SIMATIC S5
CP 521 SI
Communications Processor

Manual

EWA 4NEB 812 6072-02b

Edition 03

EWA 4NEB 812 6072-02b

Preface

Introduction

System Overview 1

Technical Description 2

Installation Guidelines 3

Principle of Operation 4

Printer Driver 5

ASCII Driver 6

Terminal Driver 7

3964(R) Driver 8

SINEC L1 Driver 9

Appendices A/B/C/D

Index

Summary

Page

Introduction xiii

1 System Overview . . . . 1 - 1

2 Technical Description . . . . 2 - 1

2.1 Approbations and Tests . . . . 2 - 1

2.2 Notes for Machine Manufacturer . . . . 2 - 2

2.3 Technical Specification . . . . 2 - 3

2.4 Memory Submodules . . . . 2 - 4

2.5 Serial Interface . . . . 2 - 5

2.6 Integral Real-Time Clock . . . . 2 - 6

2.7 Backup Battery . . . . 2 - 6

2.8 LEDs . . . . 2 - 6

2.9 Addressing . . . . 2 - 7

2.10 List of Accessories and Order Numbers . . . . 2 - 8

3 Installation Guidelines . . . . 3 - 1

3.1 Assembling and Dismantling the CP 521 SI . . . . 3 - 1

3.2 Dimensional Drawing . . . . 3 - 4

3.3 Wiring of Programmable Controllers for EMC . . . . 3 - 5
3.3.1 Routing of Cables . . . . 3 - 5
3.3.2 Equipotential Bonding . . . . 3 - 7
3.3.3 Shielding of Cables and Lines . . . . 3 - 8

3.4 Further Notes on System Configuration and Installation . . . . 3 - 10

4 Principle of Operation . . . . 4 - 1

4.1 General Principle of Operation . . . . 4 - 1
4.1.1 CPU CP 521 SI . . . . 4 - 2
4.1.2 CP 521 SI Peripheral Device . . . . 4 - 3
4.1.3 Module RAM . . . . 4 - 3

4.2 Access to the Transfer Memory . . . . 4 - 4

Page

4.3 Data Transmission Format . . . 4 - 6

4.4 Status Byte, Status of the Peripheral Device and Current Clock Data . . . 4 - 8
4.4.1 Status Byte . . . 4 - 8
4.4.2 Status of the Peripheral Device . . . 4 - 11
4.4.3 Current Clock Data . . . 4 - 12

4.5 Restart Characteristics . . . 4 - 16
4.5.1 Checking the Functional Capability of the Module . . . 4 - 16
4.5.2 Checking the Battery . . . 4 - 16
4.5.3 Memory Submodule Evaluation . . . 4 - 17
4.5.4 Clock Test . . . 4 - 17

4.6 Behaviour during Operation . . . 4 - 18
4.6.1 Printer Output . . . 4 - 18
4.6.2 Bidirectional Data Transmission . . . 4 - 19

5 Printer Driver . . . 5 - 1

5.1 Prerequisites for Using the Printer Driver . . . 5 - 2

5.2 Peripheral Interface Connections . . . 5 - 4

5.3 Assigning CP 521 SI Parameters in Print Mode . . . 5 - 6
5.3.1 Assigning the Serial Interface Parameters (Parameter Blocks 0, 1 and 2) . . . 5 - 9
5.3.2 Parameter Assignment Data for Entering Message Texts (Parameter Block 3) . . . 5 - 12
5.3.3 Parameter Assignment Data for Message Text Printout (Parameter Blocks 4 to 6) . . . 5 - 12
5.3.4 Configuring the Character Conversion Table (Parameter Block 8) . . . 5 - 16
5.3.5 Clock Correction Factor (Parameter Block 9) . . . 5 - 16
5.3.6 CP 521 SI Parameter Assignment Example . . . 5 - 17

5.4 Printing Parameter Assignment Data . . . 5 - 20

5.5 Configuring Message Texts . . . 5 - 22
5.5.1 Entering Place Holders . . . 5 - 23
5.5.2 Place Holders for Date and Time of Day . . . 5 - 25
5.5.3 Place Holders for Message Texts . . . 5 - 26
5.5.4 Place Holders for Variables . . . 5 - 29
5.5.5 Control Parameters . . . 5 - 32

5.6 Printing Message Texts . . . 5 - 35
5.6.1 Setting the Page Number . . . 5 - 37
5.6.2 Outputting Message Texts with and without CR/LF at the End . . . 5 - 38
5.6.3 Executing Page Feed . . . 5 - 41

Page

5.6.4 Outputting a Line Feed . . . . . 5 - 42
5.6.5 Deleting the Message Buffer . . . . . 5 - 42
5.6.6 Printing All Configured Message Texts . . . . . 5 - 43

5.7 Status of the CP 521 SI in Print Mode . . . . . 5 - 44

6 ASCII Driver . . . . . 6 - 1

6.1 Prerequisites for Operation with ASCII Drivers . . . . . 6 - 5

6.2 Peripheral Interface Connections . . . . . 6 - 7

6.3 Assigning the CP 521 SI Parameters for ASCII Mode . . . . . 6 - 9
6.3.1 Assigning the CP 521 SI Parameters with the Memory Submodule . . . . 6 - 10
6.3.2 Assigning the CP 521 SI Parameters in the User Program . . . . . 6 - 17

6.4 Data Transfer Between the CP 521 SI and the CPU . . . . . 6 - 21
6.4.1 Sending Message Frames . . . . . 6 - 22
6.4.2 Receiving Message Frames . . . . . 6 - 32

6.5 CPU Job Requests and CP Error Messages . . . . . 6 - 39

6.6 STEP 5 Programs for Data Transmission with ASCII Driver . . . . . 6 - 43

7 Terminal Driver . . . . . 7 - 1

7.1 Prerequisites for Using the Terminal Driver . . . . . 7 - 2

7.2 Serial Interface . . . . . 7 - 3

7.3 Assigning the CP 521 SI Parameters for Terminal Mode . . . . . 7 - 4
7.3.1 Setting the Parameter Assignment Data . . . . . 7 - 5
7.3.2 Storing the Parameter Assignment Data on the Memory Submodule . . . . . 7 - 12
7.3.3 Changing Parameter Assignment Data during Terminal Mode . . . . . 7 - 15

7.4 Data Transfer CPU CP 521 SI Terminal Outputting Message Texts on the Terminal . . . . . 7 - 17
7.4.1 Configuring the Message Texts . . . . . 7 - 17
7.4.2 Initiating Message Text Output . . . . . 7 - 18

7.5 Data Transfer Terminal CP 521 SI CPU Receiving Message Frames . . . . . 7 - 20

7.6 CPU Job Requests and CP Error Messages . . . . . 7 - 22

7.7 STEP 5 Program for Data Transmission with Terminal Driver . . . . . 7 - 24

			Page
8	**3964(R) Driver**		**8 - 1**
	8.1	Interface Lines for the 3964(R) Driver	8 - 1
	8.2	Special Features of the 3964(R) Data Transmission Protocol	8 - 6
	8.2.1	General Points on Data Transmission Procedures Using Protocols	8 - 6
	8.2.2	The 3964(R) Transmission Protocol	8 - 6
	8.3	Assigning the CP 521 SI Parameters in 3964(R) Mode	8 - 13
	8.3.1	Assigning the CP 521 SI Parameters with the Memory Submodule	8 - 14
	8.3.2	Assigning the CP 521 SI Parameters in the User Program	8 - 20
	8.4	Data Transfer with the 3964(R) Transmission Protocol	8 - 23
	8.4.1	Sending Data with the 3964(R) Transmission Protocol	8 - 23
	8.4.2	Receiving Data with the 3964(R) Transmission Protocol	8 - 28
	8.5	Error Flags in the CBS and CBR of the CP Acknowledgements	8 - 32
	8.6	STEP 5 Programs for Data Transmission with the 3964(R) Driver	8 - 35
9	**SINEC L1 Driver**		**9 - 1**
	9.1	Connecting the CP 521 SI to the BT 777 Bus Terminal	9 - 1
	9.2	Assigning CP 521 SI Parameters for SINEC L1 Operation	9 - 2
	9.2.1	Passing Parameters for SINEC L1 Operation to the Module Via the User Program	9 - 3
	9.2.2	Providing Parameters for SINEC L1 Operation Over the Memory Submodule of the Module	9 - 4
	9.3	Send Procedure from the Point of View of the Control Program	9 - 5
	9.4	Receive Procedure from the Point of View of the Control Program	9 - 9
	9.5	Point-to-Point Connection	9 - 13
	9.5.1	Connecting the SINEC L1 Slave Direct to the Interface of the CP 521 SI	9 - 13
	9.5.2	Assigning Parameters to the CP 521 SI as ”Point-to-Point Master”	9 - 14
	9.5.3	Programming Data Transfer between the CP 521 SI and the Slave	9 - 14
	9.6	STEP 5 Program for Data Transmission with the SINEC L1 Driver	9 - 15

Page

Appendices

A Summary . . . . **A - 1**

A.1 Combinations of the Most Important Parameters . . . . A - 1

A.2 ASCII Code Table . . . . A - 2

B Active and Passive Faults in Automation Equipment . . . . **B - 1**

C Function Blocks to Support Data Transmission with the CP 521 SI . . . . **C - 1**

D Connecting Cables; Ordering Form . . . . **D - 1**

Index

Preface

The CP 521 SI communications processor replaces the CP 521 input/output module. The CP 521 SI is equipped with additional data transmission procedures which considerably expand the area of application of the SIMATIC programmable controllers of the lower performance range. We have endeavoured to make the CP 521 SI as upwardly compatible as possible with the CP 521. However, in individual cases we have accepted a certain degree of incompatibility in favour of the most user-friendly solution to a problem.

The CP 521 SI (**S**erial **I**nterface) is a powerful module with its own central processing unit. It belongs to the range of input/output (I/O) modules for the S5-100U, S5-95U and S5-90U programmable controllers (PLC) and ET 200U (distributed I/O). In order to use the communications processor (CP) to its full potential, you require detailed information.

The performance capabilities of the S5-100U, S5-95U and S5-90U PLCs and the ET 200U are constantly being enhanced. The product manuals for these systems have therefore already become very large.

For this reason, the CP 521 SI has its own manual which contains only those facts and examples important for the use of this module. We have also tried to meet the increased requirements made of technical documentation. This 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 they require to work with the CP 521 SI.

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.

Modifications and improvements to the CP 521 SI, 6ES5 521-8MA21, version 5 and the CP 521 SI, 6ES5 521-8MA22, version 1 compared to previous versions of the CP 521 SI.

- ASCII mode without interpretation of RUB OUT and BACKSPACE
  The CP 521 SI supports another ASCII driver mode. The only difference of the new mode to the interpretive ASCII driver is that the RUB OUT ($7F_H$) and BACKSPACE (08_H) characters are not evaluated. All other functions, such as XON/XOFF protocol and receiving of message frames with one or two end-of-text characters are identical with the functions of the interpretive ASCII driver. We call this new ASCII mode "Interpretive ASCII driver mode ".

- Additional CPU job "Delete receive mailbox contents"
  With the job "Delete receive mailbox contents" ($D0_H$) you have now the possibility to delete the receive mailbox before receiving the relevant message frame.

Modifications and improvements to the CP 521 SI, 6ES5 521-8MA21, version 2 to 4, compared to its predecessor, the CP 521 SI, 6ES5 521-8MA21, version 1

- Additional acknowledgements of the CP 521 SI, version 2 to 4
  The CP 521 SI acknowledges all parameter assignment jobs, print jobs and the "Set real-time clock" job with the terminating frame 5000_H **and** specifies the acknowledged job in word 6 of the CP response signal.
  For assigning CP 521 SI parameters through the user program, we recommend that you use the parameter assignment FBs listed in this manual. If you use other parameter assignment FBs (e.g. FBs from the CP 521 SI, version 1, manual, please note that the terminating frame 5000_H must be deleted with a 0000_H blank job. The Send and Receive function blocks in this manual can execute only if a terminating frame is no longer pending.

- Parameter assignment data backup in the CP 521 SI, version 2 to 4
  In the event of a power failure or if the CP 521 SI is powered down, the parameter assignment data are retained in the RAM of the CP 521 SI, provided a backup battery has been inserted in the module. If the module has a backup battery (and no memory submodule is plugged in), the CP 521 SI is automatically initialized with the data in the RAM on power-up, i.e. the CP 521 SI has exactly the same initializing parameters following power-up at it had prior to being powered down.

- Operation of the CP 521 SI, version 2 to 4, in conjunction with ET 200U
  On power failure, the parameter assignment data of the CP 521 SI in an ET 200U station are lost if the CP 521 SI does **not** have a backup battery. In such an event, the CP 521 SI must be re-initialized (which takes place automatically via the CPU, provided the watchdog is switched on). If the watchdog is off, the new parameters must be assigned to the CP 521 SI via memory submodules to avoid malfunctions.

- Acknowledgements and error messages in the CBS and CBR of the CP 521 SI, version 2 to 4
  The error codes for the same information in the CBS and CBR bytes are now the same in the case of the 3964(R) driver and the SINEC L1 driver. Exception: With the SINEC L1 driver, bit 7 of the CBR has signal status ”1” after correct fetching of a frame (in the case of the 3964(R) driver, signal status ”0”).

Modifications and improvements to the CP 521 SI, 6ES5 521-8MA21, release 2, compared to its predecessor, the CP 521, 6ES5 521-8MA21.

- Connector assignments of the serial interface
  The connector assignments of the serial interface have changed. You will find the connector assignments at the beginning of Chapters 5, 6, 7, 8 and 9.

- Data format for assigning memory submodule parameters (parameters in DB1)
  All parameters in DB1 are now to be entered uniformly in ASCII code.

- Terminating frame of the CP
  In the case of the CP 521, the terminating frame (5000_H) was automatically erased from the bus after two PLC bus cycles. This is no longer the case with the CP 521 SI. The terminating frame of the CP 521 SI (5000_H) remains until overwritten by the acknowledgement of a subsequent job request.
  (Example: After evaluating the terminating acknowledgement, you write the subsequent job request 0000_H to word 0 of the CP so that the terminating acknowledgement is overwritten by the current time and date.)

- Format of fixed-point variables (printer output)
  The format ”KFa,b” is now based on the high-level languages;
  a: Number of digits to be printed
  b: Number of decimal places

- Testing the CP 521 SI functions, using the programmer "FORCE VAR" function
  Because of the serial LAN interface of the individual modules of the S5-100U spectrum, reliable testing or processing of the CP 521 SI using the "FORCE VAR" function is not possible, and is therefore not recommended.

Description of Contents

This manual is a comprehensive description of the CP 521 SI. 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, S5-95U and S5-90U PLCs and ET 200U. The ”Technical Description” contains general information on the technical specifications of the CP 521 SI, details of the serial interface (pin assignments) and addressing of the module, as well as a list of accessories.

- Installation
  The ”Hardware Installation” contains all the information you require to install the module and connect it to a printer or other peripheral device.

- Principle of Operation
  This is a general description of how data transfer between the CPU, the CP 521 SI and the I/O modules is coordinated, as well as the restart characteristics of the module.

- Functional Description
  We have devoted a chapter to each possible CP 521 SI mode (printer driver, ASCII driver, terminal driver, etc.). Each of these chapters contains the information necessary for operating the module in the relevant mode. This saves you time-consuming searches in the manual. The terminal driver is an exception here. This driver is a combination of the printer driver and the interpretive ASCII mode. In the terminal driver chapter, we restrict ourselves for this reason to the most important points and special features of this mode, and refer you to the relevant ”Driver Chapter” in this manual.
  At the end of each chapter are executable programs to illustrate the relevant job request and acknowledgement mechanism between the CPU and the CP 521 SI.

- Appendix A: Summary
  This contains information in list form to make working with the CP 521 SI easier, as well as an overview of the possible combinations of the most important module parameters.

- Appendix B: Active and Passive Faults in Automation Equipment

- Appendix C: Function Blocks for Supporting Data Communications with the CP 521 SI
  This section list the function blocks you can use for checking data transmissions and optimizing them with regard to time. You can also link these FBs into the STEP 5 programs listed in the individual "Driver Chapters” for cyclic program scanning in each driver mode of the CP 521 SI.

- Appendix D: Connecting Cable and Ordering Form
  Appendix D contains a form with which you can order a cable of your choice for connecting the CP 521 SI to the peripheral device in question.

Since the module can only be used in conjunction with the S5-100U, S5-95U and S5-90U PLCs and the ET 200U distributed I/O system, it is assumed that you have a copy of one of the relevant manuals. The basic information on the STEP 5 programming language contained in these manuals is sufficient for operating the CP 521 SI.

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.
- The overview is followed by a breakdown of the subject matter.
- Each chapter is preceded by a breakdown of its subject matter.
  The individual chapters are divided 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: Programmable controller (PLC)
- Footnotes are identified by superscripts consisting of an asterisk ”*”. The actual footnote is generally at the bottom left of the page or below the relevant table or figure.
- Lists are indicated by a bullet (•), as in this list for example, or with a dash (-). Instructions for operator actions are indicated by black triangles ( ).
- Cross references are shown as follows:
  ”(Section 6.3.2)” refers to Subsection 6.3.2.
  No references are made to individual pages.
- All dimensions in drawings etc. are given in millimeters (mm).
- Value ranges are indicated as follows: 17 to 21.
- Information of special importance is enclosed within black-edged boxes:

Note

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.

Safety-Related Guidelines for the User

This document provides the information required for the intended use of the CP 521 SI. The documentation is written for technically qualified personnel.

Qualified personnel as understood by the safety-related guidelines printed in this document or on the product itself are defined as being

- either system planning and design engineers who are familiar with the safety concept of automation equipment;
- or operating personnel who have been trained to work with automation equipment and are conversant with the contents of this document;
- or system start-up and service personnel who have been trained to repair such automation equipment and are authorized to start up, ground and tag circuits, equipment and systems in keeping with the relevant safety standards and codes of practice.

Danger notices

The following notices are intended, on the one hand, for your personal safety and, on the other, to prevent damage to the product described or equipment connected to it.

The safety notices and warnings highlighted in this manual by the terms and pictograms defined here are intended for the personal safety of both the user and his or her service personnel, as well as to prevent damage to property and equipment. The terms used in this manual and marked on the product itself have the following significance:

Warning	**Note**
indicates that death, severe personal injury or substantial damage to property can result if the proper precautions are not taken.	indicates important information about the product or the respective part of the instruction manual to which special attention is drawn.

Proper usage

⚠ **Warning**

- This equipment may only be used for the applications listed in the catalog and technical description and only in conjunction with products and components of other vendors recommended by Siemens.
- Satisfactory and safe operation of the equipment can only be guaranteed if it is properly handled (transported), stored and installed, and operated and maintained with the appropriate care.

Courses

Siemens provide SIMATIC S5 users with extensive opportunities for training. For more information, please contact your Siemens representative.

1 System Overview

Figures

1-1. System Environment of the Module . . . . 1 - 1
1-2. Complete Configuration: S5-100U with CP 521 SI and Printer . . . . 1 - 2
1-3. Complete Configuration: Point-to-Point Connection . . . . 1 - 3
1-4. Data Paths when Using the CP 521 SI in the ET 200U
Distributed I/O System . . . . 1 - 5

1 System Overview

Intelligent input/output (I/O) modules extend the area of application of SIMATIC S5 programmable controllers. They are technology-oriented and offload the central processing unit.

The CP 521 SI communications processor is a powerful and active I/O module, which can be used with the CPUs of the following S5 systems:

- S5-100U PLC (CPU 100 from 6ES5 100-8MA02 upward, CPU 102 or CPU 103).
- S5 95U/90U
- ET 200U (IM 318-B; it is recommended that the CP 521 SI's parameters be assigned with memory submodules since the parameter assignment data of the CP 521 SI may be lost on failure of the relevant station).

The CP 521 SI enables data transfer between the CPU and a peripheral device connected to the CP 521 SI. The CP 521 SI is equipped with three interfaces for this purpose:

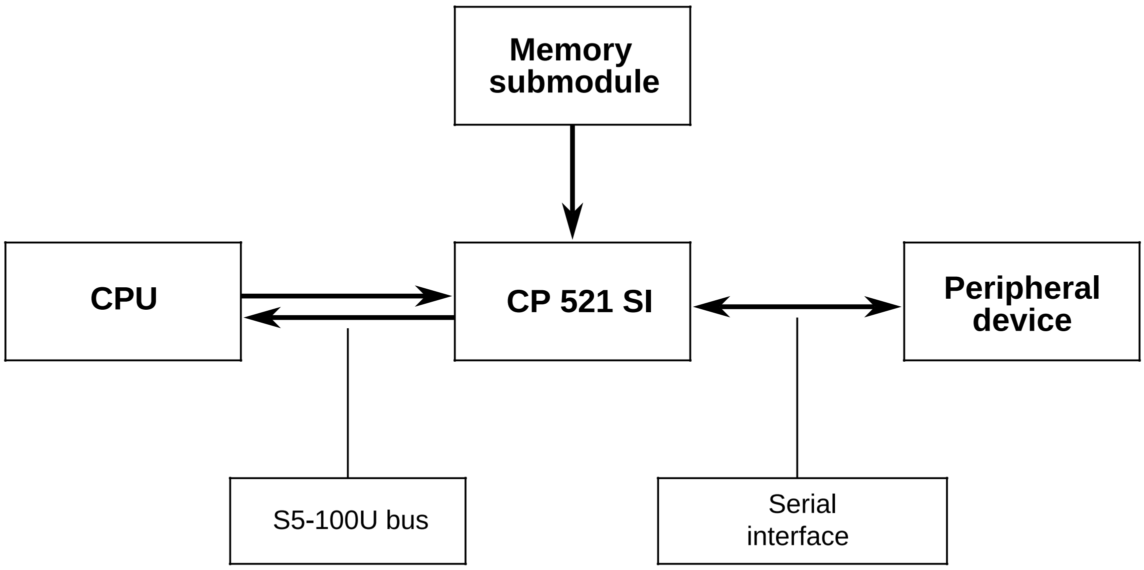


Figure 1-1. System Environment of the Module

Interface to the S5-100U bus for communications with the CPU:
The control program transfers data to the module over the S5-100U bus and evaluates information from the module.

Receptacle for memory submodule:
The memory submodule is required for parameter assignment data and message texts. The configured data is transferred to the CP 521 SI on restart.

Serial interface for data exchange with the peripheral device:
The peripheral device and the CP 521 SI are linked over a serial interface. You have the option of a TTY current loop interface or an RS-232C (V.24) interface. The signals of both interfaces are transmitted over a common subminiature D socket connector.

Assignment of serial interface parameters is supported by the DB editor of the programmers. The serial interface parameters are either stored in the memory submodule in DB1 or transferred direct over a user program.

The CP 521 SI can be used for unidirectional and bidirectional data traffic.

Unidirectional data traffic

The CP 521 SI is equipped with a printer driver for unidirectional data traffic. When a printer driver is used, the CP 521 SI assumes the following to be connected to the serial interface:

- A printerwith TTY interface (active) or
- A printer with RS-232C (V.24) interface.

This allows the user to list process states and process faults. Printing out message texts does not increase the response time of the programmab"le controller.

The following can be printed out:

- Texts configured by the useron a memory submodule in data blocks (DB) 2 to 63.
- Time of day and date provided bythe module's own clock.
- Values forvariables transferred to the CP 521 SI overthe 100U bus.

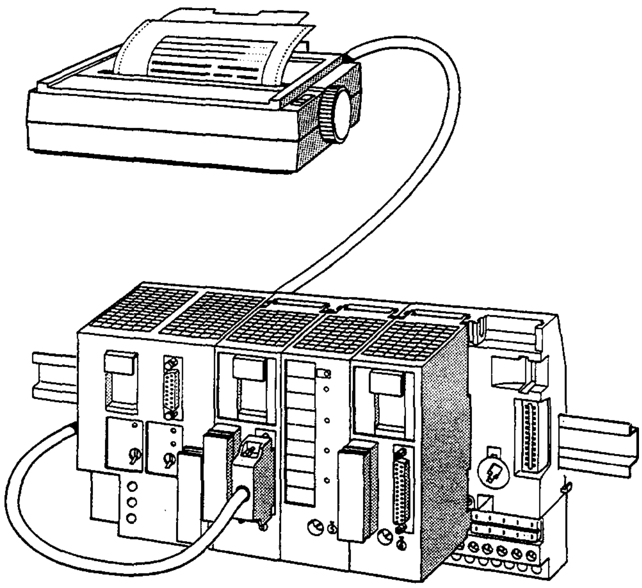


Figure 1-2. Complete Configuration: S5-1OOU with CP 521 SI and Printer

Bidirectional data traffic

The following drivers are implemented for bidirectional data traffic:

- ASCII driver, transparent
- ASCII driver, interpretive model and interpretive mode II
- "3964(R)" driver
- SINEC L1 driver, master (point-to-point connection)
- SINEC L1 driver, slave
- Terminal driver

The use of these drivers enables the transfer of data frames between the CPU and a terminal device or communications device connected to the CP 521 S1.

The CP 521 SI can be used to:

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

The following can be used as peripheral devices:

- Keyboard
- Terminal
- A further CP 521 SI
- CP 523
- CP 524/CP 525-2 (in conjunction with the special driver6ES5 897-2AB11)
- CP544
- CPU 928B
- CPU 944 (with ASCII driver, 3964(R) driver)
- PCs (e.g. in conjunction withthe PRODAVE DOS 64R software toolbox, Order No. 6ES5897-2UD1 1)
- Other peripheral devices with serial interface, e.g. barcode readers

Your choice of transmission method and peripheral devices depends on the purposes of your data transmission. Using the module in bidirectional mode, for example, allows you to network programmable controllers together (point-to-point connection).

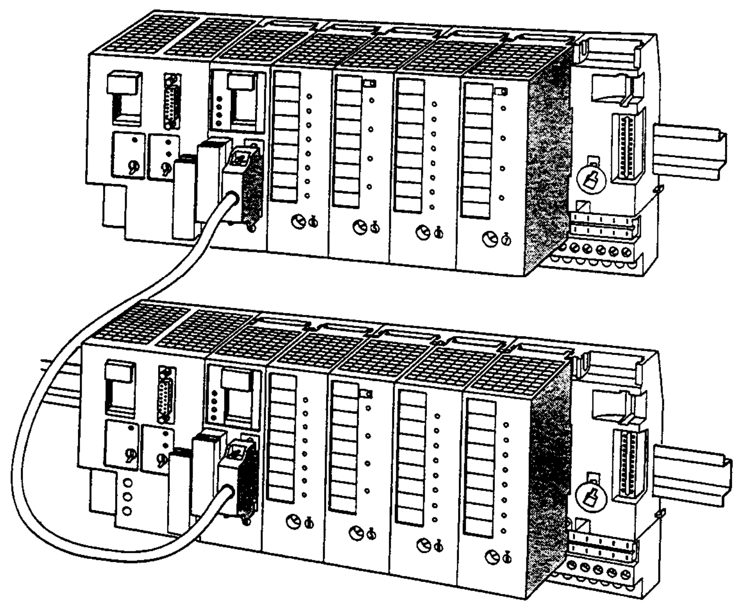


Figure 1-3. Complete Configuration: Point-to-Point Connection

Maximum data throughput:

> **Note**
>
> The maximum data throughput is six bytes of useful data per two program scan cycles.

Calculating the maximum throughput time t_D [ms]:

$$\mathbf{t_D\,[ms] = 2\times(\underbrace{t_{scan}\,[ms]}_{\text{Transmit and acknowledge CPU job}}+\underbrace{t_{scan}\,[ms]\times\frac{L_{frame}\,[bytes]}{6}}_{\text{Transmit and acknowledge 6-byte data blocks of the frame}})}$$

t_D [ms]: Max. throughput time
t_{scan} [ms]: PLC scan time=program scan cycle time+data cycle time
L_{frame} [bytes]: Frame length

This means that, given a program scan cycle time of 50 ms, up to 54 bytes can be transmitted per second.

Integral real-time clock

The CP 521 SI has its own real-time clock, which has battery backup when the module is not powered. Irrespective of the selected mode of the CP 521 SI, the clock data can be read by the CPU and used in the user program for date-dependent and time-dependent tasks.

What to remember when using the CP 521 SI in the ET 200U distributed I/O system

If you are using the CP 521 SI in the ET 200U distributed I/O system, you must allow for additional data paths and delays when transferring data between the CPU and the CP 521 SI.

Fig. 1-4 illustrates these additional data paths

between the CPU and the IM 308-B master interface module (internal station bus),
via the SINEC L2-DP LAN between the IM 308-B and the IM 318-B,
inside the IM 318-B slave interface module
between the IM 318-B slave interface module and the CP 521 SI (internal station bus).

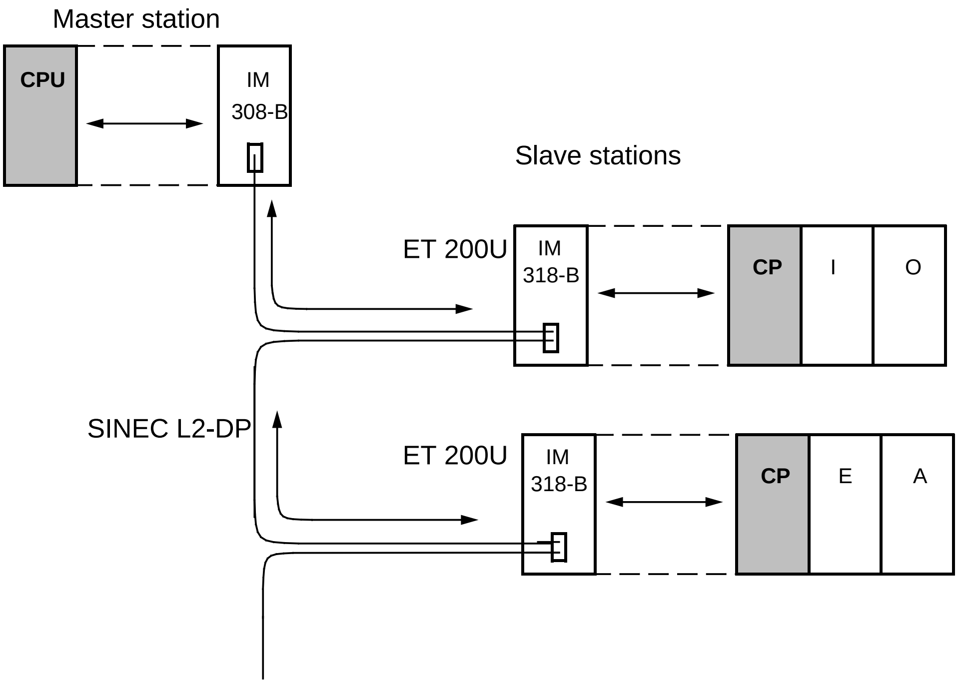


Figure 1-4. Data Paths when Using the CP 521 SI in the ET 200U Distributed I/O System

The delays occurring in the above data paths depend on the following:

- The acknowledgement mechanism between the CP 521 SI and the I/Os connected to it (ASCII driver without hardware handshaking, ASCII driver with hardware handshaking, 3964(R) protocol etc.)
- The baud rate on the SINEC L2-DP LAN and the internal station buses
- The number of ET 200U slave stations
- The number of I/O modules connected to the slave stations in addition to the CP 521 SI

Recommendations in connection with the use of the CP 521 SI in the ET 200U distributed I/O system

Note

Work with as high a baud rate as possible if you are using the CP 521 SI in the ET 200U distributed I/O system. High baud rates shorten the delays on the LAN channels and internal buses, within the IM 318-B.

Note

Appendix C of the manual lists function blocks that you can link into your user program to optimize data transfer times and for troubleshooting purposes (cyclic program scanning: OB1):

- FB 15: Measures the transfer times of the transmit and receive frames between the CPU and the CP 521 SI; this enables you to optimize the initiation of transmit and receive frames with respect to time
- FB 18: Lists the jobs sent to the CP 521 SI and the acknowledgements received from the CP 521 SI (troubleshooting support)
- FB 19: Re-initializes the CP 521 SI if no more data transfers take place between the CP and the CPU within 30 s (QW 0 and IW 0 unchanged)

Note

In the event of a power failure, the parameter assignment data of the CP 521 SI in an ET 200U station are lost if the CP 521 SI does **not** have a backup battery. In such an event, the CP 521 SI must be re-initialized (which takes place automatically via the CPU, provided the watchdog is switched on). If the watchdog is off, the parameters must be re-assigned to the CP 521 SI via memory submodules to avoid malfunctions.

2 Technical Description

2.1	Approbations and Tests	2 - 1
2.2	Notes for Machine Manufacturer	2 - 2
2.3	Technical Specification	2 - 3
2.4	Memory Submodules	2 - 4
2.5	Serial Interface	2 - 5
2.6	Integral Real-Time Clock	2 - 6
2.7	Backup Battery	2 - 6
2.8	LEDs	2 - 6
2.9	Addressing	2 - 7
2.10	List of Accessories and Order Numbers	2 - 8

Tables

2-1. Erasing and Storing Data on Memory Submodules . . . . 2 - 4
2-2a. Overview of Permissible Plug-In Memory Submodules
that can be Ordered at Present . . . . 2 - 4
2-2b. Overview of Plug-in Memory Submodules that Can Still be Used . . . . 2 - 4
2-3. Pin Assignments of the 25-Pin Subminiature D Connector of the CP 521 SI 2 - 5
2-4. Slot Addresses . . . . 2 - 7

2 Technical Description

2.1 Approbations and Tests

The general technical specifications include standards and test specifications which the CP 521 SI meets and fulfills and which were used during testing of the CP 521 SI.

UL/CSA Approbations

The following approbations have been granted for the CP 521 SI:
UL-Recognition Mark
Underwriters Laboratories (UL) to UL standard 508, Report 116536
CSA Certification Mark
Canadian standard Association (CSA) to C22.2 standard No. 142, Report LR 48323

CE-Marking

Our products meet the requirements of EU directive 89/336/EEC "Electromagnetic Compatibility" and the harmonized European standards (EN) listed therein.

In accordance with the above-mentioned EU directive, Article 10, the EU declarations of conformity are held at the disposal of the competent authorities at the address below:

Siemens Aktiengesellschaft
Bereich Automatisierungstechnik
AUT E 14
Postfach 1963

D-92209 Amberg
Federal Republic of Germany

Area of Application

SIMATIC products have been designed for use in the industrial area.
With individual approval, SIMATIC products can also be used in the domestic environment (household, business and trade area, small plants). You must acquire the individual approval from the respective national authority or testing board.

Area of Application	Requirements to:	
	Emitted interference	Immunity
Industry	EN 50081-2 : 1993	EN 50082-2 : 1995
Domestic	Individual approval	EN 50082-1 : 1992

Observing the Installation Guidelines

S5 modules meet the requirements if you observe the installation guidelines described in the manuals when installing and operating the equipment (Section 3).

2.2 Notes for the Machine Manufacturer

The SIMATIC programmable controller system is not a machine as defined in the EU Machinery Directive. There is therefore no declaration of conformity for SIMATIC with regard to the EU Machinery Directive 89/392/EEC.

The EU Machinery Directive 89/392/EEC regulates requirements relating to machinery. A machine is defined here as an assembly of linked parts or components (see also EN 292-1, Paragraph 3.1).

SIMATC is part of the electrical equipment of a machine and must therefore be included by the machine manufacturer in the declaration of conformity procedure.

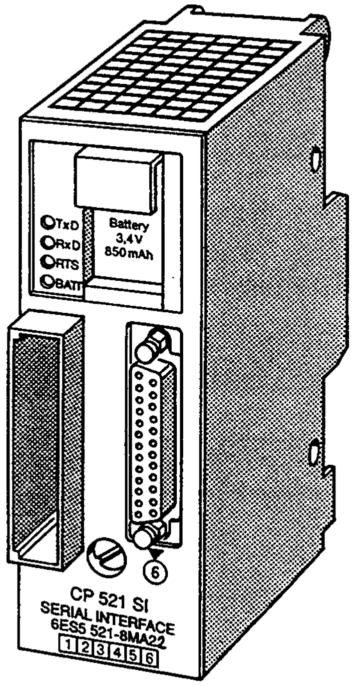
The EN 60204-1 standard (Safety of Machinery, Electrical Equipment of Machines, Part 1, Specification for General Requirements) applies for the electrical equipment of machinery.

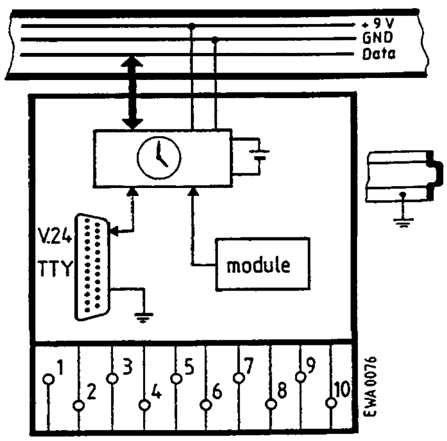
The table below is designed to help you with the declaration of conformity and to show which criteria apply to SIMATIC according to EN 60204-1 (as at June 1993)

EN 60204-1	Subject/Criterion	Remarks
Paragraph 4	General requirements	Requirements are met in the devices are mounted/installed in accordance with the installation guidelines. Please observe the explanations in "Notes on CE Marking of SIMATIC S5".
Paragraph 11.2	Digital input/output interfaces	Requirements are met.
Paragraph 12.3	Programmable equipment	Requirements are met if the devices for protection of memory contents against change by unauthorized persons are installed in locked cabinets.
Paragraph 20.4	Voltage tests	Requirements are met.

2.3 Technical Specifications

See **the manual of the relevant programmable controller for climatic, mechanical and electromagnetic conditions.**





Galvanic isolation	TTY signals are floating
Memory submodule	EPROM/EEPROM
Serial interface	RS-232C (V. 24)/TTY passive (active)
Address identifier (for ET 200U):	223
Transmission method:	Asynchronous 10-bit character frame/ 11-bit character frame
Transmission rate	110 to 9600 baud
Permissible cable length	
-TTY	Is calculated from voltage drop on cable and typical receiver voltage drop 1.5 V or typical sender voltage drop 0.9 V max.1000 m
- RS-232C (V.24)	max. 15 m (at 9600 baud)
LED displays	
-TxD (green)	Send
- RxD (green)	Receive
- RTS (green)	Ready to send
- BATT (yellow)	Battery failure
Backup battery Lithium ½AA	3.6 V/850 mAh
Backup time	at least 1 year
Degree of protection	IP 20
Permissible ambient temperature	
- Horizontal mounting	0 to 60 °C
- Vertical mounting	0 to 40 °C
Relative humidity	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

2.4 Memory Submodules

You require a plug-in memory submodule if you want to print out message texts or store parameter assignment data.

You can use two types of memory submodules:

Table 2-1. Erasing and Storing Data on Memory Submodules

Submodule Type	Delete with	Store Programs with
EPROM	UV erasing facility	PG
EEPROM	PG	PG

Table 2-2a. Overview of Permissible Plug-in Memory Submodules that Can be Ordered at Present

Submodule Type	Submodule Designation	Capacity	Programming No.
EPROM	6ES5 375 - 1LA15	1 x 8 KB	411
EPROM	6ES5 375 - 1LA21	2 x 8 KB	412
EPROM	6ES5 375 - 1LA41	2 x 16 KB	417
EEPROM	6ES5 375 - 0LC11	1 x 2 KB	202
EEPROM	6ES5 375 - 0LC31	1 x 8 KB	211
EEPROM	6ES5 375 - 0LC41	2 x 8 KB	212

Table 2-2b. Overview of Plug-in Memory Submodules that Can Still be Used

Submodule Type	Submodule Designation	Capacity	Programming No.
EPROM	6ES5 375 - 0LA15 or 11	1 x 8 KB	11
EPROM	6ES5 375 - 0LA21	2 x 8 KB	12
EPROM	6ES5 375 - 0LA41	2 x 16 KB	17

Warning

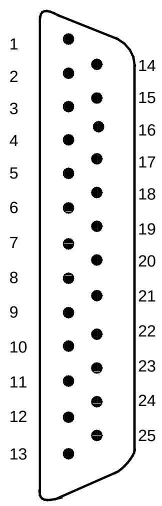
- The memory submodule of the CP 521 SI may only be plugged in and removed in the POWER OFF state.
- When storing data on a memory submodule, make sure you enter the right programming number on the programmer. Wrong programming numbers, e.g. an old programming number (-0AAxx) for a new CMOS memory submodule (-1AAxx), can lead to the destruction of the submodule.

2.5 Serial Interface

The CP 521 SI is has a serial interface port. You can choose between a current interface (TTY) or a voltage interface (RS-232C (V.24)) by changing parameters. The cables of both interfaces connect with a 25-pin subminiature D connector.

The TTY interface is designed for passive operation. If 24 V can be fed through the subminiature D connector, the TTY interface can also be used for active operation. There is no galvanic isolation in the case of an ***active*** TTY interface. Links of up to 1000 m are possible.

Table 2-3. Pin Assignments of the 25-Pin Subminiature D Connector of the CP 521 SI

View	Pin No.	Signal Name	Meaning
	1	-	Disabled
	2	TxD	Send data (V.24)
	3	RxD	Receive data (V.24)
	4	RTS	Request to send (V.24)
	5	CTS	Clear to send (V.24)
	6	DSR	Data set ready (V.24)
	7	GND	Signal ground (RS-232C (V.24))
	8	-	Disabled
	9	TTY IN+	TTY receive line+
	10	TTY IN-	TTY receive line -
	11	-	Disabled
	12	-	Disabled
	13	P24	+24 V for active TTY
	14	-	Disabled
	15	-	Disabled
	16	-	Disabled
	17	20 mA	Current source TTY *****
	18	TTY OUT+	TTY send line+
	19	20 mA	Current source TTY *****
	20	DTR	Data terminal ready
	21	TTY OUT-	TTY send line -
	22	-	Disabled
	23	-	Disabled
	24	-	Disabled
	25	-	Disabled

***** If +24 V to GND (pin 7) on pin 13

> ⚠ **Warning**
>
> The unassigned (disabled) pins of the 25-pin subminiature D socket connector must not be connected as this might lead to malfunctioning of the CP 521 SI or even its destruction.

2.6 Integral Real-Time Clock

The module has a real-time clock, which has battery backup when the module is not powered.

Setting and reading the clock is done over the CPU using a programmer.

The clock supports the following functions:

- Seconds
- Minutes
- Hours (12/24 hr mode)
- Date
- Weekday (calculated from the date)
- Month
- Year (leap years accounted for)

We have reserved a subsection in the manual for setting and reading the clock data (section 4.4.3).

2.7 Backup Battery

The module has its own backup battery for securing the real-time clock data and the contents of the RAM (parameter assignment data, message buffer) when the controller is not powered.

When a power failure occurs or when the PLC is switched off, the clock data and parameters are only saved if a backup battery is inserted in the module.

Batteries should be inserted and replaced with the PLC switched on, otherwise new clock and parameter assignment data must be entered after switching it on.

A lithium battery will last at least a year with constant backup.

> ⚠ **Warning**
>
> Lithium batteries cannot be recharged. Any attempt to do so could cause an explosion! Old batteries should always be disposed of properly.

2.8 LEDs

The module is equipped with the following LEDs:

- A green send LED
- A green receive LED
- A green request-to-send LED (RTS)
- A yellow LED for displaying battery failure (BATTERY OFF/LOW)

2.9 Addressing

The CPU references the CP 521 SI in the address area of the analog channels. The address area of the module has eight bytes of input and eight bytes of output. Input data and output data are referenced over the same address area.

Please note the following in this connection:

- The module can only be plugged into slots 0 to 7.
- The address space ranges from byte 64 to byte 127.
- Eight bytes are reserved for each slot in the process image of the inputs (PII) and the process image of the outputs (PIQ) of the CPU.

The address area specified for the process image of the inputs (PII) contains data from the module to the CPU.

Conversely, information from the CPU to the module is stored under the same addresses in the process image of the outputs (PIQ).

Table 2-4. Slot Addresses

Slot	0	1	2	3	4	5	6	7
Addresses PII/PIQ	$\boxed{\textbf{64}}$ **to** 71	$\boxed{\textbf{72}}$ **to** 79	$\boxed{\textbf{80}}$ **to** 87	$\boxed{\textbf{88}}$ **to** 95	$\boxed{\textbf{96}}$ **to** 103	$\boxed{\textbf{104}}$ **to** 111	$\boxed{\textbf{112}}$ **to** 119	$\boxed{\textbf{120}}$ **to** 127
$\boxed{}$ Starting address of a slot								

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

Example: Module in slot 3: byte 2 has address 90
Module in slot 4: byte 2 has address 98

Byte 0 of the PIQ ”Job request” defines the meaning of bytes 1 to 7 in the PIQ.

2.10 List of Accessories and Order Numbers

Memory submodules

Memory submodule (EPROM) 1 x 8 KB	6ES5 375-1LA15
Memory submodule (EPROM) 2 x 8 KB	6ES5 375-1LA21
Memory submodule (EPROM) 2 x 16 KB	6ES5 375-1LA41
Memory submodule (EEPROM) 1 x 2 KB	6ES5 375-0LC11
Memory submodule (EEPROM) 1 x 8 KB	6ES5 375-0LC31
Memory submodule (EEPROM) 2 x 8 KB	6ES5 375-0LC41

Printer

See S5-90U, S5-95U and S5-100U Programmable Controller Catalog ET 100 Intelligent Electronic Terminator	ST 52.1
or	
see Programmer Catalog	ST 59

Programmers

See Programmer Catalog	ST 59

Backup battery

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

Connecting cables

Connecting cables cannot be ordered by specifying a Siemens Order No.

However, if you require connecting cables, you can order them through a Siemens office. You will find the necessary order form in Appendix D of this manual. Complete this form, specifying the connecting cable you need, and send it to the address given at the top of the form.

3 Installation Guidelines

3.1 Assembling and Dismantling the CP 521 SI . . . . 3 - 1

3.2 Dimensional Drawing . . . . 3 - 4

3.3 Wiring of Programmable Controllers for EMC . . . . 3 - 5
3.3.1 Routing of Cables . . . . 3 - 5
3.3.2 Equipotential Bonding . . . . 3 - 7
3.3.3 Shielding of Cables and Lines . . . . 3 - 8

3.4 Further Notes on System Configuration and Installation . . . . 3 - 10

Figures

3-1. Setting the Coding Element . . . . 3 - 2
3-2. Hooking the Module onto the Bus Unit . . . . 3 - 3
3-3. Screwing the Module onto the Bus Unit . . . . 3 - 3
3-4. Dimensional Drawing of the CP 521 SI . . . . 3 - 4
3-5. Routing of Equipotential Bonding Conductor and Signal Line . . . . 3 - 7
3-6. Examples of Securing Shielding Lines with Cable Clamps . . . . 3 - 9

Tables

3-1. Possible Applications of the CP 521 SI . . . . 3 - 1
3-2. Rules for Laying of Line Combinations . . . . 3 - 5

3 Installation Guidelines

3.1 Assembling and Dismantling the CP 521 SI

Like other I/O modules of the S5-100 system, the CP 521 SI snaps onto a bus unit.
Please note the following when assembling and dismantling the CP 521 SI:

- The module must not be plugged in or removed under power.
- The memory submodule must not be inserted or removed under power.
- The connection to the peripheral device (subminiature D plug connector) must not be plugged in or removed until data transfer between the CP 521 SI and the peripheral device has been completed.
- The CP 521 SI may only be plugged into slots 0 to 7 (exceptions: Table 3-1).
- The maximum number of CP 521 SIs that can be plugged in depends on the output current from+9 V (external I/O power supply) and the available analog area of the CPU (Table 3-1).

Reminder:
The CP 521 SI is treated in the same way as a 4-channel analog module (Section 2.9). It processes 64 data bits (**=4 analog channels**) in the analog area. The module has a current consumption of **140 mA**.

Table 3-1. Possible Applications of the CP 521 SI

Application in PLC/CPU	Permiss. Slots	Number of CPs	The Number of CP 521 SIs that can be Plugged in is Restricted by
S5-90U PLC	0 to 5	2	the max. permiss. total current consumption from+9 V (30 mA in this case)
S5-95U PLC	0 to 7	4	the number of analog channels (16 in this case)
CPU 100	0 to 7	2	the number of analog channels (8 in this case)
CPU 102	0 to 7 *	4	the number of analog channels (16 in this case)
CPU 103	0 to 7 *	7	the max. permiss. current consumption from+9 V (1 A in this case)
ET 200U	0 to 7	4	the number of analog channels (16 in this case)

* Slot 7 cannot be used for the CPU 102, 6ES5 102-8MA01 and CPU 103, 6ES5 103-8MA01

Setting the coding element

The CP 521 SI has a white coding element on the back. The coding key of the CP 521 SI is fixed at number 6. The bus unit has a white rotatable piece for each slot: the coding element. If you want to snap the module onto the bus unit, you must turn the coding element to "6" also, so that the coding key can engage in the coding element.

▸ **Set the coding element of the bus unit to number 6 using a screwdriver.**

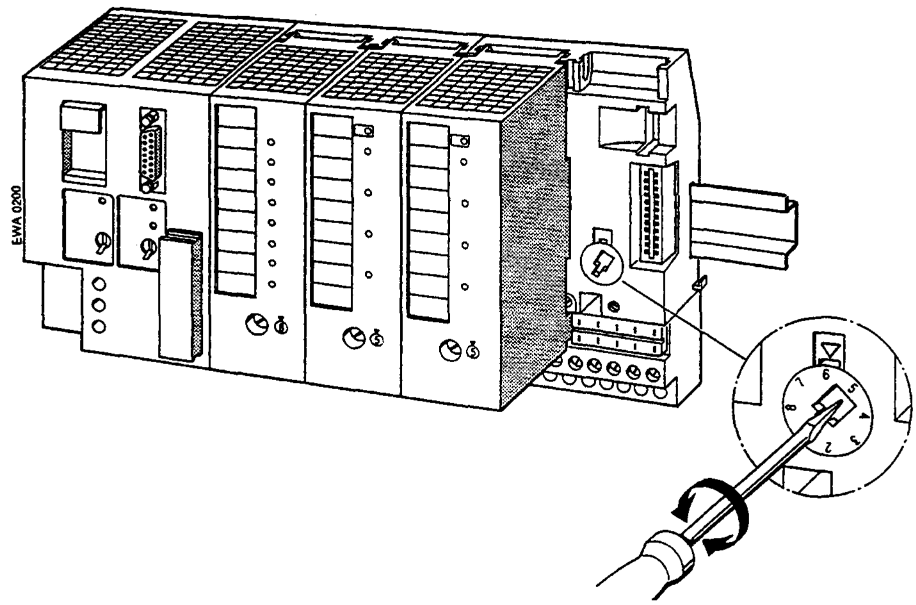


Figure 3-1. Setting the Coding Element

Assembling the CP 521 SI

The following two figures illustrate the remaining procedure:

1. **Hook the module onto the top of the bus unit and swing it down into the bus unit.**
2. **Press the module firmly into place and screw it onto the bus unit.**

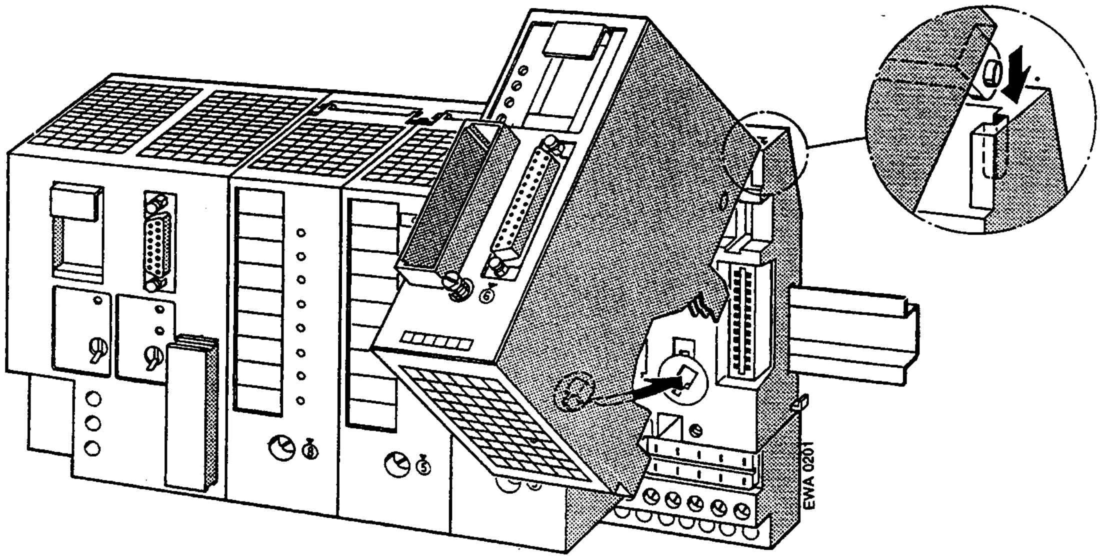


Figure 3-2. Hooking the Module onto the Bus Unit

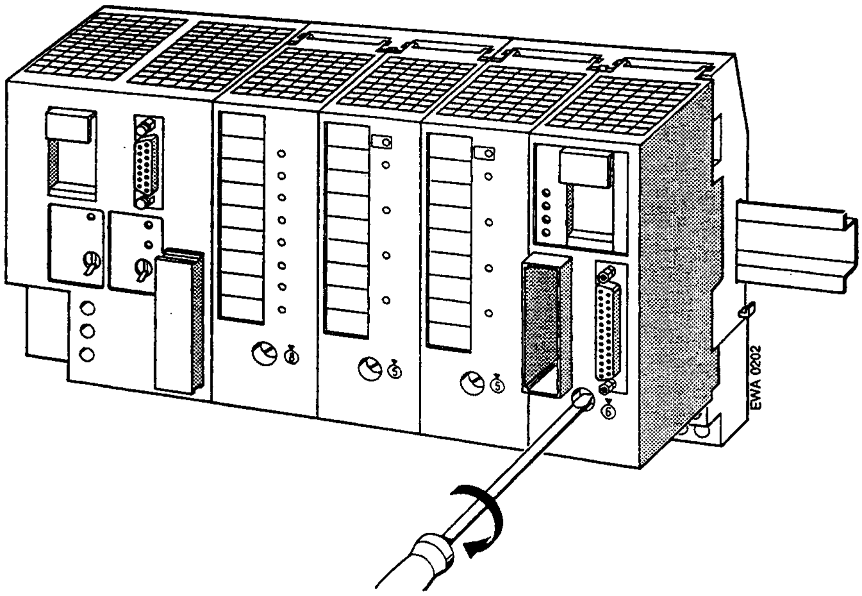


Figure 3-3. Screwing the Module onto the Bus Unit

Dismantling the CP 521 SI

- **Loosen the fixing screw and swing the module up and out of the bus unit.**

3.2 Dimensional Drawing

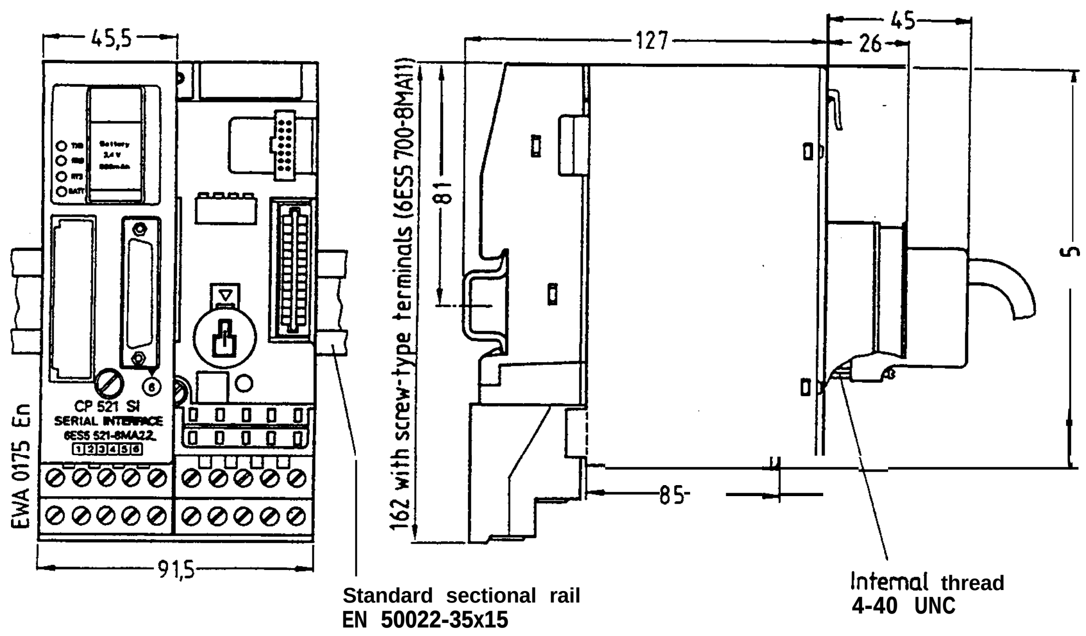


Figure 3-4. Dimensional Drawing of the CP 521 SI

3.3 Wiring of Programmable Controllers for EMC

The following section describes:

- Routing of cables within and outside cabinets
- Equipotential bonding between devices
- Connection of cable shields

3.3.1 Routing of Cables

This section covers the routing of bus, signal and supply lines. The purpose of cable routing is to suppress crosstalk between cables laid in parallel.

Routing of Cables Within and Outside Cabinets

For electromagnetically compatible routing of cables and lines, it is advisable to subdivide the lines into the following line groups and lay the groups separately.

Group A: Shielded bus and data lines (for programmer, OP, SINEC L1, SINEC L2, printer, etc.)
Shielded analog lines
Unshielded lines for DC voltage 60 V
Unshielded lines for AC voltage 25 V

Group B: Unshielded lines for DC voltage >60 V and 400 V
Unshielded lines for AC voltage >25 V and 400 V

Group C: Unshielded lines for DC and AC voltage>400 V

Group D: Lines for SINEC H1

From the combination of individual groups in the following table, you can read off the conditions for laying the line groups.

Table 3-2 Rules for Laying of Line Combinations

	Group A	Group B	Group C	Group D
Group A				
Group B				
Group C				
Group D				

Legend for the table:

Lines can be laid in common bundles or cable ducts.
Lines must be laid in separate bundles or cable ducts (without minimum clearance).
Lines within cabinets must be laid in separate bundles or cable ducts; outside the cabinets but within buildings, they must be laid over separate cable routes with a clearance of at least 10 cm (3.93 in.).
Lines must be laid in separate bundles or cable ducts with a clearance of at least 50 cm (1.64 ft.).

Routing of Cables Outside Buildings

Outside buildings, lay the lines on metal cable trays if possible. Provide the joints between cable trays with an electrical connection and ground the cable trays.

When laying lines outside buildings, you must observe the valid lightning protection and grounding measures. The following applies in general:

Lightning Protection

Where cables and lines for SIMATIC S5 controllers are to be laid outside buildings, you must apply measures for internal and external lightning protection.

Outside the buildings, lay your lines either

- in metal conduits grounded at both ends,
  or
- in concreted cable ducts with continuously connected reinforcement.

Protect the signal lines from overvoltages by means of

- varistors
  or
- inert gas-filled surge diverters.

Fit these protective devices at the cable entry into the building.

> **Note**
>
> Lightning protection measures always require an individual assessment of the entire installation. For clarification, please consult your local Siemens Office or a company specializing in lightning protection, such as Messrs. Dehn und Söhne in Neumarkt both Germany.

Equipotential Bonding

Ensure adequate equipotential bonding between the connected equipment (see section 3.3.2).

3.3.2 Equipotential Bonding

Between separate sections of an installation, potential differences can develop if

- programmable controllers and I/O devices are connected via non-floating links, or
- cable shields are connected at both ends and are grounded at different parts of the system.

Different AC supplies, for example, can cause potential differences. These differences must be reduced by installing equipotential bonding conductors to ensure functioning of the electronic components.

The following points must be observed for equipotential bonding:

- The lower the impedance of the equipotential bonding conductor, the greater is the effectiveness of equipotential bonding.
- Where shielded signal lines are laid between the relevant sections of the system and connected at both ends to the ground/protective conductor, the impedance of the additional equipotential bonding conductor must not exceed 10 % of the shield impedance.
- The cross-section of the equipotential bonding conductor must be rated for the maximum circulating current. The following cross-sections of copper have proved to be satisfactory in practice:
  - 16 mm^2 for equipotential bonding conductors of up to 200 m (656 ft.) in length
  - 25 mm^2 for equipotential bonding conductors of more than 200 m (656 ft.) in length.
- Use copper or zinc-plated steel for equipotential bonding conductors. They must be given a large-area connection to the ground/protective conductor and protect it from corrosion.
- The equipotential bonding conductor should be laid so that the smallest possible areas are enclosed between the equipotential bonding conductor and signal lines (see section 3.5).

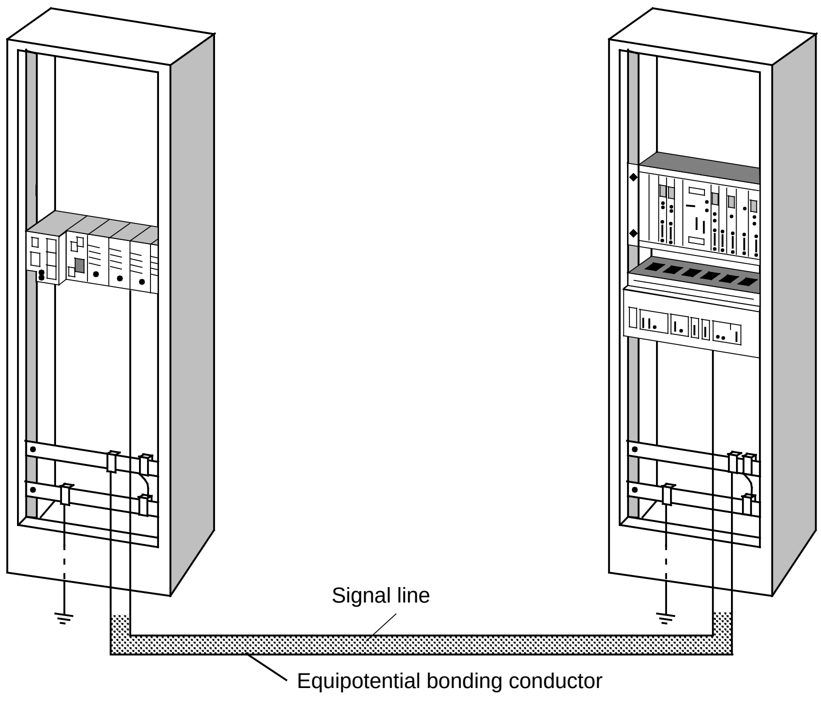


Figure 3-5. Routing of Equipotential Bonding Conductor and Signal Line

3.3.3 Shielding of Cables and Lines

Shielding is a method of attenuating magnetic, electrical or electromagnetic interference fields.

Interference currents on cable shields are passed to ground via the shield bar which is electrically connected to the housing. A low-impedance connection to the protective conductor is particularly important so that these interference currents themselves do not become an interference source.

Where possible, only use lines with a braided shield. The coverage density of the shield should be more than 80 %. Avoid lines with a foil shield because the foil can be very easily damaged by tensile strain and compression during fitting; this results in reduced effectiveness of the shield.

As a rule, line shields should always be connected at both ends. This is the only way to achieve a good degree of interference suppression in the higher frequency region.

Only in exceptional cases should the shield be connected at one end only, as this only achieves attenuation of the low frequencies. Single-ended shield connection may be more advantageous when:

- an equipotential bonding conductor cannot be laid;
- analog signals (of a few mV or mA) are to be transmitted;
- foil (static) shields are used.

With data lines for serial communication, always use metal or metallized connectors. Secure the shield of the data line to the connector case. Do **not** connect the shield to PIN 1 of the connector.

For stationary operation, it is advisable to fully strip the insulation from the shielded cable and connect it to the shield/protective conductor bar.

Note

In the event of potential differences between ground points, a circulating current may flow through the shield connected at both ends. In this case, install an additional equipotential bonding conductor (see section 3.3.2).

Please observe the following points when connecting the shield:

- Use metal cable clamps for securing the braided shield. The clamps must enclose the shield over a large area and provide a good contact (see Figure 3-6).
- Connect the shield to a shield bar immediately after the cable entry into the cabinet. Route the shield as far as the module but do not connect it there again.

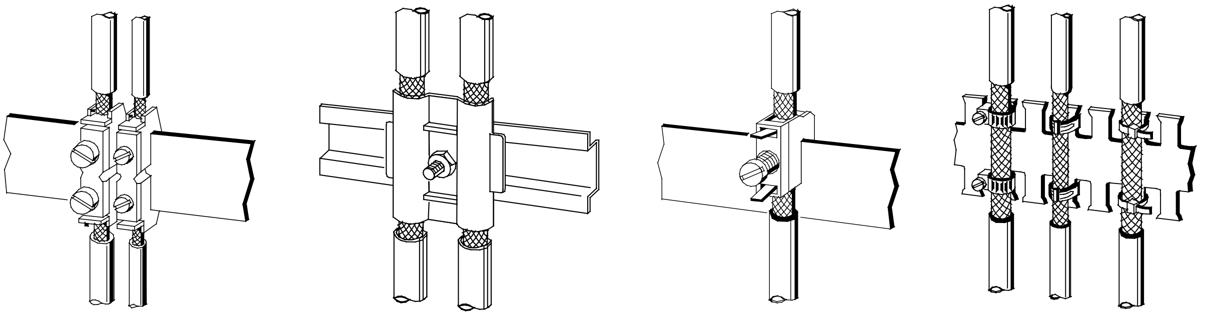


Figure 3-6. Examples of Securing Shielded Lines with Cable Clamps

3.4 Further Notes on System Configuration and Installation

Since the module is normally used as a component part of a larger system or plant, these notes are aimed at the hazard-free integration of the product in its environment.

The following are notes to be observed for the installation and startup of the product:

Warning

- Strictly follow the safety and accident prevention rules that apply in each particular case.
- In the case of equipment with a permanent power connection not provided with an isolating switch and/or fuses which disconnect all poles, a suitable isolating switch or fuses must be provided in the building wiring system (distribution board). Furthermore, the equipment must be connected to a protective ground (PE) conductor.
- In the case of equipment operated on the system voltage, make sure that the rated voltage range set coincides with the local system voltage before taking the equipment into operation.
- In the case of equipment operating on 24 V DC, make sure that the proper electrical isolation is provided between the mains supply and the 24 V supply. Use only power supply units to DIN VDE 0551 / EN 60742 und DIN VDE 0160.
- Fluctuations or deviations of the power supply voltage from the rated value should not exceed the tolerances specified in the technical specifications, otherwise function failures or dangerous conditions can occur in the electronic modules/equipment.
- Suitable measures must be taken to make sure that programs that are interrupted by a voltage dip or power supply failure resume proper operation when the power supply is restored. Care must be taken to ensure that dangerous operating conditions do not occur even momentarily. If necessary, the equipment must be forced into the ”emergency off” state.
- Emergency tripping devices in accordance with EN 60204/IEC 204 (VDE 0113) must be effective in all operating modes of the automation equipment. Resetting the emergency off device must not result in uncontrolled or undefined restart of the equipment.
- Connecting cables and signal cables must be installed in such a way that inductive and capacitive interference has no adverse effects on the automation functions.
- Automation equipment and operator controls must be installed in such a way that they are adequately protected against unintentional operation.
- To prevent wire breaks on the signal side leading to undefined states in the automation equipment, the relevant hardware and software precautions must be taken in the case of I/O connections.

4 Principle of Operation

4.1 General Principle of Operation . . . . . 4 - 1
4.1.1 CPU CP 521 SI . . . . . 4 - 2
4.1.2 CP 521 SI Peripheral Device . . . . . 4 - 3
4.1.3 Module RAM . . . . . 4 - 3

4.2 Access to the Transfer Memory . . . . . 4 - 4

4.3 Data Transmission Format . . . . . 4 - 6

4.4 Status Byte, Status of the Peripheral Device and Current Clock Data . . . . . 4 - 8
4.4.1 Status Byte . . . . . 4 - 8
4.4.2 Status of the Peripheral Device . . . . . 4 - 11
4.4.3 Current Clock Data . . . . . 4 - 12

4.5 Restart Characteristics . . . . . 4 - 16
4.5.1 Checking the Functional Capability of the Module . . . . . 4 - 16
4.5.2 Checking the Battery . . . . . 4 - 16
4.5.3 Memory Submodule Evaluation . . . . . 4 - 17
4.5.4 Clock Test . . . . . 4 - 17

4.6 Behaviour during Operation . . . . . 4 - 18
4.6.1 Printer Output . . . . . 4 - 18
4.6.2 Bidirectional Data Transmission . . . . . 4 - 19

Figures

4-1. Data Exchange Between CPU - CP 521 SI - Peripheral Device . . . . 4 - 1
4-2. Job Request Transfer and Response . . . . 4 - 2
4-3. Contents of the Transfer Memory . . . . 4 - 5
4-4. 10-Bit Character Frame . . . . 4 - 6
4-5. 11-Bit Character Frame . . . . 4 - 7
4-6. Contents of the Transfer Memory - Status Information and Current Clock Data . . . . 4 - 8

Tables

4-1. Error/Fault Numbers in Byte 0 (Status Byte) . . . . 4 - 9
4-2. Status of the Peripheral Device (Byte 1) . . . . 4 - 11
4-3. Values of the Integral Real-Time Clock . . . . 4 - 12
4-4. Module Faults . . . . 4 - 16
4-5. Errors/Faults in the Memory Submodule . . . . 4 - 17
4-6. Messages during Clock Test . . . . 4 - 17
4-7. Operating Faults (Printer Output) . . . . 4 - 18
4-8. Operating Faults (Bidirectional Data Transmission) . . . . 4 - 19

4 Principle of Operation

4.1 General Principle of Operation

Reminder:

- The CP 521 SI is has three interface ports.
  - Interface to the S5-100U bus
  - Receptacle for memory submodule
  - Serial interface for data interchange with the peripheral device
- In the process output image (PIQ) and the process input image (PII), eight bytes have been reserved per slot for the CP 521 SI. Input data and output data are referenced over the same address area.

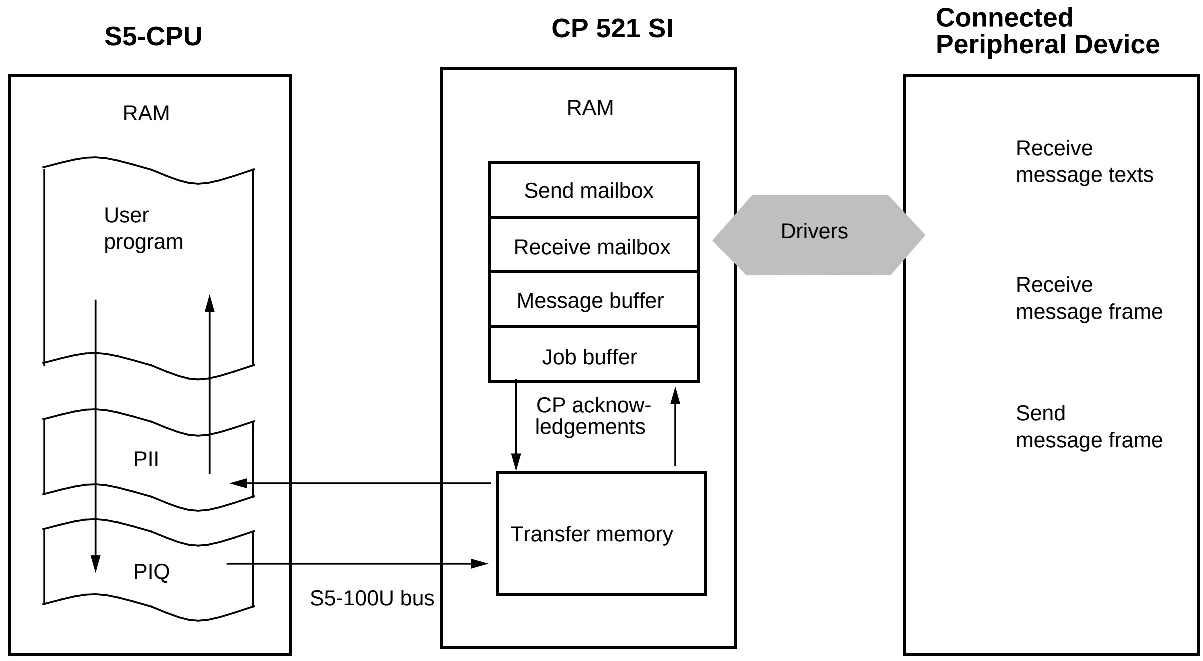


Figure 4-1. Data Exchange Between CPU - CP 521 SI - Peripheral Device

Data transfer between the CPU and the peripheral device is handled in two steps:

CPU CP 521 SI
Data interchange between the CPU and the CP 521 SI over the S5-100U bus must always be initiated by the CPU by transferring a job request.

CP 521 SI peripheral device
The CP 521 SI handles data transfer with the peripheral device autonomously. Programmable drivers are used for communications between the CP 521 SI and the peripheral device.

4.1.1 CPU CP 521 SI

Data is exchanged between the CPU and the CP 521 SI in eight-byte message blocks. The eight-byte blocks are transferred over the S5-100U bus in a **data cycle**.

Reminder:
A CPU cycle (OB1: see also S5-100U PLC Manual) consists of two different time processes.

- **Program cycle (PCyc):**
  The STEP 5 operations of the user program are executed.
  During this period, the S5-100U bus is inactive.
- **Data cycle (DCyc):**
  The data is transferred between the CPU and the CP 521 SI over the S5-100U bus.

Program cycles and data cycle alternate constantly.

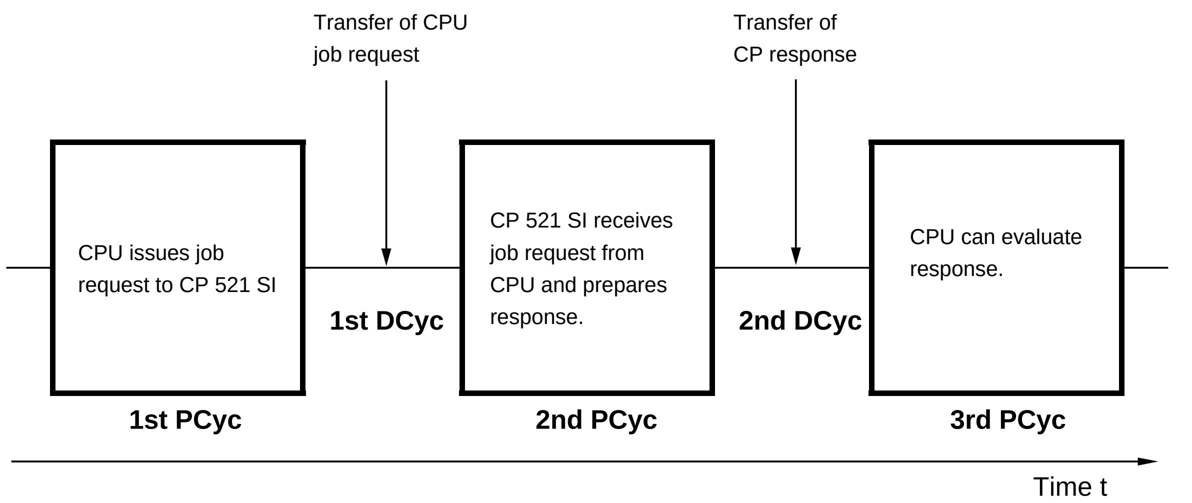


Figure 4-2. Job Request Transfer and Response

The cycle schematic shows that the CPU can scan the response to a job request from the CP 521 SI at the earliest two program cycles later. This fact is significant for the user program:

> **Note**
>
> When evaluating data from the CP 521 SI (PII), you must allow for the fact that you receive the response to a CPU job request (PIQ) at the earliest two program cycles after the job request is issued. This means that, for example, you can only evaluate the error message $4X_H$ ”Illegal job request” two program cycles after the job request has been issued.

There is an eight-byte transfer memory in the CP 521 SI for the organization of data transfers between the CPU and the CP 521 SI. The principle of data exchange between the CPU and the CP 521 SI over the transfer memory is described in Section 4.2.

4.1.2 CP 521 SI Peripheral Device

Data transfer with the peripheral device is handled autonomously by the CP 521 SI over the serial interface. The CP 521 SI is equipped with seven different drivers for this purpose.

- **Unidirectional data traffic**
  - Printer driver

- **Bidirectional data traffic**
  - ASCII driver, transparent
  - ASCII driver, interpretive mode I and II
  - 3964(R) procedure
  - SINEC L1 driver, slave
  - SINEC L1 driver, master (point-to-point)
  - Terminal driver

Depending on the parameters assigned, the CP 521 SI assumes that a peripheral device with an RS-232C (V.24) voltage interface or a TTY current interface is connected to the serial interface. The parameters of the serial interface have default settings. They can be reset within their value ranges depending on the active driver and the desired transmission method (XON/XOFF protocol, handshake mode   Section 6).

Data transfer over the serial interface is handled by the CP 521 SI according to the selected mode. For this reason, we have reserved a chapter in the manual for each of the individual drivers.

You will find the following details in each of these chapters:

- Procedure for assigning module parameters and the possible value ranges of the parameters for this mode
- The special job request and acknowledgement mechanism between the CP 521 SI and the peripheral device
- Conversion of job request handling into a STEP 5 program.

Note

Full duplex transmission is always possible between the CP 521 SI and the peripheral device. This means that the CP 521 SI can send data to the peripheral device and simultaneously receive data from the peripheral device.

4.1.3 Module RAM

The following memory areas have been set up in the module RAM for CPU job requests:

- **Job buffer**
  CPU job requests which are too complex to be processed immediately by the CP 521 SI are stored in a buffer (job buffer). Up to 30 job requests can be buffered. When the CP 521 SI has processed the job request, the latter is deleted in the job buffer.

- **Message buffer**
  The message buffer is only relevant for outputting message texts. All print job requests from the CPU are copied from the job buffer to the message buffer with date and time of day. The message buffer can hold up to 255 job requests. When the CP 521 SI has processed the print job request, it is deleted in the message buffer.

- **Send mailbox**
  The eight-byte message blocks of a message frame sent by the CPU are buffered in the send mailbox. Only when the CP 521 SI has received all message blocks of the message frame does it send the entire message frame autonomously to the peripheral device. The send mailbox holds 256 bytes and can only accommodate one message frame.

- **Receive mailbox**
  The CP 521 SI stores the data received from the peripheral device in its receive mailbox. The received data is coded into message frames and transferred further to the CPU in eight-byte message blocks. The receive mailbox holds 1 KB, and can accommodate up to 99 message frames.

4.2 Access to the Transfer Memory

The CP 521 SI has an eight-byte transfer memory for data transmission over the S5-100U bus. The CPU can write data to and read data from the transfer memory at any time:

The CPU issues a job request to the CP 521 SI in word 0, and in words 2 to 6 it can transfer further information necessary for executing the job request.

The CPU (user program) transfers the following to the transfer memory over the PIQ:

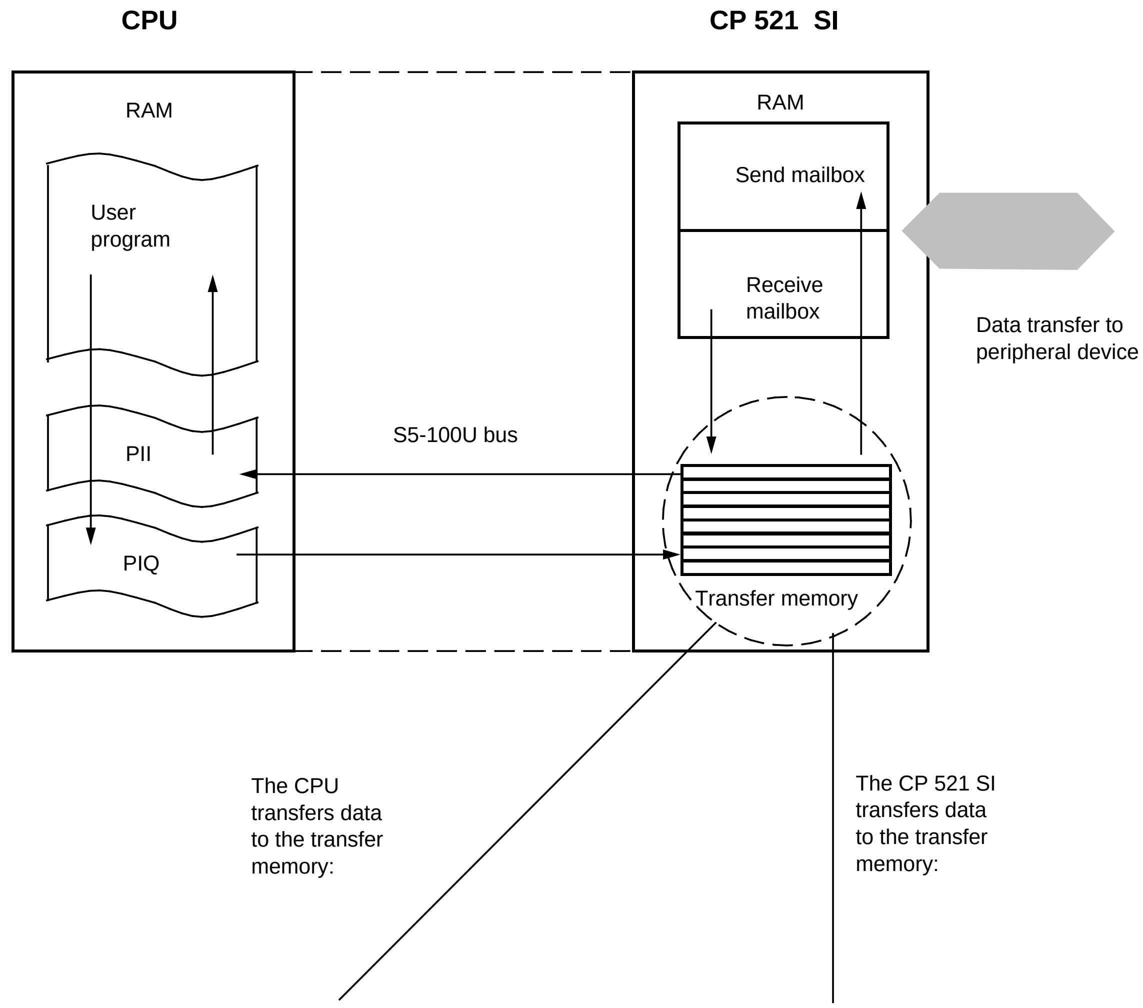
- Word 0: the job request, e.g. ”Send message block no. 1”
- Word 2, 4, 6: further necessary or possible information on the job request,
  e.g.: Data of message block no. 1

The CP 521 SI accepts the data from the transfer memory, evaluates it and writes current data into the transfer memory.

- Word 0: job request acknowledgement, e.g.: acknowledgement of job request ”Send message block no. 1”
- Word 2, 4, 6: further information on the acknowledgement,
  e.g.: ”Data valid”, number of the 1st message block, number of message frames in the receive buffer

The CPU (user program) can access the current data in the transfer memory over the PIQ.

> **Note**
>
> A CPU job request will only be processed by the CP 521 SI if it is different from the previous job request in byte 0 or byte 1 (edge evaluation). A job request will continue to be sent to the CP 521 SI until a new job request is initiated.



Byte	Contents
Byte 0	CPU job request
Byte 1	
Byte 2	Further necessary or possible information on the CPU job request
Byte 3	
Byte 4	
Byte 5	
Byte 6	
Byte 7	

Contents
CP acknowledgement
Further information on the acknowledgement

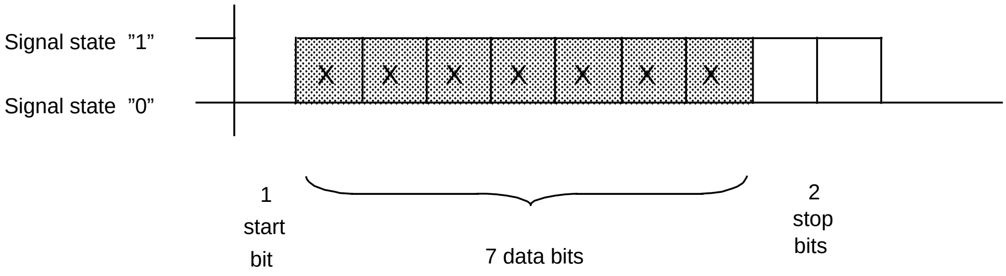
Figure 4-3. Contents of the Transfer Memory

4.3 Data Transmission Format

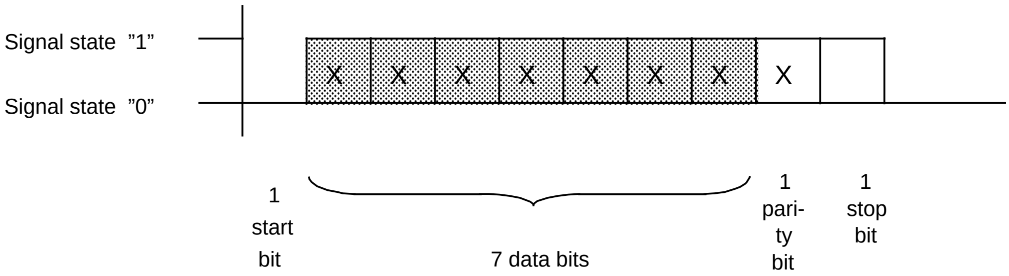
Data is transferred between the CP 521 SI and the peripheral device over the serial interface in a 10-bit or 11-bit character frame. There are three formats for each character frame. You can assign the desired data transmission format parameters in parameter block 0.

10-bit character frame:

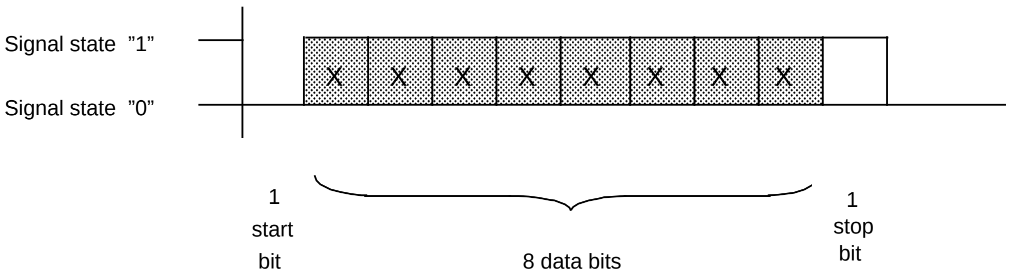
- **7 data bits:** 1 start bit, 7 data bits, 2 stop bits
  (parameter block 0: data format ”3”)



- **7 data bits:** 1 start bit, 7 data bits, 1 parity bit, 1 stop bit
  (parameter block 0: data format ”4”)



- **8 data bits:** 1 start bit, 8 data bits, 1 stop bit
  (parameter block 0: data format ” 5”)

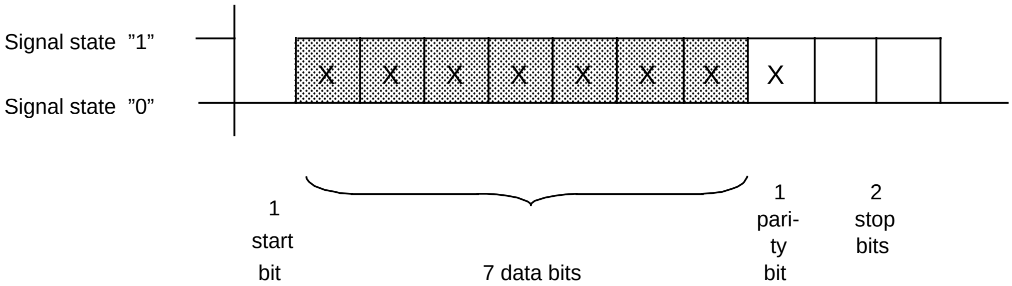


X=can assume signal state 0 or 1

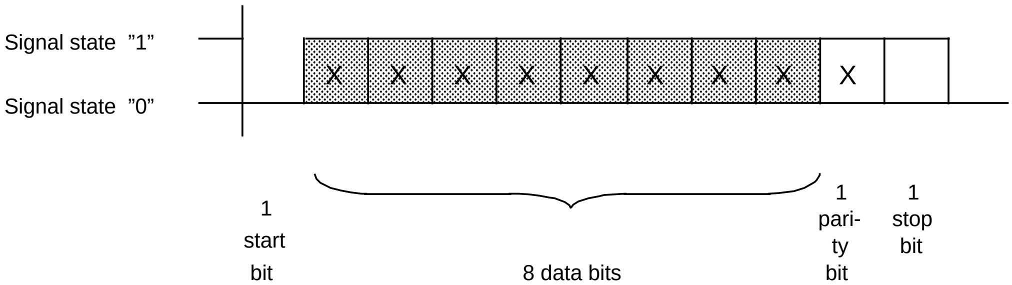
Figure 4-4. 10-Bit Character Frame

11-bit character frame:

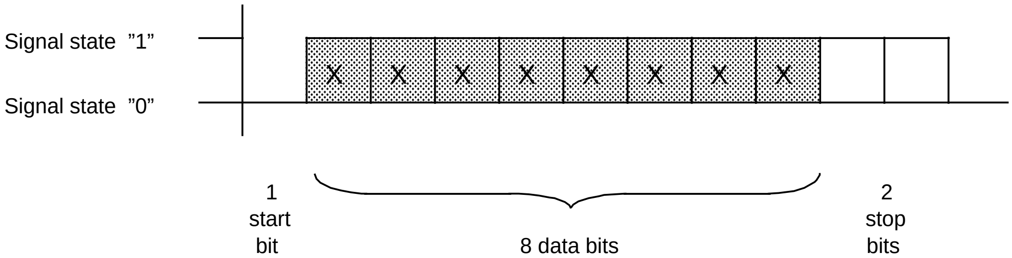
- **7 data bits:** 1 start bit, 7 data bits, 1 parity bit, 2 stop bits (parameter block 0: data format ”0”)



- **8 data bits:** 1 start bit, 8 data bits, 1 parity bit, 1 stop bit (parameter block 0: data format ”1”)



- **8 data bits:** 1 start bit, 8 data bits, 2 stop bits (parameter block 0: data format ”2”)



X=can assume signal state 0 or 1

Figure 4-5. 11-Bit Character Frame

4.4 Status Byte, Status of the Peripheral Device and Current Clock Data

If **no** message blocks have yet been transferred between the CPU and the CP 521 SI, and if no parameter assignment, print or ”Set clock” job is active, the CP 521 SI writes the following into the transfer memory:

Byte	Contents	
Byte 0	Status Byte	
Byte 1	Status PD	Weekday*
Byte 2	Day	
Byte 3	Month	
Byte 4	Year	
Byte 5	Hours	
Byte 6	Minutes	
Byte 7	Seconds	

Current clock data in BCD format (Bytes 2 to 7)

***** The weekday is calculated automatically from the clock data.

Figure 4-6. Contents of the Transfer Memory - Status Information and Current Clock Data

The following CPU job requests are followed by the same replies:

- 0000_H: ”Blank job request”

The status byte contains information on the following:

- Errors/faults occurring in data interchange between the CP 521 SI and the peripheral device
- The status of a CPU job request
- The status of the CP 521 SI.

4.4.1 Status Byte

The CP 521 SI provides the CPU with error/fault numbers in byte 0 of the transfer memory. The error/fault numbers can be read into the user program with load operations and evaluated there.

The status byte is divided into two nibbles. The nibbles are independent of each other as far as information is concerned. They can be combined in any way (e.g. 27_H: Default clock time set/no battery backup). This enables detailed error/fault evaluation.

Table 4-1. Error /Fault Numbers in Byte 0 (Status Byte)

Bits 4 to 7	Bits 0 to 3	Status	Explanation
0	0	No error	Job buffer empty
X	1	Memory submodule error	Error in memory submodule configuration
X	2	No texts	No message texts have been configured on the memory submodule
X	7	No battery backup	- Battery has not been inserted or - Battery defective
X	8	Message buffer overflow	The module cannot handle further print job requests at present. The job request must be repeated.
X	9*	Character delay exceeded	The time between two received characters is greater than the value programmed in parameter block 7.
X	A*	Parity error	The parity of the characters received does not agree with the parity programmed in parameter block 0.
X	B*	Receive after XOFF or receive after DTR="OFF"	The CP 521 SI sends XOFF and/or DTR="OFF" to the peripheral device when fewer than 20 bytes are free in the receive mailbox, or the CP 521 SI has received 99 message frames.
X	C*	Frame longer than 256 bytes	The peripheral device has sent a frame that is more than 256 bytes long.
X	E*	Receive mailbox overflow	The receive mailbox (1024 bytes) of the CP 521 SI cannot take any more frames because: - there is not enough memory space free or - 99 frames have already been stored.
X	F	CP 521 SI in restart or job buffer full	Message occurs only during restart: the clock data is invalid and the module cannot accept job requests.
1	X	Clock defective	Replace module
2	X	Default clock time set	The clock is set with the values Sunday, 1.1.90, 12:00:00.
3	X	Clock time/date error	At least one setting is outside the permissible range. The clock has not accepted the new clock data and continues with the current data.

X = Signal state not significant for the other nibble
* These error messages are output only with the ASCII driver when a frame is fetched (terminating frame $5X_H$, Section 6.5).

Table 4-1. Error/Fault Numbers in Byte 0 (Status Byte) (Cont.)

Bits 4 to 7	Bits 0 to 3	Status	Explanation
4	X	Illegal job request	You have issued the CP 521 SI with a job request which is not permissible in the relevant mode. This error message also appears if you transfer illegal parameters in a parameter assignment job request.
8	X	Hardware fault	Replace module

X=Signal state not significant for the other nibble

Example: Scanning the status of the module

An unconfigured memory submodule has been plugged into the module. Output 4.0 is to be set when the error has been detected.

STL FB100	Explanation
`:L IW 120`	`Module starting address`
`:L KH 0F00`	`Mask byte 0`
`:AW`	
`:L KH 0100`	
`:!=F`	`If byte 0 has the value 1,`
`:= Q 4.0`	`output 4.0 is set`
`:BE`	

4.4.2 Status of the Peripheral Device

In order to be able to evaluate the status of the peripheral device (e.g. printer), the following conditions must be met:

- Printer driver must be set
- TTY or RS-232C (V.24) terminal diagram according to Figure 5-2 or 5-3 (Section 5.2). This enables monitoring of the peripheral device.
- Busy signal must be programmed (parameter block 0).

When these conditions have been met, the error message ”Peripheral device not ready” will appear 20 s after the peripheral device fails (bit 4 in byte 1 is set).

Table 4-2. Status of the Peripheral Device (Byte 1)

Bit 4 to 7	Bit 0 to 3	Status
0	X	Peripheral device ready
1	X	Peripheral device not ready

X=Signal state not significant for the other nibble

The signal states of bits 0 to 3 (right nibble) of byte 1 specify the relevant weekday (Section 4.4.3).
Both nibbles are independent of each other as far as information is concerned. They can be combined in any way (e.g. 13_H: Peripheral device not ready/Tuesday).

Example: Scanning the status of the peripheral device

The module is located in slot 7 (starting address 120).
If the peripheral device is not ready, output 4.1 is to be set.

STL FB101	Explanation
:L IW 120	Module starting address
:L KH 00F0	Mask byte 1
:AW	
:L KH 0010	
:!=F	If byte 1 has the value 16
:= Q 4.1	(peripheral device not ready),
:BE	output 4.1 is set

4.4.3 Current Clock Data

The clock data is provided by the integral real-time clock and updated once per second. After module restart, the clock is set with the default value ”01.01.90 00.00.00”.

Setting the clock

Set the clock with the CPU job request **1000_H**. The following values are permissible.

Table 4-3. Values of the Integral Real-Time Clock

Byte	Meaning	Possible Values
0	Status byte	Table 4-1
1	Bits 4 to 7: Status of the peripheral device	Table 4-2
	Bits 0 to 3: Current weekday 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday	$X1_{BCD}$ to $X7_{BCD}$
2	Current day	01_{BCD} to 31_{BCD}
3	Current month	01_{BCD} to 12_{BCD}
4	Current year	00_{BCD} to 99_{BCD}
5	Current hour 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	Current minute	00_{BCD} to 59_{BCD}
7	Current second	00_{BCD} to 59_{BCD}

The settings must be entered in BCD format. You need not enter the weekday. It is calculated automatically from the date set. Enter FF_H in the relevant byte if you do not want to change the default clock setting.

If a setting is outside the permissible range, the clock will not be set. The CP 521 SI sends error message $3X_H$ ”Clock time/date error”.

Note

The CP 521 SI is supplied with the default values for the date and clock time parameters (Table 5-8). The default value for representation of the ”Current hour” is the 24-hr mode. If you want to operate the clock with the 12h mode, you must change this time parameter before setting the clock (Section 5.3).

Write CPU "Set clock" job to the PIQ:

Request from CPU to CP: Set clock

Address	Contents
Maddr+0	10_H
Maddr+1	00_H
Maddr+ 2	Day
Maddr+ 3	Month
Maddr+ 4	Year
Maddr+ 5	Hours
Maddr+ 6	Minutes
Maddr+7	Seconds

Code for "Set clock"

Evaluate CP response to the "Set clock" job:

CPU job accepted
Clock now set

Address	Contents
Maddr+0	50_H
Maddr+1	00_H
Maddr+ 2	irrelevant
Maddr+ 3	irrelevant
Maddr+ 4	irrelevant
Maddr+ 5	irrelevant
Maddr+ 6	10_H
Maddr+7	00_H

Acknow-ledged job

CP rejects job CPU request
Error in clock data/
job buffer full/wrong job

Contents
$4x_H$
xx_H
irrelevant
irrelevant
irrelevant
irrelevant
10_H
00_H

x: Status information of the CP 521 SI
(Section 4.4.1/4.4.2)

Note

The CPU acknowledgements 5000_H and $4xxx_H$ remain (i.e. are not deleted) until they are overwritten by a subsequent job. After the 0000_H blank job has been sent, the CP response again contains the current data of the integral real-time clock

Example: Setting the clock (job request 1000_H)

The module is located in slot 7 (starting address 120).

The clock is to be set to ”28.08.91 09.40.00”.

STL FB10	Explanation

`Name : CLOCK`
`: L KH 1000`
`: T QW 120`
`: L KH 2808`
`: T QW 122`
`: L KH 9109`
`: T QW 124`
`: L KH 4000`
`: T QW 126`
	CPU job request 1000_H ”Set clock”
	Write the values for the clock data into the PIQ

Example: Reading the current clock data

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

STL FB10	Explanation

`NAME : CLOCK DATA`
`: L IB 121`
`: T QB 8`
`: L IW 122`
`: T QW 9`
`: L IW 124`
`: T QW 11`
`: L IW 126`
`: T QW 13`
`: BE`
	The current clock data is read and transferred to the digital output modules

Use of the clock data

You can use the clock data in two ways:

- The current clock data can be inserted into message texts. For this purpose, you can enter place holders for date and time of day when configuring the message texts (Section 5.5.2).
- The current clock data can be read by the CPU and used in the user program.

Correction value

You can configure a correction value to improve the accuracy of the clock. See Section 5.3.5 ”Clock Correction Factor” for more detailed information.

Note

By using the integral clock, you can evaluate the following additional messages:

- Module in restart routine or job buffer full (XF_H)
  The clock data is invalid.
- Default time set ($2X_H$)
  This evaluation is relevant if you are operating the CP 521 SI without a backup battery.
- Time or date error ($3X_H$)
  This error evaluation is relevant after you have set the clock.
- Clock defective ($1X_H$)
  This error evaluation is only relevant after restart.

4.5 Restart Characteristics

When the supply voltage is restored (POWER ON), the CP 521 SI performs a restart.

The restart consists of the following:

- The send and receive mailboxes are deleted
- The functional capability of the module is checked (Section 4.5.1)
- The battery function is checked (Section 4.5.2)
- The memory submodule is evaluated (Section 4.5.3)
- The operating system sets the desired mode
- The clock test is executed (Section 4.5.4).

If errors/faults occur during restart, the CP 521 SI transfers message to the CPU in byte 0 (status byte) (Section 4.4.1).

Note

The message buffer is not deleted during restart in print mode if the module has battery backup and the memory submodule has not been replaced.

4.5.1 Checking the Functional Capability of the Module

In this part of the restart procedure, the hardware is checked for functional capability. Faults are assigned an appropriate number in byte 0 (status byte) and you can then evaluate them in the user program or with a programmer.

Table 4-4. Module Faults

Error Number in Byte 0	Message	Remedy
$1X_H$	Clock defective	Replace module
$8X_H$	Hardware fault	Replace module
XF_H	CP 521 SI in restart routine	Scan status byte for XF_H before transferring the first job request

X: Other nibble can assume different values; not significant here.

4.5.2 Checking the Battery

When the module is switched off, the clock and the RAM are powered by the module battery. The battery is checked at every restart in order to detect a possible failure. In the event of a failure, data in the RAM is deleted. If the battery voltage drops below the value required for backup, the BATTERY LOW LED lights up and error "$X7_H$" is output in byte 0.

4.5.3 Memory Submodule Evaluation

The serial interface parameters and message text parameters specified in the memory submodule are checked here. Errors are provided with an appropriate error number in byte 0 (status byte) and can be evaluated in the user program.

Table 4-5. Errors/Faults in the Memory Submodule

Error Number in Byte 0	Message	Remedy
$X1_H$	Fault in memory submodule or memory submodule not plugged in	1. PLC POWER OFF 2. Plug in (new) memory submodule 3. PLC POWER ON
$X2_H$	No message texts configured	Configure message texts Wrong or defective memory submodule plugged in?

X: Other nibble can assume different values; not significant here.

During module restart, the serial interface is initialized with the relevant parameter. Data is searched for in this order:

1. Memory submodule
2. Module RAM.

If neither the memory submodule nor the RAM contain data, data provided by the system (default values) is transferred.

4.5.4 Clock Test

This part of the restart procedure checks the integral module clock. The clock is set if the backup battery fails during PLC POWER OFF. In the basic setting, the clock has the default value ”01.01.90 00.00.00”.

Errors/faults and the basic setting are provided with an appropriate error number in byte 0 (status byte), and can be evaluated in the user program or with a programmer.

Table 4-6. Messages during Clock Test

Error Number in Byte 0	Message	Remedy
$1X_H$	Clock defective	Replace module
$2X_H$	Default clock time set	Set clock with FB10 (Section 4.4.3)

X : Other nibble can assume different values; not significant here.

4.6 Behaviour during Operation

4.6.1 Printer Output

Various faults can occur during printer operation. The following table shows the effects.

Table 4-7. Operating Faults (Printer Output)

Fault	Remarks	Effects
CPU enters STOP mode		Activated print job requests* are completed.
POWER OFF (CPU)		Activated print job requests no longer completed.
	Battery backup available	Clock data and print job requests are retained.
	not available	Clock data and print job requests are lost.
25-pin sub D connector unplugged or cable fault	BUSY line available and BUSY signal configured	Activated print job requests are completed after restoration of the connection. Perfect print quality is not guaranteed (e.g. smudged characters).
	otherwise	Activated print job requests are not completed after restoration of the connection.
POWER OFF (printer)**	BUSY line available and BUSY signal configured	Activated print job requests are completed after restoration of the connection. Perfect print quality is not guaranteed (e.g. smudged characters).
	otherwise	Activated print job requests are not completed after restoration of the connection.
	XON/XOFF protocol configured	Activated print job requests are completed after restoration of the connection.
	not configured	Activated print job requests are not completed after restoration of the connection.

* Print job requests (Section 5.6)
** Characters of the internal printer buffer are lost

4.6.2 Bidirectional Data Transmission

Various faults can occur during bidirectional data transmission. The following table shows the effects.

Table 4-8. Operating Faults (Bidirectional Data Transmission)

Fault	Effects
CPU enters STOP mode *	Data continues to be sent and received between the CP 521 SI and the peripheral device. This can lead to overflow of the receive mailbox.
POWER OFF (CPU)	• Data of the send and receive message frame is lost • Battery backup available: Clock data is retained Battery backup not available: Clock data is lost
Fault in CP-peripheral device connection or POWER OFF (peripheral device)	Data corruption occurs during data interchange (both directions)** Error message of the CP 521 SI: • Character delay error after character timeout • Peripheral device not ready (after 20 s) Permanent wire break • Error(s) in message frame(s) in receive mailbox **

* An active job request between the CPU and the CP 521 SI (send or receive job request) is interrupted. The job request must be restarted after switching over from STOP to RUN.

** Data corruption cannot occur in the case of the 3964(R) procedure and in the case of SINEC L1 operation, since the message frames are transferred in a protocol frame (Chapters 8 and 9).

5 Printer Driver

5.1 Prerequisites for Using the Printer Driver . . . . 5 - 2

5.2 Peripheral Interface Connections . . . . 5 - 4

5.3 Assigning CP 521 SI Parameters in Print Mode . . . . 5 - 6
5.3.1 Assigning the Serial Interface Parameters (Parameter Blocks 0, 1 and 2) . . . . 5 - 9
5.3.2 Parameter Assignment Data for Entering Message Texts (Parameter Block 3) . . . . 5 - 12
5.3.3 Parameter Assignment Data for Message Text Printout (Parameter Blocks 4 to 6) . . . . 5 - 12
5.3.4 Configuring the Character Conversion Table (Parameter Block 8) . . . . 5 - 16
5.3.5 Clock Correction Factor (Parameter Block 9) . . . . 5 - 16
5.3.6 CP 521 SI Parameter Assignment Example . . . . 5 - 17

5.4 Printing Parameter Assignment Data . . . . 5 - 20

5.5 Configuring Message Texts . . . . 5 - 22
5.5.1 Entering Place Holders . . . . 5 - 23
5.5.2 Place Holders for Date and Time of Day . . . . 5 - 25
5.5.3 Place Holders for Message Texts . . . . 5 - 26
5.5.4 Place Holders for Variables . . . . 5 - 29
5.5.5 Control Parameters . . . . 5 - 32

5.6 Printing Message Texts . . . . 5 - 35
5.6.1 Setting the Page Number . . . . 5 - 37
5.6.2 Outputting Message Texts with and without CR/LF at the End . . . . 5 - 38
5.6.3 Executing a Page Feed . . . . 5 - 41
5.6.4 Outputting a Line Feed . . . . 5 - 42
5.6.5 Deleting the Message Buffer . . . . 5 - 42
5.6.6 Printing All Configured Message Texts . . . . 5 - 43

5.7 Status of the CP 521 SI in Print Mode . . . . 5 - 44

Figures

5-1. Terminal Diagram CP 521 SI (TTY Passive) - Printer DR 2xx-N (TTY Active) without BUSY Line . . . 5 - 4
5-2. Terminal Diagram CP 521 SI (TTY Passive) - Printer DR 2xx-N (TTY Active) with BUSY Line . . . 5 - 5
5-3. Terminal Diagram RS-232C (V.24) Interface (Printer Driver) . . . 5 - 5
5-4. Schematic for Entering Parameter Blocks in DB1 . . . 5 - 8
5-5. Schematic for Entering Headers and Footers . . . 5 - 15
5-6. Schematic for Configuring the Character Conversion Table . . . 5 - 16
5-7. Structure of a Message Text . . . 5 - 22
5-8. Schematic Structure of a Place Holder . . . 5 - 23
5-9. Data Interchange Over the CP 521 SI . . . 5 - 35
5-10. Schematic Flowchart of the ”Print Message Text” Job Request . . . 5 - 39

Tables

5-1. Transmission Methods in Print Mode . . . 5 - 1
5-2. Pin Assignments of the 25-Pin Subminiature D Connector of the CP 521 SI 5 - 4
5-3. Contents of the Parameter Blocks . . . 5 - 7
5-4. Parameter Assignment Data for the Serial Interface (Parameter Block 0) . . 5 - 9
5-5. Parameter Assignment Data for the Serial Interface (Parameter Block 1) . . 5 - 11
5-6. Parameter Assignment Data for the Serial Interface (Parameter Block 2) . . 5 - 11
5-7. Parameter Assignment Data for Entering Message Texts (Parameter Block 3) . . . 5 - 12
5-8. Parameter Assignment Data for Message Text Printout (Parameter Blocks 4 to 6) . . . 5 - 13
5-9. Page Format . . . 5 - 14
5-10. Correction Value for the Integral Clock (Parameter Block 9) . . . 5 - 16
5-11. Overview of Place Holders Configurable in Message Texts . . . 5 - 24
5-12. Data Formats for Variables . . . 5 - 29
5-13. ”KT” Data Format Printouts (Examples) . . . 5 - 30
5-14. ”KF” Data Format Printouts (Examples) . . . 5 - 30
5-15. Value Range for the Number of Places before and after the Decimal Point . . . 5 - 30
5-16. Examples of ”KFa,b” and ”KFa.b” Data Format Printouts . . . 5 - 31
5-17. Value Ranges for the Number of Places Before and After the Decimal Point . . . 5 - 32
5-18. Examples of ”KGa,b” and ”KGa.b” Data Format Printouts . . . 5 - 32
5-19. CPU Job Requests for Printing Message Texts (Byte 0 in the PIQ) . . . 5 - 37
5-20. PIQ in the Case of the ”Print Message Texts” Job Requests . . . 5 - 38
5-21. Status Byte in Printer Driver Mode (PII) . . . 5 - 44

5 Printer Driver

The printer driver of the CP 521 SI enables printout of message texts:

- Printout of message texts stored by you in data blocks (DBs) 2 to 63 on the memory submodule.
- Insertion of current time and date into the message text to be printed. The clock data is provided by the integral real-time clock.
- Insertion of current variables in the printout.
  The variable values can be transferred to the CP 521 SI by the CPU.

> **Note**
>
> Messages can only be printed out via the CP 521 SI if a memory submodule containing at least one message text is plugged in.

You can choose between three transmission methods in print mode. All transmission methods are possible with both the RS-232C (V.24) interface and the TTY interface.

Table 5-1. Transmission Methods in Print Mode

Transmission Method	Line Used TTY/ RS-232C (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 (car. return) - LF (line feed) - FF (form feed)	1	Line break not detectable

> **Note**
>
> If XON/XOFF is programmed, all characters on the RxD line except ”XON” are interpreted as ”XOFF”.

XON/XOFF protocol

The ”XON” and ”XOFF” characters are control characters used by the CP 521 SI for controlling data transmission. The CP 521 SI receives ”XOFF” from the peripheral device when the receive buffer in the peripheral device is full. The peripheral device then receives no more data from the CP 521 SI. Data would be lost in the case of an overflow of the receive mailbox. The CP 521 SI does not receive ”XON” from the peripheral device until there is space again in the receive buffer of the peripheral device.

BUSY signal

The CP 521 SI evaluates the ”BUSY” control signal of the printer. The printer signals its readiness to receive to the CP 521 SI.

- In the case of the TTY interface: current on RxD line
- In the case of RS-232C (V.24) interface: V - 3 V on DSR line

> **Note**
>
> If you have set ”none” for XON/XOFF protocol and ”none” for BUSY signal, you must program wait times.

5.1 Prerequisites for Using the Printer Driver

The following conditions must be met before activating the printer driver:

1. Establish connection

Establish the electrical connection between the CP 521 SI and the printer in the POWER OFF state (Section 5.2).

2. Settings on the printer

The settings on the printer must agree with your configured information on the memory submodule in the parameter blocks of DB1. If, for example, you set the baud rate at 1200 bit/s on the printer, you must also set the same baud rate in DB1.

> **Note**
>
> Assign your desired serial interface parameters on the printer and only then assign the parameters in DB1 on the memory submodule.

3. Setting the CP 521 SI parameters

Store all the parameter assignment data in DB1 on the memory submodule:

- Printer interface parameters (baud rate, type of interface, BUSY signal, ...)
- Configuration data for entering message texts (function characters, end-of-text characters)
- Configuration data for printout of message text (output form for date and time, headers and footers, character conversion table ...)
- Parameters for communications drivers

By configuring a correction factor, you can improve the accuracy of the module clock.

Memory submodule parameters are assigned using a programmer in off-line mode.

4. Configuring the message texts

You must configure at least one message text in one of the DBs from DB2 to DB63 on the memory submodule.

> **Note**
>
> When you have configured a message text in a DB, you can configure further message texts in other DBs later and store them additionally on the user submodule.
> The configuring of message texts is explained in Chapter 5.5.

You can configure up to 255 message texts.

5. Initializing the CP 521 SI

Plug the configured memory submodule into the CP 521 SI in the POWER OFF state. Then you can switch the CPU to POWER ON.

6. Starting up the CP 521 SI for print mode

If you use a memory submodule, the print driver is automatically activated in the module after POWER ON.

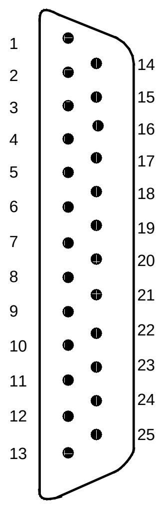
Without a memory submodule, the response of the CP 521 SI depends on whether the module has a backup battery or not:

- If there is no backup battery, the transparent ASCII driver is automatically activated following POWER-ON
- If a backup battery has been inserted, the CP 521 SI is automatically initialized with the data in the RAM on power-up (i.e. with the parameters last assigned). In other words, the module is activated in the same driver mode following POWER-ON as it was before POWER-OFF.

5.2 Peripheral Interface Connections

The CP 521 SI is equipped with a serial interface. You can choose between a current interface (TTY) or a voltage interface (RS-232C (V.24)) by changing the relevant parameters (Section 5.3). The cables of both interfaces connect with a 25-pin subminiature D connector.

Table 5-2. Pin Assignments of the 25-Pin Subminiature D Connector of the CP 521 SI

View	Pin No.	Signal Name	Meaning
	1	-	Disabled
	2	TxD	Send data (V.24)
	3	RxD	Receive data (V.24)
	4	RTS	Request to send (V.24)
	5	CTS	Clear to send (V.24)
	6	DSR	Data set ready (V.24)
	7	GND	Signal ground (RS-232C (V.24))
	8	-	Disabled
	9	TTY IN+	TTY receive line +
	10	TTY IN -	TTY receive line -
	11	-	Disabled
	12	-	Disabled
	13	P24	+24 V for active TTY
	14	-	Disabled
	15	-	Disabled
	16	-	Disabled
	17	20 mA	Current source TTY *****
	18	TTY OUT+	TTY send line +
	19	20 mA	Current source TTY *****
	20	DTR	Data terminal ready
	21	TTY OUT -	TTY send line -
	22	-	Disabled
	23	-	Disabled
	24	-	Disabled
	25	-	Disabled

***** If +24 V to GND (pin 7) on pin 13

The following figures show a few typical connections for print mode. In the ”Printer driver” mode, the CP 521 SI assumes that a printer with an RS-232C (V.24) interface or TTY interface for message text printout is connected to the serial interface.

CP 521 SI (TTY passive) - printer DR 2xx-N (TTY active) without BUSY line

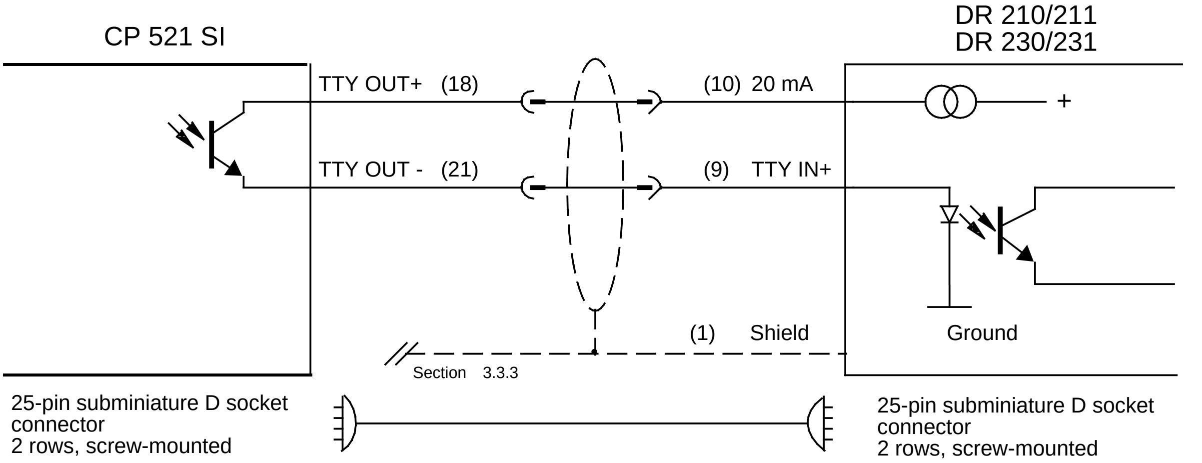


Figure 5-1. Terminal Diagram CP 521 SI (TTY Passive) - Printer DR 2xx-N (TTY Active) without BUSY Line

CP 521 SI (TTY passive) - Printer DR 2xx-N (TTY active) with BUSY line

Setting the printer: printer not ready to receive=no current

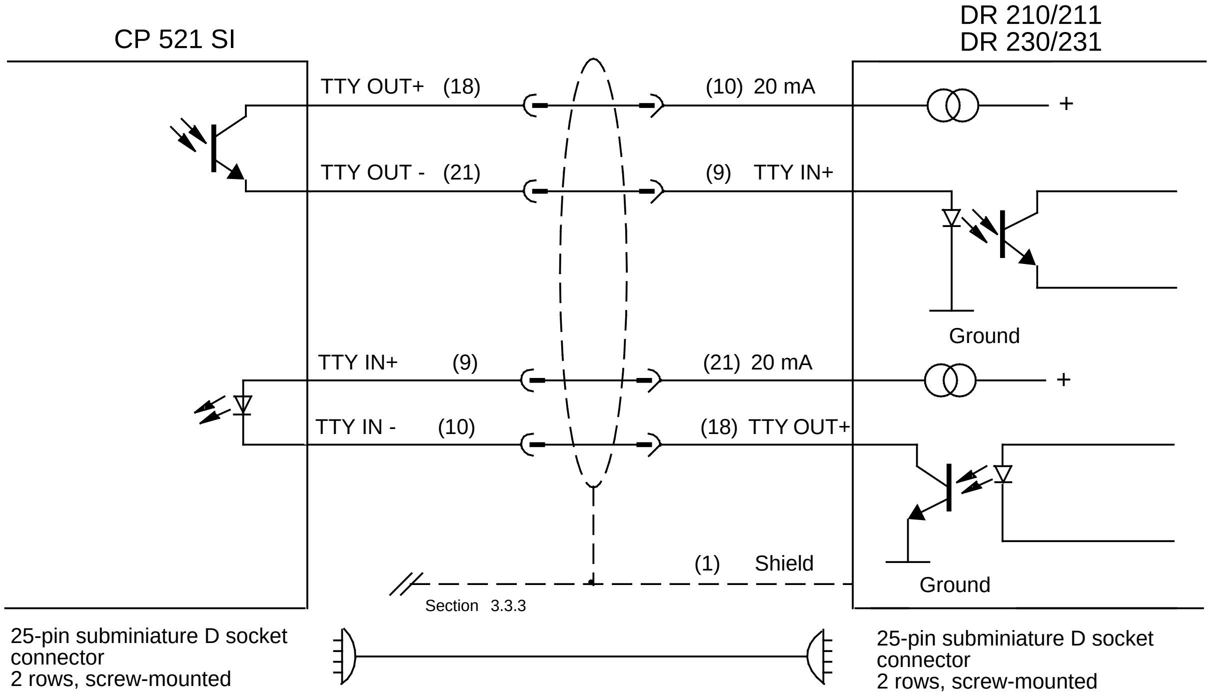
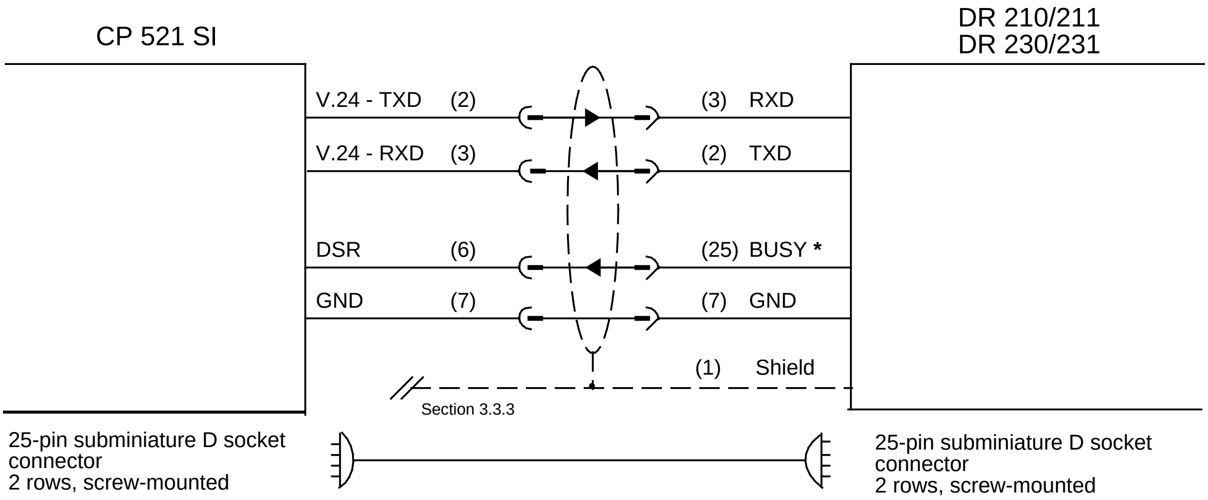


Figure 5-2. Terminal Diagram CP 521 SI (TTY Passive) - Printer DR 2xx-N (TTY Active) with BUSY Line

RS-232C (V.24) interface

Setting the printer: printer not ready to receive=negative potential



* To establish the connection shown, the Ready (Busy) signal on the DR 2xx printer must be connected to pin 25. You can do this by means of solder jumpers on the interface card of the printer (see also the manual of the DR 2xx printer).

Figure 5-3. Terminal Diagram RS-232C (V.24) Interface (Printer Driver)

5.3 Assigning CP 521 SI Parameters in Print Mode

The printer driver of the module can only be operated when a memory submodule with the necessary parameter settings has been plugged in.

Store the following on the memory submodule:

- The parameter assignment data of the CP 521 SI in data block (DB) 1
- The message texts in any of the DBs from DB2 to DB63 (Section 5.5)

The following settings must be made:

- Default for the serial interfaces
- Definition of function characters and end-of-text characters in message texts
- Definition of control characters for formatting message texts

You can also configure special characters for printing. The accuracy of the module clock can be improved by specifying a correction value.

The CP 521 SI is supplied with default values for all parameters. If you want to use other values, you must store your parameter assignment data on a memory submodule in DB1. The parameter assignment data is divided into 10 parameter blocks (parameter blocks 0 to 9) to make it as easy as possible to enter.

> **Note**
>
> Only those parameter blocks which deviate from the default values need be configured in DB1. However, all parameters in these blocks must then be assigned even if some parameters within the block correspond to the default values.

> **Note**
>
> You can print out all module parameters in the printer driver mode using the CPU job request **F0**$_H$. For those parameter blocks where parameters have not been assigned, the module default values are printed out.

Table 5-3. Contents of the Parameter Blocks

Parameter Block Number	Contents
	Assigning 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 assignment data for entering message texts:
3	Special characters for message texts
	Parameter assignment 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	Printer driver
8	Character conversion table
9	Clock correction factor

First, you must enter all the parameter blocks you wish to modify in DB1, using the programmer (PG). Then transfer DB1 to the memory submodule.

Note

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

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

The parameter blocks can be entered in DB1 according to the following schematic:

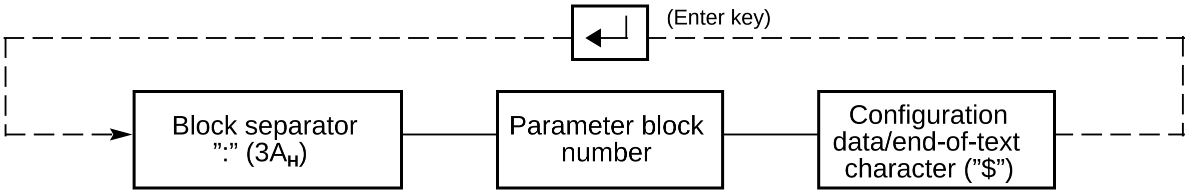


Figure 5-4. Schematic for Entering Parameter Blocks in DB1

Note

Parameters for parameter blocks 0 and 2 can be assigned in the user program. Parameters assigned in the user program have priority over those assigned in the memory submodule. The parameter assignment data in the memory submodule is valid after POWER ON as long as it is not overwritten by the user program.

Note

A CPU job request will only be processed by the CP 521 SI if it is different from the previous job request in byte 0 or byte 1 (edge evaluation). A job request will continue to be sent to the CP 521 SI until a new job request is initiated.

5.3.1 Assigning the Serial Interface Parameters (Parameter Blocks 0, 1 and 2)

Define the setting for the serial interface in parameter blocks 0, 1 and 2. Illegal entries are replaced by the relevant default values.

Table 5-4. Parameter Assignment Data for the Serial Interface (Parameter Block 0)

Block	Meaning		Value Range	Default Values on the CP 521 SI
0	Transmission rate	110 bit/s	1	
		200 bit/s	2	
		300 bit/s	3	
		600 bit/s	4	
		1200 bit/s	5	
		2400 bit/s	6	
		4800 bit/s	7	
		9600 bit/s	8	8 9600 bit/s
	Parity	even	0	0 even
		odd	1	
	(Parity bit always ”1”)	”mark”	2	
	(Parity bit always ”0”)	”space”	3	
		any	4	
	BUSY signal	no	0	0 no
		yes	1	
	Interface	TTY	0	0 TTY
		RS-232C (V.24)	1	
	Data format:			
	11-bit character frame:			
		7 data bits (with parity)	0	0 7 data bits (with parity)
		8 data bits (with parity)	1	
		8 data bits (without parity)	2	
	10-bit character frame:			
		7 data bits (without parity)	3	
		7 data bits (with parity)	4	
		8 data bits (without parity)	5	
	HW handshake	OFF*	0	0 OFF

* Hardware handshake irrelevant for printer driver

Explanation of the parameters

Baud rate (parameter block 0)

You have a choice of eight Baud rates. The default is 9600 bit/s. If you use the RS-232C (V.24) interface, the load capacity of cables longer than 15 m will have a negative effect on the Baud rate. Longer cables can be used in general if the Baud rate is reduced.

Parity (parameter block 0)

You can choose between five types of parity.

- 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 always has signal state ”1”.
- ”Space”
  The parity bit always has signal state ”0”.
- No parity check
  The signal state of the parity bit is not significant. Parity is not checked when receiving; however, when sending, the parity bit is always set to ”1”.

The default is even parity.

BUSY signal (parameter block 0)

The printer generates the BUSY signal, and indicates readiness to receive.
The RS-232C (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 module in print mode without XON/XOFF protocol. The waiting times for ”CR”, ”LF” and ”FF” are not significant when evaluating the BUSY signal.

Interface (parameter block 0)

You can choose between the RS-232C (V.24) and TTY interfaces here. See Section 2.3 for more detailed information on these interfaces.

The TTY interface is the default.

Data format (parameter block 0)

Characters are transmitted between the CP 521 SI and the peripheral device in a 10-bit or 11-bit character frame. You can choose between seven and eight data bits within these character frames (Figure 4-4, Figure 4-5).

10-bit character frame:

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

11-bit character frame:

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

The 11-bit character frame (1 start bit, 7 data bits, 1 parity bit, 2 stop bits) is the default.

Parameters for waiting times (parameter block 1)

These parameters are only relevant if you are not using an XON/XOFF protocol or BUSY signal when printing.

You can set waiting times for Carriage Return (CR), Line Feed (LF) and Form Feed (FF) in a time frame of 10 ms.

Table 5-5. Parameter Assignment Data for the Serial Interface (Parameter Block 1)

Block	Meaning	Value Range	Default Values on the CP 521 SI
1	Waiting times after CR LF FF	(000 to 999) · 10 ms (000 to 999) · 10 ms (000 to 999) · 10 ms	25=0.25 s 25=0.25 s 250=2.5 s

Parameters for XON/XOFF protocol (parameter block 2)

If you have an XON/XOFF 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. The same values must not be used when assigning parameters.

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.

Special feature: In print mode with XON/XOFF protocol, the CP 521 SI interprets each received character which is not ”XON” as ”XOFF”. The CP 521 SI then blocks output of all characters until it receives ”XON” from the peripheral device.

Table 5-6. Parameter Assignment Data for the Serial Interface (Parameter Block 2)

Block	Meaning	Value Range	Default Values on the CP 521 SI
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)

5.3.2 Parameter Assignment Data for Entering Message Texts (Parameter Block 3)

When configuring message texts (Section 5.5), you must specify an end-of-text character at the end of the message text. If you configure place holders in the message text, the place holders must be limited by a function character. You can configure which ASCII characters you want to use as function characters and end-of-text characters in parameter block 3.

Table 5-7. Parameter Assignment Data for Entering Message Texts (Parameter Block 3)

Block	Text Parameter	Value Range	Default Values on the CP 521 SI
3	Function character	ASCII character	” (22_H)
	End-of-text character	ASCII character	$ (24_H)

5.3.3 Parameter Assignment Data for Message Text Printout (Parameter Blocks 4 to 6)

Store configuration data for page layout and for inserting the date and time in parameter blocks 4, 5 and 6.

If you exceed the possible values, the values entered will be replaced with the relevant limit values.

This means

- When the lower limit is exceeded, the lower limit value is used
- When the upper limit is exceeded, the upper limit value is used.

Example: Input: Line/page ”10_H” is replaced with ”14_H”
Input: Left margin ”80_H” is replaced with ”$3C_H$”

> **Note**
>
> You can assign parameters in parameter blocks 4 and 6 with either upper or lower case characters.

Table 5-8. Parameter Assignment Data for Message Text Printout (Parameter Blocks 4 to 6)

Block	Meaning	Value Range	Default Value on the CP 521 SI
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 (Lower case also permissible)	TMJ
	Separator for date	ASCII character (20_H to $7F_H$)	”.” ($2E_H$)
	Order for time of day H =Hours M =Minutes S =Seconds	HMS, HSM, MSH, MHS, SHM, SMH (Lower case also permissible)	HMS
	Separator for time of day	ASCII character (20_H to $7F_H$)	”:” ($3A_H$)
	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_D to 255_D 0_D to 60_D O, o, T, t U, u, B, b no other characters	72_D 0_D Page number not inserted
6	Header and footer		
	Header 1 Header 2 Footer 1 Footer 2	K1 ... Text ... $ F1 ... Text ... $ F2 ... Text ... $ K2 ... Text ... $ ($=End-of-text character)	No header or footer

Page format (parameter block 5)

Table 5-9. Page Format

Line	Page Format	
1	Blank line	(if page number, header or footer has been configured)
	Blank line	(if page number, header or footer has been configured)
.	Page No.	(top, if configured)
.	Blank line	(if page number configured at top)
.	Header 1	
.	Header 2	
.	Blank line	(if header(s) configured)
.	Blank line	(if header(s) configured)
.		
.	Lines for message texts	
.	A maximum of 255 message texts can be configured.	
.	Each individual message text must not be configured with more than 136 characters.	
.	Configuration is described in Section 5.5.	
.		
.	Blank line	(if two footers have been configured)
.	Blank line	(if one footer has been configured)
.	Footer 1	
.	Footer 2	
	Blank line	(if the page number is configured at the bottom)
n	Page No.	(bottom, if configured)
	Blank line	(if page number, header or footer has been configured)
	Blank line	(if page number, header or footer has been configured)

Left margin (configurable up to 60 characters)

Number of lines per page: 20 to 255

Configuring headers and footers (parameter block 6)

Up to two headers and two footers can be configured in parameter block 6. As in message texts, place holders can be used for inserting the date, time of day and control parameters.

LF, FF and CR and place holders for variables and message texts are not evaluated. They appear in the printout as configured.

Proceed according to the following schematic:

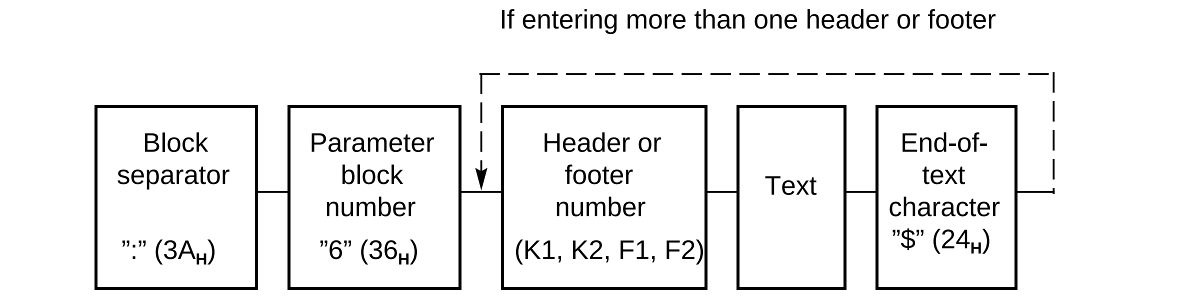


Figure 5-5. Schematic for Entering Headers and Footers

Enter the header or footer like the ASCII characters of a message text (Section 5.5).

> **Note**
>
> A header or footer must not be configured with more than 136 characters. The printed text may be longer than one line if place holders are used.

Continuous form feed

If you want to print your message texts without page format, you are recommended to assign the following message text parameters:

- Without page number (parameter block 5)
- Without header (parameter block 6)
- Without footer (parameter block 6)

This will give you continuous form feed for all message texts.

5.3.4 Configuring the Character Conversion Table (Parameter Block 8)

In order to adapt to special national characters, you can replace up to 16 ASCII characters with a code sequence of up to 16 characters in parameter block 8. Parameter block 8 must be configured according to the following schematic:

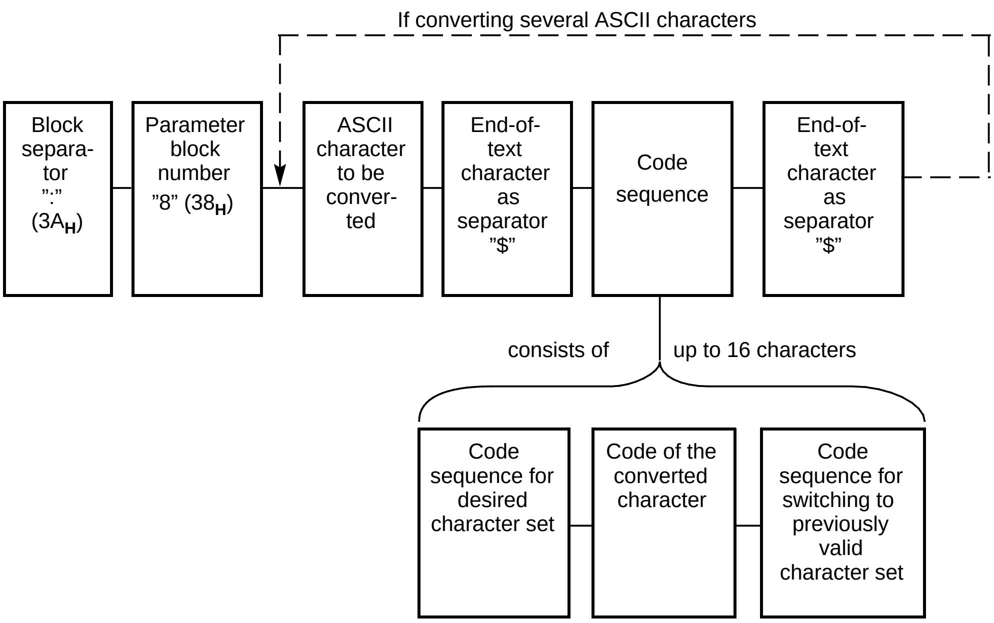


Figure 5-6. Schematic for Configuring the Character Conversion Table

5.3.5 Clock Correction Factor (Parameter Block 9)

You can configure a correction value in parameter block 9 to improve the accuracy of the module clock. The correction value is output in s/month. The month is fixed at 30 days.

Table 5-10. Correction Value for the Integral Clock (Parameter Block 9)

Parameter	Value Range	Default Value
Correction value	- 400_D to +400_D s/month	0000_D

Note

The correction value must always be specified with four digits (incl. sign)!

Example: Calculating the correction value

You have measured that the clock loses 12 s in four days. This would be 90 s in 30 days. The correction value is+090 s/month.

Entry on PG 750 in DB1	Explanation
`:   KS = ':9$+090$';`	Block separator ":" Parameter block number "9" Correction value "+090"

5.3.6 CP 521 SI Parameter Assignment Example

The CP 521 SI is installed in slot 7 (starting address 120). The parameters are to be assigned as follows:

- Parameter assignment data for entering message texts (parameter block 3)
  - Function character: " (")
  - End-of-text character: $ ($)

> **Note**
>
> If you wish to define an end-of-text character for the message text in parameter block 3, you must enter this parameter block in DB1 as the first parameter block.
> The end-of-text character (parameter block 3) functions as a separator between the various parameters assigned by a parameter block.

- Parameters for the serial interface (parameter block 0):
  - 9600 bit/s (8)
  - Even parity (0)
  - No BUSY (0)
  - RS-232C (V.24) interface (1)
  - 8 data bits (1)
  - Handshake OFF (0)

- Parameters for waiting times (parameter block 1):
  - Waiting time after CR: 100 ms (10)
  - Waiting time after LF: 100 ms (10)
  - Waiting time after FF: 2500 ms (250)

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

- Parameter assignment data for message text printout:
  - Time of day and date unchanged with respect to default values (parameter block 4)
  - Page format (parameter block 5)
    72 lines/page (072)
    5 characters left margin (05)
    Page number at top (o)

- Header (parameter block 6):
  Place holder for header 1/ (K1header 1)
  header 2 (K2header 2)
- Footer (parameter block 6):
  Place holder for footer 1/ (F1footer 1)
  footer 2 (F2footer 2)

• Configuring character conversion (parameter block 8)
  - Convert character ”+”

• Parameters for time of day correction (parameter block 9)
  - Correction factor: 1s/month (+001)

DB1 is invoked from the PG 750 programmer. You can then enter the values. Use comments (**KS**=...) to make your entries easier to understand.

Entries in DB1 from the PG 750		Erläuterung
0:	KS ='Para. assig. of the';	First of all, block 3 must be assigned
13:	S ='CP521SI';	its parameters if you wish to
25:	KS ='Parameter block_3 ';	change the default values.
34:	S =':3$"$ ';	$ = separator, " = function ch.
37:	KS ='Parameter block_0 ';	9600 baud, even parity, no
46:	S =':0$800110$';	Busy, V24, 8 bits, no handshake
51:	KS ='Parameter block_1 ';	CR = 10*10ms, LF = 10*10ms,
60:	S =':1$10$10$250$ ';	FF = 250*10 ms
67:	KS ='Parameter block_2 ';	XON = 11H, XOFF = 13H
76:	S =':2$11$13$ ';	
81:	KS ='Parameter block_4 ';	Day.month.year
90:	S =':4$TMJ.$HMS:d ';	Hour: minute: second 24h system
98:	KS ='Parameter block_5 ';	72 lines/page, left margin = 5
107:	S =':5$072$05o';	Page number at top
113:	KS ='Parameter block_6 ';	Header_1
122:	S =':6$K1header_1$K2head';	Header_2
132:	S ='er 2$F1footer_1$F2F ';	Footer_1
142:	S ='ooter_2$';	Footer_2
146:	KS ='Parameter block_8 ';	Character conversion
155:	S =':8$+$1B384B5B1B3840$';	
165:	KS ='Parameter block_9 ';	Time-of-day correction
174:	S =':9$+001$';	+ 1 second / 30 days
178:		

You can also enter all parameters in DB 1 in ASCII code (KH=hexadecimal constant), in which case it is advisable to make a list of alphanumeric characters and their ASCII codes side by side.

Appendix A contains a table of ASCII codes from which you can take the hexadecimal values and their corresponding ASCII characters.

In the following table, the same parameter assignment data have been entered as above in DB1, but in **KH** format (to save space, all comments have been omitted).

DB1	DB1 (cont.)
0: KH = 3A33;	35: KH = 4541;
1: KH = 2422;	36: KH = 4445;
2: KH = 243A;	37: KH = 525F;
3: KH = 3024;	38: KH = 3124;
4: KH = 3830;	39: KH = 4B32;
5: KH = 3031;	40: KH = 4845;
6: KH = 3130;	41: KH = 4144;
7: KH = 243A;	42: KH = 4552;
8: KH = 3124;	43: KH = 5F32;
9: KH = 3130;	44: KH = 2446;
10: KH = 2431;	45: KH = 3146;
11: KH = 3024;	46: KH = 4F4F;
12: KH = 3235;	47: KH = 5445;
13: KH = 3024;	48: KH = 525F;
14: KH = 3A32;	49: KH = 3124;
15: KH = 2431;	50: KH = 4632;
16: KH = 3124;	51: KH = 4620;
17: KH = 3133;	52: KH = 4F4F;
18: KH = 243A;	53: KH = 5445;
19: KH = 3424;	54: KH = 525F;
20: KH = 544D;	55: KH = 3224;
21: KH = 4A2E;	61: KH = 3A38;
22: KH = 2448;	62: KH = 242B;
23: KH = 4D53;	63: KH = 2431;
24: KH = 3A24;	65: KH = 4233;
25: KH = 6424;	66: KH = 3834;
26: KH = 3A35;	67: KH = 4235;
27: KH = 2430;	68: KH = 4231;
28: KH = 3732;	69: KH = 4233;
29: KH = 2430;	70: KH = 3834;
30: KH = 3524;	71: KH = 3024;
31: KH = 6F24;	72: KH = 3A39;
32: KH = 3A36;	73: KH = 242B;
33: KH = 244B;	74: KH = 3030;
34: KH = 3148;	75: KH = 3124;

5.4 Printing Parameter Assignment Data

You can print the parameter assignment data of the CP 521 SI with the CPU job request "$F0_H$". Although this job request is only permissible in print mode, all bidirectional mode parameters of the module are printed. Page feed is executed at the beginning and end of this job request.

Example of parameter assignment data printout (DR 210 printer):

```
                    CP521-PARAMETERS
                    ===========
BLOCK 0:
            BAUD RATE:          4800   (7)
            PARITY:             EVEN   (0)
            BUSY:               OFF    (0)
            V24/TTY:            V24    (1)
            FRAME:              8/11   (1)
            HANDSHAKE:          OFF    (0)

BLOCK 1:
            WAIT  CR:            30 * 10ms
            WAIT  LF:            30 * 10ms
            WAIT  FF:           250 * 10ms

BLOCK 2:
            XON:                FFH
            XOFF:               FFH

BLOCK 3:
            FUNCTION:           "  (22H)
            TEXT  END:          $  (24H)

BLOCK 4:
            DATE:               D.M.Y
            TIME:               H:M:S

BLOCK 5:
            LINES:              255
            PAGE OFFSET:        0
            PAGE NUMBER:        BOTTOM

BLOCK 6:
            HEAD  1:            NO
            HEAD  2:            NO
            FOOT  1:            NO
            FOOT  2:            NO

BLOCK 7:
            SELECTED  MODE:     PRINTER   (0)
```

(Continued from page 5 - 20)

```
ACTUAL  PARAMETERS  FOR  MODE  1
------------------------------------
DELAY  TIME:                        1  *  10ms
TEL  LENGTH:                          64 bytes

ACTUAL  PARAMETERS  FOR  MODE  2 / 6
------------------------------------
DELAY  TIME:                        1  *  10ms
TEL  END:                           0DH  00H

ACTUAL  PARAMETERS  FOR  MODE  3
------------------------------------
DELAY  TIME:                       22  *  10ms
TIME-OUT:                         200  *  10ms
FRAME  DELAY  TIME:               400  *  10ms
PRIORITY:                           LOW   (0)
BCC:                                NO    (0)
CONNECTION  BUILDUP  ATTEMPTS             6
TRANSMISSION  ATTEMPTS:                   6

ACTUAL  PARAMETERS  FOR  MODE  4
------------------------------------
SLAVE  NUMBER:                            0
```

BLOCK 8:

2BH == > 1BH 38H 4BH 5BH 1BH 38H 40H

BLOCK 9:

TIME CORRECTION: + 1 /month

5.5 Configuring Message Texts

At module power-up, the data blocks of the memory submodule (DB2 to DB63) are searched for message texts. Up to 255 message texts with a length of 136 characters can be stored. Exception: if you use place holders (Section 5.4.1) and/or a character conversion table, up to 250 characters can be evaluated.

A properly configured message text has the following construction:

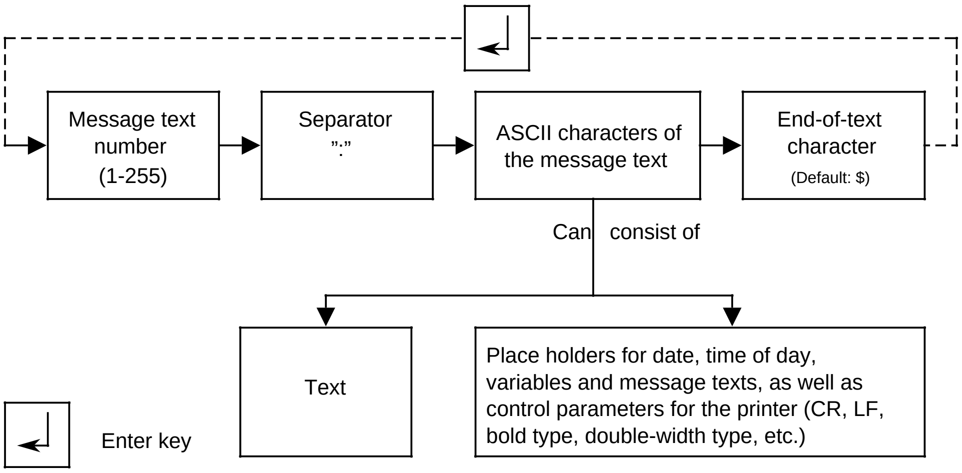


Figure 5-7. Structure of a Message Text

> **Note**
>
> Your message texts will be clearer if you alternate between data format "KS" and "S" in each input line.

Message text number

Store the message text in a data block under a 1-digit to 3-digit number (1 to 255).

Separator

You must use a colon as a separator after the message text.

ASCII characters of the message text

You must terminate every message text with an end-of-text character. You can assign the value of this character in DB1. Up to 136 characters are evaluated.
A message text can also be longer than 136 characters if you use place holders and/or a character conversion table. The maximum is 250 characters.

> **Note**
>
> If the printout of a message text is longer than 136 characters, this might interfere with the print format (depending on printer line feed, page makeup ...).

You can enter the following as ASCII characters:

- Text
  The text can contain all printable 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 DB2:

Message text 4: Excess temperature
Message text 5: Coolant loss

Entry on PG 750 in DB2	Explanation
`0: KS = '4:Excess temperature$ ';`	Message text number 4 Separator ":", message text, end-of-text character "$"
`11: S = '5:Coolant loss$';`	Message text number 5 Separator ":", message text, end-of-text character "$"

5.5.1 Entering Place Holders

Place holders are used for inserting the date, time of day, control characters, 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 (Section 5.3.3).

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 22_H=". Wrongly entered place holders are treated as text.

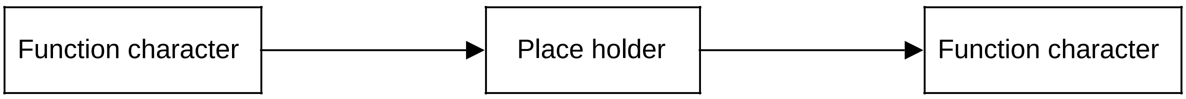


Figure 5-8. Schematic Structure of a Place Holder

Table 5-11. Overview of Place Holders Configurable in Message Texts

Place Holder	Meaning	Max. Number in One Message Text	Configurable in Header and Footer	Comment
D	Insert date	1	Yes	
T	Insert time of day	1	Yes	
KPa	Insert message text (fixed message text number)	Unlimited	No	The message text to be printed out must be 250 characters long.
KV	Insert message text (variable message text number)	3	No	The message text to be printed out must be 250 characters long. Transfer message text number to the CP 521 SI in bytes 2 to 7.
Data format (KH, KF KS, KM, KT, KC, KY)	Insert variable	3 variables per 16 bits or 1 variable per 32 bits + 1 variable per 26 bits	No	Transfer the values of the variables to the CP 521 SI in bytes 2 to 7.
Sa,b, ...	Transfer the control parameters for the printer	Max. 50	Yes	

You can enter place holders in either lower or upper case.

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 the KV place holder ”Insert message text” and the place holder for ”Insert variables”

Transfer the values for the KV place holder ”Insert message text” and the values for the place holder ”Insert variables” to bytes 2 to 7 on the CP 521 SI.
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.

5.5.2 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 (Section 5.3.3).

The following is the default output format:

- For the date: <Day>.<Month>.<Year>
- For the time of day: <Hour>.<Minute>.<Second>

The place holders can be configured either with upper-case or lower-case characters. Wrongly entered characters are treated as text.

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

You want to store the following message texts in DB3:
Message text 7: The motor went down at <time of day>
Message text 8: This is the daily listing for <date>

Entry on PG 750 in DB3	Explanation
`: KS = '7:The motor went down';`	Message text number ”7”, Separator ”:”. message text with place holder for time of day
`: S = 'at ″T″$';`	Message text, end-of-text character ”$”
`: KS = '8:This is the daily';`	Message text number ”8”, Separator ”:”, message text
`: S = 'listing for ″D″$';`	Message text, place holder for date, end-of-text character ”$”

5.5.3 Place Holders for Message Texts

These place holders enable you to insert further message texts into an existing message text.

The message text number of the message text to be inserted can be specified in two different ways:

- Direct entry of the message text
- Message text number as variable

Direct entry of the message text number ”KPa”

”a” specifies the number of the message text to be used. ”a” must be specified when configuring the message text. The values 1 to 255 are permissible for ”a”. If there are place holders in the message text called, they are treated as text and printed out.

You can use the ”KPa” place holder several times in a message text. In this way, you can combine 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 DB7:

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 down times.

Entry on PG 750 in DB7	Explanation
`0: KS= '70:The following over ';`	Message text number, separator, message text
`11: S = 'view indicates "KP73",';`	with place holder for "Insert message text 73"
`22: KS= ' "KP74" and "KP75".$';`	Message text, end-of-text character
`32: S = '73:the problems that';`	Message text no., separator, message text
`42: KS= 'have arisen during';`	Message text
`51: S = 'the monitoring period$';`	Message text, end-of-text character
`62: KS= '74:the measures ';`	Message text no., separator, message text
`70: S = 'taken$';`	Message text, end-of-text character
`73: KS= '75:the length of the';`	Message text no., separator, message text
`83: S = ' resulting';`	Message text
`88: KS= ' down times$';`	Message text, end-of-text character

Printing message text 70:

The following overview details the problems which have occurred during the monitoring phase, the measures which have been taken and the length of the resulting downtimes.

Message text number as ”KV” variable

In the case of the ”Print message text” job request, if you have configured the ”KV” variable in a message text, you must specify in bytes 2 to 7 the numbers of the message texts to be inserted:

- Byte 2 and 3 : Number of the 1st message text to be inserted (binary coded)
- Byte 4 and 5 : Number of the 2nd message text to be inserted (binary coded)
- Byte 6 and 7 : Number of the 3rd message text to be inserted (binary coded)

You can use the ”KV” place holder up to three times in one message text.

You can configure the ”KV” place holder at the same time as place holders for variables in one message text. You must then ensure that you transfer the data in bytes 2 to 7 correctly.

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

You want to store the following message text in DB8:

Message text 80: ”KV” has ”KV” because of ”KV”
Message text 81: The motor
Message text 82: thermic problems
Message text 83: water loss
Message text 84: Motor ”KH” went down because of ”KV” at ”T”
Message text 85: EMERGENCY OFF

Entry on PG 750 in DB8	Explanation
`: KS = '80:”KV” has ”KV”';` `: S = 'because of ”KV”.$';`	Message text number, separator ”:”, message text with place holders for message text, end-of-text character
`: KS = '81:The motor$';`	Message text number, separator ”:”, message text, end-of-text character
`: S = '82:thermic problems$';`	Message text number, separator ”:”, message text, end-of-text character
`: KS = '83:water loss$';`	Message text number, separator ”:”, message text, end-of-text character

`: S = '84:Motor ”KH” went';`
`: KS = 'down because of ”KV”.';`
`: S = 'at ”T”$';`
	Message text number, separator ”:”, message text with place holders for variables and inserting message text, end-of-text character

`: KS = '85:EMERGENCY OFF$';` `:`	Message text number, separator ”:”, message text, end-of-text character

Printout of message text 80 with insertion of message texts 81, 82 and 83:

The motor has thermic problems as a result of water loss.

Printout of message text 84, with the CPU transferring 20 as the variable value and 85 as the message text to be inserted:

Motor 20 went down as a result of EMERGENCY STOP at 12:00:00.

Maximum length of message texts

A message text can be up to 136 characters long. If you use place holders, the printout can be longer than 136 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 DB9:

Message text 90: The following table gives an overview of the faults which have occurred in the press controller together with <Insert message text 91> as well as <Insert message text 92> <Insert message text 93>.

Message text 91: the downtimes, the resulting waiting times for the next press controller

Message text 92: the average

Message text 93: press controller down times as recorded during the current year

Entry on PG 750 in DB9	Explanation
`0: KS = '90:The following table g';`	Message text 90 with a length of 145 characters (incl. place holders)
`12: S  = 'ives an overview of the ';`	
`24: KS = 'faults which have occurr';`	
`36: S  = 'ed in the press controll';`	
`48: KS = 'er together with "KP91" ';`	
`60: S  = 'as well as "KP92" "KP93"';`	
`72: KS = '.$';`	Message text 91 with a length of 72 characters
`73: S  = '91:downtimes, the result';`	
`85: KS = 'ing waiting times for th';`	
`97: S  = 'e next press controller$';`	
`109: KS = '92:the average$';`	Message text 92 with a length of 11 characters
`117: S  = '93:press controller down';`	
`129: KS = ' times as recorded durin';`	Message text 93 with a length of 63 characters
`141: S  = 'g the current year$';` *	
`151:`	

* This text will not be printed

Final printout:

The following table gives an overview of the faults which have occurred in the press controller together with the downtimes, the resulting waiting times for the next press controller as well as the average press controller down times as recorded during (250 characters)

5.5.4 Place Holders for Variables

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

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

The first configured place holder for variables in the message text is replaced by variable 1, the second by variable 2 and the third by variable 3.

Note

- 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 136 characters.

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

Table 5-12. Data Formats for Variables

Data Format		Value Range	Number of Printed Positions
KM	Constant: Bit pattern	16 Bit	16
KH	Constant: Hexadecimal pattern	0000_H to $FFFF_H$	4
KS	Constant: Alphanumeric characters	20_H to $7F_H$/20_H to $7F_H$	2
KT	Constant: Timebase	0.00 to 9990 *	4
KC	Constant: Count	0 to 999	3
KB	Constant: Byte	0 to 255	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 (9) b Places after point (6)	- 32768 to +32767	Configurable (9)
KG	Constant: Floating-point number	- 1.7E+38 to - 1.17E-37 1.17E-37 to 1.7E+38	
KG a,b	Constant: Floating-point number a Places before point (9) b Places after point (6)	+/- 0.000001 to 999999.9	Configurable (9)

* The time is printed out in seconds

”KT” data format

You can use the ”KT” data format to insert the values of internal timers in a message text. Load the time into the accumulator in BCD with LC TX and then transfer it to the CP 521 SI. The four digits are printed out without specifying the unit ”s”.

Table 5-13. ”KT” Data Format Printouts (Examples)

Time	3.0	24.0	207.0	8.1	46.1	840.1	1.2	93.2	516.2	4.3	69.3	423.3
Printout of the KT Variable	0.03	0.24	2.07	0.8	4.6	84.0	001	093	516	0040	0690	4230

(=space)

”KB” data format

In the KB ”Constant byte” data format, the CP 521 SI evaluates the lower-order byte of a sixteen bit variable as the value for the variable:

- Byte 3 Value of the 1st variable
- Byte 5 Value of the 2nd variable
- Byte 7 Value of the 3rd variable

”KF” data format

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

Table 5-14. ”KF” Data Format Printouts (Examples)

Variable Value	+12345	+357	- 12345	- 357
Printout of the KF Variable	12345	357	- 12345	- 357

(=space)

”KFa,b” data format

You can set the following possible parameters:

- Specify the number of places before the point with ”a”.
  The sign and the decimal point each count as one character.
- Specify the number of places after the point with ”b”. This is the same as correcting the variable value by a factor of 10^{-b}.

Table 5-15. Value Range for the Number of Places Before and After the Decimal Point

Places before the Point ”a”	1 to 9	3 to 9	4 to 9	5 to 9	6 to 9	7 to 9
Places after the Point ”b”	0	1	2	3	4	5

Errors when configuring the ”KFa,b” data format have the following effects:

- If you enter values for ”a” and ”b” outside the value range, the place holder will be treated as text and printed as configured.
- If you configure ”a” too low for the variable value to be printed, question marks ”?” will be printed in place of the variable value ”a”.
- A decimal comma is printed in the case of the ”KFa,b” data format and in the case of the ”KFa.b” format, a decimal point is printed.

Table 5-16. Examples of ”KFa,b” and ”KFa.b” Data Format Printouts

Configu-ration	Printout of KF= Variables							
	– 00001	00008	12345	– 12345	00045	– 00045	00345	– 00345
KF 6,2	– 0,01	0,08	123,45	??????	0,45	– 0,45	3,45	– 3,45
KF 7.2	– 0.01	0.08	123.45	– 123.45	0.45	– 0.45	3.45	– 3.45
KF 9,2	– 0,01	0,08	123,45	– 123,45	0,45	– 0,45	3,45	– 3,45
KF 3.0	???	8	???	???	45	– 45	345	???
KF 2,0	??	8	??	??	45	??	??	??
KF 9.5	– 0.00001	0.00008	0.12345	– 0.12345	0.00045	– 0.00045	0.0345	– 0.00345
KF 7,5	???????	0,00008	0,12345	???????	0,00045	???????	0,00345	???????

(=space)

Initial zeros are not suppressed. They are replaced by spaces.

”KG” and ”KGa,b” data formats

You can print out 32-bit floating-point numbers in a message text. A floating-point number requires four bytes of memory.

The value for the floating-point number can be transferred as follows:

- In bytes 2 to 5, if the place holder is the first or only place holder in the message text,
- In bytes 4 to 7, if the place holder is the second in the message text.

The memory requirement of four bytes has further consequences:

- Only one place holder for ”Insert floating-point number” KG or KGa,b can be configured in a message text. A second place holder of this type would be interpreted as text and printed out as configured.
- In addition to the ”Insert floating-point number” place holder, only one other ”KV” or ”Insert variable” place holder can be configured in a message text.

”KG”, ”KG.” ”KG,” data formats

If you configure the ”KG” or ”KG.” data format, 13 characters are always printed out. A decimal comma is printed in the case of the ”KG” and ”KG,” data formats and, in the case of the ”KG.” format, a decimal point is printed.

”KGa,b” data format

You can set the following parameters:

- Specify the number of places before the point with ”a”.
  The sign and the decimal point each count as one character.
- Specify the number of places after the point with ”b”.
  In contrast to the ”KFa,b” data format, the value of the variable is not changed.
- A decimal comma is printed in the case of the ”KGa,b” data format and, in the case of the ”KGa.b” format, a decimal point is printed.

Table 5-17. Value Ranges for the Number of Places Before and After the Decimal Point

Places before the Point ”a”	1 to 9	3 to 9	4 to 9	5 to 9	6 to 9	7 to 9	8 to 9
Places after the Point ”b”	0	1	2	3	4	5	6

Errors when configuring the ”KGa,b” or ”KGa.b” format have the following effects:

- If you enter values for ”a” and ”b” outside the value range, the place holder will be treated as text and printed as configured.
- If you configure ”a” too low for the variable value to be printed, question marks ”?” will be printed in place of the variable value ”a”.

Table 5-18. Examples of ”KGa,b” and ”KGa.b” Data Format Printouts

Number to be Repre-sented	**Configuration**							
	KG9.0	**KG9.4**	**KG9.6**	**KG6.3**	**KG5.4**	**KG4.1**	**KG3.1**	**KG1.0**
3.141593E+01	31	31.4159	31.415930	31.416	'KG 5.4'	31.4	31	?

5.5.5 Control Parameters

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

- Entry with the ^ character (CTRL key)
- Entry with the Sa,b place holder
- Direct with the KH format

Entry with the ”^” character

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 DR 210 printer. The default is the ECMA character set. 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 the 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 issues the printer with the job request 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 DB5:
Message text 50: <Double-width type on> Monthly overview <Double-width type off>

Entry on PG 750 in DB5	Explanation
`: KS = '50:^[8 Monthly overview^[<$';` `:`	Message text no. 50, separator ”:”, switch on character string for double-width type, message text, switch off character string for double-width type

Entry with the ”Sa,b...” place holder

If you use the ”Sa,b” place holder, you must enter the control characters in ASCII code in decimal representation. If you want to enter several control characters in succession, you must separate them with commas. You can enter up to 50 control characters in succession.

Example: Switch off the index on the DR 210. The default is the ECMA character set. This is done with the ”ESC 16” command. ”ESC” corresponds to ASCII code $1B_H=27_D$, You must enter: ”S27,16”.

Example: Configuring a message text with a place holder for control parameters (entry with Sa,b)

You want to store the following message text in DB6:
Message text 60: The proportion of H <Switch index on> 2 <Switch index off> 0 is 50%

Entry on PG 750 in DB6	Explanation
`: KS = '60:The proportion of';`	Message text number, separator ”:”, message text
`: S = 'H ”S27,18”2”S27,16” O';`	Message text ”H”, place holder for index on, message text ”2”, place holder for index off, message text ”O”
`: KS = 'is 50%.$';`	Message text, end-of-text character

Direct entry with the "KH" data 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 "S" format to the "KH" format and specify the control parameters in ASCII code. You then switch back to the "KS" or "S" data format.

> **Note**
>
> If you print out message texts with control parameters that have been entered direct using job request 8000_H "Print all configured message texts", the control parameters will be executed and will not appear as configured.

Example: Configuring a message text with a place holder for control parameters (direct entry)

You want to store the following text in DB4:
Message text 40: Motor works <Line feed> <Carriage return> Brummhausen <Line feed> <Carriage return> 9999 Wackeldorf

Entry on PG 750 in DB6	Explanation
`: KS = '40:Motor works';`	Message text no. 40, separator ":", message text
`: KH = 0A0D`	ASCII code for line feed ($0A_H$) and carriage return ($0D_H$)
`: S = 'Brummhausen';`	Message text
`: KH = 0A0D`	ASCII code for line feed ($0A_H$) and carriage return ($0D_H$)
`: KS = '9999 Wackeldorf$';` `:`	Message text, end-of-text character

5.6 Printing Message Texts

Printing of the message texts stored on the memory submodule is initiated by the CPU over the user program.

One message text per program cycle can be printed over the CP 521 SI.

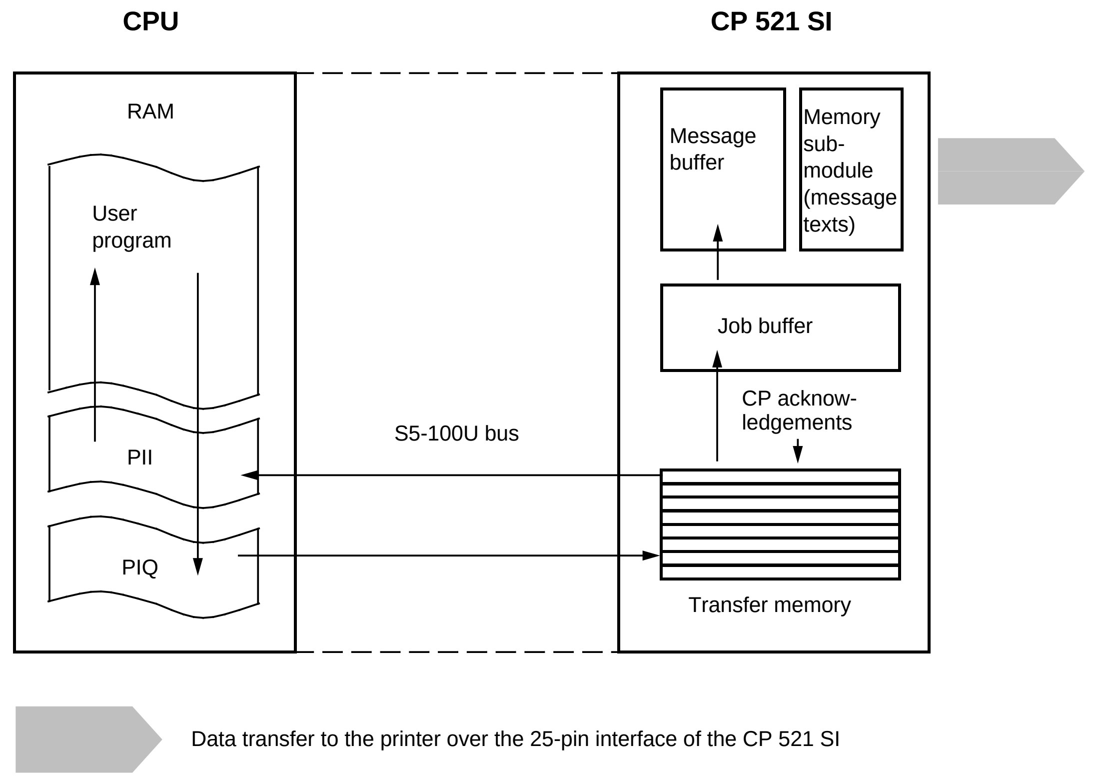


Figure 5-9. Data Interchange Over the CP 521 SI

When configuring the message texts, you have assigned a number to each text. This is important, since you must transfer the message text number to the CP 521 SI along with the CPU job request ”Print message text”. If variables have been configured in the message, the current variable values must also be transferred from the CPU to the CP 521 SI with the message text number.

If there are several message texts which the printer cannot process immediately, the CP 521 SI stores these messages in the message buffer. The print job requests are then processed in the order they are sent from the CPU.

If you want to print a complete list consisting of several lines of message, the list lines are configured as individual messages. You can print one list line per program cycle.

Contents of the transfer memory when printing out message texts

Write CPU "Print message text" job to the PIQ:

Request from the CPU to the CP: "Print message text"

Address	Contents
Maddr+0	Print job
Maddr+1	
Maddr+ 2	Further possible specifications for print job
Maddr+ 3	
Maddr+ 4	
Maddr+ 5	
Maddr+ 6	
Maddr+7	

Code for "Print message text" Table 5.19

Evaluate CP response to "Print message text" job:

CPU job accepted
Message text is printed

Address	Contents
Maddr +0	50_H
Maddr +1	00_H
Maddr + 2	irrelevant
Maddr + 3	irrelevant
Maddr + 4	irrelevant
Maddr + 5	irrelevant
Maddr + 6	Print job
Maddr +7	

CP rejects CPU job:
Job buffer full/wrong job

Contents
$4x_H$
xx_H
irrelevant
irrelevant
irrelevant
irrelevant
Print job

Acknow-ledged job

x: Status information of the CP 521 SI (Section 5.7)

> **Note**
>
> The CPU acknowledgements 5000_H and $4xxx_H$ remain (i.e. are not deleted) until they are overwritten by a subsequent job (e.g. blank job 0000_H).

CPU job requests for printing message texts

The CPU transfers the job request to the CP 521 SI in bytes 0 and 1. The job request number is stored in byte 0.

Table 5-19. CPU Job Requests for Printing Message Texts (Byte 0 in the PIQ)

Byte 0	Byte 1	Meaning	
$\boxed{20_H}$	00_H	Specify page number in byte 1	(Section 5.6.1)
$\boxed{30_H}$	XX_H	Output message with CR/LF at end XX_H : 01_H to FF_H (message text number: 1 to 255)	(Section 5.6.2)
$\boxed{40_H}$	XX_H	Output message without CR/LF at end XX_H : 01_H to FF_H (message text number: 1 to 255)	(Section 5.6.2)
$\boxed{50_H}$	00_H	Output page feed	(Section 5.6.3)
$\boxed{60_H}$	00_H	Output line feed	(Section 5.6.4)
70_H	00_H	Delete message buffer	(Section 5.6.5)
80_H	00_H	Printout of all messages	(Section 5.6.6)
$\boxed{}$	Print job requests. 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 request 80_H.

5.6.1 Setting the Page Number

With job request 20_H, you define which page number is to be printed on the next page. You must first have programmed the position of the page number in parameter block 5.

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 in hexadecimal code (01_H to FF_H)
- Specify job request 20_H in byte 0.

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

The module is plugged into slot 7 (starting 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 request and page number 14_H=20_D to word 0.

5.6.2 Outputting Message Texts with and without CR/LF at the End

Specify the message text number of the text to be printed in byte 1 in hexadecimal representation. Job request 30_H inserts a "CR/LF" at the end of the message text. Job request 40_H "Print message text without CR/LF" allows you to print several message texts, and therefore more than three variables, in one line. If your configured message texts contain, for example, only variables, you can generate tables with this job request.

Prerequisites:

- You must have configured the messages to be printed on the memory submodule. When configuring, you assign a number to each message text.
- You must have plugged the configured memory submodule into the CP 521 SI.

Operator inputs required:

- Specify the message text number in hexadecimal form in byte 1
- Specify the job request in byte 0
  - Job request 30_H: Print message text with CR/LF
  - Job request 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.

At printout, the place holders are replaced by actual variable values.

Table 5-20. PIQ in the Case of the "Print Message Texts" Job Requests

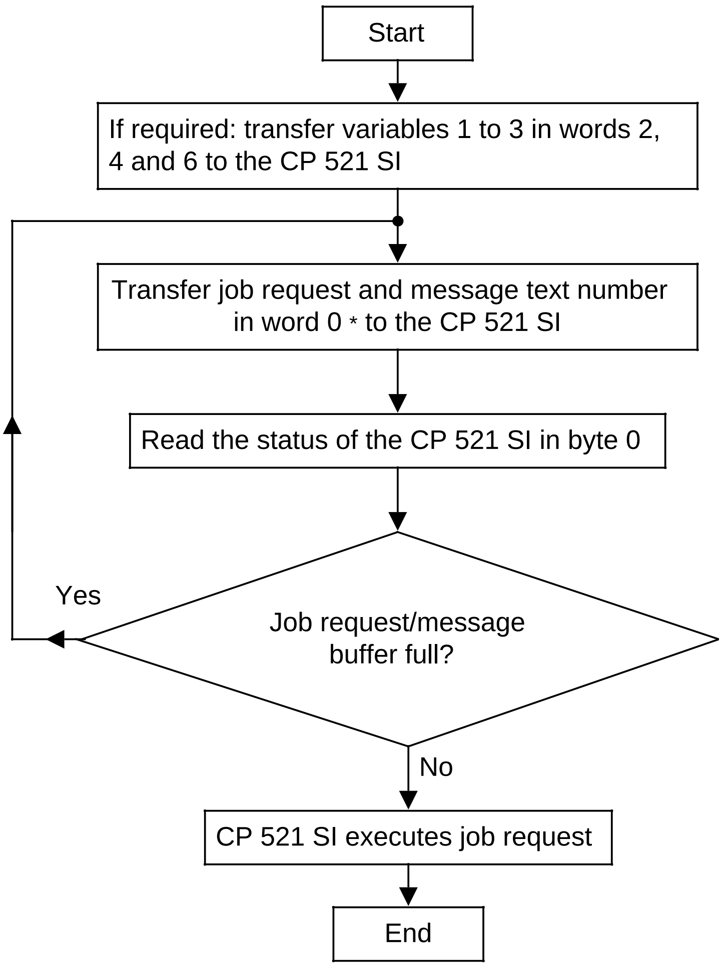
Byte	Meaning	Permissible Assignment
0	Job request numbers	30_H : Print message text with CR/LF 40_H : Print message text without CR/LF 50_H : Output page feed 60_H : Output line feed
1	Message text number	01_H to FF_H (1 to 255)
2 3	Value of 1st variable	0000_H to $FFFF_H$ (dependent on configured data format)
4 5	Value of 2nd variable	0000_H to $FFFF_H$ (dependent on configured data format)
6 7	Value of 3rd variable	0000_H to $FFFF_H$ (dependent on configured data format)

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 by the texts etc. they represent.

You can enter place holders for:

- Date and time of day (Section 5.5.2)
- Further message texts (Section 5.5.3)
- Variables (Section 5.5.4)
- Control characters for the printer (double-width type, bold type ...) (Section 5.5.5)



* Section 4.2

Figure 5-10. Schematic Flowchart of the ”Print Message Text” Job Request

The CP 521 SI can send further error messages to the CPU (Section 5.7).
For example:

- Default time of day set
- Printer not ready

Example: Initiating one-off output of a message text

The CP 521 SI is located in slot 7 (starting address 120). A memory submodule containing message text 9 is plugged into the CP 521 SI (Section 5.5). If flag 20.0 is set, message text 9 is to be printed out.
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 recognized in word 0 as a change (edge-triggered job request processing).

STL FB100	Explanation
NAME :TEXT1	
:AN F 20.0 :BEC	If flag 20.0 is not set, the program is terminated.
:L KH 3009 :T QW 120	Load job request no. 30_H and message text no. 9 into ACCUM 1 and transfer to word 0 in the CP 521 SI.
:BE	Block end

Example: Triple use of the "KV" place holder

You have configured the following message text on the memory submodule (Section 5.5):
Message text 80: The "KV" has "KV" as a result of "KV"
Message text 81: motor
Message text 82: water loss
Message text 83: thermic problems

Starting address 120 is set on the CP 521 SI. You have programmed the following operations in the user program:

STL FB105	Explanation
NAME :TEXT2 :AN F 20.1 :BEC	
:L KH 3050 :T QW 120 :L KF +81 :T QW 122 :L KF +82 :T QW 124 :L KF +83 :T QW 126	30_H: Job request no. for print job request with CR/LF, 50_H: = 80_D; no. of the message text to be printed
:BE	

Printout of message text 80 with insertion of message texts 81, 82 and 83:
The motor has thermic problems as a result of water loss.

Example: Simultaneous use of the ”KV” place holder, the place holder for the variable ”KH” and the place holder for the time of day

You have configured the following message texts on the memory submodule (Section 5.5):
Message text 84: Motor ”KH” went down as a result of ”KV” at ”T”.
Message text 85: EMERGENCY OFF

Motor 20 went down at 17:15:30.
Starting address 120 is set on the CP 521 SI. You have programmed the following operations in the user program:

STL FB106	Explanation

`NAME :TEXT3`
`:AN F 20.2`
`:BEC`	

`:L KH 3054` `:T QW 120`	30_H: = Job request no. for print job request with CR/LF, 54_H: = 84_D; no. of the message text to be printed

`:L KH 0020`
`:T QW 122`
`:L KF +85`
`:T QW 124`	

`:BE`	

Message text 84 is printed as follows (=space):
Motor 20 went down as a result of EMERGENCY OFF at 17:15:30.

5.6.3 Executing a Page Feed

Job request 50_H execute a page feed according to the page format parameters set (Section 5.3.3).

> **Note**
>
> If this job cannot be executed immediately, because, for example, there are still message texts to be printed, it is stored in the message buffer. If the ”Output page feed” job request is issued, job 60_H ”Output line feed” is deleted in the message buffer and no longer executed.

Operator inputs required

You must specify job request 50_H in byte 0.

Example: Outputting page feed with the ”FORCE VAR” programmer function

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

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

5.6.4 Outputting a Line Feed

Job request 60_H generates a blank line. If the job cannot be executed immediately, it is stored in the message buffer.

Operator inputs required

Specify job request 60_H in byte 0.

Example: Output line feed with the ”FORCE VAR” programmer function

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

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

Note

If the ”Output page feed” job request (50_H) is issued immediately afterwards, all CPU job requests 60_H in the message buffer are deleted.

5.6.5 Deleting the Message Buffer

The CP 521 SI uses job request 70_H to delete all CPU job requests 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 request 70_H in byte 0.

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

The module is located in slot 7 (starting address 120).

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

5.6.6 Printing All Configured Message Texts

When issued with job request 80_H by the CPU, the CP 521 SI prints out all configured message texts on the user submodule. This job request is used to determine which message texts are stored on the memory submodule and to check the message texts. The message texts are printed as configured with the message text number in ascending order. Place holders are not evaluated.

This CPU job request is not buffered. It is executed immediately even if other CPU job requests (Print message text) have been initiated earlier. Page feed is executed at the start and end of the job request.

You can issue the job request in two ways:

- In the user program
- In RUN mode with the "FORCE VAR" programmer function

Operator inputs required

Specify job request 80_H in byte 0.

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

The module is located in slot 7 (starting address 120).

Operand	Signal States	Meaning
QW 120	KH= 8000	Job request: "Print all configured message texts"

5.7 Status of the CP 521 SI in Print Mode

If you have activated the printer driver, you can evaluate the following CP 521 SI status information in the PII in byte 0 (status byte) (Section 4.4.1).

Table 5-21. Status Byte in Printer Driver Mode (PII)

Bits 4 to 7	Bits 0 to 3	Status	Explanation
0	0	No error	Job buffer empty
X	1	Memory submodule error	Wrongly configured memory submodule
X	2	No texts	The memory submodule contains no configured message texts
X	7	Backup battery in CP 521 SI missing	On the CP 521 SI - No battery inserted - Battery defective
X	8	Message buffer overflow	The module cannot process any further print job requests at present. The job request must be repeated.
X	F	CP 521 SI in restart or job buffer full	Message occurs only at restart: the clock data is invalid and the module cannot accept any job requests.
1	X	Clock defective	Replace module
2	X	Default clock time set	The clock is set with the values Sunday, 1.1.90, 12:00:00.
3	X	Time/date error	At least one setting is outside the permissible range. The clock has not accepted the new clock data and continues with the current data.
4	X	Illegal job request	You have issued a job request to the CP 521 SI which is not permissible in print mode. This error message also appears when illegal parameters are transferred in a parameter assignment job request (Section 5.6).
8	X	Hardware fault	

X= Signal state not significant for the other nibble

In addition, you can evaluate the status of the peripheral device in byte 1 of the PII under certain conditions (Section 4.4.2).

6 ASCII Driver

6.1 Prerequisites for Operation with ASCII Drivers . . . . 6 - 5

6.2 Peripheral Interface Connections . . . . 6 - 7

6.3 Assigning the CP 521 SI Parameters for ASCII Mode . . . . 6 - 9
6.3.1 Assigning the CP 521 SI Parameters with the Memory Submodule . 6 - 10
6.3.2 Assigning the CP 521 SI Parameters in the User Program . . . . 6 - 17

6.4 Data Transfer Between the CP 521 SI and the CPU . . . . 6 - 21
6.4.1 Sending Message Frames . . . . 6 - 22
6.4.2 Receiving Message Frames . . . . 6 - 32

6.5 CPU Job Requests and CP Error Messages . . . . 6 - 39

6.6 STEP 5 Programs for Data Transmission with ASCII Driver . . . . 6 - 44

Figures

6-1. Timing Diagram for Data Transfer Between the CP 521 SI and a Peripheral Device . . . . 6 - 4
6-2. Terminal Diagram CP 521 SI (TTY Passive) - CP 523 (TTY Active) . . . . 6 - 8
6-3. Terminal Diagram CP 521 SI - CP 521 SI (RS-232C (V.24) Interface with HW Handshake) . . . . 6 - 8
6-4. Schematic for Entering Parameter Blocks in DB1 . . . . 6 - 10
6-5. Data Interchange Over the CP 521 SI . . . . 6 - 21
6-6. Sequence Schematic for Sending Data . . . . 6 - 23
6-7. Sequence Schematic when Receiving Data . . . . 6 - 34
6-8. Structure of the STEP 5 Program for Data Transmission with ASCII Driver . 6 - 45

Tables

6-1. Possible Control Signals of the RS-232C (V.24) Interface in Handshake ON Mode . . . . 6 - 3
6-2. Pin Assignments of the 25-Pin Subminiature D Connector of the CP 521 SI 6 - 7
6-3. Parameter Blocks for ASCII Mode . . . . 6 - 9
6-4. Parameter Block Assignments on the Memory Module . . . . 6 - 11
6-5. Assignments in Parameter Block 7 (Transparent ASCII Driver) . . . . 6 - 12
6-6. Assignments in Parameter Block 7 (Interpretive ASCII Driver) . . . . 6 - 12
6-7. Configuring the CP 521 SI for Interpretive ASCII Driver Mode . . . . 6 - 16
6-8. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 0” Job Request . . . . 6 - 18
6-9. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 2” Job Request . . . . 6 - 19
6-10. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 7” Job Request (Transparent ASCII Driver) . . . . 6 - 19
6-11. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 7” Job Request (Interpretive ASCII Driver) . . . . 6 - 20
6-12. Permissible CPU Job Requests when Sending Message Frames (PIQ) . . . . . . 6 - 22
6-13. Coordination Job Request . . . . 6 - 25
6-14. Coordination Information . . . . 6 - 26
6-15. Data Transfer: Sending the 1st Message Block . . . . 6 - 27
6-16. Acknowledging the 1st Message Block . . . . 6 - 27
6-17. Data Transfer: Sending the 2nd Message Block . . . . 6 - 28
6-18. Acknowledging the 2nd Message Block . . . . 6 - 29
6-19. Data Transfer: Sending the 43rd Message Block . . . . 6 - 29
6-20. Acknowledging the Last (43rd) Message Block . . . . 6 - 30
6-21. Final Coordination Information . . . . 6 - 30
6-22. Coordination Job Request . . . . 6 - 31
6-23. Permissible CPU Job Requests when Receiving Data (PIQ) . . . . 6 - 33
6-24. Coordination Job Request . . . . 6 - 35
6-25. Data Transfer: Receive 1st Message Block . . . . 6 - 35
6-26. Acknowledging the 1st Message Block . . . . 6 - 36
6-27. Data Transfer: Receive the 25th (Last) Message Block . . . . 6 - 37
6-28. Acknowledgement for the 25th Message Block . . . . 6 - 37
6-29. Final Coordination Information . . . . 6 - 38

Tables

6-30 Permissible Job Requests in ASCII Mode . . . . 6 - 39
6-31 Message: Invalid Job Request (PII) . . . . 6 - 41
6-32 Message: No Receive Message (PII) . . . . 6 - 41
6-33 Message: Error in Receive Message (PII) . . . . 6 - 41
6-34 Status Byte in ASCII Mode (PII) . . . . 6 - 42

6 ASCII Driver

After selection of the ASCII driver, the CP 521 SI enables transfer of message frames between the CPU and a peripheral device connected to the CP 521 SI:

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

You can choose between:
- Transparent ASCII driver
- Interpretive ASCII driver mode I and Interpretive ASCII driver mode II.

Transparent ASCII driver

In transparent mode, the CP 521 SI does not interpret characters.
- XON/XOFF protocol is not possible.
- Only fixed-length message frames can be sent or received.

Interpretive ASCII driver mode I

In interpretive mode I, the CP 521 SI evaluates the following characters:
- RUB OUT ($7F_H$)
- BACKSPACE (08_H)
- XON/OFF characters (if configured)
- Character end code (if configured).

Interpretive ASCII driver mode II

In interpretive mode II, the CP 521 SI evaluates the following characters:
- XON/OFF characters (if configured)
- Character end code (if configured).

The CP 521 SI handles data transfer with the peripheral device autonomously.
The CPU initiates data exchange between the CPU and the CP 521 SI by sending a job request. See Section 6.4 for a precise description of the data exchange procedure.

Note

The following applies to the ASCII driver:
- When sending fixed-length frames (transparent mode) or frames with end-of-text characters (interpretive mode I and II), the frame length and the end character are transmitted to the CP 521 SI along with the send job (Figure 6-6).
- When frames are received, the value programmed for the frame length or the end-of-text character applies (Tables 6-10 and 6-11).
- Special case for receiving fixed-length frames:
  If you program a receive length that is **greater** than the actual length of the receive frame, you can receive frames of variable length in transparent mode by evaluating the message "$X9_H$" ($X9_H$: character delay exceeded, Section 6.5).
- If you program a receive length that is **shorter** than the actual length of the receive frame, the frame will be fragmented. The "remainder frame" is assembled to form a new frame and transmitted as such.

The permissible transmission modes depend on the following:

- Type of serial interface (TTY or RS-232C (V.24))
- Whether handshake has been configured in the case of the RS-232C (V.24) interface (handshake OFF or handshake ON)
- Whether values have been assigned for ”XON” and ”XOFF” (XON/XOFF protocol).

The time of day can be read from the module clock by the CPU also in ASCII driver 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. The ASCII driver does not require a memory submodule.

TTY interface active

In this mode, the CP 521 SI only evaluates the RxD line. When the CP 521 SI sends data to the peripheral device, the peripheral device maintains the RxD line of the CP at logic ”1” as long as no data are being sent to the CP 521 SI. If this is not the case, the CP 521 SI sends the following error messages to the CPU:

In byte 0: ”Permanent wire break” (XD_H)
In byte 1: ”Peripheral device not ready” ($1X_H$)

XON/XOFF protocol:

XON/XOFF protocol in handshake OFF mode is only possible in interpretive ASCII mode. A precondition is that you have configured characters for ”XON” and ”XOFF” in parameter block 2 (Section 6.3).

Both ”XON” and ”XOFF” are control characters used by the CP 521 SI to control data transmission.

The CP 521 SI sends ”XOFF” to the peripheral device if 20 bytes are still free in its receive buffer. The peripheral device is then prompted to send no more data to the CP 521 SI. An overflow in the receive mailbox (1 KB) would cause any data to be lost. Only when the CP 521 SI has more than 256 bytes free in its receive mailbox does it send ”XON” again to the peripheral device.
If the CP 521 SI receives the ”XOFF” character, it ceases sending further data to the peripheral device. Only when it receives the ”XON” character does the CP 521 SI continue its send job request.

RS-232C (V.24) interface in handshake OFF mode

Only the receive and send lines are relevant in handshake OFF mode.

Note

In the case of peripheral devices that do not support hardware handshaking, the CP 521 SI cannot detect failure of the peripheral device if it has been programmed for an RS-232C (V.24) interface.

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

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

Table 6-1. Possible Control Signals of the RS-232C (V.24) Interface in Handshake ON Mode

Control	Status	Meaning
Outputs		
TxD		Send data Send line is maintained at logic ”1” by the CP 521 SI (U - 3 V)
DTR	 ON OFF	**D**ata **t**erminal **r**eady CP 521 SI switched on, ready to receive CP 521 SI not switched on, not ready to receive
RTS	 ON OFF	**R**equest **t**o **s**end CP 521 SI ready to send (U 3 V) CP 521 SI not sending (U - 3 V)
Inputs		
RxD		Receive data Receive line must be maintained at logic ”1” by 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 CP 521 SI The CP 521 SI expects this as response to ”RTS” = ”ON”
	OFF	Peripheral device cannot receive characters from CP 521 SI

Note

XON/XOFF protocol is not possible if you are evaluating control signals (handshake ON).

Data is transferred between the CP 521 SI and the peripheral device as follows:
The CP 521 SI sets output ”DTR”=”ON” after restart. This indicates that the CP 521 SI is ready to operate and to receive.

Example: **CP 521 SI** wants to send data

1. CP waits for ”DSR”=”ON”
   If the peripheral device has not yet set ”DSR” to ”ON after 20 s, the CP reports error $1X_H$ ”Peripheral device not ready” to the CPU in byte 1 of its acknowledgement.
2. CP sets ”RTS”=”ON”.
3. CP waits for ”CTS”=”ON”.
   If the peripheral device has not yet set ”CTS” to ”ON after 20 s, the CP reports error XD_H ”Permanent wire break” to the CPU.
4. CP sends data.
5. CP sets ”RTS”=”OFF” after sending data.
6. Peripheral device sets ”CTS”=”OFF”.

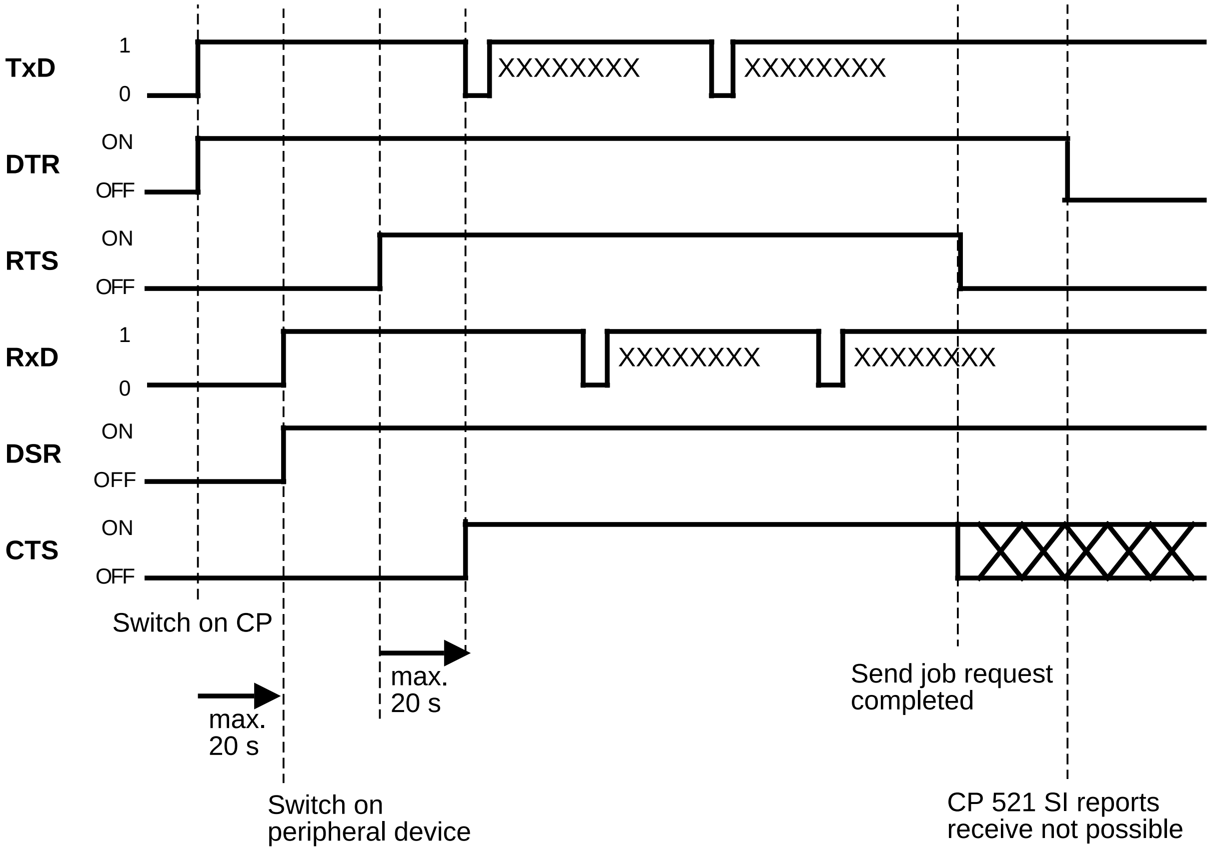


Figure 6-1. Timing Diagram for Data Transfer Between the CP 521 SI and a Peripheral Device

6.1 Prerequisites for Operation with ASCII Drivers

The following conditions must be met to use the CP 521 SI for ASCII mode:

1. Build up connection

Establish the electrical connection between the CP 521 SI and the peripheral device in the POWER OFF state.

2. Settings on the peripheral device

The settings on the peripheral device must agree with the parameter assignment data of the CP 521 SI. If, for example, your peripheral device transfers data at a rate of 2400 bit/s, you must also configure this baud rate on the CP 521 SI.

Note

- See 6.2 for configuration examples with terminal diagrams.
- See 4.3 for details of the serial interface
- See 6.3 for an explanation of assigning CP 521 SI parameters in ASCII mode.

3. Assigning CP 521 SI parameters

There are various ways of assigning the CP 521 SI parameters:

- Transfer the parameter assignment data in the user program after POWER ON or POWER recovery. This can be done after restart. A memory submodule is not necessary.
- Store the parameter assignment data in the memory submodule in DB1.
  You configure the memory submodule with a programmer in off-line mode.

You **must** specify the following parameters:

- 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.
- Parameters for data transfer (message frame length, end-of-text character, ...).

You **can** specify the following parameters:

- Specifications of XON/XOFF protocol (optional)
- Correction value for integral clock (optional)

4. Initializing the CP 521 SI

Plug the configured memory submodule into the CP 521 SI in the POWER OFF state. Then you can switch the CPU to POWER ON.

5. Startup of the CP 521 SI in the ASCII driver mode

The CP 521 SI is automatically in transparent ASCII mode after POWER ON if the following applies:

- A memory submodule is not plugged in.
- A memory submodule is plugged in on which transparent ASCII driver mode is configured.
- Transparent ASCII driver mode has been transferred to the CP 521 SI in the user program.

You can set interpretive ASCII driver mode in the following way:

- Transfer the interpretive ASCII driver mode to the CP 521 SI in the user program.
- Plug a memory submodule into the CP 521 SI which you have configured with the interpretive ASCII driver mode.

Note

If there is no memory submodule plugged into the CP 521 SI and the CP 521 SI has a backup battery, the module is automatically assigned the parameters stored in the RAM on POWER-ON (i.e. the parameters last assigned). In other words, the module is activated in the same driver mode following POWER-ON as it was before POWER-OFF.

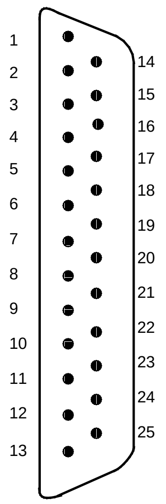
Note

You can change to another mode during operation. For this purpose, you must transfer the ”Transfer parameter assignment data” job request to the CP 521 SI. Remember, however, that you may have to reassign the serial interface parameters. The send buffer and receive buffers will be deleted if you do this.

6.2 Peripheral Interface Connections

The CP 521 SI has a serial interface port. You can choose between a current interface (TTY) or a voltage interface (RS-232C (V.24)) by setting the relevant parameters. The cables of both interfaces connect with a 25-pin subminiature D connector.

Table 6-2. Pin Assignments of the 25-Pin Subminiature D Connector of the CP 521 SI

View	Pin No.	Signal Name	Meaning
	1	-	Disabled
	2	TxD	Send data V.24)
	3	RxD	Receive data V.24)
	4	RTS	Request to send V.24)
	5	CTS	Clear to send V.24)
	6	DSR	Data ready V.24)
	7	GND	Signal ground (RS-232C (V.24))
	8	-	Disabled
	9	TTY IN+	TTY receive line+
	10	TTY IN -	TTY receive line -
	11	-	Disabled
	12	-	Disabled
	13	P24	+24 V for active TTY
	14	-	Disabled
	15	-	Disabled
	16	-	Disabled
	17	20 mA	Current source TTY *****
	18	TTY OUT+	TTY send line+
	19	20 mA	Current source TTY *****
	20	DTR	Terminal ready
	21	TTY OUT-	TTY send line -
	22	-	Disabled
	23	-	Disabled
	24	-	Disabled
	25	-	Disabled

***** If 24 V to GND (pin 7) on pin 13

The following figures show two terminal arrangements. In the ASCII driver mode, the CP 521 SI assumes a peripheral device with an RS-232C (V.24) or TTY interface to be connected to the serial interface.

> **Note**
>
> If you are using an RS-232C (V.24) interface, the CP 521 SI cannot detect failure of the peripheral device in the case of peripheral devices that do not support hardware handshaking.

CP 521 SI (TTY passive) - CP 523 (TTY active)

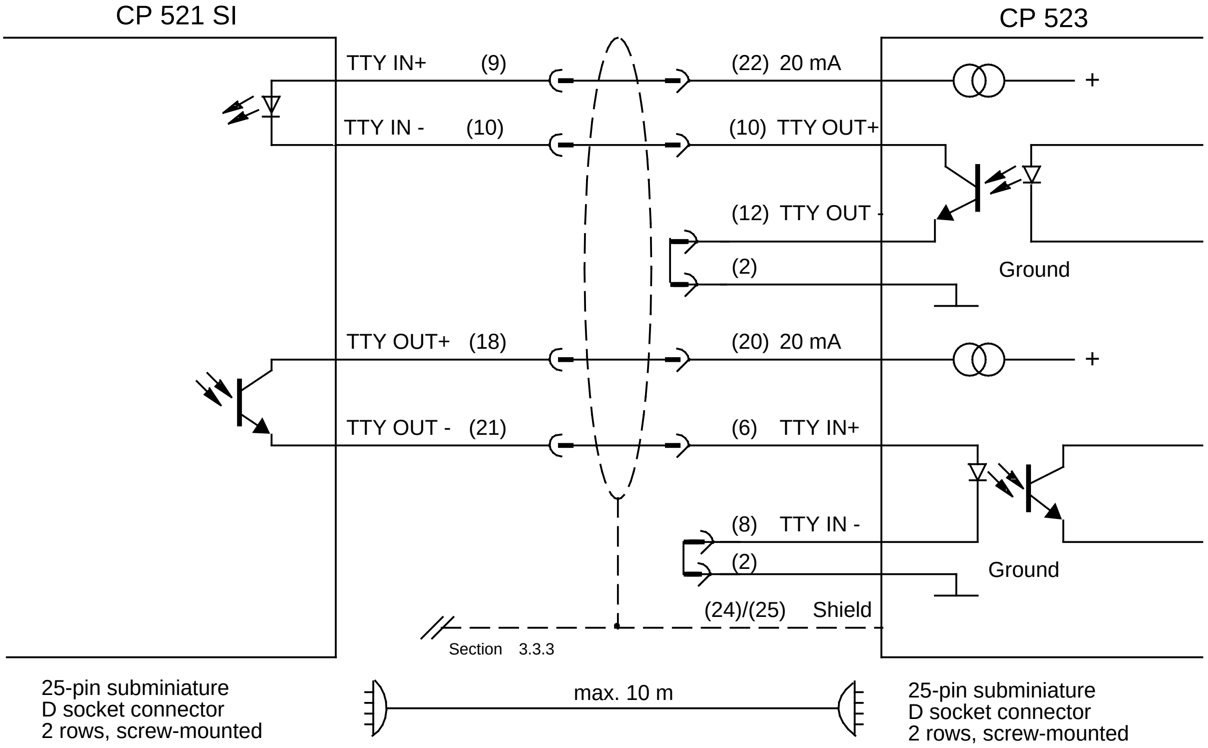


Figure 6-2. Terminal Diagram CP 521 SI (TTY Passive) - CP 523 (TTY Active)

RS-232C (V.24) interface

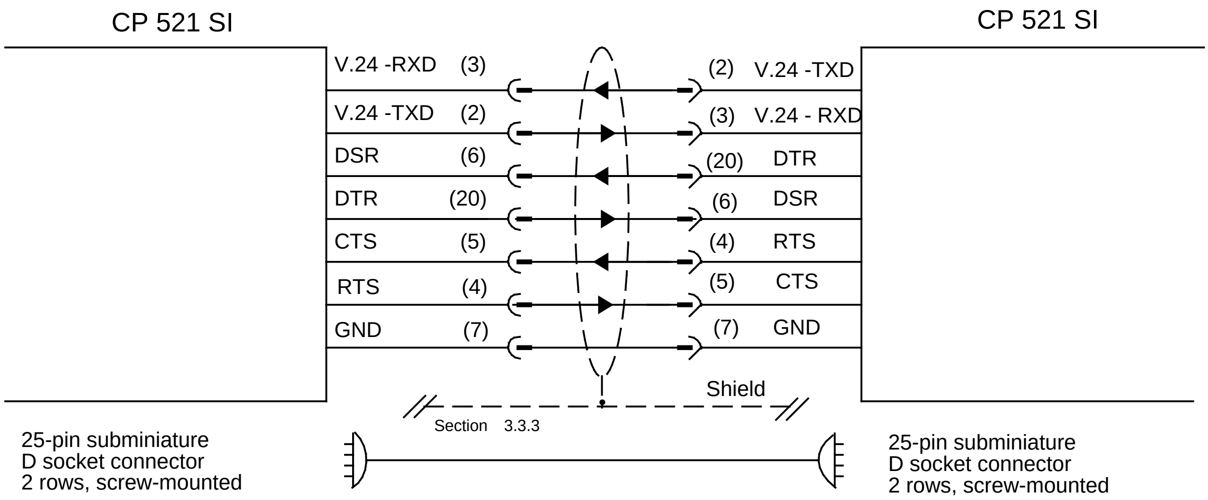


Figure 6-3. Terminal Diagram CP 521 SI - CP 521 SI (RS-232C (V.24) Interface with HW Handshake)

6.3 Assigning the CP 521 SI Parameters for ASCII Mode

The CP 521 SI is supplied with default values for the parameter assignment 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.
- Specifications of XON/XOFF protocol.
- Parameter assignment data for data transfer (message frame length, end-of-text character).
- Correction value for the accuracy of the integral clock.

If you want to assign other parameters to the CP 521 SI, there are various ways of transferring the parameter assignment data to the CP 521 SI:

- Transfer the parameter assignment data to the CP 521 SI in the user program.
  You do not require a memory submodule to operate the CP 521 SI.
- Store the parameter assignment data on a memory submodule in data block (DB) 1 and plug the configured memory submodule into the CP 521 SI (Section 6.3.1).

In order to make transfer of the parameter assignment data as simple as possible, the parameter assignment data is divided into parameter blocks.

Table 6-3. Parameter Blocks for ASCII Mode

Parameter Block Number	Parameters
0	Parameters of the serial interface
2	XON/XOFF protocol (only relevant if XON/XOFF protocol used)
7	Setting of the communications driver
9	Clock correction value

Depending on whether you want to assign your CP 521 SI parameters on a memory submodule or in the user program, read either Section 6.3.1 or 6.3.2.

- Section 6.3.1: Assigning the CP 521 SI parameters with the memory submodule
- Section 6.3.2: Assigning the CP 521 SI parameters in the user program

Note

If there is no memory submodule plugged into the CP 521 SI and the CP 521 SI has a backup battery, the module is automatically assigned the parameters stored in the RAM on POWER-ON (i.e. the parameters last assigned). In other words, the module is activated in the same driver mode following POWER-ON as it was before POWER-OFF.

6.3.1 Assigning the CP 521 SI Parameters with the Memory Submodule

You can store the parameter assignment data in DB1 on a memory submodule.

> **Note**
>
> The procedure for assigning parameters to the memory submodule in the ASCII driver mode is almost identical to the procedure in printer driver mode. The only difference is the assigning parameters to parameter block 7 in which additional data must be specified in the ASCII driver mode.

Entering the parameter blocks on the memory submodule

At the programmer (PG), you must first enter in DB1 all the parameter blocks which deviate from the default values. Then transfer DB1 to the memory submodule.

Further tips for operator inputs:

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

Schematic for entering parameter blocks

The parameter blocks can be entered in DB1 according to the following schematic.

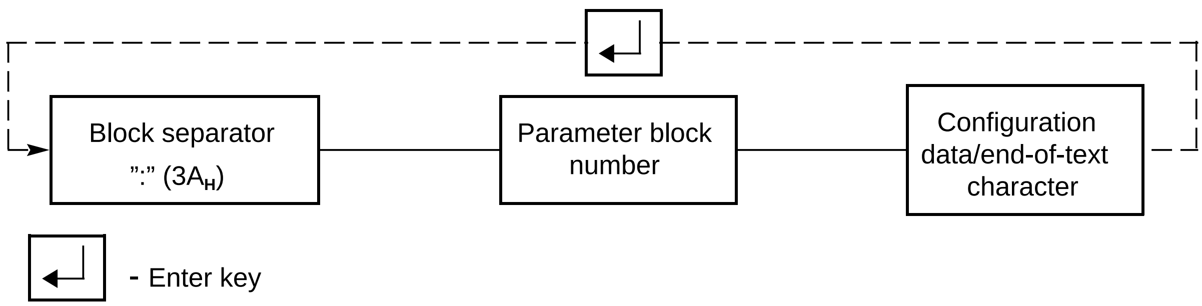


Figure 6-4. Schematic for Entering Parameter Blocks in DB1

The following pages explain the contents of the parameter block and include an example of how to enter the parameters.

Table 6-4. Parameter Block Assignments on the Memory Module

Block	Meaning		Value Range	Default Values on the CP 521 SI
0	Baud rate	110 bit/s 200 bit/s 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s	1 2 3 4 5 6 7 8	8 9600 bit/s
	Parity (Parity bit always ”1”) (Parity bit always ”0”)	even odd ”mark” ”space” any	0 1 2 3 4	0 even
	BUSY signal *	no	0	0 no
	Interface	TTY RS-232C (V.24)	0 1	0 TTY
	Data format 11-bit character frame:	7 data bits (with parity) 8 data bits (with parity) 8 data bits (without parity)	0 1 2	0 7 data bits (with parity)
	10-bit character frame:	7 data bits (without parity) 7 data bits (with parity) 8 data bits (without parity)	3 4 5	
	HW handshake	OFF ON	0 1	0 OFF
2	XON/XOFF protocol	XON character ** XOFF character ** no protocol	00_H to $7F_H$ 00_H to $7F_H$ FF_H	FF_H: no XON/XOFF protocol
7	Setting the mode	Transparent ASCII driver Interpretive ASCII driver mode I Interpretive ASCII driver mode II	1 2 7	0: Memory submodule with message texts plugged in 1: Without memory submodule***
	Character delay		1_D to 65 535_D (·10 ms)	1_D (·10 ms)
	Message frame length		1_D to 256_D	64_D
	End-of-text char. 1 **	ASCII charac. No end-of-text charac.	01_H to $7F_H$ 00_H	00_H
	End-of-text char. 2 **	ASCII charac. No end-of-text charac.	01_H to $7F_H$ 00_H	$0D_H$ (Carriage Return)
9	Clock correction value		- 400_D to $+400_D$ (s/month)	0_D

* BUSY signal irrelevant for ASCII driver

** Only relevant in interpretive mode

*** Only if there is no backup battery; if a backup battery is installed, the same driver in active after POWER-ON as before POWER-OFF.

Table 6-5. Assignments in Parameter Block 7 (Transparent ASCII Driver)

Block	Meaning	Value Range	Default Values on the CP 521 SI
7	ID for "Transparent ASCII driver" mode	1	0: Memory submodule with message texts plugged in 1: Without memory submodule *
	Character delay (decimal)	1_D to 65 535_D (* 10 ms)	1_D (* 10 ms)
	Message frame length (decimal)	1_D to 256_D	64_D

* Only if there is no backup battery; if a backup battery is installed, the same driver is active after POWER-ON as before POWER-OFF

Table 6-6. Assignments in Parameter Block 7 (Interpretive ASCII Driver)

Block	Meaning	Value Range	Default Values on the CP 521 SI
7	ID for "Interpretive ASCII driver mode I" "Interpretive ASCII driver mode II"	 2 7	0: Memory submodule with message texts plugged in 1: Without memory submodule *
	Character delay (decimal)	1_D to 65 535_D (* 10 ms)	1_D (* 10 ms)
	End-of-text char. 1 ASCII char. No end-of-text char.	01_H to $7F_H$ 00_H	00_H
	End-of-text char. 2 ASCII char. No end-of-text char.	01_H to $7F_H$ 00_H	$0D_H$ (Carriage Return)

* Only if there is no backup battery; if a backup battery is installed, the same driver is active after POWER-ON as before POWER-OFF

Note

When operating the CP 521 SI in the interpretive mode of the ASCII driver, the length specification for the message frame is not relevant if the memory submodule is used for parameter assignment. You need not enter any value for the "message frame length" parameter in DB 1.

Explanation of parameters

Note

Your choice of parameter assignment data for the serial interface depends on the interface characteristics of your peripheral device as well as on the specific application.

Baud rate (parameter block 0)

You have a choice of eight baud rates. The default is 9600 bit/s. If you use the RS-232C (V.24) interface, the load capacity of cables longer than 15 m will have a negative effect on the baud rate. Longer cables can be used in general if the baud rate is reduced.

Parity (parameter block 0)

You can choose between five types of parity.

- 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 always has signal state ”1”
- ”Space”
  The parity bit always has signal state ”0”
- No parity check
  The signal state of the parity bit is not significant. Parity is not checked when receiving; however, when sending, the parity bit is always set to ”1”.

The default is even parity.

BUSY signal (parameter block 0)

The BUSY signal is not relevant for the ASCII driver. Leave the default value ”0” (no BUSY signal) at this point in parameter block 0.

Interface (parameter block 0)

You can choose between the RS-232C (V.24) and TTY interfaces here. See Section 2.3 for more detailed information on these interfaces.

The TTY interface is the default.

Data format (parameter block 0)

Characters are transmitted between the CP 521 SI and the peripheral device in a 10-bit or 11-bit character frame. You can choose between seven and eight data bits within these character frames (Figure 4-4, Figure 4-5)

10-bit character frame:

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

11-bit character frame:

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

The 11-bit character frame (1 start bit, 7 data bits, 1 parity bit, 2 stop bits) is the default.

Hardware handshake (parameter block 0)

This parameter is only significant for the RS-232C (V.24) interface. If you set HW handshake”ON”, the ”RTS”, ”CTS”, ”DTR” and ”DSR” control signals of the RS-232C (V.24) interface are evaluated.

The default setting is HW handshake ”OFF”, i.e. the control signals are not evaluated.

XON/XOFF protocol (parameter block 2)

XON/XOFF protocol in handshake OFF mode is only possible in interpretive ASCII mode. A precondition is that you have configured characters for ”XON” and ”XOFF” in parameter block 2.

Both ”XON” and ”XOFF” are control characters used by the CP 521 SI to control data transmission. The CP 521 SI sends ”XOFF” to the peripheral device if 20 bytes are still free in its receive buffer. The peripheral device is then prompted to send no more data to the CP 521 SI. An overflow in the receive buffer (1 byte) would cause any data to be lost. Only when the CP 521 SI has more than 256 bytes free in its receive mailbox does it send ”XON” again to the peripheral device.
If the CP 521 SI receives the ”XOFF” character, it ceases sending further data to the peripheral device. Only when it receives the”XON” character does the CP 521 SI continue its send job request.

Mode (parameter block 7)

You define the desired ”ASCII driver” mode in parameter block 7. You can distinguish between interpretive ASCII mode and transparent ASCII mode.

- Transparent mode
  Set a ”**1**” in parameter block 7 to select transparent ASCII driver mode.

- Interpretive mode I
  Set a ”**2**” in parameter block 7 to select interpretive ASCII driver mode I.

- Interpretive mode II
  Set a ”**7**” in parameter block 7 to select interpretive ASCII driver mode II.

Character delay (parameter block 7)

You can determine yourself the maximum time which is permitted to elapse between two received characters (character delay). The CP 521 SI will then recognize as valid and transfer to the CPU in a message frame only those characters which have a delay within the defined limits. Select as short a character delay as your application will allow but ensure that the character delay is greater than the character transmission time. Approximately 3 1/2 times the character transmission time is recommended.

Message frame length (parameter block 7)

The ”Message frame length” parameter is relevant for transparent ASCII mode when receiving message frames. Receive message frames must have a fixed length in ASCII mode. You can define the message frame length in parameter block 7. Make sure that the same message frame length is set on the CP 521 SI and the peripheral device.

End-of-text character (parameter block 7)

The ”End-of-text character” parameter is only relevant for interpretive ASCII mode. Transmission with end marking is not possible in transparent mode.
You can configure one or two end-of-text characters for transmitting data frames of variable length. Your end-of-text characters limit the length of the data frame in each case. You can send or receive message frames with a length of up to 256 bytes.

Clock correction factor (parameter block 9):

You can configure a correction value in parameter block 9 to improve the accuracy of the module clock. Note that the correction value must be specified with four digits (incl. sign).

The correction value is output in s/month. The month is fixed at 30 days (Section 5.3.5 Clock Correction Factor (parameter block 9)).

Example: Using the PG 750 to configure data on the memory submodule for operation in ASCII mode

The CP 521 SI is installed in slot 7 (starting address 120). The module is to be operated in interpretive ASCII driver mode I and configured as follows:

- Parameters for the serial interface (parameter block 0)
  - 9600 bit/s (8)
  - Even parity (0)
  - No BUSY (0)
  - RS-232C (V.24) interface (1)
  - 8 data bits (1)
  - Handshake OFF (0)

- Parameters for the ASCII driver (parameter block 7)
  - ASCII interpretive driver mode I (2)
  - Character delay: 100 ms (10)
  - 1st end-of-text character: $0D_H$ ($0D_H$)
  - 2nd end-of-text character: $0A_H$ ($0A_H$)

- Parameters for time-of-day correction (parameter block 9)
  - Correction factor: -1s/month (-001)

The end-of-text character (parameter block 3) functions as separator between the various parameters of a parameter block.

We recommend the following procedure:

1. Enter the parameter assignment data on the programmer in DB1 with sufficient comments (KS ...).
2. Store in DB1 on diskette or hard disk.
3. Transfer DB1 to the memory submodule.

Table 6-7. Configuring the CP 521 SI for Interpretive ASCII Driver Mode

Input from PG 750 to DB1	Explanation
0: KS ='Parameters for the ';	
12: KS ='CP521SI ';	
24: KS ='Parameter block_0 ';	9600 baud, even parity, no Busy,
36: KS =':0$800110$ ';	RS232C (V.24), 8 bits, no handshake
48: KS ='Parameter block_7 ';	ASCII mode interpretive
60: KS =':7$2$10$0D$0A$ ';	char. delay=10*10ms, end-of-text char.=0D0A
72: KS ='Parameter block_9 ';	time-of-day correction
84: KS =':9-001 ';	- 1 second / 30 days
96:	

6.3.2 Assigning the CP 521 SI Parameters in the User Program

You can transfer data to the CP 521 SI for **one** parameter block at a time with the $90XX_H$ ”Transfer parameter assignment data” job request.

Contents of the transfer memory when assigning the CP 521 SI parameters in the user program

Write CPU job "Transfer parameter assignment data" to the PIQ:

Request from the CPU to the CP: Accept parameters

Address	Contents
Maddr +0	90_H
Maddr +1	Block number
Maddr+ 2	Parameter
Maddr+ 3	Parameter
Maddr + 4	Parameter
Maddr + 5	Parameter
Maddr + 6	Parameter
Maddr +7	Parameter

Code for "Transfer parameter assignment data"

Evaluate CP response to the "Transfer parameter assignment data" job in the PII:

CPU job accepted
Parameters passed

Address	Contents
Maddr +0	50_H
Maddr +1	00_H
Maddr + 2	irrelevant
Maddr + 3	irrelevant
Maddr + 4	irrelevant
Maddr + 5	irrelevant
Maddr + 6	90_H
Maddr +7	Block number

Acknow-
ledged
job

CP rejects CPU job:
Illegal parameters/CP busy/
wrong job

Contents
40_H
00_H
irrelevant
irrelevant
irrelevant
irrelevant
90_H
Block number

Note

The CPU acknowledgements 5000_H and 4000_H remain (i.e. are not deleted) until they are overwritten by a subsequent job (e.g. blank job 0000_H).

When evaluating data from the CP 521 SI (PII), allow for the fact that you receive the acknowledgement of a CPU job (PIQ) only after two program scan cycles following submission of the job.

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

- At least seven program scan cycles are required for parameter assignment in parameter blocks 0, 2 and 7.

Transferring parameter assignment data for parameter block 0

Table 6-8. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data or Block 0” Job Request

Byte	Meaning	Value Range	Default Values on the CP 521 SI
0	Job request number ”Transfer parameter assignment data”	90_H	-
1	Number of the parameter block	00_H	-
2	Baud rate 110 bit/s 200 bit/s 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s	01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	08_H 9600 Bd
3	Parity even odd (Parity bit always ”1”) ”mark” (Parity bit always ”0”) ”space” any	00_H 01_H 02_H 03_H 04_H	00_H even
4	BUSY signal * no	00_H	00_H no
5	Interface TTY RS-232C (V.24)	00_H 01_H	00_H TTY
6	Data format: 11-bit character frame 1 start bit, 7 data bits, 1 parity bit, 2 stop bits 1 start bit, 8 data bits, 1 parity bit, 1 stop bit 1 start bit, 8 data bits, 2 stop bits 10-bit character frame 1 start bit, 7 data bits, 2 stop bits 1 start bit, 7 data bits, 1 parity bit, 1 stop bit 1 start bit, 8 data bits, 1 stop bit	 00_H 01_H 02_H 03_H 04_H 05_H	00_H 7data bits (with parity)
7	HW handshake OFF ON	00_H 01_H	00_H OFF

***** BUSY signal not relevant for ASCII driver.

Transferring parameter assignment data for parameter block 2

You can assign values for the XON and XOFF characters in parameter block 2.
If you have the XON/XOFF protocol, you have a free choice of the XON/XOFF characters. ASCII code provides the codes 11_H (DC1) for the XON character and 13_H (DC3) for the XOFF character. You must not assign the same values for the ”XON” and ”XOFF” characters.

Table 6-9. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 2” Job Request

Byte	Meaning	Value Range	Default Values on the CP 521 SI
0	Job request 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	$FFFF_H$ (No XON/XOFF protocol)
3	XON/XOFF protocol XOFF character * No protocol	00_H to $7F_H$ FF_H	
4 to 7	Not significant	-	-

* Only relevant in interpretive ASCII mode

Transferring parameter assignment data for parameter block 7

Table 6-10. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 7” Job Request (Transparent ASCII Driver)

Byte	Meaning	Value Range	Default Values on the CP 521 SI
0	Job request number	90_H	-
1	Number of the parameter block and ”Transparent ASCII driver” mode ID	71_H	0: Memory submodule with message texts plugged in 1: Without memory submodule
2+3	Character delay	0001_H to $FFFF_H$ (* 10 ms)	0001_H(* 10 ms)
4+5	Message length (in bytes)	0001_H to $00FF_H$	40_H
6+7	Not significant		

* Only if there is no battery backup; if a backup battery has been installed, the same driver mode is active after POWER-ON as before POWER-OFF.

Table 6-11. Transfer Memory Assignment for the ”Transfer Parameter Assignment Data for Block 7” Job Request (Interpretive ASCII driver)

Byte	Meaning	Value Range	Default Values on the CP 521 SI
0	Job request number	90_H	-
1	Number of the parameter block and mode ID ”Interpretive ASCII driver mode I” ”Interpretive ASCII driver mode II”	 72_H 77_H	0: Memory submodule with message texts plugged in 1: Without memory submodule*
2+3	Character delay	0001_H to $FFFF_H$ (* 10 ms)	0001_H (* 10 ms)
4+5	Message length (in bytes)**	0000_H	irrelevant
6	End-of-text charac. 1 ASCII charac. No EOT charac.	01_H to $7F_H$ 00_H	00_H
7	End-of-text charac. 2 ASCII charac. No EOT charac.	01_H to $7F_H$ 00_H	$0D_H$ (Carriage Return)

* Only if there is no battery backup; if a backup battery has been installed, the same driver mode is active after POWER-ON as before POWER-OFF

** In interpretive mode of the ASCII driver, the end-of-text characters are evaluated when a frame is received.

Note

If the CP 521 SI is operated in the interpretive mode of the ASCII driver, specification of the frame length is irrelevant. Enter the value 0000_H in DB1 for the "Frame length" parameter.

Note on the character delay

You should generally select as short a character delay time as your application will allow. Make sure that the character delay is greater than the character transmission time.

For example, if you have a keyboard connected to the CP 521 SI, you must allow in the character delay for the ”Time for manual key operation” of approximately 1 s. If you link the CP 521 SI to a communications device on which you have defined a baud rate (e.g. another CP 521 SI), you are recommended to set the character delay to 3 1/2 times the character transmission time.

6.4 Data Transfer Between the CP 521 SI and the CPU

Reminder:
Data is transferred between the CPU and the peripheral device in two steps.

- First, data transfer between the CPU and the CP 521 SI is coordinated.
- Only then can the CP 521 SI send data to the peripheral device or receive data from the peripheral device.

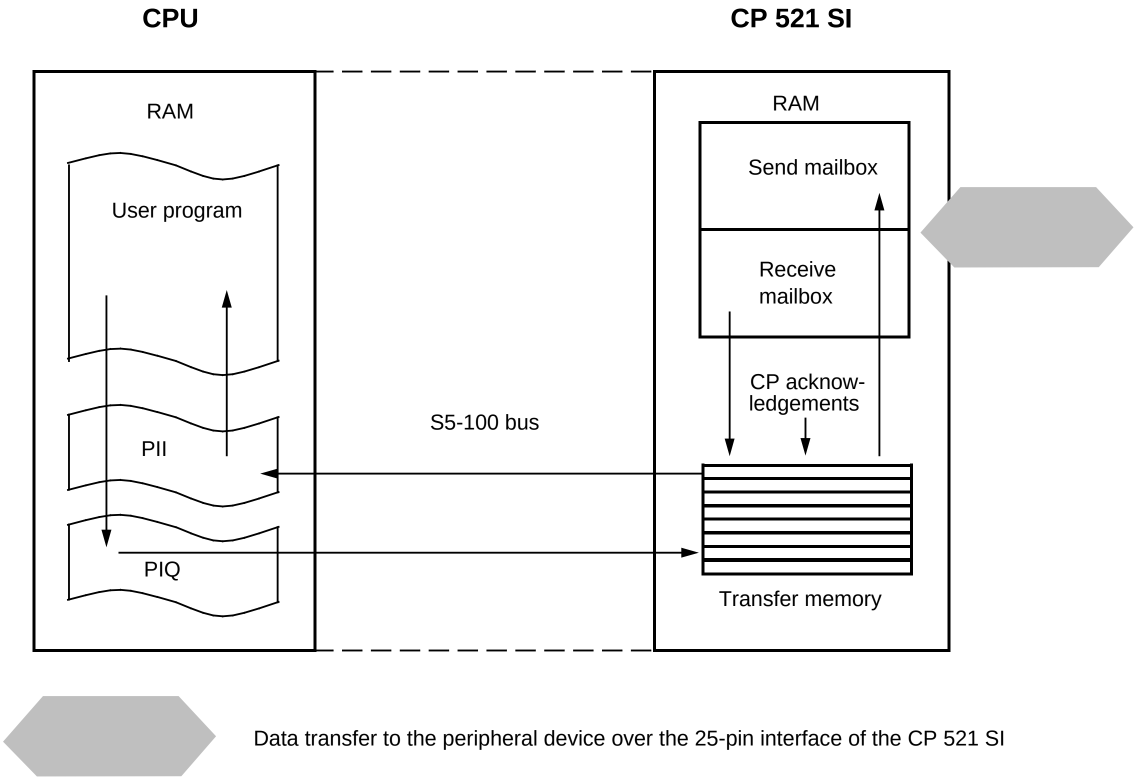


Figure 6-5. Data Interchange over the CP 521 SI

Sending

The CPU transfers message frames with a maximum permissible length of 256 bytes in blocks of eight bytes (6 bytes of useful data) to the CP 521 SI (transfer memory). The transmitted message blocks of the message frames are buffered in the send mailbox of the CP 521 SI. Only when the CP 521 SI has received all message blocks of a message frame does it transfer the whole message frame autonomously over the serial interface to the peripheral device. The send mailbox accommodates 256 bytes and can only hold **one** message frame at a time.

Receiving

The CP 521 SI can receive message frames with a length of 256 bytes from the peripheral device. The receive data is stored in the receive mailbox. The CP 521 SI encodes the received data into message frames and transfers the frames in blocks of eight bytes (6 bytes of useful data) to its transfer memory. The frames are ready to be fetched there by the CPU. The receive mailbox holds 1 KB and can accommodate up to 99 message frames.

6.4.1 Sending Message Frames

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

1. The CPU initiates data transfer with the job request $A001_H$. This job request also defines the message length. If 0 is set, one or two end-of-text characters must be assigned ( ).
2. The CP 521 SI acknowledges receipt of this job request to the CPU ( ).
3. The CPU evaluates the response and begins transferring the current data only if no errors have been detected. Transfer is started with job request $B0_H$ and specification of a block number ( ). Six bytes of data follow. The last message block transferred may contain less than six bytes depending on the send length or the end-of-text character(s).
4. The CP 521 SI writes the block into a send mailbox where it first stores the whole message. The CPU receives an acknowledgement from the CP 521 SI after every eight bytes ( ).
5. The CPU then sends the next message block to the CP, and so on. This data transfer continues until all message blocks of a message frame have been transferred. A message frame can have a maximum length of 256 bytes.
6. After the last message block has been completely received, the CP 521 SI generates a terminating acknowledgement and sends this to the CPU.
7. The CP 521 SI gathers the message blocks together into a message frame and transfers the frame automatically to the peripheral device.

Note

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

Sending message frames with specified lengths

The following is a description of which job requests the CPU uses when prompting the CP 521 SI to send messages and how the CP 521 SI acknowledges these job requests. Table 6-12 contains all the CPU job requests permissible in ASCII mode. After setting the interface parameters (90_H) (Section 6.3), job requests $A0_H$ and $B0_H$ are relevant for sending data.

Table 6-12. Permissible CPU Job Requests when Sending Message Frames (PIQ)

Byte 0	Meaning
00_H	Data not relevant
10_H	Set date and time of day; variables in bytes 1 to 7 (Section 4.4.3)
90_H	Assign interface parameters
$A0_H$	Coordinate data transfer
$B0_H$	Process data transfer CPU CP 521 SI

The schematic on the following pages shows the overall data transfer procedure for sending data. This is followed by a description of the steps shown.

Message blocks of eight bytes as used for message frames are represented. The CPU job request varies from block to block, as does the corresponding acknowledgement from the CP 521 SI.

Data is transferred over the CP 521 SI in two steps:

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

Data transfer: CPU   CP 521 SI

CPU   CP 521 SI

Byte	0	1	2	3	4	5	6	7
	$A0_H$	01_H	nn_H	nn_H	nn_H	nn_H	xx_H	xx_H
	Coord. job request	Coord. byte: SEND	Send length		End-of-text character		Irrelevant	

CPU   CP 521 SI

50_H	01_H	00_H	00_H	nn_H	xx_H	xx_H	xx_H
Coord. info.	Coord. byte	Data valid	Irrel.	Number of frames	Irrelevant		

CPU   CP 521 SI

$B0_H$	01_H	nn_H	nn_H	nn_H	nn_H	nn_H	nn_H
Data trans.	Block number	Data					

CPU   CP 521 SI

50_H	01_H	00_H	01_H	nn_H	xx_H	xx_H	xx_H
Coord. info.	Coord. byte	Data valid	Block number	Number of frames	Irrelevant		

Figure 6-6. Sequence Schematic for Sending Data

CPU CP 521 SI

$B0_H$	02_H	nn_H	nn_H	nn_H	nn_H	nn_H	nn_H
Data trans.	Block number	Data					

CPU CP 521 SI

50_H	01_H	00_H	02_H	nn_H	xx_H	xx_H	xx_H
Coord. info	Coord. byte	Data valid	Block number	Number of frames	Irrelevant		

CPU CP 521 SI

$B0_H$	nr_H	nn_H	nn_H	nn_H	nn_H	nn_H	nn_H
Data trans.	Block number	Data					

CPU CP 521 SI

50_H	01_H	00_H	nr_H	nn_H	xx_H	xx_H	xx_H
Coord. info		Data valid	Block number	Number of frames	Irrelevant		

... after the data has been sent to the peripheral device ...

CPU CP 521 SI

50_H	00_H	00_H	xx_H	nn_H	xx_H	xx_H	xx_H
Coord. info	Coord. byte reset	Data valid	Irrele- vant	Number of frames	Irrelevant		

Figure 6-6. Sequence Schematic for Sending Data (Continued)

The following is a more detailed explanation of steps   to .

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

To coordinate data transfer, job request $A0_H$ is written into byte 0 of the PIQ.

This defines the meaning of bytes 1 to 7.

Table 6-13. Coordination Job Request

Byte	Value	Meaning
0	$A0_H$	Job request: Coordinate data transfer
1	01_H	Send message
2 3	01_H 00_H	Send length in bytes; (here: 0100_H=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.

Bytes 2 and 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 521 SI evaluates the end-of-text character. If both send length and end-of-text character are 0, there is a job request error.

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

CP response to job request $A001_H$

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

Table 6-14. Coordination Information

Byte	Value	Meaning
0	50_H	Acknowledgement: Coordination information
1	01_H	”Send” coordination bit set
2	00_H 01_H 81_H	Coordination data valid Error in data transfer between CP and CPU Previous send job request not yet completed
3	00_H	Irrelevant
4	00_H to 63_H	Number of messages in receive mailbox of the CP (messages from the peripheral devices) to be read by the CPU (max. 99 messages)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Byte 0: Acknowledge job request: Coordinate ”Send” data transfer.

Byte 1: ”Send” coordination bit set. If the coordination data is invalid, the CP 521 SI 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. 99_D) are stored in the receive mailbox of the CP 512 SI.

As soon as data transfer has been coordinated (and), data transfer begins with job request $B0_H$ (to).

> **Note**
>
> The CP 521 SI usually transfers the date and time of day to the CPU in bytes 1 to 7 (Section 4.4). If byte 0 has the value 50_H or 60_H, you must not interpret these values as the time of day.

CPU job request: ”Send data” ($B0_H$)

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

Table 6-15. Data Transfer: Sending the 1st Message Block

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

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

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

CP response to job request $B001_H$

The CP 521 SI acknowledges the ”Send 1st message block of the message” job request as follows (PII):

Table 6-16. Acknowledging the 1st Message Block

Byte	Value	Meaning
0	50_H	Acknowledgement of job request: Send data
1	01_H	”Send” coordination bit set if data valid
2	00_H 01_H 81_H	Data valid Error in data transfer between CP and CPU Previous send job request not yet completed
3	01_H	Number of 1st message block
4	00_H - 63_H	Number of messages in receive buffer (max. 99)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

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

Byte 2: Indicates that the coordination data is valid or invalid.

Byte 3: Specifies the number of the message block that the CP 521 SI has just received (1 in this case).

Note

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

If you send the next job request $B0_H$ with incremented byte 1 ( ), the CP 521 SI acknowledges with incremented byte 3 ( ).

CPU job request: $B002_H$

Table 6-17. Data Transfer: Sending the 2nd Message Block

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

CP response to job request B002$_H$

Table 6-18. Acknowledging the 2nd Message Block

Byte	Value	Meaning
0	50$_H$	Acknowledgement for job request: Send data
1	01$_H$	"Send" coordination bit set if data valid
2	00$_H$ 01$_H$ 81$_H$	Coordination data valid Error in data transfer between CP and CPU Previous send job request not yet completed
3	02$_H$	Number of the 2nd message block
4	00$_H$ to 63$_H$	Number of messages in the receive buffer (max. 99)
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 and the CP acknowledgement appear as follows:

CPU job request: B02B$_H$

Table 6-19. Data Transfer: Sending the 43rd Message Block

Byte	Value	Meaning	
0	B0$_H$	Job request:	Send data
1	2B$_H$		Message 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 message blocks in one message (42 of 6 bytes and 1 of 4 bytes). This leaves only four bytes for data in the last message block (2B$_H$).

CP response to job request $B02B_H$

Table 6-20. Acknowledging the Last (43rd) Message Block

Byte	Value	Meaning
0	50_H	Acknowledgement for job request: Send data
1	01_H	”Send” coordination bit set if data valid
2	00_H 01_H 81_H	Coordination data valid Error in data transfer between CP and CPU Previous send job request not yet completed
3	$2B_H$	Number of the 43rd message block (max.)
4	00_H to 63_H	Number of messages in receive buffer (max. 99)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Final CP response

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

Table 6-21. Final Coordination Information

Byte	Value	Meaning
0	50_H	Acknowledging the job request: Send data
1	00_H	Coordination bit reset
2	00_H 01_H 81_H	Coordination data valid Error in data transfer between CP and CPU Previous send job request not yet completed
3	00_H	Irrelevant
4	00_H to 63_H	Number of messages in the receive buffer (max. 99)
5	Irrelevant	
6	Irrelevant	
7	Irrelevant	

Sending messages with end-of-text character

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

Table 6-22. Coordination Job Request

Byte	Value	Meaning
0	$A0_H$	Job request: Coordinate data transfer
1	01_H	Send message frame (send bit)
2 3	00_H 00_H	Message length (here: 0000_H=Send with end-of-text character)
4	00_H	End-of-text character (only if message length=0)
5	03_H	End-of-text character (only if message length=0)
6	Irrelevant	
7	Irrelevant	

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

Bytes 4 and 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 (byte 4=00_H).

Note

If the message is longer than 256 data bytes (>0100_H), the CP 521 SI responds with 5000_H and then with message $4X_H$ ”Job request error”. The CP 521 SI 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 rejected.

6.4.2 Receiving Message Frames

Receiving message frames is handled autonomously by the CP 521 SI.
To receive data properly, the data sent from the peripheral device must agree with the parameter assignment of the CP 521 SI, especially parameter block 7.

The peripheral device sends message frames to the CP 521 SI. The messages are gathered in a receive mailbox in the CP 521 SI. This receive mailbox has a capacity of 1 KB and can store 99 messages.

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

1. The CPU sends a job request: Coordinate data transfer ”Receive”, to the CP 521 SI.

2. The CP 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 SI sends a further block, and so on until the whole message has been transferred from the CP 521 SI to the CPU.

Note

If the time between two received characters is greater than the value you have set for the character delay (ZVZ), the characters received up to this point are valid and are transferred to the CPU as a message frame.

The CPU receives an immediate final acknowledgement 5000_H to the ”Coordinate receive” job request ($A080_H$) if there is no receive message.

Receiving fixed-length messages

If you want to receive fixed-length message frames from a peripheral device using the CP 521 SI, 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 job requests the CPU uses to allow the CP 521 SI to receive data from peripheral devices and how the CP 521 SI acknowledges these job requests. After you have set the interface parameters (90_H) (Section 6.3), job requests A0 and C0 are relevant for receiving data (Table 6-23).

Table 6-23. 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

Figure 6-7 illustrates the whole data transfer sequence when receiving message frames. This is followed by a description of the steps shown in the figure.

Message blocks of eight bytes each, as are used when receiving messages, are represented. The CPU job request and the relevant CP 521 SI acknowledgement alternate from block to block.

Data is exchanged via the CP 521 SI in two steps:

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

Data transfer: CP 521 SI CPU

CPU CP 521 SI

Byte	0	1	2	3	4	5	6	7
	$A0_H$	80_H	xx_H	xx_H	xx_H	xx_H	xx_H	xx_H
	Coord. job request	Coord. byte REC.	Irrelevant					

CPU CP 521 SI

60_H	01_H	nn_H	nn_H	nn_H	nn_H	nn_H	nn_H
Data trans.	Block number	Data					

CPU CP 521 SI

$C0_H$	01_H	xx_H	xx_H	xx_H	xx_H	xx_H	xx_H
Ack.	Block number	Irrelevant					

CPU CP 521 SI

60_H	nr_H	nn_H	nn_H	nn_H	nn_H	nn_H	nn_H
Data trans.	Block number	Data					

CPU CP 521 SI

$C0_H$	nr_H	xx_H	xx_H	xx_H	xx_H	xx_H	xx_H
Ack.	Block number	Irrelevant					

CPU CP 521 SI

50_H	00_H	00_H	00_H	nn_H	nn_H	xx_H	xx_H
Coord. info.	Coord. byte	Data valid	Irrel.	Number of frames	Bytes in last block	Irrelevant	

Figure 6-7. Sequence Schematic when Receiving Data

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

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

Table 6-24. Coordination Job Request

Byte	Value	Meaning
0	$A0_H$	Job request: 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 a message frame by setting bit 7=1. This initiates data transfer. The CP 521 SI resets this receive bit after a complete receive message has been sent to the CPU.

CP response to job request $A080_H$

If the CP 521 SI receives job request $A0_H$ (Coordinate data transfer) with receive bit 80_H (Receive message) set, and if there are no errors, the CP 521 SI acknowledges the CPU job request with 60_H in byte 0:

Table 6-25. Data Transfer: Receive 1st Message Block

Byte	Value	Meaning
0	60_H	Data transfer from the CP 521 SI to the CPU
1	01_H	Number of the 1st message block
2		
3		
4		Data
5		
6		
7		

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

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

CPU acknowledgement ($C0_H$)

The CPU acknowledges entry of the message block with "$C0_H$" in byte 0:

Table 6-26. Acknowledging the 1st Message Block

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

Byte 1: Number of the message 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 SI continues to send a message block to the CPU until it receives acknowledgement. For this reason, the CPU receives the data of the acknowledged block from the CP 521 SI once more within the same data cycle in which it was acknowledged. The CPU program must take account of this or the CP 521 SI will receive an acknowledgement with the wrong block number, and data transfer will be aborted.

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

CP response to the CPU acknowledgement $C0_H$

Table 6-27. Data Transfer: Receive the 25th (Last) Message Block

Byte	Value	Meaning
0	60_H	Data transfer from the CP 521 SI to the CPU
1	19_H	Number of the last message block: here 25_D
2		
3		
4		D a t a
5		
6		
7		

CPU acknowledgement for the 25th message block

Table 6-28. Acknowledgement for the 25th Message Block

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

Final response from the CP 521 SI

Table 6-29. Final Coordination Information

Byte	Value	Meaning
0	50_H	Coordination information
1	00_H	Coordination bit reset
2	00_H 01_H 02_H	Coordination data valid Error in data transfer between CP and CPU No receive message frame
3	00_H	Irrelevant
4	00_H to 63_H	Number of messages in receive mailbox
5	01_H to 06_H	Number of bytes in the last block
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 SI

Byte 5: Number of valid bytes in the last message block received

6.5 CPU Job Requests and CP Error Messages

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 SI acknowledges CPU job requests.

Permissible CPU job requests in ASCII driver mode

Table 6-30. Permissible Job Requests in ASCII Mode

Byte 0	Byte 1	Job Requests
00_H	00_H	Blank job request: Display current time-of-day data (Section 4.4)
10_H	00_H	Set clock (Section 4.4.3)
90_H	00_H	Transfer parameters : Parameter block 0
90_H	20_H	” ” : ” 2
90_H	$7x_H$	Transfer parameters : ” 7
90_H	71_H	Setting : Transparent ASCII mode x=1
90_H	72_H	Setting : Interpretive ASCII mode I x=2
90_H	77_H	Interpretive ASCII mode II x=7
$A0_H$	01_H	Coordinate ”Send” data transfer
$A0_H$	80_H	Coordinate ”Receive” data transfer
$B0_H$	Block No.	Coordinate ”CPU CP 521 SI” data transfer
$C0_H$	Block No.	CPU acknowledgement: Received data transferred
$D0_H$	00_H	Delete receive mailbox contents

Additional CPU job "Delete receive mailbox contents"

It is often relevant to evaluate a current message frame. With the job "Delete receive mailbox contents" ($D0_H$) you have now the possibility to delete the receive mailbox before receiving the relevant message frame.

Note

When executing the job "Delete receive mailbox contents", the serial interface of the CP 521 SI is briefly disabled. If the peripheral device connected continues to send data, the first message frame received by the CP 521 SI can be faulty or incomplete.

The interface to the CPU is also briefly occupied by the CP checkback signal "$0F_H$". Therefore **always** evaluate the CP acknowledgement to the job "Delete receive mailbox contents" before you send further CPU job requests to the CP 521 SI.
The CP 521 SI acknowledges the job "Delete receive mailbox contents" like all parameterization jobs with the terminating message "5000_H" **and** specification of the job acknowledged (here: "$D000_H$") in bytes 6 and 7 of the CP response.

Illegal job requests with error messages

If data has **not** yet been transferred and you write a job request in word 0 of the PIQ (byte 0 and byte 1) other than the job executed above, the CP 521 SI stores an error message in the status byte (PII).

The following are illegal job requests in ASCII mode which are acknowledged with the error message $4X_H$:

- All job requests concerned exclusively with print mode (20_H, 30_H, 40_H, 50_H, 60_H, 70_H, 80_H).
- All job request numbers outside the permissible value range for CPU job requests ($D1_H$ to FF_H).
- Job request number $90XX_H$ if the parameter blocks are configured with data from outside the permissible value range; if a parameter assignment job contains errors, the CP 521 SI writes the rejected job request into word 6 (bytes 6 and 7) of the CP acknowledgement (Section 6.3.2).
- Job request number $B0XX_H$ and $C0XX_H$ if data transfer has not previously been initiated.

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

Table 6-31. Message: Invalid Job Request (PII)

Byte	Value	Meaning
0	50_H	Coordination information
1	00_H	Coordination bit ”Send”/”Receive” reset
2	01_H	Invalid job request/data illegal/data transfer aborted
4	nn_H	Number of messages in receive buffer
3, 5 to 7	Irrelevant	

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

The CP 521 SI aborts a receive job request ($A080_H$) immediately if

- there is no message present in the CP 521 SI (CP acknowledgement 5000_H) or
- there is a receive message error in the CP 521 SI.

Exception: Receive messages with error $X9_H$ (character delay exceeded) are sent to the CPU. All characters received correctly in the CP 521 SI before expiry of the character delay are sent to the CPU.

Table 6-32. Message: No Receive Message (PII)

Byte	Value	Meaning
0	50_H	Coordination information
1	00_H	”Receive” coordination bit reset
2	02_H	No message
4	00_H	Number of messages in receive buffer
3, 5 to 7	Irrelevant	

Table 6-33. Message: Error in Receive Message (PII)

Byte	Value	Meaning
0	$5X_H$	Coordination information ”X”: Cause of error (Table 6-34) Exception: character delay exceeded
1	00_H	”Receive” coordination bit reset
2	01_H	Error in data transfer between CP and CPU
4	nn_H	Number of messages in the receive buffer
3, 5 to 7	Irrelevant	

The following error messages may appear **after** transmission of the ”Coordinate data transfer” coordination job request.

Table 6-34. Status Byte in ASCII Mode (PII)

Byte 0 Bit 4 to 7	Bit 0 to 3	Message
X	9	Character delay time exceeded
X	A	Parity error
X	B	Receive after XOFF or receive after DTR=”OFF”
X	C	Message length greater than 256 bytes
X	E	Receive mailbox overflow

X = Signal state not significant for the other nibble

Note

The CP response remains (i.e. is not deleted) until it is overwritten by a subsequent job (e.g. blank job 0000_H)

Character delay 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 transferred to the CPU and is not stored in the receive mailbox.

Receive after XOFF or receive after DTR=OFF (XB_H)

In the case of XON/XOFF protocol:
The CP 521 SI sends XOFF to the peripheral device if
- less than 20 bytes are free in the receive mailbox
  or
- the CP 521 SI has received more than 99 messages.

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

When using control signals (handshake ON):
The CP 521 SI sends DTR=”0” to the peripheral device if
- less than 20 bytes are free in the receive mailbox
  or
- the CP 521 SI has received more than 99 messages.

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

Error message XB_H is output if the peripheral device has sent more characters than the CP 521 SI can accommodate 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 521 SI outputs error message XC_H if it has not detected an end-of-text character after receiving 256 characters.
The message is not transferred to the CPU and not stored in the receive mailbox.

Receive mailbox overflow (XE_H)

The following can be stored in the receive mailbox:

- Up to 1024 bytes of data
- Up to 99 messages.

If these values are exceeded when receiving a message from the peripheral device, the CP 521 SI outputs an error message.
The message is not transferred to the CPU and is also not stored in the receive mailbox.

6.6 STEP 5 Programs for Data Transmission with ASCII Driver

STEP 5 program differences between

- Interpretive ASCII driver
  and
- Transparent ASCII driver

Receiving messages

The STEP 5 program for receiving messages with end-of-text characters is no different to a STEP 5 program for receiving messages of a fixed length. You must only take the following into account:

- Messages with end-of-text characters can only be evaluated correctly by the interpretive ASCII driver.
- Senders and receivers must use the same characters as end-of-text characters, otherwise ”Character delay exceeded” is reported
  (parameter block 7).

Sending messages

The STEP 5 program for sending messages must do the following depending on the driver set.
Either:

- Write the send length into bytes 2 and 3 of the transfer memory (possible both in interpretive ASCII driver mode and transparent ASCII driver mode; bytes 4 and 5 are then irrelevant)
  or
- Write the end-of-text characters into bytes 4 and 5 of the transfer memory (possible only in interpretive ASCII driver mode; bytes 2 and 3 must have the value 00_H!).

The following is a complete STEP 5 program for data transmission with the ASCII driver. The program has the following structure:

- OB1:
  - Invokes the parameter assignment FB (default: FB3; if the CP 521 SI is to be assigned its parameters with FB4, FB5 and FB6, you must replace FB3 by the corresponding FB (FB4, FB5 or FB6) in the program listing.) If the CP 521 SI has been assigned its parameters, this FB is no longer invoked.
  - Invokes FB21, which, in turn, invokes either the Send FB (FB200, FB11, FB12) or the Receive FB (FB201, FB13).
- OB21/22:
  - Defaults for the parameter assignment FB and the distributor FB21

- FB3/FB4: Parameter assignment FBs (user-configurable; can only be used for CPU 103/S5-95U PLC)
  ASCII driver, transparent (FB3)/ASCII driver, interpretive (FB4)
- FB5/FB6: Parameter assignment FBs (module starting address=104)
  ASCII driver, transparent (FB5)/ASCII driver, interpretive (FB6)
- FB11: Send FB (sending with fixed length; module starting address=104)
- FB12: Send FB (sending with end-of-text character, module starting address=104)
- FB13: Receive FB (module starting address=104)
- FB20: Pulse generation for send or receive initiation
- FB21: Switching between send and receive
- FB200: Send FB (user-configurable; can only be used for CPU103/S5-95U PLC)
- FB201: Receive FB (user-configurable; can only be used for CPU103/S5-95U PLC)
- DB21: Send data (source DB)
- DB22: Receive data (destination DB)

This list does not include the source and destination DBs for Send and Receive data, which you must have installed with a suitable length before starting the program.

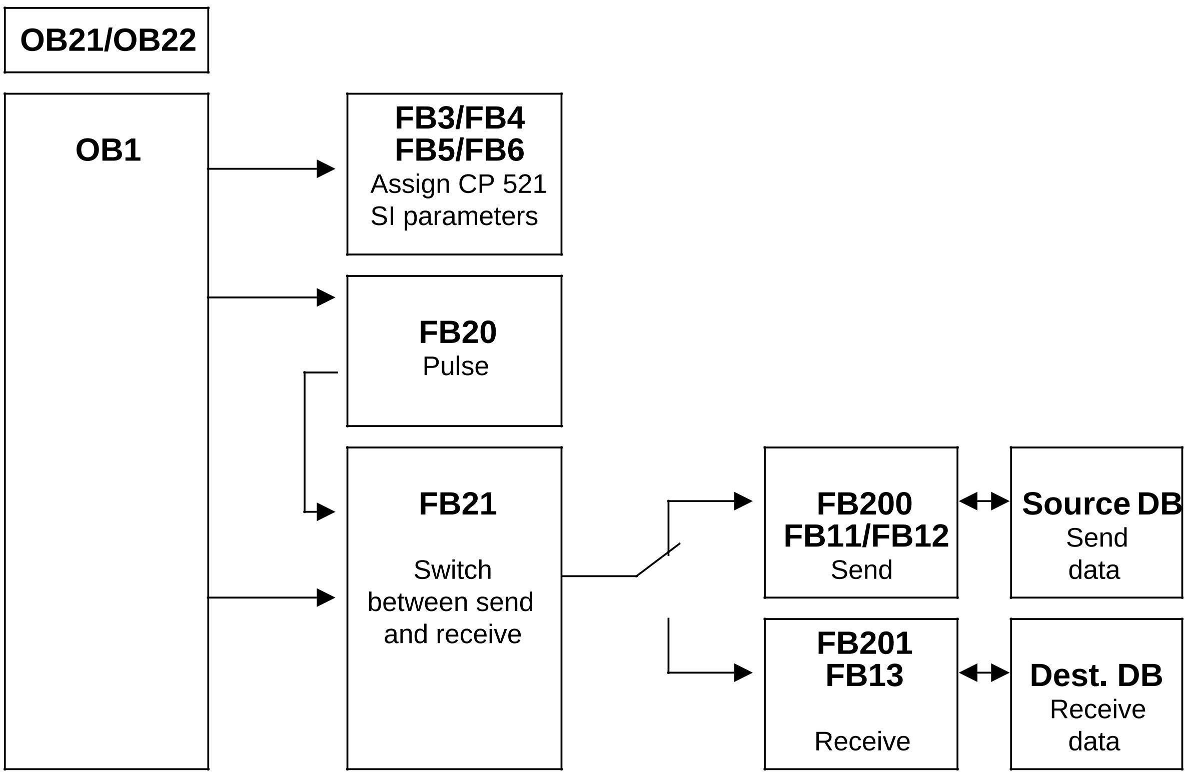


Figure 6-8. Structure of the STEP 5 Program for Data Transmission with ASCII Driver

```
OB 1                                   B:ASCII@ST.S5D             LEN=19

        SEGMENT  1     0000
        0000      :AN  F  100.7
        0001      :JC  FB   3
        0002 NAME :CP-PARAM
        0003 BGAD :    KF +104
        0004 OK   :    F  100.7
        0005      :AN  F  100.7
        0006      :BEC
        0007      :
        0008      :JU  FB  20
        0009 NAME :PULSE
        000A      :JU  FB  21
        000B NAME :DISTRIB
        000C      :
        000D      :BE

OB 21                                  B:ASCII@ST.S5D             LEN=10

        SEGMENT  1     0000
        0000      :L   KH 0000                      Defined default for
        0002      :T   FY  80                       distribution FB and
        0003      :T   FM 100                       parameter assignment FB
        0004      :BE

OB 22                                  D:ASCII@ST.S5D              LEN=8

        SEGMENT  1     0000
        0000      :JU  OB  21
        0001      :
        0002      :BE
```

```
FB 3                                  D:ASCII@ST.S5D            LEN=130

        SEGMENT  1     0000
        NAME :CP-PARAM                                  ASCII driver, transparent
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :OK        I/Q/D/B/T/C: Q  BI/BY/W/D: BI

        000B      :LW  =BGAD                           This FB uses flag words
        000C      :T   FW 102                          FW 100 to 108 to assign
        000D      :I        2                          the CP 521 SI
        000E      :T   FW 104                          parameters.
        000F      :I        2                          FW 102 to FW 108 are
        0010      :T   FW 106                          addresses of the CP.
        0011      :I        2                          FW 100 is used for execution
        0012      :T   FW 108                          of the FB
        0013      :
        0014      :DO  FW 102                          Interrogation: Is CP still
        0015      :L   IW   0                          in restart routine?
        0016      :L   KH 0F00
        0018      :AW
        0019      :L   KH 0F00
        001B      :!=F
        001C      :BEC
        001D      :***


        SEGMENT  2     001E
        001E      :A   F  100.0                        Assign parameters to first block
        001F      :JC  =M001
        0020      :L   KH 9000                         Job block 0
        0022      :DO  FW 102
        0023      :T   QW   0
        0024      :L   KH 0800                         Baud rate    Parity
        0026      :DO  FW 104
        0027      :T   QW   0
        0028      :L   KH 0001                         BUSY         Interface
        002A      :DO  FW 106
        002B      :T   QW   0
        002C      :L   KH 0100                         Data format  Handshake
        002E      :DO  FW 108
        002F      :T   QW   0
        0030      :
        0031      :DO  FW 102                          Wait for positive
        0032      :L   IW   0                          acknowledgement 5XXXH
        0033      :L   KH 5000
        0035      :AW
        0036      :L   KH 5000
        0038      :><F
        0039      :BEC
        003A      :DO  FW 108                          Check whether acknowledgement
        003B      :L   IW   0                          belongs to job, otherwise wait
        003C      :L   KH 9000                         for the right acknowledgement
        003E      :><F
        003F      :BEC
        0040 M001 :AN  F  100.0                        Set step flag
        0041      :S   F  100.0
        0042      :***
```

```
SEGMENT  3     0043
0043      :A   F  100.1                              Assign parameters to second block
0044      :JC  =M001
0045      :L   KH 9071                               Block 7    ASCII transparent
0047      :DO  FW 102
0048      :T   QW   0
0049      :L   KF 0010                               Character delay (100ms)
004B      :DO  FW 104
004C      :T   QW   0
004D      :L   KH 0100                               Receive length
004F      :DO  FW 106
0050      :T   QW   0
0051      :L   KH 0000                               0 default
0053      :DO  FW 108
0054      :T   QW   0
0055      :
0056      :DO  FW 102                                Wait for positive
0057      :L   IW   0                                acknowledgement 5XXXH
0058      :L   KH 5000
005A      :AW
005B      :L   KH 5000
005D      :><F
005E      :BEC
005F      :DO  FW 108                                Check whether acknowledgement
0060      :L   IW   0                                belongs to job, otherwise wait
0061      :L   KH 9071                               for right acknowledgement
0063      :><F
0064      :BEC
0065 M001 :AN  F  100.1                              Set step flag
0066      :S   F  100.1
0067      :
0068      :***


SEGMENT  4     0069
0069      :A   F  100.2                              Last block deletes
006A      :JC  =M001                                 acknowledgement 5000H.
006B      :L   KH 0000
006D      :DO  FW 102
006E      :T   QW   0
006F      :
0070      :DO  FW 102
0071      :L   IW   0
0072      :L   KH 5000
0074      :AW
0075      :L   KH 0000
0077      :><F
0078      :BEC
0079 M001 :AN  F  100.2
007A      :S   F  100.2
007B      :S   =OK                                   Set parameter assignment flag
007C      :BE
```

```
FB 4                                  D:ASCII@ST.S5D              LEN=130

        SEGMENT  1     0000
        NAME :CP-PARAM                                  ASCII driver, interpretive
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :OK        I/Q/D/B/T/C: Q  BI/BY/W/D: BI

        000B      :LW  =BGAD                           This FB uses flag words FW 10
        000C      :T   FW 102                          to 108 to assign the
        000D      :I        2                          CP 521 SI
        000E      :T   FW 104                          parameters.
        000F      :I        2                          FW 102 to FW 108 are addresses
        0010      :T   FW 106                          of the CP.
        0011      :I        2                          FW 100 is used for execution
        0012      :T   FW 108                          of the FB.
        0013      :
        0014      :DO  FW 102                          Interrogation: Is CP still in
        0015      :L   IW   0                          initial start status?
        0016      :L   KH 0F00
        0018      :AW
        0019      :L   KH 0F00
        001B      :!=F
        001C      :BEC
        001D      :***


        SEGMENT  2     001E
        001E      :A   F  100.0                        Assign parameters to first block
        001F      :JC  =M001
        0020      :L   KH 9000                         Job block 0
        0022      :DO  FW 102
        0023      :T   QW   0
        0024      :L   KH 0800                         Baud rate    Parity
        0026      :DO  FW 104
        0027      :T   QW   0
        0028      :L   KH 0001                         BUSY         Interface
        002A      :DO  FW 106
        002B      :T   QW   0
        002C      :L   KH 0100                         Data format  Handshake
        002E      :DO  FW 108
        002F      :T   QW   0
        0030      :
        0031      :DO  FW 102                          Wait for positive
        0032      :L   IW   0                          acknowledgement 5XXXH
        0033      :L   KH 5000
        0035      :AW
        0036      :L   KH 5000
        0038      :><F
        0039      :BEC
        003A      :DO  FW 108                          Check whether acknowledgement
        003B      :L   IW   0                          belongs to job, otherwise wait
        003C      :L   KH 9000                         for the right acknowledgement
        003E      :><F
        003F      :BEC
        0040 M001 :AN  F  100.0                        Set step flag
        0041      :S   F  100.0
        0042      :***
```

```
SEGMENT  3     0043
0043      :A   F  100.1                    Assign parameters to second block
0044      :JC  =M001
0045      :L   KH 9072                     Block 7    ASCII interpretive mode I
0047      :DO  FW 102
0048      :T   QW   0
0049      :L   KH 0010                     Character delay
004B      :DO  FW 104
004C      :T   QW   0
004D      :L   KH 0000                     Receive length
004F      :DO  FW 106
0050      :T   QW   0
0051      :L   KH 000D                     Receive end-of-text character
0053      :DO  FW 108
0054      :T   QW   0
0055      :
0056      :DO  FW 102                      Wait for positive
0057      :L   IW   0                      acknowledgement 5XXXH
0058      :L   KH 5000
005A      :AW
005B      :L   KH 5000
005D      :><F
005E      :BEC
005F      :DO  FW 108                      Check whether acknowledgement
0060      :L   IW   0                      belongs to job, otherwise wait
0061      :L   KH 9072                     for right acknowledgement
0063      :><F
0064      :BEC
0065 M001 :AN  F  100.1                    Set step flag
0066      :S   F  100.1
0067      :
0068      :***
```

```
SEGMENT  4     0069
0069      :A   F  100.2                    Last block deletes
006A      :JC  =M001                       acknowledgement 5000H.
006B      :L   KH 0000
006D      :DO  FW 102
006E      :T   QW   0
006F      :
0070      :DO  FW 102
0071      :L   IW   0
0072      :L   KH 5000
0074      :AW
0075      :L   KH 0000
0077      :><F
0078      :BEC
0079 M001 :AN  F  100.2
007A      :S   F  100.2
007B      :S   =OK                         Set parameter assignment flag
007C      :BE
```

```
FB 5                                B:ASCII@ST.S5D           LEN=103

        SEGMENT  1     0000
        NAME :CP-PARAM                              ASCII driver, transparent

        0005      :                                This FB is used to assign
        0006      :                                the CP 521 SI
        0007      :                                parameters.
        0008      :L   IW 104                      FW 100 is used for executing the FB
        0009      :L   KH 0F00
        000B      :AW                              Interrogation: Is CP still in
        000C      :L   KH 0F00                     restart routine?
        000E      :!=F
        000F      :BEC
        0010      :***


        SEGMENT  2     0011
        0011      :A   F  100.0                    Assign parameters to first block
        0012      :JC  =M001
        0013      :L   KH 9000                     Job block 0
        0015      :T   QW 104
        0016      :L   KH 0800                     Baud rate    Parity
        0018      :T   QW 106
        0019      :L   KH 0001                     BUSY         Interface
        001B      :T   QW 108
        001C      :L   KH 0100                     Data format  Handshake
        001E      :T   QW 110
        001F      :
        0020      :L   IW 104                      Acknowledgement 5XXXH
        0021      :L   KH 5000
        0023      :AW
        0024      :L   KH 5000
        0026      :><F
        0027      :BEC
        0028      :L   IW 110                      Check whether acknowledgement
        0029      :L   KH 9000                     belongs to job, otherwise wait for
        002B      :><F                             right acknowledgement
        002C      :BEC
        002D M001 :AN  F  100.0                    Set step flag
        002E      :S   F  100.0
        002F      :***


        SEGMENT  3     0030
        0030      :A   F  100.1                    Assign parameters to second block
        0031      :JC  =M001
        0032      :L   KH 9071                     Block 7    ASCII transparent
        0034      :T   QW 104
        0035      :L   KH 0001                     Character delay
        0037      :T   QW 106
        0038      :L   KH 000C                     Receive length
        003A      :T   QW 108
        003B      :L   KH 0000                     irrelevant  irrelevant
        003D      :T   QW 110
        003E      :
        003F      :L   IW 104                      Acknowledgement 5XXXH
        0040      :L   KH 5000
        0042      :AW
        0043      :L   KH 5000
        0045      :><F
        0046      :BEC
```

```
0047      :L   IW 110                             Check whether acknowledgement
0048      :L   KH 9071                            belongs to job, otherwise wait for
004A      :><F                                    right acknowledgement
004B      :BEC
004C M001 :AN  F  100.1                           Set step flag
004D      :S   F  100.1
004E      :***


SEGMENT  4     004F
004F      :A   F  100.2                           Last block deletes
0050      :JC  =M001                              acknowledgement 5000H.
0051      :L   KH 0000
0053      :T   QW 104
0054      :
0055      :L   IW 104
0056      :L   KH 5000
0058      :AW
0059      :L   KH 0000
005B      :><F
005C      :BEC
005D M001 :AN  F  100.2
005E      :S   F  100.2
005F      :S   F  100.3                           Set parameter assignment flag to
0060      :                                       prevent the parameter assignment
0061      :BE                                     FB from being processed further.
```

```
FB 6                                B:ASCII@ST.S5D              LEN=102

        SEGMENT  1     0000
        NAME :CP-PARAM                          ASCII driver, interpretive mode I

        0005      :                             This FB is used to assign the
        0006      :                             CP 521 SI
        0007      :                             parameters.
        0008      :L   IW 104                   FW 100 is used for executing the FB
        0009      :L   KH 0F00
        000B      :AW                           Interrogation: Is CP still in
        000C      :L   KH 0F00                  restart routine?
        000E      :!=F
        000F      :BEC
        0010      :***


        SEGMENT  2     0011
        0011      :A   F  100.0                 Assign parameters to first block
        0012      :JC  =M001
        0013      :L   KH 9000                  Job block 0
        0015      :T   QW 104
        0016      :L   KH 0800                  Baud rate    Parity
        0018      :T   QW 106
        0019      :L   KH 0001                  BUSY         Interface
        001B      :T   QW 108
        001C      :L   KH 0100                  Data format  Handshake
        001E      :T   QW 110
        001F      :
        0020      :L   IW 104                   Acknowledgement 5XXXH
        0021      :L   KH 5000
        0023      :AW
        0024      :L   KH 5000
        0026      :><F
        0027      :BEC
        0028      :L   IW  70                   Check whether acknowledgement
        0029      :L   KH 9000                  belongs to job, otherwise wait for
        002B      :><F                          right acknowledgement
        002C      :BEC
        002D M001 :AN  F  100.0                 Set step flag
        002E      :S   F  100.0
        002F      :***


        SEGMENT  3     0030
        0030      :A   F  100.1                 Assign parameters to second block
        0031      :JC  =M001
        0032      :L   KH 9072                  Block 7    ASCII interpretive
        0034      :T   QW 104
        0035      :L   KH 0010                  Character delay
        0037      :T   QW 106
        0038      :L   KH 0000                  Receive length
        003A      :T   QW 108
        003B      :L   KH 000D                  Receive end-of-text character
        003D      :T   QW 110
        003E      :
        003F      :L   IW 104                   Acknowledgement 5XXXH
        0040      :L   KH 5000
        0042      :AW
        0043      :L   KH 5000
        0045      :><F
        0046      :BEC
```

```
0047      :L   IW 110                           Check whether acknowledgement
0048      :L   KH 9072                          belongs to job, otherwise wait for
004A      :><F                                  right acknowledgement
004B      :BEC
004C M001 :AN  F  100.1                         Set step flags
004D      :S   F  100.1
004E      :***


SEGMENT  4     004F
004F      :A   F  100.2                         Last block deletes
0050      :JC  =M001                            acknowledgement 5000H.
0051      :L   KH 0000
0053      :T   QW 104
0054      :
0055      :L   IW 104
0056      :L   KH 5000
0058      :AW
0059      :L   KH 0000
005B      :><F
005C      :BEC
005D M001 :AN  F  100.2
005E      :S   F  100.2
005F      :S   F  100.3                         Set parameter assignment flag
0060      :BE
```

```
FB 11                                D:ASCII@ST.S5D            LEN=115

        SEGMENT  1     0000          Sending 12 bytes
        NAME :SEND

        0005      :C   DB  21                         Open send FB
        0006      :
        0007      :L   IW 104                         Is CP busy?
        0008      :L   KH 0F00
        000A      :AW
        000B      :L   KH 0F00
        000D      :!=F
        000E      :BEC
        000F      :
        0010      :A   F   80.0                       Send job already initiated
        0011      :
        0012      :JC  =M001
        0013      :
        0014      :L   IW 104                         Check whether a job can be
        0015      :L   KH C000                        initiated
        0017      :AW
        0018      :L   KH 0000
        001A      :><F
        001B      :BEC
        001C      :
        001D      :L   KH A001                        Initiate send job
        001F      :T   QW 104
        0020      :
        0021      :L   KH 000C                        Transfer send length
        0023      :T   QW 106
        0024      :
        0025      :L   KH 0000                        Transfer end-of-text character
        0027      :T   QW 108
        0028      :
        0029      :S   F   80.0
        002A      :R   F   80.1
        002B      :BEU
        002C      :
        002D M001 :L   IW 104                         Check acknowledgement
        002E      :L   KH F00F
        0030      :AW
        0031      :L   KH 5001
        0033      :><F
        0034      :JC  =M002
        0035      :
        0036      :L   IW 106                         Job OK?
        0037      :L   KH 0000
        0039      :><F
        003A      :JC  =M003
        003B      :
        003C      :L   KH B001                        Transfer block 1
        003E      :T   QW 104
        003F      :
        0040      :L   DW   0                         Transfer data word 0
        0041      :T   QW 106
        0042      :L   DW   1                         Transfer data word 1
        0043      :T   QW 108
        0044      :L   DW   2                         Transfer data word 2
        0045      :T   QW 110
        0046      :BEU
        0047      :
```

```
0048 M003 :L   IW 106                       Check acknowledgement block 1
0049      :L   KH 0001
004B      :><F
004C      :JC  =M002
004D      :
004E      :L   KH B002                      Transfer data word 2
0050      :T   QW 104
0051      :L   DW   3                       Transfer data word 3
0052      :T   QW 106
0053      :L   DW   4                       Transfer data word 4
0054      :T   QW 108
0055      :L   DW   5                       Transfer data word 5
0056      :T   QW 110
0057      :BEU
0058      :
0059 M002 :L   IW 104                       Check terminating acknowledgement
005A      :L   KH F00F
005C      :AW
005D      :L   KH 5000
005F      :><F
0060      :JC  =M004
0061      :
0062      :AN  I  106.0                     No error
0063      :JC  =M005
0064      :
0065      :AN  F   80.1
0066      :S   F   80.1
0067 M005 :R   F   80.2
0068      :R   F   80.0
0069      :
006A      :L   KH 0000                      Delete terminating acknowledgement
006C      :T   QW 104
006D M004 :BE
```

```
FB 12                                 D:ASCII@ST.S5D                   LEN=115

        SEGMENT  1     0000          Sending with end-of-text character
        NAME :SEND

        0005      :C   DB  22                          Open send FB
        0006      :
        0007      :L   IW 104                          Is CP busy?
        0008      :L   KH 0F00
        000A      :AW
        000B      :L   KH 0F00
        000D      :!=F
        000E      :BEC
        000F      :
        0010      :A   F   80.0                        Send job already initiated
        0011      :
        0012      :JC  =M001
        0013      :
        0014      :L   IW 104                          Check whether a job can
        0015      :L   KH C000                         be initiated
        0017      :AW
        0018      :L   KH 0000
        001A      :><F
        001B      :BEC
        001C      :
        001D      :L   KH A001                         Initiate send job
        001F      :T   QW 104
        0020      :
        0021      :L   KH 0000                         Transfer send length
        0023      :T   QW 106                          (in this case 0000H)
        0024      :
        0025      :L   KH 000D                         Transfer end-of-text character
        0027      :T   QW 108                          (in this case 000DH)
        0028      :
        0029      :S   F   80.0
        002A      :R   F   80.1
        002B      :BEU
        002C      :
        002D M001 :L   IW 104                          Check acknowledgement
        002E      :L   KH F00F
        0030      :AW
        0031      :L   KH 5001
        0033      :><F
        0034      :JC  =M002
        0035      :
        0036      :L   IW 106                          Job OK?
        0037      :L   KH 0000
        0039      :><F
        003A      :JC  =M003
        003B      :
        003C      :L   KH B001                         Transfer block 1
        003E      :T   QW 104
        003F      :
        0040      :L   DW   0                          Transfer data word 0
        0041      :T   QW 106
        0042      :L   DW   1                          Transfer data word 1
        0043      :T   QW 108
        0044      :L   DW   2                          Transfer data word 2
        0045      :T   QW 110
        0046      :BEU
        0047      :
```

```
0048 M003 :L   IW 106                              Check acknowledgement block 1
0049      :L   KH 0001
004B      :><F
004C      :JC  =M002
004D      :
004E      :L   KH B002                             Transfer block 2
0050      :T   QW 104
0051      :L   DW   3                              Transfer data word 3
0052      :T   QW 106
0053      :L   DW   4                              Transfer data word 4
0054      :T   QW 108
0055      :L   DW   5                              Transfer data word 5
0056      :T   QW 110
0057      :BEU
0058      :
0059 M002 :L   IW 104                              Check terminating acknowledgement
005A      :L   KH F00F
005C      :AW
005D      :L   KH 5000
005F      :><F
0060      :JC  =M004
0061      :
0062      :AN  I  106.0                            No error
0063      :JC  =M005
0064      :
0065      :AN  F   80.1
0066      :S   F   80.1
0067 M005 :R   F   80.2
0068      :R   F   80.0
0069      :
006A      :L   KH 0000                             Delete terminating acknowledgement
006C      :T   QW 104
006D M004 :BE
```

```
FB 13                                 D:ASCII@ST.S5D            LEN=102

        SEGMENT  1     0000           Receiving 12 bytes
        NAME :RECEIVE

        0005      :C   DB  22                           Open receive FB
        0006      :
        0007      :L   IW 104                           Is CP busy?
        0008      :L   KH 0F00
        000A      :AW
        000B      :L   KH 0F00
        000D      :!=F
        000E      :BEC
        000F      :
        0010      :A   F   80.4                         Receive job
        0011      :JC  =M001                            already initiated
        0012      :
        0013      :L   IW 104                           Check whether a job
        0014      :L   KH C000                          can be initiated
        0016      :AW
        0017      :L   KH 0000
        0019      :><F
        001A      :BEC
        001B      :
        001C      :L   KH A080                          Transfer receive job
        001E      :T   QW 104
        001F      :S   F   80.4
        0020      :R   F   80.5                         Reset error flag
        0021      :BEU
        0022      :
        0023 M001 :L   IW 104                           Check acknowledgement block 1
        0024      :L   KH F0FF
        0026      :AW
        0027      :L   KH 6001
        0029      :><F
        002A      :JC  =M002
        002B      :
        002C      :L   IW 106                           Read in data word 0
        002D      :T   DW   0
        002E      :L   IW 108                           Read in data word 1
        002F      :T   DW   1
        0030      :L   IW 110                           Read in data word 2
        0031      :T   DW   2
        0032      :
        0033      :L   KH C001                          Acknowledge message block 2
        0035      :T   QW 104
        0036      :BEU
        0037      :
        0038 M002 :L   IW 104                           Check acknowledgement block 2
        0039      :L   KH F0FF
        003B      :AW
        003C      :L   KH 6002
        003E      :><F
        003F      :JC  =M003
        0040      :
        0041      :L   IW 106                           Read in data word 3
        0042      :T   DW   3
        0043      :L   IW 108                           Read in data word 4
        0044      :T   DW   4
        0045      :L   IW 110                           Read in data word 5
        0046      :T   DW   5
        0047      :
```

```
0048      :L   KH C002                      Acknowledge message block 2
004A      :T   QW 104
004B      :BEU
004C      :
004D M003 :L   IW 104                       Check terminating acknowledgement
004E      :L   KH F00F
0050      :AW
0051      :L   KH 5000
0053      :><F
0054      :JC  =M004
0055      :
0056      :AN  I  106.0                     No error?
0057      :JC  =M005
0058      :
0059      :AN  F   80.5
005A      :S   F   80.5
005B M005 :R   F   80.6
005C      :R   F   80.4
005D      :L   KH 0000                      Delete terminating message
005F      :T   QW 104
0060 M004 :BE
```

```
FB 20                              B:ASCII@ST.S5D                 LEN=20

        SEGMENT  1     0000
        NAME :PULSE

        0005      :AN  F  101.1                        Flag 101.0 is set after the timer
        0006      :L   KT 030.1                        runs down and initiates a send ope-
        0008      :SP  T  101                          ration. The flag is reset following
        0009      :AN  T  101                          the initiation of a send operation
        000A      :AN  F  101.1                        and also following execution of
        000B      :=   F  101.1                        send FB. The setting for timer 101
        000C      :S   F  101.0                        should be at least greater than
        000D      :                                    twice the throughput time
        000E      :BE                                  (see Chapter 1 of the manual)


FB 21                              B:ASCII@ST.S5D                 LEN=41

        SEGMENT  1     0000
        NAME :DISTRIB

        0005      :AN  F   80.2                        Send not active
        0006      :JC  =M001
        0007      :
        0008      :JU  FB 200                          Process send FB
        0009 NAME :SEN  VAR
        000A BGAD :    KF +104
        000B Q-DB :    DB  21
        000C QANF :    KF +0
        000D QLAE :    KF +256
        000E      :R   F  101.0
        000F      :BEU
        0010      :
        0011 M001 :AN  F   80.6                        Receive not active
        0012      :JC  =M002
        0013      :
        0014      :JU  FB 201                          Process receive FB
        0015 NAME :REC VAR
        0016 BGAD :    KF +104
        0017 Z-DB :    DB  22
        0018 ZANF :    KF +0
        0019      :
        001A      :BEU
        001B      :
        001C M002 :A   F  101.0                        Set pulse flag
        001D      :S   F   80.2                        Set send flag
        001E      :R   F  101.0                        Reset pulse flag
        001F      :A   F   80.2
        0020      :BEC
        0021      :AN  F   80.6                        Set receive flag
        0022      :S   F   80.6
        0023      :BE
```

```
FB 200                              D:ASCII@ST.S5D             LEN=217

        SEGMENT  1     0000          Sending with adjustable length
        NAME :SEN  VAR
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Q-DB      I/Q/D/B/T/C: B
        DES  :QANF      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :QLAE      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

        0011      :LW  =BGAD                          Load and store
        0012      :T   FW 234                         module address
        0013      :
        0014      :DO  FW 234                         Is CP busy?
        0015      :L   IW   0
        0016      :L   KH 0F00
        0018      :AW
        0019      :L   KH 0F00
        001B      :!=F                                If yes, block end
        001C      :BEC
        001D      :***


        SEGMENT  2     001E
        001E      :A   F   80.0                       Job initiation flag
        001F      :JC  =M001                          If the flag is set,
        0020      :                                   do not initiate new job
        0021      :
        0022      :DO  FW 234                         Check whether job
        0023      :L   IW   0                         can be initiated
        0024      :L   KH C000
        0026      :AW
        0027      :L   KH 0000
        0029      :><F
        002A      :BEC
        002B      :
        002C      :L   KF +0                          Reset block counter
        002E      :T   FW 252
        002F      :LW  =QLAE
        0030      :T   FW 254
        0031 M003 :L   FW 252                         Increment block counter
        0032      :NOP 1
        0034      :T   FW 252
        0035      :L   FW 254                         Subtract the number of bytes
        0036      :NOP 1                              of a block from the length
        0038      :T   FW 254                         until zero is reached
        0039      :L   KF +0
        003B      :<=F
        003C      :JC  =M002                          Number of blocks is fixed
        003D      :JU  =M003                          Final number of blocks
        003E      :                                   not yet reached
        003F M002 :
        0040      :L   FW 234                         Set address pointer
        0041      :L   KF +2
        0043      :+F
        0044      :T   FW 236                         Module address + 2
        0045      :L   KF +2
        0047      :+F
        0048      :T   FW 238                         Module address + 4
        0049      :L   KF +2
        004B      :+F
        004C      :T   FW 240                         Module address + 6
        004D      :
```

```
004E      :L   KH A001                       Initiate send job
0050      :DO  FW 234
0051      :T   QW   0
0052      :LW  =QLAE                         Store length in word 1
0053      :DO  FW 236
0054      :T   QW   0
0055      :L   KH B001                       Store ID for acknowledgement
0057      :T   FW 242                        first block
0058      :L   KF +0                         Store comparison value
005A      :T   FW 244                        block 1
005B      :
005C      :LW  =QANF                         Data block
005D      :T   FW 246                        Beginning
005E      :L   KF +1
0060      :+F
0061      :T   FW 248                        Beginning + 1
0062      :L   KF +1
0064      :+F
0065      :T   FW 250                        Beginning + 2
0066      :
0067      :AN  F   80.0                      Set flag for send job
0068      :S   F   80.0                      initiated
0069      :R   F   80.1                      Reset error flag
006A      :BEU
006B M001 :***


SEGMENT  3     006C
006C      :DO  FW 234                        Load input word 0 and
006D      :L   IW   0                        check whether job
006E      :L   KH F00F                       has been accepted
0070      :AW
0071      :L   KH 5001
0073      :><F
0074      :JC  =M001
0075      :
0076      :DO  FW 236                        Check whether the
0077      :L   IW   0                        right block has
0078      :L   KH 00FF                       been received
007A      :AW
007B      :L   FW 244
007C      :><F
007D      :BEC
007E      :
007F      :L   FW 242                        Check whether a further
0080      :L   KH 00FF                       block may be sent
0082      :AW
0083      :L   FW 252
0084      :>F
0085      :BEC
0086      :
0087      :DO  =Q-DB                         Open source DB
0088      :L   FW 242                        Announce and
0089      :DO  FW 234                        transfer block
008A      :T   QW   0
008B      :
008C      :DO  FW 246                        Word 0
008D      :L   DW   0
008E      :DO  FW 236
008F      :T   QW   0
0090      :
```

```
0091      :DO  FW 248                     Word 1
0092      :L   DW   0
0093      :DO  FW 238
0094      :T   QW   0
0095      :
0096      :DO  FW 250                     Word 2
0097      :L   DW   0
0098      :DO  FW 240
0099      :T   QW   0
009A      :
009B      :L   FW 242                     Update block acknowledgement
009C      :L   KF +1
009E      :+F
009F      :T   FW 242
00A0      :
00A1      :L   FW 244                     Update block number
00A2      :L   KF +1
00A4      :+F
00A5      :T   FW 244
00A6      :                               Increment data word pointer
00A7      :L   FW 246                     Word 0
00A8      :L   KF +3
00AA      :+F
00AB      :T   FW 246
00AC      :
00AD      :L   FW 248                     Word 1
00AE      :L   KF +3
00B0      :+F
00B1      :T   FW 248
00B2      :
00B3      :L   FW 250                     Word 2
00B4      :L   KF +3
00B6      :+F
00B7      :T   FW 250
00B8      :BEU
00B9 M001 :***


SEGMENT  4     00BA
00BA      :DO  FW 234                     Check whether terminating
00BB      :L   IW   0                     acknowledgement present
00BC      :L   KH F00F
00BE      :AW
00BF      :L   KH 5000
00C1      :><F
00C2      :JC  =M001
00C3      :
00C4      :DO  FW 236
00C5      :L   IW   0
00C6      :L   KH 0100
00C8      :AW
00C9      :JZ  =M002
00CA      :AN  F   80.1                   Job has been terminated
00CB      :S   F   80.1                   with error
00CC M002 :R   F   80.2                   Enable for new send or
00CD      :R   F   80.0                   receive job
00CE      :
00CF      :L   KH 0000                    Delete terminating acknowledgement
00D1      :DO  FW 234
00D2      :T   QW   0
00D3 M001 :BE
```

```
FB 201                                 D:ASCII@ST.S5D            LEN=168

        SEGMENT  1     0000
        NAME :REC  VAR
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Z-DB      I/Q/D/B/T/C: B
        DES  :ZANF      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

        000E      :LW  =BGAD                      Load and store module
        000F      :T   FW 234                     address
        0010      :
        0011      :DO  FW 234                     Is CP busy?
        0012      :L   IW   0
        0013      :L   KH 0F00
        0015      :AW
        0016      :L   KH 0F00
        0018      :!=F
        0019      :BEC
        001A      :***


        SEGMENT  2     001B
        001B      :A   F   80.4                   Flag for job already
        001C      :JC  =M001                      initiated
        001D      :
        001E      :DO  FW 234                     Check whether job has
        001F      :L   IW   0                     been initiated
        0020      :L   KH C000
        0022      :AW
        0023      :L   KH 0000
        0025      :><F
        0026      :BEC
        0027      :
        0028      :L   FW 234                     Set address pointer
        0029      :L   KF +2
        002B      :+F
        002C      :T   FW 236                     Base address + 2 (word 1)
        002D      :L   KF +2
        002F      :+F
        0030      :T   FW 238                     Base address + 4 (word 2)
        0031      :L   KF +2
        0033      :+F
        0034      :T   FW 240                     Base address + 6 (word 3)
        0035      :
        0036      :L   KH A080                    Initiate receive job
        0038      :DO  FW 234
        0039      :T   QW   0
        003A      :L   KH 6001                    Load acknowledgements
        003C      :T   FW 242                     into flag words
        003D      :L   KH C001
        003F      :T   FW 244
        0040      :LW  =ZANF                      Set pointer to destination data
        0041      :T   FW 246                     Destination starting address
        0042      :L   KF +1
        0044      :+F
        0045      :T   FW 248                     Starting address + 1
        0046      :L   KF +1
        0048      :+F
        0049      :T   FW 250                     Starting address + 2
        004A      :
        004B      :S   F   80.4                   Flag for receive job initiated
        004C      :R   F   80.5
        004D      :BEU                            Unconditional block end
        004E M001 :***
```

```
SEGMENT  3     004F
004F      :DO  FW 234                       Load input word 0 and
0050      :L   IW   0                       check whether receive
0051      :L   KH F0FF                      block present
0053      :AW
0054      :L   FW 242
0055      :><F
0056      :JC  =M001
0057      :DO  =Z-DB                        Open destination DB
0058      :DO  FW 236
0059      :L   IW   0
005A      :DO  FW 246                       Word for
005B      :T   DW   0                       data word pointer 0
005C      :
005D      :DO  FW 238
005E      :L   IW   0
005F      :DO  FW 248                       Word for
0060      :T   DW   0                       data word pointer 1
0061      :
0062      :DO  FW 240
0063      :L   IW   0
0064      :DO  FW 250                       Word for
0065      :T   DW   0                       data word pointer 2
0066      :
0067      :L   FW 244                       Acknowledge data transfer
0068      :DO  FW 234
0069      :T   QW   0
006A      :
006B      :L   FW 242                       Update block message and
006C      :L   KF +1
006E      :+F
006F      :T   FW 242
0070      :
0071      :L   FW 244                       block acknowledgement
0072      :L   KF +1
0074      :+F
0075      :T   FW 244
0076      :
0077      :L   FW 246                       Increment data word pointer word 0
0078      :L   KF +3
007A      :+F
007B      :T   FW 246
007C      :
007D      :L   FW 248                       Increment data word pointer word 1
007E      :L   KF +3
0080      :+F
0081      :T   FW 248
0082      :
0083      :L   FW 250                       Increment data word pointer word 2
0084      :L   KF +3
0086      :+F
0087      :T   FW 250
0088      :BEU
0089 M001 :***


SEGMENT  4     008A
008A      :DO  FW 234                       Check whether terminating
008B      :L   IW   0                       acknowledgement received
008C      :L   KH F00F
008E      :AW
008F      :L   KH 5000
0091      :><F
0092      :JC  =M001
```

```
0093       :DO  FW 236
0094       :L   IW   0
0095       :L   KH 0100
0097       :AW
0098       :JZ  =M002
0099       :AN  F   80.5                        Job terminated
009A       :S   F   80.5                        with error
009B M002 :R   F   80.6                        Enable for new send or
009C       :R   F   80.4                        receive job
009D       :
009E       :L   KH 0000                         Delete terminating acknowledgement
00A0       :DO  FW 234
00A1       :T   QW   0
00A2 M001 :BE
```

7 Terminal Driver

7.1 Prerequisites for Using the Terminal Driver . . . . 7 - 2

7.2 Serial Interface . . . . 7 - 3

7.3 Assigning the CP 521 SI Parameters for Terminal Mode . . . . 7 - 4
7.3.1 Setting the Parameter Assignment Data . . . . 7 - 5
7.3.2 Storing the Parameter Assignment Data on the Memory Submodule . . . . 7 - 12
7.3.3 Changing Parameter Assignment Data during Terminal Mode . . . 7 - 15

7.4 Data Transfer CPU CP 521 SI Terminal Outputting Message Texts on the Terminal . . . . 7 - 17
7.4.1 Configuring the Message Texts . . . . 7 - 17
7.4.2 Initiating Message Text Output . . . . 7 - 18

7.5 Data Transfer Terminal CP 521 SI CPU Receiving Message Frames . . . . 7 - 20

7.6 CPU Job Requests and CP Error Messages . . . . 7 - 22

7.7 STEP 5 Program for Data Transmission with Terminal Driver . . . . . . 7 - 24

Figures

7-1. Terminal Diagram CP 521 SI - Terminal (RS-232C (V.24) Interface without Hardware Handshake) . . . . . 7 - 3
7-2. Schematic for Entering Headers and Footers . . . . . 7 - 10
7-3. Schematic for Configuring the Character Conversion Table . . . . . 7 - 11
7-4. Schematic for Entering Parameter Blocks in DB1 . . . . . 7 - 12
7-5. Process Output Image for Parameter Assignment . . . . . 7 - 15
7-6. Programming Method for Parameter Assignment . . . . . 7 - 16
7-7. Data Interchange over the CP 521 SI . . . . . 7 - 18
7-8. Process Image of the Outputs - Message Text Output . . . . . 7 - 19
7-9. Data Interchange over the CP 521 SI . . . . . 7 - 20
7-10. CPU Job Requests - Receiving Message Frames . . . . . 7 - 21
7-11. Structure of the STEP 5 Program for Data Transmission with the Terminal Driver . . . . . 7 - 24

Tables

7-1. Pin Assignments of the 25-Pin Subminiature D Socket Connector of the CP 521 SI . . . . . 7 - 3
7-2. Parameter Assignment Data in Terminal Mode . . . . . 7 - 4
7-3. Parameter Block Assignments . . . . . 7 - 5
7-4. Page Format . . . . . 7 - 9
7-5. Permissible Job Requests in the Terminal Driver Mode . . . . . 7 - 22
7-6. Status Byte in the Terminal Driver Mode (PII) . . . . . 7 - 23

7 Terminal Driver

Using the terminal driver enables you to connect a terminal for operator control and monitoring of your system. This allows you, for example, to follow process states on the screen of a terminal or also to intervene in the control of the process via the keyboard.

The terminal driver is a combination of the two modes, printer driver and ASCII drivers described in the previous chapters. For this reason, we will restrict ourselves in the following description to the information and special features required for startup of the terminal driver. For an understanding of the principle of data traffic between the CPU, the CP 521 SI and the peripheral device (terminal), see the following relevant chapters in this manual:

- Printer driver, Chapter 5
- ASCII driver, Chapter 6

In terminal driver mode, data transfer between the CPU and the CP 521 SI is half duplex, as it is in all other modes. Data exchange between the CP 521 SI and the peripheral device (terminal) is full duplex over the serial interface:

- Transfer of data from the CP 521 SI to the peripheral device is initiated by the CPU job request ”Output message text”. First store the message texts to be printed on the memory submodule in the CP 521 SI.
- At the same time, the CP 521 SI can receive message frames from the peripheral device. Passing data on to the CPU, e.g. keyboard inputs, is handled by the CPU job request ”Receive message frames”.

A memory submodule must be used in this mode.

Possible transmission methods depend on

- The type of serial interface (TTY or RS-232C (V.24))
- Whether handshake has been configured when using the RS-232C (V.24) interface (handshake OFF or handshake ON)
- Whether you have set XON/XOFF protocol parameters

Before starting up the terminal driver, a few preconditions must be met.

7.1 Prerequisites for Using the Terminal Driver

You must make the following settings to start up the terminal driver.

1. Establish connection

Establish the electrical connection between the CP 521 SI and the peripheral device (terminal) in the POWER OFF state.

2. Settings on the peripheral device (terminal)

You define the parameter assignment data of the CP 521 SI with the settings on the peripheral device. See the Operator's Guide of the peripheral device for a description of how to make the settings.

3. CP 521 SI parameter assignment data

Store all parameter assignment data on the memory submodule in DB1 or assign the parameters via the user program.

4. Configuring the message texts

The message texts to be printed are also stored on the memory submodule. You must configure at least one message text on the memory submodule in a DB between DB2 and DB63. You can configure up to 255 message texts.

5. Initializing the CP 521 SI

Plug the configured memory submodule into the CP 521 SI in the POWER OFF state. You can then switch the CPU to POWER ON.

6. Startup of the CP 521 SI in terminal mode

The CP 521 SI is in terminal mode after POWER ON and with a memory submodule plugged in if

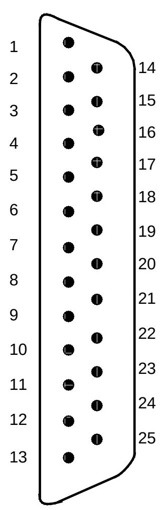
- a memory submodule is plugged in and the terminal driver mode parameters have been set on the memory submodule
  or
- the terminal driver mode has been transferred to the CP 521 SI in the user program.

> **Note**
>
> If there is no memory submodule plugged into the CP 521 SI and a backup battery has been inserted, the CP 521 SI is automatically initialized with the data in the RAM on power-up (i.e. with the parameters last assigned). In other words, the module is activated in the same driver mode following POWER-ON as it was before POWER-OFF.

7.2 Serial Interface

The CP 521 SI has a serial interface port. You can choose between a current interface (TTY) or a voltage interface (RS-232C (V.24)) by setting the relevant parameters (Section 7.3). The cables of both interfaces connect to a 25-pin subminiature D socket connector.

Table 7-1. Pin Assignments of the 25-Pin Subminiature D Socket Connector of the CP 521 SI

View	Pin No.	Signal Name	Meaning
	1	-	Disabled
	2	TxD	Send data (V.24)
	3	RxD	Receive data (V.24)
	4	RTS	Request to send (V.24)
	5	CTS	Clear to send (V.24)
	6	DSR	Data set ready (V.24)
	7	GND	Signal ground (RS-232C (V.24))
	8	-	Disabled
	9	TTY IN+	TTY receive line+
	10	TTY IN-	TTY receive line -
	11	-	Disabled
	12	-	Disabled
	13	P24	+24 V for active TTY
	14	-	Disabled
	15	-	Disabled
	16	-	Disabled
	17	20 mA	Current source TTY *****
	18	TTY OUT+	TTY send line+
	19	20 mA	Current source TTY *****
	20	DTR	Terminal ready
	21	TTY OUT-	TTY send line -
	22	-	Disabled
	23	-	Disabled
	24	-	Disabled
	25	-	Disabled

***** If +24 V to GND (pin 7) on pin 13

The following figure shows a possible connection in terminal mode. The CP 521 SI is set to the RS-232C (V.24) interface. The CP 521 SI and the terminal are linked over an RS-232C (V.24) cable.

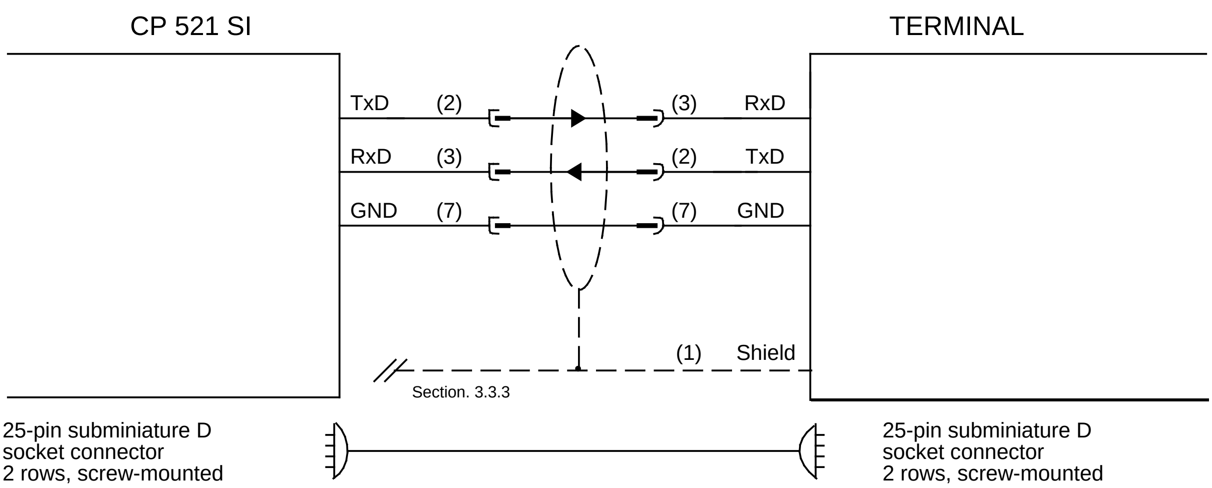


Figure 7-1. Terminal Diagram CP 521 SI - Terminal (RS-232C (V.24) Interface without Hardware Handshake)

7.3 Assigning the CP 521 SI Parameters for Terminal Mode

The CP 521 SI is supplied with default values for all parameters. If you want to use other values in terminal mode, you must assign the relevant parameters in the CP 521 SI.

The parameters have been divided into seven parameter blocks to make input as simple as possible. Division of the parameters into parameter blocks gives you the advantage that you need only assign parameters which deviate from the default values. However, you must then assign all parameters in this block even if some of the parameters within the block correspond to the default values.

The same conventions apply to assigning CP 521 SI parameters in terminal mode as apply in printer driver mode. You must only note the following:

- You do not have to set the following parameters in terminal mode:
  - Waiting times (parameter block 1)
- Parameters in parameter block 7 are assigned as for interpretive ASCII mode with the exception of the parameter for the mode.

You can define the following parameters before selecting the terminal driver:

Table 7-2. Parameter Assignment Data in Terminal Mode

Parameter	Parameter Block Number
Parameter of the serial interface	0
XON/XOFF protocol	2
Parameter for entering message texts	3
Output form for date and time of day	4
Page format	5
Header and footer	6
Setting of the communications driver	7
Character conversion table	8
Clock correction value	9

7.3.1 Setting the Parameter Assignment Data

The following table represents all parameter assignment data with which you can set the CP 521 SI in terminal mode.

See the operator's guide of your peripheral device for the possible settings on the peripheral device.

> **Note**
>
> The settings on the peripheral device must agree with the parameter assignment data of the CP 521 SI.

Table 7-3. Parameter Block Assignments

Block	Meaning		Value Range	Default Values
0	Baud rate	110 bit/s 200 bit/s 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s	1 2 3 4 5 6 7 8	8 9600 bit/s
	Parity (Parity bit always ”1”) (Parity bit always ”0”)	even odd ”mark” ”space” any	0 1 2 3 4	0 gerade
	BUSY signal (irrelevant for terminal driver)	no	0	0 no
	Interface	TTY RS-232C (V.24)	0 1	0 TTY
	Data format 11-bit character frame 10-bit character frame	 7 data bits (with parity) 8 data bits (with parity) 8 data bits (without parity) 7 data bits (without parity) 7 data bits (with parity) 8 data bits (without parity)	 0 1 2 3 4 5	0 7 data bits (11-bit character frame)
	HW handshake	OFF ON	0 1	0 no
2	XON/XOFF protocol	XON character XOFF character	ASCII character (00_H to $7F_H$) ASCII character (00_H to $7F_H$)	FF_H (no XON/XOFF protocol)
3	Function character		ASCII character	” (22_H)
	End-of-text character		ASCII character	$ (24_H)

Table 7-3. Parameter Clock Assignments (Cont.)

Block	Meaning	Value Range	Default Values
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 (lower case also permissible)	TMJ
	Separator for date	ASCII character (20_H to $7F_H$)	"." ($2E_H$)
	Order for time of day H =Hours M =Minutes S =Seconds	HMS, HSM, MSH, MHS, SHM, SMH (lower case also permissible)	HMS
	Separator for time of day	ASCII character (20_H to $7F_H$)	":" ($3A_H$)
	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_D to 255_D 0_D to 60_D O, o, T, t U, u, B, b, otherwise no pagination	72_D 0 Page number not inserted
6	Header and footer		
	Header 1 Header 2 Footer 1 Footer 2	K1 ... Text ... $ (End-of-text char.) K2 ... Text ... $ F1 ... Text ... $ F2 ... Text ... $	No header or footer
7	Setting the mode		
	Terminal driver	6	0: Memory submodule with message texts plugged in 1: Without memory submodule**
	Character delay	1_D to $65\,535_D$ (*10 ms)	1 (*10 ms)
	Message frame length	1_D to 256_D	64_D
	End-of-text character 1 ASCII character No EOT character	 01_H to $7F_H$ 00_H	00_H
	End-of-text character 2 ASCII character No EOT character	 01_H to $7F_H$ 00_H	$0D_H$*
8	Character conversion table	(next pages)	None
9	Clock correction value	-400_D to 400_D (s/month)	0

* Carriage Return

** Only if there is no backup battery; if a backup battery is installed, the same driver is active after POWER-ON as before POWER-OFF

Explanation of parameters

You define the desired mode "Terminal driver" in parameter block 7 on initial start-up of the CP 521 SI. You can also assign parameters for character delay, message frame length or one or two end-of-text characters for data reception.
The parameters you decide to assign to the serial interface (parameter block 0) will depend on the interface characteristics of your peripheral device and on the particular application.

Mode (parameter block 7)

You must set a ”6” in parameter block 7 to select the terminal driver.

Message frame length (Parameter block 7)

When receiving message frames of fixed length from the peripheral device, you define the length of the frame in parameter block 7. Make sure that the same frame length is configured on the CP 521 SI and the peripheral device.

Character delay (parameter block 7)

You can define the maximum time which is to elapse between two received characters (character delay). Only those characters whose delay is within the defined limits will be recognized by the CP 521 SI as valid and transferred to the CPU as a message frame.

End-of-text characters (parameter block 7)

When receiving message frames with end-of-text characters from the peripheral device, you must define one or two end-of-text characters in parameter block 7. You use the end-of-text characters you have chosen to limit the length of the receive frame. You can receive frames with a maximum length of 256 bytes.

Baud rate (parameter block 0)

You can choose from eight baud rates. The default is 9600 bit/s. If you use the RS-232C (V.24) interface, the load capacity of cables longer than 15 m will have a negative effect on the baud rate. Longer cables can be used in general if the baud rate is reduced.

Parity (parameter block 0)

You can choose between five types of parity.

- 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 always has signal state ”1”.
- ”space”
  The parity bit always has signal state ”0”.

- No parity check
  The signal state of the parity bit is not significant. Parity is not checked when receiving; however, when sending, the parity bit is always set to ”1”.

The default is even parity.

BUSY signal (parameter block 0)

The BUSY signal is not relevant for the terminal driver. Leave the default value ”0” (no BUSY signal) at this position in parameter block 0.

Interface (parameter block 0)

You can choose between the RS-232C (V.24) and TTY interfaces here. See Section 2.3 for more detailed information on these interfaces.
The TTY interface is the default.

Data format (parameter block 0)

Characters are transmitted between the CP 521 SI and the peripheral device in a 10-bit or 11-bit character frame. You can choose between seven and eight data bits within these character frames (Figure 4-4).

10-bit character frame:

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

11-bit character frame:

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

The 11-bit character frame (1 start bit, 7 data bits, 1 parity bit, 2 stop bits) is the default.

Hardware handshake (parameter block 0)

This parameter is only significant for the RS-232C (V.24) interface. If you set HW handshake ”ON”, the ”RTS”, ”CTS”, ”DTR” and ”DSR” control signals of the RS-232C (V.24) interface are evaluated. See Chapter 6 ”ASCII Driver” for a detailed description of the handshake mechanism.

The default setting is HW handshake ”OFF”, i.e. the control signals are not evaluated.

XON/XOFF protocol (parameter block 2)

The ”XON” and ”XOFF” characters are control characters used by the CP 521 SI for controlling data transmission. The CP 521 SI receives ”XOFF” from the peripheral device when the receive buffer in the peripheral device is full. The peripheral device then receives no more data from the CP 521 SI. Data would be lost in the case of an overflow of the receive mailbox. Only when the CP 521 SI receives the ”XON” character does it resume sending.

Function character (parameter block 3)

If you configure place holders (Section 5.5) in a message text that you want to display on the terminal screen, you must limit these place holders with a function character. You can configure which ASCII character you want to use as function character in parameter block 3.

End-of-text character (parameter block 3)

When configuring message texts (Section 5.5), you must specify an end-of-text character at the end of a text. You can store up to 255 message texts on the memory submodule. You can configure which ASCII character you want to use as end-of-text character in parameter block 3.

If you do not use the default end-of-text character ”$” and configure another ASCII character in DB1 as the end-of-text character, parameter block 3 must be the first block you enter in DB1 on the memory submodule.

Page format (parameter block 5)

Table 7-4. Page Format

Line	Page Format	
1	Blank line	(if page number, header or footer has been configured)
	Blank line	(if page number, header or footer has been configured)
.	Page No.	(top, if configured)
.	Blank line	(if page number configured at top)
.	Header 1	
.	Header 2	
.	Blank line	(if header(s) configured)
.	Blank line	(if header(s) configured)
.	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 Chapter 5.5.	
.	Blank line	(if two footers have been configured)
.	Blank line	(if one footer has been configured)
.	Footer 1	
.	Footer 2	
.	Blank line	(if the page number is configured at the bottom)
.	Page No.	(bottom, if configured)
	Blank line	(if page number, header or footer has been configured)
n	Blank line	(if page number, header or footer has been configured)

Left margin (configurable up to 60 characters)

Number of lines per page: 20 to 255

Configuring headers and footers (parameter block 6)

Up to two headers and two footers can be configured in parameter block 6. As in message texts, place holders can be used for inserting the date and time of day (Section 5.5).

Proceed according to the following schematic:

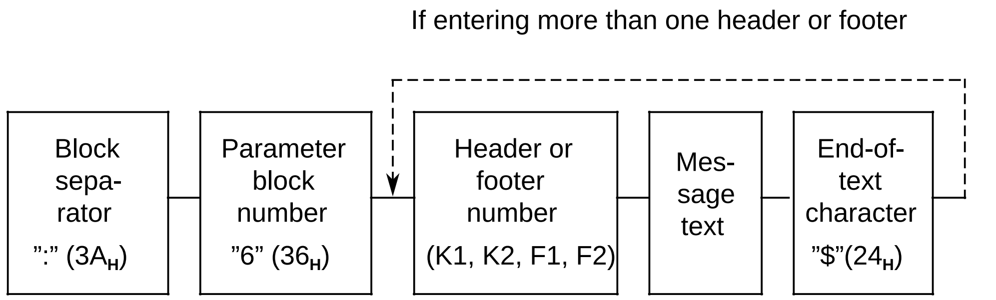


Figure 7-2. Schematic for Entering Headers and Footers

Enter the header or footer like the ASCII characters of a message text (Section 5.5).

> **Note**
>
> A header or footer must not be configured with more than 136 characters. The printed text may be longer than one line if place holders are used.

Continuous form feed

If you want to print your message texts without page format, you are recommended to configure message texts as follows:

- Without page number (parameter block 5)
- Without header (parameter block 6)
- Without footer (parameter block 6)

This will give you continuous form feed for all message texts.

Character conversion table (parameter block 8):

In order to adapt to special national characters, you can replace up to 16 ASCII characters with a code sequence of up to 16 characters in parameter block 8. Use the end-of-text character configured by you in parameter block 3 as a separator. Parameter block 8 must be configured according to the following schematic:

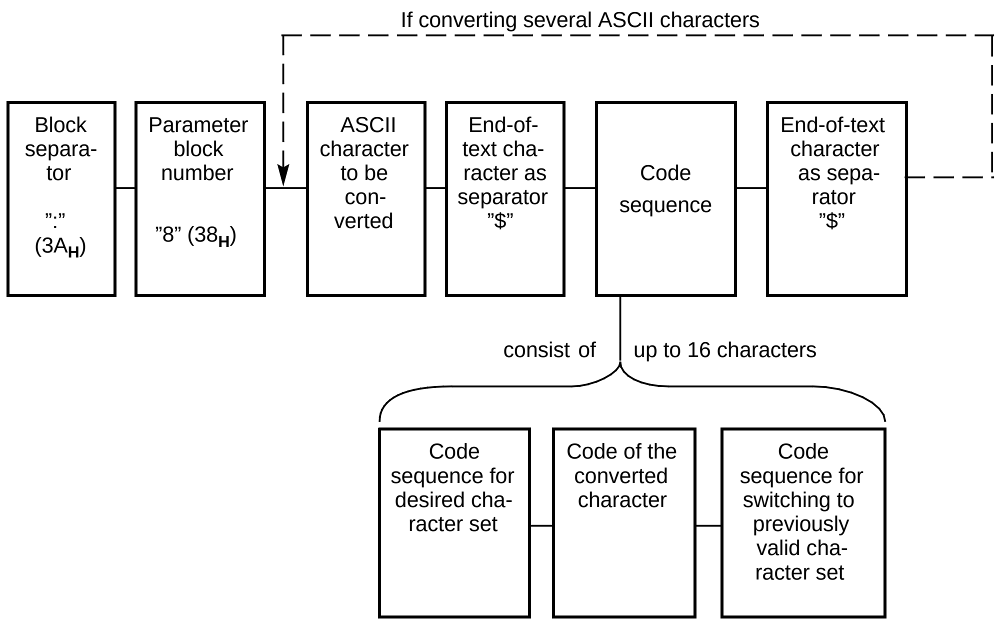


Figure 7-3. Schematic for Configuring the Character Conversion Table

Clock correction factor (parameter block 9):

You can configure a correction value in parameter block 9 to improve the accuracy of the module clock. In doing so, please note that the correction value must always be specified with four digits (incl. sign).

The correction value is output in s/month. The month is fixed at 30 days (Section 5.3.5, Clock Correction Factor; parameter block 9).

7.3.2 Storing the Parameter Assignment Data on the Memory Submodule

When configuring the parameter assignment data, you proceed in exactly the same way as for the printer driver. Store all parameter assignment data on a memory submodule in data block (DB)1. During terminal driver mode, you can change the interface parameters (parameter block 0) and the setting of the communications driver (parameter block 7) **also** via the user program (Section 7.3.3).

Note

Parameter assignment data in the user program has priority over parameter assignment data in the memory submodule. Accordingly, the parameter assignment data on the memory submodule is valid after POWER ON only insofar as it is not overwritten by the user program.

Entering the parameter blocks in the memory submodule

When you have decided on parameters suitable for your system, start entering the parameter blocks:

Plug the memory submodule into a programmer.

Transfer the entire memory submodule contents from the submodule to the programmer (diskette, hard disk).

Delete the memory submodule.

Enter all parameter blocks you want to modify in DB1.

Transfer the entire memory submodule contents back to the memory submodule.

The parameter blocks can be entered in DB1 according to the following schematic.

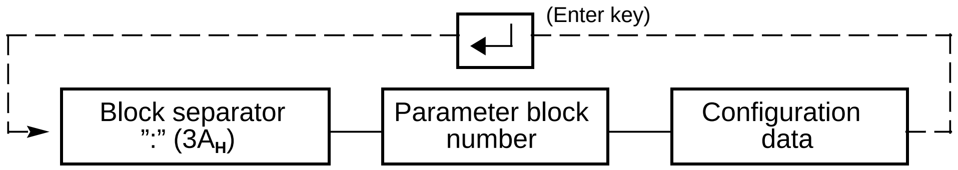


Figure 7-4. Schematic for Entering Parameter Blocks in DB1

When you have stored the parameter assignment data on the memory submodule in DB1 using the programmer, plug the configured memory submodule into its receptacle on the CP 521 SI. At the next module power-up, the parameters on the memory submodule are checked and transferred to the module.

Example: Transferring parameter assignment data for parameter blocks 0, 3, 5 and 7 to the memory submodule

Starting address 120 is set on the module. The parameters are to be assigned as follows:

- Parameters for entering message texts (parameter block 3)
  - Function character: ” (”)
  - End-of-text character: $ ($)

> **Note**
>
> If you want to define an end-of-text character for the message text in parameter block 3, this must be the first parameter block you enter in DB1.
> The end-of-text character (parameter block 3) acts as a separator between the various parameters of a parameter block.

- Parameters for the serial interface (parameter block 0):
  - 9600 bit/s (8)
  - Even parity (0)
  - No BUSY (0)
  - RS-232C/V.24) interface (1)
  - 8 data bits (1)
  - Handshake OFF (0)

- Parameters for waiting times (parameter block 1):
  - Irrelevant for the terminal driver

- XON/XOFF protocol (parameter block 2):
  - XON character: DC1 (11_H) (11)
  - XOFF character: DC3 (13_H) (13)

- Parameters for message text output:
  - Specification of the date and time of day unchanged with respect to default values (parameter block 4)
  - Page format (parameter block 5):
    72 lines/page (072)
    5 characters left margin (05)
    Page number at the top (o)
  - Header (parameter block 6):
    Place holder for header 1/ (K1header 1)
    header 2 (K2header 2)
  - Footer (parameter block 6):
    Place holder for footer 1/ (F1footer 1)
    footer 2 (F2footer 2)

- Parameters for the terminal driver (parameter block 7)
  - Terminal driver (6)
  - Character delay: 10 ms (1)
  - End-of-text character: $0D_H$ ($0D_H$)

- Configuring character conversion (parameter block 8)
  - Convert character ”+”

- Parameters for clock correction (parameter block 9)
  - Correction factor: 1s/month (+001)

Invoke DB1 from the PG 750 programmer. Then enter the values. To make these more understandable, use comments (**KS**=...).

Input from PG 750 to DB1	Explanation
0: KS ='Parameters for the';	Block 3 must first be assigned
12: S ='CP521SI';	parameters, unless you wish to
24: KS ='Parameter block_3 ';	accept the default values.
33: S =':3$"$ ';	$ = separator, " = function character
36: KS ='Parameter block_0 ';	9600 baud, even parity, no Busy,
45: S =':0$800110$';	RS232C (V.24), 8 bits, no handshake
50: KS ='Parameter block_2 ';	XON = 11H, XOFF = 13H
59: S =':2$11$13$ ';	
64: KS ='Parameter block_4 ';	Day.month.year
73: S ='4$TMJ.$HMS:d ';	Hour: minute:second 24h system
81: KS ='Parameter block_5' ;	72 lines/page, left margin = 5
90: S =':5$072$05o';	page number at top
96: KS ='Parameter block_6' ;	Header_1
105: S =':6$K1 Header_1$K2Header ';	Header 2
114: S ='2$F1Footer_1$F2F';	Footer_1
122: S ='ooter_2$';	Footer_2
126: KS ='Parameter block_7 ';	Terminal mode, character delay = 1*10ms
135: S =':7$6$1$0D$00$ ';	End-of-text character = 0DH
143: KS ='Parameter block_8 ';	Character conversion
152: S =':8$+$1B384B5B1B3840$';	
162: KS ='Parameter block_9 ';	Clock correction factor
171: S =':9$+001$';	+ 1 second/30 days
175:	

7.3.3 Changing Parameter Assignment Data during Terminal Mode

During terminal mode of the CP 521 SI you can do the following:

- Change the parameter assignment data of the serial interface (parameter block 0)
- Change the character delay and the end-of-text character (parameter block 7)
- Change from terminal driver mode to another mode (parameter block 7).

You transfer the parameters to the module over the PIQ with the user program.

> **Note**
>
> Parameter assignment data in the user program has priority over parameter assignment data in the memory submodule. Accordingly, the parameter assignment data on the memory submodule is valid after POWER ON only insofar as it is not overwritten by the user program.

We will now demonstrate how you can set the terminal driver (parameter block 7) in the user program. The procedure for reassigning the serial interface parameters (parameter block 0) is analogous (Chapter 6, Interpretive ASCII Driver).

General procedure

Store the following data in this order in the PIQ of the CPU via the user program:

Parameter assignment data in bytes 2 to 7
CPU job request 90_H ”Transfer parameter assignment data” in byte 0
Block number 7_H and mode 6_H ”Terminal driver” in byte 1.

Address	Contents	
Maddr +0	90_H	Code for "Transfer parameter assignment data"
Maddr +1	76_H	Block number and mode
Maddr + 2	00_H to FF_H	Character delay (ZVZ)
Maddr + 3	01_H to FF_H	
Maddr + 4	00_H	Message frame length
Maddr + 5	00_H	
Maddr + 6	01_H to $7F_H/00_H$	1. End-of-text character (ASCII character)/ no end-of-text character
Maddr +7	01_H to $7F_H/00_H$	2. End-of-text character (ASCII character)/ no end-of-text character

Maddr = Module address

Figure 7-5. Process Output Image for Parameter Assignment

Example: Programming method

The CP 521 SI is installed in slot 7 (starting address 120).

Assign the module parameters as follows:

- Parameters for the terminal driver (parameter block 7)
  - Terminal driver (6)
  - Character delay (ZVZ) (200 x 10 ms)
  - Message length (0)
  - 2 end-of-text characters ($0D_H$, $0D_H$)

Address	Contents
120	90_H
121	76_H
122	00_H
123	$C8_H$
124	00_H
125	00_H
126	$0D_H$
127	$0D_H$

User program

FB60

```
NAME    :BLOCK7
        :L  KH 9076
        :T  QW 120
        :L  KF+200
        :T  QW 122
        :L  KF+0
        :T  QW 124
        :L  KH 0D0D
        :T  QW 126
        :BE
```

Figure 7-6. Programming Method for Parameter Assignment

> **Note**
>
> Only one block can be transferred per program cycle, i.e. a further parameter block (e.g. block 8) can only be transferred in the next OB1 cycle.

> **Note**
>
> If you only have to evaluate individual keys (terminal), you are recommended to set the ZVZ (character delay) as low as possible.

7.4 Data Transfer CPU CP 521 SI Terminal Outputting Message Texts on the Terminal

7.4.1 Configuring the Message Texts

The same conventions apply to configuring message texts for the terminal driver as apply to configuring message texts for the printer driver. You configure all the message texts you want to transfer to the terminal on a memory submodule in data blocks DB2 to DB63. You can store up to 255 message texts on the memory submodule. Make sure you do not exceed the capacity of the memory submodule.

Store the message texts on the memory submodule under a message text number, i.e. each message text must begin with a message text number (1 to 255) followed by a separator (:). This is then followed by the ASCII characters of the message text terminated by the end code (see chapter on parameter assignment). The number of the next message text follows immediately after the end-of-text character.

You can enter the following as ASCII characters:

- Text
  You can specify all printable characters in the text (see the user's guide of the terminal).

- Place holders
  You can insert the following in a text by configuring place holders: date, time of day, variables and further texts.

Each text can have a length of 136 characters including variables.

Exception:
When using place holders (Section 5.5.1) and/or a character conversion table, up to 250 characters can be processed. If you enter further ASCII characters in addition to this, they will not be evaluated.

Since configuration of the message texts follows exactly the same procedure as for the printer driver, we will omit a repetition of the explanatory notes. You are directed instead to Section 5.5 ”Configuring Message Texts”.

7.4.2 Initiating Message Text Output

Output of the message texts stored on the memory submodule is initiated by the CPU over the user program. You use the same CPU job requests as for printer output (Section 5.6).

One message text per program cycle can be output over the CP 521 SI.

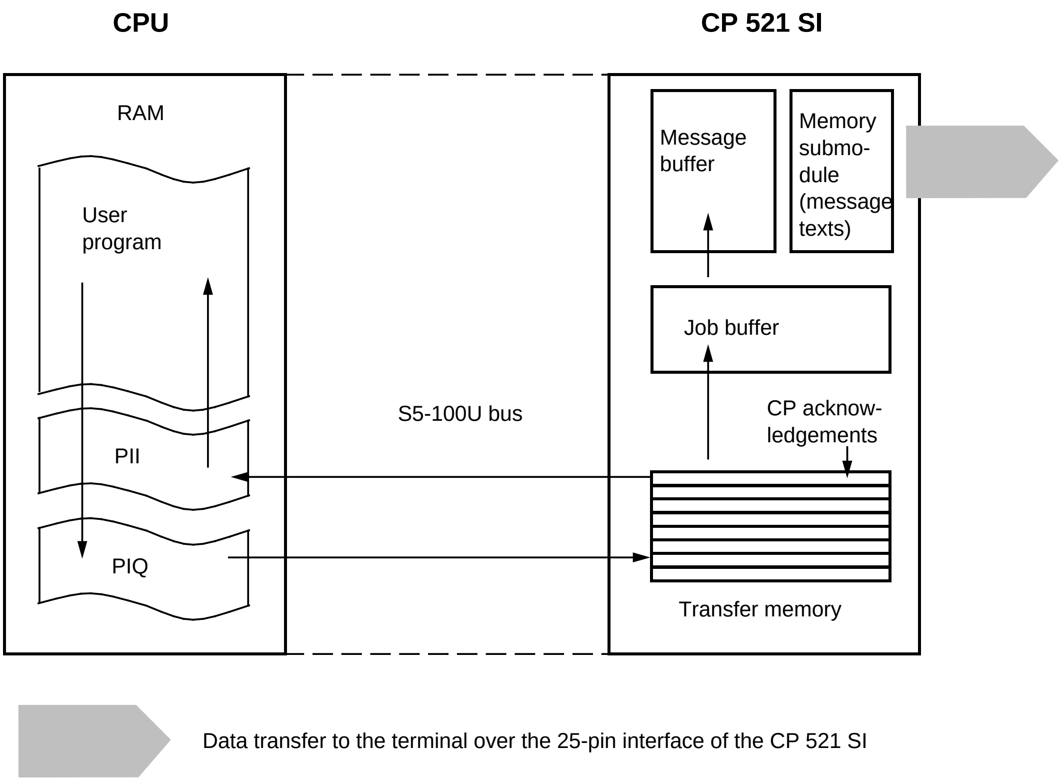


Figure 7-7. Data Interchange over the CP 521 SI

When configuring the message texts, you have assigned a number to each text. This is important, since you must transfer the message text number to the CP 521 SI along with the CPU job request ”Print message text”. This is how the CP 521 SI recognizes the message text to be output to the terminal. If variables have been configured in the message texts, the current variable values must also be transferred from the CPU to the CP 521 SI with the message text number.

If there are several message texts which the terminal cannot process immediately, the CP 521 SI stores these messages in the message buffer. The print job requests are then processed in the order in which they are sent from the CPU (FIFO=FIRST IN FIRST OUT). If you want the message texts in a particular order on the terminal, you must call the texts in this order in the user program.

General procedure

Reminder:
You can transfer eight-byte blocks over the transfer memory to the CP 521 SI.

You **must** transfer the following in the PIQ to the CP 521 SI:

The ”Printout message text” job request in byte 0
The message text number (hexadecimal) in byte 1

You **can** transfer the following in the PIQ to the CP 521 SI:

The variable values in byte 2, if you have configured place holders for variables in the message text.

You must output the variable values in the format you have configured in the message text on the memory submodule. At output, the place holders will then be replaced by the current variable values.

Address	Contents	
Maddr +0	$30_H/40_H/80_H/F0_H$	Job request numbers: 30_H: Output message text with CR/LF at the end of the line 40_H: Output message text without CR/LF at the end of the line 80_H: Printout of all message texts $F0_H$: Print parameter assignment data
Maddr +1	00_H to FF_H	Message text number (up to 255 message texts)
Maddr + 2 Maddr + 3	Value for 1st variable*	Variable value for the **first** place holder for variables occurring in the message text
Maddr + 4 Maddr + 5	Value for 2nd variable*	Variable value for the **second** place holder for variables occurring in the message text
Maddr + 6 Maddr +7	Value for 3rd variable*	Variable value for the **third** place holder for variables occurring in the message text

* The possible value range for variables depends on the configured data format
(Chapter 5, Printer Driver)

Figure 7-8. Process Image of the Outputs - Message Text Output

> **Note**
>
> See Section 7.7 for a programming example for terminal mode of the module. There are further programming examples for message text output in Chapter 5, Printer Driver.

7.5 Data Transfer Terminal CP 521 SI CPU Receiving Message Frames

In terminal mode, you can receive message frames from the peripheral device. The CP 521 SI handles the entire data transfer of the message frames in the same way as in interpretive ASCII driver mode (Chapter 6). The following explanations are therefore restricted to the permissible CPU job requests.

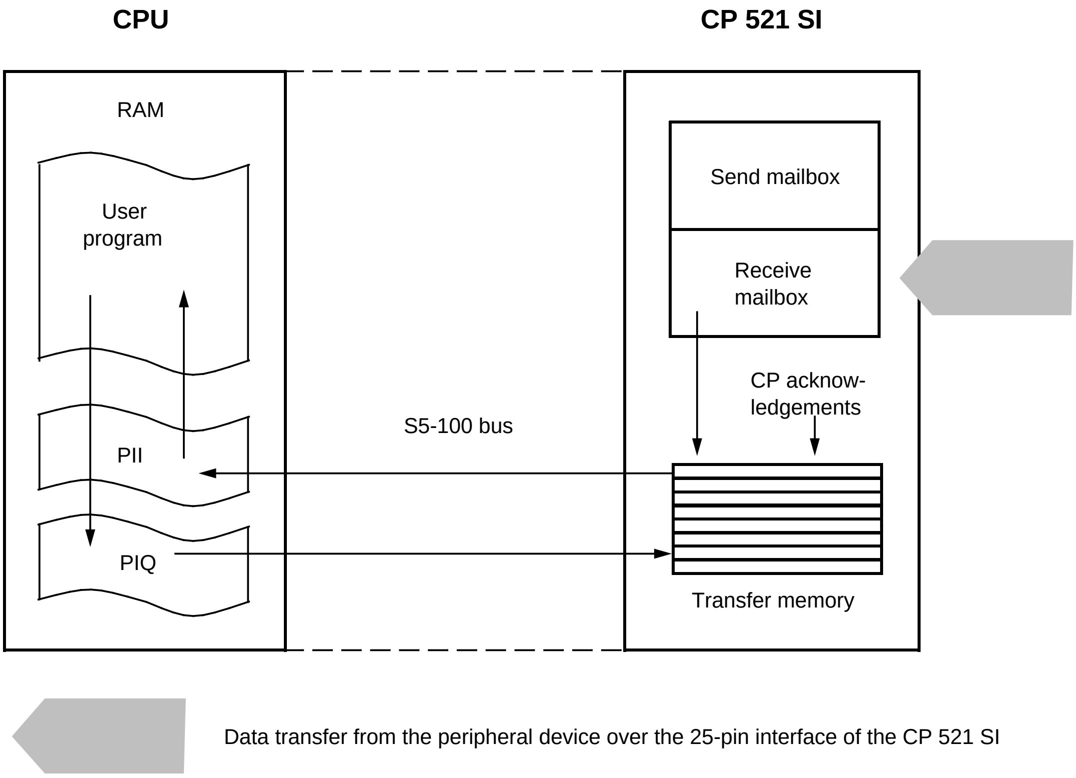


Figure 7-9. Data Interchange over the CP 521 SI

Reminder:
The CPU can read a message frame received from the terminal in eight-byte blocks from the receive buffer of the CP 521 SI. The receive buffer has a capacity of 1 KB and can store 99 message frames.

You can receive message frames with end-of-text characters or with defined length analogously to interpretive ASCII driver mode. Remember that you have to limit the length of a receive message frame using an end-of-text character selected by you (parameter block 7). You can receive message frames with a length of up to 256 bytes.

Program the CPU in such a way that it reads the message frame in eight-byte blocks (two coordination bytes and six data bytes) from the receive mailbox of the CP 521 SI:

1. The CPU sends a job request: ”Coordinate data transfer receive” to the CP 521 SI.

2. The CP 521 SI begins with data transfer. It sends the first six bytes from its receive buffer (FIFO memory).

3. The CPU sends an acknowledgement to the CP 521 SI for receipt of the data.

4. The CP 521 SI sends the next block and so on until the entire message frame has been transferred to the CPU.

In terminal mode, the following two CPU job requests are permissible for bidirectional data traffic:

- $A080_H$: Coordinate data transfer receive
- $C0xx_H$: CPU acknowledgement for data received

Address	**Contents**
Maddr +0	$A0_H$
Maddr +1	80_H
Maddr + 2	irrelevant
Maddr + 3	irrelevant
Maddr + 4	irrelevant
Maddr + 5	irrelevant
Maddr + 6	irrelevant
Maddr +7	irrelevant

Code for request to receive

Contents
$C0_H$
00_H to FF_H
irrelevant
irrelevant
irrelevant
irrelevant
irrelevant
irrelevant

Code for ”Acknowledge message block”

Number of the message block received

Maddr = Module address

Figure 7-10. CPU Job Requests - Receiving Message Frames

The CP 521 SI sends an acknowledgement to the CPU for receipt of the message frame with the terminating acknowledgement 5000_H (byte 0 and 1 in PII). Byte 5 (PII) of the terminating message contains the number of valid bytes in the last message block received.

See Chapter 6, ”ASCII Driver” for details of the special data transfer and acknowledgement mechanisms between the CPU and the CP 521 SI for receiving message frames.

7.6 CPU Job Requests and CP Error Messages

This section contains all the permissible CPU job requests for terminal driver mode. In addition, you will find the CP 521 SI error messages, which you can read in and evaluate over the PII.

Permissible CPU job requests in the terminal driver mode

Table 7-5. Permissible Job Requests in the Terminal Driver Mode

Byte 0	Byte 1	Job Requests
00_H	00_H	Blank job request: Current clock data (Section 4.4)
10_H	00_H	Set clock (Section 4.4.3)
$\boxed{20_H}$	00_H	Specify page number in byte 1 (Section 5.6.1)
$\boxed{30_H}$	XX_H	Output message with CR/LF at the end XX_H: 01_H to FF_H (message text number: 1 to 255) (Section 5.6.2)
$\boxed{40_H}$	XX_H	Output message without CR/LF at the end XX_H: 01_H to FF_H (message text number: 1 to 255) (Section 5.6.2)
$\boxed{50_H}$	00_H	Output page feed (Section 5.6.3)
$\boxed{60_H}$	00_H	Output line feed (Section 5.6.4)
70_H	00_H	Delete message buffer (Section 5.6.5)
80_H	00_H	Output all message texts (Section 5.6.6)
90_H	00_H	Transfer parameters: Parameter block 0
90_H	$7x_H$	Transfer parameters: Parameter block 7
90_H	76_H	Setting: Terminal driver x=6
$A0_H$	80_H	Coordinate data transfer ”Receive”
$C0_H$	Message block number	CPU acknowledgement: Received data accepted
$D0_H$	00_H	Delete receive mailbox contents
$F0_H$		Print parameter assignment data
$\boxed{\quad}$		Print job requests. These are written into the message buffer, if required. Job 80_H, ”Printout of all messages”, is executed immediately. In the case of the CPU job request 80_H, the message text numbers are also printed out.

If you use a job request which is illegal for the terminal driver mode, this is acknowledged by the CP 521 SI with the appropriate response (Section 5.6/6.5).

Note

When executing the job "Delete receive mailbox contents", the serial interface of the CP 521 SI is briefly disabled. If the peripheral device connected continues to send data, the first message frame received by the CP 521 SI can be faulty or incomplete.
The interface to the CPU is also briefly occupied by the CP checkback signal "$0F_H$". Therefore **always** evaluate the CP acknowledgement to the job "Delete receive mailbox contents" before you send further CPU job requests to the CP 521 SI.
The CP 521 SI acknowledges the job "Delete receive mailbox contents" like all parameterization jobs with the terminating message "5000_H" **and** specification of the job acknowledged (here: "$D000_H$") in bytes 6 and 7 of the CP response.

CP error messages

If you have activated the terminal driver, you can read out and evaluate the following status information of the CP 521 SI in the PII in byte 0 (status byte).

Table 7-6. Status Byte in the Terminal Driver Mode (PII)

Byte 0 Bit 4 to 7	Bit 0 to 3	Status
0	0	No error
X	1	Memory submodule error
X	2	No message text available
X	7	No battery backup in CP
X	8	Message buffer overflow
X	9*	Character delay exceeded
X	A*	Parity error
X	B*	Receive after XOFF or receive after DTR=”OFF”
X	C*	Message longer than 256 bytes
X	E*	Receive mailbox overflow
X	F	CP 521 SI in restart routine or job buffer full
1	X	Clock defective
2	X	Default time set
3	X	Time/date error
4	X	Illegal job request
8	X	Hardware fault

X=Signal state not significant for the other nibble

* These error messages do not appear immediately, but only after the ”Receive” coordination job request.

You will find supplementary explanations of these CP responses in the following sections of the manual:
- Printer Driver, Section 5.7
- ASCII Driver, Section 6.5

7.7 STEP 5 Program for Data Transmission with Terminal Driver

The following is a complete STEP 5 program for a terminal driver application.

- DB2 Configured message texts
- DB22: Receive data (destination DB)

- FB3: Assigning the CP 521 SI parameters in the STEP 5 program (user-configurable; can only be used for CPU 103/S5-95U PLC)
- FB10: A message is printed out in response to the keyboard entry
- FB13: Receive FB (module starting address=104)
- FB14: Calculation of the frame length for FB201
- FB21: Enable and initiate receive
- FB201 Receive FB (user-configurable; can only be used for CPU103/S5-95U PLC)

- OB1:
  - Invokes the parameter assignment FB (FB3)
    If the CP 521 SI has been assigned its parameters, this FB is no longer invoked.
  - Invokes FB21, which, in turn, invokes the receive FB (FB13)
  - Invokes FB10, which, in turn, initiates message text output
- OB21/22: Defaults for parameter assignment FB

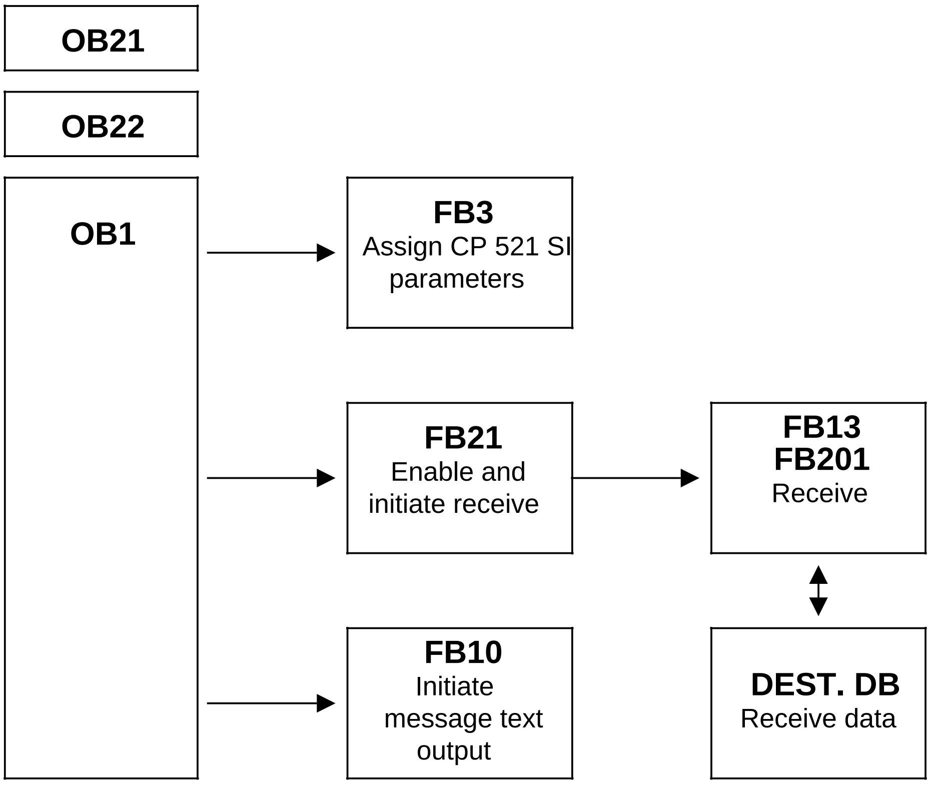


Figure 7-11. Structure of the STEP 5 Program for Data Transmission with the Terminal Driver

The receive DB (DB22), which must be installed with an adequate length before the program is started, is not listed.

```
DB2      A:TERM@@ST.S5D

0:        KS ='001:Key 1 was pressed on';
12:       KS =' ”D” at ”T”            $';
24:       KS ='002:Key 2 was pressed on';
36:       KS =' ”D” at ”T”            $';
48:       KS ='003:Key 3 was pressed on';
60:       KS =' ”D” at ”T”            $';
72:       KS ='004:Key 4 was pressed on';
84:       KS =' ”D” at ”T”            $';
96:       KS ='005:Key 5 was pressed on';
108:      KS =' ”D” at ”T”            $';
120:      KS ='006:Key 6 was pressed on';
132:      KS =' ”D” at ”T”            $';
144:      KS ='007:Key 7 was pressed on';
156:      KS =' ”D” at ”T”            $';
168:      KS ='008:Key 8 was pressed on';
180:      KS =' ”D” at ”T”            $';
192:      KS ='009:Key 9 was pressed on';
204:      KS =' ”D” at ”T”            $';
216:      KS ='010:This key provides no';
228:      KS =' code between 31H and 39';
240:      KS ='H                      $';
252:
```

```
OB 1                                B:TERM@@ST.S5D                      LEN=21

        SEGMENT  1     0000
        0000      :AN  F  100.7                        If the CP has not been assigned its
        0001      :JC  FB   3                          parameters, only FB3 is processed
        0002 NAME :CP-PARAM
        0003 BGAD :    KF +104
        0004 OK   :    F  100.7
        0005      :AN  F  100.7
        0006      :BEC
        0007      :
        0008      :JU  FB  21                          Coordinate receive
        0009 NAME :DISTRIB
        000A      :
        000B      :AN  F   80.6                        If no receive job active,
        000C      :JC  FB  10                          output message texts
        000D NAME :INIT
        000E      :
        000F      :BE
```

```
OB 21                                B:TERM@@ST.S5D                   LEN=21

        SEGMENT  1     0000
        0000      :L   KH 0000                      Defaults for parameter assignment FB
        0002      :T   FW 100
        0003      :T   FY  80
        0004      :
        0005      :BE


OB 22                                B:TERM@@ST.S5D                   LEN=7

        SEGMENT  1     0000
        0000      :JU  OB  21
        0001      :BE


FB 3                                 D:TERM@@ST.S5D                   LEN=130

        SEGMENT  1     0000
        NAME :CP-PARAM                                  Terminal Driver
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :OK        I/Q/D/B/T/C: Q  BI/BY/W/D: BI

        000B      :LW  =BGAD                        This FB uses flag words
        000C      :T   FW 102                       FW 100 to 108 to assign
        000D      :I        2                       the CP 521 SI (version A02)
        000E      :T   FW 104                       parameters.
        000F      :I        2                       FW 102 and 108 are addresses
        0010      :T   FW 106                       of the CP.
        0011      :I        2                       FW 100 is used for execution
        0012      :T   FW 108                       of the FB.
        0013      :
        0014      :DO  FW 102                       Interrogation: Is CP still
        0015      :L   IW   0                       in restart routine?
        0016      :L   KH 0F00
        0018      :AW
        0019      :L   KH 0F00
        001B      :!=F
        001C      :BEC
        001D      :***


        SEGMENT  2     001E
        001E      :A   F  100.0                     Assign parameters to first block
        001F      :JC  =BL-E
        0020      :L   KH 9000                      Job block 0
        0022      :DO  FW 102
        0023      :T   QW   0
        0024      :L   KH 0800                      Baud rate   Parity
        0026      :DO  FW 104
        0027      :T   QW   0
        0028      :L   KH 0001                      BUSY        Interface
        002A      :DO  FW 106
        002B      :T   QW   0
        002C      :L   KH 0100                      Data format  Handshake
        002E      :DO  FW 108
        002F      :T   QW   0
        0030      :
        0031      :DO  FW 102                       Wait for positive
        0032      :L   IW   0                       acknowledgement 5XXXH
        0033      :L   KH 5000
        0035      :AW
        0036      :L   KH 5000
```

```
0038      :><F
0039      :BEC
003A      :DO  FW 108                    Check whether acknowledgement
003B      :L   IW   0                    belongs to job, otherwise wait
003C      :L   KH 9000                   for right acknowledgement
003E      :><F
003F      :BEC
0040 BL-E :AN  F  100.0                  Set step flag
0041      :S   F  100.0
0042      :***


SEGMENT  3     0043
0043      :A   F  100.1                  Assign parameters to second block
0044      :JC  =BL-E
0045      :L   KH 9076                   Block 7    Terminal mode
0047      :DO  FW 102
0048      :T   QW   0
0049      :L   KH 0010                   Character delay
004B      :DO  FW 104
004C      :T   QW   0
004D      :L   KH 0000                   Receive length = 0, i.e. end-of-
004F      :DO  FW 106                    text characters are interpreted
0050      :T   QW   0
0051      :L   KH 000D                   Receive end-of-text character
0053      :DO  FW 108
0054      :T   QW   0
0055      :
0056      :DO  FW 102                    Wait for positive
0057      :L   IW   0                    acknowledgement 5XXXH
0058      :L   KH 5000
005A      :AW
005B      :L   KH 5000
005D      :><F
005E      :BEC
005F      :DO  FW 108                    Check whether acknowledgement
0060      :L   IW   0                    belongs to job, otherwise wait
0061      :L   KH 9076                   for right acknowledgement
0063      :><F
0064      :BEC
0065 BL-E :AN  F  100.1                  Set step flag
0066      :S   F  100.1
0067      :
0068      :***


SEGMENT  4     0069
0069      :A   F  100.2                  Last block deletes
006A      :JC  =BL-E                     acknowledgement 5000H.
006B      :L   KH 0000
006D      :DO  FW 102
006E      :T   QW   0
006F      :
0070      :DO  FW 102
0071      :L   IW   0
0072      :L   KH 5000
0074      :AW
0075      :L   KH 0000
0077      :><F
0078      :BEC
0079 BL-E :AN  F  100.2
007A      :S   F  100.2
007B      :S   =OK                       Set parameter assignment flag
007C      :BE
```

```
FB 10                                 D:TERM@@ST.S5D                   LEN=38

        SEGMENT  1     0000           Message text printout
        NAME :INIT

        0005       :C   DB  22                          If a key between 1 and 9
        0006       :L   FY  10                          (31H to 39H) is pressed
        0007       :L   DL   0                          on a keyboard, a message
        0008       :!=F                                 text (1 to 9) is printed out.
        0009       :BEC
        000A       :T   FY  10                          If a key outside this range
        000B       :L   KH 0040                         is pressed, message text
        000D       :>=F                                 10 is always printed out.
        000E       :JC  =M001
        000F       :L   FY  10
        0010       :L   KH 0030
        0012       :<=F
        0013       :JC  =M001
        0014       :JU  =M002
        0015 M001 :L   KH 003A
        0017       :T   DL   0
        0018 M002 :L   KH 3000                          Job 30xxH; print message text
        001A       :L   DL   0
        001B       :OW
        001C       :L   KH 0030                         Calculate number of the
        001E       :-F                                  message text
        001F       :T   QW 104
        0020       :BE
```

```
FB 13                                 D:TERM@@ST.S5D                   LEN=102

        SEGMENT  1     0000           Receiving 12 bytes
        NAME :RECEIVE

        0005       :C   DB  22                          Open receive DB
        0006       :
        0007       :L   IW 104                          Is CP busy?
        0008       :L   KH 0F00
        000A       :AW
        000B       :L   KH 0F00
        000D       :!=F
        000E       :BEC
        000F       :
        0010       :A   F   80.4                        Receive job already
        0011       :JC  =M001                           initiated
        0012       :
        0013       :L   IW 104                          Check whether job can be
        0014       :L   KH C000                         initiated
        0016       :AW
        0017       :L   KH 0000
        0019       :><F
        001A       :BEC
        001B       :
        001C       :L   KH A080                         Transfer receive job
        001E       :T   QW 104
        001F       :S   F   80.4
        0020       :R   F   80.5                        Reset error flag
        0021       :BEU
        0022       :
```

```
0023 M001 :L   IW 104                          Check acknowledgement block 1
0024      :L   KH F0FF
0026      :AW
0027      :L   KH 6001
0029      :><F
002A      :JC  =M002
002B      :
002C      :L   IW 106                          Read in data word 0
002D      :T   DW   0
002E      :L   IW 108                          Read in data word 1
002F      :T   DW   1
0030      :L   IW 110                          Read in data word 2
0031      :T   DW   2
0032      :
0033      :L   KH C001                         Acknowledge message block 2
0035      :T   QW 104
0036      :BEU
0037      :
0038 M002 :L   IW 104                          Check acknowledgement block 2
0039      :L   KH F0FF
003B      :AW
003C      :L   KH 6002
003E      :><F
003F      :JC  =M003
0040      :
0041      :L   IW 106                          Read in data word 3
0042      :T   DW   3
0043      :L   IW 108                          Read in data word 4
0044      :T   DW   4
0045      :L   IW 110                          Read in data word 5
0046      :T   DW   5
0047      :
0048      :L   KH C002                         Acknowledge message block 2
004A      :T   QW 104
004B      :BEU
004C      :
004D M003 :L   IW 104                          Check terminating acknowledgement
004E      :L   KH F00F
0050      :AW
0051      :L   KH 5000
0053      :><F
0054      :JC  =M004
0055      :
0056      :AN  I  106.0                        No error?
0057      :JC  =M005
0058      :
0059      :AN  F   80.5
005A      :S   F   80.5
005B M005 :R   F   80.6
005C      :R   F   80.4
005D      :L   KH 0000                         Delete terminating acknowledgement
005F      :T   QW 104
0060 M004 :BE
```

```
FB 14                                D:TERM@@ST.S5D                  LEN=28

        SEGMENT  1     0000          Calculating the message frame length

        This program calculates the length of a message frame in bytes;
        the values required are the number of blocks from IW0 and the number
        of valid bytes in the last message frame.
        NAME :MESS LEN

        0005      :L   FW 242                          Number of blocks from FB 201
        0006      :L   KH 00FF                         Mask out 6X
        0008      :AW
        0009      :L   KH 0002                         Subtract 2
        000B      :-F
        000C      :T   FW 252
        000D      :SLW      2
        000E      :L   FW 252
        000F      :+F
        0010      :L   FW 252
        0011      :+F
        0012      :DO  FW 238
        0013      :L   IB   1                          Add bytes from last block
        0014      :+F
        0015      :T   FW 252                          Number of bytes of the message frame
        0016      :BE

FB 21                                B:TERM@@ST.S5D                  LEN=21

        SEGMENT  1     0000
        NAME :DISTRIB

        0005 M001 :AN  F   80.6                        Receive not active
        0006      :JC  =M002
        0007      :
        0008      :JU  FB  13                          Process receive FB
        0009 NAME :RECEIVE
        000A      :
        000B      :BEU
        000C      :
        000D M002 :AN  F   80.6                        Reset receive flag
        000E      :S   F   80.6                        Set receive flag
        000F      :BE
```

```
FB 201                                D:TERM@@ST.S5D                    LEN=168

        SEGMENT  1      0000
        NAME :REC  VAR
        DES  :BGAD       I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Z-DB       I/Q/D/B/T/C: B
        DES  :ZANF       I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

        000E      :LW  =BGAD                           Load and store module address
        000F      :T   FW 234
        0010      :
        0011      :DO  FW 234                          Is CP busy?
        0012      :L   IW   0
        0013      :L   KH 0F00
        0015      :AW
        0016      :L   KH 0F00
        0018      :!=F
        0019      :BEC
        001A      :***


        SEGMENT  2      001B
        001B      :A   F   80.4                        Flag for job already
        001C      :JC  =M001                           initiated
        001D      :
        001E      :DO  FW 234                          Check whether a job can be
        001F      :L   IW   0                          initiated
        0020      :L   KH C000
        0022      :AW
        0023      :L   KH 0000
        0025      :><F
        0026      :BEC
        0027      :
        0028      :L   FW 234                          Set address pointer
        0029      :L   KF +2
        002B      :+F
        002C      :T   FW 236                          Base address + 2 (word 1)
        002D      :L   KF +2
        002F      :+F
        0030      :T   FW 238                          Base address + 4 (word 2)
        0031      :L   KF +2
        0033      :+F
        0034      :T   FW 240                          Base address + 6 (word 3)
        0035      :
        0036      :L   KH A080                         Initiate receive job
        0038      :DO  FW 234
        0039      :T   QW   0
        003A      :L   KH 6001                         Read acknowledgements
        003C      :T   FW 242                          into flag words
        003D      :L   KH C001
        003F      :T   FW 244
        0040      :LW  =ZANF                           Set pointer to destination data
        0041      :T   FW 246                          Destination starting address
        0042      :L   KF +1
        0044      :+F
        0045      :T   FW 248                          Starting address + 1
        0046      :L   KF +1
        0048      :+F
        0049      :T   FW 250                          Starting address + 2
        004A      :
        004B      :S   F   80.4                        Set flag for receive job initiated,
        004C      :R   F   80.5                        reset error flag, block end
        004D      :BEU                                 unconditionally
        004E M001 :***
```

```
SEGMENT  3     004F
004F      :DO  FW 234                    Load receive word 0 and
0050      :L   IW   0                    check with receive block
0051      :L   KH F0FF                   present
0053      :AW
0054      :L   FW 242
0055      :><F
0056      :JC  =M001
0057      :DO  =Z-DB                     Open destination DB
0058      :DO  FW 236
0059      :L   IW   0
005A      :DO  FW 246                    Word for
005B      :T   DW   0                    data word pointer 0
005C      :
005D      :DO  FW 238
005E      :L   IW   0
005F      :DO  FW 248                    Word for
0060      :T   DW   0                    data word pointer 1
0061      :
0062      :DO  FW 240
0063      :L   IW   0
0064      :DO  FW 250                    Word for
0065      :T   DW   0                    data word pointer 2
0066      :
0067      :L   FW 244                    Acknowledge data transfer
0068      :DO  FW 234
0069      :T   QW   0
006A      :
006B      :L   FW 242                    Update block message
006C      :L   KF +1
006E      :+F
006F      :T   FW 242
0070      :
0071      :L   FW 244                    and block acknowledgement
0072      :L   KF +1
0074      :+F
0075      :T   FW 244
0076      :
0077      :L   FW 246                    Increment data word pointer word 0
0078      :L   KF +3
007A      :+F
007B      :T   FW 246
007C      :
007D      :L   FW 248                    Increment data word pointer word 1
007E      :L   KF +3
0080      :+F
0081      :T   FW 248
0082      :
0083      :L   FW 250                    Increment data word pointer word 2
0084      :L   KF +3
0086      :+F
0087      :T   FW 250
0088      :BEU
0089 M001 :***


SEGMENT  4     008A
008A      :DO  FW 234                    Check whether terminating
008B      :L   IW   0                    acknowledgement present
008C      :L   KH F00F
008E      :AW
008F      :L   KH 5000
0091      :><F
0092      :JC  =M001
```

```
0093      :DO  FW 236
0094      :L   IW   0
0095      :L   KH 0100
0097      :AW
0098      :JZ  =M002
0099      :AN  F   80.5                        Job terminated with
009A      :S   F   80.5                        error
009B M002 :R   F   80.6                        Enable for new send or
009C      :R   F   80.4                        receive job
009D      :
009E      :L   KH 0000                         Delete terminating acknowledgement
00A0      :DO  FW 234
00A1      :T   QW   0
00A2 M001 :BE
```

8 3964(R) Driver

8.1	Interface Lines for the 3964(R) Driver	8 - 1
8.2	Special Features of the 3964(R) Data Transmission Protocol	8 - 6
8.2.1	General Points on Data Transmission Procedures Using Protocols	8 - 6
8.2.2	The 3964(R) Transmission Protocol	8 - 6
8.3	Assigning the CP 521 SI Parameters in 3964(R) Mode	8 - 13
8.3.1	Assigning the CP 521 SI Parameters with the Memory Submodule	8 - 14
8.3.2	Assigning the CP 521 SI Parameters in the User Program	8 - 20
8.4	Data Transfer with the 3964(R) Transmission Protocol	8 - 23
8.4.1	Sending Data with the 3964(R) Transmission Protocol	8 - 23
8.4.2	Receiving Data with the 3964(R) Transmission Protocol	8 - 28
8.5	Error Flags in the CBS and CBR of the CP Acknowledgements	8 - 32
8.6	STEP 5 Programs for Data Transmission with the 3964(R) Driver	8 - 35

Figures

8-1. Terminal Diagram CP 521 SI - CP 521 SI (TTY Interface) . . . . 8 - 3
8-2. Terminal Diagram CP 521 SI (TTY Passive) - CP 523 (TTY Active) . . . . 8 - 3
8-3. Terminal Diagram CPU 944 (TTY Active) - CP 521 SI (TTY Passive) . . . . 8 - 4
8-4. Terminal Diagram CP 521 SI (TTY Active) - CP 521 SI (TTY Active) . . . . 8 - 5
8-5. CP 521 SI - CP 521 SI as an Example of Zero Modem Mode . . . . 8 - 5
8-6. Example of an 11-Bit Character Frame . . . . 8 - 7
8-7. Error-Free Data Interchange (Send) . . . . 8 - 9
8-8. Error-Free Data Reception . . . . 8 - 10
8-9. Data Traffic with Errors . . . . 8 - 11
8-10. Resolving an Initiation Conflict . . . . 8 - 12
8-11. Schematic for Entering Parameter Blocks in DB1 . . . . 8 - 14
8-12. Sending Data over the CP 521 SI . . . . 8 - 23
8-13. Receiving Data over the CP 521 SI . . . . 8 - 28
8-14. Program Structure for Sending and Receiving with 3964(R) . . . . 8 - 35

Tables

8-1. Pin Assignments of the 25-Pin Subminiature D Socket Connector of the CP 521 SI . . . . 8 - 2
8-2. Parameter Blocks for ASCII Mode . . . . 8 - 13
8-3. Contents of the Parameter Blocks on the Memory Submodule . . . . 8 - 15
8-4. Assigning the 3964(R) Communications Protocol Parameters on the Memory Submodule . . . . 8 - 19
8-5. Transfer Memory Assignments in the Case of the ”Transfer Parameter Assignment Data for Parameter Block 0” Job Request . . . . 8 - 21
8-6. Contents of Transfer Memory in the Case of Job Request ”9073_H” . . . . 8 - 22
8-7. Contents of Transfer Memory in the Case of the Follow-Up Job Request ”$907A_H$” . . . . 8 - 22
8-8. Status Message After the Send Job Request ”$A001_H$” . . . . 8 - 32
8-9. Status Message After the Receive Job Request ”$A080_H$” . . . . 8 - 32
8-10. Error Flags in the Coordination Byte ”Send” **(CBS)** in the Case of ”3964(R)” . . . . 8 - 33
8-11. Error Flags in the Coordination Byte ”Receive” **(CBR)** in the Case of ”3964(R)” . . . . 8 - 34

8 3964(R) Driver

The 3964(R) driver enables data transmission according to the standardized 3964(R) protocol. This chapter tells you how the data is transmitted, which devices you require, how to set these devices (assign their parameters) and how to program data transmission.

The possible applications of the 3964(R) driver depend on the following:

- The peripheral device connected
- The hardware interface used.

Data transmission according to the standardized transmission protocol functions satisfactorily with all peripheral devices that also have a 3964(R) driver. In the "3964(R) driver" mode, the CP 521 SI assumes that a peripheral device with an RS-232C (V.24) or TTY interface is connected to the serial interface.

You can connect the following devices:

- MODEM
- A second CP 521 SI
- CP 523
- CP 524/CP 525-2 (in conjunction with special driver 6ES5 897-2AB11)
- CP 544
- CPU 928B
- CPU 944 (using the operating system with 3964(R) driver)
- PC (e.g. in conjunction with the PRODAVE DOS 64R software toolbox, Order No. 6ES5 897-2UD11)

8.1 Interface Lines for the 3964(R) Driver

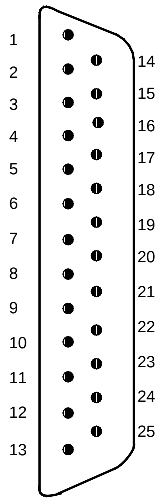
In the ”3964(R) driver” mode, only a few interface lines of the 25-pin serial interface are used. In this subsection, you will discover which interface lines these are and we will give you a few typical examples for connecting peripheral devices.

The terminal diagrams used as examples take the following into account:

- Connections with current interface (TTY)
- Connections with voltage interface (RS-232C (V.24))

The CP 521 SI is equipped with a serial interface which can be switched from a current interface (TTY) to a voltage interface (RS-232C (V.24)) by assigning the relevant parameters (Section 8.3). The lines of both interfaces connect to a 25-pin subminiature D socket connector. Those interface lines which are used for data transmission according to the 3964(R) protocol are set against a grey-shaded background in Table 8-1.

Table 8-1. Pin Assignments of the 25-Pin Subminiature D Socket Connector of the CP 521 SI

View	Pin No.	Signal Name	Meaning
	1	-	Disabled
	2	TxD	Send data (V.24)
	3	RxD	Receive data (V.24)
	4	RTS	Request to send (V.24)
	5	CTS	Clear to send (V.24)
	6	DSR	Data set ready (V.24)
	7	GND	Signal ground (RS-232C (V.24))
	8	-	Disabled
	9	TTY IN+	TTY receive line +
	10	TTY IN -	TTY receive line -
	11	-	Disabled
	12	-	Disabled
	13	P24	+24 V for active TTY
	14	-	Disabled
	15	-	Disabled
	16	-	Disabled
	17	20 mA	Current source TTY *****
	18	TTY OUT+	TTY send line +
	19	20 mA	Current source TTY*****
	20	DTR	Terminal ready
	21	TTY OUT -	TTY send line -
	22	-	Disabled
	23	-	Disabled
	24	-	Disabled
	25	-	Disabled

***** If +24 V to GDN (pin 7) on pin 13

Note

Please note when connecting with RS-232C (V.24) interface!
The 3964(R) driver does not support control signal lines (DSR, DTR, CTS, RTS). Hardware handshake is therefore not possible.

Terminal diagrams for peripheral devices (examples)

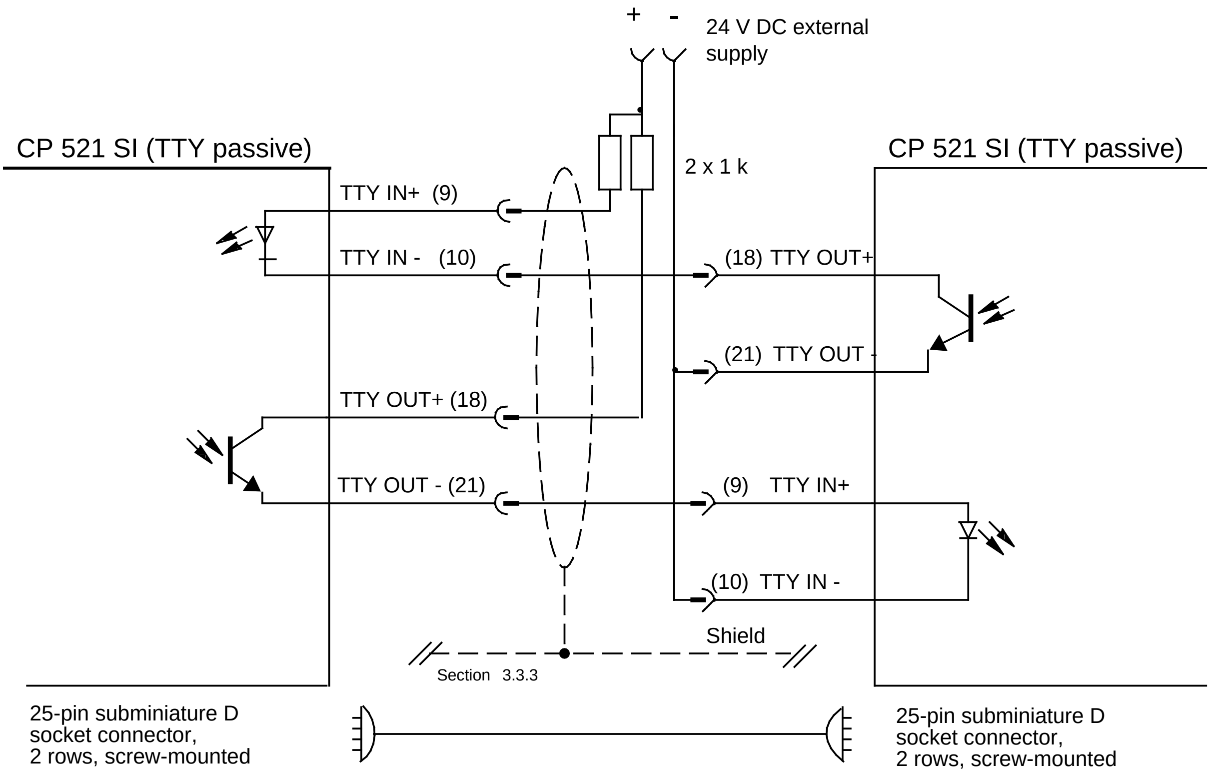


Figure 8-1. Terminal Diagram CP 521 SI - CP 521 SI (TTY Interface)

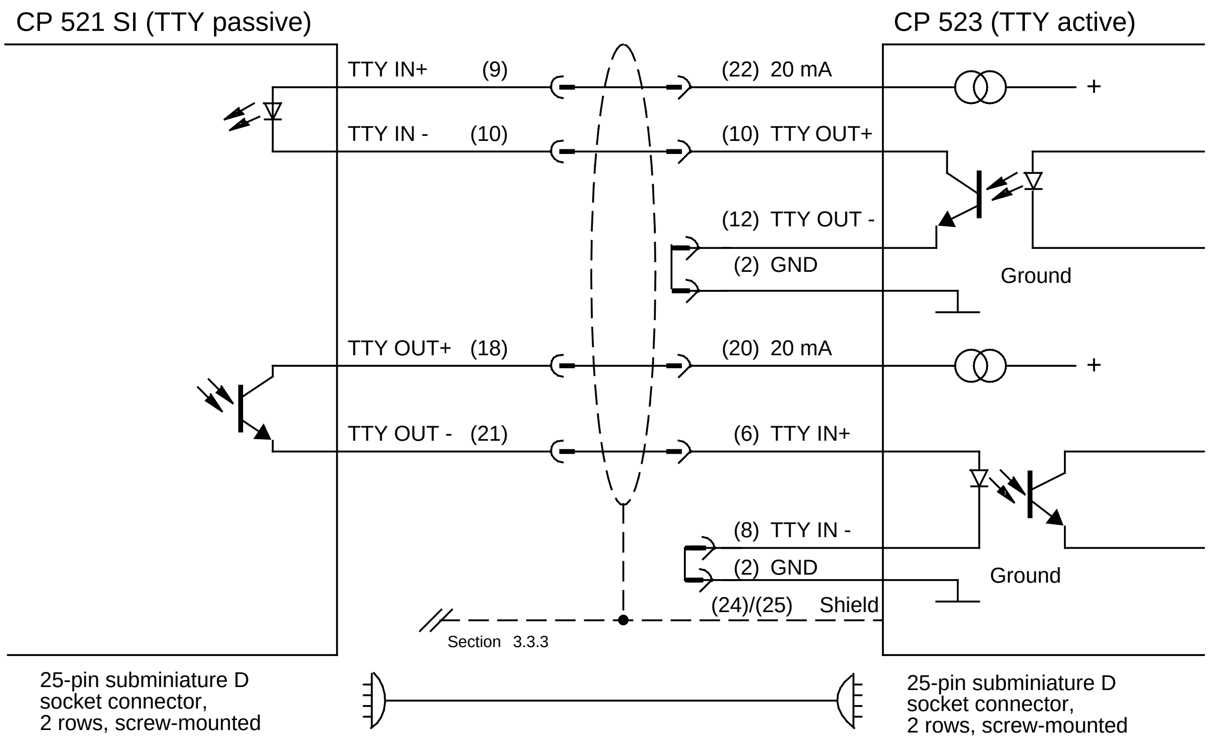


Figure 8-2. Terminal Diagram CP 521 SI (TTY Passive) - CP 523 (TTY Active)

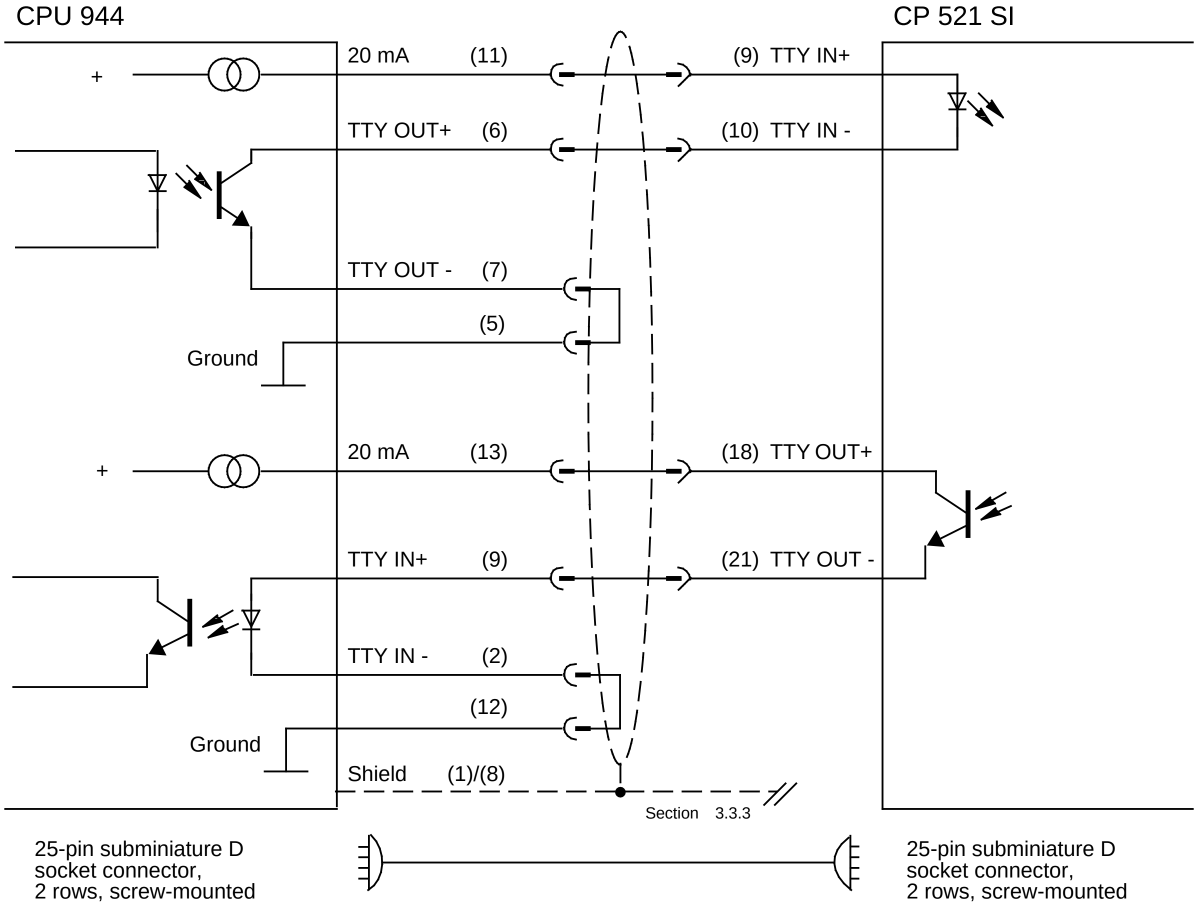


Figure 8-3. Terminal Diagram CPU 944 (TTY Active) - CP 521 SI (TTY Passive)

Active mode of the TTY interface of the CP 521 SI

If the CP 521 SI is operated in active mode (24 V DC external supply), disturbance voltages >500 V can lead to corruption of the data. In order to keep interference to a minimum, the following measures should be observed.

1. Each sender should be operated in active mode (24 V DC external supply).
2. Use an independent, stable 24 V DC supply for external supply.
3. Keep the connecting cables to the 24 V external supply as short as possible.
4. Connect the current source (pin 19) directly to the send transistor of the sender (Fig. 8.4)

Terminal Diagram CPU 944 (TTY Active) - CP 521 SI (TTY Passive)

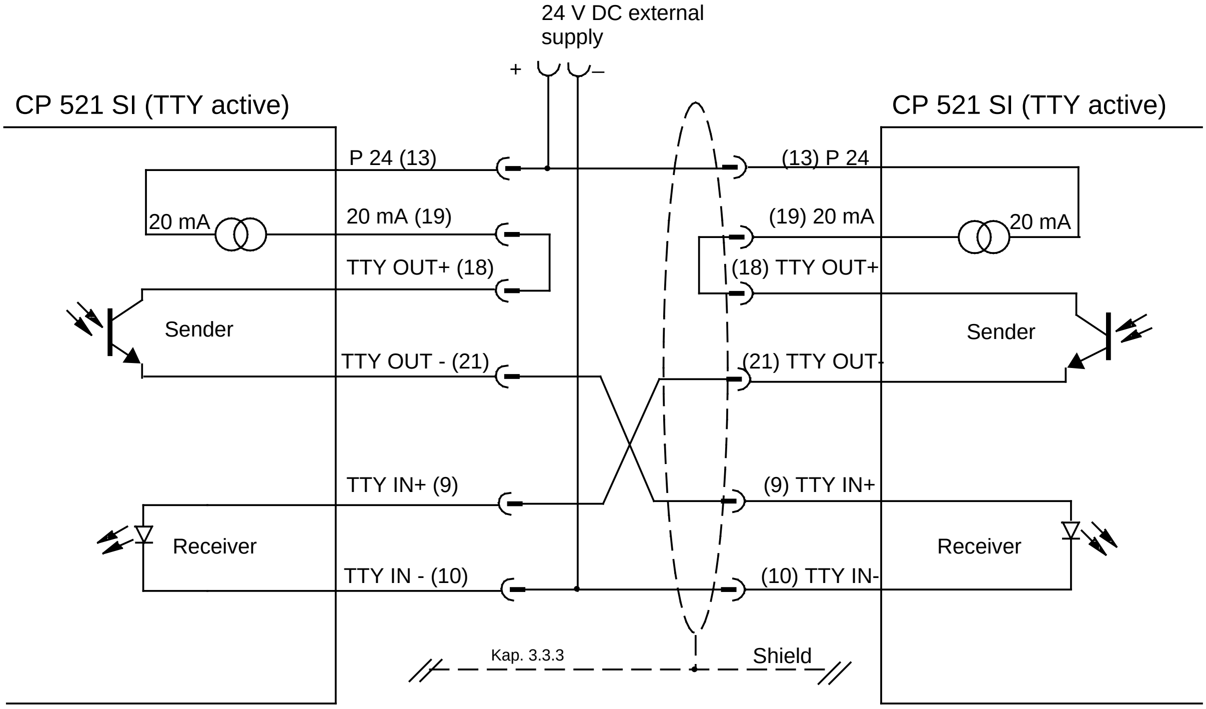


Figure 8.4 Terminal Diagramm CP 521 SI (TTY Active) - CP 521 SI (TTY Active)

Zero modem mode: **CP 521 SI - CP 521 SI as an example of the connection of two data terminal devices**

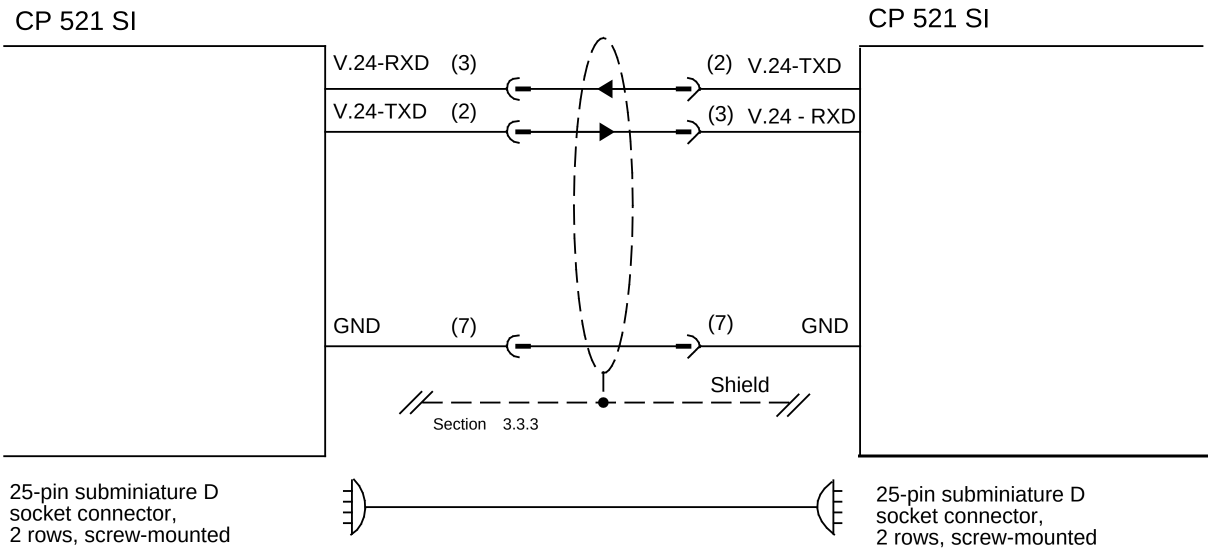


Figure 8-5. CP 521 SI - CP 521 SI as an Example of Zero Modem Mode

8.2 Special Features of the 3964(R) Data Transmission Protocol

The 3964(R) data transmission protocol is a comparatively reliable program for electronic data exchange between the CP 521 SI and a peripheral device because the data transmission is handled by a standardized protocol.

To be able to start up the CP 521 SI, it is necessary to read the following explanations. We explain the protocol in order to give a better understanding of data transmission using the 3964(R).

8.2.1 General Points on Data Transmission Procedures Using Protocols

A large number of conventions must be agreed upon for a data transmission procedure; codes, operating modes, baud rates and the algorithmic sequence of transmission. The establishment of this algorithmic sequence is referred to as a transmission protocol (or protocol for short). In general, a transmission protocol defines the following phases of data transmission:

- Request for data exchange from A to B
- Data exchange
- Termination of the data exchange

The transmission protocol is essentially a matter for the CP 521 SI. This means that the module handles data transmission autonomously according to this protocol.

8.2.2 The 3964(R) Transmission Protocol

Data transmission with protocol means that the actual data to be transmitted is enclosed within control characters.

The 3964(R) driver allows comparatively reliable data transmission due to the fact that the receiver must first signal to the sender its readiness to receive (connection buildup) and must acknowledge correct reception after successful data exchange. Data reliability is increased in the case of the 3964(R) transmission protocol by sending an additional block check character.

The 3964(R) driver interprets the following control characters:

- DLE (10_H) Data Link Escape
- STX (02_H) Start of Text
- NAK (15_H) Negative Acknowledgement
- ETX (03_H) End of Text

During the parameter assignment phase, you can also specify whether the data is to be transferred with or without a block check character. The block check character (BCC) increases transmission integrity. A distinction is made between 3964R and 3964 mode, depending on whether you want to transfer data with or without a block check character.

- With block check character: 3964R
- Without block check character: 3964

Protocol data

The 3964 and 3964R transmission protocols control the flow of data between your programmable controller and a peripheral device.

The data to be transmitted must be entered in the CP 521 SI's output buffer, and is then forwarded to the peripheral device together with the 3964 or 3964R protocol. The line protocol retries the transmission where necessary; fatal errors are flagged in the coordination byte.

Data coming from the peripheral device is entered in input buffers. If the data is received without error, it can be fetched by the CPU.

The 3964 and 3964R protocols are asynchronous, bit-serial transmission procedures. All parameters on the module and on the peripheral device, except for the **priority,** must be identical.

Control information and useful data are transmitted over the interface lines. In order to enable the receiver to recognize each character and to be able to check for error-free transmission, additional bits are prefixed or appended to each character transmitted. The character frame parameters are assigned in parameter block 0.

Example: 11-bit character frame (1 start bit, 8 data bits, 1 parity bit, 1 stop bit)

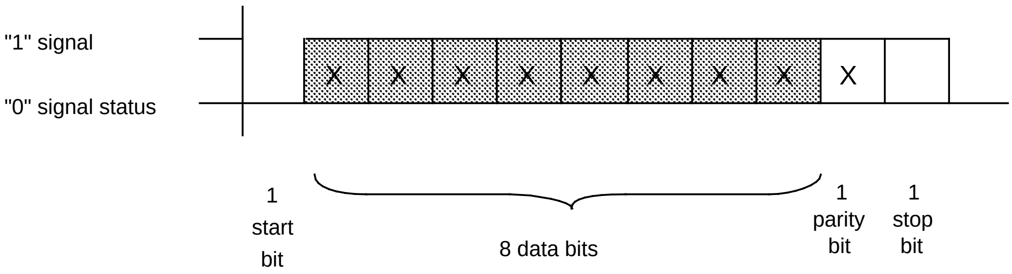


Figure 8-6. Example of an 11-Bit Character Frame

When the **3964R transmission protocol** is used, a **block check character (BCC)** is transmitted at the end of each message block to increase transmission reliability and ensure data integrity. The BCC is the even longitudinal parity (EXORing of all data bytes) of the frame transmitted or received. Its generation begins with the first useful data byte (first byte in the frame) following connection buildup, and ends on connection cleardown after the DLE ETX characters.

Sending

In order to **build up a connection**, the CP transmits the **STX control character** (start of text). If the peripheral device replies with DLE (data link escape) before time-out (QVZ) *, the protocol goes to Send mode. If the peripheral device responds with NAK (negative acknowledgement) or a random character (other than DLE), or if no response comes before timeout, the connection cannot be established. After a total of six ** unsuccessful attempts, the procedure is aborted and the cause of error flagged in coordination byte CBS.

If the connection can be established, the useful data in the send buffer is transmitted to the peripheral device at the specified transmission rate. The peripheral device monitors the interval between incoming characters, which must not exceed the **character delay baud (ZVZ) *****.

Each DLE control character found in the buffer is transmitted as two DLE characters (**double DLE**), i.e.10_H is transmitted twice.

When the buffer is empty, the CP appends the following as end-of-text identifier and waits for an acknowledgement:

- In the 3964 transmission protocol: the **DLE ETX** characters
- In the 3964R transmission protocol: the **DLE ETX BCC** characters

A DLE character from the peripheral device within the allotted time (QVZ) indicates that the data was received without error.

If the peripheral device responds with NAK, a random character, or not at all, the CP once again transmits STX. After a total of six unsuccessful attempts ** to transmit the message frame, the CP aborts the procedure, sets an error flag in the CBS, and transmits NAK to the peripheral device.

If the peripheral device sends a NAK character during an on-going transmission, the CP aborts the message frame, and attempts a retry as described above. If the partner transmits another character, the CP waits until the character delay (ZVZ) has expired, and then sends NAK to bring the peripheral device to the idle state before once again trying to transmit STX.

* Default value in 3964(R) mode: 2000 ms
** Default value: other values can be entered in parameter block 7
*** Default value: 100 ms

Example of error-free transmission:

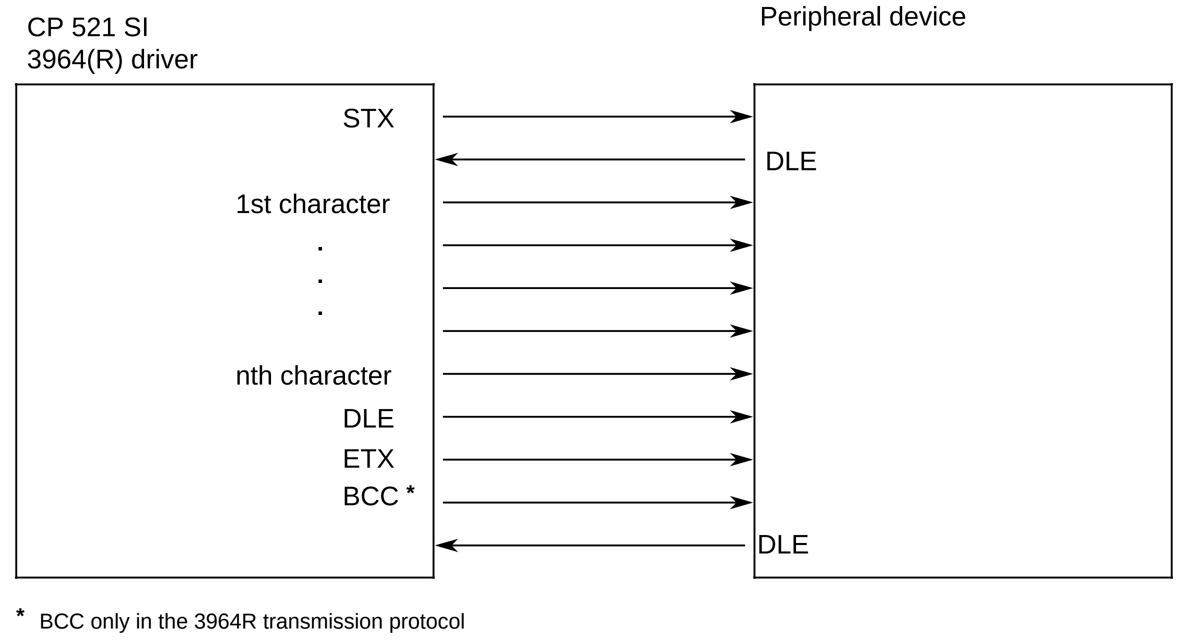


Figure 8-7. Error-Free Data Interchange (Send)

Receiving

When the 3964(R) driver has no send job request pending, the CP waits for the peripheral device to establish a connection.

If the CP receives a character (other than STX) while in the idle state, it waits until the character delay has expired (ZVZ), and then sends NAK.

If the CP receives an STX from the peripheral device and the input buffer is not full, it responds with DLE. Incoming characters are now entered in the input buffer. If two DLE characters are received in succession, only one is entered in the input buffer.

After each character, the CP waits until the character delay (ZVZ) has elapsed. If no character is received during this time, the CP sends a NAK to the peripheral device.

The CP terminates reception in different ways, depending on whether it is set to the 3964 or 3964R transmission protocol:

- With the 3964 transmission protocol (i.e. without BCC):
  If the CP detects the character string DLE ETX, it terminates reception and sends DLE to the peripheral device for a frame received without errors (or NAK for a frame received with errors).

- With the 3964R transmission protocol (i.e. with BCC):
  If the CP detects the character string DLE ETX BCC, it terminates reception. It compares the received block check character BCC with the internally generated longitudinal parity. If the block check character is correct and no other receive errors have occurred, the CP sends DLE. If the BCC is not correct, the CP sends NAK to the peripheral device, and waits for a retry. If the block cannot be received correctly after a total of six attempts *, or if the peripheral device does not initiate a retry within the block waiting time of 4 s *, the CP aborts reception.

If transmission errors (lost characters, frame error, parity error) occur during reception, the CP continues receiving until the connection is cleared down, transmits NAK to the peripheral device, and waits for a retry as described above.

Example of error-free reception:

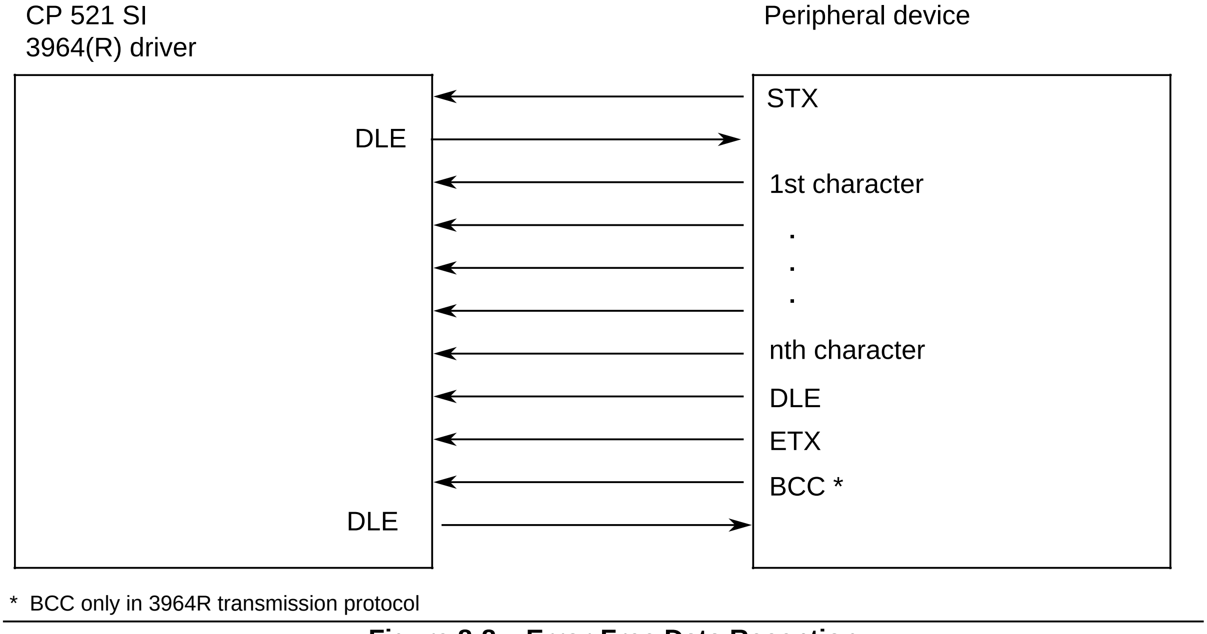
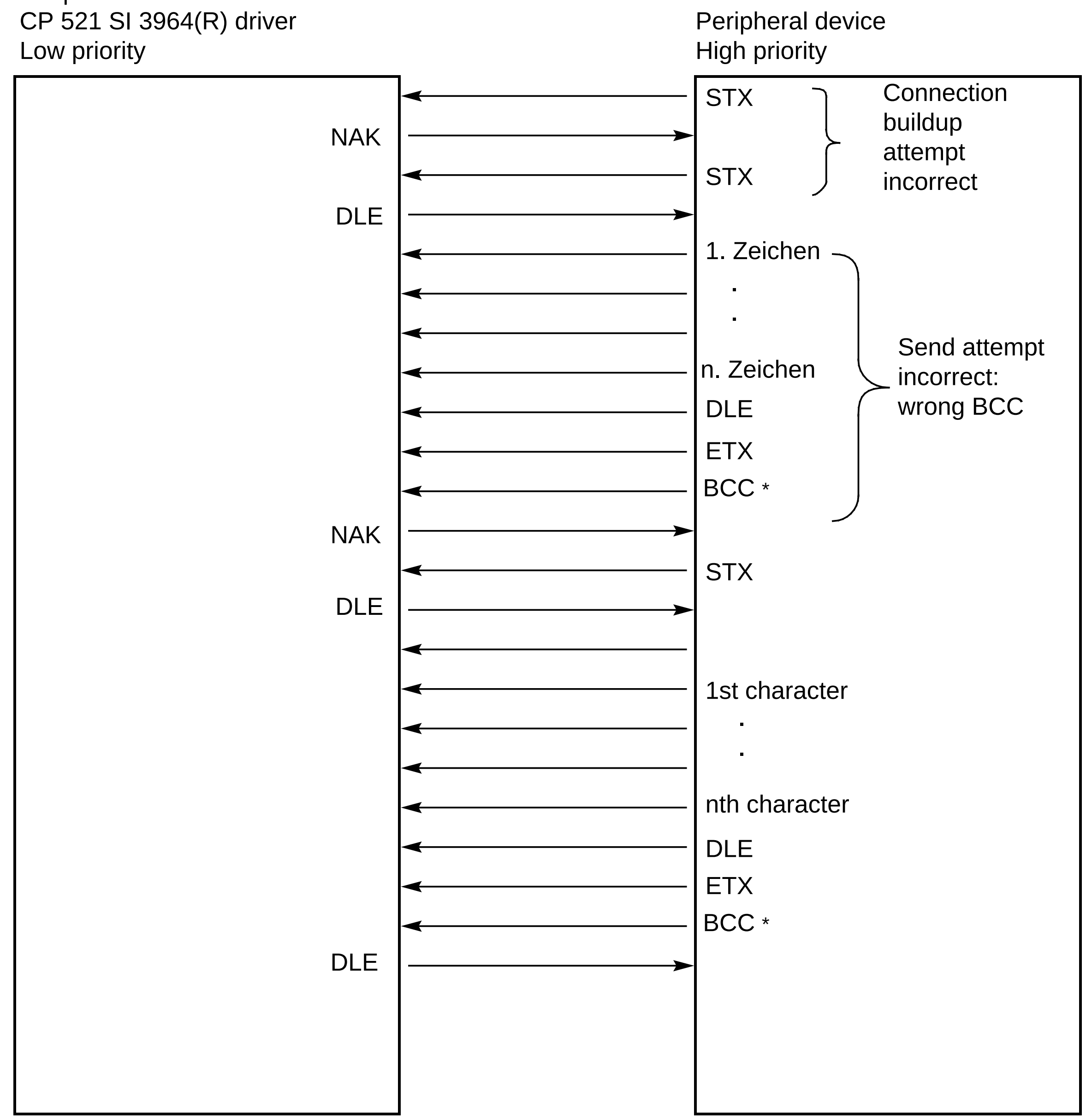


Figure 8-8. Error-Free Data Reception

***** Default value: other parameter values can be assigned in parameter block 7

Example of data traffic with errors:



* BCC only with 3964R transmission protocol

Figure 8-9. Data Traffic with Errors

Initiation conflict

If one partner in the link replies to a line bid (STX character) from the other link partner within the character delay (QVZ) with STX instead of DLE or NAK, the result is an initiation conflict, i.e. both link partners want to transmit. The low-priority partner withdraws its line bid and sends DLE. The high-priority partner sends its data as described above. Following connection cleardown, the low priority partner can send its data.

The ”Priority” parameter is set when assigning parameters in parameter block 7 in 3964R mode. Make sure that one link partner is assigned high priority and the other low priority.

Example for resolving an initiation conflict:

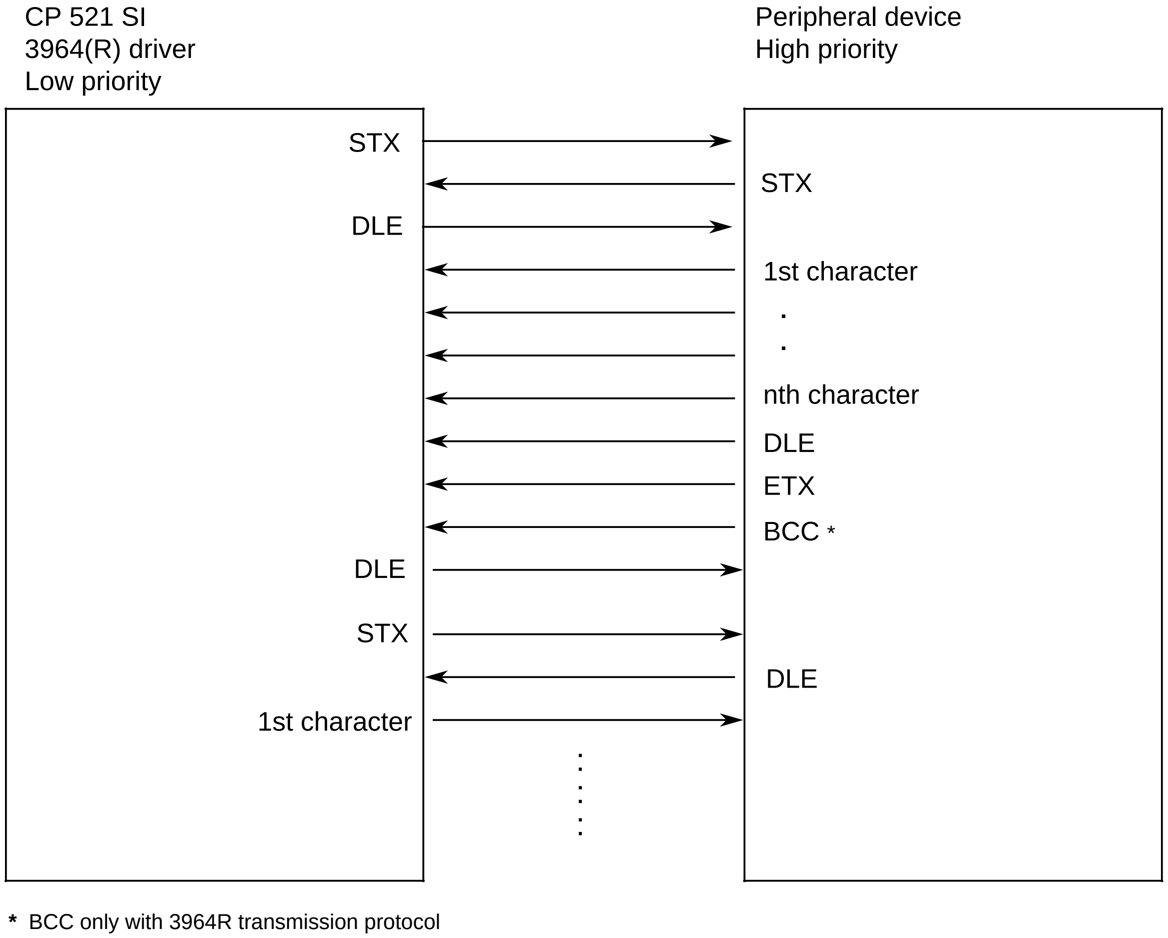


Figure 8-10. Resolving an Initiation Conflict

Protocol errors

The transmission protocol can detect errors caused by incorrect behaviour on the part of the peripheral device caused by disturbances or interference on the line.
In both cases, an attempt is first made to transmit or receive the message frame without error by initiating a retry. If this is still impossible after the maximum number of attempts has been made (or if another error occurs), the protocol aborts transmission or reception. An error code is entered in the coordination byte and the CP enters the idle state.

8.3 Assigning the CP 521 SI Parameters in 3964(R) Mode

On delivery, the CP 521 SI has already been assigned default parameters. If you wish to use other values in 3964(R) mode, you must assign the CP 521 SI the appropriate parameters.

If you want to assign other parameters to the CP 521 SI, there are various ways of transferring the parameter assignment data to the CP 521 SI:

1. Store the parameter assignment data in data block (DB) 1 on a memory submodule and plug the configured memory submodule into the CP 521 SI.
2. Transfer the parameter assignment data to the CP 521 SI in the user program. You do not require a memory submodule to operate the CP 521 SI.

In order to make transfer of the parameter assignment data as simple as possible, the parameter assignment data is divided into parameter blocks.

Table 8-2. Parameter Blocks for ASCII Mode

Parameter Block Number	Parameter
0	Parameters of the serial interface
7	Setting of the 3964(R) driver
9	Clock correction value

Depending on whether you want to assign your CP 521 SI parameters with a memory submodule or in the user program, read either Section 8.3.1 or 8.3.2.

- Section 8.3.1: Assigning the CP 521 SI parameters with the memory submodule
- Section 8.3.2: Assigning the CP 521 SI parameters in the user program

Note

If there is no memory submodule plugged into the CP 521 SI and the CP 521 SI has a backup battery, the module is automatically assigned the parameters stored in the RAM on POWER-ON (i.e. the parameters last assigned). In other words, the module is activated in the same driver mode following POWER-ON as it was before POWER-OFF.

8.3.1 Assigning the CP 521 SI Parameters with the Memory Submodule

You can store the parameter assignment data in DB1 on a memory submodule. The procedure is almost identical to that for the printer driver mode, the only difference being parameter block 7 in which additional data must be specified in the 3964(R) driver mode.

> **Note**
>
> The parameter assignment data in the user program has priority over the data on the memory submodule. Following POWER-ON, therefore, the parameter assignment data in the memory submodule is valid, provided it has not been overwritten by the user program.

Entering the parameter blocks on the memory submodule

If you have decided to use parameters adapted you your particular system, and not the defaults, start entering the parameter blocks as follows:

Plug the memory submodule into the programmer.
Transfer the complete contents of the memory module to the programmer (diskette, hard disk).
Erase the memory submodule.
Enter all the parameter blocks that you which to change in DB1.
Transfer the entire memory submodule contents back to the memory submodule.

The parameter blocks can be entered in DB1 according to the following schematic:

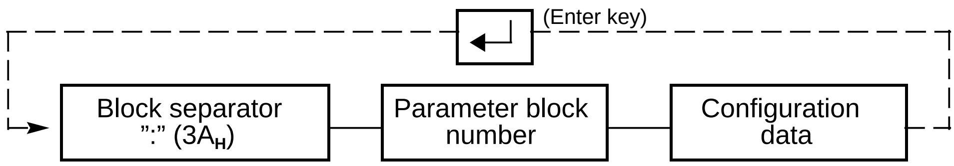


Figure 8-11. Schematic for Entering Parameter Blocks in DB1

Further tips for operator inputs:

- When entering text, alternate between lines in KS format and lines in S format.
- You can also enter comments in addition to the parameter blocks.

Once you have stored the parameter assignment data in DB1 in the memory submodule, insert the memory submodule in its receptacle on the CP 521 SI. The next time the module is powered up, the parameters on the submodule are checked and transferred to the CP 521 SI.

The following pages explain the contents of the parameter blocks and include an example of how to enter the parameters.

Table 8-3. Contents of the Parameter Blocks on the Memory Submodule

Block	Parameter		Value Range	Default Values on the CP 521 SI
0	Baud rate	110 bit/s 200 bit/s 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s	1 2 3 4 5 6 7 8	8 9600 bit/s
	Parity	even odd "mark" "space" no check	0 1 2 3 4	0 even
	BUSY signal*	no	0	0 no
	Interface	TTY RS-232C (V.24)	0 1	0 TTY
	Data format: 11-bit character:	7 data bits (with parity) 8 data bits (with parity) 8 data bits (without parity)	0 1 2	0 7 data bits (with parity)
	10-bit character:	7 data bits (without parity) 7 data bits (with parity) 8 data bits (without parity)	3 4 5	
	Hardware handshake**	OFF	0	0 OFF
7	Mode	3964(R) Driver	3	0: (Memory submodule with message texts plugged in) 1: (without memory submodule) ***
	Character delay (ZVZ)		1_D to 65535_D (·10 ms)	10_D (·10 ms)
	Timeout (QVZ)		1_D to 65535_D (·10 ms)	200_D (·10 ms)
	Block waiting time (BWZ)		1_D to 65535_D (·10 ms)	400_D (·10 ms)

* Busy signal irrelevant for the 3964(R) driver

** Hardware handshake irrelevant for the 3964(R) driver

*** Only if there is no battery backup; if a backup battery has been installed, the same driver mode is active after POWER-ON as before POWER-OFF

Table 8.3. Parameter Block Assignments on the Memory Submodule (Cont.)

Block	Parameter	Value Range	Default Values on the CP 521 SI
7	Transmission **with** block check character (3964R) (with BCC) Transmission **without** block check character (3964) (without BCC)	1 0	0 without BCC
	Priority low high	0 1	1 high priority
	Buildup attempts	1_D to 255_D	6_D
	Number of send attempts	1_D to 255_D	6_D
9	Correction value (s/month)	-400_D to $+400_D$ (s/month)	0_D

Explanation of Parameters

Baud rate (parameter block 0)

You have a choice of eight baud rates. The default is 9600 bit/s. If you are using the RS-232C (V.24) interface, you must take the load capacitance of cables longer than 15 m into account. Longer cables can generally be used if the baud rate is reduced.

Parity (parameter block 0)

You can choose between five types of parity.

- Even parity
  The parity bit is set so that the sum of the data bits (incl. parity bit) that are ”1” 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 always has signal state ”1”.
- ”Space”
  The parity bit always has signal state ”0”.
- No parity check
  The signal state of the parity bit is not significant. Parity is not checked when receiving; however, when sending, the parity bit is always set to ”1”.

The default is even parity.

BUSY signal (parameter block 0)

This parameter is not relevant to the 3964(R) driver. Leave the default value ”0” unchanged (no BUSY signal) in parameter block 0.

Interface (parameter block 0)

You can choose between the RS-232C (V.24) and TTY interfaces here. See Section 2.3 for more detailed information on these interfaces.

The TTY interface is the default.

Data format (parameter block 0)

Characters are transmitted between the CP 521 SI and the peripheral device in a 10-bit or 11-bit character frame. You can choose between seven and eight data bits within these character frames (Figures 4-4 and 4-5).

10-bit character frame:

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

11-bit character frame:

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

The 11-bit character frame (1 start bit, 7 data bits, 1 parity bit, 2 stop bits) is the default.

Hardware handshake (parameter block 0)

This parameter is only relevant for the RS-232C (V.24) interface. The HW handshake signal is not relevant for the 3964(R) driver since it cannot be evaluated or influenced for data transmission with the 3964(R) protocol. Leave the default value ”0” (hardware handshake OFF) unchanged in parameter block 0.

Mode (parameter block 7)

A ”3” must be entered here for data transmission with the 3964(R) transmission protocol.

Character delay ZVZ (parameter block 7)

You can determine the maximum time which is permitted to elapse between two received characters (character delay). The CP 521 SI will then recognize as valid and transfer to the CPU in a message frame only those characters which have a delay within the defined limits. Select as short a character delay time as your application will allow, but make sure that the character delay is greater than the character transmission time. Approximately 3 1/2 times the character transmission time is recommended.

Acknowledgement delay (timeout) (parameter block 7)

During connection buildup and connection cleardown, the time which can elapse between control characters without errors arising.

Block waiting time (parameter block 7)

Is relevant for receiving message frames when block check characters (BCC) are used for sending and receiving.

Transmission with or without block check characters (parameter block 7)

When the **3964R transmission protocol** is used, a **block check character (BCC)** is transmitted at the end of each message block to ensure data integrity. The BCC is the even longitudinal parity (EXORing of all data bytes) of the frame transmitted or received. The BCC increases transmission reliability. A distinction is made between the 3964 driver and the 3964R driver, depending on whether a block check character is used or not.

Priority (parameter block 7)

If both devices issue a send job request at the same time, the device with lower priority withdraws its line bid. For data transmission with the 3964(R) communications protocol, you must set one device with higher priority and the other device with lower priority.

Buildup attempts (parameter block 7)

Number of attempts to build up a connection. After a total of six (default value) failed attempts, the procedure is aborted and the error in connection buildup is entered in the CBS coordination byte.

Number of send attempts (parameter block 7)

Number of attempts to transmit data. After n failed attempts to transmit data correctly, the procedure is aborted.

Cause for abort:
- Either parity error or BCC error.

Clock correction factor (parameter block 9)

You can configure a correction factor in this parameter block to enhance the accuracy of the real-time clock. You must always specify the correction value as a four-digit number (incl. sign).

The correction value is output in s/month. The month is defined as having 30 days (Clock Correction Factor (parameter block 9), Section 5.3.5).

Example: Assigning parameters for the 3964(R) driver mode on the memory submodule, using the PG 750 programmer

The CP 521 SI is plugged into slot 7 (starting address 120). To operate the module in 3964(R) mode, assign the parameters as follows:

- Parameters for the serial interface (parameter block 0)
  - 9600 bit/s (8)
  - Even parity (0)
  - No BUSY (0)
  - RS-232C (V.24) interface (1)
  - 7 data bits (0)
  - Handshake OFF (0)

- Parameters for the 3964 mode (parameter block 7)
  - 3964(R) driver (3)
  - Character delay (ZVZ): 230 ms (23)
  - Timeout (QVZ): 2010 ms (201)
  - Block waiting time (BWZ): 4010 ms (401)
  - Transmission of BCCs (0)
  - High priority (1)
  - 7 buildup attempts (AV) (7)
  - 7 send attempts (SV) (7)

- Parameters for clock correction (parameter block 9)
  - Correction factor: 1s/month (+001)

The end-of-text characters (parameter block 3) act as separators between the various parameters in the parameter block.

Recommended procedure:

1. Enter the parameter assignment data in DB1 on the programmer, together with the appropriate comments (KS=...).
2. Store the data in DB1 on diskette or hard disk.
3. Transfer DB1 to the memory submodule.

Table 8-4. Assigning the 3964(R) Communications Protocol Parameters on the Memory Submodule

Input from PG 750 to DB1	Explanation
0: KS ='Parameters of the ';	
12: S ='CP521SI ';	
24: KS ='Parameter block_0 ';	9600 baud, even parity, no Busy, RS232C
36: S =':0$800100$ ';	(V.24), 7 bits, no handshake 3964 driver,
48: KS ='Parameter block_7 ';	ZVZ = 230 ms, QVZ = 2010 ms, BWZ = 4010 mm,
60: S =':7$3$23$201$401$0$1$7$7$';	without BCC, high priority, 7 AV, 7 SV
72: KS ='Parameter block_9 ';	Clock correction
84: S =':9$+001$ ';	+ 1 second / 30 days
96:	

8.3.2 Assigning the CP 521 SI Parameters in the User Program

You can transfer data for **one** parameter block at a time to the CP 521 SI with the job $90XX_H$ "Transfer parameter assignment data".

Contents of the transfer memory when assigning CP 521 SI parameters in the user program

Write the CPU job "Transfer parameter assignment data" into the PIQ:

Request from the CPU to the CP: Transfer parameters

Address	Contents
Maddr +0	90_H
Maddr +1	Block number
Maddr + 2	Parameter
Maddr + 3	Parameter
Maddr + 4	Parameter
Maddr + 5	Parameter
Maddr + 6	Parameter
Maddr +7	Parameter

Code for "Transfer parameter assignment data"

Evaluate CP response to the "Transfer parameter assignment data" job in the PII:

CPU job accepted
Parameters passed

Address	Contents
Maddr +0	50_H
Maddr +1	00_H
Maddr + 2	irrelevant
Maddr + 3	irrelevant
Maddr + 4	irrelevant
Maddr + 5	irrelevant
Maddr + 6	90_H
Maddr +7	Block number

Acknow-
ledged
job

CP rejects CPU job:
illegal parameters/CP busy/
wrong job

Contents
40_H
00_H
irrelevant
irrelevant
irrelevant
irrelevant
90_H
Block number

Note

The CP responses 5000_H and 4000_H remain (i.e. are not deleted) until they are overwritten by a subsequent job (e.g. blank job 0000_H).

When evaluating data from the CP 521 SI (PII), allow for the fact that you receive the acknowledgement of a CPU job request (PIQ) only after two program scan cycles following submission of the job request. This has the following significance for parameter assignment with the user program:

- At least seven program scan cycles are required for parameter assignment in parameter blocks 0 and 7.
- If errors (error message 40_H) occur, correspondingly more program scan cycles are required.

Transferring parameter assignment data for parameter block 0

Table 8-5. Transfer Memory Assignments in the Case of the "Transfer Parameter Assignment Data for Parameter Block 0" Job Request

Byte	Parameter		Possible Values	Default Values on the CP 521 SI
0	Job request number "Transfer parameter assignment data"		90_H	-
1	Parameter block number ("0")		00_H	-
2	Baud rate	110 bit/s 200 bit/s 300 bit/s 600 bit/s 1200 bit/s 2400 bit/s 4800 bit/s 9600 bit/s	01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	08_H 9600 bit/s
3	Parity	even odd "mark" "space" no check	00_H 01_H 02_H 03_H 04_H	00_H even
4	BUSY signal*	no	00_H	00_H no
5	Interface	TTY RS-232C (V.24)	00_H 01_H	00_H TTY
6	Data format: 11-bit character 10-bit character	 7 data bits (with parity) 8 data bits (with parity) 8 data bits (without parity) 7 data bits (without parity) 7 data bits (with parity) 8 data bits (without parity)	 00_H 01_H 02_H 03_H 04_H 05_H	00_H 7 data bits (with parity)
7	HW handshake**	OFF	00_H	00_H OFF

* Busy signal irrelevant for 3964(R) driver

** Hardware handshake irrelevant for 3964(R) driver

Transferring parameter assignment data for parameter block 7

Table 8-6. Contents of Transfer Memory in the Case of Job Request ”9073$_H$”

Byte	Parameter	Possible Values	Default Values on the CP 521 SI
0	Job request number ”Transfer parameter assignment data”	90_H	-
1	Parameter block number (”7”) and ID digit for 3964(R) driver mode (”3”)	73_H	0: Memory submodule with message texts plugged in 1: Without memory submodule *
2+3	Character delay (ZVZ)	0001_H to $FFFF_H$ (·10 ms)	16_H (220 ms)
4+5	Timeout (QVZ)	0001_H to $FFFF_H$ (·10 ms)	$00C8_H$ (2 s)
6+7	Block waiting time (BWZ)	0001_H to $FFFF_H$ (·10 ms)	0190_H (4 s)

* Only if there is no battery backup; if a backup battery has been installed, the same driver mode is active after POWER-ON as before POWER-OFF

Table 8-7. Contents of Transfer Memory in the Case of the Follow-Up Job Request ”907A$_H$”

Byte	Parameter	Possible Values	Default Values on the CP 521 SI
0	Job request number ”Transfer parameter assignment data”	90_H	-
1	Parameter block number (”7”) and ID digit for 3964(R) driver mode (follow-up job request (”A”))	$7A_H$	0: Memory submodule with message texts plugged in 1: Without memory submodule *
2	Transm. without block check character Transmission with block check character	00_H 01_H	00_H
3	Priority Low High	00_H 01_H	00_H
4	Buildup attempts	00_H to FF_H	06_H
5	Number of send attempts	00_H to FF_H	06_H
6+7	Irrelevant	-	-

* Only if there is no battery backup; if a backup battery has been installed, the same driver mode is active after POWER-ON as before POWER-OFF

8.4 Data Transfer with the 3964(R) Transmission Protocol

Data transmission is initiated by a send or receive request. Every job request is acknowledged with specific responses or error messages. Depending on the job request issued, certain job request acknowledgements/job request response messages can only be scanned in a specific program scan cycle in the user program.

8.4.1 Sending Data with the 3964(R) Transmission Protocol

Send procedure from the point of view of the user program

The following subsection describes the principle of sending data in ”3964(R) driver” mode. A detailed description is given of the eight-byte blocks which

- are transferred as a job request to the transfer memory of the CP 521 SI (PIQ) and
- are read from the transfer memory of the CP 521 SI as an acknowledgement (PII).

Reminder:
Transferring eight-byte blocks of the transfer memory and reading acknowledgements from the transfer memory refer to communications between the CPU and the CP 521 SI.

Communications between the CPU, the CP 521 SI and the peripheral device can be controlled with the ”Send coordination byte” (CBS).

Schematic of an error-free send procedure

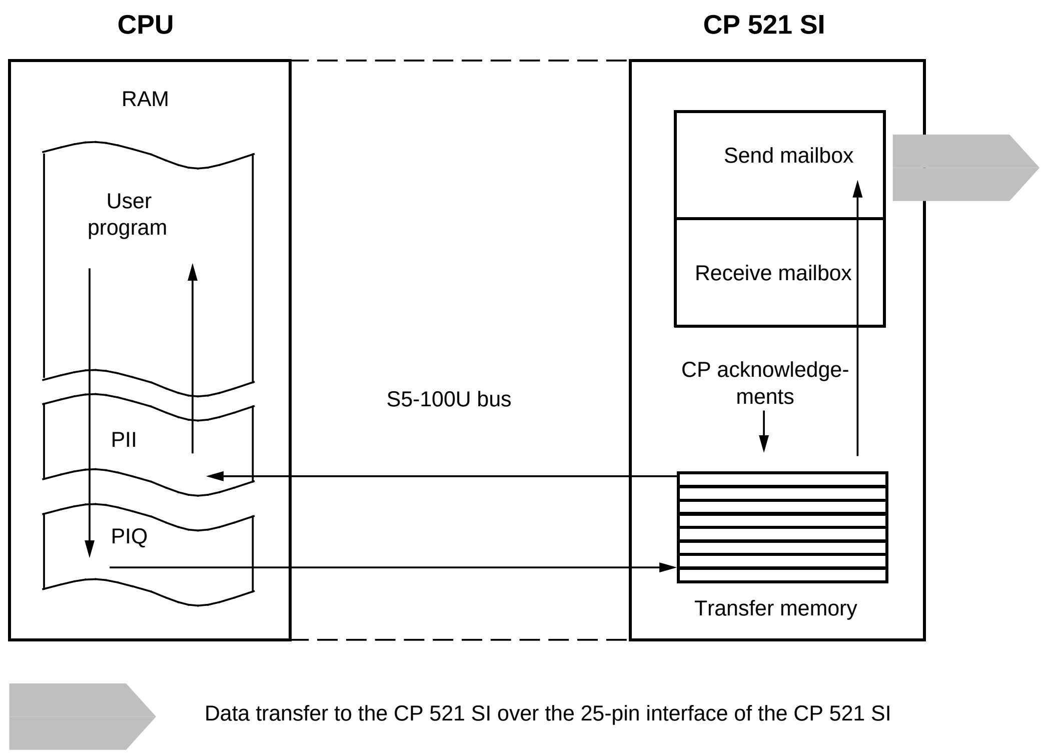


Figure 8-12. Sending Data over the CP 521 SI

A complete send sequence is represented on the following pages, showing the CPU job requests and the relevant CP responses. Job requests and responses are always eight bytes long. The send sequence is represented as follows:

- Left-hand side of the page: CPU requests and positive CP acknowledgements
- Right-hand side of page: Negative CP acknowledgements, which the CP enters if errors have arisen in data transmission.

Write send job request into the PIQ
Evaluate the CP acknowledgement for send job request
If the send job request has been accepted by the CP:
Write the first message block into the PIQ
Evaluate the CP acknowledgement for the first message block
Steps and are repeated for the next message blocks if the send job request has several frames.
The CPU sends the last message block to the CP.
Evaluate the CP acknowledgement for the last message block.
After the last message block has been written into the PIQ:
Evaluate terminating acknowledgement of the CP.

The following is a detailed description of each of these steps.

Write send job request to the PIQ
(to simplify presentation, the addresses are specified with ”Module address+X” (Maddr+X), X = 0, 1 to 7)

Request from CPU to CP:
Send data

Address	Contents
Maddr+0	$A0_H$
Maddr+1	01_H
Maddr+2	Send length of the message frame
Maddr+3	Send length of the message frame
Maddr+4	Irrelevant
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for send request

Specification of message frame length (max. 256 bytes)

Wait for one of the following CP acknowledgements:

Send request accepted
Send buffer now occupied

Address	Contents
Maddr+0	50_H
Maddr+1	01_H
Maddr+2	Irrelevant
Maddr+3	Irrelevant
Maddr+4	Message frames in receive mailbox
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

CP rejects send request

Contents
50_H
00_H
CBS *
Irrelevant
Message frames in receive mailbox
Irrelevant
Irrelevant
Irrelevant

* CBS: 0x/$0D_H$ parameter error
81_H send mailbox full, since previous send job request not yet completed (Table 8-10).

If the send job request has been accepted by the CP:
Write the first message block into the PIQ

CPU sends 1st message block to CP,
Data exchange starts/is continued.

Address	Contents
Maddr+0	$B0_H$
Maddr+1	01_H
Maddr+2	Data
Maddr+3	Data
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Code for "Message block transfer to CP" (Maddr+0)

Number of the message block (1 in this case) (Maddr+1)

Wait for acknowledgement for the previous (first) message block:

CP continues data exchange,
acknowledges previous message block

Address	Contents
Maddr+0	50_H
Maddr+1	01_H Send bit set
Maddr+2	Irrelevant
Maddr+3	No. of the message block sent
Maddr+4	Message frames in receive mailbox
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

CP aborts data exchange.
Data invalid or
wrong message block No.

Contents
50_H
00_H
CBS *
Irrelevant
Message frames in receive mailbox
Irrelevant
Irrelevant
Irrelevant

* 01_H: Errors in data transfer (wrong block number) (Table 8-10)

Steps and are repeated for the next message blocks if the send job request has several frames.

CPU sends last message block to CP

Address	Contents
Maddr+0	$B0_H$
Maddr+1	no_H
Maddr+2	Data
Maddr+3	Data
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Number of the last message block

Wait for acknowledgement for the last message block:

CP acknowledges last message block

Address	Contents
Maddr+0	50_H
Maddr+1	01_H Send bit set
Maddr+2	Irrelevant
Maddr+3	No. of the message block sent
Maddr+4	Message frames in receive mailbox
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

CP aborts data exchange

Contents
50_H
00_H
CBS *
Irrelevant
Message frames in receive mailbox
Irrelevant
Irrelevant
Irrelevant

* 01_H: Error in data transfer (wrong block number) (Table 8-10)

Wait for terminating acknowledgement after last message block has been written into the PIQ:

Terminating acknowledgement from CP to CPU

Address	Contents
Maddr+0	50_H
Maddr+1	00_H Send bit reset
Maddr+2	(Evaluate) CBS *
Maddr+3	Irrelevant
Maddr+4	Message frames in receive mailbox
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for ”Terminating acknowledgement”

* CBS: Table 8-10

This terminating message is only sent when the CP has transmitted the entire message frame to the peripheral device or when the number of buildup and repeat attempts has been reached.

8.4.2 Receiving Data with the 3964(R) Transmission Protocol

Receive procedure from the standpoint of the user program

The following subsection describes the receive principle in the ”3964(R) driver” mode. A detailed description is given of the eight-byte blocks which

- are transferred to the transfer memory of the CP 521 SI (PIQ) as a job request and
- are read from the transfer memory of the CP 521 SI (PII) as an acknowledgement.

Reminder:
Transferring eight-byte blocks into the transfer memory and reading acknowledgements from the transfer memory refer to communications between the CPU and the CP 521 SI. You can control communications between the CPU, the CP 521 SI and the peripheral device with the ”Coordination byte receive” (CBR).

Schematic of an (error-free) receive procedure

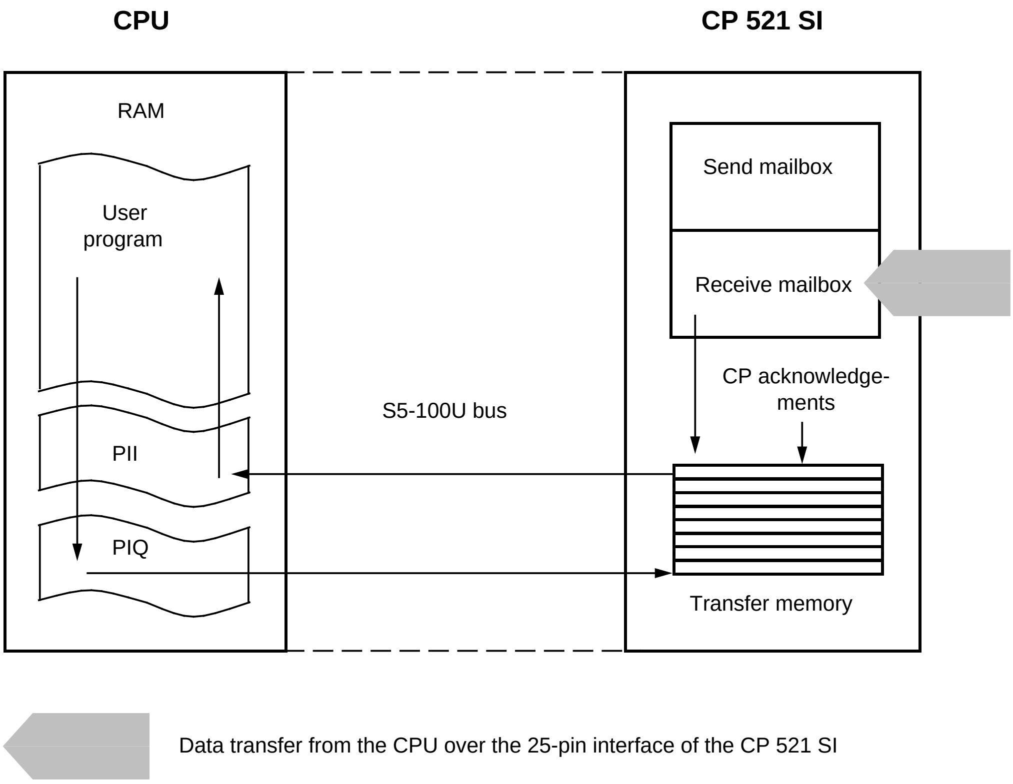


Figure 8-13. Receiving Data over the CP 521 SI

Write the receive request into the PIQ.
Evaluate the CP acknowledgement for the receive request.
If the CP acknowledgement was positive and the first message block has been read in, acknowledge the message block read in.
Read in the next message block and evaluate CP acknowledgement for this message block.
Repeat steps and until the last message block has been read in.
Evaluate terminating acknowledgement of the CP 521 SI.

Write the receive request into the PIQ:
(to simplify presentation, the addresses are specified with ”Module address+X” (Maddr+X), X = 0, 1 to 7).

Request from CPU to CP:
Receive data

Address	Contents
Maddr+0	$A0_H$
Maddr+1	80_H
Maddr+2	Irrelevant
Maddr+3	Irrelevant
Maddr+4	Irrelevant
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for receive request

Wait for one of the following CP acknowledgements:

Receive request accepted,
CP transfers 1st message block to CPU

Address	Contents
Maddr+0	60_H
Maddr+1	01_H
Maddr+2	Length of the receive message frame
Maddr+3	Length of the receive message frame
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Message block number (1 in this case)

CP rejects receive request
Message frame does not exist,
has errors or
message frame length 0

Contents
50_H
00_H
CBR *****
Irrelevant
Message frames in receive mailbox
Irrelevant
Irrelevant
Irrelevant

* 02_H: No message frame available
(Table 8-11)

If the receive job request has been accepted:

Acknowledge the first message block

Address	Contents
Maddr+0	$C0_H$
Maddr+1	01_H
Maddr+2	Irrelevant
Maddr+3	Irrelevant
Maddr+4	Irrelevant
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for ”Acknowledge message block”

Number of the message block (1 in this case)

CP transfers the next message block:

Address	Contents
Maddr+0	60_H
Maddr+1	02_H
Maddr+2	Data
Maddr+3	Data
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Message block number (2 in this case)

CP abort data exchange

Contents
50_H
00_H
CBR *****
irrelevant
Message frames in receive mailbox
irrelevant
irrelevant
irrelevant

***** 81_H: Error in data transfer (wrong block number) (Table 8-11)

Steps and are repeated if the receive job request contains several blocks.

After the last block has been read:

Wait for the following CP acknowledgement:

Address	Contents
Maddr+0	50_H
Maddr+1	00_H
Maddr+2	(Evaluate) CBR*
Maddr+3	00_H
Maddr+4	Message frames in receive mailbox
Maddr+5	Number of valid data bits of last block
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for "Terminating acknowledgement"

***** CBR: Table 8-11

Additional CPU job "Delete receive mailbox contents

It is often relevant to evaluate a current message frame. With the job "Delete receive mailbox contents" ($D0_H$) you have now the possibility to delete the receive mailbox before receiving the relevant message frame.

Note

When executing the job "Delete receive mailbox contents", the serial interface of the CP 521 SI is briefly disabled. If the peripheral device connected continues to send data, the first message frame received by the CP 521 SI can be faulty or incomplete.

The interface to the CPU is also briefly occupied by the CP checkback signal "$0F_H$". Therefore **always** evaluate the CP acknowledgement to the job "Delete receive mailbox contents" before you send further CPU job requests to the CP 521 SI.
The CP 521 SI acknowledges the job "Delete receive mailbox contents" like all parameterization jobs with the terminating message "5000_H" **and** specification of the job acknowledged (here: "$D000_H$") in bytes 6 and 7 of the CP response.

8.5 Error Flags in the CBS and CBR of the CP Acknowledgements

After a CPU send job request, the CP sends a response to the CPU. The response consists of eight bytes. The eight bytes include the CBS and additional information on the status of the CP. After a CPU receive job request, the response contains the CBR and further information on the status of the CP.

The following tables tell you what information the eight-byte response messages contain.

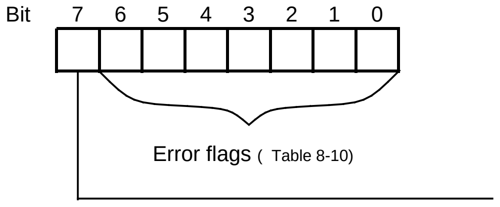
Table 8-8. Status Message After the Send Job Request "A001$_H$"

Byte	Meaning	Possible Values
0	Status byte	50$_H$
1	Send job request completed	00$_H$
2	Coordination byte "Send" (CBS)	Table (CBS)
3	Irrelevant	
4	Number of message frames in receive mailbox	00$_H$ to 63$_H$
5 to 7	Irrelevant	

Table 8-9. Status Message After the Receive Job Request "A080$_H$"

Byte	Meaning	Possible Values
0	Status byte	50$_H$
1	Receive job request completed	00$_H$
2	Coordination byte "Send" (CBR)	Table (CBR)
3	Irrelevant	
4	Number of message frames in receive mailbox	00$_H$ to 63$_H$
5 to 7	Irrelevant	

Responses in the coordination byte Send (CBS):



1: Send mailbox occupied since preceding send job active **or** job accepted and send mailbox now occupied
0: Parameter error (e.g. wrong send length) **or** send terminated and send mailbox free

Table 8-10. Error Flags in the Coordination Byte ”Send” (CBS) in the Case of ”3964(R)”

Error Flag	Meaning	Response
00_H	No errors	-
09_H	Negative acknowledgement from receiver to connection **cleardown** attempt	Receive data invalid
$0B_H$	Negative acknowledgement from receiver to connection **buildup** attempt	Receive data invalid
$0D_H$	Parameter assignment error	No data is transmitted
$0F_H$	Transmission aborted by receiver	Receive data invalid
15_H	Timeout (QVZ) during connection **buildup**	No data is transmitted
17_H	Timeout (QVZ) during connection **cleardown**	Receive data invalid
19_H	Initiation conflict; both link partners have high priority	No data is transmitted
$1B_H$	Break	Transmission is aborted
$1D_H$	Initiation conflict; both link partners have low priority	No data is transmitted

The CP acknowledgement "5000_H" is the terminating acknowledgement for a "successful" or "abortive" data transfer between the CPU and the CP 521 SI. To distinguish between "abortive" and "successful", you must always evaluate the following CBS flags in the user program:

CBS= 00_H No error when sending
CBS= 81_H Send mailbox occupied because preceding send job not yet terminated
CBS= 01_H Error when transferring data between the CPU and the CP (wrong block number)

Responses in the coordination byte Receive (CBR):

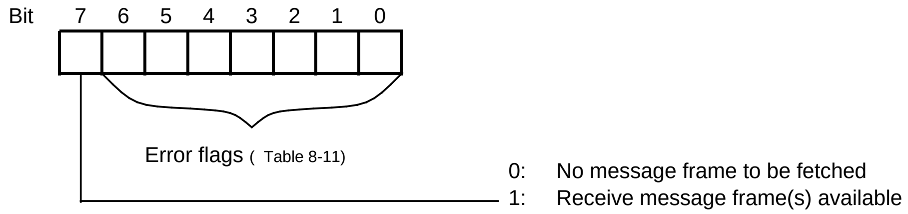


Table 8-11. Error Flags in the Coordination Byte ”Receive” (CBR) in the Case of ”3964(R)”

Error Flag	Meaning	Priority	Response
00_H	No errors	-	-
03_H	Parity error	5	Data rejected
05_H	Message frame length 0	6 (low)	-
07_H	Receive buffer full	2	Data rejected
09_H	Too many message frames received	2	Buffer full; subsequent message frames are rejected
$0B_H$	Message frame too long (256 bytes)	0 (highest)	Data rejected
$0D_H$	DLE not doubled, or no ETX after DLE	3	Data rejected
11_H	STX error; handshaking ON but no STX at start	3	Data rejected
13_H	Character delay error	4	Data rejected
15_H	Block waiting time error	2	Data rejected
17_H	Checksum error	5	Data rejected
$1B_H$	Break	1	Data rejected

The CP acknowledgement "5000_H" is the terminating acknowledgement for a "successful" or "abortive" data transfer between the CPU and the CP 521 SI. To distinguish between "abortive" and "successful", you must always evaluate the following CBS flags in the user program:

CBR = 00_H No error when receiving; no more message frames in the receive mailbox
CBR = 80_H No error when receiving; at least one receive message frame still in mailbox
CBR = 02_H No receive message frame in mailbox
CBR = 81_H Error when transferring data between the CPU and the CP (wrong block number)

8.6 STEP 5 Programs for Data Transmission with the 3964(R) Driver

This section contains a complete STEP 5 program for data transmission with the 3964(R) driver.

- DB21: Send data (source DB)
- DB22: Receive data (destination DB)

- FB3: Parameter assignment FB
- FB20: Pulse generation for FB21
- FB21: Switch over between sending and receiving
- FB200: Send FB (user-configurable; can only be used for CPU 103)
- FB201: Receive FB (user-configurable; can only be used for CPU 103)

- OB1:
  - Invokes the parameter assignment FB
    If the CP 521 SI has been assigned its parameters, this FB is no longer invoked
  - Invokes FB21, which, in turn, invokes either the send FB (FB200) or the receive FB (FB201).
- OB21/22: Defaults for the parameter assignment FB

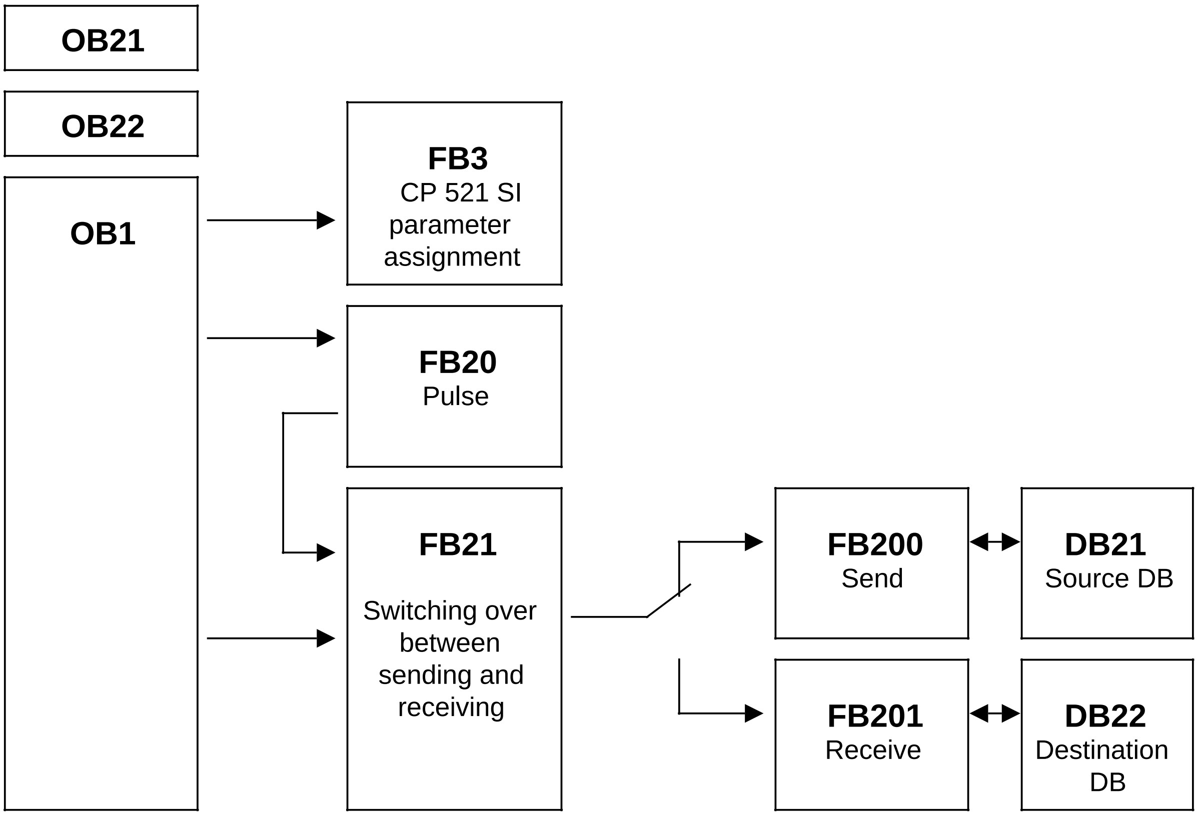


Figure 8-14. Program Structure for Sending and Receiving with 3964(R)

The most important flags (F) used in the program are listed and explained in the following:

Flags	Meaning
F 100.0	Transfer parameters for parameter block 0
F 100.1	Transfer parameters for parameter block 73
F 100.2	Transfer parameters for parameter block 7A (successor job request)
F 100.3	Parameter flag
F 100.7	Parameter assignment completed
F 101.0	Pulse flag
F 80.0	Send job request initiated
F 80.1	Error in last send job
F 80.2	Send FB active
F 80.4	Receive job request initiated
F 80.5	Error in last receive job
F 80.6	Receive FB active

In FB3 and FB200/FB201, flags 100 to 108 and flags 234 to 254 have special programming functions. These flags must therefore not be used in the rest of the program.

```
OB 1                                D:3964@@ST.S5D            LEN=20

        SEGMENT  1     0000
        0000      :AN  F  100.7                    As long as flag 100.7 has not
        0001      :JC  FB   3                      yet been set, jump to
        0002 NAME :CP-PARAM
        0003 BGAD :    KF +64                      parameter assignment FB
        0004 OK   :    F  100.7
        0005      :AN  F  100.7
        0006      :BEC
        0007      :
        0008      :JU  FB  20                      Pulse generator FB
        0009 NAME :PULSE
        000A      :
        000B      :JU  FB  21                      Receive and send job FB
        000C NAME :DISTRIB
        000D      :
        000E      :BE

OB 21                               D:3964@@ST.S5D            LEN=11

        SEGMENT  1     0000
        0000      :L   KH 0000                     Defaults for parameter assignment FB
        0002      :T   FW 100
        0003      :
        0004      :
        0005      :BE

OB 22                               D:3964@@ST.S5D            LEN=7

        SEGMENT  1     0000
        0000      :JU  OB  21
        0001      :BE
```

```
FB 3                          D:3964@@ST.S5D                      LEN=167

SEGMENT  1     0000
NAME :CP-PARAM                               3964(R) mode
DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
DES  :OK        I/Q/D/B/T/C: Q  BI/BY/W/D: BI

000B      :LW  =BGAD                         This FB uses flag words
000C      :T   FW 102                        FW 10 to 108 to assign the
000D      :I        2                        CP 521 SI
000E      :T   FW 104                        parameters.
000F      :I        2                        FW 102 and 108 are addresses
0010      :T   FW 106                        of the CP.
0011      :I        2                        FW 10 is used for executing the FB.
0012      :T   FW 108
0013      :
0014      :DO  FW 102                        Is CP still in
0015      :L   IW   0                        restart routine?
0016      :L   KH 0F00
0018      :AW
0019      :L   KH 0F00
001B      :!=F
001C      :BEC
001D      :***


SEGMENT  2     001E
001E      :A   F  100.0                      Assign parameters to first block
001F      :JC  =BL-E
0020      :L   KH 9000                       Job block 0
0022      :DO  FW 102
0023      :T   QW   0
0024      :L   KH 0800                       Baud rate    Parity
0026      :DO  FW 104
0027      :T   QW   0
0028      :L   KH 0000                       BUSY         Interface
002A      :DO  FW 106
002B      :T   QW   0
002C      :L   KH 0100                       Data format  Handshake
002E      :DO  FW 108
002F      :T   QW   0
0030      :
0031      :DO  FW 102                        Wait for positive
0032      :L   IW   0                        acknowledgement 5XXXH
0033      :L   KH 5000
0035      :AW
0036      :L   KH 5000
0038      :><F
0039      :BEC
003A      :DO  FW 108                        Check whether acknowledgement
003B      :L   IW   0                        belongs to job, otherwise wait
003C      :L   KH 9000                       for right acknowledgement
003E      :><F
003F      :BEC
0040 BL-E :AN  F  100.0                      Set step flags
0041      :S   F  100.0
0042      :***
```

```
SEGMENT  3     0043
0043      :A   F  100.1                 Assign parameters to second block
0044      :JC  =BL-E
0045      :L   KH 9073                  Block 7    3964(R) mode
0047      :DO  FW 102
0048      :T   QW   0
0049      :L   KF 0022                  Character delay
004B      :DO  FW 104                   220 ms
004C      :T   QW   0
004D      :L   KF 0200                  Timeout
004F      :DO  FW 106                   2 s
0050      :T   QW   0
0051      :L   KF 0400                  Block waiting time
0053      :DO  FW 108                   4 s
0054      :T   QW   0
0055      :
0056      :DO  FW 102                   Wait for positive
0057      :L   IW   0                   acknowledgement 5XXXH
0058      :L   KH 5000
005A      :AW
005B      :L   KH 5000
005D      :><F
005E      :BEC
005F      :DO  FW 108                   Check whether acknowledgement
0060      :L   IW   0                   belongs to job, otherwise wait for
0061      :L   KH 9073                  the right acknowledgement
0063      :><F
0064      :BEC
0065 BL-E :AN  F  100.1                 Set step flag
0066      :S   F  100.1
0067      :
0068      :***

SEGMENT  4     0069
0069      :A   F  100.2                 Assign parameters to third block
006A      :JC  =BL-E
006B      :L   KH 907A                  Block 7 (A: follow-up job request)
006D      :DO  FW 102
006E      :T   QW   0
006F      :L   KH 0000                  BCC        Priority
0071      :DO  FW 104
0072      :T   QW   0
0073      :L   KH 0606                  Buildup    send attempts
0075      :DO  FW 106
0076      :T   QW   0
0077      :L   KH 0000                  irrelevant irrelevant
0079      :DO  FW 108
007A      :T   QW   0
007B      :
007C      :DO  FW 102                   Wait for positive
007D      :L   IW   0                   acknowledgement 5XXXH
007E      :L   KH 5000
0080      :AW
0081      :L   KH 5000
0083      :><F
0084      :BEC
0085      :DO  FW 108                   Check whether acknowledgement
0086      :L   IW   0                   belongs to job, otherwise wait
0087      :L   KH 907A                  for right acknowledgement
0089      :><F
008A      :BEC
008B BL-E :AN  F  100.2                 Set step flag
008C      :S   F  100.2
008D      :***
```

```
SEGMENT  5     008E
008E      :A   F  100.3                        Last block deletes acknowledgement
008F      :JC  =BL-E                           5000H.
0090      :L   KH 0000
0092      :DO  FW 102
0093      :T   QW   0
0094      :
0095      :DO  FW 102
0096      :L   IW   0
0097      :L   KH 5000
0099      :AW
009A      :L   KH 0000
009C      :><F
009D      :BEC
009E BL-E :AN  F  100.3
009F      :S   F  100.3
00A0      :S   =OK                             Set parameter assignment flag
00A1      :BE
```

```
FB 20                              D:3964@@ST.S5D                    LEN=19

        SEGMENT  1     0000
        NAME :PULSE

        0005      :AN  F  101.1
        0006      :L   KT 030.1
        0008      :SP  T  101
        0009      :AN  T  101
        000A      :AN  F  101.1
        000B      :=   F  101.1
        000C      :S   F  101.0                        Pulse flag is set
        000D      :BE                                  after 3 seconds.
```

```
FB 21                                B:3964@@ST.S5D                   LEN=43

        SEGMENT  1     0000
        NAME :DISTRIB

        0005      :AN  F   80.2                        Send not active
        0006      :JC  =M001
        0007      :
        0008      :JU  FB 200                          Process send FB
        0009 NAME :SEND
        000A BGAD :    KF +64
        000B Q-DB :    DB  21
        000C QANF :    KF +0
        000D QLAE :    KF +12
        000E KBS  :    FY  90
        000F      :R   F  101.0
        0010      :BEU
        0011      :
        0012 M001 :AN  F   80.6
        0013      :JC  =M002
        0014      :
        0015      :JU  FB 201
        0016 NAME :RECEIVE
        0017 BGAD :    KF +64                          Receive not active
        0018 Z-DB :    DB  22
        0019 ZANF :    KF +0
        001A KBE  :    FY  91
        001B      :                                    Process receive FB
        001C      :BEU
        001D      :
        001E M002 :A   F  101.0
        001F      :S   F   80.2
        0020      :R   F  101.0
        0021      :A   F   80.2
        0022      :BEC
        0023      :AN  F   80.6
        0024      :S   F   80.6                        Set pulse flag
        0025      :BE                                  Set send flag
```

```
FB 200                               D:3964@@ST.S5D                LEN=222

        SEGMENT  1     0000          SEND
        NAME :SEND
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Q-DB      I/Q/D/B/T/C: B
        DES  :QANF      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :QLAE      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :KBS       I/Q/D/B/T/C: Q  BI/BY/W/D: BY

        0014      :LW  =BGAD                        Load module address
        0015      :T   FW 234
        0016      :
        0017      :DO  FW 234                       Is CP busy?
        0018      :L   IW   0
        0019      :L   KH 0F00
        001B      :AW
        001C      :L   KH 0F00
        001E      :!=F
        001F      :BEC                              If yes, end block conditionally
        0020      :***


        SEGMENT  2     0021
        0021      :A   F   80.0                     Flag for job initiated
        0022      :JC  =M001                        If this flag is set, do not
        0023      :                                 initiate new job
        0024      :
        0025      :DO  FW 234                       Check whether job can be
        0026      :L   IW   0                       initiated
        0027      :L   KH C000
        0029      :AW
        002A      :L   KH 0000
        002C      :><F
        002D      :BEC
        002E      :
        002F      :L   KF +0                        Reset block counter
        0031      :T   FW 252
        0032      :LW  =QLAE                        Buffer length
        0033      :T   FW 254
        0034 M003 :L   FW 252
        0035      :NOP 1                            Increment block counter
        0037      :T   FW 252
        0038      :L   FW 254                       Keep subtracting the number of
        0039      :NOP 1                            bytes of a block from the
        003B      :T   FW 254                       length until zero is reached
        003C      :L   KF +0
        003E      :<=F
        003F      :JC  =M002                        Number of blocks is fixed
        0040      :JU  =M003                        Final number of blocks not
        0041      :                                 yet reached
        0042 M002 :
        0043      :L   FW 234                       Set address counter
        0044      :L   KF +2
        0046      :+F
        0047      :T   FW 236                       Module address + 2
        0048      :L   KF +2
        004A      :+F
        004B      :T   FW 238                       Module address + 4
        004C      :L   KF +2
        004E      :+F
        004F      :T   FW 240                       Module address + 6
        0050      :
```

```
0051      :L   KH A001                              Initiate send job
0053      :DO  FW 234
0054      :T   QW   0
0055      :LW  =QLAE
0056      :DO  FW 236                               Store block length in word 1
0057      :T   QW   0
0058      :
0059      :L   KH B001                              Identifier for block
005B      :T   FW 242                               acknowledgement (first block)
005C      :L   KF +0                                Store comparison value
005E      :T   FW 244                               (block 1)
005F      :
0060      :LW  =QANF                                DB
0061      :T   FW 246                               Starting address
0062      :L   KF +1
0064      :+F
0065      :T   FW 248                               Starting address+ 1
0066      :L   KF +1
0068      :+F
0069      :T   FW 250                               Starting address+ 2
006A      :
006B      :AN  F   80.0                             Set flag for send job
006C      :S   F   80.0                             initiated
006D      :R   F   80.1                             Reset error flag
006E      :BEU
006F M001 :***


SEGMENT  3     0070
0070      :DO  FW 234                               Load input word 0 and check
0071      :L   IW   0                               whether job has been
0072      :L   KH F00F                              accepted
0074      :AW
0075      :L   KH 5001
0077      :><F
0078      :JC  =M001
0079      :
007A      :DO  FW 236                               Check whether
007B      :L   IW   0                               the right block
007C      :L   KH 00FF                              can be received
007E      :AW
007F      :L   FW 244
0080      :><F
0081      :BEC
0082      :
0083      :L   FW 242                               Check whether a further block
0084      :L   KH 00FF                              can be sent
0086      :AW
0087      :L   FW 252
0088      :>F
0089      :BEC
008A      :
008B      :DO  =Q-DB                                Open source DB
008C      :L   FW 242                               Announce block and
008D      :DO  FW 234                               transfer it
008E      :T   QW   0
008F      :
0090      :DO  FW 246                               Word 0
0091      :L   DW   0
0092      :DO  FW 236
0093      :T   QW   0
0094      :
```

```
0095      :DO  FW 248                         Word 1
0096      :L   DW   0
0097      :DO  FW 238
0098      :T   QW   0
0099      :
009A      :DO  FW 250                         Word 2
009B      :L   DW   0
009C      :DO  FW 240
009D      :T   QW   0
009E      :
009F      :L   FW 242                         Update block acknowledgement
00A0      :L   KF +1
00A2      :+F
00A3      :T   FW 242
00A4      :
00A5      :L   FW 244                         Update block number
00A6      :L   KF +1
00A8      :+F
00A9      :T   FW 244
00AA      :                                   Increment data word pointer
00AB      :L   FW 246                         Word 0
00AC      :L   KF +3
00AE      :+F
00AF      :T   FW 246
00B0      :
00B1      :L   FW 248                         Word 1
00B2      :L   KF +3
00B4      :+F
00B5      :T   FW 248
00B6      :
00B7      :L   FW 250                         Word 2
00B8      :L   KF +3
00BA      :+F
00BB      :T   FW 250
00BC      :BEU
00BD M001 :***


SEGMENT  4     00BE
00BE      :DO  FW 234                         Check whether terminating
00BF      :L   IW   0                         acknowledgement present
00C0      :L   KH F00F
00C2      :AW
00C3      :L   KH 5000
00C5      :><F
00C6      :JC  =M001
00C7      :DO  FW 236                         Read in input byte 3 and transfer
00C8      :L   IB   0                         it to the CBS
00C9      :T   =KBS
00CA      :DO  FW 236                         Send without error?
00CB      :L   IW   0
00CC      :L   KH 0100
00CE      :AW
00CF      :JZ  =M002
00D0      :AN  F   80.1                       No; error terminated with
00D1      :S   F   80.1                       error.
00D2 M002 :R   F   80.2                       Yes; enable for new send or
00D3      :R   F   80.0                       receive job
00D4      :L   KH 0000
00D6      :DO  FW 234
00D7      :T   QW   0
00D8 M001 :BE
```

```
FB 201                               D:3964@@ST.S5D              LEN=175

        SEGMENT  1     0000          RECEIVE
        NAME :RECEIVE
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Z-DB      I/Q/D/B/T/C: B
        DES  :ZANF      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :KBE       I/Q/D/B/T/C: Q  BI/BY/W/D: BY

        0011      :LW  =BGAD                        Load and store
        0012      :T   FW 234                       module address
        0013      :
        0014      :DO  FW 234                       Is CP busy?
        0015      :L   IW   0
        0016      :L   KH 0F00
        0018      :AW
        0019      :L   KH 0F00
        001B      :!=F
        001C      :BEC                              Yes, end block conditionally
        001D      :***


        SEGMENT  2     001E
        001E      :A   F   80.4                     Flag for job already
        001F      :JC  =M001                        initiated
        0020      :
        0021      :DO  FW 234                       Check whether a job
        0022      :L   IW   0                       can be initiated
        0023      :L   KH C000
        0025      :AW
        0026      :L   KH 0000
        0028      :><F
        0029      :BEC
        002A      :
        002B      :L   FW 234                       Set address pointer
        002C      :L   KF +2
        002E      :+F
        002F      :T   FW 236                       Base address + 2 (Word 1)
        0030      :L   KF +2
        0032      :+F
        0033      :T   FW 238                       Base address + 4 (Word 2)
        0034      :L   KF +2
        0036      :+F
        0037      :T   FW 240                       Base address + 6 (Word 3)
        0038      :
        0039      :L   KH A080                      Initiate receive job
        003B      :DO  FW 234
        003C      :T   QW   0
        003D      :L   KH 6001                      Read acknowledgements
        003F      :T   FW 242                       into flag words
        0040      :L   KH C001
        0042      :T   FW 244
        0043      :LW  =ZANF                        Set pointer to destination data
        0044      :T   FW 246                       Destination starting address
        0045      :L   KF +1
        0047      :+F
        0048      :T   FW 248                       Starting address + 1
        0049      :L   KF +1
        004B      :+F
        004C      :T   FW 250                       Starting address + 2
        004D      :
```

```
004E      :S   F   80.4                  Set flag for receive job
004F      :R   F   80.5                  initiated and reset
0050      :BEU                           error flag
0051 M001 :***


SEGMENT  3     0052
0052      :DO  FW 234                    Load input word 0 and
0053      :L   IW   0                    check whether receive
0054      :L   KH F0FF                   block present
0056      :AW
0057      :L   FW 242
0058      :><F
0059      :JC  =M001
005A      :DO  =Z-DB                     Open destination DB
005B      :DO  FW 236
005C      :L   IW   0
005D      :DO  FW 246                    Word for
005E      :T   DW   0                    data word pointer 0
005F      :
0060      :DO  FW 238
0061      :L   IW   0
0062      :DO  FW 248                    Word for
0063      :T   DW   0                    data word pointer 1
0064      :
0065      :DO  FW 240
0066      :L   IW   0
0067      :DO  FW 250                    Word for
0068      :T   DW   0                    data word pointer 2
0069      :
006A      :L   FW 244                    Acknowledge data transfer
006B      :DO  FW 234
006C      :T   QW   0
006D      :
006E      :L   FW 242                    Update block message and
006F      :L   KF +1
0071      :+F
0072      :T   FW 242
0073      :
0074      :L   FW 244                    block acknowledgement
0075      :L   KF +1
0077      :+F
0078      :T   FW 244
0079      :
007A      :L   FW 246                    Increment data word pointer word 0
007B      :L   KF +3
007D      :+F
007E      :T   FW 246
007F      :
0080      :L   FW 248                    Increment data word pointer word 1
0081      :L   KF +3
0083      :+F
0084      :T   FW 248
0085      :
0086      :L   FW 250                    Increment data word pointer word 2
0087      :L   KF +3
0089      :+F
008A      :T   FW 250
008B      :BEU
008C M001 :***
```

```
SEGMENT  4     008D
008D      :DO  FW 234                     Check whether terminating
008E      :L   IW   0                     acknowledgement present
008F      :L   KH F00F
0091      :AW
0092      :L   KH 5000
0094      :><F
0095      :JC  =M001
0096      :DO  FW 236                     Read in input byte 3 and
0097      :L   IB   0                     transfer it to the CBR
0098      :T   =KBE
0099      :DO  FW 236
009A      :L   IW   0
009B      :L   KH 0100
009D      :AW
009E      :JZ  =M002
009F      :AN  F   80.5                   No; receive terminated
00A0      :S   F   80.5                   with error
00A1 M002 :R   F   80.6                   Yes; enable for new send or
00A2      :R   F   80.4                   receive job
00A3      :
00A4      :L   KH 0000
00A6      :DO  FW 234
00A7      :T   QW   0
00A8 M001 :
00A9      :BE

DB21    A:3964@@ST.S5D                    LEN=272 /4

     0:     KH = 004E;
     1:     KH = 0000;
     2:     KH = 0000;
     3:     KH = 0000;
     4:     KH = 0000;
     5:     KH = 0000;
     6:     KH = 0000;
     7:     KH = 0000;
     8:     KH = 0000;

DB22    A:3964@@ST.S5D                    LEN=147 /4

     0:     KH = 0020;
     1:     KH = 0006;
     2:     KH = 1234;
     3:     KH = 1212;
     4:     KH = 1212;
     5:     KH = 1211;
     6:     KH = 1111;
     7:     KH = 1111;
     8:     KH = 1111;
     9:     KH = 1232;
    10:     KH = 2222;
    11:     KH = 0D0D;
    12:     KH = 2222;
```

Note

All flags used in the example programs in FB 200 and FB 201 are greater than 200. If you use these flags as scratchpad flags in other standard FBs, they must be smaller than 200.

9 SINEC L1 Driver

9.1 Connecting the CP 521 SI to the BT 777 Bus Terminal . . . . 9 - 1

9.2 Assigning CP 521 SI Parameters for SINEC L1 Operation . . . . 9 - 2
9.2.1 Passing Parameters for SINEC L1 Operation to the Module Via the User Program . . . . 9 - 3
9.2.2 Providing Parameters for SINEC L1 Operation Over the Memory Submodule of the Module . . . . 9 - 4

9.3 Send Procedure from the Point of View of the Control Program . . 9 - 5

9.4 Receive Procedure from the Point of View of the Control Program 9 - 9

9.5 Point-to-Point Connection . . . . 9 - 13
9.5.1 Connecting the SINEC L1 Slave Direct to the Interface of the CP 521 SI . . . . 9 - 13
9.5.2 Assigning Parameters to the CP 521 SI as ”Point-to-Point Master” . 9 - 14
9.5.3 Programming Data Transfer between the CP 521 SI and the Slave . . . . 9 - 14

9.6 STEP 5 Program for Data Transmission with the SINEC L1 Driver . . . 9 - 15

Figures

9-1. Terminal Diagram CP 521 SI - BT 777 Bus Terminal BT 777 . . . . 9 - 2
9-2. Terminal Diagram for a CP 521 SI - CP 521 SI Connection (RS-232C/V.24) . . 9 - 3
9-3. Meaning of the Parameters in DB1 in the Case of SINEC L1 Parameter Assignment . . . . 9 - 4
9-4. Sending Data over the CP 521 SI . . . . 9 - 5
9-5. Structure of the Send Coordination Byte (CBS) . . . . 9 - 8
9-6. Receiving Data over the CP 521 SI . . . . 9 - 9
9-7. Structure of the Coordination Byte Receive (CBR) . . . . 9 - 12
9-8. Terminal Diagram CP 521 SI (TTY Active) - SINEC L1 Slave . . . . 9 - 13
9-9. Structure of the STEP 5 Program for Data Transmission with the SINEC L1 Driver . . . . 9 - 15

Tables

9-1. Pin Assignments of the 25-pin Subminiature D Socket Connector on the CP 521 SI . . . . 9 - 1

9 SINEC L1 Driver

SINEC L1 is a communications system for linking SIMATIC S5 programmable controllers of the U range; it operates on the master-slave principle. The ”master” PLC must be connected to the SINEC L1 local area network (LAN) over a CP 530. You can connect an S5-100U as a slave to the SINEC L1 LAN over the CP 521 SI interface. This is described in the next section.

See the SINEC L1 Manual for information on the functional principle of SINEC L1.

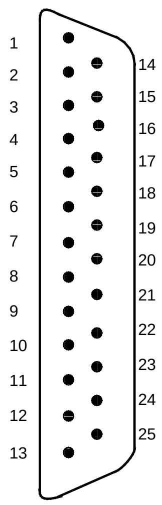
9.1 Connecting the CP 521 SI to the BT 777 Bus Terminal

One master and up to 30 slaves can be connected to the SINEC L1 LAN.

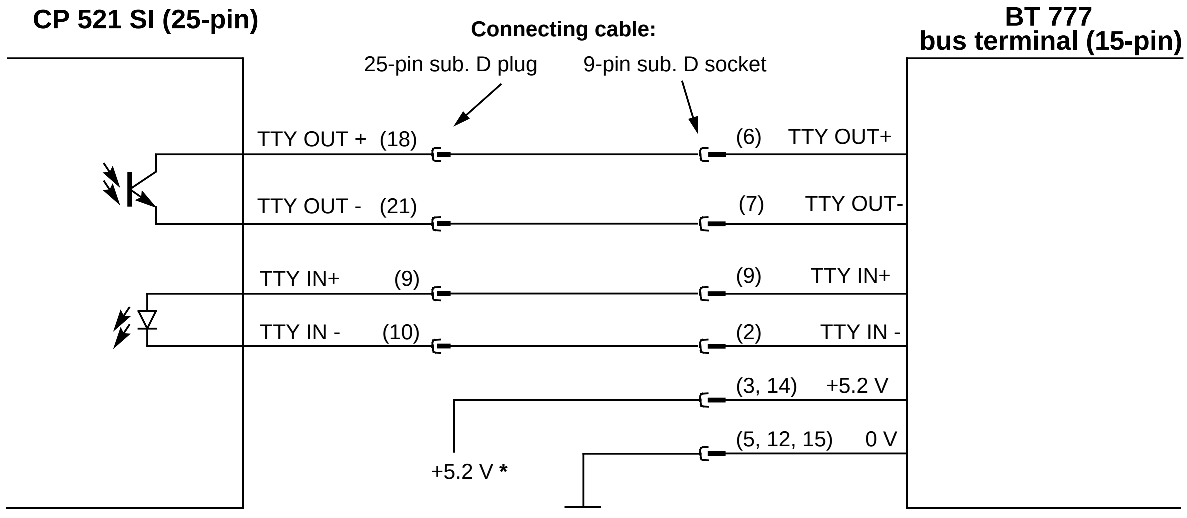
A BT 777 is required as a signal converter for every station or node - master or slave. Data is then transmitted over a four-wire shielded cable connecting the individual bus terminals to each other.

The CP 521 SI   BT 777 bus terminal connections are described in the following.

Table 9-1. Pin Assignments of the 25-pin Subminiature D Socket Connector on the CP 521 SI

View	Pin No.	Signal Name	Meaning
	1	-	Disabled
	2	TxD	Send data (V.24)
	3	RxD	Receive data (V.24)
	4	RTS	Request to send (V.24)
	5	CTS	Clear to send (V.24)
	6	DSR	Data set ready (V.24)
	7	GND	Signal ground (RS-232C (V.24))
	8	-	Disabled
	9	TTY IN+	TTY receive line+
	10	TTY IN -	TTY receive line -
	11	-	Disabled
	12	-	Disabled
	13	P24	+24 V for active TTY
	14	-	Disabled
	15	-	Disabled
	16	-	Disabled
	17	20 mA	Current source TTY *****
	18	TTY OUT+	TTY send line +
	19	20 mA	Current source TTY *****
	20	DTR	Terminal ready
	21	TTY OUT -	TTY send line -
	22	-	Disabled
	23	-	Disabled
	24	-	Disabled
	25	-	Disabled

***** If +24 V to GND (pin 7) on pin 13



***** Alternatively, you can supply the bus terminal with 5 V at terminals C and D (SINEC L1 Manual).

Figure 9-1. Terminal Diagram CP 521 SI - BT 777 Bus Terminal BT 777

9.2 Assigning CP 521 SI Parameters for SINEC L1 Operation

There are two methods of assigning parameters to the CP 521 for SINEC L1 operation:

- Transfer the parameters to the module over the PIQ in the control program
  or
- Transfer the parameters to a memory submodule for the CP 521 SI. The CP 521 SI then accepts these parameters on restart.

The following section describes the type, meaning and value ranges of the parameters for SINEC L1 operation. A sample program for assigning CP parameters on CPU restart is appended to this description.

The subsequent section shows you how to store the SINEC L1 parameters on a memory submodule.

Description of the parameters for SINEC L1 operation

To ”prepare” the CP 521 SI for SINEC L1 operation, the interface of the CP must be assigned the following parameters:

- Driver number
  The ”operating mode” of the CP 521 SI can be set with the driver number. All other parameters depend on the driver number set. There are two different driver numbers for SINEC L1 operation:
  - Driver number 4: ”normal” SINEC L1 operation; PLC with CP 521 SI is SINEC L1 slave
  - Driver number 5: point-to-point connection; PLC with CP 521 SI is SINEC L1 master for a single direct-connected slave.
- Slave number
  A slave number (range: 1 to 30) must be specified in addition only if the driver number=4.

The CP 521 SI ”assumes” these parameters to be in parameter block 7, so this parameter block number must also be specified.

9.2.1 Passing Parameters for SINEC L1 Operation to the Module Via the User Program

The following figure shows the order of the parameters in the eight-byte transfer memory (to simplify presentation, the addresses are specified with ”Module address+X” (Maddr+X), X=0, 1 to 7):

Address	Contents
Maddr+0	90_H
Maddr+1	74_H
Maddr+2	Slave No.
Maddr+3	Irrelevant
Maddr+4	Irrelevant
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for ”Assign interface parameters” job request

”7” is the parameter block, ”4” is the driver number for ”normal” SINEC L1 operation.

Only enter if the driver number=4: Slave No. 1 to 30; otherwise irrelevant

Section 9.6 "STEP 5 Program for Data Transmission with SINEC L1 Driver" contains a concrete example of parameter assignment via the user program.

Note

The CP 521 SI can also communicate via the RS-232C (V.24) interface in SINEC L1 operation (Example: Direct connection between two CP 521 SIs).
Procedure:

- Terminal diagram as per Figure 9-2.
- First transfer parameter block 0 (Section 6.3.2); all the parameters of this block must lie within the prescribed range - however, the CP interprets only the parameter for the interface (set RS-232C/V.24!)
- Then transfer parameter block 7 in the next program scan cycle (for SINEC L1).

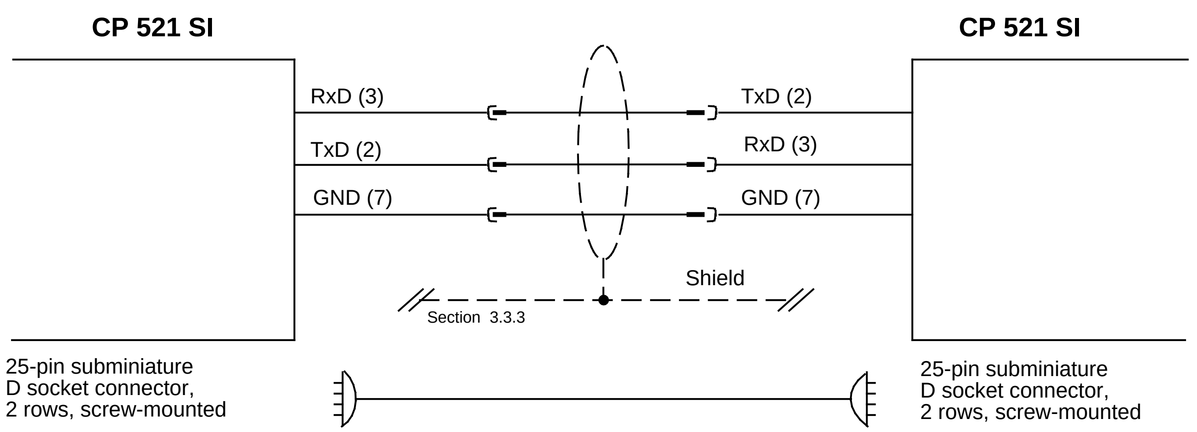


Figure 9-2. Terminal Diagram for a CP 521 SI - CP 521 SI Connection (RS-232C/V.24)

9.2.2 Providing Parameters for SINEC L1 Operation Over the Memory Submodule of the Module

If you want to transfer the parameters for SINEC L1 operation to a memory submodule for the CP 521 SI, you must first create DB1 for the parameters. Assign parameters only to parameter block 7: make sure that the individual parameters of parameter block 7 are separated from each other!

Example: Assigning parameters to the CP 521 SI in DB1 for SINEC L1 operation:

```
   DB 1

 0: KS':7$4$2$ ';
 4:
```

Slave No. 2
Separator
Driver number 4 (”normal” SINEC L1 operation)
Parameter block number 7
Parameter block separator

Figure 9-3. Meaning of the Parameters in DB1 in the Case of SINEC L1 Parameter Assignment

After you have programmed DB1, proceed as follows:

- Transfer the DB1 you have just programmed to a memory submodule with the ”EPROM/EEPROM” package
- Switch off the PLC
- Plug the E(E)PROM into the CP 521 SI
- Switch on the PLC

During restart, the CP 521 SI accepts the parameters stored in DB1 on the memory submodule.

Note

If there is no memory submodule plugged into the CP 521 SI and a backup battery has been inserted, the CP 521 SI is automatically initialized with the data in the RAM on power-up (i.e. with the parameters last assigned). In other words, the module is activated in the same driver mode following POWER-ON as it was before POWER-OFF.

9.3 Send Procedure from the Point of View of the Control Program

The following section describes the principle of sending data in mode 4 (PLC with CP 521 SI is slave on the SINEC L1 LAN). A detailed description is given of the eight-byte blocks which

- are transferred as a job request to the transfer memory of the CP 521 SI (PIQ) and
- are read from the transfer memory of the CP 521 SI as an acknowledgement (PII).

Reminder:
Transferring eight-byte blocks of the transfer memory and reading acknowledgements from the transfer memory refer to communications between the CPU and the CP 521 SI.

Communications between the CP 521 SI and a SINEC L1 partner over the SINEC L1 LAN can be controlled with the ”Send coordination byte” (CBS).

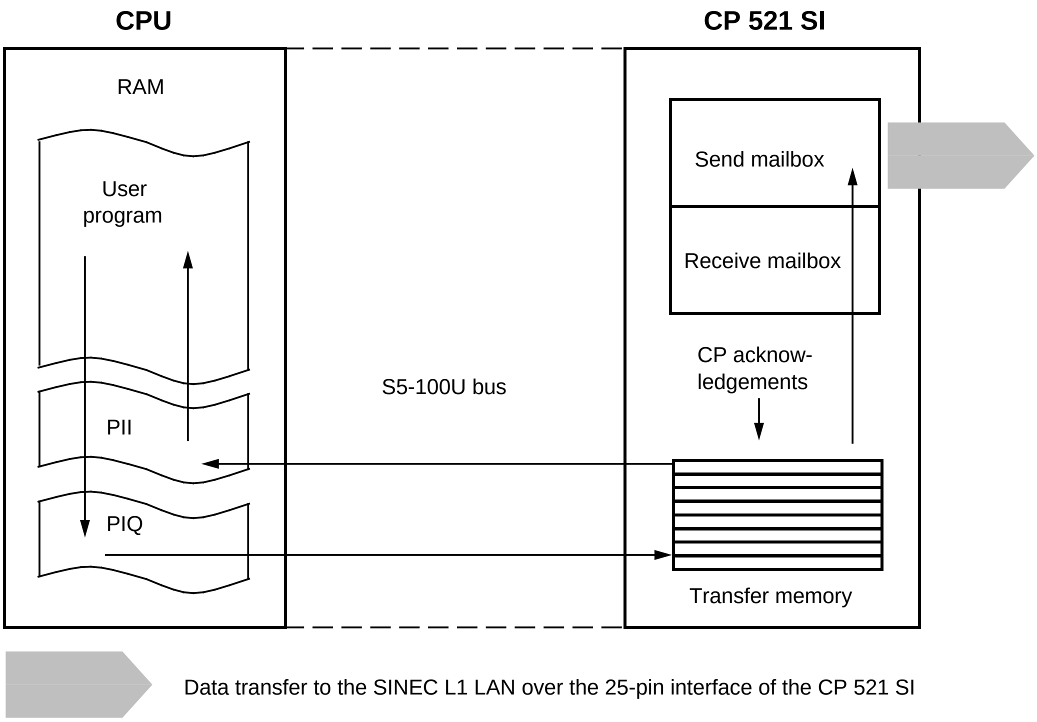


Figure 9-4. Sending Data over the CP 521 SI

Write send job request into the PIQ
Evaluate the CP acknowledgement for the send job request
If the send job request has been accepted by the CP:
Write the first message block into the PIQ
Evaluate the CP acknowledgement for the first message block
Steps and are repeated for the next message blocks if the send job request has several frames; you can write up to eleven blocks for a send job request into the PIQ because a SINEC L1 message frame can be up to 64 bytes long.
After the last message block has been written into the PIQ:
Evaluate the terminating acknowledgement of the CP.

The following is a detailed description of each of these steps.

Write the send job request into the PIQ
(to simplify presentation, the addresses are specified with "Module address+X" (Maddr+X), X=0, 1 to 7)

Address	Contents
Maddr+0	$A0_H$
Maddr+1	01_H
Maddr+2	Irrelevant
Maddr+3	Send length
Maddr+4	Destination
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for "Send job request" ($A001_H$)

Send length <65 bytes "net data"

Destination $=00_H$: Master
$=1$ to 30_H: Slave No.
$=31_H$: Broadcast (Send to all)

Wait for one of the following CP acknowledgements:

Send request accepted

Address	Contents
Maddr+0	50_H
Maddr+1	01_H
Maddr+2	Irrelevant
Maddr+3	00_H
Maddr+4	Number of messages in CP receive buffer
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Send request rejected

Contents
50_H *
00_H
CBS **
00_H
Number of messages in CP receive buffer
Irrelevant
Irrelevant
Irrelevant

* If the request has been rejected, the CBS gives an indication of the type of error:
** CBS=01_H: Error in data transfer (wrong block number)
CBS=81_H: Previous send job request is not yet completed

If the send job request has been accepted by the CP:
Write the first message block into the PIQ

Address	Contents
Maddr+0	$B0_H$
Maddr+1	01_H
Maddr+2	Data
Maddr+3	Data
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Code for "Block transfer to CP"

Number of the message block (1 in this case)

Wait for acknowledgement for the first message block:

Positive acknowledgement

Address	Contents
Maddr+0	50_H
Maddr+1	01_H
Maddr+2	Irrelevant
Maddr+3	01_H
Maddr+4	Numb. of messages in CP rec. buffer
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Message block number

Negative acknowledgement

Contents
50_H *
00_H
CBS **
01_H
Numb. of messages in CP rec. buffer
Irrelevant
Irrelevant
Irrelevant

* If the acknowledgement is negative, the CBS gives an indication of the type of error:

** The meaning of the "Maddr+2" byte in the case of a rejected job request (CBS) is explained in .

Steps and are repeated for the other message blocks if the send job request has several frames. You can write up to eleven blocks into the PIQ for a send job request because a SINEC L1 message frame can be up to 64 bytes long.

After the last message block has been written into the PIQ:
Wait for terminating aknowledgement:

Address	Contents
Maddr+0	50_H
Maddr+1	00_H
Maddr+2	CBS
Maddr+3	00_H
Maddr+4	Numb. of messages in CP rec. buffer
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for ”Terminating acknowledgement”

Bit 0 acknowledges the send procedure to the SINEC L1 node (see below)

Structure of the Send coordination byte (CBS):

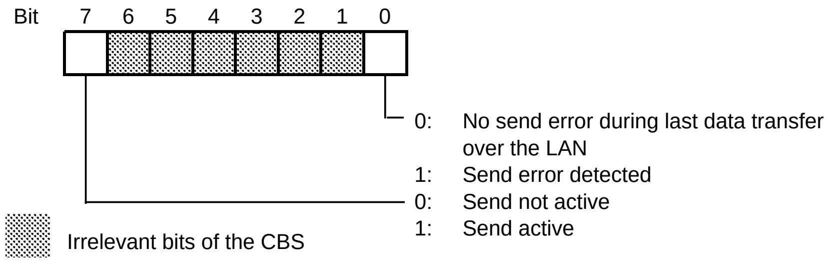


Figure 9-5. Structure of the Send Coordination Byte (CBS)

Summary:

The CP acknowledgement ”5000_H” in the ”Maddr+0” byte and ”Maddr+1” byte is the terminating acknowledgement for ”successful” or ”abortive” data transfer between the CPU and the CP 521 SI.

To differentiate between ”successful” and ”abortive”, the CBS must be evaluated in the control program in all cases:

CBS= 00_H Everything OK.
CBS= 81_H Previous send job request not yet completed.
CBS= 01_H Data transfer over the SINEC L1 LAN unsuccessful; data has not been sent
or
Error in data transfer between CPU and CP (e.g. wrong block number).

9.4 Receive Procedure from the Point of View of the Control Program

The following section describes the principle of receiving data in mode 4 (PLC with CP 521 SI is slave on the SINEC L1 LAN). A detailed description is given of the eight-byte blocks which

- are transferred as a job request to the transfer memory of the CP 521 SI (PIQ) and
- are read from the transfer memory of the CP 521 SI as an acknowledgement (PII).

Reminder:
Transferring eight-byte blocks of the transfer memory and reading acknowledgements from the transfer memory refer to communications between the CPU and the CP 521 SI.

Communications between the CP 521 SI and a SINEC L1 partner over the SINEC L1 LAN can be controlled with the ”Receive coordination byte” (CBR).

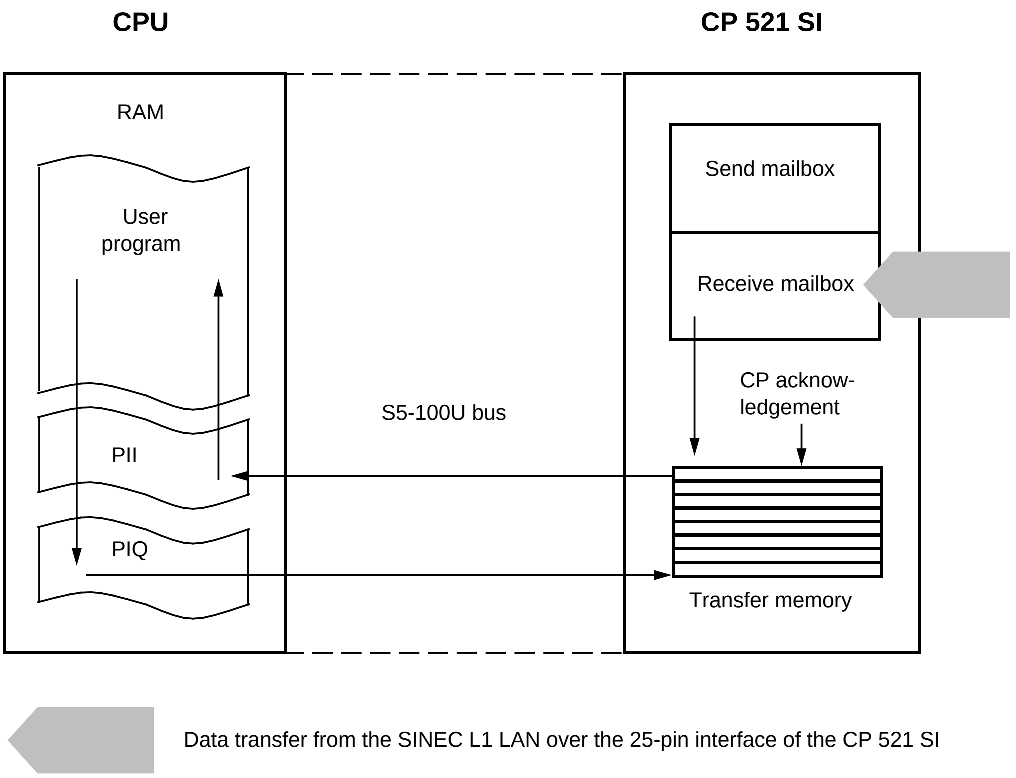


Figure 9-6. Receiving Data over the CP 521 SI

Write send job request into the PIQ
Evaluate the CP acknowledgement for the send job request
If the CP acknowledgement was positive and the first message block has been read in:
Acknowledge the message block read in
Read in the next message block and evaluate the CP acknowledgement for this message block
Steps and are repeated until the last message block has been read in.
Evaluate the terminating acknowledgement of the CP 521 SI.

Write the receive request into the PIQ
(to simplify presentation, the addresses are specified with ”Module address+X” (Maddr+X), X=0, 1 to 7).

Address	Contents
Maddr+0	A0$_H$
Maddr+1	80$_H$
Maddr+2	Irrelevant
Maddr+3	Irrelevant
Maddr+4	Irrelevant
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for ”Receive job request” (A080$_H$)

Wait for one of the following CP acknowledgements:

Receive request accepted

Address	Contents
Maddr+0	60$_H$
Maddr+1	01$_H$
Maddr+2	Length of the data packet (in bytes)
Maddr+3	Source *****
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Message block number (here: 1)

Request rejected

Contents
50$_H$ **
00$_H$
CBR ***
Irrelevant
Numb. of messages in CP rec. buffer
Irrelevant
Irrelevant
Irrelevant

***** Meaning of the ”Maddr+3” byte (source):
00$_H$: Master
1 to 30$_H$: Slave No.

****** If the acknowledgement is negative, the CBR indicates the type of error

******* The meaning of the ”Maddr+2” in the case of a rejected job request (CBR) is explained in step

If the receive job request has been accepted:
Acknowledge the first message block

Address	Contents
Maddr+0	$C0_H$
Maddr+1	01_H
Maddr+2	Irrelevant
Maddr+3	Irrelevant
Maddr+4	Irrelevant
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for "Acknowledge message block"

Number of the message block
(1 in this case)

The CP transfers the next message block:

Positive acknowledgement

Address	Contents
Maddr+0	60_H
Maddr+1	02_H
Maddr+2	Data
Maddr+3	Data
Maddr+4	Data
Maddr+5	Data
Maddr+6	Data
Maddr+7	Data

Message block number (2 in this case)

Negative acknowledgement

Contents
50_H *
00_H
CBR **
Irrelevant
Numb. of messages in CP rec. buffer
Irrelevant
Irrelevant
Irrelevant

* If the acknowledgement is negative, the CBR indicates the type of error

** The meaning of the "Maddr+2" in the case of a rejected job request (CBR) is explained in step

Steps and are repeated if the receive job request contains several blocks; you can read in up to eleven blocks per message frame because a SINEC L1 message frame can be up to 64 bytes long.

After the last block has been read:
Wait for the following terminating acknowledgement from the CP:

Address	Contents
Maddr+0	50_H
Maddr+1	00_H
Maddr+2	CBR
Maddr+3	00_H
Maddr+4	Numb. of messages in CP rec. buffer
Maddr+5	Irrelevant
Maddr+6	Irrelevant
Maddr+7	Irrelevant

Code for ”Terminating acknowledgement”

The meaning of the individual bits of the CBR can be seen in the following figure

Structure of the receive coordination byte (CBR):

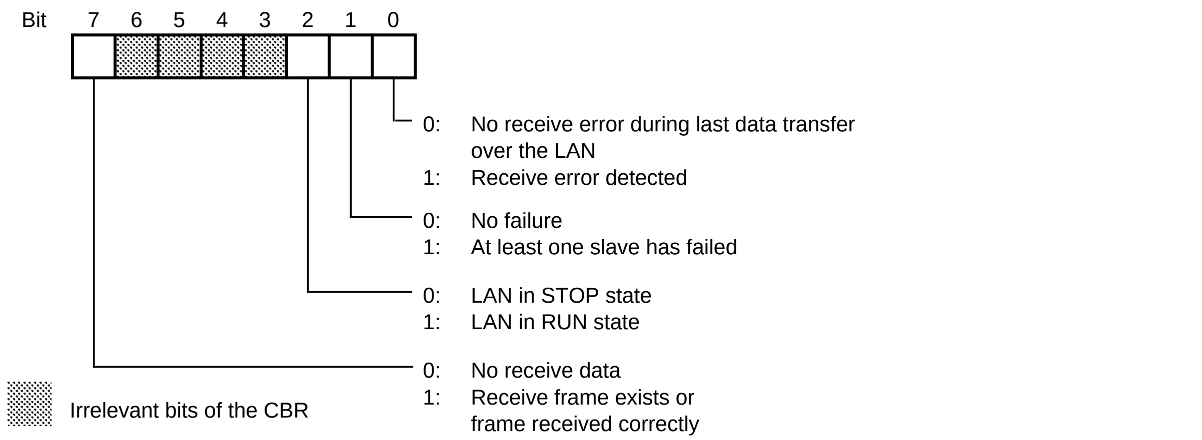


Figure 9-7. Structure of the Coordination Byte Receive (CBR)

Summary:

The CP acknowledgement ”5000_H” in the ”Maddr+0” byte and ”Maddr+1” byte is the terminating acknowledgement for ”successful” or ”abortive” data transfer between the CPU and the CP 521 SI.

To differentiate between ”successful” and ”abortive”, the CBR must be evaluated in the control program in all cases.

Note

The CPU job request $D0_H$ "Delete receive mailbox" is not relevant for the SINEC L1 driver since not more than one message frame can be entered in the receive mailbox when this driver is used.

9.5 Point-to-Point Connection

The following section is concerned with a special application of SINEC L1 operation: point-to-point connection (mode 5).

Mode 5 means: the CP 521 SI can be connected to a SINEC L1 slave without a second module being necessary. Examples of slaves are an S5-95U PLC, an S5-100U PLC (with CPU 102/103) or an S5-115U PLC.

In mode 5, the CP 521 SI is the ”Point-to-point master” for the direct SINEC L1 slave directly connected to it.

9.5.1 Connecting the SINEC L1 Slave Direct to the Interface of the CP 521 SI

The CP 521 SI can be connected to the slave in two ways:

- Over a bus cable with BT 777 bus terminals (beginning of the chapter for wiring arrangement)
  or
- Over a direct line (if both devices are less than 1000 m apart). Use a four-wire shielded cable with a cross-section of at least 0.14 mm² for this purpose. We recommend the SIMATIC cable 6ES5 707-1AA00.

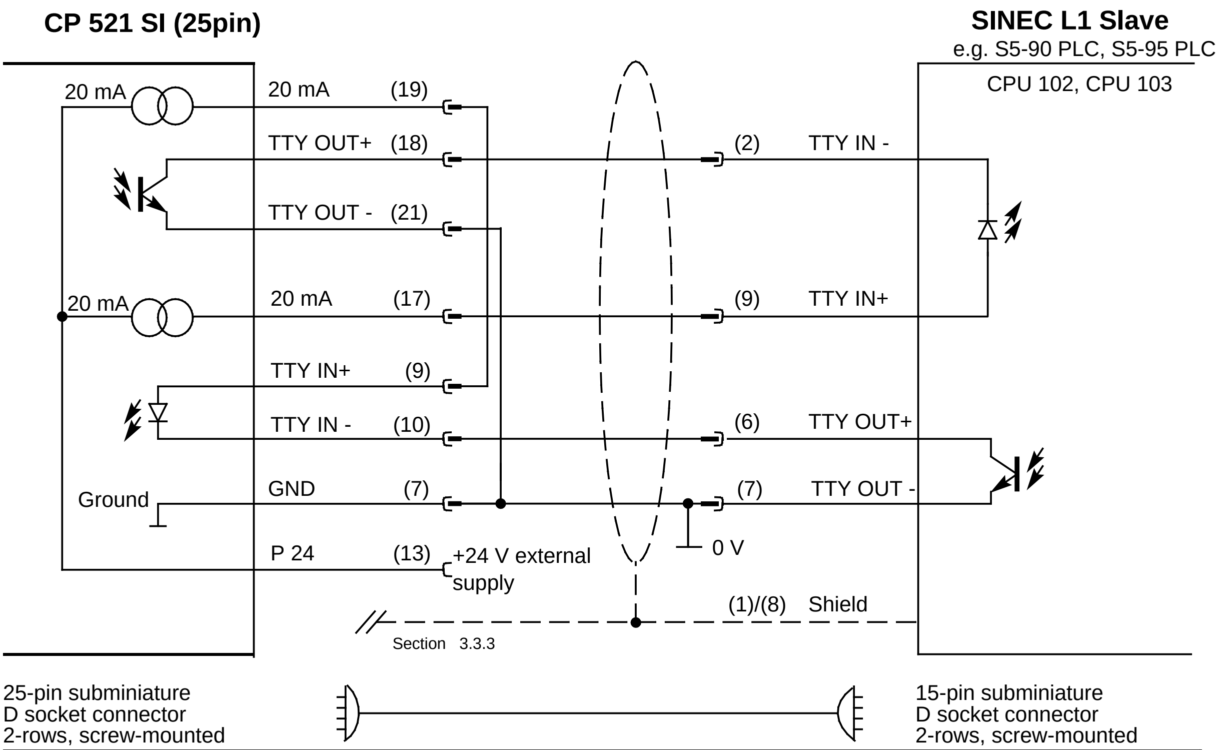


Figure 9-8. Terminal Diagram CP 521 SI (TTY Active) - SINEC L1 Slave

9.5.2 Assigning Parameters to the CP 521 SI as ”Point-to-Point Master”

When assigning parameters, you need only ensure that mode 5 is set in parameter block 7 (Section 9.2). Everything else is set automatically by the CP 521 SI.

9.5.3 Programming Data Transfer between the CP 521 SI and the Slave

You program data transfer between the CP 521 SI and the SINEC L1 slave as described in the preceding section. The slave is addressed as slave No. 1.

9.6 STEP 5 Program for Data Transmission with the SINEC L1 Driver

The following is a complete STEP 5 program for data transmission with the SINEC L1 driver, and is structured as follows:

- DB21: Send data (source DB)
- DB22: Receive data (destination DB)

- FB3/FB4: Parameter assignment FB (user-configurable)
  Assign SINEC L1 driver parameters (FB3) / Assign SINEC L1 point-to-point connection parameters (FB4)
- FB20: Pulse generation for FB21
- FB21: Switching between send and receive
- FB200: Send FB (user-configurable; can only be used for CPU 103)
- FB201: Receive FB (user-configurable; can only be used for CPU 103)

- OB1:
  - Invokes parameter assignment FB (FB3 is the default FB; if you want to connect a SINEC L1 slave to the CP 521 SI, you must replace FB3 by FB4 in the program listing)
    If the CP 521 SI has been assigned its parameters, this FB is no longer invoked
  - Invokes FB21, which, in turn, invokes either the send FB (FB200) or the receive FB (FB201).
- OB21/22: Defaults for the parameter assignment FB

The source and destination DBs for send and receive data, which you must set up before the start of the program, are not represented.

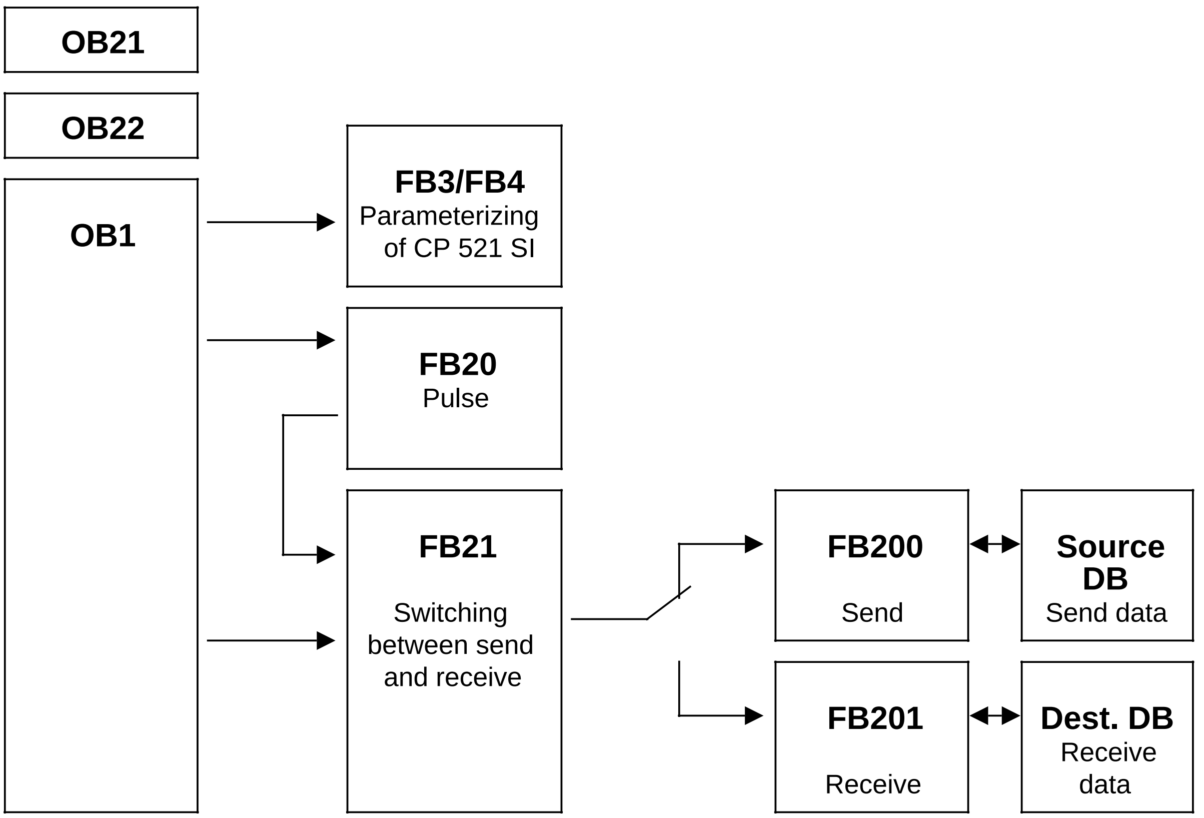


Figure 9-9. Structure of the STEP 5 Program for Data Transmission with the SINEC L1 Driver

```
OB 1                                  D:SINEC@ST.S5D                    LEN=20

        SEGMENT  1     0000
        0000      :AN  F  100.7
        0001      :JC  FB   3                  Assign CP 521 SI parameters
        0002 NAME :CP-PARAM
        0003 BGAD :    KF +64
        0004 OK   :    F  100.7
        0005      :AN  F  100.7
        0006      :BEC
        0007      :
        0008      :JU  FB  20                  Pulse generator FB
        0009 NAME :PULSE
        000A      :
        000B      :JU  FB  21                  Coordinate send and receive
        000C NAME :DISTRIB
        000D      :
        000E      :BE

OB 21                                 D:SINEC@ST.S5D                    LEN=11

        SEGMENT  1     0000
        0000      :L   KH 0000                 Defaults for parameter assignment FB
        0002      :T   FW 100
        0003      :
        0004      :
        0005      :BE

OB 22                                 D:SINEC@ST.S5D                    LEN=7

        SEGMENT 1      0000
        0000      :JU  OB  21
        0001      :BE
```

```
FB 3                                    D:SINEC@ST.S5D          LEN=130

        SEGMENT  1      0000
        NAME :CP-PARAM                                  SINEC L1 driver
        DES  :BGAD       I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :OK         I/Q/D/B/T/C: Q  BI/BY/W/D: BI

        000B       :LW  =BGAD                           This FB uses flag words
        000C       :T   FW 102                          FW 100 to 108 to assign
        000D       :I        2                          the CP 521 SI (version A02)
        000E       :T   FW 104                          parameters.
        000F       :I        2                          FW 102 and 108 are addresses of
        0010       :T   FW 106                          the CP.
        0011       :I        2                          FW 100 is used for execution
        0012       :T   FW 108                          of the FB.
        0013       :
        0014       :DO  FW 102                          Is CP still in
        0015       :L   IW   0                          restart routine?
        0016       :L   KH 0F00
        0018       :AW
        0019       :L   KH 0F00
        001B       :!=F
        001C       :BEC
        001D       :***


        SEGMENT  2      001E
        001E       :A   F  100.0                        Assign parameters to first block
        001F       :JC  =BL-E
        0020       :L   KH 9000                         Job block 0
        0022       :DO  FW 102
        0023       :T   QW   0
        0024       :L   KH 0800                         irrelevant  irrelevant
        0026       :DO  FW 104
        0027       :T   QW   0
        0028       :L   KH 0001                         irrelevant  Interface
        002A       :DO  FW 106
        002B       :T   QW   0
        002C       :L   KH 0100                         irrelevant  irrelevant
        002E       :DO  FW 108
        002F       :T   QW   0
        0030       :
        0031       :DO  FW 102                          Wait for positive
        0032       :L   IW   0                          acknowledgement 5XXXH
        0033       :L   KH 5000
        0035       :AW
        0036       :L   KH 5000
        0038       :><F
        0039       :BEC
        003A       :DO  FW 108                          Check whether acknowledgement
        003B       :L   IW   0                          belongs to job, otherwise wait
        003C       :L   KH 9000                         for right acknowledgement
        003E       :><F
        003F       :BEC
        0040 BL-E  :AN  F  100.0                        Set step flag
        0041       :S   F  100.0
        0042       :***
```

```
SEGMENT  3     0043
0043      :A   F  100.1                       Assign parameters to second block
0044      :JC  =BL-E
0045      :L   KH 9074                        Block 7    SINEC L1 mode
0047      :DO  FW 102
0048      :T   QW   0
0049      :L   KH 0200                        Slave number  irrelevant
004B      :DO  FW 104
004C      :T   QW   0
004D      :L   KH 0000                        irrelevant    irrelevant
004F      :DO  FW 106
0050      :T   QW   0
0051      :L   KH 0000                        irrelevant    irrelevant
0053      :DO  FW 108
0054      :T   QW   0
0055      :
0056      :DO  FW 102                         Wait for positive
0057      :L   IW   0                         acknowledgement 5XXXH
0058      :L   KH 5000
005A      :AW
005B      :L   KH 5000
005D      :><F
005E      :BEC
005F      :DO  FW 108                         Check whether acknowledgement
0060      :L   IW   0                         belongs to job, otherwise wait
0061      :L   KH 9074                        for right acknowledgement
0063      :><F
0064      :BEC
0065 BL-E :AN  F  100.1                       Set step flag
0066      :S   F  100.1
0067      :
0068      :***


SEGMENT  4     0069
0069      :A   F  100.2                       Last block deletes
006A      :JC  =BL-E                          acknowledgement 5000H.
006B      :L   KH 0000
006D      :DO  FW 102
006E      :T   QW   0
006F      :
0070      :DO  FW 102
0071      :L   IW   0
0072      :L   KH 5000
0074      :AW
0075      :L   KH 0000
0077      :><F
0078      :BEC
0079 BL-E :AN  F  100.2
007A      :S   F  100.2
007B      :S   =OK                            Set parameter assignment flag
007C      :BE
```

```
FB 4                                  D:SINEC@ST.S5D          LEN=130

        SEGMENT  1     0000           SINEC L1 point-to-point master
        NAME :CP-PARAM                                  SINEC L1 driver
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :OK        I/Q/D/B/T/C: Q  BI/BY/W/D: BI

        000B      :LW  =BGAD                            This FB uses flag words
        000C      :T   FW 102                           FW 100 to 108 to assign
        000D      :I        2                           the CP 521 SI (version A02)
        000E      :T   FW 104                           parameters.
        000F      :I        2                           FW 102 and 108 are addresses
        0010      :T   FW 106                           of the CP.
        0011      :I        2                           FW 100 is used for execution
        0012      :T   FW 108                           of the FB.
        0013      :
        0014      :DO  FW 102                           Is CP still in
        0015      :L   IW   0                           restart routine?
        0016      :L   KH 0F00
        0018      :AW
        0019      :L   KH 0F00
        001B      :!=F
        001C      :BEC
        001D      :***


        SEGMENT  2     001E
        001E      :A   F  100.0                         Assign parameters to first block
        001F      :JC  =BL-E
        0020      :L   KH 9000                          Job block 0
        0022      :DO  FW 102
        0023      :T   QW   0
        0024      :L   KH 0800                          irrelevant  irrelevant
        0026      :DO  FW 104
        0027      :T   QW   0
        0028      :L   KH 0000                          irrelevant  Interface
        002A      :DO  FW 106
        002B      :T   QW   0
        002C      :L   KH 0100                          irrelevant  irrelevant
        002E      :DO  FW 108
        002F      :T   QW   0
        0030      :
        0031      :DO  FW 102                           Wait for positive
        0032      :L   IW   0                           acknowledgement 5XXXH
        0033      :L   KH 5000
        0035      :AW
        0036      :L   KH 5000
        0038      :><F
        0039      :BEC
        003A      :DO  FW 108                           Check whether acknowledgement
        003B      :L   IW   0                           belongs to job, otherwise wait
        003C      :L   KH 9000                          for right acknowledgement
        003E      :><F
        003F      :BEC
        0040 BL-E :AN  F  100.0                         Set step flag
        0041      :S   F  100.0
        0042      :***
        SEGMENT  3     0043
        0043      :A   F  100.1                         Assign parameters to second block
        0044      :JC  =BL-E
        0045      :L   KH 9075                          Block 7       Point-to-point
        0047      :DO  FW 102                                         link
        0048      :T   QW   0
        0049      :L   KH 0000                          irrelevant    irrelevant
```

```
004B      :DO  FW 104
004C      :T   QW   0
004D      :L   KH 0000                          irrelevant    irrelevant
004F      :DO  FW 106
0050      :T   QW   0
0051      :L   KH 0000                          irrelevant    irrelevant
0053      :DO  FW 108
0054      :T   QW   0
0055      :
0056      :DO  FW 102                           Wait for positive
0057      :L   IW   0                           acknowledgement 5XXXH
0058      :L   KH 5000
005A      :AW
005B      :L   KH 5000
005D      :><F
005E      :BEC
005F      :DO  FW 108                           Check whether acknowledgement
0060      :L   IW   0                           belongs to job, otherwise wait
0061      :L   KH 9075                          for right acknowledgement
0063      :><F
0064      :BEC
0065 BL-E :AN  F  100.1                         Set step flag
0066      :S   F  100.1
0067      :
0068      :***


SEGMENT  4     0069
0069      :A   F  100.2                         Last block deletes
006A      :JC  =BL-E                            acknowledgement 5000H.
006B      :L   KH 0000
006D      :DO  FW 102
006E      :T   QW   0
006F      :
0070      :DO  FW 102
0071      :L   IW   0
0072      :L   KH 5000
0074      :AW
0075      :L   KH 0000
0077      :><F
0078      :BEC
0079 BL-E :AN  F  100.2
007A      :S   F  100.2
007B      :S   =OK                              Set parameter assignment flag
007C      :BE
```

```
FB 20                                 D:SINEC@ST.S5D                   LEN=19

        SEGMENT  1     0000
        NAME :PULSE

        0005      :AN  F  101.1
        0006      :L   KT 030.1
        0008      :SP  T  101
        0009      :AN  T  101
        000A      :AN  F  101.1
        000B      :=   F  101.1
        000C      :S   F  101.0                        Pulse flag is set after 3 s.
        000D      :BE


FB 21                                 B:SINEC@ST.S5D                   LEN=42

        SEGMENT  1     0000
        NAME :DISTRIB

        0005      :AN  F   80.2                        Send not active
        0006      :JC  =M001
        0007      :
        0008      :JU  FB 200                          Process send FB
        0009 NAME :SEN  VAR
        000A BGAD :    KF +64
        000B Q-DB :    DB  21
        000C KBS  :    FY  90
        000D BLLA :    KF +12
        000E ZIEL :    KF +0
        000F      :R   F  101.0
        0010      :BEU
        0011      :
        0012 M001 :AN  F   80.6                        Receive not active
        0013      :JC  =M002
        0014      :
        0015      :JU  FB 201                          Process receive FB
        0016 NAME :REC  VAR
        0017 BGAD :    KF +64
        0018 Z-DB :    DB  22
        0019 KBE  :    FY  91
        001A      :
        001B      :BEU
        001C      :
        001D M002 :A   F  101.0                        Set pulse flag
        001E      :S   F   80.2                        Set send flag
        001F      :R   F  101.0                        Reset pulse flag
        0020      :A   F   80.2
        0021      :BEC
        0022      :AN  F   80.6                        Set receive flag
        0023      :S   F   80.6
        0024      :BE
```

```
FB 200                               D:SINEC@ST.S5D         LEN=227

        SEGMENT  1     0000          Sending to slave X
        NAME :SEN  VAR
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Q-DB      I/Q/D/B/T/C: B
        DES  :KBS       I/Q/D/B/T/C: Q  BI/BY/W/D: BY
        DES  :BLLA      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :ZIEL      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

        0014      :LW  =BGAD                      Load module address
        0015      :T   FW 234
        0016      :
        0017      :DO  FW 234                     Is CP busy?
        0018      :L   IW   0
        0019      :L   KH 0F00
        001B      :AW
        001C      :L   KH 0F00
        001E      :!=F
        001F      :BEC                            If yes, block end conditionally
        0020      :***


        SEGMENT  2     0021
        0021      :A   F   80.0                   Flag for job initiated
        0022      :JC  =M001                      If this flag is set, do not
        0023      :                               initiate a new job
        0024      :
        0025      :DO  FW 234                     Check whether a job can be
        0026      :L   IW   0                     initiated
        0027      :L   KH C000
        0029      :AW
        002A      :L   KF +0
        002C      :><F
        002D      :BEC
        002E      :
        002F      :L   KF +0                      Reset block counter
        0031      :T   FW 252
        0032      :LW  =BLLA                      Buffer length
        0033      :T   FW 254
        0034 M003 :L   FW 252
        0035      :NOP 1                          Increment block counter
        0037      :T   FW 252
        0038      :L   FW 254                     Subtract the number of bytes
        0039      :NOP 1                          of a block from the length
        003B      :T   FW 254                     until zero is reached
        003C      :L   KF +0
        003E      :<=F
        003F      :JC  =M002                      Number of blocks fixed
        0040      :JU  =M003                      Final number of blocks not
        0041      :                               yet reached
        0042 M002 :
        0043      :L   FW 234                     Set address pointer
        0044      :L   KF +2
        0046      :+F
        0047      :T   FW 236                     Module address + 2
        0048      :L   KF +2
        004A      :+F
        004B      :T   FW 238                     Module address + 4
        004C      :L   KF +2
        004E      :+F
        004F      :T   FW 240                     Module address + 6
        0050      :
```

```
0051      :L   KH A001                         Initiate send job
0053      :DO  FW 234
0054      :T   QW   0
0055      :LW  =BLLA
0056      :DO  FW 236                          Store block length in word 1
0057      :T   QW   0
0058      :LW  =ZIEL
0059      :SLW      8                          Store destination to which
005A      :DO  FW 238                          the message frame is to be
005B      :T   QW   0                          sent in word 2
005C      :
005D      :L   KH B001                         Identifier for block
005F      :T   FW 242                          acknowledgement (first block)
0060      :L   KF +0                           Store comparison value
0062      :T   FW 244                          (block 1)
0063      :
0064      :L   KH 0001                         DB
0066      :T   FW 246                          Starting address
0067      :L   KF +1
0069      :+F
006A      :T   FW 248                          Starting address + 1
006B      :L   KF +1
006D      :+F
006E      :T   FW 250                          Starting address + 2
006F      :
0070      :AN  F   80.0                        Set flag for send
0071      :S   F   80.0                        job initiated
0072      :R   F   80.1                        Reset error flag
0073      :BEU
0074 M001 :***

SEGMENT  3     0075
0075      :DO  FW 234                          Load input word 0 and
0076      :L   IW   0                          check whether job has
0077      :L   KH F00F                         been accepted
0079      :AW
007A      :L   KH 5001
007C      :><F
007D      :JC  =M001
007E      :
007F      :DO  FW 236                          Check whether the right block
0080      :L   IW   0                          can be received
0081      :L   KH 00FF
0083      :AW
0084      :L   FW 244
0085      :><F
0086      :BEC
0087      :
0088      :L   FW 242                          Check whether a further block
0089      :L   KH 00FF                         can be sent
008B      :AW
008C      :L   FW 252
008D      :>F
008E      :BEC
008F      :
0090      :DO  =Q-DB                           Open source DB
0091      :L   FW 242                          Announce block and
0092      :DO  FW 234                          transfer it
0093      :T   QW   0
0094      :
0095      :DO  FW 246                          Word 0
0096      :L   DW   0
0097      :DO  FW 236
0098      :T   QW   0
```

```
0099       :
009A       :DO  FW 248                      Word 1
009B       :L   DW   0
009C       :DO  FW 238
009D       :T   QW   0
009E       :
009F       :DO  FW 250                      Word 2
00A0       :L   DW   0
00A1       :DO  FW 240
00A2       :T   QW   0
00A3       :
00A4       :L   FW 242                      Update block acknowledgement
00A5       :L   KF +1
00A7       :+F
00A8       :T   FW 242
00A9       :
00AA       :L   FW 244                      Update block number
00AB       :L   KF +1
00AD       :+F
00AE       :T   FW 244
00AF       :                                Increment data word pointer
00B0       :L   FW 246                      Word 0
00B1       :L   KF +3
00B3       :+F
00B4       :T   FW 246
00B5       :
00B6       :L   FW 248                      Word 1
00B7       :L   KF +3
00B9       :+F
00BA       :T   FW 248
00BB       :
00BC       :L   FW 250                      Word 2
00BD       :L   KF +3
00BF       :+F
00C0       :T   FW 250
00C1       :BEU
00C2 M001 :***


SEGMENT  4      00C3
00C3       :DO  FW 234                      Check whether terminating
00C4       :L   IW   0                      acknowledgement present
00C5       :L   KH F00F
00C7       :AW
00C8       :L   KH 5000
00CA       :><F
00CB       :JC  =M001
00CC       :DO  FW 236                      Read in input byte 3 and transfer
00CD       :L   IB   0                      it to the CBS
00CE       :T   =KBS
00CF       :DO  FW 236
00D0       :L   IW   0                      Send without error?
00D1       :L   KH 0100
00D3       :AW
00D4       :JZ  =M002
00D5       :AN  F   80.1                    No; job terminated
00D6       :S   F   80.1                    with error
00D7 M002 :R    F   80.2                    Yes; enable for new send or
00D8       :R   F   80.0                    receive job
00D9       :L   KH 0000                     Delete terminating acknowledgement
00DB       :DO  FW 234
00DC       :T   QW   0
00DD M001 :BE
```

```
FB 201                                D:SINEC@ST.S5D          LEN=172

        SEGMENT  1     0000          Receiving from station X
        NAME :REC  VAR
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :Z-DB      I/Q/D/B/T/C: B
        DES  :KBE       I/Q/D/B/T/C: Q  BI/BY/W/D: BY

        000E      :LW  =BGAD                           Load and store module
        000F      :T   FW 234                          address
        0010      :
        0011      :DO  FW 234                          Is CP busy?
        0012      :L   IW   0
        0013      :L   KH 0F00
        0015      :AW
        0016      :L   KH 0F00
        0018      :!=F
        0019      :BEC                                 Yes; block end conditionally
        001A      :***


        SEGMENT  2     001B
        001B      :A   F   80.4                        Flag for job already
        001C      :JC  =M001                           initiated
        001D      :
        001E      :DO  FW 234                          Check whether a job
        001F      :L   IW   0                          can be initiated
        0020      :L   KH C000
        0022      :AW
        0023      :L   KH 0000
        0025      :><F
        0026      :BEC
        0027      :
        0028      :L   FW 234                          Set address pointer
        0029      :L   KF +2
        002B      :+F
        002C      :T   FW 236                          Base address + 2 (word 1)
        002D      :L   KF +2
        002F      :+F
        0030      :T   FW 238                          Base address + 4 (word 2)
        0031      :L   KF +2
        0033      :+F
        0034      :T   FW 240                          Base address + 6 (word 3)
        0035      :
        0036      :L   KH A080                         Initiate receive job
        0038      :DO  FW 234
        0039      :T   QW   0
        003A      :L   KH 6001                         Read acknowledgements
        003C      :T   FW 242                          into flag words
        003D      :L   KH C001
        003F      :T   FW 244
        0040      :L   KH 0000                         Set pointer to destination data
        0042      :T   FW 246                          Starting address of destination
        0043      :L   KF +1
        0045      :+F
        0046      :T   FW 248                          Starting address + 1
        0047      :L   KF +1
        0049      :+F
        004A      :T   FW 250                          Starting address + 2
        004B      :
        004C      :S   F   80.4                        Flag for receive job
        004D      :R   F   80.5
        004E      :BEU
```

```
004F M001 :***
SEGMENT  3     0050
0050      :DO  FW 234                     Load input word 0 and
0051      :L   IW   0                     check whether receive
0052      :L   KH F0FF                    block present
0054      :AW
0055      :L   FW 242
0056      :><F
0057      :JC  =M001
0058      :DO  =Z-DB                      Open destination DB
0059      :DO  FW 236
005A      :L   IW   0
005B      :DO  FW 246                     Word for
005C      :T   DW   0                     data word pointer 0
005D      :
005E      :DO  FW 238
005F      :L   IW   0
0060      :DO  FW 248                     Word for
0061      :T   DW   0                     data word pointer 1
0062      :
0063      :DO  FW 240
0064      :L   IW   0
0065      :DO  FW 250                     Word for
0066      :T   DW   0                     data word pointer 2
0067      :
0068      :L   FW 244                     Acknowledge data transfer
0069      :DO  FW 234
006A      :T   QW   0
006B      :
006C      :L   FW 242                     Update block message
006D      :L   KF +1
006F      :+F
0070      :T   FW 242
0071      :
0072      :L   FW 244                     and update block number
0073      :L   KF +1
0075      :+F
0076      :T   FW 244
0077      :
0078      :L   FW 246                     Increment data word pointer word 0
0079      :L   KF +3
007B      :+F
007C      :T   FW 246
007D      :
007E      :L   FW 248                     Increment data word pointer word 1
007F      :L   KF +3
0081      :+F
0082      :T   FW 248
0083      :
0084      :L   FW 250                     Increment data word pointer word 2
0085      :L   KF +3
0087      :+F
0088      :T   FW 250
0089      :BEU
008A M001 :***


SEGMENT  4     008B
008B      :DO  FW 234                     Check whether terminating
008C      :L   IW   0                     acknowledgement present
008D      :L   KH F00F
008F      :AW
0090      :L   KH 5000
0092      :><F
```

```
0093      :JC  =M001
0094      :DO  FW 236                         Read in input byte 3 and transfer
0095      :L   IB   0                         it to the CBR
0096      :T   =KBE
0097      :DO  FW 236                         Receive without error?
0098      :L   IW   0
0099      :L   KH 0100
009B      :AW
009C      :JZ  =M002
009D      :AN  F   80.5                       No; receive terminated
009E      :S   F   80.5                       with error
009F M002 :R   F   80.6                       Yes; enable for new send or
00A0      :R   F   80.4                       receive job
00A1      :
00A2      :L   KH 0000                        Delete terminating acknowledgement
00A4      :DO  FW 234
00A5      :T   QW   0
00A6 M001 :BE
```

A	Summary	

A.1 Combinations of the Most Important Parameters . . . . A - 1

A.2 ASCII Code Table . . . . A - 2

A Summary

A.1 Combinations of the Most Important Parameters

	RS-232C (V.24)	TTY	XON/ XOFF	HW hand-shake	BUSY	Baud rate	Parity	Data format	Charac-ter delay
Printer driver	3	3	3	2	3	3	3	3	0
Interpretive ASCII	3	3	3	3	2	3	3	3	3
Transparent ASCII	3	3	2	3	2	3	3	3	3
Terminal	3	3	3	3	2	3	3	3	3
3964(R)	3	3	2	2	2	3	3	3	3
SINEC L1	3	10	2	2	2	10	10	10	2
RS-232C (V.24)		0	3	3	5	1	1	1	1
TTY	0		3	2	5	1	1	1	1
XON/ XOFF	3	3		6	7	1	1	1	1
HW handshake	3	2	6		9	1	1	1	1
BUSY	5	5	7	9		1	1	1	1
Baud rate	1	1	1	1	1		1	1	4
Parity	1	1	1	1	1	1		8	1
Data format	1	1	1	1	1	1	8		1
Character delay	1	1	1	1	1	4	1	1	

0: Not programmable
1: Parameters do not affect one another
2: Is not supported
3: Parameters not interdependent
4: Character delay ZVZ and baud rate must be in a reasonable relation to one another. Recommendation:

$$ZVZ \quad \frac{\text{2x (no. of bits in character frame) x1000}}{\text{baud rate}} \qquad \text{(in ms)}$$

5: In print mode only
6: HW handshake has priority
7: XON/XOFF has priority
8: Data format has priority
9: Busy signal or HW handshake are mode-dependent
10: Preset automatically

A.2 ASCII Code Table

Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
00	NUL	20	SP	40	@	60	,
01	SOH	21	!	41	A	61	a
02	STX	22	”	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	'	47	G	67	g
08	BS	28	(	48	H	68	h
09	HT (TAB)	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	k
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SO	2E	.	4E	N	6E	n
0F	SI	2F	/	4F	O	6F	o
10	DLE	30	0	50	P	70	p
11	DC1 (X-ON)	31	1	51	Q	71	q
12	DC2 (TAPE)	32	2	52	R	72	r
13	DC3 (X-OFF)	3	3	53	S	73	s
14	DC4 (TAPE)	34	4	54	T	74	t
15	NAK	35	5	55	U	75	u
16	SYN	36	6	56	V	76	v
17	ETB	37	7	57	W	77	w
18	CAN	38	8	58	X	78	x
19	EM	39	9	59	Y	79	y
1A	SUB	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	{
1C	FS	3C	<	5C	\	7C	I
1D	GS	3D	=	5D	]	7D	
1E	RS	3E	>	5E		7E	} (ALT MODE)
1F	US	3F	?	5F	-	7F	DEL (RUB OUT)

B Active and Passive Faults in Automation Equipment

B Active and Passive Faults in Automation Equipment

- Depending on the particular task for which the electronic automation equipment is used, both **active** as well as **passive** faults can result in a **dangerous** situation. For example, in drive control, an active fault is generally dangerous because it can result in unauthorized startup of the drive. On the other hand, a passive fault in a signalling function can result in a dangerous operating state not being reported to the operator.

- This differentiation of the possible faults and their classification into dangerous and non-dangerous faults, depending on the particular task, is important for all safety considerations in respect to the product supplied.

> **Warning**
>
> In all cases where a fault in automation equipment can result in severe personal injury or substantial damage to property, i.e. where a dangerous fault can occur, additional external measures must be taken or equipment provided to ensure or force safe operating conditions even in the event of a fault (e.g. by means of independent limit monitors, mechanical interlocks etc.).

Maintenance and Repair Procedure

If any measurement or testing work has to be carried out **on the CP 521 SI**, the rules and regulations set forth in the VBG 4.0 Accident Prevention Regulations" of the German Employees Liability Assurance Association must be observed, in particular §8 "Permissible exceptions when working on live parts".

Do not open up the CP 521 SI under any circumstances.

Repairs to an item of automation equipment may only be carried out by **Siemens service personnel** or **repair shops authorized by Siemens** to carry out such repairs

C Function Blocks to Support Data Transmission with the CP 521 SI

C Function Blocks to Support Data Transmission with the CP 521 SI

This Appendix contains a list of function blocks to support you in checking and optimizing data transmissions with the CP 521 SI.

You can link the function blocks listed into your cyclic user program (FB invoked cyclically in OB1) in all driver modes of the CP 521 SI.

The following function blocks are specially recommended for applications of the CP 521 SI in the ET 200U distributed I/O system.

- FB 15: Measures the transfer times of the send and receive message frames between the CPU and the CP 521 SI; this will enable you to optimize the initiation of send and receive operations with respect to time
- FB 17: Initializes a random DB (DB address can be specified)
- FB 18: Lists the jobs sent to the CP 521 SI and the acknowledgements received from the CP 521 SI (troubleshooting support)
- FB 19: Reassigns the CP 521 SI parameters if there is no data transferred between the CP and the CPU over a period of 30 seconds (QW 0 and IW 0 unchanged)

```
FB 15                                B:TOOL@@ST.S5D             LEN=46

        SEGMENT  1     0000          Execution time FW10 FW12

        The times (up to 0.99 seconds) are in flag words FW10 and FW12.
        This FB measures the execution time of the send and receive FBs.
        The FB assigns its own parameters via flags F80.2 and F80.6.
        The time value in the flag words is to be interpreted in 1/100 seconds.

        NAME :EXE TIME

        0005      :A   F   80.2                  When the send FB is executing,
        0006      :L   KT 999.0                  timer 10 is loaded with the
        0008      :                              value 999 and started as a
        0009      :SP  T   10                    pulse timer.
        000A      :L   KF +999                   The difference between 999
        000C      :L   T   10                    and the time value is then formed
        000D      :-F                            and, if the timer has been started,
        000E      :L   KF +999                   the difference (= execution time)
        0010      :!=F                           is stored in flag word 10.
        0011      :JC =M001
        0012      :NOP 1
        0013      :T   FW  10
        0014      :
        0015 M001 :AN  F   80.2                  The send and receive procedures are
        0016      :R   T   10                    identical, except for the fact
        0017      :                              that timer 11 is used and
        0018      :A   F   80.6                  the time is stored in
        0019      :L   KT 999.0                  flag word 12.
        001B      :SP  T   11
        001C      :L   KF +999
        001E      :L   T   11
        001F      :-F
        0020      :L   KF +999
        0022      :!=F
        0023      :JC  =M002
        0024      :NOP 1
        0025      :T   FW  12
        0026 M002 :AN  F   80.6
        0027      :R   T   11
        0028      :BE
```

```
FB 17                                B:TOOL@@ST.S5D                   LEN=37

        SEGMENT  1     0000          Initializing a DB with 0000H
        NAME :DB-ERASE
        DES  :D-DB      I/Q/D/B/T/C: B
        DES  :A-DW      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
        DES  :E-DW      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

        000E      :DO  =D-DB
        000F      :LW  =A-DW
        0010      :T   FW  60
        0011      :
        0012 M002 :L   KH 0000
        0014      :DO  FW  60
        0015      :T   DW   0
        0016      :LW  =E-DW
        0017      :L   FW  60
        0018      :<=F
        0019      :JC  =M001
        001A      :NOP 1
        001C      :T   FW  60
        001D      :JU  =M002
        001E M001 :
        001F      :BE
```

```
FB 18                                  B:TOOL@@ST.S5D                 LEN=100
                                                                      PAGE    1
SEGMENT  1     0000          Data transfer listing
This FB monitors the first two input words and the first two output words and
writes them into a DB, provided nothing has changed in the first input and first
output word. The function block can therefore record 64 scan cycles of the CPU
where something has changed in the I/O area of the CP, e.g. CP responses
to CPU jobs.

NAME :MONITOR
DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
DES  :Z-DB      I/Q/D/B/T/C: B

000B      :LW  =BGAD                         Store module address in flag
000C      :T   FW  96                        word 96.
000D      :I        2                        Store next highest address in
000E      :T   FW  94                        flag word 94.
000F      :
0010      :L   KH 0100                       If the pointer in the DB is at the
0012      :L   FW  98                        end address, a jump is made
0013      :<=F                               to label DB-V (DB FULL).
0014      :JC  =DB-V
0015      :DO  =Z-DB
0016      :
0017      :DO  FW  96                        The I/O area of the CP is read
0018      :L   IW   0                        in by indirect addressing and
0019      :DO  FW  98                        stored in the DB
001A      :T   DW   0
001B      :L   FW  98
001C      :L   KH 0001
001E      :+F
001F      :T   FW  98
0020      :
0021      :DO  FW  94
0022      :L   IW   0
0023      :DO  FW  98
0024      :T   DW   0
0025      :L   FW  98
0026      :L   KH 0001
0028      :+F
0029      :T   FW  98
002A      :
002B      :DO  FW  96
002C      :L   QW   0
002D      :DO  FW  98
002E      :T   DW   0
002F      :L   FW  98
0030      :L   KH 0001
0032      :+F
0033      :T   FW  98
0034      :
0035      :DO  FW  94
0036      :L   QW   0
0037      :DO  FW  98
0038      :T   DW   0
0039      :L   FW  98
003A      :L   KH 0001
003C      :+F
003D      :T   FW  98
003E      :
003F      :L   KH 0008                       When the FB is started, there
0041      :<F                                are not yet any words
0042      :BEC                               to compare
```

```
0043      :
0044      :L   FW  98                            A check is made to see whether IW0
0045      :D        8                            and/or QW0 has changed. If this is
0046      :T   FW  94                            not the case, the pointer in the
0047      :DO  FW  94                            DB is reset.
0048      :L   DW   0                            This part can also be omitted if
0049      :DO  FW  96                            all data cycles are to be
004A      :L   IW   0                            monitored, for instance, to detect
004B      :><F                                   any waiting times.
004C      :JC  =ENDE
004D      :L   FW  98
004E      :D        6
004F      :T   FW  94
0050      :DO  FW  94
0051      :L   DW   0
0052      :DO  FW  96
0053      :L   QW   0
0054      :><F
0055      :JC  =ENDE
0056      :L   FW  98
0057      :D        4
0058      :T   FW  98
0059 ENDE :BEU
005A DB-V :BEU                                   If monitoring is to be continuous,
005B      :L   KH 0000                           remove the BEU operation. The
005D      :T   FW  98                            pointer is then set to
005E      :BE                                    the beginning.
```

```
FB 19                                B:TOOL@@ST.S5D              LEN=64

        SEGMENT  1     0000          Watchdog for CP521xx

        This function block monitors IW0 and QW0 for any changes.
        If the words do not change, a flag is set after about 30 seconds and
        the CP is assigned new parameters.

        Flags used:

        FW110 to FW120

        Timer used:

        Timer 110

        Flag 10.3 is set when the timer has run down.

        Important: This FB assumes that jobs and acknowledgements are being constantly
        exchanged with the CP.

        NAME :WATCH-CP
        DES  :BGAD      I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

        0008      :LW  =BGAD                          Load module address and store
        0009      :T   FW 120                         it in FW 120.
        000A      :
        000B      :DO  FW 120
        000C      :L   IW   0                         Store IW0 in flag word 112
        000D      :T   FW 112
        000E      :
        000F      :DO  FW 120
        0010      :L   QW   0                         Store QW0 in flag word 114
        0011      :T   FW 114
        0012      :
        0013      :L   FW 116                         Compare old IW0 with
        0014      :L   FW 112                         new IW0;
        0015      :!=F                                if equal, jump
        0016      :JC  =M001
        0017      :T   FW 116                         Store current IW0
        0018      :A   F  110.0                       Reset equal-to flag
        0019      :R   F  110.0
        001A      :JU  =M002
        001B M001 :AN  F  110.0                       Set equal-to flag
        001C      :S   F  110.0
        001D      :
        001E      :L   FW 118                         Compare old QW0 with
        001F      :L   FW 114                         new QW0; if equal,
        0020      :!=F                                jump
        0021      :JC  =M003
        0022      :T   FW 118                         Store current QW0
        0023      :A   F  110.1                       Reset equal-to flag
        0024      :R   F  110.1
        0025      :JU  =M002
        0026 M003 :AN  F  110.1                       Set equal-to flag
        0027      :S   F  110.1
        0028      :
        0029 M002 :A   F  110.1                       If both equal-to flags have been
        002A      :A   F  110.0                       set, the timer is started and/or
        002B      :JC  =M004                          not reset; otherwise, if RLO = 0,
        002C      :A   F  110.3                       the timer runs for the preset time
        002D      :AN  F  110.3                       of, for example, 30 sec
        002E M004 :L   KT 030.2
        0030      :SD  T  110
```

```
0031     :A   T  110                          The flag is set after the preset
0032     :=   F  110.3                        time has elapsed
0033     :
0034     :AN  F  110.3                        If the flag is not set, the
0035     :BEC                                 block is ended conditionally;
0036     :L   KH 0000                         otherwise the CP is assigned
0038     :T   FW 100                          new parameters
0039     :T   FY  80
003A     :BE
```

D Connecting Cables; Ordering Form

Figures

D-1. Terminal Diagram CP 521 SI - CP 523 (RS-232C/V.24 Interface with Handshaking) . . . D - 5
D-2. Terminal diagram CP 521 SI - CP 524, CP 544, CP 525 and CPU 928B (RS-232C/V.24 Interface without Handshaking) . . . D - 5
D-3. Terminal diagram CP 524, CP 544, CP 525 and CPU 928B (TTY Active) - CP 521 SI (TTY Passive) . . . D - 6
D-4. Terminal diagram S5-90, CPU 102, CPU 103 and CPU 944 (TTY Active) - CP 521 SI (TTY Passive) . . . D - 7
D-5. Terminal Diagram PG 750 (PC 1620) (TTY Active, COM 1) - CP 521 SI (TTY Passive) . . . D - 8
D-6. Terminal Diagram PC with DF10 Interface Card (TTY Active) - CP 521 SI (TTY Passive) . . . D - 9
D-7. Terminal Diagram CP 521 SI - COM 2 of PG7xx (PC 1620), PC AT (RS-232C/V.24 Interface) . . . D - 10
D-8. Terminal Diagram CP 521 SI - COM 1 of a PG7xx, PC AT (RS-232C/V.24 Interface) . . . D - 10

D Connecting Cables; Ordering Form

Appendix D contains an ordering form (page D-3) with which you can order a connecting cable of your choice for communications between the CP 521 SI and a peripheral device.

All you have to do is copy the form, complete it and return it to the address at the top of the form.

Each connecting cable listed in the form is followed by the number of a terminal diagram in this manual for the particular cable. Typical connections not shown in preceding sections of the manual have been added on the following pages of this appendix (Figures D-1 to D-8).

<table>
<tr><td>Siemens AG
ANL A 44 - FP
Günther-Scharowsky-Str. 2

91058 Erlangen
Fed. Rep. of Germany</td><td>From
____
Name
____
Company/Dept.
____
Address
______________________</td></tr>
<tr><td>Tel.: (Nat. acc. code) 9131 / 7-32048
Fax: (Nat. acc. code) 9131 / 7-25670</td><td>Tel.:
Fax:</td></tr>
</table>

☐ ***ORDER***
Do you need one of the connecting cables described in the manual?

☐ ***QUOTATION***
Would you like us to submit a quotation for the fabrication/supply of one of these cables?

☐ ***INFORMATION***
Would you like to know more about the services SIEMENS ANL A 441 have to offer?

☞ If so, send a completed copy of this form to the above address.

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **DR 21x/23x printer without Busy signal, TTY** (Figure 5-1)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **DR 21x/23x printer with Busy signal, TTY** (Figure 5-2)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **DR 21x/23x printer RS-232C (V.24)** (Figure 5-3)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **CP 523, TTY** (Figure 6-2)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **CP 523, RS-232C (V.24)** (Figure D-1)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **CP 521 SI, TTY (Figure 8-1)**

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **CP 521 SI, RS-232C (V.24)** (Figure 6-3)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **CP 524, CP 544, CP 525, CPU 928B, terminal, RS-232C (V.24)** (Figures D-2 and 7-1)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to a **CP 524, CP 544, CP 525, CPU 928B, TTY** (Figure D-3)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to an **S5-90, S5-95, CPU 102, CPU 103, CPU 944, TTY** (Figures D-4 and 8-3)**,**

☐ Adapter cable No. 9AB4173- ____-____ Adapter cable between the CP 521 SI and the **BT 777 bus terminal (transceiver)** (Figure 9-1)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to the **PG 750 (PC 1650), TTY,** (Figure D-5)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to the **PC DF 10 interface card, TTY** (Figure D-6)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to the **PG 7xx (PC 1620), PC AT, RS-232C (V.24)/COM 2** (Figure D-7)

☐ Connecting cable No. 9AB4173- ____-____ Cable for connecting the CP 521 SI to the **PG 7xx, PC AT, RS-232C (V.24)/COM 1** (Figure D-8)

Length of connecting cable required: __________ m

Number of connecting cables required: __________ pcs

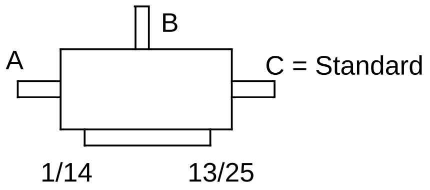
The connecting cables have standard connector shells (metallized plastics), with the cable outlet on the CP 521 SI side at the bottom. Please specify if you require other types of connector shell, shield leads or other lengths, or other cable outlet locations.

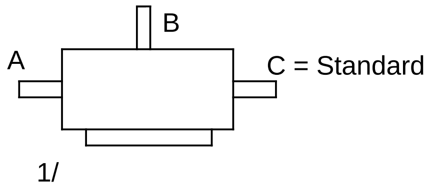
☐ Power/shield lead of other length (1 m is standard): _____ m

☐ Other cable outlets:

CP 521 SI: ☐ A ☐ B

Partner in link: ☐ A ☐ B





SERVICE

How long do you give us to fabricate your connecting cables you need?

☐ 48-hour service (express fabrication)

☐ 10 working days (normal fabrication)

How would you like to have the cables dispatched?

☐ Express

☐ Normal

Assembling the above connecting cables for you is by no means where our service ends.

We can give you any advice you might need on the phone:
There are plenty of problems that you can encounter when choosing connecting cables. No two cables are alike and one connector cannot possibly meet all requirements. So don't hesitate to get on the phone to us if you need advice, for we can offer you optimum support, whether it be in choosing special tailor-made cables for your specific requirements or simply assembling cables to your specifications ex-stock from our wide product range. You're always sure of getting just what you need.

You tell us - we do it:
The data you provide us with is taken as the basis for a quotation, which we can let you have in writing or over the phone, whichever you prefer. We can also let you have the necessary drawings etc. And should you ever need the same cable again, no problem: all the associated data on which our original quotation was based is stored, and you can place repeat orders simply by quoting the order code we give you.

Our service is a guarantee for quality
Quality assurance is written big at Siemens. It begins with the incoming goods inspection and accompanies our products right through the factory to final dispatch. And the result? The cable products we manufacture and assemble for you meet all the specified performance criteria, and have the necessary DIN ISO 9001 certification.

Connecting Cable from the CP 521 SI to a CP 523 (RS232C/V.24 Interface)

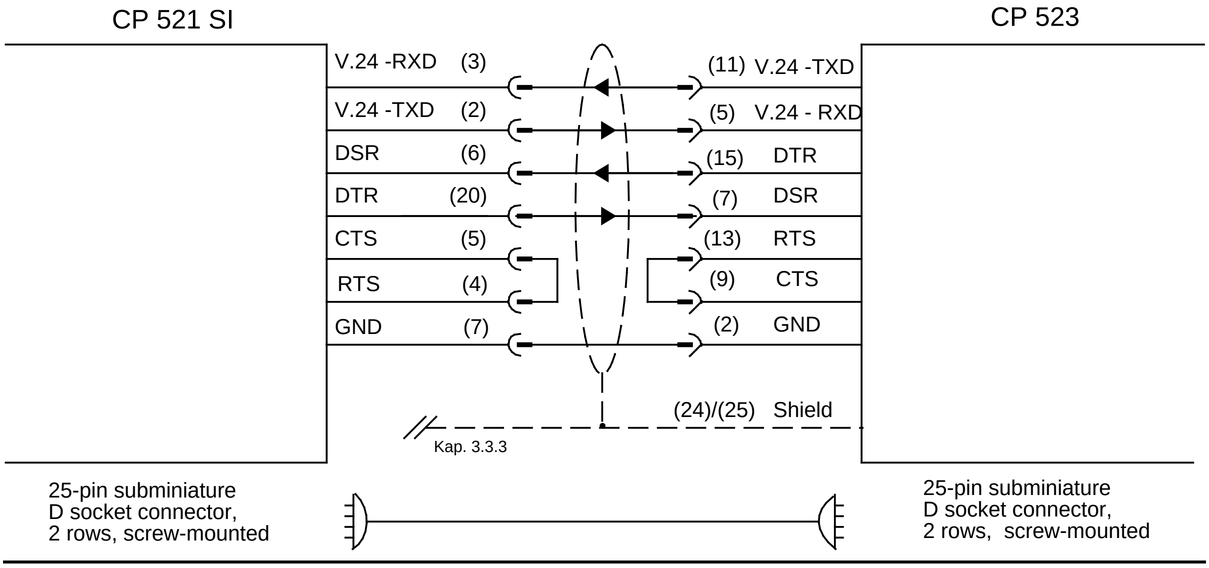


Figure D-1. Terminal Diagram CP 521 SI - CP 523 (RS-232C/V.24 Interface with Handshaking)

Connecting Cable from the CP 521 SI to a CP 524, CP 544, CP 525 or CPU 928B (RS232C/V.24 Interface)

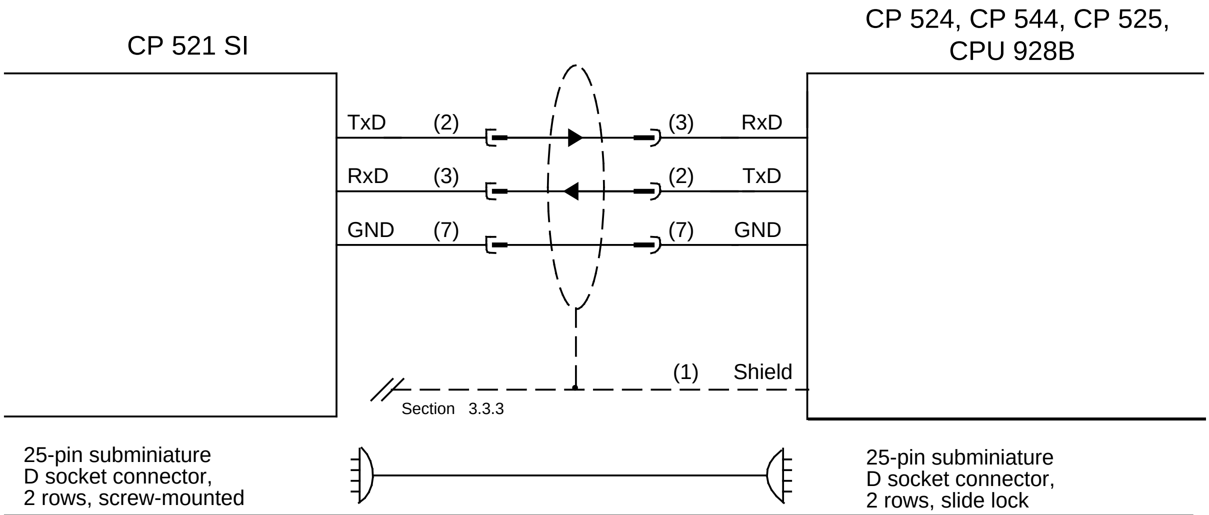


Figure D-2. Terminal Diagram CP 521 SI - CP 524, CP 544, CP 525 and CPU 928B (RS-232C/V.24 Interface without Handshaking)

Connecting Cable from the CP 521 SI to a CP 524, CP 544, CP 525, CPU 928B (TTY Interface)

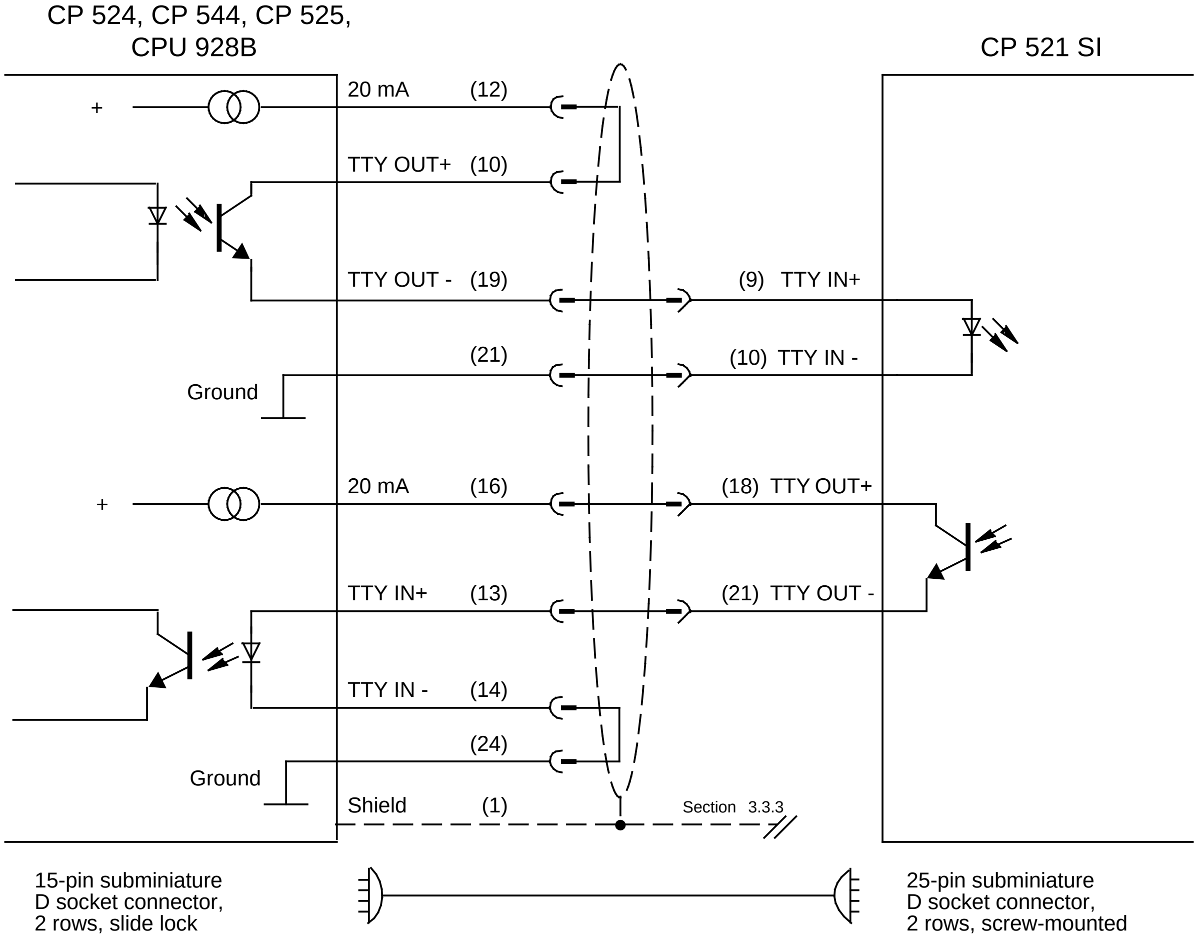


Figure D-3. Terminal Diagram CP 524, CP 544, CP 525 and CPU 928B (TTY Active) - CP 521 SI (TTY Passive)

Connecting Cable from the CP 521 SI to an S5-90, S5-95, CPU 102, CPU 103 or CPU 944 (TTY Interface)

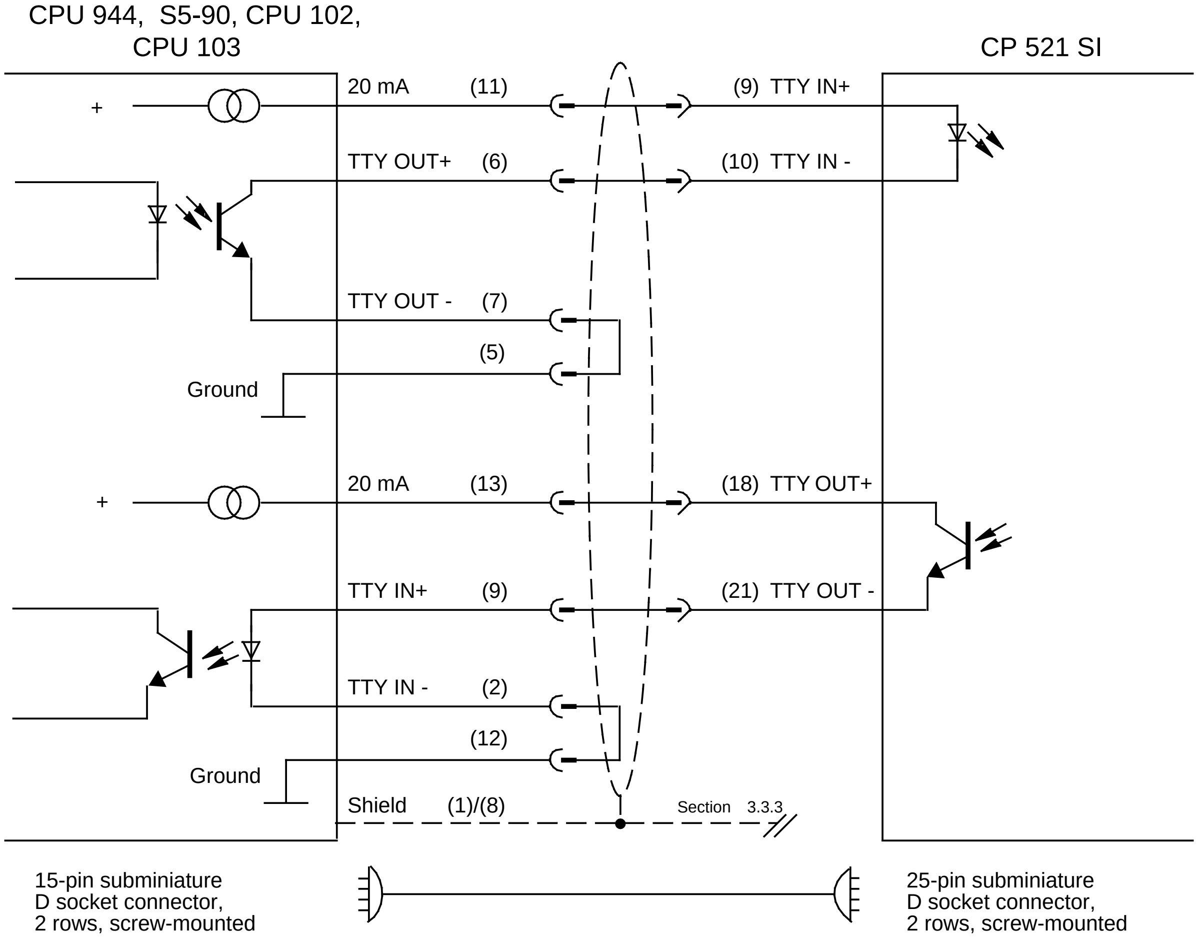


Figure D-4. Terminal Diagram S5-90, CPU 102, CPU 103 and CPU 944 (TTY Active) - CP 521 SI (TTY Passive)

Connecting Cable from the CP 521 SI (COM 1) to a PG 750 Programmer (PC 1620) (TTY Interface)

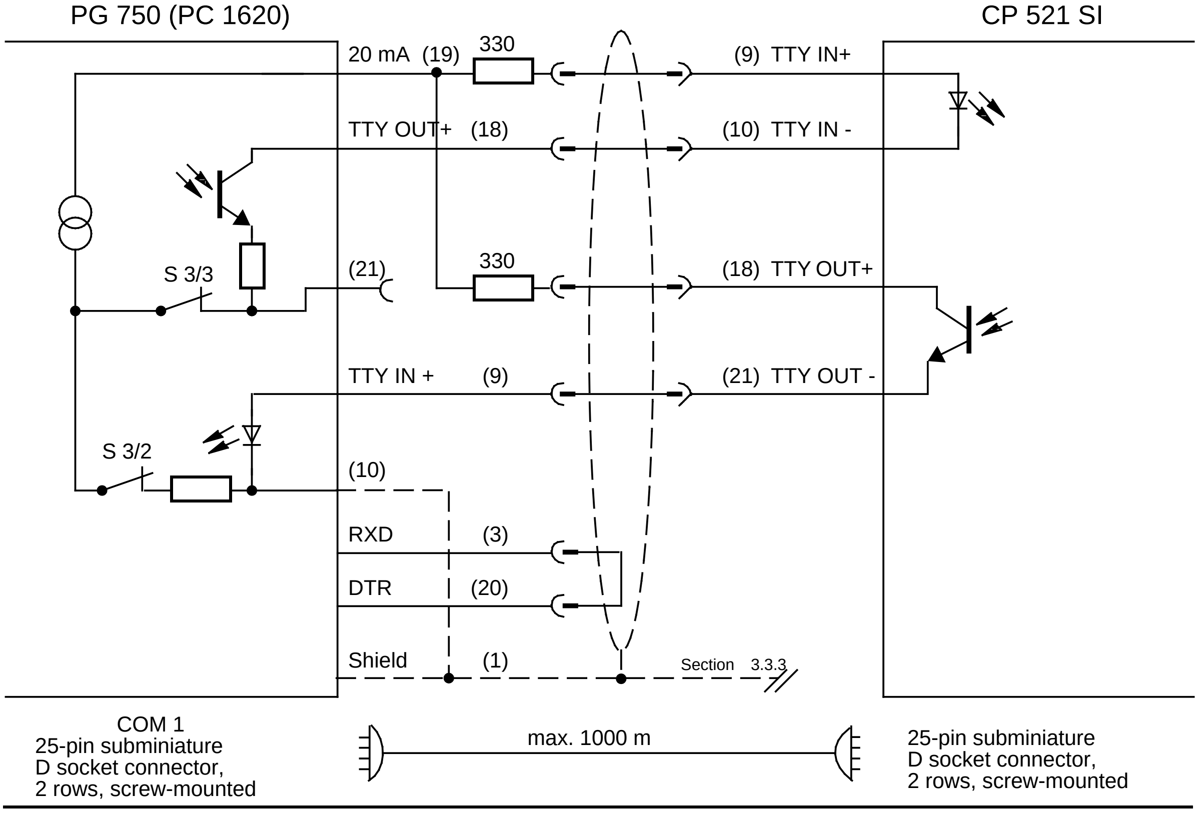


Figure D-5. Terminal Diagram PG 750 (PC 1620) (TTY Active, COM 1) - CP 521 SI (TTY Passive)

Connecting Cable from the CP 521 SI to a PC with DF10 Interface (TTY Interface)

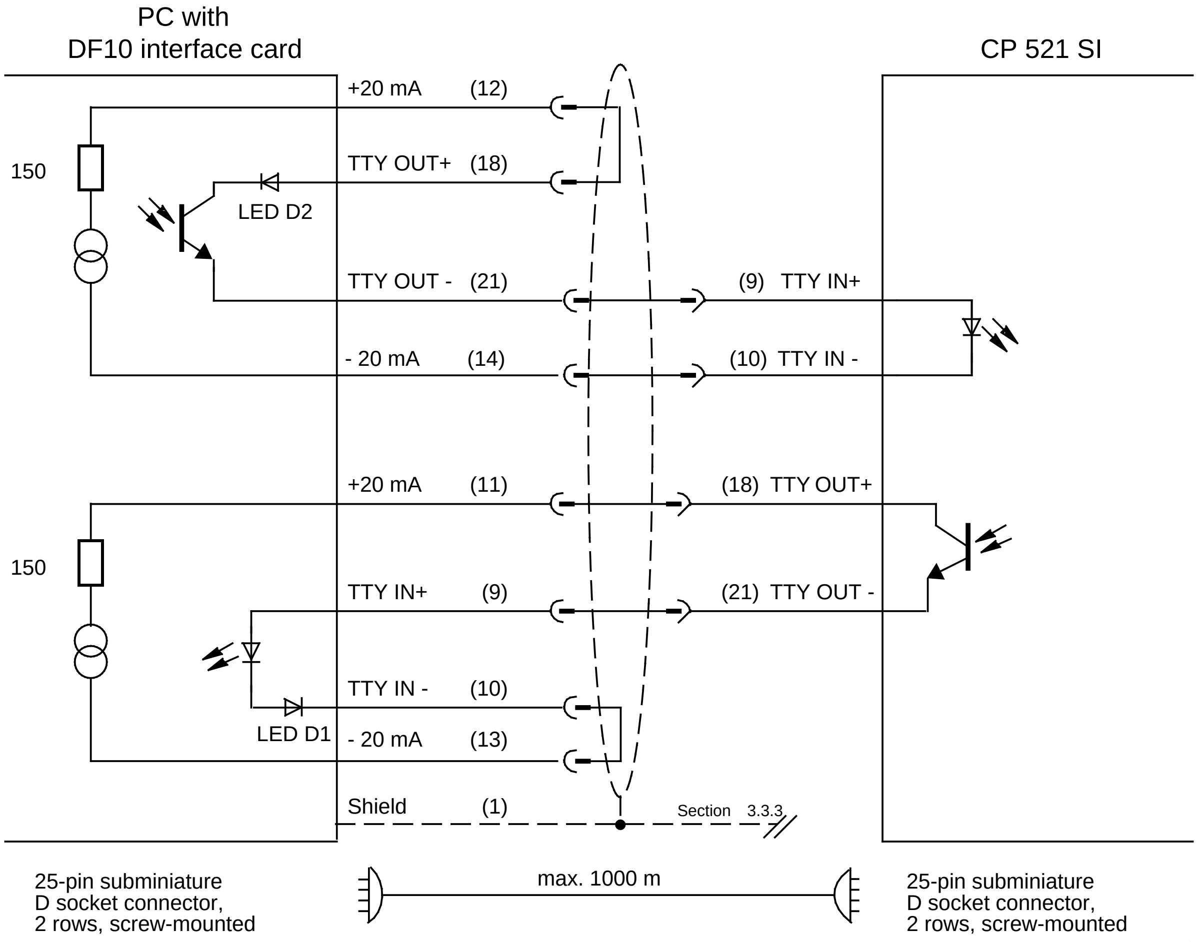


Figure D-6. Terminal Diagram for PC with DF10 Interface Card (TTY Active) - CP 521 SI (TTY Passive)

Connecting Cable from the CP 521 SI to COM 2 of a PG 7xx Programmer (PC 1620), PC AT (RS-232C/V.24 Interface)

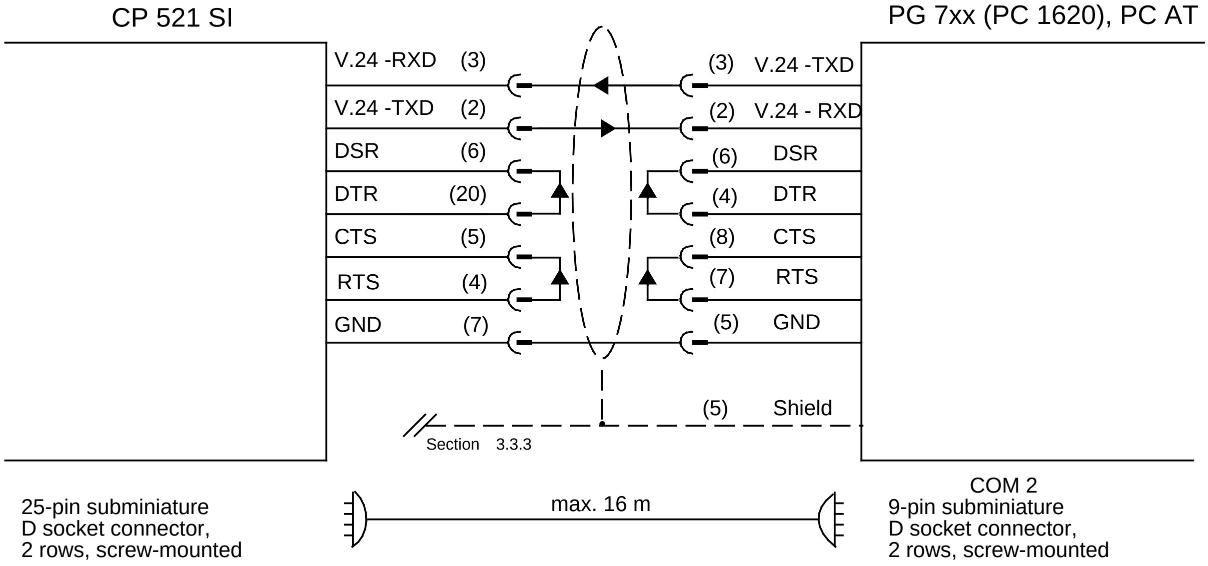


Figure D-7. Terminal Diagram CP 521 SI - COM 2 of a PG 7xx (PC 1620), PC AT (RS-232C/V.24

Connecting Cable from the CP 521 SI to COM 1 of a PG 7xx Programmer (PC 1620), PC AT (RS-232C/V.24 Interface)

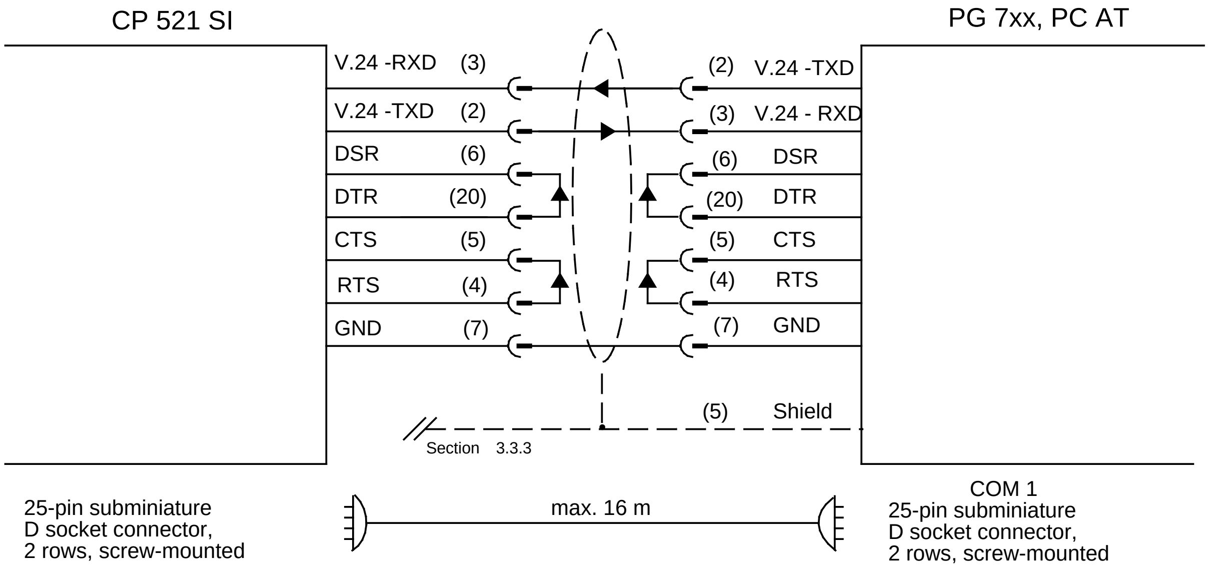


Figure D-8. Terminal Diagram CP 521 SI - COM 1 of a PG 7xx, PC AT (RS-232C/V.24 Interface)

Index

Index

A

Acknowledgement delay time
(timeout) 8-14
Addressing 2-7
ASCII driver
- CPU job 6-17
- CPU job request 6-39
- error message 6-39, 6-40
- final response 6-1, 6-30, 6-31, 6-38
- interpretive mode 6-9, 6-11
- parameter assignment data
- receiving message 6-32, 6-44
- sending message 6-22, 6-44
- setting interpretive mode 6-11, 6-12, 6-14
- setting transparent mode 6-6, 6-11, 6-14
- status information 6-39
- terminal diagram 6-8
- transmission modes 6-1
- transparent mode 6-1
ASCII mode
- setting interpretive mode 6-6
Assignment
- subminiature D connector 2-5, 5-4, 7-3
- subminiature D socket 8-2, 9-1
Approbations
- CSA 2-1
- UL

B

Backup
- clock data 2-6
Battery backup 2-6
- checking 4-16
BCC block check character
Bidirectional data traffic 1-2, 4-3
Block check character 8-6, 8-9, 8-15, 8-17, 8-21,
Block waiting time 8-14, 8-17, 8-21
BT 777 bus terminal 9-1, 9-13
Buildup attempt 8-15, 8-17, 8-21
BUSY signal 5-1, 5-2, 5-10
- parameter assignment 5-9
BWZ Block waiting time

C

Cable length 2-3
- permissible
Cabling 3-5
CBR coordination byte Receive
CBS coordination byte Send
CE marking 2-1
Character conversion table
- configuring 5-16
Character delay time 6-11, 6-12, 6-14, 6-19, 8-7, 8-8, 8-21
Character frame
- 10-bit 4-6
- 11-bit 4-7
Clock
- clock time/date error 4-12
- hardware fault 4-16, 4-17
- integral 2-6
- setting 4-12f
- test 4-17
Clock correction factor 4-15, 4-16
- configuring 5-16
Clock data 2-4
- backup
- current 4-12
- reading 4-14
Configuration data
- for entering message texts 5-12
- for message text printout 5-12, 5-13
- message text 5-22
Continuous form feed
- message text 5-15
Control character 5-23
- 3964(R) driver 8-6
Control parameter
- configuring 5-32
- transfer 5-24
Coordination byte Receive (CBR) 8-28, 8-33
Coordination byte Send (CBS) 8-7, 8-32
CPU job
- in 3964(R) driver mode 8-20, 8-23
- in ASCII driver mode 6-17
- in printer driver mode 5-36, 5-37, 6-39
- in terminal driver mode 7-22
- SINEC L1 driver 9-6
CPU job request 9-9
- SINEC L1 driver
CPU request 8-28
- 3964(R) driver
Current interface (TTY) 2-5

D

Data bit 4-6
Data cycle (DCyc) 4-2
Data format
- assigning the serial interface parameters 5-9, 5-10
- for variables 5-29 - 5-31
Data throughput
- maximum 1-4
Data traffic
- bidirectional 1-2, 4-3
- unidirectional 1-2, 4-3
Data transfer
- serial interface 4-3
Data transmission protocol 3964(R) 8-6
Date
- insert 5-24
- place holder 5-25
Date and time of day
- page format 5-13
Delete message buffer 5-37
Driver 3964(R)
- control character 8-6
- CPU job 8-20
- CPU job request 8-23
- CPU request 8-28
- error flag 8-32
- parameter assignment data 8-13, 8-14, 8-20
- protocol data 8-6
- protocol error 8-13
- receiving message frames 8-28
- sending message frames 8-23
- setting 8-14
- status information 8-32
- terminal diagram 8-3, 8-4
- terminating acknowledgement 8-26, 8-30

E

End-of-text character 5-12, 5-22, 6-11, 6-12, 6-15, 6-20, 6-31
Error flag
- 3964(R) driver 8-32
- in the coordination byte Send (CBS) 8-32
- in the coordination byte Receive (CBR) 8-33
Error message
- in ASCII driver mode 6-39, 6-40
- in printer driver mode 5-44
- in terminal driver mode 7-23
- status byte 4-9
Error numbers
- status byte 4-9
Equipotential bonding 3-6

F

Final response
- ASCII driver 6-30, 6-31, 6-38
Footer 5-13
- parameter assignment
- configuring 5-15
Format
- serial interface 4-6
Function character 5-12, 5-23

H

Handshake
- handshake OFF mode 6-2
- handshake ON mode 6-3
Hardware fault 4-16, 4-18, 4-19
Header
- configuring 5-15
- parameter assignment 5-13
HW handshake 6-11, 6-14, 6-18
- handshake OFF mode 6-14
- handshake ON mode 6-14

I

Initiation conflict 8-11
Input buffer 8-8
Installation 3-1
Integral clock 2-6
- real-time clock 2-6
- real-time hardware clock 1-4
Interface
- RS-232C (V.24) 2-5
- RS-232C (V.24) parameter assignment 5-9
- RS-232C (V.24) parameter assignment data 6-18
- serial 1-1, 2-5, 4-1, 4-3
- TTY 2-5
- TTY parameter assignment 5-9
- TTY parameter assignment data 6-18

J

Job buffer 4-1, 4-3, 5-35

L

LEDs 2-6
Lightning protection 3-6
Line feed
- output 5-37, 5-42

M

Master	9-1
Machine manufacturer	2-2
Memory assignment	6-18
- erasing data	2-2
- evaluation	4-17
Memory submodule	
- erasing data	2-4
- fault	4-17
- parameter assignment	5-8
- parameterize	5-6
- permissible	2-4
- plug	2-4
- remove	2-4
- storing data	2-4
Message buffer	4-1, 4-4, 5-35
- delete	5-42
- receiving	6-21
Message frame	
- receiving with evaluation of the message length	6-33
- receiving with the 3964(R) transmission protocol	8-28
- sending with end-of-text character	6-31
- sending	6-21
- sending with specified length	6-22
- sending with the 3964(R) transmission protocol	8-23
Message frame length	6-11, 6-12, 6-15, 6-19, 6-20, 6-25
- greater than 256 bytes	6-41
Message text	5-1, 5-3
- configuration data	5-12
- continuous form feed	5-15
- configuring	5-22
- insert	5-24
- maximum length	5-28
- number	5-22, 5-26, 5-35, 5-37
- outputting	5-38
- place holder	5-28, 6-26
- printing	5-35f, 5-43
Messages	
- printout	5-37

N

Number of frames	
- receive mailbox	6-23, 6-34

O

Output	
- line feed	5-37, 5-42
- message text	5-38
- page feed	5-37, 5-41
Overflow	
- receive mailbox	6-41

P

Page feed	
- output	5-37, 5-41
Page format	
- parameter assignment	5-13, 5-14
Page number	
- setting	5-37
Parameter assignment data	
- ASCII driver	6-9, 6-11
- 3964(R) driver	8-13, 8-14, 8-20
- printer driver	5-7, 5-9
- printing	5-20
- terminal driver	7-4, 7-5
Parameters	
- SINEC L1 driver	9-2
Parity	5-10
- bit	4-6, 8-6
- parameter assignment	5-9
Peripheral device	1-3, 2-5, 4-3
- connection	3-1
- status	4-11
PII process image of the inputs	
Pin assignments	
- subminiature D connector	6-7
PIQ process image of the outputs	
Place holder	5-22, 5-23
- entering	5-23
- for control parameters	5-24
- for date	5-24, 5-25
- for message text	5-24, 5-26
- for time of day	5-24, 5-25
- variable	5-24, 5-29
Point-to-point connection	9-13
Print driver	
- CPU job	5-36, 5-37
- error message	5-44
- parameter assignment data	5-7, 5-9
- setting	5-3
- status information	5-44
- terminal diagram	5-4, 5-5
- transmission methods	5-1
Printing	
- all message texts	5-43
- message text	5-35f

Printout
- messages 5-37
- all messages 5-37
Priority 8-6, 8-11, 8-15, 8-17, 8-21
Process image
- of the inputs (PII) 4-1
- of the outputs (PIQ) 4-1
Program cycle (PCyc) 4-2
Protocol 3964(R)
- control character 8-7
Protocol data
- 3964(R) driver 8-6
Protocol error 8-12

Q

QVZ acknowledgement delay time (timeout)

R

Real-time clock
- integral 2-6
Real-time hardware clock
- integral 1-4
Receive coordination byte (CBR) 9-9
Receive mailbox 4-1, 4-4, 8-28
- number of frames 6-34
- number of message frames 8-28
- overflow 6-41
Receiving
- fixed-length messages 6-33
- message frame 6-21
- message frame with 3964(R) transmission protocol 8-28
Restart characteristics 4-16
RS-232C (V.24) interface 2-5, 4-3
- handshake ON mode 6-3
- parameter assignment 5-9, 5-10

S

Send
- message frame with end-of-text character 6-31
Send attempt
- number 8-15, 8-17, 8-21
Send coordination byte (CBS) 8-23, 9-5
Send mailbox 4-1, 4-4
Sending
- message frame 6-21
- message frame with specified length 6-22
- with the 3964(R) transmission protocol 8-23
Separator 5-22
Serial interface 1-1, 2-3, 4-1
- data transfer 4-3
- format 4-6
Serial interface 1-1, 2-5, 4-1
- parameter assignment 5-9, 5-10
Setting
- clock 4-12f
- data 4-12f
Shielding 3-8
SINEC L1 driver
- CPU job request 9-6, 9-9
- parameters 9-2
- point-to-point connection 9-13
- receiving data 9-9
- sending data 9-5
- terminal diagram 9-2, 9-13
- terminating acknowledgement 9-8, 9-12
Slave 9-1
- number 9-2
Slot
- addresses 2-7
- permissible 3-1
Start bit 4-6, 8-6
Status
- peripheral device 4-11
Status byte
- error messages 4-8
- error numbers 4-9
Status information 4-8
- 3964(R) driver 8-32
- in ASCII driver mode 6-39
- in printer driver mode 5-44
- in terminal driver mode 7-23
Stop bit 4-6, 8-6
Subminiature D connector
- assignment 2-5, 5-4, 7-3
- pin assignment 6-7
Subminiature D socket
- assignment 8-2, 9-1
System environment 1-1

T

Technical Specifications 2-3
Terminal diagram
- ASCII driver 6-8
- 3964(R) driver 8-3, 8-4
- printer driver 5-4, 5-5
- SINEC L1 driver 9-2
- SINEC L1 slave 9-13
- terminal driver 7-3

Terminal driver 7-22
- CPU job request
- error messages 7-23
- memory submodule 7-1

Terminal driver
- parameter assignment data 7-4, 7-5
- setting 7-2, 7-7
- status information 7-23
- terminal diagram 7-3
- transmission methods 7-1

Terminating acknowledgement
- 3964(R) driver 8-26, 8-30
- SINEC L1 driver 9-8, 9-12

Time of day
- insert 5-24
- place holder 5-25

Timeout 8-7, 8-17, 8-21
Transfer memory 4-1
- access 4-4
- contents 4-8

Transmission
- with block check character 8-15, 8-21
- without block check character 8-15, 8-21

Transmission protocol
- 3964 8-7, 8-9

Transmission rate
- parameter assignment 5-9

TTY current interface 4-3
TTY interface 2-3, 4-3
- parameter assignment 5-9, 5-10

U

Unidirectional data traffic 1-2, 4-3

V

Variable
- data format 5-29 - 5-31
- insert 5-24
- place holder 5-29

Voltage interface (RS-232C (V.24)) 2-5, 4-3

W

Wait times 5-1, 5-2
- parameter assignment 5-11

X

XON/XOFF protocol 5-1, 5-2, 6-2, 6-11, 6-14, 6-19
- parameter assignment 5-11

Z

Zero modem mode 8-4
ZVZ Character delay time

Siemens AG
AUT E 148
Postfach 1963

D-92209 Amberg
Federal Republic of Germany

From:

Your Name: ____

Your Title: ____

Company Name: ____

Street: ____

City, Zip Code: ____

Country: ____

Phone: ____

Please check any industry that applies to you:

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

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.

Title of Your Manual: ______________________________

Order No. of Your Manual: ______________________________

Edition: ______________________________

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

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

Additional comments:

____

____

____

____

____

____

____

____

____

____

____