SIEMENS

Documentation Supplement

C79000-Q8576-C204-02

CP 580
6ES5 998-1AT21, Release 02

Dated 11/91

This supplement contains information and corrections for the CP 580 manual, release 02, which we received after the manual had been printed. This supplement is therefore newer and must be treated as having priority over the information in the manual.

SETUP

The SETUP can only be called with the key combination

CRTL and ALT and S

Setting the message texts

When delivered the message texts of the CP 580 are set to the German language version. The German message texts are contained in the file 'CP580.MSG' and stored in the directory 'CP580'. If you wish to change over to the English language version, you must alter the line
'SET CP580=C:\CP580\CP580.MSG' in the AUTOEXEC.BAT file as follows:

SET CP580=C:\CP580\CP580E.MSG

Caution

If the directory in which the file is present contains a large number of files (several hundred), the access times to a file increase greatly under MS-DOS. Remember this with regard to the CP functions CPRECORD and CPMASS.

Status bits from job management

Note

You cannot transmit any jobs to the CP 580 for the respective job number if bits 1 and 2 are set simultaneously in the status word. This also applies when running-up your S5 program.

Interprocessor communication flags

In the current delivery stage the **interprocessor communication flags** are **not enabled** for use!

S5 program examples

The S5 program examples in the chapters 4, 5 and 6 merely represent examples for assigning parameters to the data handling blocks for the functions CPRECORD, CPMASS and CPSHELL. Programs which are executable and autonomous can be found in Chapter 8.

Supplement to the operation of the CP 580 in the programmable controllers

Section 3.1

Note

The CP 580 is not intended for use with the S5-150U programmable controller.

Supplement to the CPRECORD function

Section 4.3.2.4

The accuracy of the acquisition cycle depends on the user applications running parallel. If the set time grid is exceeded by more than 10%, a message is entered in the logbook file:

Warning: cycle overflow > 10% - date time

However, data recording is continued.

Organization of the S5 data areas

Area (QTYP/ZTYP)	Meaning	Max. offset	Max. length	Organization
AS	Absolute address *	0xFFFFFF	4096	Word-oriented
CA	Counter cells **	255	256	Word-oriented
DB	Data word	4090	4091	Word-oriented
DX	Data word	4090	4091	Word-oriented
FA	Flag byte	255	256	Byte-oriented
IA	Input byte	127	128	Byte-oriented
PY	I/O byte	255	256	Byte-oriented

Area (QTYP/ZTYP)	Meaning	Max. offset	Max. length	Organization
QA	Output byte	127	128	Byte-oriented
RS	RS word **	511	512	Word-oriented
TA	Timer cells **	255	256	Word-oriented

* Please note in the AS data area:
The 32767 words mentioned in the descriptions of the data handling blocks cannot be used, as the CP only provides buffers of 8 Kbytes for this area.

** Offset and length are dependent on the CPU used.

Section 4.3.3.2

Supplement to the parameter BLGR

KY0,0 describes that the standard value which is dependent on the CPU is used for blocking:

S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

Section 4.3.4.1

Caution

If the CPMASS and CPRECORD functions are used simultaneously CPMASS **must** be started **before** CPRECORD.

CPRECORD outputs an acknowledgeable message if the storage space required for creating ASCII files exceeds the actual space present on the hard disk drive or the floppy disk.

Nevertheless, data recording can be started by pressing the key

↵

If you press a different key, the CPRECORD program aborts data recording.

If the CPU which has been addressed is not synchronous when starting the CPRE-CORD function, the message

"[CPDHB] CPU is not synchronized"

is displayed on the screen and stored in the logbook file. In the time grid of the acquisition cycle (parameter 12 in the configuration file) the program CPRECORD then scans the interface to the CPU until

- the parameterized interface is synchronized by the CPU,

- data recording is stopped by the key combination

  SHIFT (left) and **ALT** and ↵

- or the program CPRECORD is uninstalled from the memory by entering **CPRECORD/U**

Section 4.6

Before deinstalling the program CPRECORD using the command **CPRECORD/U** the running data recording must be stopped by pressing the keys

SHIFT (left) and **ALT** and ↵

Supplement to the CPMASS function

Assigning parameters using joker lengths

Depending on the type of data and independent of the data length which has actually been transmitted, files with the following set lengths are always created on the hard disk drive:

S5 data area	S5 data type	Name of the CP 580 file [1]	File length
Data blocks DB	DB	nnnDBaaa.S5F	8192 bytes
Extended data blocks DX	DX	nnnDXaaa.S5F	8192 bytes
Flag area F	FA	FAaaa.S5F	256 bytes
Process image of the outputs	QA	QAaaa.S5F	128 bytes
Process image of the inputs	IA	IAaaa.S5F	128 bytes
Counter cells	CA	CAaaa.S5F	512 bytes
Timer cells	TA	TAaaa.S5F	512 bytes

1) nnn = DB-/DX number, aaa = job number

When reading back the data with joker lengths you must make sure that the areas on the S5 side are large enough.

Section 5.3.3.2

Supplement to the parameter BLGR

KY0,0 describes that the standard value which is dependent on the CPU is used for blocking

S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

Section 5.3.4

Caution

CPMASS must only be installed **once**.

If the CPMASS and CPRECORD functions are used simultaneously, CPMASS **must** be started **before** CPRECORD.

Supplement to the CPSHELL function

Caution

The CP 580 functions CPMASS and CPRECORD must **not** be started via the command interpreter CPSHELL.

The bit "job completed without error" is only relevant for the communication between the CP and the CPU. It does not give any information whether the triggered job (e.g. COPY command, starting a program) was actually executed by the CP.

Section 6.3.4.3

Supplement to the parameter BLGR

KY0,0 describes that the standard value which is dependent on the CPU is used for blocking:

S5-115U	64 bytes
S5-135U	256 bytes
S5-155U	256 bytes

Section 6.4.2

The bits in the status word are only used to monitor data transfer between the CPU and the CP 580. It is not confirmed whether the command given has been executed.

Supplement to the function "free programming of the CP 580"

Section 7.4.6

If the call parameter Timeout (register CX) equals zero the driver will, in any case, immediately respond without errors. It can be scanned with the driver function "scan status".

Section 7.4.2

Note

Before calling the driver for the first time the system area of the transfer control block (TCB) must be preset with zero.

Section 7.4.2.1

The numbers 100-199 are intended to be the job numbers for user programs. All other job numbers are reserved.

Supplement to the application examples

Section 8.1.4

The configuration file is only searched for in the directory which is currently valid, i.e. the directory from which CPRECORD is started. When calling, any deviations must be stated explicitly as parameters.

Restart data recording

When using the "non permanent recording" parameter assignment, data recording is aborted after the set amount of files has been reached.

All old files must be deleted from the directory D:\EXAMPLE before you can restart data recording with the key combination

SHIFT (left) and ALT and ↵

Section 8.4

The listing of the application program CPSHELL in the directory C:\CP580\SRC is an example for free programming. However, no changes may be made to the software, as otherwise we cannot be liable for this product.
The following files in the directory C:\CP580\SRC will serve you as further application examples for free programming:
GETDB.C
PUTDB.C
SHOWDB.C.

Supplement to the reference section of the software

Section 9.5.2 Error messages from the program CPSHELL

The bits in the status word are only used to monitor data transfer between the CPU and the CP 580. It is not confirmed whether the command given has been executed.

Supplement to the reference section of the hardware

Example of a single null modem cable for connecting a PT10 to the COM 1 V.24 interface of the CP 580.

PT 10 laser printer Connector		Computer Socket	
Chassis ground	1	1	Chassis ground
RD	3	2	TD
TD	2	3	RD
Signal ground	7	7	Signal ground
DTR	20	5	Ready to send
		6	Data record ready

(Contacts which are not listed are not being used)

SIEMENS

SIMATIC S5

CP 580

Manual

Order No. 6ES5 998-1AT21
Release 02

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

Technical data subject to change

Siemens Aktiengesellschaft

Order No. 6ES5 998-1AT21
Order from: Elektronikwerk Karlsruhe

Printed in the Federal Republic of Germany

Contents

Warning

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

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

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

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

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

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

Guidelines for Handling Electrostatically Sensitive Devices (ESD)

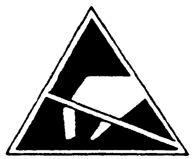
1 What is ESD?

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

They are therefore defined as
"Electrostatically Sensitive Devices"

"ESD" is the abbreviation used internationally.

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



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

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

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

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

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

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

2 When is a Static Charge Formed?

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

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

Examples of static charge:

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

3 Important Protective Measures against Static Charge

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

4 Handling of ESD Modules

- One basic rule to be observed is that electronic modules should be touched by hand only if this is necessary for any work to be done on them. Do not touch the component pins or the conductors.
- Touch components only if

  - the person is grounded at all times by means of a wrist strap

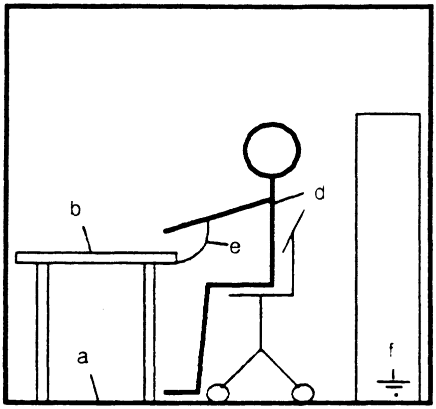
  or

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

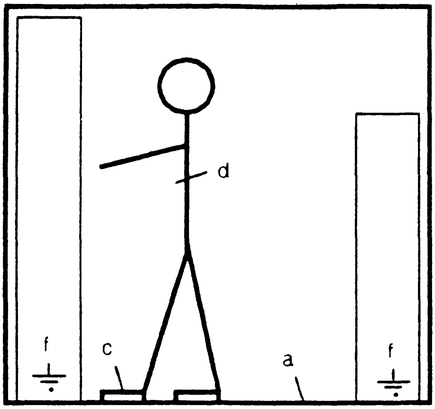
The diagram on the next page shows the required protective measures against electrostatic discharge.

5 Measurements and Modification to ESD Modules

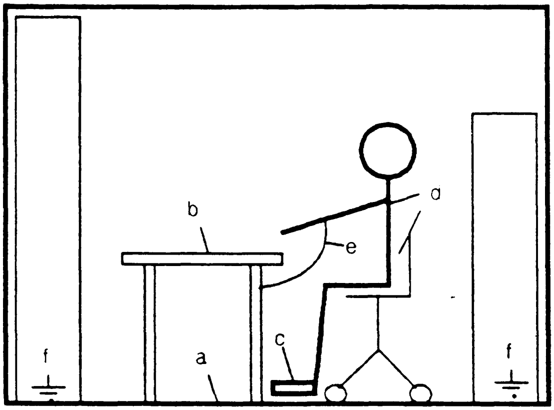
- Measurements on modules may only be carried out under the following conditions:
  - the measuring equipment is grounded (e.g. via the PE conductor of the power supply system) or
  - when electrically isolated measuring equipment is used, the probe must be discharged (e.g. by touching the metallic casing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



Sitting position



Standing position



Standing/sitting position

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

6 Shipping of ESD Modules

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

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

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

Contents

How to Use this Manual

Safety information and ESD guidelines

This information on using the manual is preceded by the "Safety information" and the "ESD (electrostatically sensitive devices) guidelines". These must be exactly observed and followed whenever working with the CP 580.

README file

You can find information produced at a later date than the printing of this manual in the README.TXT file in directory C:\CP580 on the hard disk of the CP 580. You can read and/or print this file using any word processing program.

Scope and design of the Manual

The Manual describes the following versions of the CP 580 communications processor:

6ES5 580-4UA11	2-Mbyte main memory, without coprocessor
6ES5 580-5UA11	2-Mbyte main memory, with coprocessor
6ES5 580-0UA11	4-Mbyte main memory, without coprocessor
6ES5 580-1UA11	4-Mbyte main memory, with coprocessor
6ES5 580-2UA11	8-Mbyte main memory, without coprocessor
6ES5 580-3UA11	8-Mbyte main memory, with coprocessor

The Manual is divided into two main parts:

- Parts 1 to 8 describe in sequential order the work which you must carry out as the user.
- Parts 9 to 14 provide system information in compressed form and can be used as a reference work by users with all stages of knowledge.

The Manual is thus suitable for first-time users and also as a reference work for experienced users. You can ignore parts of the Manual depending on your knowledge and the application, and concentrate on the relevant parts.

The contents of the individual parts are summarized below to help you to become acquainted with the Manual:

Part 1	The **Introduction to Application of the CP 580** describes the facilities provided by the CP 580 and describes ranges of application.
Part 2	describes the scope of delivery of the CP 580. You are also informed on all details of **Installation and Commissioning** of the hardware and software and on the elimination of faults.
Part 3	describes **Operation of the CP 580 in the S5 Programmable Controllers** and informs you of the programmable controllers in which you can use the CP 580 and with which applications multi-processor operation is possible. You will learn how the S5 CPU and the CP 580 coordinate, and which CP 580 applications can be executed simultaneously.
Part 4	describes how you can transmit process data from the S5 CPU to the CP 580 using the **Process Data Acquisition** function and how to evaluate the data on the CP 580 using standard MS-DOS programs.
Part 5	describes how you can use the CP 580 as a storage medium for S5 CPUs using the **Mass Storage Functions**.
Part 6	describes how you can activate any MS-DOS commands on the CP 580 from an S5 CPU using the **Command Interpreter** function.

Part 7	The **Free Programming** section describes how to use the system functions of the CP 580 and the functions of the data handling blocks and how to optimally adapt your programmable controller system to the demands.
Part 8	describes use of the CP 580 by means of **Application Examples** for the four following system functions: process data acquisition, mass storage functions, command interpreter and free programming.
Part 9	The **Reference Section for System Software** contains all important information on the system software in a compressed and tabular form. The SETUP routines for the CP 580, the design and parameterization of the data handling blocks used for the CP and the error bits of the system functions are described.
Part 10	The **Reference Section for Hardware** (device description) contains all important information on the hardware components of the CP 580. The function and meaning of the controls and displays of the CP 580 are explained, and the switch and jumper settings are described. The chapter also contains information on the connection of peripheral devices and on the various interfaces.
Part 11	lists the **Technical Data** of the CP 580. You will find data on e.g. the current consumption, electromagnetic compatibility and the climatic and mechanical environmental conditions.
Part 12	contains a list of **Reference Literature.**
Part 13	contains a **List of Abbreviations** and an **Index**.
Part 14	contains **Ordering Information** for accessories and peripheral devices (state at time of printing of Manual).

Contents

1 Introduction to Working with the CP 580

This section provides you with an initial overview of the CP 580 communications processor. You will learn:

- what a CP 580 is, and what facilities it provides you with,
- what applications are possible for the CP 580,
- what tasks the CP 580 handles in an automation network.

1.1 What is a CP 580 and What Facilities Does it Provide You With?

The CP 580 communications processor is an AT computer compatible with the industrial standard. It is fitted in your programmable controller subrack. It 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 backplane bus enables effective data transfer between the S5 CPU and the CP 580. 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 580 handles the aquisition, storage, management and conditioning of larger quantities of data. Standard MS-DOS user programs can be used for these functions of the communications processor. The installed software comprises the MS-DOS 3.3 operating system and a number of utilities specific to the CP 580 for communication between the MS-DOS and S5 environments.

1.2 What Applications are Possible for the CP 580?

The possible applications of the CP 580 can be divided into the four system functions:

- Process data aquisition
- Mass storage functions
- Command interpreter
- Free programming.

Process data aquisition

Examples of applications of the process data aquisition 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 of 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 580

Command interpreter

Examples of applications of the command interpreter function:

- Calling of MS-DOS commands from the S5 CPU
  - to activate programs for execution on the CP 580
- Printing of S5 CPU messages on a message printer
  - for storage of status and error messages

Free programming

Examples of applications of the free programming function:

- Use of self-generated applications
  - for optimum adaptation of your programmable controller system to the tasks
  - for adaptation of existing programs to communication via the S5 backplane bus
  - to implement the cell level of a manufacturing system according to Fig. 1.1.

1.3 What Tasks can the CP 580 Handle in an Automation Network?

The CP 580 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.

Fig. 1.1 shows a possible task of the CP 580 in the automation pyramid.

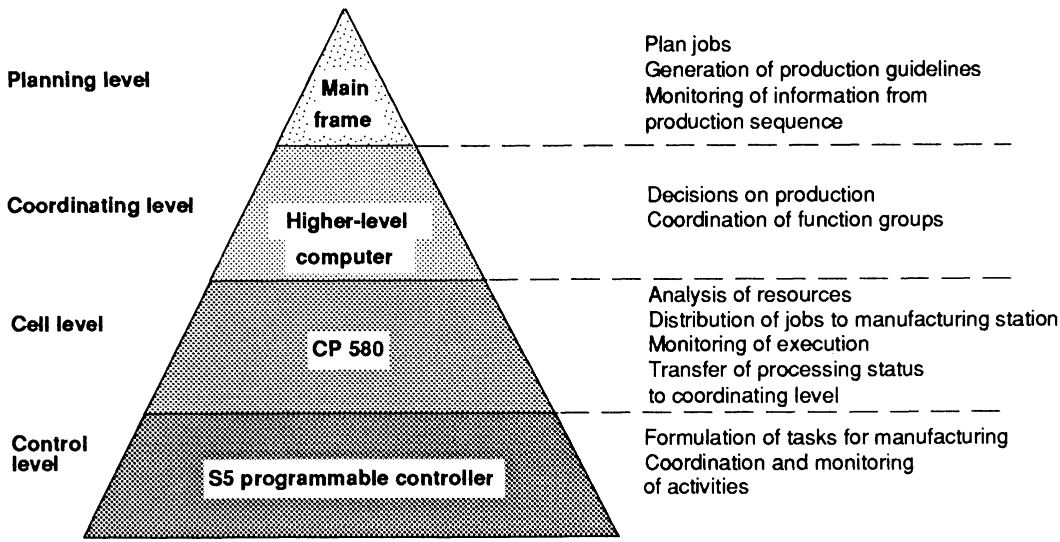


Fig. 1.1 The CP 580 communications processor in the automation pyramid

Contents

Contents

2 Installation and Commissioning

This chapter describes the installation and commissioning of the CP 580.
Please also observe the installation guidelines described in this chapter.

2.1 Unpacking and Checking the Delivered Components

1. Unpack the CP 580.

2. Retain the original packing material for subsequent transport.

Note

The CP 580 is equipped with a disk drive which is sensitive to shock and vibration. Please remember this when handling the system.
Only transport the CP 580 in its original packing material!

2.1.1 Standard Scope of Delivery

Check the scope of delivery!

On delivery the CP 580 includes:

- CP 580: hardware/software (you can only check the installed software if you connect a keyboard and monitor)
- Pocket Guide for MS-DOS
- Manual (please refer to the language-specific Order Nos., see also Chapter 14, Ordering Information).

Check that the Order No. printed on your CP 580 agrees with the configuration you ordered.

CP 580 configurations:

6ES5 580-4UA11	2-Mbyte DRAM, without coprocessor
6ES5 580-5UA11	2-Mbyte DRAM, with coprocessor
6ES5 580-0UA11	4-Mbyte DRAM, without coprocessor
6ES5 580-1UA11	4-Mbyte DRAM, with coprocessor
6ES5 580-2UA11	8-Mbyte DRAM, without coprocessor
6ES5 580-3UA11	8-Mbyte DRAM, with coprocessor

To run up the CP 580 you also need:

- A monitor with
    - Cable connector for the monitor
    - Instructions for the monitor
- A PG 750 keyboard or a PC keyboard with equivalent functions with
    - Instructions for the keyboard

Note

You do not require a keyboard or monitor for operation if you only wish to carry out the mass storage functions with the CP 580.

Note, however, that a keyboard and monitor are useful for commissioning of the mass storage functions.

2.1.2 Environmental Conditions for the CP 580

The environmental conditions and technical data applicable to the CP 580 are listed in Chapter 11, Technical Data.

Note

It is essential to observe the current consumption of the CP 580 in Chapter 11. This must be observed when equipping the programmable controller!

2.1.3 Connectable Peripheral Devices

Fig. 2.1 shows an example of the designs possible with the CP 580 and peripheral devices.

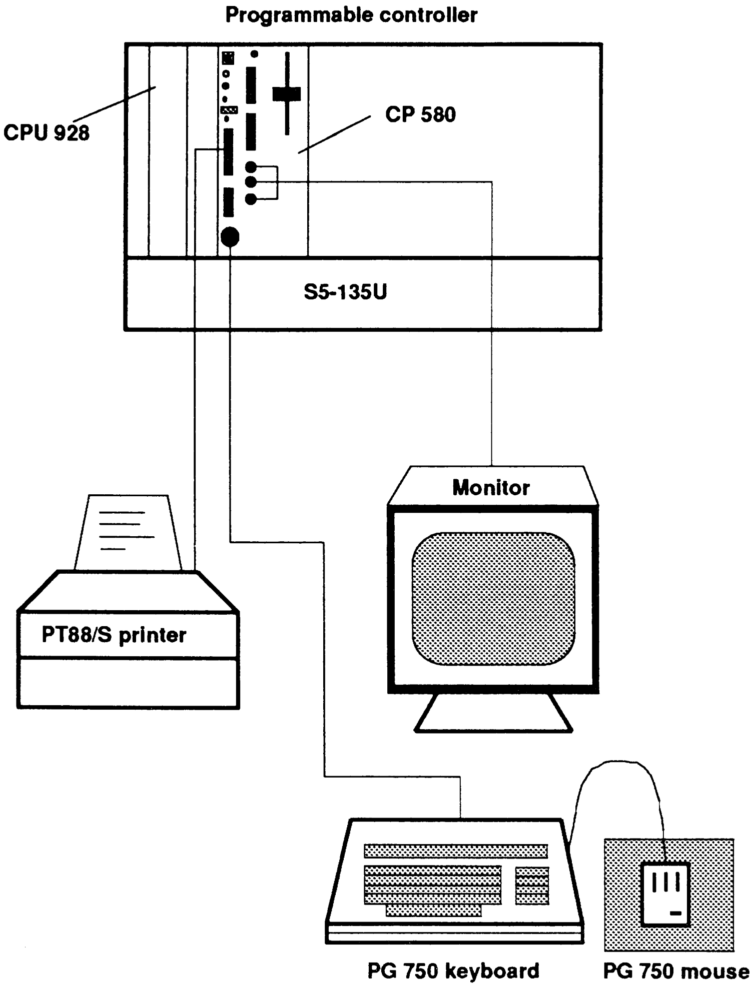


Fig. 2.1 Design with PG 750 keyboard, monitor, printer and mouse

2.2 Installation and Commissioning of Hardware

2.2.1 Check List for Installation and Commissioning

This section explains the procedure for installing and setting up the CP 580 step-by-step. Please proceed as described below.

The check list provides a summary of the individual steps.

1. Is the power supply unit in your subrack correctly dimensioned?
2. Check the jumper assignments and switch settings on the basic board and expansion board.
3. Switch off the power supply to your programmable controller rack and insert the board into the provided slot.
4. Check the position of the mode switch. The switch must be in the RUN position if you want to exchange data with the programmable controller.
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. Startup the software.

2.2.2 Switch and Jumper Settings on the CP 580

Before switching on for the first time, you must set or check the switch and jumper settings described below according to the requirements in order to ensure correct operation with the CP 580. You must carry out the following two steps:

1. Set the base interface number on the basic board
2. Check the other coding switches and plug-in jumpers on the basic and expansion boards

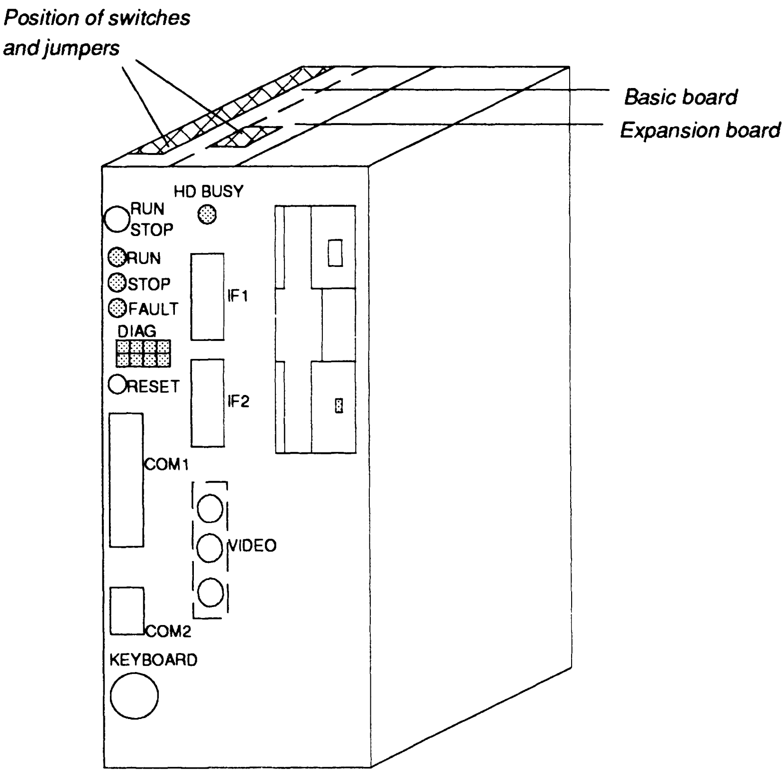


Fig. 2.2 Arrangement of basic and expansion boards and position of switches and jumpers on the CP 580

The coding switches and plug-in jumpers are located along the top edge of the basic board. Fig. 2.3 can be used to set the base interface number and to check the other coding switches and jumpers.

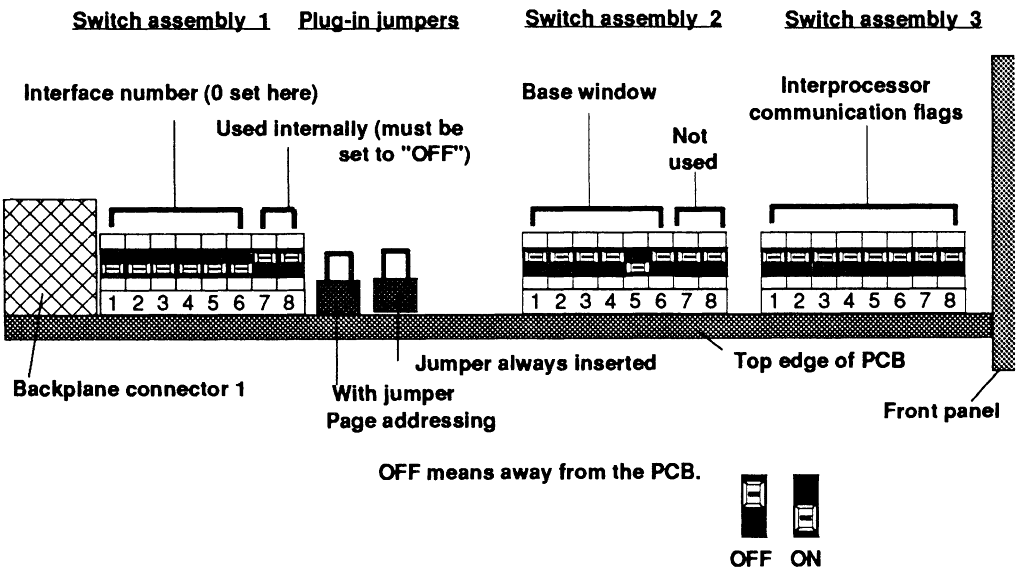


Fig. 2.3 Position of coding switches and jumpers on the basic module

2.2.2.1 Setting of the Base Interface Number (Module Address)

You must assign a base interface number to the CP 580. This corresponds to the setting of the module address with other S5 modules.

Meaning of base interface number:

Data transfer for the CP 580 takes place via the page area of the dual-port RAM. The CP 580 has 4 pages, each with a 1-kbyte memory. The 4 pages are assigned 4 successive interface numbers, where the first number is the base interface number. The base interface number can be from 0 to 252.

A differentiation is always possible between single processor and multi-processor operation.

- Single processor operation:
  - Only one page is available to you in single processor operation.
- Multi-processor operation:
  - Up to 4 CPUs can be plugged into the programmable controller in multi-processor operation of the S5-135U and S5-155U programmable controllers. Each CPU is assigned a page for data transfer with the CP 580.

There is a fixed assignment between the CPU number and the interface number.

CPU number	CP interface number
CPU 1	Base interface number
CPU 2	Base interface number +1
CPU 3	Base interface number +2
CPU 4	Base interface number +3

Setting the base interface number:

You define the base interface number by setting a number from 0 to 252 using coding switch assembly 1 (see Fig. 2.3). The next three interface numbers are assigned automatically. You must define the base interface number both for multi-processor and single processor operation. Only one page can be assigned to each CPU.

The interface number must be dividable by 4 and set as a binary value. The arrrangement of the switches on assembly 1 makes it impossible, however, for you to set a non-permissible base interface number.

To enable the CPU to address the CP 580 correctly, you must define the same interface number (parameter SSNR) when parameterizing the data handling blocks as that set on the CP 580.

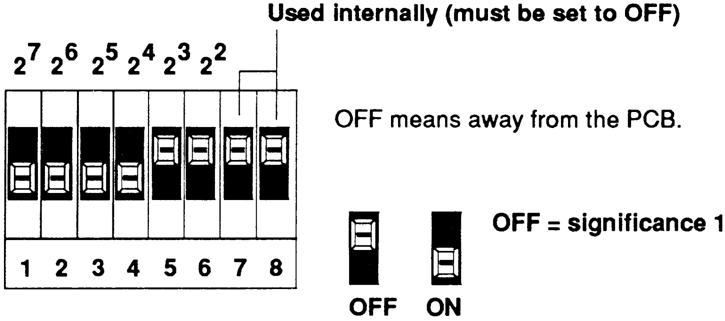
When using several CP 580 modules in one programmable controller, ensure that the interface numbers are not assigned more than once.

Significance of the individual coding switches:

Switch set to OFF means valid (significance 1)
Switch set to ON means not valid (significance 0)

Switches 7 and 8 have no significance with respect to setting of the interface number.

Example: setting of the base interface number:



Equation:

$$SSNR = n \times 2^7 + n \times 2^6 + n \times 2^5 + n \times 2^4 + n \times 2^3 + n \times 2^2$$

Example:

$$SSNR = 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 = 12$$

Base interface number = 12; interface numbers (SSNR) 12,13,14,15 are assigned

Switches 7 and 8 of assembly 1 are used internally and must always be set to OFF!

Fig. 2.4 Example of base interface number 12

2.2.2.2 Checking the Coding Switch and Jumper Settings on the Basic Board

In the case of the other coding switches and jumpers on the basic board, you need only check the factory-set switch and jumper settings.

- **Enabling or disabling of the interprocessor communication flags**

  All interprocessor communication flags are disabled in the factory setting since the supplied software of the CP 580 does not currently use communication flags. Check that all coding switches are set to OFF (see Fig. 2.3, switch assembly 3).

Note

The interprocessor communication flag area comprises 256 communication flag bytes (= 2048 communication flags). The communication flags are transferred cyclically between the CPU and the communications processors (CP) and can be used for coordination.
If you wish to address communication flags using "Free programming", please read Section 10.1.5.1.3.
This section informs you on division of the communication flag areas using switch assembly 3.

- **Checking the base window address**

  The data handling blocks supplied by Siemens are matched to the specified delivery state. The factory settings must not be changed.

  Check the settings of the coding switches as shown in Fig. 2.3 (switch assembly 2).

- **Switching between page addressing and linear addressing**

  This jumper must always be inserted (see Fig. 2.3) when using the data handling blocks since these blocks do not support linear addressing.

2.2.2.3 Checking the Switch Settings on the Expansion Board

The coding switches are already factory-set **for normal operation** and must not be changed.

Check the setting of the coding switch as shown in Fig. 2.5.

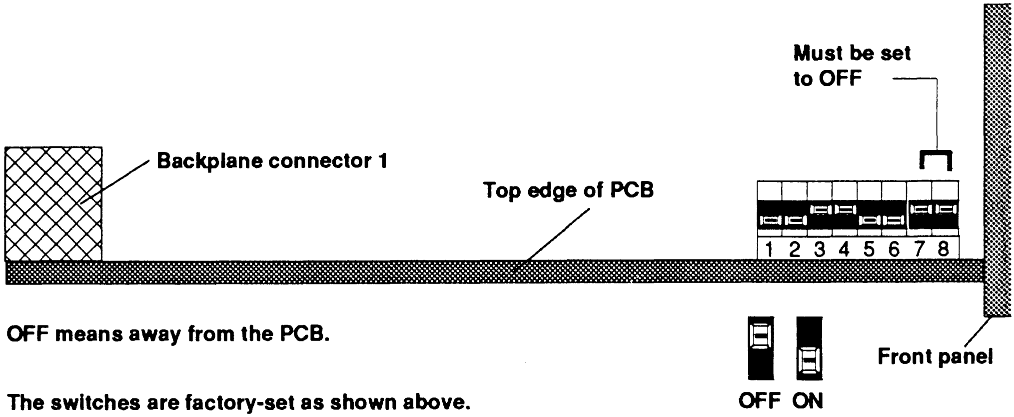


Fig. 2.5 Position of coding switch on the expansion board

2.2.2.4 Fixed Jumpers

All plug-in jumpers are inserted when the CP 580 is delivered and must not be changed.

> **Note**
>
> The basic and expansion boards contain a number of plug-in jumpers which are only used for diagnostic purposes in a test bay.

2.2.3 Installation of CP 580 into Subrack

Caution

If you work on the system with the cabinet open, observe the guidelines for the protection of electrostatically sensitive devices (ESD)!

Note that the CP 580 is inserted into different slots in the various programmable controllers (see Section 2.2.3.2).

2.2.3.1 Interference-free Hardware Configuration

Note

To ensure interference-free operation, observe the shielding measures and installation guidelines included in the respective CPU manuals (see Reference Literature).

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

The following cases can be considered:

- New installation of an S5 cabinet with a CP 580
- Extension of an already existing S5 cabinet by a CP 580.

In both cases we assume you have observed the installation guidelines. The following pages summarize some of the important points of the "Installation guidelines for SIMATIC S5".

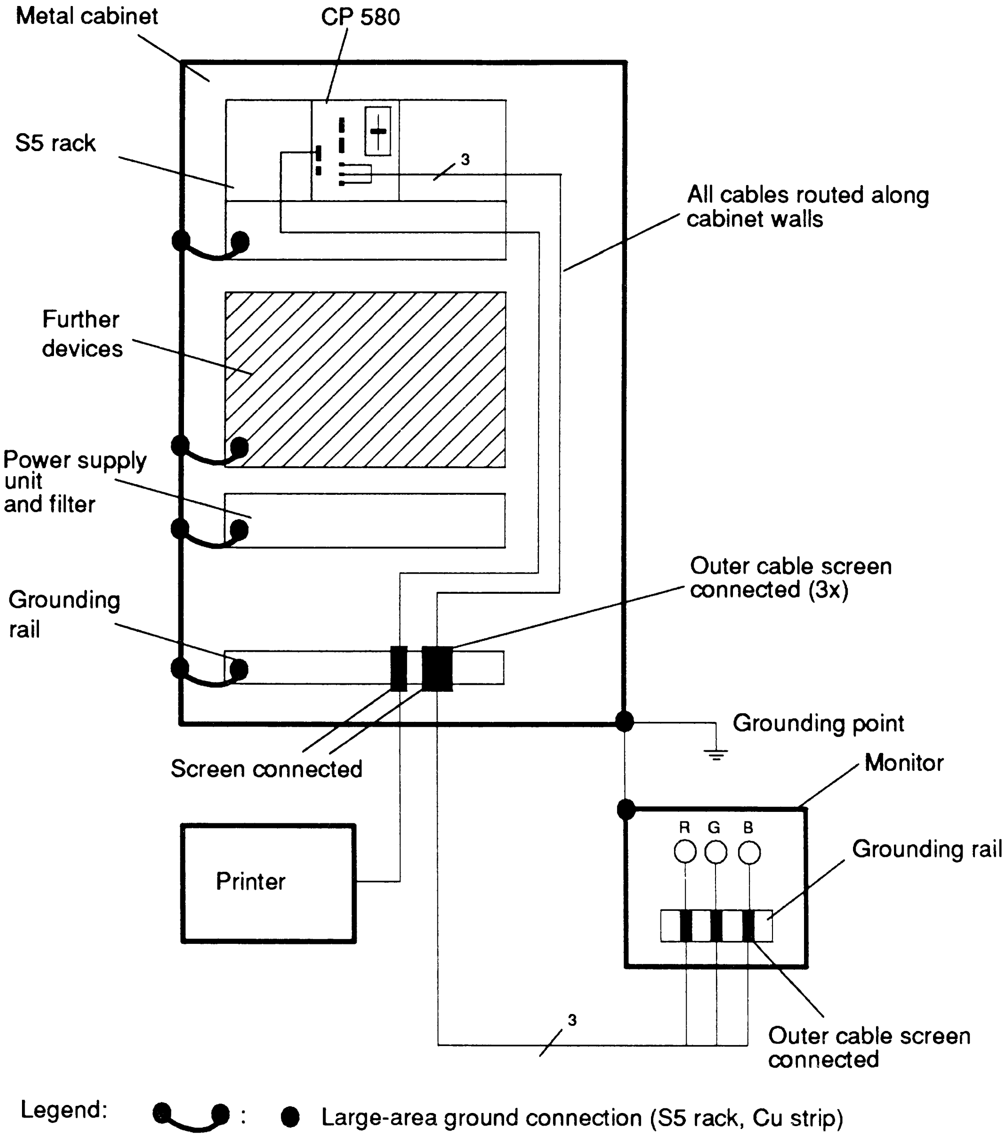


Fig. 2.6 Cabinet design

The following can be seen in Fig. 2.6:

- 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 cables carrying noise signals (e.g. monitor cables and the power supply cable for the printer) are connected directly before they bring noise signals into the cabinet. Connect all cables screened to this point (except coaxial cables with one screen). Only connect the outer screen in the case of signal cables with two screens. (In the case of a "free design" connect the cable screens close to the subrack.)
- Always route signal cables along the cabinet walls.
- Route power supply and signal cables separately.
- 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.
- Use separate cable racks for power and signal cables and position them at least 0.5 m apart.
- Ensure when fitting a SIMATIC®[1] system in an 8MF 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.
- Note when fitting analog modules in the system that their ground must only be connected once to the central reference point. It is essential to avoid multiple connections to other grounded system parts.

1) SIMATIC® is referred to below as SIMATIC.

- In the case of systems which generate a high electrostatic voltage (e.g. 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.

2.2.3.2 Usable Slots for the CP 580 in the Programmable Controller

You can use the CP 580 in the S5-115U, S5-135U and S5-155U programmable controllers.
The CP 580 occupies 4 slots of a subrack in the S5-135U and S5-155U. The adapter casing occupies 2 slots in the S5-115U.

- **S5-115U:**

Observe the following requirements for installation:

- You require an adapter casing for pivoted mounting with 4 slots (see Ordering Information, Chapter 14).
- A fan subassembly is essential (see Ordering Information, Chapter 14).
- Use a 5 V/15 A power supply.

Locations for the approved subracks:

Subrack		Possible locations
CR 700-OLB	(for central controllers)	1-2
CR 700-3	(for central controllers)	1-2

- **S5-135U:**

Note that you cannot use the CP 580 together with the CPU 921 (S processor) since this CPU is not enabled in conjunction with the CP 580.

Slots in the approved subracks:

Slot No. / Central controller	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
6ES5 135-3KA13	■	■	■	■	■	■	■									
6ES5 135-3KA21	■	■	■	■	■	■	■									
6ES5 135-3KA31	■	■	■	■	■	■	■									
6ES5 135-3KA41	■	■	■	■	■	■	■									
6ES5 135-3UA11	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
6ES5 135-3UA21	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
6ES5 135-3UA31	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
6ES5 135-3UA41	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
6ES5 135-3UA51	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
Special functions																
PG-MUX function		■		■		■		■	■	■	■					
Interrupt generation	■		■		■		■	■	■	■	■	■	■	■	■	

The special PG-MUX and interrupt generation functions cannot be used at a slot occupied or covered by a CP 580. Please observe the above assignments if you wish to use one of these two special functions of the S5-135U.

- **S5-155U:**

Slots in the approved subracks:

Slot No. / Central controller	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
6ES5 155-3KA13	■		■	■	■	■		■	■	■	■	■	■	■	■	
6ES5 155-3KA21	■		■	■	■	■		■	■	■	■	■	■	■	■	

- **Expansion units** for S5-115U/S5-135U/S5-155U

Observe the following requirements for installation:

– A power supply with fan is essential.

Slots in the approved subracks:

Slot No. / EG S5-185U	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139
6ES5 185-3UA11	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
6ES5 185-3UA21	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
6ES5 185-3UA31	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
6ES5 185-3UA41	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

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

2.2.3.3 Switching Off the Power Supply of the PLC Rack

Caution

The CP 580 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 580 into the programmable controller.

2.2.3.4 Installation of CP 580 into PLC Rack

- Now insert the CP 580 into your PLC rack. Observe the permissible slots of the various programmable controllers (Section 2.2.3.2).
- Be careful to hold the module straight and not to bend the contact springs of the guides in the subrack.
- Lock the CP 580 in the subrack.

Note

The floppy disk drive does not require transport protection if you wish to remove the CP 580 again.
The hard disk drive automatically moves into the transport position when switched off. It is not necessary to park the hard disk drive.

2.2.4 Connection of Keyboard, Monitor, Printer and Mouse

The connections for the operation devices and peripheral devices are on the front panel of the CP 580.

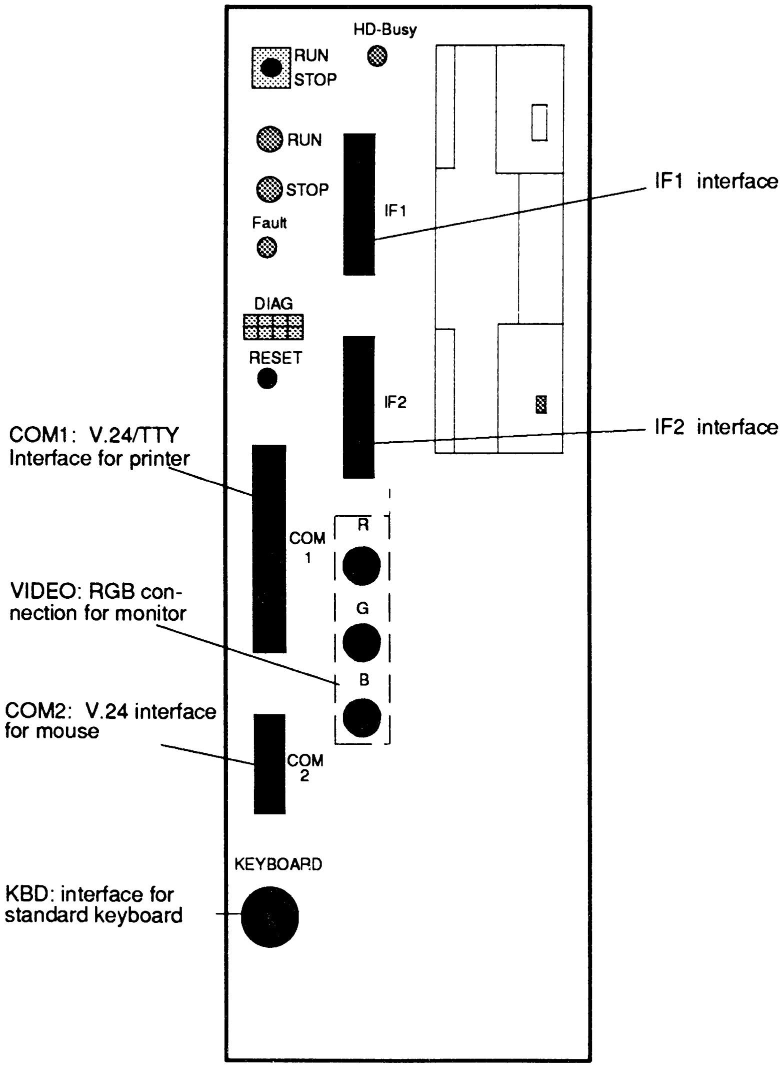


Fig. 2.7 Location of interfaces on front panel of CP 580

You can recognize the following interfaces in Fig. 2.7:

- COM 1: V.24/TTY interface for logging printer
- COM 2: V.24 interface for mouse, if required
- KBD: Interface for standard keyboard
- VIDEO: RGB connection for monitor
- IF1 (COM 3): TTY interface for optional use
- IF2 (COM 4): X.27 (RS 422) interface for optional use

A keyboard and monitor are required to use the CP 580. You can additionally connect a logging printer and a mouse.
Section 10.2, Reference Section for Hardware, shows which devices you can connect.
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.

Caution

Important note when routing connection cables for peripheral devices:

Monitor cables and connection cables between the CP 580 and the PG 750 keyboard/printer must not be routed parallel to power cables!

Route them on their own cable rack positioned at least 50 cm away from power cables.

2.2.4.1 Connection of PG 750 Keyboard

Connect the PG 750 keyboard to the keyboard interface KBD.

2.2.4.2 Connection of Monitor

Connect the monitor to the CP 580 via the VIDEO (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 14, Ordering Information). You can use these cables up to a length of 50 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 Fig. 2.8).

- Make large-area metal-metal connections.

- Isolate the electronics ground of the monitor (see Fig. 2.8) from the housing ground. The separation has already been made in the Siemens multistandard color monitor 6AV1414-0AA00 listed in Chapter 14.

- 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.

Information on positioning of 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.

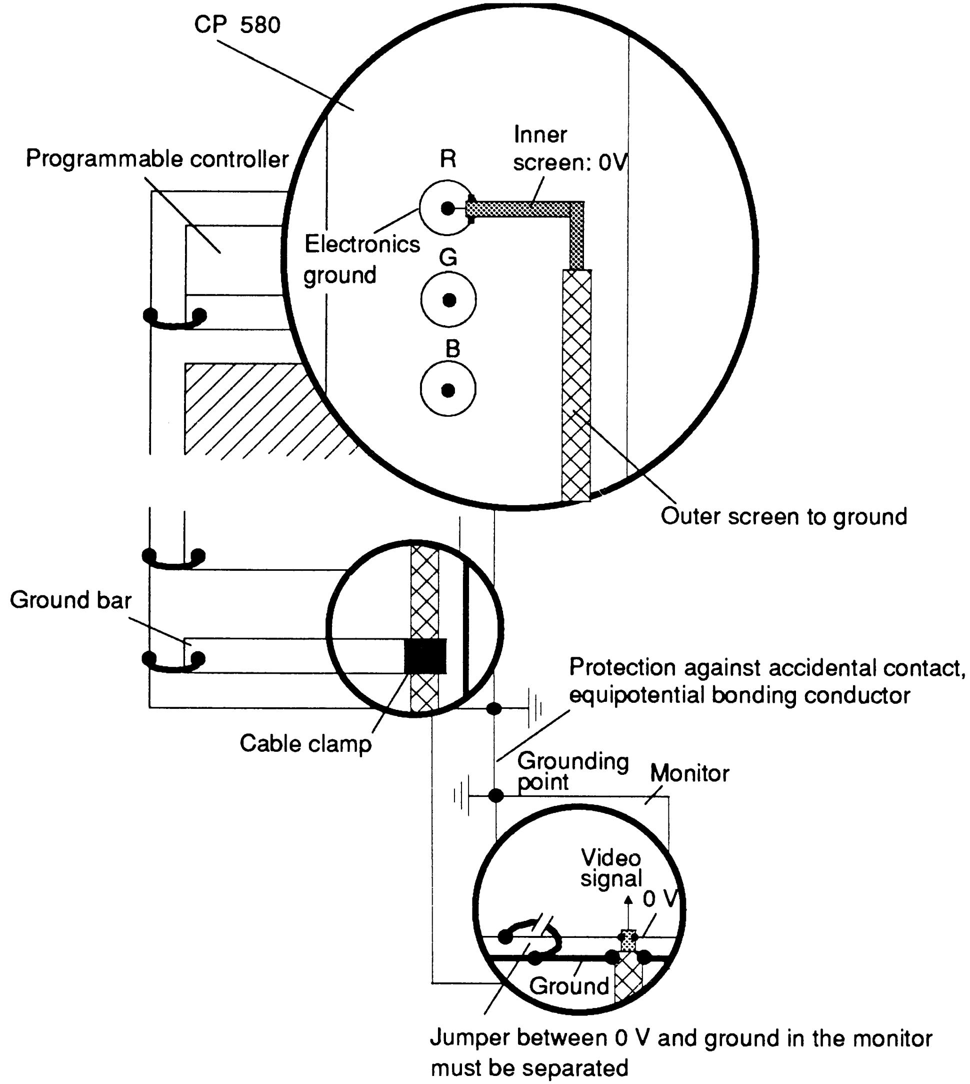


Fig. 2.8 Cable connection to grounding rails

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 580 is thus only possible to a limited extent since only short distances can be covered using these cables.

2.2.4.3 Connection of Logging Printer

- Connect a logging printer to the serial interface COM 1. Printers with V.24 or TTY interface adapters can be used.

- The PT88N/PT89N or PT88S/PT89S printers are recommended and can be connected via a V.24 or TTY interface adapter to the CP 580.

- Section 10.2.3, Reference Section for Hardware, shows you the coding switch settings and the interface adaptation for the PT88N/PT89N and PT88S/PT89S.

- Only cables with the screen grounded at both ends may be used between the CP 580 and printers.
  The standard cables listed in Chapter 14, Ordering Information, satisfy these requirements.

2.2.4.4 Connection of a Mouse

The CP 580 software supplied does not use a mouse!
It may be useful to connect a mouse in order to use additional software on the CP 580.

Connect the mouse to the CP 580 in the following manner:

- Connect any mouse to the serial interface COM 2 on the front panel.
- Connect the PG 750 mouse to the mouse plug fitted on the side of the PG 750 keyboard and the COM 2 interface on the front panel.

You must only use one of the two interfaces, since the COM 2 interface is also occupied when the PG 750 keyboard is connected!

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

The following table lists the limits for the cable length between the devices. A prerequisite is an interference-resistant hardware design as described in Section 2.2.3.1.

Table 2.1 Maximum cable lengths for operation devices and peripheral devices

Device	Length	Remarks
Printer with V.24 interface	10 m	
Printer with TTY interface	500 m	
Monitor	50 m	When using triax cables
Mouse	1.5 m	
PG 750 keyboard	1.5 m	

2.2.5 Setting the RUN/STOP Switch

Set the RUN/STOP switch to "RUN".

2.2.6 Check List Before Switching On the Power Supply

- Is the power supply in your subrack correctly dimensioned (Section 2.1.2)?
- Have you considered the environmental conditions for the CP 580 and connected peripheral devices?
- Have you made the coding switch and jumper settings on the CP 580 and peripheral devices?
- Is the CP 580 inserted into an approved location 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?

2.2.7 Switching On Peripheral Devices

Now switch on the monitor and logging printer.

2.2.8 Switching On the Power Supply to the PLC Rack

Now switch on the power supply for the programmable controller rack.

The operating system is automatically booted when the power supply is switched on.

Hardware commissioning for the CP 580 is now finished.

Please read further in Section 2.3 "Software Commissioning".

2.3 Software Commissioning

2.3.1 Procedure when Switching On for the First Time

The CP 580 runs up when you switch on the programmable controller.

The RUN and STOP displays on the front panel of the CP 580 light up permanently. The operation displays on the floppy disk drive, the hard disk drive, and the floppy disk drive again light up in succession.

The LEDs on the NUM LOCK and SCROLL LOCK keys light up briefly on the connected PG 750 keyboard.

The following message appears on the screen:

"CP 580 BIOS Version X.X"

The operation display on the hard disk drive now lights up with each access.

2.3.2 Start-up with Default SETUP for the CP 580 Basic Version

The CP 580 runs up in the basic configuration without a manual SETUP.
The RUN, STOP and FAULT displays are deleted when the CP/DHB 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.

When delivered, the hard disk contains the MS-DOS operating system, the CP 580 software (CP/DHB driver, program package "File functions") and the AUTOEXEC.BAT file.

The AUTOEXEC.BAT file is set such that the mass storage functions and the command interpreter are available following correct running-up.

To activate and test CPMASS, CPRECORD and CPSHELL, please read:
- Section 4.3.4 for process data aquisition (CPRECORD)
- Section 5.3.4 for the mass storage functions (CPMASS)
- Section 6.3.6 for the command interpreter (CPSHELL).

2.3.3 Normal Restart of CP 580

A normal restart is always carried out when you switch off the operating voltage to your PLC and then on again, if the CP 580 is ready, and if the device configuration preset in the software has been retained in the RAM as a result of the battery back-up.

2.3.4 Making Back-up Diskettes

Once the CP 580 has run-up successfully, it is essential for you to make a back-up copy of the software installed on the CP 580.
A batch file is available for saving the software.

By saving you produce a back-up floppy.

Proceed as follows:

- If you have started the command interpreter, terminate CPSHELL by pressing the key

  ESC

- Format a 3 1/2-inch HD floppy disk by entering the command FORMAT A:/S

- Then enter SAVE580.BAT on the keyboard.

- The screen outputs prompts for the operator which explain the further steps.

2.3.5 Setting the Hardware Clock of the CP 580

The hardware clock of the CP 580 is set using the installed SETUP.
Activate the SETUP program by pressing the keys

CTRL and **ALT** and **S** or **+**

and then set the time.

The CP 580 SETUP is menu-controlled and you need not make any settings apart from the time and date.

For more information see Section 9.1.

Software commissioning of the CP 580 has now been terminated.

2.4 Possible Faults and Their Elimination

2.4.1 The RUN LED does not go to RUN

The cause may be one of the following:

- A page has not been synchronized.
- The CP/DHB driver was not called correctly.
- The RUN/STOP switch is not set to RUN.

2.4.2 The STOP and FAULT LEDs do not go Off

If these LEDs go off this indicates that the CP/DHB driver has been installed in the main memory of the CP 580.

Proceed as follows if these LEDs have not gone off:

- Check whether the CP/DHB driver is called in the AUTOEXEC.BAT file.
- Check that all cables are connected correctly.
- Is the power supply for the complete subrack correctly dimensioned? Check the voltages (mains power supply as well as 5 V, 24 V in the subrack).
- Replace the power supply in the programmable controller by another unit.

2.4.3 Displays on Diagnostic Panel (DIAG)

The BIOS carries out a power-on selftest when the CP 580 is switched on. The LEDs on the diagnostic panel light up as shown in Fig. 2.9 if the CP 580 runs up without faults.
Table 10.5 in Section 10.1.4 explains the error codes of the diagnostic panel and the significance of the LEDs.

Check the display on the diagnostic panel; it must be as follows when operation is fault-free:

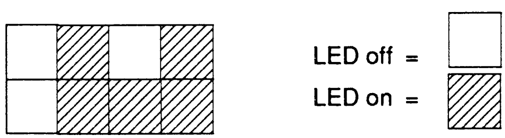


Fig. 2.9 Display on diagnostic panel following fault-free running-up of CP 580

Contents

Contents

3 Operation of CP 580 in S5 Programmable Controllers

This chapter tells you which S5 programmable controllers you can use the CP 580 in, and with which applications multi-processor operation is possible (Section 3.1). This section also tells you which resources on the CP 580 and programmable controller CPUs are used for data transfer between the CPU and CP (Section 3.2) as well as how the CPU and CP 580 basically interact (Section 3.3).
The end of this chapter provides you with some information on the simultaneous use of CP 580 applications.

3.1 Programmable Controllers for CP 580

You can operate the CP 580 with the following S5 programmable controllers and S5 CPUs:

Table 3.1 Programmable controllers and CPUs for operation with the CP 580

PLC / CPU	S5-115U	S5-135U	S5-155U
CPU 941	Yes	No	No
CPU 942	Yes	No	No
CPU 943	Yes	No	No
CPU 944	Yes	No	No
CPU 922	No	Yes	Yes
CPU 928	No	Yes	Yes
CPU 928B	No	Yes	Yes
CPU 946/947	No	No	Yes

Data can be transferred between the CP 580 and a CPU which is plugged into the same programmable controller as the CP 580 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.

3.1.1 Single Processor and Multi-processor Operation

The **CP 580** 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 580 depends on the application. In addition to true multi-processor operation, there is a mixed form of single processor and multi-processor operation where the partner CPU can be changed occasionally by measures on the CP 580.

The various operating modes can be used for the applications as follows:

- Single processor operation (data transfer with **one** CPU):
  all applications.

- Mixed operation (data transfer **alternately with one of up to four** CPUs):
  process data aquisition, free programming.

- 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 (e.g. using interprocessor communication flags or semaphores).

3.2 Operational Components

Special hardware and software components are available for data transfer between the CP 580 and one or more CPUs:

- S5 backplane bus
- Pages, dual-port RAM
- Data handling blocks (DHB)
- CP/DHB driver.

3.2.1 S5 Backplane Bus and Pages

Data transfer between a CPU and the CP 580 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 Fig. 3.1).

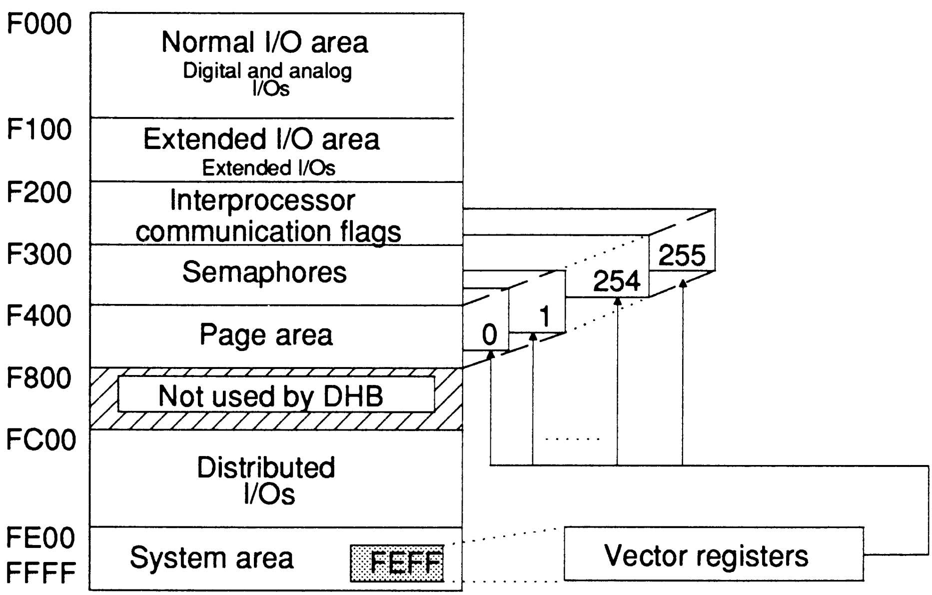


Fig. 3.1 Division of address area on S5 bus

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 580. This is carried out by setting the base interface number (see Section 2.2.2.1). These four pages are located physically in succession in the dual-port RAM on the CP 580 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.

3.2.2 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[1)] for the desired data transfer and which handle the transfer via one of the pages.

Note

The data handling blocks have different 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 4 to 7). You can find a complete summary of the DHBs in Chapter 9.

3.2.3 CP/DHB Driver

The CP/DHB driver on the CP 580 handles the data transfer with one or more CPUs. It is loaded into the memory and is called by the CP 580 applications (supplied system programs or user programs) via a special interrupt interface.
The driver remains resident in the memory. You can only delete it from the memory by removing the load command "CPDHB" from the AUTOEXEC.BAT file and then carrying out a cold restart of the CP 580.

1) STEP®5 is referred to below as STEP 5.

3.3 Principle Interaction Between CPU and CP 580

All applications of the CP 580 with respect to data transfer with a CPU take place in the following steps:

1. Synchronize CP 580 with CPU.
2. Call CP/DHB driver for special application.
3. Carry out data transfer.

3.3.1 Synchronize CP 580 with CPU

The CP 580 is synchronized with a CPU by the DHB SYNCHRON during the CPU restart. The application-specific chapters (Chapters 4 to 7) show you how to program the DHB call.

Note

The CP 580 and the CPUs inserted into the programmable controller run up simultaneously when the PLC power supply is switched on. The CP 580 requires more time than the CPUs, however. 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).

3.3.2 Call CP/DHB Driver for Special Application

The CP/DHB 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 7).

3.3.3 Carry Out Data Transfer

Once the preparatory steps 1 and 2 have been carried out, data transfer between the CPU and CP 580 takes place according to the following principle (see Fig. 3.2):

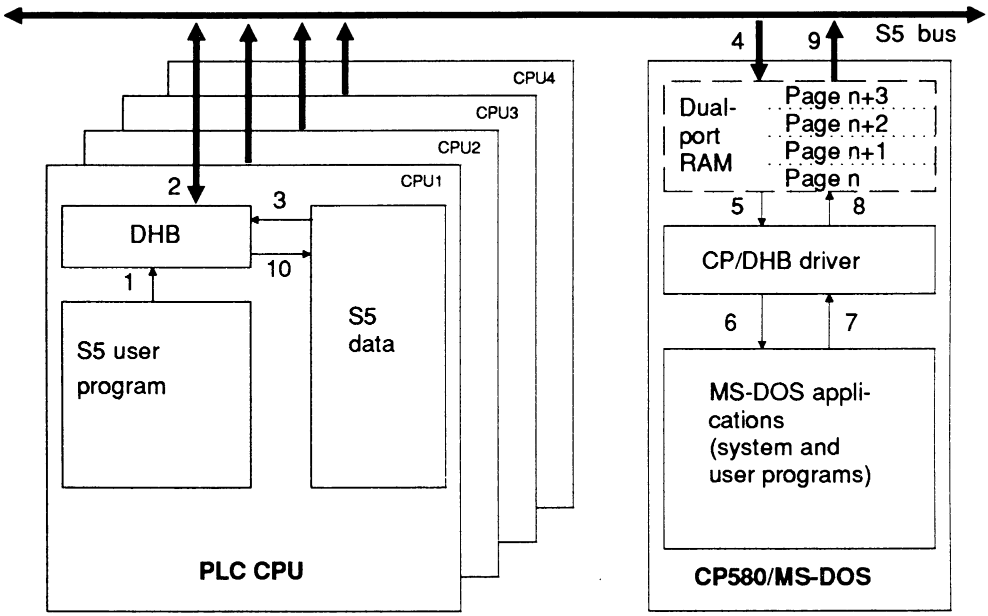


Fig. 3.2 Principle sequence for data transfer between CPU and CP 580

- 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 CP/DHB driver on the CP 580 is informed by an interrupt that data transfer with a CPU is to take place (4).

- A special entry informs the CP/DHB 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 via an interrupt from the S5 bus (9) that data are present in the page. It reads these via the S5 bus and transfers them to the S5 area (10).

3.4 Simultaneous Operation of CP 580 Applications

The system programs on the CP 580 are designed such that they can execute **all applications simultaneously**. Once you are more familiar with the functions and methods of operation of the individual applications, it is up to you to decide whether this is necessary and what effort you want to invest on the CPU side.

In the delivered state, the applications "Mass storage functions" and "Command interpreter" are automatically started by entries in the AUTOEXEC.BAT file following a system restart of the CP 580. This is a important **simultaneous** usage of **both** applications if your CP 580 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 580 is restarted.

Contents

Contents

4 Process Data Acquisition

This chapter is intended for users who wish to read process data from a CPU and to store and evaluate them on the CP 580. The chapter describes all measures and operations required for process data acquisition on the CP 580, the PG and on the CPU side:
Only simple operations are required on the CP 580 and PG for process data acquisition (see Section 4.3.2). 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 4.3.3). 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.

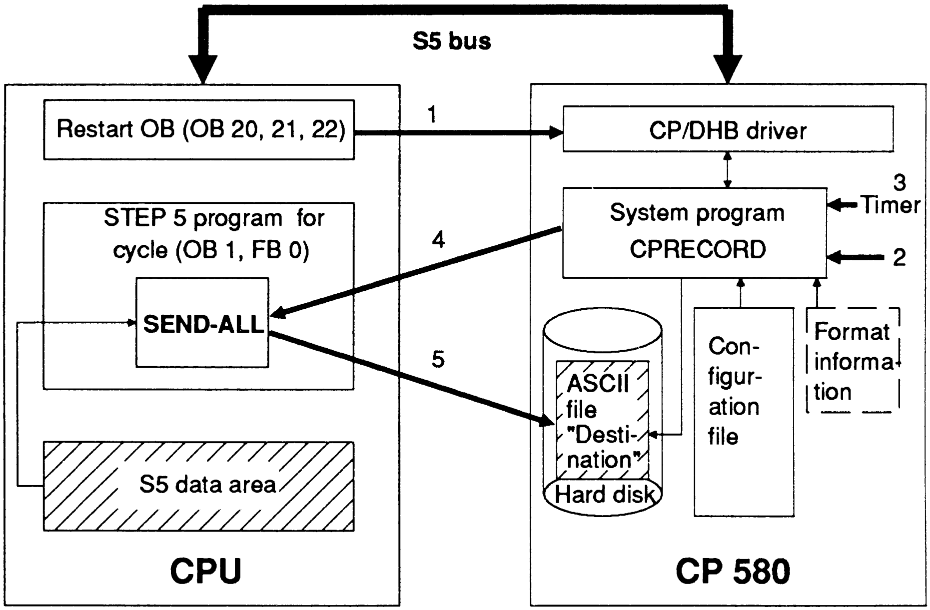
4.1 Application

You can use the system function "Process data acquisition" to acquire data on the CP 580 from various S5 data areas, e.g. data blocks, flags etc. from a CPU which is inserted in the same programmable controller as the CP 580. If several CPUs are present in the programmable controller, you can select one of the maximum of 4 CPUs for process data acquisition. The CPU from which you wish to acquire process data can be simply changed at any time by changing parameters on the CP 580.

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 580 for later evaluation. The data are converted either individually using format specifications in a file on the CP or universally (e.g. in hexadecimal format) into ASCII representation.
You can record the collected and converted process data on the CP 580 independent of the operations at the CPU side and process them using a suitable MS-DOS program (e.g. dBASE).

4.2 Principal Sequences Between CPU and CP

Fig. 4.1 shows you a summary of the principal sequences with process data acquisition using the CP 580.



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 CP/DHB 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 CP/DHB driver and CPRECORD, converted into ASCII format according to the format information and stored in the ASCII file "Destination".

Fig. 4.1 Sequences during process data acquisition

4.3 Process Data Acquisition Operations

4.3.1 Related Procedures

You must carry out the following measures in order to transmit data from a CPU to the CP 580:

Table 4.1 Measures for activation of data acquisition

Measure	on	See Section
Make sure that the CPU with which you wish to exchange data and the CP 580 are plugged into your programmable controller. The base interface no. for data transfer must be set on the CP 580.	PLC/ CP 580	4.3.2.1
Define the parameters for process data acquisition.	--	4.3.2.2
If the data are to be converted individually edit a file with format information on the CP 580 or - only with data blocks - load the PG file ??????ST.S5D with the preheader data of the data block as MS-DOS file on the CP 580.	CP 580 (PG)	4.3.2.3
Edit the configuration file CPRECORD.INI supplied on the CP 580 or your own configuration file according to the required data transfer.	CP 580	4.3.2.4
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG/ CPU	4.3.3.2
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	4.3.3.2

Measure	on	See Section
Continuation of Table 4.1:		
Activate the process data acquisition and test the sequences on the CP and CPU	CP 580/ CPU/ PG	4.3.4
Check by comparing the data (initially transfer statistical test data if necessary) that the functions are executed correctly on the CPU and CP 580	CP 580/ CPU/ PG	4.4

4.3.2 Measures on the CP

4.3.2.1 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 580 can transfer data with up to four CPUs via four pages. The pages are numbered consecutively, the number of the first page is the base interface number.

Define this number for the CPU from which you wish to read the data, and set it on the CP 580 as described in Section 2.2.2.1.

4.3.2.2 Defining Parameters for Data Acquisition

First define the parameters for process data acquisition. These are:

- CPU number
- Conversion instruction: universal, via format file or via preheader data (see Section 4.3.2.3)
- Acquisition mode (cyclic or until a parameterizable data volume is attained)
- S5 data area (see Section 4.3.2.4 for permissible data areas)
- 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
- Time data for acquisition cycle in seconds
- Maximum number of data records per ASCII file (see Section 4.4.2 for data record)
- Maximum number of ASCII files for process data acquisition
- Field delimiter (ASCII character) by means of which the individual data units in a data record are to be separated
- Path data for the ASCII files
- ASCII character for file name extension of ASCII files (e.g. TXT for the file "name.TXT")
- Setting for output of program messages in a logbook file: output on/off
- Path and file names for the logbook file
- Timeout.

4.3.2.3 Definition of Conversion Procedure

- **Universal** conversion: this is defined using the corresponding parameter in the configuration file (see Section 4.3.2.4).

- **Individual** conversion: you have the opportunity here to use a format file on the CP 580 or - in the case of data blocks - to use the data block file of the PG 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 PG since the format file is created directly on the CP 580.

- **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 (see Section 4.3.2.4).

- **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 (must be entered as ";;" because of double function) 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.

Note

The information in a format file refers to the **complete S5 area,** e.g. to a **complete data block.** It is independent of which area is actually transmitted to the CP 580. The **first format instruction** refers to the **first word** or **doubleword** in the defined S5 area.
An **exception** is the S5 area **AS**: in this case the **first format instruction** applies to the **first word** in the AS area **to be transmitted** to the CP 580.
If the S5 data area is **longer** than the result of all format instructions, the **excess data** is converted universally using **"KF".**

Example of a format file:

Name: MYFORM.FMT

Contents:

;3 Fixed-point numbers:
3KF
;4 Floating-point numbers:
4KG
;6 Characters:
6KS

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 PG 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 580 for the conversion. Proceed as follows:

- Call the S5 software package on the PG, 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 PG 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-inch floppy disk. If you do not have a PG with S5 software with the FlexOS operating system, you must transfer the S5 file from the PCP/M format onto a floppy disk formated for MS-DOS using an appropriate program.

- Use the MS-DOS command COPY to load the file from the floppy disk onto your CP 580.

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".**

4.3.2.4 Editing the Configuration File

The file CPRECORD.INI (configuration file) in the directory CP580 on the hard disk contains the parameters with which the CPRECORD program carries out process data acquisition. The file consists of ASCII characters and must contain 16 parameter lines. The **sequence** of parameters (see Table 4.2) is **compulsory!** A line can consist of up to 80 characters (without terminators) and must have the following syntax:

<table>
<tr><td rowspan="2">Parameter</td><td>Semicolon</td><td>Comment</td><td rowspan="2">CR</td><td rowspan="2">LF</td></tr>
<tr><td colspan="2">Optional</td></tr>
</table>

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 4.2 shows the meanings of the individual parameters and the permissible values. Numbers can be entered in **decimal form** (1234) or **hexadecimal form** (0xabcd), e.g. offset for AS.

Table 4.2 Parameters in the configuration file

Line No.	Parameter	Meaning	Permis-sible values
1	CPU no.	Number of CPU corresponding to slot sequence	1 to 4
2	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
3	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
4	Offset (QANF)	No. of 1st data unit to be read in the S5 area (word or byte no. - see Table 4.3)	Depending on data type and CPU

Line No.	Parameter	Meaning	Permissible values
Continuation 1 of Table 4.2:			
5	Number (QLAE)	Number of data units (words or bytes - see Table 4.3), which are to be read from the S5 area starting at "Offset"	Depending on data type and CPU
6	Destination path	Path name for ASCII files	MS-DOS syntax
7	Extension	File name extension 1)	ASCII characters
8	Number of files	Maximum number of ASCII files for process data acquisition	1 to 10 000 2)
9	Number of data records	Maximum number of data records per ASCII file (see Section 4.4.2 for data records)	1 to $(2^{31} - 1)$ 2) 4)
10	Format	"Individual" conversion: path and file name for format file or (only with DB/DX) for file with preheader data	MS-DOS syntax, file name: "??????ST.S5D" or "*.FMT"; ? = letter/ number or @, * = max. 8 letters/ numbers

Line No.	Parameter	Meaning	Permis-sible values
Continuation 2 of Table 4.2:			
10 (contd.)	Format (contd.)	"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)	 KS, KF, KG, KH, KM, KT, KC, KY, KB
11	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
12	Acquisition cycle	Cycle time in seconds in which the process data are to be read from the CPU	1 to 11 799 360
13	Acquisition mode	0: recording is terminated once the defined file number has been reached. 1: "Permanent" recording ; the oldest file is deleted when the defined file number is reached and is then overwritten (similar to cyclic 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

Line No.	Parameter	Meaning	Permissible values
Continuation 3 of Table 4.2:			
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.[3)]	1 to 3600

1) Section 4.4.1 shows you how the file name is produced.

2) Depending on available memory space. Refer to Sections 4.3.4.1 and 4.4.1 to find out how the CPRECORD program reacts if the memory space is insufficient.

3) 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.

4) Is not checked by CPRECORD for reliability. Only enter meaningful values.

Note

In the case of S5 areas which are **byte-oriented** (see Table 4.3), you should enter an **even number** for the parameter **"Number"** (line no. 5) and convert the area in the format "KY".

If you enter an **odd number**, the **last byte** of the S5 area is stored in the ASCII file in a **word format** where the Low byte of the word is invalid.

If you define the parameters "Offset" (line no. 4) and "Number" (line no. 5) 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 4.3 Organization of S5 data areas

Area (QTYP/ZTYP)	Meaning of offset/number	Organization
AS	Absolute address/number of words	Word-oriented
CA	Counter cell number/number of words	Word-oriented
DB	Data word number/number of words	Word-oriented
DX	Data word number/number of words	Word-oriented
FA	Flag byte number/number of bytes	Byte-oriented
IA	Input byte number/number of bytes	Byte-oriented
PY	Peripheral byte number/number of bytes	Byte-oriented
QA	Output byte number/number of bytes	Byte-oriented
RS	RS word number/number of words	Word-oriented
TA	Timer cell number/number of words	Word-oriented

Example of parameters in the configuration file:

Line type: **P = parameter** **C = comment**	**Line contents**	
P / No. 1	1;	Read process data from
C	;	CPU No. 1:
P / No. 2	DB;	S5 area = DB,
P / No. 3	20;	from DB no. **20**,
P / No. 4	0;	from data word **DW 0** onwards (offset = 0),
P / No. 5	8;	**8 data words**
C	;	Path name for ASCII files = **"C\CPU1"**:
P / No. 6	C\CPU1;	
P / No. 7	TXT;	File name extension: **TXT**
C	;	Max. data quantity on CP 580:
P / No. 8	200;	max. **200 ASCII files,**
P / No. 9	500;	max. **500 data records** per ASCII file
C	;	Format:
C	;	**Individual** conversion via
C	;	**preheader data** with file CP580@ST.S5D:
P / No. 10	C:\S5D\CP580@ST.S5D;	
C	;	Field delimiter:
P / No. 11	*;	separate individual data by "*"
C	**;**	**Acquisition cycle:**
P / No. 12	**10;**	acquire **data every 10 seconds**
C	;	Acquisition mode:
P / No. 13	**1;**	**"Permanent"** acquisition (acquisition
C	;	mode = 1)
C	;	Message mode/logbook file:
P / No. 14	1;	**Output** messages (message mode = 1)
C	;	and store in file "**C:\CPRECORD.LOG**"
P / No. 15	C:\CPRECORD.LOG;	
C	;	Timeout:
P / No. 16	50;	**50 seconds**

4.3.3 Programming of CPU

4.3.3.1 Principle

Programming the CPU for data transfer comprises synchronization of the CPU 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.

Fig. 4.2 shows you the positions at which you must call the two data handling blocks in your STEP 5 program.

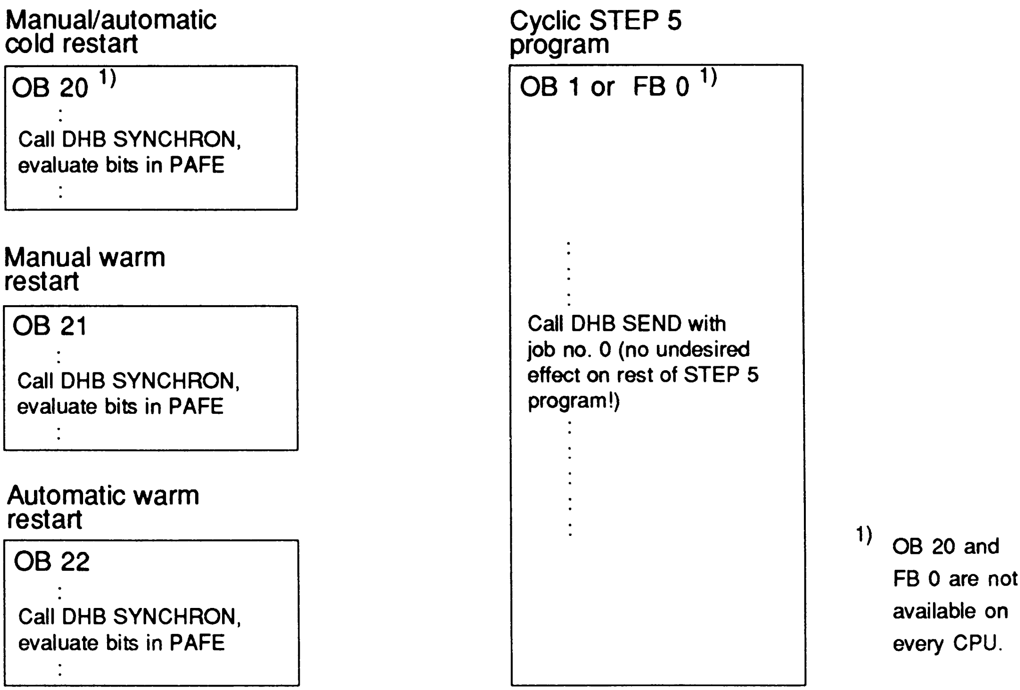


Fig. 4.2 Principle of DHB calls in STEP 5 program of CPU

4.3.3.2 Calling and Parameterizing the Data Handling Blocks

The functions of the data handling blocks which you need to transfer CPU data to the CP 580 are explained in this section as far as is necessary for programming. You can obtain further information on the data handling blocks from the corresponding descriptions of the DHBs. These descriptions are available for the S5-135U and S5-155U (see Reference Literature). The description of the DHB for the S5-115U can be found in the S5-115U Manual.

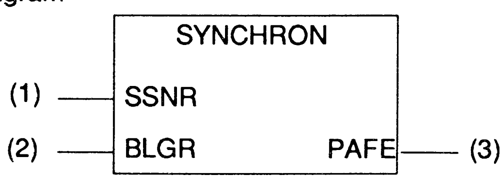
The data handling blocks have different block numbers on the various programmable controllers. The following table lists the numbers of the DHBs SYNCHRON and SEND which you require for the various programmable controllers.

Table 4.4 DHB numbers on the various programmable controllers

DHB \ PLC	S5-115U	S5-135U	S5-155U
SYNCHRON	FB 249	FB 125	FB 125
SEND	FB 244	FB 120	FB 120

DHB SYNCHRON:

Block diagram



Use the block no. FB xxx from Table 4.4

Table 4.5 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 4.3.2.1).
Permissible values:
0 to 255 [1])

BLGR: Use this parameter to define the maximum number of bytes to be transmitted to the CP 580 in a CPU cycle.
Permissible values:
0 = 256 bytes (standard setting)
1 = 16 bytes
2 = 32 bytes
3 = 64 bytes
4 = 128 bytes
5 = 256 bytes
6 = 512 bytes

PAFE Byte address for parameter error bits, e.g. FY 210. Bit no. 0 of the status byte is set to "1" in the event of a parameter error (refer to Section 4.3.4 for the meaning of the other status bits).

[1]) The base interface no. must be set in steps of four (0, 4, 8, 12 etc.)!

Note

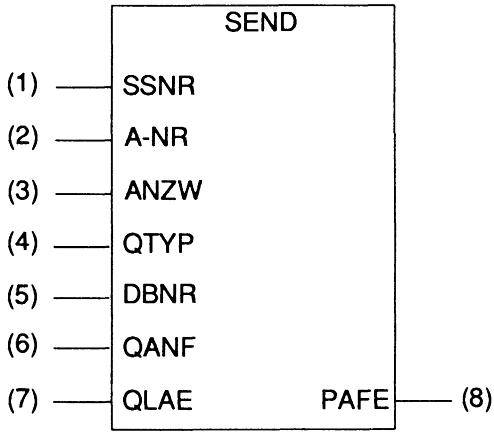
The CP 580 requires more time to run up than the CPU. Therefore you should call the DHB SYNCHRON repeatedly in a program loop until the synchronization is successful. Make sure, however, that the program loop is **always** terminated by an abort criterion (timer or loop counter).

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 and vice versa the bus loading is less with smaller frame sizes, but data transmission to the CP 580 then 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 580, the CPU receives a request if necessary from the CPRECORD program to read the CPU data whilst utilizing the CP/DHB 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 Fig. 4.2.

Block diagram



Use the block no. FB xxx from Table 4.4

Table 4.6 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	KS	These parameters are irrelevant with the function "SEND ALL"; they must nevertheless be specified (e.g. "0")
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 580 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 CP/DHB driver.

4.3.3.3 Example

Data are to be transmitted on the S5-135U from a CPU 928B (the only CPU in the PLC!) to the CP 580. The CPU 928B must then be programmed as follows:

Function block FB 111 (RESTART) is called in the restart OBs (OB 20 to OB 22). FB 111 handles synchronization of the CP 580. The necessary STEP 5 operations are programmed in OB 1 for cyclic calling of the DHB SEND.

The names of the DHBs for the S5-135U programmable controller are obtained from Table 4.2:
DHB SYNCHRON = FB 125, DHB SEND = FB 120.

The STEP 5 program consists of three parts:

a) Calling of FB 111 in the restart blocks

b) FB 111 with calling of DHB SYNCHRON

c) STEP 5 operations for calling the DHB SEND in OB 1.

a) **STEP 5 operations in OB 20, OB 21 and OB 22:**

```
       :                 STEP 5 OPERATIONS FOR
       :                 OTHER FUNCTIONS
       :
       :                 STEP 5 OPERATIONS FOR
       :                 SYNCHRONIZATION OF CP 580:
       : JU      FB 111  CALL CPSYNC
NAME   : CPSYNC
ERR    :         M 200.0 MARKER FOR PARAMETER ERROR
       :
       :                 STEP 5 OPERATIONS FOR
       :                 OTHER FUNCTIONS
       :
```

b) **Function block FB 111:**

```
SEGMENT 1
NAME    : CPSYNC
DECL    : ERR        I/Q/D/B/T/C: Q      BI/BY/W/D: BI
        :
        :
        : JU         FB 125
NAME    : SYNCHRON
SSNR    :            KY 0,12     INTERFACE NO. OF CP 580 (PAGE NO. 12)
BLGR    :            KY 0,0      FRAME SIZE = 256
PAFE    :            FY 255
        :
        : A          M 255.0     PARAMETER ERROR DETECTED
        : =          ERR         SET OUTPUT BIT
        :
        : BE
```

c) **STEP 5 operations in OB 1:**

```
        :                        STEP 5 OPERATIONS FOR
        :                        OTHER FUNCTIONS
        :
        :                        STEP 5 OPERATIONS TO
        :                        CALL THE DHB SEND:
        : JU         FB 120      CALL DHB SEND
NAME    : SEND
SSNR    :            KY 0,12     INTERFACE NO. OF CP 580 (PAGE NO. 12)
A-NR    :            KY 0,0      JOB NO. = 0 ("SEND ALL")
ANZW    :            DW 1        DW 1 AND 2 FOR TRANSFER FLAGS
QTYP    :            KS          NO SIGNIFICANCE
DBNR    :            KY 0,0      NO SIGNIFICANCE
QANF    :            KF +0       NO SIGNIFICANCE
QLAE    :            KF +0       NO SIGNIFICANCE
PAFE    :            FY 201      FLAG BYTE FOR PARAMETER ERROR
        :
        :                        STEP 5 OPERATIONS FOR
        :                        OTHER FUNCTIONS
```

4.3.4 Activation and Testing of Process Data Acquisition

4.3.4.1 Activation

You can activate process data acquisition (initially for the test) once you have carried out all preparatory measures on the CPU and CP 580:

- Make sure that the entry for loading the CP/DHB driver ("CPDHB") is present in the AUTOEXEC.BAT file in the main directory of the CP 580 hard disk. [1])

- Boot the CP 580 in order to start the CP/DHB driver. [1])

- Restart the CPU in order to synchronize it with the CP 580.

- Start process data acquisition on the CP 580:

  Enter the command **CPRECORD on**.

  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 580 is run up.

  ⇒ The program starts and outputs the following text on the monitor:

```
CPRECORD data recording for CP 580 - version 1.0
Copyright (c) Siemens AG 1991
=======================
CP/DHB driver version = xx
CP/DHB interrupt number = 0X66
CPRECORD installed...
'CPRECORD\?' indicates command syntax
Data recording started
```

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 program installs itself in the main memory.

⇒ The CPRECORD.INI configuration file is searched for in the directory in which the program is present, and is then evaluated.

⇒ If you have switched message output on in the configuration file, the start message of the program together with the date and time is entered in the logbook file.

⇒ 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 error message if the space on the drive is insufficient.
Data recording is nevertheless commenced.

⇒ The process data are transferred into the ASCII files whose path you have specified in the configuration file.

⇒ The CPRECORD program is executed in the background. All other programs can be active in the foreground; the reaction times are influenced, however.

4.3.4.2 Testing

Proceed in steps to find out whether the data transfer required from the CPU to the CP 580 is being executed correctly:

1. Check whether the STEP 5 operations for data transfer are being executed correctly on the CPU.
2. Check that your configuration file on the CP 580 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 4.5).
3. Initially transfer static test data (data block) from the CPU to the CP 580 and check whether these reach the destination file on the CP 580 correctly converted.

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 PG on-line functions (handling is explained in your programmable controller manual or in the STEP 5 Basic Package manual, see Reference Literature). Initially activate the CP 580 **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:

Error byte PAFE:

Bit No.	7 … 4	3 … 1	0
	Cause of error		*

* = Common bit: 0: no errors
1: parameter error, more details in bits 4 to 7

Table 4.7 lists all error bits which can be stored by the data handling blocks in PAFE.

Table 4.7 Causes of parameter 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)

PAFE value	Cause of error
Continuation of Table 4.7:	
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 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 Table 4.7 occurs: check your STEP 5 operations for correct parameters and compare these values with the hardware configuration of your PLC and the CP 580 if applicable. Also check that the CP 580 including the CP/DHB driver is ready.
You can proceed to the next test step if no errors have been indicated in the PAFE.

Re 2.:

Initially program a data block DB with test data on the PG such that these data can be converted universally (e.g., only enter fixed-point numbers in the DB). 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:

- S5 area: enter "DB" here.
- DB/DX number: enter the DB no. under which you have loaded the data block on the CPU.
- Number of files: only specify one file for the test.
- Number of data records: specify a clear number, e.g. 20.
- S5 format: enter the data format with which you wish to convert all the data in the DB universally.
- Cycle time: enter a large time in order to initially eliminate dynamic effects during the test.
- Permanent: switch on the storage of messages in a logbook file using "1".
- Logbook file: enter the path/name for the logbook file.
- 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 4.5).

Re 3.:

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 4.4 how the data are stored in ASCII files):

To do this, log the ASCII file on the CP 580 on a printer using the MS-DOS command PRINT and compare the log with that of the data block from the PG. Data transmission is correct if the data are identical.

Once you have carried out steps 2 and 3 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 4.3.2.3) or transfer the S5 file generated on the PG with the data and preheader information of the data block to the CP 580.

4.4 Evaluation of Acquired Process Data

This section tells you how the process data are stored on the CP 580 and where you can find the data of the data block again.

4.4.1 Storage of Process Data on the CP 580

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 4.8 Names of ASCII files

Assignment "Data type/file name"	
Data type	**File name** [1]
AS	ASmmmm.eee
RS	RSmmmm.eee
DB	nnnDmmmm.eee
DX	nnnXmmmm.eee
QA	QAmmmm.eee
IA	IAmmmm.eee
FA	FAmmmm.eee
PY	PYmmmm.eee
TA	TAmmmm.eee
CA	CAmmmm.eee

1) nnn = data block no. of DB/DX
mmmm = consecutive file number
eee = file name extension from configuration file

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 580 and starting of process data acquisition, the most recently written ASCII file is 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, created again and rewritten (the old data are then lost). Once this has been filled, the next file is recreated etc., i.e. the data are stored in a type of cyclic process.

- "Terminate recording": no further data are acquired.

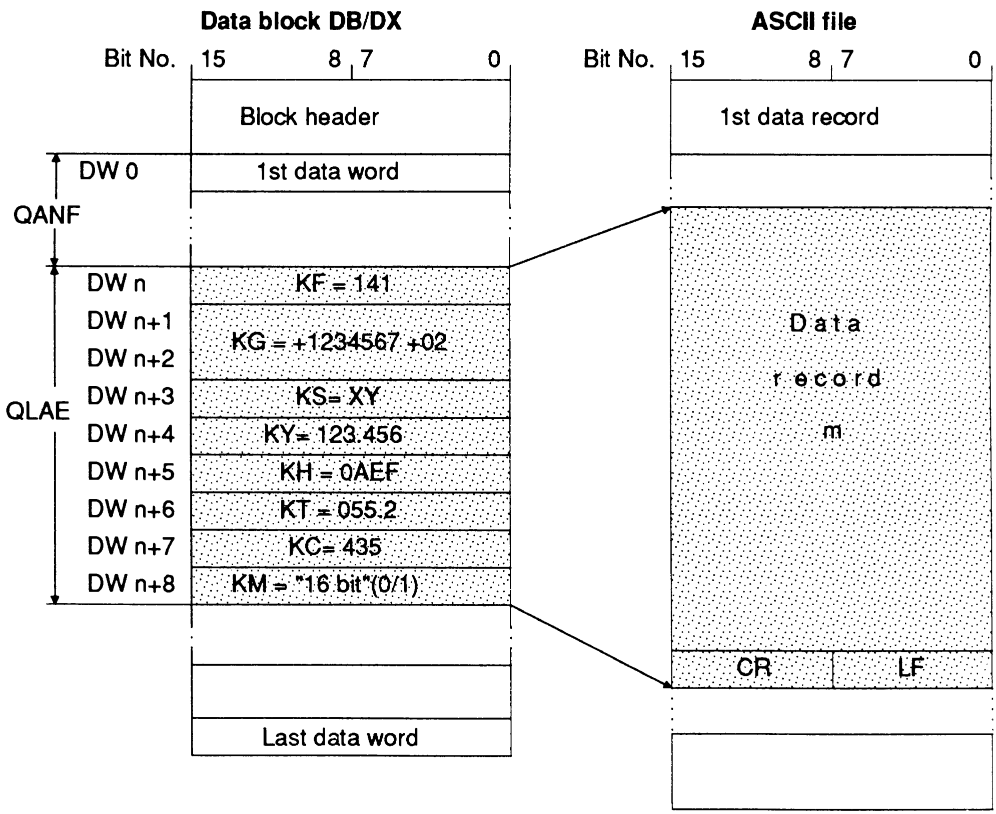
Note

The ASCII files are filled with the value "binary zero" when created. Unwritten areas of an ASCII file therefore have a value of zero.

4.4.2 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 Fig. 4.3).



QANF = offset from 1st data word = n

QLAE = number of data words to be acquired = 8

Fig. 4.3 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 Fig. 4.3):

"ʌʌ+141*+1.234567e+001*XY*123*456*0AEF*ʌʌʌ5500*435*ʌ4711(CRLF)"
ʌ = space, * = delimiter

4.4.3 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.

Note

Independent of the format specification for the individual data, **one item of 16-bit information** from the S5 area is observed and converted. In the case of byte-oriented S5 data (e.g. inputs and outputs), this means that two bytes are combined in each case.

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 4.9 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"	Repetition factor in format instruction	QA, DB, DX, IA, FA, PY
	Format instruction "KF"		
KF = +257	"ΛΛ+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

Data in S5 area	Characters in ASCII file [1)]	Array length	Application for S5 area
Continuation 1 of Table 4.9:			
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 65535) is converted into an unsigned decimal number: "∧1854"	5 characters, leading spaces	All areas
Format instruction "KT"			
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

Data in S5 area	Characters in ASCII file [1]	Array length	Application for S5 area
Continuation 2 of Table 4.9:			
Format instruction "KC"			
KC = 032	"Λ32"	3 characters, leading spaces	DB, DX, CA

1) Λ = 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 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".

4.4.4 Example of "Individual" Conversion

Table 4.10 shows you an example of how S5 data (e.g. from a data block) are converted in a format file using format instructions. "*" is declared as the delimiter following a data record.

Table 4.10 Example of individual conversion of S5 data

S5 data block		Format file/ format instruction	ASCII file	
DW No.	Contents of DW		Data record No.	ASCII character
0	KF = +123	;3 Fixed-point numbers: 3KF	0	ΛΛ+123*
1	KF=-4567		1	Λ-4567*
2	KF=+34653		2	+34653*
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: 6KS	7	EXAMPL*
12	KS = AM			
13	KS = PL			

4.5 Status Messages

If special events are determined during process data acquisition, they can be entered as messages in a logbook file on the CP 580 (see Section 4.3.2.4 "Editing the Configuration File").

Messages are generated by the CP/DHB driver and the CPRECORD program. The following two tables show you which messages can occur and what they mean.

Table 4.11 CP/DHB driver messages

Message
[CPDHB] CPU not synchronized
[CPDHB] Aborted by timeout
[CPDHB] Parameter invalid (PAFE 1)
[CPDHB] DB/DX does not exist (PAFE 2)
[CPDHB] Area too small (PAFE 3)
[CPDHB] Area does not exist (PAFE 4)
[CPDHB] Status word error (PAFE 5)

Table 4.12 CPRECORD program messages

Message	Remarks
Data recording started <Date> <Time>	Message at beginning of process data acquisition
Data recording finished <Date> <Time>	Message at end of process data acquisition
Error opening output file	MS-DOS output

Message	Remarks
Continuation 1 of Table 4.12:	
Format error in configuration file with parameter <n>	<n> = line number of faulty parameter
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
Invalid output drive	MS-DOS output
Insufficient drive capacity (<Actual value> <Required value>)	MS-DOS output
CPRECORD file cannot be removed from memory	MS-DOS output
Unknown option	Incorrect parameter in CPRECORD command (see Section 4.6)
S5D file: could not be opened	MS-DOS output
S5D file: read error	MS-DOS output
S5D file: not a root directory	Format error
S5D file: too many subdirectories (> 128)	
S5D file: no DV subdirectories	
S5D file: positioning error	MS-DOS output "Seek Error"
S5D file: too many data elements (> 128)	
S5D file: the configured DV block is missing	
S5D file: wrong block ID in DV preheader <Code>	Format error, <Code> = incorrect block code

Message	Remarks
Continuation 2 of Table 4.12:	
S5D file: wrong block no. in DV preheader (<Number>)	Format error, <Number> = incorrect block number
S5D file: DV preheader address too large (> 4095)	Format error
S5D file: Format error, first format = empty	
Output drive full	MS-DOS output
Error when writing output file	MS-DOS output
Output directory not found	MS-DOS output
Configuration file: unknown S5 area	
Insufficient number of available DOS clusters (<Actual value> <Required value>) 1)	
CPRECORD.INI not found in current directory	
Configuration file extension is not ".INI": <Config file>	<Config file> = name of configuration file
CP/DHB driver not loaded	
FMT file: could not be opened <FMT file>	MS-DOS output <FMT file> = name of format file
FMT file: unknown format in line <n> (<Format>)	<n> = line No., <Format> = incorrect format
FMT file: format error in line <n> (<Format>)	<n> = line No., <Format> = incorrect format
FMT file: too long in line <n> onwards (<Format>)	<n> = line No., <Format> = incorrect format

Message	Remarks
Continuation 3 of Table 4.12:	
S5D file format is permissible only with S5 area DB or DX	
Error when opening log file <Log file>	<Log file> = name of logbook file

1) Cluster = contiguous logical memory area on drive.

Example of an error message:

CPRECORD data recording for CP 580 version - 1.0
Copyright (c) Siemens AG 1991
============================
CP/DHB driver version = xx
CP/DHB interrupt number =0X66
CPRECORD installed...
[CPDHB] CPU not synchronized

4.6 Handling in Various Operating Situations

The CPRECORD program enables you to influence process data acquisition prior to starting or during the sequence or to obtain information on operating states and parameters using special commands and special key combinations (so-called hot keys).

- Information on CPRECORD:

  Enter the command **"CPRECORD /?"**.

- Replacing the preset configuration file:

  Start process data acquisition using the command **"CPRECORD path name"**

  e.g. **CPRECORD C:\MYDIR\MYCONF.INI**

  Following this command, CPRECORD uses the configuration file specified by you.

- Delete CPRECORD program from the CP 580 memory (process data acquisition is aborted if active when the command is entered):

  Enter the command **"CPRECORD /U".**

Note

If the acquisition of process data in non-permanent mode was terminated because the maximum file number was reached, you must enter the command **"CPRECORD /U"** before restarting acquisition.

- Display of configuration parameters:

  Simultaneously press the keys

  SHIFT (right-hand key) and ALT and ↵

  The CPRECORD program outputs the parameters from the configuration file on the screen starting at the current cursor position (see Fig. 4.4).

```
( 1) CPU number = ...
( 2) S5 area = ...
( 3) DB/DX number = ...
( 4) Data offset = ...
( 5) Data length = ...
( 6) Output directory = ...
( 7) File extension = ...
( 8) Number of files = ...
( 9) Number of data records = ...
(10) S5 format = ...
(11) Field delimiter = ...
(12) Cycle time (sec.) = ...
(13) Permanent = ...
(14) Logging = ...
(15) Log file = ...
(16) Timeout (sec.) = ...
```

Fig. 4.4 Display of configuration parameters on the screen

- Display of the current operating status:

  Simultaneously press the keys

  [ALT] and [↲]

The CPRECORD program outputs the following status variables of the process data acquisition on the screen starting at the current cursor position (see Fig. 4.5):

```
Recording active = 0/1    / 0: recording inactive   1: recording active
Timer = ...               / remaining time for recording
Record = ...              / No. of current data record in file
File = ...                / current file
Permanent cycle =         / cycle counter
```

Fig. 4.5 Display of configuration parameters on the screen

- Interrupt or continue acquisition:

  Simultaneously press the keys

  [SHIFT] (left-hand key) and [ALT] and [↲]

  The function acts like a "flip-flop": if the acquisition was active, it is interrupted by pressing the key combination. If it was interrupted by pressing the key combination, pressing again continues it.

Contents

Contents

5 Mass Storage Functions

This chapter is intended for users who wish to use the CP 580 as an additional storage medium for a CPU. The chapter describes all measures and operations required on the CP 580 and on the CPU:
Only a single setting is required on the CP 580 (see Section 5.3.2). 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 5.3.3). 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.

5.1 Application

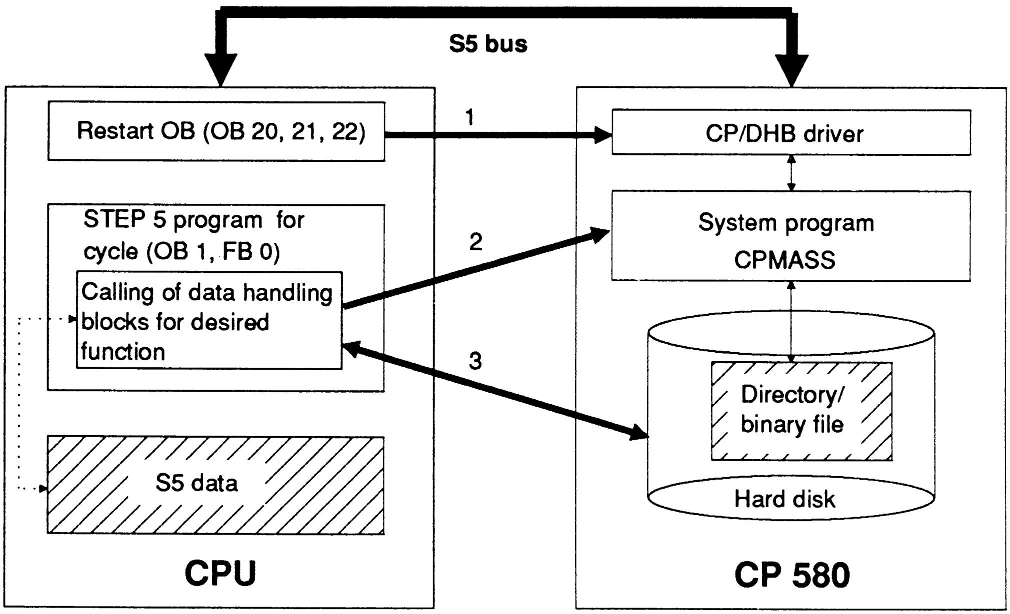
If the memory media existing on a CPU of your programmable controller are insufficient you can transfer data to the CP 580 using the mass storage functions and fetch these back - selectively if necessary - to the CPU as required. You can buffer data areas of up to 4 CPUs inserted in the same programmable controller as the CP 580 on the hard disk (or also on a floppy disk) of the CP 580 in binary form. Four hard disk directories (CPU1 to CPU4) are preset on the CP 580 for each of 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 all mass storage files (S5F files) of a directory on the hard disk of the CP 580 using a CPU.

5.2 Principle Sequences Between CPU and CP

Fig. 5.1 shows you the basic sequence of a mass storage function on the CPU and CP 580. Figs. 5.2 to 5.4 show you the basic data transfer between the CPU and CP 580, the switching over of a directory and the deletion of S5F files.

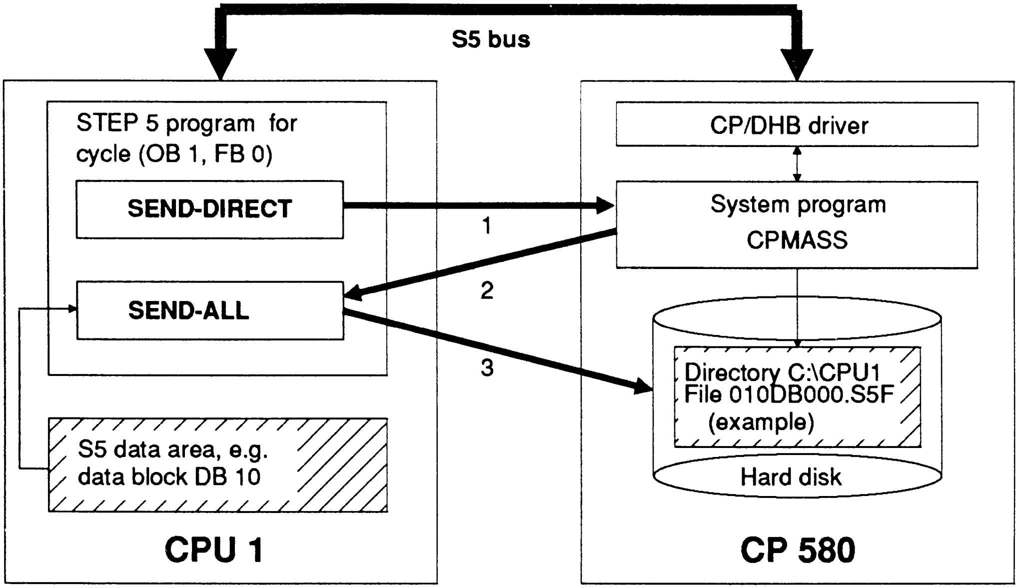


1 Synchronization of CPU during restart with CP via DHB SYNCHRON. The frame size for subsequent data transfer is set in the process.

2 Triggering of mass storage function.

3 Execution of mass storage function.

Fig. 5.1 Basic sequence of mass storage functions

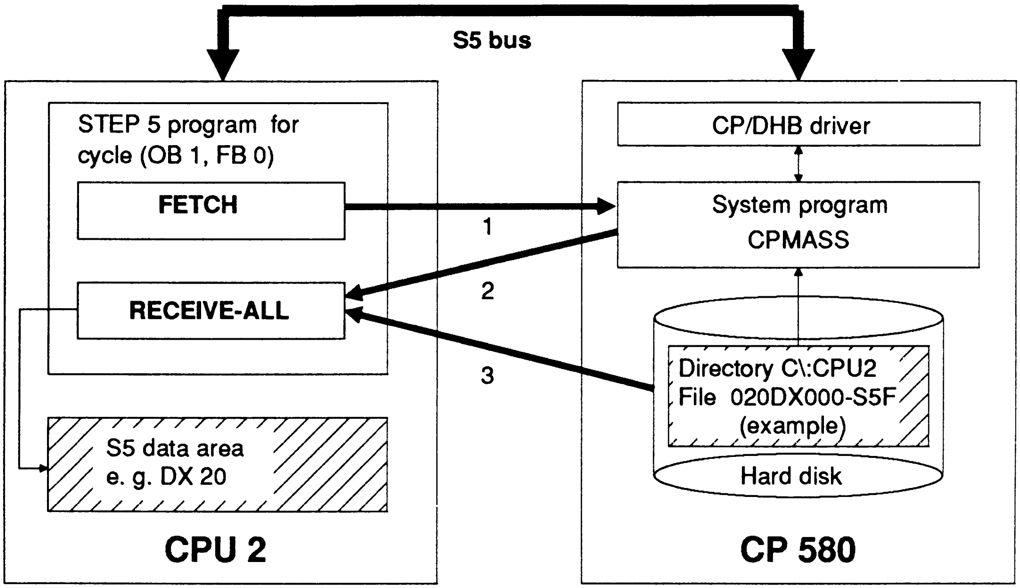


1 Call of DHB SEND/function SEND-DIRECT. The source parameters (including data block DB 10) are also defined here. By calling the DHB SEND, the S5 program activates the CP 580 and transfers the address of the data to the CPMASS system program via the CP/DHB driver. The CPMASS system program generates 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 CP/DHB 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 580 and stores them in the file 010B000.S5F (example) in the directory CPU1\ (example).

Fig. 5.2 Sequence when transmitting data from CPU 1 to the CP 580

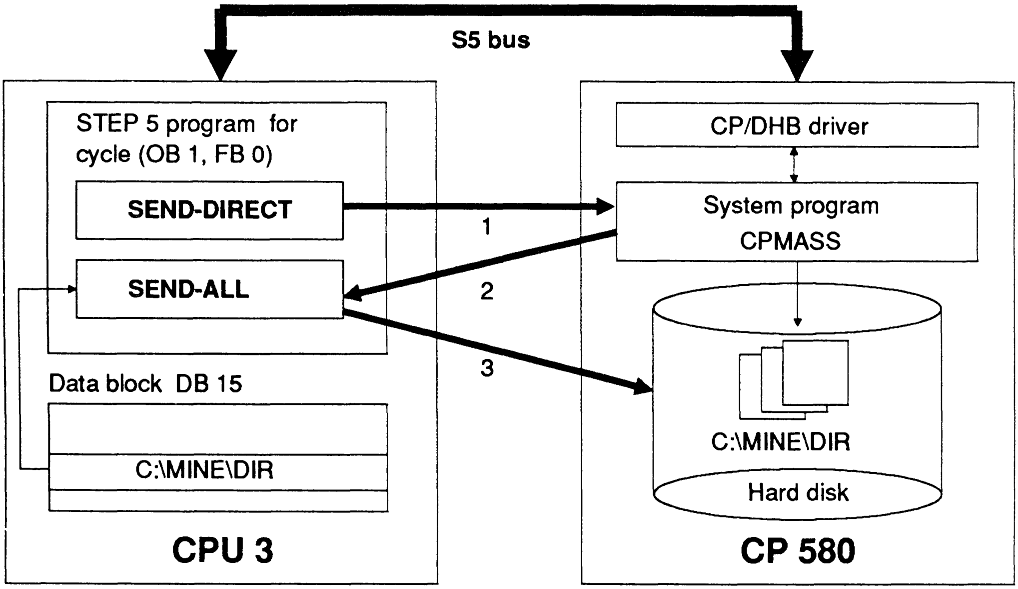


1 Call of DHB FETCH. The destination parameters (including data block DX 20) are also defined here. By calling the DHB FETCH, the S5 program activates the CP 580 and transfers the address of the data to the CPMASS system program via the CP/DHB driver. The CPMASS system program generates the file name 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 CP/DHB driver to read the data from the CP 580 and provided with the destination address.

3 RECEIVE-ALL reads the data from the CP 580 in the directory CPU2\ (example) from the file 020DX000.S5F and transmits these to CPU 2 (example) into data block DX 20 (example).

Fig. 5.3 Sequence when transmitting data from the CP 580 to CPU 2



1 Call of DHB SEND/function SEND-DIRECT. The address of a string (here in DB 15) is also specified. By calling the DHB SEND, the S5 program activates the CP 580 and transfers the string with the path name of the directory to the CPMASS system program via the CP/DHB driver.

2 The path name is transmitted to the CP 580.

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 S5F files from directory).

Fig. 5.4 Sequence when switching over a directory/deleting S5F files on the hard disk of the CP 580 from CPU 3

5.3 Mass Storage Function Operations

5.3.1 Related Procedures

You must carry out the following measures in order to transfer data between a CPU and the CP 580:

Table 5.1 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 580 are plugged into your programmable controller. The base interface No. for data transfer must be set on the CP 580.	PLC/ CP 580	5.3.2
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG/ CPU	5.3.3
Create or modify the STEP 5 program on the CPU for the cycle according to your applications of the mass storage functions (e.g. transmit or fetch data depending on a certain operating state).	PG/ CPU	5.3.3
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 (see Section 5.3.4).	PG/ CPU	5.3.4.2
Make sure that the CP/DHB driver is loaded in the memory of the CP 580 (cold restart of CP 580 if necessary). The mass storage functions are automatically started with a cold or warm restart (AUTOEXEC entry).	CP 580	--

Measure	on	See Section
Continuation of Table 5.1:		
Synchronize the CPU with the CP 580 by triggering a restart or warm 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 580.	PG/ CPU/ CP580	5.4.3

5.3.2 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 580 can transfer data with up to four CPUs via four pages. These pages must be numbered consecutively. The number of the first page is the base interface number.
  Define this number for the first CPU with which you wish to exchange data and set it on the CP 580 as described in Section 2.2.2. The pages for data transfer with three further CPUs then have consecutive numbers following the page with the base interface number.

5.3.3 Programming the CPU

5.3.3.1 Principle

Programming of the CPU for data transfer comprises the synchronization of the CPU 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.

Fig. 5.5 shows you the positions in your STEP 5 program at which you must call the two data handling blocks.

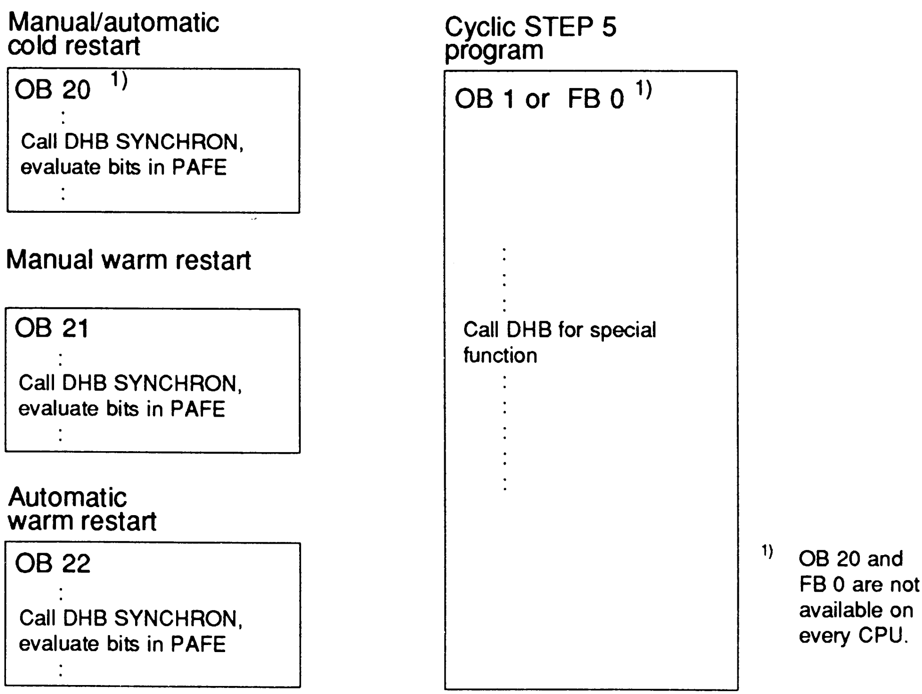


Fig. 5.5 Principle of DHB calls in the STEP 5 program of the CPU with the mass storage functions

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 statuses are to activate the various mass storage functions:
  - Following a cold restart (activation in the cycle, e.g. following scanning of bits)
  - Following a manual warm restart (activation in the cycle, e.g. following scanning of bits)
  - Following an automatic warm restart (activation in the cycle, e.g. 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 5.2 shows you the function block numbers of the DHBs for the various programmable controllers.

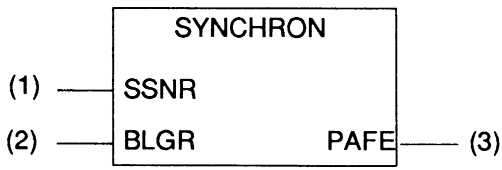
Table 5.2 DHB numbers on the various programmable controllers

PLC / DHB	S5-115U	S5-135U	S5-155U
SYNCHRON	FB 249	FB 125	FB 125
SEND	FB 244	FB 120	FB 120
RECEIVE	FB 245	FB 121	FB 121
FETCH	FB 246	FB 122	FB 122

5.3.3.2 Synchronization of the CPU

DHB SYNCHRON:

Block diagram



Use the block no.
FB xxx from
Table 5.2

Table 5.3 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 error bits (see Section 5.4)

Set the following parameters for the DHB SYNCHRON:

SSNA: Enter the number of the page via which you wish to read the data from the CPU: "(CPU No. - 1) + base interface number" (see Section 4.3.2).
Permissible values: 0 to 255 [1])

BLGR: Use this parameter to define the maximum number of bytes to be transmitted to the CP 580 in a CPU cycle.
Permissible values:
0 = 256 bytes (standard setting)
1 = 16 bytes
2 = 32 bytes
3 = 64 bytes
4 = 128 bytes
5 = 256 bytes
6 = 512 bytes

PAFE Byte address for parameter assignment error bits, e.g. FY 210. Bit no. 0 of the status byte is set to "1" in the event of a parameter error (refer to Section 5.4 for the meaning of the other status bits).

Note

The CP 580 requires 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).
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 580 then takes longer. You must decide which frame size is most favourable 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.)!

5.3.3.3 Transmission of Data from CPU to CP 580

In order to transmit data from the CPU to the CP 580, 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 5.2

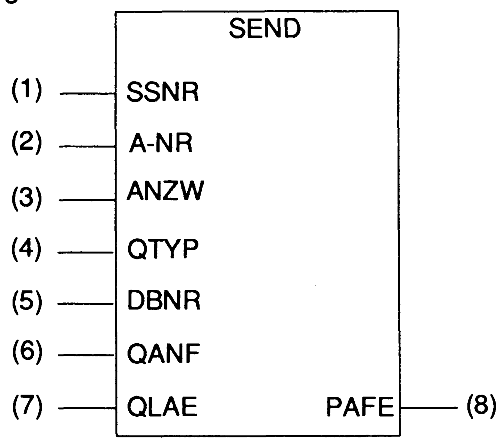


Table 5.4 Format and meaning of parameters for SEND

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no.
ANZW	I	W	Status word (see Section 5.4.2)
QTYP	D	KS	Data type of source

Parameter	Kind	Type	Meaning
Continuation of Table 5.4:			
DBNR	D	KY	Number of data block if the source is a data block
QANF	D	KF	Offset of first item of data to be read in the data area (byte or word No. e.g. DW 5 or FY 9)
QLAE	D	KF	Number of data units to be read (words or bytes)
PAFE	Q	BY	Parameter error bit (see Section 5.4.1)

SEND call with function SEND-DIRECT:

With the first call of the DHB SEND you activate the CP 580 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.** The CPMASS program generates the name of the destination file on the CP from the transferred address.
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 5.4).

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
FY for flag bytes
IA for process input image
QA for process output image
TA for timer cells
CA for counter cells
With **indirect** addressing:
XX (see DHB description)
RW (see Section 5.3.3.6)

QANF: Enter the number of the first item of data of the above type to be transmitted.

Permissible values: dependent on data type and CPU

QLAE: Specify the number of words (with DB or DX) or bytes (with the other data types) which are to be transmitted to the CP 580.

Permissible values: dependent on data type and CPU

PAFE: Byte address for parameter assignment error bits, e.g. FY 210. Bit no. 0 of the status byte is set to "1" in the event of a parameter assignment error (refer to Section 5.4 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 580. 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.
QTYP: DBNR: QANF: QLAE:	These parameters are irrelevant with SEND-ALL. They must be recorded for format reasons, however. Enter "0" for all of them.
PAFE:	As with SEND-DIRECT.

5.3.3.4 Transmission of Data from CP 580 to CPU

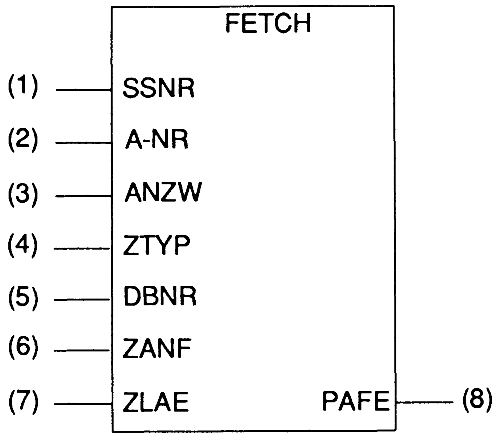
In order to transmit data from the CP 580 to the CPU, you must call the following DHB after synchronization of the CPU:

1. FETCH

2. RECEIVE/function RECEIVE-ALL

DHB FETCH:

Block diagram



Use the block no.
FB xxx from
Table 5.2

Table 5.5 Format and meaning of parameters for FETCH

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no.
ANZW	I	W	Status word (see Section 5.4.2)
ZTYP	D	KS	Data type of destination
DBNR	D	KY	Number of data block if destination is data block
ZANF	D	KF	Offset of first item of data to be written in the destination (byte or word No., e.g. DW 5 or FY 9)
ZLAE	D	KF	Number of data units to be written (words or bytes)
PAFE	Q	BY	Parameter error bits (see Section 5.4.1)

Calling of DHB FETCH:

By calling the DHB FETCH you activate the CP 580 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 flags (see Section 5.4).

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
FY for flag bytes
IA for process input image
QA for process output image
TA for timer cells
CA for counter cells
With **indirect** addressing:
XX (see DHB description)
RW (see Section 5.3.3.6)

ZANF: Enter the number of the first item of data of the above type to be written into the CPU.

Permissible values: dependent on data type and CPU

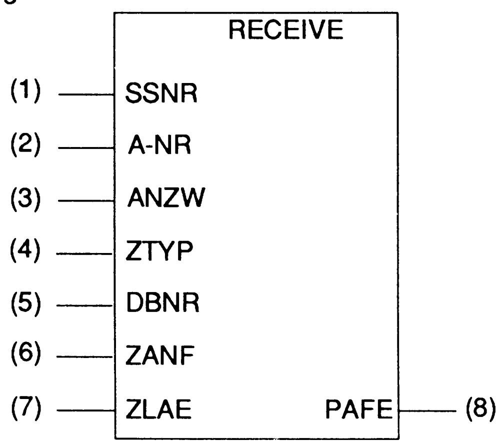
ZLAE: Specify the number of words (with DB or DX) or bytes (with the other data types) which are to be transmitted to the CPU.

Permissible values: dependent on data type and CPU

PAFE: Byte address for parameter assignment error bits, e.g. FY 210. Bit no. 0 of the status byte is set to "1" in the event of a parameter error (refer to Section 5.4 for the meaning of the other status bits).

DHB RECEIVE:

Block diagram



Use the block no. FB xxx from Table 5.2

Table 5.6 Format and meaning of parameters for RECEIVE/function RECEIVE-ALL

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no.
ANZW	I	W	Status word (see Section 5.4.2)
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 error bits (see Section 5.4.1)

RECEIVE call with function RECEIVE-ALL:

By calling the DHB RECEIVE with the function RECEIVE-ALL you trigger data transmission from the CP 580 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: Enter the address of **two** consecutive words as with FETCH.

ZTYP:	These parameters are irrelevant with
DBNR:	RECEIVE-ALL. They must be recorded for
ZANF:	format reasons, however.
ZLAE:	Enter "0" for all of them.

PAFE: As with SEND-DIRECT.

5.3.3.5 Preselection of Directory on CP 580 or Delete S5F Files

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 580 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 **S5F are deleted from the specified directory. The catalog** 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 580 and the **current working directory** (on the hard disk or floppy disk drive) **cannot** be deleted. Furthermore you cannot delete any **S5F 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 580 by mistake.

Programming of the two functions is similar to that described in Section 5.3.3.2. 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, e.g. 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 in Section 5.3.3.3.
A-NR:	Preselect directory: **201** Delete S5F files: **207**
ANZW:	As described in Section 5.3.3.2.
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 Fig. 5.6).

QLAE: Length of path name as number of words: QLAE ≥ ((1/2 number of characters) + 1).

PAFE: As described in Section 5.3.3.2.

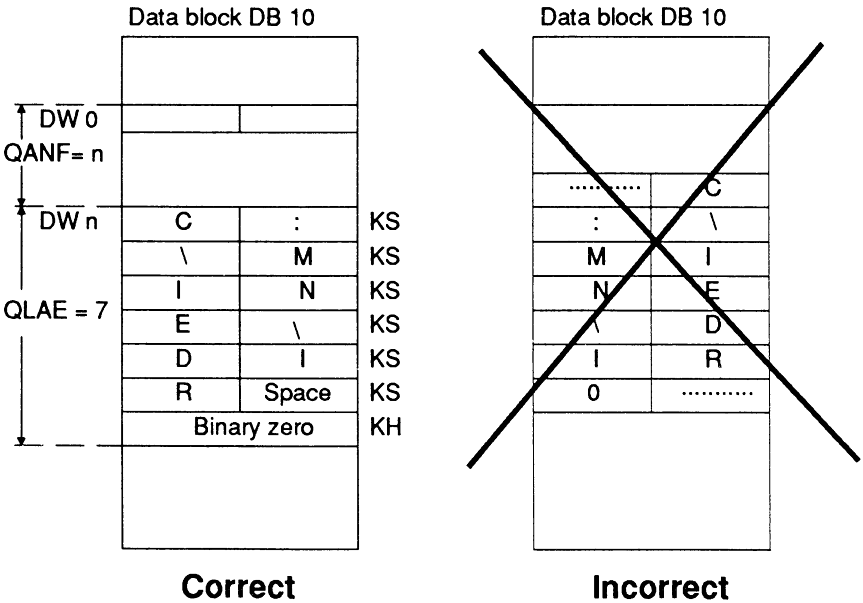


Fig. 5.6 Storage of path name in a data block

5.3.3.6 Indirect Parameterization "RW"

The DHB SEND/function DIRECT and FETCH enable you to characterize the data source and destination differently using indirect parameterization (e.g. source on CP 580 = data type IA and destination on a CPU = data type FY).
Proceed as follows:

1. Program the source and destination parameters in a data block DB or DX with the following format:

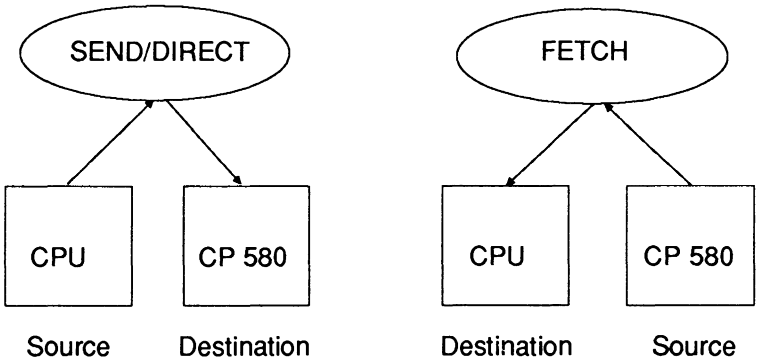
			Possibly other data		
QANF*	+0	KS	QTYP:	Type of data source, but not XX, RW and DX	**Source parameters**
	+1	KY	DBNR:	Only if QTYP DB and DX	
	+2	KF	QANF:	Initial address of data source	
	+3	KF	QLAE:	Length of data source	
	+4	KS	ZTYP:	Type of data destination, but not XX, RW and NN	**Destination parameters**
	+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,dno	If parameter in DB (dno = DB number)
	KY = 1,xno	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 580 file IAxxx.S5F is to be transmitted into DBy. 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.

5.3.3.7 Example of DHB Parameterization for Mass Storage Functions

Data from the CP 580 are to be read on the S5-135U and transmitted to the CPU 928B using the mass storage functions depending on a set input signal I 20.0 (switch):

- The data transmission is to take place at the rising edge of the input signal I 20.0.
- The CPU is the only one in the PLC and has the CPU number "1".
- 200 data words are to be read from the CP 580 (file name =000DB020.S5F) starting at DW 0 and transmitted into DB 20.

The STEP 5 program for this task consists of the following parts:

a) The CPU must be synchronized with the CP 580 in the restart blocks OB 20/21/22. The DHB SYNCHRON is called in FB 111 for this purpose. The programmable controller goes to STOP if synchronization has not been achieved following a maximum of three calls of the DHB SYNCHRON (approx. 30 s).
(The CP 580 requires more time to run up than the CPU; therefore provide a program loop for several attempts.)

b) The DHB FETCH is called in OB 1 following edge evaluation of the input switch I 20.0. The actual data transmission is carried out using the DHB RECEIVE/function RECEIVE-ALL.

The DHBs to be used for the CPU 928B have the following function block numbers:

DHB SYNCHRON:	FB 125
DHB FETCH:	FB 122
DHB RECEIVE:	FB 121

a) **STEP 5 operations in OB 20, OB 21, OB 22:**

```
         :                     STEP 5 OPERATIONS FOR
         :                     OTHER FUNCTIONS
         :                     STEP 5 OPERATIONS FOR DATA
         :                     EXCHANGE WITH CP 580:
         : JU       FB 111     CALL SYNCHRON BLOCK
NAME     : CPSYNC
REP      :          KF + 3     NUMBER OF ATTEMPTS:
         :                     STEP 5 OPERATIONS FOR
         :                     OTHER FUNCTIONS
```

Function block FB 111:

```
   SEGMENT 1
NAME     : CPSYNC
DECL     : REP      I/Q/D/B/T/C: D    KM/KH/KY/KS/KF/KT/KC/KG: KF
         : L        KF +0
         : T        FW 10      INITIALIZE LOOP
         :                     COUNTER
LOOP     :JU        FB 125     DHB SYNCHRON
NAME     :SYNCHRON
SSNR     :          KY 0,16    INTERFACE NO. OF CP 580 (PAGE NO. = 16)
BLGR     :          KY 0,0     FRAME SIZE = 256 BYTE
PAFE     :          FY 12      ERROR BYTE
         : L        KB 0
         : L        FY 12
         : != F                SYNCHRONIZATION SUCCESSFUL?
         : JC       = END      YES: LEAVE BLOCK
         : L        FW 10      INCREMENT......
         : I        1          ... LOOP COUNTER ........
         : T        FW 10      ... FW 10
         : LW       = REP      REPETITION FACTOR ...
         : < = F               ... ≤ LOOP COUNTER FACTOR?
         : JC       = LOOP     NEW ATTEMPT
         : STP                 PLC = STOP
END      : BE
```

b) **STEP 5 operations in OB 1:**

```
        :                  STEP 5 OPERATIONS FOR
        :                  OTHER FUNCTIONS
        :                  STEP 5 OPERATIONS FOR DATA
        :                  EXCHANGE WITH CP 580:
        : A      I 20.0    EDGE EVALUATION OF
        : AN     M 20.0    INPUT SIGNAL
        : =      M 20.1
        : S      M 20.0
        : AN     I  20.0
        : R      M 20.0
        : A      M 20.1
        : JU     FB 122
NAME    : FETCH
SSNR    :        KY 0,16   INTERFACE NUMBER
A-NR    :        KY 0,1    JOB NUMBER = 1
ANZW    :        FW 250    STATUS WORD
ZTYP    :        KS DB     DESTINATION: DATA BLOCK 'DB'
DBNR    :        KY 0,20   DB 20
ZANF    :        KF +0     WORD No. 0 ONWARDS
ZLAE    :        KF +200   WRITE 200 WORDS
PAFE    :        FY 13     PARAMETER ERROR
        :
        : JU     FB 127    CALL RECEIVE/FUNCTION RECEIVE-ALL
NAME    : REC-A
SSNR    :        KY 0,16   PAGE NO. = SSNR = 16
A-NR    :        KY 0,0    RECEIVE-ALL ID
ANZW    :        FW 246    STATUS WORD
PAFE    :        FY14      PARAMETER ERROR
        :
        :                  STEP 5 OPERATIONS FOR
        :                  OTHER FUNCTIONS
        :
```

 C79000-B8576-C204-01

5.3.4 Activation and Testing of the Mass Storage Functions

The AUTOEXEC.BAT file in the main directory of the hard disk is set when the CP 580 is delivered such that the CPMASS program is started for the mass storage functions with each system restart/warm restart of the CP 580. The program then expects to be triggered by a CPU in the same programmable controller as the CP 580 in order to then transfer data with it.

⇒ The CPMASS program outputs the following text on the monitor when started:

CPMASS.Vxx
CPDHB.Vxx

The program remains installed following a system restart/warm restart of the CP 580. 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 580.

5.3.4.1 File Names 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 file names for these data on the CP 580 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 580.

Table 5.7 Names and lengths of the CP 580 files for S5 data

S5 data area	S5 data type	Name of CP 580 file [1]	File length
Data blocks DB	DB	nnnDBaaa.S5F	8192 bytes
Extended data blocks DX	DX	nnnDXaaa.S5F	8192 bytes

S5 data area	S5 data type	Name of CP 580 file [1]	File length
Continuation of Table 5.7:			
Flag area F	FA	FAaaa.S5F	256 bytes
Process output image	QA	QAaaa.S5F	128 bytes
Process input image	IA	IAaaa.S5F	128 bytes
Counter cells	CA	CAaaa.S5F	512 bytes
Timer cells	TA	TAaaa.S5F	512 bytes

1) nnn = DB/DX number, aaa = job number

Note

When **writing** an S5 data area into a file, the file is always created in the length assigned to the data type (see Table 5.7). If the length of the transmitted data is **not utilized,** the **rest** of the file is filled with **binary zeros.** You must observe this if you read data out of the CP 580 file again.

5.3.4.2 Testing

Proceed as follows to establish whether your required data transfer between the CPU and CP 580 is being executed correctly:

1. Prepare a data block for the test with static data (not equal to zero!) on the PG (e.g. DB 20) and transfer the block using the PG 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 580 (e.g. 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 5.4).

4. Check whether the associated file has been created (for DB 20 = 020DB000.S5F) in the preset directory following execution of the STEP 5 program on the CP 580 (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 (e.g. file 020DB000.S5F from data block DB 20 into the file for data block DB 21 with the MS-DOS command "COPY 020DB00.S5F 021DB000.S5F").

6. Write the STEP 5 program such that it reads data once from the copied CP 580 file into the corresponding data block (e.g. the file 021DB000.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 (e.g. access other S5 data areas) or program the DHB calls for the mass storage function "Set/delete directory" if required.

5.4 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:

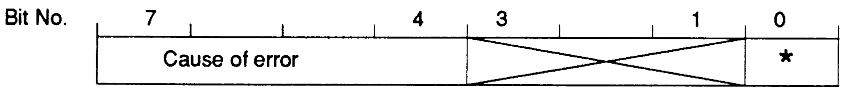
- Parameter assignment error bits
- Job status bits

5.4.1 Parameter 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 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 error, more details in bits 4 to 7.

Table 5.8 lists all bits which can be stored by the data handling blocks in the PAFE.

Table 5.8 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 does not exist or is 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

5.4.2 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 CP/DHB driver. You must specify the initial address of the status words in the parameters ANZW of the DHB calls.

Some of the status bits of the DHB SEND/function SEND-DIRECT and the DHB FETCH are important for the test. These status bits will be 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:

<table>
<tr><td>Bit No.</td><td>15 12</td><td>11 8</td><td>7 4</td><td>3 0</td></tr>
<tr><td>ANZW</td><td></td><td>CPMASS error number</td><td>Status bits from data management</td><td>Status bits from job management</td></tr>
<tr><td>ANZW+1</td><td colspan="4">Number of transmitted bytes</td></tr>
</table>

Status bits from job management (bit nos. 0 to 3):

You can obtain the respective status of a job from these status bits.

Table 5.9 Job status bits of CPMASS

Bit No.	Meaning
0	RECEIVE meaningful
1	Bit = 1: Job for data transfer taking place
2	Bit = 1: Job completed without errors
3	Bit = 1: Job completed with errors

If bits 1 and 2 are set simultaneously in the status word, you cannot send jobs to the CP 580 for the respective job number. Remember this response during the restart of your S5 program (see also Section 5.2.8).

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 5.10 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.
Table 5.11 shows you the status bits which may occur during execution of the mass storage functions.

Table 5.11 Error bits from data management

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	CP/DHB error

Value of bit No. 8 to bit No. 11	Cause of error
Continuation of Table 5.11:	
7H	S5 segment disabled (AS, RS, PY)
8H	QLAE/ZLAE too large
9H	CP 580 file write-protected (send job from CPU)
AH	Insufficient space on the set drive of the CP 580 (send job from CPU)
BH	File/directory not found on CP 580 (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 (job "Delete S5F files" from CPU)
EH	Timeout (after 50 seconds)
FH	Parameter error

Contents

Contents

6 Command Interpreter

This chapter describes how you can execute MS-DOS commands on the CP 580 using a CPU inserted together with the CP 580 in your programmable controller.
The chapter describes all measures and operations required on the CP 580 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 580 (see Section 6.3.4). 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.

6.1 Application

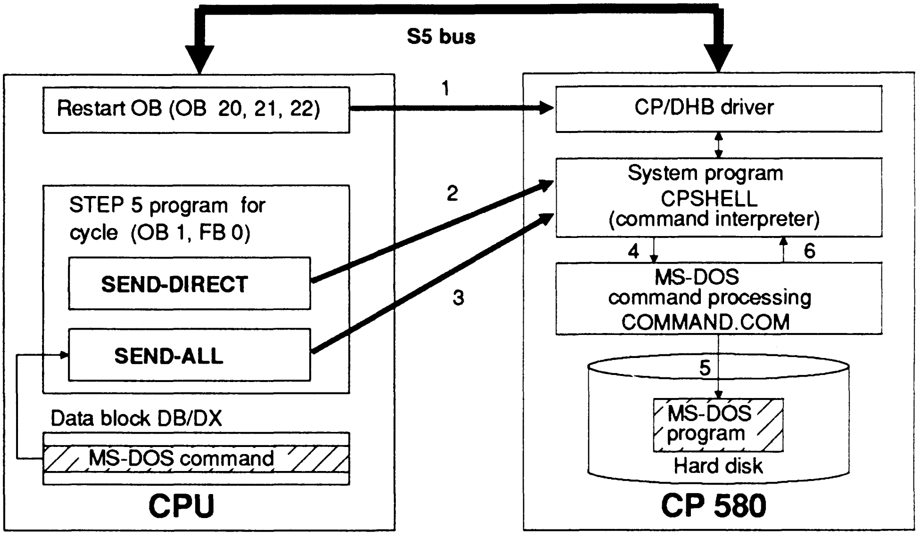
You can use the function "Command interpreter" with a CPU to execute on the CP 580 any MS-DOS commands stored in a data block DB or DX. For example, if you use your CP 580 without a keyboard because you only use the "Mass storage functions", the command interpreter can be used if necessary to copy files from the CPU onto the CP or to trigger other MS-DOS functions as required.
Refer to the information in Section 6.3.3 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 580.

6.2 Principle Sequences Between CPU and CP

Fig. 6.1 shows you the basic sequence of command interpretation on the CPU and CP 580.



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 commmand 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 the command has been executed or when the started program has been terminated, and the STEP 5 program receives a ready signal.

Fig. 6.1 Basic sequence of a command interpretation

6.3 Command Interpreter Operations

6.3.1 Related Procedures

You can use the CP 580 to execute commands which you have stored in an S5 data block DB or DX.
You must carry out the following measures:

Table 6.1 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 580 are plugged into your programmable controller. The base interface no. for data transfer must be set on the CP 580.	PLC/ CP 580	6.3.2
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.	--	6.3.3
Program a data block DB or DX with the commands you wish to execute on the CP 580.	PG/ CPU	6.3.4.1
Create or modify the restart organization blocks on the CPU such that the DHB SYNCHRON is called in each OB.	PG/ CPU	6.3.4.3
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	6.3.4.3
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	6.3.6.3 and 6.4

 C79000-B8576-C204-01

Measure	on	See Section
Continuation of Table 6.1:		
Make sure that the CP/DHB driver is loaded in the memory of the CP 580. The command interpreter is automatically started with a cold or warm restart (entry in AUTOEXEC). If the entry in the AUTOEXEC is missing you can start the driver with the CPDHB command.	CP 580	--
Synchronize the CPU with the CP 580 by triggering a restart or warm restart of the CPU.	CPU	--
Check the correct sequence when transferring a command by testing on the CPU and CP.	PG/ CPU/ CP 580	6.3.4.3 and 6.4

6.3.2 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 580 can transfer data with up to four CPUs via four pages. These pages must be numbered consecutively. The number of the first page is the base interface number.
  Define this number for the first CPU with which you wish to exchange data, and set it on the CP 580 as described in Section 2.2.2.1. The pages for data transfer with three further CPUs then have consecutive numbers following the page with the base interface number.

6.3.3 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 580 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 580 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 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 (e.g. 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).

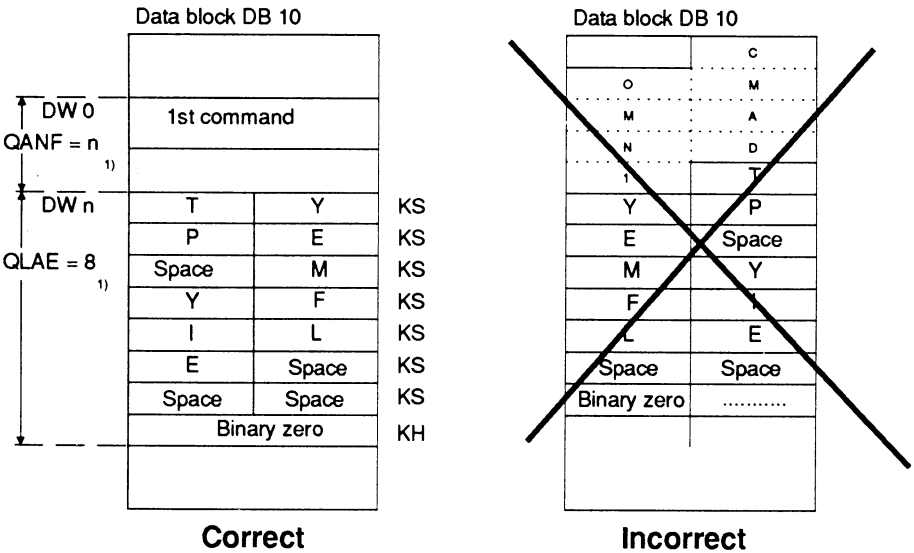
Caution

The **FORMAT** should **not be used at all** with the command interpreter, the **DELETE** command only **very carefully.**

6.3.4 Programming the CPU

6.3.4.1 Storing Commands in Data Block

Once you have defined 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 6.3.4.3) enter its initial address in the data block. This must be a **word** address (see Fig. 6.2).



1) For later calling of DHB SEND/function SEND-DIRECT

Fig. 6.2 Storage of a command in the data block

6.3.4.2 STEP 5 Operations for the Command Interpreter

Programming of the CPU for application of the command interpreter comprises synchronization of the CPU 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 SYNCRHON for synchronization and the DHB SEND for command transfer.

Fig. 6.3 shows you the positions at which you must call the two data handling blocks in your STEP 5 program.

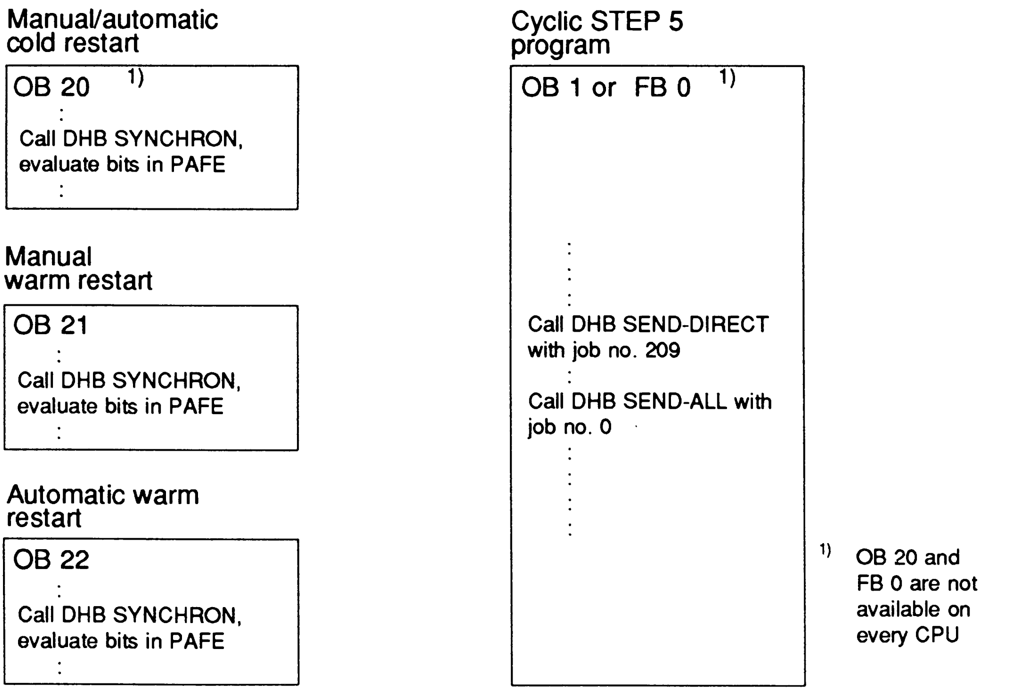


Fig. 6.3 Principle of DHB calls in the STEP 5 program of the CPU when using the command interpreter

6.3.4.3 Calling and Parameterizing the Data Handling Blocks

The functions of the data handling blocks which you need to execute commands on the CP 580 are described in this section as far as is necessary for programming. You can obtain further information on the data handling blocks from the corresponding descriptions of the DHBs. These descriptions are available for the S5-135U and the S5-155U programmable controllers (see Catalog for Order Nos.). The description of the DHBs for the S5-115U programmable controller can be found in the S5-115U Manual.

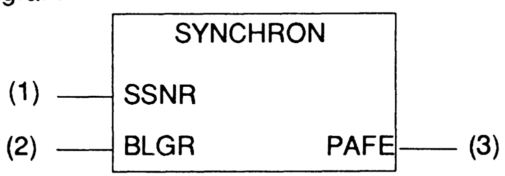
The data handling blocks have different block numbers on the various programmable controllers. The following table contains the numbers of the DHBs SYNCHRON and SEND which you require for the various programmable controllers.

Table 6.2 DHB numbers on the various programmable controllers

PLC / DHB	S5-115U	S5-135U	S5-155U
SYNCHRON	FB 249	FB 125	FB 125
SEND	FB 244	FB 120	FB 120

DHB SYNCHRON:

Block diagram



Use the block no.
FB xxx from
Table 6.2

Table 6.3 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 error bits (see Section 6.4)

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 6.3.2).
Permissible values:
0 to 255 [1])

BLGR: Use this parameter to define the maximum number of bytes to be transmitted to the CP 580 in a CPU cycle.
Permissible values:
0 = 256 bytes (standard setting)
1 = 16 bytes
2 = 32 bytes
3 = 64 bytes
4 = 128 bytes
5 = 256 bytes
6 = 512 bytes

PAFE: Byte address for parameter assignment error bits, e.g. FY 210. Bit no. 0 of the status byte is set to "1" in the event of a parameter error (refer to Section 6.4 for the meaning of the other status bits).

1) The base interface no. must be set in steps of four (0, 4, 8, 12 etc.)!

> **Note**
>
> The CP 580 requires 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).

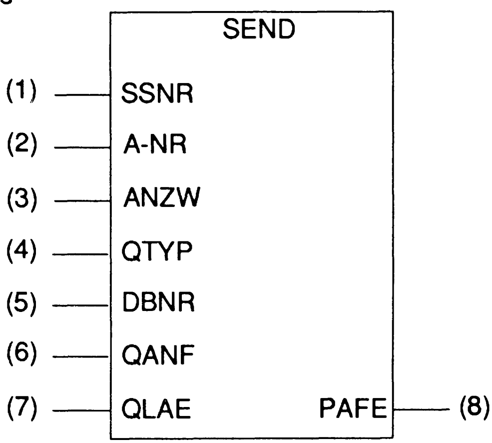
Send command to CP 580 and execute:

In order to send a command from the prepared data block to the CP 580 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 6.2

Table 6.4 Format and meaning of parameters for SEND

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no.
ANZW	I	W	Status word (see Section 6.4)
QTYP	D	KS	Type of data block
DBNR	D	KY	Number of data block
QANF	D	KF	Initial address of command in data block (word No. e.g. DW 5)
QLAE	D	KF	Length of command as number of words
PAFE	Q	BY	Parameter error bits (see Section 6.4)

SEND call with function SEND-DIRECT

With the first call of the DHB SEND you activate the CP 580 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 6.4). 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 6.4).

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 Fig. 6.2) of the command.

Permissible values: dependent on CPU

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 Fig. 6.2).

Permissible values: dependent on CPU

PAFE: Byte address for parameter assignment error bits, e.g. FY 210. Bit no. 0 of the status byte is set to "1" in the event of a parameter error (refer to Section 6.4 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 580. 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.

QTYP: These parameters are irrelevant
DBNR: with SEND-ALL. They must be recorded
QANF: for format reasons, however.
QLAE: Enter "0" for all of them.

PAFE: As with SEND-DIRECT, but a different address since the function SEND-DIRECT supplies its own status bits.

Note

With the S5-135U and S5-155U programmable controllers you can use a special DHB SEND-A instead of the call SEND/function SEND-ALL. With this DHB you need not note the irrelevant parameters (see above). See Section 9.2.1 for the FB numbers of these DHBs.

6.3.5 Example

The command "TYPE MYFILE" is to be executed on the CP 580 on the S5-135U programmable controller by means of a CPU depending on a set input signal I 127.0 (switch):

- The command is only to be executed at the rising edge of the input signal I 127.0.
- The CPU is the only one in the programmable controller and has the CPU number "1".
- The command is programmed in DB 5 starting at data word 10. In order to program a "Zero termination" easily, the command string is extended by a space 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 Fig. 6.2).

The STEP 5 program for this task consists of the following parts:

a) The CPU must be synchronized with the CP 580 in the restart blocks OB 20/21/22. The DHB SYNCHRON is called in FB 111 for this purpose. The PLC goes to STOP if synchronization has not been achieved after a maximum of three calls of the DHB SYNCHRON (approx. 30 s).
(The CP 580 requires more time to run up than the CPU; therefore provide a program loop for several attempts.)

b) The DHB SEND/function SEND-DIRECT is called in OB 1 following edge evaluation of the input switch I 127.0. The command is transmitted by means of the DHB SEND/function SEND-ALL.

c) Data block DB 5 with the command starting at DW 10.

The DHBs to be used for the CPU 928B have the following function block numbers:

DHB SYNCHRON:	FB 125
DHB SEND:	FB 120

a) STEP 5 operations in OB 20, OB 21, OB 22:

```
        :                         STEP 5 OPERATIONS FOR
        :                         OTHER FUNCTIONS
        :
        :                         STEP 5 OPERATIONS FOR DATA
        :                         EXCHANGE WITH CP 580:
        : JU   FB 111             CALL SYNCHRON BLOCK
NAME    : CPSYNC
REP     :            KF + 3       NUMBER OF ATTEMPTS
        :
        :                         STEP 5 OPERATIONS FOR
        :                         OTHER FUNCTIONS
```

Function block FB 111:

```
   SEGMENT1
NAME    : CPSYNC
DECL    : REP        I/Q/D/B/T/C: D    KM/KH/KY/KS/KF/KT/KC/KG: KF
        : L          KF +0
        : T          FW 10        INITIALIZE LOOP COUNTER
LOOP    :JU          FB 125       DHB SYNCHRON
NAME    :SYNCHRON
SSNR    :            KY 0,16      INTERFACE NO. OF CP 580 (PAGE NO. = 16)
BLGR    :            KY 0,0       FRAME SIZE = 256 BYTE
PAFE    :            FY 12        ERROR BYTE
        : L          KB 0
        : L          FY 12
        : != F                    SYNCHRONIZATION SUCCESSFUL?
        : JC         = END        YES: LEAVE BLOCK
        : L          FW 10        INCREMENT ......
        : I          1            ...     LOOP COUNTER ...............
        : T          FW 10        ...     FW 10
        : LW         = REP        REPETITION FACTOR ...............
        : < = F                   ...     ≤ LOOP COUNTER FACTOR?
        : JC         = LOOP       NEW ATTEMPT
        : STP                     PLC = STOP
END     : BE
```

b) **STEP 5 operations in OB 1:**

```
        :                      STEP 5 OPERATIONS FOR
        :                      OTHER FUNCTIONS
        :
        :                      STEP 5 OPERATIONS FOR DATA
        :                      TRANSFER WITH CP 580:
        : A       I 127.0      EDGE EVALUATION OF
        : AN      M 20.0       INPUT SIGNAL
        : =       M 20.1
        : S       M 20.0
        : AN      I 127.0
        : R       M 20.0
        :
        : A       M 20.1
        : JU      FB 120
NAME    : SEND
SSNR    :         KY 0,16      INTERFACE NUMBER
A-NR    :         KY 0,209     JOB NUMBER = 209
ANZW    :         FW 244       STATUS WORD
QTYP    :         KS DB        SOURCE: DATA BLOCK 'DB'
DBNR    :         KY 0,5       DB 5
QANF    :         KF +10       WORD No. 10 ONWARDS
QLAE    :         KF +8        READ 8 WORDS
PAFE    :         FY 13        PARAMETER ERROR
        :
        : JU      FB 120       CALL SEND/FUNCTION SEND-ALL
NAME    : SEND
SSNR    :         KY 0,16      PAGE NO. = SSNR = 16
A-NR    :         KY 0,0       SEND-ALL ID
ANZW    :         FW 252       STATUS WORD
QTYP    :         KS           NO SIGNIFICANCE
DBNR    :         KY 0,0       NO SIGNIFICANCE
QANF    :         KF +0        NO SIGNIFICANCE
QLAE    :         KF +0        NO SIGNIFICANCE
PAFE    :         FY 14        PARAMETER ERROR
        :                      STEP 5 OPERATIONS FOR
        :                      OTHER FUNCTIONS
        :
```

c) **Data block DB 5:**

(see also Fig.6.2)

DB 5

```
 0 :                          OTHER COMMAND
 1 :
 2 :
   :
   :
10 :   KS = 'TYPE MYFILE ':   COMMAND CHARACTER + 1 SPACE
17 :   KH = 0000:             TERMINATE WITH BINARY ZERO
18 :                          FURTHER COMMANDS
19 :
   :
   :
```

6.3.6 Activation and Testing of the Command Interpreter

6.3.6.1 Activation

You can activate the command interpreter (initially for the test) once you have carried out all preparatory measures on the CPU and CP 580:

- Make sure that the entry for loading the CP/DHB driver (CPDHB) is present in the AUTOEXEC.BAT file in the main directory of the CP 580 hard disk. [1])
- Trigger a warm restart on the CP 580 in order to start the CP/DHB driver. [1])
- Carry out a warm or cold restart on the CPU in order to synchronize it with the CP 580.
- Start the command interpreter on the CP 580 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 580 hard disk).

⇒ The command interpreter outputs the following message on the monitor when started:

CP 580 S5 command interpreter Vx.x
Copyright (c) Siemens AG 1991
CPSHELL ready

MS-DOS commands are now sent to the CP 580 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 580 monitor as with a keyboard input.

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 by entering **CPDHB** on the keyboard.

If messages are output by the called MS-DOS function [1]) or the started program, these also appear on the monitor.

The program CPSHELL is terminated by pressing the ESC key.

6.3.6.2 Testing

Proceed in steps to establish whether a command stored on the CPU is sent correctly to the CP 580 and executed:

1. Store a simple, "non-dangerous" 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 6.4).

4. Check that the command appears on the monitor of the CP 580 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 6.5 for handling of acknowledgement signals.

6.4 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
- Job status bits

6.4.1 Parameter 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 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:

Bit No.	7			4	3		1	0
	Cause of error							*

* = Common flag: 0: no error
1: parameter error, more details in bits 4 to 7

Table 6.5 lists all bits which can be stored by the data handling blocks in the PAFE.

Table 6.5 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 does not exist or is 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

6.4.2 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 CP/DHB 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 9.

The status words of DHB SEND/SEND-DIRECT have the following format:

Bit No.	15 12	11 8	7 4	3 0
ANZW		CPSHELL error no.		Status bits from job management
ANZW+1	Number of transmitted bytes			

Flags from job management (bit nos. 0 to 3):

You can obtain the respective status of command processing from these status bits:

Table 6.6 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

If bits 1 and 2 are set simultaneously in the status word, you cannot send jobs to the CP 580 for the respective job number. Remember this response during the restart of your S5 program (see also Section 9.2.8).

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 6.7 shows you the status bits which may occur when processing a command.

Table 6.7 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
7H	Not yet realized
8H	Not used by CPSHELL
9H	The called program cannot be loaded since the vacant memory space in the CP 580 is insufficient
AH	The called MS-DOS function or the started program address a peripheral device of the CP 580 which is not ready, e.g. the floppy disk drive does not contain a disk, the printer is not ready or the floppy disk is not formatted.

6.5 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 580.

- Handling of MS-DOS acknowledgement bits:
  If a bit is to be output which can be acknowledged by MS-DOS, because e.g. 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 6.4.2).

- 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 (e.g. COPY) from several CPUs.

Contents

Contents

7 Free Programming of the CP 580

This chapter is intended for CP 580 users who wish to use the functions of the CP/DHB driver and the S5 data handling blocks for special applications.
If you wish to program your own CP 580 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 7.2 informs you of the steps neccessary to solve your problem.
Section 7.3 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 580.
Section 7.4 informs you of the CP/DHB driver, interaction between the CPU and CP 580 and the required programming of the driver functions.
Information on testing your application can be found in Section 7.5.

7.1 Application

Free programming of the CP 580 enables you to solve special tasks which may be necessary for communication between CPUs and the CP 580, e.g. data transfer with higher-level computers or operator inputs for a CPU (e.g. modification of limits).

The CP 580 program enables communication with up to 4 CPUs. The CPUs must be plugged into the same programmable controller as the CP 580.

7.2 Procedure

7.2.1 Summary

Proceed in the order recommended in Table 7.1 to solve your special application.

Table 7.1 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 580 and the CPU(s). When trying for the first time, only use communication with **one** CPU.	--	7.2.2
Note which data handling blocks you required to solve the communication task and how these are parameterized.	--	7.3
Program your STEP 5 program with calls of the required DHBs.	PG	--
Note for the CP/DHB 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?	--	7.4
Program the MS-DOS program with which you wish to handle the communication.	CP 580	7.4
Make sure that the CPU with which you wish to exchange data and the CP 580 are inserted in your programmable controller and that the CP/DHB driver is loaded in the memory (TSR).	CP 580	2.2.3 and 7.4
Test the correct execution of your STEP 5 program initially without your MS-DOS program but with the CP/DHB driver (synchronization of CP 580 with CPU).	PG/ CPU/ CP 580	7.5

Measure	on	See Section
Continuation of Table 7.1:		
Now test your MS-DOS program and the interaction with the data handling blocks on the CPU.	PG/ CPU/ CP 580	7.5
Now extend your application - if necessary - for communication with several CPUs.	PG/ CPU/ CP 580	--

7.2.2 Analysis of Task

Before you read further in the next sections which functions of the data handling blocks and the CP/DHB 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?
  a) From the CPU to the CP 580 (CPU **sends** data)?
  b) From the CP 580 to the CPU (CPU **receives** data)?
  c) 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 7.3 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 7.4.

Note

The system programs supplied for "Mass storage functions" and "Command interpreter" occupy the following job numbers for your applications:

- Mass storage functions:	1 to 99 201 207
- Command interpreter:	209

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.

7.3 Programming of DHB Calls

Detailed knowledge on the data handling blocks (DHB) is required in order to program data transfer functions with the CP/DHB 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, e.g. the various sizes of the areas, can be obtained from the DHB descriptions for the various programmable controllers (see Reference Literature).

7.3.1 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. 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 7.3.4 and 7.3.12). 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.

7.3.2 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 CP/DHB driver can use the SEND-ALL mode to request the SEND block to transmit data areas from the CPU to the CP. The CP/DHB 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 CP/DHB driver can use the RECEIVE-ALL mode to request the RECEIVE block to transmit data areas from the CP to the CPU. The CP/DHB 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.

7.3.3 Parameters of the Data Handling Blocks

Table 7.2 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 or data destination
DBNR	Data block number
QANF/ZANF	Relative initial address within the area
QLAE/ZLAE	Number of source data or destination data
PAFE	Parameter error
BLGR	Frame size

7.3.4 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	Direct parameterization
			y = 0..255	Interface number (page number)
			$x \neq 0$	Indirect parameterization
			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 doubleword in which the processing status of a specific job is displayed. Section 7.3.6 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: DW, FW

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	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.
		KS = NN	**Without parameterization:** no data on the data source or destination. The effect of the job is determined only by the job number.
		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.
		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.

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	KY = 0,y 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.

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	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. ZANF is used analog to QANF to specify the destination data area. The permissible range of values depends on the CPU used. Please refer to the DHB description for your CPU.

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		QLAE is used to define the length of the source data area, ZLAE the length of the destination data area. The permissible range of values depends on the CPU used. Please refer to the DHB description for your CPU.
		-1	-1 means "joker length" With RECEIVE, all the data delivered by the transmitter are accepted, or as many as permitted by the range limit. With SEND, data are transmitted until the range limit has been reached.

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 byte

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 7.3.6.4.

Parameter type	Format	Assignment
Address (byte)	BY	Possible areas QA, FA The size of the areas is CPU-specific

7.3.5 Direct and Indirect Parameterization

7.3.5.1 Indirect Parameterization of SSNR, A-NR, ANZW and BLGR

The High byte of the parameter SSNR serves as a switchover 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 ≠ 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 (e.g. 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).

7.3.5.2 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          Indirect parameterization begins at DW 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
5:            :                        in the flag words FW 200 and FW 202
```

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 44
             :
             :JU    FB 244       (FB 244 only with S5-115U)
NAME         :SEND
SSNR         :KY 255,1           Indirect parameterization begins at DW 1
A-NR         :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
```

Example 3:

The parameters

SSNR and
BLGR

are to be addressed indirectly during calling of the DHB SYNCHRON.

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        Indirect parameterization begins at DW 100
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
```

7.3.5.3 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 7.3 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 7.4 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)
		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)

7.3.6 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 doubleword which is addressed by the parameter ANZW. The second part of the doubleword is the so-called length word.

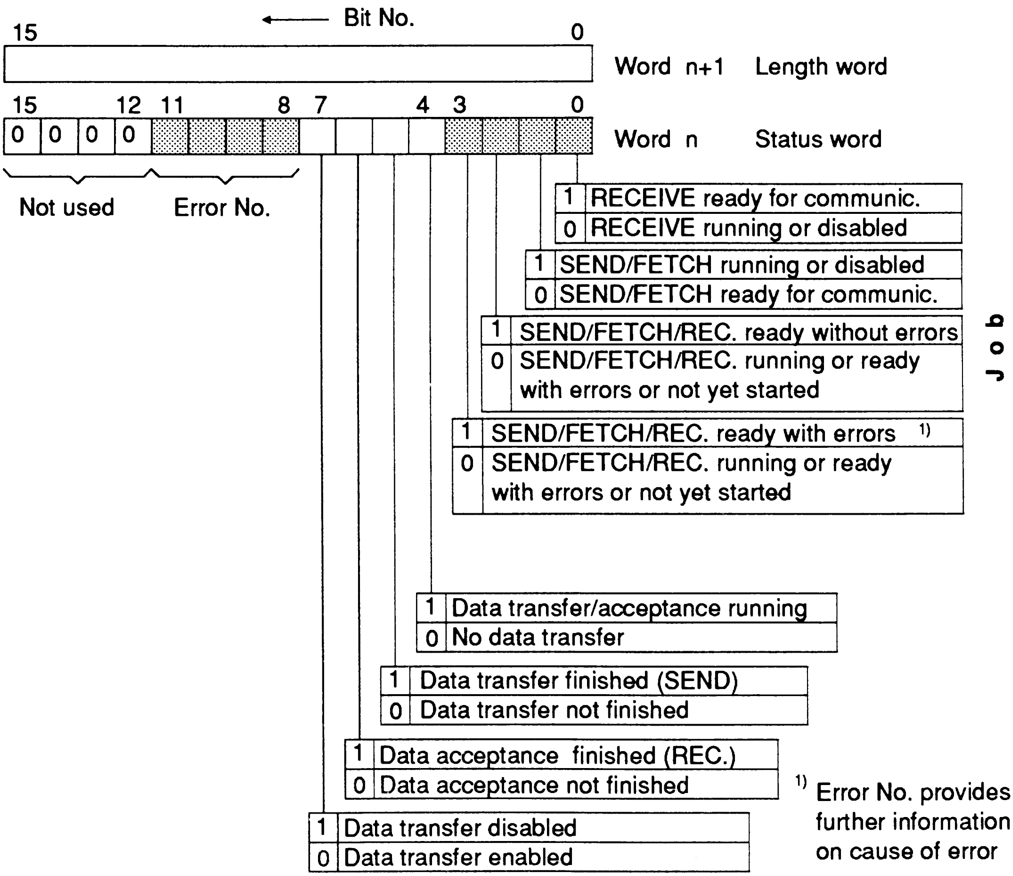


Fig. 7.1 Format of DHB status words

> **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.

7.3.6.1 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 7.5 Parameter values with indirect addressing with RW

Bit No.	Set	Delete/ overwrite	DHB evaluation	User evaluation
0	DHB CP application signals data ready	DHB CP/DHB 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 CP/DHB 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 finished

Bit No.	Set	Delete/ overwrite	DHB evaluation	User evaluation
Continuation 1 of Table 7.5:				
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 the job is triggered again	No	Scan whether the job was terminated without errors. Bit nos. 8 to 11 (error nos.) contain more information on the cause of the errors
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

Bit No.	Set	Delete/ overwrite	DHB evaluation	User evaluation
Continuation 2 of Table 7.5:				
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

7.3.6.2 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 7.6 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.

7.3.6.3 Length Word

The data handling blocks SEND and RECEIVE store in the length word the number of data (in byte) 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. 22, 25 or 26 is set, the current source or destination length is present in the length word If bit no. 23 is set, the length word specifies how much data has been transmitted up to the occurrence of the error

7.3.6.4 Status Word "Parameter Assignment Error (PAFE)"

Bit No.	7 – 4	3 – 1	0
	Cause of error		*

* = Common bit:
- 0: no errors
- 1: parameter error, more details in bits 4 to 7

Table 7.7 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

PAFE value	Cause of error
Continuation of Table 7.7:	
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

7.3.7 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 access of the CP to the CPU.
- SEND-DIRECT
  Data are transmitted to the CP with a specific job number.

Example of a call of the SEND block

STL					CSF/LAD
	:	JU	FB	xxx	FBxxx
NAME	:	SEND			
SSNR	:		KY	0,10	SEND
A-NR	:		KY	0,32	SSNR
ANZW	:		FW	14	A-NR
QTYP	:		KS	DB	ANZW
DBNR	:		KY	0,10	QTYP
QANF	:		KF	+1	DBNR
QLAE	:		KF	+33	QANF
PAFE	:		FY	13	QLAE PAFE
	:				

7.3.7.1 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 CP/DHB driver to the SEND-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. 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 580 can automatically request data from the CPU, e.g. 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/function SEND-ALL. With this DHB you need not note the irrelevant parameters (see above). See Section 9.2.3.2 for the FB numbers of these DHBs.

7.3.7.2 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
- PAFE - Specification of error byte
- QTYP - Source type
- DBNR - Number of data block
- QANF - Initial address of source
- QLAE - Number of source data

The direct mode is generally called 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.

7.3.8 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 access of the CP to the CPU
- RECEIVE-DIRECT
  Data are transmitted to the CPU with a specific block number.

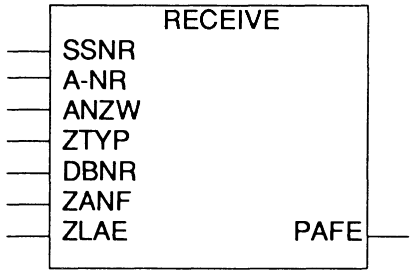
Example of a call of the RECEIVE block

STL

```
      :  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
      :
      :
```

CSF/LAD

FBxxx



7.3.8.1 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 CP/DHB 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 580 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 received 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 call RECEIVE/function RECEIVE-ALL. With this DHB you need not note the irrelevant parameters (see above). See Section 9.2.5.2 for the FB numbers of these DHBs.

7.3.8.2 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
- PAFE - Specification of error byte
- ZTYP - Destination type
- DBNR - Number of data block
- ZANF - Initial address of destination
- ZLAE - Number of destination data

The direct mode is generally called 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.

7.3.9 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.

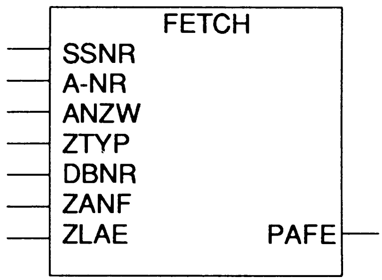
STL	CSF/LAD

STL

```
      :   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
      :
      :
```

CSF/LAD

FBxxx



7.3.9.1 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 acknowledgement. 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.

7.3.10 CONTROL Block

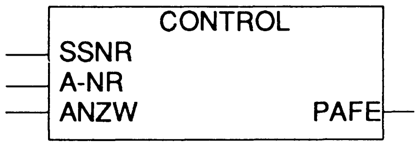
The CONTROL block updates the status word for a particular job or specifies which job is currently being processed.

STL

```
         :   JU        FB     xxx
NAME     :   CONTROL
SSNR     :             KY     0,10
A-NR     :             KY     0,101
ANZW     :             FW     20
PAFE     :             FY     24
         :
```

CSF/LAD

FBxxx



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 CP/DHB 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.

7.3.11 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 CP/DHB driver are deleted.
- RESET-DIRECT
  If the job number is ≠ 0, only the specified job of the interface is deleted.

STL	CSF/LAD
`     : JU FB xxx`	FBxxx
`NAME : RESET`	RESET
`SSNR : KY 0,10`	SSNR
`A-NR : KY 0,101`	A-NR PAFE
`ANZW : FW 20`	
`PAFE : FY 24`	
`     :`	

The block requires the following parameters:

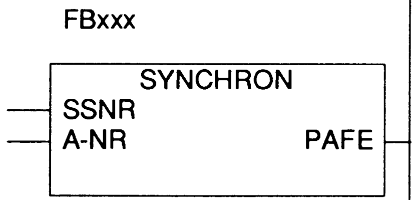
- SSNR - Interface number
- A-NR - Number of job to be deleted
- PAFE - Specification of error byte

The RESET block operates depending on the RLO and can be called from cyclic, time-controlled or alarm-controlled program sections.

7.3.12 SYNCHRON Block

The SYNCHRON block initializes the interface on the CP 580 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

<pre>   : JU   FB   xxx
NAME : SYNCHRON
SSNR :   KY   0,1
BLGR :   KY   0,5
PAFE :   FY   20
      :
      :</pre>
	

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). Synchronization is only achieved if RLO = 0 was transferred to the block.

7.4 Programming the CP 580 User Program

The CP 580 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 CP/DHB driver for this purpose.

This section provides information for programming your CP 580 program:

- Fundamental information on the CP/DHB 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 CP/DHB 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 CP/DHB driver.

7.4.1 CP/DHB Driver

The CP/DHB driver handles the data transfer between S5-CPUs and MS-DOS programs on the CP 580. 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 7.4.6).

7.4.1.1 Installation and Calling

The CP/DHB 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 580 memory. The CP/DHB driver functions are called using a software interrupt. The interrupt number is set to 66H. Fig. 7.2 shows you a simplified division of the CP 580 memory and the calling mechanism of the driver.

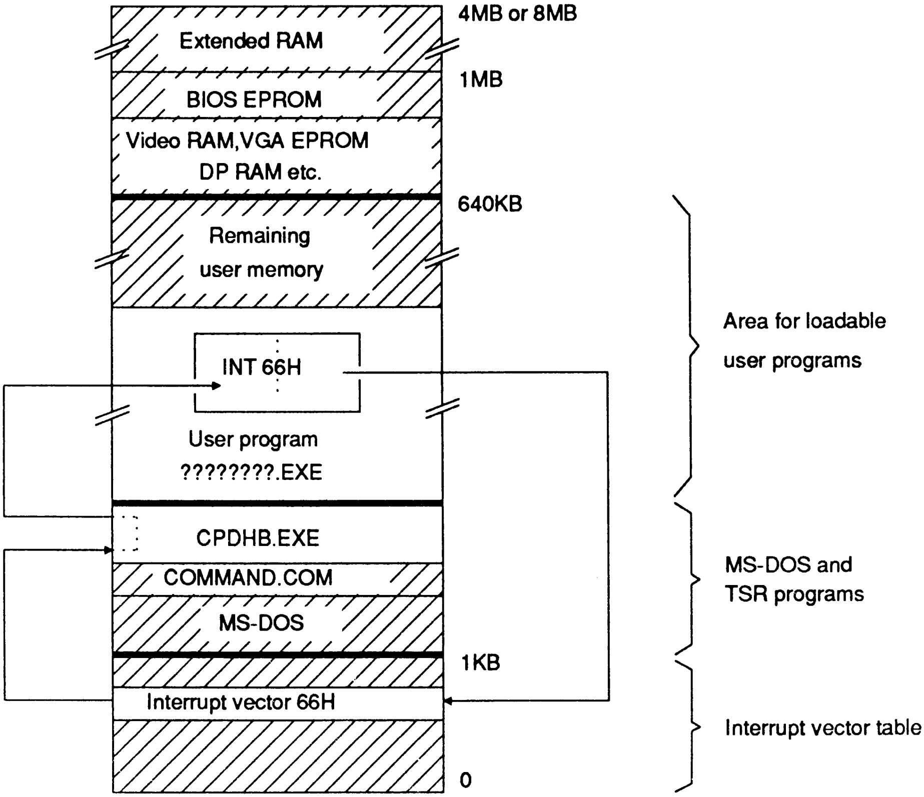


Fig. 7.2 Memory assignments of CP 580 and calling of CP/DHB driver

7.4.1.2 Parameterizing the CP/DHB Driver

If you call the CP/DHB 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. Fig. 7.3 provides an overview.

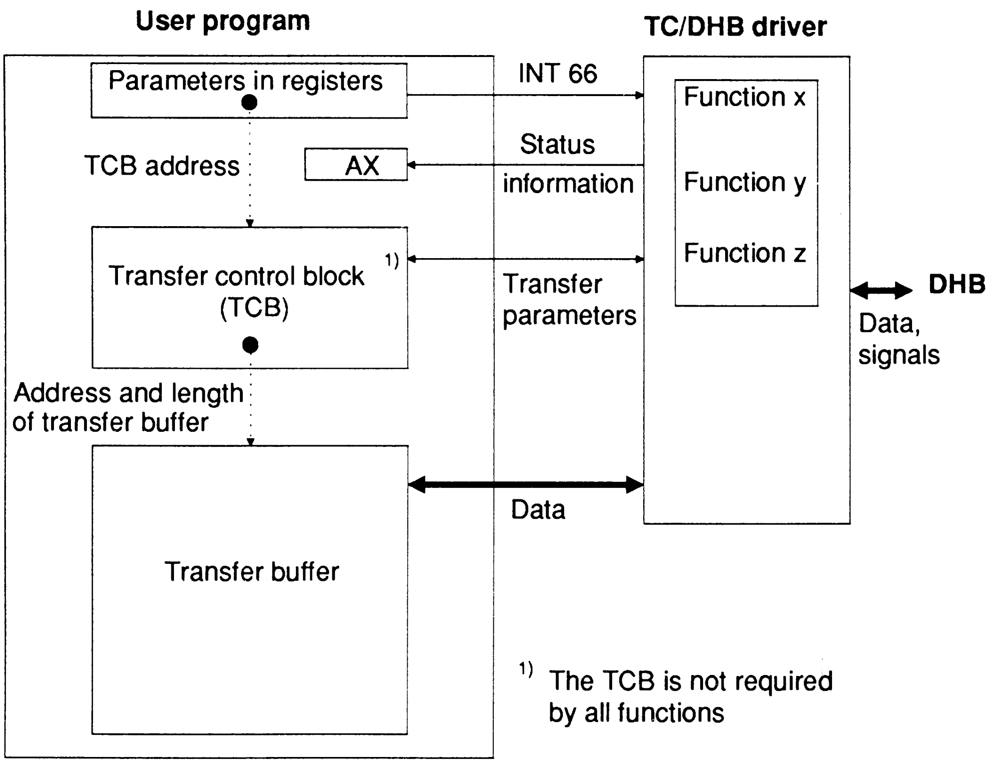


Fig. 7.3 Parameters and resources which must be provided by the user program for driver calls

- **Direct parameters and conditions codes:**

  The application of **direct** parameters to the CP/DHB driver functions and the return of condition codes is via the CPU registers of the CP 580.

- **Indirect parameters and resources:**

  You must provide indirect parameters (e.g. address and length of transfer buffer) and certain resources in a transfer control block (TCB, see Section 7.4.2). An important aid is the transfer buffer which you can provide in your program or somewhere in the CP 580 memory.

Register set:

Fig. 7.4 shows the 8086 register set with the registers provided for parameter transfer:

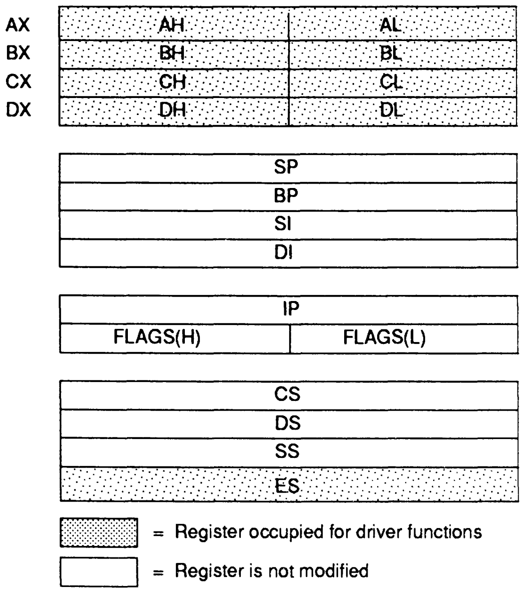


Fig. 7.4 Register set for driver call

Call: Register AX = number of desired function (see Section 7.4.3).

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 CP/DHB driver in register AX when the called program is continued.

The unmentioned registers are not evaluated by the driver and are not modified.

7.4.2 Transfer Control Block (TCB)

If you wish to send a data transfer job to the CP/DHB driver in your program, you must provide a transfer control block (TCB) 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.

Fig. 7.5 shows the structure of the TCB.

Bit No.:	15 ... 8	7 ... 0	
Word 0	DHB job number	DHB type	DHB description
Word 1	0	CPU number	Transfer parameters
Word 2	DB number	Area identification	
Word 3	Area offset		
Word 4	Area length		
Word 5	RW DB number	RW area identification	Extended transfer parameters
Word 6	RW area offset		
Word 7	RW area length		
Word 8	Offset buffer area		Parameters for buffer area
Word 9	Segment buffer area		
Word 10	Length		
Word 11	Write/read index		
Word 12	Current number of transmitted bytes		
Word 13 ... Word 50	38-byte system area		

Fig. 7.5 Structure of transfer control block (TCB)

The following subsections describe the meanings of the individual TCB components.

7.4.2.1 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.

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 7.8 shows you which codes are permissible for the DHB type byte and what the meaning of these codes is:

Table 7.8 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	

7.4.2.2 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 580 program.** The other transmission parameters are either specified by the S5 program depending on the driver function or determined by the CP 580 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 Section 3), the page number is also simultaneously defined by the CPU number in the TCB.

Byte format with CPU number:

Bit No.:	7 … 4	3 … 0
	0	CPU No.

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 7.3).

The area identification is entered by the CP/DHB 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 580 program.

DB number corresponds to the parameter DBNR with the data handling blocks (see Section 7.3). 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 7.9 explains which area identifications are permissible, what meaning they have and how they must be coded for the TCB.

Table 7.9 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
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}.

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 7.9).

Table 7.10 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
FA	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 7.9). You must take into consideration the value of the defined offset for the permissible number range.

7.4.2.3 Extended Transmission Parameters

The extended transmission parameters are only significant with the DHB parameter type RW (see Section 7.3 and description of data handling blocks). The CP/DHB 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

7.4.2.4 Parameters for the Buffer Area

You must provide a sufficiently large memory area in your program or in the RAM of the CP 580 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 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 580: the CP/DHB driver uses it to monitor the area limits. Data outside the defined area are neither read nor written by the driver.

Write/read index:

The CP/DHB modifies a byte-oriented write/read index when 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.

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 2.

7.4.3 Summary of Driver Functions

The CP/DHB 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 580, 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 7.11 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
Receive data used	14H	With
Send data used	15H	With
Scan TCB status	16H	With
Global scan of DHB job reception	17H	With
Read S5 data area	01H	Without
Write S5 data area	02H	Without
Scan driver status	30H	--

7.4.4 Example of Call of CP/DHB 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       DW          ?           ; RW area identification
db_no_rw            DW          ?           ; 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     24   DUP (?)     ; System area 24 bytes
tcb                 ENDS
                     .
                     .
.DATA
                     .
;----------------------------------------------------  Variable:
                     .
stcb                tcb    <>               ; TCB variable
buffer              DW   2000   DUP (?)     ; 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
```

7.4.5 Direct Transfer with Direct Jobs

Section 7.4.5 with its subsections explain the data transfer sequence with direct jobs and parameterization of the CP/DHB 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 CP/DHB 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.

7.4.5.1 Direct Job Sequence

The two following Figs. show you the basic sequence for data transfer with **SEND-DIRECT:**
Fig. 7.6: SEND-DIRECT with **separate** data transfer.
Fig. 7.7: SEND-DIRECT with **direct** data transfer.

In both cases the CP 580 program initially registers a TCB for SEND/FETCH with the CP/DHB 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, Fig. 7.6), or the data are explicitly transmitted with their own driver call (Fig. 7.7).

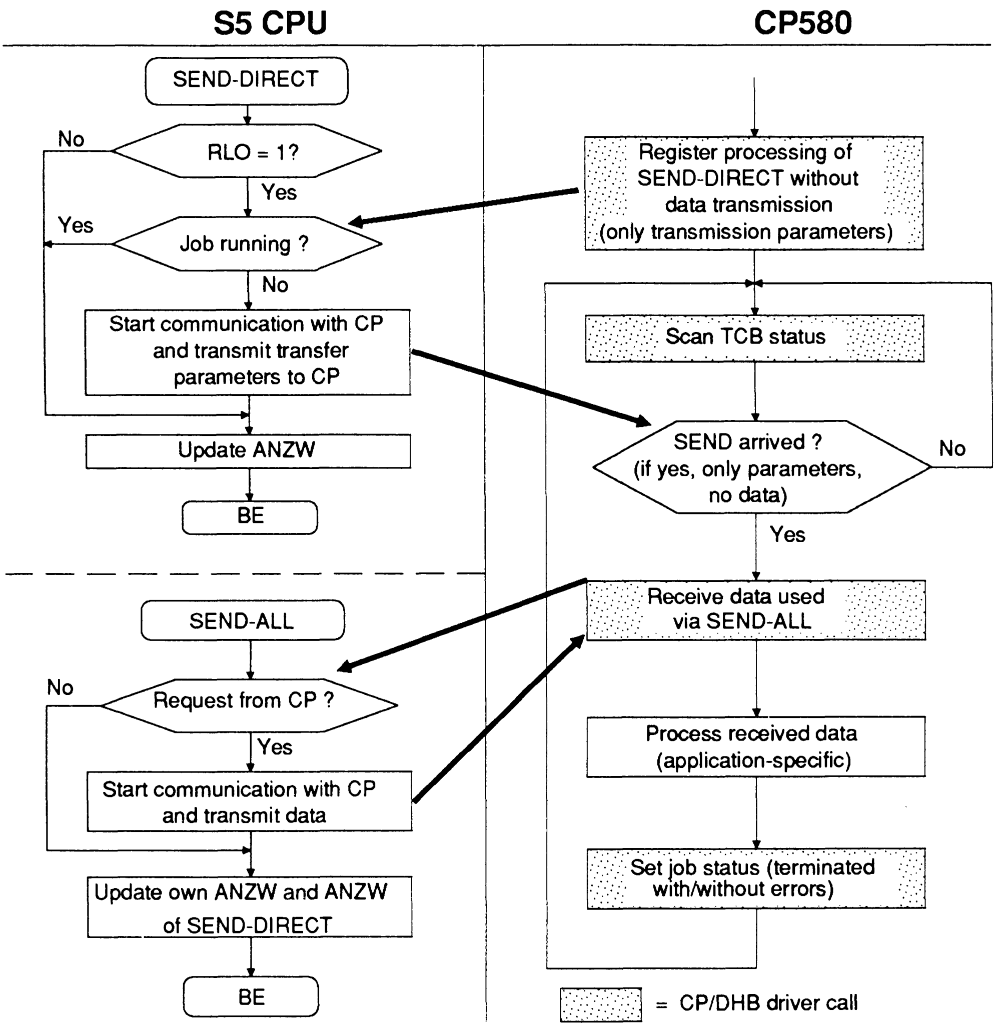


Fig. 7.6 Principle sequence "SEND-DIRECT with separate data transfer"

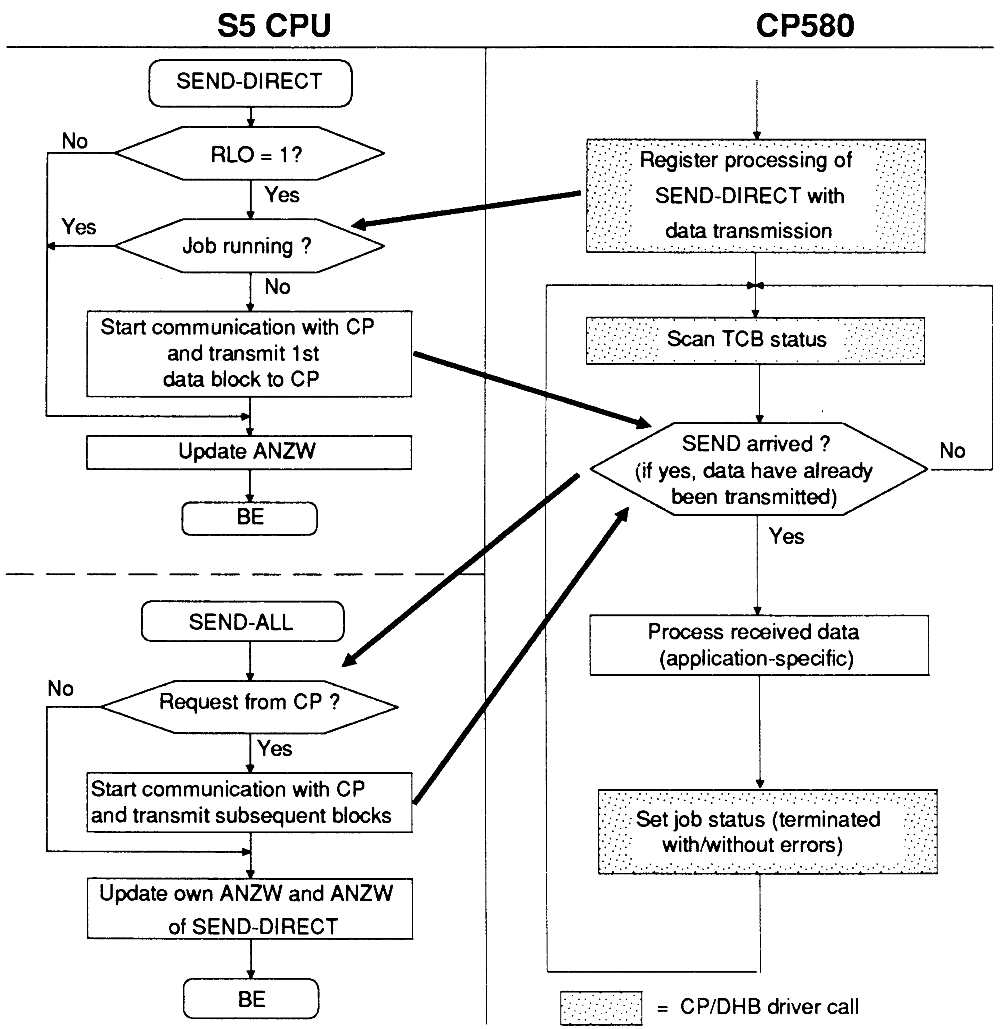


Fig. 7.7 Basic sequence "SEND-DIRECT with direct data transfer"

You can process the useful data in your CP 580 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:

Fig. 7.8 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 580 program initially registers a TCD 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 580 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.

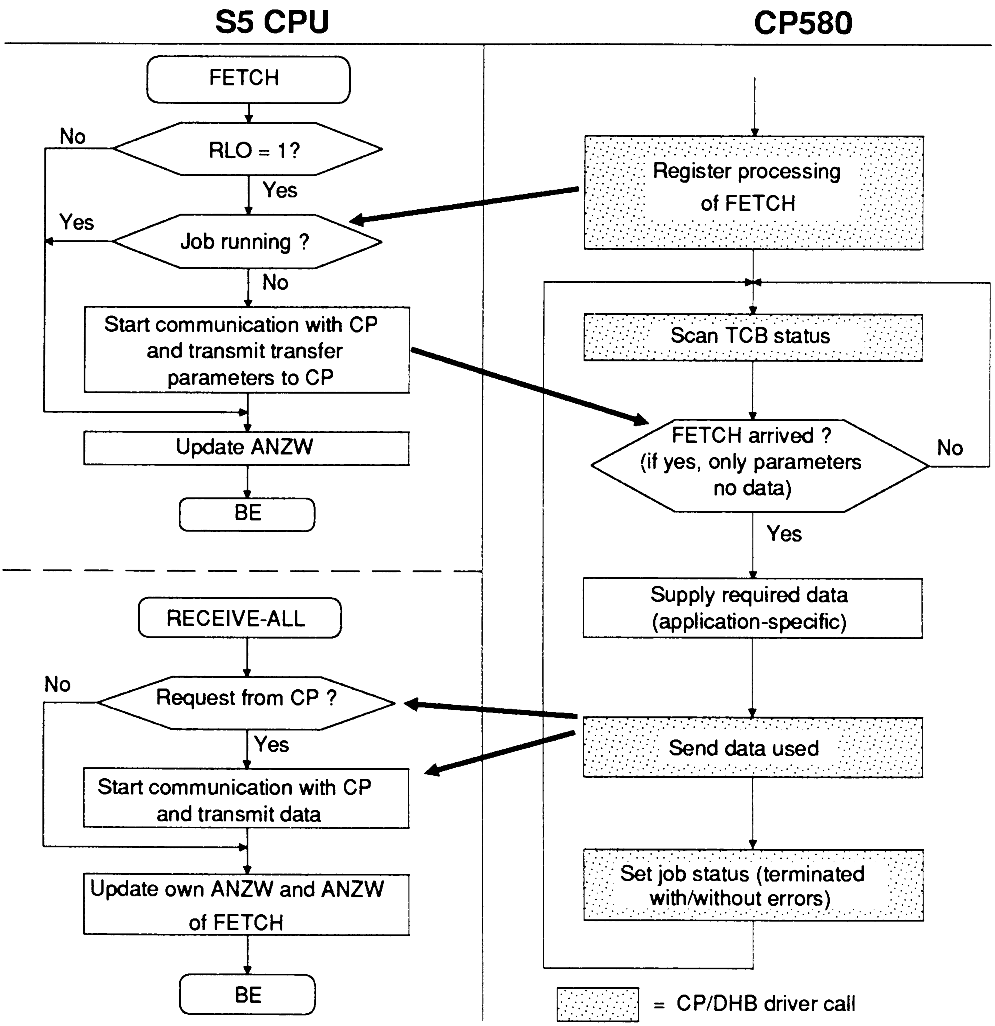


Fig. 7.8 Basic sequence FETCH

RECEIVE-DIRECT job:

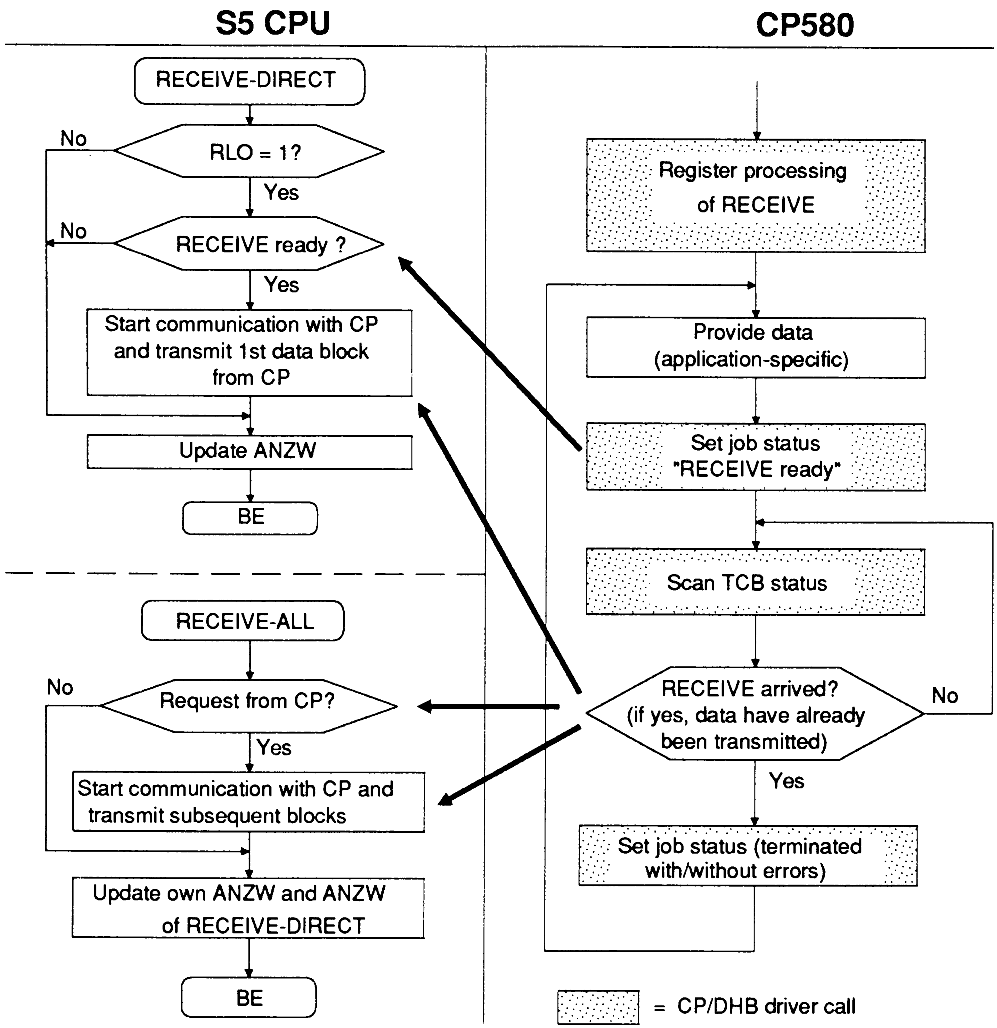


Fig. 7.9 Basic sequence RECEIVE-DIRECT

Fig. 7.9 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 580 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 580 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 580 program then signals using "Set job status" that the job has been finished, and the sequence can begin again.

7.4.5.2 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 580 program is to handle different direct jobs, you must register a transfer control block with the CP/DHB 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 CP/DHB 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 7.12 Assignment of DHB parameters to TCB transmission parameters

DHB parameter	TCB parameter
QTYP/ZTYP	Area identification
DBNR	DB number
QANF/ZANF	Area offset
QLAE/ZLAE	Area length

7.4.5.3 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 7.4.5.2.

Log on job reception:

Application:

This function is used to register a TCB for reception of direct jobs. Your CP 580 program can only wait for **one specific** DHB job with **one** TCB. A job must 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 580 program, you must first check (except wth 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

Register AX:	Negative:	Function was aborted following an error:
		- 5: TCB not logged on
		-51: aborted by negative acknowledgement
		-52: aborted because 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 by 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 580 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 (see Section 7.4.6 "Read S5 area")

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
- 4: TCB being processed

-51: aborted by negative acknowledgement
-52: aborted because 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 by 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 580 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 (see Section 7.4.6: "Read S5 area")

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
		- 4:	TCB being processed
		-51:	aborted by negative acknowledgement
		-52:	aborted because 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

Register AX:	Negative:	(continuation of condition codes)
		-80: aborted by 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 580.

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 7.4.4.4

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
		- 5: TCB is 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 CXY:	CPU number

Condition codes:

Register AX:	Positive:	30:	no job has arrived
		31:	at least one job has arrived, or a SYNCHRON or RESET has taken place
	Negative:	- 3:	CPU number incorrect

The registers which are not listed are not modified by the driver.

Log off job reception:

Application:

Before you terminate your CP 580 program, for example when data transfer has been completed, you must use this function to remove a previously registered job from the CP/DHB 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 has been terminated successfully	
	Negative:	function has been aborted following an error:	
		- 2:	CPU is not synchronized
		- 3:	CPU number is incorrect
		- 4:	TCB is 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 580 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.

7.4.5.4 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:

Bit No.	15 … 12	11 … 8	7 … 4	3 … 0
ANZW		Error No.		Status code

Status codes whose meanings are listed in the following Tables 7.13 and 7.14 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 7.13 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
98H	Job finished with CP error 9	1 0 0 1	1 0 0 0

Status codes for SEND/FETCH			
Code	Meaning	Error No. 1)	Status code
Continuation of Table 7.13:			
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 7.14 Status codes for RECEIVE

Status codes for RECEIVE			
Code	Meaning	Error No. 1) 2)	Status code 2)
06H	Job finished without error	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 1 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

Status codes for RECEIVE			
Code	**Meaning**	**Error No.** 1) 2)	**Status code** 2)
Continuation of Table 7.14:			
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) qqq = original contents are retained

7.4.6 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 CP/DHB driver only requires for these functions one SEND-ALL for transmitting from the CPU to the CP 580 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:

- DHB description: here you only need enter the number with which you wish to transfer data.

- Transmission parameters:
  - 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.

Note

If the size of the area to be transmitted exceeds the transfer frame size set using the DHB SYNCHRON, the CP/DHB driver automatically divides the area into partial frames. The transmission then requires several S5 cycles depending on the size of the area. Your CP 580 program is only continued **once 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 580.
If your CP 580 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: the function is aborted with an error report if the transmission has not been terminated following the timeout.

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
		- 4:	TCB being processed
		-51:	aborted by negative acknowledgement
		-52:	aborted because data area not accessible
		-53:	buffer too small
		-54:	aborted by SYNCHRON
		-55:	aborted by RESET

Negative: (continued)

-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 by 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 580 to a CPU and to write in an S5 data area.
If your CP 580 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 50-ms units: the function is aborted with an error report if the transmission has not been terminated following the timeout.

Condition codes: as with "Read S5 area"

The registers which are not listed are not modified by the driver.

7.4.7 Other Driver Functions

Call driver status:

Application:

You can use this function (e.g. after starting your CP 580 program) to scan whether the CP/DHB driver is correctly synchronized with a CPU.

Call parameters:

Register AX:	Function no. = **30H**
Register CX:	Number of CPU to be synchronized

Condition codes:

Register AX:	Positive:	40:	CPU is synchronized
		41:	CPU is not synchronized
	Negative:	- 3:	CPU number incorrect

The registers which are not listed are not modified by the driver.

Change interrupt for driver call:

If the preset interrupt INT-66H for calling the CP/DHB driver is already used by other applications in your CP 580 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:**

CPDHB-INTxx   where xx = hexadecimal value of new interrupt

The driver stores this value in the memory and can then only be reached via the new interrupt.

7.4.8 MS-DOS Multiplexer Interrupt (INT 2FH) of the CP/DHB 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 **CP/DHB** 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 driver	
Register EX:	Used internally by driver	

Function description:

The INT-2F function 01H checks whether the CP/DHB driver is present in the CP 580 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 7.4.7). The driver enters the release version of the driver software into BX.

Example of call of CP/DHB driver via multiplexer interrupt:

```
    CPDHBID   EQU  0EDH

    MOV  AL,01H
    MOV  AH,CPDHBID
    INT  2FH
    CMP  AL,CPDHBID    ;Driver present?
    JNE  error
       .
       .               ;Driver present!
       .
error:
       .               ;Driver not present!
       .
```

7.5 Testing the Application

7.5.1 Procedure

Proceed in the following steps to determine whether the S5 and CP 580 programs you have produced correctly handled the desired communication between the CPU and CP 580:

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.

2. Always test you 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 580, then only with the driver, and finally together with your CP 580 program (see Section 7.5.2).

4. First test your CP 580 program without the CP/DHB driver, then with the driver but without the CPU and finally together with your S5 program on a CPU (see Section 7.5.3).

5. Check that the data transferred between the CPU and CP 580 agree. Remember to observe the different storage procedures of the various data formats in the CPU and CP 580 memories (see Section 7.5.3).

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.

7.5.2 Testing the S5 Program

Online functions are available to test your S5 program on your programmer (see Reference Literature). 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 7.5.1 (without CP 580, only with CP/DHB driver, then with CP 580 program) should be observed, however.

Useful information for testing is stored in the status words. Section 7.3.6 shows you the meaning of the various bits in the status word.

7.5.3 Testing the CP 580 Program

Before commencing the test of your CP 580 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 (CP/DHB 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 CP/DHB 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 driver status bits in Section 7.5.4).

If you then test the interaction with a CPU, the bits in the CP/DHB driver provide valuable information on which logic errors have occurred.

In a test version of your CP 580 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 580 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.

7.5.4 Representation of the S5 Data in the CP 580 Memory

Data are stored in the CP 580 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 580 are correct.

The following Figs. show you how the various data formats are stored in the CPUs and in the CP 580.

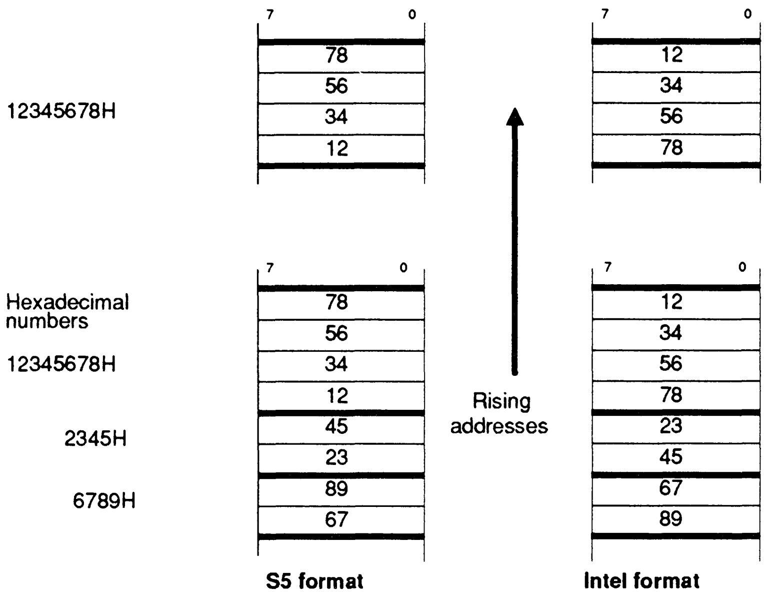


Fig. 7.10 Data representation in S5 and Intel formats

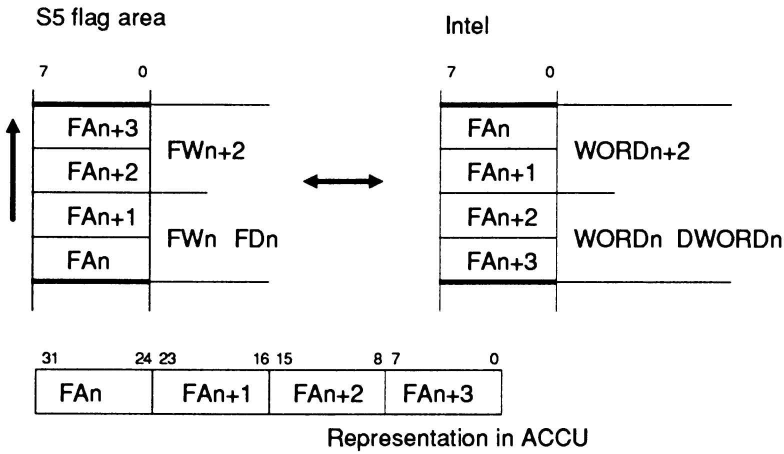


Fig. 7.11 Data representation in S5 and Intel formats

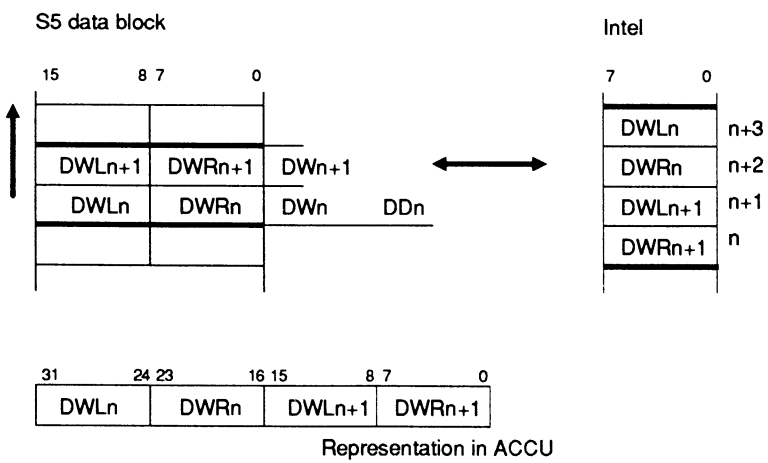


Fig. 7.12 Representation of S5 data block data in Intel format

7.6 Error Bits of the CP/DHB Driver

The CP/DHB 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, e.g. format error in TCP.
- -50 to -79: errors detected by the CP/DHB driver during transmission, e.g. an unexpected response of the CPU.
- -80 to -89: errors signalled by the DHBs during transmission, e.g. parameter with incorrect format (PAFE error 1).

Note

The ranges listed are not used completely.

Table 7.15 Error bits of the CP/DHB 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 CP/DHB 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 CP/DHB 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 CP/DHB driver and cannot be registered. This error message can only occur when registering a TCB.

Error bit in AX		Meaning
Decimal	Hexadec.	
Continuation of Table 7.15:		
-8	FFF8H	TCB unknown: The CP/DHB 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, Table 7.16 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 7.16 Error bits of the CP/DHB driver/error group -50 to -79

Error bit in AX		Meaning
Decimal	Hexadec.	
-51	FFCDH	Aborted - negative acknowledgement: Communication was commenced with the specified TCB but was aborted by the DHB.
-53	FFCCH	Aborted - data area not accessible: The bit "Data area disabled" (bit no. 7) is set in the ANZW.

Error bit in AX		Meaning
Decimal	Hexadec.	
Continuation of Table 7.16:		
-55	FFC9H	Buffer too small: The buffer specified in the TCB is too small for the length parameterized in the DHB.
-56	FFC8H	Aborted by SYNCHRON: The specified TCB was removed from the management of the CP/DHB driver by a SYNCHRON.
-57	FFC7H	Aborted by RESET: The specified TCB was removed from the management of the CP/DHB driver by a RESET.
-58	FFC6H	Unexpected response with FETCH: The FETCH block has not reacted as expected by the CP/DHB driver.
-59	FFC5H	Unexpected response with RECEIVE-DIRECT: The RECEIVE-DIRECT block has not reacted as expected by the CP/DHB driver.
-60	FFC4H	Unexpected response with RECEIVE-ALL: The RECEIVE-ALL block has not reacted as expected by the CP/DHB driver.
-61	FFC3H	Unexpected response with SEND-DIRECT: The SEND-DIRECT block has not reacted as expected by the CP/DHB driver.
-62	FFC2H	Unexpected response with SEND-ALL: The SEND-ALL block has not reacted as expected by the CP/DHB driver.
-63	FFC1H	Aborted by timeout: The DHB SEND-ALL or RECEIVE-ALL does not log on.

The following errors in Table 7.17 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 7.17 Error bits of the CP/DHB driver/error group -80 to -89

Error bit in AX		Meaning
Decimal	**Hexadec.**	
-80	FFB0H	Aborted - parameter error: This error corresponds to "PAFE 0" (is only signalled by certain CPUs).
-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".

Contents

Contents

8 Application Examples

This chapter describes the use of the CP 580 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.
- That you can run self-generated applications using the free programming function.

8.1 Process Data Acquisition

8.1.1 Task/Problem

You wish to monitor the temperature variation of a process using the CP 580 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 580. 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 (e.g. Lotus 1-2-3).

8.1.2 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 DW1 to data word DW5.

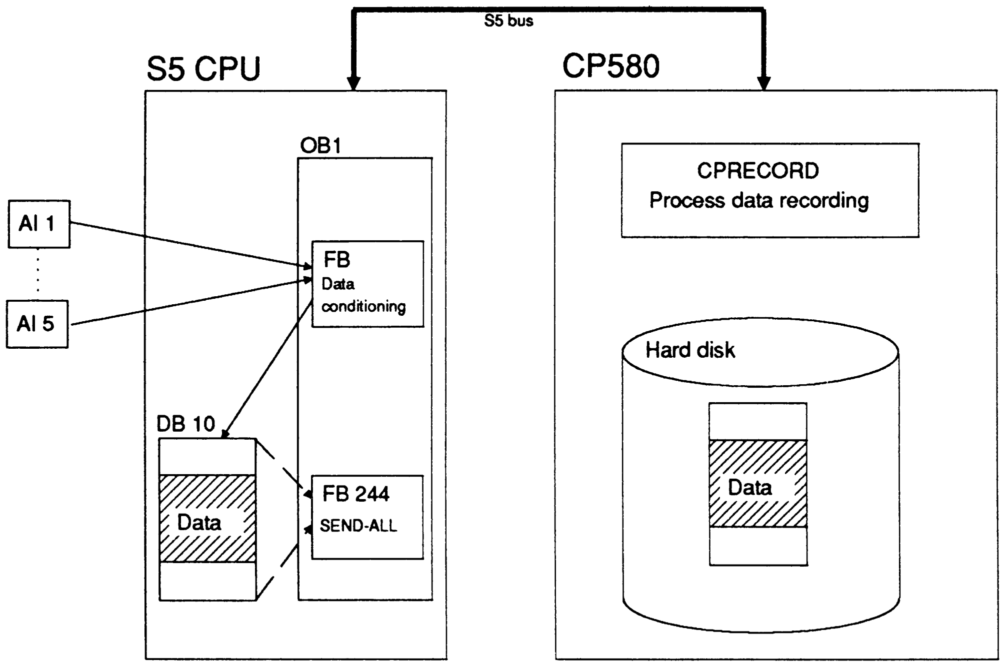


Fig. 8.1 Transmission of measured values to the data block DB 10

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 recording on the CP 580. The data can be called from the hard disk by the evaluation program.

8.1.3 Structure of Solution

The data flow between the S5 CPU and the CP 580 is as follows:

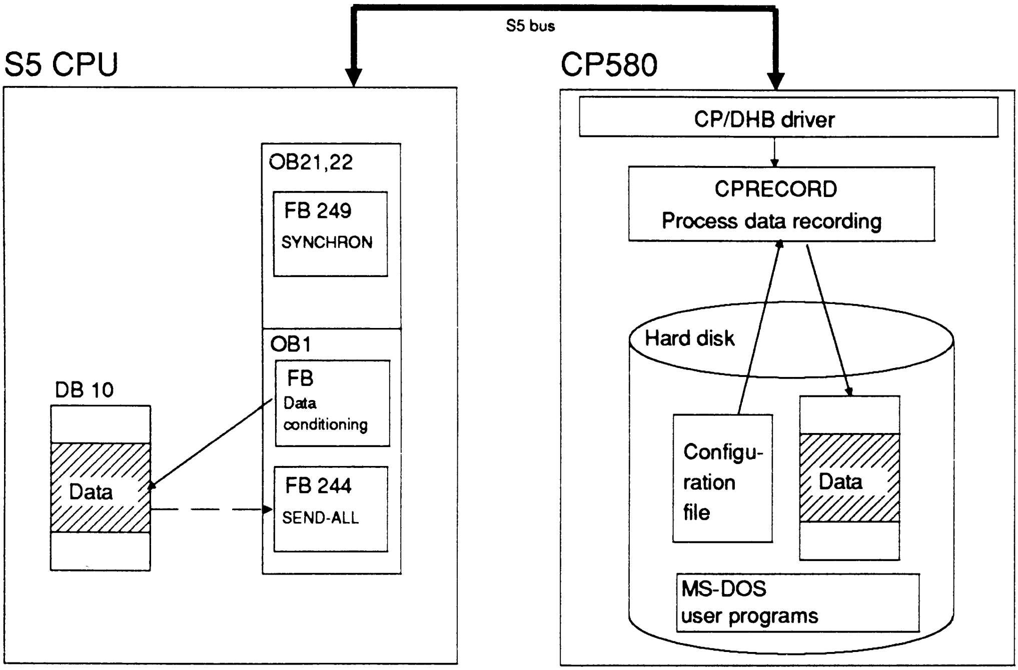


Fig. 8.2 Data transfer between S5 CPU and CP 580 for process data acquisition

- 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 backplane bus and with the assistance of the CP/DHB 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 580 and S5 CPU:

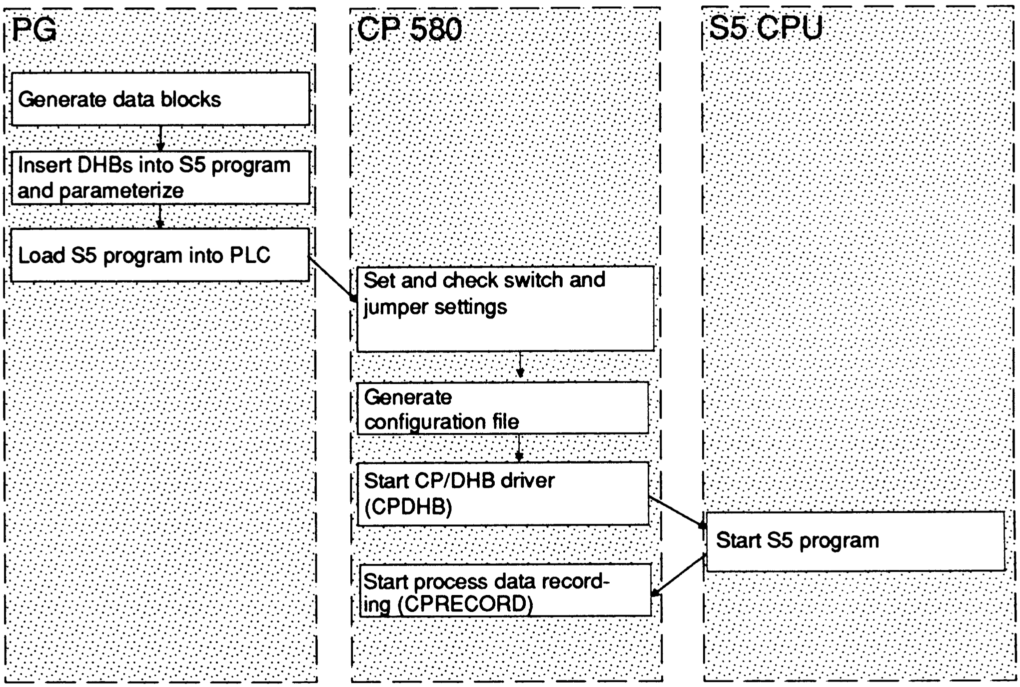


Fig. 8.3 Handling sequence for process data acquisition

8.1.4 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 DW1.

```
DB 10
   0  : KF = +0000;
   1  : KF = +0000;
   2  : KF = +0000;
   3  : KF = +0000;
   4  : KF = +0000;
   5  : KF = +0000;
```

⇒ The data from data block DB 10 are transmitted to the CP via the DHB SEND with the function SEND-ALL following triggering by the CP.

- *Installation and parameterization of data handling blocks in S5 program*

Assign parameter values to the DHB SYNCHRON and insert it into the restart organization blocks OB 21 and 22 (the OB 20 is not present with the S5-115U).

OB 21

```
SEGMENT 1          0000
0000       :                       CP 580 SYNCHRONIZATION
0001       :                       WITH STOP → RUN
0002       :JU     FB 23
0003 NAME  :SYNCP580
0004 REP   :       KF +3           REPETITION FACTOR
0005       :
0006       :BE
```

OB 22

```
SEGMENT 1          0000
0000       :                       CP 580 SYNCHRONIZATION
0001       :                       WITH POWER RETURN
0002       :JU     FB 23
0003 NAME  :SYNCP580
0004 REP   :       KF +3           REPETITION FACTOR
0005       :
0006       :BE
```

```
      FB 23

SEGMENT 1          0000
      NAME  :SYNCP580                  CP 580 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 BYTE, 6=512 BYTE
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 580 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.

```
        OB1

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
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 backplane bus.

- *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 580

- *Setting and checking the switch and jumper settings*

Set interface no. 4 and check the other switch and jumper settings (see Section 2.2.2).

- *Generation of configuration file*

You can modify the CPRECORD.INI (D:\CP580) configuration file included in the scope of delivery using any text editor.

It is better, however, to leave the supplied file unmodified for reference purposes and to generate a new configuration file on the CP, e.g. using the text editor EDLIN. 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 starting process data acquisition.

The configuration file is searched for in the current directory and in the environment path. If it is to be found in another path, you must specify this path and the name of the configuration file when starting process data acquisition.

The configuration file is as follows in our example:

```
;***CPRECORD configuration file***********************************************************
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:\EXAMPLE;  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)
60;          Acquisition cycle in seconds
0;           Data recording is terminated when the
;            defined number of files (1 in this case) is reached
1;           Logging is to be carried out
CP580.LOG;   Destination file for the logged data (is generated in the current
;            directory; if you do not require this, you must specify the the
             complete path)
50;          Timeout in seconds, 1 to 3600 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 only then become effective when you restart the program.

- *Start CP/DHB driver*

When delivered, the CP/DHB driver call (CPDHB.EXE) is present in the AUTOEXEC.BAT file, i.e. the driver is automatically started when the CP 580 is switched on. If you have deleted the driver from the AUTOEXEC.BAT file, you can also start it by entering the command **CPDHB**.

⇒ The CP/DHB driver is loaded into the main memory (TSR program).

The AUTOEXEC.BAT file is as follows when delivered:

```
path c:\;c:\system;c:\cp580
REM ** Message file:
SET CP580=c:\cp580\cp580.msg
KEYB GR,,c:\system\KEYBOARD.SYS
prompt $p$g
mode com1:9600,n,8,1,p
mode lpt1:=com1
REM ** Loading CP/DHB driver:
cpdhb
REM ** Loading CPMASS driver:
cpmass
REM ** Loading command interpreter:
cpshell
```

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 580 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 580

- *Starting process data recording*

Enter the **CPRECORD** command.

You can start the process data acquisition program either by entering the **CPRECORD** command as required or by entering the command into the AUTOEXEC.BAT file. In the latter case, process data acquisition is started automatically when the CP 580 is switched on.

⇒ The program is started and outputs the following text:

```
CPRECORD data recording for CP 580 - version 1.0
Copyright (c) Siemens AG 1991
============================
CP/DHB driver version = xx
CP/DHB interrupt number = 0X66
CPRECORD installed...
'CPRECORD \?' indicates command syntax
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 580.

⇒ 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*

Press the key combination **ALT** and ↵

⇒ 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:

```
Recording active = 1
Timer = 32
Data record = 2
File = 1
Permanent cycle = 0
```

- *Interrupt and continue data recording*

Press the key combination SHIFT (left) and ALT and ↵

⇒ Data recording is stopped. The program is still present in the main memory, however, and data recording is continued when the key combination is entered again.

- *Restarting data recording*

Once the defined number of files has been reached, CPRECORD must first be deinstalled before you can restart data recording.

Enter the **CPRECORD /U** command.

⇒ The CPRECORD program is uninstalled from the main memory.

Enter the **CPRECORD** command.

⇒ The CPRECORD program is started and installed in the main memory.

⇒ The program always searches for the file with the latest date and generates a follow-up file. The advantage of this procedure is that the old data are retained and are not overwritten.

- *Reaction to power failure during data recording*

If the CP/DHB 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**).

⇒ The program searches for the file with the latest 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 CP580.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 (e.g. 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, e.g. 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

24.05.91 14:50

You can generate XY diagrams using the command sequence "Graphic type XY".

For example, the temperature variation at a measurement point in the process (column D of the worksheet) appears as follows over a period of 1 hour:

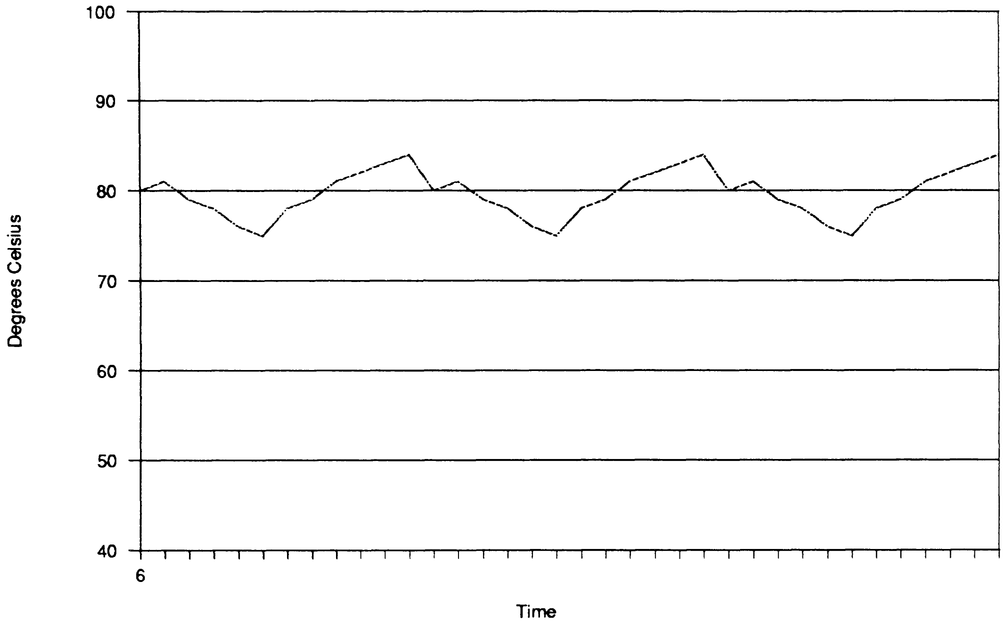


Fig. 8.4 Temperature variation at a measurement point, displayed using Lotus 1-2-3

8.2 Mass Storage Functions

8.2.1 Task/Problem

You wish to transfer large quantities of data from the S5 CPU to the CP 580 communications processor and bring the data back to the S5 CPU as required.

This may be necessary, for example, 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.

8.2.2 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 580
- Transmit data from S5 CPU to CP 580
- Transmit data from CP 580 to S5 CPU
- Delete S5F files on CP 580.

Thus without further programming knowledge on the CP 580 side, and merely using various data handling blocks which you installled 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.

8.2.3 Structure of Solution

The data flow between the S5 CPU and the CP 580 is as follows:

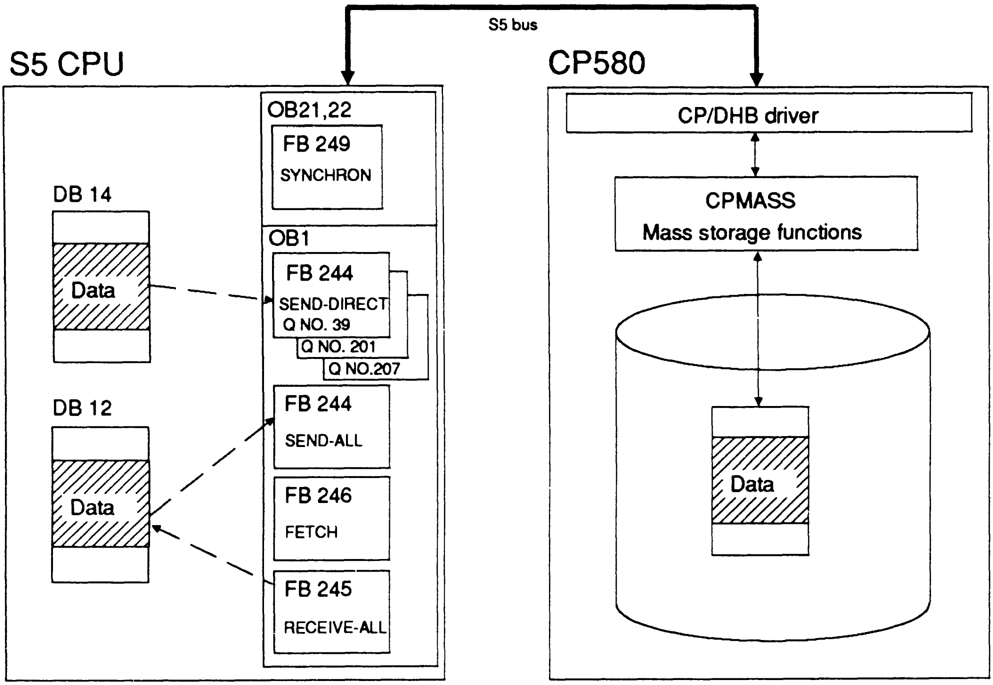


Fig. 8.5 Data transfer between S5 CPUs and CP 580 via S5 bus with mass storage functions

- 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 backplane bus and with the assistance of the CP/DHB 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 backplane bus and with the assistance of the CPMASS program and the CP/DHB 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 580 and S5 CPU:

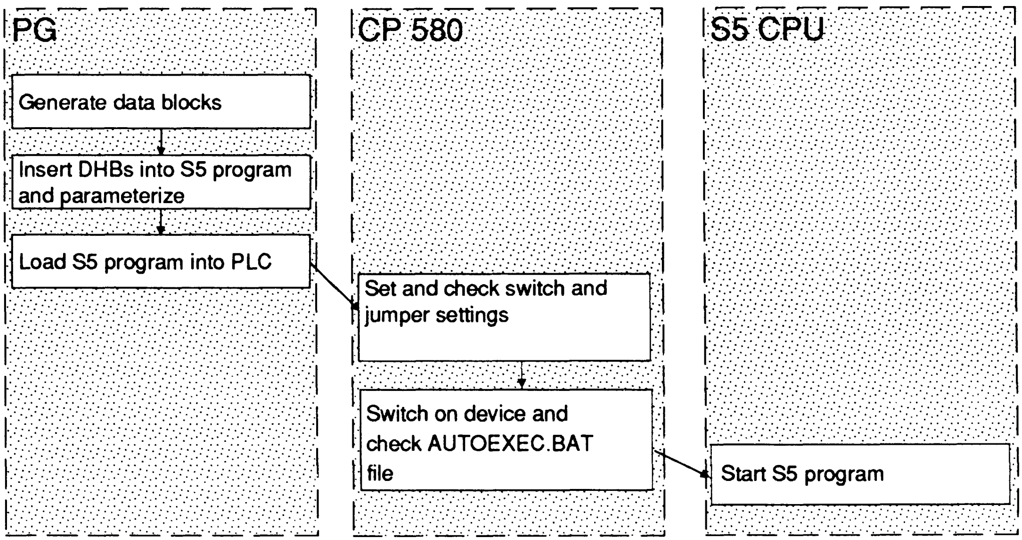


Fig. 8.6 Handling sequence for mass storage functions

8.2.4 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

- *Generation of 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 580.

```
DB 14
   0  : KS = 'C:\CPU1\DIR1';
   6  : KH = 0000H;
   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*

Assign parameter values to the DHB SYNCHRON and insert it into the restart organization blocks OB 21 and 22 (see Section 8.1.4 for listing).

⇒ The S5 CPU and the CP 580 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
0000        :                    *********************************************
0001        :                    EX. OF MASS STORAGE FUNCTION
0002        :                    CP 580/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 backplane 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 backplane 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     M 11.0              FROM DB 12 TO CP 580 FILE
001E         :=      M 11.1
001F         :S      M 11.0
0020         :AN     I 12.0
0021         :R      M 11.0
0022         :
0023         :A      M 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, PER. RANGE 1-99
0028 ANZW    :       FW 112
0029 QTYP    :       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 580 and transfers to it the address of the data to be transmitted to the CP.

⇒ The name of the destination file on the CP 580 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       M 11.2           FROM CP AND STORE IN DB 12
0031          :=        M 11.3
0032          :S        M 11.2
0033          :AN       I 12.1
0034          :R        M 11.2
0035          :
0036          :A        M 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 580 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      M 11.4           DIRECTORY ON CP 580
0045        :=       M 11.5
0046        :S       M 11.4
0047        :AN      I 12.2
0048        :R       M 11.4
0049        :
004A        :A       M 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 DIREC.
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     M 11.6          S5F FILES ON CP 580
0058          :=      M 11.7
0059          :S      M 11.6
005A          :AN     I 12.3
005B          :R      M 11.6
005C          :
005D          :A      M 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 FILES
0062 ANZW     :       FW 130
0063 QTYP     :       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 580

- *Setting and checking the switch and jumper settings*

Set interface no. 4 and check the other switch and jumper settings (see Section 2.2.2).

- *Switch on the device and check the AUTOEXEC.BAT file*

When delivered, the CP/DHB driver and the CPMASS program are present in the AUTOEXEC.BAT file (CP/DHB driver, CPMASS.EXE), i.e. they are automatically started when the CP 580 is switched on. The directories C:\CPU1, C:\CPU2, C:\CPU3, and C:\CPU4 are generated in addition.

Check whether the two programs are still present in the AUTOEXEC.BAT file. If this is not the case, enter them again.

⇒ The CP/DHB driver and the CPMASS program are loaded into the main memory and started.

⇒ The following message is output on the screen:

```
CPMASS Vxx
CPDHB Vxx
```

⇒ 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

- *Starting the 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 580 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.

8.3 Command Interpreter

8.3.1 Task/Problem

You wish to output S5 CPU messages on a message printer via the command interpreter of the CP 580 communications processor which is installed in the subrack of your S5-115U.

8.3.2 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.

The connected printer can be e.g. a PT88. Section 2.2.4 provides you with information on the printer settings required and on the connection cable.

The printer output must be routed to the COM1 interface.

8.3.3 Structure of Solution

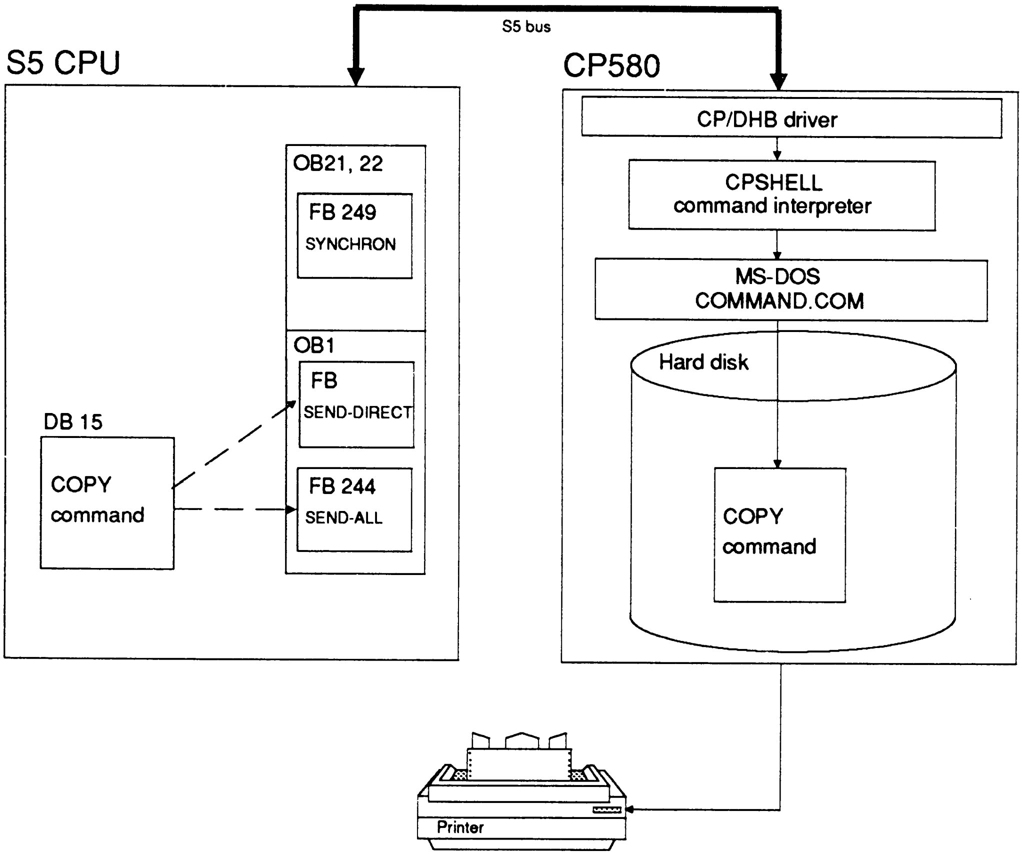


Fig. 8.7 Data transfer between S5 CPU and CP 580 when using the command interpreter

- 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.

 C79000-B8576-C204-01

- 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 580 and S5 CPU::

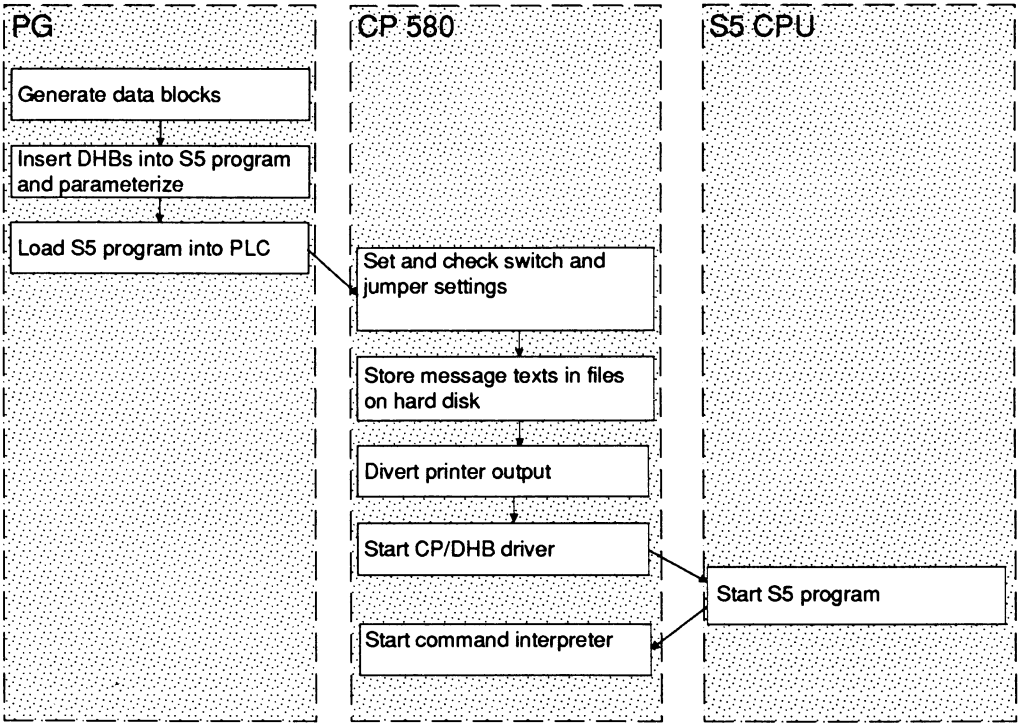


Fig. 8.8 Handling sequence when using the command interpreter

8.3.4 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

- *Generation of data blocks*

Generate the data block DB 15 and enter a COPY command for every message.

```
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 8.1.4 for listing).

⇒ The S5 CPU and the CP 580 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 1
0011        :AN     M 150.0
0012        :=      M 150.1
0013        :S      M 150.0
0014        :AN     I 13.0
0015        :R      M 150.0
0016        :
0017        :A      M 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           PARAMETER ERROR
            :
            :
            :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
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 580

- *Setting and checking the switch and jumper settings*

Set interface no. 4 and check the other switch and jumper settings (see Section 2.2.2).

- *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 580.

⇒ 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.

- *Diversion of printer output*

Enter the following information into the AUTOEXEC.BAT file (if not already included):

```
MODE COM1:9600,n,8,1,P
MODE LPT1:=COM1
```

⇒ The printer output is diverted to the COM1 interface.

- *Starting the CP/DHB driver*

Start the CP/DHB driver by switching on the CP 580.

When delivered, the CP/DHB driver (CPDHB.EXE) is present in the AUTOEXEC.BAT file. If you have deleted the driver from this file, you can also start it by entering the **CPDHB** command.

⇒ The CP/DHB 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 580 cannot be synchronized.

⇒ The command interpreter can be started if the synchronization is successful.

Operations on the CP 580

- *Starting the command interpreter*

Enter the **CPSHELL** command.

⇒ The command interpreter is started. The following start message is output on the screen:

```
CP 580 S5 command interpreter Vx.x
Copyright (c) Siemens AG 1991
CPSHELL ready
```

⇒ Any S5 messages are output on the printer.

- *Terminating the command interpreter*

Press the key **ESC**

⇒ The CPSHELL program is terminated.

8.4 Free Programming

The CP 580 communications processor provides you with the possibility for executing self-generated application programs in the PLC.

The S5 bus driver interface of the CP 580 is available for free use. You can call the CP/DHB driver from your application program using an adjustable software interrupt (INT66H).

You need not use a specific programming language. The language must only provide the facility for transferring software interrupts and accessing CPU registers.

An example of free programming is the CPSHELL command interpreter application program. The listing of the C program for the command interpreter can be found in the directory C:\CP580\SRC on the hard disk of the CP 580.

Contents

Contents

9 Reference Section for System Software

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 deals with storage of the device configuration in the software (SETUP). This is followed by a summary of the data handling blocks and their parameterization as used with the CP 580 for the various functions. The error bits and error messages are also listed for the process data acquisition, mass storage and command interpreter functions.

9.1 SETUP (Setting of Device Configuration in the Software)

9.1.1 Restart with Preset Device Configuration for the CP 580 Basic Version

The CP 580 in the basic version starts up without a manual SETUP.
Please refer to Section 2.3.2.

Diskette A:	3,5 ", 1,44 MB	
Diskette B:	Not Installed	
Hard Disk 1:	Type 17	
Hard Disk 2:	Not Installed	
Base Memory:	640 KB	
Extended Memory:	XXXX KB*	
Video Card:	VGA/EGA	
Keyboard:	Not Installed	
CPU Speed:	Fast	Numlock on at boot:No

Fig. 9.1 Default SETUP for the basic configuration

*	1024 KB	for	2-Mbyte	DRAM
	3072 KB	for	4-Mbyte	DRAM
	7168 KB	for	8-Mbyte	DRAM

9.1.2 CP 580 Restart with Modified Device Configuration

The device configuration of your CP 580 is preset. You may only carry out modifications using the SETUP program if:

- the CC/EU was without an operating voltage for longer than approx. 15 minutes and the central back-up battery of the CC/EU is flat
- a fault occurs when switching on, or
- you wish to adjust the hardware clock of the CP 580.

The SETUP program is used to inform the system of device components such as memories, drives or graphic interfaces. It is also used to adjust the time of the clock component.

Note

If you wish to select the SETUP program during normal operation, e.g. in order to set the date and time, press the following key combination:

<CTRL> <ALT> <S or +>

Certain operating systems and user programs reject the above key sequence.
Press the RESET key of the CP 580 and wait for the memory test. Abort this using the space bar. Now you can start the SETUP program before loading the operating system by simultaneously pressing the three keys

<CTRL> <ALT> <S or +>

Execute the SETUP as follows:

1. When you switch on the programmable controller, the CP 580 tries to run up. If the SETUP data do not agree with the default values, the following message appears on the screen:

   "Invalid configuration - run SETUP program Press the F1 key to continue, F2 to run the SETUP utility"

   Now press function key F2 in order to start the SETUP program.
2. A form now appears on the screen in which the SETUP data to be set are listed.

   Press any key to continue the SETUP program.
3. A form appears on the screen on which you can carry out the SETUP settings. Fig. 9.2 shows you the basic settings when the form appears.

Phoenix SETUP Utility (Version 1.00) 03
(c) Phoenix Technologies Ltd 1985, 1991 All Rights Reserved

Page 1 of 2

**** Standard System Parameters ****

System Time	:	00:00:00							
System Date	:	Jan : 01, 1991							
Diskette A	:	3.5", 1.44 MB							
Diskette B	:	Not Installed	Cyll	HD	Pre	LZ	Sec	Size	
Hard Disk 1	:	Type 17	977	5	300	977	17	40	
Hard Disk 2	:	Not Installed							
Base Memory	:	640 KB							
Extended Memory	:	XXXX KB*							
Video Card	:	VGA/EGA							
Keyboard	:	Not Installed							
CPU Speed	:	Fast	Num Lock on at boot: NO						

ESC	F1	F2		↑↓	+/-	Pg Up/Dn
Menu	Help	Sys Info		Field	Value	Page

Fig. 9.2 SETUP screen form

* 1024 KB for 2-Mbyte DRAM
3072 KB for 4-Mbyte DRAM
7168 KB for 8-Mbyte DRAM

You can move the cursor using the arrow keys.
You can make settings using the +/- keys or the numeric keys.
Use the Pg Up/Dn key to move one page forwards (or backwards) to access "Page 2 of 2" for setting the shadow RAM.
You need not make any settings here, however.

4. Set the **current time** and the **current date** in the SETUP form using the numeric keys.
5. Once you have made the settings, press the ESC key.

Fig. 9.3 shows the form which now appears on the screen.

Phoenix SETUP Utility (Version 1.00) 03
(c) Phoenix Technologies Ltd 1985, 1991 All Rights Reserved

Page 1 of 2

**** Standard System Pa**

System Time	:	**00:00:00**
System Date	:	**Jan : 01, 1991**
Diskette A	:	**3.5", 1.44 MB**
Diskette B	:	**Not Installed**
Hard Disk 1	:	**Type 17**
Hard Disk 2	:	**Not Installed**
Base Memory	:	**640 KB**
Extended Memory	:	**XXXX KB**
Video Card	:	**VGA/EGA**
Keyboard	:	**Not Installed**
CPU Speed	:	**Fast**

**** Exiting SETUP ****

ESC	**Continue with SETUP**
F4	**Save values, exit**
F5	**Load default values for all pages**
F6	**Abort SETUP without**

ESC		**F4**	**F5**	**F6**
Continue		**Save**	**Defaults**	**Abort**

Fig. 9.3 Enter SETUP data

6. Press function key F4 to enter the new SETUP data and to reboot.
   The CP 580 now runs up to the operating system level. "C:\>", the prompt of the MS-DOS operating system, appears on the monitor.

9.1.3 Setting the Date and Time

You are prompted by a form in the SETUP program in order to set the date and time. Please refer to Section 9.1.2.

9.2 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 580.
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 (see Reference Literature):

9.2.1 Summary of DHBs with the CP 580

The following table provides a summary of the DHBs used with the CP 580.

Table 9.1 Summary of DHBs

DHB	Function	S5-115U	S5-135U	S5-155U
SYNCHRON	Synchronizes CP 580 and CPU	FB 249	FB 125	FB 125
SEND-DIRECT	Activates the CP 580 and transfers the address of the required data with CPMASS	FB 244	FB 120	FB 120
SEND-ALL	Transfers the data from the CPU to the CP 580 with CPRECORD and with CPMASS	FB 244	FB 120	FB 120
SEND-A [1)]	Special SEND-ALL FB for S5-135U and S5-155U	-	FB 126	FB 126
FETCH	Activates the CP 580 and transfers the address of the required data with CPMASS	FB 246	FB 122	FB 122

DHB	Function	S5-115U	S5-135U	S5-155U
Continuation of Table 9.1:				
RECEIVE-DIRECT	Data are transmitted from the CP 580 to the CPU with a specific job number (free programming)	FB 245	FB 121	FB 121
RECEIVE-ALL	Transfers the data from the CP 580 to the CPU with CP RECORD and CPMASS	FB 245	FB 121	FB 121
REC-A [2)]	Special RECEIVE-ALL FB for S5-135U and S5-155U	-	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 CP/DHB driver	FB 248	FB 124	FB 124
RESET-DIRECT	Resets the CP/DHB 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 (e.g. 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 present in the operating system of the CPU. In the S5-135U programmable controller (CPU 922, CPU 928 and CPU 928B), the DHBs are present in the operating system of the CPU. The block headers are present on floppy disks and must be loaded into the CPU.
In the S5-155U programmable controller, the DHBs are present as a STEP 5 program on floppy disks and must be loaded into the CPU.

9.2.2 DHB SYNCHRON

The DHB SYNCHRON synchronizes the interface between the CPU and CP 580 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 580.

Table 9.2 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 error bits

9.2.3 DHB SEND

The DHB SEND has two modes

- SEND-DIRECT (job number ≠ 0)
- SEND-ALL (job number = 0)

9.2.3.1 DHB SEND-DIRECT

You require the DHB SEND-DIRECT if you wish to send data from the CPU to the CP 580 e.g. with the mass storage function. By calling the DHB SEND-DIRECT you activate the CP 580 and transfer to it the address of the data to be sent by the CPU.

Table 9.3 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 no.
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 error bits

9.2.3.2 DHB SEND-ALL

You require the DHB SEND-ALL in order to carry out the data transmission to the CP 580 e.g. with the mass storage function (CPMASS). With the CPRECORD program, you must call the DHB SEND-ALL in each cycle of the CPU.

Table 9.4 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 no.
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 error bits

9.2.4 DHB FETCH

You require the DHB FETCH if you wish to send data from the CP 580 to the CPU e.g. with the mass storage function (CPMASS). By calling the DHB FETCH you activate the CP 580 and transfer to it the address of the data to be sent to the CPU.

Table 9.5 Format and meaning of parameters of DHB FETCH

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no.
ANZW	I	W	Status word
ZTYP	D	KS	Data type of destination
DBNR	D	KY	Number of data block if destination is data block
ZANF	D	KF	Offset of first item of data to be written in the destination
ZLAE	D	KF	Number of data units to be written (words or bytes)
PAFE	Q	BY	Parameter error bits

9.2.5 DHB RECEIVE

The DHB RECEIVE has two modes

- RECEIVE-DIRECT (job number ≠ 0)
- RECEIVE-ALL (job number = 0)

9.2.5.1 DHB RECEIVE-DIRECT

You require the DHB RECEIVE with the function RECEIVE-DIRECT to send data from the CP 580 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 9.6 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 no.
ANZW	I	W	Status word
ZTYP	D	KS	Data type of destination
DBNR	D	KY	Number of data block if destination is data block
ZANF	D	KF	Starting address of destination
ZLAE	D	KF	Number of data units to be read (words or bytes)
PAFE	Q	BY	Parameter error bits

9.2.5.2 DHB RECEIVE-ALL

You require the DHB RECEIVE with the function RECEIVE-ALL in order to carry out the data transmission from the CP 580 to the CPU e.g. with the mass storage function.

Table 9.7 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 no.
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 error bits

9.2.6 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)
- CONTROL-DIRECT (job number ≠ 0)

CONTROL-ALL:
The CONTROL-ALL function indicates in the Low byte of the ANZW which job is currently being processed by the CP 580.

CONTROL-DIRECT:
A so-called job status exists in the interface for each job. This is managed by the interface and indicates e.g. 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 9.8 Format and meaning of parameters of DHB CONTROL

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no. of job to be monitored
ANZW	I	W	Status word: contains result of scan
PAFE	Q	BY	Parameter error bits

9.2.7 DHB RESET

The DHB RESET has two modes

- RESET-ALL (job number = 0)
- RESET-DIRECT (job number ≠ 0)

The RESET-ALL function (job number = 0) resets all jobs of the CP 580 interface.

With the RESET-DIRECT function (job number ≠ 0), only the specified job of the CP 580 interface is reset.

Table 9.9 Format and meaning of parameters of DHB RESET

Parameter	Kind	Type	Meaning
SSNR	D	KY	Interface number
A-NR	D	KY	Job no. of job to be reset
PAFE	Q	BY	Parameter error bits

9.2.8 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 doubleword which is addressed by the parameter ANZW. The second part of the doubleword is the so-called length word.

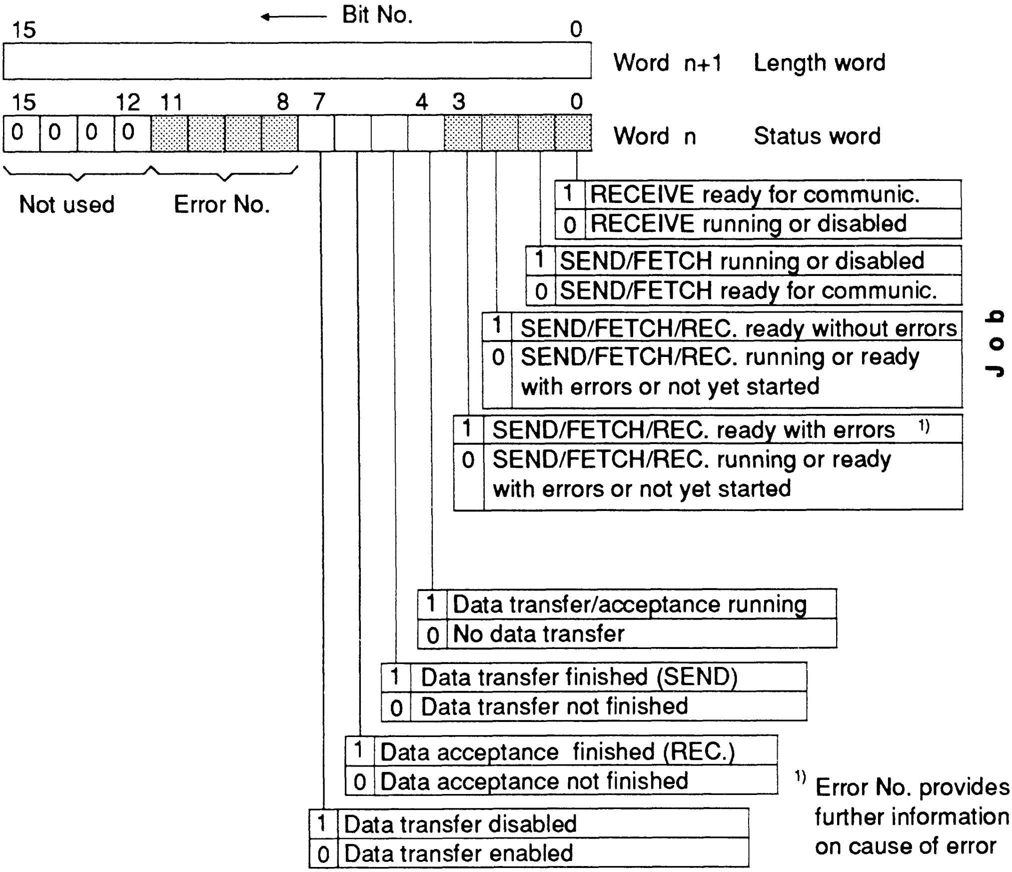


Fig. 9.4 Format of DHB status words

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 CP/DHB driver initializes the status word with 0006H. This value is retained until an application (e.g. CPMASS) logs on for a specific job number with the CP/DHB 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 580 for the respective job number as long as 0006H is present in the status word. Observe this response when restarting the S5 program.

9.2.9 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 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 9.10 shows a summary of the parameter error bits.

Table 9.10 Parameter error bits

PAFE value	Cause of error
00H	No error
11H	Source/destination parameter has incorrect format
21H	DB or DX data block does not exist or is 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)

PAFE value	Cause of error
Continuation of Table 9.10	
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

9.3 Process Data Acquisition

9.3.1 Parameters for the Configuration File (Process Data Acquisition)

Refer to Section 4.3.2 for how to edit the configuration file supplied on the CP 580 according to the desired data transfer.
Table 9.11 shows you the meaning of the various parameters and the permissible values.

Table 9.11 Parameters in the configuration file

Line No.	Parameter	Meaning	Permis-sible values
1	CPU No.	Number of CPU corresponding to slot sequence	1 to 4
2	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
3	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
4	Offset (QANF)	Number of 1st data unit to be read in the S5 area	Dependent on data type and CPU

Line No.	Parameter	Meaning	Permis-sible values
Continuation 1 of Table 9.11:			
5	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
6	Destination path	Path name for ASCII files	MS-DOS syntax
7	Extension	File name extension	ASCII characters
8	Number of files	Maximum number of ASCII files for process data recording	1 to 10 000
9	Number of data records	Maximum number of data records per ASCII file	1 to $(2^{31} - 1)$
10	Format	"Individual conversion: Path and file name for format file or (only with DB/DX) for file with preheader data	MS-DOS syntax, file name: "??????ST .S5D" or "*.FMT"; ? = letter/ number or @, * = max. 8 letters/ numbers

Line No.	Parameter	Meaning	Permis-sible values
Continuation 2 of Table 9.11:			
10 (contd.)	Format (contd.)	"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)	 KS, KF, KG, KH, KM, KT, KC, KY, KB
11	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
12	Acquisition cycle	Cycle time in seconds in which the process data are to be read by the CPU	1 to 11 799 360
13	Acquisition mode	0: recording is terminated once the defined file number has been reached 1: "permanent" recording; the oldest file is deleted when the defined file number has been reached and is then overwritten (similar to cyclic 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

Line No.	Parameter	Meaning	Permis-sible values
Continuation 3 of Table 9.11:			
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

9.3.2 Error Messages of the CP/DHB 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 4.3.2.4.

Messages are sent by the CP/DHB driver and the CPRECORD program. The two following tables show which messages can occur and what they mean.

9.3.2.1 Error Messages of the CP/DHB Driver

Table 9.12 Messages of the CP/DHB driver

Message		Remarks
[CPDHB]	CPU not synchronized	
[CPDHB]	Aborted by timeout	
[CPDHB]	Parameter invalid (PAFE 1)	
[CPDHB]	DB/DX does not exist (PAFE 2)	
[CPDHB]	Area too small (PAFE 3)	
[CPDHB]	Area does not exist (PAFE 4)	
[CPDHB]	Status word error (PAFE 5)	

9.3.2.2 Error Messages of the CPRECORD Program

Table 9.13 Messages of the CPRECORD program

Message	Remarks
Data recording started <Date> <Time>	Message at beginning of process data acquisition
Data recording finished <Date> <Time>	Message at end of process data acquisition
Error opening output file	MS-DOS output
Format error in configuration file with parameter <n>	<n> = line number of faulty parameter
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
Invalid output drive	MS-DOS output
Insufficient drive capacity (<Actual value> <Required value>)	MS-DOS output
CPRECORD file cannot be removed from memory	MS-DOS output
Unknown option	Incorrect parameter in CPRECORD command
S5D file: could not be opened	MS-DOS output
S5D file: read error	MS-DOS output

Message	Remarks
Continuation 1 of Table 9.13:	
S5D file: not a root directory	Format error
S5D file: too many subdirectories (>128)	
S5D file: no DV subdirectories	
S5D file: positioning error	MS-DOS output "Seek error"
S5D file: too many data elements (>128)	
S5D file: the configured DV block is missing	
S5D file: wrong block ID in DV preheader <Code>	Format error, <Code> = incorrect block code
S5D file: wrong block no. in DV preheader <Number>	Format error, <Number> = incorrect block number
S5D file: DV preheader address too large (>4095)	Format error
S5D file: format error, first format = empty	
Output drive full	MS-DOS output
Error when writing output file	MS-DOS output
Output directory not found	MS-DOS output
Configuration file: unknown S5 area	
Insufficient number of available DOS clusters (<Actual value> <Required value>) [1)]	
CPRECORD.INI not found in current directory	

Message	Remarks
Continuation 2 of Table 9.13:	
Configuration file extension is not ".INI": <Config file>	<Config file> = name of configuration file
CP/DHB driver not loaded	
FMT file: could not be opened <FMT file>	MS-DOS output <FMT file> = name of format file
FMT file: unknown format in line <n> (<Format>)	<n> = line No., <Format> = incorrect format
FMT file: format error in line <n> (<Format>)	<n> = line No., <Format> = incorrect format
FMT file: too long in line <n> onwards (<Format>)	<n> = line No., <Format> = incorrect format
S5D file format is permissible only with S5 area DB or DX	
Error when opening log file <Log file>	<Log file> = name of logbook file

1) Cluster = associated logic memory area on drive.

9.3.3 Hotkeys for CPRECORD

Hotkeys are key combinations with which a program executed in the background can be accessed.
The background program constantly monitors the keyboard inputs and carries out the desired function when the corresponding key combination is pressed.

Note

The initiation of functions by hotkeys may fail if you have started a program in the foreground which frequently accesses the system programs.

Table 9.14 shows you the hotkey functions.

Table 9.14 Hotkey functions for CPRECORD

Command	Function
"CPRECORD/?"	Information on CPRECORD;
"CPRECORD path name"	Replace/modify preset configuration file
"CPRECORD/U"	Remove CPRECORD from CP 580 memory
Press keys SHIFT (right key) ALT and ↵	CPRECORD outputs the parameters from the configuration file on the screen starting at the current cursor position
Press keys ALT and ↵	CPRECORD outputs the status variables of process data acquisition on the screen starting at the current cursor position
Press keys SHIFT (left key) ALT and ↵	Interrupt or continue process data acquisition

9.4 Mass Storage Functions

9.4.1 Data Handling Blocks for the Mass Storage Functions

Use the DHBs SEND-DIRECT and FETCH to trigger data transfer between the CP 580 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 9.15 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 destination directory with CP
	207	Delete directory
RECEIVE	0	Transmission of useful data (RECEIVE-ALL) CP->CPU
	-	No function
CONTROL	0	Indicates which job is currently being executed
	A-NR	Updates the status word
RESET	0	Completely reset CP/DHB driver
	A-NR	Reset CP/DHB 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

9.4.2 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 status word of the two words of the CPU addressed by ANZW (bit nos. 8 to 11).
Table 9.16 shows you the error bits which can occur when executing the mass storage functions.

Table 9.16 Error bits of 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 (AS, RS, PY)
8H	QLAE/ZLAE too large
9H	CP 580 is write-protected (transmit job from CPU)
AH	Insufficient space on the set CP 580 drive (transmit order from CPU)
BH	File/directory not found on CP 580 (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-erasable files (job "Delete S5F files" from CPU)

Value of bit no. 8 to bit no. 11	Cause of error
Continuation of Table 9.16:	
EH	Timeout (after 50 seconds)
FH	Parameter error

9.5 Command Interpreter

9.5.1 Data Handling Blocks for the Command Interpreter

If you wish to transmit a command from a preset data block to the CP 580, 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 9.17 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 580 (SEND-DIRECT)
	0	SEND-ALL function for triggering transmission of the command to the CP 580 (SEND-ALL)

9.5.2 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. 8 to 11). Table 9.18 shows you the error bits which can occur when processing a command.

Table 9.18 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 error), error numbers PAFE 1 to PAFE 5
6H	Command line is too long
7H	Not yet realized
8H	Not used by CPSHELL
9H	The called program cannot be loaded since there is insufficient memory space in the CP 580
AH	The called MS-DOS function or the started program addresses a peripheral device of the CP 580 which is not ready, e.g. there is no floppy disk in the drive, or the printer is not ready

Contents

Instructions C79000-B8576-C204

Contents

10 Reference Section for Hardware

This chapter provides you with detailed information on the hardware design of the CP 580.

You can read:

- How the CP 580 is constructed, and what meaning the switch and jumper settings on the basic and expansion boards have
- Which devices you can connect, and how you must set up a connected printer
- How the CP 580 interfaces are assigned
- What the memory and hardware interrupt assignments of the CP 580 are.

10.1 Mechanical Construction of CP 580

The CP 580 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 5 1/3 standard slots (approx. 80 mm) wide and thus occupies four S5 slots in the main frame of the programmable controller.

The device comprises:

- the basic board
- the expansion board
- the 3.5-inch floppy disk drive
- the 3.5-inch hard disk drive

The front panel contains the displays and controls as well as the interfaces to the operation and peripheral devices. Fig. 10.1 shows you the design of the CP 580.

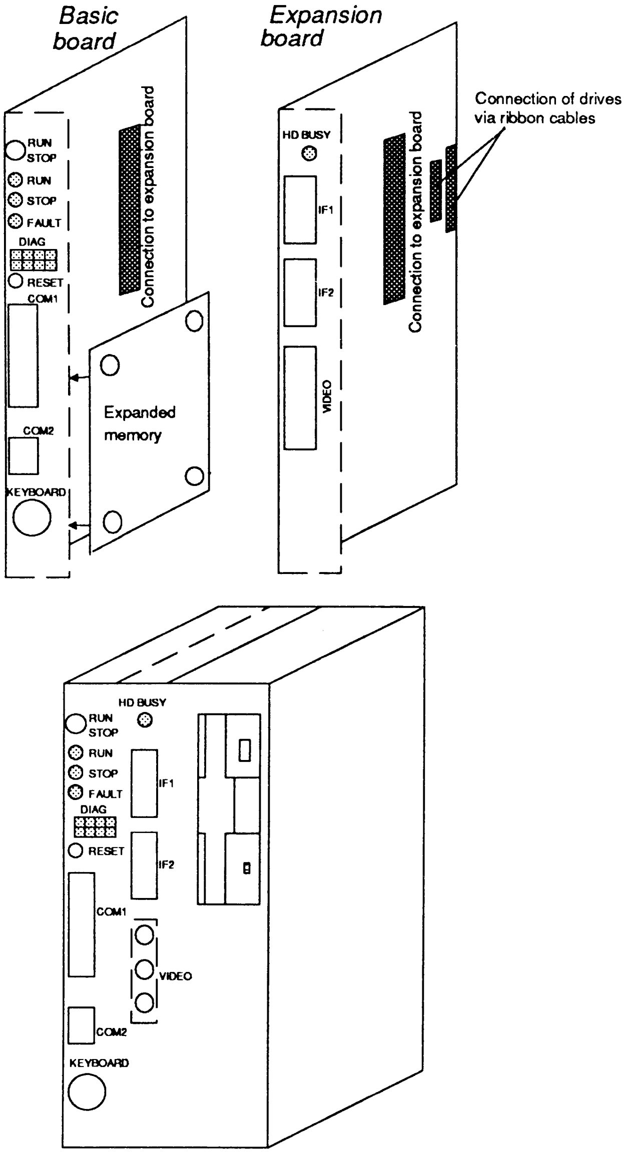


Fig. 10.1 Mechanical construction of CP 580

10.1.1 Mechanical Construction of Basic Board

The basic board comprises:

- Two 48-pin backplane connectors of series 2 for connecting the CP 580 to the S5 bus (backplane bus) of the programmable controllers (backplane connectors 1 and 2)
- One 25-way female connector (Cannon plug) for connection of a printer (COM 1)
- One 9-way female connector (Cannon plug) for connection of the optical mouse (COM 2)
- One 7-way round socket for connection of the PG 750 keyboard (KBD)
- One 128-way plug connector for connection of the expansion board
- One 96-way plug connector for connection of an expanded memory
- One 68-way socket for connection of an arithmetic processor
- One RUN/STOP switch for selection of the mode
- Diagnosis LED; RUN/STOP/FAULT LED; RESET key.

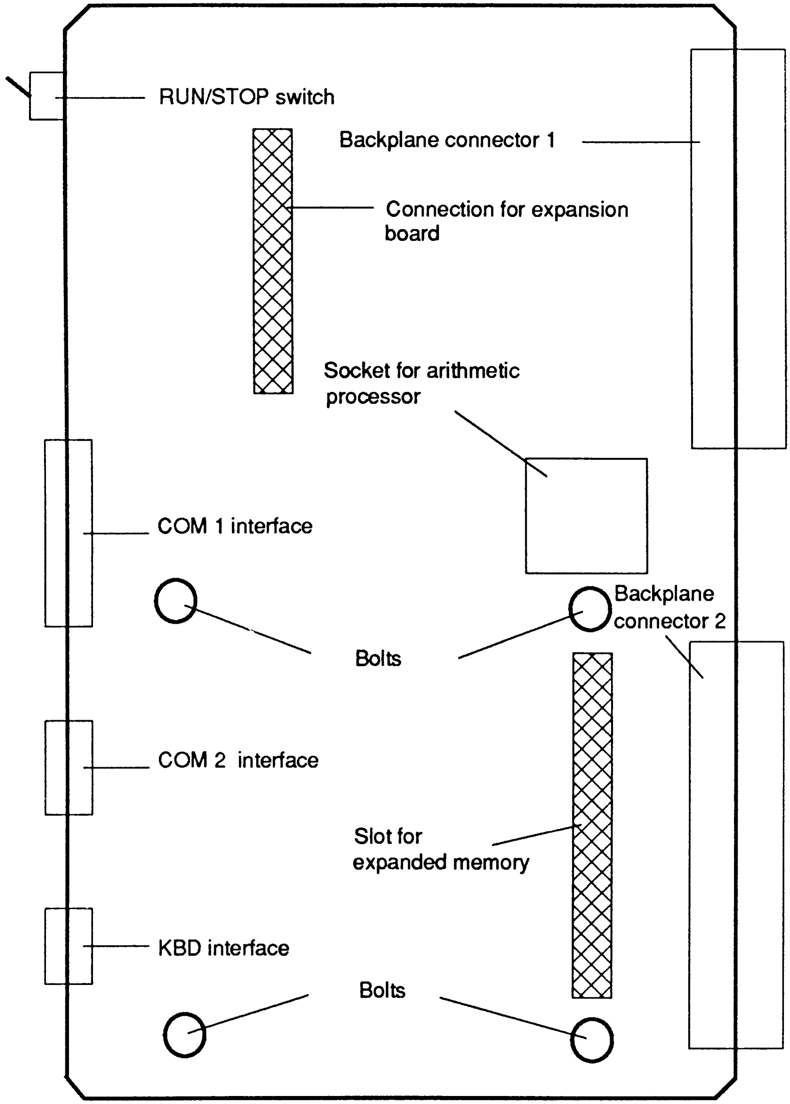


Fig. 10.2 Right-hand side of basic board

10.1.2 Mechanical Construction of Expansion Board

The expansion board comprises:

- One 48-pin backplane connector of series 2 for connecting the CP 580 to the S5 bus (backplane bus) of the programmable controllers (backplane connector 3)
- One 15-way female connector, IF1 interface (COM 3)
- One 15-way female connector, IF2 interface (COM 4)
- Three coaxial sockets for connection of the monitor
- One 128-way plug connector for connection of the basic board
- One 34-way plug connector for connection of the floppy disk drive
- One 40-way plug connector for connection of the hard disk drive
- Two 4-way plug connectors for the power supply to the floppy disk and hard disk drives
- One LED "HD busy".

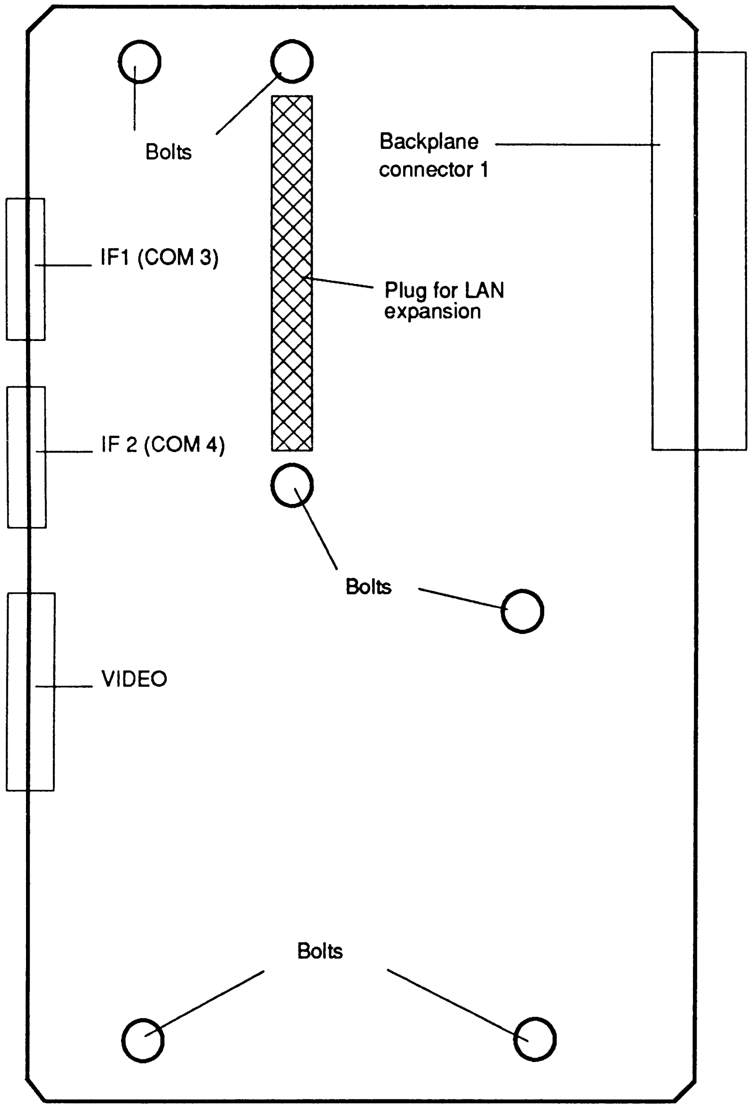


Fig. 10.3 Right-hand side of expansion board

10.1.3 Pin Assignments of Backplane Connectors

The CP 580 is connected to the wiring backplane of the programmable controller via three 48-pin backplane connectors of series 2. The pin assignments of these three connectors are shown in the following tables.

Table 10.1 Pin assignments of backplane connector 1 (top of basic board)

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

Most of the bus signals are connected to backplane connector 1 of the basic board, the module power supply is additionally connected via backplane connector 2 of the basic board and backplane connector 1 of the expansion board.

Table 10.2 Pin assignments of backplane connector 2 (bottom of basic board)

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

Table 10.3 Pin assignments of backplane connector 1 (top of expansion board)

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

10.1.4 Controls and Displays

Fig. 10.4 shows you the positions of the controls and displays on the front panel.

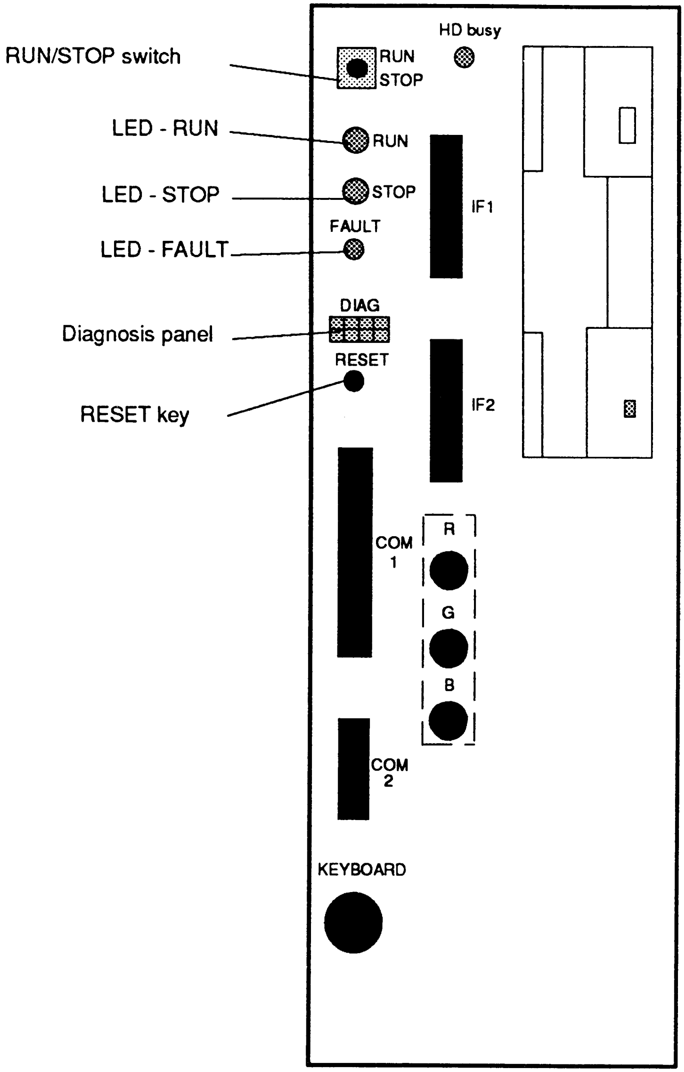


Fig. 10.4 Position of controls and displays on the front panel of the CP 580

Controls:

- **Switch** **- RUN/STOP**

  Switching from RUN to STOP:
  Bus communication at the S5 interface is disabled.
  A RESET-ALL is triggered at the CP 580 end and completely resets the CP/DHB driver. (This corresponds to triggering of a RESET-ALL by the corresponding DHB at the CPU end.)

  Switching from STOP to RUN:
  Data transfer at the S5 interface is enabled again.

- **Key** **- RESET -**
  Pressing this key completely resets the board.

Displays:

The displays are divided into:
- Operation displays
- Fault displays
- Access displays for the drives.

Operation 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 out of operation" if the mode switch is set to STOP.
  Flashing signals the status "Mode switch at RUN", but no CPU synchronized.

- **Red LED (small)** **- FAULT -**
  This LED goes off due to the BIOS during the CP restart if the S5 interface hardware is ready.

The following table shows you the meaning of the displays (assuming the CP/DHB driver is loaded):

Table 10.4 Operation and fault displays on the LEDs RUN/STOP and FAULT

FAULT LED *	RUN LED	STOP LED	Meaning
0	0	0	CP/DHB driver was correctly installed in main memory.
0	0	1 (continuous)	RUN/STOP switch in STOP position.
0	0	Flashing	RUN/STOP switch in RUN position. CP 580 ist not synchronized with any CPU.
0	1	0	RUN/STOP switch in RUN position. At least one CPU is synchronized.

* The FAULT LED goes off during the restart.
If this LED does not go off, please contact your Siemens representative.

Fault displays:

- **Row of green LEDs** **- DIAG -**

  When the CP 580 is switched on, the ROM BIOS carries out a power-on selftest. If an error occurs at the beginning of the selftest, the error code is displayed on the diagnosis LEDs via port 80H in hexadecimal format and the screen remains dark. If the selftest has continued further, an error message appears in addition on the screen.

  The significance of the diagnosis display must be interpreted as follows:

7	6	5	4
3	2	1	0

LED off = 0
LED on = 1

Error messages > 32H are internal displays of the BIOS and are of no significance to the user.

Exception:
57H appears if the CP 580 has run up correctly and MS-DOS was started.

The error codes of the fault display on the diagnosis panel have the following meaning:

Table 10.5 Meaning of error codes on the diagnosis panel

Display	Error description
01H	CPU register test running
02H	CMOS read/write error
03H	ROM BIOS checksum error
04H	Programmable timer interval error
05H	DMA initialization error
06H	DMA page register read/write error
08H	Error when checking the RAM refresh
09H	First 64K RAM test running
0AH	64K RAM chip or data line error (multi-bit error)
0BH	64K RAM odd/even logic error
0CH	64K RAM address line error
0DH	64K RAM parity error
10H	64K RAM error bit 0
11H	64K RAM error bit 1
12H	64K RAM error bit 2
13H	64K RAM error bit 3
14H	64K RAM error bit 4
15H	64K RAM error bit 5
16H	64K RAM error bit 6
17H	64K RAM error bit 7
18H	64K RAM error bit 8
19H	64K RAM error bit 9
1AH	64K RAM error bit A
1BH	64K RAM error bit B
1CH	64K RAM error bit C
1DH	64K RAM error bit D
1EH	64K RAM error bit E
1FH	64K RAM error bit F
20H	Slave DMA register error
21H	Master DMA register error
22H	Master interrupt register error
23H	Slave interrupt register error
25H	Interrupt vector is loaded
27H	Error with keyboard controller test
28H	CMOS error, checksum is generated
29H	CMOS is configured
2BH	Error when initializing screen
2CH	Error during display repetition test
2DH	Check whether video ROM present
2EH	Video ROM is started
30H	Assumption that screen is ready
31H	Assumption that black-and-white screen is ready
32H	Assumption that color monitor (40 columns) is ready
57H	No error

Access displays for the drives

Access to the drives is signalled by LEDs:

- Display for floppy disk access on the floppy disk drive
- Display for hard disk access above the interface IF1.

10.1.5 Switch and Jumper Settings on the Basic and Expansion Boards

Please refer to the installation and commissioning part in Section 2.2.2 of this Manual for the basic settings for operation of the CP 580. Figs. 10.5 and 10.8 show you the positions of the coding switches and jumpers on the basic and expansion boards. The following sections provide you with further information on the coding switches and plug-in jumpers.

10.1.5.1 Switch and Jumper Settings on the Basic Board

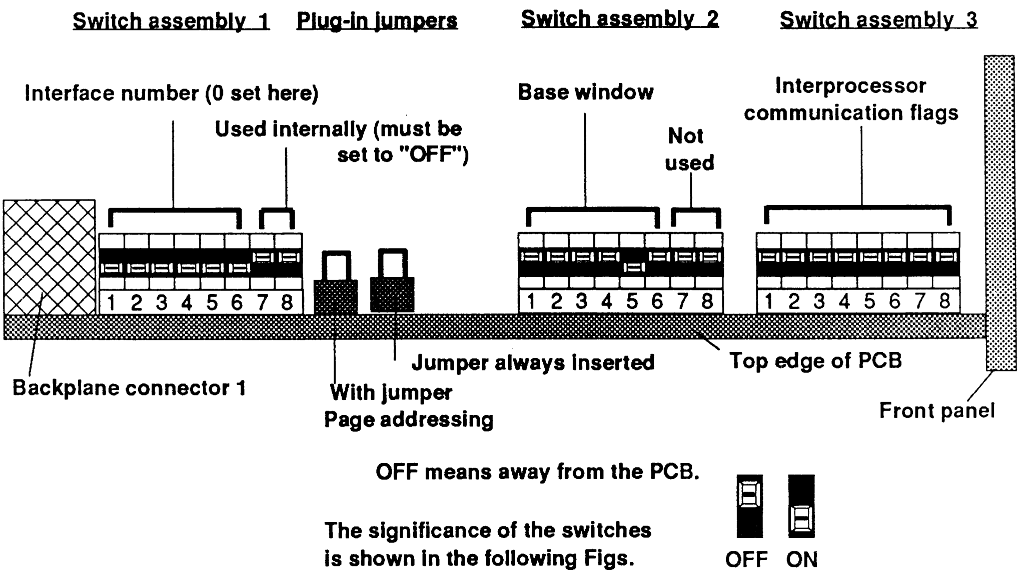


Fig. 10.5 Position of coding switches and jumpers on the basic board

The coding switches and plug-in jumpers are located along the top edge of the basic board. You can make the following settings on the basic boards:

- Setting of base interface number
- Setting of base window address
- Enabling or disabling of communication flags
- Switching from page addressing to linear addressing.

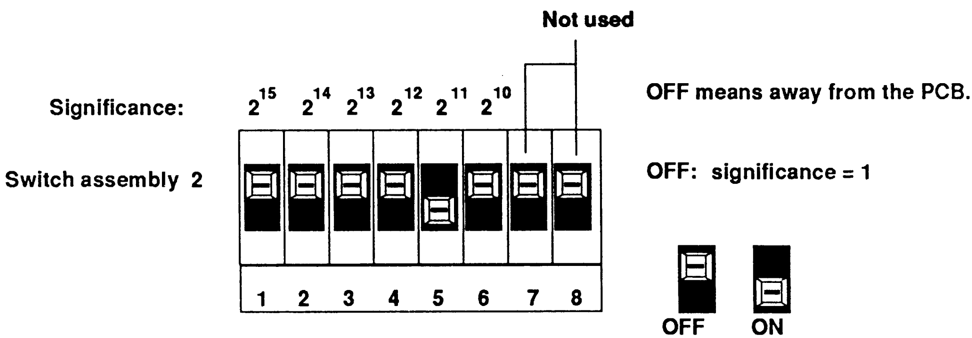
10.1.5.1.1 Base Interface Number

Refer to Section 2.2.2.1 in the installation and commissioning part of this Manual for the setting of the base interface number.

10.1.5.1.2 Setting the Base Window Address

> **Caution**
>
> The data handling blocks supplied by Siemens are matched to the specified delivery state. The factory assignments must therefore not be changed.

When selecting a page using the page selection register, this page is set into a so-called address window with a size of 1 kbyte. This is positioned as standard at F400H (61K) in the address area of the central processing unit of the programmable controller. It can be set to any other 1-K limit, however, using assembly 2 of the coding switches (see Fig. 10.6).



The address can be set in steps of 1K. This results in the above significance of switches 1 to 6 (switch 1 is assigned address 15).

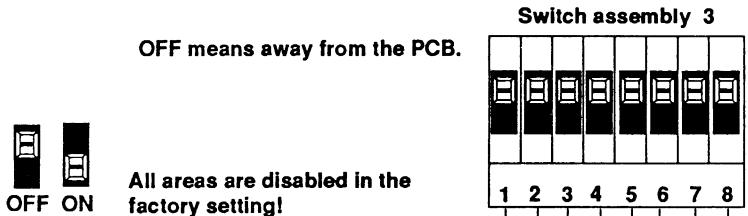
Fig. 10.6 Setting of base window address

10.1.5.1.3 Enabling or Disabling of the Interprocessor Communication Flags

All interprocessor communication flags are disabled in the factory settings since the supplied software of the CP 580 does not currently use communication flags. You can address the communication flags by means of free programming, however.

The interprocessor communication flag area comprises 256 communication flag bytes (= 2048 communication flags). The communication flags are transferred cyclically between the CPU and the communications processors and can be used for coordination.
The same applies to multi-processor operation with several CPUs in the S5-135U and S5-155U programmable controllers.
You can enable or disable the communication flags in groups of 32 flag bytes using assembly 3 of the coding switches.

- Switch ON means area enabled
- Switch OFF means area disabled.



Interprocessor communication flag bytes	Address
224 - 255	F2E0 - F2FF
192 - 223	F2C0 - F2DF
160 - 191	F2A0 - F2BF
128 - 159	F280 - F29F
96 - 127	F260 - F27F
64 - 95	F240 - F25F
32 - 63	F220 - F23F
0 - 31	F200 - F21F

Fig. 10.7 Assignment of area

10.1.5.1.4 Switching Between Page and Linear Addressing

This jumper must always be inserted when using the data handling blocks.

You can select the respective addressing mode using the plug-in jumper "Page/linear addressing" (see Fig. 10.5).

- Page addressing - jumper inserted
- Linear addressing - jumper open

10.1.5.2 Fixed Jumpers

All plug-in jumpers are inserted when the CP 580 is delivered and must not be changed.

The basic and expansion boards contain a number of plug-in jumpers which are only used for diagnostic purposes in a test bay.

10.1.5.3 Switch Settings on the Expansion Board

The coding switches are already factory-set for normal operation and must not be changed.

The coding switch assembly is on the top side of the expansion board.

The following settings can be made on the expansion board:

- Address switchover of serial interfaces IF1 and IF2 (COM 3 and COM 4)
- Interrupt switchover of serial interfaces IF1 and IF2 (COM 3 and COM 4)
- Common interrupt processing
- Information for the system via the hard disk drive.

Fig. 10.8 shows you the position of the coding switch assembly on the expansion board.

The switch settings have the following meanings:

- Switches 1 and 2 are for test purposes and must always be set to "ON" for normal operation

- Switch 3 (SEL1) is used for address switchover for the serial interfaces IF1 and IF2 (COM 3 and COM 4)

- Position OFF: IF1 is at address 03E8h - 03EFh
  IF2 is at address 02E8h - 02EFh

- Position ON: IF1 is at address 01B0h - 01B7h
  IF2 is at address 01B8h - 01BFh

- Switch 4 (SEL2) is used for interrupt switchover for the serial interfaces IF1 and IF2
  Position OFF: IF1 and IF2 use interrupt 10 (common interrupt)
  Position ON: IF1 and IF2 use interrupt 5 (common interrupt)

- Switch 5 (SEL3)
  The interrupts of the serial interfaces IF1 and IF2 are applied to a common interrupt (IRQ5 or IRQ10).
  Hardware support can be activated to process several interrupt requests should they occur simultaneously.

  Position OFF: without hardware support
  The interrupt service routine must ensure complete processing of all existing interrupts. The interrupt is blocked if an interrupt request is not processed. No further requests can then be placed by either of the two interfaces.

  Position ON: with hardware support
  An existing request is recognized following processing of an interrupt, and a further interrupt triggered. This is repeated until no further requests are present.
  Note that the hardware for recognizing further requests must be reactivated each time an interrupt is processed. This is achieved by writing and reading address 288H with any data.

- Switch 6 informs the system that the hard disk drive has a capacity of 40 Mbyte.
  Switch 6 must always be set to ON.

- Switches 7 and 8 are reserved and must be set to OFF.

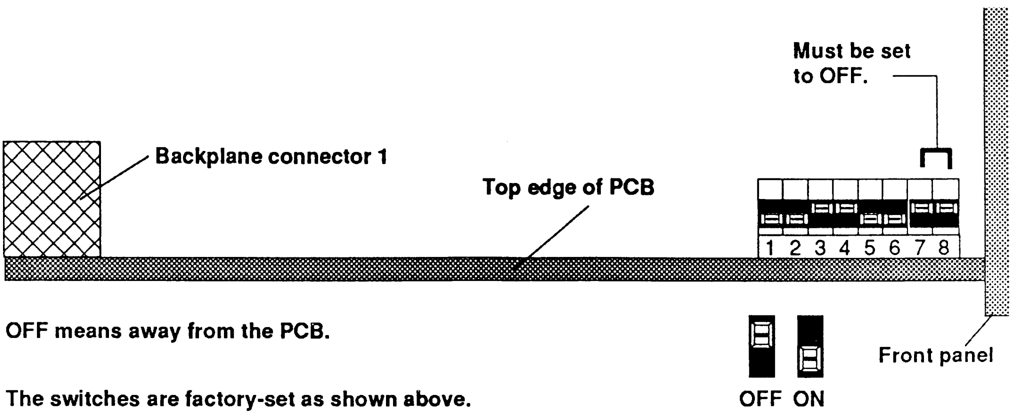


Fig. 10.8 Coding switch assembly on the expansion board

10.1.6 CP 580 Drives

The CP 580 is equipped with mass storage floppy disk and hard disk drives to store system programs, user programs and process data for subsequent evaluation. Fig. 10.9 shows you the drive positions.

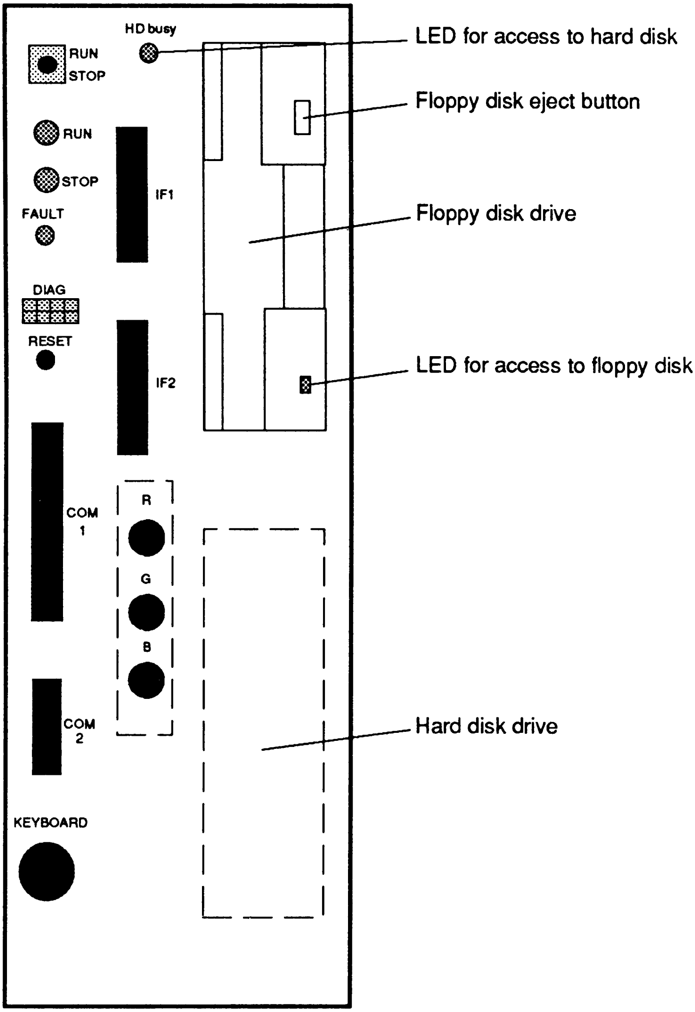


Fig. 10.9 Drive positions on the CP 580

10.1.6.1 Floppy Disk Drive

The floppy disk must not be removed when the drive lamp (LED) is on.

The 3.5-inch floppy disk drive of the CP 580 is connected to the disk interface of the 37C65 floppy controller via a 34-pin plug connector. Double-sided 3.5-inch floppy disks (80 tracks per side) can be used, either high density with 1.44 Mbytes or normal density with 720 Kbytes. The drive is automatically adapted to the type of disk by checking the HD opening on the disk.

10.1.6.2 Hard Disk Drive

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

The 3.5-inch hard disk drive of the CP 580 is connected to the AT bus via a 40-pin plug connector. The HD is equipped for this purpose with a PC/AT interface.

The hard disk has a total storage capacity of 40 Mbytes.

Logical drives can be produced using the operating system. Please refer to the description of your operating system.

The LED on the front panel of the CP 580 lights up when the hard disk drive is accessed.

10.1.7 Extension Using Device Options

There are currently no extension options.

10.2 Connection of Devices

This section describes any special considerations to be made when connecting devices, e.g. the setting of the coding switches with certain printers and selection of the printer cables.

10.2.1 Connection of a Keyboard

We recommend the connection of a PG 750 keyboard. Connect the keyboard to the KBD interface.

10.2.2 Connection of a Monitor

We recommend a monitor where the video ground is isolated from the protective ground.

It is essential to observe the installation and connection guidelines for monitors in Section 2.2.4.2!

10.2.3 Connection of PT88N/PT89N, PT88S/PT89S Printers

Recommended printers are the PT88N/PT89N, PT88S/PT89S printers or the PT10 laser printer. The PT88N/PT89N, PT88S/PT89S are connected to the CP 580 via a V.24 or TTY (20-mA current loop) interface adapter (see Section 10.2.4 for the PT10).

The printer parameters of the CP 580 are set as follows during the start-up by means of an AUTOEXEC.BAT file already contained in the standard delivery:

- Printer interface is COM 1 (the instruction C:MODE LPT1:=COM 1: is present in the AUTOEXEC.BAT file)

- Printer parameters: 9600 baud, no parity, 8 data bits, 1 stop bit, p (the instruction C:MODE COM 1:96,n,8,1,p is present in the AUTOEXEC.BAT file).
  p means: A printer with a serial interface is connected.

Before you connect the printer to the CP 580, you must set the coding switches on the central controller of the printer or on the interface adapter.

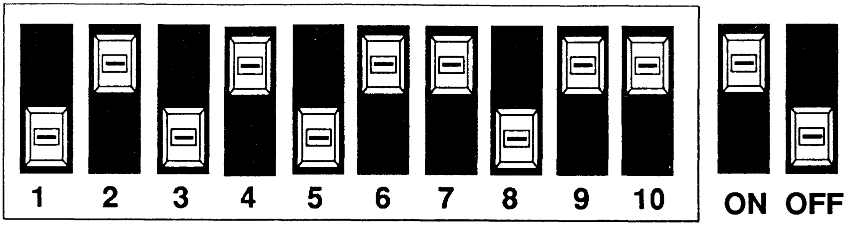
Refer to the description of your printer for more details on the meaning of the switch positions.

10.2.3.1 Setting the Coding Switches on the Central Controller

With the PT88N/PT89N and PT88S/PT89S printers you can select certain standard functions and the character set.

Use a coding switch for this purpose. The coding switch is accessible by opening the front flap of the housing. Two coding switches are present here in the PT88S/PT89S printers.

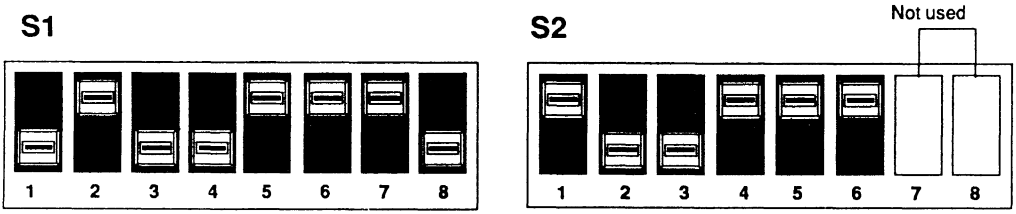
Figs. 10.10 and 10.11 show you how to set the coding switches.



PT88N/PT89N

**(German keyboard, LF = CR + LF , CR = CR,
form length 12", line feed 1/6", 80 character/line)**

Fig. 10.10 Coding switches on the central controller with PT88N/PT89N



PT88S/PT89S

**(German keyboard, CR = CR, letter spacing 1/10", normal character font,
number 0 printed without slash, form length 12", line feed 1/6",
no paper feed at end of defined form length, printer always selected,
8th bit = 1; bit 8 is evaluated)**

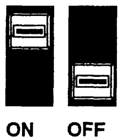


Fig. 10.11 Coding switches on the central controller with PT88S/PT89S

10.2.3.2 Setting the Coding Switches on the Interface Adapter

The following interface adapters are available since the printers can be connected either via a V.24 or TTY interface using the serial interface COM 1 of the CP 580.

- Interface adapter SAP-S1 (V.24/V.28)
- Interface adapter SAP-S2 (TTY/20 mA)
- Interface adapter SAP-S3 (TTY/20 mA; V.24)

 C79000-B8576-C204-01

- **Interface adapter SAP-S1 (V.24/V.28)**

The mode switches S1 and S2 on the interface adapter SAP-S1 must be set as follows for the PT88N/PT89N and PT88S/PT89S printers. Refer to Fig. 10.12.

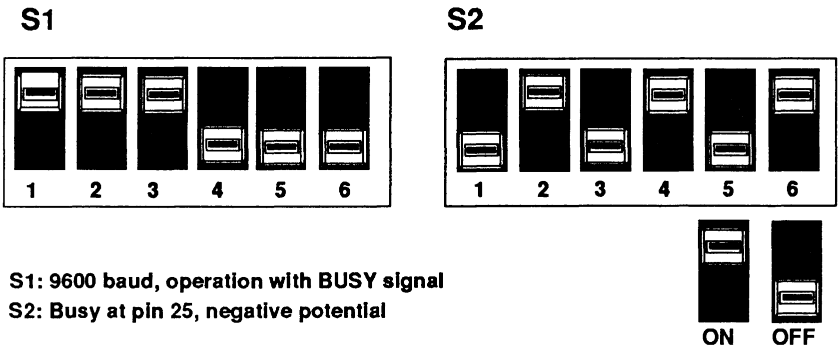


Fig. 10.12 Setting of mode switches

Fig. 10.13 shows you the position of the mode switches on the interface adapter SAP-S1.

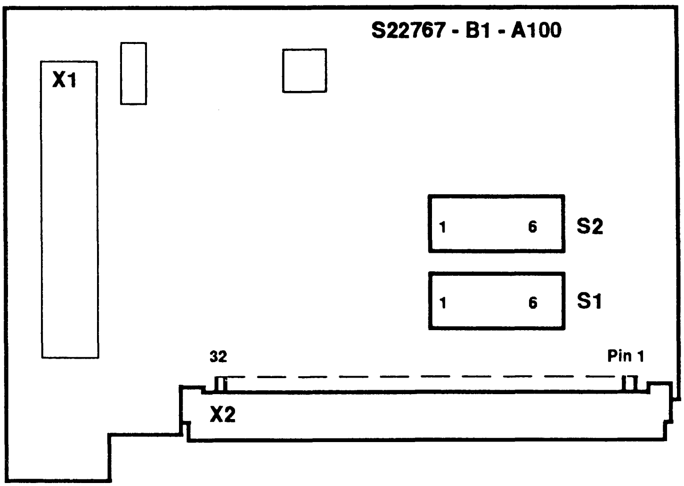
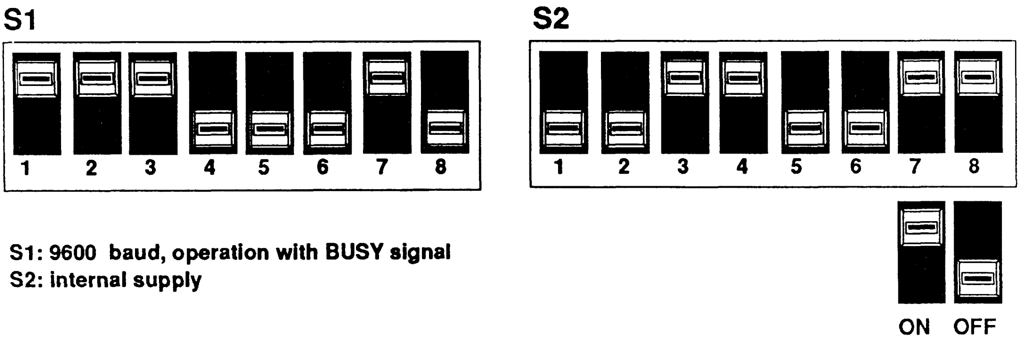


Fig. 10.13 Mode switches on the interface adapter SAP-S1

- **Interface adapter SAP-S2 (TTY/20 mA)**

The mode switches S1 and S2 on the interface adapter SAP-S2 must be set as follows for the PT88N/PT89N and PT88S/PT89S printers. Refer to Fig. 10.14.



S1: 9600 baud, operation with BUSY signal
S2: internal supply

Fig. 10.14 Setting of mode switches

Fig. 10.15 shows you the position of the mode switches on the interface adapter SAP-S2.

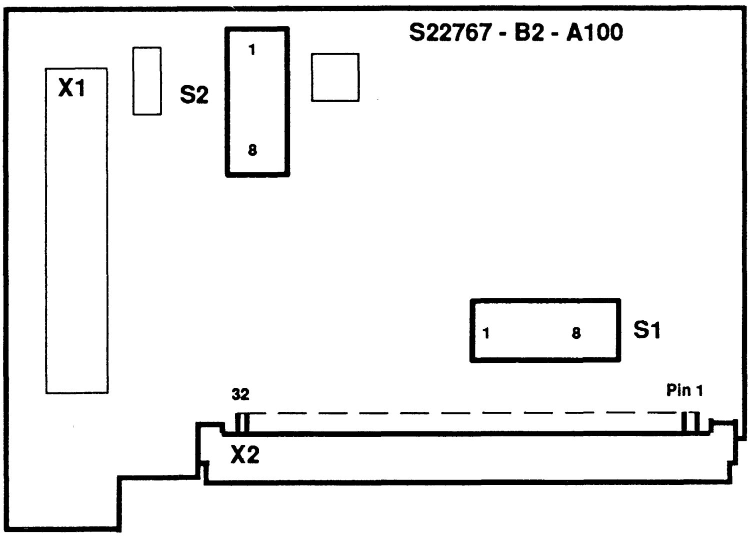
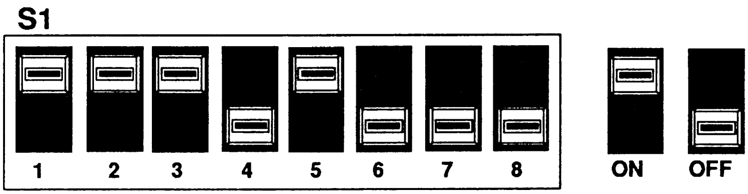


Fig. 10.15 Mode switches on the interface adapter SAP-S2

- **Interface adapter SAP-S3 (TTY/20 mA; V.24)**

The mode switches S1 to S4 on the interface adapter SAP-S3 must be set as follows for the PT88N/PT89N and PT88S/PT89S printers. Refer to Fig. 10.16.

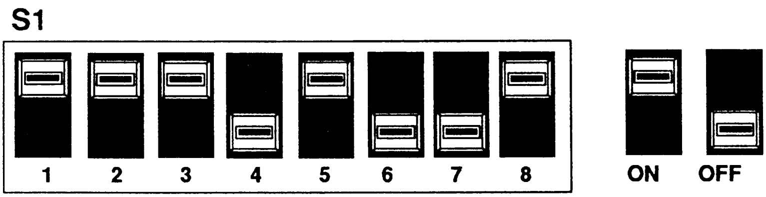
S4 in position 1 = V.24



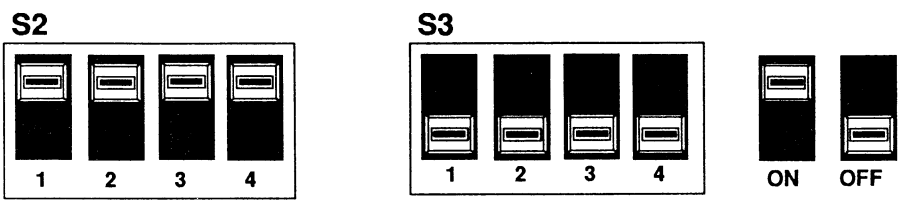
S1: 9600 baud, BUSY signal on pin 25, BUSY means negative potential

Note: **The mode switches S2 and S3 are only significant with TTY operation**

S4 in position 2 = TTY



S1: 9600 baud, BUSY with current



S2 and S3: internal supply

Fig. 10.16 Setting of mode switches

There are two different settings:
Mode switch S4 in position 1 = V.24 interface
Mode switch S4 in position 2 = TTY interface

Mode switch S4 - setting of interface version (see Table 10.6)

Table 10.6

Interface	Position
RS-232-C (V.24/V.28)	1
TTY	2

Fig. 10.17 shows you the position of the mode switches on the interface adapter SAP-S3.

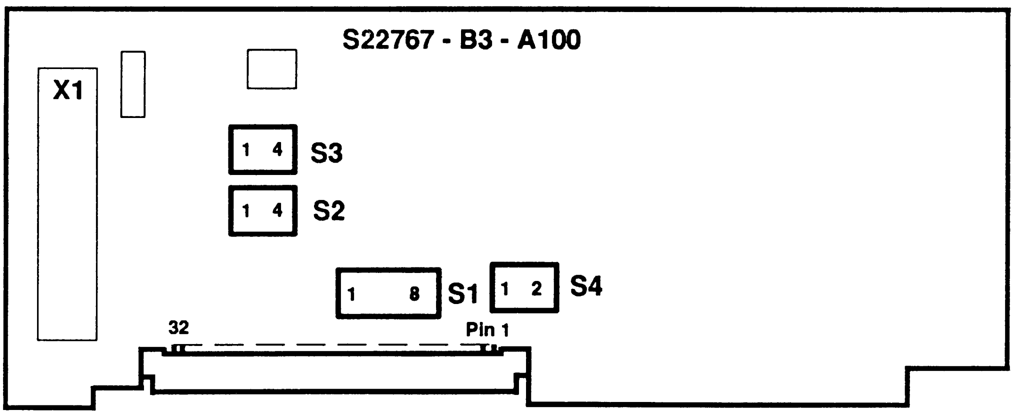


Fig. 10.17 Mode switches on the interface adapter SAP-S3

10.2.4 Connection of PT10 Laser Printer

You can connect the PT10 laser printer to V.24 interface of COM 1. It is set using the menu key in the offline status.
Further information is contained in the instructions for your printer.

Recommmended printer setting:

SYMSET= ROMAN 8
AUTO CONT= OFF
I/O= SERIAL
BAUDRATE= 9600
ROBUST XON= OFF
DTR POLARITY= HI

10.2.5 Selection of Cable Connectors for the Printer Connection

Standard cable connectors available with a variable length can be used for connecting the printer.

Note the maximum cable length with the V.24 and TTY/20-mA interface versions. Refer to Section 2.2.4.5.

Example of a simple null modem cable for connection to the V.24 interface of the COM 1 of the CP 580.

CP 580		Cable		PT88/PT89, PT88S/PT89S
Cannon plug connector, 25-pin				Cannon plug connector, 25-pin
Connector housing/screen	1		1	Connector housing/screen
RxD	3		2	TxD
TxD	2		3	RxD
			4	RTS
			5	CTS
DTR	20		6	DSR
			22	RI
Signal GND	7		7	Signal GND
RTS	4			
CTS	5			
DSR	6		25	DTR
RI	22			

Fig. 10.18 Null modem cable for connection to the V.24 interface of COM 1

Example of a cable for connection to the TTY interface of the COM 1 of the CP 580.

CP 580		Cable		PT88/PT89, PT88S/PT89S
Cannon plug connector, 25-pin				Cannon plug connector, 25-pin
Connector housing/screen	1	———	1	Connector housing/screen
+RxD	9	———	21	+20mA
- RxD	10	———	18	+TxD
+TxD	18	———	10	+20mA
-TxD	19	———	9	+RxD

Fig. 10.19 Printer cable for connection to the TTY interface of COM 1

10.2.6 Connection of a Mouse

The CP 580 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 580.

We recommend the PG 750 mouse with board. The cursor on the screen follows the movement of the mouse on the board. Thus individual operations and functions can be selected and executed simply (by clicking). To ensure that the mouse movements are converted correctly to cursor movements, the board must be positioned sideways underneath the mouse.
Refer to Fig. 10.20.

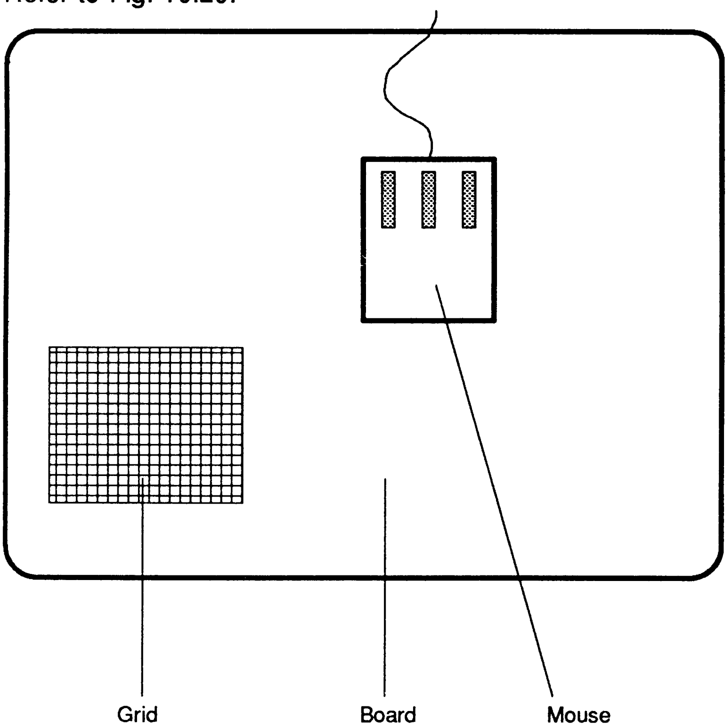


Fig. 10.20 Position of optical mouse on the board

The mouse can be connected to your CP 580 in the following manner:

- Via the serial interface COM 2 on the front panel of the CP 580
- Via the mouse plug fitted on the side of the PG 750 keyboard.

You must only use one of the two interfaces, since the COM 2 interface is also occupied when the PG 750 keyboard is connected!

Note

Other pointing devices can also be connected to the interface COM 2.

If you wish to connect a mouse from another manufacturer, note that not every make can be used at the keyboard interface!

10.3 Connector Interfaces of the CP 580

The interfaces on the front panel of the CP 580 are used to connect all operation and peripheral devices.

Observe the following when connecting the devices:

- The interfaces COM 1 and COM 2 have locking screws to secure the plug contacts.

- The interfaces IF1 and IF2 have a sliding lock to secure the plug contacts.

The interfaces IF1 and IF2 (COM 3 and COM 4) are not supported by the CP 580 software.

10.3.1 Serial Interface COM 1

The 25-pin plug designated COM 1 has the standard pin assignments of the V.24 transmission signals plus the signals for active TTY operation (20 mA). The COM 1 interface is compatible with the industrial standard. It can be used to connect printers with serial interfaces.
The control signals required for modem control are present on the 25-pin plug.

Table 10.7 Pin assignments of serial interface COM 1, V.24/TTY assignment

1	Screen	2	TXD (D1/103)
3	RXD (D2)	4	RTS (S2/105)
5	CTS (M2)	6	DSR (M1/107)
7	GND (E2)	8	DCD (M5/109)
9	TTY + RxD	10	TTY - RxD
11	-	12	-
13	-	14	-
15	-	16	-
17	-	18	TTY + TxD
19	+24V floating	20	DTR (S1)
21	TTY - TxD	22	RI (M3)
23	-	24	-
25	-	-	-

10.3.2 Serial Interface COM 2

The 9-pin plug designated COM 2 also has pin assignments for signals to the V.24 standard. Like COM 1, the interface is compatible with the industrial standard. The interface which is usually for connection of a mouse has modem control signals in addition to the V.24 data, transmit and receive lines.

Table 10.8 Pin assignments of serial interface COM 2, V.24 assignments

1	DCD (M5)
2	RxD (D2)
3	TxD (D1)
4	DTR (S1)
5	GND
6	DSR (M1)
7	RTS (S2)
8	CTS (M2)
9	RI (M3)

10.3.3 Keyboard Interface KBD

A 7-pin round socket is provided to connect a PG 750 keyboard. As an extension to the 5-pin standard, it is additionally fitted with the data lines for connecting a mouse.

Table 10.9 Pin assignments of serial interface KBD, TTL assignments

1	Clock
2	Data
3	Ground (+5 V)
4	Ground (+5 V)
5	+5 V
6	TxD (mouse/COM 2)
7	RxD (mouse/COM 2)

10.3.4 Serial Interface IF1 (COM 3)

This interface is not supported by the software.

The 15-pin plug designated IF1 is designed as a serial, asynchronous TTY interface (20-mA current loop) and conforms with the special SIMATIC requirements (pin assignments).

Table 10.10 Pin assignments of serial interface IF1, TTY assignments

1	Ground (Mext.)
2	-RxD
3	-
4	+24 V
5	Plug code
6	+TxD
7	-TxD
8	Ground (Mext.)
9	+RxD
10	+24 V ground
11	Current source 20 mA (T)
12	5 V ground
13	Current source 20 mA (R)
14	-RxD (mouse/COM 2)
15	+RxD (mouse/COM 2)

10.3.5 Serial Interface IF2 (COM 4)

This interface is not supported by the software.

The 15-pin plug designated IF2 is designed as an X.27 (RS 422) interface.

Table 10.11 Pin assignments of serial interface IF2, X.27 (RS 422) assignments

1	Ground (Mext.)
2	RxD (B)
3	+5 V
4	-
5	Plug code
6	+TxD
7	-TxD
8	5 V ground
9	+RxD
10	/Reset hardware
11	-
12	5 V ground
13	-
14	-RxD (mouse/COM 2)
15	+RxD (mouse/COM 2)

10.3.6 Video Outputs

The three coaxial sockets on the front panel of the CP 580 are used to connect a monitor via double-screen coaxial cables (triax cables). They are assigned the following signals:

- Red (R)
- Green/SYNC signal (G/S)
- Blue (B)

Caution

The sockets are non-floating.

10.4 Memory Division and Hardware Interrupts of the CP 580

10.4.1 Memory Division

In the assignment of the address areas, a differentiation is made between
- the memory address area and
- the I/O address area.

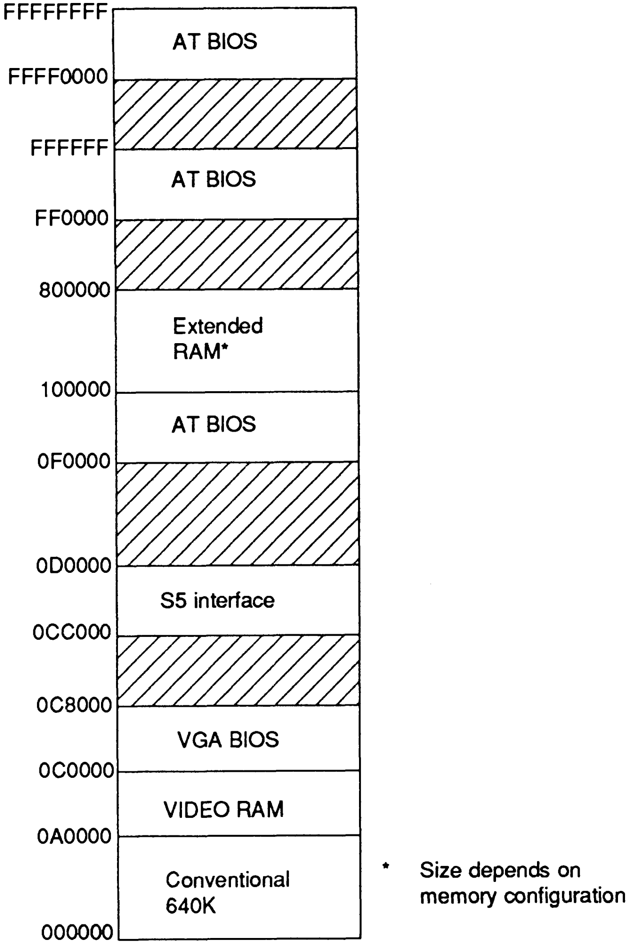


Fig. 10.21 Assignment of memory address area

Table 10.12 Assignment of I/O address area

I/O address assignment	Component/module	Hardware interrupt used
0000H-001FH	DMA controller 1 (8237)	
0020H-0021H	Interrupt controller 1 (master/8259A)	
0040H-005FH	Timer (8254)	IRQ 0
0060H&0064H	Keyboard controller (8042)	IRQ 1
0061H-006FH	Port B : timer/loudspeaker/parity system bus/ RAM	NMI
0070FH-007FH	Real-time clock, NMI enable	IRQ 8
0080FH-008FH	DMA page register	
0092H	PORT A: configuration	
00A0H-00BFH	Interrupt controller 2 (slave/8259A)	IRQ 2
00C0H-00DFH	DMA controller 2 (8237)	
00E0H-00EFH	Configuration ASICs	
00F0H-00FFH	Numeric processor (80387 SX)	IRQ 13
0100H-010FH	Reserved (SINEC H1 interface)	IRQ 12
0110H-016FH	Not used	
0170H-0177H	Reserved (Winchester controller 2)	
0178H-01EFH	Not used	
01F0H-01FFH	Winchester controller 1	IRQ 14
0200H-020FH	Reserved (game I/O)	
0210H-0277H	Reserved (e.g. RAM page)	
0278H-027FH	Reserved (parallel interface 2)	
0280H-02AFH	Reserved (e.g. RAM page)	
02B0H-02DFH	Reserved (color graphics 2/EGA)	
02E0H-02F7H	Reserved	

I/O address assignment	Component/module	Hardware interrupt used
02E8H-02EFH	IF2 presetting (switch 3)	IRQ10
02F8H-02FFH	Serial interface 2 (COM 2)	IRQ 3
0300H-031FH	Reserved (prototype card)	
0320H-033FH	Not used	
0340H-035FH	Reserved (HIGRAPH (CPU) host interface)	IRQ 11
0360H-036FH	Reserved (IBM PC NET)	
0370H-0377H	Reserved (floppy controller 2)	
0378H-037FH	Reserved (parallel interface 1)	IRQ 7
0380H-038FH	Reserved	
0390H-039FH	Reserved (SINEC H2 interface)	IRQ 12
03A0H-03AFH	Reserved	
03B0H-03BFH	B/W monitor interface or EGA/VGA	
03C0H-03CFH	Color graphics interface EGA/VGA	IRQ 9
03D0H-03DFH	Color graphics interface CGA/EGA/VGA	
03E0H-03EFH	Not used	
03E8H-03EFH	IF1 presetting (switch 3)	IRQ 10
03F0H-03F7H	Floppy controller	IRQ 6 DRQ/DACK 2
03F8H-03FFH	Serial interface 1 (COM 2)	IRQ 4
0400H-4FFFH	Reserved	
5000H-55FFH	S5 interface	IRQ 15
5600H-FFFFH	Reserved	

10.4.2 Hardware Interrupts

Table 10.13 Summary of hardware interrupts

NMI	RAM parity; I/O channel parity, mains failure
IRQ 0	82 C 84 timer 0
IRQ 1	Keyboard controller
IRQ 2	Interrupt controller 2
IRQ 3	Serial interface 2 (COM 2)
IRQ 4	Serial interface 1 (COM 1)
IRQ 5	free
IRQ 6	Floppy controller 1
IRQ 7	Parallel interface 1
IRQ 8	Clock, internal to I/O controller
IRQ 9	EGA/VGA
IRQ 10	Serial interface 3 (IF1) / serial interface 4 (IF2)
IRQ 11	Reserved (HIGRAPH)
IRQ 12	Reserved (LAN AS)
IRQ 13	Numeric processor
IRQ 14	Winchester controller 1
IRQ 15	S5 coupling

10.5 Conversion and Repairs

Note

Do not carry out any conversions!
Conversions must only be carried out by Siemens or by companies authorized by Siemens.

Contents

Contents

11 Technical Data of CP 580

11.1 Device-specific Data

Weight	approx. 3 kg
Module format	Double Eurocard format (160 mm x 233.4 mm)
Front panel width	81.28 mm (5 1/3 standard slots) Four S5 slots
Backplane connector	ES 902, series 2, 48-pin
Front plug	
COM 1	Female connector, 25-pin
COM 2	Male connector, 9-pin
KBD	Round socket, 7-pin
IF1 (COM 3)	Female connector, 15-pin
IF2 (COM 4)	Female connector, 15-pin
VIDEO	3 x coaxial sockets (color)

11.2 Power Supply

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

11.3 Current Consumption

Current consumption	+5 V: typ. 5.5 A max. 7 A during restart (approx. 5 s) +24 V: max. 0.1 A (without peripheral devices)
Back-up current (min. 2.7 V)	typ. 0.03 mA

11.4 Safety

VDE regulations	IEC 950 = EN 60 950 = DIN VDE 0805
Degree of protection	IP 00 to IEC 529

11.5 Electromagnetic Compatibility (EMC)

Radio interference suppression	To CISPR 11
Limit class	A

Noise immunity:

Conducted interference	To DIN VDE 0843 Part 4 = IEC 65(CO)39-1985(TC65)
on signal cables	To IEC 801-4 (burst): 1 kV
Noise immunity to discharges of static electricity	To IEC 801-2 (ESD): 4 kV
Noise immunity to external fields	To IEC 801-3: 3 V/m

11.6 Climatic Conditions

Temperature:	Tested to DIN IEC 68-2-1/2
Operation	+5 to +50 °C (+41 to +122 °F)
Storage/transport	- 20 to +60 °C (- 4 to +140 °F)
Temperature gradient:	
Operation	max. 10 K/h
Storage/transport	max. 10 K/h
Relative humidity:	
Operation	8 to 80 % at 25 °C (+77 °F), no condensation
Storage/transport	8 to 80 % at 25 °C (+77 °F), no condensation

Altitude (referred to mean sea level):

Operation	-50 m to +2500 m
Storage/transport	Up to 10000 m

11.7 Mechanical Environmental Conditions

Oscillations:	Tested to IEC 68-2-6
Operation	10-60 Hz: amplitude 0.035 mm;
	60-500 Hz: acceleration 4 m/s^2 (0.4 g)
Transport	5-9 Hz: amplitude 3.5 mm
	9-500 Hz: acceleration 20 m/s^2 (2 g)

Shock:	Tested to IEC 68-2-27
Operation	Half-sine: 50 m/s^2 (5 g), 11 ms
Transport	Half-sine: 500 m/s^2 (50 g), 11 ms

11.8 Logic Parameters

Processors:

32-bit microprocessor	80386SX, 20 MHz
Coprocessor	80387SX, 20 MHz
System controller	VL 82C320
ISA bus controller	VL 82C331

Memory configuration:

Main memory configuration	2-Mbyte DRAM (with/without coprocessor)
also available as version with	4-Mbyte DRAM (with/without coprocessor)
and	8-Mbyte DRAM (with/without coprocessor)

Mass storage:

Hard disk drive

Capacity	40 Mbyte
Mean access time	typ. 28 ms
MTBF	typ. 20000 h
START/STOP	> 10000 cycles

Floppy disk drive

Capacity	3.5 "/1.44 Mbyte
MTBF	10000 h

Contents

Instructions C79000-B8576-C204

Reference Literature

/1/ MS^{TM}-DOS
Operating System
for SICOMP PC and SIMATIC S5
Description

Order No. C79000-G8776-C426

/2/ Brian W. Kernighan
Dennis M. Ritchie:
The C Programming Language

Englewood Cliffs: Prentice Hall 1988

/3/ Hans Berger:
Automating with the SIMATIC S5-135U

SIEMENS AG
ISBN 3-8009-1562-6

/4/ Programmable Controllers
Basic Concepts

SIEMENS AG
ISBN 3-8009-8032-0

/5/ STEP® 5 Basic Package
Manual

Order No. 6ES5 998-0SC21

/6/ Catalog ST 59: SIMATIC S5
Programmers

Order No. E 86060-K4659-A101-A1-7600

/7/ Catalog ST 54.1: S5-135U, S5-155U and S5-155H
Programmable Controllers

Order No. E 86010-K4654-A111-A6-7600

/8/ S5-115U
Programmable Controller
Manual

Order No. 6ES5 998-0UF22

/9/ SIMATIC S5
S5-135U (CPU 928B)
Programmable Controller
Manual

Order No. 6ES5 998-2UL21

/10/ SIMATIC S5
S5-155U
Programmable Controller
CPU 946/947
Manual

Order No. 6ES5 998-0UM22

/11/ S5-135 U Programmable Controller
Handling Blocks for R Processor and
CPU 928

Order No. C79000-G8576-C366

/12/ Handling Blocks
Standard Function Blocks
CPU 946/947
S5-155U Programmable Controller

Order No. C79000-G8563-C572

/13/ Catalog ST 52.3: S5-115U and S5-115H
Programmable Controllers

Order No. E 86010-K4652-A431-A1-7600

/14/ Catalog ST 80: COROS
Operator Control and
Process Monitoring Systems

Order No. E 86060-K4680-A101-A1-7600

/15/ Ray Duncan:
Advanced MS-DOS

Redmond, Wash.: Microsoft Pr. 1986

Contents

Abbreviations

A-NR	Job number
ANZW	Status word
BIOS	Basic input output system
BLGR	Frame size
CP	Communications processor
CP/DHB driver	Driver for data handling blocks
CPU	S5 CPU
DHB	Data handling block
DMA	Direct memory access
DPR	Dual-port RAM
EMC	Electromagnetic compatibility
FD	Floppy disk
HD	Hard disk
INTxx	Software interrupt interface to CP/DHB driver
IRQxx	Hardware interrupt
KBD	Keyboard
NN	Source/destination parameter of CP
PAFE	Parameter assignment error display

PLC	Programmable controller
PG	Programmer
RW	Read/write
SSNR	Interface number
TCB	Transfer control block
TSR program	Terminate and stay resident program
XX	Indirect parameter assignment

Index

Contents

Instructions C79000-B8576-C204

14 Ordering Information

In this chapter
you can find the Order Nos. of the products mentioned or described in this Manual.

CP 580 configurations

Designation	Order No.
- 2-Mbyte main memory, without coprocessor - 2-Mbyte main memory, with coprocessor	6ES5 580-4UA11 6ES5 580-5UA11
- 4-Mbyte main memory, without coprocessor - 4-Mbyte main memory, with coprocessor	6ES5 580-0UA11 6ES5 580-1UA11
- 8-Mbyte main memory, without coprocessor - 8-Mbyte main memory, with coprocessor	6ES5 580-2UA11 6ES5 580-3UA11

Language-specific Order Nos. of the CP 580 Manuals

Designation	Order No.
- German	6ES5 998-1AT11
- English	6ES5 998-1AT21
- French	6ES5 998-1AT31
- Spanish	6ES5 998-1AT41
- Italian	6ES5 998-1AT51

S5-115U programmable controller

Designation	Order No.
Adapter casing for 4 slots	In preparation
Fan subassembly for CR 700-0LB subrack: - For supply voltage AC 115/230 V - For supply voltage DC 24 V Fan subassembly for CR 700-3 subrack: - For supply voltage AC 115/230 V - For supply voltage DC 24 V	 6ES5 981-0HB11 6ES5 981-0HB21 6ES5 981-0HA11 6ES5 981-0HA21

Monitors

Designation	Order No.
Office version (no isolation between electronics ground/protective ground!): - 14-inch multimode color monitor, degree of protection IP 20 plus the required BNC-Cannon adapter - 16-inch multimode color monitor, degree of protection IP 20 - 19-inch multistandard color monitor, degree of protection IP 20	 6AV1 472-0AA00 6AV1 908-0BA00 6AV1 473-0AA00 6AV1 475-0AA00
Industrial version (isolation between electronics ground/protective ground): - 19-inch multistandard color monitor, degree of protection IP 30 You must also order: - Ground clamping strip - Tilting console	 6AV1 414-0AA00 6AV1 901-0AL00 6AV1 901-0AK00

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 736-2...0

Length codes for cable 6ES5...

Length of cable connector	Order No. extension
	6ES5 ...-. - - - 0 ↑↑↑
1.0 m	**BB0**
1.6 m	**BB6**
2.0 m	**BC0**
2.5 m	**BC5**
3.0 m	**BD0**
3.2 m	**BD2**
5.0 m	**BF0**
8.0 m	**BJ0**
10.0 m	**CB0**
12.0 m	**CB2**
16.0 m	**CB6**
20.0 m	**CC0**
25.0 m	**CC5**
32.0 m	**CD2**
40.0 m	**CE0**
50.0 m	**CF0**

Note

You can obtain more information from the Catalogs ST 52.3, ST 54.1 and ST 80 and in the Manuals for the respective programmable controllers. (See Reference Literature.)