of combinations that can be programmed.

15.2.2.1 Messages

A maximum of 1000 messages can be programmed per CP 525 interface. Messages are not identified by names or job numbers but rather by message numbers. There is a separate directory for messages. The memory requirements for the directory depend on the number of messages programmed.

Memory required for one message

The memory required for one message consists of the following:

Memory required (in bytes)	For	
10	basic requirement (element header)	
n	n characters static text	
3	date	*
3	time	*
3	status	*
8	dynamic field (irrespective of type of variables!)	*
27 + n	maximum memory required	
10 + n	minimum memory required	

* = optional, depending on programming

Note: to save space, the static texts are automatically compressed by COM 525; i.e., more than three identical consecutive characters occupy only three bytes of memory!

Example

The string of numbers does not form part of the example, the dynamic parts are printed in bold face.

```
0        1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789012345

27.02.87 time 13.52.31 coming conv.belt 3a state STOP
```

Memory required (in bytes)	For
10	basic requirements (element header)
37	37 characters of static text (incl. spaces!)
3	date
3	time
3	status
8	dynamic field
64	total requirement for this message

Suppose that 50 messages are programmed with the same structure.

Then the total memory requirement is as follows:

Memory required (in bytes)	For
3200	total memory requirements for the 50 messages with the structure above (64 bytes per message) = 50 x 64 bytes

In addition, there is the memory required for the directory. For this example the memory is as follows:

Memory required (in bytes)	For
250	= 5 x 50 for 50 programmed messages

The total requirement for these 50 messages + directory is therefore
3450 bytes = 3200 + 250 bytes

15.2.2.2 Frames (Headers, Trailers)

The memory required for a frame includes the following

Memory required (in bytes)	For
27 sum L	basic requirement (element header) sum of the memory required for the individual frame lines (L)

In addition, there is also the memory required for the directory (as explained in 15.2.1).

Memory requirements for each frame line (L)

The memory required for each individual frame line is calculated as follows:

Memory required (in bytes)	For
5	basic requirement per line if the line is not assigned to a group
16	basic requirement per line if the line is assigned to a group
n	n characters static text
sum D	sum of the memory requirement for all the dynamic variables of a frame line

Note:

To save space the static texts are automatically compressed by COM 525; i.e., more than three identical successive characters occupy only three bytes of memory!

Memory requirements for the dynamic variables of a frame line

The memory requirements for the dynamic variables in a frame line vary depending on the type of variable as follows:

Memory required (in bytes)	For
17	basic requirement
13	date
13	time
18	process variable
	process status variable:
6 + n1	n1 = no. of char. in text for process status 1
6 + n2	n2 = no. of char. in text for process status 2
.	
6 + n8	n8 = no. of char. in text for process status 8
	space is only required for programmed process statuses!

Note:

This is always the maximum space required; taking into account the following:
- date always DMY and source "DB"
- time always HMS and source "DB"
- changes of attribute
- address in DB greater than 99 (3 digits)

By reducing any of these you can save space.

Example

The string of numbers does not belong to the example.

```
        0         1         2         3         4         5
        12345678901234567890123456789012345678901234567890123456
Line 1  +--------------------------------------------------------+
Line 2  ! Conveyor belt 03                                       !
Line 3  ! Shift:   ###  Shift engineer: ############            !
Line 4  ! "Date": ########  "Time": ########                     !
Line 5  +--------------------------------------------------------+
```

The space required for this frame is the total of the following:

Memory required (in bytes)	For
27	basic requirement (element header)
10	line 1
28	line 2
74	line 3
55	line 4
10	line 5
204	sum for the entire frame

In addition, there is also the space required for the directory (according to 15.2.1).

The memory requirements of the individual parts of the frame are calculated as follows:

Line 1:

```
+-------------------------------------------------+
```

Memory required (in bytes)	For
5	basic requirement
5	static text 1 byte for "+" (start of line) 1 byte for "+" (end of line) 3 bytes for "-" (compressed)
0	line 1 contains no dynamic parts
10	total for line 1

Line 2:

! Conveyor belt 03 !

Memory required (in bytes)	For
5	basic requirement
23	static text 18 bytes for "! conveyor belt 03" 1 byte for "!" (at end of line) 3 bytes for spaces (compressed)
0	line 2 contains no dynamic parts
28	total for line 2

Line 3:

! Shift: ### Shift engineer: ############ !

Memory required (in bytes)	For
5	basic requirement
33	static text 11 bytes for "! Shift: " 18 bytes for " Shift engineer: " 1 byte for "!" (at the end) 3 bytes for spaces (compressed)
36	for the two dynamic field 18 bytes for field 1 (process variable BYTE) 18 bytes for field 2 (process variable CHAR)
74	total for line 2

Line 4:

! "Date": ######## "Time": ######## !

Memory required (in bytes)	For
5	basic requirement
24	static text 10 bytes for "! "date": " 10 bytes for " "time": " 1 byte for "!" (at end of line) 3 bytes for spaces (compressed)
26	for the two dynamic fields 13 bytes for field 1 (DATE from DB) 13 bytes for field 2 (TIME from DB)
55	total for line 3

8

Line 5:

Memory required (in bytes)	For
5	basic requirement
5	static text 1 byte for "+" (at start of line) 1 byte for "+" (at end of line) 3 bytes for "-" (compressed)
0	line 1 contains no dynamic parts
10	total for line 5 (identical to line 1)

15.2.2.3 PC Jobs

PC jobs are started by the CPU using their job numbers. The following are PC jobs:

1. Sequential message list SML
2. Process status list PSL
3. Chained list CL
4. Current message list CML
5. Update SML
6. Update group inhibit bits (GIB)
7. New page
8. Reset page number

1. Sequential message list SML

The memory space required for the sequential message list (SML) is obtained by the following:

Memory required (in bytes)	For
18	basic requirement (element header)
2 x (i + 1)	list header (i = no. of chars. in frame name) *
k + 1	list trailer (k = no. of chars. in frame name)*
12	column no., attributes, order "date" *
12	column no., attributes, order "time" *
7	column no., attributes and no. of characters in the texts for 3 statuses (attributes) of the messages:
3 + l	l = no. of char. in text for status "coming" *
3 + m	m = no. of char. in text for status "going" *
3 + n	n = no. of char. in text for status "acknowledged" *

* = optional, depends on programming

The memory requirements for the directory must be added to this (as in 15.2.1).

2. Process status list PSL

The memory requirements for a PSL are obtained by the following:

Memory required (in bytes)	For
26	basic requirement (element header)
2 x (i + 1)	list header (i = no. of chars. in frame name) *
2 x (k + 1)	list trailer (k = no. of chars. in frame name) *
Sum L	sum of the memory required for the individual PSL lines (L) (max. 99 lines per PSL possible) this is calculated using the same formula as for individual frame lines (see 15.2.2.2)

The memory requirement for the directory must be added to these requirements (as in 15.2.1).

Memory requirements for the individual PSL lines (L)

The memory requirements for the individual PSL lines can be calculated according to the same table that the individual frame lines use:

Memory required (in bytes)	For
5	basic requirement per line if the line is **not** assigned to a group
16	basic requirement per line if the line is assigned to a group
n	n characters static text
sum D	sum of the memory requirements for all the dynamic variables of a frame line

Note:

To save space the static texts are automatically compressed by COM 525; i.e., more than three identical successive characters occupy only three bytes of memory!

Memory requirements for the dynamic variables in a PSL line

Memory requirements for the dynamic variables in a PSL line are different depending on the type of variable. You can calculate the requirements by using the following table:

Memory required (in bytes)	For
17	basic requirement
13	date or time
18	process variable
	process status variable:
6 + n1	n1 = no. of chars. in text for process status 1
6 + n2	n2 = no. of chars. in text for process status 2
.	
6 + n8	n8 = no. of chars. in text for process status 8
	space is required only for process statuses that have been programmed!

Note:

These memory requirements are always the maximum space required; they take into account the following:
- date always DMY and source "DB"
- time always HMS and source "DB"
- changes of attribute
- address in DB greater than 99 (3 digits)

By reducing any of these you can save space.

Example

Since there is very little difference between PSLs and the frame in terms of structure you can simply use the example given for calculating the memory requirements for the frame.

Included in	PSL	Frame
list header	yes	no
list trailer	yes	no
job number	yes	no
DB number	yes	yes

Assign the following to the PSL:

```
list header:  the frame  "HEAD1"
list trailer: the frame  "TRAI1"
job number:              "135"
```

Example (the string of numbers does not belong to the example)

```
0        1         2         3         4         5
12345678901234567890123456789012345678901234567890123456
+--------------------------------------------------------+
! Conveyor belt 03                                       !
! Shift:   ###  Shift engineer: ############            !
! "Date": ########  "Time": ########                     !
+--------------------------------------------------------+
```

The space required for this PSL is the total of the following:

Memory required (in bytes)	For
26	basic requirement (element header)
11	list header (2 x 5 = 2 x no. of chars. "HEAD1")
11	list trailer (2 x 5 = 2 x no. of chars. "TRAI1")
10	line 1
28	line 2
74	line 3
55	line 4
10	line 5
225	sum for the entire PSL

For calculating the memory requirements of the individual PSL lines see the example of the frame (Section 15.2.2.2).

3. Chained list (CL)

The memory requirements for the chained list are obtained from the following:

Memory required (in bytes)	For
8	basic requirement (element header)
1 + n1	n1 = no. of char. for name of 1st chained PSL
1 + n2	n2 = no. of char. for name of 2nd chained PSL
.	.
1 + n8	n8 = no. of char. for name of 8th chained PSL
	if there are fewer than 8 PSLs chained, the memory requirements are reduced accordingly e.g. with 2 PSLs only (1 + n1) + (1 + n2)

The memory requirement for the directory must be added to this (as in 15.2.1).

4. Current message list CML

The memory requirements for the current message list are obtained from the following:

Memory required (in bytes)	For	
20	basic requirement (element header)	
2 x i	list header (i = no. of char. in frame name)	*
2 x k	list trailer (k = no. of char. in frame name)	*

* = optional, depending on programming

The memory requirement for the directory must be added to this (as in 15.2.1).

5. UPDATE SML

The memory requirement for the job "UPDATE SML" is the following:

Memory required (in bytes)	For
12	total requirement

The memory requirement for the directory must be added to this (as in 15.2.1).

6. UPDATE GROUP INHIBIT BITS (GIB)

The memory requirement for the job "UPDATE GIB" is as follows:

Memory required (in bytes).	For
24	total requirement

The memory requirement for the directory must be added to this (as in 15.2.1).

7. NEW PAGE

The memory requirement for the job "NEW PAGE" is as follows:

Memory required (in bytes)	For
12	total requirement

The memory requirement for the directory must be added to this (as in 15.2.1).

8. RESET PAGE NUMBER

Suppose that PAGE NUMBER = 1

The memory requirement for the job "PAGE NUMBER = 1" is as follows:

Memory required (in bytes)	For
12	total requirement

The memory requirement for the directory must be added to this (as in 15.2.1).

Empty page

16 Special Jobs Carried Out by the CPU

Section 16 contains information about the following:

- the error message area of the SYSTAT
- the identification area SYSID
- the reading and writing of the date and time

16.1 Reading the Error Message Area of the SYSTAT

The error message area of the SYSTAT is a data area in the dual-port RAM of the CP 525. After the CP 525 recognizes an error, it enters the corresponding error number in the error message area of the SYSTAT. You decide how the CPU is to react in your STEP 5 user program on the CPU.

The CPU reads the **error message area** of the SYSTAT for the specified CP interface by means of the job

RECEIVE DIRECT 200

The area in the CPU (destination area) to which the error message area is to be transferred, is fixed when you assign parameters to the RECEIVE DIRECT 200 data block. The destination length must be at least two words.

RECEIVE DIRECT 200 is executed only when an error number has been entered in the error message area of the SYSTAT, since only then is the bit "RECEIVE - job ready" set in the ANZW (bit 0).

Note: in some cases (e.g., if no SYNCHRON has been run or if start-up errors occurred in the CP 525) the error message area cannot be read by RECEIVE DIRECT 200 (cf. note in 16.3).

Structure of the destination area

If error numbers are entered in the error message area they are located (after they have been read by RECEIVE DIRECT 200) in the selected area in the CPU (e.g. in the data block).

		Bit 7	6	5	4	3	2	1	0
Byte	0 (DL0)	-				E	0	B	R
	1 (DR0)	Error number 1							
	2 (DL1)	Error number 2							
	3 (DR1)	Error number 3							

This diagram is explained on the following page.

Byte	Bit	Status	Significance
byte 0	bit 3 (E)	0	no error in SYSTAT (not following executed RECEIVE DIRECT 200)
		1	error entered in SYSTAT note: this bit is always set if there is an error entry - i.e. with this you can check whether data were transferred with RECEIVE DIRECT 200
	bit 2 (O)	0	no error overflow (max. 3 error entries)
		1	error overflow (more than 3 errors)
	bit 1 (B)	0	no BREAK currently on the interface
		1	BREAK on the interface note: bit 1 only shows the instantaneous status and does not always mean an error. The error status BREAK is determined only by the error number in DR0, DL1 and DR1. The current BREAK status can only be indicated here if the interpreter and procedure are loaded and have been started!!
	bit 0 (R)	0	printer ready to receive
		1	printer not ready to receive longer than 1 minute (in XOFF state) possible causes: - no paper - printer offline - no ink
	bit 7, 6, 5, 4		irrelevant
byte 1			error number 1
byte 2			error number 2
byte 3			error number 3 00 = no error

For more information on error messages please see Section 17.

Example of a command sequence within a STEP 5 user program for the CPU 992 (R processor) is as follows:

```
     : .
     :O   F 1.0        ; to initiate the reading out from SYSTAT
     :ON  F 1.0        ; result of logic operation must = "1"
     :
     :JU  FB121
     NAME:RECEIVE      ; data transfer only in error entry
     SSNR: KY0,0       ; interface number 0
     A-NR: KY0,200     ; fixed job number for reading SYSTAT
     ANZW: FW10
     ZTYP: KSDB
     DBNR: KY0,10      ; dest. for error numbers is DB10
     ZANF: KF+1        ; from data word 1
     ZLAE: KF+2        ; 2 data words
     PAFE: FY5
     :
     :C   DB10
     :AN  D 1.11       ; test bit 3 of group byte
     :JC  =M001        ; jump if no error entered
     :.
     :.                ; error evaluation
     :.
     :.
     : JU FB124        ; an error was entered, i.e. the SYSTAT
     NAME:RESET        ; register on the CP 525 should be reset
     SSNR: KY0,0       ; with RESET DIRECT 200 (see following
     A-NR: KY0,200       pages)
     PAFE: FY6
     :
     :C   DB10         ; reset error entry in DB
     :L   KH0000
     :T   DW1
     :T   DW2
     :
M001:.
     :etc.
     :BE
```

8

16.2 Resetting (Clearing) the Error Message Area

The CPU resets the error message area of the SYSTAT (according to interface numbers) with the job

RESET DIRECT 200

All the entries are then cleared with the exception of the BREAK bit (B).

!IMPORTANT!: the error message area of the SYSTAT is not cleared by a SYNCHRON call.

The CP 525 can enter up to three error numbers in the error message area. Further errors cannot be signalled, unless the "old" entries have been cleared. For this reason the error message area must be reset in good time by a RESET DIRECT 200.

The destination area in the CPU must also be reset.

16.3 Reading the Whole SYSTAT

In addition to reading the error message area of the SYSTAT, the whole SYSTAT of the CP can be read by means of the job

RECEIVE DIRECT 221

The area read then contains the error messages for both interfaces (in bytes 4 to 10). This job can only be executed if there are no error numbers entered. A destination length greater than or equal to 16 bytes (8 words) is necessary.

Note: in some cases, (e.g., no SYNCHRON executed or error during the start-up of the CP 525), the SYSTAT cannot be read by means of RECEIVE DIRECT 221. It must then be read directly from the dual-port RAM (from address F660H).

The area transferred with this job is structured as follows:

	Bit 7	6	5	4	3	2	1	0
Byte 0	irrelevant						RUN	STOP
1	irrelevant							
2	sign of life							
3	irrelevant							PGR
4	E2	O2	B2	R2	E1	O1	B1	R1
5	error number 1 IF1							
6	error number 2 IF1							
7	error number 3 IF1							
8	error number 1 IF2							
9	error number 2 IF2							
10	error number 3 IF2							
11 12 13 14 15	reserved							

The individual bits or bytes have the following significance:

RUN = 1 mode selector switched to RUN
STOP = 1 mode selector switched to STOP
PGR = 1 mode selector switched to PGR
Sign of life: this byte is incremented by 8 at regular intervals on the CP

Bytes 4 to 10 cover the error message area for both interfaces (see Section 16.1 - Reading the error message area of the SYSTAT).

Error message area for IF1: bytes 5 to 7 and byte 4 (bits 0 to 3)
Error message area for IF2: bytes 8 to 10 and byte 4 (bits 4 to 7)

16.4 Reading the SYSID

An identification area is specified on each CP 525. Information is written into the identification area SYSID of the CP 525; i.e., you can find information about the following:

- the module
- the memory submodule
- the interface assignment etc.

By means of the job

RECEIVE DIRECT 223

the CPU can read the SYSID of the CP 525.

This is possible providing the field length is greater than or equal to 128 bytes; i.e., the BLGR parameter in the SYNCHRON is KY 0,4 / KY 0,5 / KY 0,6.

A destination length greater than or equal to 128 bytes is also necessary.

If the field length is less than 128 bytes the message "completed with errors" and the error number 7 is entered in the status byte and the number 1E in the error message area of the SYSTAT. The SYSID area is not then transferred.

Note: the SYSID can also be read directly from the dual-port RAM; i.e., without handling blocks (from address F680H). For more information refer to the manual for your PC.

Structure of the identification area SYSID

The SYSID contains a number of parameters that identify the CP. Each parameter is completed with carriage return (CR = ODH) - also counted in the byte specification. Parameters not assigned consist simply of CR. Some parameters have a different significance depending on the memory submodule types. All the parameters are represented in ASCII.

With the CP 525 these parameters are structured as follows:

Parameter	Byte	Description
0	0 to 9	Mem. submodule identifier/capacity: EPROM: e.g. for order number 6ES5373-0AA41 shown as:373-0AA41 RAM: memory capacity of the submod. RAM xxxKw e.g. xxx = 64
1	10 to 18	Module identifier: CP 525-2 / CP 524
2	19 to 21	Version of the CP firmware
3	22 to 41	Plant designation: EPROM: designation of programmed plant RAM: replaced by space (20H)
4	42 to 50	Generated on: EPROM: date of EPROM programming RAM: replaced by space (20H)
5 to 9	51 to 55	not used with CP 525
10	56 to 64	PROM 525 - version: PROM 525 = programming program for EPROM submodules EPROM: PROM 525 - version RAM: replaced by space (20H)

Interface 1

Parameter	Byte	Description	Example
11	65 to 67	component type	PT
12	68 to 76	interpreter name (*)	PT88
13	77 to 79	interpreter version	nn
14	80 to 88	procedure name (*)	LAUFPT88
15	89 to 91	procedure version	nn

Interface 2

Parameter	Byte	Description	Example
16	92 to 94	component type	RK
17	95 to 103	interpreter name (*)	RK512
18	104 to 106	interpreter version	nn
19	107 to 115	procedure name (*)	P3964R
20	116 to 118	procedure version	nn
	119	end identifier ETX (03H)	

nn = version number
* = name of interpreter and procedure correspond to the designations in the COM library

If the interface is not assigned, the parameters specific to the interface are replaced by spaces (20H).

16.5 CP 525: Reading and Writing the Date and Time

A CP 525 must be designated either time master or time slave.

Time slave: after powering up, the CP reacts at first as a time slave. It can update its own date and the time; however, the CPU cannot read this.

Time master: the master provides the system time; i.e., it must be able to make this available at any time.

If the CPU wishes to synchronize the CPs (at regular intervals or when certain events occur), it reads the system time from the time master and sets all the time slaves to this time. This relieves the time slaves of unnecessary updating tasks.

The CPU can set the date and time on a CP 525 at any time. It can, however, read the date and time of the CP 525 only if the latter has been designated time master.

The date/time area transferred during writing or reading, is structured as follows:

Byte	Significance
0	bit 0: master id. 1 = master 0 = slave
1	0
2	$^1/_{10}$ s $^1/_{100}$ s
3	s
4	min
5	h
6	day
7	month
8	year
9	0

In a word transfer, byte 0 corresponds to the left half of the first data word; byte 1 the right half of the first data word. The date and time are coded in BCD.

The accuracy of the clock is $^1/_{10}$ s. This means that the position $^1/_{100}$ s in byte 2 of the date/time area is always 0. When the time is written, the $^1/_{100}$ s are not accepted.

The job number 218 is fixed on the CP 525 for the jobs "read / write date and time".

SEND DIRECT 218

Using this job, the CPU declares the CP as master or slave. At the same time the CPU can also transfer the date and time to the CP.

Whether the CPU transfers the date and time depends on the source length specified (QLAE = length of the data to be transferred):

QLAE = 1 word

The CP 525 is declared as master/slave by means of the contents of bit 0 (least significant bit) in the first byte; master: bit 0 = 1 / slave: bit 0 = 0.

During the start-up or in a warm restart of the CPU, the SEND DIRECT 218 with QLAE = 1 and master bit = 1 for the CP which is to be master must be called up. The date and time on the CP 525 are not changed by QLAE = 1.

QLAE = 5 words

The CP is declared as master/slave by means of the master identifier and the date and time are transferred.

Remember that when transferring the time to a master which is to remain as master, you must set the master identifier.

You should also take into account that the date and time are adopted by the CP only during the handshake (synchronization via inputs or after reading a different clock).

Note: the source area from which the data are transferred to the CP should be a data block.

Note:

The CP 525 can be declared as time master on one or both interfaces. The date and time can only be read on the interface on which the CP is declared as master.

If the CP is declared as master on an interface and then as slave on the other interface, the CP becomes time slave on both interfaces. For example, the following situation occurs: when the date/time from the first interface is read, the error number 1F is entered in SYSTAT. When the second interface is read, the handshake is not carried out, since the bit "RECEIVE - job ready" in the ANZW (bit 0) is not set. No error message is output.

A SYNCHRON does not influence the master-slave function.

RECEIVE DIRECT 218

By means of this job the CPU can read the date and time from the CP (if the CP has already been declared time master with SEND DIRECT 218.

The destination length must be five words. Again the destination area should be a data block.

Note: the data read only has meaning if the date and time have been set at least once since the module was plugged in (with SEND DIRECT 218; length: five words).

Empty page

17 Error Handling

Errors occurring while the CP 525 is operating with the PT88/PT89 can have different causes as follows:

- on the PT88/PT89 (e.g.)

  no more paper
  no more ink
  printer offline
  printer switched off

- in the link between the CP 525 and PT88/PT89 (e.g.)

  cable disconnected
  cable defect
  wrong cable

- on the CP 525 (e.g.)

  wrong memory submodule
  no memory submodule plugged in
  job not programmed
  memory submodule not programmed
  error in user program
  hardware fault
  overflow of message FIFO

- on the CPU of the PC (e.g.)

  incorrect data (various causes possible)
  error during data exchange between CP 525 and CPU

Errors detected by the CP 525 are signalled to the CPU with as much detail as possible. You determine how the CPU is to react in your STEP 5 program on the CPU (for more details see Section 16).

You can for example have the following:

- an error message printed out on the PT88/PT89 - possibly via another CP 525 or via the other interface of the same CP 525
- the errors displayed with messages on a monitor (e.g.; with the CP 526)
- the job repeated

If there is a BREAK on the connection line, i.e. either the cable has been withdrawn or is defective or the printer is switched off, all incoming jobs are aborted with the BREAK error.

Errors are signalled to you by the following:

- an error number in the **PAFE** byte of the handling block (1.)
- an error number in the condition codeword **ANZW** of the handling block (2.)
- an error number in the error message area of the **SYSTAT** (3.)

1. The error number in the PAFE byte of the handling block (HDB)

The PAFE byte is a parameter you specify when an HDB is called. The handling blocks write to the PAFE byte if serious errors occur in connection with a CP or if parameters are incorrectly specified. If you receive a PAFE error number, this means that the data exchange between the CPU and the CP either is not started or was aborted.

A detailed description of the PAFE byte error messages can be found in the description "Using the handling blocks" in this handbook.

!IMPORTANT!: after transferring a part of a program from the PG to the CP, the error numbers 91H and C1H can occur even if there is no error. If this occurs on your system, first repeat the job affected.

2. The error numbers in the condition codeword ANZW

The condition codeword ANZW is also a parameter you specify when calling an HDB. With direct jobs (SEND DIRECT, FETCH DIRECT, RECEIVE DIRECT), ANZW indicates whether the job is running or was completed. If the job was completed with an error, an error number is entered in **bits 8 to 11** in the ANZW. Bit 3 is also set: "job terminated with error".

!IMPORTANT!: an error number is only entered in ANZW if the error occurred during the processing of a DIRECT job!

Since only four bits are available for the error numbers in ANZW, one error number normally indicates one of several error states. Because of the limited number of error numbers, the ANZW is suitable for evaluation by the STEP 5 user program.

The errors are distinguished by the various sources of error as follows:

No. for ANZW	Source
1 to 6	error in data exchange between CPU and CP 525
7	system error at the start of job processing job cannot be executed, not programmed on the CP error triggered by handling block data block contains errors
9	overflow of the message FIFO
A	errors in the dynamic parts in the PSL and frame
F	errors in the data exchange between CP 525 and PT88/PT89

Descriptions of the errors can be found in the tables on the following pages.

Note: a status byte is reserved in the dual-port RAM for each of the 223 possible jobs. If the CP detects an error in connection with a DIRECT job, it writes the error number and the bit "job terminated with error" in the status byte of the corresponding job in the dual-port RAM. To have the job status shown in the condition codeword, either

- the CONTROL HDB in the STEP 5 program must be called with the corresponding job number in the cycle (OB1), or
- the DIRECT job (SEND DIRECT) must be called with the command JU and RLO=0.

See also the description "Using the handling blocks" in this manual.

3. The error numbers in the error message area of the SYSTAT

The error message area of the SYSTAT is a data area in the dual-port RAM, which can be read by the CPU using the special job RECEIVE DIRECT 200. For each interface the error message area of the SYSTAT covers three error message bytes and an additional status byte. A detailed description can be found in Section 16 "Special jobs carried out by the CPU".

All errors recognized by the CP are entered in the error message area of the SYSTAT. If the error occurs in connection with a DIRECT job, an error number is also entered in the condition codeword.

The error entry in the SYSTAT is so detailed that it is particularly suitable for exact error analysis during commissioning. A programmed evaluation of individual error numbers is also possible.

The special job RECEIVE DIRECT 200 should be called up in OB1 to read the error message area in the SYSTAT during the commissioning phase. You can recognize the occurrence of errors by bit 0 in the ANZW of the RECEIVE DIRECT 200 being set to 1 by the CP. These error numbers are then copied into the data block specified as a parameter. If a programmer is connected to the CPU of the PC, you can display the errors directly on the monitor in the operating mode "STATUS variable" or "CONTROL variable".

To clear the error entries in the SYSTAT, call the special job RESET DIRECT 200 (for more details see Section 16).

The errors are arranged according to the various causes as follows:

No. for SYSTAT	Source
1 to 14	error in data exchange between CPU and PC
15 to 1F	system error at the start of job processing
20 to 2E	job cannot be executed error triggered by handling block errors in data block
30 to 35	errors in the dynamic parts in PSL and frame
40	error in the job list
50 to 54	error processing the messages
B0 to B8	error in data exchange between PG and CP
C0 to CA	hardware error on the module
D0 to FF	error in the data exchange between CP and PT88/PT89

The following tables provide a detailed description of the error numbers
- arranged according to the numbers for ANZW
- arranged according to the numbers for SYSTAT

and **how to remedy the errors!**

Note: the numbers of the error messages are given as hexadecimal values!

17.1 Error Messages for SYSTAT and ANZW - Arranged According to the Numbers for ANZW

No. for ANZW	No. for SYSTAT	Description
		Parameter assignment errors detected by the CPU of the PC, and signalled by the CP
1H	01H	a) Source/dest. type illegal area (start address, length) illegal (negative value)
2H	02H	b) DB/DX not present or illegal (e.g. DB0, DB1, DX0)
3H	03H	c) Area too short: (start address + length) > area
4H	04H	d) Access to area not possible for user: e.g. hardware memory too small
5H	05H	e) Wrong condition codeword: group message for all errors attributable to this ANZW

Remedy

- Check parameter assignment on CPU and CP and if necessary correct.

- Check parameter assignment on CPU and CP, if necessary set up block.

- Check parameter assignment on CPU and CP if necessary check data block/area.

- Check parameter assignment on CPU and CP - check hardware (modules).

Check whether condition codeword is valid
(see documentation for appropriate PC and processor type).

8

No. for ANZW	No. for SYSTAT	Description
6H	06H	Error code from the CPU to the CP, which the CP cannot interpret
		— or —
		Errors detected by CP during data exchange CPU <—> CP:
	10H	a) Timeout for handshake on CP
	- - - -	- - or -
	11H	b) ALL job without request or direct job despite overload
	- - - -	- - or -
	12H	c) Wrong sequence for handshake by PC
	- - - -	- - or -
	13H	d) Illegal job type or illegal job number (>223) for direct job
	- - - -	- - or -
	14H	e) Illegal acknowledgement from CPU during handshake

Remedy
Evaluate PAFE in handling block (SEND DIRECT or SEND ALL)
Check STEP 5 program, e.g., CPU in stop state or SEND ALL job missing.
Check STEP 5 program, e.g., errors in handling block.
Check STEP 5 program, e.g., errors in handling block.
Check STEP 5 program, e.g., incorrect parameters for handling block. (Evaluate PAFE in handling block SEND DIRECT or SEND ALL).
Check STEP 5 program, e.g., errors in handling block.

No. for ANZW	No. for SYSTAT	Description
		System errors at start of job processing:
7H	15H	a) Number of direct jobs that can be processed simultaneously too large for CP 525
	- - - -	- - or -
	16H	b) Current job was terminated with error status when CP was cold restarted (power on) or cold restarted from PG
	- - - -	- - or -
	17H	c) Area is masked by STEP 5 program (bit 7 im ANZW = 1)
	- - - -	- - or -
	1EH	d) Field length for "read SYSID" (RECEIVE DIRECT 223) too small
	- - - -	- - or -
	1FH	e) "Read date/time" (RECEIVE DIRECT 218) illegal, CP is set as time slave
	- - - -	- -

Remedy

Modify STEP 5 program so that maximum 10 direct jobs run simultaneously.

There is no remedy with power on. When cold restarting the CP from the PG you should make sure that no more CPU jobs are running before writing to an interface. The SYNCHRON HDB clears the job statuses in the dual-port-RAM but not the SYSTAT entry!

Check STEP 5 program.

For SYNCHRON the field length should be greater than or equal to 128 bytes or do without "Read SYSID" (data are nevertheless available in the dual-port RAM).

Before reading the date/time, the CP must be designated as master (see Section 16) - setting applies to this module! Check whether the CP has been set to slave on the other interface.

No. for ANZW	No. for SYSTAT	Description
		Job cannot be executed
7H	20H	a) A job triggered by a SEND DIRECT has not been programmed on the CP
	21H	b) CHAINED LIST (CL): File not present on CP 525. In a CL the individual PSLs to be output are specified using their names. One or more of the PSL's specified in the CL is/are not on the CP 525.
	22H	- or - c) MESSAGE: The message is marked as being on the CP it does not, however, exist.
	23H	d) FRAME: No frame (list header/trailer) has been programmed on the CP.
	24H	- or - e) The job "update SML" cannot be executed, because no SML has, as yet, been triggered on the CP 525.

Remedy

- Wrong job number on CPU => correct STEP 5 program
  or
- Wrong job number on CP => correct CP 525 program

- Transfer missing PSLs to CP 525
  or
- Correct CL program, i.e., delete the PSLs not on the CP from the CL

- Delete messages on the CP and transfer again.

- Program frame and transfer to CP 525
  or
- Program SML without frame and transfer

- Program SML (if it does not already exist)and transfer to the CP and
- Trigger SML from the CPU using its job number and start "update SML" again

No. for ANZW	No. for SYSTAT	Description
7H	27H	f) The mode selector on the front of the CP 525 is set to STOP/PGR; the CP 525 is in the wait state - PC jobs cannot be processed
		Errors triggered by handling block
	28H	a) The PC job is not a SEND job
	29H	b) Function illegal - R/W function illegal or - instead of SEND DIRECT with parameters there was a SEND DIRECT without paras.
	2AH	c) No data block or DB-No. < 3; for data transfer to the CP only the QTYP DB with a DB-No. > 2 is permitted
	2BH	d) DB start address (QANF) not 0 or source length (QLAE) > 256 Datenworte
		Errors in data block:
	2CH	a) Syntax error in DB M-RECORD - DLO not 33H - unknown parameter type (in area: 9 < parameter type < FFH illegal) - body length not correct - length transferred < body length (DRO)
	2DH	b) DB M OLDNEW: length transferred < 255 data words or Contents of DW 253 and/or DW 254 not correct

Remedy

- Do not send any new PC jobs
  or
- Switch selector to RUN

Correct STEP 5 program.

Correct STEP 5 program
- correct parameter assignment of HDB
- SML and CML require a SEND DIRECT with parameters

Correct STEP 5 program.

Correct STEP 5 program.

Check structure of DB M-RECORD and correct.

8

Correct DB M OLDNEW or transfer with correct length: DB M-OLDNEW must include the lower limit of the message numbers in DW 253 and in DW 254 the number of words containing bits to trigger messages

---

No. for ANZW	No. for SYSTAT	Description
7H	2EH	c) Update group inhibit bits: the data block containing the data word with the group inhibit bits was transferred with a length less than the DB address programmed with COM 525.

Remedy

- Transfer DB with correct length
  or
- Correct DB address with COM 525

No. for ANZW	No. for SYSTAT	Description
9H	54H	**Overflow of message FIFO:** The messages in DB M RECORD transferred to the CP 525 cannot be entered in the message FIFO since this has not yet been emptied by printing out the messages (max. 128 entries possible!). Causes: - printer is not ready to receive (the error number 0E2H or 0FFH has been entered in SYSTAT) - messages are arriving more quickly than the printer can print them out - a long list which cannot be interrupted is being printed out

Remedy

Correct and send DB M RECORD again.

Correct:
- printer switched on? cable plugged in?
  paper run out? ink run out? printer offline?

- find out why so many messages are arriving and take action to reduce the number

- wait and then send DB M RECORD again; further messages can only be processed after this list has been printed out.

No. for ANZW	No. for SYSTAT	Description
AH		**Errors in the dynamic parts of the PSL and frame:**
	30H	a) Wrong output format in PSL (data source: data block from the CPU)
	- - - -	- or -
	31H	b) Wrong output format in FRAME
	- - - -	- or -
	32H	c) Value for variable type "process status" outside range of values (0<=VALUE<= 7) (data source: data block from the CPU)
	- - - -	- or -
	33H	d) PSL: the DB address of an output field is outside the area of the data which were transferred by the CPU. The DB address indicates invalid data
	- - - -	- or -
	34H	e) Frame: the DB address of an output field is outside the area of the data which were transferred by the CPU. The DB address indicates invalid data
	- - - -	- -
	35H	f) Process status variable: the DB address of a process status variable is outside the area of the data which were transferred by the CPU. The DB address indicates invalid data

Remedy

- Check the format parameters for the dynamic fields of the PSL with COM 525 and if necessary correct
- Check the contents of the DB on the CPU side and correct

- Check the format parameters for the dynamic fields of the frame with COM 525 and if necessary correct
- Check the contents of the DB on the CPU side and correct

The contents of the data word in the data block must not exceed the specified range of values

---

- Match the DB addresses of the dynamic field to the length of the data block
- Match the length of the data block on the CPU

- Match the DB addresses of the dynamic field to the length of the data block
- Match the length of the data block on the CPU

- Match the DB address of the dynamic field to the length of the data block
- Match the length of the data block on the CPU

8

No. for ANZW	No. for SYSTAT	Description
FH	FFH	**Error in data exchange between CP and PT88/PT89:** Control character BREAK on the line

Remedy

Causes: - cable defect
- cable not plugged in
- printer not present
- printer switched off
—> check and correct

17.2 Error Messages for SYSTAT and ANZW - Arranged According to Nos. for SYSTAT

No. for SYSTAT	No. for ANZW	Description
		Parameter assignment errors detected by the CPU of the PC, and signalled by the CP
01H	1H	a) Source/dest. type illegal area (start address, length) illegal (negative value)
02H	2H	b) DB/DX not present or illegal (e.g. DB0, DB1, DX0)
03H	3H	c) Area too small: (start address + length) > area
04H	4H	d) Access to area not possible for user: e.g. hardware memory too small
05H	5H	e) Wrong condition codeword: group message for all errors attributable to this condition codeword
06H	6H	f) Error indicated by PC to the CP, which cannot be interpreted by CP

Remedy

- Check parameter assignment on CPU and CP and if necessary correct.

- Check parameter assignment on CPU and CP, if necessary set up block.

- Check parameter assignment on CPU and CP if necessary check data block/area.

- Check parameter assignment on CPU and CP - check hardware (modules).

Check whether condition codeword is valid
(see documentation for appropriate PC and processor type).

Evaluate PAFE in block on the CPU.

No. for SYSTAT	No. for ANZW	Description
		Errors detected by CP during data exchange CPU <—> CP:
10H	6H	a) Timeout for handshake on CP
- - - -	- - - -	- - or -
11H	6H	b) ALL job without request or direct job despite overload
- - - -	- - - -	- - or -
12H	6H	c) Wrong sequence for handshake by PC
- - - -	- - - -	- - or -
13H	6H	d) Illegal job type or illegal job number (>223) for direct job
- - - -	- - - -	- - or -
14H	6H	e) Illegal acknowledgement from CPU during handshake
		System errors at start of job processing:
15H	7H	a) Number of direct jobs which can be processed simultaneously too large for CP 525
- - - -	- - - -	- - or -
16H	7H	b) Current job was terminated with error status when CP was cold restarted (power on) or cold restarted from PG
- - - -	- - - -	- - or -
17H	7H	c) Area is masked by STEP 5 program (bit 7 im ANZW = 1)
- - - -	- - - -	- - or -
1EH	7H	d) Field length for "read SYSID" (RECEIVE DIRECT 223) too small
- - - -	- - - -	- - or -
1FH	7H	e) "Read date/time" (RECEIVE DIRECT 218) illegal, CP is set as time slave

Remedy

Check STEP 5 program,
e.g., CPU in stop state or SEND ALL job missing.

- -

Check STEP 5 program,
e.g., errors in handling block.

- -

Check STEP 5 program, e.g., errors in handling block.

- -

Check STEP 5 program, e.g., wrong parameters for handling block (Evaluate PAFE in handling block SEND DIRECT or SEND ALL).

- -

Check STEP 5 program,
e.g., errors in handling block.

Modify STEP 5 program so that a maximum of 10 direct jobs run simultaneously.

- -

There is no remedy with power on. When cold restarting the CP from the PG you should make sure that no more CPU jobs are running before writing to an interface. The SYNCHRON HDB clears the job statuses in the dual-port-RAM but not the SYSTAT entry.

- -

Check STEP 5 program.

- -

For SYNCHRON the field length should be greater than or equal to 128 bytes or do without "Read SYSID" (data is nevertheless available in the dual-port RAM).

- -

Before reading, the CP must be designated as master (see Section 16) - setting applies to this module. Check whether the CP has been set to slave on the other interface.

No. for SYSTAT	No. for ANZW	Description
		Job cannot be executed
20H	7H	a) A job triggered by a SEND DIRECT has not been programmed on the CP
21H	7H	b) CHAINED LIST (CL): File not present on CP 525. In a CL the individual PSLs to be output are specified using their names. One or more of the PSLs specified in the CL is/are not on the CP 525
		- or -
22H	7H	c) MESSAGE: The message is marked as being on the CP it does not, however, exist
23H	7H	d) FRAME: No frame (list header/trailer) has been programmed on the CP
		- or -
24H	7H	e) The job "update SML" cannot be executed, because no SML has, as yet, been triggered on the CP 525

Remedy

- Wrong job number on CPU => correct STEP 5 program
  or
- Wrong job number on CP => correct CP 525 program

- Transfer missing PSLs to CP 525
  or
- Correct CL program; i.e., delete the PSLs not on the CP from the CL

Delete messages on the CP and transfer again

- Program frame and transfer to CP 525
  or
- Program SML without frame and transfer

- Program SML (if it does not already exist)and transfer to the CP and
- Trigger SML from the CPU using its job number and start "update SML" again

8

No. for SYSTAT	No. for ANZW	Description
27H	7H	f) The mode selector on the front of the CP 525 is set to STOP/PGR; the CP 525 is in the wait state - PC jobs cannot be processed
28H	7H	**Errors triggered by handling block** a) The PC job is not a SEND job
29H	7H	b) Function illegal - R/W function illegal or - instead of SEND DIRECT with parameters there was a SEND DIRECT without paras.
2AH	7H	c) No data block or DB-No. < 3; for data transfer to the CP only the QTYP DB with a DB-No. > 2 is permitted.
2BH	7H	d) DB start address (QANF) not 0 or source length (QIAE) > 256 data words
2CH	7H	**Errors in data block:** a) Syntax error in DB M RECORD - DLO not 33H - unknown parameter type (in area: 9 < parameter type < FFH illegal) - body length not correct - length transferred < body length (DRO)
2DH	7H	b) DB M OLDNEW: length transferred < 255 data words or Contents of DW 253 and/or DW 254 not correct

Remedy

- Do not send any new PC jobs
  or
- Switch selector to "RUN"

Correct STEP 5 program.

Correct STEP 5 program:
- correct parameter assignment of HDB
- SML and CML require a SEND DIRECT with parameters

Correct STEP 5 program.

Correct STEP 5 program.

Check structure of DB M RECORD and correct.

8

Correct DB M OLDNEW or transfer with correct length: DB M OLDNEW must include the lower limit of the message numbers in DW 253 and in DW 254 the number of words containing bits which will trigger messages

---

No. for SYSTAT	No. for ANZW	Description
2EH	7H	c) Update group inhibit bits: the data block containing the data word with the group inhibit bits was transferred with a length less than the DB address programmed with COM 525.

Remedy

- Transfer DB with correct length
  or
- Correct DB address with COM 525

No. for SYSTAT	No. for ANZW	Description
		Errors in the dynamic parts of the PSL and frame: The contents of a data word for an output field (process status variable, process variable) cannot be output in the format which was specified when you programmed the dynamic field. Example: format CHAR DB contents 00H 00H —> error, since format CHAR only in range of values 20H to 7EH The number in SYSTAT provides information about the source of the output. The field in which the error occurs is identified in the printout, when filling characters are printed instead of the expected data.
30H	AH	a) Wrong output format in PSL (data source: data block from the CPU)
- - - -	- - - -	- or -
31H	AH	b) Wrong output format in FRAME
- - - -	- - - -	- or -
32H	AH	c) Value for variable type "process status" outside range of values (0<=VALUE<= 7) (data source: data block from the CPU)
- - - -	- - - -	- or -
33H	AH	d) PSL: the DB address of an output field is outside the area of the data which were transferred by the CPU. The DB address indicates invalid data
- - - -	- - - -	- or -
34H	AH	e) Frame: the DB address of an output field is outside the area of the data which were transferred by the CPU. The DB address indicates invalid data
- - - -	- - - -	- -

Remedy

- Check the format parameters for the dynamic fields of the PSL with COM 525 and if necessary correct.
- Check the contents of the DB on the CPU side and correct.

- Check the format parameters for the dynamic fields of the frame with COM 525 and if necessary correct.
- Check the contents of the DB on the CPU side and correct.

The contents of the data word in the data block must not be outside the specified range of values.

8

- Match the DB address of the dynamic field to the length of the data block.
- Match the length of the data block on the CPU.

- Match the DB address of the dynamic field to the length of the data block.
- Match the length of the data block on the CPU.

No. for SYSTAT	No. for ANZW	Description
35H	AH	f) Process status variable: the DB address of a process status variable is outside the area of the data transferred by the CPU. The DB address indicates invalid data.
40H	-	**Error in job list:** the job list generated with COM 525 cannot be processed by the PT88 interpreter.
50H	-	**Errors processing the messages:** a) Message on the CP 525 and content of the message record do not match. The message programmed with COM 525 contains a dynamic field for the output of a parameter. In the corresponding message record of DB M RECORD (same message number) the value OFFH is specified in the parameter type field, i.e. "message contains no parameters". Filling characters will be printed out.
51H	-	b) Wrong status specification in a message record. The message status is represented in the message record by 2 bits. The following combinations and assignments result: 00 0 GOING 01 1 COMING 10 2 ACKNOWLEDGED 11 3 not allowed The combination 11 generates this error message. Filling characters are printed out.

Remedy

- Match the DB address of the dynamic field to the length of the data block.
- Match the length of the data block on the CPU.

Indicates a defective user memory submodule. Replace the user memory submodule and program again.

Match up the message text on the CP 525 with the message record.

- -

Correct message record.

8

- -

No. for SYSTAT	No. for ANZW	Description
52H	-	c) Wrong format in message record: the values specified in the message record for date/time cannot be output in the specified format (BCD).
53H	-	d) Wrong format in message record: the parameter in the message data record does not match the format specified in the "parameter type" field.
54H	9H	**Overflow of message FIFO:** The messages in DB M RECORD transferred to the CP 525 cannot be entered in the message FIFO since this has not yet been emptied by printing out the messages (max. 128 entries possible!). Causes: - printer is not ready to receive (the error number 0E2H or 0FFH has been entered in SYSTAT) - messages are arriving more quickly than the printer can print them out - a long list which cannot be interrupted is being printed out

Remedy

Correct message record or check whether transfer/reception and storing of date/time functions correctly.

- -

Correct message record or check whether the definition of the message parameter is correct.

- -

Correct and send DB M RECORD again!

Correct:

- printer switched on? cable plugged in?
  paper run out? ink run out? printer offline?

- find out why so many messages are arriving and take action to reduce the number

- wait and then send DB M RECORD again; further messages can only be processed after this log has been printed out.

No. for SYSTAT	No. on PG	Description
		Error messages when sending a telegram from the CP to the PG:
B0H	-	a) Link establishment with STX was either not acknowledged by the PG within 300 ms acknowledgement monitoring time) or was not acknowledged with DLE ACK or DLE NAK
B1H	Err. 722 transfer error	b) Transmission (CP -> PG) aborted by PG with NAK
B2H		c) The link termination with DLE ETX acknowledged by the PG with DLE NAK
B3H		d) The link termination with DLE ETX not acknowledged by the PG with DLE ACK or DLE NAK within 300 ms (acknowledgement monitoring time)
		Error messages when receiving a telegram from the PG:
B4H	Err. 722 transfer error	a) Transfer error when receiving (CP aborts with DLE NAK)

For further information see the description "COM 525 messages" in this manual.

Remedy

- Check cable
- COM 525 loaded?
- Hardware error at PG or CP

Repeat transfer (otherwise: see B0).

Repeat transfer (otherwise see B0).

- Check cable
- COM 525 loaded?
- Hardware error at PG or CP

Repeat transfer (otherwise: see B0).

8

No. for SYSTAT	No. on PG	Description
		Errors carrying out a job from the PG:
B5H	IF.007 CP funct. unknown	a) A function is to be carried out which is unknown to the CP 525 or there is a transfer error in the exchange of telegrams.
B6H	ERR. 709 CP memory full	b) The user memory of the CP 525 is full.
B7H	ERR. 027 EPROM plugged in	c) The write access by the PG cannot be carried out, because there is no RAM submodule plugged in as the user memory of the CP 525 (either EPROM or no submodule plugged in).
B8H	ERR. 721 CP switch set to "STOP"	d) The mode selector on the CP is set to "STOP" or "PGR" (during cold restart from the PG).

Remedy

- COM 525 loaded?
- Repeat transfer

- Plug in submodule with greater capacity
- Erase unnecessary module files on the CP

Plug in RAM.

Switch mode selector on the CP to "RUN".

No. for SYSTAT	No. for ANZW	Description
		Hardware errors on the module:
C0H*)	-	a) Error found in EPROM test
C1H	-	b) RAM on module defect
C2H	-	c) Dual-port RAM on module defect
C3H	-	d) RAM on memory submodule card defect (check only when interface not loaded)
C4H*)	-	e) User memory submodule not permitted
C5H	-	f) Error in memory submodule while program is running
C6H*)	-	g) EPROM memory submodule empty or contents unknown

*): These errors are signalled only on interface 1. The CP was not synchronized; the data cannot be read by the handling block. These error messages must then be read directly from the dual-port RAM (structure from address F660H onwards - see Section 16.3).

Remedy

- Wrong or defect EPROM
- Replace EPROM

Replace module.

Replace module.

Replace RAM submodule.

- Use permitted submodule type
- Submodule or submodule controller defect

Contact customer service.

- Check whether correct EPROM submodule
- Check whether submodule is programmed

With submodules which are programmed for the CP 525-2 and used in the CP 524, this error number appears if interface 1 is not programmed.

No. for SYSTAT	No. for ANZW	Description
C8H*)	-	i) Only applies to CP 524: interface module (identifier) unknown or module not plugged in Note: printer PT88/PT89 only in conjunction with CP 525-2 !!
C9H	-	j) Interface not loaded or no cold restart carried out after loading
CAH*)	-	k) Memory submodule not plugged in

*): This error is signalled only on interface 1. The CP was not synchronized; the data cannot be read by the handling block. These error messages must then be read directly from the dual-port RAM (structure from address F660H onwards - see Section 16.3).

Remedy

- Check whether correct module, if "yes": module defect
- Check whether module plugged in correctly

- Load interface if necessary or
- Carry out cold restart

Caution: the entry in SYSTAT is not cleared by loading the interface, but only by RESET DIRECT 200.

Check whether submodule is plugged in; if "yes": submodule defective (or submodule controller defect)

No. for SYSTAT	No. for ANZW	Description
		Errors in data exchange between CP and PT88/PT89:
D0H	-	a) Procedure and interpreter on the CP 525 do not match.
E2H	-	b) Timeout: printer has not reacted for more than one minute
E3H		c) Illegal (although correct) character received: - printer type not allowed - printer or procedure (COM 525) does not have correct parameters (data rate/ no. of stop bits/parity bits)
FEH		d) Incorrect character received: - data rate not correctly set at printer or in the procedure (COM 525) - cable or connector damaged
FFH	FH	e) Control character BREAK on the line

Remedy

Check procedure/interpreter on CP 525 and if necessary replace.

- -

The following causes are possible:
- no more paper on PT88/PT89
- ink run out
- printer switched offline

=> check and deal with problem

- -

- Check whether printer is allowed, and if not replace (only PT88 and PT89 allowed).
- Compare printer parameter assignment with procedure parameter assignment and if necessary correct.

- -

Compare data rate settings on printer and procedure (COM 525) and if necessary readjust.

Check cable and connector and if necessary replace.

- -

The following causes are possible:
- cable defective
- cable not plugged in
- printer not present
- printer switched off

——> Check and correct.

18 What Does It Mean When?

What does it mean when filling characters are output by the printer?

Cause: the data in the data block of the CPU could not be converted.

Remedy: the CP 525 signals the error to the CPU. Check under the corresponding error number for the ANZW and for SYSTAT (Section 17 "Error handling").

Note: lists that contain filling characters are printed out completely. The job is, however, signalled as "terminated with error".

What does it mean when CML, PSL or CL can be interrupted by messages; the messages are, however, only printed out at the end of the lists?

Cause: to have messages printed out of order, the CP 525 must be informed that an SML has been programmed. It is aware of this as soon as the job "SML" has been initiated as least once following a cold restart or a SYNCHRON.

Remedy: initiate SML after a cold restart or SYNCHRON (e.g. with empty DB M RECORD).

What does it mean when the PG writes to the CP 525 (transfer, delete) but the operation is not carried out?

Cause: the CP 525 is processing a job and cannot be interrupted. As soon as the CP 525 can carry out the PG job the LED on the device interface will light up.

What does it mean when the date and time are output incorrectly?

Cause: If a data block has been specified as the source for date and time, the data must be entered in the specified data word in the following order:

		DL	DR	
time:	DW n	S	M	S = seconds / M = minutes
	DW n+1	H		H = hours
date:	DW m	D	M	D = day / M = month
	DW m+1	Y		Y = year

Remedy: The data must be stored in the data word as described.

Note: If the data is read with the RECEIVE DIRECT 218 job, it is entered in the data block as shown below:

	DW no.	DL	DR
time:	n	ms	S
	n+1	M	H
date:	n+2	D	M
	n+3	Y	

Note that your STEP 5 user program enters the time in the data word as shown beside *). Only then is the time output correctly.

8

What does it mean when an unwanted form feed takes place during the output of a list?

Cause: If the lines to be output are longer than the page width, then line overflow must result. This is not recorded, so that the internal line count no longer agrees with the actual number of lines printed out.

Remedy: Only program individual lines up to the maximum length so that no line overflow occurs.

What does it mean when individual lines are not printed out?

Cause: These lines may belong to inhibited groups.

Remedy: Release the groups.

Note: With SML it is possible that a list header will be printed out, but that no messages follow, since these are inhibited.

What do I do if the "form feed" job is initiated but is not carried out?

Cause: If the printer head is at the start of a page the PT88/ PT89 ignores all jobs requiring a form feed. This saves unnecessary waste of paper.

Note: When the power is switched on the printer interprets the current print head position as the start of a page.

What do I do if a CML is not output correctly?

Remedy: Before a CML is output, an SML must have been started at least once, otherwise the CP is not aware of the presettings.

What does it mean when I find that the program sometimes fits the RAM and sometimes does not, when I am transferring the CP 525 user program from the PG to the CP?

Cause: You may have transferred the whole program at one time; and then another time transferred the parts of the program separately. When the parts of the program are transferred to the CP in a different order, the memory manager may itself require more space, leaving less space for the program.

What do I do if I want to transfer new program parts to the second device interface of the CP and error 709 ("Working diskette/CP memory full") is displayed, although the user program for the first interface has just been deleted since it is no longer required?

Remedy: Transfer the entire CP 525 user program to the second interface. Before the "TOTAL PROGRAM" transfer, the corresponding interface is first deleted.