SIMATIC S5
CP 521
Printer/ASCII Communications Module

Manual

Order No. 6ES5998-0UD21

GWA4NEB812 6030-02b

◆

Siemens has developed this document for its licensees and customers. The information contained herein is the property of Siemens and may not be copied, used, or disclosed to others without prior written approval from Siemens. Users are cautioned that the material contained herein is subject to change by Siemens at any time and without prior notice.

Siemens shall not be responsible for any damages, including consequential damages, caused by reliance on material presented, including but not limited to typographical, electronic, arithmetic,or listing errors.



⚠ WARNING

Hazardous voltage.

Can cause death, severe personal injury, or substantial property damage.

Restrict use to qualified personnel.
See safety instructions.

Only qualified personnel should install or maintain this equipment after becoming thoroughly familiar with all warnings, safety notices, and maintenance procedures contained in this manual. The successful and safe operation of this equipment is dependent upon proper handling, installation, operation, and maintenance.

The following are definitions of the terms “qualified person,” “danger,” “warning,” and “caution,” as applicable for this document.

Qualified Person
One who is familiar with the installation, construction, and operation of this equipment and the hazards involved. In addition, the person should have the following qualifications:

- Be trained and authorized to use and tag circuits and equipment in accordance with established safety practices
- Be trained in the proper care and use of protective equipment in accordance with established safety practices
- Be trained in rendering first aid

DANGER

Indicates loss of life, severe personal injury, or substantial property damage will result if proper precautions are not taken.

WARNING

Indicates loss of life, severe personal injury, or substantial property damage can result if proper precautions are not taken.

CAUTION

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

STEP 5® and SIMATIC® are registered trademarks of Siemens AG.

Copyright © Siemens AG 1989
Third Edition, September 1989
Printed in the Federal Republic of Germany

◆

Preface	
Introduction	
System Overview	1
Technical Description	2
Hardware Installation	3
Notes on Operation	4
Principle of Operation and Adressing	5
Print Mode	6
ASCII Mode	7
ASCII Code Table	8
Appendices	A/B
Index	

Preface

The CP 521 is a powerful printer/ ASCII communications module for the S5-1OOU programmable controller. To be able to make the best use of the module, you require detailed information.

The performance of the S5-1 OOU is constantly being increased. For this reason, the manual of the S5- 10OU has become very large.
This manual only gives facts and examples which are relevant to the CP 521. Demands for a higher quality of technical documentation have also been met, which means:

. Standardization of terminology
- **More detailed breakdown of subjects**
- **Illustration of individual problems**
- **User-friendly arrangement of the contents**

The aim is that both users with little previous experience and SIMATIC S5 experts should find all the information that they require to work with the CP 521.

However, the applications are so numerous that not all the problems that might occur can be dealt with in one manual. For other problems, please ask your Siemens representative for advice.

Introduction

It is important to study the introduction carefully before reading the rest of the manual. This will help you to use the manual and will save you time.

Description of contents

This manual is a comprehensive description of the CP 521. The manual can be divided into blocks according to topics.

- Description
  - The "System Overview" contains information on the tasks the module can perform and on how the module is used in the S5-100U.
  - The "Technical Description" contains general information on the principle of operation of the module, technical specifications, details of the serial interface and input/output (I/O) modules, and a list of accessories.

- Installation and operation
  - The "Hardware Installation" contain all the information you require to install the module and connect it to a printer or other peripheral device (connector pin assignments and terminal diagrams).
  - "Notes on Operation" describes restart characteristics and actual operation. This chapter also contains the error messages that can occur during startup.

- Principle of operation and addressing
  - This is a general description of how data transfer between the CPU, the CP 521 and the I/O modules is coordinated.
  - This chapter also describes how the module is addressed.

- Functional description
  We have devoted a separate chapter to the "Printer mode" and the "ASCII mode". Each of these chapters contains the information necessary for programming the module for the relevant mode. This saves you time-consuming searches in the manual.

- Appendix A: Summary
  All the information you require to operate the CP 521 has been gathered together here in list form. This is for those users who have read the manual and wish to find some brief item of information quickly.

- Appendix B: Application example
  This recaps all user inputs from installation through cabling to programming illustrated by the way of example of a point-to-point connection.
  The application example illustrates the ASCII mode and contains an executable program.

Since the module can only be used in conjunction with the S5-100 U, it is assumed that you have a copy of the S5-100U manual.The basic information on the STEP 5 programming language contained in the S5-100U manual is sufficient for operating the CP 521.

Conventions

In order to improve the readability of the manual, a menu-style breakdown has been used, i.e.:

- The individual chapters can be quickly located by means of a thumb register.
- There is an overview containing the headings of the individual chapters at the beginning of the manual.
- Each chapter is preceded by a breakdown of its subject matter.
  The individual chapters are subdivided into sections and subsections. Boldface type is used for further subdivisions.
- Pages, figures and tables are numbered separately in each chapter. The page following the chapter breakdown contains a list of the figures and tables appearing in that particular chapter.

Certain conventions were observed when writing the manual. These are explained below.

- A number of abbreviations have been used.
  Example: Programmer (PG)
- Footnotes are identified by superscripts consisting of a small digit (e.g. "1") or "*". The actual footnote is generally at the bottom left of the page or below the relevant table or figure.
- Cross-references are shown as follows:
  "(7.3.2)" refers to subsection 7.3.2.
  No references are made to individual pages
- All dimensions in drawings etc. are given in millimetres/inches (mm/in.).
- Information of particular importance is framed in
  grey-bordered rectangles.
- All data in connection with the programmer refers to the English version of the PG 615 and PG 685. The relevant user manual contains a detailed description of programming procedures with this programmer.
- Alle Angaben zu den Programmiergeräten beziehen sich auf die deutschen Versionen des PG 615 und PG 685. Eine ausführliche Beschreibung der Programmierung mit diesen Geräten finden Sie in den zugehörigen Bedienungsanleitungen.

Manuals can only describe the current version of the programmer. Should modifications or supplements become necessary in the course of time, a supplement will be prepared and included in the manual the next time it is revised. The relevant version or edition of the manual appears on the cover. The present manual is edition "2". In the event of a revision, the edition number will be incremented by "1".

At the end of the manual you will find correction forms.Please enter in these forms any suggestions you may have in the way of improvements or corrections and send them to us. Your comments will help us to improve the next edition.

Courses

Siemens provide SIMATIC 55 users with extensive opportunities for training.

For more information, please contact your Siemens representatives.

Reference literature

This manual is a comprehensive description of the CP 521. Topics not specific to the CP 521 however, are only briefly dealt with. You will find more detailed information in the following literature:

- **Speicherprogrammierbare Steuerungen SPS (only available in German)**
  Volume 1: Logic and sequential controls; from the control problem to the control program.

  Günter Wellenreuther, Dieter Zastrow
  Braunschweig 1987

  Contents:
  - How a programmable controller works
  - The theory of logic control using the STEP 5 programming language for SIMATIC 55 programmable controllers.

  Order No.: ISBN 3-528-04464-0

- **Automating with the S5-115U**
  SIMATIC 55 Programmable Controllers

  Hans Berger
  Second, revised edition, Siemens AG, Berlin and Munich 1987

  Contents:
  - STEP 5 programming language
  - Program processing
  - Integral program blocks
  - Interfaces to the peripherals

  Order No.: ISBN 3-8009-1526-X

- Programming Primer for the SIMATIC (R) S5-100U
  Practical Exercises with the PG 615 Programmer

  Siemens AG, Berlin and Munich, 1988

  Contents:
  - Assembly and installation of the S5-100U
  - Introduction to programming with the PG 615

  Order No.: ISBN 3-8009-1500-6

- Remarks Form

  The Remarks Form is provided for your comments and suggestions. If you are in the United States, please use the postage-paid form.

Conventions

The following conventions are used in this book and are listed for your reference.

Convention	Definition	Example
	A box that indicates a type of hazard,describes its implications, and tells you how to avoid the hazard is safety notation. Some safety notation includes a graphic symbol representing an electrical or radio-frequency hazard. All safety notation has one of the following levels of caution: • Indicates that loss of life, severe personal injury, or substantial property damage will result if proper pre-cautions are not taken. • Indicates that loss of life, severe personal injury, or substantial property damage can result if proper pre-cautions are not taken. • Indicates that minor personal injury or property damage can result if proper precautions are not taken.	

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. Only if you are in the United States of America should you use the postage-paid form.

Returning this form assures you of receiving all changes and updates to this document.

Please check the industry that applies to you:

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

1. Is this document well organized?		Yes	No
2. Is the information easy to find?		Yes	No
3. Is the text easy to understand?		Yes	No
4. Is the document technically accurate?		Yes	No
5. Does the level of technical detail meet your requirements?		Yes	No
6. Would you like more illustrations?		Yes	No
7. Please ratethe quality of the graphics:	Excellent	Good	
	Adequate	Poor	

Additional Comments:

_____

_____

_____

_____

_____

Manual: CP 521 Order No.: 6ES5 998-8RD21 Edition 3

Name: _____

Title: _____

Company Name: _____

Address: _____

City: _____ State: _____ Zip: _____

1 System Overview

Figures

1-1. Complete Configuration: A Programmable Controller with a CP 521 and Printer . . . . . 1 - 2

1-2. Complete Configuration: Point-To-Point Connection . . 1 - 3

1 System Overview

The CP 521 is a powerful and active printer / ASCII communications processor module that can be used in the S5-100U (not with CPU 100, 6ES5 100-8MA01). The module has a dedicated processor and can be operated in print mode or ASCII mode.

Print mode enables printout of message texts. This allows you to document process states and process faults.

In ASCII mode, the module can exchange data with other peripheral devices such as terminal devices, communications modules or also other CP 521s. In this way, you can network programmable controllers (point-to-point connection).

The peripheral device and the CP 521 are connected via a serial interface. You have the option of a passive TTY current-loop interface or an RS-232-C (V.24) interface.

Parameter assignment to the printer interface and configuration of message texts is supported by the DB editor of the programmers. The parameters of the printer interface are either stored in DB 1 on a memory submodule or transferred direct in the user program.

In print mode, the CP 521 assumes a printer with a RS-232-C (V.24) or active TTY interface. The printout of message texts does not increase the response time of the programmable controller. You can have the following printed out:

- Texts configured by the user on a memory submodule (data blocks 2 to 63).
- Time of day and date from the module's own real-time clock.
- Values for variables to be transferred via the I/O bus to the CP 521.

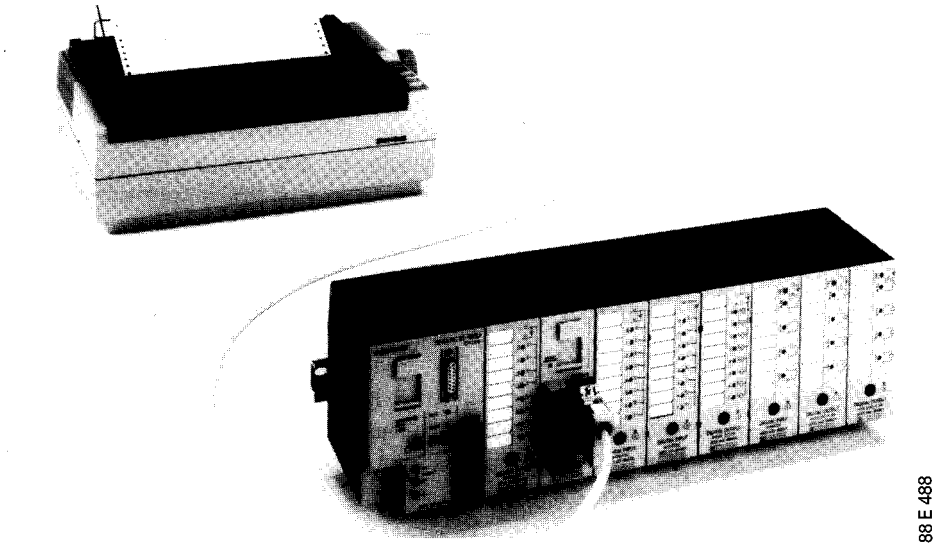


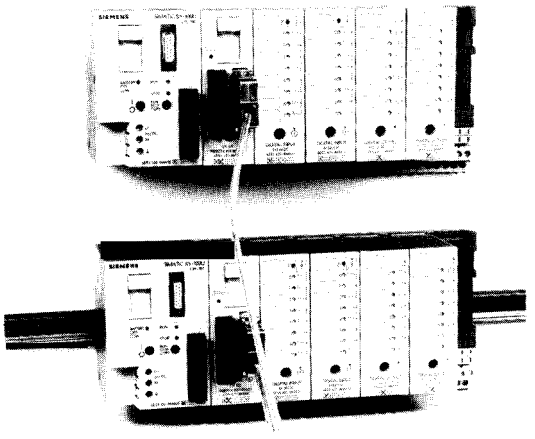
Figure 1-1. Complete Configuration: A Programmable Controller with a CP 521 and Printer

In In ASCII mode, the CP 521 assumes a peripheral device (terminal, modem, CP 523 etc.) with an RS-232-C(V.24) interface or active TTY interface. ASCII mode enables you to transmit message frames between the CPU and an peripheral device connected to the CP 521. Maximum data throughput is six bytes of useful data per two program scans. This means that a maximum of 60 bytes/s can be transmitted given a program scan time (OB 1) of, for example, 50 ms.
The CP 521 can be used to

The CP 521 can be used to

. Send fixed-length message frames
. Send variable-length message frames
- Receive fixed-length message frames
- Receive variable-length message frames

You can define the transmission mode in the user program.
The time from the module's own real-time clock can also be used in ASCII mode for tasks in the user program requiring the date and time of day.



GWA 4024/2

Figure 1-2. Complete Configuration: Point-To-Point Connection

2 Technical Description

2.1 Principle of operation of the CP 521 . . . . . 2 - 1

2.2 Technical Specifications . . . . . 2 - 4

2.3 Serial Interface CP 521 - Peripheral Device . . . . . 2 - 5
2.3.1 Print Mode . . . . . 2 .- 7
2.3.2 ASCII Mode . . . . . 2 .- 8

2.4 Memory submodule . . . . . 2 - 11

2.5 Hardware Clock . . . . . 2. - 12

2.6 Backup Battery . . . . . 2 . - 12

2.7 Accessories List and Ordering Data . . . . . 2 - 13

Figures

2-1. Schematic Representation of the CP 521 . . . . . 2 - 1
2-2. 11-Bit Character Frame with 7 Data Bits . . . . . 2 - 5
2-3. 11-Bit Character Frame with 8 Data Bits . . . . . 2 - 6
2-4. Timing Diagram for Data Transfer Between the CP and the Peripheral Device . . . . . 2 - 10

Tables

2-1. Methods of Transmission in Print Mode . . . . . 2 - 7
2-2. Valid Control Signals of the RS-232-C (V.24) Interface in Handshake ON Mode . . . . . 2 - 9
2-3. Overview of Permissible User Submodules . . . . . 2 - 11

2 Technical Description

2.1 Principle of operation of the CP 521

The CP 521 handles data transfer with a peripheral device autonomously. Data transfer between the CPU and the CP 521 must always be started by the CPU by sending a job request to the CP 521.

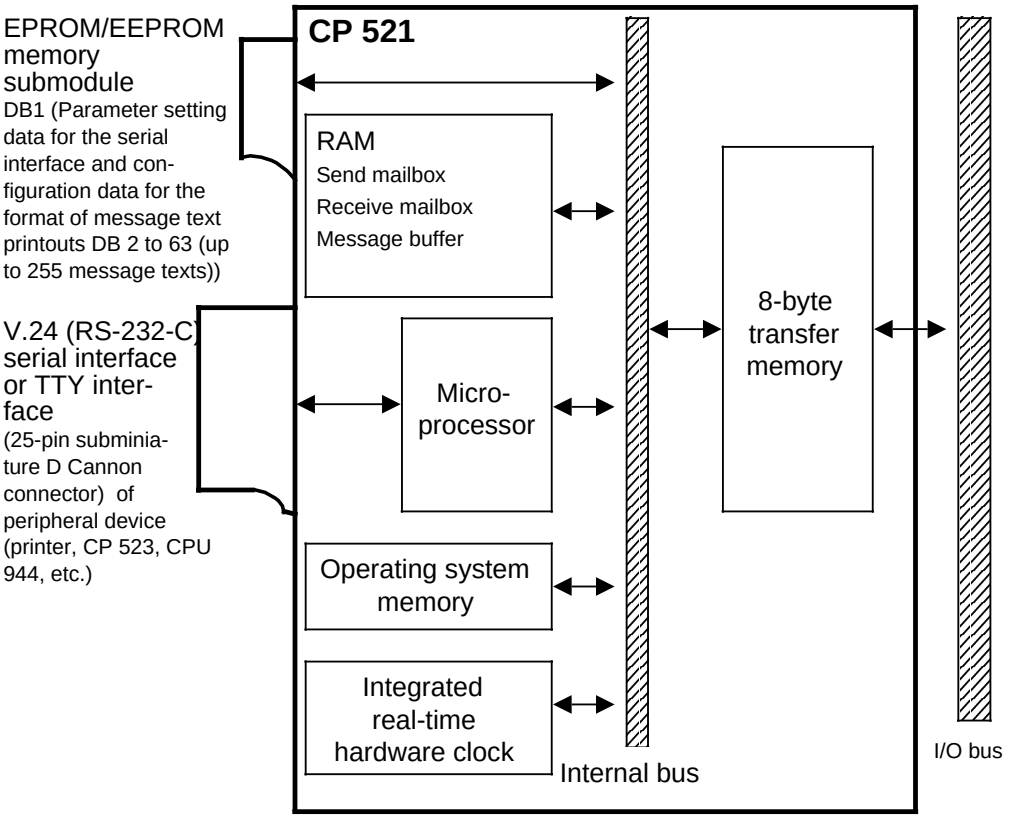


Figure 2-1. Schematic Representation of the CP 521

Typical jobs in print mode would be, for example:"Print message text No. 20" or, in ASCII mode: "Receive a message from a peripheral device.

Print mode

The "Print message text XY" job causes the message text stored in the memory submodule under the number XY to be transferred to the printer via the serial interface. Current values of variables, time of day and date can be entered in the message text during this process. You can configure up to 255 different message texts and store them in the memory submodule. You can also include in the message text control commands for the printer. These are then executed during printing (double-width type on/off).

ASCII mode

In ASCII mode, messages of up to 256 bytes in length can be transferred between the CPU and a peripheral device connected to the CP 521. On receipt of a "Send job" from the CPU, the CP 521 transfers the data from the PIQ (Process Image of the Outputs) of the CPU to a Send mailbox until the message has been completely transferred from the CPU. Then the CP 521 transfers the data autonomously from the Send maibox to the peripheral device.
Data received from the peripheral device is first stored by the CP in a Receive mailbox. From there it can be read into the PII (Process Image of the Inputs) with a "Receive job" from the CPU.

Serial interface

The CP 521 has a RS-232-C (V.24) interface and a passive TTY interface.
The following are examples of peripheral devices that you can connect to the serial interface of the CP 521:

- Keyboard
- Terminal
- CPU 944 (ASCII interface)
- CP 523
- another CP 521
- Printer with active TTY interface
- Printer with RS-232-C (V.24) interface

EPROM/EEPROM

You can use EPROM and EEPROM memory submodules with up to 8 Kbytes. The EPROM/EEPROM is configured with the programmer.

- Print mode

In print mode, the paramter setting data for the serial interface and for the format of the printout is defined on the memory submodule in data block (DB) 1. You can configure up to 255 different message texts in DBs 2 to 62.

Note:

The CP 521 can only be started up in print mode if a memory submodule containing at least one message text is plugged in.

- ASCII mode

You can define the paramter setting data for the serial interface on the memory submodule in DB 1.

Note:

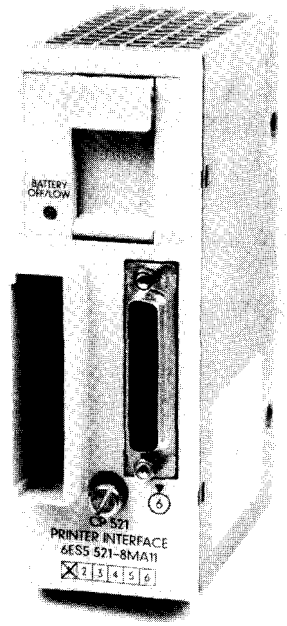
In ASCII mode, the CP 521 can also be started up without a memory submodule, in which case the parameter setting data for the serial interface is then transferred from the user program to the CP 521.

Integrated real-time hardware clock

The CP 521 has an integrated real-time hardware clock. The clock can be set and read either with a programmer or in the user program (2.5).

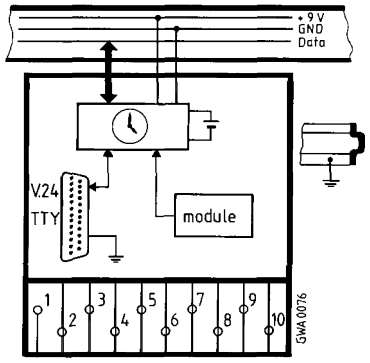
2.2 Technical Specifications

For environmental, mechanical and electromagnetic conditions see the programmable controller manual.



GWA 3963/6

Terminal assignment



Block diagram

Galvanic isolation	TTY signals are floating
Memory submodule	EPROM/EEPROM
Serial interface	V. 24(RS232C)/TTY passive
Transmission mode:	Asynchronous 7-bit mode = char frame of 11 bits (1 start bit, 7 data bits, 1 parity bits, 2 stop bits) 8-bit mode = char frame of 11 bits (start bit, 7 data bits, 1 parity bits, 2 stop bits)
Transmission rate	110 to 9600 baud
Permissible cable length - TTY (PT 88) - V.24 (RS-232-C)	 30m (98ft) 15 m (49 ft.)
Battery failure display (yellow LED)	yes
Back-up battery Lithium ½AA	 3,7 V / 850 mAh
Battery backup time	at least 1 year
Degree of protection	IP 20
Permissible ambient tern perature vertical - horizontal	 O to 60°C o to 40°C
Relative hum idity	15 to 95%
Current consumption from + 9 v (CPU)	typ. 140 mA
Power losses of the module	typ. 1.2 W
Weight approx.	500 g (1 .1 lb.)

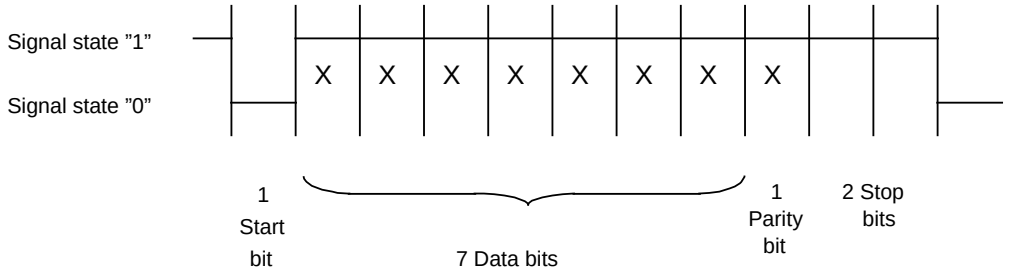
2.3 Serial Interface CP 521 - Peripheral Device

The CP 521 is equipped with a RS-232-C (V.24) interface and a passive TTY interface. You can set the serial interface parameters in two ways:

- In the user program (only significant in ASCII mode) with job 90H "Set interface parameters" parameter blocks 0 to 2 (6.3.1).
  The CP 521 copies the parameter setting data from the user submodule into the RAM area of the CP during startup. If the CPU transfers new parameter setting data to the CP with the 90H job during operation, the CP continues with the new data.
- On the user submodule in data block 1, parameter blocks 0 to 2 (6.3.1).

Data is transferred between the CP and the peripheral device in an 11-bit character frame. There are two data formats available.

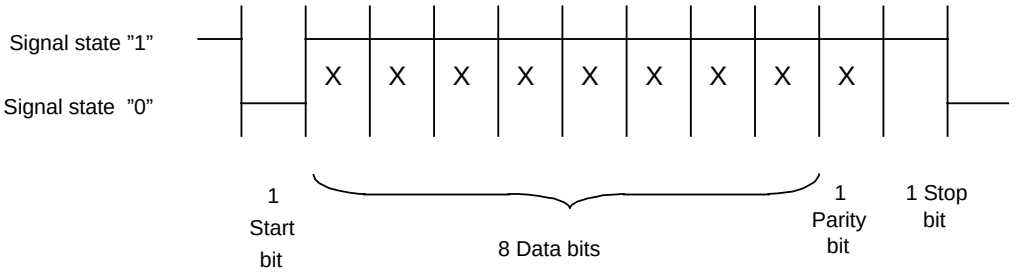
- 7 data bits: 1 start bit, 7 data bits, 1 parity bit, 2 stop bits



X = can assume signal state 0 or 1

Figure 2-2. 11-Bit Character Frame with 7 Data Bits

- 8 data bits: 1 start bit, 8 data bits, 1 parity bit, 1 stop bit



X = can assume signal state 0 or 1

Figure 2-3. 11-Bit Character Frame with 8 Data Bits

2.3.1 Print Mode

You have a choice of three methods of transmission in print mode.

Table 2-1. Methods of Transmission in Print Mode

Method of transmission	Line used TTY/RS-232-C (V.24)	Required parameters	Parameter block	Remarks
XON/ XOFF protocol	RXD	XON characters XOFF characters	2	Priority over BUSY signal and print without protocol
BUSY signal	RXD/DSR	-	0	Priority over print without protocol
without protocol	Not significant	Wait after - CR (carriage return) - LF (line feed) - FF (form feed)	1	Line break not detectable

All methods of transmission are possible both with the RS-232-C (V.24) interface and the TTY interface.

2.3.2 ASCII Mode

The possible methods of transmission depend on the following:

- Type of interface (TTY or RS-232-C (V.24))
- Whether handshaking has been configured when the RS-232-C (V.24) interface is used (handshake OFF or ON).
- Type of ASCII mode (transparent or interpretative)

TTY interface or RS-232-C (V.24) interface with handshake OFF

The CPU only uses the RXD line. If the CP receives data from the peripheral device, the latter must maintain logic "1"* on the RXD line, otherwise the CP 521 will send the "Permanent line break" error message (XD_H) to the CPU.

- Transparent ASCII mode

No XON/XOFF protocol is possible in transparent ASCII mode. Messages must be of fixed length. It is not possible to transmit messages with endidentifiers.

- Interpretative ASCII mode

If you have programmed an XON/XOFF protocol in ASCII mode, the CP sends 20 bytes to the peripheral device before the 1 Kbyte receive buffer overflows XOFF. The CP then only sends XON to the peripheral device again when the receive buffer contains more than 256 free bytes.
If the peripheral device sends more than 20 bytes after an XOFF from the CP, the message data already received is rejected. The CP sends the "Receive buffer overflow"error message (XB_H) to the CPU.

Note:

Immediately after startup, the CP 521 assumes that the communications parnter is ready to receive, even if no XON is sent.

* For the TTY interface logic "1" means: Current
For the RS-232-C (V.24) interface logic "1" means: U -3 V

RS-232-C (V.24) interface with handshake ON

The RS-232-C (V.24) interface of the CP 521 can handle the following signals in handshake ON mode:

Table 2-2. Valid Control Signals of the RS-232-C (V.24) Interface in Handshake ON Mode

Control Signal	State	Meaning
Outputs		
TXD		Send data CP holds send line at logic 1 when not transmitting (U -3 V).
DTR	ON OFF	**D**ata **T**erminal **R**eady CP 521 switched on; ready to receive CP 521 not switched on; not ready to receive
RTS	ON OFF	**R**equest **t**o **s**end CP 521 in send mode CP 521 not sending
Inputs		
RXD		Receive data Receive line must be held at logic 1 by the peripheral device (U -3 V).
DSR	ON OFF	**D**ata **s**et **r**eady Peripheral device switched on; ready to receive Peripheral device not switched on; not ready to receive
CTS	ON	**C**lear **t**o **s**end Peripheral device can receive characters from the CP 521 The CP expects this as reply to RTS = ”ON”
	OFF	Peripheral device cannot receive characters from the CP 521

Note:

An XON/XOFF protocol is not possible if you evaluate the control signals (handshake ON).

Data transfer between the CP and peripheral devices takes place as follows:
The CP 521 sets the "DTR" output after startup. This indicates that it is ready to operate and receive.

Example: The CP wants to send data

1. CP waits till DSR = ON
   If the peripheral device still has not set DSR to ON after 20 s. the CP reports the error to the CPU (peripheral device not ready).
2. CP sets RTS
3. CP waits till CTS = ON
   If the peripheral device still has not set CTS to ON after 20 ms, the CP reports the error to the CPU (peripheral device not ready).
4. CP sends data
5. CP resets RTS after sending data
6. Peripheral device sets CTS to OFF

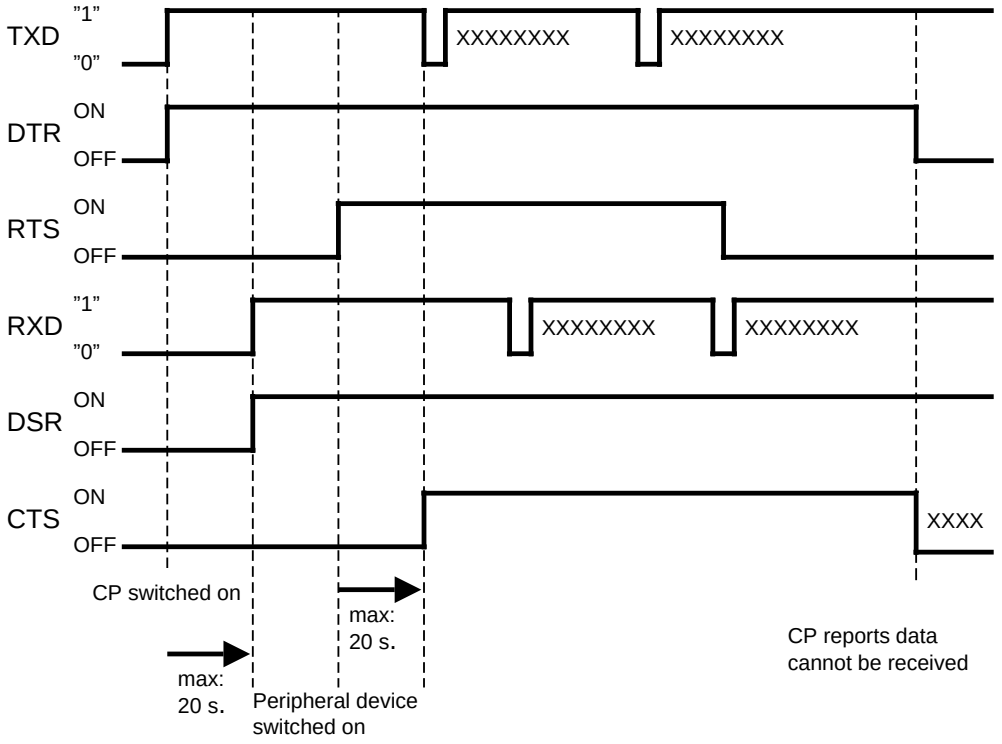


Figure 2-4. Timing Diagram for Data Transfer Between the CP and the Peripheral Device

Example: Peripheral device wants to send data

Peripheral device only checks the DTR line
If ON: data may be sent (CP ready to receive)
If OFF: Sending of data must be interrupted (CP not ready to receive)

2.4 Memory submodule

In print mode, you require a user submodule which you must configure with the following data in off-line mode using a programmer:

- Message texts in DB 2 to 63
  This DB is independent of the DBs stored in the CPU. The DBs in the user submodule and the DBs in the CPU may have the same numbers.
- Printer interface parameters and configuration data for the message texts in DB 1

You can use two types of user submodule:

- EPROM user submodule
  A UV erasing facility is required for erasing the submodule contents.
- EEPROM user submodule
  Submodule contents can be overwritten using a programmer.

Table 2-3. Overview of Permissible User Submodules

Submodule type	Order No.	Capacity
EPROM	6ES5 375 - 0LA15	8KByte
EEPROM	6ES5 375 - 0LC31	8KByte
	6ES5 375 - 0LC11	2KByte

Maximum useful memory submodule size is 8 KBytes.

Note:

The memory submodule of the CP 521 may only be plugged in or unplugged in the POWER OFF state.

2.5 Hardware Clock

The CP 521 incorporates a hardware clock, which may be provided with battery backup.
Sthe date (date, month, year) and the time of day (second, minute, hour) can be output with a message text on the printer. In this way, for example, the time of failure of a machine can be recorded. The output format for the date and time of day is defined on the memory submodule in DB 1, parameter block 4 (6.3.3).
The clock is read from the control program and can be set with the ”FORCE VAR” programmer function (6.7.2).
Otherwise the clock has the default values 1.1.89 12.00.00. on power-up.
The clock must be reset if no battery was plugged in to back-up the CP 521 during POWER OFF.
A battery can only be changed in the POWER ON state, otherwise the clock data is lost and the clock must be reset.

2.6 Backup Battery

The battery backs up only the clock data.
In the case of a mains power failure or if the PC is switched off, the clock data is only retained when a backup battery is inserted in the CP 521-
the battery should only be inserted and changed when the PC is switched on.
Otherwise new clock data must be entered on power-up.
A new lithium battery has a life of at least one year.

Note:

Battery failure is indicated by the yellow LED and signalled to the CPU in byte 0 ”Module status”.

C A U T I O N

Lithium batteries cannot be charged. -Danger of explosion!-
Used batteries must be properly disposed of.

2.7 Accessories List and Ordering Data

Memory submodules

(EPROM) memory submodule, 8 Kbytes	6ES5 375-0LA15
(EEPROM) memory submodule, 2 Kbytes	6ES5 375-0LC11
(EEPROM) memory submodule, 8 Kbytes	6ES5 375-0LC31

Printer connecting cable

You can use the same printer connecting cable that you use for connecting the programmer (PG) to the PT printer.

PT 88 RS-232-C (V.24)	6ES5 735-2BD20
PT 88 TTY	6ES5 736-0BD20

Printers

See S5-11U Programmable Controller Catalog	ST 52.1

Programmers

See Programmer Catalog	ST 59

Backup battery

Backup battery, lithium AA; 3.6 V / 850 mAh	6ES5 980-0MA11

3 Hardware Installation

3.1 Mechanical Assembly . . . . . 3 - 1

3.2 Dimension Drawing . . . . . 3 - 2

3.3 Connecting Peripheral Devices . . . . . 3 - 2

3.4 Protective Measures Against Faults on the Line . . . . . 3. - 4

3.5 Typical Configurations . . . . . 3 - 5

3.5.1 Print mode . . . . . 3 .- 5

3.5.2 ASCII Mode . . . . . 3 .- 7

Figures

3-1. Setting the Coding Element . . . . . 3 - 1
3-2. Dimension Drawing of the CP 521 . . . . . 3 - 2
3-3. Pin Assignment of the 25-pin Subminiature D Connector . . . . . 3 - 3
3-4. Pin Assignement of the TTY Interface without BUSY Signal (Print Mode) . . . . . 3 - 5
3-5. Pin Assignment of the TTY Interface with BUSY Signal (Print Mode) . . . . . 3 - 6
3-6. Pin Assignment of the RS-232-C (V.24) Interface (Print Mode) . . . . . 3 .- 6
3-7. Pin Assignment: CP 521 (TTY Passive) - CP 523 (TTY Active) (Point-to-Point Connection) . . . . . 3 - 7
3-8. Pin Assignment: CP 521-CP 521 with Handshaking . . . . . 3 - 8

3 Hardware Installation

3.1 Mechanical Assembly

. The module may only be plugged in or unplugged when the PC is in the "STOP" mode.
. The module may only be plugged into slots O to 7 on a bus unit (slot 7 is not permitted with the CPU 102, 6ES5 102-8MAO1 and the CPU 103, 6ES103-8MA01).
. Before assembling the module, the coding element of the bus unit must be set to 6.

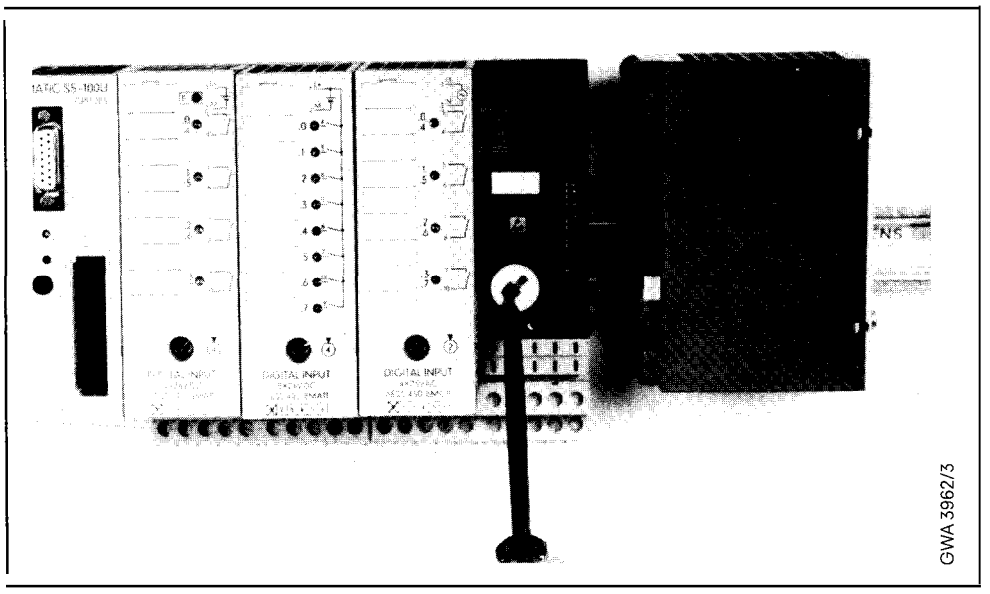


Figure 3-1. Setting the Coding Element

- The memory submodule may only be plugged in and unplugged when the power is OFF.

- The connection to the peripheral device (Sub D Cannon male connector) may only be plugged in and unplugged when the CPU is in STOP mode and data transfer between the CP and the peripheral device has been terminated.

Installing module:

- Hook in the module at the top of the bus unit,
- Swing it down onto the bus unit
- Press it firmly down
- Tighten the screw at the front.

3.2 Dimension Drawing

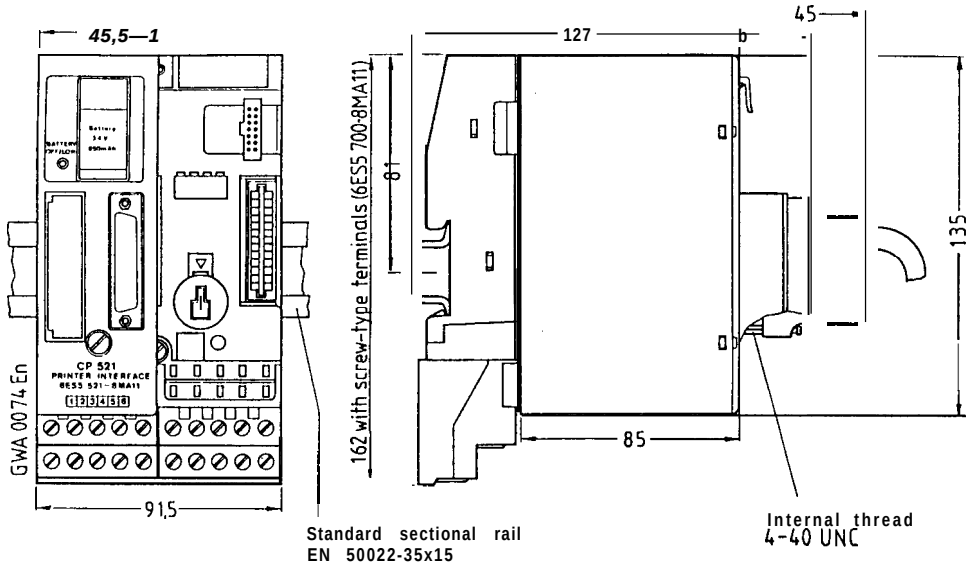


Figure 3-2. Dimension Drawing of the **CP** 521

3.3 Connecting Peripheral Devices

The module has a serial interface. Parameters can be assigned (→ 6) to set the interface as a TTY interface and RS-232-C (V.24) interface. Both interfaces are connected to a 25-pin subminiature D connector.

Pin assignment of the 25-pin subminiature D connector

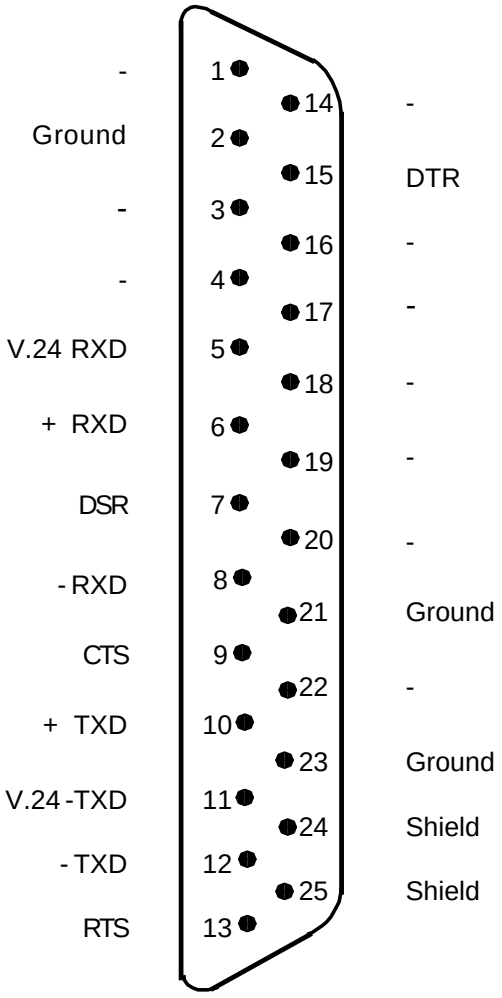


Figure 3-3. Pin Assignment of the 25-pin Subminiature D Connector

Permissible cross-section of the lines used in subminiature D connections: up to 0.5 mm^2 (20 AWG)

3.4 Protective Measures Against Faults on the Line

Please note the cable length to ensure trouble-free data transfer.

TTY interface
Note voltage drops along the cable and the send and receive elements of the module (2.2)

RS-232-C (V.24) interface
Cable lengths up to 15 m are not critical. Note the loading capacity in the case of longer cables. As a general rule, longer cables can be used by reducing the baud rate.

3.5 Typical Configurations

This section contains a few typical configurations for operating the CP 521 in print and ASCII mode.

3.5.1 Print mode

In print mode, the CP 521 assumes a printer at the I/O interface with an RS-232-C (V.24) interface or an active TTY interface for printing message texts.

Terminal Diagram of the TTY Interface without BUSY Line

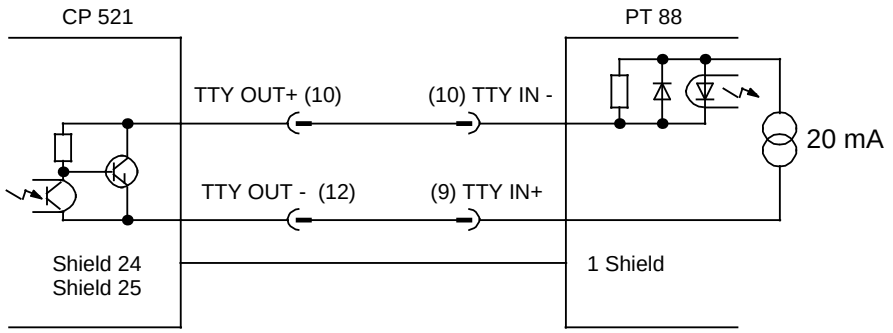
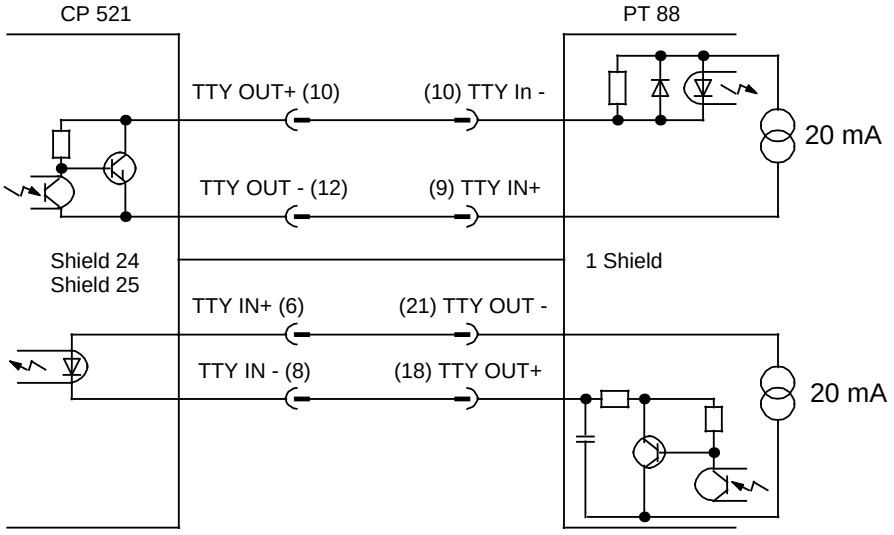


Figure 3-4. Pin Assignment of the TTY Interface without BUSY Signal (Print Mode)

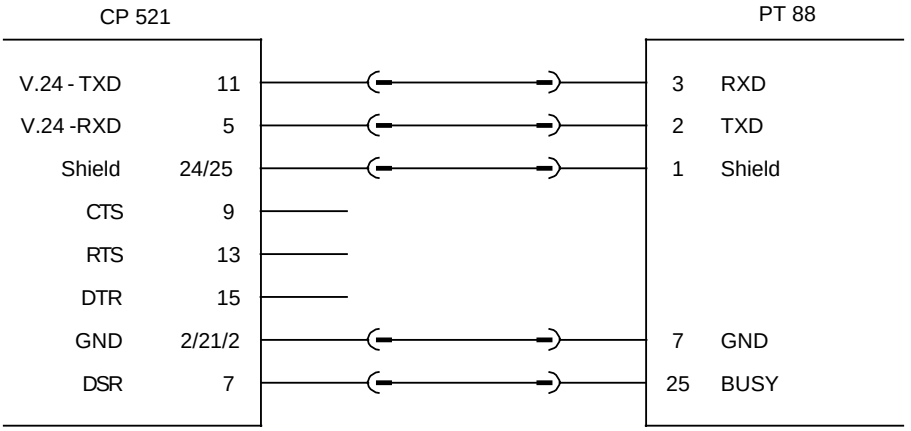
Terminal Diagram of the TTY Interface with BUSY Line



Printer setting: Printer not ready to receive = no current

Figure 3-5. Pin Assignment of the TTY Interface with BUSY Signal (Print Mode)

Terminal Diagram of the RS-232-C (V.24) Interface



Printer setting: Printer not ready to receive = negative potential

Figure 3-6. Pin Assignment of the RS-232-C (V.24) Interface (Print Mode)

3.5.2 ASCII Mode

The CP 521 assumes a peripheral device with an RS-232-C (V.24) or active TTY interface.

Terminal diagram of the CP 521 (TTY passive) - CP 523 (TTY active)

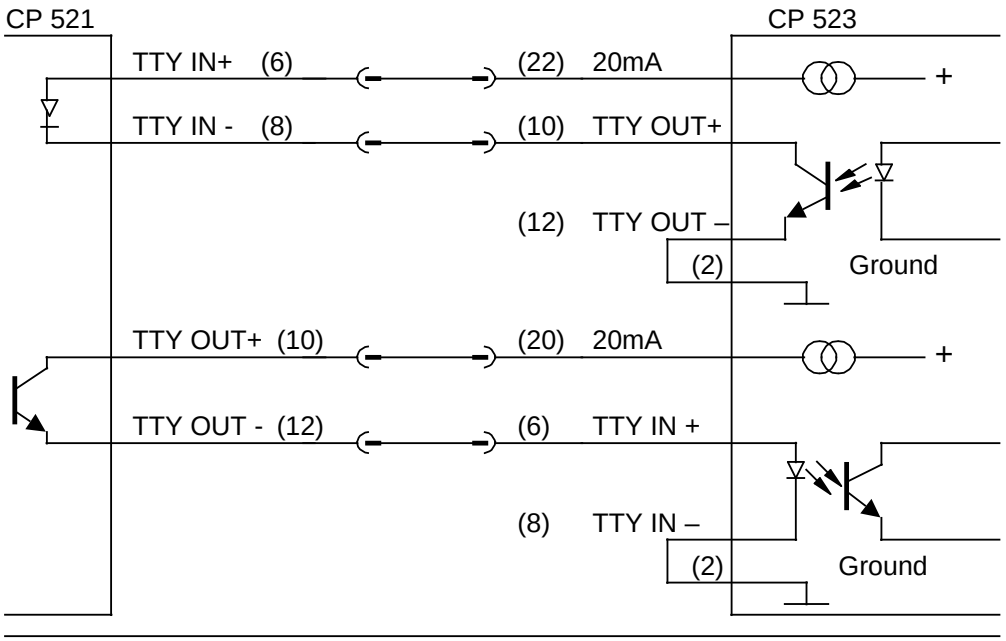


Figure 3-7. Pin Assignment: CP 521 (TTY Passive) - CP 523 (TTY Active) (Point-to-Point Connection)

Terminal Diagram of the CP 521 - CP 521 (RS-232-C (V.24))

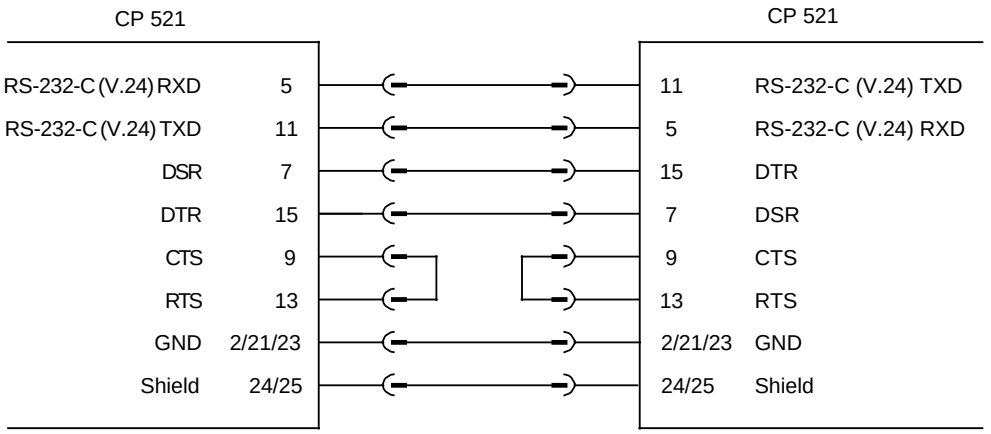


Figure 3-8. Pin Assignment: CP 521-CP 521 with Handshaking

Note:

If you use the RS-232-C (V.24) interface, the CP 521 cannot detect a "Permanent wire break". You are therefore recommended to set handshake ON.

4 Notes on Operation

4.1 Restart Characteristics . . . . . 4 - 1
4.1.1 Checking the Module . . . . . 4 - 1
4.1.2 Memory Submodule Evaluation . . . . . 4 - 2
4.1.3 Clock Test . . . . . 4 - 2

4.2 Behavior in Operation . . . . . 4 - 3
4.2.1 Print Mode . . . . . 4 - 4
4.2.2 ASCII Mode . . . . . 4 - 5

Tables

4-1. Module Errors . . . . . 4 - 1
4-2. Memory Submodule Errors . . . . . 4 - 2
4-3. Clock Test Messages . . . . . 4 - 3
4-4. Faults in Operation (Print Mode) . . . . . 4 - 4
4-5. Interruptions in Operation (ASCII Mode) . . . . . 4 - 5

4 Notes on Operation

4.1 Restart Characteristics

The CP 521 performs a restart operation when the power supply is restored (POWER ON).

The restart procedure consists of

- Deleting the message buffer
- Checking the module (4.1.1)
- Memory submodule evaluation (4.1.2)
- Clock test (4.1.3)

If errors occur during the restart procedure, the module sends an error message to the CPU in byte 0 of the PII* "Module status" (print mode 6.6, ASCII mode 7.5.3).

4.1.1 Checking the Module

In this part of the restart procedure, the proper functioning of the hardware is checked. Errors are provided with the appropriate number in byte 0 of the PII "Module status" and can be evaluated in the control program or with the programmer.

Table 4-1. Module Errors

Error number in byte 0	Error	Remedy
$1X_H$	Clock fault	Replace module
$8X_H$	Hardware fault	Replace module
$X7_H$	No backup battery	Insert new battery

X: may have different values

4.1.2 Memory Submodule Evaluation

The parameters of the serial interface and message texts specified on the module are checked here. Errors with the appropriate number in byte 0 of the PII "Module status" can be evaluated in the control program.

Table 4-2. Memory Submodule Errors

Contents of byte 0 of the PII	Error	Remedy
$X1_H$	Memory submodule fault or no memory submodule plugged in	1. PC POWER OFF 2. Plug in (new) memory submodule 3. PC POWER ON
$X2_H$	No message texts configured	Configure messages. - Wrong or defective memory submodule plugged in?

X: The other half of the byte may have different values; not significant here

4.1.3 Clock Test

This part tests the module's real-time clock. The clock is set when the battery backup fails during POWER OFF of the PC. The clock has the default setting 12.00.00 01.01.89.
Errors and the default setting are provided with the appropriate number in byte 0 of the PII "Status of the module" and can be evaluated in the control program.

Table 4-3. Clock Test Messages

Error number in byte 0	Message	Remedy
$1X_H$	Clock fault	Replace module
$2X_H$	Clock set Default setting	Set the correct time with the programmer "FORCE VAR" or "FORCE" function

X : May have different values

4.2 Behavior in Operation

You must note the following if you want to modify the configuration during operation:

- The CP 521 can only be plugged in or unplugged when the CPU is in STOP mode
- The memory submodule can only be plugged in or unplugged when the power is OFF
- To guarantee trouble-free printing, make sure the 25-pin subminiature D female connector is only unplugged when the CPU is in STOP mode and data transfer between the CP and the printer has been completed.

When the power is switched off, the battery provides backup for the clock data.

4.2.1 Print Mode

Operation in print mode may be interrupted for various reasons. The following table lists the effects of faults.

Table 4-4. Faults in Operation (Print Mode)

Fault	Comment	Effect
PC stops		Activated print jobs are completed.
POWER OFF (CPU)		Activated print jobs are not continued.
	Battery backup available	Clock data is retained
	not available	Clock data is lost
25-pin submin. D connector unplugged or cable fault	- BUSY line available and BUSY signal configured or - XON-/XOFF protocol	Activated print jobs are completed after restoration of the connection. Perfect print quality is not guaranteed (e.g. smudged characters).
	otherwise	Activated print jobs are not continued after restoration of the connection.
POWER OFF (printer)	BUSY line available and BUSY signal configured	Activated print jobs are completed after restoration of the connection. Perfect print quality is not guaranteed (e.g. smudged characters).
	otherwise	Activated print jobs are not continued after restoration of the connection.
	XON/XOFF protocol configured	Activated print jobs are completed after restoration of the connection.
	not configured	Activated print jobs are not continued after restoration of the connection.

4.2.2 ASCII Mode

Operation may be interrupted for various reasons in ASCII mode. The following table lists the effects of the faults.

Table 4-5. Interruptions in Operation (ASCII Mode)

Fault	Comment	Effect
CPU stops		Data traffic continues between CP and peripheral device. This can lead to overflow of the Receive mailbox.
POWER OFF (CPU)	Battery backup available not available	Send and receive message data are lost Clock data is retained Clock data is lost
Fault in CP-peripheral or POWER OFF (peripheral device)		Data is corrupted during transmission (in both directions) CP error message • Character delay time error after character time out • Peripheral device not ready after 20s

Note:

If you operate the RS-232-C (V.24) interface without handshaking (Handshake OFF), the CP 521 cannot detect a break in the connection between the CP and the peripheral device.

5 Principle of Operation and Addressing

5-1 Monitoring of the CP 521 by the CPU . . . . . 5 - 1

5-2 Addressing . . . . . 5 .- 3

5.2.1 Slot Addressing . . . . . 5 . - 3

Figures

5-1. General Functional Description . . . . . 5 - 2

Tables

5-1. Slot Addresses . . . . . 5 - 3

5 Principle of Operation and Addressing

This section describes how the CPU monitors the data traffic of the CP 521 and how the printer output module is addressed.

5.1 Monitoring of the CP 521 by the CPU

Every data interchange between the CP 521 and the CPU must be started by the CPU sending a job to the CP 521. The CPU controls the operations of the CP 521 as follows:

- Print mode:
  In print mode, the CP 521 outputs messages to a printer autonomously when instructed to do so by the CPU. The program continues to run independently of message text output. The CPU receives data from the CP 521 on the status of the CP 521 and the printer. In addition, the CP 521 tells the CPU if it can handle the jobs requested by the CPU.

- ASCII mode:
  In this mode, the CP 521 enables the exchange of data between the CPU and a peripheral device. The CPU reads data from the CP 521 into the PII and instructs the CP 521 to pass on data to a terminal device (terminal, keyboard, etc.) or to another CP 521.

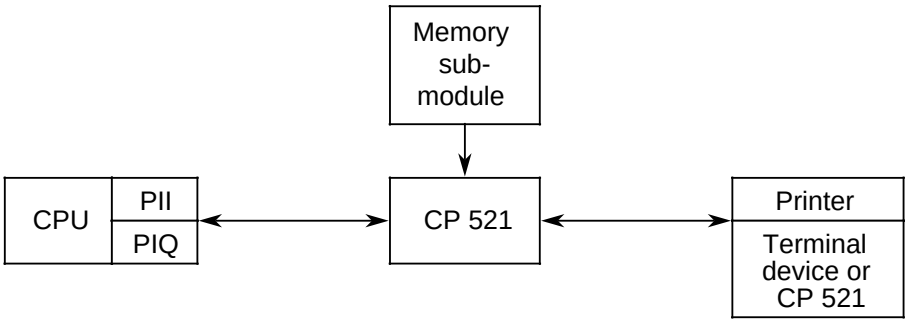


Figure 5-1. General Functional Description

The control program controls the CP 521 via the I/O bus and evaluates its data.

The I/O bus transfers 64 data bits (8 bytes) from the CPU to the CP 521 and 64 data bits (8 bytes) from the CP 521 to the CPU.

The CP 521 has three interfaces:
- An interface to the I/O bus
- Passive TTY and RS-232-C (V.24) interface to the peripheral device
- A receptacle for a memory submodule

The serial interface transfers data between the module and the peripheral device.

Compatible peripheral devices:
- Keyboard
- Terminal
- A further CP 521
- CPU 944 (ASCII interface)
- CP 523
- Printer with active TTY interface
- Printer with RS-232-C (V.24) interface

The memory submodule is only required in print mode and contains the parameters for the printer interface and the message texts. For permissible memory submodules see list of accessories (2.7).

5.2 Addressing

The CP 521 is addressed like a 4-channel analog module. Please note the following:

- The module may only be plugged into slots 0 to 7 (slot 7 is not permitted with CPU 102, 6ES5 ((usw.)) and CPU 103 6ES5 ((usw.)).
- The address space extends from 64 to 127.
- Eight bytes are reserved per slot in the process input image (PII) and in the process output image (PIQ).

5.2.1 Slot Addressing

Data from the module to the CPU is written into the specified address space of the PII. Data from the CPU to the module is stored in the same addresses in the PIQ.

Table 5-1. Slot Addresses

Slot	0	1	2	3	4	5	6	7
Addresses PII/PIQ	**64** ... 71	**72** ... 79	**80** ... 87	**88** ... 95	**96** ... 103	**104** ... 111	**112** ... 119	**120** ... 127
	Initial address of a slot							

The eight bytes reserved per slot (0 to 7) have a fixed meaning. The byte numbers given in this manual always refer to the initial address of the slot in each case. Consequently, in your control program, you must add the initial address of the slot containing the module to the byte numbers specified.

Example: Module in slot 3: byte 2 has address 90
Module in slot4: byte 2 has address 98

Byte 0 of the PIQ "Job" defines the meaning of bytes 1 to 7 in the PIQ. See Chapters 6 and 7 for more detailed information on the meaning of the bytes in print mode and in ASCII mode.

6 Print Mode

6.1 General . . . . . 6 - 1

6.2 Prerequisites for Operating in Print Mode . . . . . 6 - 1

6.3 Setting the CP 521 Parameters in Print Mode. . . . . 6 - 4
6.3.1 Setting the Serial Interface Parameters (Parameter Blocks 0, 1 and 2) . . . . . 6 - 8
6.3.2 Configuration Data for Entering Message Texts (Parameter Block 3) . . . . . 6 - 11
6.3.3 Configuration Data for Message Text Printout (Parameter Blocks 4 to 6) . . . . . 6 - 12
6.3.4 Setting the Mode Parameters (Parameter Block 7) . . . . . 6 - 15
6.3.5 Parameter Setting Example for the CP 521 in Print Mode . . . . . 6 - 16
6.3.6 Parameter Setting with the PG 615 Programmer . . . . . 6 - 18

6.4 Configuring Message Texts . . . . . 6 - 19
6.4.1 Structure of a Message Text . . . . . 6 - 19
6.4.2 Entering Place Holders . . . . . 6 - 22
6.4.3 Place Holders for Date and Time of Day . . . . . 6 - 24
6.4.4 Place Holders for Message Text . . . . . 6 - 25
6.4.5 Place Holders for Variables . . . . . 6 - 28
6.4.6 Control Parameters . . . . . 6 - 32

6.5 Data Transfer from the CPU to the CP 521 . . . . . 6 - 34
6.5.1 Data from the CP 521 to the CPU (Status of the CP 521, the printer and current clock data) . . . . . 6 - 34
6.5.2 Data from the CPU to the CP 521 (CPU Jobs) . . . . . 6 - 40

6.6 Overview of Possible Jobs in Print Mode . . . . . 6 - 41
6.6.1 Printing the Message Text (Job 30_H und 40_H) . . . . . 6 - 45
6.6.2 Setting the Time of Day and the Date (Job10_H) . . . . . 6 - 50
6.6.3 Setting the Page Numbers (Job 20_H) . . . . . 6 - 52

6.6.4 Outputting Page Feed (Job 50_H) . . . . . 6 - 53
6.6.5 Outputting Line Feed (Job 60_H) . . . . . 6 - 54
6.6.6 Deleting the Message Buffer (Job 70_H) . . . . . 6 - 55
6.6.7 Printout of All Configured Message Texts
(Job 80_H) . . . . . 6 - 56
6.6.8 Transferring Parameter Setting Data
(Job $90XX_H$) . . . . . 6 - 57

Figures

6-1. Schematic for Entering Parameter Blocks in DB 1 . . . . . 6 - 7
6-2. Schematic for Entering Headers and Footers . . . . . 6 - 14
6-3. Structure of a Message Text . . . . . 6 - 19
6-4. Structure of a Place Holder . . . . . 6 - 22
6-5. Job Transfer and Error Message Evaluation . . . . . 6 - 40

Tables

6-1. Contents of the Parameter Blocks . . . . . 6 - 5
6-2. Parameter Setting Data for the Serial Interface (Parameter Block 0) . . . . . 6 - 8
6-3. Parameter Setting Data for the Serial Interface (Parameter Blocks 1 and2) . . . . . 6 - 10
6-4 Configuration Data for Entering Message Texts (Parameter Block 3) . . . . . 6 - 11
6-5. Configuration Data for Message Text Printout (Parameter Blocks 4 to 6) . . . . . 6 - 12
6-6. Page Format . . . . . 6 - 13
6-7 Operating Modes and Parameters for ASCII Mode (Parameter Block 7) . . . . . 6 - 15
6-8. Overview of Place Holders Configurable in Message Texts . . . . . 6 - 22
6-9. Data Formats for Variables . . . . . 6 - 29
6-10. Typical ”KT” Data Format Printouts . . . . . 6 - 30
6-11. Typical ”KFa,b” Data Format Printouts . . . . . 6 - 31
6-12. Data from the Module to the CPU . . . . . 6 - 34
6-13. Status of the Module in Print Mode (Byte 0 in the PII) . . . . . 6 - 35
6-14. Status of the Printer and Day of the Week . . . . . 6 - 37
6-15. Date and Time of Day . . . . . 6 - 38
6-16. Permissible Jobs in Print Mode (Byte 0 in the PIQ) . . . . . 6 - 41
6-17. Transfer of Additional Input in the Case of the ”Print Message Text” Job . . . . . 6 - 42
6-18. Transfer of Additional Inputs in the Case of the ”Set Clock” Job . . . . . 6 - 43

Tables

6-19. PIQ in the Case of the ”Print Message Texts” . . . . . 6 - 46
6-20. PIQ in the Case of the ”Set Clock” Job . . . . . 6 - 51
6-21. PIQ in the Case of the” Setting the Serial Interface Parameters” Job . . . . . 6 - 58
6-22. PIQ in the Case of the”Setting the XON/XOFF listing parameters” Job . . . . . 6. - 59
6-23. PIQ in the Case of the ”Setting Print Mode” Job . . . . . 6 - 59
6-24. PIQ in the Case of the ”Transparent ASCII Mode” Job . . 6 - 60
6-25. PIQ in the Case of the ”Interpretive ASCII Mode” Job . . . 6 - 61

6 Print Mode

6.1 General

The CP 521 enables the output of message texts to a printer in print mode:

- Output of message texts defined by you in the memory submodule in data blocks (DB) 2 to 63.
- Insertion of the current time of day and date in the message text to be printed out. The clock data is supplied by the integrated real-time clock.
- Insertion of current variable in the printout.
  The variable values can be transferred from the CPU to the CP 521.

6.2 Prerequisites for Operating in Print Mode

The following conditions must be met in order to operate the CP 521 in print mode:

1. Printer settings

The printer settings must agree with the your configured settings in the memory submodule in the parameter blocks of DB 1. If you have set a baud rate of 1200 baud on the printer, for example, you must set the same baud rate in DB 1.

Note:

Set the parameters for the serial interface on your printer as desired and then set the parameters in DB 1 in the memory submodule.

- See 3.5.1 for configuration examples with terminal diagrams.
- Parameter setting in DB 1 is explained in 6.3.

2. Setting the CP 521 parameters

Store all parameter setting data in the memory submodule in DB 1:
- Parameters for the printer interface (baud rate, type of interface, BUSY signal, ...)
- Configuration data for entering message texts (function character, end-of-text character)
- Configuration data for the message text printout (output form for date and time of day, headers and footers, ...)

Set the memory submodule parameters using a programmer in off-line mode.

3. Configuring message texts

You must configure at least one message text in one of the DBs 2 to 63 in the memory submodule.

Note:
If you have configured a message text in a DB, you can configure further message texts later in other DBs and store them also in the memory submodule.
See 6.4 for an explanation of message text configuration.

4. Establishing connections

You must take the following measures when the CPU is in the POWER OFF state:
- Establish a connection between the CP 521 and the printer.
- Plug the configured memory submodule into the CP 521.

You can then switch the CPU on (POWER ON).

5. Startup of the CP 521 in print mode

If the CP 521 has a memory submodule, it is automatically in print mode after POWER ON.
If the CP 521 has no memory submodule, it is automatically in ASCII mode.

Changing operating modes (during operation)

You can change betwenn ”Print mode”, ”Transparent ASCII mode” and ”Interpretive ASCII mode” during operation.

Note:
Note that if you
- change the operating mode

or
- call up the programmed operating mode again

the send and receive buffers and all print jobs will be deleted.

To change the operating mode, transfer job $90XX_H$ "Transfer configuration data" to the CP 521.
Remember that the parameters must agree with the settings on your peripheral device.

6.3 Setting the CP 521 Parameters in Print Mode

The CP 521 can only be operated in print mode if an initialized memory submodule has been plugged in. Store the following in the memory submodule:

- Parameter setting data of the CP 521 in data block (DB) 1
- Message texts, optionally in DBs 2 to 63 (6.4)

The CP 521 requires the following parameter setting data:

- Parameters for the printer interface (baud rate, type of interface, BUSY signal, ...)
- Configuration data for entering message texts (function character, end-of-text character)
- Configuration data for the message text printout (output form for date and time of day, headers and footers, ...)

The CP 521 has default values for all parameters. If you want to use other values, you must store your parameter setting data in a memory submodule in DB 1. This parameter setting data is subdivided into seven parameter blocks to simplify input of parameter data as much as possible.

Note:
DB 1 is only required if the parameter setting data deviates from the default values.

Table 6-1. Contents of the Parameter Blocks

Parameter Block Number	Contents
	Setting the serial interface parameters:
0	Serial interface parameters
1	Waiting times after "CR", "LF" and "FF"
2	XON/XOFF protocol The XON/XOFF Protocol has priority over the BUSY signal
	Parameter setting data for entering message texts:
3	Text parameters
	Parameter setting data for message text printout:
4	Output form for date and time of day
5	Page format
6	Headers and footers
	Setting the mode:
7	Parameters for ASCII mode

Dividing the parameters into seven parameter blocks has the advantage that you only need to assign parameters to the block which deviates from the default values. However, this parameter block must be entered complete even if individual parameters within the block are to retain their default values. If the value entered exceeds the limit value, this value is automatically replaced by the limit value.

- The low limit is used if the value entered is below this value.
- The high limit is used if the value entered exceeds this value.

Example: Example Input: Lines per page ”10_H” is replaced by ”20_H”
Input: left margin "80_H" is replaced by "$3C_H$"

Entering the parameter blocks

You must first enter all the parameter blocks you want to change in DB 1 using the programmer. Then transfer DB1 to the memory submodule.

Note:
Proceed as follows to change a DB in the memory submodule:

1. Transfer total contents of the memory submodule from the memory submodule to the programmer (diskette, hard disk)
2. Delete memory submodule
3. Change DB
4. Transfer total memory contents back to the memory submodule

Further tips for input:

- When entering text, change the data format from KS to S and vice versa in each line.
- You can enter comments in addition to the parameter blocks.

Schematic for entering of parameter blocks

The parameter blocks can be entered in DB 1 according to the following schematic:

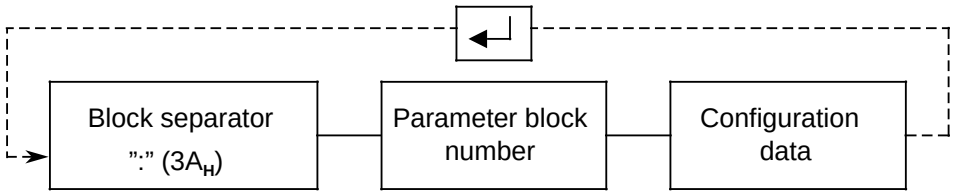


Figure 6-1. Schematic for Entering Parameter Blocks in DB 1

Note:
Parameter blocks 0, 2 and 7 can also be initialized in the user program. Parameters set in the user program have priority over those set in the memory submodule (7.3.1). The parameter setting data in the memory submodule is valid after POWER ON as long as it is not overwritten by the user program.

6.3.1 Setting the Serial Interface Parameters (Parameter Blocks 0, 1 and 2)

Store the data for the serial interface in parameter blocks 0, 1 and 2.

Table 6-2. Parameter Setting Data for the Serial Interface (Parameter Block 0)

Block	Meaning		Value Range	Default Values on the CP 521
0	Baud rate	110 Bd	1	
		200 Bd	2	
		300 Bd	3	
		600 Bd	4	4
		1200 Bd	5	
		2400 Bd	6	
		4800 Bd	7	
		9600 Bd	8	
	Parity	even	0	0
		odd	1	
	(Parity bit always ”1”)	”mark”	2	
	(Parity bit always ”0”)	”space”	3	
		none	4	
	BUSY signal	no	0	0
		yes	1	
	Interface	TTY	0	0
		RS-232-C (V.24)	1	
	Data format	7 data bits	0	0
		8 data bits	1	
	Hardware handshaking	OFF	0	0
		ON	1	

Baud rates

You have a choice of eight baud rates. The default is 600 baud. If you use the RS-232-C (V.24) interface, you must take the load capacity of cables longer than 15 m into account. Longer cables can be used in general if the baud rate is reduced.

Parity

You have a choice of five types of parity. Even parity is the default.

- Even parity
  The parity bit is set so that the sum of the data bits that are "1" (incl. parity bit) is even.
- Odd parity
  The parity bit is set so that the sum of the data bits that are "1" (incl. parity bit) is odd.
- ”Mark”
  The parity bit is always has signal state "1".
- ”Space”
  The parity bit is always has signal state "0".
- No parity
  The signal state of the parity bit is not significant. Parity is not checked.

BUSY Signal

The printer generates the BUSY signal and reports when it is ready to receive.
The RS-232-C (V.24) and TTY interfaces allow you to run a BUSY line and to evaluate the BUSY signal.
The BUSY signal is only relevant when operating the CP 521 in print mode without XON/XOFF protocol.
The waiting times for "CR", "LF" and "FF" are not significant when evaluating the BUSY signal.

Interface

You can chose between the RS-232-C (V.24) and TTY interfaces here. See 2.3 for the characteristics of the interfaces.
The TTY interface is the default.

Data format

Characters are transmitted between the CP and the peripheral device in an 11-bit character frame. You can choose between seven and eight data bits within the character frame:

- 7 data bits: 1 start bit, 7 data bits, 1 parity bit, 2 stop bits
- 8 data bits: 1 start bit, 8 data bits, 1 parity bit, 1 stop bit

The default is 7 data bits.

Hardware handshaking

This parameter is only significant in conjunction with the RS-232-C (V.24) interface.
If you set hardware handshaking ON, the "RTS", "CTS", "DTR" and "DSR" control signals of the RS-232-C (V.24) interface are evaluated. Hardware handshaking has priority over the XON/XOFF protocol.
Hardware handshaking OFF is the default, i.e. the control signals are not evaluated.

Table 6-3. Parameter Setting Data for the Serial Interface (Parameter Blocks 1 and 2)

Block	Meaning	Value Range	Default Value on the CP 521
1	Waiting time after CR LF FF	(00_H to FF_H) · 25ms (00_H to FF_H) · 25ms (00_H to FF_H) · 25ms	$0A_H$: 0,025 s $0A_H$: 0,025 s $A0_H$: 4 s
2	XON character XOFF character	ASCII character (00_H to $7F_H$) ASCII character (00_H to $7F_H$)	FF_H (No XON / XOFF - protocol)

Parameters for waiting times

These parameters are only significant if you are not using an XON/XOFF protocol in print mode and if you are not evaluating the BUSY signal.
You can set waiting times for Carriage Return (CR), Line Feed (LF) and Form Feed (FF) in a time frame of 0.25 s.

Parameters for XON/XOFF character

If you have an XON/OFF protocol, you have free choice of XON/XOFF characters. Code 11_H (DC1) is reserved in ASCII code for the XON character and code 13_H (DC3) for the XOFF character. You must not use the same values when initializing the XON/XOFF characters. If the XON/XOFF characters are valid, the printout is implemented with XON/XOFF protocol. The "Print with BUSY signal" setting and parameter block 1 (waiting times) are then insignificant.

6.3.2 Configuration Data for Entering Message Texts (Parameter Block 3)

Message texts must always contain an end-of-text character (6.4). If you configure place holders in the message text, they must be delimited by a function character.
You can configure any ASCII character you want as a function character and end-of-text character.

Table 6-4. Configuration Data for Entering Message Texts (Parameter Block 3)

Block	Text Parameter	Value Range	Default Value on the CP 521
3	Function character	ASCII character (20_H to $7F_H$)	' (27_H)
	End-of-text character	ASCII character (20_H to $7F_H$)	$ (24_H)

6.3.3 Configuration Data for Message Text Printout (Parameter Blocks 4 to 6)

If you enter an illegal value, the CP will replace it with the relevant default value.

Table 6-5. Configuration Data for Message Text Printout (Parameter Blocks 4 to 6)

Block	Meaning	Value Range	Default Value on the CP 521
4	**Date and time of day:**		
	Order for data J,Y = Year M = Month T,D = Day	JMT, JTM,MTJ, MJT, TJM, TMJ, YMD, YDM, MDY, MYD, DYM, DMY (Lowercase also permissible)	TMJ
	Separator for date	ASCII character (20_H to $7F_H$)	"." (2E)
	Order for time of day H = Hours M = Minutes S = Seconds	HMS, HSM, MSH, MHS, SHM, SMH (Lowercase also permissible)	HMS
	Separator for time of day	ASCII character (20_H to$7F_H$)	"." (2E)
	24h clock (Ger.) 12h clock (Engl.)	d, D e, E	D (24h clock)
5	**Page format**		
	Lines per page Left margin Page number top, bottom,	20_H toFF_H 00_H to$3C_H$ T, o B, u no other characters	48_H (72_D) 00_H B bottom
6	**Header and footer**		
	Header 1 Header 2 Footer 1 Footer 2	K1"Text" 80 Characters K2"Text" 80 Characters F1"Text" 80 Characters F2"Text" 80 Characters	No header or footer

Page format

The page format is defined by configuring of parameter blocks 5 and 6.

Table 6-6. Page Format

Line	Page format	
1	Blank line	(if page number, header or footer have been configured)
.	Blank line	(if page number, header or footer have been configured)
.	Page No.	(top, if configured)
.	Blank line	(if page number configured at top)
.	Header1	
.	Header2	
.	Blank line	(if one or two headers have been configured)
.	Blank line	
.	Lines for message texts A maximum of 255 message texts can be configured. Each individual message text must not be configured with more than 80 characters Configuration is described in 6.3.	
.	Blank line	(if two footers have been configured)
.	Blank line	(if one footer has been configured)
.	Footer1	
.	Footer2	
.	Blank line	(if the page number is configured at bottom)
.	Page No.	(bottom, if configured)
.	Blank line	(if one or two headers have been configured)
n	Blank line	

Left margin (configurable up to <60 characters)

Number of lines per page 32 to 255
(configurable in the range (20_H to FF_H)

Configuring headers and footers

You can configure up to two headers and two footers.
The headers and footers may contain place holders for date and time of day as well as control characters. LF, FF and CR are not evaluated. Place holders for variables and message texts are also not evaluated. They appear in the printout as configured.
Headers and footers are configured according to the following schematic:

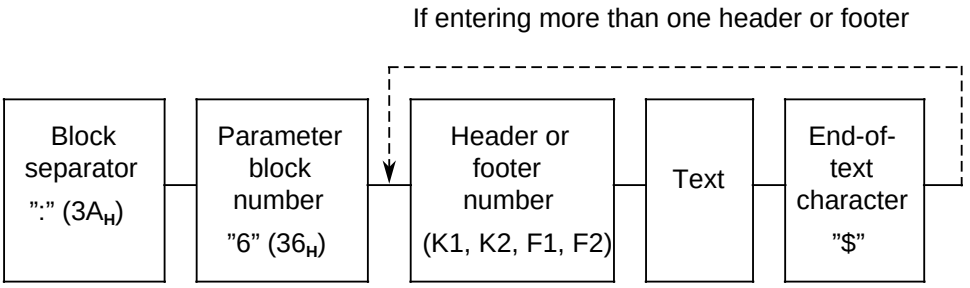


Figure 6-2. Schematic for Entering Headers and Footers

Enter the header or footer like the ASCII characters of a message text (6.4).

Note:
A header or footer must not be configured with more than 80 characters. The printed text may be longer than one line if place holders are used.

6.3.4 Setting the Mode Parameters (Parameter Block 7)

You can set the mode and the parameters for ASCII mode in parameter block 7.

If you want to work in print mode, this parameter block is insignificant for you since the CP 521 automatically sets print mode if a memory submodule is plugged in. There must be at least one configured message text in the memory submodule.

If you want to change from print mode to transparent ASCII or interpretive ASCII mode during operation, you can change parameter block 7 in the user program or with the programmer. See 7.3 for more detailed information.

Table 6-7. Operating Modes and Parameters for ASCII Mode (Parameter Block 7)

Block	Description	Value Range	Default Value
7	Operating mode	0_H: Print mode 1_H: Transparent ASCII mode 2_H: Interpretive ASCII mode	0 (with memory submodule) 1 (without memory submodule)
	Character delay time (4 digits decimal)*	(0001 to 3000) ·100ms	0001: 100ms
	Message length (3 digits decimal)	000 bis 256 bytes If length = 0, at least one end-of-text character must be entered.	064 bytes
	End-of-message character**	1 End-of-text character: 01_H to FF_H 2 End-of-text characters: 01_H toFF_H	$0D_H$ (carriage return)

* Only significant in ASCII mode

** Only significant in interpretive ASCII mode

6.3.5 Parameter Setting Example for the CP 521 in Print Mode

The CP 521 is plugged into slot 7 (intitial address 120). You can set its parameters as follows:

- Parameters for the serial interface (parameter block 7)
  - 2400 Baud (6)
  - Even parity (0)
  - No BUSY (0)
  - RS-232-C (V.24) interface (1)
  - 7 data bits (0)
  - Handshake OFF (0)

- Parameters for waiting times (parameter block 1)
  Not significant since XON/XOFF protocol should be used

- XON/XOFF protocol (parameter block 2)
  - XON character: DC 1 (11_H)
  - XOFF character: DC 3 (13_H)

- Configuration data for entering message texts
  No change with respect to default values

- Configuration data for message text printout
  - Time of day and date unchanged with respect to default values (parameter block 4)
  - Page format: 64 lines/page
    (parameter block 5) 10 characters left margin
    Page number bottom
  - Header: Test operation
    Footer: laboratory
    (parameter block 6)

- Setting the operating mode (parameter block 7)
  Not significant

DB 1 is called up on the programmer. Enter the values. Add comments to make the entries more readily understood (KS = ...).

Entry on PG 685 programmer for DB 1	Explanation

`: KS = 'Parameters for the serial';`
`: C = 'interface';`
`: KS = ':0600100';`
	":" : Block separator
	0 : Parameter block 0
	6 : 2400 Baud
	0 : Even parity
	0 : No BUSY
	1 : RS-232-C (V.24) interface
	0 : 7 data bits

`: C = 'XON/XOFF protocol';`
`: KS = ':2;`
`KH = 1113'`
	0 : Handshake OFF
	
	":" : Block separator
	2 : Parameter block 2
	11_H : XON character: DC1

`: C = 'Message text printout';`
`: KS = ':5;`
`KH = 400A`
`KS = U'`
	13_H : XOFF character: DC3
	
	":" : Block separator
	5 : Parameter block 5
	40_H : 64 lines/page
	$0A_H$: 10 characters left margin
	U : Page number bottom

`: C = 'Headers and footers';`
`: KS = ':6K1 Test operation $';`
`: C = 'F1 Laboratory $';`
	
	":" : Block separator
	6 : Parameter block 6
	K1 : Header 1 "Test operation"
	$: End-of-text character
	F1 : Footer 1
	$: End-of-text character

6.3.6 Parameter Setting with the PG 615 Programmer

Parameters must be entered in DB 1 on the memory submodule .

All parameters must be entered in ASCII code (KH = hexadecimal constant). It is advisable to use a list showing alphanumeric characters next to their ASCII codes. Chapter 8 contains an ASCII code table showing the hexadecimal value of ASCII characters.

After entering the data, transfer DB 1 to the memory submodule.

Example: Setting DB 1 parameters
Storing the same parameters in DB 1 as in the previous example, but without comments.

DB 1	Explanation
: KH = 3A30	Block separator ":", parameter block "0"
: KH = 3630	2400 baud (6)/even parity (0)
: KH = 3031	No BUSY signal (0)/RS-232-C (V.24) interface (1)
: KH = 3030	7 data bits (0)/Handshake OFF (0)
: KH = 3A32	Block separator ":"/parameter block "2"
: KH = 1113	XON character DC1 (11_H)/XOFF character DC3 (13_H)
: KH = 3A35	Block separator ":"/parameter block "5"
: KH = 400A	64 lines/page (40_H)/10 characters left margin ($0A_H$)
: KH = 5500	ASCII code for U/filter byte
: KH = 3A36	Block separator ":"/margin parameter block "6"
: KH = 4B31	Header H1
: KH = 5445	ASCII code for TE
: KH = 5354	ASCII code for ST
: KH = 4245	ASCII code for BE
: KH = 5452	ASCII code for TR
: KH = 4945	ASCII code for IE
: KH = 4220	ASCII code for B / space
: KH = 2446	End-of-text character $/footer F
: KH = 314C	1/ASCII code for L
: KH = 4142	ASCII code for AB
: KH = 4F52	ASCII code for OR
: KH = 2400	End-of-text character $/filler byte without meaning

6.4 Configuring Message Texts

You can store up to 255 message texts in data blocks 2 to 63 on the memory submodule. Every message text may be up to 80 characters long, including variables. Memory submodule capacity must be taken into account.
The message texts can be distributed over several data blocks in any sequence:

- You can store the message text in any data blocks (e.g. message text 13 in DB 20, message texts 5 and 18 in DB 7, ...).
- You can enter the DBs on the programmer in any order (DB 17, DB 3, ...).
- The message texts within a DB should follow immediately on one another. Your message texts will be clearer if you exchange "KS" and "C" in each new input line and if the message texts are separated from each other by pressing the Enter key. (Figure 6-3)
- The DBs in the memory submodule are independent of the DBs in the CPU. For example, you can store message texts in DB 54 on the memory submodule while simultaneously using a DB 54 in the user program.

6.4.1 Structure of a Message Text

When configuring message texts, you must keep to the following structure, otherwise a correct printout cannot be guaranteed.

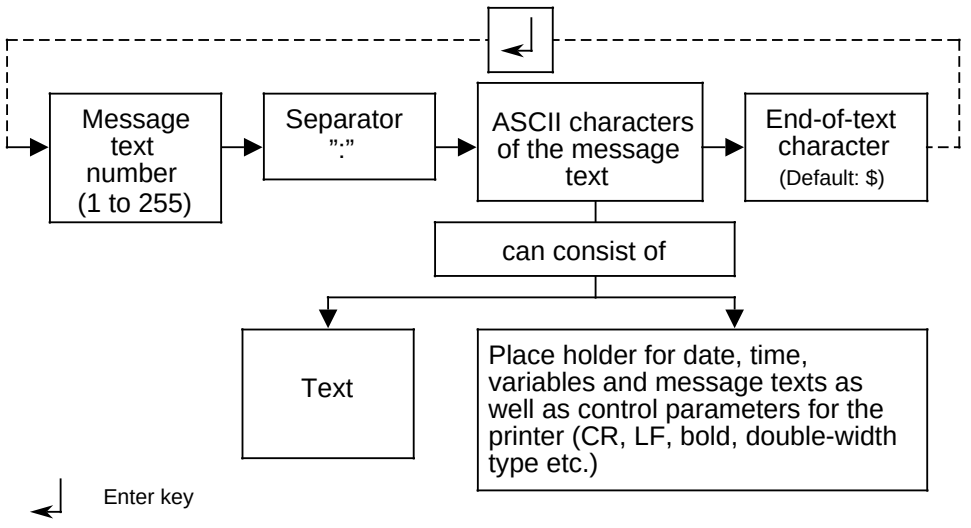


Figure 6-3. Structure of a Message Text

Message text number
The message text is stored in the DB under a message text number (1 to 255). If you want to print a message text, transfer the message text number and the "Print message text" job to the CP 521 in the user program (6.6.1).

Separators
After the message text number you must enter the separator ":".

ASCII characters in the message text
You can enter up to 80 ASCII characters of message text. If you enter more than 80 ASCII characters, these are not evaluated. The printout of the configured message text on the printer may be longer than 80 characters.

Example: You are configuring a message text which is longer than 80 characters.
The message text includes place holders for the time of day. This only requires three ASCII characters for input (e.g. 't'). On output, the place holder is replaced by the actual time of day (e.g. 15.00.00) (8 characters).
The message text printed out is thus 85 characters long.

Note:
If the printout of a message text is longer than 80 characters, this might interfere with the print format (depending on printer line feed, page makeup...)

The following can be entered as ASCII characters:
- Text
  The text can contain all printing characters (see the manual of the printer connected).
- Place holders
  The date, time of day, variables and further message texts can be inserted in a message text by configuring place holders.
- Control characters
  You can enter control characters for printer functions (double-width characters, subscript ...).

Example: Configuring message texts without place holders

You want to store the following message texts in DB 2:
Message text 4: Excess temperature
Message text 5: Coolant loss

Entry on the PG 685 programmer for DB 2	Explanation
`0: KS = '4:Excess temperature$';`	Message text number 4 Separator ":", message text End-of-text character "$"
`11: C = 'Coolant loss:$';`	Message text number 5 Separator ":", message text End-of-text character "$"

6.4.2 Entering Place Holders

Place holders are used for inserting the date, time of day, variables and further message texts.
Some place holders can also be used in headers and footers. The headers and footers are configured on the memory submodule in data block 1, parameter block 6 (6.3.3).
A place holder is enclosed between two function characters. You can configure the function characters in data block 1, parameter block 3. The default value for the function character is 27_{H} = '.
Wrongly entered place holders are treated as text.

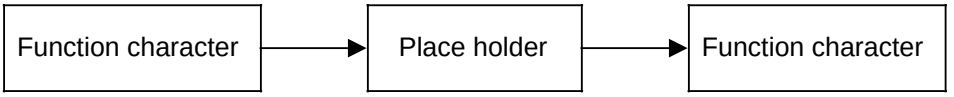


Figure 6-4. Structure of a Place Holder

Table 6-8. Overview of Place Holders Configurable in Message Texts

Place Holder	Meaning	Max. number in one message text	Configurable in header and footer	Comment
d or D	Insert Date	1	Yes	
t or T	InsertTime of Day	1	Yes	
KPa	Insert message text (fixed message text number)	Unlimited	No	The message text to be printed must be < 250 characters long
Data format (KH, KF, KS, KM, KT, KC, KY)	Insert variable	3 variables per 16 bits	No	Transfer values of the variables in bytes 2 to 7 to the CP

You can enter place holders in either lower or uppercase.

Example: Place holders for date and time of day: ”D”, ”d”, ”T”, ”t”
Place holders for variables: ”KH”, ”Kh”, ”kH”, ”kh”

Transfer of data when using place holders for "Inserting variables".

Transfer the values for the place holders "Insert variables" to bytes 2 to 7 on the CP 521.
Access to bytes 2 to 7 depends on the order of the place holders in the message text.

- The first place holder in the message text is replaced by the data in bytes 2 and 3.
- The second place holder in the message text is replaced by the data in bytes 4 and 5.
- The third place holder in the message text is replaced by the data in bytes 6 and 7.

6.4.3 Place Holders for Date and Time of Day

When printing out a message text the following takes place:
- The place holder "D" for the date is replaced by the actual date
- The place holder "T" for the time of day is replaced by the actual time.

The output format of the date and the time of day can be configured. You must do this in data block 1, parameter block 4 on the memory submodule (6.3.3). The following is the default output format:
- For the date:<Day> . <Month> . <Year>
- For time of day: <Hour> . <Minute> . <Second>

Example: Configuring message texts with place holders for date and time of day

You want to store the following message texts in DB 3:
Message text 7: The motor went down at <time of day>
Message text 8 : This is the daily listing for <date>

Entry on the PG 685 programmer for DB 3	Explanation
`: KS = '7:The motor went down`	Message text number ”7”, Separator ”:”, message text with place holder for time of day
`: C = at 'T'$';`	Message text, end-of-text character ”$”
`: KS = '8: This is the daily';`	Message text number ”8”, Separator ”:”, message text
`: C = 'listing for'D'$';`	Message text, place holder for date, end-of-text character ”$”

6.4.4 Place Holders for Message Text

The place holder is replaced by a message text on printout. The number of the message text to be replaced is specified when the message is configured.

Direct entry of the message text number "KPa"

"a" is the number of the message text to be inserted. You must specify ”a” when configuring the message text. The values 1 to 255 are allowed for "a". If the message text called contains place holders, these are treated as text and printed out.
You can use the dummy character "KPa" place holder several times in one message text. In this way you can connect several message texts and print them out together.

Example: Configuring message texts with place holders for inserting message text (KPa)

You want to store the following message texts in DB 7:
Message text 70: The following overview indicates <Insert message text 73>, <Insert message text 74> and <Insert message text 75>.
Message text 73: the problems that have arisen during the monitoring period
Message text 74: the measures taken
Message text 75: the length of the resulting downtimes

Entry on the PG 685 for DB 7	Explanation
0: KS = '70:'The following overview;	Message text number, separator, message text
12: C = 'indicates, 'KP73',';	with place holder for "Insert message text 73"
22: KS = ' 'KP74' and 'KP75'.$';	Message text, end-of-text character
32: C = '73:The problems which have';	Message text number, separator, message text
43: KS = 'arisen during';	Message text
54: C = 'the monitoring period$';	Message text, end-of-text character
63: KS = '74:the measures';	Message text number, separator, message text
73: C = 'taken$';	Message text, end-of-text character
84: KS = '75:the length of the';	Message text number, separator, message text
96: C = 'resulting';	Message text
106: KS = 'downtimes$'; 112:	Message text, end-of-text character

Printout of message text 70 including text expansions:
The following overview indicates the problems which have arisen during the monitoring period, the measures taken and the length of the resulting down-times.

Maximum length of message texts
Messages may not exceed 80 characters in length. If you use place holders, the printout may be longer than 80 characters. The maximum length of the printout is 250 characters.

Example: Configuring a message text with place holders for inserting message text.

You want to store the following text in DB 9:
Message text 90: The following table gives an overview of <Insert message text 91>, <Insert message text 92> and <Insert message text 93>.
Message text 91: faults occurring in the press controller with downtimes
Message text 92: the resulting waiting times of the next press controller
Message text 93: average press controller downtimes during the current year

Entry on the PG 685 for DB 9	Explanation
0: KS = '90:the following'; 11: C = 'table gives an overview' 23: KS = 'of 'KP91', 'KP92'; 33: C = 'and 'KP93'.$'; 39: KS = '91:the faults occurring in'; 51: C = 'the press controller with'; 63: KS = 'down'; 75: C = 'times$'; 76: KS = '92:the resulting waiting'; 88: C = 'times of the next press'; 100: KS = 'controller'; 112: C = '$'; 114: KS = '93:average press control'; 126: C = 'ler downtimes during the'; 138: KS = 'current year' 150: KS = ''; 155:	Message text 90 with a length of 68 characters Message text 91 with a length of 57 characters Message text 92 with a length of 55 characters Message text 93 with a length of 57 characters

* ▒▒▒▒ This text will not be printed

Final printout:
The following table gives an overview of the faults occurring in the press controller with downtimes, the resulting waiting times of the next press controller and average press downtimers (190 characters)

6.4.5 Place Holders for Variables

If you have configured up to three dummy characters for variables in a message text, you must specify the variables in bytes 2 to 7 in the "Print message text" job:

- Bytes 2 and 3 : Value of the 1st variable (data format as configured)
- Bytes 4 and 5 : Value of the 2nd variable (data format as configured)
- Bytes 6 and 7 : Value of the 3rd variable (data format as configured)

The place holder is replaced by the variable at printout:

- If the message text contains more than three place holders for variables, they are not interpreted as place holders but printed out as text.
- If a place holder is wrongly configured, it is also treated as text and printed out.
- When place holders are replaced by the appropriate values, the total length may be greater than 80 characters.
- The first place holder for variables configured in the message text is replaced by variable 1, the second is replaced by variable 2 and the third by variable 3.

You must specify the data format in which the variable has been transferred from the CPU for the place holder.

Table 6-9. Data Formats for Variables

Data format		Value Range	Number of Printed Positions
KM	Constant: Bit pattern	16 bits	16
KH	Constant: Hexadecimal pattern	0000_H to $FFFF_H$	4
KC/KS	Constant: Alphanumeric characters	20_H to $7F_H$ / 20_H to $7F_H$	2
	KT Constant: Timebase	1.0 to 999.3 *	4
KZ	Constant: Count	0 to 999	3
KY	Constant:Byte, byte	0 to 255, 0 to 255	7
KF	Constant: Fixed-point number	-32768 to +32767	6
KF a,b	Constant: 5-digit fixed-point number a Places before point (5) b Places after point (5) a+b 5	-32768 to +32767	Con-figurable

* The time is printed out in seconds

”KT” data format

You can use data format "KT" to insert internal times/D in the message text. Load the time in BCD with L TX into the accumulator and then transfer it to the CP. The four places are given in seconds (s) on the printer.

Table 6-10. ”KT” Data Format Printouts

Time (in STEP 5)	3.0	24.0	207.0	8.1	46.1	840.1	…	4.3	69.3	423.3
Printout of the KT variable (i_n s)	0.03	0.24	2.07	0.8	4.6	84.0	**…**	0040	0690	4230

(=space)

”KF” data format

If you configure the "KF" data format, six characters are always printed out.

Example: Printout of the "KF" data format (u = space)

```
KF  =  +12345        12345
KF  =  +357           357
KF  =  -12345       -12345
KF  =  - 357         -357
```

"KFa,b" format

You can set the following possible parameters:
- Specify the number of places before the point with "a"
- Specify the number of places after the point with "b"
- The sum of places before the point and places after the point must not exceed five.

Example: Printout of the KFa,b variables

Table 6-11. Typical "KFa,b" Data Format Printouts

Configu-ration	Printout of KF Variables					
	+7	- 7	256	- 256	32767	-32767
KF 5,0	7	-7	256	-256	32767	-32767
KF 4,0	1	-1	26	-26	3277	-3277
KF 3,0	0	-0	3	-3	328	-328
KF 2,0	0	-0	0	-0	33	-33
KF 1,0	0	-0	0	-0	3	-3
KF 0,0	7	-7	256	-256	32767	-32767
KF 4,1	0,7	-0,7	25,6	-25,6	3276,7	-3276,7
KF 3,1	0,1	-0,1	2,6	-2,6	327,7	-327,7
KF 2,1	0,0	-0,0	0,3	-0,3	32,8	-32,8
KF 1,1	0,0	-0,0	0,0	-0,0	3,3	-3,3
KF 0,1	0,0	-0,0	0,0	-0,0	0,3	-0,3
KF 3,2	0,07	-0,07	2,56	-2,56	327,67	-327,67
KF 2,2	0,01	-0,01	0,26	-0,26	32,77	-32,77
KF 1,2	0,00	-0,00	0,03	-0,03	3,28	-3,28
KF 0,2	0,00	-0,00	0,00	-0,00	0,33	-0,33
KF 2,3	0,007	-0,007	0,256	-0,256	32,767	-32,767
KF 1,3	0,001	-0,001	0,026	-0,026	3,277	-3,277
KF 0,3	0,000	-0,000	0,003	-0,003	0,328	-0,328
KF 1,4	0,0007	-0,0007	0,0256	-0,0256	3,2767	-0,2767
KF 0,4	0,0001	-0,0001	0,0026	-0,0026	0,3277	-0,3277
KF 0,5	0,00007	-0,00007	0,00256	-0,00256	0,32767	-0,32767

Each fixed-point number is represented by at least (a + 1) digits. Leading zeros are replaced by spaces, and are not suppressed.

6.4.6 Control Parameters

You can configure printer control parameters in two different ways in the message text:

- Direct with the KH format
- Entry with the ^ character (CTRL key)

Direct entry with the "KH" format
When configuring a message text, the control parameters are entered direct as hexadecimal values (data format "KH"). Use the table to find the ASCII codes of the control parameters. If you are using the programmer, you must switch from the "KS" or "C" format to the "KH" format and specify the control parameters in ASCII code. You then switch back to the "KS" or "C" data format.

Example: Direct entry: Configuring a message text with control parameters.

You want to store the following message text in DB 4:
Message text 40: Motor works <Line feed> <Carriage return>
Brummhausen <Line feed> <Carriage return> 9999
Wackeldorf

Entry on the PG 685 on DB 4	Explanation
: KS = '40: Brummhausen';	Message text no. 40, separator, message text
: KH = 0A0D	ASCII code for line feed ($0A_H$) and carriage return ($0D_H$)
: C = 'Motor works';	Message text
: KH = 0A0D	ASCII code for line feed ($0A_H$) and carriage return ($0D_H$)
: KC = '9999 Wackeldorf$';	Message text, end-of-message character

Entry with the "^" character (not with PG 615)
If the module encounters the "^" character when evaluating the message texts, it automatically subtracts 40_H from the next character.

Example: Set double-width type on the PT 88 printer.
This is done with the ESC 8 command .
The "ESC" control character has ASCII code $1B_H$. In the ASCII code table find the ASCII character with the code $1B_H + 40_H = 5B_H$.
This is the "[" character.
Entry in message text: ^[8.

Explanation: The module recognizes the "^" character.
40_H is subtracted from the ASCII code of the next character "[" ($5B_H$): $5B_H - 40_H = 1B_H$. $1B_H$ is the ASCII code for the "ESC" control character . The module assigns the printer with the job of executing "ESC 8", i.e. switching to double-width type.

Example: Configuring a message text with control parameters (^entry).

You want to store the following message text in DB 5:
Message text 50: <Double-width type on> Monthly overview <Double-width type off>

Entry on PG 685 for DB 5	Explanation
`0: KS = '50:^[8 Monthly overview^[<$';` `14:`	Message text no. 50 Separator, switch on character string for double-width type, message text, switch off character string for double-width type

See the printer manual for an overview of control parameters.

6.5 Data Transfer from the CPU to the CP 521

Data is transferred in the data cycle via the I/O bus.

6.5.1 Data from the CP 521 to the CPU (Status of the CP 521, the printer and current clock data)

The CP 521 makes the following information available cyclically via the PII of organization block OB 1:

Table 6-12. Data from the Module to the CPU

Byte	Meaning
0	Status of the CP 521
1	Printer status and day of the week
2 to 7	Date and time of day on the CP 521 clock

This information can then be read into the control program with load operations, and evaluated.

Reading the status of the module in print mode (byte 0 in the PII)

Byte 0 "Status of the module" is divided into two halves. The information in one half-byte is independent of that in the other half-byte. They can be combined as required.

The following pages contain a description of the individual bits.

Table 6-13. Status of the Module in Print Mode (Byte 0 in the PII)

Bit 4 to 7	Bit 0 to3	Status	Detailed Explanation
0	0	No error	
X	1	Memory submodule fault	Fault in submodule configuration
X	2	No text	No message texts have been configured on the memory submodule
X	6	Message text number not configured	No message text has been stored on the memory submodule under the number specified
X	7	No backup battery	On the CP 521: - no battery inserted - battery fault
X	8	Message buffer overflow	The module cannot process any further print jobs at present. The job must be repeated.
1	X	Clock defective	
2	X	Default time set	The clock has been set with the values Sunday, 1.1.89, 12.00.00.
3	X	Time of day/date error	At least one manipulated variable is outside the permissible range. The clock has not accepted the new clock data and is continuing with the current data.
8	X	Hardware error	

X: Signal status is irrelevant for the other half byte

Example: Scanning the status of the module (byte 0 in the PII)

The module is installed in slot 7 (initial address 120).
If the battery of the printer output module is defective, message text No. 8 is printed out.

STL FB 100	Explanation
NAME :FEHLER1	
:L IB 120	Read byte 0 (status byte) into ACCUM 1.
:L KH 000F	Set bits 4 to 7 to ”0” by ANDing them with 000FH.
:AW	The signal states of bits 0 to 3 stay the same.
:L KH 0007	Compare ACCUM 1 contents with 0007H. If the status byte does not contain the value 7H, the battery is functional. Processing of the block is terminated.
:><F	
:BEC	
:L KH 3008	
:T QW 120	The battery is defective. The CPU assigns the CP 521 with the job of printing message number 8.
:BE	

Status of the printer and day of the week (byte 1 in the PII)

Byte 1 is divided into two half-bytes. The status of the printer is specified in the left half-byte, bits 4 to 7, and the current day of the week is in the right half-byte, bits 0 to 3. Both are independent of each other.
They can be combined as needed.

Table 6-14. Status of the Printer and Day of the Week

Byte 1 Bit 4 to 7	Bit 0 to 3	Meaning
X	1...7	1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday
0	X	Printer ready
1	X	Printer not ready

X= Signal status irrelevant for the other half-byte

Example: Evaluating the printer status

The module is installed in slot 7 (initial address 120).
If the printer is not ready for operation, output 4.1 is set.

STL FB 100	Explanation
`NAME :FEHLER1`	Evaluting "Printer status"
`:L KH 0010`	Compare status byte with 10_H
`:L IEW 120`	
`:AW`	Combine word-wise
`:JZ M001`	If "zero", jump to label 001
`:S A 4.1`	If not "zero", set output 4.1
`M001:BE`	Block end

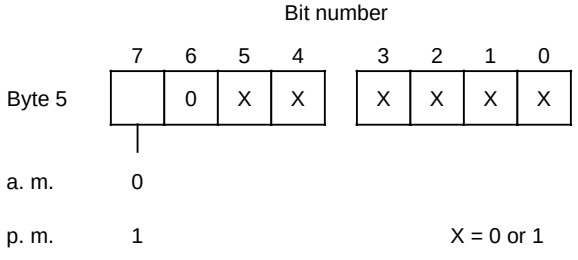
Date and time of day (bytes 1 to 7 in the PII)

Date and time of day (bytes 1 to 7 in the PII)
The printer status is specified in byte 1 bit 4. This information is independent of the current day of the week, which is specified in byte 1, bit 0 to 3.

Table 6-15. Date and Time of Day

Byte	Value Range	Meaning
1	1 . . . 7	Bit 0 to 3 : 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday Bit 4=0 : Printer ready Bit 4=1 : Printer not ready
2	01_{BCD} . . . 31_{BCD}	Day
3	01_{BCD} . . . 12_{BCD}	Month
4	00_{BCD} . . . 99_{BCD}	Year
5	00_{BCD} . . . 23_{BCD} 01_{BCD} . . . 12_{BCD} 81_{BCD} . . . 92_{BCD}	Hour 24h clock 12h clock a.m. 12h clock p.m.
6	00_{BCD} . . . 59_{BCD}	Minute
7	00_{BCD} . . . 59_{BCD}	Second

***** Note on byte 5 "Hour":



Example: Reading the current time of day

The module is installed in slot 7 (initial address 120).
The clock data is to be output on digital output modules from address 8.

STL FB 100	Explanation
NAME :UHRDATEN :L IEB 121 :T QB 8 :L IW 122 :T QW 9 :L EW 124 :T AW 11 :L EW 126 :T AW 13 :BE	Reading the current clock data and transferring it to the digital output modules.

Note:

You can also evaluate the following messages using the integrated clock:

- Clock defective ($1X_H$)
  This error evaluation is only relevant after a restart.
- Default time set ($2X_H$)
  This evaluation is only only meaningful if you are operating the CP 521 without a backup battery.
- Time of day or date error ($3X_H$)
  This error evaluation is only meaningful after you have set the clock.

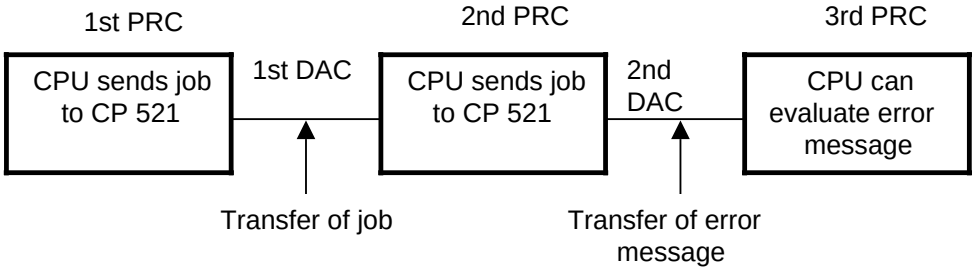
6.5.2 Data from the CPU to the CP 521 (CPU Jobs)

The CPU can transfer eight bytes to the CP 521. The CP 521 processes the jobs (bytes 0 and 1) in response to a pulse edge, i.e. it responds only to signal changes in byte 0 or byte 1.
The CPU assigns a job to the CP 521 in byte 0 and in bytes 1 to 7 it can then transfer other information required for executing the job.
The CP 521 has a message buffer. Jobs for the peripheral device are written into the message buffer. The job is deleted in the message buffer when the CP 521 has processed it. The message buffer can hold up to 255 jobs. If the CPU sends a large number of jobs within a short period, the number of jobs still to be processed increases. If there are 255 jobs waiting to be processed and the CPU sends another one, the CP 521 is no longer able to process this last print job.
In this case, the CP 521 reports "CPU job not accepted" (08_H) to the CPU in byte 0 of the PII "Status of the module".

Note:

If you must send a large number of jobs within a short period, check that there is still space in the message buffer.

You can only scan the "CPU job not accepted" error message two program cycles after the job has been assigned (Figure 6-5). Take into account the following to ensure correct scanning of the message buffer in the control program.



PRC = program cycle, DAC = data cycle

Figure 6-5. Job Transfer and Error Message Evaluation

6.6 Overview of Possible Jobs in Print Mode

The CPU transfers the job to the CP 521 in word 0. The job number is stored in byte 0.

Table 6-16. Permissible Jobs in Print Mode (Byte 0 in the PIQ)

Byte 0	Byte 1	Meaning
00_H		Data not relevant
10_H		Set clock; specify settings in bytes 1 to 7
20_H		Specify page number in byte 1
30_H		Output message with CR/LF at the end Specify message text number in byte 1
40_H		Specify message text number in byte 1 Output message without CR/LF at the end
50_H		Output page feed
60_H		Output line feed
70_H		Delete message buffer
80_H		Printout of all messages
90_H	00_H 20_H 70_H 71_H 72_H	Configuration of the serial interface Setting the XON/XOFF character parameters Print mode Transparent ASCII mode Interpretive ASCII mode
Other		Data not relevant
		Print jobs. These are written into the message buffer, if required. Job 80_H, "Printout of all messages", is executed immediately.*

***** The message text numbers are also printed out with CPU job 80H.

Printing the message text with and without CR/LF at the end (job 30_H and 430_H)
You must specify the message number in hexadecimal form in byte 1.
The "Print message text without CR/LF" job allows you to print several message texts in one line. If your configured message texts contain, for example, only variables, you can generate tables with this job.

Permissible inputs with the job "Print message text" (job 30_H and 40_H)
The following can also be transferred to the CP 521 in bytes 2, 4 and 6 when using the "Print message text" job (job 30_H and 40_H):

- Values of variables
  When the message text is printed out, the values transferred to the CP 521 are inserted in place of the place holders configured in the message text.

Table 6-17. Transfer of Additional Input in the Case of the "Print Message Text" Job

Word (Byte)	Meaning	Assignment
2 (2+3)	Value of variable 1	0000_H to $FFFF_H$
4 (4+5)	Value of variable 2	0000_H to $FFFF_H$
6 (6+7)	Value of variable 3	0000_H to $FFFF_H$

Setting the clock (job 10_H)
Specify settings in BCD code in bytes 2 to 7. Enter weekdays as digits between 1 and 7 (1 = Sunday, 2 = Monday, .., 7 = Saturday).
If any clock data is to remain unchanged, enter the value FFH in the relevant byte.
If a setting lies outside the permissible limit values, the clock is not set. The CP sends the "Time/date error" error message .

Table 6-18. Transfer of Additional Inputs in the Case of the "Set Clock" Job

Byte	Meaning of the Setting Assignment	Assignment
2	Day	01_{BCD} to 31_{BCD}
3	Month	01_{BCD} to 12_{BCD}
4	Year*	00_{BCD} to 99_{BCD}
5	Hour (Configuration of 12h clock or 24h clock in DB 1 on the memory submodule)	00_{BCD} to 23_{BCD} with 24h clock 01_{BCD} to 12_{BCD} with 12h clock a.m. 81_{BCD} to 92_{BCD} with 12h clock p.m.
6	Minute	00_{BCD} to 59_{BCD}
7	Second	00_{BCD} to 59_{BCD}

***** Leap years are taken into acacount automatically

Setting the page numbers (job 20_H)
Specify the page number in binary code in byte 1.

Executing page feed (job 50_H)
If job 60_H , "Output line feed", has not yet been executed, it will be cancelled by this job.

Outputting line feed (job 60_H)
This job generates a blank line.
A subsequent job 50_H, "Execute page feed", deletes this job, if it has not yet been executed.

Deleting the message buffer (job 70_H)
All CPU jobs stored in the buffer are deleted.

Printout of all messages (job 80_H)
The message texts are printed out as configured, with message text number in ascending order.
- No place holders are evaluated
- Printer control characters are executed.

Transfer parameter setting data (job 90_H)
On restart, the CP 521 receives the DB 1 data as initialized by you on the memory submodule. You can change the data of parameter blocks 0, 2 and 7 with this job. Memory contents remain unchanged.

- Job 9000_H (parameter block 0)
  - Parameters of the serial interface (for example changing the baud rate)
- Job 9020_H (parameter block 2)
  - Parameters for "XON" and "XOFF"protocol
- Job 9070_H, 9071_H and 9072_H (parameter block 7)
  Setting the mode:
  - Setting the printer mode with job 9070_H
  - Setting transparent ASCII mode with job 9071_H
  - Setting interpretive ASCII mode with job 9072_H

Note:

The data transferred to the CP with the "Transfer parameter setting data" job has priority over the data stored in data block 1 of the memory submodule. After POWER ON, the parameter setting data on the memory submodule is valid as long as it is not overwritten by the user program.

6.6.1 Printing the Message Text (Job 30_H and 40_H)

After the "Print out message text XY" job the CP 521 transfers message text XY to the printer. You specify whether the printer is to execute line feed (LF) or carriage return (CR) after printing message text XY.

Prerequisites

- You must have configured the message text to be printed on the memory submodule. Assign a number to each message text when configuring it.
- You must have plugged the configured memory submodule into the CP 521.

Operator inputs required

- Specify the message text number in hexadecimal form in byte 1
- Specify the job in byte 0
  - Job 30_H: Print message text with CR/LF
  - Job 40_H: Print message text without CR/LF

Additional inputs

You can store a further three 16-bit variables in bytes 2 to 7 if you have configured place holders for variables in the message text.
You must enter the data in the form in which you configured it in the message text in the memory submodule (6.4.6).
At printout, the place holders are replaced by actual variable values.

Table 6-19. PIQ in the Case of the "Print Message Texts" Jobs

Byte	Meaning	Permissible Assignment
0	Job numbers	30_H : Print message text with CR/LF 40_H : Print message text without CR/LF 50_H : Output page feed 60_H : Output line feed
1	Message text number	01_H to FF_H (1to 255)
2 3	Value of the variables	0000_H to $FFFF_H$
4 5	Value of the variables	0000_H to $FFFF_H$
6 7	Value of the variables	0000_H to $FFFF_H$

Permissible insertions when printing the message text

You can enter place holders when configuring message texts on the memory submodule. At printout, these place holders are replaced according to their meaning.

You can enter place holders for:
- Date and time of day (6.4.3)
- Further message texts (6.4.5)
- Variables (6.4.6)
- Control characters for the printer (double-width type, bold type ...) (6.4.4)

The CP 521 can report further error messages to the CPU (6.6).
E.g.:
- Default time set
- Printer not ready
- No backup battery

Example: Initiating output of a message text without job monitoring (CP replies are not evaluated)

The CP is installed in slot 7 (initial address 120). A memory submodule on which message text 9 is configured is plugged into the CP (6.4.3). Message text 9 is to be printed if flag 20.0 is set.
The following program prints message text 9 only once. If you want to print message text 9 again, you must write another value into word 0, e.g. KF = 0, and then set 9 again so that it will be detected as a change in word 0. (Edge-triggered job processing 6.5.2).

STL FB 100	Explanation
NAME :TEXT1	
0005 :AN F 20.0	If flag 20.0 is not set, the program is terminated
0006 :BEC	
0007 :L KH 3009	Load ACCUM 1 with job number 30_H and
0008 :T QWs 120	message text number 9 and transfer to the CP in word 0.
0009 :BE	Block end

Example: **Initiating output of a message text with job monitoring (CP responses are evaluated)**

The CP 521 is installed in slot 7 (initial address 120). Message text 1 (I 5.0) or 2 (I 5.1) is to be output if input I 5.0 or I 5.1 changes its signal state from "0" to "1".

STL	Explanation
A I 5.0 S F 5.0	If I 5.0 is "1", flag 5.0 is set.
A I 5.1 S F 5.1	If I 5.1 is "1", flag 5.1 is set.
AN I 5.0 A F 6.0 R F 5.0 R F 6.0	If I 5.0 and flag 6.0 are "1", flags 5.0 and 6.0 are reset.
AN I 5.1 A F 6.1 R F 5.1 R F 6.1	If I 5.1 and flag 6.1 are "1", flags 5.1 and 6.1 are reset.
O F 7.0 O F 8.0 JC= F003	If flag 7.0 or 8.0 is "1", the program jumps to label 003. Flag 7.0 or 8.0 is then set if a print job has been activated and the job has not yet been transferred. In this case, a further job is flagged (F 5.0 or F 5.1) but not yet transferred to the module.
A F 5.0 AN F 6.0 JC= F001	If flag 5.0 is set and flag 6.0 is not set, the program jumps to label 001 (job 1 transferred to the module).
A F 5.1 AN F 6.1 JC= F002 BEU	If flag 5.1 is set and flag 6.1 is not set, the program jumps to label 002 (job 2 transferred to the module)

STL (Cont.)				Explanation (Cont.)
M 001	S	F	6.0	Set flags 6.0 and 7.0 and transfer the "Output message text 1 with CR/LF" job to the module.
	S	F	7.0	
	L	KH	3001	
	T	QW	120	
	BEU			Block end
M 002	S	F	6.1	Set flags 6.1 and 7.0 and transfer the "Output message text 2 with CR/LF" job to the module.
	S	F	7.0	
	L	KH	3002	
	T	QW	120	
	BEU			Block end
M 003	A	F	7.0	If flag 7.0 is set, it is reset and flag 8.0 is set. This only happens in the control program after a job is initiated.
	R	F	7.0	
	S	F	8.0	
	BEC			Block end if F 7.0 is "1"
	L	IB	120	If flag 7.0 is reset and flag 8.0 is set, the program checks to see if the job has been transferred. This happens two program cycles after initiation of the job.
	L	KH	0008	
	!=F			
	BEC			
	A	F	8.0	If the job has not been transferred, flag 8.0 remains set and in the following program cycles the overflow scan is repeated until the CPU job is accepted. Only then is flag 8.0 reset and those jobs that have arrived in the meantime are processed.
	R	F	8.0	
	L	KH	0000	
	T	QW	120	
	BE			

6.6.2 Setting the Time of Day and the Date (Job 10_H)

You can set the clock in two ways:
- With the user program
  This is suitable, for example, if the clock is to be set for a SINEC L1 time synchronization
- In RUN mode with the "FORCE VAR" programmer function

Operator inputs required:

- Specify job number 10H in byte 0
- Specify the new settings in bytes 1 to 7

Note:

If you set the clock with the "FORCE VAR" programmer function, you must first transfer the settings in bytes 1 to 7 and then the "Set clock" job (10H) in byte 0 to the CP 521.

Settings

You must note the following with regard to settings:
- The settings must be transferred in BCD code to the CP 521.
- If you enter the value "FFH" in one of the settings, the current time remains.
- 01 = Sunday, 02 = Monday, 03 = Tuesday, 04 = Wednesday, 05 = Thursday, 06 = Friday, 07 = Saturday
- AM/PM flag (only with 12h clock)
  When specifying "Hours" settings in byte 5, you must take account of the AM/PM flag (bit 7)in the case of the 12h clock: bit 7 = 1 p.m. and bit 7 = 0 a.m.B

Table 6-20. PIQ in the Case of the "Set Clock" Job

Byte	Meaning	Value Range
0	"Set clock"Job	10_H
1	Weekday (01 = Sunday, 02 = Monday, ..., 07 = Saturday)	01_{BCD} to 07_{BCD}
2	Day	01_{BCD} to 31_{BCD}
3	Month	01_{BCD} to 12_{BCD}
4	Year	00_{BCD} to 99_{BCD}
5	Hours 24h clock 12h clock a. m. 12h clock p. m.	 00_{BCD} to 23_{BCD} 01_{BCD} to 12_{BCD} 81_{BCD} to 92_{BCD}
6	Minutes	00_{BCD} to 59_{BCD}
7	Seconds	00_{BCD} to 59_{BCD}

Example: Setting the time of day with the "FORCE VAR" programmer function

The clock must be set with the following data: Mo 03., 13.00.00 (24h clock).
The current data for month and year is to be retained.
The module is installed in slot 7 (initial address 120).

Operand	Signal states	Meaning
QW 126	KH= 0000	Minute and second
QW 124	KH= FF13	Year (retained) and hour
QW 122	KH= 03FF	Date and month (retained)
QW 120	KH= 1002	"Set clock" and day of the week

6.6.3 Setting the Page Numbers (Job 20_H)

When configuring the memory submodule, you can specify if and where a page number is to be printed. The default begins with page number 1 printed at bottom centre of the page.
You can change this default in the following ways:

You can change this default in the following ways:
- In the user program
- In RUN mode with the "FORCE VAR" programmer function

Operator inputs required
- Specify the page number in byte 1 (01_H to FF_H)
- Specify job 20_H in byte 0

Example: Setting page numbers with the "FORCE VAR" programmer function

The module is installed in slot 7 (initial address 120).
The following pages are to be printed, beginning with page number 20.

Operand	Signal States	Meaning
QW 120	KH= 2014	Transfer "Set page numbers" job and page number $14_H = 20_D$ to word 0.

6.6.4 Outputting Page Feed (Job 50_H)

When instructed by the CPU, the CP 521 orders the printer to output page feed. Page feed is implemented by executing the number of line feeds configured (parameter block 5 "Page format": lines per page).

Operator inputs required
Specify job 50_H in byte 0.

Example: Outputting page feed

The module is installed in slot 7 (initial address 120).
Page feed is to be output on the printer.

Operand	Signal States	Meaning
QW 120	KH= 5000	Transfer "Output page feed" job in byte 0.

Note:
The CP stores print jobs in a message buffer if they cannot be executed immediately because the CP is still busy with a previously assigned print job. When the "Output page feed" job is assigned, job 60_H, "Output line feed", in the message buffer is deleted and is no longer executed.

6.6.5 Outputting Line Feed (Job 60_H)

When instructed by the CPU, the CP 521 orders the printer to output line feed.

Operator inputs required
Specify job 60_H in byte 0.

Example: Outputting line feed

The module is installed in slot 7 (initial address 120).
Line feed is to be output on the printer.

Operand	Signal States	Meaning
QW 120	KH= 6000	Transfer "Output line feed" job in byte 0

Note:

When job 50_H, "Output page feed", is assigned, job 60_H, "Output line feed", in the message buffer is deleted and is no longer executed.

6.6.6 Deleting the Message Buffer (Job 70_H)

When instructed by the CPU, the CP 521 deletes all CPU jobs stored in the message buffer.

You can delete the message buffer in two ways:
- In the user program
- In RUN mode with the "FORCE VAR" programmer function

Operator inputs required
Specify job 70H in byte 0.

Example: Deleting the message buffer with the "FORCE VAR" programmer function

The module is installed in slot 7 (initial address 120).

Operand	Signal States	Meaning
QB 120	KH= 7000	Job: "Delete message buffer"

6.6.7 Printout of All Configured Message Texts (Job 80_H)

When instructed by the CPU, the CP 521 prints out all configured message texts on the user submodule. The message texts are printed as configured with message text number in ascending order. Place holders are not evaluated.

You can assign the job in two ways:
- In the user program
- In RUN mode with the "FORCE VAR" programmer function

Operator inputs required
Specify byte 0 in job 80_H.

Example: Printing out all configured message texts with the "FORCE VAR" programmer function

The module is installed in slot 7 (initial address 120).

Operand	Signal States	Meaning
QB 120	KH= 8000	Job: "Print all configured messages"

6.6.8 Transferring Parameter Setting Data (Job $90XX_H$)

The parameter setting data of the serial interface is stored on the memory submodule in parameter blocks 0, 2 and 7 of data block (DB) 1. Configure the user submodule before operation.
During operation, you can do the following with job number 90_H:

- Set new parameters for the serial interface (parameter block 0):
  Job 9000_H
- Set the XON/XOFF parameters (parameter block 2):
  Job 9020_H
- Set the operating mode and parameters (parameter block 7):
  Job 9070_H Print mode
  9071_H Transparent ASCII mode
  9072_H Interpretive ASCII mode

Operator inputs required:
- Specify the parameters of the parameter block to be changed in bytes 2 to 7
- Specify the number of the parameter block to be changed and the job in byte 0.

Notes:

The "Transfer parameter setting data" job is more or less insignificant in print mode since the configuration data for print mode is stored on the memory submodule in DB 1.
A typical practical application would be if you want to replace a defective printer at short notice by another printer whose interface must be differently initialized.
The job does not change the contents of the memory submodule on the CP 521.
After warm restart, the CP uses the data of the memory submodule that is plugged in.
See 7.3.1 for a more detailed description.

Table 6-21. PIQ in the Case of the "Setting the Serial Interface Parameters"Job

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job	90_H	-
1	Number of the parameter block	00_H	-
2	Baud rate 110 Bd 200 Bd 300 Bd 600 Bd 1200 Bd 2400 Bd 4800 Bd 9600 Bd	 01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	04_H
3	Parity even odd ”mark” (parity bit always ”1”) ”space” (parity bit always ”0”) None	 00_H 01_H 02_H 03_H 04_H	00_H
4	BUSY signal No Yes	 00_H 01_H	00_H
5	Interface TTY RS-232-C (V.24)	 00_H 01_H	00_H
6	Data format 7 data bits 8 data bits	 00_H 01_H	00_H
7	HW handshake OFF ON	 00_H 01_H	00_H

Table 6-22. PIQ in the Case of "Setting the XON/XOFF Listing Parameters Job"

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job	90_H	-
1	Number of the parameter block	20_H	-
2	XON/XOFF protocol XON character No listing	 00_H to $7F_H$ FF_H	$FFFF_H$ (No XON/XOFF-protocol)
3	XON/XOFF protocol XOFF character No listing	 00_H to $7F_H$ FF_H	
4...7	Without meaning	00_H to FF_H	-

Table 6-23. PIQ in the Case of the "Setting Print Mode" Job

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job	90_H	-
1	Number of the parameter block (7) and operating mode code (0) for "Print mode"	70_H	0: if memory submodule with message texts plugged in 1: without memory submodule
2...7	Without meaning	00_H to FF_H	0001_H=100 ms

Table 6-24. PIQ in the Case of "Transparent ASCII Mode" Job

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job	90_H	-
1	Number of the parameter block (7) and operating mode code (1) for "Transparent ASCII mode"	71_H	0: If memory submodule with message texts plugged in 1: without memory submodule
2+3	Character delay time	0001_H to $0BB8_H$	0001_H : 100 ms
4+5	Message length (in bytes)	0001_H to 0100_H	40_H : 64
6+7	Without meaning	00_H to FF_H	

Table 6-25. PIQ in the Case of the "Interpretive ASCII Mode" Job

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job	90_H	-
1	Number of the parameter block (7) and operating mode code (2) for "Interpretive ASCII mode"	72_H	0: If memory submodule with message texts plugged in 1: without memory submodule
2+3	Character delay time	0001_H to $0BB8_H$	0001_H : 100 ms
4+5	Message length (in bytes)	0000_H to 0100_H	40_H : 64
6	End-of-text character 1 — ASCII character No end-of-text character	01_H to $7F_H$ 00_H	
7	End-of-text character 2 — ASCII character No end-of-text character	01_H to $7F_H$ 00_H	

CAUTION:

The "Transfer parameter setting data" job does not change the contents of the memory submodule plugged into the CP 521.
After POWER UP, the CP 521 uses the data of the connected memory submodule that is plugged in.

CAUTION:

The following two examples must be executed in the following order:

1. Disconnect the TTY connecting cable between the CP and the printer.
2. Set printer to new data
   - RS-232-C (V.24) interface (example 1a)
   - XON/XOFF protocol (example 1b)
3. Transfer new data with programmer function to the CP
4. Reconnect CP and printer with RS-232-C (V.24) connecting cable. See 3.5.1 for terminal diagram for RS-232-C (V.24) interface.

Example 1a: Setting the RS-232-C (V.24) interface with the "FORCE VAR" programmer function

The module is installed in slot 7 (initial address 120). An initialized memory submodule is plugged into the module.

A printer is now to be operated with:
- Busy signal yes
- TTY interface
- XON/XOFF listing yes

A printer is now to be operated with:
- RS-232-C (V.24) interface
- XON/XOFF listing yes

Operand	Signal States	Meaning
QW 126	KH= 0000	Byte 6: 7 data bits; byte 7: handshake OFF
QW 124	KH= 0001	Byte 4: Busy no; byte 5: RS-232-C (V.24) interface
QW 122	KH= 0800	Byte 2: 9600 baud; bytes 3: even parity
QW 120	KH= 9000	Byte 0: Job "Transfer parameter setting data" Byte 1: Parameter block 0

Example 1b: Setting the XON/XOFF listing

The same details apply as in example 1a.

Operand	Signal States	Meaning
QW 122	KH= 1113	Byte 2: XON character (DC 1 as ASCII code) Byte 3: XOFF character (DC 3 as ASCII code)
QW 120	KH= 9020	Byte 0:"Transfer parameter setting data" job Byte 1: Parameter block 2

7 ASCII Mode

7.1 General . . . . . 7 - 1

7.2 Prerequisites for Operation in ASCII Mode . . . . . 7 - 2

7.3 Setting the CP 521 Parameters in ASCII Mode . . . . . 7 - 5
7.3.1 Setting the CP 521 Parameters with Job $90XX_H$ . . . . . 7 - 6
7.3.2 Setting the CP 521 Parameters with the Memory Submodule . . . . . 7 - 14

7.4 CPU Job Requests with CP Responses in ASCII Mode . . . . . 7 - 19
7.4.1 Sending Messages . . . . . 7 - 20
7.4.2 Receiving Messages . . . . . 7 - 39

7.5 CPU Job Requests and CP Error Messages in ASCII Mode . . . . . 7 - 54
7.5.1 Permissible Job Requests in ASCII Mode . . . . . 7 - 54
7.5.2 Illegal Jobs with Error Messages . . . . . 7 - 55
7.5.3 Further Error Messages . . . . . 7 - 57

Figures

7-1. Job Transfer and Error Evaluation . . . . . 7 - 6
7-2. Schematic for Entering Parameter Blocks in DB 1 . . . . . 7 - 14
7-3. Sequence Schematic for Sending Data . . . . . 7 - 22
7-4. Sequence Schematic when Receiving Data . . . . . 7 - 41

Tables

7-1. Parameter Blocks in ASCII Mode . . . . . 7 - 5
7-2. PIQ for the ”Transfer the Parameter setting data for parameter block 0” Job . . . . . 7 - 7
7-3. PIQ for the ”Transfer the parameter setting data for parameter block 2” Job . . . . . 7 - 8
7-4. PIQ for the ”Transfer the parameter setting data for parameter block 7” Job (Print Mode) . . . . . 7 - 9
7-5. PIQ for the ”Transfer the parameter setting data for parameter block 7” Job - (Transparent ASCII Mode) . . . . . 7 - 9
7-6. PIQ for the ”Transfer the parameter setting data for parameter block 7” Job - (Interpretive ASCII Mode) . . . . . 7 - 10
7-7. Assignments for Parameter Block 0 . . . . . 7 - 15
7-8. Assignments for Parameter Block 2 . . . . . 7 - 16
7-9. Assignments for Parameter Block 7 (Transparent ASCII Mode) . . . . . 7 - 16
7-10. Assignments for Parameter Block 7 (Interpretive ASCII Mode) . . . . . 7 - 17
7-11. Setting the CP 521 Parameters in Interpretive ASCII Mode . . . . . 7 - 18
7-13. Permissible CPU Job Requests when Sending Messages . . . . . 7 - 21

Tables (Continued)

7-14. Coordination Job . . . . . 7. - 24
7-15. Coordination Information . . . . . 7 - 25
7-16. Data Transfer: Sending the 1st Data Block . . . . . 7 - 27
7-17. Acknowledging the 1st Data Block . . . . . 7 - 28
7-18. Data Transfer: Sending the 2nd Data Block . . . . . 7 - 29
7-19. Acknowledging the 2nd Data Block . . . . . 7 - 29
7-20. Data Transfer: Sending the 43rd Data Block . . . . . 7 - 30
7-21. Acknowledging the Last (43rd) Data Block . . . . . 7 - 31
7-22. Final Coordination Information . . . . . 7 - 31
7-23. Coordination Job . . . . . 7. - 35
7-24. Permissible CPU Job Requests when Receiving Data (PIQ) . . . . . 7. - 40
7-25. Coordination Job . . . . . 7. - 42
7-26. Data Transfer: Receive the 1st Data Block . . . . . 7 - 43
7-27. Acknowledging the 1st Data Block . . . . . 7 - 44
7-28. Data Transfer: Receive the 25th (Last) Data Block . . . . . 7 - 45
7-29. Acknowledgement for the 25th Data Block . . . . . 7 - 46
7-30. Final Coordination Information . . . . . 7 - 47
7-31. Permissible Jobs in ASCII Mode . . . . . 7 - 54
7-32. Message: Invalid Job (PII) . . . . . 7 - 55
7-33. Message: No Receive Message Present (PII) . . . . . 7 - 56
7-34. Message: Error in Receive Message (PII) . . . . . 7 - 56
7-35. Status Byte in ASCII Mode (PII) . . . . . 7 - 57

7 ASCII Mode

In ASCII mode with a CPU job request, the CP 521 allows transfer of a message of up to 256 bytes between the CPU and a peripheral device connected to the CP 521:

- Communication with a terminal device (terminal, keyboard ...)
- Point-to-point connection with a further CP 521, a CP 523 or a CPU 944

7.1 General

You can choose between:

- Transparent ASCII mode
  The CP 521 does not interpret any characters in transparent ASCII mode.
  - No XON/XOFF protocol is possible.
  - Messages can only be sent or received with a fixed programmed length.
- interpretive ASCII mode
  The CP 521 evaluates the following characters in interpretive ASCII mode:
  - RUB OUT ($7F_H$)
  - BACKSPACE (08_H)
  - XON/OFF character (if initialized)
  - Character end code (if initialized)

The CP 521 handles data transfer with the peripheral device autonomously.
The CPU initiates data exchange between the CPU and the CP by sending a job. See 7.4 for a precise description of the data exchange procedure.

The time of day can be read from the integrated clock by the CPU also in ASCII mode and used in the user program for date-dependent and time-dependent tasks. If the time is not used in the control, the battery is not needed. No memory submodule is required in ASCII mode.

7.2 Prerequisites for Operation in ASCII Mode

The following conditions must be met for operating the CP 521:

1. Settings on the peripheral device
The settings on the peripheral device must agree with the parameter setting data on the CP 521. If, for example, your peripheral device is transmitting data at 2400 baud, you must also set this baud rate in the CP 521.

Note:

- See 3.5.2 for configuration examples with terminal diagrams.
- See 2.3.2 for details of the serial interface.
- Setting the CP 521 parameters in ASCII mode is explained in 7.3.

2. Setting the CP 521 parameters
There are various ways of setting the CP 521 parameters:

- Transfer the parameter setting data in the user program. No memory submodule is required.
- Transfer the parameter setting data using the "FORCE VAR" programmer function. No memory submodule is required.
- Store the parameter setting data in a memory submodule in DB 1.
  Initialize the memory submodule using a programmer in offline mode.

You must specify the following parameter setting data:

- Parameters for the interface to the peripheral device (baud rate, type of interface, handshake mode ...)
  The parameters must agree with the specifications and settings on the peripheral device.

3. Establishing connections

The CPU must be at POWER OFF to carry out the following measures:

- Establish connection between the CP 521 and the peripheral device.
- If you have stored the parameter setting data in a memory submodule, plug the memory submodule into the CP 521.

You can then switch the CPU to POWER ON.

4. Startup of the CP 521 in ASCII mode

After POWER ON the CP 521 is automatically in transparent ASCII mode if the following applies:

- No memory submodule is plugged in
- A memory submodule initialized with transparent ASCII mode is plugged in.
- Transparent ASCII mode has been transferred to the CP 521 from the user program.

You can set interpretive ASCII mode in the following ways:

- Transfer interpretive ASCII mode to the CP 521 in the user program
- Plug a memory submodule initialized with interpretive ASCII mode into the CP 521.

Changing the mode (during operation)

You can alternate between "Print mode", "Transparent ASCII mode" and "Interpretive ASCII mode" during operation.

Note:
Note that as soon as you either
- Change from one mode to another

or
- Call up the programmed mode again

via the user program, the entire send and receive buffer is deleted.

To change the mode, transfer job 90XXH ”Transfer configuration data", to the CP 521.
Remember that the parameters must agree with the settings on your peripheral device.

7.3 Setting the CP 521 Parameters in ASCII Mode

The CP 521 is supplied with default values for the parameter setting data.

- Parameters for the interface to the peripheral device (baud rate, type of interface, handshake mode ...)
- Specifications of the XON/XOFF protocol
- Parameter setting data for data transfer in ASCII mode (message length, end--of-text character).

If you want to set other CP 521 data, there are the following other ways of transferring parameter setting data to the CP:

- Transfer the parameter setting data to the CP 521 in the user program (7.3.1)
  No memory submodule is required for operating the CP 521
- Store the parameter setting data on a memory submodule in data block (DB) 1 and plug the initialized memory submodule into the CP 521 (7.3.2).
- Transfer the parameter setting data with the "FORCE VAR" programmer function.
  No memory submodule is requirerd to operate the CP 521.

In order to transfer the parameter setting data as simply as possible, the parameter setting data are divided into parameter blocks. This subdivision is the same as in print mode. Parameter blocks 1, 3, 4, 5 and 6 are not significant in ASCII mode and are not included in the table.

Table 7-1. Parameter Blocks in ASCII Mode

Parameter Block No.	Contents
0	Serial interface parameters
2	XON/XOFF protocol (only relevant if XON/XOFF protocol operated) The XON/XOFF protocol has priority over the BUSY signal
7	Desired mode Parameter setting data for data transfer in ASCII mode

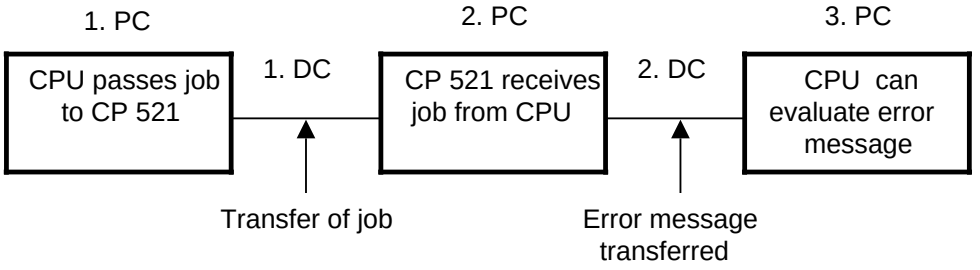
7.3.1 Setting the CP 521 Parameters with Job $90XX_H$

You can use the "Transfer parameter setting data" job to transfer data for a parameter block to the CP 521.

Note:

You can only initialize parameter blocks 0, 2 and 7 with the "Transfer parameter setting data" job. Parameter blocks 1, 3, 4, 5 and 6 are only significant in print mode. They cannot be initialized from the user program.

When evaluating data from the CP 521 (PII), you must take into account that you will not receive the correct response to a CPU job request (PIQ) until two cycles after initiation of the job.



PC=program cycle, DC = data cycle

Figure 7-1. Job Transfer and Error Evaluation

This has the following significance for parameter setting with the user program:

- At least seven program cycles are required for setting parameters in the three parameter blocks
- If errors occur (error message 08_H), correspondingly more program cycles are required.

The following pages contain

- The PIQ assignment plan for the "Transfer the parameter setting data for parameter block X" jobs
- Example program for transferring parameter setting data

Parameter setting data for parameter block 0

Table 7-2. PIQ for the "Transfer the parameter setting data for parameter block 0" Job

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Number of the parameter block	00_H	-
2	Baud rate 110 Bd 200 Bd 300 Bd 600 Bd 1200 Bd 2400 Bd 4800 Bd 9600 Bd	 01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	04_H
3	Parity even odd ”mark” (parity bit always ”1”) ”space” (parity bit always ”0”) none	 00_H 01_H 02_H 03_H 04_H	00_H
4	BUSY signal no yes	 00_H 01_H	00_H
5	Interface TTY RS-232-C (V.24)	 00_H 01_H	00_H
6	Data format 7 data bits 8 data bits	 00_H 01_H	00_H
7	HW handshake OFF ON	 00_H 01_H	00_H

Parameter setting data for parameter block 2

Table 7-3. PIQ for the "Transfer the parameter setting data for parameter block 2" Job

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Number of the parameter block	20_H	-
2	XON/XOFF protocol XON character no protocol	00_H to $7F_H$ FF_H	$FFFF_H$
3	XON/XOFF protocol XOFF character no protocol	00_H to $7F_H$ FF_H	(no XON/XOFF protocol)
4 to 7	Not significant	00_H to FF_H	-

Parameter setting data for parameter block 7

Table 7-4. PIQ for the "Transfer the parameter setting data for parameter block 7" Job (Print Mode)

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number and code for "Print mode"	70_H	0: if memory sub-module with message texts plugged in 1: without memory sub-module
2 to 7	Not significant	00_H to FF_H	

Table 7-5. PIQ for the "Transfer the parameter setting data for parameter block 7" Job- (Transparent ASCII Mode)

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number and code for "Transparent ASCII mode"	71_H	0: if memory sub-module with message texts plugged in 1: without memory sub-module
2+3	Character delay time	0001_H to $0BB8_H$	0001_H : 100 ms
4+5	Message length (in bytes)	0001_H to 0100_H	40_H : 64
6+7	Not significant	00_H to FF_H	

Table 7-6. PIQ for the "Transfer the parameter setting data-for parameter block 7" Job (Interpretive ASCII Mode)

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number and code for "Interpretive ASCII mode"	72_H	0: if memory sub-module with message texts plugged in 1: without memory submodule
2+3	Character delay time	0001_H to $0BB8_H$	0001_H : 100 ms
4+5	Message length (in bytes)	0000_H to 0100_H	40_H : 64
6	End-of-text character 1 ASCII character No end-of-text character	01_H to $7F_H$ 00_H	00_H
7	End-of-text character 2 ASCII character No end-of-text character	01_H to $7F_H$ 00_H	$0D_H$*

* $0D_H$ = carriage return

Example: Setting parameter data in the user program for operation in ASCII mode using the PG 685

The CP 521 is plugged into in slot 7 (initial address 120). The module is to be operated in interpretive ASCI mode and is initialized as follows:

- Parameters for the serial interface (parameter block 0)
  - 2400 baud (6)
  - Even parity (0)
  - No BUSY (0)
  - RS-232-C (V.24) interface (1)
  - 7 Datenbits (0)
  - Handshake OFF (0)

- Parameters for XON/XOFF (parameter block 2)
  - XON character: DC 1 (11_H)
  - XOFF character: DC 3 (13_H)

- Parameters for ASCII mode (parameter block 7)
  - Character delay time 3 s ($001E_H$)
  - Variable message length with two characters for end code ($0D0D_H$)

Example: Transferring parameter setting data for parameter blocks 0, 2 and 7

The CP 521 is plugged into in slot 7 (initial address 120). The module is to be operated in interpretive ASCI mode and is initialized as follows:

OB 1	FB 60	Explanation
:AN F 100.0 :JC FB 60 NAME :BLOCK0 :AN F 100.0 :S F 100.0 :BEC :AN F 100.1 :JC FB 61 NAME :BLOCK2 :AN F 100.1 :S F 100.1 :BEC	NAME :BLOCK0 :L KH 0000 :T QW 126 :L KH 0001 :T QW 124 :L KH 0600 :T QW 122 :L KH 9000 :T QW 120 :BE **FB 61** NAME :BLOCK2 :L KH 1113 :T QW 122 :L KH 9020 :T QW 120 :BE	Load ASCII codes for XON/XOFF characters into ACCUM 1 and transfer to the CP in bytes 2 and 3. Load the job number and the parameter block number into ACCUM 1 and transfer to the CP in bytes 0 and 1.

OB 1 (continued)	FB 62	Explanation
:AN F 100.2 :JC FB 62 NAME :BLOCK7 :AN F 100.2 :S F 100.2 :BE	NAME :BLOCK7 :L KH 0D0D :T QW 126 :L KH 0000 :T QW 124 :L KH 001E :T QW 122 :L KH 9072 :T QW 120 :BE	 Load end codes into ACCUM 1 and transfer to the CP in bytes 6 and 7 Load message lengths into ACCUM 1 and transfer to the CP in bytes 4 and 5 Load character delay time into ACCUM 1 and transfer to the CP in bytes 2 and 3 Load job number (90_H), parameter block number (7_H) and mode code (2_H) into ACCUM 1 and transfer to the CP

7.3.2 Setting the CP 521 Parameters with the Memory Submodule

You can store the parameter setting data on a memory submodule in DB 1.

Note:

The procedure for setting the memory submodule parameters in ASCII mode is almost identical to the procedure in print mode. The only difference is in setting the parameters for parameter block 7 in which additional data must be specified in ASCII mode.

Entering the parameter blocks

Using the programmer (PG), you must first enter in DB 1 all those parameter blocks which deviate from the default values. Then transfer DB 1 to the memory submodule.
Further tips for input:

- When entering text, change the data format from KS to C in alternate lines.
- You can enter comments in addition to the parameter blocks.

Schematic for entering parameter blocks
Parameter blocks can be entered in DB 1 according to the following schematic.

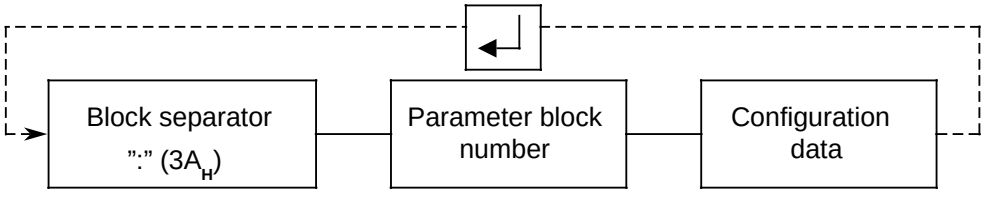


Figure 7-2. Schematic for Entering Parameter Blocks in DB 1

The following pages contain the assignment of the parameter blocks and an input example.

Note:

Parameter block assignment for setting parameters with the memory submodule corresponds to PIQ assignment when setting parameters with the user program.

Table 7-7. Assignments for Parameter Block 0

Block	Meaning	Value Range	Default Values on the CP 521
0	Baud rate 110 Bd 200 Bd 300 Bd 600 Bd 1200 Bd 2400 Bd 4800 Bd 9600 Bd	1 2 3 4 5 6 7 8	4
	Parity Even Odd ”mark” (parity bit always "1") ”space” (parity bit always "0") None	0 1 2 3 4	0
	BUSY signal no yes	0 1	0
	Interface TTY RS-232-C (V.24)	0 1	0
	Data format 7 data bits 8 data bits	0 1	0
	HW handshake OFF ON	0 1	0

Table 7-8. Assignments for Parameter Block 2

Block	Meaning	Value Range	Default Values on the CP 521
2	XON/XOFF protocol XON character no protocol	00_H to $7F_H$ FF_H	$FFFF_H$
	XON/XOFF protocol XOFF character no protocol	00_H to $7F_H$ FF_H	(no XON/XOFF protocol)

***** only possible in interpretive ASCII mode

Table 7-9. Assignments for Parameter Block 7 - (Transparent ASCII Mode)

Block	Meaning	Value Range	Default Values on the CP 521
7	Parameter block number and code for "Transparent ASCII mode"	71	0: if memory sub-module with message texts plugged in 1: without memory submodule
	Character delay time (four decimal digits)	0001 to 3000	0001 : 100 ms
	Message length (three decimal digits)	001 to 256	064

Table 7-10. Assignments for Parameter Block 7 (Interpretive ASCII Mode)

Block	Meaning	Value Range	Default Values on the CP 521
7	Parameter block number and code for "Interpretive ASCII mode"	72	
	Character delay time (four decimal digits)	0001 to 3000	
	Message length (three decimal digits)	000 to 256	
	End-of-text character 1 ASCII character / No end-of-text character	01_H to $7F_H$ / 00_H	
	End-of-text character ASCII character / No end-of-text character	01_H to $7F_H$ / 00_H	

* OD_H = carriage return

Example: Using the PG 685 to set data parameters on the memory submodule for operating in ASCII mode

The CP 521 is plugged into in slot 7 (initial address 120). The module is to be operated in interpretive ASCII mode and is initialized as follows:

- Parameters for the serial interface (parameter block 0)
  - 2400 baud (6)
  - Even parity (0)
  - No BUSY (0)
  - RS-232-C (V.24) interface (1)
  - 7 data bits (0)
  - Handshake OFF (0)
- Parameters for XON/XOFF (parameter block 2)
  - XON character: DC1 (11_H)
  - XOFF character: DC3 (13_H)
- Parameters for ASCII mode (parameter block 7)
  - Character delay time 3 s (0030)
  - Variable message length with two characters for end code ($0D0D_H$)

We recommend the following procedure:

1. Enter the parameter setting data on the programmer in DB 1 and include sufficient comments.
2. Store in DB 1 on diskette or hard disk
3. Transfer DB 1 to the memory submodule

Table 7-11. Setting the CP 521 Parameters in Interpretive ASCII Mode

Parameter Block Number	Contents

`: KS = 'Parameters for the serial';`
`: C = ' interface';`
`: KS = ':0600100';`
	:0 : Parameter block 0
	6 : 2400 baud
	0 : Even parity
	0 : No BUSY
	1 : RS-232-C (V.24) interface
	0 :7 data bits
	0 : Handshake OFF

`: C = 'Parameters for XON/XOFF';`
`: KS = ':2;`
`: KH = 1113';`
`: C = 'Parameters for ASCII';`
`: KS = 'mode';`
	:2 : Parameter block 2
	11_H : DC1 (XON character)
	13_H : DC3 (XOFF character)

`: C = ':720030000;` `: KH = 0D0D';`	:7 : Parameter block 7 2 : Interpretive ASCII mode 0030 : Character delay time 30 x 100 ms 000 : Message length 0 $0D0D_H$: End-of-text character (carriage return twice)

7.4 CPU Job Requests with CP Responses in ASCII Mode

The CPU sends and receives messages in eight-byte blocks. The messages have a maximum length of 256 bytes. Each block consists of a "header code" of two bytes (coordination bytes) and six bytes of data.
Maximum data throughput is 6 bytes of useful data per two program scans. This means that a maximum of 60 bytes/s can be transferred given a program scan time (OB 1) of e.g. 50 ms.

Note:

Coordination jobs for sending and receiving messages must be sent in separate jobs. They must not be mixed in the same job.
Send and receive jobs must also be programmed in such a way that each new job is only transferred when the previous one has been completely processed.

7.4.1 Sending Messages

The CPU transfers messages in eight-byte blocks. The messages have a maximum length of 256 bytes. Each block consists of a header code of two bytes (coordination bytes) and six bytes of data.

1. The CPU initiates data transfer with the job $A001_H$. This job also defines the message length. If it is indicated with 0, one or two end-of-text characters must be initialized .

2. The CP 521 acknowledges receipt of this job to the CPU .

3. The CPU evaluates the CP response and begins transferring the current data only if no errors have been detected. Transfer is started with job "$B0_H$" and entry of a block number . Six bytes of data follow. The last data block transferred may contain less than six bytes.

4. The CP 521 writes the block into a send mailbox where it first stores the whole message. The CPU receives an acknowledgement from the CP 521 after every eight bytes .

5. The CPU then sends the next data block to the CP, and so on. This data transfer continues until all data blocks of a message have been transferred. A message can have a maximum length of 256 bytes.

6. The CP 521 gathers the data blocks together into a message and transfers the message automatically to a peripheral device.

Note:

Only one message can be stored in the send mailbox of the CP 521. The CPU can only send the next message to the CP when this message has been transferred to the peripheral device.

Sending fixed-length messages

The following is a description of which jobs the CPU uses when prompting the CP 521 to send messages and how the CP 521 acknowledges these jobs. Table 7-13 contains all the CPU jobs permissible in ASCII mode. After setting the interface parameters (90_H) (7.3), jobs $A0_H$ and $B0_H$ are relevant for sending data.

Table 7-13. Permissible CPU Job Requests when Sending Messages (PlQ)

Byte 0	Meaning
00_H	Data not relevant
10_H	Set date and time of day; variable in bytes 1 to 7
90_H	Set interface parameters
$A0_H$	Coordinate data transfer
$B0_H$	Process data transfer CPU CP

Data is transferred via the CP 521 in two steps:

- First, data transfer between the CPU and the CP 521 is coordinated
- Only then is data transferred.

The schematic on the following two pages shows the overall data transfer procedure for sending data.

Data blocks of eight bytes are represented as used for sending messsages. The CPU job varies from block to block and also the corresponding acknowledgement from the CP 521:

Data transfer from the CPU to the CP 521

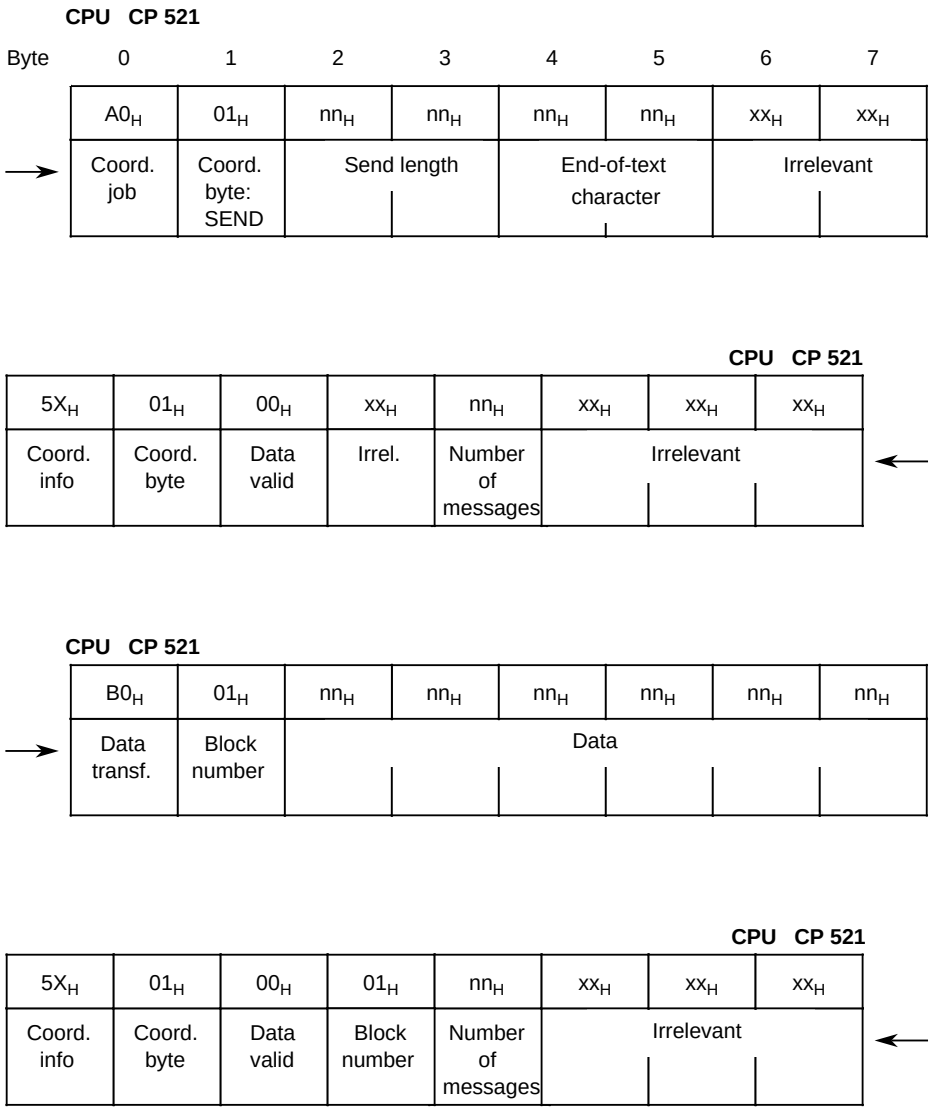
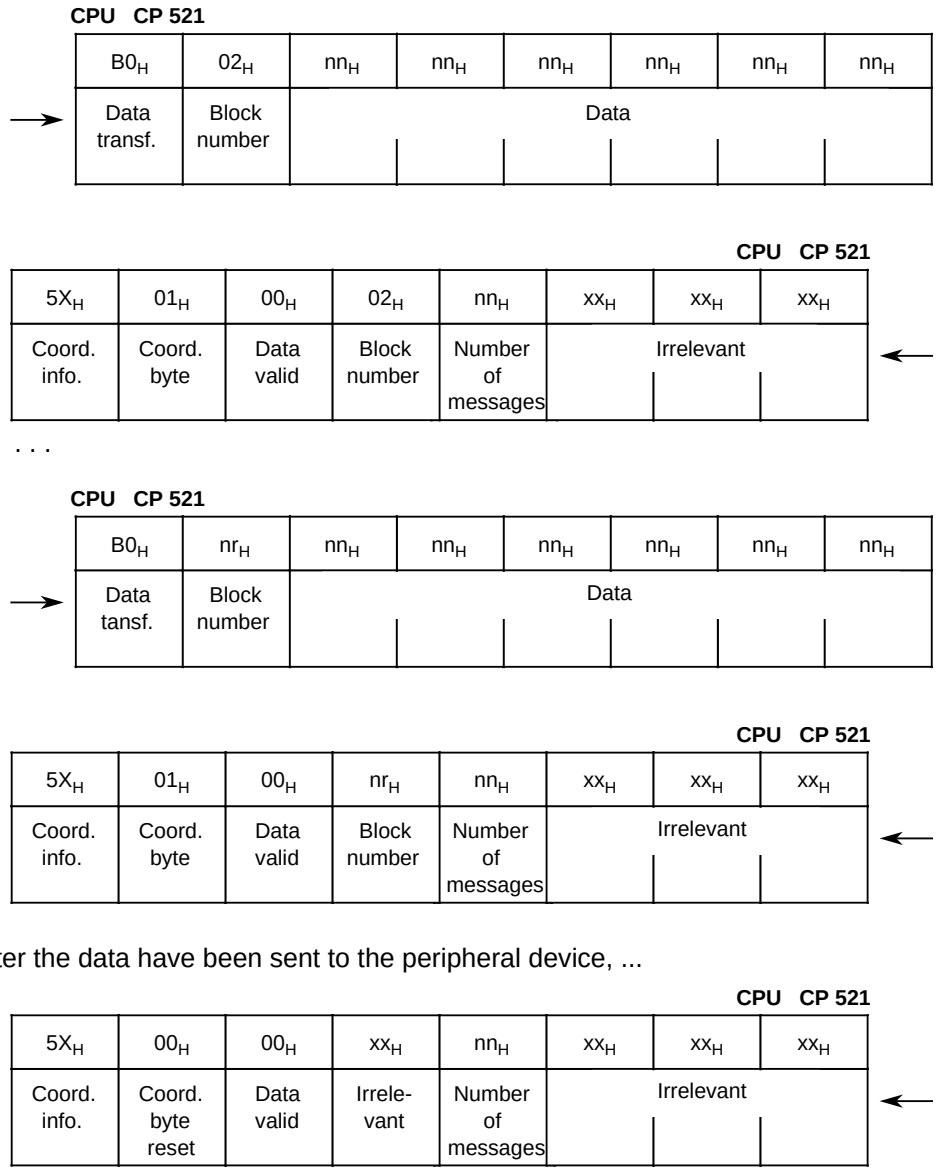


Figure 7-3. Sequence Schematic for Sending Data



... after the data have been sent to the peripheral device, ...

Figure 7-3. Sequence Schematic for Sending Data (Continued)

Steps to are explained more fully in the following.

CPU job request: Coordinate data transfer

To coordinate data transfer, job $A0_H$ is written into byte 0 of the PIQ. This defines the meaning of bytes 1 to 7.

Table 7-14 Coordination Job

Byte	Value	Meaning
0	$A0_H$	Job: Coordinate data transfer
1	01_H	Send message (send bit)
2	01_H	Send length (here: 256 bytes)
3	00_H	Send length (here: 256 bytes)
4	00_H	Not significant
5	00_H	Not significant
6	Irrelevant	
7	Irrelevant	

Byte 1: You allocate send permission for a message by setting bit 0 = 1. This initiates data transfer. The CP 521 resets this bit after passing the message on to the peripheral device.

Bytes 2 u. 3: Bytes 2 and 3 indicate the message length. The length can lie within the range 0001_H to 0100_H. You must have set this length on the receive device because the send length is not transferred, in contrast to the end-of-text character.

In the case of a message length of 0, the CP evaluates the end-of-text character. If both send length and end-of-text character are 0, there is a job error.

If data is to be sent from the CPU to a peripheral device, the parameters "Send length" and "End-of-text character" must be transferred with the send job.

When sending data with end-of-text characters, these end-of-text characters are transferred also in both directions, but the send length is not transferred. For this reason the CP 521 setting must agree with the setting on the peripheral device.

CP response to job $A001_H$

When the CP 521 receives the "Coordinate data transfer" job $A0_H$ with send bit 01H ("Send message") set, and if there is no error, the CP 521 acknowledges the CPU as follows (PII):

Table 7-15. Coordination Information

Byte	Value	Meaning
0	$5X_H$	Acknowledgement: Coordination information
1	01_H	"Send" coordination bit set
2	00_H / FF_H	Coordination data valid/invalid
3	Irrelevant	
4	00_H to 64_H	Number of messages in receive mailbox of the CP (messages from the peripheral devices) to be read by the CPU (max. 100 messages)
5	Irrelevant	
6	irrelevant	
7	irrelevant	

Byte 0: Acknowledge job: Coordinate "Send" data transfer. Status messages and error messages can be stored in the lower half byte (7.5)

Byte 1: "Send" coordination bit set. If the coordination data is invalid, the CP 521 resets the "Send bit" to 00_H.

Byte 2: indicates whether the coordination data is valid (00_H) or invalid (FF_H).

Byte 4: indicates how many messages (max. 100_{Dec}) are stored in the receive mailbox of the CP.

As soon as data transfer has been coordinated (1 and 2), data transfer begins with job $B0_H$ (3 to 8).

Note:

The CP 521 usually transfers the date and time of day to the CPU in bytes 1 to 7 (6.6). If byte 0 has the value $5X_H$ or $6X_H$, you must not interpret these values as the time of day.

CPU job request: "Send data" ($B0_H$)

After you have output job $A0_H$ and received the corresponding acknowledgement from the CP 521, output job $B001_H$ with the 1st data block of the message (PIQ).

Table 7-16. Data Transfer: Sending the 1st Data Block

Byte	Value	Meaning
0	$B0_H$	Job: Send data
1	01_H	Number of the 1st data block
2		Data
3		
4		
5		
6		
7		

Byte 0: Job: Send data. After this job $B0_H$, the CP 521 interprets bytes 2 to 7 as data.

Byte 1: Indicates the number of the data block being transferred (here: 1). You must increment this block number in your user program - beginning at 01_H - each time you send a data block to the CP 521. This continues until the whole data message has been transferred to the CP 521.

CP response to job $B001_H$

The CP 521 acknowledges the "Send 1st data block of the message" job as follows (PII):

Table 7-17. Acknowledging the 1st Data Block

Byte	Value	Meaning
0	$5X_H$	Acknowledgment of job: Send data
1	01_H	"Send" coordination bit set if data valid
2	00_H	Data valid
3	01_H	Number of 1st data block
4	00_H - 64_H	Number of messages in receive buffer (max. 100)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

All the bytes in the data block except byte 3 have the same meaning as in Table 2.

Byte 3: Indicates the number of the data block which the CP 521 has just received (here: 1).

Note:

Please note that the CP 521 only responds to a signal change in byte 0 or 1 of the PIQ. Since job "$B0_H$" in byte 0 does not change as long as the data of a message is being sent, it is imperative that you incrementthe block numbers in byte 1 until all data blocks of the message have been transferred. If you send the wrong block number, data transfer is stopped with the response 5000H and the CP 521 then signals "Job error" ($4X_H$) in status byte 0 of the module. The data is lost.

If you send the next job B0H with incremented byte 1 (5), the CP 521 acknowldges with incremented byte 3 (6).

CPU job request: $B002_H$

Table 7-18. Data Transfer: Sending the 2nd Data Block

Byte	Value	Meaning
0	$B0_H$	Job: Send data
1	02_H	Number of the 2nd data block
2		Data
3		
4		
5		
6		
7		

CP response to job $B002_H$

Table 7-19. Acknowledging the 2nd Data Block

Byte	Value	Meaning
0	$5X_H$	Acknowledgement for job: Send data
1	01_H	"Send" coordination bit set if data valid
2	00_H	Data valid
3	02_H	Number of the 2nd data block
4	00_H - 64_H	Number of messages in the receive buffer (max. 100)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

If you send a message of variable length and you use the maximum message length, the last CPU job request (7) and the CP acknowledgement (8) appear as follows:

CPU job request : $B02B_H$

Table 7-20. Data Transfer: Sending the 43rd Data Block

Byte	Value	Meaning	
0	$B0_H$	Job:	Send data
1	$2B_H$		Data block number 43 (max.)
2		D a t a	
3		D a t a	
4		D a t a	(Byte 255)
5		D a t a	(Byte 256)
6			Irrelevant
7			Irrelevant

Note:

Since a message may not exceed 256 bytes, you can transfer a maximum of 43 data blocks in one message (42 of 6 bytes and 1 of 4 bytes). This leaves only four bytes for data in the last data block ($2B_H$).

CP response to job $B02B_H$

Table 7-21. Acknowledging the Last (43rd) Data Block

Byte	Value	Meaning
0	$5X_H$	Acknowledgement for job: Send data
1	01_H	"Send" coordination bit set if data valid
2	00_H	Coordination data valid
3	$2B_H$	Number of the 43rd data block (max.)
4	00_H ... 64_H	Number of messages in receive buffer (max. 100)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Final CP response

After the whole data message has been transferred from the CP 521 to the peripheral device, the CPU receives a last message from the CP 521. This resets the coordination bit.

Table 7-22. Final Coordination Information

Byte	Value	Meaning
0	$5X_H$	Acknowledging the job: Send data
1	00_H	Coordination bit reset
2	00_H	Data valid
3	Irrelevant	
4	00_H ... 64_H	Number of messages in the receive buffer (max. 100)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Example: Sending a fixed-length message

Let's assume you have the CP 521 in slot 7 (initial address 120) and you want to send a fixed-length message (12 bytes) to the peripheral device. You have stored the data for the message in data block (DB) 20.

STL FB 1		Explanation
NAME	:SENDEN	Send fixed length (12 bytes)
	:C DB 20	Open source DB
	:	
	:AN F 100.1	Flag 100.1 (e.g. edge flag) initiates data
	:AN F 80.0	transfer
	:BEC	
	:	
	:A F 80.0	If flag 80.0 is set (coordination job
	:	transferred to CP), jump to label 001.
	:	
	:JC =M001	The coordination job is transferred to the
	:L KH A001	CP in byte 0 and permission to send or
	:T QW 120	receive is transferred in byte 1.
	:	
	:	
	:L KH 000C	When sending with a specified length,
	:T AW 122	the length of the data message is entered
	:	in bytes 2 and 3.
	:	
	:S F 80.0	The coordination job has been initiated.
	:BEU	The block has been completed.
	:	
	:	
M001	:L IW 120	Read in of bytes 0 and 1
	:L KH F00F	If the coordination data transferred
	:AW	from the CPU to the CP are invalid
	:L KH 5001	(byte 1 = 00_H), jump to label 002.
	:><F	
	:	
	:	
	:JC =M002	

STL FB 1 (Continued)			Explanation (Continued)
	:		
	:L	IW 122	Read in of bytes 2 and 3
	:L	KH 0000	If the coordination data is invalid (after coordination job), or if a data block is acknowledged, jump to label 003.
	:><F		
	:		
	:		If the coordination data were invalid, the 1st data block is transferred.
	:JC	=M003	
	:		Entry in byte 0: $B0_H$
	:		(Data transfer from the CPU to the CP 521)
	:		Entry in byte 1: 01_H
	:L	KH B001	(Data block No. 1)
	:T	QW 120	The data is entered in bytes 2 to 7 and transferred to the CP.
	:		
	:		
	:L	DW 0	
	:T	QW 122	
	:L	DW 1	
	:T	QW 124	
	:L	DW 2	
	:T	QW 126	
	:BEU		Block end
	:		
M003	:L	IW 122	If data block No. 1 has not yet been acknowledged by the CP, jump to label 002
	:L	KH 0001	
	:><F		
	:JC	=M002	
	:L	KH B002	If data block No. 1 has been acknowledged by the CP, the second data block is transferred to the CP by the CPU.
	:T	QW 120	
	:L	DW 3	
	:T	QW 122	
	:L	DW 4	
	:T	QW 124	Block end
	:L	DW 5	
	:T	QW 126	
	:BEU		

STL FB 1 (Continued)	Explanation (Continued)
: M002 :L IW 120 :L KH F00F :AW :L KH 5000 :! = F : : :R F 80.0 : :BE	If data has been transferred from the CPU to the CP free of errors, and if the send length entered in the coordination job has been reached, flag 80.0 is reset.

Sending variable-length messages

Data transfer with variable-length messages is similar to transfer with fixed-length messages. They differ in that when sending with variable-length in conjunction with job "$A001_H$" (cf.: 1 sending messages) one or two end-of-text characters must be transferred in bytes 4 and 5.

Table 7-23. Coordination Job

Byte	Value	Meaning
0	$A0_H$	Job: Coordinate data transfer
1	01_H	Send message (send bit)
2	00_H	Message length (here: variable length)
3	00_H	Message length (here: variable length)
4	00_H	End-of-text character (only if message length = 0)
5	$7F_H$	End-of-text character (only if message length = 0)
6	Irrelevant	
7	Irrelevant	

Bytes 2 a. 3: You can define the length of a message here (0000_H to 0100_H). If you want to send variable-length messages, program 0000_H = variable length here. You must then define another one or two end-of-text characters in bytes 4 and 5.

Bytes 4 a. 5: You define your end-of-text characters here. If you want to use two end-of-text characters, fill both bytes. If you only use one end-of-text character, fill byte 5 only. Only one end-of-text character is used in the example.

Note:

If the message is longer than 256 data bytes ($<0100_H$), the CP 521 responds with 5000_H and then with message $4X_H$ "Job error". The CP 521 responds with the same error message if it detects no end-of-text character(s) after receiving 256 data bytes. The data received is then aborted.

Example: Sending a variable-length message (only for CPU 103)

You have plugged into the CP 521 in slot 7 (initial address 120) and you want to transfer a message of undefined length to a peripheral device. The data for the message is taken from DB 20. When the CP has accepted the send job, flag 0.0 is to be set. Both end-of-text characters must be in one data word.

STL FB 2	Explanation
NAME :SENDEN	Send with variable length
:C DB 20	Source data block
:	
:A F 100.1	Flag 100.1 (e.g. edge flag) initiates data
:R F 80.0	transfer
:	
:	
:A F 80.7	All data has been transferred to the CP
:JC =M001	Wait for CP acknowledgement
:	
:A F 80.0	
:JC =M002	
:L KH A001	Transfer "Send" coordination
:T QW 120	information to the CP
:L KH 0000	Message length
:T QW 122	
:L KH 0D0D	End-of-text character
:T QW 124	
:S F 80.0	
:	
:	Auxiliary register preset
:L KH 0000	Reset block acknowledgement
:T FW 22	
:L KH B001	Transfer of the first data block
:T FW 24	
:L KF +0	Set pointer in source DB to its initial value
:T FW 30	
:L KF +1	
:T FW 32	
:L KF +2	
:T FW 34	
:	
:BEU	

STL FB 2 (Continued)	Explanation (Continued)
: M002 :L IW 120 :L KH F00F :AW :L KH 5001 :><F :JC =M001 :L IW 122 :L FW 22 :><F :JC =M001 :L FW 24 :T QW 120 :B FW 30* :L DW 0 :T QW 122 :B FW 32* :L DW 0 :T QW 124 :B FW 34* :L DW 0 :T QW 126 :L QW 122 :L KH 0D0D :!=F :S F 80.7 :L QW 124 :L KH 0D0D :!=F :S F 80.7 :L QW 126 :L KH 0D0D :!=F :S F 80.7 :L FW 22 :L KF +1 :+F :T FW 22	Has the CP acknowledged the transferred data block? Check of the number of the acknowledged data block Transfer of the next data block with the relevant number Read out the data byte to be transferred from the source DB and transfer to the CP. If the data byte to be transferred contained the end code "0D0D", wait for acknowledgement "5X00" from the CP. Increment by 1 the block number in the data transfer job "BONR" (FW 24), the block number in the block acknowledgement "5X01" (FW 22) and the pointer in the source DB.

STL FB 2 (Continued)	Explanation (Continued)
:L FW 24 :L KF +1 :+F :T FW 24 :L FW 30 :L KF +3 :+F :T FW 30 :L FW 32 :L KF +3 :+F :T FW 32 :L FW 34 :L KF +3 :+F :T FW 34	
M001 :L IW 120 :L KH F00F :AW :L KH 5000 :><F :JC =M003	Acknowledgement from CP after transfer of the last data block
: : :	If the acknowledgement from the CP is positive, transfer of the next data package is prepared.
:L KH 0000 :T FW 22	Reset block acknowledgement
:L KH B001 :T FW 24	Transfer of the first data block
:L KF +0 :T FW 30	Set pointer in source DB to its initial value.
:L KF +1 :T FW 32 :L KF +2 :T FW 34	
: : :R F 80.7	* The FB only runs with the CPU 103; the operation: "Process flag word" (DO FW) is not included in the operation sets of the CPU 100 and the CPU 102.
M003 :BE	

7.4.2 Receiving Messages

If you want to receive data from a peripheral device using the CP 521, you must program both devices with the same message length or end-of-message character.

The periheral device sends messages to the CP 521 where they are collected in one receive mailbox. This receive buffer has a capacity of 1 Kbyte and can store 100 messages.

You program the CPU so that it reads out the existing messages from the receive mailbox in eight-byte blocks (two job bytes and six data bytes):

1. The CPU sends a job: Coordinate data transfer "Receive" to CP 521. You have previously set the message length, end-of-message character and permissible character delay time on the CP in the relevant parameter blocks.
2. The CP 521 starts data transfer. It sends the first six bytes of the "oldest" message in the receive buffer (FIFO memory).
3. The CPU acknowledges the data received.
4. The CP 521 sends a further block, and so on until the whole message has been transferred from the CP 521 to the CPU.

The CPU receives an immediate final acknowledgement (6) $5X00_H$ to the "Coordinate receive" job ($A080_H$) if no receive message is present.

Receiving fixed-length messages

If you want to receive fixed-length data messages from a peripheral device using the CP 521, you must define the message length with which both devices are to work. If the CP is to receive data, it must be configured with the same length as the sending peripheral device since the send length is not transferred.

The following is a description of which jobs the CPU uses to allow the CP 521 to receive data from peripheral devices and how the CP 521 acknowledges these jobs. After you have set the interface parameters (90_H) (7.3), jobs A0 and C0 are relevant for receiving data (Table 7-24).

Table 7-24. Permissible CPU Job Requests when Receiving Data (PIQ)

Byte 0	Meaning
00_H	Data not relevant
$A0_H$	Coordinate data transfer
$C0_H$	Acknowledge CPU via data received

Data is exchanged via the CP 521 in two steps:

- First, data transfer between the CPU and the CP 521 is coordinated.
- Only then can the CP 521 transfer data to the CPU which it has received from peripheral devices and buffered in the receive mailbox.

Figure 7-4 illustrates the whole data transfer sequence when receiving data messages.

Data blocks of eight bytes each are represented as they are used when receiving messages. The CPU job request and the relevant CP 521 acknowledgement change from block to block.

Data transfers: from the CP 521 to the CPU

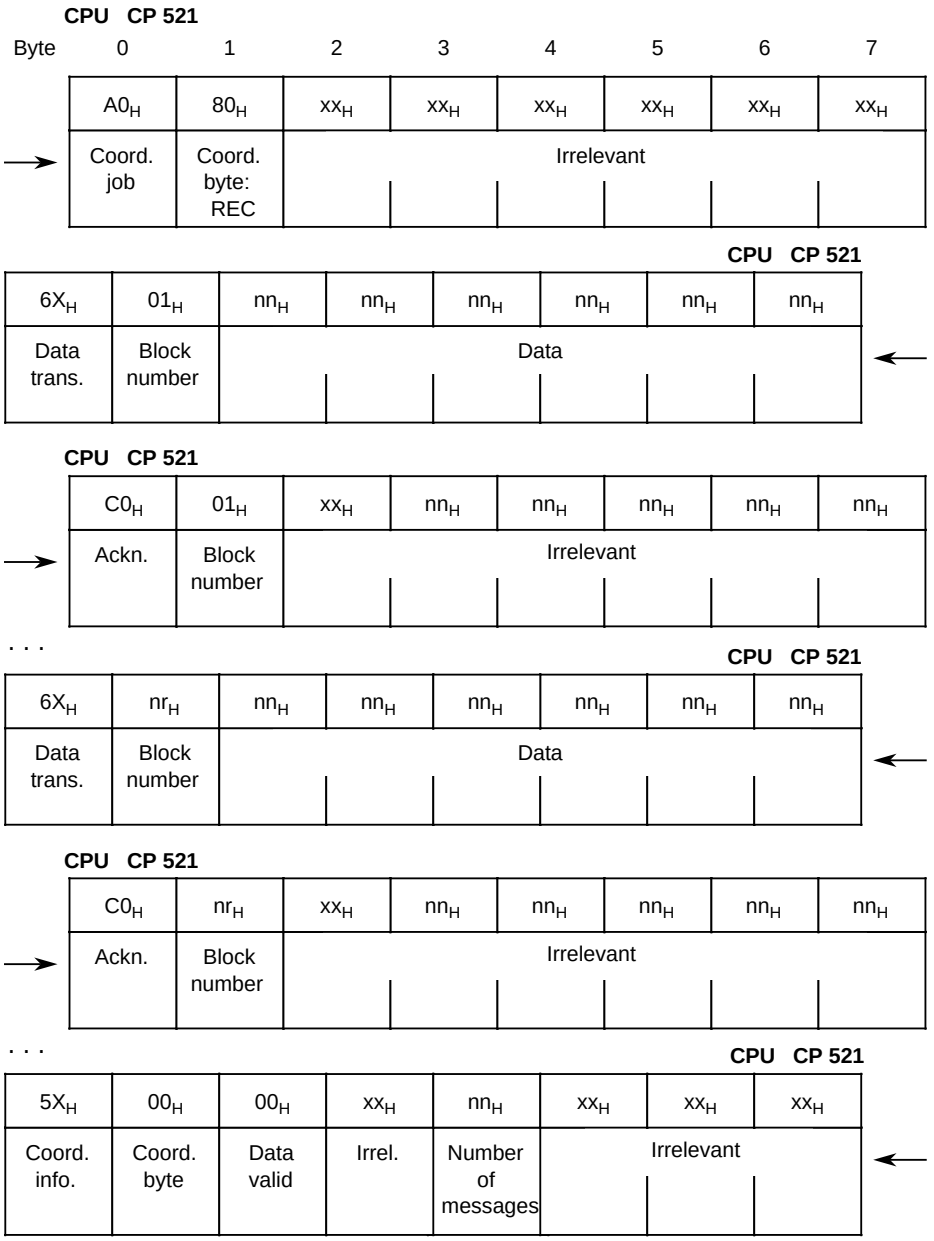


Figure 7-4. Sequence Schematic when Receiving Data

CPU job request: Coordinate data transfer ($A0_H$)

To coordinate data transfer write job $A0_H$ into byte 0 of the PIQ. This defines the meaning of bytes 1 to 7.

Table 7-25. Coordination Job

Byte	Value	Meaning
0	$A0_H$	Job: Coordinate data transfer
1	80_H	Receive message (receive bit)
2	Irrelevant	
3	Irrelevant	
4	Irrelevant	
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Byte 1: You give permission to receive for a message by setting bit 7 = 1. This initiates data transfer. The CP 521 resets this receive bit after a complete receive message has been sent to the CPU.

CP response to job $A080_H$

If the CP 521 receives job A0H (Coordinate data transfer) with receive bit 80_H (Receive message) set, and if there are no errors, the CP 521 acknowledges the CPU job request with $6X_H$ in byte 0:

Table 7-26. Data transfer: Receive the 1st Data Block

Byte	Value	Meaning
0	$6X_H$	Data transfer from the CP 521 to the CPU
1	01_H	Number of the 1st data block
2		Data
3		
4		
5		
6		
7		

Byte 1: Indicates the number of the data block which the CP 521 is transferring to the CPU. The CP 521 increments the block number by 1 continuously starting with block 01_H, for as long as it is transferring a data block to the CPU.

Byte 2 to 7: Data which is transferred from the CP 521 to the PII.

CPU acknowledgement ($C0_H$)

The CPU acknowledges input of the data block with "$C0_H$" in byte 0:

Table 7-27. Acknowledging the 1st Data Block

Byte	Value	Meaning
0	$C0_H$	Acknowledges entry of the 1st data block
1	01_H	Number of the data block
2	Irrelevant	
3	Irrelevant	
4	Irrelevant	
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Byte 1: Number of the data block which the CPU has received (here 01_H). The acknowledged block number must agree with the number of the block sent, otherwise data transfer will be aborted.

Note:

The CP 521 continues to send a data block to the CPU until it receives acknoledgement. For this reason, the CPU receives the data of the acknowledged block from the CP 521 once more within the same data cycle in which it was acknowledged. The CPU program must take account of this or the CP 521 will receive an acknowledgement with the wrong block number and data transfer will be aborted.

Data transfer continues in the manner described till the CPU has acknowledged the last data block of a message ( ). The CP 521 terminates data transfer with a final reponse ( ):

CP response to the CPU acknowledgement $C0_H$

Table 7-28. Data Transfer: Receive the 25th (Last) Data Block

Byte	Value	Meaning
0	$6X_H$	Data transfer from the CP 521 to the CPU
1	19_H	Number of the last data block: here 25D
2		
3		
4		D a t a
5		
6		
7		

CPU acknowledgement for the 25th data block

Tabelle 7-29. Acknowledgement for the 25th Data Block

Byte	Value	Meaning
0	$C0_H$	Acknowledges entry of the last data block
1	19_H	Number of the 25th data block
2	Irrelevant	
3	Irrelevant	
4	Irrelevant	
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Final response from the CP 521

Table 7-30. Final Coordination Information

Byte	Value	Meaning
0	$5X_H$	Coordination information
1	00_H	Coordination bit reset
2	00_H	Data valid
3	Irrelevant	
4	00_H - 64_H	Number of messages in receive mailbox
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Byte 1: The coordination bit is reset

Byte 2: Data valid

Byte 4: Number of messages in the receive mailbox of the CP 521

Example: Receiving a fixed-length message
Let's assume you have plugged into the module in slot 7 (initial address 120) and you want to receive twelve-byte-long data messages with the CP 521. Write the incoming data into data block (DB) 40.

STL FB 3		Explanation
NAME	:EMPFANG	Receive fixed length (12 bytes)
	:C DB 40	Open target DB
	:L IW 120	
	:L KH 0F00	
	:AW	
	:L KH 0800	
	:!=F	Job not processed by CP
	:BEC	
	:U F 90.0	If flag 90.0 (transfer coord. job) is set,
	:JC =M001	jump to label 001
	:L KH A080	The coordination job is transferred to
	:T QW 120	the CP in byte 0 and the send permission
	:	in byte 1.
	:S F 90.0	Coordination job initiated.
	:BEU	Block completed.
M001	:AN F 90.1	Wait cycle after Stop Run or POWER UP
	:S F 90.1	till CP status reported.
	:BEC	
	:L KH 0000	If no new job has been initiated after
	:T QW 120	the FB has been run, 0000_H (data not
	:	relevant) is transferred to the CP
	:L EW 120	Read in of bytes 0 and
	:L KH F00F	
	:AW	If no data has been sent from the CP to
	:L KH 6001	the CPU (byte 0 = 50_H), jump to label
	:><F	002.
	:JC =M002	
	:L IW 122	Read in of bytes 2 to 7
	:T DW 0	Read in the 6 data bytes transferred
	:L IW 124	from the CP to the CPU in block one
	:T DW 1	(byte 1 = 01_H)
	:L IW 126	
	:T DW 2	

STL FB 3 (Continued)	Explanation (Continued)
:L KH C001 :T QW 120 :BEU :	and transfer the acknowledgement for block No. 1 to the CP. The block is completed
M002 :L IW 120 :L KH 6002 :><F : :JC =M003 :	Read in of bytes 0 and 1 If no data has been sent to the CPU from the CP (byte 0 = 50_H), jump to label 003.
:L IW 122 :T DW 3 :L IW 124 :T DW 4 :L IW 126 :T DW 5	Read in of bytes 2 to 7 Read in the six data bytes transferred to the CPU by the CP in block two (byte 1 = 02_H)
:L KH C002 :T QW 120 : : :	and transfer the acknowledgement for block No. 2 to the CP. The next job can be initiated.
:BEU M003 :	The block is completed.
:L IW 120 :L KH F00F :AW :L KH 5000 :!=F : :R F 90.0 : :BE	If the CP acknowledges a coordination job or the complete transfer of a data message without errors, flag 90.0 is reset. The next job can be initiated.

Receiving variable-length messages

Variable-length messages are received in the same way as fixed-length messages.
If you want to receive variable-length data messages from a peripheral device using the CP 521, you must define the message end code for both devices with the job 9072_H in word 6.

Example: Receiving variable-length messages (only for CPU 103)
Let's assume you have plugged into the module in slot 7 (initial address 120) and you want to receive data messages from a peripheral device using the CP 521. Write the data you receive into data block (DB) 40.

STL FB 4	Explanation
NAME :EMPFANG	Receiving with variable length
:C DB 40	Open target DB
:	
:L KH 0000	Transfer "Data not relevant" to the
:T QW 120	CP 521
:	
:A F 90.0	Initiate from the CP 521 to the CPU data
:JC =M001	transfer
:L KH A080	Request data from CP 521 (coordination
:T QW 120	information)
:S F 90.0	Coordination information sent
:	
:	Auxiliary register preset: data transfer
:L KH 6001	from CP 521 to CPU with
:T FW 22	block number
:L KH C001	Acknowledgement to CP 521 data
:T FW 24	block transferred
:L KF +0	Preset pointer in target DB with initial
:T FW 30	value
:L KF +1	
:T FW 32	
:L KF +2	
:T FW 34	

STL FB 4 (continued)	Explanation (continued)
: :BEU : M001 :L IW 120 :L KH F00F :AW :L FW 22 :><F :JC =M002 :L IW 122 :L KH 0D0D :!=F :JC =LOOP :TAK :B FW 30* :T DW 0 :L IW 124 :L KH 0D0D :!=F :JC =LOOP :TAK :B FW 32* :T DW 0 :L IW 126 :L KH 0D0D :!=F :JC =LOOP :TAK :B FW 34* :T DW 0 LOOP :L FW 24 :T AW 120	Block end Has data been transferred from the CP 521 to the CPU? ” N O ” Store data received in the target DB. The end codes read in with the data (here: $0D0D_H$) are not transferred to the target DB. An even number of data bytes must be received so that the end-of-text characters are in the same data word. Transfer data block acknowledgement to the CP 521.

* The FB only runs with the CPU 103;
The operation: "Process flag word" (DO FW) is not included in the operation set of the CPU 100 and the CPU 102.

STL FB 4 (Continued)	Explanation (Continued)
: :L FW 22 :L KF +1 :+F :T FW 22 :L FW 24 :L KF +1 :+F :T FW 24 :L FW 30 :L KF +3 :+F :T FW 30 :L FW 32 :L KF +3 :+F :T FW 32 :L FW 34 :L KF +3 :+F :T FW 34	Increment the block number and the pointer in the target DB by 1 at "6XNR" (FW 22) and "C0NR" (FW 24).
:S F 90.7 :BEU	Data is being received Block end
M002 : :L IW 120 :L KH F00F :AW :L KH 5000 :><F :JC =ENDE	CP acknowledgement after transfer of the complete message error-free? ” N O ”

STL FB 4 (Continued)	Explanation (Continued)
: : : : :R F 90.0 :R F 90.7 :L KH 6001 :T FW 22 :L KH C001 :T FW 24 :L KF +0 :T FW 30 :L KF +1 :T FW 32 :L KF +2 :T FW 34 : ENDE : :BE	Prepare next message after positive acknowledgement Reset status flag Preset acknowledgements and target DB pointer with their initial values.

7.5 CPU Job Requests and CP Error Messages in ASCII Mode

All the CPU job requests permissible in ASCII mode are listed in this section. There is also a complete list of error messages with which the CP 521 acknowledges CPU job requests.

7.5.1 Permissible CPU Job Requests in ASCII Mode

Table 7-31. Permissible Jobs in ASCII Mode

Byte 0	Byte 1	Jobs
00_H		
10_H		Set clock
Adapt interface:		
90_H	00_H	Transfer parameters : Parameter block 0
90_H	20_H	" " : " 2
90_H	$7X_H$	Transfer parameters : " 7
90_H	70_H	Setting : Print mode x=0
90_H	71_H	Setting : Transparent ASCII mode x=1
90_H	72_H	Setting : Interpretive ASCII mode x=2
Coordinate data transfer:		
$A0_H$		
$A0_H$	01_H	Coordinate "Send" data transfer
$A0_H$	80_H	Coordinate "Receive" data transfer
Execute data transfer:		
$B0_H$	Block No.	Coordinate "CPU CP" data transfer
$C0_H$	Block No.	CPU acknowledgement: Received data transferred

7.5.2 Illegal Jobs with Error Messages

If data has not yet been transferred and you write a job in word 0 of the PIQ (byte 0 and byte 1) other than the job executed above, the CP 521 stores an error message in the status byte (PII).
The following are illegal jobs in ASCII mode which are acknowledged with the error messsge $4X_H$:

- All jobs concerned exclusively with print mode (20_H,30_H, 40_H, 50_H, 60_H, 70_H, 80_H)
- All job numbers outside the permissible value range for CPU job requests ($D0_H$ to FF_H)
- Job number $90XX_H$ if the parameter blocks are initialized with data from outside the permissible value range
- Job number $B0XX_H$ if data transfer has not previously been initiated.

CPU job requests transferred to the CP during a data transfer must belong to the data traffic. All jobs other than 00_H and $B0_H$ + block number cause data transfer to be aborted. The CP 521 acknowledges this as follows:

Table 7-32. Message: Invalid Job (PII)

Byte	Value	Meaning
0	$5X_H$	Coordination information
1	00_H	Coordination bit "Send"/"Receive" reset
2	FF_H	Invalid job / data illegal / data transfer aborted
4	nn_H	Number of messages in receive buffer
3, 5 to 7	Irrelevant	

Message $4X_H$ "Illegal job" is transmitted if the CPU tries to continue data transfer after it has been aborted ($B0XX_H$).

The CP 521 aborts a receive job immediately ($A080_H$) if
- there is no message present in the CP (CP acknowledgement 5000_H) or
- there is a receive message error in the CP.

Exception: Receive messages with error 09_H (character delay time exceeded) are sent to the CPU. All characters received correctly in the CP before message 09_H is sent are sent to the CPU.

Table 7-33. Message: No Receive Message Present (PII)

Byte	Value	Meaning
0	$5X_H$	Coordination information
1	00_H	Receive" coordination bit reset
2	00_H	Valid job
4	00_H	Number of messages in receive buffer
3, 5 to 7	Irrelevant	

Table 7-34. Message: Error in Receive Message (PII)

Byte	Value	Meaning
0	$5X_H$	Coordination information "X": Cause of error/exception: character delay time exceeded
1	00_H	”Receive" coordination bit reset
2	00_H	Valid job
4	nr_H	Number of receive messages
3, 5 to 7	Irrelevant	

7.5.3 Further Error Messages

he CP 521 signals further errors stored in byte 0 (status byte) of the module, which you can read in and evaluate via the PII. Byte 0, which documents the status of the module, is divided into two half bytes whose values can be combined with each other in any way (e.g. 28_H: default time set / CP busy). This makes detailed error evaluation possible.

Table 7-35. Status Byte in ASCII Mode (PII)

Byte 0 Bit 4 to 7	Byte 0 Bit 0 to 3	Status
0	0	No error
X	7	No backup battery in CP;
X	8	CPU job request not accepted
X	9*	Character delay time exceeded
X	A*	Parity error
X	B*	Receive after XOFF or receive after DTR = 0
X	C*	Message length greater than 256 bytes
X	D	Permanent wire break on peripheral device line
X	E*	Receive mailbox overflow

X= Signal status for the other half byte irrelevant

***** These error messages do not appear until after the coordination job "Receive data" for this message.

Table 7-35. Status Byte in ASCII Mode (PII) (Continued)

Byte 0 Bit 4 to 7	Byte 0 Bit 0 to 3	Status
1	X	Clock defective
2	X	Default time set
3	X	Time/date error
4	X	Illegal job
5	X	Coordination information (only after job "Coordinate data transfer")
6	X	Data transfer from the CP 521 to the CPU
8	X	Hardware error

X= Signal status for other byte half irrelevant

Backup battery missing ($X7_H$)
- No battery has been inserted in the CP 521, or
- the battery is defective.

Job not accepted ($X8_H$)
This message is sent to the CPU on two occasions:
- A CPU job request cannot be accepted because the previous job has not yet been processed by the CP operating system. This response can be scanned two program scans after allocation of the CPU job request.
- In the case of message buffer overflow.
  The CP operating system writes the CPU job requests identified on page 6-41 into the message buffer for further evaluation. This buffer can only be filled by the CPU job requests "03_H" and "04_H". CPU job requests "02_H", "05_H" and "06_H" are also stored in the message buffer but they cannot cause a buffer overflow although this can be caused after the CP has received 255 CPU "Print message texts" jobs. The CP message can only be scanned 3 program scan cycles after allocation of the job, as with all other responses.

Character delay time exceeded ($X9_H$)
The time between two received characters is greater than the value set in parameter block 7.
The data received up to the error is transferred to the CPU as a message.

Parity error (XA_H)
The parity of received characters does not agree with the parity configured in parameter block 0.
The message is not transferrred to the CPU and is not stored in the receive mailbox.

Receive after XOFF or receive after DTR = 0 (XB_H)
In the case of XON/XOFF protocol:
The CP 521 sends XOFF to the peripheral device if
- less than 20 bytes are free in the receive mailbox
- the CP has received more than 100 messages.

The CP 521 sends XON again only when more than 256 bytes are free in the receive mailbox.

When using MODEM control signals:
The CP 521 sends DTR = 0 to the peripheral device if
- less than 20 bytes are free in the receive mailbox
- the CP has received more than 100 messages.

The CP 521 sends DTR = 1 again only when more than 256 bytes are free in the reveive mailbox.

Error message XB_H is output if the peripheral device has sent more characters than the CP can accomodate in the receive mailbox. The message is not transferred to the CPU and not stored in the receive mailbox.

Message lengths greater than 256 bytes (XC_H)
The peripheral device has sent a message which is longer than 256 bytes.
The CP outputs error message XC_H if it has not detected an end-of-text character after receiving 256 characters.
The message is not tansferred to the CPU and not stored in the receive mailbox.

Permanent wire break on the peripheral device line (XD_H)
The peripheral device must
- hold the RXD& line constantly at logic "1"
- hold the CTS& line constantly at "ON" (U 3V) when using MODEM control signals

while sending.
Otherwise an error message is output.

Receive mailbox overflow (XE_H)
The following can be stored in the receive mailbox:
- Up to 1024 bytes of data
- Up to 100 messages.

If these values are exceeded when receiving a message from the peripheral device, the CP outputs an error message.
The message is not transferred to the CPU and not stored in the receive mailbox.

Coordination information ($5X_H$)
The CP outputs this message in response to the "Coordinate data transfer" job.
Bytes 1 to 7 contain moere detailed information on data transfer (7.4.1).

Default time set ($2X_H$)
The clock is set with the values Sunday 01.01.89 112.00.00.

Time/date error ($3X_H$)
At least one manipulated variable is outside the permissible range.
The clock has not accepted the new clock data and continues to operate with the old data.

Illegal job ($4X_H$)
A job which is illegal in ASCII mode has been written into word 0 of the transfer memory. You will find an overview of permissible jobs in ASCII in 7.5.

8 ASCII Code Table

8 ASCII Code Table

Hex	**ASCII**	**Hex**	**ASCII**	**Hex**	**ASCII**	Hex	ASCII
00	NUL	10	DLE	20	SP	30	0
01	SOH	11	DC1 (X-ON)	21	!	31	1
02	STX	12	DC2 (TAPE)	22	”	32	2
03	ETX	13	DC3 (X-OFF)	23	#	3	3
04	EOT	14	DC4 (TAPE)	24	$	34	4
05	ENQ	15	NAK	25	%	35	5
06	ACK	16	SYN	26	&	36	6
07	BEL	17	ETB	27	'	37	7
08	BS	18	CAN	28	(	38	8
09	HT (TAB)	19	EM	29	)	39	9
0A	LF	1A	SUB	2A	*	3A	:
0B	VT	1B	ESC	2B	+	3B	;
0C	FF	1C	FS	2C	,	3C	<
0D	CR	1D	GS	2D	-	3D	=
0E	SO	1E	RS	2E	.	3E	>
0F	SI	1F	US	2F	/	3F	?

ASCII Code Table (Continued)

Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
40	@	50	P	60		70	p
41	A	51	Q	61	a	71	q
42	B	52	R	62	b	72	r
43	C	53	S	63	c	73	s
44	D	54	T	64	d	74	t
45	E	55	U	65	e	75	u
46	F	56	V	66	f	76	v
47	G	57	W	67	g	77	w
48	H	58	X	68	h	78	x
49	I	59	Y	69	i	79	y
4A	J	5A	Z	6A	j	7A	z
4B	K	5B	[	6B	k	7B	{
4C	L	5C	\	6C	l	7C	I
4D	M	5D	]	6D	m	7D	
4E	N	5E		6E	n	7E	} (ALT MODE)
4F	O	5F	-	6F	o	7F	DEL (RUB OUT)

Appendix : Summary

A.1	CPU Jobs (PIQ)	A. - 1
A.2	CP Responses (PII).	A - 3
A.3	Parameter Setting Data for Print Mode (User Submodule)	A - 6
A.4	Parameter Setting Data for ASCII Mode (User Program)	A. - 9
A.5	Slot Adressing	A. - 12
A.6	Status Message and Error Messages	A - 12
A.7	Typical Printouts of the ”KFa, b” Data Format.	A - 13

Tables

A-1. PIQ for Job ”Transfer Parameter Setting Data for Parameter Block 0” . . . . . A - 9

A-2. PIQ for Job ”Transfer Parameter Setting Data for Parameter Block 2” . . . . . A - 10

A-3. PIQ for Job ”Transfer Parameter Setting Data for Parameter Block 7” (Print Mode) . . . . . A - 10

A-4. PIQ for Job ”Transfer Parameter Setting Data for Parameter Block 7” (Transparent ASCII Mode) . . . . . A - 11

A-5. PIQ for Job ”Transfer Parameter Setting Data for Parameter Block 7” (Interpretive ASCII Mode) . . . . . A - 11

A.1 CPU Jobs (PIQ)

Byte	Data not relevant	Set clock	Set page number	Output text	Output page feed/line feed	Delete message buffer	Print all texts
0	00	10	20	30 / 40	50 / 60	70	80
1		Weekday 01-07$_H$	Page number settings 01-FF$_H$	Message No. 01-FF$_H$			
2		BCD format Day 01-31		High Byte Variable			
3		Month 01-12		Low Byte			
4		Year 00-99		High Byte			
5		Hour 01-12 a.m. 00-23 81-92 p.m.		Variable *2* Low Byte			
6		Minute 00-59		High Byte Variable 3			
7		Second 00-59		Low Byte			

CPU Jobs (PIQ) (Continued)

Byte	Set interface para-meters	Set interface para-meters	Set interface para-meters	Set interface para-meters	Set interface para-meters	Coordinate data transfer	Execute data transfer	CPU acknow-ledgement
0	90	90	90	90	90	A0	B0	C0
1	Block 0 00	Block 2 20	Block 7 70 Print mode	Block 7 71 ASCII-transpar.	Block 7 72 ASCII-interpret.	01_H / 80_H Send/receive job	Data block No. 01 - $2B_H$	Data block No. 01 - $2B_H$
2	Baudrate 1-8	XON		Delay time		Mess. length	Data	
3	Parity 0-4	XOFF		0000_H to $0BB8_H$		0000_H to 0100_H	Data	
4	Busy 0 / 1			Mess. length		End-of-text character	Data	
5	TTY / RS-232-C (V.24) 0 / 1			000_H to 0100_H		End-of-text character	Data	
6	Data format 7 D Bit: 0 8 D Bit: 1				End-of-text character		Data	
7	Hand-shake Off / On 0 / 1				End-of-text character		Data	

A.2 CP Responses (PII)

PII (byte 0)	Half Byte				Half Byte				
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
No error	0	0	0	0	0	0	0	0	No error
Clock defective	0	0	0	1	0	0	0	1	Memory submodule not plugged in or empty
Default time set	0	0	1	0	0	0	1	0	No message text configured or memory submodule empty
Time/date error	0	0	1	1	x	x	x	x	Not used
Job error/ illegal job	0	1	0	0	x	x	x	x	Not used
Coordination information	0	1	0	1	x	x	x	x	Not used
Data transfer CP CPU	0	1	1	0	0	1	1	0	Message text No. not configured
Not used	x	x	x	x	0	1	1	1	CP backup battery missing
Hardware error	1	0	0	0	1	0	0	0	CPU job not accepted/ CP clock data invalid
					1	0	0	1	Delay time exceeded
					1	0	1	0	Parity error
					1	0	1	1	Receive after XOFF or after DTR = 0
					1	1	0	0	Message length > 265 bytes
					1	1	0	1	Permanent breaks on peripheral line
					1	1	1	0	Receive mailbox overflow

CP Responses (PII) (Continued)

PII (byte 1)	Half Byte				Half Byte				
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Peripheral device ready	0	0	0	0	X	X	X	X	
	0	0	0	1	X	X	X	X	
Peripheral device not ready	X	X	X	X	0	0	0	1	Sunday
	X	X	X	X	0	0	1	0	Monday
	X	X	X	X	0	0	1	1	Tuesday
	X	X	X	X	0	1	0	0	Wednesday
	X	X	X	X	0	1	0	1	Thursday
	X	X	X	X	0	1	1	0	Friday
	X	X	X	X	0	1	1	1	Saturday
Permission to send*	0	0	0	0	0	0	0	1	
Permission to receive**	1	0	0	0	0	0	0	0	
Block No.	0	0	0	0	0	0	0	0	
	to								
	1	1	1	1	1	1	1	1	
PII (byte 2)									
	$01 - 31_{BCD}$								Day
Coordination data valid*	0	0	0	0	0	0	0	0	
Coordination data invalid**	1	1	1	1	1	1	1	1	
	Data**								

* after byte 0=5x
** after byte 0=6x

CP Responses (PII) (Continued)

PII (byte 3)	Half Byte				Half Byte				
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
	01 - 12_{BCD}								Month
	Irrelevant* or block no. for sending								
	Data**								
PII (byte 4)									
	00 - 99_{BCD}								Year
Number of messages in receive buffer*	00 - 64_{H}								
	Data**								
PII (byte 5)									
	00 - 23_{BCD} 24h 01 - 12_{BCD} 12h a.m. 81 - 92_{BCD} 12h p.m.								Hour
	Irrelevant*								
	Data**								
PII (byte 6)									
	00 - 59_{BCD}								Minute
	Irrelevant*								
	Data**								
PII (byte 7)									
	00 - 59_{BCD}								Second
	Irrelevant*								
	Data**								

***** after byte 0=5x ****** after byte 0=6x

A.3 Parameter Setting Data for Print Mode (User Submodule)

Block	Meaning		Value Range	Default Values on the CP 521
0	Baud rate	110 Bd	1	
		200 Bd	2	
		300 Bd	3	
		600 Bd	4	4
		1200 Bd	5	
		2400 Bd	6	
		4800 Bd	7	
		9600 Bd	8	
	Parity	even	0	0
		odd	1	
	(Parity bit always ”1”)	”mark”	2	
	(Parity bit always ”0”)	”space”	3	
		None	4	
	BUSY signal	No	0	0
		Yes	1	
	Interface	TTY	0	0
		RS-232-C (V.24)	1	
	Data format	7 Data bits	0	0
		8 Data bits	1	
	HW handshake	OFF	0	0
		ON	1	
1	Waiting time after	CR	(00_H to FF_H) · 25ms	$0A_H$: 0,25 s
		LF	(00_H to FF_H) · 25ms	$0A_H$: 0,25 s
		FF	(00_H to FF_H) · 25ms	$A0_H$: 4 s
2	XON character		ASCII character (11_H, 20_H to $7F_H$)	FF_H (No XON / XOFF listing)
	XOFF character		ASCII character (13_H,20_H to $7F_H$)	

Block	Meaning	Value Range	Default Values on the CP 521
3	Function character	ASCII character (20_H to $7F_H$)	' (27_H)
	End-of-text character	ASCII character (20_H to $7F_H$)	$ (24_H)
4	**Date and time of day:**		
	Order for date J,Y = Year M = Month T,D = Day	JMT, JTM,MTJ, MJT, TJM, TMJ, YMD, YDM, MDY, MYD, DYM, DMY	TMJ
	Separator for date	ASCII character (20_H to $7F_H$)	"." (2E)
	Order for time of day H = Hours M = Minutes S = Seconds	HMS, HSM, MSH, MHS, SHM, SMH	HMS
	Separator for time of day	ASCII character (20_H to $7F_H$)	"." (2E)
	24h clock (Ger.) 12h clock (US/UK)	d, D e, E	D (24h clock)
5	**Page format**		
	Lines per page Left margin Page number Top, Bottom,	20_H to FF_H 00_H to $3C_H$ O, o U, u No other characters	48_H (72_D) 00_H B bottom
6	**Header and footer**		
	Header 1 Header 2 Footer 1 Footer 2	K1"Text" 80 characters K2"Text" 80 characters F1"Text" 80 characters F2"Text" 80 characters	No headers or footers

Block	Meaning	Value Range	Default Values on the 521
7	Operating mode	0_H: Print mode 1_H: Transparent ASCII mode 2_H: ASCII mode	0 (with user sub-module) 1 (without user submodule)
	Character delay time* (four decimal digits)	(0001 to 3000) ·100ms	0001: 100ms
	Message length* (three decimal digits)	000 to 256 bytes If length = 0, at least one end code must be entered	064 bytes
	End-of-message identifier**	1 End-of-text character: 01_H to $7F_H$ 2 End-of-text characters: 01_H to $7F_H$	$0D_H$ (carriage return)

***** Only significant in ASCII mode

****** Only significant in interpretive ASCII mode

A.4 Parameter Setting Data for ASCII Mode (User Program)

Table A-1. PIQ for Job "Transfer Parameter Setting Data for Parameter Block 0"

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number	00_H	-
2	Baud rate 110 Bd 200 Bd 300 Bd 600 Bd 1200 Bd 2400 Bd 4800 Bd 9600 Bd	 01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	04_H
3	Parity even odd ”mark” (Parity bit always ”1”) ”space” (Parity bit always ”0”) None	 00_H 01_H 02_H 03_H 04_H	00_H
4	BUSY signal No Yes	 00_H 01_H	00_H
5	Interface TTY RS-232-C (V.24)	 00_H 01_H	00_H
6	Data format 7 data bits 8 data bits	 00_H 01_H	00_H
7	HW handshake OFF ON	 00_H 01_H	00_H

Table A-2. PIQ for Job "Transfer Parameter Setting Data for Parameter Block 2"

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number	20_H	-
2	XON/XOFF protocol XON character no protocol	 00_H to $7F_H$ FF_H	 $FFFF_H$
3	XON/XOFF protocol XOFF character no protocol	 00_H to $7F_H$ FF_H	 (No XON/XOFF protocol)
4 to 7	No significance	00_H to FF_H	-

Table A-3. PIQ for Job "Transfer Parameter Setting Data for Parameter Block 7" (Print Mode)

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number and Identifier for "Print mode"	70_H	0: If memory submodule with message texts plugged in 1: Without memory submodule
2 to 7	No significance	00_H to FF_H	

Table A-4. PIQ for Job "Transfer Parameter Setting Data for Parameter Block 7" (Transparent ASCII Mode)

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number and Identifier for "Transparent ASCII mode"	71_H	0: If memory submodule with message texts plugged in 1: Without memory submodule
2+3	Character delay time	0001_H to $0BB8_H$	0001_H : 100 ms
4+5	Message length (in bytes)	0000_H to 0100_H	40_H : 64_D
6+7	No significance	00_H to FF_H	

Table A-5. PIQ for Job "Transfer Parameter Setting Data for Parameter Block 7" (Interpretive ASCII Mode)

Byte	Meaning	Value Range	Default Values on the CP 521
0	Job number	90_H	-
1	Parameter block number and Identifier for "Interpretive ASCII mode"	72_H	0: f memory submodule with message texts plugged in 1: Without memory submodule
2+3	Character delay time	0001_H to $0BB8_H$	0001_H : 100 ms
4+5	Message length (in bytes)	0000_H to 0100_H	40_H : 64_D
6	End-of-text character 1 ASCII character No end-of-text character	01_H to $7F_H$ 00_H	00_H
7	End-of-text character 2 ASCII character No end-of-text character	01_H to $7F_H$ 00_H	$0D_H$*

***** $0D_H$ = carriage return

A.5 Slot Addressing

Slot	0	1	2	3	4	5	6	7
Addresses PII/PIQ	64 ... 71	72 ... 79	80 ... 87	88 ... 95	96 ... 103	104 ... 111	112 ... 119	120 ... 127

A.6 Status Messages and Error Messages

Byte 0 Bit 4 to 7	Byte 0 Bit 0 to 3	Status
0	0	No error
X	7	CP backup battery missing
X	8	CP active/job not processed
X	9	Character time out
X	A	Parity error
X	B	Receive after XOFF or receive after DTR = 0
X	C	Message length greater than 256 bytes
X	D	Permanent break on I/O line
X	E	Receive mailbox overflow
1	X	Clock defective
2	X	Default time set
3	X	Time/date error
4	X	Illegal job
5	X	Coordination information (only after job "Coordinate data transfer")
6	X	Data transfer CP 521 CPU
8	X	Hardware error

X= Signal state for the other half byte not relevant

A.7 Typical Printouts of the ”KFa, b” Data Format

Configur-ation	Printout of the Variables KF					
	+7	-7	256	-256	32767	-32767
KF 5,0	7	-7	256	-256	32767	-32767
KF 4,0	1	-1	26	-26	3277	-3277
KF 3,0	0	-0	3	-3	328	-328
KF 2,0	0	-0	0	-0	33	-33
KF 1,0	0	-0	0	-0	3	-3
KF 0,0	7	-7	256	-256	32767	-32767
KF 4,1	0,7	-0,7	25,6	-25,6	3276,7	-3276,7
KF 3,1	0,1	-0,1	2,6	-2,6	327,7	-327,7
KF 2,1	0,0	-0,0	0,3	-0,3	32,8	-32,8
KF 1,1	0,0	-0,0	0,0	-0,0	3,3	-3,3
KF 0,1	0,0	-0,0	0,0	-0,0	0,3	-0,3
KF 3,2	0,07	-0,07	2,56	-2,56	327,67	-327,67
KF 2,2	0,01	-0,01	0,26	-0,26	32,77	-32,77
KF 1,2	0,00	-0,00	0,03	-0,03	3,28	-3,28
KF 0,2	0,00	-0,00	0,00	-0,00	0,33	-0,33
KF 2,3	0,007	-0,007	0,256	-0,256	32,767	-32,767
KF 1,3	0,001	-0,001	0,026	-0,026	3,277	-3,277
KF 0,3	0,000	-0,000	0,003	-0,003	0,328	-0,328
KF 1,4	0,0007	-0,0007	0,0256	-0,0256	3,2767	-0,2767
KF 0,4	0,0001	-0,0001	0,0026	-0,0026	0,3277	-0,3277
KF 0,5	0,00007	-0,00007	0,00256	-0,00256	0,32767	-0,32767

Every fixed-point number is represented by at least (a+1) characters. Initial zeros are not repressed but replaced by spaces.

Appendix B: Application Example

B.1 Task . . . . . B. - 1

B.2 Device Configuration . . . . . B - 1

B.3 Terminal Assignments for the RS-232-C (V.24) Interface . . . . . B - 2

B.4 Setting the Parameters on the Serial Interface of the CP 521 . . . . . B. - 3

B.5 Programming the Programmable Controller . . . . . B - 4

B.6 Programming Blocks (DBs, FBs, OBs) . . . . . B - 5

B.7 Testing an Example Program . . . . . B - 11

Figures

B-1. Configuration Schematic: Point-to-Point Connection of Two S5-100Us . . . . . B. - 1
B-2. CP 521- CP 521 Terminal Assignments with Handshake . . . . . B .- 2

Tables

B-1. Setting the CP 521 Parameters in Interpretive ASCII Mode . . . . . B - 3

B Application Example

B.1 Task

Two S5-100U programmable controllers are to exchange six bytes of data. For this purpose, a point-to-point connection is established between both PCs (ASCII mode). Both PCs contain a program for sending and receiving data messages.

B.2 Device Configuration

The device configuration is shown in Figure B-1. Both CP 521s are installed in slot 0 and linked via an RS-232-C (V.24) cable (incl. handshake).

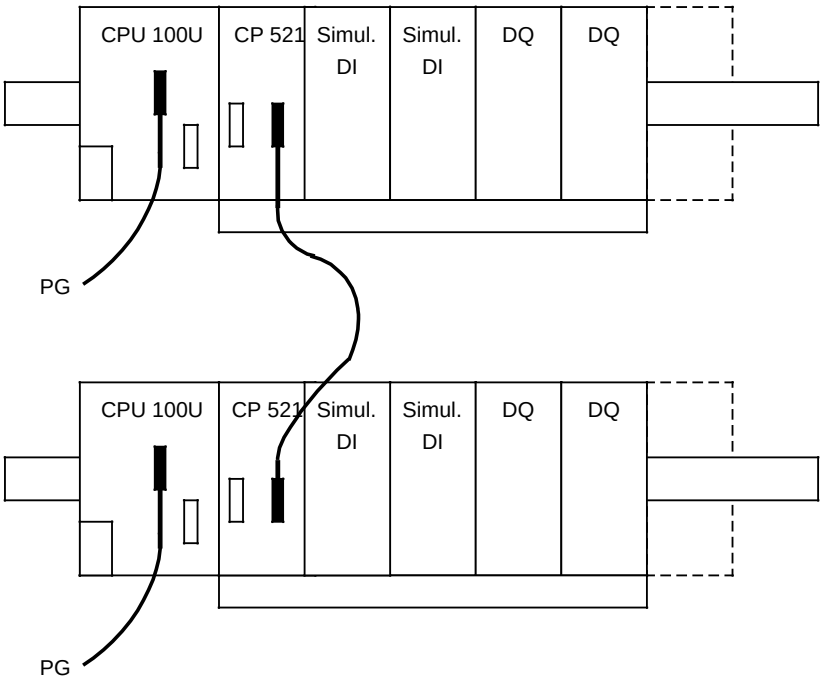


Figure B-1. Configuration Schematic: Point-to-Point Connection of Two S5-100Us

B.3 Terminal Assignments for the RS-232-C (V.24) Interface

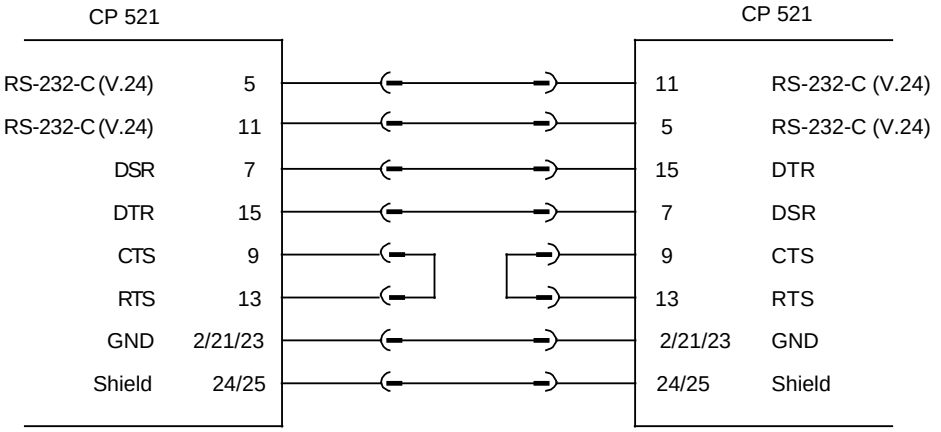


Figure B-2. CP 521 - CP 521 Terminal Assignments with Handshake

B.4 Setting the Parameters on the Serial Interface of the CP 521

To establish a point-to-point connection, you must set the ASCII mode parameters for the serial interface of the CP 521. Store all parameter setting data in DB 1 on a user submodule (EPROM/EEPROM):

- Enter all parameter setting data in DB 1 using the programmer
- Transfer DB 1 to two user submodules one after the other
- Plug the user submodules into both CP 521s.

Enter the following data for the example program in DB 1 of the user submodule:

Table B-1. Setting the CP 521 Parameters in Interpretive ASCII Mode

Parameter Block Number	Explanation
`: KS = ':0800111';`	:0 : Parameter block 0 8 : 9600 baud 0 : Even Parity 0 : No BUSY signal 1 : RS-232-C (V.24) interface 1 :8 Data bits 1 : Handshake ON
`: KS = ':710050006;`	:7 : Parameter block 7 1 :Transparent ASCII mode 0050 : Character delay time 50 x 100 ms=5s 006 : Message length 6 bytes

B.5 Programming the Programmable Controller

Before entering the statement list, we would like to explain the basic settings of the example program.
The program is designed so that both programmable controllers are permanently ready to receive. Transmission on a 200 ms timebase (5 Hz). The CPU expects to receive messages constantly and switches to "Send" every 200 ms for a send message. You can enter and change the send data via simulator modules.

Both programmable controllers store
- Send data in DW 0 to DW 2 of DB 2
  and
- Receive data in DW 0 to DW 2 of DB 3

Proceed as follows when programming:
- Enter the blocks in the programmer in the order specified
- Transfer them to both programmable controllers.

Order for entering the blocks:
- DB2 Send data (DW 0 to DW 2)
- DB3 Receive data (DW 0 to DW 2)

- FB 1 Send program
- FB 2 Receive program
- FB3 Send takt program (5Hz)

- OB1 Test program

B.6 Programming Blocks (DBs, FBs, OBs)

DB 2	Explanation
0: KH = 0000 1: KH = 0000 2: KH = 0000	DW 0 test, send IB 1 and 2 .

DB 3	Explanation
0: KH = 0000 1: KH = 0000 2: KH = 0000	DW 0 test, receive IB 1 and 2 .

STL FB 1				Explanation
NAME	:SEND			Send 6-byte message
	:A	F	101.0	Send active?
	:JC	=M001		
	:			
	:L	KH	A001	Coordinate send
	:T	QW	64	Slot 0
	:			
	:L	KH	0006	Message length 6 bytes
	:T	QW	66	
	:			
	:S	F	101.0	Send active!
	:BEU			
	:			
M001	:L	KH	F00F	Remove error responses
	:L	IW	64	
	:AW			
	:L	KH	5001	Check CP acknowledge
	:><F			
	:			
	:JC	=M002		
	:			
	:L	EW	66	Job valid?

STL FB 1 (Continued)				Explanation (Continued)
	:L	KH	0000	
	:><F			
	:			
	:JC	=M003		
	:			
	:L	KH	B001	Send block 1
	:T	QW	64	
	:			
	:Q	DB	2	Send 6 bytes from DB 2
	:L	DW	0	
	:T	QW	66	
	:L	DW	1	
	:T	QW	68	
	:L	DW	2	
	:T	QW	70	
	:			
	:BEU			
	:			
M003	:L	IW	66	Block 1 acknowledged by CP?
	:L	KH	0001	
	:><F			
	:			
	:JC	=MOO2		
	:			
	:BEU			
	:			
M002	:L	KH	F00F	Remove error messages
	:L	IW	64	
	:AW			
	:L	KH	5000	
	:!=F			Send job completed?
	:			
	:R	F	101.0	Reset send active
	:			
	:R	F	100.1	Reset start flag send
	:			
	:BE			

STL FB 2				Explanation
NAME	:Empfang			Read 6-byte message
	:L	IW	64	Job processed by CP?
	:L	KH	0F00	
	:AW			
	:L	KH	0800	
	:!=F			
	:BEC			
	:			
	:A	F	102.0	Receive active?
	:JC	=M001		
	:			
	:L	KH	A080	Coordinate receive
	:T	QW	64	
	:			
	:S	F	102.0	Receive active!
	:BEU			
	:			
M001	:AN	F	102.1	Wait cycle after STOP-RUN till CP status
	:S	F	102.1	reported
	:BEC			
	:			
	L	KH	0000	Reset jobs
	:T	QW	64	
	:			
	:L	KH	F00F	Data received from CP?
	:L	IW	64	
	:AW			
	:L	KH	6001	
	:><F			
	:			
	:JC	=M002		
	:			
	:A	DB	3	Store 6 bytes of receive data in DB 3
	:L	IW	66	
	:T	DW	0	
	:L	IW	68	

STL FB 2 (Continued)				Explanation (Continued)
	:T	DW	1	
	:L	IW	70	
	:T	DW	2	
	:			
	:L	KH	C001	Acknowledge from CPU to CP
	:T	QW	64	
	:			
	:BEU			
	:			
M002	:L	KH	F00F	Receive message complete?
	:L	IW	64	
	:AW			
	:L	KH	5000	
	:!=F			
	:			
	:R	F	102.0	Reset receive active
	:			
	:BE			

STL FB 3	Explanation
NAME :CPU <-> CP	Data exchange CPU CP 521
:AN T 1	free-running 5 Hz timebase
:L KT 010.0	100 ms pause
:SI T 2	
:	
:AN T 2	
:L KT 010.0	100 ms pulse
:SI T 1	
:	
:A T 1	
:AN F 100.0	Send edge flag
:S F 100.0	
:S F 100.1	
:	
:AN T 1	
:R F 100.0	
:	
:A F 100.1	Send job in 200 ms time if no receive
:A F 102.0	active
:JC FB 1	
NAME :SEND	
:	
:AN F 101.0	Basic setting receive if no send job active
:JC FB 2	
NAME :Empfang	
:	
:BE	

STL OB 1	Explanation
:Q DB 2 :L IW 1 :T DW 0 :	Test Store input byte 1 and 2 in send buffer
:JU FB 3 :CPU <-> CP :	Data exchange CPU CP 521
:Q DB 3 :L DW 0 :T QW 3 :	Test Display receive data in output byte 3 and 4
:BE	

B.7 Testing an Example Program

You can test the program after entering all blocks:

- Enter all send data in the send PC via the simulator module (input byte 1 and 2).
- This data is transferred to the receive PC
- It appears in the digital output modules (output byte 3 and 4) as an LED pattern.

Explanation:

The send data is
- Written into DW 0 of DB 2 in the send PC
- Transferred to the receive PC as a data message via the CP 521 point-to-point connection
- Written into DW = of DB 3 as receive data
- Transferred to the digital output modules of the receive PC.

Index

Index

A

Addressing 5-3
ASCII characters
- message text 6-20
ASCII mode 1-1, 1-3, 2-2, 2-3, 5-1
- interpretive 2-8, 6-61, 7-1, 7-3, 7-17, 7-18, A-11
- interrupt 4-5
- parameter setting data A-9
- transparent 2-8, 6-15, 6-60, 7-1, 7-3, 7-16, A-11

B

Backup battery 7-59
- missing
Battery 2-12
- backup 2-12
- failure 2-12
Baud rate 6-9
Block number 7-20, 7-28
Block separator 7-14
BUSY signal 2-7, 6-9

C

Cable length 3-4
Character delay time
- exceeding 7-59
Clock
- setting 6-43, 6-50, 6-51
Clock test 4-2
Configuration
- header 6-14
- footer 6-14
Configuration data 6-2, 6-11
- message text printout 6-12
Configuring
- message texts on the memory submodule 6-46
Configuring message texts
- printing out 6-56
Connection to the peripheral device
- plugging in 3-1
- unplugging 3-1
Control
- character 6-20
- command 2-2
- parameters 6-32
Coordination
- data transfer 7-24, 7-35
Coordination information 7-25, 7-60
- final 7-31, 7-47
Coordination jobs 7-19
CP 521
- parameter setting 6-16
- setting 6-4, 7-2, 7-6
- startup 7-3
CP 521 parameters
- setting 6-2
CP response 7-19, 7-20, 7-25, A-3, A-4, A-5

CPU
- acknowledgement 7-44
- job 6-40, A-1, A-2
- job requests 7-19, 7-54

D

Data
- interchange 5-1
- traffic 5-1

Data block 7-20, 7-21, 7-27, 7-28
- receiving 7-43
- sending 7-27

Data format 2-5, 6-10, 6-29, 6-30, 6-31, 6-32
Data message 7-31
Data throughput 1-3, 7-19
Data transfer 2-1, 2-10, 6-34, 7-1, 7-20, 7-21
- coordinate 7-24, 7-35, 7-42

Date
- entering 2-2
- error 7-61
- inserting 6-1, 6-22
- setting 6-50

DB
- changing 6-6

DB 1
- parameter setting 6-18

Default time
- setting 7-61

Default values 6-4, 7-5

E

Entering
- message texts 6-11
- place holders 6-22

EPROM/EEPROM 2-3
Error 4-1, 4-2
- date 7-61
- time 7-61

Error evaluation 7-57
Error message 4-1, 6-46, 7-54, 7-57, A-12
- ASCII mode 7-55
- transfer 6-40, 7-6

Error number 4-1, 4-3

F

Footer
- configured 6-14

Function character 6-11, 6-22
Functional description
- general 5-1

H

Handshake 2-8, 2-9, 6-10
Hardware clock
- integrated 2-12

Header
- configured 6-14

I

Initial address 5-3
Interface 6-9

J

Job 6-44
- not accepted 7-59
- CPU 6-40, A-1, A-2
- illegal 7-61
- transfer 6-40, 7-6

Jobs (PIQ)
- in print mode 6-41

L

Length of message
- fixed 7-35
- variable 7-35

Line feed 6-54
- outputting 6-43

M

Mark 6-9
Mechanical assembly 3-1
Memory submodule 2-2, 2-11, 2-13
- evaluation 4-2
- parameters setting 7-14
- plugging in 3-1
- unplugging 3-1

Message 2-2, 7-20, 7-30, 7-35
- error A-12
- printing 6-44
- status A-12

Message buffer 6-40
- deleting 6-44, 6-55

Message frame
- receive 1-3
- send 1-3
- transmission 1-3

Message length 7-20, 7-24
- greater than 256 bytes 7-60

Message text 1-1, 1-2, 2-2, 6-19
- ASCII characters 6-20
- configured in the memory submodule 6-45
- configuring 6-2, 6-19
- entering 6-11
- inserting 6-22
- maximum length 6-27
- number 6-19, 6-20, 6-25
- output 6-1, 6-47
- print 6-42, 6-45
- setting 6-5
- with control parameters 6-32, 6-33
- with place holders 6-25, 6-27
- without place holders 6-21

Message text printout
- configurarion data 6-12

Message texts on the memory submodule
- configuring 6-46

Message, receiving
- fixed-length 7-48
- variable-length 7-50

Message, sending
- variable length 7-35, 7-32, 7-36

Methods of transmisson 2-7, 2-8
Mode 1-1, 6-44
- changing 7-4
- setting 6-5

Mode parameters 6-15
- setting 6-15

Module
- status (PII) 6-34

O

Operating mode 6-15
- changing 6-3, 6-15
- print mode 6-15
- setting 6-57

Operating mode ASCII mode
- interpretive 6-15
- transparent 6-15

Output message 5-1

P

Page feed 6-53
- executing 6-43

Page format 6-13

Page number
- setting 6-43, 6-52

Parameter block 6-4, 6-5, 6-6, 7-5, 7-15
- entering 6-7
- entering in DB 1 7-14
- input 6-6

Parameter block number 7-14

Parameter setting 6-18
- CP 521 6-16
- DB 1 6-18
- with memory submodule 7-14

Parameter setting data 2-3, 2-5, 6-2, 6-4
- ASCII mode A-9
- print mode A-6
- transferring 6-44, 6-57, 7-5, 7-6, 7-7

Parity 6-9

Parity bit 2-5, 2-6, 6-9, 6-10
- ”mark” 6-9
- ”space” 6-9

Parity error 7-59

Peripheral device 1-1, 2-2
- compatible 5-1

Permanent wire break 7-60
- on the peripheral device line

Permission to receive 7-42

PII 5-3, 6-38, 7-6, 7-55, 7-56, 7-57, 7-58, A-3, A-4, A-5

Pin assignment
- Subminiature D connector 3-3
- TTY interface 3-5, 3-6
- V.24 interface 3-6

PIQ 5-3, 6-46, 6-51, 6-58, 6-59, 6-60, 6-61, 7-6, 7-7, 7-8, 7-15, A-9, A-10, A-11
- in interpretive ASCII mode 7-10
- in transparent ASCII mode 7-9
- in print mode 7-9

Place holder 6-11, 6-20, 6-22, 6-28, 6-46
- date 6-24
- entering 6-22
- message text 6-24, 6-25
- time 6-24
- variable 6-28

Point-to-point connection 1-3, 3-7, 7-1, B-1
Print mode 1-1, 1-2, 2-2, 2-3, 5-1, 6-59, A-10
- interrupt 4-3
- jobs (PIQ) 6-41
- operating mode 6-15
- parameter setting data A-6
- status (PII) 6-37
Printout of all messages 6-44
Process input image (PII) 5-3
Process output image (PIQ) 5-3
Protocol
- XON 2-7, 2-8, 6-11, 6-59
- XOFF 2-7, 2-8, 6-11, 6-59

R

Real-time hardware clock
- integrated 2-3
Receive
- after DTR 7-59
- after XOFF 7-59
Receive buffers
- deleted 6-3
Receive jobs 7-19
Receive mailbox 2-2, 7-39
- overflow 7-60
Receive message 7-19, 7-39, 7-42
Restart procedure 4-1

S

Send buffers
- deleted 6-3
Send jobs 7-19
Send mailbox 2-2
Send permission 7-24
Sending messages 7-19, 7-20, 7-21
Separator 6-19, 6-20
Serial interface 1-1, 2-2, 2-5
- setting 6-5, 6-8, 6-57, 6-58, B-3
Setting
- CP 521 6-4, 7-2, 7-6
- CP 521 parameters 6-2
- for entering message texts 6-5
- mode parameters 6-15
- serial interface 6-5, 6-8
Settings (clock) 6-50
Signals 2-9
Slot 5-3
Slot addressing 5-3
Space 6-9
Start bit 2-5, 2-6, 6-10
Startup
- CP 521 7-3
Status message A-12
Stop bit 2-5, 2-6, 6-10
Subminiature D connector 3-2
- pin assignment 3-3

T

Terminal diagram
- TTY interface 3-5, 3-6
- V.24 interface 3-6
Text entering 6-7

Time
- error 7-61
- inserting 6-22

Time of day 2-2
- inserting 6-1
- reading 6-39
- setting 6-50

Transmission mode 1-3
TTY current-loop interface 1-1
TTY interface 2-2
- pin assignment 3-5, 3-6
- terminal diagram 3-5, 3-6

U

Unplugging
- module 3-1

User submodule 2-11

V

V.24 interface 1-1, 2-2
- setting 6-62
- terminal diagram 3-6

Variable 1-2, 2-2, 6-28
- inserting 6-1, 6-22, 6-23
- data formats 6-29

W

Waiting times 6-10

X

XON/XOFF listing
- setting 6-63

XON/XOFF protocol 2-7, 2-8, 6-11, 6-59

Siemens AG
AUT E1114B
Postfach 1963
Werner-von-Siemens-Str. 50

D-92209 Amberg
Fed. Rep. of Germany

Suggestions

Corrections

For publication:

Title: **CP 521 Printer / ASCII Communications Module**

From:

Name

Firm/Dept.

Address

Tel. No.

Order No.: 6ES5 998-0UD21

Edition: 4

Should you come across any printing errors when reading this publication, we would ask you to inform us accordingly, using this form. We would also welcome any suggestions you may have in the way of improvement.

Suggestions and /or corrections