SIMATIC S5

CP 523
Serial I/O Module

Manual

Order No. 6ES5 998-0DD21

EWA 4NEB 811 6044-02a

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

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

WARNING

Hazardous voltage.

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

Restrict use to qualified personnel.
See safety instructions.

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

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

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

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

DANGER

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

WARNING

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

CAUTION

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

Second Printing, march 1991
Printed in the Federal Republic of Germany

Preface

This manual is a revised, updated edition of the original CP 523 manual. The revision was necessary because the CP 523 has been equipped with an additional communication control procedure, thus upgrading the number of transmission modes in ”Communications mode” from two to three:

- Transparent mode
- Interpretive mode
- 3964(R) mode (***new***)

The characteristics of each mode are discussed in Section 2.3.2.

The CP 523 is a powerful I/O module which can be used in the S5-115U, S5-135U, S5-150U, S5-155U and S5-115F programmable controllers. In order to make best use of the serial I/O module, you require comprehensive information. For this reason, the CP 523 serial I/O module has its own manual dealing only with those facts and examples that are relevant to the module. Demands for a higher quality of technical documentation have also been met, which means

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

The aim is that both users with little previous experience and SIMATIC S5 experts should find all the information that they require to work with the CP 523.

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

Introduction

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

Description of Contents

This manual is a comprehensive description of the CP 523. 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 S5-115U, S5-135U, S5-150U, S5-155U and S5-115F.
  - The "Technical Description" contains general information on the principle of operation of the module, technical specifications, details of the serial interface and input/output (I/O) modules and a list of accessories.

- Installation and operation
  - The chapter on "Hardware Installation" tells you which programmable controllers and expansion units the module can be used with, how it is assembled and how to connect it to a printer or I/O module (connector pin assignments and terminal diagrams).
  - "Notes on Operation" describes the restart characteristics and actual operation. This chapter also contains the error messages that can occur during startup.

- Addressing
  This chapter demonstrates how to address the module and describes the function of the transfer memory.

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

- Appendix A: Summary
  All the information you require to operate the CP 523 has been gathered together here in list form. This has been designed for those users who have read the manual and wish to find some brief item of information quickly.

Conventions

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

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

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

- A number of abbreviations have been used.
  Example: Central processing unit (CPU)
- Footnotes are identified by superscripts consisting of a small digit (e.g. ”^{1}”), or ”*”. The actual footnote is generally at the bottom left of the page or below the relevant table or figure.
- Cross-references are shown as follows:
  ”(7.3.2)" refers to subsection 7.3.2.
  No references are made to individual pages.
- All dimensions in drawings etc. are given in millimetres/inches (mm/in.).
- Information of particular importance is framed in
  grey-bordered rectangles.
- All program examples have been generated in statement list form and always refer to the same slot.
- All data in connection with the programmer refers to the German version of the PG 685. The relevant user manual contains a detailed description of programming procedures with this programmer.
- The ”Reader's Note” in Section 7 provides references to various subsections, thus making it easier to find the subsections relevant to your application or specific requirements.

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

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

Courses

Siemens provide SIMATIC S5 users with extensive opportunities for training.

For more information, please contact your Siemens representatives.

Reference Literature

This manual is a comprehensive description of the CP 523 serial I/O module. Topics not specific to the CP 523, however, are only briefly dealt with. You will find more detailed information in the following literature:

- **Programmable Controllers**
  Volume 1: Logic and Sequencing Control; From the Task to the Program.

  Günter Wellenreuther, Dieter Zastrov
  Braunschweig 1987

  Contents:
  - Method of operation of a programmable controller
  - Theory of control technology using the STEP 5 programming language for the SIMATIC S5 programmable controllers.

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

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

  Hans Berger
  2nd Edition, Berlin and Munich: Siemens AG, 1989

  Contents:
  - STEP 5 programming language
  - Program scanning
  - Integral program blocks
  - I/O interfaces

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

Conventions

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

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

Figures

1 System Overview

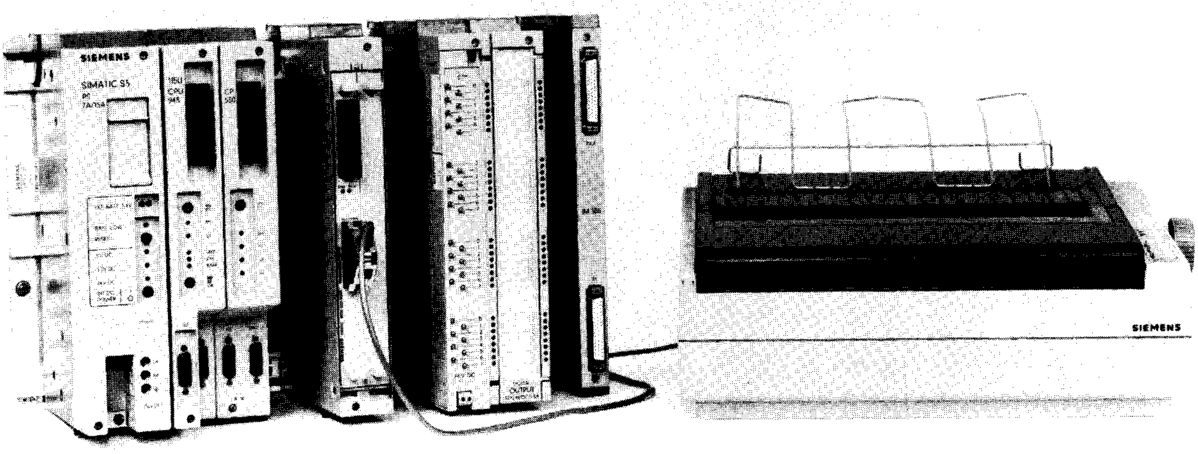
Intelligent input/output modules (1/0s) expand the application area of 51 MATIC S5 programmable controllers. They are technology-oriented and offload the central processing unit by preprocessing input signals.

The CP 523 serial I/O module is an intelligent I/O module, which can be used in the S5-1 15U, S5-135U, S5-150U, S5-155U and S5-1 15F programmable controllers. It can be operated without special COM software and offers applications which until now could only be implemented with expensive and functional y complex modules.

The CP 523 can be used in "Print mode" and "Communications mode". It has **a** built-in real-time clock, which can be backed up by the battery of the power supply unit. The clock data can be read by the CPU and used in the user program for date-dependent and time-dependent tasks.

Print mode
Message texts can be printed out in this mode. This allows you to list process states and process faults.

- Printers with TTY or RS-232-C (V.24) interfaces can be connected
- The printer interface can be configured (baud rate, BUSY signal, etc.)
- The format of the page to be printed can be configured (headers, footers, margins, etc.)
- Configuration of up to 4095 different message texts in data blocks on a memory submodule
- You can provide for the following when configuring message texts:
  - Insertion of the date or time of day in the printout
  - Insertion of current variables in the printout (pressure, temperature, etc.)
  - Transfer of printer control parameters (double-width type on/off, boldface type, etc.)



EWA 4036/1

Figure 1-1. S5-1 15U Programmable Controller with CP 523 and Printer

Communications Mode

In "Communications mode", the CP 523 can communicate over the I/O interface with an I/O unit (terminal, CP 523, CPU 944, etc.) equipped with a V.24 (RS 232C) or TTY interface. This mode enables the transfer of data frames between the CPU and an I/O device connected to the CP 523.

The CP 523 provides a total of three communication control procedures, i.e. two protocol-free procedures and one with a standardized protocol:

. Protocol-free data transmission
- Transparent mode, i.e. data transmission without receiver-side end identifier sequence
- Interpretive mode, i.e. data transmission with receiver-side end identifier sequence.

. Data transmission with protocol
-3964 (R) mode

Your application program determines the modes you want to use.

You can also evaluate the module's real-time clock in your program for date and time-dependent tasks.

The FE 200 "SEND" and FB 201 "RECEIVE" function blocks allow user-friendly bidirectional transfer of message frames with a length of up to 256 bytes. A thorough description, with examples, is given of how to use the function blocks.

If you decide to forego the convenience of the FB 200 and FB 201 function blocks in favour of higher data transmission speeds, you can communicate with the CP 523 direct from the user program. This method of data exchange is also described in detail with examples.

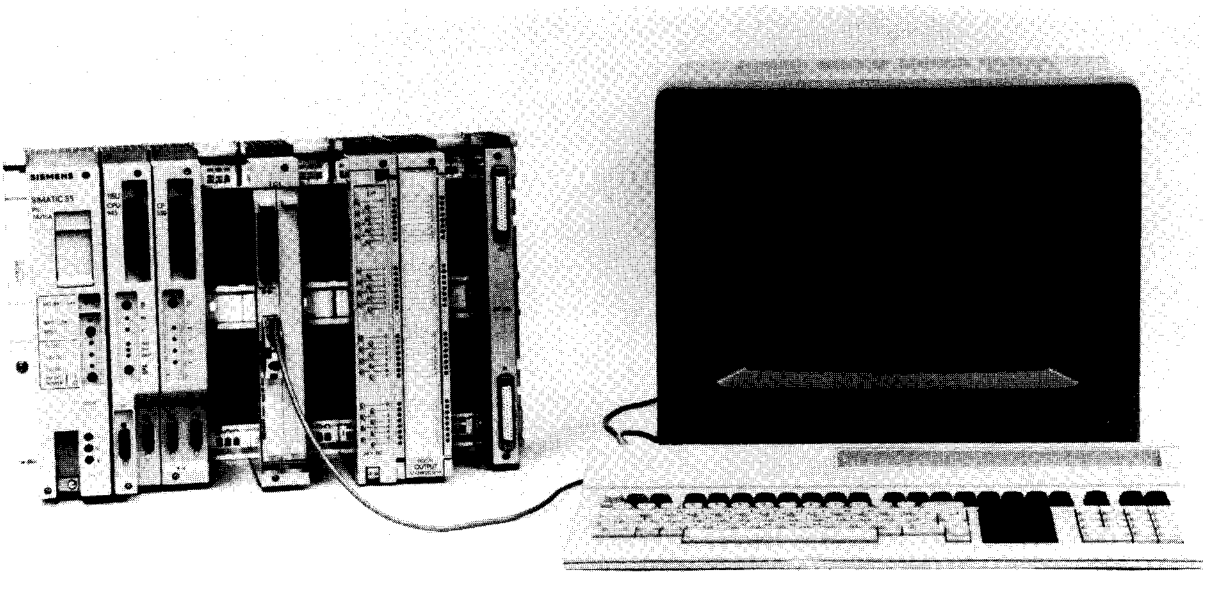


Figure 1-2. S5-115U Programmable Controller with CP 523 and Terminal

Figures

Tables

2 Technical Description

2.1 Principle of the CP 523

The CP 523 handles data transfer with a peripheral device autonomously. Data transfer between the CPU and the CP 523 must always be started by the CPU by sending a job request to the CP 523. Typical jobs would be, for example, ”Print message text number 20” in Print mode or ”Receive a message frame from a peripheral device” in Communications mode.

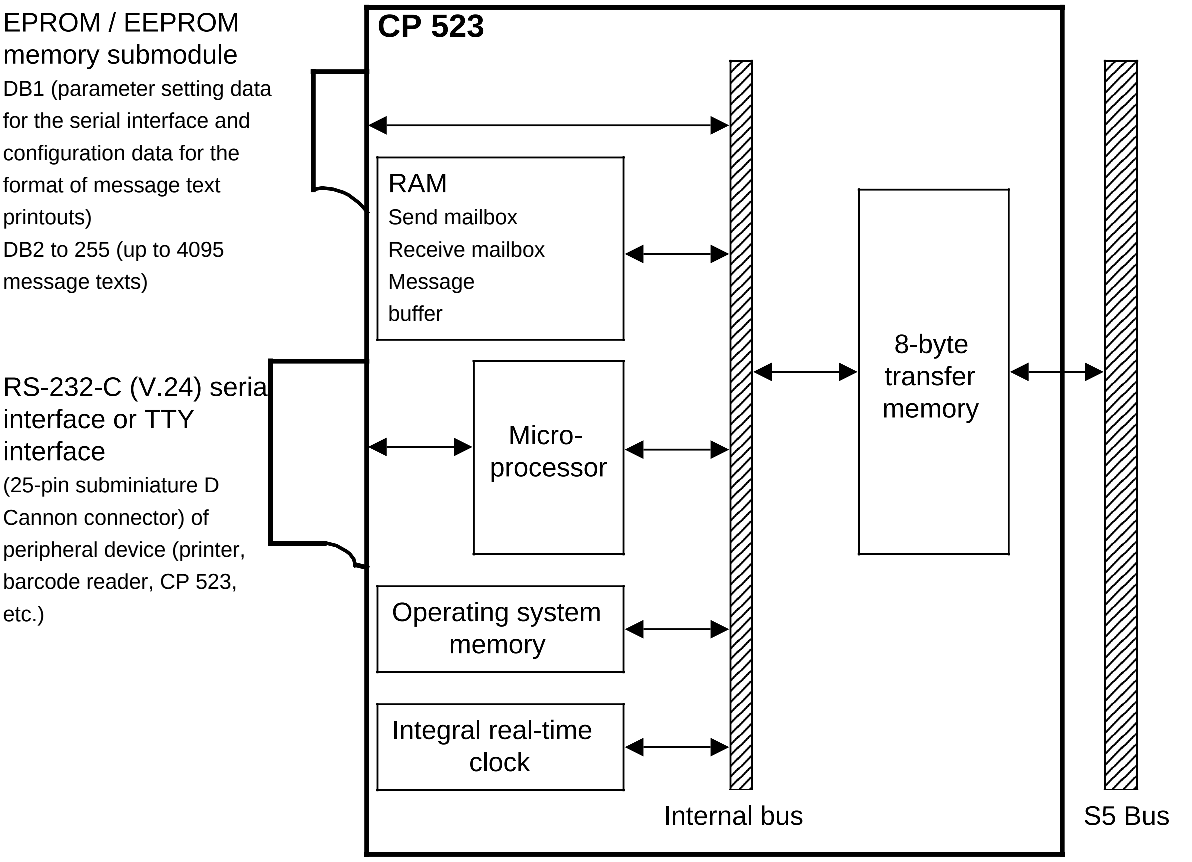

EPROM / EEPROM memory submodule
DB1 (parameter setting data for the serial interface and configuration data for the format of message text printouts)
DB2 to 255 (up to 4095 message texts)
RS-232-C (V.24) seria interface or TTY interface
(25-pin subminiature D Cannon connector) of peripheral device (printer, barcode reader, CP 523, etc.)
CP 523
RAM
Send mailbox
Receive mailbox
Message buffer
Micro-processor
Operating system memory
Integral real-time clock
8-byte transfer memory
Internal bus
S5 Bus

Figure 2-1. Schematic Representation of the CP 523

Print mode
The "Print message text XY" job request causes the message text stored in the memory submodule under the number XY to be transferred to the printer via the serial interface. Current values of variables, time of day and date can be entered in the message text during this process. You can configure up to 4095 different message texts and store them in the memory submodule. You can also include control commands for the printer in the message text. These are then executed on printout (double-width type on/off).

Communications mode

In Communications mode, message frames of up to 256 bytes in length can be transferred between the CPU and a peripheral device connected to the CP 523.

When it receives a "SEND" request from the CPU, the CP 523 transfers the data from the transfer memory to a Send mailbox. The CP 523 then transfers the data autonomously from the Send mailbox to the peripheral device.

Data received from the peripheral device is first stored by the CP in a Receive mailbox. On receiving a "RECEIVE" request from the CPU, the data is forwarded to the transfer where it can be read out by the CPU.

S5 bus and transfer memory

Data is transferred between the CPU and CP 523 via the S5 bus. Data is stored in an eight-byte transfer memory in the CP 523. Addresses for the transfer memory are derived from the initial address of the module and an offset of 0 to 7.

Only when word 0 of the transfer memory has been overwritten with T PW <Initial address of the module> does the CP 523 read the data from the transfer memory and update the transfer memory with current values. These current values can then be read by the user program with L PW statements.

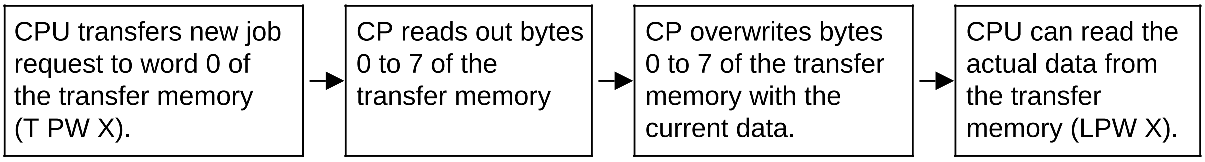

CPU transfers new job request to word 0 of the transfer memory (T PW X).
CP reads out bytes 0 to 7 of the transfer memory
CP overwrites bytes 0 to 7 of the transfer memory with the current data.
CPU can read the actual data from the transfer memory (LPW X).

Figure 2-2. Using the Transfer Memory

You can transfer additional information to the CP in words 2, 4 and 6 before the request is submitted. This depends on the job in question. See the job descriptions for the precise explanation.

Serial interface

The CP 523 has an RS-232-C (V.24) interface and a TTY interface.
The following are examples of peripheral devices that you can connect to the serial interface of the CP 523:

- Printer with passive or active TTY interface
- Printer with RS-232-C (V.24) interface
- Barcode reader
- Keyboard
- Terminal
- MODEM
- Another CP 523
- CPU 944 (ASCII interface)
- CP 521
- PC

EPROM/EEPROM

You can use EPROM and EEPROM memory submodules of up to 32K bytes. The EPROM/EEPROM is configured with the programmer.

- Print mode

In Print mode, the parameter setting data for the serial interface and for the format of the printout is defined in data block 1 (DB 1) on the memory submodule. You can configure up to 4095 different message texts in DBs 2 to 255.

Note:
The CP 523 can be operated in Print mode only if a memory submodule containing at least one message text is plugged in.

- Communications mode

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

Note:
In Communications mode, the CP 523 can also be operated without a memory submodule, in which case the parameter setting data for the serial interface is then transferred from the user program to the CP.

Integral real-time hardware clock

The CP 523 has an integral real-time hardware clock. The clock can be set and read either with a programmer or in the user program.
The clock is optimized for an ambient temperature of 25 °C. A correction factor can be programmed to compensate for this temperature dependency. This correction factor can be transferred to the CP from the user program or it can be programmed in DB 1 on the memory submodule.

Addressing

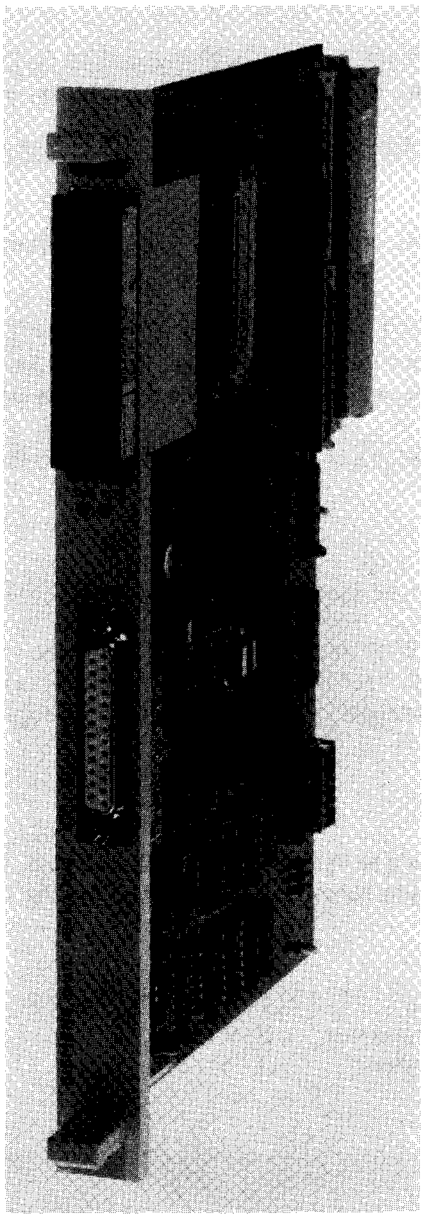
The CP 523 is addressed as an eight-byte input/output module in I/O areas P, O, IM3 and IM4. You can set the desired start address of the module in eight-byte steps via DIP switches on the module (5).

Note:
Byte numbers 0 to 7 specified in this manual always refer to the start address set:

Example: You have set start address 128 with the DIP switches.
Byte 0 has address 128+0=128
Byte 7 has address 128+7=135

2.2 Technical Specifications

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



EWA 4036/2

Galvanic isolation		TTY signals are floating
Memory submodule		EPROM/E EPROM
Serial Interface		V.24 (RS 232 C)/TTY
Transmission mode:		Asynchronous 10-bit character frame 11 -bit character frame
Transmission rate		110 to 9600 baud
Permissible cable length		
TTY active		10 m/33 ft.
- TTY passive		1000 m/3280 ft Voltage drop Receiver typ. **15V** **Sender** typ. **0.9 v** at 20 mA
- RS-232-C (V.24)		15 m149 ft.
Battery backup time		Dependent on the battery backup in the central controller: at least 1 year
Degree of protection		IP **20**
Permissible ambient temperature		
- vertical		0 **to** 60”C
- horizontal		**0** to 40°C
Relative humidity		**1**5% to 95%
Current consumption from + 5 V		typ. 130 mA
Power losses of the module	typ	1.2 **w**
Weight	approx.	**300 g**
LEDs		
2 green LEDs	RCV	CP 523 receiving data
	SEN	CP 523 sending data

2.3 Serial Interface CP - Peripheral Device

The CP 523 is equipped with an RS-232-C (V.24) interface and a TTY interface. You can set the serial interface parameters in two ways:

- In the user program with job number 90_H "Set interface parameters" parameter block (7.3.2).
  The CP 523 copies the parameter setting data from the submodule into a RAM area on the CP during startup. If the CPU transfers new parameter setting data to the CP during operation, the CP continues with the new data.
- In data block 1, parameter block 0, on the user submodule (Print mode 6.3.1, Communications mode 7.3.1).

Data is transferred between the CP and the peripheral device in a 10-bit or 11-bit character frame. Three data formats are available for each character frame.

10-Bit-character frame

- 1 start bit, 7 data bits, 2 stop bits (data format 3 in parameter block 0)
- 1 start bit, 7 data bits, 1 parity bit, 1 stop bit (data format 4 in parameter block 0)
- 1 start bit, 8 data bits, 1 stop bit (data format 5 in parameter block 0)

11-Bit-character frame

- 1 start bit, 7 data bits, 1 parity bit, 2 stop bits (data format 0 in parameter block 0)
- 1 start bit, 8 data bits, 1 parity bit, 1 stop bit (data format 1 in parameter block 0)
- 1 start bit, 8 data bits, 2 stop bits (data format 2 in parameter block 0)

1st example:

Character frame: 10-bit
Data format: 7 data bits (1 start bit, 7 data bits, 1 parity bit, 1 stop bit)

Bit no.:

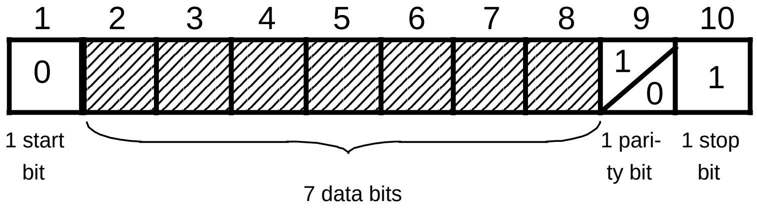


Figure 2-3. 10-Bit Character Frame with 7 Data Bits, 1 Parity Bit and 1 Stop Bit

2nd example:

Character frame: 11-bit
Data format: 8 data bits (1 start bit, 8 data bits, 2 stop bits)

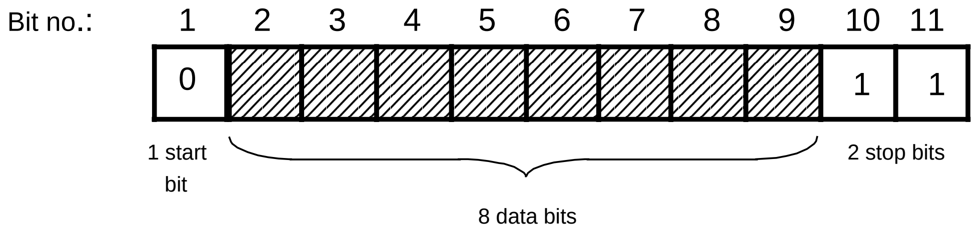


Figure 2-4. 11-Bit Character Frame with 8 Data Bits and 2 Stop Bits

Note:

When using the RS-232-C (V.24) interface, the readiness of the sending or receiving device can only be recognized via the control signals. You are therefore recommended to set Handshaking ON mode.

2.3.1 Print Mode

You have a choice of three methods of transmission in Print mode. All three are possible with both the RS-232-C (V.24) and TTY interfaces.

Table 2-1. Methods of Transmission in Print Mode

Method of Transmission	Line Used TTY/ RS-232-C (V.24)	Required Parameters	Parameter Block	Remarks
XON/XOFF protocol	RXD	XON characters XOFF characters	2	Priority over BUSY signal and print without protocol
BUSY signal	RXD/DSR	-	0	Priority over print without protocol
Without protocol	Irrelevant	Wait after - CR (carriage return) - LF (line feed) - FF (form feed)	1	Line break not detectable

2.3.2 Communications Mode

The possible methods of transmission depend on the following:

- Type of interface (TTY or RS-232-C (V.24)
- Whether handshaking has been configured when the RS-232-C (V.24) interface is used (Handshaking OFF or ON).
- Type of Communications mode.
  - Transparent (without send-side end identifier sequence)
  - Interpretive (with send-side end identifier sequence)
  - 3964 (R)

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

The CPU evaluates only the RXD circuit. If the CP receives data from the peripheral device, the latter must maintain logic "1"[*] on the RXD line, otherwise the CP 523 will send the following error flags to the CPU: in byte 0: "Permanent line break" (XD_H) and in byte 1: "Peripheral device not ready" ($1X_H$).

- Transparent mode
  No XON/XOFF protocol is possible in Transparent mode. Message frames received by a peripheral device must be of fixed length.

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

- 3964(R) mode
  This communication control procedure makes it possible to connect peers which also use the 3964(R) line procedure to the CP 523.

  3964(R) mode enables comparatively reliable data transmission, as the receiver must signal its readiness to receive data (connection buildup) and acknowledge once it has done so. Reliability and data integrity are enhanced in 3964(R) mode by an additional block check character.

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

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

The RS-232-C (V.24) interface of the CP 523 can handle the following signals in handshake ON mode:

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

Control signal	State	Meaning
Outputs		
TXD		Send data CP holds send line at logic 1 when not transmitting (V -3 V).
DTR	 ON OFF	**D**ata **T**erminal **R**eady CP 523 switched on; ready to receive CP 523 not switched on; not ready to receive
RTS	 ON OFF	**R**equest **t**o **s**end CP 523 ready to send CP 523 not sending
Inputs		
RXD		Receive data Receive line must be held at logic 1 by the peripheral device (V -3 V).
DSR	 ON OFF	**D**ata **s**et **r**eady Peripheral device switched on; ready to receive Peripheral device not switched on; not ready to receive
CTS	 ON	**C**lear **t**o **s**end Peripheral device can receive characters from the CP 523 The CP expects this as reply to RTS = "ON"
	OFF	Peripheral device cannot receive characters from the CP 523

Note:

An XON/XOFF protocol is not possible if you evaluate the control signals (handshake ON).
In 3964(R) mode, handshaking is not possible (hardware signals are neither evaluated nor affected).

Data transfer between the CP and peripheral device takes place as follows:
The CP 523 sets the "DTR" output after startup. This indicates that the CP is operable and ready to receive.

Example: The CP wants to send data

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

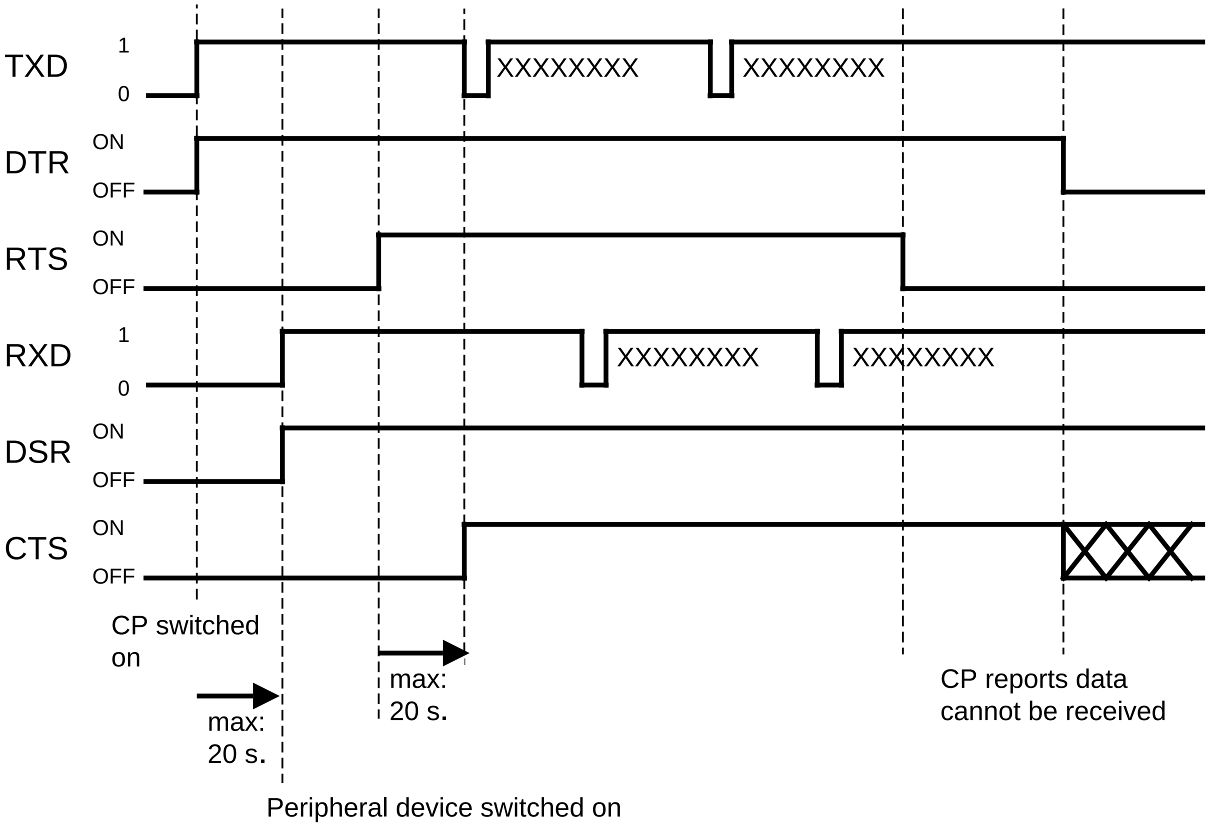


Figure 2-5. Timing Diagram for Data Transfer between the CP and the Peripheral Device

Example: Peripheral device wants to send data

The peripheral device only checks the DTR line
If ON: Data may be sent (CP ready to receive)
If OFF: Data transmission must be interrupted (CP not ready to receive)

2.4 Memory Submodule

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

- Message texts in DB 2 to 255
  These DBs are independent of the DBs stored in the CPU. The DBs in the memory submodule and the DBs in the CPU may have the same numbers.
- Printer interface parameters and configuration data for the message texts in DB 1.

You can use two types of user submodule:

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

Table 2-3. Overview of Permissible User Submodules

Submodule Type	Order No.	Capacity
EPROM	6ES5 375 - 0LA15	8 Kbyte
	6ES5 375 - 0LA21	16 Kbytes
	6ES5 375 - 0LA41	32 Kbytes
EEPROM	6ES5 375 - 0LC11	2 Kbytes
	6ES5 375 - 0LC31	8 Kbytes
	6ES5 375 - 0LC41	16 Kbytes

2.4.1 Memory Submodule Evaluation

Several error flags can be stored by the CP in byte 0 in conjunction with the memory submodule.

Table 2-4. Error Flags in Conjunction with the Memory Submodule

Contents of byte 0	Error	Remedy
$X1_H$	Memory submodule defective	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 user submodule plugged in?

X= Value for other half-byte irrelevant

2.5 Integral Hardware Clock

The CP incorporates a hardware clock which is backed up by the battery of the PLC power supply module.

Using the clock data
You can use the clock data in two ways:

- The current clock data can be inserted in a message text. You can insert spaceholders for date and time of day for this purpose when configuring the message texts (6.4).
- The current clock data can be read out by the CPU and used in the user program (6.6 or 7.4.2).

Battery backup of the clock data
The clock is backed up by the battery of the PLC power supply module.
The clock starts with the default values Sunday 1.1.90 12:00:00 on power-up.
The power supply module battery should only be replaced in the POWER ON state, as otherwise the clock data is lost and the clock must be reset.

Default setting
The clock is factory-set to Sunday 1.1.90 12:00:00

Setting the time and date
You can set the clock:

- In the user program (see Section 6.7.2 for details)
- With the "FORCE VAR" programmer function (refer to the example ”Writing to word 0 of the transfer buffer with the CPU 944 using ”FORCE VAR” in Section 5.2).

Output format of the clock data in the message text
You can configure how the current clock data is to be output to the printer (6.3.3).
The default is:

- For the date: <Day> . <Month> . <Year>
- For the time: <Hours> : <Minutes> : <Seconds>

Accuracy of the integral real-time clock
The accuracy of the clock is $t_{\mathbf{g}}$=±2s/day.

Temperature dependency of the clock
The temperature dependency $T_{\mathbf{D}}$ of the clock is $T_{\mathbf{D}}=3.5\ (T_{\mathbf{A}}-15)^2$ ms/day at an ambient temperature of $T_{\mathbf{A}}$.

Correction value
You can configure a correction value to enhance the accuracy of the clock (Print mode   6.3.6, Communications mode   7.3.1 and 7.3.2). The correction value is given in s/month. A month is defined as 30 days.

Example: Determining the correction value

You have observed that the clock loses 12 s in 4 days. This would be 90 s in 30 days. The correction value is then 90 s/month.

2.6 Accessories List and Ordering Data

Memory submodules

Memory submodule (EPROM)		8 Kbytes	6ES5 375-0LA15
”	”	16 Kbytes	6ES5 375-0LA21
”	”	32 Kbytes	6ES5 375-0LA41
Memory submodule (EEPROM)			
”	”	8 Kbytes	6ES5 375-0LC31
”	”	16 Kbytes	6ES5 375-0LC41

Printer connecting cable

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

PT 88 RS-232-C (V.24)	3.2 m	6ES5 735-2BD20
PT 88 TTY	3.2 m	6ES5 735-1BD20

Printers

See S5-115U Programmable Controller Catalog	ST 52.3
and/or	
S5-135U and S5-150U Programmable Controller Catalog	ST 54.1

Figures

3 Hardware Installation

3.1 Mechanical Assembly

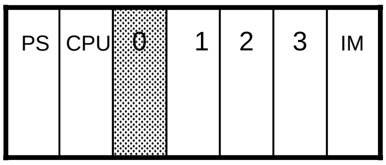
3.1.1 Compatible Programmable Controllers and Expansion Units

The CP 523 can be used in the S5-115U, S5-115F (in both cases with adapter casings), S5-135U, S5-150U and S5-155U programmable controllers - central controllers (CCs) or expansion units (EUs) - without fans.
The permissible module locations are as follows:

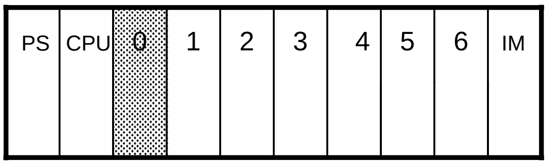
Use in the S5-115U

- Central controller

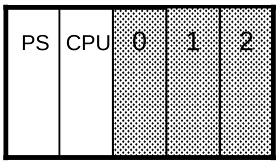
CR 700-0LA subrack



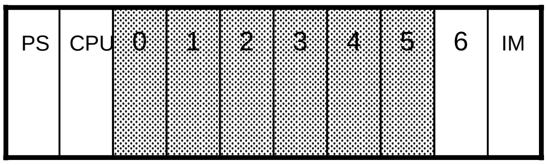
CR 700-1 subrack



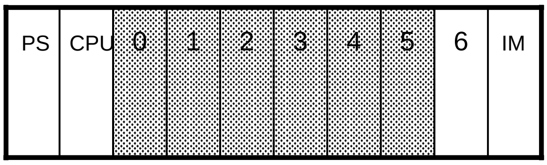
CR700-0LB subrack



CR 700-2 subrack

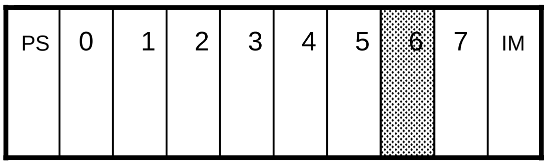


CR 700-3 subrack

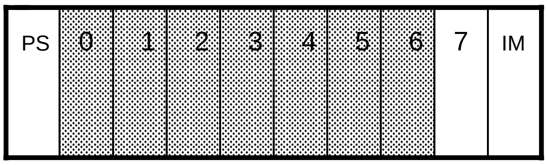


- Expansion unit

ER 701-2 subrack



ER 701-3 subrack



▨ Possible locations

Note:
The 304 and 314 or 307/317 or 301/310 interface modules are required when using the CP 523 in an expansion unit.

Use in the S5-135U, S5-150U and S5-155U

- S5-135U central controller (Only in the normal I/O i.e. P area)

3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163
░		░	░	░	░	░	░	░	░	░	░	░	░	░	░	░				

- S5-150U central controller (Only in the normal I/O i.e. P area)

3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163
░	░												░	░	░	░	▓	▓		

- S5-155U central controller (Only in the normal I/O i.e. P area)

3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163
				░	░	░	░	░	░	░	░	░	░	░	░	░	░	░	░	░

- S5-185U expansion unit

3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163
	░	░	░	░	░	░	░	░	░	░	░	░	░	░	░	░	░			

- S5-186U expansion unit

3	19	35	51	67	83	99	115	131	147	163
	░	░	░	░	░	░	░			

░ Possible locations

▓ Possible locations without battery backup

3.1.2 installing the CP 523

You must note the following when installing and unplugging the CP 523:

. The module may only be plugged in or unplugged when the PLC is in the POWER OFF state.

- The memory submodule may only be plugged in or unplugged when the PLC is at POWER OFF.
- The connecting cable to the peripheral device (Cannon subminiature D male connector) may only be plugged in and unplugged when the CPU is at STOP and data transfer between the CP and the peripheral device has been term i nated.

Use in the S5-1 15 U/S5-l 15F

You require an adapter casing for the CP 523 (6ES5 491-OLB11)

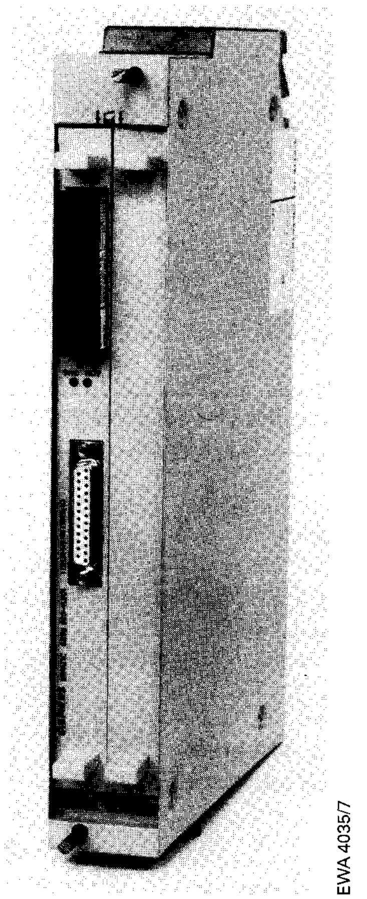


Figure 3-1. Installing the CP 523 in the Adapter Casing(6ES5491-OLB1 1)

Use in the S5-135U/S5-150U and S5-155U

The CP 523 is mounted direct onto the module rack.

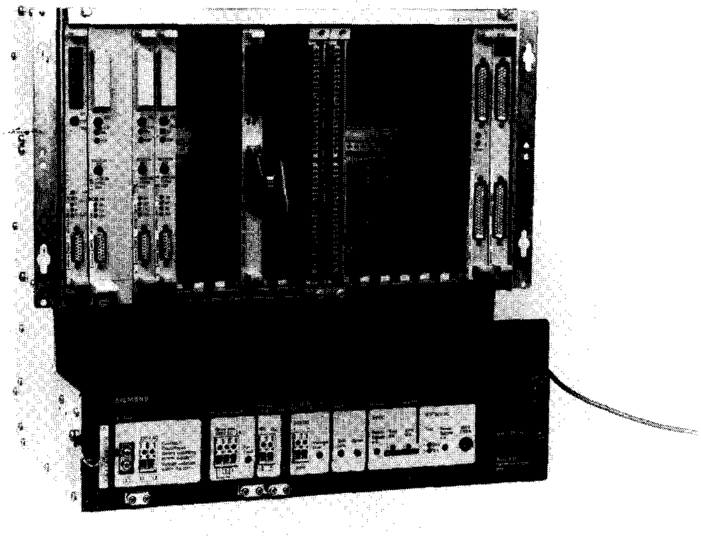


Figure 3-2. Installing the CP 523 in the S5-135U Central Controller

3.1.3 Settings on the Module

You must set the address of the module on switch bank S 1 (→ Section 5):

. Select the initial address with switches S 1.1 to S1.5

- Define the addresses in the selected I/O **area (P, O, IM 3 or IM 4) in steps of eight bytes using switches S 1.6 to S1.8**
  128, 136 to 248 in the P area
  O, 8 to 248 in the O, IM 3 and IM 4 areas

Only the P area can be selected when using the S5- 1 15u.

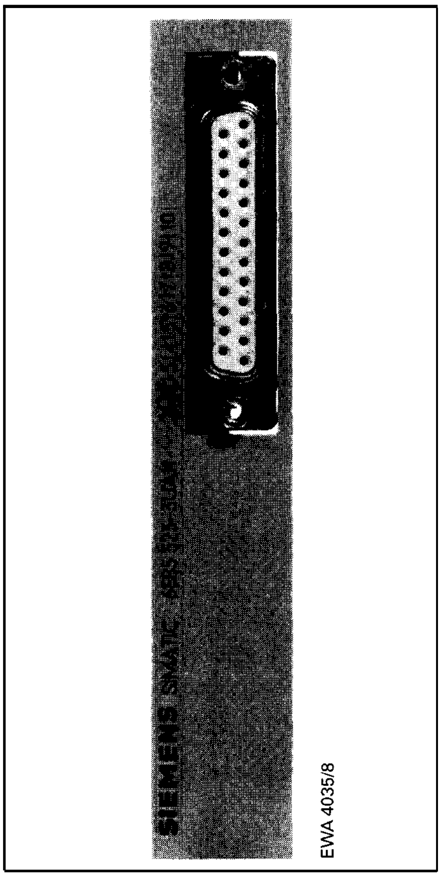
Note:
The modules are factory-set to start address 128 in the P (normal 1/0) area. Make sure that no other modules reserve this address space before putting the module into service.

CAUTION:
When using the module in the S5-1 15F, you must separate the jumpers in the jumper header (→ Figure 5-1) if the device connected (e.g. printer) has no safe electrical isolation to VDE 0160. This ensures safe electrical lsolation between the peripheral device and the S5-1 15F.

3.2 Wiring

The module has a serial interface port. You can configure either the TTY interface or the RS-232-C (V.24) interface (-D Section 6). The cables from both interfaces are run to a 25-pin Cannon subminiature D female connector.

3.2.1 Connections



Backplane connector XI

Cannon subminiature D female connector (25-pin)

Shield

Fastening screw, 4-40 VNC-2B thread

Figure 3-3. Connectors

Permissible conductor cross section for the Cannon subminiature D connector: up to 0.5 mm^2 (20 AWG)

3.2.2 Pin Assignments of the 25-Pin D Subminiature Connector

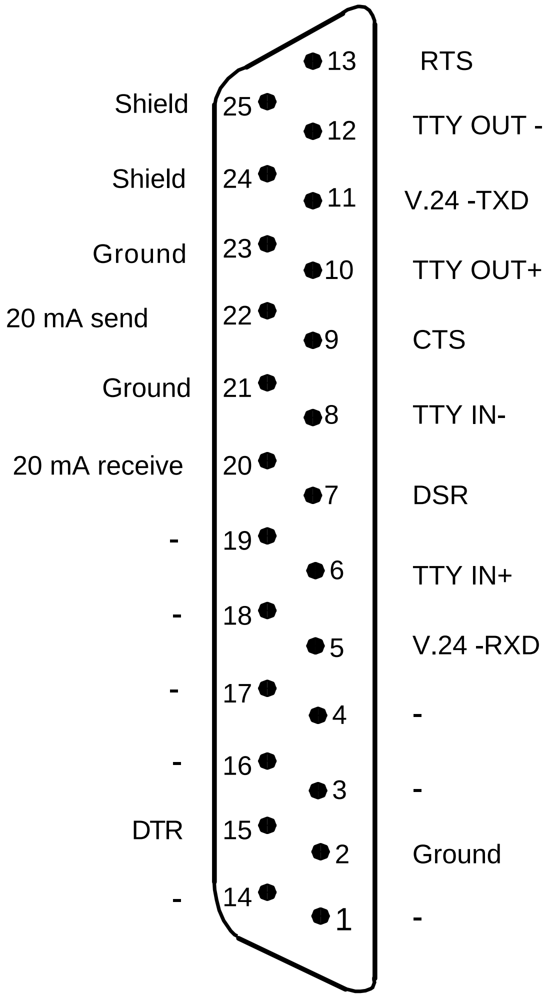


Figure 3-4. Pin Assignments of the 25-Pin D Subminiature Connector

3.2.3 Pin Assignments of Backplane Connector X1

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

Figure 3-5. Pin Assignments of Backplane Connector X1

3.3 Cable Length

The cable lengths given below are contingent to error-free data transmission.

TTY interface

•	CP 523 active TTY	: maximum cable length 10 m/33 ft.
•	Peripheral device active TTY	: Refer to the User's Guide for the relevant device for the permissible cable length (maximum 1000 m/3280 ft).

Note:
Voltage drops on the cable and the module's sending and receiving elements must be noted when using longer cables (2.2).

RS-232-C (V.24) interface

Cable length is not critical up to 15 m/49 ft. In general, longer cables can be used in conjuction with lower baud rates.

3.4 Terminal Diagrams

This section contains typical terminal diagrams for connecting the following:

- A printer (PT 88)
- A point-to-point connection (data terminal equipment DTE)
  - CP 523 to CP 523
  - CP 523 to CPU 944
- Modem link (data communications equipment DCE)

3.4.1 Terminal Diagrams for Print Mode

The CP 523 assumes a printer with an RS-232-C (V.24) or TTY interface as the peripheral device.

CP 523 (passive TTY) to PT 88 (active TTY) without BUSY signal

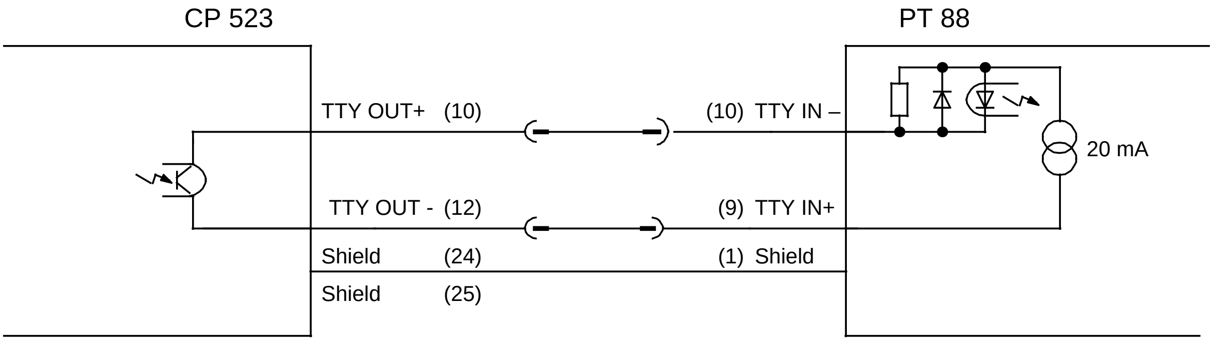


Figure 3-6. Pin Assignments for CP 523 (Passive TTY) to PT 88 (Active TTY) without BUSY Signal

CP 523 (passive TTY) to PT 88 (active TTY) with BUSY signal

Printer setting: Printer not ready to receive = no current

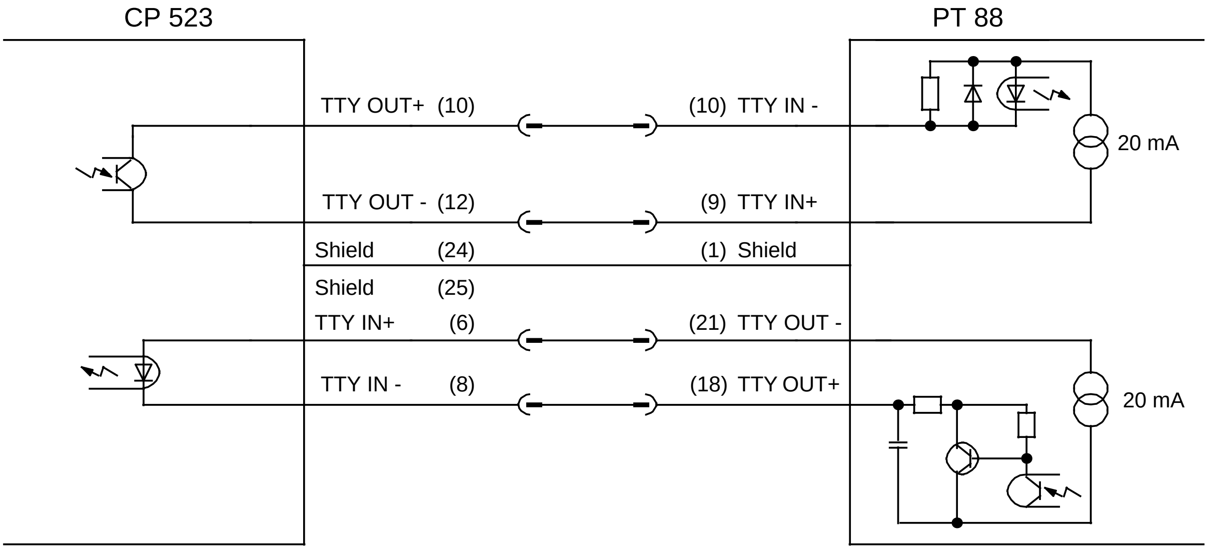


Figure 3-7. Pin Assignments for CP 523 (Passive TTY) to PT 88 (Active TTY) with BUSY Signal

RS-232-C (V.24) interface

Printer setting: Printer not ready to receive = no current

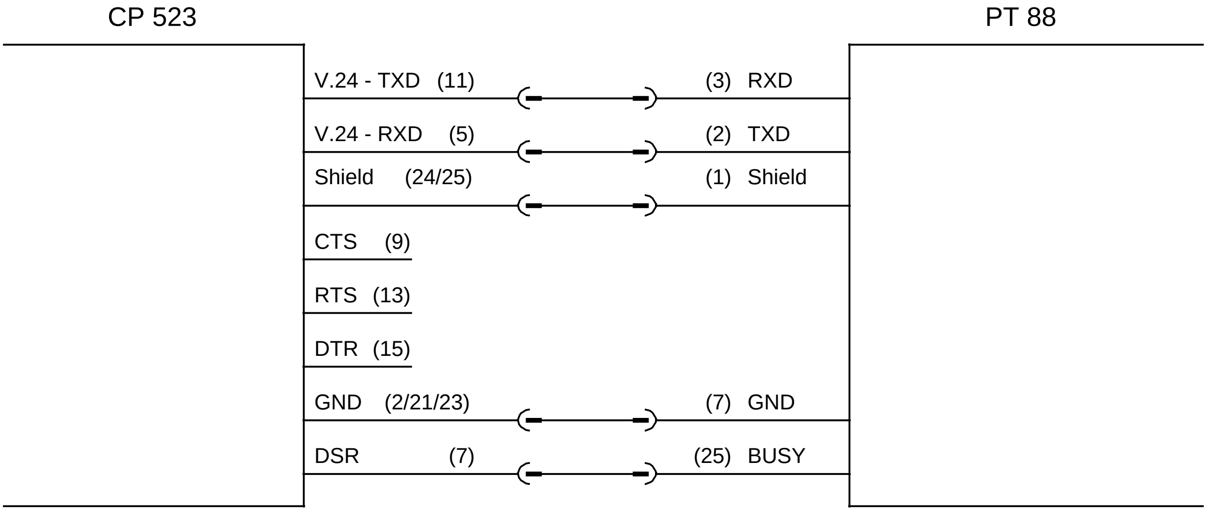


Figure 3-8. Pin Assignment of the RS-232-C (V.24) Interface (Print Mode)

3.4.2 Terminal Diagrams Communications Mode

The CP 523 assumes the following as peripheral device:

- Data terminal equipment, e.g. CP 521, CP 523, CPU 944
- Data communications equipment, e.g. a MODEM

CP 523 to CP 523 (TTY interface)

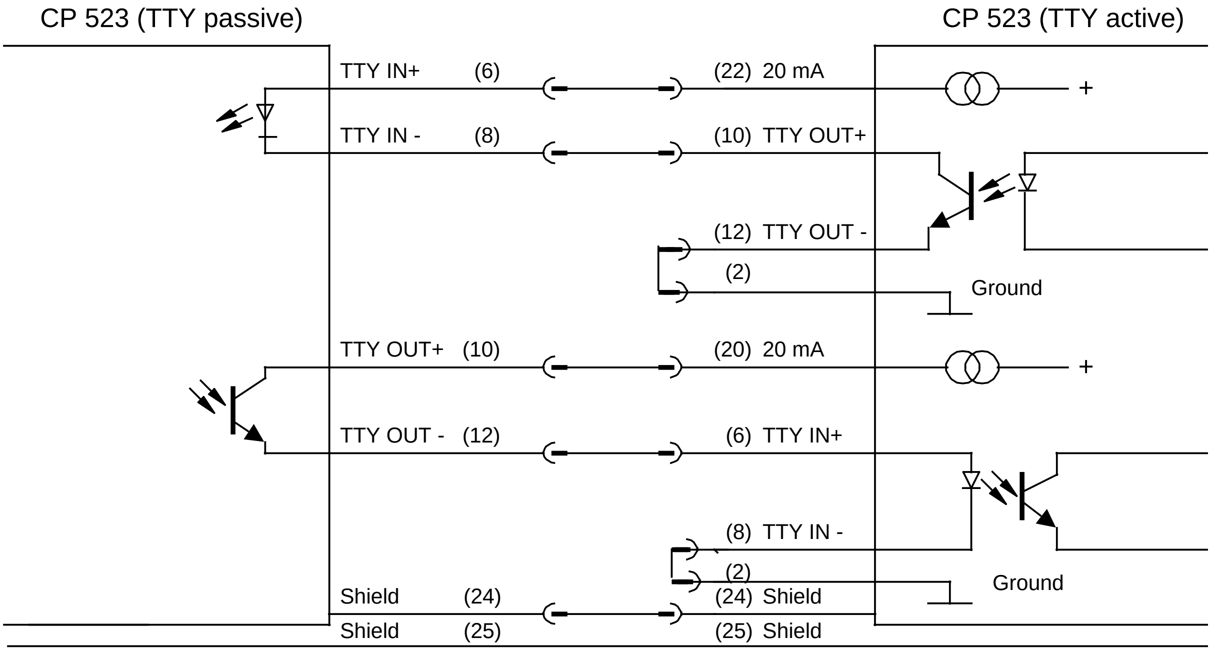


Figure 3-9. Pin Assignments for CP 523 to CP 523 (TTY Interface)

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

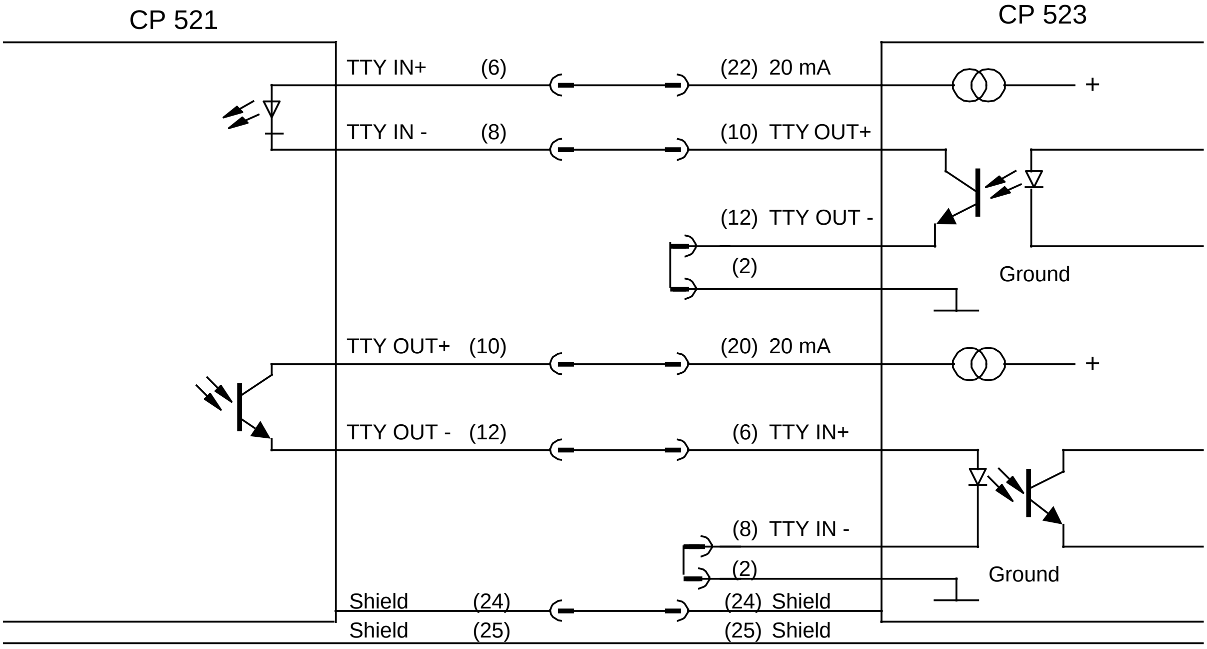


Figure 3-10. Pin Assignments for CP 523 (TTY Passive) to CP 523 (TTY Active)

CPU 944 (TTY active) - CP 523 (TTY passive)

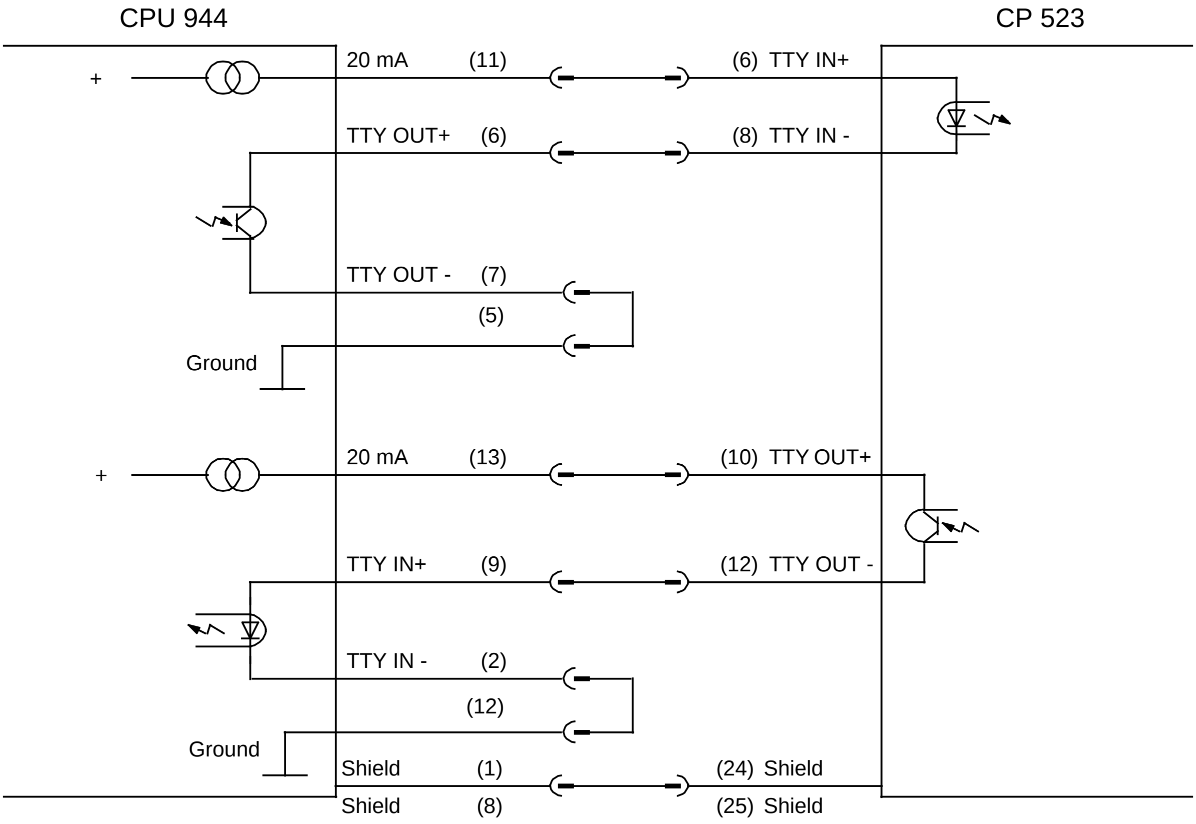


Figure 3-11. Pin Assignments for CPU 944 (TTY Active) to CP 523 (TTY Passive)

Zero modem operation: Connecting two DTEs taking the CP 523 to CP 523 example

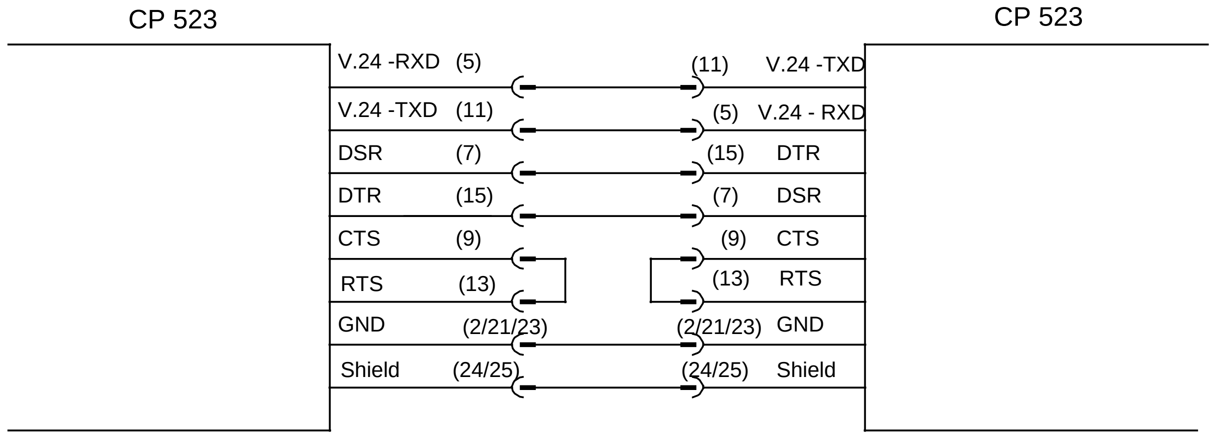


Figure 3-12. Zero Modem Operation Taking the CP 523 to CP 523 as Example

Connecting a DTE to a DCE taking the CP 523 to modem (SIEMENS 2425 B DX) as example

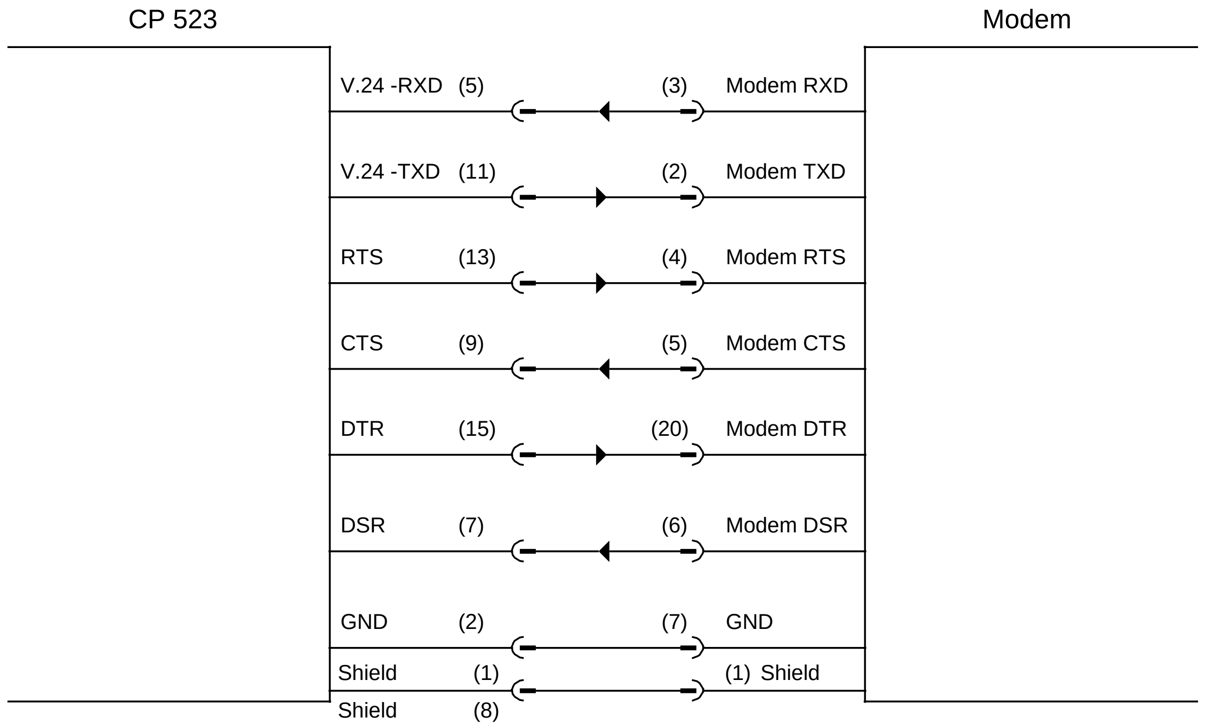


Figure 3-13. Pin Assignments for CP 523 to Modem

Note:

Only relevant in the case of communications using an RS-232-C (V.24) interface!
The ”3964(R) communications mode” data transmission procedure does not support control lines (DSR, DTR, CTS, RTS).

Tables

4 Notes on Operation

4.1 Restart Characteristics

The CP 523 executes a restart routine when the power supply is restored (POWER ON).

The restart procedure consists of the following:

- Clearing the Send and Receive mailboxes
- Checking the module (4.1.1)
- Memory submodule evaluation (4.1.2)
- Clock test (4.1.3)

If errors occur during the restart procedure, the CP 523 forwards an error flag to the CPU in byte 0 "Module status" (Print mode 6.6, Communications mode 7.4).

Note:
The message buffer is not deleted on restart in Print mode if the module was battery-backed and the memory submodule has not been replaced.

4.1.1 Checking the Module

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

Table 4-1. Module Errors/Faults

Error/Fault Number in Byte 0	Error	Remedy
$1X_H$	Clock defective	Replace module
$8X_H$	Hardware fault	Replace module
$X7_H$	No battery backup	Insert new battery in the power supply module
XF_H	CP in restart routine	Scan the status byte for XF_H before issuing the first job request

X : can assume different values

During the restart procedure, the CP signals "Module busy" (XF_H) in the status byte. The CP cannot accept any jobs during this time. For this reason, you must scan the module's status byte (byte 0) for the XF_H before issuing the first job request.

OB 22	FB 99	Explanation
JU FB99	M001 :L KH 0000 :T PW 128* :L KH 000F :L PY 128 :AW :!=F :JC =M001 :BE	Wait loop until the CP 523 has completed the restart routine.

* 5.2

Note:
The restart organization block OB 22 is only processed if the CPU is in RUN mode before you switch from POWER OFF to POWER ON.
Variations in the restart behaviour of the CP 523 or its communications partner can lead to the loss of data during the restart phase.

4.1.2 Memory Submodule Evaluation

The serial interface parameters and message texts specified on the module are checked in this routine. Errors are assigned an appropriate number in byte 0 "Module status" and can be evaluated in the control program.

Table 4-2. Memory Submodule Faults

Error Number in Byte 0	Error	Remedy
$X1_H$	Memory submodule defective	PLC POWER OFF plug in functional memory submodule PLC POWER ON
$X2_H$	No message texts configured	Configure message texts (6.4) Wrong or defective memory submodule plugged in?

X : may assume different values

4.1.3 Clock Test

The module's real-time clock is tested. The clock is set when the battery back-up fails during PLC POWER OFF. The clock has the default setting 12:00:00 01.01.90.
Errors and the default setting are provided an appropriate number in byte 0 "Module status" and can be evaluated in the control program.

Table 4-3. Clock Test Errors

Error Number in Byte 0	Message	Remedy
$1X_H$	Clock defective	Replace module
$2X_H$	Clock set Default setting	Transfer the correct time to the CP with the programmer "FORCE VAR" Junction

X : may assume different values

4.2 Performance during Operation

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

- The CP 523 may be plugged in or unplugged only when the power is switched off.
- The memory submodule may only be plugged in or unplugged when the power is OFF.
- To guarantee trouble-free printing, make sure the 25-pin subminiature D female connector is only unplugged when the CPU is in STOP mode and data transfer between the CP and the printer has been completed.

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

4.2.1 Print Mode

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

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

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

* Print jobs (6.5)

** Characters in the internal printer buffer are lost

4.2.2 Communications Mode

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

Table 4-5. Faults in Operation (Communications Mode)

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

* The data transfer between CPU and CP (Send or Receive) is aborted. The job must be restarted following transition from STOP to RUN. This is of particular relevance for the S5-135U / 150U / 155U.

** Data cannot be corrupted in 3964(R) mode, as the message frames are transmitted in a protocol message frame (7.4).

Note:

If you operate the RS-232-C (V.24) interface without handshaking (Handshake OFF), the CP 523 cannot detect a break in the connection between the CP and the peripheral device.

Figures

Tables

5 Address Assignment

This section describes

- how to set the module address (5.1)
- when the CP accepts data from the CPU (5.2)
- when the CP supplies current data for the CPU (5.2)

5.1 Setting the Start Address and the I/O Area

The CP incorporates an eight-byte transfer memory. All data between the CPU and the CP is exchanged via this area, which can be read and written to by the CPU. You must set the following with the DIP switches on the S1 switch bank:

- The desired start address with switches S1.1 to S1.5
- The desired I/O area with switches S1.6 to S1.8

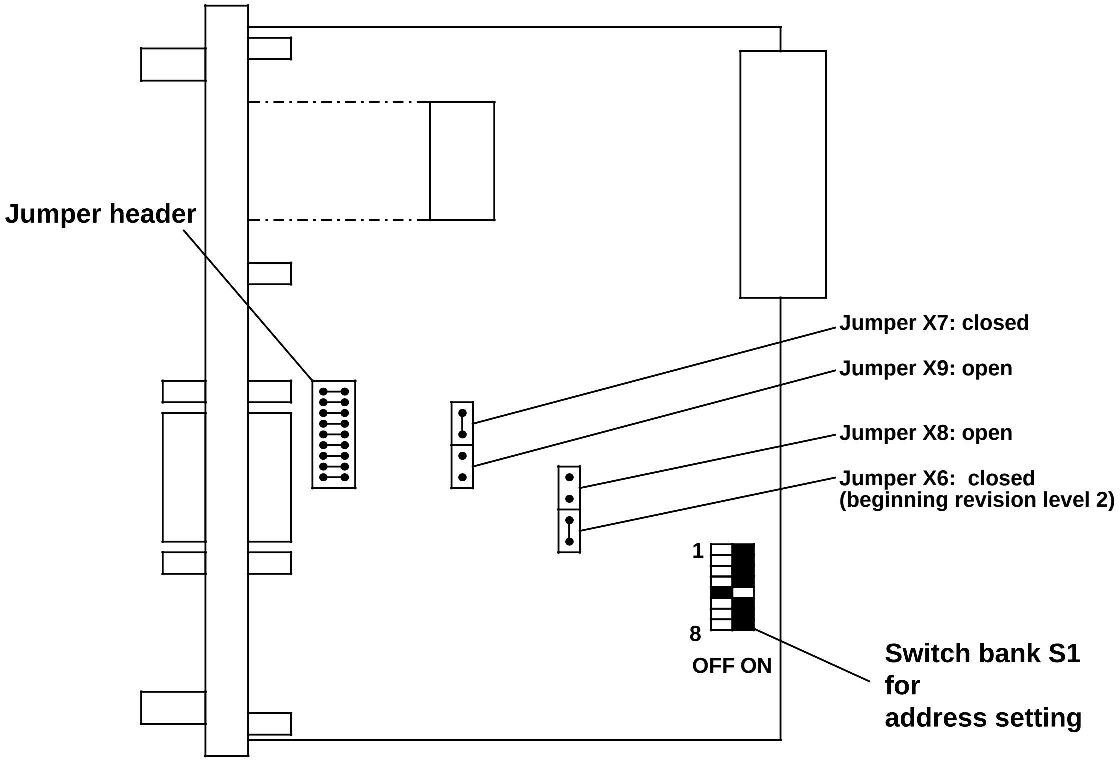


Figure 5-1. Address Switch Locations

Note:

The modules are factory-set to start address 128 set in the P (normal I/O) area. Make sure before startup that there are not several modules reserving the same address space.

The byte and word numbers specified in this manual always refer to the start address that you have set:

Example: You have set start address 128 using DIP switches S1.1 to S1.5
Word 0 has address 128 + 0 = 128, word 6 address 128 + 6 = 134.

Table 5-1. Switch Settings on Switch Bank S1 for Defining the Start Address

Start Address (not in the P Area)	Switch setting Switch bank S1					Start Address	Switch Setting Switch Bank S1				
	1	2	3	4	5		1	2	3	4	5
0	▄	▄	▄	▄	▄	128	▄	▄	▄	▄	▀
8	▀	▄	▄	▄	▄	136	▀	▄	▄	▄	▀
16	▄	▀	▄	▄	▄	144	▄	▀	▄	▄	▀
24	▀	▀	▄	▄	▄	152	▀	▀	▄	▄	▀
32	▄	▄	▀	▄	▄	160	▄	▄	▀	▄	▀
40	▀	▄	▀	▄	▄	168	▀	▄	▀	▄	▀
48	▄	▀	▀	▄	▄	176	▄	▀	▀	▄	▀
56	▀	▀	▀	▄	▄	184	▀	▀	▀	▄	▀
64	▄	▄	▄	▀	▄	192	▄	▄	▄	▀	▀
72	▀	▄	▄	▀	▄	200	▀	▄	▄	▀	▀
80	▄	▀	▄	▀	▄	208	▄	▀	▄	▀	▀
88	▀	▀	▄	▀	▄	216	▀	▀	▄	▀	▀
96	▄	▄	▀	▀	▄	224	▄	▄	▀	▀	▀
104	▀	▄	▀	▀	▄	232	▀	▄	▀	▀	▀
112	▄	▀	▀	▀	▄	240	▄	▀	▀	▀	▀
120	▀	▀	▀	▀	▄	248	▀	▀	▀	▀	▀

Table 5-2. Settings on Switch Bank S1 for Defining the I/O Area

I/O Area	Absolute Address in RAM	Switch Setting Switch BankS1			Remarks
		6	7	8	
P (normal I/O) area	F080$_H$...F0FF$_H$	▄	▄	▄	Beginning start address128
not in the S5-115U					
Q (extended I/O) area	F100$_H$...F1FFF$_H$	▀	▄	▄	... Can only be read and written to with LIR and TIR
IM 3	FC00$_H$...FCFF$_H$	▄	▀	▀	Can only be read and written to with LIR and TIR
IM 4	FD00$_H$...FDFF$_H$	▀	▀	▀	

▀ =on ▄ =off

5.2 Access to the Transfer Memory

The CPU can write data to the transfer memory at any time.
After the CPU has written the relevant data into word 0,

- the CP 523 fetches data from the transfer memory
- the CP 523 updates the transfer memory with current data
- the CPU can read the current data from the transfer memory

This has the following consequences for the order in which data is transferred from the CPU to the transfer memory:

- Words 2, 4 and 6 in the transfer memory must be written first if the job requires it. If you want to print a message text into which three variables are to be inserted, for example, the actual variable values must be transferred in words 2, 4 and 6.
- Finally, the job number must be written to the transfer memory in word 0.

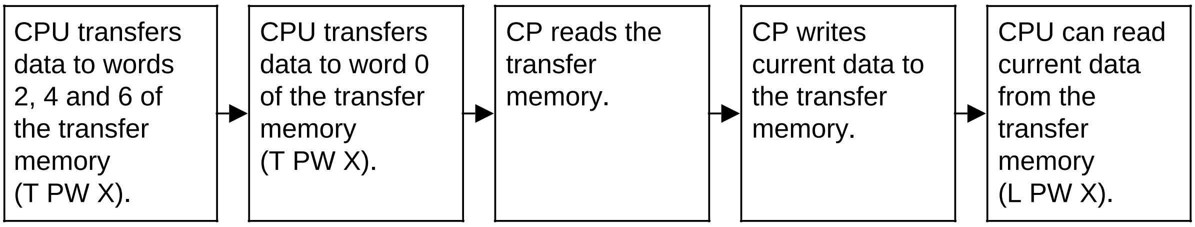

CPU transfers data to words 2, 4 and 6 of the transfer memory (T PW X).
CPU transfers data to word 0 of the transfer memory (T PW X).
CP reads the transfer memory.
CP writes current data to the transfer memory.
CPU can read current data from the transfer memory (L PW X).

Figure 5-2. Using the Transfer Memory

Note:
If you do not observe this order, the new job may be executed with the wrong values.

Special feature of the CPU 944 when accessing the transfer memory

If you are using the CPU 944, you may not write data to word 0 of the transfer memory with T PW <Start address of the module>. You must write the data to word 0 byte for byte in the following order:

1. T PY <Start address of the module> (byte 0)
2. T PY <Start address of the module + 1> (byte 1).

Example: Writing to word 0 of the transfer memory with the CPU 944 in the user program

Initial address 128 is set in the P area on the CP. The job number for "Print message text No. 20" is to be written to word 0.

STL FB XYZ	Explanation
L KH 0030	Load ACCUM 1 with the job number "Print message text".
T PY 128	Transfer the low byte of ACCUM 1 to the CP.
L KF+20	Load messsage text number into ACCUM 1.
T PY 129	Transfer the low byte of ACCUM 1 to the CP.

Example: Writing to word 0 of the transfer memory with the CPU 944 using the programmer's "FORCE VAR" function

Start address 128 is set in the P area on the CP. The job number for "Print all configured message texts" (8000_H) is to be transferred to word 0 using the "FORCE VAR" function.
Only when you convert a peripheral (I/O) word (PW) to a flag word (FW) can you force it with the "FORCE VAR" function. For this reason, store PW 128 in FW 128 of FY 10. FW 128 can then be forced in the "FORCE VAR" screen form. FW 228 is an auxiliary word for comparing the old/new values.

Operand	Signal States	Explanation

PY 128
PY 129
	KH=80
	KH=00
		Write the "Print all configured message texts" job number 8000_H to bytes 0 (80_H) and 1 (00_H) of the transfer memory

STL FB 10	Explanation
:	
:L FW 228	Load old value
:L FW 128	Load new value
:! = F	Compare for equal
:BEC	Block end if equal
	Update old value
:L FW 128	if not equal and
:T FW 228	
:L FY 128	Transfer flag word byte by byte
:T PY 128	to the peripheral (I/O) word
:L FY 129	
:T PY 129	
:BE	Block End

Figures

Tables

6 Print Mode

6.1 General

The CP 523 enables the output of message texts to a printer in Print mode:

- Output of message texts defined by you in data blocks (DB) 2 to 255 on the memory submodule.
- Insertion of the actual time of day and date in the message text to be printed out. The clock data is supplied by the integral real-time clock.
- Insertion of current variables in the printout.
  The variable values can be transferred from the CPU to the CP 523.

6.2 Prerequisites for Operation in Print Mode

The following conditions must be met in order to operate the CP 523 in Print mode:

1. Printer settings

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

Note:
Set the parameters for the serial interface as required and then set the parameters in DB 1 in the memory submodule.

- See 3.4.1 for configuration examples with terminal diagrams.
- Parameter setting in DB 1 is explained in 6.3.

2. Setting the CP 523 parameters

Store all parameter setting data in DB 1 on the memory submodule :

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

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

3. Configuring message texts

You must configure at least one message text in one of the DBs 2 to 255 on the memory submodule.

Note:

If you have configured a message text in a DB, you can configure further message texts later in other DBs and store them on the memory submodule.
See 6.4 for information on configuring message texts.

4. Establishing connections

You must take the following measures when the programmable controller is switched off:

- Install the CP 523 in the central controller or the expansion unit.
- Establish a connection between the CP 523 and the printer.
- Plug the configured memory submodule into the CP 523.

You can then power up the programmable controller (POWER ON).

5. Startup of the CP 523 in Print mode

If the CP 523 has a memory submodule, it is automatically in Print mode after power-up.
If the CP 523 has no memory submodule, it is automatically in Transparent mode.

Changing operating modes (during operation)

You can change back and forth between "Print mode", "Transparent mode" and "Interpretive mode" during operation.

Note:

If you

- change the mode
  or
- reinvoke the programmed mode,
  the send and receive buffers and all print requests will be deleted.

To change the operating mode, forward the job number for "Transfer parameter setting data" (90XX$_H$) to the CP 523. Remember that the parameters must agree with the settings on your peripheral device.

6.3 Setting the CP 523 Parameters in Print Mode

The CP 523 can only be operated in Print mode if an initialized memory submodule has been plugged in. Store the following in the memory submodule:

- Parameter setting data for the CP 523 in data block (DB) 1
- Message texts in DB 2 to 255 (6.4)

The CP 523 has default values for all parameters.

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

If you want to configure the CP 523 with other data, you must store the configuring data on a memory submodule in DB 1. This data is subdivided into ten parameter blocks to simplify entry as much as possible.

Table 6-1. Contents of the Parameter Blocks

Parameter Block Number	Contents
	Setting the serial interface parameters:
0	Serial interface parameters
1	Waiting times after "CR", "LF" and "FF" (only relevant if BUSY signal "0" and no XON/XOFF protocol has been configured)
2	XON/XOFF protocol (only relevant if XON/XOFF protocol is in use) The XON/XOFF protocol has priority over the BUSY signal
	Configuring data for entering message texts:
3	Text parameters
	Configuring data for message text printout:
4	Output format for date and time of day
5	Page format
6	Headers and footers (only relevant if headers and footers are to be printed)
	Setting the mode:
7	Parameters for Communications mode
8	**Character conversion table**
9	**Correction value for the integral clock**

Dividing the parameters into ten parameter blocks enables you to initialize only the parameter block you want to modify. However, this parameter block must be entered in its entirety even if individual parameters within the block are to retain their default values.

Entering the parameter blocks

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

Note:

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

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

Further tips for input:

- When entering text, alternate between data format KS and S for each line. This increases the clarity of presentation on the programmer screen.
- You can enter comments in addition to the parameter blocks.

Schematic for entering parameter blocks

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

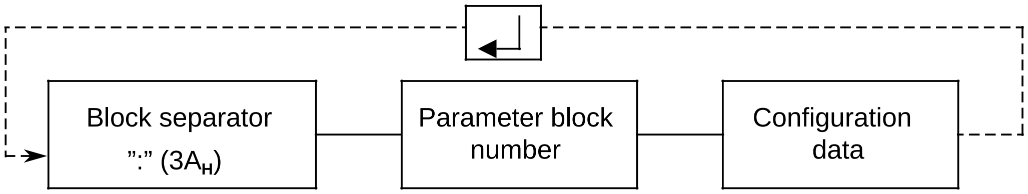


Figure 6-1. Schematic for Entering Parameter Blocks in DB 1

Note:

Parameter blocks 0, 2 and 7 can also be initialized in the user program. Parameters set in the user program have priority over those set on the memory submodule (7.3.2).

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

Store the data for the serial interface in parameter blocks 0, 1 and 2. If you enter an illegal value, it will be replaced by the default value.

Table 6-2. Parameter Setting Data for the Serial Interface

Parameter Block	Description	Value Range	Default Value on the CP 523
0	Baud rate 110 Bd	1	
	200 Bd	2	
	300 Bd	3	
	600 Bd	4	4
	1200 Bd	5	
	2400 Bd	6	
	4800 Bd	7	
	9600 Bd	8	
	Parity even	0	0
	odd	1	
	”mark”	2	
	”space”	3	
	no parity check	4	
	BUSY signal no	0	0
	yes	1	
	Interface TTY	0	0
	V.24	1	
	Data format: Parity:		
	11-bit character 7 data bits (yes)	0	0
	frame 8 data bits (yes)	1	
	8 data bits (no)	2	
	10-bit character 7 data bits (no)	3	
	frame 7 data bits (yes)	4	
	8 data bits (no)	5	
	Hardware handshaking OFF	0	0
	ON	1	
1	Waiting time after CR	(00_H to FF_H) · 25ms	$0A_H$ 0.25 s
	LF	(00_H to FF_H) · 25ms	$0A_H$ 0.25 s
	FF	(00_H to FF_H) · 25ms	$A0_H$ 4 s
2	XON character	ASCII character (01_H ... $7F_H$)	FF_H (no XON / XOFF - protocol)
	XOFF character	ASCII character (01_H ... $7F_H$)	FF_H (no XON / XOFF - protocol)

Baud rate

You have a choice of eight baud rates. The default is 600 baud. If you use the RS-232-C (V.24) interface, you must take the load capacitance of cables longer than 15 m into account. If you operate the CP 523 with an active TTY interface, a cable length of up to 10 m/33 ft. is possible. With a passive TTY interface, cable lengths of up to 1000 m/3280 ft. are possible.
Longer cables can be implemented in general in conjunction with reduced baud rates.

Parity

You have a choice of five types of parity.

- Even parity
  The parity bit is set when the number of data bits with a value of ”1” is uneven.
- Odd parity
  The parity bit is set when there is an even number of data bits with a value of ”1”.
- Mark
  The parity bit is always "1".
- Space
  The parity bit is always "0".
- No parity check
  The signal state of the parity bit is not significant. Parity is not checked when receiving; however, the parity bit is always set to "1" when sending.

Even parity is the default.

BUSY signal

The BUSY signal is only relevant if you operate 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

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

Data format

Characters are transmitted between the CP and the peripheral device in a 10-bit or 11-bit character frame. You can choose between seven and eight data bits within the character frame:

10-bit-character frame:

• 1 start bit, 7 data bits, 2 stop bits	(data format 3 in parameter block 0)
• 1 start bit, 7 data bits, 1 parity bit, 1 stop bit	(data format 4 in parameter block 0)
• 1 start bit, 8 data bits, 1 stop bit	(data format 5 in parameter block 0)

11-bit-character frame:

• 1 start bit, 7 data bits, 1 parity bit, 2 stop bits	(data format 0 in parameter block 0)
• 1 start bit, 8 data bits, 1 parity bit, 1 stop bit	(data format 1 in parameter block 0)
• 1 start bit, 8 data bits, 2 stop bits	(data format 2 in parameter block 0)

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

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

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

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

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

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

Parameter Block	Text Parameter	Value Range	Default Value on the CP 523
3	End-of-text character	ASCII character (01_H to $7F_H$)	$ (24_H)
	Function character	ASCII character (01_H to $7F_H$)	” (22_H)

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

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

- If the lower limit is exceeded, the lower limit value is set.
- If the upper limit is exceeded, the upper limit value is set.

Example:

Input: Lines per page "10_H" is replaced by "14_H"
Input: Left margin "80_H" is replaced by "$3C_H$"

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

Parameter Block	Description	Value Range	Default Value on the CP 523
4	Date and time of day: Order for date J,Y = Year M = Month T,D = Day	Any combination of year, month and day is permissible (lowercase 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
		(lowercase also permissible)
			HMS

	Separator for time of day	ASCII character (20_H to $7F_H$)	”:” ($3A_H$)
	24h clock 12h clock	d, D e, E	D (24h clock)
5	Page format Lines per page Left margin Page number top bottom none	 14_H to FF_H 00_H to $3C_H$ o, O, h, H u, U, f, F other characters	 48_H (72_D) 00_H u
6	Header and footer Header 1 Header 2 Footer 1 Footer 2	 K1”Text”, H1”Text” K2”Text”, H2”Text” F1”Text” F2”Text”	No header or footer

Page format

The page format is defined by the information given in parameter blocks 5 and 6.

Table 6-5. 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. at top
.		Blank line	(if page number configured at top)
.		Header 1	
.		Header2	
.		Blank line	(if header(s) configured)
.		Blank line	(if header(s) have been configured)
.			
.			
.			Lines for message texts
.			A maximum of 4095 message texts can be configured.
.			A message text may not exceed 136 characters
.			Configuration is described in 6.4.
.			
.			
.			
.		Blank line	(if footer(s) configured)
.		Blank line	(if footer(s) configured)
.		Footer 1	
.		Footer 2	
.		Blank line	(if the page number is configured at bottom)
.			Page number at bottom
.		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
(configurable within the range 14_H to FF_H)

Note:

The CP 523 counts the lines on a page using the line feeds (LF) sent to the printer.
If you print 15 message texts with CR/LF, for example, the CP counts 15 lines. The CP does not count the additional lines for message texts longer than one line.

Configuring headers and footers

You can configure up to two headers and two footers.

The headers and footers may contain spaceholders for the date and time of day as well as for control characters. Spaceholders for variables and message texts are not evaluated, and appear in the printout as configured.

Headers and footers are configured according to the following schematic:

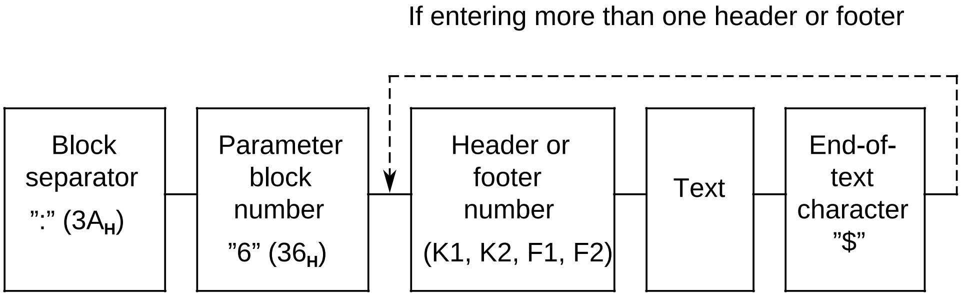

If entering more than one header or footer
Block separator ":" (3A$_H$)
Parameter block number "6" (36$_H$)
Header or footer number (K1, K2, F1, F2)
Text
End-of-text character "$"

Figure 6-2. Schematic for Entering Headers and Footers

Enter the header or footer in the same manner as the ASCII characters of a message text (6.4).

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

Continuous printout

It is recommended that the message text printout be configured as followes if you want to output your message texts to screen:

• without page number	(parameter block 5)
• without header	(parameter block 6)
• without footer	(parameter block 6)

You will get so a continous printout of all message texts.

6.3.4 Configuring the Character Conversion Table (Parameter Block 8)

Up to sixteen ASCII characters from the printer character set can be converted to another character set for the purpose of adapting to special national characters.

Block 8 must be configured according to the following schematic:

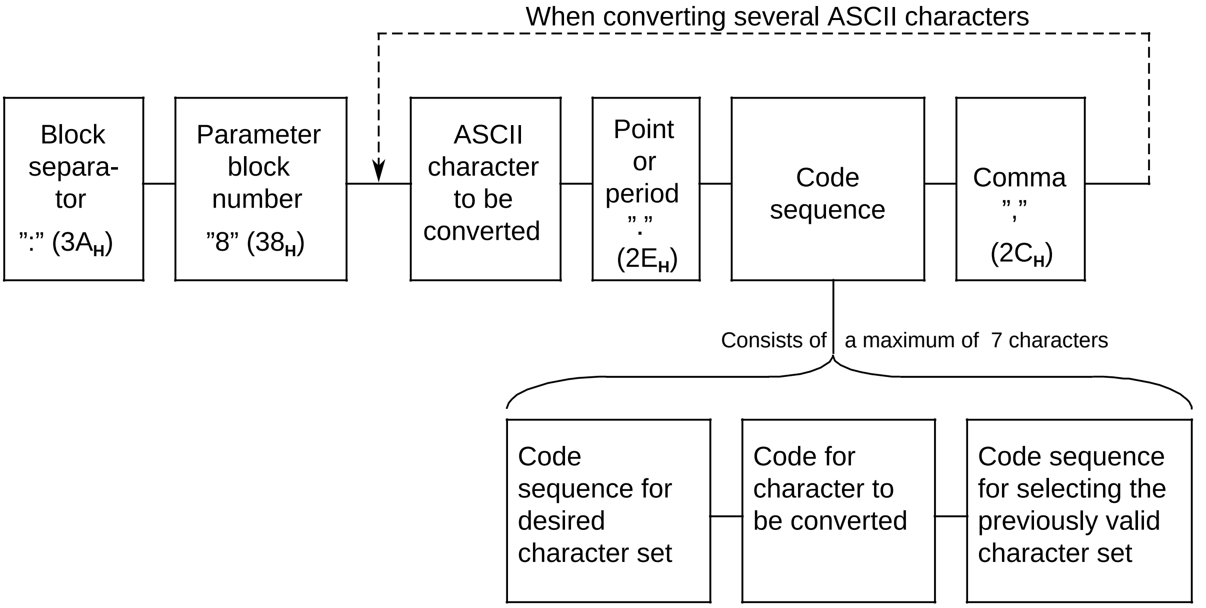

When converting several ASCII characters
Block separa-tor
":" (3A$_H$)
Parameter block number
"8" (38$_H$)
ASCII character to be converted
Point or period
"."
(2E$_H$)
Code sequence
Comma
","
(2C$_H$)
Consists of a maximum of 7 characters
Code sequence for desired character set
Code for character to be converted
Code sequence for selecting the previously valid character set

Figure 6-3. Schematic for Configuring the Character Conversion Table

Example: Configuring the character conversion table for the PT 88/89

The printer is initialized for the international character set. It also contains the character sets of other languages, among them German.

You want to print the following:

1. The character "Ä" ($5B_H$ of the German character set)
   The character with the ASCII code "$5B_H$" "[" of the international character set must be converted.
2. The character "Ö" ($5C_H$ of the German character set)
   The character with the ASCII code "$5C_H$" "\" of the international character set must be converted.

Selecting the German character set: $1B_H$ 28_H $4B_H$ [ASCII character:ESC(K]
Selecting the international character set: $1B_H$ 28_H 40_H [ASCII character:ESC(@].

Entry on PG 675 for DB 1	Explanation
`KS = :8`	Block separator ”:”, Parameter block ”8”,
`KH = 5B2E`	Character to be converted ”[”, Point/period”.”
`KH = 1B28`	Code sequence for German character set
`KH = 4B5B`	Code cont., character "Ä"
`KH = 1B28`	Code sequence for international character set
`KH = 402C`	Code cont., comma ”,”*
`KH = 5C2E`	Character to be converted ”\”, Point/period”.”
`KH = 1B38`	Code sequence for German character set
`KH = 4B5C`	Code cont., character "Ö"
`KH = 1B28`	Code sequence for international character set
`KH = 402C`	Code cont., comma ”,”*

* The comma tells the CP 523 that conversion of the previous character is complete

6.3.5 Configuring the Correction Value for the Integral Clock (Parameter Block 9)

You can configure a correction value to increase the accuracy of the clock. The correction value is printed in s/month. A month is defined as 30 days.

Table 6-6. 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 as a four-digit number (incl. sign)!

Example: Determining the correction value

You have observed that the clock loses 12 s in 4 days. This would amount to 90 s in 30 days. The correction value is then +090 s/month.

Entry on the PG for DB 1	Explanation
`:   KS = ':9+090';`	Block separator ”:” Parameter block number ”9” Correction value "+090"

6.3.6 Example for Configuring the CP 523 in Print Mode

Start address 128 is set on the CP 523. Configure the CP as follows:

- Parameters for the serial interface . . . . (parameter block 0)
  - 2400 baud (6)
  - Even parity (0)
  - No BUSY signal (0)
  - RS-232-C (V.24) interface (1)
  - 7 data bits (11-bit frame) (0)
  - Handshake OFF (0)

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

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

- Configuration data for entering message texts . . . . (parameter block 3)
  No change with respect to default values

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

Entry on the PG 685 for DB 1	Explanation

:   KS = 'Parameters for the serial';
:   S = 'interface';
:   KS = ':0600100';
	Parameter block 0
	2400 baud (6_H), Even parity (0_H),
	No BUSY signal (0_H), RS-232-C (V.24) interface (1_H),
	7 data bits (0_H), (11-bit frame)
	Handshaking OFF (0_H),

:   S = 'XON/XOFF protocol';
:   KS = ':2';
:   KH = 1113
	Parameter block 2
	XON character: DC 1 (11_H)
	XOFF character: DC 3 (13_H)

:   S = 'Message text printout';
:   KS = ':5';
    KH = 400A
    KS = 'B'
:   S = 'Headers and footers';
:   KS = ':6K1Test mode$';
:   S = 'F1Laboratory$';
	Parameter block 5
	64 lines/page (40_H), 10 characters left margin ($0A_H$)
	Page number at bottom
	Parameter block 6
	Header 1: Test mode
	Footer 1. Laboratory

:	

6.4 Configuring Message Texts

Store the message texts on the memory submodule in data blocks 2 to 255 under a message text number in the range 1 to 4095. Each message text may comprise up to 136 characters, including variables. Memory submodule capacity must be taken into account.

The message texts can be distributed over several data blocks in any sequence:

- You can store the message texts in any data blocks (e.g. message text 13 in DB 20, message texts 5 and 18 in DB 7, ...).
- You can enter the DBs on the programmer in any order (DB 17, DB 3, ...).
- The message texts within a DB should be continuous. Your message texts will be clearer if you separate the message texts from each other by pressing the enter key (Figure 6-4).
- The DBs on the memory submodule are independent of the DBs in the CPU. For example, you can store message texts in DB 54 on the memory submodule while simultaneously using a DB 54 in the user program.

6.4.1 Structure of a Message Text

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

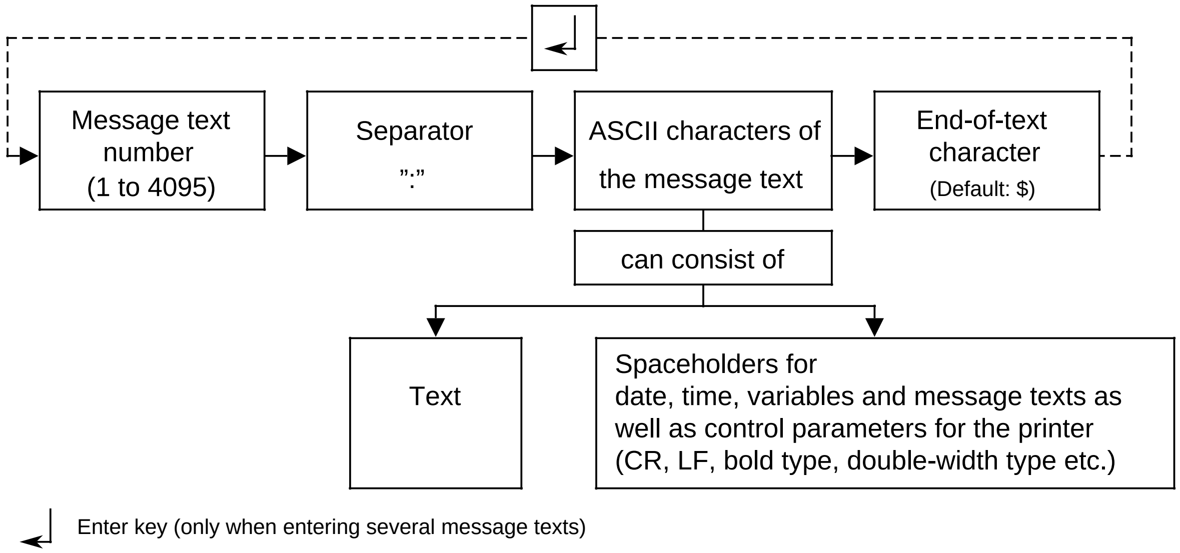

Message text number (1 to 4095)
Separator ":"
ASCII characters of the message text
End-of-text character (Default: $)
can consist of
Text
Spaceholders for date, time, variables and message texts as well as control parameters for the printer (CR, LF, bold type, double-width type etc.)
Enter key (only when entering several message texts)

Figure 6-4. Structure of a Message Text

Note:
To make screen presentation clearer, alternate between data formats "KS" and "S" in the input lines.

Message text number
The message text is stored in the DB under a message text number (1 to 4095).If you want to print a message text, transfer the message text number and a "Print message text" request to the CP 523 in the user program (6.7.1).

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

ASCII characters in the message text
You can enter up to 136 ASCII characters of message text. Characters exceeding 136 are not evaluated. The printout of the configured message text on the printer may be longer than 136 characters.

Example: You configure a message text which is longer than 136 characters.

The message text includes spaceholders for the time of day. This requires an entry of only three ASCII characters (e.g. "t"). On output, the spaceholder is replaced by the current time of day (e.g. 15:00:00) (8 characters).
The message text printed out is thus 141 characters long.

Note:

A message text of more than 136 may corrupt the print format (depending on printer line feed, page makeup...).

The following can be entered as ASCII characters:

- Text
  The text can contain all printable characters (see the manual for the printer).
- Spaceholders
  You can insert the following in a message text by configuring spaceholders:
  - The date
  - The time of day
  - Variables
  - Further message texts

  In addition, you can enter spaceholders for control parameters (double-width characters, subscript...).

Example: Configuring message texts without spaceholders

You want to store the following message texts in DB2:
Message text 4: Excess temperature
Message text 5: Coolant loss

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

6.4.2 Entering Spaceholders

Spaceholders are used for inserting the date, time of day, variables, control parameters and further message texts.

Some spaceholders can also be used in headers and footers. The headers and footers are configured on the memory submodule in data block 1, parameter block 6 (6.3.3).

A spaceholder is enclosed between function characters. You can configure the function characters in data block 1, parameter block 3. The default value for the function characters is 22_H = ".

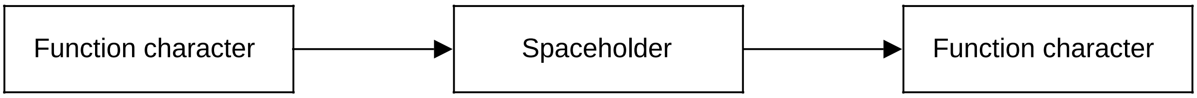


Figure 6-5. Structure of a Spaceholder

Wrongly entered spaceholders are treated as text.

Table 6-7. Overview of Spaceholders Configurable in Message Texts

Space-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	
Sa,b, . . .	Transfer printer control parameters	max. 50	Yes	
KPa	Insert message text (fixed message text number)	Unlimited	No	The message text to be printed must be 250 characters long
KV	Insert message text (variable message text number)	3	No	The message text to be printed must be 250 characters long. Transfer message text numbers in bytes 2 to 7 to the CP.
Data format (KH, KF, KG...)	Insert variable	Three 16-bit variables or one 32-bit variable + one 16-bit variable	No	Transfer values of the variables in bytes 2 to 7 to the CP

You can enter spaceholders in either lower or uppercase.

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

Transfer of data when using the "Insert message text" spaceholder KV and the "Insert variables" spaceholders.

Transfer the messsage text numbers for the "Insert message text" spaceholder KV and the values for the "Insert variables" space-holders to bytes 2 to 7 on the CP.

Access to bytes 2 to 7 depends on the order of the spaceholders in the message text.

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

Special rules apply to the use of spaceholders for "Insert floating-point number" (6.4.5).

6.4.3 Spaceholders for Date and Time of Day

When printing out a message text the following takes place:

- The spaceholder "D" for the date is replaced by the current date.
- The spaceholder "T" for the time of day is replaced by the current time.

The output format of the date and the time of day can be configured. You must do this in data block 1, parameter block 4, on the memory submodule (6.3.3).

The following is the default output format:

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

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

You want to store the following message texts in DB 3:

Message text 7:	The motor went down at <time of day>.
Message text 8:	This is the daily listing for <date>.
Message text 9:	The entire plant in Shop 3 was switched off on <date> at <time of day>.

Entry on the PG 685 for DB 3	Explanation
`: KS = '7:The motor went down';`	Message text number 7, separator ”:”, with spaceholder for time of day
`: S = 'at ”T”$';`	Message text, end-of-text character
`: KS = '8:This is the daily';`	Message text number 8, separator ”:”,
`: S = 'listing for "D".$';`	Message text, spaceholder for date, end-of-text character
`: KS = '9:The entire plant';`	Message text number 9, separator ”:”,
`: S = 'in Shop 3 was switched off';`	Message text with spaceholder for date and time of day
`: KS = 'on "D" at "T".$';`	Message text, end-of-text character
`:`	

6.4.4 Spaceholders for Control Parameters

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

- Entry with the ^ character (CTRL key)
- Entry with the spaceholder Sa, b
- Direct entry with data format KH

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: Selecting double-width type on the PT 88.
Double-width type is selected on the PT 88 printer with the ESC 8 command.
The "ESC" control character has the ASCII code $1B_H$. Find the ASCII character with the code $1B_H$ +40_H = $5B_H$ in the ASCII code table. It is the character "[".
Entry in message text: ^[8.

Explanation: The module detects 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 control character "ESC". The module instructs the printer to execute the "ESC 8" job, i.e. select double-width type.

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

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

Entry on the PG 685 for DB 5	Explanation

`:   KS = '50:^[8 Monthly overview^[<$';`
`:`
	Message text number 50, separator ":", select character string for double-width type, deselect character string for double-width type

Entry with the spaceholders "Sa, b, ..."

If you use the spaceholders "Sa, b", you must enter the control character in ASCII code in decimal form. If you want to enter several control characters consecutively, you must separate them with commas. You can enter a maximum of 50 control characters consecutively.

Example: Deselecting the subscript on the PT 88.
The subscript on the PT 88 printer is deselected with the "ESC16" command.
”ESC” corresponds to the ASCII code $1B_H$=27_D,
You must enter: ”S27,16”.

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

You want to store the following message text in DB 6:
Message text 60: The proportion of H<Select subscript>2<Deselect subscript>0 is 50 %

Entry on the PG 685 for DB 6	Explanation
`:   KS = '60:The proportion of';`	Message text number, separator ”:”, Message text,
`:   S  = 'H ″S27,18″2″S27,16″ O';`	Message text ”H”, spaceholder for on, Message text ”2”, spaceholder for off, Mesage text ”O”

`:   KS = 'is 50%.$';`
`:`
	Message text, end-of-text character

Direct entry with the "KH" format

The control parameters are entered direct with the "KH" format during configuration of a message text.

You must use the table to find the ASCII codes of the control parameters. Switch from the "KS" or "C" format to the "KH" format on the programmer and specify the control parameters in ASCII code. Then switch back to the "KS" or "C" data format.

Note:

If you use the "Print all configured message texts" job 8000$_H$ to print message texts in which control parameters have been entered direct, these 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 message text in DB 4:
Message text 40: Motor works <Line feed> <Carriage return> Newhaven <Line feed>
<Carriage return> 9999 Wackeldorf

Input at the PG 685 for DB 4	Explanation
`: KS = '40:Engine Plant';`	Message text no. 40, separator ":", message text
`: KH = 0A0D`	ASCII code for line feed (0A$_H$) and carriage return (0D$_H$)
`: S = 'Newhaven';`	Message text
`: KH = 0A0D`	ASCII code for line feed (0A$_H$) and carriage return (0D$_H$)

`: KS = '9999 Wackeldorf$';`
`:`
	Message text, end-of-message character

6.4.5 Spaceholders for Message Text

The spaceholder is replaced by a message text on printout. The number of the message text to be replaced can be specified in two ways:

- The number is specified when the message is configured
- The number is transferred to the CP as a variable in the user program.

Direct entry of the message text number "KPa"

”a” is the number of the message text to be inserted. You must specify "a" when configuring the message text. The values 1 to 4095 are allowed for "a". If the message text called contains spaceholders, these are treated as text and printed out.
You can use the "KPa" spaceholder several times in one message text. In this way, you can combine several message texts and print them out together.

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

You want to store the following message texts in DB 7:

Message text 70:	The following overview indicates <Insert message text 73>, <Insert message text 74> and <Insert message text 75>.
Message text 73:	the problems that have arisen during the monitoring period
Message text 74:	the countermeasures taken
Message text 75:	the length of the resulting downtimes

Entry on the PG 685 for DB 7	Explanation

`: KS = '70:The following over';`
`: S  = 'view indicates, ”KP73”,';`
	Message text number, separator ”:”, message text with spaceholder for "Insert message text 73"

`: KS = '”KP74” and ”KP75”.$';`	Message text , end-of-text character
`: S  = '73:the problems that';`	Message text number, separator ”:”, Message text
`: KS = 'have arisen during';`	Message text
`: S  = 'the monitoring period$';`	Message text , end-of-text character
`: KS = '74:the countermeasures';`	Message text number, separator ”:”, Message text
`: S  = 'taken$';`	Message text , end-of-text character
`: KS = '75:the length of the';`	Message text number, separator ”:”, Message text
`: S  = 'resulting';`	Message text

`: KS = 'down times$';`
`:`
	Message text , end-of-text character

Printout of message text 70:
The following overview indicates the problems that have arisen during the monitoring period, the measures taken and the length of the resulting downtimes.

Message text number as "KV" variable

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

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

You can use the "KV" spaceholder up to three times in one message text.

You can use the "KV" spaceholder in a message text simultaneously with spaceholders for variables. You must then make sure that you transfer the data in bytes 2 and 7 correctly.

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

You want to store the following message texts in DB 8:

Message text 80:	The "KV" has "KV" as a result of "KV".
Message text 81:	Motor
Message text 82:	Water loss
Message text 83:	Thermal problems
Message text 84:	Motor "KH" is down because of "KV" at "T".
Message text 85:	EMERGENCY OFF

Entry on the PG 685 for DB 8	Explanation

`:   KS = '80:The "KV" has "KV" as';`
`:   S  = 'a result of "KV".$';`
	Message text number, separator ”:”, Message text with spaceholder for message, end-of-text character

`:   KS = '81:Motor$';`	Message text number, separator ”:”, Message text, end-of-text character
`:   S  = '82:water loss$';`	Message text number, separator ”:”, Message text, end-of-text character
`:   KS = '83:thermal problems$';`	Message text number, separator ”:”, Message text, end-of-text character

`:   S  = '84:The Motor "KH" is';`
`:   KS = 'down because of "KV"';`
`:   S  = 'at ”T”$';`
	Message text number, separator ”:”, Message text with spaceholder for variable 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 message texts 81, 82 and 83 inserted:
The motor has thermal problems as a result of water loss.
Printout of message text 84 when the CPU transfers 20 as the variable value and message text 85 as the message text to be inserted:
Motor 20 is down because of EMERGENCY OFF at 12:00:00.

Maximum length of message texts

Messages may not exceed 136 characters in length. If you use space- holders, the printout may be longer than 136 characters. The maximum length of the printout is 250 characters.

Example: Configuring a message text with spaceholders for inserting message text.

You want to store the following text in DB 9:

Message text 90:	The following table privides a general overview of <Insert message text 91>, <Insert message text 92> and <Insert message text 93>.
Message text 91:	the faults occurring in the press controller with associated downtimes
Message text 92:	the resulting waiting times to be observed by the next press controller
Message text 93:	the average press controller downtime in the course of the last twelve months

Entry on the PG 685 for DB 9	Explanation

`: KS = '90:The following table';`
`: S  = ' provides a general';`
`: KS= ' overview of ″KP91″,';`
`: KS = ' ″KP92″' and ″KP93″. $ ';`
	Message text 90 with a length of 78 characters

`: KS = '91:the faults occurring in';`
`: S  = ' the press controller';`
`: KS = ' with associated downtim';`
`: S  = 'es $ ';`
	Message text 91 with a length of 71 characters

`: KS = '92:the resulting waiting';`
`: S  = ' times to be observed by';`
`: KS = ' the next press controll';`
`: S  = 'er $ ';`
	Message text 92 with a length of 72 characters

`: KS = '93:the average press';`
`: S  = ' controller downtime in';`
`: KS = ' the course of the last';`
`: KS = ' twelve months $ '; *`
`:`
	Message text 93 with a length of 88 characters

* ▒ This text is not printed out

Final printout:
The following table provides a general overview of the faults occurring in the press controller with associated downtimes, the resulting waiting times to be observed by the next press controller and the average press controller downtime (250 characters).

6.4.6 Spaceholders for Variables

The spaceholder is replaced by the variable at printout.
You must specify the variables in bytes 2 to 7 in the "Print message text" 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 spaceholder configured for variables in the message text is replaced by the variable 1, the second by variable 2 and the third by variable 3.

Note:

- If the message text contains more than three spaceholders for variables, they are not interpreted as spaceholders but printed out as text.
- If a spaceholder is wrongly configured, it is also treated as text and printed out.
- When spaceholders are replaced by the appropriate values, the total length may be greater than 136 characters.

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

Table 6-8. Data Formats for Variables

Data Format		Value Range	Number of Printed Positions
KM	Constant: Bit pattern	16 bits	16
KH	Constant: Hexadecimal pattern	0000$_H$ to FFFF$_H$	4
KC/KS	Constant: Alphanumeric characters	20$_H$ to 7F$_H$ / 20$_H$ to 7F$_H$	2
KT	Constant: Time base	1.0 to 999.3 *	4
KZ	Constant: Count	0 to 999	3
KY	Constant: Byte, byte	0 to 255, 0 to 255	7
KB	Constant: Byte	0 to 255	3
KF	Constant: Fixed-point number	-32768 to+32767	6
KFa,b KFa.b	Constant: Fixed-point number a Number of characters to be printed (9) b Decimal places	-32768 to+32767	Configurable (9)
KG KG.	Constant: Floating-point number	-1,7E38 to -1,4E-37 1,4E37 to 1,7E-38	13
KGa,b KGa.b	Constant: Floating-point number a Number of characters to be printed (9) b Decimal places (6)	-999 999 to 999 999 0.000001 to 9999999	Configurable (9)

* The time is printed out in seconds ("s").

"KB" data format

The CP evaluates the low-order byte of a 16-bit variable as the value for the variable in the "Constant byte" KB format:

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

"KT" data format

You can use the "KT" format to insert the values of internal timers in a message text. To do so, use LD TX to load the time in BCD into the accumulator and then transfer it to the CP. Four digits are always printed out without the seconds unit.

Table 6-9. Typical Printout of the "KT" Data Format

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)

"KF" data format

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

Table 6-10. Typical Printout of the "KF" Data Format

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

(= Space)

"KFa.b" data format

You can set the following parameters:

- Specify the total number of characters to be printed with "a".
  The sign and the decimal point each count as one character.
- Specify the number of places afer the point with "b". This is the same as correcting the variable by a factor of 10^{-b}.
- In the case of data format "KFa,b", a comma (between thousands) is printed out and, in the case of "KFa.b", a decimal point.

Table 6-11. Value Ranges for the Number of Characters to be Printed and Places After the Point ("KFa.b" Format)

Character to be Printed "a"	1 to 9	3 to 9	4 to 9	5 to 9	6 to 9	7 to 9
Decimal Places "b"	0	1	2	3	4	5

Errors occurring when setting the "KFa,b" or "KFa.b" format parameters have the following effects:

- If you specify values outside the value range for "a" and "b", the spaceholder is treated as text and printed as configured.
- If you configure "a" with a value too small for the variable values to be printed, "?" characters will be printed instead of the variable value.

Table 6-12. Typical "KFa.b" Data Format Printouts

Configu-ration	Printout of the KF = Variable							
	–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)

"KG" and "KGa.b" data formats

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

Note:

In the S5-135U, floating-point numbers have only a 16-bit mantissa. Bits 0 to 7 (byte n + 3) are always 0.

Floating-point numbers can be used in the S5-115U if the standard function blocks for floating-point arithmetic are used.

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

- In bytes 2 to 5 if the spaceholder is the only one or the first one in the message text.
- In bytes 4 to 7 if the spaceholder is the second one in the message text.

In addition, the memory requirement of four bytes has the following effects:

- Only one spaceholder per message text can be initialized in KG or KGa.b for "Insert floating-point number". A second spaceholder of this type would be interpreted as text and printed as configured.
- In addition to a spaceholder for "Insert floating-point number", only one further "KV" or "Insert variable" spaceholder can be configured per message text.

"KG", "KG." and "KG,", data format

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

"KGa.b" data format

You can set the following parameters:

- Specify the total number of characters to be printed with "a".
  The sign and the decimal point each count as one character.
- Configure the number of places after the point with "b".
  The variable value is not changed here, in contrast to the "KFa.b" data format.
- In the case of data format "KFa,b", a comma (between thousands) is printed out and, in the case of "KFa.b", a decimal point.

Table 6-13. Value Ranges for the Number of Characters to be Printed and Decimal Places

Characters to be Printed "a"	1...9	3...9	4...9	5...9	6...9	7...9	8...9
Decimal Places "b"	0	1	2	3	4	5	6

Errors occurring when configuring the "KGa,b" format have the following effects:

- If you specify values for "a" and "b" outside the value range, the spaceholder is treated as text and printed as configured.
- If you configure "a" with a value too small for the variable values to be printed, "?" characters will be printed instead of the variable value.

Table 6-14. Typical "KGa.b" Data Format Printouts

Number to be Represented	Configuration KG9,0	KG9,4	KG9,6	KG6,3	KG5,4
3,141593E+01	31	31,4159	31,415930	31,416	'KG 5,4'

Number to be Represented	Configuration KG4,1	KG3,1	KG1,0
3,141593E+01	31,4	31	?

6.5 Processing Job Requests in Print Mode

The CP 523 handles data transfer with the printer autonomously. Data transfer between the CPU and the CP 523 must always be initiated by the CPU. The CP 523 has an eight-byte transfer memory, which can be used by the CPU and the CP 523.

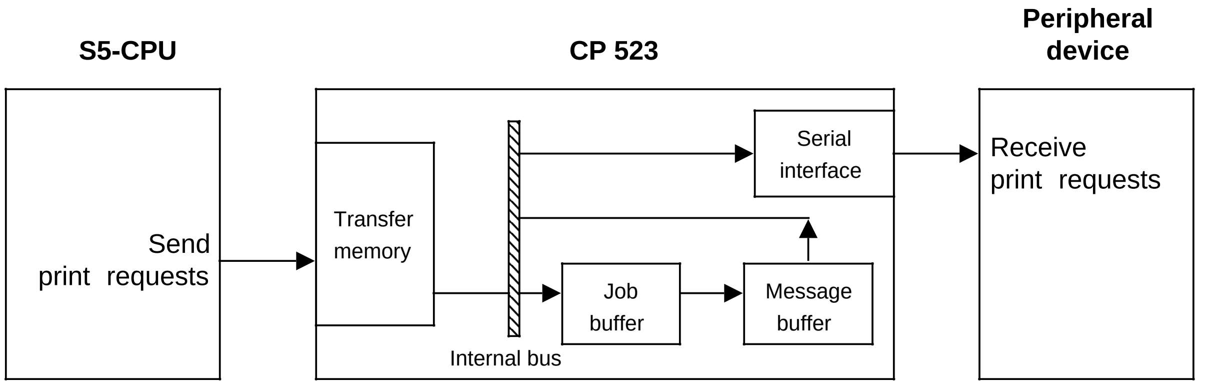


Figure 6-6. Schematic Representation of Job Order Processing in Print Mode

The CPU transfers the following in the transfer memory

• Word 0:	the job request, e.g.: "Print message text No. 20"
• Words 2,4,6:	further necessary or possible information on the job, e.g.: value of a variable to be inserted in the message text at printout.

The CP 523 writes the following to the transfer memory after a job has been executed:

• Byte 0:	Status of the module
• Byte 1:	Status of the printer and current day
• Byte 2 to 7:	The remaining current clock data

Access to the CP 523 transfer memory

The CPU can read the transfer memory at any time with L PW statements and write to it with T PW statements.

The CP 523 must be able to detect the instant at which the data transferred by the CPU is valid. Data in the transfer memory is recognized as valid by the CP 523 whenever the CPU transfers a (print) request in word 0 of the transfer memory. As soon as word 0 has been written, the CP 523 evaluates all the transfer memory data and overwites it with the current data (words 0 to 6).

This has the following consequences for the order in which data is transferred from the CPU to the transfer memory:

- First, words 2, 4 and 6 in the transfer memory must be written if the job request requires it. If you want to print a message text into which three variables are to be inserted, for example, the current values must be transferred to words 2, 4 and 6.
- Finally, the job number must be written into word 0 of the transfer memory (special feature of the CPU 944   5.2).

If you do not adhere to this order, the new job request will be executed with the wrong values.

+ Function of the job buffer

The CPU stores print requests in the job buffer immediately without checking them (a full eight bytes from the transfer memory in each case). The CP then empties the job buffer into the message buffer. It can happen that the CPU updates the transfer memory faster than the CP writes requests from the job buffer into the message buffer. The number of job requests in the job buffer increases as a result of this. The job buffer can hold a maximum of 31 job requests. Additional requests cannot be accepted. The CP reports "Job buffer full" (08_H) in the transfer memory. This can happen if you send more than 31 print requests consecutively to the CP.

If you nevertheless want to send more than 31 print requests, you can proceed in one of the following ways:

- Scan the status of the module for "Job buffer empty?" (03_H) after 31 print requests. If the CP stores this response (00_H) in the status byte, you can send a further 30 requests.

- If the "Entries already in the job buffer" message (03_H) appears, you can issue at least one further print request.

- Scan the status of the module for "Job buffer full?" (08_H) after 31 print requests. If the CP stores this response in the status byte, the job request must be reissued.

Function of the message buffer

The CP 523 is equipped with a message buffer. Job requests to the printer (print requests) are stored in this message buffer. The CP forwards the print requests to the printer from the message buffer. The print requests are sent in the same order as they are received from the CPU (FIFO).

The print request is deleted in the message buffer once the CP 523 has serviced it. If the CPU sends a large number of print requests within a short time, the number of pending requests will increase. If there are 255 job requests waiting to be serviced and the CPU sends another request, the CP 523 cannot process this last request. In this case, the CP 523 flags "Job/message buffer full" (08_H) to the CPU in byte 0 "Status of the module".

The CP reports "Job / message buffer full" 08_H in the following two cases:

1. The job buffer is full because more than 31 print requests have been sent consecutively from the CPU to the CP (CPU faster than CP).

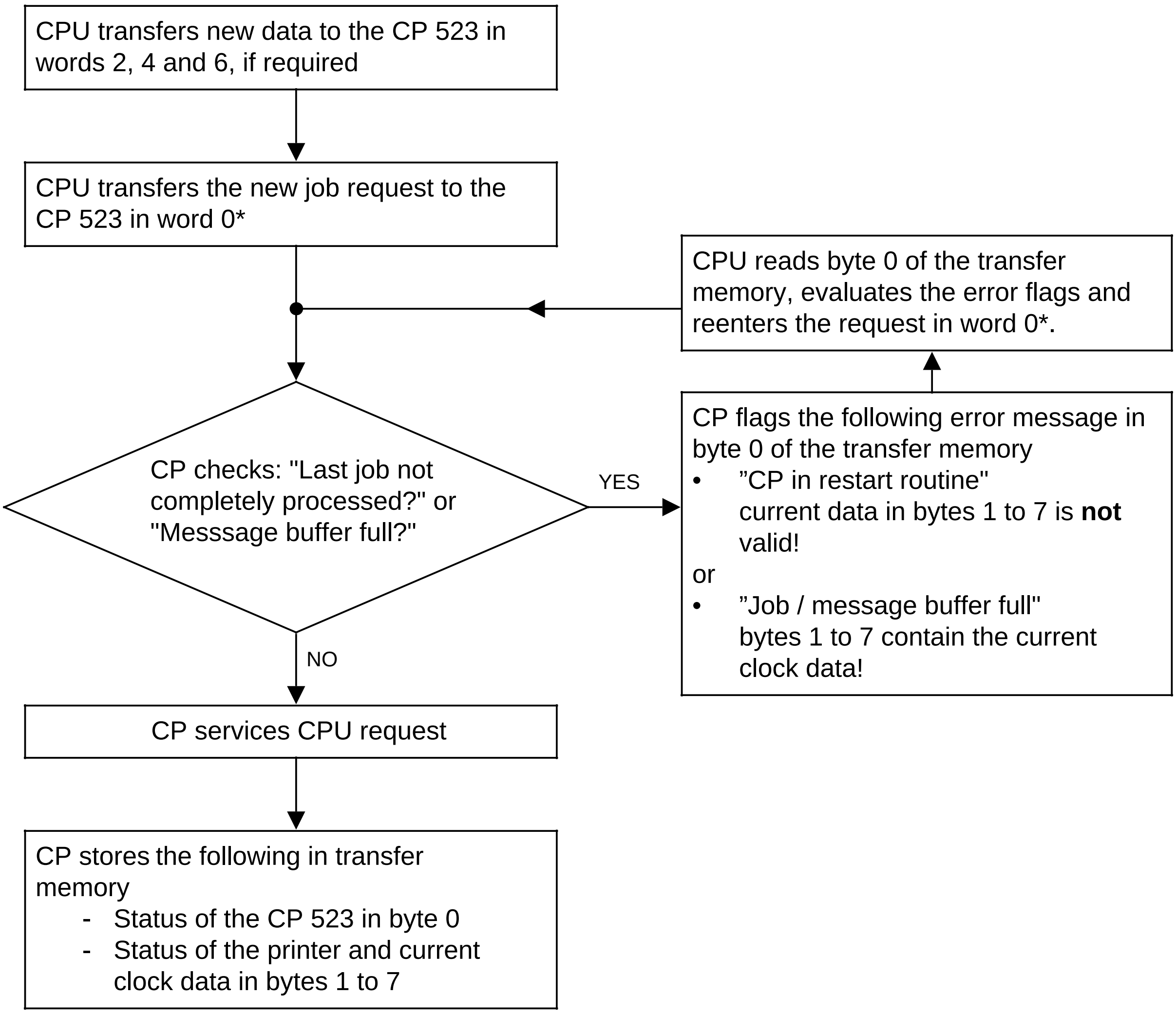
2. The message buffer is full because more than 255 print requests are waiting to be serviced (CPU and CP faster than printer).

Note:

If you have to issue a large number of print requests within a short period, check to make sure that the message buffer still has capacity.

- See 6.6 for a description of how to proceed when checking the message buffer.

The following is a schematic of job request processing. Entry of the print request in the message buffer is not shown in the figure. It is only important for the user to know if a job request has been accepted by the CP 523 and will be serviced.


CPU transfers new data to the CP 523 in words 2, 4 and 6, if required
CPU transfers the new job request to the CP 523 in word 0*
CPU reads byte 0 of the transfer memory, evaluates the error flags and reenters the request in word 0*.
CP checks: "Last job not completely processed?" or "Messsage buffer full?"
YES
CP flags the following error message in byte 0 of the transfer memory
• ”CP in restart routine" current data in bytes 1 to 7 is **not** valid!
or
• ”Job / message buffer full" bytes 1 to 7 contain the current clock data!
NO
CP services CPU request
CP stores the following in transfer memory
- Status of the CP 523 in byte 0
- Status of the printer and current clock data in bytes 1 to 7

***** 5.2

Figure 6-7. Schematic Representation of Job Request Processing

6.6 Status of the CP 523 and the Printer, and Reading the Current Clock Data

When you have written to word 0 of the transfer memory, the module makes the following current data available in the transfer memory which you can read with direct peripheral access:

- The status of the module in byte 0
- The status of the printer and the current day in byte 1
- The remaining current clock data in bytes 2 to 7.

Reading the status of the module in Print mode (byte 0)

The "Status of the module" byte 0 is divided into two half-bytes. Each half-byte is independent of the other. They can be combined in any way.

Table 6-15. Status of the Module in Print Mode (Byte 0)

Bit		Status	Detailed Explanation
4 to 7	0 to 3		
0	0	No error	Job buffer empty
X	1	Memory submodule defective	
X	2	No text	No message texts have been configured on the memory submodule
X	3	Entries in job buffer	
X	7	No backup battery	The battery in the power supply module is - either not inserted - or defective
X	8	Job / message buffer full	The module cannot process any further print requests at present. The job request must be repeated.
0	F	CP in restart routine	Message only occurs at startup: The clock data is invalid and the module cannot accept any job request.
1	X	Clock defective	Replace module
2	X	Default time set	The clock has been set with the values Sunday, 1.1.90, 12:00:00.
3	X	Time of day / date error	At least one setting is outside the permissible range. The clock has not accepted the new clock data and is continuing with the current data.
4	X	Illegal job request	You have issued a request to the CP which is not permissible in Print mode (6.7).
8	X	Hardware fault	Replace module

X= Signal state irrelevant for other half-byte

Example: Scanning the status of the module

The module is set to start address 128.
An unconfigured memory submodule is plugged into the module. When the error is detected, output 4.0 is to be set.

STL FB 100	Explanation
`NAME :ERROR1`	
`:L KH 0000`	Transfer nonrelevant data to the CP. The CP then updates the transfer memory with current data.
`:T PW 128*`	
`:L KH 000F`	Evaluation of "Status of the module" byte.
`:L PY 128`	Compare status byte with 0002_H.
`:AW`	
`:L KH 0002`	
`:><F`	If not equal,
`:BEC`	block End
`:S Q 4.0`	Otherwise, set output 4.0
`:BE`	Block End.

* 5.2

Status of the printer and day (byte 1)

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

Table 6-16. Status of the Printer and Day of the Week

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

X= Signal state irrelvant for other half-byte

Example: Evaluating the status of the printer

The module is set to start address 128.
If the printer is not ready, output 4.1 is to be set.

STL FB 101	Explanation
`NAME :ERROR2`	
`:L KH 0000`	Transfer nonrelevant data to the CP. The CP then updates the
`:T PW 128*`	transfer memory with current data.
`:L KH 00F0`	Evaluation of "Status of the printer" byte.
`:L PY 129`	Compare status byte with 10_H.
`:AW`	If not equal,
`:L KH 0010`	block End
`:><F`	Otherwise, set output 4.1.
`:BEC`	Block end.
`:S Q 4.1`	
`:BE`	

* 5.2

Date and time of day (bytes 1 to 7)

The CP 523 supplies the current clock data in BCD.
The status of the printer is specified in byte 1, bit 4. This information is independent of the current day of the week, which is specified in byte 1, bits 0 to 3.

Table 6-17. Status of the Printer, Date and Time of Day

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

Example: Reading the current clock data

The module is set with to start address 128.
The clock data is to be output on digital output modules beginning address 8.

STL FB 102	Explanation

`NAME :CLOCKDAT`
`:L KH 0000`
`:T PW 128*`
`:L PY 129`
`:T QB 8`
`:L PW 130`
`:T QW 10`
`:L PW 132`
`:T QW 12`
`:L PW 134`
`:T QW 14`
`:BE`
	Load nonrelevant data into accumulator 1 and transfer it to word 0 of the CP so that the CP transfers current data to the transfer memory.
	Read current clock data and transfer it to digital output modules.

* 5.2

Note:

You can also evaluate the following in connection with the integral clock:

- Module in restart routine (XF_H)
  The clock data is invalid.
- Default time of day set ($2X_H$)
  This evaluation is relevant if you are operating your programmable controller without battery backup.
- Time of day or date error ($3X_H$)
  This error evaluation is relevant after you have set the clock.
- Clock fault ($1X_H$)
  This error evaluation is only relevant after a restart.

6.7 Overview of Permissible Jobs in Print Mode

The CPU transfers the job request to the CP 523 in word 0. The job number is stored in byte 0.

Table 6-18. Permissible Job Request to the CP 523 in Print Mode

Byte 0		Byte 1		Job Request
7 6 5 4	3 2 1 0	7 6 5 4	3 2 1 0	
0	0	0	0	Read status byte, status of the printer and current clock data
$\boxed{0}$	Message text number			Print message text with CR/LF at end
1	0	Weekday setting		Set clock (settings in bytes 1 to 7)
$\boxed{2}$	0	Page number		Set page number
$\boxed{3}$	Message text number			Print message text with CR/LF at end
$\boxed{4}$	Message text number			Print message text without CR/LF at end
$\boxed{5}$	0	0	0	Execute form feed
$\boxed{6}$	0	0	0	Execute line feed
7	0	0	0	Clear message buffer
8	0	0	0	Print all messages
				Transfer parameter setting data
9	0	0	0	- Set serial interface parameters
9	0	2	0	- Set XON/XOFF character parameters
9	0	7	0	- Select Print mode

□ = Print requests. These are written to the message buffer as required.

Printing out message texts with and without CR/LF at the end (job numbers 0, 3 and 4)

You must specify the message text number in binary code in byte 0, bit 3 and in byte 1.
You can print more than one message text on one line with "Print message text without CR/LF". If your configured message texts contain no variables, for example, you can use this job order to generate tables.

Note:

The CP 523 counts the lines on a page on the basis of the line feeds sent to the printer.
If you want to print 15 message texts without CR/LF, for example, the CP counts 15 lines. The CP does not count the additional lines printed in those cases where a message text is longer than one line and the printer has consequently inserted an automatic line feed.

Additional information for "Print message text" (job numbers 0, 3 and 4)

The following can be additionally transferred to the CP 523 in words 2, 4 and 6 for "Print message text" (job numbers 0, 3 and 4):

- Values of variables
  When printing the message text, the value transferred to the CP 523 is inserted instead of the first spaceholder for variables configured in the message text.
- Number of a message text to be inserted into a message text
  When printing a message text, the message text with the number transferred to the CP 523 is inserted in place of the spaceholder for message texts.

Table 6-19. Transferring Additional Information for "Print Message Text"

Word (Byte)	Meaning	Assignment
2 (2+3)	Value of the variables or number of the message text to be inserted	0000_H to $FFFF_H$ (Dependent on the configured data format) 0000_H to $0FFF_H$
4 (4+5)	Value of the variables or number of the message text to be inserted	0000_H to $FFFF_H$ (Dependent on the configured data format) 0000_H to $0FFF_H$
6 (6+7)	Value of the variables or number of the message text to be inserted	0000_H to $FFFF_H$ (Dependent on the configured data format) 0000_H to $0FFF_H$

Setting the time of day (job number 10_H)

Specify the variables in BCD in bytes 1 to 7. Specify weekdays as digits between 1 and 7 (1 = Sunday, 2 = Monday, ..., 7 = Saturday).
If an item of clock data is to remain unchanged, enter the value FFH in the relevant byte.
If a variable is outside the permissible range, the clock will not be set.
The CP flags the "Time of day/date error".

Table 6-20. Transferring Additional Information for "Print Message Text"

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

* Bit 7 = 1 in the case of the 12h clock

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

Execute form feed (job number 5000_H)
If the "Output line feed" request 6000_H has not yet been serviced, it is cancelled by this request.

Execute line feed (job number 6000_H)
This job request generates a blank line. A subsequent "Execute form feed" request 5000_H cancels this request if it has not yet been serviced.

Clear message buffer (job number 7000_H)
All print requests stored in the message buffer are deleted.

Print all messages (job number 8000_H)
The message texts are printed as configured, with the message text number at the beginning of the line.

Transfer parameter setting data (job number 90_H)
On restart, the CP 523 accepts the data in DB 1 as you configured it on the memory submodule. You can use this job request to change part of the parameter setting data on the CP 523. You can change the data in parameter blocks 0, 2 and 7. The contents of the memory submodule remain unchanged.

- Job number 9000_H (parameter block 0 of DB 1 on the memory submodule)
  - Parameters for the serial interface (changing the baud rate, for example)
- Job number 9020_H (parameter block 2 of DB 1 on the memory submodule)
  - Parameters for the "XON" and "XOFF" protocols
- Job number 9070_H, (parameter block 7 of DB 1 on the memory submodule)
  Setting the Print mode with job number 9070_H.

Note:
The data transferred to the CP with the "Transfer parameter setting data " request has priority over the data stored on the memory submodule in DB 1. The parameter setting data on the memory submodule is valid after power-up, provided it has not been overwritten by the user program.

Illegal job orders in Print mode
If you write a job number into word 0 of the transfer memory other than the numbers listed above, the CP flags "Illegal job request" ($4X_H$) in the status byte. The "Coordinate data transfer" request $A001_H$, for example, is permissible in Communications mode but not in Print mode. In Print mode, this request would result in error flag $4X_H$ in the status byte.

This error will also be flagged if you transfer illegal parameter setting data with "Transfer parameter setting data".

6.7.1 Print Message Text (Job Numbers 0, 3 and 4)

The CP 523 forwards message text XY to the printer when it receives a "Print out message text XY" request. You can specify whether the printer is to execute a line feed (LF) or carriage return (CR) after printing message text XY.

Prerequisites

- You must have configured the message text on the memory submodule. Assign a number to each message text when configuring it.
- You must have plugged the configured memory submodule into the CP 523.

Specify

- The message text number in binary in byte 0, bits 0 to 3 and in byte 1
- The job number in byte 0, bits 4 to 7
  - Job numbers 0 and 3: Print message text with CR/LF
  - Job number 4: Print message text without CR/LF

Additional specificaitons

You can store a further three 16-bit variables in bytes 2 to 7 if you have configured spaceholders for variables in the message text.
You must enter the data in the format in which you configured it in the message text on the memory submodule (6.4.5).
At printout, the spaceholders are replaced by the current variable values.

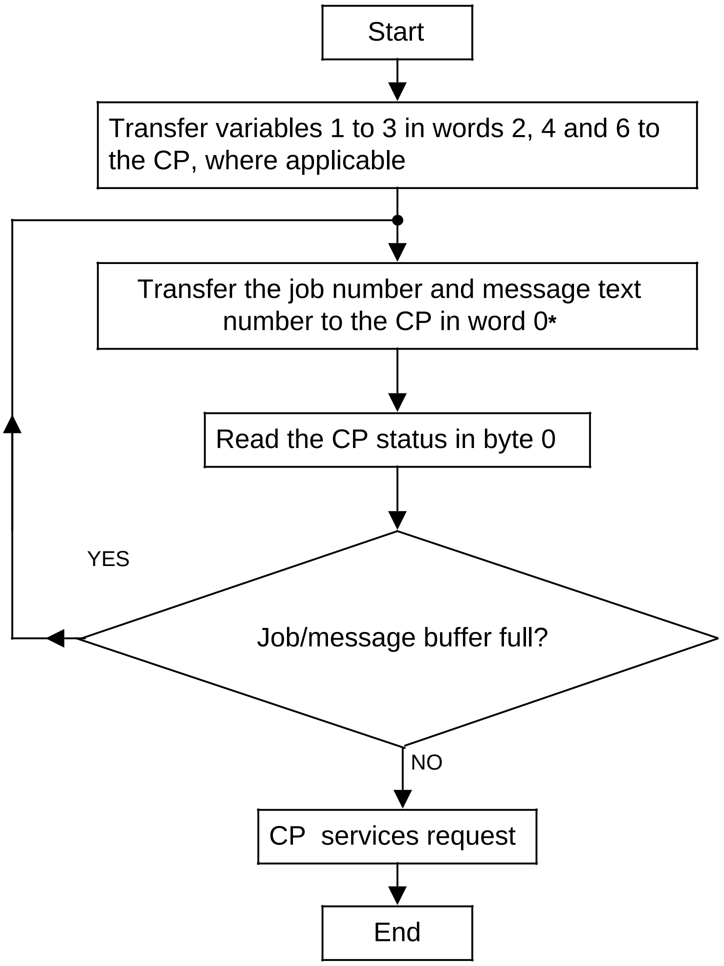
Table 6-21. Contents of the Transfer Memory for ”Print Message Text”

Byte	Bit	Meaning	Permissible Assignment
0	4 to 7	Job number	0_H = Print message text with CR/LF 3_H = Print message text with CR/LF 4_H = Print message text without CR/LF
	0 to 3	Message text number	001$_H$ to FFF$_H$ (1 to 4095)
1	0 to 7		
2	0 to 7	- Value of the variable or - number of the message text to be inserted	0000$_H$ to FFFF$_H$ Depending on the configured data format 001$_H$ to FFF$_H$ (1 to 4095)
3	0 to 7		
4	0 to 7	- Value of the variable or - number of the message text to be inserted	0000$_H$ to FFFF$_H$ Depending on the configured data format 001$_H$ to FFF$_H$ (1 to 4095)
5	0 to 7		
6	0 to 7	- Value of the variable or - number of the message text to be inserted	0000$_H$ to FFFF$_H$ Depending on the configured data format 001$_H$ to FFF$_H$ (1 to 4095)
7	0 to 7		

Permissible insertions when printing the message text
You can enter spaceholders in the message text when configuring message texts on the memory submodule. These spaceholders are replaced accordingly on printout.
You can enter spaceholders for the following:

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

Figure 6-8. is a schematic representation of "Print message text". Prerequisite is that a configured memory submodule is plugged into the CP.


Start
Transfer variables 1 to 3 in words 2, 4 and 6 to the CP, where applicable
Transfer the job number and message text number to the CP in word 0*
Read the CP status in byte 0
YES
Job/message buffer full?
NO
CP services request
End

* 5.2

Figure 6-8. Schematic for "Print Message Text"

The CP 523 can report further errors to the CPU (6.6).
For example:

- Default clock time set
- Printer not ready
- No battery backup

Example: Initiation of one-off message text printout

Start address 128 is set on the CP 523. A memory submodule configured with message text 9 is plugged into the CP (6.4.3). Message text 9 is to be printed if flag 20.0 is set.
The following program prints message text 9 once.

STL FB 103	Explanation
NAME :TEXT1	
:AN F 20.0	The program is terminated if flag 20.0 is not set.
:BEC	
:L KF +9	Load job number 0 and message text 9 into accumulator 1 and
:T PW 128*	transfer in word 0 to the CP.
:R F 20.0	
:BE	Block end

* 5.2

Example: Using the "KV" spaceholder three times

You have configured the following message texts on the memory submodule (6.4.5):

Message text 80:	The "KV" has "KV" because of "KV".
Message text 81:	Motor
Message text 82:	Water loss
Message text 83:	Thermal problems

Start address 128 is set on the CP 523. You have programmed the following statements in the user program:

STL FB 105	Explanation
NAME :KVTEST1	
:L KF+81	Load ACCUM 1 with the message text number of the 1st message text
:T PW 130	to be inserted and transfer it to bytes 2 and 3 of the CP.
:L KF+82	Load ACCUM 1 with the message text number of the 2nd message text
:T PW 132	to be inserted and transfer it to bytes 4 and 5 of the CP.
:L KF+83	Load ACCUM 1 with the message text number of the 3rd message text
:T PW 134	to be inserted and transfer it to bytes 6 and 7 of the CP.
:L KF+80	Load ACCUM 1 with the message text number of the message text to
:T PW 128*	be inserted and the "Print message text" request and transfer to bytes
	0 and 1 of the CP.

* 5.2

Printout of message text 80 if message texts 81, 82 and 83 are inserted:
The motor has thermal problems because of water loss.

Example: Simultaneous use of the "KV" spaceholder, the "KH" spaceholder for variables and the spaceholder for the time of day.

You have configured the following message texts on the memory submodule (6.4.5):
Message text 84: The motor went down as a result of "KV" at "T".
Message text 85: EMERGENCY OFF

Motor 20 went down at 17:15:30. Start address 128 is set on the CP 523. You have programmed the following statements in the user program:

STL FB 106	Explanation
`NAME :KVTEST2`	

`:L  KH 0020`
`:T  PW 130`
	Load motor number 20 into ACCUM 1 and transfer to bytes 2 and 3 of the CP.

`:L  KF+85`
`:T  PW 132`
	Load the message text number of the message text to be inserted into ACCUM 1 and transfer to bytes 4 and 5 of the CP.

`:L  KF+84`
`:T  PW 128`
	Load ACCUM 1 with the message text number of the message text to be printed and the "Print message text" request and transfer to bytes 0 and 1 of the CP.

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

6.7.2 Setting the Time and Date (Job Number 10_{H})

Specify:

- The new settings in bytes 1 to 7
- Job number 10_{H} in byte 0

In each case, you must transfer the settings first and the "Set clock" request last to the CP 523.

Settings

You must note the following as regards the settings:

- The settings must be transferred to the CP 523 in BCD.
- If you enter "FF_{H}" in a setting, the current clock data is retained.
- Specify the days of the week as follows:
  01 = Sunday, 02 = Monday, 03 = Tuesday, 04 = Wednesday, 05 = Thursday, 06 = Friday, 07 = Saturday
- AM/PM flag (only in the case of the 12h clock)
  When specifying the "Hours" setting in byte 5, you must take account of the AM/PM flag (bit 7) in the case of the 12h clock: bit 7 = 1 p.m. and bit 7 =0 a.m.

Table 6-22. Contents of the Transfer Memory for "Set Clock" Job

Byte	Meaning	Value range
0	Job number for ”Set Clock”	10_{H}
1	Day of the week (01 = Sunday, 02 = Monday, ..., 07 = Saturday)	01_{BCD} . . .07_{BCD}
2	Day	01_{BCD} . . .31_{BCD}
3	Month	01_{BCD} . . .12_{BCD}
4	Year	00_{BCD} . . .99_{BCD}
5	Hours 24 h clock 12 h clock a. m. 12 h clock p. m.	 00_{BCD} . . .23_{BCD} 01_{BCD} . . .12_{BCD} 81_{BCD} . ..92_{BCD}
6	Minutes	00_{BCD} . . .59_{BCD}
7	Seconds	00_{BCD} . . .59_{BCD}

6.7.3 Setting the Page Number (Job Number 20_{H})

When configuring the memory submodule, you can define whether and where a page number is to be printed in parameter block 5. The default is pagination beginning with page number 1, printed at the bottom centre of the page.

Specify

- The page number in byte 1 (01_{H} to FF_{H})
- Job number 20_{H} in byte 0.

6.7.4 Form Feed (Job Number "5000_H")

When prompted by the CPU, the CP 523 instructs the printer to execute a form feed. A form feed is implemented by executing the number of line feeds configured (parameter block 5 "Page format": lines per page). Headers, footers and page numbers are also printed depending on the configuration.

Specify
Job number 5000_H in word 0.

Example: Form feed
Start address 128 is set for the module.
A from feed is to be executed when flag 25.0 is set

STL FB 107	Explanation
NAME :FORMFEED	
0005 :AN F 25.0	The program is terminated if flag 25.0 is not set.
0006 :BEC	
0007 :L KH 5000	Load ACCUM 1 with the job number for "Form feed" and transfer
0008 :T PW 128*	to the CP in word 0.
0009 :R F 25.0	Job accepted by CP. Reset F 25.0
000C :BE	Block End

* 5.2

6.7.5 Line Feed (Job Number "6000_H")

When prompted by the CPU, the CP 523 instructs the printer to execute a line feed.

Specify
Job number "6000_H" in word 0.

Example: Line feed
The module is set to start address 128.
A line feed is to be executed when flag 25.1 is set.

STL FB 108	Explanation
NAME :LINEFEED	
0005 :AN F 25.1	The program is terminated if flag 25.1 is not set.
0006 :BEC	
0007 :L KH 6000	Load ACCUM 1 with the job number for ”Line feed” and transfer
0008 :T PW 128*	to the CP in word 0.
0009 :R F 25.1	Job accepted by CP. Reset F 25.1
000C :BE	Block End

* 5.2

6.7.6 Clearing the Message Buffer (Job Number "7000$_H$")

When prompted by the CPU, the CP 523 cancels all CPU requests stored in the message buffer.

Specify
Job number "7000$_H$" in word 0.

Example: Clearing the message buffer with the Programmer's "FORCE VAR" function

You can force a peripheral (I/O) word (PW) using the programmer "FORCE VAR" function only when you have written it into a flag word (FW). For this purpose, store PW 128 in FW 128 of FY 10 (6.7.7). FW 128 can then be forced in the "FORCE VAR" screen form. FW 228 is an auxiliary word for old/new value comparison.
The module is set to start address 128.

Operand	Signal states	Meaning
PW 128	KH= 7000*	"Clear message buffer" request

* 5. 2

6.7.7 Printing All Configured Message Texts (Job Number "8000$_H$")

When prompted by the CPU, the CP 523 prints out all the configured message texts stored on the memory submodule. This job is used for defining and checking the message texts stored on the user submodule. Spaceholders are not evaluated.

Specify
Job number "8000$_H$" in word 0.

Example: Printing all configured messages with the programmer "FORCE VAR" function

You can force a peripheral (I/O) word (PW) using the programmer's "FORCE VAR" function only when you have written it into a flag word (FW). For this purpose, store PW 128 in FW 128 of FY 10 (6.7.7). FW 128 can then be forced in the "FORCE VAR" screen form. FW 228 is an auxiliary word for old/new value comparison.
The module is set to start address 128.

Operand	Signal States	Meaning
PW 128	KH= 8000*	"Print all configured messages" request

* 5.2

STL FB 10	Explanation
:	
:L FW 228	Load old value
:L FW 128	Load new value
:! = F	Compare for equal
:BEC	Block End if equal
:L FW 128	Update old value if not equal
:T FW 228	
:L FY 128	
:T PY 128	transfer flag word byte by byte
:L FY 129	to the peripheral word
:T PY 129	
:BE	Block end

6.7.8 Transferring the Parameter Setting Data (Job Number 90_H)

You can transfer data for a parameter block to the CP 523 with a "Transfer parameter setting data" request:

- Data for the serial interface (corresponds to parameter block 0 on the memory submodule)
- Characters for XON/XOFF protocol (corresponds to parameter block 2 on the memory submodule)
- Setting the mode and the parameters (corresponds to parameter block 7 on the memory submodule)

Note:

"Transfer parameter setting data" job is almost without significance in Print mode since the parameter setting data for Print mode are stored in DB 1 on the memory submodule.
A typical application might be if you want to replace a defective printer temporarily with another printer whose interface has been configured differently.
The job does not change the contents of the memory submodule plugged into the CP 523.
After a warm restart, the CP continues to use the data on the memory submodule.
See 7.3.2 for a detailed description.

Figures

Tables

Tables

7 Communications Mode

In Communications mode, the CP 523 permits transfer of a frame of up to 256 bytes between the CPU and a peripheral device connected to the CP 523 in response to a CPU job request:

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

7.1 General

You can choose between the following:

- Transparent mode
  The CP 523 does not interpret any characters in Transparent mode.
  - No XON/XOFF protocol is possible.
  - Only fixed-length message frames can be received. Exception: If you program very skillfully, you can also receive variable-length frames by evaluating ”ZVZ” (i.e. character time-out) (7.4.2 under the heading ”character time-out”).
- Interpretive mode
  The CP 523 evaluates the following characters in Interpretive mode:
  - RUB OUT (7F$_H$)
  - BACKSPACE (08$_H$)
  - XON / OFF character (if programmed)
  - 1 or 2 end characters
- 3964(R) mode
  The CP 523 interprets the following characters in 3964(R) mode:
  - DLE (10$_H$)
  - STX (02$_H$)
  - NAK (15$_H$)
  - ETX (03$_H$)

  During the parameter setting phase, you can also specify whether the data frames are to be transferred with or without a block check character. The block check character (BBC) increases transmission reliability. 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

The CP 523 handles data transfer with the peripheral device autonomously.
The CPU initiates data exchange between the CPU and the CP by sending a job request. See 7.6 and 7.7 for a detailed description of the data exchange procedure.

See Chapter 8 for a description of function blocks which allow user-friendly handling and control of data exchange.

The time of day can be read from the integral clock by the CPU in Communications mode and used in the application program for date-dependent and time-dependent tasks.

Message text printout and editing as in Print mode is not possible in Communications mode. For this reason, no memory submodule is required in Communications mode.

7.1.1 3964 and 3964R Protocols (relevant in 3964(R) mode only)

Protocol data

The 3964 and 3964R protocols control the flow of data between your programmable controller and the partner in the communications link.

The data to be transmitted must be entered in the CP 523's output buffer, and is then forwarded to the partner in the link 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 parameter in the link is entered in input buffers. If the data is received without error, it can be fetched by the CPU for post-processing.

The 3964 and 3964R protocols are asynchronous, bit-serial transmission procedures. All parameters on the module and on the partner in the communications link, except for the priority, must be identical.

Control information and useful data are transmitted over the connecting cables. 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 is initialized in the parameter block.

Sample character frame:

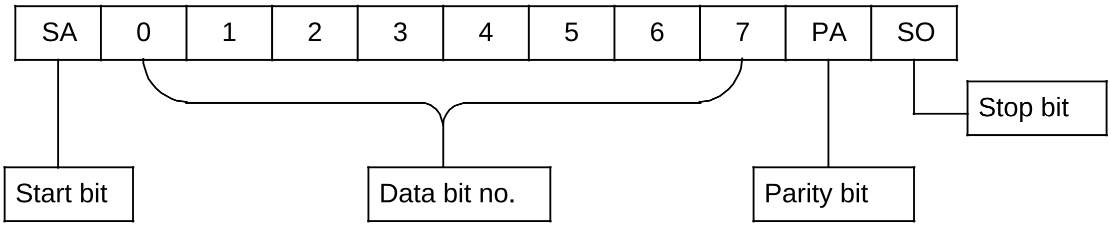


Figure 7-1. Example of an 11-Bit Character Frame

When the **3964R protocol** is used, a **block check character (BCC)** is transmitted at the end of each data frame to increase transmission reliability and ensure data integrity. The BCC is the even longitudinal parity (EXORing of all data bits) of the frame transmitted or received. Its generation begins with the first useful data byte (1st byte in the frame) following connection buildup, and ends on connection cleardown after the DLE ETX character.

Sending

In order to **build up a connection**, the CP transmits the **STX control character** (start of text). If the partner in the link replies with DLE (data link escape) before time-out (QVZ)*, the protocol goes to Send mode. If the partner responds with NAK (negative acknowledgement) or a random character (other than DLE), or if no response comes before time-out, 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.

***** Default value in 3964(R) mode: 2000 ms

****** Default: other values can be entered in parameter block 7

If the connection can be established, the useful data in the output buffer is transmitted to the partner in the link at the specified baud rate. The partner in the link monitors the interval between incoming characters, which may not exceed the **character delay time (ZVZ)***.

Each DLE character found in the buffer is transmitted as two DLE characters (**double DLE**), i.e. 10_H is transmitted twice for each DLE character in the output buffer.

When the buffer is empty, the CP appends

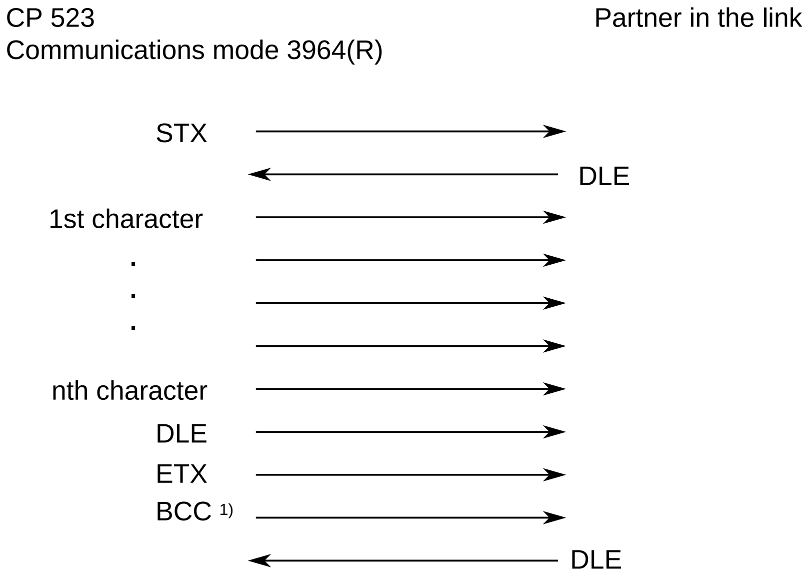
DLE ETX	in **3964 mode** and
DLE ETX BCC	in **3964R mode**

as end-of-text identifier and waits for an acknowledgement. A DLE character from the partner within the allotted time (QVZ) indicates that the data was received without error.

If the partner in the link 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, enters an error flag in the CBS, and transmits NAK to the partner in the data link.

If the partner in the link sends a NAK character during an in-progress 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 time (ZVZ) has expired and then sends NAK to bring the partner to the idle state before once again transmitting STX.

Example of error-free transmission:



1) Only in 3964R mode

Figure 7-2. Error-Free Data Interchange (Send)

* Default value in 3964(R) mode: 220 ms
Value can be initialized in parameter block 7

** Default value: Other values can be initialized in parameter block 7

Receiving

When no Send request is pending in 3964(R) mode, the CP waits for the link partner to establish a connection.

If the CP receives a character (other than STX) while in the idle state, it waits until the character delay time has expired (ZVZ) and then sends NAK.

If the CP receives an STX from the partner and the input buffer is not full, it replies 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 partner in the link.

How the CP terminates reception depends on whether it is set for 3964 or 3964R mode:

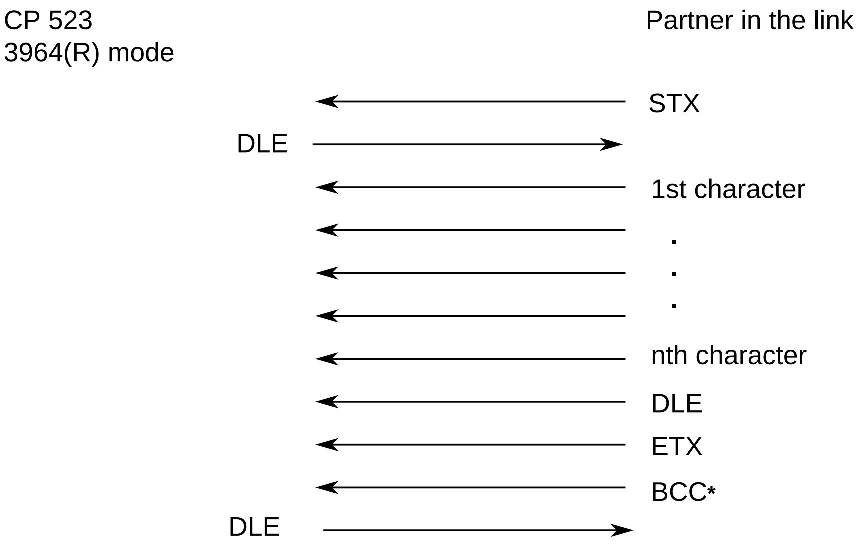
3964 mode:
If the CP detects a DLE ETX string, it terminates reception and sends DLE (if the message frame was received without error) or NAK (if it was not) to the link partner.

3964R mode:
The CP terminated reception when it receives DLE ETX BCC. The CP then compares the block check character with the longitudinal parity it has generated internally.
If the BCC is correct and no other receive error occured, the CP transmits DLE. If the BCC is incorrect, the CP sends NAK and waits for a retry. If the frame cannot be received without error after a total of six attempts* or if the link partner does not initiate a retry with the block (frame) delay time of 4 s*, the CP aborts reception.

* Default value: Other values can be initialized in parameter block 7

If transmission errors occur during reception (characters are lost, framing error, parity error), the CP continues receiving until the connection is cleared down, transmits NAK to the link partner, and waits for a retry as described above.

Example of error-free data interchange:



*) BCC in3964**R** mode only

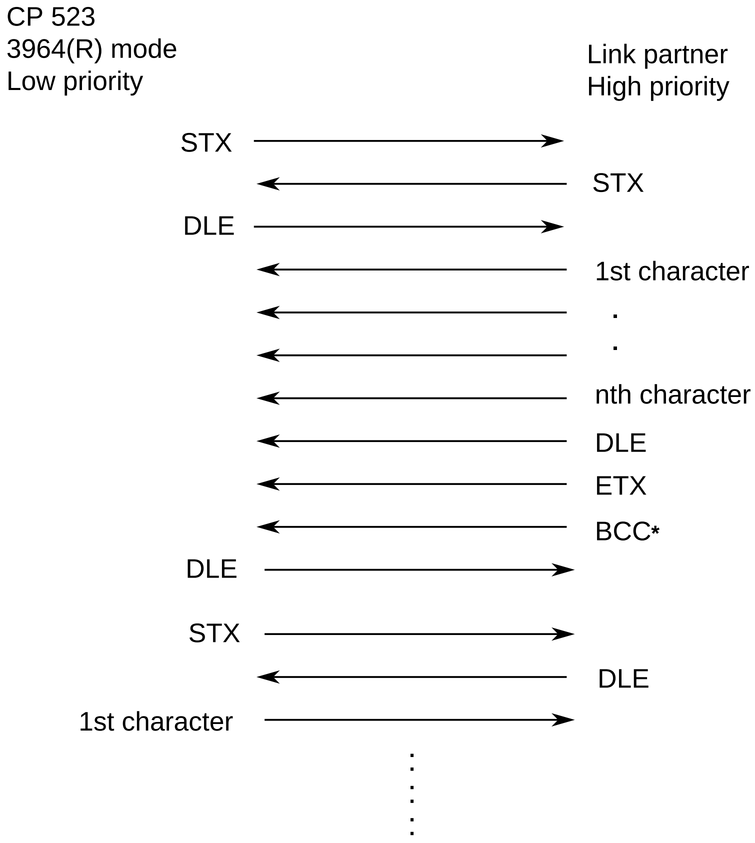
Figure 7-3. Error-Free Data Interchange (Receive)

Initiation conflict

If one partner in the link replies to a line bid (STX character) from the other link partner 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 retracts 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 initializing 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:



***** BCC in 3964**R** mode only

Figure 7-4. Resolving an Initiation Conflict

Protocol Errors

The protocol can detect errors caused by incorrect behaviour on the part of the link partner and errors 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 goes to the idle state.

7.2 Prerequisites for Operation in Communications Mode

The following conditions must be met for operating the CP 523:

1. Settings on the peripheral device

The settings on the peripheral device must agree with the parameter setting data for the CP 523. If your peripheral device is transmitting data at 2400 baud, for example, you must also set this baud rate for the CP 523.

Note:

- See 3.4.2 for configuration examples with terminal diagrams.
- See 2.6.2 for details of the serial interface.
- Setting the CP 523 parameters in Communications mode is explained in 7.3.

2. Setting the CP 523 parameters

There are two ways of setting the CP 523 parameters:

- Store the parameter setting data on a memory submodule in DB 1.
  Initialize the memory submodule with a programmer in offline mode
- Transfer the parameter setting data in the user program after power-up or recovery of power.
  The restart organization blocks are used for this purpose.
  No memory submodule is required.

You must specify the following parameter setting data:

- Parameters for the interface to the peripheral device (baud rate, type of interface, handshake mode ...)
- Specifications pertaining to the XON/XOFF protocol (optional)
- Parameter setting data for data transfer in Communications mode (message frame length, end-of-text character)
- Correction value for the integral clock (optional)

Note:

If the CP 523 and its communications partner have different restart characteristics, data may be lost during the restart phase. Check that your peripheral device is ready to receive before you send data.

3. Establishing connections

The programmable controller must be switched off to carry out the following measures:

- Install the CP 523 in the central controller or expansion unit.
- Establish the connection between the CP 523 and the peripheral device.
- If you have stored the parameter setting data in a memory submodule, plug the memory submodule into the CP 523.

You can then switch the PLC on.

4. CP 523 startup in Communications mode

After power recovery, the CP 523 is automatically in Transparent mode when:

- No memory submodule is plugged in.
- A memory submodule initialized for Transparent mode is plugged in.
- Transparent mode has been forwarded to the CP 523 from the user program.

You can set Interpretive or 3964(R) mode in the following ways:

- Transfer Interpretive or 3964(R) mode to the CP 523 in the user program
- Plug a memory submodule initialized for Interpretive or 3964(R) mode into the CP 523.

Note:
You can change modes during operation. To do so, you must issue the "Transfer parameter setting data" request to the CP 523, but remember that you may have to reconfigure the serial interface. The Send and Receive buffers are cleared.

7.3 Setting the CP 523 Parameters in Communications Mode

The CP 523 is supplied with default values for all parameters.

- Parameters for the interface to the peripheral device (baud rate, type of interface, handshake mode ...)
  The parameters must conform to the specifications and settings on/in the peripheral device.
- Specifications of the XON/XOFF protocol
- Parameter setting data for data transfer in Communications mode (message length, end-of-text character).
- Correction value for the accuracy of the integral clock
  The integral clock is set for an ambient temperature of 35 °C. If the CP 523 is used in other ambient temperatures, you can increase the accuracy of the clock by configuring a correction value.

If you want to initialize the CP 523 with different data, there are ways of transferring parameter setting data to the CP:

- Store the parameter setting data on a memory submodule in data block (DB) 1 and plug the initialized memory submodule into the CP 523 (7.3.1).
- Transfer the parameter setting data to the CP 523 in the user program. The restart organization blocks are used for this purpose (7.3.2). No memory submodule is required for operating the CP 523.

In order to transfer the parameter setting data as simply as possible, the data is divided into parameter blocks. This subdivision is the same as in Print mode. Parameter blocks 1, 3, 4, 5, 6 and 8 are only significant in Print mode and are not included in the table.

Table 7-1. Parameter Blocks in Communications Mode

Parameter Block No.	Contents
0	Serial interface parameters
2	XON/XOFF protocol (only relevant if XON/XOFF protocol is used in Interpretive mode)
7	Desired mode Parameter setting data for data transfer in Communications mode
9	Correction value for the integral clock

Reader's note: Refer to Section 7.3.1 or 7.3.2, depending on whether you want to set your CP 523 parameters via a memory submodule or in the user program:
Section 7.3.1: Setting the CP 523 parametersvia a memory submodule
Section 7.3.2: Setting the CP 523 parametersin the user program

7.3.1 Setting the CP 523 Parameters with the Memory Submodule

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

Note:
The procedure for setting the memory submodule parameters in Communications mode is almost identical to the procedure in Print mode. The only difference is in configuring parameter block 7, in which additional data must be specified in Communications mode.

Entering the parameter blocks

First, you must enter all the parameter blocks to be changed in DB 1 on the programmer. Then transfer DB 1 to the memory submodule.

Further tips for input:

- To make the programmer screen presentation clearer, alternate between the "KC" and "C" formats.
- You can enter comments in addition to the parameter blocks.

Schematic for entering parameter blocks

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

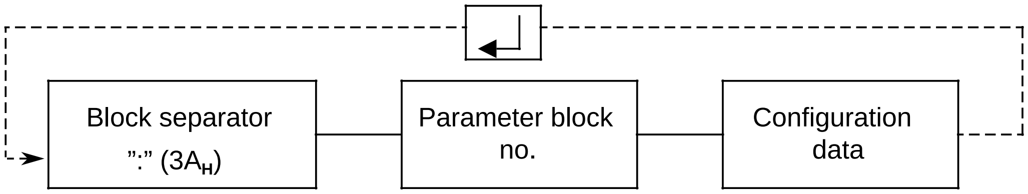


Figure 7-5. Schematic for Entering Parameter Blocks in DB 1

The following pages contain the parameter block assignments and an example for entering parameter setting data with the PG 685.

Note:
Parameter block assignment for setting parameters with the memory submodule corresponds to the assignment of bytes 2 to 7 of the transfer memory when setting parameters with the user program.

Table 7-2. Parameter Block Assignments on the Memory Submodule

Parameter Block	Meaning	Value Range	Default Values on the CP 523
0	Baud rate 110 Bd	1	4
	200 Bd	2	
	300 Bd	3	
	600 Bd	4	
	1200 Bd	5	
	2400 Bd	6	
	4800 Bd	7	
	9600 Bd	8	
	Parity even	0	0
	odd	1	
	”mark”	2	
	”space”	3	
	no parity check	4	
	BUSY signal no	0	0
	Interface TTY	0	0
	V.24	1	
	Data format: Parity:		
	11-bit character frame 7 data bits (yes)	0	0
	8 data bits (yes)	1	
	8 data bits (no)	2	
	10-bit character frame 7 data bits (no)	3	
	7 data bits (yes)	4	
	8 data bits (no)	5	
	HW handshaking OFF	0	0
	ON	1	
2 Significant only in Interpretive mode	XON character	01_H to $7E_H$	$FFFF_H$ (No XON/XOFF protocol)
	XOFF character	01_H to $7E_H$	

Table 7-2. Parameter Block Assignments on the Memory Submodule (Continued)

7	Mode Print mode Transparent mode Interpretive mode 3964(R) mode	 0 1 2 3	0 (for memory submodule configured with message texts) 1 (no memory submodule)
	Parameters for Transparent mode		
	Character delay time (1 to 3000) ·10 ms	0001_D to 3000_D	0001 6 10 ms
	Message frame length (1 to 256 bytes) One (byte 7) or two (bytes 6 and 7) end characters must be specified for a length of 0 bytes.	000_D to 256_D	64
	Parameters for Interpretive mode		
	Character delay time (1 to 3000) ·10 ms	0001_D to 3000_D	0001 6 10 ms
	Message frame length (0 byte) One (byte 7) or two (bytes 6 and 7) end characters must be specified for a length of 0 bytes.	000_D	
	End characters (only when frame length= 0) 1 end character 2 end character	 0001_H to $00FF_H$ 0101_H to $FFFF_H$	$0D_H$

Table 7-2. Parameter Block Assignments on the Memory Submodule (Continued)

7	**Parameters for 3964(R) mode**		
	Character delay time (ZVZ) (1 to 65535) ·10 ms	00001_D to 65535_D	22_D
	Time-out (QVZ) (1 to 65535) ·10 ms	00001_D to 65535_D	00200_D
	Block delay time (BWZ) (1 to 65535) ·10 ms	00001_D to 65535_D	00400_D
	Transfer **with** block check character (3964R) Transfer **without** block check character (3964)	1 0	0
	Priority Low High	0 1	1
	Connection buildup attempts	001_D to 255_D	6
	No. of send attempts	001_D to 255_D	6
9	Correction value (s/month)	- 400_D to+400_D	000_D

Explanation of data transmission parameters

Baud rate

You have a choice of eight baud rates. The default is 600 baud. If you are using the RS-232-C (V.24) interface, you must take the load capacitance of cables longer than 15 m/29 ft. into account. If you operate the CP 523 with the active TTY interface, cable lengths of up to 10 m/330 ft. are possible. Cable lengths of up to 1000 m/3280 ft. can be implemented using the passive TTY interface.
Longer cables can be used if the baud rate is reduced.

Parity

You have a choice of five types of parity.

- Even parity
  The parity bit is set when the number of data bits with a ”1” value is odd.
- Odd parity
  The parity bit is set when the number of data bits with a ”1” value is even.
- ”Mark”
  The parity bit is always "1".
- ”Space”
  The parity bit is always "0".
- No parity check
  The signal state of the parity bit is not significant. Parity is not checked when data is received but is always set to "1" when sending.

The default is even parity.

Interface

You can choose between the RS-232-C (V.24) interface or the TTY interface. (See 2.3 for the interface characteristics).
The default is the TTY interface.

Data format

Characters are transmitted between the CP and the peripheral device in a 10-bit or 11-bit character frame. You can choose between 7 and 8 data bits within the character frame:

10 -bit character frame

• 1 start bit, 7 data bits, 2 stop bits	Data format 3 in parameter block 0
• 1 start bit, 7 data bits, 1 parity bit, 1 stop bit	Data format 4 in parameter block 0
• 1 start bit, 8 data bits, 1 stop bit	Data format 5 in parameter block 0

11-bit character frame

• 1 start bit, 7 data bits, 1 parity bit, 2 stop bits	Data format 0 in parameter block 0
• 1 start bit, 8 data bits, 1 parity bit, 1 stop bit	Data format 1 in parameter block 0
• 1 start bit, 8 data bits, 2 stop bits	Data format 2 in parameter block 0

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

Hardware handshake (HW handshake)

This parameter is only significant for the RS-232-C (V.24) interface.

If you set HW handshaking ON, the "RTS", "CTS", "DTR" and "DSR" control signals of the RS-232-C (V.24) interface are evaluated. The XON/XOFF protocol is not possible.

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

The HW handshake signals are neither evaluated nor affected in 3964(R) mode.

Parameter for XON/XOFF character

If you use the XON/XOFF protocol, you have a free choice of the XON/XOFF character. ASCII code provides the codes 11_H (DC1) for the XON character and 13_H (DC3) for the XOFF character. You may not use the same values for the XON and XOFF characters.

Data transfer is executed with the XON/XOFF protocol when valid values are used for XON and XOFF.

XON/XOFF is not possible in Transparent mode (mode 1) or 3964(R) mode (mode 3).

Reader's note: Section 7.3.1 is arranged so that the parameter list (Table 7-2) and the parameter descriptions which follow it are succeeded by the subsections

- Initializing TRANSPARENT MODE (page 7-16)
- Initializing INTERPRETIVE MODE (page 7-18)
- Initializing 3964(R) mode (page 7-21)

When you have chosen the mode you want to use, you can limit your reading to the appropriate subsection.

The description of parameter block 9 applies for all modes, and was therefore presented only once on page 7-42.

Setting ”TRANSPARENT MODE” parameters via the MEMORY SUBMODULE

When you want to use Transparent mode to tranfer your datas, you must enter all parameter setting data in parameter blocks 0 and 7.

Table 7-3. Contents of the Parameter Blocks for Transparent Mode

Param-eter Block	Description	Value Range	Default Values on the CP 523
0	Baud rate 110 Bd 200 Bd 300 Bd 600 Bd 1200 Bd 2400 Bd 4800 Bd 9600 Bd	 1 2 3 4 5 6 7 8	4
	Parity even odd ”mark” ”space” no parity check	 0 1 2 3 4	0
	BUSY signal no	0	0
	Interface TTY V.24	 0 1	0
	Data format: Parity: 11-bit character frame 7 data bits (yes) 8 data bits (yes) 8 data bits (no) 10-bit character frame 7 data bits (no) 7 data bits (yes) 8 data bits (no)	 0 1 2 3 4 5	0
	HW handshaking OFF ON	 0 1	0
7	Mode Printer mode Transparent mode Interpretive mode 3964(R) mode	 0 1 2 3	0 (for memory sub-module with configured message texts) 1 (no memory submodule)
	Character delay time (1 to 3000) ·10 ms	0001_D to 3000_D	0001 6 10 ms
	Message frame length (1 to 256 bytes)	001_D to 256_D	64

Example: Setting parameters on the MEMORY SUBMODULE for ”TRANSPARENT MODE” with the PG 685

Start address 128 is set on the CP 523. The module is to be operated in ”Transparent mode”.
Parameters are set as follows:

- Parameters for the serial interface (parameter block 0)
  - 2400 Bd (6)
  - Even parity (0)
  - No BUSY signal (0)
  - V.24 interface (1)
  - 7 data bits (11-bit character frame) (0)
  - Handshake OFF (0)

- Parameters for Communications mode (parameter block 7)
  - Character delay time 300 ms (0030_D)
  - Message frame size 6 bytes, no end-of-text character

It is recommended that you proceed as follows:

1. Enter parameter setting data in DB 1 on the programmer and include detailed commentary.
2. Store DB 1 on floppy disk or hard disk.
3. Transfer DB 1 to the memory submodule.

Table 7-4. Setting Parameters for Transparent Mode on the Memory Submodule

Entry on PG 685	Description	
: KS = 'Parameters for the seri';	0	Parameter block 0
: S = 'al interface';	6	2400 Bd
: KS = ':0600100';	0	Even parity
	0	No BUSY signal
	1	V.24 interface
	0	7 data bits (11-bit character frame)
	0	Handshake OFF
: S = 'Parameters for Communications';	7	Parameter block 7
: KS = 'mode';	1	Transparent mode
: S = ':710030006';	0030_D	Character delay time 300 ms
	006_D	Frame size 6 bytes

Setting parameters for ”INTERPRETIVE MODE” via the MEMORY SUBMODULE

When you have decided to transmit your data in Interpretive mode, you enter all parameter setting data in parameter blocks

Table 7-5. Parameter Block Assignments for Interpretive Mode

Parameter Block	Description	Value Range	Default Values on the CP 523
0	Baud rate 110 Bd	1	4
	200 Bd	2	
	300 Bd	3	
	600 Bd	4	
	1200 Bd	5	
	2400 Bd	6	
	4800 Bd	7	
	9600 Bd	8	
	Parity even	0	0
	odd	1	
	”mark”	2	
	”space”	3	
	no parity check	4	
	BUSY signal no	0	0
	Interface TTY	0	0
	V.24	1	
	Data format: Parity:		
	11-bit character frame 7 Data bits (yes)	0	0
	8 Data bits (yes)	1	
	8 Data bits (yes)	2	
	10-bit character frame 7 Data bits (no)	3	
	7 Data bits (yes)	4	
	8 Data bits (no)	5	
	HW handshake OFF	0	0
	ON	1	
2	XON character	01_H to $7E_H$	$FFFF_H$ (no XON/XOFF protocol)
	XOFF character	01_H to $7E_H$	

Table 7-5. Parameter Block Assignments for Interpretive Mode (Continued)

7	Mode Print mode Transparent mode Interpretive mode 3964 (R) mode	 0 1 2 3	0 (for memory submodule configured with message texts) 1 (no memory submodule)
	Character delay time (1 to 3000) · 10 ms	0001_D to 3000_D	0001 (6 10 ms)
	Frame length (0 bytes) One (byte 7) or two (bytes 6 and 7) end-of-text characters must be specified when the frame length is 0 bytes.	000_D	000
	End-of-text characters (only when frame length = 0) 1 end character 2 end character	 0001_H to $00FF_H$ 0101_H to $FFFF_H$	$0D_H$

Example: Using the PG 685 to set data parameters on the MEMORY SUBMODULE for ”INTERPRETIVE MODE”.

Start address 128 is set on the CP 523. The module is to be operated in Interpretive mode.
You can set the module parameters as follows:

- Parameters for the serial interface (parameter block 0)
  - 2400 baud (6)
  - Even parity (0)
  - No BUSY signal (0)
  - RS-232-C (V.24) interface (1)
  - 7 data bits (11-bit frame) (0)
  - Handshake OFF (0)

- Parameters XON/XOFF (parameter block 2)
  - XON character: DC1 (11_H)
  - XOFF character: DC3 (13_H)

- Parameters for Communications mode (parameter block 7)
  - Character delay time 300 ms (0030_D)
  - Variable frame length with two end-of-text characters ($0D0D_H$)

The following procedure is recommended:

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

Table 7-6. Setting the CP 523 Parameters in Interpretive Mode on the Memory Submodule

Entry on the PG 685	Explanation

`: KS = 'Parameters for the serial';`
`: S = 'Interface';`
`: KS = ':0600100';`
	0 Parameter block 0
	6 2400 baud
	0 Even parity
	0 No BUSY signal
	1 RS-232-C (V.24) interface
	0 7 data bits (11-bit frame)
	0 Handshaking OFF

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

`: S = 'Parameters for ASCII';`
`: KS = 'mode';`
`: S = ':720030000';`
 	
`: KH = 0D0D;`
	7 Parameter block 7
	2 Interpretive mode
	0030_D Character delay time 300 ms
	000_D Message frame length 0
	$0D0D_H$ End-of-text characters (two carriage returns)

Setting Parameters for ”3964(R) MODE” via the MEMORY SUBMODULE

When you have decided upon 3964(R) mode, you enter your data in blocks 0 and 7. Enter your data in the same way as in the other modes. ”3964(R)” mode requires additional parameters in parameter block 7.

Table 7-7. Parameter Block Assignments for 3964(R) Mode

Parameter Block	Description			Value Range	Default Values on the CP 523
0	Baud rate		110 Bd 200 Bd 300 Bd 600 Bd 1200 Bd 2400 Bd 4800 Bd 9600 Bd	1 2 3 4 5 6 7 8	4
	Parity		even odd ”mark” ”space” no parity check	0 1 2 3 4	0
	BUSY signal		No	0	0
	Interface		TTY V.24 (RS 232 C)	0 1	0
	Data format:		Parity:		
	11-bit character frame	7 data bits 8 data bits 8 data bits	(yes) (yes) (no)	0 1 2	0
	10-bit character frame	7 data bits 7 data bits 8 data bits	(no) (yes) (no)	3 4 5	
	HW handshake		OFF ON	0 1	0

Table 7-7. Parameter Block Assignments for 3964(R) Mode (Continued)

7	Mode Print mode Transparent mode Interpretive mode 3964(R) mode	 0 1 2 3	0 (for memory submodule configured with message texts) 1 (no memory submodule)
	Character delay time (ZVZ) (1 to 65535) ·10 ms	00001_D to 65535_D	22_D
	Time-out (QVZ) (1 to 65535) ·10 ms	00001_D to 65535_D	00200_D
	Block delay time (BWZ) (1 to 65535) ·10 ms	00001_D to 65535_D	00400_D
	Transfer **with** block check character (3964R) Transfer **without** block check character (3964)	1 0	 0
	Priority Low High	 0 1	 1
	Connection buildup attempts	001_D to 255_D	6
	No. of Send attempts	001_D to 255_D	6

Example: Setting parameters on the MEMORY SUBMODULE for ”3964(R)” MODE over the PG 685

Start address 128 is set on the CP 523. The module is to be operated in ”3964(R)” mode.
Parameters are to be set as follows:

- Parameters for the serial interface (parameter block 0)
  - 2400 baud (6)
  - Even parity (0)
  - No BUSY signal (0)
  - V.24 (RS 232 C) interface (1)
  - 7 data bits (11-bit frame) (0)
  - Handshaking OFF (0)

- Parameters for ”3964(R) mode” (parameter block 7)
  - Character delay time 1 s (00100)
  - Time-out 2 s (00200)
  - Block delay time 4 s (00400)
  - With block check character (1)
  - Low priority (0)
  - Buildup attempts (012)
  - No. of Send attempts (003)

The following procedure is recommended:

1. Enter the initialization data in DB 1 using the programmer, and include sufficient commentary.
2. Store DB 1 on floppy or hard disk.
3. Transfer DB 1 to the memory submodule.

Table 7-8. Setting Parameter for 3964(R) Mode on the Memory Submodule

Entry on the PG 685	Description	
: KS = 'Parameters for the serial';	0	Parameter block 0
: S = 'interface';	6	2400 baud
: KS = ':0600100';	0	Even parity
	0	No BUSY signal
	1	V.24 (RS232C) interface
	0	7 data bits (11-bit frame)
	0	Handshaking OFF
: S = 'Parameters for';	:7	Parameter block 7
: KS = '3964(R) mode';	3	3964(R) mode
: S = ':73001000020000400010012003';	00100_D	Character delay time
	00200_D	Time-out
	00400_D	Block delay time
	1	3964R (with BCC)
	0	Low priority
	012_D	Connection buildup attempts
	003_D	No. of Send attempts

7.3.2 Setting the CP 523 Parameters in the User Program

You can transfer data for a parameter block to the CP 523 with a ”Transfer parameter setting data” request ($90xx_H$).

Note:
You can only set parameters for parameter blocks 0, 2, 7 and 9 in the user program with ”Transfer parameter setting data”. Parameter blocks 1, 3 to 6 and 8 are only significant in Print mode.

Reader's note: You will find a list of parameters (Tables 7-9 to 7-12) in Section 7.3.2. For editorial reasons, the parameter blocks are discussed in the order 0, 7, 2, 9.

The parameter tables are followed by the subsections

- Setting parameters for TRANSPARENT MODE (page 7-28)
- Setting parameters for INTERPRETIVE MODE (page 7-32)
- Setting parameters for 3964(R) MODE (page 7-38)

When you have selected the mode you want, you can limit your reading to the appropriate subsection.

Table 7-9. Transfer Memory Assignments for ”Transfer Initialization Data for Parameter Block 0”

Byte	Meaning	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”	90_H	-
1	Bit 4 to 7: Parameter block number Bit 0 to 3: Only significant for parameters block 7 and 9	00_H	-
2	Baud rate 110 baud	01_H	
	200 baud	02_H	04_H
	300 baud	03_H	
	600 baud	04_H	
	1200 baud	05_H	
	2400 baud	06_H	
	4800 baud	07_H	
	9600 baud	08_H	
3	Parity even	00_H	
	odd	01_H	00_H
	”mark”	02_H	
	”space”	03_H	
	no parity check	04_H	
4	BUSY signal No	00_H	00_H
5	Interface TTY	00_H	00_H
	V.24	01_H	
6	Data format: Parity:		
	11-bit character 7 data bits (yes)	00_H	00_H
	frame 8 data bits (yes)	01_H	
	8 data bits (no)	02_H	
	10-bit character 7 data bits (no)	03_H	
	frame 7 data bits (yes)	04_H	
	8 data bits (no)	05_H	
7	HW handshake OFF	00_H	00_H
	ON	01_H	

Table 7-10. Transfer Memory Assignments for “Transfer Parameter Setting Data for Parameter Block 7“

	71_H			72_H			73_H			$7A_H$			
	Transparent Mode			Interpretive Mode			3964(R) Mode			3964(R) Mode Follow-Up Request			
›yte	Description	Permissible Values	Default Values on the CP 523	Description	Permissible Values	Default Values on the CP 523	Description	Permissible Values	Default Values on the CP 523	Description	Permis-sible Values	Default Values or the CP 523	›yte
0	Job no	90_H		Job no	90_H		Job no.	90_H		Job no.	90”		0
1	Bits 4 to 7: Parameter block number Bits O to 3: Mode	71_H		ˌits 4 to 7: arameter block ıumber ˌits O to 3: /lode	72_H		›its 4 to 7: 'arameter block ıumber ›its Oto 3: /lode	73_H		Bits 4 to 7: Parameter block number Bits O to 3: Mode	$7A_H$		1
2	Character delay time (1 to 3000).10 ms	0001_H to $0BB8_H$	0001_{HH} = 10 ms	:haracter delay ıme 1 to 3000). 10 ms	0001_H to $0BB8_H$	0001. = 10 ms	:haracter delay ime (ZVZ) 1 to 65535).10 ms	0001. to $FFFF_H$	16_H (220 ms)	Transmiss. without block check char. Transm. with block check character	00H 01H	00”	2
3										Low Priority High	00. 01H	01H	3
4	Frame length (1 to 256 bytes)	0001” to 0100_H	0040,	Frame length (O bytes)	0000,	0000”	ˈime-out QVZ) 1 to 65535) .10 ms	0001. to $FFFF_H$	$00C8_H$ (2000 ms = 2s)	Connect. buildup attempts (1 to 256)	00_H to FF_H	06_H	4
5										Number of retries (1 to 255)	00_H to FF_H	06.	5
6	Irrelevant			:nd-of-text-charac-ters, 1 end-of-text character in byte 7 (Byte 6 = O) or2 end-of-text cha-racters in bytes 6 and 7	0001. to $00FF_H$ 0101. to $FFFF_H$	$0D_H$	›lock delay time BWZ) 1 to 65535) ● 10 ms	0001_H to $FFFF_H$	0190. (4000 ms = 4s)	Irrelevant			6
7													7

Transferring parameter setting data for parameter block 2
(relevant in Interpretive mode only)

Table 7-11. Transfer Memory Assignments for ”Transfer Parameter Setting Data for Parameter Block 2”

Byte	Meaning	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”	90_H	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Only significant for parameter blocks 7 and 9	20_H	-
2	XON character	01_H to $7E_H$	$FFFF_H$ (No XON/XOFF protocol)
3	XOFF character	01_H to $7E_H$	
4 to 7	Not significant	-	-

Transferring parameter setting data for parameter block 9

Table 7-12. Transfer Memory Assignments for the "Transfer Parameter Setting Data for Parameter Block 9"

Byte	Meaning	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”	90_H	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Sign for correction value 0 positive 1 negative	9_H 0_H 1_H	-
2+3	Correction value (s/month)	0_D to 400_D	0000_D
4 to 7	Not significant	-	-

Setting parameters for ”TRANSPARENT MODE” in the USER PROGRAM

Transferring the parameter setting data for parameter block 0

Table 7-13. Transfer Memory Assignments for ”Transfer Parameter Setting Data for Parameter Block 0”

Byte	Description			Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”			90_H	-
1	Bits 4 to 7:	Parameter block number			-
	Bits 0 to 3:	Relevant for parameter blocks 0 and 9 only		00_H	
2	Baud rate		110 Bd	01_H	
			200 Bd	02_H	04_H
			300 Bd	03_H	
			600 Bd	04_H	
			1200 Bd	05_H	
			2400 Bd	06_H	
			4800 Bd	07_H	
			9600 Bd	08_H	
3	Parity		even	00_H	
			odd	01_H	00_H
			”mark”	02_H	
			”space”	03_H	
			no parity check	04_H	
4	BUSY signal		No	00_H	00_H
5	Interface		TTY	00_H	00_H
			V.24	01_H	
6	Data format:		Parity:		
	11-bit character	7 data bits	(yes)	00_H	00_H
	frame	8 data bits	(yes)	01_H	
		8 data bits	(no)	02_H	
	10-bit character	7 data bits	(no)	03_H	
	frame	7 data bits	(yes)	04_H	
		8 data bits	(no)	05_H	
7	HW handshake		OFF	00_H	00_H
			ON	01_H	

Example: Transferring parameter setting data for parameter block 0

Start address 128 is set on the CP 523.
You can set its parameters as follows:

Parameters for the serial interface (parameter block 0)

• 2400 baud	(06_H)
• Even parity	(00_H)
• No BUSYsignal	(00_H)
• RS-232-C (V.24) interface	(01_H)
• 7 data bits (11-bit character frame)	(00_H)
• Handshaking OFF	(00_H)

STL OB 22	STL FB 110	Explanation
:JU FB 99	NAME :BLOCK0	
NAME :SYNC**	:L KH 0000	Handshaking OFF
:JU FB 110	:T PW 134	7 data bits (11-bit character frame)
NAME :BLOCK0	:L KH 0001	No BUSY signal
:BE	:T PW 132	RS-232-C (V.24) interface
	:L KH 0600	2400 baud
	:T PW 130	Even parity
	:L KH 9000	Transfer "Parameter setting data
	:T PW 128 *	parameter block 0" job number
	:BE	

* 5.2
** The program waits at FB 99 until the CP 523 can accept job request (4.1)

Note:
Restart organization block OB 22 is only processed if the CPU is in RUN mode before you switch from "POWER OFF" to "POWER ON".

Note:
The status byte can be scanned approx. 30 ms after the request was issued in order to check whether correct parameter setting data was transferred (X4 is returned to indicate errored values (7.4.1)).

Transferring parameter setting data for parameter block 7

Table 7-14. Transfer Memory Assignments for ”Transfer Initialization Data for Parameter Block 7”

Byte	Meaning	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”	90_H	-
1	Bits 4 to 7: Parameter block number	7	-
	Bits 0 to 3: Mode		0: If a memory submodule with message texts is plugged in 1: Without memory submodule
	Print mode	0	
	Transparent mode	1	
	Interpretive mode	2	
	”3964(R)” mode	3	
	3964(R) mode	A	
	Follow-up request		
2+3	Character delay time (1 to 3000) · 10 ms	0001_H to $0BB8_H$	0001_H 6 10 ms
4+5	Frame length (1 to 256 bytes)	0001_H to 0100_H	0040_H
6+7	Irrelevant	-	-

* Only possible in Interpretive mode

Example: Transferring the parameter setting data for parameter block 7

Start address 128 is set on the CP 523. The module is to be operated in Transparent mode. The message frames to be transferred have a length of 6 bytes. An end-of-text character is not required. The character delay time may be 3 s (3 s=001E$_H$).

STL OB 22	STL FB 111	Explanation
:JU FB 99	NAME :BLOCK7	
NAME :SYNC **	:L KH 0000	Load no end-of-text characters into ACCUM 1 and transfer to the CP in bytes 6 and 7.
:JU FB 111	:T PW 134	
NAME :BLOCK7	:L KH 0006	Load frame length into ACCUM 1 and transfer to the CP in bytes 4+5.
:BE	:T PW 132	
	:L KH 001E	Load character delay time into ACCUM 1 and transfer to the CP in bytes 2+3.
	:T PW 130	
	:L KH 9071	Load job number (90_H), parameter block number (7_H) and mode code (1_H) into ACCUM 1 and transfer to the CP
	:T PW 128 *	
	:BE	

* 5.2

** The program in FB 99 waits until the CP 523 can accept job requests (4.1)

Note:

The status byte can be scanned approx. 30 ms after the request was issued in order to check whether correct parameter values were forwarded (X4 is returned if values are false (7.4.1)).

Setting parameters for INTERPRETIVE MODE in the USER PROGRAM

Transferring the parameter setting data for parameter block 0

Table 7-15. Transfer Memory Assignment for ”Transfer Parameter Setting Data for Parameter Block 0”

Byte	Description	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”	90_H	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Relevant for parameter blocks 7 and 9 only	00_H	-
2	Baud rate 110 baud 200 baud 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud	01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	04_H
3	Parity even odd ”mark” ”space” no parity check	00_H 01_H 02_H 03_H 04_H	00_H
4	BUSY signal No	00_H	00_H
5	Interface TTY V.24	00_H 01_H	00_H
6	Data format: Parity: 11-bit character frame 7 data bits (yes) 8 data bits (yes) 8 data bits (no) 10-bit character frame 7 data bits (no) 7 data bits (yes) 8 data bits (no)	 00_H 01_H 02_H 03_H 04_H 05_H	00_H
7	HW handshake OFF ON	00_H 01_H	00_H

Example: Transferring parameter setting data for parameter block 0

Start address 128 is set on the CP 523.
Set CP parameters as follows:

Parameters for the serial interface (parameter block 0)

• 2400 baud	(06_H)
• Even parity	(00_H)
• No BUSY system	(00_H)
• V.24 (RS 232C) interface	(01_H)
• 7 data bits (11-bit character frame)	(00_H)
• Handshake OFF	(00_H)

STL OB 22	STL FB 112	Explanation
`:JU FB 99`	`NAME :BLOCK0`	
`NAME :SYNC**`	`:L KH 0000`	Handshake OFF
`:JU FB 112`	`:T PW 134`	7 data bits (11-bit character frame)
`NAME :BLOCK0`	`:L KH 0001`	No BUSY signal
`:BE`	`:T PW 132`	V.24 (RS 232C) interface
	`:L KH 0600`	2400 baud
	`:T PW 130`	Even parity
	`:L KH 9000`	Transfer job number for ”Parameter setting data for parameter block 0”
	`:T PW 128*`	
	`:BE`	

***** 5.2
****** FB 99 waits until the CP 523 can accept job requests (4.1)

Note:
Restart OB 22 is processed only when the CPU is at RUN before switching from ”POWER OFF” to ”POWER ON”.

Note:
The status byte can be scanned approx. 30 ms after issuing the job request to check on whether correct parameter values were forwarded (X4 is returned to indicate false values (7.4.1)).

Transferring parameter setting data for parameter block 2

Table 7-16. Transfer Memory Assignments for ”Transfer Parameter Setting Data for Parameter Block 2”

Byte	Description	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting data”	90_H	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Relevant for parameter blocks 7 and 9 only	20_H	-
2	XON character	01_H to $7E_H$	$FFFF_H$ (No XON/XOFF protocol)
3	XOFF character	01_H to $7E_H$	
4 to 7	Irrelevant	-	-

Example: Transferring parameter setting data for parameter block 2

STL OB 22	STL FB 113	Explanation
:JU FB 99 NAME :SYNC** :JU FB 113 NAME :BLOCK2 :BE	NAME :BLOCK2 :L KH 1113 :T PW 130 :L KH 9020 :T PW 128* :BE	Load ASCII codes for XON and XOFF character into ACCUM 1, transfer to CP in bytes 2 and 3. Load ACCUM 1 with job number and parameter block number and transfer to CP in bytes 0 and 1.

* 5.2

** FB 99 waits until the CP 523 can accept job requests (4.1)

Note:

The status byte can be scanned approx. 30 ms after the request was issued to check whether correct parameter values were forwarded (X4 is returned to indicate false values (7.4.1)).

Transferring parameter setting data for parameter block 7

Table 7-17. Transfer Memory Assignments for ”Transfer Parameter Setting Data for Parameter Block 7”

Byte	Description	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting”	90_H	-
1	Bits 4 to 7: Parameter block number	7	-
	Bits 0 to 3: Mode		0: when memory submodule with message texts is plugged in
	Print mode	0	
	Transparent mode	1	
	Interpretive mode	2	
	”3964(R)” mode	3	1: no memory subm.
2+3	Character delay time (1 to 3000) · 10 ms	0001_H to $0BB8_H$	0001_H 6 10 ms
4+5	Message frame length (1 to 256 bytes) One (Byte 7) or two (bytes 6 and 7) end-of-text characters must be specified if message frame length is 0 bytes	0000_H	0040_H
6+7	End-of-text characters* (only when message frame length= 0)		No end-of-text character
	1 end-of-text character (Byte 6=0)	0001_H to $00FF_H$	
	2 end-of-text characters	0101_H to $FFFF_H$	

* Possible in Interpretive mode only

Example: Transferring parameter setting data for parameter block 7

Start address 128 is set on the CP 523. The module is to be operated in Interpretive mode. The message frames to be transmitted have different lengths; the end of the message frame is therefore signalled with $0D0D_H$ ($0D0D_H$=two carriage returns). The character delay time may be 300 ms (300 ms=$001E_H$).

STL OB 22	STL FB 114	Explanation
:JU FB 99 NAME :SYNC ** :JU FB 112 NAME :BLOCK7 :BE	NAME :BLOCK7 :L KH 0D0D :T PW 134	Load end characters into ACCUM 1 and transfer to CP in bytes 6 and 7.
	:L KH 0000 :T PW 132	Load message frame length into ACCUM 1 and transfer to CP in bytes 4 and 5.
	:L KH 001E :T PW 130	Load character delay time into ACCUM 1 and transfer to CP in bytes 2 and 3.
	:L KH 9072 :T PW 128 * :BE	Load job number (90_H), parameter block number (7_H) and mode code (2_H) into ACCUM 1 and transfer to the CP.

* 5.2

** FB 99 waits until the CP 523 can accept job requests (4.1)

Note:

The status byte can be scanned approx. 30 ms after the request was issued to check whether correct parameter values were forwarded (X4 is returned to indicate false values (7.4.1)).

Setting parameters for the 3964(R) MODE in the USER PROGRAM

Transferring the parameter setting data for parameter block 0

Table 7-18. Transfer Memory Assignments for ”Transfer Parameter Setting Data for Parameter Block 0”

Byte	Description	Possible Values	Default Values on the CP 523
0	Job number ”Transfer parameter setting”	90_H	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Relevant for parameter blocks 7 and 9 only	 00_H	-
2	Baud rate 110 baud 200 baud 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud	01_H 02_H 03_H 04_H 05_H 06_H 07_H 08_H	04_H
3	Parity even odd ”mark” ”space” no parity check	00_H 01_H 02_H 03_H 04_H	00_H
4	BUSY signal No	00_H	00_H
5	Interface TTY V.24	00_H 01_H	00_H
6	Data format: Parity: 11-bit character frame 7 data bits (yes) 8 data bits (yes) 8 data bits (no) 10-bit character frame 7 data bits (no) 7 data bits (yes) 8 data bits (no)	 00_H 01_H 02_H 03_H 04_H 05_H	 00_H
7	HW handshake OFF ON	00_H 01_H	00_H

Example: Transferring parameter setting data for parameter block 0

Start address 128 is set on the CP 523.
Set CP parameters as follows:

Parameters for the serial interface (parameter block 0)

- 2400 baud (06_H)
- Even parity (00_H)
- No BUSY signal (00_H)
- V.24 (RS232C) interface (01_H)
- 7 data bits (11-bit character frame) (00_H)
- Handshake OFF (00_H)

STL OB 22	STL FB 115	Explanation
:JU FB 99	NAME :BLOCK0	
NAME :SYNC**	:L KH 0000	Handshake OFF
:JU FB 115	:T PW 134	7 data bits (11-bit character frame)
NAME :BLOCK0	:L KH 0001	No BUSY signal
:BE	:T PW 132	V.24 (RS232C) interface
	:L KH 0600	2400 baud
	:T PW 130	Even parity
	:L KH 9000	Transfer job number for ”Parameter
	:T PW 128*	setting data for parameter block 0”
	:BE	

* 5.2

** FB 99 waits until the CP 523 can accept job requests (4.1)

Note:

Restart OB 22 is processed only when the CPU is at RUN before switching from POWER OFF to POWER ON.

Special features pertaining to parameter setting of ”3964(R) mode” in the user program

As in ”Transparent” and ”Interpretive” mode, you transfer all parameter setting data to the CP with job number $90XX_H$ ”Transfer parameter setting data”. You enter the parameter setting data in the same way as in the other modes, except that you must transfer a number of additional parameters for ”3964(R) mode”. These added parameters are defined in parameter block 7. Since it is not possible to transfer all required parameters in 8 bytes, the parameter setting of parameter block 7 is divided into two steps. The job numbers for these steps are 9073_H (1st job request) and $907A_H$ (follow-up job request).
The contents of the transfer memory for job numbers 9073_H and $907A_H$ are listed in Tables 7-19 and 7-20.

Transfer parameter setting data for parameter block 7

Table 7-19. Transfer Memory Assignments for Job Number "9073$_H$"

Byte	Description	Possible Values	Default Values on the CP 523
0	Job number "Transfer initialization data"	90$_H$	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Mode Print mode Transparent mode Interpretive mode 3964(R) mode 3964(R) mode Follow-up request	7 0 1 2 3 A	- 0: Memory submodule with message texts 1: No memory submodule
2+3	Character delay time (ZVZ) (1 to 65535) · 10 ms	0001$_H$ to FFFF$_H$	16$_H$ (220 ms)
4+5	Time-out (QVZ) (1 to 65535) · 10 ms	0001$_H$ to FFFF$_H$	00C8$_H$ (2 s)
6+7	Message delay time (BWZ) (1 to 65535) · 10 ms	0001$_H$ to FFFF$_H$	0190$_H$ (4 s)

Table 7-20. Transfer Memory Assignment for Follow-up Job Request "9074$_H$"

Byte	Description	Possible Values	DefaultValues on the CP 523
0	Job number "Transfer initialization data"	90$_H$	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Mode Print mode Transparent mode Interpretive mode 3964(R) mode 3964(R) mode Follow-up job request	7 0 1 2 3 A	- 0: Memory submodule with message texts 1: No memory submodule
2	Without block check character With block check character	00$_H$ 01$_H$	00
3	Priority Low High	00$_H$ 01$_H$	01$_H$
4	Connection buildup attempts	00$_H$ to FF$_H$	06$_H$
5	Number of Send attempts	00$_4$ to FF$_H$	06$_H$
6+7	Irrelevant		

Example: Transferring parameter setting data for parameter block 7

Start address 128 is set on the CP 523. Set the parameters as follows in ”3964(R) mode”:

• Message frame delay time:	4000 ms
• Time-out:	2000 ms
• Character delay time:	220 ms
• Connection buildup attempts:	6
• Number of Send attempts:	6
• Transmission:	with block check character (BCC)
• Priority:	low

STL OB 22	STL FB 116	Explanation
:JU FB 99 NAME : SYNC* :JU FB 116 NAME :BLOCK7 :BE	NAME :BLOCK7 :L KH 0190 :T PW 134	Load message frame delay time (400 · 10 ms) into ACCUM 1 and transfer to CP in bytes 6 and 7
	:L KH 00C8 :T PW 132	Load acknowledge delay time (200_D · 10 ms) into ACCUM 1 and transfer to CP in bytes 4 and 5
	:L KH 0016 :T PW 130	Load character delay time (22_D · 10 ms) into ACCUM 1 and transfer to CP in bytes 2 and 3
	:L KH 9073 :T PW 128**	Load job number (90_H), parameter block number (7_H) and mode code (3_H) into ACCUM 1 and transfer to CP
	:L KH 0606 :T PW 132	Load connection buildup attempts (6) and number of Send attempts (6) into ACCUM 1 and transfer to CP in bytes 4 and 5
	:L KH 0100 :T PW 130	Load ”with BCC” (01_H) and ”low priority” (00_H) into ACCUM 1 and transfer to CP in bytes 2 and 3
	:L KH 907A :T PW 128** :BE	Load job number (90_H), parameter block number (7_H) and follow-up request code (A_H) into ACCUM 1 and transfer to CP

* FB 99 waits until the CP 523 can accept job requests (4.1)

** (5.2)

Note:

The status byte can be scanned approx. 30 ms after the job request is issued to check whether correct initialization values were forwarded (X4 is returned to indicate errored values (7.4.1)).

Transfer parameter setting data for parameter block 9

Byte	Description	Possible Values	Default Values on the CP 523
0	Job number "Transfer parameter setting data"	90_H	-
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Sign for correction value 0 positive 1 negative	9_H 0_H 1_H	-
2+3	Correction value (s/month)	0_D to 400_D	0000_D
4 to 7	Not significant	-	-

Example: Transferring parameter setting data for parameter block 9

You have discovered that the clock loses 12 seconds in 4 days, which totals to 90 seconds in 30 days. The correction value is thus 90 s/month.

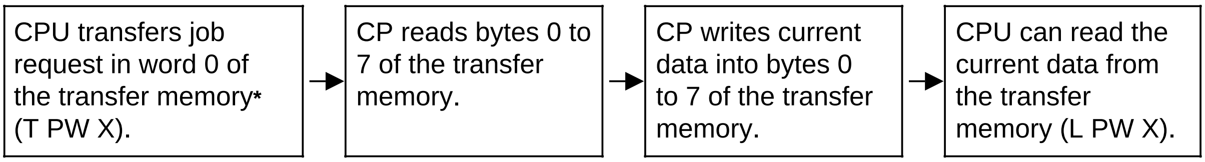
STL OB 22	STL FB 117	Description
:JU FB 99 NAME : SYNC** :JU FB 117 NAME :BLOCK9 :BE	NAME :BLOCK9 :L KF +90 :T PW 130 :L KH 9090 :T PW 128* :BE	Load correction value into ACCUM 1 and transfer to CP in bytes 2 and 3 Load job number and parameter block number into ACCUM 1 and transfer to CP in bytes 0 and 1

* 5.2

** FB 99 waits until the CP 523 can accept job requests (4.1)

7.4 CP 523 Feedback Information

The CP 523 has an eight-byte transfer memory for data exchange with the CPU.
The CP 523 reads the data from the transfer memory and updates the transfer memory with current values only when the start address of the module has been written into word 0 of the transfer memory with T PW in the user program. The current data can then be read in the user program with L PW statements.



CPU transfers job request in word 0 of the transfer memory* (T PW X).

CP reads bytes 0 to 7 of the transfer memory.

CP writes current data into bytes 0 to 7 of the transfer memory.

CPU can read the current data from the transfer memory (L PW X).

* 5.2

Figure 7-6. Using the Transfer Memory

The meaning of the data written into the transfer memory by the CP depends on:

- the job number that was transferred to the CP in word 0
  and
- the initialized mode

Table 7-21. Job Requests and the Associated Feedback Information

Job Requests		Initialized Mode	Return Info from CP See Section
0000_H:	Read status byte, status of I/O device and current clock data (7.4.1)	1, 2, 3	7.4.1
1000_H:	Set clock (6.7.2)	1, 2, 3	7.4.1
$90XX_H$:	Transfer parameter setting data (7.3.2)	1, 2	7.4.1
		3	7.4.1
$A0XX_H$:	Coordinate data transfer (7.6 and 7.7)	1	7.4.2
		2	7.4.2
		3	7.4.2

7.4.1 Reading the Status Byte, the Status of the Peripheral Device and the Current Clock Data

As soon as job request 0000_H has been transferred to the CP, the CP writes the following into the transfer memory:

- The status byte (byte 0)
- The status of the peripheral device (byte 1)
- The current clock data (bytes 1 to 7)

The return info is the same for job requests:

- 1000_H: ”Set clock”
- $90XX_H$: Transfer parameter setting

Table 7-22. Status Information and Current Clock Data

Byte	Meaning	Possible Values
0	Status byte	7.24
1	Status of the peripheral device Bits 4 to 7: Peripheral device ready Peripheral device not ready Bits 0 to 3: Current day of the week 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday 5=Thursday, 6=Friday, 7=Saturday	 $0X_{BCD}$ $1X_{BCD}$ $X1_{BCD}$ 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 24 h clock 12 h clock a.m. 12 h 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}

Status byte (byte 0)

The status byte provides information on

- errors during data interchange between CP and I/O device
- the status of a CPU job
- the status of the CP 523
- battery backup

The status byte is divided into two half-bytes. Both half-bytes are independent of each other. They can be combined in any way.

Table 7-23. Status Byte (Byte 0) in Communications Mode Following ”Coordinate Data Transfer”

Byte 0 Bits 4 to 7	 Bits 0 to 3	Status
0	0	No error
X	1	Memory submodule defective
X	3	Entries in job buffer (init. requests only)
X	7	No battery backup
X	8	Job buffer full (init. requests only)
0	F	CP restart in progress
1	X	Clock defective
2	X	Default time set
3	X	Time/data error
4	X	Illegal job request
8	X	Hardware fault

No battery backup ($X7_H$)

Either
- no battery has been inserted in the power supply module, or
- the battery is defective

Default time set ($2X_H$)

The clock is set to Sunday 01.01.90 12:00:00.

Data/time error ($3X_H$)

At least one variable is out of range.
The new data were rejected and the clock retains the old setting.

Illegal job request ($4X_H$)

A job number not permitted in Communications mode was entered in word 0 of the transfer memory. See 7.5 for a list of permissible job numbers for Communications mode. This error code is also used to flag invalid parameters in a parameter setting request.

Example: Evaluating the status of the peripheral device

Start address 128 is set on the module.
If the peripheral device is not ready for operation, output 4.1 is set.

STL FB 118	Explanation
NAME :ERROR3	
:L KH 0000	Transfer nonrelevant data to the CP. The CP then updates the transfer memory with the current data.
:T PW 128*	
:L KH 0010	”Status of the peripheral device” evaluation
:L PY 129	
:AW	Set bits 0 to 3 to ”0”
:><F	Compare ACCUM 1 and ACCUM 2: If not equal,
:BEC	Block End
:S Q 4.1	Otherwise, set output 4.1
:BE	Block End

* 5.2

Example: Reading the current clock data

Start address 128 is set on the module.
The clock data is to be output to digital output modules beginning address 8.

STL FB 102	Explanation
NAME :CLOCKDAT	
:L KH 0000	Load ACCUM 1 with nonrelevant data and transfer to the CP in word 0 so that the CP will transfer the current data to the transfer memory.
:T PW 128*	
:L PY 129	
:L KH 000F	Read the current clock data and transfer to the digital output modules.
:AW	
:T QB 8	
:L PW 130	
:T QW 10	
:L PW 132	
:T QW 12	
:L PW 134	
:T QW 14	
:BE	

* 5.2

Note:

When using the integral clock, you can evaluate the following additional flags in the status byte:

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

7.4.2 Reading Coordination Information after "Coordinate Data Transfer"

As soon as the "Coordinate data transfer" request has been transferred to the CP, the CP 523 writes coordination information into bytes 1 to 7 of the transfer memory. This information can be read out with load operations.

This coordination info is available in Transparent and Interpretive mode after the following requests:

- A000$_H$: Read status byte and coordination info
- A001$_H$: Send message frame
- A080$_H$: Receive message frame

In ”3964(R) mode”, different coordination info is returned when these same job requests are issued. This info is discussed at the end of this section.

Table 7-24. Coordination Information after "Coordinate Data Transfer"

Byte	Meaning	Possible Values
0	Status byte	5X$_H$
1	Permission to send and receive CPU can neither send nor receive CPU can send CPU can receive CPU can send and receive	 00$_H$ 01$_H$ 80$_H$ 81$_H$
2+3	Message length in bytes*	0000$_H$ to 0100$_H$
4+5	Irrelevant	0000$_H$
6	Number of message frames (calculated by the CP after a send request with fixed message length)	00$_H$ to 20$_H$
7	Number of message frames in Receive mailbox (The CP can buffer up to 100 message frames in the Receive mailbox. The sum of the length of these messages may not exceed 1024 bytes. When a message has been entered in the Receive mailbox, bit 7 of byte 1 is ”0”.)	00$_H$ to 64$_H$

* Only after "Receive message" (A080$_H$); in the case of an illegal Send request FF00$_H$

Status byte (byte 0)
In the status byte, bits 4 to 7 always have the value 5$_H$ following a "Coordinate data transfer" request. Bits 0 to 3 contain error flags.

Table 7-25. Status Byte (Byte 0) in Communications Mode after ”Coordinate Data Transfer”

Byte 0		Status
Bits 4 to 7	Bits 0 to 3	
5	9*	Character delay time exceeded
5	A*	Parity error
5	B*	Receive after XOFF or after DTR = 0
5	C*	Frame exceeds 256 bytes
5	D	Continuous break in line to I/O device
5	E*	Receive mailbox overflow

* Signal state irrelevant for the other half-byte

Character delay time exceeded (59_H)
The interval between two characters exceeds the value set in parameter block 7.
The data received up until this point is transferred as message frame to the CPU.
This is advantageous when you want to receive message frames of varied lengths although a fixed message frame size has been specified. This works when an I/O device sends variable-length data message frames without specific end-of-text characters and when the message frames arrive at long intervals (manual entry).

Parity error ($5A_H$)
The parity of received characters does not agree with the parity configured in parameter block 0.
The message frame is not transferred to the CPU and is not stored in the Receive mailbox.

Receive after XOFF or receive after DTR = OFF ($5B_H$)
In the case of XON/XOFF protocol:
The CP 523 sends XOFF to the peripheral device if

- less than 20 bytes are free in the Receive mailbox
- the CP has received 99 message frames.

The CP 523 sends XON again only when more than 256 bytes are free in the Receive mailbox.

When using MODEM control signals:
The CP 523 sends DTR = OFF to the peripheral device if

- less than 20 bytes are free in the Receive mailbox
- the CP has received 99 message frames.

The CP 523 sends DTR = ON again only when more than 256 bytes are free in the Receive mailbox.

Error XB_H is flagged if the peripheral device has sent more characters than the CP can accommodate in the Receive mailbox. The message frame is not transferred to the CPU and not stored in the Receive mailbox.

Message frames exceeding 256 bytes ($5C_H$)

The peripheral device has sent a message frame that is longer than 256 bytes.

The CP flags error $5C_H$ if it has not detected an end-of-text character after receiving 256 characters.

The message frame is not transferred to the CPU and not stored in the Receive mailbox.

Permanent wire break in the peripheral device cable ($5D_H$)

During the transmission, the peripheral device must

- hold the RXD line constantly at logic "1" or
- hold the CTS line constantly at ”ON” when using MODEM control signals (”ON”: V 3 V).

Otherwise an error is flagged.

After a permanent break has been remedied, the sender can resume transmission when the receiver has returned "XON". Data arriving at the receiver before the XON may be lost.

Receive mailbox overflow ($5E_H$)

The following can be stored in the Receive mailbox:

- Up to 1024 bytes of data
- Up to 99 message frames.

If these values are exceeded when receiving a message frame from the peripheral device, the CP flags an error.

The message frame is not transferred to the CPU and not stored in the Receive mailbox.

Example: Reading coordination information after "Send message frame"

Start address 128 is set on the CP 523. A four-byte message frame is to be sent. The message frame is stored in DB 20 in DW 0. Flag 100.3 is to be set if the CP 523 accepts the job. The program does not describe how F100.3 is reset. F 100.3 could be reset when another job request is forwarded to the CP.

STL FB 119	Explanation
NAME :SEND1	
:A F 100.3	If flag 100.3 is set, the CP 523 has accepted the job request,
:BEC	then Block End
:L KH 0004	Load message frame length 4 bytes into ACCUM 1 and transfer
:T PW 130	to word 2 of the transfer memory
:L KH A001	Load "Send message frame" (A001$_H$) into ACCUM 1 and transfer
:T PW 128*	to word 0 of the transfer memory
:L KH 0001	Check whether the send request has been accepted
:L PW 128	
:AW	
:><F	
:BEC	If no, conditional Block End
:S F 100.3	If yes, set flag 100.3 and Send message frame
:C DB 20	Open DB 20
:L DW 1	Transfer DW 1
:T PW 130	to word 2 of the transfer memory
:L DW 0	Transfer DW 0
:T PW 128*	to word 0 of the transfer memory
:BE	Block End

* 5.2

Note:

The program can only run if DB 20 has been generated on the CPU. You should create DB 20 as follows, since it is also accessed in the following examples:

DW 0	:	KH=A53D
DW 1	:	KH=34E1
DW 2	:	KH=9002
DW 3	:	KH=8765
DW 4	:	KH=6ED4
DW 5	:	KH=0D0D
DW 6	:	KH=0000
DW 7	:	KH=0000

Example: Reading coordination information after "Receive message frame"

Start address 128 is set on the CP 523. If the CP has received message frames from the peripheral device, these are to be transferred to the CPU. The message frames transferred from the peripheral device have a fixed length of four bytes. The message frame is to be stored on the CPU in DB 21 beginning DW 0.

STL FB 120	Explanation
NAME :RECEIVE1	
:L KH A080	Load "Receive message frame" into ACCUM 1 and transfer
:T PW 128*	to word 0 of the transfer memory.
:L KH 0080	Check that the Receive mailbox is empty
:L PW 128	Read the coordination data
:AW	
:><F	If the Receive mailbox is empty,
:BEC	Block End
:C DB 21	If it is not empty, open DB 21 and read out message frame data
:L KH 0000	
:T PW 128*	
:L PW 130	Read word 2 of the transfer memory
:T DW 1	and store in DW 1
:L PW 128	Read word 0 of the transfer memory
:T DW 0	and store in DW 0
:BE	Block End

* 5.2

Example: Parity check

Start address 128 is set on the module. FB 120 is to be invoked when a parity error is detected. The error is processed in FB 120, for instance by requesting the peripheral device to retransmit the frame.

STL FB 121	Explanation
:L KH A080	Fetch data frame
:T PW 128*	from CP 523
:L PW 128	Read module status
:L KH 0F00	Blank error
:AW	
:L KH 0A00	Has a parity error been detected?
:!=F	If yes, jump to FB 120;
:JC FB 120	If no, block end
NAME :PARCHECK	
:BE	

* 5.2

Special features of ”3964(R) mode”

In ”3964 R” mode, you receive different return info after submitting a ”Coordinate data transfer” request than you receive in Transparent or Interpretive mode. The return info for ”3964(R) mode” is presented separately in the following.

In ”3964R” mode, there are three job requests which provide different return info:

1. Job request A000$_H$ interrogates the status of the CP 523
2. Job number A001$_H$ is a Send request
3. Job number A080$_H$ is a Receive request

Job number A000$_H$, which checks the status of the CP 523, should be submitted prior to every Send and Receive request.

The following tables list the status information returned by the CP 523 in all three cases.

Table 7-26. CP 523 Status Information for ”A000$_H$”

Byte	Description	Possible Values
0	Status byte	50$_H$
1	Permission to send and receive CPU can neither send nor receive CPU can send CPU can receive CPU can send and receive	 00$_H$ 01$_H$ 80$_H$ 81$_H$
2	Coordination byte 'send' (CBS)	Table 7-30
3 - 7	Not significant	

Table 7-27. Status Information for Send Request ”A001$_H$”

Byte	Description	Possible Values
0	Status byte	50$_H$
1	Send request accepted Send request rejected	01$_H$ 00$_H$
2	Coordination byte 'send' (CBS)	Table 7-30
3 - 5	Not significant	
6	Number of message blocks in Send message frame	00$_H$ to 20$_H$
7	Number of message frames in Receive mailbox	00$_H$ to 64$_H$

Table 7-28. Status Information for Receive Request ”A080$_H$”

Byte	Description	Possible Values
0	Status byte	50$_H$
1	Coordination byte 'receive' (CBR)	Table 7-31
2+3	Message frame length in bytes	0000$_H$ to 0100$_H$
4+5	Irrelevant	
6	Number of message blocks in Receive message frames	00$_H$ to 20$_H$
7	Number of message frames in Receive mailbox	00$_H$ to 64$_H$

All information presented in the next three tables (7-29, 7-30, 7-31) is relevant in 3964(R) mode only.

Table 7-29. Return Info and Error Flags in the Coordination Byte 'Send' (CBS) in ”3964(R)” Mode

Bit								Description	Response
7	6	5	4	3	2	1	0		
0 1	XX_H*							**Return info for A000$_H$**	
								Send buffer free Send buffer not free	
0	XX_H*							**Return info for A001$_H$**	
								The information returned in the CBS following submission of a Send request (A001$_H$) depends on the value in byte 1. If byte 1 contains 00$_H$, CBS bit 7=0 means either that a parameter error was detected or that the Send request could not be serviced even though the Send buffer was free, as it was not preceded by a”check module status” request.	
1	XX_H*							CBS bit 7= 1 means that the Send buffer is not free because the previous Send request is still being serviced. If byte 1 is 01$_H$, CBS bit 7=1 means that the request was accepted; the Send buffer is now reserved for this request.	

***** See Error Flags

Table 7-29. Return Info and Error Flags in the Coordination Byte 'Send' (CBS) in ”3964(R)” Mode (Continued)

		Error flags for $A000_H$ and $A001_H$	
	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 error	No data is transmitted
	$0F_H$	Transmission aborted by receiver	Receive data invalid
	15_H	Time-out (QVZ) during connection buildup	No data is transmitted
	17_H	Time-out (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
	00_H	No errors	

Table 7-30. Return Info and Error Flags in the Coordination Byte 'Receive' (CBR) in ”3964(R) Mode”

	Bit	Description	Response
7	6 5 4 3 2 1 0		
		Return info for A080$_H$	
1 0	XX$_H$*	Request accepted No message frame to fetch	
		Error flags for A080$_H$	
	03_H	Parity error	Data rejected
	05_H	Frame length 0	
	07_H	Receive buffer full	Data rejected
	09_H	Too many message frames received	Buffer full, subsequent message frames will be rejected
	$0B_H$	Frame too long (256 bytes)	Data rejected
	$0D_H$	DLE not doubled or no ETX after DLE	Data rejected
	11_H	STX error, handshaking ON but no STX	Data rejected
	13_H	Character time-out	Data rejected
	15_H	Message frame time-out	Data rejected
	17_H	Checksum error	Data rejected
	$1B_H$	Break	Data rejected
	00_H	No errors	

***** See error message

When several errors occur simultaneously, the error codes are output in the priority shown below:

Table 7-31. Error Priority in ”3964(R) Mode”

Error Code	Description	Priority
		High
$0B_H$	Message frame too long	0
$1B_H$	Break	1
07_H	Input buffer full	2
09_H	Too many message frames	2
15_H	Message frame time-out	2
$0D_H$	DLE error	3
11_H	STX error	3
13_H	Character time-out	4
03_H	Parity error	5
17_H	BCC error	5
05_H	Message frame length 0	6
		Low

7.5 Overview of Permissible Job Requests in Communications Mode

Data exchange between the CPU and the CP 523 is always initiated by a job request from the CPU. The CPU transfers the job request to the CP 523 in word 0. The job number is stored in byte 0.

Table 7-32. Permissible Job Requests to the CP 523 in Communications Mode

Job Request	Byte 0		Byte 1	
	7 6 5 4	3 2 1 0	7 6 5 4	3 2 1 0
Read the status byte, the status of the peripheral device and the current clock data	0	0	0	0
Set the clock (settings in bytes 1 to 7)	1	0	Day of the week	
Transfer parameter setting data				
Initialize the serial interface	9	0	0	0
Set the XON/XOFF parameters	9	0	2	0
Select Print mode	9	0	7	0
Select Transparent mode	9	0	7	1
Select Interpretive mode	9	0	7	2
Select 3964(R) mode	9	0	7	3
Select 3964(R) mode follow-up job request	9	0	7	A
Correction value for the integral clock positive	9	0	9	0
Correction value for the integral clock neg.	9	0	9	1
Coordinate data transfer				
Read status byte and coordination information	A	0	0	0
Send message frame	A	0	0	1
Receive message frame	A	0	8	0

Illegal job requests in Communications mode

If you write a job request into word 0 of the transfer memory other than one of those listed above, the CP flags "Illegal job request" error $4X_H$ in the status byte.

The "Print all message texts" request 8000_H, for example, is permissible in Print mode but not in Communications mode. Error $4X_H$ would be flagged in the status byte in Communications mode.
You also receive this error flag if you transfer illegal parameter setting data with the "Transfer parameter setting data" request. In all cases (except $A0XX_H$ ”Coordinate data transfer”) you must wait 30 ms after initiating a job request before reading out the status byte.

Note:

In all cases (except $A0XX_H$ ”Coordinate data transfer”) you must wait 30 ms after initiating a job request before reading out the status byte.

Note:

- The "Transfer parameter setting data" requests (job number 90_H) are explained in section 7.3. These job requests are used to set the CP 523 parameters.
- The "Set clock" request is identical in Communications mode and Print mode (6.7.2).

7.6 Sending Message Frames to a Peripheral Device (Job Number A001$_H$)

The CPU can forward up to 256 bytes to the CP 523 with one Send request. This data is referred to as a message frame in the following.

Schematic for ” sending a message frame"

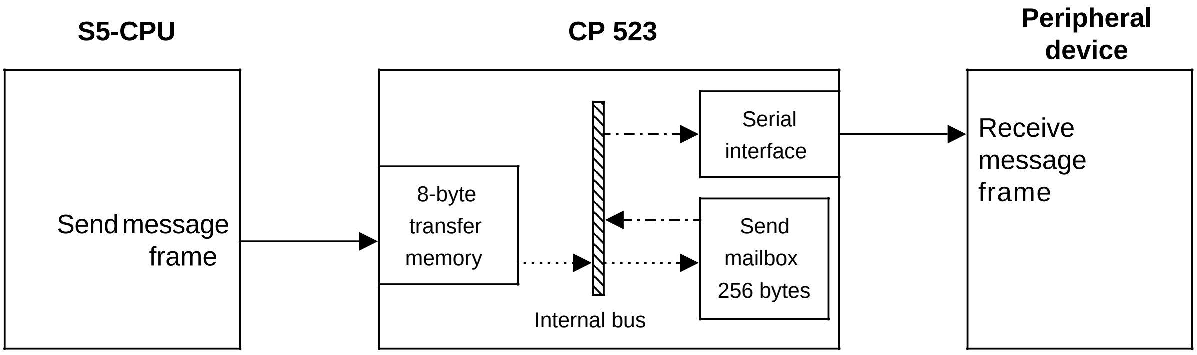


Figure 7-7. Sending Message Frames to a Peripheral Device

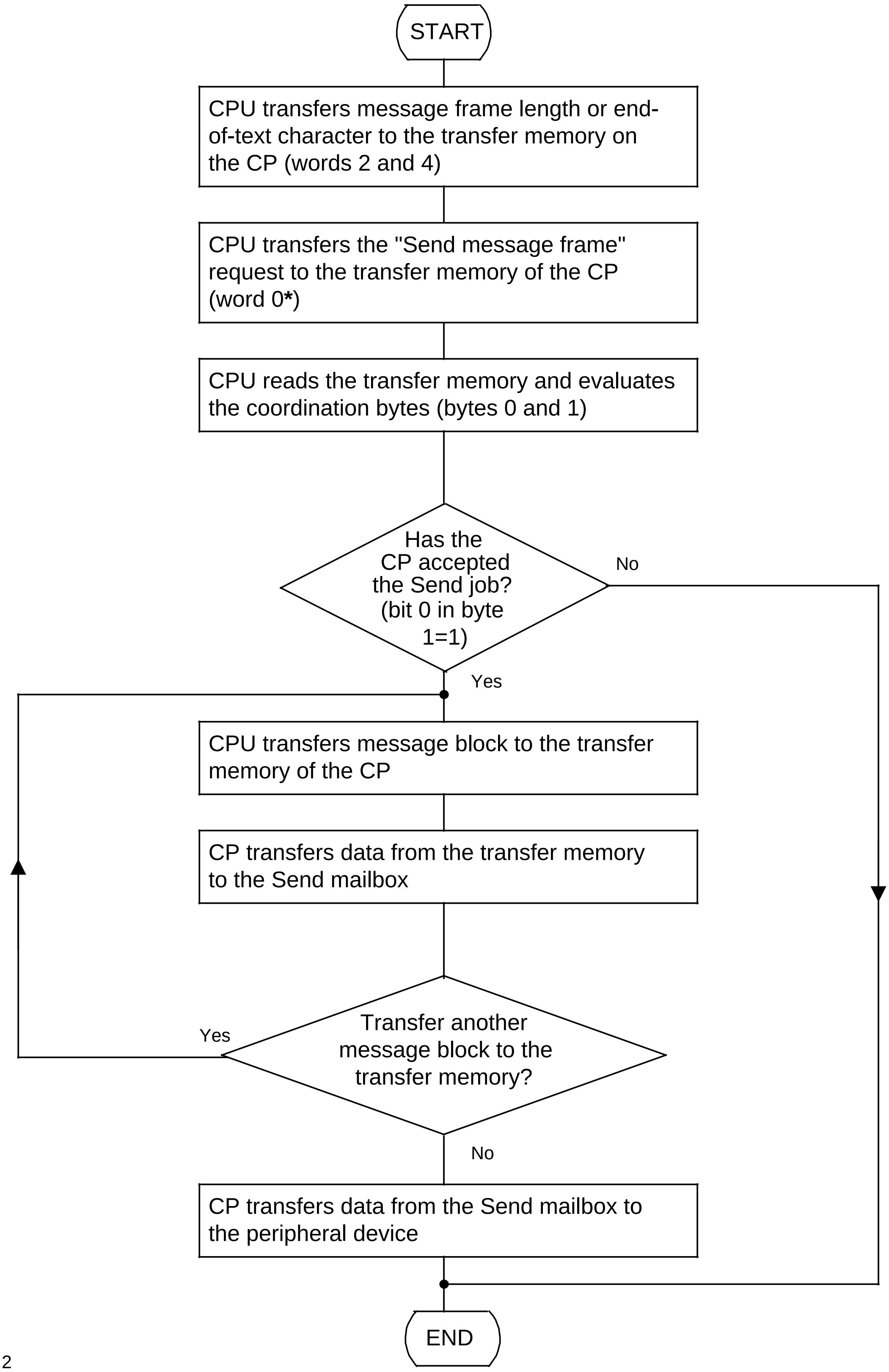
Data transfer between the CPU and the CP is always initiated by the CPU with a job request for "Coordinate data transfer for sending a message frame". When the CP has accepted the send job, the CPU transfers the message frame in message blocks of eight bytes.
The CP 523 has a Send mailbox of 256 bytes for storing the data from the CPU. The CP 523 accepts data from the transfer memory in the Send mailbox if the CPU has written the relevant data into word 0 of the transfer memory (·······▶).
After the CP has received the complete message frame, it transfers the message frame autonomously from the Send mailbox to the peripheral device via the serial interface (-·-·▶).
The CP can only accept another Send request when it has transferred the complete message frame to the peripheral device.

The following two pages contain a graphic representation of this procedure:

- "Send message frame" flowchart
- Transfer memory assignments from the user program and from the CP 523 for servicing the Send request

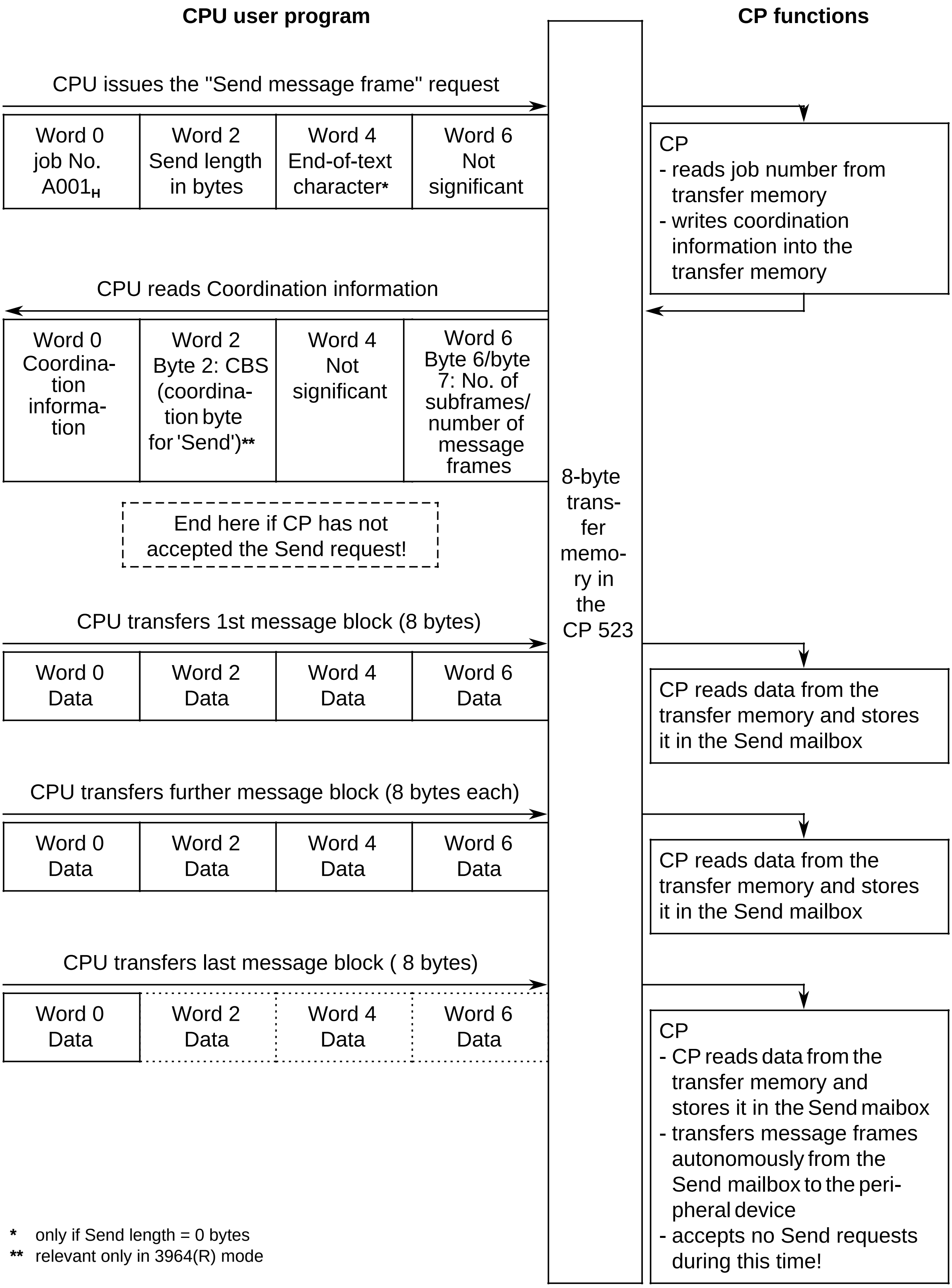
Note:

See 8.1 for a printout and explanation of the statement list of the "SEND" function block. The function block is a user-friendly interface for handling "Send message frame" requests.
Use of the function block is also explained in 8.1.


START
CPU transfers message frame length or end-of-text character to the transfer memory on the CP (words 2 and 4)
CPU transfers the "Send message frame" request to the transfer memory of the CP (word 0*****)
CPU reads the transfer memory and evaluates the coordination bytes (bytes 0 and 1)
Has the CP accepted the Send job? (bit 0 in byte 1=1)
No
Yes
CPU transfers message block to the transfer memory of the CP
CP transfers data from the transfer memory to the Send mailbox
Yes
Transfer another message block to the transfer memory?
No
CP transfers data from the Send mailbox to the peripheral device
END

* 5.2

Figure 7-8. Schematic for "Send Message Frame"



CPU user program

CP functions

8-byte transfer memory in the CP 523

CPU issues the "Send message frame" request

Word 0 job No. $A001_H$	Word 2 Send length in bytes	Word 4 End-of-text character*	Word 6 Not significant

CP
- reads job number from transfer memory
- writes coordination information into the transfer memory

CPU reads Coordination information

Word 0 Coordination information	Word 2 Byte 2: CBS (coordination byte for 'Send')**	Word 4 Not significant	Word 6 Byte 6/byte 7: No. of subframes/ number of message frames

End here if CP has not accepted the Send request!

CPU transfers 1st message block (8 bytes)

Word 0 Data	Word 2 Data	Word 4 Data	Word 6 Data

CP reads data from the transfer memory and stores it in the Send mailbox

CPU transfers further message block (8 bytes each)

Word 0 Data	Word 2 Data	Word 4 Data	Word 6 Data

CP reads data from the transfer memory and stores it in the Send mailbox

CPU transfers last message block (8 bytes)

Word 0 Data	Word 2 Data	Word 4 Data	Word 6 Data

CP
- CP reads data from the transfer memory and stores it in the Send maibox
- transfers message frames autonomously from the Send mailbox to the peripheral device
- accepts no Send requests during this time!

* only if Send length = 0 bytes

** relevant only in 3964(R) mode

Figure 7-9. Contents of the Transfer Memory for "Send Message Frame"

Message frame structure

A message frame consists of all the data transferred in response to a Send or Receive request. In the case of the CP 523, a message frame can be up to 256 bytes long. The CPU sends and receives a message frame in message blocks of eight bytes.

The CP 523 can detect the end of a message frame sent by the CPU in two ways:

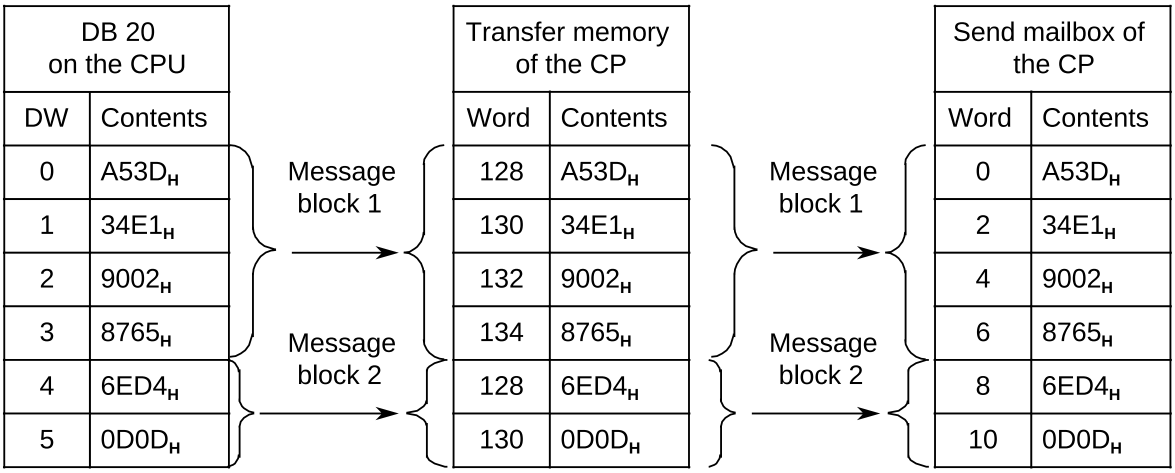
- You specify the message frame size in bytes with the "Send message frame" request.
- You specify one or two end-of-text characters with the "Send message frame" request. You must specify 0 bytes as the Send message frame size.
  The end-of-text characters must be identical to those defined when setting the CP 523 parameters (7.3)

Transferring a message frame

After the CP 523 has accepted a Send request, the CPU transfers the message frame to the transfer memory in message blocks of eight bytes. After the CPU has written word 0, the CP accepts the data from the transfer memory and stores it in the Send mailbox.
This means you transfer words 2, 4 and 6 to the transfer memory first and then word 0.

Example: Transferring a message frame with a length of 12 bytes.

Start address 128 is set on the CP 523. The message frame is stored in data block 20 beginning from DW 0.



DB 20 on the CPU	
DW	Contents
0	A53D$_H$
1	34E1$_H$
2	9002$_H$
3	8765$_H$
4	6ED4$_H$
5	0D0D$_H$

Transfer memory of the CP	
Word	Contents
128	A53D$_H$
130	34E1$_H$
132	9002$_H$
134	8765$_H$
128	6ED4$_H$
130	0D0D$_H$

Send mailbox of the CP	
Word	Contents
0	A53D$_H$
2	34E1$_H$
4	9002$_H$
6	8765$_H$
8	6ED4$_H$
10	0D0D$_H$

Figure 7-10. Forwarding Data with "Send Message Frame"

7.6.1 Sending Message Frames with the Message Frame Length Specification

You specify the message frame length in bytes with the Send request in word 2.

Note:

The length of the message frame sent by the CPU can vary from the length configured in parameter block 7 (7.3). The length configured in parameter block 7 refers to message frames received by the CP from a peripheral device.

Procedure for ”Send message frames with message frame length specification”

1. The CPU transfers the following in the transfer memory
   - The message frame length in bytes in word 2
   - The "Coordinate data transfer for sending message frames" job number into word 0.

2. The CPU reads the coordination information from word 0 of the transfer memory.
   If the CP has not accepted the job (bit 0 in byte 1= "0"), no message frame can be sent.
   The CPU can read the number of message blocks to be sent from byte 6 of the transfer memory. The number is calculated by the CP using the length specified in the job request. If you specify a message frame length of 12 bytes, for example, the CP stores the value 02_H in byte 6 for two message blocks.

3. CP transfers message frame in 8-byte message blocks to the transfer memory.

Note:

The CP stores the number of message frames in the Receive mailbox in byte 7, so you can also determine after issuing a Send request whether you can receive a message.

Example: Sending a message frame with a length of 12 bytes

Start address 128 is set on the CP. The message frame is stored in data block 20 beginning from DW 0. Flag 100.0 is set if the CP has accepted the job. The STL does not describe how F 100.0 is to be reset. You could reset F 100.0 when another job request has been forwarded to the CP.

STL FB 122	Explanation
`NAME:SEND2`	
`:A F 100.0`	The CP 523 has accepted the job if flag 100.0 is set,
`:BEC`	then Block End
`:L KH 000C`	Load length 12 bytes into ACCUM 1 and
`:T PW 130`	transfer to word 2 of the transfer memory
`:L KH A001`	Load "Send frame" job ($A001_H$) into ACCUM 1 and transfer to word 0
`:T PW 128*`	of the transfer memory
`:L KH 0001`	Check that Send request has been accepted
`:L PW 128`	
`:AW`	
`:><F`	
`:BEC`	If no, then Block End
`:S F 100.0`	If yes then set flag 100.0 and send message frame
`:C DB 20`	Open DB 20
`:L DW 3`	Transfer DW 3 to
`:T PW 134`	word 6 of the transfer memory
`:L DW 2`	Transfer DW 2 to
`:T PW 132`	word 4 of the transfer memory
`:L DW 1`	Transfer DW 1 to
`:T PW 130`	word 2 of the transfer memory
`:L DW 0`	Transfer DW 0 to
`:T PW 128*`	word 0 of the transfer memory
	The CP transfers the contents of the transfer memory to the Send mailbox
`:L DW 5`	Transfer DW 5 to
`:T PW 130`	word 2 of the transfer memory
`:L DW 4`	Transfer DW 4 to
`:T PW 128*`	word 0 of the transfer memory
`:BE`	The CP transfers the contents of words 2 and 0 in the transfer memory to the Send mailbox.
	The CP then sends the message frame from the Send mailbox to the peripheral device

* 5.2

7.6.2 Sending Message Frames with End-of-Text Character Specification

Specify one or two end-of-text characters with the Send request in word 4. The CP detects the end of a message frame autonomously through these end-of-text characters.

Note:

The end-of-text characters of the message frame sent by the CPU may differ from the end-of-text characters configured in parameter block 7 (7.3). The end-of-text characters configured in parameter block 7 refer to the message frames received by the CP from a peripheral device.

Procedure for sending message frames with end-of-text characters

1. The CPU transfers the following to the transfer memory.
   - The end-of-text characters (word 4)
   - Message frame length of 0 bytes (word 2)
   - The "Coordinate data transfer for sending message frame" job number (A001$_H$) (word 0).

2. The CPU reads the coordination information from word 0 of the transfer memory. If the CP has not accepted the job (bit 0 ="0"), no message frame can be sent.

3. The CPU transfers the message frame to the transfer memory in message blocks of eight bytes.

Note:

The CP stores the number of message frames in the Receive mailbox in byte 7, so you can also determine after a Send job whether you can receive a message frame.

Example: Sending a message frame with end-of-text characters 0D0D$_H$

Start address 128 is set on the CP. The message frame is stored in DB 20 beginning DW0. Flag 100.1 is set if the CP has accepted the job. The program does not describe how F 100.1 is reset. You could reset F 100.1 when another job request has been transferred to the CP. FB 123 can be assigned the relevant parameters. You can choose the start address of the module and end-of-text characters. They must be specified when the FB is invoked.

Table 7-33. Flags Used in FB 123

Flags	Meaning
F 100.1 F 100.2	Message frame transmitted End-of-text character flag set
Flag words	
FW 242 FW 244 FW 250 FW 252 FW 254	Data word pointer Auxiliary data word pointer Start address Current address of the transfer memory End-of-text character

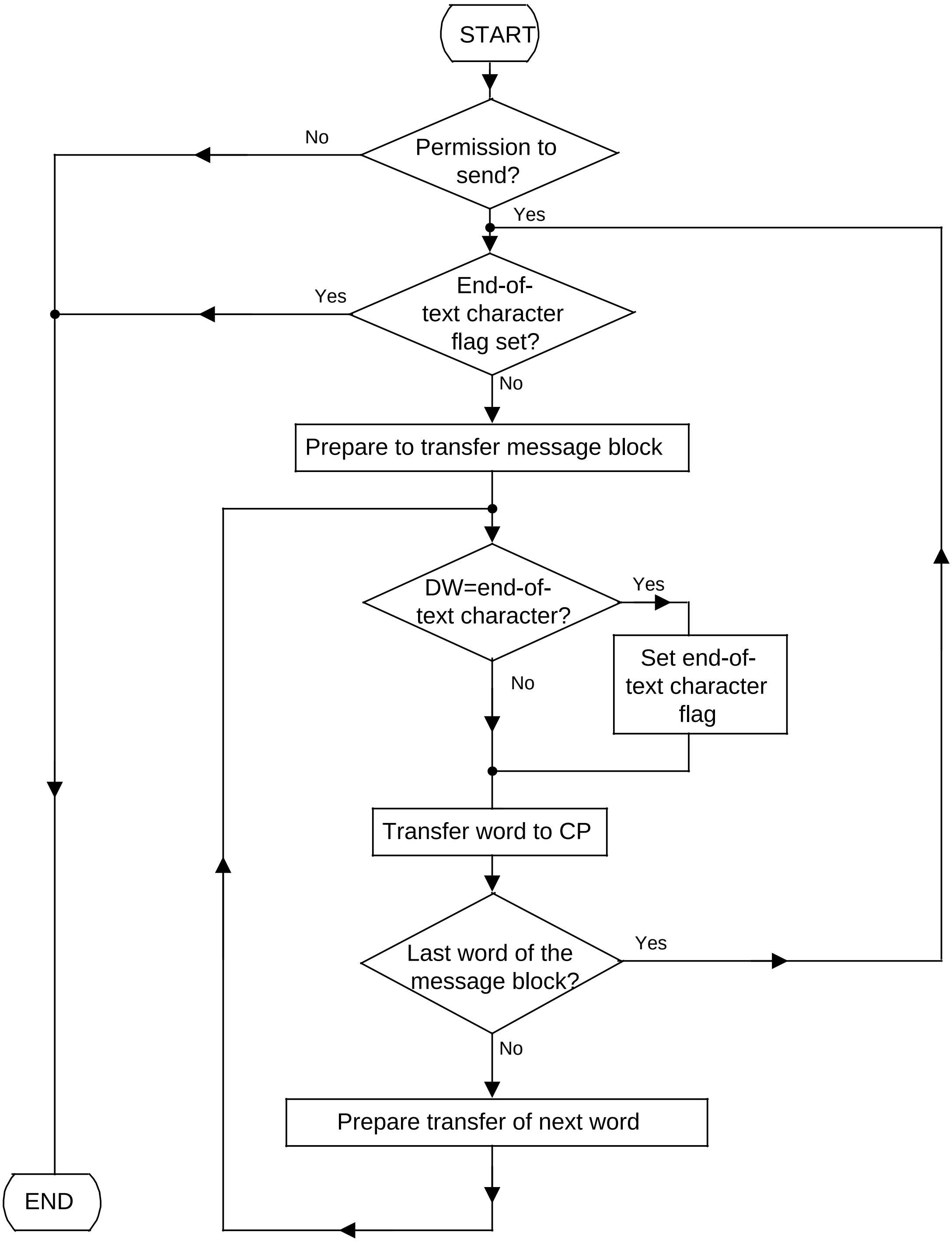

START
Permission to send?
No
Yes
End-of-text character flag set?
Yes
No
Prepare to transfer message block
DW=end-of-text character?
Yes
No
Set end-of-text character flag
Transfer word to CP
Last word of the message block?
Yes
No
Prepare transfer of next word
END

Figure 7-11. Flowchart for FB 123 "SEND 3"

FB 123 **LEN=103**

```
Segment 1    0000
NAME :SEND 3
DES  :BADR      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KF
DES  :ENZE      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KH
```

STL FB 123	Explanation
:C DB 2	Open source data block.
:A F 100.1	Has a message frame been sent?
:JC =M001	If yes, Block End.
:LW =BADR	Load module start address,
:L KF +4	Set byte 4 - address of the CP 523
:+F	and store.
:T FW 250	
:LW =ENZE	Load end-of-text character
:DO FW 250	and transfer to CP 523
:T PW 0	in bytes 4 and 5.
:L FW 250	Set byte 2 - address of the CP 523
:ADD KF -2	and store.
:T FW 250	
:L KF +0	Load message frame length and
:DO FW 250	transfer to the CP 523 in bytes 2 and 3.
:T PW 0	
:L FW 250	Set byte 0 - address of the CP 523 and
:ADD KF -2	store.
:T FW 250	
:L KH A001	Transfer "Send" coordination request
:DO FW 250	to the CP 523 in bytes 0 and 1.
:T PW 0	
:L KH 0001	Check acknowledgement of
:DO FW 250	coordination job from the CP 523.
:L PW 0	
:AW	Can data be sent?
:><F	If no,
:JC =M001	Block End.
:S F 100.1	Set message frame flag.
:LW =BADR	Load module start address
:T FW 250	and store.
:LW =ENZE	Load end-of-text character
:T FW 254	and store.
:L KH FFFF	Preset data word pointer.
:T FW 242	
:L KH FFFC	Preset auxiliary data word pointer.
:T FW 244	

STL FB 123	Explanation
`M003  :A   F  100.2`	If end-of-text character flag set,
`      :JC  =M001`	block end.
`      :L   FW 242`	Load data word pointer,
`      :ADD KF +4`	increment by 4
`      :T   FW 242`	and store.
`      :L   FW 244`	Load auxiliary data word pointer,
`      :ADD KF +4`	increment by 4
`      :T   FW 244`	and store.
`      :L   FW 250`	Set byte 6 - CP 523 address
`      :ADD KF +6`	and store.
`      :T   FW 252`	
`      :`	
`M004  :L   FW 254`	Load end-of-text character and
`      :DO  FW 242`	compare with the data word to be
`      :L   DW   0`	transferred.
`      :!=F`	If data word = end-of-text character,
`      :S   F  100.2`	set end-of-text character flag.
`      :DO  FW 252`	Transfer data word
`      :T   PW   0`	to CP 523.
`      :L   FW 242`	Transfer message block (8 data bytes) to
`      :L   FW 244`	the CP 523?
`      :><F`	If no,
`      :JC  =M002`	transfer next data word.
`      :L   FW 242`	Prepare
`      :ADD KF +3`	transfer
`      :T   FW 242`	of next message block.
`      :JU  =M003`	
`M002  :L   FW 242`	Set number of the next data word and
`      :ADD KF -1`	store.
`      :T   FW 242`	
`      :L   FW 252`	Set transfer area on the CP 523 and
`      :ADD KF -2`	store.
`      :T   FW 252`	
`      :JU  =M004`	
`M001  :`	
`      :BE`	Block End

7.6.3 Sending Message Frames with 3964(R) Protocol

In principle, data transfers between the CPU and the CP 523 are handled the same way as in Transparent mode. The additional return information about the protocol itself is entered in a so-called coordination byte (CBS, CBR).
CBS = Coordination byte 'send', CBR = coordination byte 'receive'.

You specify the message frame length in word 2 when you submit the request.

”Send message frames with the 3964(R) protocol”

1. The CPU forwards the following in the transfer memory:
   - The message frame length in bytes in word 2
   - The job number for ”Coordinate data transfer for sending message frame” in word 0.

2. The CPU reads the coordination info out of word 0 of the transfer memory.
   No message frame can be transmitted if the CP rejects the request (bit 0 in byte 1=”0”).
   Together with the coordination info, the following is made available to the CPU:
   - The coordination byte 'send' (CBS), including error flags, in byte 2
   - The number of message blocks comprising the frame in byte 6

   On the basis of the length specification given in the job request, the CP automatically computes the number of message blocks in the frame to be transmitted.
   If, for example, you specified a message frame length of 12 bytes, the CP enters the value 02_H in byte 6, i.e. 2 message blocks.

3. The CPU transfers the message frame in 8-byte message blocks to the transfer memory.

Note:

The CP enters the number of message frames in the Receive mailbox in byte 7, thus making it possible for you to ascertain whether you can receive a message frame even after a Send request has been submitted.

Example: Sending a 12-byte message frame with 3964(R) protocol

Base address 128 is set on the CP. The message frame is in data block 20 beginning data word DW 0. Flag 100.0 is set when the CP has accepted the Send request. The statement list does not show how or when this flag is reset. This could be done, for instance, after a new request has been submitted to the CP.

STL FB 124	Explanation
NAME:SEND2	
:L KH A000	”Check module status” request
:T PW 128*	
:L PY 130	
:T FY 110	Transfer CBS to flag byte 110 (error analysis)
:A F 100.0	The CP accepted the job request when flag 100.0 is set,
:BEC	then Block End
:L KH 000C	Load message frame size 12 bytes into ACCUM 1 and
:T PW 130	forward to word 2 from transfer memory
:L KH A001	Load ”Send message frame” request (A001$_H$) into
:T PW 128*	transfer to word 0/transfer memory
:L PY 130	
:T FY 110	Transfer CBS to flag byte 110
:L KH 0001	Check to see if request accepted
:L PW 128	
:AW	
:><F	
:BEC	If not, Block End conditional
:S F 100.0	If so, set flag 100.0 and send message frame
:C DB 20	Open DB 20
:L DW 3	DW 3 to
:T PW 134	transfer memory word 6
:L DW 2	DW 2 to
:T PW 132	transfer memory word 4
:L DW 1	DW 1 to
:T PW 130	transfer memory word 2
:L DW 0	DW 0 to
:T PW 128*	transfer memory word 0
	The CP forwards the contents of transfer memory to the Send mailbox
:L DW 5	DW 5 to
:T PW 130	transfer memory word 2
:L DW 4	DW 4 to
:T PW 128*	transfer memory word 0
:BE	The CP forwards words 2 and 0 of the transfer memory to the Send mailbox, then transmits the message frame from the Send mailbox to the I/O device.

* 5.2

7.7 Receiving a Message Frame from a Peripheral Device (A080$_H$)

The CPU can receive up to 256 bytes with a single job request. This data is referred to as a message frame in the following.

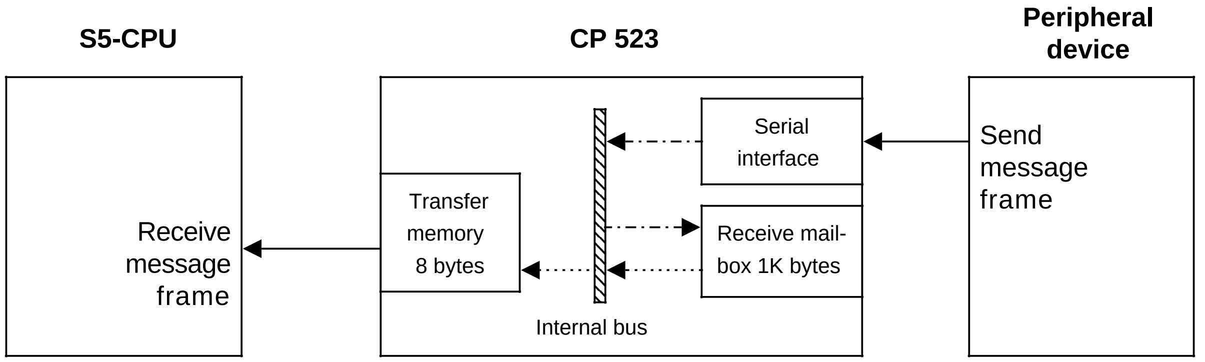


Figure 7-12. Receiving a Message from a Peripheral Device

The CP 523 receives message frames from the peripheral device via a serial interface and stores them in the Receive mailbox. A maximum of 99 message frames with a total length of 1024 bytes can be stored there.
When the CP 523 is receiving message frames from the peripheral device, it can simultaneously

- Receive a message frame from the CPU
- Send a message frame to the CPU

The CP outputs message frames to the CPU in the order in which they were received from the peripheral device. The CP 523 transfers new data from the Receive mailbox to the transfer memory if the CPU has written the relevant data into word 0 of the transfer memory.

Data transfer between the CPU and the CP is always initiated by the CPU with "Coordinate data transfer for receiving a message frame" request. As coordination information, the CP 523 indicates if there are message frames in the Receive mailbox (bit 7 = "1"). The CPU can also fetch message frames from the CP which have been fragmented because the character delay time (ZVZ) was exceeded. In these cases, all data received up until the instant of error is forwarded as message frame to the CPU (except in 3964(R) mode, in which case the error is flagged in the CBR).
The CP 523 transfers the following to the CPU, together with the coordination information:

- The length (in bytes) of the next message frame in word 2
- The number of message blocks in the next message frame in byte 6
- The number of message frames in the Receive mailbox in byte 7

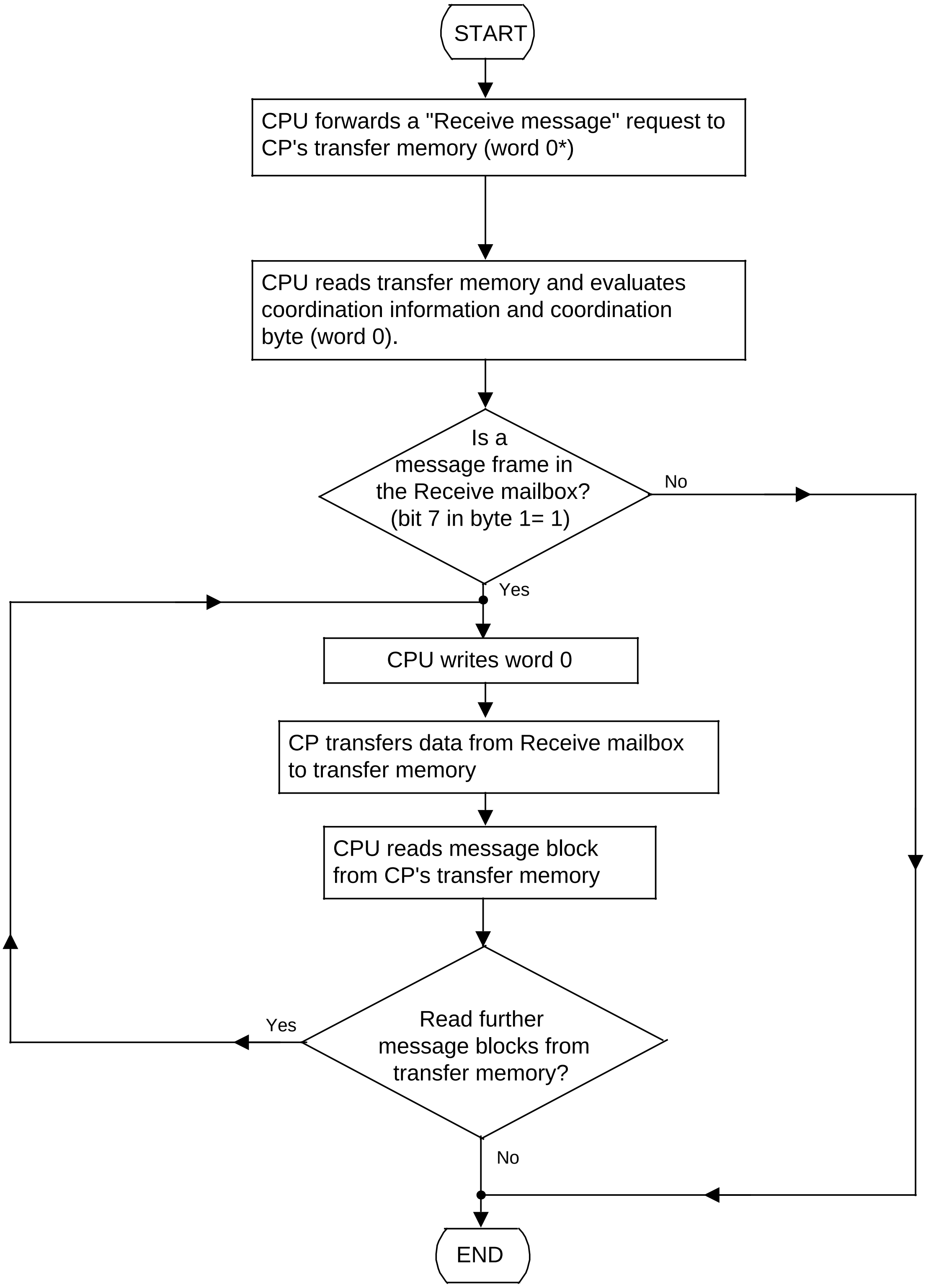
If a message frame has been received, the CPU can read it out in message blocks of eight bytes.

The next two pages contain a graphic representation of this procedure.

- "Receive message frame" flowchart
- Transfer memory assignments from the user program and from the CP 523 for a Receive request.

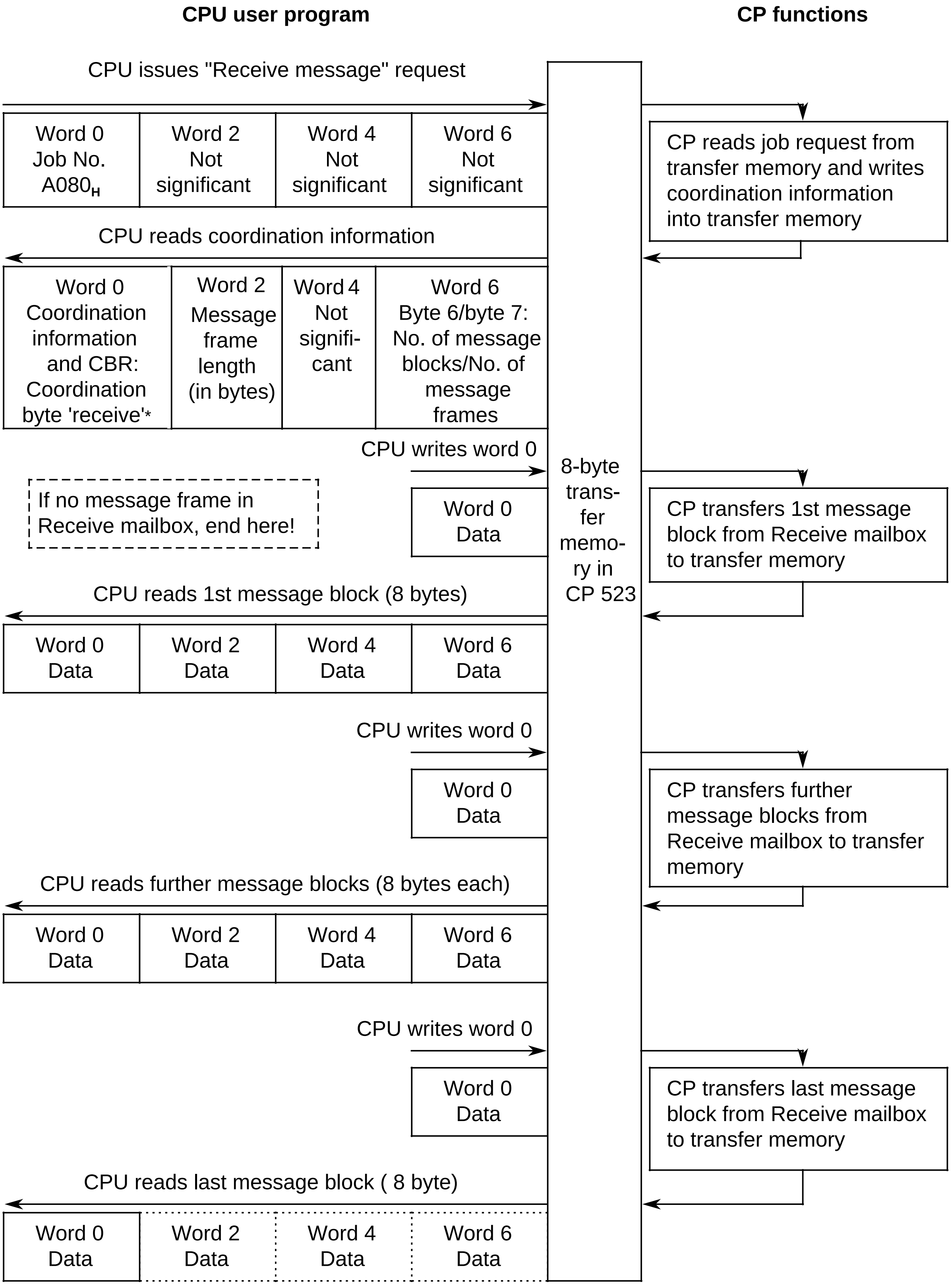
Note:

The statement list of the "RECEIVE" function block is explained in 8.2. The function block is a user-friendly interface for handling "Receive message frame". Use of the function block is also explained in 8.2.


START
CPU forwards a "Receive message" request to CP's transfer memory (word 0*)
CPU reads transfer memory and evaluates coordination information and coordination byte (word 0).
Is a message frame in the Receive mailbox? (bit 7 in byte 1= 1)
No
Yes
CPU writes word 0
CP transfers data from Receive mailbox to transfer memory
CPU reads message block from CP's transfer memory
Read further message blocks from transfer memory?
Yes
No
END

* 5.2

Figure 7-13. Schematic for "Receive Message"


CPU user program
CP functions
CPU issues "Receive message" request
Word 0 Job No. A080H
Word 2 Not significant
Word 4 Not significant
Word 6 Not significant
CP reads job request from transfer memory and writes coordination information into transfer memory
CPU reads coordination information
Word 0 Coordination information and CBR: Coordination byte 'receive'*
Word 2 Message frame length (in bytes)
Word 4 Not signifi- cant
Word 6 Byte 6/byte 7: No. of message blocks/No. of message frames
CPU writes word 0
8-byte trans- fer memo- ry in CP 523
If no message frame in Receive mailbox, end here!
Word 0 Data
CP transfers 1st message block from Receive mailbox to transfer memory
CPU reads 1st message block (8 bytes)
Word 0 Data
Word 2 Data
Word 4 Data
Word 6 Data
CPU writes word 0
Word 0 Data
CP transfers further message blocks from Receive mailbox to transfer memory
CPU reads further message blocks (8 bytes each)
Word 0 Data
Word 2 Data
Word 4 Data
Word 6 Data
CPU writes word 0
Word 0 Data
CP transfers last message block from Receive mailbox to transfer memory
CPU reads last message block (8 byte)
Word 0 Data
Word 2 Data
Word 4 Data
Word 6 Data

* Relevant only in 3964(R) mode.

Figure 7-14. Transfer Memory Assignments for "Receive Message Frame"

Message frame structure

A message frame is defined as all data transferred after a Send or Receive request. A message frame can be up to 256 bytes in length in the case of the CP 523. The CPU sends and receives message frames in message blocks of eight bytes.

The CP 523 can detect the end of a message frame sent by the peripheral device in three ways:

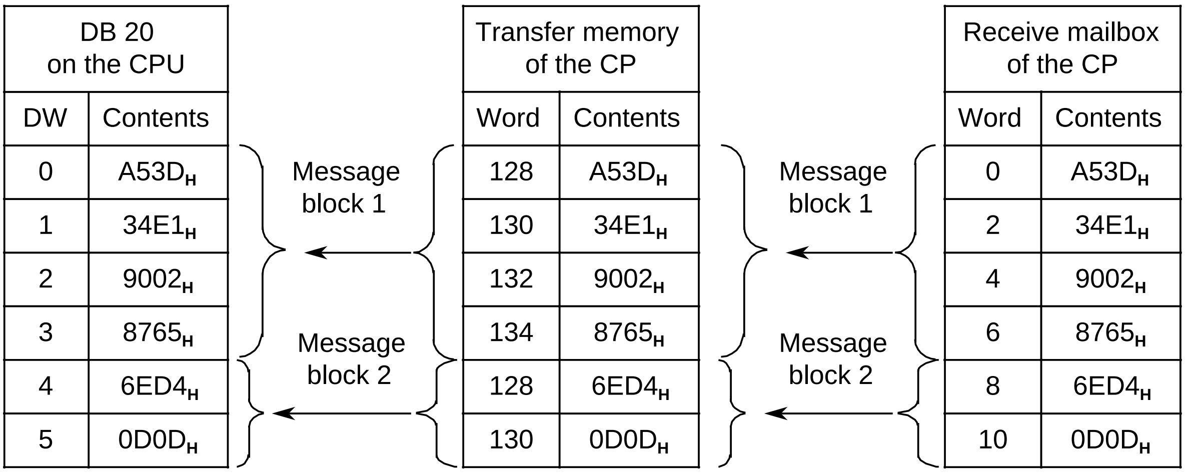
- You define the message frame length in parameter block 7 during configuring.
- You specify one or two end-of-text characters in parameter block 7 when you set the parameters in the parameter block. In this case, the message frame size must be 0 bytes.
  The CP 523 must be initialized for Interpretive mode.
  The setting of parameters is explained in 7.3.
- In 3964(R) mode, the procedure automatically computes the message frame size on the basis of end identifier string DLE ETX (BCC).

Transferring the message frame

After the CP 523 has accepted the Receive request, it transfers the first message block from the Receive mailbox to the transfer memory. After the CPU has read a message block, it must write word 0 of the transfer memory. The CP then updates the transfer memory.

Example: Receiving a message frame with a length of 12 bytes.

Start address 128 is set on the CP 523. The message frame is to be stored in data block 20 beginning DW 0.



DB 20 on the CPU	
DW	Contents
0	A53D$_H$
1	34E1$_H$
2	9002$_H$
3	8765$_H$
4	6ED4$_H$
5	0D0D$_H$

Transfer memory of the CP	
Word	Contents
128	A53D$_H$
130	34E1$_H$
132	9002$_H$
134	8765$_H$
128	6ED4$_H$
130	0D0D$_H$

Receive mailbox of the CP	
Word	Contents
0	A53D$_H$
2	34E1$_H$
4	9002$_H$
6	8765$_H$
8	6ED4$_H$
10	0D0D$_H$

Figure 7-15. Data Interchange with "Receive Message Frame"

7.7.1 Receiving Message Frames with Evaluation of the Specified Message Frame Length

You can receive a fixed-length message frame from the CP with a Receive request.

Prerequisites

- You must specify the fixed length in parameter block 7 when initializing the CP 523.
- You must also make sure that the peripheral device sends only message frames of this length to the CP.

Schematic for ”Receiving message frames with evaluation of the specified message frame length”

1. The CPU transfers the "Coordinate data transfer for receive message frame" request (A080$_H$) to word 0 of the transfer memory

2. The CPU reads the coordination information from word 0 of the transfer memory. If there is no message frame in the Receive mailbox (bit 7 in byte 1="0"), no message frame can be received. The following information is available in the transfer memory together with the coordination information:
   - Word 2: Length of the next message frame to be received (in bytes)
   - Byte 6: Number of message blocks in the next message frame to be received
   - Byte 7: Total number of message frames in the Receive buffer

3. The CPU writes word 0 of the transfer memory.
   The CP updates the transfer memory.

4. The CPU reads the message frame from the transfer memory in message blocks of eight bytes.
   The CPU reads the 1st message block from the transfer memory (eight bytes).

5. The CPU writes word 0 of the transfer memory.
   The CP updates the transfer memory.

6. CPU reads further message blocks from the transfer memory.

7. Steps 5 and 6 are repeated until the CPU has read the last message block.

Example: Receiving a message frame with a length of 12 bytes

Start address 128 is set on the module. The message frame is to be stored in data block 21 beginning DW0. Flag 101.0 is to be set if a message frame is in DB 21. The STL does not explain how flag 101.0 is reset. You could reset the flag when you have evaluated the message frame and it can be overwritten in DB 21.

STL FB 125	Explanation
NAME :RECEIVE2	
:A F 101.0	Flag 101.0 is set if a message frame is in DB 21
:BEC	Then conditional Block End
:L KH A080	Load "Receive message frame" request in ACCUM 1,
:T PW 128*	transfer to word 0 of the transfer memory
:L KH 0080	and check if the Receive mailbox is empty
:L PW 128	
:AW	Read and evaluate coordination information
:><F	If Receive mailbox is empty,
:BEC	then conditional Block End
:S F 101.0	If not empty, then set flag 101.0
:C DB 21	and open DB 21
:L KH 0000	
:T PW 128*	
:L PW 134	Read word 6 of the transfer memory
:T DW 3	and store in DW3
:L PW 132	Read word 4 of the transfer memory
:T DW 2	and store in DW2
:L PW 130	Read word 2 of the transfer memory
:T DW 1	and store in DW1
:L PW 128	Read word 0 of the transfer memory
:T DW 0	and store in DW0
:L KH 0000	
:T PW 128*	Write word 0 of the transfer memory
:L PW 130	Read word 2 of the transfer memory
:T DW 5	and store in DW5
:L PW 128	Read word 0 of the transfer memory
:T DW 4	and store in DW4
:BE	Block End

* 5.2

7.7.2 Receiving Message Frames with Evaluation of the Specified End-of-Text Characters

After a Receive request, you can receive a message frame from the CP which ends with end-of-text characters. The CP checks the end-of-text characters. You can evaluate the receive data in the user program.

Prerequisites

- You must specify the end-of-text characters in parameter block 7 when initializing the CP 523 (7.3)
- You must make sure that the peripheral device sends only message frames containing these end-of-text characters to the CP.
- The CP 523 must be set for Interpretive mode. You must define this in parameter block 7 at the initializing stage (7.3).

Schematic for ”Receive a message frame with evaluation of the specified end-of-text characters”

1. The CPU transfers the "Coordinate data transfer for receive message frame" request (A080$_H$) to word 0 of the transfer memory

2. The CPU reads the coordination information from word 0 of the transfer memory. If there is no message frame in the Receive mailbox (bit 7 in byte 1 = "0"), no message frame can be received. Byte 7 also contains the number of message frames in the Receive buffer.

3. The CPU reads the message frame from the transfer memory in message blocks of eight bytes.
   The CPU reads the message block from the transfer memory and checks that the end-of-text characters have been transferred.

4a. If end-of-text characters have been received, then Block End.
4b. If no end-of-text character was transferred, then write word 0 of the transfer memory.
The CP then updates the transfer memory.

5. Return to step 3.

Example: Receiving a message frame with the end-of-text character 0D0D$_H$

Start address 128 is set on the CP 523. The message frame is to be stored in DB 22 beginning DW 0. Flag 101.1 is to be set if a message frame is stored in DB 22. The STL does not explain how flag 101.1 is reset. You could reset the flag when you have evaluated the message frame and it can be overwritten in DB 22. FB 126 can be assigned the relevant parameters. You have a free choice of start address of the module and end-of-text characters. These must be specified when the FB is called. FB 126 receives only one message frame, and does not evaluate byte 7 of the transfer memory (number of message frames in the Receive mailbox).

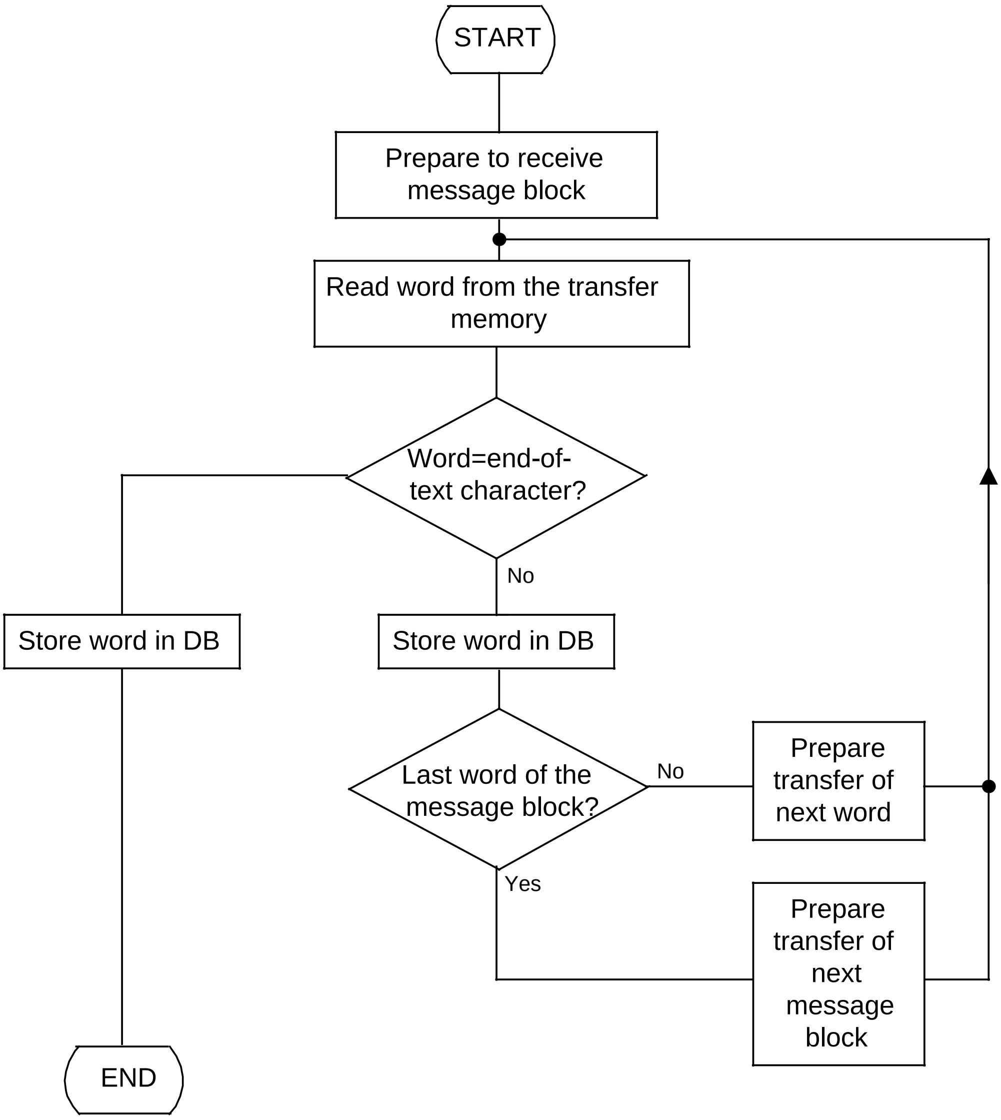

START
Prepare to receive message block
Read word from the transfer memory
Word=end-of-text character?
No
Store word in DB
Store word in DB
Last word of the message block?
No
Yes
Prepare transfer of next word
Prepare transfer of next message block
END

Figure 7-16. Flowchart for FB 126 "RECEIVE 3"

FB 126 **LEN=85**

```
NETWORK 1    0000
NAME :RECEIVE4
DES  :BADR      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KF
DES  :ENZE      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KH
```

STL FB 126	Explanation
:C DB 3	Open target data block.
M001 :A F 101.1	Message frame in DB?
:JC =M001	If yes, then Block End.
:LW =BADR	Load and store
:T FW 222	module start address.
:L KH A080	Load Receive request
:DO FW 222	
:T PW 0	and transfer to CP.
:L KH 0080	Message frame
:DO FW 222	
:L PW 0	in Receive mailbox?
:AW	
:><F	If no,
:JC =M001	then Block End.
:S F 101.1	Set message frame flag.
:LW =BADR	Load module address,
:ADD KF +6	set byte 6 address
:T FW 222	and store.
:LW =ENZE	Load end-of-text character
:T FW 224	and store.
:L KH 0000	Preset data word pointer.
:T FW 232	
:L KH 0003	Preset auxiliary data word pointer.
:T FW 234	
M003 :	
:LW =BADR	Load module address
:T FW 220	and store.
:L KH 0000	Request next message block
:DO FW 220	by writing byte 0.
:T PW 0	
M004 :L FW 224	Is the next data byte equal to
:DO FW 220	the end-of-text character?
:L PW 0	
:!=F	If yes,
:S F 101.2	set End flag.

STL FB 126	Explanation
:DO FW 232	Store data bytes
:T DW 0	in destination DB.
:A F 101.2	Block End if
:JC =M001	end-of-text character detected.
:L FW 232	Has the message block
:L FW 234	been completely transferred?
:><F	If no, read in two further
:JC =M002	data bytes.
:L FW 234	Set auxiliary data word pointer
:ADD KF +4	for the next message block.
:T FW 234	
:L FW 232	Set auxiliary data word pointer
:ADD KF +1	for the first two data bytes
:T FW 232	of the next message block.
:JU =M003	Read in next message block.
M002 :L FW 232	Increment data word pointer by 1.
:ADD KF +1	
:T FW 232	
:L FW 220	Increment current address
:ADD KF +2	on the CP by 2.
:T FW 220	
:JC =M004	Read in next word
M001 :	of message block.
:BE	Block End

7.7.3 Receiving Message Frames with 3964(R) Protocol

Basically, data transfers between the CPU and the CP 523 are handled the same way as in Transparent mode. The additional return information pertaining to the itself is entered in a coordination byte (CBS, CBR).
CBS = coordination byte ”send”, CBR = coordination byte ”receive”.

Schematic for ”Receive message frames with the 3964(R) protocol”

1. The CPU forwards a ”Coordinate data transfer for receive message frame” request (job number A080$_{H}$) in word 0 of the transfer memory.

2. The CPU reads out the coordination info from word 0 of the transfer memory, and the CBR (coordination byte ”receive”). A message frame can be received only when the Receive mailbox contains one (bit 7 of the CBR = ”0” when there is no message frame in the Receive mailbox). Together with the coordination info, the following information is made available to the user in the transfer memory:
   - Word 2: Length (in bytes) of the message frame that is to be received next
   - Byte 6: Number of message blocks in the next message frame to be received
   - Byte 7: Total number of message frames in the Receive buffer

3. The CPU writes to word 0 of the transfer memory.
   The CP updates the transfer memory.

4. The CPU reads the message frame out of the transfer memory in message blocks of 8 bytes each.
   The CPU reads the first message block (8 bytes) out of the transfer memory.

5. The CPU writes to word 0 of the transfer memory.
   The CP updates the transfer memory.

6. The CPU reads out the next message block.

7. Steps 5 and 6 are repeated until the CPU has read out the last message block.

Example: Receiving a 12-byte message frame with 3964(R) protocol

Base address 128 is set on the CP. The message frame is to be stored in data block 21 beginning DW 0. Flag 101.0 is to be set as soon as a message frame is entered in data block 21. The statement list does not explain how or when flag 101.0 is reset. This can be done, for example, when the message frame has been evaluated and data block 21 can be overwritten.

STL FB 127	Explanation
`NAME :RECEIVE2`	
`:A    F    101.0`	Flag 101.0 is set when a message frame has been entered in DB 21,
`:BEC`	then Block End conditional
`:L    KH   A080`	Load ”Receive message frame” request into ACCUM 1
`:T    PW   128*`	Transfer to word 0 in transfer memory
`:L    PY   129`	Load CBR into flag byte 111 (for error
`:T    FY   111`	analysis if necessary)
`:L    KH   0080`	Check to see if Receive mailbox is empty
`:L    PW   128`	
`:AW`	Read and analyze coordination info
`:><F`	When Receive mailbox is empty,
`:BEC`	then Block End conditional.
`:S    F    101.0`	If it is not, set flag 101.0 and
`:C    DB   21`	open DB 21.
`:L    KH    0000`	
`:T    PW   128*`	
`:L    PW   134`	Read transfer memory word 6 and
`:T    DW   3`	transfer to DW 3
`:L    PW   132`	Read transfer memory word 4 and
`:T    DW   2`	transfer to DW 2
`:L    PW   130`	Read transfer memory word 2 and
`:T    DW   1`	transfer to DW 1
`:L    PW   128`	Read transfer memory word 0 and
`:T    DW   0`	transfer to DW 0
`:L    KH   0000`	
`:T    PW   128*`	Write to transfer memory word 0
`:L    PW   130`	Read transfer memory word 2 and
`:T    DW   5`	transfer to DW 5
`:L    PW   128`	Read transfer memory word 0 and
`:T    DW   4`	transfer to DW 4
`:BE`	

* 5.2

8 Using the "SEND" and "RECEIVE" Function Blocks

Figures

Tables

8 Using the "SEND" and "RECEIVE" Function Blocks

The "SEND" and "RECEIVE" function blocks, FB 200 and FB 201, offer you a user-friendly interface which considerably simplifies use of "Send message frame to peripheral device" job (A001$_H$) and "Receive message frame from peripheral device" (A080$_H$). You can initiate the transfer of a message frame immediately by calling FB 200 or FB 201 instead of having to program the send or receive program yourself.

To transfer messages frames, you need only assign the relevant FB parameters. The next two sections (8.1 and 8.1.1) describe FB parameter setting. Transfer of a message frame is initiated by calling FB 200 or FB 201. The FBs have a status byte containing information on the CP, the peripheral device and data transfer.

Note :

These FBs can execute on CPU 941 to CPU 944. They can also execute in the 135U, 150U and 155U if network 3 is omitted (or ”reprogrammed” for other PLCs). Normally, network 3 can be omitted when sufficient space is provided in the source and destination DBs.

8.1 Using "SEND" FB 200

"SEND" FB 200 transfers a message frame of fixed or variable length from the CPU to the peripheral device. You must store the message frame to be transferred in a data block (source data block) before calling "SEND" FB 200. When you invoke the "SEND" FB 200, you must specify the following:

- The start address of the CP 523
- The number of the source data block from which the message frame is to be transferred to the CP 523
- The number of the source data word beginning in which the message frame is stored.
- The length of the message frame to be transferred (number of source data words) or of two end-of-text characters
- A byte address where "SEND" FB 200 is to store the status byte.
- A byte address where "SEND" FB 200 is to store the coordination byte for send (CBS) (3964(R) only).
- Specification as to whether with or without computer-computer link.

Send is initiated when the RLO is 1 when FB 200 is invoked.

8.1.1 Calling "SEND" FB 200

"SEND" FB 200 can be called in cyclic programs and in programs servicing timed* interrupts and process* interrupts.

STL	LAD / CSF

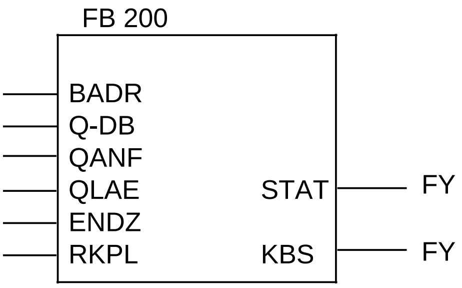
`              : JU   FB 200`
`NAME           : SEND`
`DES:           : BADR : KF`
`DES:           : Q-DB : B`
`DES:           : QANF : KF`
`DES:           : QLAE : KF`
`DES:           : ENDZ : KH`
`DES:           : STAT : FY`
`DES:           : KBS  : FY`
`DES:           : RKPL : BI`
	 FB 200
	BADR
	Q-DB
	QANF
	QLAE
	ENDZ
	RKPL
	STAT
	KBS
	FY
	FY

Table 8-1. Block Parameters for "SEND" FB 200

Name (Identif.)	Parameter Type	Data Type	Description
BADR	D	KF	Start address of the module
Q-DB	B		Number of the source data block
QANF	D	KF	Number of the first source data word
QLAE	I	BY	Number of data words to be transferred (message frame length: max. 128 data words): Forwarding of a number in KF format in the relevant byte
ENDZ	D	KH	Hexadecimal pattern for the end-of-text character (ASCII character)
STAT	Q	BY	Status byte
KBS	Q	BY	Coordination byte 'send' (CBS)
RKPL	I	BI	0: For Interpretive and Transparent mode (no flagging in CBS) 1: For 3964(R) mode (flagging in CBS)

* Please note that FB 200 uses scratch flags.

8.1.2 Setting ”SEND” FB 200 Parameters

If you want to send a message frame of specific size, specify the length of the message frame to be sent (in words) in the QLAE parameter. The "ENDZ" parameter is not significant here. In 3964(R) mode, the message frame size must be specified.

If you want to send a message frame with end-of-text characters (variable length), specify "zero" in the QLAE parameter and two end-of-text characters in the "ENDZ" parameter. Both end-of-text characters must be in the same data word.

Example: End-of-text character 0D $0D_H$

valid:	DW n	KH:	44 44	invalid:	DW n	KH:	44 0D
	DW n+1	KH:	0D 0D		DW n+1	KH:	0D 00

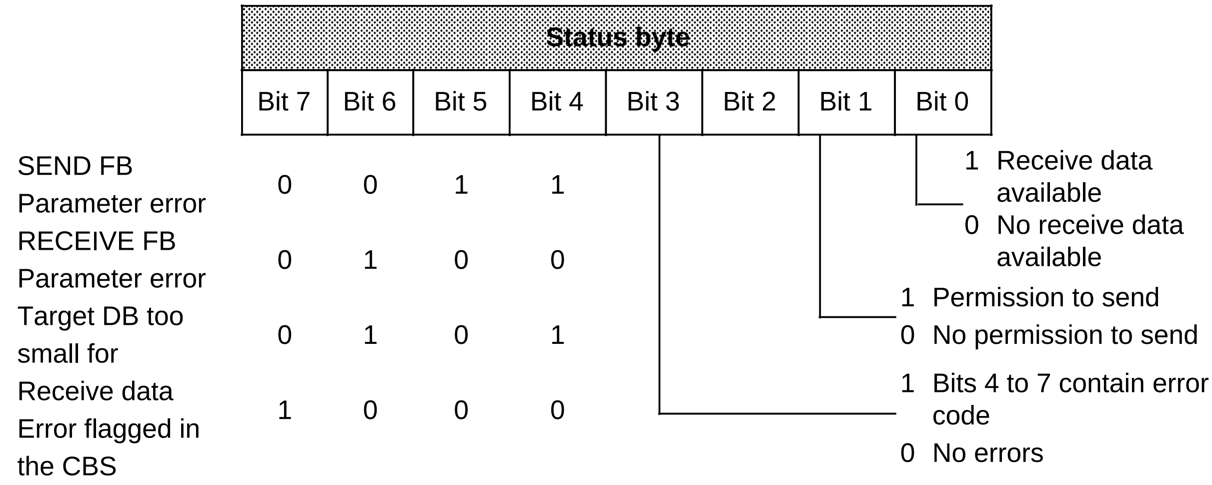
Table 8-2. Setting the "ENDZ" and "QLAE" Parameters

ENDZ (End-of-text char.)	QLAE (Frame length)	Transmission mode
= 0	>< 0	Message frame of specific length
>< 0	>< 0	Message frame of specific length
= 0	= 0	NOT PERMISSIBLE
>< 0	= 0	Message frame with end-of-text characters

8.1.3 Structure of the Status Byte

The status bytes of "SEND" FB 200 and "RECEIVE" FB 201 are identical. The status byte gives information on whether

- Receive data is available (bit 0 = 1)
- The Send mailbox is empty (bit 1 = 1)
  If the Send mailbox is occupied, the Send job initiated by calling "SEND" FB 200 can not be executed.
- Errors have occurred (bit 3 = 1)
  If bit 3 is set, bits 4 to 7 contain the error code.



Status byte

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SEND FB Parameter error	0	0	1	1				
RECEIVE FB Parameter error	0	1	0	0				
Target DB too small for Receive data	0	1	0	1				
Error flagged in the CBS	1	0	0	0				

Bit 0:
1 Receive data available
0 No receive data available

Bit 1:
1 Permission to send
0 No permission to send

Bit 3:
1 Bits 4 to 7 contain error code
0 No errors

Figure 8-1. Structure of the "STAT" Status Byte

8.1.4 Schematic of the Principle of Operation of ”SEND” FB 200

Note:

This section is intended merely to explain the method of operation of "SEND" FB 200. You do not require this information to use the "SEND" FB 200.

FB 200 services the Send request according to the following schematic:

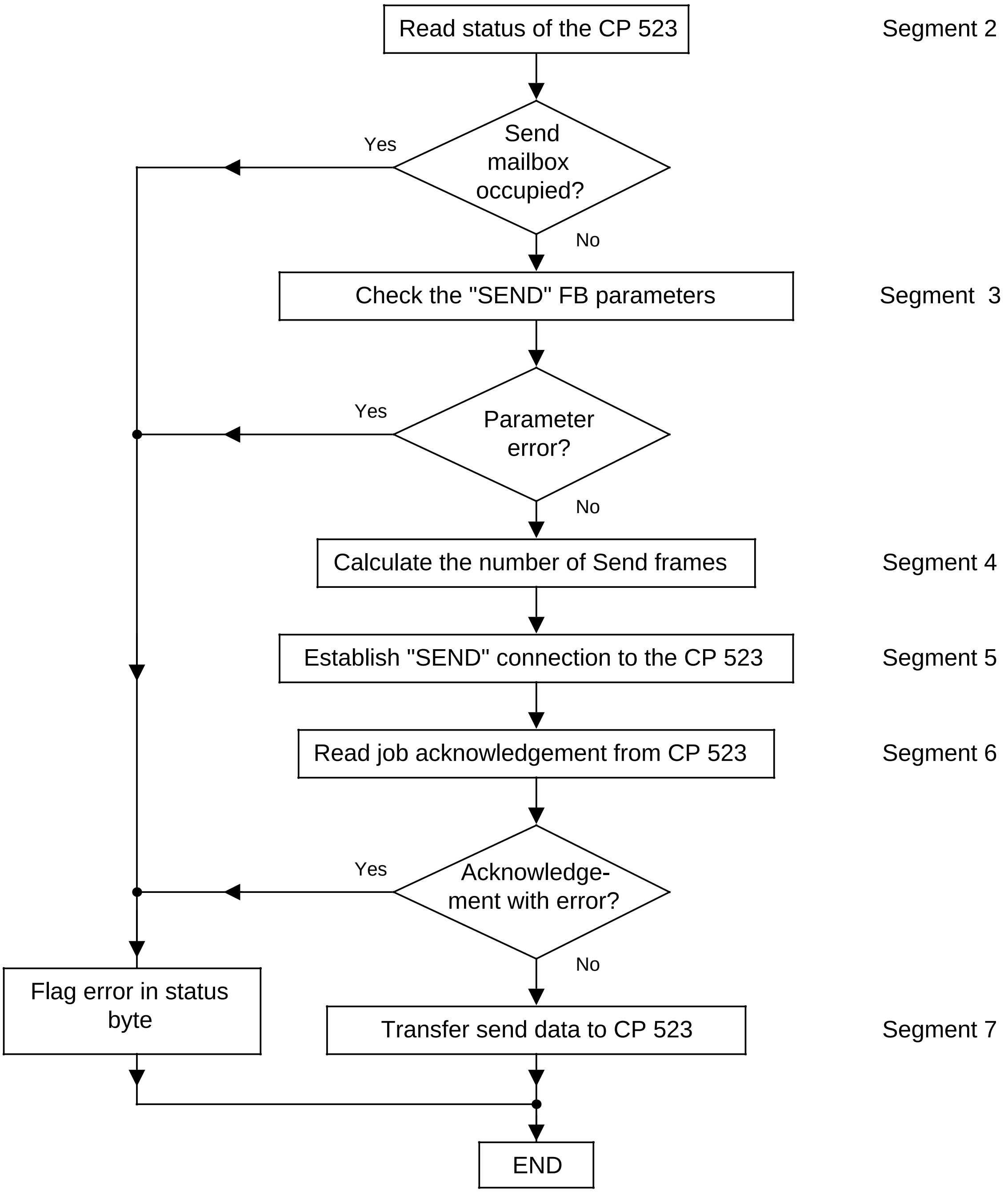
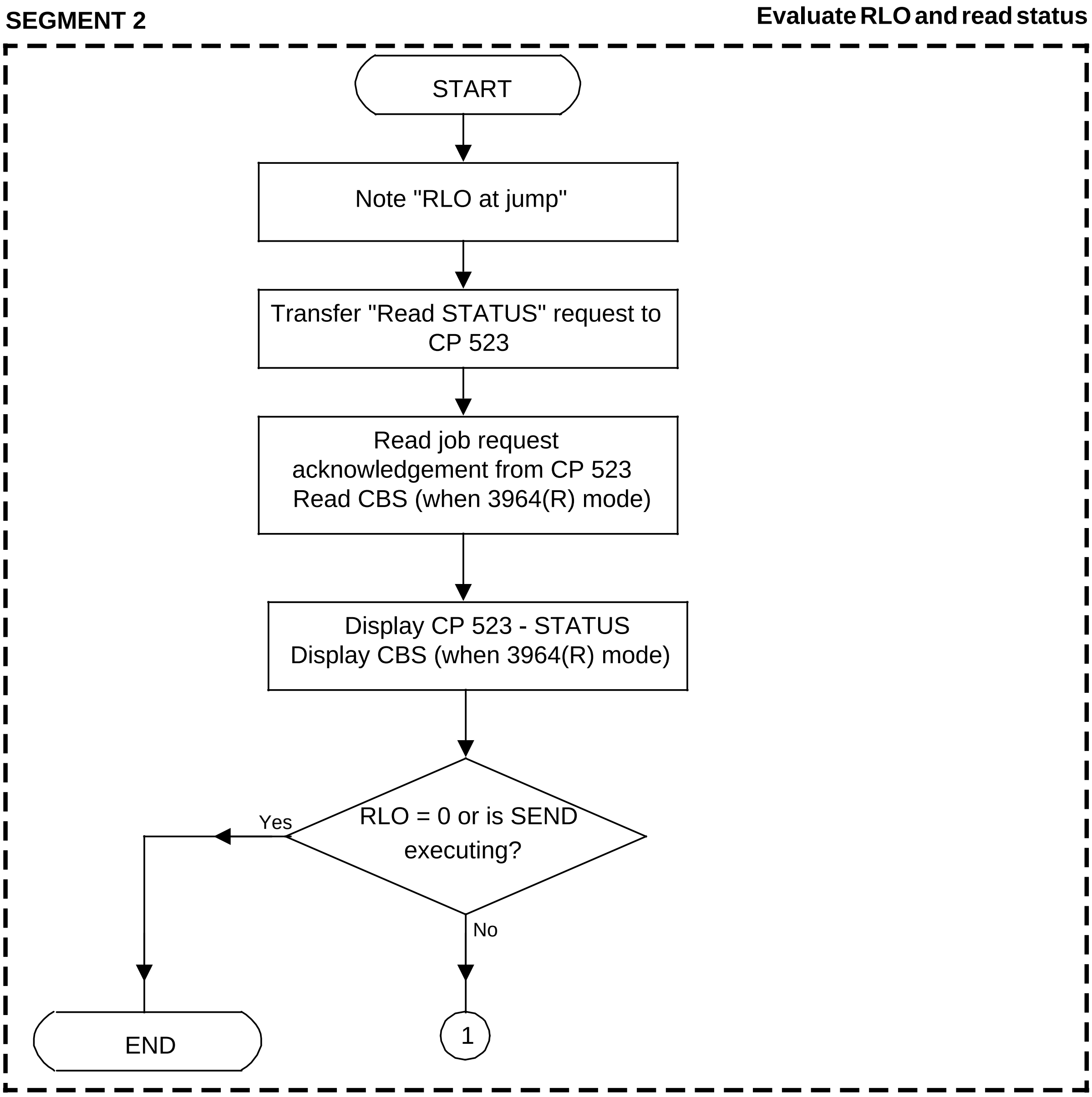
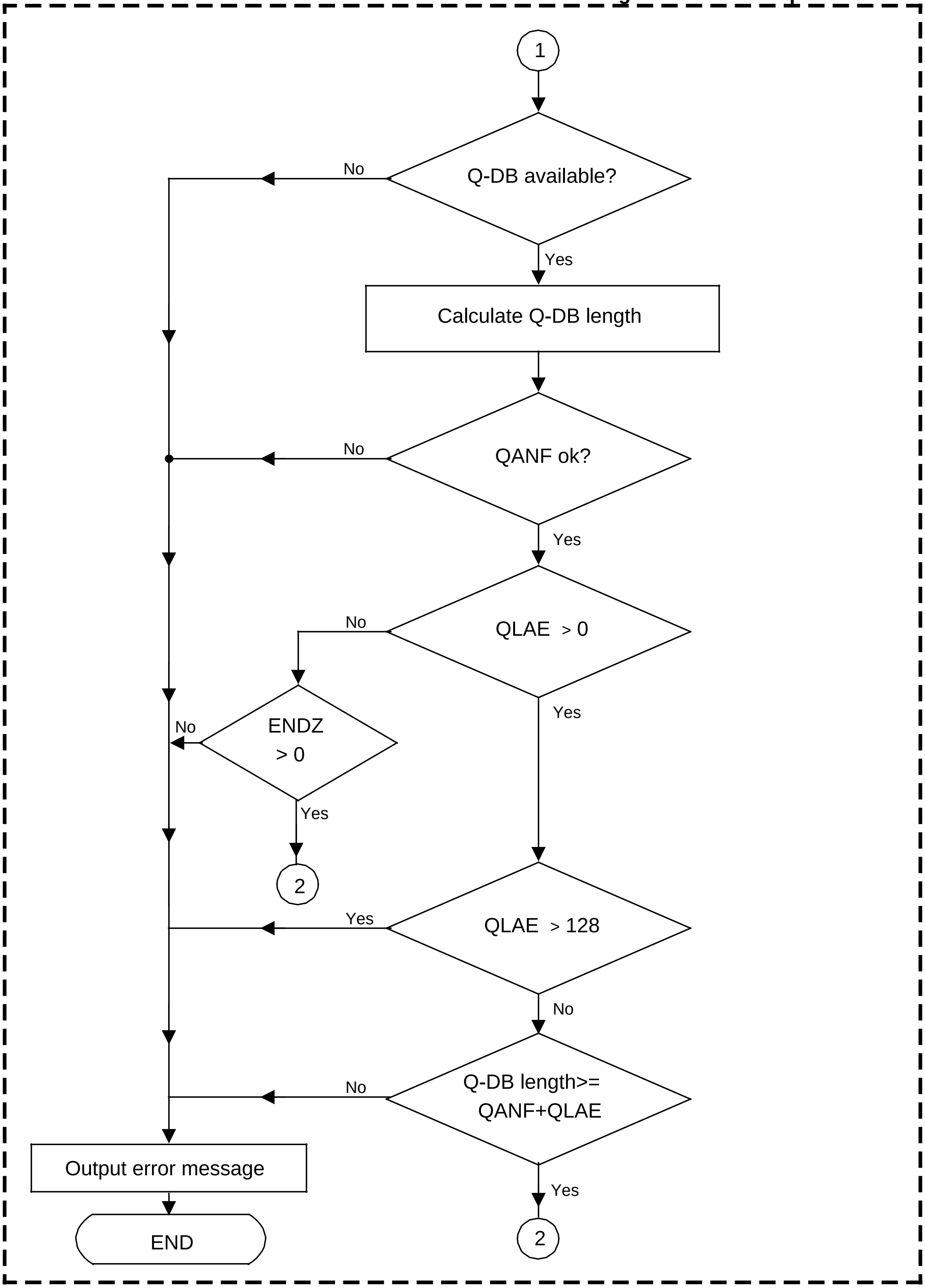

Read status of the CP 523
Segment 2
Send mailbox occupied?
Yes
No
Check the "SEND" FB parameters
Segment 3
Parameter error?
Yes
No
Calculate the number of Send frames
Segment 4
Establish "SEND" connection to the CP 523
Segment 5
Read job acknowledgement from CP 523
Segment 6
Acknowledge-ment with error?
Yes
No
Flag error in status byte
Transfer send data to CP 523
Segment 7
END

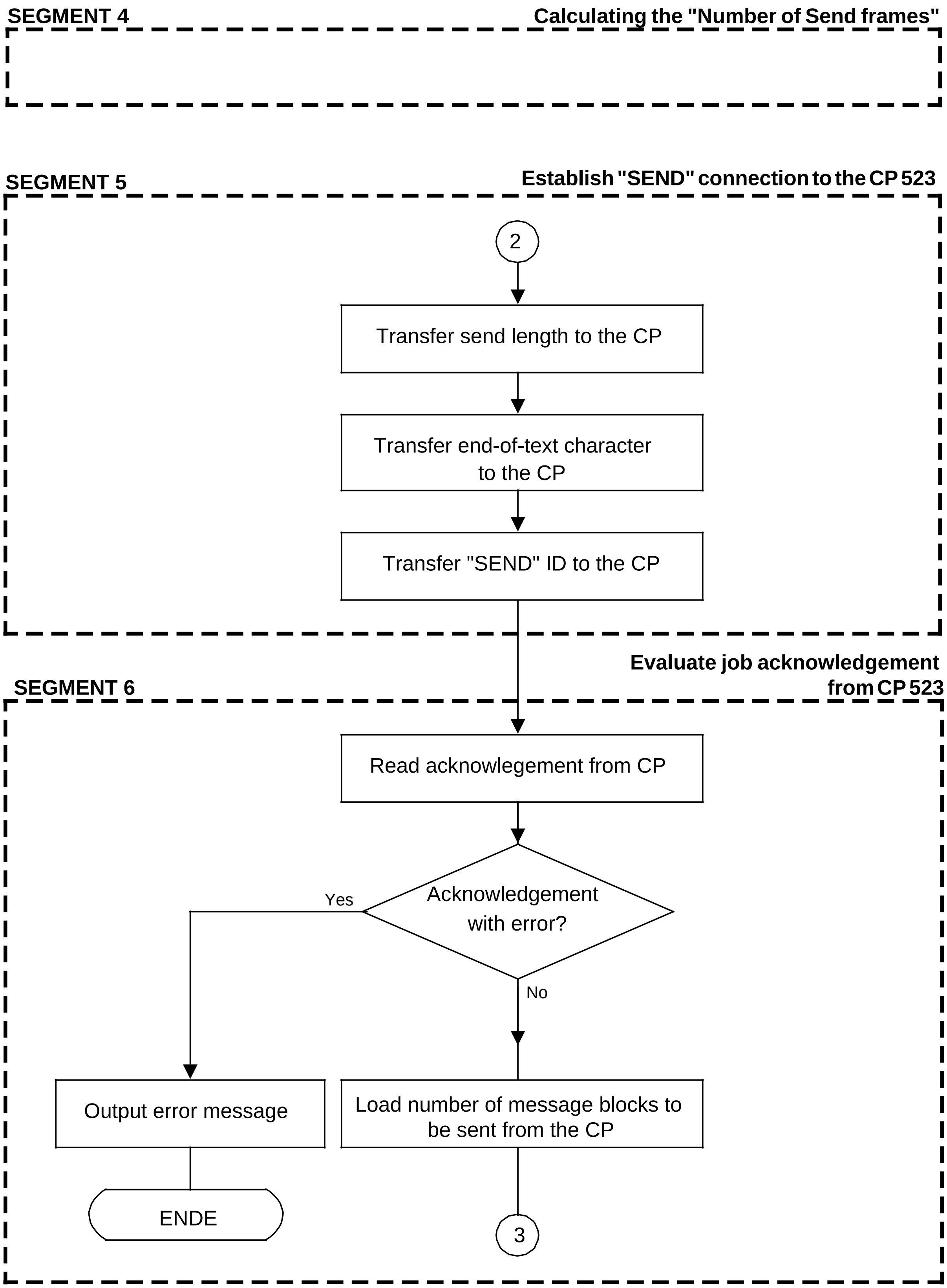
Figure 8-2. Schematic for "Send Frame" (FB 200)

8.1.5 Flowchart of "SEND" FB 200

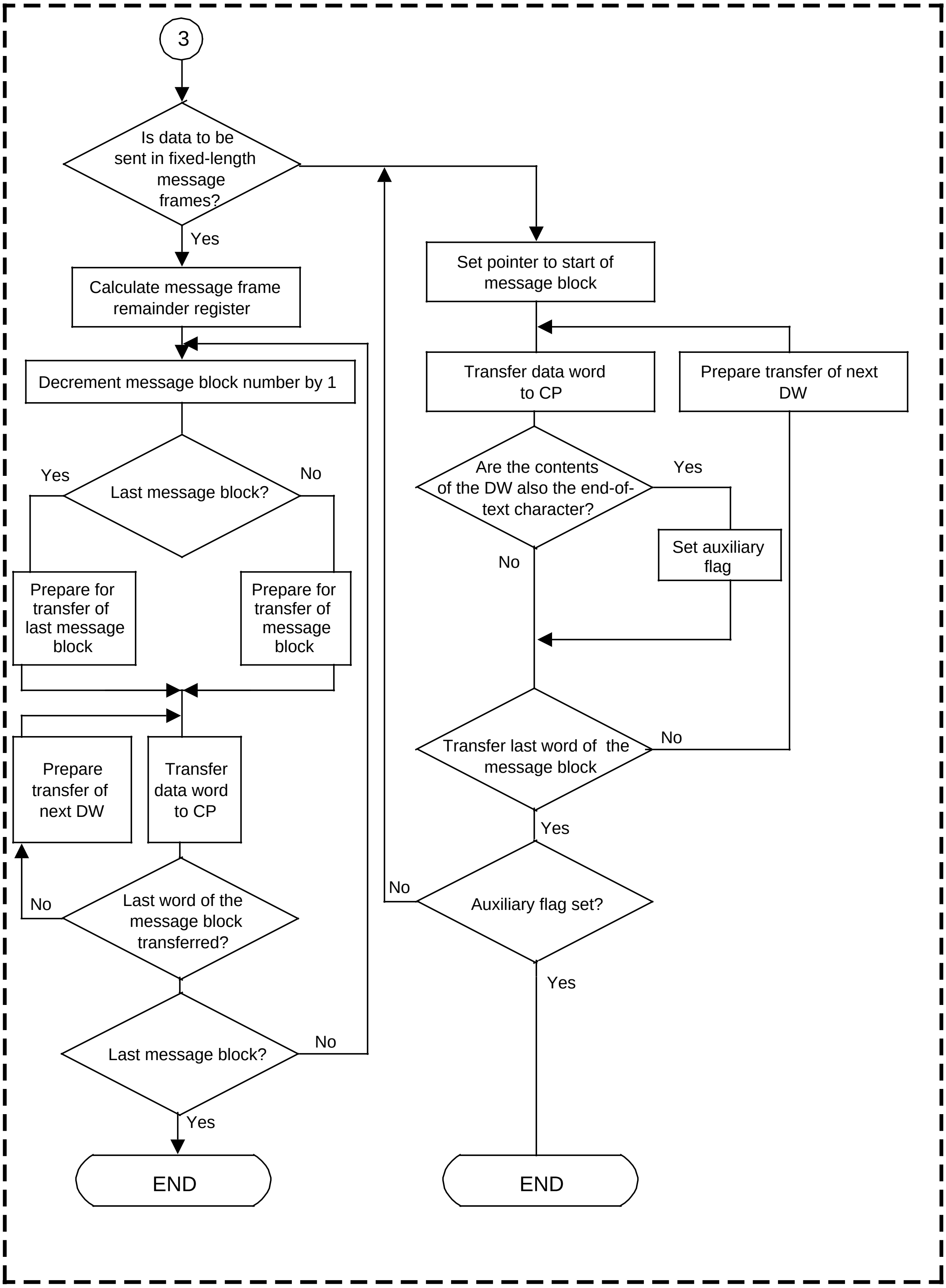

SEGMENT 2
Evaluate RLO and read status
START
Note "RLO at jump"
Transfer "Read STATUS" request to CP 523
Read job request acknowledgement from CP 523 Read CBS (when 3964(R) mode)
Display CP 523 - STATUS Display CBS (when 3964(R) mode)
RLO = 0 or is SEND executing?
Yes
No
END
1

SEGMENT 3 **Checking the ”SEND” FB parameters**




SEGMENT 4
Calculating the "Number of Send frames"
SEGMENT 5
Establish "SEND" connection to the CP 523
2
Transfer send length to the CP
Transfer end-of-text character to the CP
Transfer "SEND" ID to the CP
Evaluate job acknowledgement from CP 523
SEGMENT 6
Read acknowlegement from CP
Acknowledgement with error?
Yes
No
Output error message
Load number of message blocks to be sent from the CP
ENDE
3

SEGMENT 7 **Transferring Send data to the CP 523**


3
Is data to be sent in fixed-length message frames?
Yes
Calculate message frame remainder register
Decrement message block number by 1
Yes
Last message block?
No
Prepare for transfer of last message block
Prepare for transfer of message block
Prepare transfer of next DW
Transfer data word to CP
No
Last word of the message block transferred?
Last message block?
No
Yes
END
Set pointer to start of message block
Transfer data word to CP
Prepare transfer of next DW
Are the contents of the DW also the end-of-text character?
Yes
Set auxiliary flag
No
Transfer last word of the message block
No
Yes
No
Auxiliary flag set?
Yes
END

8.1.6 Flags in "SEND" FB 200

Explanation of the flags used by "SEND" FB:

Flags	Symbol	Meaning
F 255.0	H-FLAG-1	Auxiliary flag for buffering the RLO at jump
F 254.0	H-FLAG-2	Auxiliary flag for STAT bit "SEND executing"
F 237.0	H-FLAG-3	Auxiliary flag for detecting end-of-text characters
FY 238.0	H-BYTE-2	Auxiliary flag byte for buffering CBS
FY 254	H-BYTE-1	Auxiliary flag byte for buffering STAT
FW 238	HILFREG	Auxiliary flag word for buffering the data words to be transferred
FW 240	B-AN-S	"Number of send frames" register
FW 242	BADR-REG	Module address register (DO FW register for CP access)
FW 244	QANF-REG	QANF register (auxiliary register data word block pointer for Q-DW-IND)
FW 246	Q-DW-IND	Q data word index register (DO FW register for fetching data words from the Q-DB)
FW 248	B-REST-R	Frame remainder register (number of data words in the last message block)
FW 250	BYTE1IND	Byte 1 index register (CP byte address register for data transfer between CPU and CP, DO FW register)
FW 252	BYTE2IND	Byte 2 index register (CP byte address register for data transfer between CPU and CP, DO FW register)
FW 254	HILFSREG	Auxiliary flag word for calculating Q-DB lengths

FB 200	LEN=371

```
Segment 1     0000
NAME :SEND
DES  :BADR      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KF
DES  :Q-DB      I/Q/D/B/T/C: B
DES  :QANF      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KF
DES  :QLAE      I/Q/D/B/T/C: I  BI/BY/W/D: BY
DES  :ENDZ      I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KH
DES  :STAT      I/Q/D/B/T/C: Q  BI/BY/W/D: BY
DES  :KBS       I/Q/D/B/T/C: Q  BI/BY/W/D: BY
DES  :RKPL      I/Q/D/B/T/C: I  BI/BY/W/D: BI
001  :***
```

STL FB 200	Explanation
Segment2 001E	
001E :	-------------------
001F :	For CPUs
0020 :	941
0021 :	942
0022 :	943
0023 :	944
0024 :	-------------------
0025 := F 255.0	Save RLO
0026 :	
0027 :LW =BADR	Load CP 523 module address
0028 :T FW 242	and store
0029 :	
002A :	Write KOOR ID
002B :L KH 00A0	”Read status”
002D :DO FW 242	to CP 523 word 0
002E :T PY 0	
002F :L FW 242	
0030 :I 1	
0031 :T FW 242	
0032 :L KH 0000	
0034 :DO FW 242	
0035 :T PY 0	
0036 :	
0037 :DO FW 242	Fetch acknowledgement
0038 :L PY 0	”Read status”
0039 :T FY 254	from CP 523 and store
003A :	
003B :L KF +0	Reset auxiliary flag byte
003D :T FY 238	
003E :	

STL FB 200	Explanation
`003F      :AN  =RKPL`	no computer link
`0040      :JC  =NORK`	
`0041      :L   FW 242`	Set pointer to base
`0042      :I        1`	address + 2 (byte 2 = CBS
`0043      :T   FW 242`	for computer link)
`0044      :`	
`0045      :DO  FW 242`	Read CBS (when computer link)
`0046      :L   PY   0`	
`0047      :T   =KBS`	output and
`0048      :T   FY 238`	buffer
`0049      :`	
`004A      :L   FW 242`	Set pointer back to module
`004B      :D        1`	base address + 1
`004C      :T   FW 242`	
`004D NORK :`	
`004E      :L   FW 242`	Set pointer back to
`004F      :D        1`	module base address
`0050      :T   FW 242`	
`0051      :`	
`0052      :A   F   254.0`	Does CP523 report ”send in progress”?
`0053      :JC  =NEIN`	”NO”
`0054      :`	
`0055      :L   KH 0002`	Load STAT bit ”Send in progress ”
`0057      :JU  =STAT`	jump to STAT
`0058 NEIN :`	
`0059      :L   KH 0000`	STAT bit ”Send mailbox empty”
`005B STAT :`	
`005C      :T   =STAT`	Output STAT flag
`005D      :`	
`005E      :L   FY 238`	Error code in CBS
`005F      :L   KH 007F`	(when computer link)?
`0061      :AW`	
`0062      :L   KB 0`	
`0063      :!=F`	
`0064      :JC  =KBOK`	CBS ok
`0065      :L   KH 0088`	Output error code for
`0067      :T   =STAT`	”CBS error / computer link”
`0068      :`	
`0069 KBOK :`	
`006A      :AN   F 255.0`	RLO = ”0” on jump?
`006B      :ON   F 254.0`	or Send mailbox occupied?
`006C      :BEC`	”End” of program
`006D      :`	
`006E      :***`	

STL FB 200	Explanation
Segment 3 006F	
006F :	Check whether Q DB available
0070 :LW =Q-DB	Load Q DB parameter
0071 :SLW 8	Isolate DB number from
0072 :SRW 7	Q DB parameter
0073 :	
0074 :L KH E400	Calculate start address
0076 :+F	of the Q DB using
0077 :LIR 0	the block address list
0078 :	
0079 :L KB 0	Q DB available?
007A :!=F	
007B :JC =FE03	" E R R O R "
007C :	
007D :TAK	Calculate the length of the Q DB
007E :ADD KF -2	
0080 :LIR 0	
0081 :ADD KF -5	
0083 :T FW 254	and buffer
0084 :	
0085 :	Check the QANF
0086 :LW =QANF	QANF less than 0 ?
0087 :L KB 0	
0088 :<F	
0089 :JC =FE03	" E R R O R "
008A :	
008B :TAK	QANF > 255 ?
008C :L KB 255	
008D :>F	
008E :JC =FE03	" E R R O R "
008F :	
0090 :	Check QLAE
0091 :L =QLAE	QLAE > 0 ?
0092 :L KB 0	
0093 :>F	
0094 :JC =CONT	" C O N T I N U E "
0095 :	
0096 :LW =ENDZ	Check end-of-text,
0097 :L KH 0000	characters for > 0 ?
0099 :>F	
009A :JC =PAOK	"P A R A M E T E R S O.K."
009B :JU =FE03	otherwise "E R R O R "
009C :	
009D CONT :	
009E :L =QLAE	QLAE > 128?

STL FB 200	Explanation
`009F      :L   KB 128`	
`00A0      :>F`	
`00A1      :JC  =FE03`	"ERROR"
`00A2      :`	
`00A3      :`	Check to make sure that
`00A4      :LW  =QANF`	QANF+QLAE not > Q-DB length
`00A5      :LW  =QLAE`	
`00A6      :+F`	
`00A7      :L   FW 254`	Load calculated Q-DB length
`00A8      :>F`	
`00A9      :JC  =FE03`	"ERROR"
`00AA      :TAK`	QANF+QLAE<=255?
`00AB      :L   KF +255`	
`00AD      :<=F`	
`00AE      :JC  =PAOK`	Then OK
`00AF FE03 :`	
`00B0      :L   KH 0038`	Output "FB PARAMETER
`00B2      :T   =STAT`	ERROR"
`00B3      :`	
`00B4      :BEU`	Program "END"
`00B5 PAOK :`	
`00B6      :***`	
`Segment 4      00B7`	
`00B7      :`	
`00B8      :***`	
`Segment 5      00B9`	
`00B9      :`	
`00BA      :L   FW 242`	Increment pointer to
`00BB      :I       2`	base address +2 (byte 2)
`00BC      :T   FW 242`	
`00BD      :`	
`00BE      :L   =QLAE`	Write length (bytes) of the
`00BF      :SLW      1`	frame to be transferred
`00C0      :DO  FW 242`	to bytes 2 and 3
`00C1      :T   PW   0`	of the CP 523
`00C2      :`	
`00C3      :L   FW 242`	Set pointer to base
`00C4      :I        2`	address + 4 (byte 4)
`00C5      :T   FW 242`	and store
`00C6      :`	
`00C7      :LW  =ENDZ`	Write end-of-text characters 1 and 2
`00C8      :DO  FW 242`	to bytes 4 and 5
`00C9      :T   PW   0`	of the CP 523

STL FB 200	Explanation
00CA :	
00CB :L FW 242	Set pointer to base address
00CC :D 4	of the CP 523
00CD :T FW 242	and store
00CE :	
00CF :	
00D0 :L KH 00A0	Write KOOR ID "SEND"
00D2 :DO FW 242	to word 0
00D3 :T PY 0	of the CP 523
00D4 :L FW 242	
00D5 :I 1	
00D6 :T FW 242	
00D7 :L KH 0001	
00D9 :DO FW 242	
00DA :T PY 0	
00DB :	
00DC :L FW 242	Set pointer back
00DD :D 1	to base address
00DE :T FW 242	
00DF :***	
Segment 6 00E0	
00E0 :DO FW 242	Read status byte 0 from CP 523
00E1 :L PY 0	
00E2 :SLW 4	Isolate error bits
00E3 :T FY 254	and buffer
00E4 :	
00E5 :L KF +0	Reset aux. flag byte
00E7 :T FY 238	
00E8 :	
00E9 :AN =RKPL	No computer link
00EA :JC =NORK	
00EB :	
00EC :L FW 242	Increment pointer to base address + 2
00ED :I 2	(byte 2 = CBS when
00EE :T FW 242	computer link)
00EF :	
00F0 :DO FW 242	Read CBS,
00F1 :L PY 0	
00F2 :T =KBS	output
00F3 :T FY 238	and buffer
00F4 :	
00F5 :L FW 242	Set pointer back to
00F6 :D 2	base address

STL FB 200	Explanation
00F7 :T FW 242	
00F8 :	
00F9 :L KB 0	Error in status byte
00FA :L FY 254	
00FB :!= F	
00FC :JC = NEIN	”NO ERROR”
00FD :	
00FE :S F 254.3	Set error flag for STAT
00FF :L FY 254	
0100 :T =STAT	Output STAT flags
0101 :	
0102 :BEU	Program ”END”
0103 NEIN :	
0104 :L FY 238	Contains CBS error flags (when computer link)
0105 :L KH 007F	
0107 :AW	
0108 :L KB 0	
0109 :!=F	
010A :JC =KBOK	CBS ok
010B :L KH 0088	Output error ”Computer link error in CBS”
010D :T =STAT	
010E :	
010F KBOK :	
0110 :	
0111 :L FW 242	Set pointer to base address +6
0112 :ADD KF +6	
0114 :T FW 242	
0115 :	
0116 :DO FW 242	The CP loads and stores no. of message blocks to be sent
0117 :L PY 0	
0118 :T FW 240	
0119 :	
011A :L FW 242	Set pointer back to base address
011B :ADD KF -6	
011D :T FW 242	
011E :	
011F :***	
NETWORK 7 0120	
0120 :A F 237.0	Reset aux. flag for Send with EOT-characters
0121 :R F 237.0	
0122 :	
0123 :LW =QANF	Load QANF parameter and store (QANF register)
0124 :T FW 244	
0125 :	

STL FB 200	Explanation
0126 :L =QLAE	
0127 :L KH 0000	
0129 :>F	
012A :JC =TELE	Jump to Send with fixed length
012B :	
012C :	
012D :	***
012E :	* SEND WITH END-OF-TEXT CHARACTERS *
012F :	***
0130 BLOC :	
0131 :L FW 244	Increment QANF register by 3
0132 :ADD KF +3	
0134 :T FW 244	and store
0135 :	
0136 :T FW 246	QANF register to DWINDEX register
0137 :	
0138 :L FW 242	Set pointer to base address +6
0139 :ADD KF +6	(byte 6)
013B :T FW 250	and store
013C :	
013D :DO =Q-DB	Open source DB
013E :	
013F WORT :	
0140 :DO FW 246	Load the data word to be transferred
0141 :L DW 0	into the ACCUM
0142 :T FW 238	and buffer
0143 :L FY 238	
0144 :DO FW 250	Transfer 1st word to CP 523
0145 :T PY 0	
0146 :L FW 250	
0147 :I 1	
0148 :T FW 250	
0149 :L FY 239	
014A :DO FW 250	
014B :T PY 0	
014C :L FW 250	
014D :D 1	
014E :T FW 250	
014F :	
0150 :L FW 238	Check whether
0151 :LW =ENDZ	word transferred is
0152 :!=F	EOT char.
0153 :S F 237.0	Set aux. flag for EOT char.
0154 :	recognition
0155 :L FW 250	Last word of message block

STL FB 200	Explanation
0156 :L FW 242	transferred?
0157 :><F	
0158 :JC =VORB	”Preparations” for next message block
0159 :	
015A :A F 237.0	Did this message block contain the end-
015B :JC =END	of-text character?->”END”
015C :	
015D :L FW 244	Increment QANF register by 1
015E :ADD KF +1	
0160 :T FW 244	and store
0161 :	
0162 :JU =BLOC	”NEXT MESSAGE BLOCK”
0163 :	
0164 VORB :	
0165 :L FW 250	Decrement current CP 523
0166 :D 2	address by 2
0167 :T FW 250	and store
0168 :	
0169 :L FW 246	Decrement data word index
016A :D 1	register by 1
016B :T FW 246	and store
016C :JU =WORT	”NEXT WORD”
016E :	
016F :	****************************
0170 :	* SEND WITH FIXED FRAME *
0171 :	* LENGTH *
0172 :	****************************
0173 :	
0174 TELE :	Compute frame remainder register:
0175 :	
0176 :L FW 240	Number of message blocks * 4
0177 :SLW 2	(number in words)
0178 :L =QLAE	Buffer number of words
0179 :-F	Subtract message frame length from
017A :T FW 248	number of message frames (in words)
017B :	and store
017C :L KH 0004	
017E :L FW 248	
017F :-F	
0180 :T FW 248	----------------------------
0181 :	
0182 NBLO :	
0183 :L FW 240	Decrement number of message blocks by 1
0184 :D 1	

STL FB 200	Explanations
0185 :T FW 240	and store
0186 :	
0187 :L KB 0	Is the message block to be transferred
0188 :!=F	the last subframe?
0189 :JC =LBLO	”LAST MESSAGE BLOCK”
018A :	
018B :L FW 244	Increment QANF register by 3
018C :ADD KF +3	
018E :T FW 244	and store
018F :	
0190 :T FW 246	QANF register to DWINDEX register
0191 :	
0192 :L FW 242	Set pointer to base
0193 :ADD KF +6	address +6 (byte 6)
0195 :T FW 250	and store
0196 :	
0197 :JU =JUM1	
0198 :	
0199 LBLO :	
019A :L FW 244	Calculate data word index register
019B :L FW 248	
019C :D 1	
019D :+F	
019E :T FW 246	and store
019F :	
01A0 :L FW 248	Compute corresponding
01A1 :SLW 1	byte address
01A2 :D 2	
01A3 :L FW 242	
01A4 :+F	
01A5 :T FW 250	and store
01A6 :	
01A7 JUM1 :	
01A8 :DO =Q-DB	Open source DB
01A9 NWOR :	
01AA :DO FW 246	Load data byte to be transferred
01AB :L DL 0	into ACCUM
01AC :DO FW 250	
01AD :T PY 0	and transfer to CP
01AE :L FW 250	Increment pointer
01AF :I 1	
01B0 :T FW 250	
01B1 :DO FW 246	Load data byte to
01B2 :L DR 0	into ACCUM
01B3 :DO FW 250	

STL FB 200	Explanations
`01B4      :T   PY   0`	and forward to CP
`01B5      :L   FW 250`	Decrement pointer
`01B6      :D        1`	
`01B7      :T   FW 250`	
`01B8      :`	
`01B9      :L   FW 250`	Last word of message block
`01BA      :L   FW 242`	forwarded?
`01BB      :><F`	
`01BC      :JC  =NADR`	”NEXT ADDRESS”
`01BD      :`	
`01BE      :L   FW 240`	Is message block
`01BF      :L   KB   0`	to be transferred
`01C0      :!=F`	the last message block?
`01C1      :JC  =END`	”END”
`01C2      :`	
`01C3      :L   FW 244`	Increment QANF register by 1
`01C4      :ADD KF + 1`	
`01C6      :T   FW 244`	and store
`01C7      :`	
`01C8      :JU  =NBLO`	”NEXT MESSAGE BLOCK”
`01C9 NADR :`	
`01CA      :L   FW 250`	Decrement pointer for byte
`01CB      :D        2`	address by 2
`01CC      :T   FW 250`	and store
`01CD      :`	
`01CE      :L   FW 246`	Decrement data word
`01CF      :D        1`	index register by 1
`01D0      :T   FW 246`	and store
`01D1      :`	
`01D2      :JU  =NWOR`	”NEXT WORD”
`01D3 END  :`	
`01D4      :L   KH 0002`	Load STAT bit ”SEND in progress”
`01D6      :T =STAT`	Output STAT bit
`01D7      :`	
`01D8      :A   F 237.0`	Reset auxiliary flag
`01D9      :R   F 237.0`	for end-of-text character
`01DA      :`	
`01DB      :BE`	

8.2 Using "RECEIVE" FB 201

"RECEIVE" FB 201 transfers message frames from the CP 523 to the CPU.
You must specify the following when calling "RECEIVE" FB 201:

- The start address of the CP 523
- The number of the target data block in which the message frame is to be stored on the CP 523
- The number of the target data word starting from which the message frame is to be stored
- A byte address where "RECEIVE" FB 201 can store the number of data words received
- A byte address where "RECEIVE" FB 201 is to store the status byte
- A byte address where ”RECEIVE” FB 201 is to store the CBR (= KBE)

Receive enable is RLO = 1.

8.2.1 Calling and Setting "RECEIVE" FB 201 Parameters

"RECEIVE" FB 201 can be called in cyclic programs and in programs servicing timed interrupts* and process interrupts*.

STL	LAD/CSF

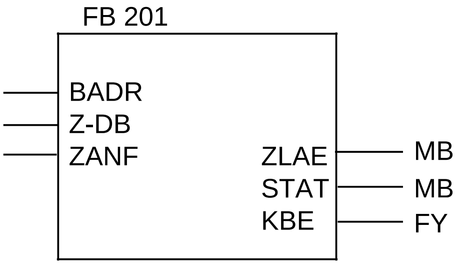
`       : JU   FB 201`
`NAME   : RECEIVE`
`DES    : BADR : KF`
`DES    : Z-DB : B`
`DES    : ZANF : KF`
`DES    : ZLAE : MB`
`DES    : STAT : MB`
`DES    : KBE`
	

Table 8-3. Block Parameters for "RECEIVE" FB 201

Name	Parameter Type	Data Type	Description
BADR	D	KF	Initial address of the module
Z-DB	B		Number of the destination data block
ZANF	D	KF	Number of the 1st destination data word
ZLAE	Q	BY	Number of data words received (destination length)
STAT	Q	BY	Status byte
KBE	Q	BY	Coordination byte 'receive' (relevant only in 3964(R) mode)

Note:

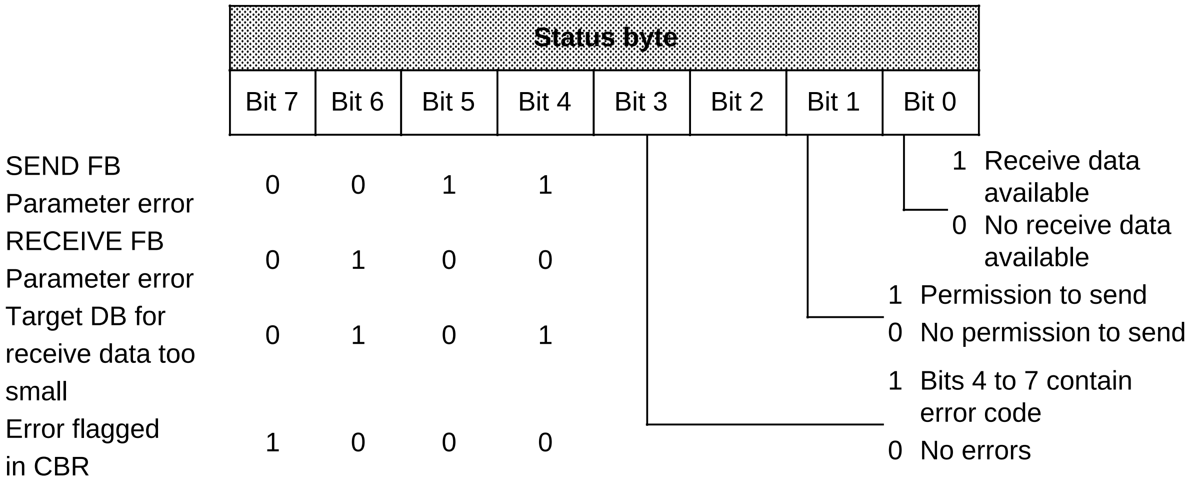
The scan time may be exceeded if several message blocks are received.
Remedy: Program OB 31 "Set scan time" in FB 201 (address 00F0, network 6).

* Please note that FB 201 uses scratch flags.

8.2.2 Structure of the Status Byte

The status bytes for "SEND" FB 200 and "RECEIVE" FB 201 are identical. The status byte contains information about whether:

- Receive data is available (bit 0 = 1)
- The Send mailbox is empty (bit 1 = 1)
  If the Send mailbox is occupied, the Send job initiated by calling "SEND" FB 200 is not executed.
- Errors have occurred (bit 3 = 1)
  If bit 3 is set, bits 4 to 7 contain the error code.



Status byte

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SEND FB Parameter error	0	0	1	1				
RECEIVE FB Parameter error	0	1	0	0				
Target DB for receive data too small	0	1	0	1				
Error flagged in CBR	1	0	0	0				

Bit 0:
1 Receive data available
0 No receive data available

Bit 1:
1 Permission to send
0 No permission to send

Bit 3:
1 Bits 4 to 7 contain error code
0 No errors

Figure 8-3. Structure of the "STAT" Status Byte

8.2.3 Schematic Representation of the Method of Operation of "RECEIVE" FB 201

Note:

This section is intended merely to explain how "RECEIVE" FB 201 works. You do not require this information to use "RECEIVE" FB 201.

The FB 201 services the Receive request according to the following schematic:

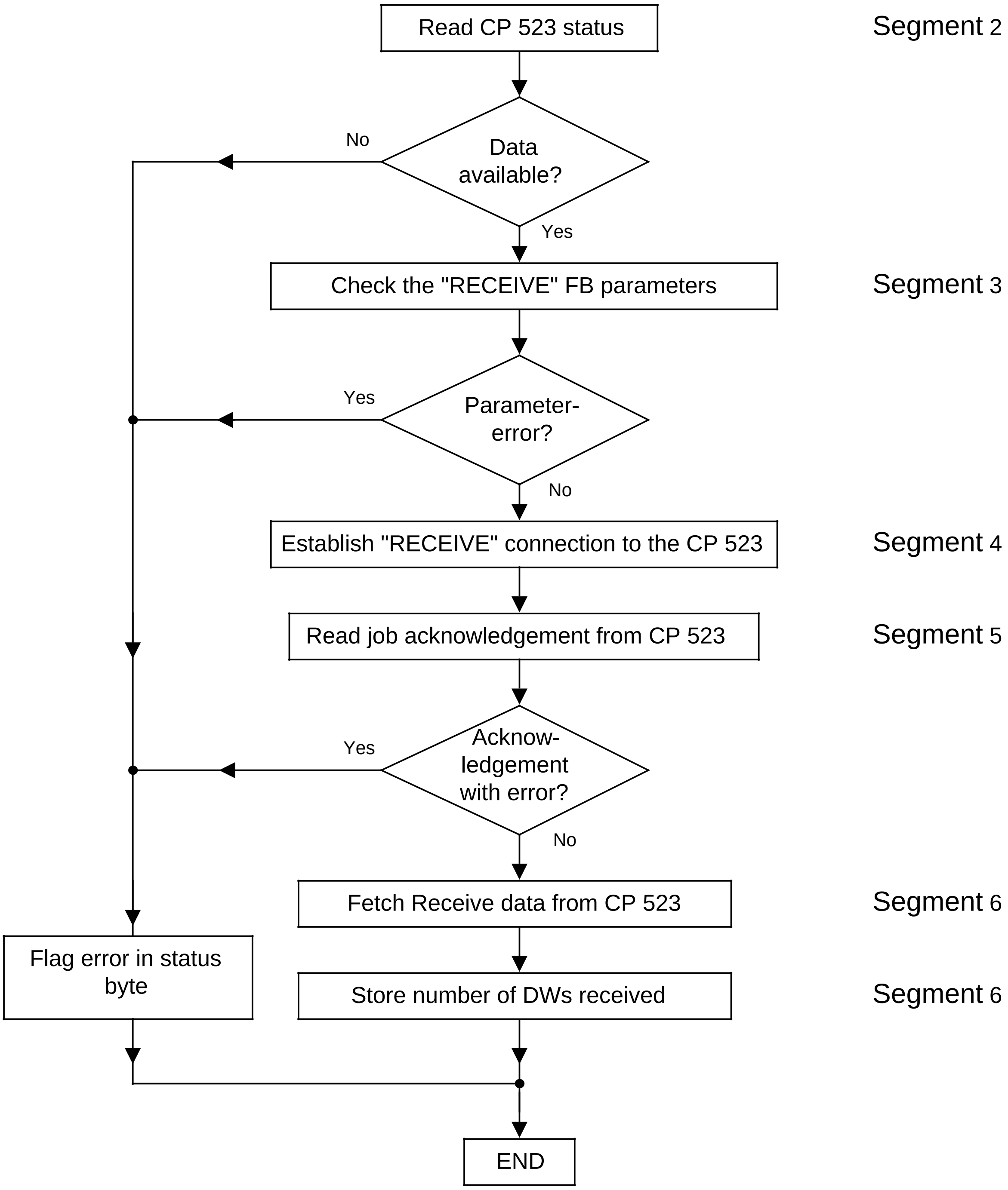
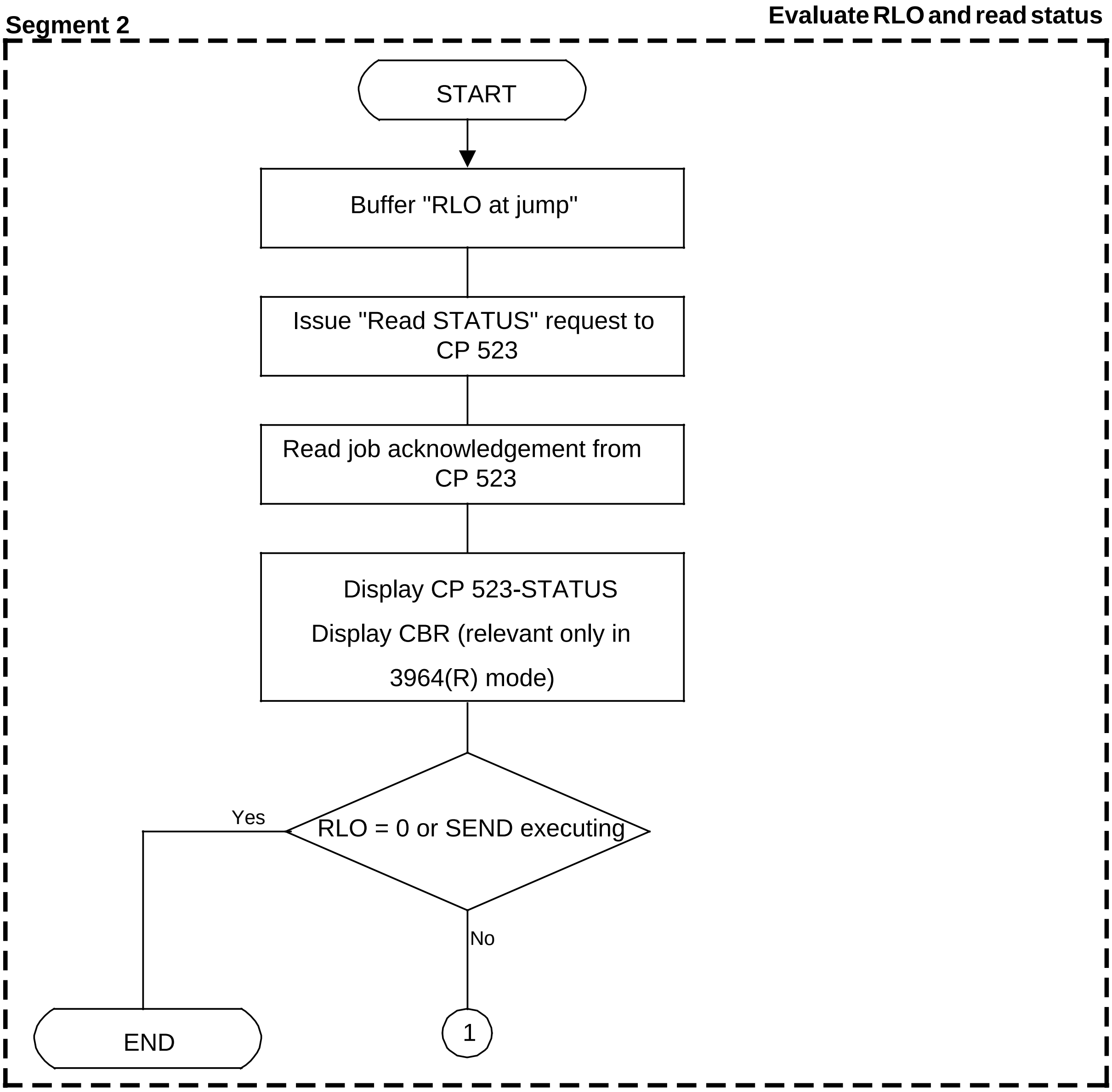
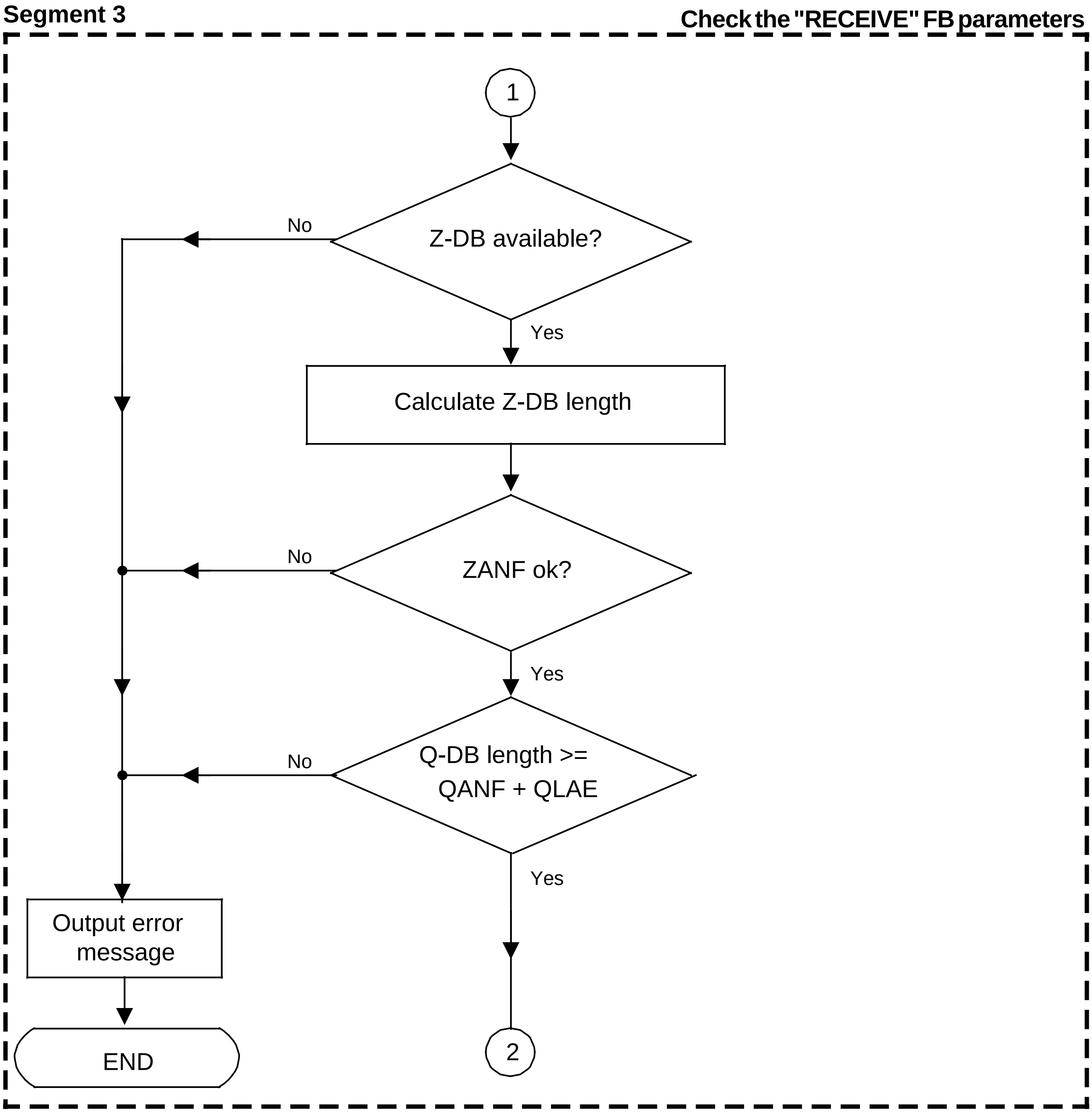
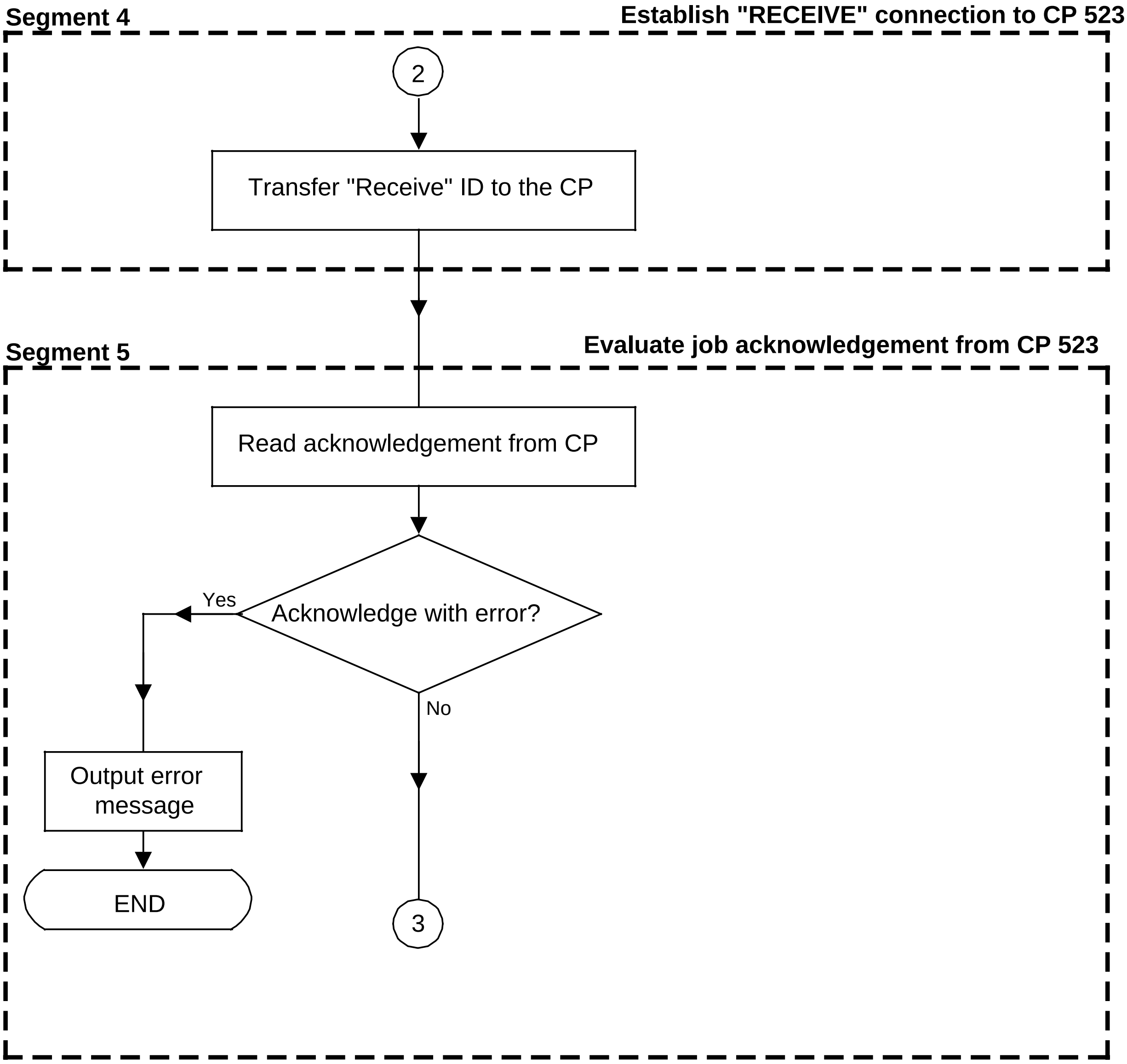

Read CP 523 status
Segment 2
Data available?
No
Yes
Check the "RECEIVE" FB parameters
Segment 3
Parameter-error?
Yes
No
Establish "RECEIVE" connection to the CP 523
Segment 4
Read job acknowledgement from CP 523
Segment 5
Acknow-ledgement with error?
Yes
No
Fetch Receive data from CP 523
Segment 6
Flag error in status byte
Store number of DWs received
Segment 6
END

Figure 8-4. Schematic for "Receive Frame" (FB 201)

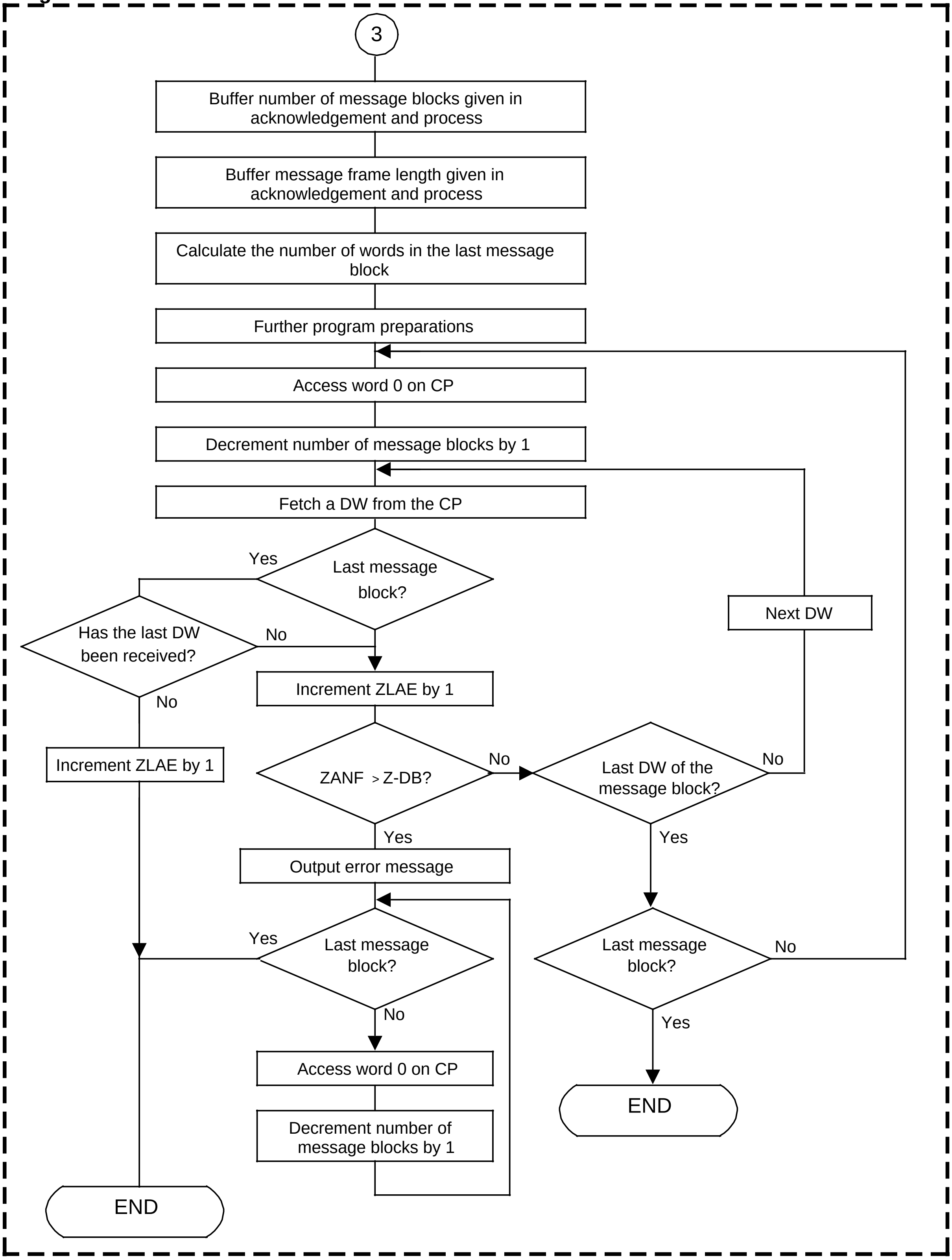
8.2.4 Flowchart of "RECEIVE" FB 201


Segment 2
Evaluate RLO and read status
START
Buffer "RLO at jump"
Issue "Read STATUS" request to CP 523
Read job acknowledgement from CP 523
Display CP 523-STATUS
Display CBR (relevant only in 3964(R) mode)
RLO = 0 or SEND executing
Yes
No
END
1


Segment 3
Check the "RECEIVE" FB parameters
1
Z-DB available?
No
Yes
Calculate Z-DB length
ZANF ok?
No
Yes
Q-DB length >=
QANF + QLAE
No
Yes
Output error
message
END
2


Segment 4
Establish "RECEIVE" connection to CP 523
2
Transfer "Receive" ID to the CP
Segment 5
Evaluate job acknowledgement from CP 523
Read acknowledgement from CP
Acknowledge with error?
Yes
No
Output error message
END
3

Segment 6 **Fetch Receive data from CP 523**


3
Buffer number of message blocks given in acknowledgement and process
Buffer message frame length given in acknowledgement and process
Calculate the number of words in the last message block
Further program preparations
Access word 0 on CP
Decrement number of message blocks by 1
Fetch a DW from the CP
Yes
Last message block?
Has the last DW been received?
No
Next DW
No
Increment ZLAE by 1
Increment ZLAE by 1
ZANF > Z-DB?
No
Last DW of the message block?
No
Yes
Yes
Output error message
Yes
Last message block?
Last message block?
No
No
Yes
Access word 0 on CP
END
Decrement number of message blocks by 1
END

8.2.5 Flags in "RECEIVE" FB 201

Explanation of the flags used by "RECEIVE" FB 201:

Table 8-4. Overview of the Flags Used by "RECEIVE" FB 201

Flag	Symbol	Meaning
F 255.0	H-FLAG-1	Auxiliary flag for buffering the RLO at jump
F 254.7	H-FLAG-3	Auxiliary flag for STAT bit "Data available"
FY 254	H-BYTE-1	Auxiliary flag byte for buffering STAT
FY 246	H-BYTE-2	Auxiliary flag byte for processing the last message block
FW 230	ZLAE-REG	"Number of data words fetched from the CP (= ZLAE parameter") register
FW 232	Z-DB-LAE	Auxiliary register for storing length of destination DB
FW 234	BADR-IND	Module address index register for word-serial reading from CP
FW 236	ZANF-REG	Register for Z-DB start address for DWs (DO FW register, store data words in Z DB)
FW 238	B-AN-E	”Number of message blocks" register
FW 240	ANZ-E-W	Number of words to be received
FW 242	BADR-REG	Module address register (DO FW register for CP access)
FW 244	W-LE-BL	Number of words in last message block

FB 201	LEN=374

```
Segment 1     0000
NAME :EMPFANG
DES  :BADR       I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KF
DES  :Z-DB       I/Q/D/B/T/C: B
DES  :ZANF       I/Q/D/B/T/C: D  KM/KH/KY/KC/KF/KT/KZ/KG: KF
DES  :ZLAE       I/Q/D/B/T/C: Q  BI/BY/W/D: BY
DES  :STAT       I/Q/D/B/T/C: Q  BI/BY/W/D: BY
DES  :KBE        I/Q/D/B/T/C: Q  BI/BY/W/D: BY

0017      :***
```

STL FB 201	Explanations
`Segment 2     0018`	
`0018      :`	------------------------------
`0019      :`	For CPUs
`001A      :`	941
`001B      :`	942
`001C      :`	943
`001D      :`	944
`001E      :`	------------------------------
`001F      :`	
`0020      :=   F 255.0`	Save RLO
`0021      :`	
`0022      :LW  =BADR`	Load and store module
`0023      :T   FW 242`	address
`0024      :`	
`0025      :L   KH 00A0`	Write ”Read status” KOOR ID
`0027      :DO  FW 242`	to bytes 0 and 1
`0028      :T   PY   0`	of the CP 523
`0029      :L   FW 242`	
`002A      :I        1`	
`002B      :T   FW 242`	
`002C      :L   KH 0000`	
`002E      :DO  FW 242`	
`002F      :T   PY   0`	
`0030      :DO  FW 242`	Fetch acknowledgement to
`0031      :L   PY   0`	”Read status”
`0032      :T   FY 254`	from CP and buffer
`0033      :`	
`0034      :L   FW 242`	Set pointer back to
`0035      :D        1`	module base address
`0036      :T   FW 242`	
`0037      :`	Does CP report 'Data available'?
`0038      :AN  F 254.7`	”N O”

STL FB 201	Explanations
0039 :JC =NEIN	
003A :	Load STAT bit ”Data available” and
003B :L KH 0001	jump to 'output STAT'
003D :JU =STAT	
003E NEIN :	STAT bit ”No data”
003F :L KH 0000	
0041 STAT :	Output STAT bit
0042 :T =STAT	
0043 :	
0044 :L =KBE	Reset KBE.7 (Request
0045 :L KH 007F	accepted) but do not
0047 :AW	reset error flags
0048 :T =KBE	
0049 :	RLO ”0” at jump?
004A :AN F 255.0	or no data available?
004B :ON F 254.7	Program ”End”
004C :BEC	
004D :	
004E :***	
NETWORK 3 004F	
004F :	Check whether Z-DB is available
0050 :LW =Z-DB	Load Z-DB parameter
0051 :SLW 8	Isolate the DB number from
0052 :SRW 7	the Z-DB parameter specification
0053 :	
0054 :L KH E400	Calculate the start address
0056 :+F	of the Z DB using the
0057 :LIR 0	block address list
0058 :	
0059 :L KB 0	Z-DB available?
005A :!=F	
005B :JC =FE04	”E R R O R”
005C :	
005D :TAK	Calculate length of the Z-DB
005E :ADD KF -2	
0060 :LIR 0	Gross length in words
0061 :ADD KF -5	Subtract length of block header
0063 :T FW 232	and store
0064 :	
0065 :LW =ZANF	Check ZANF information
0066 :L KB 0	ZANF less than 0?
0067 :<F	
0068 :JC =FE04	”E R R O R”

STL FB 201	Explanations
0069 :	
006A :TAK	ZANF greater than 255?
006B :L KB 255	
006C :>F	
006D :JC =FE04	”E R R O R”
006E :	
006F :TAK	ZANF >Z-DB length?
0070 :L FW 232	
0071 :<=F	
0072 :JC =PAOK	”N O”
0073 FE04 :	
0074 :L KH 0048	Output ”FB parameter
0076 :T =STAT	error”
0077 :	
0078 :BEU	PROGRAM ”END”
0079 PAOK :	
007A :***	
NETWORK 4 007B	
007B :L KH 00A0	Write KOOR ID ”RECEIVE”
007D :DO FW 242	to CP 523
007E :T PY 0	word 0
007F :L FW 242	
0080 :I 1	
0081 :T FW 242	
0082 :L KH 0080	
0084 :DO FW 242	
0085 :T PY 0	
0086 :	
0087 :L FW 242	Set pointer back to
0088 :D 1	module base address
0089 :T FW 242	
008A :***	
NETWORK 5 008B	
008B :DO FW 242	Read byte 0,
008C :L PY 0	isolate error bit
008D :SLW 4	and buffer
008E :T FY 254	
008F :L FW 242	Increment pointer to byte 1
0090 :I 1	
0091 :T FW 242	

STL FB 201	Explanations
0092 :DO FW 242	Read byte 1 (acknowledgement or
0093 :L PY 0	CBR when computer link)
0094 :T =KBE	and output
0095 :T FY 246	or buffer
0096 :	
0097 :L FW 242	Set pointer back to
0098 :D 1	module base address
0099 :T FW 242	
009A :	
009B :L KB 0	Acknowledgement with error?
009C :L FY 254	
009D :!=F	
009E :JC =NEIN	”NO ERROR”
009F :	
00A0 :S F 254.3	Set error bit for STAT
00A1 :L FY 254	
00A2 :T =STAT	Output STATbit
00A3 :	
00A4 :L KH 0090	Does the CP report
00A6 :L FY 254	delay time exceeded?
00A7 :AW	
00A8 :!=F	
00A9 :JC =NEIN	”YES” -> message is output
00AA :	in the status byte
00AB :	but the function block
00AC :	is not interrupted
00AD :	
00AE :BEU	Program ”END”
00AF NEIN :	
00B0 :L FY 246	Error flagged in CBR?
00B1 :L KH 007F	(error flag is isolated)
00B3 :AW	
00B4 :L KB 0	
00B5 :!=F	
00B6 :JC =KBOK	Coordination byte OK
00B7 :L KH 0088	Output error ”CBR error/computer link
00B9 :T =STAT	
00BA :BEU	Program ”END”
00BB KBOK :	
00BC :***	
NETWORK 6 00BD	
00BD :L KB 0	Reset auxiliary register 2
00BE :T FB 246	
00BF :	

STL FB 201	Explanations
00C0 :	
00C1 :L FW 242	Set pointer to base
00C2 :I 6	address +6 (byte 6)
00C3 :T FW 242	
00C4 :	
00C5 :DO FW 242	Read no. of message blocks in
00C6 :L PY 0	Receive message frame from the CP,
00C7 :D 1	decrement by 1,
00C8 :SLW 2	convert to number of words
00C9 :	(one message block comprises 4 words)
00CA :T FW 238	and store
00CB :	
00CC :L FW 242	Set pointer to base
00CD :ADD KF -4	address +2 (byte 2)
00CF :T FW 242	
00D0 :	
00D1 :DO FW 242	Read length of Receive
00D2 :L PW 0	message frame in the bytes from the CP,
00D3 :SRW 1	convert to number of words
00D4 :T FW 240	and store
00D5 :	
00D6 :L FW 238	Subtract the number of message blocks
00D7 :-F	in words from this
00D8 :T FW 244	and store
00D9 :	The result of the subtraction
00DA :	is the number of words
00DB :	in the last message block
00DC :	
00DD :L FW 238	Correct the number
00DE :SRW 2	of message blocks
00DF :I 1	
00E0 :T FW 238	
00E1 :	
00E2 :	
00E3 :L KB 0	Reset the ZLAE register
00E4 :T FW 230	
00E5 :	
00E6 :LW =ZANF	Load ZANF
00E7 :T FW 236	and store
00E8 :	
00E9 :L FW 242	Set module address register
00EA :D 2	to CP 523 base address
00EB :T FW 242	and load into
00EC :T FW 234	module address
00ED :	index register

STL FB 201	Explanations
00EE :DO =Z-DB	Open destination DB
00EF :	
00F0 NBLO :JU OB 31*	(Retrigger scan time)
00F1 :L KH 0000	Access word 0 on CP 523
00F3 :DO FW 242	for transferring a message block
00F4 :T PY 0	
00F5 :L FW 242	
00F6 :I 1	
00F7 :T FW 242	
00F8 :L KH 0000	
00FA :DO FW 242	
00FB :T PY 0	
00FC :	
00FD :L FW 242	Set pointer back to
00FE :D 1	base address
00FF :T FW 242	
0100 :	
0101 :L FW 238	Decrement no. of message blocks
0102 :D 1	for Receive message frames
0103 :T FW 238	and store
0104 NWOR :	
0105 :DO FW 234	Fetch a data word from CP 523
0106 :L PW 0	
0107 :	
0108 :DO FW 236	and store
0109 :T DW 0	in Z-DB
010A :	
010B :L FW 238	Last message block?
010C :L KB 0	
010D :><F	
010E :JC =KEND	”N O”
010F :	
0110 :L FY 246	Increment auxiliary register by 1
0111 :I 1	
0112 :T FY 246	and store
0113 :	(shows the number of the word
0114 :	received in the last message block)
0115 :L FW 244	and compare with the number of words
0116 :<F	in the last message block to see
0117 :	if the last word was received
0118 :JC =KEND	”NO”
0119 :	
011A :L FW 230	Increment ZLAE register by 1
011B :ADD KF +1	
011D :T FW 230	and store

* In 115 CPUs only (if required)

STL FB 201	Explanations
011E :	
011F :JU =END	
0120 :	
0121 KEND :	
0122 :L FW 230	Increment ZLAE register by 1
0123 :ADD KF +1	
0125 :T FW 230	and store
0126 :	
0127 :L FW 236	Increment ZANF register by 1
0128 :ADD KF +1	
012A :T FW 236	and store
012B :	
012C :L FW 232	ZANF register >DB length?
012D :D 1	
012E :<=F	
012F :JC =JUM1	”N O”
0130 :	
0131 :L KH 0058	Buffer ”Z-DB too small for total
0133 :T FY 254	Receive data” error message
0134 :	
0135 :	
0136 ENTL :	
0137 :L FW 238	Last message block?
0138 :L KB 0	
0139 :!=F	
013A :JC =END	”Y E S”-->END
013B :	
013C :L KH 0000	Access word 0 on CP 523
013E :DO FW 242	for transferring a message block
013F :T PY 0	
0140 :L FW 242	
0141 :I 1	
0142 :T FW 242	
0143 :L KH 0000	
0145 :DO FW 242	
0146 :T PY 0	
0147 :L FW 242	Set pointer to base address
0148 :D 1	
0149 :T FW 242	
014A :	
014B :L FW 238	Decrement message block
014C :D 1	counter register by 1 and store
014D :T FW 238	
014E :	
014F :JU =ENTL	Empty Receive mailbox loop

STL FB 201	Explanations
0150 JUM1 :	
0151 :L FW 242	Last word of the message block read?
0152 :ADD KF +6	
0154 :L FW 234	
0155 :!=F	
0156 :JC =LWOR	”YES”
0157 :	
0158 :ADD KF +2	Increment BADR index register by 2 and store
015A :T FW 234	
015B :	
015C :JU =NWOR	”NEXT WORD”
015D :	
015E LWOR :L FW 242	Set module index register to module base address
015F :T FW 234	
0160 :	
0161 :L FW 238	Last message block?
0162 :L KB 0	
0163 :><F	
0164 :JC =NBLO	”NO” -> next message block
0165 :	
0166 END :	
0167 :	
0168 :L FW 230	Output count register
0169 :T =ZLAE	
016A :	
016B :L FY 254	Output status
016C :T =STAT	
016D :	
016E :***	
Segment 7 0170	
016F :BE	

8.2.6 Execution Times for FB 200 and 201

Table 8-5. Execution Times for FB 200 and 201 (in ms)

CPU	FB	RLO = 0 (no op.)	12 Bytes Fixed Mess. Frame Length	12 Bytes with End--of-Text Character	12 Bytes with 3964(R) Protocol	256 Bytes Fixed Mess. Frame Length	256 Bytes with End-of-Text Character	256 Bytes with 3964(R) Protocol
941	200	7.5	50	55	52	520	550	520
	201	7.5	50	50	48	550	560	550
942	200	5	20	20	24	240	210	220
	201	5	20	20	21	250	225	225
943	200	2.5	14	15	18	150	140	160
	201	2.5	14	16	15	165	165	165
944	200	0.5	3.5	3.5	3.5	27	30	26
	201	0.5	6.5	6.5	6.5	90	90	88

A Summary

Tables

A Summary

A.1 Parameter Setting in Print Mode

Table A-1. Parameter Setting Data for Print Mode (6.3)

Param. Block	Description	Value Range	Default Value on the CP 523
0	Baud rate 110 baud 200 baud 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud	1 2 3 4 5 6 7 8	4
	Parity even odd ”mark” ”space” no parity check	0 1 2 3 4	0
	BUSY signal No Yes	0 1	0
	Interface TTY RS-232-C (V.24)	0 1	0
	Data format Parity: 11-bit character frame 7 data bits (yes) 8 data bits (yes) 8 data bits (no) 10-bit character frame 7 data bits (no) 7 data bits (yes) 8 data bits (no)	 0 1 2 3 4 5	 0
	HW handshaking OFF ON	0 1	0
1	Waiting time after CR LF FF	(00$_H$ to FF$_H$) · 25ms (00$_H$ to FF$_H$) · 25ms (00$_H$ to FF$_H$) · 25ms	0A$_H$ 0.250 s 0A$_H$ 0.250 s A0$_H$ 4 s
2	XON character XOFF character	ASCII character (01$_H$ to 7F$_H$) ASCII character (01$_H$ to 7F$_H$)	FF$_H$ (no XON/XOFF protocol) FF$_H$ (no XON/XOFF protocol)
3	End-of-text character Function character	ASCII character (01$_H$ to FF$_H$) ASCII character (01$_H$ to FF$_H$)	$ (24$_H$) ” (22$_H$)

Table A-1. Parameter Setting Data for Print Mode (Continued)

4	Date and time of day display Order for date J,Y = Year M = Month T,D = Day	Any combination of year, month and day is permissible	TMY
	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	HMS

	Separator for time of day	ASCII character (20_H to $7F_H$)	”:” ($3A_H$)
	24h clock (German) 12h clock (English)	d, D e, E	D (24h clock)
5	Page format		
	Lines/page	14_H to FF_H	48_H (72_D)
	Left margin	00_H to $3C_H$	00_H
	Page number		
	Top	o, O, h, H	
	Bottom	u, U, f, F	u
	None	Other character	
6	Header and footer		No headers or footers
	Header 1	K1”Text”, H1”Text”	
	Header 2	K2”Text”, H2”Text”	
	Footer 1	F1”Text”	
	Footer 2	F2”Text”	
8	Character conversion table	16 characters can be converted with a code sequence consisting of up to 7 characters.	None
9	Correction value for integral clock	-400_D to $+400_D$ s/month	0000_D

A.2 Job Requests in Print Mode

Table A-2. Permissible Job Requests to the CP 523 in Print Mode (CPU to CP) (6.7)

Byte 0		Byte 1		Job Request
7 6 5 4	3 2 1 0	7 6 5 4	3 2 1 0	
0	0	0	0	Read status byte, status of the printer and current clock data
$\boxed{0}$	Message text number			Print message text with CR/LF at end
1	0	Day of the week setting		Set clock (settings in bytes 1 to 7)
$\boxed{2}$	0	Page number		Print message text with CR/LF at end
$\boxed{3}$	Message text number			Print message text without CR/LF at end
$\boxed{4}$	Message text number			Print message text without CR/LF at end
$\boxed{5}$	0	0	0	Form feed
$\boxed{6}$	0	0	0	Line feed
7	0	0	0	Clear message buffer
8	0	0	0	Print all messages
 9 9 9	 0 0 0	 0 2 7	 0 0 0	Transfer parameter setting data - Set serial interface parameters - Set XON/XOFF character parameters - Select Print mode

□ = Print requests. These are written into the message buffer if necessary.

"Print message text" (6.7.1)

Table A-3. Transferring Additional Information for "Print Message Text" (CPU to CP)

Word (Byte)	Meaning	Assignment
2 (2+3)	Value of the variable or number of the message text to be inserted	0000_H to $FFFF_H$ (dependent on the configured data format) 0000_H to $0FFF_H$
4 (4+5)	Value of the variable or number of the message text to be inserted	0000_H to $FFFF_H$ (dependent on the configured data format) 0000_H to $0FFF_H$
6 (6+7)	Value of the variable or number of the message text to be inserted	0000_H to $FFFF_H$ (dependent on the configured data format) 0000_H to $0FFF_H$

"Set time of day and date" (6.7.2)

Table A-4. Transferring Additional Information for "Set Time of Day and Date" (CPU to CP)

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

* Bit 7 = 1 in the case of the 12h clock p.m.

A.3 Feedback Information in Print Mode

Table A-5. Status of the Module in Print Mode (Byte 0) (6.6)

Byte 0		Status	Detailed Explanation
Bits 4 to 7	0 to 3		
0	0	No error	Job buffer empty
X	1	Defective memory submodule	
X	2	No texts configured	No message texts configured on the memory submodule
X	3	Entries in job buffer	
X	7	No battery backup	Either - No battery has been inserted in the power supply module or - The battery is defective
X	8	Job buffer/message buffer full	The module cannot service any further print requests at present. The request must be repeated.
0	F	CP in restart routine	Message appears only on restart. The clock data is invalid. The module cannot accept any job requests.
1	X	Clock defective	Replace module
2	X	Default clock time set	The clock is preset with the values Sunday, 1:1:90, 12:00:00.
3	X	Time of day/date error	At least one setting is outside the permissible range. The clock has not accepted the new clock data and is continuing with the current data.
4	X	Illegal job request	You have submitted a request to the CP which is not permissible in Print mode (6.7).
8	X	Hardware fault	Replace module

X= Signal state not significant for other half-byte

Table A-6. Status of the Printer, Date and Time of Day

Byte	Value Range	Meaning
1	01_{BCD} to 07_{BCD} 11_{BCD} to 17_{BCD}	Bits 0 to 3 : 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday Bit 4=0 : Printer ready Bit 4=1 : Printer not ready
2	01_{BCD} to 31_{BCD}	Day
3	01_{BCD} to 12_{BCD}	Month
4	00_{BCD} to 99_{BCD}	Year
5	00_{BCD} to 23_{BCD} 01_{BCD} to 12_{BCD} 81_{BCD} to 92_{BCD}	Hour in 24h clock in 12h clock a. m. (bit 7=0) in 12h clock p. m. (bit 7=1)
6	00_{BCD} to 59_{BCD}	Minute
7	00_{BCD} to 59_{BCD}	Second

A.4 Setting Parameters in Communications Mode

- Setting parameters via the memory submodule: Enter the data in DB 1 (7.3.1)
- Setting parameters via the user program: "Transfer parameter setting data for parameter block X" (7.3.2)

Table A-7. Assignments of the Parameter Blocks in Communications Mode (Parameter Blocks 0, 2 and 9)

Parameter block	Byte	Description		Value Range	Default Value on the CP 523
0	2	Baud rate	110 baud	1	4
			200 baud	2	
			300 baud	3	
			600 baud	4	
			1200 baud	5	
			2400 baud	6	
			4800 baud	7	
			9600 baud	8	
	3	Parity	even	0	0
			odd	1	
			”mark”	2	
			”space”	3	
			no parity check	4	
	4	BUSY signal	No	0	0
	5	Interface	TTY	0	0
			RS-232-C (V.24)	1	
	6	Data format:	Parity:		
		11-bit character frame 7 data bits	(yes)	0	0
		8 data bits	(yes)	1	
		8 data bits	(no)	2	
		10-bit character frame 7 data bits	(no)	3	
		7 data bits	(yes)	4	
		8 data bits	(no)	5	
	7	HW handshake	OFF	0	0
			ON	1	
2	2	XON character*		01_H to $7F_H$	$FFFF_H$ (no XON/XOFF protocol)
	3	XOFF character*		01_H to $7F_H$	
7		see Table A-7a.			
9	2+3	Correction value (s/month)		-400_D to $+400_D$	0000_D

* Only possible in interpretive ASCII mode

Table A-7a. Assignments of the Parameter Blocks in Communications Mode (Parameter Block 7)

	71$_H$			72$_H$			73$_H$			7A$_H$			
	Transparent Mode			Interpretive Mode 1			3964(R) Mode			3964(R) Mode Follow-Up Request			
Byte	Description	Permissible Values	Default Values on the CP 523	Description	Permissible Values	Default Values on the CP 523	Description	Permissible Values	Default Values on the CP 523		Permissible Values	Default Values on the CP 523	Byte
0	Iob number	90$_H$			90$_H$		Job number	90$_H$		Job number	90$_H$		
1	Bits 4 to 7: Parameter block number Bits 0 to 3: Mode	71$_H$		Bits 4 to 7: Parameter block number Bits 0 to 3: Mode	72$_H$		Bits 4 to 7: Parameter block number Bits 0 to 3: Mode	73$_H$		Bits 4 to 7: Parameter block number Bits 0 to 3: Mode	7A$_H$		
2	Character delay time (1 to 3000) • 10ms		0001$_H$ = 0 ms	Character delay time (1 to 3000) • 10ms		0001$_H$ = 10 ms	Character delay time (ZVZ) (1 to 65535) • 10ms	to FFFF$_H$	16$_H$ (220 ms)	Without block check character With block check character	00$_H$ 01$_H$	00$_H$	2
3										Priority Low High	00$_H$ 01$_H$	01$_H$	3
4	Message frame size (1 to 256 bytes)	0001$_H$ to 0100$_H$		Message frame size (0 byte)			Time-out (QVZ) (1 to 65535) •10ms	0001$_H$ to FFFF$_H$	00C8$_H$ (2000 ms = 2s)	Connect. build-up attempts (1 to 255)	00$_H$ to FF$_H$	06$_H$	4
5										Number of retries (1 to 255)	00$_H$ to FF$_H$	06$_H$	5
6	Irrelevant			End-of-text character 1 end-of-text character in byte 7 (byte 6 = 0)	0001$_H$ to 00FF$_H$	0D$_H$	Block delay time (BWZ) (1 to 65535) •10ms	0001$_H$ to FFFF$_H$	0190$_H$ (4000 ms = 4s)	Irrelevant			6
7				or 2 end-of-text characters in bytes 6 and 7	0101$_H$ to FFFF$_H$								7

A.5 Job Requests in Communications Mode

Table A-8. Permissible Job Requests to the CP 523 in Communications Mode

Job Request	Byte 0		Byte 1	
	7 6 5 4	3 2 1 0	7 6 5 4	3 2 1 0
Read status byte, status of the peripheral device and current clock data	0	0	0	0
Set clock (settings in bytes 1 to 7)	1	0	Day of the week setting	
Transfer parameter setting data				
Initialize serial interface	9	0	0	0
Set the XON/XOFF parameters	9	0	2	0
Select Print mode	9	0	7	0
Select Transparent mode	9	0	7	1
Select Interpretive mode	9	0	7	2
Select 3964(R) Mode	9	0	7	3
Select 3964(R) Mode Follow-Up Request	9	0	7	A
Correction value for integral clock positive	9	0	9	0
Correction value for integral clock negative	9	0	9	1
Coordinate data transfer				
Read status byte and coordination info	A	0	0	0
Send message frame	A	0	0	1
Receive message frame	A	0	8	0

"Send fixed-length message frame" or send with 3964(R) (7.6.1)

Table A-9. Transfer Memory Assignments for ”Send Fixed-Length Message Frame" (CPU to CP)

Byte	Meaning	Value Range
0	Job number for "Coordinate data transfer"	$A0_H$
1	"Send message frame" identifier	01_H
2+3	Send length in bytes	0001_H to 0100_H
4-7	Not significant	00_H to FF_H

”Send frame with end-of-text character” (7.6.2)

Table A-10. Transfer Memory Assignments for "Send Message Frame with End-of-Text Characters" (CPU to CP)

Byte	Meaning	Value Range
0	Job number for ”Coordinate data transfer"	$A0_H$
1	"Send message frame" identifier	01_H
2+3	Send length in bytes	0000_H
4	2nd end-of-text character no 2nd end-of-text character	01_H to FF_H 00_H
5	1st end-of-text character	01_H to FF_H
6+7	Not significant	00_H to FF_H

"Receive message frame" (7.7)

Table A-11. Transfer Memory Assignments for "Receive Message Frame" (CPU to CP)

Byte	Meaning	Value Range
0	Job number for ”Coordinate data transfer"	$A0_H$
1	"Receive message frame" identifier	80_H
2-7	Not significant	00_H to FF_H

A.6 Feedback Information in Communications Modes 1 and 2

Table A-12. Coordination Information for "Coordinate Data Transfer" (CP to CPU) (7.4)

Byte	Meaning	Possible Values
0	Status byte	$5X_H$
1	Permission to send and receive CPU can neither send nor receive CPU can send CPU can receive CPU can send and receive	 00_H 01_H 80_H 81_H
2+3	Message frame length in bytes*	0000_H to 0100_H
4+5	Not significant	0000_H
6	Number of message blocks of the next message frame in the Receive mailbox	00_H to 20_H
7	Number of message frames in the Receive mailbox*	00_H to 64_H

* Only after "Receive message frames" request $A080_H$

Table A-13. Status Information and Current Clock Data (CP to CPU)

Byte	Meaning	Possible Values
0	Status byte	7.4
1	Status of the peripheral device Bits 4 to 7: Peripheral device ready Peripheral device not ready Bits 0 to 3: Current day of the week 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday 5=Thursday, 6=Friday, 7=Saturday	 $0X_{BCD}$ $1X_{BCD}$ $X1_{BCD}$ 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 24 h 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}

Table A-14. Status Byte in Communications Mode (CP to CPU) (7.4)

Byte 0 Bits 4 to 7	 Bits 0 to 3	Status
0	0	No error
X	1	No memory submodule
X	3	Entries in job buffer (for parameter setting requests only)
X	7	No battery backup
X	8	Job buffer full (for parameter setting requests only)
0	F	CP restart in progress
1	X	Clock defective
2	X	Default time of day set
3	X	Time of day/date error
4	X	Illegal job request
8	X	Hardware fault
Coordination information (only after "Coordinate data transfer" request)		
5	9*	Character delay time exceeded
5	A*	Parity error
5	B*	Receive after XOFF or receive after DTR = 0
5	C*	Message frame exceeds 256 bytes
5	D	Permanent break in peripheral device cable
5	E*	Receive mailbox overflow

X= signal state not significant for other half-byte

* These errors are not flagged until a "coordinate data transfer" request to receive data has been serviced for this message frame.

A.7 Feedback Information in Communications Mode 3

Table A-15. Status Info from the CP 523 for Job Number ”A000$_H$”

Byte	Description	Possible Values
0	Status byte	50$_H$
1	Permission to send and receive CPU can neither send nor receive CPU can send CPU can receive CPU can send and receive	 00$_H$ 01$_H$ 80$_H$ 81$_H$
2	Coordination byte 'send' (CBS)	Table A-18.
3 - 7	Not significant	

Table A-16. Status Info for Send Request ”A001$_H$”

Byte	Description	Possible Values
0	Status byte	50$_H$
1	Send request accepted Send request rejected	01$_H$ 00$_H$
2	Coordination byte 'send' (CBS)	Table A-18.
3 - 5	Not significant	
6	Number of message blocks in message frame	00$_H$ to 20$_H$
7	Number of message frames in Receive mailbox	00$_H$ to 64$_H$

Table A-17. Status Info for Receive Request ”A080$_H$”

Byte	Description	Possible Values
0	Status byte	50$_H$
1	Coordination byte 'receive' (CBR)	Table A-19.
2+3	Message frame length in bytes	0000$_H$ to 0100$_H$
4+5	Not significant	
6	Number of message blocks in the Receive message frame	00$_H$ to 20$_H$
7	Number of message frames in the Receive mailbox	00$_H$ to 64$_H$

Table A-18. Return Info and Error Flags in the Coordination Byte 'Send' (CBS) in 3964(R) Mode

Bit								Description	Response
7	6	5	4	3	2	1	0		
								Return info for job request $A000_H$	
0 1	XX_H*							Send buffer free Send buffer not free	
								Return info for job request $A001_H$	
0	XX_H*							The info returned in the CBS following issue of Send request $A000_H$ depends on the value in byte 1. If byte 1 is 00_H, 0 in bit 7 of the CBS means that either a parameter error was detected or that the Send buffer is free but the current Send request cannot be serviced because a Send request was issued following transfer of a message frame which was not preceded by a 'Check module status' request.	
1	XX_H*							If bit 7 in the CBS is ”1”, the Send buffer is reserved because the preceding Send request is still being serviced. If byte 1 is 01_H, bit 7 = 1 means that the request was accepted; the Send buffer is now reserved.	
								Errors flagged for $A000_H$ and $A001_H$	
	09_H							The receiver acknowledged with NAK during connection cleardown	Receive data invalid
	$0B_H$							the receiver acknowledged with NAK during connection buildup	Receive data invalid
	$0D_H$							Parameter error	No data is transmitted
	$0F_H$							Transmission aborted by receiver	Receive data invalid
	15_H							QVZ (time-out) during connection buildup	No data is transmitted
	17_H							QVZ (time-out) 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
	00_H							No errors	

* see error message

Table A-19. Return Info and Error Flags in the Coordination Byte 'Receive' (CBR) in 3964(R) Mode

Bit		Description	Response
7	6 5 4 3 2 1 0		
		Return info for job request A080$_H$	
1 0	XX$_H$*	Request accepted No frame to fetch	
		Error flagged for A080$_H$	
	03$_H$	Parity error	Data is rejected
	05$_H$	Message frame length 0	
	07$_H$	Receive buffer full	Data is rejected
	09$_H$	Too many message frames received	Buffer full, subsequent message frames will be rejected
	0B$_H$	Message frame too long (256 bytes)	Data is rejected
	0D$_H$	DLE not doubled or no ETX after DLE	Data is rejected
	11$_H$	STX error, handshaking not started with STX	Data is rejected
	13$_H$	Character delay time exceeded	Data is rejected
	15$_H$	Message frame delay time exceeded	Data is rejected
	17$_H$	Checksum error	Data is rejected
	1B$_H$	Break	Data is rejected
	00$_H$	No errors	

* see error message

A.8 Combinations of the Most Important Parameters

	Printer Mode	Interpretive Mode	Transparent Mode	3964(R) Mode	V.24 (RS232C)	TTY	XON/ XOFF	HW Handshake	BUSY	Baud Rate	Parity	Data Format	Character Delay Time
Printer Mode		0	0	0	3	3	3	2	3	3	3	3	0
Interpretive Mode	0		0	0	3	3	3	3	2	3	3	3	3
Transparent Mode	0	0		0	3	3	2	3	2	3	3	3	3
3964(R) Mode	0	0	0		3	3	2	2	2	3	3	3	3
V.24 (RS232C)	3	3	3	3		0	3	3	5	1	1	1	1
TTY	3	3	3	3	0		3	2	5	1	1	1	1
XON/ XOFF	3	3	2	2	3	3		6	7	1	1	1	1
HW Handshake	2	3	3	2	3	2	6		9	1	1	1	1
BUSY	3	2	2	2	5	5	7	9		1	1	1	1
Baud Rate	3	3	3	3	1	1	1	1	1		1	1	4
Parity	3	3	3	3	1	1	1	1	1	1		8	1
Data Format	3	3	3	3	1	1	1	1	1	1	8		1
Character Delay Time	0	3	3	3	1	1	1	1	1	4	1	1	

0: Cannot be initialized
1: Parameters do not affect one another
2: Is not supported
3: Parameters not interdependent
4: ZVZ and baud rate must be in reasonable relation to one another. Recommendation:

$$\mathrm{ZVZ} \quad \frac{\text{2x (no. of bits in character frame) x1000}}{\text{baud rate}} \qquad \text{(in ms)}$$

5: In Print mode only
6: HW handshake has priority
7: XON/XOFF has priority
8: Data format has priority
9: BUSY signal and HW handshake are mode-dependent

B SIEMENS Addresses Worldwide

European Companies and Representatives

Austria
Siemens AG Österreich
Vienna
Bregenz
Graz
Innsbruck
Klagenfurt
Linz
Salzburg

Belgium
Siemens S.A.
Brussels
Liège
Siemens N.V.
Brussels
Antwerp
Gent

Bulgaria
RUEN office of the
INTERPRED corporation,
agency of the
Siemens AG Sofia
Sofia

Czechoslovakia
EFEKTIM
Engineering Consultants,
Siemens AG
Prague

Denmark
Siemens A/S
Copenhagen, Ballerup
Hojbjerg

Federal Republic
of Germany
Branch offices of the
Siemens AG
Berlin (West)
Bremen
Dortmund
Düsseldorf
Essen
Frankfurt/Main
Hamburg

Federal Republic
of Germany (continued)
Hanover
Cologne
Mannheim
Munich
Nuremberg
Saarbrücken
Stuttgart

Finland
Siemens Osakeyhtiö
Helsinki

France
Siemens S.A.
Paris, Saint-Denis
Lyon, Caluire-et-Cuire
Marseilles
Metz
Seclin (Lille)
Strasbourg

Great Britain
Siemens Ltd.
London, Sunbury-on-Thames
Birmingham
Bristol, Clevedon
Congleton
Edinburgh
Glasgow
Leeds
Liverpool
Newcastle

Greece
Siemens A.E.
Athens
Thessaloniki

Hungary
SICONTACT GmbH
Budapest

Iceland
Smith & Norland H/F
Reykjavik

Ireland
Siemens Ltd.
Dublin

Italy
Siemens S. p. A.
Milan
Bari
Bologna
Brescia
Casoria
Florence
Genoa
Macomer
Padua
Rome
Turin

Luxemburg
Siemens S.A.
Luxembourg

Malta
J.R. Darmanin & Co., Ltd.
Valletta

Netherlands
Siemens Nederland N.V.
The Hague

Norway
Siemens A/S
Oslo
Bergen
Stavanger
Trondheim

Poland
PHZ Transactor S.A.
Warsaw
Gda sk-Letnica
Katowice

Portugal
Siemens S.R.A.L.
Lisbon
Faro
Leiria
Porto

Romania
Siemens birou de
consulta ii tehnice
Bukarest

Spain
Siemens S.A.
Madrid

Sweden
Siemens AB
Stockholm
Eskilstuna
Göteborg
Jönköping
Luleå
Malmö
Sundsvall

Switzerland
Siemens-Albis AG
Zürich
Bern
Siemens-Albis S.A.
Lausanne, Renens

Turkey
ETMA
Istanbul
Adana
Ankara
Bursa
Izmir
Samsun

USSR
Siemens AG Agency
Moscow

Yugoslavia
General Export
OOUR Zastupstvo
Belgrade
Ljubljana
Rijeka
Sarajewo
Skopje
Zagreb

Non-European Companies and Representatives

Africa

Algeria
Siemens Bureau
Alger
Algier

Angola
Tecnidata
Luanda

Burundi
SOGECOM
Bujumbara

Egypt
Siemens Resident
Engineers
Cairo-Mohandessin
Alexandria
Centech
Zamalek-Cairo

Ethiopia
Addis Electrical
Engineering Ltd.
Addis Abeba

Ivory Coast
Siemens AG
Succursale Côte d'Ivoire
Abidjan

Kenya
Achelis (Kenya) Ltd.
Nairobi

Libya
Siemens AG
Branch Office Libya
Tripoli

Mauritius
Rey & Lenferna Ltd.
Port Louis

Morocco
SETEL
Société Electrotechnique
et de Télécommunications S.A.
Casablanca

Mozambique
Siemens Resident
Engineer
Maputo

Namibia
Siemens Resident
Engineer
Windhoek

Nigeria
Electro Technologies
Nigeria Ltd. (Eltec)
Lagos

Rwanda
Etablissement Rwandais
Kigali

Simbabwe
Electro Technologies
Corporation (Pvt.) Ltd.
Harare

South Africa
Siemens Ltd.
Johannesburg
Cape Town
Durban
Middleburg
Newcastle
Port Elizabeth
Pretoria

Sudan
National Electrical &
Commercial Company
(NECC)
Khartoum

Swaziland
Siemens (Pty.) Ltd.
Mbabane

Tanzania
Tanzania Electrical
Services Ltd.
Dar-es-Salaam

Tunesia
Sitelec S.A.
Tunis

Zaire
SOFAMATEL S.P.R.L.
Kinshasa

Zambia
Electrical Maintenance
Lusaka Ltd.
Lusaka
Mining projects:
General Mining
Industries Ltd.
Kitwe

America

Argentina
Siemens S.A.
Buenos Aires
Bahía Blanca
Córdoba
Mendoza
Rosario

Bolivia
Sociedad Comercial e
Industrial Hansa Ltd.
La Paz

Brazil
Siemens S.A.
São Paulo
Belém
Belo Horizonte
Brasília
Campinas
Curitiba
Florianópolis
Fortaleza
Porto Alegre
Recife
Rio de Janeiro
Salvador de Bahía
Vitoria

Canada
Siemens Electric Ltd.
Montreal, Québec
Toronto, Ontario

Chile
INGELSAC
Santiago de Chile

Colombia
Siemens S.A.
Bogotá
Baranquilla
Cali
Medellín

Costa Rica
Siemens S.A.
San José

Ecuador
Siemens S.A.
Quito
OTESA
Guayaquil
Quito

El Salvador
Siemens S.A.
San Salvador

Guatemala
Siemens S.A.
Ciudad de Guatemala

Honduras
Representaciones Electroindustriales S. de R.L.
Tegucigalpa

Mexico
Siemens S.A.
México, D.F.
Culiacán
Gómez Palacio
Guadalajara
León
Monterrey
Puebla

Nicaragua
Siemens S.A.
Managua

Paraguay
Rieder & Cia., S.A.C.I.
Asunción

Peru
Siemsa
Lima

Uruguay
Conatel S.A.
Montevideo

Venezuela
Siemens S.A.
Caracas
Valencia

United States
of America
Siemens Energy &
Automation Inc.
Roswell, Georgia

Asia

Bahrain
Transitec Gulf
Manama
or
Siemens Resident Engineer
Abu Dhabi

Bangladesh
Siemens Bangladesh Ltd.
Dhaka

Hong Kong
Jebsen & Co., Ltd.
Hong Kong

India
Siemens India Ltd.
Bombay
Ahmedabad
Bangalore
Calcutta
Madras
New Dehli
Secundarabad

Indonesia
P.T.Siemens Indonesia
Jakarta
P.T. Dian-Graha Elektrika
Jakarta
Bandung
Medan
Surabaya

Iran
Siemens Sherkate
Sahami Khass
Teheran

Iraq
Samhiry Bros. Co. (W.L.L.)
Baghdad
or
Siemens AG (Iraq Branch)
Baghdad

Japan
Siemens K.K.
Tokyo

Jordan
Siemens AG (Jordan
Branch)
Amman
or
A.R. Kevorkian Co.
Amman

Korea (Republic)
Siemens Electrical
Engineering Co., Ltd.
Seoul
Pusan

Kuwait
National & German
Electrical and Electronic
Service Co. (INGEECO)
Kuwait, Arabia

Lebanon
Ets. F.A. Kettaneh S.A.
Beirut

Malaysia
Siemens AG
Malaysian Branch
Kuala Lumpur

Oman
Waleed Associates
Muscat
or
Siemens Resident Engineers
Dubai

Pakistan
Siemens Pakistan
Engineering Co., Ltd.
Karachi
Islamabad
Lahore
Peshawer
Quetta
Rawalpindi

People's Republic of China
Siemens Representative Office
Beijing
Guangzhou
Shanghai

Philippine Islands
Maschinen & Technik Inc.
(MATEC)
Manila

Qatar
Trags Electrical Engineering
and
Air Conditioning Co.
Doha
or
Siemens Resident Engineer
Abu Dhabi

Saudi Arabia
Arabia Electric Ltd.
(Equipment)
Jeddah
Damman
Riyadh

Sri Lanka
Dimo Limited
Colombo

Syria
Siemens AG
(Damascus Branch)
Damascus

Taiwan
Siemens Liaison Office
Taipei
TAI Engineering Co., Ltd.
Taipei

Thailand
B. Grimm & Co., R.O.P.
Bangkok

United Arab Emirates
Electro Mechanical Co.
Abu Dhabi
or
Siemens Resident Engineer
Abu Dhabi
Scientechnic
Dubai
or
Siemens Resident Engineer
Dubai

Asia (continued)

Yemen (Arab Republic)
Tihama Tractors &
Engineering Co.o., Ltd.
Sanaa
or
Siemens Resident Engineer
Sanaa

Australasia

Australia
Siemens Ltd.
Melbourne
Brisbane
Perth
Sydney

New Zealand
Siemens Liaison Office
Auckland

Index

Siemens AG
AUT E1114B
Postfach 1963
Werner-von-Siemens-Str. 50

D-92209 Amberg
Fed. Rep. of Germany

From (please fill in)

Name ______________________

Company/Department ______________________

Address ______________________

Telephone /

Suggestions: □ **Corrections:** □

CP 523 Manual Release 3 (6ES5 998-0DD21)

Have you discovered any printing errors in this manual? If so, please list them on the preprinted form.
We would also appreciate any suggestions for improvement.

....

....

....

....

....

....

....

....

....

....

....

....

....

....