

SIEMENS

SIMATIC

CP 581

User Manual

This manual has the order number:
6ES5998-2AT22

08/99
Release 02

Preface, Contents

Installation and Commissioning, Hardware Reference Information

Introduction to Application of the
CP 581

1

Installation and Commissioning

2

Reference Section for Hardware

3

Technical Data of the CP 581

4

Ordering Information

5

System Software

Operation of CP 581 in
S5 Programmable Controllers

6

Process Data Acquisition
CP RECORD

7

Mass Storage Functions

8

Command Interpreter

9

Virtual S5 Drive

10

Free Programming of the
CP 581

11

Application Examples

12

Reference Section for System
Software

13

Appendices

List of Abbreviations

A

Guidelines for Handling
Electrostatic Sensitive Devices
(ESD)

B

Index

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:



Danger

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



Warning

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



Caution

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

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this manual.

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:



Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

SIMATIC[®], SIMATIC NET[®] and SIMATIC HMI[®] are registered trademarks of SIEMENS AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Copyright © Siemens AG 1997 All rights reserved

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

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D-90327 Nuernberg

Disclaimer of Liability

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

© Siemens AG 1997
Subject to change without prior notice.

Preface

Safety Information and ESD Guidelines

In this manual you will find the “Safety Information” on the rear of the front sheet and the “ESD (electrostatically sensitive devices) Guidelines” in the appendix. These must be exactly observed and followed whenever working with the CP 581.

README File

You can find information produced at a later date than the printing of this manual in the README.TXT in directory C:\CP581 on the hard disk of the CP 581. You can read and/or print this file using any word-processing program.

Scope of Validity

The manual describes the following versions of the CP 581 and corresponding CP 581 software:

a. Hardware versions

6ES5581-0ED13	CP 581 basic board with 80486DX processor,
6ES5581-0ED14	CP 581 basic board with 80486DX processor,
6ES5581-0EE15	CP 581 basic board with Pentium processor,
6ES5581-1EE15	CP 581 basic board with Pentium processor,
6ES5581-3LA11	CP 581 mass storage module with 540 MB
6ES5581-4LA11	CP 581 mass storage module with 1.4 GB
6ES5581-0RA12	CP 581 AT slot module

b. CP 581 System software

6ES5835-8MD01 from V 2.5 to V 2.8 (only for the CP 581 with 80486 processor)

c. CP 581 System software

6ES5835-8MD01 from V 2.9 and above (for the CP 581 with Pentium or 80486 processor)

Where not otherwise stated, all information in this manual applies to both the CP 581 with the 80486DX processor and the CP 581 with the Pentium processor.

How the Manual is Organized

The manual is divided into two parts:

- Part 1 describes the installation and commissioning of the CP 581 and contains detailed information on the hardware in the form of a reference section.
- Part 2 describes how you can use the CP 581 system software for your applications and provides instructions for using the various features and functions. Additionally, a reference section provides important information on various applications.

Note

The basic board 6ES5581-0ED14 differs from the 6ES5581-0ED13 only in the higher working frequency of the processor.

6ES5581-0ED13: 75 MHz / 100 MHz

6ES5581-0ED14: 100 MHz / 133 MHz

The basic board 6ES5581-0EE15 differs from the 6ES5581-1EE15 only in graphics mode.

6ES5581-0EE15: RGB

6ES5581-1EE15: Standard VGA

The mass storage module 6ES5581-4LA11 differs from the 6ES5581-3LA11 only in larger hard disk capacity.

6ES5581-3LA11: 540 MB

6ES5581-4LA11: 1440 MB

**Information on Use
of Part 1**

Part 1 is divided into two main parts:

Chapters 1 and 2 are *task-oriented* and describe in sequential order the work which you must carry out as the user. Chapters 3 to 5 are *system-oriented*; they provide information in compressed form and can be used as a reference work for users of all levels of experience.

Part 1 is thus suitable for first-time users and also as a reference work for experienced users. You can skip sections of the manual depending on your knowledge and the application, and concentrate on the relevant parts.

The contents of the individual chapters of Part 1 are summarized below to help you to become acquainted with the manual:

Chapter 1

The **Introduction to Application of the CP 581** describes the possible uses offered by the CP 581 as well as its application scope.

Chapter 2

describes the scope of delivery of the CP 581 components, peripheral configurations and the mechanical assembly of the CP 581 components into a complete module. It also provides all you need to know about the **installation** of the hardware and the **commissioning** of the software.

Chapter 3

The Hardware Reference Section (device description) contains all of the relevant information on the hardware components of CP 581. Each component is described separately. This chapter also contains the SETUP description and information for developers of their own system software.

Chapter 4

provides the technical data of the CP 581. You will find data on, for example, current consumption, electromagnetic compatibility and the climatic and mechanical ambient conditions.

Chapter 5

contains ordering information for accessories and peripheral devices (valid at the time of manual publication).

Information on Use of Part 2

Part 2 of the manual describes the application of the CP 581 system software. It is divided into two main parts:

Chapters 6 to 12 are *task-oriented* and describe in sequential order the work which you must carry out as the user.

Chapter 13 is *system-oriented*; it provides information in compressed form and can be used as a reference work for users of all levels of experience.

Part 2 is thus suitable for first-time users and also as a reference work for experienced users. You can skip sections of the manual depending on your knowledge and the application, and concentrate on the relevant parts.

The contents of the individual chapters of Part 2 are summarized below to help you to become acquainted with the manual:

Chapter 6 describes the **Operation of the CP 581 in the S5 Programmable Controllers** and informs you of the programmable controllers in which you can use the CP 581 and with which applications multiprocessor operation is possible. It describes how the S5 CPU and the CP 581 work together and which CP 581 applications can be executed simultaneously.

Chapter 7 describes how you can transmit process data from the S5 CPU to the CP 581 using the **Process Data Acquisition** function and how to evaluate the data on the CP 581 using standard MS-DOS programs.

Chapter 8 describes how you can use the CP 581 as a storage medium for S5 CPUs using the **Mass Storage Functions**.

Chapter 9 describes how you can activate any MS-DOS commands on the CP 581 from an S5 CPU using the **Command Interpreter** function.

Chapter 10 describes how you can use the **Virtual S5 Drive** function to address S5 data handling blocks DB/DX like MS-DOS files from the CP 581.

Chapter 11 The **Free Programming** section describes how to use the system functions of the CP 581 and the functions of the data handling blocks and how to apply them to optimally adapt your programmable controller system to the tasks at hand.

Chapter 12 describes use of the CP 581 using **Application Examples** for the four system functions, process data acquisition, mass storage functions, command interpreter and free programming.

Chapter 13, Reference Section for System Software, contains all important information on the system software in a compressed, table form. The SETUP routines for handling blocks used for the CP 581, format, parameterization and the error bits of the four system functions are described.

Contents

1	Introduction to Application of the CP 581	1-1
1.1	What is a CP 581 and What Facilities Does it Provide You With?	1-2
1.2	What Applications are Possible for the CP 581?	1-4
1.3	What Tasks can the CP 581 Handle in an Automation Network?	1-6
2	Installation and Commissioning	2-1
2.1	Unpacking and Checking the Delivered Components	2-2
2.2	General Information on Commissioning of the CP 581	2-4
2.3	Connectable Peripheral Devices/Interface Configuration	2-5
2.3.1	Configuration 1: Short Distance CP 581 <-> Operator Console	2-5
2.3.2	Configuration 2: Long Distance CP 581 <-> Operator Console	2-7
2.4	Checklist for Installation and Commissioning	2-8
2.5	Switch and Jumper Settings on the CP 581 Components	2-9
2.6	Installing the Memory Modules and the Onboard Silicon Disk into the CP 581 Pentium	2-9
2.6.1	Installing the Memory Modules into the CP 581 Pentium	2-10
2.6.2	Installing the Onboard Silicon Disk into the CP 581 Pentium	2-11
2.7	Assembly of Individual Components into Complete Module	2-12
2.7.1	Information on Assembly of Individual Components	2-12
2.7.2	Assembly of Mass Storage Module on Basic Board	2-13
2.7.3	Assembly of AT Slot Modules	2-15
2.7.4	Assembly of an AT Board onto an AT Slot Module	2-25
2.8	Installation of Complete Module into Subrack	2-27
2.8.1	Interference-Free Hardware Configuration	2-27
2.8.2	Selection of CP 581 Slots in the Programmable Controller	2-29
2.8.3	Switching Off the Power Supply of the PLC Rack	2-31
2.8.4	Installation of CP 581 into PLC Rack	2-31
2.9	Connection of Operation Devices and Peripheral Devices	2-32
2.9.1	Connection of Keyboard	2-32
2.9.2	Connection of Monitor	2-33
2.9.3	Connection of a Printer	2-35
2.9.4	Connection of a Mouse	2-35
2.9.5	Maximum Cable Lengths for Connection of Operation Devices and Peripheral Devices	2-36
2.10	Commissioning (with a Mass Storage Module)	2-37
2.11	Installation and Commissioning of the Software	2-37
2.11.1	Installation with Mass Storage Module	2-38

2.11.2	SETUP for Individual Operation of the CP 581 Basic Board with Silicon Disk	2-41
2.12	CPLINK	2-46
2.12.1	Use, Functions and Structure of CPLINK	2-46
2.12.2	Installation of CPLINK	2-49
2.12.3	Technical Information	2-52
2.13	Remote Operation via Serial Interface using CPLINK	2-54
2.13.1	Remote BIOS Setup for the CP 581 DX	2-55
2.13.2	Remote BIOS Setup for the CP 581 Pentium	2-57
3	Reference Section for Hardware	3-1
3.1	Basic Board	3-2
3.1.1	Mechanical Construction	3-2
3.1.2	Controls and Displays	3-6
3.1.3	Switch Settings for the CP 581 80486 DX	3-9
3.1.4	Switch Settings for the CP 581 Pentium	3-11
3.1.5	Connection of Printers	3-14
3.1.6	Connection of Keyboard	3-14
3.1.7	Connection of a Mouse	3-15
3.1.8	Connection of a Monitor	3-15
3.1.9	External Interface Assignments	3-16
3.1.10	Internal Interface Assignments	3-24
3.1.11	Memory and Hardware Interrupt Assignments	3-26
3.2	Mass Storage Module	3-29
3.2.1	Mechanical Construction	3-29
3.2.2	Positions of Floppy Disk Drive and Interfaces on Front Panel	3-31
3.2.3	Floppy disk drive	3-32
3.2.4	Hard Disk Drive	3-32
3.2.5	Switch and Jumper Settings	3-33
3.2.6	Connection of Devices to Centronics Interface (LPT)	3-34
3.2.7	Connection of Devices to the Serial Interface COM 4	3-34
3.2.8	External Interface Assignments of Mass Storage Module	3-35
3.2.9	Serial Interface COM 4	3-36
3.2.10	Internal Interface Assignments	3-37
3.3	AT Slot Module	3-38
3.3.1	Mechanical Construction of AT Slot Module	3-38
3.3.2	Switch and Jumper Settings	3-39
3.3.3	External Interface Assignments	3-39
3.3.4	Internal Interface Assignments	3-41
3.4	Remote Terminal Interface	3-42
3.4.1	Power Supply of RTI Via Remote Supply From CP 581	3-44
3.4.2	Power Supply of RTI by Local, External Supply	3-44
3.4.3	Connection of Keyboard to the RTI	3-44
3.4.4	Connection of a Mouse to the RTI	3-44
3.4.5	Jumper Assignments on the RTI Module	3-45
3.4.6	Interface and Pin Assignments on the RTI	3-46
3.4.7	CP 581 <-> RTI Connection Cable	3-46
3.5	Y Adapter	3-48
3.5.1	Assignments of Y Adapter Keyboard/COM 2	3-48

3.5.2	Assignments of Y Adapter COM 1/COM 3	3-49
3.6	Silicon Disk	3-50
3.6.1	Drive Assignments of Silicon Disks (OSD and MC)	3-50
3.6.2	Formatting of Silicon Disks	3-51
3.7	BIOS Setup for CP 581 80486 DX	3-52
3.8	BIOS Setup for CP 581 Pentium	3-61
3.9	Conversion and Repairs	3-70
3.10	Information for Developers of Own System Software	3-70
3.10.1	S5 interface	3-70
3.10.2	Interrupt Processing on S5 Interface	3-77
3.10.3	Other Status Registers	3-79
3.10.4	Peripheral Byte Addressing	3-80
4	Technical Data of the CP 581	4-1
4.1	Basic Board	4-3
4.2	Mass Storage Module	4-8
4.3	AT Slot Module	4-12
4.4	RTI Module	4-17
5	Ordering Information	5-1
6	Operation of CP 581 in S5 Programmable Controllers	6-1
6.1	Programmable Controllers for CP 581	6-2
6.1.1	Single Processor and Multiprocessor Operation	6-2
6.2	Operational Components	6-3
6.2.1	S5 Backplane Bus and Pages	6-3
6.2.2	Data Handling Blocks	6-4
6.2.3	CPDHB Driver	6-4
6.2.4	Principle Interaction Between CPU and CP 581	6-6
6.2.5	Synchronize CP 581 with CPU	6-6
6.2.6	Call CPDHB Driver for Special Application	6-6
6.2.7	Carry Out Data Transfer	6-7
6.3	Simultaneous Operation of CP 581 Application	6-8
6.4	Notes on Use of CP 581 System Software	6-10
7	Process Data Acquisition CPRECORD	7-1
7.1	Application	7-2
7.2	Principal Sequences Between CPU and CP	7-3
7.3	Process Data Acquisition Operations	7-4
7.3.1	Related Procedures	7-4
7.3.2	Measures on the CP	7-5
7.3.3	Programming of CPU for Data Transfer	7-13
7.3.4	Activation, Deactivation and Testing of Process Data Acquisition	7-19
7.4	Evaluation of Acquired Process Data	7-26
7.4.1	Storage of Process Data on the CP 581	7-26
7.4.2	Structure of Process Data in the ASCII Files	7-28

7.4.3	Converting the Individual Data	7-29
7.4.4	Example of "Individual" Conversion	7-31
7.5	Control of Process Data Acquisition	7-32
7.5.1	Control of Acquisition by a CP 581 Command	7-32
7.5.2	Control of Acquisition by a CPU	7-34
7.6	Information on Special Handling	7-37
7.6.1	Information on Parameters and Status	7-37
7.6.2	Reactivating Process Data Acquisition	7-39
7.7	Status Messages	7-41
8	Mass Storage Functions	8-1
8.1	Application	8-2
8.2	Principle Sequences Between CPU and CP	8-3
8.3	Mass Storage Function Operations	8-7
8.3.1	Related Procedures	8-7
8.3.2	Measures on the CP	8-7
8.3.3	Programming the CPU	8-8
8.3.4	Activation and Testing of the Mass Storage Functions	8-26
8.4	Error Bits	8-29
8.4.1	Parameter Assignment Error Bits	8-29
8.4.2	Job Status Bits	8-31
8.5	Setting of Timeout	8-33
9	Command Interpreter	9-1
9.1	Application	9-2
9.2	Principle Sequences Between CPU and CP	9-2
9.3	Command Interpreter Operations	9-4
9.3.1	Related Procedures	9-4
9.3.2	Measures on the CP	9-5
9.3.3	Defining the Command Output	9-5
9.3.4	Programming the CPU	9-6
9.3.5	Example of DHB Parameter Assignment for the Command Interpreter ..	9-12
9.3.6	Activation/Termination and Testing of the Command Interpreter	9-15
9.4	Error bits	9-17
9.4.1	Parameter Assignment Error Bits	9-17
9.4.2	Job Status Bits	9-18
9.5	Special Features During Command Interpretation	9-20
10	Virtual S5 Drive	10-1
10.1	Application	10-2
10.2	Principle Sequences Between CPU and CP	10-2
10.3	Preparation of Application	10-4
10.3.1	Related Procedures	10-4
10.3.2	Measures on the CP	10-5
10.3.3	Programming the CPU	10-10

10.4	Activation, Deactivation and Testing of the Function	10-17
10.4.1	Activation	10-17
10.4.2	Deactivation	10-19
10.4.3	Testing	10-19
10.5	Use of Virtual S5 Drive	10-21
10.5.1	Operations with Directories and Files	10-21
10.5.2	Writing in S5 Data Blocks	10-22
10.5.3	Representation of S5 Data on the CP 581	10-23
10.5.4	Use of MS-DOS Commands and Applications	10-26
10.5.5	Error Status Bits	10-27
10.6	Conversion of Preheader Data Using DVCONVRT	10-30
11	Free Programming of the CP 581	11-1
11.1	Application	11-3
11.2	Procedure	11-3
11.2.1	Summary	11-3
11.2.2	Analysis of Task	11-4
11.3	Programming of DHB Calls	11-5
11.3.1	General Information	11-5
11.3.2	Available Data Handling Blocks	11-6
11.3.3	Parameters of Data Handling Blocks	11-7
11.3.4	Parameter Description	11-7
11.3.5	Direct and Indirect Parameterization	11-12
11.3.6	Format and Meaning of the Status Word	11-16
11.3.7	SEND Block	11-20
11.3.8	RECEIVE Block	11-22
11.3.9	FETCH Block	11-24
11.3.10	CONTROL Block	11-25
11.3.11	RESET Block	11-26
11.3.12	SYNCHRON Block	11-27
11.4	Programming the CP 581 User Program	11-28
11.4.1	CPDHB Driver	11-29
11.4.2	Transfer Control Block (TCB)	11-32
11.4.3	Summary of Driver Functions	11-37
11.4.4	Example of Call of CPDHB Driver	11-38
11.4.5	Data Transfer with Direct Jobs	11-39
11.4.6	Data Transfer Without Direct Jobs	11-52
11.4.7	Other Driver Functions	11-55
11.5	Testing the Application	11-58
11.5.1	Procedure	11-58
11.5.2	Testing the S5 Program	11-58
11.5.3	Testing the CP 581 Program	11-59
11.5.4	Representation of the S5 Data in the CP 581 Memory	11-59
12	Application Examples	12-1
12.1	Example "Process Data Acquisition"	12-2
12.1.1	Task/Problem	12-2
12.1.2	Starting to Solve the Problem	12-2
12.1.3	Structure of Solution	12-3
12.1.4	Individual Working Steps	12-5

12.2	Example "Mass Storage Functions"	12-13
12.2.1	Task/Problem	12-13
12.2.2	Starting to Solve the Problem	12-13
12.2.3	Structure of Solution	12-14
12.2.4	Individual Working Steps	12-16
12.3	Example "Command Interpreter"	12-22
12.3.1	Task/Problem	12-22
12.3.2	Starting to Solve the Problem	12-22
12.3.3	Structure of Solution	12-22
12.3.4	Individual Working Steps	12-24
12.4	Application Examples in the Directory "EXAMPLE"	12-28
13	Reference Section for System Software	13-1
13.1	Data Handling Blocks	13-3
13.1.1	Summary of DHBs with the CP 581	13-3
13.1.2	DHB SYNCHRON	13-4
13.1.3	DHB SEND	13-5
13.1.4	DHB FETCH	13-6
13.1.5	DHB RECEIVE	13-6
13.1.6	DHB CONTROL	13-8
13.1.7	DHB RESET	13-8
13.1.8	Status Word	13-9
13.1.9	Parameter Assignment Error Bits	13-10
13.2	Options of the CPDHB Driver	13-11
13.2.1	Options in the Load Command/Setting the Base Interface Number	13-11
13.2.2	Options on the Installed Driver	13-11
13.3	Process Data Acquisition	13-12
13.3.1	Parameters for the Configuration File	13-12
13.3.2	Command Options of CPRECORD and CPRECCTL	13-14
13.3.3	Control of Process Data Acquisition from a CPU	13-16
13.3.4	Error and Status Messages of the CPDHB Driver and the CPRECORD Program	13-17
13.4	Mass storage functions	13-20
13.4.1	Data handling blocks for the mass storage functions	13-20
13.4.2	Error Bits of CPMASS Program	13-21
13.5	Command Interpreter	13-22
13.5.1	Data Handling Blocks for the Command Interpreter	13-22
13.5.2	Error Bits of CPSHELL Program	13-23
13.6	Virtual S5 Drive	13-24
13.6.1	Data Handling Blocks for the Virtual S5 Drive	13-24
13.6.2	Sequence Parameters for the Virtual Drive	13-25
13.6.3	Format File	13-27
13.6.4	Error Status Bits	13-28
13.7	Free Programming	13-30
13.7.1	DHB Status Bits	13-30
13.7.2	Calling and Parameterizing the CPDHB Driver	13-33
13.7.3	Functions of the CPDHB Driver	13-36
13.7.4	Error Bits of the CPDHB Driver	13-37

A List of Abbreviations A-1

B Guidelines for Handling Electrostatic Sensitive Devices (ESD) B-1

 B.1 What is ESD? B-2

 B.2 Electrostatic Charging of Persons B-3

 B.3 General Protective Measures Against Electrostatic Discharge Damage . B-4

Index Index-1

Introduction to Application of the CP 581

1

This chapter provides you with an initial overview of the CP 581.
You will learn:

- What a CP 581 is, and which facilities it provides you with,
- What applications are possible for the CP 581,
and
- What tasks the CP 581 handles in an automation network.

1.1 What is a CP 581 and What Facilities Does it Provide You With?

Definition of a CP 581

The CP 581 is an AT computer compatible with the industrial standard. It is fitted in your programmable controller subrack.

It consists of hardware modules which you can combine in certain configurations:

- Basic board with 80486DX or Pentium processor,
- Mass storage module with floppy disk drive and hard disk,
- AT slot module for plugging in a short AT board.

Hardware Configurations

You can operate the CP 581 in the following hardware configurations:

- a) Only with basic board,
- b) With basic board and mass storage module,
- c) With basic board, mass storage module and one AT slot module,
- d) With basic board, mass storage module and two AT slot modules,
- e) With basic board and one AT slot module,
or
- f) With basic board and two AT slot modules.

The memory card can be addressed using MS-DOS just like a floppy disk drive, the onboard silicon disk just like a hard disk.

You can order the memory card as a **"Flash card"** or a **"RAM card"** (see Chapter "Ordering Information").

Note that when using a RAM card, the data you have saved on the card will be lost as soon as you remove the RAM card from the CP 581, or the CP 581 from the PLC rack. Data will also be lost if the battery for the PLC rack does not work.

You require a mass storage module for installation of the system software, but you can connect this module to a different CP 581 following the installation. If you wish to operate the basic board on its own or together with one or two AT slot modules, you must transfer your software to a memory card or to the onboard silicon disk.

To do this you can also use the CP LINK program, with which you can load your software from a programming device/PC to the CP 581 (to the memory card or OSD only).

The CP 581 provides you with additional computing performance in conjunction with the S5 CPU in order to solve your automation task. Direct communication with the S5 CPU via the internal S5 rearpanel bus enables effective data transfer between the S5 CPU and the CP 581.

To enable a meaningful distribution of tasks between the components of the programmable controller, the S5 CPU is assigned execution of the control tasks, whereas the CP 581 handles the acquisition, storage, management and conditioning of larger quantities of data. Standard MS-DOS user programs can be used for this CP 581 application. The software required for this comprises the MS-DOS operating system and a number of utilities specific to the CP 581 for communication between the MS-DOS and S5 environments.

1.2 What Applications are Possible for the CP 581?

The possible applications of the CP 581 can be divided into the five system functions:

- Process data acquisition,
- Mass storage functions,
- Command interpreter,
- Virtual S5 drive,
- Free programming.

Note

You can use the STEP 5 basic package for PCs when using the CP 581 with a basic board and mass storage module (see Chapter 5 - Ordering Information).

Process Data Acquisition

Examples of applications for the process data acquisition function:

- Recording of process data and subsequent processing using standard MS-DOS programs
 - To evaluate and analyze the process,
 - For central management of process data,
 - For long-term quality monitoring,
 - For statistics (data compression, short-term storage, quality assurance, optimization).
- Evaluation of measured data in conjunction with a modular message printing system:
 - For continuous monitoring of binary process signals,
 - For monitoring of process operations,
 - To unload the CPU of the programmable controller system in the case of comprehensive logging operations.
- Handling of data management for all programmable controllers in networked systems to unload the user memory on the S5 CPU.

Mass Storage Functions

Examples of applications for the mass storage functions:

- Transfer of larger quantities of process data to unload the user memory on the S5 CPU.
- Buffering of process data should the next higher computer level fail to prevent data losses.
- Recipe management in weighing and dosing systems (quantity control) to transfer recipes which are not currently required to the memory of the CP 581.

Command Interpreter

Examples of applications for the command interpreter function:

- Calling of MS-DOS commands from the S5 CPU to activate programs for execution on the CP 581.
- Printing of S5 CPU messages on a message printer for storage of status and error messages.

Virtual S5 Drive

Examples of applications for the virtual S5 drive function:

- Access to S5 data blocks from the CP 581
 - For simple handling using MS-DOS commands (for example DIR, TYPE, PRINT, COPY),
 - For adaptation of existing programs for data transfer with S5 data blocks using MS-DOS file functions,
 - For free programming with data blocks, for example in C with "fopen()", "fread()", "fwrite()", etc., for data exchange with S5 data blocks.

Free Programming

Examples of applications for the free programming function:

- Use of self-generated applications in ASM, C, PASCAL, etc.
 - For optimum adaptation of your programmable controller system to the tasks,
 - For adaptation of existing programs to communication via the S5 rearpanel bus,
 - To implement the cell level of a manufacturing system according to Figure 1.1

CPLINK

Examples of applications for the CPLINK function:

- Loading the onboard silicon disk or memory card from a programming device/PC in remote operation; the memory of the CP 581 can be completely "empty."
- Physical link to the programming device via the COM 1 serial interface.
- Access from the programming device/PC to the CP 581 via two additional logical drives.

1.3 What Tasks can the CP 581 Handle in an Automation Network?

The CP 581 as a computer local to the process can establish connections between the process, i.e. the manufacturing operations, on the one hand and the computers of the coordinating and planning levels, i.e. the management, on the other.

Figure 1-1 shows a possible task of the CP 581 in the automation pyramid.

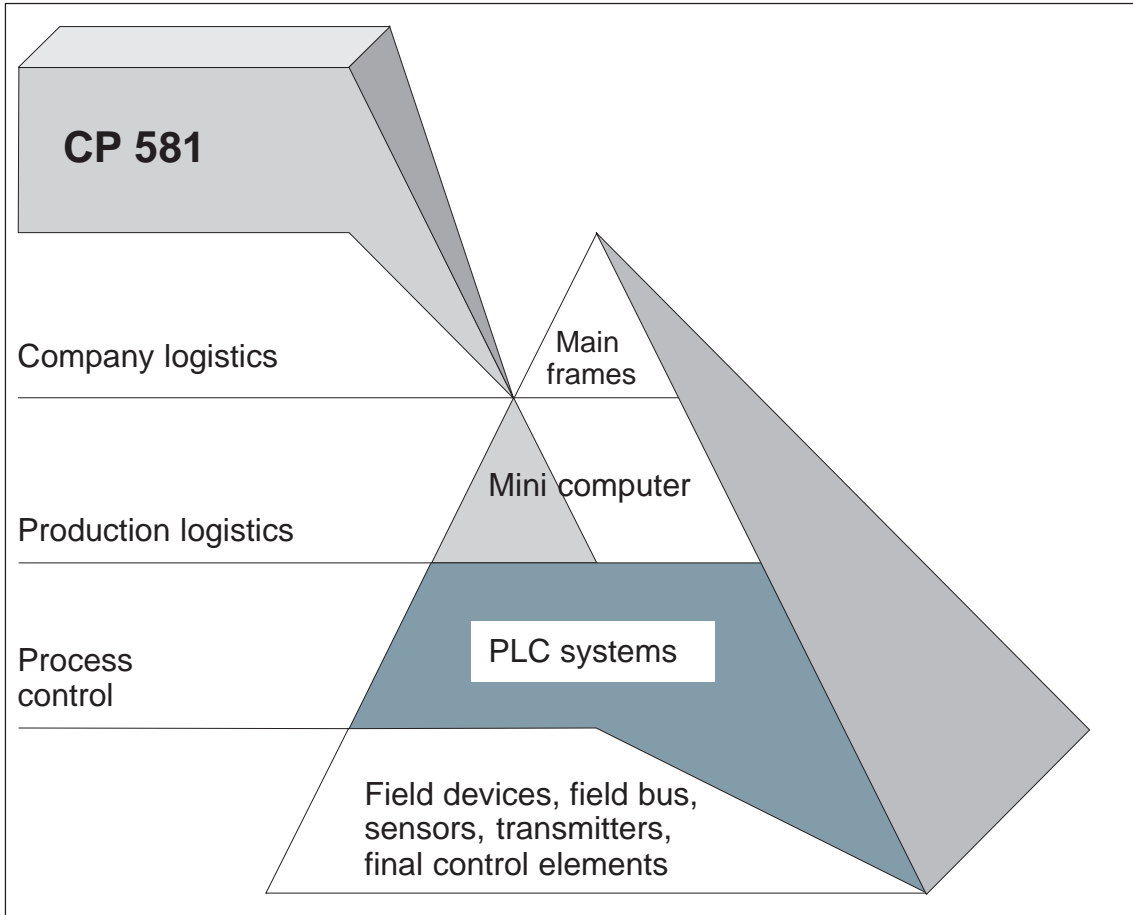


Figure 1-1 The CP 581 Communications Processor in the Automation Pyramid

Installation and Commissioning

2

This chapter describes the installation and commissioning of the CP 581.
Please also observe the installation guidelines described in this chapter.

Chapter Overview

Section	Description	Page
2.1	Unpacking and Checking the Delivered Components	2-2
2.2	General Information on Commissioning of the CP 581	2-4
2.3	Connectable Peripheral Devices/Interface Configuration	2-5
2.4	Checklist for Installation and Commissioning	2-8
2.5	Switch and Jumper Settings on the CP 581 Components	2-9
2.6	Installing the Memory Modules and the Onboard Silicon Disk into the CP 581 Pentium	2-9
2.7	Assembly of Individual Components into Complete Module	2-12
2.8	Installation of Complete Module into Subrack	2-27
2.9	Connection of Operation Devices and Peripheral Devices	2-32
2.10	Commissioning (With a Mass Storage Module)	2-37
2.11	Installation and Commissioning of the Software	2-37
2.12	CPLINK	2-46
2.13	Remote Operation via Serial Interface using CPLINK	2-54

2.1 Unpacking and Checking the Delivered Components

1. Unpack the CP 581 components.
2. Retain the original packing material for subsequent transport.

This manual was ordered separately. Please observe the language-specific order numbers (see Chapter 5, Ordering Information).

Standard Scope of Delivery of Basic Board

Check the scope of delivery! On delivery the CP 581 basic board includes:

- CP 581 basic board: hardware

Check that the order No. printed on your CP 581 basic board agrees with the configuration you ordered.

Current configuration of the CP 581 basic board:

- 6ES5 581-0ED14
CP 581 basic board with 80486 DX processor without user memory or silicon disk
- 6ES5 581-0EE15
CP 581 basic board with Pentium processor and RGB interface without user memory or silicon disk
- 6ES5 581-1EE15
CP 581 basic board with Pentium processor and VGA interface without user memory or silicon disk

Standard Scope of Delivery of Mass Storage Module

Check the scope of delivery! On delivery the CP 581 mass storage module includes:

- 6ES5 581-4LA11
CP 581 mass storage module and
- Assembly kit.

Note

The CP 581 mass storage module is equipped with a disk drive which is sensitive to shock and vibration. Please remember this when handling the system.

Only transport this module in its original packing material!

Standard Scope of Delivery of AT Slot Module

Check the scope of delivery! On delivery the CP 581 AT slot module includes:

- CP 581 AT slot module and
- Assembly kit.

**Standard
Scope of Delivery
of Remote
Terminal Interface
(RTI)**

Check the scope of delivery! On delivery the remote terminal interface includes:

- RTI module.

You additionally require the

- Connection cable for RTI listed in Chapter 5, Ordering Information.

Y Adapter

You require so-called Y adapters to connect a keyboard, a mouse or devices with standard plugs:

- Y adapter for keyboard/mouse with:
 - 26-pin high-density plug for connection to the Keyboard/COM 2 interface of the CP 581,
 - 7-pin round socket for connection of a keyboard, and
 - 9-pin D-subminiature plug for connection of a mouse.

The following transitional adapters are enclosed:

- 7-pin round socket or Mini-Din round socket for connection of a keyboard with Mini-Din round socket (PS/2).
- Y adapter for devices with standard plugs for serial interfaces with:
 - 26-pin high-density plug for connection to the COM 1/COM 3 or COM 4 interface of the CP 581,
 - 25-pin subminiature plug for the COM 1 interface, and
 - 25-pin subminiature plug for the COM 3 interface.

You will find the order numbers for the Y adapters in Chapter 5, Ordering Information.

2.2 General Information on Commissioning of the CP 581

To commission the CP 581 you also need:

- A monitor with
 - Cable connector for the monitor and
 - Instructions for the monitor
- A keyboard with
 - Instructions for the keyboard.

The environmental conditions and technical data applicable to the CP 581 are listed in Chapter 4, Technical Data.

Note

It is essential to observe the current consumption of the CP 581 when equipping the programmable controller! This can be obtained from Chapter 4, Technical Data.

2.3 Connectable Peripheral Devices/Interface Configuration

2.3.1 Configuration 1: Short Distance CP 581 <--> Operator Console

Figure 2-1 and 2-2 show the connections possible to the CP 581 when using a basic board and a mass storage module in the interface configuration “Short distance CP 581 <--> operator console.”

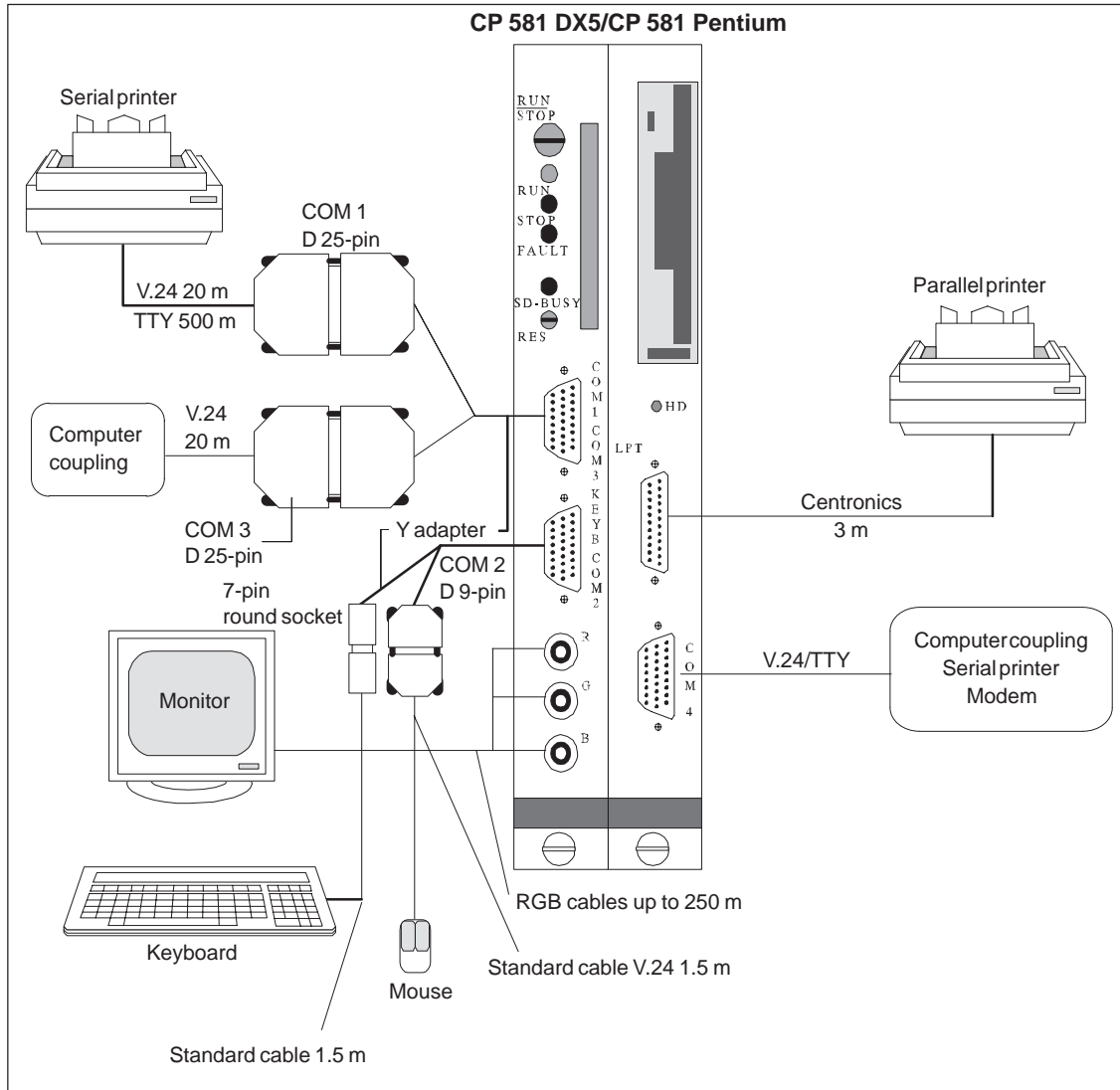


Figure 2-1 Possible Connections with Keyboard, Monitor, Printer, and Mouse (Configuration 1.1, RGB Interface)

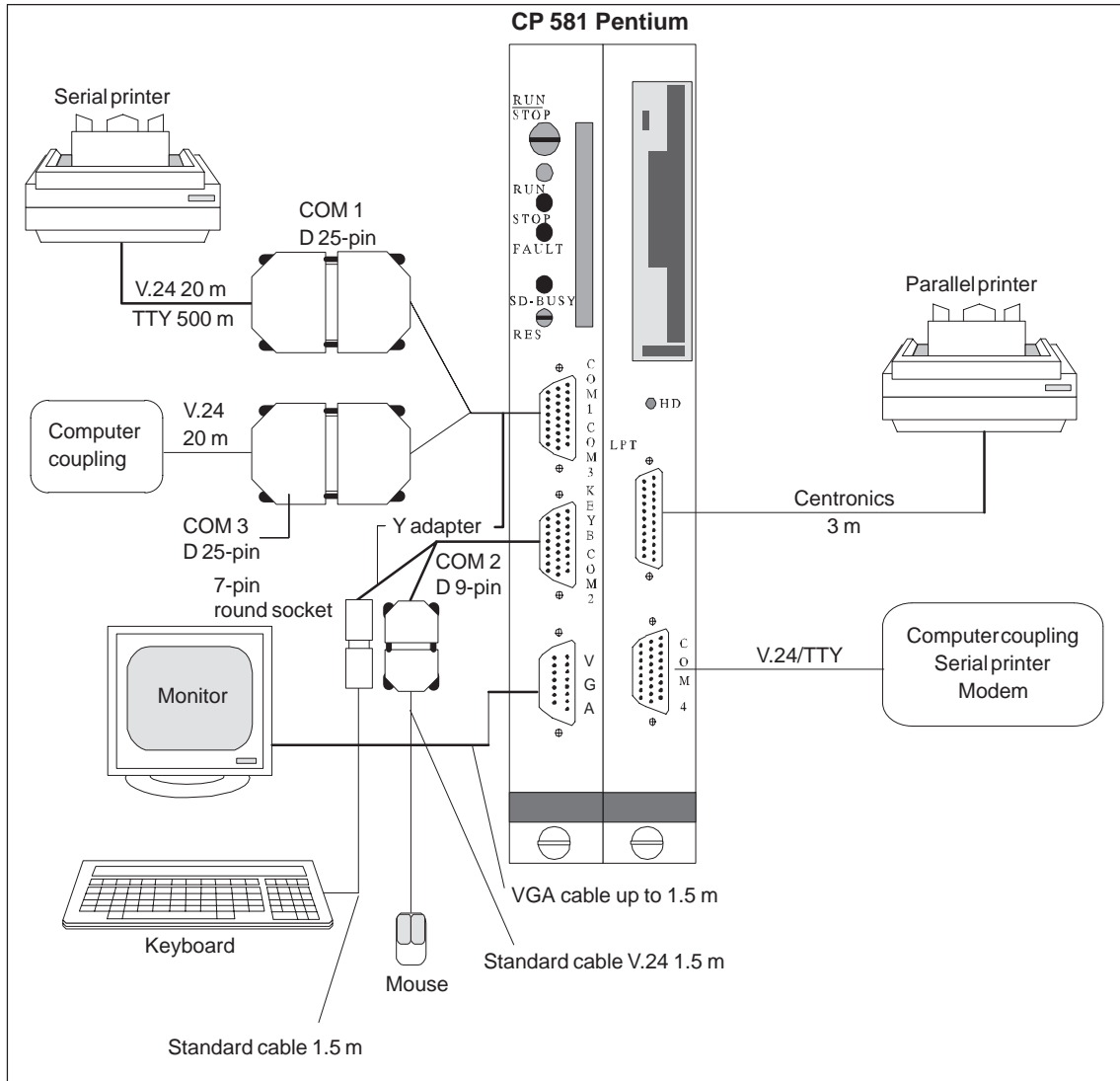


Figure 2-2 Possible Connections with Keyboard, Monitor, Printer, and Mouse (Configuration 1.2, VGA Interface)

2.3.2 Configuration 2: Long Distance CP 581 <-> Operator Console

Figure 2-3 shows the connections possible to the CP 581 when using a basic board and a mass storage module in the interface configuration “Long distance CP 581 <-> operator console.”

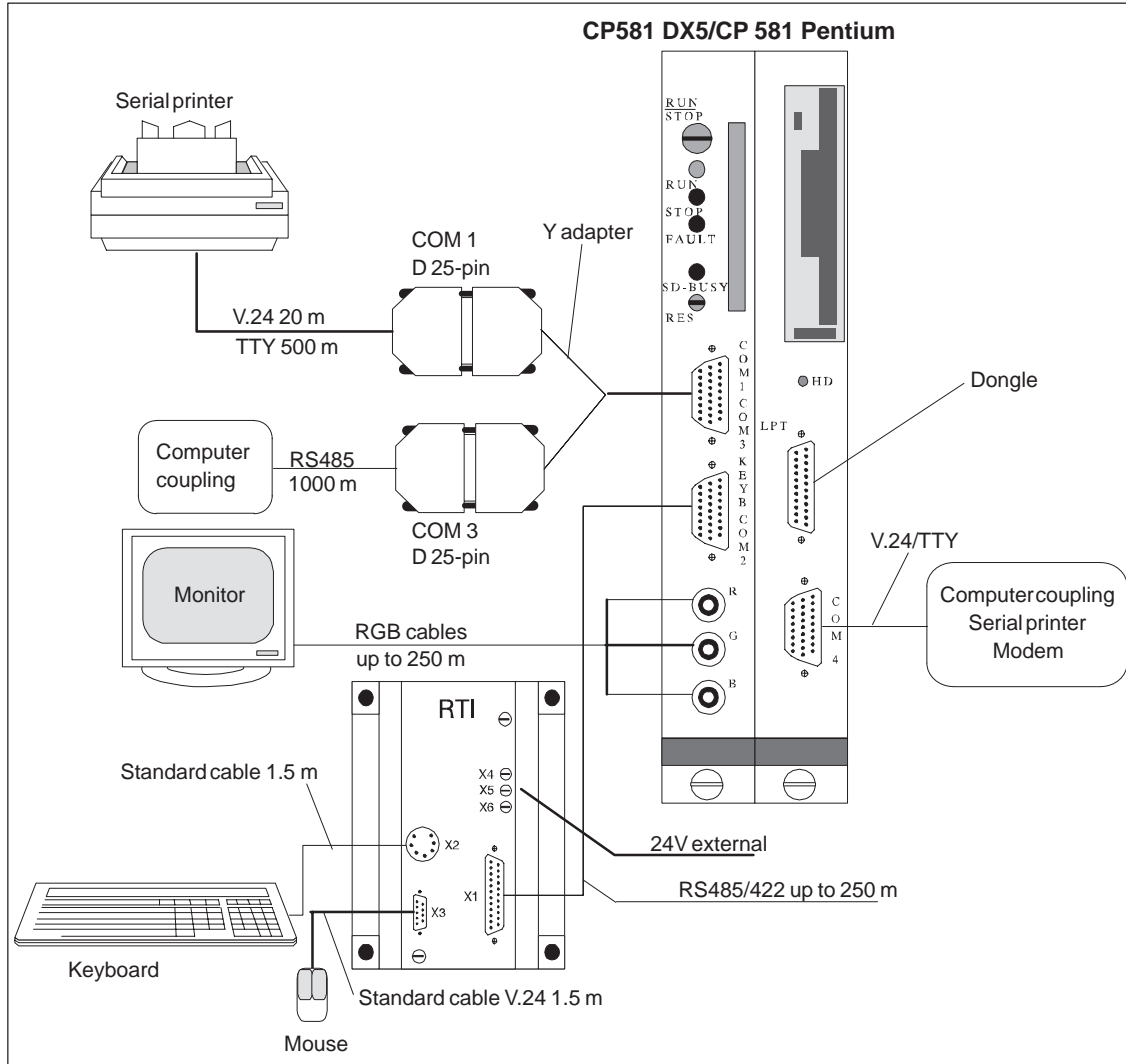


Figure 2-3 Possible Connections with Keyboard, Monitor, Printer, and Mouse (Configuration 2)

2.4 Checklist for Installation and Commissioning

This section explains the procedure for installing and setting up the CP 581 components step-by-step. Please proceed as described below:

1. Is the power supply unit in your subrack correctly dimensioned?
2. Fit the CP 581 components to produce the complete CP 581 module.
3. Switch off the power supply to your programmable controller rack and insert the complete CP 581 module into the provided slot.
4. Check the position of the mode switch.
The switch must be in the "RUN" position.
5. Connect the required operation devices and peripheral devices.
6. Switch the peripheral devices on.
7. Switch the power supply to your programmable controller rack on again.
8. Set the date and time using the SETUP menu (when setting up for the first time).
9. Install MS-DOS.
10. Start up MS-DOS and generate backup copies of MS-DOS and the CP 581 system software.
11. Install the CP 581 system software.
12. Start up the CP 581 system software.

Note

When setting up for the first time, you require either the mass storage module, or the CPLINK program and onboard silicon disk or memory card, for installation of the system software. When further setting up, you can operate the basic board on its own.

2.5 Switch and Jumper Settings on the CP 581 Components

The coding switches and plug-in jumpers are factory-set on the basic board, mass storage module and AT slot module. You need not carry out any checks or adjustments provided you wish to use the CP 581 components in their default configuration.

Note

You can refer to the functions of the coding switches and plug-in jumpers in the Reference Section for Hardware:
Section 3.1.3 for the basic board,
Section 3.2.5 for the mass storage module and
Section 3.3.2 for the AT slot module.

2.6 Installing the Memory Modules and the Onboard Silicon Disk into the CP 581 Pentium



Caution

It is essential to observe the guidelines for handling electrostatically sensitive devices (ESD guidelines - see Appendix) when assembling the individual components.

2.6.1 Installing the Memory Modules into the CP 581 Pentium

The DRAM memory modules **must always** be installed in pairs. When doing so, be sure both memory modules are the same type (check the order number). Use only the DRAM modules that are listed in Chapter 5, Ordering Information.

Combining memory modules of different organizations, access rates or module types (EDO or Fast Page Mode) **is not permitted**.

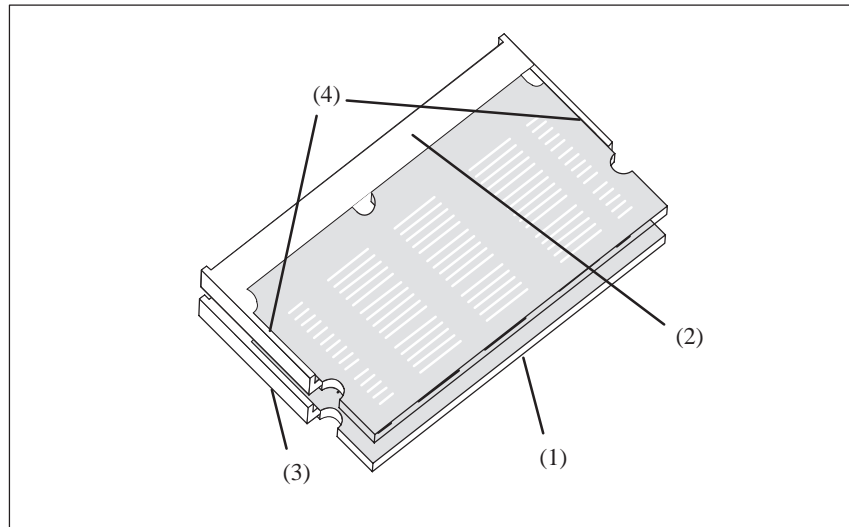
For the CP 581 Pentium, the memory modules are installed into a multipurpose DRAM socket (X204). Both modules are then located, one under the other, in this socket.

To install the modules, proceed as follows (also refer to the following figure):

1. Feed the first memory module at an angle into the lower socket (1) and press it gently down into a horizontal position. When the module is correctly positioned, the lower latches (3) on the left and right of the socket have to click into place.
2. Feed the second memory module at an angle into the upper socket (2) above the first memory module and press it gently down into a horizontal position. When the module is correctly positioned, the upper latches (4) on the left and right of the socket have to click into place.

Note

Do not use force when pushing the memory modules into their sockets. If there is too much resistance when pressing the memory modules into place, check their positions in the sockets.



2.6.2 Installing the Onboard Silicon Disk into the CP 581 Pentium

Use only the onboard silicon disk that is listed in Chapter 5, Ordering Information.

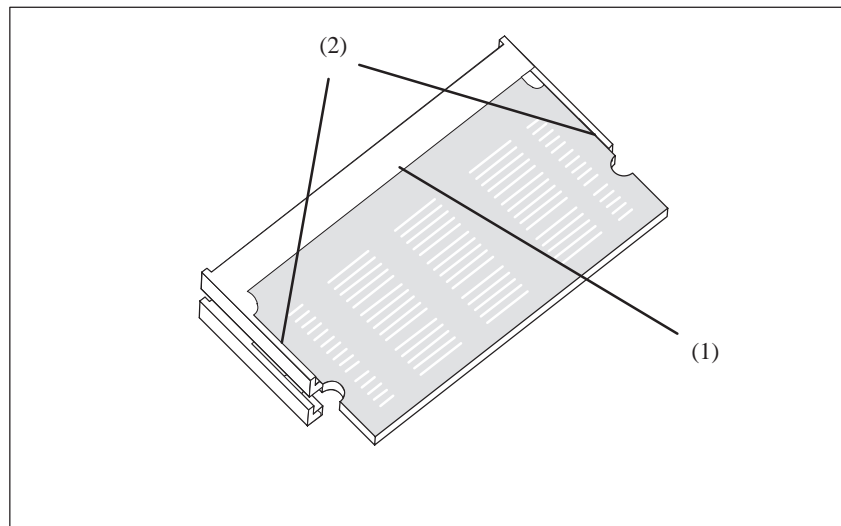
For the CP 581 Pentium, the onboard silicon disk is installed into a multipurpose DRAM socket (X252). You may **only use the upper socket** for installing the onboard silicon disk.

To install the disk, proceed as follows (also refer to the following figure):

1. Feed the onboard silicon disk at an angle into the upper socket (1) and press it gently down into a horizontal position. When the disk is correctly positioned, the upper latches (2) on the left and right of the socket have to click into place.

Note

Do not use force when pushing the onboard silicon disk into its socket. If there is too much resistance when pressing the onboard silicon disk into place, check its position in the socket.



2.7 Assembly of Individual Components into Complete Module



Caution

It is essential to observe the guidelines for handling electrostatically sensitive devices (ESD guidelines - see Appendix) when assembling the individual components.

2.7.1 Information on Assembly of Individual Components

The following sections describe the assembly of all possible combinations of the individual components:

- Mass storage module on basic board,
- 1 AT slot module on basic board,
- 2 AT slot modules on basic board,
- 1 AT slot module on mass storage module, and
- 2 AT slot modules on mass storage module.

You are additionally informed of how you can equip an AT slot module with an AT board.

The mass storage module and the AT slot modules each have a set of accessories. These are indicated in the parts lists (Figures 2-5 and 2-7). Each part in these figures is assigned a number which is referred to in the following assembly diagrams.

If you are looking for a part during assembly, you can find it in the corresponding parts list using its number.

Plugging Together of Modules

The assembly instructions for the various combinations include a step for connecting the modules by tongue and groove. Figure 2-4 shows you how the modules are connected in this manner.

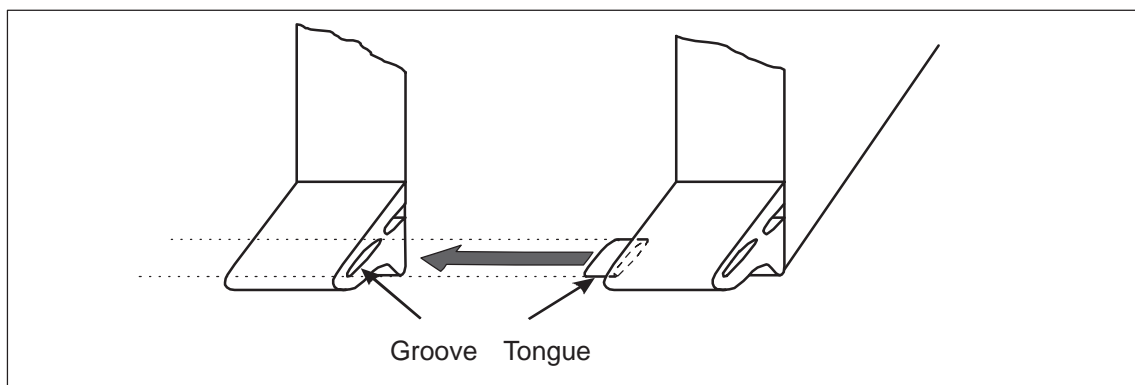


Figure 2-4 Connection of Modules Using Tongue and Groove

2.7.2 Assembly of Mass Storage Module on Basic Board

Assembly Accessories for the Mass Storage Module

An assembly kit C79458-L965-D11 is supplied with the mass storage module. This set contains all screws and spacers required for assembling the basic board as well as a TORX offset screwdriver (see Figure 2-5).

The numbers of the individual parts are used in the following assembly diagram.






Assembly kit C79458-L965-D11 for assembly of mass storage module			
	Part No.	Quantity	Designation
	1	3	Spacer, 12.7 mm
	2	2	Spacer, 18.7 mm
	3	4	Screw, 6 mm long
	4	6	Screw, 12 mm long
		1	Torx offset screwdriver, M 2.5

Figure 2-5 Assembly accessories for the mass storage module

Assembly Procedure

Proceed as follows according to Figure 2-6 in order to assemble the modules:

1. Screw the spacers “1” and “2” onto the basic board according to Figure 2-6. Insert the screws “3” and “4” from the solder side (the side of the module provided with a cover).
2. Plug the mass storage unit onto the basic board, ensuring that the tongue on the eject lever of the mass storage module latches into the groove in the eject lever of the basic board (see Figure 2-4). The plugs of the two modules must be completely connected together.
3. Screw the two modules together as in Figure 2-6. If you wish to fit the AT slot module immediately, **do not tighten** the screws identified “A.”

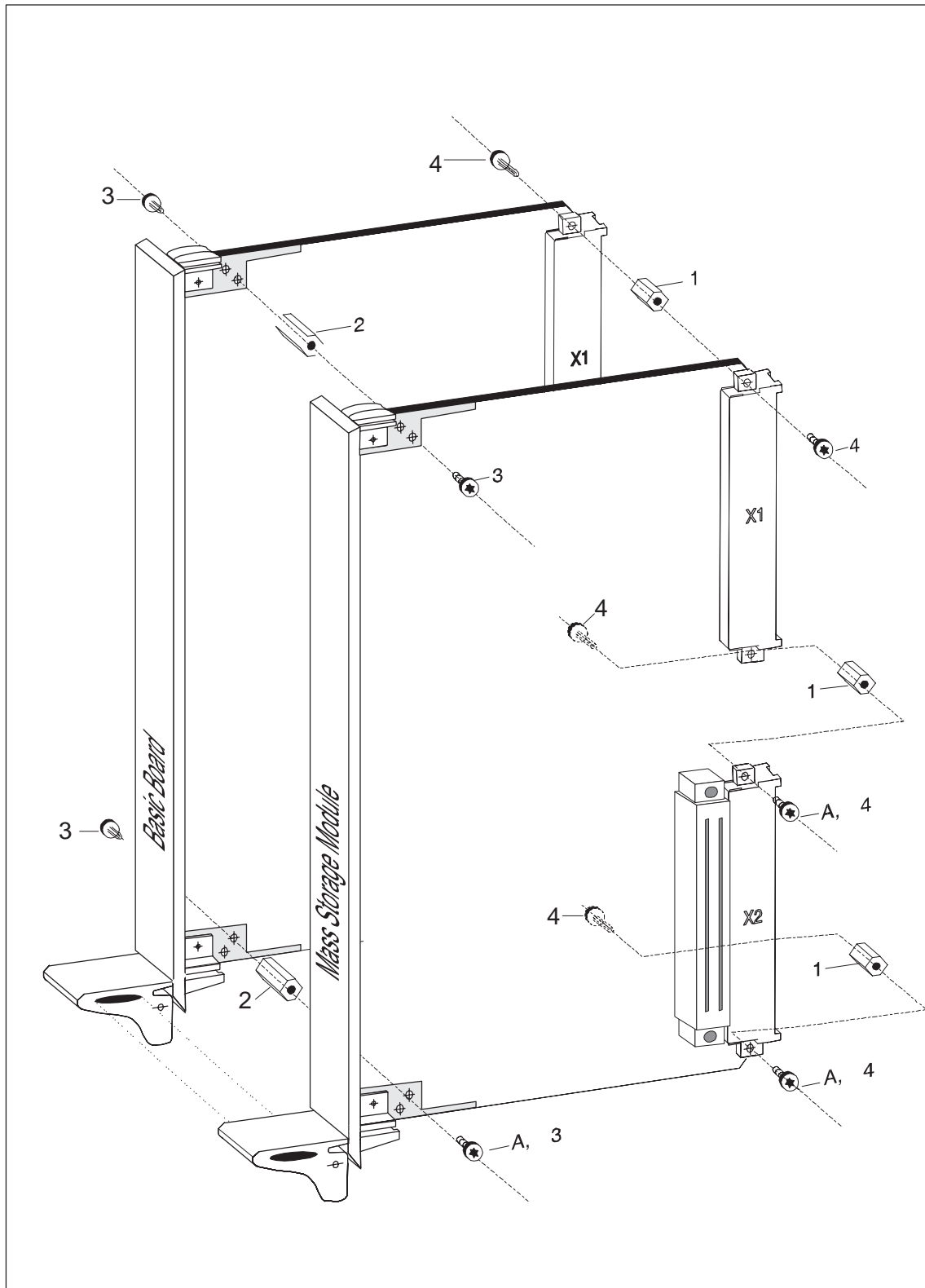


Figure 2-6 Assembly of the Mass Storage Module onto the Basic Board

2.7.3 Assembly of AT Slot Modules

Assembly Accessories for the AT Slot Modules

An assembly kit C79458-L965-D12 is supplied with each AT slot module. This set contains all screws and spacers required for assembling the basic board, mass storage module or AT slot module as well as a TORX offset screwdriver (see Figure 2-7).

The numbers of the individual parts are used in the following assembly diagrams. Not all parts of the assembly kit are required depending on the configuration of the modules.










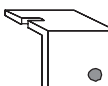
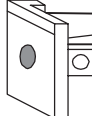
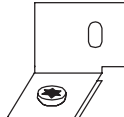

Assembly kit C79458-L965-D12 for assembly of AT slot module			
	Part No.	Quantity	Designation
	1	2	Spacer, 12.7 mm
	2	1	Spacer, 18.7 mm
	3	1	Spacer, 15.7 mm
	4	4	Spacer, 18.7 mm
	5	1	Spacer bolt, 17.7 mm
	6	2	Spacer bolt, 12.7 mm
	7	1	Spacer bolt, 17.7 mm
	8	8	Screw, 6 mm long
	9	2	Screw, 12 mm long
	10	1	Clamp
	11	1	Front panel, top
	12	1	Support
	12	1	Torx offset screwdriver, M 2.5

Figure 2-7 Assembly Accessories for the AT Slot Modules

**Assembly of an
AT Slot Module
onto the Basic
Board**

Proceed as follows according to Figure 2-8 in order to assemble the modules:

1. Screw the spacer “2” and the spacer bolt “5” onto the basic board according to Figure 2-8. Insert the screw “8” from the solder side (the side of the module provided with a cover). Make sure that you use the inner hole of the two adjacent holes.
2. If you wish to fit a second AT slot module immediately, screw the spacer identified “B” onto the first AT slot module.
3. Plug the AT slot module onto the basic board, ensuring that the tongue on the eject lever of the AT slot module latches into the groove in the eject lever of the mass storage module (see Figure 2-4). The plugs of the two modules must be completely connected together.
4. Attach clamp “10” onto the mounted system such that the right-hand edge of the basic board front panel extends into the slit of the clamp, and the clamp rests on the component side of the AT slot module. Screw the clamp onto the module if you do not wish to fit a second AT slot module immediately.
5. If you do not wish to fit a further AT slot module, screw the two modules together using the three screws identified “A” in Figure 2-9 and fit the part “Front panel, top” (“11”) onto the spacer bolt “5.”

If you wish to fit a second AT slot module immediately, proceed as described in the section “Assembly of two AT slot modules onto the basic board.”

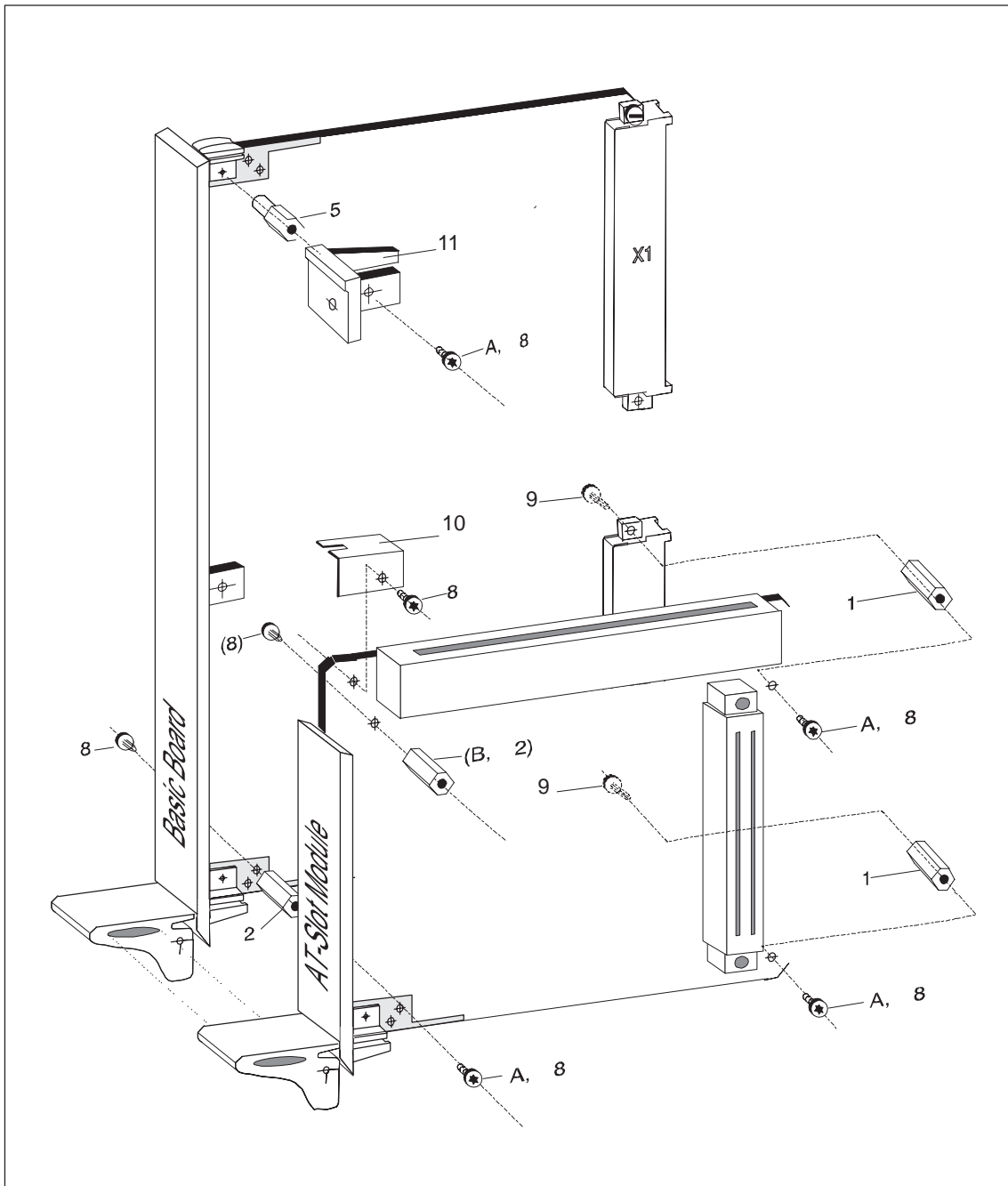


Figure 2-8 Assembly of an AT Slot Module onto the Basic Board

**Assembly of Two
AT Slot Modules
onto the Basic
Board**

Proceed as follows according to Figure 2-9 in order to assemble the second AT slot module:

1. If the first AT slot module has already been fitted completely:
Remove the four screws identified "A." The screws are still required for the further assembly (identified by "A" in Figure 2-9). Unscrew the clamp "B" and remove the first AT slot module from the basic board.
2. Screw spacer "C" onto the first AT slot module, connect the basic board and the first AT slot module together and screw on clamp "B."
As illustrated in Figure 2-9, screw spacer bolt '4' into the first AT slot module.
3. Connect the second AT slot module to the first AT slot module, making sure that the tongue at the eject lever of the second AT slot module latches into the groove in the eject lever of the first AT slot module (see Figure 2-4). The plugs of the two modules must be completely connected together.
4. Screw the two AT slot modules together as in Figure 2-9 using the three screws identified "A" and one screw identified "8."
5. Fit the first part, "Front panel, top," onto the spacer bolts identified by "5" in Figure 2-9; tighten the part using spacer bolt "7" (Figure 2-9). Using the fourth screw "4", subsequently fit the second part "Front panel, top" onto the spacer bolt "7" screwed in for securing the first part.

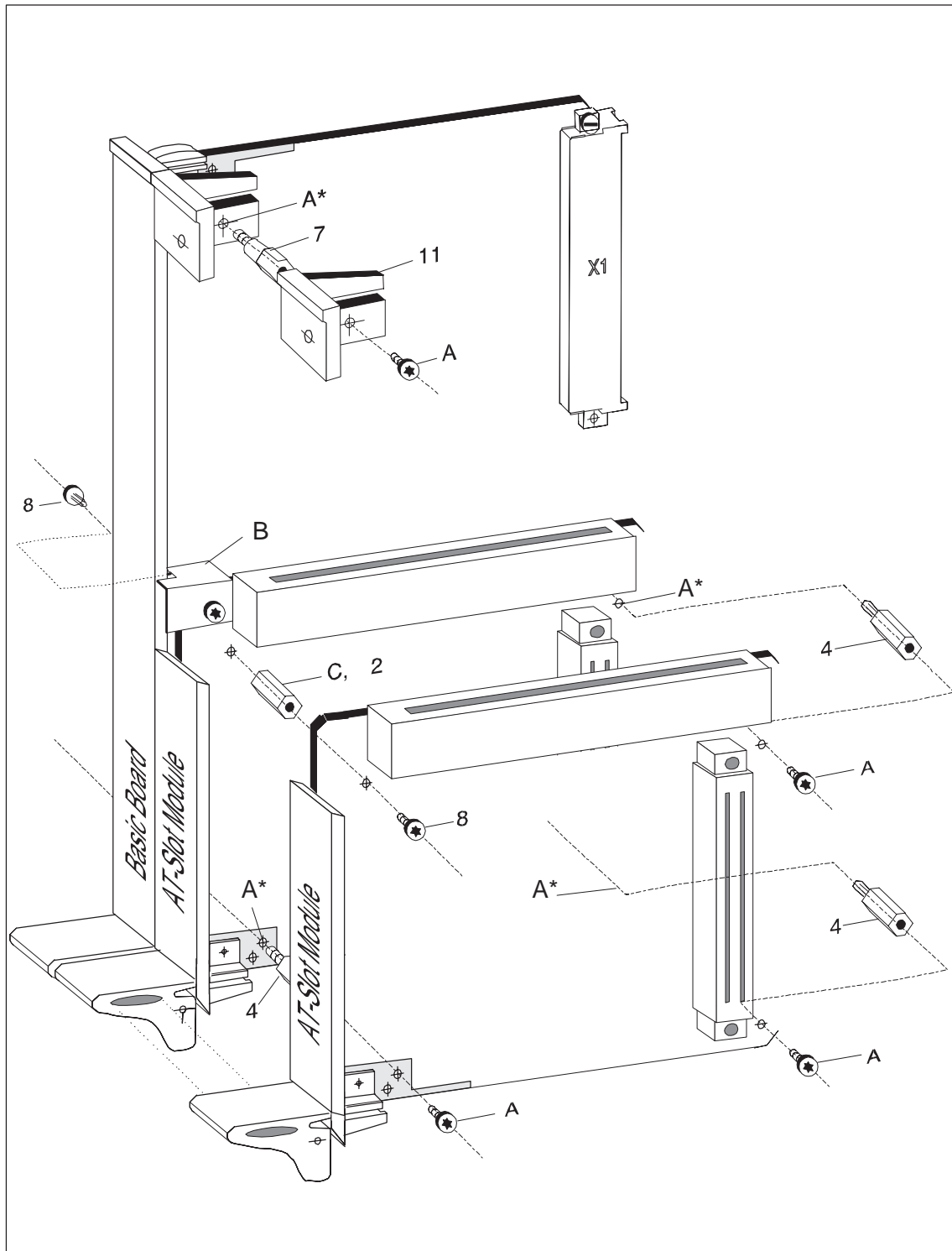


Figure 2-9 Assembly of Two AT Slot Modules onto the Basic Board

**Assembly of an
AT Slot Module
onto the Mass
Storage Module**

Proceed as follows according to Figure 2-10 in order to assemble the modules:

1. If the mass storage module has already been fitted completely:
Loosen the two screws identified “A*” and “B*”. Screw “A*” is required for the further assembly (identified by “A” in Figure 2-10), screws “B*” are replaced by the screws identified by “A, 8” in Figure 2-10.
2. Screw in the spacer bolts “3,” “4” and “6” according to Figure 2-10.
3. Plug the AT slot module onto the mass storage module, ensuring that the tongue on the eject lever of the AT slot module latches into the groove in the eject lever of the mass storage module (see Figure 2-4). The plugs of the two modules must be completely connected together.
4. If you do not wish to fit a further AT slot module, screw the two modules together using the screw identified “A” and the three screws identified “A, 8” in Figure 2-10 and fit the part “Front panel, top” (“11”) onto the spacer bolt “5” using a screw “A, 8.”
If you wish to fit a second AT slot module immediately, proceed as described in Section 6.4.5.

Note

If the spacer bolts “6” in Figure 2-10 do not immediately grip when screwing in, slightly loosen the screws identified “4” in Figure 2-6 (1/4 revolution), screw in the bolts and tighten the screws again.

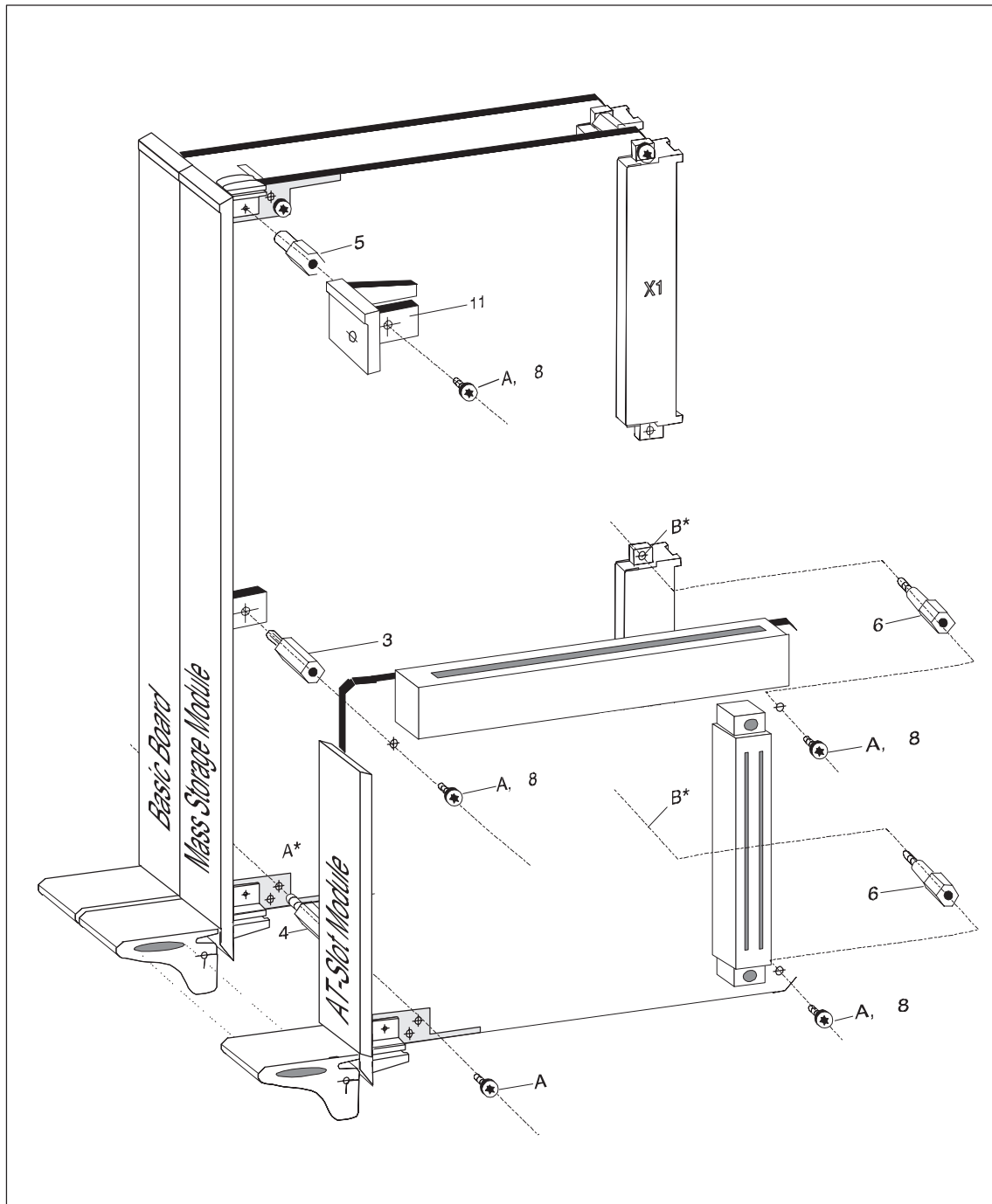


Figure 2-10 Assembly of an AT Slot Module onto the Mass Storage Module

**Assembly of Two
AT Slot Modules
onto the Mass
Storage Module**

Proceed as follows according to Figure 2-11 in order to assemble the second AT slot module:

1. If the first AT slot module has already been fitted completely:
Remove the five screws identified "A*". The screws are required for the further assembly (identified "A" in Figure 2-11).
2. Screw in the spacer bolts as in Figure 2-11.
3. Plug the second AT slot module onto the first AT slot module, ensuring that the tongue on the eject lever of the second AT slot module latches into the groove in the eject lever of the first AT slot module (see Figure 2-4). The plugs of the two modules must be completely connected together.
4. Screw in the two AT slot modules together as in Figure 2-11 using the four screws identified "A."
5. Fit the first part, "Front panel, top," onto the spacer bolts identified by "5" in Figure 2-10; tighten the part using spacer bolt "7" (Figure 2-11). Using the fifth screw "4", subsequently fit the second part "Front panel, top" onto the spacer bolt "7" screwed in for securing the first part.

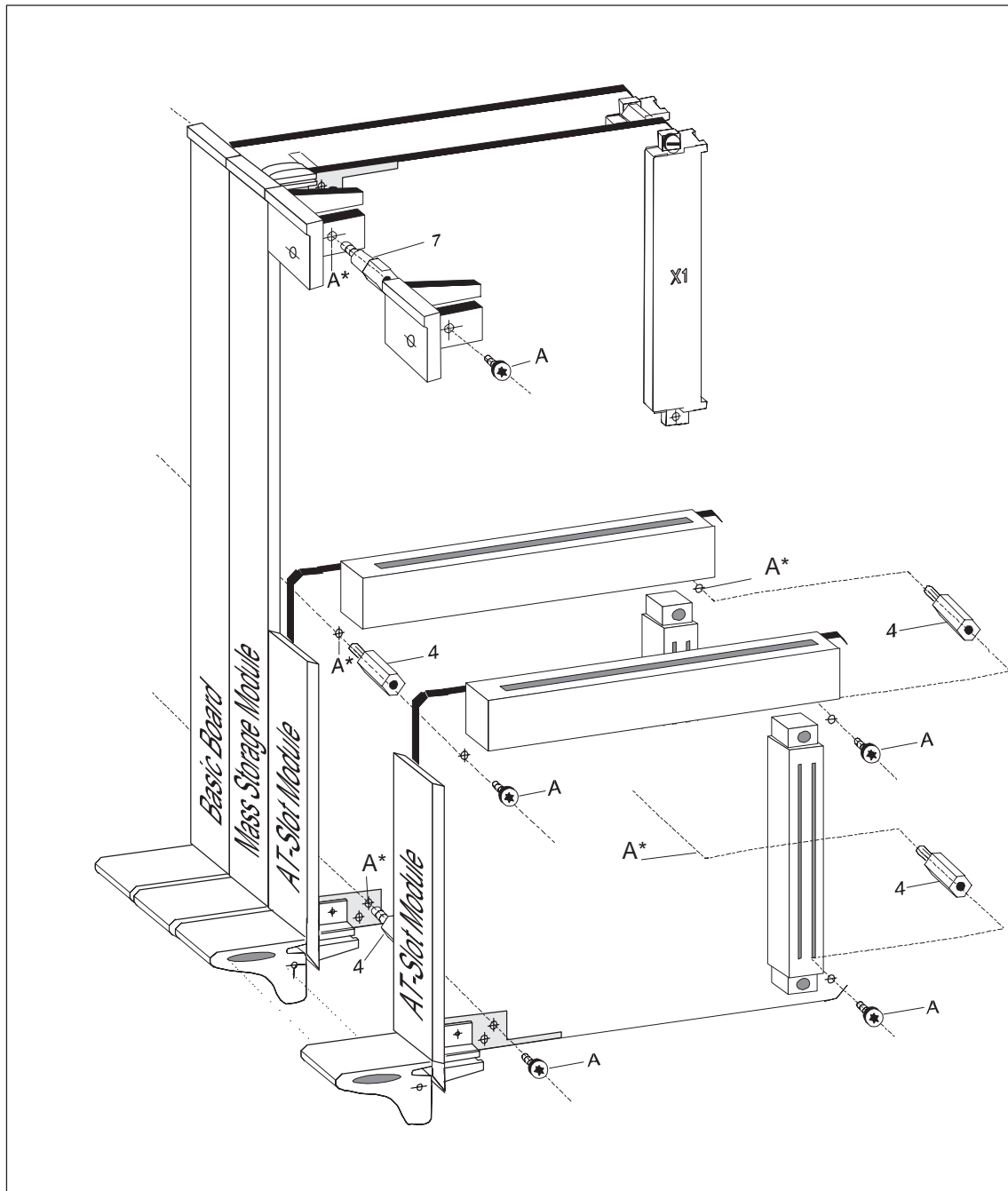


Figure 2-11 Assembly of Two AT Slot Modules onto the Mass Storage Module

2.7.4 Assembly of an AT Board onto an AT Slot Module

You require the part Nos. 8 (screw, 8 mm long) and 12 (support) from the assembly kit C79458-L965-D12 (see Figure 2-7) to fit an AT board onto an AT slot module.

Proceed as follows according to Figure 2-12 when fitting:

1. Insert the AT board from above into the female connector of the AT slot module as shown in Figure 2-12.
2. Insert the support “12” above the angled bracket on the AT board; the screw of the support projects into the groove of the bracket. Tighten the screw of the support.
3. As shown in Figure 2-12, insert screw “8” through the support into part “11” which you fitted with the AT slot module, and tighten the screw.

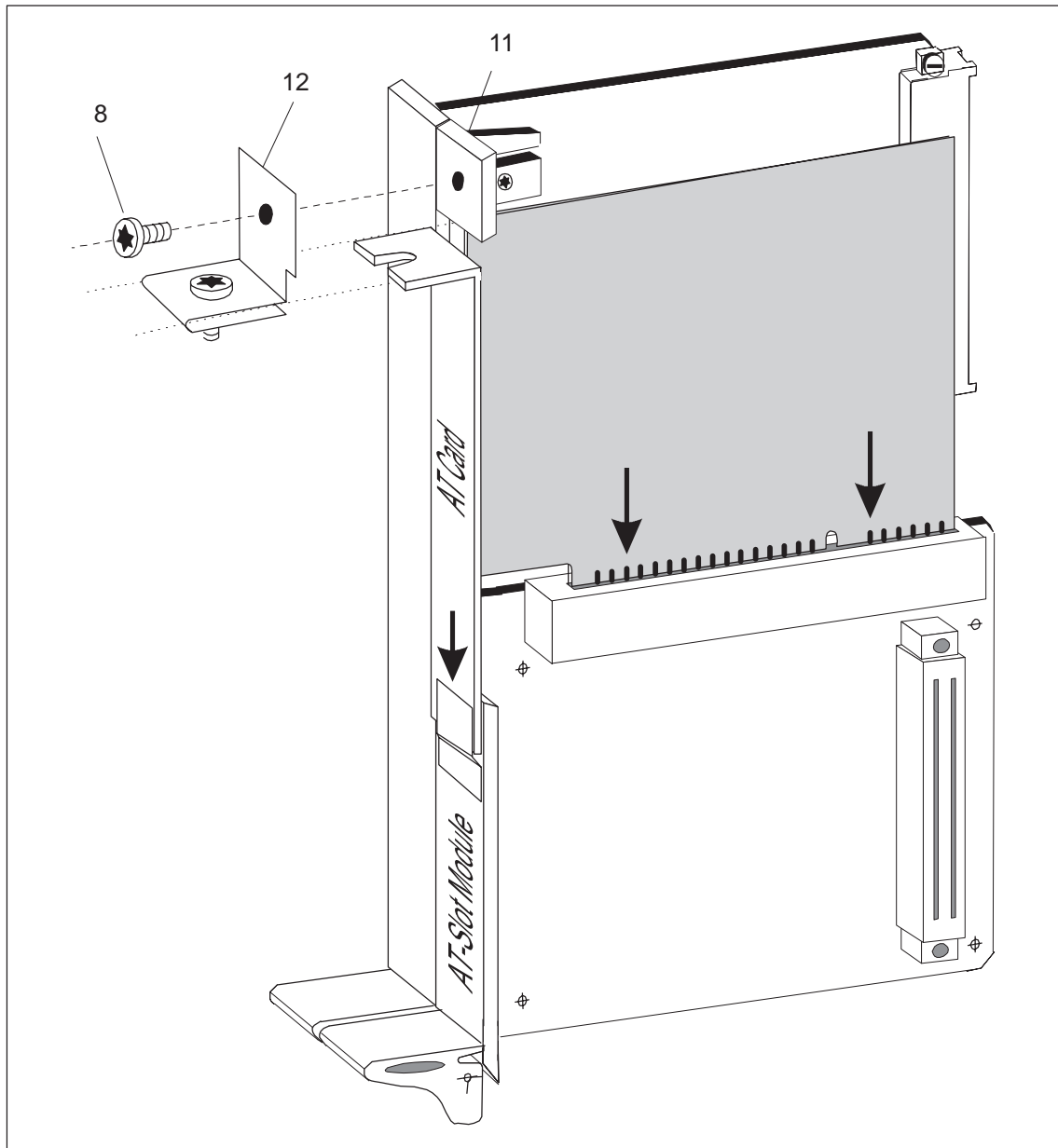


Figure 2-12 Assembly of an AT Board onto an AT Slot Module

2.8 Installation of Complete Module into Subrack

2.8.1 Interference-Free Hardware Configuration

Note

To ensure interference-free operation, observe the shielding measures and installation guidelines included in the respective programmable controller manuals (for example, S5-135U).

Keep any interfering signals resulting from the process as far away as possible from the installation.

We assume you have observed the “SIMATIC S5 Installation Guide.” The following pages summarize some of the important points of these guidelines.

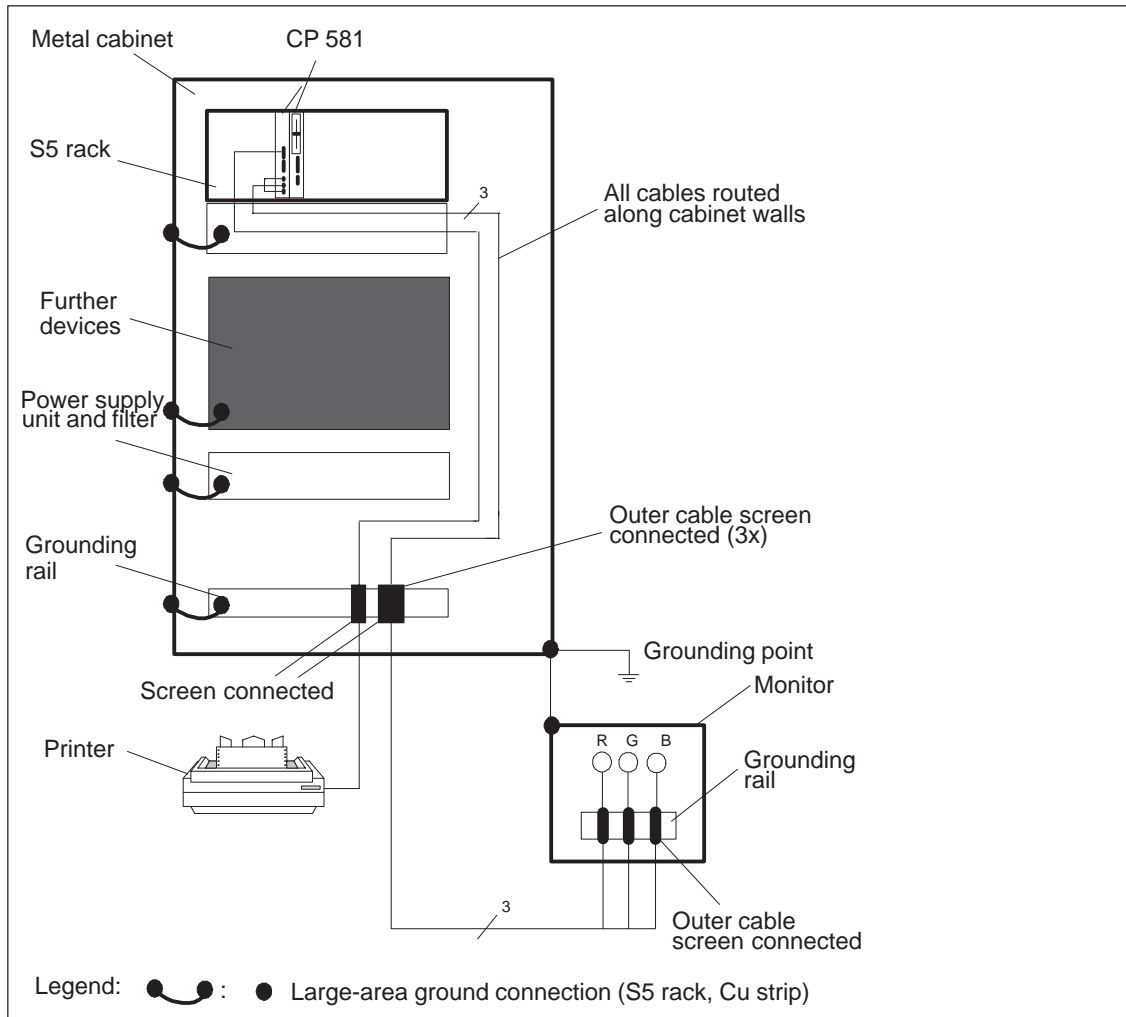


Figure 2-13 Cabinet Design

The following can be seen in Figure 2-13:

- Devices which could carry noise signals from outside into the cabinet assembly should be fitted as near to the bottom of the cabinet as possible.
- Fit the grounding rails directly at the cabinet inlet so that the cables carrying noise signals (for example, monitor cables and the power supply cable for the printer) can be connected directly there. Connect all cables screened to this point (except coaxial cables with one screen). Only connect the outer screen in the case of the signal cables with two screens.
- Always route signal cables along the cabinet walls.
- Route power supply and signal cables separately.
- Use separate cable racks for power and signal cables and position them at least 0.5 m apart.
- Ensure that all ground connections in the cabinet are made with a large area contact.
- Connect doors and cabinet walls to the grounded housing support.
- Ensure when fitting a SIMATIC system in a cabinet that the cabinet is grounded.
- Relays and contactors should be interference-suppressed on site.
- Ensure that the differences in potential between various system components are as small as possible.
- In the case of systems which generate a high electrostatic voltage (for example, textile machines, special napping machines), connect the ground lines of the machine components subject to interfering signals to a separate signal ground (large area ground contact with building construction, armoring) which is isolated from the central grounding point of the S5 cabinet.

2.8.2 Selection of CP 581 Slots in the Programmable Controller

You can use the CP 581 in the S5-115U, S5-135U, S5-155U, S5-135U/155U and S5-155H programmable controllers.

- S5-115U:

Observe the following requirements for installation:

- You require an adapter casing with 2 or 4 slots (see Ordering Information, Chapter 5).
- Use a 5 V/15 A power supply.

Locations for the approved central controllers:

Central controller	possible locations
6ES5 700-0LB11 (CR 700-0LB)	0
6ES5 700-3LA12 (CR 700-3)	0, 1, 2

- S5-135U:

Note that you cannot use the CP 581 together with the CPU 921 (S processor).

Slots for the approved central controllers (CC) 6ES5 135-....:

Slot No. CC	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
-3KA13			■	■	■	■	■	■	■													
-3KA21			■	■	■	■	■	■	■													
-3KA31			■	■	■	■	■	■	■													
-3KA41			■	■	■	■	■	■	■													
-3UA11			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA21			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA31			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA41			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA51			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					

The PG-MUX and interrupt generation functions cannot be used at a slot occupied by a CP 581 component.

- S5-135U/155U

Slots for the approved central controllers (CC) 6ES5 188-....:

Slot No. CC	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
-3UA12			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA22			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA32			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
-3UA52			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					

- S5-155U:

Slots for the approved central controllers (CC) 6ES5 155-....:

Slot No. CC	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
-3UA11			■		■	■	■	■		■	■	■	■	■	■	■	■					
-3UA21			■		■	■	■	■		■	■	■	■	■	■	■	■					

- Expansion unit EG S5-185U

Slots for the approved expansion units (EU) 6ES5 185-....:

Slot No. EU	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
-3UA13			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
-3UA23			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
-3UA33			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
-3UA43			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				

(Each S5 185U expansion unit with the IM 304/314 or 307/317 interface modules)

2.8.3 Switching Off the Power Supply of the PLC Rack



Caution

The CP 581 must not be inserted or removed with the power supply switched on.

It is therefore essential for you to switch off the power supply for the PLC rack before inserting the CP 581 into the programmable controller.

When switching the power supply in the CC or the EU off via the enable input, it is possible that write accesses to the hard disk could lead to a loss of data.

We recommend:

Only switch the CC/EU off via the enable input when the hard disk drive, onboard silicon disk, memory card or floppy disk is not being accessed.

2.8.4 Installation of CP 581 into PLC Rack



Caution

When working on the system with the cabinet open, observe the general safety regulations (for example, VDE 100) as well as the guidelines for protective measures for electrostatically sensitive devices (ESD).

- Now insert the CP 581 into your PLC rack. Observe the permissible slots of the various programmable controllers (Section 2.8.2).
- Be careful to hold the module straight and not to bend the contact springs of the guides in the subrack.
- Lock the CP 581 in the subrack.

2.9 Connection of Operation Devices and Peripheral Devices

The connections for the operation devices and peripheral devices are on the front panel of the basic board and mass storage module.

Figures 2-1 and 2-3 show which devices can be connected, and where.

You can find comprehensive information on **all** connection possibilities of the CP 581 in the corresponding sections in the Reference Section for Hardware.

A keyboard and monitor are required to use the CP 581.

You can additionally connect a printer and a mouse.

When connecting the peripheral devices, we recommend that you use the standard cable connectors provided by Siemens for reasons of interference resistance within the complete system.

You require the Y adapters listed in Chapter 5, Ordering Information, in order to use the standard cable connectors.



Caution

Important note when routing connection cables for peripheral devices: Monitor cables and connection cables between the CP 581 and the keyboard/printer/mouse must not be routed parallel to power cables! Route them on their own cable rack positioned at least 50 cm away from the power cables.

2.9.1 Connection of Keyboard

- Connection via the Y adapter Keyboard/COM 2

Connect the keyboard to the 7-pin round socket of the Y adapter.
(If necessary, use a Din → Mini-Din adapter.)

- Connection via the remote terminal interface

Connect the keyboard to the 7-pin round socket X2 of the RTI module.

2.9.2 Connection of Monitor

RGB Interface

Note the following points:

- You must only use double-screened coaxial cables (triax cables) if the environment is not conducive to EMC (see Chapter 5, Ordering Information). You can use these cables up to a length of 250 m without further measures.
- When using triax cables, only connect the outer screen of the monitor cable to the housing potential of the process monitor (see Figure 2-14).
- Make large area metal-metal connections.
- Isolate the electronics ground of the monitor (see Figure 2-14) from the housing ground. The separation has already been made in the Siemens multistandard color monitor 6AV1 414-0AA00 listed in Chapter 6.
- Connect the monitor and the programmable controller to the same phase.
- Different ground potentials between the rack and the monitor housing may lead to picture hum (horizontal dark bars). For this reason and for protection against accidental contact, connect an equipotential bonding conductor between the S5 cabinet and the monitor housing. The equipotential bonding conductor must be dimensioned by carrying out measurements on the system such that a difference in potential of 200 mV is not exceeded.
- The terminating resistor for the monitor must be 75 Ohms.

VGA Interface

Note the following points:

- The line length may not be longer than 1.5 m.
- Connect the monitor and the programmable controller to the same phase.
- Different ground potentials between the rack and the monitor housing may lead to picture hum (horizontal dark bars). For this reason and for protection against accidental contact, connect an equipotential bonding conductor between the S5 cabinet and the monitor housing. The equipotential bonding conductor must be dimensioned by carrying out measurements on the system such that a difference in potential of 200 mV is not exceeded.
- The terminating resistor for the monitor must be 75 Ohms.

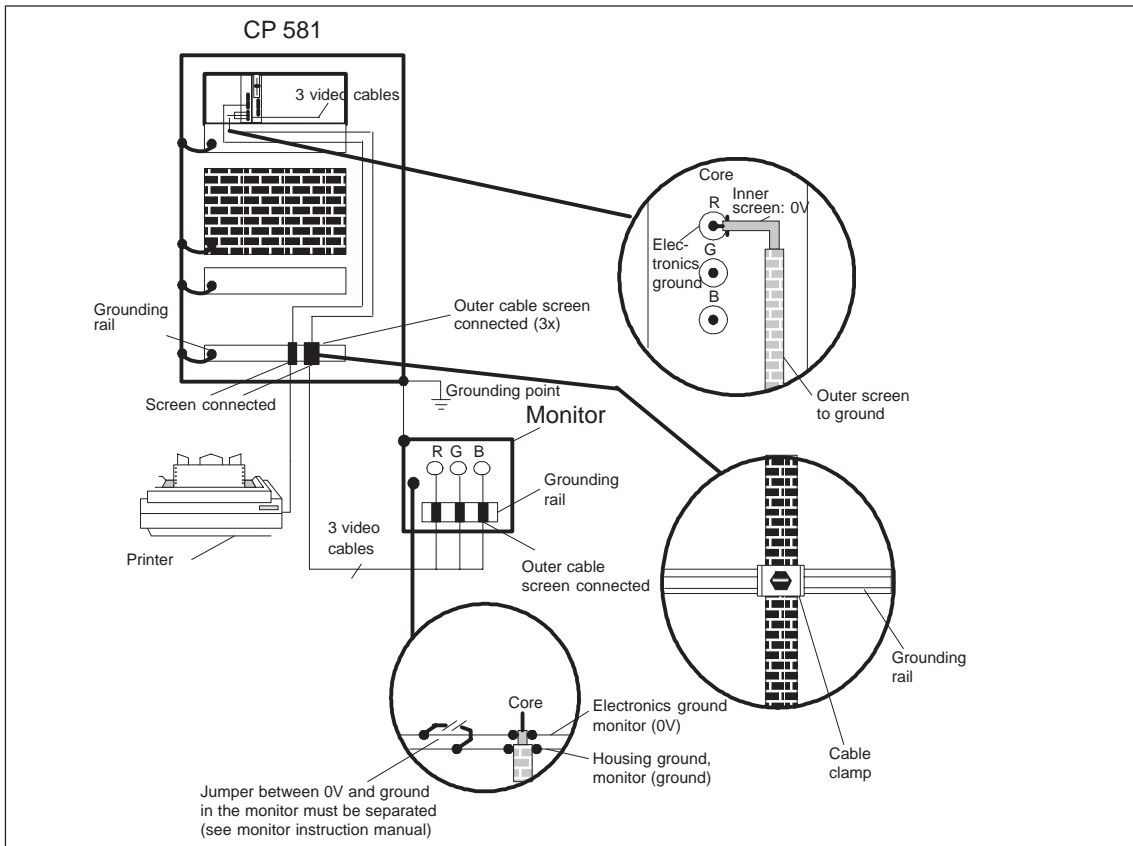


Figure 2-14 Cable Connection to Grounding Rails

Information on Positioning Monitors

- When positioning monitors ensure that the distance between two monitors in the case of asynchronous operation is at least 15 cm since picture interferences could otherwise occur.
Exception: monitors with Mu Metal screening
- The monitor and external magnetic sources should be sufficiently far apart.
- Do not place the monitors in steel racks or on steel desks. Magnetization of the surrounding sheet steel panels may lead to color or picture distortions.
- Do not install monitors in the vicinity of transformers, walkie-talkies, loudspeaker magnets and power cables.
- External magnetic fields can be suppressed by using Mu Metal screening.

Special Conditions When Using Office Monitors

- You must not use office monitors with a plastic housing metal-coated on the inside in environments not conducive to EMC since the internal metal surface cannot be connected to the external grounding rail at a later date. The separation between the electronics ground and the housing ground of the monitors which is essential for environmental conditions not conducive to EMC is not possible with most office monitors.
- You can only use such office monitors together with single-screened coaxial cables. Connection to the CP 581 is thus only possible to a limited extent since only short distances can be covered using these cables.

2.9.3 Connection of a Printer

- Connect a printer with serial interface to COM 1/3 (basic board) or COM 4 (mass storage module) using the corresponding Y adapter (see Chapter 5, Ordering Information).
Connect a printer with parallel interface to LPT (mass storage module).
- We recommend the use of Siemens printers.
Further information on these printers such as the technical data and order nos. for accessories (printer cables, interfaces, etc.) can be found in Sections 3.1.4, 3.2.6 and 3.2.7 in the Reference Section for Hardware and in Chapter 5, Ordering Information.

Note

Only cables with the screen grounded at both ends may be used between the CP 581 and printers.

Note that the MS-DOS command PRINT cannot be used with the TTY interface.

2.9.4 Connection of a Mouse

The CP 581 software supplied does not use a mouse!

It may be useful to connect a mouse in order to use additional software on the CP 581.

Connect the mouse to the CP 581 in the following manner:

- Connection via the Y adapter Keyboard/COM 2
Connect the mouse to the 9-pin D-subminiature plug of the Y adapter.
- Connection via the remote terminal interface
Connect the mouse to the 9-pin D-subminiature plug X3 of the RTI module. The mouse cannot be connected to the PG 750 keyboard when using the RTI!

2.9.5 Maximum Cable Lengths for Connection of Operation Devices and Peripheral Devices

The following table lists the limits for the cable lengths between the devices. A prerequisite is an interference resistant hardware configuration as described in Section 2.8.1.

Table 2-1 Maximum Cable Lengths for Operation Devices and Peripheral Devices

Device	Length Configuration 1 (local)	Length Configuration 2 (remote)
Printer with V.24 interface	20 m	-
Printer with TTY interface	-	500 m
Printer with parallel interface	3 m	-
Monitor (when using triax cables)	250 m	250 m
Monitor (when using standard VGA !)	1.5 m	-
Mouse	1.5 m	250 m (with RTI)
Keyboard	1.5 m	250 m (with RTI)

2.10 Commissioning (with a Mass Storage Module)

Set the RUN/STOP switch to “RUN.”

Checklist before switching on the power supply

Before you switch on the power supply, check using the following list that all preparations have been made:

- Have you considered the environmental conditions for the CP 581 and connected peripheral devices?
- Is the CP 581 inserted into an approved slot in the subrack?
- Are the peripheral devices connected correctly?
- Have you connected all the cable screens correctly?
- Have you installed the monitor correctly and considered the electromagnetic environmental conditions?
- Have you observed the S5 installation guidelines for programmable controllers?

Switching on peripheral devices

Now switch on the monitor and the printer.

Switching on the power supply to the PLC rack

Now switch on the power supply to the programmable controller rack. The following processes are then executed on the CP 581 or the connected peripheral units:

- The CP 581 runs up.
- The STOP and FAULT LEDs light up until you have made a correct SETUP (see Section 3.7).
- The operation display on the hard disk drive on the mass storage module now lights up every time it is accessed.

Hardware commissioning of the CP 581 is now finished.

If you wish to install and commission the software, read further in Section 2.11, “Installation and Commissioning of the Software.”

2.11 Installation and Commissioning of the Software

Note

You can find the current delivery stage of the CP 580/CP 581 system software in the DIR.TXT file.

Please note that you require the mass storage module or the CPLINK program for initial commissioning.

The CP 581 in the design with a basic board and mass storage module runs up with the preset SETUP parameters. You must still set the date and time in SETUP during initial commissioning, however.

2.11.1 Installation with Mass Storage Module

You have received the MS-DOS operating system and the CP 581 system software on floppy disks. You must install both software packages on the hard disk.

Proceed as follows:

1. Set up the hard drive.
2. Install MS-DOS on the hard disk.
3. Make backup copies of the original MS-DOS and CP 581 disks.
4. Install the CP 581 system software on the hard disk.

Set up the Hard Drive

When the mass storage module is shipped, the hard drive is not partitioned or formatted. To use the hard drive, you will have to partition it and format it. To do so, you will need a bootable memory card (see Chapter) or floppy disk. Copy the MS-DOS programs FORMAT and FDISK onto either of these to bootable media. Then, using the booting medium, boot the CP 581. You can now use the MS-DOS program, FDISK, to partition and the hard drive and then FORMAT to format it. For more information, refer to your MS-DOS manual.

Installation of MS-DOS

Refer to your MS-DOS manual for information on how to install MS-DOS.

For the CP 581 DX:

Driver **EMM386.EXE**: If you select the function “Upper memory blocks,” you must enter the option “**X=CC00-DFFF**” (dual port RAM and flash page) or “**X=CC00-CFFF**” if you do not use a silicon disk.

For the CP 581 Pentium:

Driver **EMM386.EXE**: If you select the function “Upper memory blocks,” you must enter the option “**X=CA00-CFFF**” (dual-port RAM and flash page) or “**X=CC00-CFFF**” if you do not use a silicon disk.

Making Backup Floppy Disks

Use the MS-DOS command **DISKCOPY** to make backup disks. This function physically copies a source disk onto the target disk. The backup disks need therefore not be formatted. They must be the same type as the supplied disks, however, i.e. 3.5" HD disks.

Since the CP 581 only has **one** floppy disk drive, you must enter the same drive name for the source and target in the start command for **DISKCOPY**.

Proceed as follows:

1. Start the function using the command **C:DISKCOPY A: A:**
DISKCOPY requests you to alternately insert the source and target disks into the drive.
2. Insert the requested disk into the drive, close the drive and acknowledge the request by pressing any key.
3. Carry out the operations for every supplied disk until all disks have been copied.

Installing the CP 581 System Software

Insert the supplied disk with the CP 581 system software into the drive and enter the command

A:INSTALL C:

When you install the system software (using the **INSTALL** command), ensure that you choose the correct variation of the CP581.

The **INSTALL** program now copies the CP 581 system software from the floppy disk onto the hard disk. **INSTALL** will inform you on the screen of any operations you need to make.

Setting the Hardware Clock of the CP 581 in the SETUP

The date and time are set on the hardware clock of the CP581 using **SETUP**.

MS-DOS Functions



Caution

Do not use a write cache with the CP 581 system software as you can with SMARTDrive in Windows 3.1.



Caution

Ensure that no data traffic is present on the S5 bus when you use the MS-DOS command **FORMAT**. The **PRINT** and **COPY** commands prevent communication with **CPMASS** and **CPRECORD** and should therefore be used with caution.

Initial CP 581 Startup

The AUTOEXE.BAT file is optionally initialized by INSTALL such that the CPDHB driver, the mass storage functions and the command interpreter are available when the CP has been run up correctly.

The RUN, STOP and FAULT displays are set by the CPDHB driver according to the current operating state when the driver is called.

The RUN display only lights up if the RUN/STOP switch is set to RUN and at least one page is synchronized.

Activation of Supplied Applications

To activate and test CPRECORD, CPMASS, CPSHELL and S5REMOTE/S5REMOTF, please read the following chapters in Part 2 of the manual:

- “Process Data Acquisition” (CPRECORD),
- “Mass Storage Functions” (CPMASS),
- “Command Interpreter” (CPSHELL), and
- “Virtual S5 drive” (S5REMOTE).

Normal Restart of CP 581

A normal restart is always carried out when you switch off the operating voltage to your PLC and then on again, if the CP 581 is ready, and if the device configuration preset in the software has been retained in the RAM as a result of the battery backup.

Activation of Serial Interface for the Printer

Remove the comment identification “REM” from the line in the AUTOEXE.BAT file with the entry “EM MODE LPT1:=COM1”.

Note

The installation of the system software for the CP 581 with a basic board and mass storage module has now been completed.

Refer to the following Section 2.11.2 for the operations you must carry out for individual operation of a basic board. Section 3.7 provides you with a summary of all possible SETUP settings.

2.11.2 SETUP for Individual Operation of the CP 581 Basic Board with Silicon Disk

You can operate a CP 581 basic board with silicon disk even without a mass storage module. The MS-DOS operating system can be booted both from the onboard silicon disk and from the memory card. These two storage media are addressed by the user program like conventional drives.

The logical drive numbers when operating the basic board without a mass storage module are then as follows:

Drive designation	Data medium
a:	Memory card of basic board
c:	Onboard silicon disk of basic board

When operating the CP 581 without a mass storage module, **no conventional** drives may be specified in the SETUP menu.

Select Memcard or Memory Card as drive A.

Select the drive to be used for booting from the **“Boot Features”** or **“Boot Sequence”** menu. You have the following possibilities:

- If you are booting from the memory card, enter “A,C” or “A: then C:”.
- If you are booting from the onboard silicon disk, enter “C,A” or “C: then A:”.

In a CP 581 without a mass storage unit, the default setting in the BIOS setup is such that the CP is booted from the memory card. If no bootable memory card is present, the CP is booted from the onboard silicon disk (providing this is bootable).

Booting from Memory Card or Onboard Silicon Disk

You must first format the storage medium from which you wish to boot. The necessary operations differ depending on the configuration of your system.

If a hard disk is present, this is automatically designated as drive C:. If an onboard silicon disk is present, this is designated as drive D:. You do not need to make any changes in the SETUP.

If a hard disk is not present, the onboard silicon disk is designated as drive C:. This occurs automatically. You do not need to make any changes in the SETUP.

**Initial status:
basic board and mass storage module present**

- Initializing a bootable memory card:

Note

When running up the CP 581, the memory card that is to be formatted must be inserted.

- Enter the memory card as “Drive B:” in SETUP and boot from the hard disk (Drive C).

- Format the memory card using the MS-DOS command **FORMAT**:

FORMAT B: /U /S

- Format the memory card using:

FORMAT B: /S

- Transfer the software which you require (MS-DOS, system software for CP 581 etc.) onto the memory card.

- Enter the memory card as “Drive A:” in SETUP (using the cursor keys and the Tab key). Click on “OK.”

- In the “**Boot Features**” or “Boot Sequence” menu, enter “A,C” or “A: then C:”. Click on “OK.”

- Carry out a warm restart: booting is now carried out from the memory card. You can now remove the mass storage module.

- Initializing a bootable onboard silicon disk (OSD):

- Generate a bootable system floppy (FORMAT command with option “/S”) or memory card according to the SETUP entry “Drive A.”

- Copy the MS-DOS programs FORMAT and FDISK onto the floppy disk or memory card.

- Delete the hard disk from the “AT HARD DISK” or “MAIN >PRIMARY MASTER” menu in SETUP (see Section 3.7). In the “Boot Features” SETUP menu, enter A, C or “A: then C:” as the “Boot Sequence” (see Section 3.7).

Note

Another possibility would be to remove the mass storage module. You can only do this, however, if you have a bootable memory card.

- Initialize a “Primary DOS partition” on the onboard silicon disk using the MS-DOS command **FDISK**: (MS-DOS will only allow you to do this on a hard disk with the drive designation C:. This is why the hard disk had to be removed from the mass storage module.)

After the command **FDISK** has been entered, a menu with 5 possible selections appears. “1” appears as the current hard disk above the menu.

Select the option “1” by pressing the Return key.

A submenu appears with 3 possible selections. Select the default setting (option “1”) by pressing the Return key.

Answer the following question “Should the maximum available?” with “Y.”

Press the ESC key twice in order to leave **FDISK**. The operating system is subsequently rebooted.

- Format the onboard silicon disk using the **FORMAT** command:

FORMAT C: /U /S

- Transfer the software which you require (MS-DOS, system software for CP 581 etc.) onto the onboard silicon disk.
- Now activate the previously initialized partition on the onboard silicon disk using the **FDISK** command:

Select the function “2” (enter number and press Return key) in the **FDISK** menu.

Enter “1” as the active partition (enter number and press Return key).

Press the ESC key twice to leave **FDISK**.

- In **SETUP**, now enter “C,A” or “C: then A:” as the “Boot Sequence.”
- Carry out a warm restart: booting is now carried out from the onboard silicon disk.

**Initial status:
only basic board without mass storage module present**

Initializing a bootable memory card:

In order to be able to generate a bootable memory card without mass storage module, the onboard silicon disk of the basic board used for this must be initialized and formatted, or CPLINK must be used (see Section 2.12).

If the onboard silicon disk of the basic board used is initialized and formatted, you can proceed as when creating a bootable memory card with mass storage module. Use the onboard silicon disk instead of the hard disk, and enter the memory card as "Drive A:" in the SETUP. Enter "A:" as the drive in the FORMAT command.

Initializing a bootable onboard silicon disk:

(The onboard silicon disk cannot be formatted or initialized for booting if neither a bootable memory card with the MS-DOS programs FDISK and FORMAT nor a mass storage module is present.) Exception: when using CPLINK.

If a bootable memory card is present:

1. Insert a bootable memory card with the MS-DOS programs FORMAT and FDISK.
2. Initialize a "Primary DOS partition" on the onboard silicon disk using the MS-DOS command FDISK.

After the command FDISK has been entered, a menu with 5 possible selections appears. "1" appears as the current hard disk above the menu.

Select the option "1" by pressing the Return key.

A submenu appears with 3 possible selections. Select the default setting (option "1") by pressing the RETURN key.

Answer the following question "Should the maximum available" with "Y."

Leave FDISK. The operating system is subsequently rebooted.

3. Format the onboard silicon disk using the **FORMAT** command:

FORMAT C: /U /S

4. Transfer the software which you require (MS-DOS, system software for CP 581 etc.) onto the onboard silicon disk.
5. Now activate the previously initialized partition on the onboard silicon disk using the **FDISK** command:

Select the function “2” (enter number and press RETURN key) in the FDISK menu.

Enter “1” as the active partition (enter number and press RETURN key).

Press the ESC key twice to leave FDISK.
6. In SETUP, now enter “C,A” or “C: then A:” for the “**Boot Sequence**” or “**Boot Options**”.
7. Carry out a warm restart: booting is now carried out from the onboard silicon disk.

Note

You must not use the MS-DOS commands “MIRROR” and “UNFORMAT” on the silicon disks since the status prior to formatting cannot then be recreated.

Note

The flash EPROMs are deleted when formatting or overwriting the memory card and onboard silicon disk. The number of delete operations is currently limited to approx. 10000 as a result of the physical characteristics of the flash EPROMs.

2.12 CPLINK

Startup program for a CP 581 without mass storage module using a programming device/PC.

2.12.1 Use, Functions and Structure of CPLINK

Characteristics and Purpose of CPLINK

CPLINK is a software package whose use enables the silicon disk or memory card of a CP 581 to be accessed by a programming device/PC in remote mode.

The block-oriented device driver CPLINK.EXE generates two additional drives in the programming device, where the first drive addresses the onboard silicon disk and the second drive addresses the memory card, in each case like a standard MS-DOS drive.

The physical connection is made via the **COM 1 serial interface** of the CP 581.

The COM interface on the programming device and the data transmission rate can be set between 9.6 kbps and 115.2 kbps.

The CP 581 automatically recognizes the baud rate set on the programming device when configuring CPLINK.

The “Remote Setup” function enables additional access via the serial interface to the SETUP stored in the BIOS of the module (prerequisite:CPLINK is installed).

Device Drivers

The CPLINK functions are implemented by one driver in the BIOS (EPROM) of the CP 581 (remote) and another in the programming device (local). The block device driver CPLINK.EXE is designed such that it can be addressed by MS-DOS using logical sector numbers.

The CPLINK drivers provide the file system of MS-DOS (≥ 5.0) with an additional access function to the remote drives (subsequent identification on programming device drives, for example, d:, e: or e:, f:).

Data Transfer

The file sectors passed on to the MS-DOS device driver are transmitted to or from the storage media on the remote module via the V.24 interface (see Figure 2-15). An interface for logical sector numbers is implemented in the silicon disk driver of the module in order to handle the data transfer (SW interrupt 60 h).

A declared character sequence is sent to the CP 581 during the initialization following loading of the driver into the system memory. The module recognizes the set baud rate from the signal. Once this has been recognized, the CP 581 sends an acknowledgment character, and the message “LINK established (COM?, ? bps)” appears on the screen.

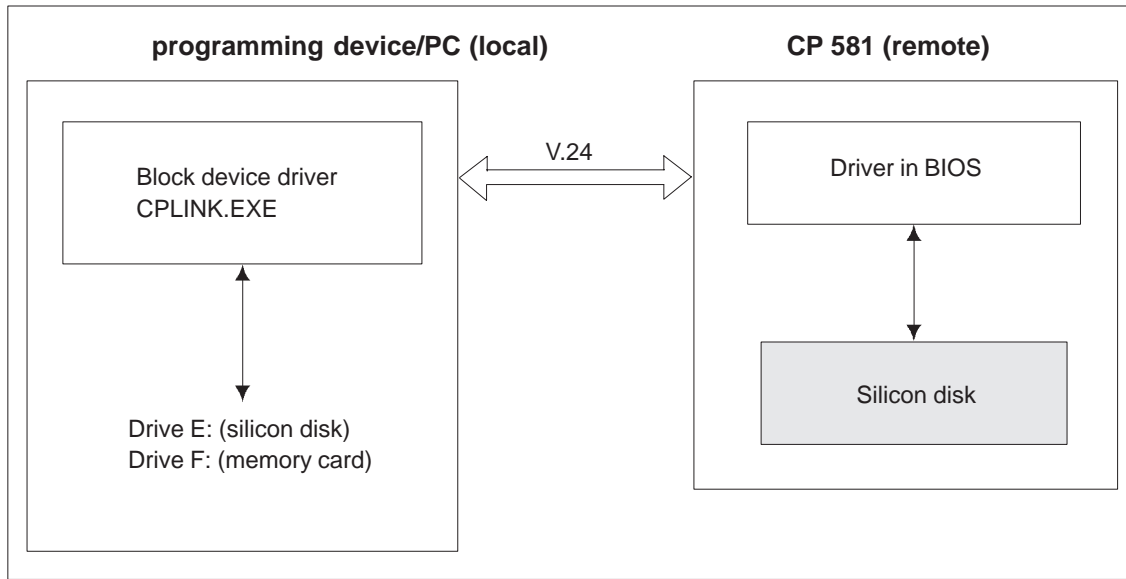


Figure 2-15 Data Transfer

CPLINK Structure

The V.24 handler in the two devices results in the serial transmission and reception of commands and data blocks. A generic I/O control interface is installed in the device driver to enable memory formatting using the standard MS-DOS command FORMAT.

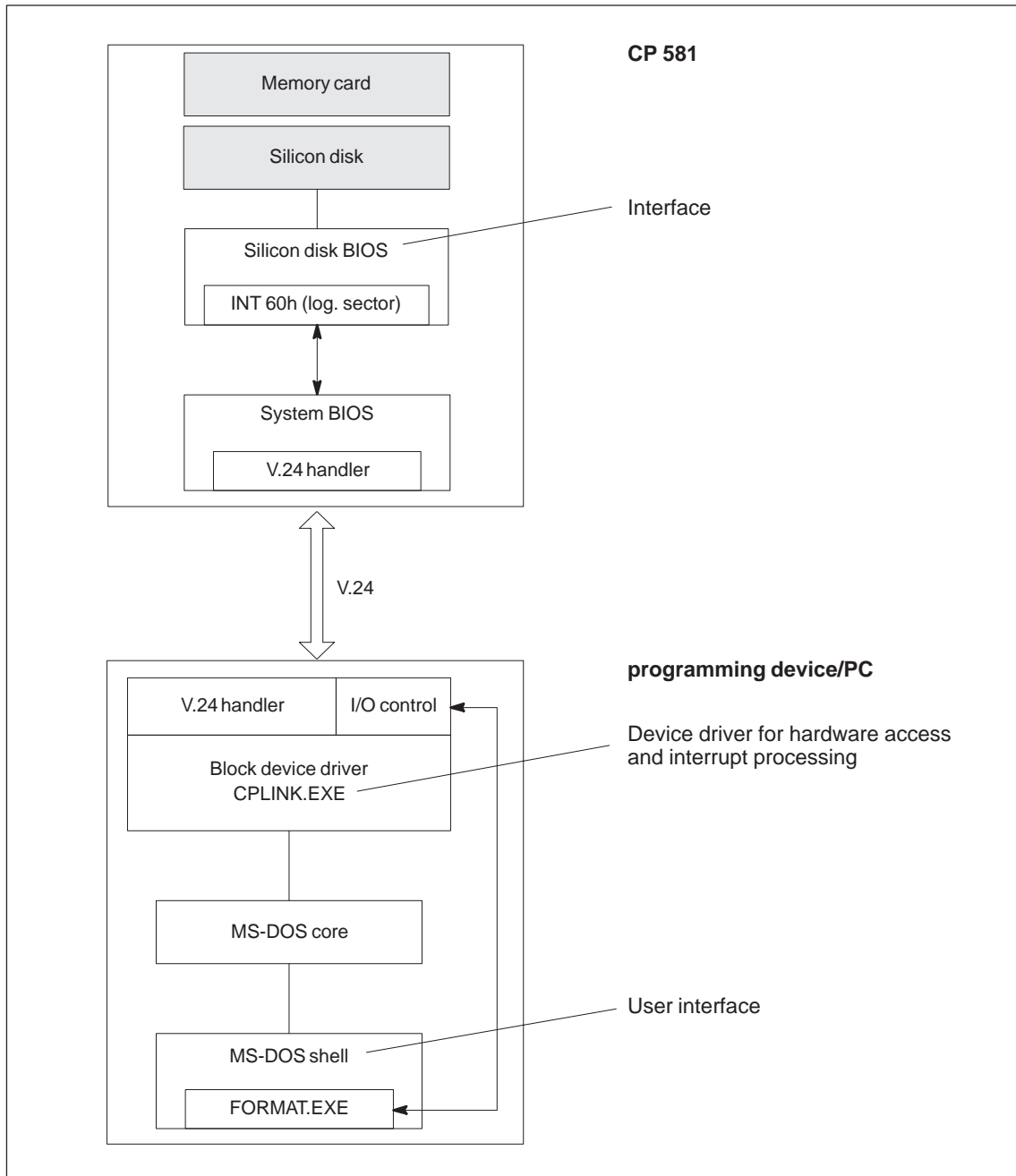


Figure 2-16 CPLINK Structure

Relevant Literature

- MS-DOS Programming Manual, Ray Duncan (editor)
- Writing DOS Device Drivers in C, P.M. Adams

2.12.2 Installation of CPLINK

Area of Use

CPLINK is implemented by a driver in the BIOS in the CP 581 (remote) and a block device driver CPLINK.EXE in the programming device/PC (local). CPLINK.EXE can also be called as an EXE file. Prerequisite is a flash BIOS (version 2.0 or later) in the programming device/PC.

CPLINK enables remote computer modules such as the CP 581 to be loaded and started using the programming device.

When delivered in the basic version (without mass storage module), these devices are “empty,” i.e. neither an operating system nor the CP software is loaded. CPLINK handles the task of installing the software onto the silicon disk or memory card of the remote module in a simple manner.

Delivery: Diskette (CP 581 system software)
File: CPLINK.EXE
Name/call: CPLINK

Note

A programming device/PC can only be connected to the CP 581 when the CP switch is in the STOP position.

Loading of Driver CPLINK

1. Copy the file “CPLINK.EXE” from the floppy disk onto the hard disk of your programming device/PC, for example, into the root directory.
2. Call the file C:\CONFIG.SYS using the MS-DOS editor “EDIT.”
3. Install the device driver using the entry:
DEVICE=CPLINK.EXE [c] [b].
or
DEVICEHIGH=CPLINK.EXE [c][b].
Refer to the table overleaf for the options [].
4. Leave the editor.
5. Carry out a cold or warm restart on your programming device.

The driver is loaded by DOS in the process, and the following message is output:

CPLINK Driver V ?, ?, Copyright xxxx Siemens AG
CPLINK Drives ?: and ?: installed.
Waiting for handshake from remote driver.
LINK established (COM?, ???bps).

CPLINK waits 10 s for the handshake from the remote driver. Without a reaction, CPLINK aborts with the error message “Remote V24 Comm Error.” In this case, the baud rate of the remote driver can be set later during initialization using the command “CPLINK /i”.

The operating system now provides two additional drives whose designations depend on the number of programming device drives. The corresponding designations are appended in each case, for example, if the drives a: to d: are present or defined, “e:” is used for the onboard silicon disk and “f:” for the memory card.

Installation Options

When entering the device driver into the CONFIG.SYS file, you must specify the interface used on the programming device/PC and the desired baud rate in the form of options if your installation does not correspond to the default values [c] = 1 and [b] = 1.

Examples:

DEVICE=CPLINK.EXE 1 1 Connection via COM 1 with 115200 bps
 DEVICEHIGH=CPLINK.EXE 2 4 Connection via COM 2 with 9600 bps

The following table shows the possible settings for the interface (COM) and the data transmission rate (baud rate). Over larger distances, operation is more reliable the smaller you select the baud rate.

Table 2-2 CPLINK: Settings for the COM Interface

Option [c]	Interface on the local station	Option [b]	Transmission rate (baud)
1 *	COM 1	1*	115 200 bps
2	COM 2	2	38 400 bps
3	COM 3	3	19 200 bps
4	COM 4	4	9 600 bps

* Default

Subsequent Initialization

The following text appears when calling CPLINK without one of the options /s, /i, /r or /a:

CPLINK V ?., Copyright xxxx Siemens AG

Syntax device-driver: device=cplink.exe <<c>> <>

<<c>> : 1=COM1, 2=COM2, 3=COM3, 4=COM4

<> : 1=115.2Kbps, 2=38.4Kbps, 3=19.2Kbps, 4=9.6Kbps

Syntax exe-file: CPLINK <<option>>

/s : Display status of device driver

/i : Initialize local and remote driver

/a : Abort remote driver (=)>> remote boot)

/r : Reset remote CP

When calling “CPLINK /s,” the designations (letters) of the installed drives are displayed:

CPLINK Device driver installed as drive?: and ?:

By calling “CPLINK /i” you initialize the drivers in the remote station (CP 581) and in the local station.

The V.24 interface in the local station (programming device/PC) is reset in the process, and a character sequence sent to the partner station for identification of the baud rate.

The CP 581 sends an acknowledgment character when it has recognized the current baud rate. The following message appears on the programming device:

Initializing local and remote driver - successful.

If the RUN/STOP switch on the CP 581 is set to STOP, the V.24 connection is established prior to booting of the station, and the baud rate determined. The following message appears on the screen as soon as the baud rate has been determined:

CPLINK V ?., Copyright xxxx Siemens AG

Waiting for handshake at COM1 (press any key to quit)

LINK established (???)bps)

If you wish to clear the V.24 connection and to boot the remote station, press any key on the CP 581 or set the RUN/STOP switch to RUN.

Renewed Start of Driver in Remote Station

The call “CPLINK /r” triggers a warm restart of the CP 581 and restarts the CPLINK driver. The local driver CPLINK.EXE, for example, can then be reinstalled with a different baud rate.

The following message appears on the screen of the programming device/PC:

Resetting remote CP - Done.

Aborting the Data Connection

The call “CPLINK /a” aborts processing of the device driver in the remote station (CP 581).

The system BIOS of the CP 581 is then continued and booted. The following message appears on the programming device/PC:

Aborting remote driver - Done.

2.12.3 Technical Information

Establishing / Clearing the Connection

The RUN/STOP switch is scanned in the BIOS of the CP 581 prior to booting. The V.24 connection is established if the switch is set to STOP, otherwise booting is carried out.

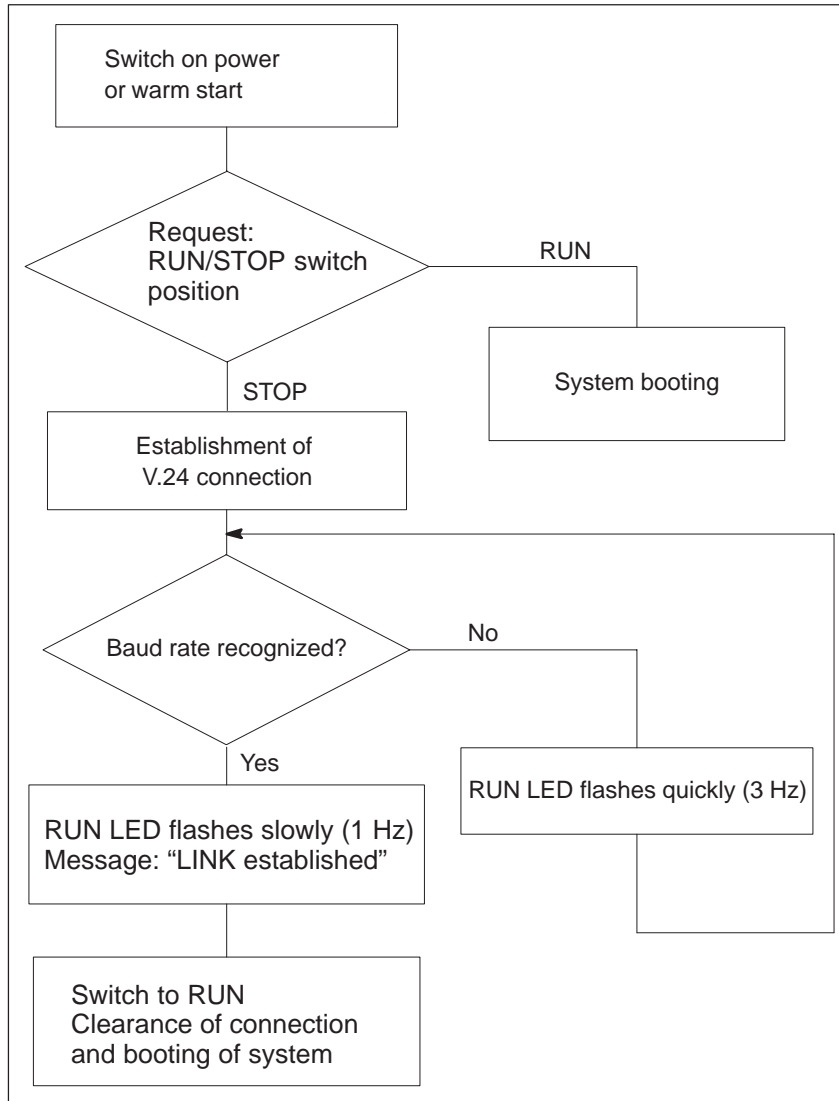


Figure 2-17 Establishing/Clearing the Connection in the Course of Booting

While the V.24 connection is being established, the current baud rate of the remote station is determined. To do this, the CP 581 waits with alternating baud rates for the agreed character sequence. The RUN LED flashes quickly at 3 Hz if the current baud rate has not yet been determined. Once the baud rate has been recognized, the LED flashes slowly at 1 Hz and the following message appears on the screen:

“LINK established (? bps).”

The V.24 connection is cleared following setting of the RUN/STOP switch to RUN, and the CP 581 is subsequently booted. The same procedure can be triggered by pressing any key on the CP 581.

V.24 Cable

A simple null modem cable is sufficient as the connecting cable for CPLINK applications, for example, with a 25-pin contact strip connector on the CP 581 (COM 1) and a 9-pin socket on the programming device (COM 2).

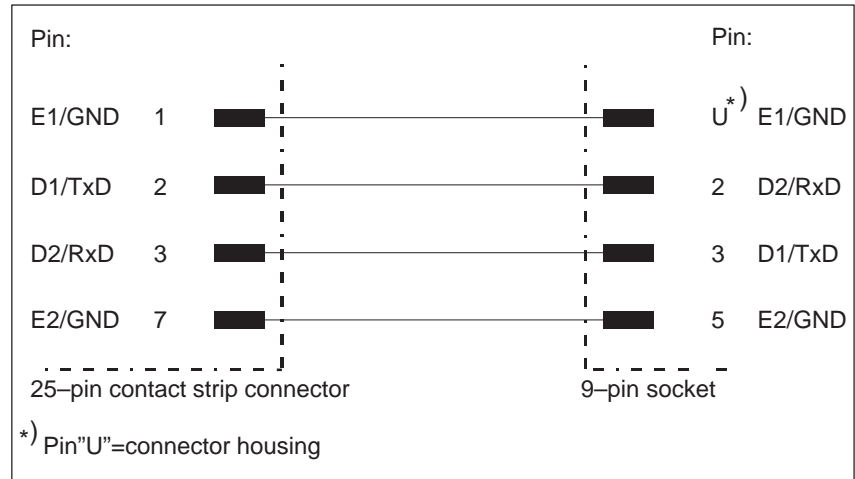


Figure 2-18 Assignment of Connecting Cable for CPLINK Application

Limitations with DOS Functions

The following drive-based DOS functions **cannot be used in the case of remote handling with CPLINK:**

Table 2-3

DOS Function	Task / effect
FASTOPEN	Stores directories in the system memory
FDISK	Set up new hard disk

Error Handling

The V.24 data transmission is timeout monitored and subjected to a cyclic redundancy check.

Abbreviations used:

- BIOS Basic input output system
- bps bits per second
- GND Ground (chassis, screen or plug housing)
- OSD Onboard silicon disk
- RxD Receive data
- TxD Transmit data
- V.24 Transmission standard: serial, asynchronous

2.13 Remote Operation via Serial Interface using CPLINK

Formatting the Silicon Disk or Memory Card

The Memory Card cannot be formatted unless it has been entered in BIOS setup (see “Configuring BIOS setup”). The standard DOS command FORMAT creates sectors on the memory medium, in this case the silicon disk or memory card. The root directory and the file allocation table (FAT) are created at the same time.

Always use the FORMAT instruction together with the option “/u”. Formatting of the memory medium is then carried out unconditionally.

A boot sector is also created if you inform FORMAT by means of the option “/s” to carry out a system transmission (MS-DOS) simultaneously with the formatting. It is then possible to boot the CP 581 from the memory card or the onboard silicon disk.

The following specification of the FORMAT instruction results in formatting of the silicon disk including creation of a partition table and copying of the system files (in this case the programming device possesses four real/logical drives):

```
FORMAT e: /u /s
```

Note

Although the onboard silicon disk serves as the hard disk, you do not need to use FDISK.

Just format the disk via CPLINK.

Configuring BIOS Setup

Using an ANSI terminal or an ANSI terminal emulation (see Section 2.13.1) or the HOSTKEY program (see Section 2.13.2), you can enter the BIOS setup of the CP 581 with the help of a programming device/PC. To accomplish this, you will have to establish a serial connection from the programming device/PC to the CP 581.

In BIOS setup, enter the Memory Card as drive A. BIOS setup is described in Sections 3.7 and 3.8.

Installing the CP System Software

Once you have entered the Memory Card in BIOS setup, you can install the system software.

Using the MS-DOS command, “FORMAT e: /s”, prepare the MS-DOS system onto the silicon disk, then install the CP 581 system software by entering “install e:”.

Loading of User Software

```
copy *.* e:\  
copy autoexec.bat e\  
copy config.sys e\  

```

2.13.1 Remote BIOS Setup for the CP 581 DX

Preparing Remote Setup Mode for the DX

To operate the remote SETUP, you must connect an ANSI terminal or an ANSI terminal emulation to the COM 1 or COM 2 of the CP 581.

If the connection is to be established when the CP runs up, you must keep the “Q” key pressed. As soon as the interfaces have been synchronized, the message “U” appears on the terminal screen.

When connecting the CP 581 to the terminal or the terminal emulation, you should use a connecting cable with the assignment listed in the following table.

Table 2-4 Connecting Cable Assignment for Remote Setup Mode

Signal	PIN	Connection	PIN	Signal
E1/GND	U	Connected to	U	E1/GND
D2/RxD	2		2	D1/TxD
D1/TxD	3		3	D2/RxD
E2/GND	5		7	E2/GND
S2/RTS	7		5	U5/CTS
U2/CTS	8		4	S2/RTS
9-pin Sub-D socket			PIN “U” = housing (screen) Length: 10 m maximum	
PG/PC COM 2				CP 581 COM 1
CP 581 COM 2				PG/PC COM 1

RUN/STOP Switch

The setting of the RUN/STOP switch is irrelevant for the remote setup.

Configuring the COM Interface for the programming device/PC

For remote setup mode, the following settings are required for the COM interface for the programming device/PC:

- 8 data bits
- no parity
- 1 stop bit
- Transmission rate: 9600, 19200 or 38400 bps
- ANSI

Starting BIOS Setup

After you have carried out all preparations and have configured the interface for the programming device/PC, start the CP 581 DX. Press and hold the “Q” key on the programming device/PC for the entire time while the CP 581 DX is booting up. As soon as the interfaces have been synchronized, the message “U” appears on the terminal screen.

You can then call up BIOS setup from the terminal or terminal emulation by pressing the <ESC> key. The terminal or terminal emulation then displays the BIOS setup screen of the CP 581 DX, which you can use to define your configuration.

If the RUN/STOP switch is set to STOP after you have left the setup function, you can begin the CPLINK operation by starting CPLINK on the PC or programming device.

If the RUN/STOP is set to RUN after you have left the setup function, the CP 581 DX automatically boots.

2.13.2 Remote BIOS Setup for the CP 581 Pentium

Preparing Remote Setup Mode for the Pentium

For remote setup mode, connect the COM1 or COM2 interface of a programming device/PC to the COM1 interface of the CP 581 Pentium. A simple null-modem cable is sufficient for the connection (see Figure 2-18).

After you have established the serial connection, start the HOSTKEY program on the programming device/PC (included with the CP 581 system software as of Version 2.9).

The HOSTKEY program temporarily assigns the keyboard of the CP 581 Pentium to the programming device/PC monitor, rerouting the screen output and the keyboard input from/to the CP 581 Pentium.

The setting of the RUN/STOP switch is irrelevant for remote setup mode.

Settings for HOSTKEY

The HOSTKEY program has various call options, which are illustrated in the following table.

Syntax:

HOSTKEY[/?][/Cx][Sxx][{NOF}[/V]

Example for the HOSTKEY call:

HOSTKEY /C2/NOF

Option	Meaning
/?	Displays the options on the screen
/C1	COM 1
/C2	COM 2
/S96	9600 baud
/S192	19200 baud
/S384	38400 baud
/S576	57600 baud
/S1152	115200 baud
/NOF ¹⁾	Skip check for disk when loading HOSTKEY
/V ²⁾	Output HOSTKEY message texts to the monitor
Call without parameters	Activates the COM1 interface at 115.2 k baud

- 1) If the /NOF option is not used, a floppy disk must be present in drive A: of the programming device/PC. Specifying the /NOF option is recommended.
- 2) The HOSTKEY message texts are displayed on the monitor along with the boot texts. If you only want to have the boot texts displayed, do not include the /V option when calling up HOSTKEY.

Controlling HOSTKEY

Once HOSTKEY has been started and a connection is detected, subsequent keyboard input is sent to the CP 581 Pentium.

The following control commands are exceptions to this and are sent to HOSTKEY:

- CTRL-X Aborts HOSTKEY
- CTRL-ALT-F10 Reboots the CP 581 Pentium *)

*)Only applies if there a HOSTKEY connection exists.

Starting BIOS Setup

After you have (1) called up the HOSTKEY program and (2) started the CP 581 Pentium (note the order of these events!), you will see the boot process of the CP 581 Pentium on the screen emulation of the programming device/PC .

Wait until the “Remote-Setup enabled*” message appears on the screen and then press the F2 key on the programming device/PC keyboard. The screen emulation of the programming device/PC then displays the BIOS setup screen of the CP 581 Pentium, which you can use to define your configuration.

If the RUN/STOP switch is set to STOP after you have left the setup function, you can begin the CPLINK operation. To do so, HOSTKEY must be closed on the programming device/PC and CPLINK must be started.

If the RUN/STOP is set to RUN after you have left the setup function, the CP 581 Pentium automatically boots.

Terminating HOSTKEY

After you have closed BIOS setup, you can end the HOSTKEY program by pressing the CTRL-X key combination.

If you do not terminate HOSTKEY, it will again detect a connection the next time you start the CP 581 Pentium. If, at this point, you do not enter BIOS setup, the connection is canceled automatically by the CP 581 Pentium (screen message: “*Remote-Function terminated*”), i.e. screen emulation output is not available on the programming device/PC and keyboard input from the programming device/PC to the CP 581 Pentium is not possible.

Note

HOSTKEY and Microsoft WindowsNT

The HOSTKEY program can cause procedure problems under Microsoft WindowsNT. WindowsNT operating system generally supports DOS-based interface programs (like HOSTKEY) but not yet sufficiently, making the ability to use HOSTKEY dependent on your concrete PC configuration. If you run into such problems, we recommend using another PC with a different operating system or rebooting the programming device/PC using the MS-DOS 6.22 operating system.

3

Reference Section for Hardware

This chapter provides you with detailed information on the hardware design of the CP 581. Each CP 581 component is considered separately. This chapter is divided accordingly.

You can read:

- How the CP 581 components are constructed, and what meaning the switch and jumper settings have,
- Which devices you can connect,
- How the CP 581 interfaces are assigned,
- What the memory and hardware interrupt assignments of the CP 581 are,
- Which interface assignments you require for the development of your own CP 581 system software or your own drivers,
- Information for the developers of own system software.

Chapter Overview

Section	Description	Page
3.1	Basic Board	3-2
3.2	Mass Storage Module	3-29
3.3	AT Slot Module	3-38
3.4	Remote Terminal Interface	3-42
3.5	Y Adapter	3-48
3.6	Silicon Disk	3-50
3.7	BIOS Setup for CP 581 80486 DX	3-52
3.8	BIOS Setup for CP 581 Pentium	3-61
3.9	Conversion and Repairs	3-70
3.10	Information for Developers of Own System Software	3-70

3.1 Basic Board

3.1.1 Mechanical Construction

The CP 581 basic board (printed circuit board) is of double Eurocard format and thus suitable for the ES 902 modular packaging system. The front panel width is 1 1/3 standard slots (approx. 20 mm) wide and thus occupies one S5 slot in the main frame of the programmable controller.

The front panel contains the displays and controls as well as the interfaces to the operation and peripheral devices. Figures 3-1 and 3-2 show you the design of the CP 581 basic board.

The basic board CP 581 with 80486 DX processor consists of:

- Two 48-pin rearpanel connectors X1 and X2 according to DIN 41 612, series 2, for connecting the CP 581 basic board to the S5 bus (rearpanel bus) of the programmable controllers (rearpanel connectors 1 and 2)
- One 26-pin high-density plug for connecting a printer (COM 1) and/or computer coupling (COM 3) using a Y adapter
- One 26-pin high-density plug for connecting a keyboard, a mouse, or a light pen (COM 2) using a Y adapter or RTI module
- Three coax midi sockets for connecting the monitor
- One 128-pin female connector for connecting the mass storage module (internal ISA bus)
- One RUN/STOP switch for selecting the mode
- RUN/STOP/SD BUSY/FAULT LEDs, RESET key
- One processor core
- One S5 bus interface (slave)
- VGA graphics with resolution of 1024 x 768 pixels and up to 256 colors (see Chapter 4, Table 4-1)
One keyboard interface and one keyboard controller
- Slot for RAM module (X205)
- Slot for flash disk module (X204)

The basic board CP 581 with Pentium processor consists of:

- Two 48-pin rearpanel connectors X1 and X2 according to DIN 41 612, series 2, for connecting the CP 581 basic board to the S5 bus (rearpanel bus) of the programmable controllers (rearpanel connectors 1 and 2)
- One 26-pin high-density plug for connecting a printer (COM 1) and/or computer coupling (COM 3) using a Y adapter
- One 26-pin high-density plug for connecting a keyboard or a mouse (COM 2) using a Y adapter or RTI module
- three COAX midi jacks or a 15-pin VGA plug for connecting the monitor
- One 128-pin female connector for connecting the mass storage module (internal ISA bus)
- One RUN/STOP switch for selecting the mode
- RUN/STOP/SD BUSY/FAULT LEDs, RESET key
- One processor core
- One S5 bus interface (slave)
- VGA graphics with resolution of 1024 x 768 pixels and up to 256 colors (see Chapter 4, Table 4-2)
One keyboard interface and one keyboard controller
- Two slots for RAM modules (X204)
- Slot for flash disk module (X252)

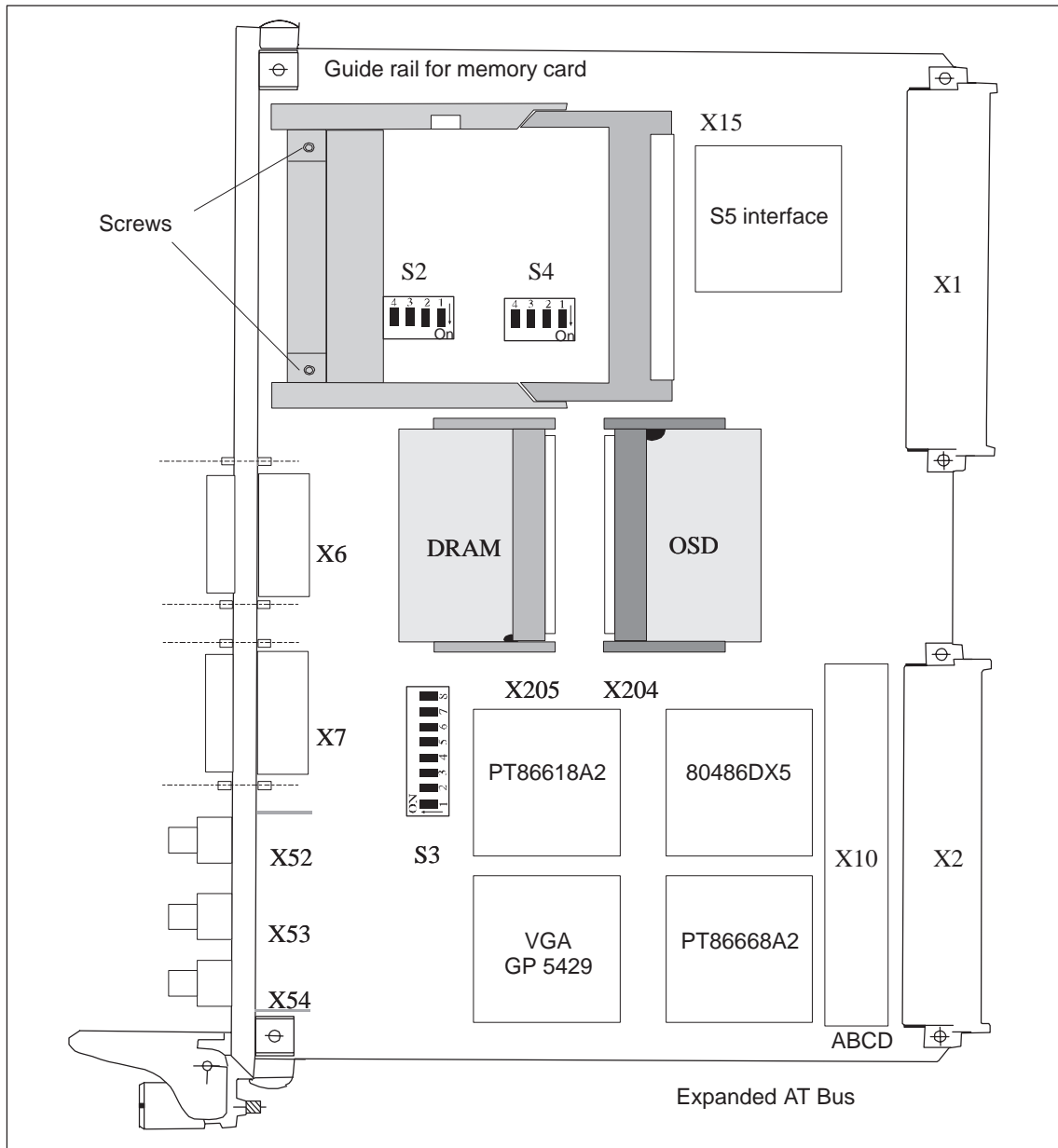


Figure 3-1 Layout of the CP 581 Basic Board (80486 DX)

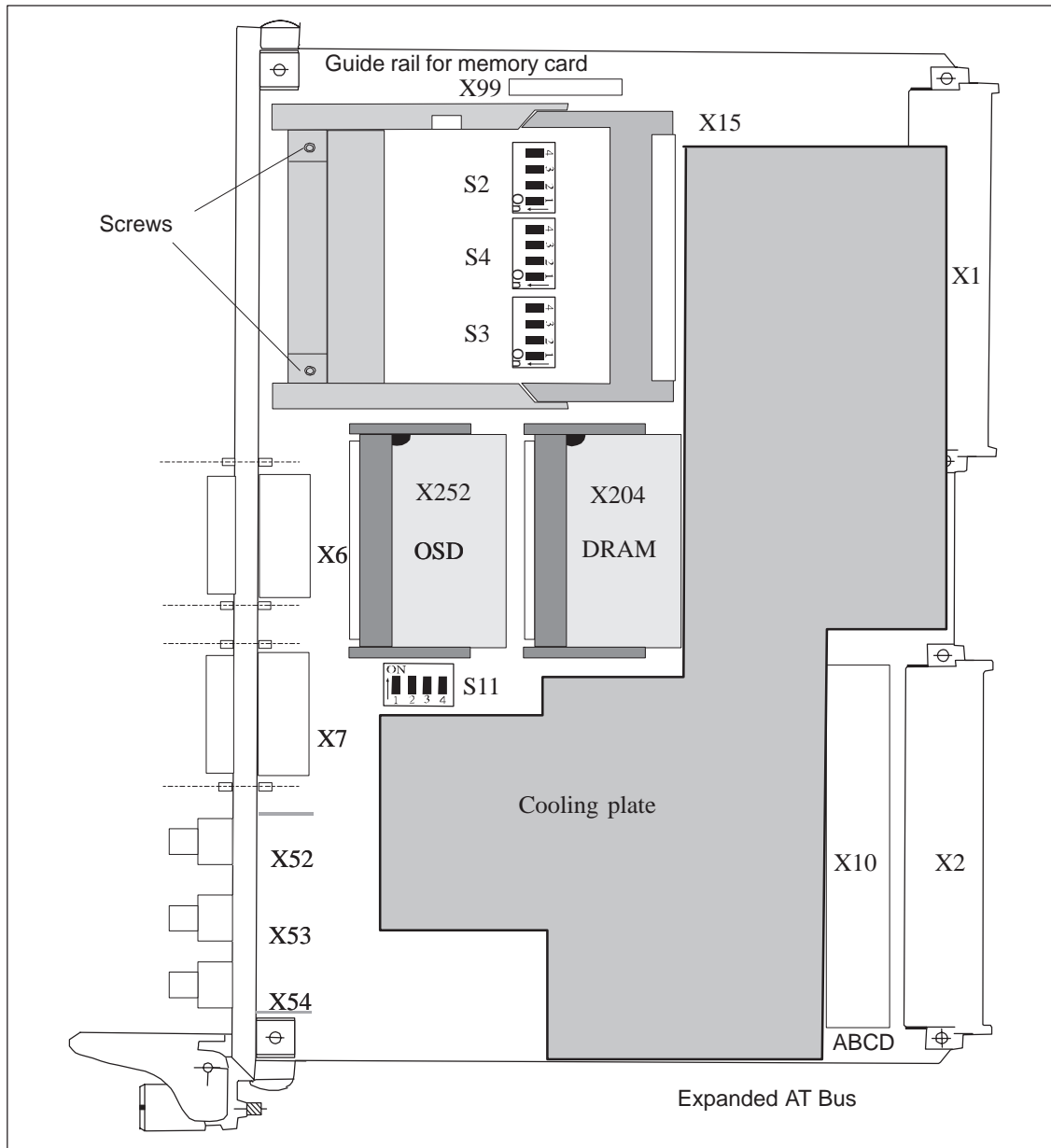


Figure 3-2 Layout of the CP 581 Basic Board (Pentium with RGB)

Instead of plugs X52, X53, and X54, a standard X5 VGA plug is on the CP 581 basic board with Pentium and VGA.



Caution

Risk of damage to the product.

If you operate the board without the cooling plate, it can heat up to the extent that it is destroyed.

Never remove the cooling plate.

3.1.2 Controls and Displays

Figure 3-3 shows you the positions of the controls and displays on the front panel of the basic board.

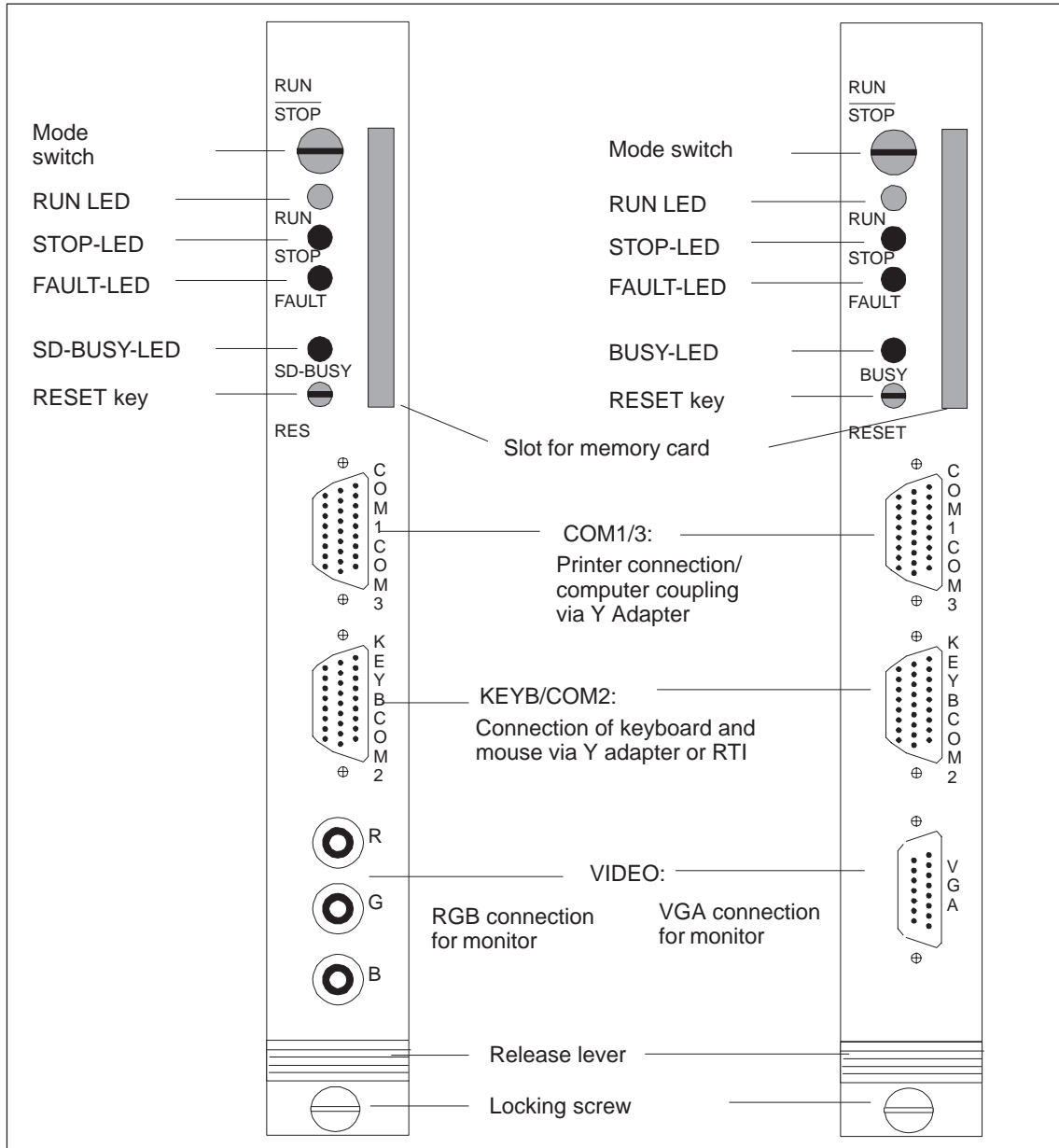


Figure 3-3 Position of Controls, Displays and Interfaces on the Basic Board.

Controls:

RUN/STOP switch:

Switching from RUN to STOP:

Bus communication at the S5 interface is disabled.

A RESET ALL is triggered at the CP 581 end and completely resets the CPDHB driver. (This corresponds to triggering of a RESET ALL by the corresponding DHB at the CPU end). If the switch is set to STOP when the computer is run up, CPLINK is activated.

Switching from STOP to RUN:

Data transfer at the S5 interface is enabled again.

RESET key:

Pressing this key completely resets the board.

Communication is then aborted in an undefined manner! The error condition code "Interface not ready" is passed on to the S5 CPU.

Displays:

The displays are divided into:

- Operation displays
- Fault displays
- Access displays for the drives.

Operation and fault displays:

- Green LED: RUN
Signals the status "S5 interface in operation"; the LED can only light up if the mode switch is set to RUN and if at least one CPU is synchronized.
- Red LED: STOP
Continuous light signals the status "S5 interface not in operation" if the mode switch is set to STOP.
Flashing signals the status "Mode switch at RUN," but no CPU synchronized.
- Red LED: FAULT
This LED is deleted by the BIOS during the CP restart if the CP hardware is ready.
- Green LED: SD BUSY
Lights up during access operations to the onboard silicon disk and the memory card.

The following table shows you the meaning of the displays:

Table 3-1 Operation and Fault Displays on the LEDs

FAULT LED *	RUN LED	STOP-LED	Meaning
0	0	0	CPDHB driver is not yet loaded on the CP 581.
0	0	1 (continuous)	RUN/STOP switch in STOP position
0	0	Flashing	RUN/STOP switch in RUN position. CP 581 is not synchronized with any CPU.
0	1	0	RUN/STOP switch in RUN position. At least one CPU is synchronized.
1	0	1	BIOS error stop (error booting the CP 581)
0	Flashing: 3x per sec.	0	CPLINK active without connection
0	Flashing: 1x per sec.	0	CPLINK active with connection

* The FAULT LED goes off during the restart. If this LED lights up without the STOP LED, an error has occurred.

3.1.3 Switch Settings for the CP 581 80486 DX

The coding switches present on the basic board have been factory set. You need not check or reset them provided you wish to use the CP 581 components in their default configuration.

The following sections provide you with further information on the coding switches.

Figure 3-1 shows the position of the switches.

Table 3-2 Switch S2 for the CP 581 80486 DX

No.	Function	ON	OFF	Effect	
				6ES5581-0ED13	6ES5581-0ED11
1	Processor frequency	X		75MHz (default)	100MHz (default)
			X	100 MHz	133 MHz
2	VGA operation	X		VGA switched on (default setting)	
			X	VGA switched off *)	
3	RI on COM 2	X		RI/COM 2 (default setting)	
			X		
4	Light pen at COM 2	X		Light pen (default setting)	
			X		

*) It is possible to connect an external VGA card to the AT slot

Note

Operation with a processor frequency of 133 MHz (80486-DX5-133) is only permitted with forced ventilation.

Table 3-3 Switch S4 for the CP 581 80486 DX

No.	Function	ON	OFF	Effect
1	RxD COM 2 RS485	X		RxD RS485 ¹⁾
			X	
2	RxD COM 2 RS422	X		RxD RS422 (default setting) ^{1) 2)}
			X	(default setting)
3	RxD COM 3 RS485	X		RxD RS485 ³⁾
			X	(default setting)
4	RxD COM 3 RS422	X		RxD RS422 (default setting) ³⁾
			X	

1) Only either "1" or "2" may be set to ON, never both switches simultaneously

2) Mouse operation via RTI

3) Only either "3" or "4" may be set to ON, never both switches simultaneously

Table 3-4 RS 422 Communication with the CP 581 80486 DX

COM Port	COM 3 (Sub D25 of the Y adapter)	COM 2 (Sub D9 of the Y adapter)
Relevant cables: RxD+ (—>) RxD- (<—) TxD+ (—>) TxD- (<—)	Pins: 11 13 12 14	
RS 422 mode Activate: RTS bit in modem control register must be set to 1	Address: 3ECh for example outportb (0x3EC,02)	Address: 2FCh for example outportb (0x2FC,02)
Hardware settings:	CP581DX switch S4: Switch3:OFF Switch4:ON	CP581DX switch S4: Switch1:OFF Switch2:ON

Table 3-5 RS 485 Communication with the CP 581 80486 DX

COM Port	COM 3 (Sub D25 of the Y adapter)	COM 2 (Sub D9 of the Y adapter)
Relevant signals: TxD+ (—>) TxD- (<—)	Pins: 12 14	
Switching direction: RTS bit in modem control register is manipulated Transmit: RTS-Bit=1 Receive: RTS-Bit=0	Address: 3ECh For example: outportb (0x3EC,02) outportb (0x3EC,02)	Address: 2FCh for example: outportb (0x2FC,02) outportb (0x2FC,02)
Hardware settings:	CP581DX switch S4: Switch3:ON Switch4:OFF	CP581DX switch S4: Switch1:ON Switch2:OFF

Table 3-6 Switch S3

No.	Function	ON	OFF	Effect
1	COM 3 interrupt	X		Interrupt 4
			X	No interrupt (default setting)
2	COM 3 interrupt	X		Interrupt 5
			X	No interrupt (default setting)
3	COM 3 interrupt	X		Interrupt 10
			X	No interrupt (default setting)
4	COM 3 interrupt	X		Interrupt 11 (default setting)
			X	No interrupt
5	COM 1 TTY transmit	X		TTY active (default setting)
			X	TTY passive
6	COM 1-TTY receive	X		TTY active (default setting)
			X	TTY passive
7	Keyboard	X		PG 750 keyboard
			X	Standard keyboard (default setting)
8	May not be changed	X		
			X	(default setting)

3.1.4 Switch Settings for the CP 581 Pentium

The coding switches present on the basic board have been factory set. You need not check or reset them provided you wish to use the CP 581 components in their default configuration.

The following sections provide you with further information on the coding switches.

Figure 3-2 shows the position of the switches.

Table 3-7 Switch S2 for the CP 581 Pentium

No.	Function	OFF	ON	No.	Effect
1	COM 3 interrupt		X	5	Interrupt 4
		X			No interrupt (default setting)
2	COM 3 interrupt		X	6	Interrupt 5
		X			No interrupt (default setting)
3	COM 3 interrupt		X	7	Interrupt 10
		X			No interrupt (default setting)
4	COM 3 interrupt		X	8	Interrupt 11 (default setting)
		X			No interrupt

Table 3-8 Switch S3 for the CP 581 Pentium

No.	Function	OFF	ON	No.	Effect
1	COM 1-TTY receive		X	5	TTY active (default setting)
		X			TTY passive
2	COM 1 TTY transmit		X	6	TTY active (default setting)
		X			TTY passive
3	Keyboard		X	7	PG 750 keyboard
		X			Standard keyboard (default setting)
4	External VGA switched on		X	8	External VGA on
		X			External VGA off (default setting)

Table 3-9 Switch S4 for the CP 581 Pentium

No.	Function	OFF	ON	No.	Effect
1	RxD COM 2 RS485		X	5	RxD RS485 ¹⁾
		X			(default setting)
2	RxD COM 2 RS422		X	6	RxD RS422 (default setting) ^{1) 2)}
		X			
3	RxD COM 3 RS485		X	7	RxD RS485 ³⁾
		X			(default setting)
4	RxD COM 3 RS422		X	8	RxD RS422 (default setting) ³⁾
		X			

1) Only either "1" or "2" may be set to ON, never both switches simultaneously

2) Mouse operation via RTI

3) Only either "3" or "4" may be set to ON, never both switches simultaneously

Table 3-10 RS 422 Communication with the CP 581 Pentium

COM Port	COM 3 (Sub D25 of the Y adapter)	COM 2 (Sub D9 of the Y adapter)
Relevant cables: RxD+ (—>) RxD- (<—) TxD+ (—>) TxD- (<—)	Pins: 11 13 12 14	
RS 422 mode Activate: RTS bit in modem control register must be set to 1	Address: 3ECh for example outportb (0x3EC,02)	Address: 2FCh for example outportb (0x2FC,02)
Hardware settings:	CP581 Pentium switch S4: Switch3:OFF Switch4:ON	CP581 Pentium switch S4: Switch1:OFF Switch2:ON

Table 3-11 RS 485 Communication with the CP 581 Pentium

COM Port	COM 3 (Sub D25 of the Y adapter)	COM 2 (Sub D9 of the Y adapter)
Relevant signals: TxD+ (—>) TxD- (<—)	Pins: 12 14	
Switching direction: RTS bit in modem control register is manipulated Transmit: RTS-Bit=1 Receive: RTS-Bit=0	Address: 3ECh For example: outportb (0x3EC,02) outportb (0x3EC,02)	Address: 2FCh for example: outportb (0x2FC,02) outportb (0x2FC,02)
Hardware settings:	CP581 Pentium switch S4: Switch3:ON Switch4:OFF	CP581 Pentium switch S4: Switch1:ON Switch2:OFF

Table 3-12 Switch S11 Frequency Setting for the CP 581 Pentium

No. 1	No. 2	No. 3	No. 4	Pentium clock rate
ON	ON	OFF	OFF	75 MHz (default setting)
OFF	ON	ON	OFF	133 MHz, only permitted with forced ventilation

Settings that are not listed in the table are not allowed.

3.1.5 Connection of Printers

We recommend the following Siemens printers if you wish to connect a printer to COM 1 of the Y adapter COM1/COM 3:

Table 3-13 Recommended Printers

Printer	Printing principle	Paper	Order number
DR 215-N	9-needle printer	A4	6AP1800-0BB00
DR 216-N	9-needle printer	A3	6AP1800-0BD00
DR 235-N	24-needle printer	A4	6AP1800-0BF00
DR 236-N	24-needle printer	A3	6AP1800-0BH00

All printers listed here have the following common technical data:

- Centronics, V.24 and TTY interfaces
- ANSI, EPSON and IBM-compatible (DR 215-N and DR 216-N)
- ANSI, EPSON, IBM, NEC and ECMA-compatible (DR 235-N and DR 236-N)
- Low noise level.

Refer to Chapter 5, Ordering Information, for the order nos. of suitable interfaces and cable connectors.

Please note that you must order the interfaces and cable connectors separately.

You can obtain further information from the corresponding printer descriptions.

3.1.6 Connection of Keyboard

We recommend the connection of a PG 750 keyboard.

- Connection via the Y adapter Keyboard/COM 2
Connect the keyboard to the 7-pin round socket of the Y adapter. (If necessary, use the Din → Mini-Din adapter provided.)
- Connection via the remote terminal interface
Connect the keyboard to the round socket X2 of the RTI module.

3.1.7 Connection of a Mouse

The CP 581 software supplied does not use a mouse.

You can nevertheless use a mouse if this is supported by the MS-DOS programs which you use on the CP 581.

We recommend the connection of a PG 750 mouse.

- Connection via the Y adapter Keyboard/COM 2
Connect the mouse to the 9-pin D-subminiature plug of the Y adapter.
- Connection via the remote terminal interface
Connect the mouse to the 9-pin D-subminiature plug X3 of the RTI module.
The mouse must not be connected to the PG 750 keyboard!

3.1.8 Connection of a Monitor

We recommend a monitor where the video ground can be separated from the protective ground.

It is essential that you observe the design and connection guidelines for monitors in Section 2.9.2!

3.1.9 External Interface Assignments

When connecting the operation devices and the peripheral devices, please note that you require the Y adapter listed in Chapter 5, Ordering Information, when using standard cables.

Combined Plug X6 for COM 1 and COM 3

The combined plug for the serial interfaces COM 1 and COM 3 is designed as a 26-pin high-density plug.

The serial interface COM 1 has V.24 transmission signals plus the signals for active TTY mode (20 mA). It is compatible with the industrial standard and can be used to connect printers with serial interfaces.

The serial interface COM 3 has V.24 and X27 transmission signals and can be used for computer coupling.

The signals from COM 1 and COM 3 are converted to the 25-pin D-subminiature plug via the Y adapter COM 1/COM 3.

Table 3-14 Pin Assignments of Plug X6

PIN	Signal	Signal class
1	RI	COM 1 V.24
2	DCD	COM 1 V.24
3	RTS	COM 1 V.24
4	RxD+	COM 3 X27
5	RxD-	COM 3 X27
6	TxD+	COM 3 X27
7	TxD-	COM 3 X27
8	RTS	COM 3 V.24
9	CTS	COM 3 V.24
10	DSR	COM 1 V.24
11	RxD	COM 1 V.24
12	GND	COM 1 V.24
13	P24V	COM 1 TTY
14	RxD+	COM 1 TTY
15	TxD+	COM 1 TTY
16	DCD	COM 3 V.24
17	DSR	COM 3 V.24
18	GND	COM 3 V.24
19	TxD	COM 1 V.24
20	DTR	COM 1 V.24
21	CTS	COM 1 V.24
22	RxD-	COM 1 TTY
23	TxD-	COM 1 TTY
24	TxD	COM 3 V.24
25	RxD	COM 3 V.24
26	DTR	COM 3 V.24

Note

The COM 3 interface does not have an RI signal.

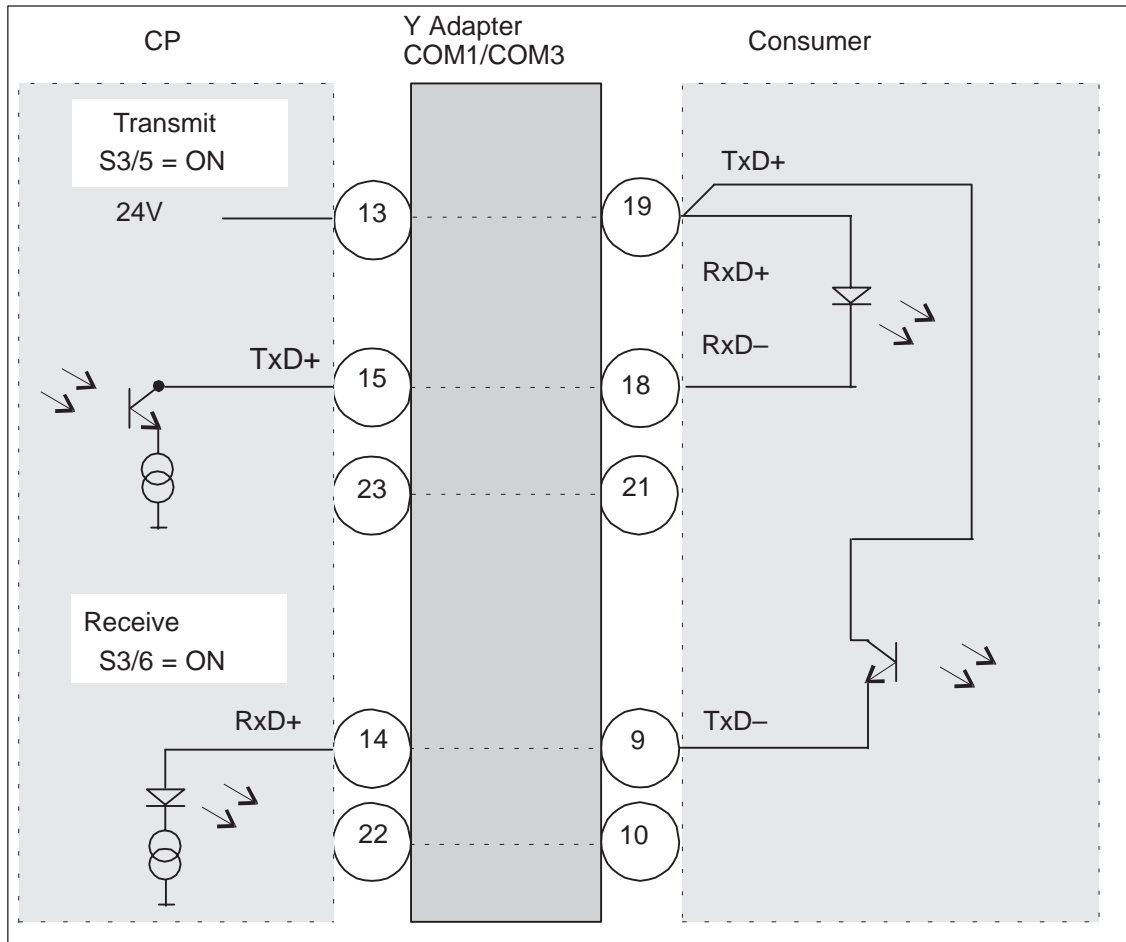


Figure 3-4 TTY Connection: CP 581 Active

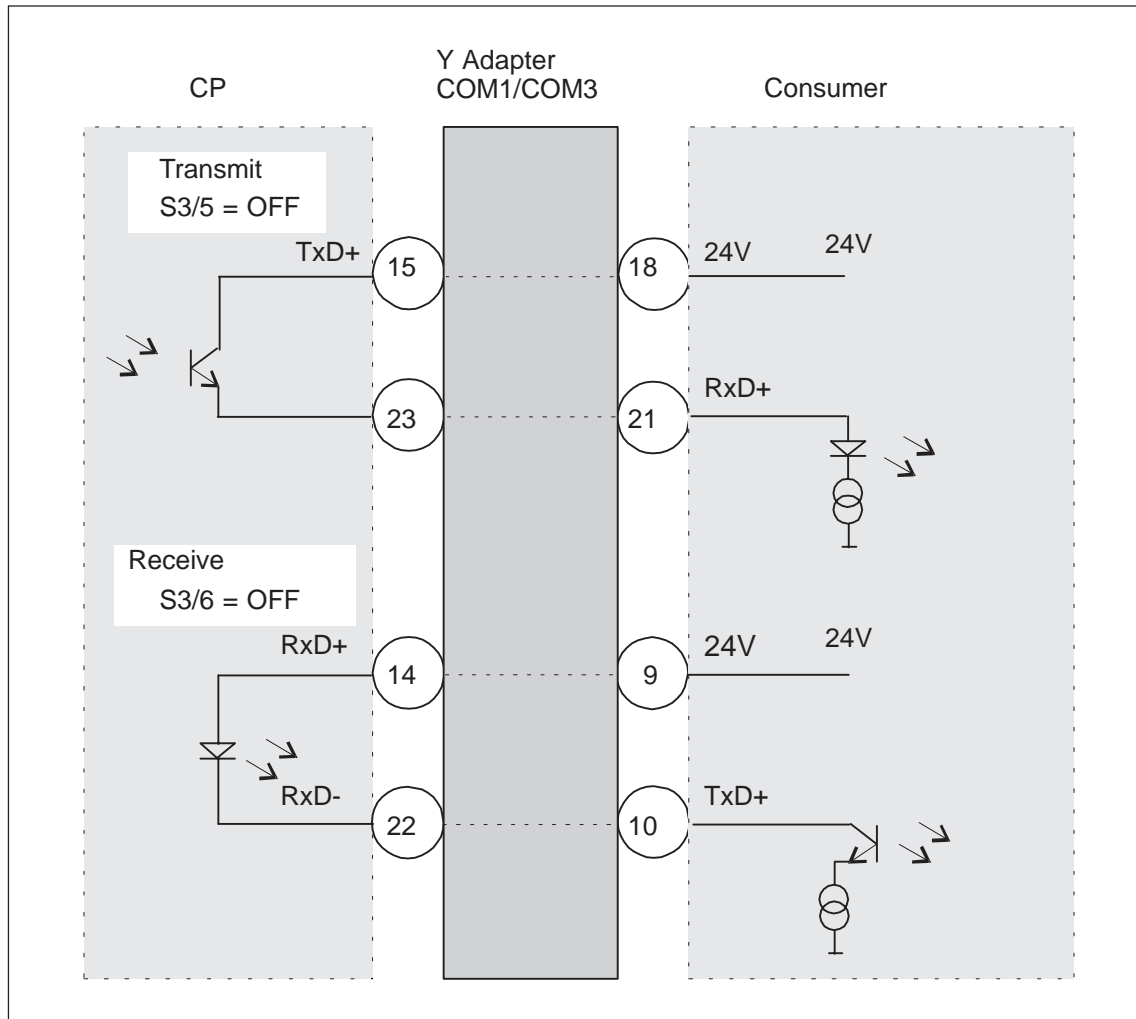


Figure 3-5 TTY Connection: CP 581 Passive

Note

You must not use the standard programming device cable to connect a programming device via the TTY interface COM 1. Pin 22 (CP side) must not be connected to Pin 10. A special cable should be used.

Combined Plug X7 for the Keyboard and COM2 (Mouse)

The combined plug for the keyboard and the serial interface COM 2 is designed as a 26-pin high-density plug.

It has TTL signals for the keyboard interface, V.24 signals for the mouse connection, RS 485 signals for the RTI interface and a 24 V power supply.

The signals are converted to the round keyboard socket (KBD connection) and a 9-pin D-subminiature plug via the Y adapter Keyboard/COM 2.

You can cover larger distances (max. 250 m) between the CP 581 and the keyboard/mouse using the remote terminal interface.

Table 3-15 Pin Assignments of Plug X7 for Keyboard and COM 2

Pin	Signal	Signal class
1	K_DATA	Keyboard direct
2	K_CLK	Keyboard direct
3	P5V	Keyboard direct
4	TxD	COM 2 V.24
5	RTS	COM 2 V.24
6	DTR	COM 2 V.24
7	Not used	
8	RTS -	COM 2 remote
9	RTS +	COM 2 remote
10	K_DATA+	Keyboard remote
11	K_CLK+	Keyboard remote
12	GND	
13	RxD	COM 2 V.24
14	CTS	COM 2 V.24
15	DSR	COM 2 V.24
16	RI	COM 2 V.24
17	DCD	COM 2 V.24
18	M24V	Remote supply
19	K_DATA-	Keyboard remote
20	K-CLK-	Keyboard remote
21	GND	
22	RxD+	COM 2 remote
23	RxD-	COM 2 remote
24	TxD+	COM 2 remote
25	TxD-	COM 2 remote
26	P24V	Remote supply

Plug X15 for a Memory Card

68-pin connector for a memory card

Table 3-16 Pin Assignments of Plug X15 for a Memory Card

Pin	Signal	Function	I/O
1	GND	Ground	
2	D3	Data bit 3	I / O
3	D4	Data bit 4	I / O
4	D5	Data bit 5	I / O
5	D6	Data bit 6	I / O
6	D7	Data bit 7	I / O
7	CE1_N	Enable low byte	I
8	A10	Address bit 10	I
9	RD_N	Read	I
10	A11	Address bit 11	I
11	A9	Address bit 9	I
12	A8	Address bit 8	I
13	A13	Address bit 13	I
14	A14	Address bit 14	I
15	WR_N	Write	I
16	DERR_N	Data error	O
17	VCC	Power supply	
18	VPP1	Programmable voltage low byte	
19	A16	Address bit 16	I
20	A15	Address bit 15	I
21	A12	Address bit 12	I
22	A7	Address bit 7	I
23	A6	Address bit 6	I
24	A5	Address bit 5	I
25	A4	Address bit 4	I
26	A3	Address bit 3	I
27	A2	Address bit 2	I
28	A1	Address bit 1	I
29	A0	Address bit 0	I
30	D0	Data bit 0	I / O
31	D1	Data bit 1	I / O
32	D2	Data bit 2	I / O
33	WP	Write protect	O
34	GND	Ground	
35	GND	Ground	
36	CD1/DS_N	Cdetect 1/data save	I / O
37	D11	Data bit 11	I / O
38	D12	Data bit 12	I / O
39	D13	Data bit 13	I / O
40	D14	Data bit 14	I / O

Table 3-16 Pin Assignments of Plug X15 for a Memory Card

Pin	Signal	Function	I/O
41	D15	Data bit 15	I / O
42	CE2_N	Enable high bit	I
43	RFSH	Refresh	I
44	Ubatt	Ubatt external	
45			
46	A17	Address bit 17	I
47	A18	Address bit 18	I
48	A19	Address bit 19	I
49	A20	Address bit 20	I
50	A21	Address bit 21	I
51	VCC	Power supply	
52	VPP2	Programmable voltage high byte	
53	A22	Address bit 22	I
54	A23	Address bit 23	I
55	A24/SEL1	Address bit 24 *	I
56	A25/SEL0	Address bit 25 *	I
57			
58			
59			
60			
61	EKS_N	Enable identifier bit voltage	I
62	BVD2	Battery detect 2	O
63	BVD1	Battery detect 1	O
64	D8	Data bit 8	I / O
65	D9	Data bit 9	I / O
66	D10	Data bit 10	I / O
67	CD2_N	Cdetect	O
68	GND	Ground	

- SEL0 and SEL1 are selection signals for hybrid modules

Leading signal connections:

17, 51 (V_{cc})

1, 34, 35, 68 (ground)

Lagging signal connections:

36 (CD1/DS_N)

67 (CD2)

RGB Video Outputs

You can connect a VGA monitor to the three coax midi sockets on the front panel of the CP 581 basic board via a double-screened coaxial cable (triax cable) up to a distance of 250 m.

The sockets are used for the following signals:

- Red (R),
- Green/sync signal (G/S) and
- Blue (B).

**Caution**

The sockets are not floating.

VGA Video Outputs only for the Pentium CP 581

You can connect a standard VGA monitor to the 15-pin standard VGA plug (X5). To do so, you may use a standard VGA cable with a maximum length of 1.5 m.

Table 3-17 Pin Assignments of the VGA Terminal

Pin	Signal
1	Red
2	Green
3	Blue
4	
5	Ground
6	Ground
7	Ground
8	Ground
9	
10	Ground
11	
12	Monitor ID (DDC cycle)
13	Horizontal synchronization
14	Vertical synchronization
15	Monitor ID (DDC data)

Conversion of RGB Signals to VGA Multisync

Using the RGB/VGA adapter listed in Chapter 5 (Ordering Information), you can convert the RGB signal onto a 15-pin standard VGA plug for laboratory or test purposes.

Note

Note when using this adapter circuit that:

- The noise immunity is reduced
- The monitor can only be installed up to a max. distance of 1.5 m
- It is not sufficient if the monitor is a multisync or multifrequency monitor - it must additionally be able to derive the sync signals from the green signal.

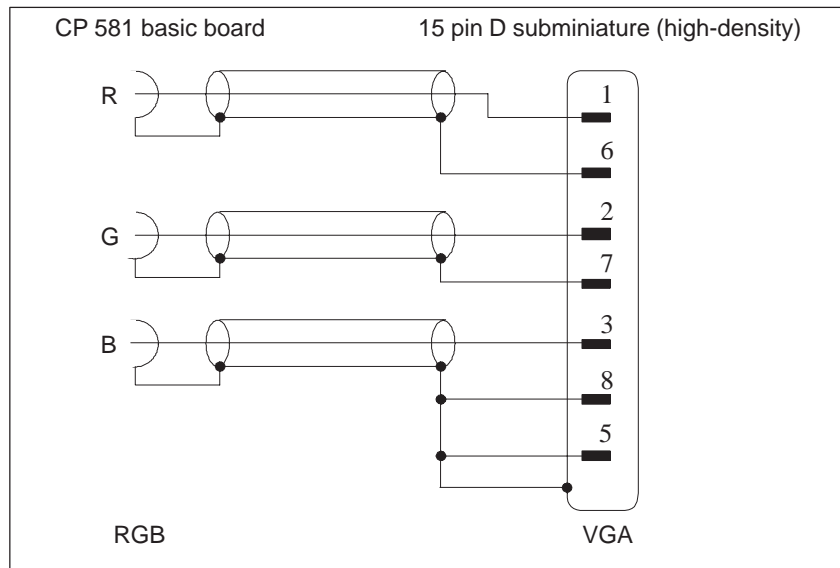


Figure 3-6 Wiring Diagram of RGB/VGA Adapter

3.1.10 Internal Interface Assignments

Backplane Connectors X1 and X2

The CP 581 basic board is connected to the bus board of the programmable controller via two 48-pin backplane connectors of series 2. The pin assignments of these two backplane connectors are shown in the following tables.

Table 3-18 Pin Assignment of Backplane Connector X1 (Top of Basic Board)

	d	b	z
2	-	Ground	+5 V
4	UBATT	-	-
6	ADB 12	ADB 0	-
8	ADB 13	ADB 1	/MEMR
10	ADB 14	ADB 2	/MEMW
12	ADB 15	ADB 3	/RDY
14	IRA_N	ADB 4	DB 0
16	IRB_N	ADB 5	DB 1
18	IRC_N	ADB 6	DB 2
20	IRD_N	ADB 7	DB 3
22	-	ADB 8	DB 4
24	-	ADB 9	DB 5
26	-	ADB 10	DB 6
28	/DSI	ADB 11	DB 7
30	-	-	-
32	-	Ground	-

The bus signals are mainly connected to backplane connector X1, the module power supply is additionally connected via backplane connector X2.

Table 3-19 Pin Assignments of Backplane Connector X2 (Bottom of Basic Board)

	d	b	z
2	-	Ground	-
4	-	-	-
6	-	-	-
8	-	-	-
10	-	-	-
12	-	-	-
14	-	-	/NAU
16	-	-	-
18	-	-	-
20	-	-	-
22	-	-	-
24	-	-	-
26	-	-	-
28	-	-	-
30	-	M2	M2
32	-	Ground	+ 24 V

**Internal,
Expanded ISA
Bus Connector
X10**

Table 3-20 Pin Assignments of ISA Bus Connector X10

Row	A	B	C	D
Pin	Signal			
1	P12V	P5V	NC	P5V
2	P12V	SD15	SD14	GND
3	SD7	SD6	SD13	P5V
4	SD4	SD5	SD12	IDEN_N
5	SD3	SD10	SD11	GND
6	SD2	SD0	SD9	SYSCLK2
7	SD1	GND	SD8	GND
8	IOCHK_N	GND	IOCHRDY	CLISA14M
9	SA0	SA1	AEN	GND
10	SA2	SA3	SA13	
11	SA4	SA5	SA14	
12	SA6	SA7	SA15	
13	SA8	SA9	SA16	GND
14	SA10	SA11	SA17	
15	SA12	SBHE_N	SA18	
16	LA17	LA18	SA19	
17	LA20	LA21	LA19	
18	LA23	RESDRV	LA22	GND
19	IOW_N	BALE	OWS_N	DRQ3
20	IOR_N	GND	SMEMW_N	DACK3_N
21	NC	IRQ 6	SMEMR_N	GND
22	DRQ0	IRQ 12	DACK0_N	IRQ 3
23	IRQ 5	GND	IRQ 9	IRQ 7
24	IRQ 11	SYSCLK	IRQ 10	IRQ 4
25	IRQ 15	GND	IRQ 14	GND
26	TC	DACK1_N	REFRSH_N	1_8432MHz
27	DRQ1	P5V	IOCS16_N	GND
28	DRQ2	DACK2_N	MCS16_N	NAU
29	DRQ5	GND	MASTER_N	P5V
30	DACK5_N	DRQ7	MEMR_N	GND
31	DRQ6	DACK7_N	MEMW_N	1M
32	DACK6_N	P5V	GND	P24V

Plug X99

For internal purposes.
Do not switch!

3.1.11 Memory and Hardware Interrupt Assignments

Memory Assignments

The address areas are divided into:

- The memory address area and
- The I/O address area.

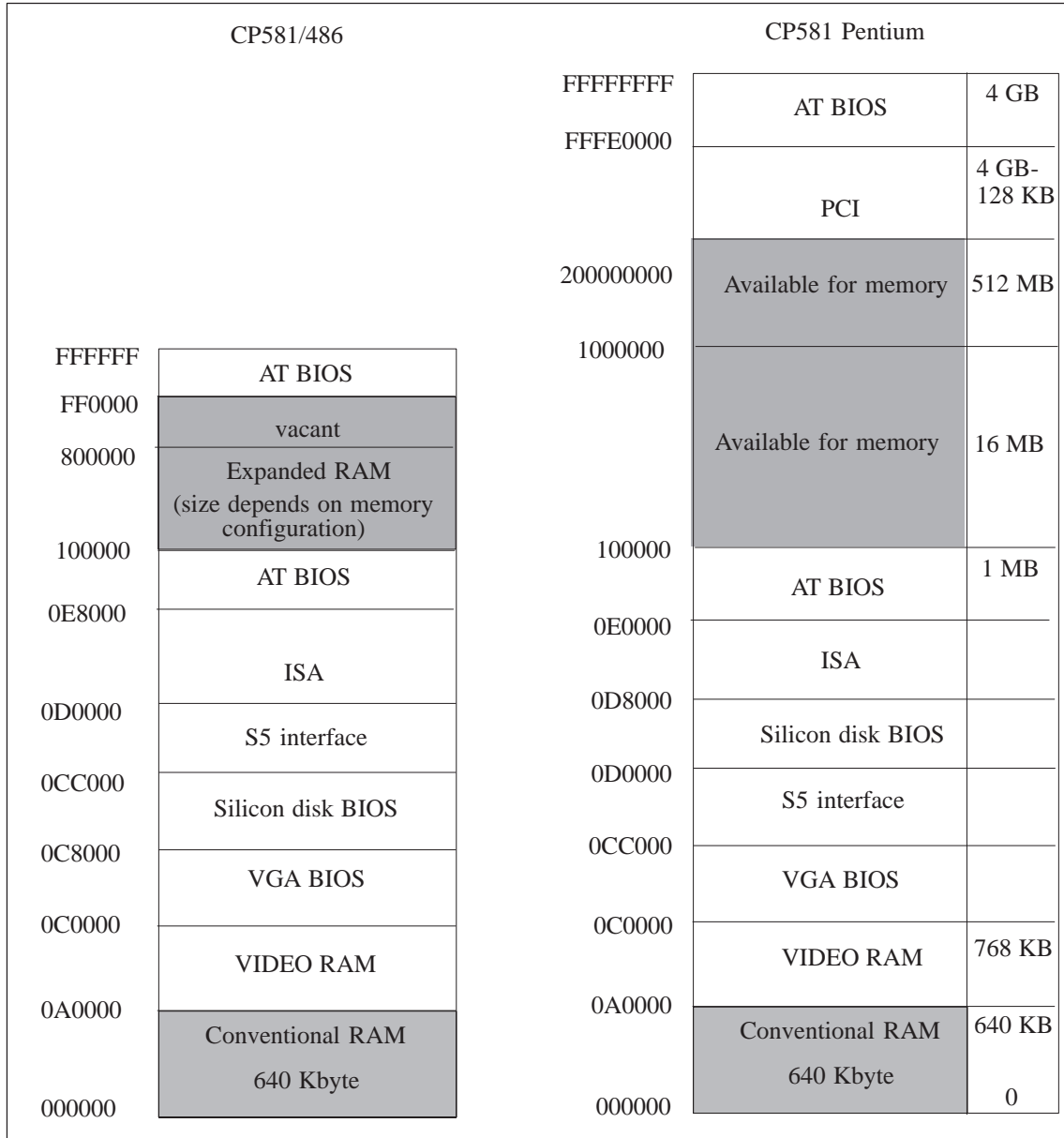


Figure 3-7 Assignment of Memory Address Area

Table 3-21 Assignment of I/O Address Area

Address area	Component/module
0000H-001FH	DMA controller 1
0020H-0021H	Interrupt controller 1
0022H-0003H	Reserved
0040H-005FH	Timer
0060H-0064H	Keyboard controller, port B:, port C, port D
0070FH-007FH	Real-time clock, NMI enable
0080FH-009FH	DMA page register
00A0H-00A1H	Interrupt controller 2
00A2H-00BFH	Reserved
00C0H-00DFH	DMA controller 2
00E0H-00EFH	Reserved
00F0H-00FFH	Numeric processor
0100H-010FH	Reserved for SINEC H1
0110H-014FH	Reserved for CP-58x internal
0150H-016FH	Not used
0170H-0177H	Reserved for hard disk controller 2
0178H-01EFH	Not used
01F0H-01FFH	Hard disk controller 1
0200H-0207H	Reserved (game I/O)
0208H-0277H	Reserved
0278H-027FH	Reserved (parallel interface 2)
0280H-02E7H	Reserved
02E8H-02EFH	Serial interface COM 4, mass storage
02F0-02F7H	Reserved
02F8H-02FFH	Serial interface COM 2, basic board
0300H-031FH	Reserved
0320H-033FH	Not used
0340H-035FH	Reserved
0360H-036FH	Reserved (IBM PC NET)
0370H-0377H	Reserved (floppy controller 2)
0378H-037FH	LPT 1, mass storage
0380H-03AFH	Reserved
03B0H-03BFH	B/W monitor interface or EGA/VGA
03C0H-03CFH	Color graphics interface EGA/VGA
03D0H-03DFH	Color graphics interface CGA/EGA/VGA
03E0H-03E7H	Not used
03E8H-03EFH	Serial interface COM 3, basic board
03F0H-03F7H	Floppy controller
03F8H-03FFH	Serial interface COM 1, basic board
0400H-4FFFH	Reserved
5000H-55FFH	S5 interface and silicon disk interface for the CP 581
5600H-FFFFH	Reserved

**Hardware
Interrupts**

Table 3-22 Summary of Hardware Interrupts

Interrupt number	Meaning
NMI	RAM parity, AT bus error (IOCHCHK), RESET key, power failure
IRQ 0	System timer
IRQ 1	Keyboard
IRQ 2	Interrupt controller 2
IRQ 3	COM 2
IRQ 4	COM 1 (COM 3)
IRQ 5	Not used (COM 3, COM 4)
IRQ 6	Floppy disk drive
IRQ 7	LPT
IRQ 8	Real-time clock
IRQ 9	Software interrupt derived from 0AH (IRQ 2)
IRQ 10	Not used (COM 3, COM 4)
IRQ 11	COM 3 (COM 4)
IRQ 12	COM 4
IRQ 13	Numeric processor
IRQ 14	Hard disk
IRQ 15	S5 interface

Interrupts 4, 5, 10 and 11 can be reconfigured using switch S3 for the CP 581 80486 DX and using switch S2 for the CP 581 Pentium.

3.2 Mass Storage Module

3.2.1 Mechanical Construction

The CP 581 mass storage module (printed circuit board) is of double Eurocard format and thus suitable for the ES 902 modular packaging system. The front panel width, including the integrated floppy disk and hard disk drives, is 1 1/3 standard slots (approx. 20 mm) wide and thus occupies one S5 slot in the main frame of the programmable controller.

The front panel contains the BUSY LEDs for the hard disk and floppy disk drives, the reject key of the 3.5-inch floppy disk drive and interfaces to the peripheral devices. Figure 3-8 shows you the design of the CP 581 mass storage module.

The mass storage module comprises:

- Two 48-pin backplane connectors according to DIN 41 612, series 2, for connecting the CP 581 mass storage module to the S5 bus (backplane bus) of the programmable controllers (backplane connectors 1 and 2)
- One 25-pin D-subminiature plug (LPT) for connecting a dongle or a printer with parallel interface
- One 26-pin high-density plug for the parameterizable serial interface COM 4
- One 128-pin plug connector for connecting the basic board (internal ISA bus)
- One LED "HD"
- One hard disk (2.5-inch) with IDE interface
- One 3.5-inch slimline floppy disk drive.

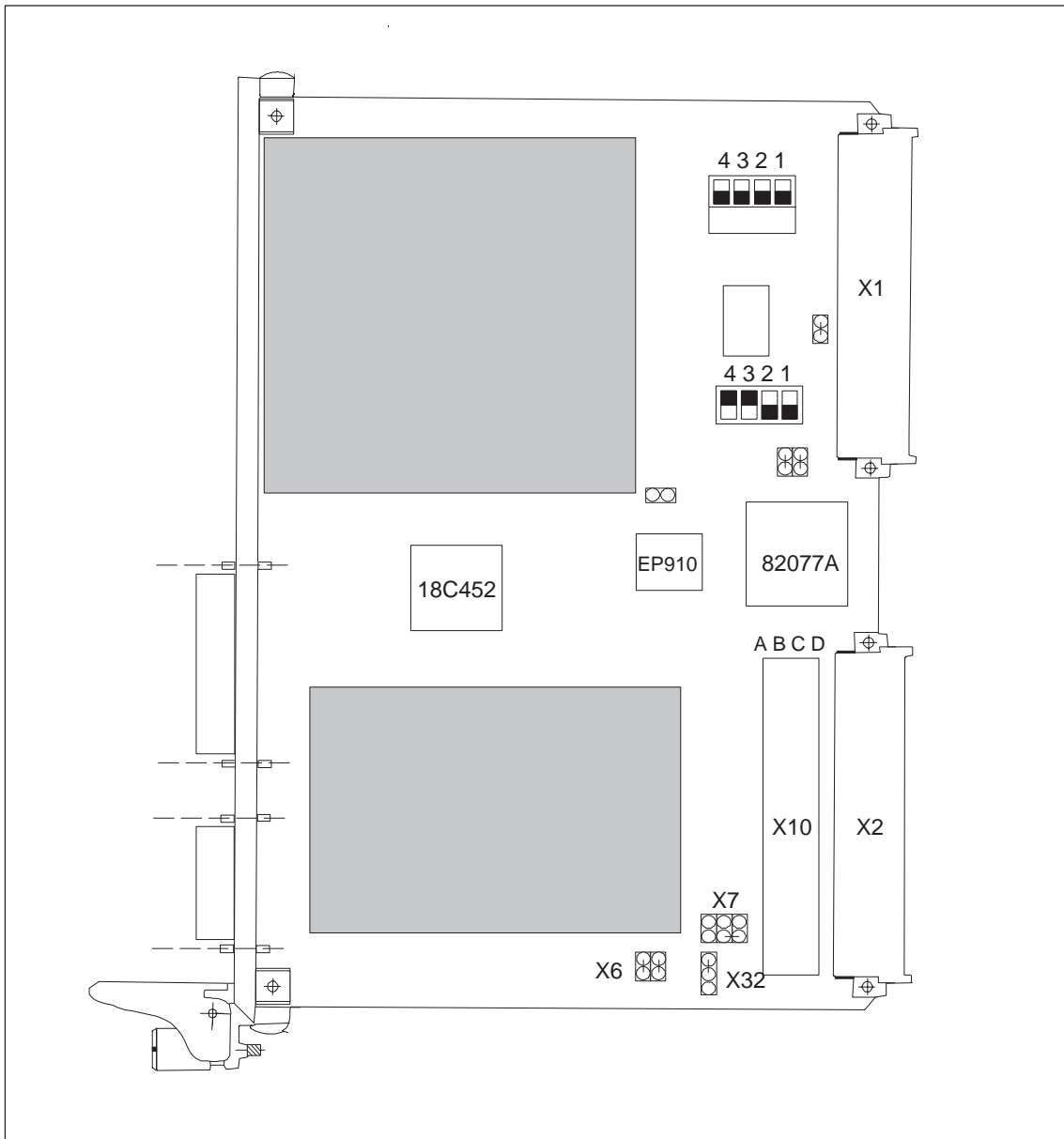


Figure 3-8 Design of CP 581 Mass Storage Module

3.2.2 Positions of Floppy Disk Drive and Interfaces on Front Panel

Figure 3-9 shows you the positions of the floppy disk drive and the interfaces on the front panel of the mass storage module.

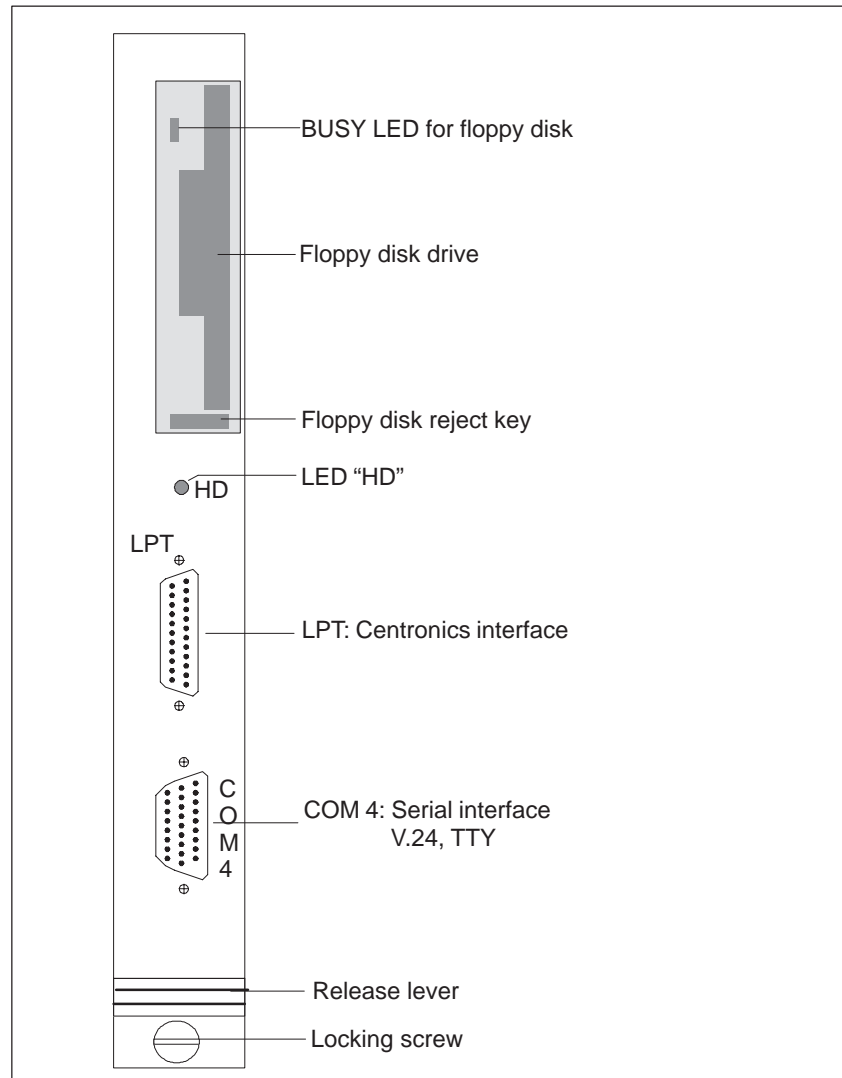


Figure 3-9 Front Panel of Mass Storage Module

3.2.3 Floppy disk drive

The floppy disk must not be removed while the mode LED on the disk drive is lit.

The 3.5-inch floppy disk drive of the CP 581 mass storage module is connected to the disk interface of the 37C65 floppy controller via a 26-pin connector. Double-sided 3.5-inch floppy disks can be used (80 tracks per side), either high density with 1.44 Mbytes or double density with 720 Kbytes. The drive is automatically adapted to the type of disk by scanning the HD opening in the disk.

3.2.4 Hard Disk Drive

The hard disk drive automatically moves to the transport position when switched off.

The 2.5-inch hard disk drive of the CP 581 mass storage module is connected to the AT bus via a 50-pin plug connector. The hard disk is equipped with a PC/AT interface.

The total capacity of the hard disk of version -4LA11 is 1.4 Gbytes.

Logical drives can be produced depending on the operating system. Please refer to the description of the operating system for more information.

The LED "HD" lights up on the front panel of the CP 581 mass storage module when the hard disk is being accessed.

3.2.5 Switch and Jumper Settings

The coding switches and plug-in jumpers present on the mass storage module have been factory-set. You need not check or reset them provided you wish to use the CP 581 components in their default configuration.

The following sections provide you with further information on the coding switches and plug-in jumpers. Figure 3-8 shows the positions of the switches and jumpers.

Switch S101

Table 3-23 Default Setting of Switch S 101

Switch	Default		ON	OFF
	ON	OFF		
1	X		Spare	
2	X		COM 4 switched on	COM 4 switched off
3		X	Spare	
4		X	Spare	

Plug-in jumper X6

Table 3-24 Default Setting of Plug-in Jumper X6

PIN _x - PIN _y	Effect
1 - 2	“TTY transmit” active (jumper inserted on delivery)
3 - 4	“TTY receive” active (jumper inserted on delivery)
No jumper	TTY passive

Plug-in jumper X7

Table 3-25 Default Setting of Plug-in Jumper X7

PIN _x - PIN _y	Effect
1 - 2	COM 4 interrupt at IRQ 5
5 - 3	COM 4 interrupt at IRQ 11
2 - 4	COM 4 interrupt at IRQ 12 (default setting)
5 - 6	COM 4 interrupt at IRQ 10
No jumper	COM 4 interrupt not connected

Plug-in jumper X32

Table 3-26 Default Setting of Plug-in Jumper X32

PIN _x - PIN _y	Effect
1 - 2	LPT interrupt at IRQ 7 (default setting)
2 - 3	LPT interrupt at IRQ 5
No jumper	LPT interrupt not connected

3.2.6 Connection of Devices to Centronics Interface (LPT)

The printers recommended in Section 3.1.4 can be connected to the Centronics interface.

Please refer to Chapter 5, Ordering Information, for the order numbers of devices for connection to the Centronics interface and of suitable cable connectors.

Please note that you must order the cable connectors separately.

3.2.7 Connection of Devices to the Serial Interface COM 4

You can connect external devices with a serial interface to COM 4. Proceed as described in Section 3.1.4.

The Y adapter for COM 1/COM 3 can also be used for COM 4. In this case the “COM 1 side” of the Y adapter is a standard COM 4 interface.

3.2.8 External Interface Assignments of Mass Storage Module

Parallel Interface (LPT1)

The parallel interface LPT (Centronics) has a 25-pin D-subminiature plug. It is used to connect a printer up to 3 m away or for connecting a dongle for protected programs.

Table 3-27 Signal Assignment of the Parallel Interface LPT (Centronics)

Pin	Signal	Signal class I/O
1	STB_N	O
2	PDB0	O (data)
3	PDB1	O (data)
4	PDB2	O (data)
5	PDB3	O (data)
6	PDB4	O (data)
7	PDB5	O (data)
8	PDB6	O (data)
9	PDB7	O (data)
10	ACK_N	I
11	BUSY	I
12	PE	I
13	SLC_IN	I
14	AFD_N	O
15	ERR_N	O
16	INIT_N	I
17	SLIN	O
18-25	GND	Ground

3.2.9 Serial Interface COM 4

The plug of interface COM 4 is designed as a 26-pin high-density plug.

The serial interface COM 4 has the standard assignments of the V.24 transmission signals plus the signals for active TTY operation (20 mA).

Table 3-28 Signal Assignments of the Serial Interface COM 4

Pin	Signal	Signal class
1	RI	COM 4 V.24
2	DCD	COM 4 V.24
3	RTS	COM 4 V.24
4	Not used	
5	Not used	
6	Not used	
7	Not used	
8	Not used	
9	Not used	
10	DSR	COM 4 V.24
11	RxD	COM 4 V.24
12	GND	COM 4 V.24
13	P24V	COM 4 TTY
14	RxD+	COM 4 TTY
15	TxD+	COM 4 TTY
16	Not used	
17	Not used	
18	GND24V	
19	TxD	COM 4 V.24
20	DTR	COM 4 V.24
21	CTS	COM 4 V.24
22	RxD-	COM 4 TTY
23	TxD-	COM 4 TTY
24	Not used	
25	Not used	
26	Not used	

3.2.10 Internal Interface Assignments

Backplane Connectors X1 and X2

The CP 581 mass storage module is connected to the bus board of the programmable controller via two 48-pin backplane connectors of series 2. The pin assignments of these two backplane connectors are shown in the following tables

Table 3-29 Pin Assignments of Backplane Connector X1 (top of mass storage module)

	d	b	z
2	-	Ground	+5 V
4	-	-	-
6	-	-	-
8	-	-	-
10	-	-	-
12	-	-	-
14	-	-	-
16	-	-	-
18	-	-	-
20	-	-	-
22	-	-	-
24	-	-	-
26	-	-	-
28	-	-	-
30	-	-	-
32	-	Ground	-

Table 3-30 Pin Assignments of Backplane Connector X2 (bottom of mass storage module)

	d	b	z
2	-	Ground	+5 V
4	-	-	-
6	Ground	-	-
8	-	-	-
10	-	-	-
12	-	-	-
14	-	-	-
16	-	-	-
18	-	-	-
20	-	-	-
22	-	-	-
24	-	-	-
26	-	-	-
28	-	-	-
30	-	M2	M2
32	-	Ground	-

Internal, Expanded ISA Bus Connector X10

The mass storage module is connected to the basic board via the 128-pin plug X10. Please refer to Section 3.1.10 for the pin assignments of this plug.

3.3 AT Slot Module

The CP 581 AT slot module does not have a direct connection to the S5 bus (backplane bus) of a programmable controller and does not have a backplane connector. The front panel width is 1 1/3 standard slots (approx. 20 mm).

The following AT slot module version is available:

6ES5 581-0RA12 for CP 581

3.3.1 Mechanical Construction of AT Slot Module

The AT slot module has:

- A 128-pin connector for connection to a basic board, mass storage module or AT slot module
- An AT bus plug for connection of a short AT bus expansion module.

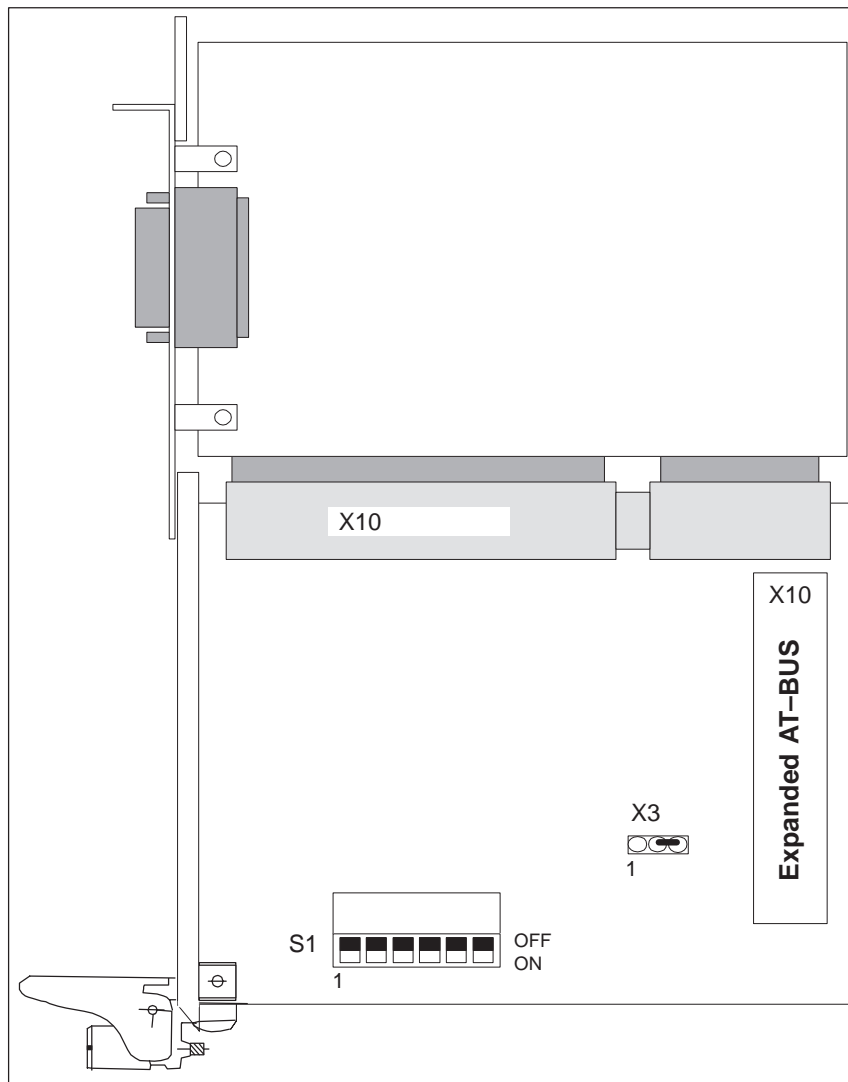


Figure 3-10 Design of CP 581 AT Slot Module (with AT Extension Card)

3.3.2 Switch and Jumper Settings

The coding switches and plug-in jumpers present on the AT slot module have been factory set. Refer to Figure 3-10 for the positions of the switches and jumpers.

Table 3-31 AT Slot Jumper Setting/Switch S1 (see Figure 3-10)

Switch	Default		ON	OFF
	ON	OFF		
1		X	I/O range > 32k disabled with 6ES5 581 0RA12.	No effect
2		X	Not allowed	
3		X	Not allowed	
4		X	Not allowed	
5		X	Not allowed	
6		X	Not allowed	

3.3.3 External Interface Assignments

The AT slot module has a 98-pin standard direct plug for short AT cards (see Figure 3-11). It has no interfaces on the front panel.

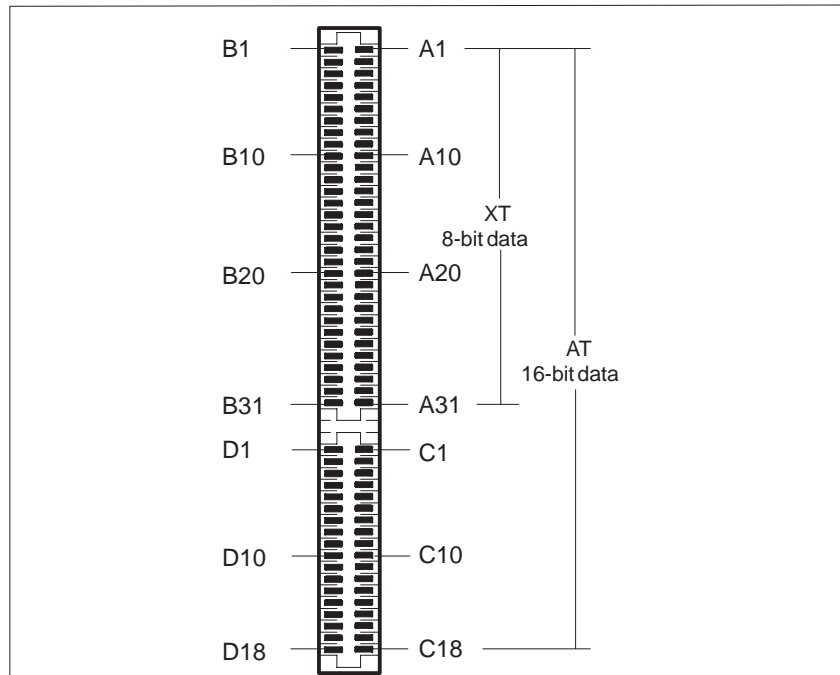


Figure 3-11 98-Pin Standard Direct Plug

The pin assignments of the external interface are shown in the following table.

Table 3-32 Pin Assignments of External Interface

Pin	Signal name	PIN	Signal name
XT signals			
B1	GND	A1	I/OCHCK_N
B2	RESET DR	A2	SD7
B3	P5V	A3	SD6
B4	IRQ 9	A4	SD5
B5	N5V	A5	SD4
B6	DRQ2	A6	SD3
B7	N12V	A7	SD2
B8	0WS	A8	SD1
B9	P12V	A9	SD0
B10	GND	A10	I/OCHRDY_N
B11	SMEMW_N	A11	AEN
B12	SMEMR_N	A12	SA19
B13	I/OW_N	A13	SA18
B14	I/OR_N	A14	SA17
B15	DACK3_N	A15	SA16
B16	DRQ3	A16	SA15
B17	DACK1_N	A17	SA14
B18	DRQ1	A18	SA13
B19	REFRESH	A19	SA12
B20	SYSCLK	A20	SA11
B21	IRQ 7	A21	SA10
B22	IRQ 6	A22	SA9
B23	IRQ 5	A23	SA8
B24	IRQ 4	A24	SA7
B25	IRQ 3	A25	SA6
B26	DACK2_N	A26	SA5
B27	T/C	A27	SA4
B28	BALE	A28	SA3
B29	P5V	A29	SA2
B30	OSC	A30	SA1
B31	GND	A31	SA0

Table 3-32 Pin Assignments of External Interface

Pin	Signal name	PIN	Signal name
AT extension			
D1	MEMCS_16_N	C1	SBHE_N
D2	I/OCS16_N	C2	LA23
D3	IRQ 10	C3	LA22
D4	IRQ 11	C4	LA21
D5	IRQ 12	C5	LA20
D6	IRQ 15	C6	LA19
D7	IRQ 14	C7	LA18
D8	DACK0_N	C8	LA17
D9	DRQ0	C9	MEMR_N
D10	DACK5_N	C10	MEMW_N
D11	DRQ5	C11	SD8
D12	DACK6_N	C12	SD9
D13	DRQ6	C13	SD10
D14	DACK7_N	C14	SD11
D15	DRQ7	C15	SD12
D16	P5V	C16	SD13
D17	MASTER_N	C17	SD14
D18	GND	C18	SD15

3.3.4 Internal Interface Assignments

The connection between the AT slot module and the basic board, mass storage module or AT slot module is made via the ISA bus connector X10.

Please refer to Section 3.1.10 for the pin assignments of this plug.

3.4 Remote Terminal Interface

The remote terminal interface (RTI) enables you to use a keyboard and a mouse up to 250 m away from the CP 581.

This is necessary if the CP 581 is fitted in an instrument cabinet or if the workstation is located more than 1.5 m away from the CP 581 for other reasons.

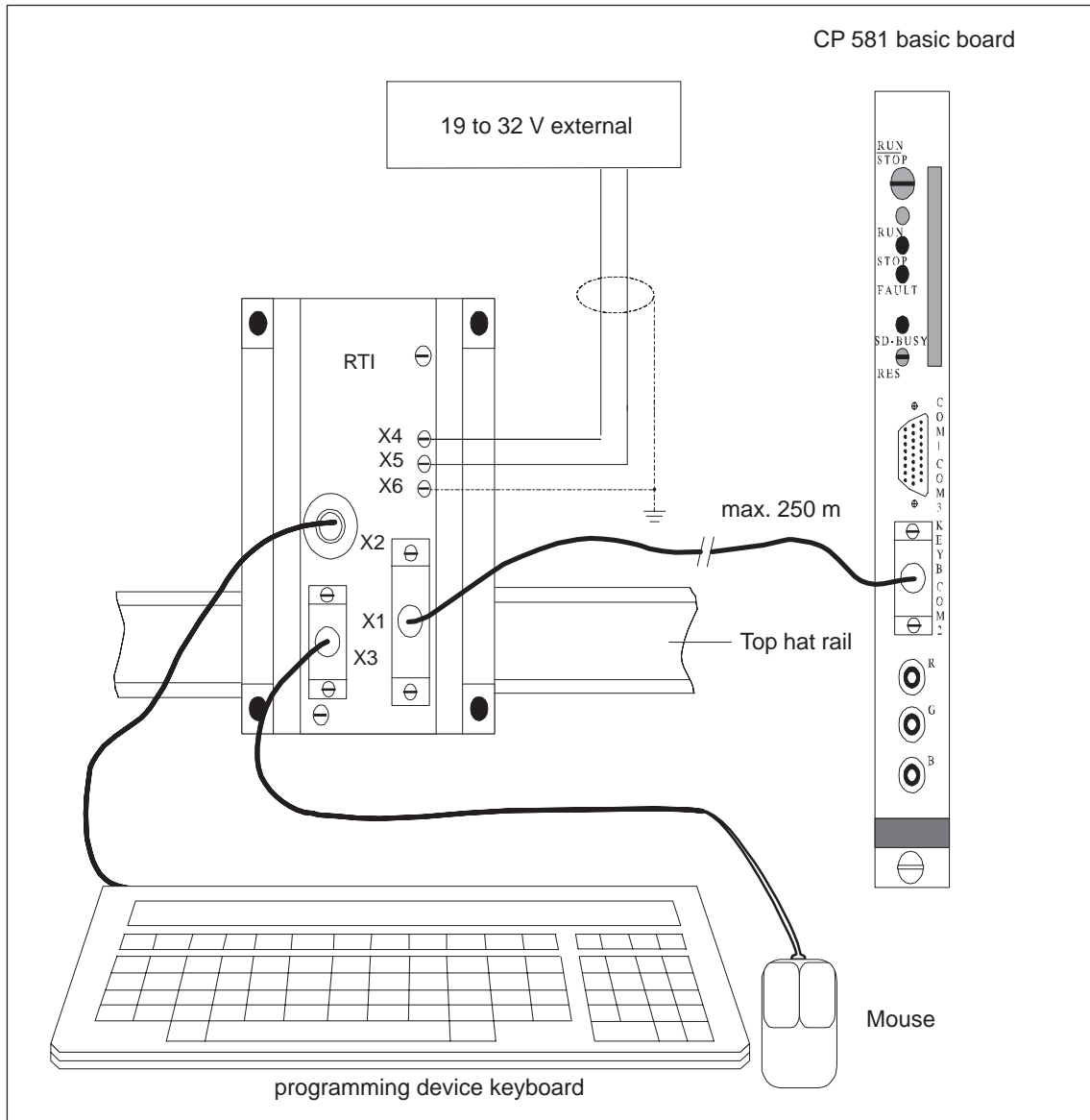


Figure 3-12 CP 581 Configuration with Remote Terminal Interface with Mouse

Figure 3-12 shows you how the RTI with mouse is connected to a CP 581 basic board and the components required. These are:

- Data and power supply lines
- RTI module
- Keyboard
- Serial V.24 mouse

The power supply is connected:

- For short distances (≤ 50 m) via the data line from the CP 581
- For long distances (> 50 m) by a local, external supply (DC 19 to 32 V).

3.4.1 Power Supply of RTI Via Remote Supply From CP 581

The voltage for the remote supply is derived from the 24 V supply of the central controller. The RTI has a floating DC/DC converter. Do not exceed the permissible current loading for the 24 V supply of the central controller.

3.4.2 Power Supply of RTI by Local, External Supply

You must provide a 24 V supply for the RTI if the distance is >50 m or if the current loading of the central controller is too high. Connect the voltage source to the provided pins X4 and X5. The polarity must be correct (see Table 3-33).



Caution

Only an extra-low voltage of ≤ 60 V DC safely isolated from the supply voltage may be used for the 24 V DC power supply. The safe isolation can be applied according to the requirements in, among others, VDE 0100 part 410/HD 384-4-41/IEC 60364-4-41 (as functional extra-low voltage with safe isolation) or in VDE 0805/EN 60950/IEC 60950 (as a safety extra-low voltage, or 'SELV') or VDE 0106 part 101.

3.4.3 Connection of Keyboard to the RTI

Connect the keyboard to the X2 interface (see Figure 3-12).

3.4.4 Connection of a Mouse to the RTI

Connect the mouse to the X3 interface (see Figure 3-12).

Note

Note when using the programming device keyboard
Always connect the mouse to the interface X3.
Connecting the mouse to the programming device keyboard when using the RTI is **not allowed!**

3.4.5 Jumper Assignments on the RTI Module

The following figure shows the internal design of the RTI module, including the positions of the jumpers and plugs.

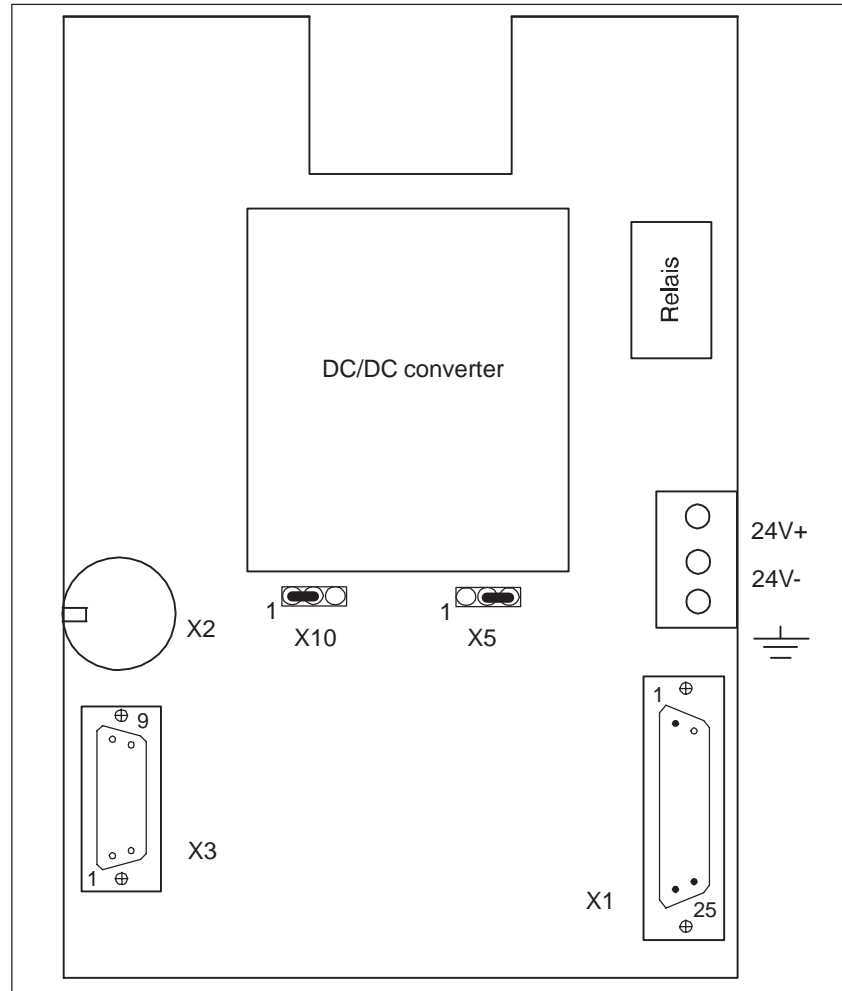


Figure 3-13 Internal Design of RTI Module

- Plug-in jumper X10

PINx - PINy	Effect
1 - 2	IBM compatible keyboard (default setting)
2 - 3	Keyboard

- Plug-in jumper X5

PINx - PINy	Effect
1 - 2	Light pen mode ¹⁾
2 - 3	Mouse mode (default setting)

¹⁾ Only with the CP 581 486 DX

3.4.6 Interface and Pin Assignments on the RTI

Table 3-33 RTI Interface and Pin Assignments

Plug	Plug type	Function
X1	D25 subminiature socket	Data and power supply interface to the CP 581
X2	7-pin DIN socket	Connection of Keyboard
X3	D9 subminiature socket	Mouse connection, V.24/COM 2
X4	Screw terminal	Optional supply, +19 to +32 V (pos.)
X5	Screw terminal	Optional supply, -0 V (neg.)
X6	Screw terminal	Equipotential bonding terminal/screen

3.4.7 CP 581 <-> RTI Connection Cable

The connection cable has the following two functions:

- Function 1:
RTI data and power supply cable, connection to KEYB/COM 2 of the CP 581 basic board.
- Function 2:
Printer cable V.24, connection to COM 1/COM 3 of the CP 581 basic board or COM 4 of the CP 581 mass storage module and printer with V.24 interface.

Note

The connections of your printer may differ from the signal assignments of the cable. Compare your printer connections with the assignments of the RTI cable (see tables below). You will find the required information in your printer manual.

**CP 581 <=> RTI
Connection
Cable**

Table 3-34 Signal Assignments of the CP 581 <=> RTI Connection Cable

CP 581, 26-pin high-density plug	Signal	RTI, D25 subminiature plug	Remarks
10	XDATA	20	Twisted pair of conductors
19	XDATA_N	03	
11	XTAKT	02	Twisted pair of conductors
20	XTAKT_N	06	
12	XGND	07	Twisted pair of conductors
21	XGND	25	
22	XRXD	18	Twisted pair of conductors
23	XRXD_N	9	
14	Not used	21	Twisted pair of conductors
15	Not used	10	
24	XTXD	14	Twisted pair of conductors
25	XTXD_N	15	
26	P24V	16	Twisted pair of conductors
18	M24V	17	
Housing	SCREEN	01	
8 ¹⁾	XRTS_N	23	Twisted pair of conductors
9 ¹⁾	XRTS_N	24	

¹⁾ with RTI cable 6ES5 714-1xxx1 not used

**CP 581 <=>
Printer, V.24
Connection
Cable**

Table 3-35 Signal Assignments of the CP 581 <=> Printer, V.24 Connection Cable

CP 581, 26-pin high-density plug	Signal		Printer, D25 subminiature plug	Remarks
10	DSR	DTR	20	Twisted pair of conductors
19	TXD	RXD	03	
11	RXD	TXD	02	Twisted pair of conductors
20	DTR	DSR	06	
12	GND		07	Twisted pair of conductors
21	CTS	BUSY	25	
22	Not used		18	
23	Not used		9	
14	Not used		21	
15	Not used		10	
24	Not used		14	
25	Not used		15	
26	Not used		16	
18	Not used		17	
Housing	Screen		01	
8	Not used		23	
9	Not used		24	

3.5 Y Adapter

3.5.1 Assignments of Y Adapter Keyboard/COM 2

Table 3-36 Y Adapter Pin Assignments

CP 581, 26-pin high-density plug	Signal	Signal class	Keyboard 7-pin round socket	Mouse, D9 subminiature plug
1	K_DATA	Keyboard, direct	2	-
2	K_CLK	Keyboard, direct	1	-
3	P5V	Keyboard, direct	5	-
4	TxD	COM 2, V24	6	3
5	RTS	COM 2, V24	-	7
6	DTR	COM 2, V24	-	4
12	GND		4	5
13	RxD	COM 2, V24	7	2
14	CTS	COM 2, V24	-	8
15	DSR/Detect	COM 2, V24	-	6
16	RI/5 V	COM 2, V24	-	9
17	DCD/Touch	COM 2, V24	-	1
21	GND		3	-

3.5.2 Assignments of Y Adapter COM 1/COM 3

Table 3-37 Assignments of Y Adapter COM 1/COM 3

CP 581, 26-pin high-density plug	Signal	Signal class	D25 subminiature plug COM 1	D25-subminiature plug COM 3
1	RI	COM 1, V24	22	-
2	DCD	COM 1, V24	8	-
3	RTS	COM 1, V24	4	-
4	RxD+	COM 3, X27	-	11
5	RxD-	COM 3, X27	-	13
6	TxD+	COM 3, X27	-	12
7	TxD-	COM 3, X27	-	14
8	RTS	COM 3, V24	-	4
9	CTS	COM 3, V24	-	5
10	DSR	COM 1, V24	6	-
11	RxD	COM 1, V24	3	-
12	GND	COM 1, V24	7	-
13	P24V	COM 1, TTY	19	-
14	RxD+	COM 1, TTY	9	-
15	TxD+	COM 1, TTY	18	-
16	DCD	COM 3, V24	-	8
17	DSR	COM 3, V24	-	6
18	GND	COM 3, V24	-	7
19	TxD	COM 1, V24	2	-
20	DTR	COM 1, V24	20	-
21	CTS	COM 1, V24	5	-
22	RxD-	COM 1, TTY	10	-
23	TxD-	COM 1, TTY	21	-
24	TxD	COM 3, V24	-	2
25	RxD	COM 3, V24	-	3
26	DTR	COM 3, V24	-	20

3.6 Silicon Disk

Semiconductor Memories

The CP 581 basic board enables you to use semiconductor memories such as a floppy disk or hard disk drive. Flash EPROMs are used as semiconductor memories which you can apply as follows:

- As 4-Mbyte onboard silicon disk:
This silicon disk with flash EPROMs simulates a hard disk drive from which MS-DOS can be booted and on which user programs can be stored. When using a CP 581 basic board with onboard silicon disk, it is possible to operate it without a mass storage module. This does not require a memory card.
- As memory card:
These memory cards simulate a floppy disk drive from which the MS-DOS operating system can also be booted. They can additionally be used for interchanging user software and user data. You can use the memory card without a mass storage module. This does not require an onboard silicon disk.
Memory cards are available either with flash EPROMs or with RAMs (see Chapter 5, Ordering Information).

Note

Note that when using a RAM card, the data you have saved on the card will be lost as soon as you remove the RAM card from the CP 581 or remove the CP 581 from the PLC rack.

3.6.1 Drive Assignments of Silicon Disks (OSD and MC)

The onboard silicon disk and the memory card are addressed by MS-DOS like conventional drives where the onboard silicon disk simulates a hard disk drive and the memory card a floppy disk drive.

When configuring the CP 581 with a basic board and a mass storage module, the BIOS of the CP 581 manages up to 4 physical drives. The logical drive designations are then as follows:

Drive designation	Data medium
A:	Floppy disk drive of mass storage module
B:	Memory card of basic board
C:	Hard disk of mass storage module
D:	Onboard silicon disk of basic board

The following drive designations result when operating with the CP 581 basic board without a mass storage module:

Drive designation	Data medium
A:	Memory card of basic board
C:	Onboard silicon disk of basic board

3.6.2 Formatting of Silicon Disks

You must format the onboard silicon disk and the memory card just like conventional drives using MS-DOS commands.

Formatting the Memory Card

Before running up the CP 581, insert the memory card which is to be formatted. You can format the memory card using the following MS-DOS command:

FORMAT x:/U

You must enter “A” or “B” for “x” according to the settings in SETUP (see Section 3.7). You can also use lower-case letters in the commands.

If the memory card is to be bootable, you must **additionally** enter the option **'/S'** in the format command.

Note

When running up the CP 581, the memory card which is to be formatted or another card of the same type must be inserted.

Formatting of Onboard Silicon Disk

Proceed as follows when formatting the onboard silicon disk:

1. Create a partition on the onboard silicon disk using the MS-DOS command “FDISK.”
2. Format the created partition using the following MS-DOS command:

FORMAT x: /U

You must enter “C” or “D” for “x” according to the settings in SETUP (see Section 3.7). You can also use lower case letters in the commands.

If the partition is to be bootable, you must **additionally** enter the option **'/S'** in the format command.

3. If you wish to make the partition bootable, you must activate it by means of the MS-DOS command “FDISK” following the formatting.

Note

If you wish to boot from the onboard silicon disks, you must remove a hard disk on the mass storage module in the SETUP menu (see Section 3.7) before activating the corresponding partition, and reboot from the floppy disk drive or memory card.

You must not use the MS-DOS commands “MIRROR” and “UNFORMAT” on the silicon disks since the status prior to formatting cannot then be recreated.

3.7 BIOS Setup for CP 581 80486 DX

Switching on the PLC / Cold Restart

When you switch on the PLC or cold boot the CP 581, the BIOS starts a “Power On Self Test” (POST) and outputs the results in the POST window.

The following screen appears:

Power BIOS Version 1.00
 Copyright (C) 1984-1995, Award Software, Inc.
 ??/??/?? - Picopower - Redwood - 314Q4000000

SIEMENS SIMATIC CP 581 DX? Version 2.4 (VGA CL-GD5429)

CPU Type	AM486DX5xxxMHz
Math CoProcessor.	Internal
Base Memory.	640 Kb
Extended Memory	3072 Kb
BIOSShadowRAM	Enabled
VideoShadowRAM	Enabled
FloppyDrives	1 Found
HardDrives	xxx ¹⁾
SerialPorts	4 Found
ParallelPorts	1 Found
TestingBaseMemory	640 Kb
TestingExtendedMemory	xxKb ²⁾

Press CTRL-ALT-ESC to enter setup

1)2) xxx = depends on the type of hard drive
 depends on the memory configuration

Warm Restart

When you warm restart the CP 581, the following screen appears:

PowerBIOS Version 1.00
 Copyright (C) 1984-1995, Award Software, Inc.
 ??/??/?? - Picopower - Redwood - 314Q4000000

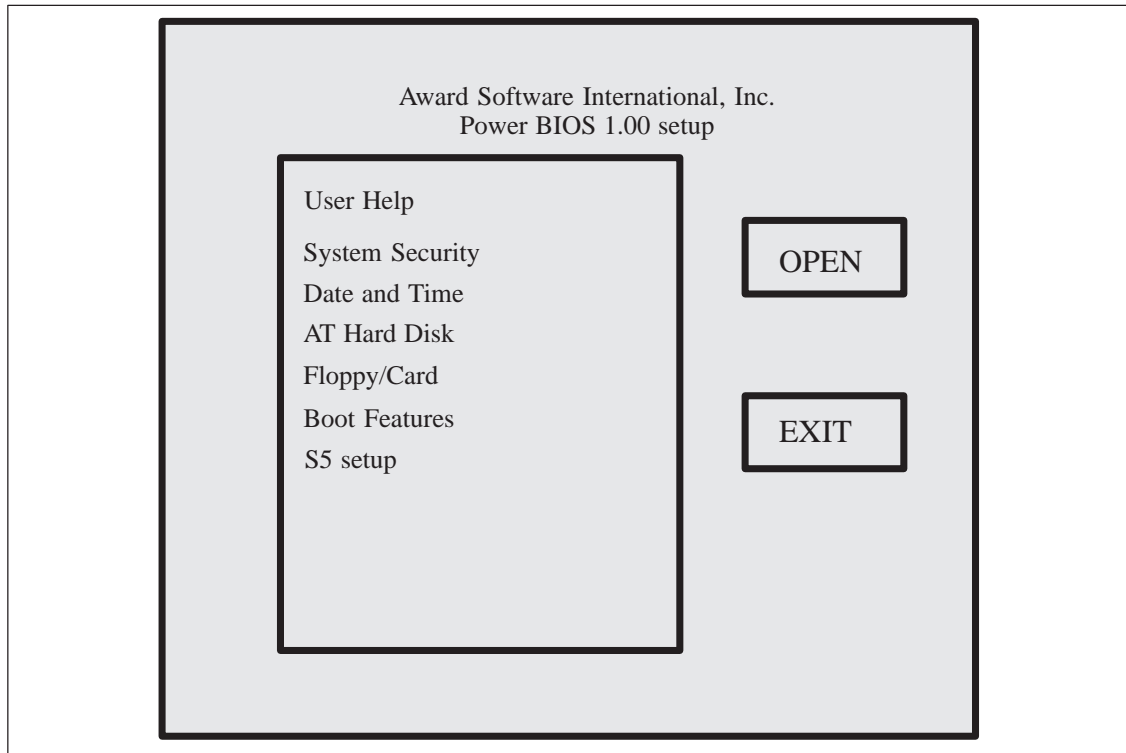
Press CTRL-ALT-ESC to enter setup

Changing the Set BIOS Parameters

If you want to change the set BIOS parameters, you must activate the setup selections page immediately after one of the previous screens by pressing the following key combination:

CTRL and **ALT** and **ESC**

The BIOS setup selection page is then displayed (see the following figure).



Using the cursor keys

↑ and ↓ (“cursor up” or “cursor down”)

select the setup page in which you want to change parameters and confirm your selection with RETURN. The selected setup page then appears on the screen.

Use the following keys or key combinations to select and change individual setup parameters:

- Tab key ⇨ :

With this key you position the cursor on the next option (= field with border) or on the next parameter within an option.

- Key combination <SHIFT> + ⇨ :

With this key you position the cursor on the previous option (= field with border) or on the previous parameter within an option.

- Cursor keys ↑ and ↓ (“cursor up” and “cursor down”):
With these keys you change the settings of a parameter: either the numerical value (for example, the hours for the time of the day) or another alternative for a parameter (for example, the parameter “Mode” in the option PAGE FRAMES in the S5 setup).
- Spacebar:
With the spacebar you select one or more parameters in a parameter group. The individual parameters in a group are identified by “[].” You can recognize whether a parameter is selected or not by means of the check mark ‘✓’ in the square brackets.
- RETURN key :
With this key you confirm that an option has been selected.
- ESC key:
When you have selected a setup page, you can return to the selection menu at any time using this key.

Setup Pages

This section describes the individual setup pages which you can display via the selection menu in order to change parameters.

- User help:

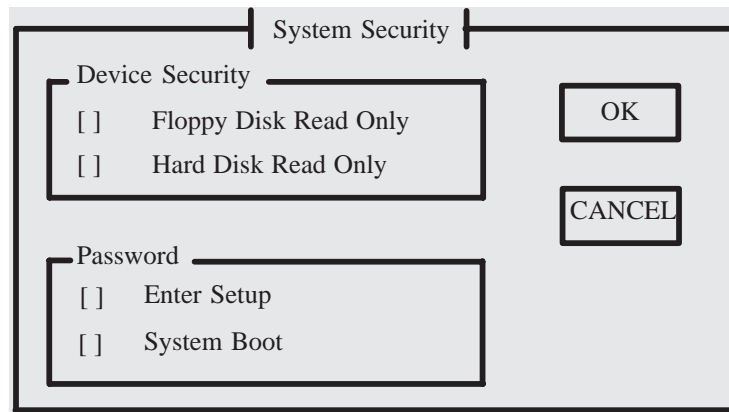
User Help

Use TAB / SHFT-TAB Keys to toggle through items

OK	Press ENTER to save changes and close page
CANCEL	Select and press ENTER or press ESC to close page without saving
DEF	Select and press ENTER to load default values in page

Checkbox	Radio Button	List Box	Edit Box
<div style="text-align: center;">[] Option</div> Use SPACE Bar to enable / disable checkbox option	<div style="text-align: center;"> (*) Option 1 () Option 2 () Option 3 </div> Use Cursor Keys to select an option	<div style="border: 1px solid black; padding: 5px; text-align: center;"> Option 1 Option 2 Option 3 </div>	<div style="border: 1px solid black; padding: 5px; text-align: center; width: 50px;">0</div> Use Cursor Keys or type in the number or string

- System security:



Parameter group “Password”:

“Enter Setup:”

You can protect the setup settings by entering a password under the option “Enter Setup.”

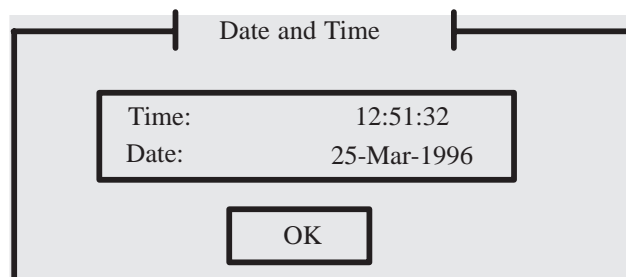
If setup recognizes that a password has been entered when it starts up, it will only allow the date and time to be changed if an incorrect password is entered.

“System Boot”:

You can also enter a second password under the option “System Boot” to protect the CP 581 DX4 startup. The requirement for this is that you have assigned a password for “Enter Setup.”

If setup recognizes that a password has been entered for booting, you can only boot the CP if you enter the correct password. Setup allows three attempts at entering the password. If you enter an incorrect password three times, you can only attempt it again after switching the mains power off and on again.

- Date and Time :



Using the “Date and Time” page you can set the current date and time.

- AT Hard Disk:

AT Hard Disk
Drive Wait Time: 8 sec

AUTO All Drives
OK
CANCEL

PrimaryIDEController										Trans	Block	32-	Fast
	Auto	Type	Cyl	Heads	SPT	Precomp	L-Zone	Size	Mode	Mode	Bit	PIO	
Master:	<input checked="" type="checkbox"/>	USR1	xxx	xx	xx	NONE	xxxx	xxx	LBA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Slave													
Slave:	<input type="checkbox"/>	NONE	0	0	0	0	0	0	Normal	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Using the “AT Hard Disk” you can set the hard disk parameters for drive C:.

As default, the option “Auto Ide Detection” is switched on (“Auto” = “[]”). This option determines the hard disk parameters **automatically**.

With this option, the LBA mode is also set for hard disks which can work in LBA Mode. For hard disks which cannot work in LBA Mode, the normal mode is set.

Note

If you use a hard disk ≥ 512 Mbytes which was formatted in normal mode, you must switch off the option “Auto Ide Detection” (“Auto” = “[]”) and enter the values as hard disk parameters which were used when the hard disk was formatted.

The CP 581 generally formats the hard disk in normal mode.

If an OSD (onboard silicon disk) is inserted, this can be addressed as drive D: (with a hard disk) or C: (without a hard disk). The OSD is not displayed in the AT Hard Disk page.

The transfer modes “Block Mode” and “32-Bit” are used for faster data exchange. Owing to the ISA interface, however, Block Mode and 32-Bit Mode do not improve performance significantly.

Following a CMOS error the default setting “Auto” (Auto Ide Detection) is switched on.

The options in the parameter group “Slave” are switched off as the default and **must not be changed**.

The option “**Fast PIO**” must not be set.

The value for “Drive Wait Time” is a monitoring time when waiting for a drive to start up. To change the startup time you can set the existing preset of 8 seconds to a new value in the range from 0 to 15 seconds.

- Floppy/Card:

Floppy/Card

Drive A:	Drive B:
<input checked="" type="radio"/> NONE	<input checked="" type="radio"/> NONE
<input type="radio"/> 360 Kb	<input type="radio"/> 360 Kb
<input type="radio"/> 1.2 Mb	<input type="radio"/> 1.2 Mb
<input type="radio"/> 720 Kb	<input type="radio"/> 720 Kb
<input type="radio"/> 1.44 Mb	<input type="radio"/> 1.44 Mb
<input type="radio"/> Memcard	<input type="radio"/> Memcard

OK
CANCEL

Using the “Floppy/Card” page you set the parameters for the drives A: and B:. The parameters marked with “(*)” are selected or preset.

- Boot Features :

Boot Features

OK CANCEL DEFAULT

<p>Boot Sequence</p> <p><input checked="" type="radio"/> A:, C:</p> <p><input type="radio"/> C:, A:</p>	<p>Keyboard State</p> <p><input checked="" type="checkbox"/> Typematic Settings Enabled</p> <p>Typematic Rate: 6 (char/s)</p> <p>Typematic Delay: 250 (msec)</p> <p><input checked="" type="checkbox"/> Num Lock On</p>
<p>Halt On...</p> <p><input checked="" type="checkbox"/> Halt on all Errors Except for...</p> <p><input type="checkbox"/> Keyboard Errors</p> <p><input type="checkbox"/> Disk Errors</p>	<p>System Memory</p> <p><input checked="" type="checkbox"/> Quick Memory Test</p>

Using the “Boot Features” page you set the boot parameters. The parameters marked with “(*)” and “[✓]” are selected or preset.

Parameter group “**Halt On**”:

Independent of the setting the module always runs through this. According to the selections, any errors occurring during booting are displayed.

- S5 Setup

S5Setup

<p>CONTROL</p> <p><input type="checkbox"/> S5 interface</p> <p><input checked="" type="checkbox"/> Page frames</p> <p><input type="checkbox"/> I/OPointer</p> <p><input type="checkbox"/> Pointerread</p>	<p>COMMUNICATIONFLAGS</p> <p><input type="checkbox"/> 0- 31 (F200H-F21FH)</p> <p><input type="checkbox"/> 32- 63 (F220H-F23FH)</p> <p><input type="checkbox"/> 64- 95 (F240H-F25FH)</p> <p><input type="checkbox"/> 96- 127 (F260H-F27FH)</p> <p><input type="checkbox"/> 128- 159 (F280H-F29FH)</p> <p><input type="checkbox"/> 160- 191 (F2A0H-F2BFH)</p> <p><input type="checkbox"/> 192- 223 (F2C0H-F2DFH)</p> <p><input type="checkbox"/> 224- 255 (F2E0H-F2FFH)</p>
<p>PAGEFRAMES</p> <p>FRAMES: 4</p> <p>Mode: Paged</p> <p>PageNumber: 0</p> <p>Base: F400H</p>	
<p>I/O POINTER</p> <p>Range: P periphery</p> <p>Base: FX00H</p>	<input type="button" value="OK"/> <input type="button" value="CANCEL"/>

Table 3-38 Parameters and Default Settings with BIOS S5 Page: CONTROL

Control

Input field	Possible inputs	Description
S5 interface	[.]	Access via S5 bus completely disabled.
	[v]	Access via S5 bus completely enabled.
Page frames	[.]	Access of page frames via S5 bus disabled.
	[v]	Access of page frames via S5 bus enabled.
I/O Pointer	[.]	I/O-mapped access operations disabled.
	[v]	I/O-mapped access operations enabled.
Pointer read	[.]	Reading of I/O address pointers disabled.
	[v]	Reading of I/O address pointers enabled.

Table 3-39 Parameters and Default Settings for BIOS S5 PAGE:Page Frame

Page Frame

Input field	Possible inputs	Description
Frames	1, 2, 4*, 8	Number of pages ¹⁾
Mode	Paged *	Access operations via the S5 bus takes place in page mode
	Linear	Access operations via the S5 bus take place with linear addressing
Page Number	0 - 255 (modulo FRAMES)	Base interface number (default setting 0)
Base	0000 - FC00H (modulo 400H)	Page base address (default setting F400H)

¹⁾ If you use the system software of the CP 581, you may only enter the value "4" here.

Table 3-40 Parameters and Default Settings with BIOS S5 Page: I/O Pointer

I/O Pointer

Input field	Possible inputs	Description
Range	P area *	Initial address of I/O-mapped periphery points to the P area (F000H - F0FFH)
	Q area	Initial address of I/O-mapped periphery points to the Q area (F100H - F1FFH)
Base	0 - 255 (modulo 8)	Base address for I/O-mapped access (default setting 0)

Table 3-41 Parameters and Default Settings with BIOS S5 Page: Communication Flags

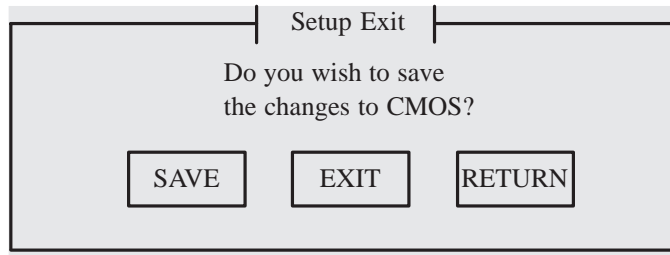
Communication Flags

Input field	Flag	Possible inputs
F200-F21F	0 - 31	<input type="checkbox"/>
		<input type="checkbox"/>
F220-F23F	32 - 63	<input type="checkbox"/>
		<input type="checkbox"/>
F240-F25F	64 - 95	<input type="checkbox"/>
		<input type="checkbox"/>
F260-F27F	96 - 127	<input type="checkbox"/>
		<input type="checkbox"/>
F280-F29F	128 - 159	<input type="checkbox"/>
		<input type="checkbox"/>
F2A0-F2BF	160 - 191	<input type="checkbox"/>
		<input type="checkbox"/>
F2C0-F2DF	192 - 223	<input type="checkbox"/>
		<input type="checkbox"/>
F2E0-F2FF	224 - 255	<input type="checkbox"/>
		<input type="checkbox"/>

Saving the Changed Setup Parameters

When you have set all the setup parameters according to your hardware configuration and system requirements, you must then display the selection menu page again. To do this, exit the last setup page you edited using ESC.

Now select the option EXIT in the selection menu and confirm your selection with RETURN. The “Setup Exit” page is then displayed (see following figure).



You can now display the following options:

- With SAVE you save all the changes you made in the setup and reboot the system.
- Use EXIT (or with ESC) to boot the system without saving the setup changes (apart from date and time - these are entered immediately after a value is entered and the tab key is pressed).
- With RETURN you return to the selection menu.

3.8 BIOS Setup for CP 581 Pentium

Switching on the PLC / Cold Restart

When you switch on the PLC or cold boot the CP 581, the BIOS starts with the following messages.

```
PhoenixBios x.x Release x.x
Copyright 1985 - 1998 Phoenix Technologies Ltd.
All Rights Reserved
->
Build Time mm/dd/yy
Siemens CP 581 Pentium xxxVx.xx

Press  to enter setup
```

After these messages, the driver for the silicon disk is loaded. The following messages are displayed during this:

```
Silicon Disc Driver
*** A&D AS 31x *** (pp)
Port I/O
Copyright (c) Siemens AG 1992 - 96
Silicon Drive B: D: installed on CP 581
```

Changing the Set BIOS Parameters

If you want to change the set BIOS parameters, you have press the function key immediately after the first message appears.

The BIOS setup selection page is then displayed (see Figure 3-14).

Use the following keys to modify the BIOS parameters in this and subsequent screens:

Use the cursor keys (“cursor left” and “cursor right”) to select the “Main”, “S5-Interface” and “Exit” pages.

Use the following keys to select and change individual setup parameters:

- ↑ and ↓ (“cursor up” or “cursor down”)

Use this key to move the cursor to the next option (= highlighted field) or to the next parameter of option.

- Cursor keys + and - :

Use these keys to change the settings of a parameter: either the numerical value (for example, the hours for the time of the day) or another alternative for a parameter (for example, the parameter “Mode” in the option PAGE FRAMES in the S5 Interface).

- RETURN key :
Use this key to call up a list of possible values for the selected parameter or another setup page.
- ESC key:
Use this key to return to the prior setup selection page. The ESC key returns you to the “Exit” page from within the “Main” and “S5 Interface” pages.

Setup Pages

This section describes the individual setup pages.

Note

In the BIOS of the CP 581 Pentium, password protection for the BIOS setup and system boot is not possible. Nor is it possible to write protect floppy disks and hard drives using the BIOS setup.

Main

You can set the following parameters on this page:

Main	S5 - Interface	Exit	Item Specific Help
System Time:		[10:01:20]	<hr/> <Tab>, <Shift-Tab>, or <Enter> selects field.
System Date:		[12/02/98]	
Legacy Diskette A:		[1.44/1.25 MB 3"]	
Legacy Diskette B:		[Memory Card]	
▶ Primary Master		[1445 MB]	
▶ Primary Slave		[None]	
▶ Keyboard Features			
Boot sequence		[A: then C:]	
System Memory		640 KB	
Extended Memory		15360 MB	
F1 Help	Select Item	-/+ Change Values	F9 Setup Defaults
Esc Exit	Select Menu	Enter Select > Sub-Menu	F10 Save and Exit

Figure 3-14 “Main” Selection Page

Parameter	Explanation
System Time	Sets the current time
System Date	Sets the current date
Legacy Diskette A:	Options for drive A <ul style="list-style-type: none">• Disabled• 1.44/1.25 MB 3.5"• Memory card
Legacy Diskette B:	Options for drive B <ul style="list-style-type: none">• Disabled• Memory card
Primary Master	Selects the hard drive parameters, calls a further setup page
Primary Slave	Irrelevant for the CP 581 Pentium
Keyboard Features	Selects the keyboard parameters, calls a further setup page
Boot sequence	Selects the boot sequence <ul style="list-style-type: none">• A: then C:• C: then A:• C: only
System Memory	Displays the amount of system memory
Extended Memory	Displays the amount of extended memory

Main → Using the “Primary Master” you can set the hard disk parameters for
Primary Master drive C:.

Main					
Primary Master [1445MB]			Item Specific Help		
Type:	[Auto]	User = you enter parameters of hard disk drive installed at this connection. Auto = autotypes hard disk drive installed here. 1-39 = you select pre-determined type of hard disk drive installed here. CD-ROM=a CD-ROM drive is installed here. ATAPI Removable=removable disk drive is installed here.			
Cylinders:	[2000]				
Heads:	[16]				
Sectors:	[63]				
Maximum Capacity:	1445MB				
LBA Mode Control:	[Enabled]				
F1 Help	Select Item	-/+	Change Values	F9	Setup Defaults
Esc Exit	Select Menu	Enter	Select > Sub-Menu	F10	Save and Exit

Figure 3-15 “Primary Master” Selection Page

Parameter	Explanation
Type	<ul style="list-style-type: none"> • Auto: Hard drive parameters are detected automatically • User: User-defined hard drive parameters • None: No hard drive present

By default, the option “Auto Ide Detection” is enabled (“Type” = “Auto”).

With this option, the LBA mode is also set for hard disks which can work in LBA Mode. For hard disks which cannot work in LBA Mode, the normal mode is set.

If an OSD (onboard silicon disk) is inserted, this can be addressed as drive D: (with a hard disk) or C: (without a hard disk). The OSD is not displayed in the “Primary Master” page.

Following a CMOS error the default setting “Auto” (Auto Ide Detection) is switched on.

Main → The options here are switched off by default and **must not be changed**.
Primary Slave

Main →
Keyboard Features

Main	
Keyboard Features	Item Specific Help
Numlock: [Auto] Keyboard auto-repeat rate: [30 / sec] Keyboard auto-repeat delay: [1/2 sec]	Selects Power-on state for Numlock
F1 Help Select Item -/+ Change Values F9 Setup Defaults Esc Exit Select Menu Enter Select > Sub-Menu F10 Save and Exit	

Figure 3-16 “Keyboard Features” Selection Page

Parameter	Explanation
Numlock	Selects the status of the Numlock key after reset <ul style="list-style-type: none"> • Auto: Automatic setting • On: Numlock on • Off: Numlock off
Keyboard auto-repeat rate	Selects the keyboard repeat rate Allowed values are from 2/s to 30/s
Keyboard auto-repeat delay	Selects the keyboard delay Allowed values are from 1/4 s to 1 s

S5 Interface

Main	S5 - Interface	Exit	Item Specific Help
S5 Interface: [Disabled] Page-Frames: [Enabled] Frames: [4] Mode: [PAGED] Page-Number: [0] Base: [??00] [244]			Enable or disable access to S5 bus
I/O-Pointer [Disabled] Range: [P-PERIPHERIE] Base: [FX..] [0]			
Pointer Read [Disabled]			
▶ S5 Communication Flags			
F1 Help	Select Item	-/+ Change Values	F9 Setup Defaults
Esc Exit	Select Menu	Enter Select > Sub-Menu	F10 Save and Exit

Figure 3-17 “S5 Interface” Selection Page

The parameters and default settings for the “S5 Interface” setup page are listed in the following.

Parameter	Explanation
S5 Interface	Disable or enable access via S5 bus <ul style="list-style-type: none"> • Disabled (access blocked, default) • Enabled (access allowed)
Page frames	Disable or enable page access via S5 bus <ul style="list-style-type: none"> • Disabled (access blocked) • Enabled (access allowed, default)
I/O Pointer	Disable or enable I/O-mapped peripheral access <ul style="list-style-type: none"> • Disabled (access blocked, default) • Enabled (access allowed)
Pointer read	Disable or enable reading of I/O address pointers <ul style="list-style-type: none"> • Disabled (access blocked, default) • Enabled (access allowed)

Page Frame

Parameter	Explanation
Frames	Number of pages ¹⁾ <ul style="list-style-type: none"> • 1 • 2 • 4 (default) • 8
Mode	Type of access via S5 bus <ul style="list-style-type: none"> • Paged (default) • Linear
Page Number	Base interface number <ul style="list-style-type: none"> • 0 - 255 (modulo frames, default 0)
Base	Page base address <ul style="list-style-type: none"> • 0000 - FC00H (default F400H)

¹⁾ If you use the system software of the CP 581, you may only enter the value "4" here.

I/O Pointer

Parameter	Explanation
Range	Initial address of I/O-mapped periphery points to the P area (F000H - F0FFH) or to the Q area (F100H - F1FFH) <ul style="list-style-type: none"> • P periphery (default) • Q area
Base	Base address for I/O-mapped access <ul style="list-style-type: none"> • 0 - 255 (modulo 8, default 0)8

**S5 Interface →
S5 Communication
Flags**

Main		S5 - Interface		EXIT		
S5 Communication Flags				Item Specific Help		
0 -	31	(F200h - F21Fh)	[Disabled]	Communication Flags 0–31		
0 -	63	(F220h - F23Fh)	[Disabled]			
0 -	95	(F240h - F25Fh)	[Disabled]			
0 -	127	(F260h - F27Fh)	[Disabled]			
0 -	159	(F280h - F29Fh)	[Disabled]			
0 -	191	(F2A0h - F2BFh)	[Disabled]			
0 -	223	(F2C0h - F2DFh)	[Disabled]			
0 -	255	(F2E0h - F2FFh)	[Disabled]			
F1	Help	Select Item	-/+	Change Values	F9	Setup Defaults
Esc	Exit	Select Menu	Enter	Select > Sub-Menu	F10	Save and Exit

Figure 3-18 “S5-Communication Flags” Selection Page

The parameters and default settings for the “S5 Communication Flags” setup page are listed in the following.

Areas		Selection
0 - 31	(F200h - F21Fh)	<ul style="list-style-type: none"> • Enabled • Disabled (default)
0 - 63	(F220h - F23Fh)	
0 - 95	(F240h - F25Fh)	
0 - 127	(F260h - F27Fh)	
0 - 159	(F280h - F29Fh)	
0 - 191	(F2A0h - F2BFh)	
0 - 223	(F2C0h - F2DFh)	
0 - 255	(F2E0h - F2FFh)	

Exit

Main		S5 - Interface		Exit		
Exit Saving Changes Exit Discarding Changes Load Setup Defaults Discard Changes Save Changes				Item Specific Help		
				Exit System Setup and save your changes to CMOS.		
F1	Help	Select Item	-/+	Change Values	F9	Setup Defaults
Esc	Exit	Select Menu	Enter	Select > Sub-Menu	F10	Save and Exit

Figure 3-19 “Exit” Selection Page

Selection	Explanation
Exit Saving Changes	Save changes and exit
Exit Discarding Changes	Exit without saving changes
Load Setup Defaults	Load system default values
Discard Changes	Discard your changes, returning all parameters to their original values
Save Changes	Save your changes without exiting

3.9 Conversion and Repairs

Note

Do not carry out any conversions apart from those described in Chapter 2!
Repairs may only be carried out by Siemens or by companies authorized by Siemens.

3.10 Information for Developers of Own System Software

This section contains information for users who only use the CP 581 hardware and wish to develop their own system software and/or drivers.

3.10.1 S5 interface

Table 3-42 Memory Assignment of S5 Interface

Hardware Structure

CC00:4000	
CC00:3FE0	Control register
CC00:2300	Not used
CC00:2200	Communication flag
CC00:2100	Not used
CC00:2000	I/Os
CC00:1C00	Page 7
CC00:1800	Page 6
CC00:1400	Page 5
CC00:1000	Page 4
CC00:0C00	Page 3
CC00:0800	Page 2
CC00:0400	Page 1
CC00:0000	Page 0

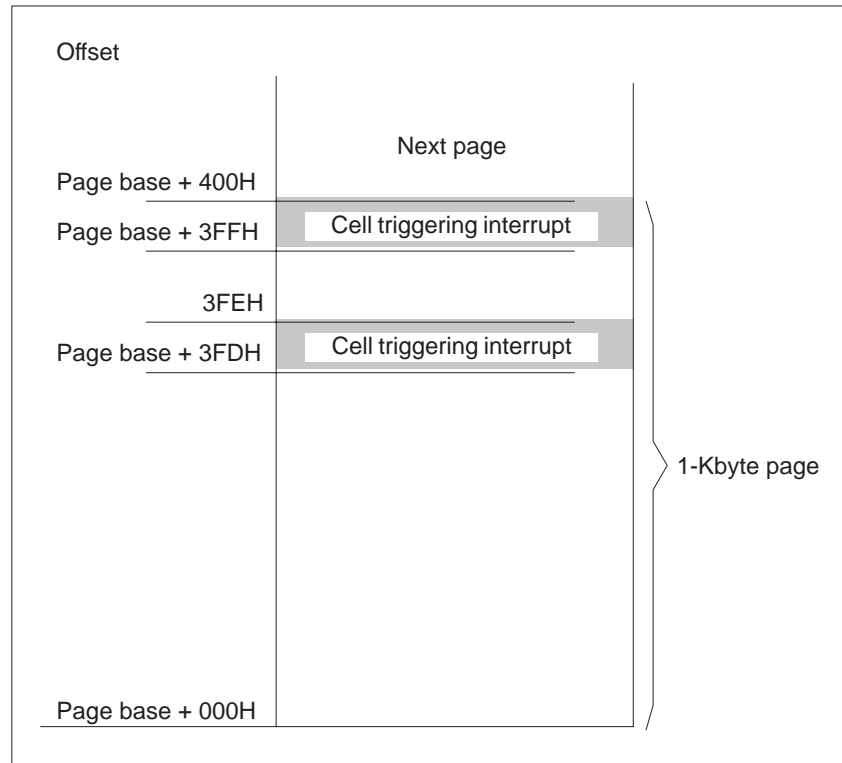


Figure 3-20 Interrupt Triggering with S5 Interface

Each page has two interrupt-triggering cells. A status bit in the interrupt **status** register is assigned to each of these cells (see Figure 3-21). The associated status bit is set when writing the S5 CPU into one of these cells.

An enable bit is assigned to each page in the interrupt **enable** register. Only if the enable bit of a page is set can its interrupts reach the IRQ 15 line and thus trigger an interrupt.

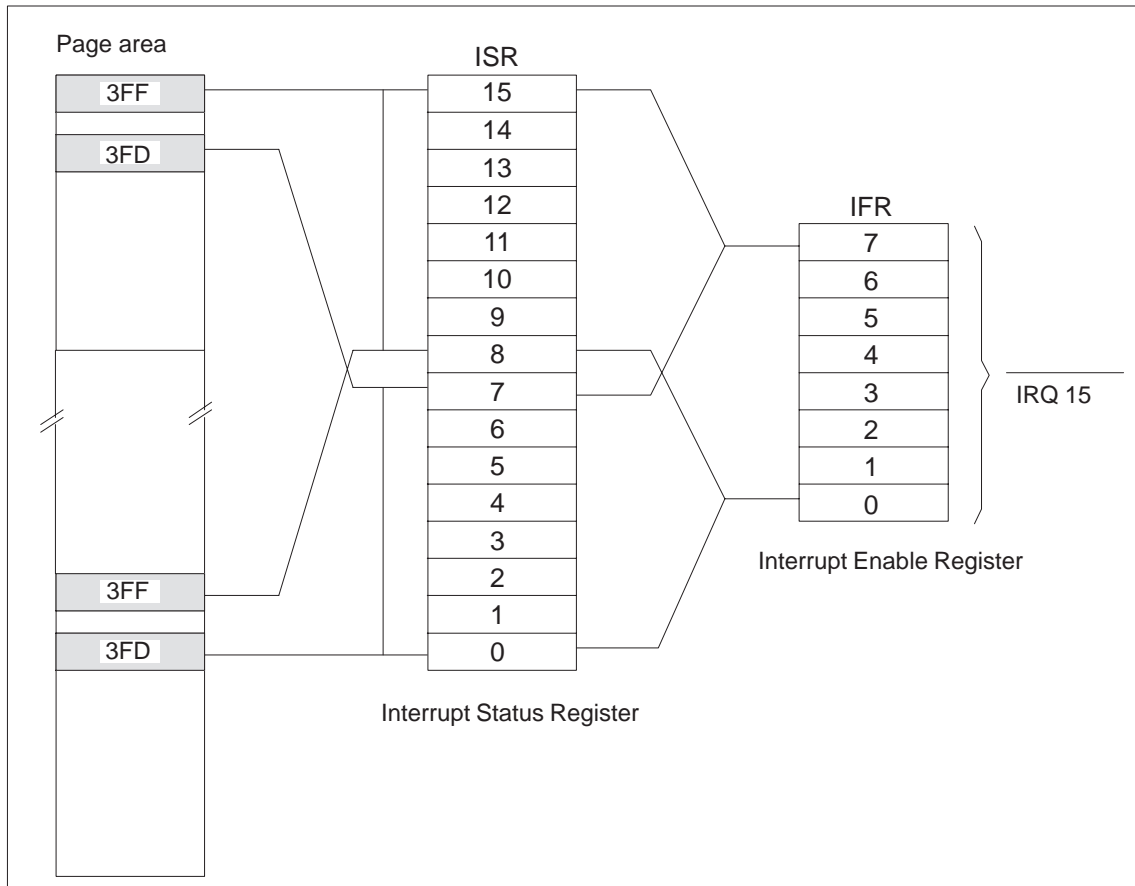


Figure 3-21 Interrupt Register with S5 Interface

Initialization

Global enable for S5 interface

Write port 5007H

Bit No.								Meaning
7	6	5	4	3	2	1	0	
No significance							1	Switch on S5 interface
No significance							0	Switch off S5 interface

**S5 Bus Parameters
(S5 Control
Register (CTLS5))**

Read/write memory CC00:3FE0

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
No significance				SB			S5EN		IOEN	KEN	LIEN	EPB	AZR	KA	

KA Determines the number of pages

KA		Number of pages
0	0	1
0	1	2
1	0	4 (default with reset)
1	1	8

AZR AZR = 0 means that reading of the I/O address pointers (IOAZ 0 - 3) is disabled. Reading is enabled with AZR = 1. (AZR = 1 is the default setting with reset.)

EPB Determines the initial address for the I/O mapped periphery.

EPB	Initial address
0	P area (F000H - F0FFH)
1	Q area (F100H - F1FFH) (default with reset)

LIEN LIEN = 0 means that access is via the S5 bus in page mode. LIEN = 1 means that access is with linear addressing. (LIEN = 0 is the default setting with reset.)

KEN KEN = 0 means that page access via the S5 bus is disabled. This is enabled with KEN = 1. (KEN = 1 is the default setting with reset.)

IOEN IOEN = 0 means that I/O-mapped periphery access is disabled. It is enabled with IOEN = 1. (IOEN = 1 is the default setting with reset.)

S5EN S5EN = 0 means that access via the S5 bus is completely disabled. It is enabled with S5EN = 1. (S5EN = 1 is the default setting with reset.)

SB0 - SB3 SB0 - SB3 has no hardware function. SB0 - SB3 can also be read by the S5 bus by means of a read operation at one of the address pointers IOAZ0 - IOAZ3. (Assumption: status information can be passed on to the S5 in this manner.)

AT Bus Parameters Read/write memory CC00:3FE2
(ISA Control Register (CTLISA))

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
No significance					ZEN	MEN	CS16	WS	TWD	WZEN	WOEN	WD0	L2	L1	L0

L0 L0 = 1 switches the RUN LED on. L0 = 0 switches it off.

L1 L1 = 1 switches the STOP LED on. L1 = 0 switches it off.

L2 L2 = 1 switches the SD BUSY LED on. L2 = 0 switches it off.

WD0 WD0 is valid if WOEN = 1 is set. In this case WD0 indicates the status of the watchdog. This has expired if WD0 = 1, or not yet expired if WD0 = 0.

WOEN WOEN = 1 means that the watchdog output is enabled. It is disabled if WOEN = 0.

WZEN WZEN = 1 means that the watchdog counter is enabled. It is disabled if WZEN = 0.

TWD TWD = "1" means that the watchdog counter is loaded with the contents of the watchdog register and restarted.

WS WS = 1 means that an additional wait cycle is inserted when accessing the dual-port RAM. WS can be set to 0.

CS16 CS16 = 1 means that access to the S5 interface is in 16-bit mode. Access is in 8-bit mode with CS16 = 0. CS16 can only be read. 8-bit mode is fixed for the CP 581.

MEN MEN = 0 means that access to the dual-port RAM is disabled at the AT end. It is enabled with MEN = 1.

ZEN ZEN = 1 means that the counter interface is enabled. It is disabled with ZEN = 0.

Page Parameter Register (KAPA) Read/write memory CC00:3FE4

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KABA						No significance			SSNR						

SSNR SSNR defines the base interface number under which the first page is addressed. If further pages are defined, these are located directly after the first page and occupy the following interface numbers according to their quantity.

KABA KABA determines the page base address under which the page is addressed on the S5 bus. Bit 15 represents address A15, bit 14 represents address A14 etc. Example: 111101 results in the S5 bus address F400H (this value is required for the DHB). 111101 is the default setting following a reset.

Communication Flag Register (KOME) Read/write memory CC00:3FE6

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
No significance								KM							

KM KM determines which communication flags are present in the dual-port RAM of the CP 581. The following table describes the addresses and flag numbers assigned to the respective bits.

Bit No.*	Address in CP 581 communication flag area	Flag	S5 addresses
0	CC00:2200 - CC00:221F	0 - 31	F200H - F21FH
1	CC00:2220 - CC00:223F	32 - 63	F220H - F23FH
2	CC00:2240 - CC00:225F	64 - 95	F240H - F25FH
3	CC00:2260 - CC00:227F	96 - 127	F260H - F27FH
4	CC00:2280 - CC00:229F	128 - 159	F280H - F29FH
5	CC00:22A0 - CC00:22BF	160 - 191	F2A0H - F2BFH
6	CC00:22C0 - CC00:22DF	192 - 223	F2C0H - F2DFH
7	CC00:22E0 - CC00:22FF	224 - 255	F2E0H - F2FFH

* KM 1: enable / KM 0: disable

3.10.2 Interrupt Processing on S5 Interface

Interrupt State Register 1 Read/write memory CC00:3FF2

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KIF								KID							

KIF0 - KIF7 Interrupt from page i triggered at the page offset address 3FFH

KID0 - KID7 Interrupt from page i triggered at the page offset address 3FDH

Note

The interrupts are acknowledged by writing a “1” at the bit position to be acknowledged.

Interrupt State Register 2 Read/write memory CC00:3FF4

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
No significance										S	R	WDI	ZI	SI	RI

RI RI = 1 means that an interrupt has been triggered by the RUN/STOP switch. The switch is changed in the process from STOP to RUN.

SI SI = 1 means that an interrupt has been triggered by the RUN/STOP switch. The switch is changed in the process from RUN to STOP.

ZI ZI = 1 means that an interrupt has been triggered by the counter. A new counted value can be read in the counter register.

WDI WDI = 1 means that an interrupt has been triggered by the expired watchdog.

R R = 1 means that the RUN/STOP switch is in the RUN position.

S S = 1 means that the RUN/STOP switch is in the STOP position.

Note

The interrupts are acknowledged by writing a “1” at the bit position to be acknowledged.

Interrupt Enable Register Read/write memory CC00:3FF6

Bit No.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEG	No significance				IEZ	EWD	ERS	IEK							

IEK0 - IEK7 With IEKi = 1, the interrupts are enabled by page i (offset 3FDH and 3FFH); the interrupts are disabled with iEKi = 0.

IERS IERS = 1 means that RUN/STOP switch interrupts are enabled; they are disabled with IERS = 0.

IEWD IEWD = 1 means that watchdog interrupts are enabled; they are disabled with IEWD = 0.

IEZ IEZ = 1 means that counter interrupts are enabled; they are disabled with IEWD = 0.

IEG IEG = 1 means that the interrupts are all enabled; they are disabled with IEG = 0.

S5 Bus Interrupts

Port	Bit No.								Meaning
	7	6	5	4	3	2	1	0	
Write 5001H	No significance							1	Activate INT A
	No significance							0	Deactivate INT A
Write 5002H	No significance							1	Activate INT B
	No significance							0	Deactivate INT B
Write 5003H	No significance							1	Activate INT C
	No significance							0	Deactivate INT C
Write 5004H	No significance							1	Activate INT D
	No significance							0	Deactivate INT D

3.10.3 Other Status Registers

Battery Failure Indication Read port 5000H

Bit No.								Meaning	
7	6	5	4	3	2	1	0		
No significance								1	Battery failure
No significance								0	No battery failure

Power Supply Failure Indication Read port 5001H

Bit No.								Meaning	
7	6	5	4	3	2	1	0		
No significance								1	No power supply failure
No significance								0	Power supply failure

3.10.4 Peripheral Byte Addressing

The I/O byte access implemented by the CP 581 is an indirect type of addressing for which a memory area of 256 bytes is available. Access is via an address pointer **IOAZ (3:0)** and the associated data register **IODR (3:0)**. Both have a data word width of 8 bits.

A write operation on the address pointer sets the byte offset address provided for the I/O operation. A byte value can be read or written via the data register. Four sets of these registers exist (see Figure 3-22).

An operation is carried out if the S5 address is in the 256-byte I/O area (0xF000 to 0xF0FF or 0xF100 to 0xF1FF) and the S5 addresses **S5A (7:3)** agree with the I/O base address set in the CP 581 in register **IOBA**.

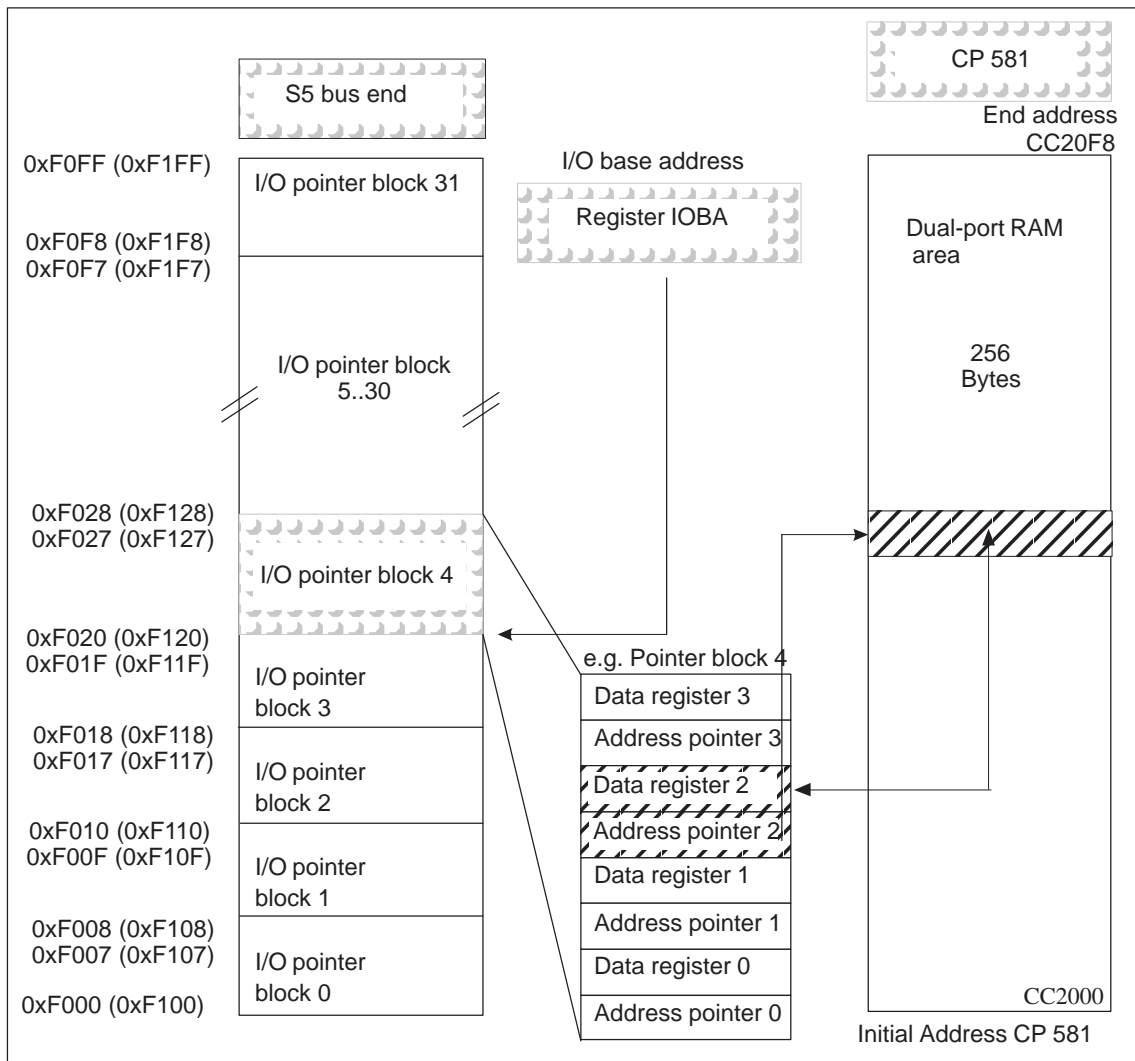


Figure 3-22 I/O Mapped Peripheral Access

Technical Data of the CP 581

Chapter Overview

Section	Description	Page
4.1	Basic Board	4-3
4.2	Mass Storage Module	4-8
4.3	AT Slot Module	4-12
4.4	RTI Module	4-17

Approvals

The following approvals have been granted to the CP581:

UL Recognition Mark

Underwriters Laboratories (UL) in accordance with Standard UL 508 Report E85972

CSA Certification Mark

Canadian Standard Association (CSA) in accordance with Standard C 22.2 No. 142 Report LR 63533

CE Certificate



SIMATIC products meet the requirements of the following EU guidelines.

EMC Guidelines

Our products meet the requirements of the EU Guidelines 89/336/EEG "Electromagnetic Compatibility".

SIMATIC products are designed for industrial application.

Application Fields

For the SIMATIC, the following application field applies in accordance with the CE Certificate:

Application Field	Requirements for	
	Interference outputs	Noise immunity
Industry	EN 50081-2: 1993	EN 50082-2: 1995

Observe the Installation Guidelines

The installation guidelines and safety instructions specified in the System Manual PLC S5-135U/155U must be observed when commissioning and operating the SIMATIC S5. In addition, the following rules must be observed when using certain boards.

Device Installation

Programmable controllers of the SIMATIC S5-135U/155U series must be installed in metal cabinets in accordance with these installation guidelines.

Working on Cabinets

To protect the boards from static electrical discharge, the operator must discharge him/herself before opening cabinets.

Safety Requirements for Assembly

The SIMATIC S5 programmable control system is in accordance with the IEC 61131-2 standard and thus, in accordance with the EU 73/23/EEG low-voltage guideline and in accordance with UL/CSA Certification, is an “open type” system.

To satisfy the requirements for reliable operation in regard to mechanic strength, flame resistance, stability and contact protection, the following installation alternatives are prescribed:

- Installation in a suitable cabinet
- Installation in a suitable housing case
- Installation in a suitably designed, closed operating room.

Statement of Conformity

The availability of the EU Statements of Conformity are maintained for the responsible authorities in accordance with the above-mentioned EU guidelines:

Siemens Aktiengesellschaft
Bereich Automatisierungs- und Antriebstechnik
A&D AS E48
P.O. Box 1963
D-92209 Amberg

4.1 Basic Board

Device-specific Data

	CP 581 80486 DX	CP 581 Pentium
Weight	Approx. 0.5 kg	Approx. 0.5 kg
Module format	Double Eurocard format (160 mm x 233.4 mm)	Double Eurocard format (160 mm x 233.4 mm)
Front panel width	20 mm, 1 1/3 standard slots	20 mm, 1 1/3 standard slots
Backplane connector	ES 902, series 2, 48-pin	ES 902, series 2, 48-pin
Front plugs: Keyboard/COM 2	Female connector, 26-pin (D-subminiature, high-density)	Female connector, 26-pin (D-subminiature, high-density)
COM 1/COM 3	Female connector, 26-pin (D-subminiature, high-density)	Female connector, 26-pin (D-subminiature, high-density)
Video: RGB	3 x coax midi sockets	3 x coax midi sockets (-0EE15)
Video: VGA		15-pin socket (-1EE15)

Power Supply

	CP 581 80486 DX	CP 581 Pentium
Power Supply	+5 V DC, tolerance +/- 5% +24 V DC, tolerance + 25%/-15%	+5 V DC, tolerance +/- 5% +24 V DC, tolerance + 25%/-15%

Current Consumption

	CP 581 80486 DX 5	CP 581 Pentium
Current consumption	at +5 V DC: max. 1.8 A for DX5/133MHz max. 1.6 A for DX5/100MHz at +24V DC: max. 0.1 A	at +5 V DC: max. 3.5 A (133 MHz) max 3.0 A (75 MHz) at +24V DC: max. 0.1 A
Backup current	Max. 0.05 mA	Max. 0.05 mA

Safety

	CP 581 80486 DX/Pentium
Standards	Tested to DIN EN 61131-2 ≅ IEC 61131-2
Degree of protection	IP20 to IEC 60529/DIN 40050

**Electromagnetic
Compatibility
(EMC)**

	CP 581 80486 DX/Pentium
Radio interference suppression Limit class	To EN 55011 A ²⁾
Conducted interference on alternating voltage power supply cables (AC 230V) to EN 61000-4-4/IEC 61000-4-4 (Burst) to IEC 61000-4-5 Cable to cable (µs pulse) Cable to ground (µs pulse)	2 kV 1 kV 2 kV
Direct voltage power supply cables (DC 24 V) to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV
Signal cables to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV ¹⁾
Noise immunity to discharges of static electricity in accordance with EN 61000-4-2 / IEC 61000-4-2 (ESD) ²⁾	Noise immunity of 4 kV contact discharge (8 kV air discharge) is guaranteed if installed correctly. (See System Manual PLC S5-135U/155U, Chapter 3)
Noise immunity to electromagnetic high frequency field ²⁾ pulse-modulated to ENV 50140 / IEC 61000-4-3	80 to 100 MHz 10 V/m 80% AM (1 kHz)
Noise immunity to electromagnetic high frequency field ²⁾ amplitude-modulated to ENV 50204	900 MHz 10 V/m 50% ED
Noise immunity to high frequency, sinusoidal to ENV 50141	0.15 to 80 MHz 10 V 80% AM

- 1) Signal cables which do not serve the process control, for example, connections to external peripherals, etc.: 1 kV
2) With cabinet door closed

**Climatic
Conditions**

	CP 581 80486 DX/Pentium
Climatic and environmental conditions (Tested to DIN EN 60068-2- 1/2/3)	
Ambient temperature for operation (Air supply measured at lower air inlet on device)	+ 5 to 55 °C
Transport and storage temperature	- 10 to + 60 °C
Temperature gradient Operation Transport/storage (If delivered below 0°C, allow at least 3h for temperature adjustment because of possible condensation)	Max. 10 K/h Max. 20 K/h
Relative humidity: Operation, transport and storage	Max. 95% at 25 °C, no condensation
Altitude: Operation Transport and storage	- 1000 m to + 1500 m above sea level (1080 hPa to 860 hPa) - 1000 m to + 3500 m above sea level (1080 hPa to 660 hPa)

**Mechanical
Ambient
Conditions**

CP 581 80486 DX/Pentium	
Mechanical requirements	Suitable for installation in stationary non-vibration-proof appliances; also in ships and vehicles if appropriate specifications are observed (must not be connected to the engine).
Oscillations:	Tested to DIN EN 60068-2-6
Operation	10-58 Hz: amplitude 0.075 mm 58-150 Hz: acceleration 9.81 m/s ² (1 g)
Shock:	Tested to DIN EN 60068-2-27
Operation	Half-sine: 150 m/s ² , 11 ms

**Logic
Parameters**

Processor	6ES5581-0ED14	6ES5581-0ED13	6ES5581-xEE15
	80486DX5 - 133	80486DX4 - 100	Pentium
Coprocessor	Integrated		
Chipset	SEQUOIA/Cirrus Logic		INTEL HX Chipset
L2-Cache	-		512 Kbytes
Main memory	DRAM (see Chapter 5, Ordering Information)		
Keyboard controller	82C42PC		82C42PE
COM 1/COM 2	82C552 (16C552)		
COM 3	16C550		
VGA controller	VGA modes(Cirrus Logic BIOS V1.xx with CLGD 5429)		Chips&Technologies CT 69000
Video RAM	1 Mbyte		2 Mbyte
S5 Interface	SIEMENS-ASIC max. 8 Kbyte dual-port RAM 256 interprocessor communication flags I/O area watchdog		

Table 4-1 VGA Modes (Cirrus Logic BIOS V ?? with CLGD 5429 ¹⁾) CP 581 80486 DX

Mode	Text/ graphics	Color/BW	Resolution columns/ lines	Number of colors	Horizontal frequency/ kHz	Vertical Frequency/Hz
0, 1	Text	Color	360 x 400	16	31.5	70
2, 3	Text	Color	720 x 400	16	31.5	70
4, 5	Graphics	Color	320 x 200	4	31.5	70
6	Graphics	Color	640 x 200	2	31.5	70
7	Text	BW	720 x 400	2	31.5	70
D	Graphics	Color	320 x 200	16	31.5	70
E	Text	Color	640 x 200	16	31.5	70
F	Text	BW	640 x 350	2	31.5	70
10	Text	Color	640 x 350	16	31.5	70
11	Text	Color	640 x 480	2	31.5	60
12	Graphics	Color	640 x 480	16	31.5	60
12+	Graphics	Color	640 x 480	16	37.9	72
13	Graphics	Color	320 x 200	256	31.5	70
14	Text	Color	1056 x 400	16	31.5	70
54	Text	Color	1056 x 350	16	31.5	70
55	Text	Color	1056 x 350	16	31.5	70
58, 6A	Graphics	Color	800 x 600	16	35.2	56
58, 6A	Graphics	Color	800 x 600	16	37.8	60
58, 6A	Graphics	Color	800 x 600	16	48.1	72
5C	Graphics	Color	800 x 600	256	35.2	56
5C	Graphics	Color	800 x 600	256	37.9	60
5C	Graphics	Color	800 x 600	256	48.1	72
5D	Graphics	Color	1024 x 768	16	35.5	87
5D	Graphics	Color	1024 x 768	16	48.3	60
5D	Graphics	Color	1024 x 768	16	56	70
5D	Graphics	Color	1024 x 768	16	58	72
5F	Graphics	Color	640 x 480	256	31.5	60
5F	Graphics	Color	640 x 480	256	37.9	72

¹⁾ See also Cirrus Logic BIOS V1.2, Cirrus Logic

Table 4-2 VGA Modes (CT BIOS V ?? with CT 69000¹⁾) CP 581 Pentium

Mode	Text/ graphics	Color/BW	Resolution columns/ lines	Number of colors	Horizontal frequency/ kHz	Vertical Frequency/Hz
0, 1	Text	Color	360 x 400	16	31.5	70
2, 3	Text	Color	720 x 400	16	31.5	70
4, 5	Graphics	Color	320 x 200	4	31.5	70
6	Graphics	Color	640 x 200	2	31.5	70
7	Text	BW	720 x 400	2	31.5	70
D	Graphics	Color	320 x 200	16	31.5	70
E	Text	Color	640 x 200	16	31.5	70
F	Text	BW	640 x 350	2	31.5	70
10	Text	Color	640 x 350	16	31.5	70
11	Text	Color	640 x 480	2	31.5	60
12	Graphics	Color	640 x 480	16	31.5	60
13	Graphics	Color	320 x 200	256	31.5	70
30, 70	Graphics	Color	640 x 480	256	31.5	60
31, 71	Graphics	Color	640 x 400	256	31.5	70
32, 72	Graphics	Color	800 x 600	256	37.9	60
32, 72	Graphics	Color	800 x 600	256	46.9	75
34, 74	Graphics	Color	1024 x 768	256	48.4	60
34, 74	Graphics	Color	1024 x 768	256	60	75
40	Graphics	Color	640 x 400	32k	31.5	60
41	Graphics	Color	640 x 400	64k	31.5	60
42	Graphics	Color	800 x 600	32k	37.9	60
43	Graphics	Color	800 x 600	64k	37.9	60
44	Graphics	Color	1024 x 768	32k	48.4	60
45	Graphics	Color	1024 x 768	64k	48.4	60
50	Graphics	Color	640 x 400	16M	31.5	60
52	Graphics	Color	800 x 600	16M	37.9	60
58, 6A	Graphics	Color	800 x 600	16	35.2	56
58, 6A	Graphics	Color	800 x 600	16	37.8	60
58, 6A	Graphics	Color	800 x 600	16	48.1	72

1) See also CT BIOS V1.0, Chips&Technologies

4.2 Mass Storage Module

Device-specific Data

Weight	Approx. 0.9 kg
Module format	Double Eurocard format (160 mm x 233.4 mm)
Front panel width	20 mm, 1 1/3 standard slots
Backplane connector	ES 902, series 2, 48-pin
Front plugs: LPT	Female connector, 25-pin. (D-subminiature, IBM-compatible)
COM 4	Female connector, 26-pin (D-subminiature, high density)

Power Supply

Supply voltages	+5 V, tolerance +/- 5% +24 V, tolerance + 25%/- 15%
-----------------	--

Current Consumption

Current consumption	+5 V: typically 0.5 A Max. 1 A during restart (approx. 500 ms) +24 V: max. 0.1 A
Backup current	Not required

Safety

Standards	Tested to DIN EN 61131-2 ≅ IEC 61131-2
Degree of protection	IP20 to IEC 60529/DIN 40050

**Electromagnetic
Compatibility
(EMC)**

Radio interference suppression Limit class	To EN 55011 A ²⁾
Conducted interference on alternating voltage power supply cables (AC 230V) to EN 61000-4-4/IEC 61000-4-4 (Burst) to IEC 61000-4-5 Cable to cable (µs pulse) Cable to ground (µs pulse)	2 kV 1 kV 2 kV
Direct voltage power supply cables (DC 24 V) to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV
Signal cables to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV ¹⁾
Noise immunity to discharges of static electricity in accordance with EN 61000-4-2 / IEC 61000-4-2 (ESD) ²⁾	Noise immunity of 4 kV contact discharge (8 kV air discharge) is guaranteed if installed correctly. (See System Manual PLC S5-135U/155U, Chapter 3)
Noise immunity to electromagnetic high frequency field ²⁾ pulse-modulated to ENV 50140 / IEC 61000-4-3	80 to 100 MHz 10 V/m 80% AM (1 kHz)
Noise immunity to electromagnetic high frequency field ²⁾ , amplitude-modulated to ENV 50204	900 MHz 10 V/m 50% ED
Noise immunity to high frequency, sinusoidal to ENV 50141	0.15 to 80 MHz 10 V 80% AM

- 1) Signal cables which do not serve the process control, for example, connections to external peripherals, etc.: 1 kV
2) With cabinet door closed

Climatic Conditions

Climatic and environmental conditions (Tested to DIN EN 60068-2- 1/2/3)	
Ambient temperature: Input air measured at the lower device air intake With inserted diskette Operation with forced ventilation Operation without forced ventilation	+ 5 °C to + 40 °C +5 °C to 55 °C +5 °C to +40 °C
Transport and storage temperature	- 10 to +60 °C
Temperature gradient Operation Transport/storage (If delivered below 0°C, allow at least 3h for temperature adjustment because of possible condensation)	Max. 10 K/h Max. 20 K/h
Relative humidity: Operation, transport and storage	Max. 95% at 25 °C, no condensation
Altitude: Operation Transport and storage	- 1000 m to + 1500 m above sea level (1080 hPa to 860 hPa) - 1000 m to + 3500 m above sea level (1080 hPa to 660 hPa)

Mechanical Environmental Conditions

Mechanical requirements	Suitable for installation in stationary non-vibration-proof appliances; also in ships and vehicles if appropriate specifications are observed (must not be connected to the engine).
Oscillations:	Tested to DIN EN 60068-2-6
Operation	10-58 Hz: amplitude 0.035 mm 58-500 Hz: acceleration 4.9 m/s ² (0.5 g) (measured on drive)
Transport	5 – 9 Hz: amplitude 3.5 mm 9 - 500 Hz: acceleration 9.81m/s ² (1g)
Shock:	Tested to DIN EN 60068-2-27
Operation	Half-sine: 50 m/s ² (5 g), 11 ms (measured on drive)
Transport	Half-sine: 500 m/s ² (50 g), 11 ms

Note

The specified limits are as a result of the hard disk drive. The values must not be exceeded on the drive.

**Logic
Parameters**

Hard disk CP 581 MLFB No.: 6ES5 581-4LA11	IBM 2.5" 1440 MB IDE Interface
MTBF Start/stop cycles	100.000 h > 40.000
Floppy disk drive Type Floppy controller MTBF,	TEAC FD-05-HF (3.5", 1.44 Mbyte) Intel 82077A 30,000 h (typical operation)
Interfaces LPT/COM 4,	16C452

4.3 AT Slot Module

Device-specific Data

Weight	Approx. 0.3 kg
Module format	See also Figure 4-1
Front panel width	20 mm, 1 1/3 standard slots
Backplane connector	None
Front connector	None

Note

A maximum of 2 AT slot modules can be operated simultaneously in **one** CP 581 system.

Power Supply

Supply voltages	+5 V, tolerance +/-5%
-----------------	-----------------------

Current Consumption

Current consumption without AT module	+5 V: typically 0.3 A
Backup current	Not required

Maximum Loading of AT Slot Modules by AT Interfaces

Table 4-3 Voltages and Maximum Loads for One AT Slot Module

Voltage	Tolerance range	Max. current
+ 5 V	4.75 V to 5.25 V	4 A
- 5 V	- 4.4 V to - 5.3 V	70 mA
+ 12 V	11.7 V to 12.3 V	500 mA
- 12 V	- 10.9 V to - 13.5 V	100 mA

Note

All voltages available on the AT bus of the AT slot module are generated by the 5 V power supply. This means that all loads also have an effect on the 5 V supply of the AT slot module.

The 5 V supply for 2 AT slot modules must not be loaded by more than 6 A. It should additionally be noted that 20 W must not be exceeded per AT slot module due to the risk of high dissipation heat. **Without forced ventilation**, this value is reduced to 7.5 W.

Calculation of Current Consumption of AT Modules

Using the following equation you can check whether the power supply will be overloaded. (I' in this equation is the current consumed by a second AT slot module).

$$I_{(12V)*4} + I_{(5V)} + 0.3 \text{ A} + [I'_{(+12V)*4} + I'_{(5V)} + 0.3 \text{ A}] < 6 \text{ A}$$

The loading of the -5 V and -12 V supplies need not be considered. The maximum values listed in Table 4-3 must not be exceeded, however.

Calculation Example for an Assumed Configuration:

Voltage	Current Consumption	Factor	Loading of 5 V supply
1st AT module (in AT Slot B. 1)			
+ 12 V	0.3 A	4	1.2 A
+ 5 V	1.0 A	1	1.0 A
Basic load	0.3 A	1	0.3 A
Total current/heat loss of 1st AT module			2.5 A/ 12.5 W
2nd AT module (in AT Slot B. 2)			
+ 12 V	0.05 A	4	0.2 A
+ 5 V	1.0 A	1	1.0 A
Basic load	0.3 A	1	0.3 A
Total current/heat loss of 2nd AT module			1.5 A/ 7.5 W

In this example, the total current of the 1st AT board is 2.5 A and that of the 2nd AT board is 1.5 A. The load incurred by the 1st AT board requires forced ventilation.

This configuration is therefore only permissible with fans. The 2nd AT module on its own could also be operated without fans.

The total current in this example is 4 A and thus below the maximum value of 6 A. This configuration is therefore permissible.

Safety

Standards	Tested to DIN EN 61131-2 ≅ IEC 61131-2
Degree of protection	IP20 to IEC 60529/DIN 40050

**Electromagnetic
Compatibility
(EMC)**

Radio interference suppression Limit class	To EN 55011 A ²⁾
Conducted interference on alternating voltage power supply cables (AC 230V) to EN 61000-4-4/IEC 61000-4-4 (Burst) to IEC 61000-4-5 Cable to cable (µs pulse) Cable to ground (µs pulse)	2 kV 1 kV 2 kV
Direct voltage power supply cables (DC 24 V) to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV
Signal cables to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV ¹⁾
Noise immunity to discharges of static electricity in accordance with EN 61000-4-2 / IEC 61000-4-2 (ESD) ²⁾	Noise immunity of 4 kV contact discharge (8 kV air discharge) is guaranteed if installed correctly. (See System Manual PLC S5-135U/155U, Chapter 3)
Noise immunity to electromagnetic high frequency field ²⁾ pulse-modulated to ENV 50140 / IEC 61000-4-3	80 to 100 MHz 10 V/m 80% AM (1 kHz)
Noise immunity to electromagnetic high frequency field ²⁾ , amplitude-modulated to ENV 50204	900 MHz 10 V/m 50% ED
Noise immunity to high frequency, sinusoidal to ENV 50141	0.15 to 80 MHz 10 V 80% AM

1) Signal cables which do not serve the process control, for example, connections to external peripherals, etc.: 1 kV

2) With cabinet door closed

Note

The values depend on the AT module used.

**Climatic
Conditions**

Climatic and environmental conditions (Tested to DIN EN 60068-2- 1/2/3)	
Ambient temperature for operation (Air supply measured at lower air inlet on device)	+ 5 to + 55 °C
Transport and storage temperature	- 10 to + 60 °C
Temperature gradient Operation Transport/storage (If delivered below 0°C, allow at least 3h for temperature adjustment because of possible condensation)	Max. 10 K/h Max. 20 K/h
Relative humidity: Operation, transport and storage	Max. 95% at 25 °C, no condensation
Altitude: Operation Transport and storage	- 1000 m to + 1500 m above sea level (1080 hPa to 860 hPa) - 1000 m to + 3500 m above sea level (1080 hPa to 660 hPa)

**Mechanical
Environmental
Conditions**

Mechanical requirements	Suitable for installation in stationary non-vibration-proof appliances; also in ships and vehicles if appropriate specifications are observed (must not be connected to the engine).
Oscillations:	Tested to DIN EN 60068-2-6
Operation	10-58 Hz: amplitude 0.075 mm 58-150 Hz: acceleration 9.81 m/s ² (1 g)
Shock:	Tested to DIN EN 60068-2-27
Operation	Half-sine: 150 m/s ² , 11 ms

**Permissible
Dimensions of AT
Modules**

The following Figure indicates the maximum and minimum dimensions in mm which the AT modules for the CP 581 must observe. Correct installation in the CP 581 system is not possible if the dimensions are exceeded or not reached.

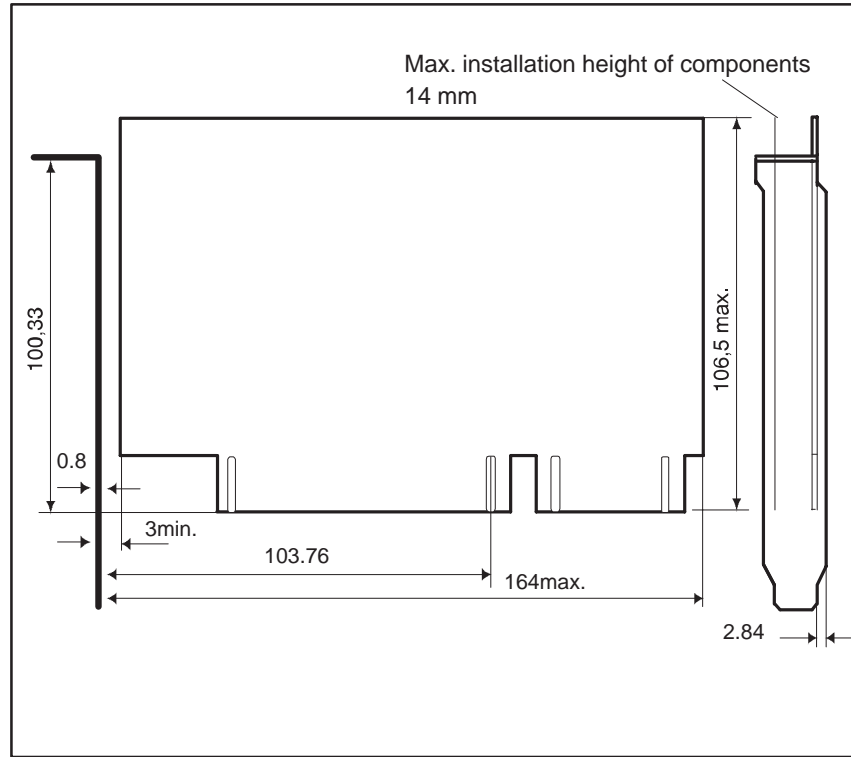


Figure 4-1 Dimensions of AT Modules to be installed in the CP 581 System (in mm)

4.4 RTI Module

Device-specific Data

Weight	Approx. 0.35 kg
Format	See Figure 4.2
Front plugs: Host	Female connector, 25-pin (D-subminiature)
Version -3AA12: Mouse/light pen in connection with CP 581	Female connector, 9-pin (D-subminiature)
Keyboard	Round socket, 7-pin (D-subminiature)

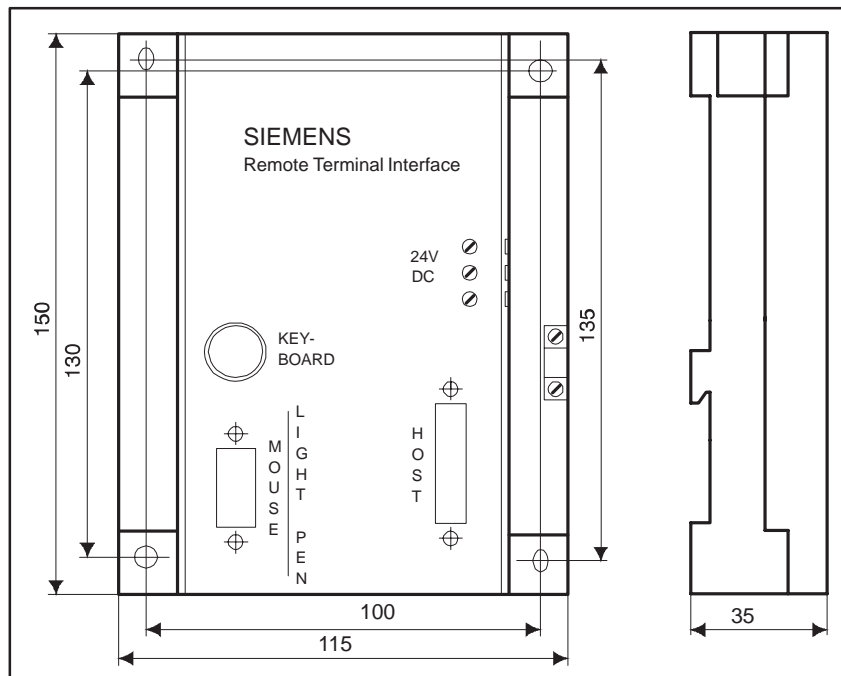


Figure 4-2 Dimensions of the RTI Module (in mm)

Power Supply

Supply voltages	+24 V, tolerance 19 to 32 V, to be supplied acc. to VT 0160.
-----------------	--



Caution

Only an extra-low voltage of ≤ 60 V DC safely isolated from the supply voltage may be used for the 24 V DC power supply. The safe isolation can be applied according to the requirements in, among others, VDE 0100 part 410/HD 384-4-41/IEC 60364-4-41 (as functional extra-low voltage with safe isolation) or in VDE 0805/EN 60950/IEC 60950 (as a safety extra-low voltage, or 'SELV') or VDE 0106 part 101.

Current Consumption

Current consumption	+24 V: max. 0.2 A
---------------------	-------------------

Safety

Standards	Tested to DIN EN 60950 = IEC 60950
Degree of protection	IP00 to DIN 40 050/IEC 60529

Electromagnetic Compatibility (EMC)

Radio interference suppression Limit class	To EN 55011 A ²⁾
Conducted interference on alternating voltage power supply cables (AC 230V) to EN 61000-4-4/IEC 61000-4-4 (Burst) to IEC 61000-4-5 Cable to cable (μ s pulse) Cable to ground (μ s pulse)	2 kV 1 kV 2 kV
Direct voltage power supply cables (DC 24 V) to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV
Signal cables to EN 61000-4-4 / IEC 61000-4-4 (Burst)	2 kV ¹⁾
Noise immunity to discharges of static electricity in accordance with EN 61000-4-2 / IEC 61000-4-2 (ESD) ²⁾	Noise immunity of 4 kV contact discharge (8 kV air discharge) is guaranteed if installed correctly.
Noise immunity to electromagnetic high frequency field ²⁾ pulse-modulated to ENV 50140 / IEC 61000-4-3	80 to 100 MHz 10 V/m 80% AM (1 kHz)
Noise immunity to electromagnetic high frequency field ²⁾ , amplitude-modulated to ENV 50204	900 MHz 10 V/m 50% ED
Noise immunity to high frequency, sinusoidal to ENV 50141	0.15 to 80 MHz 10 V 80% AM

1) Signal cables which do not serve the process control, for example, connections to external peripherals, etc.: 1 kV

2) With cabinet door closed

**Climatic
Conditions**

Climatic and environmental conditions (Tested to DIN EN 60068-2- 1/2/3)	
Ambient temperature for operation (Air supply measured at lower air inlet on device)	+ 5 to + 55 °C
Transport and storage temperature	- 10 to + 60 °C
Temperature gradient Operation Transport/storage (If delivered below 0°C, allow at least 3h for temperature adjustment because of possible condensation)	Max. 10 K/h Max. 20 K/h
Relative humidity: Operation, transport and storage	Max. 95% at 25 °C, no condensation
Altitude: Operation Transport and storage	- 1000 m to + 1500 m above sea level (1080 hPa to 860 hPa) - 1000 m to + 3500 m above sea level (1080 hPa to 660 hPa)

**Mechanical
Environmental
Conditions**

Mechanical requirements	Suitable for installation in stationary non-vibration-proof appliances; also in ships and vehicles if appropriate specifications are observed (must not be connected to the engine).
Oscillations:	Tested to DIN EN 60068-2-6
Operation	10-58 Hz: amplitude 0.075 mm 58-150 Hz: acceleration 9.81 m/s ² (1 g)
Shock:	Tested to DIN EN 60068-2-27
Operation	Half-sine: 150 m/s ² , 11 ms

5

Ordering Information

In this chapter you can find the order numbers of the products mentioned or described in the manual.

CP 581 Ordering Information

Designation	Order No.
CP 581-486DX2	6ES5 581-0ED14
CP 581 Pentium with RGB	6ES5 581-0EE15
CP 581 Pentium with VGA	6ES5 581-1EE15
DRAM module:	
8 Mbytes	6ES7 478-1AN00-0AA0
16 Mbytes	6ES7 478-1AP00-0AA0
32 Mbytes	6ES7 478-1AQ00-0AA0
Onboard silicon disk (4 Mbytes)	6ES7 478-1BM10-0AA0

Remote Terminal Interface (RTI)

Designation	Order No.
RTI module	6ES5 751-3AA11

Connection Cables and Adapters

Designation	Order No.
Connection cable for RTI (-3AA12)	6ES5 714-3xxx1
Y adapter (COM 1/COM 3)	6ES5 714-2AS01
Y adapter (keyboard/COM 2)	6ES5 714-2AT01
RGB/VGA adapter	6ES5 714-2AV01

AT Slot Module

Designation	Order No.
AT Slot Module	6ES5 581-0RA12

Mass Storage Module

Designation	Order No.
Mass storage module with 1.4 GByte hard disk	6ES5 581-aLA11

Memory Cards

Note

The flash EPROMs must be deleted when formatting and overwriting a flash memory card. The number of delete operations is currently limited to approx. 10 000 as a result of the physical characteristics of the flash EPROMs.

Designation	Order No.
Flash EPROM, 1 Mbyte	6ES7 952-1KK00-0AA0
Flash EPROM, 2 Mbytes	6ES7 952-1KL00-0AA0
Flash EPROM, 4 Mbytes	6ES7 952-1KM00-0AA0
Flash EPROM, 8 Mbytes	6ES7 952-1KP00-0AA0
Flash EPROM, 16 Mbytes	6ES7 952-1KS00-0AA0

CP 581 Software

Designation	Order No.
CP 581 system software (international)	6ES5 835-8MD01

SIMATIC S5 STEP 5 Programming Package and PG Connection Cable

Designation	Order No.
STEP 5 Programming Package for PC, 3.5" disks and CD-ROM	5-language (d, e, f, i, s) 6ES5 894-0MA04
programming device connection cable	6ES5 734-2xxx0

Manuals

Designation	Order No.
CP 581 manual, German	6ES5 998-2AT12
CP 581 manual, English	6ES5 998-2AT22

Printer

Designation	Order No.
DR 215-N, 9-needle printer, A4, 8 1/4" x 11 3/4"	6AP1800-0BB00
DR 216-N, 9-needle printer, A3, 16 1/2" x 11 3/4"	6AP1800-0BD00
DR 235-N, 24-needle printer, A4, 8 1/4" x 11 3/4"	6AP1800-0BF00
DR 236-N, 24-needle printer, A3, 16 1/2" x 11 3/4"	6AP1800-0BH00

Interface Modules and Character Set Modules

Designation	Order No. for DR 215/DR 216	Order No. for DR 235/DR236
Centronics with standard character set	6AP1800-0AF00	6AP1800-0AF20
V.24 / TTY Multi-purpose Interface	6AP1800-0AG10	6AP1800-0AG30

Standard Cable Connectors

Designation	Order No.
Centronics, 2 m long	6AP1901-0AL00
V.24 cable, 25-pin plug onto 25-pin plug, 5 m long	6AP1901-0AS00

Keyboard and Mouse

Designation	Order No.
Standard TK200 Keyboard	6GF6710-1AA
Mouse, 2-button	6GF6810-1AA

Monitors

Designation	Order No.
Industrial monitors:	
- PM 36/C2, 14" (see Catalog ST 80) Desktop version Chassis	6AV8011-1FE22-0CA0 6AV8021-1FE22-0CA0
- SCM 2196-I, 21" Desktop version	6GF6100-1BA
Office monitors:	
- SCM 1795, 17" Desktop version	6GF6120-1MC
- SCM 2195, 21" Desktop version	6GF6100-1PB

Monitor Cable

Designation	Order No.
Double-screened coaxial cable (triax cable) Standard length 3.2 m with sockets (Order No. applies to one length of cable)	6ES5 736-2xxx0

Coax Midi Socket

Designation	Order No.
Coax midi socket	W79072-X104

**Length Codes
for Monitor
Cable**

Length of cable connector	Order No. extension for cable connector
1.0 m 1.6 m 2.0 m	6ES5 736-2 xxx0 ↑↑↑ BB0 BB6 BC0
2.5 m 3.0 m 3.2 m	C5 BD0 BD2
5.0 m 8.0 m 10.0 m	BF0 BJ0 CB0
12.0 m 16.0 m 20.0 m	CB2 CB6 CC0
25.0 m 32.0 m 40.0 m	CC5 CD2 CE0
50.0 m	CF0

PLC 115U

Designation	Order No.
Adapter casing for 2 slots	6ES5 491-0LB12
Adapter casing for 4 slots	6ES5 491-0LD11

Note

You can obtain more information from the Catalogs cA01, ST 52.1, ST 52.3, ST 54.1 and ST 80 and in the manuals for the respective programmable controllers.

6

Operation of CP 581 in S5 Programmable Controllers

This chapter tells you which S5 programmable controllers you can use the CP 581 in, and with which applications multi-processor operation is possible (Section 6.1). This section also tells you which resources on the CP 581 and programmable controller CPUs are used for data transfer between the CPU and CP (Section 6.2) as well as how the CPU and CP 581 basically interact (Section 6.3).

The end of this chapter provides you with some information on the simultaneous use of CP 581 applications.

Section	Description	Page
6.1	Programmable Controllers for CP 581	6-2
6.1.1	Single Processor and Multiprocessor Operation	6-2
6.2	Operational Components	6-3
6.2.1	S5 Backplane Bus and Pages	6-3
6.2.2	Data Handling Blocks	6-4
6.2.3	CPDHB Driver	6-4
6.2.4	Principle Interaction Between CPU and CP 581	6-6
6.2.5	Synchronize CP 581 with CPU	6-6
6.2.6	Call CPDHB Driver for Special Application	6-6
6.2.7	Carry Out Data Transfer	6-7
6.3	Simultaneous Operation of CP 581 Application	6-8
6.4	Notes on Use of CP 581 System Software	6-10

6.1 Programmable Controllers for CP 581

You can operate the CP 581 in the S5-115U, S5-135U and S5-155U programmable controllers using all S5 CPUs.

Data can be transferred between the CP 581 and a CPU which is plugged into the same programmable controller as the CP 581 by means of the connection to the S5 backplane bus of the programmable controller. In addition to application-dependent settings on the CP, you must always adapt your STEP 5 programs on the CPU for the planned data transfer. You should have experience in programming S5 programmable controllers before carrying this out.

6.1.1 Single Processor and Multiprocessor Operation

The **CP 581** enables data transfer with **up to four** CPUs on the same programmable controller. Whether several CPUs can be inserted depends on the **type of programmable controller**, however. Whether multi-processor operation is possible with the CP 581 depends on the application.

The various operating modes can be used for the applications as follows:

- Single-processor operation
(Data transfer with **one** CPU: all applications).
- Multi-processor operation
(Data transfer **simultaneously with up to four** CPUs:
mass storage function, command interpreter and free programming).

Note

It may be necessary to take coordination measures on the associated CPUs for multi-processor operation. Please refer to the description of your programmable controller for the means with which you can carry out coordination (for example, using interprocessor communication flags or semaphores).

6.2 Operational Components

Special hardware and software components are available for data transfer between the CP 581 and one or more CPUs:

- S5 backplane bus,
- Pages, dual-port RAM,
- Data handling blocks (DHB),
- CPDHB driver.

6.2.1 S5 Backplane Bus and Pages

Data transfer between a CPU and the CP 581 or also the process peripherals is via the S5 bus. The S5 bus is assigned a specific address area as far as the CPU is concerned (see Figure 6-1).

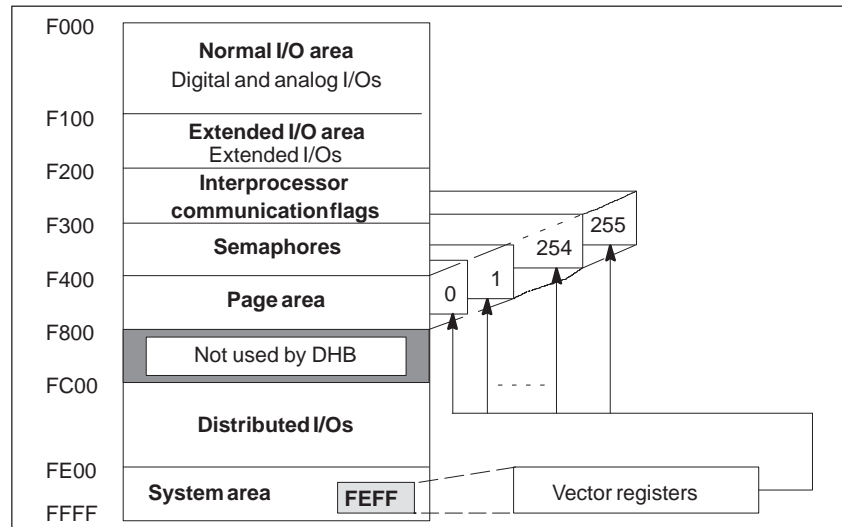


Figure 6-1 Division of Address Area on S5 Bus

Part of this area is reserved for the so-called “pages.” A page has an address area of 1024 bytes. This address area can be addressed several times using the page number (selection via vector registers); there is a total of 256 pages with the numbers 0 to 255.

Four pages with consecutive numbers are defined for the data transfer with the CP 581. This is carried out by setting the base interface number (see Section 6.2.3). These four pages are located physically in succession in the dual-port RAM on the CP 581 module. Each of the four pages is assigned to a CPU. If less than 4 CPUs are inserted into the programmable controller, correspondingly fewer pages are used.

6.2.2 Data Handling Blocks

Data transfer is controlled on the CPUs by the so-called data handling blocks (DHB). These are special function blocks which are called by the STEP 5 program for the desired data transfer and which handle the transfer via one of the pages.

Note

The data handling blocks have different function block numbers (FB numbers) for the various programmable controllers. The numbers of the DHBs required in each case are listed in the application-specific chapters (Chapters 7 to 11). You can find a complete summary of the DHBs in Chapter 13.

6.2.3 CPDHB Driver

The CPDHB driver on the CP 581 handles the data transfer with one or more CPUs. It is loaded into the memory and is called by the CP 581 applications (supplied system programs or user programs) via a special interrupt interface. The driver remains resident in the memory. You can delete it from the memory again using a special command, however.

Options of the CPDHB Driver in the Load Command - Setting the Base Interface Number

On loading the CPDHB driver, you can enter various options fixing the base interface number and changing certain preset characteristics of the driver or of the system software.

In the default setting of the installed AUTOEXEC.BAT file, the CPDHB driver is loaded without options. To enter options you must edit the AUTOEXEC.BAT file and enter the desired options into the load command.

The load command with options has the following syntax:

CPDHB /ssnr<ddd> /noreset /resync /norunstop /int<xx> /f /check

You can also use lower-case letters instead of upper-case letters for the **command**, and also the “-” character instead of the “/” character.

The **options** have to be written lower-case. They can be noted or omitted as desired; their sequence is also optional. Their meanings are explained in the table below. The “<>” characters only identify text parameters and are not used in commands.

Table 6-1 Meaning of Options in the CPDHB Load Command

Option	Meaning																																												
ssnr<ddd>	<p>This option permits the setting of the base interface number (= number of the first page) for data transfer to the CPUs:</p> <p>If no option is specified, the system will use the default SETUP value (see Part 1). To change the base interface number, you must enter the number you require.</p> <p>ddd = base interface number set in steps of 4, i.e. 0, 4, 8 etc. up to 252 max.</p>																																												
noreset	<p>The option prevents a system reset (warm reboot) using the key combination CTRL-ALT-DEL; resetting using the RESET key can still be initiated.</p>																																												
resync	<p>The driver recognizes any synchronization previously initiated by a CPU. A new synchronization of CP 581 by the CPU is then no necessary.</p>																																												
norunstop	<p>This option suppresses the function of the RUN/STOP switch.</p>																																												
int<xx>	<p>Change interrupt for driver call:</p> <p>If the default interrupt INT-66H is already occupied by your software, you can change the interrupt number using this option.</p> <p>xx = number of new interrupt as hexadecimal number.</p>																																												
f	<p>This option activates the processing of the system error display (LED FAULT and Port 80h).</p> <p>Example for the call: cphtb -f</p>																																												
check	<p>This option allows you to read any system errors that have occurred. If a system error has occurred, the LED FAULT will be lit. Only the first error is ever saved.</p> <p>Example for the call: cphtb -check</p> <p>Possible system errors:</p> <table border="0"> <tr><td>0</td><td>No error</td></tr> <tr><td>1</td><td>Unknown task or response</td></tr> <tr><td>2</td><td>Monitoring of buffer size</td></tr> <tr><td>3</td><td>Zero pointer found</td></tr> <tr><td>4</td><td>SEND ALL request acknowledged</td></tr> <tr><td>5</td><td>SEND ALL response acknowledged negative</td></tr> <tr><td>6</td><td>SEND ALL unexpected response</td></tr> <tr><td>7</td><td>SEND DIRECT request acknowledged negative</td></tr> <tr><td>8</td><td>SEND DIRECT response acknowledged negative</td></tr> <tr><td>9</td><td>SEND DIRECT unexpected response</td></tr> <tr><td>10</td><td>RECEIVE ALL request acknowledged negative</td></tr> <tr><td>11</td><td>RECEIVE ALL response acknowledged negative</td></tr> <tr><td>12</td><td>RECEIVE ALL unexpected response</td></tr> <tr><td>13</td><td>RECEIVE DIRECT request acknowledged negative</td></tr> <tr><td>14</td><td>RECEIVE DIRECT response acknowledged negative</td></tr> <tr><td>15</td><td>RECEIVE DIRECT unexpected response</td></tr> <tr><td>16</td><td>FETCH request acknowledged negative</td></tr> <tr><td>17</td><td>FETCH response acknowledged negative</td></tr> <tr><td>18</td><td>Response, although no handshake was started</td></tr> <tr><td>19</td><td>Unknown task number or response</td></tr> <tr><td>20</td><td>Multiple call from IRQ15</td></tr> <tr><td>21</td><td>DEFAULT IRQ7</td></tr> </table>	0	No error	1	Unknown task or response	2	Monitoring of buffer size	3	Zero pointer found	4	SEND ALL request acknowledged	5	SEND ALL response acknowledged negative	6	SEND ALL unexpected response	7	SEND DIRECT request acknowledged negative	8	SEND DIRECT response acknowledged negative	9	SEND DIRECT unexpected response	10	RECEIVE ALL request acknowledged negative	11	RECEIVE ALL response acknowledged negative	12	RECEIVE ALL unexpected response	13	RECEIVE DIRECT request acknowledged negative	14	RECEIVE DIRECT response acknowledged negative	15	RECEIVE DIRECT unexpected response	16	FETCH request acknowledged negative	17	FETCH response acknowledged negative	18	Response, although no handshake was started	19	Unknown task number or response	20	Multiple call from IRQ15	21	DEFAULT IRQ7
0	No error																																												
1	Unknown task or response																																												
2	Monitoring of buffer size																																												
3	Zero pointer found																																												
4	SEND ALL request acknowledged																																												
5	SEND ALL response acknowledged negative																																												
6	SEND ALL unexpected response																																												
7	SEND DIRECT request acknowledged negative																																												
8	SEND DIRECT response acknowledged negative																																												
9	SEND DIRECT unexpected response																																												
10	RECEIVE ALL request acknowledged negative																																												
11	RECEIVE ALL response acknowledged negative																																												
12	RECEIVE ALL unexpected response																																												
13	RECEIVE DIRECT request acknowledged negative																																												
14	RECEIVE DIRECT response acknowledged negative																																												
15	RECEIVE DIRECT unexpected response																																												
16	FETCH request acknowledged negative																																												
17	FETCH response acknowledged negative																																												
18	Response, although no handshake was started																																												
19	Unknown task number or response																																												
20	Multiple call from IRQ15																																												
21	DEFAULT IRQ7																																												

Options on the Installed CPDHB Driver

(The notation is as with the load command)

Uninstall driver:

CPDHB /u

You can carry out a defined abort of the driver using this command and delete it from the memory. If you have loaded further TSR programs, you must initially delete these from the memory in the reverse order to which you installed them.

List driver options on the screen:

CPDHB /h or CPDHB /?

6.2.4 Principle Interaction Between CPU and CP 581

All applications of the CP 581 with respect to data transfer with a CPU take place in the following steps:

1. Start CPDHB driver.
2. Synchronize CP 581 with CPU.
3. Application (for example, CPMASS) calls CPDHB driver.
4. Carry Out Data Transfer

6.2.5 Synchronize CP 581 with CPU

The CP 581 is synchronized with a CPU by the DHB SYNCHRON during the CPU restart. The application-specific chapters (Chapters 7 to 11) show you how to program the DHB call.

Note

The CP 581 and the CPUs inserted into the programmable controller run up simultaneously when the PLC power supply is switched on. The CP 581, however, may require more time than the CPUs. If you call the DHB SYNCHRON in the restart organization blocks of a CPU, you must repeat this in a program loop until the fault LED “Interface not ready” is no longer set by the DHB. Make sure, however, that the **program loop is always terminated** by means of an abort criterion (timer or loop counter).

6.2.6 Call CPDHB Driver for Special Application

The CPDHB driver is called for the respective application by the supplied system programs when you start the corresponding program. If you use free programming, your application program must call the driver via a declared software interrupt (see Chapter 11).

6.2.7 Carry Out Data Transfer

Once the preparatory steps 1 and 2 have been carried out, data transfer between the CPU and CP 581 takes place according to the following principle (see Figure 6-2):

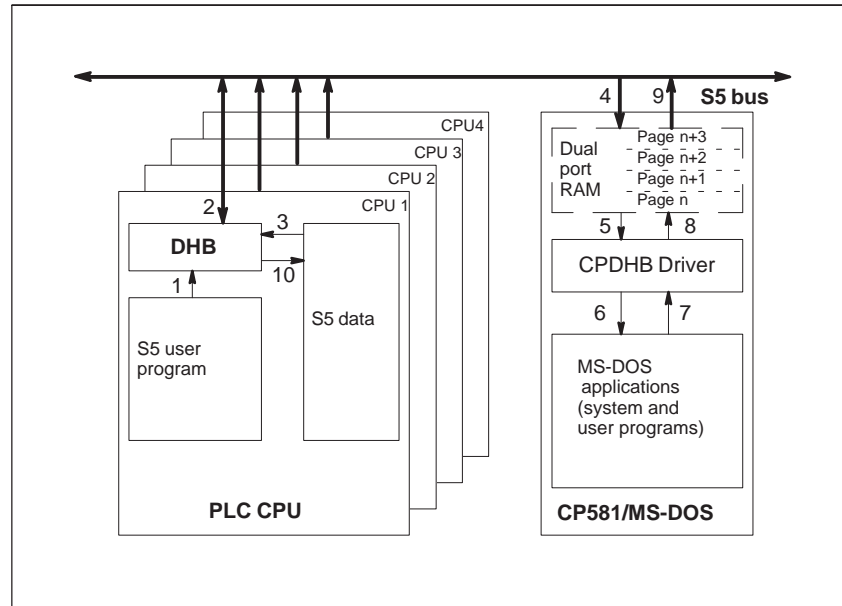


Figure 6-2 Principle Sequence for Data Transfer between CPU and CP 581

- The S5 user program starts the data transfer by calling a DHB and specifies the parameters for the source or destination of the data transfer (1).
- The DHB enters the interface number supplied by the user program as a page No. into the vector register and initially transfers job parameters into the page (2).
- If data are to be transmitted from the CPU to the CP, data are transferred from the S5 area into the page via the S5 bus following a further DHB call (3).
- The CPDHB driver on the CP 581 is informed by an interrupt that data transfer with a CPU is to take place (4).
- A special entry informs the CPDHB driver of the page via which data are to be transferred, i.e. with which CPU (5).
- If data have been transmitted by the CPU, the driver reads them from the page and transfers them to the MS-DOS application (6).
- If data are to be transmitted to the CPU, the driver requests the data from the MS-DOS application and writes them into the page (7). In this case, another DHB call is output by the S5 program on the CPU following initialization of data transfer. The DHB recognizes that data are present in the page (9). It reads these via the S5 bus and transfers them to the S5 area (10).

6.3 Simultaneous Operation of CP 581 Application

The system programs on the CP 581 are designed such that they can execute **all applications simultaneously**. Whether you wish to utilize this facility completely or partially depends on your application problem. You should only make the decision when you are more familiar with the available applications and have read the corresponding sections of the manual. Table 6-2 provides you with a coarse summary of the sequence of applications and the measures which you must make.

When installing the CP 581 system software you can optionally define the entries in the AUTOEXEC.BAT file such that the applications “Mass storage functions” and “Command interpreter” are automatically started following a system restart of the CP 581. This is an important **simultaneous** usage of **both** applications if your CP 581 does not have a keyboard.

Once you have decided which applications are to be used, you can edit the AUTOEXEC.BAT accordingly (start commands) such that these are automatically started each time the CP 581 is restarted.

Table 6-2 Global Sequence of Applications and Required Measures

Application	Global sequence	Required measures
Process Data Acquisition CPRECORD	Read S5 data from a CPU and store in an ASCII file of the CP 581. Data transfer: - From CPU to CP 581 - Initiative: CP 581,	- Edit configuration file on CP 581 - Define conversion of S5 data on CP 581 - Adapt STEP 5 programs for transmission of S5 data
Mass Storage Functions CPMASS	Transfer S5 data from a CPU to the CP 581, and fetch back as required. Data transfer: - From CPU to CP 581 - From CP 581 to CPU - Initiative: CPU	- Adapt STEP 5 programs for the transmission and reception of S5 data
Command Interpreter CPSHELL	Read MS-DOS command from data block of a CPU and execute on CP 581. Data transfer: - From CPU to CP 581 - Initiative: CPU	- Edit S5 data block with command(s) - Adapt STEP 5 programs for transmission of a command to the CP 581

Table 6-2 Global Sequence of Applications and Required Measures

Application	Global sequence	Required measures
Virtual S5 Drive S5Remote	Read and write operations on S5 data areas using MS-DOS commands or file functions of MS-DOS applications with a virtual S5 drive of the CP 581 Data transfer: - From CPU to CP 581 - From CP 581 to CPU - Initiative: CP 581	<ul style="list-style-type: none"> - Adapt STEP 5 programs for read and write operations on S5 data by the virtual S5 drive - Edit format file if required for ASCII conversion of S5 data on CP 581 - Enter name of virtual S5 drive and format file if applicable when activating the application
Free programming of CP 581	Individual user program on CP 581 determines communication Data transfer: - Depending on user program - Initiative: depending on user program	<ul style="list-style-type: none"> - Write special MS-DOS program in assembler- or higher programming language - Adapt STEP 5 programs for communication with special MS-DOS program

6.4 Notes on Use of CP 581 System Software

Keyboard Inputs

Delays may occur in communication between the CP 581 and the CPUs if you make many keyboard entries within a short time. In this case the DHB of an affected CPU signals a configuration error (C1H) to the S5 programs although the transmission is continued without data errors in the next cycle.

This effect may also occur in seldom cases when removing and inserting the keyboard plug. Please therefore observe the following in the case of critical applications:

- Never remove or insert the keyboard plug during operation.
- Do not change the set value of the repetition rate for the keyboard.

Resetting the System (Aborting CPDHB Driver)

Use the following key combination to reset the system:



The driver is then aborted in a defined manner and the system booted (warm restart). The CPUs with which communication is currently being carried out are additionally informed of the abort by the driver using a configuration error status bit (this status bit is also output by the driver when aborting using CPDHB /u).

If you reset the system using the **RESET** key, the driver is **aborted in an undefined manner**. If you then carry out a CPU restart using the driver option RESYNC, there may be an incorrect reaction of a CPU connected to the CP 581.

Using Another Language (Language Switchover)

The **CPDHB**, **CPMASS**, **S5REMOTE/S5REMOTF**, **DVCONVRT** and **INSTALL** programs output messages in the language (English, German or French) which corresponds to the so-called “country code” under MS-DOS.

If you wish to convert the messages to a different language, you must change the instruction in the **CONFIG.SYS** file, for example for US English (country code 001):

```
COUNTRY=001,437,C:\DOS\COUNTRY.SYS
```

The **CPRECORD**, **CPRECCTL** and **CPSHELL** programs output messages in the language which is defined by the following entry in the **AUTOEXEC.BAT** file.

```
SET CP580=C:\CP581\CP580?.MSG
```

where:

- ? = ‘e’ for English,
- ? = ‘d’ for German and
- ? = ‘f’ for French.

You must change this entry accordingly to switch over to another language.

S5 Abbreviations in English

For English speaking countries, certain S5-specific abbreviations are different.

The CP 581 system software applies the English S5 abbreviations for the following **COUNTRY** codes (and corresponding countries):

- COUNTRY-Code 001: USA
- COUNTRY-Code 044: Great Britain
- COUNTRY-Code 061: Australia

The following table shows the English S5 abbreviations and the associated German abbreviations.

Table 6-3 S5 Abbreviations in English and German

S5 abbreviation	German/other language	English
Flag area	MB	FA
Input area	EB	IA
Output area	AB	QA
Peripheral area	PB	PY
Counter cell area	ZB	CA
Timer cell area	TB	TA
Counter constant	KC	KZ
Character constant	KS	KC

Process Data Acquisition CPRECORD

7

This chapter is intended for users who wish to read process data from a CPU and to store and evaluate them on the CP 581. The chapter describes all measures and operations required for process data acquisition on the CP 581, the programming device and on the CPU side:

Only simple operations are required on the CP 581 and programming device for process data acquisition (see Section 7.3.2). You must adapt your STEP 5 programs on the CPUs from which you wish to acquire data to enable the data transfer to take place (see Section 7.3.3). For this you should have experience in programming programmable controllers.

The chapter also provides you with information on everything you must know for evaluation of the acquired data, as well as information on special operations during process data acquisition.

The programming example in Section 7.3.3 only serves to explain the DHB parameters for process data acquisition. Complete programs which can be executed can be found in Chapter 12.

Section	Description	Page
7.1	Application	7-2
7.2	Principal Sequences Between CPU and CP	7-3
7.3	Process Data Acquisition Operations	7-4
7.3.1	Related Procedures	7-4
7.3.2	Measures on the CP	7-5
7.3.3	Programming of CPU for Data Transfer	7-13
7.3.4	Activation, Deactivation and Testing of Process Data Acquisition	7-19
7.4	Evaluation of Acquired Process Data	7-26
7.4.1	Storage of Process Data on the CP 581	7-26
7.4.2	Structure of Process Data in the ASCII Files	7-28
7.4.3	Converting the Individual Data	7-29
7.4.4	Example of "Individual" Conversion	7-31
7.5	Control of Process Data Acquisition	7-32
7.5.1	Control of Acquisition by a CP 581 Command	7-32
7.5.2	Control of Acquisition by a CPU	7-34
7.6	Information on Special Handling	7-37
7.6.1	Information on Parameters and Status	7-37
7.6.2	Reactivating Process Data Acquisition	7-39
7.7	Status Messages	7-41

7.1 Application

You can use the system function "Process data acquisition" to acquire S5 data on the CP 581 from up to 4 CPUs which are inserted in the same programmable controller as the CP 581. The data may be present in up to 8 different S5 data areas such as, for example data blocks, flags etc., even distributed amongst several CPUs.

You can read the data globally or selectively from the S5 data area at specific intervals and store them in one or more files on the CP 581 for later evaluation. The data are converted either individually using format specifications in a file on the CP or universally (for example, in hexadecimal format) into ASCII representation.

The start and end of the recording for an area can be controlled from the CP 581 or from the CPU.

You can record the collected and converted process data on the CP 581 independent of the operations at the CPU side and process them using a suitable MS-DOS program (for example, dBASE).

7.2 Principal Sequences Between CPU and CP

Figure 7-1 shows you a summary of the principal sequences with process data acquisition using the CP 581.

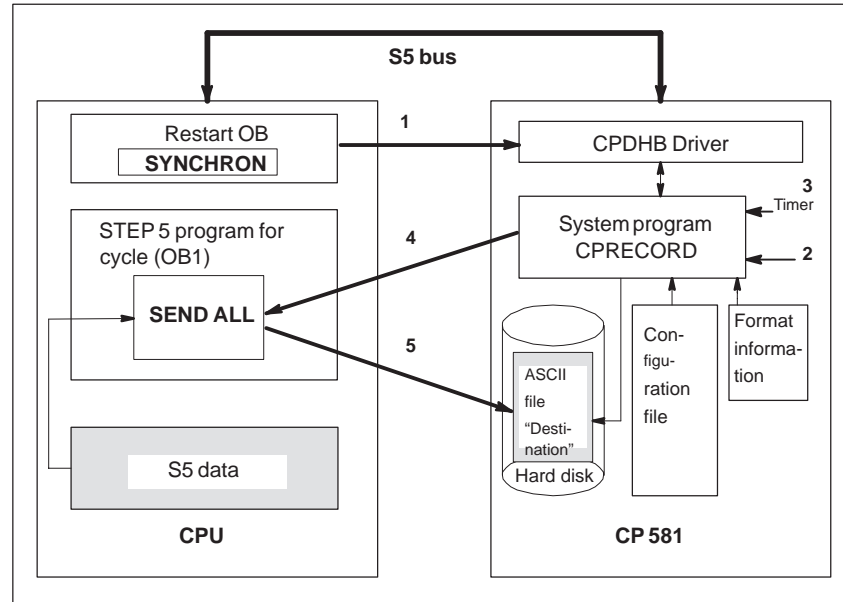


Figure 7-1 Sequences during Process Data Acquisition

- 1 Synchronization of CPU during restart with CP via DHB SYNCHRON. The frame size for data transfer is defined in the process.
- 2 Starting of system program CPRECORD, timer is set by CPRECORD with the cycle time from the configuration file.
- 3 Timer activates CPRECORD cyclically.
- 4 CPRECORD transfers via CPDHB driver trigger to DHB SEND/function SEND ALL with information on the data source and number.
- 5 DHB SEND transmits all required data to the CP following triggering. They are fetched from there by the CPDHB driver and CPRECORD, converted into ASCII format according to the format information and stored in the ASCII file "Destination."

7.3 Process Data Acquisition Operations

7.3.1 Related Procedures

You must carry out the following measures in the suggested sequence in order to apply the process data acquisition function:

Table 7-1 Measures for the Application of Process Data Acquisition

Measure	on	See section
Make sure that the CPU with which you wish to exchange data and the CP 581 are plugged into your programmable controller. The base interface No. for data transfer must be set on the CP 581.	PLC CP 581	7.3.2
Define the S5 data areas (max. 8) on the CPU(s) from which you wish to acquire the data, and define how the acquisition is to be controlled for the individual areas (you can switch the acquisition on and off for every area from the CPU or the CP 581).	--	7.3.2
If the data are to be converted individually edit a file with format information on the CP 581 for each S5 data area or - only with data blocks - load the programming device file ?????ST.S5D with the preheader data of the data block as MS-DOS file on the CP 581.	CP 581 or PG	7.3.2
Initially edit the configuration file CPRECORD.INI supplied on the CP 581 or your own configuration file for one S5 data area.	CP 581	7.3.2
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG CPU	7.3.3
Create or modify the STEP 5 program on the CPU for the cycle such that the DHB SEND is called in each cycle with the function SEND ALL.	PG CPU	7.3.3
Call the CPRECORD program for immediate starting of process data acquisition and test the program sequence on the CPU.	CP 581 CPU PG	7.3.4
Determine by comparing the data (initially transfer statistical test data if necessary) that the functions are executed correctly on the CPU and CP 581.	CP 581 CPU PG	7.4
If you wish to acquire data from more than one S5 area: extend the configuration file by the parameter sets for the other S5 data areas (first uninstall CPRECORD and subsequently call again).	CP 581	7.3.2 and 7.3.3
If you wish to selectively switch the acquisition on and off for the individual data records (CP and/or CPU) or only trigger it from the CPU: please read Section 7.5 and carry out the measures listed there.	CP 581 and/or CPU	7.5

7.3.2 Measures on the CP

Setting the Base Interface Number

So-called pages are used to address the memory areas when transferring data on the S5 bus. The pages have a fixed assignment to the modules involved in data transfer.

The CP 581 can transfer data with up to four CPUs via four successive pages. The number of the first page is the base interface number.

Define this number for the first inserted CPU, and set it on the CP 581 as described in Section 1.2.3. The pages for reading the data from three further CPUs consecutively follow the page with the base interface number.

Defining Parameters for Data Acquisition

First define the following declarations for the acquisition. These are:

- For the complete data acquisition:
 - Assignment of S5 areas to up to max. 8 parameter sets,
 - Time for acquisition cycle in seconds.
- Per S5 data area:
 - CPU number,
 - S5 data area,
 - With data blocks: number of data block on CPU,
 - Initial address of first data unit to be read in the data area,
 - Number of data units to be read,
 - Path data for the ASCII files,
 - ASCII character for the name extension of ASCII files e. g. TXT for the file “name.TXT”),
 - Maximum number of ASCII files for the S5 data area,
 - Maximum number of data records per ASCII file for the S5 data area,
 - Conversion instruction: universal, via format file or via preheader data,
 - Field delimiter (ASCII character) by means of which the individual data units in a data record are to be separated,
 - Acquisition mode (cyclic or until a parameterizable data volume is attained),
 - Setting for output of program messages in a logbook file: output on/off,
 - Path and file names for the logbook file,
 - Timeout.

Definition of Conversion Procedure

Universal conversion:

This is defined using a parameter in the configuration file.

Individual conversion:

You have the opportunity here to use a format file on the CP 581 or - in the case of data blocks - to use the data block file of the programming device with the preheader information.

Conversion Using Format File

Conversion using a format file provides the advantage that you can use it on **all** S5 data areas (not only on data blocks) and that you are independent of your programming device since the format file is created directly on the CP 581.

- **Name** of the format file and **directory**:

The name of the format file can be selected as desired according to the MS-DOS standard and must have the file name extension **FMT**. The directory for the format file is specified in the configuration file.

- **Structure** of the format file:

A format file consists of one or more text lines (ASCII characters). Each line can consist of one or more **format instructions** or a **comment**. It must be terminated by **CR and LF**.

- **Syntax** of a **format instruction**:

A format instruction has the following syntax:

Repetition factor (optional)	Format data	Delimiter
---------------------------------	-------------	-----------

- Repetition factor:

Specifies how many successive items of data are to be converted with the same data format. If it is not specified, the subsequent format instruction is implicitly assigned the repetition factor "1."

Permissible values: 1 to 4091

- Format data:

Permissible values: KC for S5 format KC,
 KF for S5 format KF,
 KG for S5 format KG,
 KH for S5 format KH,
 KM for S5 format KM,
 KS for S5 format KS,
 KT for S5 format KT,
 KY for S5 format KY.

- Delimiter:

Permissible values: (Character/ASCII decimal equivalent):
 Space / 32,
 Comma / 44,
 Horizontal tab/9,
 Semicolon/59,
 CR + LF/13 + 10.

- **Comment line:**

A line is not interpreted if it commences with the character ";". You can then use comments in a format file.

**Example of a
Format File:**

Name: MYFORM.FMT

Contents:

;3 fixed-point numbers:
3KF
;4 floating-point numbers:
4KG
;6 characters:
3KC
;2 Bit patterns:
2KM

**Conversion of
Data of a Data
Block Using
Preheader Data:**

If you wish to acquire process data using a data block DB or DX, you can use the preheader data of this block for the conversion:

The preheader data are generated on the programming device when programming the data block and contain information on the formats of the individual data words in the data block. You must make the preheader data available on the CP 581 for the conversion. Proceed as follows:

- Call the S5 software package on the programming device, define a data block DB or DX and enter the data in the desired format. The data have the function of token characters here. Store the data block on the programming device in a file ??????ST.S5D (?????? = any sequence of letters/numbers, "@" as filler); following storing, this file contains the data of the data block and the preheader information.
- Copy this file in MS-DOS format onto a 3.5" floppy disk. If you do not have a programming device with S5 software with the MS-DOS or FlexOS operating system, you must transfer the S5 file from the PCP/M format onto a floppy disk formatted for MS-DOS using an appropriate program.
- Use the MS-DOS command COPY to load the file from the floppy disk onto your CP 581.

Note

The data will be converted incorrectly if the preheader data do not agree with the data block.

If the data block is **longer than the preheader data**, the data of the data block for which **no** preheader information exists are converted with the format "**KF**."

**Editing the
Configuration File**

The file CPRECORD.INI (configuration file) in the directory CP 581 on the hard disk contains the parameters with which the CPRECORD program carries out process data acquisition.

The configuration file consists of ASCII characters and has the following structure:

“Acquisition cycle” (= 1st parameter line in the file)
Parameter set for 1st S5 data area (must be present)
Parameter set for 2nd S5 data area (optional)
.
.
Parameter set for 7th S5 data area (optional)
Parameter set for 8th S5 data area (optional)

– **Acquisition cycle:**

Time in seconds (max. 11 799 360) for the cycle in which **all** S5 data areas are to be acquired.

You must enter “0” here if you only wish to trigger acquisition from the CPUs (see Section 7.5.2).

– **Parameter set:**

You must store a data record in the configuration file for each S5 data area from which you wish to acquire data. The file may have up to 8 parameter sets. Each parameter set consists of 15 parameter lines (see Table 7-2) which are consecutively numbered from 2 to 16. The **sequence** of parameters is **compulsory!** A line may be up to 80 characters long (without terminators) and must have the following syntax:

Parameter	Semicolon	Comment	CR	LF
	Optional			

A line is not evaluated as a parameter line if its first character is a semicolon. You can use this feature to continue long comments concerning a parameter in a second line. If the semicolon is to be part of a parameter (this is only the case with the parameter “Field delimiter”), you must write two semicolons in succession.

Table 7-2 shows the meanings of the individual parameters and the permissible values. Numbers can be entered in **decimal form** (1234) or **hexadecimal form** (0xabcd), for example, offset for AS.

Table 7-2 Format of a Parameter Set in the Configuration File

Line No.	Parameter	Meaning	Permissible values
2	CPU No.	Number of CPU corresponding to slot sequence	1 to 4
3	S5 area (QTYP)	Specification of data source on CPU: QA for output area AS for absolute address RS for RS word DB for data block DB DX for data block DX IA for input area FA for F flag area PY for I/O area TA for timer cell area CA for counter cell area,	QA, AS, RS, DB, DX, IA, FA, PY, TA, CA
4	Block No.	Number of data block DB or DX on the CPU, if the data source is a data block; this parameter has no significance for the other data (the parameter line must still be present, however!)	Depending on CPU
5	Offset (QANF)	No. of first data unit to be read in the S5 area (word or byte No. - see Table 7-3)	Depending on data type and CPU
6	Number (QLAE)	Number of data units (words or bytes - see Table 7-3), which are to be read from the S5 area starting at "Offset"	Depending on data type and CPU
7	Destination path	Path name for ASCII files in which the data of the S5 area are to be stored ¹⁾	MS-DOS syntax
8	Extension	File name extension ²⁾	ASCII characters
9	Number of files	Maximum number of ASCII files for data acquisition from the S5 area	1 to 10 000 ³⁾
10	Number of data records	Maximum number of data records per ASCII file (for data record, see Section 7.4.2).	1 to $(2^{31} - 1)$ ^{3) 4)}
11	Format	<p>"Individual" conversion: path or file name for format file or (only with DB/DX) for file with preheader data</p> <p>"Universal" conversion: KS for 2-character constant KF for fixed-point number KG for floating-point number KH for hexadecimal number KM for bit pattern KT for timer value KC for counter value KY for 2-byte decimal number KB for 1-byte decimal number (only meaningful for data areas IA, QA, PY and FA)</p>	<p>MS-DOS syntax, file name: "?????ST.S5D" or "*.FMT"; ? = letter/number or @, * = max. 8 letters/numbers</p> <p>KS, KF, KG, KH, KM, KT, KC, KY, KB</p>
12	Field delimiter	Characters by means of which the individual data of a data are to be separated in the ASCII files (preset value: space)	ASCII characters

Table 7-2 Format of a Parameter Set in the Configuration File

Line No.	Parameter	Meaning	Permissible values
13	Acquisition mode	0: recording is terminated once the defined file number has been reached (“non-permanent”) 1: “Permanent” recording; the oldest file is deleted when the defined file number is reached and is then overwritten (similar to cycle mode)	0 and 1
14	Message mode	CPRECORD can store error messages in a logbook file: 0 = store no messages 1 = store messages	0 and 1
15	Logbook file	Path name for logbook file	MS-DOS syntax
16	Timeout	Maximum time for duration of a transmission procedure; the time is specified in seconds ⁵⁾	1 to 3600

- 1) For clarity reasons with several acquired S5 areas you should specify an individual directory for each parameter set.
- 2) Section 7.4.1 shows you how the file name is produced.
- 3) Depending on available memory space. Refer to Sections 7.3.4 and 7.4.1 to find out how the CPRECORD program reacts if the memory space is insufficient.
- 4) Is not checked by CPRECORD for reliability.
- 5) The time between triggering of data transmission and its completion is compared with the defined timeout. If it is exceeded 3 times in succession, the CPRECORD program aborts process data acquisition and terminates itself.

Note

In the case of S5 areas which are **byte-oriented** (see Table 7-3), you should use the universal conversion with the format “KB.”

The accuracy of the acquisition cycle is influenced by programs which you execute parallel to CPRECORD. If the set cycle time is exceeded by **more than 10%**, CPRECORD enters the message “Warning: cycle overflow >>10 %” with the date and time into the logbook file and continues data recording.

Remember when defining the number of files that very many files in a directory (several hundred) result in a great increase in the access times to a file.

If you define the parameters “Offset” (line No. 5) and “Number” (line No. 6) for the S5 area, you can see in the following table whether the S5 data are stored byte-oriented or word-oriented in the CPU memory.

Table 7-3 Organization of S5 Data Areas

Area (QTYP/ZTYP)	Meaning	Max. offset	Max. length	Organization
QA	Output byte	127	128	Byte-oriented
AS	Absolute address ¹⁾	0xFFFFF	4096	Word-oriented
RS	RS word ²⁾	511	512	Word-oriented
DB	Data word	4090	4091	Word-oriented
DX	Data word	4090	4091	Word-oriented
IA	Input byte	127	128	Byte-oriented
FA	Flag byte	255	256	Byte-oriented
PY	Periphery byte	255	256	Byte-oriented
TA	Timer cells ²⁾	255	256	Word-oriented
CA	Counter cells ²⁾	255	256	Word-oriented

1) Note in the data area AS:

The area "32 767 words" specified in the data handling block descriptions /8/ /9/ cannot be used since the CP for this area only makes an 8-Kbyte buffer available.

2) Offset and length depend on CPU used.

Note

When you use the process data acquisition for the first time, you should initially only store **one** parameter set in the configuration file and extend it as necessary following testing.

**Example
Configuration File
Contents:**

Line contents	Ps/z = C =	Parameter set/ line No. comment line
;*** CP-581 :Configuration file for process data acquisition *****	C	
10 ;Acquisition cycle: 10 seconds	-- /1	
;-----1st parameter set:	C	
1 ;CPU No. = 1 (process data from CPU with No. 1)	P1 /2	
DB ;S5 area = DB	P1/3	
5 ;DB No. = 5	P1/4	
0 ;Offset = 0 (read from data word DW 0 onwards)	P1/5	
2000 ;Number = 2000 (2000 data words)	P1/6	
C:\CPU1 ;Path name for ASCII files = "C\CPU1"	P1/7	
TXT ;File name extension: TXT	P1/8	
80 ;Number of files: max. 80 ASCII files	P1/9	
1 ;Number of data records: max. 1 data record per ASCII file	P1/10	
; Individual conversion via	C	
; preheader data with file CP581@ST.S5D:	C	
C:\S5D\CP581@ST.S5D;	P1/11	
* ;Field delimiter = * (separate individual data by '*')	P1/12	
1 ;Acquisition mode = 1 ("permanent" acquisition)	P1/13	
1 ;Signalling mode = 1 (output messages)	P1/14	
; Logbook file: messages in file "C:\CPRECORD.LOG"	C	
C:\CPRECORD.LOG;	P1/15	
50 ;Timeout = 50 seconds	P1/16	
;-----2nd parameter set:	C	
2 ;CPU No. = 2 (process data from CPU with No. 2)	P2/2	
DX ;S5 area = DX	P2/3	
11 ;DX No. = 11	P2/4	
100 ;Offset = 100 (read from data word DW 100 onwards)	P2/5	
300 ;Number = 300 (300 data words)	P2/6	
C:\CPU2 ;Path name for ASCII files = "C\CPU2"	P2/7	
DAT ;File name extension: DAT	P2/8	
40 ;Number of files: max. 40 ASCII files	P2/9	
2 ;Number of data records: max. 2 data records per ASCII file	P2/10	
; Universal conversion:	C	
KH ;Format = KH (S5 format for hexadecimal numbers)	P2/11	
\$;Field delimiter = \$ (separate individual data by '\$')	P2/12	
0 ;Acquisition mode = 0	P2/13	
(end of acquisition when	C	
; max. file number reached)	P2/14	
0 ;Signalling mode = 0 (no output of messages)	C	
; Logbook file: messages in file "C:\CPU2\MESS.LOG"	P2/15	
C:\CPU2\MESS.LOG;	P2/16	
30 ;Timeout = 30 seconds	C	
;-----End of configuration file		

7.3.3 Programming of CPU for Data Transfer

Principle

Programming the CPU for data transfer comprises synchronization of the CP 581 during the restart and the cyclic calling of a special function block for transmitting the CPU data. For both functions you require the so-called “Data handling blocks” (DHB): the DHB SYNCHRON for synchronization and the DHB SEND for data transfer.

Figure 7-2 shows you the positions at which you must call the two data handling blocks in your STEP 5 program.

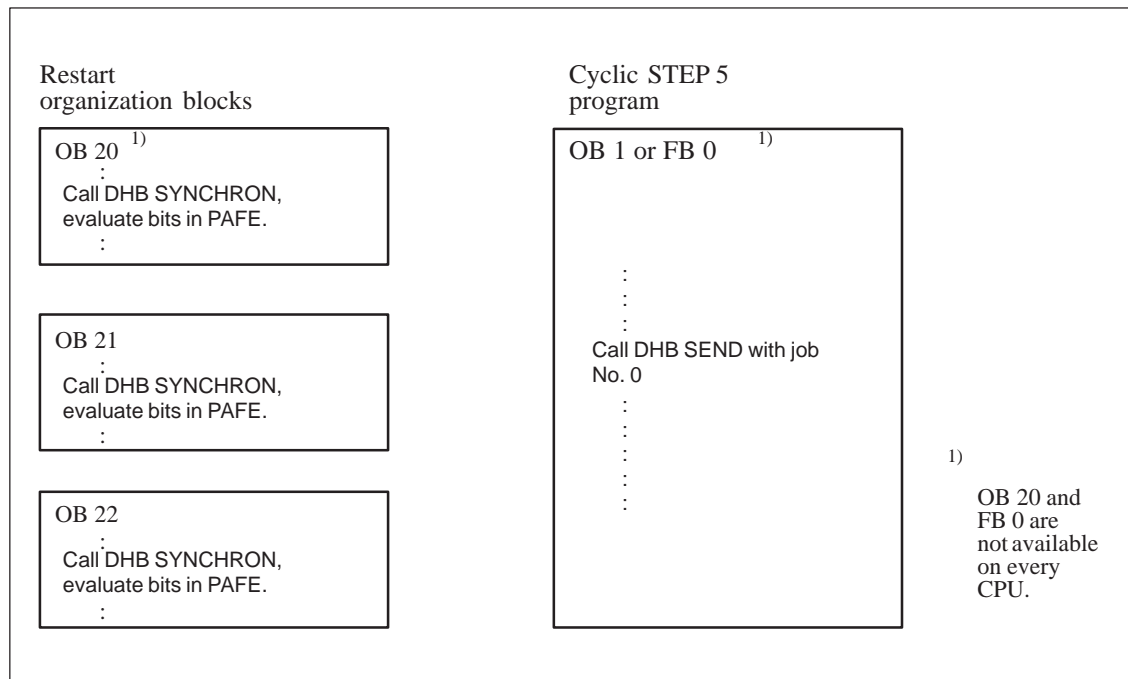


Figure 7-2 Principle of DHB Calls in STEP-5 Program of CPU

Calling and Parameterizing the Data Handling Blocks

The functions of the data handling blocks which you need to transfer CPU data to the CP 581 are explained in this section as far as is necessary for programming.

If you use an S5-135U or S5-155U, you can call the DHB SEND-A which is available there instead of the DHB SEND in your STEP 5 program. You can obtain information on these DHBs from the corresponding descriptions of the DHBs for the S5-135U and S5-155U programmable controllers (see /8/ and /9/ Reference Literature).

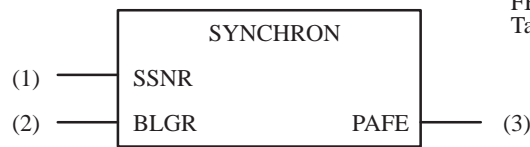
The data handling blocks have different block numbers on the various programmable controllers. The following table lists the numbers of the DHBs SYNCHRON and SEND which you require for the various programmable controllers.

Table 7-4 DHB Numbers on the Various Programmable Controllers

	S5-115U	S5-135U	S5-155U
DHB SYNCHRON	FB 249	FB 125	FB 125
DHB SEND	FB 244	FB 120	FB 120

DHB SYNCHRON:

Block diagram



Use the block No.
FB xxx from
Table 2.4

Table 7-5 Format and Meaning of Parameters for SYNCHRON

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
BLGR	D	KY	Frame size
PAFE	Q	BY	Parameter assignment error bits

Set the following parameters for the DHB SYNCHRON:

– SSNR:

Enter the number of the page by means of which you wish to read the data from the CPU: “(CPU No. - 1) + base interface No.” (see Section 7.3.2).

Permissible values: 0 to 255 ¹⁾

– BLGR:

Use this parameter to define the maximum number of bytes to be transmitted to the CP 581 in a CPU cycle.

Permissible values: 0 = Standard values (see table)

1 = 16 bytes

2 = 32 bytes

3 = 64 bytes

4 = 128 bytes

5 = 256 bytes

6 = 512 bytes

1) The base interface No. must be set in steps of four (0, 4, 8, 12 etc.)!

Standard values for BLGR = 0	
S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

– PAFE:

Byte address for parameter error bits. Bit No. 0 of the status byte is set to “1” in the event of a parameter error (refer to Section 7.3.4 for the meaning of the other status bits).

Note

The CP 581 may require more time to run up than the CPU. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Make sure, however, that the program loop is **always** terminated by an abort criterion (timer or loop counter).

Longer data blocks are transmitted faster if you set the frame size BLGR larger, but the S5 bus is also under a greater work load. On the other hand, the bus load is less with smaller frame sizes, but data transmission to the CP 581 takes longer.

You must decide which frame size is most favorable for your CPU by considering the complete operation on the programmable controller.

DHB SEND:

Following synchronization by the CP 581, the CPU receives a request if necessary from the CPRECORD program to read the CPU data whilst utilizing the CPDHB driver to transmit the required data. To enable the CPU to correctly satisfy this request, the DHB SEND with parameterization for the function “SEND ALL” must be called in each CPU cycle as shown in Figure 7-2.

Block diagram

Use the block No.
FB xxx from
Table 2.4

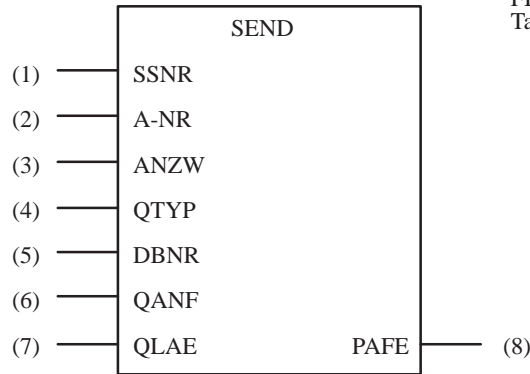


Table 7-6 Format and Meaning of Parameters for SEND/SEND ALL

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
QTYP	D	KC	These parameters are irrelevant with the function "SEND ALL"; they must nevertheless be specified.
DBNR	D	KY	
QANF	D	KF	
QLAE	D	KF	
PAFE	Q	BY	Parameter assignment error bits

Set the following parameters for the DHB SEND:

– SSNR:

Enter the number of the page via which you wish to read the data from the CPU: corresponding to the call of the DHB SYNCHRON.

– A-NR:

Job number: you must enter zero here for the function "SEND ALL."

– ANZW:

Address of **two** successive words. These words are used by the data handling blocks to store job-related status bits. **These words are deleted** during transfer with the CP 581 for **process data acquisition**.

Permissible addresses: FW 0 to 252
 DW 0 to 254

– PAFE:

As with DHB SYNCHRON.

Note

The parameters required for data transmission (QTYP, DBNR, QANF and QLAE) are made available following calling of the DHB SEND using the SEND ALL function:

The CPRECORD program determines them from the data in the configuration file. They are made available to the DHB SEND via the CPDHB driver.

Example

Data are to be transmitted on the S5-135U from a CPU 928B (the only CPU in the PLC) to the CP 581. By means of the CPRECORD function, the CPU is ordered by the CP 581 to send the data specified and simultaneously receives the corresponding parameters.

The S5 program for this job consists of the following parts:

- Calling of FB 111 in the restart blocks,
- FB 111 with calling of DHB SYNCHRON,
- OB 1 for cyclic program processing; only one call of the DHB SYNCHRON is necessary with the SEND ALL function.

Error handling will take place in the PB1 block which is not described in more detail here since error response is application-specific.

The data handling blocks to be used with the CPU 928B are listed in Table 7-2 and are numbered as follows:

DHB SYNCHRON: FB 125
 DHB SEND: FB 120 ¹⁾

- 1) For S5-135U and S5-115U programmable controllers, there are special "ALL blocks". Therefore, you can also use FB 126 (SEND ALL) instead of FB 120 (SEND).

• **STEP 5 operations in OB 20, OB 21 and OB 22:**

```

SEGMENT 1           0000

0000           :
0001           :                   Other potential S5 operations
0002           :
0003           :
0004           :SPA   FB 111       synchronization of CP 581
0005   NAME :CPSYNC
0006   REP  :       KF +3       number of SYNCHRON attempts
0007           :
0008           :
0009           :                   Other potential S5 operations
000A           :
000B           :BE
    
```

• **Function block FB 111:**

```

SEGMENT 1          0000
NAME : CPSYNC
DECL : REP          I/Q/D/B/T/C: D   KM/KH/KY/SC/KF/KT/KC/KG: KF

0008      :L      KB 0
0009      :T      FY 10      Initialize loop counter
000A LOOP :          Return marker
000B      :SPA   FB 125      Synchronize CP 581
000C NAME :SYNCHRON
000D SSNR :      KY 0,16     Page No. = SSNR = 16
000E BLGR :      KY 0,0      Frame size = 0 => 256 bytes
000F PAFE :      FY 11      Parameter error in FY 11
0010      :AN      F 11.0
0011      :JC      =END      The block is completed without error
0012      :          only if no parameter error has occurred.
0013      :L      FY 10      Increase loop counter by one on every
0014      :I      1          SYNCHRON run (abort criteria).
0015      :T      FY 10
0016      :L      FY 10
0017      :LW     =REP      Check whether the maximum permissible
0018      :<=F     number of runs has been reached.
0019      :JC      =LOOP
001A      :STS      PLC STOP following a PAFE, or if the
001B      :          set number of runs has been exceeded.
001C END   :          SYNCHRON completed successfully.
001D      :BE
    
```

• **STEP 5 operations in OB 1:**

```

SEGMENT 1          0000 Cyclic program processing
0000      :
0001      :          Other potential S5 operations
0002      :
0003      :SPA   FB 120      SEND ALL
0004 NAME :SEND
0005 SSNR :      KY 0,16     Page No. = SSNR = 16
0006 A-NR :      KY 0,0      SEND ALL identifier
0007 ANZW :      FW 12
0008 QTYP :      KS          irrelevant
0009 DBNR :      KY 0,0      irrelevant
000A QANF :      KF +0       irrelevant
000B QLAE :      KF +0       irrelevant
000C PAFE :      FY 16      Parameter error
000D      :A      F 16.0     Parameter errors occurring are dealt with
000E      :JC     PB 1       in PB 1. Error handling is dependent on the user
000F      :          and will not be explained here.
0010      :
0011      :          Other potential S5 operations
0012      :BE
    
```

7.3.4 Activation, Deactivation and Testing of Process Data Acquisition

Activation

You can activate process data acquisition once you have carried out all preparatory measures on the CPU and CP 581:

- Make sure that the entry for loading the CPDHB driver (“CPDHB”) is present in the AUTOEXEC.BAT file in the main directory of the CP 581 hard disk. ¹⁾
- Boot the CP 581 in order to start the CPDHB driver. ¹⁾
- Restart the CPU in order to synchronize it with the CP 581.
- Define the sequence options for CPRECORD. These must be specified in the command for starting the program and have the following syntax:

CPRECORD /Cconfdat /? /H /N /S /O /A

- 1) If the load command for the driver has been removed from the AUTOEXEC.BAT file for a specific reason, you can load the driver using the keyboard input “CPDHB.”

The meanings of the options are explained in the following table. You can also use “-” for the option code instead of “/”, and also lower-case letters for the option instead of upper-case letters.

Table 7-7 Meaning of Options in the Start Command for CPRECORD

Option	Meaning (notation)
C	Switch over the configuration file to a file other than the preset CPRECORD.INI in the current working directory. confdat = path and file name of the configuration file with which CPRECORD is to work (MS-DOS notation).
? or H	Output help text and explanations of CPRECORD
N	Switch off message output on screen during operation.
S	“Suspend” process data acquisition: Use this option if you wish to specifically control data acquisition from the individual S5 areas using the CP 581 or a CPU (please read Section 7.5 before using this option). Data acquisition is not started immediately in this case but only following the input of a special CPRECORD command or a job from the CPU (see Section 7.5).

Table 7-7 Meaning of Options in the Start Command for CPRECORD

Option	Meaning (notation)
O	Working with "old" configuration file: You must start CPRECORD using this option if you have already edited a configuration file for the first CPRECORD version (V1.0) and wish to use it further.
Q	Automatic mode: All acknowledgment requests are suppressed by CPRECORD (for example, if the memory on the destination device is insufficiently large when starting the acquisition). You can use the option if you wish to automatically start CPRECORD using a command entry in the AUTOEXEC.BAT file.

- Now start the program using the **CPRECORD** command and the desired options.

If you have already debugged the data process acquisition, you can also enter the command into the AUTOEXEC.BAT file. In this case the process data acquisition is automatically started when the CP 581 is run up.

- The following actions are executed on the CP 581 once the command has been entered:
 - The CPRECORD program is started by MS-DOS and the option parameters are applied.
 - CPRECORD outputs the following start message on the screen:

```
=====
CPRECORD data recording - Version x.x
Copyright (c) Siemens AG 1992
```

If you have switched on the output of messages in the configuration file, CPRECORD enters the start message of the program together with the date and time into the logbook file.

- CPRECORD searches for the configuration file CPRECORD.INI in the directory in which it is located, and evaluates this file.
- If CPRECORD does not determine any errors following the start ¹⁾, it subsequently outputs the following messages and installs itself as a TSR program:


```
<>CPRECORD installed ...
'CPRECORD \?' indicates command syntax
```

1) In the event of a major error, a corresponding error message appears on the screen (see Section 7.7). CPRECORD is not installed in this case.

- CPRECORD then checks whether there is sufficient space for **all** ASCII files (in the case of small data quantities, 2048 bytes are provided per file) on the destination drive which you have specified in the configuration file in the path name for the ASCII files.

CPRECORD outputs an acknowledgeable message if the space on the drive is sufficient. You can nevertheless start data recording in this case by pressing the key



(a suspended start is carried out if you have entered the option “S”). Press any other key if you wish to abort data recording, however.

- If you did **not** enter the option “S” (suspended) when starting, CPRECORD subsequently outputs the following message for each parameter set in the configuration file:
“[n] Data recording started”
when n = number of parameter set.
- If you have entered the option “S” (suspended) in the start command, the message “Data recording suspended” appears instead of the message lines “Data recording started.”
- If data acquisition is active, the process data of CPRECORD are transferred to the ASCII files whose path you have specified in the corresponding parameter sets of the configuration file.
- CPRECORD is executed in the background. Other programs can be active in the foreground; the reaction times are influenced, however.

If the addressed CPU is not synchronous when CPRECORD is started, it outputs the following message on the screen and stores it in the logbook file (p = number of associated parameter set):

```
=====
CPRECORD data recording - Version x.x
Copyright (c) Siemens AG 1992
[p][CP-HTB] CPU is not synchronous
CPRECORD installed ...
‘CPRECORD \?’ indicates command syntax
```

CPRECORD then scans the interface to the CPU according to the cycle time (first parameter in the configuration file) until

- the parameterized interface is synchronized by the CPU,
 - the data recording is stopped
- or
- CPRECORD is removed from the memory (deactivated).

Note

If you wish to use the CPMASS and CPRECORD functions **simultaneously**, it is essential to start CPMASS **before** CPRECORD since you cannot otherwise uninstall CPRECORD.

Deactivation

Enter the following command if you wish to deactivate CPRECORD and remove it from the memory:

CPRECORD /U

(you can also use “-” for the option code instead of “/”, and also lower-case letters instead of upper-case letters).

If the process data acquisition was still active for one or more S5 areas when this command was entered, it is aborted when the command is executed.

Testing

Proceed in steps to find out whether the data recording you require is being executed correctly:

1. Check whether the STEP 5 operations for data transfer are being executed correctly on the CPU.
2. Edit the configuration file such that data are only acquired from **one** S5 area (only one parameter set!).
3. Check that your configuration file on the CP 581 has the correct format and is logically correct (the CPRECORD program informs you of faults in the sequence by means of error messages - see Section 7.6).
4. Initially transfer static test data (data block) from the CPU to the CP 581 and check whether these reach the destination file on the CP 581 correctly converted. Activate CPRECORD without options in the start command to carry this out.

Re 1.:

In order to determine whether the STEP 5 operations programmed by you on the CPU are being correctly executed for data transfer, you can test them with the programming device online functions (handling is explained in your programmable controller manual or in the “STEP 5 Basic Package” manual, /4/). Initially activate the CP 581 **without** process data acquisition (“idling” of CPU).

The data handling blocks store flags in the agreed PAFE byte when they are called, from which you can draw conclusions on any errors:



K = Common bit:

- 0: no errors
- 1: parameter assignment error, more details in bits 4 to 7

Table 7-8 lists all error bits which can be stored by the data handling blocks in PAFE.

Table 7-8 Cause of Parameter Assignment Errors

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format
21H	DB or DX data block not present or illegal (e. g. DB 0 or DX 0 with QTYP = DB or DX)
31H	Area too small or total of initial address (QANF/ZANF) and length (QLAE/ZLAE) too large (with all QTYP/ZTYP)
41H	Area does not exist or is illegal (with QTYP/ZTYP = AS, AB, EB, PB)
51H	Status word (address) faulty
61H	Dependent on CPU
71H	Interface does not exist
81H	Interface not ready
91H	Interface overloaded
A1H	Dependent on CPU
B1H	Job No. illegal or frame size (SYNCHRON) illegal
C1H	Interface does not react, or interface does not react at correct time, or interface rejects job
D1H	Dependent on CPU
E1H	Dependent on CPU
F1H	Dependent on CPU

Evaluation of the **status word** is not meaningful with the CPRECORD function.

You must eliminate the mentioned cause of the error if one of the causes listed in the table occurs: check your STEP 5 operations for correct parameters and compare these values with the hardware configuration of your PLC and the CP 581 if applicable. Also check that the CP 581 including the CPDHB driver is ready.

You can proceed to the next test step if no errors have been indicated in the PAFE.

Re 2.:

It is sufficient for the first test of the process data acquisition to only record data from **one** S5 area and to check the data recording using these. If the test phase is concluded successfully, you can extend the configuration file by further parameter sets if required (a maximum of 8 parameter sets are processed by CPRECORD).

Re 3.:

Program a data block DB with test data, for example, the fixed-point numbers 100 to 119, on the programming device; these data can then be converted universally. Before you load the data block into the CPU, ensure that it will not be overwritten by dynamic data from your STEP 5 program for the test.

Edit the following parameters in the CPRECORD.INI configuration file:

- Acquisition cycle:
enter 30 seconds as the time so that the test data are available quickly.
- Values for **one** (the first) parameter set:
 - S5 area:
enter 'DB' here.
 - DB/DX number:
enter the DB No. under which you have created the data block and loaded into the CPU, for example, 20.
 - Number of files:
only specify one file for the test.
 - Number of data records:
specify a clear number, for example, 20.
 - S5 format:
enter the data format with which you wish to convert all the data in the DB universally, for example, KF.
 - Cycle mode:
enter '0' for "non-permanent" so that the acquired data are not overwritten during evaluation.
 - Permanent:
switch on the storage of messages in a logbook file using '1.'
 - Logbook file:
enter the path and name for the logbook file,
for example, C:\CPRECORD.LOG
 - Timeout:
enter 20 seconds.

You must specify all the above parameters according to your test plan. Then start process data acquisition.

If the CPRECORD program determines errors or faults during interpretation of the configuration file or during process data acquisition, it stores corresponding error messages in the set logbook file (see Section 7.7).

Re 4.:

If you have carried out test steps 1 and 2, and if no error bits are registered on the CPU and CP, check whether the data have been completely transferred from the CPU and correctly converted (read in Section 7.4 how the data are stored in ASCII files):

To do this, log the ASCII file on the CP 581 on a printer using the MS-DOS command PRINT and compare the log with that of the data block from the programming device. Data transmission is correct if the data are identical.

Once you have carried out steps 3 and 4 using data converted universally, repeat these steps with “individual” conversion using a data block in which mixed data formats occur (you must of course match the configuration file to the modified test conditions). To do this you must either edit a format file (see Section 7.3.2) or transfer the S5 file generated on the programming device with the data and preheader information of the data block to the CP 581.

7.4 Evaluation of Acquired Process Data

This section tells you how the process data are stored on the CP 581 and in which files you can find the S5 data again.

7.4.1 Storage of Process Data on the CP 581

What File Names are Used?

The names of the ASCII files are generated from the data type and a four-digit consecutive number. The file number extension is taken from the configuration file.

Table 7-9 Names of ASCII Files

Assignment "data type/file name"	
Data Type	File Name ¹⁾
CA	ASmmmm.eee
DB	RSmmmm.eee
DX	nnnDmmmm.eee
DX	nnnXmmmm.eee
AS	QAmmmm.eee
IA	IAmmmmm.eee
PY	FAmmmmm.eee
QA	PYmmmm.eee
RS	TAmmmmm.eee
TA	CAmmmmm.eee

- 1) nnn = data block No. of DB/DX
 mmmm = consecutive file number
 eee = file name extension from configuration file

Important!

If you record data from S5 areas with the same name from several CPUs, identical names may occur in the ASCII files and result in destination file data being overwritten! You can prevent this by assigning either different destination directories or different name extensions for the ASCII files when editing the configuration file in the parameter sets.

Example: File names for DB 20 with file name extension 'TXT:'

020D0000.TXT for 1st ASCII file
020D0001.TXT for 2nd ASCII file
020D0002.TXT for 3rd ASCII file
020D0003.TXT for 4th ASCII file, etc.

How are the ASCII Files Written?

Following a warm system restart on the CP 581 (for example, after a power failure) and starting of process data acquisition, CPRECORD searches for the most recently written ASCII file assigned the consecutive number "n" using the time stamp (MS-DOS entry in file directory) independent of the mode of acquisition. The newly acquired process data are then stored in a new file with the number "n+1." If no data were initially present for the type of data acquired, the file is first created with the consecutive number "0."

If the last file (corresponding to the number of files in the configuration file) also becomes full during acquisition, **or if no more memory space is available**, the procedure is continued depending on which acquisition mode you have specified in the configuration file:

- "Permanent recording:"
in the next acquisition cycle, the file with the consecutive number "0" is deleted and recreated (the old data are then lost and rewritten). Once this has been filled, the next file is recreated etc., i.e. the data are stored in a type of cyclic process.
- "non-permanent recording:"
acknowledgment request or end of recording (see also Section 7.6.2).

7.4.2 Structure of Process Data in the ASCII Files

What is a Data Record?

All data which are read by the CPU and stored converted in an ASCII file within **one** acquisition cycle constitute a data record. A data record is terminated by the ASCII characters CR and LF (see Figure 7-3).

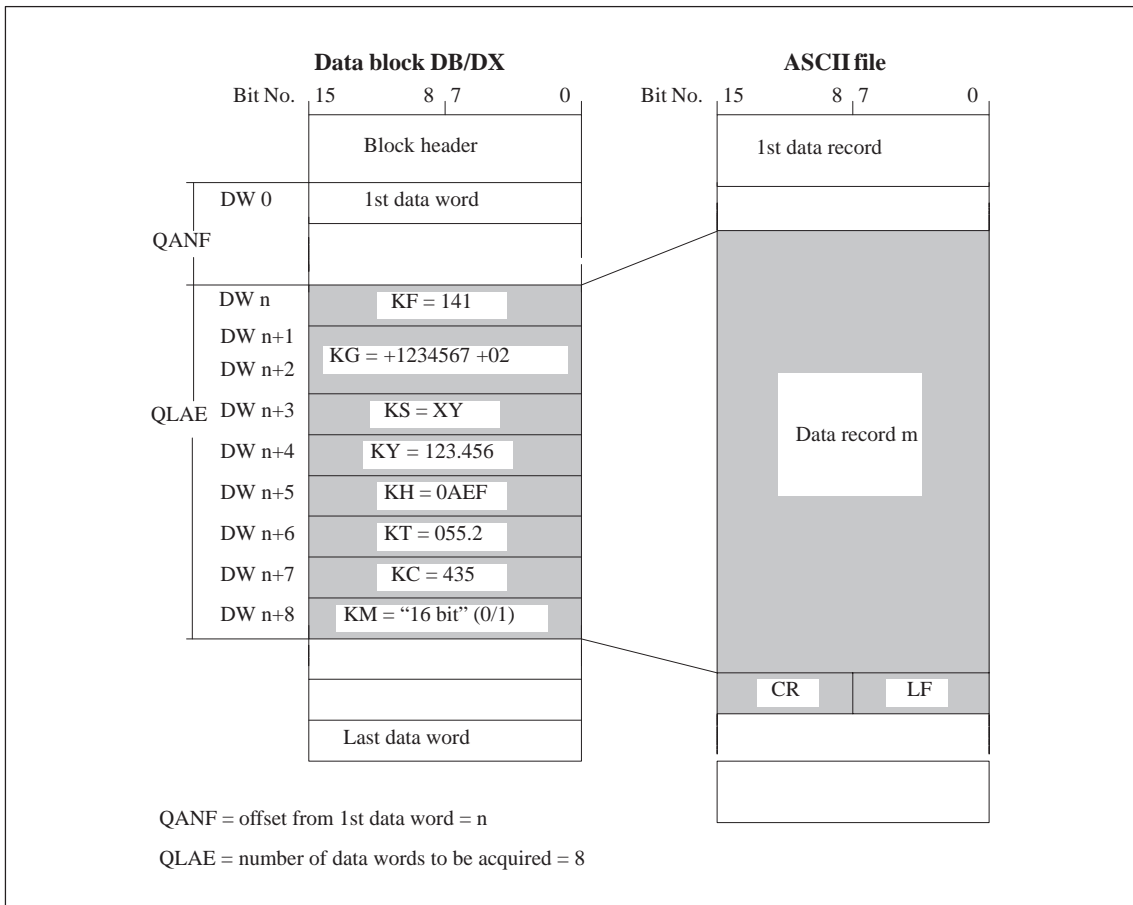


Figure 7-3 Storage of a Data Record in an ASCII File

What is the Data Record Format?

A data record contains the individual process data from the data area of the CPU which you defined. Each item of data is separated from the next by the field delimiter defined in the configuration file.

Example

(corresponding to Figure 7-3):

```
"^+141*+1.234567e+001*XY*123*456*0AEF*^5500*435*^4711(CRLF)"
^= space, * = delimiter
```

7.4.3 Converting the Individual Data

The individual data of the S5 area are converted according to the following procedure:

- “Individual” conversion using preheader data of a DB/DX or using a format file edited by you.
- “Universal” conversion of a complete S5 area.

Individual conversion:

In the case of individual conversion, the read S5 area is divided into individual data using the information in the preheader data or the format file on the data type and type repetition, and these individual data are converted corresponding to the type.

The following table shows you the results obtained in the ASCII file from individual conversion and the S5 areas for which you can use individual conversion.

Table 7-10 Individual Conversion of S5 Data

Data in S5 area	Characters in ASCII file ¹⁾	Array length	Application for S5 area
Format instruction ‘KS’ ²⁾			
KS = AXBYCZ	“AXBYCZ”	2 * repetition factor in format instruction	QA, DB, DX, IA, FA, PY
Format instruction ‘KF’			
KF = +257	“^^+257”	6 characters, leading spaces	QA, DB, DX, IA, FA, PY
Format instruction ‘KG’ ³⁾			
KG = +5500000+01	“+5.500000e+000” e = characters for exponent Mantissa: sign, 1 digit before decimal point, 6 digits after decimal point Exponent: sign, 3 digits	14 characters, leading zeros	DB, DX, poss. FA
Format instruction ‘KH’			
KH = 073E	”073E”	4 characters	All areas
Format instruction ‘KM’			
KM = 0000 0111 0011 1110	The value of the bit pattern (0 to 65 535) is converted into an unsigned decimal number: “^1854”	5 characters, leading spaces	All areas
Format instruction ‘KT’			

Table 7-10 Individual Conversion of S5 Data

Data in S5 area	Characters in ASCII file ¹⁾	Array length	Application for S5 area
KT = 032.2	The time value is calculated in hundredths of a second from the "Value" and the "Timebase:" "^^^3200"	7 characters, leading spaces	DB, DX, TA
Format instruction 'KY'			
KY = 007,062	Two successive bytes of the S5 area are converted into two 3-digit decimal numbers and stored in two arrays : "^^7"delimiter"^^62"	2 * 3 characters, leading spaces	All areas
Format instruction 'KZ'			
KC = 032	"^32"	3 characters, leading spaces	DB, DX, CA

1) ^ = space

2) A string is stored in the ASCII file with the format "KS." The end criterion of the file is:

- a) a change in format,
- b) a binary zero in the S5 data stream.

3) Only those S5 data are correctly converted into floating-point numbers which have been generated either via a KG format with the data handling block editor on the programming device (with data blocks) or with an S5 floating point operation.

Universal Conversion

The individual data are converted by a universal conversion in the same manner as with individual conversion. In this case, however, the complete S5 area read is converted with **the same format**.

The following format symbols are possible:

<KB>, <KS>, <KF>, <KG>, <KH>, <KM>, <KT>, <KY> and <KC>.

Each data unit from the S5 area is mapped in a character array in the ASCII file. An exception to this procedure is the format instruction KS: since in contrast to all other formats neither the length of the S5 data nor the length of the string following conversion are fixed in this case (in the case of individual conversion they are defined by the repetition factor), the complete S5 area read is mapped in **one** character array. Each data record in the ASCII file therefore contains only **one** character array with the universal format instruction "KS."

Note

The information in a format file refers to the **complete S5 area**, for example, to the flag area **FA 0 to FA 255**. The information depends on which S5 area you have defined for the **transfer** in the configuration file by means of the parameters "Offset" and "Number."

The **first format instruction** refers to the **first word** or **double word** in the defined S5 area.

An **exception** is the S5 area **AS**: In this case the **first format instruction** refers to the **first word** in the AS area which is **to be transferred** to the CP 581.

If the S5 data area is **longer** than the result of all format instructions, the **excess data** is universally converted using "**KF**."

7.4.4 Example of “Individual” Conversion

Table 7-11 shows you an example of how S5 data (for example, from a data block) are converted in a format file using format instructions. “*” is declared as the field delimiter.

Table 7-11 Example of Individual Conversion of S5 Data

S5 data block		Format file/ format instruction	ASCII file	
DW No.	Contents of DW		Field No.	ASCII characters
0	KF = +123	;3 fixed-point numbers: 3KF	0	^^+123*
1	KF = -4567		1	^-4567*
2	KF = +32123		2	+32123
3	KG = +5500000+01	;4 floating-point numbers: 4KG	3	+5.500000e+000*
4				
5	KG = +5500000+03		4	+5.500000e+002*
6				
7	KG = -3410000-02		5	-3.410000e-003*
8				
9	KG = -1234567+00		6	-1.234567e-001*
10				
11	KS = EX	;6 characters: 3KC	7	EXAMPL*
12	KS = AM			
13	KS = PL			
14	KM = 1111 1010 0000 0000	;2 bit patterns: 2KM	8	64000*
15	KM = 0000 1111 0000 1111		9	^3855*

7.5 Control of Process Data Acquisition

You can use the following procedure if you do not wish to let process data acquisition run continuously following activation of CPRECORD but to only use it for specific S5 data areas:

1. Switching on and off of acquisition by a CP 581 command (Section 7.5.1).
2. Switching on and off of acquisition by a CPU (Section 7.5.2).
3. Triggering of single acquisition by a CPU (Section 7.5.2).

For procedures 1) and 2) you must suspend the activation using option “S” when activating CPRECORD (see Section 7.3.4), data acquisition must **not** be suspended for procedure 3). The data are stored as described in Section 7.4.1 irrespective of whether you use the control procedures or not.

7.5.1 Control of Acquisition by a CP 581 Command

The control program CPRECCTL (CTL = control) is available to control the acquisition by a CP 581 command. You can use this to switch the (for example, suspended) acquisition on and off for one, several or all parameter sets.

Use the following commands:

- Switch on acquisition: ¹⁾

CPRECCTL /B<p>

- Switch off acquisition: ¹⁾

CPRECCTL /E<p>

p = identification of parameter set or sets; **all** parameter sets are switched on or off if “p” is not specified.

- 1) You can also use “-” instead of “/”, and lower-case letters instead of upper-case letters. B (b) is for “begin” and E (e) for “end.”

Notation of Code 'p'

- p = decimal number for **one** parameter set ($1 \leq \text{decimal number} \leq 8$)
- p = hexadecimal number 0Xhh for **several** parameter sets. The number of a parameter set is then recorded as a bit with the value "1" of a byte. The hexadecimal number results from linking all bits of the desired parameter sets using the OR operation.

Assignment between parameter set No. and bit:

- 1 = 0000 0001
- 2 = 0000 0010
- 3 = 0000 0100
- 4 = 0000 1000
- 5 = 0001 0000
- 6 = 0010 0000
- 7 = 0100 0000
- 8 = 1000 0000

Command Examples

- CPRECCTL /B3 The acquisition for parameter set 3 is switched on (decimal number '3').
- CPRECCTL /B0X88 The acquisition for parameter sets 4 and 8 is switched on (hexadecimal number from bit pattern '1000 1000').
- CPRECCTL /E7 The acquisition for parameter set 7 is switched off (decimal number '7').
- CPRECCTL /E0X0F The acquisition for parameter sets 1 to 4 is switched off (hexadecimal number from bit pattern '0000 1111').
- CPRECCTL /E The acquisition is switched off for all parameter sets.

7.5.2 Control of Acquisition by a CPU

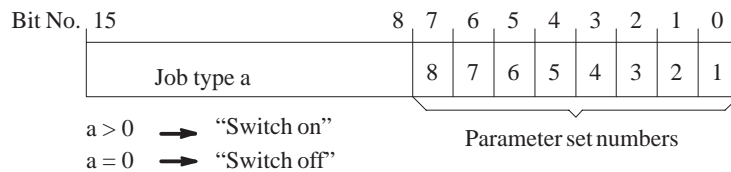
You can also influence the acquisition by the CPUs from which you collect process data on the CP 581: use the data handling block DHB SEND, function SEND DIRECT, in order to send specific control jobs to the CP 581. You must program the required calls of the DHB SEND in addition to the call described in Section 7.3.3 (SEND ALL). You should have experience in STEP 5 programming for this.

Switch Acquisition On/Off

(A-NR = 210)

You must call the DHB SEND/SEND DIRECT with the job number **210** in each case to switch on and off. You can define in a job control word (flag or data word) whether the acquisition is to be switched on or off and for which parameter sets the job is valid.

Format of job control word for switching on/off:



Trigger Acquisition

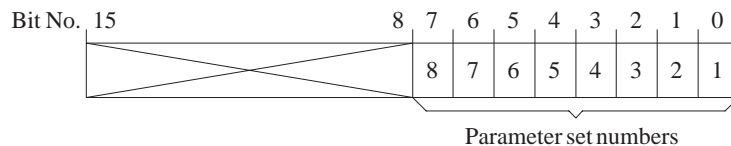
(A-NR = 211)

You must **not** use the option "S" (suspended) when activating CPRECORD if you wish to trigger the acquisition of process data from a CPU. You must enter "0" for the acquisition cycle parameter in the configuration file so that the acquisition does not immediately begin when CPRECORD is started.

Furthermore, the acquisition must **not be switched off** or - in "non-permanent recording" mode - **not have been terminated**.

Triggering is carried out by calling the DHB SEND/SEND DIRECT using job number **211**. You can define the parameter sets for which the job is valid in a job control word (flag or data word).

Format of job control word for triggering:

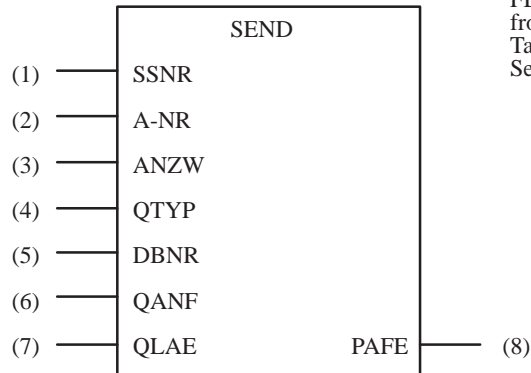


Calling the DHB SEND/Function DIRECT

Incorporate the SEND DIRECT calls into your STEP 5 program such that they send a specific control job to the CP 581 depending on a process event (for example, input signal) with which you wish to influence the acquisition.

You must store the bit information for controlling the acquisition in the job control word before you call the DHB SEND/SEND DIRECT.

Block diagram



Use the block No. FB xxx from Table from Table 2.4 in Section 2.3.3

Table 7-12 Format of Parameters for SEND/SEND DIRECT

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
QTYP	D	KC	Data type of source
DBNR	D	KY	Number of data block if source is data block
QANF	D	KF	Offset of first item of data to be read in the data area
QLAE	D	KF	Number of data units to be read
PAFE	Q	BY	Parameter assignment error bits

Set the following current parameters for the call:

– SSNR:

Enter the same number as you programmed for the process data acquisition when calling the DHB SYNCHRON.

– A-NR:

Enter the following values as the job number:

210 to switch the acquisition on or off

211 to trigger a single acquisition

– ANZW:

Address of two successive words. These words are used by the data handling blocks to store job-related status bits.

– QTYP:

Enter either FA (flag area) or DB/DX (data block DB/DX) as the data type in which you have stored the job control word.

– QANF:

Enter the number of the flag or data word in which you have stored the job control word.

– QLAE:

Always enter the number '1' with QTYP = DB/DX or '2' with QTYP = FA since the data to be transmitted only consists of the job control word.

– PAFE:

Byte address for parameter assignment error bits, for example, FY 216. Bit No. 0 of the status byte is set to '1' in the event of a parameter error (refer to Section 7.3.4 for the meaning of the status bits.)

7.6 Information on Special Handling

This section explains how you can obtain parameter and status information on the process data acquisition, and what you must do in certain cases.

7.6.1 Information on Parameters and Status

The system service routine CPRECCTL is available to provide you with information on the process data acquisition in a similar manner to the control. You can use this routine:

- To examine parameter sets of the configuration file,
- To examine the acquisition status of a specific parameter set,
- To obtain information on the syntax of all CPRECCTL commands.

You can only obtain information if you have activated the CPRECORD program!

Notation of information commands:

You can also use “-” for the option code instead of “/”, and also lower-case letters for the command and option instead of upper-case letters.

Examination of Parameter Set of the Configuration File

Enter the command

CPRECCTL /Cp

and enter the number of the parameter set you wish to examine for “p.” The cycle time and all parameters of the selected parameter set from the current configuration file then appear on the screen starting at the current cursor position (see following figure).

```

( 1)  Cycle time (sec.) = ...
[p] (2)  CPU number = ...
[p] (3)  S5 area = ...
[p] (4)  DB/DX number = ...
[p] (5)  Data offset = ...
[p] (6)  Data length = ...
[p] (7)  Data recording directory = ...
[p] (8)  Output file extension = ...
[p] (9)  Number of files = ...
[p] (10) Number of data records = ...
[p] (11) S5 format = ...
[p] (12) Field delimiter = ...
[p] (13) Permanent = ...
[p] (14) Logging = ...
[p] (15) Log file = ...
[p] (16) Timeout (sec.) = ...

```

Figure 7-4 Display of Configuration Parameters on the Screen
(p = number of parameter set)

View Current Status

Enter the command

CPRECCTL /Sp

and enter the number of the parameter set you wish to examine for “p.” The cycle time and all parameters of the selected parameter set from the current configuration file then appear on the screen starting at the current cursor position (see following figure).

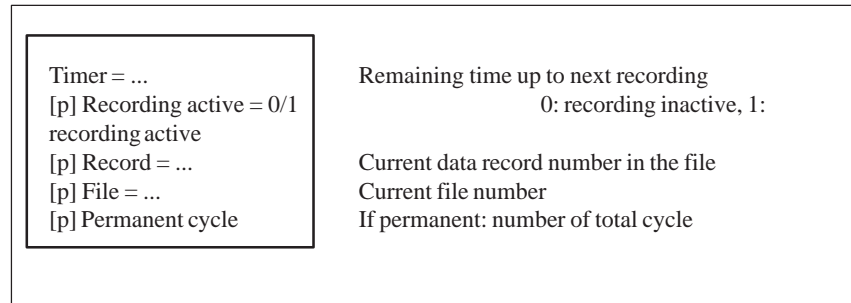


Figure 7-5 Display of current status for a parameter set on the screen (p = number of parameter set)

Obtaining Information on the Syntax of all CPRECCTL Commands

Enter the command

CPRECCTL /H or CPRECCTL /?

Examples of Information Commands

- CPRECCTL /C5 Output configuration parameters from parameter set 5
- CPRECCTL /C8 Output configuration parameters from parameter set 8
- CPRECCTL /S7 Output status for parameter set 7
- CPRECCTL /S1 Output status for parameter set 1

7.6.2 Reactivating Process Data Acquisition

If you record data from an S5 area in non-permanent mode (code “0” in the configuration file), CPRECORD terminates the recording for this area when the maximum number of ASCII files and the maximum number of data records in the last ASCII file have been reached. You are informed of this by the message

[n] Data recording finished

where “n” is the number of the parameter set for which the data acquisition was terminated.

In the following cases you must reactivate the process data acquisition for further recording following such a message:

- Only data from one S5 area were acquired (only **one** parameter set in the configuration file).
- The recording for all other parameter sets has already been terminated.
- You wish to restart the acquisition for the S5 area whose recording has been terminated.

Proceed as follows:

- Remove the CPRECORD program from the memory using the command **CPRECORD /U** (any parameter sets for which the acquisition is still running are deactivated by this command).
- Reactivate the process data acquisition as described in Section 7.3.4 (command CPRECORD with the desired options).

The procedure is executed as follows if CPRECORD determines during reactivation that ASCII files already exist for an S5 data area:

1. The volume of the ASCII files for a parameter set has not yet been used up:

CPRECORD continues the acquisition for the parameter set. The data are stored further in the ASCII files as described in Section 7.4.1.

2. The volume of the ASCII files for a parameter set has been used up:

- The recording is to take place in mode “permanent recording:” CPRECORD begins the recording new. The data are restored in the files as described in Section 7.4.1.

- The recording is to take place in mode “Terminate recording:”

If CPRECORD was started **with the option ‘A’** (automatic), the acquisition for the parameter set is terminated immediately (without message).

If CPRECORD was started **without the option ‘A,’**

CPRECORD outputs the following acknowledgment message on the screen:

[p] Data recording finished!

[p] Type <ENTER> to continue anyway

Then press the key <ENTER> if you wish to restart the acquisition (the old data are overwritten!).

Press any other key if the acquisition is not to be started.

7.7 Status Messages

If special events are determined during process data acquisition, they can be entered as messages in a logbook file on the CP 581 (see Section 1.3.2. "Editing the Configuration File").

Messages are generated by the CPDHB driver and the CPRECORD program. The following two tables show you which messages can occur and what they mean (p = number of associated parameter set).

Table 7-13 CPDHB Driver Messages

Message
[p] [CPDHB] CPU not synchronized
[p] [CPDHB] Aborted - data area not accessible
[p] [CPDHB] Aborted by SYNCHRON
[p] [CPDHB] Aborted by RESET
[p] [CPDHB] Aborted by timeout
[p] [CPDHB] Aborted - parameter error
[p] [CPDHB] Parameter invalid (PAFE 1)
[p] [CPDHB] DB/DX does not exist (PAFE 2)
[p] [CPDHB] Area too small (PAFE 3)
[p] [CPDHB] Area does not exist (PAFE 4)
[p] [CPDHB] Status word error (PAFE 5)

Table 7-14 CPRECORD Program Messages

Message	Remarks
Unknown option	Incorrect option in command CPRECORD
Data recording suspended	With option "S"
Configuration file does not exist: <Config file>	<Config file> = name of configuration file
Error when opening configuration file <Config file>	MS-DOS output, <Config file> = name of configuration file
CPRECORD.INI not found in current directory	File was deleted or current directory is incorrect
Configuration file extension is not ".ini"	Change name of configuration file
No parameter set has been found	
CPDHB driver not loaded	Load driver
CPRECORD file cannot be removed from memory	MS-DOS output
[p] Data recording started -<Date><Time>	Message at beginning of process data acquisition
[p] Data recording finished -<Date><Time>	Message at end of process data acquisition
[p] Error opening output file	MS-DOS output

Table 7-14 CPRECORD Program Messages

Message	Remarks
[p] Format error in configuration file with parameter <n>	<n> = line number of faulty parameter
[p] Invalid output drive	MS-DOS output
[p] Insufficient drive capacity (<Actual value> <Required value>)	MS-DOS output
[p] S5D file: could not be opened	MS-DOS output
[p] S5D file: read error	MS-DOS output
[p] S5D file: not a root directory	Format error
[p] S5D file: too many subdirectories (> 128)	
[p] S5D file: not a DV subdirectory	
[p] S5D file: positioning error	MS-DOS output "Seek Error"
[p] S5D file: too many data elements (> 128)	
[p] S5D file: the configured DV block is missing	
[p] S5D file: wrong block ID in DV preheader (<Code>)	Format error, <Code> = incorrect block code
[p] S5D file: wrong block number in DV preheader (<Number>)	Format error, <Number> = incorrect block number
[p] S5D file: DV preheader address too large (> 4095)	Format error
[p] S5D file: Format error, first format = "empty"	
[p] Output drive full	MS-DOS output
[p] Error when writing output file	MS-DOS output
[p] Output directory not found	MS-DOS output
[p] Configuration file: unknown S5 area	
[p] Insufficient number of available ¹⁾ DOS clusters (<Actual value> <Required value>)	
[p] FMT file: could not be opened <FMT file>	MS-DOS output <FMT file> = name of format file
[p] S5 file: unknown format in line <n> (<Format>)	<n> = line No., <Format> = incorrect format
[p] S5 file: format error in line <n> (<Format>)	<n> = line No., <Format> = incorrect format
[p] S5 file: too long in line <n> onwards (<Format>)	<n> = line No., <Format> = incorrect format
[p] S5D file format is permissible only with S5 area DB or DX	
[p] Error when opening log file <Log file>	<Log file> = name of logbook file
[p] Warning: cycle overflow > 10%	
[p] Parameter set does not exist	

1) Cluster = contiguous logical memory area on drive

Example of an Error Message:

[3] Warning: cycle overflow > 10 % - Wed Jul 22 12:21:04 1992

Mass Storage Functions

This chapter is intended for users who wish to use the CP 581 as an additional storage medium for a CPU. The chapter describes all measures and operations required on the CP 581 and on the CPU:

Only a single setting is required on the CP 581 (see Section 8.3.2). You must match your STEP 5 programs according to the required functions on the CPUs with which you wish to use the mass storage functions (see Section 8.3.3). For this you should have experience in programming programmable controllers.

This chapter also provides information on the response of the mass storage functions on the occurrence of faults.

The programming example in Section 8.3.3 only serves to explain the DHB parameters for CPMASS. You can find complete programs which can be executed in Chapter 12.

Chapter Overview

Section	Description	Page
8.1	Application	8-2
8.2	Principle Sequences Between CPU and CP	8-3
8.3	Mass Storage Function Operations	8-7
8.3.1	Related Procedures	8-7
8.3.2	Measures on the CP	8-7
8.3.3	Programming the CPU	8-8
8.3.4	Activation and Testing of the Mass Storage Functions	8-26
8.4	Error Bits	8-29
8.4.1	Parameter Assignment Error Bits	8-29
8.4.2	Job Status Bits	8-31
8.5	Setting of Timeout	8-33

8.1 Application

If the memory media existing on a CPU of your programmable controller are insufficient you can transfer data to the CP 581 using the mass storage functions and fetch these back - selectively if necessary - to the CPU as required. You can buffer data areas of up to 4 CPUs inserted in the same programmable controller as the CP 581 on the hard disk (or also on a floppy disk) of the CP 581 in binary form. Four hard disk directories (CPU1 to CPU4) are preset on the CP 581 for these CPUs.

You can use a CPU to change its preset directory by specifying a drive and/or a special directory for data storage.

The mass storage functions also provide you with the facility for deleting one or all of the mass storage files (S5F files<\$I S5F files>) of a directory on the hard disk of the CP 581 using one CPU.

8.2 Principle Sequences Between CPU and CP

Figure 8-1 shows you the basic sequence of a mass storage function on the CPU and CP 581. Figures 3.2 to 3.4 show you the basic data transfer between the CPU and CP 581, the switching over of a directory and the deletion of S5F files.

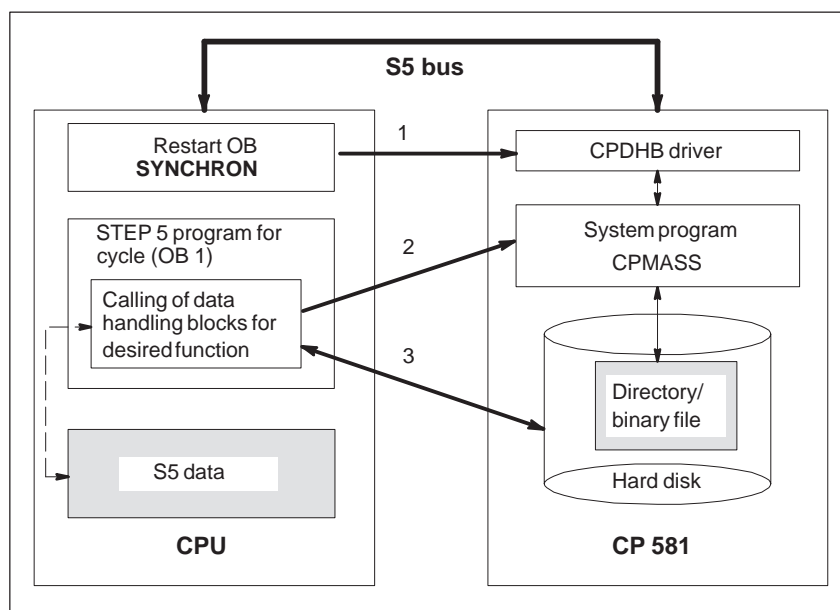


Figure 8-1 Sequence when Transmitting Data from CPU 1 to the CP 581

- 1 Synchronization of CPU during restart with CP via DHB SYNCHRON.
The frame size for subsequent data transfer is set in the process.
- 2 Triggering of mass storage function (SEND DIRECT).
- 3 Execution of mass storage function (SEND ALL).

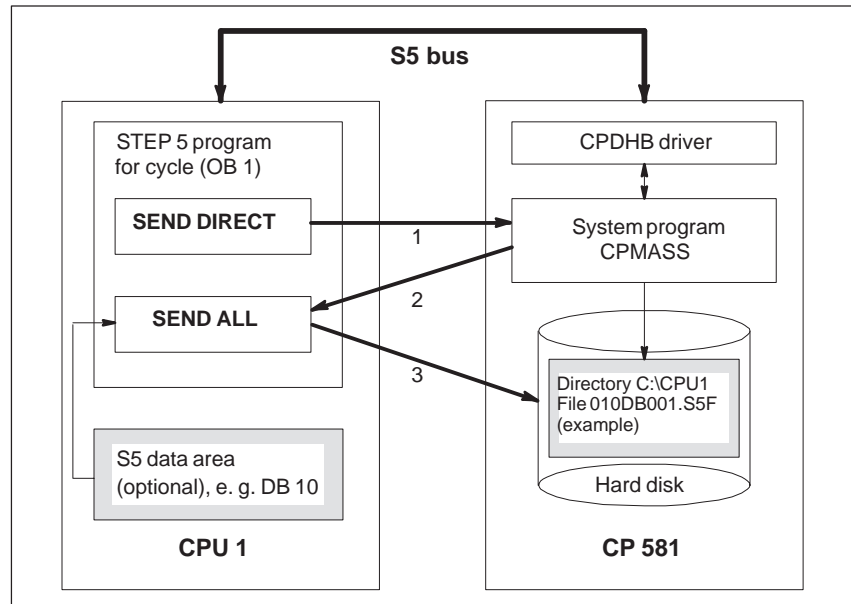


Figure 8-2 Sequence when Transmitting Data from CP 581 to the CPU 2

- 1 Call of DHB SEND/function SEND DIRECT. Here, among other things, the source parameters (including data block DB 10) are defined. By calling the DHB SEND, the S5 program activates the CP 581 and transfers the address of the data to the CPMASS system program via the CPDHB driver. The CPMASS system program generates the file name for storing the data from the transferred address.
- 2 Call of DHB SEND/function SEND ALL. The DHB SEND is activated for transmission of S5 data by the CPMASS system program via the CPDHB driver and provided with the address of the data.
- 3 SEND ALL transmits the data from CPU1 (example) out of the S5 data area to the CP 581 and stores them in the file 010DB001.S5F (example) in the directory CPU1\ (example).

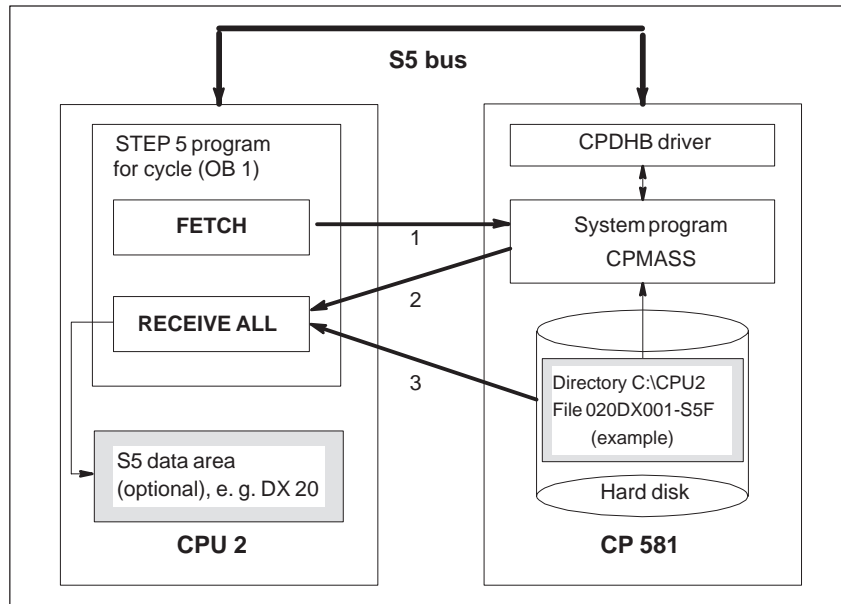


Figure 8-3 Sequence when Switching Over a Directory or when Deleting S5F Files on the Hard Disk of the CP 581 from CPU 3

- 1 Call of DHB FETCH. Here, among other things, the destination parameters (including data block DX 20) are defined. By calling the DHB FETCH, the S5 program activates the CP 581 and transfers the address of the data to the CPMASS system program via the CPDHB driver. The CPMASS system program generates the file name for reading the data from the transferred address.
- 2 Call of DHB RECEIVE/function RECEIVE ALL: The DHB RECEIVE is activated by the CPMASS system program via the CPDHB driver to read the data from the CP 581 and provided with the destination address.
- 3 RECEIVE ALL reads the data from the CP 581 in the directory CPU2\ (example) from the file 020DX001.S5F and transmits these to CPU 2 (example) into data block DX 20 (example).

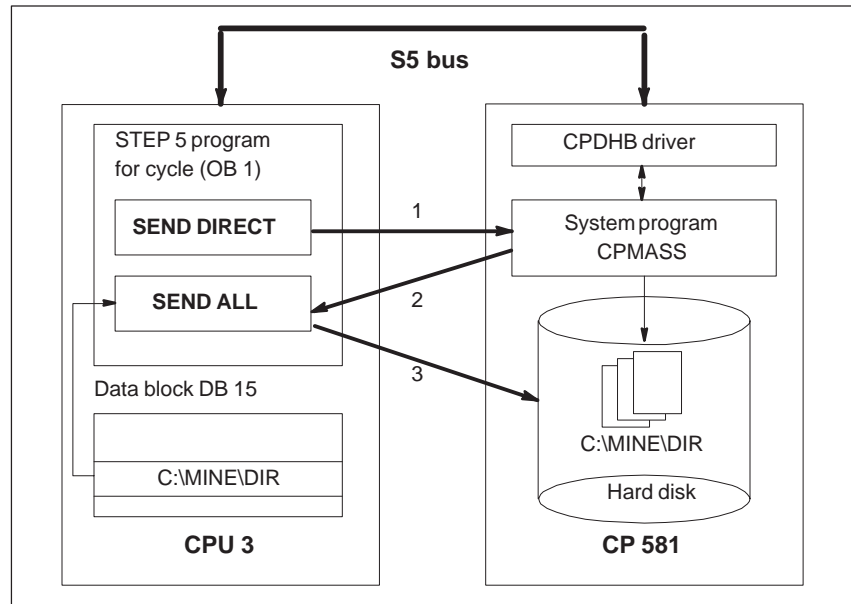


Figure 8-4 Sequence when Switching Over a Directory or when Deleting S5F Files on the Hard Disk of the CP 581 from CPU 3

- 1 Call of DHB SEND/function SEND DIRECT. Here, among other things, the address of a string (here in DB 15) is specified. By calling the DHB SEND, the S5 program activates the CP 581 and transfers the string with the path name of the directory to the CPMASS system program via the CPDHB driver.
- 2 The path name is transmitted to the CP 581.
- 3 Once all characters of the path name have been transferred to the CPMASS system program, this carries out the desired function (switchover to directory or delete S5 files from directory).

8.3 Mass Storage Function Operations

8.3.1 Related Procedures

You must carry out the following measures in the indicated sequence in order to transfer data between a CPU and the CP 581:

Table 8-1 Measures to Activate the Mass Storage Functions

Measure	on	See section
Make sure that the CPU with which you wish to exchange data and the CP 581 are plugged into your programmable controller. The base interface No. for data transfer must be set on the CP 581.	PLC CP 581	8.3.2
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG CPU	8.3.3
Create or modify the STEP 5 program on the CPU for the cycle according to your applications of the mass storage functions (for example, transmit or fetch data depending on a certain operating state).	PG CPU	8.3.3
Carry out a test on the CPU to make sure that the DHBs to be called in the restart/warm restart or cyclically are actually called correctly.	PG CPU	
Make sure that the CPDHB driver is loaded in the memory of the CP 581 (cold restart of CP 581 if necessary). The mass storage functions are automatically started with a cold or warm restart (AUTOEXEC.BAT entry).	CP 581	--
Synchronize the CPU with the CP 581 by triggering a restart of the CPU.	CPU	--
Check the correct sequence of desired functions by testing on the CPU and checking the file directories on the CP 581.	PG CPU CP 581	

8.3.2 Measures on the CP

Only one preparatory measure is required on the CP to use the mass storage functions:

Setting the base interface number:

So-called pages (page frames) are used to address the memory areas when transferring data via the S5 bus. These pages have a fixed assignment to the modules involved with data transfer.

The CP 581 can transfer data with up to four CPUs via four successive pages. The number of the first page is the base interface number.

Define this number for the first inserted CPU and set it on the CP 581 as described in Section 6.2.3. The pages for data transfer with three further CPUs then have consecutive numbers following the page with the base interface number.

8.3.3 Programming the CPU

Principle

Programming of the CPU for data transfer comprises the synchronization of the CP 581 during the restart and the calling of special function blocks for specific applications of the mass storage functions. You require so-called “data handling blocks” (DHB) for both functions: the DHB SYNCHRON for synchronization. The other DHBs which you additionally require, and where these must be called, depend on your special application of the mass storage function.

Figure 8-5 shows you the positions in your STEP 5 program at which you must call the two data handling blocks.

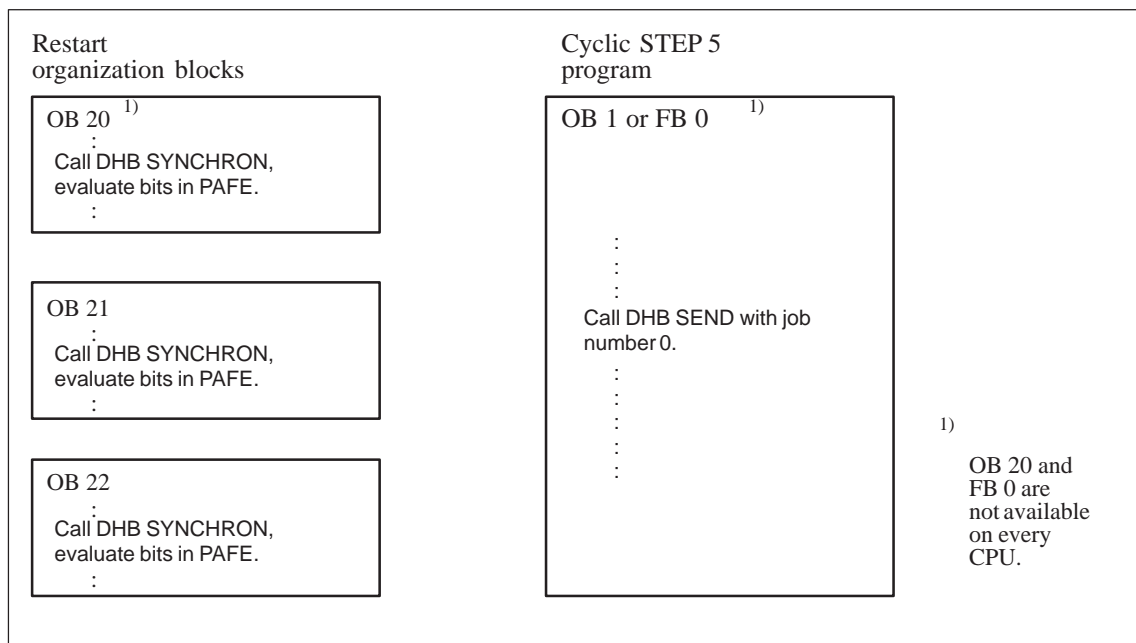


Figure 8-5 Principle of DHB Calls in the STEP 5 Program of the CPU with the Mass Storage Functions

Before you commence with programming or adaptation of your STEP 5 program, consider the following:

- Which mass storage functions are to be used?
- Which events or operating states are to activate the individual mass storage functions?
 - Following a cold restart (activation in the cycle, for example, following scanning of bits),
 - Following a manual warm restart (activation in the cycle, for example, following scanning of bits),
 - Following an automatic warm restart (activation in the cycle, for example, following scanning of bits),
 - Depending on an input signal (activation in cycle following scanning of the input signal).

Once you have answered these questions, you can read in the following sections which data handling blocks you require and which parameters must be assigned to these for the respective application.

Table 8-2 shows you the function block numbers of the DHBs for the various programmable controllers. ¹⁾

- ¹⁾ If you use an S5-135U or S5-155U, you can also use the DHBs SEND-A and RECEIVE-A which are available there instead of the DHBs SEND and RECEIVE with the function ALL.
 You can obtain information on these DHBs from the corresponding descriptions of the DHBs for the S5-135U /8/ and the S5-155U /9/ programmable controllers.

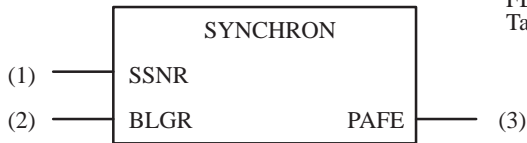
Table 8-2 DHB Numbers on the Various Programmable Controllers

	S5-115U	S5-135U	S5-155U
DHB SYNCHRON	FB 249	FB 125	FB 125
DHB SEND	FB 244	FB 120	FB 120
DHB RECEIVE	FB 245	FB 121	FB 121
DHB FETCH	FB 246	FB 122	FB 122

Synchronization of the CPU

DHB SYNCHRON

Block diagram



Use the block No. FB xxx from Table 8-2.

Table 8-3 Format and Meaning of Parameters for SYNCHRON

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
BLGR	D	KY	Frame size
PAFE	A	BY	Parameter Assignment Error Bits

Set the following parameters for the DHB SYNCHRON:

– SSNR:

Enter the number of the page via which you wish to read the data from the CPU: “(CPU No. - 1) + base interface No.” (see Section 8.3.2).

Permissible values: 0 to 255 ¹⁾

– BLGR:

Use this parameter to define the maximum number of bytes to be transmitted to the CP 581 in a CPU cycle.

Permissible values: 0 = standard values (see table)
 1 = 16 bytes
 2 = 32 bytes
 3 = 64 bytes
 4 = 128 bytes
 5 = 256 bytes
 6 = 512 bytes

Standard values for BLGR = 0	
S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

– PAFE:

Byte address for parameter assignment error bits. Bit No. 0 of the status byte is set to “1” in the event of a parameter assignment error (refer to Section 8.4.1 for the meaning of the other status bits).

Note

The CP 581 may require more time to run up than the CPU. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Make sure, however, that the program loop is **always** terminated by an abort criterion (timer or loop counter).

Longer data blocks are transmitted faster if you set the frame size BLGR larger, but the S5 bus is also loaded more. Vice versa the bus loading is less with smaller frame sizes, but data transmission to the CP 581 then takes longer. You must decide which frame size is most favorable for your CPU by considering the complete operation on the programmable controller.

1) The base interface No. must be set in steps of four (0, 4, 8, 12 etc.)!

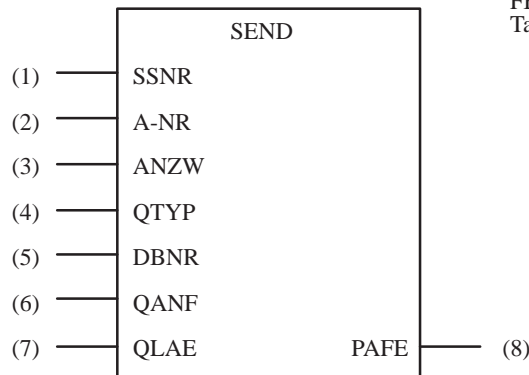
Transmission of Data from CPU to CP 581/Delete S5F File

In order to transmit data from the CPU to the CP 581 or to delete a mass storage file (S5F file), you must program the call for DHB SEND twice in the cyclic STEP 5 program (OB 1 or FB 0):

1. SEND/function SEND DIRECT
2. SEND/function SEND ALL

DHB SEND:

Block diagram



Use the block No. FB xxx from Table 8-2.

Table 8-4 Format and Meaning of Parameters for SEND

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	E	W	Status word
QTYP	D	KS	Data type of source
DBNR	D	KY	Number of data block if the source is a data block (with QTYP = AS and CPU 946/947: page number)
QANF	D	KF	Offset of first item of data to be read in the data area (byte or word number)
QLAE	D	KF	Number of data units to be read (words or bytes)
PAFE	A	BY	Parameter Assignment Error Bits

SEND call with function SEND DIRECT:

With the first call of the DHB SEND you activate the CP 581 via your STEP 5 program and assign it the address of the data to be transmitted by the CPU. You must ensure that **the result of the previous logic operation (RLO) has a value of "1"** when calling the DHB SEND. CPMASS generates the name of the destination file on the CP from the data type (the block number in addition with DB/DX) and the job number.

Set the following parameters for the DHB SEND for the function SEND DIRECT:

– SSNR:

Enter the number of the page via which you wish to read the data from the CPU: corresponds to the call of the DHB SYNCHRON.

– A-NR:

Enter a value from 1 to 99 as the job number.

– ANZW:

Address of **two** successive words. These words are used by the data handling blocks to store job-related bits (see Section 8.4.2).

Permissible values: FW 0 to 252
 DW 0 to 254

– QTYP:

Enter which type of data you wish to transmit.

Permissible values: DB for data block DB
 DX for data block DX
 FA for flag bytes
 IA for process input image
 QA for process output image
 TA for timer cells
 CA for counter cells
 AS for absolute memory addresses

With **indirect** addressing:
XX (see DHB description)
RW (see Page 8-19)

– QANF:

Enter the number of the first item of data to be transmitted from the S5 area you specified under QTYP.

– QLAE:

Transmit data:

Enter the number of words or bytes which are to be transmitted. If you enter “-1” here (joker length), the DHB transmits the data from QANF up to the end of the area.

Delete file:

Enter “0.”

– PAFE:

Byte address for parameter assignment error bits, for example, FY 111. Bit No. 0 of the status byte is set to “1” in the event of a parameter assignment error (refer to Section 8.4.1 for the meaning of the other status bits).

SEND call with function SEND ALL:

The second call of the DHB SEND using the function SEND ALL triggers data transmission to the CP 581. Set the following parameters for the DHB SEND:

- SSNR:
Enter the number of the page via which you wish to read the data from the CPU: corresponds to the call of the DHB SYNCHRON.
- A-NR:
Always enter "0" as the job number.
- ANZW:
Address of **two** consecutive words as with SEND DIRECT; enter a **different** address here, however.
- QTYP, DBNR, QANF, QLAE:
These parameters are irrelevant with SEND ALL. They should be noted, however.
- PAFE:
As with SEND DIRECT, but enter a **different** address here.

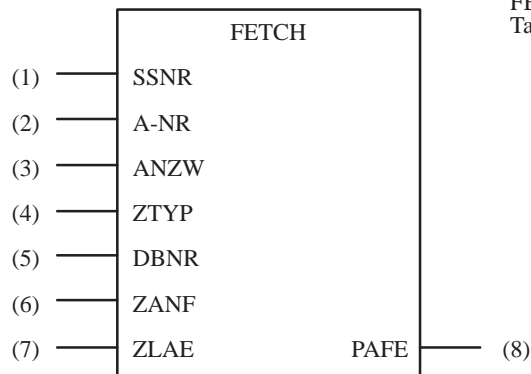
Transmission of Data from CP 581 to CPU

In order to transmit data from the CP 581 to the CPU, you must call the following DHB after synchronization of the CP 581 in the cyclic program:

1. FETCH,
2. RECEIVE/function RECEIVE ALL.

DHB FETCH:

Block diagram



Use the block No. FB xxx from Table 8-2.

Table 8-5 Format and Meaning of Parameters for FETCH

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	E	W	Status word
ZTYP	D	KS	Data type of destination
DBNR	D	KY	Number of data block if destination is data block (with QTYP = AS and CPU 946/947: page number)
ZANF	D	KF	Offset of first item of data to be written in the destination (byte or word number)
ZLAE	D	KF	Number of data units to be written (words or bytes)
PAFE	A	BY	Parameter Assignment Error Bits

Calling DHB FETCH:

By calling the DHB FETCH you activate the CP 581 via your STEP 5 program and transfer the address of the data to be transmitted to the CPU to it. The CPMASS program generates the name of the source file on the CP from the address.

Set the following parameters for the DHB FETCH:

- SSNR:
Enter the number of the page via which you wish to read the data from the CPU: corresponds to the call of the DHB SYNCHRON.
- A-NR:
Enter a value from 1 to 99 as the job number.
- ANZW:
Address of **two** successive words. These words are used by the data handling blocks to store job-related bits (see Section 8.4.2).
Permissible values: FW 0 to 252
 DW 0 to 254

- ZTYP:

Enter which data type you wish to have on the CPU.

Permissible values: DB for data block DB
 DX for data block DX
 FA for flag bytes
 IA for process input image
 QA for process output image
 TA for timer cells
 CA for counter cells
 AS for absolute memory addresses

With **indirect** addressing:
 XX (see DHB description)
 RW (see Page 8-19)

- ZANF:

Enter the number of the first item of data of the above type to be written into the CPU.

- ZLAE:

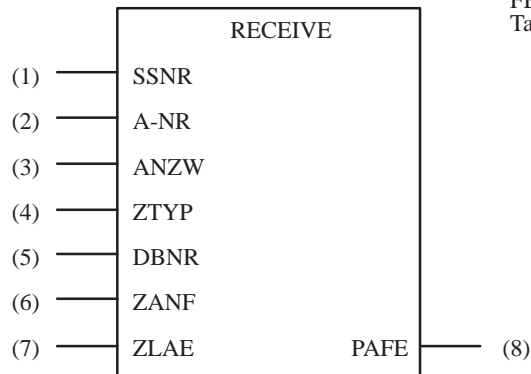
Enter the number of words or bytes which are to be transmitted to the CPU. If you enter “-1” here (joker length), the DHB enters data into the S5 area until the end of the S5F file or the end of the S5 area has been reached on the CPU.

- PAFE:

Byte address for parameter assignment error bits. Bit No. 0 of the status byte is set to “1” in the event of a parameter assignment error (refer to Section 8.4.1 for the meaning of the other status bits).

DHB RECEIVE:

Block diagram



Use the block No. FB xxx from Table 8-2.

Table 8-6 Format and Meaning of Parameters for RECEIVE/
Function RECEIVE ALL

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	E	W	Status word
ZTYP	D	KS	These parameters are irrelevant with the function RECEIVE ALL; they must be specified for format reasons, however.
DBNR	D	KY	
ZANF	D	KF	
ZLAE	D	KF	
PAFE	A	BY	Parameter Assignment Error Bits

RECEIVE call with function RECEIVE ALL:

By calling the DHB RECEIVE with the function RECEIVE ALL you trigger data transmission from the CP 581 to the CPU. Set the following parameters for the DHB RECEIVE:

- SSNR:
Enter the number of the page via which you wish to read the data from the CPU: corresponds to the call of the DHB SYNCHRON.
- A-NR:
Always enter "0" as the job number.
- ANZW:
Address of **two** consecutive words as with FETCH; enter a **different** address here, however.
- QTYP, DBNR, QANF, QLAE:
These parameters are irrelevant with RECEIVE ALL. They should be noted, however.
- PAFE:
As with FETCH, enter a **different** address here, however.

Preselection of Directory on CP 581/Delete all S5F Files in Directory

If necessary for your application of the mass storage functions, you can use a CPU to adjust the directory presetting for four assigned CPUs (directory C:\CPU1 to C:\CPU4) to a different directory. The new directory may also be on a drive other than "C:"

You can also delete **all** files from a directory which have been created by the CPMASS program on the CP 581 for the mass data (S5F files). The same rules apply to the selection of this directory as to the setting.

The function is executed as follows:

First all **files** which have the file name extension **SSF** are deleted from the specified directory. The **directory itself** is also deleted if the directory has **no other files** and **no further subdirectories**. If the directory has subdirectories, the CPMASS program only deletes the files from the selected directory but not the subdirectories and the directory itself.

The **main directory on the hard disk** of the CP 581 and the **current working directory** (on the hard disk or floppy disk drive) **cannot** be deleted. Furthermore you cannot delete any **SSF files** which have the file attribute **read-only** (MS-DOS).



Caution

Proceed carefully when using the delete function so that you do not destroy important information on the CP 581 by mistake.

Programming of the two functions is similar to that described on page 7-14. The two DHB calls SEND/SEND DIRECT and SEND/SEND ALL are used. Proceed as follows:

- Make sure that the path name of the addressed directory is stored in a data block of the CPU such that it commences at a **word limit**. Spaces at the end of the path name are not evaluated.

The following syntax must be observed:

dr:\dir1\dir2\...\dirn\name (possibly spaces) **0**

where

dr:
MS-DOS drive designation,

for example, C or A, **always with “\”**, even if only drive information.

dir1 to dirn:
Name of directories which are present
in the hierarchy prior to the selected directory, **with “\”**.

name:
Name of directory into which data are to be written or
from which data are to be read or deleted,
without “\”:

0:
The end of the string must be terminated by a byte
with the binary value “0.”

- Call the DHB SEND/function SEND DIRECT using the following parameters:
 - SSNR:
 - As described on page 8-11.
 - A-NR:
 - Preselect directory: **201**
 - Delete S5F files: **207**
 - ANZW:
 - As described on page 8-11.
 - QTYP:
 - DB for data block DB or DX for data block DX.
 - DBNR:
 - Number of data block in which the name of the desired directory is stored.
 - QANF:
 - Offset of 1st data word with the path name (the path name must commence at a **word limit**, see Figure 8-6).
 - QLAE:
 - Length of path name as number of words:
 - $QLAE \geq ((1/2 \text{ number of characters}) + 1)$.
 - PAFE:
 - As described on page 8-11.

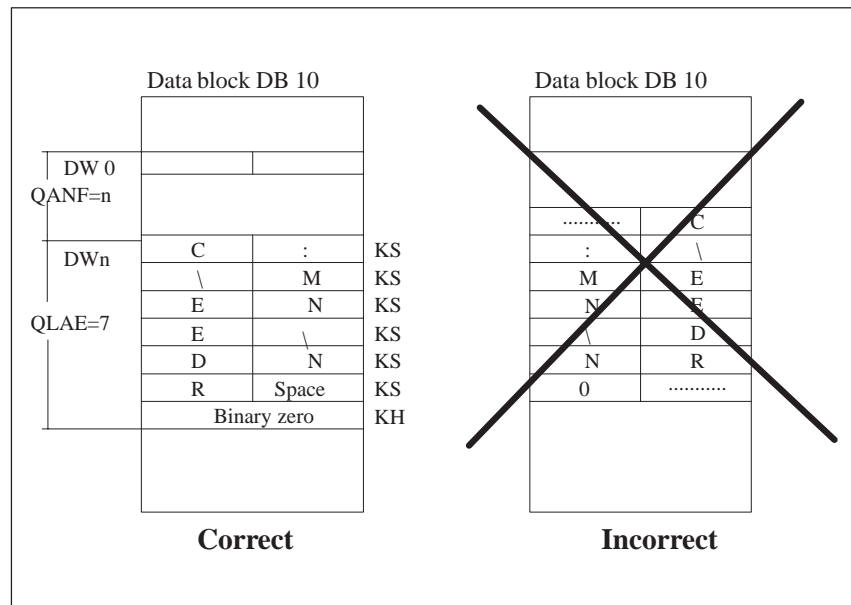


Figure 8-6 Storage of Path Name in a Data Block

**Indirect
Parameterization
'RW'**

The DHB SEND/function DIRECT and DHB FETCH enable you to characterize the data source and destination differently using indirect parameterization (for example, source on CP 581 = data type IA and destination on a CPU = data type FA).

Proceed as follows:

1. Program the source and destination parameters in a data block DB or DX with the following format:

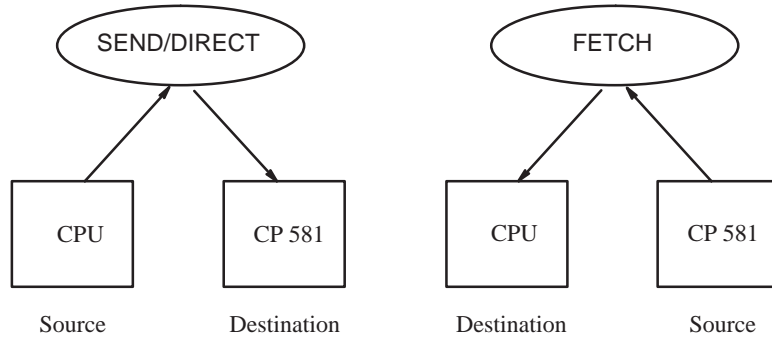
			possibly other data		
QANF*	+0	KS	QTYP:	Type of data source, but not XX, RW and NN	Source para- meters
	+1	KY	DBNR:	Only if QTYP DB and DX	
	+2	KF	QANF:	Initial address of data source	
	+3	KF	QLAE:	Length of data source	
	+4	KS	ZTYP:	Type of data destination, but not XX, RW and NN	Desti- nation para- meters
	+5	KY	DBNR:	Only if QTYP DB and DX	
	+6	KF	ZANF:	Initial address of data destination	
	+7	KF	ZLAE:	Length of data source	
			possibly other data		

2. Parameterize the call of the DHB SEND/DIRECT or FETCH for indirect parameterization with the following special values:

- QTYP/ZTYP:
 - RW For indirect parameterization "RW"
- DBNR:
 - KY = 0,dnr If parameter in DB (dno = DB number)
 - KY = 1,xnr If parameter in DX (xno = DX number)
- QANF/ZANF:
 - QANF* Number of first data word in parameter block of DB/DX (see diagram)

Observe the following peculiarities when using indirect parameterization with “RW:”

- What is the data source, what is the data destination?



- How is the length of the data area to be transmitted determined?

With **SEND/DIRECT**:

by **QLAE**

With **FETCH**:

by **ZLAE**

Whether the length is determined by **words** or **bytes** depends on the organization of the source or destination area.



Caution

Undefined data may occur in the destination when using “FETCH” to transmit data from a byte-oriented area into a word-oriented area or vice versa.

Example: CP 581 file IAxxx.S5F is to be transmitted into DB y. The destination length is word-oriented. More words are transmitted than the length of the file IAxxx.S5F. Random values may be written into the destination DB during the process since the transmission is executed via a transfer buffer.

Examples of DHB Parameterization for Mass Storage Functions

Part 1: Application of direct parameter assignment

On the S5-135U programmable controller, data are to be transmitted via the mass storage function CPMASS from a CPU 928B to the CP 581 or vice-versa. Data exchange is controlled via two input signals and according to the following pattern:

I 20.2	I 20.1	Action
1	0	Send data from CPU to CP 581
0	1	Fetch data from CP 581 to the CPU
1	1	Do not initiate job, but terminate program
<p>Either the SEND job or the FETCH job can be executed separately with the job number 1, but not both simultaneously. The STEP 5 program should ensure this is the case.</p>		

On the CP 581 data are to be transferred via the **010DB001.S5F** file. This file is created when data is sent and serves as data source when fetching data.

On the CPU, the area from data words DW 0 to DW 49 (50 data words) of the data block DB 10 are to be used as the data source and destination.

The STEP 5 program for this task consists of the following parts:

1. calling of FB 111 in the restart blocks,
2. FB 111 with calling of the DHB SYNCHRON,
3. OB 1 for cyclic program processing with job coordination and job initiation,
4. FB 115 for sending data,
5. FB 116 for fetching data.

Error handling takes place in the PB 1 block which is not described in more detail here, since error handling is specific to the application.

The DHBs to be used for the CPU 928B are listed in Table 1-2 and have the following function block numbers:

DHB SYNCHRON	FB 125
DHB SEND:	FB 120
DHB FETCH:	FB 122
DHB SEND-A:	FB 126
DHB REC-A:	FB 127

• **STEP 5 operations in OB 20, OB 21 and OB 22:**

```

SEGMENT 1          0000

0000      :
0001      :          Other potential S5 operations
0002      :
0003      :
0004      :JU   FB 111   Synchronize CP 581
0005 NAME :CPSYNC
0006 REP  :    KF +3   Number of SYNCHRON attempts
0007      :
0008      :
0009      :          Other potential S5 operations
000A      :
000B      :BE
    
```

• **Function block FB 111:**

```

SEGMENT 1          0000
NAME : CPSYNC
DECL : REP          I/Q/D/B/T/C: D   KM/KH/KY/KS/KF/KT/KC/KG: KF

0008      :L      KB 0
0009      :T      FY 10      Initialize loop counter
000A      :
000B LOOP :          Return marker
000C      :JU     FB 125      Synchronize CP 581
000D NAME :SYNCHRON
000E SSNR :      KY 0,16     SSNR = Page No.= 16
000F BLGR :      KY 0,0      Frame size = 0 => 256 bytes
0010 PAFE :      FY 11      Parameter error in FY 11
0011      :
0012      :AN     F 11.0
0013      :JC     =END       The block is completed without error only if
0014      :          no parameter error has occurred.
0015      :L      FY 10      Increase loop counter by one
0016      :I          1      on every SYNCHRON run
0017      :T      FY 10      (abortcriteria)
0018      :
0019      :L      FY 10      Check whether the maximum
001A      :LW     =REP       permissible number of
001B      :<=F          runs has been
001C      :JC     =LOOP     reached.
001D      :
001E      :STS          PLC STOP following a PAFE, or if the
001F      :          set number of SYNCHRON
0020      :          runs has been exceeded
0021      :
0022 END  :          SYNCHRON completed successfully
0023      : BE

```

• **STEP 5 operations in OB 1:**

```

SEGMENT 1          0000 Cyclic program processing
0000      :
0001      :
0002      :          Other potential S5 operations
0003      :
0004      :
0005      :A      I 20.0      Coordination of SEND and FETCH:
0006      :A      I 20.1      only one of the two jobs may be
0007      :BEC          started at any one time!
0008      :
0009      :JU     FB 115      Send data
000A NAME :CPSEND
000B ANST :      I 20.0      Trigger via input I 20.0
000C      :
000D      :JU     FB 116      Fetch data
000E NAME :CPFETCH
000F ANST :      I 20.1      Trigger via input I 20.1
0010      :
0011      :JU     FB 126      SEND ALL
0012 NAME :SEND-A
0013 SSNR :      KY 0.16

```

(continued on next page)

(OB 1 cont.:)

```

0014 A-NR :    KY 0,0
0015 ANZW :    FW 30
0016 PAFE :    FY 34
0017      :
0018      :JU  FB 127  RECEIVE ALL
0019 NAME :REC-A
001A SSNR :    KY 0,16
001B A-NR :    KY 0,0
001C ANZW :    FW 35
001D PAFE :    FY 39
001E      :
001F      :O   F 34.0  PAFE in SEND ALL
0020      :O   F 39.0  PAFE in RECEIVE ALL
0021      :JC  PB 1    Errors occurring are dealt with in PB 1.
0022      :                    (This will not be explained in detail here.)
0023      :BE

```

• **Function block FB 115**

```

SEGMENT 1      0000 Send data from the CPU to the CP
NAME :CPSEND
DECL :TRIG     I/Q/D/B/T/C: I BI/BY/W/D: BI

0008      :A   =ANST  Form positive edge of the trigger
0009      :AN  F 100.0  event
000A      :S   F 100.0
000B      :=   F 100.1  Edge flag
000C      :AN  =ANST
000D      :R   F 100.0
000E      :
000F      :A   F 100.1
0010      :JC  FB 120  Start actual sending
0011 NAME :SEND
0012 SSNR :    KY 0,16  SSNR = Page No. = 16
0013 A-NR :    KY 0,1  A-NR = Job number = 1
0014 ANZW :    FW 20
0015 QTYP :    KS DB   Data to be sent is in DB 10
0016 DBNR :    KY 0,10  from DW 0 to DW 49 (= 50 words).
0017 QANF :    KF +0    Access to file: 010DB001.S5F
0018 QLAE :    KF +50
0019 PAFE :    FY 24
001A      :
001B      :O   F 21.3  Job completed with error
001C      :O   F 24.0  Parametererror
001D      :JC  PB 1    Errors occurring are dealt with in PB 1.
001E      :                    (This will not be explained in detail here.)
001F      :BE

```

• **Function block FB 116**

```

SEGMENT 1      0000 Fetch data from CPU to CP
NAME :CPFETCH
DECL :TRIG     I/Q/D/B/T/C: I BI/BY/W/D: BI

0008      :A   =ANST  Form positive edge of the trigger
0009      :AN  F 100.2  event
000A      :S   F 100.2

```

(continued on next page)

(OB 1 cont.):

```

000B      := F 100.3   Edge flag
000C      :AN =ANST
000D      :R   F 100.2
000E      :
000F      :A   F 100.3
0010      :JC  FB 122   Start actual fetching
0011  NAME :FETCH
0012  SSNR :    KY 0,16  SSNR = Page No. = 16
0013  A-NR :    KY 0,1   A-NR = Job number = 1
0014  ANZW :    FW 25
0015  QTYP :    KS DB    Data to be fetched are in DB 10
0016  DBNR :    KY 0,10  from DW 0 to DW 49 (= 50 words).
0017  QANF :    KF +0    File created: 010DB001.S5F
0018  QLAE :    KF +50
0019  PAFE :    FY 29
001A      :
001B      :O   F 26.3   Job completed with error
001C      :O   F 29.0   Parametererror
001D      :JC  PB 1     Errors occurring are dealt with in PB 1.
001E      :
001F      :BE          (This will not be explained in detail here.)

```

Part 2: Application of indirect parameterization “RW”

• **Example 1: Data transfer from a CPU to the CP 581**

Job number: 1

Data source: Input bytes IB 0 to IB 127 on the CPU

Data destination: File 020DB001, DW 0 to DW 63, on the CP 581

Interface No.: 0 (= page No.)

The DHB SEND is parameterized via “RW:”

```

NAME :SEND
SSNR :    KY 0,0        Interface No.
A-NR :    KY 0,1        Job No.
ANZW :    FW 0
QTYP :    KS RW        The source and destination parameters
DBNR :    KY 0,10      are stored in data block DB 10
QANF :    KF +10       starting at data word DW 10.
QLAE :    KF +8        Value without significance
PAFE :    FY 4

```

The source and destination parameters DB 10 and DW 10 to DW 17 in the data block (where DW10 to 13 = source parameters and DW14 to 17 = destination parameters) have the following values:

```

DW10: KS IA      QTYP:   Input byte
DW11: KY 0,0     DBNR:
DW12: KF +0      QANF:   IB 0
DW13: KF +128    QLAE:   128 bytes
DW14: KS DB      ZTYP:   Data block (file), type DB
DW15: KY 0,20    DBNR:   File for DB 20
DW16: KF +0      ZANF:   DW 0
DW17: KF +64     ZLAE:   No significance 1)

```

1 The transmission length is determined by QLAE.

- **Example 2: Data transfer from CP 581 to a CPU**

Job number:

2

Data source:

File 020DB002, DW 0 to DW 63, on the CP 581

Data destination:

Data block DX 30, DW 100 to DW 163,

on the CPU

Interface No.:

0 (= page No.)

The DHB FETCH is parameterized via "RW:"

NAME :FETCH

SSNR :	KY 0,0	InterfaceNo.
A-NR :	KY 0,2	Job No.
ANZW :	FW 0	
QTYP :	KS RW	The source and destination parameters
DBNR :	KY 0,10	are stored in data block DB 10
QANF :	KF +20	starting at data word DW 20.
QLAE :	KF +8	Value without significance
PAFE :	FY 14	

The source and destination parameters DB 10 and DW 20 to DW 27 in the data block (where DW20 to 23 = source parameters and DW24 to 27 = destination parameters) have the following values:

DW20: KS DB	QTYP:	Data block (file), type DB
DW21: KY 0,20	DBNR:	Data block No. 20
DW22: KF +0	QANF:	DW 0
DW23: KF +64	QLAE:	No significance ¹⁾
DW24: KS DX	ZTYP:	Data block, type DX
DW25: KY 0,30	DBNR:	File for DX 30
DW26: KF +100	ZANF:	DW 100
DW27: KF +64	ZLAE:	64 words

¹ The transmission length is determined by ZLAE

8.3.4 Activation and Testing of the Mass Storage Functions

When installing the CP 581 system software you can optionally define the entries in the AUTOEXEC.BAT file such that the CPMASS program is started for the mass storage functions with each system restart/warm restart of the CP 581. The program then expects to be triggered by a CPU in the same programmable controller as the CP 581 in order to then transfer data with it.

The CPMASS program outputs the following text on the monitor when started:

```
CPMASS Vx.x  
Copyright (c) Siemens 1992
```

The program remains installed following a system restart/warm restart of the CP 581. If you do not wish to use it, and therefore wish to delete it from the memory, you must remove the **CPMASS** command from the AUTOEXEC.BAT file and carry out a cold restart on the CP 581.

Note

If you wish to use the “Mass storage functions” and the “Process data acquisition” **simultaneously**, it is essential to start the CPMASS program **before** the CPRECORD program since you cannot otherwise uninstall CPRECORD.

Name, Length and Contents of a File for CPU Data on the CP

Before commencing with the test of your STEP 5 program for application of the mass storage functions, you should know how the CPMASS program generates the CP 581 files (S5F files) for these data depending on the type of CPU data. You can then later check for all data whether these have arrived from an S5 data area on the CP 581.

Table 8-7 Names and Lengths of CP 581 Files for S5 Data

S5 data area	S5 data type	Name of CP 581 file ¹⁾	Max. file length (joker length)
Data blocks DB	DB	nnnDBaaa.S5F	8182 bytes
Extended data blocks DX	DX	nnnDXaaa.S5F	8182 bytes
Flag area F	FA	FAaaa.S5F	256 bytes
Process output image	QA	QAaaa.S5F	182 bytes
Process input image	IA	IAaaa.S5F	182 bytes
Counter cells	CA	CAaaa.S5F	512 bytes
Timer cells	TA	TAaaa.S5F	512 bytes
Absolute memory address	AS ²⁾	ASaaa.S5F	8192 bytes ²⁾

1 nnn = DB-/DX number, aaa = job number

2 Please read the following text “Special features of the S5 area AS”

Writing the CP 581 files (S5F files)

– File length:

The length of a CP 581 file results from the total of the parameter values QANF and QLAE which you have entered in the call of the DHB SEND/SEND DIRECT when transmitting the data (see page 8-11).

If an S5F file is written more than once, the file length is the largest value of this total (except with data type AS). If you enter the joker length in a send job of the CPU, the S5F file is assigned the maximum length, except with AS (see table).

– File contents:

For each S5 area except AS, the S5F file represents a memory image of the S5 area commencing with QANF (offset) “0.” If you write a file with a QANF value >“0,” the preceding area of the S5F file is assigned invalid or undefined data.

Following each transmit job of the CPU into an S5F file which is present or not yet present, the **valid data** of this file are always in the memory area “QANF to (QANF+QLAE-1).”

Special features of the S5 area AS

When transmitting S5 data of type AS, observe the following special features compared to the other S5 areas:

- A buffer area of only 8192 bytes is available for the mass storage functions when transmitting S5 data. A maximum of 8192 bytes is therefore transmitted to the CP 581 per transmitter job. The AS area on a CPU may be up to 2 Mbytes, however, depending on the type of CPU. With the CPU 946/947, the parameter **DBNR** must contain the page number of the area. If you wish to transmit an area larger than 8192 bytes to the CP or read such an area from the CPU, you must specify several transmitter and read orders with **different** job numbers to transmit this area.
- The S5 data are written into the S5F file starting at the beginning of the file and independent of the value of the parameter QANF (offset). There are therefore no invalid or undefined data in a file for the data type AS. When writing back to the CPU, the parameter ZANF specifies the 1st absolute address starting at which the data are to be stored on the CPU. Therefore use the same value (= 1st absolute address) for QANF and ZANF with AS if you wish to address the same AS area when writing and reading.

Caution

Only use jobs to transmit AS areas from the CP 581 to the CPU (calling of DHB FETCH) if you are fully acquainted with handling of the CPU in the system. Unspecified CPU data (especially system data) may be overwritten if the function is used incorrectly, and may trigger a system crash of the CPU.

Testing

Proceed as follows to establish whether your required data transfer between the CPU and CP 581 is being executed correctly:

1. Prepare a data block for the test with static data (not equal to zero!) on the programming device (for example, DB 20) and transfer the block using the programming device to the CPU whose STEP 5 program you wish to test for data transfer.
2. Write the STEP 5 program such that it initially only sends data once from the CPU to the CP 581 (for example, following scanning of an input in the restart/warm restart).
3. Determine whether calling of the DHBs SYNCHRON, FETCH and RECEIVE ALL is executed on the CPU without error bits being indicated (see Section 8.4).
4. Check whether the associated file has been created (for DB 20 = 020DB001.S5F) in the preset directory following execution of the STEP 5 program on the CP 581 (CPU1 for the CPU with interface No. 1, CPU2 for interface No. 2 etc.). If the file does not exist, check the results of steps 1 to 3 to determine where the cause of the fault is to be found.
5. Copy the file with the CPU data into another file in the same directory in order to fetch this file back to the CPU (for example, file 020DB001.S5F from data block DB 20 into the file for data block DB 21 with the MS-DOS command "COPY 020DB001.S5F 021DB001.S5F").
6. Write the STEP 5 program such that it reads data once from the copied CP 581 file into the corresponding data block (for example, the file 021DB001.S5F into data block DB 21).
7. Compare the data of the transmitted data block with that of the received data block. If they agree, you can assume that data transfer with the mass storage functions is executed correctly. You can then cancel any modifications made to your STEP 5 program (for example, access other S5 data areas) or program the DHB calls for the mass storage function "Set/delete directory" if required.

8.4 Error Bits

You can evaluate the bits of the data handling blocks in order to test your STEP 5 program for the mass storage functions. Two types of bits are available:

- Parameter assignment error bits
- and
- Job status bits

8.4.1 Parameter Assignment Error Bits

All data handling blocks check the transferred parameters for syntax and logical errors when called. In addition, they check whether the addressed interface is available for the triggered function.

Whether parameter assignment errors have occurred and which ones can be seen from the status byte whose address you have specified by the parameter PAFE when calling the DHB:



– Error byte PAFE

* = Common bit:

0:

no error

1:

parameter assignment error, more details in bits 4 to 7.

Table 8-8 lists all bits which can be stored by the data handling blocks in the PAFE.

Table 8-8 Parameter Assignment Error Bits

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format
21H	DB or DX data block not present or illegal (e. g. DB 0 or DX 0 with QTYP = DB or DX)
31H	Area too small or total of initial address (QANF/ZANF) and length (QLAE/ZLAE) too large (with all QTYP/ZTYP)
41H	Area does not exist or is illegal (with QTYP/ZTYP = AS, QA, IA, PY)
51H	Status word (address) faulty
61H	Dependent on CPU

Table 8-8 Parameter Assignment Error Bits

PAFE value	Cause of error
71H	Interface does not exist
81H	Interface not ready
91H	Interface overloaded
A1H	Dependent on CPU
B1H	Job number illegal or frame size (SYNCHRON) illegal
C1H	Interface does not react, or interface does not react at correct time or interface rejects job
D1H	Dependent on CPU
E1H	Dependent on CPU
F1H	Dependent on CPU

8.4.2 Job Status Bits

Bits are set in two declared status words with the mass storage functions by means of the DHBs SEND, RECEIVE and FETCH and by the CPMASS program via the CPDHB driver. You have defined the initial address of the status words in the parameter ANZW in the DHB calls.

Some of the status bits of the DHB SEND/SEND DIRECT and the DHB FETCH are important for the test. These status bits are explained later. You can obtain more information on the DHB status bits from the DHB descriptions.

The status words of the DHB SEND/SEND DIRECT and FETCH have the following format:

Bit No.	15	12	11	8	7	4	3	0
ANZW			CPMASS CPMASS	Status bits from data management		Status bits from job management		
ANZW+1	Number of transmitted bytes							

Status bits from job management (bit nos. 0 to 3):

You can obtain the respective status of a job from these status bits:

Table 8-9 Status Bits from Job Management

Bit No.	Meaning
0	No significance
1	Bit = 1: Job for data transfer taking place
2	Bit = 1: Job completed without errors
3	Bit = 1: Job completed with errors

Note

You cannot send jobs to the CP 581 for the respective job number if bits 1 and 2 are set simultaneously in the status word. Note this response when restarting your S5 program.

Status bits from data management (bit nos. 4 to 7):

These status bits inform you on the current state of the triggered data transfer:

Table 8-10 Status Bits from Data Management

Bit No.	Meaning
4	Bit = 1: Data exchange taking place
5	Bit = 1: Data transfer completed
6	Bit = 1: Data accepted
7	Bit = 1: Data transfer/acceptance disabled: you can set and delete this bit as required.

Special status bits of the CPMASS program (bit nos. 8 to 11):

If the CPMASS program determines when processing a job from a CPU that it cannot be handled without faults, it stores this in the first status word of the two words of the CPU error bits addressed with ANZW.

The following table shows you the status bits which may occur during execution of the mass storage functions.

Table 8-11 Error Bits from the CPMASS Program

Value of bit No. 8 to bit No. 11	Cause of error
0H	No error
1H to 5H	DHB error (also referred to as PLC or CPU error), error numbers PAFE 1 to PAFE 5
6H	CPDHB error
7H	S5 segment disabled (RS, PY)
8H	QLAE/ZLAE too large
9H	CP 581 file write-protected (send job from CPU)
AH	Insufficient space on the set drive of the CP 581 (send job from CPU)
BH	File/directory not found on CP 581 (read job, job "Set directory/delete S5F files" from CPU)
CH	Path name with incorrect syntax (job "Set directory/delete S5F files" from CPU)
DH	Directory was not completely deleted since it contains subdirectories or non-deletable files or external files (not *.S5F) or is the current directory (job "Delete S5F files" from CPU)
EH	Timeout (after 30 seconds)
FH	Parameter assignment error

8.5 Setting of Timeout

CPMASS monitors the timing of data exchange and sends an error status bit to the data handling blocks if the set timeout is exceeded before a job has been completely processed.

The timeout is preset to 30 s. You can change this value using the command:

CPMASS /T<time>

You can enter values from 1 to 3640 for “time;” this specifies the timeout in seconds.

You can also use “-” in the command instead of “/”, and also lower-case letters instead of upper-case letters.

9

Command Interpreter

This chapter describes how you can execute MS-DOS commands on the CP 581 using a CPU inserted together with the CP 581 in your programmable controller.

The chapter describes all measures and operations required on the CP 581 and the CPU:

You must adapt your STEP 5 programs for the command interpreter function on the CPU from which you wish to execute the MS-DOS commands on the CP 581 (see Section 9.3.4). For this you should have experience in programming programmable controllers.

This chapter also provides you with information on the response of the command interpreter when errors occur.

The programming example in Section 9.3.5 only serves to explain the DHB parameters for CPSHELL. You can find complete programs which can be executed in Chapter 12.

Chapter Overview

Section	Description	Page
9.1	Application	9-2
9.2	Principle Sequences Between CPU and CP	9-2
9.3	Command Interpreter Operations	9-4
9.3.1	Related Procedures	9-4
9.3.2	Measures on the CP	9-5
9.3.3	Defining the Command Output	9-5
9.3.4	Programming the CPU	9-6
9.3.5	Example of DHB Parameter Assignment for the Command Interpreter	9-12
9.3.6	Activation/Termination and Testing of the Command Interpreter	9-15
9.4	Error Bits	9-17
9.4.1	Parameter Assignment Error Bits	9-17
9.4.2	Job Status Bits	9-18
9.5	Special Features During Command Interpretation	9-20

9.1 Application

You can use the function “Command interpreter” with a CPU to execute on the CP 581 any MS-DOS commands stored in a data block DB or DX. For example, if you use your CP 581 without a keyboard because you only use the “Mass storage functions,” the command interpreter can be used if necessary to copy files from the CPU onto the CP or to trigger other MS-DOS functions as required.

Refer to the information in Section 9.3.3 when selecting the commands.

Once the command interpreter has been started, you cannot enter commands via a keyboard apart from the command to terminate the command interpreter.

The command interpreter can be addressed by up to four CPUs which are inserted in the same programmable controller as the CP 581.

9.2 Principle Sequences Between CPU and CP

Figure 9-1 shows you the basic sequence of command interpretation on the CPU and CP 581.

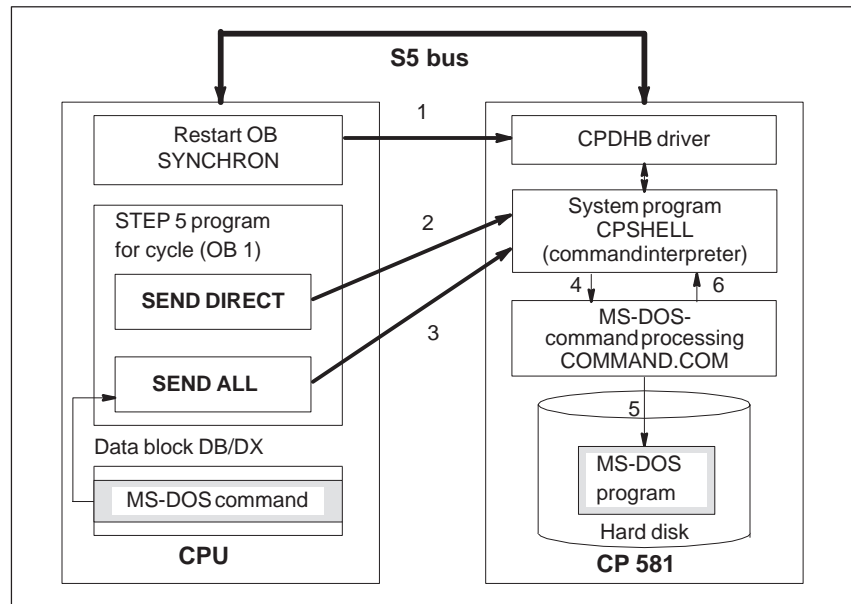


Figure 9-1 Basic Sequence of a Command Interpretation

- 1 Synchronization of CPU during restart and warm restart with CP via DHB SYNCHRON.
- 2 Triggering of command interpreter.
- 3 Read command line from data block and transmit to CPSHELL command interpreter via CP/DHB driver.
- 4 CPSHELL transfers the received command line to the MS-DOS command interpreter COMMAND.COM. This handles further processing under MS-DOS as with a command input via the keyboard.
- 5 In the case of commands which cannot be executed internally by COMMAND.COM, the corresponding program file (xxx.EXE, xxx.COM or xxx.BAT) is started via the currently set path of the hard disk.
- 6 MS-DOS hands over control to CPSHELL again when it has finished the command processing or when the started program has been terminated. The STEP 5 program then receives a status bit from CPSHELL indicating that the communication for the command has been terminated.

9.3 Command Interpreter Operations

9.3.1 Related Procedures

You can use the CP 581 to execute commands on itself which you have stored in an S5 data block DB or DX. You must carry out the following measures in the indicated sequence:

Table 9-1 Measures to Activate the Command Interpreter

Measure	on	See section
Make sure that the CPU with which you wish to execute the commands and the CP 581 are plugged into your programmable controller. The base interface No. for data transfer must be set on the CP 581.	PLC CP 581	9.3.2
Define which command is to be executed, and when: define the associated condition (input, flag) for when the cyclic STEP 5 program is to call which command and assign this condition with the associated initial address of the start of the command in the data block.	--	9.3.3
Program a data block DB or DX with the commands you wish to execute on the CP 581.	PG CPU	9.3.4
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG CPU	9.3.4
Create or modify the STEP 5 program on the CPU for the cycle (OB 1 or FB 0) according to the defined conditions for command execution and the syntax of the command data block.	PG CPU	9.3.4
Make sure that all DHBs are called correctly during cold or warm restart or cycle by testing via CPU.	PG CPU	9.3.6 and 9.4
Make sure that the CPDHB driver is loaded in the memory of the CP 581. The command interpreter can be automatically started as an option with a cold or warm restart (entry in AUTOEXEC.BAT during software installation). If the entry in the AUTOEXEC.BAT is missing you can start the driver with the CPDHB command.	CP 581	--
Synchronize the CP 581 with the CPU by triggering a restart of the CPU.	CPU	--
Check the correct sequence when transferring a command by testing on the CPU and CP.	PG CPU CP 581	9.3.4 and 9.4

9.3.2 Measures on the CP

Only one preparatory measure is necessary on the CP to use the command interpreter:

Setting the base interface number:

So-called pages (page frames) are used to address the memory areas when transferring data on the S5 bus. These pages have a fixed assignment to the modules involved with data transfer. The CP 581 can transfer data with up to four CPUs via four successive pages. These pages must be numbered consecutively.

The number of the first page is the base interface number. Define this number for the first inserted CPU with which you wish to exchange data, and set it on the CP 581 as described in Section 6.2.3.

The pages for data transfer with three further CPUs then have consecutive numbers following the page with the base interface number.

9.3.3 Defining the Command Output

To prepare the programming operations on the CPU, you must first answer the following questions:

- **Which commands are to be output?**

It is certainly not relevant to use the complete spectrum of MS-DOS commands on a CPU.

Therefore first define the commands you wish to execute with the CPU. You are allowed to load your own MS-DOS programs onto the CP 581 for specific applications and to start these from the CPU using their name.

- **When is a specific command to be output?**

The command string must be transmitted to the CP 581 in the cyclic STEP 5 program using data handling blocks. Since it is not necessary to output a command in each cycle, however, you must define start signals for this which can be scanned by the cyclic program. These can be inputs or flags, for example.

- **Which directory is to be addressed using a command?**

If a command (for example, TYPE) is output using MS-DOS, it usually refers to a file of the currently set path. Since the CPU does not know what the current path is, you must define the path in the command.

Examples:

```
TYPE C:\MYDIR\MYFILE
*** Output MYFILE file
```

```
C:\MYDIR\MYPROG
*** Call MYPROG program
```

- **Are commands output by several CPUs?**

If you wish to use the command interpreter from several CPUs, you must consider whether coordination measures are also necessary (refer to the description of your programmable controller on how to implement these).



Caution

The **FORMAT** command should **not be used at all** with the command interpreter, the **DELETE** command only **very carefully**.

You must **not** start the applications “Process data acquisition” (**CPRECORD** program) and “Mass storage functions” (**CPMASS** program) using the command interpreter.

9.3.4 Programming the CPU

Storing Commands in Data Block

Once you have defined one or more commands, you must program these as strings in a data block DB or DX. The command string must be terminated by a binary zero. Spaces at the end of a string prior to the binary zero are not evaluated.

When activating a command (see Section 9.3.4, Page 9-6) enter its initial address in the data block. This must be a **word** address (see Figure 9-2).

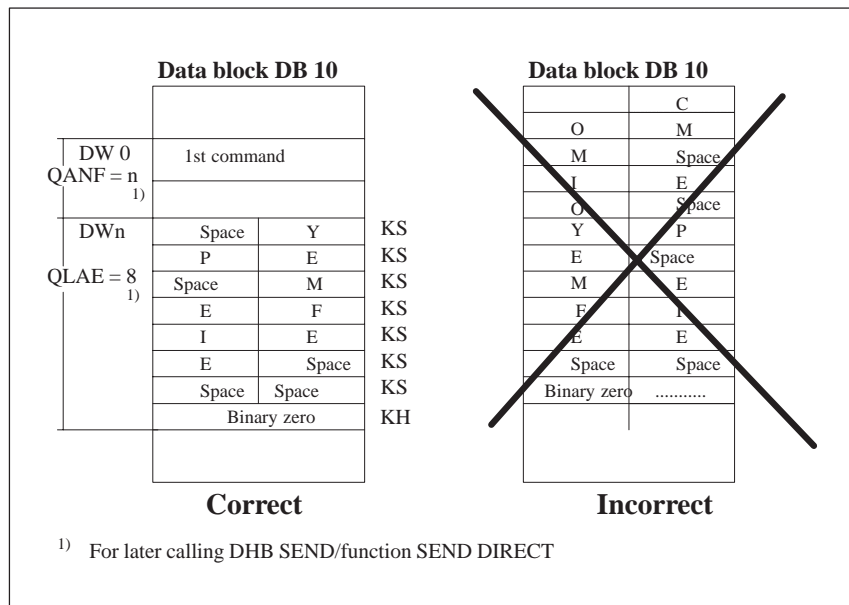


Figure 9-2 Storage of a Command in the Data Block

STEP 5 Operations for the Command Interpreter

Programming of the CPU for application of the command interpreter comprises synchronization of the CP 581 during the restart and the cyclic calling of a special function block for transmitting the command string. For both functions you require the so-called "Data handling blocks" (DHB); the DHB SYNCHRON for synchronization and the DHB SEND for command transfer.

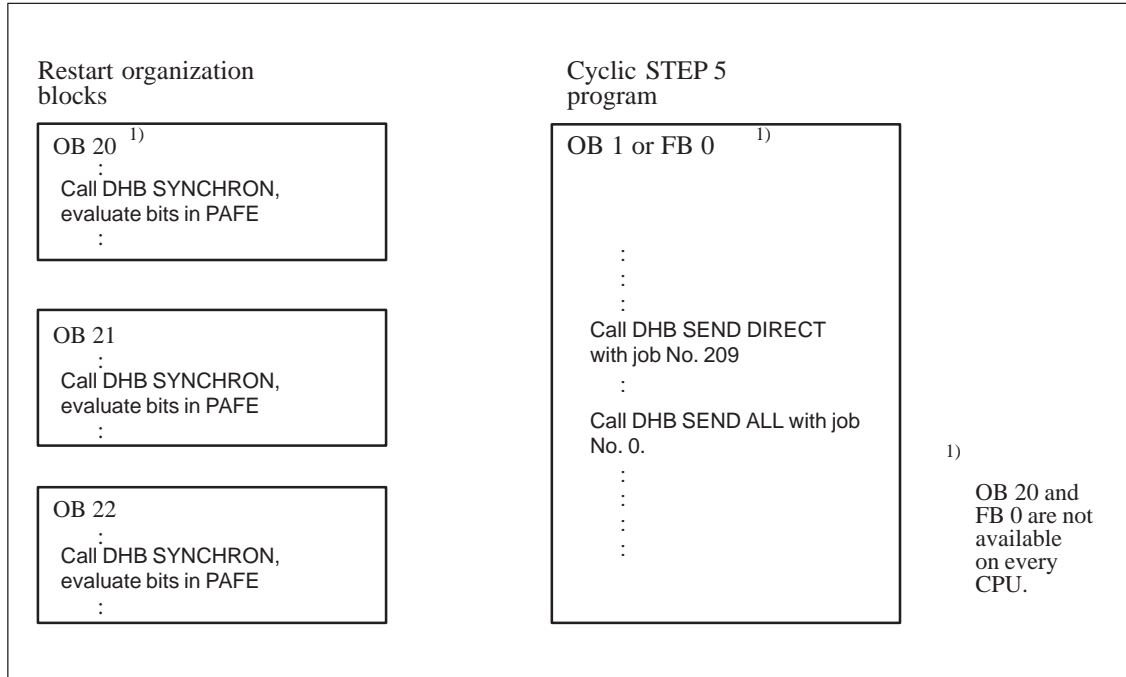


Figure 9-3 Principle of DHB Calls in the STEP 5 Program of the CPU when Using the Command Interpreter

Figure 9-3 shows you the positions at which you must call the two data handling blocks in your STEP 5 program.

Calling and Parameterizing the Data Handling Blocks

The functions of the data handling blocks which you need to execute commands on the CP 581 are described in this section as far as is necessary for programming.

If you use an S5-135U or S5-155U, you can also call the DHB SEND-A which is available there instead of the DHB SEND in your STEP 5 program. You can obtain information on these DHBs from the corresponding descriptions of the DHBs for the S5-135U and S5-155U programmable controllers (see /8/ and /9/ Reference Literature).

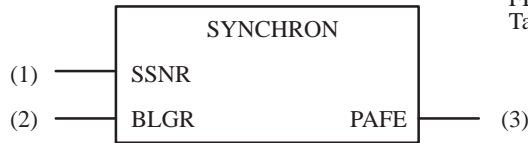
The data handling blocks have different block numbers on the various programmable controllers. The following table contains the numbers of the DHBs SYNCHRON and SEND which you require for the various programmable controllers.

Table 9-2 DHB Numbers on the Various Programmable Controllers

	S5-115U	S5-135U	S5-155U
DHB SYNCHRON	FB 249	FB 125	FB 125
DHB SEND	FB 244	FB 120	FB 120

DHB SYNCHRON

Block diagram



Use the block No.
FB xxx from
Table 9-2.

Table 9-3 Format and Meaning of Parameters for SYNCHRON

Parameter	Kind	Type	Meaning
SSNR	E	KY	Interface number
BLGR	E	KY	Frame size
PAFE	Space	BY	Parameter assignment error bits

Set the following parameters for the DHB SYNCHRON:

– SSNR:

Enter the number of the page via which you wish to read the data from the CPU: “(CPU No. - 1) + base interface No.” (see Section 9.3.2).

Permissible values: 0 to 255 ¹⁾

– BLGR:

Use this parameter to define the maximum number of bytes to be transmitted to the CP 581 in a CPU cycle.

Permissible values: 0 = standard values (see table)
 1 = 16 bytes
 2 = 32 bytes
 3 = 64 bytes
 4 = 128 bytes
 5 = 256 bytes
 6 = 512 bytes

1) The base interface No. must be set in steps of four (0, 4, 8, 12 etc.)

Standard values for BLGR = 0	
S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

– PAFE:

Byte address for parameter assignment error bits. Bit No. 0 of the status byte is set to “1” in the event of a parameter assignment error (refer to Section 9.4.1 for the meaning of the other status bits).

Note

The CP 581 may require more time to run up than the CPU. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Make sure, however, that the program loop is **always** terminated by an abort criterion (timer or loop counter).

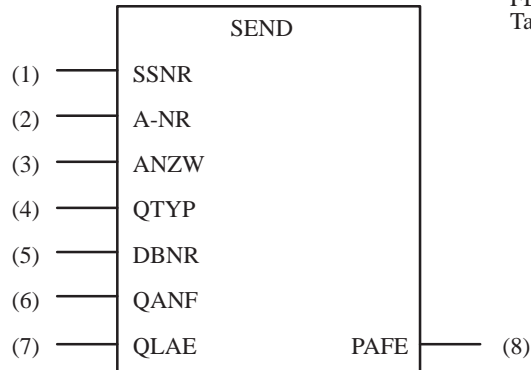
Activate command:

In order to send a command from the prepared data block to the CP 581 and to execute it, you must program the call DHB SEND twice in the cyclic STEP 5 program:

1. SEND/Function SEND DIRECT
2. SEND/Function SEND ALL

DHB SEND

Block diagram



Use the block No. FB xxx from Table 9-2.

Table 9-4 Format and Meaning of Parameters for SEND

Parameter	Kind	Type	Meaning
SSNR	E	KY	Interface number
A-NR	E	KY	Job number
ANZW	E	W	Status word
QTYP	E	KS	Type of data block
DBNR	E	KY	Number of data block
QANF	E	KF	Initial address of command in data block (word number, for example DW 5)
QLAE	E	KF	Length of command as number of words
PAFE	Space	BY	Parameter assignment error bits

SEND call with function SEND DIRECT:

With the first call of the DHB SEND you activate the CP 581 via your STEP 5 program and assign it the address of the command to be sent by the CPU. You must ensure that **result of the previous logic operation (RLO) has a value of “1” when calling the DHB SEND**. To make sure that a previously output command has been completed, you must check in the status word whether another job is still active (see Section 9.4.2). In this case you must not send a new command.

Set the following parameters for the DHB SEND for the function SEND DIRECT:

- SSNR:
Enter the number of the page via which you wish to read the command from the CPU: corresponds to the call of the DHB SYNCHRON.
- A-NR:
Enter the value **209** as the job number.
- ANZW:
Address of **two** successive words. These words are used by the data handling blocks to store job-related status bits (see Section 9.4.2).
Permissible values: FW 0 to 252
 DW 0 to 254
- QTYP:
Enter the type of data block in which the command is stored.
Permissible values: DB for data block DB
 DX for data block DX
- QANF:
Enter the initial address (= No. of data word at which the command commences - see Figure 9-2) of the command.

- QLAE:
Enter the number of words of which the command string consists (= 1/2 number of characters including spaces and terminating word with binary zero - see Figure 9-2).
- PAFE:
Byte address for parameter assignment error bits. Bit No. 0 of the status byte is set to "1" in the event of a parameter assignment error (refer to Section 9.4.1 for the meaning of the other status bits).

SEND call with function SEND ALL:

The second call of the DHB SEND using the function SEND ALL triggers transmission of the command to the CP 581. Set the following parameters for the DHB SEND:

- SSNR:
Enter the number of the page via which you wish to read the data from the CPU: corresponds to the call of the DHB SYNCHRON.
- A-NR:
Always enter "0" as the job number.
- ANZW:
Address of **two** consecutive words as with SEND DIRECT; enter a **different** address here, however.
- QTYP, DBNR, QANF, QLAE:
These parameters are irrelevant with SEND ALL. They should be noted, however.
- PAFE:
As with SEND DIRECT, but enter a **different** address here.

9.3.5 Example of DHB Parameter Assignment for the Command Interpreter

The command "TYPE MYFILE" is to be executed on the CP 581 on the S5-135U programmable controller by means of a CPU depending on a set input signal I 20.2:

- The command is only to be executed at the rising edge of the input signal I 20.2.
- The CPU is the only one in the programmable controller and has the CPU number "1."
- The command is programmed in DB 10 starting at data word 8. In order to program a "Zero termination" easily, the command string is extended by space(s) and the word following the string defined by the instruction "KH = 0000" with two **binary zero** bytes. The complete command to be transmitted thus has a length of 8 words (see Figure 9-2).

The STEP 5 program for this task consists of the following parts:

1. Calling FB 111 in the restart blocks
2. FB 111 with the call of the DHB SYNCHRON
3. OB 1 for the cyclic program processing with the trigger events
4. FB 117 to send MS-DOS commands
5. Data block DB 10 with the command

Error handling is dealt with in PB 1 which is not explained in more detail since error handling is specific to the application.

The data handling blocks which are to be used for the CPU are listed in Table 9-2 and have the following function block numbers:

DHB SYNCHRON	FB 125
DHB SEND:	FB 120
DHB SEND-A:	FB 126

• **STEP 5 operations in OB 20, OB 21 and OB 22:**

SEGMENT 1	0000	
0000	:	
0001	:	Other potential S5 operations
0002	:	
0003	:	
0004	:JU	FB 111 Synchronize CP 581
0005	NAME:CPSYNC	
0006	REP :	KF +3 Number of SYNCHRON attempts
0007	:	
0008	:	
0009	:	Other potential S5 operations
000A	:	
000B	:BE	

• **Function block FB 111:**

```

SEGMENT 1          0000
NAME :CPSYNC
DECL :REP          I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KF

0008      :L      KB 0
0009      :T      FY 10      Initialize loop counter
000A      :
000B LOOP :          Return marker
000C      :JU FB 125      Synchronize CP 581
000D NAME :SYNCHRON
000E SSNR :      KY 0,16      SSNR = Page No. = 16
000F BLGR :      KY 0,0      Frame size = 0 => 256 bytes
0010 PAFE :      FY 11      Parameter error in FY 11
0011      :
0012      :AN      F 11.0
0013      :JC      =END      The block is completed without error only
0014      :          if no parameter error has occurred.

0015      :L      FY 10      Increase loop counter by one
0016      :I      1          on every SYNCHRON run
0017      :T      FY 10      (abortcriteria)
0018      :
0019      :L      FY 10      Check whether the maximum
001A      :LW      =REP      permissible number of runs
001B      :<=F          has been reached.
001C      :JC      =LOOP
001D      :
001E      :STS          PLC STOP following a PAFE, or
001F      :          if the set number of SYNCHRON
0020      :          attempts has been exceeded.
0021      :
0022 END :          SYNCHRON completed successfully
0023      :BE

```

• **STEP 5 operations in OB 1:**

```

SEGMENT 1          0000 Cyclic program processing
0000      :
0001      :
0002      :          Other potential S5 operations
0003      :
0004      :
0005      :JU      FB 117      Send command to be executed
0006 NAME :DOS:COM
0007 TRIG :      I 20.2      Trigger via input I 20.2
0008      :
0009      :JU      FB 126      SEND ALL
000A NAME :SEND-A
000B SSNR :      KY 0,16
000C A-NR :      KY 0,0
000D ANZW :      FW 30
000E PAFE :      FY 34
000F      :
0010      :A      F 34.0      Parameter error in SEND ALL
0011      :JC      PB 1      PB 1 deals with the error.
0012      :          (This will not be explained in detail here.)
0013      :

```

(continued on next page)

(OB 1 cont.):

```
0014      :
0015      :           Other potential S5 operations
0016      :
0018      :
0019      :BE
```

• **Function block FB 117:**

```
SEGMENT 1      0000 Execute MS-DOS command
NAME :DOS:COM
DECL :TRIG      I/Q/D/B/T/C: I BI/BY/W/D: BI

0008      :A      =TRIG      Form positive edge of the
0009      :AN      F 100.4    trigger event
000A      :S      F 100.4
000B      :=      F 100.5    Edge flag
000C      :AN      =TRIG
000D      :R      F 100.4
000E      :
000F      :A      F100.5
0010      :JC      FB 120     Start actual sending
0011 NAME :SEND
0012 SSNR :      KY 0,16     SSNR = Page No. = 16
0013 A-NR :      KY 0,209    A-NR = Job number = 209
0014 ANZW :      FW 40
0015 QTYP :      KS DB      Command to be executed is in
0016 DBNR :      KY 0,10    DB 10 from DW 8 onwards and is 8 words
0017 QANF :      KF +8      long.
0018 QLAE :      KF +8
0019 PAFE :      FY 44
001A      :
001B      :O      F 41.3     Job completed with error
001C      :O      F 44.0     Parametererror
001D      :JC      PB 1      PB 1 deals with the error.
001E      :           (This will not be explained in detail here.)
0020      :
0021      :BE
```

• **Data block DB 10:**

```
0:      KH = 0000;           The file MYFILE should be output on the
1:      KH = 0000;           screen using CPSHELL.
2:      KH = 0000;           The command string is in DW 8 to 14.
3:      KH = 0000;           In DW 15 binary zero is required
4:      KH = 0000;           to terminate the command. Other data
5:      KH = 0000;           may be positioned in front of and behind
6:      KH = 0000;           this command. The file MYFILE must be present
7:      KH = 0000;           in the current directory.
8:      KS = 'TYPE MYFILE;'
```

```
15:     KH = 0000;
16:     KH = 0000;
17:
```

9.3.6 Activation/Termination and Testing of the Command Interpreter

Activation/ Termination

You can activate the command interpreter (initially for the test) once you have carried out all preparatory measures on the CPU and CP 581:

- Make sure that the entry for loading the CPDHB driver (CPDHB) is present in the AUTOEXEC.BAT file in the main directory of the CP 581 hard disk. ¹⁾
- Trigger a warm restart on the CP 581 in order to start the CPDHB driver. ¹⁾
- Carry out a warm or cold restart on the CPU in order to synchronize it with the CP 581.

- **Start:**

Start the command interpreter on the CP 581 by entering the CPSHELL command (if you have already debugged use of the command interpreter and wish to automatically start it during a system restart or warm restart, you must enter the CPSHELL command in the AUTOEXEC.BAT file in the main directory of the CP 581 hard disk).

- **Terminate:**

You can terminate the command interpreter by pressing the ESC key.

The command interpreter outputs the following message on the monitor when started:

```
CPSHELL Vx.x
```

```
Copyright (c) Siemens AG 1991
```

```
CPSHELL ready
```

MS-DOS commands are now sent to the CP 581 from the CPU which you have programmed for the command interpreter application and according to the conditions defined by you, and then executed. The commands are output on the CP 581 monitor as with a keyboard input.

If messages are output by the called MS-DOS function ¹⁾ or the started program, these also appear on the monitor.

¹⁾ If the load command for the driver has been removed from the AUTOEXEC.BAT file for a specific reason, you can load the driver using the keyboard input "CPDHB."

Testing

Proceed in steps to establish whether a command stored on the CPU is sent correctly to the CP 581 and executed:

1. Store a simple, "safe" MS-DOS command in a data block on the CPU. For example, you can output the AUTOEXEC.BAT file from the main directory of the hard disk on the monitor using the command "TYPE AUTOEXEC.BAT" (to ensure that the string has an even number of characters, enter a space after the command in the data block; this has no effect when the command is executed).
2. Generate the STEP 5 program such that the command is triggered by an input (keyboard).
3. Check that calling of the DHBs SYNCHRON, SEND DIRECT and SEND ALL are executed on the CPU without error bits (see Section 9.4).
4. Check that the command appears on the monitor of the CP 581 and that the AUTOEXEC.BAT file is output.

If the command is not correctly transmitted to the CP and executed, you must evaluate the status bits stored by the DHB in the PAFE and in the status words, correct your program as necessary, and repeat the test.

- 1 See Section 9.5 for handling of acknowledgment signals.

9.4 Error bits

You can evaluate the bits of the data handling blocks in order to test your STEP 5 program for use of the command interpreter. Two types of bits are available:

- Parameter assignment error bits

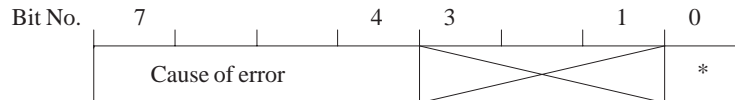
and

- Job status bits

9.4.1 Parameter Assignment Error Bits

All data handling blocks check the transferred parameters for syntax and logical errors when called. In addition, they check whether the addressed interface is available for the triggered function.

Whether parameter assignment errors have occurred and which ones can be seen from the status byte whose address you have specified by the parameter PAFE when calling the DHB:



– Error byte PAFE:

* = Common flag

0:

no error

1:

parameter assignment error, more details in bits 4 to 7

Table 2-1 lists all bits which can be stored by the data handling blocks in the PAFE.

Table 9-5 Parameter Assignment Error Bits

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format
21H	DB or DX data block not present or illegal (e. g. DB 0 or DX 0 with QTYP = DB or DX)
31H	Area too small or total of initial address (QANF/ZANF) and length (QLAE/ZLAE) too large (with all QTYP/ZTYP)
41H	Area does not exists or is illegal (with QTYP/ZTYP = AS, QA, IA, PY)
51H	Status word (address) faulty
61H	Dependent on CPU

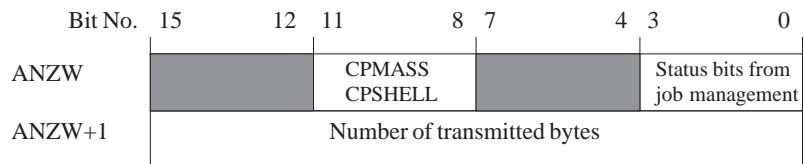
Table 9-5 Parameter Assignment Error Bits

PAFE value	Cause of error
71H	Interface does not exist
81H	Interface not ready
91H	Interface overloaded
A1H	Dependent on CPU
B1H	Job number illegal or frame size (SYNCHRON) illegal
C1H	Interface does not react, or interface does not react at correct time, or interface rejects job
D1H	Dependent on CPU
E1H	Dependent on CPU
F1H	Dependent on CPU

9.4.2 Job Status Bits

When using the command interpreter, bits are set in two declared status words by the DHB SEND and the CPSHELL program via the CPDHB driver. You have defined the initial address of the status words in the parameter ANZW in the DHB calls. Some of the status bits of the DHB SEND/function SEND DIRECT are of importance for the test. These status bits are explained later. You can obtain more information on the DHB status bits from Chapter 13.

The status words of DHB SEND/SEND DIRECT have the following format:



Status bits from job management (bit nos. 0 to 3):

You can obtain the respective status of job processing from these status bits:

Table 9-6 Bits from Job Management

Bit No.	Meaning
0	Bit = 0: Not relevant here
1	Bit = 1: Job for command execution taking place
2	Bit = 1: Job completed without errors
3	Bit = 1: Job completed with errors

The status bits of the job management only indicate the **status of data exchange** between CPU and CP 581. They do not provide information on whether a command (for example,COPY) has been executed properly.

Note

You cannot send jobs to the CP 581 for the respective job number if bits 1 and 2 are set **simultaneously** in the status word. Note this response when restarting your S5 program.

Special flags of the CPSHELL program (bit nos. 8 to 11):

If the CPSHELL command interpreter determines when processing a command that this cannot be executed without an error, it sets error bits in the first status word of the two CPU words addressed with ANZW.

Table 9-7 shows you the status bits which may occur when processing a command.

Table 9-7 Error Bits from the CPSHELL Command Interpreter

Value of bit No. 8 to bit No. 11	Cause of error
0H	No error
1H to 5H	DHB error (also referred to as PLC or CPU error), error numbers PAFE 1 to PAFE 5
6H	Command line is too long
9H	The called program cannot be loaded since the vacant memory space in the CP 581 is insufficient
AH	The called MS-DOS function or the started program addresses a peripheral device of the CP 581 which is not ready, for example, there is no floppy disk in the drive or the printer is not ready.

9.5 Special Features During Command Interpretation

You must observe the following special features during processing of a command when selecting commands or programs which you wish to execute or start using a CPU:

- **User programs:**
A command interpretation is only terminated when a started program has been terminated (only then can a new command be sent by a CPU). This requires that a user program can only wait for an operator input if a keyboard is present on the CP 581.
- **Handling of MS-DOS acknowledgment bits:**
If a bit is to be output which can be acknowledged by MS-DOS, because for example, a floppy disk is not present in an addressed drive, this output is suppressed by CPSHELL. A bit is set in the first status word of the DHB call SEND on the CPU from which the command was sent (see Section 9.4.2).
- **Commands from several CPUs:**
If you wish to use the command interpreter from several CPUs, you must coordinate the command output of the individual CPUs. Refer to the description of your programmable controller in order to learn how this is carried out.



Caution

Data losses may occur in the event of uncoordinated execution of certain commands (for example, COPY) from several CPUs.

Virtual S5 Drive

10

This chapter is intended for users who wish to address S5 data handling blocks of a CPU like MS-DOS files on a “Virtual drive” using the CP 581. The chapter describes all measures and operations required on the CP 581 and the CPU: Extensive preparations are only necessary on the CP 581 if you wish to read or store the S5 data as ASCII characters and not as binary data (see Section 10.3.2). You must adapt your STEP 5 programs on the CPUs in order to use the virtual S5 drive (Section 10.3.3). For this you should have experience in programming programmable controllers. Section 10.3.3 provides you with information you must observe when using the virtual S5 drive. This section also provides you with information on the reactions to errors and on error displays.

The programming example in Section 10.3.3 (page 10-14) only serves to explain the DHB parameters for data exchange. Complete programs which can be executed can be found in Chapter 12.

Chapter Overview

Section	Description	Page
10.1	Application	10-2
10.2	Principle Sequences Between CPU and CP	10-2
10.3	Preparation of Application	10-4
10.3.1	Related Procedures	10-4
10.3.2	Measures on the CP	10-5
10.3.3	Programming the CPU	10-10
10.4	Activation, Deactivation and Testing of the Function	10-17
10.4.1	Activation	10-17
10.4.2	Deactivation	10-19
10.4.3	Testing	10-19
10.5	Use of Virtual S5 Drive	10-21
10.5.1	Operations with Directories and Files	10-21
10.5.2	Writing in S5 Data Blocks	10-22
10.5.3	Representation of S5 Data on the CP 581	10-23
10.5.4	Use of MS-DOS Commands and Applications	10-26
10.5.5	Error Status Bits	10-27
10.6	Conversion of Preheader Data Using DVCONVRT	10-30

10.1 Application

The virtual S5 drive function handles the S5 data blocks DB and DX like MS-DOS files on a virtual drive. You can access data blocks of up to 4 CPUs from the CP 581 using this function, and read or also write back S5 data. Data exchange is either binary (S5REMOTE program) or formatted (S5REMOTF - ASCII conversion when reading, reverse conversion when writing).

The function is suitable for rapidly obtaining information on the CP 581 on the extent and contents of S5 data blocks or for writing data into S5 data blocks. You can process the data transmitted from an S5 data block to the CP 581 in binary mode or as ASCII files using appropriate MS-DOS applications.

You may only use those MS-DOS commands or MS-DOS applications for the virtual S5 drive which are **network-compatible** (see /1/ Reference Literature).

10.2 Principle Sequences Between CPU and CP

Figure 10-1 shows you the principle functional sequence with formatted reading of data from an S5 data block and subsequent output of these data on the monitor.

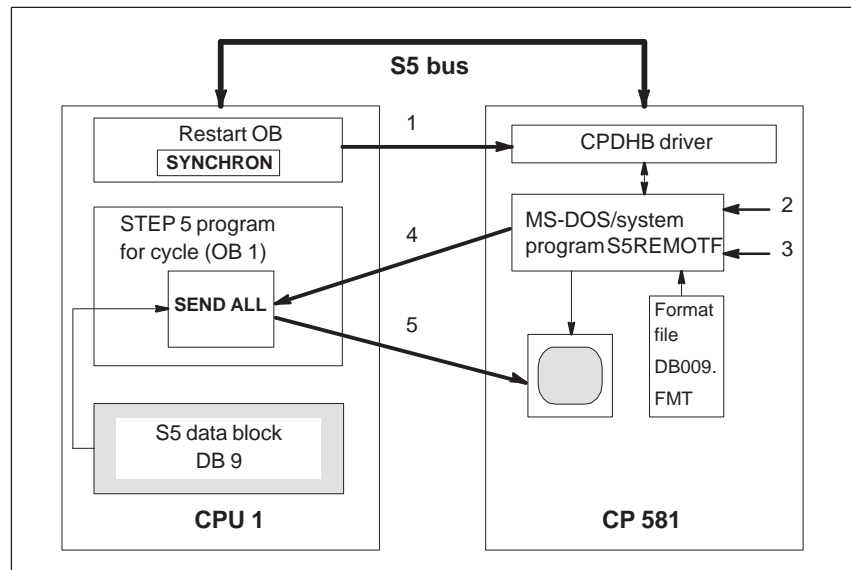


Figure 10-1 Basic Sequence when Listing an S5 Data Block with S5REMOTF

- 1 Synchronization of CPU during restart with CP via DHB SYNCHRON. The frame size for subsequent data transfer is set in the process.
- 2 The S5REMOTF program is loaded and the parameters are set for formatting.
- 3 The command 'TYPE S:\CPU1\DB009.S5A' is used:
- 4 to read the data from the S5 data block DB 9 on CPU 1 and
- 5 to convert the data via the format file DB009.FMT into ASCII characters and to output these on the monitor.

Figure 10-2 shows the basic sequence when copying a binary file from the CP 581 to a CPU.

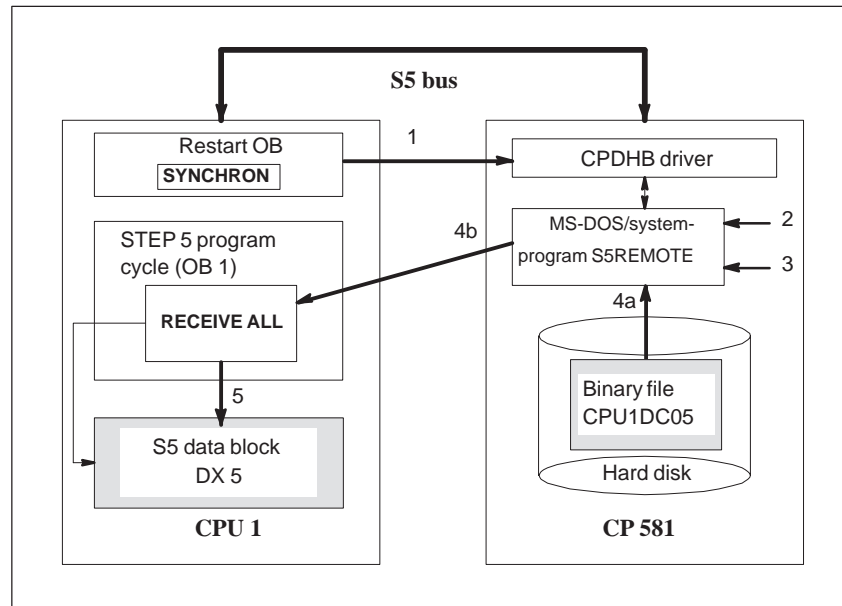


Figure 10-2 Basic Sequence when Copying a Binary File into an S5 Data Block with S5REMOTE

- 1 Synchronization of CPU during restart with CP via DHB SYNCHRON. The frame size for subsequent data transfer is set in the process.
- 2 The S5REMOTE program is loaded and the parameters are set.
- 3 The command 'COPY C:CPU1DX05 S:\CPU1\DX005.S5B' is used:
- 4 to read the local binary file CPU1DX05 from drive C of the CP 581 (4a), to send it to the CPU 1 (4b), and
- 5 to copy it into the S5 block DX 05.

10.3 Preparation of Application

10.3.1 Related Procedures

You must carry out the following preparations in the suggested sequence in order to use the virtual S5 drive function:

Table 10-1 Measures for Preparation of Application

Measure	on	See Section
Make sure that the CPU with which you wish to exchange data and the CP 581 are plugged into your programmable controller. The base interface number for data transfer must be set on the CP 581.	PLC CP 581	10.3.2
Define whether S5 data have to be converted into ASCII code for display on the CP 581.	--	10.3.2
If ASCII conversion is necessary: define the assignment "Data block - conversion rules" and select the conversion convention(s) for the S5REMOTF program.	--	10.3.2
If you wish to output or process the data of one or more data blocks as ASCII characters : edit one or more files on the CP 581 using format inputs or load the programming device files <code>?????ST.S5D</code> with the preheader data of the data block(s) as an MS-DOS file on the CP 581 and convert these into formatted files.	CP 581 or PG	10.3.2
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG CPU	10.3.3
Create or modify the STEP 5 program on the CPU for the cycle such that data can be sent to the CP 581 using DHB calls and also received from it.	PG CPU	10.3.3
Make sure by carrying out a test on the CPU that the DHB to be called in the restart/warm restart or cyclically is called correctly.	PG CPU	--
Make sure that the CPDHB driver is loaded in the memory of the CP 581 (possibly cold restart of CP 581).	CP 581	--
Synchronize the CP 581 with the CPU by triggering a restart of the CPU.	CPU	--
Load the S5REMOTE or S5REMOTF program and enter the appropriate parameters.	CP 581	10.4.1
Check the correct sequence of desired functions by testing on the CPU and with an MS-DOS command which addresses the set virtual S5 drive.	PG CPU CP 581	10.4.3

10.3.2 Measures on the CP

Setting the Base Interface Number

So-called pages (page frames) are used to address the memory areas when transferring data on the S5 bus. These pages have a fixed assignment to the modules involved in data transfer.

The CP 581 can transfer data with up to four CPUs via four successive pages. The number of the first page is the base interface number.

Define this number for the first inserted CPU, and set it on the CP 581 as described in Section 6.2.3. The pages for data transfer with three further CPUs consecutively follow the page with the base interface number.

Defining the S5 Data Representation

The data from the S5 data blocks can be stored and represented on the CP 581 as binary data or as ASCII characters.

Before using the “virtual S5 drive” function, answer the following questions:

- Must the S5 data be converted?
- Which data blocks must be converted for formatted reading and writing?

Use the S5REMOTE program for **binary** storage. No further preparations are necessary on the CP 581.

Use the S5REMOTF program if you wish to store or display the S5 data as **ASCII characters** (“formatted”) with the CP 581, and declare a formatting mode. You have the following possibilities:

- Only **one** particular S5 data block of **one** CPU is converted.
- **All** S5 data blocks of **one** CPU are converted.
- **One** particular S5 data block of **all** CPUs is converted.
- **All** S5 data blocks of **all** CPUs are converted.

In all cases, enter **one** format declaration for the conversion.

Processing Steps for ASCII storage/representa tion

Step 1: Define parameters for formatting

You parameterize the mode for converting the S5 data later when installing the **S5REMOTF** program with the option **F** (see Section 10.4.1). You should already define the parameters now, however, since the further processing steps are influenced by them. The option has the following syntax:

“-F format CPU No. DB/DX No. Field delimiter”

The meanings of the parameters are shown in the following table.

Table 10-2 Parameters of the ‘F’ Option for Conversion of S5 Data

Parameters	Meaning	Permissible values
Format	Path and name of file with format specification	MS-DOS syntax, file name for data block: DB001 to DB255 for DB ¹⁾ DX000 to DX255 for DX
CPU No.	Number of CPU with which the S5 data to be converted are to be exchanged	1 to 4 for a particular CPU or * for all connected CPUs
DB/DX No.	Number of data block on the CPU which is to be converted according to the specification	1 to 255 for DB 1 to DB 255 ¹⁾ 256 to 511 for DX 0 to DX 255 or * for all DBs and DXs
Field delimiter	The individual S5 data in the converted ASCII files on the CP 581 are separated from another by the field delimiter (see Section 10.5.2). The component is optional; if it is not specified the S5 data are separated by a space.	ASCII characters; numbers and the characters - / < > + . are illegal.

1) The data blocks which you can read or write depend on the CPU (system data blocks).

Examples of “-F” parameters:

1. Formatting of data block DB 9 on CPU 1 using the format file DB009.FMT. The delimiter is the default setting “Space:”

```
“-F DB009.FMT 1 9”
```

2. Formatting of data block DX 4 on CPU 2 using the format file DX004.FMT. The delimiter is “*.”

```
“-F DX004.FMT 2 260 *”
```

(the numbers for DX are 256 to 511)

3. Formatting of all DB and DX data blocks of all CPUs using the format file ALLFORM.FMT. The delimiter is “\$.”

```
“-F ALLFORM.FMT * * $”
```

Step 2a: Editing the format file :

If you wish to process S5 data blocks formatted on the CP 581, you must prepare one or more format files. You must observe the following conventions for these files:

- **Name of the format file and directory:**
The name of the format file can be selected as desired according to the MS-DOS standard.
- **Structure of the format file:**
A format file consists of one or more text lines (ASCII characters). Each line can consist of one or more format instructions or a comment. It must be terminated by CR and LF.

- **Syntax of a format instruction:**

A format instruction has the following syntax:

Repetition factor (optional)	Format data	Delimiter
---------------------------------	-------------	-----------

- **Repetition factor:**

Specifies how many successive items of data are to be converted with the same data format. If it is not specified, the subsequent format instruction is implicitly assigned the repetition factor “1.” Since two characters are always stored under the S5 format KS, the repetition factor with the format instruction KS always specifies twice the number of characters to be processed.

Permissible values: 1 to 4091

- **Format data:**

Permissible values: KC for S5 format KC,
KF for S5 format KF,
KG for S5 format KG,
KH for S5 format KH,
KM for S5 format KM,
KS for S5 format KS,
KT for S5 format KT,
KY for S5 format KY.

- **Delimiter:**

Permissible values: (Character / ASCII decimal equivalent):
Space / 32,
Comma / 44,
Horizontal tab. / 9,
CR + LF / 13 + 10.

- **Comment line:**

A line is not interpreted if it commences with the character ‘;’. You can then use comments in a format file.

Example of a format file:

Name: MYFORM.FMT

Contents:

;3 fixed-point numbers:

3KF

;4 floating-point numbers:

4KG

;6 characters:

3KC

;2 Bit patterns:

2KM

**Step 2b: Generation of a format file
from the preheader data of a data block**

If you have already generated data blocks for your STEP 5 program on the programming device, you can also use the preheader data of these data blocks for the conversion. You need not edit a format file in this case.

The preheader data are generated on the programming device when programming the data block and contain information on the formats of the individual data words in the data block. In order to convert the S5 data, you must make the preheader data available on the CP 581 and convert them into a format file. Proceed as follows:

- Call the S5 software package on the programming device, define a data block DB or DX and enter the data in the desired format. The data have the function here of token characters. Store the data block on the programming device in a file ??????ST.S5D (?????? = any letters/sequence of numbers, “@” as filler); following storage, this file contains the data of the data block and the preheader information.
- Copy this file in MS-DOS format onto a 3.5” floppy disk. If you do not have a programming device with S5 software under the MS-DOS or FlexOS operating system, you must transfer the S5 file from the PCP/M format onto a floppy disk formatted for MS-DOS using an appropriate program.
- Load the file from the floppy disk onto your CP 581 using the MS-DOS command COPY.
- Convert the file with the preheader data into a format file with the name extension **FMT** using the system service routine **DVCONVTR** (see Section 10.6).

10.3.3 Programming the CPU

Principle

Programming of the CPU for the virtual S5 drive comprises the synchronization of the CP 581 during the restart and the calling of special function blocks for data exchange with the CP 581. You require so-called “data handling blocks” (DHB) for both functions: the DHB SYNCHRON for synchronization, the DHB SEND for transmitting data and the DHB RECEIVE for receiving data.

The following figure shows you the positions in your STEP 5 program at which you must call data handling blocks.

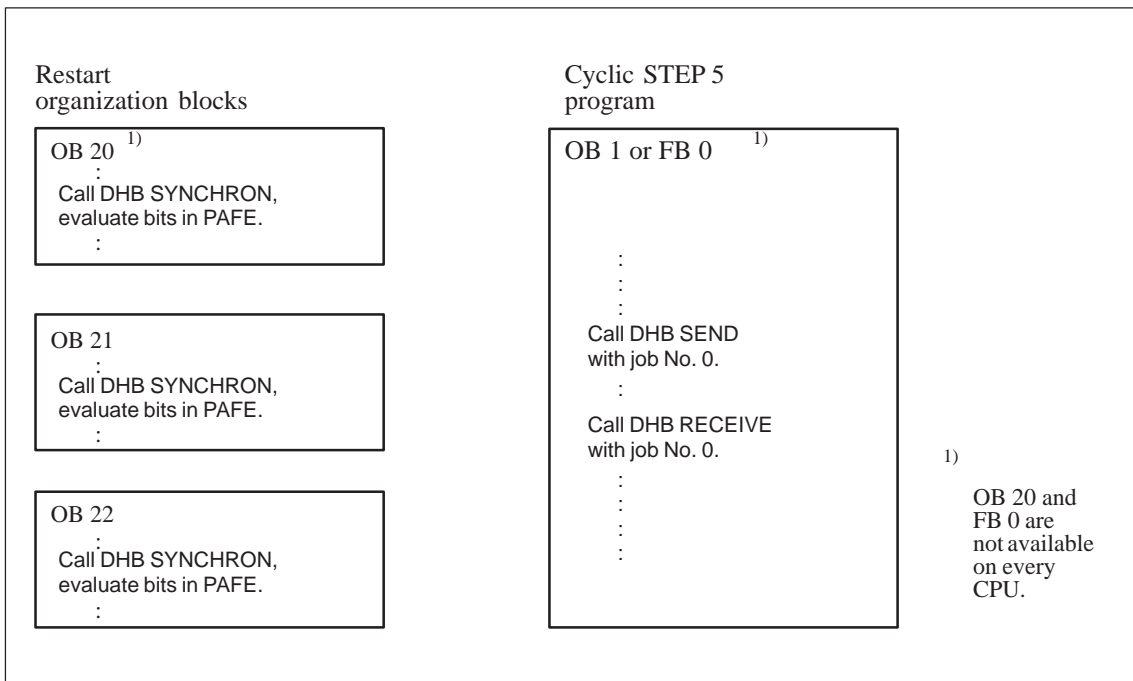


Figure 10-3 Principle of DHB Calls in the STEP 5 Program of the CPU for the Virtual S5 Drive

Calling and Parameterizing the Data Handling Blocks

The functions of the data handling blocks which you require for data exchange with the CP 581 are described in this section as far as is necessary for programming.

If you use an S5-135U or S5-155U programmable controller, you can call the DHB SEND-A and DHB REC-A available there in your STEP 5 program instead of the DHB SEND. You can obtain information on these DHBs from the corresponding descriptions of the DHBs for the S5-135U and S5-155U programmable controllers (see /8/ and /9/ Reference Literature).

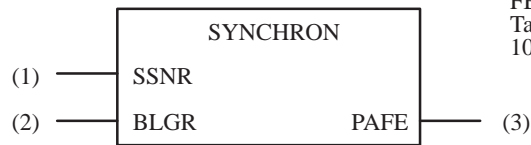
The data handling blocks have different block numbers on the various programmable controllers. The following table contains the numbers of the DHBs SYNCHRON, SEND and RECEIVE which you require for the various programmable controllers.

Table 10-3 DHB Numbers on the Various Programmable Controllers

	S5-115U	S5-135U	S5-155U
DHB SYNCHRON	FB 249	FB 125	FB 125
DHB SEND	FB 244	FB 120	FB 120
DHB RECEIVE	FB 245	FB 121	FB 121

DHB SYNCHRON

Block diagram



Use the block No.
FB xxx from
Table
10-3.

Table 10-4 Format and Meaning of Parameters for SYNCHRON

Parameters	Kind	Type	Meaning
SSNR	D	KY	Interface number
BLGR	D	KY	Frame size
PAFE	Q	BY	Parameter assignment error bits

Set the following parameters for the DHB SYNCHRON:

– SSNR:

Enter the number of the page via which you wish to read the data from the CPU: “(CPU No. - 1) + base interface No.” (see Section 10.3.2).

Permissible values: 0 to 255 ¹⁾

– BLGR:

Use this parameter to define the maximum number of bytes to be transmitted to the CP 581 in a CPU cycle.

Permissible values: 0 = standard values (see following table)

1 = 16 bytes

2 = 32 bytes

3 = 64 bytes

4 = 128 bytes

5 = 256 bytes

6 = 512 bytes

1) The base interface number must be set in steps of four (0, 4, 8, 12 etc.)!

Standard values for BLGR = 0	
S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

– PAFE:

Byte address for parameter assignment error bits. Bit No. 0 of the status byte is set to “1” in the case of a parameter assignment error (refer to Section 10.5.5 on page 10-28 for the meaning of the other status bits).

Note

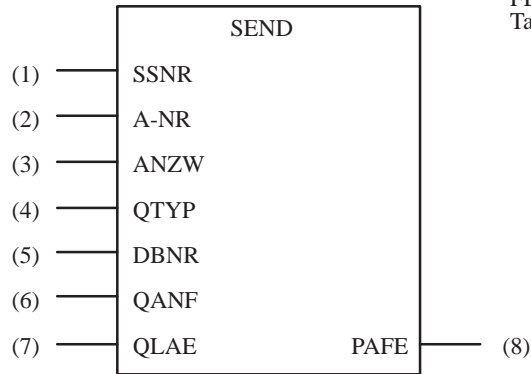
The CP 581 may require more time to run up than the CPU. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Make sure, however, that the program loop is **always** terminated by an abort criterion (timer or loop counter).

Larger data quantities are transmitted faster the larger you set the frame size BLGR. The S5 bus is loaded more in the process, however. Vice-versa, the bus loading is lower with smaller frame sizes, but data transmission to the CP 581 takes longer. You must decide which frame size is most favorable for your CPU depending on the aspect of the complete operation on the programmable controller.

DHB SEND and DHB RECEIVE:

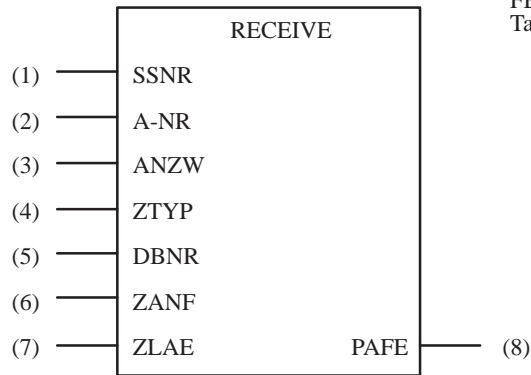
The CPU receives a specific request for each transmission process when exchanging data with the CP 581. The initiation is from the S5REMOTE/S5REMOTF program. This initiation is passed on by the CPDHB driver to the CPU. To enable the CPU to react to it, the DHB SEND with the parameter assignment for the function “SEND ALL” and the DHB RECEIVE with the parameter assignment for the function “RECEIVE ALL” must be called in every cycle.

Block diagram



Use the block No. FB xxx from Table 10-3.

Block diagram



Use the block No. FB xxx from Table 10-3.

Table 10-5 Format and Meaning of Parameters for SEND and RECEIVE (ALL)

Parameters	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
QTYP ZTYP	D	KS	These parameters are irrelevant with the functions "SEND ALL" and "RECEIVE ALL;" they must be entered for format reasons, however.
DBNR	D	KY	
QANF ZANF	D	KF	
QLAE ZLAE	D	KF	
PAFE	Q	BY	Parameter assignment error bits

- **STEP 5 operations in OB 20, OB 21 and OB 22:**

```

SEGMENT 1          0000

0000              :
0001              :           Other potential S5 operations
0002              :
0003              :
0004              :JU   FB 111   Synchronize CP 581
0005  NAME :CPSYNC
0006  REP  :      KF +3   Number of SYNCHRON attempts
0007              :
0008              :
0009              :           Other potential S5 operations
000A              :
000B              :BE

```

- **Function block FB 111:**

```

SEGMENT 1          0000
NAME :CPSYNC
DECL :REP          I/Q/D/B/T/Z: D KM/KH/KY/KS/KF/KT/KC/KG: KF

0008              :L   KB 0
0009              :T   FY 10   Initialize loop counter
000A              :
000B  LOOP :           Return marker
000C              :JU FB 125   Synchronize CP 581
000D  NAME :SYNCHRON
000E  SSNR :      KY 0,16   Page No. = SSNR = 16
000F  BLGR :      KY 0,0    Frame size = 0 => 256 bytes
0010  PAFE :      FY 11    Parameter error in FY 11
0011              :
0012              :AN   F 11.0
0013              :JC   =END   The block is completed without error
0014              :           only if no parameter error has occurred.

0015              :L   FY 10   Increase loop counter by one on
0016              :I     1       every SYNCHRON run
0017              :T   FY 10   (abort criteria)
0018              :
0019              :L   FY 10   Check whether the maximum
001A              :LW  =REP   permissible number of runs
001B              :<=F   has been reached.
001C              :JC   =LOOP
001D              :
001E              :STS   PLC STOP following a PAFE, or
001F              :           if the set number of SYNCHRON attempts
0020              :           has been exceeded.
0021              :
0022  END  :           SYNCHRON completed successfully.
0023              :BE

```

• **STEP 5 operations in OB 1:**

```

SEGMENT 1          0000 Cyclic program processing
0000              :
0001              :
0002              :           Other potential S5 operations
0003              :
0004              :JU   FB 120   Send data to CP 581
0005  NAME :SEND
0006  SSNR :   KY 0,16   Page No. = SSNR = 16
0007  A-NR :   KY 0,0    Job No. = 0 ("SEND ALL")
0008  ANZW :   FW 30
0009  QTYP :   KS       irrelevant
000A  DBNR :   KY 0,0    irrelevant
000B  QANF :   KF +0     irrelevant
000C  QLAE :   KF +0     irrelevant
000D  PAFE :   FY 34
000E              :
000F              :A   F 34.0   Parameter error in SEND
0010              :JC   PB 1    Errors are dealt with in PB 1.
0011              :           (This will not be explained in detail here.)
0012              :
0013              :JU   FB 121   Receive data from CP 581
0014  NAME :RECEIVE
0015  SSNR :   KY 0,16   Page No. = SSNR = 16
0016  A-NR :   KY 0,0    Job No. = 0 ("RECEIVE ALL")
0017  ANZW :   FW 40
0018  ZTYP :   KS       irrelevant
0019  DBNR :   KY 0,0    irrelevant
001A  ZANF :   KF +0     irrelevant
001B  ZLAE :   KF +0     irrelevant
001C  PAFE :   FY 44
001D              :
001E              :A   F 44.0   Parameter error in RECEIVE
001F              :JC   PB 1    Errors are dealt with in PB 1.
0020              :           (This will not be explained in detail here.)
0021              :
0022              :           Other potential S5 operations
0023              :
0024              :BE

```

10.4 Activation, Deactivation and Testing of the Function

10.4.1 Activation

The following two system programs are available for the virtual S5 drive function: **S5REMOTE** for **binary** conversion and **S5REMOTF** for **formatted** conversion of the S5 data.

Once you have completed all preparations on the CPU and CP 581, you can activate the virtual S5 drive via one of the two programs:

- Make sure that the entry for loading the CPDHB driver (“CPDHB”) is present in the AUTOEXEC.BAT file in the main directory of the CP 581 hard disk. ¹⁾
- Trigger a reset (boot) on the CP 581 in order to start the CPDHB driver. ¹⁾
- Carry out a restart on the CPU in order to synchronize the CP 581.
- Define the sequence parameters for S5REMOTE/S5REMOTF. You must enter these parameters with the following syntax in the command for loading the program:

S5REMOTE LW /N /T /C /H /? or

S5REMOTF LW /N /T /F /C /H /?

- Parameter LW:

Logical name for the virtual S5 drive. The drive name must be a permissible MS-DOS drive (“Letter:” or “Letter”) and must not be higher in the alphabet than the name set in the CONFIG.SYS file for “lastdrive.”

- Options N, T, F, C, H/?:

These parameters are optional. Their meaning is listed in the following table. You can also use “-” for the option code instead of “/”, and also lower-case letters for the option instead of upper-case letters.

- 1) If the load command for the driver was removed from the AUTOEXEC.BAT file for essential reasons, you can load the driver by entering “CPDHB” on the keyboard.

Table 10-6 Meaning of the Optional Parameters in the Command “S5REMOTE/S5REMOTF”

Option	Meaning (notation)
N	No error messages are output on the monitor.
T	Timeout in seconds. Values from 1 to 3640 are permissible; the default value is 30 seconds.
F	Format “F Format CPU No. DB/DX No. [Field delimiter]” (see Section 10.3.2 / [..] = optional): This option is only permissible for the S5REMOTF program. It is used to define the conversion mode. If you do not enter this option with S5REMOTF, data transfer is binary as with S5REMOTE.
C	The deletion of DB/DX is rejected.
H/?	Output of explanations on S5REMOTE/S5REMOTF.

- Enter the **S5REMOTE** command in order to load S5REMOTE, or the **S5REMOTF** command in order to load S5REMOTF together with the desired parameters.
 - S5REMOTE or S5REMOTF is loaded by MS-DOS into the memory of the CP 581 and the parameters are assigned.
 - If the virtual S5 drive with a network-compatible function is addressed in an MS-DOS command or in an MS-DOS application, MS-DOS activates the S5REMOTE or S5REMOTF program using an interrupt. The program then handles the desired function as if the addressed S5 data were present on a real drive.

Command examples

- s5remote s -n**
- Start S5REMOTE program
 - Virtual drive “S” (s)
 - No output of error messages (-n)
- S5REMOTE g /T20**
- Start S5REMOTE program
 - Virtual drive “G” (g)
 - Monitoring time = 20 seconds (/T20)
- s5remotf p -f db009.fmt 1 9**
- Start S5REMOTF program
 - Virtual drive “P” (p)
 - Convert data of a data block (-f...):
 - Data block DB 9 (9)
 - on CPU 1 (1)
 - via format file DB009.FMT (db009.fmt)

10.4.2 Deactivation

In order to deactivate the “virtual S5 drive” function you must remove the installed S5REMOTE or S5REMOTF program from the memory using the command:

S5REMOTE /U or S5REMOTF /U

You can also use “-” instead of the character “/”, and also “u” instead of “U.”

10.4.3 Testing

In order to find out whether you can correctly handle the data transfer with one or more CPUs via the virtual S5 drive, proceed in the following steps:

1. Prepare a data block on the PG (e. g.DB 20) for the test and fill it with decimal numbers which are also easy to recognize in a hexadecimal representation (for example 17/11H, 33/21H, 65/41H, 129/81H etc.). Transfer the block using the PG to the CPU with the number 1.
2. Make sure that the calls of the DHBs SYNCHRON, SEND/ALL and RECEIVE/ALL on the CPU with the number 1 are executed without errors.
3. Activate the virtual S5 drive under the name “S:” initially using the S5REMOTE program.
4. Enter the MS-DOS command.

DIR S:\CPU1

The directory S:\CPU1\ is then listed and all data blocks DB and DX present on the CPU under the name “DBnnn.S5B” or “DXnnn.S5B” (the generation of the file names is described in Section 10.5.1.)

5. Copy the test data block (DB 20) from the virtual S5 drive into a supplementary directory (C:\S5CPU1) using the MS-DOS command.

COPY S:\CPU1\DB020.S5B C:\S5CPU1

6. List the contents of the file DB020.S5B in the directory S5CPU1 as hexadecimal numbers using a suitable MS-DOS program and check whether the values agree with those of the data block.
7. Generate a format file DB020.FMT in the directory S5CPU1 on the CP 581. Select “KF” as the format for the complete test data block (DB 20).
8. Deactivate the S5REMOTE program and activate the S5REMOTF program.
Enter the following values for the formatting mode:

-F C:\S5CPU1\DB020.FMT 1 20 *

(Path name of the format file “C:\S5CPU1\DB020.FMT”, CPU 1, data block **DB 20**, field delimiter “*”)

- Repeat steps 4. to 6. with the same data block (DB 20) and the following modification:

The data block DB 20 must appear with the name “DB020.S5A” in the directory S:\CPU1. Copy this file into the supplementary directory and list the file contents using the MS-DOS command

TYPE C:\S5CPU1\DB020.S5A

The values of the data block must now appear on the monitor as decimal numbers separated by “*”.

If the function is not executed correctly, you must evaluate the status bits or error messages of the S5REMOTE/S5REMOTF programs stored by the DHB in PAFE and carry out appropriate corrections on the STEP 5 program or the sequence parameters. You can then repeat the test completely or start at a particular step.

10.5 Use of Virtual S5 Drive

10.5.1 Operations with Directories and Files

Directories

The S5REMOTE/S5REMOTF programs generate the directories CPU1 to CPU4 on the virtual S5 drive for four CPUs. If one of the four CPUs is not connected or is not synchronous, the corresponding directory is displayed by the programs as being empty when listed with DIR.

You cannot generate any further subdirectories in the four directories of the virtual S5 drive.

Files

File names

The names for the S5 data blocks are derived from the DB names and numbers. The name extension provides information on whether a format was specified for an S5 file or not:

DB001.S5B to **DB255.S5B** for **binary** files for the data blocks **DB**
DX000.S5B to **DX255.S5B** for **binary** files for the data blocks **DX**

DB001.S5A to **DB255.S5A** for **ASCII** files for the data blocks **DB**
DX000.S5A to **DX255.S5A** for **ASCII** files for the data blocks **DX**

Attributes

You can assign the same file attributes to the S5 files as to other MS-DOS files, for example “+r” for the read-only attribute.

Time stamp

The time stamp is switched off by the system programs for the files of the virtual S5 drive (date = 0, time = 0). When copying an S5 file into a local CP 581 file, the former is assigned the current time stamp, however.

Copying

You can copy S5 files from the virtual S5 drive onto a local drive of the CP 581 (for example hard disk or floppy disk). If you have agreed on a format for an S5 file, the S5REMOTF program converts the contents of this file into ASCII characters prior to copying (see Section 10.5.3).

Writing into the destination file is carried out as with a physical drive (MS-DOS function).

Listing and Printing

S5 files can be listed on the monitor or printed out using the MS-DOS commands TYPE or PRINT. In order to achieve a readable output, you should only use these commands on S5 files for which you have specified a format. The S5 data are then converted by S5REMOTF into ASCII characters as when copying.

In order to output a binary S5 file on a monitor or printer, use a suitable MS-DOS program which converts the binary data, for example into hexadecimal numbers (also refer to Section 10.5.3!)

10.5.2 Writing in S5 Data Blocks

You can copy files from local drives of the CP 581 into an S5 data block DB or DX. If you have specified a format for the corresponding S5 file, the S5REMOTF program converts the ASCII data into the binary representation prior to transmission.



Caution

Writing into S5 data blocks is only possible in the RUN status of a CPU. Faulty data may result in major interferences in the technological sequence on your programmable controller. Only use this function with extreme caution.

You should take the following precautionary measures if you wish to write data into a CPU via the virtual S5 drive:

- Assign all S5 files of the CPU into which you do not wish to write with the file attribute “+r” (read-only).
- If possible only write using your own MS-DOS applications where you have carried out coordination and safety measures in accordance with your S5 programs.
- Use the same format file to write an ASCII file as you used when reading these S5 data.
- Never change the length of strings of the S5 format “KS,” only change their contents (do not delete or insert characters!).
- If you wish to edit the data of a data block on the CP 581, only write the result back to the CPU in test mode (like a new or corrected PB or FB) and not in “hot” automation mode.

Handling of S5 data blocks when writing

- The destination data block must be present; it is **not** created!
- If the source and destination data (binary representation) have different lengths:
 - The length of the source data is shorter than that of the S5 data: the data block is overwritten from the beginning by the data present in the MS-DOS file. The remaining S5 data in the data block are retained.
 - The length of the source data is longer than that of the S5 data: as many source data are written into the data block as fit in. The remainder is not transmitted. The complete S5 data in the data block are overwritten.

10.5.3 Representation of S5 Data on the CP 581

Binary Representation

The data are stored in the CP 581 in the Intel format. This differs from the representation of the S5 data in a CPU.

You must take this difference into consideration if you wish to process binary data read from a CPU on the CP 581 or to compare their correctness with the S5 data.

The following figure shows you how the data of an S5 data block are stored in the CP 581.

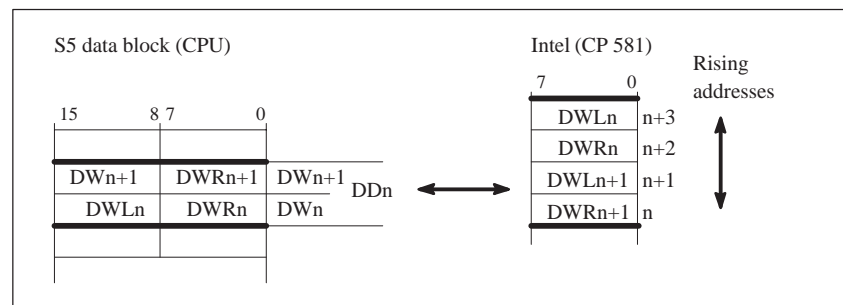


Figure 10-4 Binary Representation of S5 Data on the CPU and on the CP 581

ASCII Representation

In the case of the ASCII representation, the S5 data are stored on the CP 581 according to the declared format. The following table shows you which strings can result from the individual S5 formats.

Table 10-7 Representation of S5 Data in an ASCII File

Data in S5 format	Characters in ASCII file ¹⁾	Field length
Format instruction 'KS' ²⁾		
KS = AXBYCZ	"AXBYCZ"	2 * repetition factor in format instruction
Format instruction 'KF'		
KF = +257	"^^+257"	6 characters, leading spaces
Format instruction 'KG' ³⁾		
KG = +5500000+01	"+5.500000e+000" e = characters for exponent Mantissa: sign, 1 number before decimal point, 6 numbers after decimal point Exponent: sign, 3 digits	14 characters, leading zeros

Table 10-7 Representation of S5 Data in an ASCII File

Data in S5 format	Characters in ASCII file ¹⁾	Field length
Format instruction 'KH'		
KH = 073E	"073E"	4 characters, leading zeros
Format instruction 'KM'		
KM = 0000 0111 0011 1110	The value of the bit pattern (0 to 65 535) is converted into an unsigned decimal number: " ^1854"	5 characters leading spaces
Format instruction 'KT'		
KT = 032.2	The time value is calculated in hundredths of a second from "Value" and "Timebase:" "^^^3200"	7 characters leading spaces
Format instruction 'KY'		
KY = 007,062	Two successive bytes of the S5 area are converted into two three-digit decimal numbers without sign and stored in two fields : "^^"Delimiter"^62"	2 * 3 characters leading spaces
Format instruction 'KC'		
KC = 032	"^32"	3 characters leading spaces

1) ^ = space

2) A string is stored in the ASCII file with the format "KS." The end criterion of the file is:

- a) a change in format,
- b) a binary zero in the S5 data stream.

3) Only those S5 data are correctly converted into floating-point numbers which have been generated either via a KG format with the data handling block editor on the programming device (with data blocks) or with an S5 floating-point operation.

Example for ASCII Representation

The following table shows you an example of how S5 data of a data block are converted and represented using format instructions or stored in an ASCII file. “*” should be declared as the field delimiter.

Tab: Example of ASCII storage of S5 data

Table 10-8 Example of ASCII Storage of S5 Data

S5 data block		Format file/ format instruction	ASCII file	
DW No.	Contents of DW		Field No.	ASCII characters
0	KF = +123	;3 fixed-point numbers: 3KF	0	^^+123*
1	KF = -4567		1	^-4567*
2	KF = +32123		2	+32123
3	KG = +5500000+01	;4 floating-point numbers: 4KG	3	+5.500000e+000*
4				
5	KG = +5500000+03		4	+5.500000e+002*
6				
7	KG = -3410000-02		5	-3.410000e-003*
8				
9	KG = -1234567+00		6	-1.234567e-001*
10				
11	KS = EX	;6 characters: 3KC	7	EXAMPL*
12	KS = AM			
13	KS = PL			
14	KM = 1111 1010 0000 0000	;2 bit patterns: 2KM	8	64000*
15	KM = 0000 1111 0000 1111		9	^3855*

10.5.4 Use of MS-DOS Commands and Applications

You can address the virtual S5 drive using all MS-DOS commands which are network compatible, i.e. which only address a defined drive logically and not physically. You must not use several MS-DOS commands (as well as their equivalent MS-DOS CALLS in applications) for the virtual S5 drive, however. These are listed in the table below.

Network drivers such as MS-NET which use the DOS-internal network identification cannot access the virtual S5 drive since this is used by S5REMOTE and S5REMOTF themselves.

Note

The attempt to delete an S5 data block is rejected **without an error message** (default setting).

It is possible to signal such attempts at deleting by the virtual S5 drive by using the option “C” in the S5REMOTE/S5REMOTF commands (see Section 10.4.1).

The following table shows you which MS-DOS commands you must not use for the virtual S5 drive.

Tab: Illegal MS-DOS commands

Table 10-9 Illegal MS-DOS Commands

Illegal MS-DOS commands for the virtual S5 drive, and functions of equal significance in MS-DOS applications
BACKUP
CHKDSK
DEL (ERASE)
DISKCOMP
DISKCOPY
DRIVPARM (in CONFIG.SYS file)
EXPAND
FASTOPEN
FDISK
FORMAT
JOIN
LABEL
MIRROR
MKDIR (MD)
RECOVER
REN (RENAME)

Table 10-9 Illegal MS-DOS Commands

Illegal MS-DOS commands for the virtual S5 drive, and functions of equal significance in MS-DOS applications
RESTORE
RMDIR (RD)
SUBST
SYS
UNDELETE
UNFORMAT

10.5.5 Error Status Bits

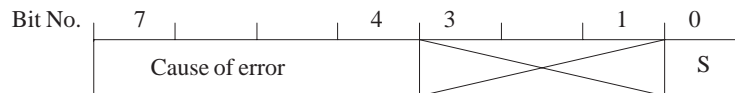
The following information is provided on errors when using the virtual drive:

- Parameter assignment error bits from the data handling blocks (PAFE),
 - Error bits from MS-DOS (extended error code; these bits are only of significance with self-developed MS-DOS applications),
- and
- Error messages from S5REMOTE/S5REMOTF.

Parameter Assignment Error Bits

All data handling blocks check the transferred parameters for syntax and logical errors when called. In addition, they check whether the addressed interface is available for the triggered function.

Whether parameter assignment errors have occurred, and which ones, can be seen from the status byte whose address you have specified by the parameter PAFE when calling the DHB:



– Error byte PAFE:

S = Common error:

0:

no error

1:

parameter assignment error, more details in bits 4 to 7

Table 10-10 lists all bits which can be stored by the data handling blocks in PAFE.

Table 10-10 Parameter Assignment Error Bits

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format (this bit is permissible in the S5-115U because DX is omitted).
21H	DB or DX data block not present or illegal (e.g. DB 0 or DX 0 with QTYP = DB or DX) This bit always occurs with the virtual S5 drive since all numbers of the DB and DX data blocks are addressed by S5REMOTE/S5REMOTF with each data transfer.
31H	Area too small or total of initial address (QANF/ZANF) and length (QLAE/ZLAE) too large (with all QTYP/ZTYP).
41H	Area does not exist or is illegal (with QTYP/ZTYP = AS, QA, IA, PY)
51H	Status word (address) faulty
61H	Dependent on CPU
71H	Interface does not exist
81H	Interface not ready
91H	Interface overloaded
A1H	Dependent on CPU
B1H	Job number illegal or frame size (SYNCHRON) illegal
C1H	Interface does not react, or interface does not react at correct time, or interface rejects job
D1H	Dependent on CPU
E1H	Dependent on CPU
F1H	Dependent on CPU

Error Status Bits of MS-DOS (Extended Error Code)

If you develop your own MS-DOS programs for using the virtual S5 drive, you can evaluate the status "Extended error code" of MS-DOS following the occurrence of an error. The MS-DOS function **59h** is available for this purpose, or the C library function **dosexterr ()** for C programs. If you wish to use these functions, please refer to the appropriate literature for calling and parameterizing them.

If you have called one of the above-mentioned functions, you obtain the pointer (address) to a data block with the following error information:

Pointer Û	Word 0	Extended Error Code	
	Word 1	Error class	Suggested action
	Word 2	Error locus	

The information in “Error class,” “Suggested action” and “Error locus” have no practical significance when accessing the virtual S5 drive, except with “Format error.” The following status bits may occur in the extended error code with MS-DOS file functions which address the virtual drive:

Table 10-11 Meaning of Extended Error Codes

Extended Error Code	Cause of error
2	The file was not found
3	The path was not found
5	Access rejected
18	No further files
80	Directory cannot be created
160	Format error (when writing) ¹⁾
174	DB/DX do not exist
195	Aborted by timeout
201	Aborted by reset
202	Aborted by SYNCHRON
205	Aborted by negative acknowledgment
254	CPU is not synchronous

1) Word address of the faulty format instruction in error class (High byte) and suggested action (Low byte)

Table 10-12 Error Messages of S5REMOTE/S5REMOTF

Error message	Remarks
DOS version must be 3.10 or higher	
Drive letter too high	Remedy: increase “lastdrive” in CONFIG.SYS
Drive already exists	Select other letter for virtual drive
Unknown S5 format in line x	Error in the format file
Format list overflow in line x	The format file contains formats for more than 4091 data words
Unknown option	
CPDHB driver not found	Remedy: load driver
CPU x not synchronous	Remedy: carry out restart on CPU and call DHB SYNCHRON
<cpu> must be 1.. 4 or ‘*’	
<db> must be 1.. 511 or ‘*’	
Illegal timeout	The value for the timeout (sec.) is not in the range from 1 to 3640.
*** CPU x: Error y	y is an error bit of the CPDHB driver, see Section 13.7.4.

Error Messages of S5REMOTE/S5REMOTF

10.6 Conversion of Preheader Data Using DVCONVRT

The DVCONVRT program is supplied together with the other system software of the CP 581.

You can use it to convert preheader data associated with a data block DB or DX, which you have copied in the file `?????ST.S5D` from the PG onto the CP 581 (as described in Section 10.3.2), into a format file. The files generated from the preheader data using DVCONVRT have the name extension **FMT**.

Preheader Data for the Data Blocks

When programming S5 blocks on the programming device, enter the name for an S5D file in which all blocks for a specific field of application are to be stored. Format information on the block data is stored as so-called "Preheader data" by the PG software for each data block DB and DX which you store in this file.

The following relationship exists between the names of the data blocks and the names of the preheader information:

- S5 data in DB nnn: Preheader data in DV nnn
- S5 data in DX mmm: Preheader data in DVX mmm

Procedure 1)

Proceed as follows:

1. Copy the S5D file whose preheader data you wish to convert into format files for the virtual S5 drive onto a 3.5" floppy disk in MS-DOS format on the programming device (for example the file `MYBLOCK.S5D`).
2. Use the MS-DOS command `COPY` to load the file from the floppy disk onto your CP 581, for example into directory `S5_FORM`.
3. Select the directory with the S5D file as the current directory, for example `S5_FORM`.
4. List all DV and DVX blocks of the file by means of the DVCONVRT program using the command

DVCONVRT *.S5D /L where * = name of the S5D file, e. g. `MYBLOCK`

5. Convert the required preheader data into a format file using the following commands:

Converting from DV block: **DVCONVRT *.S5D n**
where * = name of S5D file, e. g. `MYBLOCK`
n = number of DV block, e. g. 20

Converting from DVX block: **DVCONVRT *.S5D m X**
where * = name of S5D file, e. g. `MYBLOCK`
m = number of DVX block, e. g. 50

- 1) You can also use lower-case letters in the commands and file names instead of upper-caseletters.

Example

The preheader data for block DB 10 are to be converted:

The preheader data for DB 10 are stored in the S5D file MYBLOCK.S5D on a 3.5" floppy disk. Preheader data are additionally included in this file in the blocks DV 11, DVX 30 and DVX 31.

- Copy file from drive A: into directory S5_FORM of drive C:

```
COPY A:MYBLOCK C:\S5_FORM
```

- Select directory S5_FORM on drive C:

```
CD S5_FORM
```

- List all blocks with preheader information:

```
DVCONVRT MYBLOCK.S5D /L
```

- Convert block DV 10 into file MYBLOCK.FMT:

```
DVCONVRT MYBLOCK.S5D 10
```


Free Programming of the CP 581

This chapter is intended for CP 581 users who wish to use the functions of the CPDHB driver and the S5 data handling blocks for special applications.

If you wish to program your own CP 581 applications, you should be well experienced in programming S5 programmable controllers and have detailed knowledge on the assembler language of MS-DOS assemblers.

If you are acquainted with high-level languages, it may be sufficient to have a basic knowledge of the assembler language in addition to experience in STEP 5 programming in order to program an interface module which can be called for the driver functions using a high-level programming language.

Section 11.2 informs you of the steps necessary to solve your problem.

Section 11.3 provides you with an introduction to the use of the data handling blocks for parameterization and evaluation of bits for communication between the CPU and CP 581.

Section 11.4 informs you of the CPDHB driver, interaction between the CPU and CP 581 and the required programming of the driver functions.

Information on testing your application can be found in Section 11.5.

Chapter Overview

Section	Description	Page
11.1	Application	11-3
11.2	Procedure	11-3
11.2.1	Summary	11-3
11.2.2	Analysis of Task	11-4
11.3	Programming of DHB Calls	11-5
11.3.1	General Information	11-5
11.3.2	Available Data Handling Blocks	11-6
11.3.3	Parameters of Data Handling Blocks	11-7
11.3.4	ParameterDescription	11-7
11.3.5	Direct and Indirect Parameterization	11-12
11.3.6	Format and Meaning of the Status Word	11-16
11.3.7	SEND Block	11-20
11.3.8	RECEIVE Block	11-22
11.3.9	FETCH Block	11-24
11.3.10	CONTROL Block	11-25
11.3.11	RESET Block	11-26
11.3.12	SYNCHRON Block	11-27

Section	Description	Page
11.4	Programming the CP 581 User Program	11-28
11.4.1	CPDHB Driver	11-29
11.4.2	Transfer Control Block (TCB)	11-32
11.4.3	Summary of Driver Functions	11-37
11.4.4	Example of Call of CPDHB Driver	11-38
11.4.5	Data Transfer with Direct Jobs	11-39
11.4.6	Data Transfer Without Direct Jobs	11-52
11.4.7	Other Driver Functions	11-55
11.5	Testing the Application	11-58
11.5.1	Procedure	11-58
11.5.2	Testing the S5 Program	11-58
11.5.3	Testing the CP 581 Program	11-59
11.5.4	Representation of the S5 Data in the CP 581 Memory	11-59

11.1 Application

Free programming of the CP 581 enables you to solve special tasks which may be necessary for communication between CPUs and the CP 581, for example data transfer with higher-level computers or operator inputs for a CPU (possible modification of limits).

The CP 581 program enables communication with up to 4 CPUs. The CPUs must be plugged into the same programmable controller as the CP 581.

11.2 Procedure

11.2.1 Summary

Proceed in the order recommended in Table 11-1 to solve your special application.

Table 11-1 Recommended Procedure to Solve an Application Problem

Measure	on	See section
First make an exact analysis of what communication is to take place between the CP 581 and the CPU(s). When trying for the first time, only use communication with one CPU.	--	11.2.2
Note which data handling blocks you require to solve the communication task and how these are parameterized.	--	11.3
Program your STEP 5 program with calls of the required DHBs.	PG	--
Note for the CPDHB driver: - How is it called by an MS-DOS program? - What is the interaction with the DHB? - When must it be called? - With which DHB does it cooperate? - How must it be parameterized?	--	11.4
Program the MS-DOS program with which you wish to handle the communication.	CP 581	11.4
Make sure that the CPU with which you wish to exchange data and the CP 581 are inserted in your programmable controller and that the CPDHB driver is loaded in the memory (TSR).	CP 581	11.4
Test the correct execution of your STEP 5 program initially without your MS-DOS program but with the CPDHB driver (synchronization of CP 581 with CPU).	PG CPU CP 581	11.5
Now test your MS-DOS program and the interaction with the data handling blocks on the CPU.	PG CPU CP 581	11.5
Now extend your application - if necessary - for communication with several CPUs.	PG CPU CP 581	--

11.2.2 Analysis of Task

Before you read further in the next sections which functions of the data handling blocks and the CPDHB driver you require and how you should program your application, first answer the following questions with respect to the task:

- Is communication to take place with **one or several CPUs**?
(If communication is to take place with several CPUs, first answer the rest of the questions for one CPU and implement the communication for this one. When this takes place correctly, repeat the analysis for several CPUs.)
- In **which direction** are data to be transmitted?
 - From the CPU to the CP 581 (CPU **sends** data),
 - From the CP 581 to the CPU (CPU **receives** data)
 - or
 - in both directions (CPU **sends and receives** data)?
- How large is the data volume to be exchanged with one process (job)?

Once you have answered these questions, inform yourself first in Section 11.3 on the functions of the data handling blocks (DHB). Define which DHBs you require taking into consideration your replies to the questions, and program or modify your STEP 5 program. You can then draft and generate your MS-DOS program for communication in accordance with the DHB calls in your STEP 5 program and after studying Section 11.4.

Note

The system programs supplied for “Mass storage functions” and “Command interpreter” occupy the following job numbers for your application:-

Process data acquisition:	210 and 211
- Mass storage functions:	1 to 99, 201, 207
- Command interpreter:	209

The numbers > 200 are always reserved for system functions.

If you require job numbers for your applications (only with so-called “direct” jobs) and wish to use one of the system programs in addition to your application, you must not assign the corresponding job numbers.

11.3 Programming of DHB Calls

Detailed knowledge on the data handling blocks (DHB) is required in order to program data transfer functions with the CPDHB driver. The most important characteristics of the DHBs which you require for programming are explained in the following subsection. Information on the properties of DHBs which affect programming of the CPU, for example, the various sizes of the areas, can be obtained from the DHB descriptions for the various programmable controllers (/5/, /8/, /9/ Reference Literature).

11.3.1 General Information

Communication between the S5 CPU and the CP is via the so-called page area which is present on the S5 bus at address F400H and which occupies an address area of 1024 bytes. The S5 CPU accesses the dual-port RAM of the CPs via this address. Data transfer between the CP and the S5 CPU is via this dual-port RAM. A so-called vector register is required since all pages of all CPs are present in the same address area. The CPU writes the number of the desired page into the vector register before accessing the page area. The page number (also referred to as the interface number) must be set on the respective CP using a switch assembly or the system software. All CPs in a PLC must occupy different page numbers, otherwise there will be address conflicts.

The previously described sequence is completely handled by the data handling blocks. It is only necessary to define the interface number (SSNR) when programming a DHB. The DHB automatically selects the page

A data handling block can only exchange a limited quantity of data with the CP during a cycle. This quantity is referred to as a data frame. You can set the frame size in various steps from 16 to 512 byte using the DHB SYNCHRON (see Sections 11.3.4 and 11.3.12). If the quantity of data to be transmitted is larger than the selected frame size, several CPU cycles are required in order to transfer all the data.

11.3.2 Available Data Handling Blocks

- **SEND:**

The SEND block is able to transmit data areas from the CPU to the CP. The SEND block can transmit a maximum of one data frame.

- **SEND ALL:**

A special operating mode of the SEND block is SEND ALL. The CPDHB driver can use the SEND ALL mode to request the SEND block to transmit data areas from the CPU to the CP. The CPDHB driver defines the address of the CPU data.

- **RECEIVE:**

The RECEIVE block can transmit data areas from the CP to the CPU. Like the SEND block, a RECEIVE block can only transmit one data frame.

- **RECEIVE ALL:**

A special operating mode of the RECEIVE block is RECEIVE ALL. The CPDHB driver can use the RECEIVE ALL mode to request the RECEIVE block to transmit data areas from the CP to the CPU. The CPDHB driver defines the address of the CPU data.

- **FETCH:**

The FETCH block itself cannot transmit data. RECEIVE ALL is always required here for the actual data transfer. The FETCH block is only used to trigger the CP to supply and transmit the required data.

- **SYNCHRON, RESET, CONTROL:**

These data handling blocks are not directly involved in data transmission. They handle auxiliary functions which are described in more detail in the following sections.

11.3.3 Parameters of Data Handling Blocks

Table 11-2 Parameters of the Data Handling Blocks

Designation	Meaning
SSNR	Interface number (page number)
A-NR	Job number
ANZW	Status word
QTYP/ZTYP	Type of data source
DBNR	Data block number
QANF/ZANF	Relative initial address within the area
QLAE/ZLAE	Number of source data or destination data
PAFE	Parametererror
BLGR	Frame size

11.3.4 Parameter Description

The formal operands which must be assigned when using the data handling blocks have the following meanings:

SSNR - Interface Number

The parameter SSNR is used to define the logical number of the interface (page) to which the respective job refers.

Parameter type	Format	Assignment
Data (byte)	KY	KY= x,y x = 0 Directparameterization y = 0..255 Interface number (page number) x ≠ 0 Indirectparameterization y = 0..255 Data word number. The parameters SSNR, A-NR and ANZW are stored in the current DB starting with the specified data word.

**A-NR -
Job Number**

The jobs for an interface are differentiated by this number.

Parameter type	Format	Assignment
Data (byte)	KY	KY= x,y x Parameter x is irrelevant y = 0 ALL mode selected (not with FETCH) y = 1..223 Direct mode selected with the number of the job to be executed.

**ANZW -
Status Word**

Use this parameter to enter the address of a double word in which the processing status of a specific job is displayed. Section 11.3.6 explains the evaluation of the status word.

Parameter type	Format	Assignment
Address (word)	W	x = 0..255 Address of status word with direct parameterization Permissible range: FW 0 to FW 252 DW 0 to DW 254

Note

Note that the status word always occupies two words or four bytes. The data handling blocks cannot detect whether the assignments overlap.

QTYP/ZTYP - Type of Data Source or Data Destination

Assign these parameters with ASCII characters which define the type of data source (with SEND) or the data destination (with RECEIVE or FETCH).

Parameter type	Format	Assignment
Data (characters)	KS	<p>KS = QA, AS, RS, DB, DX, IA, FA, PY, TA, CA Direct parameterization: the data on the data source/destination directly affect the parameters QTYP/ZTYP, DBNR, QANF/ZANF, QLAE/ZLAE.</p> <p>KS = NN Without parameterization: no data on the data source or destination. The effect of the job is determined only by the job number.</p> <p>KS = XX Indirect parameterization: either the parameter set for the data source or that for the data destination is present in a data area specified by the parameters DBNR and QANF/ZANF.</p> <p>KS = RW Indirect parameterization: two parameter sets are always present in a data area specified by the parameters DBNR and QANF/ZANF: one set for the data source and then one set for the data destination.</p>

DBNR - Data Block Number

If you wish to assign DB, RW or XX to the parameters QTYP/ZTYP, you must define the number of the desired data block with this parameter.

Parameter type	Format	Assignment
Data (byte)	KY	<p>KY = 0, y for data blocks DB 1, y for data blocks DX y = 3 to 255 Number of the data block in which the data are present. DBNR is only relevant to data blocks (DB/DX). An exception is the area AS (absolute addresses) with the CPU 946/947 where the remaining addresses 2^{16} to 2^{19} are stored in DBNR.</p>

**QANF/ZANF -
Initial Address of
Data Block of
Source or
Destination**

With indirect parameterization - assignment of RW or XX to QTYP/ZTYP - enter the number of the DW here at which the parameter block commences. With direct parameterization, QANF/ZANF refers to the defined area.

Parameter type	Format	Assignment
Data (fixed point)	KF	<p>QANF is used to specify the first item of data to be transmitted within the source data area relative to the start of the area.</p> <p>ZANF is used analog to QANF to specify the destination data area.</p> <p>The permissible range of values depends on the CPU used. Please refer to the DHB description for your CPU.</p>

**QLAE/ZLAE -
Length of Data
Block of Source or
Destination**

In the case of direct parameterization, the length is understood as the number of bytes or words depending on the specification of the source or destination type.

Parameter type	Format	Assignment
Data (const.)	KF	<p>QLAE is used to define the length of the source data area, ZLAE the length of the destination data area.</p> <p>The permissible range of values depends on the CPU used. Please refer to the DHB description for your CPU.</p> <p>-1 -1 means "joker length"</p> <p>With RECEIVE, all the data delivered by the transmitter are accepted, or as many as permitted by the range limit.</p> <p>With SEND, data are transmitted until the range limit has been reached.</p>

BLGR - Frame Size

This parameter defines the maximum size of the data frame which can be transferred between the PLC and the CP in one DHB cycle. It is only relevant to the SYNCHRON block. The execution time for the data transfer mainly depends on the defined frame size. Using the DHB description for your CPU you must decide which execution times are possible for your special application. Note with small frame sizes, i.e. short execution times, that several CPU cycles may be necessary for data transfer depending on the quantity of data.

Parameter type	Format	Assignment
Data (byte)	KY=	0, y Frame size
		y = 0 64 bytes for S5-115U 256 bytes for S5-135U/155U
		y = 1 16 bytes
		y = 2 32 bytes
		y = 3 64 bytes
		y = 4 128 bytes
		y = 5 256 bytes
		y = 6 512 bytes
		y = 7..254 like y = 0
		y = 255 512 bytes

**PAFE -
Indication with
Parameter Error**

Enter a byte to be set when the block detects a parameter error. Evaluation of the parameter error is explained in Section 11.3.6 (page 11-19).

Parameter type	Format	Assignment
Address (byte)	BY	Possible areas QA, FA
		The size of the areas is CPU-specific.

11.3.5 Direct and Indirect Parameterization

Indirect Parameterization of SSNR, A-NR, ANZW and BLGR

The high byte of the parameter SSNR serves as a switch-over criterion for direct or indirect parameterization of the parameters SSNR, A-NR, ANZW and BLGR.

- High byte of SSNR = 0, means **direct parameterization**:
SSNR, A-NR, ANZW or BLGR are directly defined in the block called.
- High byte of SSNR \neq 0, means **indirect parameterization**:
SSNR, A-NR and ANZW or BLGR are stored in the opened data block starting at the data word specified in the low byte of SSNR.

SSNR and **A-NR** have the same data format (KY) in both parameterization modes. The formats are different in the status word **ANZW**. Whereas the address of the status word (for example FW 100) can be specified during direct parameterization, additional information on the area of the status word must be specified with indirect parameterization. This area is specified in ASCII code in the data word which precedes the status word:

FW for status word in flag area
DB for status word in data area

The ANZW address is present in data format KY in the following data word of the parameter area in the DB, and additionally the block number with DB (in the first byte of the KY format).

Examples of Indirect Parameterization

The parameters

SSNR,
A-NR and
ANZW

are to be addressed indirectly.

- **Example 1:** A flag address is specified for the status word.

STEP 5 commands for DHB call:

```

:
:
:C  DB 44  Opening of DB 44
:
:JU  FB 244  (FB 244 only with S5-115U)
NAME :SEND
SSNR  :KY 255,1  Code for indirect parameterization 1)
A-NR  :KY 0,0    Not relevant
ANZW  :FW 0      Not relevant
:
:
```

Parameters in data block DB 44:

DB44

```

0:          Any data
1:      KY 0,1:  DW 1: the interface number is '1.'
2:      KY 0,31: DW 2: the job has the number '31.'
3:      KS FW:  DW 3: the status word is in the flag area.
4:      KY 0,200: DW 4: the status word is displayed in the
5:      :          flag words FW 200 and FW 202.
```

- 1) The data area for parameterization begins with data word DW 1.

- **Example 2:** The status word is to be present in a data block.

STEP 5 commands for DHB call:

```

:
:
:C DB 24 Opening of DB 24
:
:JU FB 244 (FB 244 only with S5-115U)
NAME :SEND
SSNR :KY 255,1 Code for indirect parameterization 1)
ANR :KY 0,0 Not relevant
ANZW :FW 0 Not relevant

```

Parameters in data block DB 24:

DB24

```

0: DW 0: Any data
1: KY 0,1 DW 1: the interface number is '1.'
2: KY 0,31 DW 2: the job has the number '31.'
3: KS DB DW 3: the status word is in the data block.
4: KY 222,10 DW 4: the address of the status word is:
5: : "DB 222, DW 10 and DW 11."

```

ANZW in DB 222:

DB222

```

10: : DW 10: status word
11: : DW 11: length word

```

- 1) The data area for parameterization begins with data word DW 1.

- **Example 3:**

The parameters
SSNR and
BLGR

are to be addressed indirectly while the DHB SYNCHRON is being called.

STEP 5 commands for DHB call:

```

:
:
:C DB 49 Opening of DB 49
:JU FB 249 (FB 249 only with S5-115U)
NAME :SYNCHRON
SSNR :KY 255,100 Code for indirect parameterization 1)
BLGR :KY 0,0 Not relevant

```

Parameters in data block DB 49:

DB49

```

100: KY 0,10 DW 100: the interface number is '10.'
101: KY 0,6 DW 101: the frame size is set to 512
bytes.

```

- 1) The data area for parameterization begins with data word DW 100.

**Indirect
Parameterization
of QTYP/ZTYP,
DBNR,
QANF/ZANF and
QLAE/ZLAE**

When parameterizing QTYP or ZTYP with RW or XX, the data on the data source or destination are obtained from a data area. The starting address of this data area is defined by the value of the parameter QANF. When parameterizing indirectly with XX, the following data must be entered in the data block defined by the formal operand “DBNR:”

Table 11-3 Indirect Addressing of Transmission Parameters

Address in data block		Parameter type	Assignment	Explanation
QANF	+ 0	KS	QA, AS, RS, DB, DX, IA, FA, PY, TA, CA	Definition of type of source or destination (QTYP/ZTYP)
	+ 1	KY	3 to 255	Number of DB with source or destination type DB (DBNR) (High byte = 0)
	+ 2	KF	Range of values depends on area and CPU	Initial address of source or destination data area (QANF/ZANF)
	+ 3	KF	Range of values depends on area and CPU	Length of source or destination data area (QLAE/ZLAE)

In the case of indirect parameterization with RW, the data in the data block with the number “DBNR” must have the following contents:

Table 11-4 Parameter Values with Indirect Addressing with RW

Address in data block		Parameter type	Assignment	Explanation
Parameters for source data area				
QANF	+ 0	KS	QA, AS, RS, DB, DX, IA, FA, PY, TA, CA	Specification of source type (QTYP)
	+ 1	KY	3 to 255	Number of DB with source type DB (DBNR) (High byte = 0)
	+ 2	KF	Range of values depends on area and CPU	Initial address of source data area (QANF)
	+ 3	KF	Range of values depends on area and CPU	Length of source data area (QLAE/ZLAE)
Parameters for destination data area				
	+ 4	KS	QA, AS, RS, DB, DX, IA, FA, PY, TA, CA	Specification of destination type (ZTYP)
	+ 5	KY	3 to 255	Number of DB with destination type DB (DBNR) (High byte = 0)
	+ 6	KF	Range of values depends on area and CPU	Initial address of destination data area (ZANF)
	+ 7	KF	Range of values depends on area and CPU	Length of destination data area (ZLAE)

11.3.6 Format and Meaning of the Status Word

Information on the status of job handling is stored in the status word. You define the address of the status word during parameterization. The information can then be read and evaluated from there. The status word is part of a double word which is addressed by the parameter ANZW. The second part of the double word is the so-called "length word."

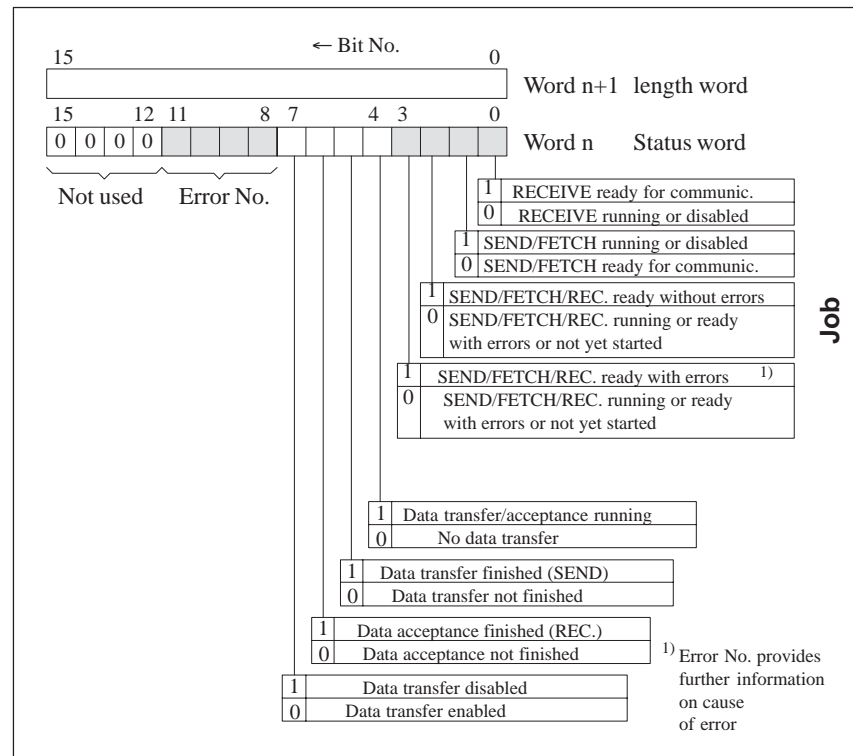


Figure 11-1 Format of DHB Status Word

Note

Assign an individual status word to each job used. If you must send two DHB calls in succession for a job (SEND - SEND ALL, FETCH/RECEIVE - RECEIVE ALL), individual status words must always be provided for each call since these are handled separately by the DHBs specified.

If bits 1 and 2 are set **simultaneously** in the status word, you cannot send jobs to the CP 581 for the respective job number. Note this response when starting your S5 program (see also Section 13.1.8).

Meaning of Status Bits (Bit Nos. 0 to 7) The status bits of the status word provide information on execution of a DHB call. They also serve as input information for the DHB itself and then influence its execution.

Table 11-5 Meaning of the Bits in the Status Word

Bit No.	Set	Delete/ overwrite	DHB evaluation	User evaluation
0	DHB CP application signals "data ready"	DHB CPDHB driver signals "start of communication"	RECEIVE With bit No. 0= 1 communication with the CP is started if RLO = 1	Scan whether data present for RECEIVE.
1	DHB CPDHB driver signals "start of communication"	DHB CP application signals "job processed"	SEND/FETCH With Bit No. 1 = 0, communication with the CP is started if RLO = 1	Scan whether job being executed
2	DHB The CP application signals "job finished without error."	DHB If job triggered again.	No	Scan whether job finished without error.
3	DHB The CP application signals "job finished with error."	DHB If job triggered again.	No	Scan whether job finished with error; more information on the cause of the error is contained in bit nos. 8 to 11 (error No.)
4	SEND/RECEIVE If data transfer for a job has been started.	SEND/RECEIVE If data transfer for a job has been completed.	No	Scan whether data are currently being transmitted.
5	SEND If data transfer to the CP has been terminated.	SEND If data transfer for a new job has commenced.	No	Scan whether the data block of a new job has already been transferred to the PLC and when a new data record can be provided for a current job.
6	RECEIVE If the transfer of data has been terminated for a job.	RECEIVE If the data transfer for a new job has been commenced.	No	Scan whether the data block of the current job has already been transmitted to the CPU.
7	User Disable access to data area.	User Enable access to data area.	SEND/RECEIVE No data transfer takes place with bit No. 7 = 1, an error message is output to the CP.	No

Meaning of Error Numbers

The error numbers are only valid if bit No. 3 of the status word is set at the same time. They influence these error numbers by means of their CP program.

Table 11-6 Meaning of Error Numbers

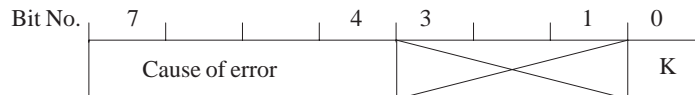
Error No.	Meaning
0000	No error
0001..0101 1..5	DHB error(also referred to as PLC or CPU error), error number as in status byte "Parameter error" (PAFE).
0110..1111 6..F	CP error You can use these error numbers for your application.

Length Word

The data handling blocks SEND and RECEIVE store in the length word the number of data (number of bytes) already transferred with the respective job. With the ALL functions, the blocks SEND and RECEIVE enter the job number for which they were active in the current cycle in the Low byte. The job number "0" (idling) means that no job was processed. The following table indicates how the length word is influenced.

Write	Delete/ overwrite	Evaluate
SEND/RECEIVE During data transfer: number of bytes transmitted.	SEND/FETCH/ RECEIVE By overwriting with next job.	User If bit No. 2, 5 or 6 is set, the current source or destination length is present in the length word. If bit No. 3 is set, the length word specifies how much data has been transmitted up to the occurrence of the error.

**Status Byte
"Parameter
Assignment Error
(PAFE)"**



K = Common bit:

- 0: no errors
- 1: parameter error, more details in bits 4 to 7

Table 11-7 Meaning of Bits in Status Byte PAFE

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format
21H	DB or DX data block not present or illegal (e. g. DB 0 or DX 0 with QTYP = DB or DX)
31H	Area too small or total of initial address (QANF/ZANF) and length (QLAE/ZLAE) too large (with all QTYP/ZTYP)
41H	Area does not exist or is illegal (with QTYP/ZTYP = AS, QA, IA, PY)
51H	Status word (address) faulty
61H	Dependent on CPU
71H	Interface does not exist
81H	Interface not ready
91H	Interface overloaded
A1H	Dependent on CPU
B1H	Job number illegal or frame size (SYNCHRON) illegal
C1H	Interface does not react, or interface does not react at correct time, or interface rejects job.
D1H	Dependent on CPU
E1H	Dependent on CPU
F1H	Dependent on CPU

11.3.7 SEND Block

The SEND block is used to transmit data areas from the CPU to the CP. There are two modes:

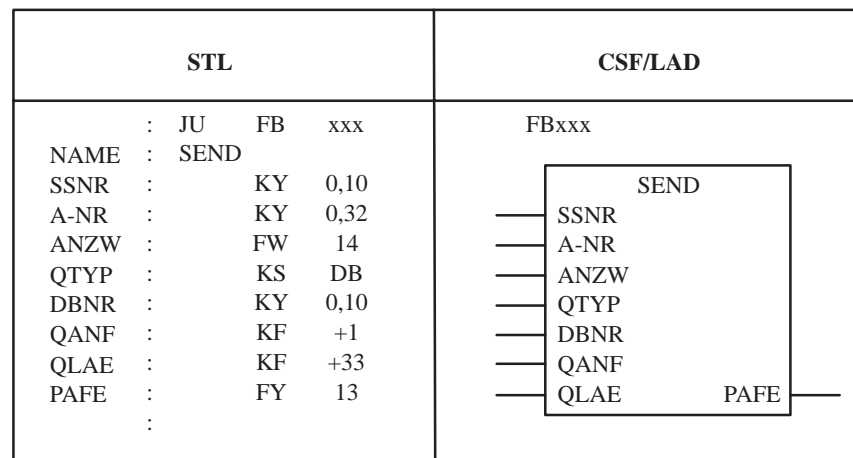
- SEND ALL:

The function block serves as a substitute for a direct memory read operation of the CP to the CPU.

- SEND DIRECT:

Data are transmitted from the CPU to the CP with a specific job number.

Example of a call of the SEND block



Description of the SEND ALL Mode

The block requires the following parameters for this function:

- SSNR - Interface number,
- A-NR - Job number (preset to "0"),
- ANZW - Specification of status word,
- PAFE - Specification of error byte.

All other parameters are irrelevant with this job. A value must nevertheless be entered for the irrelevant parameters but it is not evaluated by the data handling blocks.

The following parameters are transferred by the CPDHB driver to the SEND block during communication:

- Address of status word of the direct job responsible for triggering,
- Specification of data type,
- Number of data,
- Initial address of data area.

The following bits are evaluated or modified in the status word of the associated job:

- Bit No. 7: data transfer disabled,
- Bit No. 5: data transfer finished,
- Bit No. 4: data transfer running.

The number of data to be transmitted for the respective job is indicated by the block in the length word which follows the status word of the associated direct job.

The SEND block must be called at least once per interface in the control program in operating mode “ALL” if:

- The CP 581 can automatically request data from the CPU, for example, if you use the CPRECORD program.
- A job is triggered by SEND DIRECT, but the application only requests data from the CPU with the assistance of background communication.
- The quantity of data to be transferred to the CP using SEND DIRECT is larger than the frame size set.

Note

With the S5-135U and S5-155U programmable controllers you can use a special DHB SEND-A instead of the call SEND ALL mode. With this DHB you need not note the irrelevant parameters. See Section 13.1.1 for the FB numbers of these DHBs.

Description of the SEND DIRECT Mode

The DIRECT mode operates with the following parameters:

- SSNR - Interface number,
- A-NR - Job number,
- ANZW - Specification of status word,
- QTYP - Source type,
- DBNR - Number of data block,
- QANF - Initial address of source,
- QLAE - Number of source data,
- PAFE - Specification of error byte.

The DIRECT mode is generally called up in the cyclic part of the CPU program. The block can also be called during interrupt or alarm processing, but the status word is then not updated cyclically. This function must then be handled by the CONTROL block.

Two conditions must be fulfilled for data transfer or activation of the SEND job:

- RLO = 1 was transferred to the function block,
- Bit No. 1 = 0 in the status word.

Only the status word is updated if RLO = 0 (idling) is transferred.

11.3.8 RECEIVE Block

The RECEIVE block is used to receive data from the CP by the CPU. There are two modes:

- RECEIVE ALL:

The function block serves as a substitute for a direct memory read option of the CP to the CPU.

- RECEIVE DIRECT:

Data are transmitted from the CP to the CPU with a specific job number.

Example of a call of the RECEIVE block:

STL	CSF/LAD
: JU FB xxx NAME : RECEIVE SSNR : KY 0,10 A-NR : KY 0,32 ANZW : FW 14 ZTYP : KS DB DBNR : KY 0,10 ZANF : KF +1 ZLAE : KF +33 PAFE : FY 13 : :	FBxxx <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center; margin: 0;">RECEIVE</p> <p>— SSNR</p> <p>— A-NR</p> <p>— ANZW</p> <p>— ZTYP</p> <p>— DBNR</p> <p>— ZANF</p> <p>— ZLAE PAFE —</p> </div>

Description of the RECEIVE ALL Mode

The block requires the following parameters for this function:

- SSNR - Interface number,
- A-NR - Job number (preset to "0"),
- ANZW - Specification of status word,
- PAFE - Specification of error byte.

All other parameters are irrelevant with this job. A value must nevertheless be entered for the irrelevant parameters but it is not evaluated by the data handling blocks.

The following parameters are transferred by the CPDHB driver to the RECEIVE ALL block during communication:

- Address of status word of the direct job responsible for triggering,
- Specification of data type,
- Number of data,
- Initial address of data area.

The following bits are evaluated or modified in the status word of the associated job:

- Bit No. 7: data transfer disabled,
- Bit No. 6: data receive finished,
- Bit No. 4: data receive running.

The number of data to be transmitted for the respective job is indicated by the block in the length word which follows the status word of the associated direct job.

The RECEIVE block must be called at least once per interface in the control program in operating mode “ALL” if:

- The CP 581 can automatically send data to the CPU,
- A job is triggered by FETCH, and the application transmits the data to the CPU with the assistance of background communication,
- A job is triggered by RECEIVE DIRECT, and the application transmits the data to the CPU with the assistance of background communication,
- The quantity of data to be sent by the CP using RECEIVE DIRECT is larger than the frame size set.

Note

With the S5-135U and S5-155U programmable controllers you can use a special DHB RECEIVE-A instead of the RECEIVE ALL mode. With this DHB you need not note the irrelevant parameters. See Section 13.1.1 for the FB numbers of these DHBs.

**Description of the
RECEIVE DIRECT
Mode**

The DIRECT mode operates with the following parameters:

- SSNR - Interface number,
- A-NR - Job number,
- ANZW - Specification of status word,
- ZTYP - Destination type,
- DBNR - Number of data block,
- ZANF - Initial address of destination,
- ZLAE - Number of destination data,
- PAFE - Specification of error byte.

The DIRECT mode is generally called up in the cyclic part of the CPU program. The block can also be called during interrupt or alarm processing, but the status word is then not updated cyclically. This function must then be handled by the CONTROL block.

Two conditions must be fulfilled for data receive or activation of the RECEIVE job:

- RLO = 1 was transferred to the function block,
- Bit No. 0 = 1 in the status word.

Only the status word is updated if RLO = 0 (idling) is transferred.

11.3.9 FETCH Block

The FETCH block is used like the RECEIVE block to transfer data from the CP to the CPU. The FETCH block does not have an ALL mode.

STL	CSF/LAD
: JU FB xxx NAME : FETCH SSNR : KY 0,10 A-NR : KY 0,32 ANZW : FW 14 ZTYP : KS DB DBNR : KY 0,10 ZANF : KF +1 ZLAE : KF +33 PAFE : FY 13 :	FBxxx <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center; margin: 0;">FETCH</p> <p>SSNR</p> <p>A-NR</p> <p>ANZW</p> <p>ZTYP</p> <p>DBNR</p> <p>ZANF</p> <p>ZLAE</p> <p style="text-align: right;">PAFE</p> </div>

Description of the FETCH Function

All parameters must be assigned when calling "FETCH." The destination parameters (ANZW, ZTYP, DBNR, ZANF, ZLAE) are transferred to the CPU during the acknowledgment. As soon as the application has collected the required data at the CP end, they are transferred to the CPU using a RECEIVE ALL. The FETCH block itself does not transmit or accept any data.

The FETCH job is activated if the following conditions are satisfied:

- RLO = 1 was transferred to the function block,
- Bit No. 1 = 0 in the status word.

Note

Note that bit No. 1 is used in the status word for SEND and FETCH.

The FETCH block can be called from the cyclic, time-controlled or interrupt-controlled program section. The status word is updated by the FETCH or CONTROL block.

11.3.10 CONTROL Block

The CONTROL block updates the status word for a particular job or specifies which job is currently being processed.

STL				CSF/LAD	
	:	JU	FB	xxx	FBxxx
NAME	:	CONTROL			
SSNR	:		KY	0,10	
A-NR	:		KY	0,101	
ANZW	:		FW	20	
PAFE	:		FY	24	
	:				

Description of the CONTROL Function

The following parameters are required for this function:

- SSNR - Interface number,
- A-NR - Number of job to be monitored,
- ANZW - Specification of status word to be updated,
- PAFE - Specification of error byte.

Assignment of parameter A-NR with 0:

The number of the last job processed is transmitted to the Low byte of the status word. The CPDHB driver accepts the associated job number into job line 0 with each communication.

Processing of the block does not depend on the RLO. The CONTROL block should be called in the cyclic part of the CPU program, however.

11.3.11 RESET Block

The RESET block deletes a job which is running via the specified interface. There are two modes of the RESET block:

- **RESET ALL:**

When assigning 0 to the job number, all jobs of the CPDHB driver are deleted.

- **RESET DIRECT:**

If the job number is $\neq 0$, only the specified job of the interface is deleted.

STL	CSF/LAD
<pre> : JU FB xxx NAME : RESET SSNR : KY 0,10 A-NR : KY 0,101 ANZW : FW 20 PAFE : FY 24 : </pre>	<pre> FBxxx [RESET] SSNR A-NR PAFE </pre>

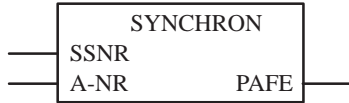
The block requires the following parameters:

- SSNR - Interface number,
- A-NR - Number of job to be deleted,
- PAFE - Specification of error byte.

The RESET block operates depending on the RLO and can be called from cyclic, time-controlled or alarm-controlled program sections.

11.3.12 SYNCHRON Block

The SYNCHRON block initializes the interface on the CP 581 for communication with the data handling blocks when restarting the programmable controller. The data handling blocks can only operate correctly following synchronization.

STL	CSF/LAD
: JU FB xxx NAME : SYNCHRON SSNR : KY 0,1 BLGR : KY 0,5 PAFE : FY 20 : :	FBxxx 

The following parameters must be assigned:

- SSNR - Interface number,
- BLGR - Frame size,
- PAFE - Specification of error byte.

The frame size defines how many data (bytes) can be transmitted during one cycle of SEND or RECEIVE. Since larger data quantities are divided into a corresponding number of individual frames, the set frame size largely determines the transmission time in the case of large quantities of data.

The SYNCHRON block must be called in one of the restart OBs (20, 21, 22).

11.4 Programming the CP 581 User Program

The CP 581 user program must handle data transfer together with the DHB calls programmed at the CPU end such that the desired communication is achieved. The user program uses the CPDHB driver for this purpose.

This section provides information for programming your CP 581 program:

- Fundamental information on the CPDHB driver.
- When you must call the driver for which function (interaction with DHB calls on the CPU).
- How you must parameterize the driver for the individual functions.

Note

Calling and parameterizing the CPDHB driver are described for assembler programming. If you wish to generate your user program in a high programming language, you must program an interface module for this language in assembler via which you call the CPDHB driver.

11.4.1 CPDHB Driver

The CPDHB driver handles the data transfer between S5 CPUs and MS-DOS programs on the CP 581. The purpose and structure of the data to be transmitted are of no significance to the driver. This is only concerned with the handling of the communication protocol using the data handling blocks.

Use of the data handling blocks must be **matched** between the **S5 application** and the **MS-DOS program**: the driver calls required to handle communication must be provided for each call of a data handling block in the S5 program (see Section 11.4.6).

Installation and Calling

The CPDHB driver is installed with MS-DOS as a TSR program (terminate and stay resident). The driver program remains resident in the memory following the installation, and MS-DOS returns to the command line. In this manner MS-DOS enables several programs to be loaded simultaneously in the memory. The driver program and the application program are therefore present as separate MS-DOS programs in the CP 581 memory. The CPDHB driver functions are called using a software interrupt. The interrupt number is set to 66H. Figure 11-2 shows you a simplified division of the CP 581 memory and the calling mechanism of the driver.

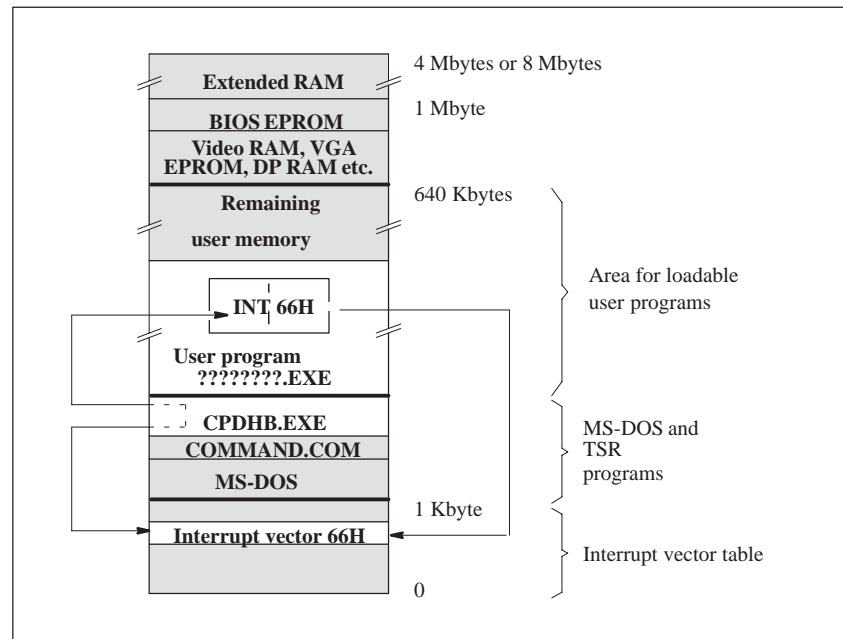


Figure 11-2 Memory Assignment of CP 581 and Calling of CPDHB Driver

Parameterizing the CPDHB Driver

If you call the CPDHB driver in your program in order to execute a specific function, you must specify the job exactly using parameters and provide the driver with means to handle the job depending on the function. Figure 11-3 provides an overview.

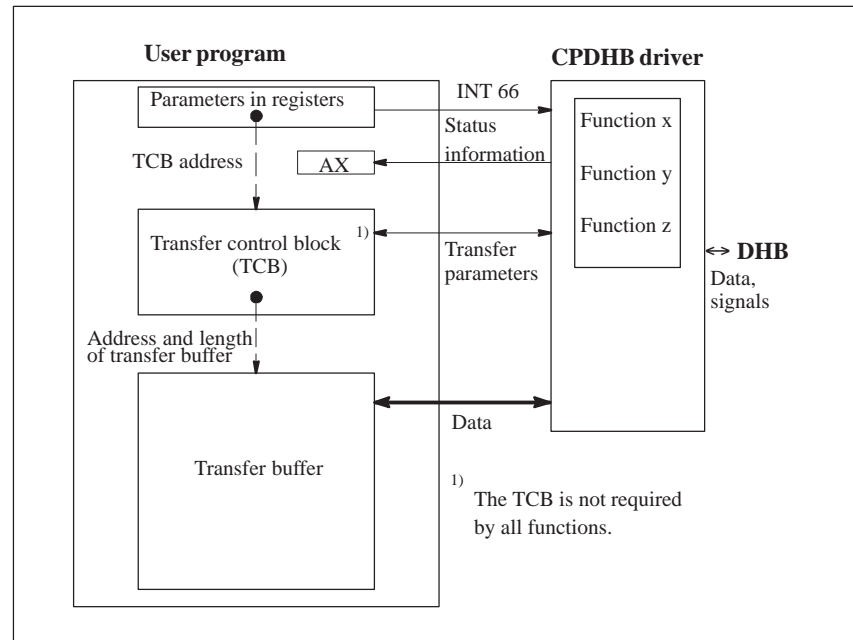


Figure 11-3 Parameters and Resources which must be provided by the User Program for Driver Calls

- **Direct parameters and condition codes:**

The application of direct parameters to the CPDHB driver functions and the return of condition codes is via the CPU registers of the CP 581.

- **Indirect parameters and resources:**

You must provide indirect parameters (for example, address and length of transfer buffer) and certain resources in a transfer control block (TCB, see Section 11.4.2). An important aid is the transfer buffer which you can provide in your program or somewhere on a vacant memory area of the CP 581.

Register set:

Figure 11-4 shows the 8086 register set with the registers provided for parameter transfer:

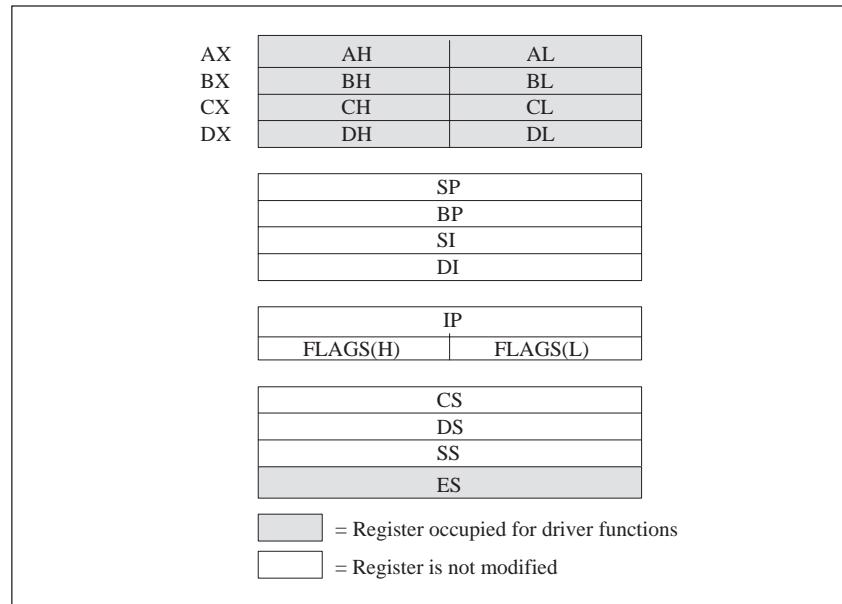


Figure 11-4 Register Set for Driver Call

Call:

Register AX = number of desired function (see Section 11.4.3).

The registers BX, CX, DX and ES are used specific to the function.
(You must store the offset in BX and the segment address of the TCB in ES for all transfer jobs.)

Result:

Condition codes for a job are transferred by the CPDHB driver in register AX when the called program is continued.

The unmentioned registers are not evaluated by the driver and are not modified.

11.4.2 Transfer Control Block (TCB)

If you wish to send a data transfer job to the CPDHB driver in your program, you must provide a transfer control block (TCB) (see following figure) in the program. All data relevant to the transfer are stored in the TCB. The address of the TCB is transferred as a parameter from the program to the driver when a transfer function is called. The TCB is managed by the driver until the transfer is terminated.

Note

You must preset the system area of the transfer control block to zero prior to the first call.

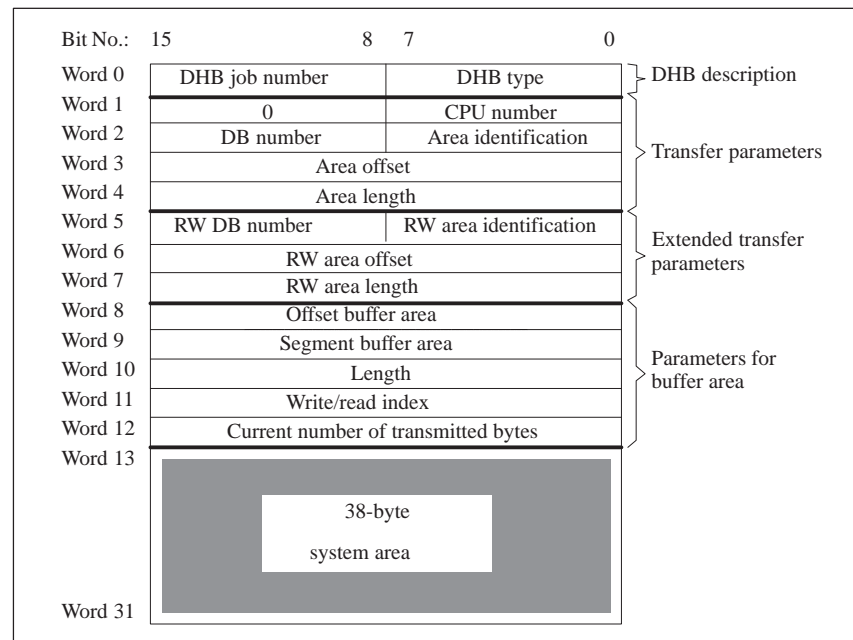


Figure 11-5 Structure of Transfer Control Block (TCB)

The following subsections describe the meanings of the individual TCB components.

DHB Description

The DHB description determines what data handling blocks are used for communication and also the mode. The DHB description also includes the job number via which the data handling block handles a job.

DHB job number:

The DHB job number corresponds to the A-NR when parameterizing the DHB. The meaningful range for the job number is 1 to 223. A number outside this range is rejected. You must only use the numbers 100 to 199 for the free programming. The other numbers are reserved for system programs.

DHB type:

Use the DHB type to specify which DHB is to be addressed and in which mode data transfer is to take place.

Table 11-8 shows you which codes are permissible for the DHB type byte and what the meaning of these codes is:

Table 11-8 Permissible Codes of DHB Type

Permissible DHB type coding	Meaning	
01H	SEND DIRECT	Only data are transmitted; transmission parameters are only transferred if more than one frame must be transmitted (i. e. if an additional call SEND ALL is required).
41H	SEND DIRECT/FETCH	Only transmission parameters are transferred; data must be transmitted with an additional call SEND ALL or RECEIVE ALL.
02H	RECEIVE DIRECT	

Transmission Parameters

The transmission parameters comprise the CPU number, area identification, data block number, area offset and area length. The S5 data to be transmitted are addressed using the transmission parameters.

The **CPU number** must always be specified by the **CP 581 program**. The other transmission parameters are either specified by the S5 program depending on the driver function or determined by the CP 581 program.

CPU number:

The CPU number defines the CPU in the programmable controller with which communication is to take place. Since the CPUs have a fixed assignment to the pages (see Chapter 6), the page number is also simultaneously defined by the CPU number in the TCB.

Bit No.:	7	4	3	0
	0		CPU No.	

Byte format with CPU number:

CPU1: CPU No. = 0 0 0 1
 CPU2: CPU No. = 0 0 1 0
 CPU3: CPU No. = 0 1 0 0
 CPU4: CPU No. = 1 0 0 0

Area identification and DB number:

The **area identification** defines the S5 area to be transmitted. One byte is reserved in the TCB for the area identification. The area identification corresponds to the parameters QTYP/ZTYP with the data handling blocks (see Section 11.3).

The area identification is entered by the CPDHB driver when the job arrives in the case of the transfer functions which operate with the direct jobs. This only takes place, however, if the DHB has recognized that subsequent blocks are necessary or if it is defined in the DHB type that only parameters are transferred.

You must enter the area identification in the TCB in the case of the transfer functions which only use the DHB function "ALL" since the address of the S5 data can be defined here by the CP 581 program.

The **DB number** corresponds to the parameter DBNR with the data handling blocks (see Section 11.3). With the S5 data area DB or DX, the number of the data block to be transferred is entered here (the entry is as with the area identification). The entry for the DB number has no significance with all other S5 areas.

Table 11-9 explains which area identifications are permissible, what meaning they have and how they must be coded for the TCB.

Table 11-9 Assignment of QTYP/ZTYP to the Area Identifications

QTYP/ZTYP	S5 area	Area identification
QA	Process output image	04H
AS	Absolute memory addresses ¹⁾	09H
RS	System data area	08H
DB	Data blocks DB	01H
DX	Extended data blocks DX	0AH
IA	Process input image	03H
FY	F flag area	02H
PY	I/O modules	05H
TA	Timer cells	07H
CA	Counter cells	06H

1) With the CPU 946/947 the parameter DBNR is used with absolute memory addresses (AS) to specify the addresses 2^{16} to 2^{19} .

Area offset:

The area offset is used to specify the starting address within the selected S5 area. The area offset is interpreted differently depending on the S5 data area since these areas are organized differently (see Table 11-10).

Table 11-10 Meaning of Area Offset and Organization of S5 Data Areas

Area (QTYP/ZTYP)	Meaning of area offset ¹⁾	Organization
QA	Output byte number	Byte-oriented
AS	Absolute address	Word-oriented
RS	RS word number	Word-oriented
DB	Data word number	Word-oriented
DX	Data word number	Word-oriented
IA	Input byte number	Byte-oriented
FY	Flag byte number	Byte-oriented
PY	I/O byte number	Byte-oriented
TA	Timer cell number	Word-oriented
CA	Counter cell number	Word-oriented

1) Refer to the description of your programmable controller to see which ranges are permissible for the various offsets.

Area length:

The area length defines the number of elements to be transmitted. As with the area offset, it is also necessary to consider the different organization of the S5 data areas (see Table 11-10). You must take into consideration the value of the defined offset for the permissible number range.

**Extended
Transmission
Parameters**

The extended transmission parameters are only significant with the DHB parameter type RW (see Section 11.3 and description of data handling blocks). The CPDHB driver stores the additional parameters for this parameter type in the TCB area "Extended parameters."

The parameter type RW can be used for SEND and FETCH, but the following differentiation must be observed:

- Applicable to SEND:
 - Transmission parameters:
 - source parameters
 - Extended transmission parameters:
 - destination parameters
- Applicable to FETCH:
 - Transmission parameters:
 - destination parameters
 - Extended transmission parameters:
 - source parameters

Parameters for the Buffer Area

You must provide a sufficiently large memory area in your program or in the RAM of the CP 581 for the useful data to be transmitted.

Offset/segment buffer area:

Use the parameters “**Offset buffer area**” and “**Segment buffer area**” to define the **address** of the memory area (far pointer) into which the data are to be written or from which they are to be read.

Length:

Use the parameter “Length” to define the size of the buffer. The value is entered as “Number of bytes” and can be up to FFFFH.

With the size of the buffer you determine the length of the data area to be transferred with the CP 581: the CPDHB driver uses it to monitor the area limits. Data outside the defined area are neither read nor written by the driver.

Write/read index:

With the functions “Send/receive useful data,” the CPDHB modifies a byte-oriented write/read index for transferring data. When transferring data into or out of the buffer by the driver, the current buffer area is always addressed via the write/read index relative to the initial address of the buffer. The write/read index has no effect with the functions “Write/receive S5 data area.”

Current number of transmitted bytes:

The driver enters the current number of transmitted bytes into the TCB during handling of a transfer job. This value is accepted by the DHB into status word (length word) 2.

11.4.3 Summary of Driver Functions

The CPDHB driver differentiates between two types of transfer jobs:

- Data transfer **with** direct jobs
and
- Data transfer **without** direct jobs.

Data Transfer with Direct Jobs

If the initiative for data transmission is to be with the S5 program in your application, you must use the direct jobs (SEND DIRECT, RECEIVE DIRECT and FETCH).

Data Transfer without Direct Jobs

If you wish to trigger data transmission from the CP 581, call a driver function “Data transfer without direct job.” In this case, one SEND ALL or RECEIVE ALL data handling block is sufficient in the CPU cycle depending on the transmission direction. Data transmission is then carried out when the corresponding driver function is called without a direct influence of the S5 program.

The following table provides you with a summary of all driver functions:

Table 11-11 Summary of Driver Functions

Function	Function No.	With/without direct job
Log on reception of job	11H	With
Log off reception of job	12H	With
Set job status	13H	With
Received data used	14H	With
Send data used	15H	With
Scan TCB status	16H	With
Global scan of DHB job reception	17H	With
Delete background job	18H	With
Read S5 data area	01H	Without
Write S5 data area	02H	Without
Scan driver status	30H	--

11.4.4 Example of Call of CPDHB Driver

Call a driver function from an assembler program:

```

                                TITLE      BSP1
                                .
                                .
;----- Declare Transfer Control Block:
                                .
                                .
tcb                                STRUCT      ; TCB structure
dhb_type                          DB          ?      ; DHB type (SEND, RECEIVE/FETCH)
dhb_job_no                        DB          ?      ; DHB job number
cpu_no                             DW          ?      ; CPU No.
area_ident                        DB          ?      ; Area identification
db_no                              DB          ?      ; DB number
area_offset                       DW          ?      ; Area offset
area_length                       DW          ?      ; Area length
area_ident_rw                     DB          ?      ; RW area identification
db_no_rw                          DB          ?      ; RW DB No.
area_offset_rw                    DW          ?      ; RW area offset
area_length_rw                    DW          ?      ; RW area length
offset_buffer                     DW          ?      ; Offset buffer area
seg_buffer                        DW          ?      ; Segment buffer area
sizeof_byte                       DW          ?      ; Length of buffer area
sl_index                          DW          ?      ; Write/read index
size_transfer_byte                DW          ?      ; Current number of transmitted bytes
system                            DB          38      ; System area 38 bytes
tcb                                ENDS

                                .
                                .
.DATA
                                .
;----- Variable:
                                .
stcb                               tcb    <0>      ; TCB variable
buffer                            DW    2000      ; buffer area 2000 words
                                .
.CODE
START
                                .
                                .
;----- Assign relevant values to TCB:
                                .
                                .
                                mov    stcb.cpu_no,01h      ; CPU No. = 1
                                mov    stcb.area_ident,01h   ; Area identification = 1 ( DB )
                                mov    stcb.area_offset,1d   ; Area offset = 1
                                mov    stcb.area_length,1000d ; Area length = 1000
                                mov    stcb.db_no,100d       ; DB No. = 100
                                mov    ax,SIZE buffer
                                mov    stcb.sizeof_byte,ax   ; Length of buffer area
                                mov    ax,OFFSET buffer
                                mov    stcb.offset_buffer,ax ; Offset buffer area
                                mov    ax,SEG buffer
                                mov    stcb.seg_buffer,ax    ; Segment buffer area
                                .
                                .
;----- Call function "Read S5 area:"
                                .
                                .
                                mov    ax,01h      ; Function number = 01H
                                mov    bx,OFFSET stcb ; Offset TCB
                                mov    es,SEG stcb  ; Segment TCB
                                mov    cx,100d     ; Timeout parameter (100 timer ticks)

                                int    66h      ; Driver call

                                cmp    ax,0h      ; Function terminated without errors
                                jne    error

                                .
                                .
error:                            ; Error handling
                                .
                                .
CODE                                ENDS
                                END    START

```

11.4.5 Data Transfer with Direct Jobs

This section explains the data transfer sequence with direct jobs and parameterization of the CPDHB driver calls required.

In the case of direct jobs, data transmission is initiated by the S5 program from the CPU. The data handling blocks permit the following 3 types of direct jobs:

- SEND DIRECT,
- FETCH,
- RECEIVE DIRECT.

The jobs listed are only accepted by the CPDHB driver if their reception was previously **registered** with the driver.

Note

When transferring a TCB to the driver it is not possible to differentiate between SEND and FETCH since the status word (ANZW) only provides one bit in the job status for both types of DHB.

Direct Job Sequence

The two following figures show you the basic sequence for data transfer with **SEND DIRECT**:

Figure 11-6: SEND DIRECT with **separate** data transfer,

Figure 11-7: SEND DIRECT with **direct** data transfer.

In both cases the CP 581 program initially registers a TCB for SEND/FETCH with the CPDHB driver and then cyclically checks the status of the TCB. The useful data are transmitted once a SEND job has arrived. This depends on the type of DHB which you have specified in the TCB:

either this has already taken place when the SEND DIRECT arrives (direct data transfer, Figure 11-7), or the data are explicitly transmitted with their own driver call (Figure 11-6).

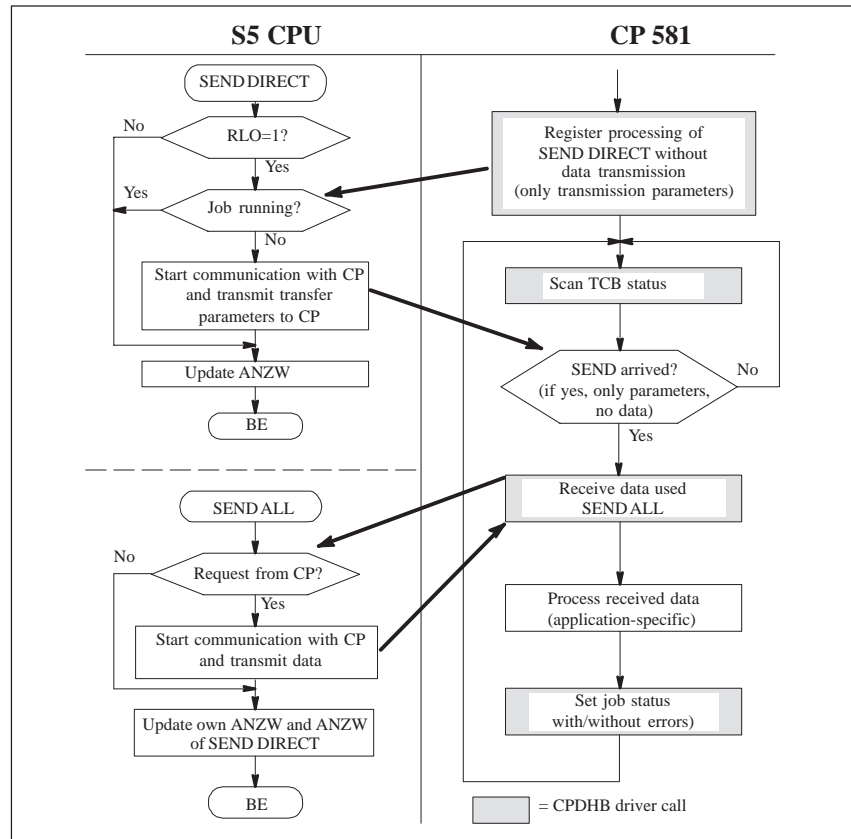


Figure 11-6 Basic Sequence “SEND DIRECT with Separate Data Transfer”

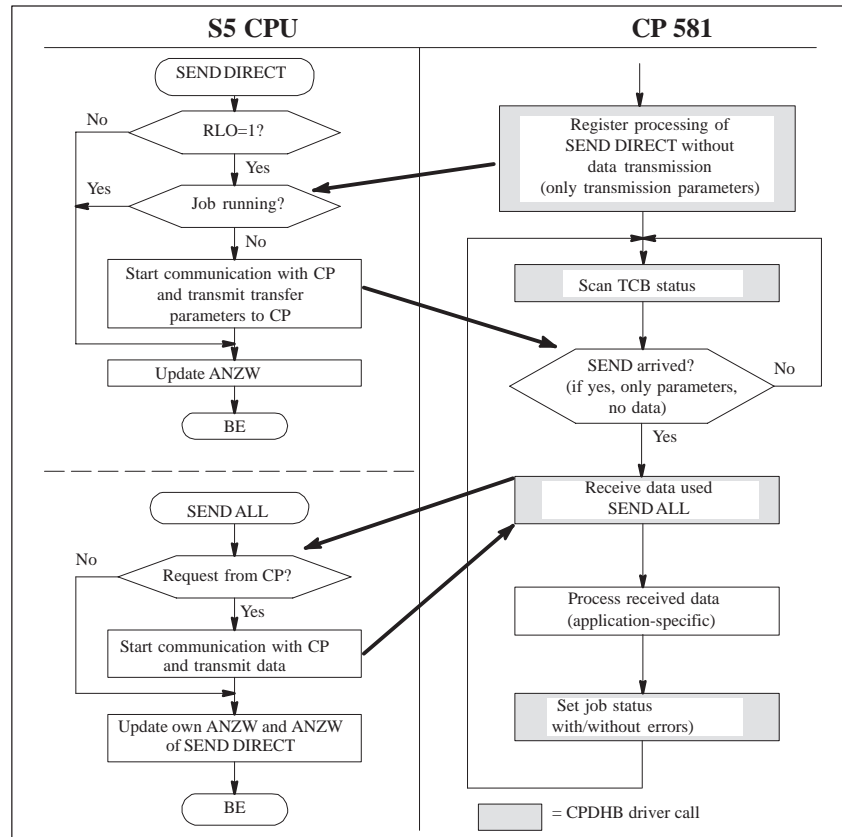


Figure 11-7 Basic Sequence "SEND DIRECT with Direct Data Transfer"

You can process the useful data in your CP 581 program once they have been transferred.

You must inform the S5 program of the end of transmission by means of a driver call "set job status." Another transmission can then usually take place immediately.

FETCH sequence:

Figure 11-8 shows the sequence of a data transfer with FETCH:

The sequence is similar to that with SEND except that the transfer direction is reversed.

The CP 581 program initially registers a TCB for SEND/FETCH with the driver. It then waits until the required FETCH job has been placed by the S5 program at the CPU end.

Once the CP 581 program has recognized that the job has been received, it must provide the data required for the transfer and send these to the CPU.

You must inform the S5 program of the end of the transmission as with SEND DIRECT by means of a driver call "Set job status." The next job can then usually be processed immediately.

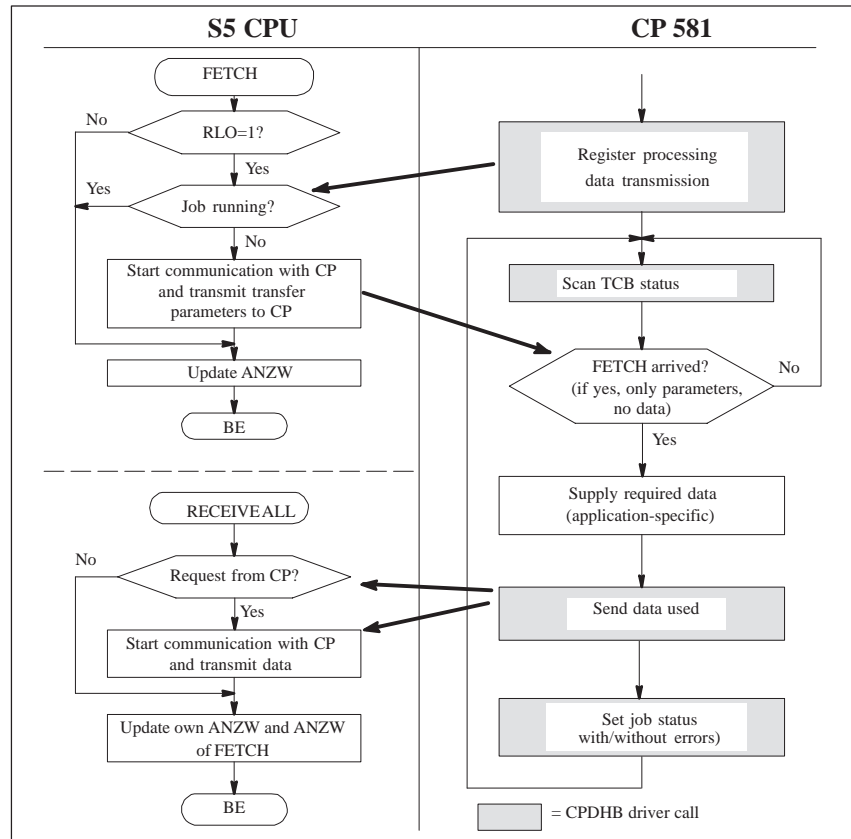


Figure 11-8 Basic Sequence FETCH

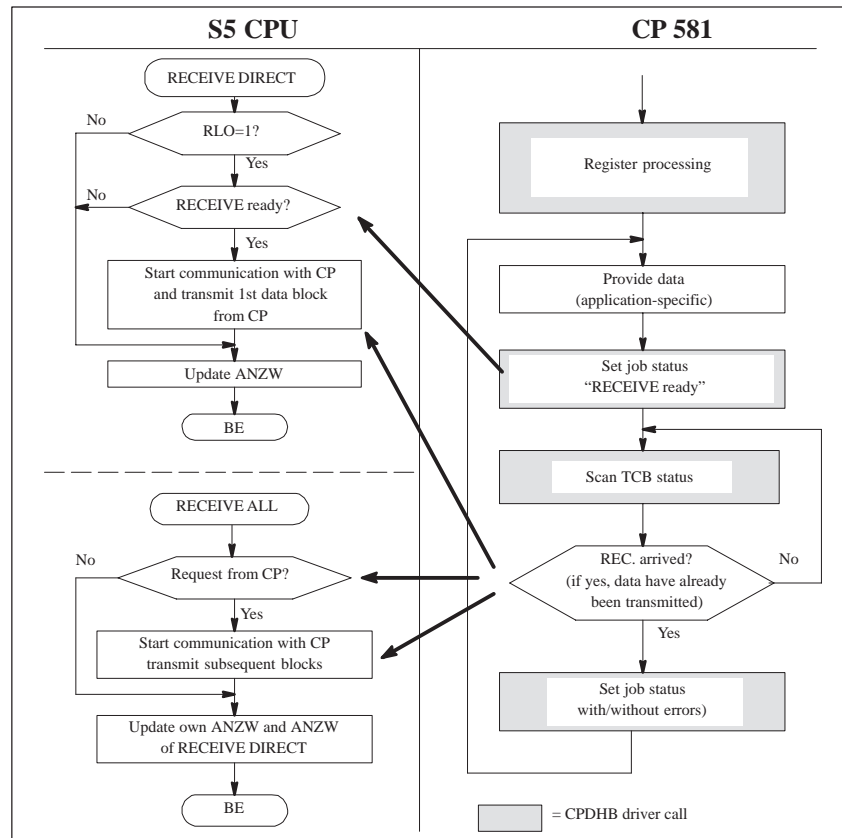
RECEIVE DIRECT job:

Figure 11-9 Basic Sequence RECEIVE DIRECT

Figure 11-9 shows the sequence of a data transfer with RECEIVE DIRECT.

As with SEND and FETCH, you must first register a TCB with the driver in your CP 581 program. The DHB description of this TCB must contain the code for the RECEIVE block, however.

Once the data for the transmission are ready at the CP end, the CP 581 program must indicate to the S5 program that it is ready for communication by means of a corresponding driver call "Set job status."

The useful data are then usually transmitted directly to the CPU when the RECEIVE call arrives, i.e. the data used have already been transmitted when the user program detects that a RECEIVE job has arrived.

The CP 581 program then signals using "Set job status" that the job has been finished, and the sequence can begin again.

TCB for Transfer Functions with Direct Jobs

You must enter the following parameters in the TCB for transfer functions with direct jobs:

- Number of CPU with which the data are to be exchanged,
- DHB description,
- DHB job number for the job to be handled,
- Address (offset and segment) and length of the buffer area.

If your CP 581 program is to handle different direct jobs, you must register a transfer control block with the CPDHB driver for **every job** present in it. It is only possible to wait for **one direct job** with **one** TCB.

The transmission parameters for area identification, DB number, area offset and area length result from the parameter settings of the DHB in the S5 program at the CPU end and are entered from the CPDHB driver into the TCB during handling of the job.

The following table shows you the assignment between the parameter settings of the DHB and the resulting transmission parameters:

Table 11-12 Assignment of DHB Parameters to TCB TransmissionParameters

DHB parameter	TCB parameter
QTYP/ZTYP	Area identification
DBNR	DB number
QANF/ZANF	Area offset
QLAE/ZLAE	Area length

Parameterization of Driver Functions for Direct Jobs

This subsection lists the driver functions you require for direct jobs and explains how you must parameterize the driver calls for these functions.

Before calling a driver function you must assign parameters to the TCB as described in Section 11.4.5 (page 11-44).

Log on job reception:

Application:

This function is used to register a TCB for reception of direct jobs. Your CP 581 program can only wait for **one specific** DHB job with **one** TCB. A job may only be registered **once**.

Call parameters:

Register AX: Function No. = **11H**
 Register BX: TCB address/offset
 Register ES: TCB address/segment

Condition codes:

Register AX:

0: function has been terminated successfully

Negative: function has been aborted following an error:

- 2: CPU is not synchronized
- 3: CPU number is incorrect
- 6: DHB type is incorrect
- 7: TCB is already logged on
- 9: job number is too high
- 10: job number is already in use

The registers which are not listed are not modified by the driver.

Scan TCB status:
Application:

Once you have registered reception of a job in your CP 581 program, you must first check (except with RECEIVE DIRECT) whether a direct job has arrived in order to then send or receive data. The function "Scan TCB status" is available for this purpose.

When handling a data transfer with RECEIVE DIRECT, you must use the function following provision of the data and setting of the status to scan whether the data have been fetched by the CPU.

The driver checks the status of the defined TCB once the function has been called. If a direct job has arrived, the driver provides more detailed information on the job in AX.

The TCB addressed in the call must previously have been transmitted to the driver for management using the function "Log on job reception."

Call parameters:Register AX: Function No. = **16H**

Register BX: TCB address/offset

Register ES: TCB address/segment

Condition codes:

Register AX:

- Positive: 10: TCB waiting
- 11: SEND job has arrived
- 12: RECEIVE job has arrived
- 13: FETCH job has arrived
- 14: SEND job with RW parameters has arrived
- 15: FETCH job with RW parameters has arrived
- 16: SEND job with NN parameters
- 17: RECEIVE job with NN parameters
- 18: FETCH job with NN parameters
- 19: SEND ALL job finished
- 20: RECEIVE ALL job finished

Register AX:

Negative: function has been aborted following an error:

- 5 TCB not logged on
- 51 aborted - negative acknowledgment
- 52 aborted - data area not accessible
- 53 buffer too small
- 54 aborted by SYNCHRON
- 55 aborted by RESET
- 56 unexpected response with FETCH
- 57 unexpected response with RECEIVE DIRECT
- 58 unexpected response with RECEIVE ALL
- 59 unexpected response with SEND
- 60 unexpected response with SEND ALL
- 61 aborted by timeout

- 80 aborted - parameter error
- 81 parameter invalid
- 82 DB/DX does not exist
- 83 area too small
- 84 area does not exist
- 85 status word error

The registers which are not listed are not modified by the driver.

Receive data used:

Application:

You must call this function in your CP 581 program if you wish to receive data from a CPU with a job "SEND DIRECT with separate data transfer." You may only call the function if the program has previously received a SEND DIRECT job from the CPU.

The TCB addressed in the call must previously have been transmitted to the driver for management using the function "Log on job reception."

Call parameters:

- Register AX: Function No. = **14H**
- Register BX: TCB address/offset
- Register ES: TCB address/segment
- Register CX: Timeout in 55-ms units:

- ≠ 0: The function is aborted with an error status bit if the transmission has not been terminated within the set time.
- = 0: The calling program is continued immediately, the transmission is executed in the background (background job). In this case you can recognize the end of the transmission with the function "Scan TCB status."

Condition codes:

Register AX:

0: function finished successfully

Negative: function has been aborted following an error:

- 2 CPU not synchronized

- 3 wrong CPU number

- 4 TCB busy

-51 aborted - negative acknowledgment

-52 aborted - data area not accessible

-53 buffer too small

-54 aborted by SYNCHRON

-55 aborted by RESET

-56 unexpected response with FETCH

-57 unexpected response with RECEIVE DIRECT

-58 unexpected response with RECEIVE ALL

-59 unexpected response with SEND

-60 unexpected response with SEND ALL

-61 Aborted by timeout

negative: -80 aborted - parameter error

-81 parameter invalid

-82 DB/DX does not exist

-83 area too small

-84 area does not exist

-85 status word error

The registers which are not listed are not modified by the driver.

Send data used:

Application:

You must call this function in your CP 581 program if you wish to send data to a CPU with a job FETCH.

You may only call the function if the program has previously received a FETCH job from the CPU.

Call parameters:Register AX: Function No. = **15H**

Register BX: TCB address/offset

Register ES: TCB address/segment

Register CX: Timeout in 55-ms units:

≠ 0: The function is aborted with an error status bit if the transmission has not been terminated within the set time.

= 0: The calling program is continued immediately, the transmission is executed in the background (background job). In this case you can recognize the end of the transmission with the function "Scan TCB status."

Condition codes:

Register AX:

0: function finished successfully

Negative: function has been aborted following an error:

- 2 CPU not synchronized
- 3 wrong CPU number
- 4 TCB being processed

- 51 aborted - negative acknowledgment
- 52 aborted - data area not accessible
- 53 buffer too small
- 54 aborted by SYNCHRON
- 55 aborted by RESET
- 56 unexpected response with FETCH
- 57 unexpected response with RECEIVE DIRECT
- 58 unexpected response with RECEIVE ALL
- 59 unexpected response with SEND
- 60 unexpected response with SEND ALL
- 61 aborted by timeout

- 80 aborted - parameter error
- 81 parameter invalid
- 82 DB/DX does not exist
- 83 area too small
- 84 area does not exist
- 85 status word error

The registers which are not listed are not modified by the driver.

Set job status:

Application:

You use this function at the end of a direct job to inform the S5 program via the status word of the corresponding DHB that the job has been completely processed. At the same time you inform the S5 program whether errors have occurred and which type of errors have been detected.

With a RECEIVE DIRECT job you must additionally inform the S5 program with this function that data are ready for transfer on the CP 581.

The status of the TCB addressed by the function call must previously have been scanned using the function "Scan job reception" (except with RECEIVE DIRECT).

The TCB parameters are not changed.

Call parameters:

Register AX: Function No. = **13H**

Register BX: TCB address/offset

Register ES: TCB address/segment

Register CX: CH = 0, CL = status identification, see Section 11.4.5
(page 11-50)

Condition codes:

Register AX:

0: function finished successfully

Negative: function has been aborted following an error:

- 2 CPU not synchronized

- 3 wrong CPU number

- 5 TCB not logged on

The registers which are not listed are not modified by the driver.

Global scan of DHB job reception:**Application:**

You can use this function to check the status of all registered TCBs for a CPU. If at least one job is present, this is passed on to the calling program.

A TCB is not required for the function.

Call parameters:Register AX: Function No. = **17H**

Register CX: CPU number

Condition codes:

Register AX:

Positive: 30: no job received

31: job received or a SYNCHRON
or RESET has taken place

Negative: - 3 wrong CPU number

The registers which are not listed are not modified by the driver.

Log off job reception:**Application:**

Before you terminate your CP 581 program, for example when data transfer has been completed, you must use this function to remove a previously registered job from the CPDHB driver management again. The TCB of the job must previously have been transferred to the driver for management using the function "Log on job reception."

Call parameters:Register AX: Function No. = **12H**

Register BX: TCB address/offset

Register ES: TCB address/segment

Condition codes:

Register AX:

0: function finished successfully

Negative: function has been aborted following an error:

- 2 CPU not synchronized

- 3 wrong CPU number

- 4 TCB not logged on

- 8 TCB unknown

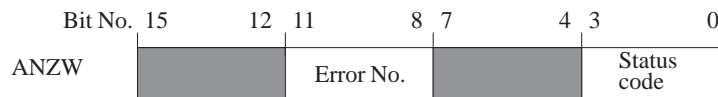
The registers which are not listed are not modified by the driver.

**Caution**

If you do **not** log off the used TCB before terminating your program, the driver writes the CP 581 memory when a call occurs with the TCB address known to it, and this can lead to data losses. Furthermore, the corresponding S5 program is not informed that data transfer is not currently possible.

Status Codes for Direct Jobs

You use the status code to inform the S5 program of the status of a direct job. The code has 8 bits or 2 hexadecimal digits. The job status is stored in the right-hand digit, an error number in the left-hand digit. The status code is stored in the 1st DHB status word using the driver and the corresponding DHB:



Status codes whose meanings are listed in the following Tables 11-13 and 11-14 are defined for the transfer jobs SEND/FETCH and RECEIVE.

Error numbers from 6 to 15 (CP error 6 etc.) are provided in the status codes. You must specifically assign these numbers for your application and interpret them accordingly in your S5 program.

Table 11-13 Status Codes for SEND/FETCH Jobs

Status codes for SEND/FETCH			
Code	Meaning	Error No. 1)	Status code
04H	Job finished without errors	0 0 0 0	0 1 0 0
08H	Job finished with DHB error	0 n n n	1 0 0 0
68H	Job finished with CP error 6	0 1 1 0	1 0 0 0
78H	Job finished with CP error 7	0 1 1 1	1 0 0 0
88H	Job finished with CP error 8	1 0 0 0	1 0 0 0

Table 11-13 Status Codes for SEND/FETCH Jobs

Code	Meaning	Error No. 1)	Status code
98H	Job finished with CP error 9	1 0 0 1	1 0 0 0
A8H	Job finished with CP error 10	1 0 1 0	1 0 0 0
B8H	Job finished with CP error 11	1 0 1 1	1 0 0 0
C8H	Job finished with CP error 12	1 1 0 0	1 0 0 0
D8H	Job finished with CP error 13	1 1 0 1	1 0 0 0
E8H	Job finished with CP error 14	1 1 1 0	1 0 0 0
F8H	Job finished with CP error 15	1 1 1 1	1 0 0 0

1) nnn = DHB errors 1 to 5

Table 11-14 Status Codes for RECEIVE

Status codes for RECEIVE			
Code	Meaning	Error No. 1) 2)	Status code 2)
06H	Job finished without errors	0 0 0 0	0 1 1 0
0AH	Job finished with DHB error	0 n n n	1 0 1 0
6AH	Job finished with CP error 6	0 1 1 0	1 0 1 0
7AH	Job finished with CP error 7	0 1 1 1	1 0 1 0
8AH	Job finished with CP error 8	1 0 0 0	1 0 0 0
9AH	Job finished with CP error 9	1 0 0 1	1 0 1 0
AAH	Job finished with CP error 10	1 0 1 0	1 0 1 0
BAH	Job finished with CP error 11	1 0 1 1	1 0 1 0
CAH	Job finished with CP error 12	1 1 0 0	1 0 1 0
DAH	Job finished with CP error 13	1 1 0 1	1 0 1 0
EAH	Job finished with CP error 14	1 1 1 0	1 0 1 0
FAH	Job finished with CP error 15	1 1 1 1	1 0 1 0
03H	RECEIVE ready	q q q q	q q 1 1

1) nnn = DHB errors 1 to 5

2) qq = original contents are retained

11.4.6 Data Transfer Without Direct Jobs

You can use the two driver functions “Read S5 area” and “Write S5 area” for applications which need not be triggered via the S5 program.

Any data area (data blocks, flags, I/Os etc.) can be read from a CPU or written into it using these functions without a SEND DIRECT, RECEIVE DIRECT or FETCH initiating the transfer procedure in addition.

Within the cyclic execution of the CPU, the CPDHB driver only requires for these functions one SEND ALL for transmitting from the CPU to the CP 581 and one RECEIVE ALL for receiving.

TCB for transfer functions without direct jobs:

You must supply the TCB with the following parameters for transfer functions without direct jobs:

- Transmission parameters:
 - CPU number,
 - Area identification,
 - DB No. (only with DB/DX),
 - Area offset and area length of the source or destination data area in the CPU.
- The pointer to the buffer area:
 - Segment,
 - Offset,
 - Length.

The write/read index is always preset to “0” by the CPDHB driver.

Note

If the size of the area to be transmitted exceeds the transfer frame size set using the DHB SYNCHRON, the CPDHB driver automatically divides the area into partial frames. The transmission then requires several S5 cycles depending on the size of the area. Your CP 581 program only continues **after the complete area has been transmitted.**

Read S5 area:

Application:

You can use this function to read an S5 data area in a CPU and to transmit it to the CP 581.

If your CP 581 program calls this function, it is only continued when the complete data area specified in the TCB has been transmitted.

You must enter the following parameters in the TCB before calling the function (see TCB description).

Call parameters:

Register AX: Function No. = **01H**
 Register BX: TCB address/offset
 Register ES: TCB address/segment
 Register CX: Timeout in 55-ms units:

- ≠ 0: The function is aborted with an error status bit if the transmission has not been terminated within the set time.
 = 0: The calling program is continued immediately, the transmission is executed in the background (background job). In this case you can recognize the end of the transmission with function "Scan TCB status."

Condition codes:

Register AX:

- 0: function finished successfully
 Negative: function has been aborted following an error:
- 2 CPU not synchronized
 - 3 wrong CPU number
 - 4 TCB being processed
 - 51 aborted - negative acknowledgment
 - 52 aborted - data area not accessible
 - 53 buffer too small
 - 54 aborted by SYNCHRON
 - 55 aborted by RESET
 - 56 unexpected response with FETCH
 - 57 unexpected response with RECEIVE DIRECT
 - 58 unexpected response with RECEIVE ALL
 - 59 unexpected response with SEND
 - 60 unexpected response with SEND ALL
 - 61 aborted by timeout
 - 80 aborted - parameter error
 - 81 parameter invalid
 - 82 DB/DX does not exist
 - 83 area too small
 - 84 area does not exist
 - 85 status word error

The registers which are not listed are not modified by the driver.

Write S5 area:

Application:

You can use this function to transmit data from the CP 581 to a CPU and to write in an S5 data area.

If your CP 581 program calls this function, it is only continued when the complete data area specified in the TCB has been transmitted.

You must enter the following parameters in the TCB before calling the function (see TCB description).

Call parameters:

Register AX: Function No. = **02H**

Register BX: TCB address/offset

Register ES: TCB address/segment

Register CX: Timeout in 55-ms units:

≠ 0: The function is aborted with an error status bit if the transmission has not been terminated within the set time.

= 0: The calling program is continued immediately, the transmission is executed in the background (background job). In this case you can recognize the end of the transmission with the function "Scan TCB status."

Condition codes: as with "Read S5 area"

The registers which are not listed are not modified by the driver.

11.4.7 Other Driver Functions

Delete background job:

Application:

You can use this function to delete a background job which you have sent using the timeout "0" in order to send or receive data (function numbers 01H, 02H, 14H and 15H). The associated TCB is then also deleted.

The driver returns the error status bit -5 if the TCB cannot be deleted because the data transmission has been terminated in the meantime.

Call parameters:

Register AX: Function No. = **18H**
 Register BX: TCB address/offset
 Register ES: TCB address/segment

Condition codes:

Register AX:
 0: function finished successfully
 Negative: function has been aborted following an error:
 - 2 CPU not synchronized
 - 3 wrong CPU number
 - 5 TCB not logged on

Call driver status:

Application:

You can use this function (for example, after starting your CP 581 program) to scan whether the CPDHB driver is correctly synchronized with a CPU.

Call parameters:

Register AX: Functions No. = **30H**
 Register CX: Number of CPU to be synchronized

Condition codes:

Register AX:
 Positive: 41 CPU synchronous
 40 CPU not synchronous
 Negative: -3 wrong CPU number

The registers which are not listed are not modified by the driver.

Change interrupt for driver call:

Change interrupt for driver call:
--

If the preset interrupt INT-66H for calling the CPDHB driver is already used by other applications in your CP 581 user system, you can assign a different software interrupt to the driver.

This function **cannot** be activated via a **driver call**, but via the following **command**:

CPHTB -INTxx with xx: hexadecimal value of new interrupt;
 permissible values: 60h to 66h

The driver stores this value in the memory and can then only be reached via the new interrupt.

MS-DOS Multiplexer Interrupt (INT 2FH) of the CPDHB Driver

The **MS-DOS** multiplexer interrupt (INT-2FH) is a **special access** to a TSR program. If a program calls INT-2FH, **all installed TSR programs** are called **in succession**. A special identification number (ID number) in the register AH determines **which** TSR program is to be addressed. Register AL can inform the program on which function is to be executed.

The **CPDHB** driver provides only **one** function via INT-2FH: it checks whether the driver is installed and simultaneously provides useful information on the driver.

Call parameters:

Register AL: **01H**
Register AH: **EDH**

Condition codes:

Register AX: EDEDH if the driver is present and if the function
 was executed
Register BX: Driver version
Register CL: Interrupt number for driver call
Register DX: Used internally by the driver
Register EX: Used internally by the driver

Function description:

The INT-2F function 01H checks whether the CPDHB driver is present in the CP 581 memory. A copy of the ID number is supplied in AL if the driver is present. The driver enters the current interrupt number for the job interface into CL since the preset number can only be modified by a command to the driver (see Section 11.4.7). The driver enters the release version of the driver software into BX.

- **Example of call of CPDHB driver via multiplexer interrupt:**

```

CPHTBID EQU 0EDH

MOV AL,01H
MOV AH,CPHTBID
INT 2FH
CMP AL,CPHTBID ;Driverpresent?
JNE error
.
. ;Driverpresent!
error:
. ;Driver not present!
.

```

11.5 Testing the Application

11.5.1 Procedure

Proceed in the following steps to determine whether the S5 and CP 581 programs you have produced correctly handled the desired communication between the CPU and CP 581:

1. Make sure that no system programs for standard applications (process data acquisition etc.) are active during the complete test phase, thus preventing side effects. You must remove the corresponding start commands from the AUTOEXEC.BAT file for this purpose if they are entered there.
2. Always test your programs initially with **one** CPU, even if you wish to exchange data later with several CPUs.
3. First test your S5 programs without the CP 581, then only with the driver, and finally together with your CP 581 program (see Section 11.5.2).
4. First test your CP 581 program without the CPDHB driver, then with the driver but without the CPU and finally together with your S5 program on a CPU (see Section 11.5.3).
5. Check that the data transferred between the CPU and CP 581 agree. Remember to observe the different storage procedures of the various data formats in the CPU and CP 581 memories (see Section 11.5.3).
6. If data transfer with a CPU is executed correctly, you can incorporate further CPUs into the sequence if this is required for your application and is permissible with your programmable controller, or you can also activate the standard application programs step-by-step. Consider whether special coordination methods are necessary in this case. If so, you must first implement these measures in your programs.

11.5.2 Testing the S5 Program

Online functions are available to test your S5 program on your programmer (see /4/). You can determine with these functions whether the DHB calls are processed in your S5 program and what information is stored in the status word of the DHB calls.

The individual procedure depends on your special application and cannot be recommended universally. The sequence of basic steps listed in Section 11.5.1 (without CP 581, only with CPDHB driver, then with CP 581 program)

Useful information for testing is stored in the status words. Section 11.3.6 shows you the meaning of the various bits in the status word.

11.5.3 Testing the CP 581 Program

Before commencing the test of your CP 581 program, you must clarify which test aids are available. This depends on the language in which you have written your program.

Irrespective of the programming language and the test aids, it is difficult to test a program if another program is active simultaneously (CPDHB driver).

Therefore initially test your program without real driver calls (you can simulate these via an auxiliary subroutine, replace INT commands by subroutine calls) in order to check the logical sequence.

If the program is executed correctly during this “dry test,” you can call the CPDHB driver using correct INT commands in your program. A CPU should not yet be active. Your program must be informed in this test step by the driver that the addressed CPU is not synchronous (see result status bits of the driver functions).

If you then test the interaction with a CPU, the bits in the CPDHB driver provide valuable information on when logic errors have occurred.

In a test version of your CP 581 program you should therefore output the status bits on the monitor or on a printer with a reference to the special call following each driver call.

If your CP 581 program “crashes” during the interactions, you should perhaps carry out a “mixed test” with real and simulated driver calls in which the simulated driver calls are replaced step-by-step by real calls.

11.5.4 Representation of the S5 Data in the CP 581 Memory

Data are stored in the CP 581 in the Intel data format. This differs from the representation of the S5 data in a CPU.

You must take this difference into consideration if you wish to check whether the data transferred between CPU and CP 581 are correct.

The following figures show you how the various data formats are stored in the CPUs and in the CP 581.

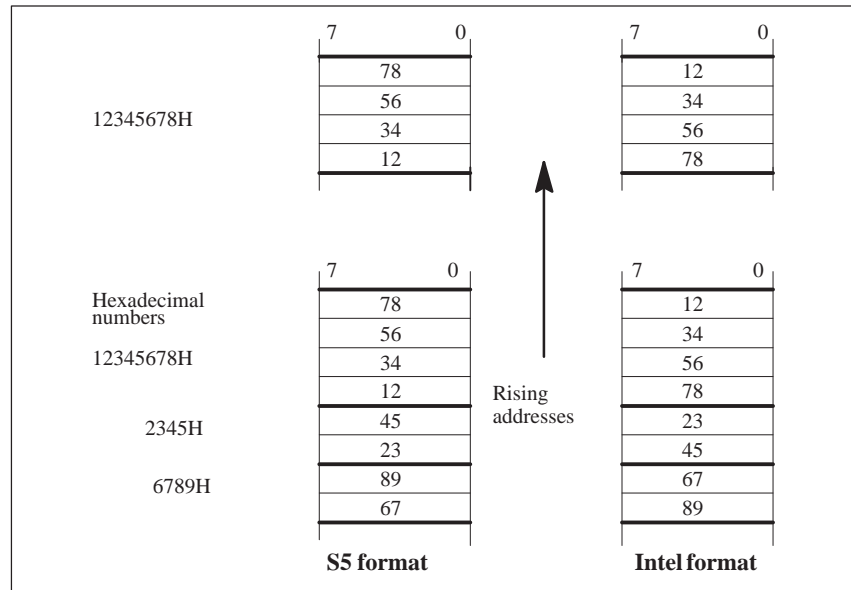


Figure 11-10 Data Representation in S5 and Intel Formats

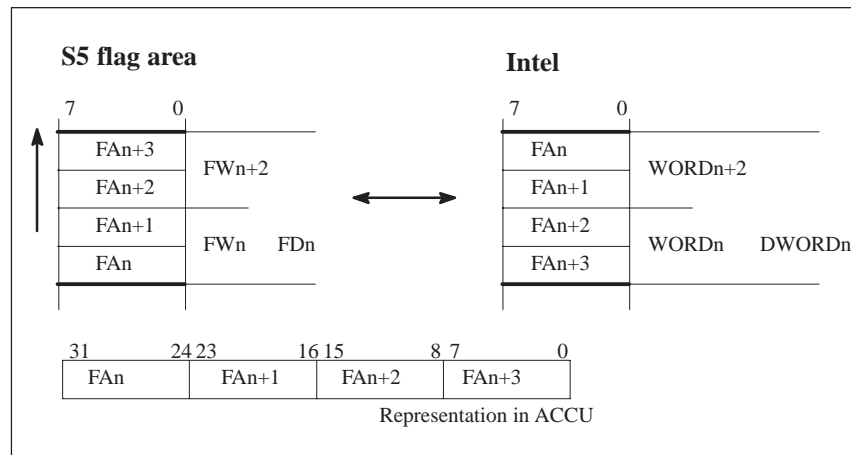


Figure 11-11 Representation of S5 Flag Data in Intel Format

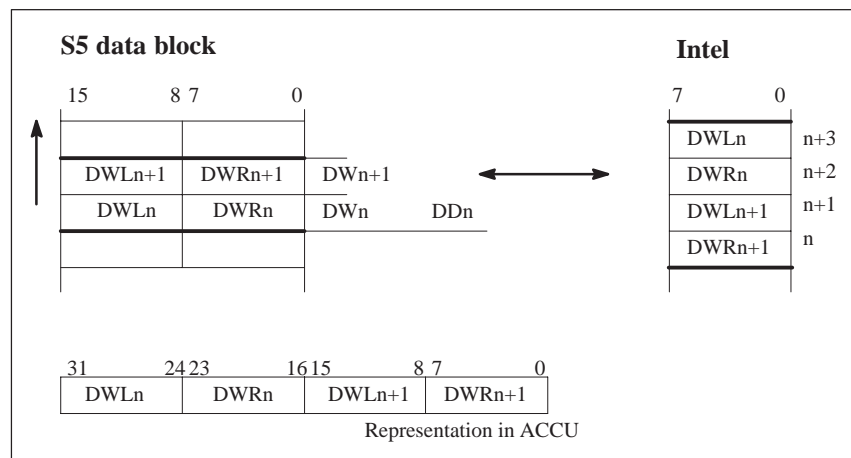


Figure 11-12 Representation of S5 Data Block Data in Intel Format

12

Application Examples

This chapter describes the use of the CP 581 communications processor by means of application examples. You learn:

- How you can monitor the temperature variation of a process using the process data acquisition function,
- How you can use the mass storage functions to transfer data from the S5 CPU to the CP and back again,
- How you can use the command interpreter to output messages from the S5 CPU on a message printer using an MS-DOS command,
- Which structure the hard disk directory EXAMPLE has with a collection of application examples (following the installation of the system software).

Chapter Overview

Section	Description	Page
12.1	Example "Process Data Acquisition"	12-2
12.1.1	Task/Problem	12-2
12.1.2	Starting to Solve the Problem	12-2
12.1.3	Structure of Solution	12-3
12.1.4	Individual Working Steps	12-5
12.2	Example "Mass Storage Functions"	12-13
12.2.1	Task/Problem	12-13
12.2.2	Starting to Solve the Problem	12-13
12.2.3	Structure of Solution	12-14
12.2.4	Individual Working Steps	12-16
12.3	Example "Command Interpreter"	12-22
12.3.1	Task/Problem	12-22
12.3.2	Starting to Solve the Problem	12-22
12.3.3	Structure of Solution	12-22
12.3.4	Individual Working Steps	12-24
12.4	Application Examples in the Directory "EXAMPLE"	12-28

12.1 Example “Process Data Acquisition”

12.1.1 Task/Problem

You wish to monitor the temperature variation of a process using the CP 581 communications processor which is installed in the subrack of your S5-115U programmable controller.

The measurement points are to be sampled 60 times per hour over a period of 12 hours, and the temperature values stored on the hard disk of the CP 581. The process has 5 temperature measurement points. At the end of the 12 hours, the values are to be evaluated using an MS-DOS user program (for example, Lotus 1-2-3).

12.1.2 Starting to Solve the Problem

The analog values measured at the measurement points must be standardized and stored in the correct sequence in data block DB 10 from data word DW 1 to data word DW 5.

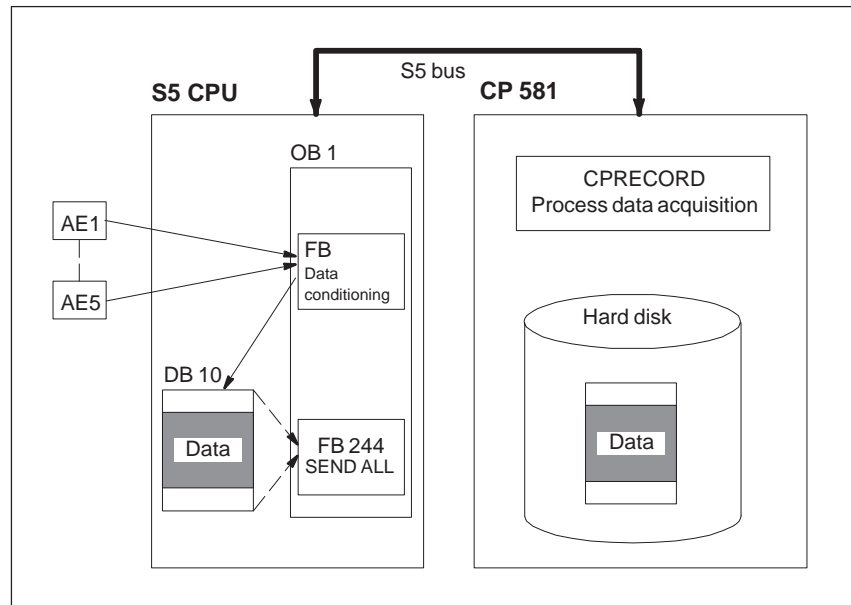


Figure 12-1 Transmission of Measured Values to the Data Block DB 10

The data are transferred to the hard disk of the CP via the data handling block SEND with the function SEND ALL in the OB 1 and via the CPRECORD program for process data acquisition on the CP 581. The data can be called from the hard disk by the evaluation program.

12.1.3 Structure of Solution

The data flow between the S5 CPU and the CP 581 is as follows:

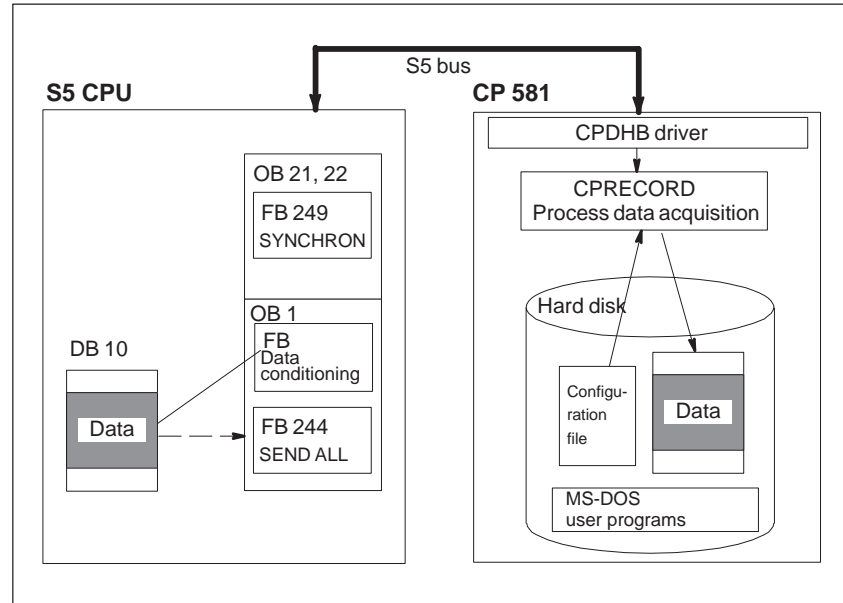


Figure 12-2 Data Transfer between S5 CPU and CP 581 for Process Data Acquisition

- The FB SYNCHRON initializes the interface, and the frame size is agreed upon between the interface and the S5 CPU.
- The FB “Data conditioning” stores the measured data in standardized form in the data block DB 10.
- The FB SEND with the function SEND ALL transfers the data from the DB 10 to the hard disk of the CP via the S5 rearpanel bus and with the assistance of the CPDHB driver and the “process data acquisition” program. The configuration file provides the parameters for the “process data acquisition” program.
- MS-DOS user programs are used to evaluate the data on the hard disk.

For this purpose you must carry out the following measures on the PG, CP 581 and S5 CPU:

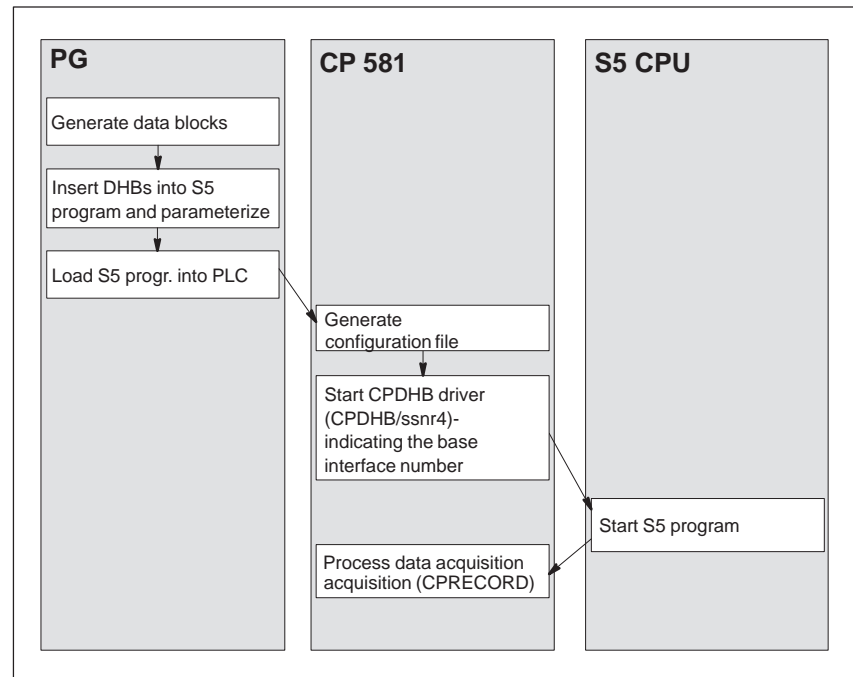


Figure 12-3 Handling Sequence for Process Data Acquisition

12.1.4 Individual Working Steps

The operations you must carry out on the programmer (PG), the communications processor (CP) and the S5 CPU are explained in more detail on the following pages.

Operations on the PG

- **Generation of data blocks**

Generate the data block DB 10 such that 5 temperature values can be stored as fixed-point numbers. The first value should be present in data word DW 1.

```
DB 10
0      :      KF = +0000;
1      :      KF = +0000;
2      :      KF = +0000;
3      :      KF = +0000;
4      :      KF = +0000;
5      :      KF = +0000;
6      :
```

- The data from data block DB 10 are transmitted to the CP via the DHB SEND with the function SEND ALL following triggering (activation of the process data acquisition) by the CP.

- **Installation and parameterization of data handling blocks in S5 program**

Parameterize the DHB SYNCHRON and insert it into the restart organization blocks OB 21 and 22.

(The OB 20 is not available for AG S5-115U.)

OB 21

```
SEGMENT 1      0000
0000          :      CP 581 synchronization
0001          :      with STOP → RUN
0002          :JU   FB 23
0003 NAME:CPSYNC
0004 REP      :      KF +3
0005          :
0006          :BE
```

OB 22

```
SEGMENT 1      0000
0000          :      CP 581 synchronization
0001          :      with power return
0002          :JU   FB 23
0003 NAME:CPSYNC
0004 REP      :      KF +3
0005          :
0006          :BE
```

FB 23

```

SEGMENT 1          0000
  NAME : CPSYNC          CP 581 synchronization
  DECL : REP I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF

0008      :L   KF +0      Repetition counter
000A      :T   FW 14      preset with 0
000B      :
000C LOOP :JU   FB 249
000D NAME :SYNCHRON
000E SSNR :    KY 0,4      Interface number
000F BLGR :    KY 0,6      1 = 16 bytes, 6 = 512 bytes
0010 PAFE :    FY 12
0011      :L   FW 14      Loop counter
0012      :I   1
0013      :T   FW 14,
0014      :L   KF +0      Evaluate PAFE
0016      :L   FY 12
0017      :!=F
0018      :JC   =END      Synchron. successful
0019      :
001A      :L   FW 14      Loop counter
001B      :LW   =REP
001C      :!=F      Repetitions completed
001D      :JN   =LOOP      Repeat synchronization
001E      :STP      Error reaction STOP
001F END   :BE
  
```

- The FB 23 handles the assignment of parameters in the restart OB.
The different restart times of the S5 CPU and the CP are compensated by the repetition factor for the FB SYNCHRON.
- The S5 CPU and the CP 581 are synchronized in “STOP/RUN mode” (OB 21) and in “Automatic warm restart” (OB 22) of the programmable controller by means of the DHB SYNCHRON in the restart OB.

Call the data handling block SEND with the function SEND ALL in the organization block OB 1 and assign its parameters.

```

OB 1
SEGMENT 1,
      :
      :
0013      :JU    FB 244    DHB SEND (SEND ALL)
0014 NAME :SEND
0015 SSNR :    KY 0,4    Interface 4
0016 A-NR :    KY 0,0    A-NR 0 = SEND ALL function
0017 ANZW :    FW 20
0018 QTYP :    KS NN
0019 DBNR :    KY 0,0
001A QANF :    KF +0
001B QLAE :    KF +0
001C PAFE :    FY 13
      :
      :
      :BE

```

- The DHB SEND with the function SEND ALL transfers the measured values from the S5 CPU to the CP via the S5 rearpanel bus.

Operations on the CP 581

- **Loading the S5 program into the programmable controller**

Set the programmable controller to STOP and load the S5 program into the PLC.

- **Generation of configuration file**

You can modify the CPRECORD.INI (C:\EXAMPLE\CPRECORD) configuration file included in the scope of delivery using any text editor.

It is better, however, to leave this file unmodified for reference purposes and to generate a new configuration file on the CP, for example using the text editor EDIT. Any name can be used for the file, but the file name extension must be .INI. You must specify the name of the new configuration file when activating process data acquisition.

CPRECORD searches for the configuration file in the current directory, i.e. in the directory from which it was started. If you wish to use another file or directory, you must specify this in the command CPRECORD in the "/Cconfdat" option.

The configuration file has only one parameter set in our example. Its contents are as follows:

```

;***CPRECORD Configuration file*****
60;      Recording cycle in seconds
1;      Data are read from CPU 1
DB;     SEND ALL accepts the data from the S5 area DB
10;     Number of the DB which contains the relevant data
1;      Data offset in words with respect to start of DB
5;      Data length
D:\BEISPIEL; Directory path in which the ASCII files are stored
;      (the directory must already have been created)
TXT;    File name extension for the ASCII files, the file name
;      is assigned by CPRECORD (in this case 010DXXXX.TXT).
1;      Maximum number of files
720;    Maximum number of data records per file
KF;     Uniform fixed-point data format in DB
;;      Delimiter between individual entries in ASCII file
;      (semicolon in this case)
0;      Data recording is terminated when the
;      defined number of files (1 in this case) is reached
1;      Logging is to be carried out
CP581.LOG; Destination file for the logged data (is generated in the current directory;
;      if you do not require this, you must specify the complete path).
50;     Timeout in seconds, 1 to 3.600 s is permissible
    
```

- The configuration file contains the parameters with which the CPRECORD program carries out the process data acquisition.
- The configuration file is only evaluated when the CPRECORD program is started, i.e. possible modifications of the configuration file do not become effective until you restart the program.

- **Start CPDHB driver and set base interface number**

Following installation of the system software, the start call for the CPDHB driver is present in the AUTOEXEC.BAT file, but excluding the `ssnr` option, i. e. the base interface number is set to the `SETUP` value (default "0," see volume 1) when the driver is started. You must add the `ssnr <ddd>` option with the base interface No. 4 to the driver call. Then, the start call is: **CPDHB /ssnr4**.

The driver is automatically started when the CP 581 is switched on. If you have deleted the start call from the AUTOEXEC.BAT file, you can also start the driver by entering the command **CPDHB /ssnr4**.

- The CPDHB driver is loaded into the main memory (TSR program).

Operations on the S5 CPU

- **Starting the S5 program**

Start the S5 program by restarting the S5 CPU.

- OB 21 or 22 is executed depending on the restart mode. The DHB SYNCHRON is activated, and the status word ANZW and the parameter error byte PAFE are evaluated. In our example, the PLC is set to STOP if the S5 CPU and the CP 581 cannot be synchronized.
- Process data acquisition can be started on the CP if the synchronization is successful. An error message (“CPU not synchronized”) is output if the S5 CPU and the CP are not synchronized.

Operations on the CP 581

- **Activating process data acquisition**

Enter the **CPRECORD** command.

You can activate process data acquisition either by entering the **CPRECORD** command as required or by entering the command **CPRECORD /A** into the AUTOEXEC.BAT file. In the latter case, process data acquisition is automatically activated when the CP 581 is switched on.

- The program is started and outputs the following text:

```
CPRECORD data recording - Version xx
Copyright (c) Siemens AG 1992
CPRECORD installed ...
'CPRECORD \?' indicates command syntax
[1] Data recording started
```

- The program installs itself in the main memory as a TSR program.
- The configuration file is evaluated.
- The start message of the program is entered into the logging file together with the date and time.
- The measured values are transferred to the hard disk of the CP 581.
- The CPRECORD program is executed in the background. Other programs can be executed in the foreground; reaction times are influenced though.

- **Scan status of data recording**

Enter the command CPRECTL /S1 in the control program CPRECTL (S = option “Status,” 1 = number of parameter set):

- The status of data recording is output during operation. It is displayed, for example, that data recording is in operation, that the timer is at 32 seconds, that data record 2 in file 1 is currently being processed and that the data recording is not permanent:

```
Timer = 32
[1] Recording active = 1
[1] Data record = 2
[1] File = 1
[1] Permanent cycle = 0
```

- **Interrupt and continue data recording**

Enter the command **CPRECCTL /E1** in the control program **CPRECCTL** to interrupt and **CPRECCTL /B1** to continue (E = option “End,” B = “Begin,” 1 = number of parameter set):

- Data recording is stopped by the command “...E1.” The program is still present in the main memory, however, and data recording is continued by entering the command “...B1.”
- The program always searches for the file with the newest date and creates the follow-up file. The advantage of this procedure is that the old data are retained and are not overwritten.

- **Restarting data recording**

Data recording is aborted once the defined number of files has been reached. You must first remove the **CPRECORD** program from the memory using the command **CPRECORD /U** and delete the old data from the directory **D:\EXAMPLE** before you can restart by activating data recording again.

- **Reaction to power failure during data recording**

If the **CPDHB** driver and **CPRECORD** are entered in the **AUTOEXEC.BAT** file, the data recording is automatically started when the power returns.

If this is not the case, load the two programs again by entering the commands (**CPDHB** and **CPRECORD**).

- **CPRECORD** searches for the file with the newest date and generates the follow-up file. Old data are not lost.

- **Logging of messages**

Since you have selected logging in the configuration file, all messages from **CPRECORD** are entered in the log file **CP581.LOG** which is also specified in the configuration file. This applies both to error messages and to screen outputs generated when starting or accessing the program.

You can also delete the log file again if necessary (for example, if you no longer require old messages). A new log file is generated automatically as soon as new messages are output.

- **Evaluation of recorded data**

The data stored in the file D:\EXAMPLE\010D0000.TXT can be evaluated using an MS-DOS program, for example Lotus 1-2-3, and output, for example, in the form of a temperature curve.

The file D:\EXAMPLE\010D0000.TXT contains the measured values as fixed-point numbers:

```
Data record 1:  ___+81; ___+80; ___+80; ___+79; ___+79;
                ___+79; ___+79; ___+79; ___+78; ___+78;
                ___+78; ___+78; ___+77; ___+77; ___+77;
                :       :       :       :       :
Data record 720: ___+80; ___+79; ___+79; ___+79; ___+78;
```

Transfer the above-mentioned file to your Lotus worksheet using the command sequence "Transfer external values." Enter the times for the X-axis into column A depending on the task. The worksheet then appears approximately as follows:

```
A:A1:                                     READY
A   A   B   C   D   E   F   G   H
1   6           81  80  80  79  79
2           79  79  79  78  78
3           78  78  77  77  77
4           76  76  76  75  75
5           75  77  77  77  78
6           78  78  78  79  79
7           79  79  80  80  80
8           81  81  81  82  82
9           82  82  83  83  83
10          83  84  84  84  84
11          84  85  85  84  84
12          80  80  81  81  81
13          81  80  80  79  79
14          79  79  79  78  78
15          78  78  77  77  77
16          76  76  76  75  75
17          75  77  77  77  78
18          78  78  78  79  79
19          79  79  80  80  80
20          81  81  81  82  82
22.07.92 14:50
```


12.2 Example “Mass Storage Functions”

12.2.1 Task/Problem

You wish to transfer large quantities of data from the S5 CPU to the CP 581 communications processor and bring the data back to the S5 CPU as required.

This may be necessary if process data are to be transferred from an S5-115U to a master computer, and the master computer fails or the link is faulty. In this case the process data are to be buffered on the hard disk of the CP to prevent data losses. Once the master computer is available again, it can request transmission of the buffered data from the CPU.

12.2.2 Starting to Solve the Problem

The following four functions of the CPMASS mass storage program are available to solve the problem:

- Preselect directory on CP 581,
- Transmit data from S5 CPU to CP 581,
- Transmit data from CP 581 to S5 CPU,
- Delete S5F files on CP 581.

Thus without further programming knowledge on the CP 581 side, and merely using various data handling blocks which you installed in your STEP 5 user program,

- you can determine the directory in which the data to be transmitted are to be stored
- activate and execute data transmission and
- delete S5 files which are not required.

In the following description of how to solve the problem, the individual CPMASS functions are triggered via inputs.

12.2.3 Structure of Solution

The data flow between the S5 CPU and the CP 581 is as follows:

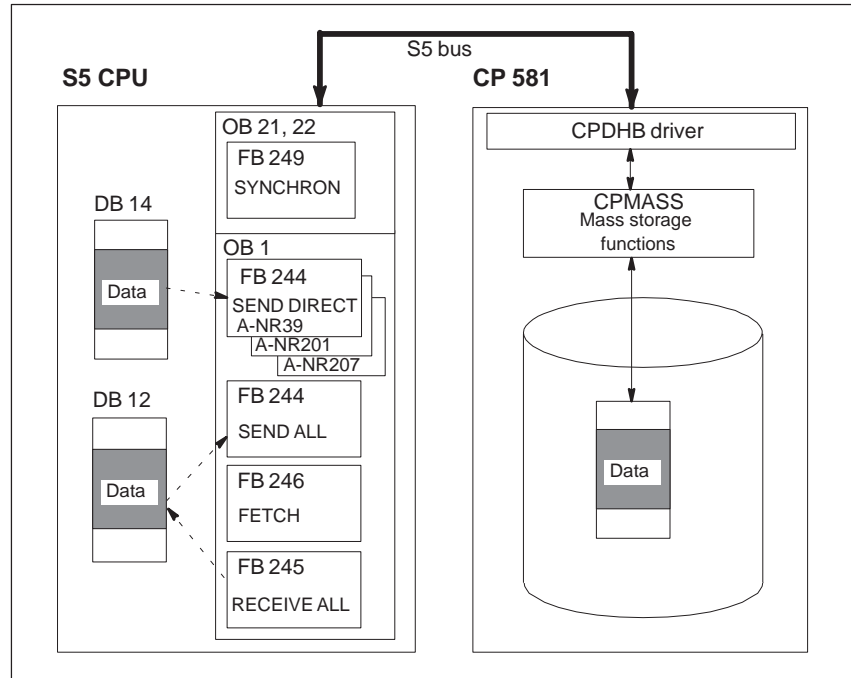


Figure 12-5 Data Transfer between S5 CPUs and CP 581 via S5 Bus with Mass Storage Functions

- The FB SYNCHRON initializes the interface, and the frame size is agreed upon between the interface and the S5 CPU.
- The FB SEND with the function SEND DIRECT and job No. 201 selects the directory on the hard disk of the CP into which the data are to be transmitted. If a directory is not preset, the data are stored in the directory C:\CPU1, C:\CPU2, C:\CPU3 or C:\CPU4, with the S5-115U in C:\CPU1. These directories are automatically created if they are not present when CPMASS is started.
- The FB SEND with the function SEND DIRECT triggers transmission of the data to the CP.
- The FB SEND with the function SEND ALL transfers the data from the DB 12 to the hard disk of the CP via the S5 rearpanel bus and with the assistance of the CPDHB driver and the CPMASS program.
- The FB FETCH triggers transmission of the data to the S5 CPU.
- The FB RECEIVE with the function RECEIVE ALL transfers the data from the hard disk of the CP to the DB 12 on the S5 CPU via the S5 rearpanel bus and with the assistance of the CPMASS program and the CPDHB driver.
- The FB SEND with the function SEND DIRECT and job No. 207 deletes the directory specified in DB 14.

For this purpose you must carry out the following measures on the PG, CP 581 and S5 CPU:

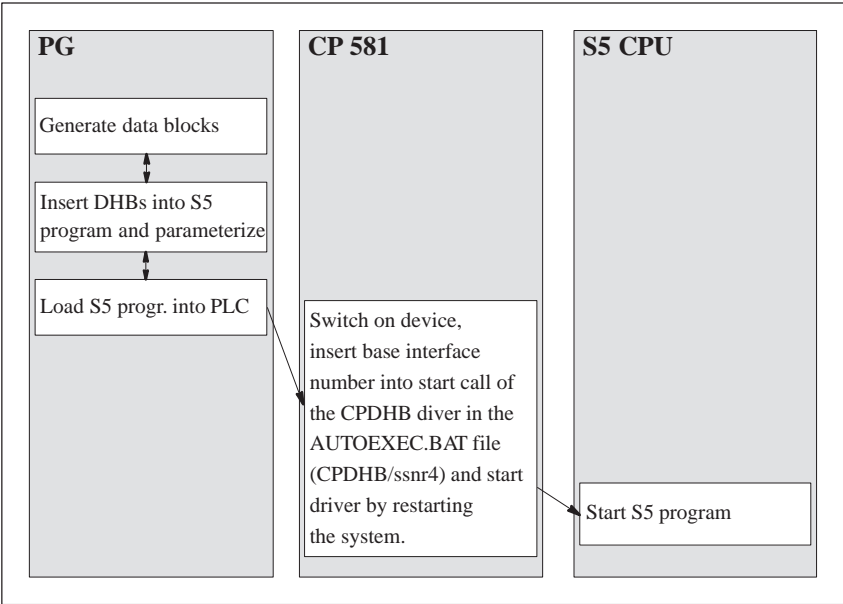


Figure 12-6 Handling Sequence for Mass Storage Functions

12.2.4 Individual Working Steps

The operations you must carry out on the programmer (PG), the communications processor (CP) and the S5 CPU are explained in more detail on the following pages.

Operations on the PG

- **Generate data blocks**

Generate the data block DB 14 and enter the target directory for the data you wish to transmit from the S5 CPU to the CP 581.

```
DB 14
  0      : KS = 'C:\CPU1\DIR1';
  6      : KH = 0000;
  7      :
          :
```

- The path information in DB 14 is transferred to the CP via the DHB SEND with the function SEND DIRECT and job No. 201.

Generate the data block DB 12 with 2043 data words. It must contain the data you wish to transfer from the S5 CPU to the CP.

```
DB 12                                LEN = 2048
  0      : KH = AAAA;
  1      : KH = 0001;
  2      : KH = 0001;
  3      : KH = 0001;
          :
          :
 2042   : KH = 0001;
 2043   :
```

- The data from data block DB 12 are transmitted to the hard disk of the CP via a DHB SEND with the function SEND ALL following triggering by the DHB SEND with the function SEND DIRECT.

- **Installation and parameterization of data handling blocks in S5 program**

Parameterize the DHB SYNCHRON and insert it into the restart organization blocks OB 21 and 22 (for list, see Section 12.1.4).

- The S5 CPU and the CP 581 are synchronized in “STOP/RUN mode” (OB 21) and in “Automatic warm restart” (OB 22) of the programmable controller by means of the DHB SYNCHRON in the restart OB.

Call the data handling block SEND with the function SEND ALL in the organization block OB 1 and assign its parameters.

```

OB 1
SEGMENT
0000      : *****
0001      : Example of mass storage function
0002      : CP 581/S5-115U
0003      : *****
0004      :
0005      :JU   FB 244   DHB SEND (SEND ALL)
0006 NAME :SEND
0007 SSNR :   KY 0,4
0008 A-NR :   KY 0,0
0009 ANZW :   FW 100
000A QTYP :   KS
000B DBNR :   KY 0,0
000C QANF :   KF +0
000D QLAE :   KF +0
000E PAFE :   FY 104
000F      :

```

- The DHB SEND with the function SEND ALL transfers the data from the S5 CPU to the CP via the S5 rearpanel bus.

Call the data handling block RECEIVE with the function RECEIVE ALL in the organization block OB 1 and assign its parameters.

```

OB 1
000F      :
0010      :JU   FB 245   DHB RECEIVE (RECEIVE ALL)
0011 NAME :RECEIVE
0012 SSNR :   KY 0,4
0013 A-NR :   KY 0,0
0014 ANZW :   FW 106
0015 ZTYP :   KS
0016 DBNR :   KY 0,0
0017 ZANF :   KF +0
0018 ZLAE :   KF +0
0019 PAFE :   FY 110
001A      :

```

- The DHB RECEIVE with the function RECEIVE ALL transfers the data from the CP to the S5 CPU via the S5 rearpanel bus.

Call the data handling block SEND with the function SEND DIRECT in the organization block OB 1 and assign its parameters.

```

OB 1
001B      :
001C      :A   I 12.0           Trigger "writing of data
001D      :AN  F 11.0           from DB 12 to CP 581 file"
001E      :=   F 11.1
001F      :S   F 11.0
0020      :AN  I 12.0
0021      :R   F 11.0
0022      :
0023      :A   F 11.1
0024      :JU  FB 244           DHB SEND (SEND DIRECT)
0025      NAME :SEND
0026      SSNR :    KY 0,4       Interface 4
0027      A-NR :    KY 0,39      Job No. 39, poss. range 1-99
0028      ANZW :    FW 112
0029      QTYF :    KS DB        Source DB
002A      DBNR :    KY 0,12      No. 12
002B      QANF :,    KF +0       Initial address
002C      QLAE :    KF +2043     Number of data words
002D      PAFE :    FY 116
002E      :

```

- The DHB SEND with the function SEND DIRECT activates the CP 581 and transfers to it the address of the data to be transmitted to the CP.
- The name of the destination file on the CP 581 is generated from the data block number (DBNR) and the job number (A-NR). The destination file is "012DB039.S5F" in our example.
- In order to differentiate between the files during cyclic data recording of the same DB, it is necessary either to increment the job number or to switch over the directory.

Call the data handling block FETCH in the organization block OB 1 and assign its parameters.

```

OB 1
002E      :
002F      :A      I 12.1      Trigger "reading of data
0030      :AN     F 11.2      from CP and storage in DB 12"
0031      :=     F 11.3
0032      :S     F 11.2
0033      :AN     I 12.1
0034      :R     F 11.2
0035      :
0036      :A     F 11.3
0037      :JU    FB 246      DHB FETCH
0038      NAME :FETCH
0039      SSNR :   KY 0,4      Interface 4
003A      A-NR :   KY 0,39     Job No. 39
003B      ANZW :   FW 118
003C      ZTYP :   KS DB      Destination DB
003D      DBNR :   KY 0,12     No. 12
003E      ZANF :   KF +0      Initial address
003F      ZLAE :   KF +2043    Number of data words
0040      PAFE :   FY 122
0041      :

```

- The DHB FETCH activates the CP 581 and transfers to it the address of the data to be transmitted to the S5 CPU.

Call the data handling block SEND with the function SEND DIRECT and job number 201 in the organization block OB 1 and assign its parameters.

```

OB 1
0042      :
0043      :A     I 12.2      Trigger "preselection of
0044      :AN     F 11.4      directory on CP 581"
0045      :=     F 11.5
0046      :S     F 11.4
0047      :AN     I 12.2
0048      :R     F 11.4
0049      :
004A      :A     F 11.5
004B      :JU    FB 244      DHB SEND (SEND DIRECT)
004C      NAME :SEND
004D      SSNR :   KY 0,4      Interface 4
004E      A-NR :   KY 0,201    Job No. 201, preselect directory
004F      ANZW :   FW 124
0050      QTYP :   KS DB      Directory and path data
0051      DBNR :   KY 0,14     in DB 14
0052      QANF :   KF +0      Initial address
0053      QLAE :   KF +7      Number of data words
0054      PAFE :   FY 128
0055      :

```

- If the DHB SEND with the function SEND DIRECT and job number 201 is called, it transfers a string to the CP which contains the path data of the destination directory for the data transmission.
- If the string is larger than the set frame size, the DHB SEND with the function SEND ALL transmits the remaining data to the CP.

Call the data handling block SEND with the function SEND DIRECT and job number 207 in the organization block OB 1 and assign its parameters.

OB 1

```

0055      :
0056      :A   I 12.3           Trigger "deletion of
0057      :AN  F 11.6           S5F files on CP 581"
0058      :=   F 11.7
0059      :S   F 11.6
005A      :AN  I 12.3
005B      :R   F 11.6
005C      :
005D      :A   F 11.7
005E      :JU  FB 244           DHB SEND (SEND DIRECT)
005F  NAME :SEND
0060  SSNR :    KY 0,4           Interface 4
0061  A-NR :    KY 0,207        Job No. 207, delete S5F file
0062  ANZW :    FW 130
0063  QTYF :    KS DB           Directory and path data
0064  DBNR :    KY 0,14        in DB 14
0065  QANF :    KF +0           Initial address
0066  QLAE :    KF +7           Number of data words
0067  PAFE :    FY 134
0068      :
0069      :BE
    
```

- All S5F files present in the specified directory are deleted if the DHB SEND with the function SEND DIRECT and job number 207 is called. The complete directory is deleted if it only contains S5F files and no subdirectories. The current directory and files with the attribute "Read only" are not deleted (MS-DOS).
- Only use the delete function of the CPMASS program if you are sure that information you still require cannot be deleted by mistake.

Operations on the CP 581

- **Switch on the device and enter the base interface number in the start call of the CPDHB driver (AUTOEXEC.BAT file)**

Following installation of the system software, the AUTOEXEC.BAT file contains the calls for the CPDHB driver and the CPMASS program depending on the option you selected during the installation. The driver and CPMASS are automatically started in this case when the CP 581 is switched on. The directories C:\CPU1, C:\CPU2, C:\CPU3, and C:\CPU4 are generated in addition.

The driver call is entered without the "/ssnr" option; the base interface number consequently is set to the value entered in SETUP (default "0," see volume 1) when the driver is started. Add the base interface number to the driver call in the AUTOEXEC.BAT file. The call then is **CPDHB /ssnr4**.

Check whether the start call for CPMASS is still present in the AUTOEXEC.BAT file. If this is not the case, enter it again.

Now restart the system.

- The CPDHB driver and the CPMASS program are loaded into the main memory and started.
- The following message is output on the screen:

```
CPMASS Vx.x
Copyright (c) Siemens AG 1992
```

- The CPMASS program is executed in the background. Other programs can be started in the foreground, but the reaction times are influenced.

Operations on the S5 CPU

- **Start S5 program**

Start the S5 program by restarting the S5 CPU.

- OB 21 or 22 is executed depending on the restart mode with the S5-115U. The DHB SYNCHRON is activated, and the status word ANZW and the parameter error byte PAFE are evaluated. In our example, the PLC is set to STOP if the S5 CPU and the CP 581 cannot be synchronized.
- The mass storage functions are active if the synchronization is successful and wait for the trigger for data transmission from the S5 CPU.

12.3 Example “Command Interpreter”

12.3.1 Task/Problem

You wish to output S5 CPU messages on a message printer via the command interpreter of the CP 581 communications processor which is installed in the subrack of your S5-115U.

12.3.2 Starting to Solve the Problem

The process alarms are acquired by the PLC via digital inputs and evaluated by the user program.

The DHB SEND with the function SEND DIRECT is called when an alarm appears. A command line is stored in data block DB 15 for each alarm. The DHB SEND with the function SEND DIRECT transmits the corresponding command line to the CP. This is transferred from there to MS-DOS by the CPSHELL command interpreter. The operating system subsequently carries out the desired function.

A text file must be stored on the hard disk of the CP for each message and must contain the text to be output on the printer. All text files are present in the directory D:\MESS.

12.3.3 Structure of Solution

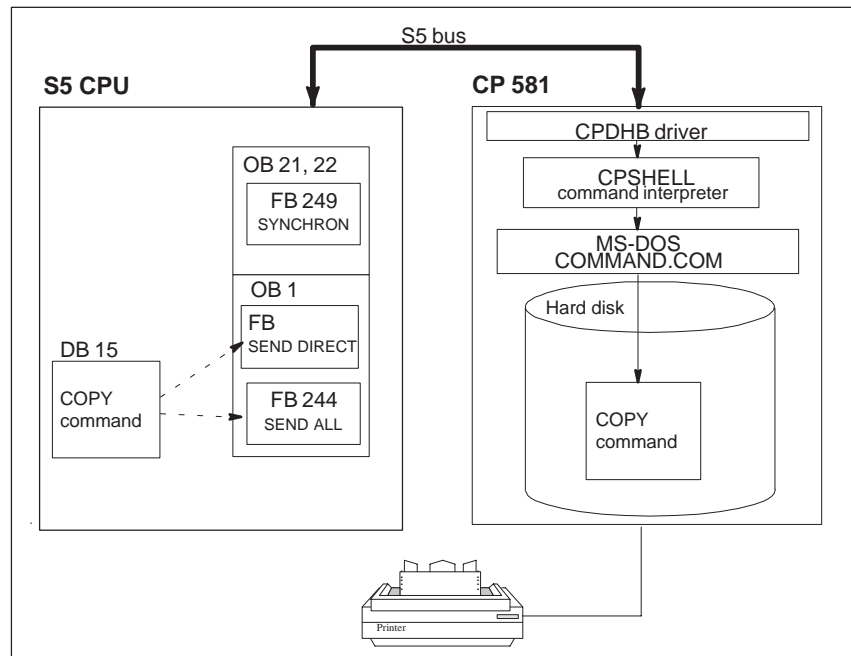


Figure 12-7 Data Transfer between S5 CPU and CP 581 when Using the Command Interpreter

- The FB SYNCHRON initializes the interface, and the frame size is agreed upon between the interface and the S5 CPU.
- The FB SEND with the function SEND DIRECT transfers the command line to CPSHELL.
- If the command line is longer than the set frame size. The FB SEND with the function SEND ALL transmits the remaining data upon a request from the CP.
- The CPSHELL program transfers the command line to MS-DOS (COMMAND.COM).
- COMMAND.COM handles command processing under MS-DOS.

For this purpose you must carry out the following measures on the PG, CP 581 and S5 CPU:

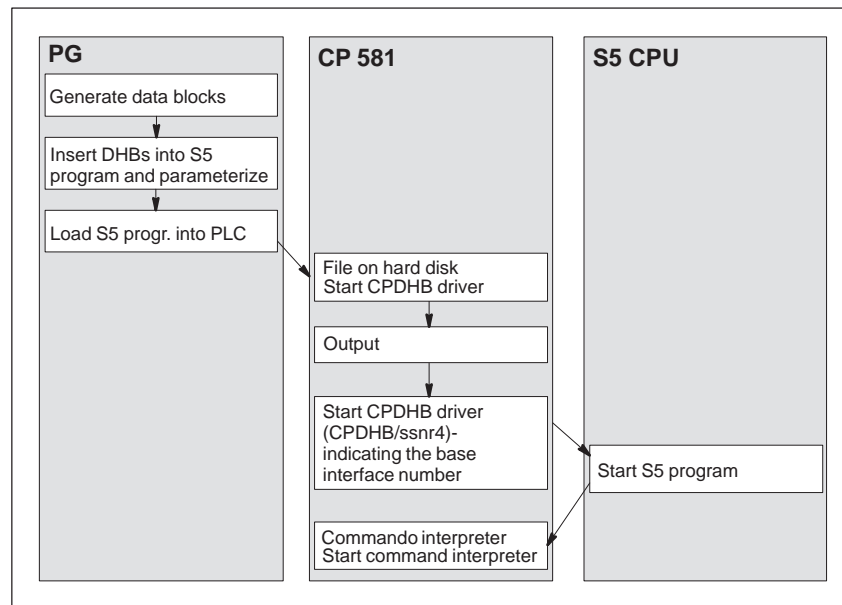


Figure 12-8 Handling Sequence when Using the Command Interpreter

12.3.4 Individual Working Steps

The operations you must carry out on the programmer (PG), the communications processor (CP) and the S5 CPU are explained in more detail on the following pages.

Operations on the PG

- **Generate data blocks**

Generate the data block DB 15 and enter a COPY command for every message.

DB 15

```
DB 15
0      : KH = 0000;
1      : KS = 'COPY D:\MESS\MESS1.TXT L';
13     : KS = 'PT1';
15     : KH = 0000;
16     : KS = 'COPY D:\MESS\MESS2.TXT L';
28     : KS = 'PT1';
30     : KH = 0000;
31     : KS = 'COPY D:\MESS\MESS3.TXT L';
43     : KS = 'PT1';
45     : KH = 0000;
      :
```

- The COPY command relevant in each case is transmitted from the data block DB 15 to MS-DOS (COMMAND.COM) via the data handling block SEND with the function SEND DIRECT.

- **Installation and parameterization of data handling blocks in S5 program**

Parameterize the DHB SYNCHRON and insert it into the restart organization blocks OB 21 and 22 (see Section 12.1.4 for listing).

- The S5 CPU and the CP 581 are synchronized in “RUN/STOP mode” (OB 21) and in “Automatic warm restart” (OB 22) of the PLC by means of the DHB SYNCHRON in the restart OB.

Call the data handling block SEND with the function SEND DIRECT in the organization block OB 1 and assign its parameters.

```

OB 1
  SEGMENT 1
    :
    :
0010    :A    I 13.0          Trigger message text MESS1
0011    :AN   F 150.0
0012    :=   F 150.1
0013    :S    F 150.0
0014    :AN   I 13.0
0015    :R    F 150.0
0016    :    +
0017    :A    F 150.1
0018    :JU   FB 244
0019  NAME :SEND          DHB SEND (SEND DIRECT)
001A  SSNR :    KY 0,4    Interface number
001B  A-NR :    KY 0,209  Job number
001C  ANZW :    FW 30    Status word
001D  QTYP :    KS DB    Source type data block
001E  DBNR :    KY 0,15  DB number
001F  QANF :    KF +1    Start of command line
0020  QLAE :    KF +15   Data length
0021  PAFE :    FY 34    Parametererror
    :
    :
    :BE

```

- The DHB SEND with the function SEND DIRECT transfers the command line to the command interpreter on the CP.

Call the data handling block SEND with the function SEND ALL in the organization block OB 1 and assign its parameters.

```

OB 1
  SEGMENT 1
    :
    :
0024      :JU   FB 244   DHB SEND (SEND ALL)
0025 NAME :SEND +
0026 SSNR :    KY 0,4
0027 A-NR :    KY 0,0
0028 ANZW :    FW 20
0029 QTYP :    KS NN
002A DBNR :    KY 0,0
002B QANF :    KF +0
002C QLAE :    KF +0
002D PAFE :    FY 13
    :
    :
    :BE
  
```

- The DHB SEND with the function SEND ALL is only required if the command line in the DHB SEND with the function SEND DIRECT is longer than the set frame size. In this case the DHB SEND with the function SEND ALL transmits the remaining data which could not be transmitted “directly.”

- **Loading the S5 program into the programmable controller**

Set the programmable controller to STOP and load the S5 program into the PLC.

Operations on the CP 581

- **Storage of message texts in files on hard disk**

Create a file with the corresponding message text for each message in the directory MESS on drive D:\ of the CP 581.

- The message text is output on the printer if a process alarm is present and if the COPY command required is transmitted from the command interpreter to MS-DOS.

- **Starting the CPDHB driver and setting the base interface number**

Following installation of the system software, the start call for the CPDHB driver is present in the AUTOEXEC.BAT file, but excluding the “ssnr option,” i. e. the base interface number consequently is set to the value entered in SETUP (default “0,” see volume 1) when the driver is started.

Enter the base interface number ‘4’ for the option “/ssnr<ddd>” in the start call in the AUTOEXEC.BAT file. Then, the start call is: **CPDHB /ssnr4**.

The driver is automatically started when the CP 581 is switched on. If you have deleted the start call from the AUTOEXEC.BAT file, you can also start the driver by entering the command **CPDHB /ssnr4**.

- The CPDHB driver is loaded into the main memory (TSR program).

Operations on the S5 CPU

Operations on the S5 CPU

- **Start S5 program**

Start the S5 program by restarting the S5 CPU.

- OB 21 or 22 is executed depending on the restart mode. The DHB SYNCHRON is activated, and the status word ANZW and the parameter error byte PAFE are evaluated. In our example, the PLC is set to STOP if the S5 CPU and the CP 581 cannot be synchronized.
- The command interpreter can be started if the synchronization is successful.

Operations on the CP 581

- **Starting the command interpreter**

Enter the **CPSHELL** command.

- The command interpreter is started. The following start message is output on the screen:

```
CPSHELL Vx.x  
Copyright (c) Siemens AG 1991  
CPSHELL ready
```

- Any S5 messages are output on the printer.

- **Terminating the command interpreter**

Press the <ESC> key.

- The CPSHELL program is terminated.

12.4 Application Examples in the Directory “EXAMPLE”

Once you have installed the CP 581 system software, you can find a collection of application examples in the **EXAMPLE** directory. The collection consists of text files which explain the task and execution of an example, sample files for configurations and format instructions, S5 files and source code files for MS-DOS applications (for example C source code).

The **READ.ME** file in the **EXAMPLE** directory provides you with a summary of the collection of examples and the subdirectories and files in which they are stored.

The directory **EXAMPLE** has the following basic structure (possibly with further subdirectories).

Directory **EXAMPLE**

- Subdirectory **CPRECORD**
×× for applications of CPRECORD
- Subdirectory **CPMASS**
×× for applications of CPMASS
- Subdirectory **CPSHELL**
×× for applications of CPSHELL
- Subdirectory **S5REMOTE**
× for applications of S5REMOTE/S5REMOTF
- Subdirectory **C**
×× for applications of free programming
×× in the programming language C
- Subdirectory **PASCAL**
×× for applications of free programming
×× in the programming language PASCAL
- File **READ.ME** with explanations
↳ on the collection of examples

Reference Section for System Software

13

This chapter provides you - mainly in tabular form - with a complete summary of the permissible parameters, the error bits and the error messages. The first section contains a summary of the data handling blocks and their parameterization as used with the CP 581 for the various functions. Furthermore this chapter provides you with information on the CPDHB driver and CP 581 applications.

Chapter Overview

Section	Description	Page
13.1	Data Handling Blocks	13-3
13.1.1	Summary of DHBs with the CP 581	13-3
13.1.2	DHB SYNCHRON	13-4
13.1.3	DHB SEND	13-5
13.1.4	DHB FETCH	13-6
13.1.5	DHB RECEIVE	13-6
13.1.6	DHB CONTROL	13-8
13.1.7	DHB RESET	13-8
13.1.8	Status Word	13-9
13.1.9	Parameter Assignment Error Bits	13-10
13.2	Options of the CPDHB Driver	13-11
13.2.1	Options in the Load Command/Setting the Base Interface Number	13-11
13.2.2	Options on the Installed Driver	13-11
13.3	Process Data Acquisition	13-12
13.3.1	Parameters for the Configuration File	13-12
13.3.2	Command Options of CPRECORD and CPRECCTL	13-14
13.3.3	Control of Process Data Acquisition from a CPU	13-16
13.3.4	Error and Status Messages of the CPDHB Driver and the CPRECORD Program	13-17
13.4	Mass Storage Functions	13-20
13.4.1	Data Handling Blocks for the Mass Storage Functions	13-20
13.4.2	Error Bits of CPMASS Program	13-21
13.5	Command Interpreter	13-22
13.5.1	Data Handling Blocks for the Command Interpreter	13-22
13.5.2	Error Bits of CPSHELL Program	13-23

Section	Description	Page
13.6	Virtual S5 Drive	13-24
13.6.1	Data Handling Blocks for the Virtual S5 Drive	13-24
13.6.2	Sequence Parameters for the Virtual S5 Drive	13-25
13.6.3	Format File	13-27
13.6.4	Error Status Bits	13-28
13.7	Free Programming	13-30
13.7.1	DHB Status Bits	13-30
13.7.2	Calling and Parameterizing the CPDHB Driver	13-33
13.7.3	Functions of the CPDHB Driver	13-36
13.7.4	Error Bits of the CPDHB Driver	13-37

13.1 Data Handling Blocks

This section provides you with a summary of the functions and parameters of the data handling blocks (DHB) used for the CP 581.

You can find detailed descriptions of the data handling blocks in the corresponding documents. These are available as individual descriptions for the S5-135U and S5-155U programmable controllers. The description of the DHB for the S5-115U programmable controller is included in the S5-115U manual.

13.1.1 Summary of DHBs with the CP 581

The following table provides a summary of the DHBs used with the CP 581.

Table 13-1 Summary of DHBs

DHB	Function	S5-115U	S5-135U	S5-155U
SYNCHRON	Synchronizes CP 581 and CPU	FB 249	FB 125	FB 125
SEND DIRECT	Activates the CP 581 and transfers to it the source address of the required data to be sent to the CP	FB 244	FB 120	FB 120
SEND ALL	Transfers the data from the CPU to the CP 581	FB 244	FB 120	FB 120
SEND-A ¹⁾	Special SEND ALL function block for S5-135U and S5-155U programmable controllers	-	FB 126	FB 126
FETCH	Activates the CP 581 and transfers to it the destination address of the required data to be read by the CP	FB 246	FB 122	FB 122
RECEIVE DIRECT	Data are transmitted from the CP 581 to the CPU with a specific job number	FB 245	FB 121	FB 121
RECEIVE ALL	Transfers the data from the CP 581 to the CPU	FB 245	FB 121	FB 121
REC-A ²⁾	Special RECEIVE ALL function block for S5-135U and S5-155U programmable controllers	-	FB 127	FB 127
CONTROL ALL	Indicates which job is currently being executed	FB 247	FB 123	FB 123
CONTROL DIRECT	Updates the status word	FB 247	FB 123	FB 123
RESET ALL	Completely resets the CPDHB driver	FB 248	FB 124	FB 124
RESET DIRECT	Resets the CPDHB driver only for one specific job	FB 248	FB 124	FB 124

1) The function block SEND-A differs from the function block SEND in that the parameters QTYP, DBNR, QANF and QLAE are omitted.

2) The function block REC-A differs from the function block RECEIVE in that the parameters ZTYP, DBNR, ZANF and ZLAE are omitted.

The following applies to both function blocks:

In cases where these parameters are irrelevant (for example, SEND ALL function, RECEIVE ALL function), use of these blocks saves memory space as well as writing and increases the program transparency.

Note

In the S5-115U programmable controller, the DHBs are completely present in the operating system of the CPU. In the S5-135U programmable controller (CPU 922, CPU 928 and CPU 928B), the code of the DHBs is also present in the operating system of the CPU. The block **headers** are delivered on floppy disks, however, and must be loaded into the CPU.

In the S5-155U programmable controller, the DHBs are delivered as a STEP 5 program on floppy disks and must be loaded into the CPU.

13.1.2 DHB SYNCHRON

The DHB SYNCHRON synchronizes the interface between the CPU and CP 581 with a cold restart (OB 20), a manual warm restart (OB 21) or an automatic warm restart following a power failure (OB 22). You must call the DHB SYNCHRON in the restart OB of the CPU for each interface of the CP 581.

Table 13-2 Format and Meaning of Parameters of DHB SYNCHRON

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
BLGR	D	KY	Frame size
PAFE	Q	BY	Parameter Assignment Error Bits

13.1.3 DHB SEND

The DHB SEND has two modes

- SEND DIRECT (job number \neq 0) and
- SEND ALL (job number = 0).

The DHB SEND DIRECT

You will need the DHB SEND DIRECT if, for example, you want to send data from the CPU to the CP 581 with the mass storage function. By calling the DHB SEND DIRECT you activate the CP 581 and transfer to it the address of the data to be sent by the CPU.

Table 13-3 Format and Meaning of Parameters of DHB SEND for the SEND DIRECT Function

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
QTYP	D	KS	Data type of source
DBNR	D	KY	Number of data block if source is data block
QANF	D	KF	Offset of first item of data to be read in the data area
QLAE	D	KF	Number of data units to be read (words or bytes)
PAFE	Q	BY	Parameter assignment error bits

The DHB SEND ALL

You will need the DHB SEND ALL, for example, to carry out the data transfer to the CP 581 with the mass storage function (CPMASS). With the CPRECORD program and the virtual S5 drive, you must call the DHB SEND ALL in each cycle of the CPU.

Table 13-4 Format and Meaning of Parameters of DHB SEND for the SEND ALL Function

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
QTYP	D	KS	These parameters are irrelevant with the function SEND ALL; they must be specified for format reasons, however.
DBNR	D	KY	
QANF	D	KF	
QLAE	D	KF	
PAFE	Q	BY	Parameter assignment error bits

13.1.4 DHB FETCH

You will need the DHB FETCH if, for example, you wish to transmit data from the CP 581 to the CPU with the mass storage function (CPMASS). By calling the DHB FETCH you activate the CP 581 and transfer to it the address of the data to be sent to the CPU.

Table 13-5 Format and Meaning of Parameters of DHB FETCH

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
QTYP	D	KS	Data type of destination
DBNR	D	KY	Number of data block if destination is data block
QANF	D	KF	Offset of first item of data to be written in the destination
QLAE	D	KF	Number of data units to be written (words or bytes)
PAFE	Q	BY	Parameter assignment error bits

13.1.5 DHB RECEIVE

The DHB RECEIVE has two modes

- RECEIVE DIRECT (job number \neq 0) and
- RECEIVE ALL (job number = 0).

The DHB RECEIVE DIRECT

You require the DHB RECEIVE with the function RECEIVE DIRECT to transmit data from the CP 581 to the CPU with a specific job number (application with "Free programming"). The direct mode is called, amongst others, in the cyclic part of the CPU program. The block can also be called during interrupt or alarm processing, but the status word is then not updated cyclically. This task must then be handled by the CONTROL block.

Table 13-6 Format and Meaning of Parameters of DHB RECEIVE for the RECEIVE DIRECT Function

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
ZTYP	D	KS	Data type of destination
DBNR	D	KY	Number of data block if destination is data block

Table 13-6 Format and Meaning of Parameters of DHB RECEIVE for the RECEIVE DIRECT Function

Parameter	Kind	Type	Meaning
ZANF	D	KF	Starting address of destination
ZLAE	D	KF	Number of data units to be read (words or bytes)
PAFE	Q	BY	Parameter assignment error bits

The DHB RECEIVE ALL

The DHB RECEIVE is required with the RECEIVE ALL function, for example, to carry out the data transfer from the CP 581 to the CPU with the mass storage function. With the virtual S5 drive, you must call the DHB RECEIVE ALL in each CPU cycle.

Table 13-7 Format and Meaning of Parameters of DHB RECEIVE for the RECEIVE ALL Function

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number
ANZW	I	W	Status word
ZTYP	D	KS	These parameters are irrelevant with the function RECEIVE ALL; they must be specified for format reasons, however.
DBNR	D	KY	
ZANF	D	KF	
ZLAE	D	KF	
PAFE	Q	BY	Parameter assignment error bits

13.1.6 DHB CONTROL

You can use the DHB CONTROL to scan status information of the interface. The block has two modes

- CONTROL ALL (job number = 0) and
- CONTROL DIRECT (job number \neq 0)

CONTROL ALL

The CONTROL ALL function indicates in the Low byte of the ANZW which job is currently being processed by the CP 581.

CONTROL DIRECT

A so-called job status exists in the interface for each job. This is managed by the interface and indicates, for example, whether a job is (still) running, whether it has been terminated without errors or terminated with a particular error. The CONTROL DIRECT function transfers the job status selected by the parameter A-NR into the status word.

Table 13-8 Format and Meaning of Parameters of DHB CONTROL

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number of job to be monitored
ANZW	I	W	Status word: contains result of scan
PAFE	Q	BY	Parameter assignment error: Error Status Bits

13.1.7 DHB RESET

The DHB RESET has two modes

- RESET ALL (job number = 0) and
- RESET DIRECT (job number \neq 0).

The RESET ALL function (job number = 0) resets all jobs of the CP 581 interface.

With the RESET DIRECT function (job number = \neq 0), only the specified job of the CP 581 interface is reset.

Table 13-9 Format and Meaning of Parameters of DHB RESET

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job number of job to be reset
PAFE	Q	BY	Parameter assignment error: Error Status Bits

13.1.8 Status Word

Information on the status of job processing is stored in the status word. You define the address of the status word when parameterizing. The information can then be read and evaluated from here.

The status word is part of a double word which is addressed by the parameter ANZW. The second part of the double word is the so-called “length word.”

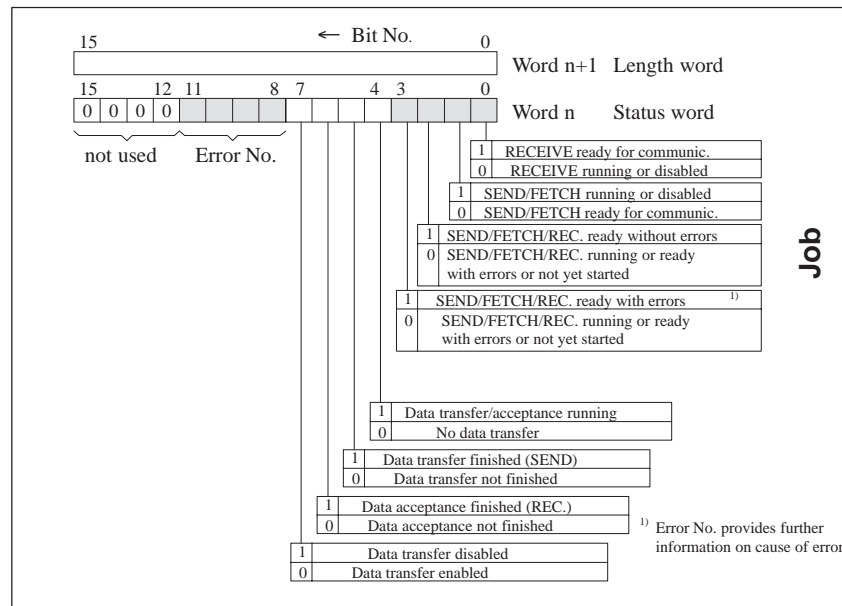


Figure 13-1 Format of DHB Status Words

Please note:

- Assign a new status word to each job used.
- If you must send two DHB calls in succession for a job (SEND - SEND ALL, FETCH/RECEIVE - RECEIVE ALL), individual status words must always be provided for each call since these are managed separately by the specified DHB.
- Following synchronization, the CPDHB driver initializes the status word with 0006H. This value is retained until an application (for example, CPMASS) logs on for a specific job number with the CPDHB driver. The status word is then 0 (with SEND) or 1 (with RECEIVE) until the first job has been processed.

Note

You cannot send jobs to the CP 581 for the respective job number as long as 0006H is present in the status word. Observe this response when restarting the S5 program.

13.1.9 Parameter Assignment Error Bits

All data handling blocks check the transferred parameters for syntax errors and logic errors when they are called. They also check the addressed interface to establish whether it is available for the triggered function.

Whether parameter assignment errors have occurred, and if so which ones, can be observed in the status byte whose address you have defined by means of the parameter PAFE when calling the DHB.

Table 2-1 shows a summary of the parameter assignment error bits.

Table 13-10 Parameter Assignment Error Bits

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format
21H	DB or DX data block not present or illegal (e.g. DB 0 or DX 0 with QTYP = DB or DX)
31H	Area too small or total of initial address (QANF/ZANF) and length (QLAE/ZLAE) too large (with all QTYP/ZTYP)
41H	Area does not exist or is illegal (with QTYP/ZTYP = AS, QA, IA, PY)
51H	Status word (address) faulty
61H	Dependent on CPU
71H	Interface does not exist
81H	Interface not ready
91H	Interface overloaded
A1H	Dependent on CPU
B1H	Job number illegal or frame size (SYNCHRON) illegal
C1H	Interface does not react, or interface does not react at correct time, or interface rejects job
D1H	Dependent on CPU
E1H	Dependent on CPU
F1H	Dependent on CPU

13.2 Options of the CPDHB Driver

Conventions:

The **CPDHB** command can either be written in lower-case or in upper-case letters; **options** must always be written in lower-case letters. You can use “/” and “-” as **option characters**. The “<>” characters exclusively serve to identify text parameters and must not be used when writing commands.

13.2.1 Options in the Load Command/Setting the Base Interface Number

CPDHB /ssnr<ddd> /noreset /resync/ norunstop /int<xx>

Table 13-11 Meaning of Options in the CPDHB Load Command

Option	Meaning
ssnr<ddd>	Used to set the base interface number (= number of the 1st page) for data transmission to the CPUs: The preset value (without “ssnr” option) in this case is ‘0.’ If you want to use a different base interface number, you must indicate the value you require. ddd = base interface number set in steps of four (0, 4, 8, etc.) up to 252 max.
noreset	The option prevents a system reset (“warm start”) after using the key combination CTRL-ALT-DEL; a reset can however be executed by pressing the RESET key.
resync	The driver recognizes any synchronization previously initiated by a CPU. A new synchronization of CP 581 by the CPU is then no longer necessary.
norunstop	The option suppresses the function of the RUN/STOP switch.
int<xx>	Change interrupt for driver call: If the preset interrupt INT-66H is already occupied by the software, you can change it via this option. xx = number of the new interrupt as hexadecimal number.

13.2.2 Options on the Installed Driver

Uninstalling Driver **CPDHB /u**

Using this command, you can abort the driver and remove it from the memory. If you have loaded further programs, you must first delete them from the memory in the reverse order to which they were installed, i.e. commencing with the program you installed last.

Listing Driver Options on the Screen **CPDHB /h or CPDHB /?**

13.3 Process Data Acquisition

13.3.1 Parameters for the Configuration File

Refer to Section LEERER MERKER for how to edit the configuration file supplied on the CP 581 according to the desired data transfer. You can store up to eight parameter sets in the configuration file where each parameter set contains the acquisition parameters for one S5 area.

The 1st line of the configuration file must contain the time for the acquisition cycle in seconds (max. 11 799 369 seconds). This is followed by the data for the parameter sets.

The following table shows you the meaning of the various parameters and the permissible values.

Table 13-12 Meaning of the Parameters in a Parameter Set

Line No.	Parameter	Meaning	Permissible values
2	CPU No.	Number of CPU corresponding to slot sequence	1 to 4
3	S5 area (QTYP)	Specification on data source on CPU: QA for output area AS for absolute address RS for RS word DB for data block DB DX for data block DX IA for input area FA for F flag area PY for I/O area TA for timer cell area CA for counter cell area	QA, AS, RS, DB, DX, IA, FA, PY, TA, CA
4	Block No.	Number of data block DB or DX on the CPU if the data source is a data block; this parameter has no significance for the other data (the parameter line must still be present, however!)	Dependent on CPU
5	Offset (QANF)	Number of 1st data unit to be read in the S5 area	Dependent on data type and CPU
6	Number (QLAE)	Number of data units (words or bytes) which are to be read from the S5 area starting at "Offset"	Dependent on data type and CPU
7	Destination path	Path name for ASCII files in which the data of the S5 area are to be stored	MS-DOS syntax
8	Extension	File name extension	ASCII characters
9	Number of files	Maximum number of ASCII files for process data recording from the S5 area	1 to 10 000
10	Number of data records	Maximum number of data records per ASCII file	1 to (2 ³¹ - 1)

Table 13-12 Meaning of the Parameters in a Parameter Set

Line No.	Parameter	Meaning	Permissible values
11	Format	<p>“Individual” conversion: Path and file name for format file or (only with DB/DX) for file with preheader data</p> <p>“Universal” conversion: KS for 2-character constant KF for fixed-point number KG for floating point number KH for hexadecimal number KM for bit pattern KT for timer value KC for counter value KY for 2-byte decimal number KB for 1-byte decimal number (only relevant for data areas IA, QA, PY and FA)</p>	<p>MS-DOS syntax, file name: “?????ST.S5D” or “*.FMT”; ? = letter/number or @, * = max. 8 letters/numbers</p> <p>KS, KF, KG, KH, KM, KT, KC, KY, KB</p>
12	Field delimiter	Characters by means of which the individual data of a data record are to be separated in the ASCII files (preset value: space)	ASCII characters
13	Acquisition mode	<p>0: recording is terminated once the defined file number has been reached.</p> <p>1: “permanent” recording: the oldest file is deleted when the defined file number has been reached and is then overwritten (similar to cyclic mode)</p>	0 and 1
14	Message mode	CPRECORD can store error messages in a logbook file: 0 = store no messages 1 = store messages	0 and 1
15	Logbook file	Path name for logbook file	MS-DOS syntax
16	Timeout	Maximum time for duration of a transmission procedure; the time is specified in seconds	1 to 3600

13.3.2 Command Options of CPRECORD and CPRECCTL

Options in the CPRECORD Command

The options are written like the command with upper-case or lower-case letters and preceded by the code “/” or “-”.

1. Command for starting CPRECORD (activation of process data acquisition):

CPRECORD /Cconfdat /? /H /N /S /O /A

Table 13-13 Meaning of Options in the CPRECORD Command

Option	Meaning (notation)
C	Switch over the configuration file to a file other than the preset CPRECORD.INI in the current directory. confdat = path and file name of the configuration file with which CPRECORD is to work (MS-DOS notation).
? or H	Output help text and explanations of command.
N	Switch off message output on screen during operation.
S	“Suspend” process data acquisition: Use this option if you wish to specifically control data acquisition from the individual S5 areas using the CP 581 or a CPU. Data acquisition is not started immediately in this case but only following the input of a special CPRECORD command or a job from the CPU (see Section 13.5).
O	Working with “old” configuration file: You must start CPRECORD using this option if you have already edited a configuration file for the first CPRECORD version (V1.0) and wish to use it further.
Q	Automatic mode: All acknowledgment requests are suppressed by CPRECORD (for example, if the memory on the destination device is insufficiently large when starting the acquisition). You can use the option if you wish to automatically start CPRECORD using a command entry in the AUTOEXEC.BAT file.

2. Command to deactivate process data acquisition and remove CPRECORD from the memory:

CPRECORD /U

Options in the CPRECCTL Command

Using the CPRECCTL command you can control the process data acquisition (suspended but not terminated) using the CP 581 to obtain information on the parameters of the configuration file and the acquisition status.

The options are written like the command with upper-case or lower-case letters and preceded by the code “/” or “-”:

CPRECCTL /B<p> /E<p> /Cp /Sp /? / H

Table 13-14 Meaning of the Options in the CPRECCTL Command

Option	Meaning (notation)
Bp B	Switching on of acquisition for parameter set(s) (B = begin) p = decimal number for a set, hexadecimal number for several sets ¹⁾ Switch on acquisition for all parameter sets
Ep I	Switching off of acquisition for parameter set(s) (E = end) p = decimal number for a set, hexadecimal number for several sets ¹⁾ Switch off acquisition for all parameter sets
C	Output configuration parameters for parameter set "p" ²⁾
S	Output current status for parameter set "p" ²⁾
? or H	Output help text and explanations on command

1) Hexadecimal number "0Xhh" with bit 0 = set No. 1, bit 1 = set No. 2 etc.

2) As decimal number

13.3.3 Control of Process Data Acquisition from a CPU

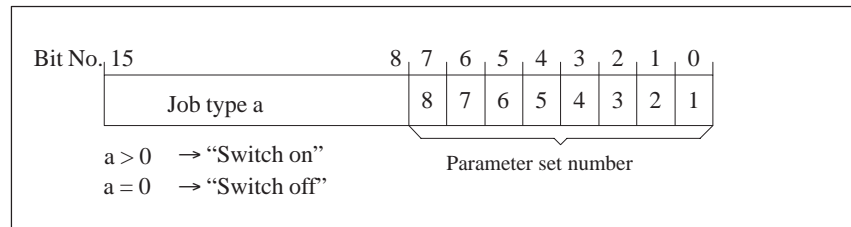
You can control the process data acquisition from a CPU by calling the data handling block DHB SEND with the function SEND DIRECT in the STEP 5 program:

(Refer to Section 7.5.2 for details).

Switch Acquisition On/Off for Specific Parameter Sets

Proceed as follows:

1. Process data word or flag word with control information (see figure).

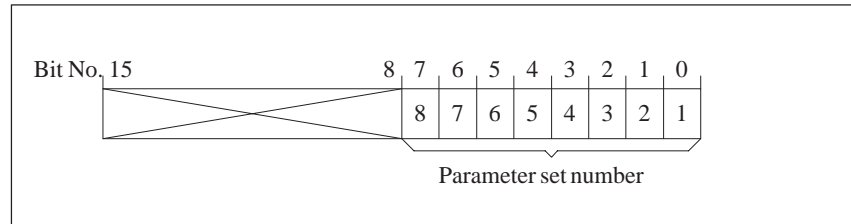


2. Call DHB SEND with job No. **210**

Trigger Acquisition for Specific Parameter Sets

The acquisition must **not** be activated with option "S" (suspended) and **must not have been terminated**.

1. Process data word or flag word with control information (see figure).



2. Call DHB SEND with job No. **211**.

13.3.4 Error and Status Messages of the CPDHB Driver and the CPRECORD Program

If special events are detected during process data acquisition, you can enter these events as messages in a logbook file. You must have entered the name of this logbook file in the configuration file. Please refer to Section 7.3.2.

Messages are sent by the CPDHB driver and the CPRECORD program. The two following tables show which messages can occur and what they mean.

The letter “p” present in square brackets in front of the messages indicates the number of the parameter set to which the message refers.

Error Messages of the CPDHB Driver

Table 13-15 Messages of the CPDHB Driver

Message
[p] [CPHTB] CPU not synchronized
[p] [CPHTB] Aborted - data area not accessible
[p] [CPHTB] Aborted by SYNCHRON
[p] [CPHTB] Aborted by RESET
[p] [CPHTB] Aborted by timeout
[p] [CPHTB] Aborted - parameter error
[p] [CPHTB] Parameter invalid (PAFE 1)
[p] [CPHTB] DB/DX does not exist (PAFE 2)
[p] [CPHTB] Area too small (PAFE 3)
[p] [CPHTB] Area does not exist (PAFE 4)
[p] [CPHTB] Status word error (PAFE 5)

Error Messages of the CPRECORD Program

Table 13-16 Messages of the CPRECORD Program

Message	Remarks
Unknown option	Incorrect option in CPRECORD command
Data acquisition suspended	With option “S”
Configuration file does not exist: <Config file>	<Config file> = name of configuration file
Error when opening configuration file <Config file>	MS-DOS output, <Config file> = name of configuration file

Table 13-16 Messages of the CPRECORD Program

Message	Remarks
CPRECORD.INI not found in current directory	File was deleted or current directory is incorrect
Configuration file extension is not ".ini"	Change name of configuration file
No parameter set has been found	
CPDHB driver not loaded	Load driver
CPRECORD file cannot be removed from memory	MS-DOS output
[p] Data recording started <Date> <Time>	Message at beginning of process data acquisition
[p] Data recording finished <Date> <Time>	Message at end of process data acquisition
[p] Error opening output file	MS-DOS output
[p] Format error in configuration file with parameter <n>	<n> = line number of faulty parameter
[p] Invalid output drive	MS-DOS output
[p] Insufficient drive capacity (<Actual value> <Required value>)	MS-DOS output
[p] S5D file: could not be opened	MS-DOS output
[p] S5D file: read error	MS-DOS output
[p] S5D file: not a root directory	Format error
[p] S5D file: too many subdirectories (> 128)	
[p] S5D file: no DV subdirectories	
[p] S5D file: positioning error	MS-DOS output "Seek Error"
[p] S5D file: too many data elements (> 128)	
[p] S5D file: the configured DV block is missing	
[p] S5D file: wrong block ID in DV preheader <Code>	Format error, <Code> = incorrect block code
[p] S5D file: wrong block number in DV preheader (<Number>)	Format error, <Number> = incorrect block number
[p] S5D file: DV preheader address too large (> 4095)	Format error
[p] S5D file: Format error, 1st format = "empty"	
[p] Output drive full	MS-DOS output
[p] Error when writing output file	MS-DOS output
[p] Output directory not found	MS-DOS output
[p] Configuration file: unknown S5 area	

Table 13-16 Messages of the CPRECORD Program

Message	Remarks
[p] Insufficient number of available ¹⁾ DOS clusters (<Actual value> <Required value>)	
[p] FMT file: could not be opened <FMT file>	MS-DOS output <FMT file> = name of format file
[p] S5 FMT file: unknown format in line <n> (<Format>)	<n> = line No. <Format> = incorrect format
[p] S5 FMT file: format error in line <n> (<Format>)	<n> = line No. <Format> = incorrect format
[p] S5 FMT file: too long in line <n> onwards (<Format>)	<n> = line No. <Format> = incorrect format
[p] S5D file format is permissible only for S5 area DB or DX	
[p] Error when opening log file <Log file>	<Log file> = name of logbook file
[p] Warning: cycle overflow > 10 %	
[p] Parameter set does not exist	

1) Cluster = contiguous logical memory area on drive

13.4 Mass storage functions

13.4.1 Data handling blocks for the mass storage functions

Use the DHBs SEND DIRECT and FETCH to trigger data transfer between the CP 581 and CPU.

The actual transfer of the useful data is carried out using SEND ALL and RECEIVE ALL.

You can additionally use the DHBs CONTROL and RESET.

Table 13-17 Data Handling Blocks for the Mass Storage Functions

Data handling block	A-NR	Function
SEND	0	Transmission of useful data (SEND ALL) CPU → CP
	1..99	Trigger for writing a data area (-> transmit address of useful data to CP)
	201	Switch over or create destination directory with CP
	207	Delete S5F files from a directory (the directory itself is subsequently deleted if it does not contain any other files or subdirectories)
RECEIVE	0	Transmission of useful data (RECEIVE ALL) CP → CPU
CONTROL	0	Indicates which job is currently being executed
	A-NR	Updates the status word
RESET	0	Completely reset CPDHB driver
	A-NR	Reset CPDHB driver only for one job
FETCH	1..99	Trigger for reading a data area (-> transmit address of useful data to CP)
SYNCHRON	-	Synchronize CP and CPU

13.4.2 Error Bits of CPMASS Program

If the CPMASS program determines when processing a job from a CPU that the job cannot be handled without errors, it sets error bits in the first of the two words of the CPU addressed by ANZW (bit nos. 8 to 11).

The following table shows you the error bits which can occur when executing the mass storage functions.

Table 13-18 Error Status Bits of the CPMASS Program

Value of bit No. 8 to bit No. 11	Cause of error
0H	No error
1H to 5H	DHB error (also referred to as PLC or CPU errors), error numbers PAFE 1 to PAFE 5
6H	CPDHB error
7H	S5 segment disabled (RS, PY)
8H	QLAE/ZLAE too large
9H	CP 581 is write-protected (transmit job from CPU)
AH	Insufficient space on the set CP 581 drive (transmit order from CPU)
BH	File/directory not found on CP 581 (read job, job "Set directory/delete S5F files" from CPU)
CH	Path name has incorrect syntax (job "Set directory/delete S5F files" from CPU)
DH	Directory was not completely deleted since it contains subdirectories or non-deletable files or external files (not "*.S5F") or is the current directory (job "Delete S5F files" from CPU)
EH	Timeout (after 30 seconds)
FH	Parameter assignment error

13.5 Command Interpreter

13.5.1 Data Handling Blocks for the Command Interpreter

If you wish to transmit a command from a preset data block to the CP 581, you must call the DHB SEND twice in the STEP 5 program, once with the function SEND DIRECT and then with the function SEND ALL.

Table 13-19 Data Handling Blocks for the Command Interpreter

Data handling block	A-NR	Function
SYNCHRON	-	Synchronize CP and CPU
SEND	209	Transfer of address of command to be sent from the CPU to the CP 581 (SEND DIRECT)
	0	SEND ALL function for triggering transmission of the command to the CP 581 (SEND ALL)

13.5.2 Error Bits of CPSHELL Program

If the CPSHELL command interpreter determines when processing a command from a CPU that the command cannot be handled without errors, it sets error bits in the first status word of the two words of the CPU addressed by ANZW (bit nos. 9 to 11).

The following table shows you the error bits which can occur when processing a command.

Table 13-20 Error Bits of the CPSHELL Command Interpreter

Value of bit No. 8 to bit No. 11	Cause of error
0H	No error
1H to 5H	DHB error (also referred to as PLC or CPU errors), error numbers PAFE 1 to PAFE 5
6H	Command line is too long
9H	The called program cannot be loaded since there is insufficient memory space in the CP 581
AH	The called MS-DOS function or the started program addresses a peripheral device of the CP 581 which is not ready, for example, there is no floppy disk in the drive, or the printer is not ready

Note

The error bits only provide information on whether the data exchange for transfer of the commands to the CP 581 was carried out correctly. You cannot absolutely conclude, however, that **the command, too**, was carried out correctly.

13.6 Virtual S5 Drive

13.6.1 Data Handling Blocks for the Virtual S5 Drive

The DHBs SEND ALL and RECEIVE ALL must be called cyclically on the CPU for data exchange between the CP 581 and the CPU via the virtual S5 drive.

Table 13-21 Data Handling Blocks for the Virtual S5 Drive

Data handling block	A-NR	Function
SEND	0	Transmission of DB/DX data (SEND ALL) CPU → CP
RECEIVE	0	Transmission of DB/DX data (RECEIVE ALL) CP → CPU
SYNCHRON	-	Synchronize CP and CPU

13.6.2 Sequence Parameters for the Virtual Drive

- Start program for binary data exchange:

S5REMOTE LW /N /T /C /U /H /?

- Start program for formatted data exchange (ASCII representation on CP):

S5REMOTF LW /N /T /F /C /U /H /?

Parameter LW:

Logical name for the virtual S5 drive. The drive name must be a permissible MS-DOS drive (“Letter:”) and must not be higher in the alphabet than the name set for “lastdrive” in the CONFIG.SYS file.

The parameters N, T, F, U, H and ? are optional. You can also use “-” for the option code instead of “/”, and also lower-case letters for the option instead of upper-case letters.

Table 13-22 Meaning of the Optional Parameters in the Command “S5REMOTE/S5REMOTF”

Option	Meaning (notation)
N	No error messages are output on the monitor.
T	Timeout in seconds. Values between 1 and 3640 are permissible; the default setting is 30 seconds.
F	Format “F Format CPU No. DB/DX No. [Field delimiter]” Data exchange is also in binary mode in this program if this option is missing in the S5REMOTF command.
C	Deletion of DB/DX is rejected.
U	The S5REMOTE/S5REMOTF program is uninstalled.
H/?	Output of explanations on S5REMOTE/S5REMOTF.

The components of the F option are separated from one another by spaces and have the following meaning:

Table 13-23 Components of the F Option

Parameter	Meaning	Permissible values
Format	Path and name of file with format specification	MS-DOS syntax, file name for data block: DB001 to DB255 for DB ¹⁾ DX000 to DX255 for DX
CPU No.	Number of CPU with which the S5 data to be converted are to be exchanged	1 to 4 for a particular CPU or * for all connected CPUs

Table 13-23 Components of the F Option

Parameter	Meaning	Permissible values
DB-/DX No.	Number of data block on CPU which is to be converted according to the specification	1 to 255 for DB 1 to DB 255 ¹⁾ 256 to 511 for DX 0 to DX 255 or * for all DBs and DXs
Field delimiter	The individual S5 data in the converted ASCII files on the CP 581 are separated from one another by the field delimiter (see Section 10.5.2). The component is optional; if it is not specified, the S5 data are separated by a space.	ASCII characters; numbers and the characters - / < > + . are illegal

1) The data blocks which you can read or write depend on the CPU (system data blocks).

13.6.3 Format File

- Name of the format file and directory:

The name of the format file can be selected as desired according to the MS-DOS standard.

- **Structure** of the format file:

A format file consists of one or more text lines (ASCII characters). Each line can consist of one or more format instructions or a comment. It must be terminated by CR and LF.

- **Syntax** of a format instruction:

A format instruction has the following syntax:

Repetition factor (optional)	Format data	Delimiter
---------------------------------	-------------	-----------

- Repetition factor:

Specifies how many successive items of data are to be converted with the same data format. If it is not specified, the subsequent format instruction is implicitly assigned the repetition factor “1.” Since two characters are always stored under the S5 format KS, the repetition factor with the format instruction KS always specifies twice the number of characters to be processed.

Permissible values: 1 to 4091

- Format data:

Permissible values: KC for S5 format KC,
KF for S5 format KF,
KG for S5 format KG,
KH for S5 format KH,
KM for S5 format KM,
KS for S5 format KS,
KT for S5 format KT,
KY for S5 format KY.

- Delimiter:

Permissible values: (Character/ASCII decimal equivalent):
Space / 32,
Comma / 44,
Horizontal tab. / 9,
CR + LF/13 + 10.

- **Comment line:**

A line is not interpreted if it commences with the character “;”.

13.6.4 Error Status Bits

Extended error code of MS-DOS

The codes listed in the following table may occur as MS-DOS extended error codes in the case of MS-DOS file functions which address the virtual S5 drive (scanning via MS-DOS function **59h** or C library function **dosexterr** ()), result: pointer to “extended error code”).

Pointer Û	Word 0	Extended error code	
	Word 1	Error class	Suggested action
	Word 2	Error locus	

Table 13-24 Extended Error Codes of MS-DOS

Extended error code	Cause of error
2	The file was not found
3	The path was not found
5	Access rejected
18	No further files
80	Directory cannot be created
160	Format error (when writing) ¹⁾
174	DB/DX do not exist
195	Aborted by timeout
201	Aborted by RESET
202	Aborted by SYNCHRON
205	Aborted by negative acknowledgment
254	CPU is not synchronous

1) Word address of the faulty format instruction in error class (High byte) and suggested action (Low byte)

Table 13-25 Error Messages of S5REMOTE/S5REMOTF

Error Messages of S5REMOTE/S5REMOTF

Error message	Remarks
DOS version must be 3.10 or higher	
Drive letter too high	Remedy: increase “lastdrive” in CONFIG.SYS
Drive already exists	Select other letter for virtual drive
Unknown S5 format in line x	Error in format file
Format list overflow in line x	The format file contains formats for more than 4091 data words

Table 13-25 Error Messages of S5REMOTE/S5REMOTF

Error message	Remarks
Unknown option	
CPDHB driver not found	Remedy: Load driver
CPU x not synchronous	Remedy: carry out restart on CPU and call DHB SYNCHRON
<cpu> must be 1 .. 4 or '*'	
<db> must be 1 .. 511 or '*'	
Illegal timeout	The value for the timeout (sec.) is not in the range from 1 to 3640
*** CPU x: Error y	y is an error bit of the CPDHB driver see Section 13.7.4.

13.7 Free Programming

13.7.1 DHB Status Bits

Meaning of Status Bits (Bits 0 to 7) in the DHB Status Word

The status bits in the status word provide information on the execution of a DHB call. They also serve as input information for the DHB itself and then influence its execution.

Table 13-26 Meaning of Error Bits in the Status Word

Bit No.	Set	Delete/ overwrite	DHB evaluation	User evaluation
0	DHB: CP application signals “data ready”	DHB: CP/HB driver signals “start of communication”	RECEIVE With bit No. 0= 1 communication with the CP is started if RLO = 1	Scan whether data present for RECEIVE
1	DHB: CPDHB driver signals “start of communication”	DHB: CP application signals “job processed”	SEND/FETCH With bit No. 1= 0 communication with the CP is started if RLO = 1	Scan whether job being executed
2	DHB: The CP application signals “job finished without error”	DHB: If job triggered again	No	Scan whether job finished without error
3	DHB: The CP application signals “job finished with error”	DHB: If job triggered again	No	Scan whether job finished with error; more information on the cause of the error is contained in bit nos. 8 to 11 (error No.)
4	SEND/RECEIVE If data transfer for a job has been started	SEND/RECEIVE If data transfer for a job has been completed	No	Scan whether data are currently being transmitted
5	SEND If data transfer to the CP has been terminated	SEND If data transfer for a new job has commenced	No	Scan whether the data block of a new job has already been transferred to the PLC and when a new data record can be provided for a current job
6	RECEIVE If the transfer of data has been terminated for a job	RECEIVE If the data transfer for a new job has commenced	No	Scan whether the data block of the current job has already been transmitted to the CPU
7	User: Disable access to data area	User: Enable access to data area	SEND/RECEIVE No data transfer takes place with bit No. 7 = 1 an error message is output to the CP	No

1) Bit nos. 8 to 11 (error nos.) contain more information on the cause of errors.

Meaning of Error Nos. (Bits 8 to 11) in the DHB Status Word

Table 13-27 Meaning of Error Numbers

Error No.	Meaning
0000	No error
0001..0101 1..5	DHB error (also referred to as PLC or CPU error), error number as in status byte "Parameter assignment error" (PAFE)
0110..1111 6..F	CP error You can use these error numbers for your application.

Status Codes of the DHB Jobs

Table 13-28 Status Codes for SEND/FETCH Jobs

Status codes for SEND/FETCH			
Code	Meaning	Error No. 1)	Status code
04H	Job finished without errors	0 0 0 0	0 1 0 0
08H	Job finished with DHB error	0 n n n	1 0 0 0
68H	Job finished with CP error 6	0 1 1 0	1 0 0 0
78H	Job finished with CP error 7	0 1 1 1	1 0 0 0
88H	Job finished with CP error 8	1 0 0 0	1 0 0 0
98H	Job finished with CP error 9	1 0 0 1	1 0 0 0
A8H	Job finished with CP error 10	1 0 1 0	1 0 0 0
B8H	Job finished with CP error 11	1 0 1 1	1 0 0 0
C8H	Job finished with CP error 12	1 1 0 0	1 0 0 0
D8H	Job finished with CP error 13	1 1 0 1	1 0 0 0
E8H	Job finished with CP error 14	1 1 1 0	1 0 0 0
F8H	Job finished with CP error 15	1 1 1 1	1 0 0 0

1) nnn = DHB errors 1 to 5

Table 13-29 Status Codes for RECEIVE

Status codes for RECEIVE			
Code	Meaning	Error No. 1) 2)	Status code 2)
06H	Job finished without errors	0 0 0 0	0 1 1 0
0AH	Job finished with DHB error	0 n n n	1 0 1 0
6AH	Job finished with CP error 6	0 1 1 0	1 0 1 0
7AH	Job finished with CP error 7	0 1 1 1	1 0 1 0
8AH	Job finished with CP error 8	1 0 0 0	1 0 0 0

Table 13-29 Status Codes for RECEIVE

Code	Meaning	Error No. 1) 2)	Status code 2)
9AH	Job finished with CP error 9	1 0 0 1	1 0 1 0
AAH	Job finished with CP error 10	1 0 1 0	1 0 1 0
BAH	Job finished with CP error 11	1 0 1 1	1 0 1 0
CAH	Job finished with CP error 12	1 1 0 0	1 0 1 0
DAH	Job finished with CP error 13	1 1 0 1	1 0 1 0
EAH	Job finished with CP error 14	1 1 1 0	1 0 1 0
FAH	Job finished with CP error 15	1 1 1 1	1 0 1 0
03H	RECEIVE ready	q q q q	q q 1 1

1) nnn = DHB errors 1 to 5

2) qq = original contents are retained

13.7.2 Calling and Parameterizing the CPDHB Driver

Principle of Driver Call

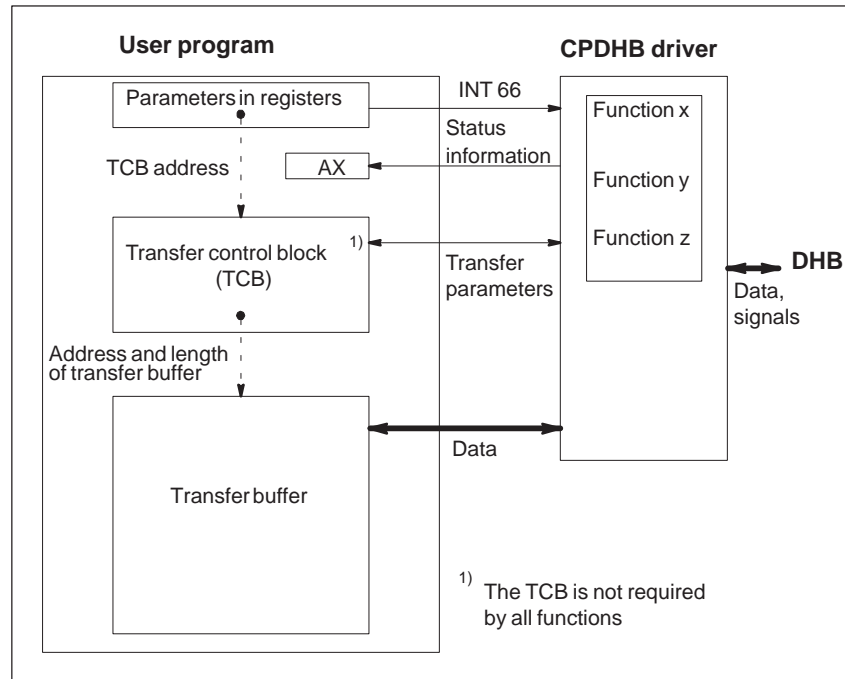


Figure 13-2 Parameters and Resources which must be provided by User Program for Driver Calls

8086 Register Set Call: Register AX = Number of desired function

The registers BX, CX, DX and ES are used specific to the function (you must store the offset in BX and the segment address of the TCB in ES for all transferred jobs)

Result: Condition codes in register AX

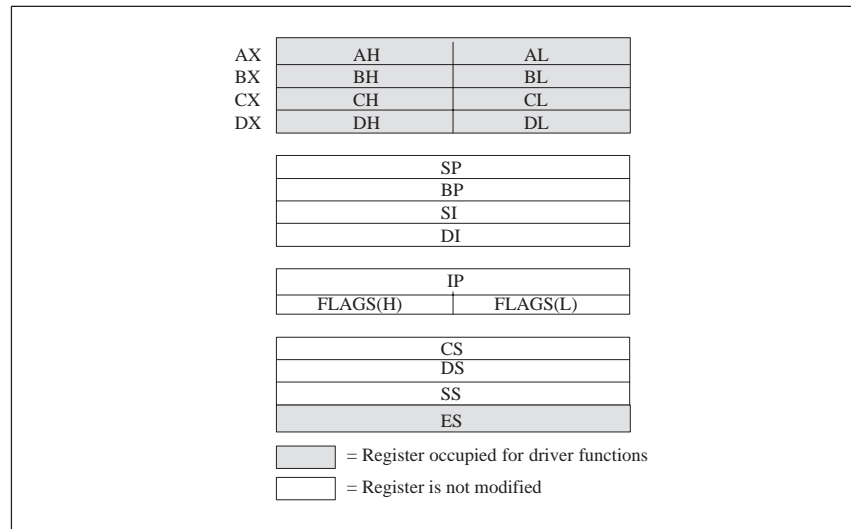


Figure 13-3 Register Set for Driver Call

Structure and Parameterization of Transfer Control Block

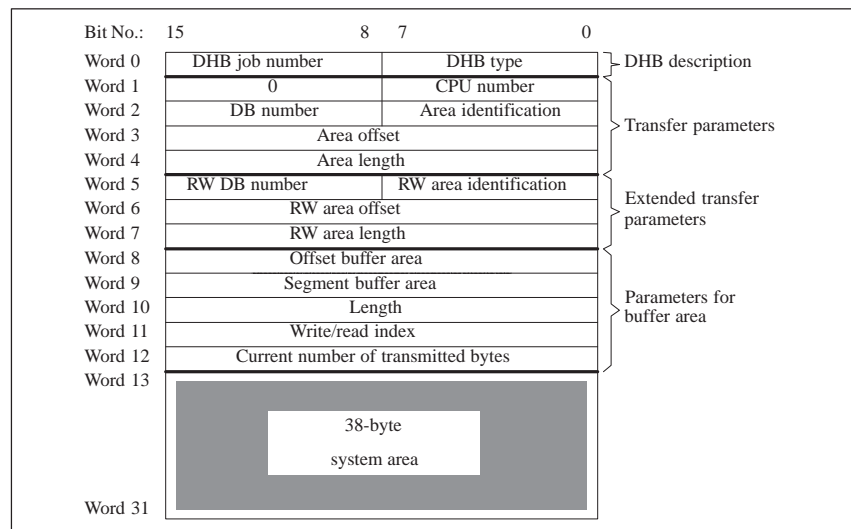


Figure 13-4 Structure of Transfer Control Block (TCB)

Table 13-30 Permissible Codes of DHB Type

Permissible DHB type coding	Meaning	
01H	SEND DIRECT	Only data are transmitted; transmission parameters are only transferred if more than one frame must be transmitted (i. e. if an additional call SEND ALL is required).
41H	SEND DIRECT/FETCH	Only transmission parameters are transferred; data must be transmitted with an additional call SEND ALL or RECEIVE ALL.
02H	RECEIVE DIRECT	

Table 13-31 Assignment of QTYP/ZTYP to the Area Identifications

QTYP/ZTYP	S5 area	Area identification
QA	Process output image	04H
AS	Absolute memory addresses ¹⁾	09H
RS	System data area	08H
DB	Data blocks DB	01H
DX	Extended data blocks DX	0AH
IA	Process input image	03H
FA	F flag area	02H
PY	I/O modules	05H
TA	Timer cells	07H
CA	Counter cells	06H

- 1) With the CPU 946/947, the parameter DBNR is used with absolute memory addresses (AS) to specify the addresses 2^{16} to 2^{19} .

13.7.3 Functions of the CPDHB Driver

Table 13-32 Functions and Function Numbers of the CPDHB Driver

Function	Functions No.	With/without direct job
Log on reception of job	11H	With
Log off reception of job	12H	With
Set job status	13H	With
Receive data used	14H	With
Send data used	15H	With
Scan TCB status	16H	With
Global scan of DHB job reception	17H	With
Delete background job	18H	With
Read S5 data area	01H	Without
Write S5 data area	02H	Without
Scan driver status	30H	--

13.7.4 Error Bits of the CPDHB Driver

The CPDHB driver stores status bits in the 8086 register AX when the calling program is continued. These indicators consist of a decimal number with sign and provide you with information on the execution of the called function:

- Values ≥ 0 : function was terminated successfully,
- Values < 0 : function was aborted.

A negative execution of a function does not always mean that a serious error has occurred.

The negative values are divided into three categories:

- -1 to -49:
errors which occur immediately following calling of the function, such as format errors in the TCB,
- -50 to -79:
errors detected by the CPDHB driver during transmission, such as unexpected CPU behaviour,
- -80 to -89:
errors signaled by the DHBs during transmission, such as parameter with incorrect format (PAFE error 1).

Note

The ranges listed are not used completely.

Table 13-33 Error Bits of the CPDHB Driver/Error Group -1 to -49

Error bit in AX		Meaning
decimal	hexadec.	
-1	FFFFH	Unknown function number: The function number specified in AX is not assigned to a function.
-2	FFFEH	CPU not synchronized: The function cannot be executed since the CPDHB driver is not synchronized with the desired CPU.
-3	FFFDH	Wrong CPU number: The CPU number specified in the TCB has an incorrect format.
-4	FFFCH	TCB busy: The specified TCB was previously set to a status by another function which does then not permit processing with the desired function.
-5	FFFBH	TCB is not logged on: The specified TCB was not registered with the CPDHB driver; the desired function cannot be carried out.
-6	FFFAH	Wrong DHB type: The DHB type code specified in the TCB has the incorrect format. The format of the DHB type is checked when registering a TCB.
-7	FFF9H	TCB already logged on: The TCB is already in the management of the CPDHB driver and cannot be registered. This error message can only occur when registering a TCB.
-8	FFF8H	TCB unknown: The CPDHB driver has determined by means of the TCB status that the TCB is in its management. The TCB address does not agree with the stored address, however. This error message can only occur when logging off the TCB.
-9	FFF7H	Job number too high: The job number specified in the TCB is too large. The job number is only checked when logging on the TCB.
-10	FFF6H	Job number already in use: The job number specified in the TCB is already occupied by another TCB.

To complete the information, the following table also contains error bits which really should not occur. If one of these bits (-58 to -62) occurs nevertheless, please contact your Siemens representative.

Table 13-34 Error Bits of the CPDHB Driver/Error Group -50 to -79

Error bit in AX		Meaning
decimal	hexadec.	
-51	FFCDH	Aborted - negative acknowledgment: Communication was commenced with the specified TCB but was aborted by the DHB
-52	FFCCH	Aborted - data area not accessible: The bit "Data area disabled" (bit No. 7) is set in the ANZW.
-53	FFCBH	Buffer too small: The buffer specified in the TCB is too small for the length parameterized in the DHB.
-54	FFCAH	Aborted by SYNCHRON: The specified TCB was removed from the management of the CPDHB driver by a SYNCHRON.
-55	FFC9H	Aborted by RESET: The specified TCB was removed from the management of the CPDHB driver by a RESET.
-56	FFC8H	Unexpected response with FETCH: The FETCH block has not reacted as expected by the CPDHB driver.
-57	FFC7H	Unexpected response with RECEIVE DIRECT: The RECEIVE DIRECT block has not reacted as expected by the CPDHB driver.
-58	FFC6H	Unexpected response with RECEIVE ALL: The RECEIVE ALL block has not reacted as expected by the CPDHB driver.
-59	FFC5H	Unexpected response with SEND DIRECT: The SEND DIRECT block has not reacted as expected by the CPDHB driver.
-60	FFC4H	Unexpected response with SEND ALL: The SEND ALL block has not reacted as expected by the CPDHB driver.
-61	FFC3H	Aborted by timeout: The DHB SEND ALL or RECEIVE ALL does not log on.

The following errors are so-called DHB errors and are detected by the DHB on the CPU. The reason for these errors is either incorrect parameterization of the DHB on the CPU or faulty assignment of transmission parameters in the TCB.

Table 13-35 Error Bits of the CPDHB Driver/Error Group -80 to -89

Error bit in AX		Meaning
decimal	hexadec.	
-80	FFB0H	Aborted - parameter assignment error: This error corresponds to "PAFE 0."
-81	FFAFH	Parameter invalid: The specified source/destination parameters (transmission parameters) are faulty. - This error corresponds to "PAFE 1."
-82	FFAEH	DB/DX does not exist: The specified data block DB or DX does not exist or the number is illegal. - This error corresponds to "PAFE 2."
-83	FFADH	Area too small: The specified area is too small or the total of the initial address and the length is too large. - This error corresponds to "PAFE 3."
-84	FFACH	Area does not exist: The specified area does not exist or is illegal. This error corresponds to "PAFE 4."
-85	FFABH	Status word error: The address of the status word is within an unknown S5 area or outside a permissible area. - This error corresponds to "PAFE 5."

List of Abbreviations

A

BIOS	Basic Input Output System
CP	Communications processor
CPDHB driver	Driver for data handling blocks
CPU	S5 CPU
DMA	Direct memory access
DPR	Dual-port RAM
EMC	Electromagnetic compatibility
FD	Floppy disk
HD	Hard disk
DHB	Data handling block
INTxx	Software interrupt interface for the CPDHB driver
IRQxx	Hardware interrupt
KBD	Keyboard
PG	Programmer
TCB	Transfer control block
TSR program	Terminate and stay-resident program

Guidelines for Handling Electrostatic Sensitive Devices (ESD)

B

Chapter Overview

Section	Description	Page
B.1	What is ESD?	B-2
B.2	Electrostatic Charging of Persons	B-3
B.3	General Protective Measures Against Electrostatic Discharge Damage	B-4

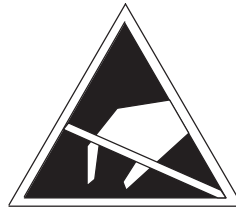
B.1 What is ESD?

Definition

All electronic boards are equipped with highly integrated modules and components. Due to their design, these electronic elements are very sensitive to overvoltage and thus to any electrostatic discharge.

The **E**lectrostatic **S**ensitive **D**evices are commonly referred to by internationally accepted abbreviation **ESD**.

Electrostatic sensitive devices are labeled with the following symbol:



Caution

Electrostatic sensitive devices are subject to voltages that are far below the voltage values that can still be perceived by human beings. These voltages are present if you touch a component or the electrical connections of a module without previously being electrostatically discharged. In most cases, the damage caused to a board by an overvoltage is not immediately noticeable and results in total damage only after a prolonged period of operation.

B.2 Electrostatic Charging of Persons

Charging

Any person with a non-conductive connection to the electrical potential of its surroundings can be charged electrostatically.

Figure B-1 shows you the maximum values for electrostatic voltages which can build up on a person coming into contact with the materials indicated in the figure. These values conform with the specifications of IEC 801-2.

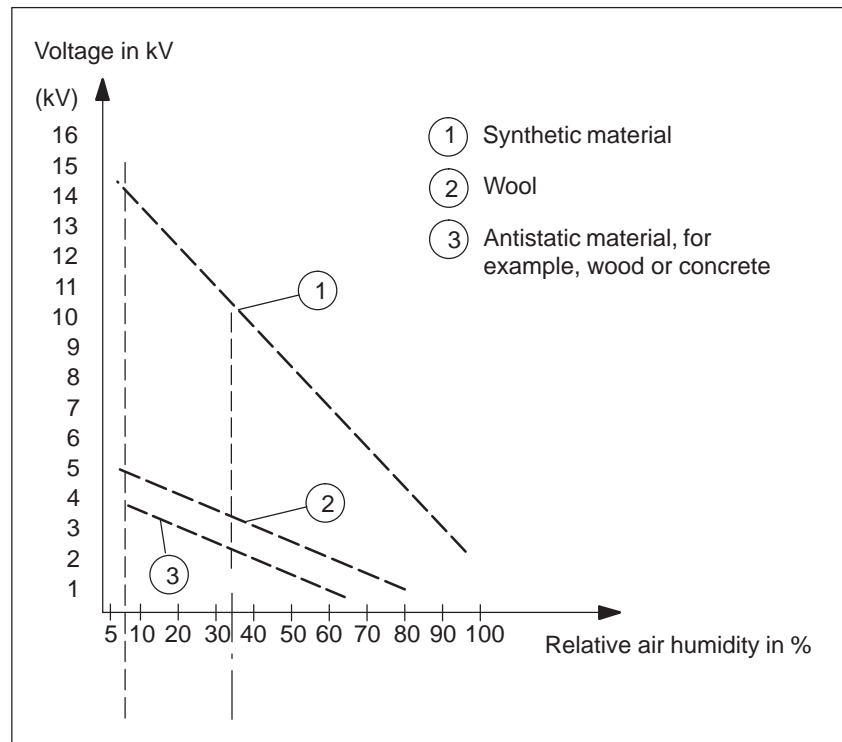


Figure B-1 Electrostatic Voltages which can build up on a Person

B.3 General Protective Measures Against Electrostatic Discharge Damage

Ensure Sufficient Grounding

Make sure that the personnel, working surfaces, and packaging are sufficiently grounded when handling electrostatic sensitive devices. You thus avoid electrostatic charging.

Avoid Direct Contact

Do not touch electrostatic sensitive devices unless it is unavoidable (for example during maintenance work). Hold modules without touching the pins of components or printed conductors. In this way, the discharged energy cannot affect the sensitive devices.

If you have to carry out measurements on a module, you must discharge your body before you start the measurement by touching grounded metallic parts. Use grounded measuring devices only.

Index

A

A-NR, 7-16, 7-35, 7-36, 8-11, 8-12, 8-13, 8-14, 8-16, 8-18, 9-10, 9-11, 10-13, 10-14, 11-8, 13-5, 13-6, 13-7, 13-8
Acquisition cycle, 7-8
Acquisition mode, 7-10, 13-13
Adapter casing, 2-29
ANZW, 7-16, 8-11, 8-12, 8-13, 8-14, 8-16, 9-10, 9-11, 10-13, 10-14, 11-8, 11-16, 11-50, 13-5, 13-6, 13-7, 13-8, 13-9
Application examples, 12-1
 Command interpreter, 12-22
 in directory "EXAMPLE", 12-28
 mass storage functions, 12-13
 process data acquisition, 12-2
ASCII file, 7-27, 10-21, 10-25
ASCII representation, 10-6, 10-23
 example, 10-25
AT slot module, 4-12

B

Background job, 11-37, 11-46, 11-47, 11-53
Backplane connector, 3-24
Base interface number, 6-3, 6-4, 7-5, 8-7, 9-5, 10-5, 12-8, 12-21, 12-26, 13-11
Basic board, 3-2, 3-3, 4-3
Binary representation, 10-23
Binary storage, 10-5
BLGR, 7-14, 8-9, 8-10, 9-8, 10-11, 11-11, 13-4

C

Central controllers, 2-29
Coaxial cable, 2-28

Command interpreter, 2-40
 activating a command, 9-6, 9-9
 Application example, 12-22
 applications, 1-5, 9-1
 command activation, 9-4
 data handling blocks, 13-22
 handling sequence, 12-23
 start, 12-27
 storing commands, 9-6
 terminate, 12-27
 working steps, 12-24
Comment. *Siehe* Comment line; comment line
Configuration file, 7-7, 12-7
 editing, 7-7
 example, 7-12
 parameter set, 7-8
 parameters, 13-12
CONTROL, 11-25, 13-8
Controls, 3-7
Conversion, 3-70, 10-5
Conversion of process data, 7-29
CP 581
 definition, 1-2
 operating system, 1-3
 possible applications, 1-4

CPDHB driver, 2-40, 6-4, 11-28
 call driver status, 11-55
 calling, 11-29
 change interrupt for driver call, 11-56
 delete background job, 11-55
 DHB description, 11-32
 error bits, 13-37
 error messages, 13-17
 functions, 11-37
 global scan of DHB job reception, 11-49
 installation, 11-29
 log off job reception, 11-49
 log on job reception, 11-44
 options, 6-4, 6-6
 parameterization, 11-30
 read S5 area, 11-52
 receive data used, 11-46
 register set, 11-30
 scan TCB status, 11-45
 send data used, 11-47
 set job status, 11-48
 start, 12-8, 12-26
 transmission parameters, 11-33
 write S5 area, 11-54

CPLINK, 2-46
 applications, 1-5
 data exchange, 2-46
 installation options, 2-50
 structure, 2-47

CPMASS
 error bits, 8-32
 preselect directory, 8-16, 12-20
 start, 8-26, 12-21

CPRECCTL, command options, 13-14, 13-17

CPRECORD
 command options, 13-14
 error messages, 13-17
 log messages, 12-10
 start, 7-19, 12-9
 start message, 12-9
 starting, 13-14

CPRECORD.INI, 7-7, 12-7

CPSHELL
 error bits, 9-19, 13-23
 start, 12-27

D

Data handling blocks, 6-4
 parameters, 11-7
 summary, 11-6, 13-3

Data record, 7-28

Data transfer with direct jobs, 11-37

Data transfer without direct jobs, 11-37, 11-52

Data type, 7-26, 8-11, 8-14

DBNR, 8-11, 8-14, 8-18, 8-19, 9-10, 10-13,
 11-9, 13-5, 13-6

Delimiter, 7-6, 10-8, 13-27

Destination path, 7-9, 13-12

DOS function, 2-53

Dual-port RAM, 6-3, 11-5

E

Electromagnetic Compatibility (EMC), 2-33

Equipotential bonding, 2-28, 2-33

Error numbers, 11-18

ESD guidelines, C-1

Expansion unit, 2-30

Extended error code, 10-28, 13-28

Extension, 2-2, 7-9, 7-26, 13-12

F

F option, 10-6

Fault displays, 3-7

FETCH, 8-13, 11-24, 13-6
 call in application example, 12-19

Field delimiter, 7-8, 7-9, 10-6, 10-25, 13-13,
 13-26

Floppy disk drive, 3-32

Floppy disks, 3-32

Format, 7-9, 10-6, 13-13, 13-25

Format data, 7-6, 10-8, 13-27

Format file
 for process data acquisition, 7-6
 for virtual S5 drive, 10-7, 13-27

Format instruction, 7-6, 7-29, 10-8, 10-23

Formatting mode, 10-5

Formatting rules, 7-30, 10-5

Frame size, 7-14, 8-9, 9-8, 10-11, 11-11, 13-4

Free programming, 11-1
 applications, 1-5, 11-1
 procedure, 11-3

Front panel, 2-32

G

Grounding rails, 2-28

H

Hard disk drive, 3-32
 Hardware, design, 2-27

I

Indirect parameterization, 8-19, 11-12
 Individual conversion, 7-6, 7-9, 7-29, 13-13
 example, 7-31
 Installation guidelines, 2-27
 Interaction between CPU and CP 581, 6-6
 Interface
 COM 2, 3-19
 parallel interface for printer, 2-36
 TTY, 2-36
 V.24, 2-36
 Video, 3-22
 Interface COM 1, 3-16
 Interface COM 3, 3-16
 Interface number, 7-14, 7-16, 7-35, 8-9, 8-11,
 8-14, 8-16, 9-8, 9-10, 10-11, 10-13, 13-4,
 13-5, 13-6, 13-7, 13-8
 Interfering signals , 2-27, 2-28

J

Job number, 7-16, 7-35, 8-11, 8-14, 8-16, 9-10,
 10-13, 13-5, 13-6, 13-7
 Job status bits, 8-31, 9-18
 Joker length, 8-12, 8-15, 8-26

K

Keyboard, connection, 3-14
 Keyboard input, 6-10

L

Language switchover, 6-11
 Length word, 11-18
 Logbook file, 7-10, 13-13
 Logic parameters, 4-5

M

Manuals, 5-2

Mass storage functions, 2-40, 12-13, 12-16
 application example, 12-13
 applications, 1-5, 8-1
 data handling blocks, 13-20
 handling sequence, 12-15
 procedures for activation, 8-7
 start, 12-21
 working steps, 12-16
 Mass storage module, 4-8
 Mechanics, 4-5
 Memory cards, 5-2
 Message mode, 7-10, 13-13
 Monitor, 5-3
 connection, 3-15
 monitor cable, 2-33, 2-36
 monitor housing, 2-33
 office monitors, 2-35
 Monitor Cable, 5-3
 Mouse, 2-35
 connection, 3-15
 MS-DOS acknowledgment bits, 9-20
 Multi-processor operation, 6-2
 Multiplexer interrupt, 11-56

N

Normal restart, 2-40

O

OB 21/22, 12-5
 Offset, 7-9, 7-10, 7-28, 13-12
 Operating modes, 6-2
 Operation displays, 3-7
 Operational components, 6-3
 Ordering Information, 5-1, 9-1

P

PAFE, 7-14, 7-15, 7-16, 8-9, 8-10, 8-11, 8-12,
 8-14, 8-15, 8-16, 9-8, 9-9, 9-10, 9-11, 10-11,
 10-12, 10-13, 11-11, 11-19, 13-4, 13-5, 13-6,
 13-7, 13-8, 13-10
 Page, 6-3, 11-5
 Page addressing, 6-3
 Parameter assignment error (PAFE), 11-18

Parameter assignment error bits, 7-22, 8-29, 9-10, 9-17, 10-13, 10-27, 13-5, 13-6, 13-7, 13-10

Peripheral devices, 2-32, 2-36

PLC rack, 2-31

Power supply, 2-29

Preheader data, 7-7, 10-9
 convert in format file (for S5REMOTF), 10-30

Printer, 5-2

Process data, evaluation, 7-26

Process Data Acquisition, Display Status, 12-10

Process data acquisition
 activation, 12-9, 13-14
 application example, 12-2
 applications, 1-4, 7-1
 conversion using format file, 7-6
 conversion using preheader data, 7-7
 defining parameters for data acquisition, 7-5
 definition of conversion procedure, 7-6
 description of ASCII files, 7-27
 evaluation using Lotus 1–2–3, 12-11
 handling sequence, 12-4
 information, 7-37
 measures for application, 7-4
 reactivation, 7-39
 status messages, 7-41
 switching on/off, 7-32, 7-34, 13-16
 triggering, 7-34, 13-16
 universal conversion, 7-6
 working steps, 12-5

Q

QANF, 7-9, 7-28, 8-11, 8-12, 8-18, 8-19, 9-6, 9-10, 11-10, 13-5, 13-6, 13-12

QLAE, 7-9, 7-28, 8-11, 8-12, 8-18, 8-20, 9-6, 9-10, 9-11, 11-10, 13-5, 13-6, 13-12

QTYF, 7-9, 7-10, 8-11, 8-12, 8-18, 8-19, 9-10, 11-9, 13-5, 13-6, 13-12

R

Rack, 2-29

RECEIVE, 13-6
 call in application example, 12-17

RECEIVE ALL, 10-12, 11-22

RECEIVE DIRECT, 11-23

Repair, 3-70

Repetition factor, 7-6, 10-8, 13-27

Representation of S5 data, 10-23, 11-59

RESET, 11-26, 13-8

RESET key, 3-7

Resetting the system, 6-10

RTI module, 4-17

RUN/STOP switch, 3-7

RW, 8-19

S

S5 area, 7-9, 7-10, 11-52, 11-54, 13-12

S5 data representation, 10-5

S5A file, 10-21

S5B files, 10-21

S5F files, 8-17

S5REMOTF. *Siehe* Deactivate S5REMOTF; S5REMOTF Sequence Parameters; S5REMOTF, commands; S5REMOTF, error messages; S5REMOTF, sequence parameters; Starting S5REMOTF

S5REMOTF/S5REMOTF. *See* commands

SEND ALL, 7-15, 8-13, 9-11, 10-12, 11-20, 13-5
 call in application example, 12-7, 12-17

SEND DIRECT, 8-11, 9-10, 11-21, 13-5
 call in application example, 12-18, 12-19, 12-20, 12-25

Slots, 2-29, 2-30

Software installation, 2-38

SSNR, 7-14, 7-16, 7-35, 7-36, 8-9, 8-10, 8-11, 8-12, 8-13, 8-14, 8-16, 9-8, 9-10, 9-11, 10-11, 10-13, 10-14, 11-7, 13-4, 13-5, 13-6, 13-7, 13-8

Status bit, 11-17, 13-30

Status byte parameter assignment error, 11-18

Status codes, 11-50, 11-51, 13-31, 13-32

Status word, 7-16, 8-11, 8-14, 8-16, 9-10, 10-13, 13-5, 13-6, 13-7, 13-9

SYNCHRON, 7-13, 9-7, 10-10, 11-27, 13-4
 call in application example, 12-5

T

Technical Data , 4-1

Terminate command interpreter, 9-20

Timeout, 7-10, 10-18, 11-46, 11-47, 11-53, 11-54, 13-13

Transfer control block (TCB), 11-32, 11-44, 11-52

Transport position, 3-32

Triax cable, 2-33

U

Universal conversion, 7-9, 7-30, 13-13
Using another language, 6-11

V

Vector register, 6-3, 11-5
Virtual S5 drive

- activation, 10-17
- applications, 1-5, 10-2
- conversion mode, 10-6
- data handling blocks, 13-24
- handling, 10-4
- MS-DOS applications, 10-26
- MS-DOS commands, 10-26
- MS-DOS error bits, 10-28
- operations with directories and files, 10-21
- writing in S5 data blocks, 10-22

Z

ZANF, 8-14, 8-15, 8-19, 11-10, 13-7
ZLAE, 8-14, 8-15, 8-20, 11-10, 13-7
ZTYP, 7-10, 8-14, 8-15, 8-19, 11-9, 13-6

To:
Siemens AG
A&D AS E 81
Oestliche Rheinbrueckenstr. 50
D-76181 Karlsruhe

From:
Your Name: _____
Your Title: _____
Your Company: _____
Street: _____
City, Zip Code: _____
Country: _____
Phone: _____

Please check any industry that applies to you:

- | | |
|--|---|
| <input type="checkbox"/> Automotive | <input type="checkbox"/> Pharmaceutical |
| <input type="checkbox"/> Chemical | <input type="checkbox"/> Plastic |
| <input type="checkbox"/> Electrical Machinery | <input type="checkbox"/> Pulp and Paper |
| <input type="checkbox"/> Food | <input type="checkbox"/> Textiles |
| <input type="checkbox"/> Instrument and Control | <input type="checkbox"/> Transportation |
| <input type="checkbox"/> Nonelectrical Machinery | <input type="checkbox"/> Other _____ |
| <input type="checkbox"/> Petrochemical | |



